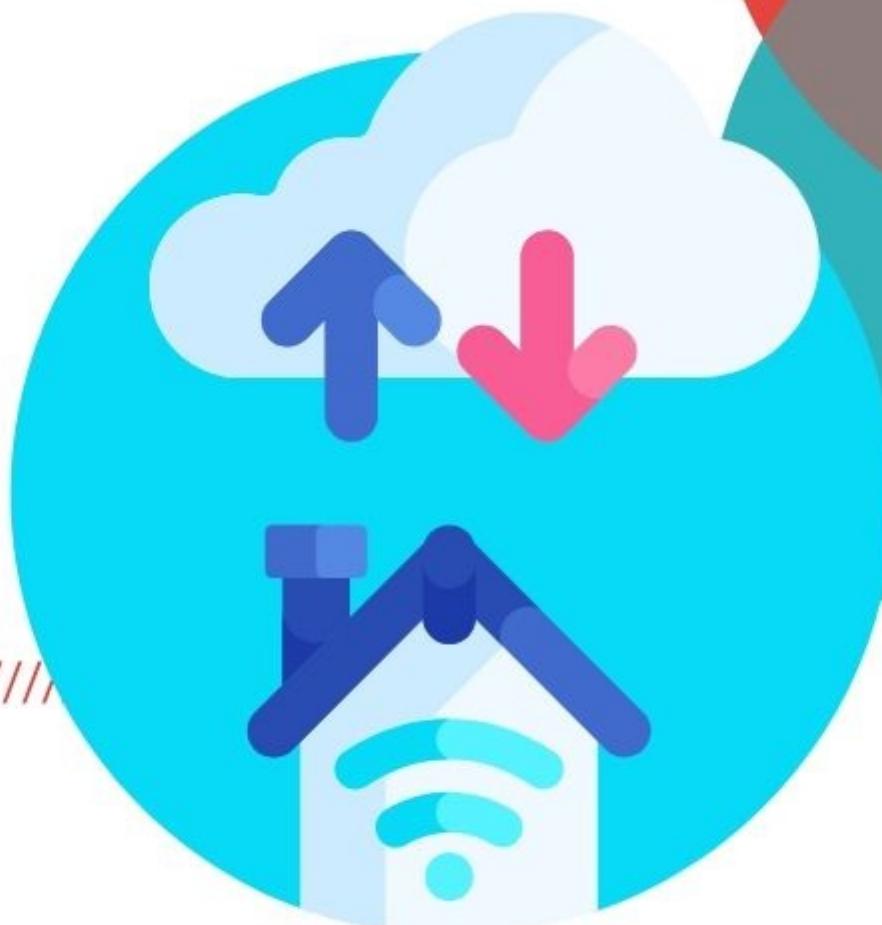


DOMÓTICA CON ARDUINO



CENTRO ARAGONÉS de TECNOLOGÍAS para la EDUCACIÓN

LICENCIA Y AUTORÍA:
VER CRÉDITOS

Tabla de contenido

Introducción	1.1
--------------	-----

1. Fundamentos Domótica con Arduino

1.1 ¿Qué es la Domótica con Arduino?	2.1
1.2 ¿Qué tengo que saber sobre Arduino?	2.2
1.3 Kit de préstamo de CATEDU	2.3

2 Programación en bloques mBlock

2.1 Programación mBlock	3.1
2.2 Preparando mBlock y Arduino	3.2
2.3 INTERRUPTOR CREPUSCULAR	3.3
2.4 APERTURA PUERTA	3.4
2.5 ALARMA AGUA	3.5
2.6 JOYSTICK	3.6
2.7 ALARMA	3.7

3 Programación en código

3.1 Entorno de programación	4.1
3.2 Maqueta	4.2
3.3 APERTURA DE PUERTA	4.3
3.4 JOYSTICK	4.4
3.5 ALARMA AGUA	4.5
3.6 FUEGO	4.6
3.7 LÁSER	4.7
3.8 BLUETOOTH	4.8

4. Para saber más...

4.1- Pensamiento computacional	5.1
4.2 Grupo Robótica Educativa	5.2
4.3 Muro	5.3
Créditos	5.4

DOMOTICA CON ARDUINO

Utilizaremos las posibilidades del **Arduino** para crear esta maqueta de domótica. La programación será de dos maneras:

- Gráfica con bloques para primaria.
- Textual por código para secundaria.

Conocimientos previos

Para la parte de programación en bloques, **ES NECESARIO QUE SEPAS LOS CONCEPTOS BÁSICOS DE SCRATCH**: Scratch

Para la parte de programación en código **ES NECESARIO QUE SEPAS LOS COMANDOS BÁSICOS DE : Código Arduino**.

No obstante, si no lo tienes reciente, te recordaremos lo esencial.



Vector de fondo creado por macrovector - www.freepik.es

1.1 ¿Dónde encaja este curso?

Arduino es una placa electrónica abierta para la creación de prototipos, muy flexible por lo tanto lo tenemos en varias propuestas de nuestra hoja de ruta de [Pensamiento computacional](#)

Si te fijas en [esa hoja de ruta](#), se puede clasificar los cursos en:

- Utilización del Arduino con una **programación gráfica con bloques** (nivel Primaria):
 - [Nuestra propuesta con la Shield Echidna](#)
 - [Arduino con mBlock, sin Shield](#)
- Utilización del Arduino sin una programación en **código** (nivel Secundaria-Bach-FP)
 - [Arduino con código, con la Shield Edubásica o sin Shield](#)

Este curso encaja en los dos, con bloques y con código para poderse utilizar tanto en primaria como hasta niveles de bachillerato. Este curso utiliza una pequeña Shield pero es innecesaria si utilizas una placa Protoboard normal, ver 1.1.4.

1.1.1 ¿Cual es el objetivo?

Nuestra propuesta es utilizar el Arduino con módulos adaptados que permitan recrear una casa inteligente y utilizando las dos posibilidades.

Mira el siguiente vídeo para que veas qué es lo que se va a hacer en este curso:

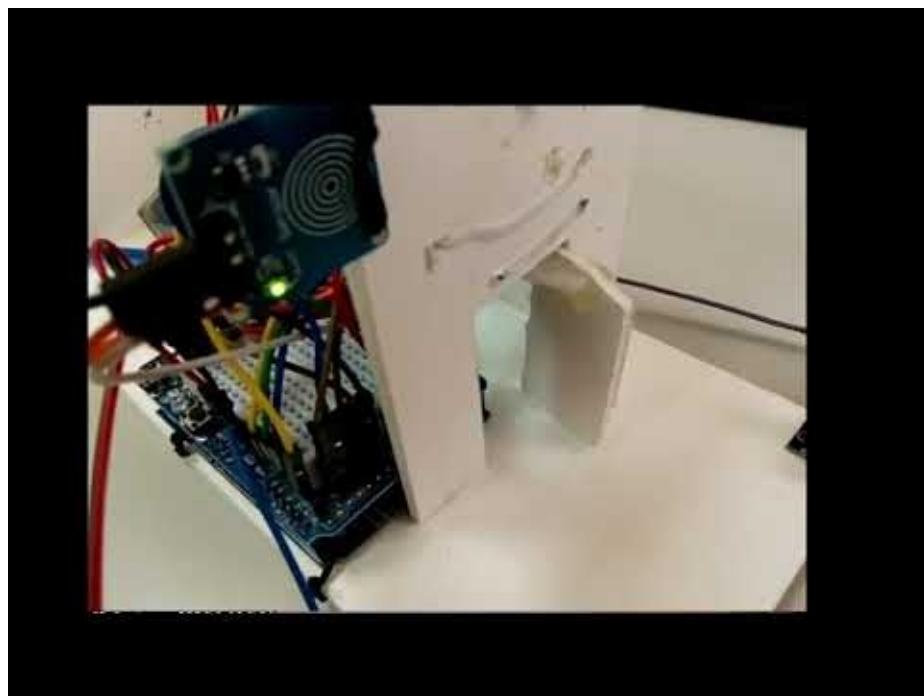


Imagen - YouTube video thumbnail

[Video link](#)

1.1.2 ¿El objetivo es crear la casa?



No, el objetivo es aprender las diferentes posibilidades que tiene el Arduino utilizando módulos de bajo coste. La casa domótica es la excusa para llegar a este objetivo.

Luego puedes utilizar estos mismos materiales para hacer muchas cosas diferentes, por ejemplo:

- Con el sensor de humedad y el servo puedes hacer un [riega macetas](#)
- Con el sensor de ultrasonidos y el buzzer puedes hacer un [piano invisible](#)
- Con el LDR y el láser puedes hacer [otro tipo de alarma con piezas de lego](#)
- Con el sensor de ultrasonidos y el led RGB puedes hacer un [semáforo-distancia](#)
- Con el sensor de ultrasonidos, el servo y el diodo RGB puedes hacer un [tractor entrando en el corral](#)
- Más opciones, o si encuentras alguno, lo puedes colgar tú mismo en el [muro que te proponemos](#).

1.1.3 ¿Tengo que montarlo todo?

No, no.. es flexible: si te asusta tanto cable, puedes ir haciendo los retos por separado, desmontando los anteriores y así queda más limpia la maqueta.

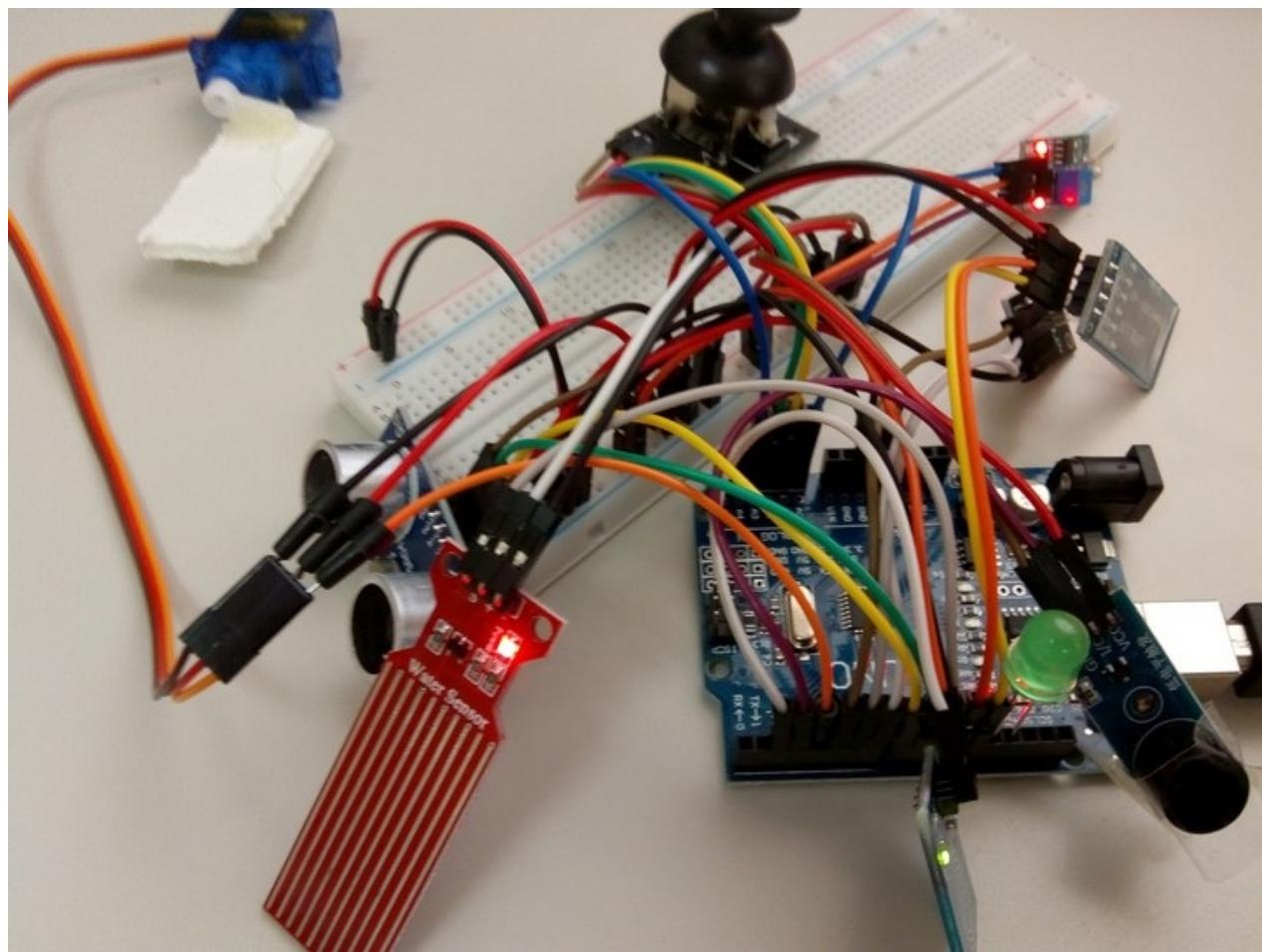
Pero ánimo, no seas un gallina !!



via [GIPHY](#)

1.1.3 ¿Tengo que montar la maqueta?

Tampoco es necesario, de hecho, en el módulo 2 Programación en bloques las fotos y vídeos son sin montar la casa domótica, pero lo puedes hacer.

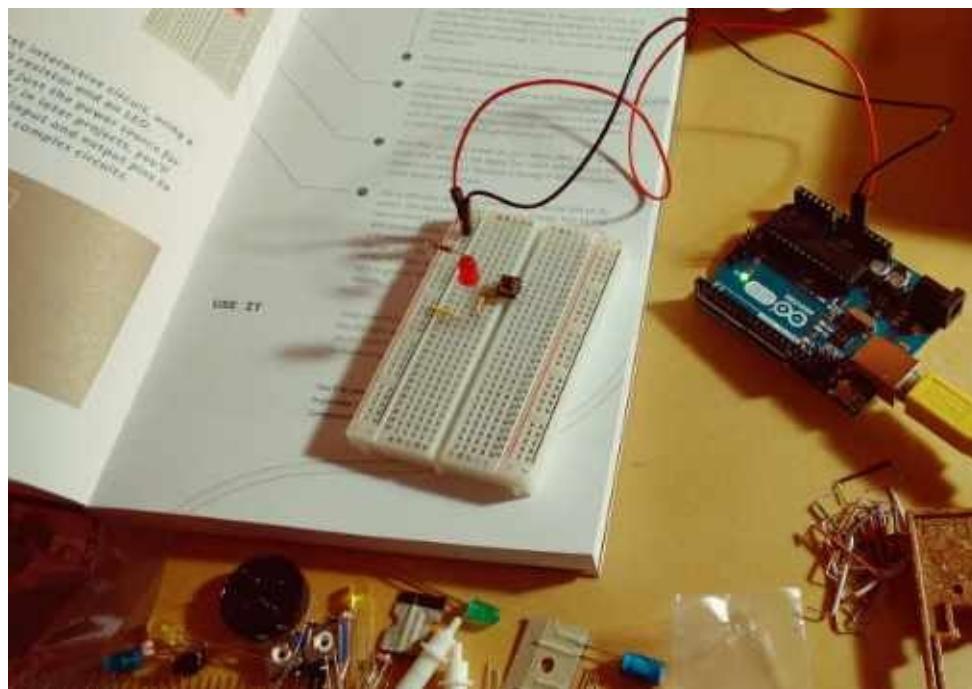


En el módulo 3 con código se explica cómo montar la maqueta, y los vídeos y fotos son con la casa domótica.

1.1.4 ¿Que es eso de una Shield?

Una Shield , dicho pronto, mal pero claro **es una placa electrónica para reducir la electrónica**, echa un vistazo [a esta página](#).

Sin Shield:



Con Shield:

Realmente utilizaremos una pequeña Shield, (no sé si realmente llamarla así, pues no tiene electrónica integrada) y es una pequeña placa Protoboard, pero perfectamente puedes utilizar una placa Protoboard normal sin esta Shield.

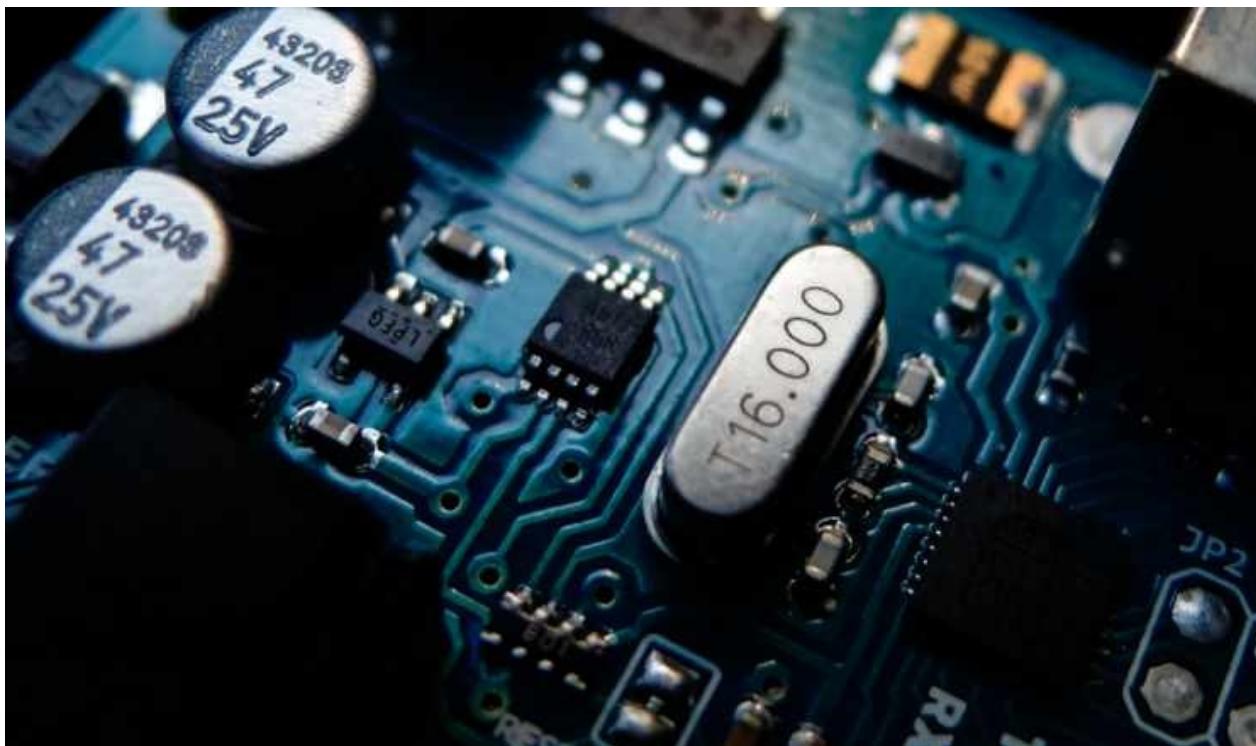
La utilizamos por cuestión de ahorro de espacio.



1.2.- ¿Qué tengo que saber sobre Arduino?

Tienes que ver y entender estas páginas:

- [Información básica sobre Arduino](#)
 - ¿Qué es el Arduino?
 - ¿Qué es un microcontrolador?
 - ¿Entradas y salidas?
 - ¿Dónde se conectan los sensores a las entradas digitales o analógicas?
 - ¿Qué utilidad tienen las entradas digitales?
- [Hardware de Arduino](#)
 - Placa Arduino
 - Entradas y salidas
 - Pines de la placa
- [Alimentación eléctrica de Arduino](#)
 - Mediante el cable USB conectado al ordenador
 - Utilizando una fuente de alimentación externa conectada al jack de Arduino (fuente de voltaje, adaptador de corriente, batería o portapilas)
- [Entradas analógicas](#)
 - Para una señal de entrada desde 0V a 5V el Arduino lo interpreta desde 0 a 1023.
- [¿Qué son las salidas digitales etiquetadas con PWM \(~\)?](#)
 - Aunque no tiene salidas analógicas, podemos "simular" con estas salidas digitales valores medios entre 0V y 5V.



1.3 Kit de préstamo de CATEDU

Nuestra propuesta es utilizar componentes electrónicos que se pueden adquirir en muchas tiendas de electrónica, incluso online, no es necesario ir a los gigantes Amazon o AliExpress, las tiendas electrónicas españolas ofrecen venta online con mejores precios y servicios que los anteriores.

Nuestra propuesta cuesta alrededor de 30€ y sinceramente es muy completo y más barato frente a otros packs de domótica que ofrecen otras empresas. Los kits que se venden son bastante caros, y es mejor comprar los componentes por separado [ver opinión de Luis Llamas](#)

COMPRA-KIT-DOMOTICA-ARDUINO	
<u>Componente</u>	
Arduino UNO R3 compatible CH340	
Cable USB 30cm	
Mini Servo de 180º para RC, sg90	
Hoja1	Hoja2 Hoja3
	> <

Aunque para empezar puedes hacerlo con este kit de iniciación, por unos 20€, y luego ir ampliando:

Kit de iniciación : Hoja 1	
COMPONENTE	PRECIO UN
Arduino UNO	
Cable USB 30cm	
Diodo Láser	
Hoja 1	> <

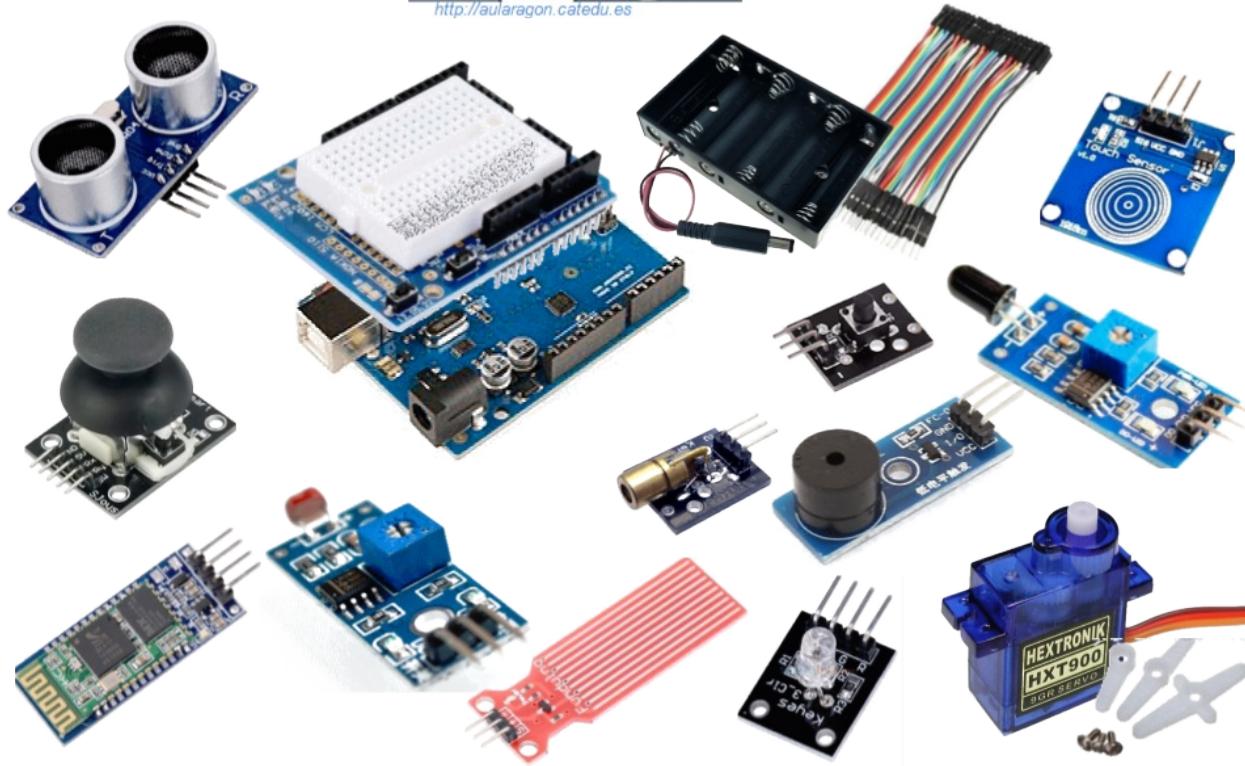
La descripción de cada componente lo tienes aquí:

Pack CATED DOMOTICA

Con ARDUINO



**KIT DOMOTICA CON ARDUINO
PARA EL CURSO DE AULARAGON**
tutorial
www.catedu.es-robotica



Y lo iremos conectando a las diferentes entradas y salidas de Arduino:

COMPONENTE	CONEXIÓN EN ARDUINO
LDR	A0
EJE X JOYSTICK	A1
EJE Y JOYSTICK	A2
SWITCH JOYSTICK	A3
PULSADOR ALARMA	A4
PULSADOR TACTIL	D2
BUZZER ACTIVO	D3
SERVO	D4
RGB AZUL	D5
RGB ROJO	D6
RGB VERDE	D7
SENSOR AGUA	D8
SENSOR LLAMA	D9
LASER	D10
TX DE BLUETOOTH	D11
RX DE BLUETOOTH	D12
LED VERDE	D13

SENSOR ULTRASONIDOS	TRG D12 ECHO D13
---------------------	------------------

Evidentemente cuando usemos el sensor de ultrasonidos prescindiremos del Bluetooth y del LED verde

2.1 Programación mBlock

Es un programa especializado en el manejo de los robots de Makeblock (ver cursos de mBot en Aularagon), estos robots están basados en Arduino por lo tanto este programa también puede programarlos



Se puede descargar gratuitamente en <http://www.mblock.cc/>

Recomendamos en Windows la versión 3

mBlock 3 for PC
The perfect combination of Scratch and Arduino code, designed for programming for kids
Hardware:
[mBot/Ranger/Ultimate2.0/Arduino](#)
V3.4
Latest macOS recommended/Win7 and above/
WinXP/Chrome OS/Linux
[Previous Versions](#) [Release Logs](#)

Online Create

WinXP

Win7 and above

Mac OS

Chrome OS

Linux

2.1.1 NO PROGRAMAMOS DIRECTAMENTE EN EL ARDUINO

mBlock (y los otros S4A, Snap4Arduino... también) **NO PROGRAMAMOS DIRECTAMENTE EN EL ARDUINO**. Lo que hace el ordenador es traducir el lenguaje SCRATCH y en el Arduino hay instalado un Firmware para ir escuchando y ejecutando lo que manda el ordenador.

EN EL CAPITULO 3 PROGRAMAREMOS DIRECTAMENTE EN EL ARDUINO

2.1.2 VENTAJAS

Pues te permite interactuar el Arduino y el ordenador, por ejemplo podemos hacer que cuando el detector de humedad detecte agua, que salga por pantalla un fondo acuático.

2.1.3 DESVENTAJAS

Pues sí, **en primer lugar hay que cargar dentro del Arduino el Firmware exclusivo de mBlock** para que Arduino haga caso a mBlock

En segundo lugar, al tener **nuestro ordenador como intermediario, se come los recursos** y puede no ir o ir lento nuestro programa. (típico de los intermediarios)

Y además, necesitamos el ordenador conectado al Arduino, o sea, trabaja como un esclavo del ordenador.



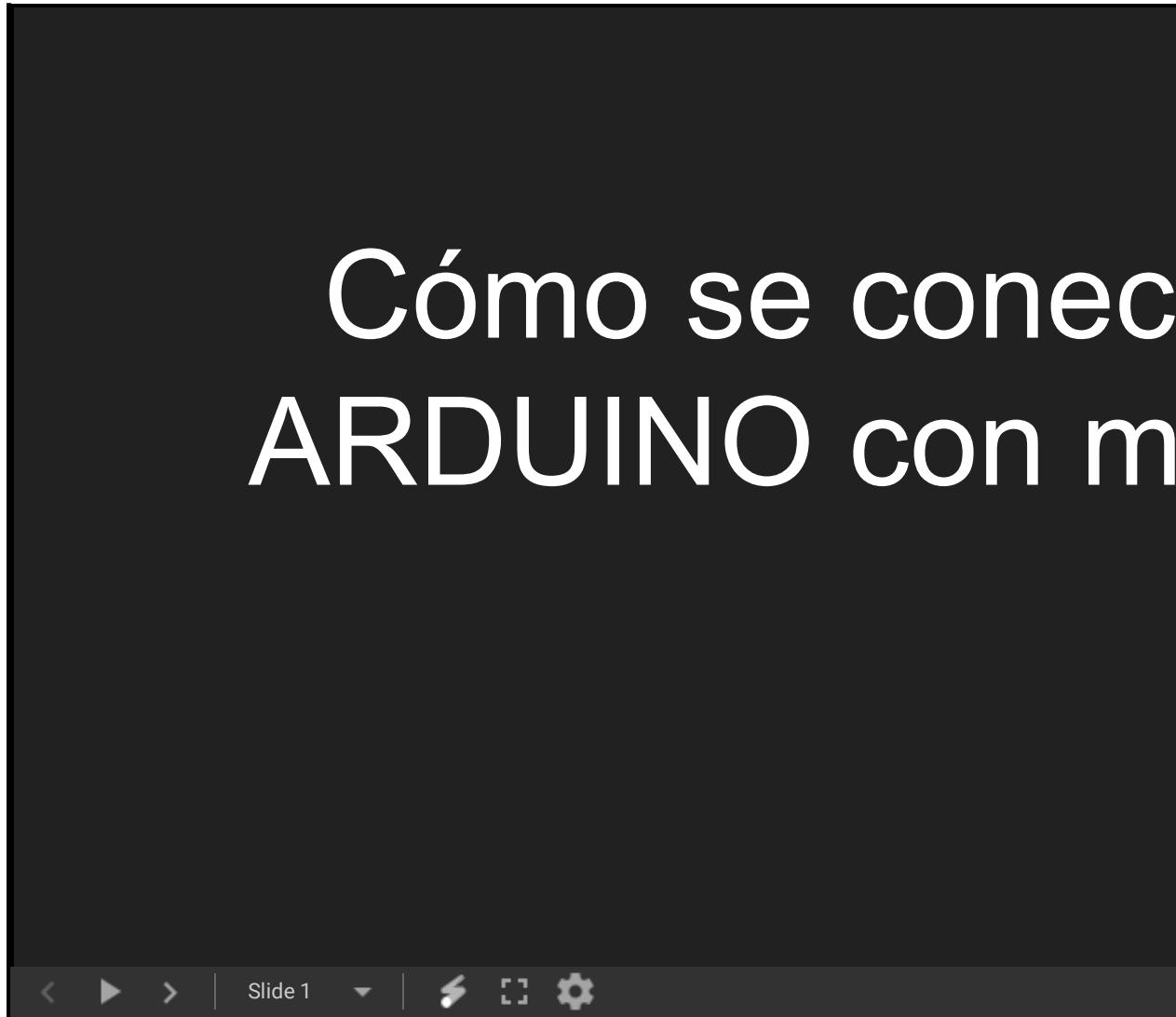
via GIPHY

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:



2.3 INTERRUPTOR CREPUSCULAR

2.3.1 Conocimientos previos

SENSOR DE LUZ LDR

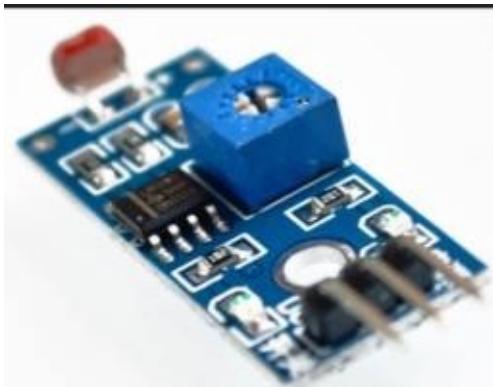
2.3.1.1 SENSOR DE LUZ LDR

El LDR es una resistencia que varía su valor con la luz, cuanto más **OSCURO** más grande es su valor, por lo tanto por la ley de Ohm este módulo nos da una señal analógica que aumenta con la oscuridad. Para saber más del LDR te recomendamos [esta página de Luis Llamas](#).

2.3.1.1.1 Valores umbrales

Por software: Los valores analógicos, [tal y como dice aquí](#), van desde 0 a 1023, luego hay que elegir un valor umbral, nosotros hemos elegido 500.

Por hardware: Puedes regular el potenciómetro que tiene el módulo para que produzca el cambio cuando lo deseas:

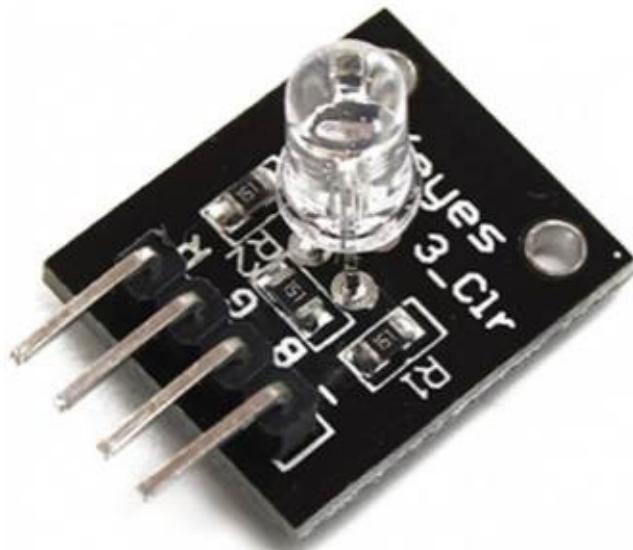


MODULO LED RGB

2.3.1.2 Módulo LED RGB Keyes

Este módulo tiene 4 pines que podemos proporcionar valores analógicos desde 0V a 5V para conseguir diferentes colores:

Pines B = Blue G = Green R = Red (-) = GND



De momento conectaremos el pin G (Green) a una salida digital normal (pin 7), luego sólo se puede encender o apagar.

Más adelante usaremos los colores Blue y Red a salidas PWM

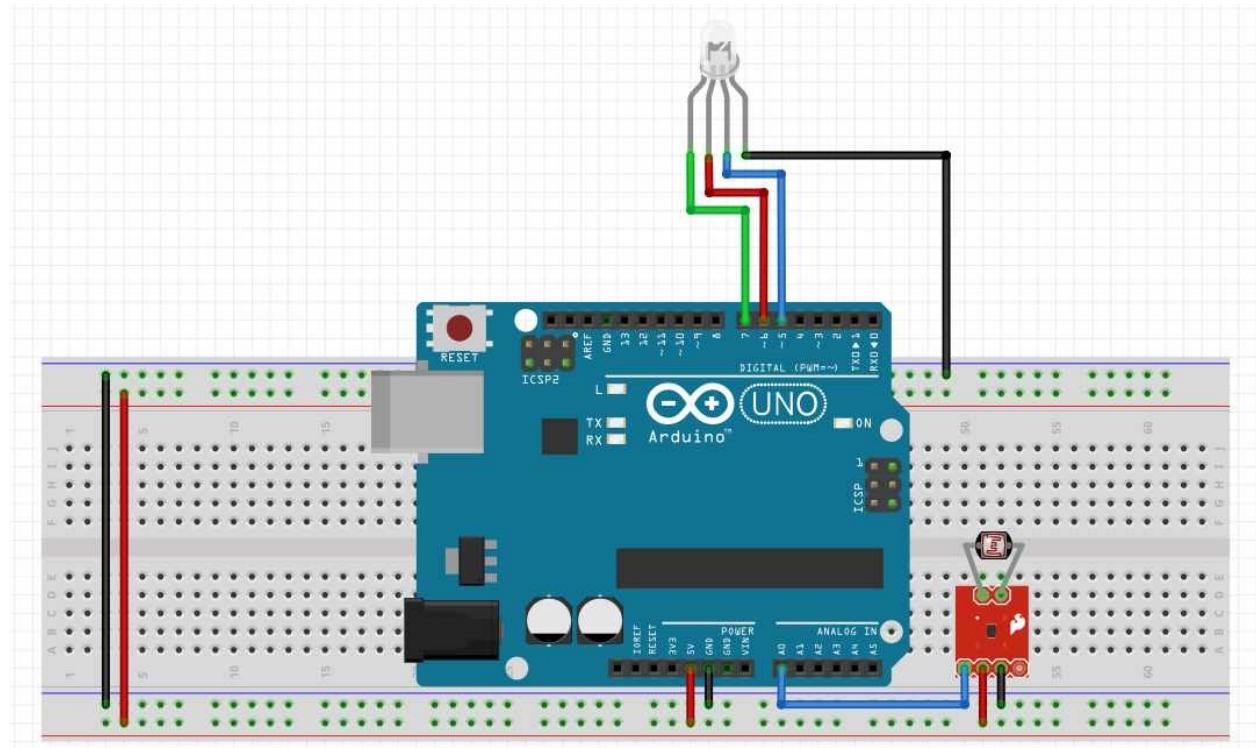
En [esta página](#) puedes encontrar más detalles de este módulo y un ejemplo curioso de utilización sin cables.

2.3.2 Objetivo

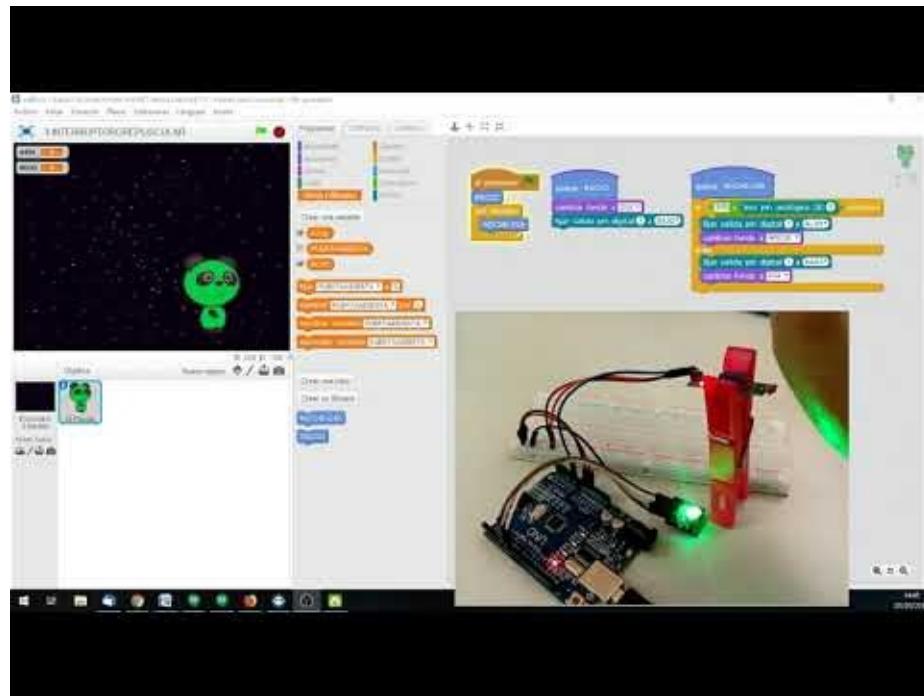
- Cuando el sensor LDR detecte oscuridad
 - El led RGB se enciende verde
 - Cambiamos a fondo tipo "noche"
- En caso contrario
 - el led RGB está apagado
 - el fondo es día

2.3.3 Esquema

- El módulo LDR lo conectamos al pin analógico A0
- El LED RGB lo conectamos
 - Pin 5 digital - Blue
 - Pin 6 digital - Red
 - Pin 7 digital - Green



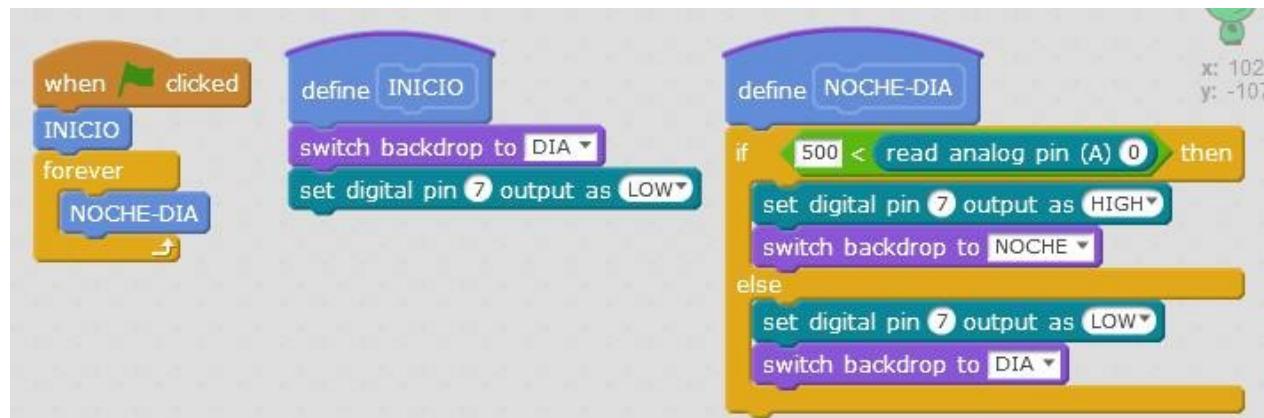
2.3.4 Video



[Video link](#)

2.3.5 Solución

El programa lo puedes descargar [aquí](#)



2.4 Apertura puerta

2.4.1 Conocimientos previos

SERVO

2.4.1.1 SERVOMOTORES

Visita [esta página](#) para que veas dos vídeos y entiendas la diferencia entre:

- Servos convencionales
- Servos de rotación continua



Tranquilo, que el servo que nos interesa es un servo convencional donde simplemente se fija el ángulo con la instrucción en mBlock

fijar ángulo del pin 4 del servo a 90°

Si quieres saber más sobre servos te aconsejamos visitar [esta página](#).

INTERRUPTOR TÁCTIL

2.4.1.2 Interruptor táctil



No tiene mucho misterio: detecta una acumulación de carga y dispara un pulso positivo. Si quieres saber más mira [esta página](#).

BUZZER ACTIVO

2.4.1.3 Buzzer activo



La diferencia con el pasivo es que no es necesario enviarle pulsos para que emita una frecuencia, sólo tenemos que dar la orden y él reproduce un tono.

Ojo que funciona con lógica negativa es decir:

- si queremos que suene tenemos que enviar un LOW.
- si queremos que no suene tenemos que enviar un HIGH

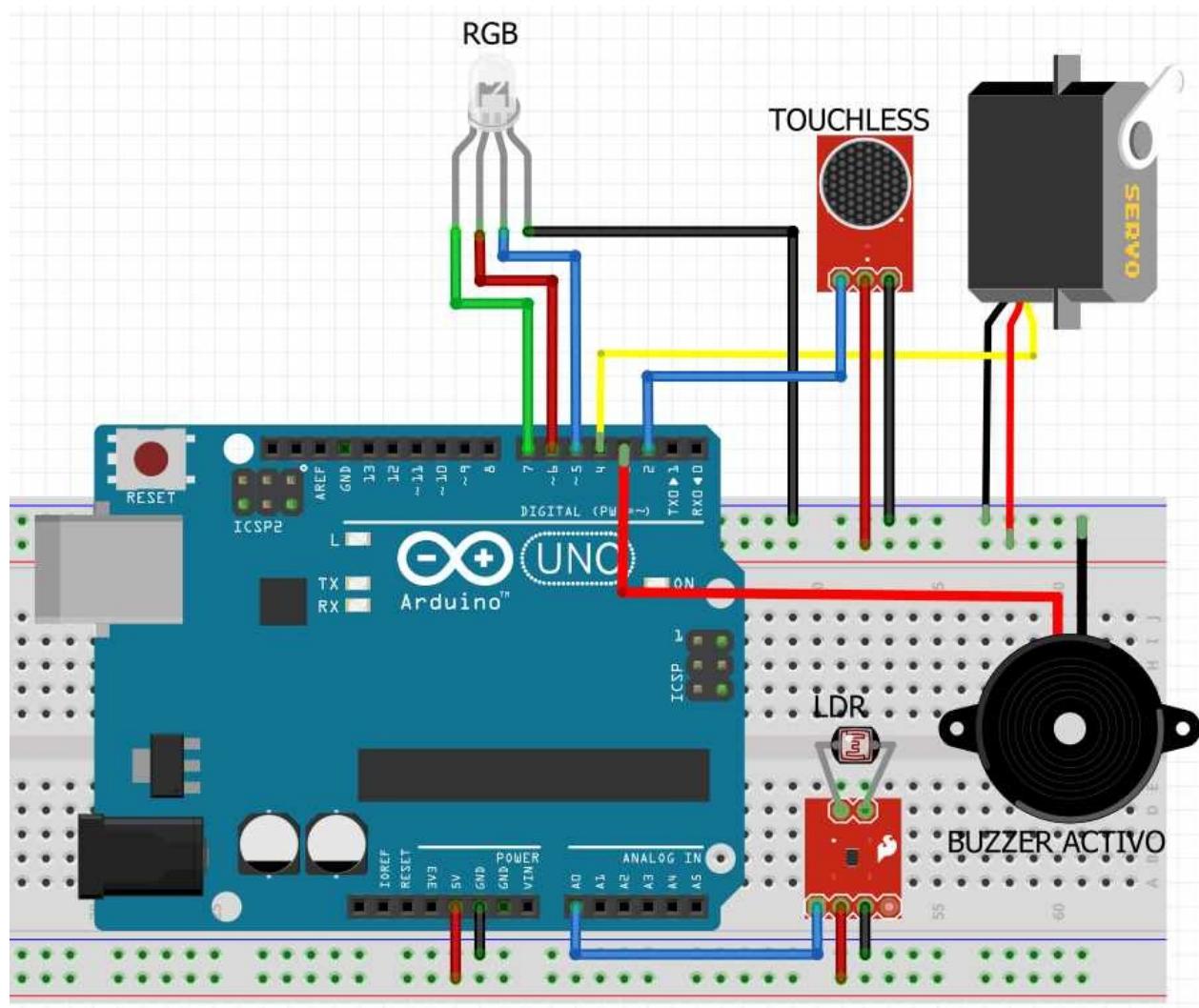
Si quieres saber más de este componente visita [esta página](#)

2.4.2 Objetivo

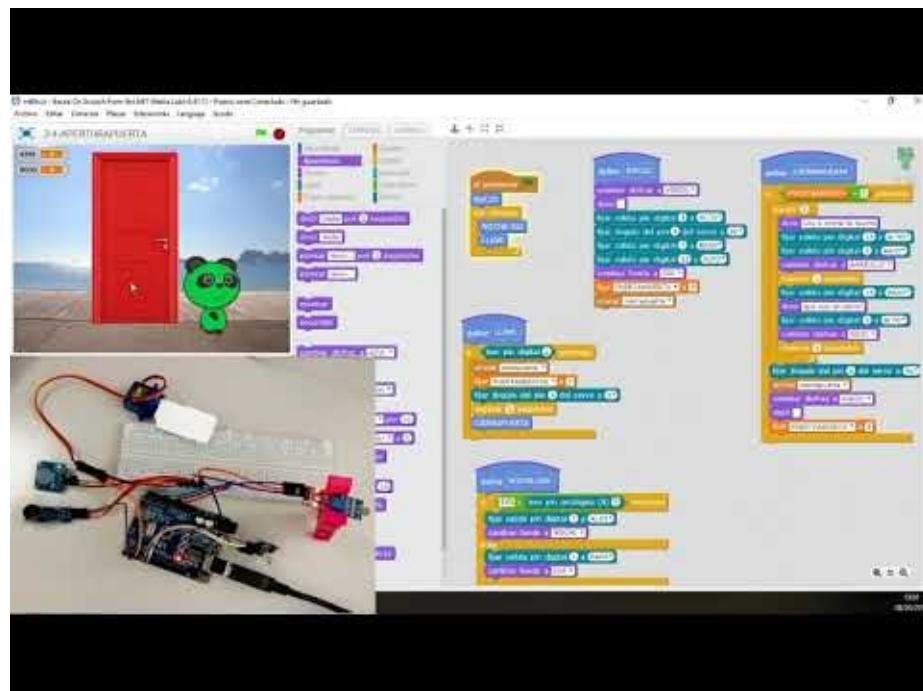
- Cuando se pulse el interruptor táctil (sería como una llave táctil)
 - Se abre la puerta
- Al cabo de 5 segundos, tiempo suficiente para entrar
 - Se avisa que la puerta se va a cerrar con 3 pulsos buzzer y cambiando el color el sprite del Panda de mBlock
 - Se cierra la puerta

2.4.3 Esquema

- pin 2 digital: El interruptor táctil touchless.
- pin 3 digital: El buzzer activo.
- pin 4 digital: Servo de la puerta.



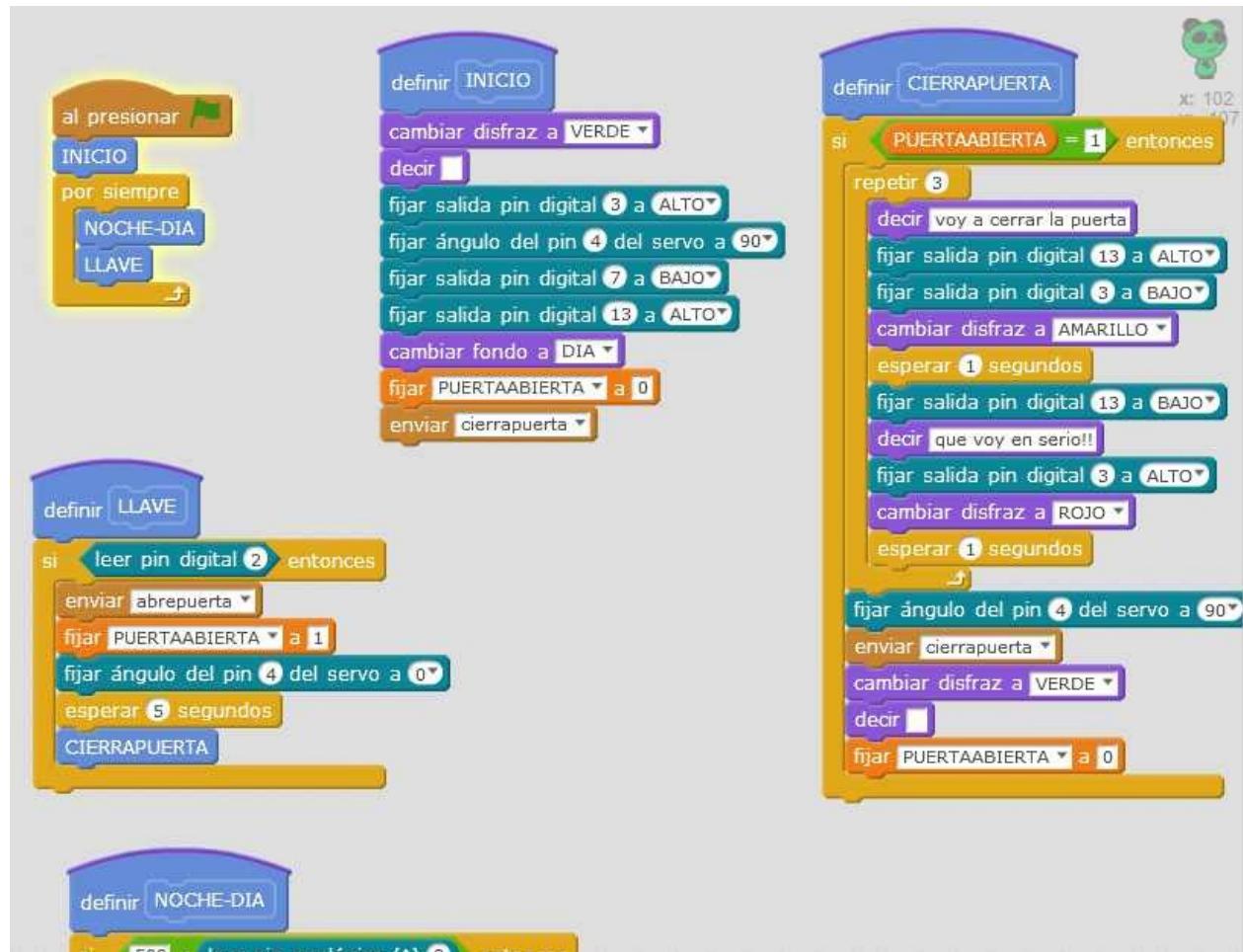
2.4.4 Video



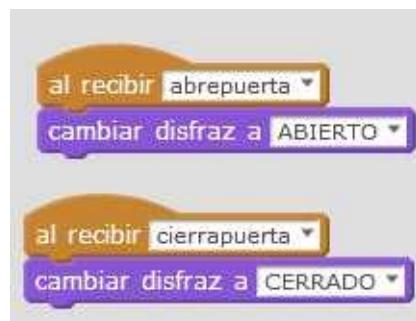
Video link

2.4.5 Solución

El programa lo puedes descargar [aquí](#)



El objeto **puerta** tiene este sencillo programa:



2.5 ALARMA AGUA

2.5.1 Conocimientos previos

SENSOR DE AGUA LDR

2.5.1.1 SENSOR DE AGUA



Es un sensor muy sencillo: Al mojar los conductores, aumenta la conductividad y traduciéndolo en una salida HIGH en su pin de salida S. Si quieres saber más de este sensor visita [esta página](#).

Con sólo tener los dedos húmedos ya lo detecta.

Hay que tener en cuenta que sólo queremos detectar AGUA o NO AGUA, o sea digital. Hay otros sensores de humedad que están indicados para medir la humedad del suelo, (tienen forma de dos pinchos para clavarlos) en estos casos que queremos medir la cantidad de humedad la señal es más precisa y analógica. [Ver](#)

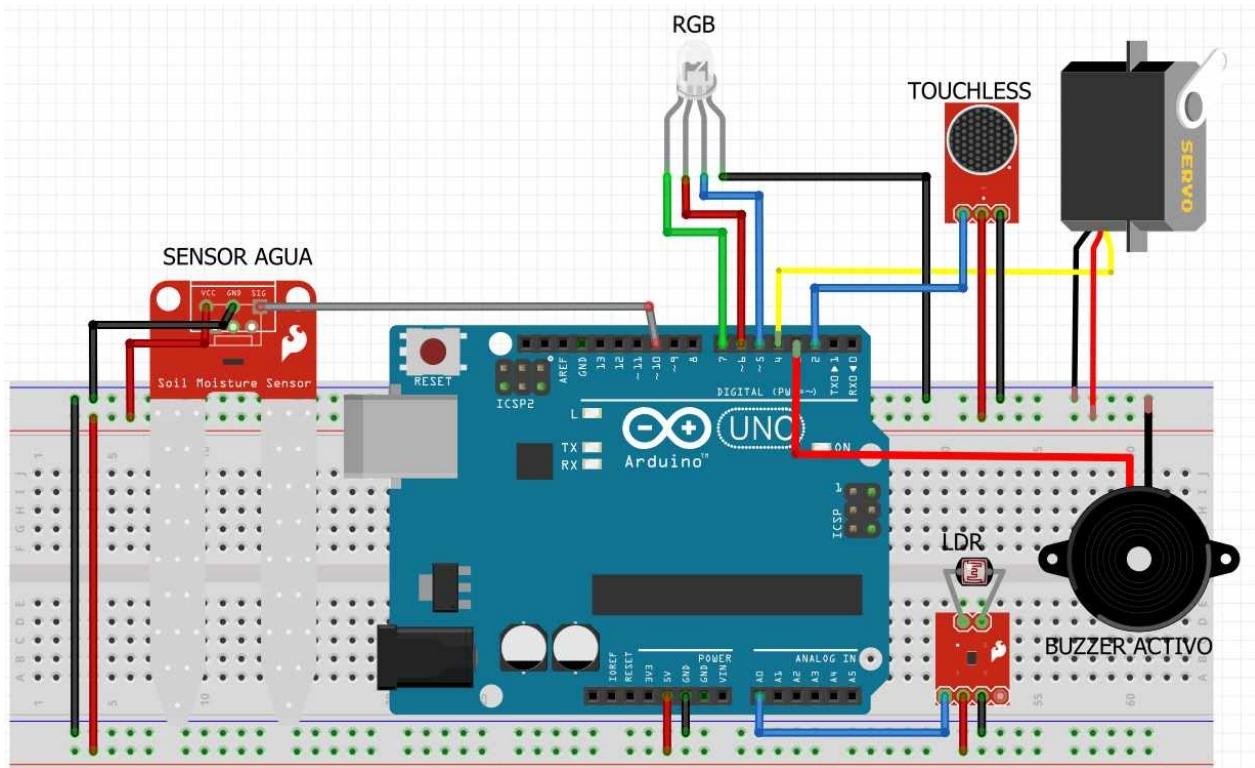
2.5.2 Objetivo

- Si detecto agua :
 - Aviso de agua en el fondo y en el objeto Panda
 - Cierre la puerta pues entra el agua de la lluvia con el mismo procedimiento que en apartado anterior.

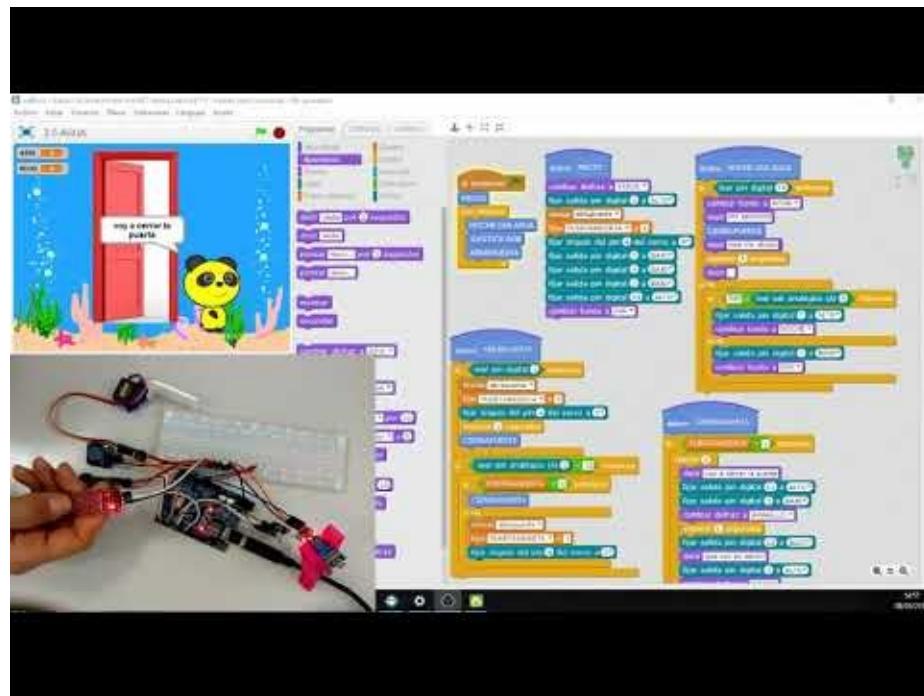
2.5.3 Esquema

- Pin 10 : Sensor de agua

El esquema el sensor de agua es uno de humedad, pero había otro en Fritzing :(



2.5.4 Vídeo

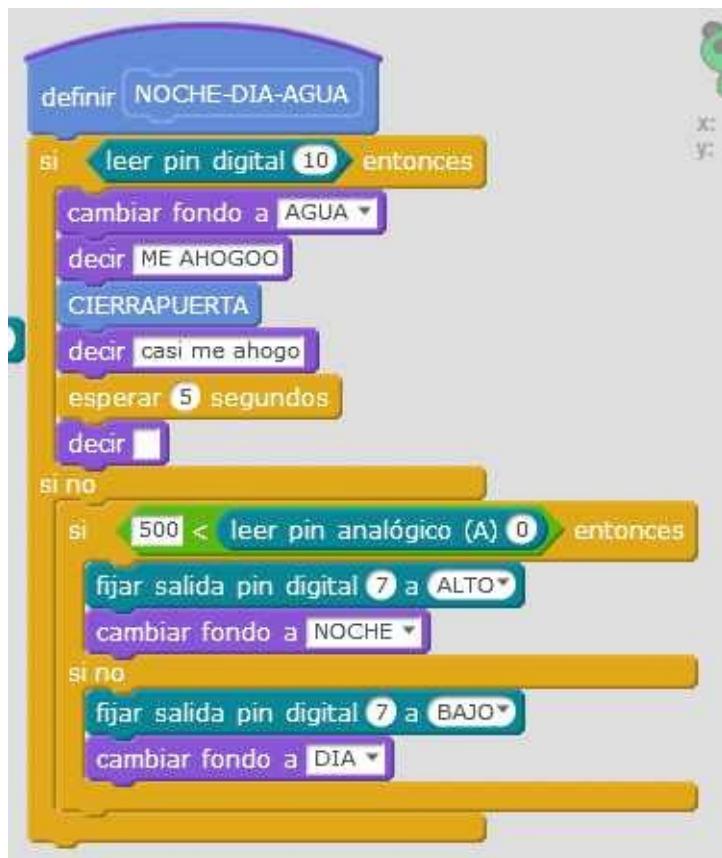


[Video link](#)

2.5.5 Solución

El programa lo puedes descargar [aquí](#)

Simplemente se ha añadido al bloque del interruptor crepuscular pues es el bloque que cambia los fondos y el aspecto del avatar Panda



2.6 JOYSTICK

2.6.1 Conocimientos previos

JOYSTICK

2.6.1.1 JOYSTICK

Yo no sé qué tiene este componente que llama la atención a los chavales



Así que vamos a aprovecharlo: Tiene 2 potenciómetros según los ejes X e Y y un interruptor central. Las dos primeras salidas son analógicas, el microruptor lo trataremos también analógico, pues nos faltarán entradas y salidas digitales cuando lleguemos al final en [3.8 bluetooth](#).

Si quieres saber más de este componente visita [esta página](#).

SALIDAS PWM

2.6.1.2 SALIDAS PWM

Como Arduino no tiene valores de salida analógicos, utilizaremos los pines PWM (~) para simular tensiones variables de salida.

De esta manera la luz Roja y Azul serán variables pudiendo crear el ambiente que queramos.

¿no sabes lo que es PWM (~)? eso es que no te has leído [esto](#).



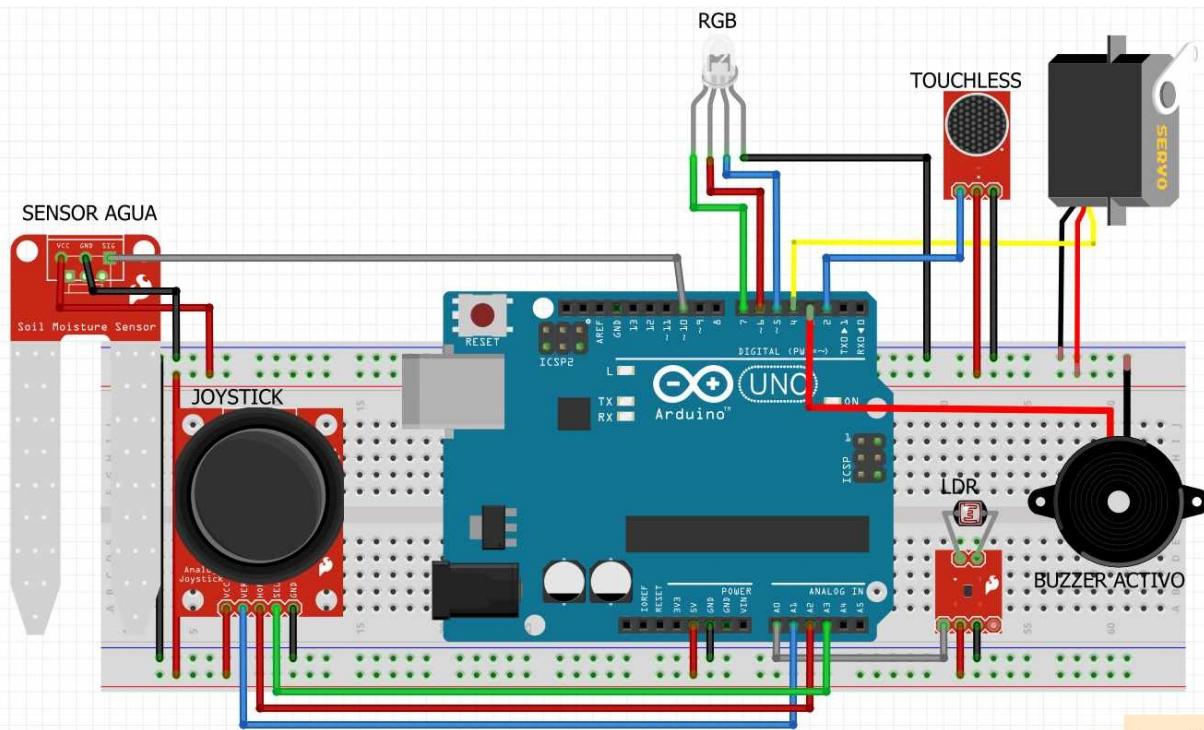
2.6.2 Objetivo

- Si muevo en el eje X el Joy stick la luz RGB se enciende AZUL proporcionalmente a la inclinación del mando Joy stick.
- Si muevo en el eje Y el Joy stick la luz RGB se enciende ROJO proporcionalmente a la inclinación del mando Joy stick.
- Si pulso el Switch central
 - Si la puerta estaba cerrada la abre
 - Si la puerta estaba abierta, la cierra, con los avisos correspondientes.

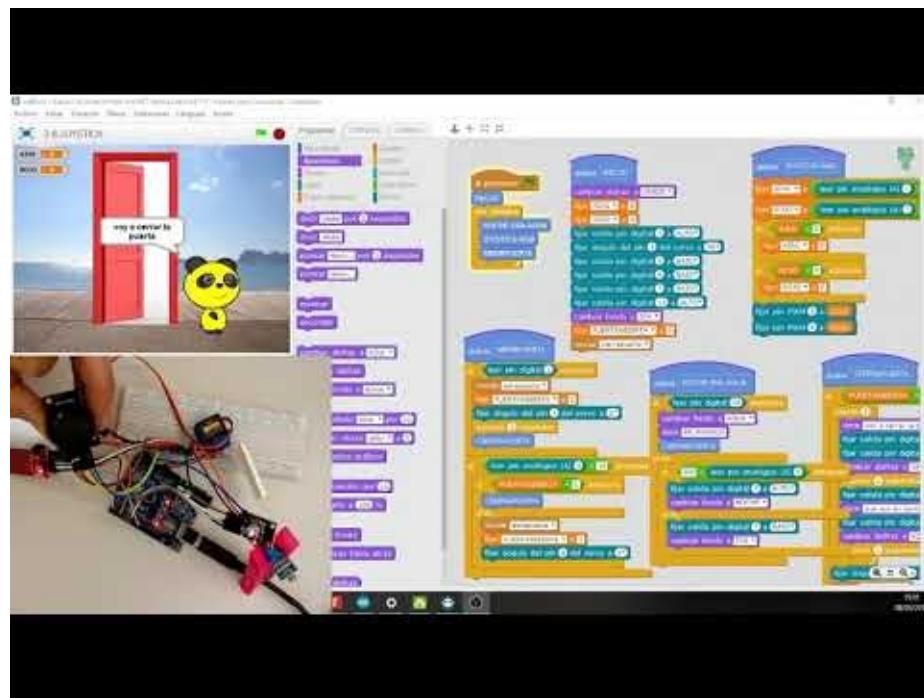
2.6.3 Esquema

Al ser un componente analógico lo conectaremos en las entradas analógicas:

- La salida eje X que gradúa la luz azul en A1
- La salida eje Y que gradúa la luz roja en A2
- La salida Switch central en A3



2.6.4 Vídeo



[Video link](#)

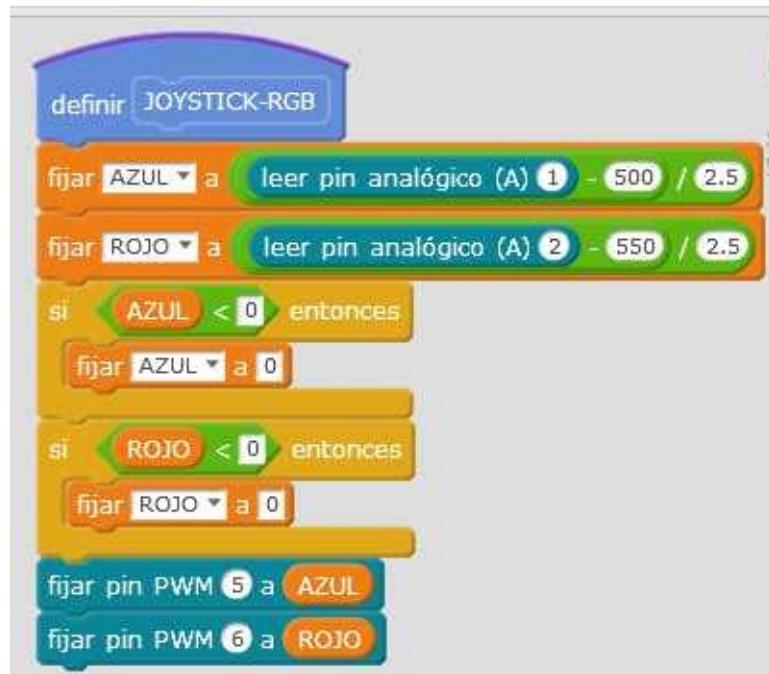
2.6.5 Solución

El programa lo puedes descargar [aquí](#)

Se ha creado un nuevo bloque JOYSTICK-RGB que lee los pines de entrada A2 y A3.

Son valores desde 0 hasta +1024 pero la posición central del Joystick está en 500, luego tenemos que contar desde 500 hasta 1024, luego hay que convertirlos a 0-255

Esos valores ya normalizados transmitirlos a los pines digitales donde está conectado el pin rojo y azul del led RGB por PWM



También se ha añadido a LLAVE un condicional, según el valor A3 que es el Switch central del Joystick que decida si abrir o no la puerta según lo almacenado en una variable PUERTAABIERTA



2.7 ALARMA

2.7.1 Conocimientos previos

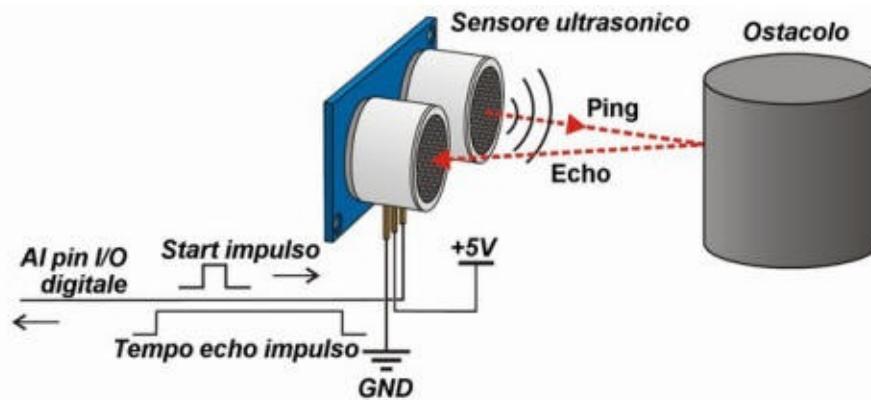
Sensor distancia por ultrasonidos

2.7.1.1 SENSOR DISTANCIA POR ULTRASONIDOS



Este sensor mide las distancias utilizando el eco:

- Un ojo marcado con la T es un altavoz: Emite un sonido ultrasónico fuera de lo audible
- El otro ojo marcado con una R es un micrófono que detecta el pulso emitido por T.



Por software hay que calcular la distancia utilizando la fórmula $v=e/t$ donde v es la velocidad del sonido.

Si quieres saber más de este sensor mira esta página de [Luis Llamas](#).

Por que elegimos este sensor

2.7.1.2 POR QUÉ ELEGIMOS ESTE SENSOR DISTANCIA POR ULTRASONIDOS

Utilización en mBlock

Este sensor es muy preciso y direccional, su código en mBlock es muy sencillo:

```
read ultrasonic sensor trig pin 13 echo pin 12
```

Utilización en código.

En código este sensor implica mucho más código, [lo puedes ver aquí](#), por eso en el apartado 3.7 LASER utilizaremos otra forma de hacer una alarma, con un láser y además un sensor de llama en [3.6 FUEGO](#).

Alternativas

Podriamos usar uno más profesional, por ejemplo [DETECTOR DE MOVIMIENTO CON ARDUINO Y SENSOR PIR](#), pero su ángulo de sensibilidad es tan abierto que todo el rato se dispara. (un rollo si se utiliza en clase).

2.7.2 Objetivo

- Si se pulsa el botón de activación
 - Si la alarma no está activada
 - **Activa** la alarma, es decir *está vigilando*.
 - Si la alarma está activada
 - **Desactiva** la alarma, *deja de vigilar*.
 - Si la alarma está disparada
 - Anula el disparo y desactiva la alarma
- Si la alarma está activada:
 - Está encendido el led verde para indicar que *está vigilando*.
 - Si detecta un *intruso* a menos de 10 cm
 - Se **dispara** la alarma, es decir se enciende la luz roja y el buzzer de forma intermitente, no se apaga hasta que se pulsa el interruptor.

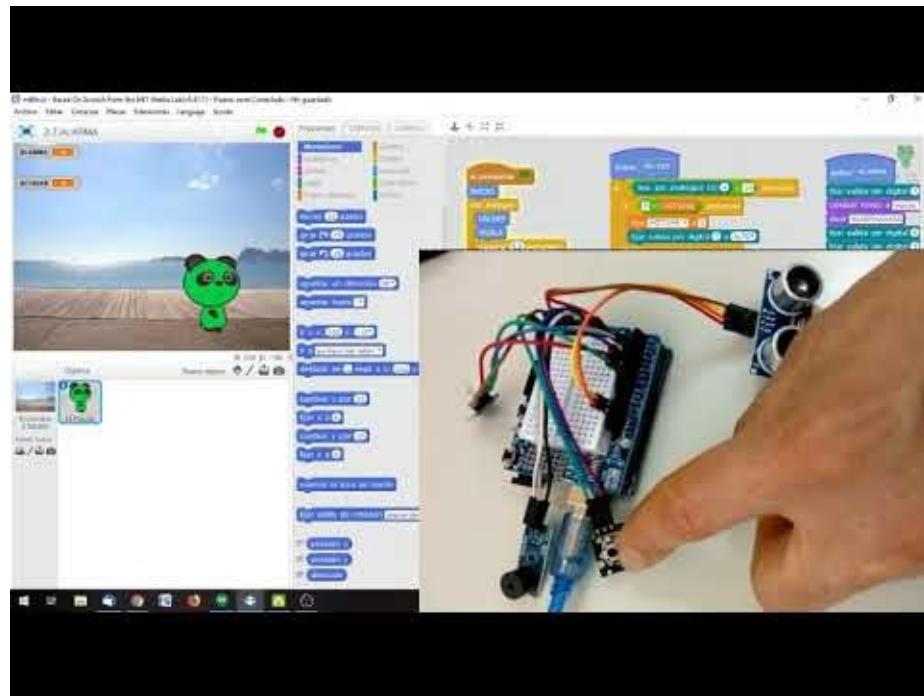
OJO

Haremos un proyecto **totalmente nuevo**, debido a que mBlock no aguanta con todo, incluso este programa suele *colgarse* luego de vez en cuando hay que dar a *Actualizar Firmware* Ya lo comentamos en desventajas en el capítulo [2.1 Programando con mBlock](#)

2.7.3 Conexiones

- Entradas y salidas digitales
 - D3 Buzzer
 - D5 Blue de led RGB
 - D6 Red de led RGB
 - D7 Green de led RGB
 - D12 Echo del sensor de ultrasonidos
 - D13 Trg del sensor de ultrasonidos
- Entradas y salidas analógicas
 - A4 Pulsador

2.7.4 Video



[Video link](#)

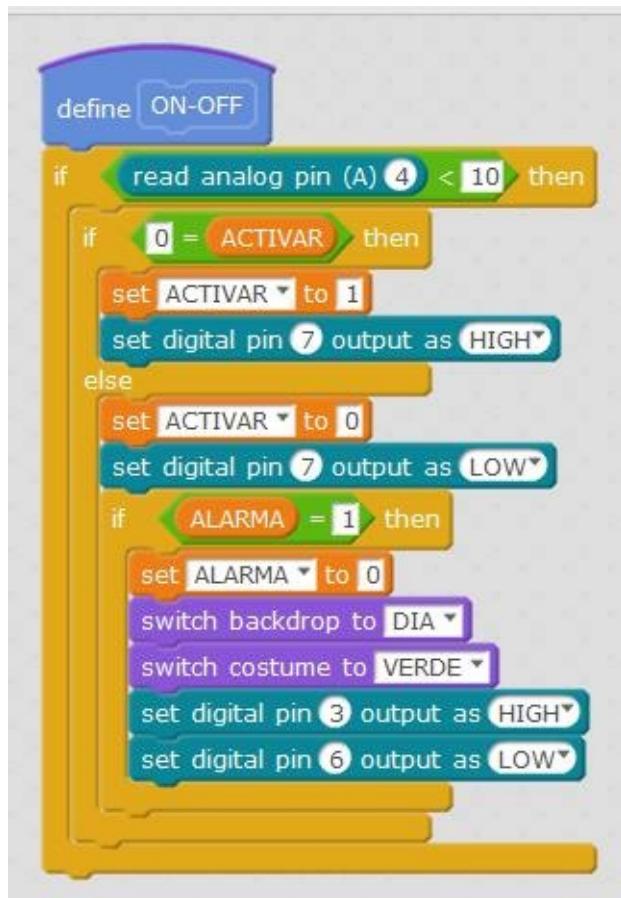
2.7.5 Solución

El programa lo puedes descargar [aquí](#)

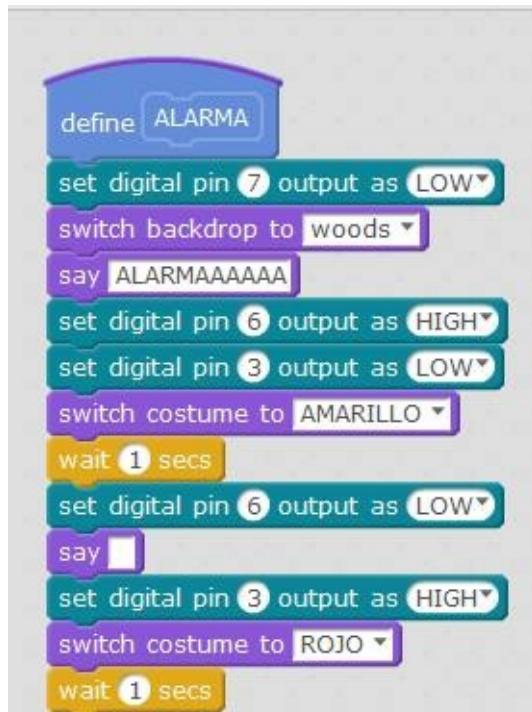
Haremos programación con bloques, este es el programa principal:



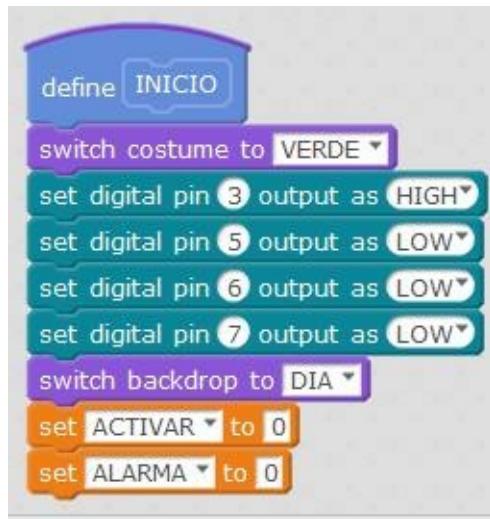
La activación o no de la alarma y la anulación del disparo:



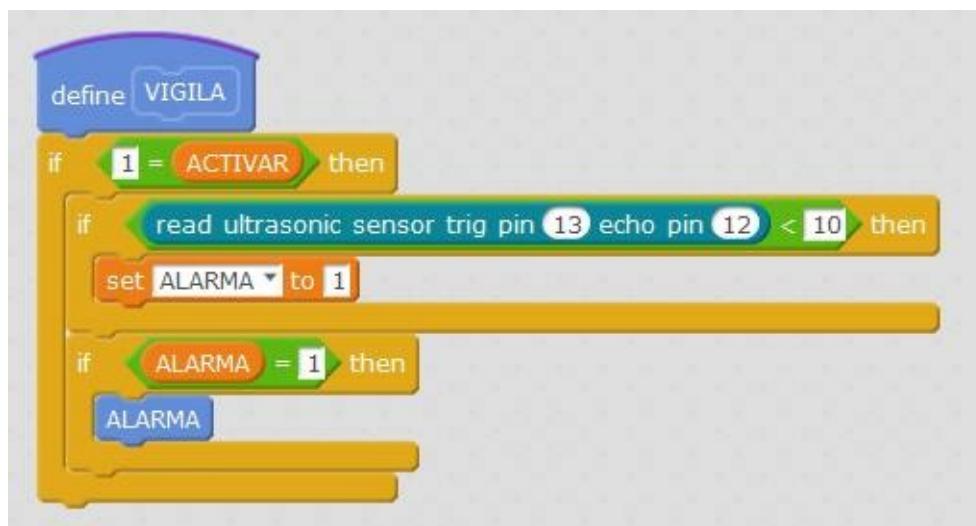
El sonido de la alarma y la intermitencia del rojo



Al inicio hay que resetear todo:



El bloque de vigilar



Entorno de programación

Entorno de programación

Necesitarás el **entorno de desarrollo Arduino IDE** (IDE, Integrated development environment) (aquí <https://www.arduino.cc/en/Main/Software> para descárgatelo)

OJO, existe la versión web del editor <https://create.arduino.cc/editor> para trabajar online, puedes probarlo, pero seguimos recomendando la versión de escritorio por ser más rápida, no obstante es una buena solución si trabajas en varios equipos y quieras que tus proyectos estén disponibles en cualquier equipo.

En Linux puede salir este mensaje "*can't open device "/dev/ttyUSB0": Permission denied*" donde 0 puede ser otro número, la solución [aquí](#)

Está constituido por un **editor de texto** para escribir el código, un **área de mensajes**, una barra de herramientas con botones para las funciones comunes, y una serie de menús.

Arduino utiliza para escribir el código fuente o programa de aplicación lo que denomina "sketch" (programa). Estos programas son escritos en el editor de texto. Existe la posibilidad de cortar/pegar y buscar/reemplazar texto.



Permite la conexión, por USB, con el hardware de Arduino para cargar los programas y comunicarse con ellos.



Y permite varias placas, tenemos que elegir la nuestra, en el KIT de CATEDU es Arduino UNO pero si tienes otro modelo este curso seguro que puede ser válido:



En el área de mensajes se muestra información mientras se cargan los programas y también muestra errores.

Lo importante es cuando pinchamos en la flecha de subir nuestro programa, no salga ningún error, sino simplemente "Subido".



¿Cómo se programa Arduino?

Las partes principales de un programa hecho en Arduino son: Bloque de inclusión de módulos y declaración de variables, bloque de configuración **void setup()** donde se indica el modo de funcionamiento de los pines (entrada y salida), comunicación serie, etc... y bloque de ejecución continua **void loop()**, en este bloque se incluyen las acciones que queremos que realice el programa. Se ejecutará línea a línea de forma secuencial y continua. Cuando llegue a la última instrucción incluida en la función **loop()** volverá a ejecutar la primera y continuará en un bucle infinito.



¿Arduino tiene que estar continuamente conectada a un ordenador?

Sólo es necesario que esté conectado al ordenador mediante el USB para cargar los programas o para visualizar en tiempo de ejecución datos del programa mediante la consola serie. El ordenador proporciona la energía eléctrica suficiente para que funcionen los programas, pero una vez cargado el programa en la memoria del microcontrolador de Arduino se puede desconectar del USB y alimentar a la tarjeta mediante una fuente externa mediante el jack de alimentación con un margen de (5 a 20 Voltios). El programa cargado en Arduino queda grabado permanentemente aunque cese el suministro eléctrico.

¿Qué voy a aprender con este manual?

Aprenderás a realizar pequeños proyectos y prácticas cuya base de control es la tarjeta Arduino y en algunos casos la shield “Edubásica”. EDUBÁSICA es una tarjeta que se coloca sobre Arduino y lleva integrados muchos de los componentes básicos para realizar las prácticas de electrónica, y ciertos proyectos tecnológicos de una manera muy sencilla, aunque todas las prácticas se pueden implementar sin el uso de esta tarjeta. La idea del manual es integrar los contenidos del currículo en 4º de Enseñanza Secundaria Obligatoria (alumnos de 15-16 años) con diferentes proyectos para que el alumno consiga un aprendizaje significativo de la Tecnología.

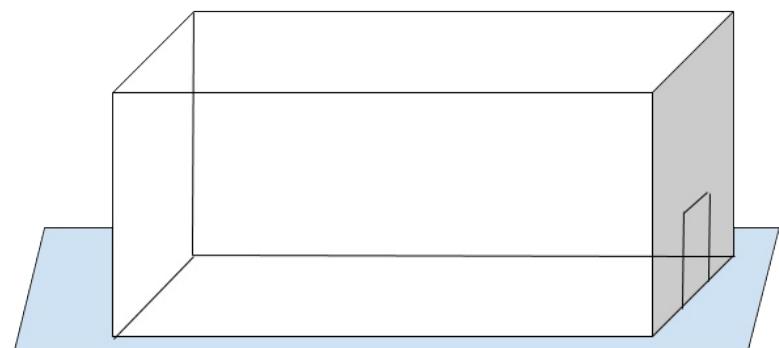
Para una mayor información y manejo de la instalación del entorno de programación, lenguaje de programación y librerías se encuentra en la página web de la comunidad Arduino:

www.arduino.cc (portal en inglés, más actualizada).

www.arduino.es (portal en español).

3.2 MAQUETA

Compramos una hoja de cartón piedra de formato A3 (el doble de un folio) y lo cortamos por la mitad a la larga, y en dos cuadrados la otra mitad. En uno de esos cuadrados recortamos la puerta, en total sólo 3 paredes y el suelo es la otra parte de arriba:



La mejor forma de pegarlo es con pistola, pero también clavando agujas o las dos a vez.

Para ir poniendo los componentes lo más fácil y recicitable es perforar, con una simple punta de bolígrafo:



Y atarlo con alambre plastificado típicos de las bolsas de pan de molde:



El soporte del servomotor con la puerta es difícil, una opción es pegarlo con cola.



Tanto el cartón piedra como el alambre se puede conseguir en cualquier bazar a precios muy económicos.

3.3 APERTURA DE PUERTA

3.3.1 Objetivo

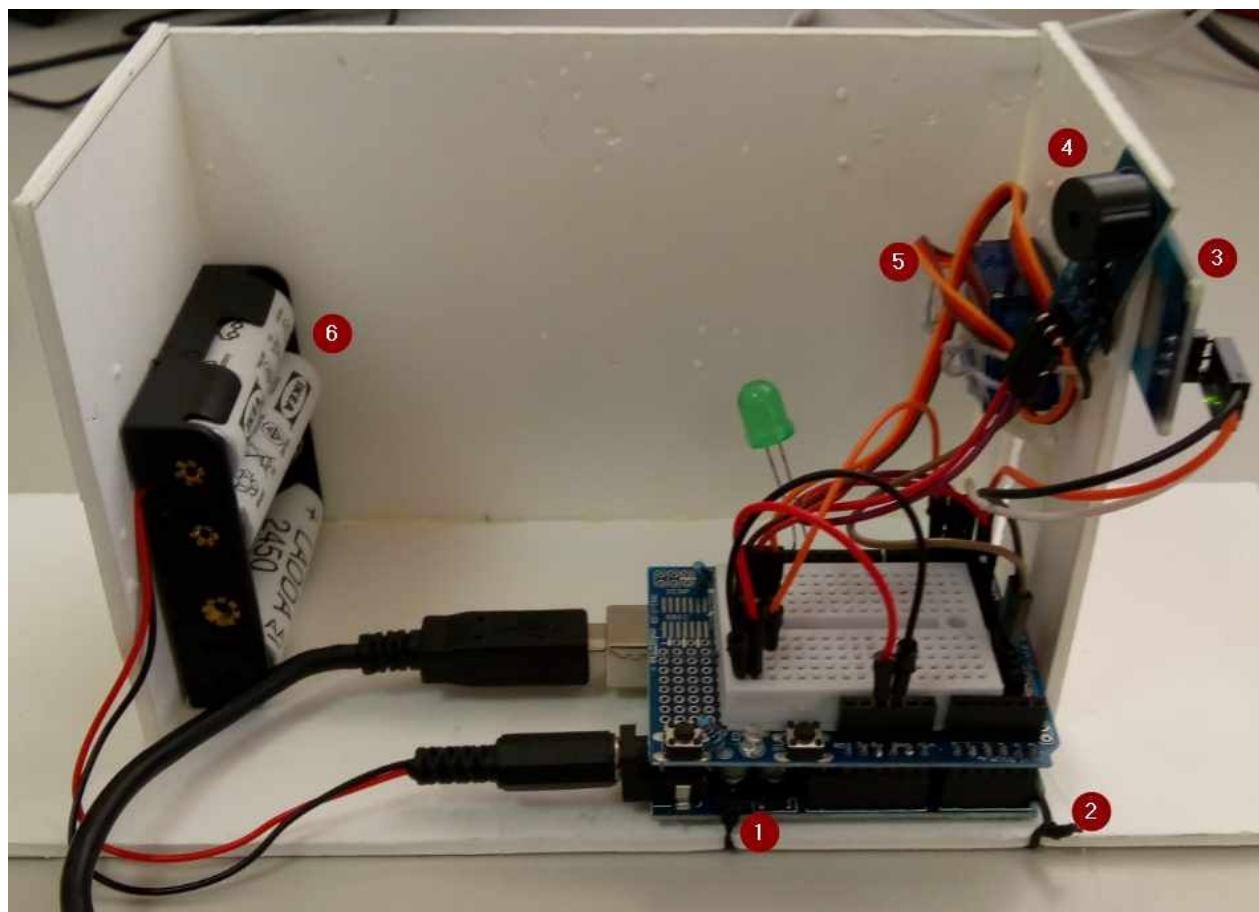
Esta vez vamos a centrarnos con la apertura de la puerta, dejaremos el LDR para la alarma

- Cuando se pulse el interruptor táctil (sería como una llave táctil)
 - Se abre la puerta
- Al cabo de 5 segundos, tiempo suficiente para entrar
 - Se avisa que la puerta se va a cerrar con 3 pulsos buzzer
 - Se cierra la puerta
- Por la pantalla del ordenador saldrán mensajes de apertura y cierre

3.3.2 Maqueta

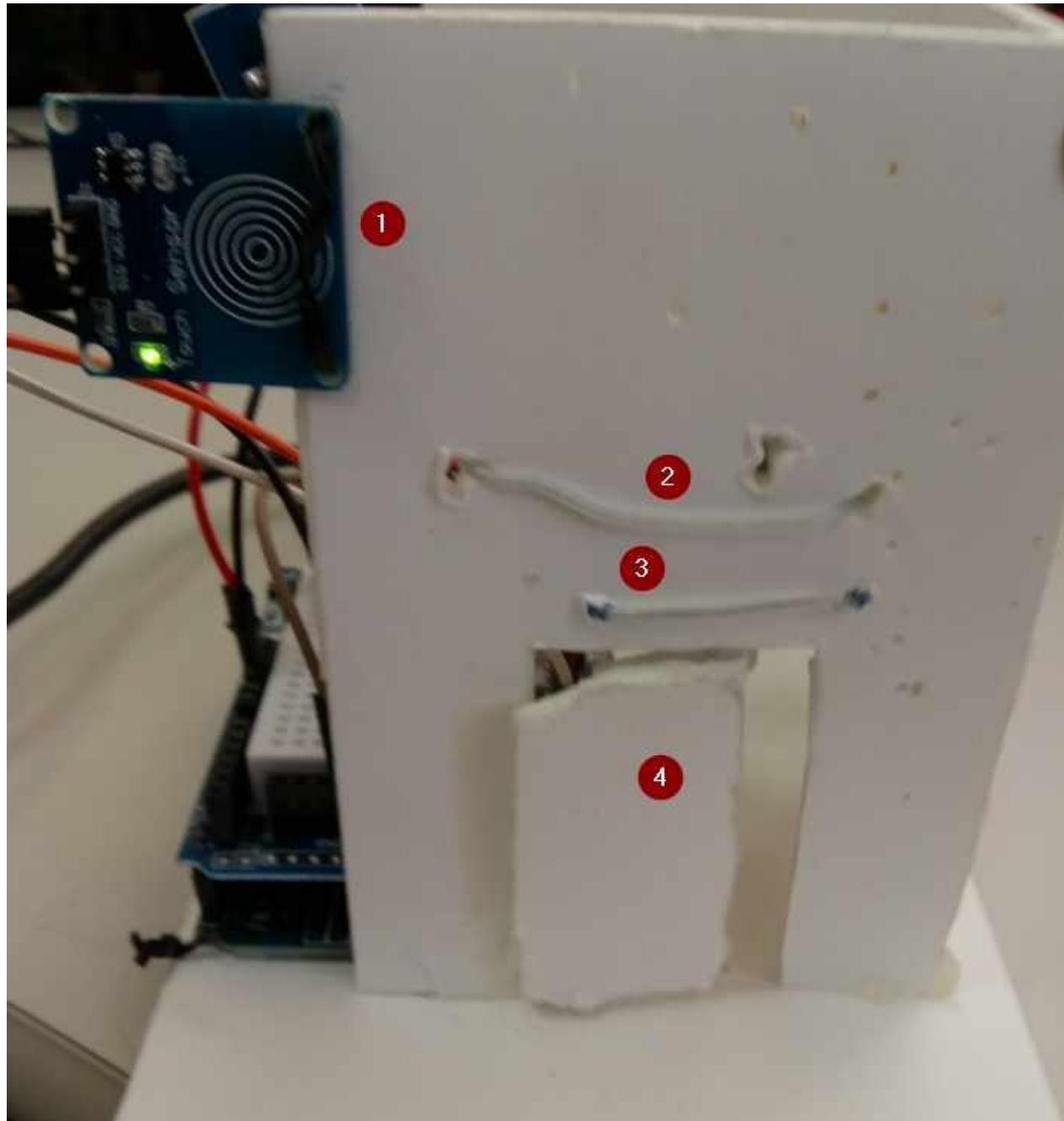
Esta vez en la maqueta ponemos los siguientes elementos

1. Fijación placa Arduino
2. Fijación placa Arduino
3. Interruptor táctil
4. Buzzer
5. Servo
6. Portapilas



Detalle por delante

1. Interruptor táctil
2. Fijación servo
3. Fijación servo
4. Puerta *si, ya sé, un poco chapuza ¿y qué pasa?*

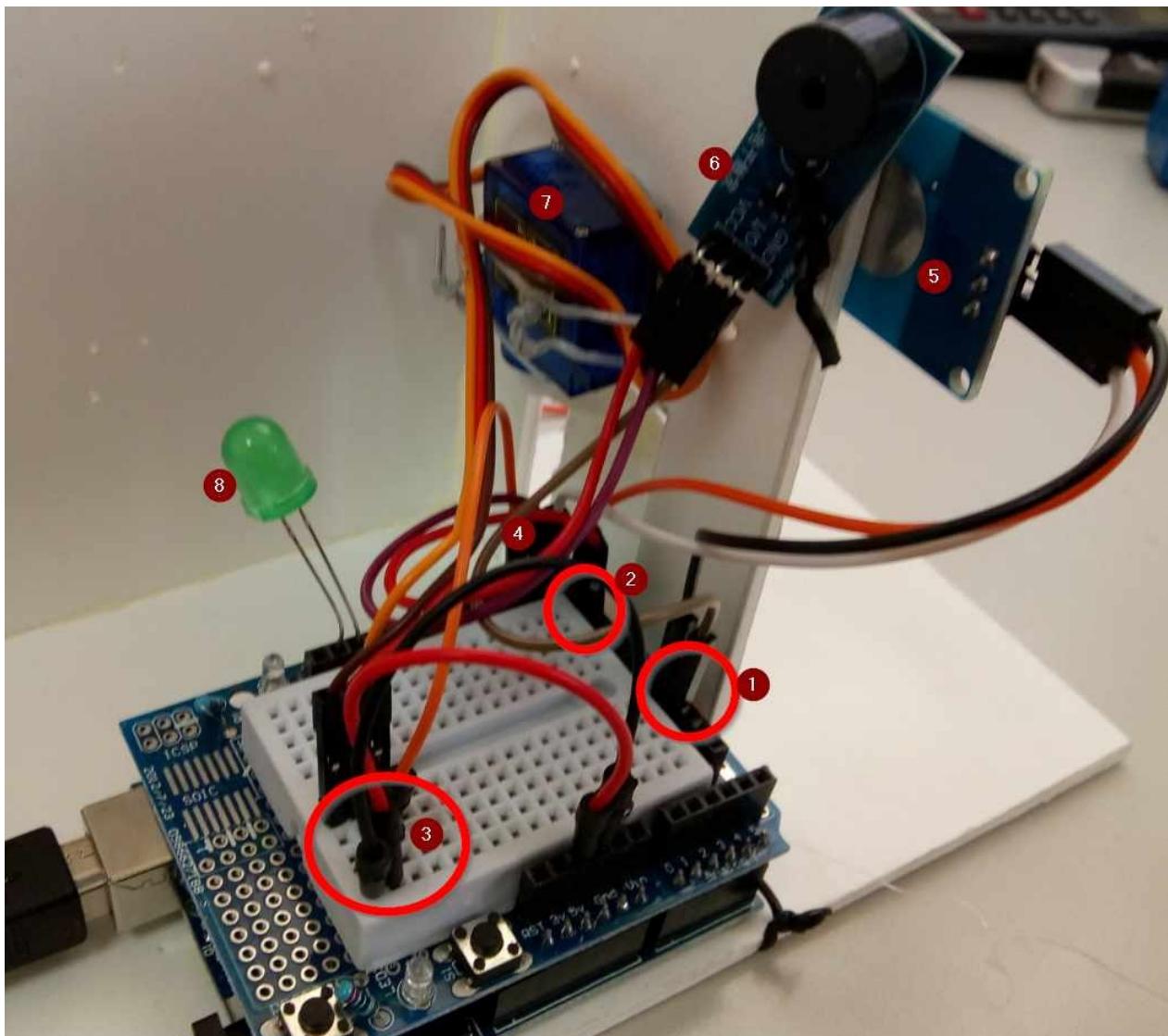


3.3.3 Esquema de conexiones

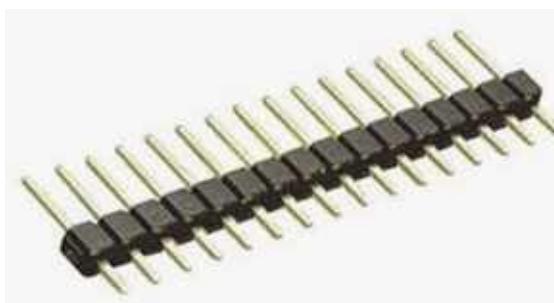
Es el mismo mostrado en [2.4 APERTURA DE LA PUERTA](#), pero ahora vamos montando la maqueta:

1. Terminales GND
2. Terminales +5V
3. Terminales Servo van aparte (ver nota) con D4 al control del servo
4. Terminales
 - i. D2 Interruptor táctil
 - ii. D3 Buzzer activo
5. Interruptor táctil

6. Buzzer activo
7. Servo
8. Led el pin largo en D13 y el corto en GND.

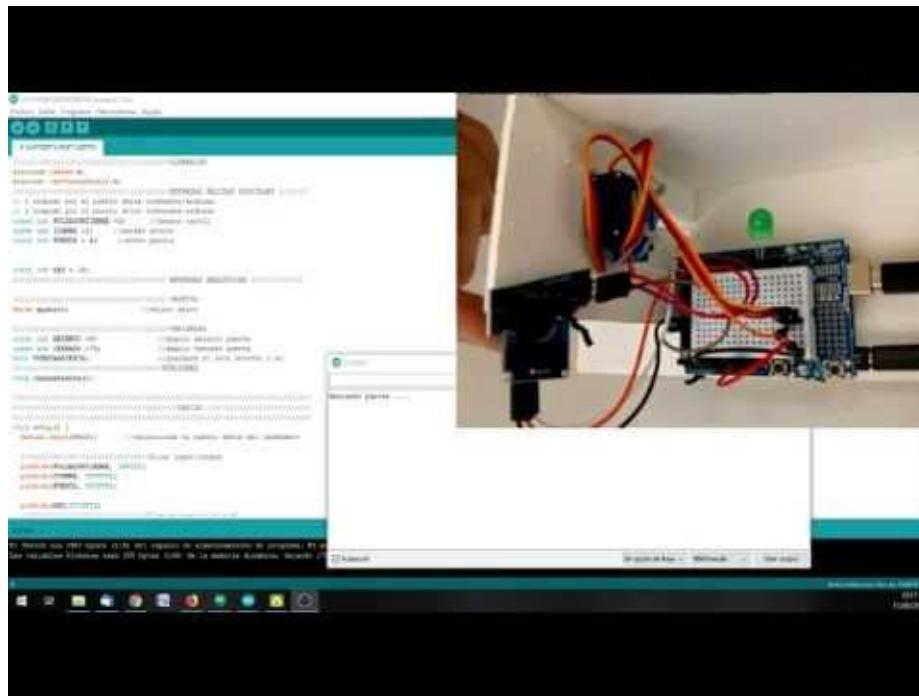


Nota: El servo al tener terminales hembra, se ha optado por conectarlo directamente en la placa protoboard utilizando machos como el de la figura:



Nota: Sí, ya sé que conectar el **LED DIRECTAMENTE EN D13** no es una buena práctica, pero internamente Arduino tiene unas resistencias en cada entrada/salida de 30k que lo puede aguantar, o sea, no es lo correcto pero no pasa nada (y como es muy cómodo, y necesitamos espacio, optamos por lo práctico). Si lo pones al revés no funcionará (no se estropea nada pero asegúrate el pin largo al D13 y el corto al GND).

3.3.4 Video



[Video link](#)

3.3.5 Código

```
//////////////////////////////LIBRERIAS
#include <Servo.h>

//////////////////////////////ENTRADAS SALIDAS DIGITALES //////////////////
// 0 ocupado por el puerto serie ordenador-arduino
// 1 ocupado por el puerto serie ordenador-arduino
const int PULSADORTIMBRE =2;          //sensor tactil
const int TIMBRE =3;                  //buzzer activo
const int PUERTA = 4;                //servo puerta

const int LED = 13;
////////////////////////////// ENTRADAS ANALÓGICAS //////////////////

////////////////////////////// OBJETOS
Servo myservo;                      //objeto servo

//////////////////////////////VARIABLES
const int ABIERTO =0;                 //abierto puerta
const int CERRADO =75;                //cerrado puerta, se ha modificado con respecto al
cierre total pues tropezaba con la pared
bool PUERTAABIERTA;                 //guardará si está abierto o no
//////////////////////////////FUNCIONES
void CerrarPuerta();

//////////////////////////////INICIO //////////////////
```

```

//////////void setup() {
  Serial.begin(9600);          //inicializar el puerto serie del ordenador

  ////////////fijar input/output
  pinMode(PULSADORTIMBRE, INPUT);
  pinMode(TIMBRE, OUTPUT);
  pinMode(PUERTA, OUTPUT);

  pinMode(LED, OUTPUT);
  ////////////fijar situación inicial
  digitalWrite(TIMBRE, HIGH);   // timbre apagado
  myservo.attach(PUERTA);      //servo en el pin correspondiente
  myservo.write(CERRADO);     //puerta cerrada
  PUERTAABIERTA = false;

}

//////////BUCLE ///////////
//////////void loop() {

  //////////// timbre ///////////
  if (digitalRead(PULSADORTIMBRE)){
    Serial.println("Abriendo puerta .... ");
    PUERTAABIERTA=true;
    myservo.write(ABIERTO);
    delay(3000);
    CerrarPuerta();
  }

}

//////////FUNCION CERRAR PUERTA/////////
//////////void CerrarPuerta(){

  if (PUERTAABIERTA){
    for (int i=1;i<=3;i++){
      Serial.println("CERRANDO PUERTA !!!");
      digitalWrite(LED, HIGH);
      digitalWrite(TIMBRE, LOW);
      delay(1000);
      digitalWrite(TIMBRE, HIGH);
      digitalWrite(LED, LOW);
      delay(1000);
    }
    myservo.write(CERRADO);
    PUERTAABIERTA=false;
    Serial.println("Puerta cerrada");
  }
}

```


3.4 JOYSTICK

3.4.1 Objetivos

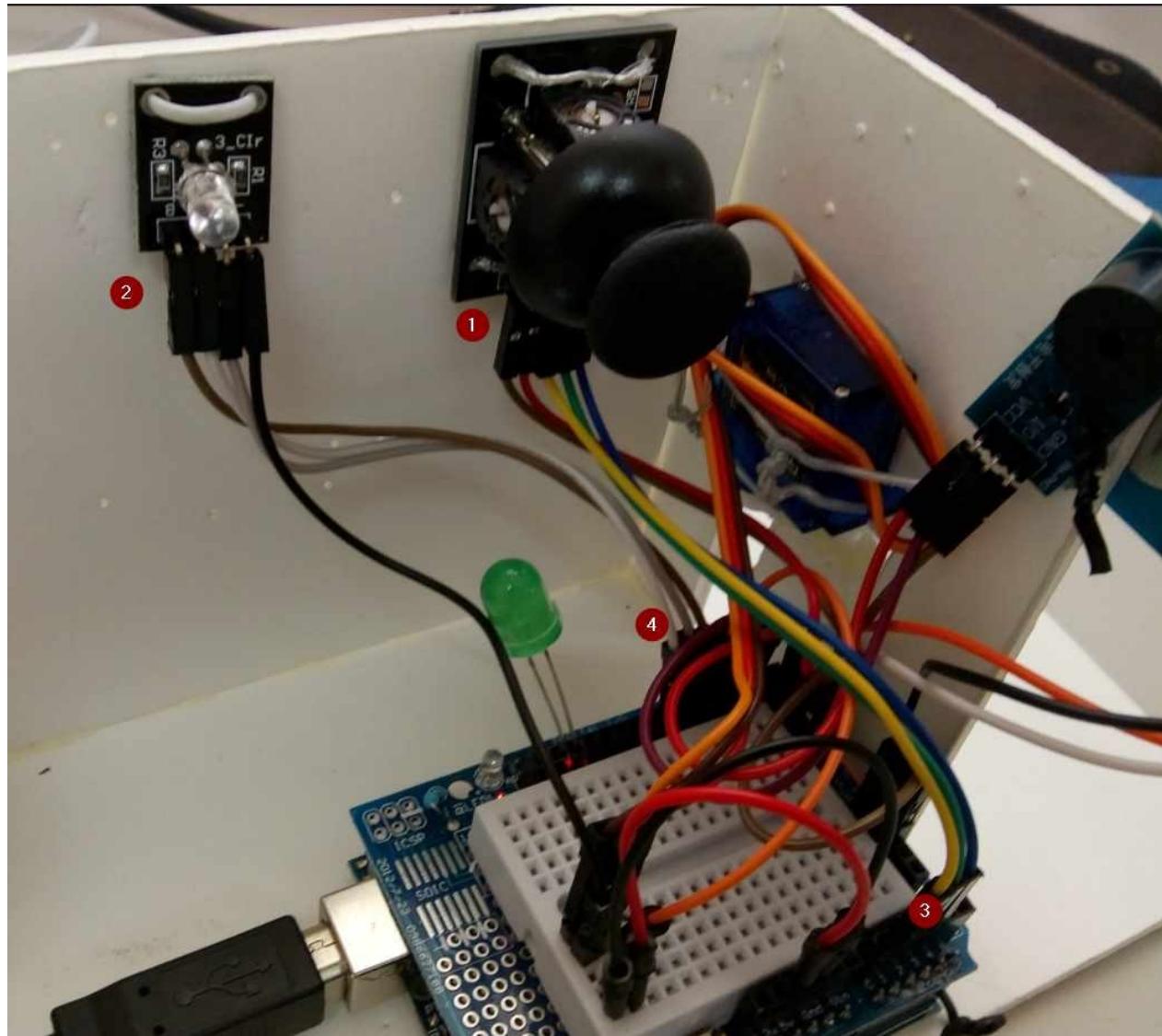
Ahora vamos a utilizar el Joystick para dos intenciones:

- Aprovechando el SWITCH central:
 - Si se hace una pulsación larga:
 - la puerta se abre (si estaba cerrada)
 - o se cierra (si estaba abierta)
- El mando del Joystick nos regulará una luz ambiental
 - El eje X en azul
 - El eje y el rojo
 - Un valor intermedio es un valor de los dos

3.4.2 Maqueta

Vamos ampliando nuestra casa domótica con la luz RGB y el Joystick:

1. Joystick fijado en la pared
2. Led RGB fijado en la pared
3. Conexiones Joystick en entradas analógicas
4. Conexiones de RGB a las salidas digitales



3.4.3 Esquema eléctrico

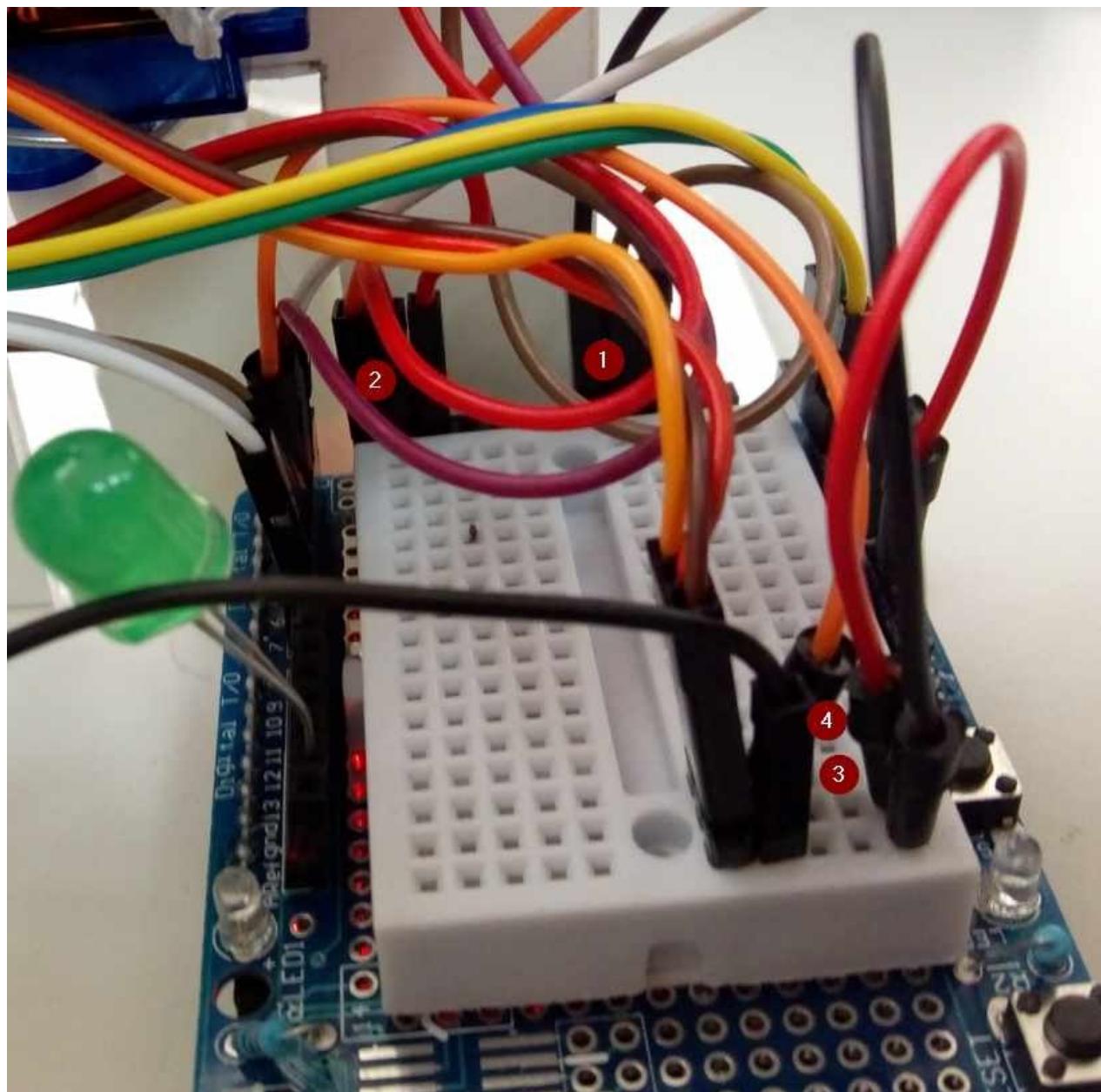
Es igual que cuando vimos [2.6 Joystick](#):

1. Terminales GND del led RGB y del Joystick
2. Terminal +5V del Joystick
3. Otra opción de conectar el terminal GND
4. Otra opción de conectar +5V

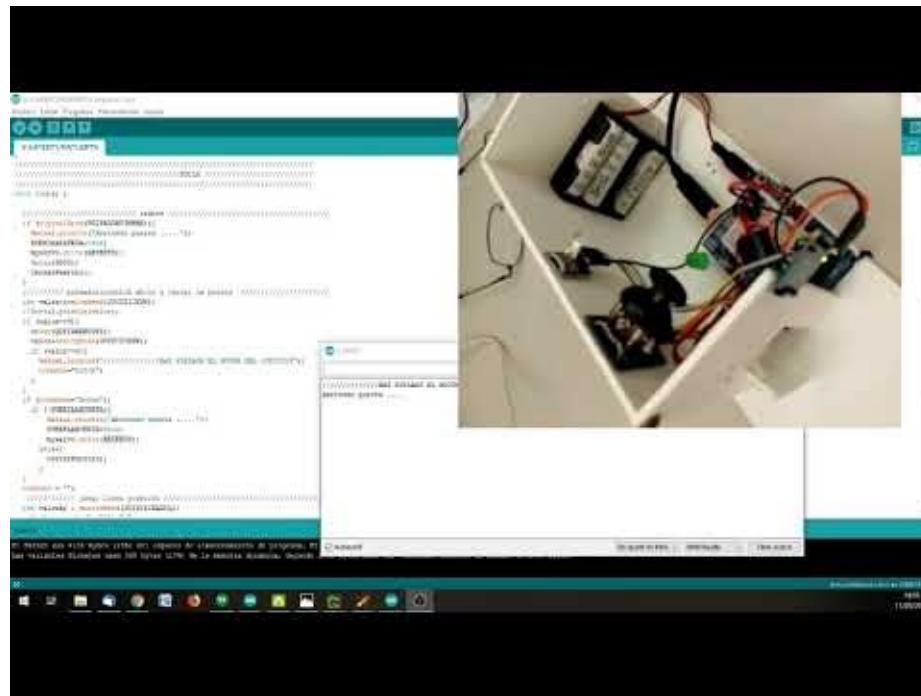
y las demás conexiones igual que antes:

- D5 PWM al Rojo del RGB (tiene que ser PWM)
- D6 PWM al Azul del RGB (tiene que ser PWM)
- D7 al Verde del RGB (luego lo utilizaremos)
- A1 al EJEX JOYSTICK
- A2 al EJEY JOYSTICK
- A3 al SWITCH JOYSTICK

Nota: El microruptor del Joystick **es digital** luego lo apropiado es conectarlo en una entrada salida digital, pero como vamos escasos de entradas y salidas digitales (ya verás como esto se complica), lo trataremos como analógico.



3.4.4 Vídeo



[Video link](#)

3.4.5 Código

```
//////////////////////////////LIBRERIAS
#include <Servo.h>
#include <SoftwareSerial.h>
//////////////////////////////ENTRADAS SALIDAS DIGITALES //////////////////
// 0 ocupado por el puerto serie ordenador-arduino
// 1 ocupado por el puerto serie ordenador-arduino
const int PULSADORTIMBRE =2;      //sensor tactil
const int TIMBRE =3;      //buzzer activo
const int PUERTA = 4;      //servo puerta
const int AZUL = 5;      //led RGB
const int ROJO = 6;      //led RGB
const int VERDE = 7;      //led RGB

const int LED = 13;
////////////////////////////// ENTRADAS ANALÓGICAS //////////////////
const int JOYSTICKAZUL = 1;
const int JOYSTICKROJO = 2;
const int JOYSTICKSW = 3;
////////////////////////////// OBJETOS
Servo myservo;           //objeto servo

//////////////////////////////VARIABLES
const int ABIERTO =0;          //ángulo abierto puerta
const int CERRADO =75;         //ángulo cerrado puerta, se ha modificado con respecto
a 90 que es cierre total pues tropezaba con la pared
bool PUERTAABIERTA;          //guardará si está abierto o no
const int QUITAREBOTES = 1000; //el tiempo para eliminar rebotes
```

```

String command = ""; // guardará la respuesta desde el BT
bool ENCENDIDO; //guardará que debe de dejar la luz encendida
//////////////////////////////FUNCIONES
void CerrarPuerta();

//////////////////////////////INICIO //////////////////////////////
void setup() {
    Serial.begin(9600); //inicializar el puerto serie del ordenador

    /////////////////////fijar input/output
    pinMode(PULSADORTIMBRE, INPUT);
    pinMode(TIMBRE, OUTPUT);
    pinMode(PUERTA, OUTPUT);
    pinMode(AZUL, OUTPUT);
    pinMode(ROJO, OUTPUT);
    pinMode(VERDE, OUTPUT);
    pinMode(LED,OUTPUT);

    /////////////////////fijar situación inicial
    digitalWrite(TIMBRE,HIGH); // timbre apagado
    myservo.attach(PUERTA); //servo en el pin correspondiente
    myservo.write(CERRADO); //puerta cerrada
    PUERTAABIERTA = false;
    digitalWrite(AZUL,LOW); //luz rgb apagada
    digitalWrite(ROJO,LOW); //luz rgb apagada
    digitalWrite(VERDE,LOW); //luz verde apagada

}

//////////////////////////////BUCLE //////////////////////////////
void loop() {

    ////////////////// timbre ///////////////////////////////
    if (digitalRead(PULSADORTIMBRE)){
        Serial.println("Abriendo puerta .... ");
        PUERTAABIERTA=true;
        myservo.write(ABIERTO);
        delay(3000);
        CerrarPuerta();
    }
    ////////// pulsadorjoystick abrir y cerrar la puerta //////////////////
    int valor=analogRead(JOYSTICKSW);
    //Serial.println(valor);
    if (valor==0){
        delay(QUITAREBOTES);
        valor=analogRead(JOYSTICKSW);
        if (valor==0){
            Serial.println("//////////HAS PULSADO EL BOTÓN DEL JOYSTICK");
            command="boton";
        }
    }
}

```

```

if (command=="boton"){
    if (!PUERTAABIERTA){
        Serial.println("Abriendo puerta .... ");
        PUERTAABIERTA=true;
        myservo.write(ABIERTO);
    }else{
        CerrarPuerta();
    }
}
command = "";
////////////////// juego luces joystick /////////////////////////////////
int valorAz = analogRead(JOYSTICKAZUL);
valorAz = (valorAz-500)/2.5;
if (valorAz<5) valorAz =0;
if (!ENCENDIDO)analogWrite(AZUL,valorAz);
int valorRo = analogRead(JOYSTICKROJO);
valorRo = (valorRo-550)/2.5;
if (valorRo<5) valorRo =0;
if (!ENCENDIDO) analogWrite(ROJO,valorRo);

}

/////////////////////////////// FUNCION CERRAR PUERTA/////////////////
void CerrarPuerta(){
    if (PUERTAABIERTA){
        for (int i=1;i<=3;i++){
            Serial.println("CERRANDO PUERTA !!!");
            digitalWrite(LED,HIGH);
            digitalWrite(TIMBRE,LOW);
            delay(1000);
            digitalWrite(TIMBRE,HIGH);
            digitalWrite(LED,LOW);
            delay(1000);
        }
        myservo.write(CERRADO);
        PUERTAABIERTA=false;
        Serial.println("Puerta cerrada");
    }
}

```

3.5 ALARMA AGUA

3.5.1 Objetivos

Ahora vamos a utilizar el sensor de agua para prevenir "accidentes" en casa

- Si el sensor de agua detecta líquido
 - Suena una alarma de 10 pulsos mientras esté funcionando
 - Si la puerta estuviera abierta
 - La cierra pues entraría agua

3.5.2 Maqueta

Ampliamos con el sensor de agua en una pared

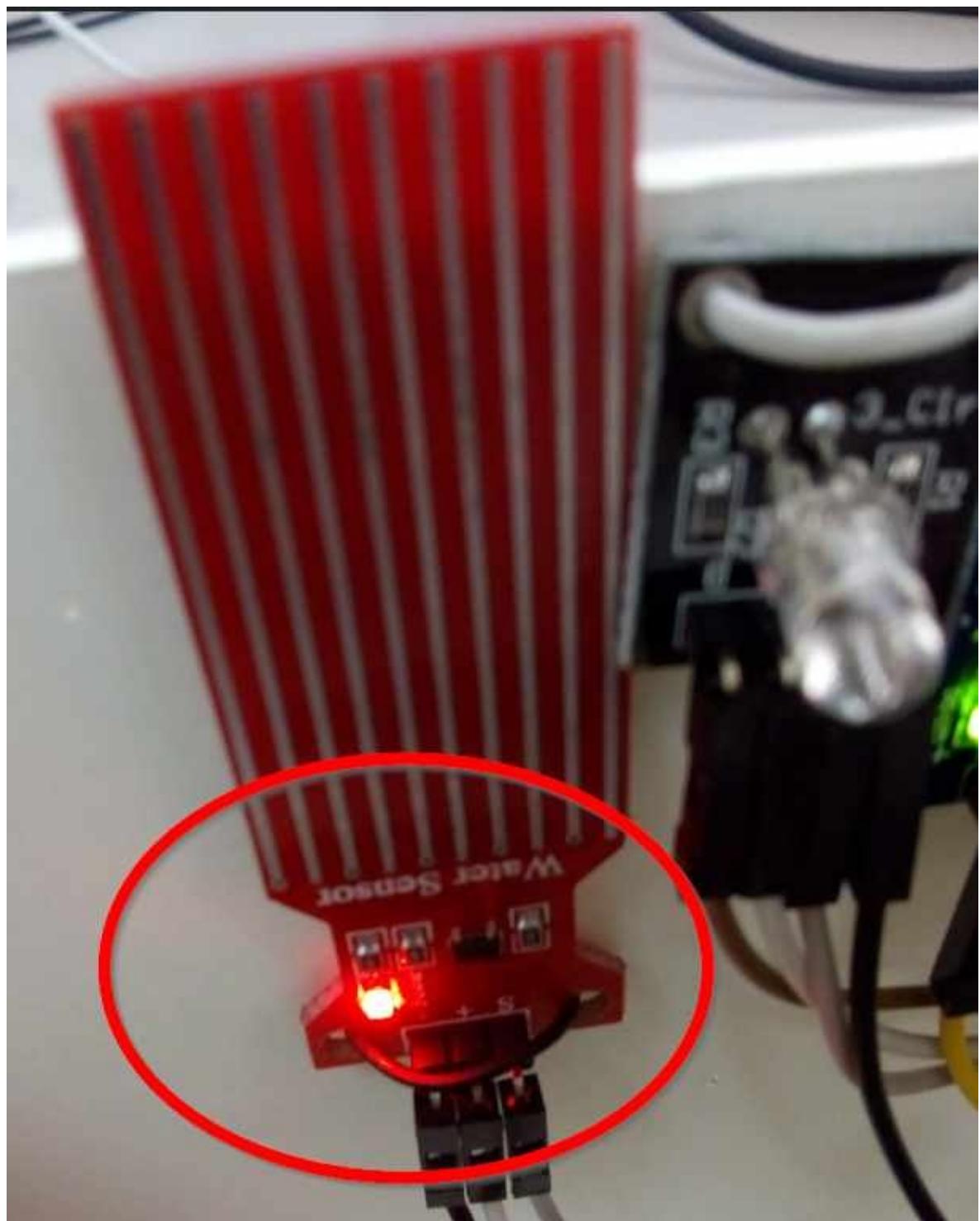
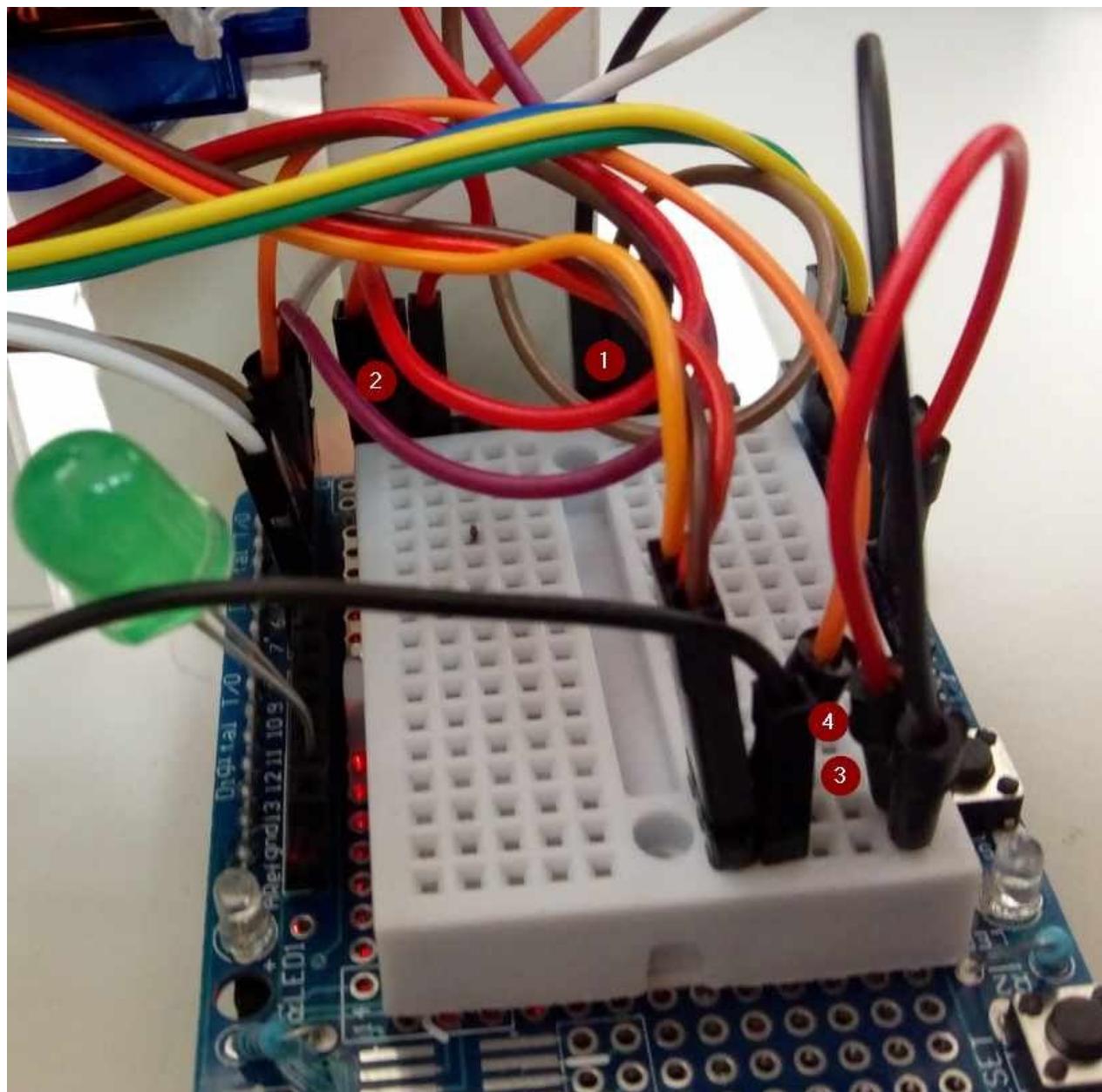


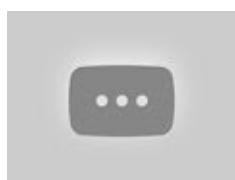
Imagen -

3.5.3 Esquema eléctrico

- Sensor de Agua en D8
- Alimentación GND y +5V del sensor de agua o en 1 y 2 o en 3 y 4



3.5.4 Vídeo



[Video link](#)

3.5.6 Código

```
//////////LIBRERIAS
#include <Servo.h>
#include <SoftwareSerial.h>
//////////ENTRADAS SALIDAS DIGITALES //////
// 0 ocupado por el puerto serie ordenador-arduino
```

```

// 1 ocupado por el puerto serie ordenador-arduino
const int PULSADORTIMBRE =2;          //sensor tactil
const int TIMBRE =3;                  //buzzer activo
const int PUERTA = 4;                //servo puerta
const int AZUL = 5;                  //led RGB
const int ROJO = 6;                  //led RGB
const int VERDE = 7;                  //led RGB
const int AGUA =8;                   //sensor agua

const int LED = 13;
////////////////////////////// ENTRADAS ANALÓGICAS ///////////////////
const int JOYSTICKAZUL = 1;
const int JOYSTICKROJO = 2;
const int JOYSTICKSW = 3;
////////////////////////////// OBJETOS
Servo myservo;                      //objeto servo

//////////////////////////////VARIABLES
const int ABIERTO =0;                 //ángulo abierto puerta
const int CERRADO =75;                //ángulo cerrado puerta, se ha modificado con respecto
a 90 que es cierre total pues tropezaba con la pared
bool PUERTAABIERTA;                 //guardará si está abierto o no
const int QUITAREBOTES = 1000;        //el tiempo para eliminar rebotes
String command = "";                 // guardará la respuesta desde el BT
bool ENCENDIDO;                     //guardará que debe de dejar la luz encendida
//////////////////////////////FUNCIONES
void CerrarPuerta();
void Alarma(String Que);

//////////////////////////////INICIO ///////////////////
void setup() {
    Serial.begin(9600);           //inicializar el puerto serie del ordenador

    /////////////////////fijar input/output
    pinMode(PULSADORTIMBRE, INPUT);
    pinMode(TIMBRE, OUTPUT);
    pinMode(PUERTA, OUTPUT);
    pinMode(AZUL, OUTPUT);
    pinMode(ROJO, OUTPUT);
    pinMode(VERDE, OUTPUT);
    pinMode(LED,OUTPUT);
    pinMode(AGUA,INPUT);

    /////////////////////fijar situación inicial
    digitalWrite(TIMBRE,HIGH); // timbre apagado
    myservo.attach(PUERTA);    //servo en el pin correspondiente
    myservo.write(CERRADO);   //puerta cerrada
    PUERTAABIERTA = false;
    digitalWrite(AZUL,LOW);   //luz rgb apagada
    digitalWrite(ROJO,LOW);   //luz rgb apagada
    digitalWrite(VERDE,LOW);  //luz verde apagada
}

```

```

}

//////////////////////////////BUCLE //////////////////////////////
//////////////////////////////BUCLE //////////////////////////////
void loop() {

    ////////////////// timbre /////////////////////////////////
    if (digitalRead(PULSADORTIMBRE)){
        Serial.println("Abriendo puerta .... ");
        PUERTAABIERTA=true;
        myservo.write(ABIERTO);
        delay(3000);
        CerrarPuerta();
    }
    ////////// pulsadorjoystick abrir y cerrar la puerta /////////////////////
    int valor=analogRead(JOYSTICKSW);
    //Serial.println(valor);
    if (valor==0){
        delay(QUITAREBOTES);
        valor=analogRead(JOYSTICKSW);
        if (valor==0){
            Serial.println("////////////////HAS PULSADO EL BOTÓN DEL JOYSTICK");
            command="boton";
        }
    }
    if (command=="boton"){
        if (!PUERTAABIERTA){
            Serial.println("Abriendo puerta .... ");
            PUERTAABIERTA=true;
            myservo.write(ABIERTO);
        }else{
            CerrarPuerta();
        }
    }
    command = "";
    ////////// juego luces joystick /////////////////////////////////
    int valorAz = analogRead(JOYSTICKAZUL);
    valorAz = (valorAz-500)/2.5;
    if (valorAz<5) valorAz =0;
    if (!ENCENDIDO)analogWrite(AZUL,valorAz);
    int valorRo = analogRead(JOYSTICKROJO);
    valorRo = (valorRo-550)/2.5;
    if (valorRo<5) valorRo =0;
    if (!ENCENDIDO) analogWrite(ROJO,valorRo);
    ////////// agua /////////////////////////////////
    if (digitalRead(AGUA)){
        Alarma("AGUA !!!");
        if (PUERTAABIERTA) CerrarPuerta();
    }
}

```

```
/////////////////////////////// FUNCION CERRAR PUERTA //////////////////
void CerrarPuerta(){
    if (PUERTAABIERTA){
        for (int i=1;i<=3;i++){
            Serial.println("CERRANDO PUERTA !!!");
            digitalWrite(LED,HIGH);
            digitalWrite(TIMBRE,LOW);
            delay(1000);
            digitalWrite(TIMBRE,HIGH);
            digitalWrite(LED,LOW);
            delay(1000);
        }
        myservo.write(CERRADO);
        PUERTAABIERTA=false;
        Serial.println("Puerta cerrada");
    }
}

/////////////////////////////// FUNCION ALARMA //////////////////
void Alarma(String Que){
    for (int i = 1;i<=10;i++){
        digitalWrite(TIMBRE,LOW);
        digitalWrite(LED,HIGH);
        String mensaje="ALARMA !!!!! "+Que;
        Serial.println(mensaje);
        delay(100);
        digitalWrite(TIMBRE,HIGH);
        digitalWrite(LED,LOW);
        mensaje="ATENCIÓN !!!!! "+Que;
        Serial.println(mensaje);
        delay(100);
    }
}
```

3.6 FUEGO

3.6.1 Objetivos

Tenemos que hacer dos programaciones

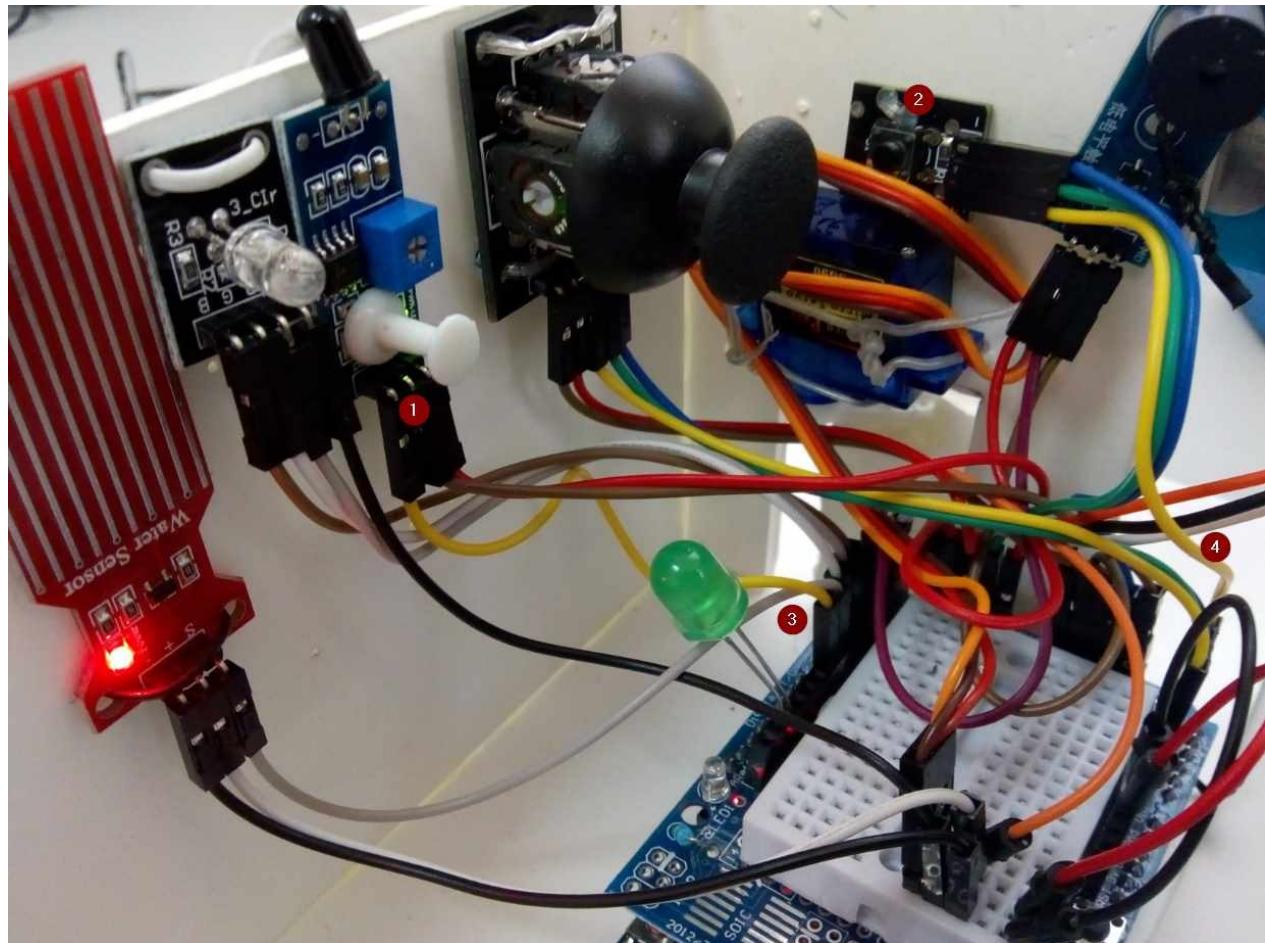
- Activar o no la alarma
 - cuando hay una pulsación larga en el pulsador escondido, se activa o se desactiva la alarma
 - si se ha disparado la alarma tiene que apagarla
- Disparo de la alarma
 - Si la alarma está activada, vigilando y se produce una detección, la alarma tiene que sonar hasta que la desactivamos con una pulsación larga del pulsador.

3.6.2 Maqueta

Fijaremos en la pared el sensor llama que lo vimos anteriormente en 2.7 y el pulsador de activación o no de la alarma.

1. Sensor llama
2. Pulsador analógico
3. Conexión sensor llama a la entrada digital
4. Conexión del pulsador analógico a la entrada analógica

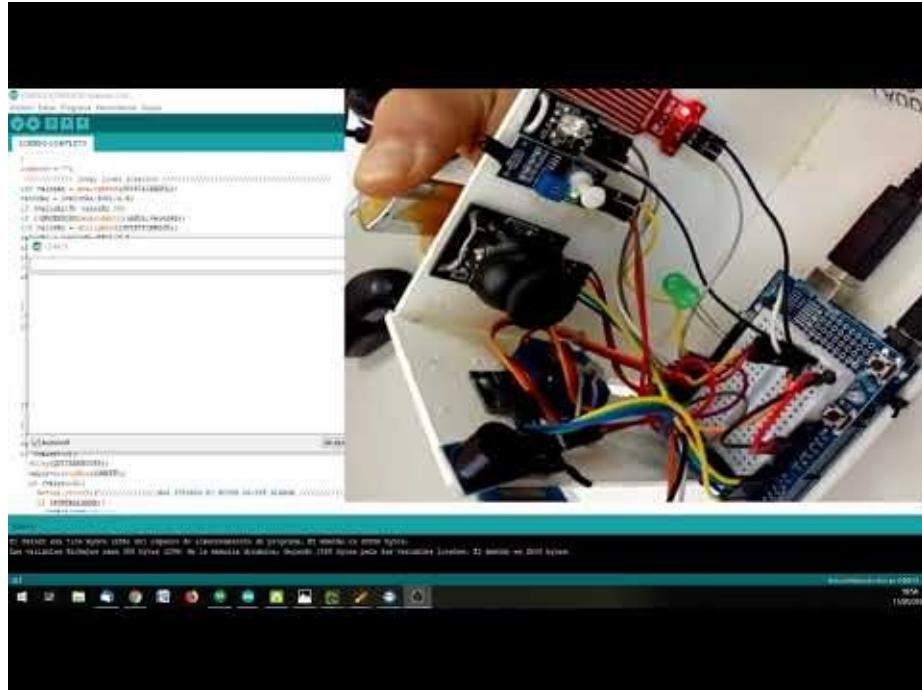
Las alimentaciones de los dos sensores igual que en las páginas anteriores.



3.6.3 Esquema eléctrico

- Sensor llama a D9
- Pulsador analógico a A4

3.6.4 Vídeo



[Video link](#)

3.6.5 Código

```
//////////LIBRERIAS
#include <Servo.h>
#include <SoftwareSerial.h>
//////////ENTRADAS SALIDAS DIGITALES //////
// 0 ocupado por el puerto serie ordenador-arduino
// 1 ocupado por el puerto serie ordenador-arduino
const int PULSADORTIMBRE =2;           //sensor tactil
const int TIMBRE =3;                   //buzzer activo
const int PUERTA = 4;                 //servo puerta
const int AZUL = 5;                   //led RGB
const int ROJO = 6;                   //led RGB
const int VERDE = 7;                  //led RGB
const int AGUA =8;                   //sensor agua
const int LLAMA=9;                  //sensor LLAMA

const int LED = 13;
////////// ENTRADAS ANALÓGICAS //////////
```

```

const int JOYSTICKAZUL = 1;           //potenciómetro eje x del joystick
const int JOYSTICKROJO = 2;           //potenciómetro eje y del joystick
const int JOYSTICKSW = 3;             //switch del joystick
const int ONOFF = 4;                 //interruptor que apaga o enciende la alarma
////////////////////////////////////////////// OBJETOS ///////////////////
Servo myservo;                      //objeto servo

//////////////////////////////VARIABLES///////////////////////////
const int ABIERTO =0;                //ángulo abierto puerta
const int CERRADO =75;               //ángulo cerrado puerta, se ha modificado con respecto
a 90 que es cierre total pues tropezaba con la pared
bool PUERTAABIERTA;                 //guardará si está abierto o no
const int QUITAREBOTES = 1000;        //el tiempo para eliminar rebotes
String command = "";                // guardará la respuesta desde el BT
bool ENCENDIDO;                    //guardará que debe de dejar la luz encendida
bool PONERALARMA;                  //guardará que las alarmas están activas
String QUESENSOR;                  //guarda qué sensor se ha disparado;
bool DISPARO;                       //guardará que ha ocurrido un disparo de alarma
//////////////////////////////FUNCIONES
void CerrarPuerta();
void Alarma(String Que);

//////////////////////////////INICIO ///////////////////
//////////////////////////////funciones/////////////////////
void setup() {
    Serial.begin(9600);           //inicializar el puerto serie del ordenador

    /////////////////////fijar input/output
    pinMode(PULSATORTIMBRE, INPUT);
    pinMode(TIMBRE, OUTPUT);
    pinMode(PUERTA, OUTPUT);
    pinMode(AZUL, OUTPUT);
    pinMode(ROJO, OUTPUT);
    pinMode(VERDE, OUTPUT);
    pinMode(LED, OUTPUT);
    pinMode(AGUA, INPUT);
    pinMode(LLAMA, INPUT);

    /////////////////////fijar situación inicial
    digitalWrite(TIMBRE,HIGH);   // timbre apagado
    myservo.attach(PUERTA);      //servo en el pin correspondiente
    myservo.write(CERRADO);     //puerta cerrada
    PUERTAABIERTA = false;
    digitalWrite(AZUL,LOW);     //luz rgb apagada
    digitalWrite(ROJO,LOW);     //luz rgb apagada
    digitalWrite(VERDE,LOW);    //luz verde apagada
    PONERALARMA = false;
    DISPARO = false;
    QUESENSOR = "";
}

//////////////////////////////BUCLE ///////////////////
//////////////////////////////funciones/////////////////////

```

```

void loop() {

    ///////////////// timbre /////////////////////////////////
    if (digitalRead(PULSADORTIMBRE)){
        Serial.println("Abriendo puerta .... ");
        PUERTAABIERTA=true;
        myservo.write(ABIERTO);
        delay(3000);
        CerrarPuerta();
    }
    ////////////// pulsadorjoystick abrir y cerrar la puerta /////////////////////
    int valor=analogRead(JOYSTICKSW);
    //Serial.println(valor);
    if (valor==0){
        delay(QUITAREBOTES);
        valor=analogRead(JOYSTICKSW);
        if (valor==0){
            Serial.println("////////////////HAS PULSADO EL BOTÓN DEL JOYSTICK");
            command="boton";
        }
    }
    if (command=="boton"){
        if (!PUERTAABIERTA){
            Serial.println("Abriendo puerta .... ");
            PUERTAABIERTA=true;
            myservo.write(ABIERTO);
        }else{
            CerrarPuerta();
        }
    }
    command = "";
    ////////////// juego luces joystick /////////////////////////////////
    int valorAz = analogRead(JOYSTICKAZUL);
    valorAz = (valorAz-500)/2.5;
    if (valorAz<5) valorAz =0;
    if (!ENCENDIDO)analogWrite(AZUL,valorAz);
    int valorRo = analogRead(JOYSTICKROJO);
    valorRo = (valorRo-550)/2.5;
    if (valorRo<5) valorRo =0;
    if (!ENCENDIDO) analogWrite(ROJO,valorRo);
    ////////////// agua /////////////////////////////////
    if (digitalRead(AGUA)){
        Alarma("AGUA !!!");
        if (PUERTAABIERTA) CerrarPuerta();
    }
    ////////////// ALARMA FUEGO /////////////////////////////////
    if (PONERALARMA){
        digitalWrite(VERDE,HIGH);
        if (digitalRead(LLAMA)==0){
            DISPARO=true;
            QUESENSOR ="FUEGO";
        }
        //////////////// DISPARO /////////////////////
    }
}

```

```

    if (DISPARO) Alarma(QUESENSOR);
}else{
    digitalWrite(VERDE,LOW);
}
////////////////// ONOFF ///////////////////
valor=analogRead(ONOFF);
if (valor==0){
    delay(QUITAREBOTES);
    valor=analogRead(ONOFF);
    if (valor==0){
        Serial.println("////////////////HAS PULSADO EL BOTÓN ON/OFF ALARMA //////////////////");
        if (PONERALARMA){
            PONERALARMA=false;
            DISPARO = false;
        }else{
            PONERALARMA=true;
        }
    }
}
}

/////////////////////////////// FUNCION CERRAR PUERTA///////////////////
void CerrarPuerta(){
if (PUERTAABIERTA){
    for (int i=1;i<=3;i++){
        Serial.println("CERRANDO PUERTA !!!");
        digitalWrite(LED,HIGH);
        digitalWrite(TIMBRE,LOW);
        delay(1000);
        digitalWrite(TIMBRE,HIGH);
        digitalWrite(LED,LOW);
        delay(1000);
    }
    myservo.write(CERRADO);
    PUERTAABIERTA=false;
    Serial.println("Puerta cerrada");
}
}

/////////////////////////////// FUNCION ALARMA ///////////////////
void Alarma(String Que){
for (int i = 1;i<=10;i++){
    digitalWrite(TIMBRE,LOW);
    digitalWrite(LED,HIGH);
    String mensaje="ALARMA !!!!!!"+Que;
}
}

```

```
Serial.println(mensaje);
delay(100);
digitalWrite(TIMBRE,HIGH);
digitalWrite(LED,LOW);
mensaje="ATENCIÓN !!!!"+Que;
Serial.println(mensaje);
delay(100);
}
}
```

3.7 ALARMA LÁSER

3.7.1 Conocimientos previos

El diodo láser es un elemento motivador, barato y fácil de usar con el Arduino pues se activa digitalmente:



Si quieras saber más de este componente, te recomendamos [esta página de Luis Llamas](#). Si tienes que comprar uno, te recomendamos que no sea superior a 5mW, pues puede dañar permanentemente la retina del ojo [+info]. El modelo que te proponemos es de 1mW, no obstante, **EVITA SIEMPRE QUE EL LÁSER APUNTE A LOS OJOS** especialmente con niños.

3.7.1 Objetivos

Es el mismo objetivo que el la alarma FUEGO pero que esta vez:

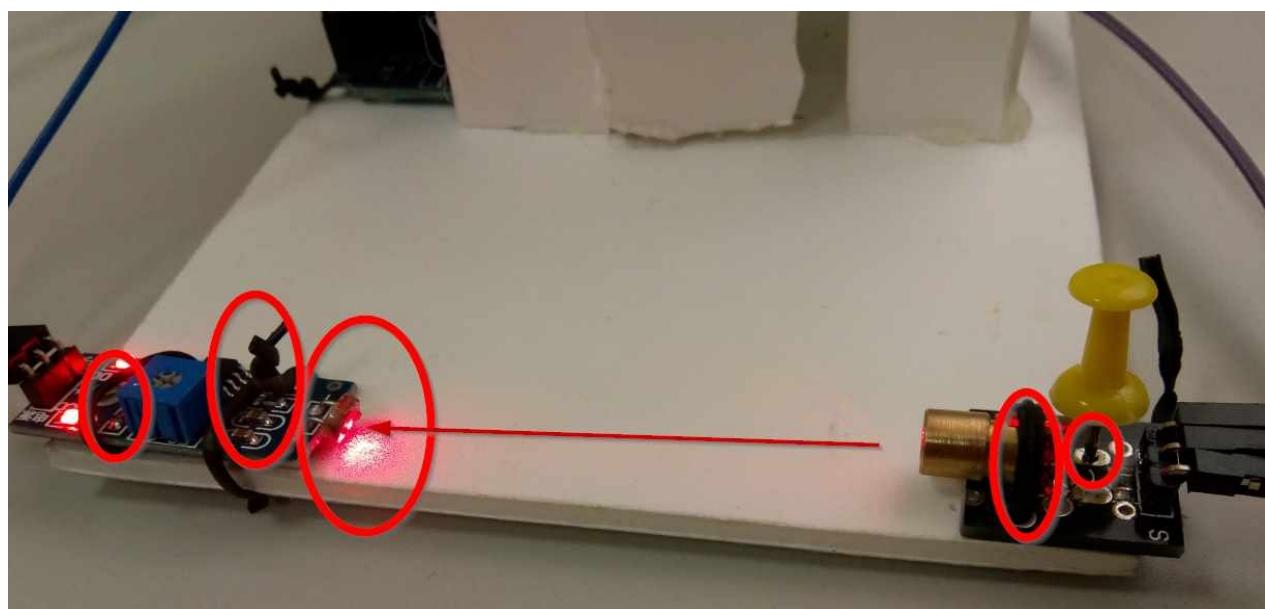
- Si activo la alarma el láser tiene que encenderse
 - Una vez activada si se corta el láser, por lo tanto el valor del LDR sube, la alarma se dispara.
- Si desactivo la alarma
 - La alarma se apaga si se ha disparado
 - El laser de apaga.



3.7.3 Maqueta

En este caso **hay que tener cuidado con la fijación del láser y el LDR para que apunte al LDR** y otra cuestión son los cables: al instalarse fuera de la casa hay que utilizar cables largos o añadir dos M-H consecutivos.

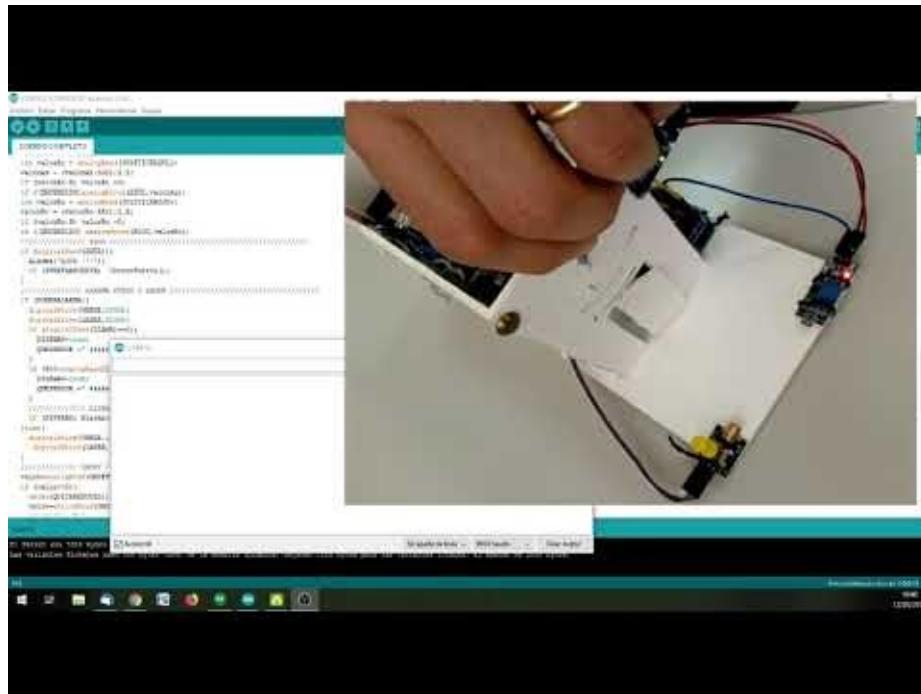
- Hemos utilizado dos fijaciones con alambre el LDR para que quede fijo.
- En el láser hemos fijado con una chincheta para fijar el láser además de alambre



3.7.4 Conexiones eléctricas

- La alimentación del láser y el del LDR igual que en los casos anteriores
- El control del LÁSER a la salida digital D10
- La lectura del LDR a la entrada analógica A0

3.7.5 Vídeo



[Video link](#)

3.7.6 Código

```
//////////////////////////////LIBRERIAS
#include <Servo.h>
#include <SoftwareSerial.h>
//////////////////////////////ENTRADAS SALIDAS DIGITALES //////
// 0 ocupado por el puerto serie ordenador-arduino
// 1 ocupado por el puerto serie ordenador-arduino
const int PULSADORTIMBRE =2;           //sensor tactil
const int TIMBRE =3;                    //buzzer activo
const int PUERTA = 4;                  //servo puerta
const int AZUL = 5;                    //led RGB
const int ROJO = 6;                    //led RGB
const int VERDE = 7;                   //led RGB
const int AGUA =8;                     //sensor agua
const int LLAMA=9;                    //sensor LLAMA
const int LASER=10;                   //diodo LASER

const int LED = 13;
////////////////////////////// ENTRADAS ANALÓGICAS ///////////////////
const int LDR = 0;                     //sensor luz LDR
const int JOYSTICKAZUL = 1;            //potenciómetro eje x del joystick
const int JOYSTICKROJO = 2;            //potenciómetro eje y del joystick
const int JOYSTICKSW   = 3;            //switch del joystick
```

```

const int ONOFF      = 4;           //interruptor que apaga o enciende la alarma
////////////////////////////////////////////// OBJETOS/////////////////
Servo myservo;                  //objeto servo

//////////////////////////////////////////////VARIABLES/////////////////
const int ABIERTO =0;             //ángulo abierto puerta
const int CERRADO =75;            //ángulo cerrado puerta, se ha modificado con respecto
a 90 que es cierre total pues tropezaba con la pared
bool PUERTAABIERTA;              //guardará si está abierto o no
const int QUITAREBOTES = 1000;     //el tiempo para eliminar rebotes
String command = "";              // guardará la respuesta desde el BT
bool ENCENDIDO;                 //guardará que debe de dejar la luz encendida
bool PONERALARMA;                //guardará que las alarmas están activas
String QUESENSOR;                //guarda qué sensor se ha disparado;
bool DISPARO;                     //guardará que ha ocurrido un disparo de alarma
//////////////////////////////////////////////FUNCIONES
void CerrarPuerta();
void Alarma(String Que);

//////////////////////////////////////////////INICIO ///////////////////
void setup() {
    Serial.begin(9600);           //inicializar el puerto serie del ordenador

    /////////////////////fijar input/output
    pinMode(PULSADORTIMBRE, INPUT);
    pinMode(TIMBRE, OUTPUT);
    pinMode(PUERTA, OUTPUT);
    pinMode(AZUL, OUTPUT);
    pinMode(ROJO, OUTPUT);
    pinMode(VERDE, OUTPUT);
    pinMode(LED, OUTPUT);
    pinMode(AGUA, INPUT);
    pinMode(LLAMA, INPUT);
    pinMode(LASER, OUTPUT);
    /////////////////////fijar situación inicial
    digitalWrite(TIMBRE,HIGH);   // timbre apagado
    myservo.attach(PUERTA);       //servo en el pin correspondiente
    myservo.write(CERRADO);       //puerta cerrada
    PUERTAABIERTA = false;
    digitalWrite(AZUL,LOW);      //luz rgb apagada
    digitalWrite(ROJO,LOW);      //luz rgb apagada
    digitalWrite(VERDE,LOW);      //luz verde apagada
    digitalWrite(LASER,LOW);
    PONERALARMA = false;
    DISPARO = false;
    QUESENSOR = "";
}

//////////////////////////////////////////////BUCLE ///////////////////
void loop() {

```

```

/////////// timbre /////////////
if (digitalRead(PULSADORTIMBRE)){
    Serial.println("Abriendo puerta .... ");
    PUERTAABIERTA=true;
    myservo.write(ABIERTO);
    delay(3000);
    CerrarPuerta();
}

///////// pulsadorjoystick abrir y cerrar la puerta ///////////
int valor=analogRead(JOYSTICKSW);
//Serial.println(valor);
if (valor==0){
    delay(QUITAREBOTES);
    valor=analogRead(JOYSTICKSW);
    if (valor==0){
        Serial.println("//HAS PULSADO EL BOTÓN DEL JOYSTICK");
        command="boton";
    }
}
if (command=="boton"){
    if (!PUERTAABIERTA){
        Serial.println("Abriendo puerta .... ");
        PUERTAABIERTA=true;
        myservo.write(ABIERTO);
    }else{
        CerrarPuerta();
    }
}
command = "";
///////// juego luces joystick ///////////
int valorAz = analogRead(JOYSTICKAZUL);
valorAz = (valorAz-500)/2.5;
if (valorAz<5) valorAz =0;
if (!ENCENDIDO)analogWrite(AZUL,valorAz);
int valorRo = analogRead(JOYSTICKROJO);
valorRo = (valorRo-550)/2.5;
if (valorRo<5) valorRo =0;
if (!ENCENDIDO) analogWrite(ROJO,valorRo);

///////// agua ///////////
if (digitalRead(AGUA)){
    Alarma("AGUA !!!");
    if (PUERTAABIERTA) CerrarPuerta();
}

///////// ALARMA FUEGO y LASER ///////////
if (PONERALARMA){
    digitalWrite(VERDE,HIGH);
    digitalWrite(LASER,HIGH);
    if (digitalRead(LLAMA)==0){
        DISPARO=true;
        QUESENSOR =" ##### FUEGO #####";
    }
    if (800<analogRead(LDR)) {
}
}

```

```

DISPARO=true;
QUESENSOR =" ##### LADRON ######";
}
////////// DISPARO //////
if (DISPARO) Alarma(QUESENSOR);
else{
  digitalWrite(VERDE,LOW);
  digitalWrite(LASER,LOW);
}
////////// ONOFF //////
valor=analogRead(ONOFF);
if (valor==0){
  delay(QUITAREBOTES);
  valor=analogRead(ONOFF);
  if (valor==0){
    Serial.println("//////////HAS PULSADO EL BOTÓN ON/OFF ALARMA //////////");
    if (PONERALARMA){
      PONERALARMA=false;
      DISPARO = false;
    }else{
      PONERALARMA=true;
    }
  }
}
}

////////// FUNCION CERRAR PUERTA/////////
void CerrarPuerta(){
if (PUERTAABIERTA){
  for (int i=1;i<=3;i++){
    Serial.println("CERRANDO PUERTA !!!");
    digitalWrite(LED,HIGH);
    digitalWrite(TIMBRE,LOW);
    delay(1000);
    digitalWrite(TIMBRE,HIGH);
    digitalWrite(LED,LOW);
    delay(1000);
  }
  myservo.write(CERRADO);
  PUERTAABIERTA=false;
  Serial.println("Puerta cerrada");
}
}

////////// FUNCION ALARMA //////

```

```
void Alarma(String Que){  
    for (int i = 1;i<=10;i++){  
        digitalWrite(TIMBRE,LOW);  
        digitalWrite(LED,HIGH);  
        String mensaje="ALARMA !!!!!!"+Que;  
        Serial.println(mensaje);  
        delay(100);  
        digitalWrite(TIMBRE,HIGH);  
        digitalWrite(LED,LOW);  
        mensaje="ATENCIÓN !!!!!!"+Que;  
        Serial.println(mensaje);  
        delay(100);  
    }  
}
```

3.8 BLUETOOTH

3.8.1 Conocimientos previos

Tienes que visitar las siguientes páginas de la [Unidad 4 Comunicaciones con Arduino](#):

- [¿Qué es el HC-06?](#)
- [La APP que tienes que intalarte](#)
- [Vincular tu móvil con el HC-06](#)

Aprende a configurar los botones de la APP !! diapositiva 12 pero en vez de Up, Down, Right y Left los que se establezcan en 3.8.2



- [Configuración avanzada](#) pues **nosotros conectaremos el HC-06 en los pines digitales D11 y D12** y no en D0 y D1 pues están ocupados con la comunicación del ordenador.

3.8.2 Objetivos

Controlar nuestra casa con el móvil, para ello vamos a definir los siguientes comandos:

COMANDO	VOZ	DATO	descripción
---------	-----	------	-------------

Comando 1	abrir	A	abrir la puerta y cierra automáticamente
Comando 2	puerta	P	abrir/cerrar la puerta
Comando 3	alarma	L	activar/desactivar la alarma
Comando 4	pit	T	hace un pit
Comando 5	rojo	R	enciende luz interior roja
Comando 6	azul	B	enciende luz interior azul
Comando 7	apaga	O	apaga luces interiores

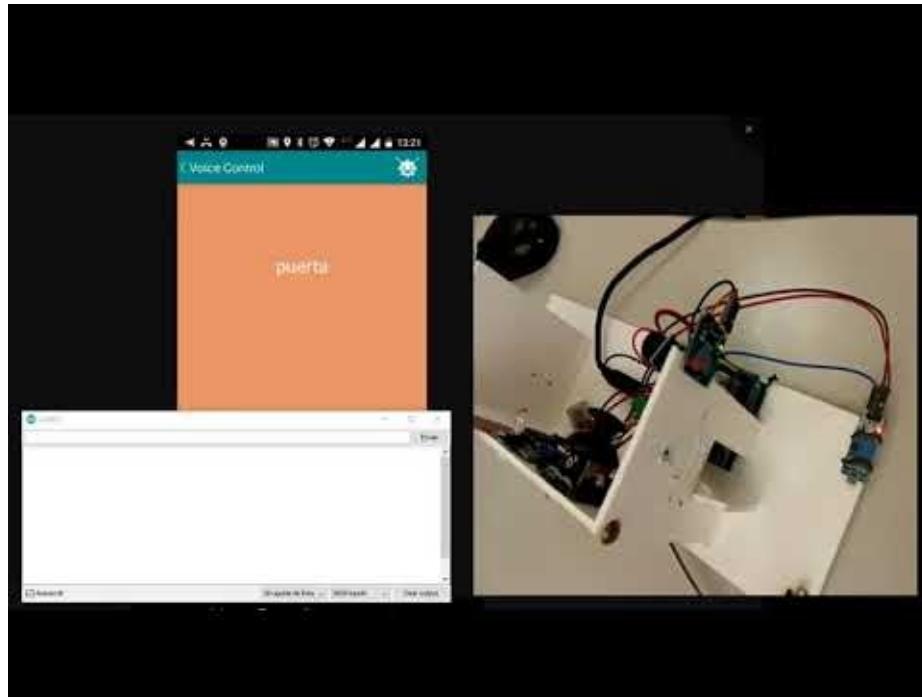
Lo tienes que hacer así:



3.8.3 Conexión eléctrica

- Conectaremos Vcc y GND del HC06 igual que antes en la placa protoboard *sí ya sé que casi no queda sitio, es el último, lo prometo*
- TX de HC06 al pin D11
- RX de HC06 al pin D12

3.8.4 Vídeo



[Video link](#)

3.8.5 Código

```
//////////////////////////////LIBRERIAS
#include <Servo.h>
#include <SoftwareSerial.h>
//////////////////////////////ENTRADAS SALIDAS DIGITALES //////////////////
// 0 ocupado por el puerto serie ordenador-arduino
// 1 ocupado por el puerto serie ordenador-arduino
const int PULSADORTIMBRE =2;           //sensor tactil
const int TIMBRE =3;                    //buzzer activo
const int PUERTA = 4;                  //servo puerta
const int AZUL = 5;                   //led RGB
const int ROJO = 6;                   //led RGB
const int VERDE = 7;                  //led RGB
const int AGUA =8;                   //sensor agua
const int LLAMA=9;                   //sensor LLAMA
const int LASER=10;                  //diodo LASER
const int RX = 11;                   //transmisión Bluetooth TX
const int TX = 12;                   //transmisión Bluetooth RX
const int LED = 13;                  //led interno
////////////////////////////// ENTRADAS ANALÓGICAS /////////////////////
const int LDR = 0;                   //sensor luz LDR
const int JOYSTICKAZUL = 1;          //potenciómetro eje x del joystick
const int JOYSTICKROJO = 2;          //potenciómetro eje y del joystick
const int JOYSTICKSW = 3;            //switch del joystick
const int ONOFF = 4;                 //interruptor que apaga o enciende la alarma
////////////////////////////// OBJETOS///////////////////
Servo myservo;                      //objeto servo
SoftwareSerial Serial1(RX, TX); // objeto puerto serie1 del Bluetooth
//////////////////////////////VARIABLES///////////////////
```

```

const int ABIERTO =0;           //ángulo abierto puerta
const int CERRADO =75;          //ángulo cerrado puerta, se ha modificado con respecto
a 90 que es cierre total pues tropezaba con la pared
bool PUERTAABIERTA;            //guardará si está abierto o no
const int QUITAREBOTES = 1000;   //el tiempo para eliminar rebotes
String command = "";           // guardará la respuesta desde el BT
bool ENCENDIDO;                //guardará que debe de dejar la luz encendida
bool PONERALARMA;              //guardará que las alarmas están activas
String QUESENSOR;              //guarda qué sensor se ha disparado;
bool DISPARO;                  //guardará que ha ocurrido un disparo de alarma
///////////////////////////////FUNCIONES
void CerrarPuerta();
void Alarma(String Que);
void ComandoBT();

//////////////////////////////INICIO //////////////////////////////
void setup() {
    Serial.begin(9600);          //inicializar el puerto serie del ordenador
    Serial1.begin(9600);          //CONEXION SERIE PARA EL MODULO BT
    //////////////////////////////fijar input/output
    pinMode(PULSADORTIMBRE, INPUT);
    pinMode(TIMBRE, OUTPUT);
    pinMode(PUERTA, OUTPUT);
    pinMode(AZUL, OUTPUT);
    pinMode(ROJO, OUTPUT);
    pinMode(VERDE, OUTPUT);
    pinMode(LED, OUTPUT);
    pinMode(AGUA, INPUT);
    pinMode(LLAMA, INPUT);
    pinMode(LASER, OUTPUT);
    //////////////////////////////fijar situación inicial
    digitalWrite(TIMBRE,HIGH);   // timbre apagado
    myservo.attach(PUERTA);       //servo en el pin correspondiente
    myservo.write(CERRADO);       //puerta cerrada
    PUERTAABIERTA = false;
    digitalWrite(AZUL,LOW);      //luz rgb apagada
    digitalWrite(ROJO,LOW);      //luz rgb apagada
    digitalWrite(VERDE,LOW);      //luz verde apagada
    digitalWrite(LASER,LOW);
    PONERALARMA = false;
    DISPARO = false;
    QUESENSOR = "";
    command="";
}
//////////////////////////////BUCLE //////////////////////////////
void loop() {
    ////////////////////// bluetooth /////////////////////
    ComandoBT();
}

```

```

//////////////// timbre /////////////////////////////////
if ((digitalRead(PULSADORTIMBRE))||(command=="A")){
    Serial.println("Abriendo puerta .... ");
    PUERTAABIERTA=true;
    myservo.write(ABIERTO);
    delay(3000);
    CerrarPuerta();
}
//////////////// pulsadorjoystick abrir y cerrar la puerta /////////////////////
int valor=analogRead(JOYSTICKSW);
//Serial.println(valor);
if (valor==0){
    delay(QUITAREBOTES);
    valor=analogRead(JOYSTICKSW);
    if (valor==0){
        Serial.println("///////////HAS PULSADO EL BOTÓN DEL JOYSTICK");
        command="P";
    }
}
if (command=="P"){
    if (!PUERTAABIERTA){
        Serial.println("Abriendo puerta .... ");
        PUERTAABIERTA=true;
        myservo.write(ABIERTO);
    }else{
        CerrarPuerta();
    }
}
//////////////// juego luces joystick /////////////////////////////////
int valorAz = analogRead(JOYSTICKAZUL);
valorAz = (valorAz-500)/2.5;
if (valorAz<5) valorAz =0;
if (!ENCENDIDO)analogWrite(AZUL,valorAz);
int valorRo = analogRead(JOYSTICKROJO);
valorRo = (valorRo-550)/2.5;
if (valorRo<5) valorRo =0;
if (!ENCENDIDO) analogWrite(ROJO,valorRo);
if (command=="R") analogWrite(ROJO,255);
if (command=="B") analogWrite(AZUL,255);
if (command=="O"){
    analogWrite(ROJO,0);
    analogWrite(AZUL,0);
}
//////////////// agua /////////////////////////////////
if (digitalRead(AGUA)){
    Alarma("AGUA !!!");
    if (PUERTAABIERTA) CerrarPuerta();
}
//////////////// ALARMA FUEGO y LASER /////////////////////////////////
if (PONERALARMA){
    digitalWrite(VERDE,HIGH);
    digitalWrite(LASER,HIGH);
    delay(QUITAREBOTES); //se pone un poco de retraso pues había falsos disparos
}

```

```

if (digitalRead(LLAMA)==0){
    DISPARO=true;
    QUESENSOR =" ##### FUEGO ######";
}
if (800<analogRead(LDR)){
    DISPARO=true;
    QUESENSOR =" ##### LADRON ######";
}
////////// DISPARO ///////////
if (DISPARO) Alarma(QUESENSOR);
}else{
    digitalWrite(VERDE,LOW);
    digitalWrite(LASER,LOW);
}
////////// ONOFF ///////////
valor=analogRead(ONOFF);
if (valor==0){
    delay(QUITAREBOTES);
    valor=analogRead(ONOFF);
    if (valor==0){
        Serial.println("//////////HAS PULSADO EL BOTÓN ON/OFF ALARMA ///////////");
        command="L";
    }
}
if (command=="L"){
    if (PONERALARMA){
        PONERALARMA=false;
        DISPARO = false;
    }else{
        PONERALARMA=true;
    }
}
if (command=="T") Alarma("PIIIIT");
}

///////////////////////////////
//////////////////FUNCION CERRAR PUERTA///////////////////
///////////////////////////////

void CerrarPuerta(){
    if (PUERTAABIERTA){
        for (int i=1;i<=3;i++){
            Serial.println("CERRANDO PUERTA !!!");
            digitalWrite(LED,HIGH);
            digitalWrite(TIMBRE,LOW);
            delay(1000);
            digitalWrite(TIMBRE,HIGH);
            digitalWrite(LED,LOW);
            delay(1000);
        }
    }
}

```

```

    }
    myservo.write(CERRADO);
    PUERTAABIERTA=false;
    Serial.println("Puerta cerrada");
}
}

////////////////////////////// FUNCION ALARMA ///////////////////
////////////////////////////// FUNCION LEER COMANDO BLUETOOTH////////////////

void Alarma(String Que){
    for (int i = 1;i<=10;i++){
        digitalWrite(TIMBRE,LOW);
        digitalWrite(LED,HIGH);
        String mensaje="ALARMA !!!!!!"+Que;
        Serial.println(mensaje);
        delay(100);
        digitalWrite(TIMBRE,HIGH);
        digitalWrite(LED,LOW);
        mensaje="ATENCIÓN !!!!!!"+Que;
        Serial.println(mensaje);
        delay(100);
    }
}

void ComandoBT(){
    if((command=="R")||(command=="B")){
    }else{
        command="";
    }
    if (Serial1.available()) {
        while(Serial1.available()) {
            command += (char)Serial1.read();
        }
        Serial.println("Comando recibido por Bluetooth = " );
        Serial.println(command);
    }
}
}

```

Pensamiento computacional

¿Dónde se encaja este robot? ¿se puede comparar este robot con otros robots de otros cursos que hacemos desde CATEDU?

Esta es la hoja de ruta que proponemos, no se tiene que tomar al pie de la letra, pero intenta ayudar al profesorado que tenga una visión global de tanta oferta:

Como se puede ver **DOMOTICA CON ARDUINO** es una alternativa muy buena y barata para que los docentes que quieran empezar con la robótica con Arduino.

RoboTICa

Oferta de formación en Pensamiento computacional del Centro Aragonés de Tecnologías para la Educación.



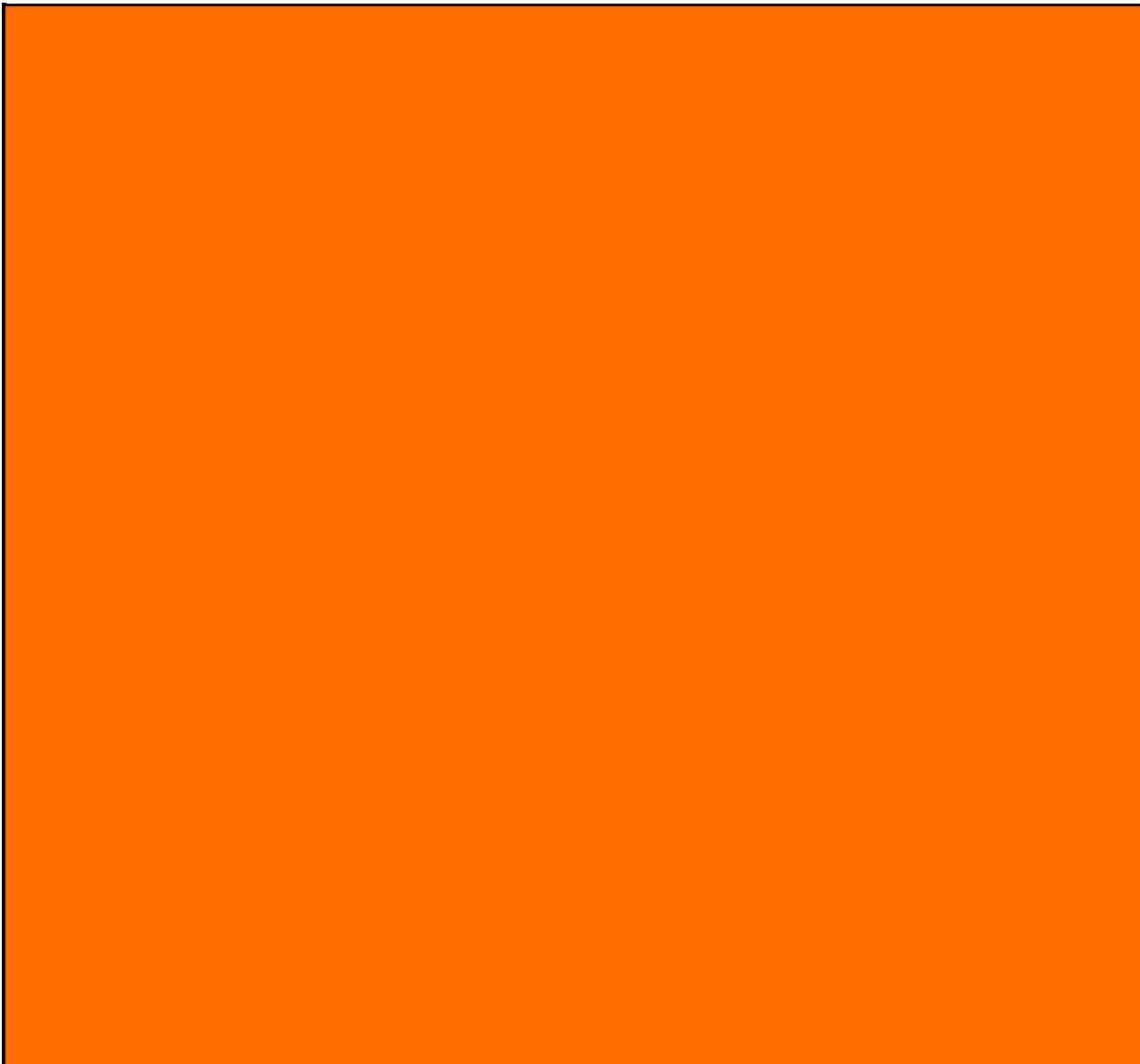
Grupo ROBOTICA EDUCATIVA EN ARAGÓN

Tenemos un grupo en Telegram de profesorado interesado en la Robótica Educativa en Aragón, si estás interesado en unirte, envía un mensaje por Whatsapp o Telegram a CATEDU 623197587 y te enviaremos un enlace.



4.3 Muro

Pon si quieres poner ejemplos tuyos o de otros interesantes:



■ ■

Créditos

Autoría

- Javier Quintana Peiró

Colaboradores:

- _ en _

Cualquier observación o detección de error por favor aquí soporte.catedu.es

Los contenidos se distribuyen bajo licencia Creative Commons tipo BY-NC-SA.

