

# 3 Programación en código

- Entorno de programación
- Maqueta
- APERTURA DE PUERTA
- JOYSTICK
- ALARMA AGUA
- FUEGO
- LÁSER
- BLUETOOTH APP->ARDUINO
- BLUETOOTH APP<->ARDUINO

# Entorno de programación

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

OJO, existe **la versión online** del editor <https://create.arduino.cc/editor>.

Es una buena solución si trabajas en varios equipos y quieres que tus proyectos estén disponibles en cualquier equipo.

**ATENCIÓN** para usar la versión online, tienes que instalar en tu ordenador el software **AGENT**

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

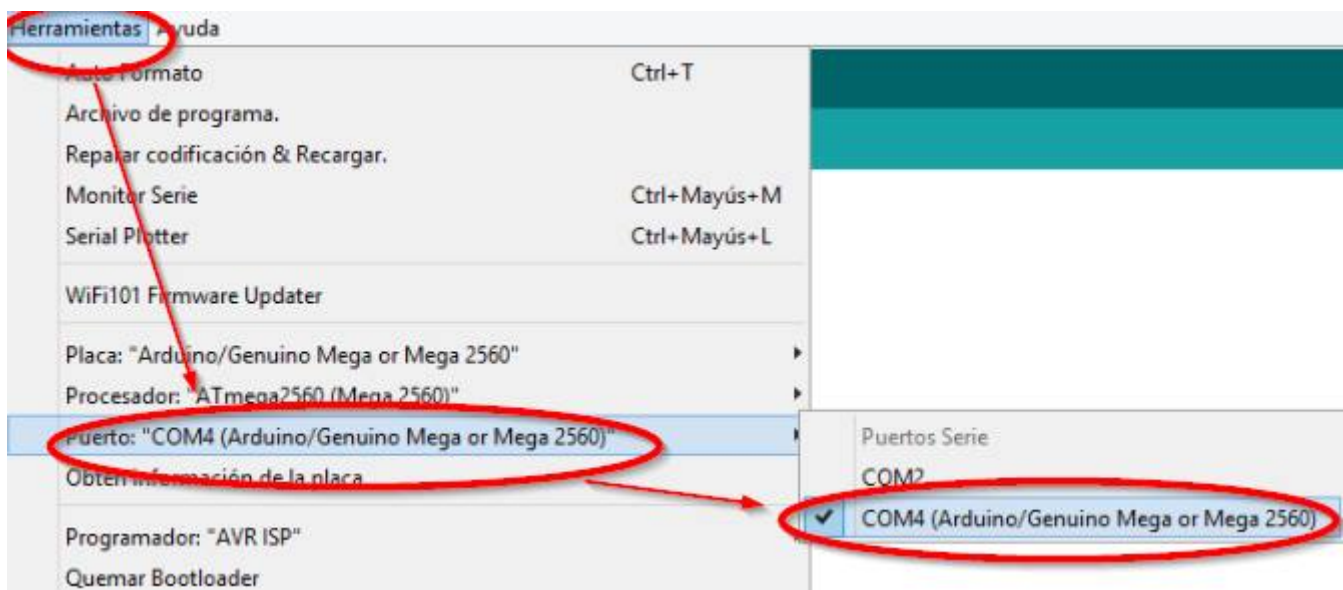
En Linux puede salir este mensaje "can't open device "/dev/ttyUSB0": Permission denied" donde 0 puede ser otro número, la solución [aquí](#)

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.

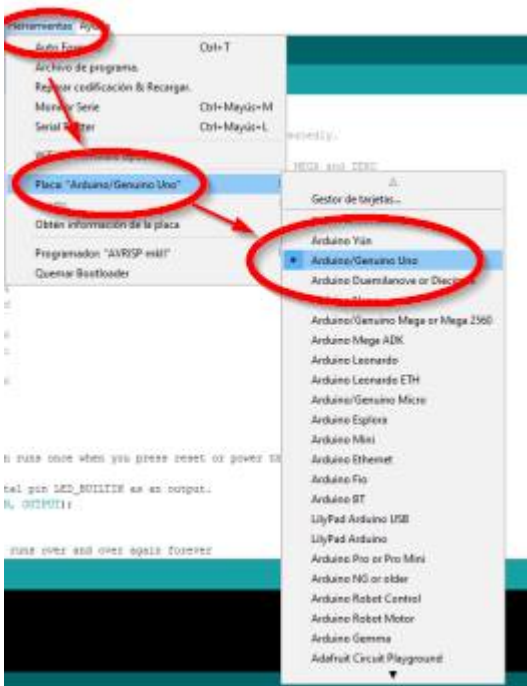
Arduino utiliza para escribir el código fuente o programa de aplicación lo que denomina "sketch" (programa). Estos programas son escritos en el editor de texto. Existe la posibilidad de cortar/pegar y buscar/remplazar texto.



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 pinchemos 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.

Knob

```
// Controlling a servo position using a potentiometer (variable resistor)
// by Michal Rinott <http://people.interaction-ivrea.it/m.rinott>

#include <Servo.h>

Servo myservo; // create servo object to control a servo

int potpin = 0; // analog pin used to connect the potentiometer
int val; // variable to read the value from the analog pin

void setup()
{
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
}

void loop()
{
  val = analogRead(potpin);
  val = map(val, 0, 1023, 0, 179);
  myservo.write(val);
  delay(15);
}
```

Descripción del programa

Módulos y declaración de variables

Bloque de configuración

Bloque de ejecución continua

```
// reads the value of the potentiometer (value between 0 and 1023)
// scale it to use it with the servo (value between 0 and 180)
// sets the servo position according to the scaled value
// waits for the servo to get there
```

## ¿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.

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](http://www.arduino.cc) (portal en inglés, más actualizada).
- [www.arduino.es](http://www.arduino.es) (portal en español).

# 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 la vez.

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



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

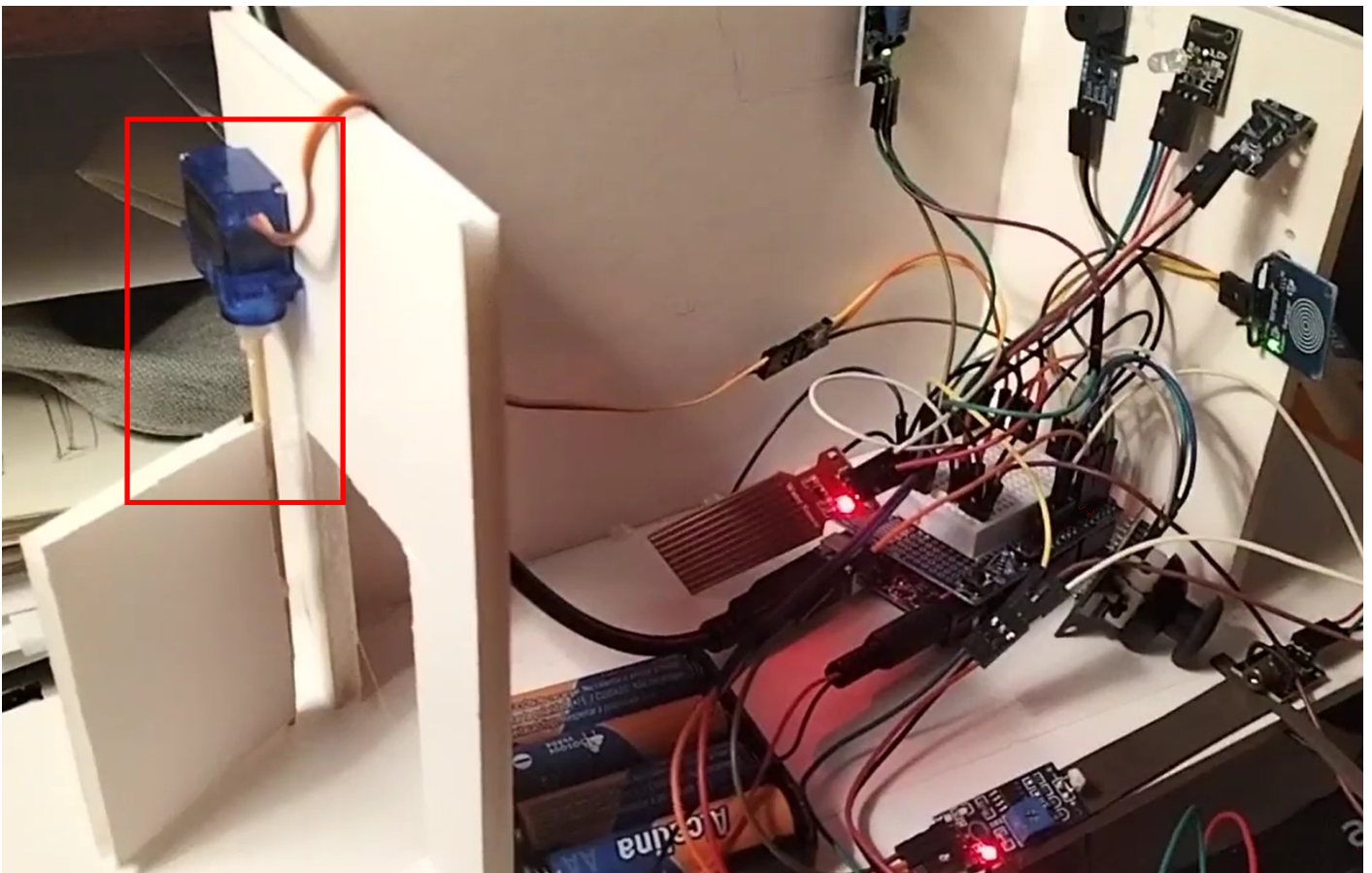


El soporte del servomotor con la puerta es difícil, una opción es pegarlo con una pistola de pegamento.



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

Otra opción para la apertura de la puerta es usar un palillo :







# APERTURA DE PUERTA

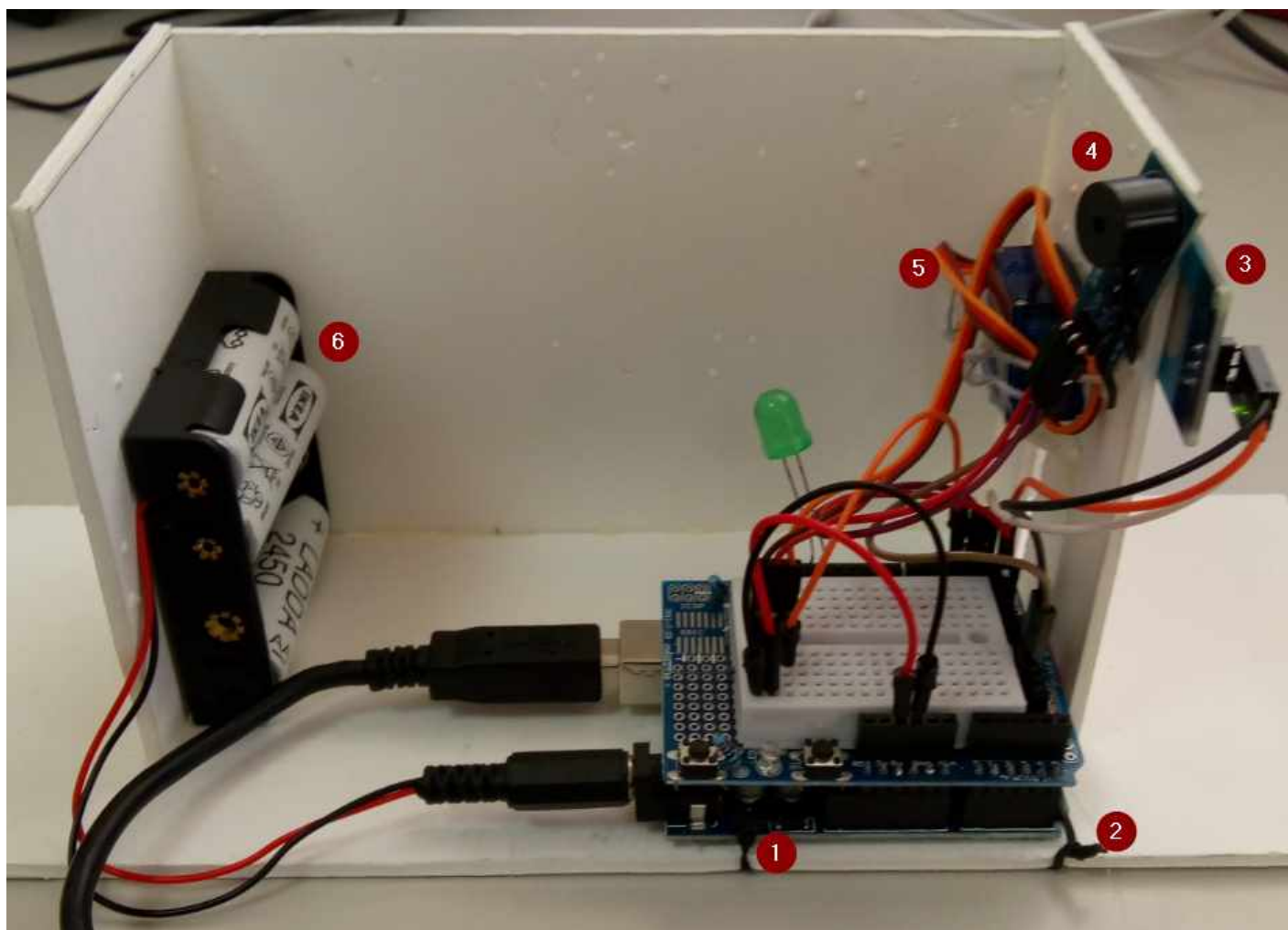
## Reto

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

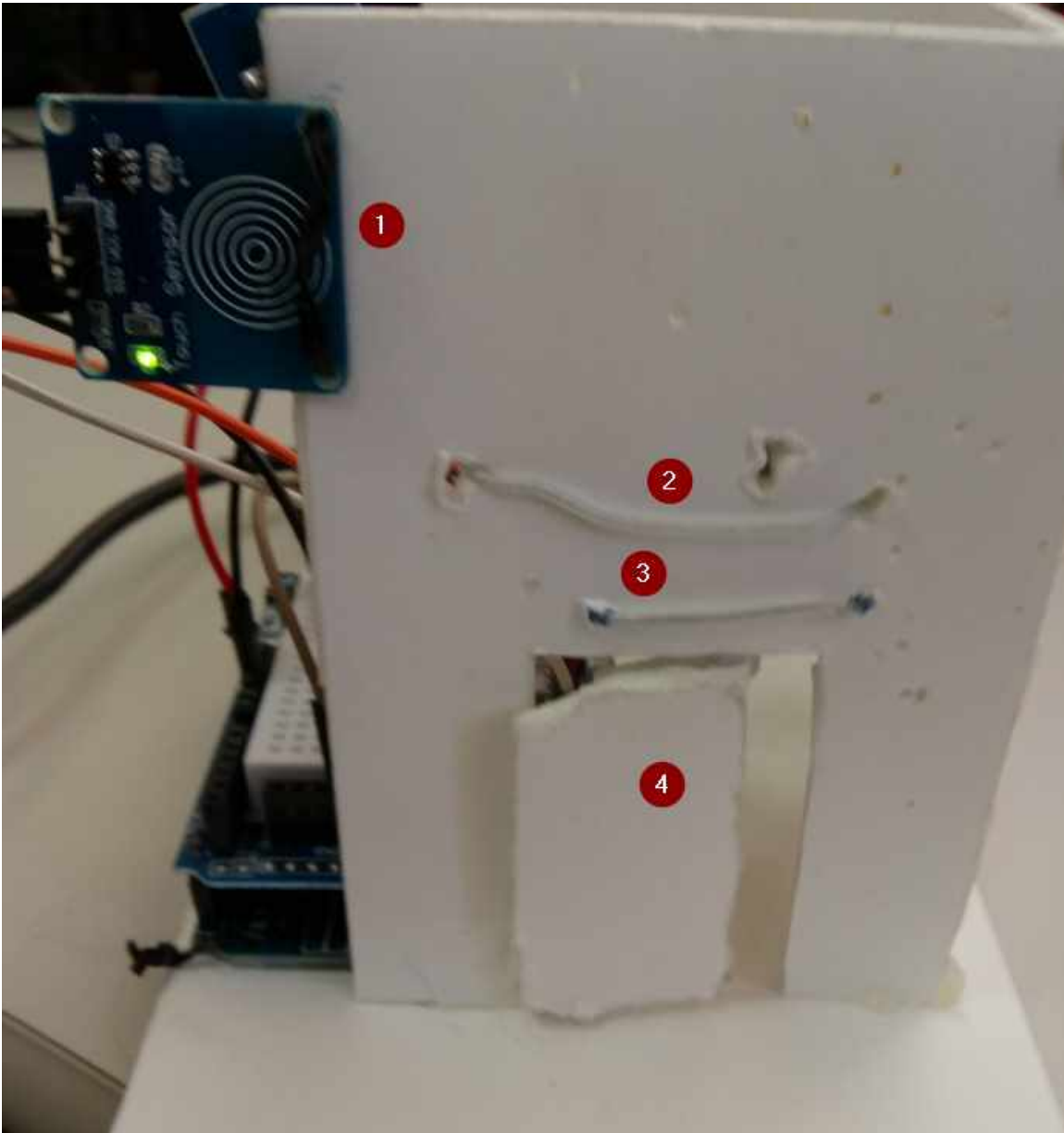
## Solución maqueta

Esta vez en la maqueta ponemos los siguientes elementos 1. Fijación placa Arduino 1. Fijación placa Arduino 1. Interruptor táctil 1. Buzzer 1. Servo 1. Portapilas



Detalle por delante

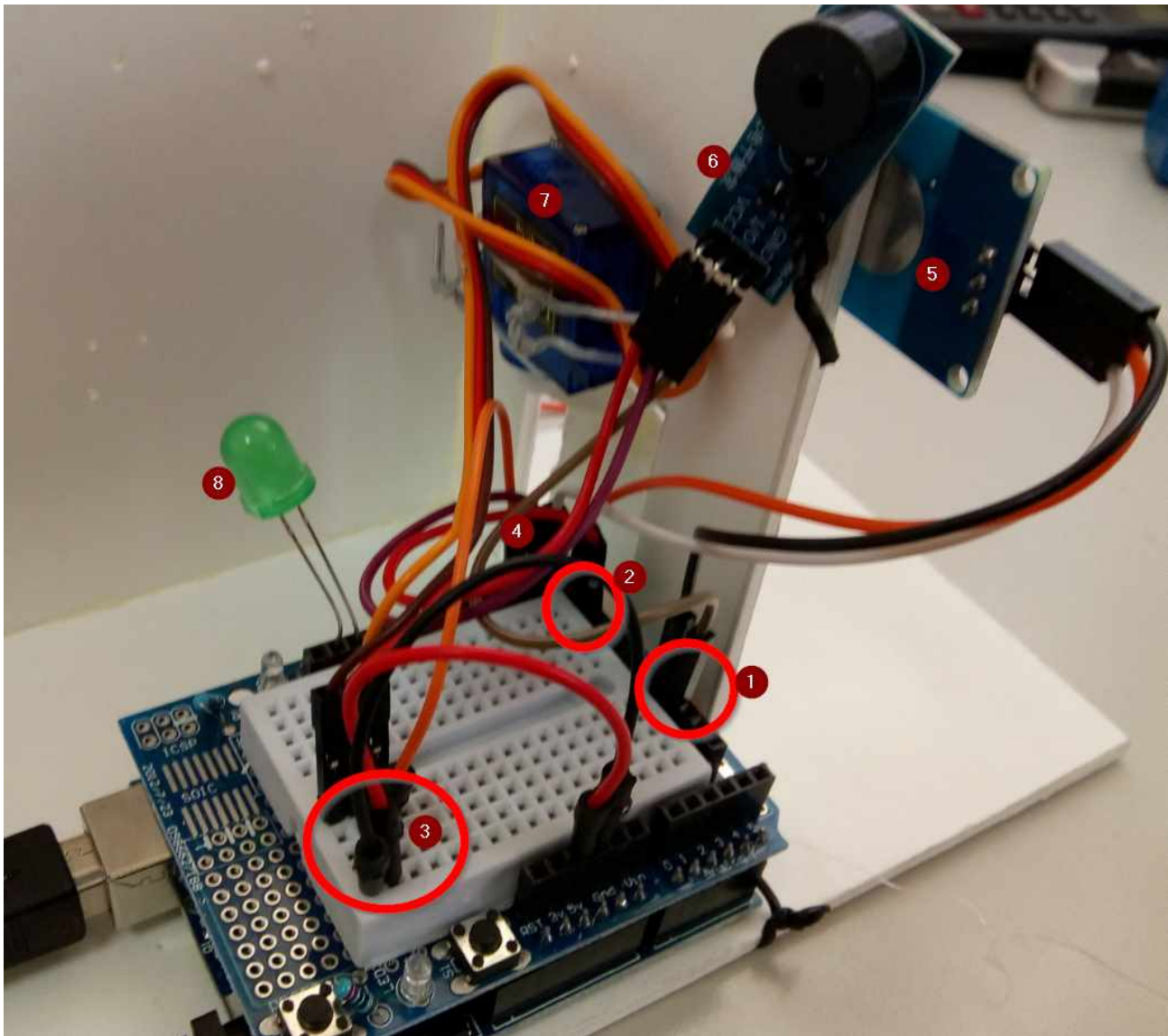
1. Interruptor táctil
2. Fijación servo
3. Fijación servo
4. Puerta \*si, ya sé, un poco chapuza ¿y qué pasa? \*



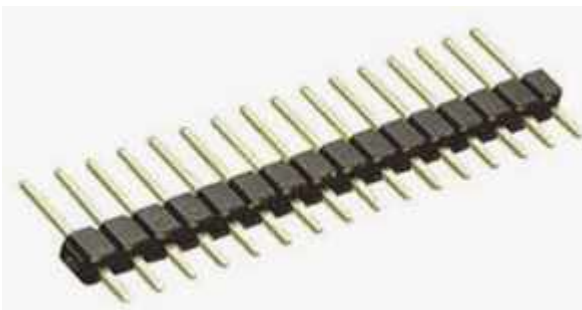
## Solución esquema de conexiones

Es el mismo mostrado en [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
  1. D2 Interruptor táctil
  2. 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 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).

## Solución video

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

## Solución código

<https://create.arduino.cc/editor/javierquintana/f89a2537-117b-4140-806f-5c79ef4f684a/preview>

<https://create.arduino.cc/editor/javierquintana/f89a2537-117b-4140-806f-5c79ef4f684a/preview?embed>

# JOYSTICK

## Reto

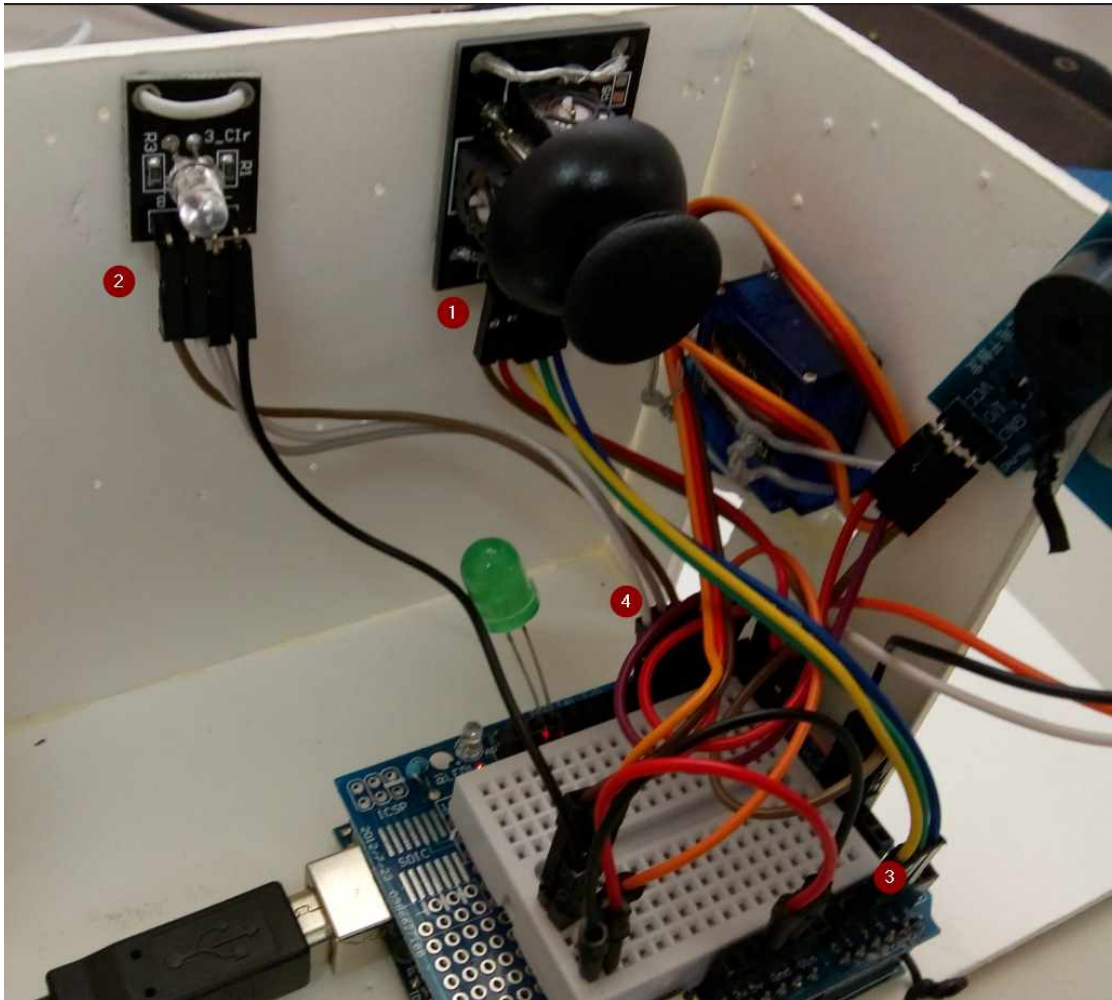
Ahora vamos a utilizar el Joystick para dos intenciones:

- 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 en rojo
- Un valor intermedio es un valor de los dos

## Solución 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



## Solución esquema eléctrico

Es igual que cuando vimos Joystick:

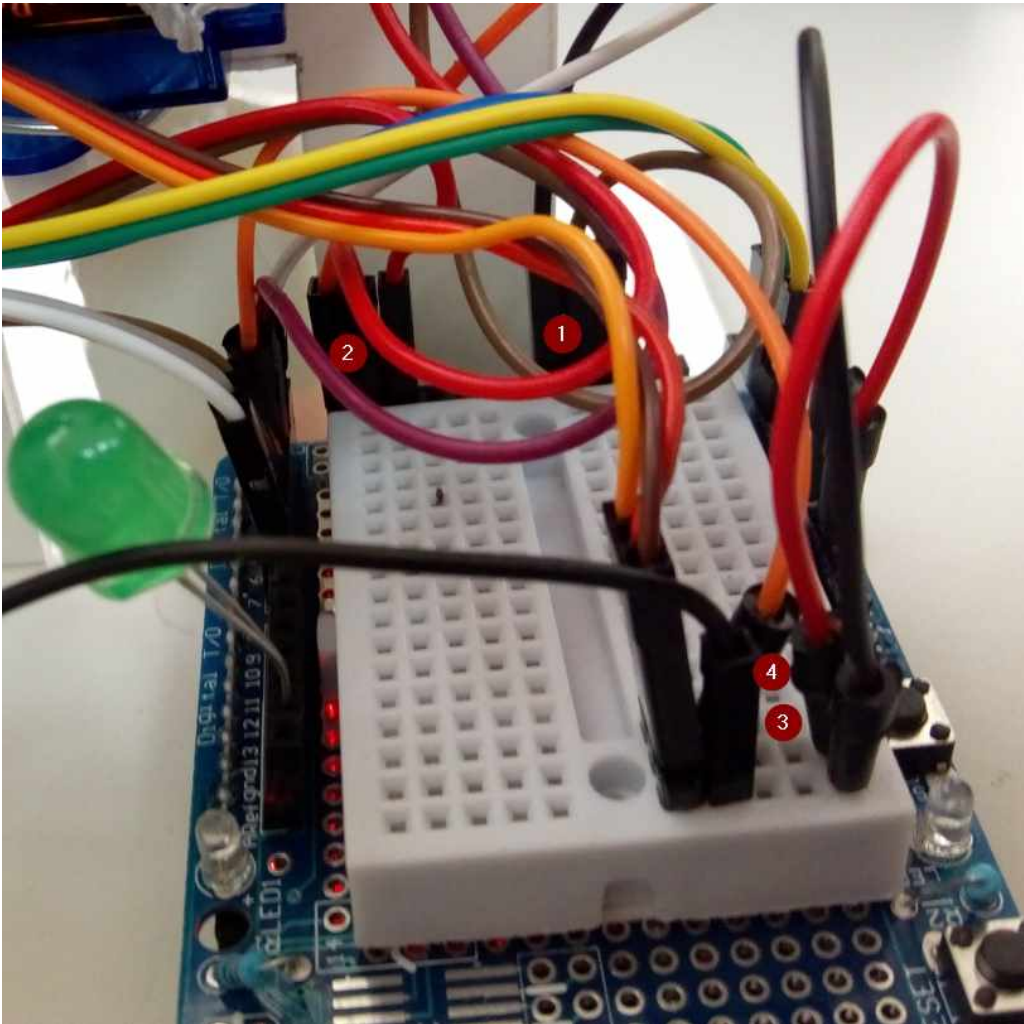
1. Terminales GND del led RGB y del Joystick
2. Terminal +5V del Joystick
3. Otra opción de conectar el terminal GND
4. Otra opción de conectar +5V

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 microrruptor del Joystick es digital luego lo apropiado es conectarlo en una entrada salida digital, pero como vamos escasos de entradas y salidas digitales (ya verés como esto se complica), lo trataremos como analógico.





## Solución vídeo

<https://www.youtube.com/embed/Cgi4k5cM4I4>

## Solución código

<https://create.arduino.cc/editor/javierquintana/f4dfc995-5a42-4d45-a911-238f67b0c169/preview>

<https://create.arduino.cc/editor/javierquintana/f4dfc995-5a42-4d45-a911-238f67b0c169/preview?embed>



# ALARMA AGUA

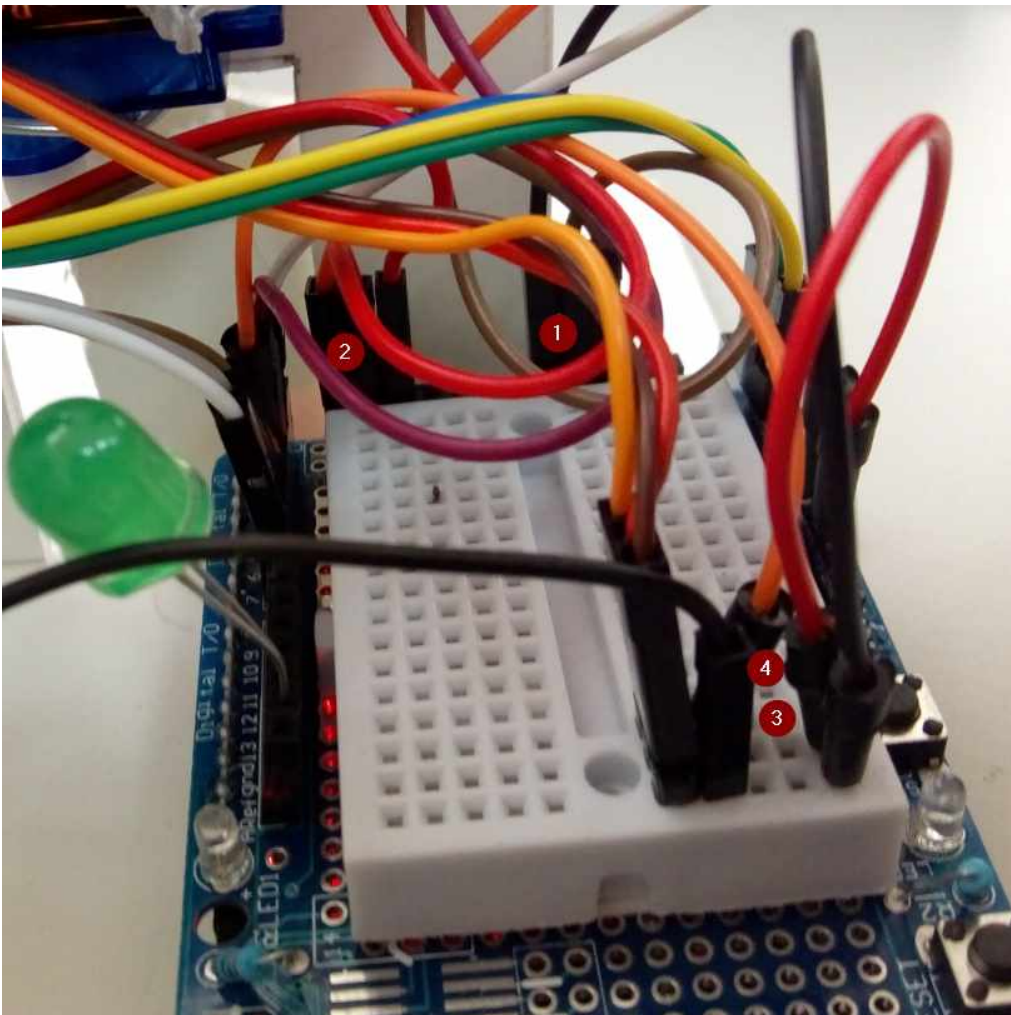
## Reto

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

## Solución esquema eléctrico

- Sensor de Agua en D8 igual que vimos en la parte de bloques
- Alimentación GND y +5V del sensor de agua o en 1 y 2 o en 3 y 4



## Solución vídeo

<https://www.youtube.com/embed/olf0gFJ20G>

## Solución código

<https://create.arduino.cc/editor/javierquintana/3eaca00b-a7b7-4357-a191-c5cfd33c004a/preview>

<https://create.arduino.cc/editor/javierquintana/3eaca00b-a7b7-4357-a191-c5cfd33c004a/preview?embed>

# FUEGO

**ATENCIÓN** hemos pensado la utilización del sensor de fuego sólo en la parte de CODIGO, pues es para nivel de secundaria.

**Implica la utilización de un mechero, con el peligro que conlleva. Utilizar este ejemplo en clase bajo responsabilidad y supervisión del docente.**

## Reto

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 actividad, vigilando y se produce una detección, la alarma tiene que sonar hasta que la desactivamos con una pulsación larga del pulsador.

## Solución 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

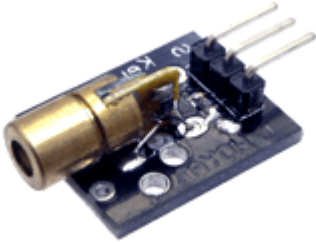
Las alimentaciones de los dos sensores igual que en las páginas anteriores.





# LÁSER

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



Si quieres 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.

## Reto

- 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 se apaga.

## Solución vídeo

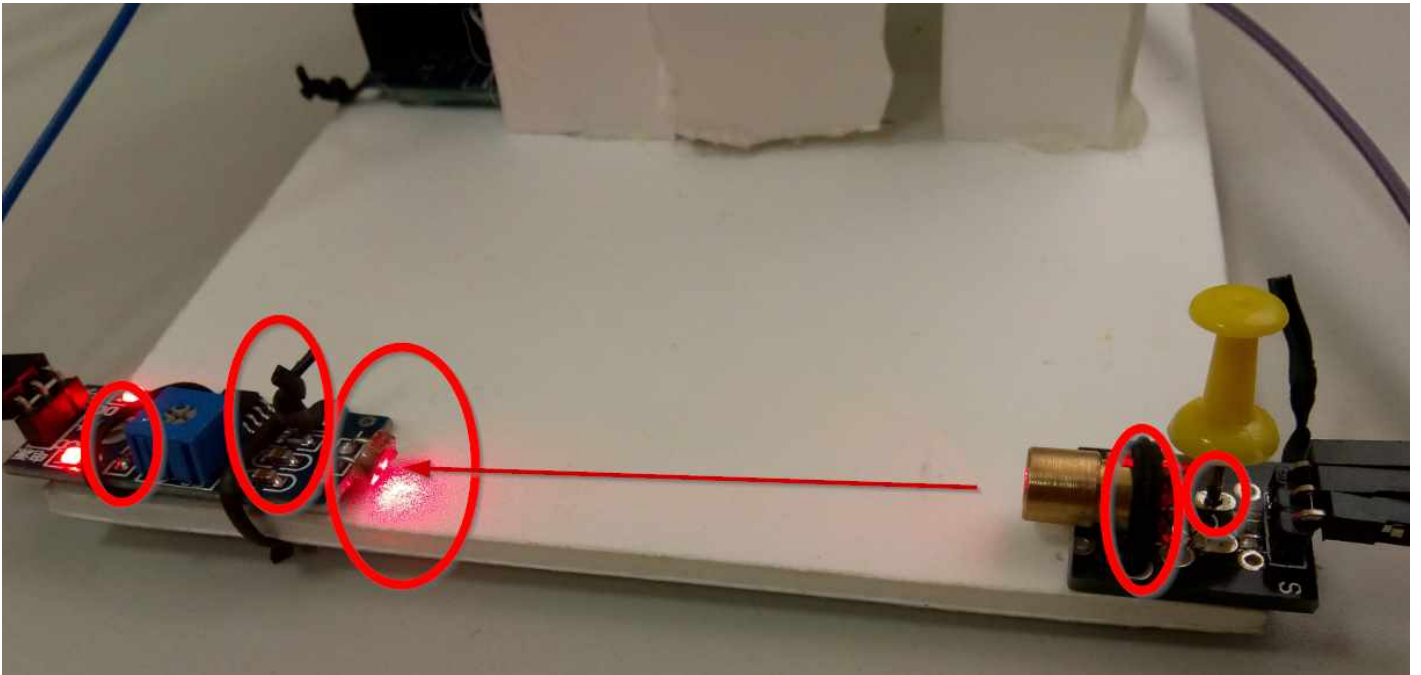
[https://www.youtube.com/embed/pmb\\_weSjZ04](https://www.youtube.com/embed/pmb_weSjZ04)

## Solución 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
- Truco, si la maqueta va a estar fija, una buena gota con la pistola de pegamento es mano de santo para que se quede fijo



## Solución 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

## Solución código

Aquí lo tienes

<https://create.arduino.cc/editor/javierquintana/dfca79db-6299-47c1-beef-7db3350c2cee/preview>

<https://create.arduino.cc/editor/javierquintana/dfca79db-6299-47c1-beef-7db3350c2cee/preview?embed>

# BLUETOOTH APP->ARDUINO

## 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 instalar
- Vincular tu móvil con el HC-06 con tu móvil

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

Configuración avanzada pues **nosotros somos pitos, y 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.

El RX del HC-06 en 11 y el TX del HC-06 en el 12

## RETO BÁSICO: Apertura de puerta y encendido láser.

### Enunciado reto básico

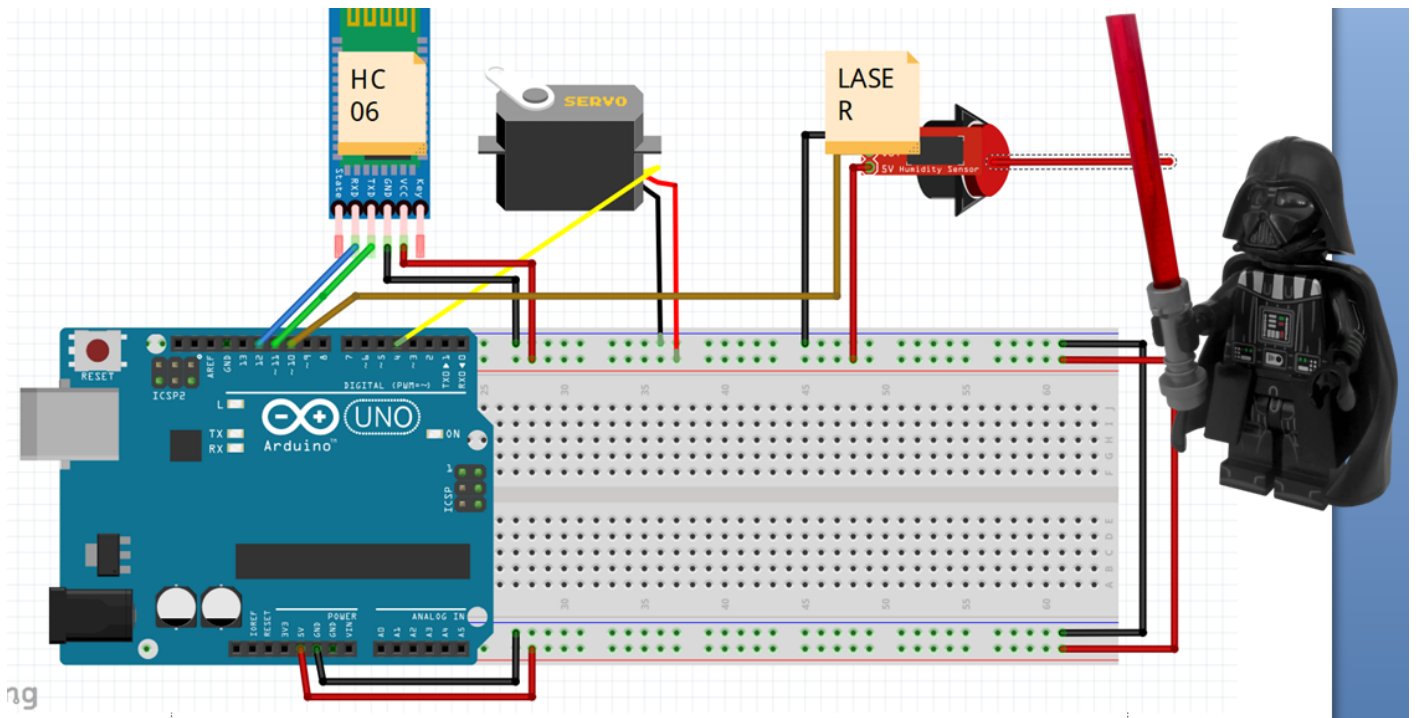
Vamos a realizar :

- Que desde la APP ->Placa
  - Con el comando A abre la puerta
  - Con el comando R que cierre la puerta
  - Con el comando L que se encienda o se apague el láser (es decir, si estaba encendido, que se apague, y viceversa)



## Conexiones reto básico

- Bluetooth HC-06 RX al 12 y TX al 11
- Puerta servo D4
- Láser al D10



El programa es este <https://create.arduino.cc/editor/javierquintana/88cfa21b-a86c-428c-95c4-a89a62a9218d/preview>

<https://create.arduino.cc/editor/javierquintana/88cfa21b-a86c-428c-95c4-a89a62a9218d/preview?embed>

## Reto básico Resultado

[https://www.youtube.com/shorts/vSjZBe\\_iElQ](https://www.youtube.com/shorts/vSjZBe_iElQ)

[https://www.youtube.com/embed/vSjZBe\\_iElQ](https://www.youtube.com/embed/vSjZBe_iElQ)

## Reto avanzado

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:

## Solución 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

## Solución vídeo

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

## Solución código

<https://create.arduino.cc/editor/javierquintana/e14ca107-67ba-4508-a5a0-6012213b4c1d/preview>

<https://create.arduino.cc/editor/javierquintana/e14ca107-67ba-4508-a5a0-6012213b4c1d/preview?embed>

# BLUETOOTH APP<->ARDUINO

## Nos preguntamos...

P: ¿Y pasar datos de nuestra maqueta a la APP ?

R: Se puede pero tendremos que hacer la APP a medida

P: ¿Podríamos tener libertad de poner botones?

R: Por supuesto,

Recuerda que con la APP que existen en GOOGLE PLAY como la utilizada en el capítulo anterior <https://libros.catedu.es/books/domotica-con-arduino/page/bluetooth> la comunicación es PLACA<-APP

## RETO BÁSICO

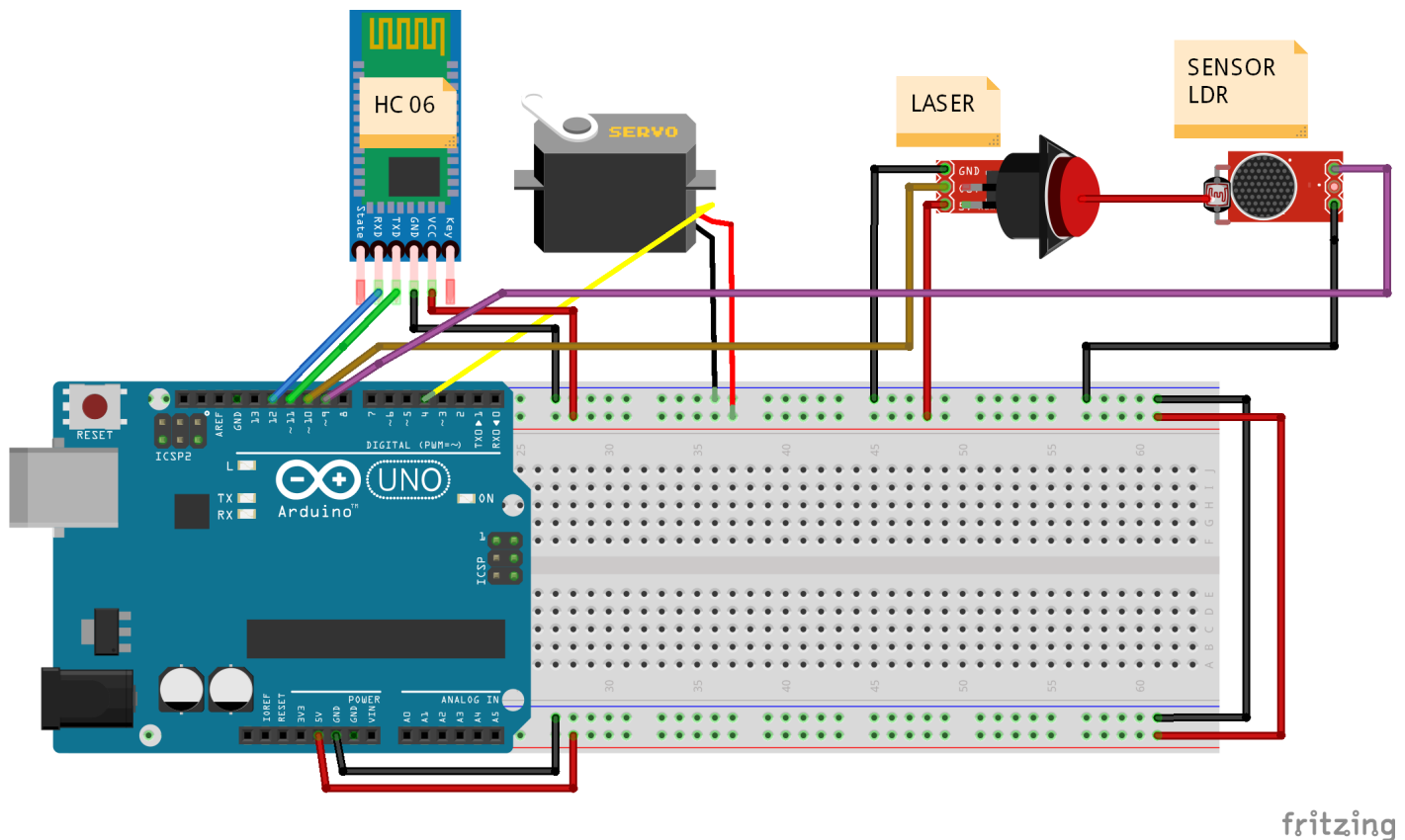
### Enunciado reto básico

Vamos a realizar :

- Que desde la APP ->Placa
  - Con el comando A abre la puerta
  - Con el comando R que cierre la puerta
  - Con el comando L que se encienda o se apague el láser (es decir, si estaba encendido, que se apague, y viceversa)
- Que desde la Placa ->APP
  - Si detecta ladrón que avise

### Conexiones reto básico

- Bluetooth HC-06 RX al 12 y TX al 11
- Puerta servo D4
- Láser D10 apuntando al LDR
- Sensor LDR D9



## RETO BÁSICO LA APP EN APP INVENTOR

los pasos son parecidos a los tratados en <https://libros.catedu.es/books/arduino-en-el-aula/page/app-tdr-steam-app-inventor-blocks>

Lo tienes en <https://gallery.appinventor.mit.edu/?galleryid=286eb7c2-792d-436c-b7d0-3d9e1e0592aa>

Para llevar la APP al móvil mira <https://libros.catedu.es/books/arduino-en-el-aula/page/llevar-la-app-creada-en-app-inventor-al-movil>

## RETO BÁSICO PROGRAMA EN EL ARDUINO CON CODIGO

El programa lo tienes aquí <https://create.arduino.cc/editor/javierquintana/ea8fa750-8cd1-48fd-8ea1-65bac09b78ce/preview>

<https://create.arduino.cc/editor/javierquintana/ea8fa750-8cd1-48fd-8ea1-65bac09b78ce/preview?embed>

**BUG:** Cuando el laser pasa de APAGADO a ENCENDIDO, el LDR es más lento que el programa Arduino y detecta LADRÓN luego desaparece, pero ya se ha quedado el fondo de pantalla rojo.

**RETO:** Habría que poner un pequeño retardo delay(500); para solucionar este problema ¿Dónde?

## Resultado video

[https://www.youtube.com/shorts/RZ\\_0EX4Oull?feature=share](https://www.youtube.com/shorts/RZ_0EX4Oull?feature=share)

[https://www.youtube.com/embed/RZ\\_0EX4Oull](https://www.youtube.com/embed/RZ_0EX4Oull)

# RETO AVANZADO

Como en la página anterior

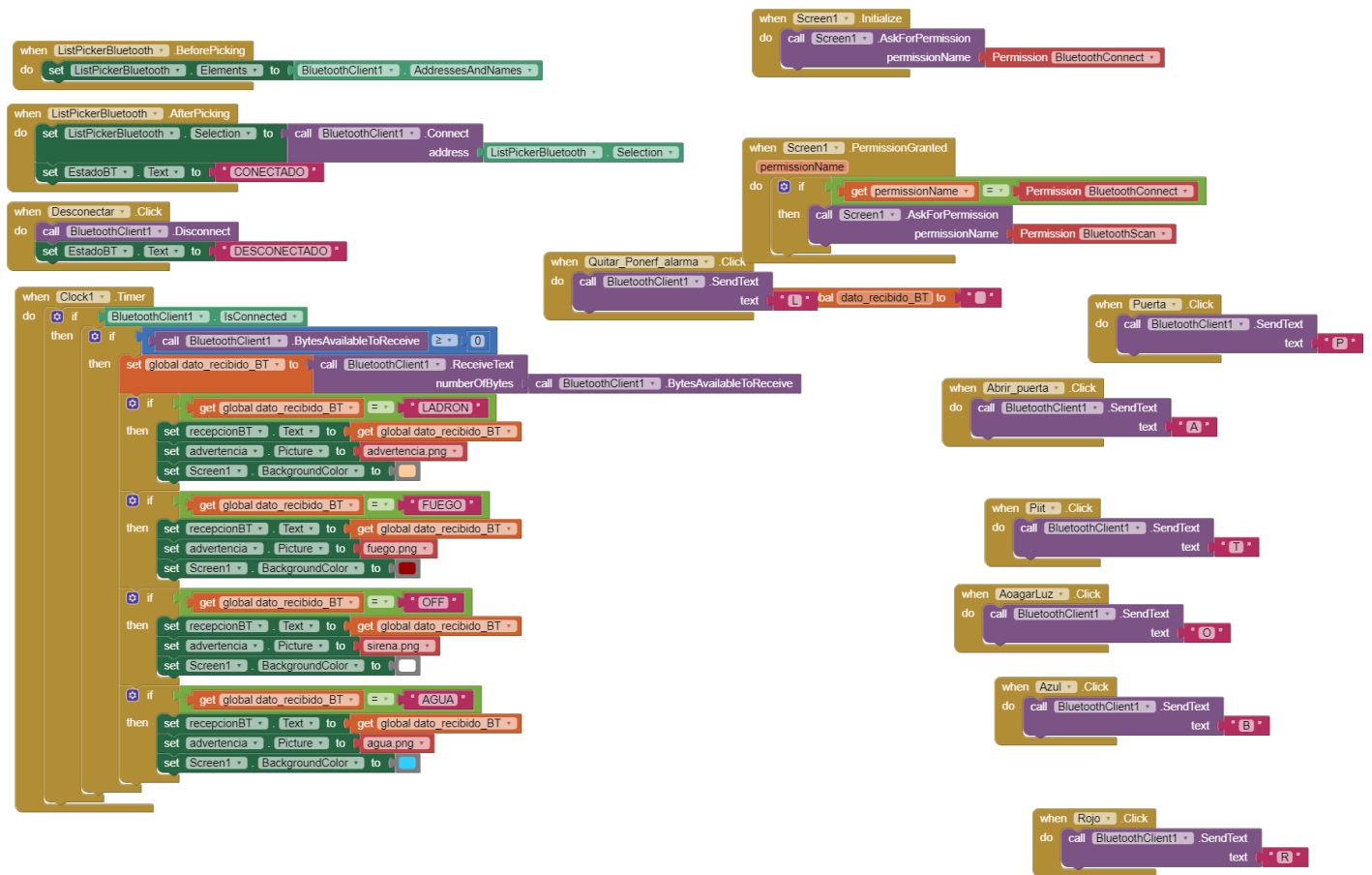
## RETO AVANZADO LA APP EN APP INVENTOR

los pasos son parecidos a los tratados en <https://libros.catedu.es/books/arduino-en-el-aula/page/app-tdr-steam-app-inventor-blocks>

Pero en la vista de **Designer**, hemos añadido más botones, imagen, etc...



Y la vista de **Blocks** es más completa



El programa lo tienes aquí :

<https://gallery.appinventor.mit.edu/?galleryid=b4979892-5df3-4225-90be-538a64d469e0>

## RETO AVANZADO EL PROGRAMA EN EL ARDUINO

Y hemos modificado el programa en Arduino para que envíe los comandos por bluetooth

Busca los comandos **Serial1.print(...);**

<https://create.arduino.cc/editor/javierquintana/b2c39d6f-d0ef-44e2-bb5b-2d9351dddb94/preview>

<https://create.arduino.cc/editor/javierquintana/b2c39d6f-d0ef-44e2-bb5b-2d9351dddb94/preview?embed>

## RETO AVANZADO RESULTADO

<https://youtu.be/gtuJzYvTBJM>

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