

PRACTICAS AVANZADO

- Utilización de varios sensores y activadores a la vez
- Programación en funciones y uso de variables
- Continuación de lectura de sensores por el puerto serie

- [Alarma láser](#)
- [Piano con teclado invisible](#)
- [Aparca coches](#)
- [Estación meteorológica](#)

Alarma láser

Objetivo

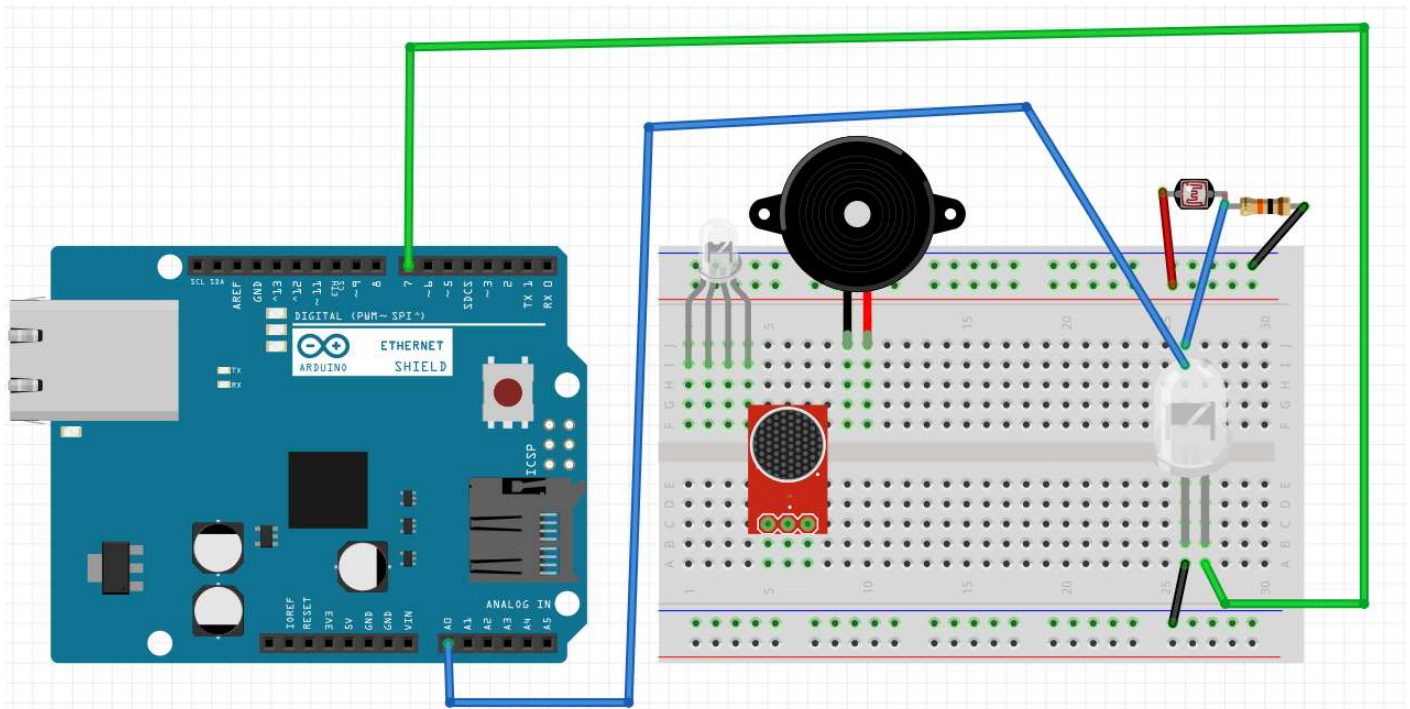
Realizaremos una alarma laser con las siguientes condiciones:

- Al tocar el pulsador táctil se **activará** la alarma, es decir, se encenderá el láser que apunta al sensor LDR
- Si la alarma está activada y "el ladrón" corta el haz láser, entonces la alarma **se dispara**
 - el disparo consistirá en avisos acústicos y visuales
- para salir del estado de disparo o de activación se pulsará el botón **reset** (no es necesario ninguna programación especial para cumplir este ítem)

https://www.youtube.com/embed/_dQcLtzTNQ

Cableado

Tenemos que ser cuidadosos de apuntar el láser (representado en esta figura por un led) al LDR, en caso contrario el Arduino se disparará de inmediato al cargar el programa.



Programa

Se aconseja el uso de funciones y variables para simplificar la programación :

OJO si se dispara la alarma, pon más retardo de 500ms

<http://www.arduinoblocks.com/web/project/772739>

Inicializar

```

Establecer VIGILANCIA = falso
Establecer DISPARO-ALARMA = falso
Escribir digital Pin 7 OFF

```

Bucle

```

+ si Pulsador táctil Pin 2
hacer Establecer VIGILANCIA = verdadero
+ si VIGILANCIA
hacer Escribir digital Pin 7 ON
Esperar 500 milisegundos
+ si Nivel de luz (LDR) Pin A0 % < 95
hacer Establecer DISPARO-ALARMA = verdadero
repetir mientras DISPARO-ALARMA
hacer DISPARA ALARMA

```

+ para DISPARA ALARMA

```

Led RGB Cátodo común Pin R 9 Pin G 10 Pin B 11 R 255 G 255 B 255
Zumbador Pin 3 Ms 255 Hz Tono (Hz) SI
Led RGB Cátodo común Pin R 9 Pin G 10 Pin B 11 R 000 G 000 B 000
Zumbador Pin 3 Ms 255 Hz Tono (Hz) RE

```

Mejora esta alarma añadiendo otro sensor, el sensor de Ultrasonidos, si detecta un objeto cercano que se dispare la alarma.

Otra mejora:: Añade un retardo al disparo de la alarma al activar uno de los sensores, por ejemplo el láser. Igual que en los casos reales, el sensor de la puerta principal tiene retardo para que el "dueño" tenga un tiempo para *desactivar* la alarma antes de que se dispare. El otro sensor (el de ultrasonidos) no tendría retardo, estaría por ejemplo en una habitación por lo tanto no tiene que tener retardo, el dueño nunca entra a la casa por ahí.

Piano con teclado invisible

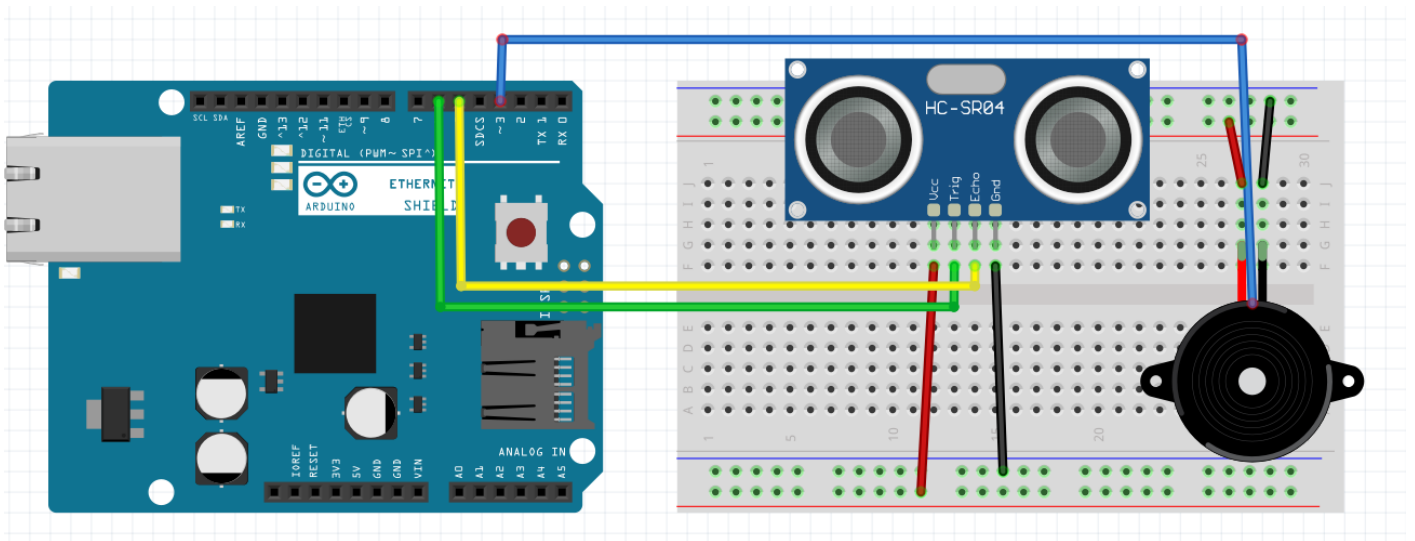
Objetivo

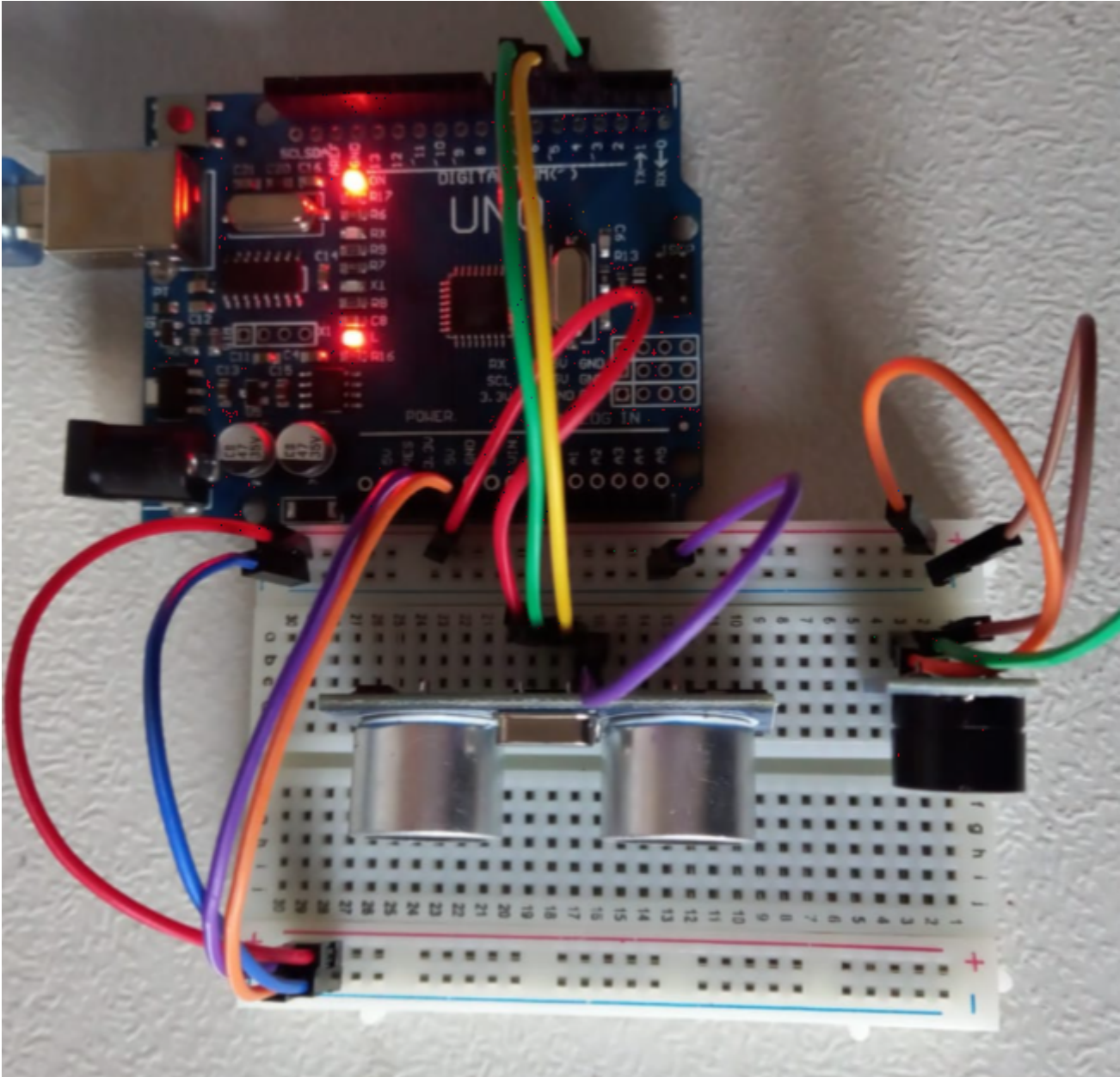
Vamos a tocar diferentes notas según la distancia que mida el sensor de ultrasonidos.. Cuanto más cerca, más agudo.

<https://www.youtube.com/embed/8hzQUQSXN68>

Cableado

Conectaremos el sensor de ultrasonidos y el zumbador tal y como lo hemos hecho en anteriores prácticas.





Programa

<http://www.arduinoblocks.com/web/project/780618>

Inicializar

Bucle

```
Establecer distancia = Distancia (cm) [Trigger] 6 [Echo] 5
+ si distancia = 0
hacer Establecer distancia = 9999
+ si distancia < 40
hacer Establecer frecuencia = 10000 x 1 ÷ distancia
Zumbador Pin 3 Ms 500 Hz frecuencia
```

Aparca coches

Objetivo

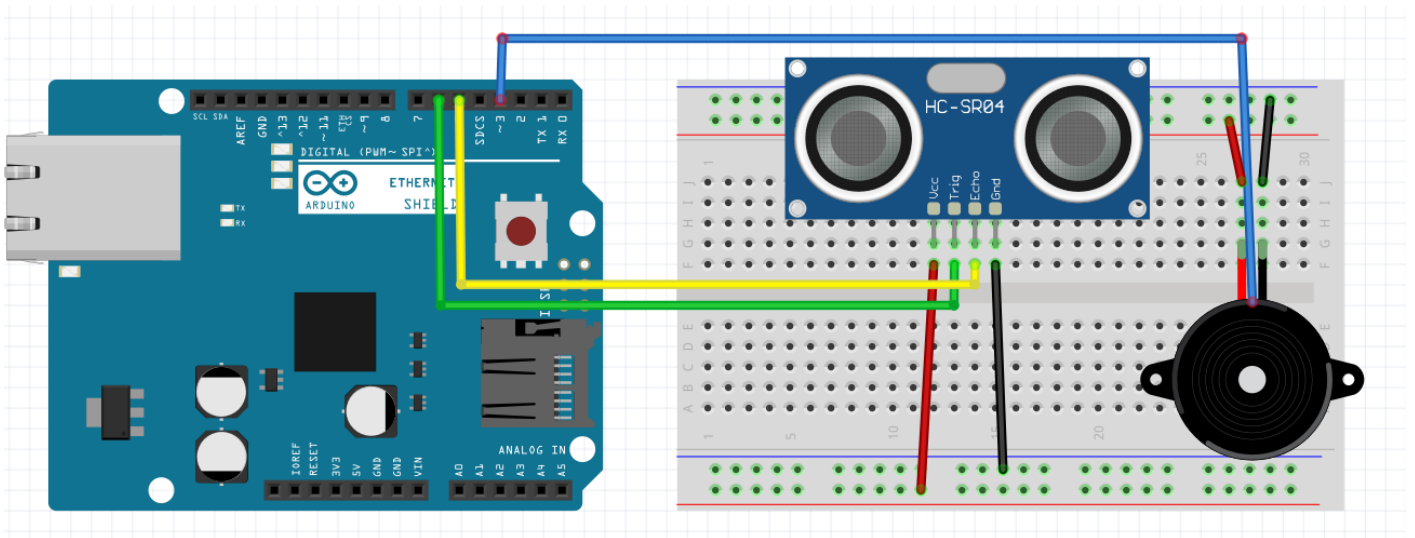
Simular el sensor asistente de aparcamiento que mediante una señal acústica nos avisa de la cercanía de un obstáculo.

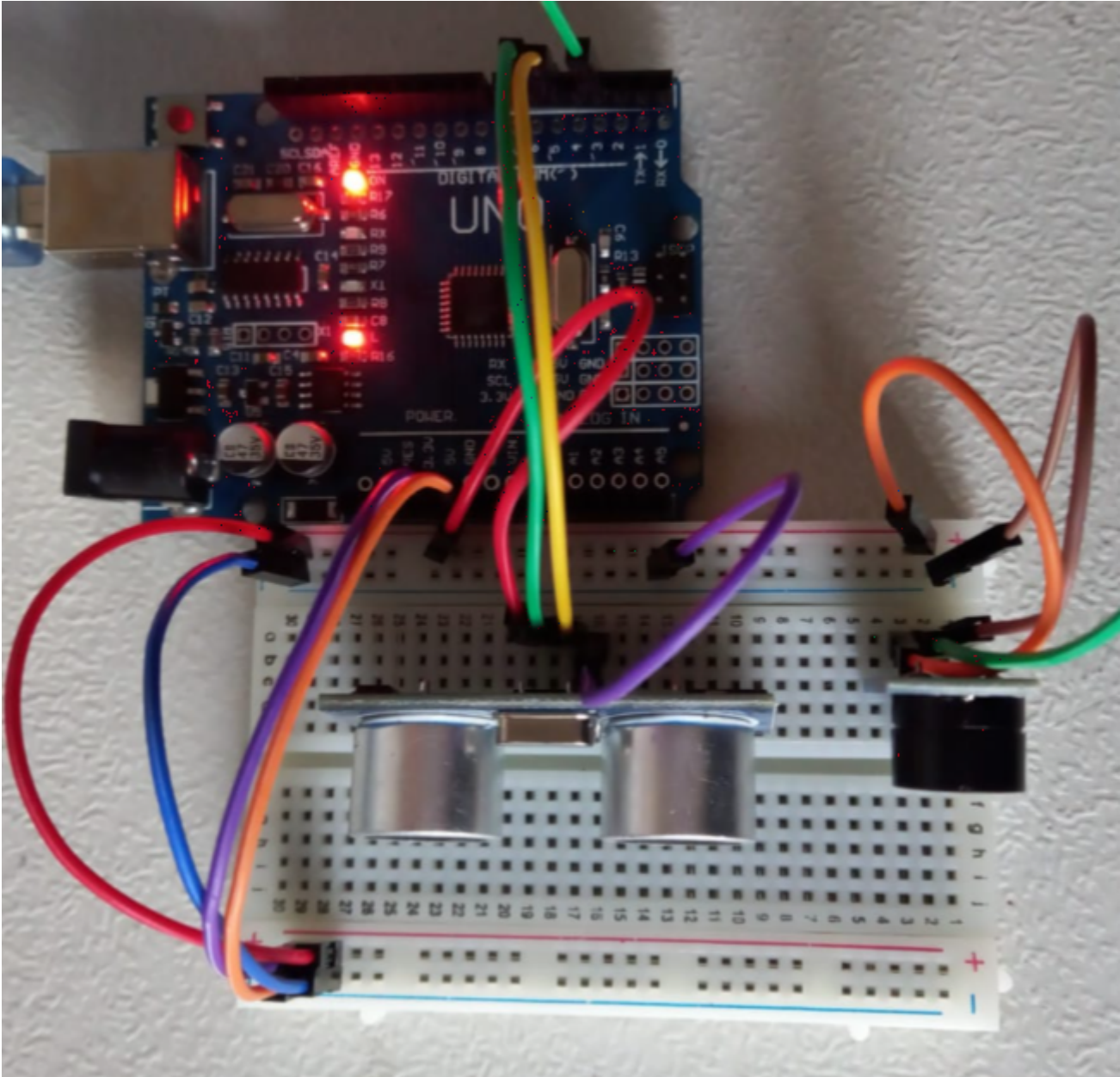
Es parecido a la práctica anterior, pero aquí el tono es el mismo. Lo que cambia es la frecuencia de los pulsos

<https://www.youtube.com/embed/2JfD6w3deZk>

Cableado

Es igual que en la práctica anterior





Programa

<http://www.arduinoblocks.com/web/project/780629>



```
graph TD
    subgraph Bucle [Bucle]
        A[Establecer distancia = Distancia (cm) [Trigger] 6 [Echo] 5]
        B[+ si distancia = 0]
        C[hacer Establecer distancia = 9999]
        D[+ si distancia < 60]
        E[hacer Establecer pausa = 10 x distancia]
        F[Esperar pausa milisegundos]
        G[Zumbador Pin 3 Ms 500 Hz 800]
        B --- C
        D --- E
        E --- F
        F --- G
    end
```

The image shows a Scratch-style block-based programming script for a distance sensor. It starts with a 'Bucle' (Loop) block. Inside the loop, the first block is 'Establecer distancia = Distancia (cm) [Trigger] 6 [Echo] 5'. This is followed by a conditional block '+ si distancia = 0' which contains a 'hacer Establecer distancia = 9999' block. Another conditional block '+ si distancia < 60' contains a 'hacer Establecer pausa = 10 x distancia' block, followed by an 'Esperar pausa milisegundos' block, and finally a 'Zumbador Pin 3 Ms 500 Hz 800' block. The number '60' in the second conditional block is highlighted with a yellow box.

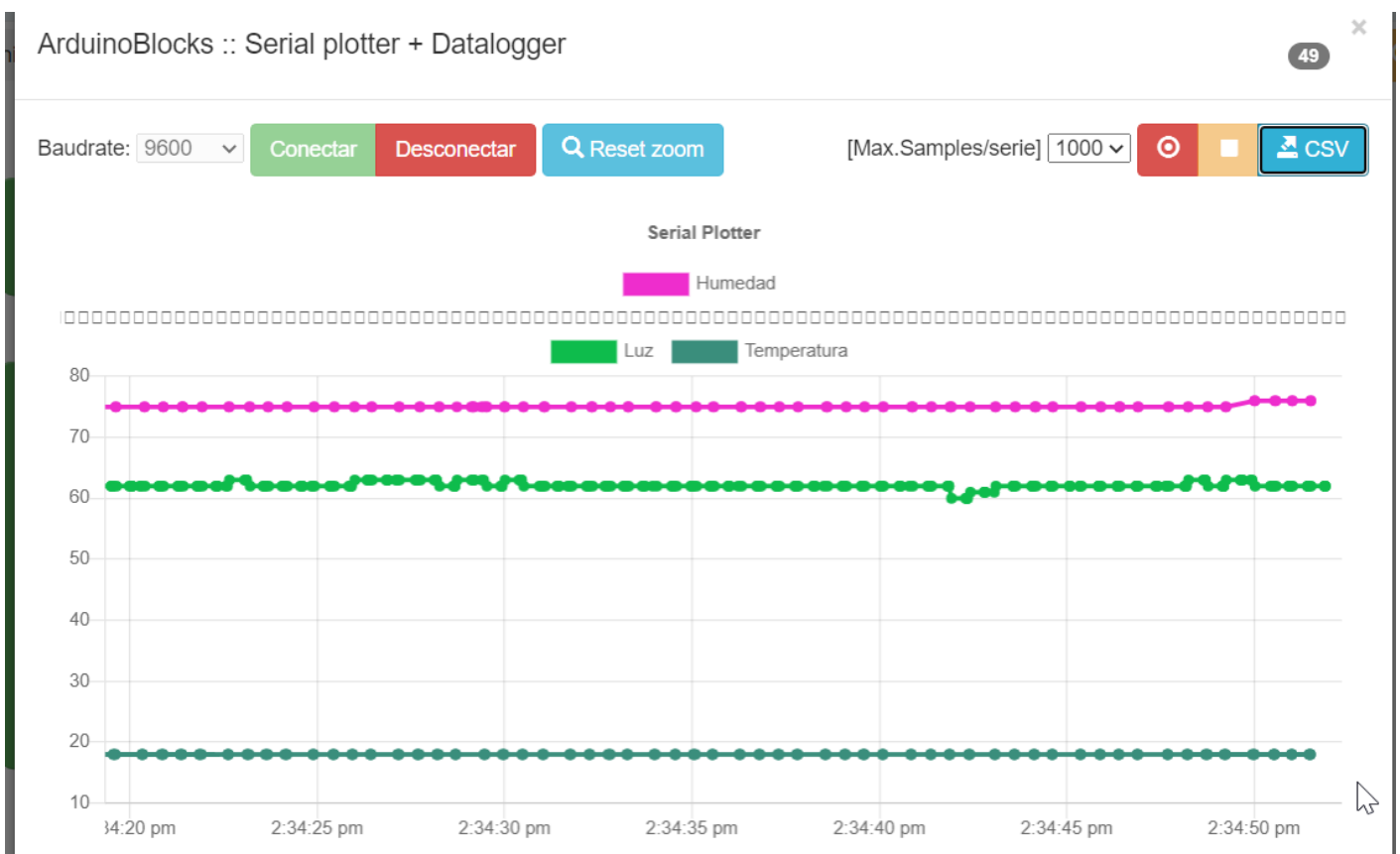
Estación meteorológica

Objetivo

Igual que en la [práctica de lectura de puertos](#) podemos visualizar los datos de sensores ambientales :

- DHT11 que mide en el mismo pin la humedad y la temperatura
- El sensor LDR que mediría la luminosidad, proporcional a la nubosidad existente.

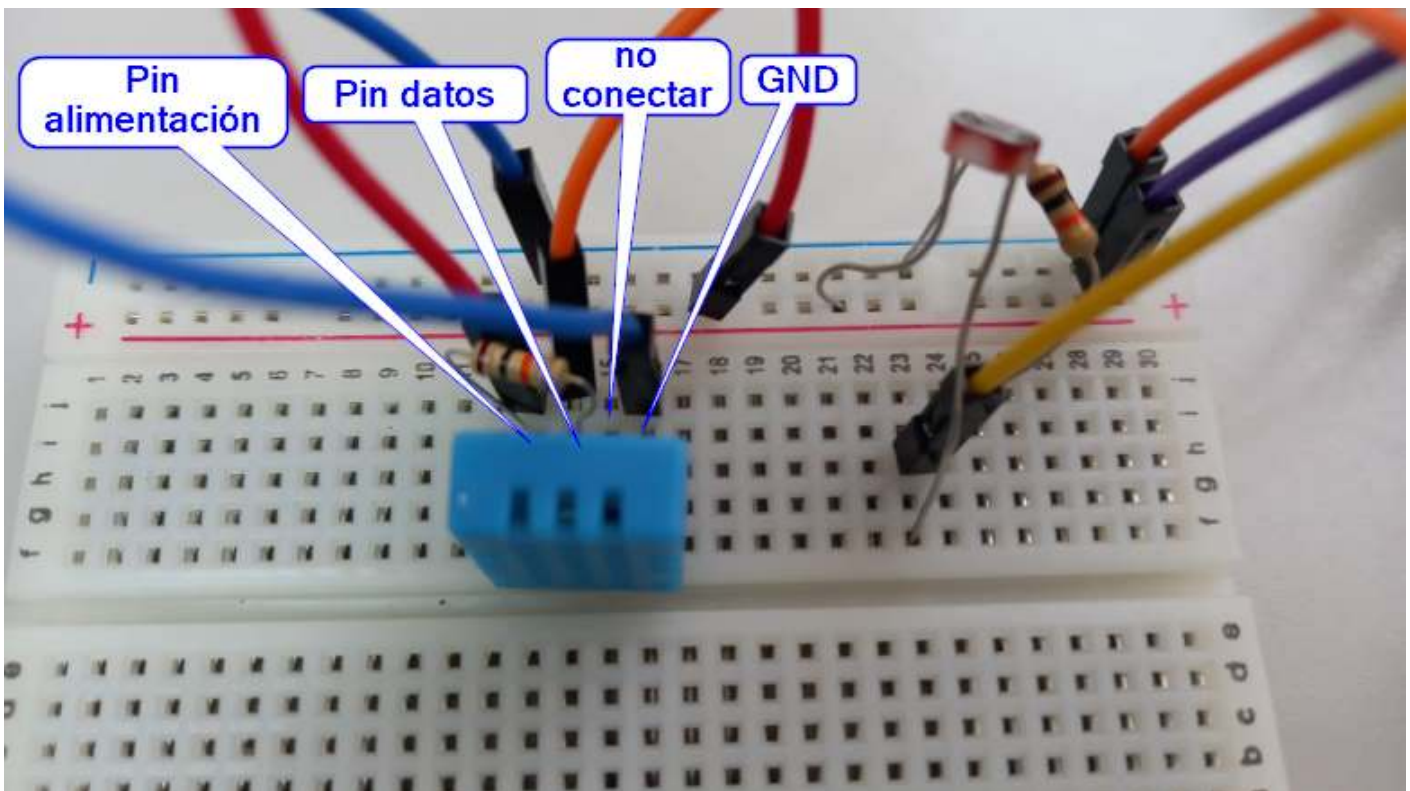
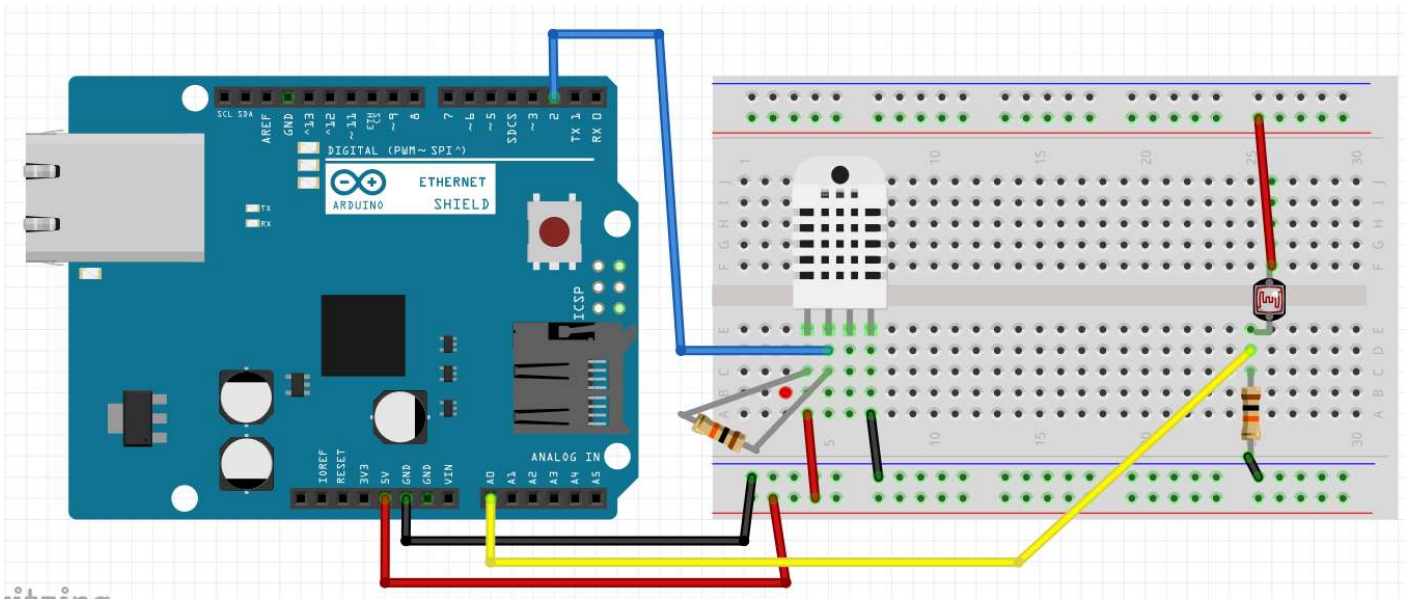
Podemos ver las variaciones lanzando vaho en el sensor y tapando el LDR.



Cableado

Vamos a practicar la lectura de sensor DHT11 que es sencillo de utilizar pues como todos, tiene dos pines de alimentación y un pin de lectura, el otro pin, el penúltimo **no se conecta** :

Ojo, para que funcione correctamente hay que poner una resistencia entre el pin de datos y el de alimentación



Programa

<http://www.arduinoblocks.com/web/project/780757>



Inicializar

>_ Iniciar Baudios 9600 ▾

Bucle

>_ Plotter “ Temperatura ” Valor DHT-11 Temperatura °C ▾ Pin 2 ▾

>_ Plotter “ Humedad ” Valor DHT-11 Humedad % ▾ Pin 2 ▾

>_ Plotter “ Luz ” Valor Nivel de luz (LDR) Pin A0 ▾ % ▾

Esperar 500 milisegundos