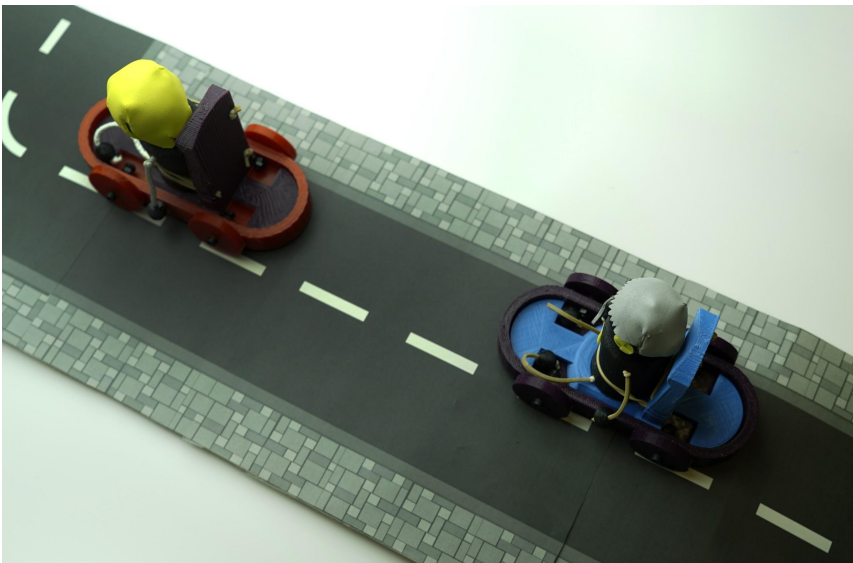


# Programa tu sistema de aviso de la distancia de seguridad.



*En este proyecto crearás un sistema que te permita alertar a los coches o ciclistas de que no están cumpliendo con la distancia de seguridad adecuada.*

**Nivel de dificultad:**  
*Fácil*

**Tiempo estimado:**  
*20 minutos*

## **Materiales:**

- 1 ordenador.
- Bitbloq.
- Placa controladora.
- 1 sensor de ultrasonidos.
- 1 zumbador.

## ¡En marcha!

Antes de empezar, es importante que conozcamos cómo funciona Bitbloq, y cómo usar y programar algunos componentes, como son, el zumbador (actuador) y el sensor de ultrasonidos (sensor).

Para ello, primero creamos dos programas sencillos para entender cómo funcionan. Para empezar, programaremos que suene el zumbador.

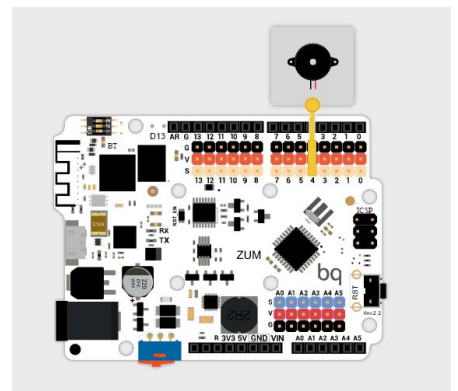
### ¿Qué es el zumbador?

El zumbador es un componente electrónico con el que podemos hacer sonidos. Si sabes usarlo, podrás programar canciones, o, como en este caso, alertas.

- **¿Cómo conectar el zumbador a la placa?, ¿y en Bitbloq?**

Antes de conectar un componente a la placa, deberemos pensar si es un componente **analógico**, aquellos que pueden tener más de dos valores; o **digital**, aquellos con sólo dos valores.

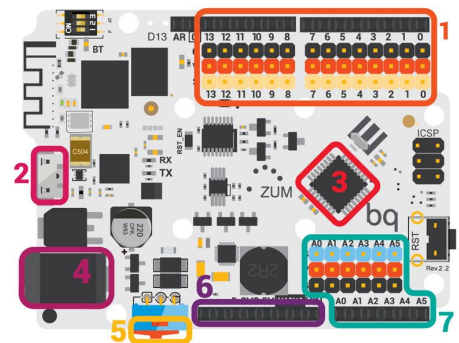
En nuestro caso, el zumbador es un componente digital ya que solo tiene dos estados, sonando (1) o en silencio (0). Por ello, deberemos conectarlos a los pines digitales de nuestra placa controladora.



- **¿Cuáles son los pines digitales de la placa controladora?**

Son los que corresponden con el número 1 de la siguiente imagen.

\* Debemos intentar no utilizar los pines digitales 0 y 1 de la placa, ya que a través de estos dos pines, la placa se comunica con el ordenador y otros aparatos por lo que en ocasiones pueden dar problemas.



Ahora que ya sabemos qué es y qué tipo de componente (digital), lo conectaremos a la placa controladora física. Y tras esto, deberemos ir a Bitbloq y en la pestaña *Hardware*, arrastrar nuestra placa y el zumbador, conectándolo al mismo pin que utilizamos en la placa física.

- **¿Cómo lo programo?**

Para hacer sonar el zumbador, deberemos poner en el *Bucle principal (Loop)* el siguiente bloque:

Sonar el zumbador   zumbador ▼   con la nota   Do ▼   durante   2000   ms

Como puedes ver en el bloque, tenemos que decirle al programa 3 cosas sobre el zumbador:

- El nombre del componente: en este caso, le hemos llamado “zumbador”, pero puedes llamarlo como quieras (alarma, alerta, Elena, Matías...).
- La nota que queremos que suene: así podrás construir tu canción.
- La duración: el tiempo que quieres que se quede sonando. Está en milisegundos (ms), para que te hagas una idea, 1000 ms son 1 segundo, 500 ms son medio segundo, etc. Así podrás elegir mucho mejor el tiempo para tus canciones y sonidos.

Una vez que tengamos el programa, conectamos la placa al ordenador y clicamos sobre el botón cargar. Tras cargarse el programa, el zumbador empezará a sonar. Si tienes problemas para sonar el zumbador, consulta el siguiente enlace: <http://diwo.bq.com/antes-de-empezar-con-bitbloq-2/>.

¡Ya tienes tu programa! Ahora puedes probar a complicar éste (o crear uno nuevo) poniendo más bloques con diferentes notas para hacer la escala musical o incluso una melodía.

Ya tienes dominado el zumbador, ahora vamos a por la otra parte que necesitamos para la alerta: el ultrasonidos.

## ¿Qué es el ultrasonidos?

El sensor de ultrasonidos es un componente que nos permite saber a qué distancia está un objeto.

- **¿Cómo funciona un sensor de ultrasonidos?**

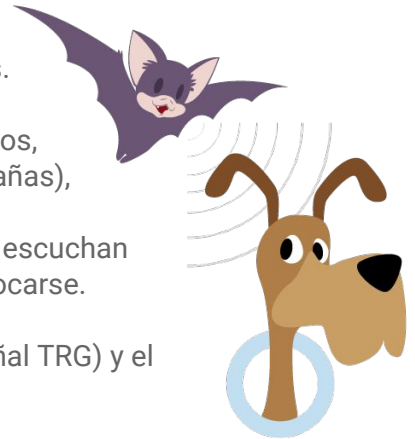
Un sensor de ultrasonidos funciona como los ‘ojos’ de los murciélagos.

¿Sabes cómo ven los murciélagos por la noche?

Lo que hacen los murciélagos es lanzar unos gritos cortos y muy agudos, esperan a que el sonido rebote en las cosas (como el eco en las montañas), y lo oigan de nuevo.

Si tardan mucho en escucharlo, será porque el objeto está lejos; y si lo escuchan muy pronto, es porque está cerca y tendrá que tener cuidado de no chocarse.

El sensor hace lo mismo, uno de los “ojos” emite el sonido (cable o señal TRG) y el otro que detecta ese sonido cuando vuelve (señal ECH).



- ¿Cómo lo conecto a la placa?

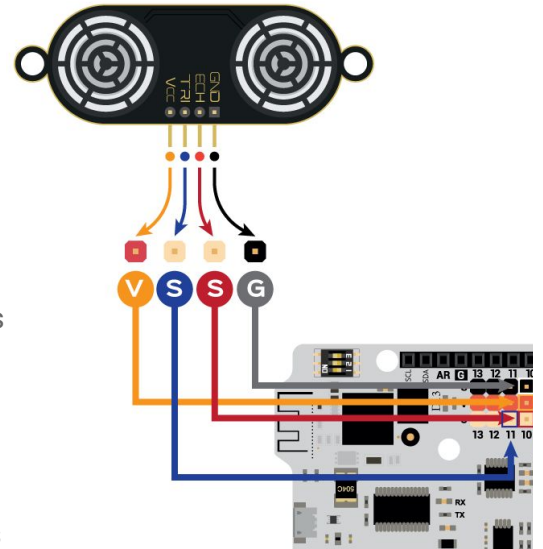
Como el sensor de ultrasonidos nos permite saber muchos valores distintos, puedes pensar que será analógico...Pero no, en realidad el sensor no me dice la distancia, sólo manda una señal cuando ha detectado, y el programa calcula la distancia. Así que su estado es “no estoy midiendo” (0) o “sí estoy midiendo” (1). Por esto, lo conectamos a los pines digitales.

El componente tiene 4 cables, que son para:

- VCC: se conecta a la tensión (Voltage) de la placa (pines rojos de la placa).
- GND: se conecta a la masa (Ground) de la placa (pines negros).

VCC y GND son como la pila del sensor, le dan energía.

- TRG y ECH: son señales, así que se conectan a los pines blancos o amarillos.

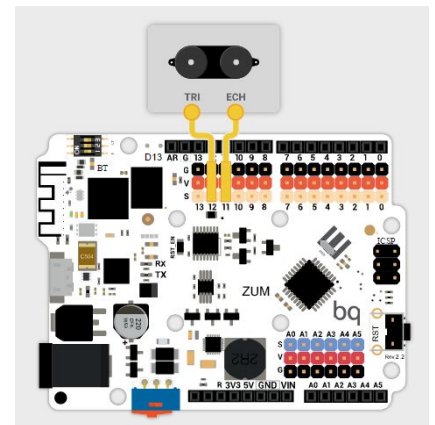


- ¿Y en Bitbloq?

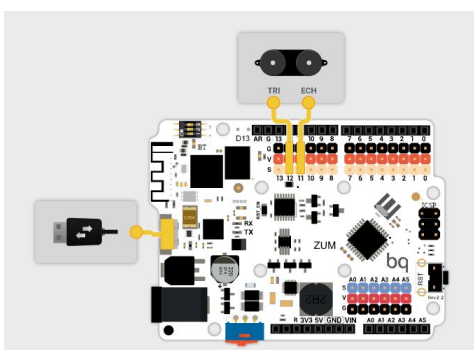
Como puedes ver en la imagen de la conexión en Bitbloq, lo que le importa a la placa son las señales (cables TRG y ECH), por lo que debemos hacer coincidir los pines donde hemos conectado estos cables a la placa física con los de la placa de bitbloq.

- ¿Cómo lo programo?

Para leer el sensor con Bitbloq se usa el siguiente bloque, que nos dirá la distancia **en centímetros** hasta el objeto que haya delante.



Pero para poder ver los valores que va leyendo, usaremos el puerto serie, que nos permitirá ver la distancia que mide el sensor de ultrasonidos en la pantalla del ordenador. Para ello, añadimos el componente a la placa y ponemos en la parte *Bucle principal* (Loop) los siguientes bloques:



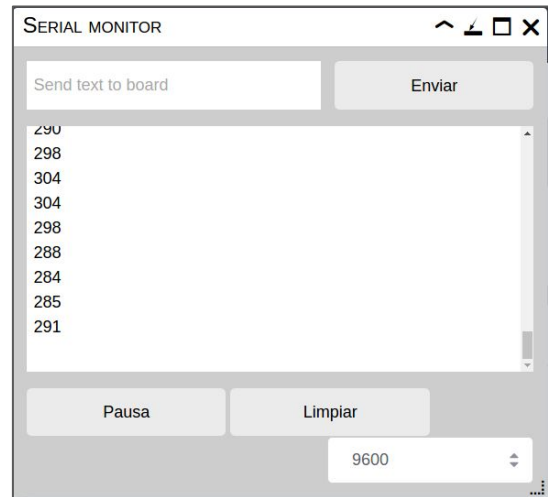
— Bucle principal (Loop)



Cuando ya tengas el programa, conecta tu placa al ordenador, clics sobre el botón cargar y selecciona "Mostar Serial Monitor" en la opción "Ver". De esta forma, aparecerán los valores o distancia detectada por el sensor de ultrasonidos. Prueba acercando y alejando la mano al sensor para observar cómo cambian los valores.

Para más información, consulta los siguientes post:

- <http://diwo.bq.com/abre-los-ojos-el-sensor-de-ultrasonidos/> - Sensor de ultrasonidos
- <http://diwo.bq.com/comunicandose-con-puerto-serie-bitbloq2/> - Puerto serie



## ¡A Programar!

Ahora viene lo bueno, ya tienes la conexión hecha y sólo te falta juntar todo esto para hacer tu alerta. Para ello, deberás programar cuándo tiene que sonar la alerta. ¿Cómo?

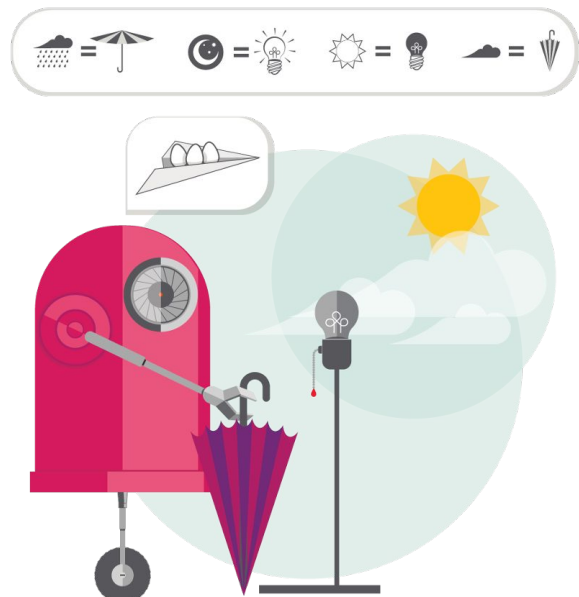
Una pista, necesitarás hacer que el programa pueda elegir cuándo alertar al conductor. Para ello, tendrás que utilizar un condicional que te permita indicar al programa que si el ultrasonidos detecta una distancia menor a la necesaria, el zumbador suene.

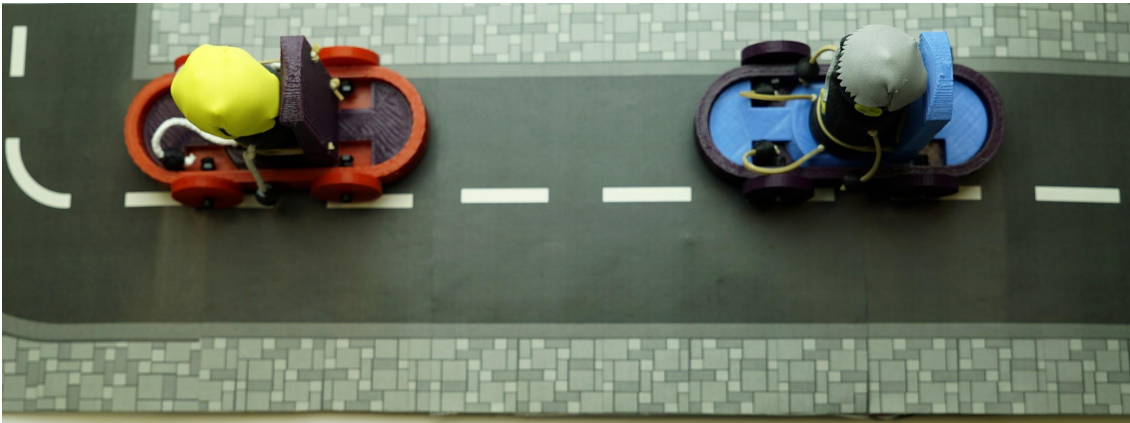
### • ¿Qué es un condicional?

El bloque condicional nos permite hacer que el programa decida qué hacer según una condición. Un ejemplo de la vida diaria es el siguiente: si llueve, abriré el paraguas. De lo contrario, si no llueve, lo cerraré.

En el caso de la alarma, tendremos que programar que si distancia es menor a un valor, por ejemplo 70 cm, suene el zumbador.

¡Anímate a programar la alarma! Si no lo consigues o tienes problemas, mira en la siguiente página y sigue los pasos que se proponen.



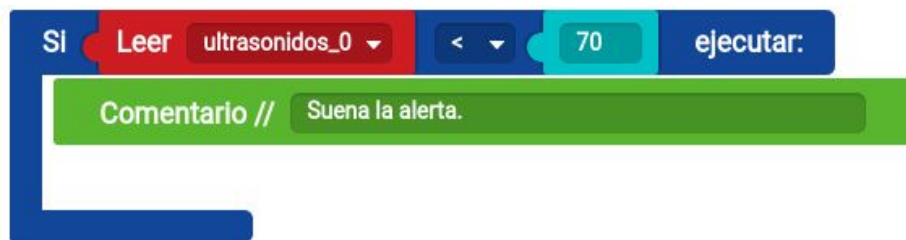


Si te has atascado, o salen cosas raras, puedes seguir estos consejos:

1) ¿Cómo uso el condicional?

Lo primero que tendrás que preguntarte es: ¿Qué es lo que tiene que hacer el sensor de ultrasonidos?

Como puedes ver en la imagen, hay dos opciones: el coche está a suficiente distancia (por ejemplo, más de 70cm), o está muy cerca (menos de 70cm). De esta manera, si está a **menos de 70cm**, el zumbador sonará.



El número puedes ponerlo desde el bloque "Matemáticas".

En el programa de ejemplo sólo he puesto la primera parte del condicional porque sólo necesito que haga algo distinto si estoy cerca; si estoy lejos, no necesito que el programa haga nada. Pero tú puedes hacer, por ejemplo, que haya una melodía si es seguro, y una alerta si está muy cerca; o que encienda las luces si está a suficiente distancia.

2) Ya sólo queda poner el zumbador, que será igual que en la prueba que has hecho antes:



**¡Y ya tienes lista tu alerta!**

¿Se te ocurre cómo mejorarla?

¿Es siempre la misma distancia o dependerá de la carretera y velocidad a la que vayas?