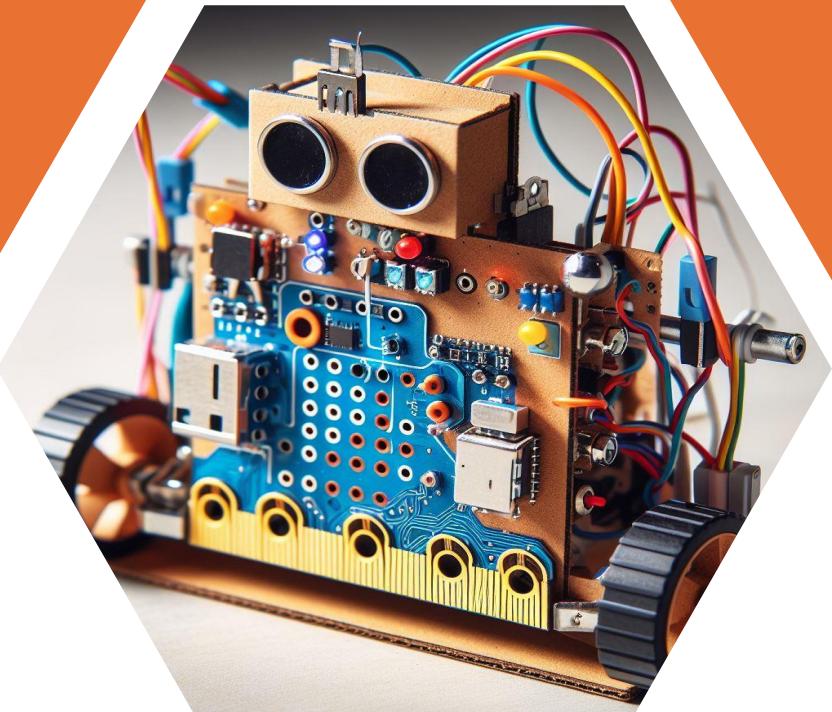
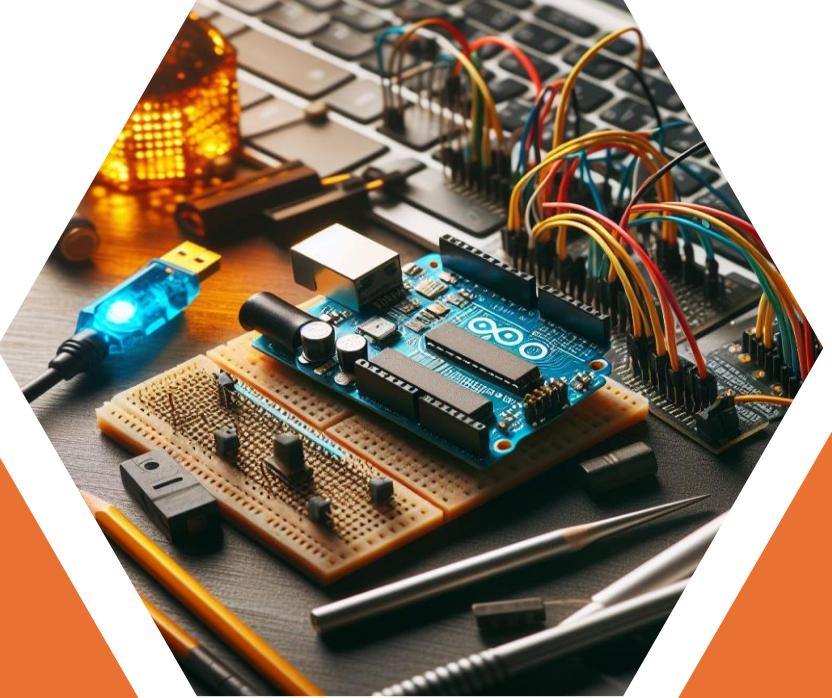


Robótica educativa con micro:bit

Enero-Abril 2025



CEIP ANTONIO MACHADO



Temporalización:



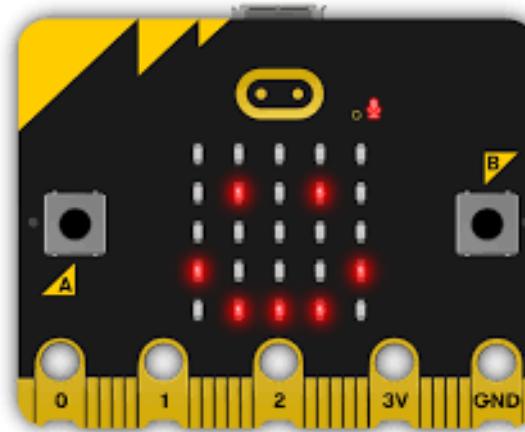
- Sesión 1: Conociendo la placa micro:bit
- **Sesión 2: Aprendiendo a programar**
- Sesión 3: Componentes externos
- Sesión 4: Robots: Cutebot, Cardboard robots...



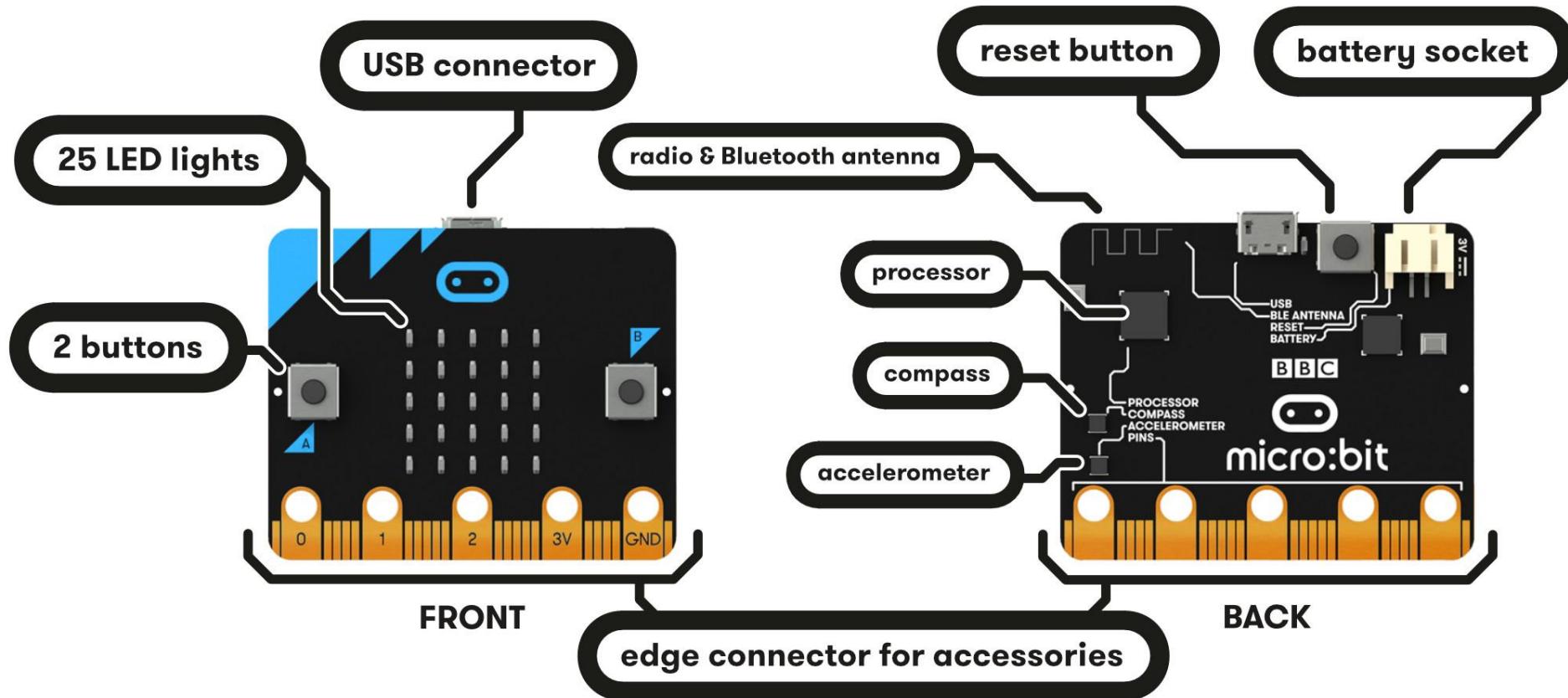


2^a Sesión:

Aprendiendo a programar



Placa micro:bit

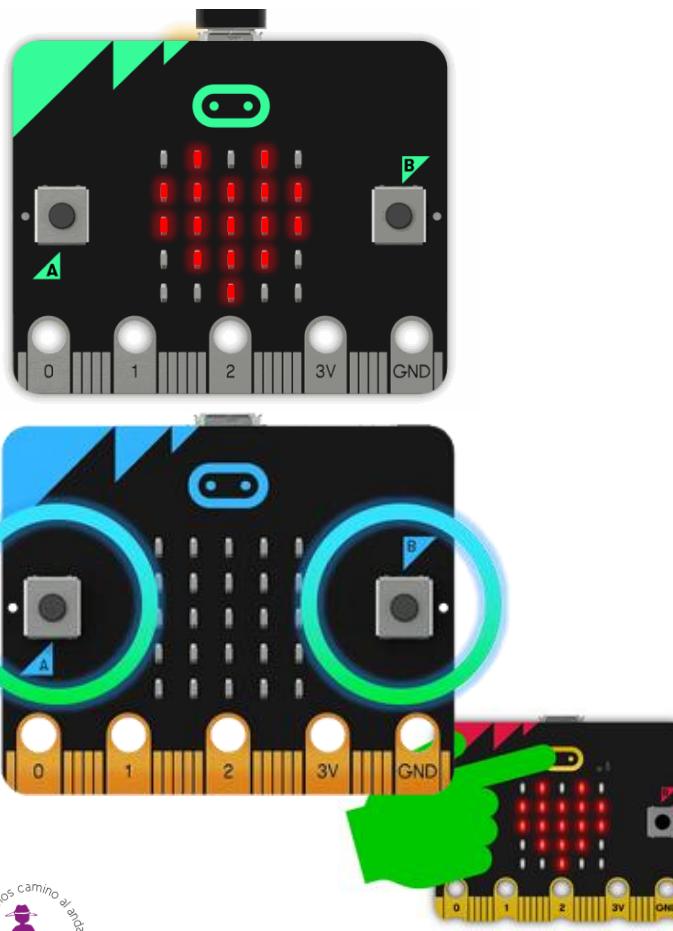


Editor MakeCode

A screenshot of the Microsoft MakeCode editor for micro:bit. The interface includes:

- A top navigation bar with the Microsoft logo, a micro:bit icon, tabs for "Bloques" (Blocks) and "JavaScript", a search icon, and a sign-in button.
- A left sidebar showing a digital representation of a micro:bit board with pins labeled A, B, 0, 1, 2, 3V, and GND. It also lists categories: Básico, Entrada, Música, LED, Radio, Bucles, Lógica, Variables, Matemática, and Extensiones.
- A workspace with two script blocks: "al iniciar" (when start) and "para siempre" (forever).
- A large purple puzzle piece icon and the text "Microsoft MakeCode".
- Buttons at the bottom for "Descargar" (Download), a file name "primerprograma", and various control icons.

Think:



Reto 1: LEDs y botones



Vamos a realizar un programa que realice lo siguiente:

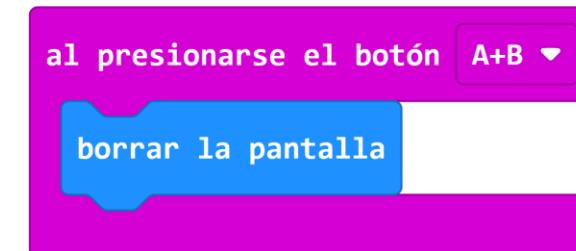
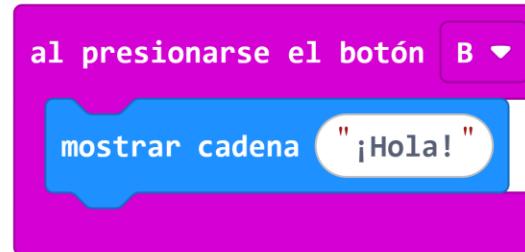
- Si pulsamos el **botón A** que muestre una imagen prediseñada
- Si pulsamos el **botón B** aparezca un texto corto
- Si pulsamos el **botón capacitivo** (logotipo) que muestre una imagen diseñada por vosotr@s.

Ampliación:

- Investiga como puedes borrar las imágenes mostradas en la matriz de LEDs agitando la placa micro:bit o pulsando dos botones a la vez (A+B).

Create:

Reto 1: LEDs y botones



Think:



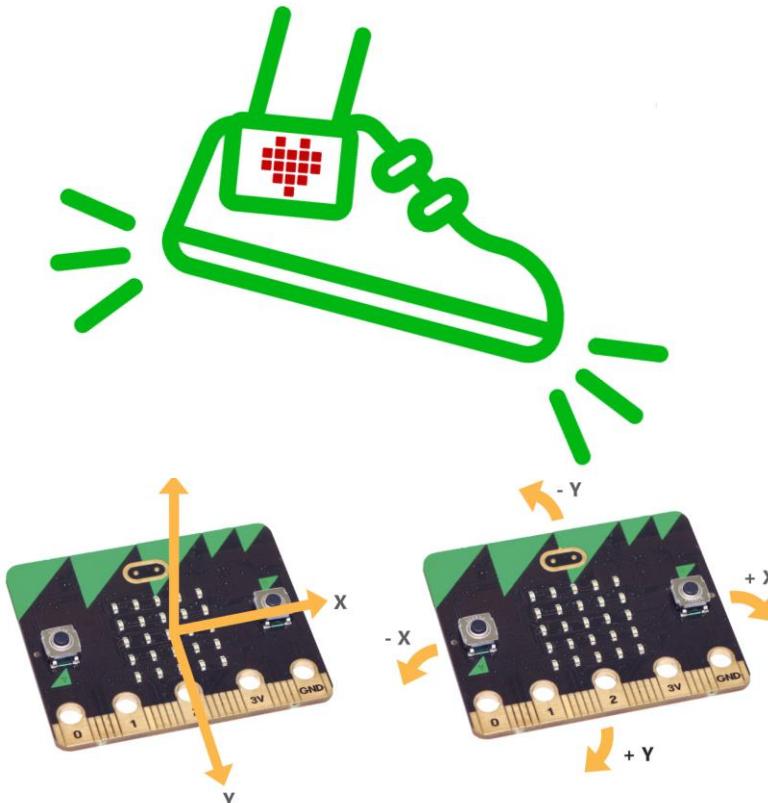
Reto 2: Podómetro

Vamos a realizar un programa que realice la función de podómetro, para ello:

- Crearemos una **variable** para contar los pasos
- Usaremos la función agitar para incrementar el número de pasos.
- Cada vez que se agite mostrará el número de pasos registrado.

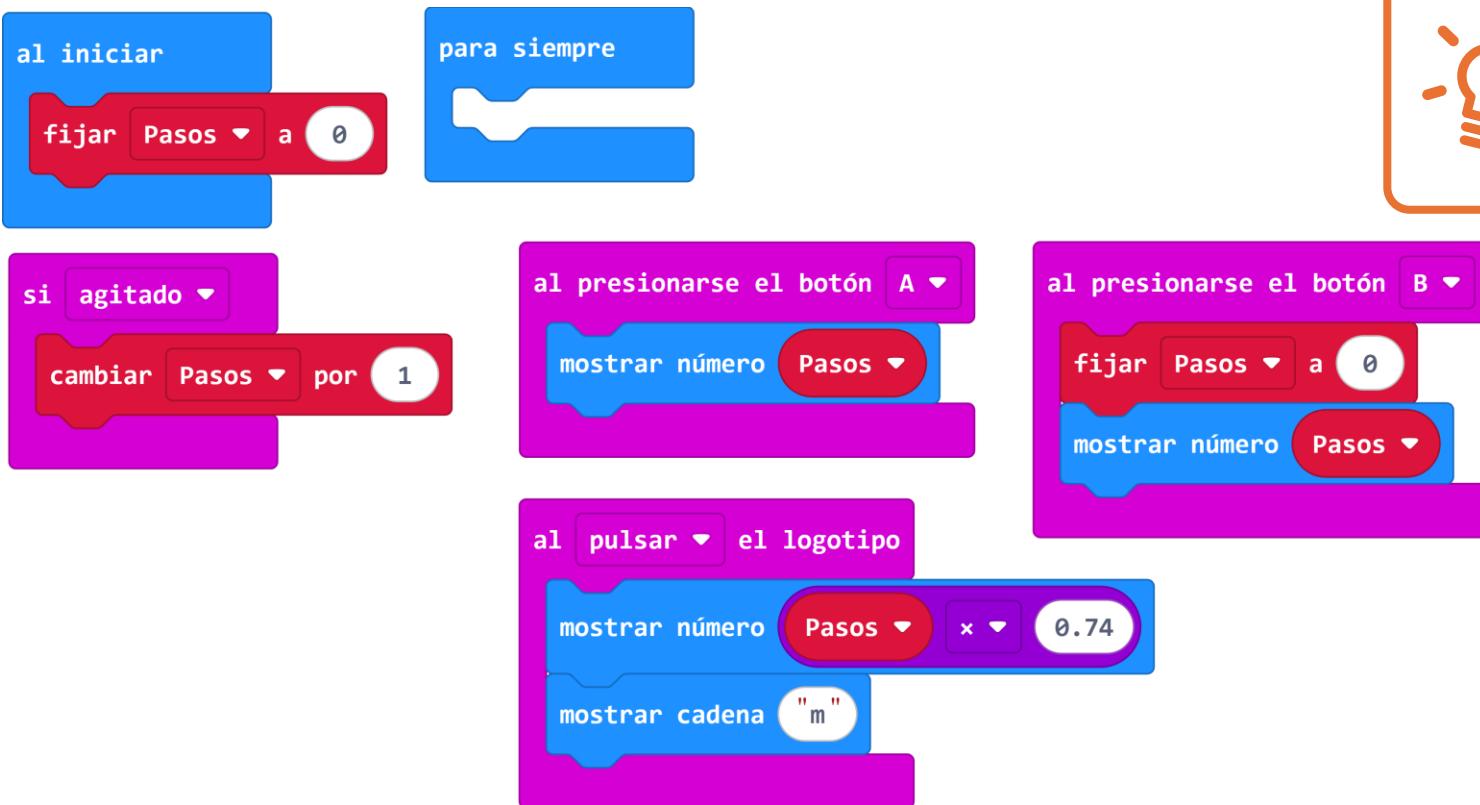
Ampliación:

- Quita la instrucción que muestra el número de pasos cuando agitas la placa y utiliza el **botón A** para mostrar el total. Con el **botón B** resetea el contador.
- Utiliza los bloques de **matemáticas** para calcular la distancia real recorrida y muéstralala pulsando el **botón del logotipo**.



Create:

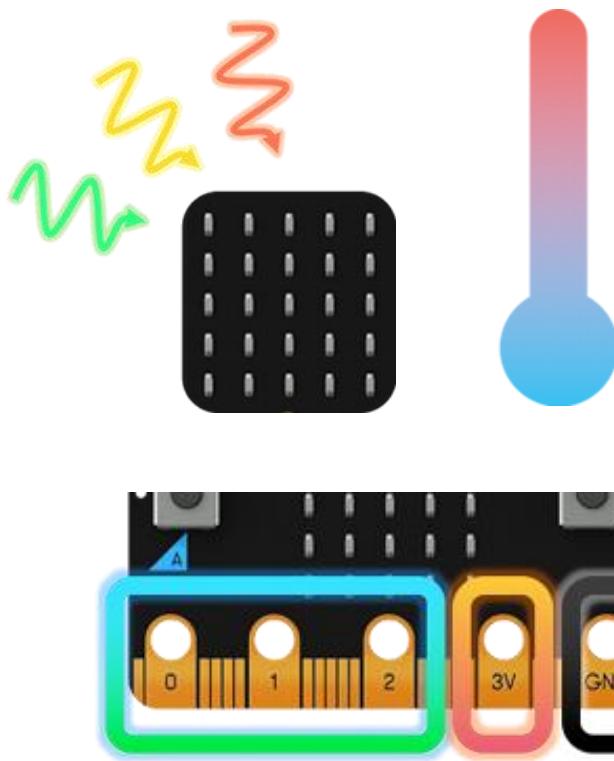
Reto 2: Podómetro



- Puedes aprovechar e introducir las medidas de longitud y sus múltiplos. Incluso medir un paso de l@s alumn@s para hacer el cálculo de la distancia recorrida.



Think:



Reto 3: Sensor de luz y de temperatura

Vamos a realizar un programa que detecte cuando hay mucha o poca luminosidad.

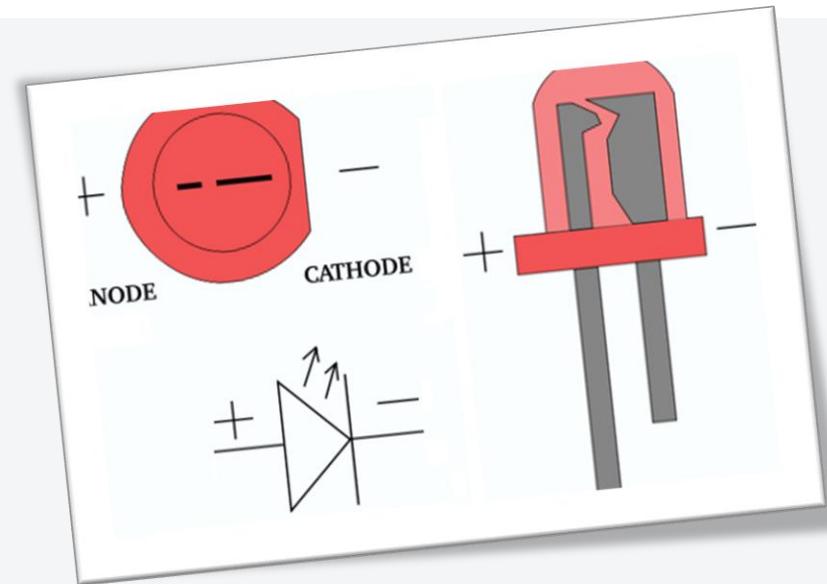
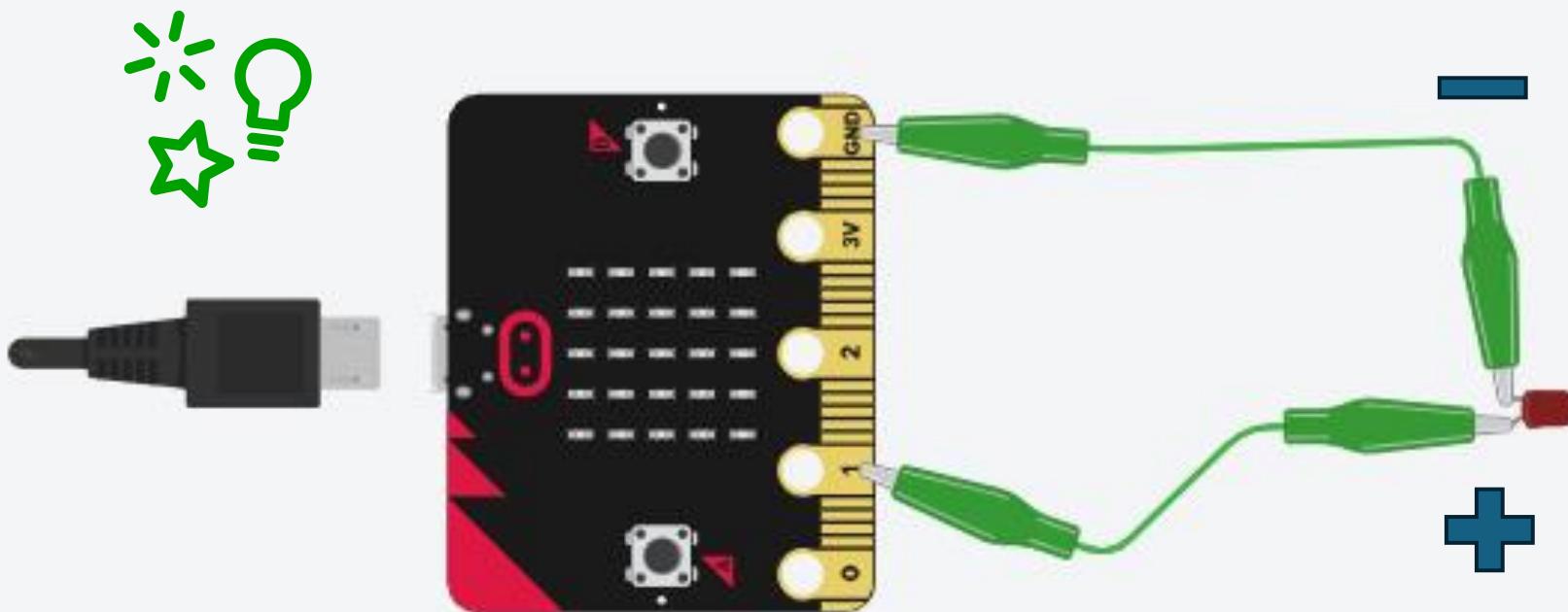
- Utilizaremos la instrucción **nivel de luz** dentro del bloque **Entrada**
- Utilizaremos la instrucción **condicional** dentro del bloque Lógica para comprobar el nivel de luz y mostrar un sol cuando haya mucha luz y una luna cuando exista poca.

Ampliación:

- Realiza el mismo programa, pero con el sensor de temperatura
- Investiga como encender un pin, por ejemplo, el pin2 y actívalo cuando haya poca luz y desactívalo cuando haya mucha luz. En ese pin podríamos conectar un diodo LED externo para que se comportase como una farola.

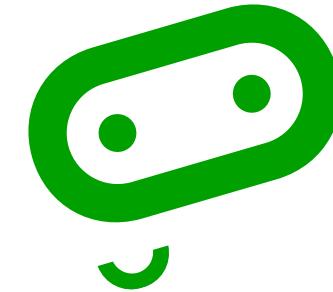
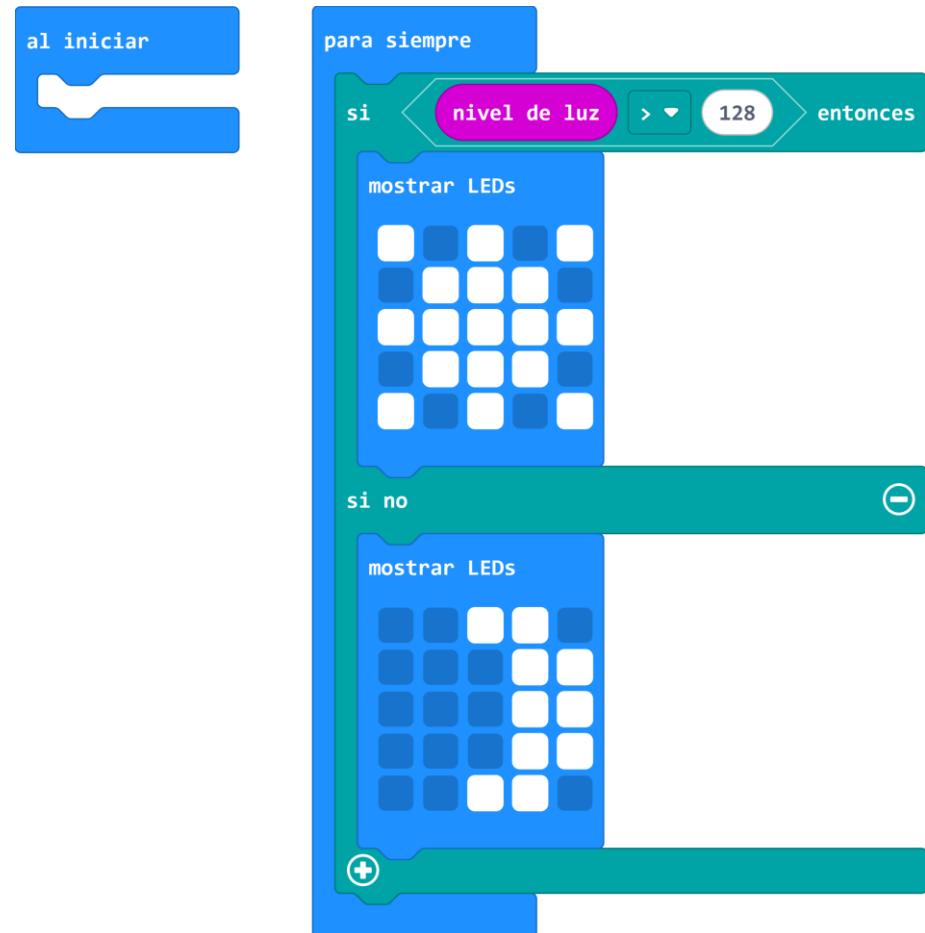
Connect:

Conexión de un diodo LED

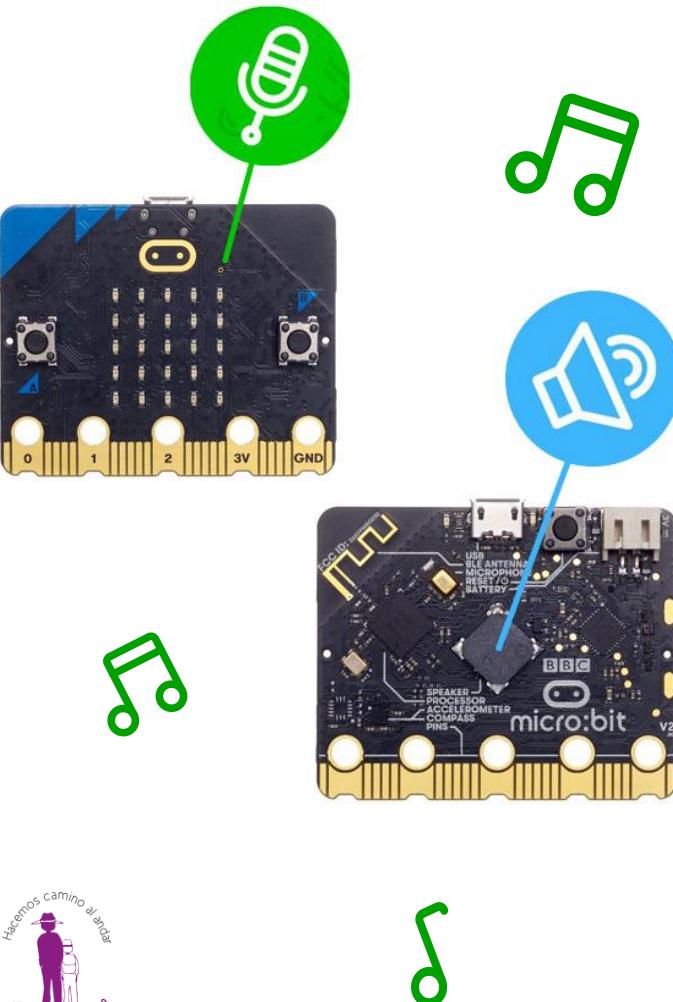


Create:

Reto 3: Sensor de luz y temperatura



Think:



Reto 4: Detector de sonido alto



Vamos a realizar un programa que detecte un sonido alto para que nos avise cuando haya mucho ruido en clase.

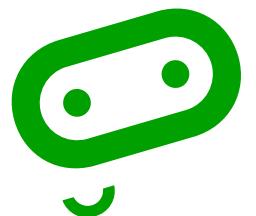
- Utilizaremos la instrucción al **detectar sonido ...** del bloque de **Entrada**. Lo configuraremos en **alto**. Cuando ocurra esto mostraremos una cara triste. Realizaremos la misma operación con otra instrucción configurada en **silencioso** y mostraremos una cara soniente.
- El nivel de sonido tiene un rango de 0 a 255, cuanto mayor el es valor, mayor es el volumen del sonido que detecta. Puedes configurar el valor del rango alto a través de una instrucción, investiga en **Entrada**.

Ampliación:

- Vamos a añadir un sonido de aviso cuando haya mucho ruido. Programa un pitido cada medio segundo cuando haya mucho ruido.

Create:

Reto 4: Detector de Sonido alto



Scratch script for a micro:bit project:

```
al iniciar
  establecer el umbral de sonido alto ▾ al valor 128
para siempre
  al detectar el sonido alto ▾
    mostrar ícono [grid v]
    reproducir secuencia tono [Do medio] durante [1 ▾ pulso] en modo [hasta que termine ▾]
    pausa (ms) [500 ▾]
  fin
  al detectar el sonido silencioso ▾
    mostrar ícono [grid v]
  fin
```

Think:



Reto 5: Juego, piedra, papel o tijera

Vamos a realizar un programa simule el juego de piedra, papel o tijera.



- Necesitaremos una instrucción que nos genere un número aleatorio, entre 1 y 3, los tres objetos (piedra, papel o tijera), **escoger al azar** (bloque Matemática). Utiliza ese bloque para almacenar el resultado la instrucción en una variable.
- Ahora utilizaremos la instrucción **condicional** (bloque Lógica) para comparar el valor de esa variable con 1, 2 o 3, piensa como puedes realizarlo.
- Para generar el número aleatorio vamos a utilizar el bloque agitar, cada vez que se agite tendrá que generar un número aleatorio y mostrar una imagen, piedra (1), papel (2) o tijera (3).

Ampliación:

- Configura un botón para borrar el objeto mostrado.

Create:

Reto 5: Piedra, papel o tijera

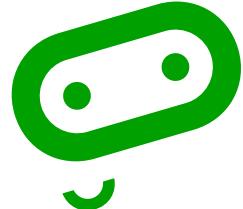


al iniciar

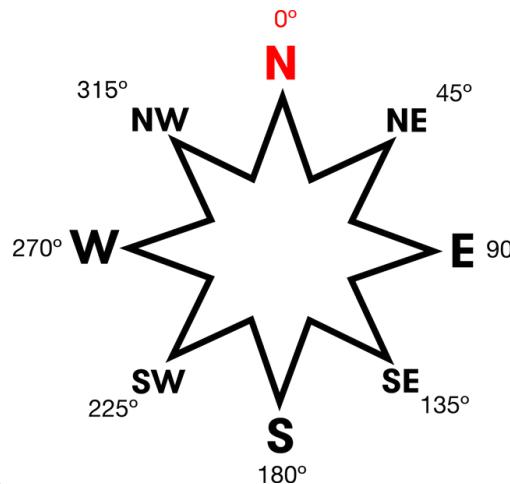
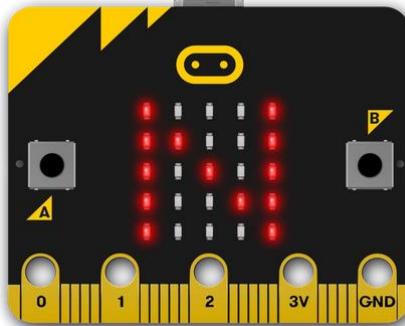
para siempre



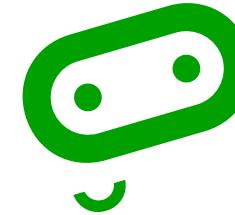
```
si agitado
fijar n_aleatorio a escoger al azar de 1 a 3
si n_aleatorio = 1 entonces
mostrar ícono ☒
si no, si n_aleatorio = 2 entonces -
mostrar ícono ☓
si no
mostrar ícono ☔
+
al presionarse el botón A
borrar la pantalla
```



Think:



Recursos:



- Lecciones por edades: <https://microbit.org/teach/featured/>
- Guía para trabajar con los ODS y micro:bit:
<https://microbit.org/teach/do-your-bit/#global-goals-guides>

