José Antonio Vacas Martínez

# Introducción a la Programación y la robótica con micro:bit

## Versión 0.99



Licencia CC by SA

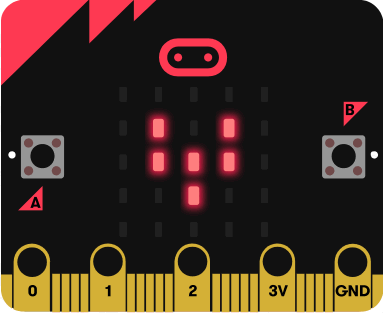
## by @javacasm

### José Antonio Vacas Martínez

### https://elCacharreo.com

## Mayo 2021

# Taller micro:bit para el Museo de las Ciencias de Granada



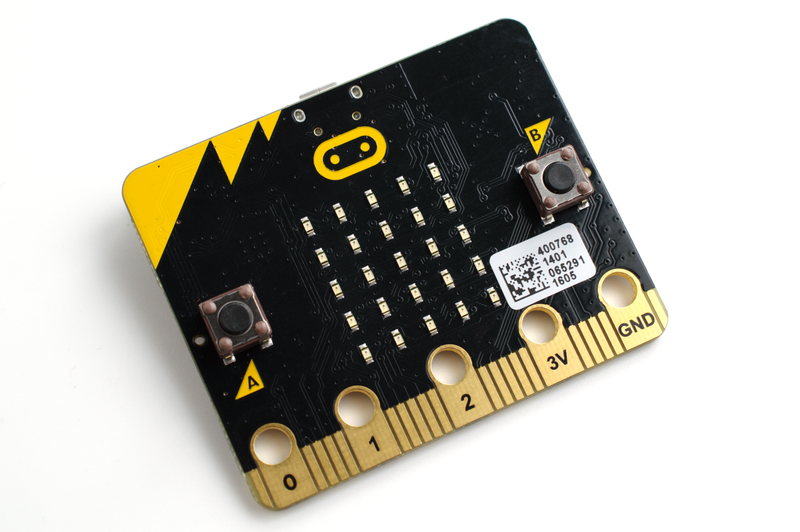
Love with micro:bit

 by @javacasm

En este taller vamos a ...

* Aprender qué es micro:bit y para qué podemos usarla
* Programar micro:bit con bloques usando [MakeCode](https://makecode.microbit.org/)
* Conectar sensores con los que mediremos temperatura, luz y sonido
* Controlar motores
* Encender y apagar luces
* Producir sonidos

Para ello usaremos la placa [micro:bit](https://microbit.org/)



y el [Kit de Iniciación a la electrónica de MonkMakes](http://www.monkmakes.com/mb_kit.html) que tiene los siguientes componentes:

* Placa de sensores de sonido, luz y temperatura
* Relé de estado sólido
* Motor
* Luz incandescente
* Altavoz



## [¿Qué es micro:bit?](./QueEsMicrobit.md)

## [Programando micro:bit con MakeCode](./Taller.md)

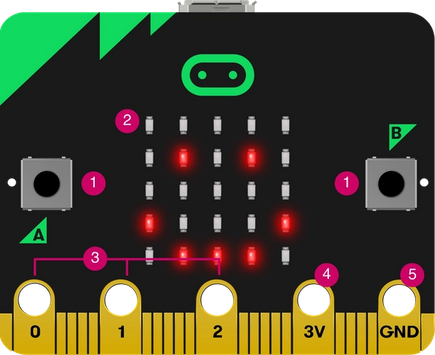
## ¿Qué es [micro:bit](http://microbit.org)?

micro:bit es un pequeño micro-ordenador programable, diseñado para hacer fáciles y divertidos la enseñanza y aprendizaje de temática #STEAM.

Nos permite trabajar la computación física, haciendo "tocable" el pensamiento computacional de una forma sencilla.

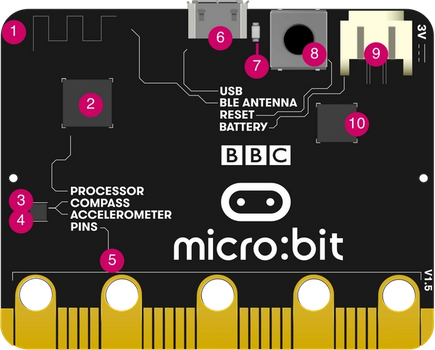
Beneficiada de la experiencia de otros muchos sistemas similares anteriores (como Arduino o Raspberry pi) incluye en una pequeña tarjeta una gran cantidad de sensores, leds y botones, permitiéndonos hacer una gran cantidad de proyectos distintos usando sólamente la tarjeta micro:bit

Esta es una breve descripción de su hardware:



micro:bit front

1. Botón A y Botón B
2. 5x5 leds rojos
3. Conectores para pinza de cocodrilo P0, P1, P2 (GPIO)
4. Conector 3V (Voltaje)
5. Conector GND (Tierra)



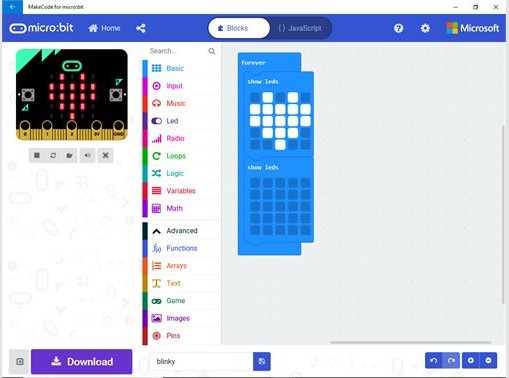
micro:bit back

1. Antena de radio y bluetooth
2. Procesador y sensor de temperatura
3. Brújula
4. Acelerómetro
5. Pines (GPIO)
6. Conector microUSB (alimentación y carga de programas)
7. Led de alimentación y/o programación
8. Botón reset
9. Conector de alimentación (2 x pilas 1.5V)
10. Chip USB

(Tomado de [Overview de microbit.org](https://microbit.org/get-started/user-guide/overview/))

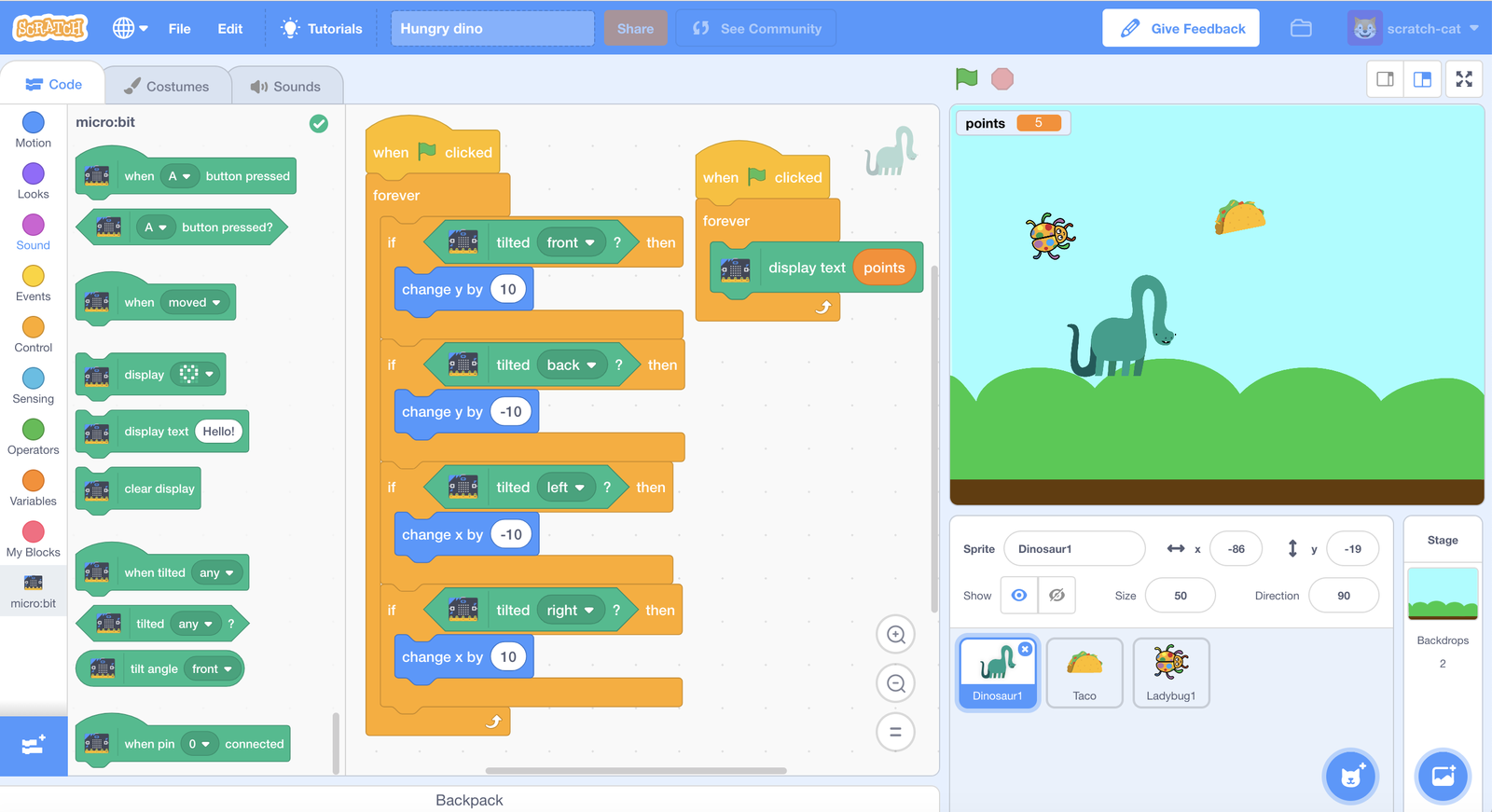
## Programación

Podemos programarla usando bloques con el editor [**MakeCode**](https://makecode.microbit.org)



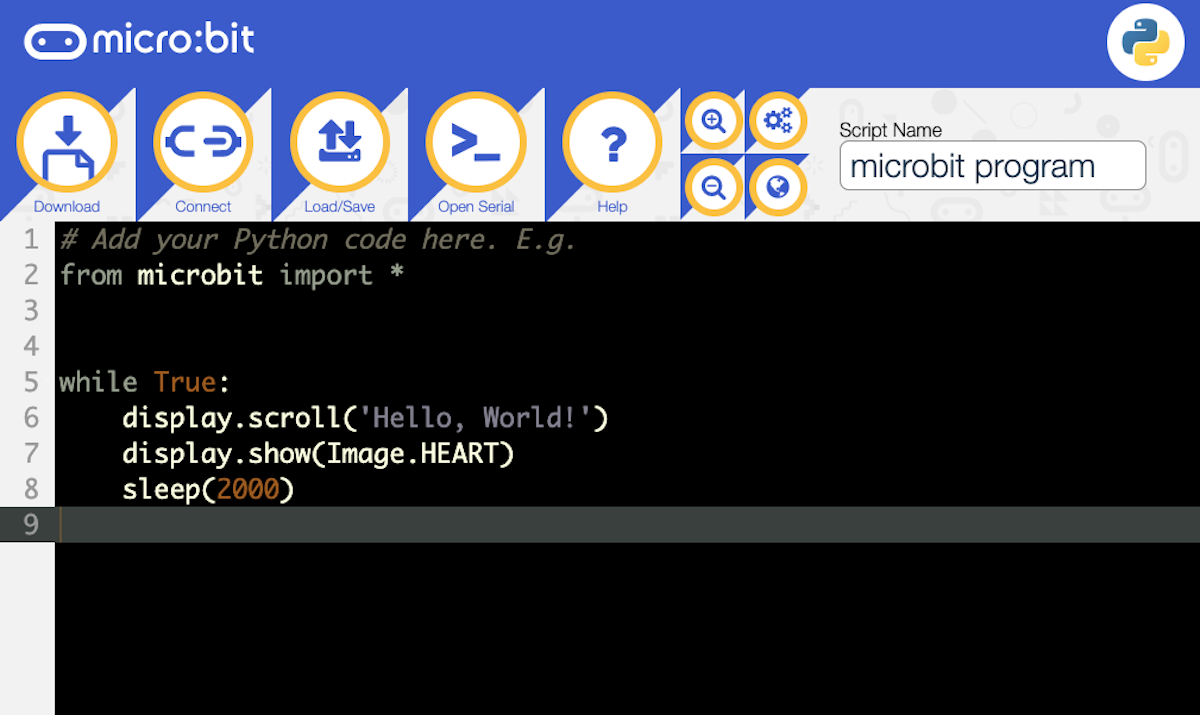
MakeCode editor

o [**Scratch**](https://scratch.mit.edu/microbit) con la extensión micro:bit



Scratch con la extensión de micro:bit

o con lenguajes de programación como [**Python**](https://python.microbit.org/v/2), Javascript, e incluso con el C++ de Arduino



micropython para micro:bit

Trabajar con Scratch permite mezclar parte de la funcionalidad de nuestra micro:bit (botones, leds, acelerómetro y conexión a los pines 0,1 y 2) con el resto de posibilidades de Scratch: personajes, sonidos, dibujar en pantalla, vídeo, música,... usándola como un mando remoto de nuestro ordenador. Perdemos algunas de las características de micro:bit como la brújula y el sensor de temperatura pero ganamos en interacción.

Para usar toda la funcionalidad de micro:bit debemos usar el editor de bloques de MakeCode o Python.

Para hacerlo también podemos usar tanto ordenadores como tablets.

# Programación con bloques de Micro:bit

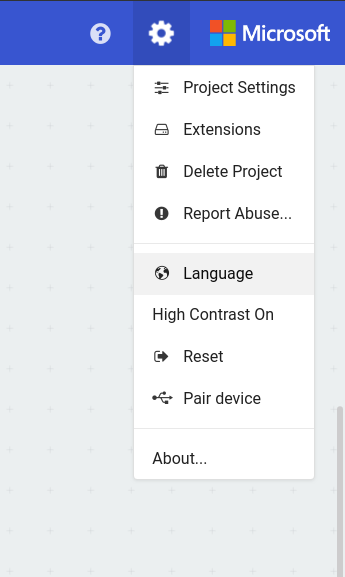
Vamos a programar nuestra micro:bit con bloques usando el editor de [Makecode](https://makecode.microbit.org)

## IDE/Entorno de programación



1. Paletas con los bloques de programación. Las paletas tienen un color determinado, igual que los bloques que contienen.
2. Tapete/Lienzo: donde arrastramos los bloques de nuestro programa
3. Simulador: va ejecutando el programa que hemos hecho
4. Botón deshacer
5. Botones de Zoom para ampliar/achicar
6. Configuración. Podemos seleccionar el idioma
7. Nombre de nuestro programa

Si no te aparece en tu idioma, puedes cambiarlo pulsando el icono del engranaje arriba a la derecha y seleccionar el idioma que prefieras



## Mostrando iconos

Hemos visto que nuestra micro:bit tiene 25 leds (5x5) en la parte delantera. Vamos a mostrar iconos.

Arrastraremos 2 bloques "Mostrar icono" desde la paleta Básico (1) al Tapete (2)



[Primer Proyecto: iconos](https://makecode.microbit.org/_E3Pe8zJRaXVg)

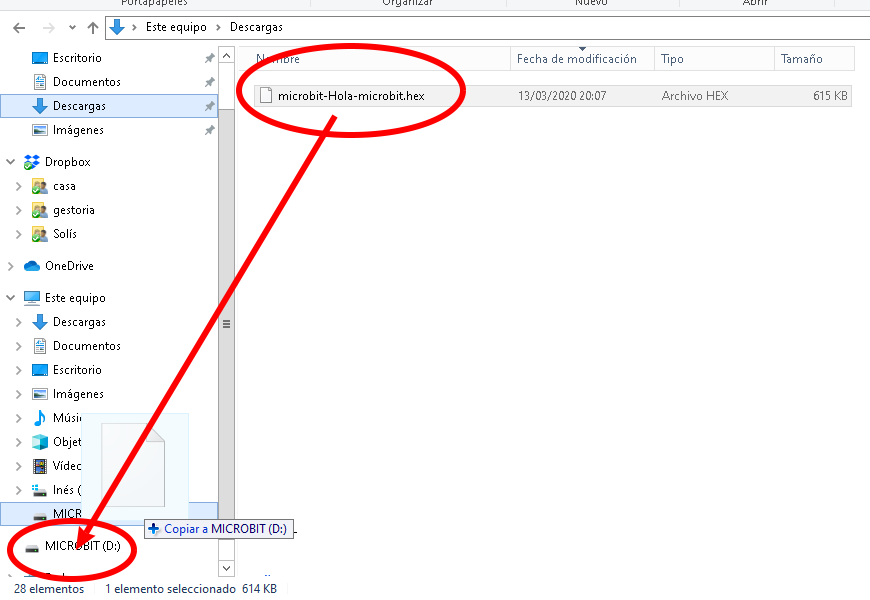
Damos nombre a nuestro programa "Emoticono" escribiéndolo en la caja del nombre (7)



Veremos como se muestran los iconos en el Simulador (3)

Ahora vamos a pasar nuestro programa a la micro:bit:

1. Conectamos la micro:bit por USB.
2. Veremos que aparece en el ordenador como si hubiéramos conectado un USB llamado "Microbit"
3. Descargamos el programa pulsando el botón "Descargar"
4. Copiamos el fichero a la unidad llamada Microbit



## Haciendo ruído

Vamos a hacer música, usando los bloques de la paleta "Música". La placa micro:bit incluye muchos sensores, vamos a usar uno de ellos, el acelerómetro, que detecta cuando la movemos.

Vamos a arrastrar desde la paleta "Entrada" el bloque "si agitado" y dentro el bloque "play melody" desde la paleta "Música"



[Música](https://makecode.microbit.org/_Rsfe6X6vMED1)

Vemos que el simulador nos indica que hace falta conectar unos cascos/altavoces. Si acercamos el ratón al simulador veremos que se mueve la placa ¡Podemos simular que la agitamos! Es un simulador muy completo. El simulador sonará al mover el ratón sobre la placa.

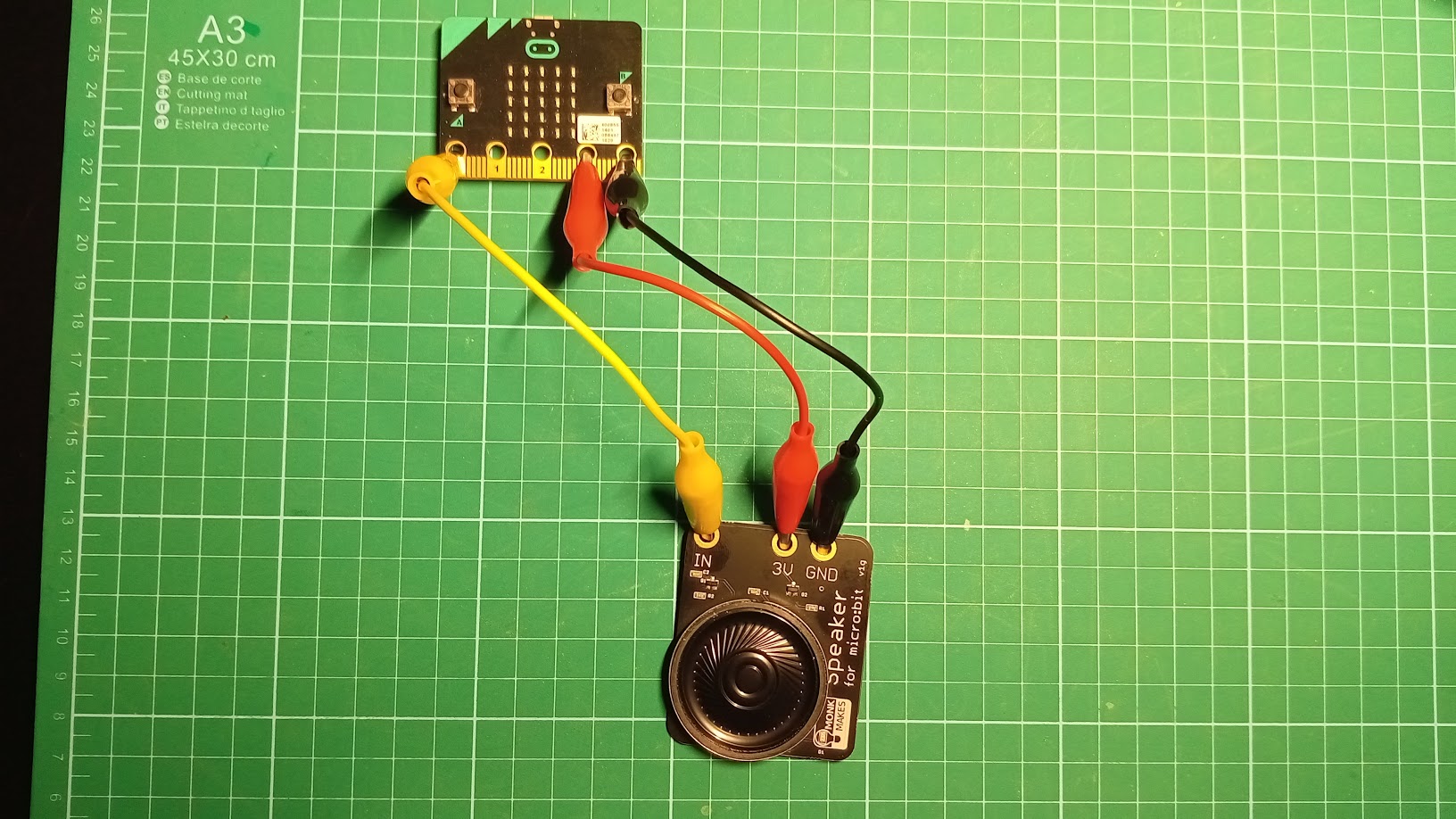
Ahora descargamos el programa como antes. Si tenemos una micro:bit v2 sonará al agitarla.

La versión 1 de la placa micro:bit no tiene altavoz y no sonará nuestro programa, por ello vamos a conectar un altavoz. Usaremos el [Kit de Iniciación a la electrónica de MonkMakes](http://www.monkmakes.com/mb_kit_es)

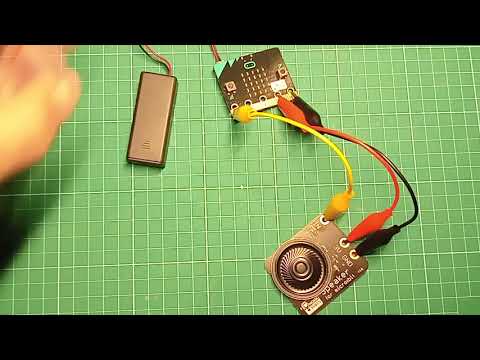


Usaremos 3 cables de pinzas cocodrilo para conectar el altavoz/Speaker y la micro:bit. Podemos usar cables de cualquier color, pero es mejor seguir un esquema, conectando:

|  |  |  |
| --- | --- | --- |
| micro:bit | Speaker | cable |
| GND | GND | cable negro |
| 3V | 3V | cable rojo |
| 0 | IN | cable amarillo |



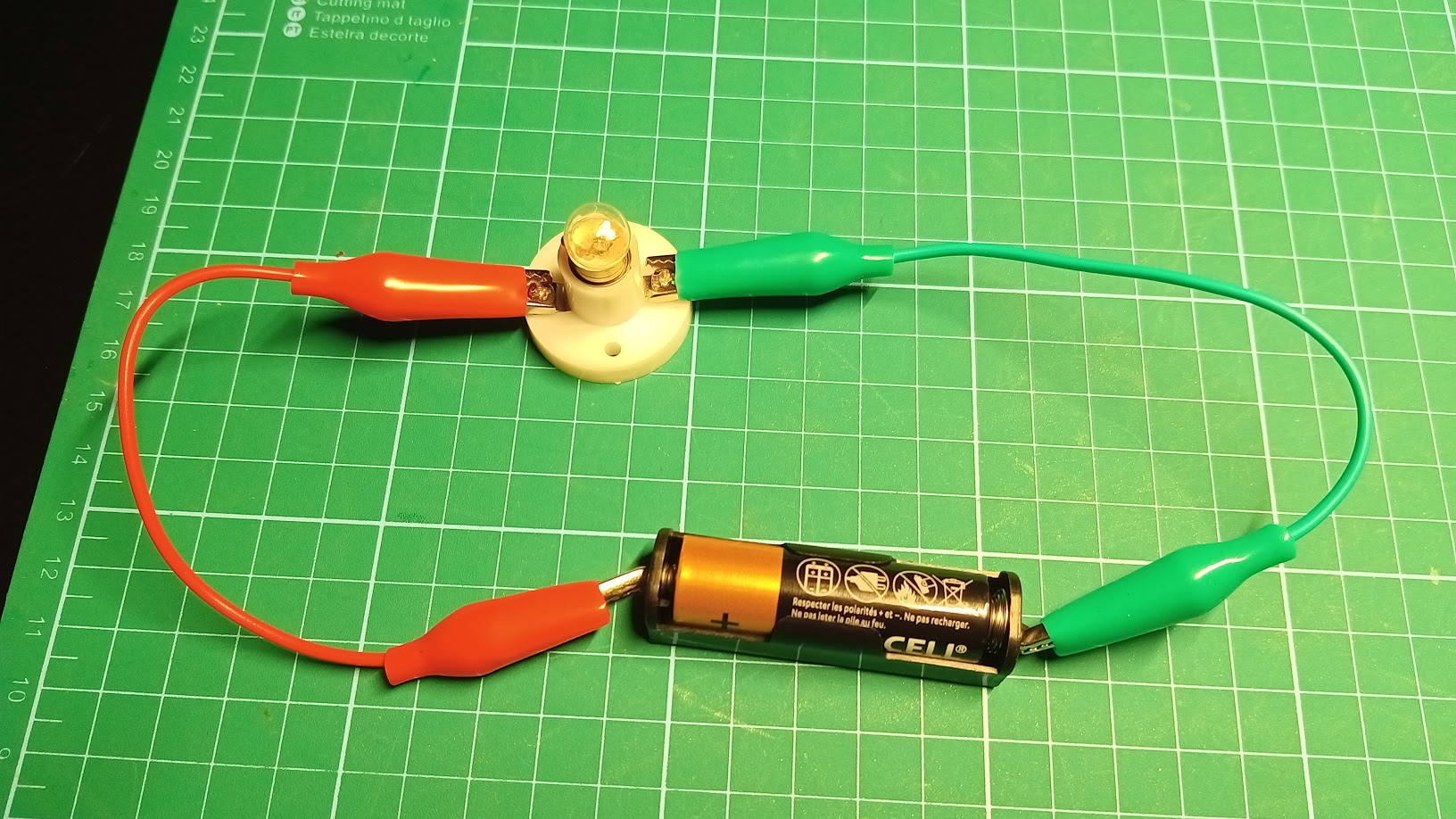
Veremos cómo suena:

[](https://youtu.be/VwGuElgN2t8)

[Vídeo: Reproduciendo música al agitar micro:bit](https://youtu.be/VwGuElgN2t8)

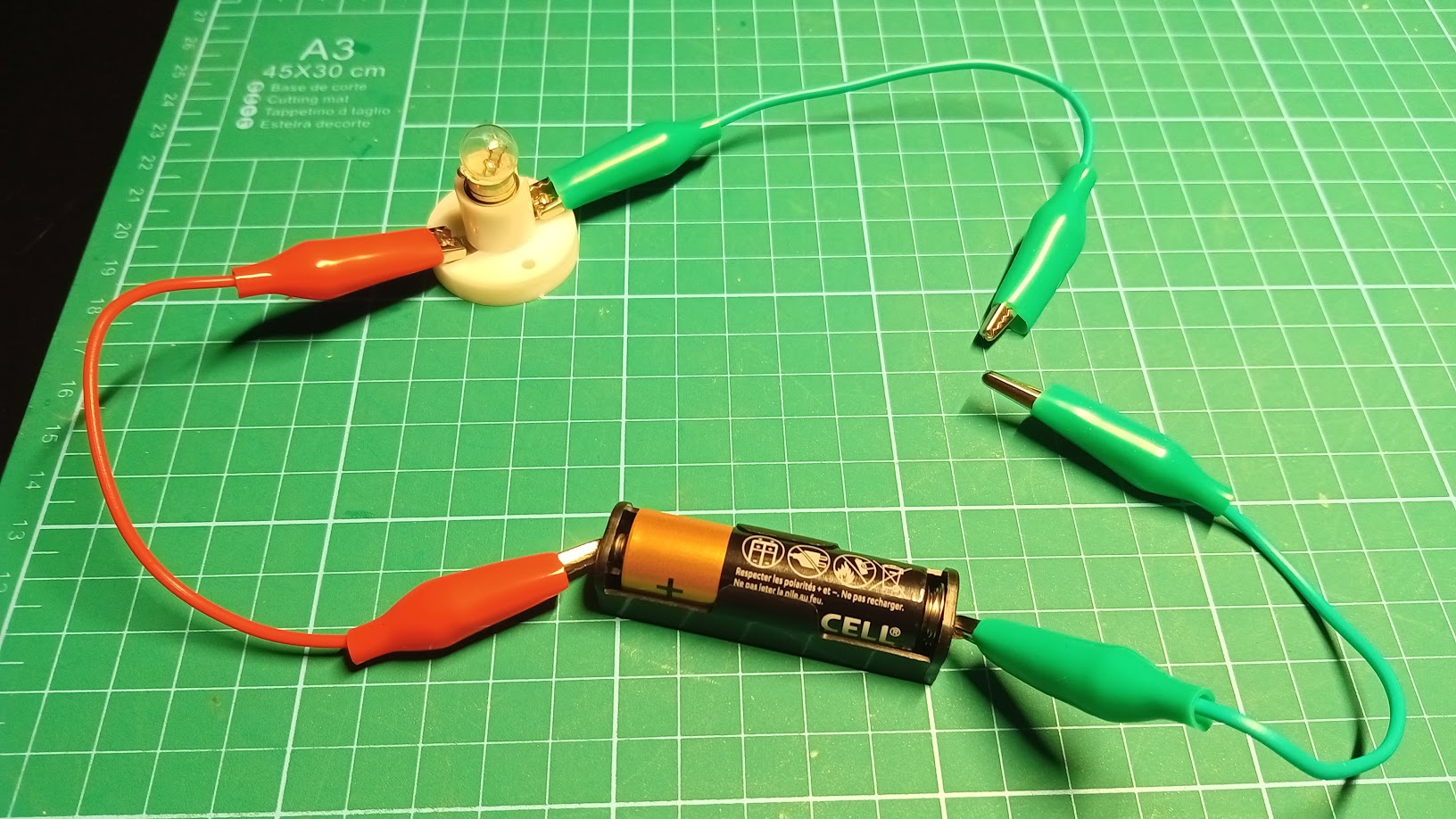
## Controlando corrientes y potencia externa: interruptor electrónico

Vamos a empezar haciendo un sencillo circuito eléctrico con la bombilla, la pila y algunos cables:



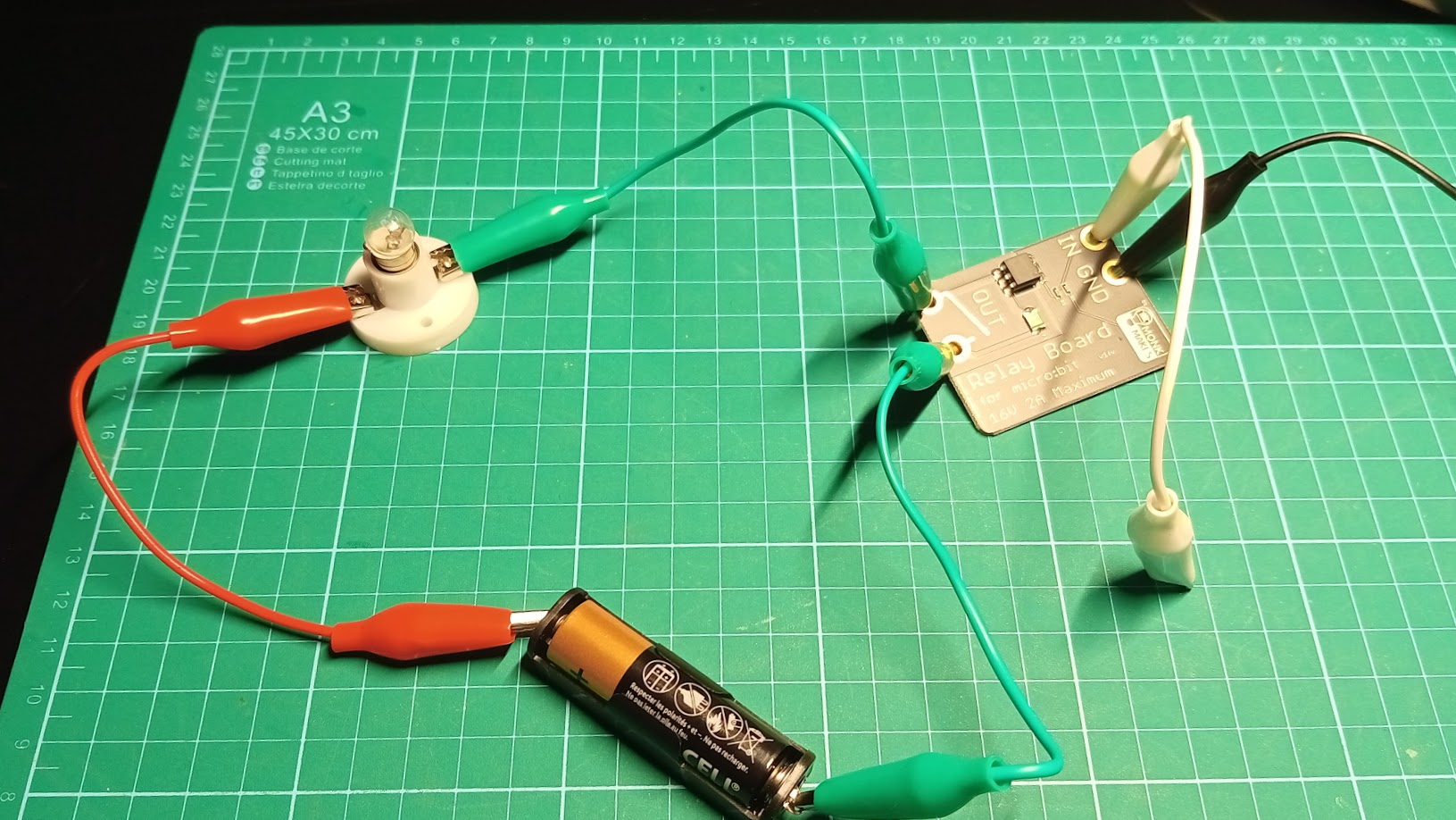
Al conectarlo se encenderá la bombilla.

Ahora vamos a hacer un sencillo interruptor hecho con cables. Sustituiremos el cable verde por 2 cables verdes.



Cuando juntemos los cables verdes se cerrará el circuito y se encenderá la bombilla.

Ahora vamos a insertar interruptor electrónico, un **Relé/Relay** que podremos controlar desde la programación de la micro:bit. Conectaremos los 2 cables verdes en la salidas **OUT** de la placa del relé. No importa cuál pongamos en cada una:



El relé nos permite controlar más potencia que lo que puede hacer la micro:bit directamente.

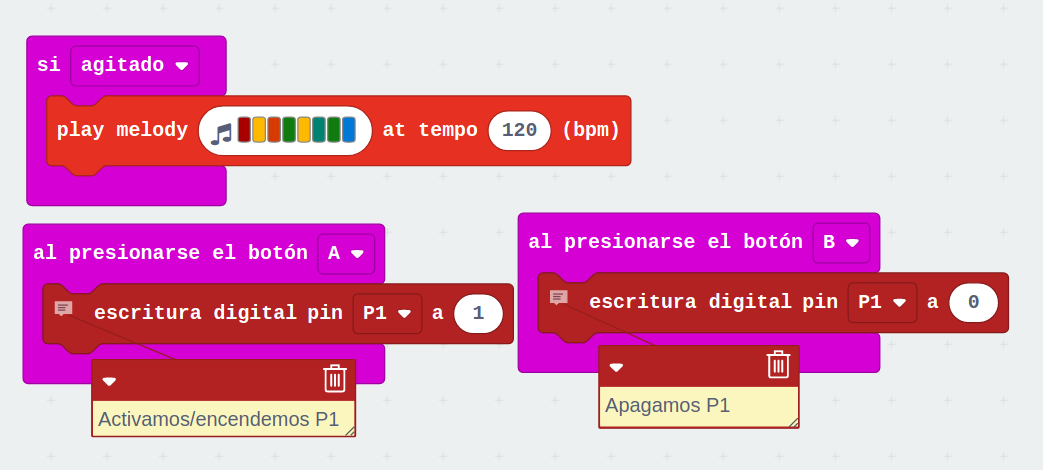
Usaremos 2 cables para conectar el Relé a la micro:bit

|  |  |  |
| --- | --- | --- |
| micro:bit | Relay | cable |
| GND | GND | cable negro |
| 1 | IN | cable blanco |

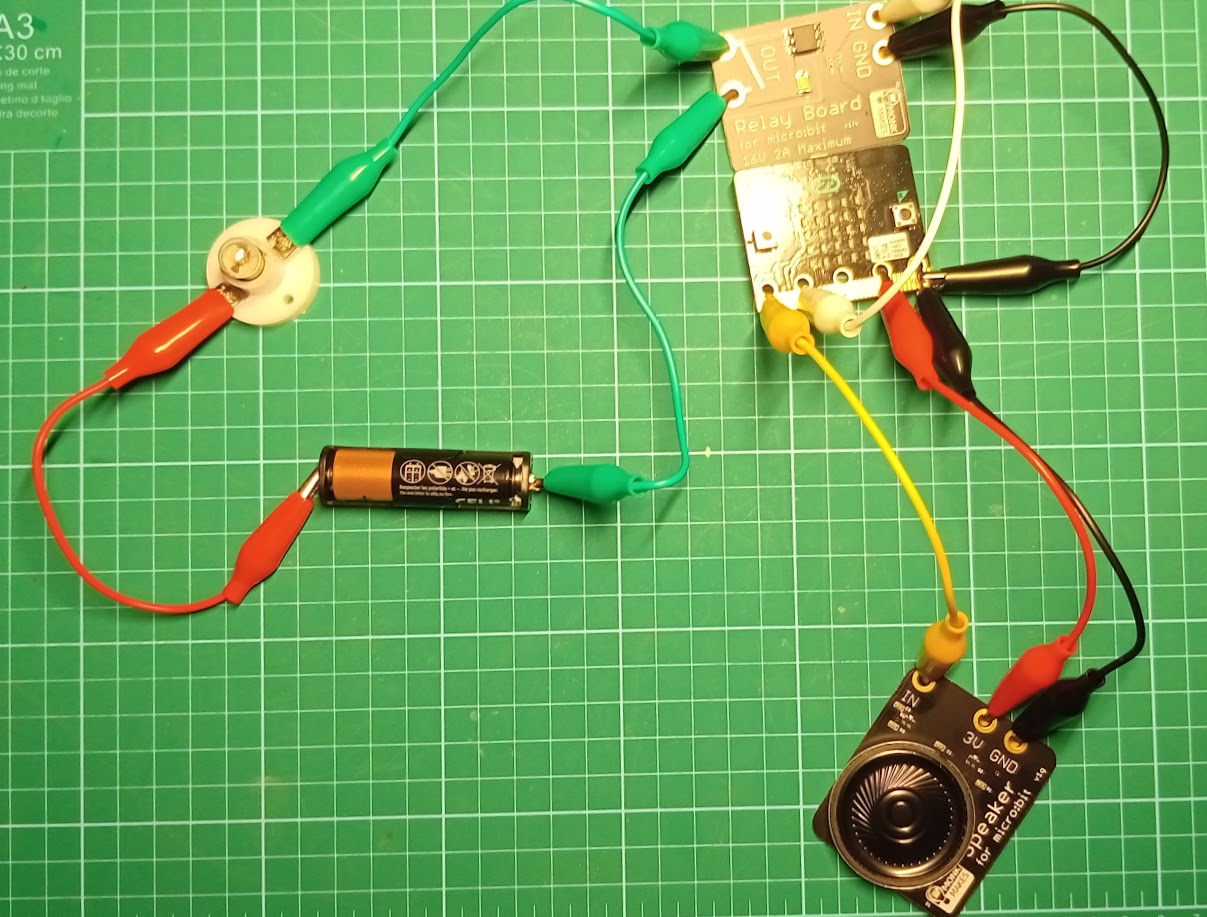
Vamos a hacer un programa para encender y apagar con los botones A y B.

* Cuando pulsemos el botón A encenderemos la salida P1 de la micro:bit
* Cuando pulsemos el botón B apagaremos la salida P1 de la micro:bit

Usaremos el bloque "Al pulsar el botón A" de la paleta "Entrada" y "escritura digital pin P1" desde la paleta "Pines" de "Avanzados". Seleccionamos "P1" y ponemos "1" para encender y "0" para apagar:



[Música + Luz](https://makecode.microbit.org/_9cCRiFb7C2ys)



* Descargamos el programa a la placa
* Pulsamos el botón A y se encenderá la bombilla
* Pulsamos el botón B y se apagará

Si queremos controlar un motor/ventilador en lugar de la bombilla, sólo tenemos que sustituir la bombilla por el motor. Al cambiarlo tendremos un ventilador controlado desde un micro:bit.

Podemos invertir el sentido de giro del ventilador, cambiando entre sí los cables del motor.

## Termostato

Otro de los sensores que incluye la micro:bit es el sensor de temperatura, situado en la parte de atrás:

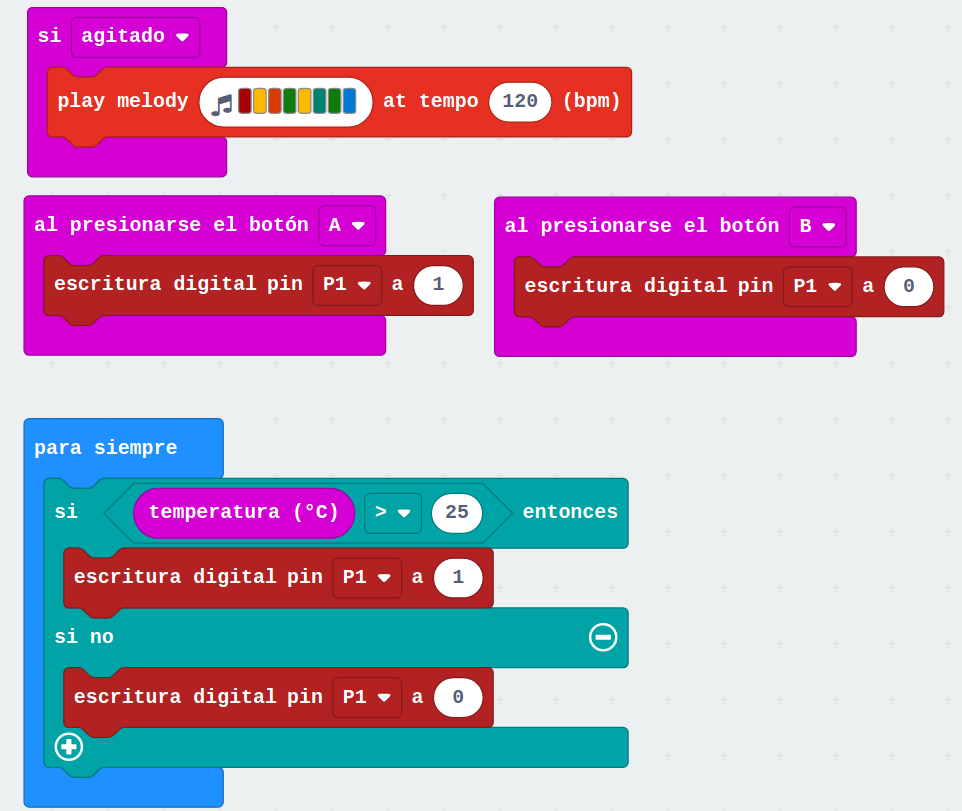


Podemos encontrar el valor de la temperatura de la placa en la paleta "Entrada".

Un termostato es un aparato que hace algo cuando la temperatura es mayor que un valor y otra cosa cuando sea menor.

Para ello vamos a usar el bloque "Si ... entonces si no" y el bloque hexagonal "operador >" ambos de la paleta "Lógica"

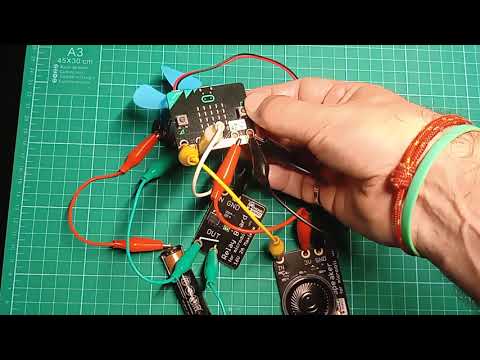
* Ponemos el bloque "Si ... entonces si no" dentro del bloque "para siempre" porque necesitamos que se esté revisando siempre esta condición
* Añadimos el bloque hexagonal "operador >" en la parte de la condición
* En un lado del operador ponemos el valor de la temperatura, de la paleta "Entrada"
* Ponemos en el otro lado el valor de la temperatura que usaremos, por ejemplo 25.
* En caso de que la temperatura sea mayor encenderemos el ventilador con el bloque "escritura digital pin P1" a 1
* En "si no " apagaremos el ventilador con el bloque "escritura digital pin P1" a 0



[Proyecto: Música, Ventilador y termostato](https://makecode.microbit.org/_DzJHtaPoT4dT)

Para probarlo:

* Descargamos el programa
* Tocamos el sensor de temperatura para calentarlo
* Se encenderá el ventilador
* Apuntamos el ventilador hacia el sensor, para enfriarlo
* El ventilador se apaga

[](https://youtu.be/9PxjRF-k8-g)

[Vídeo: Ventilador controlado con termostato usando micro:bit](https://youtu.be/9PxjRF-k8-g)