# Retos con microbit


- Panel LED
- Jugando con los LED: bloques básicos de código
- Sensor temperatura
- Sensor luminosidad y sonido
- Pulsadores
- Acelerómetro
- Brújula
- Radio
- Pines entrada y salida


Copyright 2026

# Panel LED

La tarjeta consta de 25 LEDs de color rojo, estos se pueden programar de forma independiente. Permite mostrar números, letras e imágenes. El sistema está preparado para desplazarse de forma automática en el supuesto de que no quepan los caracteres a mostrar.

Micro bit position des DEL By Ugopedia [CC BY-SA 4.0 ], from Wikimedia Commons

# Reto Hola Mundo.

Cuando se aprende a programar, se llama “Hola Mundo” al primer programa que se crea. Este código imprime el texto «¡Hola Mundo!» en un dispositivo de visualización.

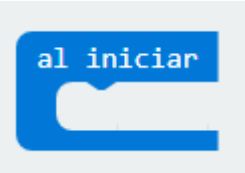
## Objetivo.

Al conectar la micro:BIT, se debe mostrar el texto «¡¡¡Hola Mundo!!!» en el panel LED de la tarjeta.

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

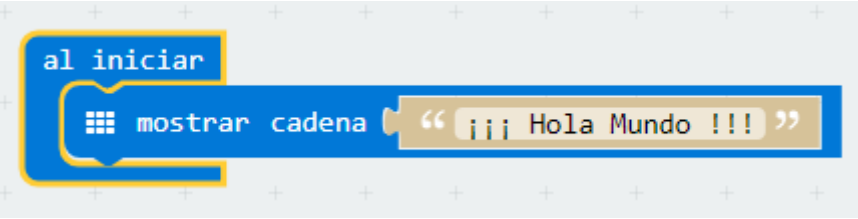
## Descripción del código.

Se propone usar el evento al iniciar, situado en la categoría Básico. Este bloque permite ejecutar el código introducido, cuando se inicia la micro:BIT




A continuación se usará el bloque mostrar cadena, también localizado en la categoría Básico e introducir el texto “¡¡¡ Hola Mundo !!!”. Este bloque permite mostrar caracter a caracter, el texto introducido.

El código quedará de la siguiente manera.




De forma inmediata comenzará a leerse en el panel LED la cadena de texto introducida. El texto no cabe en su totalidad por lo que este se desplazará de forma automática.

## Simulador

Lo bueno que tiene el editor es que permite compartirlo y embeberlo en una página web. Aquí tienes el resultado de embeber el anterior código:

https://makecode.microbit.org/---run?id=_3bPEot71XXJA

## Propuesta.

Modificar el código, para que se muestre el nombre del programador.

# Reto Animando los iconos.

La animación es una técnica que logra crear sensación de movimiento a imágenes estáticas. Se consigue mediante una secuencia de imágenes ordenadas, que al ser mostradas consecutivamente, consiguen generar la ilusión visual de movimiento.

## Objetivo.

Se propone, que al conectar la micro:BIT, se muestre en el panel LED un corazón que late.

https://www.youtube.com/embed/Cf2ov3A-ICE

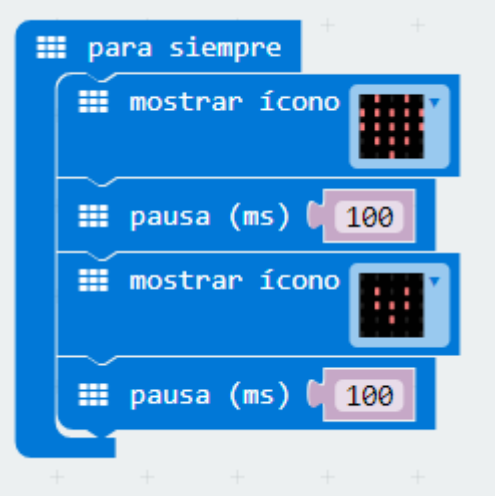
## Descripción del código.

Se usará el evento “para siempre”, situado en la categoría Básico..

Situar en el interior el bloque anterior la instrucción mostrar icono, usar el llamado corazón.

Agregar el bloque pausa ms (100) localizado en la categoría Básico.

Repetir los dos bloques anteriores, pero seleccionando el icono “corazón pequeño”.




De forma inmediata se visualiza en el simulador un corazón que simula latir.

## Simulador

https://makecode.microbit.org/---run?id=_fxMcfc0KHUmd

## Propuesta.

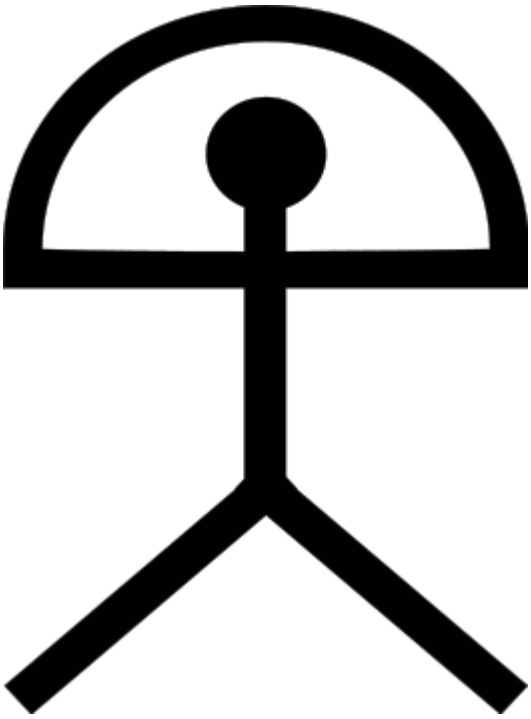
Hacer un código que muestre la animación de una tijera cortando.

# Reto 3 Diseñando iconos.

MakeCode permite diseñar de forma gráfica los iconos a mostrar en el display LED. Para este menester dispone de un bloque específico situado en la categoría Básico. El bloque se llama “mostrar LEDs” y permite al usuario diseñar sus propios iconos marcando de forma independiente los LEDs que se quieren iluminar.

## Objetivo.

Diseñar un icono con forma de Indalo. Se trata de una figura rupestre encontrado en una cueva de la provincia de Almería y que representa una figura humana con los brazos extendidos y un arco sobre sus manos. Actualmente se considera un símbolo de la provincia de Almería.

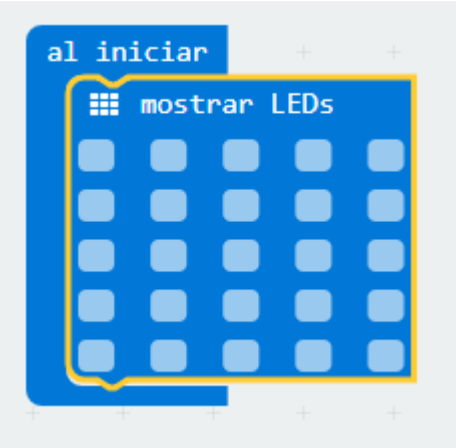


https://www.youtube.com/embed/ag2ulUyaQqg

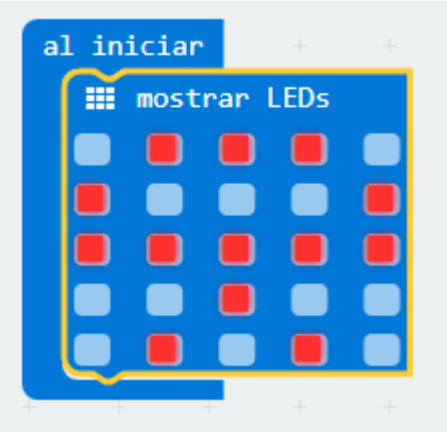
## Descripción del código.

Se usará el evento al iniciar, situado en la categoría Básico..

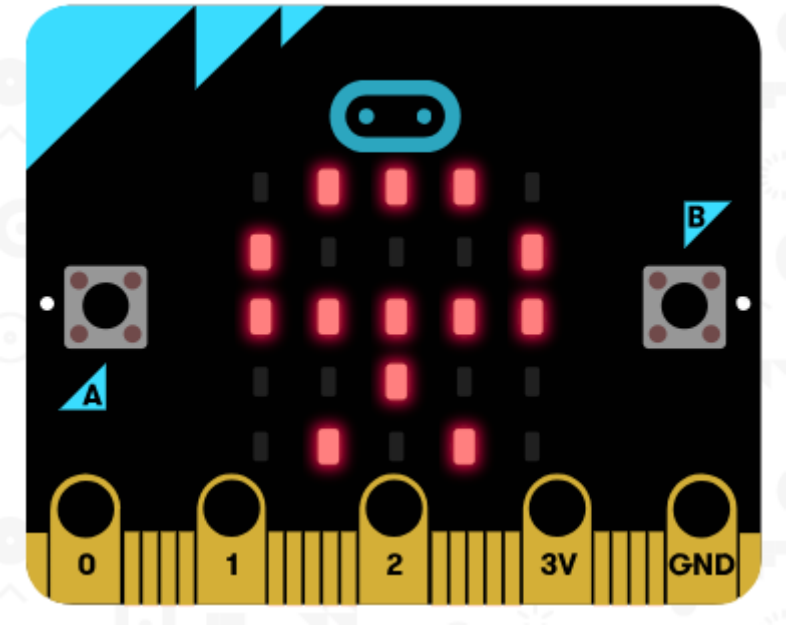
Situar en el interior el bloque anterior, la instrucción “mostrar LEDs”, este se localiza en la categoría Básico. Este bloque permite marcar de forma independiente los LEDs que se quieren encender.




Pulsar sobre los LEDs que se quieren iluminar hasta conseguir el icono deseado.




Deberá aparecer en el simulador el icono con la forma diseñada.



# Propuesta.

Diseñar un código que muestre un logotipo diseñado por el programador.

Copyright 2026

# Jugando con los LED: bloques básicos de código

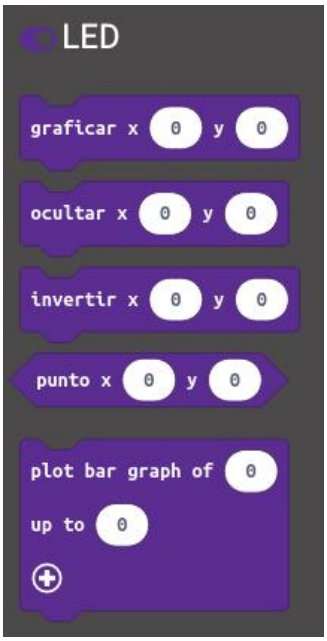
El menú **Básico** contiene los bloques más usados para programar micro:bit. En la página anterior se han visto los bloques de eventos **al iniciar** y **para siempre**. También se han utilizado los comandos **mostrar icono**, **pausa (ms)** y **mostrar cadena**.

El bloque **mostrar LEDs** sirve para dibujar iconos personalizados. Basta con hacer clic en los puntos para dibujar la figura deseada como en el caso del gato de la figura.

El menú contiene además los bloques **borrar la pantalla**, **mostrar flecha** y **mostrar número**, cuyos nombres son lo suficientemente indicativos de sus funciones.

El menú **LED** posibilita el control individual de los 25 LED de la pantalla.

Copyright 2026




**Graficar** enciende el LED de coordenadas x e y. La coordenada 0, 0 corresponde al LED situado en la esquina superior izquierda de la pantalla, mientras que la coordenada 4, 4 es la del LED de la esquina inferior derecha. El LED central es el de coordenadas 2, 2.

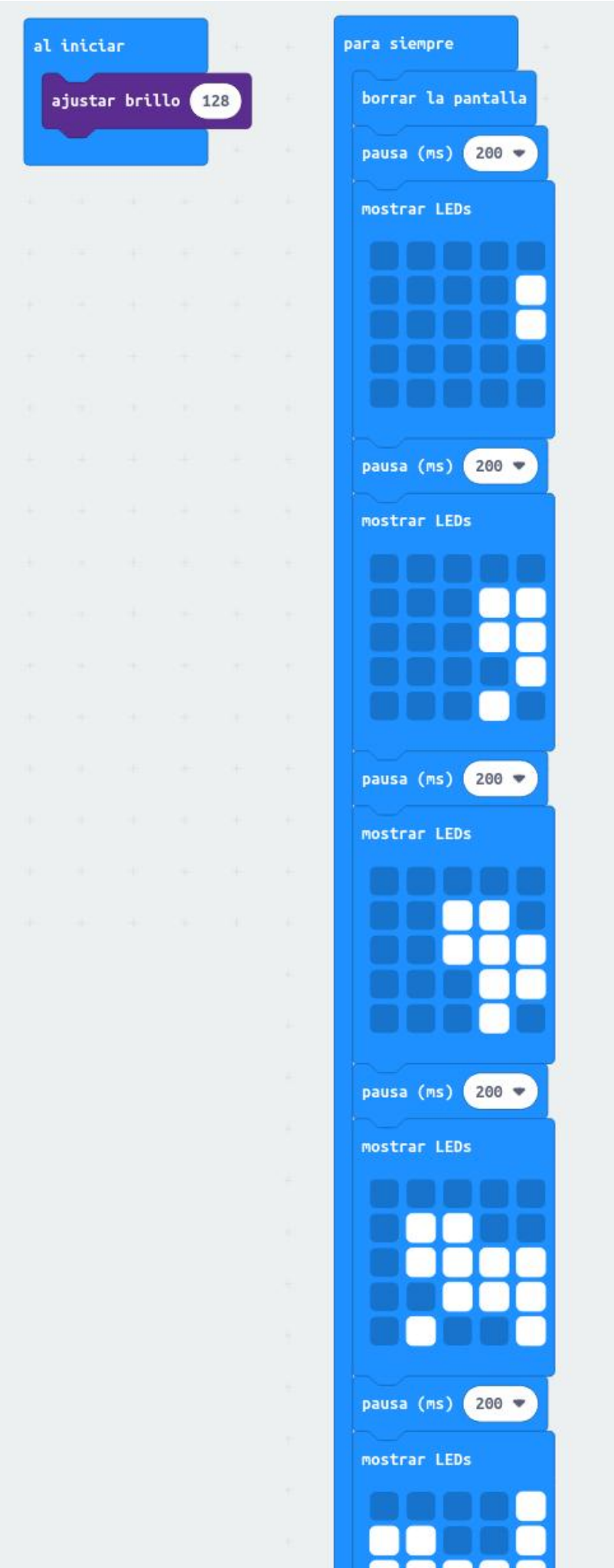
**Ocultar** apaga el LED de coordenadas x e y.

**Invertir** cambia el estado  del LED de coordenadas x e y. Si estaba apagado lo enciende; si estaba encendido lo apaga.

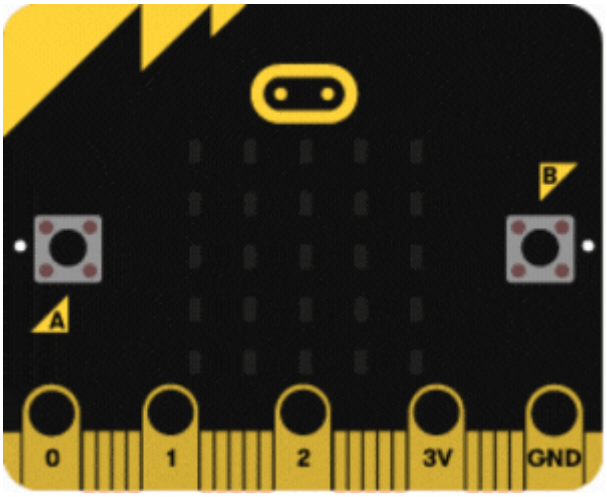
**Punto** informa sobre si el punto de coordenadas x e y está encendido o apagado. Este tipo de bloques se explicará en la sección de variables, lógica y matemáticas.

**Plot bar graph of** crea una barra vertical para representar un parámetro. Resulta un bloque muy útil para visualizar gráficamente las medidas de los sensores de micro:bit. En la sección siguiente se verá con más detalle aplicado a un ejemplo.

El siguiente programa crea una animación con los bloques básicos. Las **animaciones** son una excelente forma de introducir al alumnado de menor edad en los fundamentos de la programación a través de secuencias simples y en las imágenes digitales.


al iniciar
ajustar brillo 128
para siempre
borrar la pantalla
pausa (ms) 200
mostrar LEDs
pausa (ms) 200
mostrar LEDs
pausa (ms) 200
mostrar LEDs
pausa (ms) 200
mostrar LEDs
pausa (ms) 200
mostrar LEDs

Nótese que al inicio se ha ajustado el brillo de la pantalla. La función **ajustar brillo** se encuentra pulsando sobre los tres puntos suspensivos bajo el menú **LED**. El brillo máximo es de 255, por lo que 128 representa un valor medio.



# Sensor temperatura

El sensor de temperatura integrado en la placa detecta la temperatura ambiente en grados Celsius.

CelsiusKelvin

By Celsius_kelvin_estandar_1954.png: Homo logosderivative work: kismalac (Celsius_kelvin_estandar_1954.png) [CC BY-SA 3.0 or GFDL], via Wikimedia Commons

# Reto Termómetro digital.

Un termómetro es un instrumento que sirve para medir la temperatura; el más habitual consiste en un tubo capilar de vidrio cerrado y terminado en un pequeño depósito que contiene una cierta cantidad de mercurio o alcohol, el cual se dilata al aumentar la temperatura o se contrae al disminuir y cuyas variaciones de volumen se leen en una escala graduada.

Los termómetros digitales son aquellos que, valiéndose de dispositivos transductores, utilizan circuitos electrónicos para convertir en números las pequeñas variaciones de tensión obtenidas, mostrando finalmente la temperatura en un visualizador.

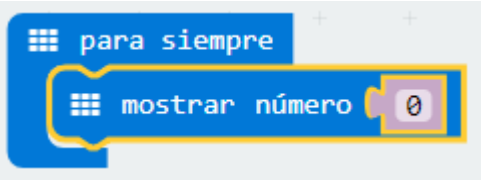
## Objetivo.

Mostrar en el panel LED la temperatura detectada por el sensor de temperatura integrado en la micro:BIT.

https://www.youtube.com/embed/Avl3wo4Rkww

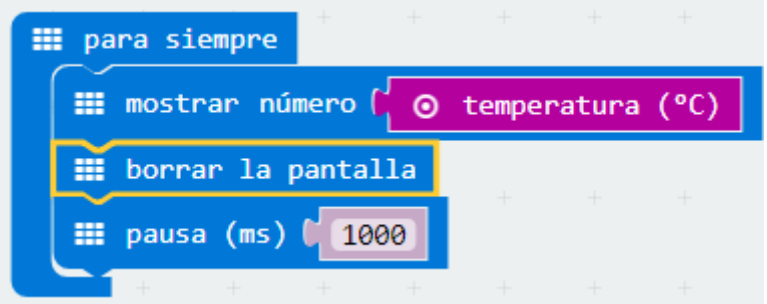
## Descripción del código.

Se propone iniciar el programa usando el evento “para siempre”. Añadir el bloque, localizado en la categoría Básico, mostrar número. Este bloque muestra en la pantalla LED el número introducido, desplazándose si es mayor de 1 cifra.

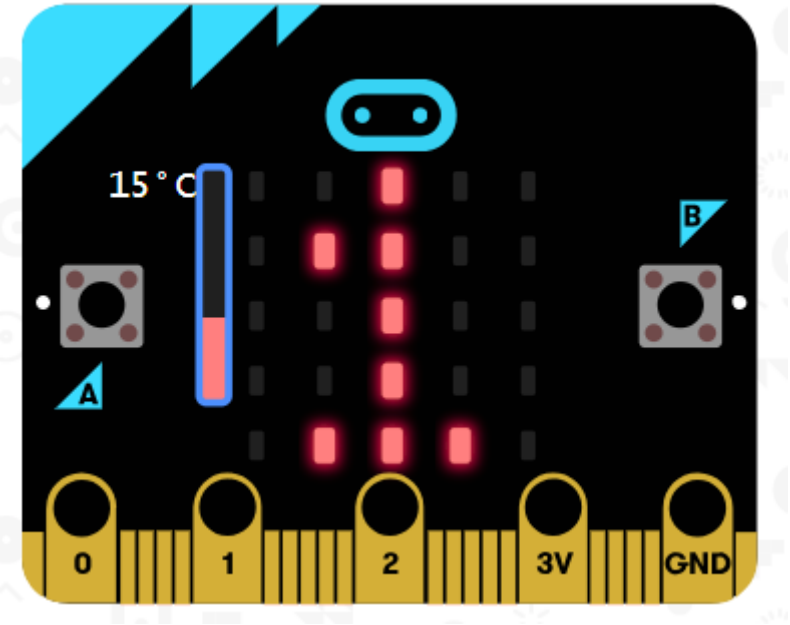



Sustituir el número “0” por el valor obtenido del sensor de temperatura. Este bloque se localiza en la categoría Entrada.

A continuación se borra la pantalla y se introduce una pausa de 1000 ms. Quedando el programa de la siguiente forma:




En el simulador aparecerá una barra vertical que permite modificar la temperatura, siendo la temperatura marcada la mostrada en el panel LED.



## Propuesta.

Hacer un programa que al pulsar el botón A muestra la temperatura en grados celsius y al pulsar el botón B la temperatura mostrada sea en grados Fahrenheit.

# Reto Aviso de placas de hielo.

La mayoría de coches incorporan en el salpicadero, junto a la pantalla que marca la temperatura exterior, un testigo con forma de copo de nieve, que avisa de la posibilidad de que haya placas de hielo en la carretera. Este testigo se suele iluminar cuando la temperatura baja de 3 grados centígrados.

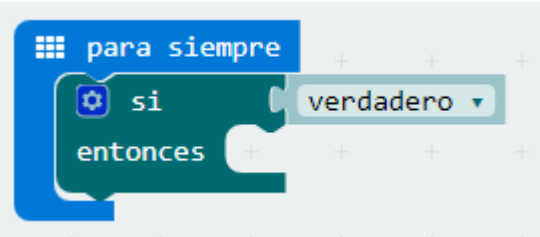
## Objetivo.

Mostrar en el panel LED de la micor:BIT un icono con forma de copo de nieve cuando la temperatura esté por debajo de 3 grados.

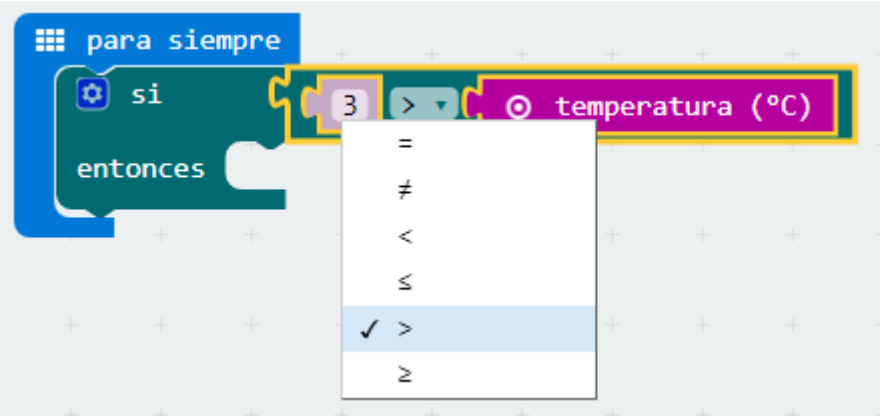
https://www.youtube.com/embed/T8Ely3YaHhg

## Descripción del código.

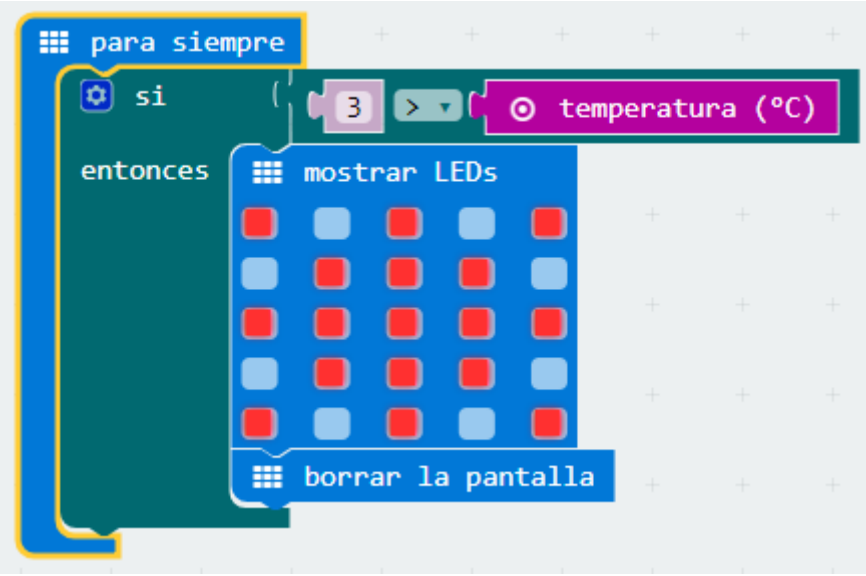
Para iniciar el programa, se propone usar el evento para siempre. Posteriormente se debe añadir el operador lógico si … entonces situado en la sección Lógica. Este bloque verifica si es verdad que se cumple una condición, si es así, ejecuta las instrucciones introducidas.



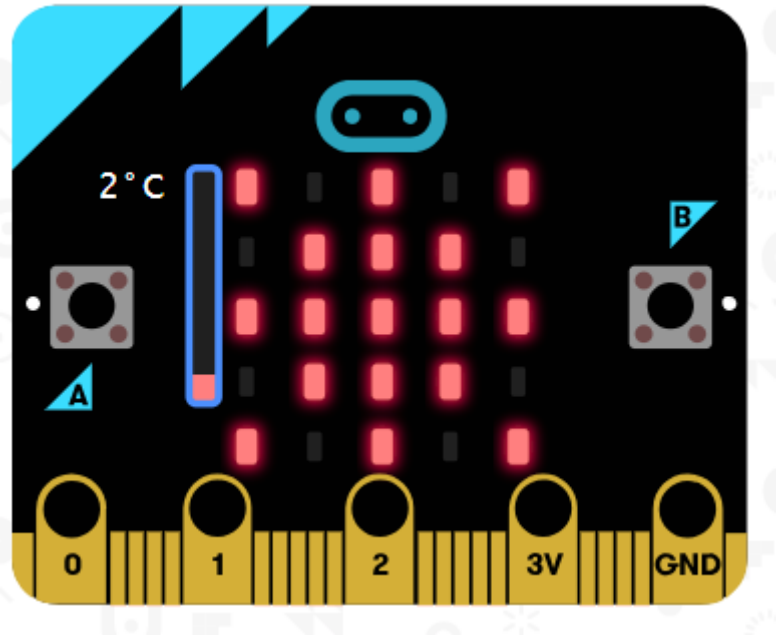

Para comprobar si la temperatura está por debajo de los 3º centígrados se usará el operador que devuelve verdadero siempre que 3 sea mayor que el valor reportado por el sensor de temperatura. El bloque se localiza en la categoría Lógica.




Para mostrar un icono con forma de copo de nieve se usará el bloque mostrar LED, también se debe añadir el bloque borrar la pantalla para que no se quede encendido el panel LED cuando suba la temperatura de 3º centígrados




Una vez terminado el código, aparecerá el símbolo de hielo, cuando la temperatura sea menor de 3º centígrados.




## Propuesta.

Añadir un aviso sonoro cuando la temperatura sea inferior a 3 grados celsius.

# Reto Temperatura óptima de una nevera

La temperatura óptima de un frigorífico es de 7ºC, mientras que la temperatura de un congelador debe estar entorno a los -18ºC. Algunas neveras incorporan un avisador acústico que se activa cuando la temperatura no es la óptima.

## Objetivo.

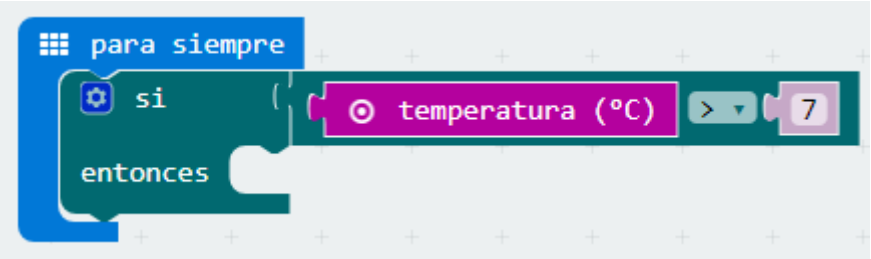
Mostrar en el panel LED de la micro:BIT la temperatura del frigorífico y activar una alarma cuando la temperatura esté por encima de 7ºC

https://www.youtube.com/embed/AJHt2rd805E

## Descripción del código.

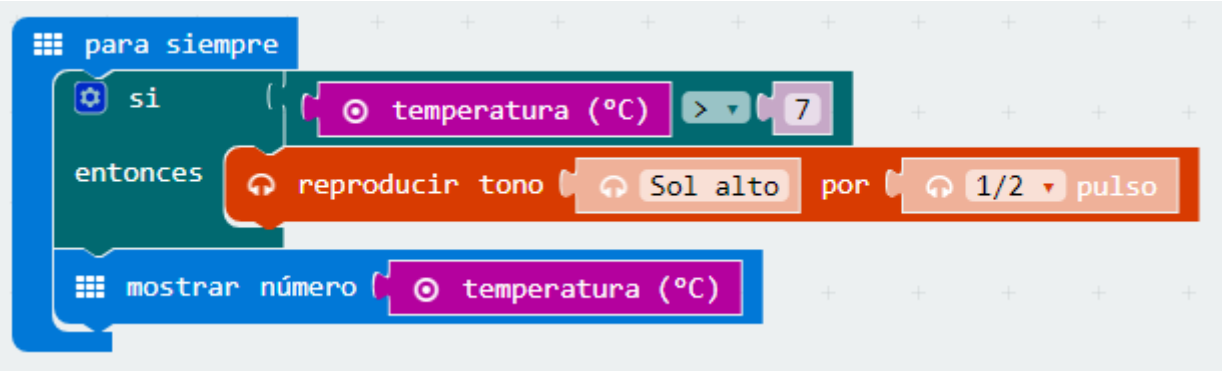
En este reto se incorpora un altavoz como actuador externo. Se conectará uno de los polos al GND y el otro al PIN 0.

Para iniciar el programa se usará el bloque “para siempre”. Dentro se situará el condición “si entonces”. La condición a verificar será: si la temperatura es mayor de 7 grados.




Si el valor es verdadero entonces se reproducirá un tono de aviso.

También se muestra la temperatura medida.




En el simulador se puede variar la temperatura para comprobar el correcto funcionamiento del programa.

## Propuesta.

Modificar el código para monitorizar el congelador.

# Sensor luminosidad y sonido

Los LEDs de la placa micro:BIT también pueden actuar como entrada haciendo que detecten la luz ambiente.

Artist's rendition of the new light-bending metamaterial

By Photo Credit: Keith Drake [Public domain], via Wikimedia Commons

# Reto Interruptor crepuscular.

Cuando la intensidad de la luz cae por debajo de un nivel de iluminación predeterminado y ajustado con anterioridad, el interruptor crepuscular enciende la iluminación. Por el contrario, si la intensidad de la luz es mayor que el nivel prefijado, los interruptores apagan la iluminación.

## Objetivo.

Crear un programa que mida el nivel luminoso existente, para ello usará el sensor de luz de la micro:BIT. Posteriormente se comprobará si este valor es inferior a 50, si es así, se encenderán todos leds de la placa a su máxima intensidad.

https://www.youtube.com/embed/xZu3Cxfzf0s

## Descripción del código.

Se usará el evento para siempre, para iniciar el programa. Se introducirá la condición si … entonces, si no, para comprobar si se está por encima o por debajo del valor teórico tomado como límite en esta ocasión 50. Si el valor es inferior a 50, se fijará la intensidad de los LEDs al máximo usando el bloque ajustar brillo 255. Si el valor es superior a 50 se usará el bloque borrar la pantalla para, de esta forma, apagar todos los LEDs.

El código quedaría de la siguiente forma, se podrá comprobar su funcionamiento variando la intensidad en el simulador de la tarjeta.

## Brillo adaptativo

La matriz de LED de micro:bit es también un sensor que puede captar tanto la luz visible como la invisible. En este último caso se encuentra la luz infrarroja emitida por los mandos a distancia.

Los valores de la variable **nivel de luz** del menú **Entrada** se encuentran comprendidos entre 0 y 1023. Como en el caso del sensor de sonido, estos valores **no tienen correspondencia con ninguna unidad física**, como el lux.

> Para poder obtener buenas medidas de luz, la matriz de LED debe estar descubierta. Las fundas para micro:bit, especialmente las de color oscuro, pueden alterar las medidas del sensor de luz.

De acuerdo con Martínez de Carvajal (2019), el sensor de luz arrojará aproximadamente las siguientes medidas:

- 0 en condiciones de oscuridad absoluta.
- 100 en una habitación con luz artificial.
- 400 en un día muy nublado.
- 1023 a pleno sol.

Podemos retomar el proyecto del sonómetro y acompañarlo de una pantalla con brillo adaptativo, como la de los teléfonos móviles. La pantalla debe iluminarse con más brillo cuanto mayor sea la luz ambiental.

El evento **al presionarse el botón A** medirá el nivel de sonido y lo mostrará gráficamente con la ya conocida barra.

Usaremos un evento de tiempo **cada ms** para medir la luz ambiental cada segundo y así poder ajustar el brillo de pantalla al nivel de luz medido. De esta forma, a mayor luz ambiental, mayor brillo de pantalla. Como el brillo máximo es de 255 y el nivel de luz máximo medible es de 1023, dividiremos la lectura del nivel de luz entre 4 usando un bloque aritmético **/** del menú **Matemática**

.

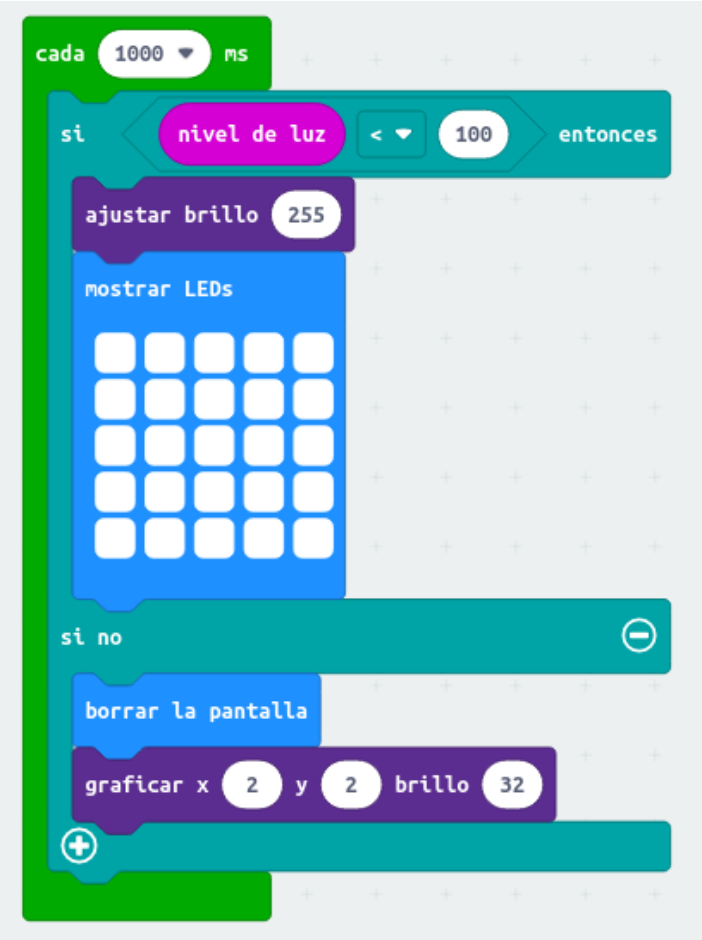
## Luz crepuscular

Vamos a dar solución a un reto propuesto por Muñoz (2022) programando un automatismo que encienda la pantalla de **LED** cuando la luz ambiental caiga por debajo de un cierto valor, por ejemplo 100.

Todos los algoritmos programados hasta este momento se ejecutan linealmente desde el primer bloque colocado en la parte superior hasta el último situado abajo. Sin embargo, para solucionar el reto de la luz crepuscular necesitaremos que el flujo del programa siga caminos distintos según se cumpla o no una determinada condición.

Empezaremos evaluando la luz ambiental cada segundo mediante un evento de tiempo **cada ms**. Esto significa que el automatismo tendrá un tiempo de reacción de un segundo.

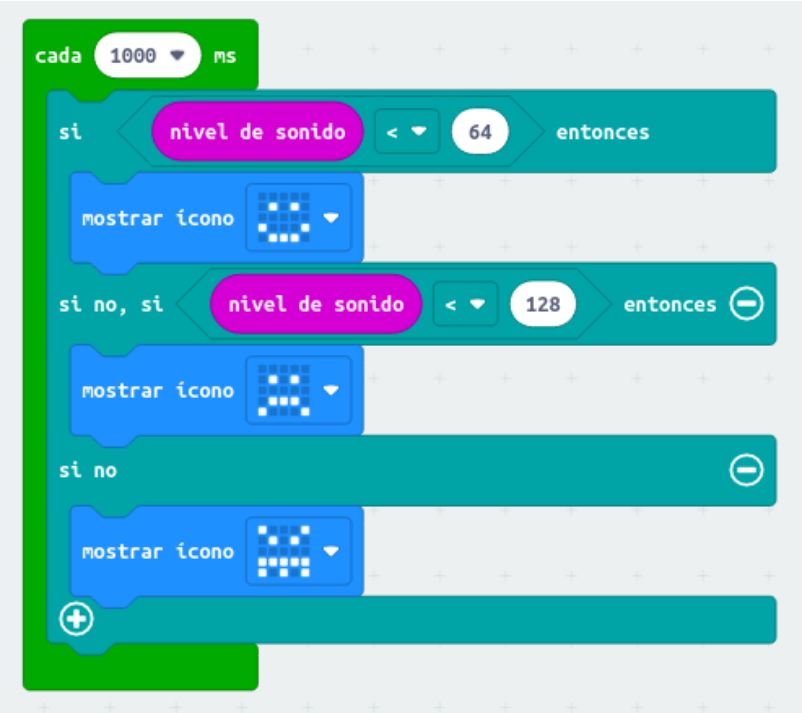
Dentro del evento de tiempo, usaremos la estructura **si entonces si no**, disponible en el menú **Lógica** , para encender con el máximo brillo la luz de la matriz de **LED** si la luz ambiental es inferior a 100 o para apagar la matriz en caso contrario. Tras el apagado de la pantalla dejaremos un pequeño punto en el centro iluminado a baja intensidad como piloto de funcionamiento.




El bloque comparador **<** (menor que) usado dentro de la estructura **si entonces** también se encuentra dentro del menú **Lógica**.

## Semáforo del ruido en el aula

Make Code permite construir sentencias condicionales más complejas. El código del semáforo del ruido emplea un bloque **si entonces si no, si entonces si no**. Para construir este bloque se ha utilizado la estructura **si entonces si no** del ejemplo anterior y se ha añadido una condición suplementaria pulsando sobre el pequeño círculo con el símbolo +. De igual manera, pulsando sobre los círculos con los signos - podremos eliminar condiciones.

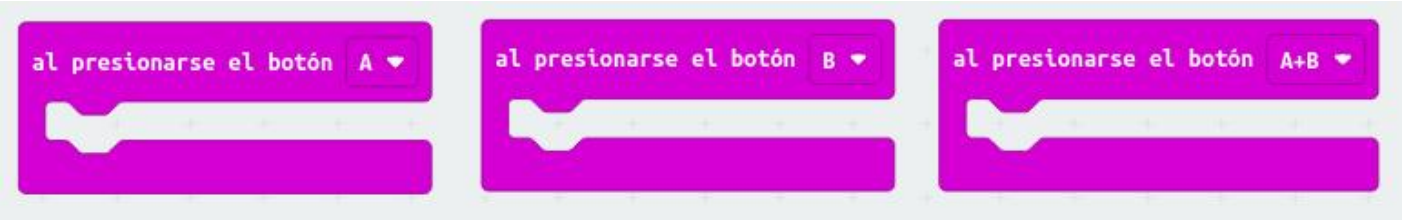



Así pues, si el nivel de sonido es inferior a 64, la pantalla mostará un icono sonriente. En caso contrario, si el nivel de sonido es alto pero todavía inferior a 128, se mostrará una cara triste. FInalmente, si el sonido es igual a superior a 128, se mostrará una cara enfadada.

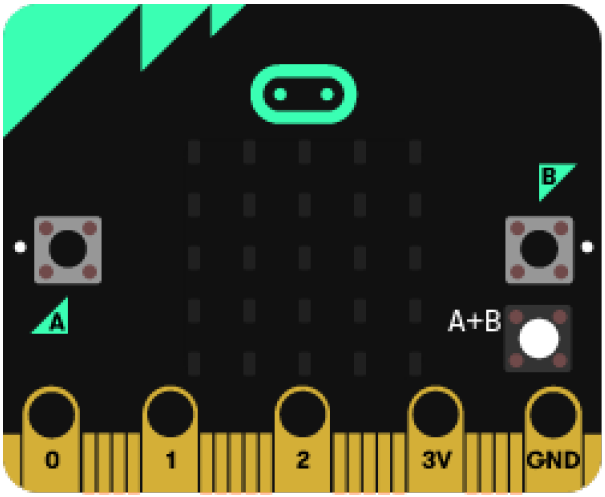
# Pulsadores

## Pulsadores mecánicos

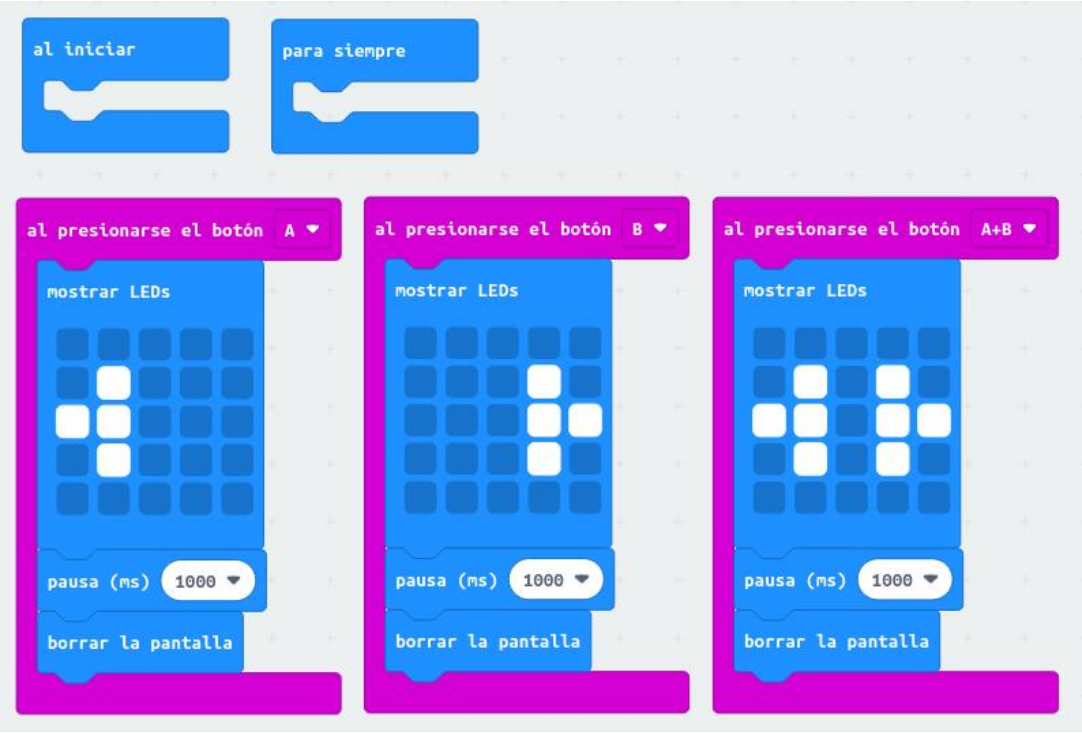
El menú **Entradas** contiene los bloques que permiten captar datos de los sensores integrados en micro:bit. Para que la placa lleve a cabo acciones al pulsar los botones **A**, **B** ó **A** y **B** a la vez, disponemos del bloque de eventos denominado **al presionarse el botón**:




Al arrastrar el bloque anterior desde la caja de herramientas hasta el área de programación y seleccionar la opción de pulsación **A+B**, la imagen de la tarjeta cambia para mostrar la posibilidad de pulsar con el ratón sobre un nuevo botón virtual que simula la pulsación simultánea de los botones A y B.



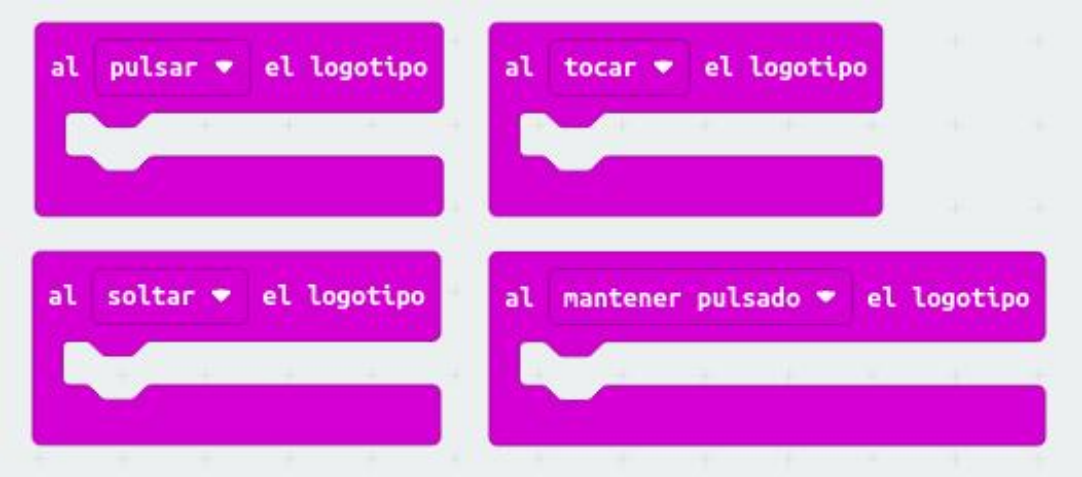

Para que micro:bit realice alguna acción específica al pulsar un botón, bastará con introducir los bloques de código necesarios dentro de un bloque de evento de pulsación. El siguiente programa mostrará una flecha apuntando hacia la izquierda al pulsar el botón **A**, una flecha apuntando hacia la derecha al pulsar el botón **B** y dos flechas al pulsar los botones **A+B**. Transcurrido 1 s, se borrará la pantalla.




Los eventos **al iniciar** y **para siempre** han quedado vacíos simplemente porque no hemos previsto que se lleve a cabo ninguna acción inicial ni permanente.

## Pulsador táctil

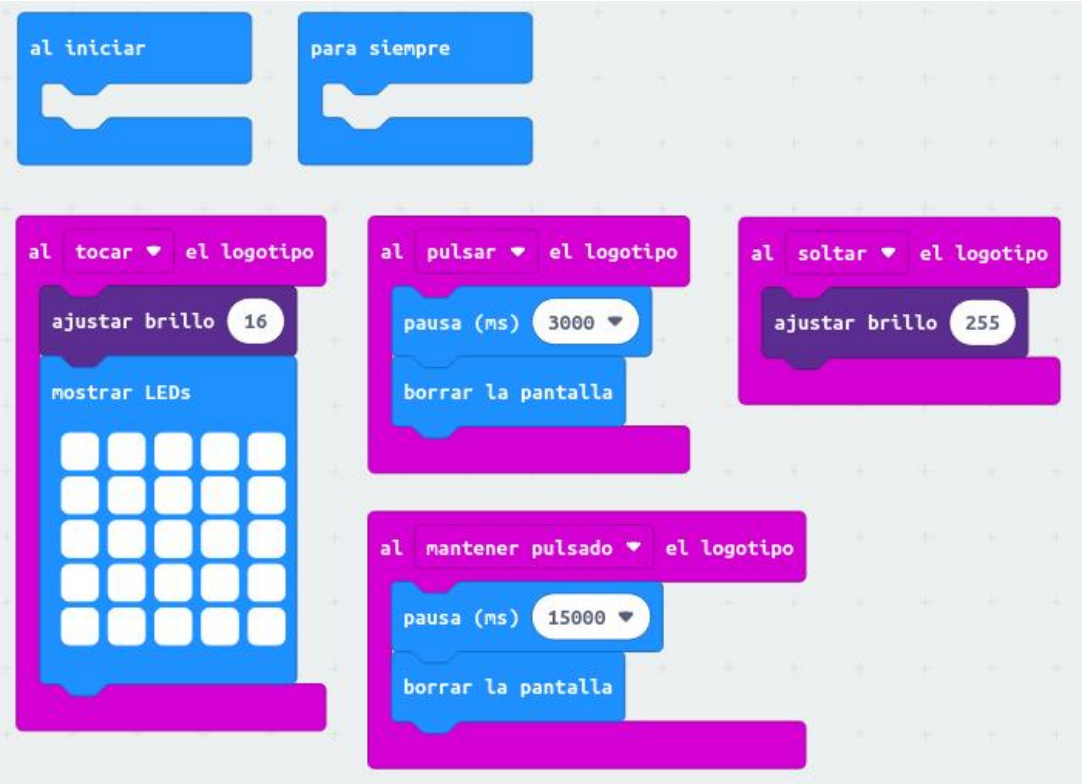
Desde la versión 2 de la tarjeta el **logo** situado en el anverso es un **sensor táctil**. El menú **Entradas** dispone de un bloque específico que puede ajustarse mediante un menú desplegable para capturar el evento deseado.




Como puede verse, los eventos asociados al logo son **pulsar**, **tocar**, **soltar** y **mantener pulsado**. **Cada vez que se toque el logo se producirán tres eventos:**

- Evento **al tocar el logotipo** en el instante en que se toca el logo.
- Evento **al soltar el logotipo** justo en el momento en el que se levanta el dedo del logo.
- Evento **al pulsar el logotipo** justo en el momento en el que se levanta el dedo del logo siempre que la pulsación haya sido corta.
- Evento **al mantener pulsado el logotipo** justo en el momento en el que se levanta el dedo del logo siempre que la pulsación haya sido larga.

El siguiente programa simula una **luz temporizada**.




Mada más tocar el logo, el evento **al tocar el logotipo** hará brillar la pantalla con una luz tenue. Al levantar el dedo, el evento **al soltar el logotipo** iluminará la pantalla a plena potencia. Si la pulsación ha sido corta, el evento **al pulsar el logotipo** mantendrá la luz encendida durante 3 s. Si la pulsación ha sido larga, el evento **al mantener pulsado el logotipo** mantendrá la luz encendida durante 15 s.

# Reto ¿Qué botón has pulsado?

A veces puede interesar usar un pulsador para iniciar, pausar o detener un código o subrutina. Algunos robots incorporan botoneras para esta finalidad.

## Objetivo.

Crear un programa que muestre en pantalla la letra del pulsador que se ha accionado.

Pruébalo aquí:

https://makecode.microbit.org/---run?id=_ibiDd0EKA15r

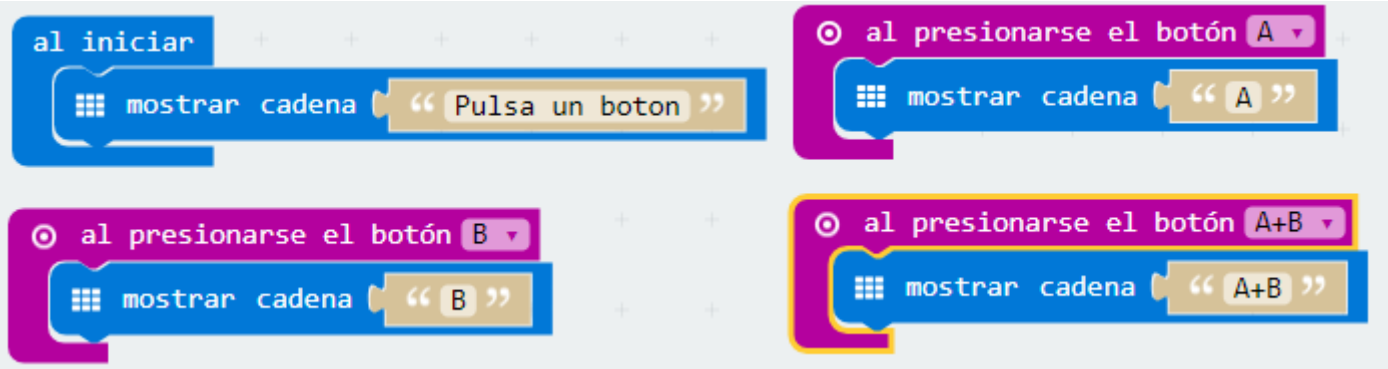
## Descripción del código.

Se usará el bloque al iniciar para que al principio se muestre el texto “Pulsa un botón”. La micro:BIT no muestra caracteres acentuados, por lo que dejará un hueco si se pone la tilde en la palabra botón.

Posteriormente se usará el evento al presionar el botón A y se introducirá el bloque mostrar cadena donde se sustituirá el texto por defecto por la letra “A”.

Esto mismo se repetirá cuando se pulsa el botón B y la pulsación combinada de A+B.

El código quedaría de la siguiente forma:




## Propuesta.

Crear una caja de música. Al pulsar el botón A haga hará sonar una melodía. El bloque necesario para hacer sonar una melodía se encuentra en la categoría Música.

# Acelerómetro

El acelerómetro mide la aceleración de tu micro:bit. Se activa cuando tu placa se mueve y también puede detectar otras acciones como agitar, girar y hasta soltar tu micro:bit en caída libre!

Micro bit 3 axes de l accéléromètre

By Ugopedia [CC BY-SA 4.0 ], from Wikimedia Commons

# Reto Dado electrónico.

Un dado es un objeto cúbico usado en juegos de azar. En las caras del cubo aparecen puntos que representan distintos números.

Se propone crear un dado electrónico que, usando el panel LED, muestre el resultado de obtener un número al azar entre 1 y 6.

## Objetivo.

Tras agitar la micro:BIT, se deberá calcular un número aleatorio cuyo valor esté entre 1 y 6. Posteriormente, se mostrará en el panel LED el valor obtenido.

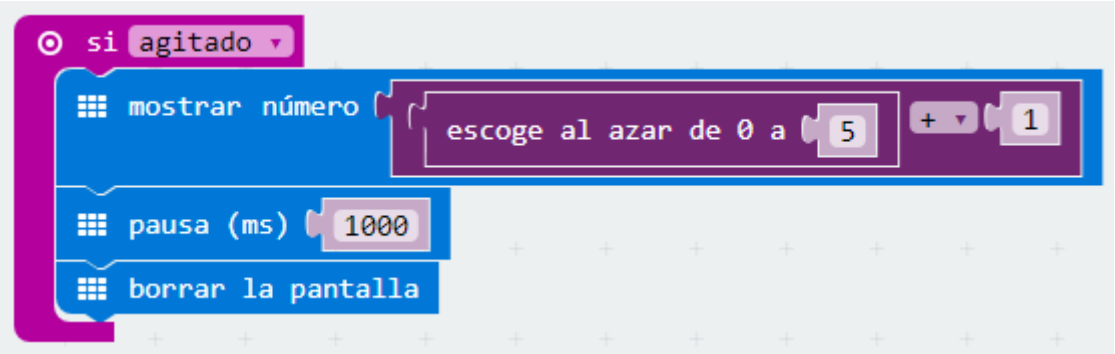
Pruébalo ¡¡lo puedes sacudir pulsando en *Shake*!! este simulador es genial !!

https://makecode.microbit.org/---run?id=_A1Tcyy1zLRgP

## Descripción del código.

Como evento de inicio del programa se usará el bloque si agitado, se mostrará un número escogido al azar entre 0 y 5, al que se le sumará 1. Se suma 1 dado que el bloque escoge al azar, tiene como valor fijo de inicio el 0.

Tras una pausa de 1 segundo se borra la pantalla, quedando a la espera de que se vuelva agitar la micro:BIT.




## Propuesta.

Diseñar un programa que permita jugar a la ruleta además se deberá indicar si es rojo o negro o si gana la banca.

# Reto Cara o cruz.

El juego de cara o cruz es un juego de azar en el que se emplea una moneda. Gana quien acierte qué lado de la moneda (de los dos posibles) caerá cara arriba. Cada uno de los dos lados tiene un nombre distintivo y son mencionados como opciones para ser elegidas por los participantes. Por lo tanto, los participantes pueden ser dos equipos o dos personas y así cada bando tiene el 50% de probabilidades de acertar.

Este juego es frecuentemente empleado como un mecanismo para tomar decisiones o sortear, ya sea entre amigos o en forma informal, pero también es completamente aceptado en ciertas circunstancias, por ejemplo al inicio de un partido de fútbol profesional.

## Objetivo.

La micro:BIT se situará con el panel LED hacia abajo. Al posicionarlo hacia arriba, se mostrará una cara o una cruz, ganando el que haya escogido la opción mostrada.

## Descripción del código.

En primer lugar se creará una variable que se puede llamar “azar”.

Se usará como evento de inicio el bloque si pantalla hacia arriba. A continuación se asigna a la variable “azar”, un valor al azar entre 0 y 1.

Usando el operador lógico si entonces si no se elige como opción es la ganadora. Se asigna la carita feliz si la variable “azar” toma un valor de cero y en caso contrario ganará la cruz. Se mostrará la opción ganadora usando el bloque mostrar icono.

Tras una pausa de un segundo se borra la pantalla.

Para volver a repetir el proceso se deberá poner la micro:BIT con el panel hacia abajo y al volver a poner el panel hacia arriba se repetirá el código programado

.

Para ver el funcionamiento del programa se recomienda volcar el código a una micro:BIT. En el simulador, el proceso no es tan intuitivo.

## Propuesta.

Hacer un programa que permita jugar a “Pares o Nones” contra la micro:BIT

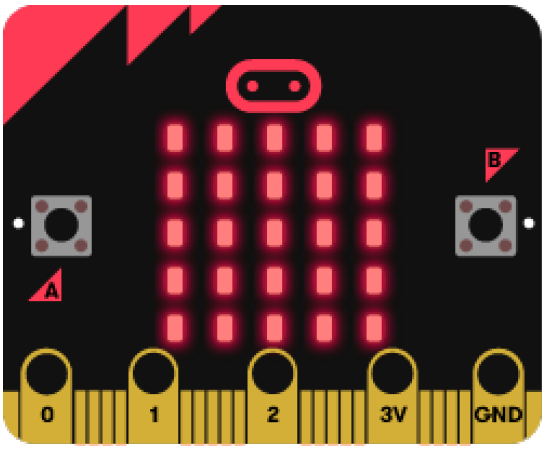
# Brújula

La brújula detecta el campo magnético terrestre por lo que se puede saber en qué dirección está orientada la micro:bit. (Necesita ser calibrada para asegurar un resultado preciso.)

## Calibrar la brújula

La primera vez que grabamos un programa sale este mensaje por la pantalla:

**Tilt to Fill Screen** *Inclinar para llenar la pantalla*




Luego lo que tenemos que hacer **después** de que aparezca el texto anterior, sale un punto que irá rellenando la pantalla a medida que lo vas inclinando.

Objetivo: que se llene toda la pantalla de leds.

Con un vídeo se ve mejor

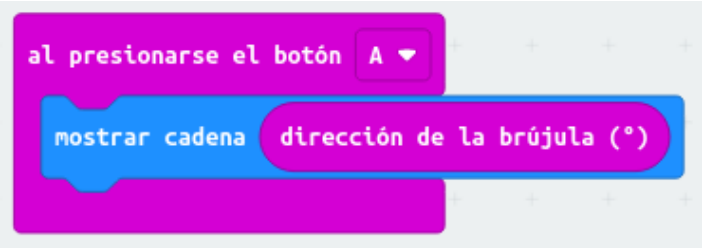
https://www.youtube.com/embed/jvK0ql_u5F0

*Pagina oficial de calibración.*

## Brújula numérica

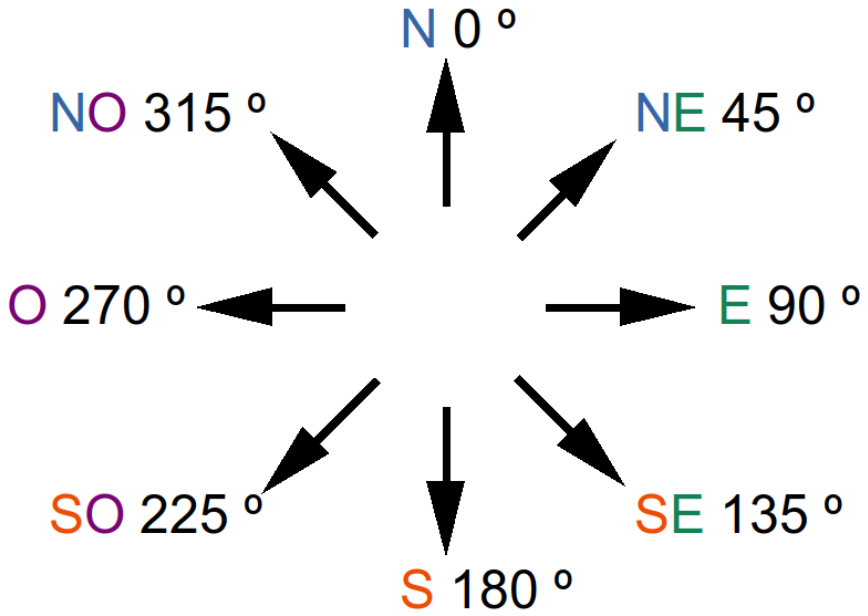
El sensor de campo magnético integrado en micro:bit también puede usarse para medir el **campo magnético terrestre**. De esta forma es posible determinar la dirección del norte magnético y programar así una brújula.

De acuerdo con la esencia de micro:bit bastan unas pocas líneas de código para realizar una brújula digital. El código mostrado a continuación indica en pantalla **la dirección a la que apunta el logotipo** cada vez que se presiona el botón A.




Como el resto de bloques de sensores, el bloque **dirección de la brújula (º)** se encuentra disponible en el menú **Entrada**.

La dirección será mostrada en **grados sexagesimales**. Conviene recordar que el valor de 0º corresponde al norte y el de 180º al sur. La siguiente rosa de los vientos puede servir para clarificar la correspondencia entre los puntos cardinales y los rumbos asociados:




Por ejemplo, y de acuerdo con la rosa de los vientos, una lectura de 340º indicará que el logotipo de micro:bit apunta en dirección casi norte, concretamente nornoroeste.

La **brújula funcionará mejor en exteriores**, ya que el sensor magnético puede ver alteradas sus lecturas por la presencia de masas metálicas en el interior de los edificios, como las de los electrodomésticos o las armaduras de pilares y forjados.

Es posible que al encender micro:bit, aparezca un mensaje en inglés solicitando agitar la tarjeta para **calibrar el sensor**. Esta cuestión se trata en el apartado siguiente.

# Reto Conocer la orientación

El norte magnético es la dirección que señala la aguja imantada de una brújula, dirección que no coincide con la del Polo Norte geográfico. Se puede conocer la orientación si se conocen los grados de desviación respecto el Norte Magnético.

## Objetivo.

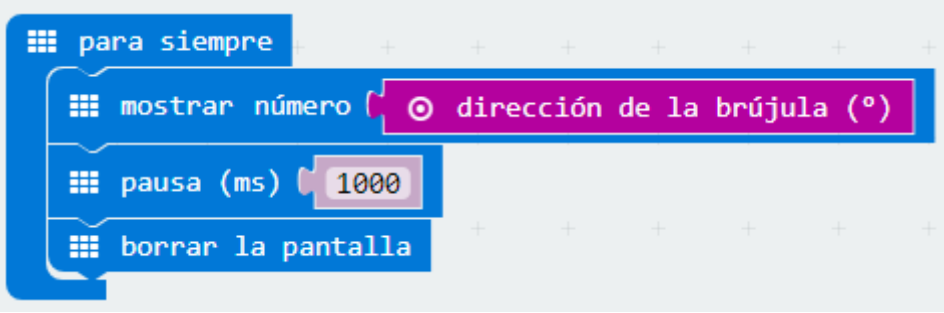
Tras calibrar la brújula, nos dirá la desviación en grados respecto al Norte Magnético de la micro:BIT.

## Descripción del código.

Usar el evento para siempre, para iniciar el programa.

Mostrar el valor del sensor dirección de la brújula (º) que ofrece el ángulo de desviación respecto el Norte Magnético.

Esperar 1 segundo y borrar la pantalla.




### Brújula analógica

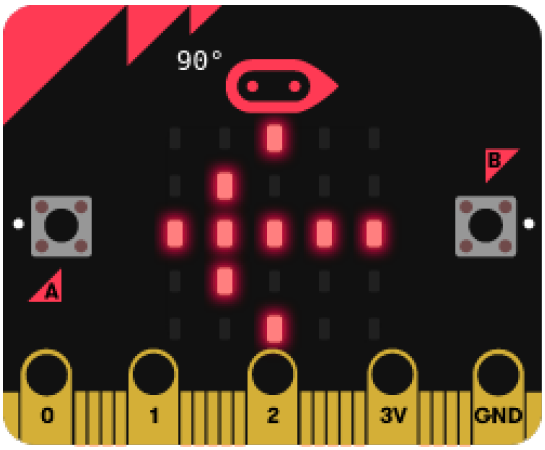
Vamos a completar la funcionalidad de la brújula añadiendo un cursor que marque continuamente la dirección del norte. Usaremos un **evento de tiempo** para leer el sensor y actualizar el dibujo del cursor cada medio segundo.

Dentro del evento de tiempo, el código dibujará una flecha hacia arriba cuando la dirección de micro:bit sea mayor que 315º o menor que 45º, es decir, cuando el logotipo apunte aproximadamente al norte. Los bloques condicionales y los comparadores usados en el código se

encuentran en el menú **Lógica**.

Sin embargo, este programa sólo mostrará la flecha cuando el logo apunte al norte, borrando la pantalla en caso contrario. Para que haya una flecha en pantalla apuntando permanentemente al norte, habrá que pulsar sobre el signo **+** para añadir líneas al bloque condicional.

> Nótese que cuando el logo de micro:bit apunta al este, dirección de la brújula (º) < 135, hay que dibujar la flecha apuntando a la izquierda (flecha oeste), y que cuando el logo apunta al oeste, hay que mostrar una flecha apuntando a la derecha (flecha este). Sólo de esta forma la flecha apuntará siempre al norte.



## Reto Aviso sonoro de orientación Norte.

Cuando se necesita orientarse con un mapa, lo que primero que hay que hacer, es situar el mapa con orientación norte. Para ellos se debe usar una brújula.

**Objetivo**.

Crear un programa que mediante un aviso acústico se advierta de que la micro:BIT está orientada al Norte.

**Descripción del código.**

Usar el evento para siempre, para iniciar el programa.

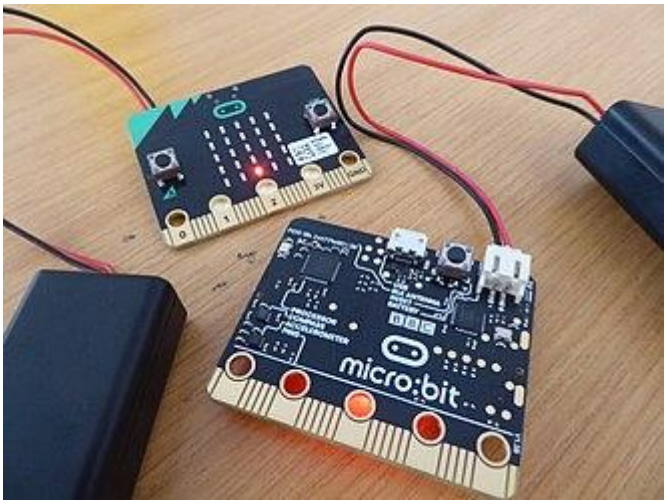
Crear una variable donde se almacenará el valor del sensor obtenido por el bloque dirección de la brújula (º).

Se introduce el condicional si entonces si no para comprobar en qué intervalo se encuentra el valor obtenido por el sensor.

Si el valor se encuentra entre 315º y 45º la micro:BIT se encuentra orientada al Norte, por lo que se mostrará una N y se reproducirá una nota. Si no se cumple la hipótesis, se borra la pantalla.

# Radio

La radio te permite comunicar tu micro:bit con otras micro:bit. Por ejemplo, puedes conectar todas las tarjetas dentro de un aula a una misma emisora, usarla para enviar mensajes entre ellas y mucho más!.




*By Libraries Taskforce (Micro:bits) [CC BY 2.0 ], via Wikimedia Commons*

# Reto Sensor inalámbrico.

En algunas situaciones se puede dar el caso de que no se pueda tener conectado el sensor con la unidad receptora de datos mediante cable. Cuando esta situación se presenta una opción es usar sensores inalámbricos.

Usar sensores inalámbricos aporta numerosas ventajas. Entre ellas destacan la seguridad, la centralización de la monitorización y el ahorro de costes por no tener que cablear la instalación.

Un sensor inalámbrico está compuesto por el sistema recolector de datos, un sistema de emisión de datos y una fuente de alimentación. Lo más habitual es que la transmisión de datos sea mediante radiofrecuencia. Por otro lado, se encuentra un sistema receptor de datos conectado al sistema de almacenamiento y procesado.

## Objetivo.

Diseñar un sistema inalámbrico de recolección de datos. La micro:BIT emisora se situará en el interior de una nevera y emitirá la temperatura. La unidad exterior mostrará el valor de la temperatura.

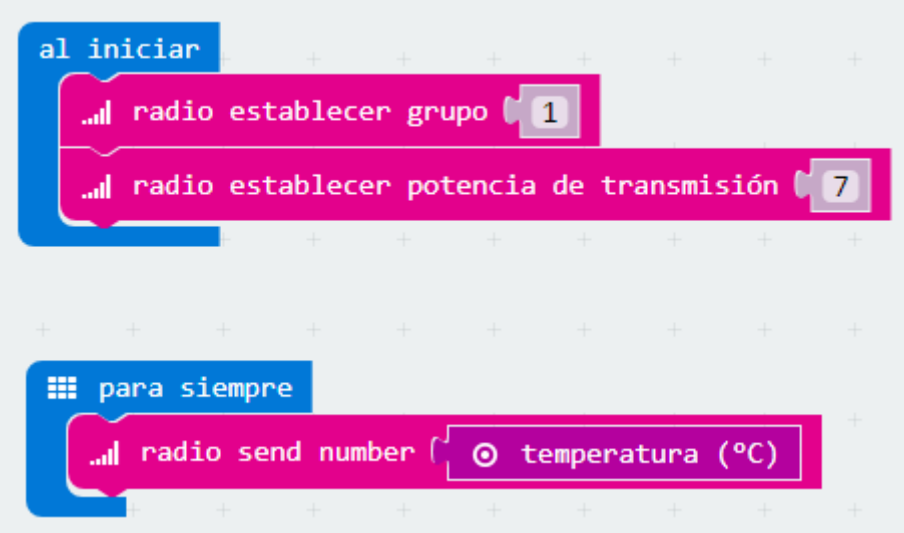
## Descripción del código.

Se deberán crear dos códigos, uno para la micro:BIT emisora y otro para la receptora.

Para que las tarjetas se puedan comunicar entre sí, ambas deben estar en el mismo grupo de comunicación. También se debe emitir con la máxima potencia para evitar la atenuación que sufre la tarjeta emisora al encontrarse dentro de la nevera. Por este motivo, se deberá usar el bloque radio establecer potencia de transmisión situado en la categoría Radio y usar el máximo valor que es 7. Se recomienda usar el evento al iniciar para establecer el grupo de comunicación y asignar la potencia de transmisión.

micro:BIT emisora:

Tras establecer el grupo de emisión en 1 y poner la potencia de emisión al máximo, se enviará el valor de la temperatura mediante el bloque radio send number.




micro:BIT receptora:

La cadena enviada se guarda en la micro:BIT receptora en una variable que por defecto se llama “receivedNumber”. Se usará el bloque mostrar número, para visualizar la temperatura medida en la nevera. Tras un segundo se apagará el panel LED. No olvidar establecer al inicio el grupo de comunicación.

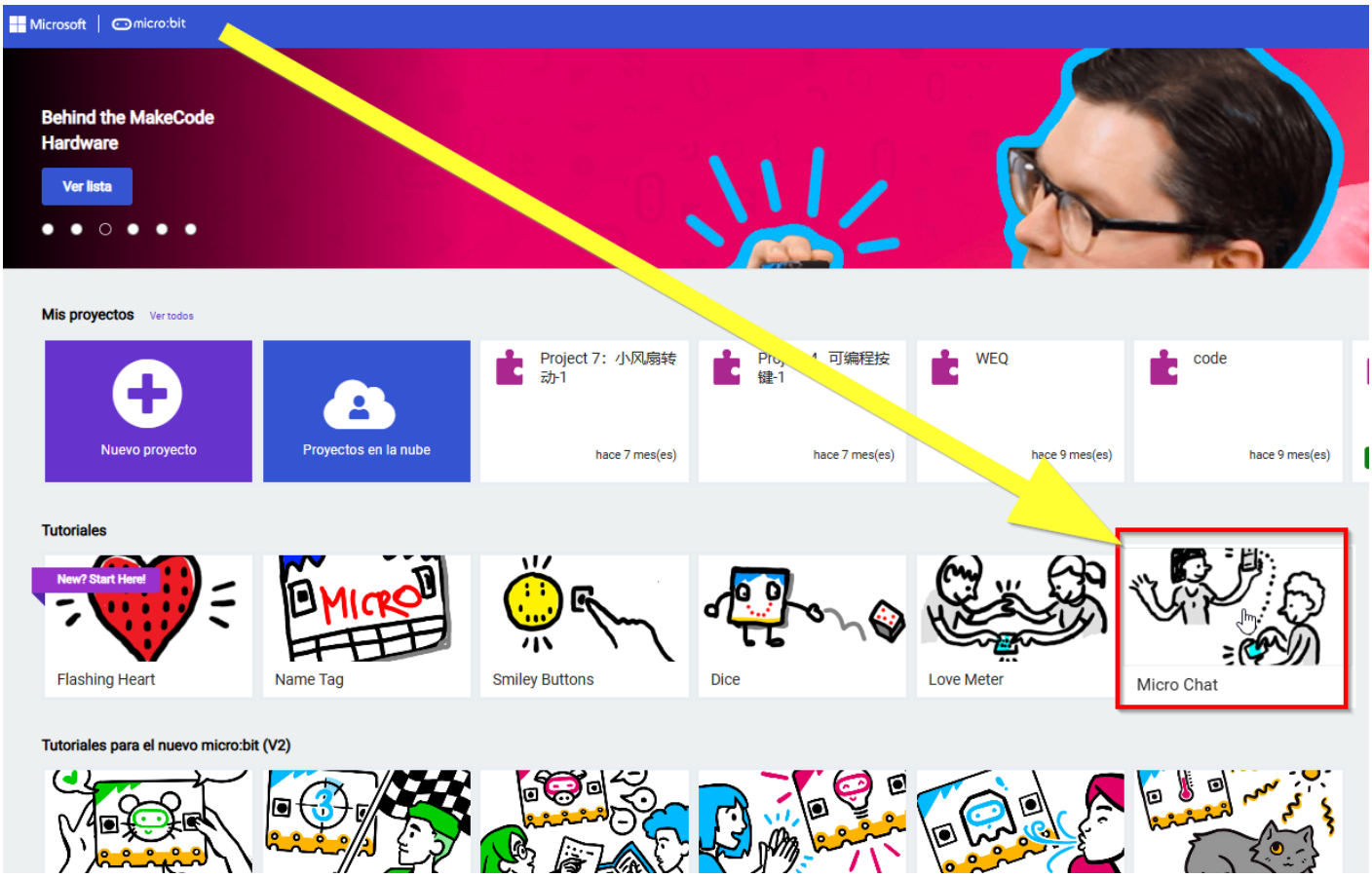
El código de la tarjeta receptora quedaría de la siguiente forma:

# Propuesta.

Se propone mostrar una animación en el panel de la micro:BIT emisora, que confirme que el programa está en marcha.

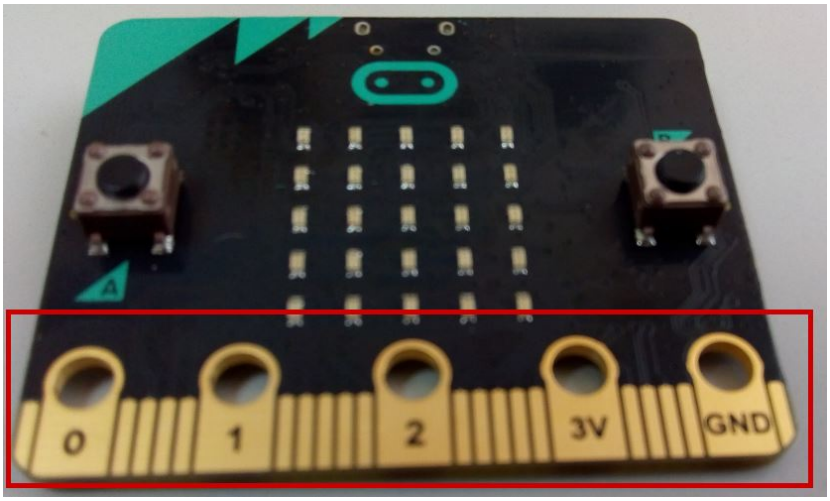
## Reto

Realiza el programa que propone makecode https://makecode.microbit.org/ : Que al pulsar el botón A de la microbit, se transmita por el grupo tu nombre y los demás lo reciban y lo muestra por pantalla.
IDEAL PARA PRESENTACIONES DE GRUPOS !!!!!


Microsoft
micro:bit
Behind the MakeCode Hardware
Ver lista
Mis proyectos
Ver todos
Nuevo proyecto
Proyectos en la nube
Project 7：小风扇转动-1
hace 7 mes(es)
hace 7 mes(es)
WEQ
hace 9 mes(es)
code
hace 9 mes(es)
Tutoriales
New? Start Here!
Flashing Heart
Name Tag
Smiley Buttons
Dice
Love Meter
Micro Chat
Tutoriales para el nuevo micro:bit (V2)

# Pines entrada y salida

La micro:BIT dispone de 25 conectores situados en el borde inferior . A través de ellos se podrán programar motores, LEDs o cualquier otro componente o sensor externo.




~ Biografía del autor ~

## Reto ¿Hay continuidad?

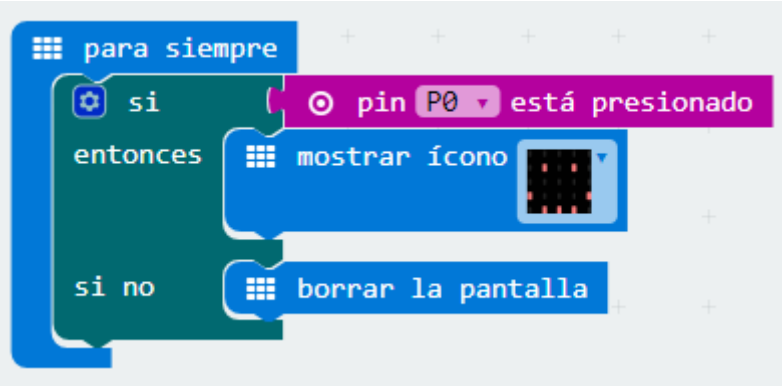
En algunas ocasiones interesa conocer si un cable tiene continuidad en toda su longitud. Para comprobar la continuidad se suele usar un multímetro, midiendo la resistencia del conductor. Una lectura de resistencia infinita indicaría una interrupción en alguna parte del interior del cable.

### Objetivo.

Diseñar un programa que permita comprobar si un cable tiene continuidad. Si no está cortado, mostrar una cara feliz.

### Descripción del código.

Se usará el bloque pin (P0) está presionado de la categoría Entrada para comprobar la continuidad. Se propone emplear el evento para siempre y usar un condicional que verifique si el P0 está presionado. Si se cumple la condición se muestra una carita feliz, en caso contrario, el panel permanece apagado. El código podría quedar de la siguiente forma:




Para comprobar el funcionamiento se puede usar un cable con bananas conectados al GND, si se conecta con el P0 y el cable tiene continuidad, deberá aparecer una carita feliz en el panel LED.

## Propuesta.

Modificar el código para que cuando el cable a comprobar tenga continuidad haga sonar la melodía “jump up” y si no tiene continuidad que se emita la melodía “jump down”.

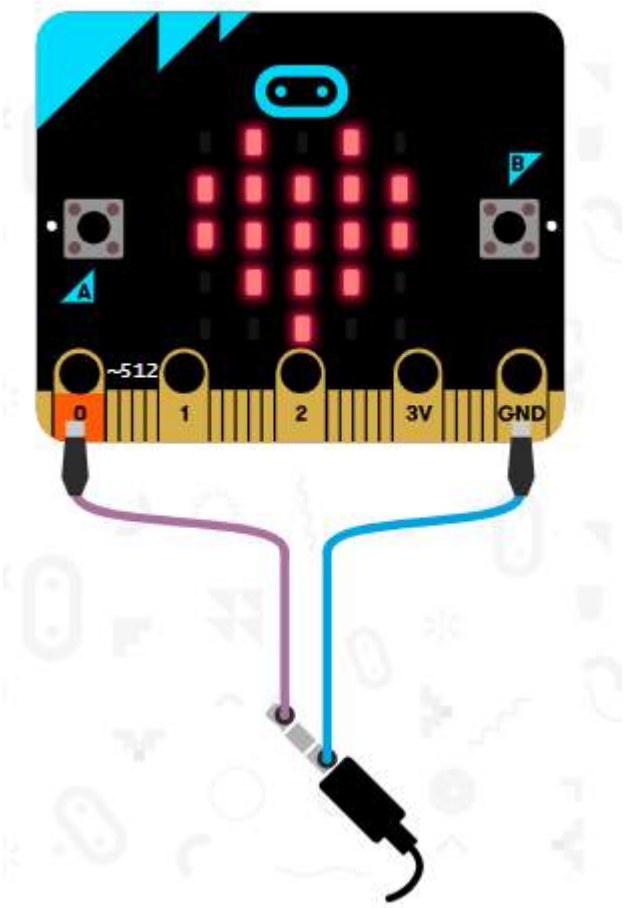
# Reto Música

Se puede añadir música utilizando unos cascos o altavoces de PC.. conectandolo al puerto P0 (pon sonido a tu ordenador y lo oirás):

https://makecode.microbit.org/---run?id=_PmV0UdLJrWRs

# Conexión con el "minijack"

Para la conexión se puede utilizar unas pinzas de cocodrilo (ver conexión)




https://www.youtube.com/embed/Di6b-SMuC-g

Si no se tienen pinzas de cocodrilo, haciendo un poco DYC (lease *mañoso*) con papel de aluminio y celo (ver cómo se hace)

# Descripción del código

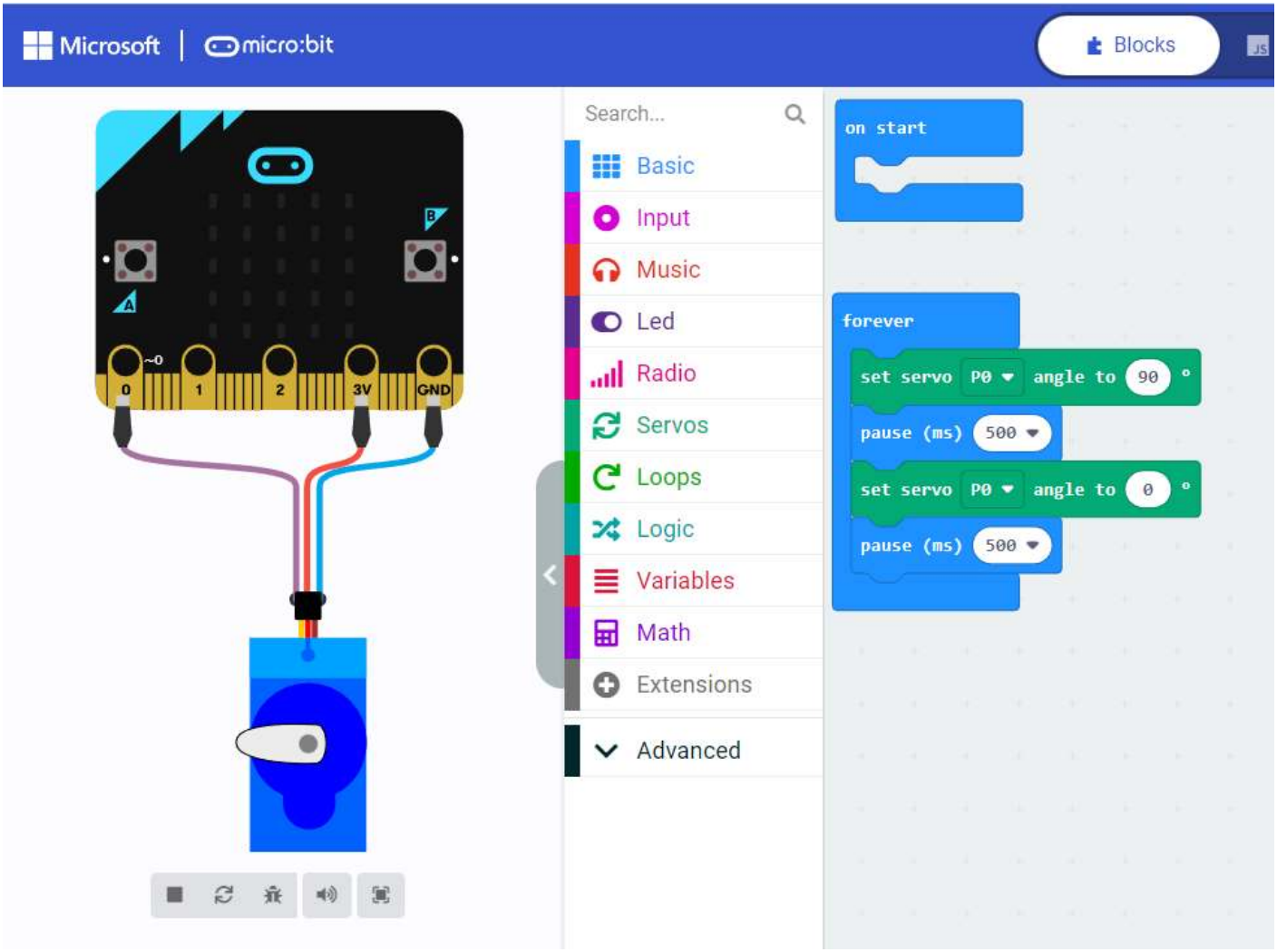
El código es muy fácil:

Hay que tener cuidado con lo de *repitiendo una vez* prueba no ponerlo y verás que se van mezclando melodías cada vez que ser repite el bucle y al final no se sabe qué es lo que está sonando

https://makecode.microbit.org/#pub:_PsreyC1tVbfg

# No intentes relés directamente

A pesar de que microbit parece que se puede poner directamente, no tiene suficiente energía para mover un motor, aunque sea un servo sencillo:




La única manera es:

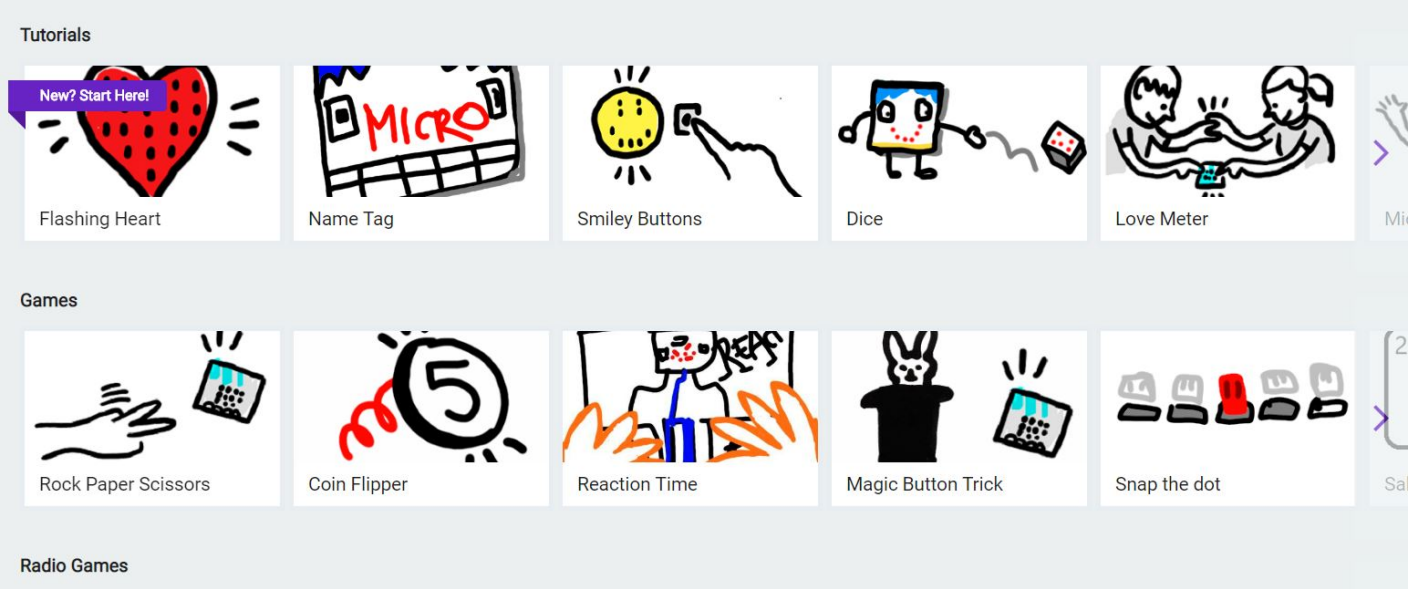
- O utilizar una placa de expansión, hay muchas en el marcado, un ejemplo es Microbit y Smartcar
- O sin placa, hacerlo manualmente, dando energía con unas pilas, lo tienes explicado aquí y este es el resultado https://youtube.com/shorts/Hx3_5GM-umc

https://www.youtube.com/embed/Hx3_5GM-umc

También venden tarjetas para hacer intermediarias entre estos elementos y la microbit, el Smartcar es un ejemplo.

# Reto libre

Ahora te retamos a lo siguiente: Entra en Makecode https://makecode.microbit.org y **ELIGE UN PROYECTO** EL QUE QUIERAS DE LOS QUE APARECEN **Y HAZLO**




Tendrás que hacerlo si quieres contestar a la pregunta test del curso correctamente !!!