

Microbit

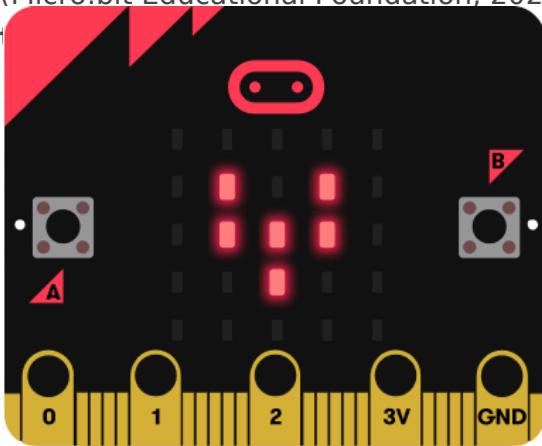
- [¿Qué es la microbit?](#)
- [Aprender pensando y aprender haciendo](#)
- [Descripción de la placa micro:bit](#)
- [Accesorios para micro:bit](#)
- [El entorno de trabajo de MakeCode](#)
- [Nuestro primer programa](#)
- [Jugando con los LED: bloques básicos de código](#)
- [Interacción con micro:bit por medio de los pulsadores](#)
- [Aprender con un led la diferencia entre analógico y digital](#)
- [Bloques de entradas: sonómetro, magnetómetro y acelerómetro con una sola línea de código](#)
- [Bucles y eventos de tiempo: un termómetro](#)
- [Bloques condicionales: luz crepuscular](#)
- [Brújula digital](#)
- [Nivel de burbuja usando el giroscopio](#)

¿Qué es la micro:bit?

La placa micro:bit

Micro:bit es un pequeño ordenador del tamaño de media tarjeta de crédito creado inicialmente por la BBC (British Broadcasting Corporation) en 2015 con el fin de promover el desarrollo de la robótica y el pensamiento computacional entre la población escolar de entre 11 y 15 años del Reino Unido.

Actualmente su uso está extendido entre **39 millones de escolares de más de 60 países** (Micro:bit Educational Foundation, 2023) con más de **5 millones de tarjetas distribuidas por**



Placa BBC micro:bit. Micro:bit Educational Foundation CC BY-

SA 4.0

El hardware es 100% libre, y está gestionado por una fundación sin ánimo de lucro fundada en 2016, la **micro:bit Educational Foundation**.

Micro:bit no es el primer proyecto de alfabetización informática de la BBC. Ya en 1981 produjo el **BBC Micro**, un ordenador de 8 bit programable mediante varios lenguajes de programación, entre ellos un lenguaje BASIC muy avanzado para la época. De este ordenador llegaron a venderse 1,5 millones de unidades, principalmente en el Reino Unido. Fue el ordenador con el que las escuelas británicas comenzaron la enseñanza de la informática.



BBC Micro. Dominio público.

Volviendo a micro:bit, el desarrollo del proyecto ha sido llevado a cabo por **29 socios tecnológicos** de primera línea. Por ejemplo, la implementación del Bluetooth corrió a cargo de la fundación propietaria de la marca, Bluetooth SIG, asociación privada sin ánimo de lucro.

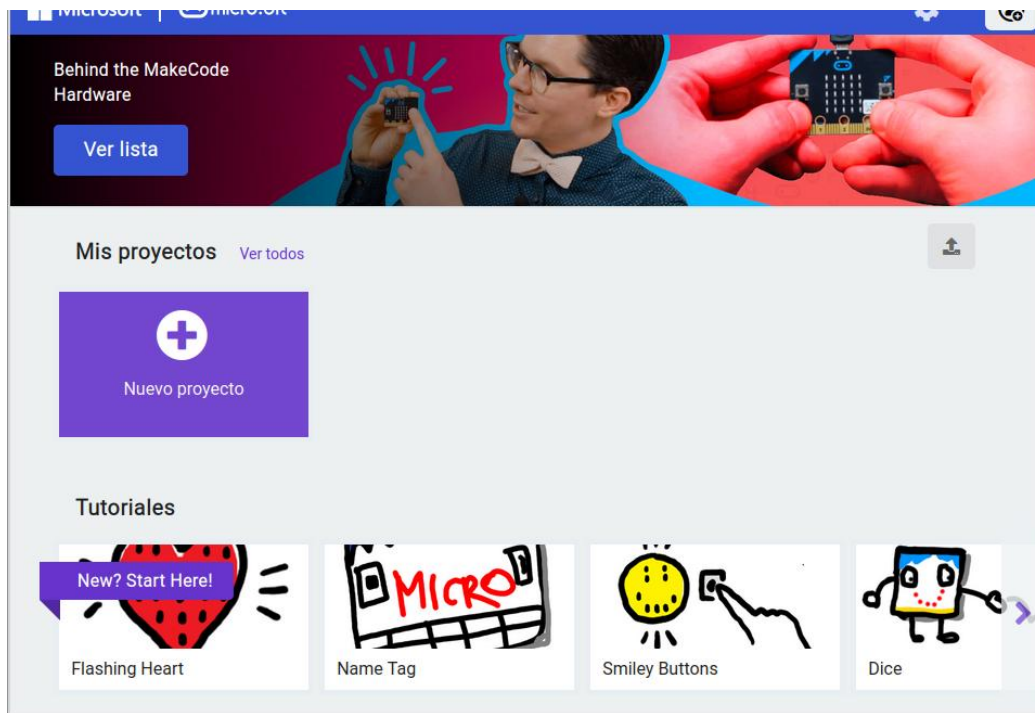
Micro:bit es **económico**; tanto las placas como los accesorios producidos por terceras empresas tienen un precio muy contenido.

El sistema destaca por su **alta integración de software y hardware**: basta un clic de ratón para cargar las librerías necesarias para que funcione cualquier complemento robótico, como sensores, pantallas, tarjetas de Internet de las Cosas, robots, casas domóticas, etc.

Programación de micro:bit

La programación de la placa se realiza desde un ordenador a través de un navegador cualquiera, estando disponibles **12 lenguajes de programación**. Dado su carácter abierto, existen múltiples soluciones de programación, aunque las más común es MakeCode.

El sitio **MakeCode** permite programar con bloques y también en Python y en Java, traduciendo de un lenguaje a otro instantáneamente. **No se necesita ningún registro** en la plataforma para poder programar. Nuestros programas se guardan en el servidor de MakeCode siempre que no utilicemos nuestro navegador de Internet en modo incógnito o que no tengamos activado el borrado automático de cookies.



Sitio web de Make Code.

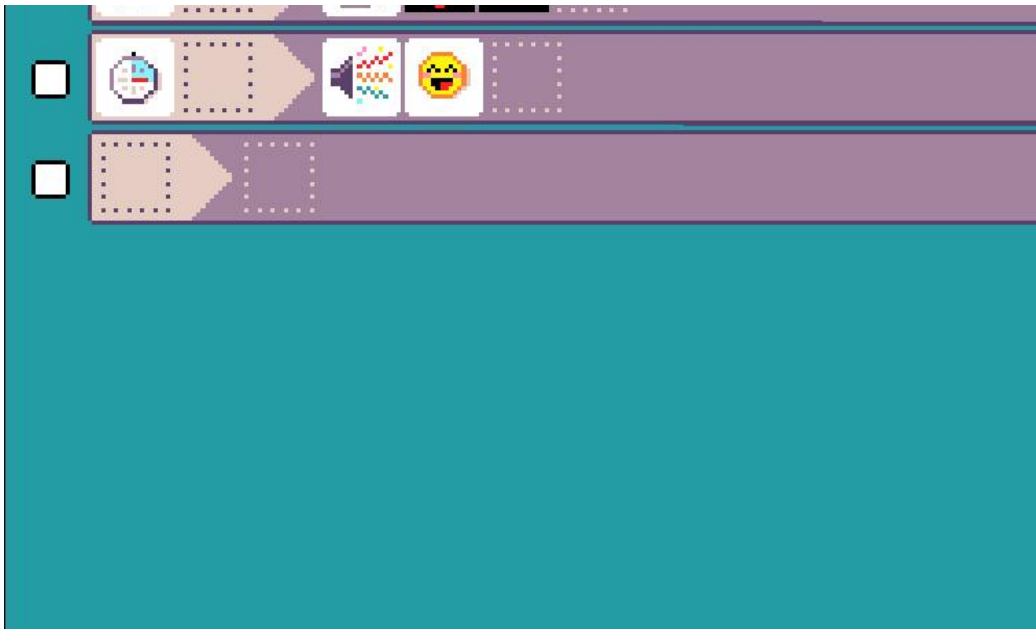
Captura de pantalla

Los **programas también pueden guardarse descargados en el ordenador** compilados en código de máquina. Al subir de nuevo el programa en el editor, se realiza una decompilación automática al lenguaje de bloques, Python o Java. Los programas guardados en código de máquina se pueden cargar directamente en micro:bit, que en el escritorio de un ordenador aparece como una simple unidad USB.

MakeCode contiene además **múltiples recursos** como tutoriales, vídeos, fichas de programación, cursos para el profesorado, ejemplos y propuestas de proyectos y experimentos en varios idiomas.

Otra solución muy usada para programar micro:bit es **MicroPython**, creada por Python Software Foundation, otra organización sin ánimo de lucro.

MicroCode permite que los más pequeños, a partir de los 6 años de edad, programen micro:bit mediante un sistema de fichas dispuestas en líneas de acción. Están disponibles un tutorial introductorio en 20 idiomas, una guía del usuario y muchos ejemplos. El proyecto es de código abierto.



Ejemplo de programa con

MicroCode. Captura de pantalla

Micro:bit también es programable en **Scratch** con sólo añadir una extensión al editor.

Todos los entornos de desarrollo descritos disponen de un **simulador de micro:bit**, por lo que ni siquiera resulta necesario disponer de una tarjeta física para aprender a programar.

Características de micro:bit

Características de **micro:bit V2**:

- Procesador de 64 MHz.
- 512 KB de RAM Flash y 128 KB de RAM.
- Matriz de 5 x 5 LED rojos.
- Dos pulsadores mecánicos y un tercer pulsador de apagado y reset.
- Pulsador táctil en el logotipo.
- Micrófono y altavoz.
- Acelerómetro y compás.
- Sensores de luz y de temperatura.
- Bluetooth de bajo consumo.
- Alimentación a 3 V o por USB.
- 25 pines de entradas y salidas.
- GPIO, PWM, I2C y SPI.
- 200 mA disponibles en las salidas para alimentar accesorios.

La placa micro:bit que se encontraba disponible en el mercado antes de la introducción de la V2 era la denominada **V1.5**. Se distingue de la V2 en que carece de micrófono, de altavoz y de

pulsador táctil en el logo. Internamente tiene un procesador menos potente y menos memoria RAM. Tampoco tiene piloto de encendido ni puede apagarse desde el pulsador de reset.

Aprender pensando y aprender haciendo

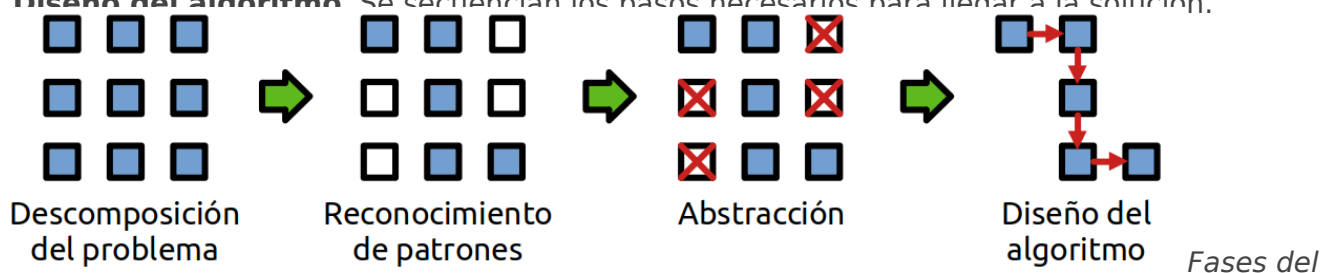
La introducción del **pensamiento computacional** en los currículos es tan reciente que no existe una definición precisa ni del concepto ni de su alcance. Esta cuestión comentada por Serrano (2022) es resuelta por el mismo autor, quien proporciona una definición operativa generada desde una perspectiva histórica:

“ El pensamiento computacional está compuesto por un conjunto de procesos de pensamiento involucrados en la formulación de problemas para que sus soluciones puedan representarse como pasos computacionales y algoritmos.

Según esta definición el pensamiento computacional tiene como propósito **resolver problemas**. Un problema parte de una situación nueva en la que el proceso de resolución es desconocido. Es por ello que la simple memorización de procedimientos para automatizar una tarea, no pone en juego los procesos mentales propios del pensamiento computacional.

Los procesos mentales básicos involucrados en el pensamiento computacional son:

- **Descomposición.** Se analiza y se fragmenta el problema en subproblemas más simples.
- **Identificación de patrones.** Los patrones son situaciones ya experimentadas cuya solución es previamente conocida o bien situaciones que se presentan repetidamente en el problema.
- **Abstracción.** Se pasan por alto los detalles que carecen de importancia.
- **Diseño del algoritmo.** Se secuencian los pasos necesarios para llegar a la solución.



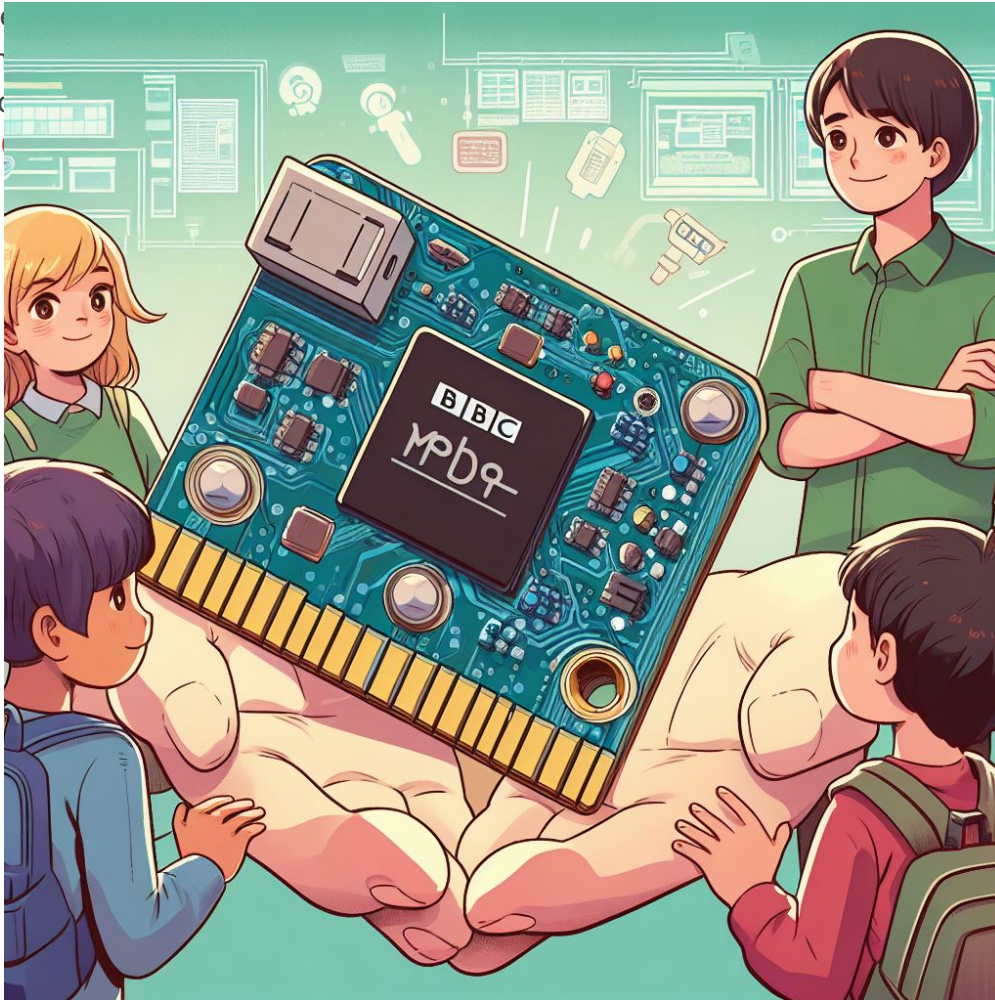
pensamiento computacional. Elaboración propia

Si la computación permite **aprender pensando**, la robótica educativa contribuye a **aprender haciendo**. Por este motivo, la robótica es denominada **computación física**. La robótica materializa el algoritmo, que puede de esta forma interactuar con el mundo real.

Más complicado resulta determinar **para qué introducir el pensamiento computacional en el aula**. Serrano (2022) cita dos corrientes principales, la que limita su desarrollo al ámbito del desarrollo de la competencia digital del alumnado y la que sostiene que el pensamiento computacional puede ser trabajado en todas las disciplinas con el objetivo de contribuir a la mejora global de los aprendizajes.

La **fundación micro:bit** sigue el primer enfoque, basando su visión en el riesgo de perpetuación y de profundización de las desigualdades sociales que producen las tecnologías digitales. Con su **Teoría del Cambio** (Micro:bit Educational Foundation, 2023c) pretende que todas las personas

de m co 2
r y trabajar en el mundo
on socios tecnológicos,
nes de estudiantes en



Alegoría de BBC micro:bit

generada con IA de Bing.

El segundo enfoque tiene en cuenta el perfecto encaje del pensamiento computacional en las **metodologías activas** que se derivan de la **corriente pedagógica constructorista**, que postula que el sujeto construye su conocimiento a través de la acción orientada a la creación. Dos metodologías muy adecuadas para introducir el pensamiento computacional son el aprendizaje basado en problemas y el aprendizaje basado en juegos.

Por otro lado, las actividades de robótica y de pensamiento computacional contribuyen a desarrollar toda una serie de **habilidades secundarias** o periféricas que no conviene desdeñar. Estas habilidades comunes a todas las **metodologías activas** son, entre otras, la creatividad, el espíritu crítico, la autonomía personal y las capacidades comunicativas.

Dado el gran número de personas y países involucrados en los programas de micro:bit, diversas organizaciones han realizado **estudios de impacto** desde los primeros años. Por ejemplo, un estudio de la BCC (2017) encontró que tras un año con micro:bit, la mayoría del alumnado de 11 años de edad consideraba que programar era fácil y que cualquiera podía hacerlo. Además se produjo un incremento significativo de estudiantes que consideraban que las TIC podían ser una opción profesional para su futuro. Este incremento fue del 70% entre las chicas. En cuanto al profesorado, alrededor de un 80% valoró muy positivamente la introducción de micro:bit en el aula.

Descripción de la placa micro:bit

Presentación de micro:bit

Micro:bit es una **pequeña placa de circuito impreso** de 52 mm x 42 mm, dimensiones que corresponden aproximadamente al tamaño de media tarjeta de crédito. El circuito dispone sus componentes electrónicos al aire, sin ninguna carcasa, y se encuentran soldados por las dos caras de la placa.

La placa **micro:bit es segura** para su uso con el alumnado de 8 a 14 años (Micro:bit Educational Foundation, 2023b). Ha sido ampliamente probada y cumple con todos los requisitos de seguridad y compatibilidad medioambiental exigibles. En el siguiente enlace se muestran todas las normas y estándares que cumple la tarjeta:

<https://microbit.org/compliance/>

El siguiente vídeo, pensado para personas sin conocimientos de robótica, proporciona una primera idea acerca de las posibilidades y del manejo de la placa micro:bit.

<https://www.youtube.com/embed/u2u7UJSRuko>

Kits

Micro:bit se encuentra disponible comercialmente como placa simple (*micro:bit single*) o como parte de un kit. El coste medio en el año 2023 de una placa es de unos 22 €. Por un precio un poco más elevado se puede adquirir un kit más completo, cuyo contenido está pensado como un **equipo de iniciación**.



Contenido de un kit de inicio.

Elaboración propia

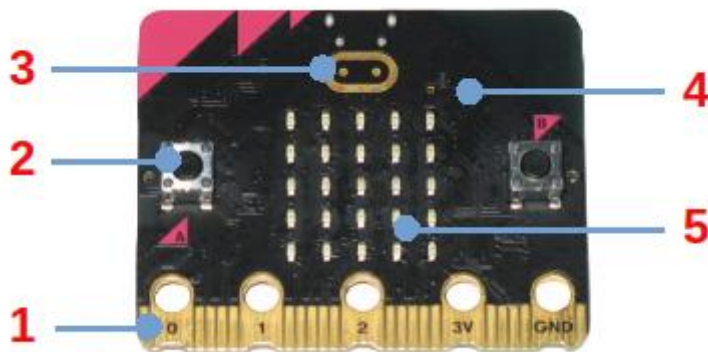
El kit presentado en la imagen contiene:

- Una **placa micro:bit** versión 2. El color de la decoración de la placa puede ser magenta, azul, verde o amarillo.
- Una pequeña **guía icónica de inicio rápido**.
- Una hoja con **recomendaciones de seguridad** en varios idiomas.
- Un **cable corto micro USB** para conectar la placa al ordenador.
- Una cajita **portapilas** con conector JST.
- Dos **pilas alcalinas** IEC LR03 (AAA).

Componentes electrónicos integrados en la placa

El anverso

mostrado en la figura.



Anverso de la placa BBC micro:bit.

Elaboración propia

La parte marcada con el número **1** es una **fila de contactos o pines** que permiten conectar la placa a múltiples accesorios, como altavoces, robots, sensores o actuadores. Dispone de 25 pines,

5 de ellos de anillo. Estos últimos están pensados para conectar cables por medio de pinzas o bananas.

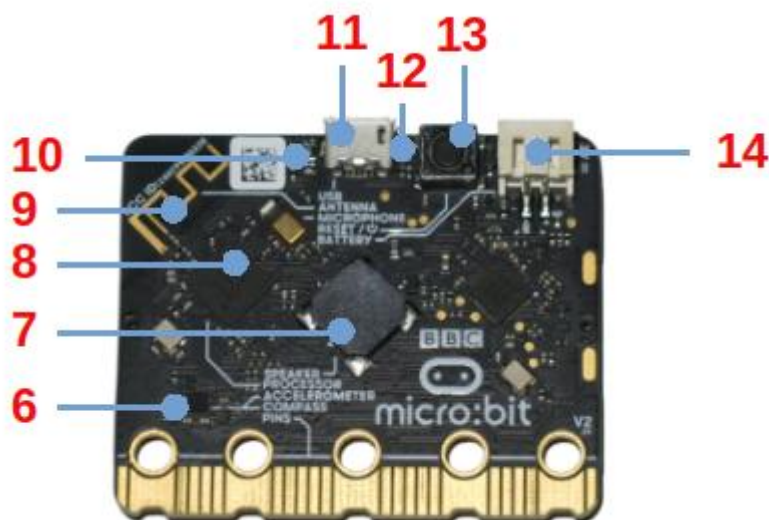
El número **2** corresponde al **pulsador A**. En el lado derecho de la placa existe otro **pulsador B**. Ambos pulsadores son programables y permiten al usuario comunicarse con micro:bit.

El número **3** apunta al **pulsador táctil**. Se trata de una novedad de la versión 2 de micro:bit. El pulsador puede distinguir entre pulsaciones rápidas y pulsaciones prolongadas.

El sensor marcado con el número **4** es un **micrófono**. Es otra novedad de la versión 2 de la placa. Aunque puede grabar sonido, las grabaciones resultan de una calidad muy pobre. Es por ello que el micrófono se usa sobre todo para medir el ruido ambiental. Un **diodo LED de color rojo** indica que el micrófono está en funcionamiento.

La **matriz de 25 LED rojos** está marcada con el número **5**. Se usa para representar mensajes de texto e iconos. Su intensidad es ajustable. Sirve también como **sensor del nivel de luz** ambiental.

El reverso de la placa es un poco más complejo. Aquí se encuentran soldados los dispositivos que
pr, de los sensores y de los circuitos
in ción USB o Bluetooth.



Reverso de la placa BBC micro:bit.

Elaboración propia

El circuito número **6** es un **acelerómetro y un compás**. Este dispositivo permite medir aceleraciones y giros, así como campos magnéticos.

El componente **7** es un pequeño **zumbador** que puede reproducir tonos y sonidos. Su potencia es muy baja, pero micro:bit puede también conectarse a un altavoz externo.

El procesador es el circuito marcado con el número **8**. Además contiene un **sensor de temperatura**.

La **antena Bluetooth** se indica con el número **9**. Gracias al Bluetooth incorporado, micro:bit puede enviar y recibir datos por radio a cortas distancias.

Cuando el **LED de color rojo 10** está encendido indica que la placa está conectada a una fuente de alimentación. Micro:bit puede ser alimentado por un ordenador o por una batería externa través de un cable USB o bien por un par de pilas.

La placa puede comunicarse con un ordenador mediante el **conector micro USB 11**. Mediante esta conexión micro:bit puede ser programado y también puede enviar al ordenador los datos que recoja con sus sensores.

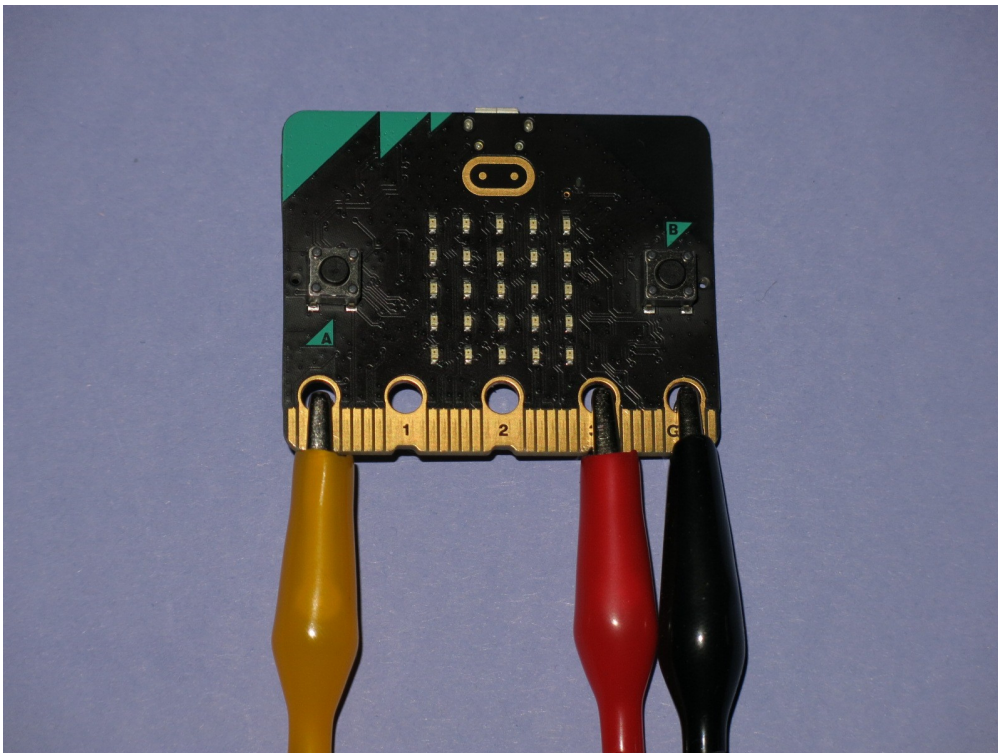
El **LED ámbar 12** parpadea cuando la placa está conectada a un ordenador y se encuentra emitiendo o recibiendo datos.

El **pulsador 13** se usa para **reiniciar** el programa contenido en la placa. Una pulsación prolongada apaga micro:bit, pero sólo si la placa es de la versión 2.

El **conector JST 14** permite conectar el portapilas para alimentar la placa con dos pilas alcalinas de 1,5 V. También se pueden usar pilas recargables; las clásicas pilas recargables de NiMH o níquel-hidruro metálico que se venden en algunos supermercados hacen que tanto micro:bit como sus accesorios funcionen sin problemas.

Accesorios para micro:bit

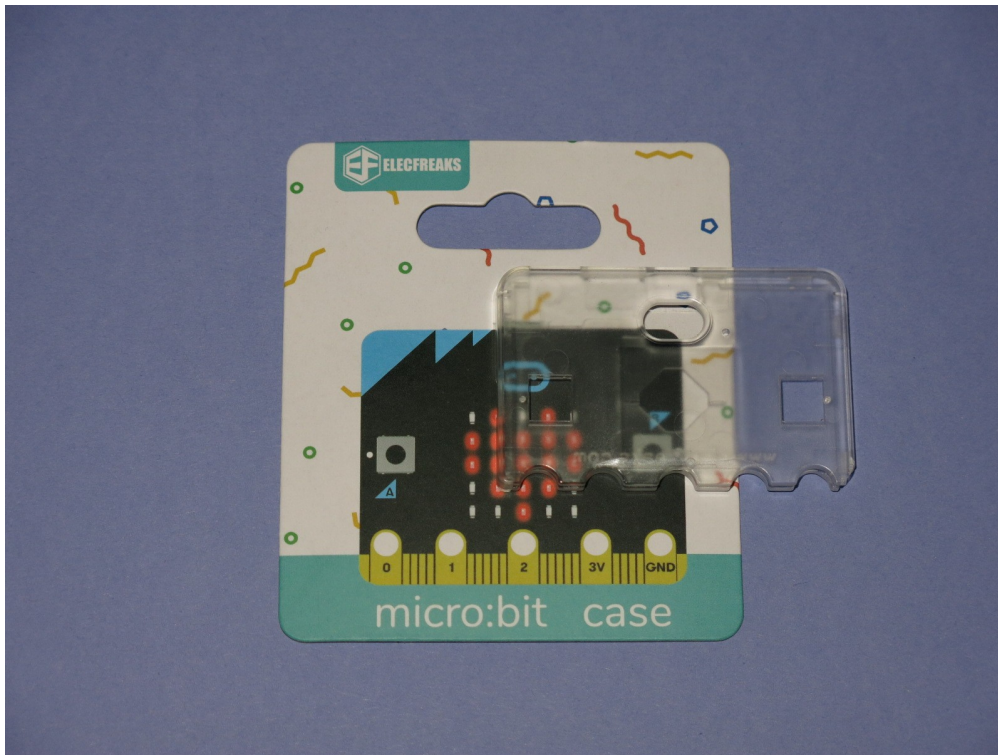
Micro:bit puede ser expandida con **múltiples accesorios fabricados por empresas asociadas**. Los sensores, las placas de expansión o los robots se acoplan a la tarjeta y se programan de forma simple e intuitiva. También es posible conectar micro:bit a dispositivos no pensados específicamente para la placa, como altavoces o sensores autofabricados, al estilo de [Makey Makey](#). En estos casos es muy conveniente disponer de cables de conexión con pinzas de cocodrilo. Las pinzas se conectan a los pines circulares números 0, 1 y 2 y a los pines GND (masa de alimentación) y 3V (salida de alimentación de 3 volt).



Pinzas de cocodrilo conectadas a micro:bit V2. Elaboración propia

Debemos tener cuidado de no cortocircuitar los pines GND y 3V.

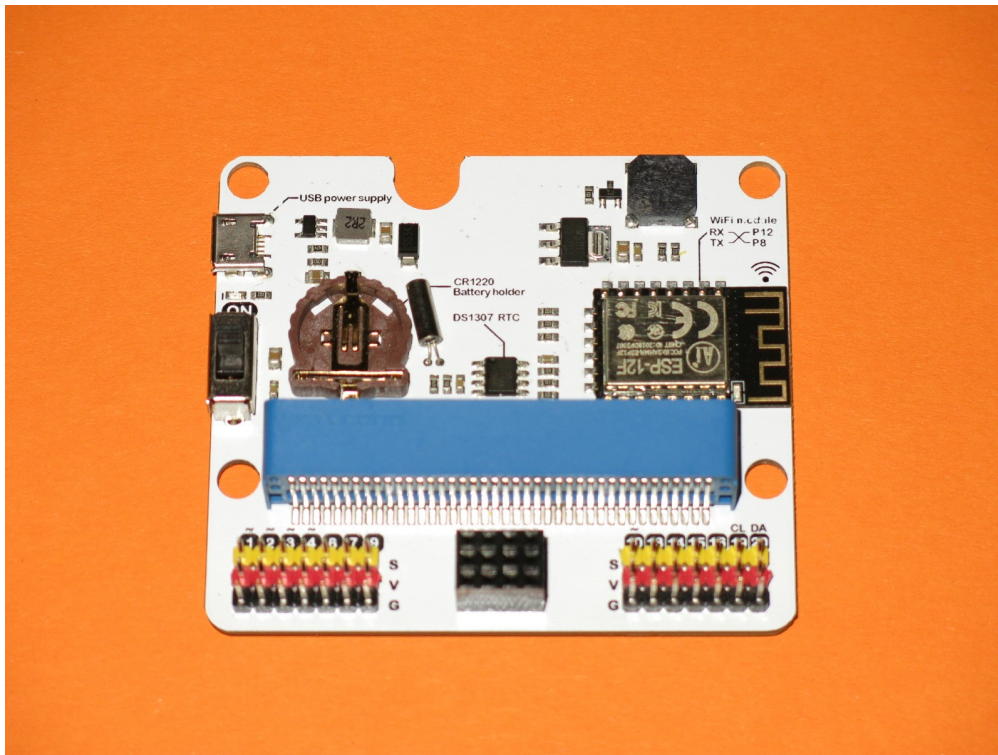
Las **fundas de plástico** tienen un precio muy bajo y protegen la placa eficazmente, pero hay que tener cuidado de adquirirlas de color transparente, o el sensor de luz podría no funcionar.



Funda para micro:bit V2. Elaboración propia

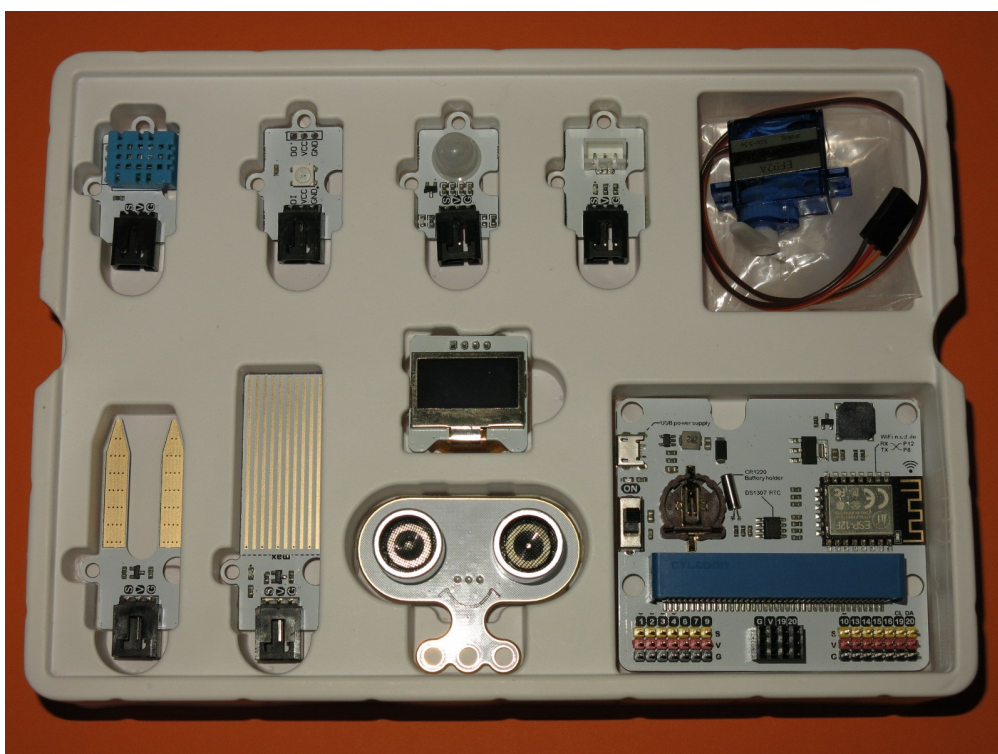
Quizás los accesorios más potentes sean las **placas de expansión**. En la figura se muestra una **placa IoT** (Internet de las Cosas) del fabricante [ElecFreaks](https://www.electfreaks.com/). La tarjeta micro:bit se inserta directamente en la ranura azul. La placa dispone de un reloj de cuarzo alimentado por una pila de litio de larga duración, de un módulo de comunicación wifi y de conectores de pines (GVS) para añadir sensores. El precio de la placa ronda los 20€ en 2023.

Los terminales GVS se emplean para conectar sensores y actuadores. La G se refiere al terminal de masa (polo negativo, marcado en negro), la V al terminal de alimentación (polo positivo, marcado en rojo) y la S (marcada en amarillo) al terminal que recoge la señal del sensor o del actuador.



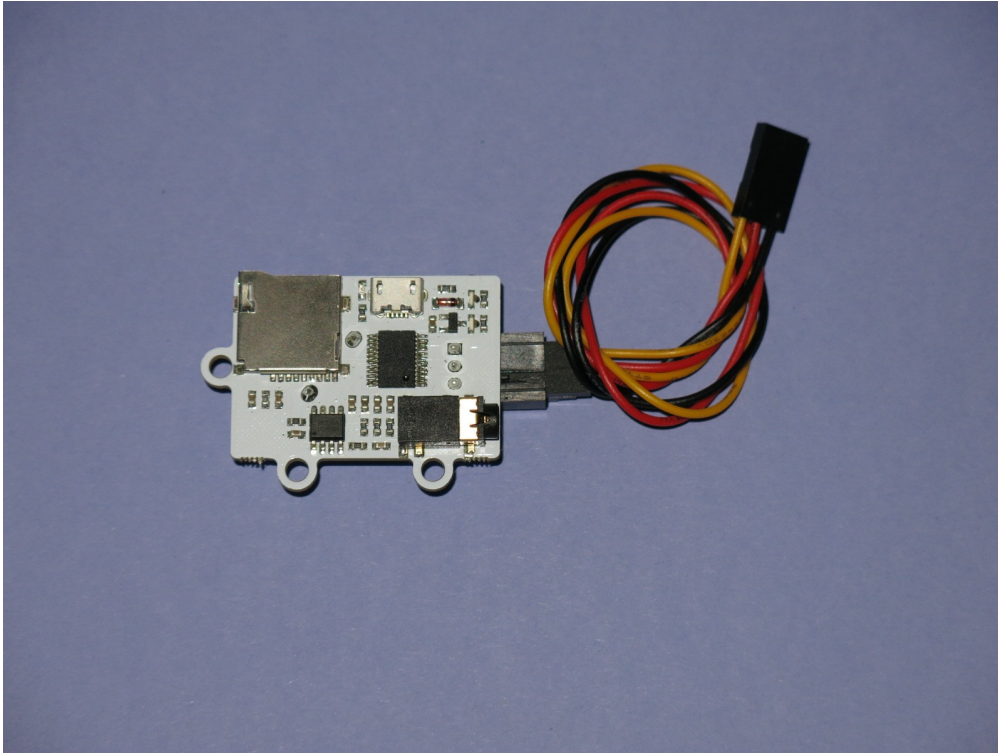
Placa IoT de ElecFreaks. Elaboración propia

Los sensores vienen generalmente agrupados en kits. La figura es un kit de [ElecFreaks](#) orientado a actividades relacionadas con la **agricultura**. Además de una placa IoT para monitorizar a distancia nuestros cultivos, también hay sensores de humedad del suelo, de humedad ambiental, de lluvia, de luz y de temperatura, además de un servo para abrir y cerrar ventanas, un sensor de ultrasonidos para detectar intrusos y una pantalla OLED para mostrar información. El precio del kit es de unos 60€ en 2023.



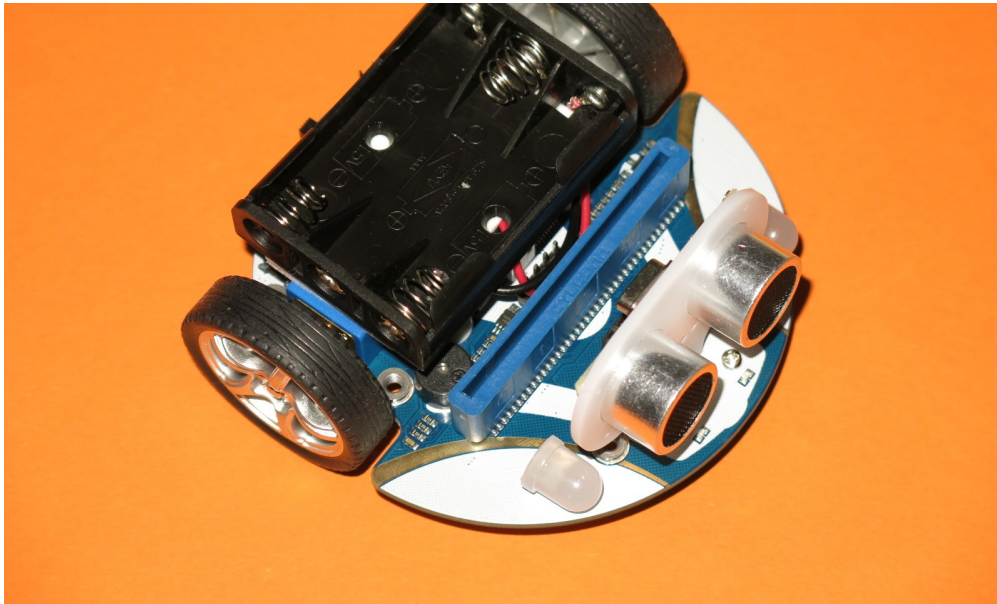
Kit de agricultura de Elecbreaks. Elaboración propia

Algunos accesorios añaden grandes funcionalidades, como el **reproductor de audio mp3** de la fotografía. Incorpora un altavoz de más calidad y un lector de tarjetas micro SD. La cantidad de archivos reproducibles sólo está limitada por la capacidad de la tarjeta.



Reproductor de audio en formato mp3. Elaboración propia

Los **robots** como **Cute Bot**, disponen de dos ruedas controlables individualmente para ajustar la velocidad y la dirección. Además incorporan sensores para el seguimiento de líneas, luces de colores, sensores de distancia para evitar obstáculos y sensores de infrarrojos para el control con mandos a distancia. Su precio en 2023 es de unos 35€.

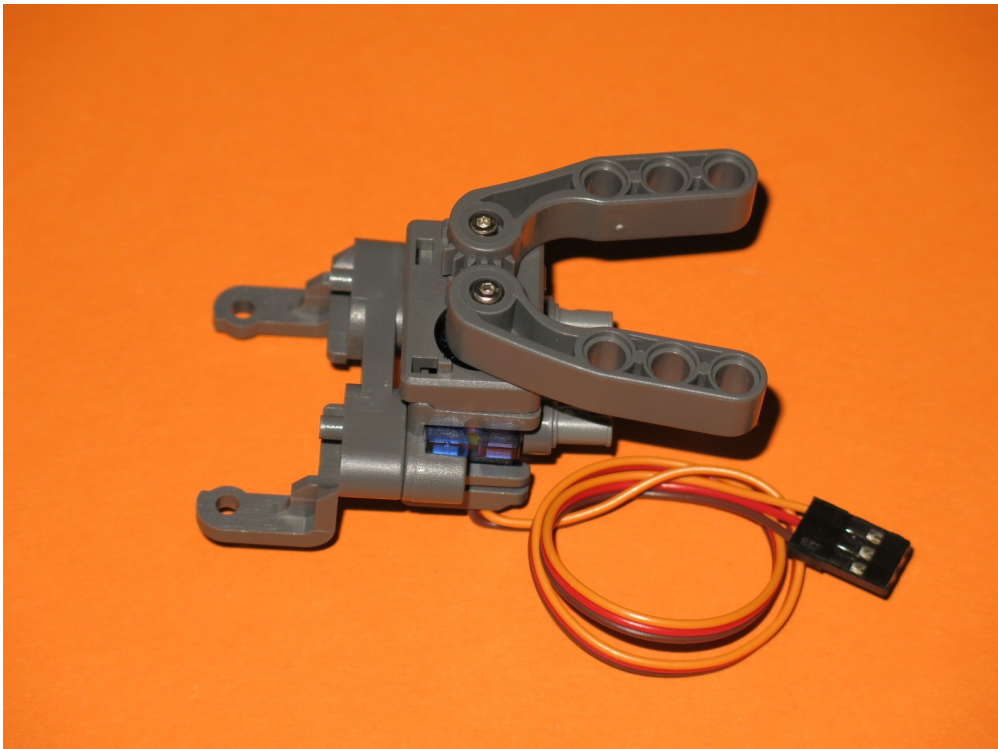


Robot Cute Bot de

Elecfreaks. Elaboración propia

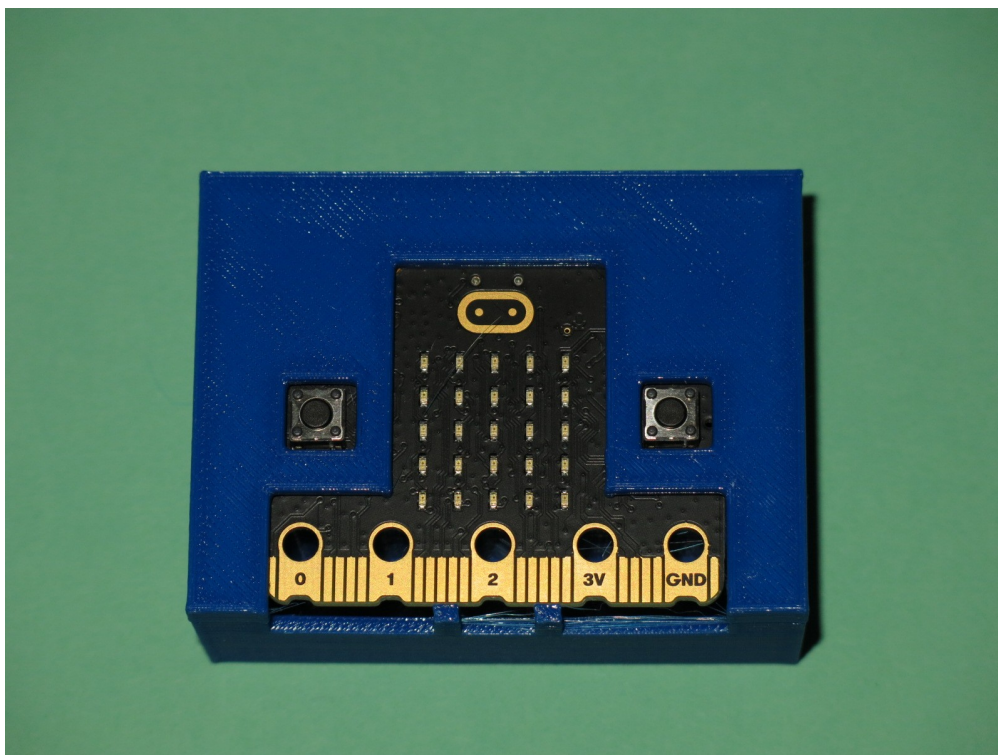
El Maqueen es también muy parecido ver comparativa en <https://libros.catedu.es/books/microbit-car/page/cars-para-microbit>

El **brazo manipulador** de la fotografía se conecta a pines GVS como los de una placa de expansión o como los de un robot. Abre y cierra las pinzas gracias a un servo incorporado y es compatible con juegos de construcción de bloques.



Brazo manipulador. Elaboración propia

Es posible encontrar archivos para imprimir accesorios en 3D en repositorios como [Thingiverse](https://www.thingiverse.com/). En la fotografía se muestra una cajita de plástico para proteger la placa micro:bit y contener la caja portapilas.



Caja impresa en 3D para micro:bit según diseño de University of Bristol. CC BY-NC. Elaboración propia

El entorno de trabajo de MakeCode

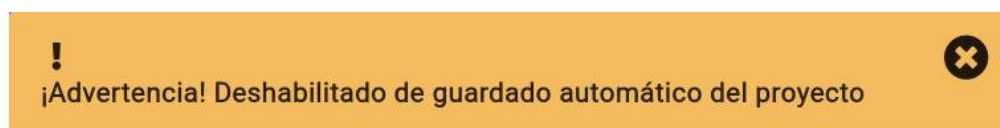
“ El **punto de partida** para el estudio de un lenguaje de programación pasa inevitablemente por los **lenguajes gráficos** o de bloques que facilitan su estudio al mostrar de forma visual las instrucciones del algoritmo programado. (Morales, 2023)

Aunque micro:bit dispone de 12 lenguajes de programación, la mayoría de ellos textuales, parece apropiado iniciar al alumnado en la **codificación con bloques**. Generalmente, la realización de programas con bloques es el paso siguiente a los juegos desconectados de codificación, a los robots de suelo y los paneles de programación desenchufados.

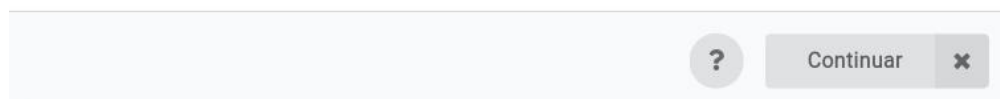
MakeCode ofrece un entorno integrado, simple e intuitivo de **codificación por bloques**. Hay que decir que en el entorno de MakeCode también están disponibles los lenguajes **Java** y **Python**, con la ventaja de que un programa desarrollado mediante bloques puede ser traducido instantáneamente a cualquiera de los otros dos lenguajes. También es posible realizar la operación inversa, es decir, traducir un programa en Java o Python a bloques.

Para acceder al entorno basta con abrir en un navegador cualquiera el enlace:

<https://makecode.microbit.org/>. Si nuestro navegador tiene desactivadas las cookies o navega en



No podemos guardar tus proyectos en este momento. Todavía puedes guardar manualmente tu proyecto mediante la descarga directa o compartiendo tu proyecto.

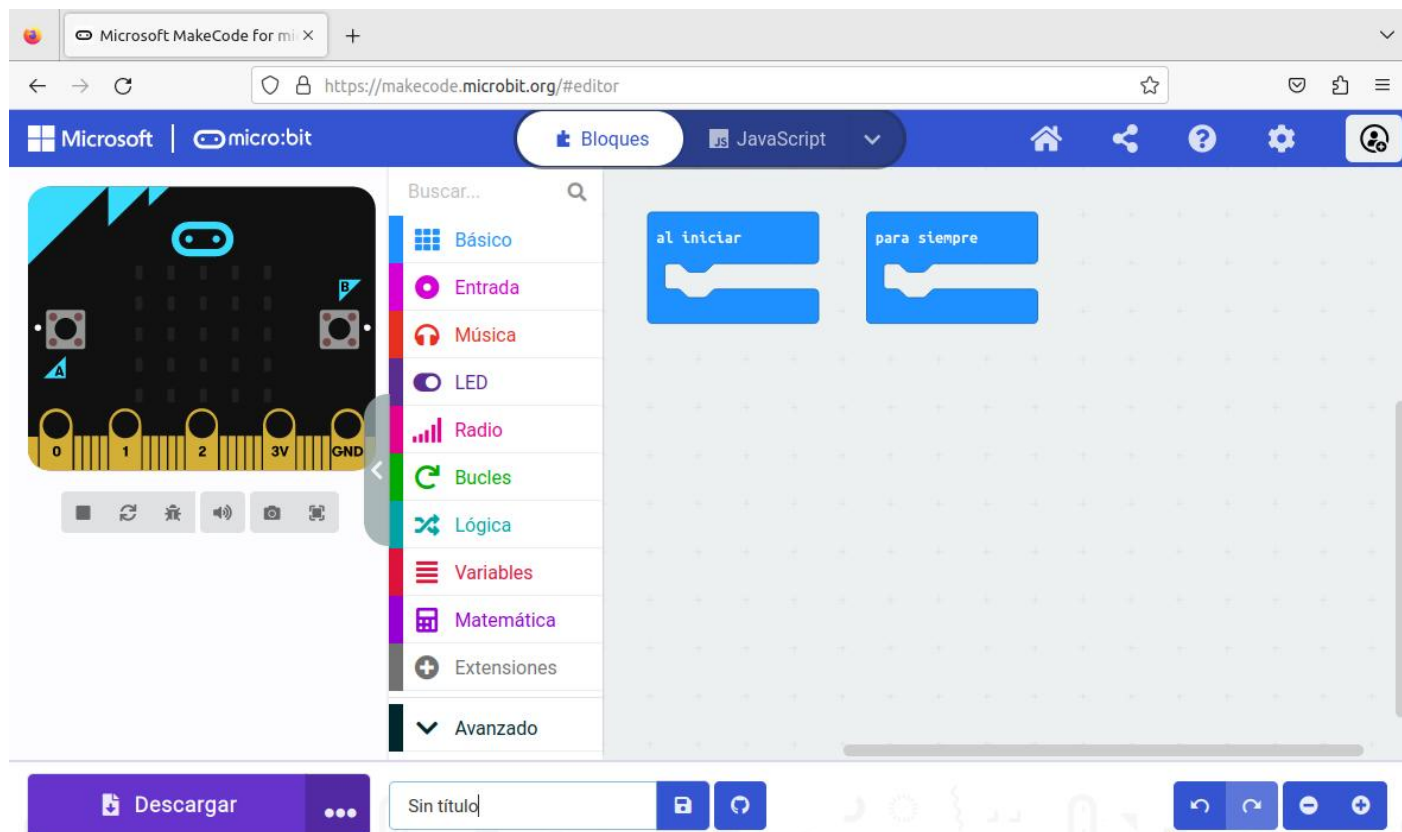


Pulsando sobre el botón de

continuar aparecerá la ventana principal de MakeCode. Para iniciar el entorno de programación habremos de pulsar sobre el botón de **nuevo proyecto**:



MakeCode nos pedirá seguidamente **dar un nombre** cualquiera al proyecto y pulsar el **botón de crear**, tras lo cual se abrirá la ventana del editor.



Entorno Microsoft MakeCode. Captura de pantalla

Para trabajar con MakeCode no se requiere registro. No obstante, pulsando sobre el icono de la esquina superior derecha, situado en la barra de herramientas, es posible acceder a MakeCode desde una cuenta de Google o de Microsoft. Esta característica permite guardar en la nube los diseños realizados y acceder a ellos desde cualquier ordenador.

Podemos ajustar el **idioma de la interfaz** pulsando sobre la rueda dentada de la esquina superior derecha. Están disponibles 33 idiomas, incluyendo el español.

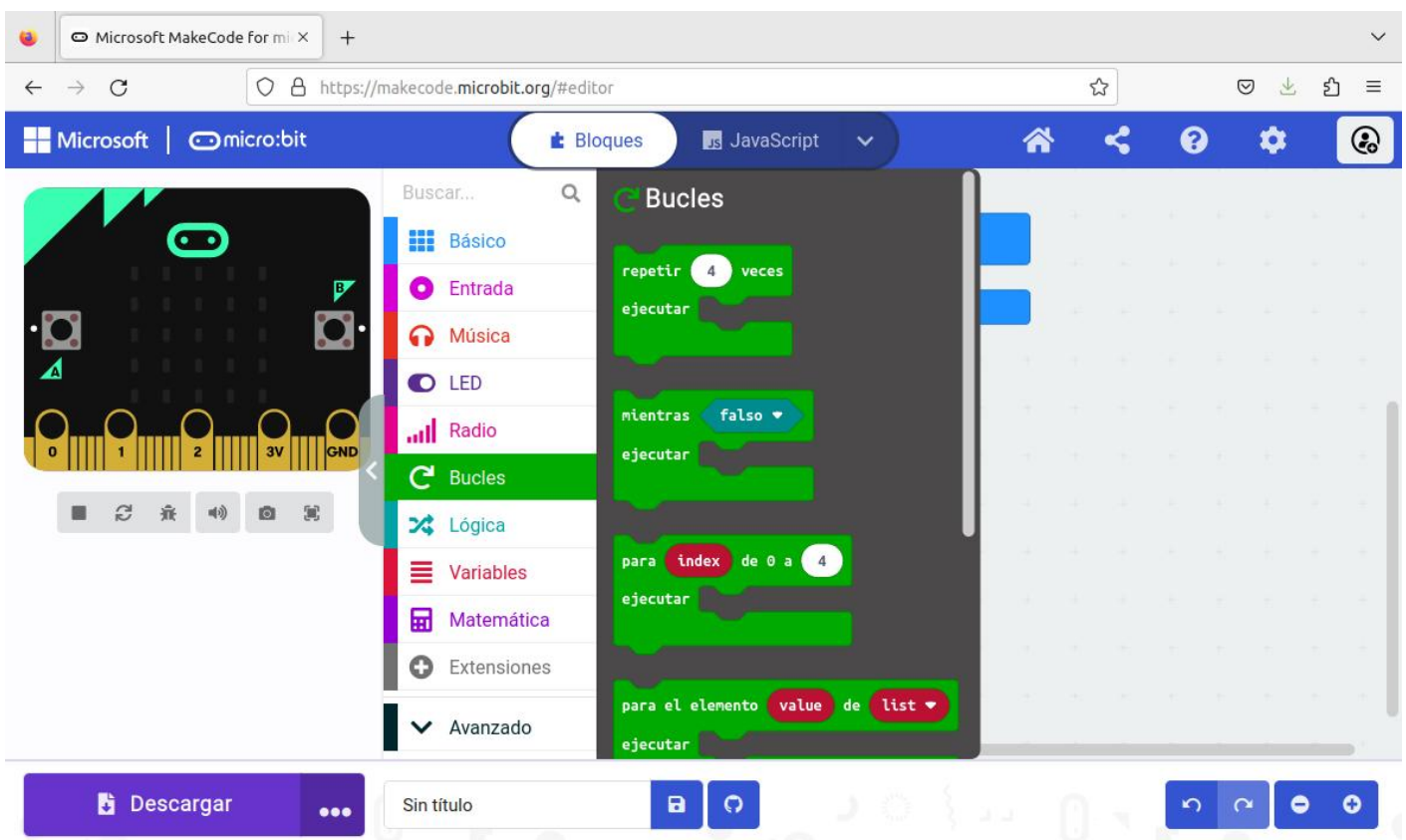
El panel de la izquierda representa el anverso de una placa micro:bit. Se trata de un **simulador**, que mostrará en tiempo real el efecto del programa en desarrollo. Nótese que no es necesario disponer de una placa real para realizar los primeros pasos en programación con micro:bit.

Con el puntero del ratón se puede interactuar con la placa y pulsar sus botones, agitarla o cambiar las condiciones simuladas de luz o de ruido. Por otro lado, los controles situados bajo la placa nos permiten:



- Apagar o encender el simulador.
- Reiniciar el simulador, es decir, el programa en ejecución.
- Entrar en modo de depuración para localizar y solucionar errores.
- Activar o desactivar el altavoz del ordenador.
- Tomar una fotografía de la placa.
- Mostrar el simulador en pantalla completa.

El panel central es una **caja de herramientas** en forma de menú desplegable con todos los **bloques de programación** disponibles. Los bloques pueden ser arrastrados directamente con el ratón desde el menú. Cada color del menú se identifica con un tipo de acción o de estructura de control. Por ejemplo, el color verde se reserva para los bloques que permiten ejecutar acciones repetitivas o bucles.



La **caja de búsqueda** colocada sobre el menú desplegable es muy útil para encontrar rápidamente el bloque más adecuado para realizar la acción deseada en cada momento.

El panel de la derecha es el **área de programación**. Aquí se escriben los programas arrastrando y encajando los bloques de código. Cuando se abre el navegador aparecen por defecto dos bloques azules de eventos del **menú básico**:

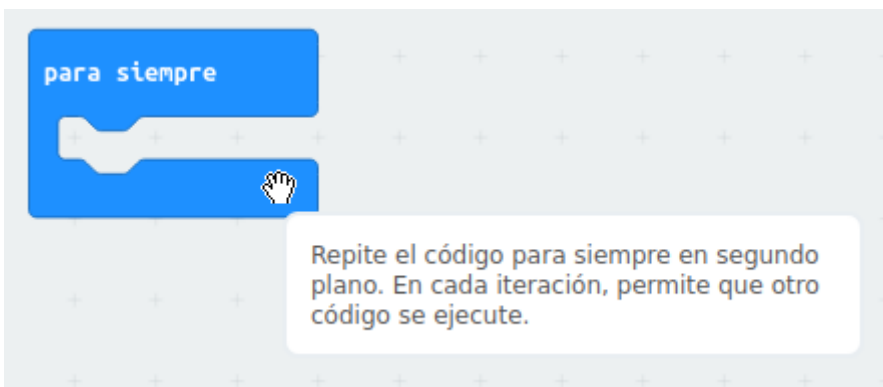
- **Al iniciar**, es la estructura que contendrá el algoritmo que será ejecutado por la placa sólo una vez, tras ser encendida o reiniciada.

- **Para siempre**, es el bucle que contendrá el algoritmo que se ejecutará indefinidamente, una y otra vez, cuando acabe la ejecución del algoritmo de inicio.

Pulsando con el botón derecho del ratón sobre un bloque se desplegará un **menú contextual** que dará varias opciones, como duplicar, borrar u obtener ayuda sobre el uso del bloque.[C](#)



Al colocar el puntero del ratón sobre un bloque, el editor mostrará una **etiqueta de ayuda** con una breve explicación sobre la función del bloque.



<https://makecode.microbit.org/reference> contiene explicaciones detalladas de los bloques de MakeCode.

En la parte inferior de la ventana de MakeCode se dispone la **barra de herramientas**, con botones para realizar acciones como descargar el programa en un placa micro:bit conectada al ordenador, guardar el programa en nuestro ordenador o en un repositorio en línea, hacer y deshacer y ampliar o reducir la imagen.

Nuestro primer programa

El primer programa que se realiza cuando se comienza a aprender un nuevo lenguaje consiste casi siempre en presentar un **saludo por pantalla**. Micro:bit es más original, puesto que propone como primer programa un **corazón palpitante**.

Vamos a combinar el saludo inicial con el corazón. De esta forma ilustraremos el funcionamiento de los bloques de eventos **al inicio** y **para siempre**.

Se trata de seguir los pasos explicados en la página anterior y crear un programa llamado **Hola**. Tras la apertura del editor pulsaremos sobre el menú **Básico** y arrastraremos y encajaremos los bloques para configurar el siguiente programa:



Para escribir el texto ¡Hola! basta con hacer clic sobre la caja blanca del bloque **mostrar cadena**. De igual forma, los iconos de los bloques **mostrar ícono** y los números de los bloques **pausa (ms)** pueden seleccionarse pulsando sobre las flechas, acción que despliega menús de selección.

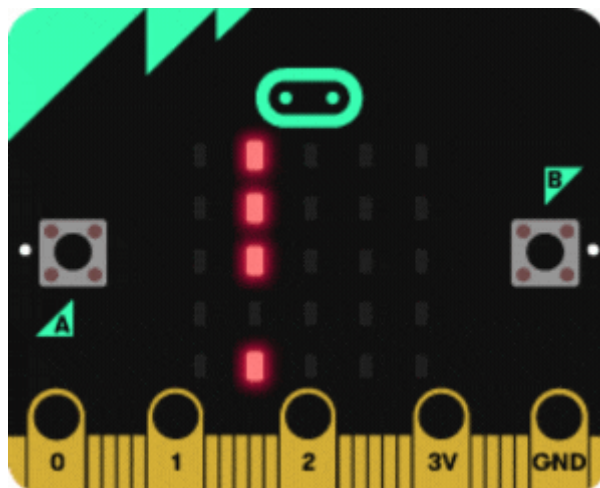


Como en el caso de la caja de texto, también puede teclearse directamente el número de milisegundos del bloque **pausa (ms)**.

¿Qué hace el programa? Cuando se enciende la placa, el **algoritmo de inicio mostrará una vez** mediante la matriz de LED la cadena de texto "¡Hola!". Como la pantalla es muy pequeña, micro:bit deslizará el texto horizontalmente, de derecha a izquierda.

Seguidamente, el **algoritmo principal** mostrará en pantalla el icono de un corazón grande y entrará en pausa 500 milisegundos, es decir, medio segundo. Pasado este tiempo, micro:bit mostrará un corazón pequeño y esperará otro medio segundo. Este algoritmo se ejecutará indefinidamente hasta que se apague la placa, se pulse el botón de reinicio o se acaben las pilas. El efecto visual será el de un **corazón palpitante**.

Nada más terminar de colocar el último bloque, el **simulador de MakeCode** mostrará en pantalla el resultado del programa en ejecución. Para reiniciar el programa hay que pulsar sobre el icono de reinicio, situado justo bajo el dibujo de la placa micro:bit.



Cargar el programa en micro:bit

Hay que conectar primero la placa al ordenador por medio del cable USB. Después de unos instantes se instalará el programa en la tarjeta micro:bit y se podrá utilizar.



La unidad Micro:bit en el

escritorio de Linux Mint

Si la tarjeta es nueva, llevará cargado de fábrica un programa de demostración que empezará a ejecutarse nada más ser conectada al puerto USB. El programa mostrará mensajes, iconos en movimiento y sonidos y pedirá al usuario que calibre el compás magnético.

Seguidamente hay que pulsar sobre el botón **Descargar**, acción que abrirá el diálogo de guardado de archivos del sistema operativo. Bastará con seleccionar la unidad MICROBIT y confirmar la descarga.



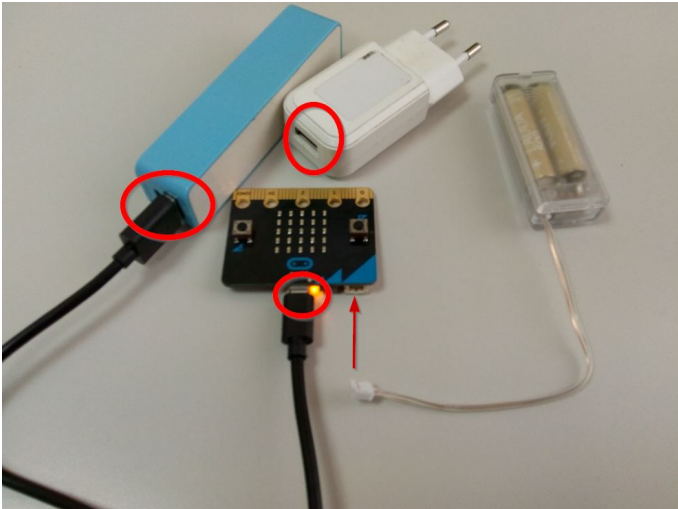
Si tienes Vitalinux, te recomendamos visitar [esta página](#).

Mientras dure la transferencia de datos, el **LED ámbar** situado junto al conector USB de micro:bit **parpadeará rápidamente**, quedando fijo cuando el programa haya sido transferido.

La transferencia de archivos a micro:bit lleva algo de tiempo, especialmente si el programa es largo. Tras la transferencia, el programa empezará a ejecutarse inmediatamente sobre la tarjeta micro:bit.

Sólo es posible cargar un programa a la vez en micro:bit, por lo que cada nuevo programa se reescribe sobre el anterior.

Si se desea que micro:bit funcione desconectado del ordenador, simplemente hay que alimentarlo por medio de la caja de pilas o mediante una batería externa o un cargador USB. El programa se ejecutará de igual forma, pues se encuentra guardado en la memoria de la placa.



Jugando con los LED: bloques básicos de código

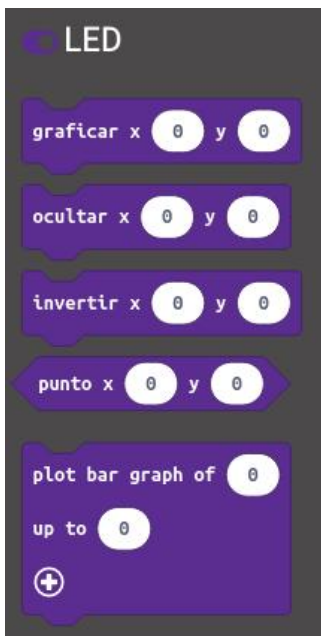
El menú **Básico** contiene los bloques más usados para programar micro:bit. En la página anterior se han visto los bloques de eventos **al iniciar** y **para siempre**. También se han utilizado los comandos **mostrar icono**, **pausa (ms)** y **mostrar cadena**.

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.



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

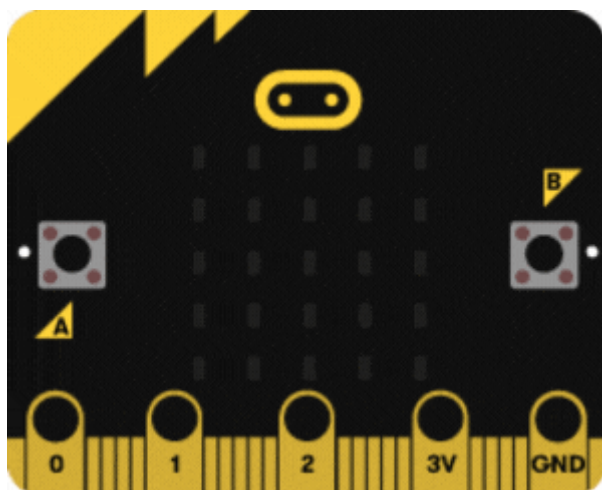


pausa (ms) 200

mostrar LEDs



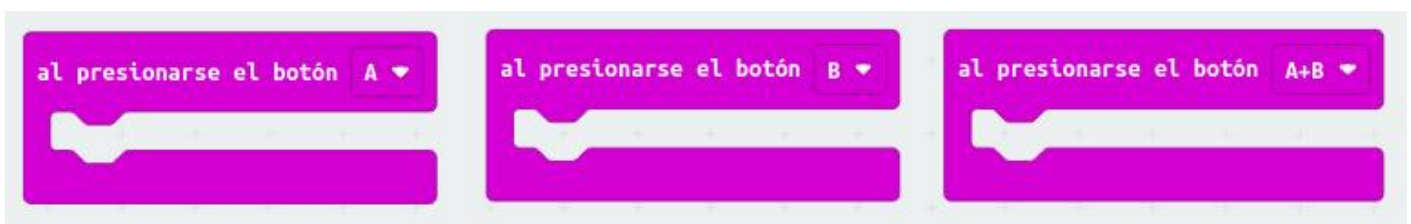
Nótese que al inicio se ha ajustado el brillo de la pantalla. La función **ajustar brillo** se encuentra pulsando sobre los tres puntos suspensivos bajo el menú **LED**. El brillo máximo es de 255, por lo que 128 representa un valor medio.



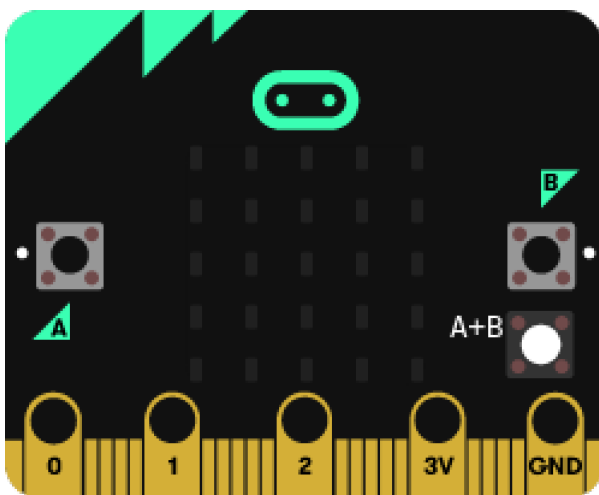
Interacción con micro:bit por medio de los pulsadores

Pulsadores mecánicos

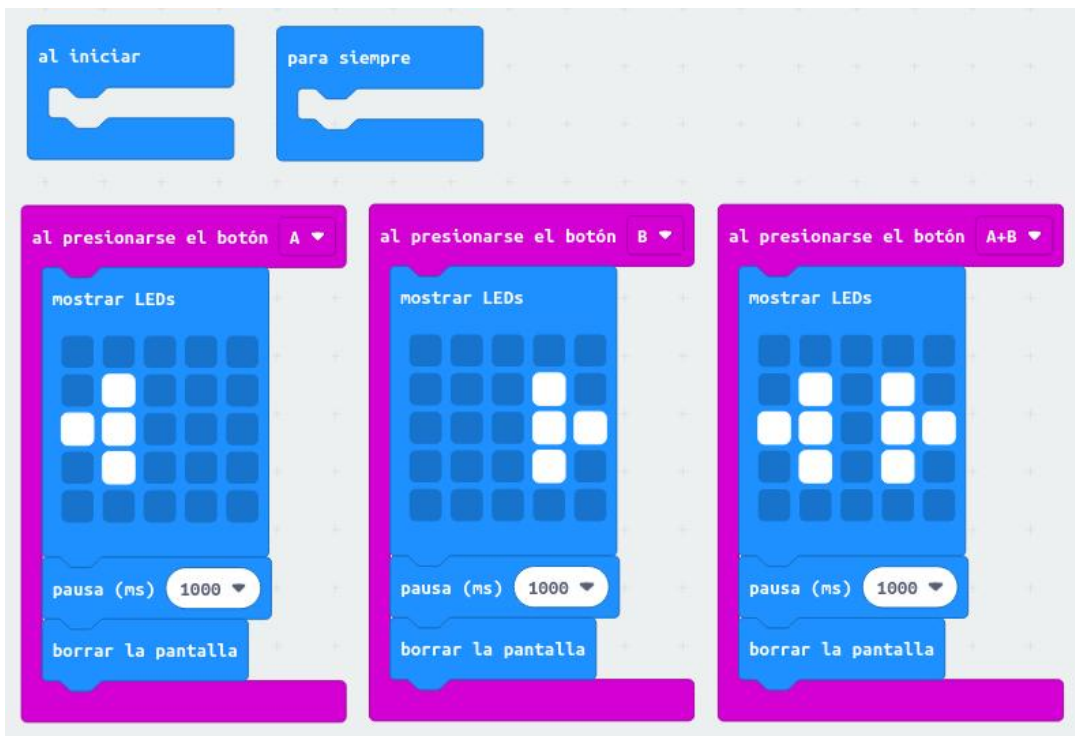
El menú **Entradas** contiene los bloques que permiten captar datos de los sensores integrados en micro:bit. Para que la placa lleve a cabo acciones al pulsar los botones **A**, **B** ó **A** y **B** a la vez, disponemos del bloque de eventos denominado **al presionarse el botón**:



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.



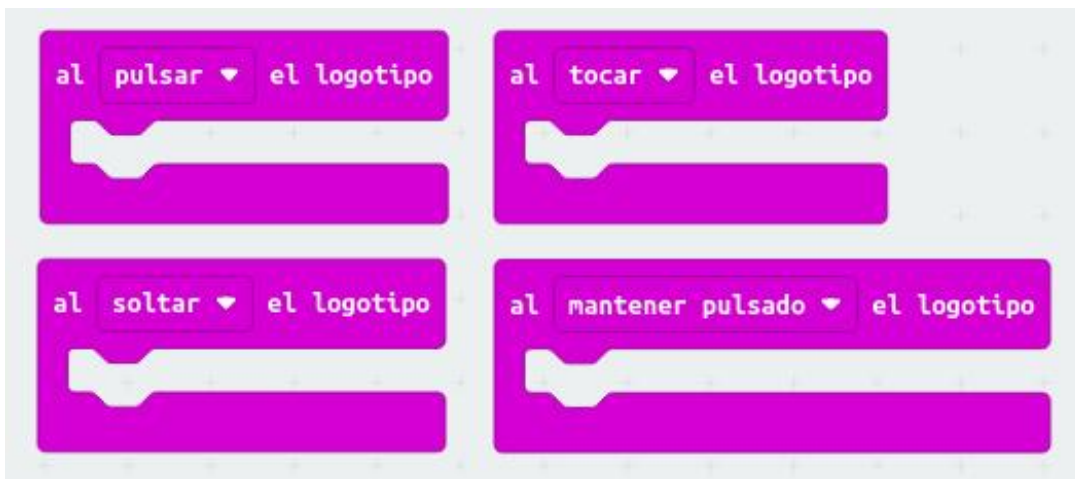
Para que micro:bit realice alguna acción específica al pulsar un botón, bastará con introducir los bloques de código necesarios dentro de un bloque de evento de pulsación. El siguiente programa mostrará una flecha apuntando hacia la izquierda al pulsar el botón **A**, una flecha apuntando hacia la derecha 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.

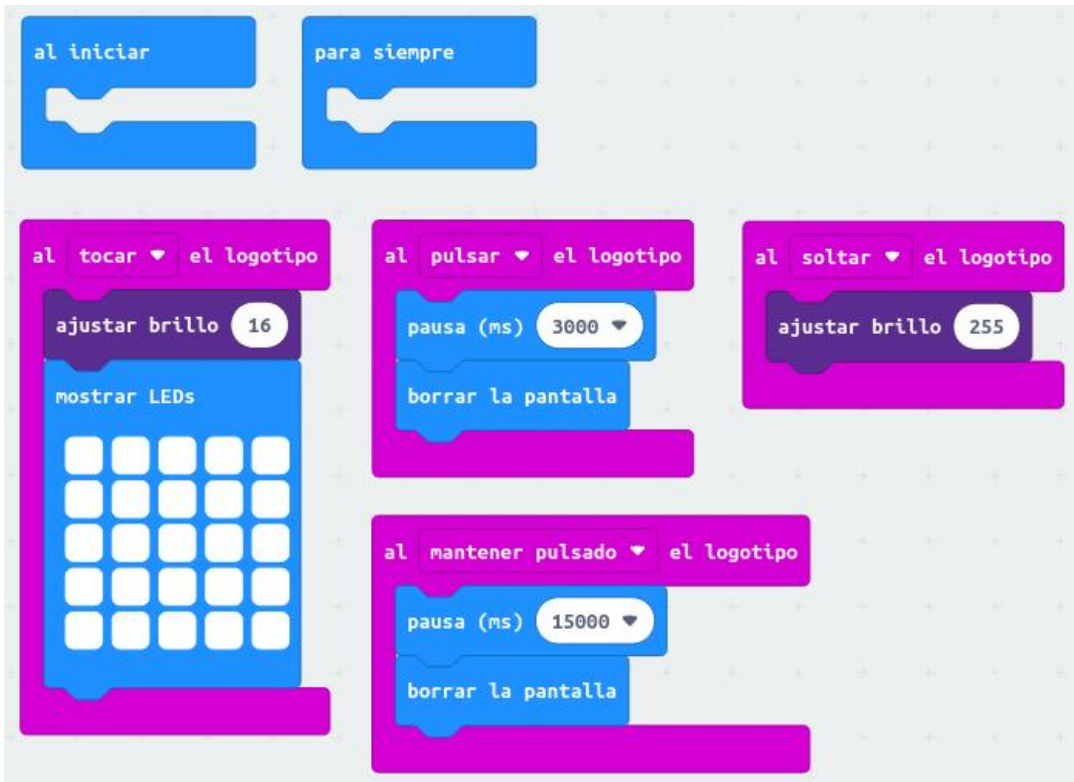


Como puede verse, los eventos asociados al logo son **pulsar**, **tocar**, **soltar** y **mantener pulsado**. **Cada vez que se toque el logo se producirán tres eventos:**

- 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.

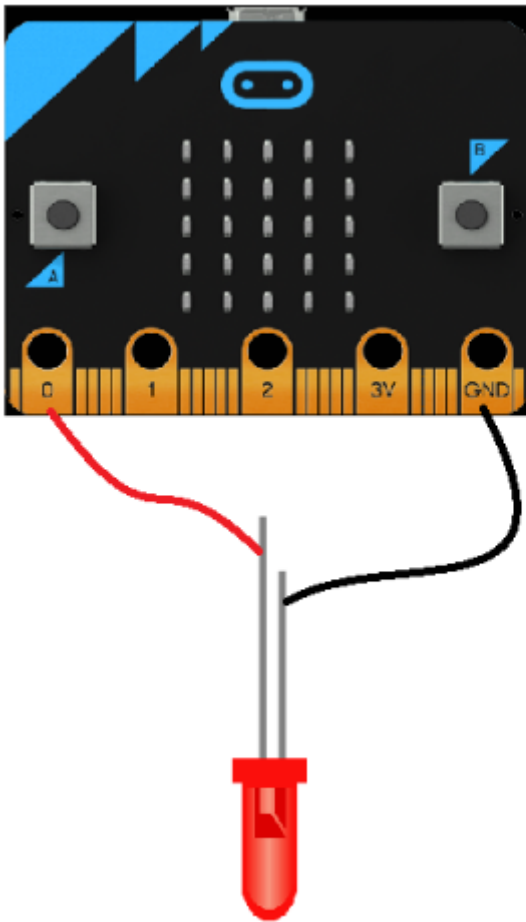
El siguiente programa simula una **luz temporizada**.



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.

Aprender con un led la diferencia entre analógico y digital

Vamos a conectar un led, el pin corto (-) al GND y el otro al pin0



Ulrich Pedersen Dah & Ture Reimer-Mattesén Center for Underisningsmidler CPU

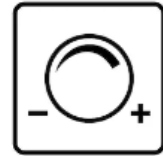
Vamos a ver la diferencia entre estos dos métodos de encender y apagar la luz

Digital



OFF ↔ ON

Analog



0 → 512 → 1023

Ulrich Pedersen Dah & Ture Reimer-Mattesens Center for Underisningsmidler CPU

El programa en digital es sencilllo

Digital



OFF ↔ ON

on button A ▼ pressed

digital write pin P0 ▼ to 0

show number 0

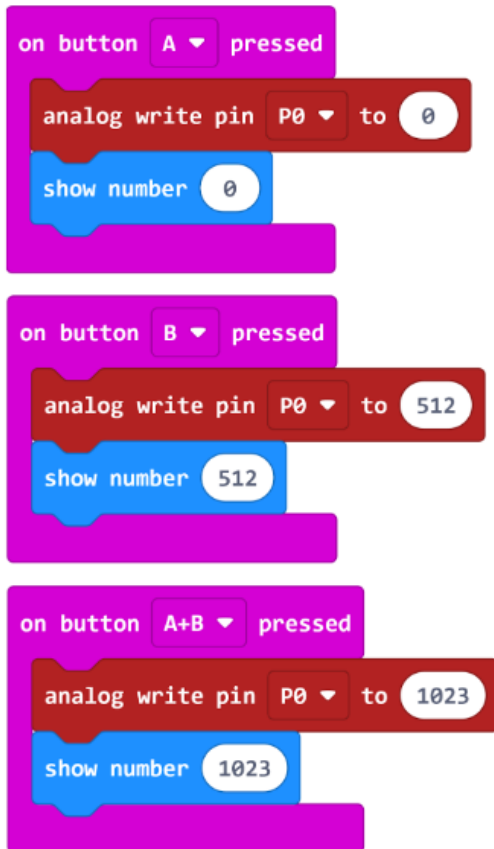
on button B ▼ pressed

digital write pin P0 ▼ to 1

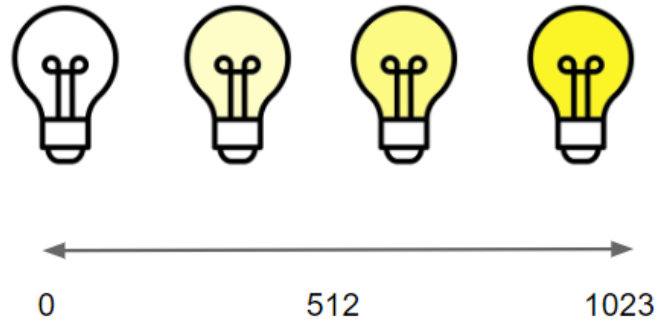
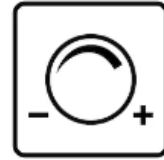
show number 1

Ulrich Pedersen Dah & Ture Reimer-Mattesens Center for Underisningsmidler CPU

Y el programa en analógico



Analog 1



Ulrich Pedersen Dah & Ture Reimer-Mattesen Center for Underisningsmidler CPU

Bloques de entradas: sonómetro, magnetómetro y acelerómetro con una sola línea de código

Sensor de sonido

Para usar los sensores integrados de micro:bit no es necesario cargar ni inicializar bibliotecas de código. Las medidas de los sensores se encuentran disponibles en el menú **Entrada** en forma de **variables**. En el lenguaje de bloques las variables se representan mediante rectángulos de extremos redondeados.

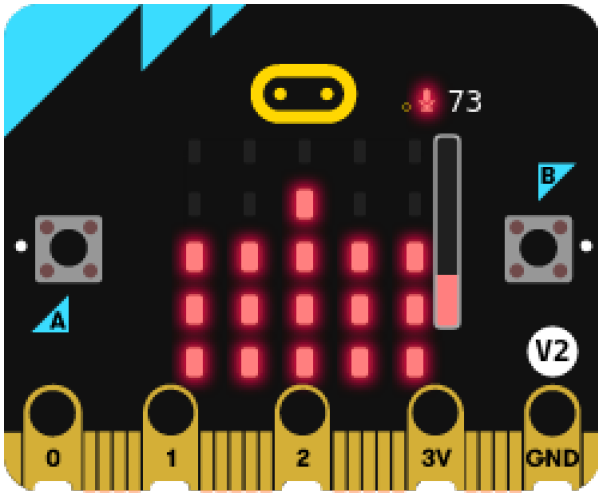
La versión 2 de micro:bit dispone de un **micrófono** que además de grabar sonidos puede medir el nivel de ruido. La **variable nivel de sonido** nos dará lecturas entre 0 (nivel mínimo de sonido) y 255 (nivel máximo). Estos niveles no se corresponden con ninguna unidad física, como el dB por ejemplo, y deben usarse con fines comparativos.

La razón de que algunos sensores de micro:bit proporcionen medidas entre 0 y 255, es que con un byte (8 bits) sólo se pueden representar $2^8 = 256$ números distintos, es decir, el 0 y los 255 primeros números naturales.

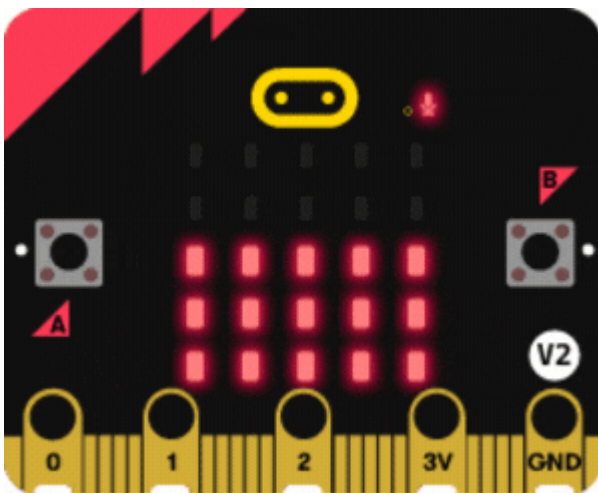
El bloque **plot bar graph of** del menú **LED** permite construir un sencillo medidor de sonido ambiente. Como 255 es un valor muy alto de intensidad de sonido, ajustamos el rango de medida de la barra, **up to**, a la mitad, es decir, a 128. Así la barra reflejará mejor el sonido de una voz o el sonido ambiental normal.



La variable **nivel de sonido** debe arrastrarse desde el menú **Entrada**. En el momento en el que la variable haya sido introducida en el programa, el simulador de micro:bit cambiará, mostrando una **barra ajustable** que simulará el nivel de sonido captado por el micrófono. El valor numérico del nivel de sonido simulado también será mostrado al lado del **LED** del micrófono.



Tras descargar el programa en la placa real, la matriz de LED representará continuamente el sonido recogido por el micrófono en forma de barra vertical. El **LED del micrófono** iluminado indicará que micro:bit está captando sonido.



Una tarjeta micro:bit ejecutando este programa puede agotar un par de pilas alcalinas IEC R03 (AAA) en unas 40 horas (Frost 2018). Para **ahorrar energía** y prolongar la autonomía del medidor podemos reducir tanto el **brillo de la pantalla** como el **número de medidas por segundo** que realiza el sensor. Para conseguir esto último introduciremos en el bucle **para siempre** un bloque **pausa (ms)**. Si el bloque se ajusta a 100 ms, el sensor sólo realizará 10 mediciones del nivel de sonido cada segundo.



Magnetómetro y acelerómetro

Con una mínima modificación, el código anterior puede usarse para monitorizar aquellas **magnitudes que puedan variar rápidamente**. Por ejemplo, podemos usar el sensor integrado de campo magnético (magnetómetro) para medir el campo magnético de la Tierra, el de una imán o el de una masa de hierro.

Podemos acceder al sensor mediante la variable **fuerza magnética (μT)**, que proporciona la **inducción magnética** medida en **microtesla**. Al cargar el programa, micro:bit comenzará a medir el campo magnético terrestre que varía, según la localización, entre 25 y 65 μT . Nótese que el magnetómetro no limita sus medidas al valor de 255.



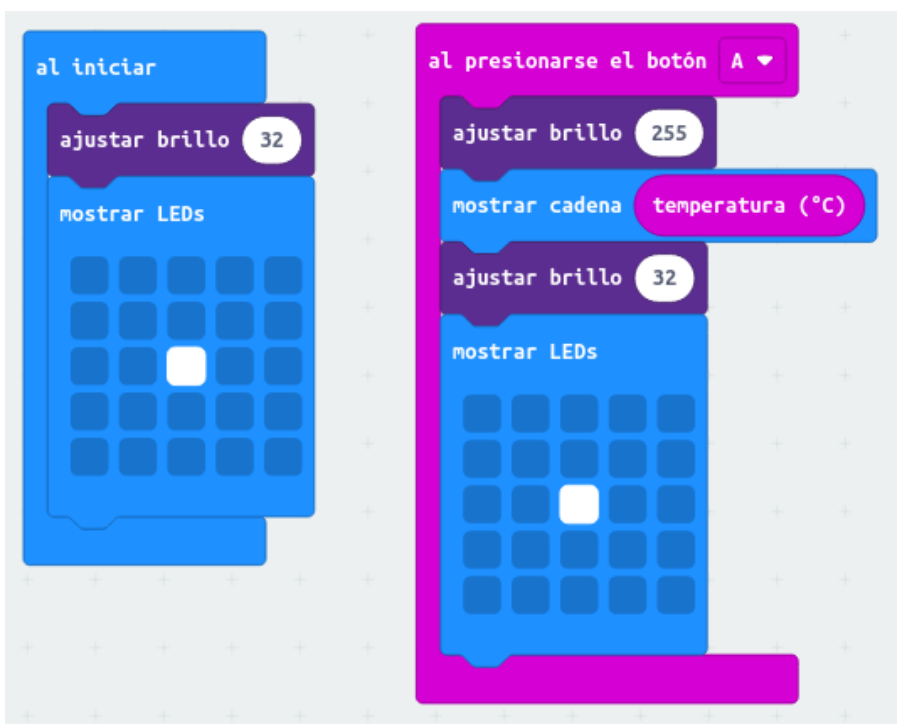
Otra medida interesante es la de la **aceleración de la placa**. La variable de acceso al acelerómetro se llama **aceleración (mg)** y proporciona las aceleraciones medidas en milésimas de g. Cuando la placa esté en reposo medirá la **aceleración de la gravedad terrestre**, que es de 1 g. Los movimientos bruscos de la placa en cualquier dirección deberían alterar el valor medido.



Bucles y eventos de tiempo: un termómetro

Programación del termómetro

El siguiente programa proporcionará el valor numérico de la **temperatura ambiente en grados Celsius** cada vez que se pulse el **botón A**. La variable **temperatura (°C)** se encuentra disponible en el menú **Entrada**.



El evento **al iniciar** comienza encendiendo un punto de la matriz de LED a modo de **piloto de funcionamiento**. El brillo de la pantalla se ajusta a un valor bajo para conseguir un **bajo consumo de energía**.

Por otro lado, **cada vez que se pulse el botón A**, ocurrirá un evento del tipo **al presionarse el botón A** que subirá el brillo de la pantalla al máximo (255) y mostrará la temperatura mediante una cadena de texto deslizable, para volver más tarde a dejar encendido el piloto de funcionamiento a bajo brillo.

La temperatura indicada será algo superior a la ambiental. Esto ocurre porque el sensor de temperatura se encuentra en el microprocesador y éste se calienta ligeramente cuando la placa está en funcionamiento. Martínez de Carvajal (2019) establece el error

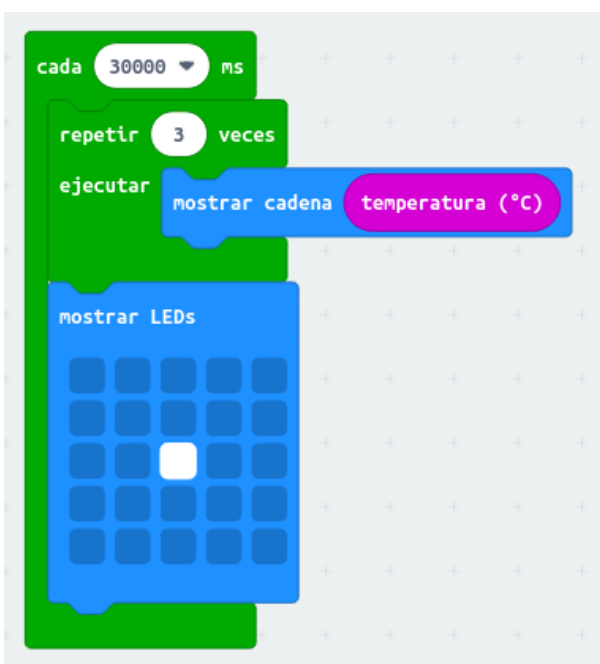
medio en 3°C, por lo que habrá que restar 3 al valor mostrado en pantalla para obtener la temperatura real. La **manipulación de la placa con los dedos** también contribuye al calentamiento y al error en la medida de la temperatura.

Podemos añadir un **evento de tiempo** para que el termómetro muestre la temperatura cada cierto tiempo. Para ello debemos usar el evento **cada ms**, dentro del menú **Bucles**.



Cada 30000 ms, o cada **30 segundos**, el programa mostrará la temperatura aunque no haya sido pulsado el botón A. Dentro del bucle de tiempo no se sube el brillo, así que los dígitos se mostrarán con bajo brillo para ahorrar batería.

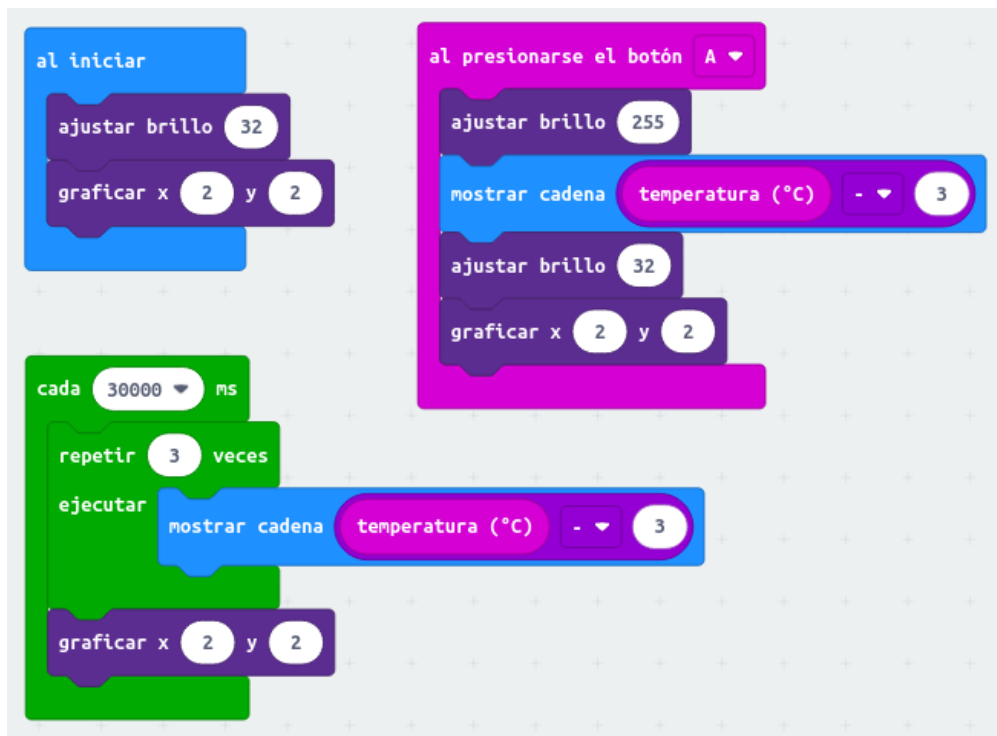
Es posible mejorar un poco más el programa haciendo que micro:bit muestre tres veces la temperatura cada 30 segundos. Podríamos repetir sin más la sentencia **mostrar cadena temperatura (°C)** tres veces dentro del bucle de tiempo, pero en su lugar vamos a usar un bucle del tipo **repetir veces**, que también se encuentra en el menú **Bucles**.



Nótese que el programa ejecuta un bucle cada 30 segundos, y que dentro de ese bucle se ejecuta otro bucle que muestra la temperatura tres veces seguidas. Al hecho de introducir un bucle dentro de otro se le llama **anidar bucles**.

Mejorando la lectura del sensor de temperatura

Para corregir el error de 3 grados Celsius en la lectura del sensor, bastará con restar 3 al valor de la variable **temperatura (°C)**. El menú **Matemática** contiene bloques para realizar operaciones aritméticas. Si se usa el bloque de resta - dentro de los bloques **mostrar cadena** resulta sencillo realizar la corrección necesaria. El código del termómetro completo quedará:



Los dos bloques **mostrar LED** han sido sustituidos por dos bloques **graficar x y** para conseguir que el código sea algo más compacto.

Bloques condicionales: luz crepuscular

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.

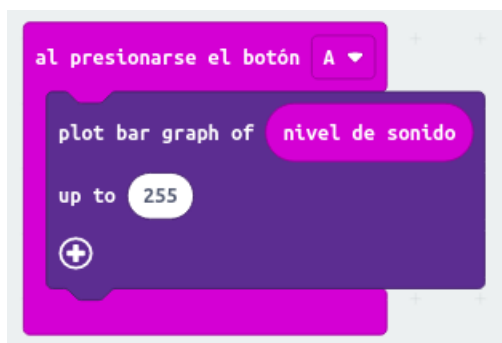
Para poder obtener buenas medidas de luz, la matriz de LED debe estar descubierta. Las fundas para micro:bit, especialmente las de color oscuro, pueden alterar las medidas del sensor de luz.

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**



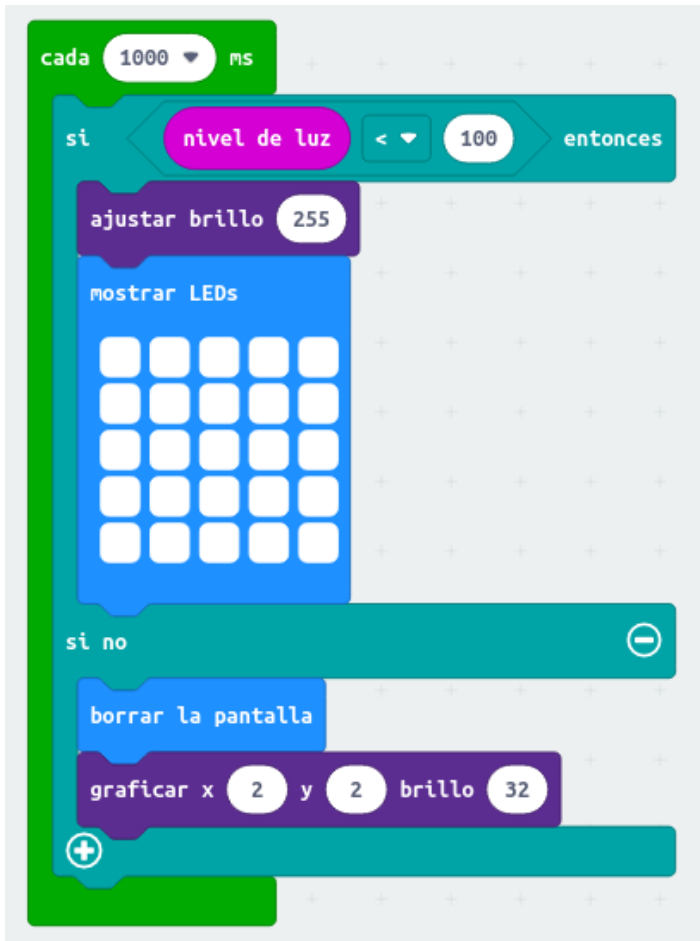
Luz crepuscular

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.

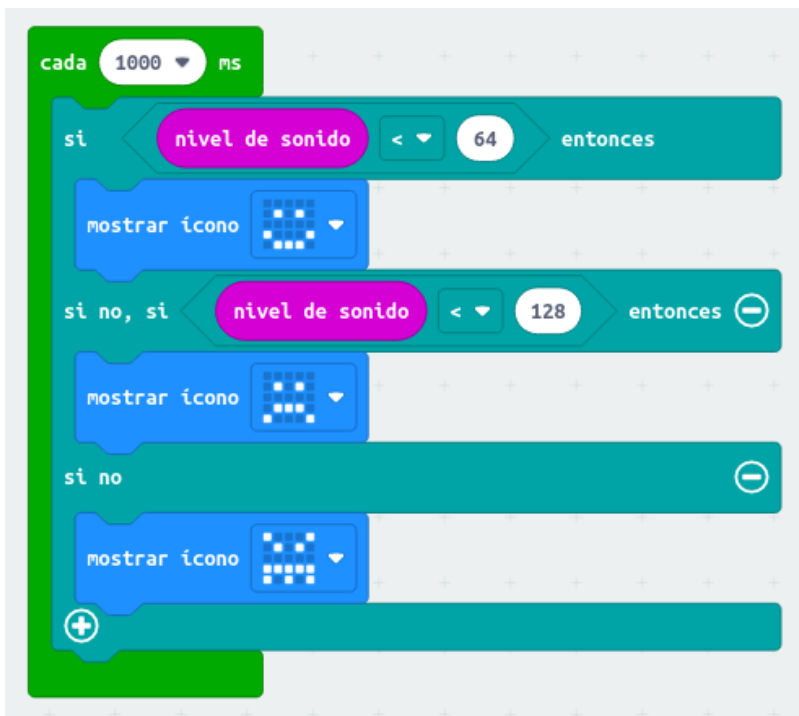
Dentro del evento de tiempo, usaremos la estructura **si entonces si no**, disponible en el menú **Lógica**, para encender con el máximo brillo la luz de la matriz de **LED** si la luz ambiental es inferior a 100 o para apagar la matriz en caso contrario. Tras el apagado de la pantalla dejaremos un pequeño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



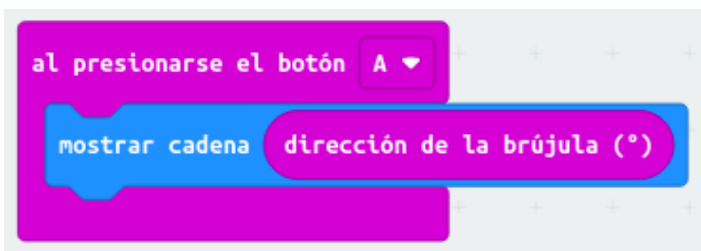
Así pues, si el nivel de sonido es inferior a 64, la pantalla mostrará un icono sonriente. En caso contrario, si el nivel de sonido es alto pero todavía inferior a 128, se mostrará una cara triste. Finalmente, si el sonido es igual a superior a 128, se mostrará una cara enfadada.

Brújula digital

Una brújula numérica

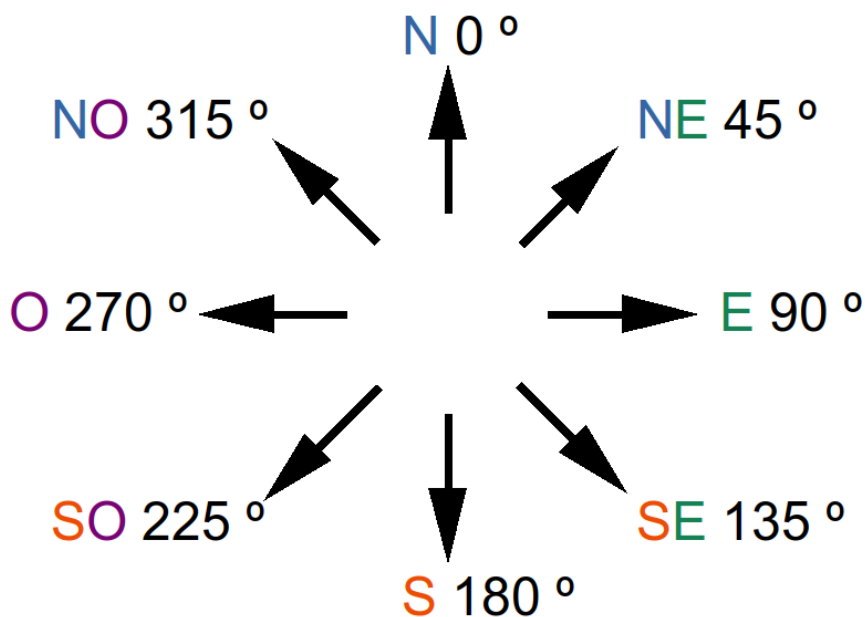
El sensor de campo magnético integrado en micro:bit también puede usarse para medir el **campo magnético terrestre**. De esta forma es posible determinar la dirección del norte magnético y programar así una brújula.

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 (°)** se encuentra disponible en el menú **Entrada**.

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



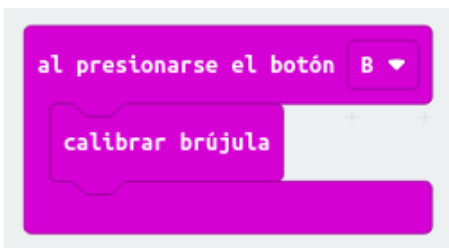
Por ejemplo, y de acuerdo con la rosa de los vientos, una lectura de 340° indicará que el logotipo de micro:bit apunta en dirección casi norte, concretamente nornoroeste.

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.

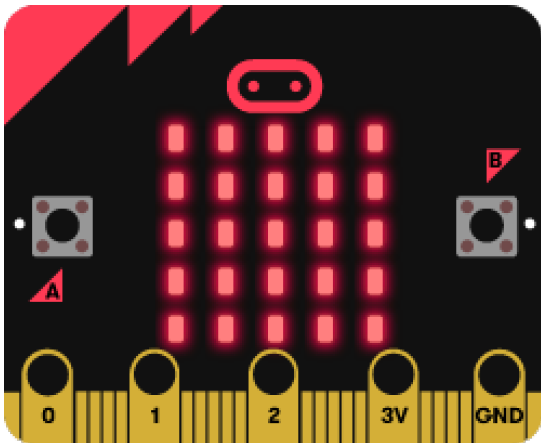
Calibrando el compás

Al igual que la brújula de un teléfono móvil, el compás de micro:bit necesita ser calibrado para proporcionar medidas fiables. Los puntos suspensivos **... más** bajo el menú **Entrada** despliegan otro menú que contiene el bloque **calibrar brújula**.



Añadiendo el **evento** mostrado, nuestra brújula digital permitirá ser calibrada cada vez que se pulse el botón B.

El proceso de calibración es muy simple, ya que sólo requiere mover la placa en todas las direcciones hasta que todos los **LED** de la pantalla estén iluminados.



Una 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**.

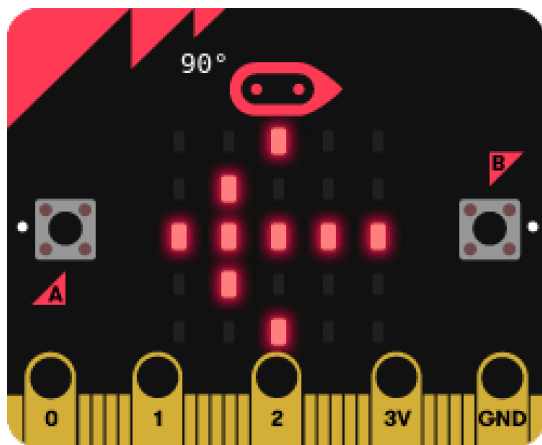


Sin embargo, este programa sólo mostrará la flecha cuando el logo apunte al norte, borrando la pantalla en caso contrario. Para que haya una flecha en pantalla apuntando permanentemente al norte, habrá que pulsar sobre el signo + para añadir líneas al bloque condicional.



Nótese que cuando el logo de micro:bit apunta al este, $\text{dirección de la brújula (°)} < 135$, 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.

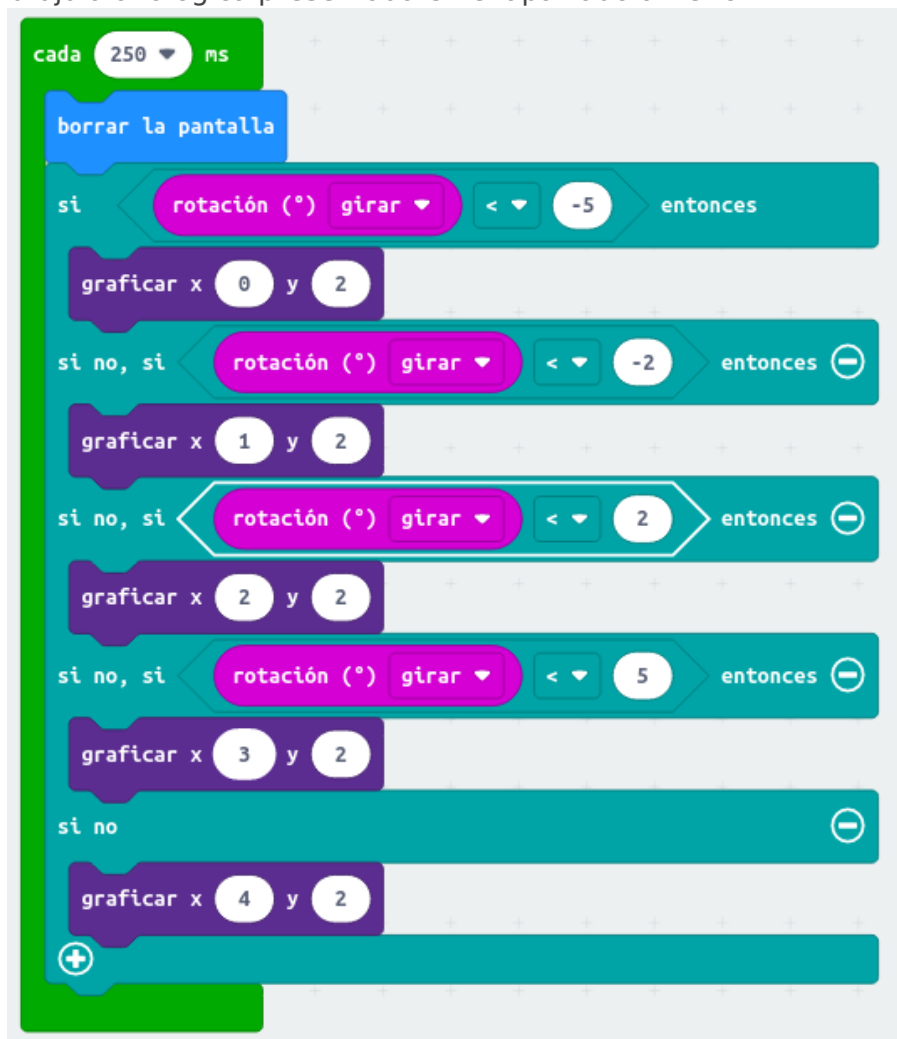


Nivel de burbuja usando el giroscopio

Mediante el **sensor de fuerza**, micro:bit puede determinar para cada uno de los tres ejes coordenados las proyecciones de la aceleración de la gravedad y, a partir de ellas, el **giro de la placa** con respecto al plano horizontal.

Vamos a usar la medida del giro de la placa para programar un sencillo nivel de burbuja. La burbuja será un punto luminoso en la pantalla **LED** de micro:bit. Cuando el punto se encuentre en el centro de la pantalla, cuyas coordenadas son (2,2), micro:bit estará nivelado. Si micro:bit está desnivelado hacia la izquierda o hacia la derecha, el punto se dibujará desplazado en esas direcciones.

La estructura del código, compuesto por múltiples sentencias condicionales, es muy similar al de la brújula analógica presentada en el apartado anterior.



El código consta de **un único evento temporal** dentro del cual se evalúa el giro de la placa cada 250 ms y que, en función del ángulo de inclinación, enciende el punto correspondiente. Por ejemplo, si la placa se inclina hacia la izquierda con una rotación inferior a -5° , se encenderá el punto situado más a la izquierda, cuyas coordenadas son (0,2). En caso contrario, si la placa está inclinada hacia la izquierda menos de -2° , se encenderá el siguiente punto, de coordenadas (1,2), y así sucesivamente.

