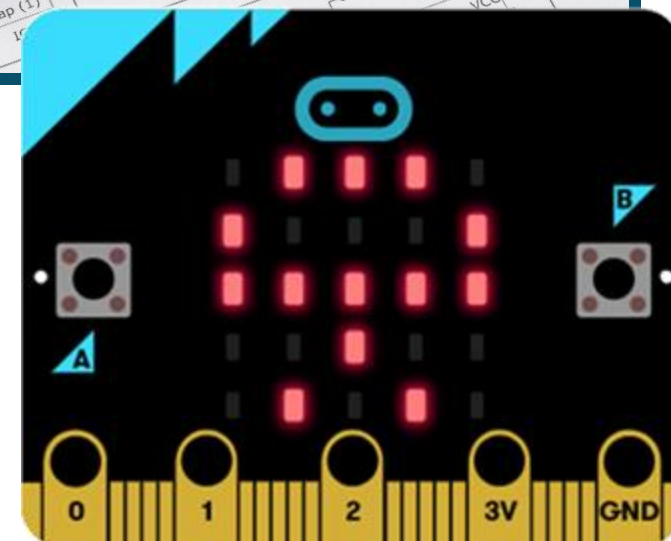
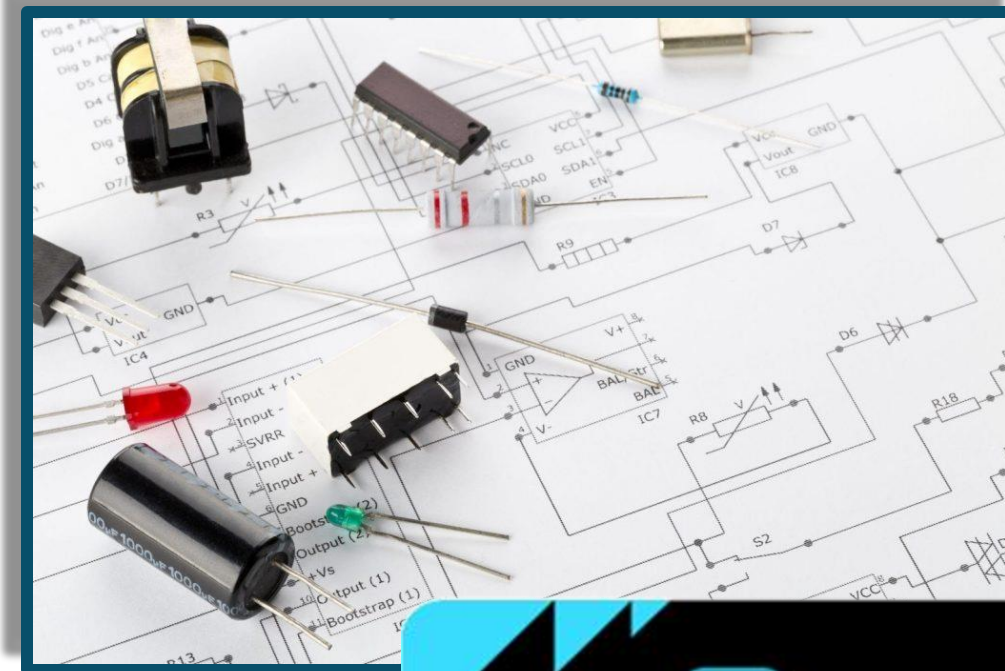


Programa Código Escuela 4.0

Primeras clases con Micro:bit y Maqueen ESO.



Índice

- Introducción
- Robótica.
- Como cargar el programa en la micro:bit.
- Como emparejar el ordenador a la micro:bit.
- Como descargar el programa del ordenador .
- Programa 1. Corazón palpitante.
- Programa 2. Insignia de emoción.
- Sensores.
- Programa 3. Sensores.
- Programa 4. Sensor y gráfica de barras.
- Programa 5. Lámpara inteligente.
- Programa 6. Not.
- Programa 7. Lámpara con palmada.
- Maqueen, librerías y extensiones.
- Motores
- Programa 8. Moviendo a Maqueen.
- Programa 9. Esquiva obstáculos.
- Programa 10. Sigue manos.
- Programa 11. Control manos.
- Otros programas



Introducción

La placa **micro:bit** es una herramienta sencilla y motivadora para iniciar al alumnado de Primaria en la programación y la robótica. Su entorno visual permite crear proyectos rápidamente, lo que favorece la creatividad, la experimentación y el pensamiento computacional.



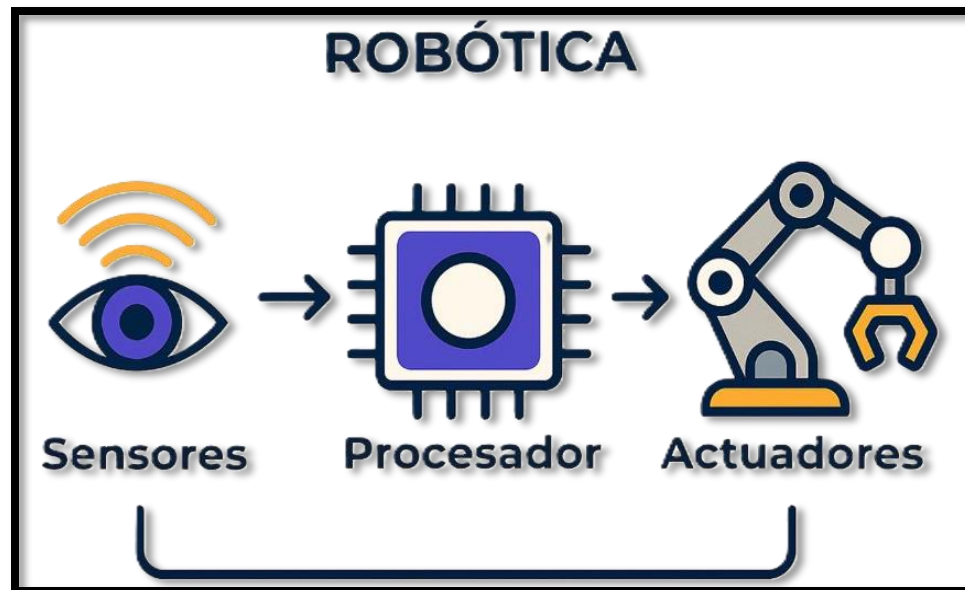
En esta presentación trabajaremos fundamentalmente la **estructura condicional**, viendo la gran variedad de resultados que podemos obtener cambiando los sensores y actuadores.

También introducimos el uso de la **puerta NOT** en varios programas.

Para cada práctica habrá una pequeña explicación, tutorial y el propio programa para descargar.

Robótica

La **robótica** es una rama de la tecnología que combina **mecánica**, **electrónica** e **informática** para diseñar y construir robots capaces de percibir su entorno mediante **sensores**, procesar información con un **controlador o procesador**, y responder en el mundo real a través de **actuadores**.



Su objetivo es crear máquinas que puedan realizar tareas de forma autónoma o semiautónoma, desde movimientos simples hasta acciones complejas.

Como cargar el programa en la micro:bit

Para programar la placa **micro:bit** utilizamos la [página de MakeCode](#). Hay dos maneras de hacerlo: emparejar la micro:bit al ordenador y descargar el programa.

Emparejar la placa

Al emparejar la placa podemos pasar los programas a la micro:bit con un solo clic.

Además, si se vuelve a conectar la misma placa al mismo ordenador normalmente se acuerda y no hace falta volver a realizar el proceso.

También permite transmitir la información de la micro:bit al ordenador.

El mayor inconveniente es que a veces se desvincula y hay que reiniciar la conexión.

No se puede utilizar el navegador Firefox para emparejar la placa.

Descargar el programa

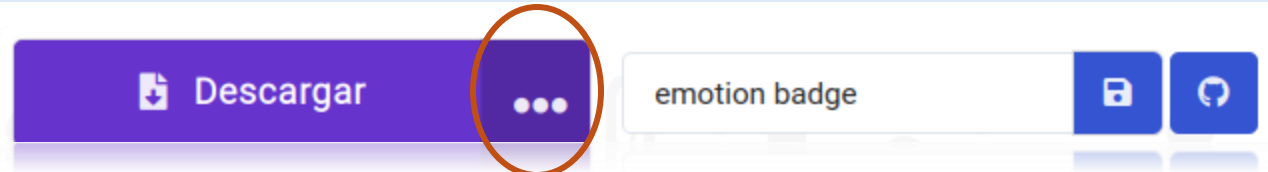
Descargar el programa es más laborioso, pero funciona siempre.

No permite transmitir la información de la placa al ordenador.

El alumnado tendrá que manejarse con carpetas y mover documentos: puede ser un inconveniente o una ventaja si queremos que aprendan a manejar el sistema operativo.

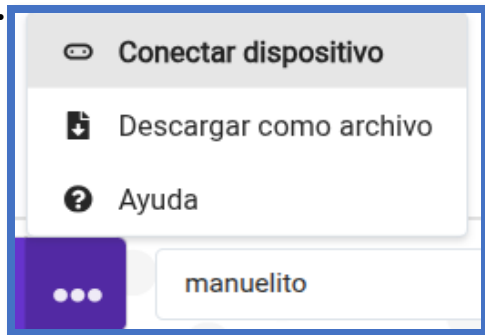
Como emparejar el ordenador a la micro:bit

Para programar la placa **micro:bit** utilizamos la [página de MakeCode](#). Ahora vamos a aprender como pasar el programa del ordenador a la placa emparajendo la tarjeta.

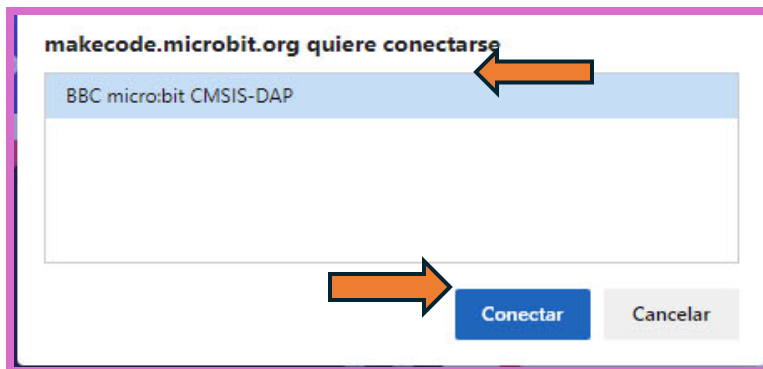


El primer paso es conectar la micro:bit al ordenador con el cable USB. Después podemos:

Le damos a los tres puntitos, al hacerlo nos aparecen varias opciones, escogemos **conectar dispositivo**.



Nos aseguramos de que la micro:bit está conectada y le damos a siguiente. Nos aparece un mensaje de confirmación y presionamos listo.

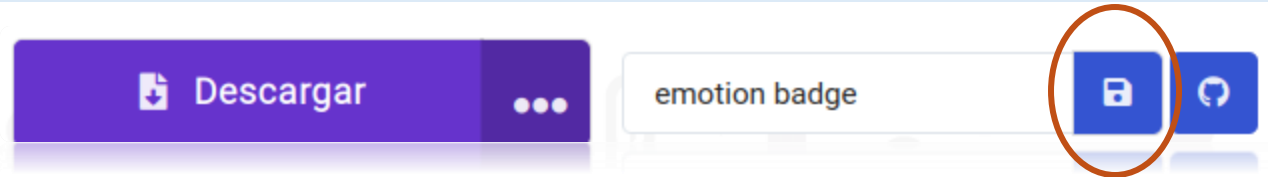


En la parte de arriba nos aparecerá un mensaje. Primero seleccionamos el nombre de la micro:bit y luego le damos a conectar.

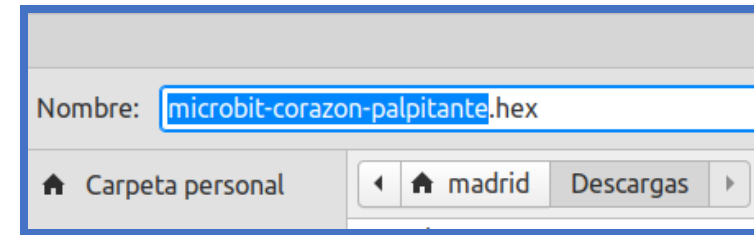
Tras hacer estos pasos para pasar el programa a la micro:bit tenemos que darle al botón descargar (a la izquierda de los tres puntitos).

Como descargar el programa del ordenador

Para programar la placa **micro:bit** utilizamos la [página de MakeCode](#). Ahora vamos a aprender como pasar el programa del ordenador a la placa descargándolo.

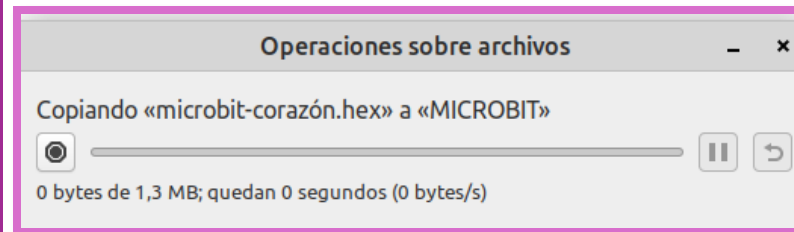


Al darle al botón con forma de disquete se descarga el programa en nuestro ordenador. Nos aparecerá una pantalla desde la que podemos elegir el nombre del programa y dónde lo descargamos.

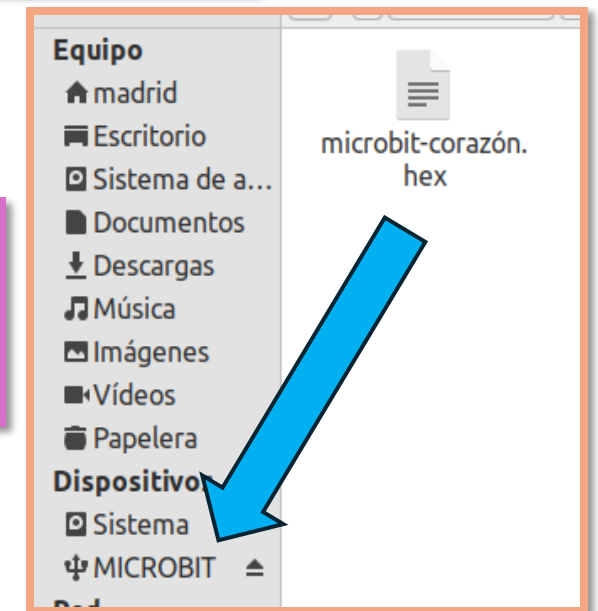


Una vez descargado abrimos la carpeta dónde está el programa, normalmente Descargas.

Ahora solo tenemos que arrastrar el archivo a la carpeta que pone MICROBIT. Al hacerlo nos aparecerá un mensaje diciendo que se está cargando, al finalizar se cierra esa ventana.

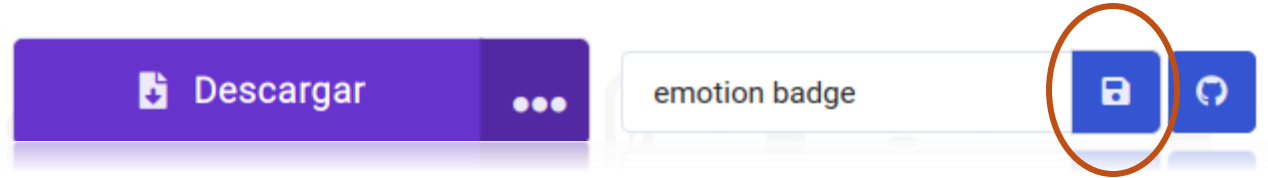


También se puede copiar el archivo usando el botón derecho o Ctrl+C y pegarlo en la carpeta de MICROBIT.



Cómo guardar y compartir proyectos

Tal y cómo hemos visto antes podemos ir a la [página de MakeCode](#) y descargar el programa.

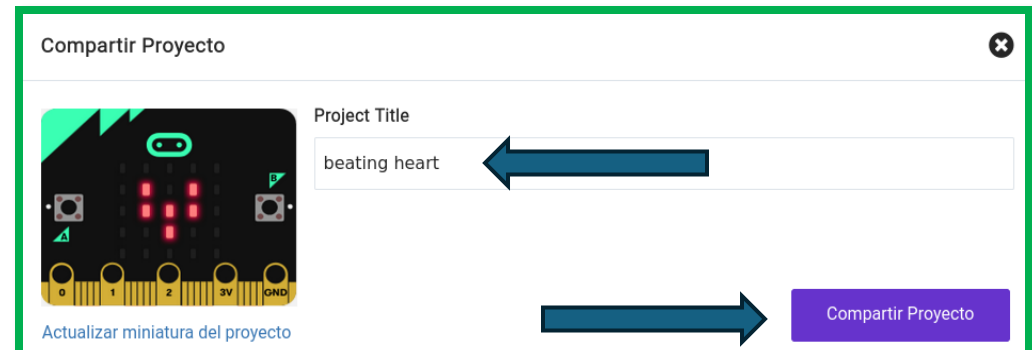


El programa descargado lo podemos subir a una nube, guardar en un USB, mandarlo por correo electrónico,...

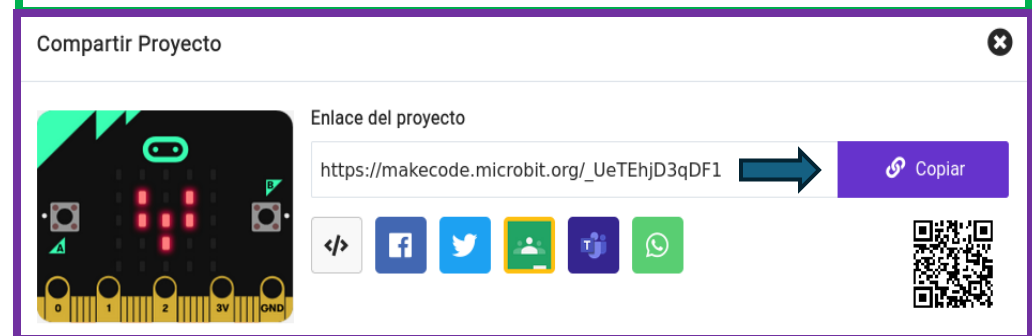
Otra opción es ir a la parte superior derecha de la página y dar al botón de compartir.



Se abre una ventana desde la que podemos cambiar el nombre al proyecto. Después seleccionamos compartir proyecto.



Ahora podemos copiar el enlace y compartirlo con quien queramos. También podemos guardarlo en lugar del archivo para modificarlo posteriormente si lo deseamos.



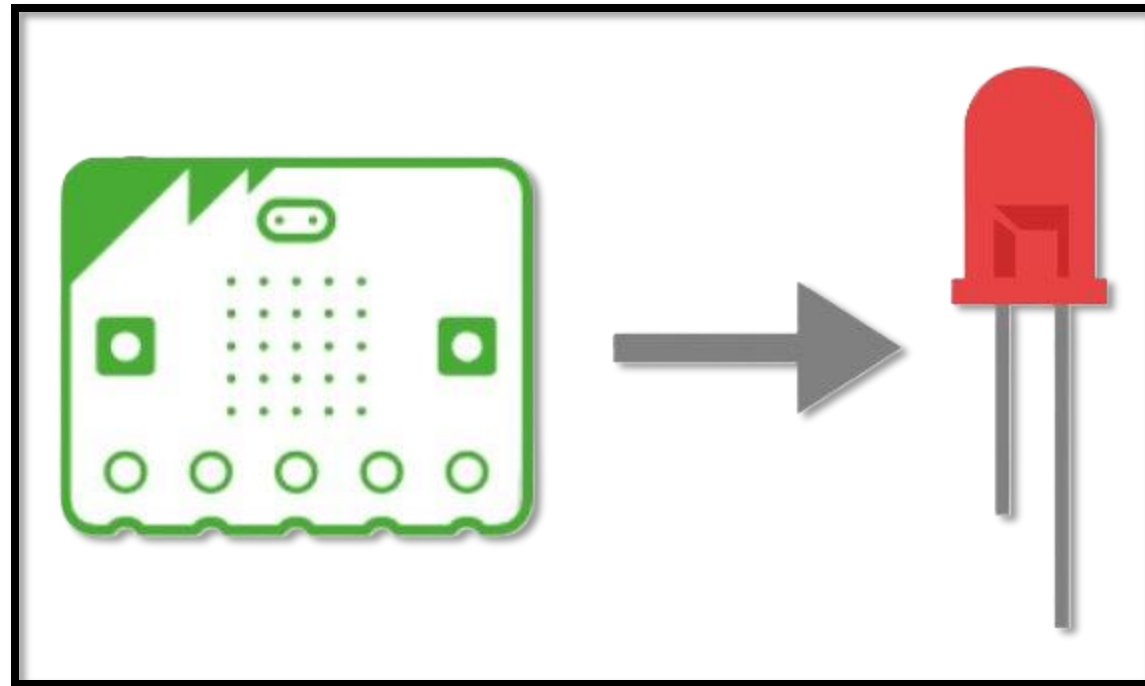


Programa 1: Corazón palpitante. Explicación.

Vamos a crear un programa **secuencial** en el que vayan apareciendo distintas imágenes



En este caso la programación va directamente a los actuadores, en concreto al display LED.





Programa 1: Corazón palpitante



Descripción: crea un programa **secuencial** en el que aparezca un corazón grande durante 500 ms y luego uno pequeño durante 500 ms.

Cuando hayas terminado:

- Prueba a cambiar los tiempos de espera y observa cómo se modifica el programa.
- Prueba a utilizar otros íconos para crear una secuencia que te guste. [Aquí tienes un ejemplo.](#)

Bloques que vamos a usar:



Básico

mostrar ícono



pausa (ms)

100

[Enlace al tutorial](#)

[Enlace al programa](#)

para siempre

mostrar ícono



pausa (ms)

500

mostrar ícono



pausa (ms)

500



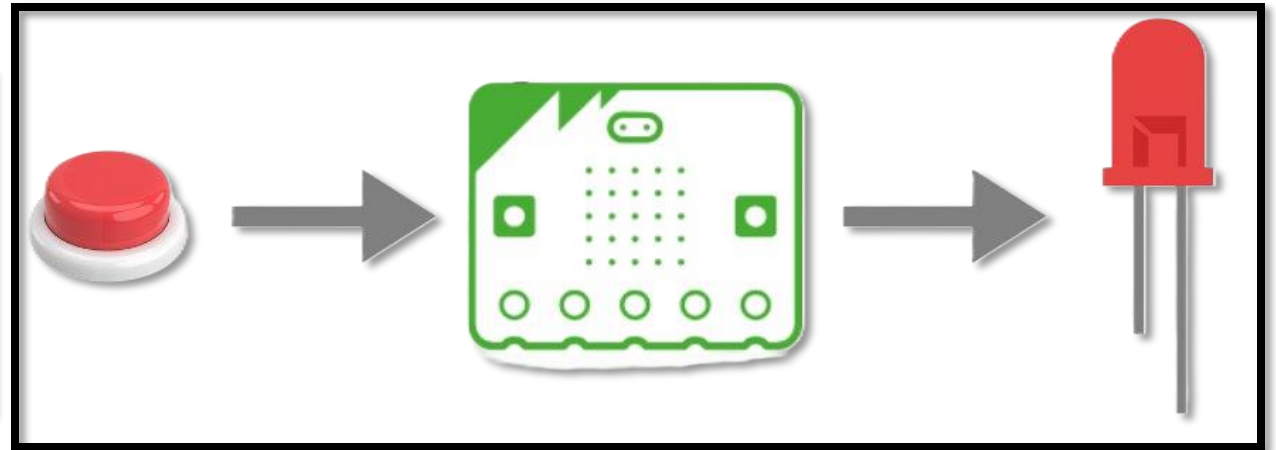
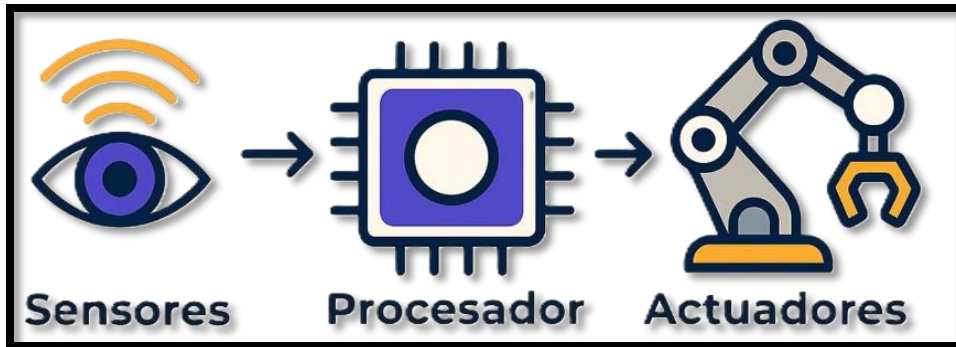
Programa 2: Insignia de emoción. Explicación.



Vamos a crear un programa en el que aparezcan **distintas imágenes según presionemos el botón A o el botón B.**

Al utilizar el bloque al presionar el botón A estamos utilizando eventos. Los **eventos** son **acciones o sucesos que desencadenan la ejecución de un conjunto de bloques**. Funcionan como “disparadores”: cuando ocurre algo específico, el programa responde ejecutando las instrucciones asociadas. Después de hacerlo volverá a la programación del ***para siempre***.

Podemos considerar los pulsadores como sensores ya que recogen información del entorno (si lo estamos presionando o no).



El pulsador envía información a la placa diciendo si está siendo accionado o no. La placa según esté programada responderá mandando a los LEDs que se enciendan.



Programa 2: Insignia de emoción



Descripción: crea un programa **secuencial** en el cuál cuando pulsas el botón A, el programa reacciona mostrando una cara feliz en la pantalla. Cuando pulsas el botón B muestra una carita triste. ***Presionar el botón A o B es un evento que dispara la programación de ese bloque.***

Cuando hayas terminado:

- Prueba otros iconos como dormido, confundido. También otras entradas, como agitado.
- Prueba a crear tus propios emoticonos usando la pantalla LED.

Bloques que vamos a usar:



Básico



Entrada

mostrar ícono

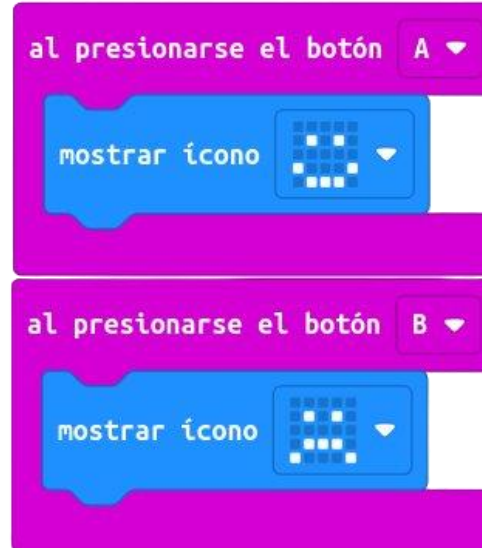


al presionarse el botón A



[Enlace al tutorial](#)

[Enlace al programa](#)

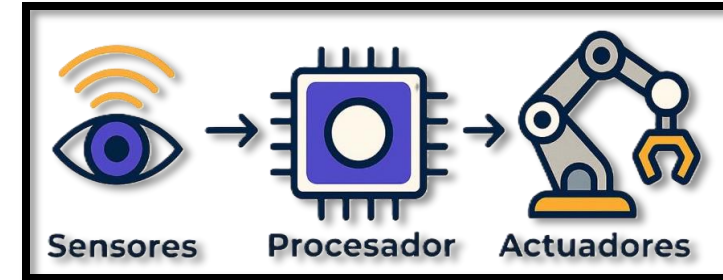




Sensores



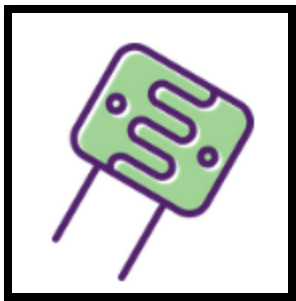
A partir de ahora vamos a incluir la información que los **sensores** recaban del exterior. Aunque sean programas secuenciales y no tomen decisiones **estamos interactuando con el ambiente.**



Los sensores en la micro:bit suelen dar valores entre 0 (mínimo) y 255 (máximo) . **0** podría significar “nada de luz”, o “sin sonido”, según el sensor. **255** sería “máxima luz”, o “mucho sonido”. Los valores entre medias (como 120, 180, 30...) representan niveles intermedios.

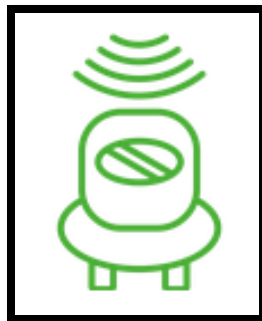
Una excepción es el sensor de temperatura que nos dará un valor en °C.

Los sensores que más vamos a usar son:



Sensor de luz.

Está en la primera fila de LEDs. Va de 0 a 255.



Sensor de ruido.

Micrófono, está arriba a la derecha. Va de 0 a 255.



Pulsadores.

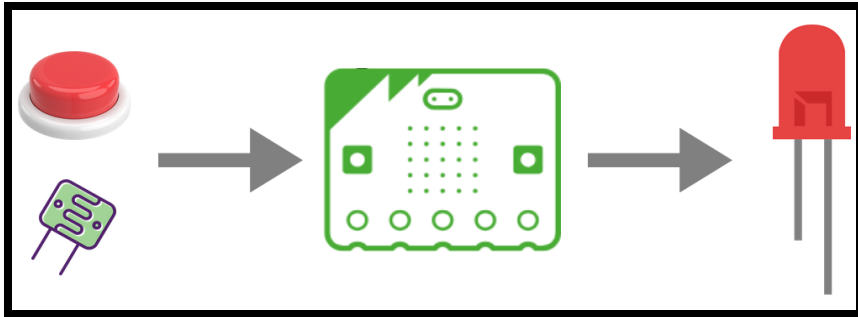
Al accionarlos envían un uno, sino envían un cero.



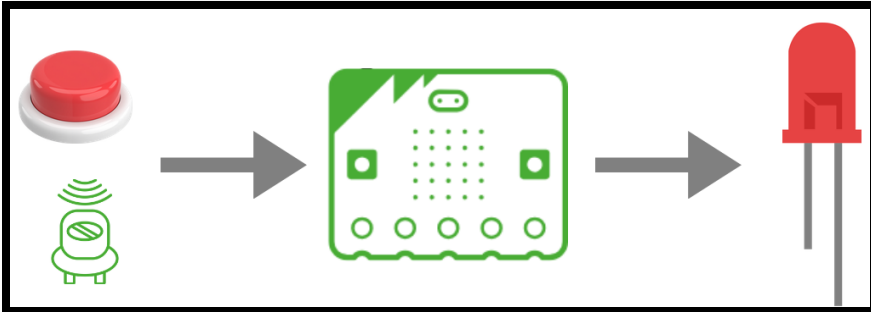
Programa 3: Sensores. Explicación.



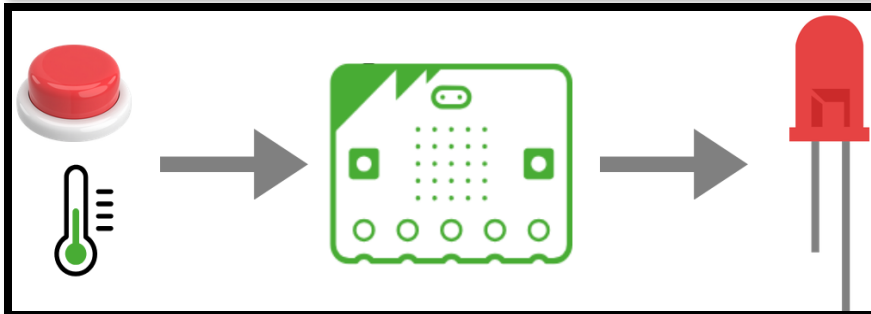
Vamos a utilizar distintos eventos para visualizar el valor que obtenemos de distintos sensores.



Al accionar un pulsador mandamos a la placa el valor del sensor de **luz**. Esta envía la información a los LEDs para visualizarla.



Al accionar un pulsador mandamos a la placa el valor del sensor de **ruido**. Esta envía la información a los LEDs para visualizarla.



Al accionar un pulsador mandamos a la placa el valor del sensor de **temperatura**. Esta envía la información a los LEDs para visualizarla.



Programa 3: Sensores



Descripción: crea un programa **secuencial** en el cuál cuando pulsas el botón A, el programa muestra el valor del nivel de luz, cuando presionas B el valor de la temperatura y al pulsar el logotipo muestra el nivel de sonido. ***Estamos usando el sensor de luz, el de temperatura y el micrófono de la placa.***

Cuando hayas terminado:

- Prueba a cambiar la luz que le llega, temperatura,... y observa como cambian los valores.
- Prueba otros sensores de la placa.

Bloques que vamos a usar:



Básico



Entrada

mostrar ícono



mostrar número

0

al presionarse el botón A

al pulsar el logotipo

nivel de luz

temperatura (°C)

nivel de sonido

al pulsar el logotipo

mostrar cadena "S"

mostrar número nivel de sonido

al presionarse el botón B

mostrar cadena "T"

mostrar número temperatura (°C)

al presionarse el botón A

mostrar cadena "L"

mostrar número nivel de luz

[Enlace al programa](#)

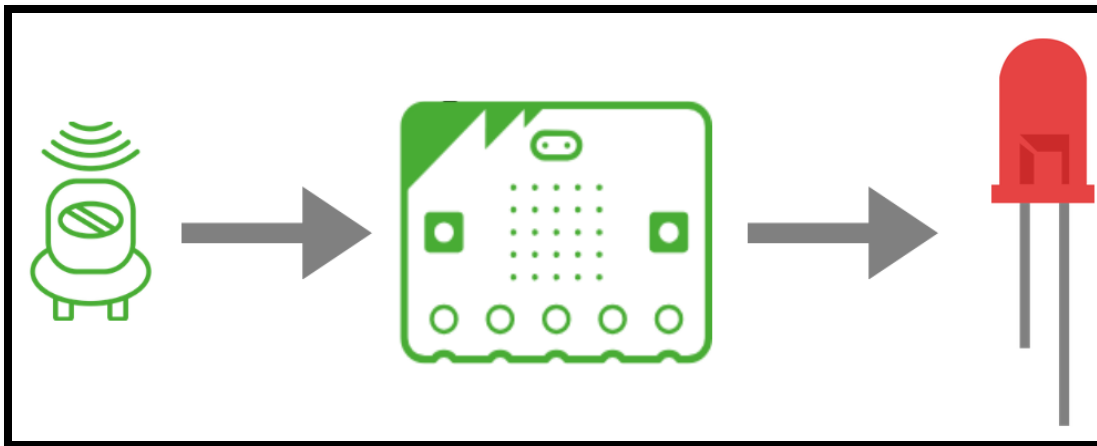
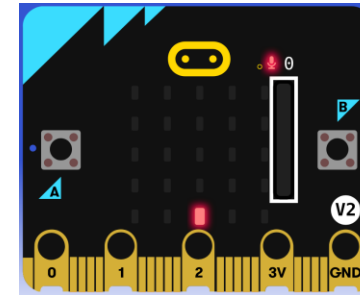
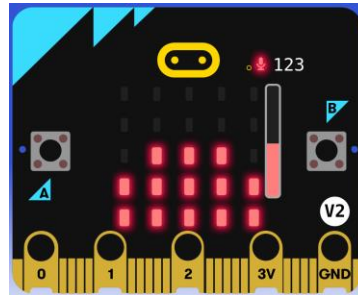
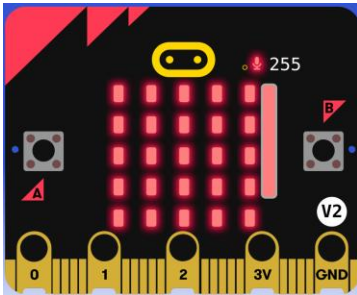


Programa 4: Sensor y gráfica de barras. Explicación.



A veces el valor numérico que nos da un sensor no nos aporta mucha información. En esta práctica vamos a aprender a visualizar la medida del sensor de otra manera.

El número de LEDs encendidos será proporcional a la medida del sensor. Si está en el máximo todos los LEDs estarán iluminados, en el mínimo todos apagados.



Mandamos a la placa el valor del sensor de **ruido**. Esta envía la información a los LEDs para visualizarla.



Programa 4: Sensor y gráfica de barras



Descripción: crea un programa **secuencial** en el que la micro:bit nos muestre el valor de un sensor usando la gráfica de barras del display LED. Cuanto mayor sea el valor más LEDs estarán iluminados. Resulta útil para ver cómo varía la información que recoge un sensor.

Cuando hayas terminado:

- Prueba a cambiar la luz que le llega a la placa y observa cómo cambia el display de LEDs.
- Prueba con otros sensores de la placa.

Bloques que vamos a usar:



[Enlace al programa](#)



El primer valor del bloque indica el número que queremos graficar y el segundo el valor máximo que puede tener.

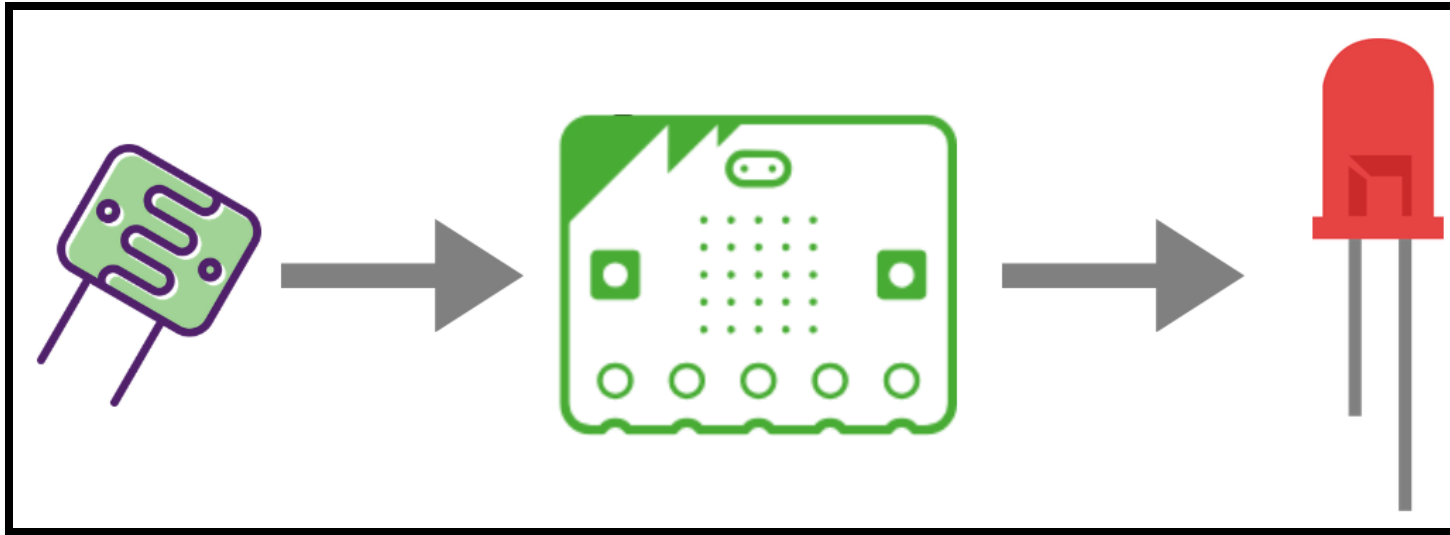


Programa 5: Lámpara inteligente. Explicación.



Vamos a empezar con los programas **condicionales**: el programa **elige qué hacer dependiendo de si una condición se cumple o no**.

A lo largo del resto de prácticas vamos a utilizar estructuras de programación muy similares a esta para aprender cuán útil puede llegar a resultar.



Mandamos a la placa el valor del sensor de **luz**. La **condición** para que los LEDs se enciendan es que el valor obtenido sea menor que el deseado. En caso contrario se apagarán.



Programa 5: Lámpara inteligente



Descripción: crea un programa **condicional** en el que se ilumine la pantalla de la micro:bit cuando detecte que no hay luz.

Cuando hayas terminado:

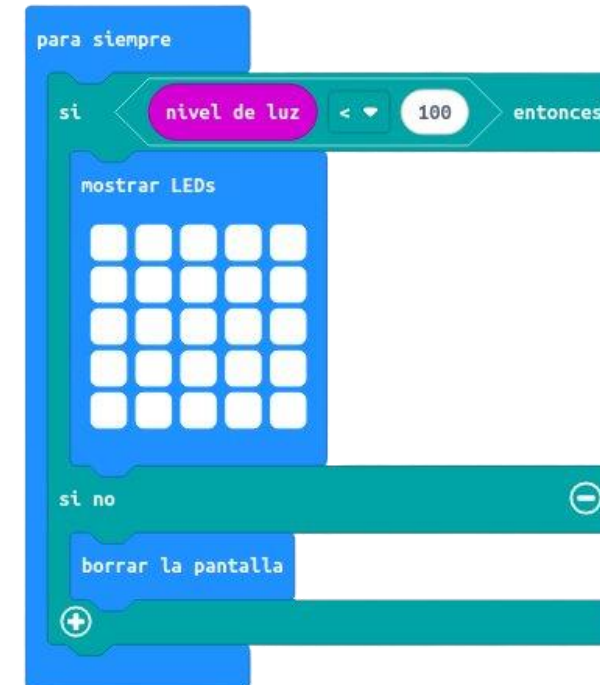
- Prueba a poner otro icono cuando no hay luz, por ejemplo, una luna o una estrella.
- Utiliza distintas iluminaciones o iconos en función de la luz detectada.

Bloques que vamos a usar:



[Enlace al tutorial](#)

[Enlace al programa](#)

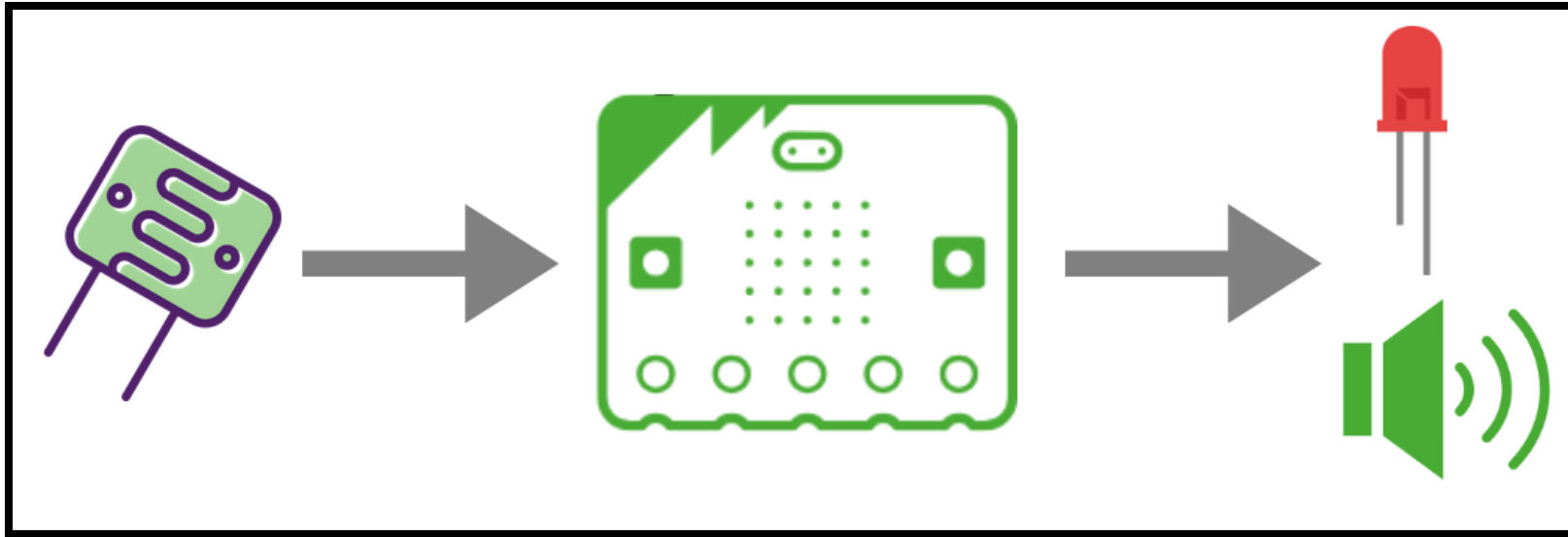




Programa 6: Detector de presencia. Explicación.

Este programa es muy similar al anterior. Lo que queremos hacer es **cuando disminuya la luz** (una persona pase delante de la placa o simplemente pasemos la mano por delante) **se active una alarma**.

Para que el programa funcione correctamente es importante ajustar los tiempos de espera y el valor que va hacer que salte la alarma.



En esta práctica nuestros actuadores serán el display de LEDs y el altavoz de la placa.



Programa 6: Detector de presencia.



Descripción: crea un programa **condicional** en el que cambie el icono que se muestra en la placa y suene una alarma cuando disminuye la luz (como si alguien pasara por delante).

Cuando hayas terminado:

- Prueba con distintos valores de nivel de luz hasta ajustarlo a las condiciones de tu clase.
- Crea un detector de ruido que muestre el icono X cuando sobrepase un cierto valor.

Bloques que vamos a usar:



Básico



Entrada



Lógica



Música

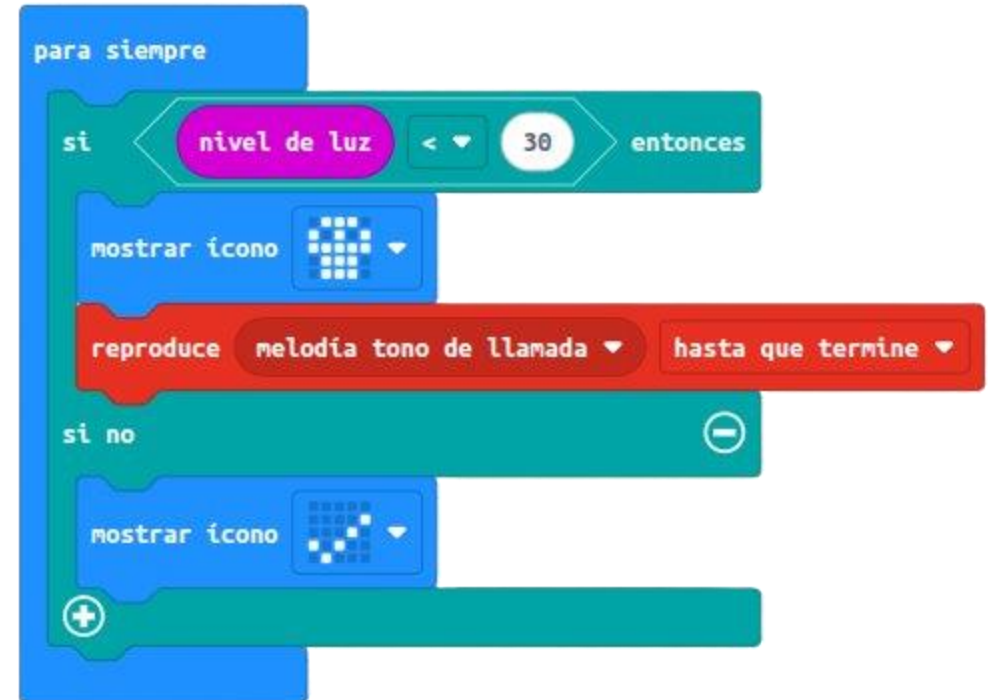


nivel de luz



Presiona las flechas para cambiar la melodía y el modo en que se reproduce.

[Enlace al programa](#)





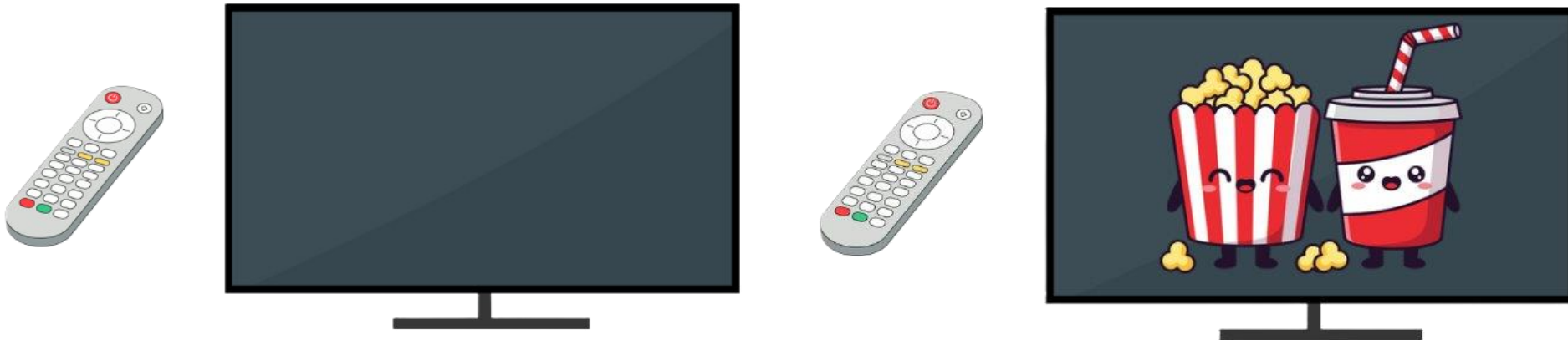
Not



La función lógica NOT sólo se puede utilizar cuando estamos trabajando con lógica booleana, es decir, la variable sólo puede tomar dos valores: 1 o 0.

Se suele utilizar la expresión **negar una variable**, esto significa:

- Si el valor de la variable es uno al negarla pasará a ser cero.
- Si el valor de la variable es cero al negarla pasará a ser uno.



Para entenderlo mejor pensemos en el funcionamiento de un mando a distancia de una televisión.

Cuando está **apagada (valor = 0)** al presionar el botón se **enciende (valor = 1)**.

Si queremos apagar la televisión utilizamos el mismo botón, que pasar del valor 1 al valor 0.



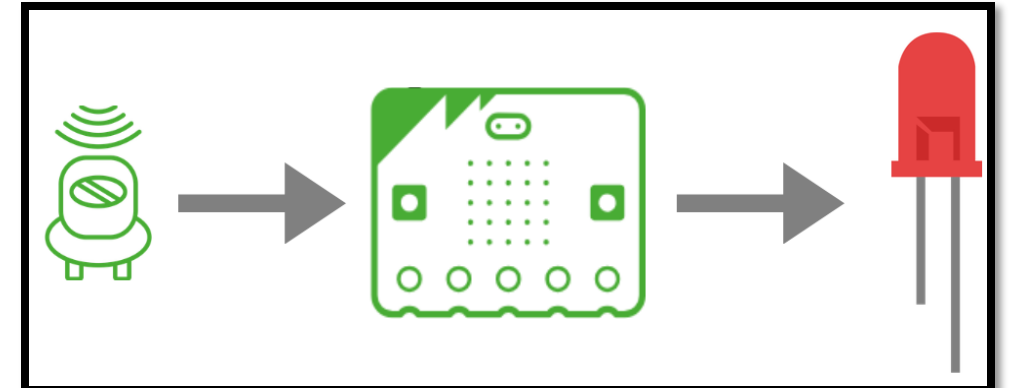
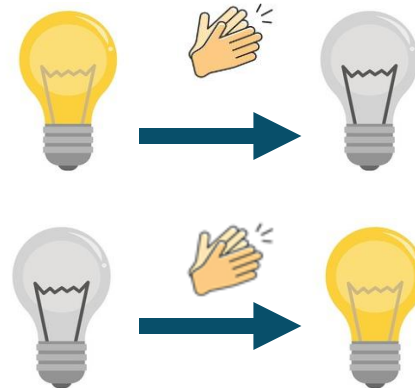
Programa 7: Lámpara con palmada. Explicación.



Usaremos una puerta NOT para conseguir encender/apagar los LEDs dando una palmada. **Utilizamos el evento al detectar el sonido alto para negar la variable**, que iniciamos en cero.

En esta práctica vamos a **crear una variable**. Puedes pensar en la variable con una caja con una etiqueta en la que puedes cambiar lo que hay dentro. Probablemente estés familiarizado con las variables puntos, vidas,... de los videojuegos.

Si iniciamos la variable en cero, al dar una palmada pasará a estar en uno. Con la segunda palmada volverá a cero y así sucesivamente. Luego sólo tenemos que utilizar una estructura condicional igual a las anteriores, haciendo que cuando la variable esté en uno se enciendan los LEDs y cuando este en cero se apaguen.





Programa 7: Lámpara con palmada.



Descripción: programa la Microbit para que el estado de las luces de la matriz LED cambie (de encendido a apagado o de apagado a encendido) cuando la Microbit detecte un aplauso.

Cuando hayas terminado:

- Prueba a añadir un sonido cuando cambie de apagado a encendido.
- Usar la puerta NOT para elaborar nuevos programas.

Bloques que vamos a usar:



Básico



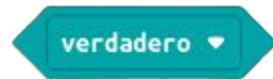
Entrada



Lógica



Variables



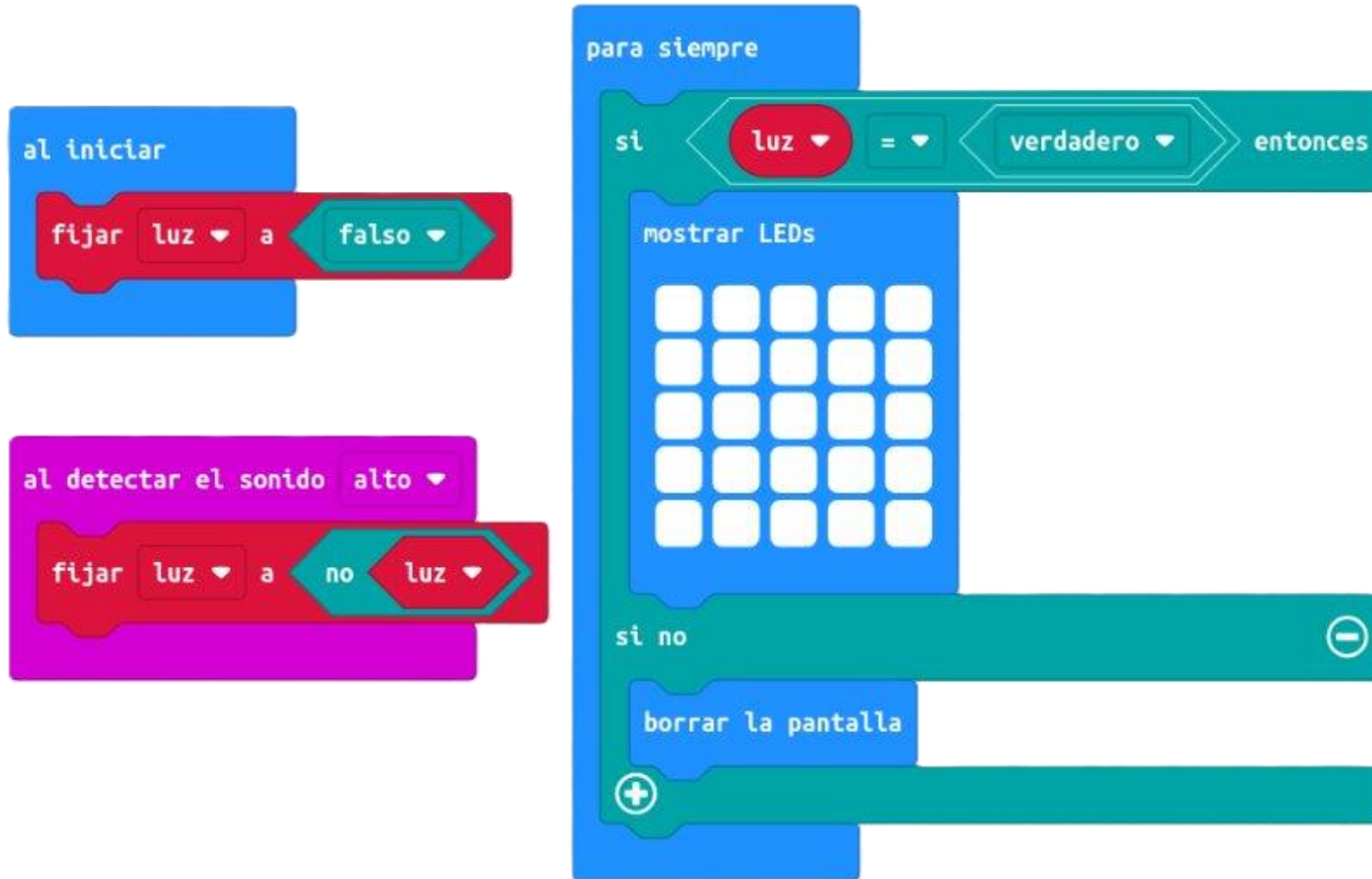
Crear una variable...



Ten en cuenta que hasta que no crees una variable no aparecerán los bloques fijar a, cambiar a,...



Programa 7: Lámpara con palmada. Programación.



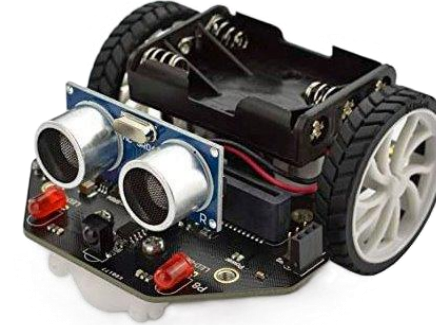
[Enlace al programa](#)



Maqueen

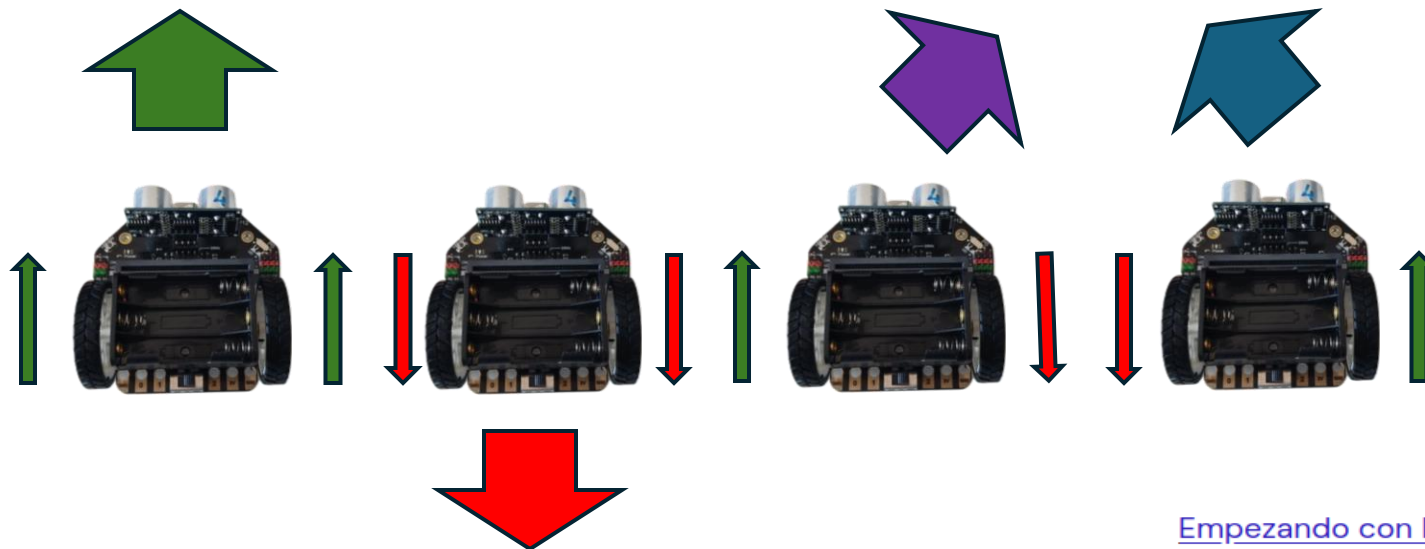


Maqueen es un pequeño coche robot que podemos controlar con Micro:bit. Además de los sensores de la placa tiene un sensor de ultrasonidos con el que podemos medir distancias.



Ahora **nuestros actuadores serán principalmente los motores del robot y utilizaremos la programación para controlar su movimiento.**

El movimiento según el giro de los motores es:



Para que el robot gire haremos que los motores se muevan en direcciones contrarias. Si sólo se mueve un motor le va a costar más hacer el giro.

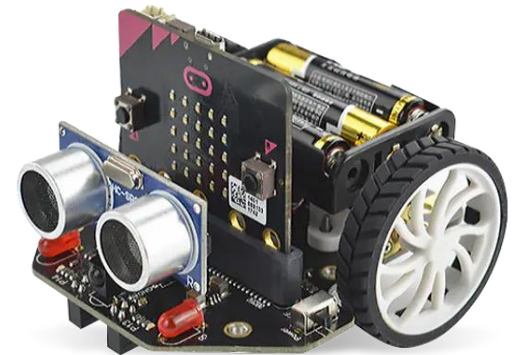
Extensiones



Un **lenguaje de programación** siempre dispondrá de una serie de **funciones básicas** implementadas que el programador puede utilizar.

Por ejemplo, tanto Scratch como MakeCode tienen funciones para sumar, restar, comparar...

Sin embargo, en algunas ocasiones pueden ser necesarias funciones más complejas o específicas, por ejemplo, cuando deseamos programar un robot concreto, como Maqueen, o utilizar un sensor específico. En estos casos deberemos instalar lo que se denomina una extensión.



Al instalar una **extensión** aparecerán una serie de funciones específicas, disponibles en los bloques de programación del menú, que se podrán utilizar en el programa.

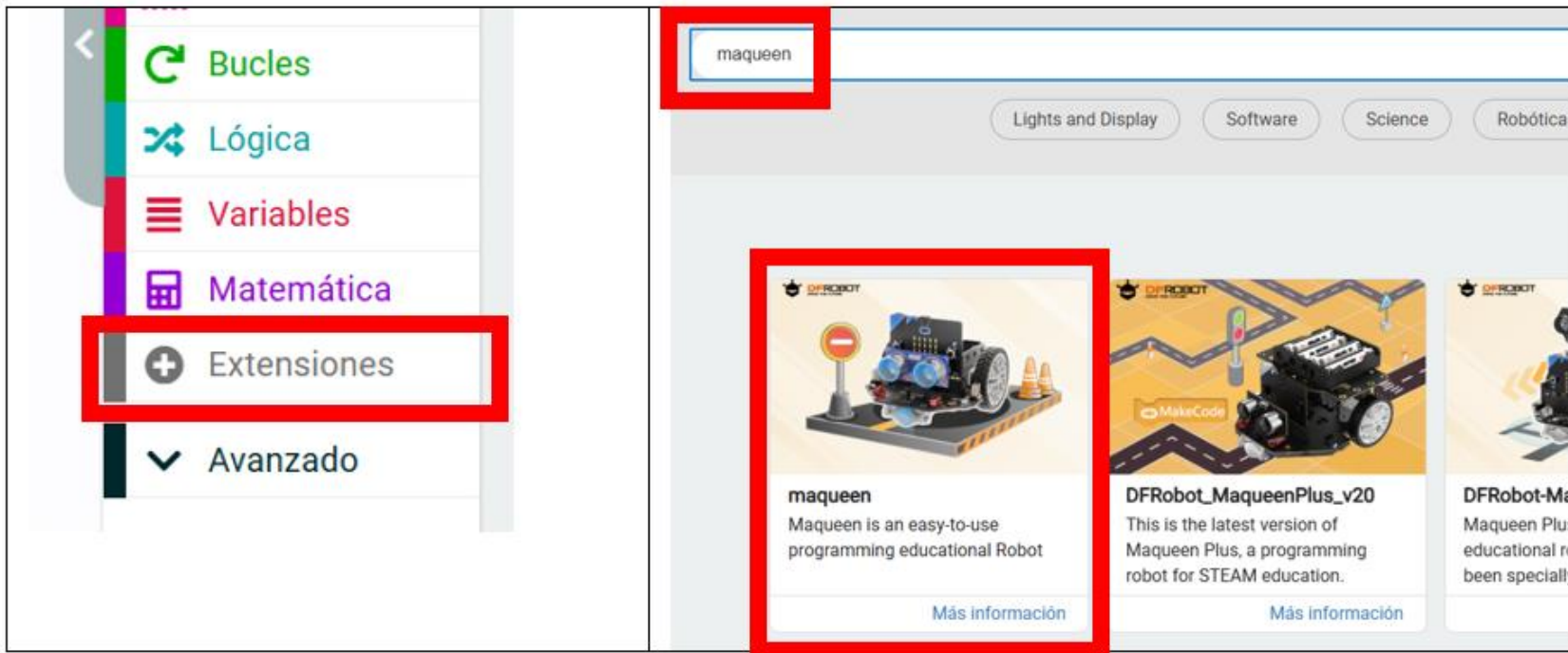
Estas extensiones no se cargan en el programa por defecto ya que lo haría mucho más pesado y poco funcional. Además, puede haber problemas de compatibilidad entre unas extensiones y otras, por lo que es conveniente instalar únicamente las extensiones que se vayan a utilizar.

Extensión de Maqueen



Para instalar la extensión de Maqueen, crea un nuevo proyecto en [Makecode](#) y sigue los siguientes pasos:

- 1) Haz click en "**Extensiones**", en el menú de la izquierda.
- 2) Escribe "**Maqueen**" en la barra del buscador.
- 3) Selecciona la extensión indicada en la imagen.



[Enlace a MakeCode con la extensión de Maqueen cargada](#)

Motores

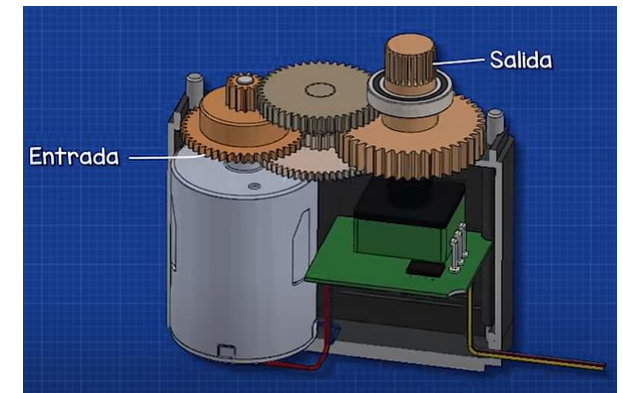
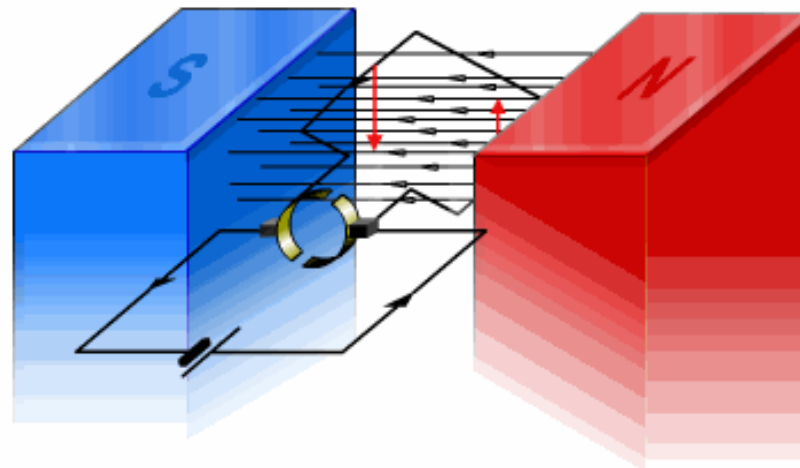
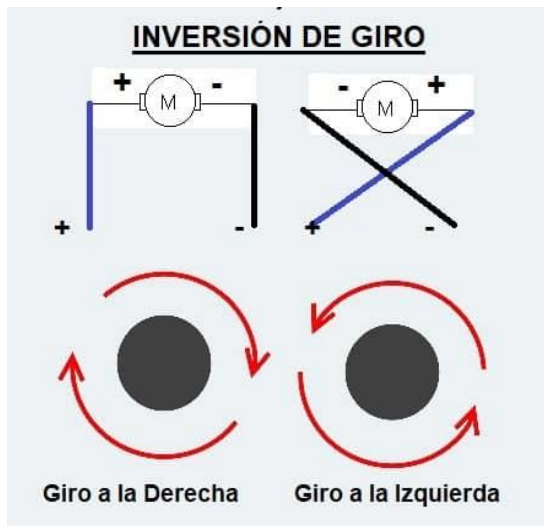


Los motores siempre giran en el sentido de la corriente (de positivo a negativo).

Al hacer circular corriente a través de las bobinas de un motor dentro de un campo magnético se genera sobre las propias bobinas un par de fuerzas que las hacen girar como se puede ver abajo.

El robot Maqueen dispone de **dos motores de rotación continua**, a los que se conectan las ruedas, que se pueden controlar, variando tanto su **velocidad** como su **sentido de giro**.

Además, se pueden conectar otro tipo de motores, conocidos como **servomotores**, que permiten realizar **giros controlados**, indicando los grados a los que se debe desplazar su eje (ver vídeo explicativo).



[Explicación del funcionamiento de un servomotor](#)

Programación de motores



Los motores de **rotación continua** del Maqueen se controlan con el siguiente bloque, en el que podrás seleccionar el **motor** que deseas mover (izquierda/derecha/ambos), el **sentido de giro** (avanzar/retroceder) y la **velocidad** de giro (entre 0 y 255). Para cambiar el sentido de giro lo que hace el programa es invertir la polaridad de los pines.



En caso de conectar algún **servomotor** al robot Maqueen, podrás programarlo con el bloque de abajo. En primer lugar, deberás seleccionar el **puerto** en el que está conectado (S1 o S2) y después indicarás el valor del **ángulo** (entre 0 y 180).



Estos bloques se encuentran en las extensiones Maqueen V4 y V5

 Maqueen v4

 Maqueen v5

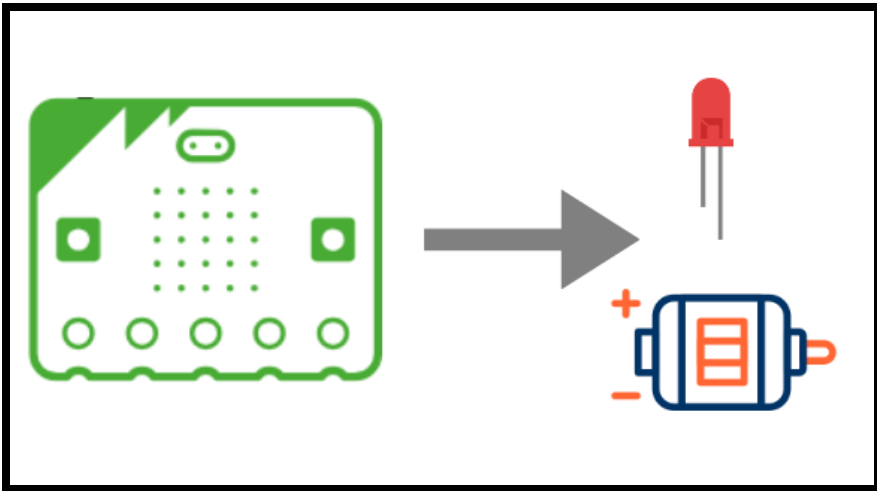


Programa 8: Moviendo a Maqueen. Explicación.

Para empezar a trabajar con Maqueen vamos a realizar un programa secuencial sencillo en el que se mueva y encienda las luces.

Es importante tener cuidado con los tiempos de espera y la velocidad de los motores para controlar bien su movimiento.

Vamos a usar un bucle para evitar tener que repetir los mismos bloques varias veces. **Los bucles nos permiten que se ejecute varias veces la misma secuencia de instrucciones**



En esta práctica no estamos usando sensores y **la programación va directamente a los actuadores.**

Ten en cuenta que **los motores no son precisos** y es fácil que se tuerza un poco, aunque lo programes para que vaya recto. O que no gire igual a la izquierda que a la derecha.....



Programa 8: Moviendo a Maqueen.



Descripción: programa la Microbit para que Maqueen se mueva dibujando un cuadrado u otra figura geométrica de tu elección. Cada LED debe iluminarse según el giro de su motor: cuando vaya hacia delante apagado y hacia atrás encendido.

Cuando hayas terminado:

- Programa a Maqueen para que haga otros movimientos.

Bloques que vamos a usar:



Básico



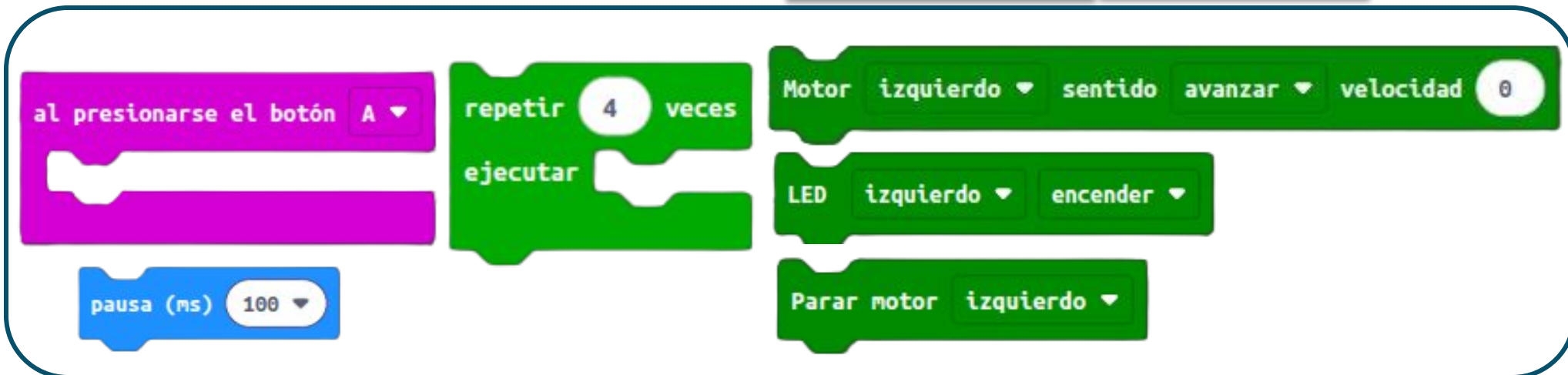
Entrada



Maqueen v4

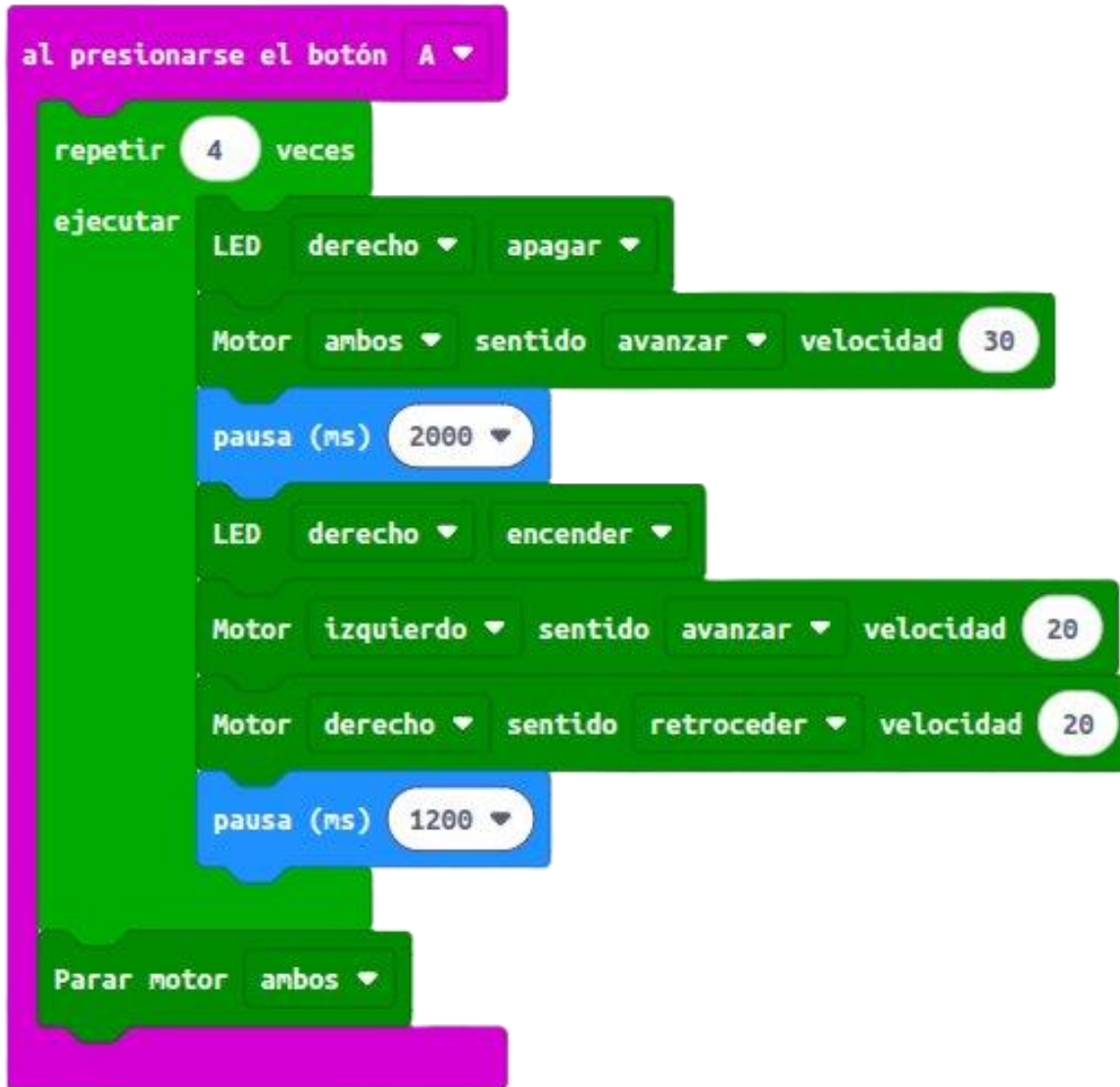


Bucles





Programa 8: Moviendo a Maqueen. Programación.



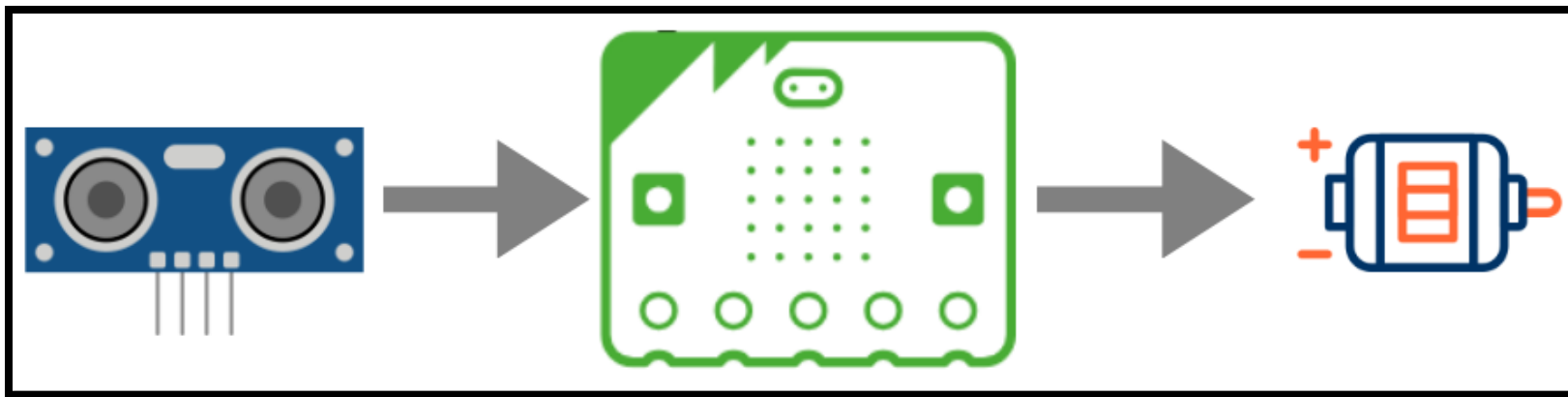
[Enlace al programa](#)



Programa 9: Esquiva obstáculos. Explicación.

Una vez aprendido el funcionamiento básico del robot vamos a hacer un programa **condicional** que impida que el robot se choque frontalmente.

Utilizaremos el sensor de ultrasonidos para ver a qué distancia se encuentra el objeto más sencillo. Dentro de la categoría de Maqueen hay un bloque específico que nos permite hacerlo con facilidad.



El sensor de ultrasonidos manda la distancia a la placa. Nuestro programa hará que cambie el giro de los motores cuando la distancia detectada sea pequeña, evitando así que se choque.



Programa 9: Esquiva obstáculos.



Descripción: cuando el sensor de ultrasonidos detecte un objeto a menos de 5 cm debe ir hacia atrás durante un corto espacio de tiempo y girar durante un tiempo aleatorio para evitar el choque.

Cuando hayas terminado:

- Programa a Maqueen para que haga otros movimientos cuando detecte un objeto.

Bloques que vamos a usar:



Básico



Lógica



Maqueen v4

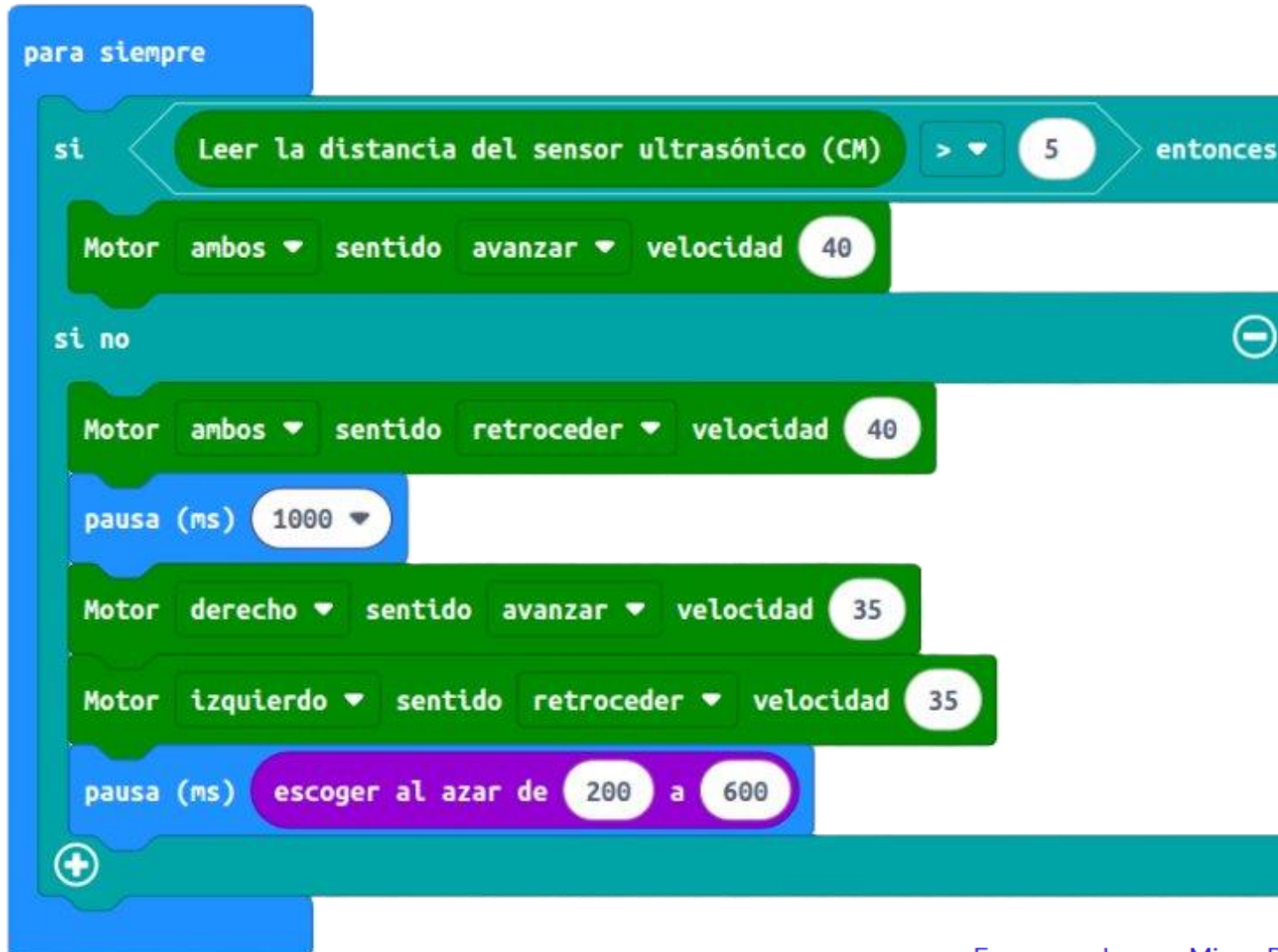


Matemática





Programa 9: Esquiva obstaculos. Programación.



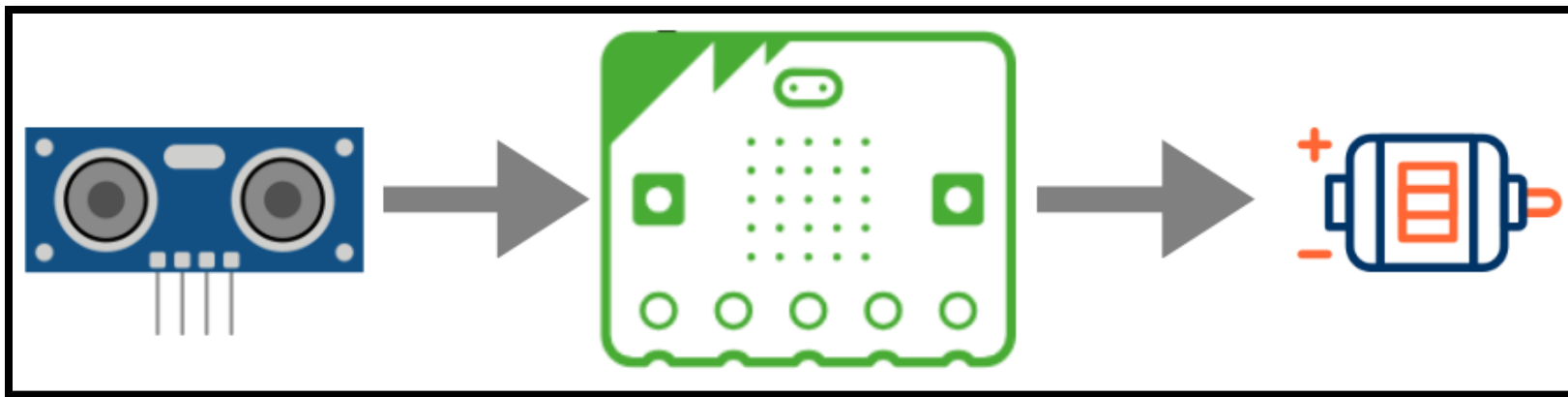
[Enlace al programa](#)



Programa 10: Sigue manos. Explicación.

Con unos pequeños cambios podemos pasar de un robot que esquive obstáculos a uno que siga nuestra mano.

Utilizaremos un programa **condicional**: cuando el sensor de ultrasonidos detecte un objeto cercano se moverá en línea recta y si no se quedará girando sobre sí mismo.



El sensor de ultrasonidos manda la distancia a la placa. Nuestro programa hará que cambie el giro de los motores cuando la distancia detectada sea pequeña.



Programa 10: Sigue manos.



Descripción: cuando el sensor de ultrasonidos detecte un objeto a menos de 8 cm avanzará en línea recta. De lo contrario girará sobre sí mismo.

Cuando hayas terminado:

- Programa a Maqueen para que haga otros movimientos cuando no detecte un objeto.

Bloques que vamos a usar:



Básico



Lógica



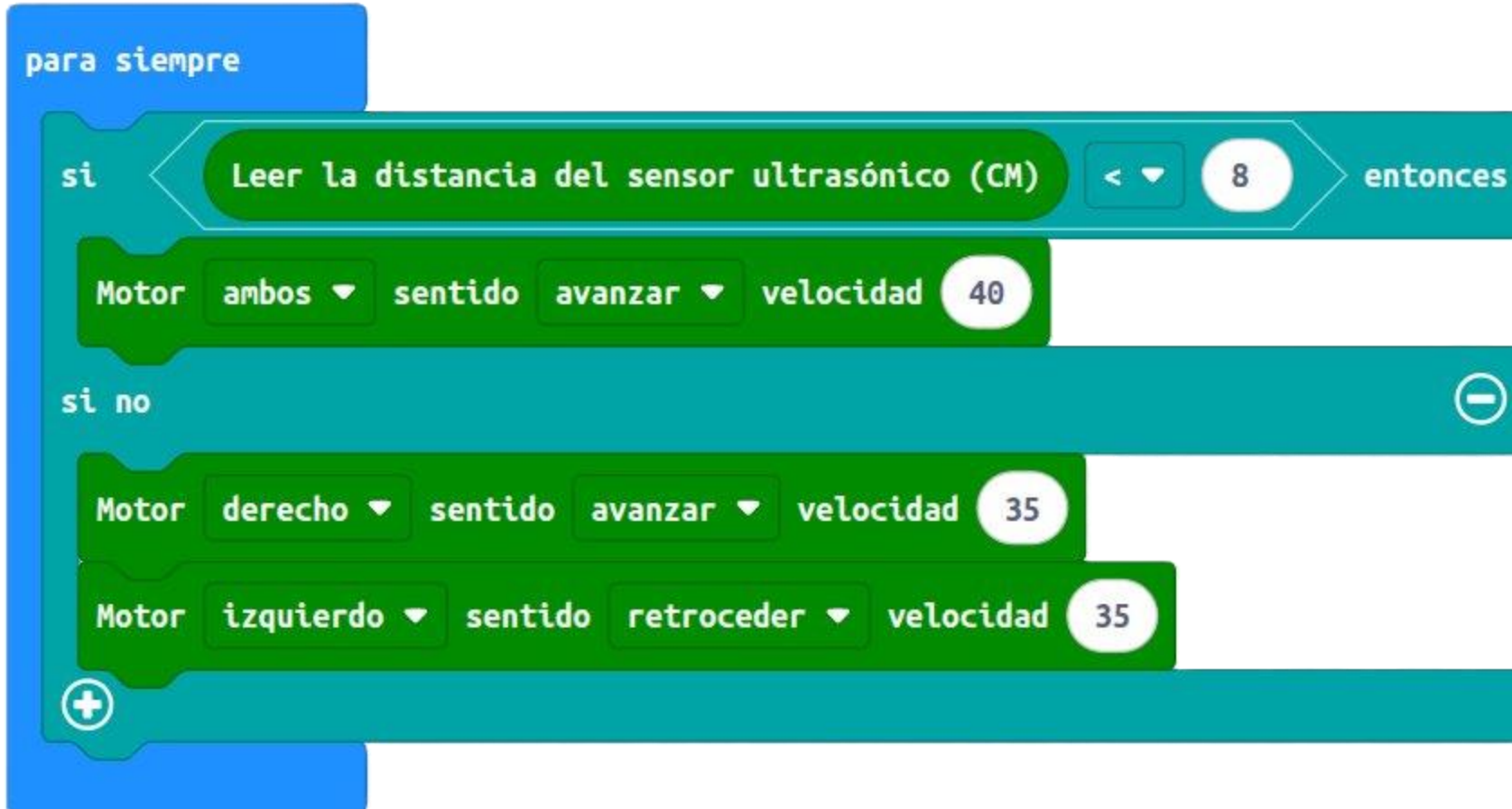
Maqueen v4



Leer la distancia del sensor ultrasónico (CM)



Programa 9: Sigue manos. Programación.



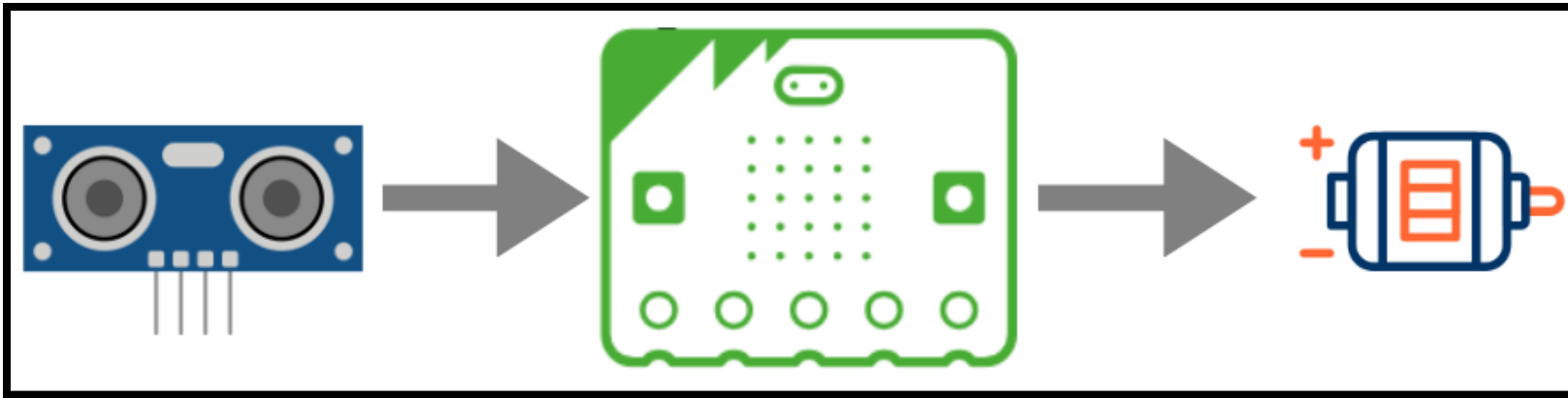
[Enlace al programa](#)



Programa 11: Control con manos. Explicación.

Con unos pequeños cambios podemos modificar el programa para que cambie de girar a moverse en línea recta cada vez que detecte un objeto cercano.

Utilizaremos un programa **condicional**: cuando el sensor de ultrasonidos detecte un objeto cercano negará una variable y el movimiento del robot dependerá de si la variable está en 1 o en 0. Es muy parecido al programa lámpara con palmada. Ten en cuenta que debes tener un tiempo de espera para que la variable no cambie de valor varias veces seguidas cuando detecte tu mano.



El sensor de ultrasonidos manda la distancia a la placa. Nuestro programa hará que cambie el giro de los motores según el estado de una variable que depende de la distancia detectada por el ultrasonidos.



Programa 10: Control manos.



Descripción: cuando el sensor de ultrasonidos detecte un objeto a menos de 8 cm ne/False de la variable el robot girará o se moverá en línea recta. Cada vez que cambie de estado la variable deja un pequeño tiempo de espera.

Cuando hayas terminado:

- Cambia velocidades y tiempos de espera hasta que el robot responda de manera adecuada.

Bloques que vamos a usar:



Básico



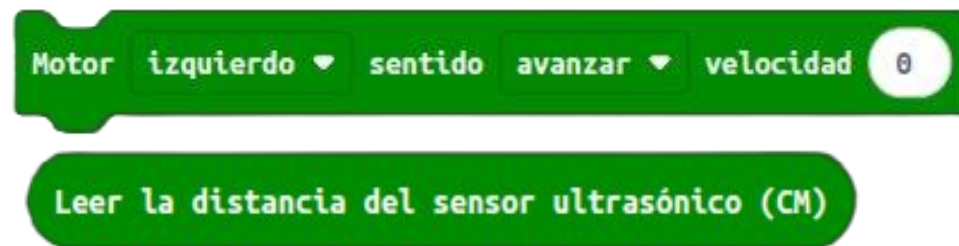
Lógica



Maqueen v4



Variables

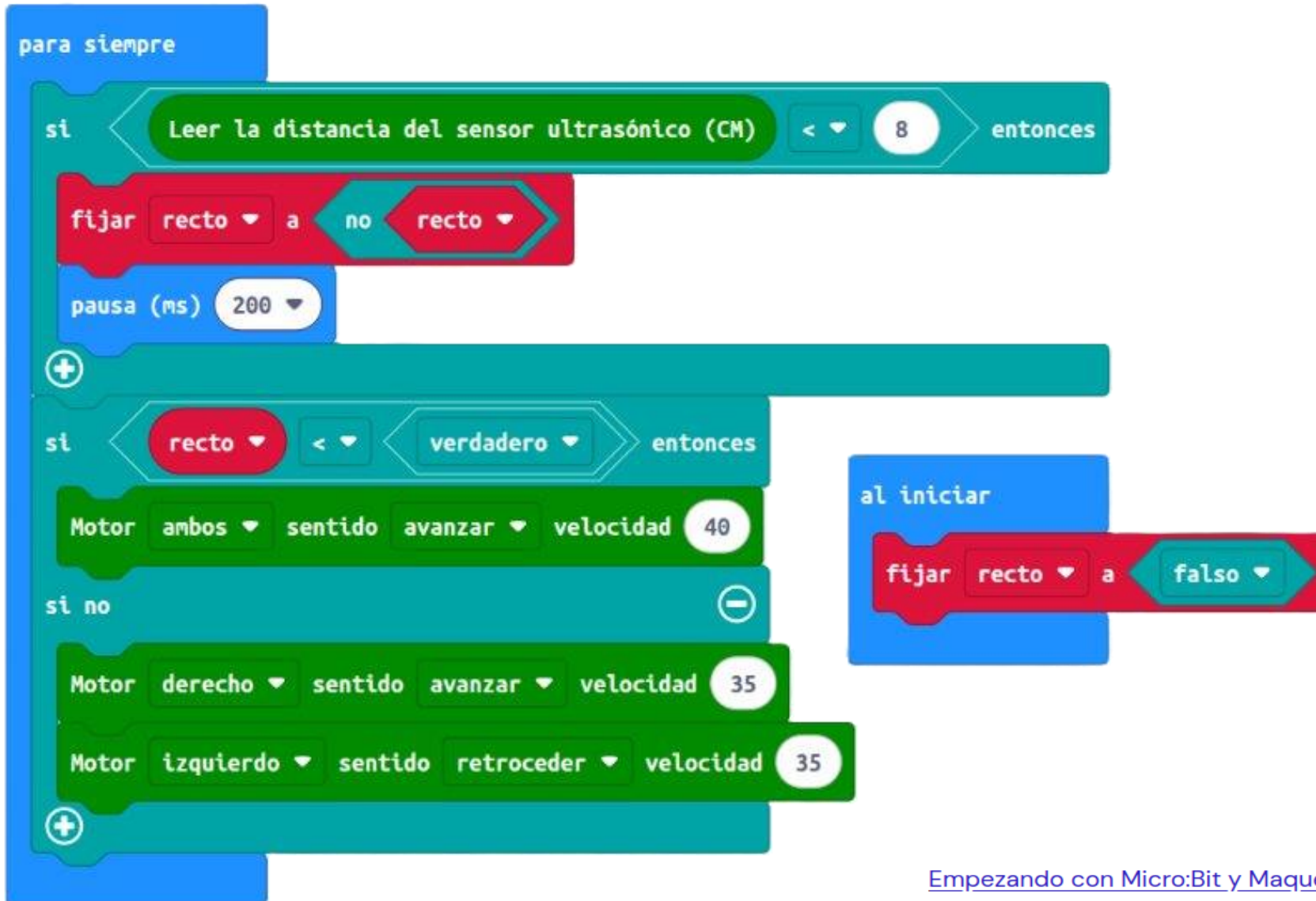


Crear una variable...





Programa 9: Control manos. Programación.



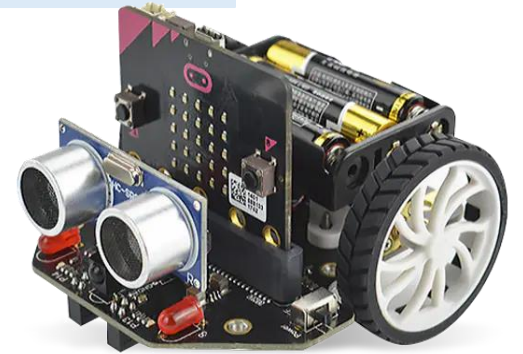
[Enlace al programa](#)

Otros programas



La mayoría de los programas que hemos hecho con Maqueen utilizan el sensor de ultrasonidos. Aquí proponemos otros programas que se pueden realizar utilizando otros sensores.

- **Robot sigue luz:** el robot sólo va en línea recta cuando detecte una luz alta, sino se queda girando.
- **Robot anti oscuridad:** cuando la luz sea muy baja el robot va hacia atrás y gira de manera aleatoria.
- **Robot movimiento con palmada:** Al detectar un ruido alto el robot cambia su movimiento.
- **Robot esquiva obstáculos con palmada:** añade al robot esquiva obstáculos que se pare/encienda al detectar un ruido alto.



¡¡Y por supuesto cualquier otra combinación que se te ocurra utilizando varios sensores!!



¡Muchas gracias!

