
SMART CUTE- BOT

Abacus es una organización de referencia en la educación, la cultura y el ocio que crea y distribuye contenidos y productos de calidad para contribuir a un país y un mundo mejores desde los valores del cooperativismo.

Tanto Habilis, la plataforma de creación de contenidos educativos, como RO-BOTICA, un referente en la distribución de soluciones tecnológicas, se engloban dentro de Abacus Educación.

HABILIS

En Habilis nos dedicamos a la creación de contenido educativo innovador, a la formación docente y al acompañamiento de las escuelas en la transformación digital, con más de 80 asesores pedagógicos.

RO-BOTICA

RO-BOTICA abrió en 2007 el primer espacio de Europa dedicado a robótica educativa. Desde entonces, distribuye e implementa soluciones tecnológicas de robótica y programación en centros de todas las etapas educativas. Contribuye al desarrollo competencial digital y tecnológico de la sociedad, facilitando e impulsando vocaciones científico-tecnológicas, la alfabetización STEAM, la educación inclusiva, equitativa y de calidad.

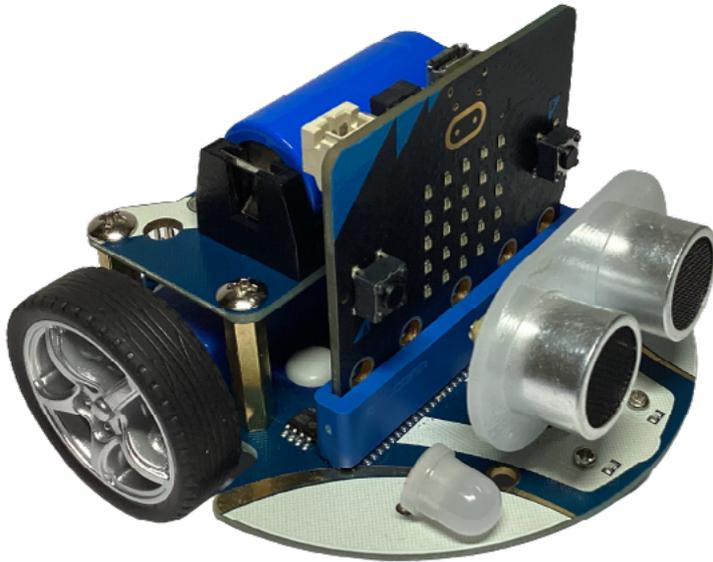
Consulta nuestro catálogo:



ÍNDICE

1.	Descripción	3
2.	¿Qué hay en la caja?	4
2.1.	¿Cuáles son los componentes del robot?	5
2.2.	¿Cuál es la función de cada componente?	6
3.	Prepara el robot	7
4.	Programa desde el ordenador	8
4.1.	Conoce el entorno de programación	8
4.2.	Conecta la placa	11
4.3.	Comienza a programar	11
4.4.	Guarda y comparte programas	18
5.	Programa desde una tableta	19
5.1.	Conoce el entorno de programación	19
5.2.	Conecta la placa	20
5.3.	Conoce el entorno de programación	22
5.4.	Transfiere el programa a la placa	22
6.	¿Te atreves con los retos?	23
6.1.	Posibles soluciones	30
7.	consejos de seguridad	33
8.	Anexo	34

DESCRIPCIÓN



Smart Cutebot es un robot que tiene un diseño de vehículo y que se controla mediante la placa micro:bit.

Incorpora una placa de extensión (shield) que permite conectar de forma sencilla la placa micro:bit y una serie de componentes, algunos de ellos ya soldados a la propia base: led RGB, sensores de infrarrojos, zumbador y sensor de distancia por ultrasonido.

La base del robot también incorpora puertos de tres pines que posibilitan incluir otros sensores y actuadores.

Como funciona con la placa micro:bit, el sistema de programación que utiliza es Microsoft MakeCode. Solo hay que añadirle la extensión de bloques de programación correspondiente.

INFORMACIÓN

ETAPA EDUCATIVA

3^{er} ciclo 1^o y 2^o
EP ESO

CONECTIVIDAD



ENTORNO DE PROGRAMACIÓN

Oficiales:

Microsoft
MakeCode



Micro:bit Python



DISPOSITIVOS



SISTEMAS OPERATIVOS

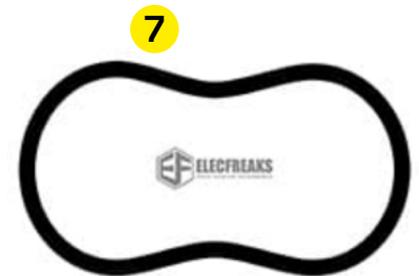
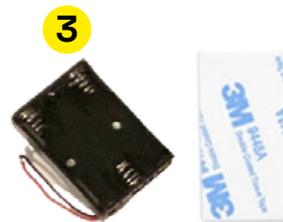
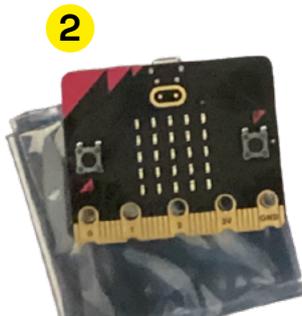
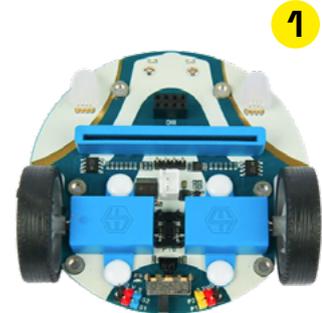


linkatedu®

2

SMART CUTEBOT

¿QUÉ HAY EN LA CAJA?



1. Smart Car Cutebot
2. Placa BBC micro:bit
3. Soporte para 3 pilas AAA y adhesivo
4. Sensor de distancia por ultrasonido
5. Cable micro-USB
6. Guía de uso
7. Circuito

Paquete batería recargable:



Soporte de la batería, batería y cable de conexión.



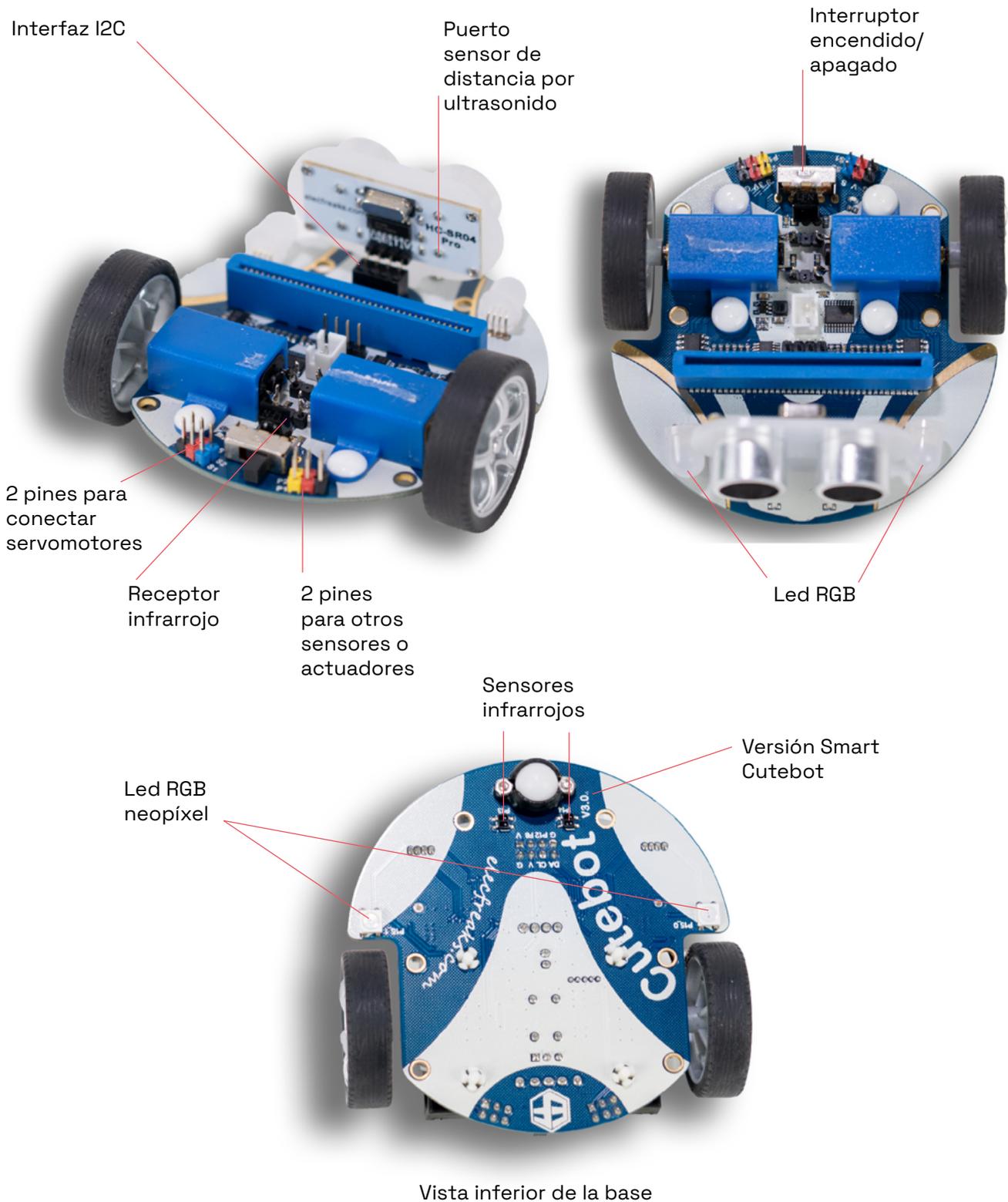
Piezas de plástico para montar soportes.



Destornillador, espaciadores metálicos, tornillos y tuercas.

2.1. SMART CUTEBOT

¿CUÁLES SON LOS COMPONENTES DEL ROBOT?



¿CUÁL ES LA FUNCIÓN DE CADA COMPONENTE?

Los sensores y los actuadores tienen las siguientes funcionalidades:

MOTORES

El robot se puede mover hacia delante y hacia atrás, y girar a la izquierda y a la derecha. Los motores se pueden programar para controlar la velocidad de giro de las ruedas.

SENSOR DE ULTRASONIDOS

Detecta y mide la distancia de los objetos que tiene delante del sensor. Por ejemplo, el robot puede detectar obstáculos y se puede programar el movimiento de los motores para que se pare o reaccione.

RECEPTOR INFRARROJO

Con el receptor infrarrojo, el robot puede ser controlado por un mando a distancia.

INTERRUPTOR DE ENCENDIDO Y APAGADO

Para encender y apagar la placa de extensión.

LED RGB

Los ledes emiten luces de distintos colores.

SENSORES INFRARROJOS

Los sensores infrarrojos permiten seguir un recorrido trazado en negro sobre una superficie blanca.

PUERTO DE EXPANSIÓN IO

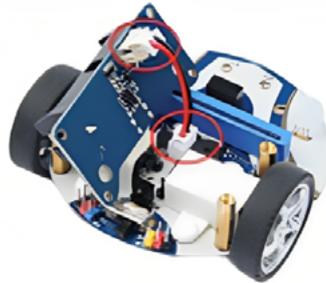
Posibilita, además, conectar otros sensores y actuadores. IO provee de *input* y *output*: señales de entrada y de salida.

PREPARA EL ROBOT

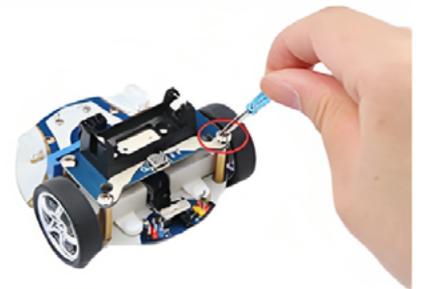
- 1** Monta el soporte de la batería sobre la base de Cutebot:



A) Atornilla los separadores metálicos a la base con las hembras.

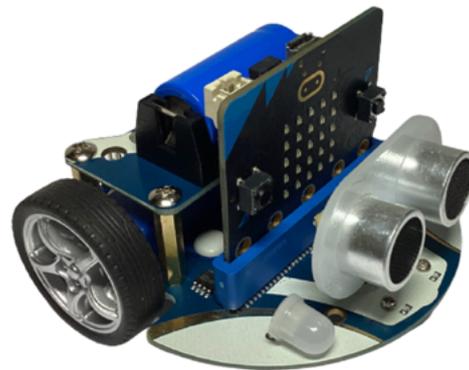
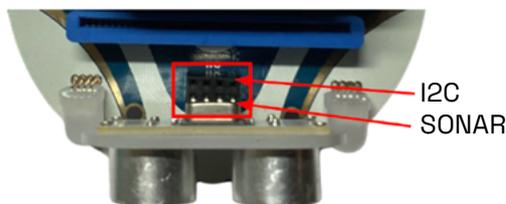


B) Conecta las clavijas del cable al soporte y a la base.

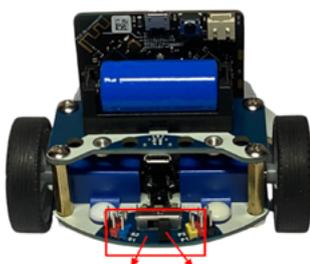


C) Utiliza el destornillador y los tornillos para fijar el soporte.

- 2** Conecta el sensor de ultrasonido al puerto SONAR, inserta la batería recargable y la placa micro:bit a la ranura del Cutebot.

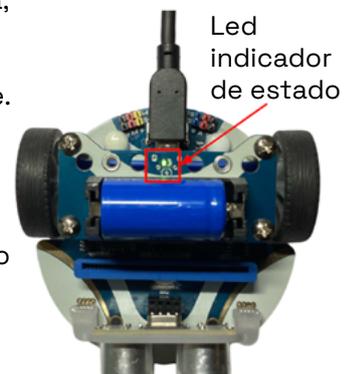


- 3** Para encender Cutebot, utiliza el interruptor que encontrarás en la parte trasera.



Encendido Apagado

- 4** Para cargar la batería, conecta un cable micro-USB al puerto situado en el soporte. Cuando la batería esté cargándose, se iluminará un led indicando el estado de carga: rojo cuando se esté cargando y verde al finalizar la carga completa.



PROGRAMA DESDE EL ORDENADOR

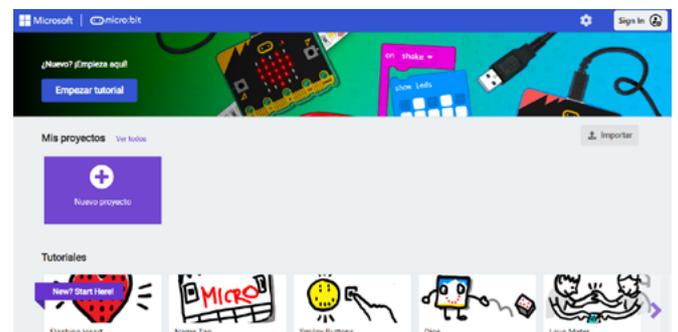
4.1. CONOCE EL ENTORNO DE PROGRAMACIÓN

1 Antes de comenzar a programar Smart Cutebot, necesitas disponer de una placa micro:bit, un cable micro-USB, el robot montado con la batería recargable y un dispositivo para programar (un ordenador).

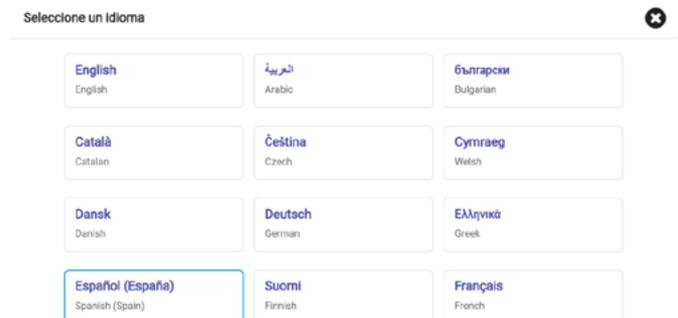


2 Una vez tengas todo el material necesario, accede al entorno de programación en línea Microsoft MakeCode a través del enlace siguiente: <https://makecode.microbit.org/>.

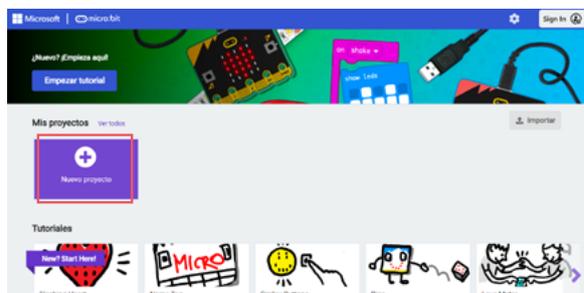
Se recomienda abrir el entorno de programación desde el navegador Google Chrome, Chromium o Microsoft Edge para que el proceso de descarga de la programación sea más sencillo.



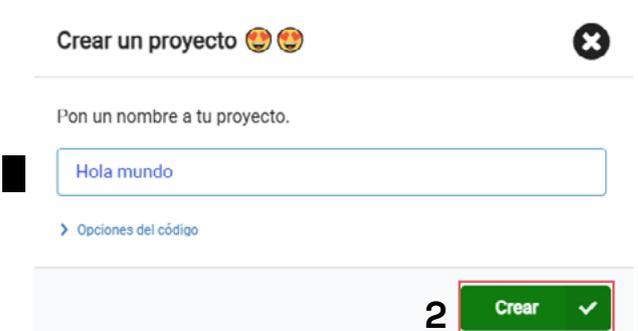
3 Selecciona el idioma del entorno de programación desde el icono del engranaje situado en la parte derecha del menú superior (disponible en castellano) .



4 Haz clic en “Nuevo proyecto” para abrir el editor de programación.

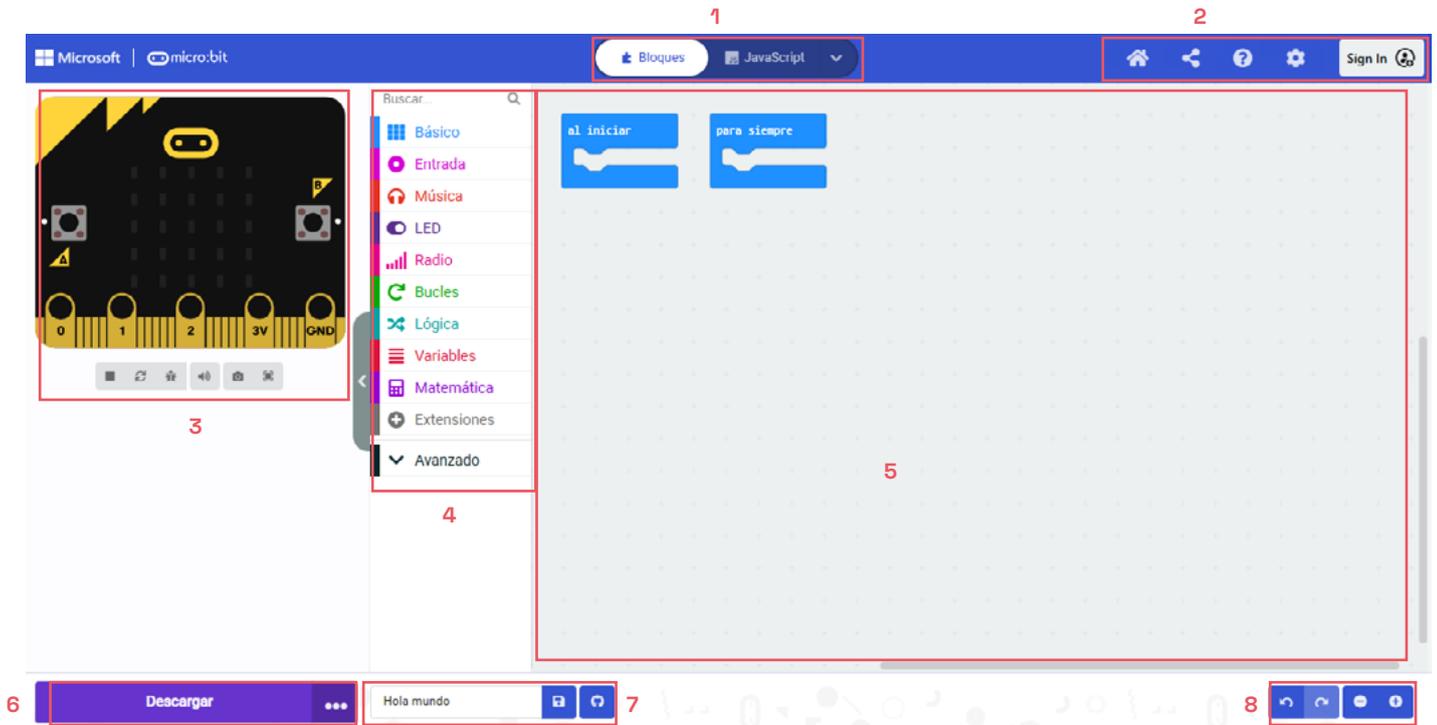


Tras ello, nombra el proyecto **(1)** y haz clic en el botón “Crear” **(2)**.



4.1. SMART CUTEBOT

Automáticamente, se abrirá la aplicación para programar la placa micro:bit.



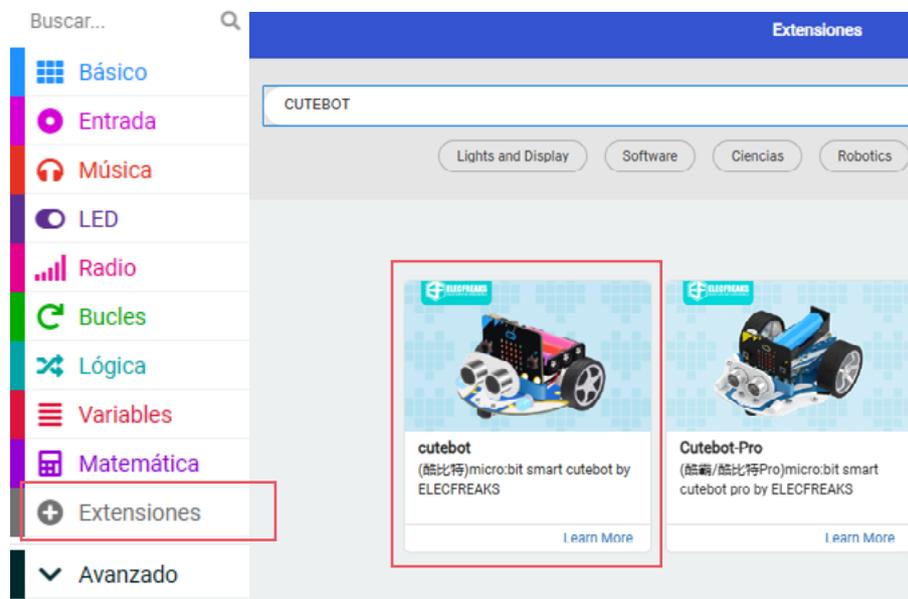
En la parte superior del entorno encontraremos un primer menú. En la parte central hay un menú deslizante (1) que permite escoger cómo quieres programar tu placa, con JavaScript o con Python. En la parte de la derecha del menú encontraremos diferentes opciones (2):

1. Volver al menú principal del entorno de programación.
2. Para compartir tu programa.
3. Opción de ayuda de los programas.
4. Configuración del proyecto, extensiones, conectar dispositivos, eliminar el proyecto, cambiar el idioma y cambiar el contraste, entre otros.
5. Iniciar la sesión. Puedes crearte un usuario para tener siempre tus programas guardados en tu cuenta de MakeCode.

El entorno también dispone de un simulador de la placa programable (3), un menú central donde encontrar todos los bloques de programación (4) y, a la derecha, una zona de programación donde arrastrar y encajar los bloques (6).

Para acabar, encontramos un menú inferior con más opciones. A la izquierda se encuentra el botón de descargar (6) los programas a la placa. En la parte central, un espacio para nombrar y guardar (7) un programa, y diferentes botones (8) para “deshacer” y “rehacer” y “disminuir” o “aumentar” el zoom.

En el menú central de bloques de programación encontraremos de color gris la categoría “Extensiones”. Clica en ella. Aquí encontrarás todas las distintas extensiones de MakeCode. En este caso, para programar el robot Smart, escribe en el buscador “Cutebot”.



Después de pulsar en la extensión que quieras, se te abrirán dos categorías de programación más en el menú central de bloques de programación: CuteBot y Neopixel.



A continuación, te explicamos en qué consiste cada grupo de bloques. Mira el apartado 8 de Anexo de este documento para ver los bloques de programación de cada categoría:

- **CuteBot (verde):** bloques de programación para Smart Cutebot. Encontraremos todos los bloques para mover el robot, encender o apagar los ledes y programar el IR, entre otros.
- **Neopixel (azul):** bloques de programación para hacer funcionar los ledes RGB llamados neopixel que están en la parte inferior de la base del Smart Cutebot.

4.2. SMART CUTEBOT

CONECTA LA PLACA

La micro:bit es la placa controladora del robot Cutebot. Para programar la micro:bit desde el ordenador, utiliza el cable micro-USB para conectar la placa micro:bit al ordenador.

Puedes apagar el Smart Cutebot con el interruptor de la parte posterior o quitar la micro:bit cuando se le transfiera el programa. Recuerda encender el Smart Cutebot cuando esté listo.

Al hacerlo, automáticamente se abrirá una ventana de archivos, como si fuera una unidad de memoria externa USB llamada "MICROBIT".

A partir de ahora, ya puedes comenzar a programar el Smart Cutebot. ¡Ya tienes el robot y el entorno de programación a punto!

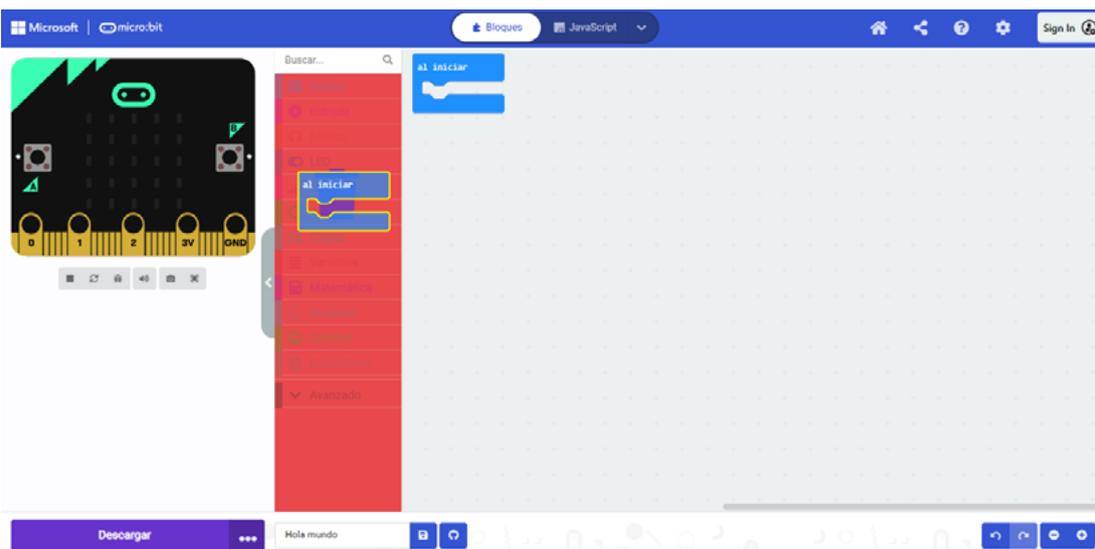


4.3. SMART CUTEBOT

COMIENZA A PROGRAMAR

Antes de empezar a programar, recuerda estas indicaciones:

- Para añadir los bloques a la zona de programación, clicas encima de ellos y arrástralos uno a uno hasta la zona de programación.
- Para juntar los bloques entre ellos, es importante que, cuando los arrastres, los coloques uno bajo el otro.
- Para borrar bloques de programación que no necesites, arrástralos al menú central de los bloques de programación.



- Puedes ampliar o reducir el zoom de la zona de programación clicando sobre los botones "-" o "+" que encontrarás en la parte derecha inferior del entorno de programación.
- Asegúrate de que estás programando con bloques. Clicas sobre "Bloques" que encontrarás en la parte central superior del entorno.

4.3.

SMART CUTEBOT

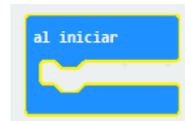
¡Movimiento y luces!

Para hacer este programa, los principales bloques de programación que utilizarás para programar el Cutebot son: “Básico”, “Entrada”, “Bucles”, “Lógica”, “CuteBot” y “Neopixel”.

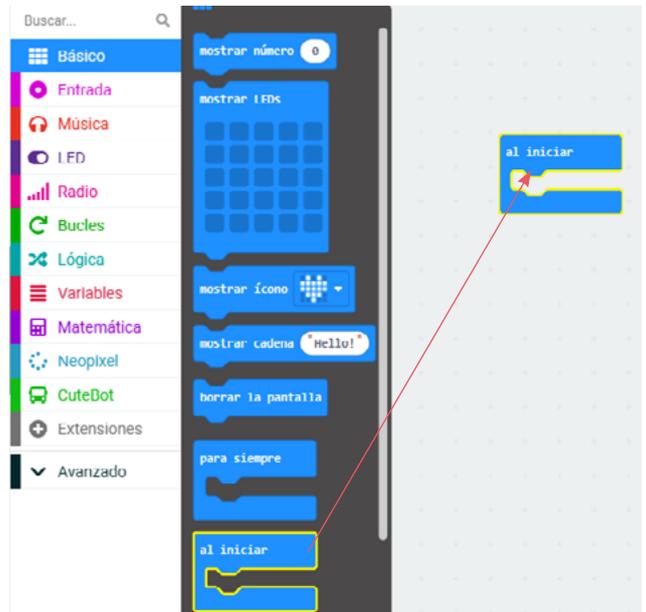
Comienza por crear un programa sencillo. Haz que el robot se mueva en línea recta hacia delante durante 5 segundos, que se ilumine el led derecho de un color y el izquierdo de otro, y que vuelva hacia atrás, a la posición inicial.

Para hacerlo, sigue los siguientes pasos:

- 1 En caso de no tenerlo ya en la zona de programación, ve al grupo azul “Básico” y busca el bloque “al iniciar” . Arrastra este bloque a la zona de programación.



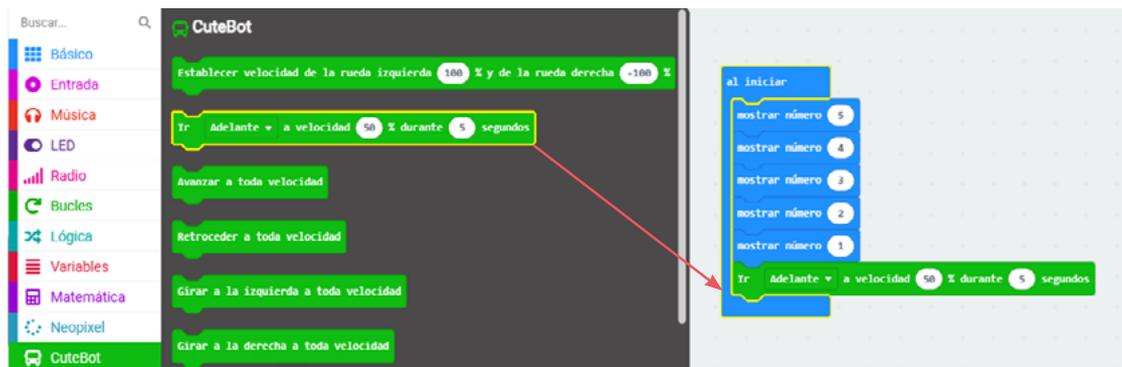
- 2 Antes de darle la orden de avanzar, haremos una cuenta atrás (5, 4, 3, 2, 1). De este modo, una vez descarguemos el programa, tendremos unos segundos de margen para colocar el robot en el suelo antes de que se mueva hacia delante. Para llevar esto a cabo, dentro de la categoría Básico selecciona y arrastra cinco veces a la zona de programación el bloque “mostrar número”.



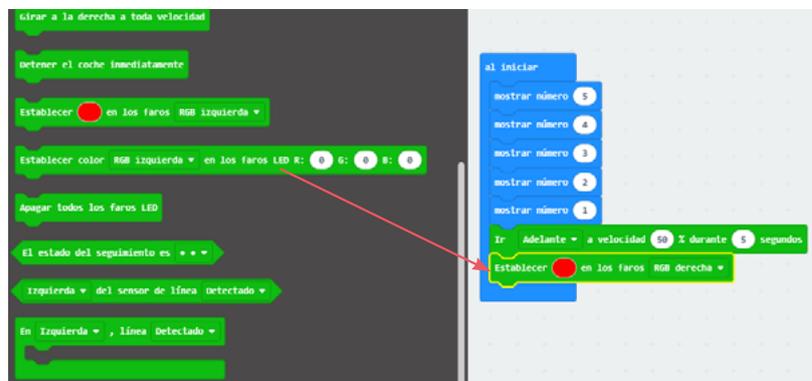
- 3 Colócalos uno tras otro y cambia el número “0” por otros números.



- 4 Ahora es importante que programes lo que quieras que haga el robot una vez comience el programa. Primero, que avance durante 5 segundos. Para hacerlo, como se trata de una acción que quieres que desarrolle el robot, deberás ir al grupo de programación verde “CuteBot”. En este grupo encontrarás el bloque “Ir -adelante- a velocidad 50% durante 5 segundos” Ir Adelante a velocidad 50 % durante 5 segundos. Clicla y arrástralo a la zona de programación. Deberás soltarlo dentro del bloque “al iniciar”. Este bloque incluye las tres consignas solicitadas: mover el Smart Cutebot, moverlo en dirección hacia delante y moverlo durante 5 segundos. En cuanto a la velocidad, se puede ir jugando con el porcentaje, según si quieres que vaya más o menos rápido.



- 5 El siguiente paso es encender el led derecho de un color (cualquiera). Ve a la categoría “CuteBot” y selecciona el bloque “Establecer color -RGB derecha- en los faros led” set LED headlights RGB direcc color. Colócalo bajo el bloque anterior.



- 6 Cambia el bloque de color, de rojo a azul, haciendo clic sobre el icono de color.



4.3. SMART CUTEBOT

7 Arrastra de nuevo el bloque “Establecer color -RGB derecha- en los faros led” y despliega el campo “RGB derecha” clicando en la flecha blanca y selecciona “RGB izquierda”.

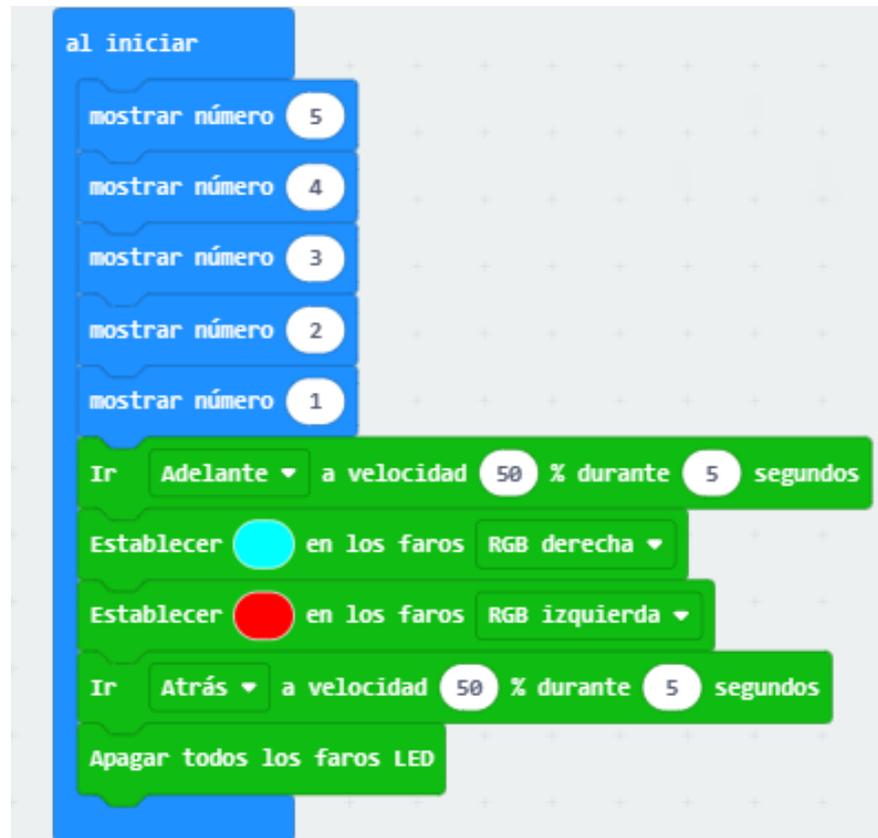


8 Ahora haz que el Cutebot vuelva a la posición de origen, retrocediendo. Arrastra otra vez el bloque “Ir - adelante - a velocidad 50% durante 5 segundos” y despliega el campo “Adelante” clicando en la flecha blanca. Selecciona “Atrás”.



9 Para acabar el programa, apaga todos los ledes arrastrando y encajando el bloque “apagar todos los faro led” de la categoría “CuteBot”.

10 Así quedaría el programa:



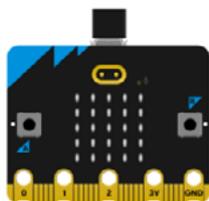
11 Finalmente, descarga el programa pulsando el botón "Descargar" situado en la parte inferior izquierda del editor.

Hay dos formas de descargar el programa en la placa micro:bit, dependiendo del navegador que utilices:

A Si trabajas desde el navegador Chrome, Chromium o Edge:

Estos navegadores te permiten vincular la placa para que el proceso de descarga sea más rápido. Con la placa conectada al ordenador, pulsa en botón “Descargar”  que encontrarás en la parte izquierda inferior del entorno de programación. Se abrirá una ventana para mostrarte cómo vincular la placa micro:bit al ordenador. Sigue los pasos del tutorial:

1. Conecta tu micro:bit al ordenador.



 Siguiente

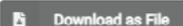
2. Apareja tu micro:bit a tu navegador 

Press the Pair button below.

A window will appear in the top of your browser.

Select the micro:bit device and click Connect.



 Download as File

 Pair

Si te fijas en el botón de “Descargar”, verás que aparece un nuevo icono (el logotipo de micro:bit). Eso quiere decir que está correctamente vinculada.

 Descargar 

Ahora sí, haz clic en el botón “Descargar” para descargar el programa directamente a la placa. Verás que el led amarillo de su parte posterior emite una luz intermitente mientras se está transfiriendo el programa. Una vez cargado en la placa, se ejecutará automáticamente.

B Si utilizas otro navegador, la opción de vincular la placa desaparece. Sigue estos pasos para descargar el programa:

Con la placa conectada al ordenador, pulsa el botón “Descargar” . Verás que se descarga un archivo en formato .hex. Arrastra este fichero a la unidad de memoria externa USB llamada “MICROBIT”.



Verás que el led amarillo de la parte posterior de la placa emite una luz intermitente mientras se está transfiriendo el programa. Una vez cargado este en la placa, se ejecutará automáticamente.

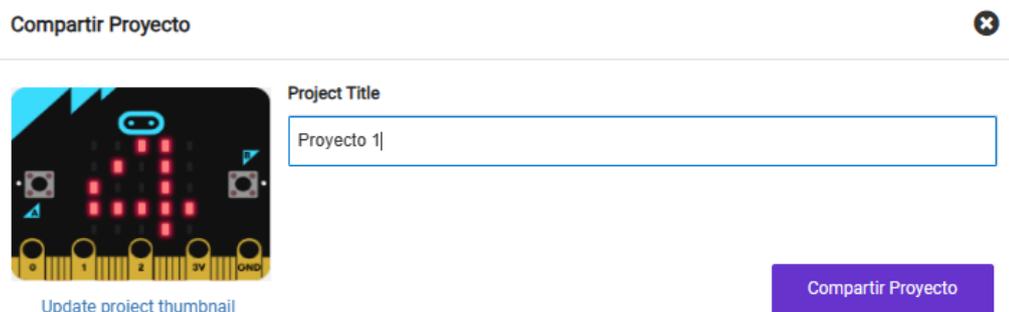
Importante: si trabajas con un cable micro-USB, te aconsejamos que, antes de transferir el programa a la placa micro:bit, la desenchufes del Smart Cutebot. Transfiere el programa y, con el robot apagado, vuélvela a enchufar al Smart Cutebot. Cuando acabes, coloca el robot en el suelo y enciéndelo. De esta manera, aunque hayamos programado unos segundos de espera antes de que el robot se mueva, nos aseguramos de tener tiempo para desconectar el cable micro-USB o para colocar el robot en el suelo antes de que empiece a moverse.



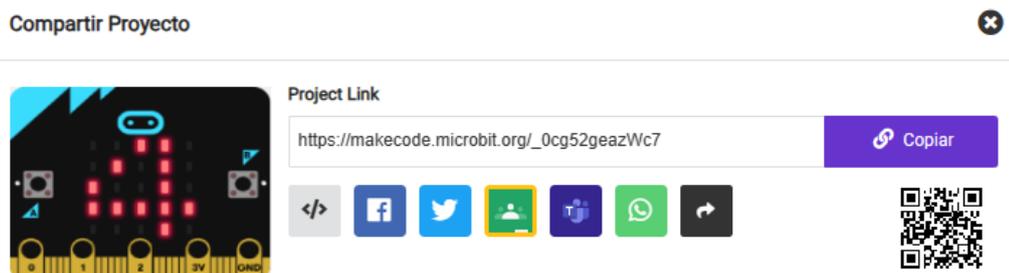
GUARDA Y COMPARTE PROGRAMAS

Puedes guardar y exportar tus programas al ordenador pulsando el icono de “guardar”,  situado en el menú inferior del entorno de programación. También puedes compartir los programas pulsando el icono “compartir” , situado en la esquina derecha del menú superior.

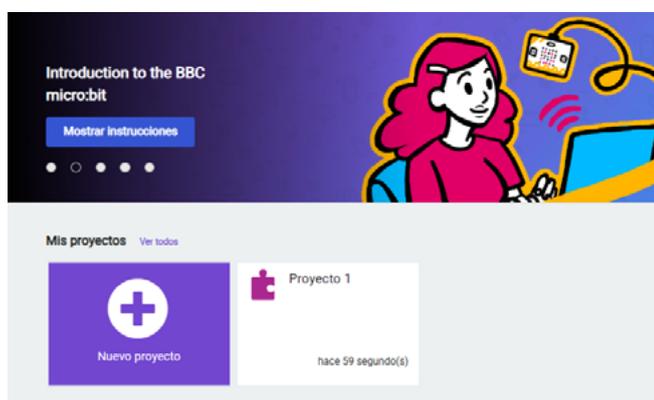
Al clicar sobre el icono, se abrirá una ventana emergente con la opción de compartir.



Puedes compartir un enlace directo de tu programa, publicarlo en las redes sociales o incrustarlo en tu página web o blog. También lo puedes compartir mediante un código QR.



Por otro lado, todos los programas que realices con Makecode quedarán guardados automáticamente en tu perfil.



PROGRAMA DESDE UNA TABLETA

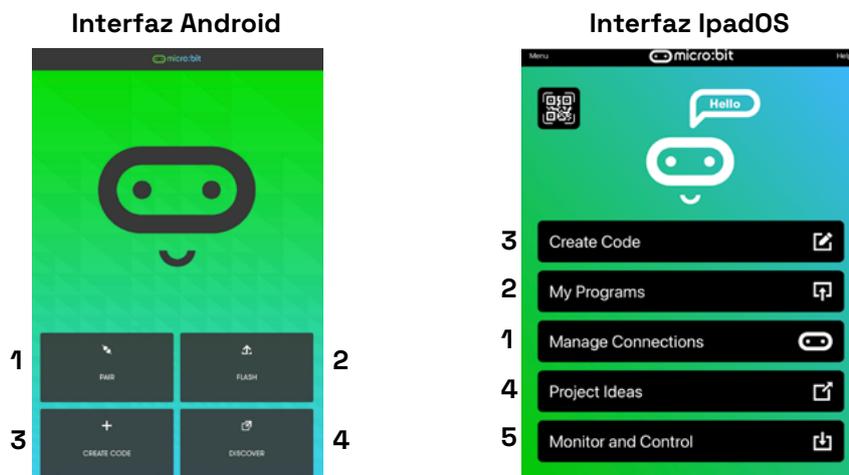
5.1. CONOCE EL ENTORNO DE PROGRAMACIÓN

Para empezar a programar Smart Cutebot desde la tableta, necesitas disponer de una placa micro:bit, un cable micro-USB, el robot montado con las tres pilas AAA puestas y suficiente batería y una tableta.



Para programar la placa micro:bit desde la tableta, tienes que instalar la aplicación “micro:bit” desde la Play Store (Android) o la App Store (iPadOS).

Una vez tengas la aplicación descargada, ábrela y verás su menú principal. La interfaz del menú de las tabletas con sistema operativo Android y iPadOS varía un poco, tal y como puedes comprobar en las siguientes imágenes:



A continuación te describiremos las siguientes opciones de los menús:

- 1) **Pair (Android) / Manage Connections (IpadOS):** para conectar la micro:bit a la tableta digital mediante *bluetooth*.
- 2) **Flash (Android) / My Programs (IpadOS):** son diferentes programas que puedes descargar en la micro:bit.
- 3) **Create Code:** es el entorno de programación de micro:bit, es decir, el Microsoft MakeCode.
- 4) **Discover (Android) / Project Ideas (IpadOS):** te redirige a una página web de micro:bit con diferentes ejemplos de proyectos que puedes crear.
- 5) **Monitor and Control (IpadOS):** para controlar la micro:bit desde la tableta digital y, al mismo tiempo, ver desde esta qué ocurre en la placa.

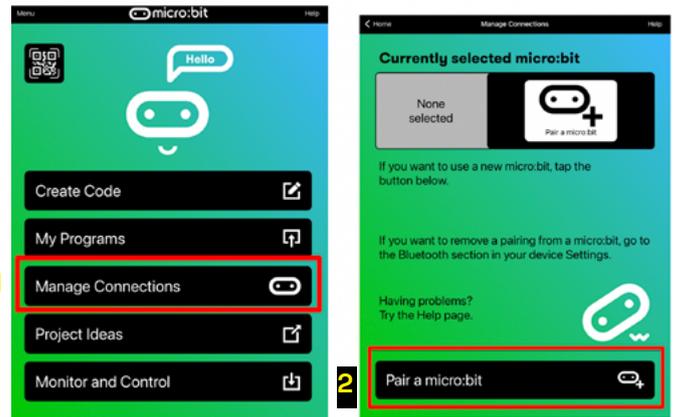
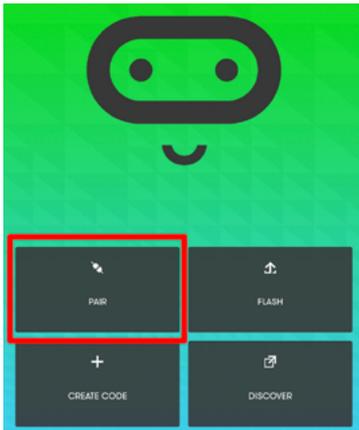
5.2. SMART CUTEBOT

CONECTA LA PLACA

1 Alimenta la micro:bit insertándola en la ranura Cutebot y enciende el robot con el interruptor delantero. Para vincular la micro:bit a la tableta:

a) Si lo haces con una tableta Android, pulsa “Pair” y sigue las instrucciones.

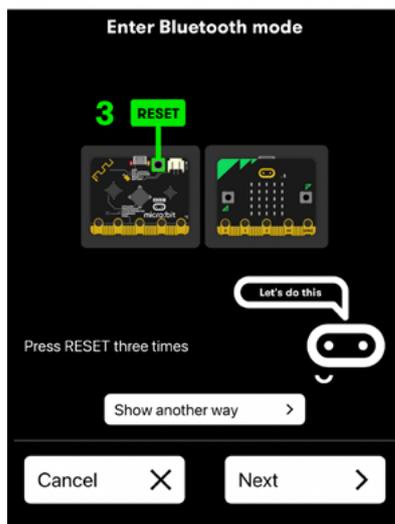
b) Si lo haces con una tableta iPadOS, pulsa la opción “Manage Connections” y, seguidamente, la opción “Pair a micro:bit”.



2 Al llegar a este punto, sigue las indicaciones para conectar el modo *bluetooth* de la placa. Tienes dos formas de activar el *bluetooth*:

a) Pulsa tres veces el botón de reinicio de la placa física. Una vez hecho, presiona el botón “Next” en la pantalla.

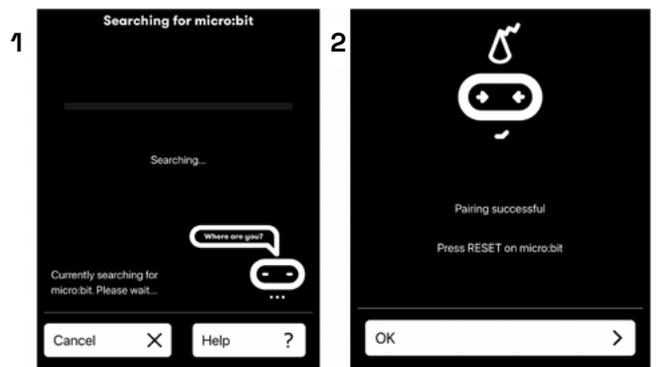
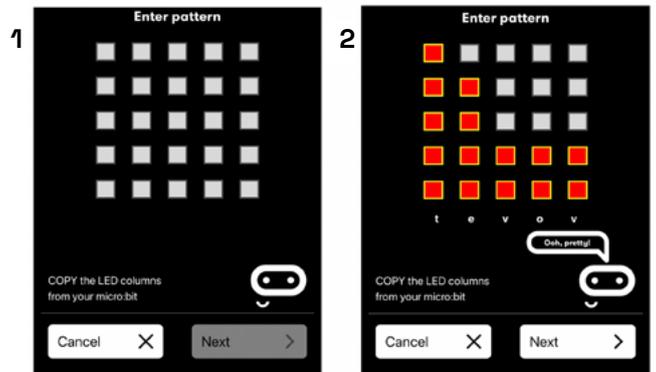
b) Pulsa a la vez los botones A y B de la placa y, después, pulsa y suelta el botón de reinicio. Una vez hecho, presiona el botón “Next” en la pantalla.



5.2.

SMART CUTEBOT

- 3 Reproduce el patrón de luces que se muestra en la micro:bit física en la pantalla de tu *app*, clicando sobre los ledes en blanco. Después presiona “Next”.
- 4 Cuando pulses “Next”, la aplicación te avisará de que, a veces, durante el proceso de vinculación puede que se necesite un código de verificación. Esto lo puedes conseguir presionando sobre el botón A de la placa. Clica en “Next” para seguir con la vinculación.
- 5 La aplicación comenzará a buscar y vincular la placa. Es posible que durante este proceso el dispositivo móvil te pida vincular la placa mediante una pequeña notificación. Puedes comprobar si hay tal notificación abriendo el centro de notificaciones del dispositivo deslizando por la pantalla de arriba abajo. Acepta esta vinculación desde el dispositivo. Una vez vinculada, pulsa el botón de reinicio de la placa y dale a “OK”.



5.3. SMART CUTEBOT

CONOCE EL ENTORNO DE PROGRAMACIÓN

Cuando tu placa esté vinculada a la tableta, vuelve al menú principal de la aplicación y comienza a programar presionando la opción “Create Code”. Una vez cliques en esta opción, se abrirá el entorno de programación Microsoft MakeCode. Este entorno es exactamente el mismo que se utiliza para programar la placa a través del entorno. Mira el punto 4.1. “Conoce el entorno de programación” de este documento para conocer la interfaz de programación Microsoft Makecode.

Una vez que consigas el entorno de programación, ya puedes comenzar a codificar. Consulta el apartado 4.3. “Comienza a programar” de este documento para hacer un primer programa. Cuando tengas el programa listo, ve al punto siguiente, 5.4. “Transfiere el programa a la placa”, para saber cómo transferir el programa a la placa.

5.4. SMART CUTEBOT

TRANSFIERE EL PROGRAMA A LA PLACA

Para transferir el programa a la placa, pulsa sobre el mismo botón lila “Descargar”. La aplicación te pedirá que vuelvas a conectar el *bluetooth* de la placa. Recuerda, puedes hacerlo pulsando tres veces el botón posterior (reinicio) de la placa o apretando A+B y, posteriormente, pulsando y soltando el botón de reinicio. Sigue siempre las indicaciones que te va mostrando la aplicación. Cuando el *bluetooth* esté conectado, el programa se transferirá a la placa y verás cómo se reproduce directamente.

Importante: cada vez que hagas un programa y lo transfieras a la placa, deberás repetir esta operación.

¿TE ATREVES CON LOS RETOS?

NIVEL I

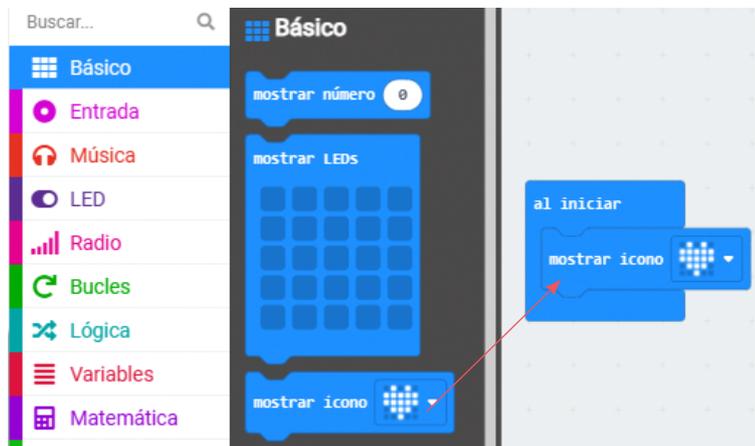
¡NO CHOQUES!

Imagina que Cutebot está conduciendo en línea recta. De repente, se encuentra con un obstáculo en medio del camino. Prográmalo para que, si se encuentra este obstáculo a menos de 5 cm, se pare.

- 1 Para hacer el programa, abre un nuevo proyecto. Vuelve al menú principal del entorno, clicando en el icono de inicio (casa)  y haz clic en "Nuevo proyecto". Recuerda: tendrás que volver a añadir el bloque de extensiones "cuteBot".

Antes de que el robot se mueva, haremos un pequeño programa para que se muestre una imagen durante unos segundos..

Dentro del bloque "al iniciar" arrastra el bloque "mostrar icono"  de la categoría "Básico".



- 2 Arrastra y coloca bajo el anterior orden el bloque "pausa (ms) 100"  y cambia el "100" por 5 segundos "5000 ms".



- 3 Seguidamente, coloca debajo el bloque "borrar la pantalla"  de la categoría "Básico".



4 Hasta aquí la primera parte del programa. Ahora, tenemos que hacer que, cuando el robot detecte un objeto con ultrasonido HC-SR04 a 5 centímetros o menos, se pare.

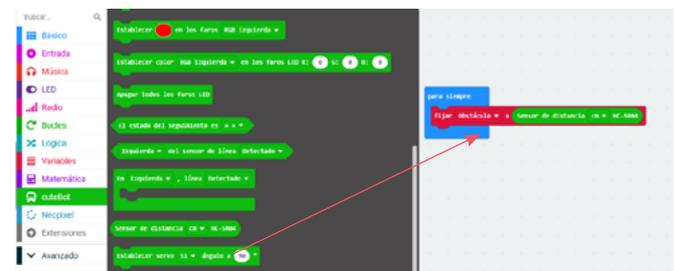
Para ello, utilizaremos una variable. Una variable es como una caja etiquetada con un nombre donde puedes guardar y cambiar información, como números o palabras, para que el programa o el robot la pueda utilizar en diferentes situaciones. En nuestro caso, almacenaremos la distancia donde se encuentra el obstáculo. Para crear la variable, ve al grupo rojo “Variables”, pulsa sobre “crear una variable” y llámala “obstáculo”.

5 Verás que, al crearla, se te abren tres bloques diferentes. Arrastra el bloque “fijar obstáculo a 0” y sitúalo en el bloque “para siempre”.

6 Dentro del 0 del bloque “fijar obstáculo a 0” hay que indicar la distancia a la que se encuentra el objeto que detectar. Para hacerlo, como vuelve a ser una acción específica de Smart Cutebot, busca dentro del grupo verde cuteBot el bloque “sensor distancia cm HC-SR04”.

7 Hasta aquí tenemos programado que el robot vaya almacenando la distancia del sensor HC-SR04 hasta la variable “obstáculo”. Ahora hay que programarla para que, si este valor es inferior a 5, se pare, y, si no, que continúe avanzando.

Como hablamos de una condición (si..., si no...), busca dentro de la categoría “Lógica” el bloque “si-verdadero-entonces / si no” y arrástralo bajo el bloque “fijar obstáculo a -sensor distancia cm HC-SR04-”.



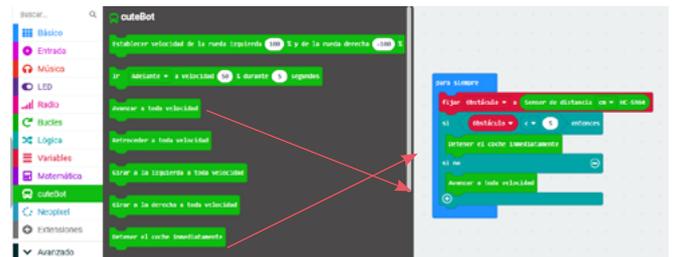
8 Para indicar que el valor de la distancia sea menor a 5, busca el bloque “ $0 < 0$ ” dentro de la categoría “Lógica” y arrástralo dentro del “verdadero” del bloque “si - verdadero- entonces”.



9 Ahora tienes que cambiar los valores de los dos ceros. En el primero, has de arrastrar el bloque “obstáculo” que encontrarás dentro del grupo rojo “Variables”. En el segundo, clicas sobre él para borrar el 0 y escribir un 5. El bloque te tiene que quedar “obstáculo < 5”.



10 Dentro del bloque “si obstáculo < 5 entonces” tienes que indicar qué quieres que haga en cada opción. Como son acciones que tiene que hacer el Smart Cutebot, las encontrarás dentro del grupo verde “cuteBot”. En el primer espacio tienes que arrastrar y soltar el bloque “parar el coche inmediatamente” para que, cuando el obstáculo esté a 5 centímetros o menos, se pare. En el segundo espacio, justo bajo el “entonces”, arrastra el bloque verde “avanzar a toda velocidad”.



11 Hasta aquí has realizado el programa para resolver el reto planteado, ahora solo tienes que transferir el programa.

Utiliza tu propia mano como obstáculo para que el sensor de distancia detecte que tiene algo delante y el robot se pare.

Importante: como le hemos proporcionado unos segundos de espera para mostrar una imagen en la pantalla led, el robot no comenzará a avanzar de manera inmediata. Recuerda que tienes que desenchufar el cable micro-USB y colocar el robot en el suelo.

NIVEL II

ACELERACIÓN

Comenzamos con Smart Cutebot parado. Prográmalo para que acelere gradualmente y que, cuando llegue a la velocidad máxima, se pare.

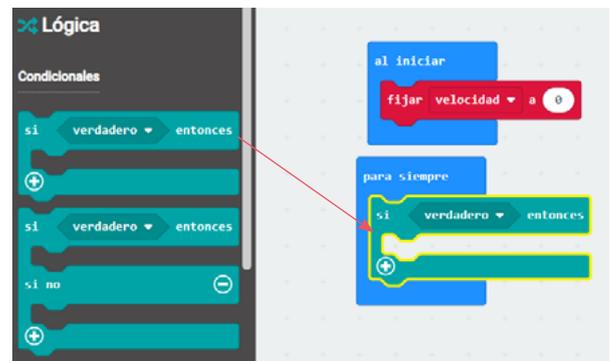
Es decir, tienes que comenzar con el robot a una velocidad 0 y, poco a poco, el valor de la velocidad tiene que ir aumentando hasta que llegue al 100 %. Y una vez que la velocidad sea 100 %, el robot debe pararse.

Abre un nuevo proyecto. Vuelve al menú principal del entorno clicando el icono de inicio (casa)  y haz clic a "Nuevo proyecto". Recuerda que tienes que volver a añadir el bloque de extensiones "cuteBot".

- Comienza el programa creando una variable. Esta irá almacenando en todo momento el valor de la velocidad del robot, e irá cambiando constantemente. Para crear la variable, ve al grupo rojo "Variables" y pulsa sobre "crea" una variable y llámala "velocidad". Verás que, al crearla, se te abren tres bloques diferentes. Arrastra el bloque "fijar velocidad a 0"  y sitúalo bajo el bloque "al iniciar". Con esto estás ordenando al robot que, cuando comience el programa, el valor de la velocidad siempre será 0, es decir, que siempre que pueda tiene que comenzar parado.



- Dentro del bloque "al iniciar" ya no tienes que añadir ninguno más. Ahora te has de centrar en el bloque "para siempre". Como queremos que la velocidad no supere el 100 %, antes de decirle que se mueva, prográmalo para que reinicie el valor de la velocidad cuando esta llegue a 100. Para hacerlo, busca dentro del grupo azul "Lógica" el bloque "si -verdadero- entonces"  y arrástralo dentro del bloque "para siempre".



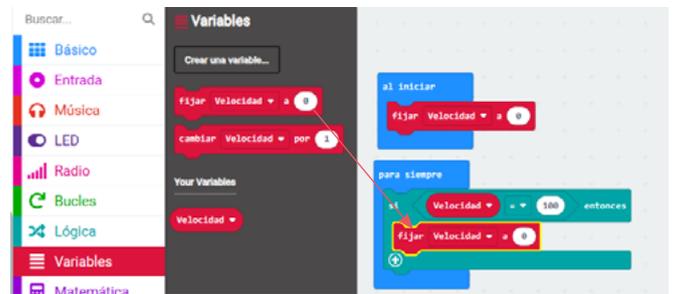
- Ahora le tienes que decir que, si el valor de velocidad es 100, que vuelva a establecerse en 0. Primero busca dentro del grupo "Lógica" el bloque "0 = 0"  y añádelo dentro de "verdadero" del bloque "si -verdadero- entonces".



4 Tras ello tienes que cambiar los valores del bloque “0 = 0”. En este caso, queremos que sea la velocidad 100. Por tanto, primero busca dentro del grupo rojo “Variables” el bloque “velocidad” **Velocidad** y arrástralo a donde esté el primer 0. En segundo lugar, clicas sobre el segundo 0, bórralo y escribe 100. Una vez hecho esto, el bloque te debería de quedar así: “si velocidad = 100 entonces”.



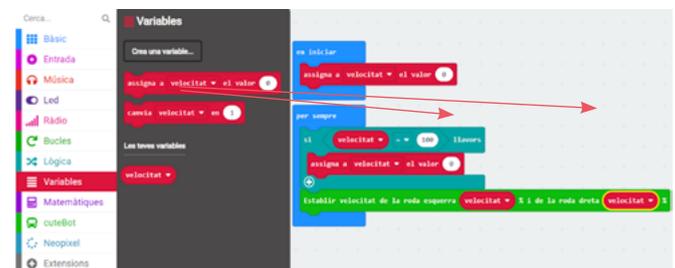
5 Cuando la velocidad sea 100, quieres que reinicie el valor de la velocidad. Busca dentro del grupo rojo “Variables” el bloque “fijar velocidad a 0” **fijar velocidad a 0** y colócalo dentro del bloque “si velocidad = 100 entonces”.



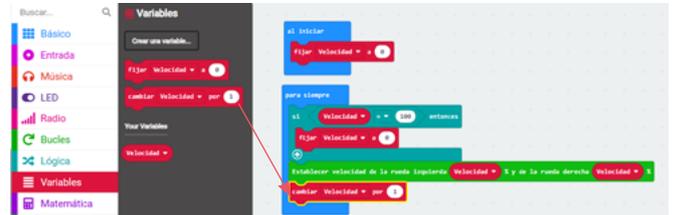
6 Te falta programar que el robot se mueva a la velocidad indicada y que vaya aumentando cada vez. Primero haremos que se mueva. Como es un bloque relacionado con el robot, busca dentro del grupo verde “cuteBot” el bloque “establecer velocidad de la rueda izquierda 100% y de la rueda derecha -100%” y arrástralo debajo del bloque “si velocidad = 100”.



7 Más adelante programaremos que la velocidad vaya aumentando y se guarde este valor en la variable “velocidad”. Arrastra dos bloques “velocidad” **Velocidad** del grupo rojo de “Variables” y colócalos uno sobre el 100% y otro sobre el -100% del bloque “establecer velocidad de la rueda izquierda 100% y de la rueda derecha -100%”. El bloque te debería quedar así: “establecer velocidad de la rueda izquierda velocidad % y de la rueda derecha velocidad %”.



- 8 Finalmente, te falta ir aumentando la velocidad. Busca el bloque “cambia velocidad en 1” **“Cambiar Velocidad por +1”** dentro del grupo rojo “Variables” y arrástralo hasta debajo del bloque “establecer velocidad de la rueda izquierda velocidad % y de la rueda derecha velocidad %”.



Hasta aquí has realizado el programa para resolver el reto planteado, ahora solo tienes que transferirlo tal y como se ha explicado anteriormente.

Importante: si trabajas con un cable micro-USB, te aconsejamos que, antes de transferir el programa a la placa micro:bit, la desenchufes del Smart Cutebot. Transfiere el programa y, con el robot apagado, vuelve a enchufarla al Smart Cutebot.

Al acabar, coloca el robot en el suelo y enciéndelo.

Puedes cambiar los valores para hacer que el Smart Cutebot acelere más rápido.

NIVEL III

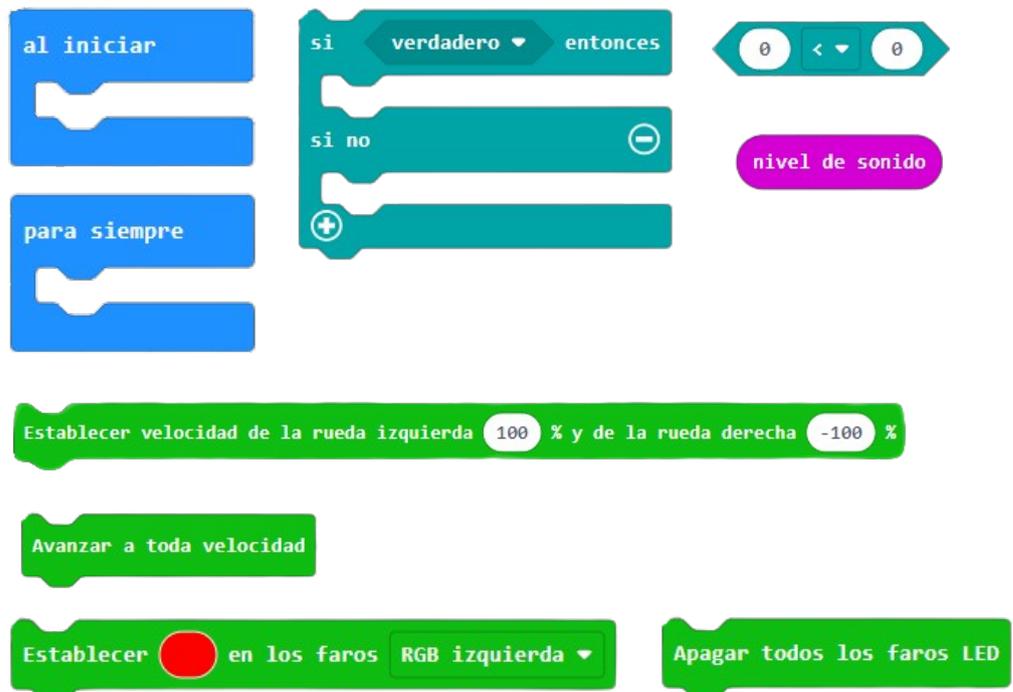
TÚNELES

Imagina que el robot es un coche real, que va conduciendo tranquilamente de día, sin llevar las luces encendidas. La carretera por donde circula, no obstante, tiene muchos túneles muy largos para los que se necesita encender las luces.

- 1 Programa el Smart Cutebot para que se mueva a una velocidad alta, pero que, cuando entre en una zona oscura (como podría ser un túnel), reduzca la velocidad y encienda los dos ledes RGB.

Para hacer el programa, necesitarás los siguientes bloques de programación que encontrarás dentro de las categorías: “Básico”, “Lógica”, “Entrada” y “cuteBot”.

Bloques clave:



- 2 Recuerda que tienes que abrir un nuevo proyecto. Vuelve al menú principal del entorno clicando el icono de inicio (casa)  y haz clic en “Nuevo proyecto”. Tendrás que volver a añadir el bloque de extensiones “cuteBot”.

Una vez que acabes de programar, transfíere el programa a la placa. Importante: si trabajas con cable micro-USB, te aconsejamos que, antes de transferir el programa a la placa micro:bit, la desenchufes del Smart Cutebot. Transfiere el programa y, con el robot apagado, vuelve a enchufarla al Smart Cutebot. Al acabar, coloca el robot en el suelo.

Puedes utilizar mesas, sillas o carpetas para conseguir zonas oscuras mientras el Cutebot avanza por el suelo.

6.1. SMART CUTEBOT

POSIBLES SOLUCIONES

NIVEL III TÚNELES

Imagina que el robot es un coche real, que va conduciendo tranquilamente de día, sin llevar las luces encendidas. La carretera por donde circula, no obstante, tiene muchos túneles muy largos para los que se necesita encender las luces. Programa el Smart Cutebot para que se mueva a una velocidad alta, pero que, cuando entre en una zona oscura (como podría ser un túnel), reduzca la velocidad y encienda los dos ledes RGB.

- 1 Primero, abre un nuevo proyecto. Para ello, vuelve al menú principal del entorno clicando el icono de inicio (casa)  y haz clic en "Nuevo proyecto". Tendrás que volver a añadir el bloque de extensiones "cuteBot".

Dentro del bloque azul "al iniciar" , arrastra el bloque "avanzar a toda velocidad"  de la categoría verde "CuteBot". De esta manera, al iniciar el programa, el coche directamente se moverá.



- 2 Tienes programado que el robot se mueva directamente a máxima velocidad. Ahora le has de programar que los ledes se enciendan y la velocidad disminuya si hay poca luz. Para ello necesitas un condicional de la categoría "Lógica", arrastra el bloque "si verdadero entonces"  y suéltalo dentro del bloque "para siempre".



- 3 Para indicar el nivel de luz, necesitas arrastrar el bloque de comparación "0 < 0"  que encontrarás en la categoría "Lógica" y situarlo dentro de "verdadero".



6.1. SMART CUTEBOT

4 Tras ello, arrastra el bloque “nivel de luz” que encontrarás dentro de la categoría rosa “Entrada” y suéltalo en el primer “0” del bloque “ $0 < 0$ ”.



5 El sensor de luz mide la intensidad de luz ambiental que recibe el panel led de la placa. El sensor devuelve un valor de entre 0 (oscuridad total) y 255 (máxima claridad). Cuanto mayor sera el número, más cantidad de luz recibe la placa.

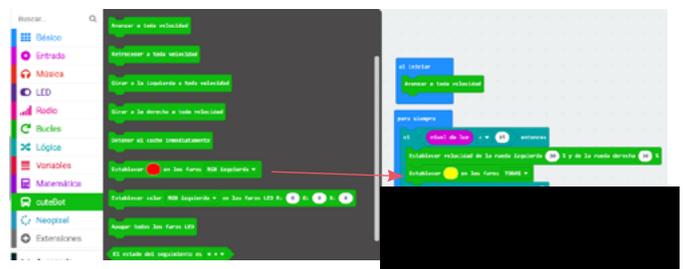
En este caso, queremos que los faros ledes se enciendan cuando esté oscuro. Para conseguirlo, podemos indicar que el nivel de luz sea inferior a 85. Por tanto, cambia el segundo 0 del bloque “nivel de luz < 0” por el valor del nivel de luz, es decir, 85.



6 Después indica qué quieres que haga el robot cuando el nivel de luz sea bajo. Para ello, arrastra el bloque “establecer velocidad de la rueda izquierda 100% y de la rueda derecha -100%” debajo de “si nivel de luz < 85 entonces”. Este bloque lo encontrarás dentro de la categoría “cuteBot”. Después cambia los valores por 30% en ambos casos.



7 Falta añadir que se enciendan los dos ledes. Para hacerlo, arrastra el bloque “establecer color en los faros RGB amarillo” bajo el bloque “establecer velocidad de la rueda izquierda 30% y de la rueda derecha 30%”. El bloque lo encontrarás dentro de la categoría verde “cuteBot”. Para hacer que solo se encienda el led derecho, pulsa sobre “derecha” y clicla la opción “TODAS”.

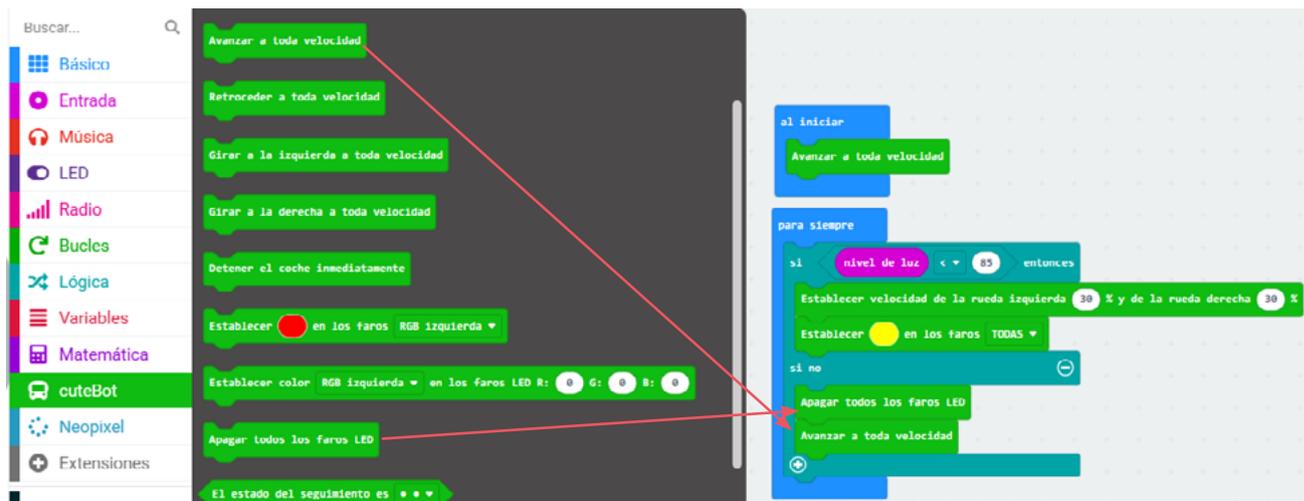


6.1.

SMART CUTEBOT

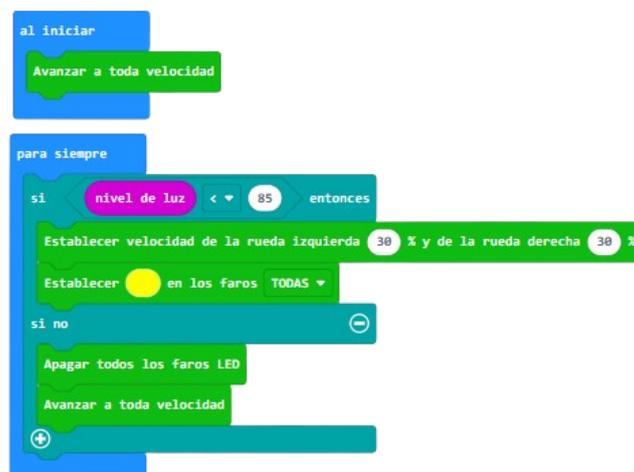
- 8 Finalmente, solo te falta indicar que, si el nivel de luz no es bajo, se apaguen todos los ledes y vuelva a avanzar a toda velocidad. Para hacerlo, arrastra los bloques “apagar todos los faros ledes” y “avanzar a toda velocidad” que encontrarás en la categoría verde “cuteBot”, debajo de “entonces”.

Así es como queda el programa final:



- 9 Una vez acabes de programar, transfíere el programa a la placa.

Importante: si trabajas con cable micro-USB, te aconsejamos que, antes de transferir el programa a la placa micro:bit, la desenchufes del Smart Cutebot. Transfiere el programa y, con el robot apagado, vuelve a enchufarla al Smart Cutebot. Al acabar, coloca el robot en el suelo y enciéndelo.



CONSEJOS DE SEGURIDAD

- Cuando Smart Cutebot esté en funcionamiento, no bloquee su movimiento ni tampoco permitas que el robot se cargue demasiado. El movimiento del robot es impulsado por motores y, si se bloquean o se cargan excesivamente, provocarán daños en el robot.
- No desmontes ni alteres de ninguna otra manera la estructura del kit.
- No utilices Smart Cutebot con agua o con las manos mojadas.
- No dobles ni presiones las clavijas utilizadas por los sensores y los pines de entradas y salidas.
- No coloques Smart Cutebot en algún lugar donde se pueda caer con facilidad.
- Utiliza el interruptor de la base para apagar Smart Cutebot cuando no se esté usando.
- Si se prevé un periodo largo de inactividad, retira la batería de su soporte.
- No uses objetos afilados para rallar o rascar partes de la placa base ni del carro.
- No guardes ni utilices Smart Cutebot en entornos extremadamente calientes o fríos.

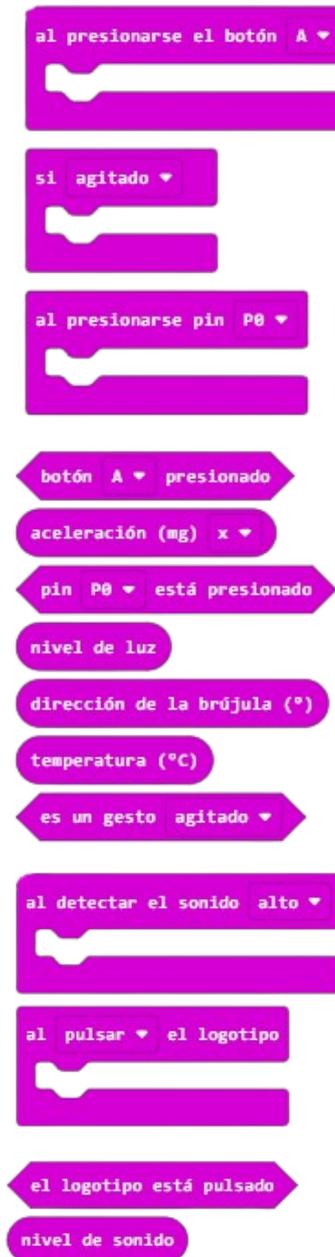
BLOQUES DE PROGRAMACIÓN

A continuación, explicaremos en qué consiste cada grupo de programación.

Básico (azul): bloques de programación de la pantalla de led de la placa micro:bit para iniciar o para los programas que elaboremos.



Entrada (rosa): bloques de programación de entradas (botones, sensores...).



Música (naranja): bloques de programación para hacer que la placa suene a través de su zumbador.



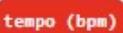
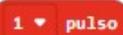
Tono



Volumen



Tempo



micro:bit (v2)



Led (lila): bloques de programación para ledes.

graficar x 0 y 0

invertir x 0 y 0

ocultar x 0 y 0

punto x 0 y 0

plot bar graph of 0
up to 0
+

... más

graficar x 0 y 0 brillo 255

punto x 0 y 0 brillo

brillo

ajustar brillo 255

activar leds falso

detener animación

establecer modo de visualización blanco y negro

Radio (rosa): bloques de programación para conectar dos o más placas Smart Cutebot y enviarse entre ellas mensajes.

Grupo

radio establecer grupo 1

Enviar

radio enviar número 0

radio enviar valor name = 0

radio enviar cadena ""

Recibir

al recibir radio receivedNumber

al recibir radio name value

al recibir radio receivedString

paquete recibido intensidad de señal ▾

... más

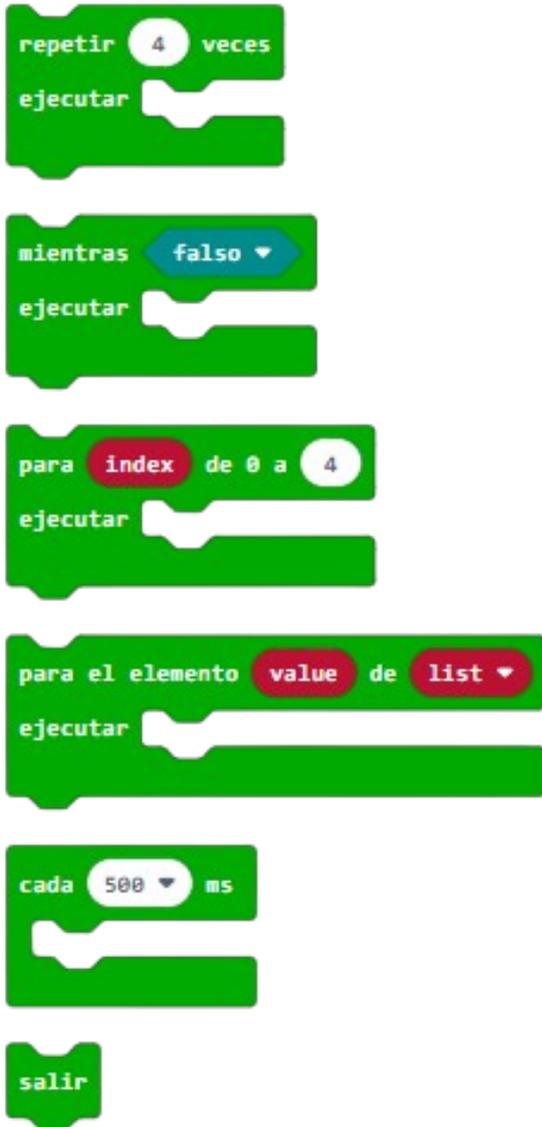
radio establecer potencia de transmisión 7

radio establecer número de serie de transmisión verdadero ▾

establecer banda de frecuencia de radio 0

radio lanzar evento
desde fuente MICROBIT_ID_BUTTON_A ▾
con valor MICROBIT_EVT_ANY ▾

Bucles (verde): bloques de programación para crear bucles (repeticiones, cada..., continúa...).

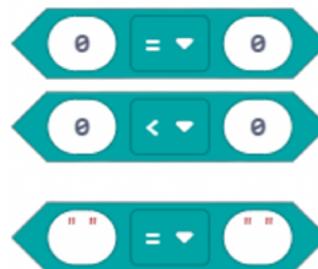


Lógica (azul): bloques de programación de condiciones, comparaciones y booleano.

Condicionales



Comparación



Booleano



Variables (rojo): bloques de programación para crear y editar variables.

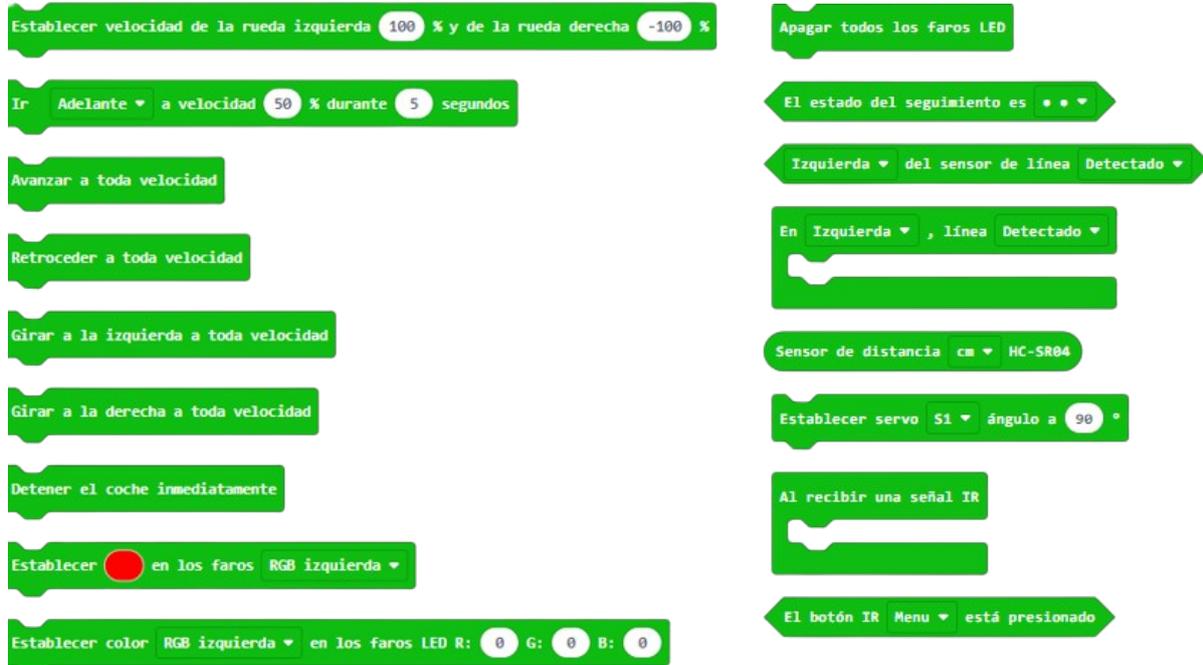
En la categoría “Variables” no hay bloques por defecto. Una vez se crea la variable, aparecen bloques para trabajar con ella.

Matemática (lila): bloques de programación matemáticos (operaciones, azar...).

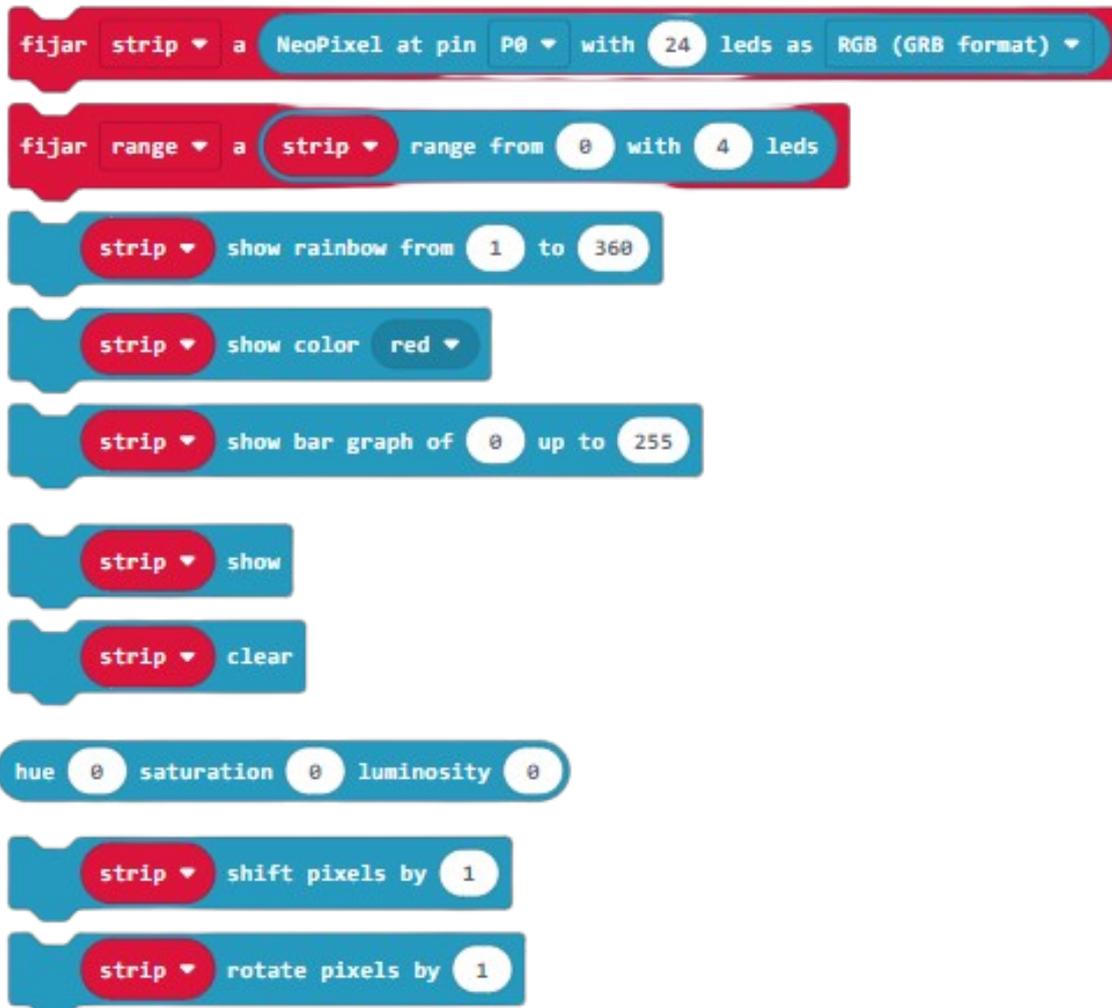


Extensiones: gracias a sus “pines”, se pueden añadir componentes externos a la placa, un motor o un ultrasonido. Esta categoría permite que se añadan a los bloques de programación de los diversos componentes compatibles.

CuteBot (verde): bloques de programación para Smart Cutebot. Se encuentran en todos los bloques para hacer mover el robot, encender o apagar los ledes y programar el IR, entre otros..



Neopixel (azul): bloques de programación para hacer funcionar el Rainbow LED.



Dentro del apartado de “avanzar”, hay más categorías de bloques de programación que hacen referencia a funciones, listas y pines, entre otros.