

# Introducción

## 1. Enfoque pedagógico

Con Makeblock Education, el desarrollo de conocimientos y habilidades es la combinación de herramientas digitales y físicas que trabajan juntas para ofrecer una experiencia de aprendizaje práctica. Los estudiantes se involucran activamente con el tema de la lección para resolver un problema o crear algo nuevo. Permitiéndoles desarrollar conceptos teóricos a partir de experiencias prácticas, así como transferir y aplicar estos conceptos a tareas prácticas profundizando sus conocimientos. Esto incluye aprender de los errores mediante un análisis detallado y sistemático de los procesos involucrados.

Con el aprendizaje práctico, los estudiantes gestionan y fomentan su proceso educativo desempeñando un papel más activo durante la clase, en vez de simplemente escuchar una conferencia del profesor.

Esta introducción explica de forma general las nuevas características del mBot2 y los beneficios de su uso en la educación. Muestra brevemente los pasos para comenzar a programar el robot con mBlock5, el editor basado en bloques de Makeblock Education, y describe a continuación las Actividades de Introducción individuales. Estas actividades son una experiencia de aprendizaje con diferentes escenarios y explican las nuevas funciones paso a paso. Partiendo de referencias reales en las tareas asignadas, se analizan los contextos de los sensores, los actuadores y el pensamiento computacional, junto con ejemplos de códigos fáciles de entender y ampliar para cada tarea. Las actividades siempre proponen involucrar más medios digitales y pensar en el futuro para aplicar el aprendizaje a nuevos desafíos. Se termina con una reflexión de la actividad realizada.

## 2. Presentación del mBot2

El mBot2 es un robot educativo de última generación diseñado para la informática y la educación STEAM. Al tener capacidades ampliadas lo convierte en una solución de iniciación ideal para el primer ciclo de educación secundaria, pero también puede llegar a usarse hasta en la educación secundaria superior y más allá. El mBot2 está diseñado para que los estudiantes realicen lecciones



interactivas e inteligentes que sean atractivas y divertidas, y reflejen también aplicaciones del mundo real de la tecnología punta.

El mBot2 funciona con el CyberPi, un microcontrolador potente y versátil. Sus sensores y actuadores integrados (micrófono y altavoz, unidad de medición inercial con giroscopio y sensor de aceleración, un sensor de luz, botones para operaciones de menú y una pantalla en color) se complementan con un sensor ultrasónico de distancia y un sensor de seguimiento de línea con cuatro elementos RGB.

Junto a esta gama de sensores y actuadores, el mBot2 es capaz de comunicarse por Wi-Fi, permite una amplia gama de aplicaciones en temas de Codificación, Robótica, Ciencia de Datos e Inteligencia Artificial, en relación con otras materias como las Matemáticas, la Física. etc.

Por ejemplo, los profesores pueden realizar una conexión inalámbrica entre varios mBots2 en un aula para crear una red local de robots que se comuniquen entre ellos, compartan información y realicen tareas. También pueden usar un CyberPi autónomo como dispositivo inteligente para comunicarse con el mBot2, creando un ecosistema inteligente o un divertido control remoto. Cuando está conectado a Internet, el mBot2 puede realizar funciones avanzadas como el reconocimiento de voz o la comunicación con la nube para obtener información.

En las siguientes subsecciones se realizará una presentación de los sensores y actuadores específicos del mBot2, con cuyos conocimientos podrás realizar las actividades de introducción. También encontrarás una descripción general de dichos sensores y actuadores al final de este documento.

He aquí una descripción general de los diferentes componentes del mBot2:

### Placa del mBot2

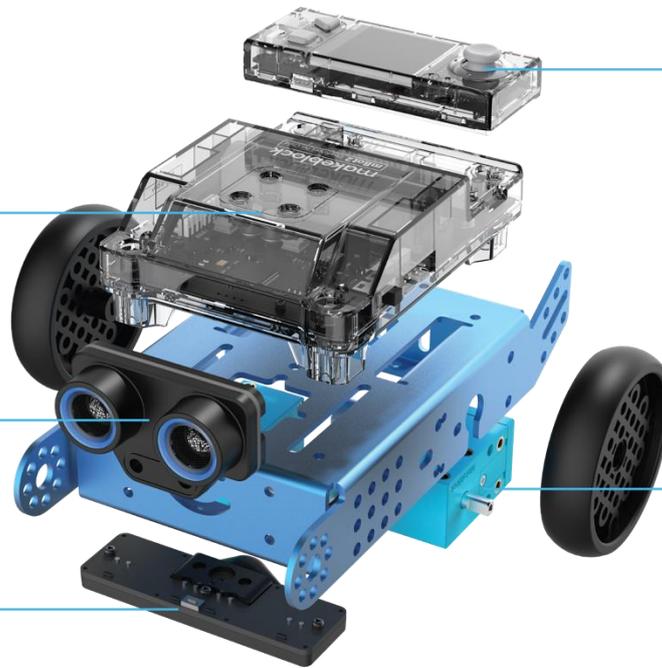
Compatible con una amplia variedad de componentes externos. Incluye una batería recargable incorporada.

### Sensor ultrasónico 2

La detección de objetos incluye 8 LED programables para una interacción mejorada.

### Sensor Quad RGB

Las 4 sondas del sensor admiten el reconocimiento de colores, así como programas de detección de líneas básicos y avanzados.



### CyberPi

Microprocesador ESP-32 para comunicación inalámbrica. Compatibilidad con codificación basada en bloques y codificación Python.

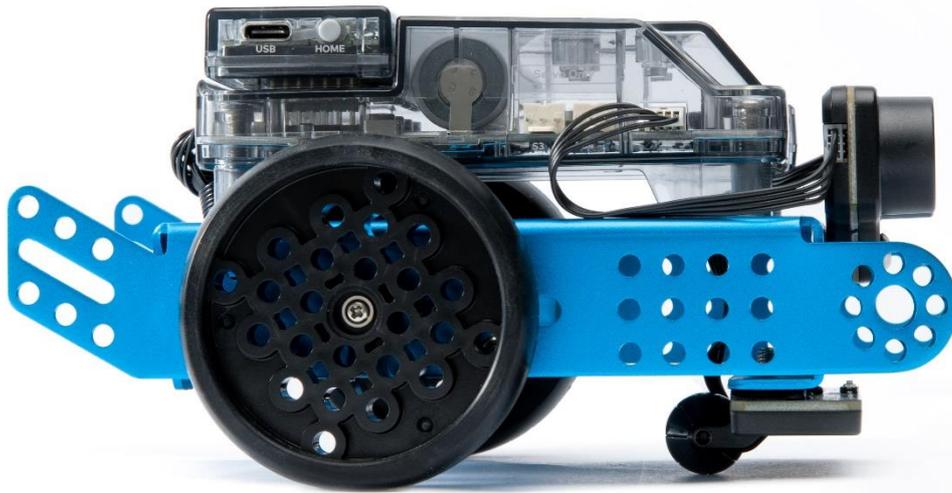
### Motores codificadores

Miden la rotación con una precisión de 1 grado, la distancia recorrida y controlan con precisión la velocidad hasta 200 RPM.

## 2.1 Motores codificadores

Los motores del mBot2 están equipados con codificadores ópticos que permiten realizar controles de alta precisión. Así, los estudiantes pueden controlar con precisión la rotación, la velocidad y la posición de las ruedas y el robot. Además, los motores también se pueden usar como servomotor, e incluso como botones de mando, para enviar datos al sistema, al igual que un sensor.

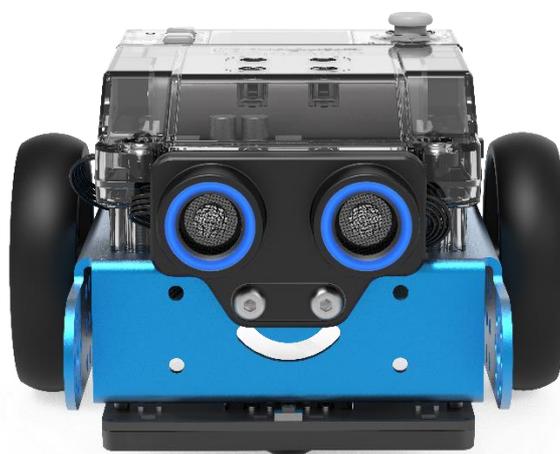
Gracias a estas funciones, las lecciones pueden ser más realistas y educativas que las lecciones realizadas con robots que tienen menos control de sus motores. Las actividades pueden integrar conceptos matemáticos, por ejemplo, conduciendo distancias precisas, calculando giros exactos e incluso trazando una ruta a través de un laberinto y transfiriendo los resultados al ordenador.



La lección 1 de las Actividades de Introducción ofrece un preámbulo de los diferentes bloques de código para los Motores Codificadores en el entorno de programación mBlock, así como una tarea fácil para que los estudiantes exploren las posibilidades por sí mismos.

## 2.2 Sensor ultrasónico y sensor Quad RGB

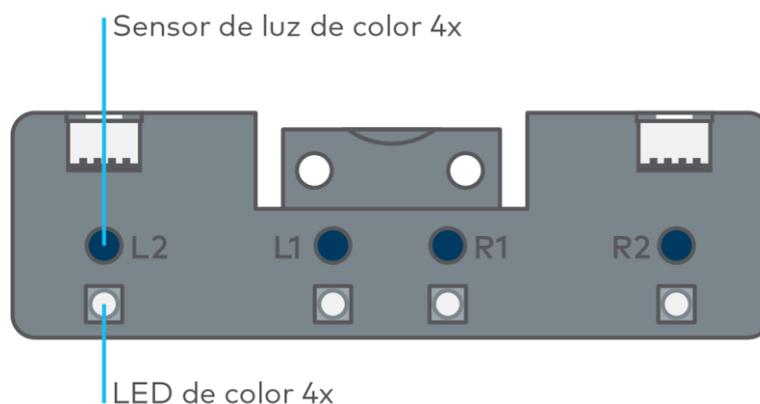
El sensor ultrasónico que viene con el mBot2 puede realizar lecturas más precisas y consistentes que sus predecesores. Además, está equipado con LED azules que permiten nuevas formas de interacción con el robot. Los estudiantes pueden usar estas luces adicionales para mostrar las respuestas.



El mBot2 tiene otro componente avanzado: el sensor Quad RGB. Este sensor tiene la capacidad no solo de identificar colores, sino también de rastrear líneas para ayudar al mBot2 a seguir un trazado, o detectar cruces o giros de 90°, todo ello al mismo tiempo.

Las funciones básicas se pueden realizar con pocos conocimientos previos, pero debido a sus características avanzadas, es importante saber utilizarlas de forma óptima para sacar el mayor provecho del sensor.

En lugar de usar 2 sensores para rastrear una sola línea frente al robot, el sensor Quad RGB tiene 4 sensores que le permiten identificar una gama más amplia de condiciones: mientras sigue un trazado, el sensor puede identificar cruces a la izquierda, derecha o incluso a ambos lados, lo que significa que el mBot2 puede navegar a través de mapas más complejos que otros robots. Y dado que cada uno de los cuatro sensores es un sensor con color propio, las marcas de colores pueden indicar posiciones en el mapa: el robot "conoce" su ubicación.

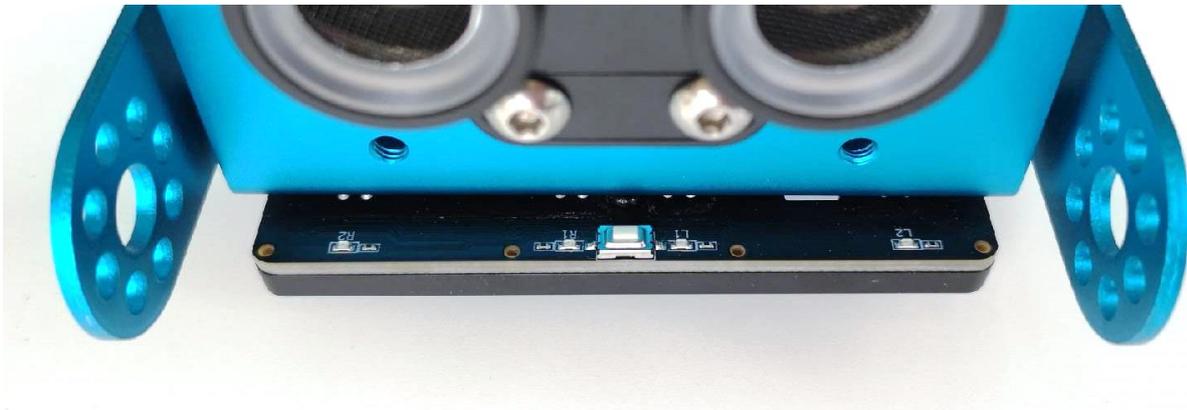


Este rango de detección más amplio se puede utilizar para mapas con una red de líneas con múltiples mBot2 moviéndose entre diferentes estaciones como en un "Almacén Inteligente", o simular las carreteras de una ciudad, con sus diferentes señales de tráfico y normas de circulación.

Los sensores de seguimiento de línea anteriores utilizaban luces infrarrojas que no son visibles para el ojo humano. En cambio, estos sensores operan en una sola longitud de onda, mientras que el nuevo sensor puede ver tres longitudes de onda diferentes simultáneamente, como el ojo humano: rojo, verde y azul. Nuestro cerebro interpreta una mezcla de la intensidad de estas longitudes de onda como percepción del color. El sensor funciona de manera similar y, además de

percibir colores, intenta diferenciar un trazado del fondo para realizar tareas de seguimiento de línea.

Con el mapa proporcionado, tenemos marcadores de color adicionales dentro del recorrido para activar acciones adicionales si se programan en consecuencia. Para que el sensor interprete estos colores como parte de la línea, debe calibrarse con el color más brillante del trazado; el amarillo. Coloca el sensor sobre el código de color amarillo con el mBot2 encendido y haz doble clic en el botón pequeño de la parte superior del sensor (consulta la siguiente imagen). Cuando los LED comiencen a parpadear, pasa el sensor por el color y el fondo blanco hasta que deje de parpadear (2-3 segundos). Después, el color será considerado como parte de la pista (para obtener información más detallada, consulta la información adicional sobre el sensor Quad RGB en diferentes entornos y en la lección 5).



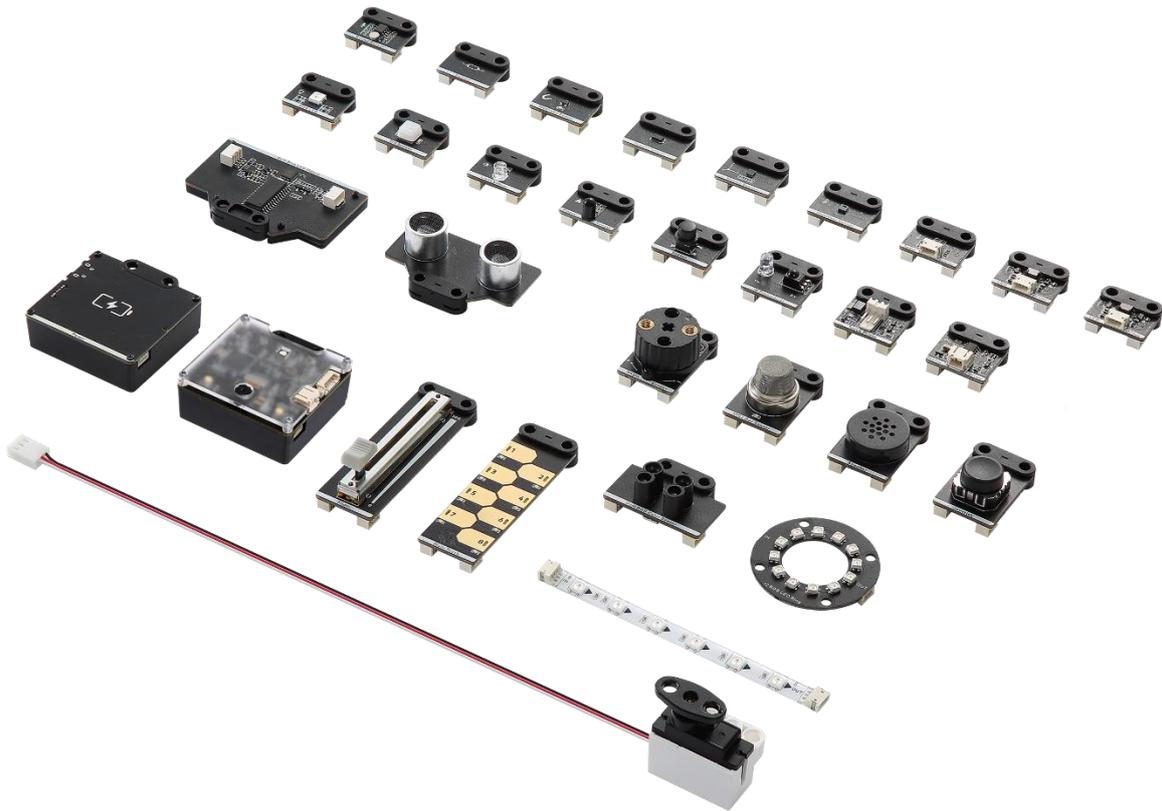
La lección 5 de las Actividades de Introducción ofrece una explicación de los diferentes bloques que están disponibles en mBlock para el sensor Quad RGB, y los estudiantes también pueden poner en práctica sus conocimientos con una tarea sencilla.

## 2.3 Más posibilidades con el mBot2

El alcance de las actividades se puede ampliar aún más combinando el mBot2 con los módulos mBuild desarrollados por Makeblock Education. Gracias a su Unidad de Microcontrolador (MCU) interna, estos sensores y actuadores inteligentes pueden conectarse directamente sin la necesidad de utilizar cableado o configuraciones complejas, permitiendo a los estudiantes pasar más tiempo concibiendo ideas y creando. Estos componentes utilizan un solo tipo de conector que

no se puede invertir, por lo que los estudiantes no pueden cometer errores al conectarlos. Esto les da más confianza y, por lo tanto, una mejor experiencia de aprendizaje.

He aquí ejemplos de algunos sensores y actuadores: cámara inteligente, multitáctil, control deslizante, sensor de temperatura, etc. Con todos estos sensores, los profesores tienen una variedad de opciones para crear escenarios reales y adaptarse a las necesidades de enseñanza, el plan de estudios y a los intereses de los estudiantes.



Pero solo con el mBot2, ya tienes muchas opciones (ver las actividades). Además, la placa del mBot2 ofrece conexiones adicionales: las interfaces integradas de 2 y 3 pines se pueden utilizar para conectar directamente motores de CC, servos, tiras de LED e incluso otros componentes fácilmente disponibles, incluida una amplia gama de sensores compatibles con Arduino, o incluso sensores personalizados.

### 3. Programar el mBot2

mBlock es la plataforma de codificación para el mBot2, diseñada para ofrecer una experiencia educativa mejorada y una trayectoria continua de crecimiento para el estudiante. Gracias a las extensiones de mBlock5, los educadores pueden incluir fácilmente algunas de las tecnologías más recientes e influyentes en sus lecciones, como el Internet de las Cosas o la Inteligencia Artificial. Además, integrando la codificación basada en bloques y Python, mBlock5 ofrece una vía de

aprendizaje para que los estudiantes desarrollen habilidades computacionales, desde básicas a profesionales.

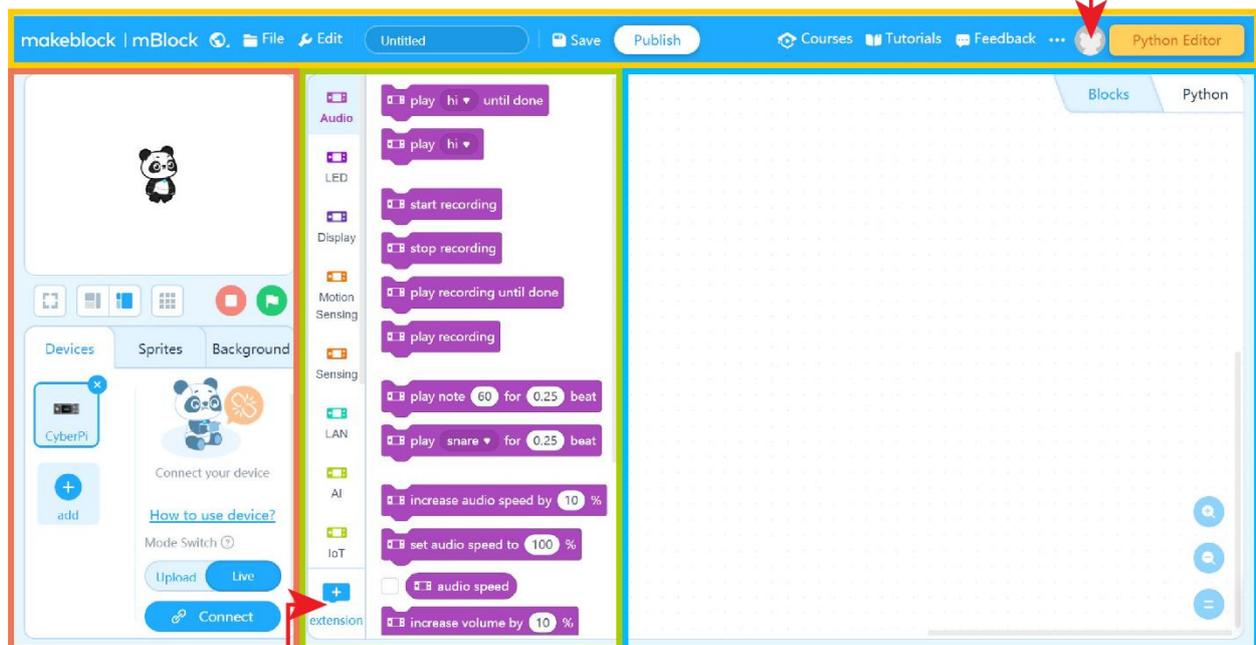
mBlock5 se puede utilizar como *software* instalado en ordenadores, portátiles y dispositivos móviles, o en un navegador web. Es compatible con diferentes sistemas operativos como Windows, Mac, Linux, Chromebook, iOS y Android. Al ser de código abierto, mBlock5 ofrece la oportunidad de crear nuevas extensiones para *software* y *hardware*, permitiendo a los educadores personalizar las herramientas de codificación de acuerdo con sus necesidades. Los usuarios también pueden buscar y compartir proyectos en la comunidad Makeblock.

Empezar a programar con mBlock es tan fácil como arrastrar y soltar bloques. La interfaz mBlock tiene los siguientes elementos:

### Barra de menú

Seleccionar el idioma, abrir o guardar un archivo, buscar programas de ejemplo, etc.

Configura tu cuenta para acceder a los servicios en la nube



### Área de escenario

Ver el escenario del proyecto, personalizar los sprites y los fondos, conectar los dispositivos.

### Área de bloques

Biblioteca de bloques de código clasificados por categorías de colores.

### Área de scripts

Arrastre los bloques de codificación a esta área y organícelos para controlar el escenario de los dispositivos conectados.

Centro de extensiones

Podrás encontrar información detallada sobre las características y el uso de mBlock en <https://education.makeblock.com/help/category/mblock-block-based/>

## 4. Resumen de las actividades

Las Actividades de Introducción del mBot2 están desarrolladas para estudiantes de 11 a 14 años y sus educadores. En general, las lecciones están diseñadas para ser accesibles, a pesar de tener niveles de complejidad diferentes, que van en aumento.

Cada lección presenta una característica diferente y emocionante del mBot2, además de algunos conceptos básicos de programación. La complejidad de las lecciones aumenta a medida que el alumno avanza en su aprendizaje, por lo que el conocimiento adquirido en cada lección ayuda a realizar actividades más avanzadas. Por ejemplo, los movimientos del robot se introducen en la Lección 1 y esto se utiliza en muchas de las lecciones posteriores. Las posibilidades del sensor Quad RGB se presentan en la lección 5, para ser utilizadas posteriormente en la lección 8.

A continuación encontrarás una breve descripción de las lecciones:

Nombre de la actividad	Descripción	Conceptos clave
1. Desplazarse	Los estudiantes descubren el mBot2 y el <i>software</i> mBlock y aprenden a dirigir el robot con precisión. Este conocimiento se utilizará en la mayoría de las siguientes lecciones. Los estudiantes también diseñarán laberintos sencillos y programarán el mBot2 para navegar (manualmente) por ellos.	Movimientos precisos y bloques de código correspondientes.
2. Detección = datos	Los estudiantes trabajarán con los diferentes sensores integrados en el mBot2; aprenderán a utilizarlos con sus bloques de código correspondientes y a visualizar los datos de los sensores en la pantalla integrada a todo color.	Modo de funcionamiento de los sensores. Diferentes enfoques para mostrar y visualizar los datos en la pantalla. Diferencias entre el modo Tiempo Real y el modo Carga en mBlock 5.
3. Escuchar a mBot2	Los estudiantes aprenderán a controlar el altavoz y el micrófono con los bloques de código en mBlock 5. También crearán un programa en el que el mBot2 reproduce un sonido grabado si cumple una determinada condición mientras se desplaza.	Conversor texto-voz (TTS) y Reconocimiento de voz (Speech to Text, STT) mediante el altavoz y el micrófono integrados. Ejecutar múltiples tareas en paralelo.
4. Ver con sonido	Los estudiantes aprenderán qué es el ultrasonido y cómo se utiliza en un sensor. También crearán un programa para hacer que el mBot2 se desplace en un circuito detectando obstáculos en la carretera.	Detectar un obstáculo o una distancia utilizando el sensor ultrasónico. Utilizar circuitos y declaraciones condicionales para hacer que el mBot2 se desplace evitando obstáculos.
5. Hacer turismo	Los estudiantes aprenderán cómo funciona un sensor de color y cómo se usan en la vida real. Programarán el mBot2 para que se convierta en un autobús turístico que visita diferentes puntos en una ciudad. Este conocimiento también se aplicará en las lecciones 7 y 8.	Modo de funcionamiento (física de la luz) del sensor de color/seguidor de línea. Identificación de color y línea: hacer que el mBot2 siga una línea y realice acciones basadas en la detección de colores.

6. Conducción cuidadosa	Los estudiantes aprenderán a utilizar el acelerómetro y el giroscopio del mBot2 y sus bloques de código. Programarán el mBot2 para ajustar su comportamiento de conducción si detecta pendientes en la carretera.	Modo de funcionamiento de giroscopios y acelerómetros (como Unidades de Medida Inercial, IMU). Codificar el mBot2 para adaptarse a las condiciones de la carretera en función de los datos de IMU.
7. Un juego en red	Los estudiantes aprenderán a tener varios mBot2 comunicándose entre sí de forma inalámbrica sin un punto de acceso WIFI. Programarán un juego sencillo en el que varios mBot2 buscan un color y el primero en encontrarlo gana. Este conocimiento también se utilizará en la lección 8.	Transferencia de datos inalámbrica en redes <i>ad-hoc</i> . Intercambio de datos en circuitos y eventos.
8. El mBot2 a tu servicio	Al aprender a configurar una conexión WIFI con el mBot2, los estudiantes también aprenderán a utilizar el reconocimiento de voz de a bordo. Aplicarán estos conocimientos en una actividad en la que el mBot2 se convierte en un robot camarero que habla con sus clientes.	Utilizar del modo de infraestructura WIFI con el mBot2 para el reconocimiento de voz y la síntesis del habla. Descargar la informática pesada, como el reconocimiento de voz, a los servicios en la nube. Estructurar el código aplicando "bloques propios" (funciones).
9. El mBot2 en la naturaleza	En esta lección especial, los estudiantes aprenderán algunos principios de la Inteligencia Artificial utilizando la extensión Teachable Machine de mBlock5. Aplicarán sus conocimientos para recrear un ecosistema natural donde el mBot2 se comporta como un animal.	Aprender sobre el Aprendizaje Automático y aplicarlo con el procesamiento local únicamente en la programación basada en bloques. Establecer un nuevo protocolo de comunicación entre el mBot2 y el ordenador.

A lo largo de las lecciones se motiva a los estudiantes, con el consentimiento de los educadores, a documentar los resultados de su aprendizaje en vídeo y a publicarlos. De esta forma podrán mostrar su trabajo con orgullo y dirigir su conversación sobre Informática y educación STEAM en el aula.

# Lección 1: Desplazarse

**Tema:** STEAM

**Curso(s):** A partir de 5º

**Duración:** 45 minutos

**Dificultad:** Principiante

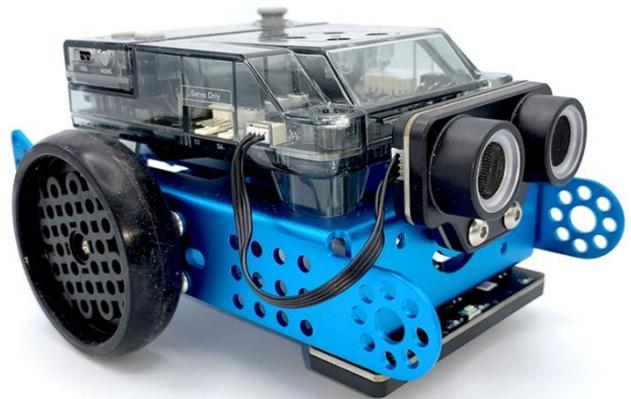
## ★ Objetivos de la lección

*Al final de esta lección, los estudiantes serán capaces de:*

- Dirigir el mBot2 con precisión.

## ★ Resumen

Los robots son máquinas autónomas que reemplazan el esfuerzo humano; pueden percibir su entorno y ejecutar programas informáticos para tomar decisiones y realizar acciones. Un programa informático es un conjunto de instrucciones y condiciones ejecutadas por un ordenador para realizar una tarea. Es posible que pienses que nunca has visto o usado un robot, pero esa posibilidad es mínima. Los robots pueden tomar muchas formas diferentes y tener diferentes capacidades según su contexto. Por ejemplo, los robots están presentes en los hogares (un robot aspirador o un robot cortacésped). También puedes encontrarlos en las fábricas de montaje de productos. Incluso hemos enviado robots a otros planetas.



## 🔗 Puntos de enfoque

*Al final de la lección, los estudiantes conocerán:*

- Qué movimientos puede realizar el mBot2
- Qué diferentes bloques de programación se pueden utilizar para que el mBot2 se mueva

## Lista de verificación previa a la lección

¿Qué necesitas?

- PC u ordenador portátil (con salida USB) con el *software* mBlock instalado, la versión web (también para Chromebook) o una tableta con la aplicación mBlock instalada
- El mBot2 con el CyberPi
- Un cable USB-C o un adaptador (*dongle*) Bluetooth Makeblock
- Papel A3
- Bolígrafos

## Plan de la lección

Duración	Contenido
5 minutos	<b>1. Calentamiento</b> <ul style="list-style-type: none"><li>• Los sensores y los datos en la vida diaria.</li><li>• ¿Qué es el CyberPi?</li></ul>
10 minutos	<b>2. Práctica</b> <ul style="list-style-type: none"><li>• Familiarizarse con los diferentes bloques de programación del mBot2.</li></ul>
25 minutos	<b>3. Prueba</b> <ul style="list-style-type: none"><li>• Programar tu propio robot a través de un laberinto creado por ti.</li></ul>
5 minutos	<b>4. Conclusión</b> <ul style="list-style-type: none"><li>• Hora del espectáculo: muestra lo que has hecho con tu robot en una película corta y divertida para poder comentarlo más adelante.</li><li>• Si tu profesor lo permite, comparte el resultado final en las redes sociales con el hashtag #mBot2inclass</li><li>• Reflexión: ¿De que estas más orgulloso/a? ¿Qué te gustaría mejorar de tu robot?</li></ul>

## 1. Calentamiento (5 min)

### Paso 1: Calentamiento

Este paso consta de dos partes:

1. Robots en la vida diaria
2. Conocer el mBot2

#### 1. Robots en la vida diaria

Tenemos robots en muchos sitios diferentes. Algunas personas los tienen en sus hogares y otros trabajan con ellos todos los días. Los robots pueden tomar muchas formas diferentes y tener diferentes capacidades según su contexto.

¿Puedes mencionar tres robots con los que tengas que lidiar (de forma regular) en la vida diaria?



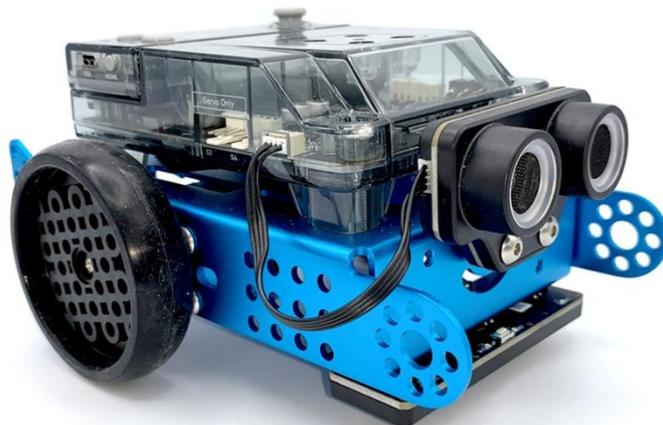
Probablemente se han desarrollado muchos más robots que nunca has visto. Busca en Internet tres robots más que puedas utilizar en la vida diaria.

Es muy probable que tengas que manejar robots más adelante en tu vida profesional o personal. Hay robots que trabajan en muchas empresas diferentes. Por lo general, estos robots realizan tareas repetitivas, que deben realizarse con gran precisión o que son peligrosas. En esta lección trabajaremos principalmente en la programación de un robot para que se mueva con precisión. ¿Puedes pensar en tres profesiones o negocios en los que es muy importante que un robot trabaje con mucha precisión?

## 2. Conocer el mBot2

El mBot2 es un robot programable equipado con diferentes componentes que le permiten sentir, actuar y comunicarse con su entorno. El mBot2 está diseñado para que los estudiantes aprendan sobre Informática y Tecnología, y para que conozcan las formas en que pueden usar un robot en la vida real.

El mBot2 se puede programar con la ayuda de mBlock. En mBlock, la programación puede ser tan fácil como arrastrar y ensamblar bloques. Como verás en esta lección, puedes, por ejemplo, programar el mBot2 para atravesar un laberinto, con tu ayuda (no de forma autónoma). El mBot2 viene con un par de motores especiales, que pueden registrar la rotación del eje y, por lo tanto, la velocidad y la distancia que ha recorrido el robot. Se denominan motores codificadores debido al sensor que llevan incorporado (codificador). Este tipo de motores permite controlar parámetros específicos como el ángulo de rotación y la velocidad de los motores. Aprenderás a utilizar estos motores en los siguientes pasos.



## 2. Práctica (10 min)

### Paso 2: Práctica

Este paso consta de dos partes:

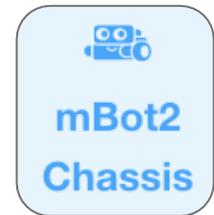
1. Familiarizarse con los diferentes bloques de programación del mBot2.
2. Recrear y probar algunos ejemplos de programación para controlar el mBot2.

### 1. Familiarizarse con los diferentes bloques de programación del mBot2

Como has podido observar anteriormente, los robots se pueden utilizar de muchas formas diferentes. Si has reflexionado sobre el objetivo de un robot, lo más probable es que sepas lo

que el robot necesita para ayudarte. Es posible que algunas de estas tareas necesiten ejecutarse con cierta precisión.

Cuando comiences a programar el mBot2, te darás cuenta de que hay muchos bloques de código diferentes que puedes utilizar para hacer que el robot se mueva. Encontrarás estos bloques en la categoría "Chásis" de mBlock 5. Estos bloques de código son azules.



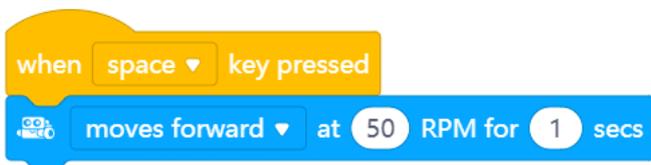
A continuación puedes ver algunos ejemplos de los bloques de código necesarios para hacer que el robot se mueva. Algunos de ellos parecen similares, pero todos hacen algo diferente. En esta lección, trabajaremos en modo Tiempo Real. Asegúrate de que se ha seleccionado el modo correcto en el *software*. Las diferencias entre el modo Tiempo Real y el modo Carga se explicarán en la lección 2.

#### Bloque de códigos:



Este bloque de códigos te permite mover el mBot2 hacia adelante, hacia atrás, hacia la izquierda y hacia la derecha a una velocidad de rotación específica de las ruedas y durante varios segundos.

El siguiente ejemplo muestra cómo puedes hacer que el mBot2 se mueva durante dos segundos a una velocidad de 50 rotaciones por minuto. Esto es útil, por ejemplo, cuando el mBot2 tiene que empujar una carga hacia adelante.

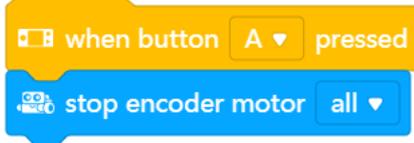


#### Bloque de códigos:

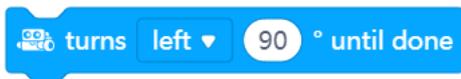


Este bloque de códigos te permite hacer que tu robot se detenga. Este puede ser un bloque útil durante una fase de prueba. Si un programa no funciona como se esperaba, puedes utilizar este bloque para hacer que el robot se pare inmediatamente.

Por ejemplo, puedes hacer que el robot se detenga cuando se presiona el botón A del CyberPi. Para hacer esto, usa el ejemplo de programación siguiente.

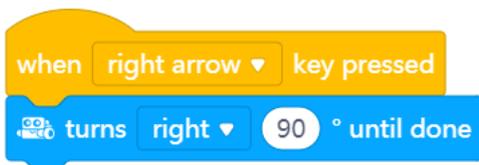
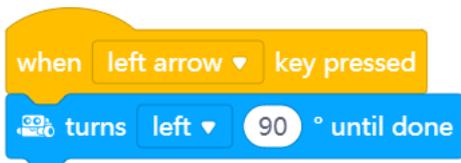


### Bloque de códigos:



Con este bloque de códigos puedes hacer que el mBot2 gire varios grados y puedes elegir el sentido de la rotación; izquierda o derecha.

Cuando se establece el ejemplo de programación siguiente, se puede controlar el mBot2 usando las teclas de dirección. Al presionar la flecha derecha, el mBot2 girará 90 grados a la derecha, al presionar la flecha izquierda, girará 90 grados a la izquierda.

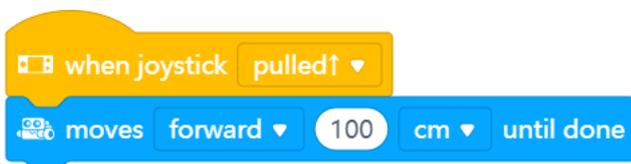


### Bloque de códigos:



Con este bloque de códigos puedes hacer que el mBot2 se mueva hacia adelante o hacia atrás una cierta distancia.

Cuando utilizas el ejemplo de programación siguiente, el mBot2 se moverá 100 cm hacia adelante. Para comenzar, mueve la palanca de mando del CyberPi hacia arriba.



### Bloque de códigos:

encoder motor EM1 rotates at 50 RPM, encoder motor EM2 rotates at 50 RPM

También puedes controlar los motores del mBot2 de forma independiente, y este es uno de los bloques que puedes usar para eso.

En el siguiente ejemplo, el mBot2 se moverá en una curva durante 3 segundos antes de detenerse. Observa que el valor de uno de los motores es negativo. Esto se debe a que los motores están montados en direcciones opuestas, por lo que para mover el robot en una dirección, tienes que hacer que uno de los motores gire en la dirección opuesta. ¿Qué pasaría si ambas ruedas girasen a 40 RPM?

when button B pressed  
encoder motor EM1 rotates at 50 RPM, encoder motor EM2 rotates at -30 RPM  
wait 3 seconds  
stop encoder motor all

### Bloque de códigos:

encoder motor (1) EM1 rotated angle (°)

Los motores de mBot2 pueden medir su velocidad y rotación, por lo que también pueden leer estos valores. La velocidad medida es la velocidad de rotación en Rotaciones por Minuto ( $360^\circ = 1$  rotación; en ciencias, es más común utilizar "grados por segundo" ( $1 \text{ RPM} = 6^\circ$  por segundo)). En mBlock 5, si marcas la casilla de este bloque, podrás leer los valores en el escenario encima del panda. El siguiente ejemplo de programación devuelve los valores de la rotación a cero y a continuación ordena al mBot2 que avance durante un segundo. ¿Qué valor lees en el escenario después de que el robot se haya movido?

when joystick middle pressed  
reset encoder motor all rotated angle  
moves forward at 50 RPM for 1 secs



### 2. Recrear y probar algunos ejemplos de programación para controlar el mBot2.

A continuación podrás ver pares de bloques. Prueba cada par en paralelo. ¿Qué ocurre?

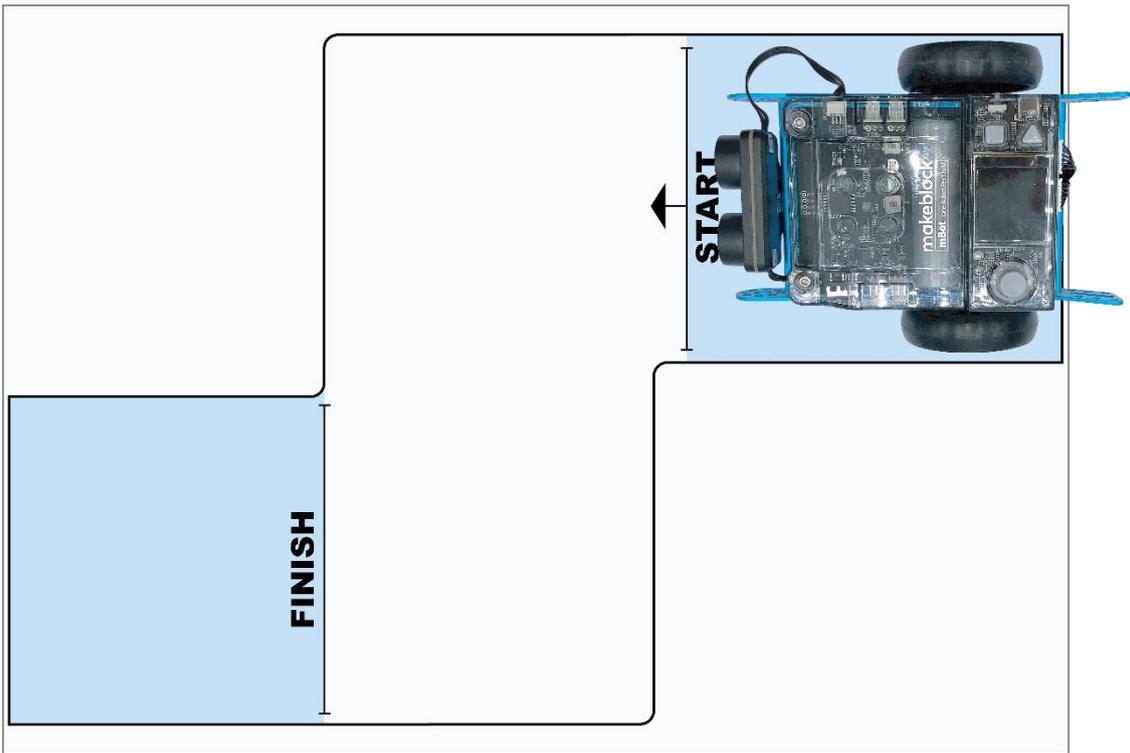
¿Cuáles son las diferencias entre los bloques de código?

	Opción 1	Opción 2
1		
2		
3		

## 3. Prueba (25 min)

### Paso 3: Prueba

Es hora de que tu mBot2 se ponga a trabajar. Coge un papel A3 y dibuja un laberinto por el que tu robot debe pasar. No hagas un dibujo complicado y ten en cuenta el ancho del robot. El mBot2 no tiene que encontrar el camino correcto por sí solo. El mBot2 solo necesita recorrer el camino que has programado. ¿Necesitas inspiración? A continuación encontrarás un ejemplo.



Utiliza los conocimientos adquiridos en el "Paso 2" de esta lección. Por supuesto, puedes experimentar más por ti mismo con los diferentes ejemplos de programación que encontrarás en mBlock5.

Al realizar esta tarea, es útil seguir el siguiente esquema paso a paso.

	Explicación
Paso 1: ¿Qué quieres hacer?	<ul style="list-style-type: none"> <li>• ¿Qué camino quieres que siga el mBot2?</li> <li>• ¿De qué partes consta el laberinto?</li> <li>• ¿Qué distancias debe recorrer el robot?</li> <li>• ¿El robot tiene que girar? ¿Hacia qué lado y cuántos grados?</li> </ul>
Paso 2: ¿Qué necesitas?	<ul style="list-style-type: none"> <li>• ¿Qué necesitas además del mBot2?</li> </ul>
Paso 3: ¿Qué bloques de código necesitas para hacer que el mBot2 se desplace?	<ul style="list-style-type: none"> <li>• ¿Qué vas a hacer para que el mBot2 se desplace?</li> <li>• ¿Qué bloques de código usarás?</li> <li>• Describe brevemente cómo funciona tu programa (pseudocódigo/lenguaje natural, diagrama de flujo o Lenguaje Unificado de Modelado (UML))</li> <li>• Si necesitas más explicaciones, puedes hablar con tus compañeros de clase, con el profesor o investigar sobre el tema. También tienes ayuda disponible para cada bloque de códigos en mBlock.</li> </ul>
Paso 4: Pruebas e implementación	<ul style="list-style-type: none"> <li>• ¿Está lista la primera versión? ¡Pruébala! Durante la ronda de pruebas, anota las áreas de mejora.</li> <li>• Trabaja en los puntos de mejora hasta que tu mBot2 atraviese tu laberinto sin errores.</li> </ul>

¿Es posible hacer que el mBot2 atraviese el laberinto sin problemas? A continuación, ponte a prueba con un laberinto más difícil. O pídele a uno de tus compañeros de clase que te prepare un recorrido.

## 4. Conclusión (5 min)

### Paso 4: Conclusión

¿Has conseguido desplazar el robot por el laberinto sin errores?

En esta lección se han presentado los robots en la vida diaria y se ha presentado también el mBot2. Sabes cómo controlar los movimientos del mBot2 y qué bloques de programación puedes utilizar para ello.

Ha llegado el momento de realizar una breve reflexión. Piensa por ti mismo y discute con el grupo:

- ¿Qué crees que ha salido bien?
- ¿Qué podría mejorarse?
- ¿Qué partes de la lección te han resultado fáciles y cuáles más difíciles?
- ¿Sobre qué te gustaría recibir más información?
- ¿Quién podría ayudarte con eso?

## Lección 2: Detección = datos

**Tema:** STEAM

**Curso(s):** A partir de 5º

**Duración:** 45 minutos

**Dificultad:** Principiante

### ★ Objetivos de la lección

Al final de esta lección, los estudiantes serán capaces de:

- Reconocer y utilizar los bloques de código para dirigir los sensores
- Mostrar los datos de los sensores en la pantalla del mBot2
- Crear su propio programa informático en mBlock para el mBot2 para controlar los sensores y mostrar los datos en la pantalla



### ★ Resumen

Un robot puede ser un juguete divertido, pero ¿sabías que también puedes usar un robot en la investigación? Un robot tiene varios sensores (= sentidos) con los que puede percibir su entorno y también recopilar datos. Por ejemplo, puedes hacer que el mBot2 mida el sonido y la temperatura. Puedes leer los datos en la pantalla del mBot2 y utilizarlos para investigar la densidad del sonido o la temperatura del aula, por ejemplo.

### 🔗 Puntos de enfoque

Al final de esta lección, los estudiantes sabrán:

- Qué sensores tiene el mBot2
- Cómo funcionan los sensores del mBot2
- Cuál es la diferencia entre el modo Tiempo Real y el modo Carga
- Cómo mostrar los datos de los sensores en la pantalla del CyberPi

## Lista de verificación previa a la lección

¿Qué necesitas?

- PC u ordenador portátil (con salida USB) con el *software* mBlock instalado, la versión web (también para Chromebook) o una tableta con la aplicación mBlock instalada
- El mBot2 con el CyberPi
- Un cable USB-C o un adaptador (*dongle*) Bluetooth Makeblock

## Plan de la lección

Esta lección consta de cuatro pasos y tiene una duración de 45 minutos.

Duración	Contenido
5 minutos	<b>1. Calentamiento</b> <ul style="list-style-type: none"><li>• Los sensores y los datos en la vida diaria</li><li>• ¿Qué es el CyberPi?</li></ul>
10 minutos	<b>2. Práctica</b> <ul style="list-style-type: none"><li>• Familiarizarse con los sensores del mBot2.</li><li>• Ampliar y probar algunos ejemplos de programación de los sensores.</li><li>• Mostrar los datos de los sensores en la pantalla.</li><li>• Diferenciar entre el modo Tiempo Real y el modo Carga.</li></ul>
25 minutos	<b>3. Prueba</b> <ul style="list-style-type: none"><li>• Escribir tu propio programa para el robot.</li></ul>
5 minutos	<b>4. Conclusión</b> <ul style="list-style-type: none"><li>• Hora del espectáculo: muestra lo que has hecho con tu robot en una película corta y divertida para poder comentarlo más adelante.</li><li>• Si tu profesor lo permite, comparte el resultado final en las redes sociales con el hashtag #mBot2inclass</li><li>• Reflexión: ¿De que estas más orgulloso/a? ¿Qué te gustaría mejorar de tu robot?</li></ul>

## 1. Calentamiento (5 min)

### Paso 1: Calentamiento

Este paso consta de dos partes:

1. Los sensores y los datos en la vida diaria
2. ¿Qué es el CyberPi?

#### 1. Los sensores y los datos en la vida diaria

Los sensores que recopilan datos se encuentran en muchos lugares diferentes de la vida diaria. Mucho más de lo que probablemente piensas en un principio. Por ejemplo, cuando oscurece, la iluminación exterior de tu casa se enciende automáticamente con la ayuda de sensores. O la calefacción se apaga automáticamente cuando hace demasiado calor en el aula. La cantidad de luz exterior y el calor en el aula se registran mediante sensores. Tú y tus compañeros de clase, ¿podéis dar otros ejemplos?

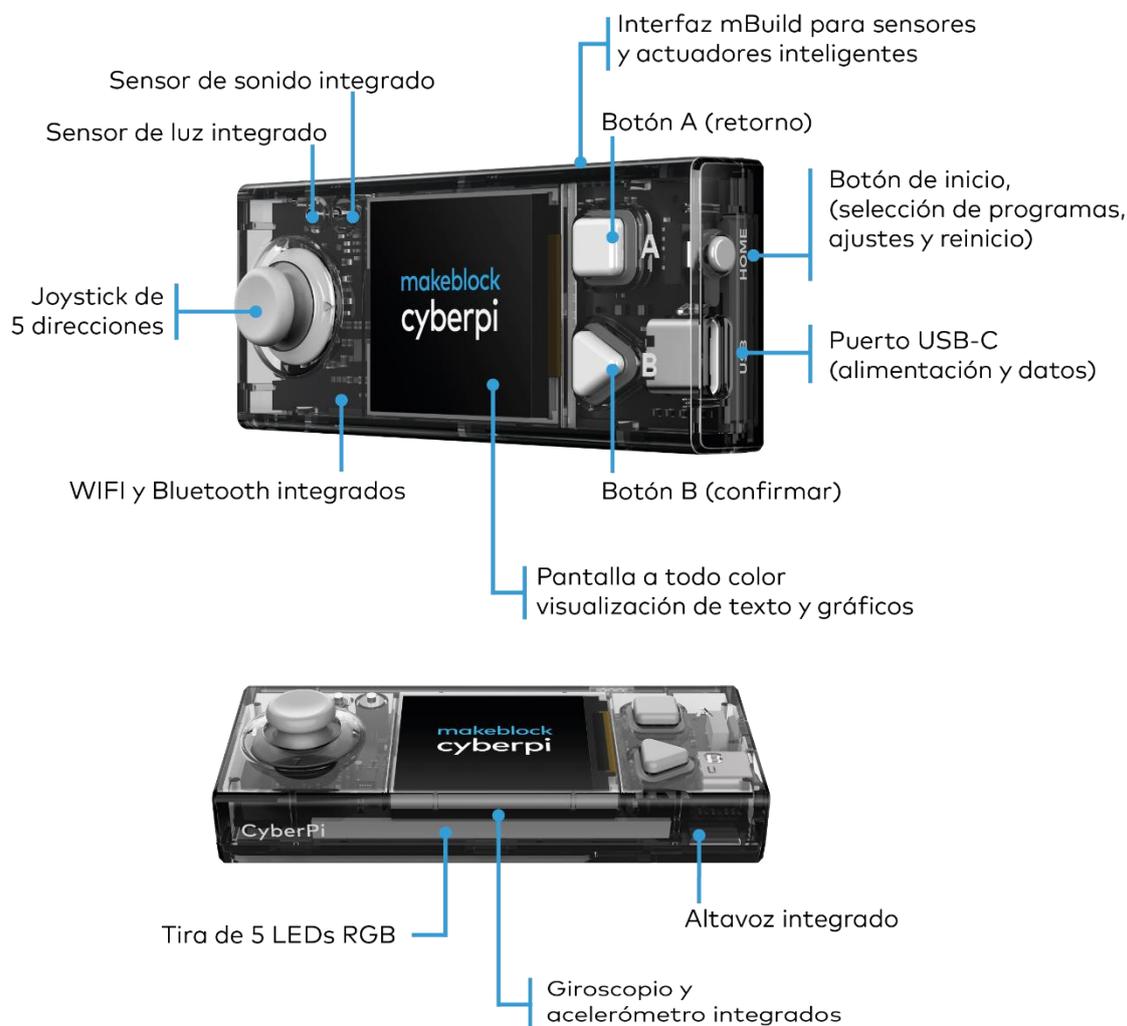
#### 2. ¿Qué es el CyberPi?

Para que el mBot2 funcione, debes escribir un programa informático. Lo podrás hacer en el editor de código mBlock. El programa informático que debes escribir consta de una serie de órdenes que el mBot2 debe ejecutar.



Hay un CyberPi montado en el mBot2. Un CyberPi es un pequeño microordenador programable. Darás órdenes al CyberPi usando los bloques de código de mBlock. El CyberPi devuelve estas órdenes al mBot2. Puedes retirar el CyberPi del mBot2.

El CyberPi tiene muchas funciones diferentes, como un micrófono, un altavoz y una palanca de mando. También hay muchos sensores en el CyberPi. Observa la siguiente imagen.



## 2. Práctica (10 min)

### Paso 2: Práctica

Este paso consta de cuatro partes:

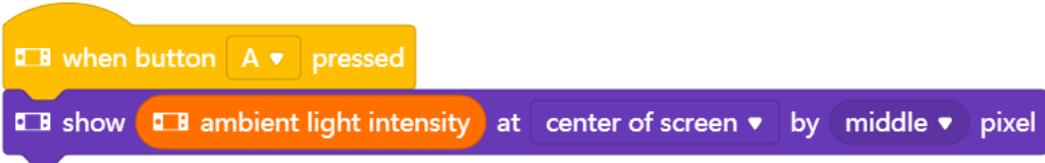
1. Familiarizarse con los sensores del mBot2.
2. Ampliar y probar algunos ejemplos de programación de los sensores.
3. Mostrar los datos de los sensores.
4. Diferenciar entre el modo Tiempo Real y el modo Carga.

## 1. Conocer los sensores del mBot2

Todos los robots funcionan con sensores. Los sensores pueden compararse a tus sentidos (=gusto, tacto, olfato, oído, vista). El mBot2 "ve" su entorno a través de estos sensores. Hay diferentes tipos de sensores que permiten "ver" al mBot2, como por ejemplo:

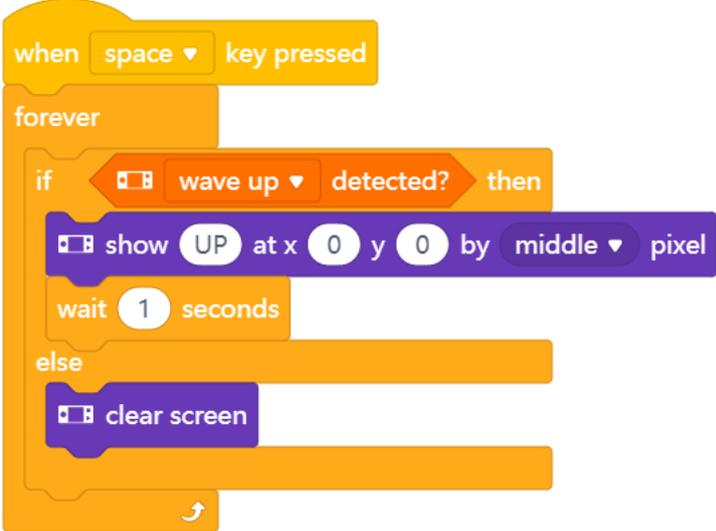
- Sensor de luz
- Sensor de sonido
- Giroscopio y acelerómetro
- Sensor Quad RGB
- Temporizador

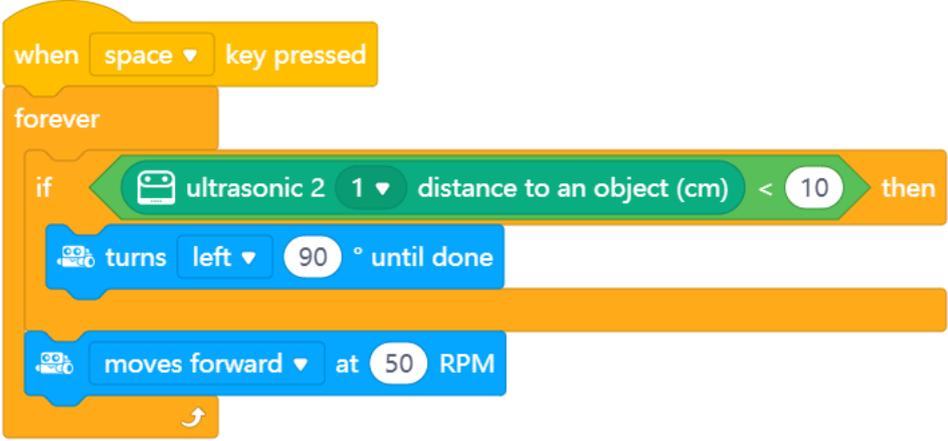
En el cuadro siguiente se enumeran los sensores del mBot2. Cada sensor incluye una breve explicación y un ejemplo de programación.

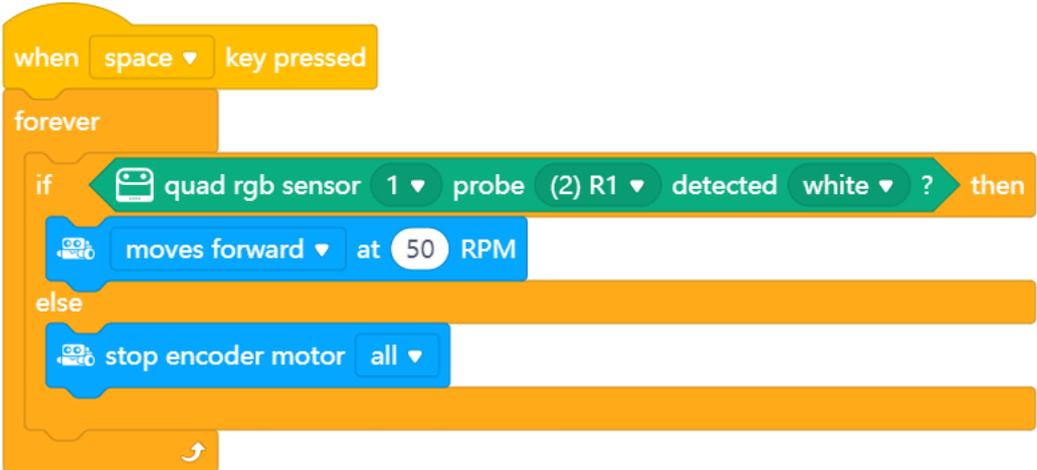
Tipo de sensor	¿Qué hace este sensor?
Sensor de luz	<p>La luz tiene cierta fuerza (= energía luminosa). El sensor de luz es un dispositivo que convierte la energía luminosa en energía eléctrica.</p> <p>Un sensor de luz se utiliza a menudo, por ejemplo para:</p> <ul style="list-style-type: none"> <li>• Ajustar automáticamente el brillo de la pantalla de los <i>smartphones</i> teniendo en cuenta el entorno;</li> <li>• Controlar la iluminación de los hogares o encender automáticamente los faros de los vehículos.</li> </ul> <p>En el siguiente ejemplo de programación, la energía luminosa del entorno se muestra en la pantalla del CyberPi. Se puede ampliar el ejemplo de programación para que el mBot2 deje de desplazarse, por ejemplo, cuando la energía luminosa del entorno esté por debajo de cierto valor, como cuando oscurece.</p>
	<b>Ejemplo de programación</b>
	 <pre> when button A pressed   show ambient light intensity at center of screen by middle pixel           </pre>



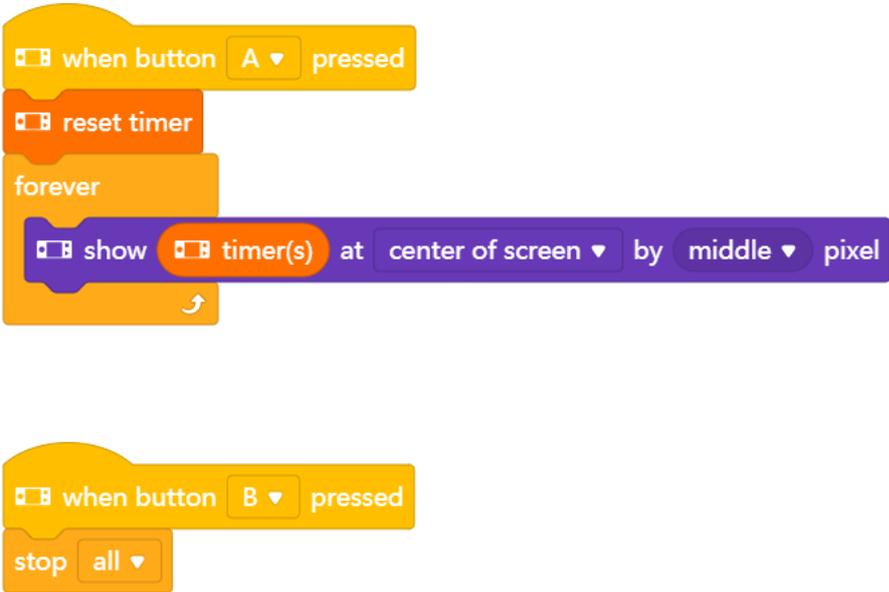
Tipo de sensor	¿Qué hace este sensor?
Sensor de sonido	<p>El sonido es una vibración mecánica que se propaga mediante ondas. En el aire, son fluctuaciones de presión y densidad. Si la vibración está en el alcance audible (entre 16 y 20 000 vibraciones por segundo) y es lo suficientemente intensa, podemos escucharla como tonos o sonidos. El poder del sonido se llama intensidad del sonido.</p> <p>El sensor del micrófono convierte el sonido en una señal eléctrica que se puede evaluar en términos de tono y potencia.</p> <p>Un sensor de sonido se usa a menudo para:</p> <ul style="list-style-type: none"><li>• Llamadas telefónicas, grabación de sonido</li><li>• Sistemas de asistencia controlados por voz, como en un teléfono móvil o en un <i>o en una Casa inteligente</i></li></ul> <p>En el siguiente ejemplo de programación, la intensidad del sonido del entorno se muestra en la pantalla del CyberPi. Puedes ampliar el ejemplo de programación para que el mBot2 mida constantemente la intensidad del sonido durante un recorrido por el aula y lo muestre en la pantalla.</p>
	<p style="text-align: center;"><b>Ejemplo de programación</b></p>
	 <pre>when button A pressed show loudness at center of screen by middle pixel</pre>

Tipo de sensor	¿Qué hace este sensor?
<p>Giroscopio y acelerómetro</p>	<p>Un giroscopio mide los movimientos de inclinación, más precisamente la velocidad de los movimientos de giro/inclinación. Un acelerómetro mide el cambio de velocidad. Estos sensores se pueden implementar como componentes micromecánicos en un componente electrónico. Ambos sensores proporcionan información diferente sobre la posición en el espacio.</p> <p>Un giroscopio se usa a menudo para, por ejemplo:</p> <ul style="list-style-type: none"> <li>• Mantener los barcos estables en alta mar, o</li> <li>• Mantener el equilibrio de un Segway o un "hoverboard" para que no te caigas demasiado rápido utilizándolo</li> </ul> <p>Los sensores de aceleración, a su vez, indican si un <i>smartphone</i> está colocado hacia arriba o hacia abajo, o si un vehículo está sufriendo un accidente (cambio violento y muy rápido de velocidad) para activar los <i>airbag</i>.</p> <p>En el siguiente ejemplo de programación, se muestra en la pantalla el movimiento de inclinación del mBot2. Puedes ampliar el ejemplo de programación para que el mBot2 mida y registre constantemente el movimiento de inclinación en la pantalla durante un recorrido por el aula. Aprenderás más sobre este sensor y sus capacidades en la Lección 6.</p>
	<p><b>Ejemplo de programación</b></p>
	 <pre> when space key pressed   forever     if wave up detected? then       show UP at x 0 y 0 by middle pixel       wait 1 seconds     else       clear screen   </pre>

Tipo de sensor	¿Qué hace este sensor?
<p>Sensor ultrasónico</p>	<p>El sonido, que es una vibración en forma de variaciones de densidad y presión, se propaga en el aire a una velocidad constante en forma de ondas (la velocidad del sonido en el aire es aproximadamente de 334 m/s). Cuanta más alta es la vibración, más alto aparece el sonido (hasta un límite, por encima del cual las personas ya no pueden percibir este sonido). Estas vibraciones muy elevadas se denominan ultrasonidos. Dado que se conoce la velocidad del sonido en el aire, el sonido se puede utilizar para calcular las distancias a objetos. Para ello, se emite un sonido y se mide cuánto tarda el sonido en reflejarse en el objeto. Esta onda de sonido reflejada también se llama eco. Los ultrasonidos se utilizan generalmente para este objetivo.</p> <p>Por ejemplo, un sensor ultrasónico se utiliza a menudo:</p> <ul style="list-style-type: none"> <li>• en los procedimientos de formación de imágenes, p. ej. en el embarazo (cada píxel es una medida de distancia), o</li> <li>• en el control de robots para prevenir colisiones</li> </ul> <p>En el siguiente ejemplo de programación, el sensor ultrasónico se utiliza para evitar que el mBot2 choque contra un obstáculo. Cuando el mBot2 está a menos de 10 cm de un obstáculo, el robot gira 90° a la izquierda y luego simplemente continúa desplazándose. Puedes ampliar este ejemplo de programación para que el mBot2 se mueva aleatoriamente por un aula sin chocar con mesas y sillas.</p> <p>Aprenderás más sobre este sensor y sus capacidades en la Lección 4.</p>
	<p><b>Ejemplo de programación</b></p>
	 <pre> when space key pressed   forever loop     if ultrasonic 2 1 distance to an object (cm) &lt; 10 then       turns left 90° until done       moves forward at 50 RPM   </pre>

Tipo de sensor	¿Qué hace este sensor?
<p>Sensor Quad RGB</p>	<p>El sensor Quad RGB consta de cuatro sensores de luz y color individuales. Miden la intensidad de la luz que entra en el sensor desde áreas rojas, verdes y azules del espectro de luz. Esto permite al sensor detectar colores de objetos que se encuentran delante de él, como marcas en el suelo, y también permite que el robot siga una línea negra para orientarse.</p> <p>Un sensor Quad RGB se usa ampliamente en aplicaciones como:</p> <ul style="list-style-type: none"> <li>• mostrar a un robot de almacén un recorrido a través del almacén (se utilizan también diferentes colores para diferentes recorridos),</li> <li>• asegurarse de que se utiliza exactamente el color elegido en pintura</li> </ul> <p>En el siguiente ejemplo de programación, el mBot2 avanza cuando el robot ve el color blanco. Puedes ampliar el ejemplo de programación para que el mBot2 realice un recorrido a través del aula, se detenga o gire en función de diferentes colores.</p> <p>Aprenderás más sobre este sensor y sus capacidades en la Lección 5.</p>
	<p><b>Ejemplo de programación</b></p>
	 <pre> when space key pressed   forever     if quad rgb sensor 1 probe (2) R1 detected white ? then       moves forward at 50 RPM     else       stop encoder motor all   </pre>

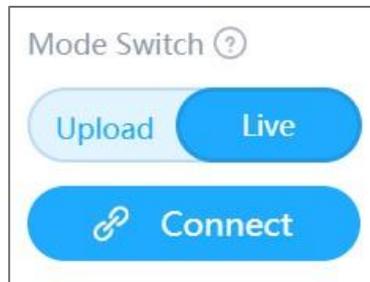


Tipo de sensor	¿Qué hace este sensor?
Temporizador	<p>El temporizador es una especie de cronómetro que indica el tiempo en segundos desde que se ha encendido o reiniciado el CyberPi. Este contador también se puede poner a cero mediante un comando para facilitar las mediciones de tiempo.</p> <p>Por ejemplo, puedes usar el temporizador para hacer una carrera entre varios mBots2 y medir exactamente su velocidad en la línea de meta. El siguiente ejemplo de programación te mostrará cómo configurar y usar el temporizador. Pulsando la tecla "A", el temporizador se pone a cero y luego se muestra de forma permanente en la pantalla. Puedes detener la pantalla y el programa, en su conjunto, pulsando la tecla "B". Intenta ampliar el ejemplo de programación para que el temporizador comience a contar cuando el mBot2 empieza a moverse.</p>
<b>Ejemplo de programación</b>	
	

## 2. Probar y ampliar ejemplos de programación para los sensores

En el cuadro anterior, hay un ejemplo de programación para cada sensor. Re crea estos ejemplos de programación en mBlock y pruébalos. Piensa en una ampliación para dos de los ejemplos de programación. El cuadro anterior ya menciona una sugerencia de ampliación para cada ejemplo de programación. ¡Quizás conozcas una ampliación mucho mejor!

A medida que escribes un programa informático, puedes probar inmediatamente lo que estás creando. Hazlo usando el modo Tiempo Real. Selecciona el modo Tiempo Real moviendo el interruptor de modo hacia la derecha. Observa la siguiente imagen.



### 3. Mostrar los datos de los sensores

Al probar los ejemplos de programación, has visto que cada sensor muestra algo en la pantalla del CyberPi. Por ejemplo, el sensor de luz registra la intensidad de la luz del entorno. El temporizador registra el tiempo y el sensor giroscópico controla en qué dirección se inclina el mBot2. Todo lo que registra un sensor se conoce como datos.

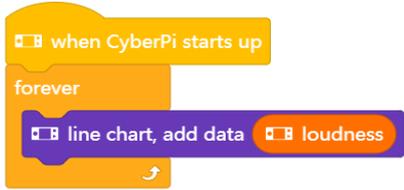
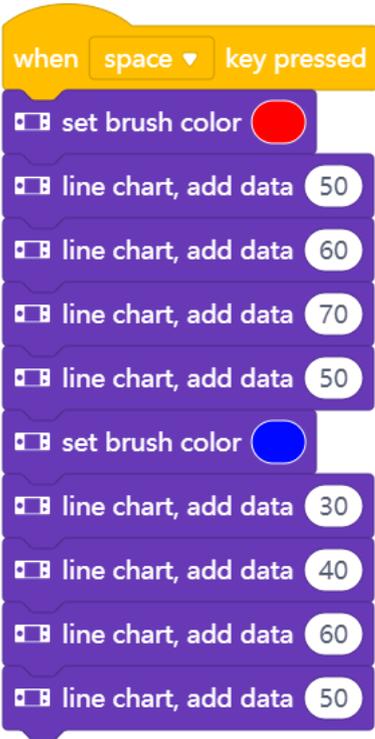
Puedes mostrar los datos de un sensor en la pantalla del mBot2. En los ejemplos de programación, has visto que puedes mostrar los datos, por ejemplo, con un número o un texto.

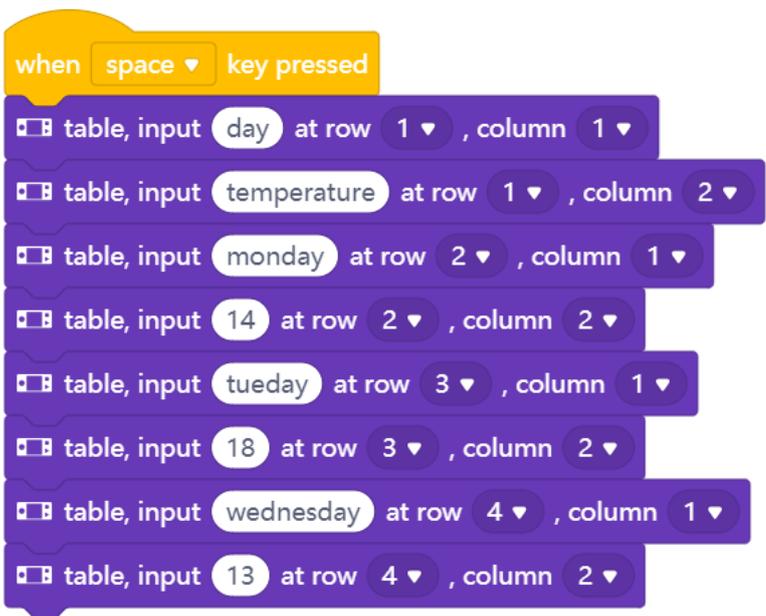




Puedes mostrar los datos de los sensores de diferentes formas en la pantalla del mBot2. Puedes utilizar diferentes bloques de código para ello. Estos bloques de código se pueden encontrar en la categoría 'Mostrar' de mBlock.

En el siguiente diagrama, podrás ver tres ejemplos de cómo mostrar datos en la pantalla de CyberPi. Reproduce los ejemplos de programación y pruébalos en modo Tiempo Real. Observa lo que sucede en la pantalla del mBot2.

Ejemplo de programación	
Diagrama lineal	
Diagrama lineal (color)	

Cuadro	 <pre> when space key pressed   table, input day at row 1, column 1   table, input temperature at row 1, column 2   table, input monday at row 2, column 1   table, input 14 at row 2, column 2   table, input tuesday at row 3, column 1   table, input 18 at row 3, column 2   table, input wednesday at row 4, column 1   table, input 13 at row 4, column 2           </pre>
--------	--

#### 4. Modo Tiempo Real y modo Carga

Tienes que transferir al CyberPi el programa informático que creaste en mBlock.

CyberPi pone a trabajar al mBot2 teniendo en cuenta los comandos que escribiste en mBlock.

A medida que escribes un programa informático, puede probar directamente lo que estás haciendo. Hazlo usando el modo Tiempo Real. Selecciona el modo Tiempo Real moviendo el interruptor de modo hacia la derecha. ¡No desconectes el CyberPi del ordenador mientras realizas pruebas en modo Tiempo Real!

El CyberPi también tiene un modo Carga. En el modo Carga, el programa informático se transfiere al CyberPi. El programa se almacena en el CyberPi hasta que lo reemplaces por otro programa. Simplemente puedes desconectar el CyberPi.

En el siguiente diagrama podrás ver las diferencias entre el modo Tiempo Real y el modo Carga.



Modo	Diferencias explicadas
Modo Tiempo Real	<ul style="list-style-type: none"><li>• Los programas se ejecutan en tu ordenador (= tu PC u ordenador portátil). No se almacenarán en CyberPi.</li><li>• El CyberPi debe permanecer conectado a tu ordenador.</li><li>• El editor de código mBlock siempre debe estar abierto en tu ordenador.</li><li>• Este modo se utiliza para programar en el escenario.</li></ul>
Modo Carga	<ul style="list-style-type: none"><li>• Los programas se ejecutan en el CyberPi en lugar de en tu ordenador y también se almacenan en el CyberPi.</li><li>• No es necesario que el CyberPi esté conectado a tu ordenador.</li><li>• No es necesario que el editor de código mBlock esté abierto en tu ordenador.</li><li>• Un programa cargado en CyberPi se guarda en CyberPi hasta que lo reemplaces por otro programa.</li></ul>

### 3. Prueba (25 min)

#### Paso 3: Prueba

Ya has aprendido mucho sobre los sensores del mBot2 y cómo mostrar los datos de los sensores en la pantalla del CyberPi. Ahora vas a recopilar datos por ti mismo utilizando los sensores del mBot2 y vas a mostrarlos en la pantalla del CyberPi. Lo harás mientras desplazas el mBot2 por el aula o la escuela. Tú decides qué datos recopilar y cómo mostrarlos en la pantalla del CyberPi.

Utiliza los conocimientos adquiridos en el "Paso 2" de esta lección. Por supuesto, puedes experimentar todo lo que quieras por ti mismo con los diferentes ejemplos de programación que encontrarás en mBlock.

Al analizar esta tarea, es útil seguir el siguiente esquema paso a paso. ¿Tienes idea de lo que quieres hacer? Si es así, coméntalo primero con tu profesor para ver si es factible.

	Explicación
Paso 1: ¿Qué quieres hacer?	<ul style="list-style-type: none"> <li>• ¿Qué camino quieres que siga el mBot2?</li> <li>• ¿Qué datos quieres que recopile el mBot2?</li> </ul>
Paso 2: ¿Qué necesitas?	<ul style="list-style-type: none"> <li>• ¿Qué necesitas además del mBot2?</li> </ul>
Paso 3: ¿Qué bloques de código necesitas para hacer que el mBot2 se desplace?	<ul style="list-style-type: none"> <li>• ¿Qué vas a hacer para que el mBot2 se desplace?</li> <li>• ¿Qué datos mandarás recopilar al mBot2?</li> <li>• ¿Qué bloques de código usarás?</li> <li>• Describe brevemente cómo funciona tu programa (pseudocódigo/lenguaje natural, diagrama de flujo o Lenguaje Unificado de Modelado (UML))</li> <li>• Si necesitas más explicaciones, puedes hablar con sus compañeros de clase, el profesor o investigar sobre el tema. También tienes ayuda disponible para cada bloque de códigos en mBlock.</li> </ul>
Paso 4: ¿De qué manera quieres mostrar los datos de los sensores en la pantalla?	<ul style="list-style-type: none"> <li>• ¿Cómo quieres mostrar los datos en la pantalla?</li> <li>• ¿Qué bloques de código usarás?</li> <li>• Describe brevemente cómo funciona tu programa (pseudocódigo/lenguaje natural, diagrama de flujo o Lenguaje Unificado de Modelado (UML))</li> <li>• Si necesitas más explicaciones, puedes hablar con sus compañeros de clase, el profesor o investigar sobre el tema. También tienes ayuda disponible para cada bloque de códigos en mBlock.</li> </ul>
Paso 5: Pruebas e implementación	<ul style="list-style-type: none"> <li>• ¿Está lista la primera versión? ¡Pruébala! Durante la ronda de pruebas, anota las áreas de mejora.</li> <li>• Trabaja en los puntos de mejora hasta que el mBot2 haga exactamente lo que tenías en mente.</li> <li>• ¿Lo has conseguido? Graba el resultado final y pregúntale a tu profesor si puedes publicarlo en las redes sociales con el hashtag #mBot2inclass</li> </ul>

## 4. Conclusión (5 min)

### Paso 4: Conclusión

¿Qué datos ha recopilado tu mBot2? ¿Ha salido bien? ¿Has conseguido ver todos los datos en la pantalla del CyberPi?

En esta lección, has aprendido sobre los diferentes sensores del mBot2 y dónde puedes encontrarlos en la vida diaria. Sabes cómo programar estos sensores y cómo mostrar los datos en la pantalla. También has aprendido cuáles son las diferencias entre los modos Tiempo Real y Cargar Programa de mBlock.

Ha llegado el momento de realizar una breve reflexión. Piensa por ti mismo y discute con el grupo:

- ¿Qué crees que ha salido bien?
- ¿Qué podría mejorarse?
- ¿Qué partes de la lección te han resultado fáciles y cuáles más difíciles?
- ¿Sobre qué te gustaría recibir más información?
- ¿Quién podría ayudarte con eso?

## Lección 3: Escuchar a mBot2

**Tema:** STEAM

**Curso(s):** A partir de 5º

**Duración:** 45 minutos

**Dificultad:** Principiante

### ★ Objetivos de la lección

*Al final de esta lección, los estudiantes serán capaces de:*

- Reconocer y utilizar los bloques de código para controlar el altavoz y el micrófono
- Crear su propio programa informático en mBlock para hacer que el mBot2 grabe sonidos y los reproduzca
- Ejecutar múltiples tareas de programación en paralelo en un programa informático

### ★ Resumen

A menudo, los robots pueden hablar. Tomemos como ejemplo el robot aspirador que usa un mensaje de voz para avisarte que tiene poca batería. O un robot de juguete que canta una canción. El mBot2 también puede producir y grabar sonidos. Para ello, tiene un micrófono y un altavoz. Con el micrófono, el mBot2 graba sonidos y el altavoz reproduce el sonido grabado.

### 🔗 Puntos de enfoque

*Al final de esta lección, los estudiantes sabrán:*

- Cómo utilizar el altavoz en codificación y robótica
- Cómo grabar y reproducir sonidos
- Cómo ejecutar múltiples tareas de programación en paralelo

## Lista de verificación previa a la lección

¿Qué necesitas?

- PC u ordenador portátil (con salida USB) con el *software* mBlock instalado, la versión web (también para Chromebook) o una tableta con la aplicación mBlock instalada
- El mBot2 con el CyberPi
- Un cable USB-C o un adaptador (*dongle*) Bluetooth Makeblock

## Plan de la lección

Esta lección consta de cuatro pasos y tiene una duración de 45 minutos.

Duración	Contenido
5 minutos	<b>1. Calentamiento</b> <ul style="list-style-type: none"><li>• Altavoces en la vida diaria</li><li>• ¿Cómo funciona el altavoz?</li></ul>
10 minutos	<b>2. Práctica</b> <ul style="list-style-type: none"><li>• Familiarizarse con los diferentes bloques de código del altavoz y el micrófono.</li><li>• Reproduce, prueba y amplía algunos ejemplos de programación del altavoz y el micrófono.</li></ul>
25 minutos	<b>3. Prueba</b> <ul style="list-style-type: none"><li>• Escribir tu propio programa para el robot.</li></ul>
5 minutos	<b>4. Conclusión</b> <ul style="list-style-type: none"><li>• Hora del espectáculo: muestra lo que has hecho con tu robot en una película corta y divertida para poder comentarlo más adelante.</li><li>• Si tu profesor lo permite, comparte el resultado final en las redes sociales con el hashtag #mBot2inclass</li><li>• Reflexión: ¿De que estas más orgulloso/a? ¿Qué te gustaría mejorar de tu robot?</li></ul>

## ☰ Actividades

# 1. Calentamiento (5 min)

### Paso 1: Calentamiento

Este paso consta de dos partes:

1. Altavoces en la vida diaria
2. ¿Cómo funciona el altavoz?

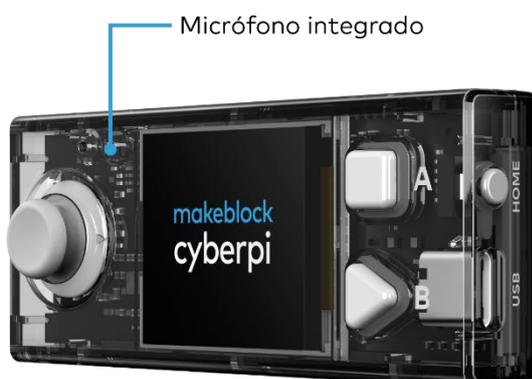
#### 1. Altavoces en la vida diaria

Los altavoces se utilizan a tu alrededor en la vida diaria. Encontrarás grandes sistemas de altavoces en teatros, salas de conciertos y salas de eventos. Estos altavoces se utilizan para amplificar el sonido de un gran grupo de personas. Encontrarás sistemas de altavoces más pequeños en diferentes aparatos, como televisores, instrumentos musicales y *smartphones*. Tú y tus compañeros de clase de tu grupo, ¿podéis mencionar otras aplicaciones?

#### 2. ¿Cómo funciona el altavoz?

El objetivo de un altavoz es producir sonido. Los altavoces convierten las ondas electromagnéticas en ondas de sonido para conseguir este objetivo. Además de su objetivo principal, un altavoz también puede convertir las ondas sonoras en ondas electromagnéticas. Los micrófonos son dispositivos que utilizan el mismo principio, pero están optimizados para convertir ondas de sonido en señales eléctricas.

En el mBot2 hay un altavoz y un micrófono. El micrófono está en la parte superior izquierda del CyberPi. El altavoz está en la parte frontal del CyberPi. Con el micrófono puedes grabar sonidos. Estos sonidos se pueden reproducir más tarde con el altavoz.



Utilizando el micrófono y el altavoz junto con los otros sensores del mBot2, puedes, por ejemplo, reproducir tu propio texto grabado en un momento específico. Por ejemplo, piensa en un autobús público que te informa de la parada a la que vas a llegar. Existen otras innumerables posibilidades para usar el micrófono y el altavoz. Junto con los compañeros de clase de tu grupo, piensa en más ejemplos en los que el mBot2 puede utilizar el altavoz y el micrófono.

## 2. Práctica (10 min)

### Paso 2: Práctica

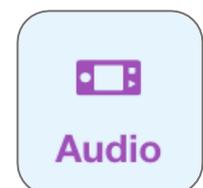
Este paso consta de dos partes:

1. Familiarizarse con los diferentes bloques de código del altavoz y el micrófono.
2. Reproducir, probar y ampliar algunos ejemplos de programación del altavoz.

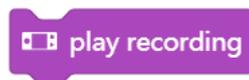
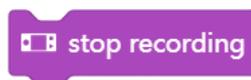
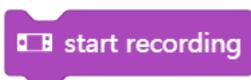
### 1. Conocer los diferentes bloques de código del altavoz y el micrófono

En mBlock 5, hay varios bloques de código que puedes utilizar para programar el altavoz y el micrófono. Encontrarás estos bloques de código en la categoría "Audio" del área de bloques en mBlock. Estos bloques de código son morados.

En el siguiente cuadro verás algunos de estos bloques de código para programar el altavoz y el micrófono.



#### Bloque de códigos:



Al grabar y reproducir un sonido, en realidad siempre usas tres bloques de código. Con el bloque de código "Iniciar grabación" haces que el CyberPi empiece a grabar un sonido. La señal eléctrica del micrófono se almacena en la memoria del CyberPi y la grabación se detiene automáticamente después de 10 segundos. También puedes indicar cuándo quieres detener la grabación. Esto se hace con el bloque de código "Detener grabación". El sonido grabado se almacena temporalmente en la memoria del CyberPi. ¿Qué ocurre si apagas el mBot2? El sonido grabado se perderá. La reproducción del sonido grabado se controla mediante el bloque de código "Reproducir grabación".

En el siguiente ejemplo de programación, cuando pulsas el botón A, el CyberPi comienza a grabar. Después de 3 segundos, la grabación se detiene. Cuando presionas el botón B, escucha lo que se ha grabado. Utilizando dos bloques de eventos, se pueden ejecutar dos tareas de programación independientemente una de la otra sin la necesidad de un circuito principal.

```

when button A pressed
  start recording
  wait 3 seconds
  stop recording

when button B pressed
  play recording
  
```

### Bloque de códigos:

```

play recording until done
  
```

Acabas de aprender a reproducir un sonido grabado. Con este bloque de código, te aseguras de que el sonido grabado por el micrófono se reproduce hasta el final. Esto significa que la ejecución de más bloques se bloquea hasta que este haya finalizado. Una vez finalizado el sonido, puedes hacer que el mBot2 ejecute otra acción. Por ejemplo, hacer un determinado sonido para saber que la grabación ha finalizado.

En el ejemplo de programación de la derecha, el CyberPi empieza a grabar cuando pulsas el botón A. Si pulsas el botón B, la grabación se detiene. Si mueves la palanca de mando hacia abajo, escucharás lo que el mBot2 ha grabado. Después de terminar la grabación completa, escucharás un sonido de "aviso". Los bloques de código de este ejemplo se encuentran en una estructura denominada de circuito. Esto garantiza que la introducción de datos de los botones y la palanca de mando está controlada de forma continua.

```

when space key pressed
  forever
    if button A pressed? then
      start recording
    if button A pressed? then
      stop recording
    if joystick pulled down? then
      play recording until done
      play prompt
  
```

## 2. Reproducir, probar y ampliar algunos ejemplos de programación del altavoz

En el cuadro anterior, cada bloque de código del altavoz va acompañado de un ejemplo de programación. Recrea estos ejemplos de programación en mBlock y pruébalos. Piensa en una ampliación de un ejemplo de programación. Prueba los ejemplos de programación y tu propia ampliación en modo Tiempo Real.

## 3. Prueba (25 min)

### Paso 3: Prueba

Ya has aprendido mucho sobre el altavoz y el micrófono del mBot2. Ahora vas a trabajar con ellos por ti mismo. También vas a utilizar el sensor ultrasónico. Este sensor puede detectar un objeto a distancia. Aprenderás más sobre el sensor ultrasónico en la Lección 4.

En esta tarea, harás que el mBot2 dé una vuelta aleatoria por el aula. Aleatorio significa que no predeterminas un recorrido. Si el mBot2 se acerca a un obstáculo, como una silla, mesa o mochila, escucharás un sonido y el mBot2 se dará la vuelta. Grabarás el sonido con el micrófono. El sonido podría ser, por ejemplo, un pitido fuerte o el sonido de un coche disminuyendo de velocidad. También puedes grabar un texto o hacer que el mBot2 reproduzca una canción.

¡Es bastante difícil crear un programa informático para realizar estas acciones por ti solo! Afortunadamente, puedes obtener ayuda. En la página siguiente podrás consultar dos ejemplos de un programa informático. Puedes ampliar y adaptar estos programas informáticos por ti mismo. ¡Sé creativo!

### Ejemplo 1

El mBot2 emite un sonido cuando se acerca a un obstáculo y se da la vuelta. Antes, has grabado el sonido tú mismo.

```

when button A pressed
  start recording
  wait 3 seconds
  stop recording

when button B pressed
  forever
    if ultrasonic 2 1 distance to an object (cm) < 50 then
      play recording until done
      encoder motor all rotates by 90 degrees
    else
      moves forward at 50 RPM
  
```

### Ejemplo 2

El mBot2 se desplaza y, cuando la luz ambiental es baja, reproduce un sonido que grabaste anteriormente. Cuando el mBot2 se acerca a un obstáculo, emite un sonido de advertencia y se da la vuelta.

```

when button A pressed
  start recording
  wait 3 seconds
  stop recording

when button B pressed
  forever
    if ultrasonic 2 1 distance to an object (cm) < 50 then
      play warning until done
      encoder motor all rotates by 90 degrees
    else
      moves forward at 50 RPM
      if ambient light intensity < 40 then
        play recording until done
  
```



Utiliza los conocimientos adquiridos en el "Paso 2" de esta lección. Por supuesto, puedes experimentar todo lo que quieras por ti mismo con los diferentes ejemplos de programación que encontrarás en mBlock.

Al analizar esta tarea, es útil seguir el siguiente esquema paso a paso. ¿Tienes idea de lo que quieres hacer? Si es así, coméntalo primero con tu profesor para ver si es factible.

	<b>Explicación</b>
Paso 1: ¿Qué quieres hacer?	<ul style="list-style-type: none"><li>• ¿Cuándo debería darse la vuelta el mBot2?</li><li>• ¿Qué sonido debería emitir el mBot2 cuando se acerca a un obstáculo?</li></ul>
Paso 2: ¿Qué necesitas?	<ul style="list-style-type: none"><li>• ¿Qué necesitas además del mBot2?</li></ul>
Paso 3: ¿Qué bloques de código necesitas para hacer que el mBot2 se desplace?	<ul style="list-style-type: none"><li>• ¿Qué vas a hacer para que el mBot2 se desplace?</li><li>• ¿Qué bloques de código usarás?</li><li>• Describe brevemente cómo funciona tu programa (pseudocódigo/lenguaje natural, diagrama de flujo o Lenguaje Unificado de Modelado (UML))</li><li>• Si necesitas más explicaciones, puedes hablar con sus compañeros de clase, el profesor o investigar sobre el tema. También tienes ayuda disponible para cada bloque de códigos en mBlock.</li></ul>
Paso 4: ¿De qué manera quieres mostrar los datos de los sensores en la pantalla	<ul style="list-style-type: none"><li>• ¿Qué sonido debería hacer el mBot2?</li><li>• ¿Qué bloques de código usarás?</li><li>• Describe brevemente cómo funciona tu programa (pseudocódigo/lenguaje natural, diagrama de flujo o Lenguaje Unificado de Modelado (UML))</li><li>• Si necesitas más explicaciones, puedes hablar con sus compañeros de clase, el profesor o investigar sobre el tema. También tienes ayuda disponible para cada bloque de códigos en mBlock.</li></ul>

Paso 5: Pruebas e implementación

- ¿Está lista la primera versión? ¡Pruébala! Durante la ronda de pruebas, anota las áreas de mejora.
- Trabaja en los puntos de mejora hasta que tu mBot2 haga exactamente lo que tenías en mente.
- ¿Lo has conseguido? Graba el resultado final y pregúntale a tu profesor si puedes publicarlo en las redes sociales con el hashtag #mBot2inclass

## 4. Conclusión (5 min)

### Paso 4: Conclusión

¿Cómo ha ido esta tarea? ¿El mBot2 ha realizado inmediatamente lo que tenías en mente?

En esta lección, has aprendido cómo funcionan los altavoces y dónde puedes encontrarlos en la vida diaria. Sabes cómo programar el CyberPi y el mBot para que realicen grabaciones utilizando el micrófono incorporado y reproduzcan los sonidos a través de los altavoces.

También sabes cómo ejecutar varios programas de forma independiente entre sí.

Ha llegado el momento de realizar una breve reflexión. Piensa por ti mismo y discute con el grupo:

- ¿Qué crees que ha salido bien?
- ¿Qué podría mejorarse?
- ¿Qué partes de la lección te han resultado fáciles y cuáles más difíciles?
- ¿Sobre qué te gustaría recibir más información?
- ¿Quién podría ayudarte con eso?

# Lección 4:

## Ver con sonido

**Tema:** STEAM

**Curso(s):** A partir de 5º

**Duración:** 45 minutos

**Dificultad:** Principiante

### ★ Objetivos de la lección

*Al final de esta lección, los estudiantes sabrán:*

- Qué es el ultrasonido y cómo funciona un sensor basado en ultrasonido
- Aplicaciones de estos sensores en la vida diaria y en robótica

### ★ Resumen

Los robots autónomos deben ser conscientes de su entorno cuando se desplazan sin supervisión para evitar colisiones con obstáculos o personas. Esto también se aplica al mBot2. Para ello, el mBot2 tiene un sensor específico en la parte delantera. Se llama sensor ultrasónico y permite al robot detectar objetos en su camino utilizando sonidos que los humanos no pueden oír, de ahí el nombre ultrasonido.

### 🔗 Puntos de enfoque

*Al final de esta lección, los estudiantes serán capaces de:*

- Utilizar un sensor ultrasónico para la detección de obstáculos y distancia
- Hacer que el mBot2 reaccione ante los obstáculos y evitarlos al desplazarse

## Lista de verificación previa a la lección

¿Qué necesitas?

- PC u ordenador portátil (con salida USB) con el *software* mBlock instalado, la versión web (también para Chromebook) o una tableta con la aplicación mBlock instalada
- El mBot2 con el CyberPi
- Un cable USB-C o un adaptador (*dongle*) Bluetooth Makeblock
- Obstáculos pequeños (el tamaño mínimo es de 10 x 10 x 10 cm)
- Un espacio de trabajo de al menos 1m<sup>2</sup> para que el mBot2 pueda desplazarse

## Plan de la lección

Esta lección consta de cuatro pasos y tiene una duración de 45 minutos.

Duración	Contenido
5 minutos	<b>1. Calentamiento</b> <ul style="list-style-type: none"> <li>• Sensores ultrasónicos en la vida diaria</li> <li>• ¿Cómo funciona un sensor ultrasónico?</li> </ul>
10 minutos	<b>2. Práctica</b> <ul style="list-style-type: none"> <li>• Conocer los diferentes bloques de código del sensor ultrasónico.</li> <li>• Recrear y probar algunos ejemplos de programación del sensor ultrasónico.</li> </ul>
25 minutos	<b>3. Prueba</b> <ul style="list-style-type: none"> <li>• Escribir tu propio programa para el robot.</li> </ul>
5 minutos	<b>4. Conclusión</b> <ul style="list-style-type: none"> <li>• Hora del espectáculo: muestra lo que has hecho con tu robot en una película corta y divertida para poder comentarlo más adelante.</li> <li>• Si tu profesor lo permite, comparte el resultado final en las redes sociales con el hashtag #mBot2inclass</li> <li>• Reflexión: ¿De que estas más orgulloso/a? ¿Qué te gustaría mejorar de tu robot?</li> </ul>

### 1. Calentamiento (5 min)

#### Paso 1: Calentamiento

Este paso consta de dos partes:

1. Sensores ultrasónicos en la vida diaria
2. ¿Cómo funciona el sensor ultrasónico?

#### 1. Sensores ultrasónicos en la vida diaria

El ultrasonido es un sonido con una frecuencia muy alta. Tan alta que la gente no puede oírlo. Con micrófonos especiales, este sonido se puede grabar y mostrar en un ordenador: parecen vibraciones en forma de onda. Dado que el ultrasonido tiene una frecuencia muy alta, la forma de la onda se representa con valles y picos muy juntos. El ultrasonido puede ser útil. Por ejemplo, puedes utilizar las vibraciones del ultrasonido para limpiar objetos, o también para detectar objetos.

Es así como los delfines y los murciélagos usan el ultrasonido. Emiten un sonido ultrasónico, como un grito corto. Cuando este sonido rebota en algo, captan con sus oídos las vibraciones (el eco) que vienen de vuelta. Así es como saben dónde están otros animales, por ejemplo, o si hay algún obstáculo en su camino. El radar que tenemos hoy utiliza el mismo principio, pero con ondas de radio en lugar de sonido.

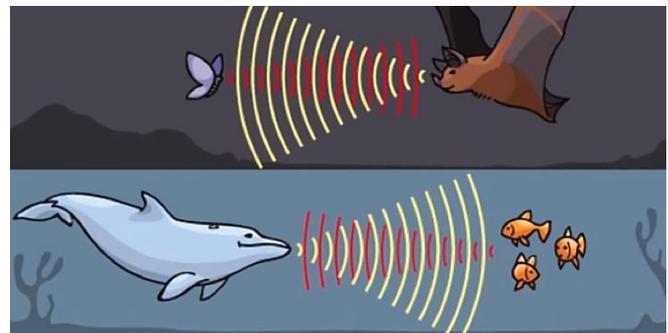


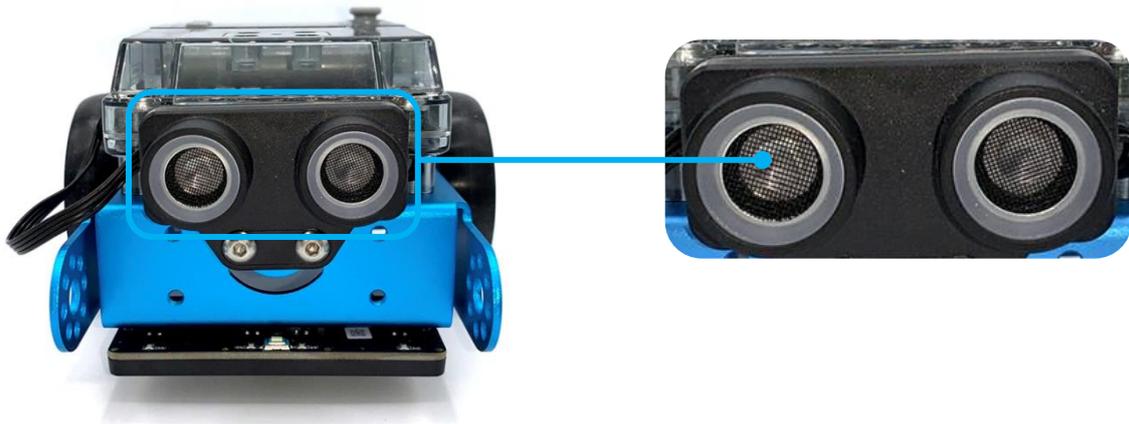
Figura 2. Ejemplo del murciélago y el delfín

#### 2. ¿Cómo funciona el sensor ultrasónico?

El mBot2 tiene un transmisor y un receptor ultrasónicos. Se encuentran en la parte frontal del mBot2, en los pequeños cilindros que se pueden considerar como "ojos". Este sensor envía breves ráfagas de sonidos ultrasónicos y escucha si hay eco. ¿Qué ocurre cuando el mBot2 se acerca a un obstáculo? El sonido ultrasónico viene de vuelta al mBot2. Según el tiempo que

tarda el sonido en volver al sensor, el mBot2 calcula internamente la distancia hasta dicho obstáculo.

Estos datos se pueden utilizar para decidir la reacción del robot: se lo comunicas al robot programándolo. Un ejemplo de acción podría ser que el mBot2 deje de desplazarse o se dé la vuelta.



## 2. Práctica (10 min)

### Paso 2: Práctica

Este paso consta de dos partes:

1. Familiarizarse con los diferentes bloques de código del sensor ultrasónico
2. Recrear y probar algunos ejemplos de programación del sensor ultrasónico.

#### 1. Familiarizarse con los diferentes bloques de código del sensor ultrasónico

En mBlock, hay varios bloques de código para el sensor ultrasónico que puedes utilizar en tus programas. Para ello, debes añadir la extensión "Sensor ultrasónico 2" de la biblioteca de extensiones. Encontrarás estos bloques de código en la categoría "Sensor ultrasónico" del área de bloques en mBlock. Estos bloques de código son verdes.



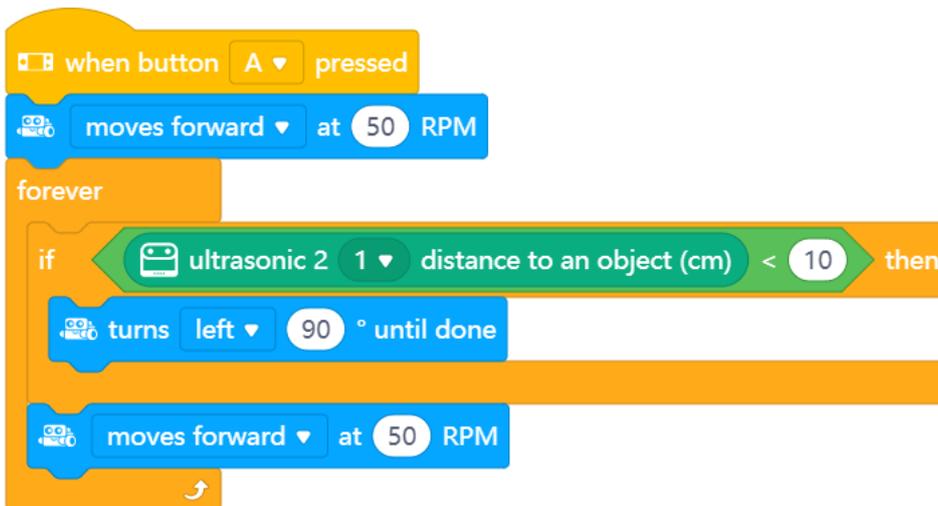
En el siguiente cuadro verás algunos ejemplos de los bloques de código para programar el sensor ultrasónico.

## Bloque de códigos:

 ultrasonic 2 1 ▼ distance to an object (cm)

Con este bloque de código se mide la distancia entre el sensor y un obstáculo. El alcance de detección es de entre 3 y 300 cm. Puedes utilizar el valor de la distancia para hacer que el robot realice una determinada acción. Por ejemplo, puedes evitar que el mBot2 choque con los obstáculos.

En el siguiente ejemplo de programación, el mBot2 avanza. Cuando el mBot2 está a menos de 10 cm de un obstáculo, el robot gira 90° a la izquierda. Después, el mBot2 avanza de nuevo.

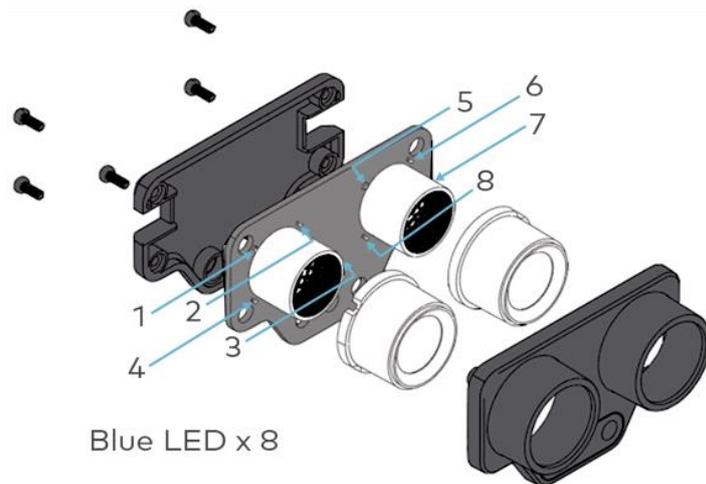


```
when button A pressed
  moves forward at 50 RPM
  forever
    if ultrasonic 2 1 distance to an object (cm) < 10 then
      turns left 90 ° until done
      moves forward at 50 RPM
```

**Bloque de códigos:**



El sensor ultrasónico de mBot2 tiene ocho luces LED azules. Puedes utilizar estas luces para que el mBot2 muestre una determinada "emoción", por ejemplo, como una comunicación visual no verbal general. Haz que las luces LED brillen mucho cuando el mBot2 esté feliz y que brillen menos cuando esté triste. Hay un bloque de código específico para mostrar "emociones" predefinidas. En la siguiente imagen podrás ver dónde están ubicadas exactamente las luces LED.



En mBlock hay varios bloques de código que te permitirán programar el brillo de las luces LED: puedes configurar el brillo de los LED disminuyendo o aumentando el brillo en un cierto porcentaje. O puedes establecer el brillo directamente en un porcentaje específico. El rango es de 0 % a 100 %. Ambas acciones se pueden ejecutar para un solo LED o para todos al mismo tiempo. En el siguiente ejemplo de programación, el brillo de la lámpara LED 1 aumenta en un 50 %.





## 2. Recrear y probar algunos ejemplos de programación del sensor ultrasónico

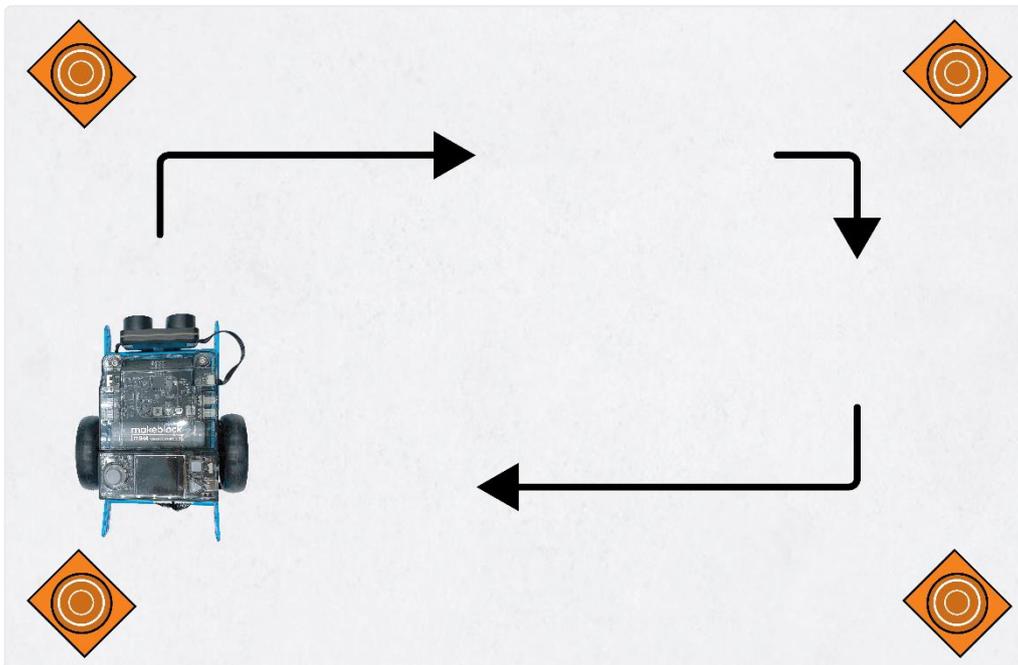
En el cuadro anterior, cada bloque de código del sensor ultrasónico va acompañado de un ejemplo de programación. Recrea estos ejemplos de programación en mBlock y pruébalos. Elabora tus propias ideas sobre cómo extender la aplicación de un ejemplo de programación.

### 3. Prueba (25 min)

#### Paso 3: Prueba

Ya has aprendido mucho sobre el sensor ultrasónico del mBot2. Ahora vas a crear tu propio programa informático en mBlock utilizando el sensor ultrasónico.

En esta tarea, harás que el mBot2 realice "círculos" de forma infinita. Para ello, coloca un objeto en cada vértice. Programa el mBot2 de tal manera que haga un giro en cada objeto y regrese al punto de partida. Observa la siguiente imagen.



Utiliza los conocimientos adquiridos en el "Paso 2" de esta lección. Por supuesto, puedes experimentar con los diferentes ejemplos de programación que encontrarás en mBlock. Al realizar esta tarea, es útil seguir el siguiente esquema paso a paso. ¿Tienes idea de lo que quieres hacer? Si es así, coméntalo primero con tu profesor para ver si es factible.

	Explicación
Paso 1: ¿Qué quieres hacer?	<ul style="list-style-type: none"> <li>¿De qué forma debería moverse el mBot2?</li> </ul>
Paso 2: ¿Qué necesitas?	<ul style="list-style-type: none"> <li>¿Qué necesitas además del mBot2?</li> </ul>
Paso 3: ¿Qué bloques de código necesitas para hacer que el mBot2 se desplace?	<ul style="list-style-type: none"> <li>¿Qué debes hacer para que el mBot2 cambie de dirección en cada esquina?</li> <li>¿Qué bloques de código necesitas para realizar esta acción?</li> </ul>



	<ul style="list-style-type: none"><li>• Describe brevemente cómo funciona tu programa (pseudocódigo/lenguaje natural, diagrama de flujo o Lenguaje Unificado de Modelado (UML))</li><li>• Si necesitas más explicaciones, puedes hablar con sus compañeros de clase, el profesor o investigar sobre el tema. También tienes ayuda disponible para cada bloque de códigos en mBlock</li></ul>
Paso 4: Pruebas e implementación	<ul style="list-style-type: none"><li>• ¿Está lista la primera versión? ¡Pruébala! Durante la ronda de pruebas, anota los puntos de mejora</li><li>• Trabaja en los puntos de mejora hasta que tu mBot2 se desplace sin errores</li><li>• ¿Lo has conseguido? Graba el resultado final y pregúntale a tu profesor si puedes publicarlo en las redes sociales con el hashtag #mBot2inclass</li></ul>

## 4. Conclusión (5 min)

### Paso 4: Conclusión

¿Has conseguido que el mBot2 diera una vuelta?

En esta lección, has aprendido sobre los sensores ultrasónicos y dónde puedes encontrarlos en la vida diaria. Sabes cómo programar el sensor ultrasónico del mBot2, por ejemplo, para evitar que el mBot2 choque contra obstáculos. También sabes cómo usar las luces LED en el sensor ultrasónico para hacer que el mBot2 muestre cierta emoción.

Ha llegado el momento de realizar una breve reflexión. Piensa por ti mismo y discute con el grupo:

- ¿Qué crees que ha salido bien?
- ¿Qué podría mejorarse?
- ¿Qué partes de la lección te han resultado fáciles y cuáles más difíciles?
- ¿Sobre qué te gustaría recibir más información?
- ¿Quién podría ayudarte con eso?

## Lección 5: Hacer turismo

**Tema:** STEAM

**Curso(s):** A partir de 5º

**Duración:** 45 minutos

**Dificultad:** Principiante

### ★ Objetivos de la lección

*Al final de esta lección, los estudiantes sabrán:*

- Cómo funcionan los sensores de color
- Su aplicación en la vida diaria y en robótica

### ★ Resumen

Imagínate que estás en una ciudad extranjera, realizando un recorrido en autobús por los lugares emblemáticos. Puede que ya hayas hecho este viaje, y si no, es posible que conozcas las imágenes gracias a la televisión o a Internet. En esta lección, el mBot2 se transformará en un autobús turístico.

Ya que sería muy aburrido para un conductor conducir la misma ruta todos los días (y explicar cada lugar emblemático una y otra vez), vamos a programar este autobús. De esta manera, las personas pueden disfrutar de un agradable viaje sin que el conductor se aburra realizando tareas repetitivas.

### 🔗 Puntos de enfoque

*Al final de esta lección, los estudiantes serán capaces de:*

- Hacer que el mBot2 siga una línea
- Hacer que el mBot2 ejecute acciones basadas en colores

## Lista de verificación previa a la lección

¿Qué necesitas?

- PC u ordenador portátil (con salida USB) con el *software* mBlock instalado, la versión web (también para ChromeBook) o una tableta con la aplicación mBlock instalada
- El mBot2 con el CyberPi
- Un cable USB-C o un adaptador (*dongle*) Bluetooth Makeblock

## Plan de la lección

Esta lección consta de cuatro pasos y tiene una duración de 45 minutos.

Duración	Contenido
5 minutos	<b>1. Calentamiento</b> <ul style="list-style-type: none"><li>• Sensores de color en la vida diaria</li><li>• ¿Cómo funcionan los sensores de color? ¿Qué tiene de específico el sensor de color del mBot2?</li></ul>
10 minutos	<b>2. Práctica</b> <ul style="list-style-type: none"><li>• Familiarizarse con los diferentes bloques de código del sensor Quad RGB</li><li>• Recrear y probar ejemplos de programación del sensor Quad RGB ultrasónico</li><li>• ¿Qué hacer para que el mBot2 siga una línea?</li></ul>
25 minutos	<b>3. Prueba</b> <ul style="list-style-type: none"><li>• Escribir tu propio programa para el robot</li></ul>
5 minutos	<b>4. Conclusión</b> <ul style="list-style-type: none"><li>• Hora del espectáculo: muestra lo que has hecho con tu robot en una película corta y divertida para poder comentarlo más adelante</li><li>• Si tu profesor lo permite, comparte el resultado final en las redes sociales con el hashtag #mBot2inclass</li><li>• Reflexión: ¿De que estas más orgulloso/a? ¿Qué te gustaría mejorar de tu robot?</li></ul>

## 1. Calentamiento (5 min)

### Paso 1: Calentamiento

Este paso consta de dos partes:

1. Sensores de color en la vida diaria
2. ¿Cómo funcionan los sensores de color? ¿Qué tiene de específico el sensor de color del mBot2?

#### 1. Sensores de color en la vida diaria

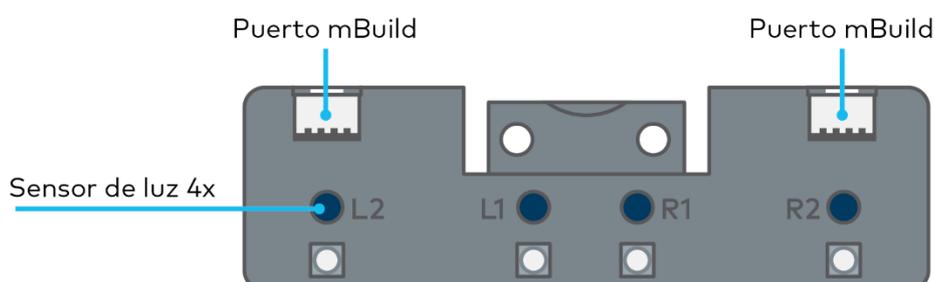
Los sensores de color pueden detectar la intensidad de la luz y los colores. Encontrarás este tipo de sensor en muchos lugares diferentes de la vida diaria.

Por ejemplo, un robot de almacén puede encontrar la ruta correcta a través de un almacén basándose en diferentes colores en el suelo. Este tipo de sensor también se utiliza si se requiere una combinación de colores precisa, por ejemplo, para pintar vehículos. Un dispositivo de medición con un sensor de color, el medidor de color, puede comprobar si se usa el color adecuado para pintar con aerosol un vehículo. ¿Tú y tus compañeros de clase de tu grupo, podéis pensar en otros ejemplos?

#### 2. ¿Cómo funcionan los sensores de color? ¿Qué tiene de específico el sensor de color del mBot2?

Un sensor de color consta de tres sensores (internos) diferentes que miden la luz reflejada por un objeto en función de la intensidad de los valores de los colores Rojo, Verde y Azul. Los sensores de color también se denominan "sensores RGB" (por las siglas de los colores en inglés). El sensor decodifica cualquier color en tres valores. Una "mezcla" específica de estos valores representa un color.

El mBot2 tiene cuatro de estos sensores RGB integrados en un solo sensor. Observa la siguiente imagen.





Este es el sensor Quad RGB que se encuentra en el mBot2. Los sensores RGB se llaman L1, L2, R1 y R2 (L para la parte izquierda y R para la derecha). Detectan automáticamente los valores RGB del color reflejado y comparan la mezcla de los valores internamente con diferentes colores predeterminados. Esto hará que la codificación sea mucho más fácil porque el sensor informa sobre el color y no es necesario que compruebes los códigos RGB tú mismo. El sensor puede detectar 6 colores diferentes además del blanco y el negro. En lugar del color, puede informar sobre tonos de gris o líneas y uniones.

Los sensores Quad RGB permiten que el mBot2 siga una línea en el suelo y reaccione a diferentes marcadores de color. El sensor Quad RGB se encuentra en la parte frontal del mBot2. Los cuatro sensores están en la parte inferior. En la parte superior encontrarás los nombres de cada uno de ellos, para poder identificar y programar los sensores de una manera específica. Observa las siguientes imágenes.



*Sensor Quad RGB, vista de la parte frontal*

*Vista de la parte superior*

*Vista de la parte inferior*

El pequeño botón en la parte superior del sensor se utiliza para la calibración. Si haces doble clic en él, los LED comenzarán a parpadear y podrás "deslizar" el mBot2 con el sensor a través de la línea que debe seguir. Esto calibrará el sensor para diferenciar la línea del fondo. Si estás utilizando una línea negra sobre una superficie blanca, normalmente este paso no es necesario. El mapa entregado con el mBot2 tiene colores marcados dentro del propio recorrido para reaccionar a los códigos de color. Esto podría hacer que el sensor malinterprete un color brillante (como el amarillo) del fondo en lugar de la pista. Por lo tanto, como primer ejercicio, calibra el sensor con el código de color amarillo. Así el sensor reconocerá automáticamente y de forma fiable este color y todos los colores "más oscuros", como la línea.

Cuando programes el mBot2, usarás uno o más de estos cuatro sensores para verificar ciertos colores en el suelo durante el recorrido. Puedes hacer que el mBot2 ejecute un comando cuando detecte un color. Por ejemplo, "parar" cuando detecte el color rojo e "ir a la izquierda" cuando detecte el color verde.

## 2. Práctica (10 min)

### Paso 2: Práctica

Este paso consta de tres partes:

1. Familiarizarse con los diferentes bloques de código del sensor Quad RGB
2. Recrear y probar algunos ejemplos de programación del sensor Quad RGB
3. ¿Qué hacer para que el mBot2 siga una línea?

#### 1. Familiarizarse con los diferentes bloques de código del sensor Quad RGB

En mBlock, hay varios bloques de código que puedes utilizar para programar el sensor Quad RGB. Encontrarás estos bloques de código en la categoría "Quad RGB" del área de bloques en mBlock. Estos bloques de código son verdes.



En el siguiente cuadro verás algunos de los bloques de código que puedes utilizar para programar el sensor Quad RGB. Los primeros cuatro bloques de código sirven para realizar (sencillas) identificaciones de línea. Se pueden utilizar para seguimientos de línea sencillos y avanzados, pero también para detectar cruces y giros bruscos (90°). La calibración del sensor para diferenciar diferentes pistas de color se explica brevemente en la sección anterior. Los LED azules de la parte superior del sensor indican la detección de la línea/fondo. Si están apagados, detectan un color oscuro (línea), si son azules brillantes, indican un color brillante (fondo). Por lo tanto, el estado del LED indica la reflectividad del suelo o el mapa.

#### Bloque de códigos:



Estos bloques de código solo tienen en cuenta los dos sensores internos L1 y R1 para la detección de la línea. Estos bloques están diseñados para un seguimiento sencillo de línea. Utiliza el primer bloque en declaraciones condicionales. Utiliza el segundo bloque si deseas presentar el valor en la pantalla o guardarlo en una variable.

Si el sensor se encuentra encima de una línea negra, ambos LED internos deben estar apagados. Dado que cada sensor (L1, R1) puede detectar la línea o el fondo con este bloque, hay 4 combinaciones posibles. Las combinaciones estarán representadas por números de 0 a 3 (como referencia, el bloque también muestra la secuencia binaria):

L1	R1	Significado	Estado (Decimal)
		Sensor sobre fondo blanco, ambos LED están encendidos	0
		R1 detecta una línea negra	1
		L1 detecta una línea negra	2
		Ambos sensores internos detectan la línea negra	3

Recuerda que los LED muestran el estado de fondo (encendido o "verdad" lógica = fondo detectado), mientras que este bloque de código detecta el color de la línea ("verdad" lógica = línea detectada).

Si deseas utilizar más de un sensor Quad RGB, puedes indicarle al programa qué sensor utilizar con el primer número en todos los siguientes bloques de código. Observa la siguiente imagen. El número 1 indica el primer sensor conectado, el número 2 el segundo sensor conectado y así sucesivamente.

El siguiente ejemplo utilizará la función de detección de línea de este bloque y detendrá el robot tan pronto como llegue a una línea negra. Este ejemplo se cambiará más tarde al seguimiento de línea.

```

when button A pressed
  moves forward at 50 RPM
  wait until quad rgb sensor 1 L1, R1's line status is (3) 11 ?
  stop encoder motor all
  
```

Dado que la reacción durante la conducción puede ser de máxima prioridad en cuanto al tiempo se refiere, es importante utilizar el modo Carga. Puedes intentarlo por ti mismo y ver la diferencia si cambias de modo.

### Bloque de códigos:



Utiliza el primer bloque en declaraciones condicionales. Utiliza el segundo bloque si deseas presentar el valor en la pantalla o guardarlo en una variable.

Estos bloques siguen exactamente la misma lógica que los anteriores, excepto que tienen un área de detección más grande y pueden identificar cruces sin dejar de seguir la línea.

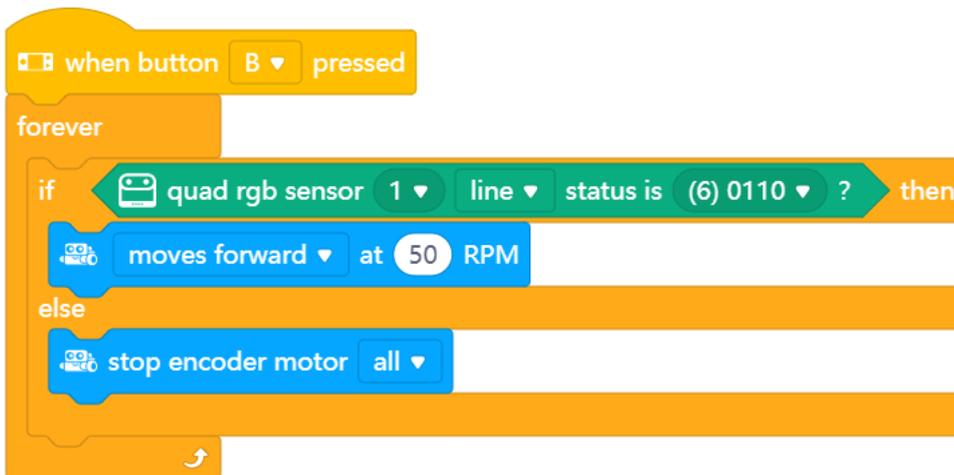
Recuerda, los bloques verifican el estado de la línea, por lo que si uno de los cuatro sensores identifica una línea (oscura), el dígito binario correspondiente se establecerá en "1" (lógica verdadera), pero los LED se apagarán (no hay reflejo en este sensor desde el fondo).

Teniendo ahora 4 sensores individuales, la gama se incrementa a 16 estados diferentes. La siguiente tabla presenta los estados más relevantes:

Secuencia binaria	Significado	Estado (Decimal)
0000	los 4 sensores sobre fondo blanco (sin pista)	0
0010	solo R1 sobre línea negra	2
0100	solo L1 sobre línea negra	4
0110	sensores interiores R1 y L1 sobre línea negra	6
0111	giro brusco a la derecha (L1 y R1 sobre línea negra, cruce a la derecha; cruce en forma de L a la derecha)	7
1110	giro brusco a la izquierda (L1 y R1 sobre línea negra, cruce a la izquierda; cruce en forma de L a la izquierda)	14
1111	Cruce en forma de T (L1 y R1 sobre línea negra, cruce a la derecha y la izquierda)	15

Un cruce en forma de X solo se puede detectar desplazándose sobre el cruce en forma de T y verificando si la línea del medio continúa.

El siguiente ejemplo de programa hará que el robot se desplace sobre una línea negra, siempre que haya una, con los dos sensores internos. Se detendrá si se sale de la pista, se detecta un cruce o la línea termina repentinamente:



#### Bloque de códigos:



Con este bloque de código, selecciona uno de los sensores Quad RGB (L2, L1, R1, R2) y determina si debe reconocer una línea, fondo, blanco, negro o cualquiera de los 5 colores restantes (rojo, verde, azul, amarillo, turquesa, morado). Para la detección de línea y el seguimiento, se recomienda utilizar los bloques anteriores porque verifican más de un sensor a la vez (más rápido y de forma más precisa ya que el robot no se mueve entre lecturas). Pero puedes utilizar este bloque para detectar, por ejemplo, un marcador de color rojo en el lado izquierdo del robot, mientras el robot sigue una línea.

En el siguiente ejemplo de programación, si pulsas el botón B, el mBot2 se desplace hacia adelante (asumiendo que sigue una línea) y hace un giro de 90° cuando detecta un marcador rojo en el lado izquierdo de la pista (imaginaria).

```

when button A pressed
  moves forward at 50 RPM
  wait until quad rgb sensor 1 probe (4) L2 detected red
  turns left 90 ° until done
  stop encoder motor all
  
```

### Bloque de códigos:

```

quad rgb sensor 1 deviation (-100~100)
  
```

Este bloque de código mide la distancia entre el robot y una pista negra. Puedes utilizarlo para un seguimiento de líneas avanzado y suave. El bloque de código de seguimiento de línea anterior utiliza una triple distinción: en la pista, desviado a la izquierda o desviado a la derecha (más la detección de cruces añadida en algunos casos). Pero en la vida diaria, al andar en bicicleta, tendrías que ajustar la dirección a la curva con mayor precisión. Este bloque permite un seguimiento de línea proporcional: cuanto más se desvíe el robot de la pista, más se dirigirá en la dirección opuesta. Si la desviación de la pista es de 0, la dirección también es de 0 (recta).

En el paso 3, analizaremos cómo podemos utilizar este bloque para seguir una línea "suavemente" programando un seguidor de línea proporcional. El siguiente ejemplo de programa mostrará la desviación en la pantalla. Puedes intentar mover el robot en la pista con la mano:

```

when CyberPi starts up
  forever
    show quad rgb sensor 1 deviation (-100~100) at center of screen by middle pixel
  
```

## 2. Recrear y probar algunos ejemplos de programación del sensor Quad RGB

En el diagrama anterior, cada bloque de código del sensor Quad RGB va acompañado de un ejemplo de programación. Recrea estos ejemplos de programación en mBlock y pruébalos. Elabora tus propias modificaciones sobre el código de un ejemplo de programación.

## 3. ¿Qué hacer para que el mBot2 siga una línea?

La base de la tarea del "Paso 3" de esta lección es que el mBot2 seguirá el recorrido en un mapa, siendo un autobús turístico. Para ello, deberás utilizar los ejemplos de código explicados y probados hasta ahora y modificarlos como corresponda.

Primero utiliza este bloque de código:



Si el robot está en la pista (tanto L1 como R1 detectan una línea, valor código 3), el robot debe desplazarse en línea recta. Si solo uno de los dos sensores internos identifica una línea, entonces deberá girar a la izquierda o derecha como corresponda.

Utiliza estos bloques de conducción (solo se muestra el giro a la derecha):



Arrancan los motores según sea necesario y se ejecutará el siguiente bloque de código inmediatamente. Dado que no sabes la distancia hasta el robot o cuándo aparecerá una curva, los comandos de conducción no pueden incluir ningún tiempo o distancia de conducción (lo mismo ocurre para girar). Cuando se reitere un ciclo de programa, se comprobarán los nuevos datos del sensor y se seleccionará la conducción como corresponda.

Si estás utilizando el mapa suministrado, asegúrate de calibrar el sensor como se describe al color más brillante (sección amarilla en la pista).

### 3. Prueba (25 min)

#### Paso 3: Prueba

Ya has aprendido mucho sobre el sensor Quad RGB del mBot2. Ahora vas a crear tu propio programa informático en mBlock utilizando el sensor Quad RGB.

En esta tarea convertirás el mBot2 en un vehículo turístico. El autobús turístico hace un recorrido por los lugares emblemáticos de una ciudad. En la caja del mBot2 hay un mapa con un recorrido. El autobús sigue la línea negra del mapa. Ya has programado un simple robot que sigue una línea en el apartado anterior. Ahora ha llegado el momento de añadir más mejoras. Para que el programa sea más fácil de entender, puedes "construir" tus propios bloques de codificación. Toda la parte de seguimiento de líneas del programa puede añadirse a un simple comando, por ejemplo, "simple\_línea". La opción roja/rosa de "Mis Bloques" del menú de códigos permite definir un bloque personalizado y estructurar tu código de una manera agradable. El ejemplo de programa siguiente utiliza bloques personalizados.



Además de seguir la pista, el autobús ejecuta una acción en algunos de los lugares emblemáticos. Los lugares emblemáticos están indicados mediante un área de color (rojo, amarillo, verde y azul).

Puedes inventarte las acciones que el autobús debe realizar. ¿Vas a hacer que el autobús recorra París? Entonces podría ser divertido escuchar el himno nacional francés. O las campanas del Big Ben si el autobús hace un recorrido por Londres.

Por supuesto, también puedes hacerlo más sencillo. En el siguiente ejemplo, haz que el autobús dé la vuelta en el marcador rojo y ejecute diferentes actividades en otros marcadores de color.



Utiliza los conocimientos adquiridos en el "Paso 2" de esta lección. Por supuesto, puedes experimentar todo lo que quieras por ti mismo con los diferentes ejemplos de programación que encontrarás en mBlock. ¿Te has bloqueado? En ese caso, puedes utilizar el siguiente ejemplo de programación. Puedes copiarlo y adaptarlo de acuerdo con tus propios deseos y preferencias.

```
when joystick middle pressed
repeat until button B pressed?
  simple_line_follow
  if quad rgb sensor 1 probe (2) R1 detected blue ? then
    moves forward at 10 RPM for 1 secs
  if quad rgb sensor 1 probe (2) R1 detected red ? then
    turns left 180 ° until done
    moves forward 4 cm until done
  if quad rgb sensor 1 probe (2) R1 detected green ? then
    turns left 90 ° until done
    wait 1 seconds
    turns right 180 ° until done
    wait 1 seconds
    turns left 90 ° until done
    moves forward 4 cm until done
stop encoder motor all
```

Este programa se inicia si pulsas la tecla B y se detiene presionando A. Realiza un simple seguimiento de línea usando el código personalizado "Mis Bloques" y luego comprueba si el sensor R1 detecta el color azul (desplazarse muy lentamente), rojo (dar la vuelta), o verde (mirar a ambos lados). Una vez realizadas las actividades, para asegurarse de que no se vuelva a reconocer el color, el robot avanza 4 cm cada vez.

El bloque "simple\_line\_follow" se define de la siguiente manera:

```

define simple_line_follow
if quad rgb sensor 1 L1, R1's line status is (3) 11 ? then
  moves forward at 50 RPM
if quad rgb sensor 1 L1, R1's line status is (1) 01 ? then
  turns right at 20 RPM
if quad rgb sensor 1 L1, R1's line status is (2) 10 ? then
  turns left at 20 RPM
if quad rgb sensor 1 L1, R1's line status is (0) 00 ? then
  moves backward at 10 RPM
  
```

### Versión avanzada:

En el paso anterior, se ha explicado la desviación de la línea. Este bloque informa sobre la distancia de desviación del recorrido del robot (de -100 a +100; 0 para justo en el centro) y se puede usar para calcular la dirección del robot. Se trata de un seguimiento de línea proporcional; cuanto más se desvíe el robot, más necesitará dirigirse en sentido opuesto. Si el robot está fuera de pista en la parte derecha, la desviación es positiva (los robots deben conducir hacia la izquierda); en caso de valores negativos, el robot está fuera de pista en la parte izquierda (necesita conducir hacia la derecha).

Para implementar esto, necesitas una velocidad básica, que es la velocidad necesaria para avanzar en una pista recta. Para girar hacia la izquierda o hacia la derecha, una de las ruedas debe girar más rápido que la otra; cuanto mayor es la diferencia, mayor es el giro.

Para calcular la dirección, debemos sumar la desviación a la velocidad básica de un motor y restarla del otro. Para permitir un ajuste más preciso, se utilizará un factor para ajustar la intensidad de la desviación en la dirección, ya que el rango de desviación en sí es demasiado grande para utilizarse directamente para controlar los motores.

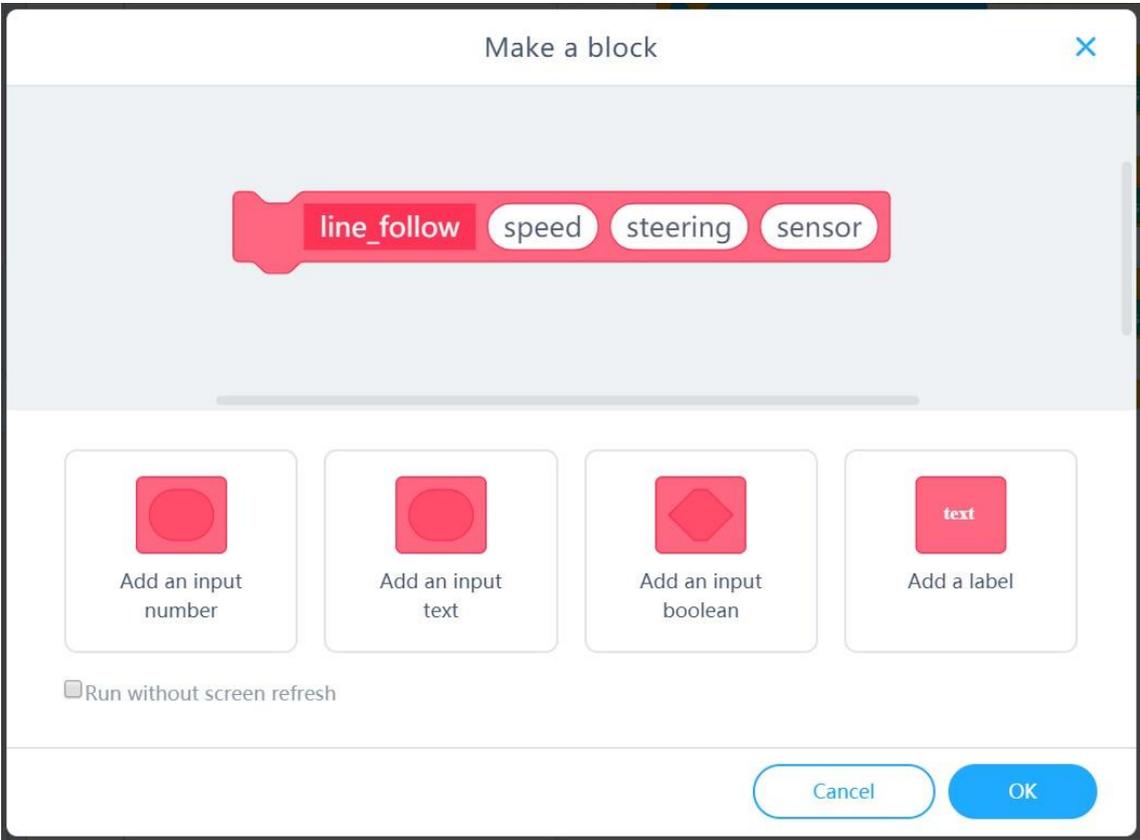


Pero, ¿a qué motor se le debe sumar o restar el valor calculado? Una desviación positiva significa que el robot está fuera de pista en el lado derecho y debe girar a la izquierda. Esto significa que la rueda derecha debe funcionar más rápido; añadimos la desviación escalada a la rueda derecha y, por lo tanto, la restamos de la izquierda.

$$\text{speed} + \text{steering} * \text{sensor}$$

$$\text{speed} - \text{steering} * \text{sensor}$$

Al configurar "Mis bloques", puedes añadir entradas y darles un nombre que signifique algo:



Lo único que queda por tener en cuenta es que el bloque de movimiento (mostrado a continuación) requiere valores negativos para que el motor derecho avance, y positivos para que el motor izquierdo avance (ya que están montados uno frente al otro).

encoder motor EM1 rotates at 50 RPM, encoder motor EM2 rotates at 50 RPM

Para finalizar el cálculo, los valores del motor derecho deben multiplicarse por -1. Esta es la definición del bloque de seguimiento de línea proporcional:

```

define line_follow speed steering sensor
set left to (-1 * speed + steering * sensor)
set left to (1 * speed - steering * sensor)
encoder motor EM1 rotates at left RPM, encoder motor EM2 rotates at right RPM
  
```

En el circuito principal de un programa, simplemente debes utilizar este bloque con un comando:

line\_follow 50 0.6 quad rgb sensor 1 deviation (-100~100)

Esto solicita al subprograma con una velocidad básica de 50 y un factor de dirección de 0,6 y también envía la desviación actual de la lectura del sensor. Puedes realizar pruebas con los valores. Si el robot "oscila" (cambiando rápidamente de izquierda a derecha, etc.), el factor de dirección es demasiado alto. Si se desvía incluso en curvas poco pronunciadas, es demasiado bajo.

Al trabajar en esta tarea, es útil seguir el siguiente esquema paso a paso. ¿Tienes idea de lo que quieres hacer? Si es así, coméntalo primero con tu profesor para ver si es factible.

	Explicación
Paso 1: ¿Qué quieres hacer?	<ul style="list-style-type: none"> <li>¿De qué forma debería moverse el mBot2?</li> </ul>
Paso 2: ¿Qué necesitas?	<ul style="list-style-type: none"> <li>¿Qué necesitas además del mBot2?</li> </ul>
Paso 3: ¿Qué bloques de código necesitas para hacer que el mBot2 se desplace?	<ul style="list-style-type: none"> <li>¿Qué debes hacer para que el mBot2 cambie de dirección en cada esquina?</li> <li>¿Qué bloques de código necesitas para realizar esta acción?</li> </ul>



	<ul style="list-style-type: none"><li>• Describe brevemente cómo funciona tu programa (pseudocódigo/lenguaje natural, diagrama de flujo o Lenguaje Unificado de Modelado (UML))</li><li>• Si necesitas más explicaciones, puedes hablar con sus compañeros de clase, el profesor o investigar sobre el tema. También tienes ayuda disponible para cada bloque de códigos en mBlock</li></ul>
Paso 4: Pruebas e implementación	<ul style="list-style-type: none"><li>• ¿Está lista la primera versión? ¡Pruébala! Durante la ronda de pruebas, anota los puntos de mejora</li><li>• Trabaja en los puntos de mejora hasta que tu mBot2 se desplace sin errores sobre la pista</li><li>• ¿Lo has conseguido? Graba el resultado final y pregúntale a tu profesor si puedes publicarlo en las redes sociales con el hashtag #mBot2inclass</li></ul>

## 4. Conclusión (5 min)

### Paso 4: Conclusión

¿Has conseguido programar el mBot2 como un vehículo turístico que circula por una pista? En esta lección has aprendido sobre el sensor Quad RGB y dónde puedes encontrarlo en la vida diaria. Sabes cómo programar el sensor Quad RGB del mBot2 para que el robot siga una línea. También has aprendido a programar el mBot2 para que circule por un recorrido basado en diferentes colores.

Ha llegado el momento de realizar una breve reflexión. Piensa por ti mismo y discute con el grupo:

- ¿Qué crees que ha salido bien?
- ¿Qué podría mejorarse?
- ¿Qué partes de la lección te han resultado fáciles y cuáles más difíciles?
- ¿Sobre qué te gustaría recibir más información?
- ¿Quién podría ayudarte con eso?

## Lección 6: Conducción cuidadosa

**Tema:** STEAM

**Curso(s):** A partir de 5º

**Duración:** 45 minutos

**Dificultad:** Principiante

### ★ Objetivos de la lección

*Al final de esta lección, los estudiantes serán capaces de:*

- Reconocer y utilizar los bloques de código para el giroscopio y el acelerómetro en el CyberPi
- Construir su propio programa informático en mBlock para medir los cambios en la inclinación del mBot2 y ajustar sus movimientos como corresponde

### ★ Resumen

Un giroscopio mide los movimientos de inclinación, más precisamente la velocidad de los movimientos de giro/inclinación. Un acelerómetro mide el cambio de velocidad. Ambos sensores proporcionan información diferente sobre la posición en el espacio.

Combinando ambos tipos de datos, se puede calcular la posición y el movimiento de vehículos, robots o incluso de sondas espaciales. En carreteras con baches, por ejemplo, los robots pueden conducir más lento para evitar vuelcos.



### 🔗 Puntos de enfoque

*Al final de esta lección, los estudiantes sabrán:*

- Cómo funcionan los giroscopios y acelerómetros y como se pueden combinar sus mediciones.
- Cómo hacer que un robot reaccione a cambios repentinos y detecte carreteras accidentadas o con baches

## Lista de verificación previa a la lección

¿Qué necesitas?

- PC u ordenador portátil (con salida USB) con el *software* mBlock instalado, la versión web (también para ChromeBook) o una tableta con la aplicación mBlock instalada
- El mBot2 con el CyberPi
- Un cable USB-C o un adaptador (*dongle*) Bluetooth Makeblock

## Plan de la lección

Esta lección consta de cuatro pasos y tiene una duración de 45 minutos.

Duración	Contenido
5 minutos	<b>1. Calentamiento</b> <ul style="list-style-type: none"><li>• Giroscopios y acelerómetros en la vida diaria</li><li>• ¿Qué movimientos de inclinación puede detectar el mBot2?</li></ul>
10 minutos	<b>2. Práctica</b> <ul style="list-style-type: none"><li>• Familiarizarse con los diferentes bloques de programación del giroscopio y el acelerómetro.</li><li>• Recrear y probar algunos ejemplos de programación de estos sensores.</li></ul>
25 minutos	<b>3. Prueba</b> <ul style="list-style-type: none"><li>• Escribir tu propio programa para el robot</li></ul>
5 minutos	<b>4. Conclusión</b> <ul style="list-style-type: none"><li>• ¿Qué has aprendido en esta lección?</li><li>• Hora del espectáculo: muestra lo que has hecho con tu robot en una película corta y divertida para poder comentarlo más adelante</li><li>• Si tu profesor lo permite, comparte el resultado final en las redes sociales con el hashtag #mBot2inclass</li><li>• Reflexión: ¿De que estas más orgulloso/a? ¿Qué te gustaría mejorar de tu robot?</li></ul>

## 1. Calentamiento (5 min)

### Paso 1: Calentamiento

Este paso consta de dos partes:

1. Giroscopios y acelerómetros en la vida diaria
2. ¿Qué movimientos de inclinación puede detectar el mBot2?

### 1. Giroscopios y acelerómetros en la vida diaria

¿Cómo identifican su posición los dispositivos como los *smartphones* (mantener en posición vertical, pantalla hacia arriba/abajo)? ¿Cómo determinar si un choque es lo suficientemente grave como para activar los *airbags*? Durante siglos, calcular la posición actual sin una referencia externa precisa ha supuesto un desafío en el mar cuando el cielo estaba cubierto, y sigue siendo una tarea muy importante para artículos domésticos, robots e incluso sondas espaciales. Todos usan al menos dos sensores diferentes, giroscopios y acelerómetros. Un giroscopio funciona con un cristal que está bajo tensión eléctrica. Esto convierte la dirección de la fuerza de inclinación en una señal eléctrica. Los giroscopios miden la velocidad de rotación en grados por segundo y, con la ayuda de las matemáticas, también pueden calcular el ángulo de rotación. Pero este último no es tan preciso: los giroscopios "se desvían" con el paso del tiempo. Los acelerómetros informan del cambio de posición (aumento o disminución de la velocidad) y, con respecto a la gravedad de la tierra, informan sobre la aceleración gravitacional (en comparación con la caída libre). Ambos tipos de sensores pueden medir en cada eje espacial, informando sobre la velocidad de rotación o la aceleración en los ejes X, Y y Z.

Combinando los datos, tanto del giroscopio como del acelerómetro, se pueden calcular la dirección y la fuerza de los movimientos (en realidad, este cálculo no es muy preciso, pero sirve como una muy buena estimación). En robótica, estos sensores se utilizan para estabilizar vehículos en terrenos irregulares (o en vuelo) y ayudarles a mantener su trayectoria programada. En lo que respecta al mBot2, estos sensores están integrados en un pequeño chip en el CyberPi.

Por lo tanto, denominaremos al sensor combinado como "sensor de movimiento".

## 2. ¿Qué movimientos de inclinación puede detectar el mBot2?

Antes de aprender a programar en mBlock5 con el sensor de movimiento, es útil saber qué movimientos de inclinación puede medir el mBot2. Puedes mostrar las medidas de los movimientos de inclinación en la pantalla del CyberPi. Ya has aprendido a hacer esto en la Lección 2.

El mBot2 puede detectar los siguientes movimientos de inclinación:



## 2. Práctica (10 min)

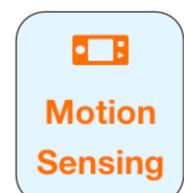
### Paso 2: Práctica

Este paso consta de dos partes:

1. Familiarizarse con los diferentes bloques de código del sensor de movimiento
2. Recrear y probar algunos ejemplos de programación del sensor de movimiento

### 1. Familiarizarse con los diferentes bloques de código del sensor de movimiento

En mBlock hay varios bloques de código para el sensor de movimiento que puedes utilizar en tus propios programas. Estos bloques de código se pueden encontrar en la categoría "Detección de movimiento" del campo de bloque en mBlock. Estos bloques de código son de color naranja.



El siguiente cuadro proporciona una explicación detallada de los bloques de código junto con ejemplos de códigos de programa para probarlos.

### Bloque de códigos:

**shaking strength**

Este bloque de código informa la fuerza con la que tiembla el mBot2. Con un valor de "0", el mBot2 no tiembla en absoluto. Con un valor de "100" se detecta un movimiento de temblor muy fuerte. En el siguiente ejemplo de programación, se muestra en la pantalla la fuerza del temblor del mBot2. Si el mBot2 para de temblar, la pantalla se pone en blanco.

```

when CyberPi starts up
  forever
    show shaking strength at center of screen by middle pixel
  
```

### Bloque de códigos:

**waving speed**

Este bloque de código se utiliza para registrar la velocidad de ondulación. La velocidad de ondulación se mide en una escala de 0 a 100. Estas unidades no representan la velocidad en m/s o en grados/s. Son solo una simple referencia.

En el siguiente ejemplo de programación, se muestra en la pantalla la velocidad de ondulación del mBot2. Si el mBot2 para de ondular, la pantalla se pone en blanco.

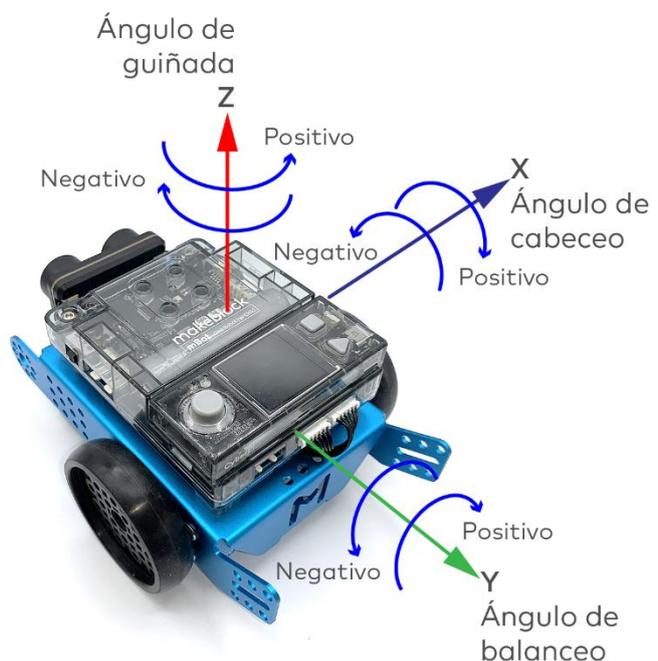
```

when CyberPi starts up
  forever
    show waving speed at center of screen by middle pixel
  
```

## Bloque de códigos:

 tilted forward ▾ angle (°)

Cada inclinación hace un movimiento en un cierto ángulo en una cierta dirección. Al especificar un ángulo, los ejes X, Y y Z se utilizan para referirse a la dirección. Además de estos términos matemáticos, también se les conoce como balanceo, cabeceo y bandazo. Estos términos provienen de la navegación de aviones. "Balanceo" es el eje de adelante hacia atrás, "cabeceo" hace referencia a los giros arriba/abajo (eje de izquierda a derecha) y el "bandazo" es el eje de arriba hacia abajo responsable de los giros izquierda/derecha. Observa la siguiente imagen.



Los ángulos están representados en grados:

- Ángulo de balanceo: de  $-180^{\circ}$  a  $+180^{\circ}$
- Ángulo de bandazo: de  $-180^{\circ}$  a  $+180^{\circ}$
- Ángulo de cabeceo: de  $-90^{\circ}$  a  $+90^{\circ}$

En el siguiente programa, el ángulo de cabeceo se mostrará en la línea superior y si se detecta un movimiento de inclinación hacia adelante o hacia atrás, las palabras "adelante" o "atrás" aparecen en la última línea. ¿Puedes identificar el umbral de detección?

```

when clicked
  forever
    show label 1 tilted backward(pitch) angle (°) at center of screen by middle pixel
    if tilted forward ? then
      show label 1 FORWARD at center of screen by middle pixel
    else
      if tilted backward ? then
        show label 1 BACKWARD at center of screen by middle pixel
      else
        show label 1 ○ at center of screen by middle pixel
  
```

Intenta experimentar con las diferentes configuraciones del menú desplegable. Has analizado el movimiento de inclinación en el cabeceo/eje X. Si cambias a inclinación izquierda/derecha, recuerda cambiar también los bloques Boolean informados:

```

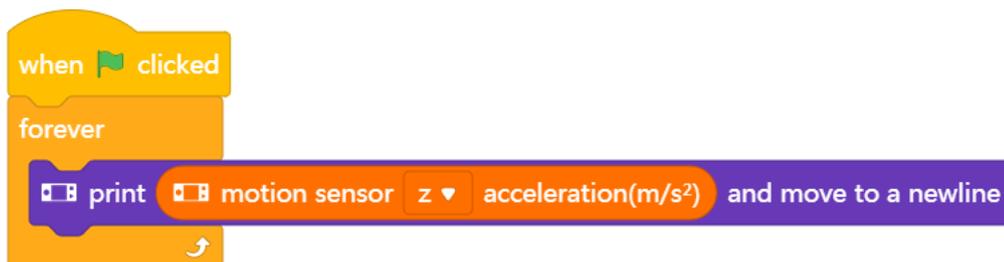
when clicked
  forever
    show label 1 tilted right(roll) angle (°) at center of screen by middle pixel
    if tilted right ? then
      show label 1 RIGHT at center of screen by middle pixel
    else
      show label 1 ○ at center of screen by middle pixel
  
```

## Bloque de códigos:

**motion sensor** x ▼ **acceleration(m/s<sup>2</sup>)**

Hasta ahora, hemos analizado los movimientos de inclinación, pero los sensores de movimiento incorporan otro sensor, el acelerómetro. Mide los cambios en la velocidad del mBot2 a lo largo del tiempo. Dichos cambios pueden apreciarse por ejemplo en un aumento lento de la velocidad de conducción, así como una parada repentina o incluso un impacto lateral de otro vehículo. Dado que estas medidas se basan en los ejes X, Y y Z del robot, puedes detectar un choque si buscas un cambio rápido en la aceleración. El acelerómetro indica la aceleración en m/s<sup>2</sup>.

En el siguiente ejemplo de programación, se muestra en la pantalla la aceleración del mBot2 en el eje Z. Mueve el mBot2 con la mano y comprueba la pantalla. ¿Qué lecturas muestra si el robot "choca" contra tu mano o un libro? ¿Puedes cambiar el programa y simular un impacto lateral?



```
when clicked
  forever
    print motion sensor z acceleration(m/s2) and move to a newline
```

## Bloque de códigos:

**angle speed around** x ▼ **axis(°/s)**

Este bloque de código muestra la velocidad a la que gira el mBot2 en torno a un eje determinado. A esto también se le llama velocidad angular. La velocidad angular se muestra en grados por segundo (°/s).

En el siguiente ejemplo de programación, se muestra en la pantalla la velocidad angular del mBot2 en torno al eje X.



```
when space key pressed
  forever
    show angle speed around x axis(°/s) at center of screen by middle pixel
```

## Bloque de códigos:



Este bloque de código muestra el ángulo al que gira el mBot2 en torno a un eje determinado. El ángulo se muestra en grados (°) y se mide relativamente. La posición inicial se establece cuando se enciende el mBot2. Estos datos se calculan a partir de la velocidad de rotación que informa el giroscopio. Puede desplazarse con el tiempo y no es tan estable. Intenta inclinar el mBot2 hacia los lados con el siguiente programa y verifica los valores. ¿Vuelven a su estado inicial?

En el siguiente ejemplo de programación, se muestra en la pantalla el ángulo al que gira el mBot2 en torno al eje Y.



Hay dos aspectos importantes: este bloque hará el seguimiento de los movimientos de rotación superiores a una revolución. Utiliza este bloque para contar cuántas veces ha girado el robot. Mirando el eje Z, una rotación en el sentido de las agujas del reloj aumentará los valores informados por este bloque,



mientras que el ángulo de rotación en torno al eje Z disminuye:



Después de 3 rotaciones de 360°, el último bloque informará "-1080°", pero el bloque anterior solo mostrará "0" (de regreso a la posición inicial, ignorando múltiples rotaciones). Si luego se gira en sentido contrario a las agujas del reloj 4 veces (una vez más anteriormente), el ángulo de "rotate clockwise" (girar\_en\_el\_senso\_de\_las\_agujas\_del\_reloj) vuelve a estar en 0°, mientras que el "rotated angle around z axis" (ángulo\_de\_rotación\_en\_torno\_al\_eje\_Z) muestra ahora 360° (todas las figuras sin giro del giroscopio).

Si solo necesitas tener en cuenta pequeños movimientos de rotación para los ejes X e Y, los bloques de "ángulo inclinado (\_\_\_)" tratados anteriormente proporcionan una medición más rigurosa.

Hay una excepción: las rotaciones en torno al eje Z/bandazo solo se pueden medir con la ayuda del giroscopio. Si quieres medir los grados, y el mBot2 está girando hacia la izquierda/derecha, tienes que trabajar con estos datos.

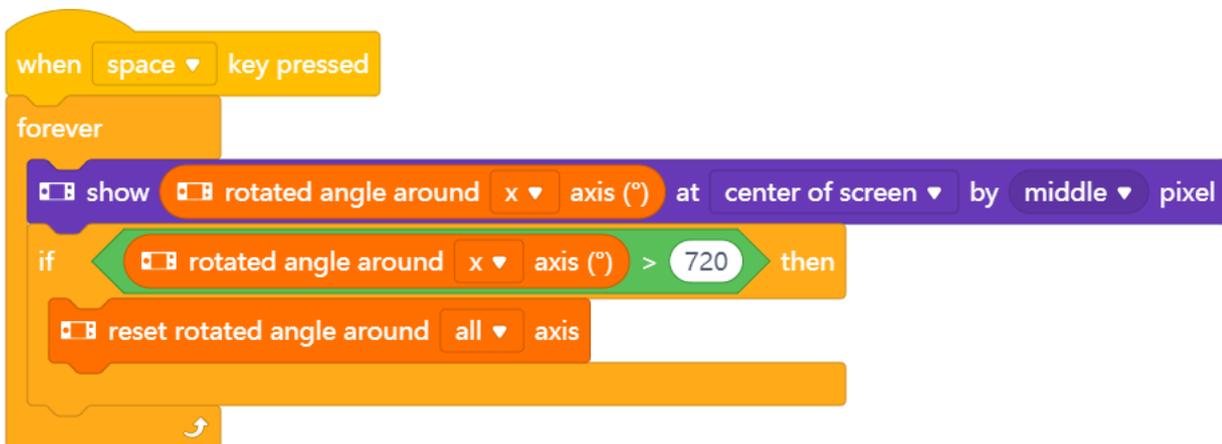
### Bloque de códigos:



reset rotated angle around all axis

Si quieres determinar el ángulo que forma el mBot2, es más fácil medir el cambio desde el valor 0. Este bloque de código restablece el valor medido de un ángulo al valor inicial 0, definiendo una nueva posición inicial para futuros giros.

En el siguiente ejemplo de programación, se muestra en la pantalla el ángulo al que gira el mBot2 en torno al eje Z. Si el ángulo es superior a 720°, el valor medido del ángulo se establece en 0 y se vuelve a medir.



```
when space key pressed
  forever
    show rotated angle around x axis (°) at center of screen by middle pixel
    if rotated angle around x axis (°) > 720 then
      reset rotated angle around all axis
```

Recuerda, hay dos bloques para medir la rotación: uno solo tiene en cuenta las rotaciones completas, el otro realiza el conteo. Este bloque de reinicio establece todos los bloques "rotated angle around \_ axis" (rotación\_alrededor\_del\_eje) (o los seleccionados) a cero. No tiene ningún efecto en el bloque "girar en sentido de las agujas del reloj"/"girar en sentido contrario a las agujas del reloj". Este tiene su propio bloque de reinicio:



reset yaw angle

Cualquier reinicio a cero se aplica únicamente a las rotaciones informadas por el sensor giroscópico solo. El ángulo \_\_\_ inclinado siempre informa desde un plano perfectamente nivelado perpendicular al centro de la tierra y no se puede establecer en cero cuando uno quiere.

## 2. Recrear y probar algunos ejemplos de programación del sensor de movimiento

En el cuadro anterior cada bloque de código del sensor de movimiento va acompañado de un ejemplo de programación. Recrea estos ejemplos de programación en mBlock y pruébalos.

El sensor de movimiento del CyberPi registra los movimientos de inclinación del mBot2 cuando prueba los ejemplos de programación anteriores; por lo tanto, solo necesitas el CyberPi. Si te resulta más práctico, retira el CyberPi del mBot2 y realiza los movimientos de inclinación con el CyberPi tú mismo durante la prueba. Puedes probar los ejemplos de programación en modo Tiempo Real. La batería está en la placa del mBot2, por lo que, para ejecutar los programas, deberás conectar el CyberPi al ordenador utilizando el cable USB-C.

## 3. Prueba (25 min)

### Paso 3: Prueba

Ya has aprendido mucho sobre el sensor de movimiento del mBot2. Ahora vas a crear tu propio programa informático en mBlock utilizando este sensor.

En esta tarea vas a crear una pista llena de baches para el mBot2. Puedes construir esta pista a partir de, por ejemplo, piedras (pequeñas) o plastilina sobre cartón. La idea es que el mBot2 se desplace por la pista. Si el mBot2 tiembla (con fuerza), debe desplazarse más lentamente.

También puedes colocar un objeto en el mBot2 que no debe caerse mientras conduces.

Es una tarea complicada, ¿no? Afortunadamente, ¡no tienes que inventarlo todo por ti mismo!

A continuación encontrarás un ejemplo de programación que puedes ampliar. En este ejemplo de programación, el mBot2 disminuirá la velocidad si realiza movimientos de inclinación demasiado grandes.

```

when space key pressed
  reset rotated angle around all axis
  repeat until button B pressed?
    if tilted right(roll) angle (°) > 10 or tilted right(roll) angle (°) < -10 then
      moves forward at 20 RPM
    else
      moves forward at 40 RPM
  stop encoder motor all

```

Utiliza los conocimientos adquiridos en el "Paso 2" de esta lección. Por supuesto, puedes experimentar todo lo que quieras por ti mismo con los diferentes ejemplos de programación que encontrarás en mBlock.

Al analizar esta tarea, es útil seguir el siguiente esquema paso a paso.

	Explicación
Paso 1: ¿Qué quieres hacer?	<ul style="list-style-type: none"> <li>¿Qué camino quieres que siga el mBot2?</li> <li>¿Cuándo tiene que disminuir la velocidad el mBot2?</li> <li>¿Quieres que los sensores del mBot2 muestren también los datos en la pantalla?</li> </ul>
Paso 2: ¿Qué necesitas?	<ul style="list-style-type: none"> <li>¿Qué necesitas además del mBot2?</li> </ul>
Paso 3: ¿Qué bloques de código necesitas para hacer que el mBot2 se desplace?	<ul style="list-style-type: none"> <li>¿Qué vas a hacer para que el mBot2 se desplace?</li> <li>¿Qué bloques de código usarás?</li> <li>Describe brevemente cómo funciona tu programa (pseudocódigo/lenguaje natural, diagrama de flujo o Lenguaje Unificado de Modelado (UML))</li> <li>Si necesitas más explicaciones, puedes hablar con sus compañeros de clase, el profesor o investigar sobre el tema. También tienes ayuda disponible para cada bloque de códigos en mBlock</li> </ul>

<p>Paso 4: ¿Cómo quieres mostrar los datos de los sensores en la pantalla?</p>	<ul style="list-style-type: none"> <li>• ¿Cómo quieres mostrar los datos en la pantalla?</li> <li>• ¿Qué bloques de código usarás?</li> <li>• Describe brevemente cómo funciona tu programa (pseudocódigo/lenguaje natural, diagrama de flujo o Lenguaje Unificado de Modelado (UML))</li> <li>• Si necesitas más explicaciones, puedes hablar con sus compañeros de clase, el profesor o investigar sobre el tema. También tienes ayuda disponible para cada bloque de códigos en mBlock</li> </ul>
<p>Paso 5: Pruebas e implementación</p>	<ul style="list-style-type: none"> <li>• ¿Está lista la primera versión? ¡Pruébala! Durante la ronda de pruebas, anota las áreas de mejora</li> <li>• Trabaja en los puntos de mejora hasta que tu mBot2 haga exactamente lo que tenías en mente</li> <li>• ¿Lo has conseguido? Graba el resultado final y pregúntale a tu profesor si puedes publicarlo en las redes sociales con el hashtag #mBot2inclass</li> </ul>

## 4. Conclusión (5 min)

### Paso 4: Conclusión

¿Cómo ha ido esta tarea? ¿Has conseguido mantener el objeto en el mBot2 mientras conducías?

En esta lección, has aprendido sobre sensores para medir el movimiento en torno a los ejes y la aceleración; el giroscopio y el acelerómetro. Están integrados en un único componente pequeño colocado en el CyberPi. Estos sensores combinados se utilizan para determinar la posición de un robot en el espacio 3D. Se denominan Unidades de Medida Inercial (IMU por sus siglas en inglés). Has aprendido dónde puedes encontrarlos en la vida diaria. Sabes qué movimientos de inclinación y cambios de movimiento puede medir el mBot2 y cómo puedes utilizar diferentes bloques de código para aprovechar estas medidas en tus programas con el mBot2.

Ha llegado el momento de realizar una breve reflexión. Piensa por ti mismo y discute con el grupo:

- 
- ¿Qué crees que ha salido bien?
  - ¿Qué podría mejorarse?
  - ¿Qué partes de la lección te han resultado fáciles y cuáles más difíciles?
  - ¿Sobre qué te gustaría recibir más información?
  - ¿Quién podría ayudarte con eso?

## Lección 7: Un juego en red

**Tema:** STEAM

**Curso(s):** A partir de 5º

**Duración:** 45 minutos

**Dificultad:** Principiante

### ★ Objetivos de la lección

*Al final de esta lección, los estudiantes serán capaces de:*

- Hacer que varios mBot2 se comuniquen entre sí de forma inalámbrica sin un punto de acceso WIFI
- Escribir su propio programa informático en mBlock para controlar múltiples mBot2
- Aplicar una selección aleatoria en un programa informático

### ★ Resumen

LAN es la abreviatura de Red de Área Local (Local Area Network en inglés). El mBot2, con su módulo WIFI incorporado, establece automáticamente una conexión de red en cuanto hay varios mBot2 o CyberPi en la misma habitación dentro de un radio de 30 metros. A través de esta conexión de red, el mBot2 y el CyberPi pueden comunicarse entre sí y ejecutar comandos.

### 🔗 Puntos de enfoque

*Al final de esta lección, los estudiantes sabrán:*

- Qué es una conexión LAN
- Cómo configurar esta conexión y dónde aplicarla
- Cómo funciona la selección aleatoria en programación

## Lista de verificación previa a la lección

¿Qué necesitas?

- PC u ordenador portátil (con salida USB) con el *software* mBlock instalado, la versión web (también para ChromeBook) o una tableta con la aplicación mBlock instalada
- El mBot2 con el CyberPi (varios mBot2 y/o CyberPi si es posible, para el paso de Prueba)
- Un cable USB-C o un adaptador (*dongle*) Bluetooth Makeblock
- Tarjetas de colores (se sugiere el azul)

## Plan de la lección

Esta lección consta de cuatro pasos y tiene una duración de 45 minutos.

Duración	Contenido
5 minutos	<b>1. Calentamiento</b> <ul style="list-style-type: none"><li>• Una conexión LAN en la vida diaria</li><li>• Cómo configurar una conexión LAN con el mBot2 y el CyberPi</li></ul>
10 minutos	<b>2. Práctica</b> <ul style="list-style-type: none"><li>• Familiarizarse con los diferentes bloques de código para configurar una conexión LAN</li><li>• Recrear y probar programando ejemplos de la conexión LAN</li></ul>
25 minutos	<b>Sección 3 - Prueba</b> <ul style="list-style-type: none"><li>• Crear tu propio juego</li></ul>
5 minutos	<b>4. Conclusión</b> <ul style="list-style-type: none"><li>• Hora del espectáculo: muestra lo que has hecho con tu robot en una película corta y divertida para poder comentarlo más adelante</li><li>• Si tu profesor lo permite, comparte el resultado final en las redes sociales con el hashtag #mBot2inclass</li><li>• Reflexión: ¿De que estas más orgulloso/a? ¿Qué te gustaría mejorar de tu robot?</li></ul>

## 1. Calentamiento (5 min)

### Paso 1: Calentamiento

Este paso consta de dos partes:

1. Una conexión LAN en la vida diaria
2. Cómo configurar una conexión LAN con el mBot2 y el CyberPi

#### 1. Una conexión LAN en la vida diaria

LAN significa Red de Área Local y describe una red de ordenadores y periféricos con una cierta extensión espacial, generalmente dentro de un edificio. La red permite que los ordenadores intercambien datos entre sí dentro de esta red y (sin que sea obligatorio) con otras redes como Internet. Por lo tanto, necesitan algún tipo de conexión (por cable o inalámbrica/WIFI por ondas de radio) y un "lenguaje" común (llamado protocolo). Normalmente, los ordenadores individuales se conectan a un dispositivo de administración que conmuta y enruta el tráfico de datos ("router"). En una red WIFI, este es un punto de acceso dedicado o el router de Internet ofrece un punto de acceso integrado.

¿Puedes mencionar algunos ejemplos en los que se utiliza una red LAN? En esta lección, el mBot2 y el CyberPi son los diferentes dispositivos que se comunican entre sí a través de una conexión LAN.

#### 2. Cómo configurar una conexión LAN con el mBot2 y el CyberPi

Para configurar una conexión LAN entre el mBot2 y el CyberPi no necesitas mucho. El mBot2 y el CyberPi (o los mBot2 entre ellos) podrán comunicarse automáticamente entre sí de forma inalámbrica. No es necesario conectarse a un punto de acceso Wi-Fi o router.

¿Y si se utiliza un router? En este caso, todos los mBot2 y CyberPi tienen que estar en el mismo canal del router. Esto debe establecerse en la configuración del router.

## 2. Práctica (10 min)

### Paso 2: Práctica

Este paso consta de dos partes:

1. Familiarizarse con los diferentes bloques de código para la comunicación LAN
2. Recrear y probar algunos ejemplos de programación de la conexión LAN

### 1. Familiarizarse con los diferentes bloques de código para la comunicación LAN

En mBlock, hay varios bloques de código que puedes utilizar para configurar una conexión LAN. Encontrarás estos bloques de código en la categoría "LAN" del área de bloques en mBlock. Estos bloques de código son verdes.



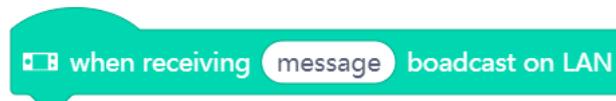
El siguiente cuadro muestra un ejemplo de estos bloques de código para configurar una conexión LAN.

#### Bloque de códigos:



Este bloque de código te permite enviar un mensaje a todos los demás mBot2 y CyberPi de la misma red. Puedes nombrar el mensaje (como un tema).

#### Bloque de códigos:



En lugar de buscar constantemente una transmisión recibida, hay un bloque de eventos. El código vinculado a este bloque solo se ejecuta una vez si se recibe un mensaje con el tema especificado.

En el siguiente ejemplo de programación, hay dos mBot2 y/o CyberPi que se comunican entre sí a través de una conexión LAN utilizando estos dos bloques de código. El primer programa es la "unidad de transmisión", destinado al primer mBot2 o CyberPi. El segundo programa es el código del receptor para el segundo mBot2 o CyberPi. Puedes cargar ambas partes del código en CyberPi/mBot2 ya que el código que se encuentra debajo se asegurará de que la unidad emisora no reciba su propio mensaje.

```

when button A pressed
  broadcast message welcome on LAN
  
```

```

when receiving welcome broadcast on LAN
  turns left 360 ° until done
  play hi until done
  
```

Después de cargar los códigos de programación, debes reiniciar el mBot2 y/o CyberPi. Ahora, cuando pulses el botón A del primer mBot2 o CyberPi, el segundo mBot2 o CyberPi recibirá el mensaje enviado (por ejemplo, "bienvenido"). ¿Qué ocurre si pulsas B en el segundo CyberPi o mBot2? ¿Qué tendrías que cambiar para devolver diferentes mensajes?

#### Bloque de códigos:

```

broadcast message message with value 1 on LAN
  
```

Este bloque de código te permite enviar un mensaje (con un tema determinado como antes) y un valor específico (número o texto) a todos los demás mBot2 y CyberPi de la red.

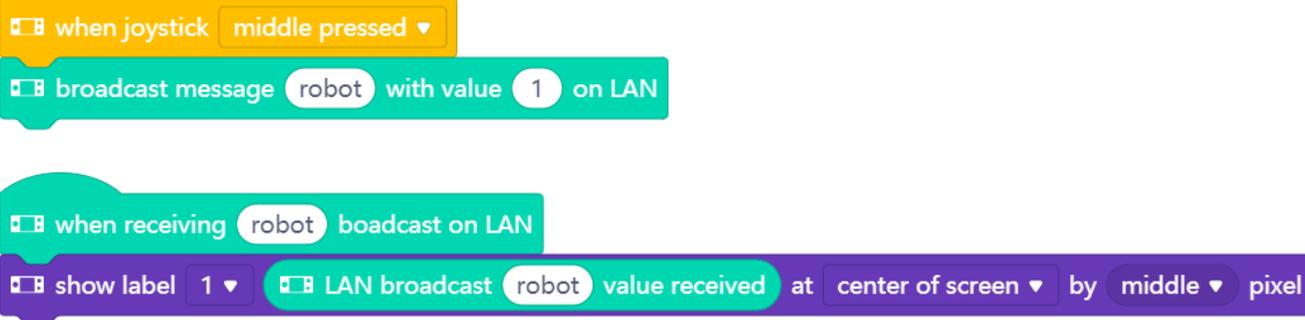
#### Bloque de códigos:

Para el código de recepción, puede usar el mismo bloque de eventos y especificar el tema al que debe reaccionar exactamente como en el ejemplo anterior. Para hacer uso del valor (texto o número), hay un bloque de envío de información dedicado:

```

LAN broadcast message value received
  
```

En el siguiente ejemplo de programación, hay dos mBot2 y/o CyberPi que se comunican entre sí a través de una conexión LAN utilizando estos dos bloques de código. El primer programa es la "unidad de transmisión", destinado al primer mBot2 o CyberPi. El segundo programa es el código del receptor para el segundo mBot2 o CyberPi. Puedes cargar ambas partes del código en CyberPi/mBot2 ya que el código que se encuentra debajo se asegurará de que la unidad emisora no reciba su propio mensaje.



Después de cargar los códigos de programación, debes reiniciar el mBot2 y/o CyberPi. Si mueves ahora la palanca de mando del primer mBot2 o CyberPi, el segundo mBot2 o CyberPi recibirá el mensaje enviado (valor 1).

## 2. Recrear y probar algunos ejemplos de programación de la conexión LAN

En el cuadro anterior, cada bloque de código del LAN va acompañado de un ejemplo de programación. Recrea estos ejemplos de programación en mBlock y pruébalos. Elabora tus propias ampliaciones de un ejemplo de programación.

## 3. Prueba (25 min)

### Paso 3: Prueba

Ya has aprendido mucho sobre la conexión LAN del mBot2. Ahora vas a crear tu propio programa informático en mBlock utilizando la comunicación LAN.

En esta tarea vas a programar un juego por ti mismo. Se llama "Buscando... el color azul" En este juego, un grupo de mBot2 busca el color azul en el suelo. Si uno de los mBot2 encuentra el color, informará a los demás a través de un mensaje LAN y ganará el juego.

¡Suenan complicado! Afortunadamente, no tienes que programarlo todo por ti mismo. A continuación encontrarás un ejemplo de programación. Se debe cargar el mismo programa en todos los mBot2 para que todos sigan las mismas instrucciones.

```

when button A pressed
  forever
    if quad rgb sensor 1 probe (2) R1 detected blue ? or quad rgb sensor 1 probe (3) L1 detected blue ? then
      broadcast message blue on LAN
      stop encoder motor all
      LED all displays
      play yeah until done
      stop all
    else
      if ultrasonic 2 1 distance to an object (cm) < 15 then
        moves backward at 60 RPM for 2 secs
      else
        random_moves
  
```

```

when receiving blue broadcast on LAN
  stop encoder motor all
  LED all displays
  play sad until done
  stop all
  
```

Para hacer que el mBot2 se desplace de forma aleatoria, puedes utilizar el siguiente bloque de código:

```
pick random 1 to 10
```

Este bloque de código se utiliza para seleccionar aleatoriamente un valor dentro de un cierto rango. Se puede asignar una instrucción particular a cada valor del rango. De esta forma, el mBot2 puede elegir de forma no lineal entre una serie de instrucciones. Para que tu programa sea más fácil de entender, puedes separar los bloques de esta instrucción y colocarlos debajo de un bloque personalizado, que luego puedes utilizar en tu programa principal:

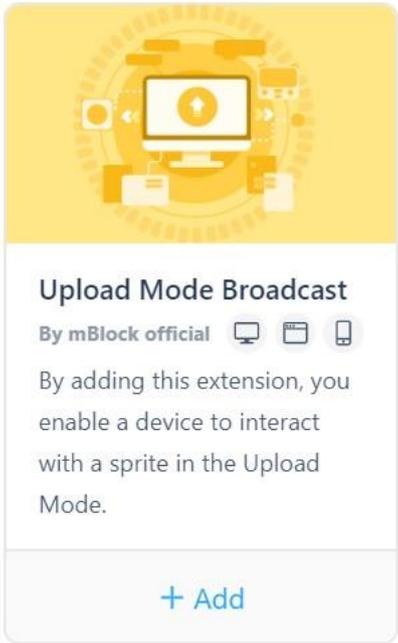
```

define random_moves
set turn to pick random 1 to 2
if turn = 1 then
  moves forward at 60 RPM for 1 secs
  turns left 120 ° until done
if turn = 2 then
  moves forward at 60 RPM for 1 secs
  turns right 120 ° until done

```

¿Solo tienes un mBot2? En ese caso, puedes simular la comunicación entre dos dispositivos incluyendo un objeto en el escenario de mBlock. El escenario es el área en la esquina superior izquierda de la pantalla de mBlock.

Tienes que añadir una nueva extensión a los bloques de código para el objeto y el mBot2, llamada "Transmisión en modo de carga". Te permite enviar y recibir mensajes entre el objeto y el robot mientras el robot está en modo Carga, similares a los mensajes enviados a través de la comunicación LAN.

	Extensión	Añadir
Para el objeto		<ul style="list-style-type: none"> <li>• Haz clic en "Objetos"</li> <li>• Haz clic en "Extensiones"</li> <li>• Haz clic en "Extensiones de objeto"</li> <li>• Busca la extensión denominada "Transmisión en modo de Carga" y añádela</li> </ul>

<p>Para el mBot2</p>		<ul style="list-style-type: none"> <li>• Haz clic en "Dispositivos"</li> <li>• Haz clic en "Extensiones"</li> <li>• Haz clic en "Extensiones de dispositivo"</li> <li>• Busca la extensión denominada "Transmisión en modo de Carga" y añádela (puede que no esté en la primera página de la Librería de Extensiones)</li> </ul>
----------------------	--	--

En este ejemplo de programación, cuando el mBot2 ha encontrado el color, enviará un mensaje al objeto panda en el escenario de mBlock, y el panda dirá que el mBot ha encontrado el color.

```

when button A pressed
  forever
    if (quad rgb sensor 1 probe (2) R1 detected blue ? or quad rgb sensor 1 probe (3) L1 detected blue ?) then
      send upload mode message blue
      stop encoder motor all
      LED all displays for 3 secs
      stop all
    else
      if (ultrasonic 2 distance to an object (cm) < 15) then
        moves backward at 60 RPM for 2 secs
      else
        random_moves
  
```

```

when receiving upload mode message blue
  say mBot2 found the color!
  play sound yes until done
  
```

Puedes copiar los ejemplos de programación anteriores, pero también puedes inventarte nuevos. Para ello es útil utilizar paso a paso el siguiente esquema. ¿Tienes idea de lo que quieres hacer? Si es así, coméntalo primero con tu profesor para ver si es factible.



	Explicación
Paso 1: ¿Qué quieres hacer?	<ul style="list-style-type: none"><li>• ¿Qué quieres programar?</li></ul>
Paso 2: ¿Qué necesitas?	<ul style="list-style-type: none"><li>• ¿Qué material necesitas para realizar esta acción?</li></ul>
Paso 3: ¿Qué bloques de código necesitas?	<ul style="list-style-type: none"><li>• ¿Qué vas a hacer para que el mBot2 se desplace y se comunique?</li><li>• ¿Qué bloques de código usarás?</li><li>• Describe brevemente cómo funciona tu programa (pseudocódigo/lenguaje natural, diagrama de flujo o Lenguaje Unificado de Modelado (UML))</li><li>• Si necesitas más explicaciones, puedes hablar con sus compañeros de clase, el profesor o investigar sobre el tema. También tienes ayuda disponible para cada bloque de códigos en mBlock.</li></ul>
Paso 4: Pruebas e implementación	<ul style="list-style-type: none"><li>• ¿Está lista la primera versión? ¡Pruébala! Durante la ronda de pruebas, anota los puntos de mejora</li><li>• Trabaja en los puntos de mejora hasta que el mBot2 haga exactamente lo que tenías en mente</li><li>• ¿Lo has conseguido? Graba el resultado final y pregúntale a tu profesor si puedes publicarlo en las redes sociales con el hashtag #mbot2inclass</li></ul>

## 4. Conclusión (5 min)

### Paso 4: Conclusión

¿Has conseguido programar el juego?

En esta lección has aprendido qué es una conexión LAN y dónde puedes encontrarla en la vida diaria. Sabes cómo configurar una conexión LAN entre diferentes dispositivos. También has aprendido a crear tu propia red LAN utilizando el mBot2 y el CyberPi. Además, sabes cómo añadir una selección aleatoria a un programa informático.

Ha llegado el momento de realizar una breve reflexión. Piensa por ti mismo y discute con el grupo:

- ¿Qué crees que ha salido bien?
- ¿Qué podría mejorarse?
- ¿Qué partes de la lección te han resultado fáciles y cuáles más difíciles?
- ¿Sobre qué te gustaría recibir más información?
- ¿Quién podría ayudarte con eso?

# Lección 8:

## El mBot2 a tu servicio

**Tema:** STEAM

**Curso(s):** A partir de 5º

**Duración:** 45 minutos

**Dificultad:** Intermedia

### ★ Objetivos de la lección

Al final de esta lección, los estudiantes serán capaces de:

- Hacer que el mBot2 lleve a cabo tareas utilizando el reconocimiento de voz y la conexión WIFI
- Combinar el reconocimiento de voz con variables para diálogos
- Enviar mensajes a través de la mensajería en la nube
- Utilizar declaraciones Boolean con datos del sensor

### ★ Resumen

El reconocimiento de voz es la tecnología que convierte la palabra hablada (pero también oraciones completas) en texto. Reconocer incluso una sola palabra simplemente por la variación en la frecuencia es una tarea bastante computacional. Y para ser práctico, el reconocimiento de voz debe funcionar independientemente de una voz en cuestión: una voz alta o baja, hablar lento o rápido... no debería conllevar ninguna diferencia. El reconocimiento de voz es parte de la interacción del usuario en la electrónica de consumo (como *smartphones*, televisores, hogares inteligentes), pero también para los sistemas de asistencia. En esta actividad, programarás un breve diálogo con el mBot2 actuando como un robot de servicio. También puedes ampliar esta tarea más adelante para simular un diálogo de usuario para otras tareas.

### 🔗 Puntos de enfoque

Al final de esta lección, los estudiantes sabrán:

- Qué es el reconocimiento de voz y cómo aplicarlo
- Cómo realizar una conexión de Internet utilizando WIFI
- Cómo utilizar el reconocimiento de voz y desarrollar un diálogo con el usuario
- Enviar mensajes a ordenadores remotos través de la mensajería en la nube

- Utilizar variables para describir un escenario en un proceso (como realizar un pedido, interacción de usuario)

## Lista de verificación previa a la lección

¿Qué necesitas?

- PC u ordenador portátil (con salida USB) con el *software* mBlock instalado, la versión web (también para ChromeBook) o una tableta con la aplicación mBlock instalada
- El mBot2 con el CyberPi
- Un cable USB-C o un adaptador (*dongle*) Bluetooth Makeblock

## Plan de la lección

Esta lección consta de cuatro pasos y tiene una duración de 45 minutos.

Duración	Contenido
5 minutos	<b>1. Calentamiento</b> <ul style="list-style-type: none"> <li>• El reconocimiento de voz en la vida diaria</li> <li>• Conocer el reconocimiento de voz con el mBot2</li> </ul>
10 minutos	<b>2. Práctica</b> <ul style="list-style-type: none"> <li>• Configurar una conexión WIFI</li> <li>• Programar y aplicar el reconocimiento de voz</li> <li>• Mensajería en la nube</li> </ul>
25 minutos	<b>3. Prueba</b> <ul style="list-style-type: none"> <li>• Programar el mBot2 para que se convierta en un camarero robot</li> </ul>
5 minutos	<b>4. Conclusión</b> <ul style="list-style-type: none"> <li>• Hora del espectáculo: muestra lo que has hecho con tu robot en una película corta y divertida para poder comentarlo más adelante</li> <li>• Si tu profesor lo permite, comparte el resultado final en las redes sociales con el hashtag #mBot2inclass</li> <li>• Reflexión: ¿De que estas más orgulloso/a? ¿Qué te gustaría mejorar de tu robot?</li> </ul>

### 1. Calentamiento (5 min)

#### Paso 1: Calentamiento

Este paso consta de dos partes:

1. El reconocimiento de voz en la vida diaria
2. Conocer el reconocimiento de voz con el mBot2

#### 1. El reconocimiento de voz en la vida diaria

El reconocimiento de voz es la tecnología que puede identificar palabras y oraciones en el sonido de la voz humana y las traduce a texto. Esta es una tarea muy computacional, ya que todos los algoritmos tienen que detectar un cambio de frecuencia a lo largo del tiempo. Pero incluso esto difiere de una persona a otra. Piensa en que existen voces bajas, altas, que hablan rápido o en los dialectos. Incluso el estado de ánimo puede cambiar el "tono" de una misma oración. La investigación de décadas en informática y lingüística permite que estas tareas complejas se incorporen cada vez más en la vida diaria.

Algunos sistemas de reconocimiento requieren una forma de "entrenamiento", donde el usuario lee textos específicos conocidos por el algoritmo de detección, adaptando un modelo informático al tono de voz individual. Otros solo identifican palabras muy específicas; se utilizan principalmente en los sistemas de diálogo de usuario. Algunos sistemas pueden reconocer y traducir textos independientemente del contexto.

El CyberPi y el mBot2 pueden utilizar el reconocimiento de voz sin necesidad de una codificación compleja. Sin embargo, dado que el resultado es "solo" texto, el programa tiene que analizarlo y procesarlo más.

El reconocimiento de voz se utiliza en productos de consumo electrónicos como televisores, *smartphones* u hogares inteligentes para ofrecer mayor comodidad al usuario: solicitar una conexión, anotar una cita en el calendario o cambiar la temperatura y la luz de la habitación con solo decirlo.

Además de la comodidad, el reconocimiento de voz puede ser de gran ayuda para las personas discapacitadas o proporcionar seguridad adicional. Por ejemplo, un conductor en un coche puede aprobar, utilizando solo la voz, un cambio de navegación debido a los inminentes atascos de tráfico, en lugar de utilizar botones o una interfaz mediante una pantalla.

En esta actividad, utilizarás algunas de las funciones del reconocimiento de voz.

## 2. Conocer el reconocimiento de voz con el mBot2

El mBot2 ofrece un reconocimiento de voz basado en la codificación de bloques: es tan sencillo como iniciar el reconocimiento durante un cierto tiempo (2, 3 o 5 segundos) utilizando un solo bloque y recuperar el texto reconocido como una cadena por otro bloque. Los bloques de codificación se explicarán a continuación.

Este reconocimiento de voz puede comprender muchos idiomas diferentes y no se limita a un pequeño conjunto de palabras predefinidas. Por lo tanto, es una tarea muy computacional. Es tan exigente que el mBot2 o el CyberPi no pueden gestionarlo por sí mismos... necesitan solicitar ayuda a través de Internet. Por eso, en primer lugar, se debe establecer una conexión a Internet y también es necesario crear una cuenta de usuario. La configuración de una cuenta de usuario genera una "clave" para iniciar la sesión en los servicios utilizados (para obtener más información sobre el uso de datos y la privacidad, consulta a continuación).

Tal y como lo hiciste en la actividad 3, el reconocimiento de voz graba el audio, y a continuación envía esta grabación a una red muy potente de ordenadores para poder procesarla. Ellos se encargan de la traducción a texto y de enviárselo de vuelta al CyberPi/mBot2. La red de ordenadores generalmente se conoce como "la nube". No es un ordenador específico, sino más bien un gran grupo compuesto de ordenadores, cambiando las tareas computacionales entre ellos.

También puedes utilizar Internet para transmitir datos entre múltiples CyberPi en diferentes redes y ubicaciones (como en la escuela y en casa). Este servicio digital también se procesa en la nube.

### Utilización de los datos y nota de privacidad:

El reconocimiento de voz requiere una conexión a Internet para transferir el audio grabado y enviar el texto transcrito. El audio grabado se procesa en servidores en Europa, la nube de Microsoft Azure, pero no se almacenará de forma permanente. Para utilizar los servicios, se necesita una "clave", que se generará automáticamente al crear una cuenta de usuario en Makeblock.com. Los bloques de codificación utilizarán esta clave internamente. No es necesario introducir esta clave manualmente en ningún lugar.

Aparte de una dirección de correo electrónico y una contraseña elegida personalmente, no es obligatorio especificar ninguna otra información personal (como nombres, imágenes, género...). La conexión a Internet requiere un acceso WIFI a través de SSID (el nombre de la red inalámbrica) y una contraseña. Los inicios de sesión basados en certificados no son compatibles. El SSID y la contraseña se almacenan con el programa en texto sin formato, por lo

que usar un punto de acceso temporal podría ser una buena idea. Idealmente, puedes utilizar un *smartphone* con una tarifa plana de datos.

## 2. Práctica (10 min)

### Paso 2: Práctica:

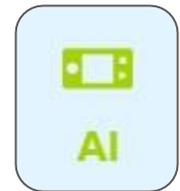
En este paso, te familiarizarás con los diferentes bloques de código de las funciones AI e IoT.

Este paso consta de tres partes:

4. Configurar una conexión WIFI
5. Programar y aplicar el reconocimiento de voz
6. Mensajería y comunicación en LAN y en la nube

### 1. Configurar una conexión WIFI

Abre mBlock y conecta el CyberPi en la pestaña 'Dispositivos'. Cambia al modo Carga. Una vez conectado, busca la categoría de bloque AI (Inteligencia Artificial) en los bloques de código y haz clic en ella. Ahora verás las funciones disponibles, incluido el reconocimiento de voz (si los bloques están en gris, significa que has olvidado hacer el cambio al modo Carga...).



#### Bloque de códigos:



Este bloque inicia los intentos de conexión a la red inalámbrica proporcionada. Necesitas un enrutador o un punto de acceso que permita a los dispositivos conectarse a la red y tener acceso a Internet. Es posible que algunas escuelas solo permitan dispositivos conocidos: cada adaptador de red (con cable o inalámbrico) tiene un número único, la dirección MAC. Si este número es desconocido para estas redes seguras, rechazan la conexión incluso si la contraseña es correcta.

La forma más fácil de evitar esto es crear un punto de acceso temporal, por ejemplo, utilizando un *smartphone* con una tarifa plana de datos que se utiliza solo para el CyberPi/mBot2 en una lección (consulta la normativa legal de tu escuela).

Cualquier bloque que siga a este bloque de código se ejecutará inmediatamente, por lo que no espera a que se establezca una conexión ("sin bloqueo"). Tienes que codificar una rutina de espera con el siguiente bloque:

#### Bloque de códigos:

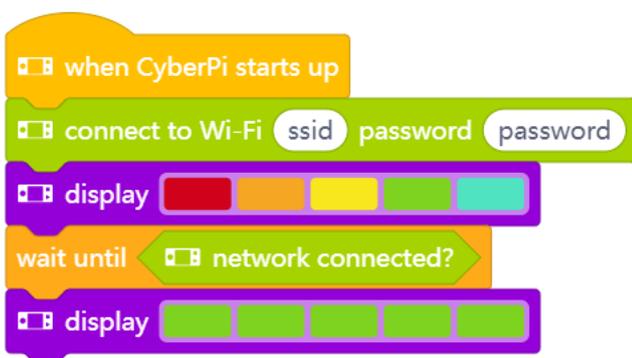


Este bloque de Boolean comprueba si se ha podido realizar la conexión a la red; de ser así, informa un Verdadero. Puedes utilizar esto junto con un bloque de espera:



Esta combinación detiene todo el código que se ejecuta en el mismo hilo hasta que se establece con éxito una conexión. Es posible que quieras hacer esto más fácil de usar en un escenario posterior, informando sobre los intentos de conexión, indicando una conexión exitosa o incluso añadiendo un tiempo de espera (como un reinicio si no se puede establecer la conexión durante 30 segundos).

Para utilizar el reconocimiento de voz, tienes que conectar el CyberPi a Internet. A continuación, podrás ver el ejemplo de programación que debes utilizar para que el CyberPi se conecte a Internet. Tan pronto como el CyberPi esté conectado a Internet, lo verás en la pantalla y los LED se volverán verdes.



Estos ejemplos utilizan un enfoque basado en hechos, por lo que un hilo del programa intenta habilitar la conexión a Internet al iniciarse. Los otros hilos inician el reconocimiento. Pero por ahora, no comprueba si la conexión a Internet puede establecerse antes de continuar, ni da ninguna indicación al usuario sobre lo que está sucediendo, excepto si se ha realizado



correctamente. Es posible que quieras cambiar esto en un escenario posterior, cuando pruebes tus propias ideas...

Asegúrate de introducir el nombre (SSID) y la contraseña de la red WIFI correctamente (la imagen muestra un ejemplo). Ambos distinguen entre mayúsculas y minúsculas. Conecta el mBot2 a tu ordenador con un cable USB y activa el modo Carga. Recrea el código de programación de arriba y prueba si el código funciona. Te aconsejamos hacerlo con un punto de acceso desde tu teléfono móvil, ya que una red escolar a menudo sólo permite la conexión a dispositivos conocidos (direcciones MAC de la lista blanca). Asegúrate de que la tarifa de datos cubre el uso de Internet. Ahora conecta el CyberPi a tu ordenador.

Has conectado el mBot2 a Internet. Ahora puedes comunicarte con el mBot2 a través de la nube. Esto es lo que se llama Internet de las Cosas (IoT).

Si estás utilizando un CyberPi diferente para mostrar el estado (como el pedido enviado a la cocina o el proceso de entrega), tiene que estar conectado a la misma red WIFI. Esto es para asegurarse de que ambos dispositivos (mBot2-Cyberpi y CyberPi autónomo) están utilizando el mismo canal en la red WIFI. El router/punto de acceso decide qué canales utilizar. Esto garantiza que todos los dispositivos están en el mismo canal para el intercambio de datos. También puedes consultar la comunicación LAN en la lección 7.

## 2. Programar y aplicar el reconocimiento de voz

Ahora vas a programar y probar el reconocimiento de voz. Vas a complementar el código de programación en el que el mBot2 se conecta al WIFI con un código para reconocimiento de voz.

### Bloque de códigos:



Este bloque de código se utilizará en un ejemplo a continuación; hace lo contrario del reconocimiento de voz. Con este bloque puedes hacer que CyberPi/mBot2 lea en voz alta un texto entregado. Aquí, el texto se transfiere al servicio en la nube y el archivo de audio sintetizado se envía de vuelta. Tiene una detección automática del idioma utilizado, por lo que el idioma hablado debe ser el mismo que el utilizado en el texto.

### Bloque de códigos:

 recognize (4) English ▼ 3 secs

El reconocimiento de voz está disponible en 12 idiomas diferentes. Si tu idioma nativo no es compatible, puedes utilizar el inglés. Este bloque de código inicia una grabación de audio para un tiempo específico y lo envía al servicio en la nube para su detección. Para recibir los resultados, necesitas el siguiente bloque de código:

### Bloque de códigos:

 speech recognition result

Se puede acceder a los resultados del reconocimiento de voz a través de este bloque de envío de información. Puedes guardar el resultado en una variable, por si necesitas reutilizarlo, o mostrarlo en la pantalla o procesarlo posteriormente con algún análisis (como "¿contiene el texto la palabra 'sí'?").

### Bloque de códigos:

 translate hello into Chinese ▼

Las características de la Inteligencia Artificial (IA) permiten incluso la traducción de textos a otros idiomas. Este bloque de código es un bloque de envío de información que permite traducir el texto proporcionado a un idioma (compatible) que elijas. Podrías utilizar este bloque junto con los anteriores y los conocimientos adquiridos en las actividades previas y construir un traductor de voz: pulsa un botón y traduce 5 segundos de discurso a un idioma extranjero... Incluso podrías probar diferentes idiomas y poner a prueba tus conocimientos de idiomas extranjeros. 😊

En el siguiente ejemplo de programación, el mBot2 está programado para que al pulsar el botón B, el mBot2 reconozca la voz durante 3 segundos. El resultado se muestra en la pantalla del CyberPi y el reconocimiento es hablado. Recrea el siguiente código de programación y pruébalo. Comprueba lo que has dicho y lo que se ha "entendido" y traducido.  
¿Funciona? ¡Mira qué más puedes hacer con el reconocimiento de voz!

```

when CyberPi starts up
  connect to Wi-Fi ssid password password
  display [red][orange][yellow][green][cyan]
  wait until [network connected?]
  display [green][green][green][green][green]

```

```

when button B pressed
  recognize (4) English 3 secs
  show label 1 [speech recognition result] at center of screen by middle pixel
  speak auto [speech recognition result]

```

También puedes utilizar el reconocimiento de voz para controlar el mBot2 con la voz. Puedes asignar ciertas instrucciones a determinados resultados del reconocimiento de voz, pero ten en cuenta lo específico que eres con los resultados. Por ejemplo, puedes crear la instrucción para que solo reconozca la palabra "sí" cuando se pulse el botón B. Pero, ¿qué ocurre si dices "definitivamente sí" o "sí, por supuesto"? Una coincidencia directa con...

```

if [speech recognition result] = "yes" then

```

... no funcionaría... El resultado del reconocimiento sería diferente (la cadena "sí" no es igual a la cadena "definitivamente sí"). Para permitir un conjunto más amplio de respuestas que signifiquen todas "sí", puede comprobar simplemente si la cadena de respuesta contiene la palabra "sí":

```

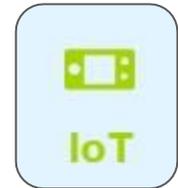
if [speech recognition result] contains yes ? then

```

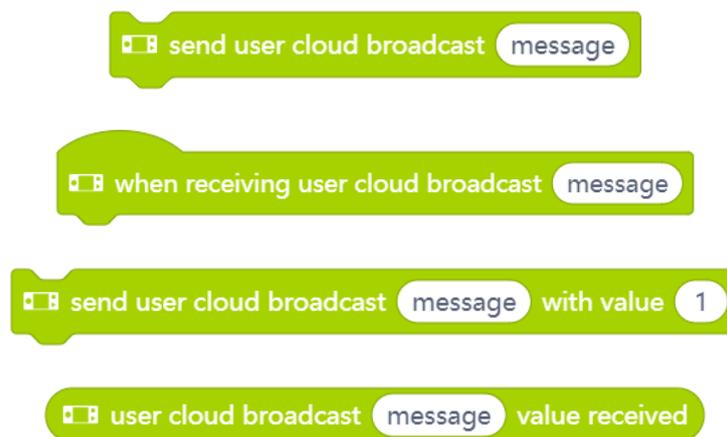
### 3. Mensajería y comunicación en LAN y en la nube

También puedes utilizar el resultado del reconocimiento de voz para interactuar con varios mBot2 o CyberPi. En la lección 7 has aprendido sobre la conexión LAN y cómo configurarla.

Además de enviar un mensaje en la LAN, que requiere que los dispositivos sean "locales" entre sí, puedes utilizar otro servicio en la nube para enviar mensajes a los dispositivos en lugares remotos, siempre y cuando tengan una conexión a Internet. Estos bloques se encuentran en la categoría de bloques IoT (Internet de las Cosas).



Estos bloques funcionan de forma idéntica a los que se encargan de la comunicación por LAN:



Para asegurarte de que la parte remota solo puede comunicarse con los dispositivos que quieres, necesitas una cuenta de usuario Makeblock. Con la creación de la cuenta se generará una "clave" para la comunicación en la nube. Esta clave está oculta dentro de los bloques de codificación y es única para cada usuario. Para que 2 dispositivos remotos se comuniquen a través de Internet, ambos tienen que estar programados desde la misma cuenta de usuario. Los bloques de difusión en la nube tienen la ventaja de poder comunicarse en todo el mundo, pero no son útiles para una comunicación rápida: el mensaje puede tardar unos segundos en llegar. Las transmisiones LAN, en cambio, son rápidas, pero funcionan solo a nivel local. A continuación, puedes ver un ejemplo de programación que resume lo anterior. Recrea el programa primero y prueba si todo funciona con el reconocimiento y la comunicación entre el CyberPi y otro mBot2. Para este ejercicio, necesitas cargar el mismo código de programa en el modo Carga a ambos dispositivos.

```

when CyberPi starts up
  connect to Wi-Fi ssid password password
  display [red][orange][yellow][green][cyan]
  wait until [network connected?]
  display [green][green][green][green][green]
  show label 1 Shall the robot drive? at center of screen by middle pixel
  show label 2 press B to answer at middle at bottom by middle pixel

```

```

when button B pressed
  recognize (4) English 3 secs
  show label 1 [speech recognition result] at center of screen by middle pixel
  if [speech recognition result contains yes?] then
    broadcast message yes on LAN

```

```

when receiving yes broadcast on LAN
  moves forward 10 cm until done

```

Después de asegurarte de que todo funciona bien, puedes modificar y ampliar el ejemplo. ¿Tal vez con comandos direccionales? Piensa en la lección 1 y en el ejercicio de desplazar el mBot2 manualmente a través de un laberinto... ¿Puedes hacer lo mismo con comandos de voz?

## 3. Prueba [25 minutos]

### Paso 3: Prueba

Ahora ya sabes cómo crear una conexión WIFI, programar el reconocimiento de voz y desplegar la mensajería en la nube o en la LAN. Ha llegado el momento de empezar por ti mismo. Vas a programar un robot de servicio amable: un robot camarero. En algunos países ya hay restaurantes en los que un robot viene a anotar el pedido y trae la comida de la cocina. Observa el siguiente vídeo: <https://www.youtube.com/watch?v=FFCPKmLAZb4>

El robot camarero que vas a fabricar debe ser capaz de hacer una serie de cosas:

- Desplazarse en línea recta
- Detenerse en una zona roja (tu mesa), anotar un pedido y enviarlo a la cocina
- Opción adicional: si se dispone de un segundo mBot2 o CyberPi, entonces mostrar los pedidos a la cocina (para que puedan empezar a preparar la comida)
- Continuar desplazándose hasta que el mBot2 encuentre una superficie verde (la cocina) y se detenga para recoger la comida de la cocina
- Opcional: si se dispone de un segundo mBot2 o CyberPi, entonces notificar se está entregando la comida

La secuencia de inicio es casi idéntica a la de los ejemplos anteriores. Recuerda introducir el SSID y la contraseña correctas:

```

when CyberPi starts up
  connect to Wi-Fi ssid password password
  display [red][orange][yellow][green][cyan]
  wait until [network connected?]
  display [green][green][green][green][green]
  show Waiterbot at center of screen by middle pixel
  wait 1 seconds
  clear screen

```

El siguiente bloque de código se programa en el mismo campo de *script* de comandos del ejemplo anterior. En este bloque de código, indicas al mBot2 que comience a desplazarse y que se detenga en una marca roja para anotar el pedido. Después, el mBot2 sigue desplazándose hasta una marca verde. Los marcadores de colores indican la mesa y la cocina. Para empezar de forma fácil, asumimos que están en línea recta. Si todo funciona correctamente, puedes ampliar el ejercicio aún más e incluir los conocimientos que aprendiste en lecciones anteriores. Haz que el robot siga una línea hasta la mesa (el mapa entregado tiene marcadores rojos y verdes en la pista que se pueden usar para este propósito). La lección 5 explica todo sobre el seguimiento de línea...

Como un ejercicio extra al reconocimiento de voz, permaneciendo en la línea recta entre la mesa y la cocina, puedes utilizar la mensajería en la nube o LAN para enviar el pedido a otro mBot2 o CyberPi para que lo muestre.

```

when joystick middle pressed
  clear screen
  moves forward at 50 RPM
  wait until [quad rgb sensor 1 probe (3) L1 detected red ?] or [quad rgb sensor 1 probe (2) R1 detected red ?]
  stop encoder motor all
  get_order
  moves forward at 50 RPM
  wait until [quad rgb sensor 1 probe (3) L1 detected green ?] or [quad rgb sensor 1 probe (2) R1 detected green ?]
  stop encoder motor all
  broadcast message food in delivery on LAN

```

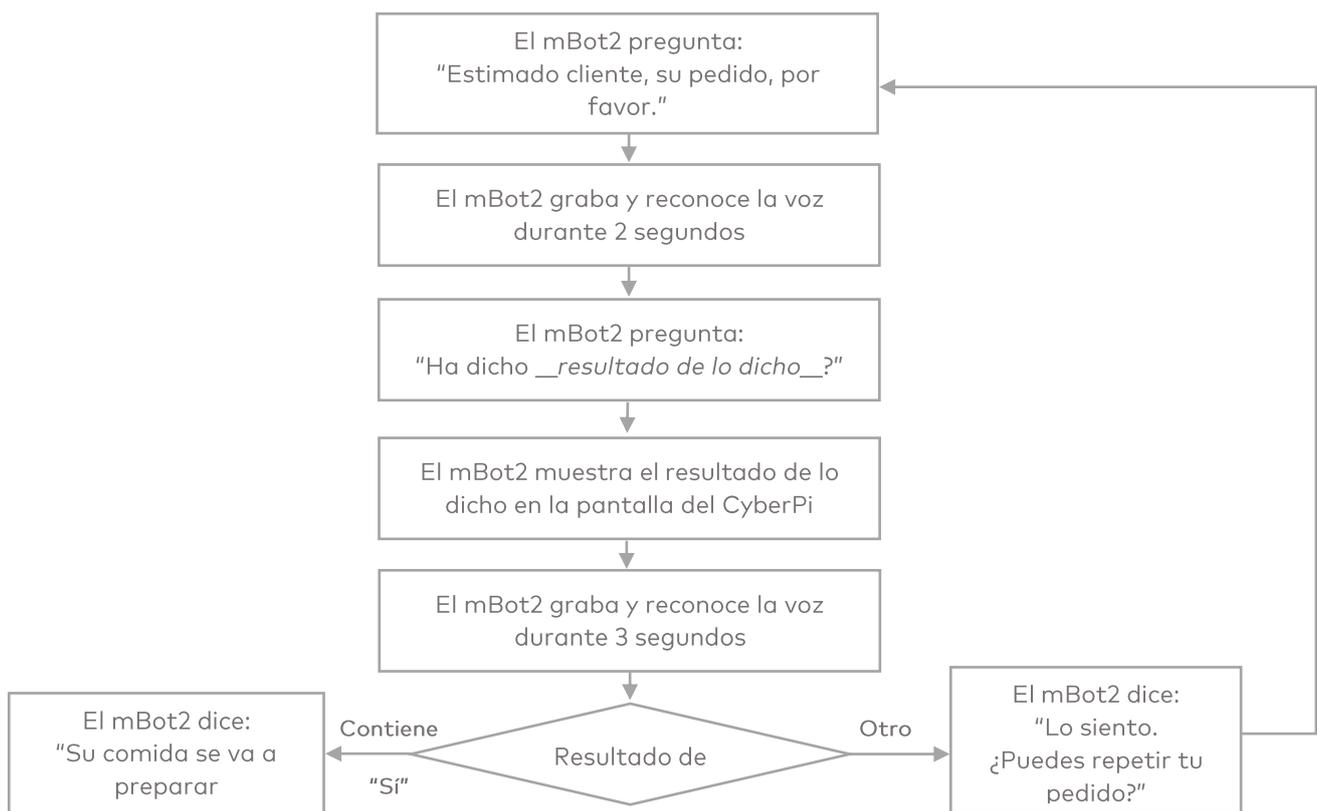
Tienes que crear tú mismo el bloque "get\_order" en "Mis bloques". Esto permite realizar un resumen general del programa principal. Se trata de desplazarse hasta los marcadores de colores recuperando el pedido en el proceso.

Para definir realmente lo que significa "get\_order", pensemos primero en un proceso de pedido:

1. Pregunta por el pedido
2. Reconoce la respuesta
3. Verifica que el pedido que se ha entendido es correcto
4. En caso afirmativo, continúa (envía el pedido a la cocina y finaliza la secuencia del pedido)

En caso negativo, pregunta de nuevo.

La verificación es algo sensato para evitar pedidos incorrectos. Para traducir estos pasos de forma estructurada, puedes utilizar un diagrama de flujo como el siguiente:



A partir de aquí, es mucho más fácil crear los bloques de código necesarios. Empiezas con el bloque de código "get\_order". Creas de nuevo este bloque de códigos en 'Mis bloques'.

```

define get_order
  set order_status to 0
  LED all displays green
  speak auto Dear Guest! Your orders please?
  repeat until order_status > 0
    LED all displays red
    recognize (4) English 2 secs
    turn off LED all
    set order to speech recognition result
    show order at center of screen by middle pixel
    LED all displays green
    speak auto join Did you say order
    LED all displays red
    recognize (4) English 1 secs
    turn off LED all
    show speech recognition result at center of screen by middle pixel
    if speech recognition result contains yes ? then
      set order_status to 1
      speak auto Your meal will be prepared immediately.
      broadcast message food with value order on LAN
    else
      speak auto Sorry. Please repeat your order
  
```

Además, hay que crear dos variables. En una variable guardaremos el resultado del reconocimiento del pedido, de modo que si todo se ha entendido correctamente, el pedido se envíe a la cocina. (Atención: comprobar con sí/no: la respuesta a esta pregunta anularía el resultado del reconocimiento anterior en el bloque de informe. Es por ello que tenemos que guardar el pedido reconocido en una variable).

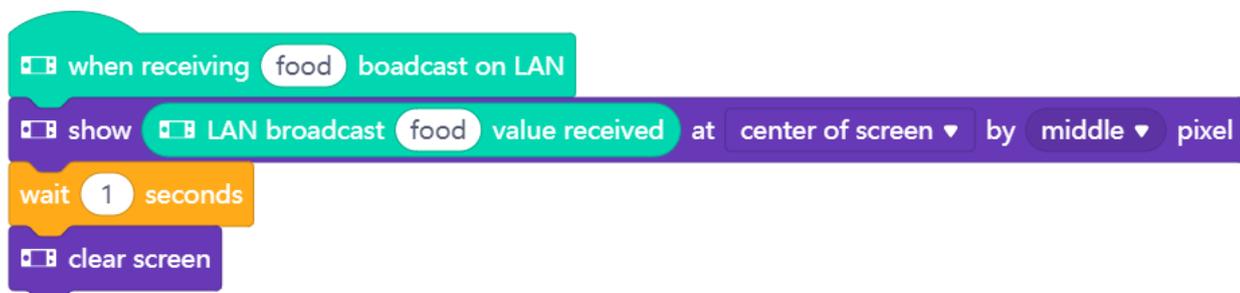
La segunda variable mantiene el seguimiento de todo el proceso: ¿el pedido está completo (es decir, se ha entendido correctamente)? Si no es así, el proceso de pedido volverá a empezar

desde el principio. Este segundo uso de la variable informa sobre los datos del proceso (escenario).

Puedes utilizar los conocimientos de las lecciones anteriores para conseguir ayuda con el programa. Por ejemplo, encender los LEDs para que el usuario entienda que el robot está a punto de hablar (verde) o escuchar (rojo), o puedes utilizar la pantalla del camarero-mBot2 para mostrar los datos. Utiliza esto en los puntos de control de tu programa, para evaluar si tu código funciona y proporcionar comentarios a los usuarios.

Como un ejercicio extra, puedes utilizar la mensajería en la nube o LAN para enviar datos a otro mBot2. Para obtener una mejor experiencia de aprendizaje, encontrarás ejemplos sobre cómo resolver tareas específicas en esta lección. Puede consultarlos si tienes problemas con tu propio código u para obtener ideas sobre cómo gestionar una situación específica. En general, es mejor hacer que un borrador funcione y perfeccionar el código en repeticiones en vez de intentar que todo esté listo y programado perfectamente de una vez. Empieza con la secuencia de conducción y en lugar de introducir el bloque "get\_order", simplemente haz que suene un pitido en ese momento para mostrar que todo está funcionando hasta el momento. A continuación, trabaja en el proceso de pedido, pero sigue desplazándote. De esta forma puedes repetir más rápido sin que el robot tenga que desplazarse entre los marcadores cada vez que quieras probar tu código. En un escenario final, combina los bloques para resolver la tarea que se te ha planteado.

En los ejemplos de programación anteriores, también enviaste un mensaje a un segundo mBot2 como ejercicio adicional. Por ejemplo, en el "proceso de pedido" has dicho que una vez que el cliente confirme el pedido, el mBot2 envía un mensaje mediante la nube o LAN a un segundo mBot2 o CyberPi (para mostrar los pedidos entrantes en la cocina). Ahora tienes que programar el segundo mBot2 para recibir el mensaje.





En el siguiente ejemplo de programación, la cocina recibe el pedido y lo muestra. Crea el ejemplo de programación siguiente en el mismo *script*. Colócalo al lado del primer ejemplo de programación. Así mantienes el *script* claro.

En el ejemplo de programación "controlar un robot", has programado que tras detectar la zona verde se envíe un mensaje de la nube al cliente para informarle de que su pedido está en camino. Esto también está programado en el mismo campo de *script* junto a los otros bloques. Podrías utilizar otro CyberPi (como un dispositivo inteligente en la mesa del cliente) para informar sobre el estado del pedido:

```
when receiving food in delivery broadcast on LAN
  play LED animation rainbow until done
  play yummy until done
  turn off LED all
```

Cuando hayas programado todo, tu campo de *script* será así:

```
when CyberPi starts up
  connect to Wi-Fi ssid password password
  display [red, orange, yellow, green, blue]
  wait until network connected?
  display [green, green, green, green, green]
  show Waiterbot at center of screen by middle pixel
  wait 1 seconds
  clear screen

when receiving food broadcast on LAN
  show LAN broadcast food value received at center of screen by middle pixel
  wait 1 seconds
  clear screen

when receiving food in delivery broadcast on LAN
  play LED animation rainbow until done
  play yummy until done
  turn off LED all

when joystick middle pressed
  clear screen
  moves forward at 50 RPM
  wait until quad rgb sensor 1 probe (3) L1 detected red ? or quad rgb sensor 1 probe (2) R1 detected red ?
  stop encoder motor all
  get_order
  moves forward at 50 RPM
  wait until quad rgb sensor 1 probe (3) L1 detected green ? or quad rgb sensor 1 probe (2) R1 detected green ?
  stop encoder motor all
  broadcast message food in delivery on LAN
```

```

define get_order
  set order_status to 0
  LED all displays
  speak auto Dear Guest! Your orders please?
  repeat until order_status > 0
  LED all displays
  recognize (4) English 2 secs
  turn off LED all
  set order to speech recognition result
  show order at center of screen by middle pixel
  LED all displays
  speak auto join Did you say order
  LED all displays
  recognize (4) English 1 secs
  turn off LED all
  show speech recognition result at center of screen by middle pixel
  if speech recognition result contains yes ? then
  set order_status to 1
  speak auto Your meal will be prepared immediately.
  broadcast message food with value order on LAN
  else
  speak auto Sorry. Please repeat your order
  
```

Tienes que cargar este código en todos los mBot2 y/o CyberPi que quieras utilizar. Puede ser el mismo código porque se conectan automáticamente a través de la conexión LAN o de la nube, y el que envía el mensaje no tendrá efecto sobre él.

Crea una línea recta para que el mBot2 se desplazca a lo largo de ella y asegúrate de que hay un marcador rojo y verde amplios. Carga el código y prueba si el robot camarero procesa los pedidos de los clientes.

Por supuesto, el ejercicio siempre se puede mejorar. Ajusta el código donde creas que puede ser mejor o quedar más bonito. ¿Quizás quieras empezar con un seguidor de línea que anote los pedidos? ¿O ampliar este ejercicio y tener un mBot2 camarero que anote los pedidos y un mBot2 repartidor que traiga la comida? Puedes colaborar con otros equipos de tu clase e integrar sus robots...

Discute en clase posibles ampliaciones como estas cuando revises los progresos realizados hasta el momento y crea un caso de uso con múltiples robots. No es necesario que tenga lugar en un restaurante...

## 4. Conclusión (5 min)

### Paso 4: Conclusión

¿Has conseguido programar el camarero-bot?

En esta lección, has aprendido sobre el reconocimiento de voz y la síntesis de voz y cómo aplicarlos a través de un servicio en la nube. Has conseguido conectar el mBot2 a Internet y así enviar datos de forma local o incluso remota.

La creación de un diálogo (como para pedir una comida) ha sido parte del ejercicio, y has aprendido a filtrar la respuesta de texto por palabras de activación específicas para tener una conversación más "natural" (sin esperar que el usuario responda solo con respuestas específicas).

Ha llegado el momento de realizar una breve reflexión. Piensa por ti mismo y discute con el grupo:

- ¿Qué crees que ha salido bien?
- ¿Qué podría mejorarse?
- ¿Qué partes de la lección te han resultado fáciles y cuáles más difíciles?
- ¿Sobre qué te gustaría recibir más información?
- ¿Quién podría ayudarte con eso?

## Lección 9: El mBot2 en la naturaleza

**Tema:** STEAM

**Curso(s):** A partir de 5°

**Duración:** 45 minutos

**Dificultad:** principiante

### ★ Objetivos de la lección

Al final de esta lección, los estudiantes serán capaces de:

- Integrar el Aprendizaje Automático (Machine Learning) en la programación del mBot2
- Hacer que varias características técnicas del mBot2 funcionen juntas
- Establecer una comunicación entre el robot y el ordenador, aunque el robot funcione de forma independiente en modo Carga para la interacción entre el *hardware* físico (robot) y el escenario del objeto del *software* mBlock

### ★ Resumen

El Aprendizaje Automático (Machine Learning) es una forma débil de inteligencia artificial (IA). La Inteligencia Artificial se refiere a dispositivos o máquinas que imitan la inteligencia humana. Basándose en los datos recopilados, el Aprendizaje Automático permite a los sistemas informáticos o robóticos elegir acciones basadas en el historial de datos y la probabilidad sin necesidad de formular explícitamente un algoritmo para ello.

### 🔗 Puntos de enfoque

Al final de la lección, los estudiantes conocerán:

- Qué es el Aprendizaje Automático (Machine Learning) y cómo aplicarlo,
- Cómo hacer que el mBot2 funcione basándose en condiciones,
- Cómo funcionan las diferentes características técnicas del mBot2 juntas en un ecosistema,
- Establecer un protocolo de comunicación entre el robot que ejecuta programas de forma independiente y el ordenador

## Lista de verificación previa a la lección

¿Qué necesitas?

- PC u ordenador portátil (con salida USB) con el *software* de mBlock instalado, la versión web (también para Chromebook)
- El mBot2 con el CyberPi
- Un cable USB-C o un adaptador (*dongle*) Bluetooth Makeblock
- Una cámara web, en el portátil o externa
- Una impresora (opcional) o papel, lápices y rotuladores para dibujar o un *smartphone* para mostrar fotos

## Plan de la lección

Esta lección consta de cuatro pasos y tiene una duración de 45 minutos.

Duración	Contenido
5 minutos	<b>1. Calentamiento</b> <ul style="list-style-type: none"><li>• Aprendizaje Automático en la vida diaria</li><li>• Conocer el Aprendizaje Automático</li></ul>
25 minutos	<b>2. Práctica</b> <ul style="list-style-type: none"><li>• Aplicar el Aprendizaje Automático</li><li>• El Aprendizaje Automático combinado con el mBot2 en mBlock</li><li>• Establecer un protocolo de comunicación con los programas en ejecución internos</li></ul>
10 minutos	<b>3. Prueba</b> <ul style="list-style-type: none"><li>• Tarea final: el ecosistema de un ratón</li></ul>
5 minutos	<b>4. Conclusión</b> <ul style="list-style-type: none"><li>• Hora del espectáculo: muestra tu ecosistema en una película corta y divertida para poder comentarlo más adelante.</li><li>• Si tu profesor lo permite, comparte el resultado final en las redes sociales con el hashtag #mBot2inclass</li><li>• Reflexión: ¿De que estas más orgulloso/a? ¿Qué te gustaría mejorar de tu robot?</li></ul>

## 1. Calentamiento (5 min)

### Paso 1: Calentamiento

Este paso consta de tres partes:

1. Aprendizaje Automático en la vida diaria
2. Conocer la extensión del Aprendizaje Automático en mBlock
3. Aplicar el Aprendizaje Automático

#### 1. Aprendizaje Automático en la vida diaria

El Aprendizaje Automático (Machine Learning) funciona sobre una base estadística: se proporcionan datos de entrenamiento al algoritmo, como imágenes de diferentes objetos con sus descripciones. A continuación, la fase de aprendizaje intenta distinguir mejor entre estas imágenes y hacer unir cada imagen a la categoría correcta. Después de esta fase, los patrones aprendidos se prueban con nuevos conjuntos y se puede reforzar el aprendizaje. La diferencia con los programas de clasificación "clásicos" es que no hay una indicación clara de antemano sobre cómo decidir entre las diferentes categorías, como la diferenciación entre colores.

Al "aprender de los datos pasados", el programa informático puede adaptarse mejor a situaciones que cambian. Piensa en el reconocimiento facial en tu teléfono: las primeras veces después de configurarlo, es más difícil que el teléfono te identifique correctamente, pero con el tiempo, el algoritmo "aprende" a identificar tu rostro también en diferentes entornos. El Aprendizaje Automático se usa ampliamente en la clasificación de imágenes (por ejemplo, en las redes sociales) para analizar automáticamente qué muestran las imágenes cargadas: un gato, un perro, muebles, personas, etc.

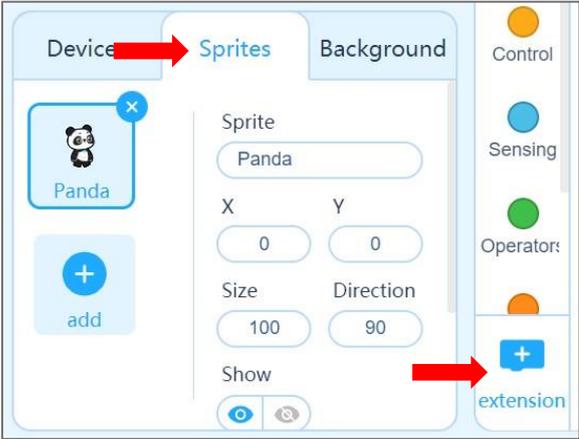
El Aprendizaje Automático se considera como "Inteligencia Artificial débil", ya que el programa informático no es consciente de sí mismo ni puede "pensar" como nosotros, pero su comportamiento actual está influenciado por los datos recopilados y analizados.



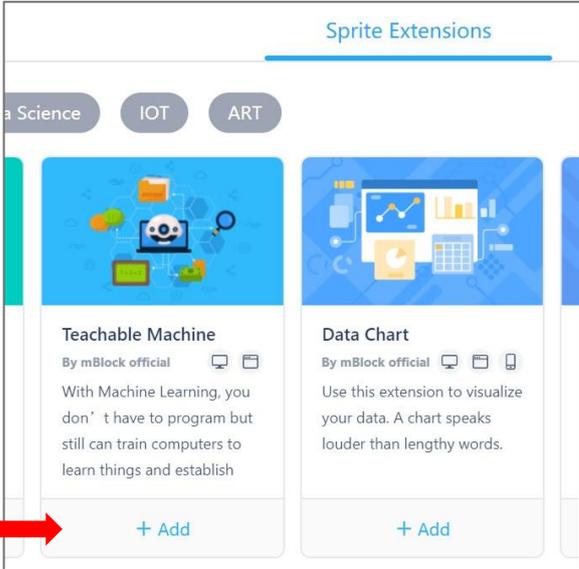
## 2. Conocer la extensión del Aprendizaje Automático en mBlock

En mBlock hay una herramienta de Aprendizaje Automático. Es necesario añadir esta herramienta desde la biblioteca de extensiones. Sigue los siguientes pasos:

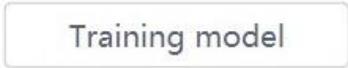
- Haz clic en "Objetos".
- Haz clic en "extensiones" (en la parte inferior, en el centro de la pantalla).



- Ahora entrarás en la biblioteca de extensiones. En la parte superior de la pantalla, haz clic en "Extensiones de objeto".
- Busca la extensión "Maquina Enseñable" y añádela.



- La extensión se añadirá ahora a los bloques de código. Puedes reconocer la extensión por la abreviatura TM en verde.
- Asegúrate de tener una cámara web conectada.
- Haz clic en el botón "Modelo de entrenamiento".



La extensión consiste en un modelo de entrenamiento en el que se puede "enseñar" a diferenciar entre conjuntos de imágenes tomadas con una cámara web. El modelo básico que ves consta de 3 categorías. Si quieres más, haz clic en "crear un nuevo modelo" y elige el número de categorías.

Consejos:

- Piensa de antemano en el número de categorías y en el reconocimiento. No puedes añadir una categoría más tarde, tendrás que empezar de nuevo desde el principio.
- Elige 1 categoría adicional para utilizarla como "fondo". Así, si quieres reconocer 2 animales diferentes utilizando las fotos que sostienes delante de la cámara web, elige 3 categorías: La tercera será aquella en la que no aparezca ninguna foto de un animal. De lo contrario, "obligas" al modelo a decidir, por ejemplo, entre un gato y un perro, aunque la cámara web muestre el fondo vacío.
- El fondo debe ser generalmente simple y estático (por ejemplo, evitar: aula con alumnos, cámara web mirando a los alumnos, etc.) con un claro contraste de los objetos que se deben reconocer.

Las imágenes se analizarán solo en el ordenador y el resultado se puede guardar. ¡Las imágenes en sí no se guardarán! Por lo tanto, si guardas el archivo del programa en mBlock y lo vuelves a abrir más tarde, no verás las imágenes originales con las que se ha entrenado el modelo. Siempre se pueden añadir nuevas imágenes a las categorías existentes, pero no se pueden eliminar las individuales (solo se podrán borrar todas las imágenes y aprendizajes de una categoría).

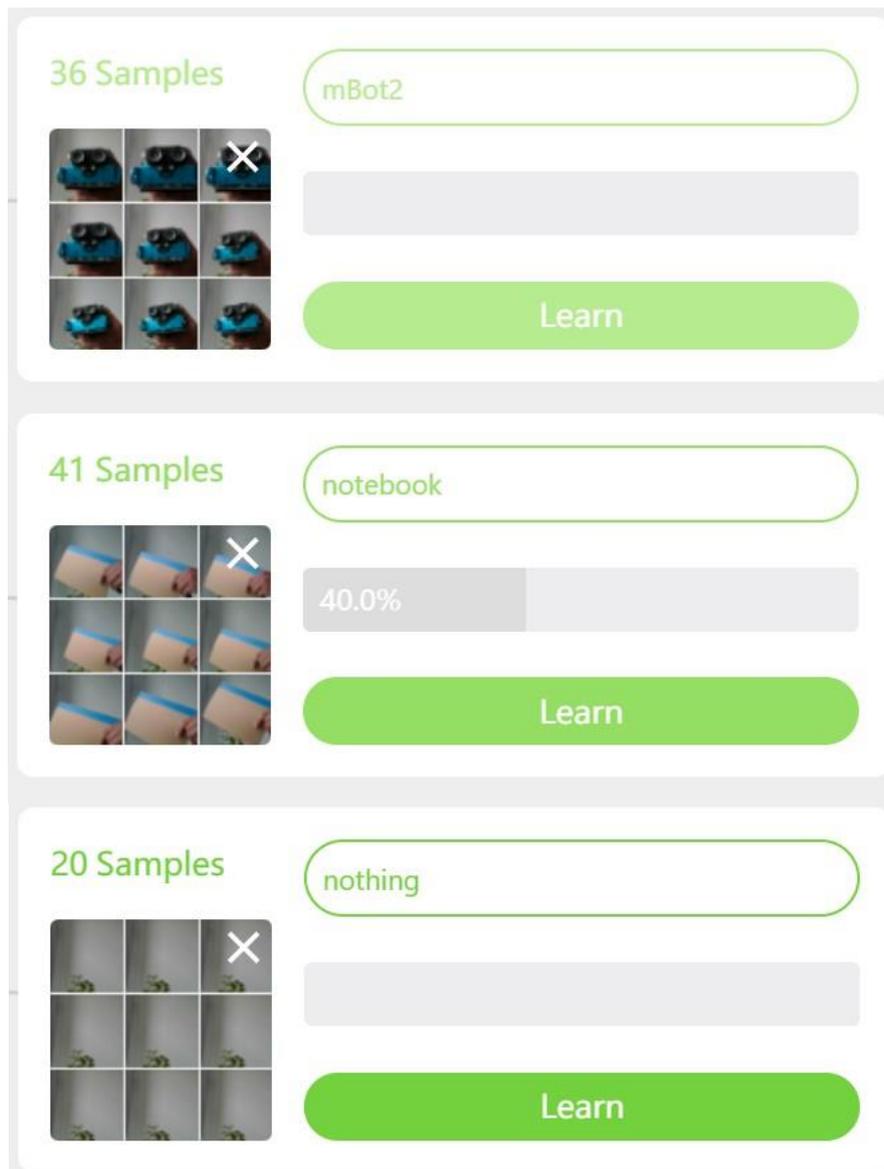
Puedes "enseñar" directamente las categorías utilizando imágenes tomadas con la cámara.

Para este ejercicio utilizarás las 3 categorías estándar:

1. Puedes empezar con el mBot2. Colócalo en el escritorio y alinea la cámara para obtener una buena vista, o mantenla firme en tu mano si utilizas un ordenador portátil con cámara incorporada. A continuación, haz clic en el botón "Aprender". Mantén el botón pulsado hasta que la cámara haya grabado al menos 20 muestras. Encima del mosaico de muestras, puedes ver el número de muestras guardadas para esta categoría. Dale un nombre a esta categoría, por ejemplo "mBot2".
2. Ahora elige otra cosa, por ejemplo, un cuaderno, colócalo frente a la cámara y haz clic en el botón "Aprender". Mantén el botón pulsado hasta que la cámara haya grabado al

menos 20 muestras. Dale a esta categoría un nombre diferente, por ejemplo "cuaderno".

3. Ahora graba el "fondo". Esto se debería utilizar siempre que ninguno de los otros objetos se presente a la cámara. Mantén pulsado el botón "Aprender" hasta que la cámara haya grabado al menos 20 muestras. Dale a esta categoría un nombre diferente, por ejemplo "fondo".



Ahora has "entrenado" al Aprendizaje Automático con diferentes objetos (mBot2, cuaderno y fondo) que el algoritmo reconocerá en la imagen de la cámara. Para cada categoría podrás ver una barra que mide un porcentaje. Esto representa la confianza que el algoritmo tiene en el resultado del reconocimiento. Puedes aumentar el porcentaje de confianza aumentando el número de muestras y haciendo algunas pequeñas variaciones, como cambiar el ángulo del objeto. Esto ayudará a la cámara a reconocer los objetos aún mejor y más rápido.

Puedes probar el reconocimiento en tiempo real sin necesidad de utilizar ninguna codificación en este escenario. Añade imágenes hasta que el reconocimiento sea rápido y preciso (sin cambiar entre los resultados y mostrando un alto porcentaje > 90 %). ¿Estás satisfecho? Entonces haz clic en "Utilizar el modelo". Ahora volverás a la pantalla para codificar a nivel de objeto.

Ahora hay tres nuevos bloques de código. Un bloque informará el resultado del reconocimiento (tal como lo has escrito), el segundo es la confianza para una categoría determinada y el tercero es el bloque Boolean para declaraciones condicionales. Te permite elegir la categoría mediante un menú desplegable. Si haces clic en uno de los bloques de código, aparecerá una nueva ventana. Esta es la ventana de reconocimiento que muestra la imagen de la cámara y el resultado del reconocimiento. Ahora tu cámara web funciona como un sensor inteligente, identificando diferentes objetos o imágenes.



## 2. Práctica (25 min)

### Paso 2: Práctica

Este paso consta de tres partes:

1. Aplicar el Aprendizaje Automático
2. El Aprendizaje Automático combinado con el mBot2
3. Establecer una conexión LAN

#### 1. Aplicar el Aprendizaje Automático

Has "enseñado" al modelo de Aprendizaje Automático dos objetos diferentes, el mBot2 y un cuaderno, además del fondo. Ahora puedes utilizar el resultado del reconocimiento como



cualquier otro bloque de sensor en tu código de programa. Puedes utilizar declaraciones condicionales para diferenciar entre los objetos y asignar una determinada acción por separado a cada resultado.

Ahora vas a intentar hacer que el objeto (en este caso el panda) hable cuando la cámara web reconozca un objeto aprendido.

A la derecha verás un ejemplo de programación. Reproduce el ejemplo de programación y mira lo que hace el panda cuando la cámara web reconoce tus objetos aprendidos.

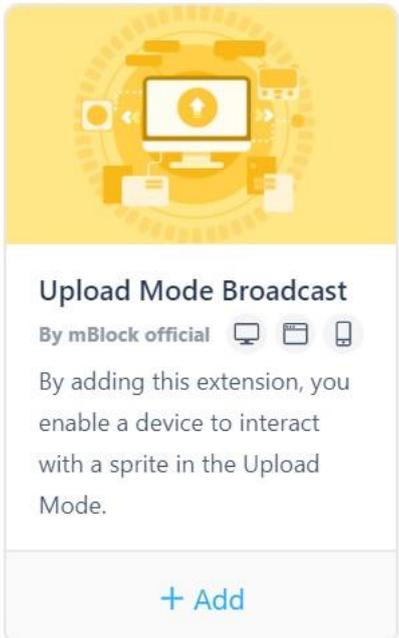
¿Ha salido correctamente? Luego investiga qué otras posibilidades hay. ¿Puedes hacer que el panda salte o camine?

```
when this sprite clicked
  forever loop
    if recognition result is mBot2 then
      say Hello!
    if recognition result is notebook then
      say time to write!
    if recognition result is nothing then
      say ...
```

### 2. El Aprendizaje Automático combinado con el mBot2

Ahora ya sabes cómo utilizar a extensión de Aprendizaje Automático en el reconocimiento de imágenes y controlar un objeto, por ejemplo el panda. Ahora vas a aprender a comunicarte desde el ordenador hasta el mBot2 ejecutando su propio programa. Para ello, el mBot2 tiene que estar "conectado" al ordenador, a través de un cable USB o del Makeblock Bluetooth Dongle. El adaptador (*dongle*) permite que el robot se mueva independientemente del ordenador. Si utilizas un cable USB, asegúrate de realizar únicamente pequeños movimientos del mBot2, como pequeños giros y distancias.

Con esta conexión, el objeto (en este caso el panda) puede enviar datos al mBot2. El mBot2 recibe estos datos y el programa que se está ejecutando en el robot puede reaccionar a ellos. Tienes que añadir una nueva extensión a los bloques de código para el objeto y el mBot2, llamada "Transmisión en modo de carga".

	Extensión	Añadir
Para el objeto		<ul style="list-style-type: none"> <li>• Haz clic en "Objetos"</li> <li>• Haz clic en "Extensiones"</li> <li>• Haz clic en "Extensiones de objeto"</li> <li>• Busca la extensión llamada "Transmisión en modo de Carga" y añádela</li> </ul>
Para el mBot2		<ul style="list-style-type: none"> <li>• Haz clic en "Dispositivos"</li> <li>• Haz clic en "Extensiones"</li> <li>• Haz clic en "Extensiones de dispositivo"</li> <li>• Busca la extensión denominada "Transmisión en modo de Carga" y añádela (puede que no esté en la primera página de la Librería de Extensiones)</li> </ul>

De este modo, la extensión se añadirá tanto para el mBot2 como para el objeto. En mBlock aparece en azul oscuro. Haz clic en la nueva extensión y verás los diferentes bloques de código asociados a ella. Básicamente, la extensión añade un protocolo de comunicación. Te permitirá enviar y recibir mensajes entre el objeto y el robot mientras el robot está en modo Carga.

### 3. Establecer la conexión con el mBot2

Ya has instalado todas las extensiones necesarias para que el *hardware* y el *software* puedan comunicarse entre sí. Ahora vas a programar el mBot2 para controlarlo con el Aprendizaje Automático. La idea es que el mBot2 realice acciones basadas en los resultados de reconocimiento. Has programado una tarea similar anteriormente solo para el objeto (mBot2, cuaderno y fondo).

Ahora, los resultados del reconocimiento deben enviarse y recibirse a través de la conexión entre el Bot2 y el ordenador (ya sea por cable o de forma inalámbrica a través de Bluetooth Dongle). El código de recepción utiliza hechos, por lo que se ejecuta el código correspondiente



una vez recibido el mensaje. Con los hechos no hay necesidad de comprobar si hay un sensor o un mensaje en un circuito, lo que hace que el código sea más eficiente y más fácil de entender. Tienes que crear dos programas informáticos: uno para el dispositivo (mBot2) y otro para el objeto que utiliza el accesorio de Aprendizaje Automático:

Programa objeto

Programa mBot2

```

when this sprite clicked
  forever
    if recognition result is mBot2 ? then
      send upload mode message mBot2
    if recognition result is notebook ? then
      send upload mode message notebook
    if recognition result is nothing ? then
      send upload mode message nothing
  
```

```

when receiving upload mode message mBot2
  play yeah until done
  moves forward at 50 RPM for 2 secs

when receiving upload mode message notebook
  play surprised until done
  moves backward at 50 RPM for 2 secs
  
```

Después de reprogramar los bloques de código anteriores, asegúrate de que tu ordenador está conectado al mBot2 si utilizas un cable (no desconectes el cable). Idealmente, deberían estar conectados a través de Bluetooth utilizando Makeblock Bluetooth Dongle, porque esto permite que el mBot2 se desplace con más flexibilidad.

<p>Si usas el <i>dongle</i>:</p>	<p>Instala el <i>dongle</i> en el ordenador y enciende el mBot2. Tienes que emparejar el mBot2 con el Bluetooth Dongle. Para ello, haz clic en el botón del <i>dongle</i>. El LED comenzará a parpadear y el <i>dongle</i> intentará conectarse al controlador Makeblock más cercano (como mBot2 y otros). En una situación que tenga lugar en el aula, asegúrate de que esto se hace equipo por equipo, para garantizar la coincidencia correspondiente entre Dongle y mBot2. Una vez que estén emparejados, escucharás un pitido del mBot2. Esto significa que el ordenador y el mBot2 han establecido una conexión inalámbrica.</p>
----------------------------------	--



El Makeblock Bluetooth Dongle recordará el último dispositivo emparejado y se volverá a conectar instantáneamente.

Si no tienes el Bluetooth Dongle de Makeblock, puedes comprobar la compatibilidad del Bluetooth en tu ordenador a través del siguiente enlace: <https://www.yuque.com/makeblock-help-center-en/cyberpi/bluetooth-compatibility>

Ahora cambia mBlock a modo Carga y conecta el mBot2 a mBlock. Haz clic en el botón "Conectar" y carga el programa informático. ¡Ahora puedes empezar a hacer pruebas!

### 3. Prueba (25 min)

#### Paso 3: Prueba

Tienes todo el conocimiento y las habilidades para usar el mBot2 en cualquier situación. Ahora vas a trabajar con el ecosistema de un ratoncito. Un ratón vive en la naturaleza donde debe buscar comida y agua. También existen muchos peligros, como los depredadores que se alimentan de ratones.

En esta tarea, programarás el mBot2 para que actúe como un pequeño ratón que busca comida y huye de los depredadores.

Encuentra una imagen de un águila volando y un poco de maíz. El maíz es el alimento del ratoncito. El águila es la cazadora. Puedes imprimir estas imágenes o usar tu *smartphone* para mostrarlas en la cámara web. También puedes dibujar alimentos y depredadores si quieres. Simplemente utiliza el mismo método que has decidido para "enseñárselos" a tu algoritmo usando la extensión Aprendizaje Automático y para probar el programa.

Programa el mBot2 para que si muestras el maíz a la cámara, el mBot2 se mueva hacia adelante (si utilizas el cable, haz un pequeño movimiento), pero si muestras la imagen del águila, se mueva hacia atrás.

También puedes añadir más código: si el ratoncito ve comida, dice "¡qué rico!" y cuando ve el águila, podemos escuchar los latidos de su corazón.

Al crear el programa informático para esta tarea, es útil seguir el siguiente esquema paso a paso.

	Explicación
Paso 1: ¿Qué quieres hacer?	<ul style="list-style-type: none"> <li>• ¿Qué quieres programar?</li> </ul>
Paso 2: ¿Qué necesitas?	<ul style="list-style-type: none"> <li>• ¿Qué necesitas además del mBot2?</li> </ul>
Paso 3: ¿Qué bloques de código necesitas?	<ul style="list-style-type: none"> <li>• ¿Qué vas a hacer para que el mBot2 se desplace?</li> <li>• ¿Qué bloques de código usarás?</li> </ul>



	<ul style="list-style-type: none"><li>• Describe brevemente cómo funciona tu programa (pseudocódigo/lenguaje natural, diagrama de flujo o Lenguaje Unificado de Modelado (UML))</li><li>• Si necesitas más explicaciones, puedes hablar con sus compañeros de clase, el profesor o investigar sobre el tema. También tienes ayuda disponible para cada bloque de códigos en mBlock.</li></ul>
Paso 4: Pruebas e implementación	<ul style="list-style-type: none"><li>• ¿Está lista la primera versión? ¡Pruébala! Durante la ronda de pruebas, anota los puntos de mejora</li><li>• Trabaja en los puntos de mejora hasta que el mBot2 haga exactamente lo que tenías en mente</li><li>• ¿Lo has conseguido? Graba el resultado final y pregúntale a tu profesor si puedes publicarlo en las redes sociales con el hashtag #mBot2inclass</li></ul>

## 4. Conclusión (5 min)

### Paso 4: Conclusión

¿Has sido capaz de recrear el ecosistema del ratón?

En esta lección has aprendido sobre el Aprendizaje Automático y cómo aplicarlo. También sabes cómo usar una comunicación (y por qué preferiblemente inalámbrica) entre un objeto y el mBot2, y utilizarla para intercambiar datos entre ambos. Además, has hecho que varias características técnicas del mBot2 funcionen juntas.

Ha llegado el momento de realizar una breve reflexión. Piensa por ti mismo y discute con el grupo:

- ¿Qué crees que ha salido bien?
- ¿Qué podría mejorarse?
- ¿Qué partes de la lección te han resultado fáciles y cuáles más difíciles?
- ¿Sobre qué te gustaría recibir más información?
- ¿Quién podría ayudarte con eso?