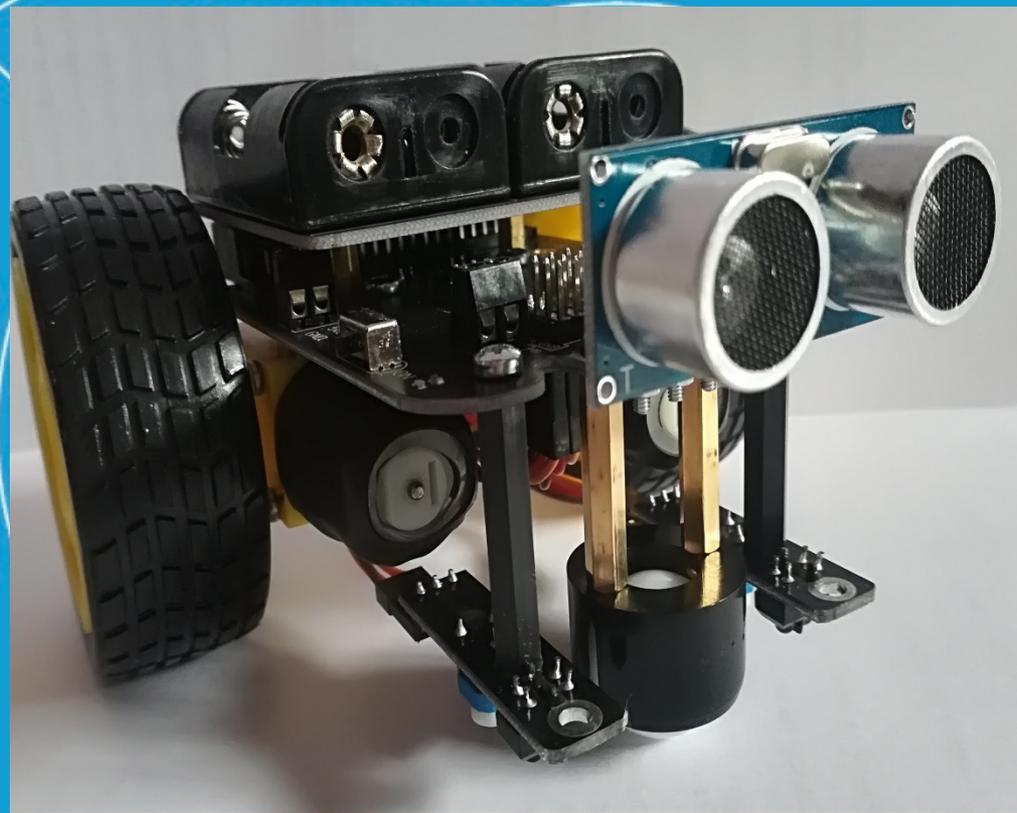


Mini-Taller de Robótica Móvil

Programa un robot de sumo

Ven a la Universidad de Alcalá
y construye TU robot



Actividad orientada a estudiantes de Enseñanza Secundaria y Bachillerato



Universidad
de Alcalá



Departamento de
electrónica



GOBIERNO
DE ESPAÑA

MINISTERIO
DE ECONOMÍA, INDUSTRIA
Y COMPETITIVIDAD



FUNDACIÓN ESPAÑOLA
PARA LA CIENCIA
Y LA TECNOLOGÍA

Mini-Taller de Robótica Móvil

Programa un robot de sumo



Universidad
de Alcalá



Departamento de
electrónica



GOBIERNO
DE ESPAÑA

MINISTERIO
DE ECONOMÍA, INDUSTRIA
Y COMPETITIVIDAD

FECYT



FUNDACIÓN ESPAÑOLA
PARA LA CIENCIA
Y LA TECNOLOGÍA

Planificación de Actividades

Introducción del Taller

- ¿Qué es un robot? ¿Qué partes tiene?
- El robot TuBot
 - Motores
 - Sensores
- Entorno de programación
 - Descripción general



Taller de robótica

- Moviendo el robot
- Detectando obstáculos
- Detectando el tatami
- Compitiendo en Sumo

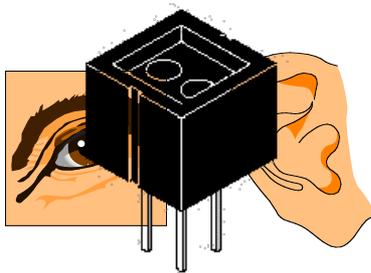
Competición de Sumo

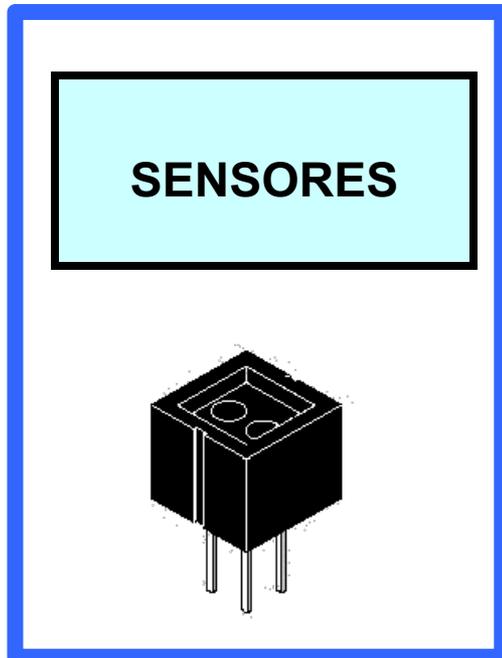


Hispabot'04 Final Panzer - Bestia



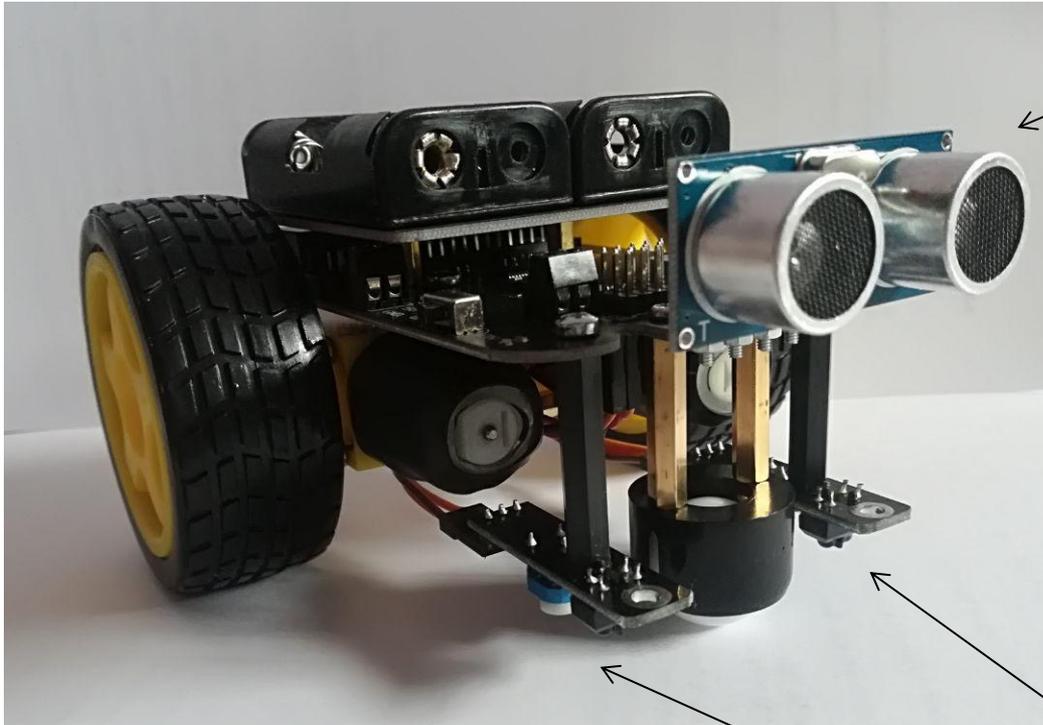
SENSORES





Sensores de TuBot

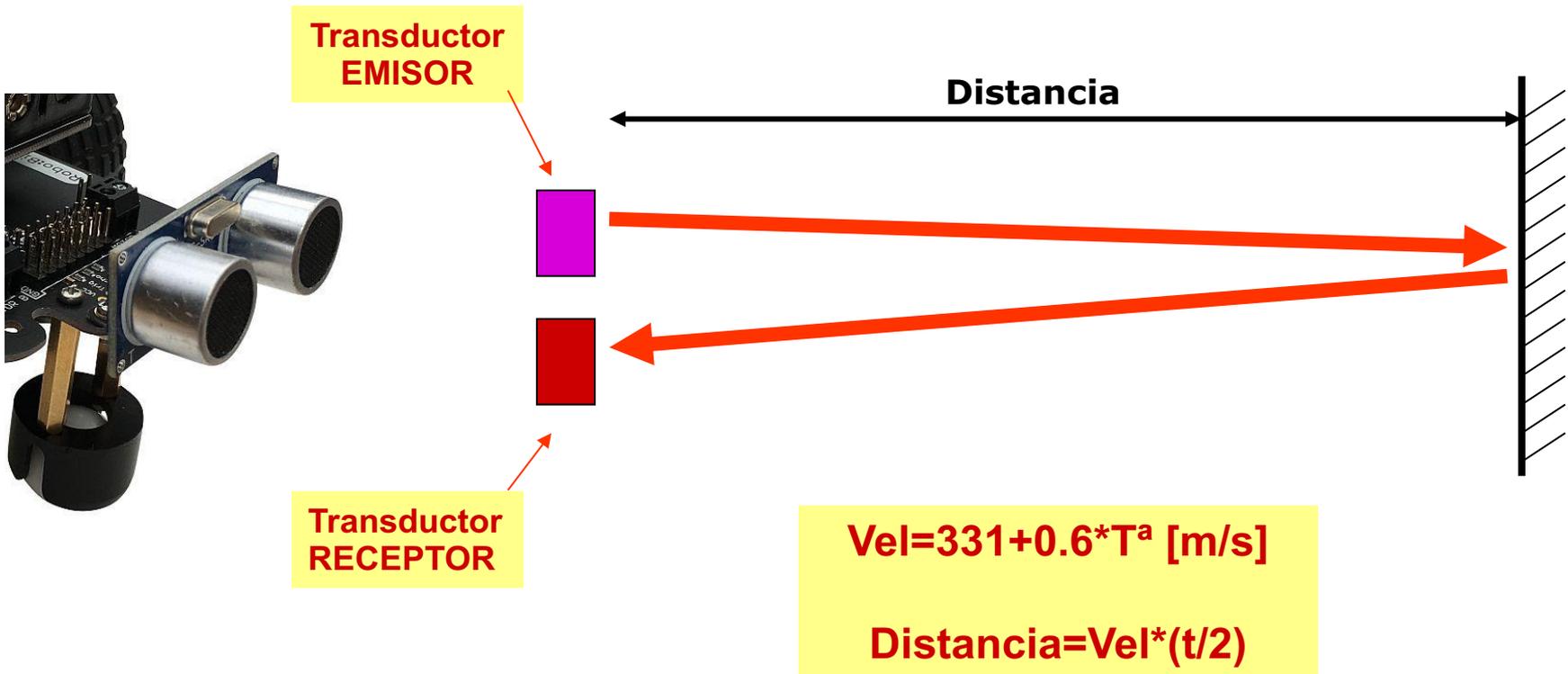
Sensor de Distancia
por Ultrasonidos



Sensor de Suelo
Sensores infrarrojos reflectivos

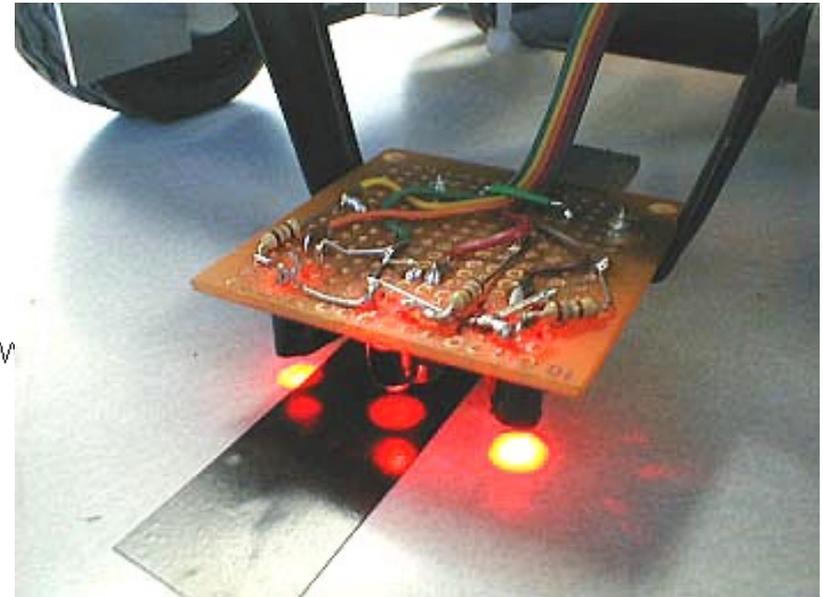
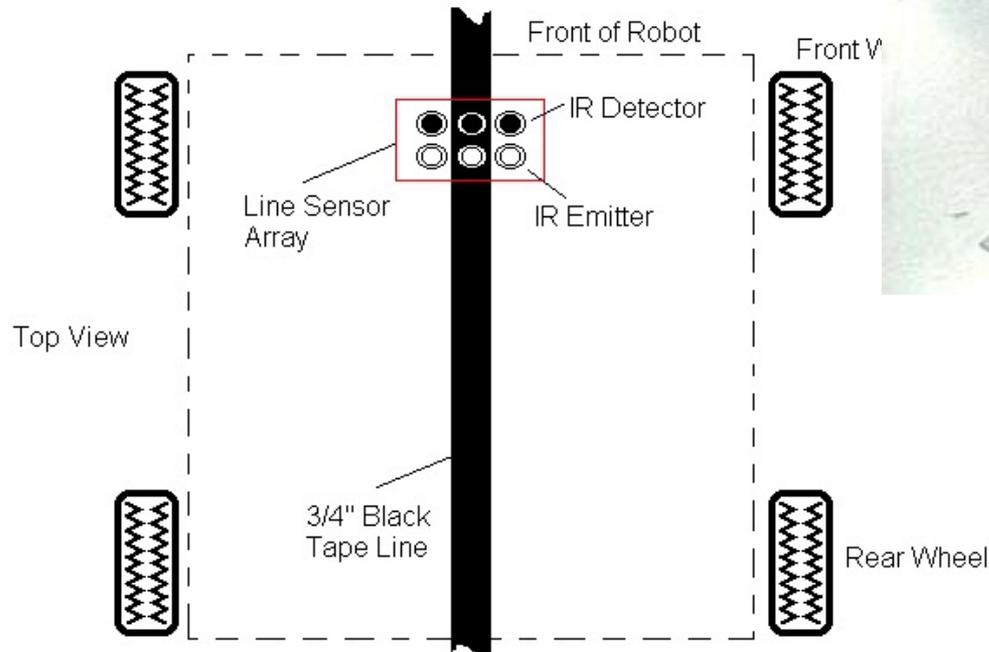
Medida de Distancia: Ultrasonidos

- El sonido tiene una velocidad de propagación de 330m/s aprox.
- Se mide el tiempo que tarda el sonido en ir y volver



Detección de Línea o Suelo

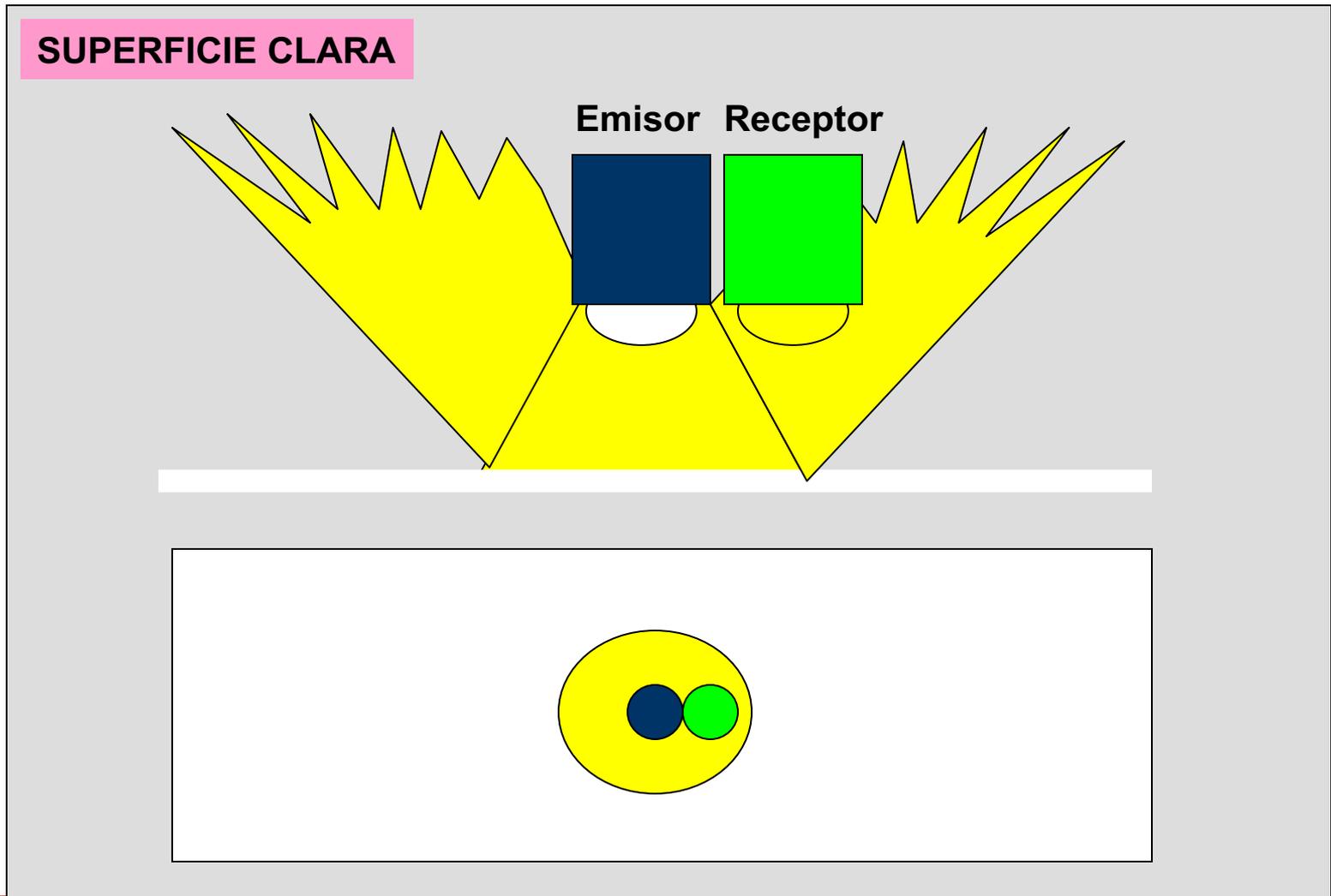
- Sensor de infrarrojos reflectivo: detección de blanco o negro



Detección de Línea o Suelo

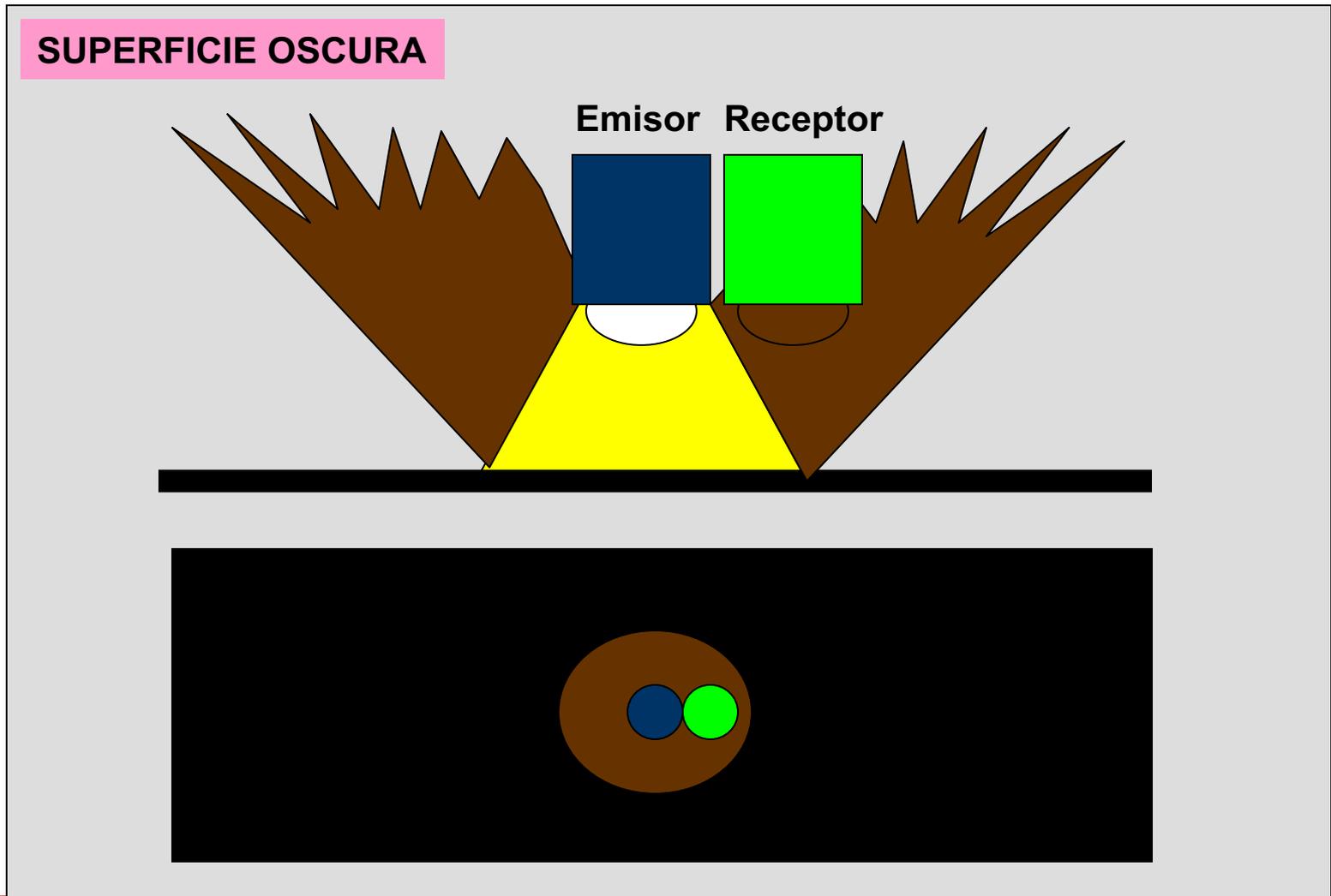


- Sensor de infrarrojos reflectivo: detección de blanco o negro



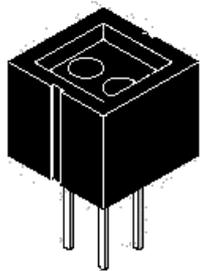
Detección de Línea o Suelo

- Sensor de infrarrojos reflectivo: detección de blanco o negro





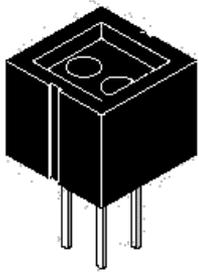
SENSORES



ACTUADORES



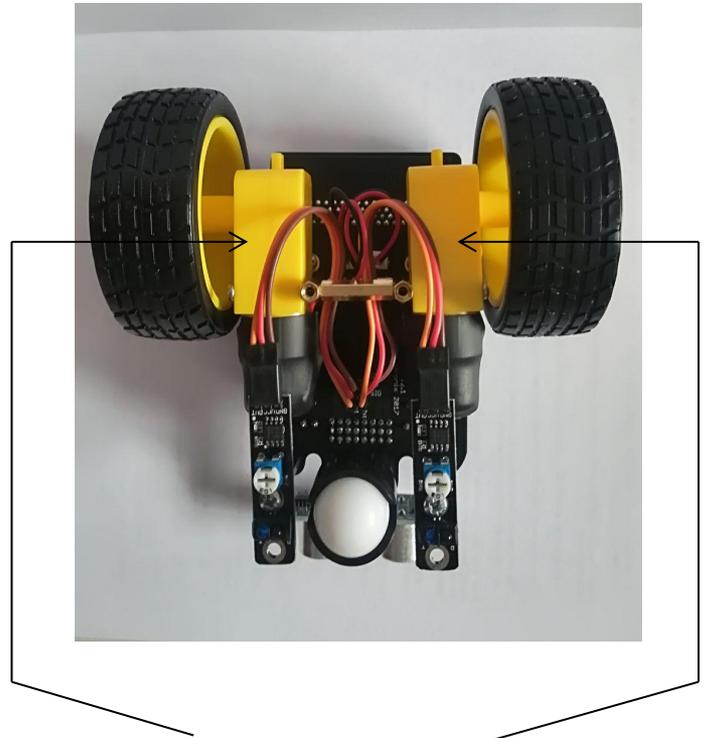
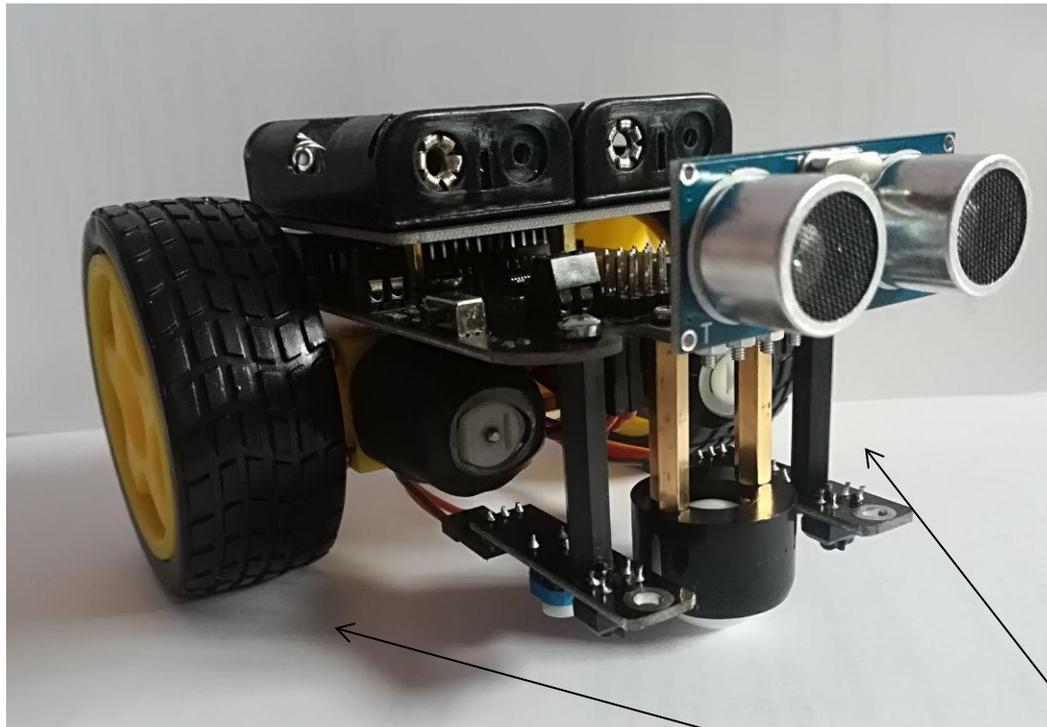
SENSORES



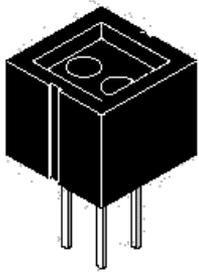
ACTUADORES



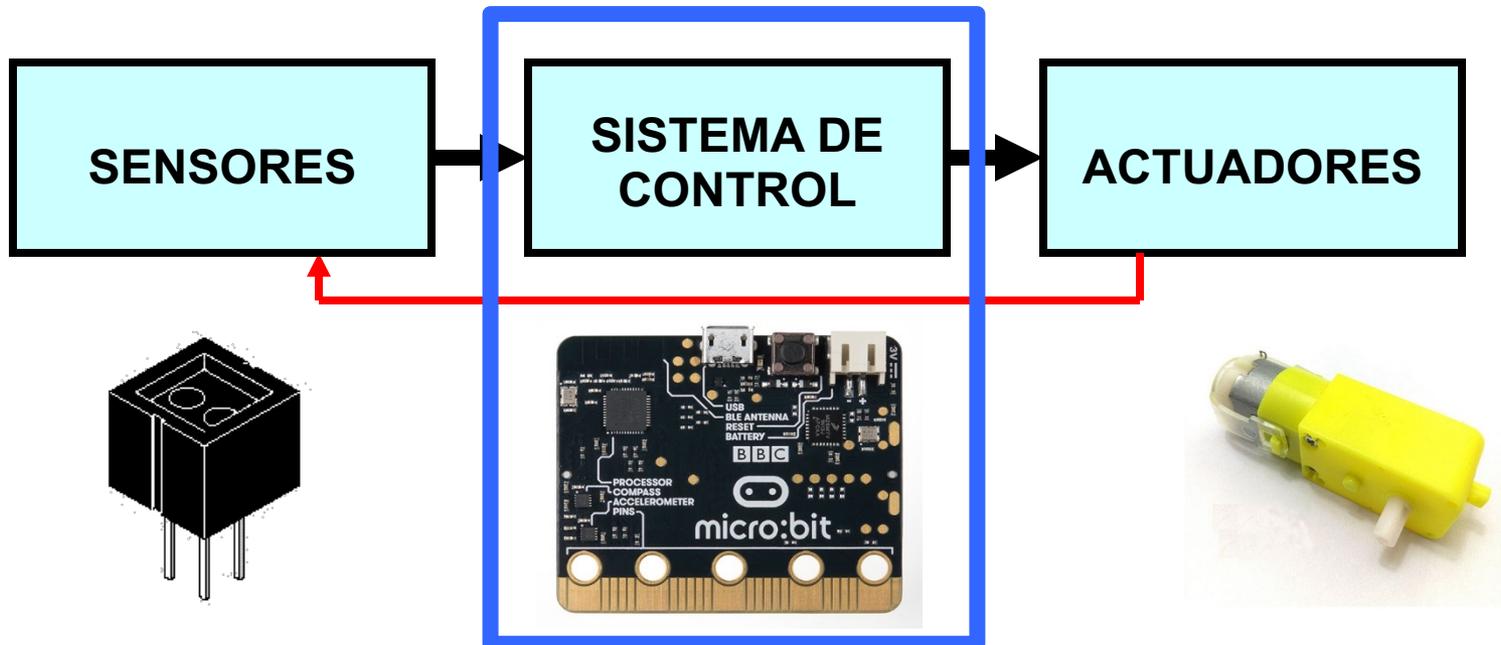
Actuadores de TuBot



Dos motores

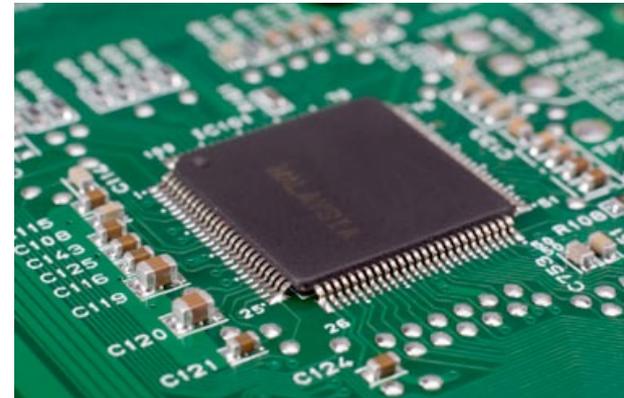


Elementos de un Robot Móvil Autónomo



Programa y compilador

- Un **Programa** se puede escribir en muchos **lenguajes** diferentes.
- El microprocesador realmente entiende sólo órdenes en **código máquina** (**unos y ceros**)

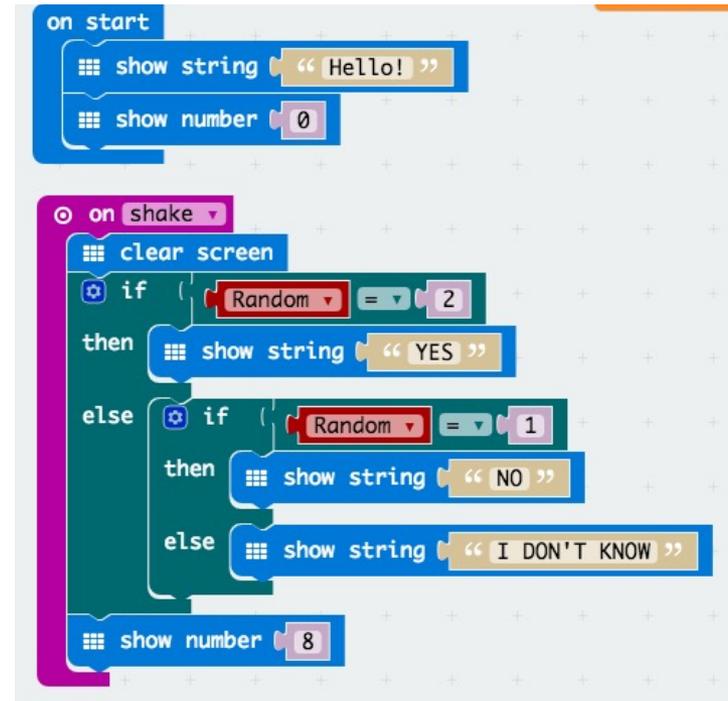


Source: <http://pacotraver.files.wordpress.com/2011/11/interprete.jpg>

http://2.bp.blogspot.com/_Pm8qvnCsVOI/TCww_SAuznI/AAAAAAAAAA4/9asQgJGiQMw/s1600/MICRO.jpg

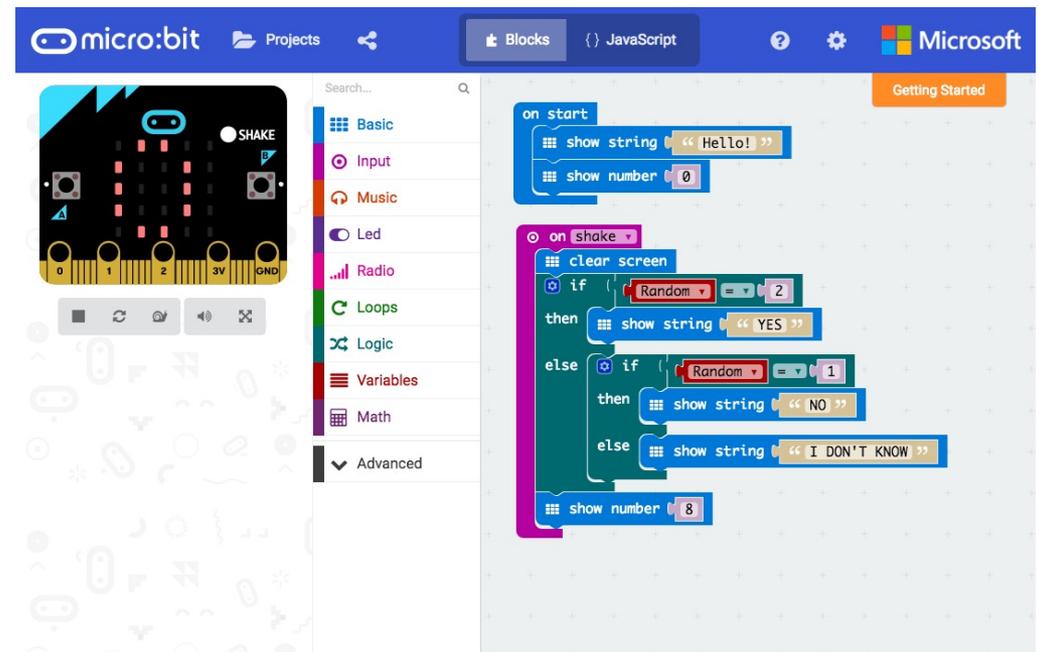
Programa y compilador

- Un **Programa** se puede escribir en muchos **lenguajes** diferentes.
- El microprocesador realmente entiende sólo órdenes en **código máquina** (**unos y ceros**)
- Para hacerlo más fácil vamos a utilizar un **lenguaje gráfico** que el ordenador traduce a **código máquina** (**unos y ceros**)



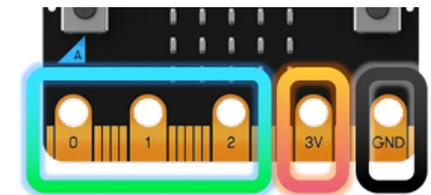
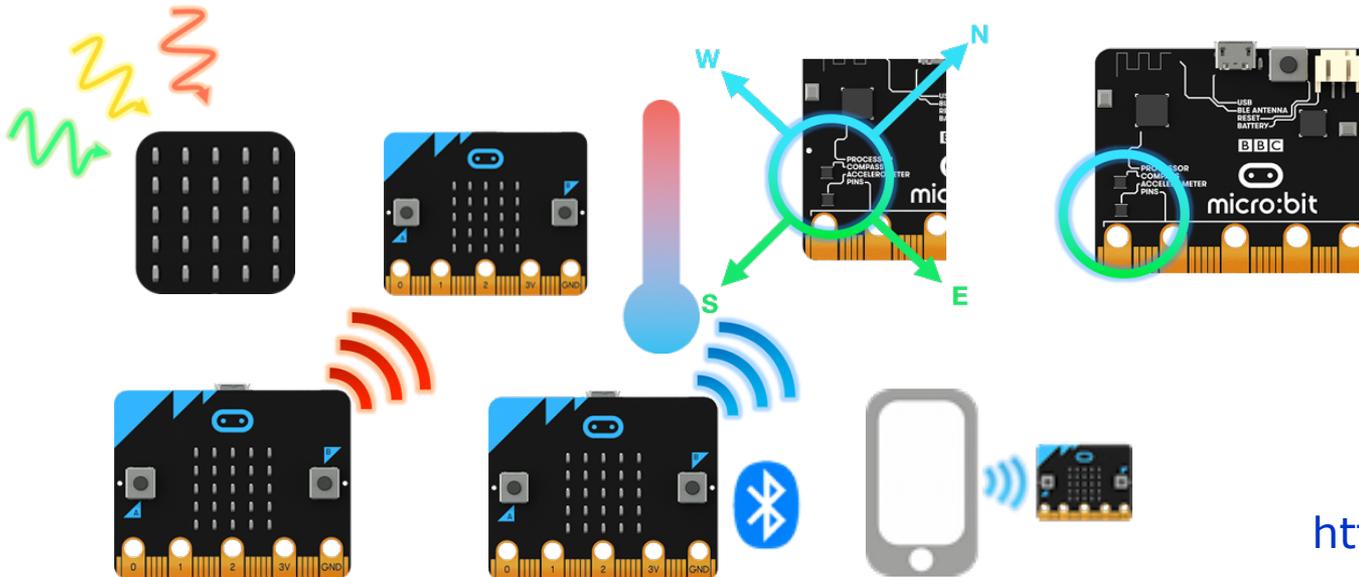
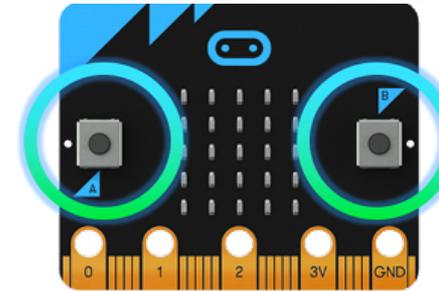
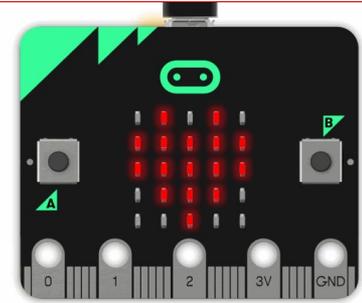
MICROBIT: buen sistema de iniciación

- ❑ Plataforma diseñada como elemento de **iniciación a la programación y a la electrónica.**
- ❑ Muy **sencillo** de utilizar.
- ❑ Dispone de una enorme **comunidad de usuarios.**



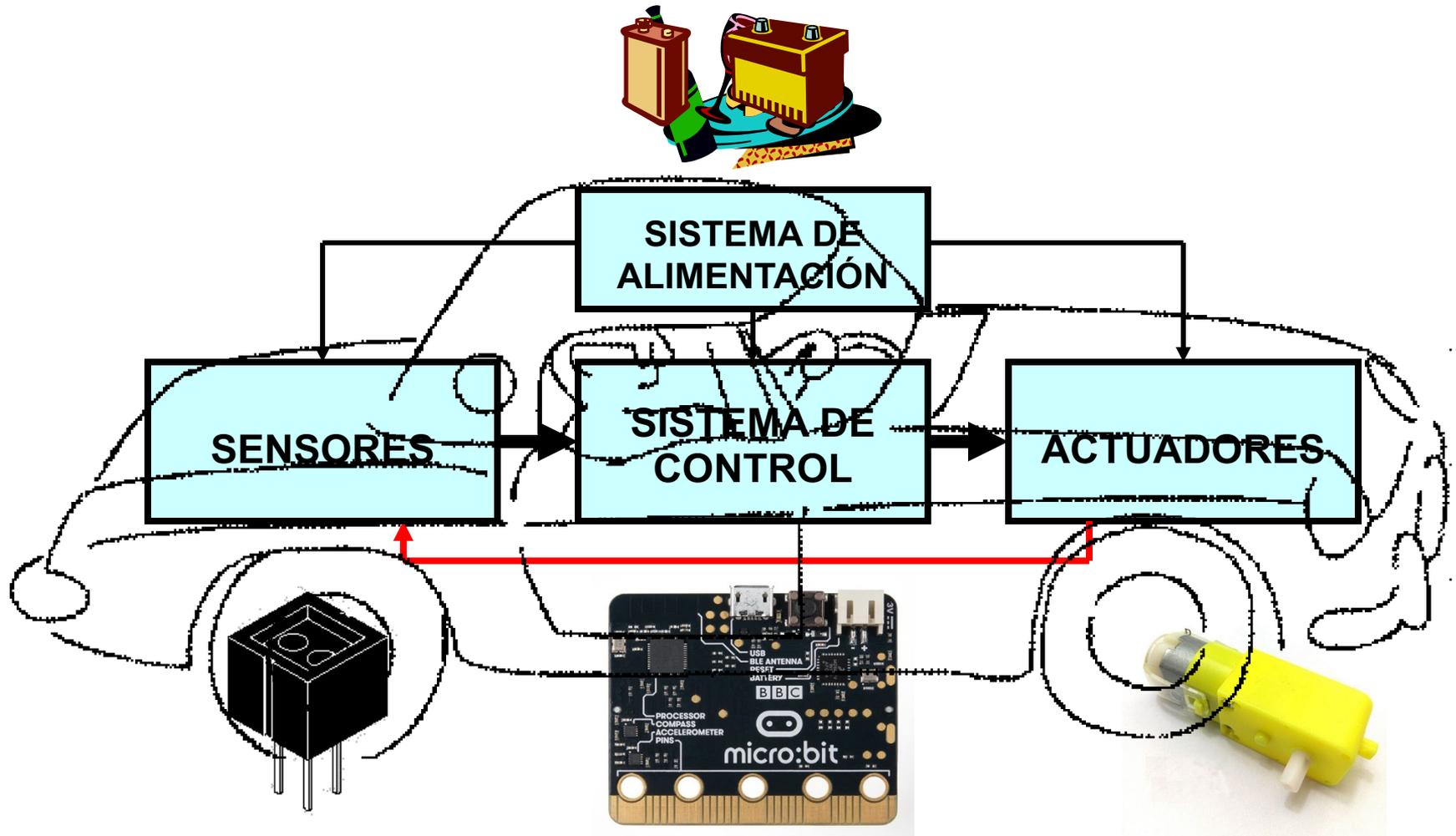
MICROBIT: buen sistema de iniciación

- ❑ Matriz de 5x5 LEDs capaz de visualizar mensajes
- ❑ Dos pulsadores
- ❑ Sensor de luz ambiente
- ❑ Sensor de temperatura
- ❑ Brújula
- ❑ Acelerómetros en tres ejes
- ❑ Capacidad de comunicación Bluetooth y radio

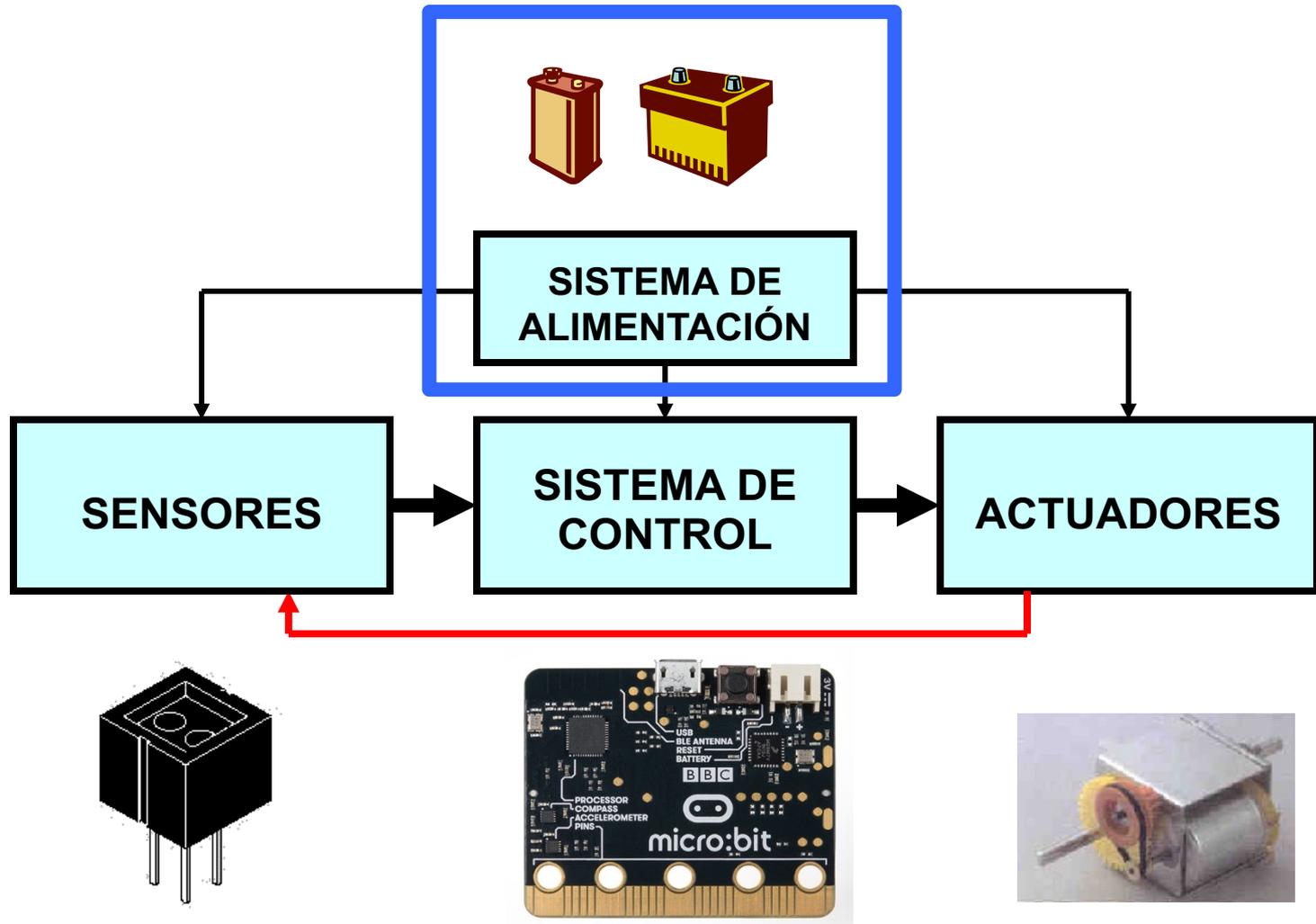


<http://microbit.org/es/>

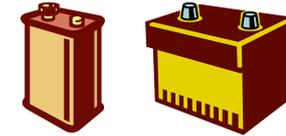
Elementos de un Robot Móvil Autónomo



Elementos de un Robot Móvil Autónomo



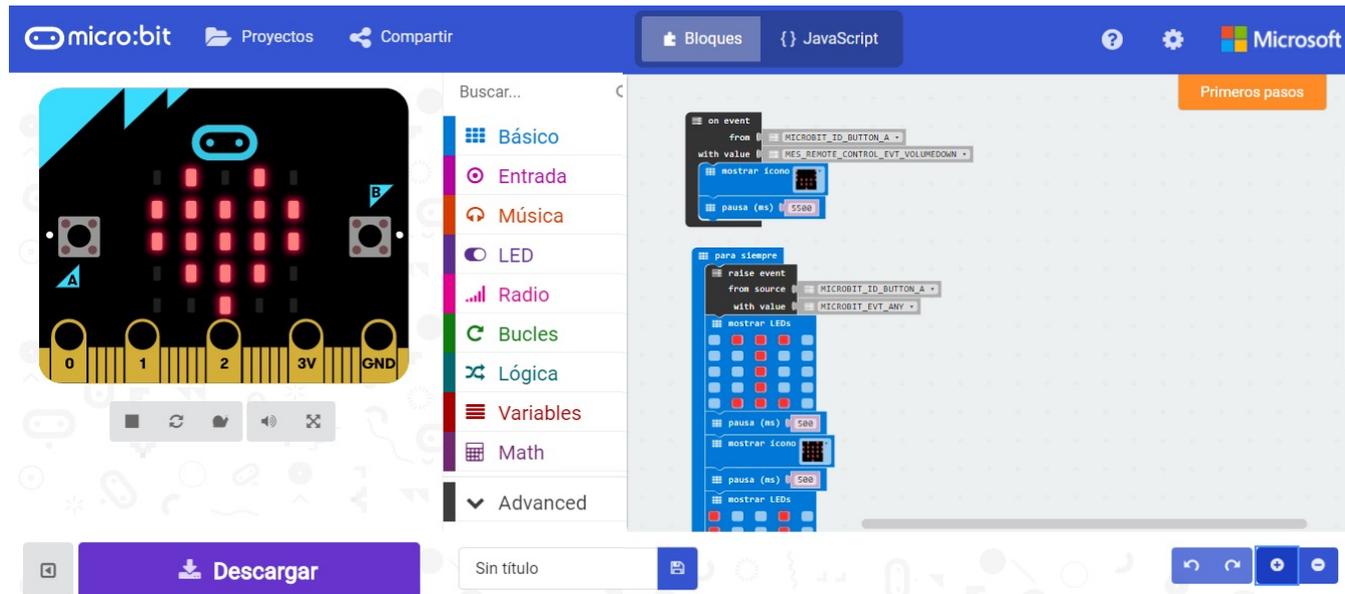
Elementos de un Robot Móvil Autónomo

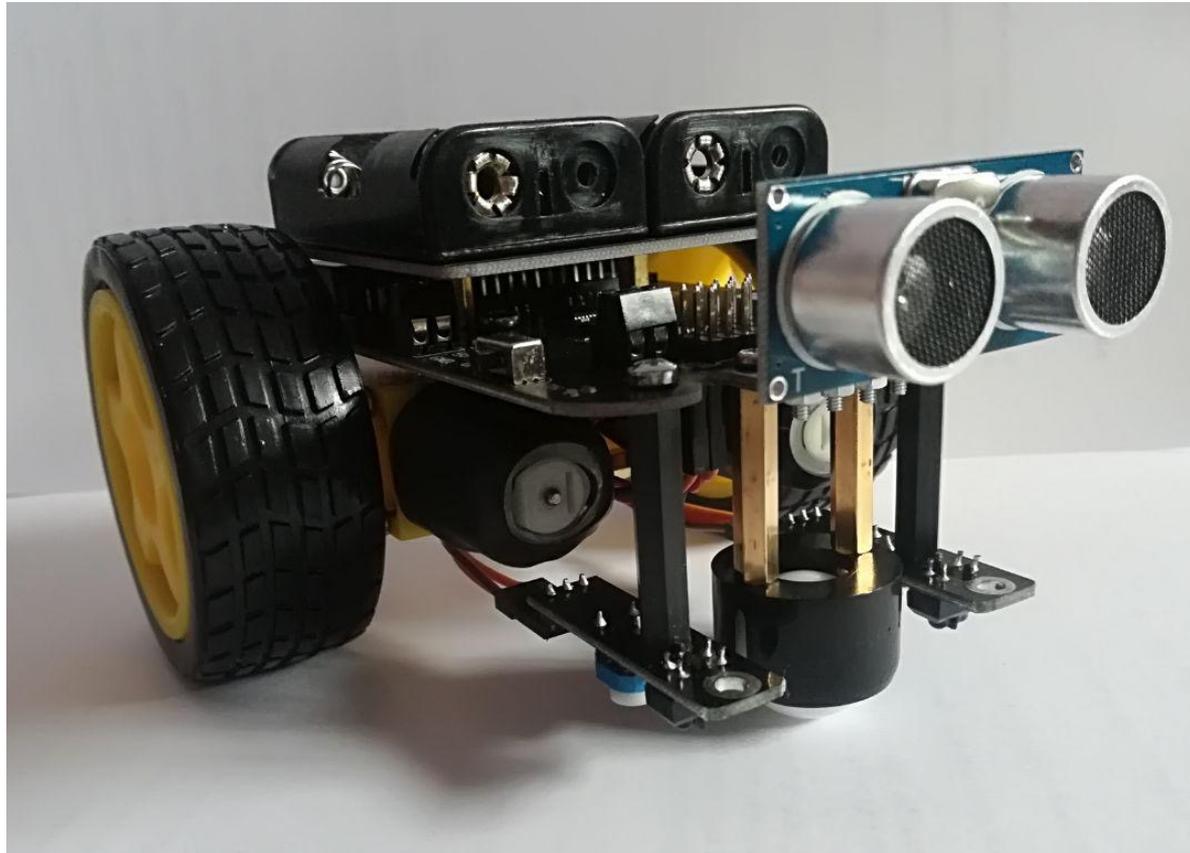


**SISTEMA DE
ALIMENTACIÓN**

Programación con MicroBit

- ❑ Vamos a utilizar un **lenguaje de programación gráfico: Editor de Bloques de JavaScript** desarrollado por Microsoft MakeCode para MicroBit
 - Software de programación OnLine
 - Su uso es gratuito
 - *Versión web* → <https://makecode.microbit.org/#lang=es>
 - Si te quedas sin internet: ¡No pasa nada! Sigue funcionando 😊





Paso 0.a: Metodología

1. En el Taller te iremos enseñando **paso a paso** cómo programar el robot para que haga lo que tú quieras.
2. El robot ya está montado. Si ves que hay algún cable o tornillo suelto (o que se está soltando) díselo a los profesores.
3. Sigue las instrucciones paso a paso. **No te saltes ninguno.**
4. **Lee detenidamente** las instrucciones de cada uno de los pasos.
 - Si algo no entiendes **pregunta** a los profesores.
5. En cada paso **se parte de un ejemplo** ya hecho. Hay que cargarlo, probarlo y entender lo que hace.
 - Si no entiendes lo que hace. **Pregunta** a los profesores

Paso 0.b: Metodología

6. En cada paso se proponen **ejercicios** para que hagas modificando el programa de base.
 - Hay ejercicio **marcados en verde** (fáciles de hacer) y otros **marcados en rojo** (para los más avanzados, no es necesario hacerlos) hay otros de dificultad intermedia de **color naranja.**
7. Las **primeras instrucciones son más guiadas**, las últimas requerirán que pongas más de tu parte.
8. **Si terminas antes** que los demás, intenta mejorar el robot para que sea más competitivo.

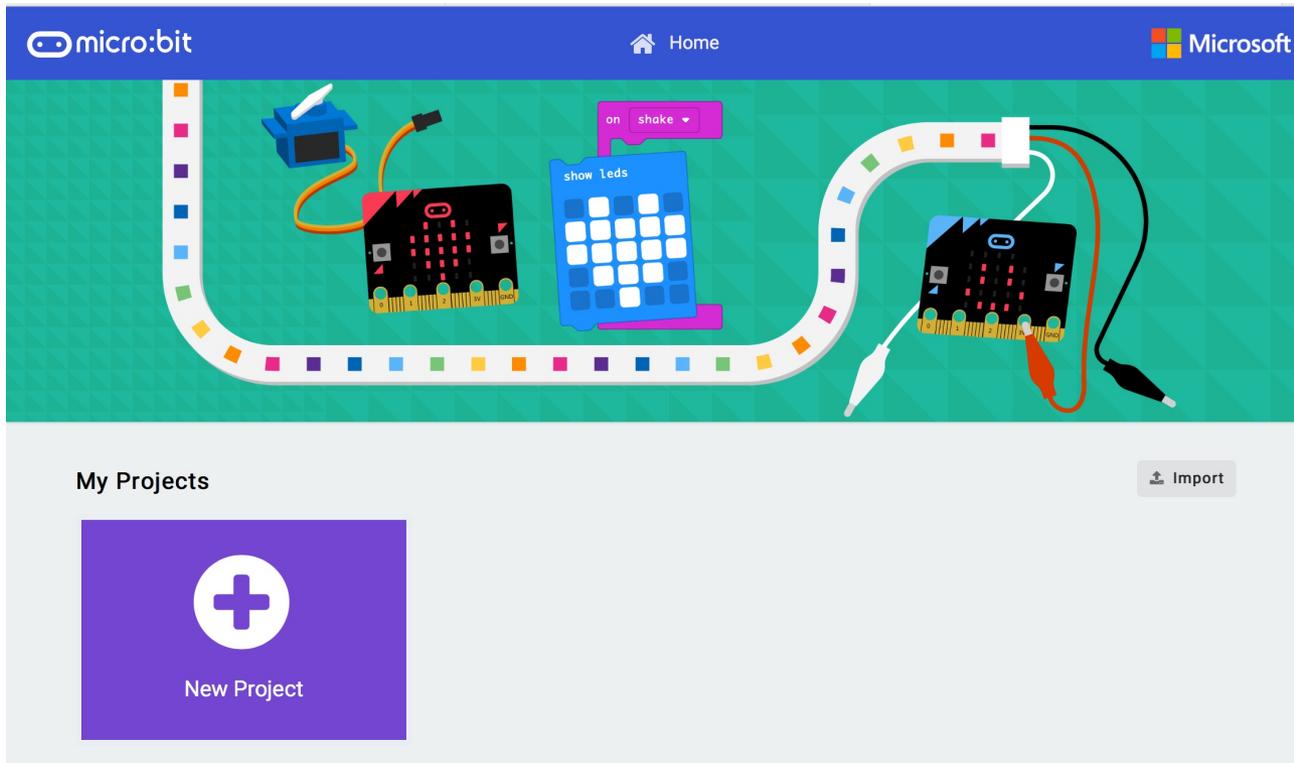
Paso 1a: Abriendo el entorno de programación



- ❑ Abre la aplicación **MicroBit**:

<https://makecode.microbit.org>

- ❑ La aplicación tiene la apariencia de la figura



- ❑ **Pulsa en "New Project"**

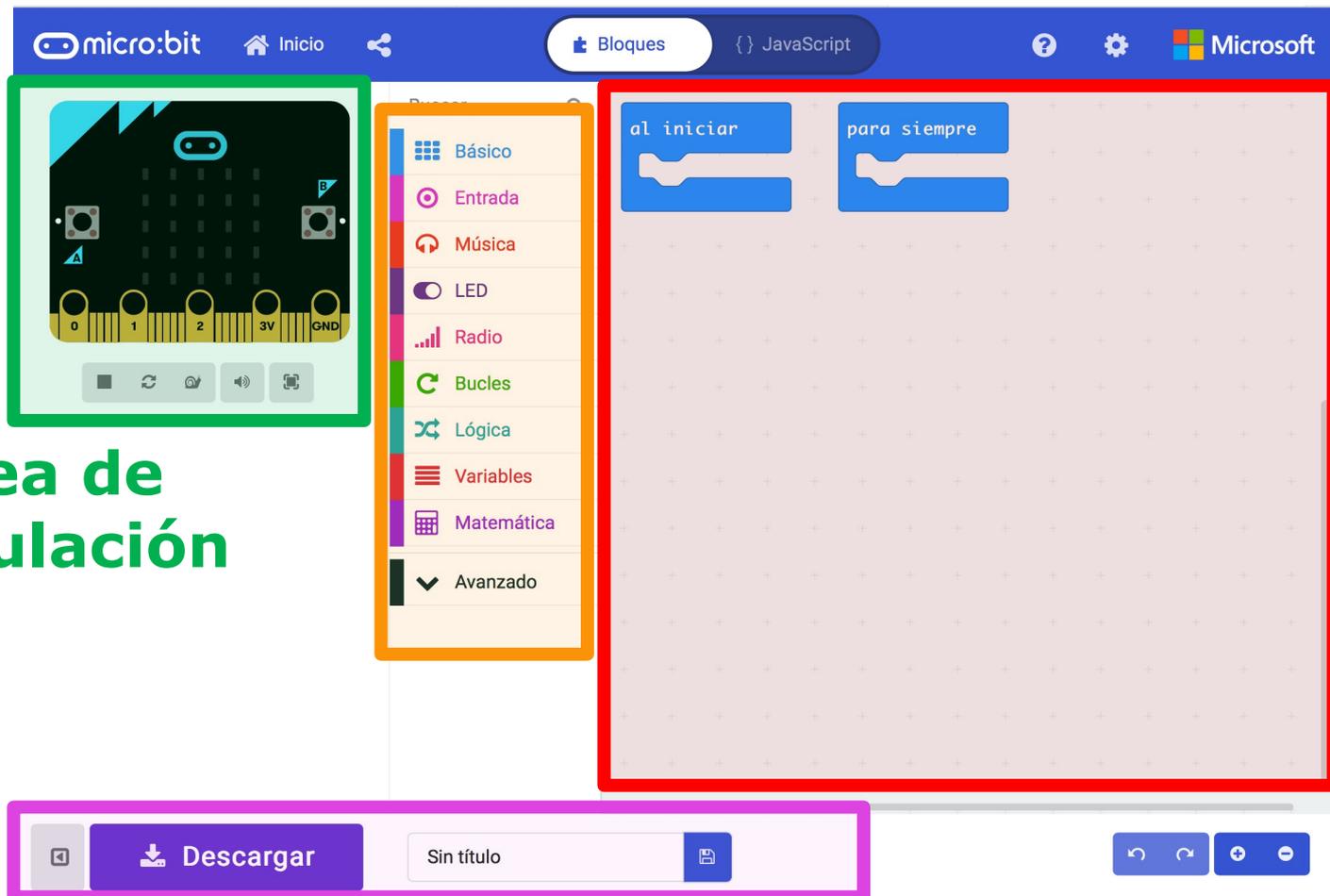
Paso 1a: Abriendo el entorno de programación



- La aplicación tiene la apariencia de la figura

Área de Instrucciones

Área de programación



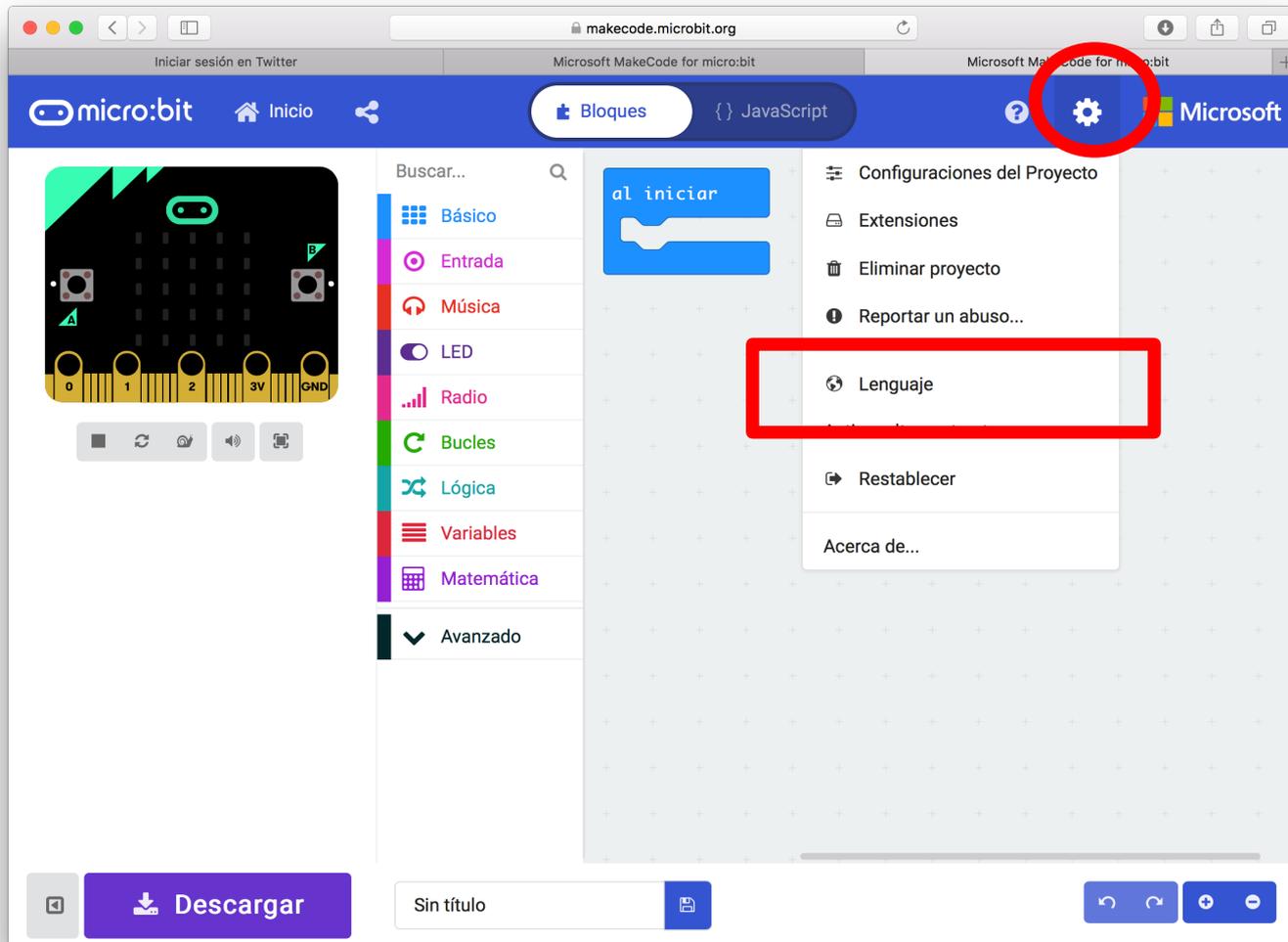
Área de Simulación

Descarga

Paso 1a: Abriendo el entorno de programación



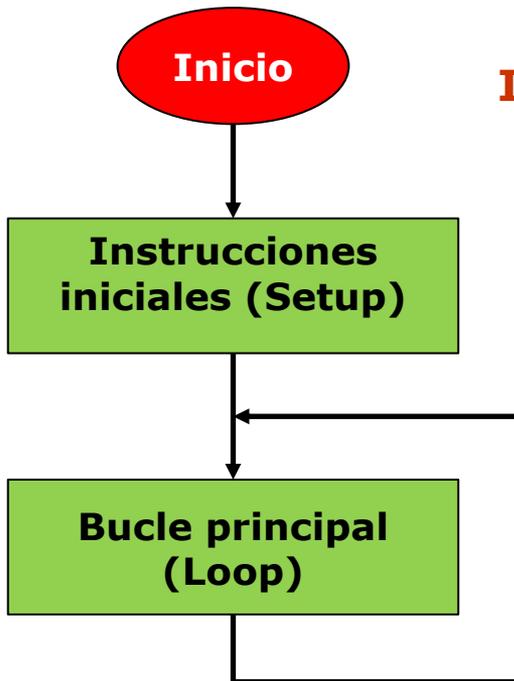
❑ Configuración del idioma





□ Área de Programación

■ PARTES DE UN PROGRAMA



Instrucciones Iniciales (Setup)

Al iniciar:

- Se ejecutan **UNA vez** al iniciar el programa

al iniciar

El programa se mete aquí dentro

Bucle Principal (Loop)

Para siempre:

- Dentro se escribe el programa que se va a ejecutar **TODO EL RATO**

para siempre

El programa se mete aquí dentro

Paso 1c: Abriendo el entorno de programación



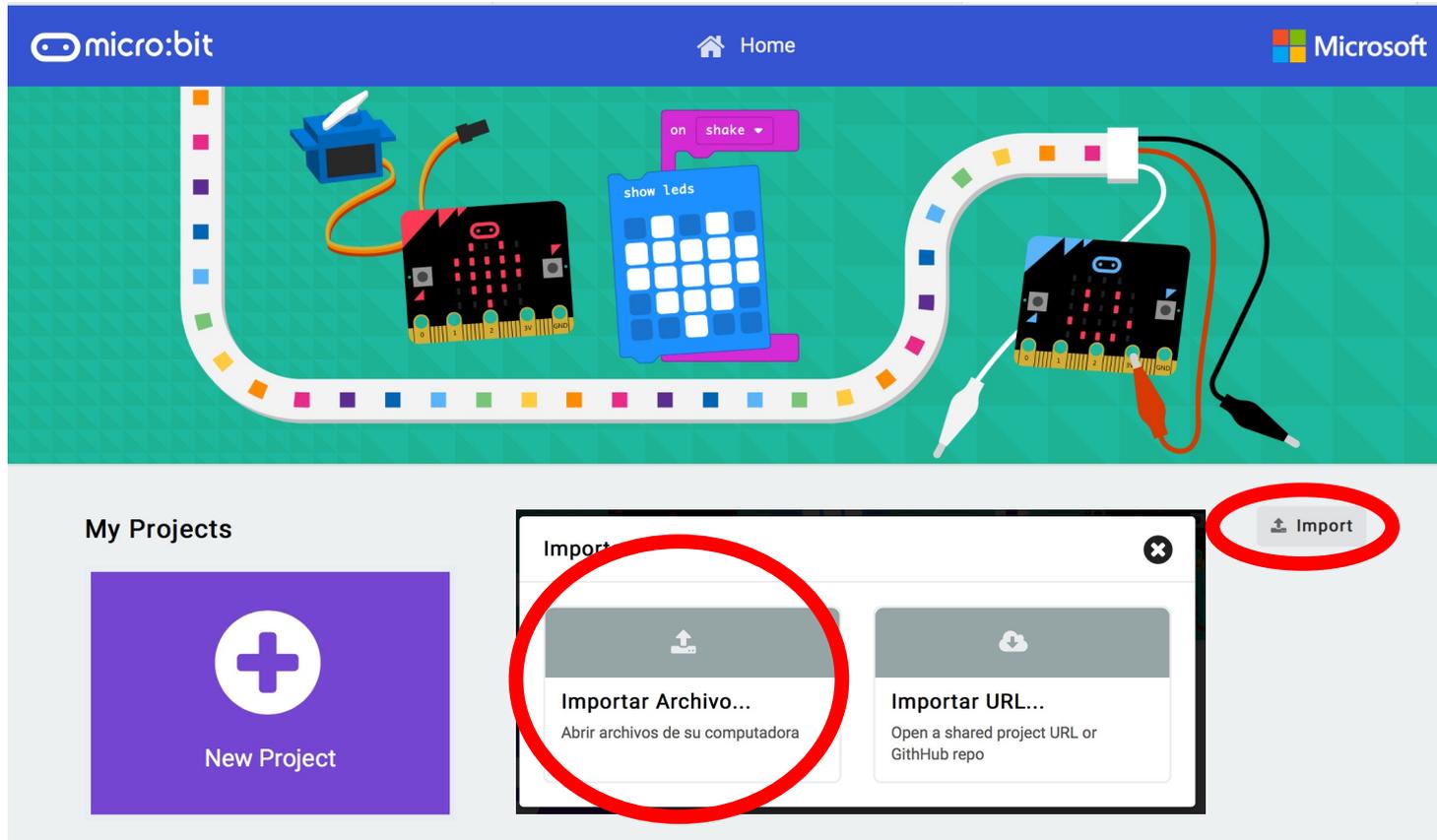
- En cada paso habrá un ejemplo a ejecutar

Para cargar los proyectos:

1. Se selecciona **PROYECTOS**, en la parte superior
2. Seleccionar **IMPORTAR ARCHIVO** desde el ordenador
3. Con cada ejercicio nuevo, abrir su **PASO** correspondiente:
 - Paso_3_EncendiendoUnLed.hex
 - Paso_4_SensorDeDistancia.hex
 - Paso_5_DetectandoObjeto.hex
 - Paso_6_MoviendoMotores.hex
 - Paso_7_DetectandoObstaculos.hex
 - Paso_8_SensoresSueloConFunciones.hex

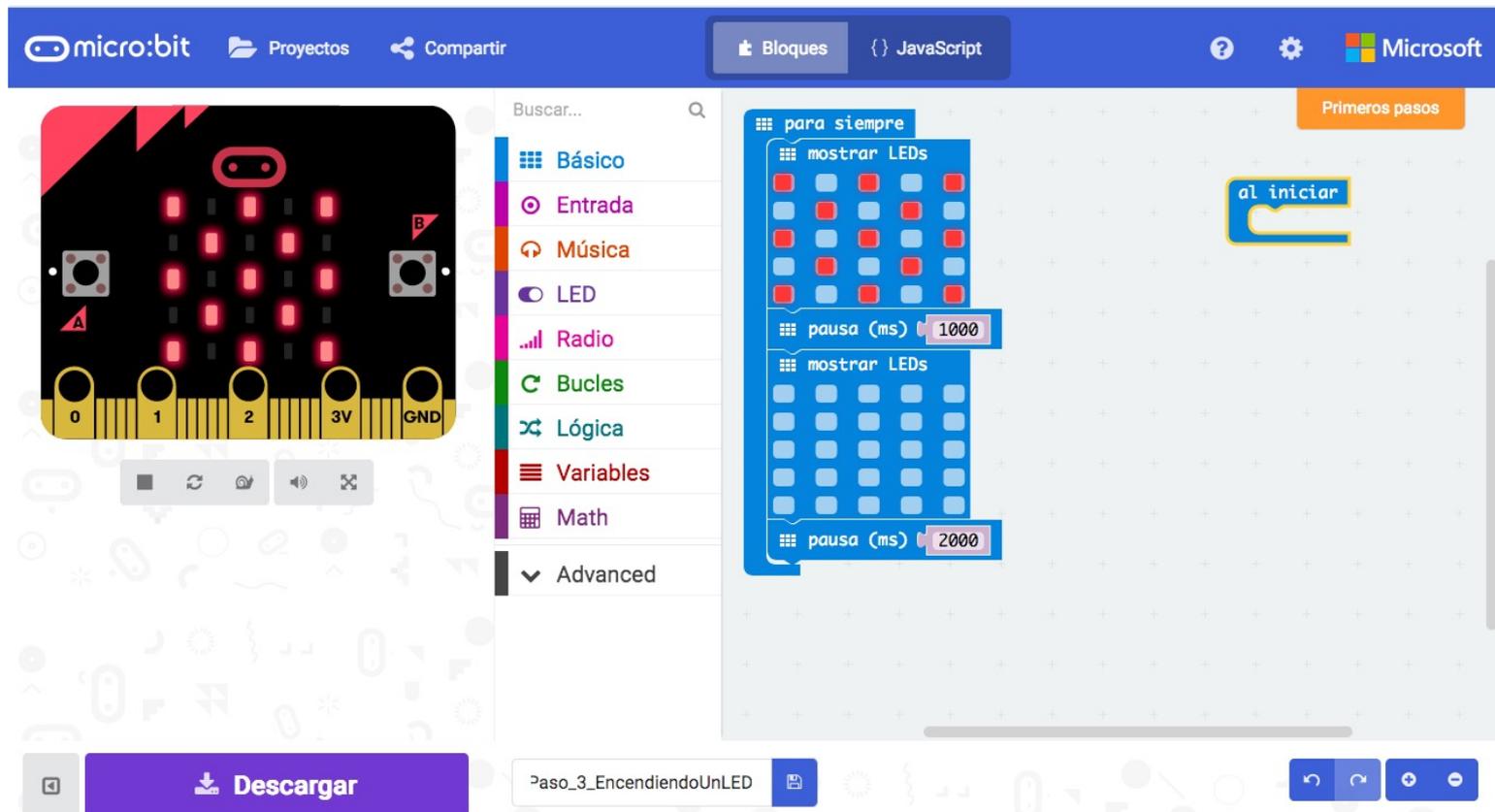
Paso 2.a: Abriendo el primer programa

- Carga el programa **Paso_3_EncendiendoUnLED.hex**
 - Volver a la página principal: <https://makecode.microbit.org/#>
 - También pulsando en "Home" o "Inicio" (Arriba a la izquierda)
 - Importar (a la derecha)



Paso 2.b: Abriendo el primer programa

- ❑ Carga el programa **Paso_3_EncendiendoUnLED.hex**
 - Proyectos (Arriba a la izquierda) → Importar Archivo
 - Este programa hace que los LEDs se enciendan y se apaguen



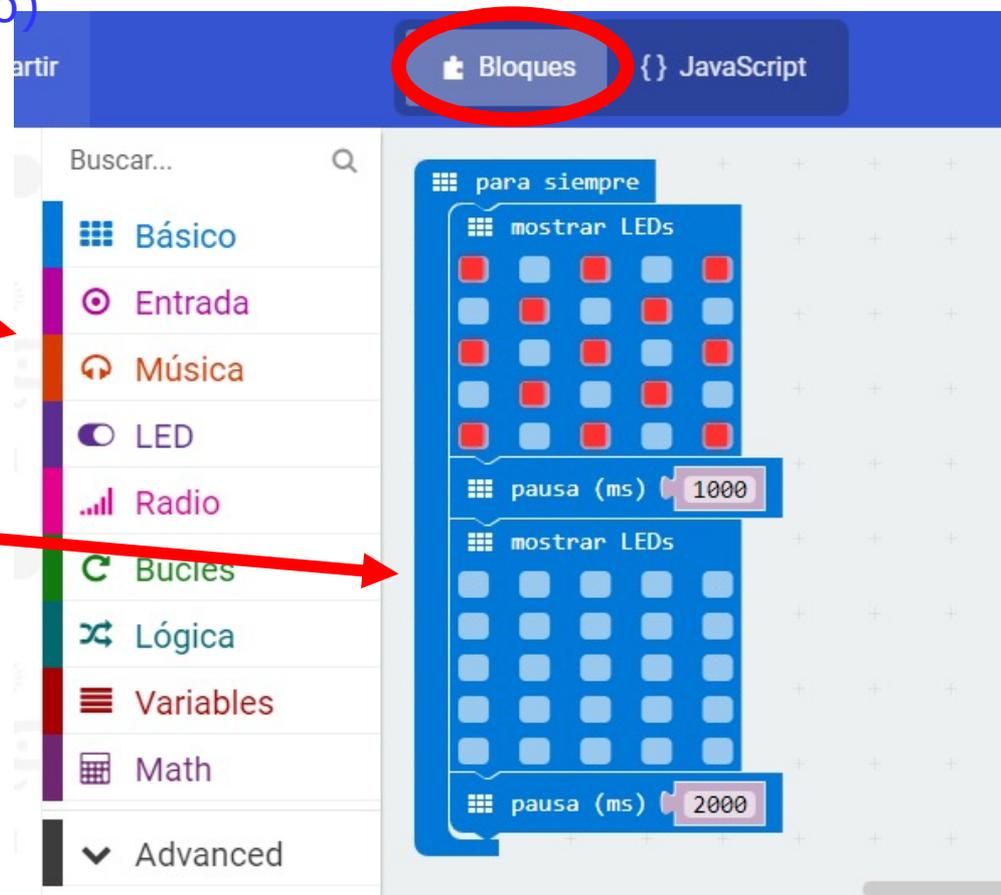
Paso 2.c: Abriendo el primer programa

□ Área de Programación

- Permite introducir un programa en la tarjeta de control
- Usaremos la pantalla de Bloques (JavaScript es más complejo)

- A la izquierda:
Los bloques que se van a utilizar
- A la derecha: El programa

- Los colores del menú de bloques es el mismo que el de los bloques que lo contienen



Paso 2.d: Abriendo el primer programa

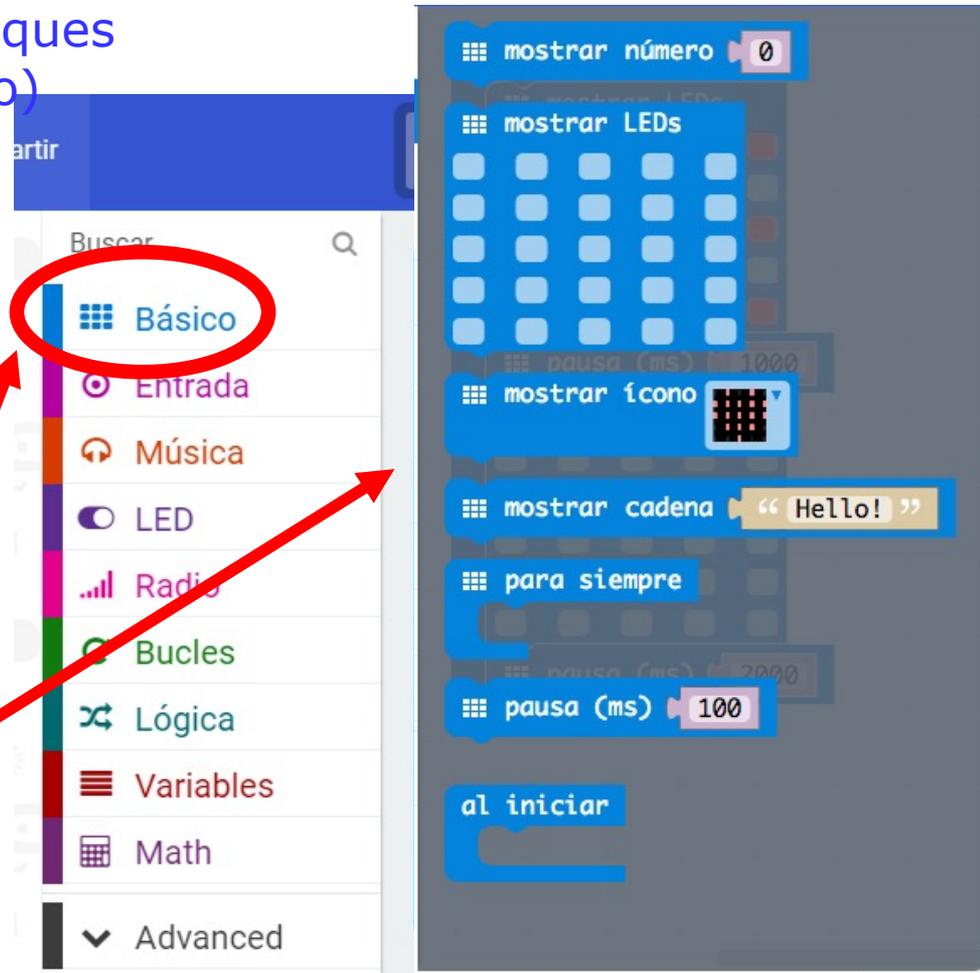
□ Área de Programación

- Permite introducir un programa en la tarjeta de control
- Usaremos la pantalla de Bloques (JavaScript es más complejo)

- A la izquierda:
Los bloques que se van a utilizar

- A la derecha: El programa

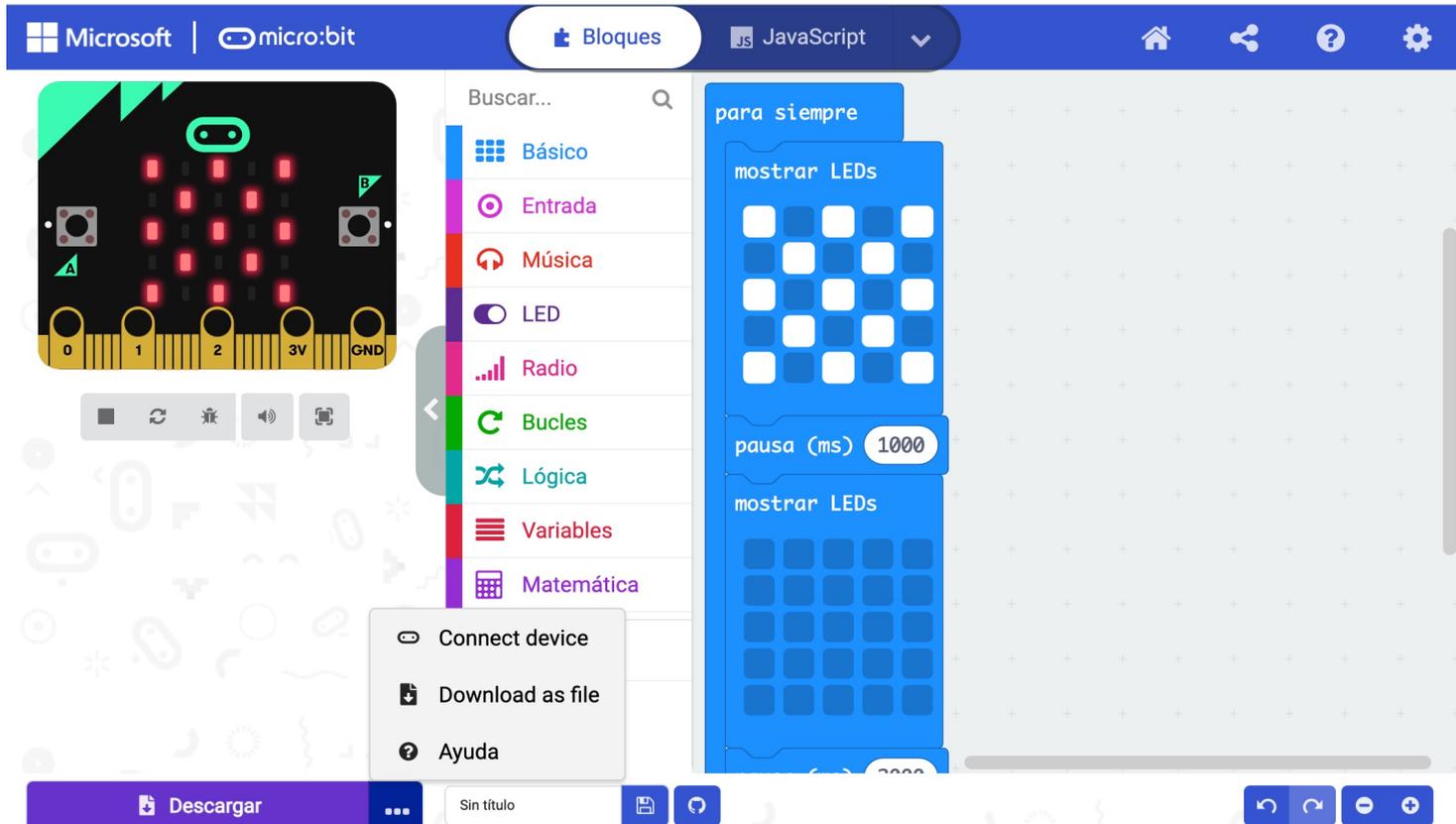
- Los colores del menú de bloques es el mismo que el de los bloques que lo contienen



Paso 2.e: Abriendo el primer programa

□ Descarga el programa en la Tarjeta

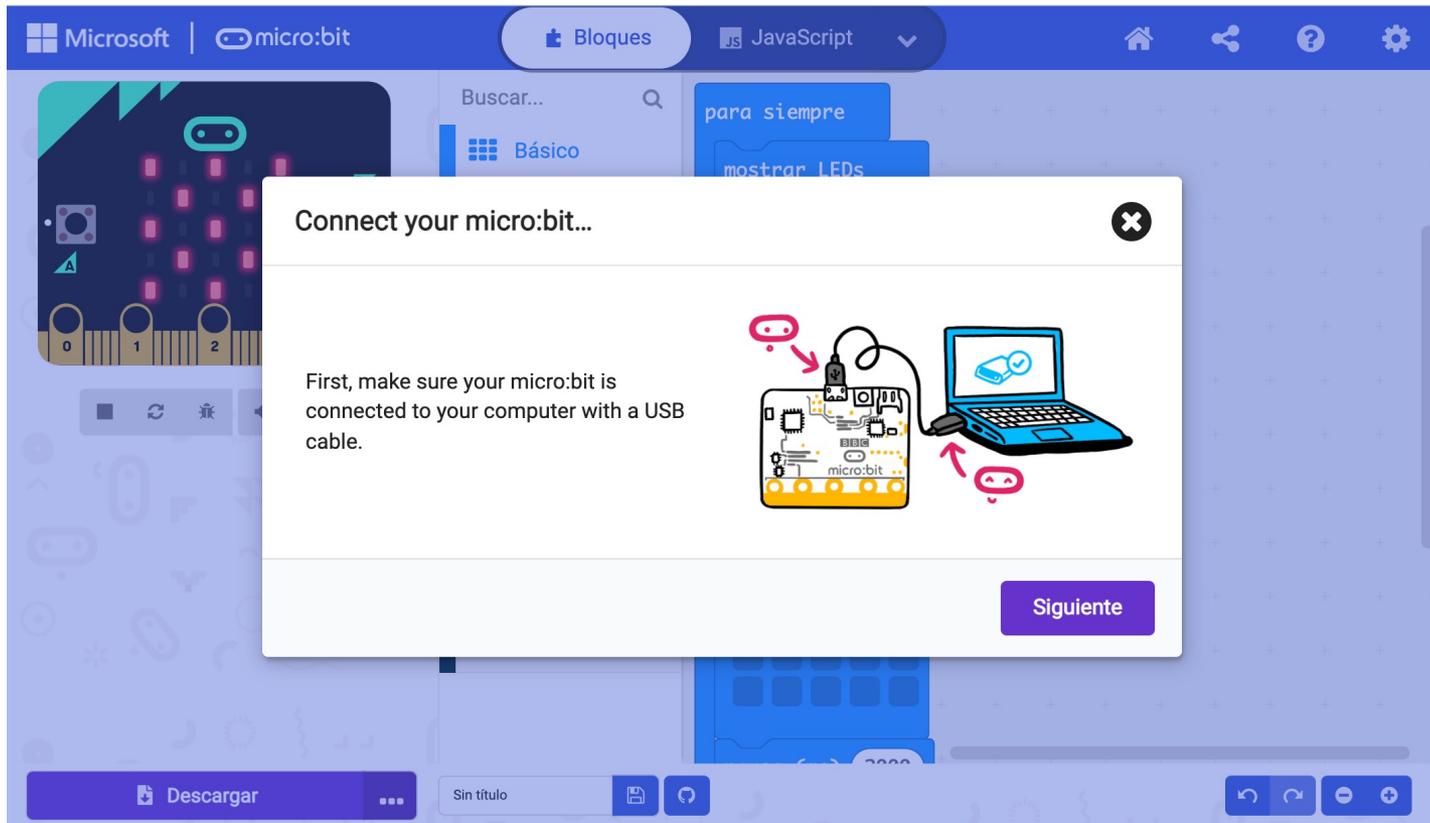
- Una vez hecho el programa hay que descargarlo en la tarjeta para probarlo
 - Conecta la tarjeta MicroBit al ordenador con el cable USB
 - Aparecerá un nuevo disco USB
 - Conectar la tarjeta con el programa



Paso 2.e: Abriendo el primer programa

❑ Descarga el programa en la Tarjeta

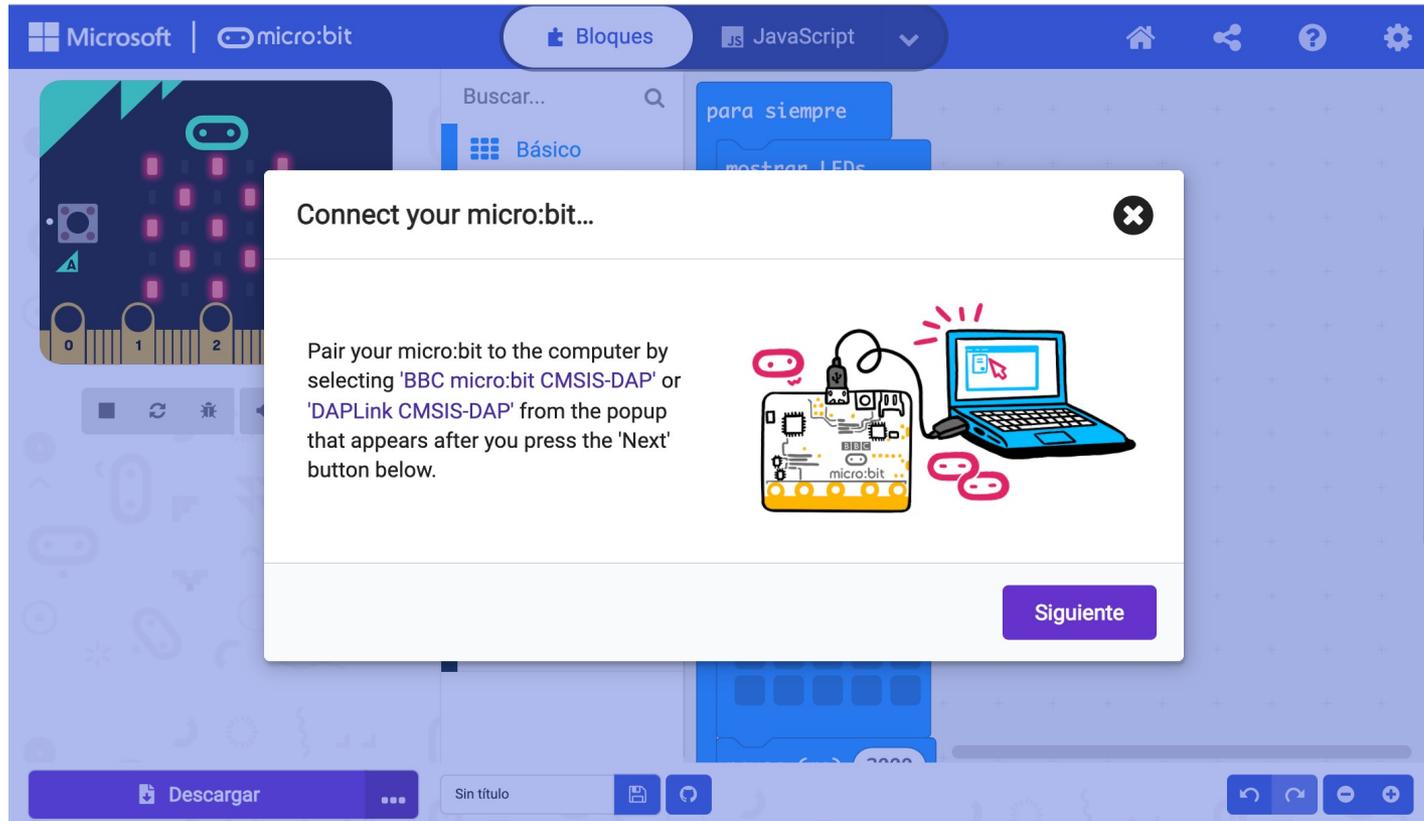
- Una vez hecho el programa hay que descargarlo en la tarjeta para probarlo
 - ❑ Conecta la tarjeta MicroBit al ordenador con el cable USB
 - ❑ Aparecerá un nuevo disco USB
 - ❑ Conectar la tarjeta con el programa



Paso 2.e: Abriendo el primer programa

❑ Descarga el programa en la Tarjeta

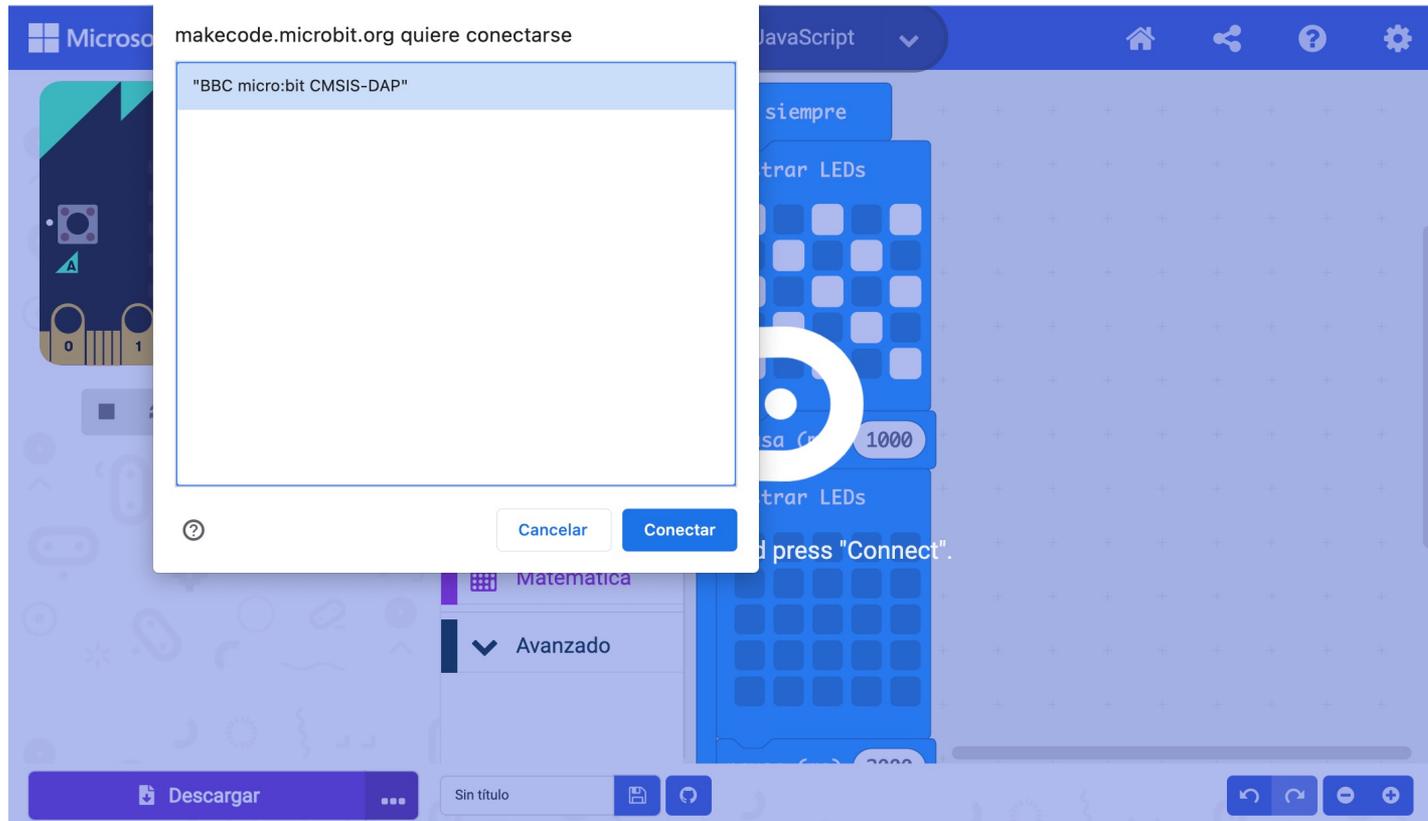
- Una vez hecho el programa hay que descargarlo en la tarjeta para probarlo
 - ❑ Conecta la tarjeta MicroBit al ordenador con el cable USB
 - ❑ Aparecerá un nuevo disco USB
 - ❑ Conectar la tarjeta con el programa



Paso 2.e: Abriendo el primer programa

❑ Descarga el programa en la Tarjeta

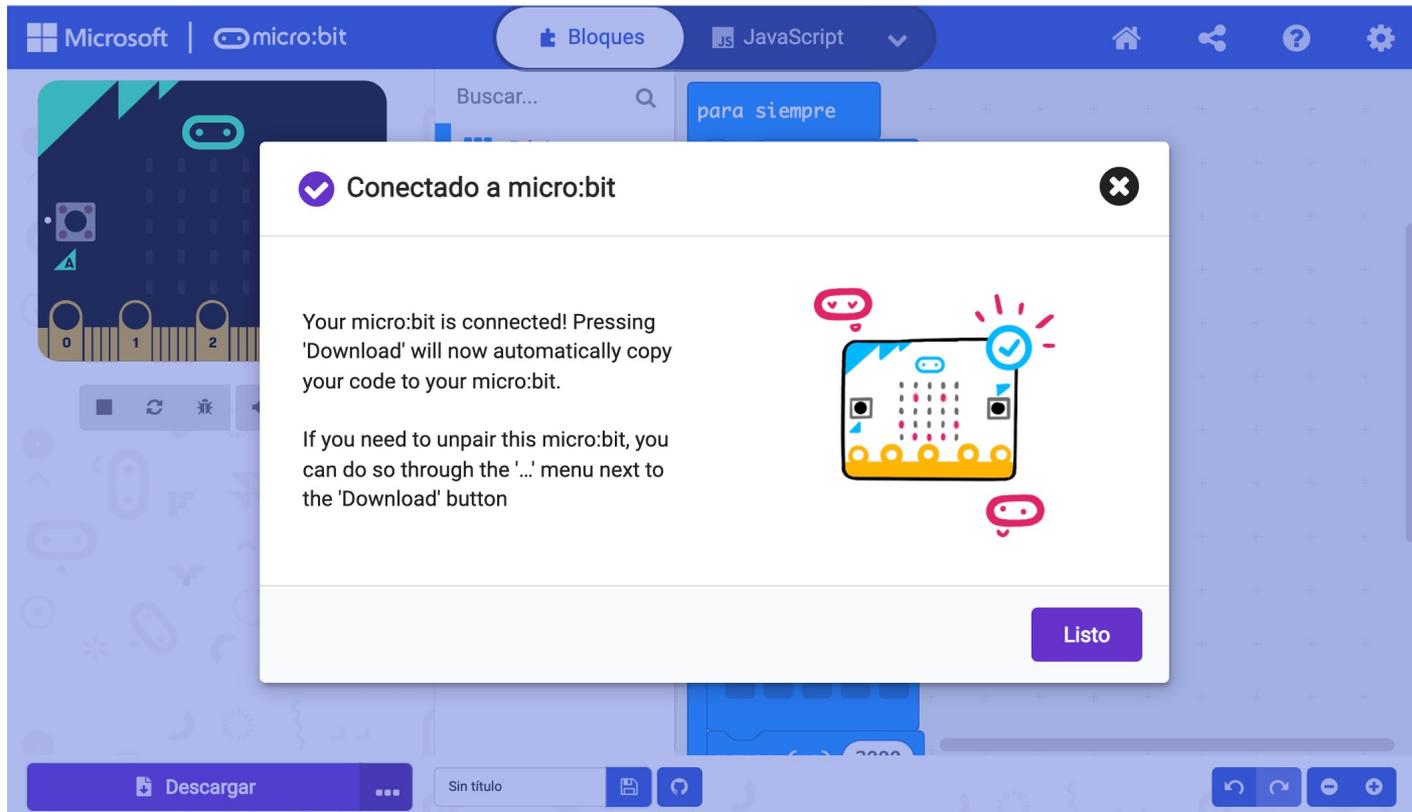
- Una vez hecho el programa hay que descargarlo en la tarjeta para probarlo
 - ❑ Conecta la tarjeta MicroBit al ordenador con el cable USB
 - ❑ Aparecerá un nuevo disco USB
 - ❑ Conectar la tarjeta con el programa



Paso 2.e: Abriendo el primer programa

❑ Descarga el programa en la Tarjeta

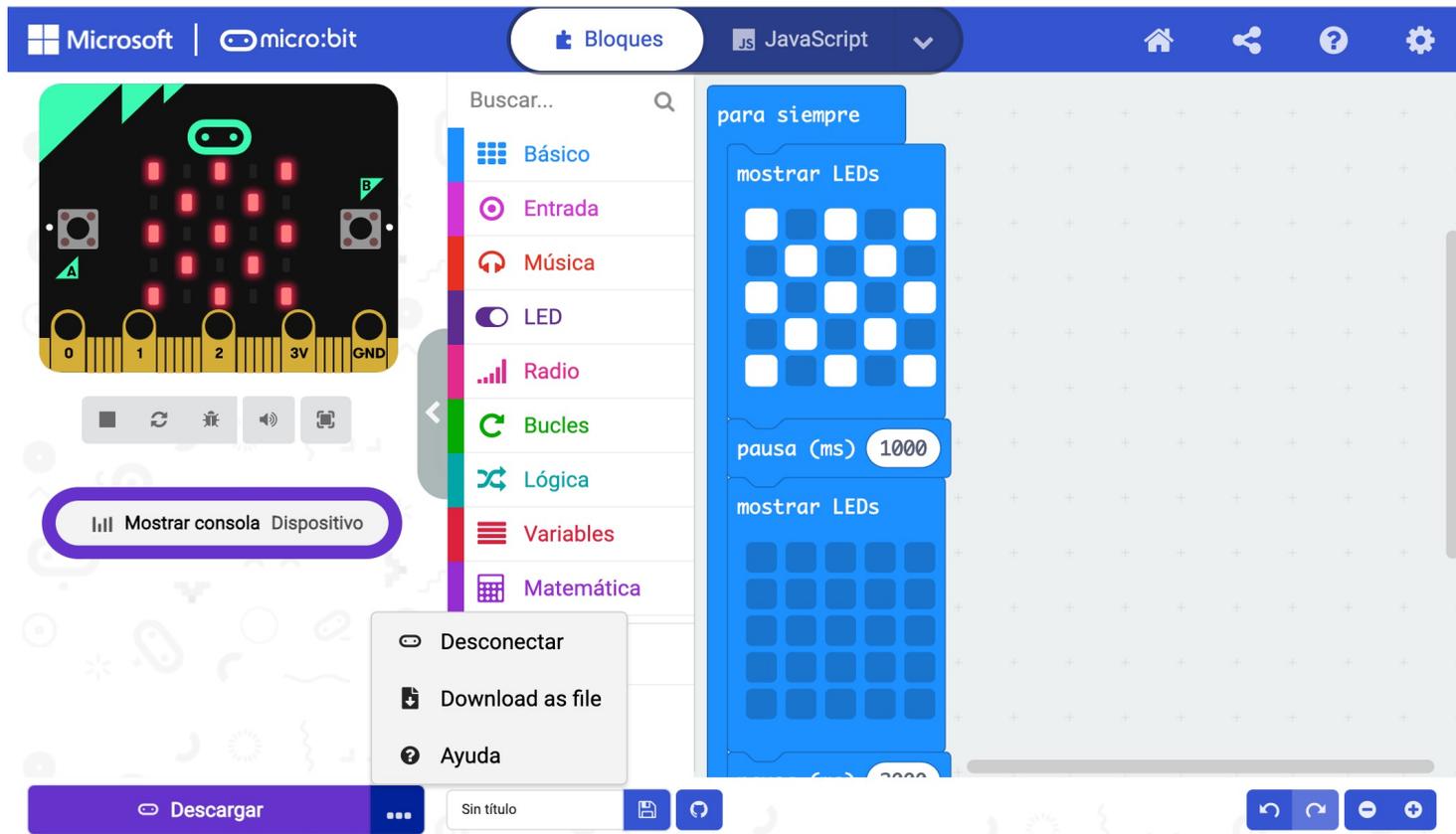
- Una vez hecho el programa hay que descargarlo en la tarjeta para probarlo
 - ❑ Conecta la tarjeta MicroBit al ordenador con el cable USB
 - ❑ Aparecerá un nuevo disco USB
 - ❑ Conectar la tarjeta con el programa



Paso 2.e: Abriendo el primer programa

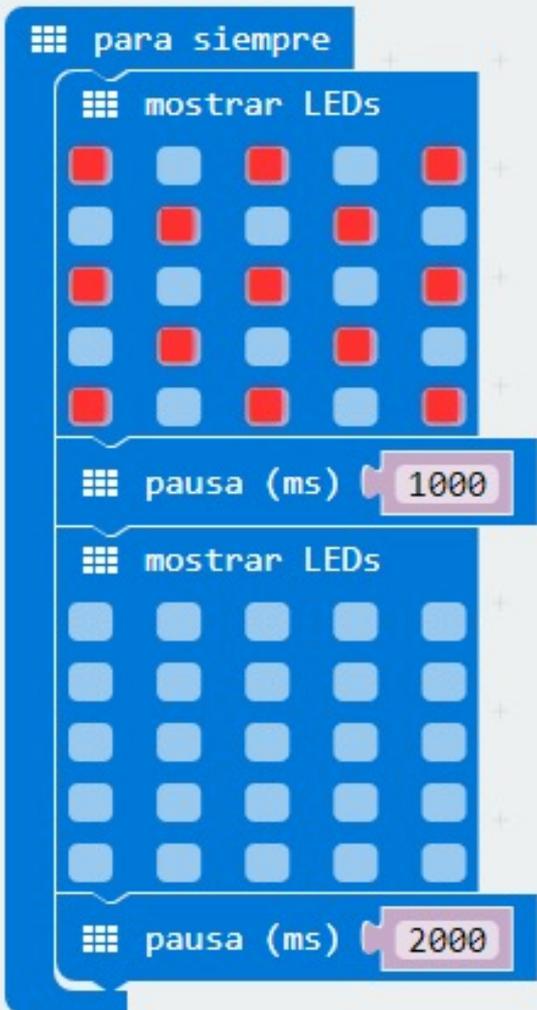
❑ Descarga el programa en la Tarjeta

- Una vez hecho el programa hay que descargarlo en la tarjeta para probarlo
 - ❑ Descargar el programa una vez conectado



Paso 3: Programa el parpadeo de LEDs

- ❑ Carga el programa **Paso_3_EncendiendoUnLED.hex**



Observa que realiza el programa:

Enciende los LEDs durante **1000 milisegundos**.
Apaga los LEDs durante **2000 milisegundos**.

- ❑ **Ejercicio**

Modifica el programa para que parpadee más rápido. Pruébalo.

- ❑ **Ejercicio**

Cambia el dibujo a mostrar por la pantalla

- ❑ **Ejercicio**

Haz que antes de nada muestre un corazón durante 2 segundos una sola vez

- ❑ **Ejercicio**

Haz que se muestre tu nombre al inicio.

Paso 4.a: Probando el sensor de distancia

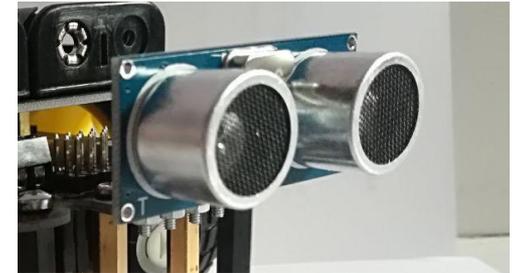
- Abre el **Paso_4_SensorDeDistancia.hex**

Para utilizar el sensor de distancia ...

Hay que utilizar un paquete de objetos llamado **SONAR** que contiene un bloque llamado **ping** que indica dónde está conectado el sensor y que tipo de medida se quiere que haga.



El bloque "ping" lee la **DISTANCIA** leída por el sensor de ultrasonidos



Paso 4.b: Probando el sensor de distancia

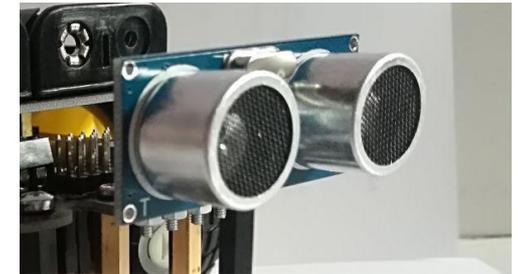
- Abre el **Paso_4_SensorDeDistancia.hex**

Para utilizar el sensor de distancia ...

Hay que utilizar un paquete de objetos llamado **SONAR** que contiene un bloque llamado **ping** que indica dónde está conectado el sensor y que tipo de medida se quiere que haga.



El bloque "ping" lee la **DISTANCIA** leída por el sensor de ultrasonidos



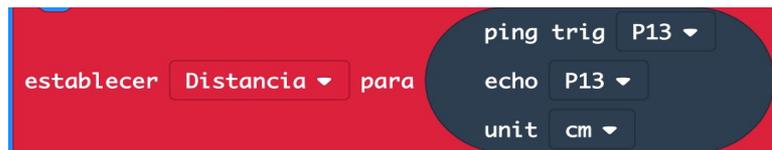
También hay que utilizar una variable donde guardar el dato medido

Las variables sirven para almacenar datos

Los bloques relacionados en: **Variables**

El **Bloque Establecer** guarda un valor en una variable.

Las variables se pueden definir en **"Ejecutar una variable"**

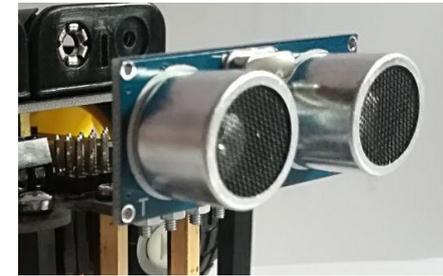


Paso 4.c: Probando el sensor de distancia

- Abre y descarga el **Paso_4_SensorDeDistancia_v3.hex** en el robot

Observa lo que realiza el programa

Envía al ordenador el valor de distancia medido tras introducirlo en la variable distancia. En el "Consola del dispositivo" deberían aparecer números.



Ejercicio

Comprueba cómo cambia la medida de distancia al poner la mano delante del robot a diferentes distancias.

Paso 4.c: Probando el sensor de distancia

- Abre y descarga el **Paso_4_SensorDeDistancia_v3.hex** en el robot

Observa lo que realiza el programa

Envía al ordenador el valor de distancia medido tras introducirlo en la variable distancia. En el "Consola del dispositivo" deberían aparecer números.

The screenshot shows the micro:bit simulator interface. On the left is a virtual representation of the micro:bit board with pins labeled 0, 1, 2, 3V, and GND. The top bar includes 'Microsoft | micro:bit', 'Bloques', and 'JavaScript'. On the right, there are controls for the device: 'Dispositivo', a pause button, a download button, and a refresh button. Below these is a graph showing a signal that fluctuates between 0.00 and 238.00. A red box highlights the console output area, which shows the following data:

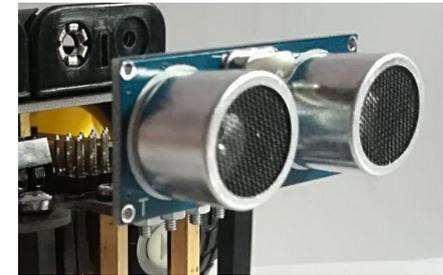
9	42
41	42
2	42
43	42
26	42

Paso 4.c: Probando el sensor de distancia

- Abre y descarga el **Paso_4_SensorDeDistancia_v3.hex** en el robot

Observa lo que realiza el programa

Envía al ordenador el valor de distancia medido tras introducirlo en la variable distancia. En el "Monitor Serie" deberían aparecer números.



Ejercicio

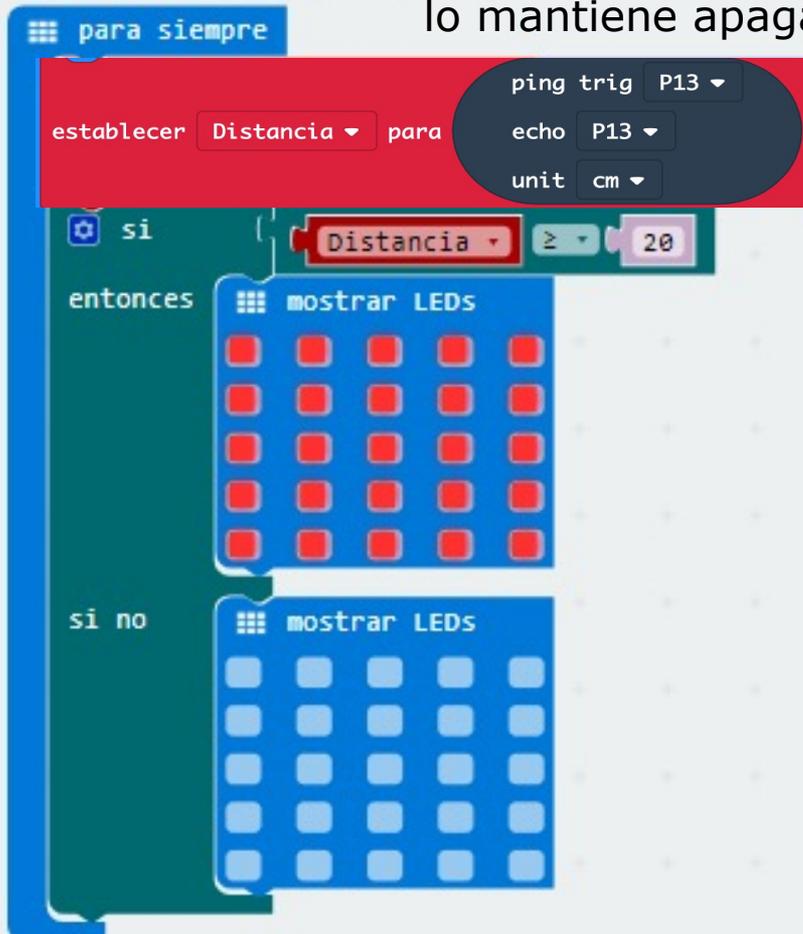
Cambia las unidades de medida en us para ver el tiempo que tarda el ultrasonido en rebotar

Paso 5: Detectando un objeto

- ❑ Carga el programa **Paso_5_DetectandoObjeto.hex**

Observa lo que realiza el programa

Si detecta un objeto a más de 20 cm enciende el display y si no lo mantiene apagado.



❑ Ejercicio

- Modifica el programa para encienda la pantalla cuando la distancia sea **menor** de 10cm.

❑ Ejercicio

- Modifica el programa para que identifique tres distancias diferentes.

❑ Ejercicio

- Modifica el programa para que el LED parpadee cuando detecte un objeto cercano.

Paso 6.a: Moviendo los motores

- Carga el programa **Paso_6_MoviendoMotores.hex**

Para utilizar los motores...

Hay que utilizar un paquete de objetos llamado **Bitbot** que se puede descargar de: <https://github.com/4tronix/BitBot>

Se utilizan los bloques Drive Motor indicando el motor (derecho, izquierdo o ambos) y la velocidad que puede tomar valores de 0 a 100% para ambos sentidos



Si se considerara interesante, el valor de la velocidad podría obtenerse del contenido de una variable.

Paso 6.b: Moviendo los motores

- ❑ Carga el programa **Paso_6_MoviendoMotores.hex**

Observa lo que realiza el programa

```
para siempre
  move right motor(s) forward at speed 60 %
  pausa (ms) 1000
  move right motor(s) forward at speed 0 %
  pausa (ms) 1000
  move left motor(s) reverse at speed 60 %
  pausa (ms) 1000
  move left motor(s) reverse at speed 0 %
  pausa (ms) 1000
```

Mueve el motor **derecho** a una velocidad de **60%** durante un **1** segundo

Luego **para** el motor **derecho**

Mueve el motor **izquierdo** a una velocidad de **60** de sentido **contrario**

Para el motor **izquierdo**

Paso 6.c: Moviendo los motores

- ❑ Carga el programa **Paso_6_MoviendoMotores.hex**

```
para siempre
  move right motor(s) forward at speed 60 %
  pausa (ms) 1000
  move right motor(s) forward at speed 0 %
  pausa (ms) 1000
  move left motor(s) reverse at speed 60 %
  pausa (ms) 1000
  move left motor(s) reverse at speed 0 %
  pausa (ms) 1000
```

- ❑ **Ejercicio:**

- Cambia el programa para que se mueva todo el robot en línea recta 2 segundos hacia delante y 2 segundos hacia atrás.

- ❑ **Ejercicio**

- Modifica el programa para que el robot realice la trayectoria de un cuadrado girando en las esquinas.

Paso 7.a: Detectando obstáculos

- ❑ Carga el programa **Paso_7_DetectandoObstaculos.hex**

Observa lo que realiza el programa

Si la **distancia** (que lee el ultrasonidos) es **mayor que 20 cm**, **se mueven las dos** ruedas.

Si no, es decir, si la distancia es **menor o igual que 20 cm**, se **paran**.

```
para siempre
  establecer Distancia para ping trig P13
  echo P13
  unit cm
  si Distancia > 20 entonces
    move both motor(s) forward at speed 40 %
  si no
    move both motor(s) forward at speed 0 %
```

❑ Ejercicio

Modifica el programa para que el robot gire en el sitio y, cuando detecte un **objeto a menos de 10 cm**, vaya a por él en línea recta.

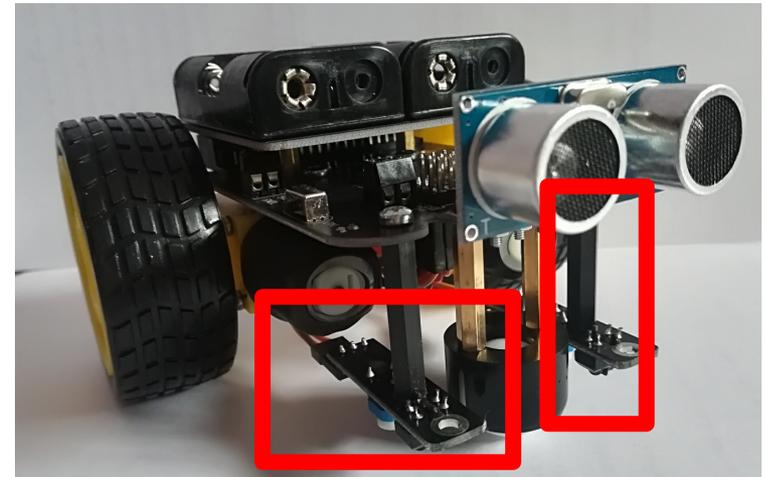
Paso 8.a: Leyendo los sensores de suelo

- ❑ Carga el programa **Paso_8_SensoresSuelo.hex**

Para utilizar los sensores de suelo...

Hay que utilizar un paquete de objetos llamado **Bitbot** que se puede descargar de: <https://github.com/4tronix/BitBot>

Se utilizan los bloques Read Line Sensor indicando el sensor (derecho o izquierdo). Puede tomar el valor 0 (blanco) o 1 (negro).



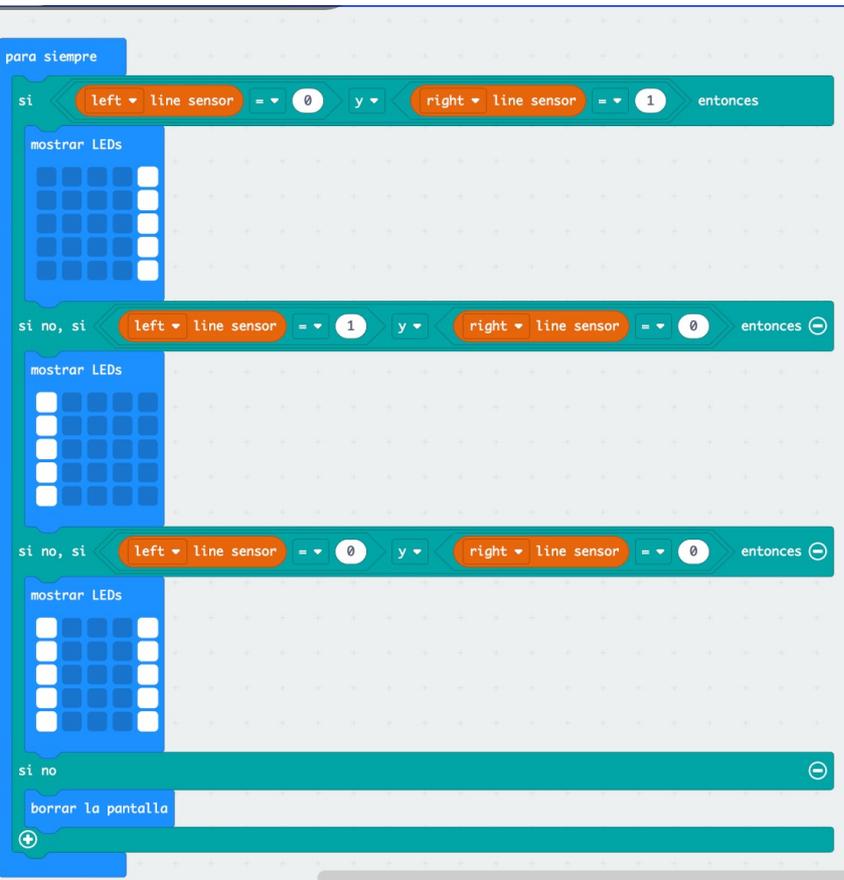
Los sensores tienen un led que se enciende cuando detecta blanco y se apaga cuando detecta negro. El umbral puede ajustarse con un destornillador

Paso 8.b: Leyendo los sensores de suelo

- ❑ Carga el programa **Paso_8_SensoresSuelo.hex**

Observa que realiza el programa:

Representa en pantalla una línea del lado del sensor que detecta blanco



Si (detecta blanco solo sensor izquierdo)
Pinta línea a la derecha

Si no, si (detecta solo sensor derecho)
Pinta línea a la izquierda

Si no, si detectan los dos sensores
Pinta dos líneas

Si no
No pinta nada

Paso 9.a: Nos movemos sin salirnos

- Teniendo en cuenta todo lo visto anteriormente, realiza los siguientes ejercicios:

□ Ejercicio:

- Realiza un programa que haga que el robot avance cuando esté sobre negro y se detenga cuando llegue a la línea blanca.

□ Ejercicio:

- Realiza un programa que haga para que el robot avance hasta que detecte blanco y en ese momento, retroceda durante un cierto tiempo, gire, y siga avanzando.
- El robot debería moverse dentro del Tatami sin salirse.

□ Ejercicio:

- Modifica el programa para que, si mientras está girando detecta un oponente, vaya a por él.
- Este ya es un programa que detecta al oponente

Paso 9.b: Luchador de sumo

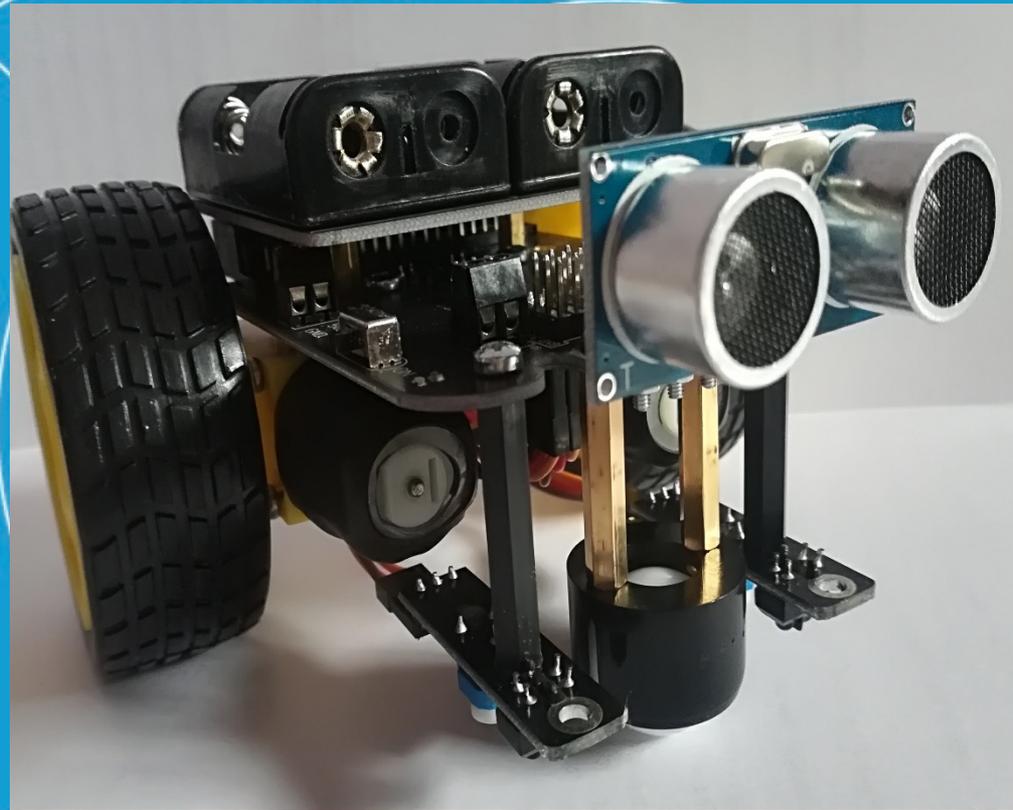
- Ya estás listo para programar el luchador de sumo. Busca al oponente, detéctale, empújale, pero no te salgas tú solo.



Mini-Taller de Robótica Móvil

Programa un robot de sumo

Ven a la Universidad de Alcalá
y construye TU robot



Universidad
de Alcalá



Departamento de
electrónica



GOBIERNO
DE ESPAÑA

MINISTERIO
DE ECONOMÍA, INDUSTRIA
Y COMPETITIVIDAD

FECYT



FUNDACIÓN ESPAÑOLA
PARA LA CIENCIA
Y LA TECNOLOGÍA

Actividad orientada a estudiantes de Enseñanza Secundaria y Bachillerato