

# Mini-Taller de Robótica Móvil

## Programa un robot de sumo



Universidad  
de Alcalá



Departamento de  
**e**lectrónica

# Mini-Taller de Robótica Móvil

## Programa un robot de sumo

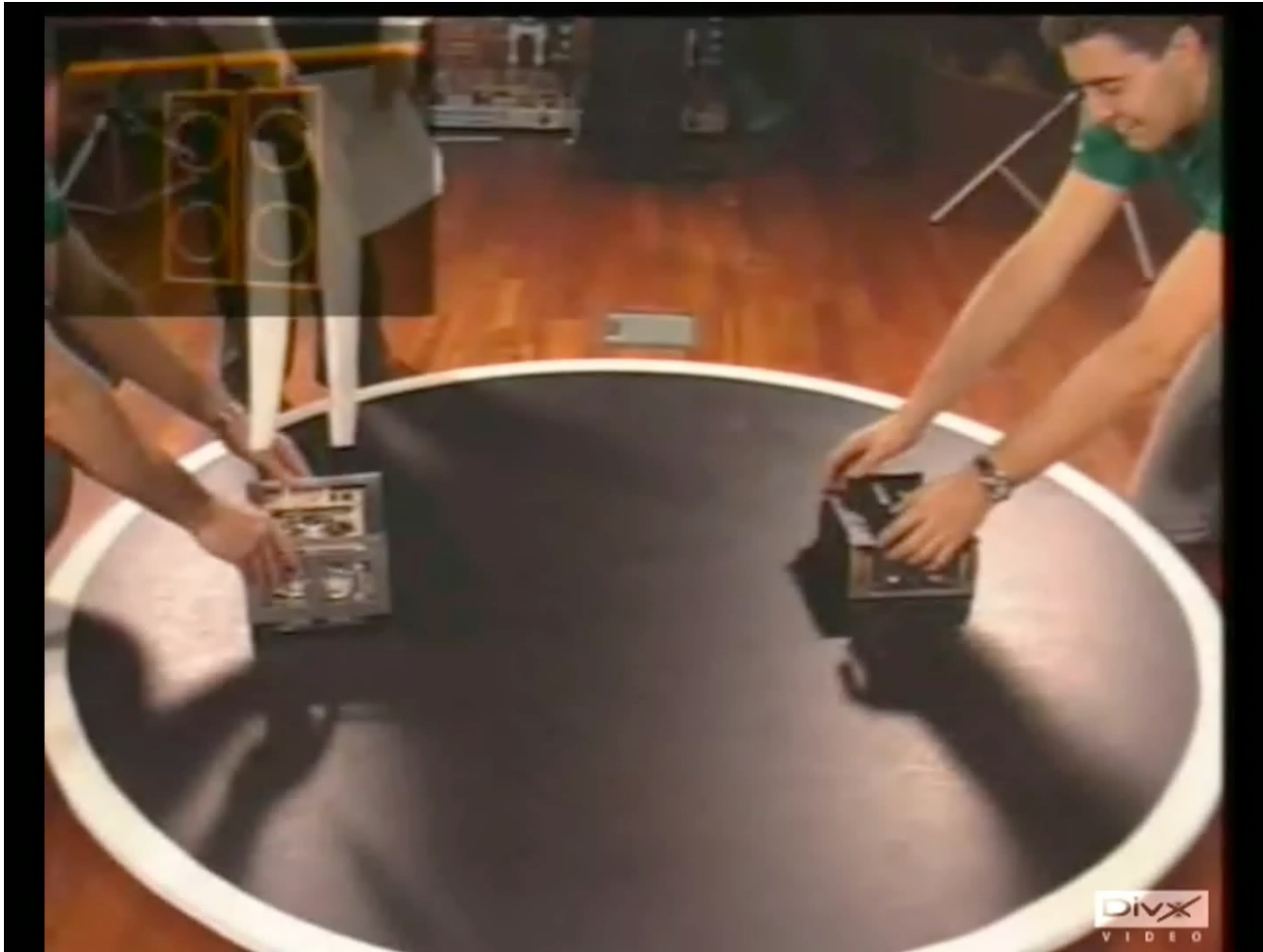


Universidad  
de Alcalá



Departamento de  
**e**lectrónica

# Competición de Sumo



<https://www.youtube.com/watch?v=Pe2VDgQFvio>

**Hispabot'04 Final Panzer - Bestia**

# Planificación de Actividades

## Introducción del Taller

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

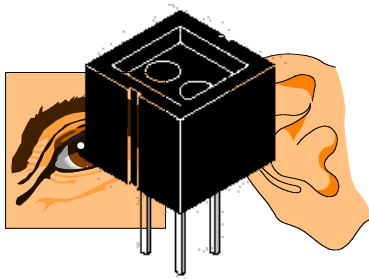
## Taller de robótica

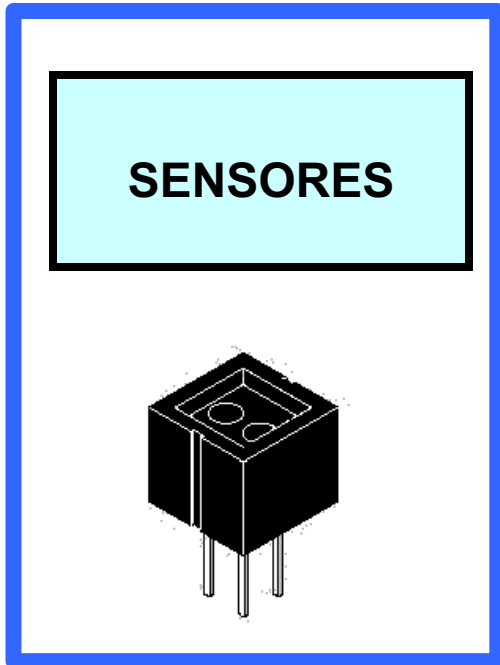
- Moviendo el robot
- Detectando obstáculos
- Detectando el tatami
- Compitiendo en Sumo



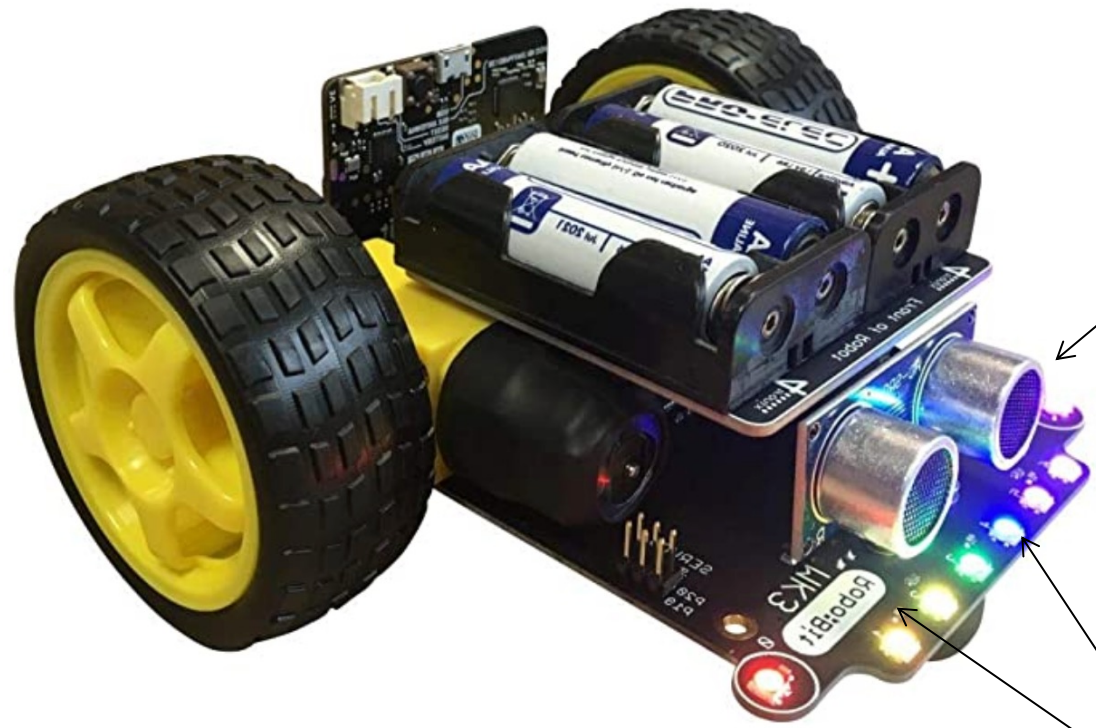


**SENSORES**





# Sensores de TuBot

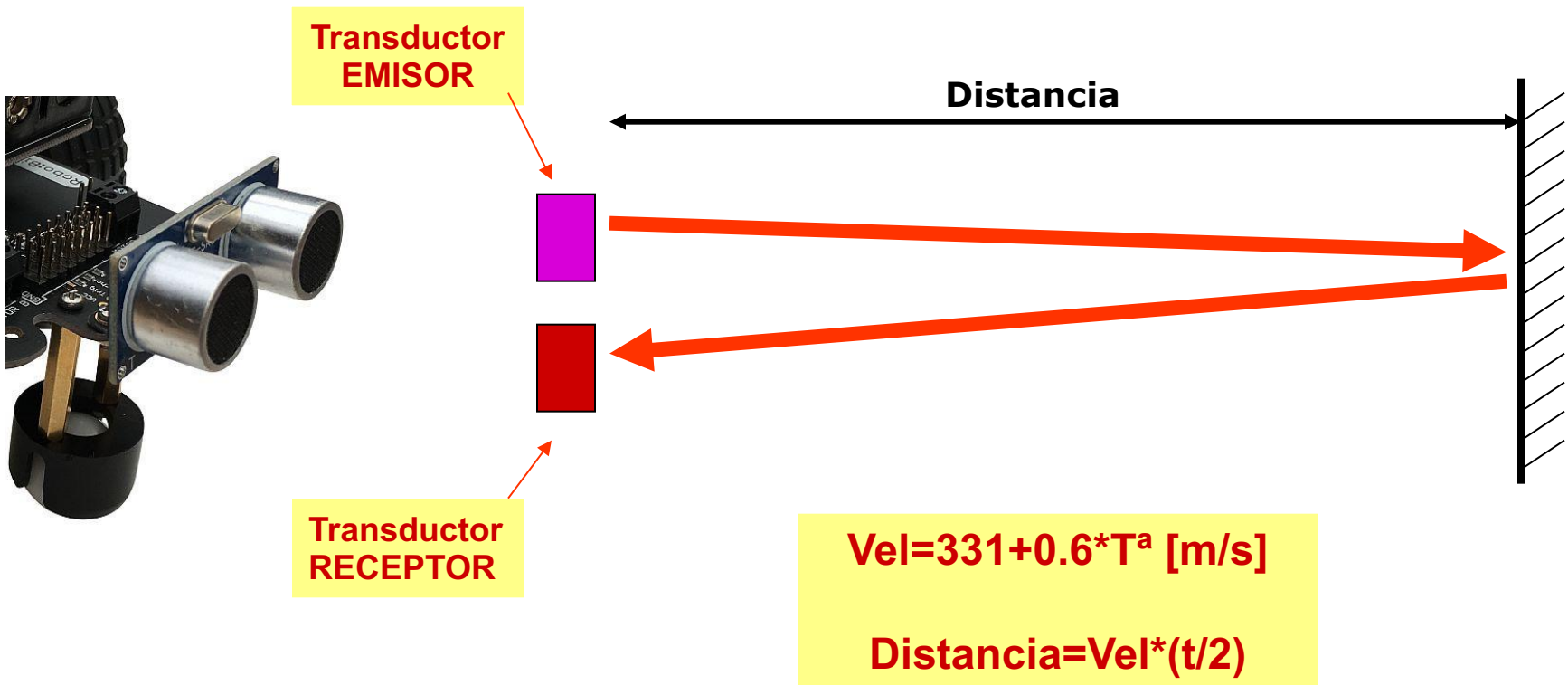


Sensor de Distancia  
por Ultrasonidos

Sensor de Suelo  
Sensores infrarrojos reflectivos

# Medida de Distancia: Ultrasonidos

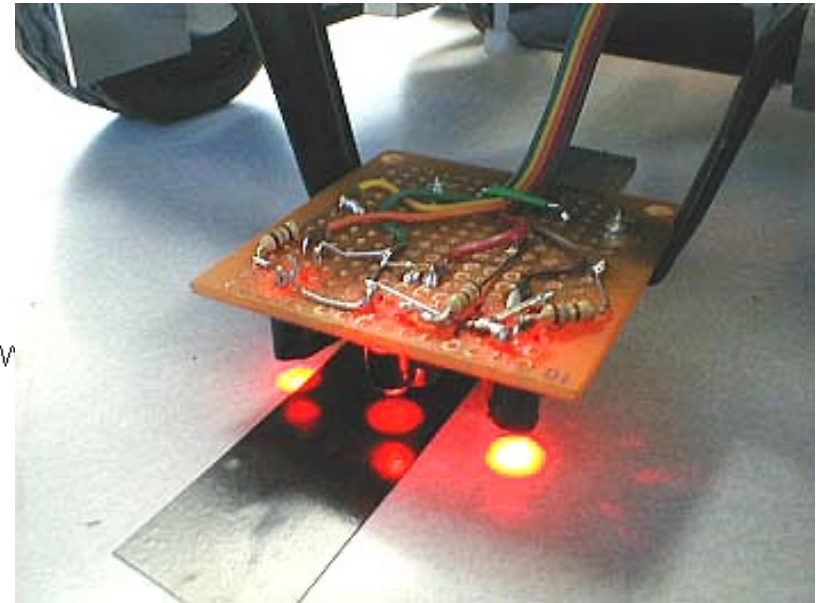
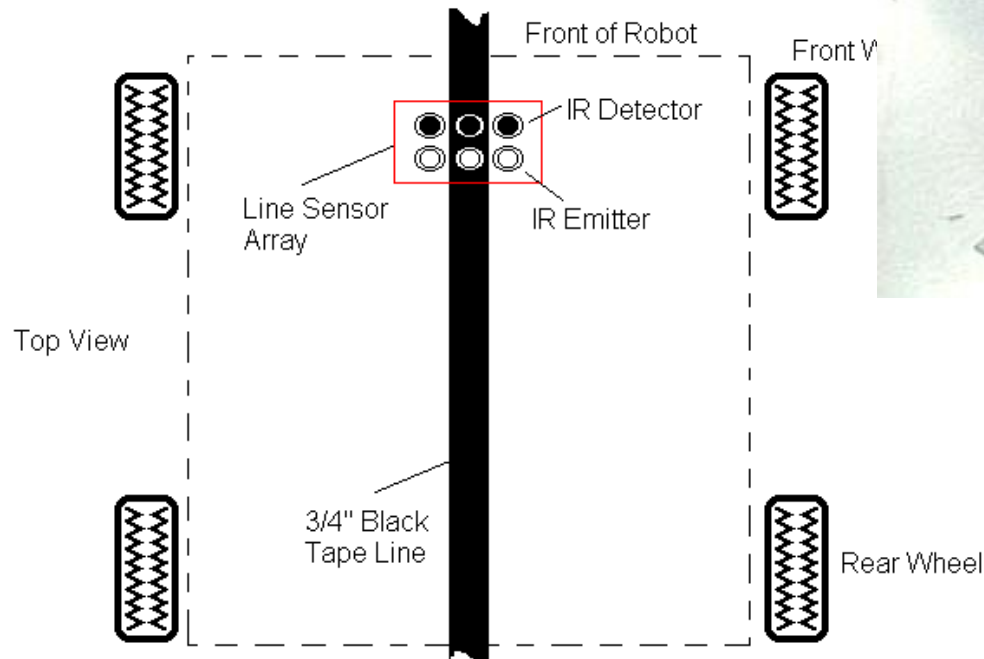
- El sonido tiene una velocidad de propagación de 330m/s aprox.
- Se mide el tiempo que tarda el sonido en ir y volver





# Detección de Línea o Suelo

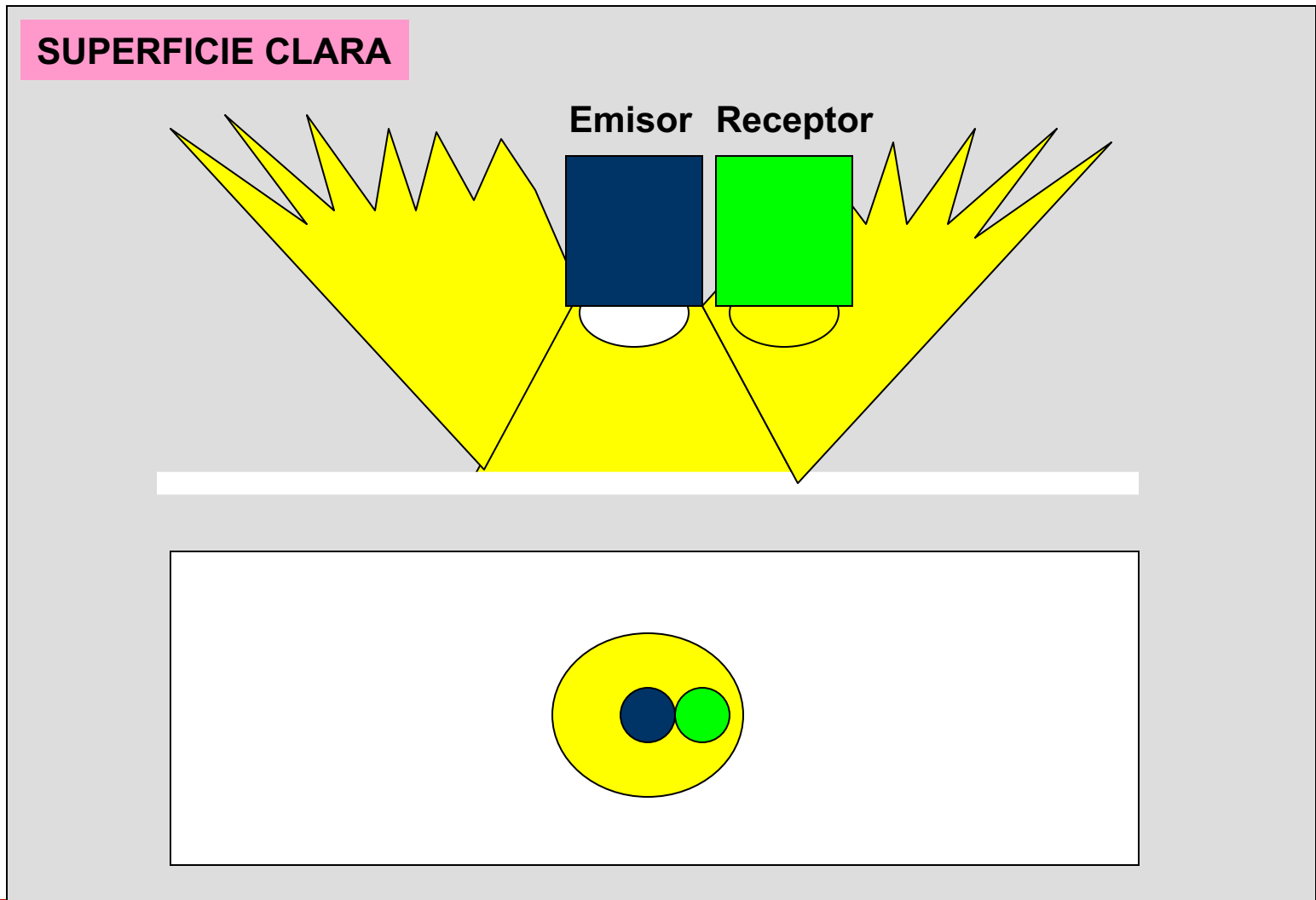
- Sensor de infrarrojos reflectivo: detección de blanco o negro



# Detección de Línea o Suelo

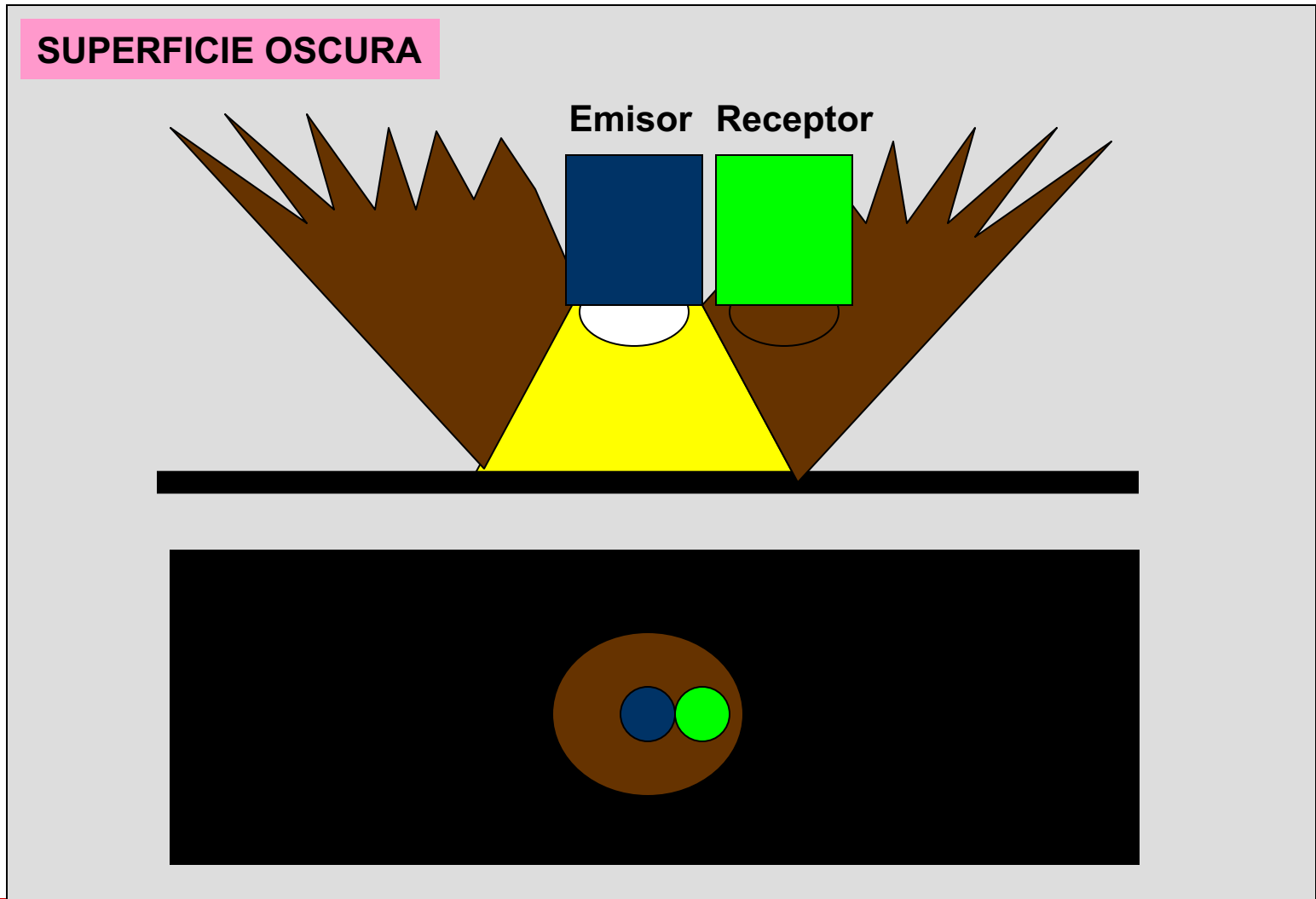


- Sensor de infrarrojos reflectivo: detección de blanco o negro

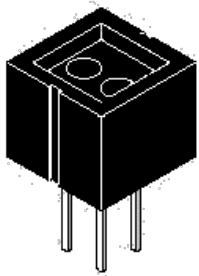


# Detección de Línea o Suelo

- Sensor de infrarrojos reflectivo: detección de blanco o negro



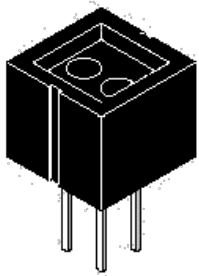
**SENSORES**



**ACTUADORES**



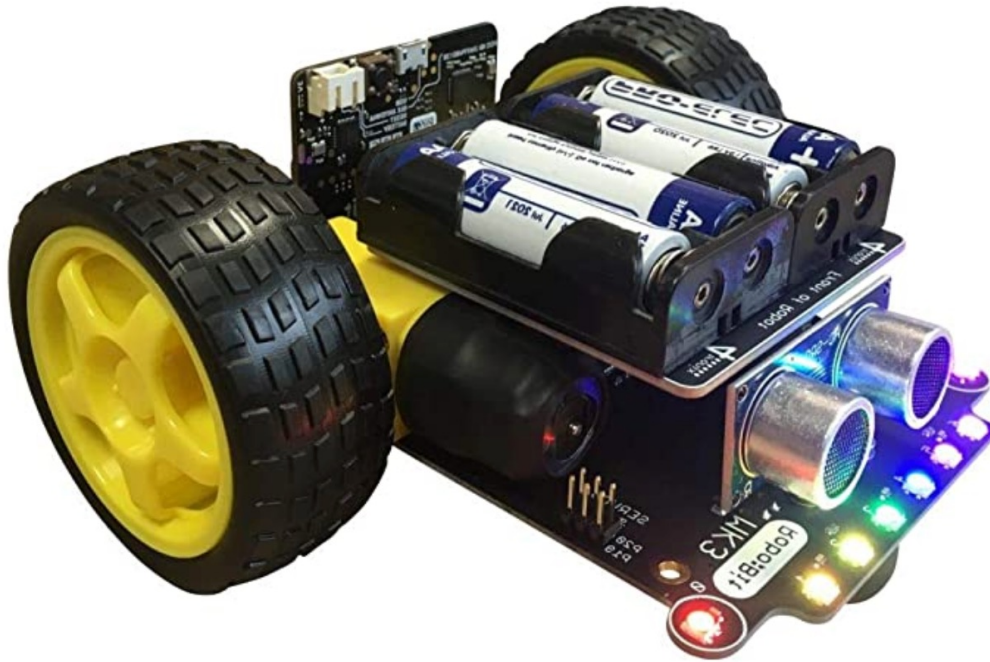
**SENSORES**



**ACTUADORES**

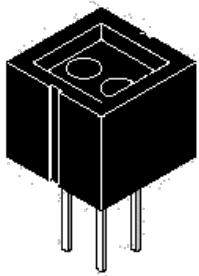
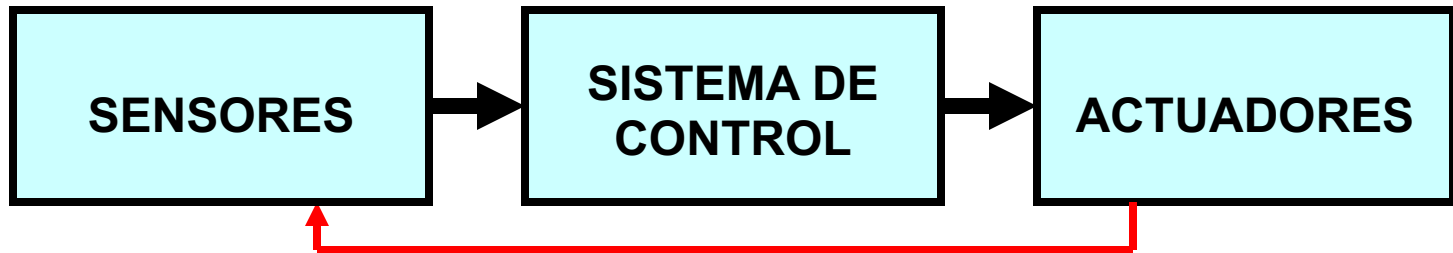


# Actuadores de TuBot

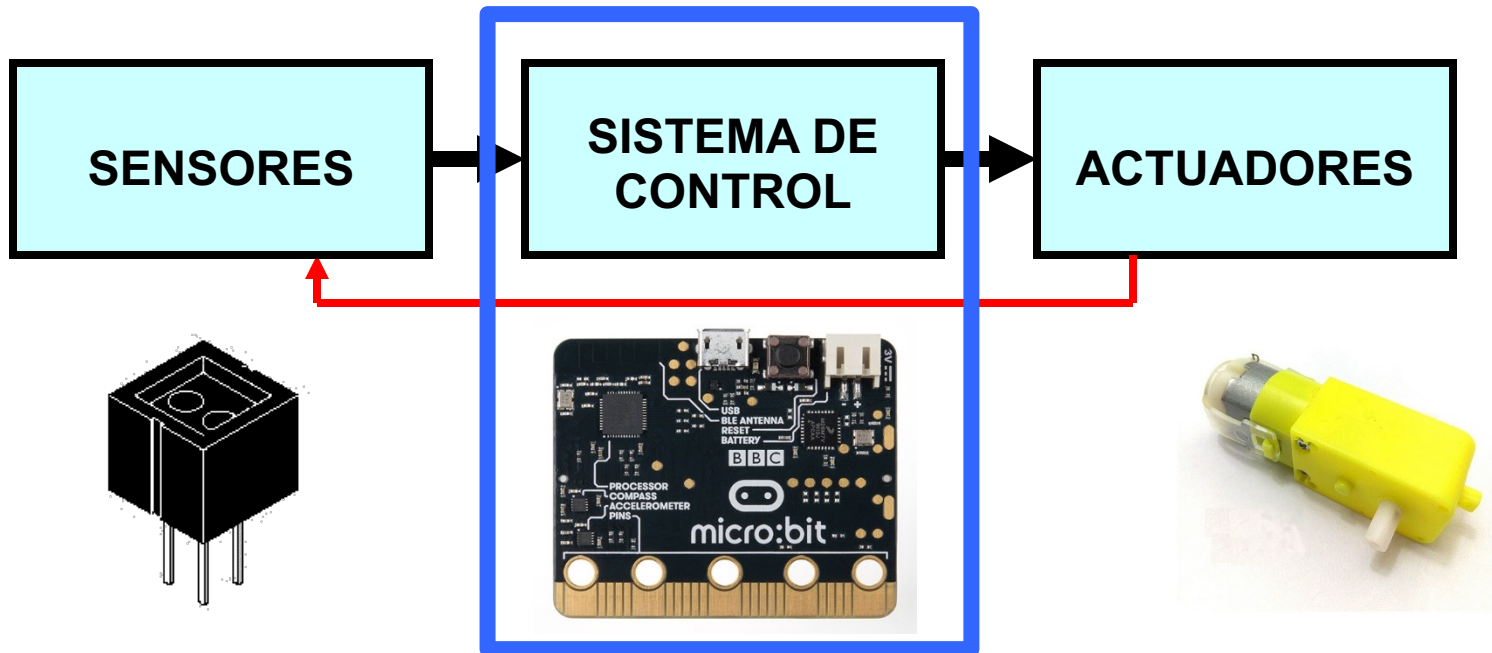


Dos motores

# Elementos de un Robot Móvil Autónomo



# Elementos de un Robot Móvil Autónomo





# Sistema Basado en Microprocesador

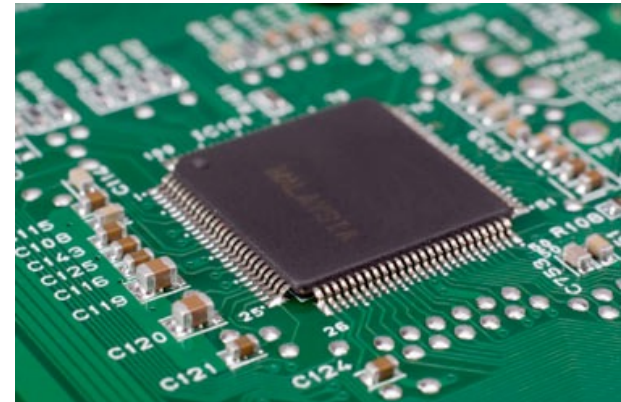
- ❑ El **Microprocesador** es el cerebro de un ordenador
- ❑ Para funcionar necesita un **Programa** donde se indican las **órdenes** que el procesador debe seguir.
- ❑ **Elementos** de un Sistema basado en Microprocesador:
  - CPU (Central Processing Unit)
  - Memoria
  - Periféricos



Source: <http://www.callegranvia.com/images/product/375/1b78d5c0b624c9a3c966e0854829dfe6.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**)

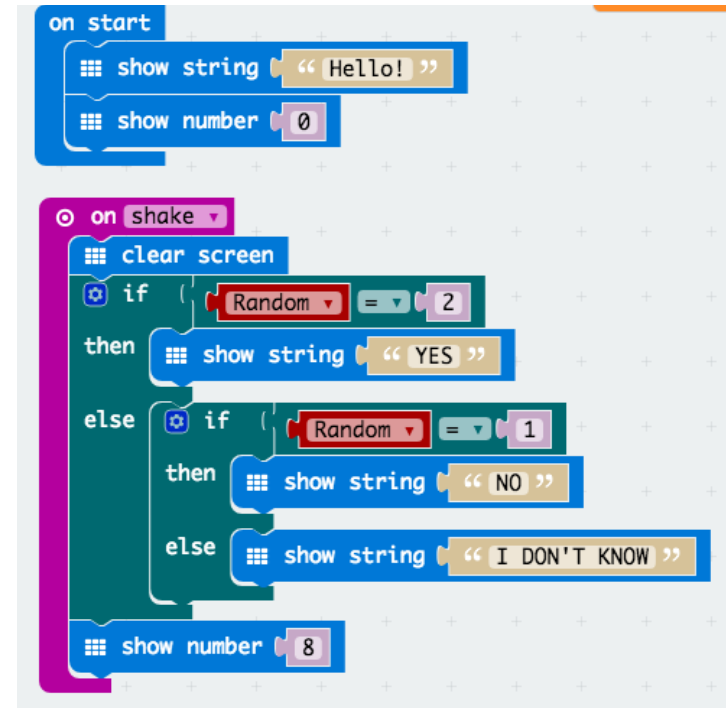


Source: <http://pacotraver.files.wordpress.com/2011/11/interprete.jpg>

[http://2.bp.blogspot.com/\\_Pm8qvnCsVOI/TCwv\\_SAuznI/AAAAAAAAA4/9asQgJGiQMw/s1600/MICRO.jpg](http://2.bp.blogspot.com/_Pm8qvnCsVOI/TCwv_SAuznI/AAAAAAAAA4/9asQgJGiQMw/s1600/MICRO.jpg)

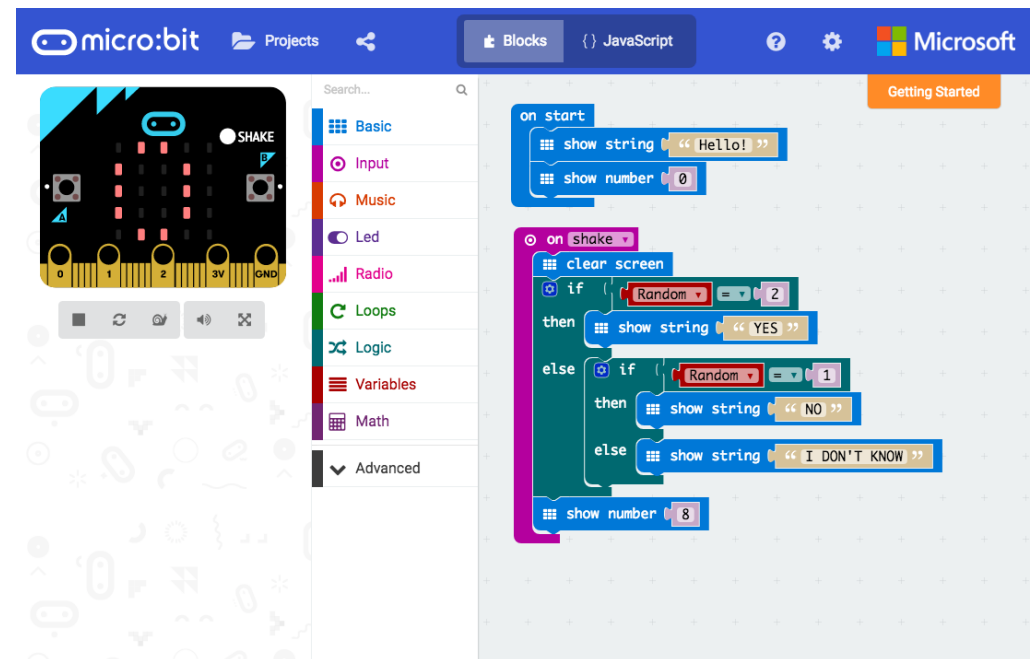
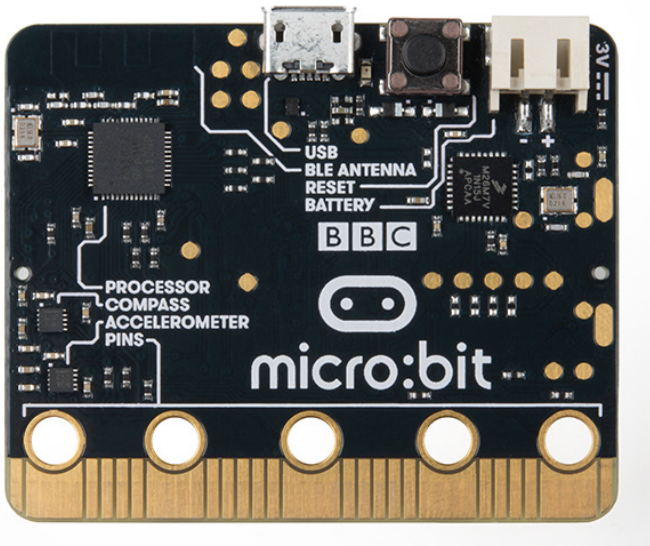
# Programa y compilador

- ❑ 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**)



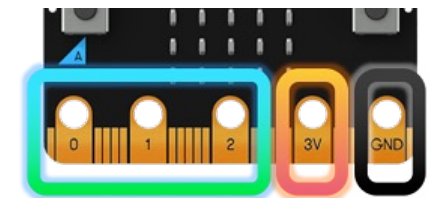
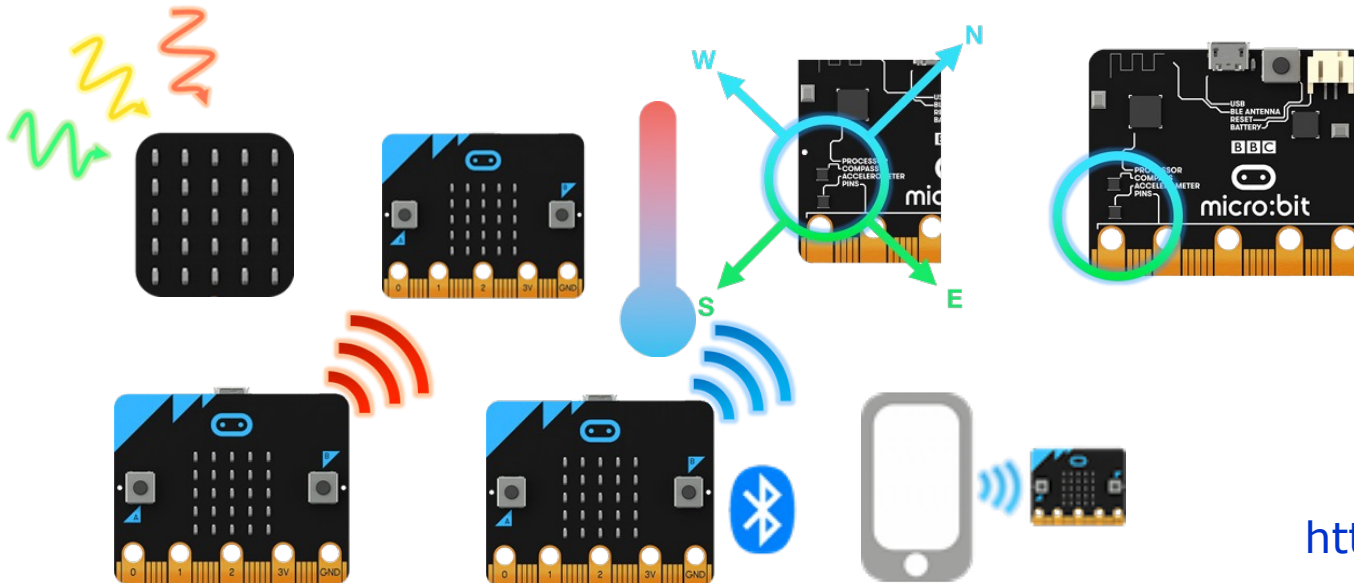
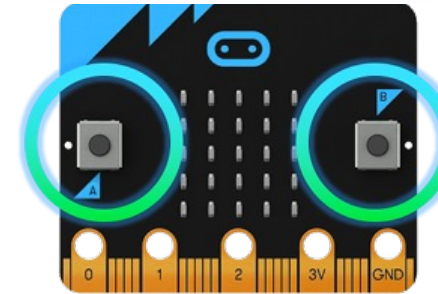
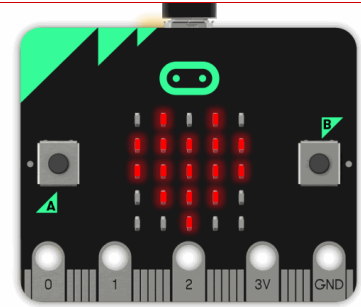
# MICROBIT: buen sistema de iniciación

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



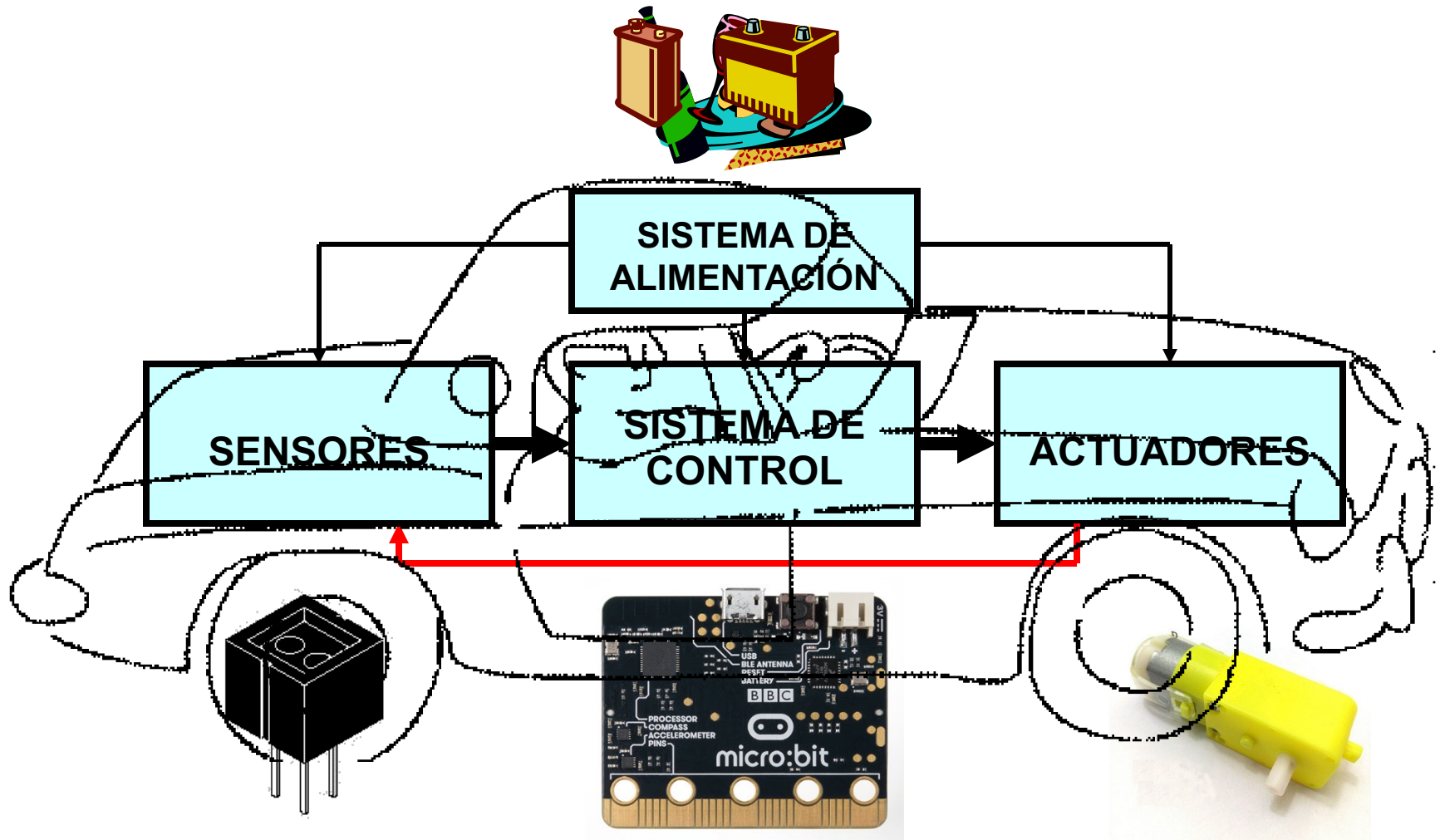
# MICROBIT: buen sistema de iniciación

- ❑ 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

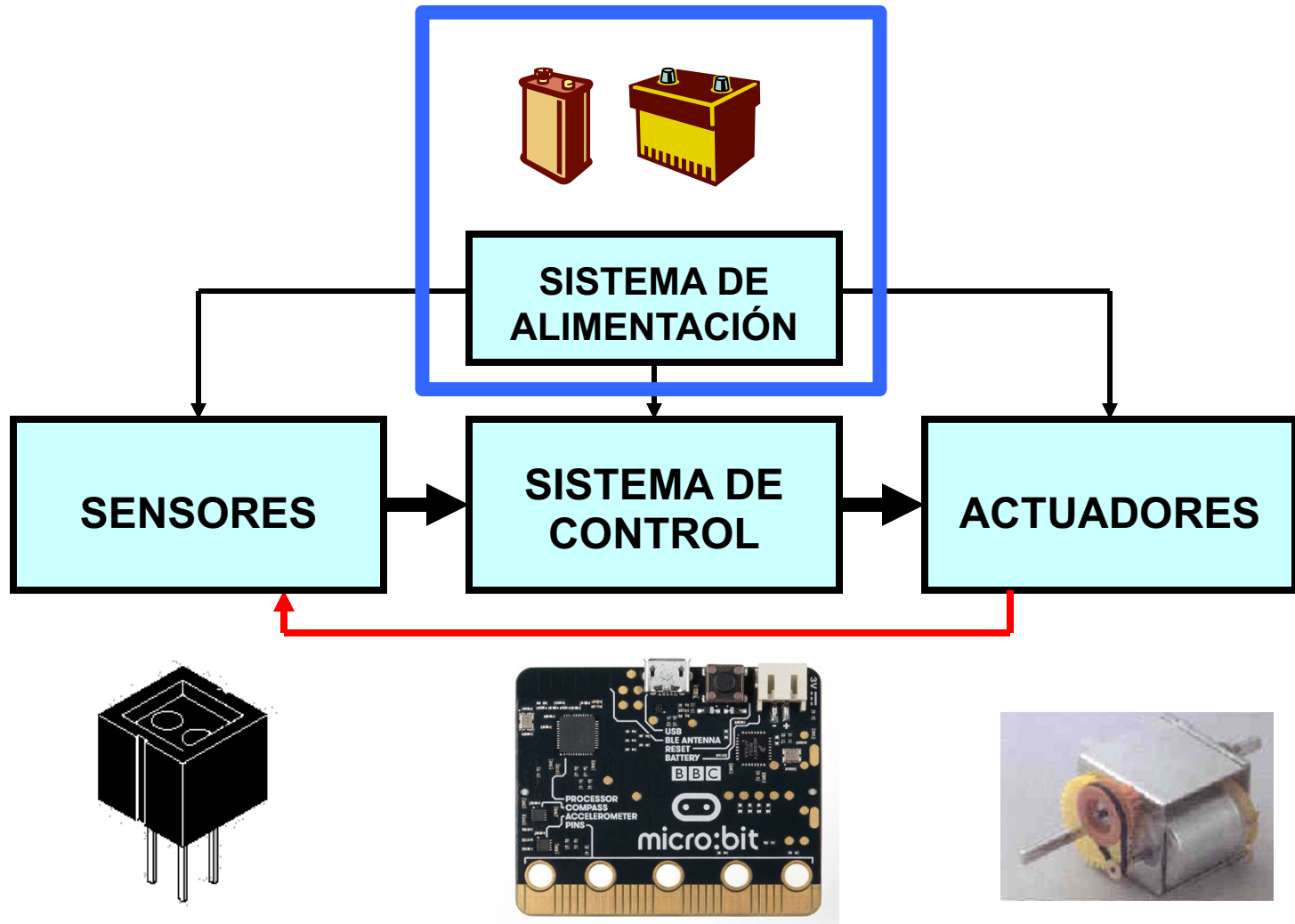


<http://microbit.org/es/>

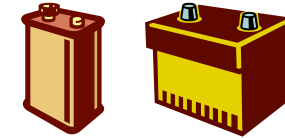
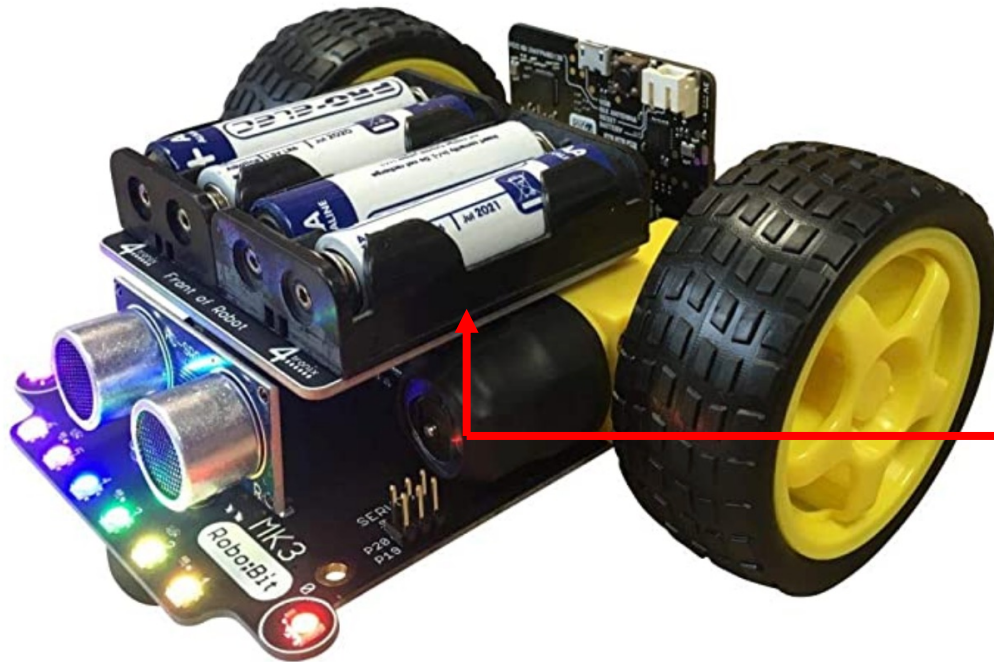
# Elementos de un Robot Móvil Autónomo



# Elementos de un Robot Móvil Autónomo



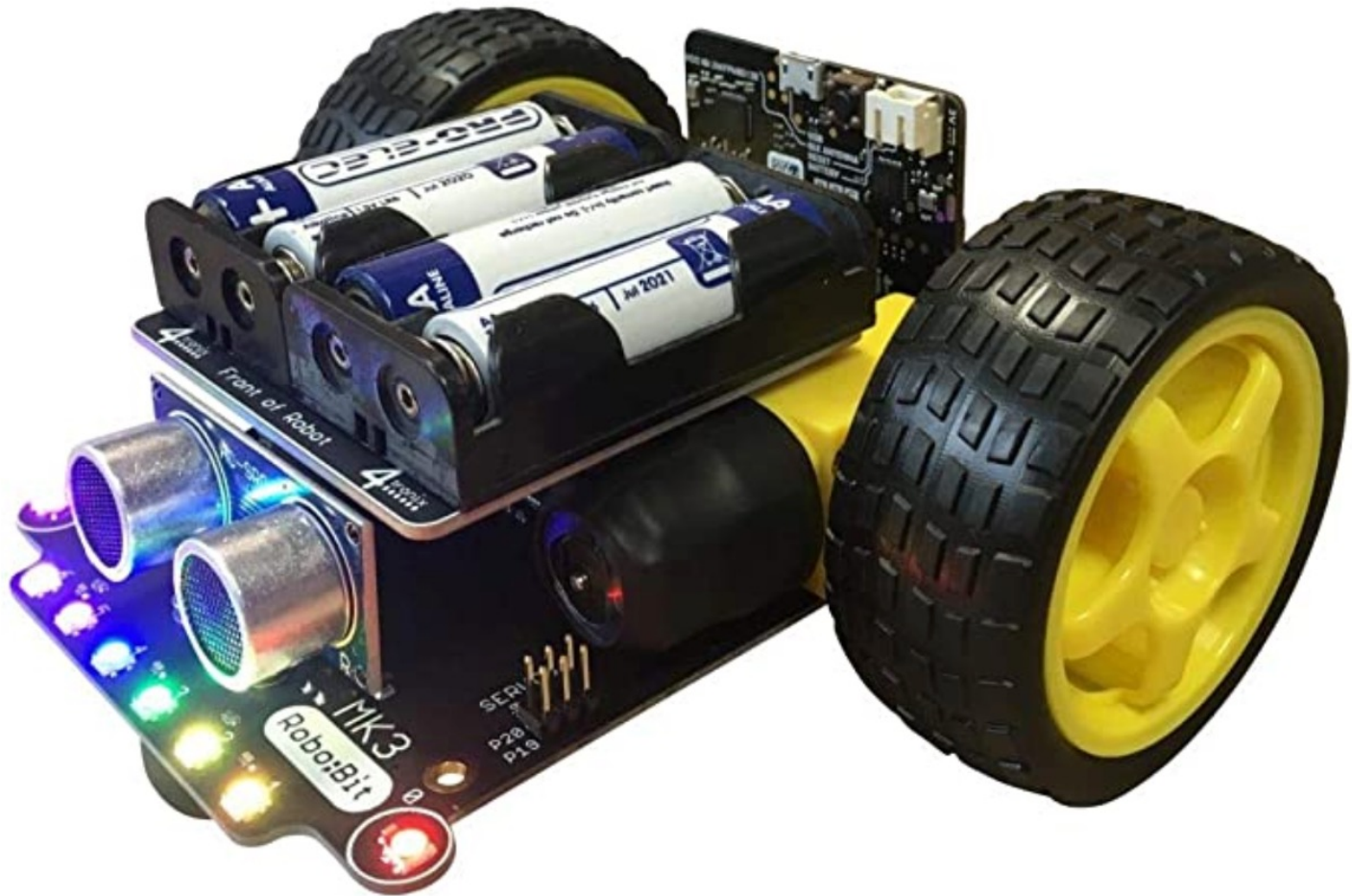
# Elementos de un Robot Móvil Autónomo



**SISTEMA DE  
ALIMENTACIÓN**



# Programación



# 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 ejercicios **marcados en verde** (hay que hacerlos) 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

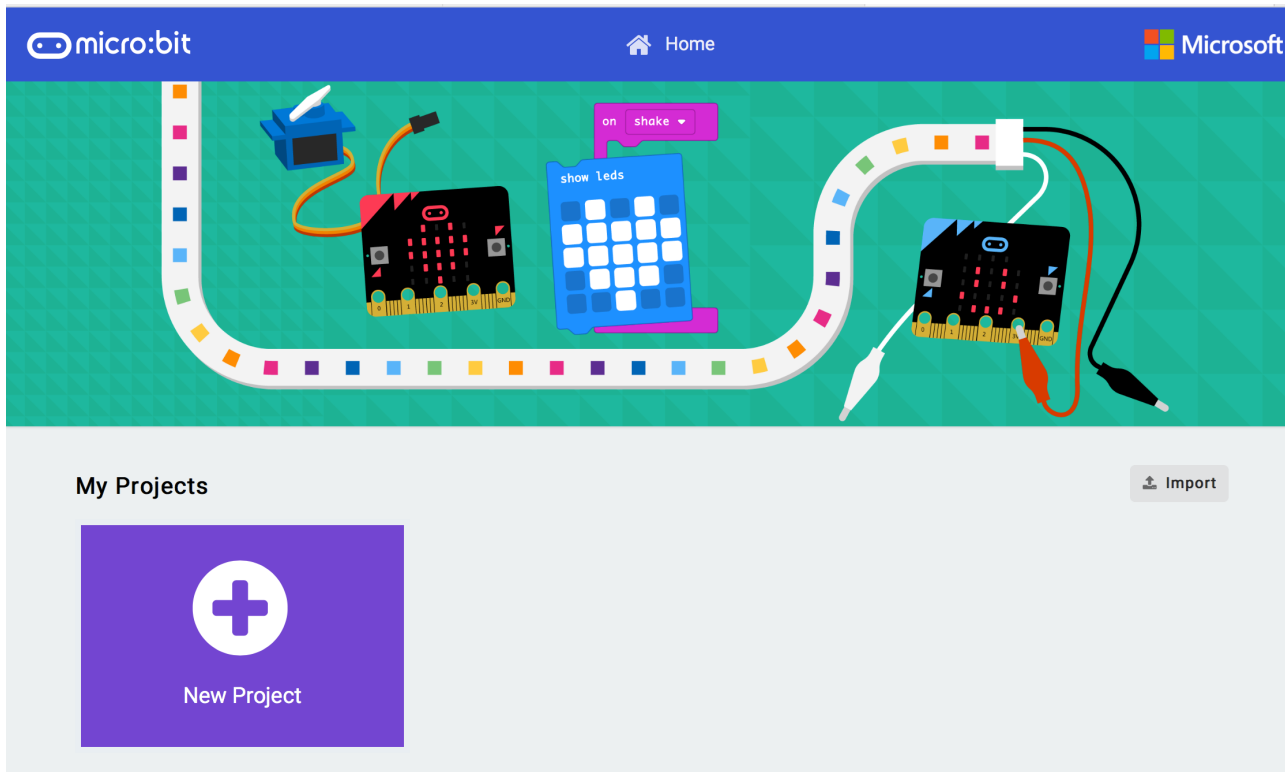


- ❑ Abre la aplicación **MicroBit** con <https://makecode.microbit.org>



Google Chrome  
Navegador web :

- ❑ La aplicación tiene la apariencia de la figura



- ❑ **Pulsa en "New Project"**

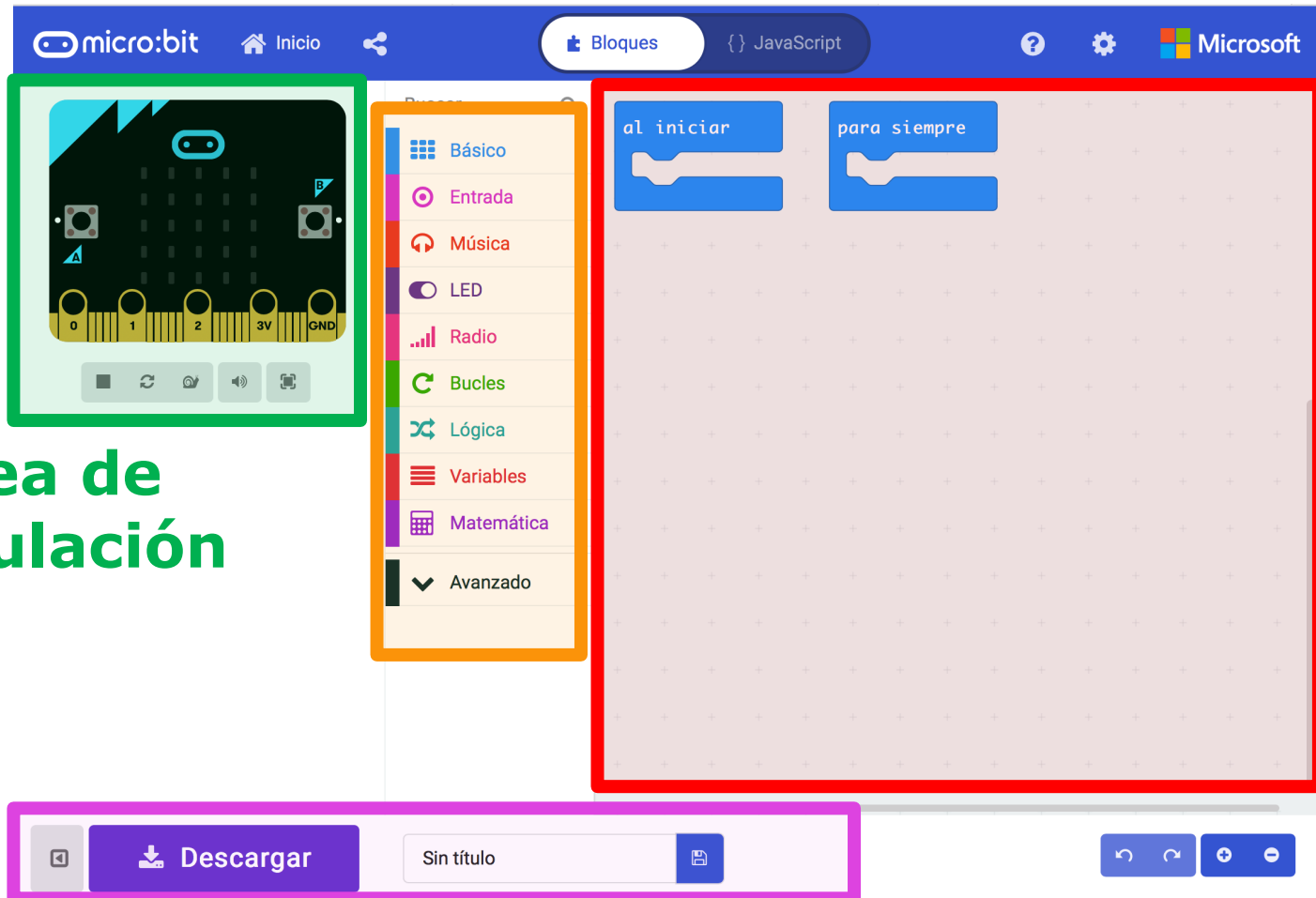
# Paso 1: Abriendo el entorno de programación



- La aplicación tiene la apariencia de la figura

**Área de Instrucciones**

**Área de programación**



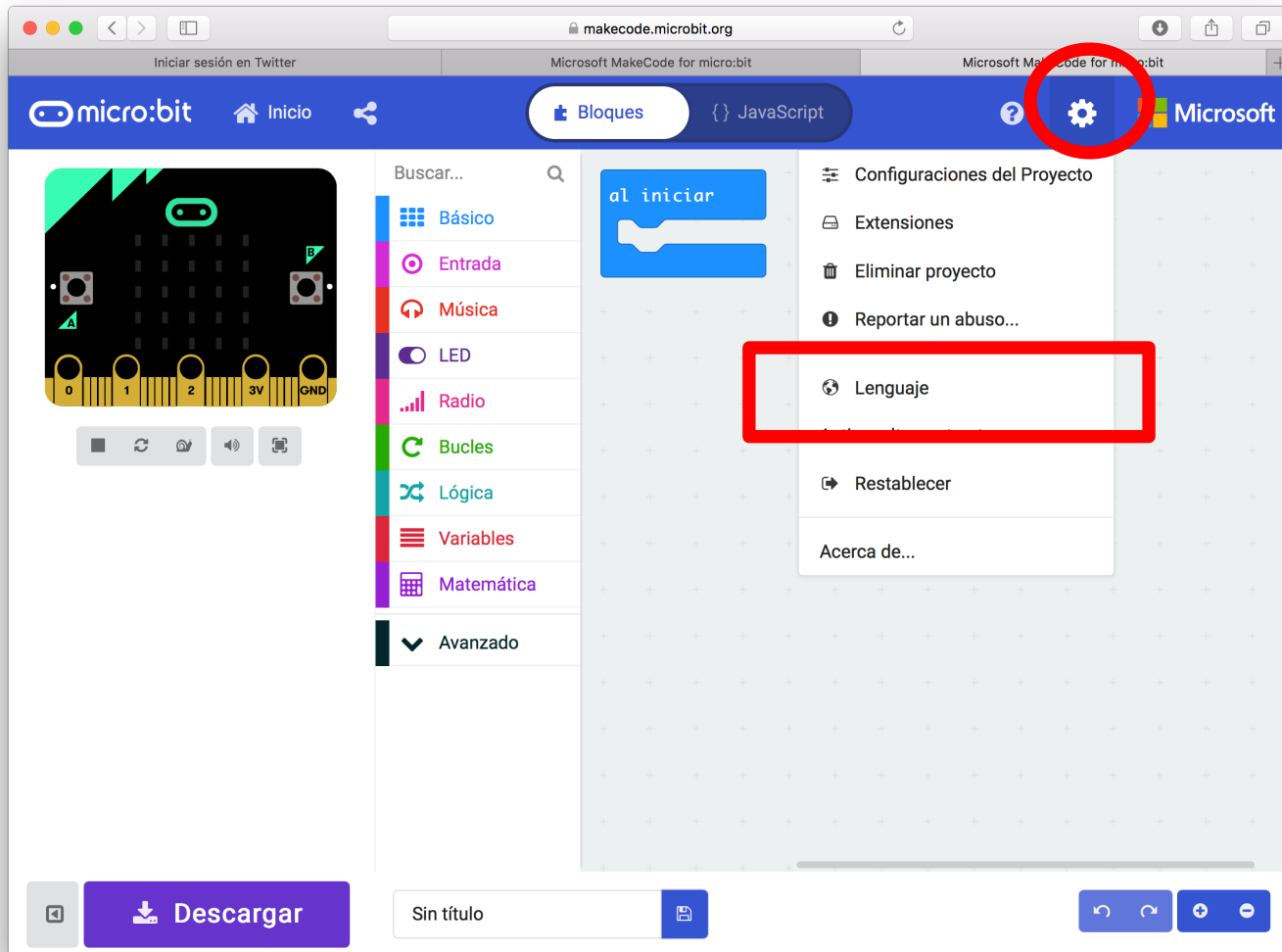
**Área de Simulación**

**Descarga**

# Paso 1a: Abriendo el entorno de programación



## ❑ Configuración del idioma

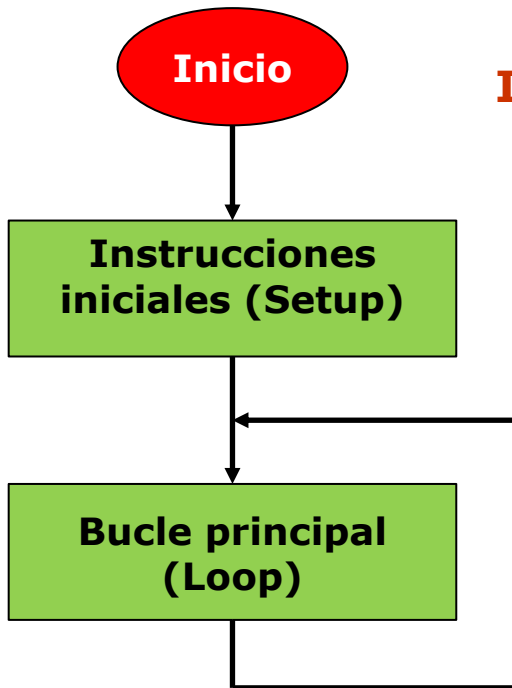


# Paso 1b: Abriendo el entorno de programación



## □ 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

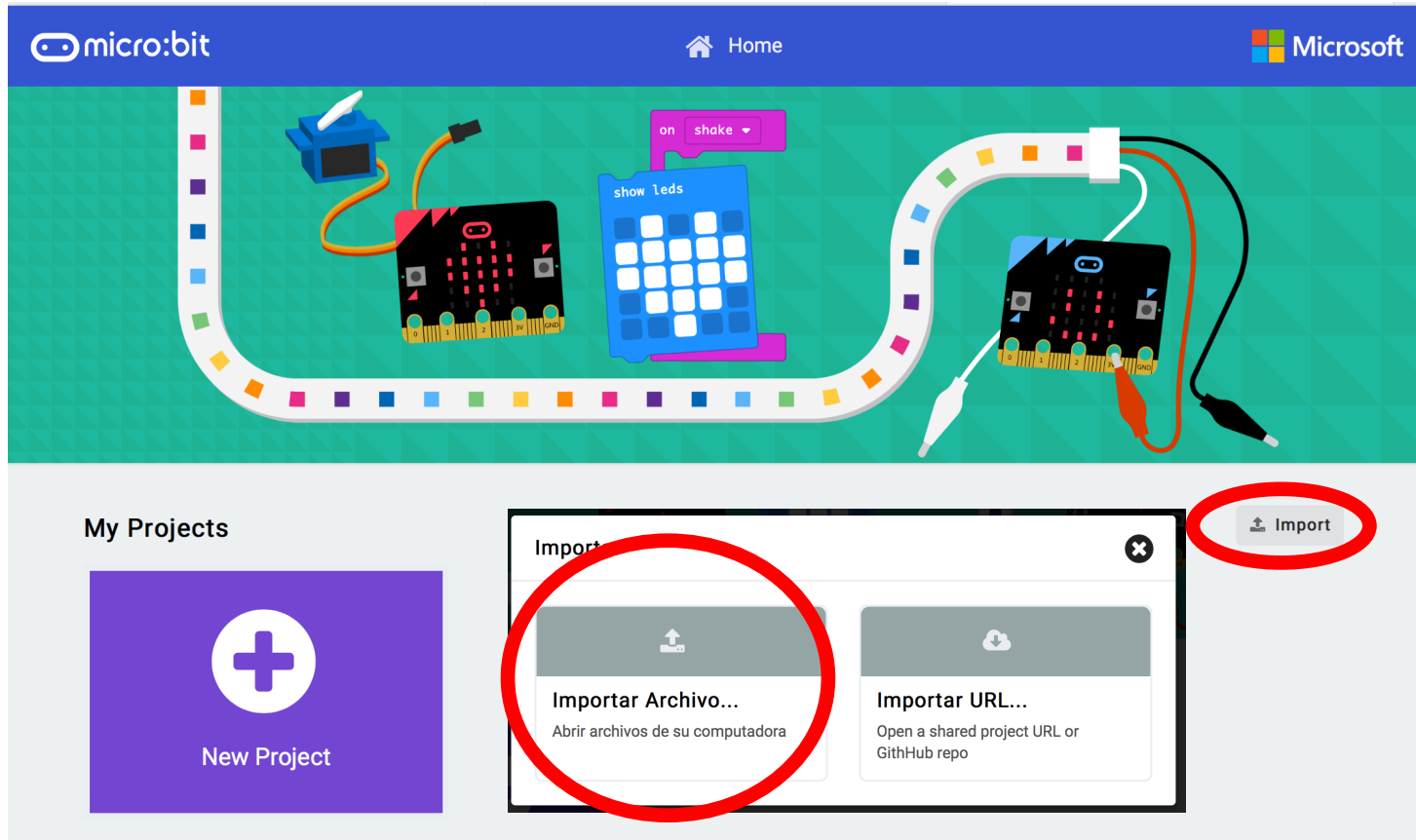
Para cargar los proyectos:

1. Se selecciona **PROYECTOS**, en la parte superior
2. Seleccionar **IMPORTAR ARCHIVO** desde el ordenador
3. Con cada ejercicio nuevo, abrir su **PASO** correspondiente:
  - 3\_EncendiendoUnLed.hex
  - 4\_SensorDeDistancia.hex
  - 5\_DetectandoObjeto.hex
  - 6\_MoviendoMotores.hex
  - 7\_DetectandoObstaculos.hex
  - 8\_SensoresSueloConFunciones.hex



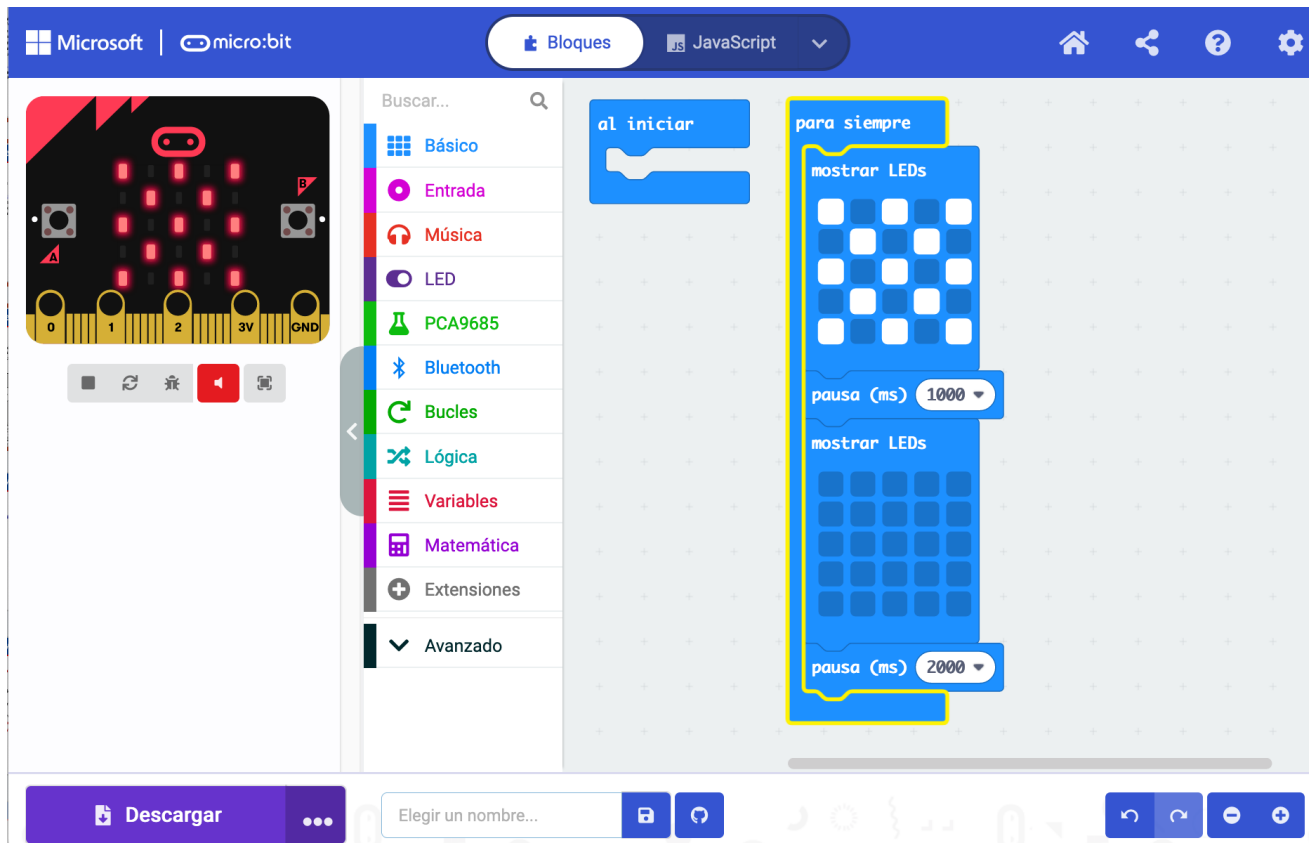
# Paso 2.a: Abriendo el primer programa

- 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



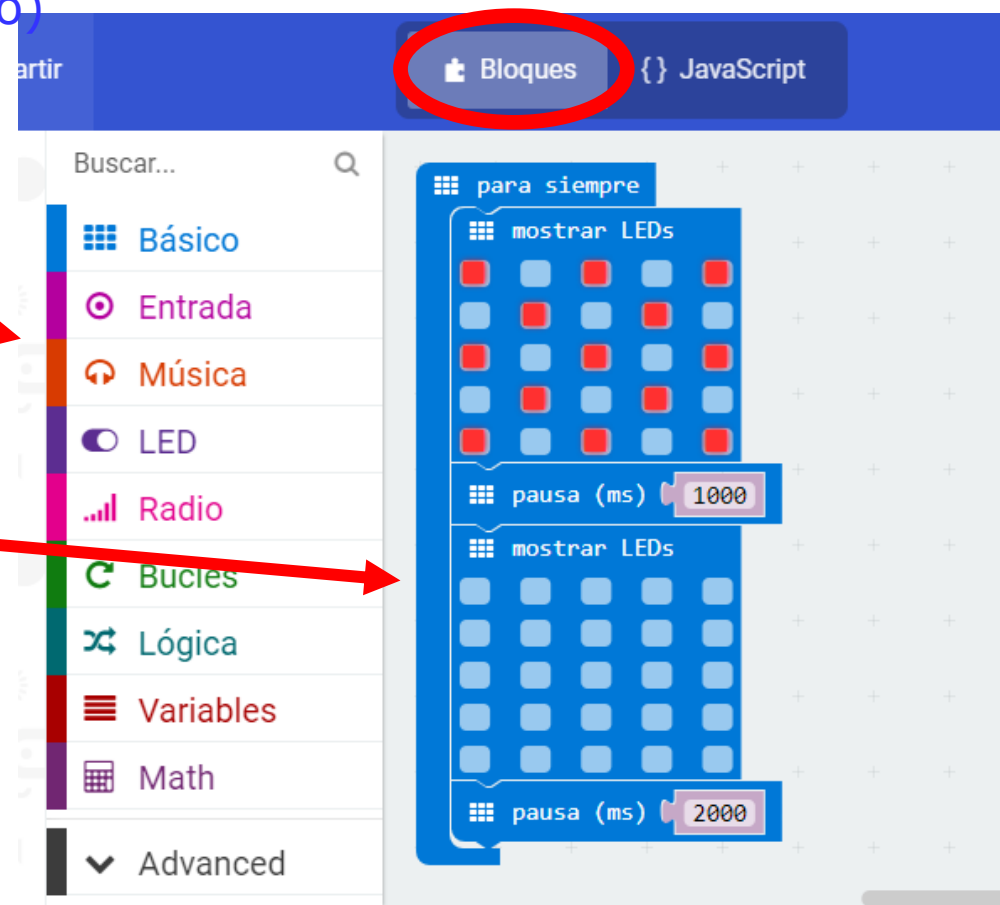
# Paso 2.c: Abriendo el primer programa

## □ 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

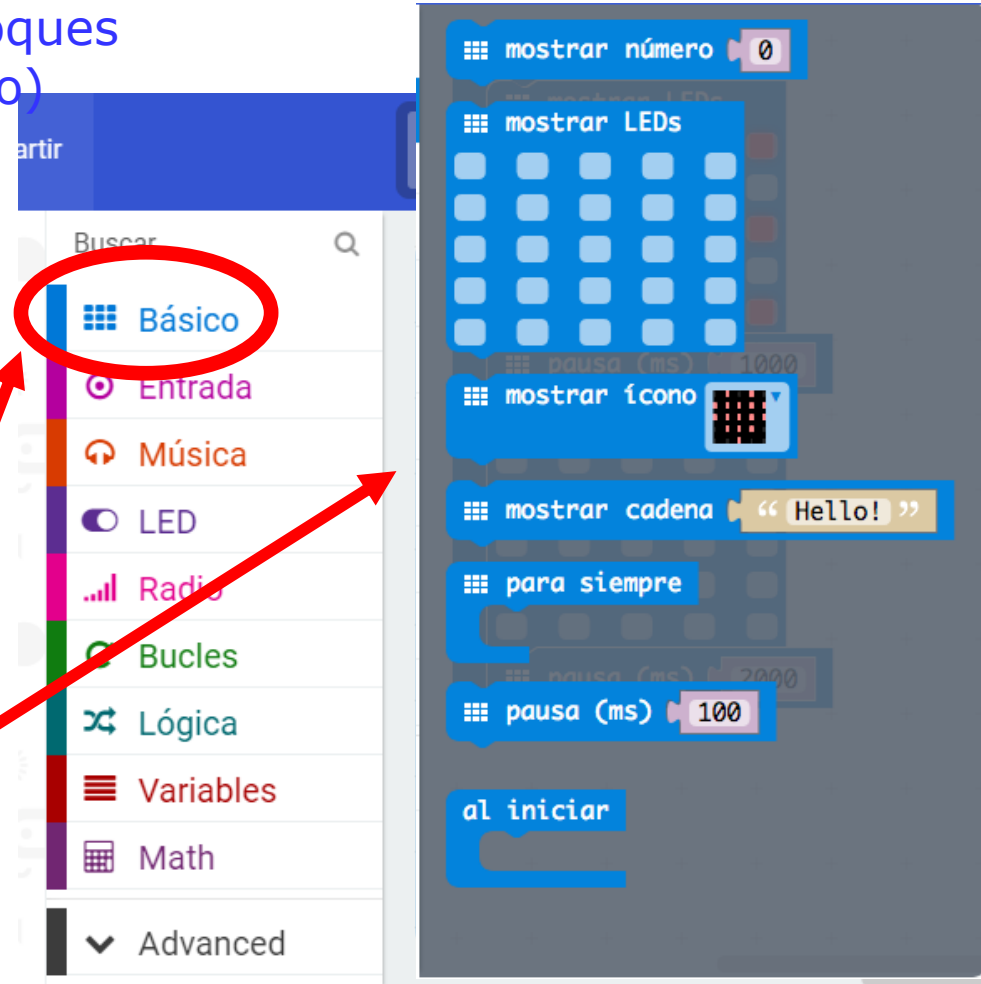
## □ Área de Programación

- 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

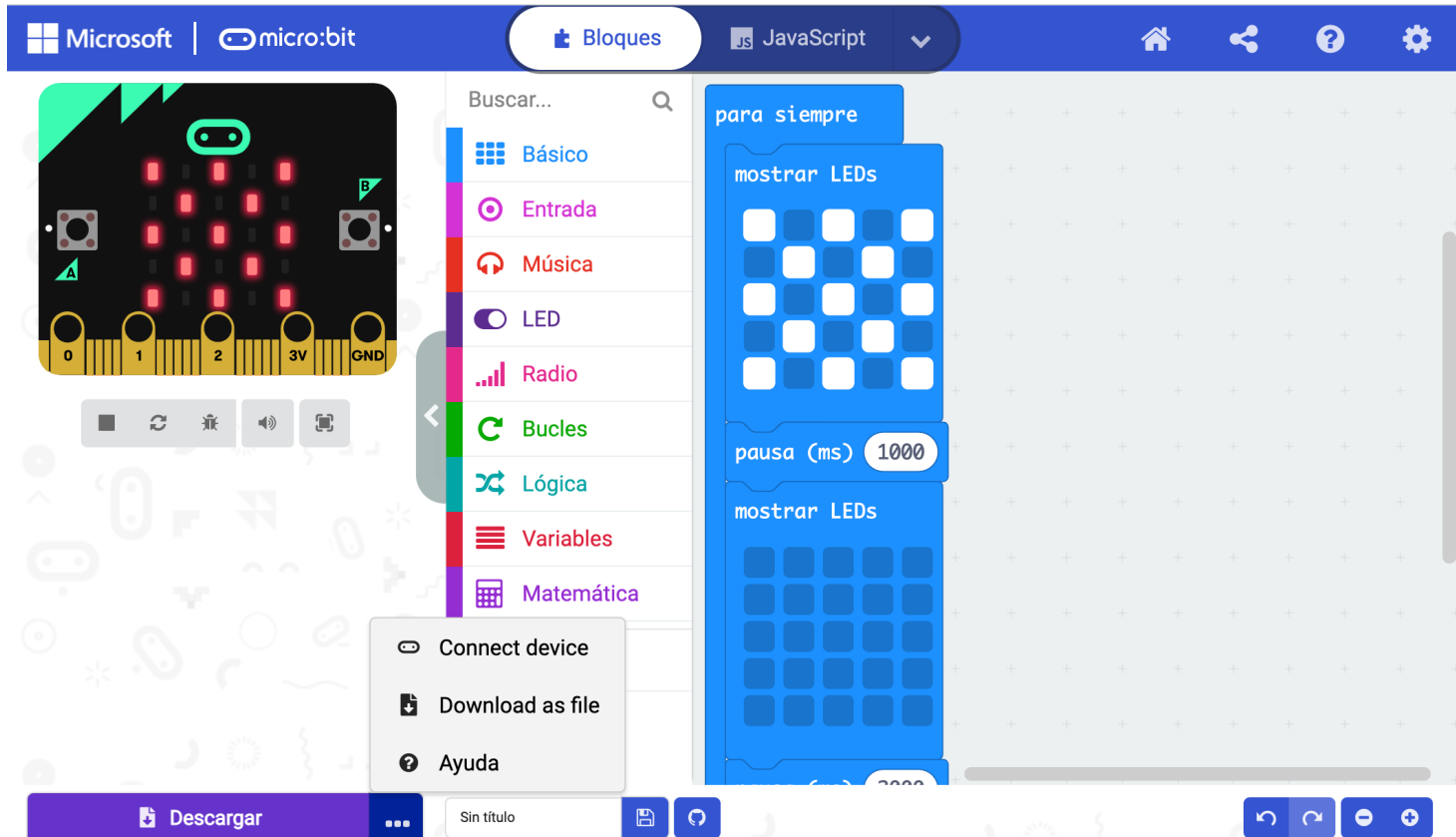
- Los colores del menú de bloques es el mismo que el de los bloques que lo contienen



# Paso 2.e: Abriendo el primer programa

## ❑ Descarga el programa en la Tarjeta

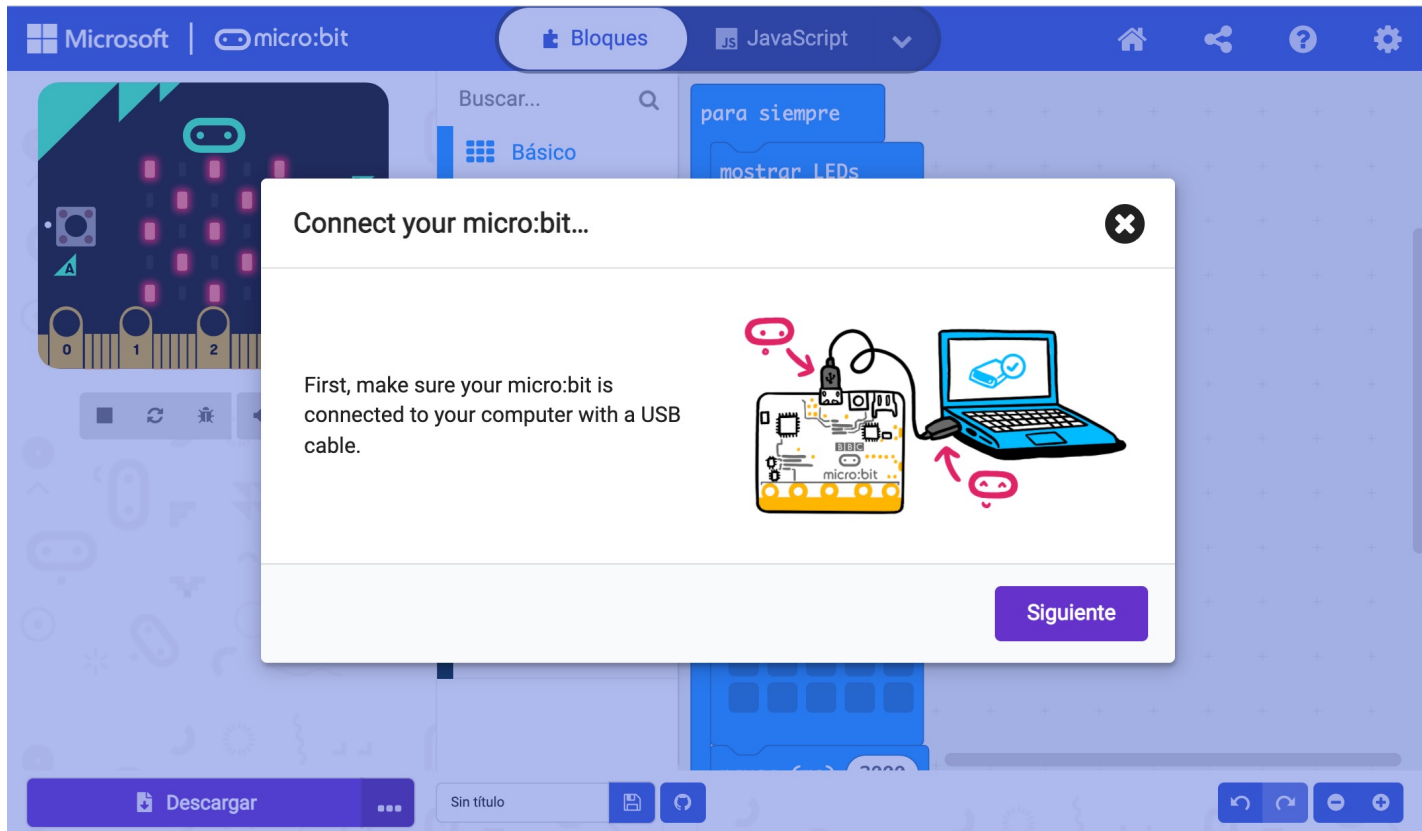
- Una vez hecho el programa hay que descargarlo en la tarjeta para probarlo
  - ❑ Conecta la tarjeta MicroBit al ordenador con el cable USB
  - ❑ Aparecerá un nuevo disco USB
  - ❑ Conectar la tarjeta con el programa



# Paso 2.e: Abriendo el primer programa

## ❑ Descarga el programa en la Tarjeta

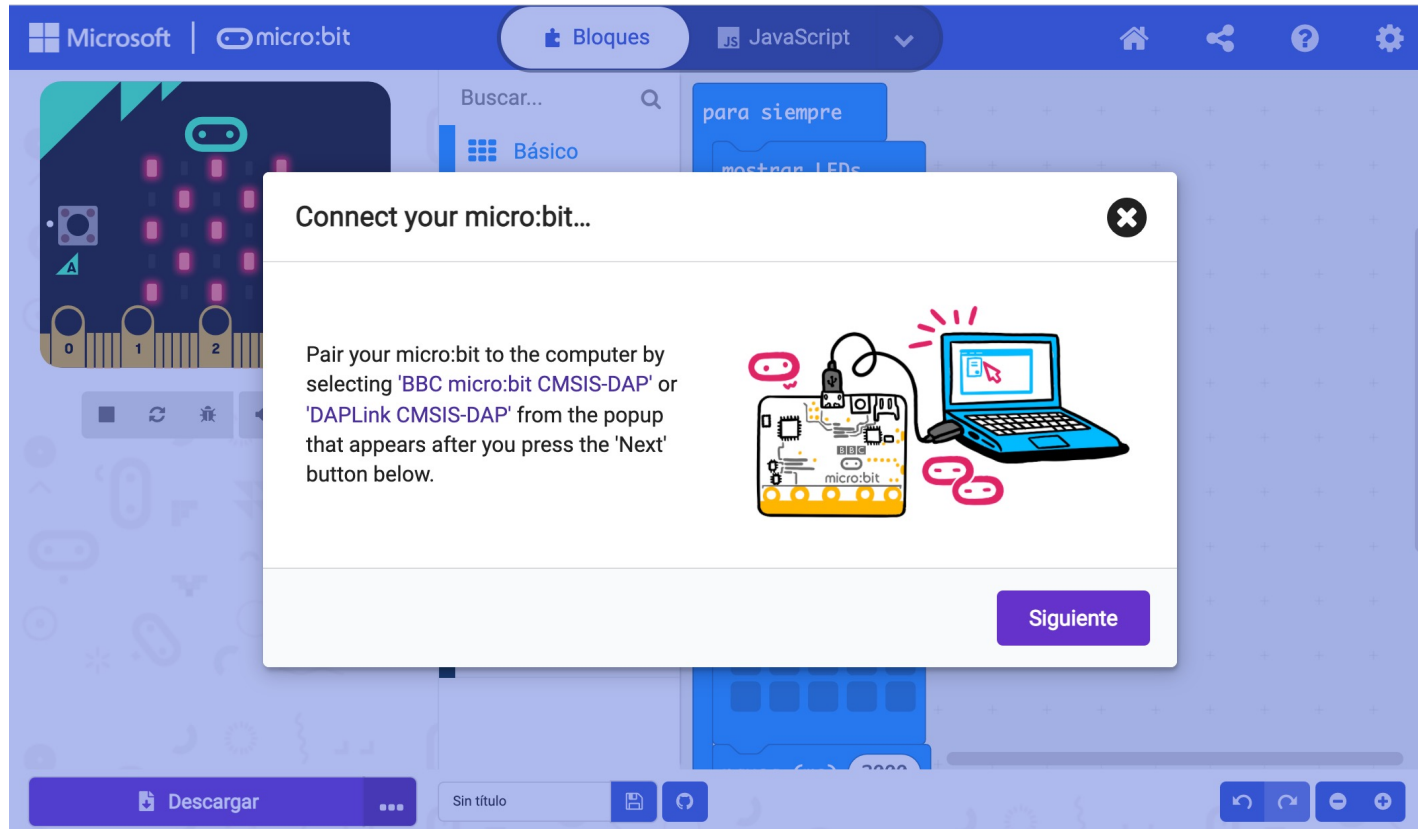
- Una vez hecho el programa hay que descargarlo en la tarjeta para probarlo
  - ❑ Conecta la tarjeta MicroBit al ordenador con el cable USB
  - ❑ Aparecerá un nuevo disco USB
  - ❑ Conectar la tarjeta con el programa



# Paso 2.e: Abriendo el primer programa

## ❑ Descarga el programa en la Tarjeta

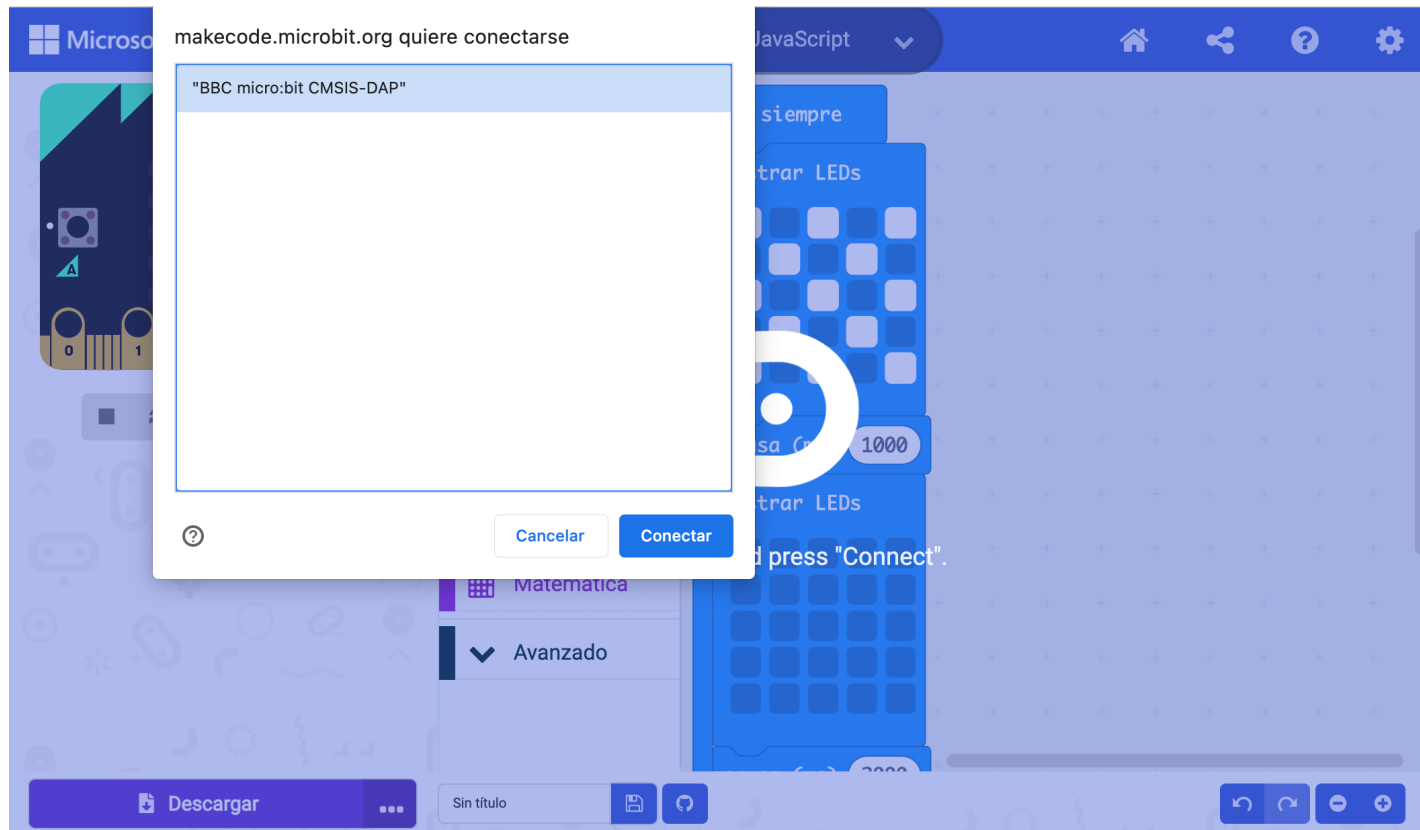
- Una vez hecho el programa hay que descargarlo en la tarjeta para probarlo
  - ❑ Conecta la tarjeta MicroBit al ordenador con el cable USB
  - ❑ Aparecerá un nuevo disco USB
  - ❑ Conectar la tarjeta con el programa



# 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

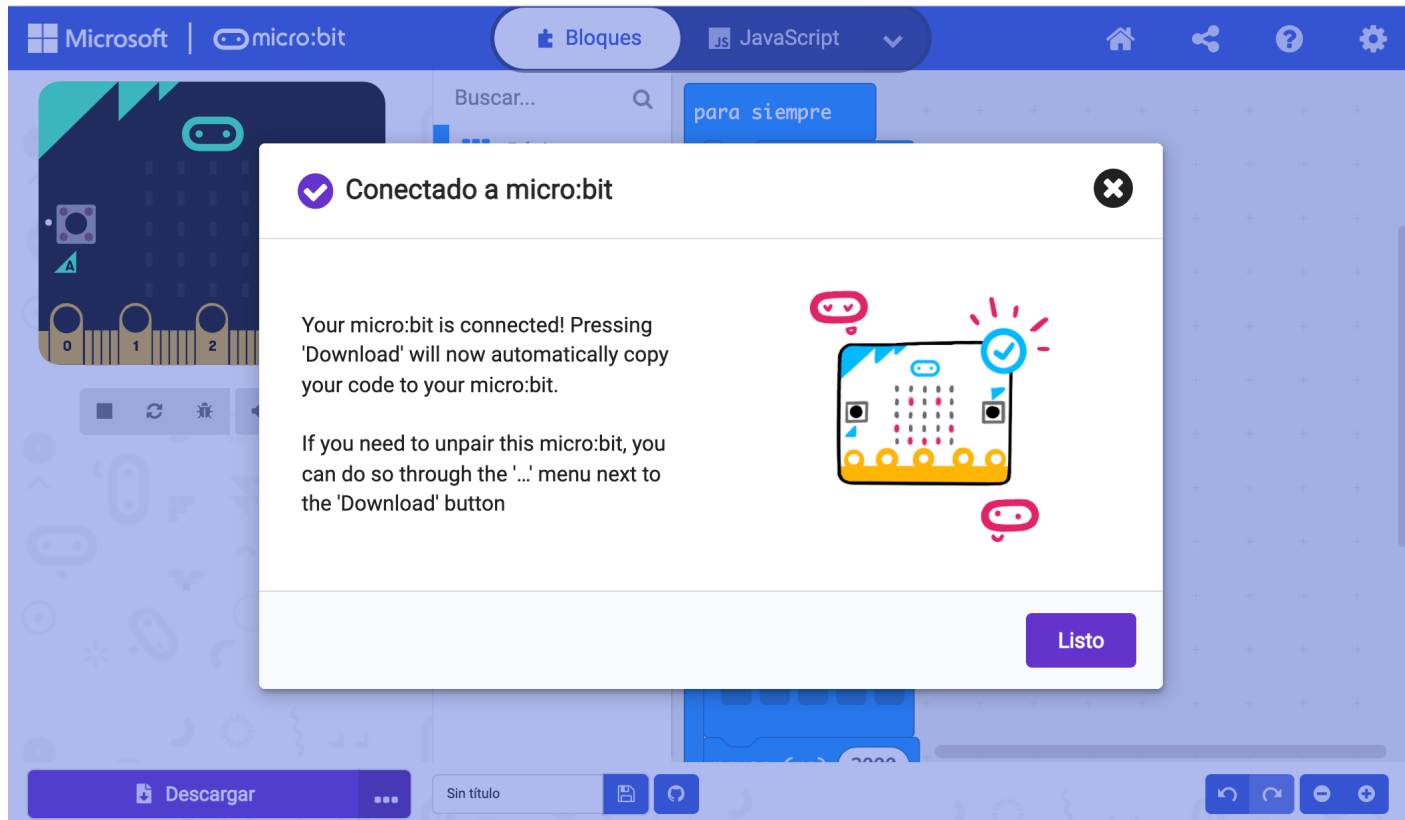




# Paso 2.e: Abriendo el primer programa

## ❑ Descarga el programa en la Tarjeta

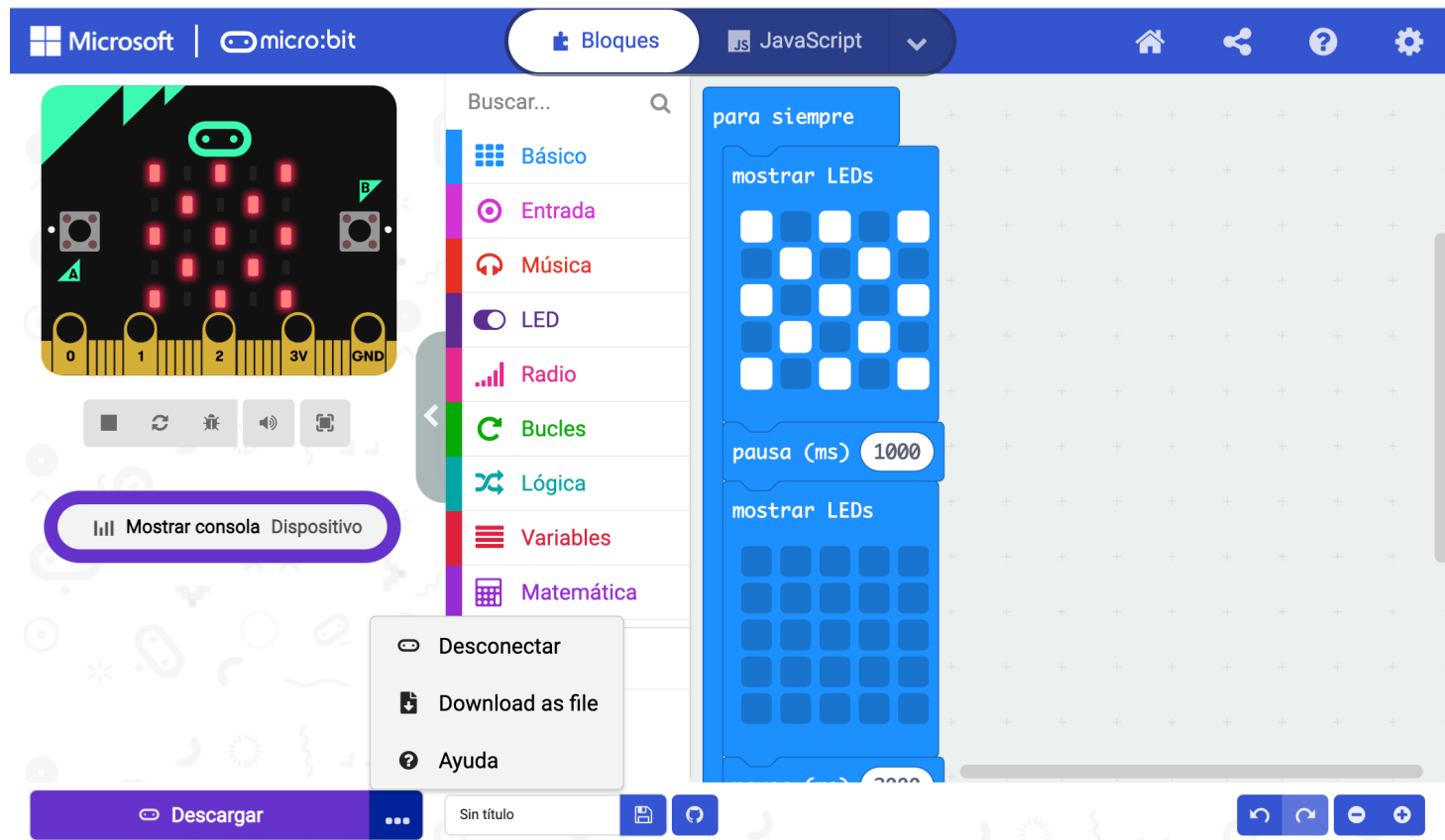
- Una vez hecho el programa hay que descargarlo en la tarjeta para probarlo
  - ❑ Conecta la tarjeta MicroBit al ordenador con el cable USB
  - ❑ Aparecerá un nuevo disco USB
  - ❑ Conectar la tarjeta con el programa



# Paso 2.e: Abriendo el primer programa

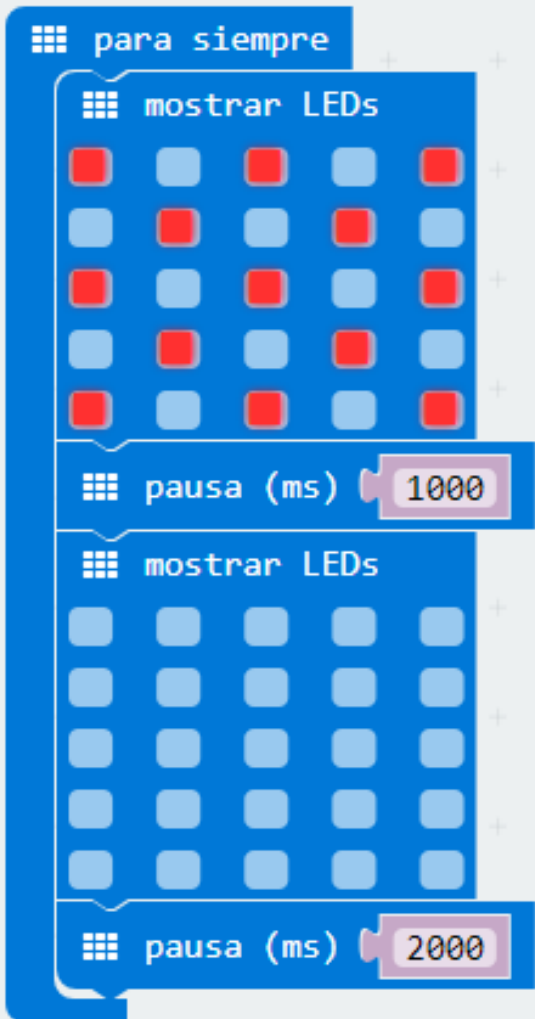
## ❑ Descarga el programa en la Tarjeta

- 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 **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**

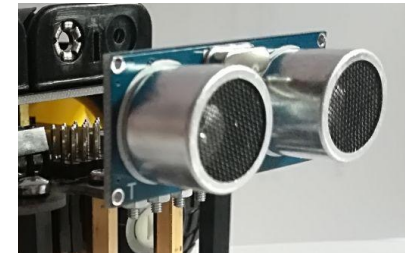
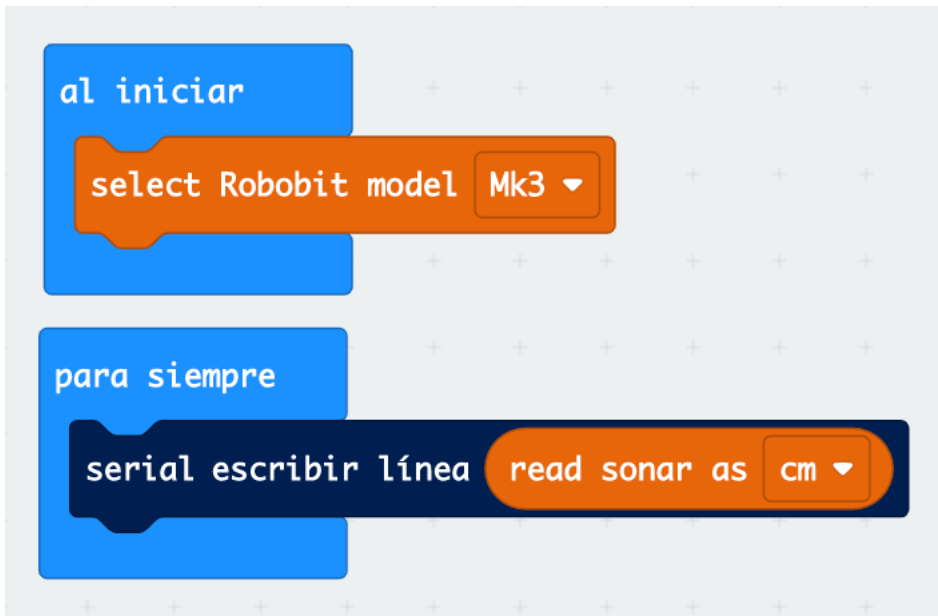
Haz que se muestre tu nombre al inicio.

# Paso 4.a: Probando el sensor de distancia

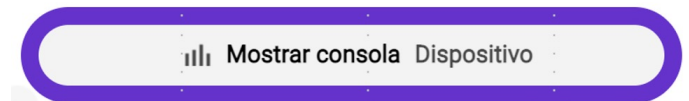
- Abre y descarga el **4\_SensorDeDistancia.hex** en el robot

## 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.



Envía al ordenador el valor de distancia medido.



## Ejercicio

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

# Paso 4.b: Probando el sensor de distancia

- Abre y descarga el **4\_SensorDeDistancia.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.

Microsoft | micro:bit

Bloques JavaScript

Volver atrás

Dispositivo

238.00

0.00

42

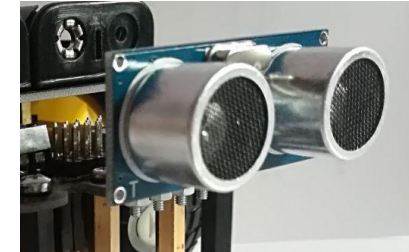
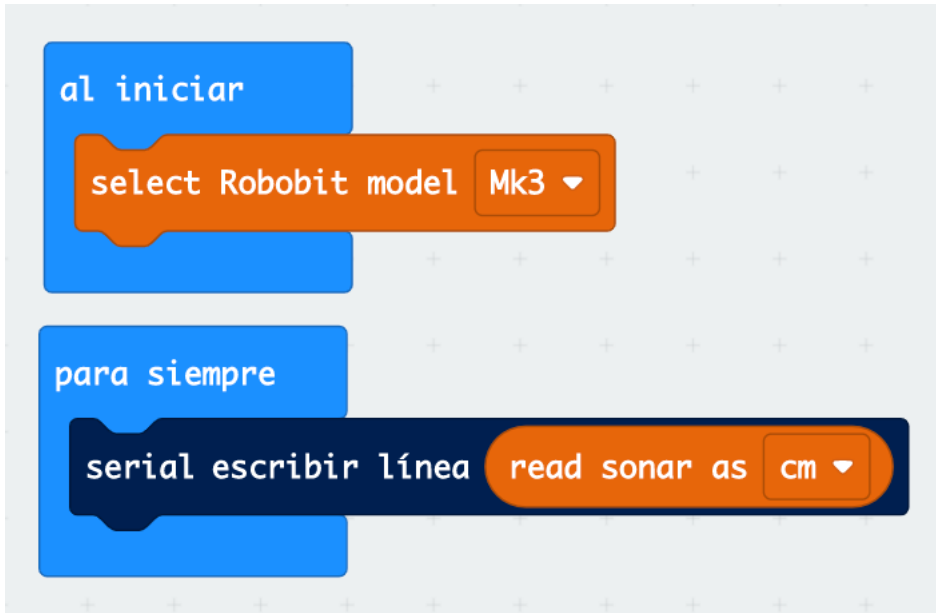
Mostrar consola Simulador

Mostrar consola Dispositivo

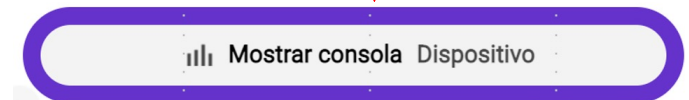
```
9 42
41
2 42
43
26 42
```

## Paso 4.c: Probando el sensor de distancia

- Abre y descarga el **4\_SensorDeDistancia.hex** en el robot



Envía al ordenador el valor de distancia medido.



### **Ejercicio**

Modifica el programa para que envíe un dato cada 1 segundo.

### **Ejercicio**

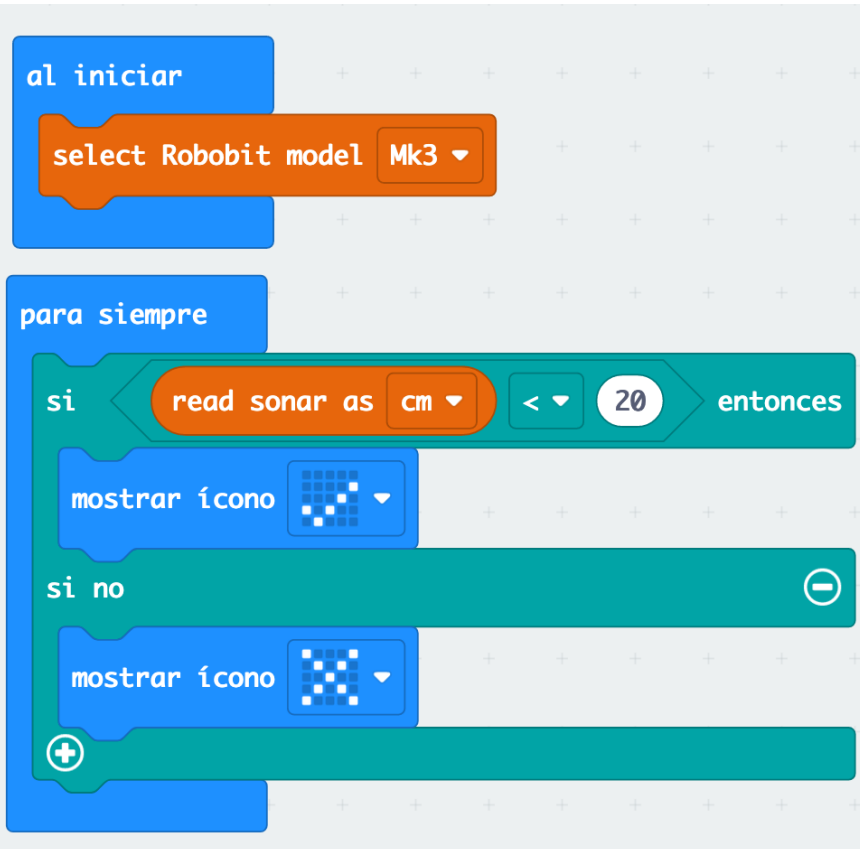
Cambia las unidades de medida en us para ver el tiempo que tarda el ultrasonido en rebotar

## Paso 5: Detectando un objeto

- ❑ Carga el programa **5\_DetectandoObjeto.hex**

### Observa lo que realiza el programa

Si detecta un objeto a menos de 20 cm pone un "√" en el display y si no una "X".



### ❑ Ejercicio

- Modifica el programa para que responda a una distancia **menor** de 10cm.

### ❑ Ejercicio

- Modifica el programa para que identifique tres distancias diferentes.

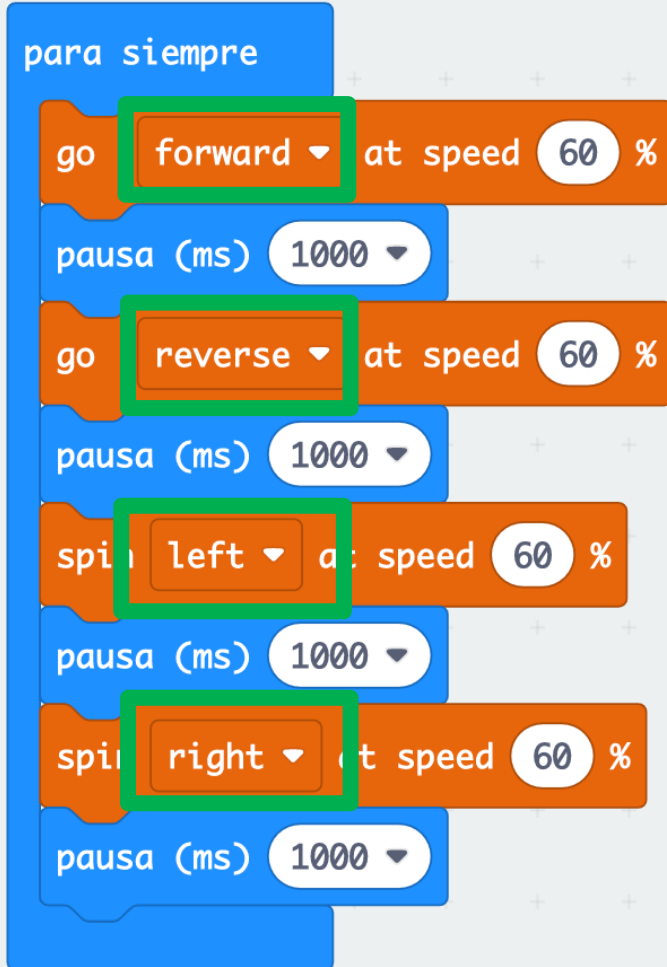
### ❑ Ejercicio

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

## Paso 6.a: Moviendo los motores

- ❑ Carga el programa **6\_MoviendoMotores.hex**

**Observa lo que realiza el programa**



Mueve el robot hacia **delante** a una velocidad del **60%** durante un **1 segundo**

Luego mueve el robot hacia **detrás** a una velocidad del **60%** durante un **1 segundo**

Gira el robot hacia la **izquierda** a una velocidad de **60%** durante un **1 segundo**

Gira el robot hacia la **derecha** a una velocidad de **60%** durante un **1 segundo**



## Paso 6.b: Moviendo los motores

- ❑ Carga el programa **6\_MoviendoMotores.hex**

para siempre

go forward ▾ at speed 60 %

pausa (ms) 1000 ▾

go reverse ▾ at speed 60 %

pausa (ms) 1000 ▾

spin left ▾ at speed 60 %

pausa (ms) 1000 ▾

spin right ▾ at speed 60 %

pausa (ms) 1000 ▾

- ❑ **Ejercicio:**

- Modifica el programa para que se mueva 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: Detectando obstáculos

- ❑ Carga el programa **7\_DetectandoObstaculos.hex**

## Observa lo que realiza el programa

Si la **distancia** (que lee el ultrasonidos) es **mayor que 20** cm, el robot **avanza**.

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

```
para siempre
  si read sonar as cm > 20 entonces
    go forward at speed 60 %
  si no
    stop with no brake
```

## ❑ Ejercicio

Modifica el programa para que el robot avance en línea recta y cuando detecte un **obstáculo a menos de 10 cm**, vaya para atrás un poco y gire otro poco.

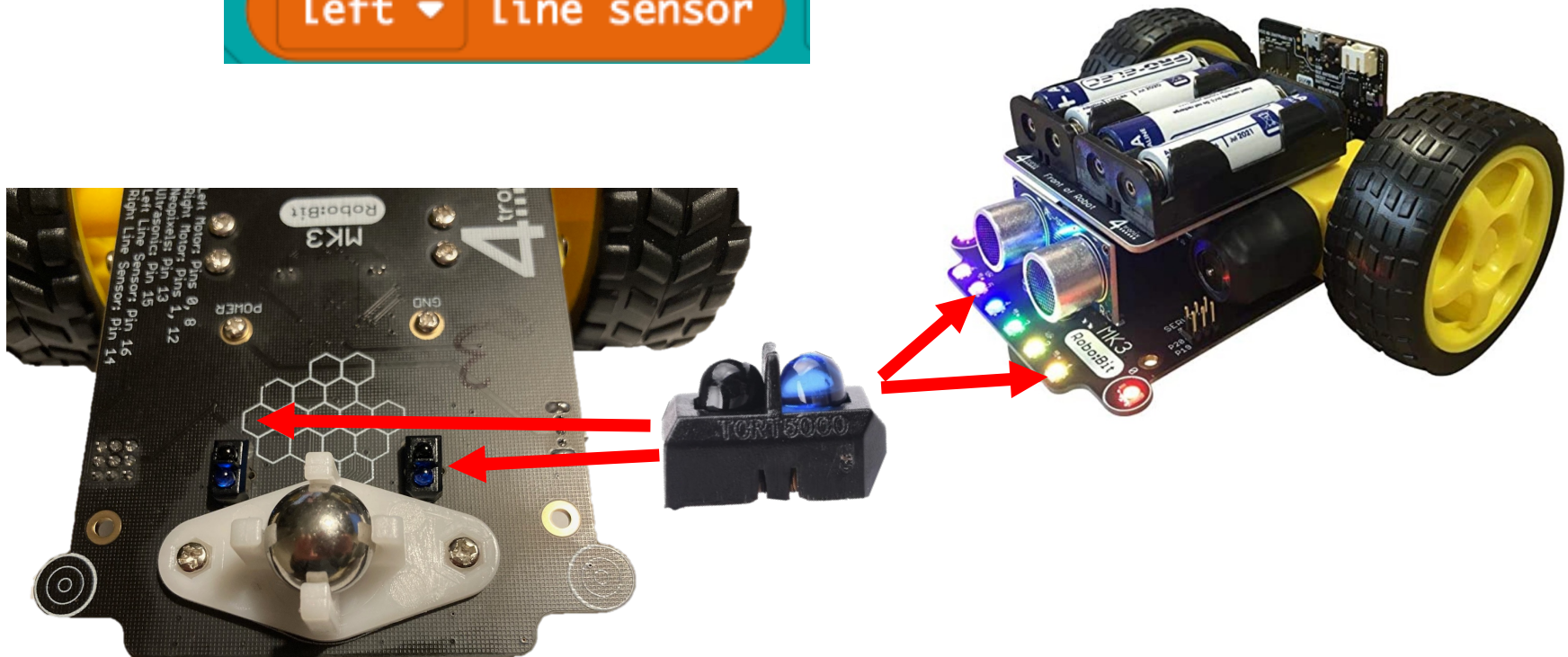
Esto producirá un comportamiento de navegación evitando obstáculos.

# Paso 8.a: Leyendo los sensores de suelo

- ❑ Carga el programa **8\_SensoresSuelo.hex**

## Para utilizar los sensores de suelo...

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



# Paso 8.b: Leyendo los sensores de suelo

- ❑ Carga el programa **8\_SensoresSuelo.hex**

## Observa que realiza el programa:

Representa en pantalla una línea del lado del sensor que detecta blanco. Si no lo detecta bien, puede que haya que mover el potenciómetro

```
para siempre
si read left line sensor == 1 y read right line sensor == 0 entonces
  mostrar LEDs
si no, si read left line sensor == 0 y read right line sensor == 1 entonces
  mostrar LEDs
si no, si read left line sensor == 0 y read right line sensor == 0 entonces
  mostrar LEDs
si no
  borrar la pantalla
```

**Si** detecta **blanco solo con el sensor derecho**, pinta línea a la derecha

**Si no, si** detecta **blanco solo con el sensor izquierdo**, pinta línea a la izquierda.

**Si no, si** detectan **blanco los dos sensores**, pinta dos líneas

**Si no** (si no detecta blanco), no pinta nada

## Paso 9: 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 con cualquiera de los dos sensores.

- **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

**Nota:** la microbit tarda casi un segundo en pintar en el display. Mientras pinta no comprueba los sensores. **Es conveniente no dedicar tiempo a pintar.**

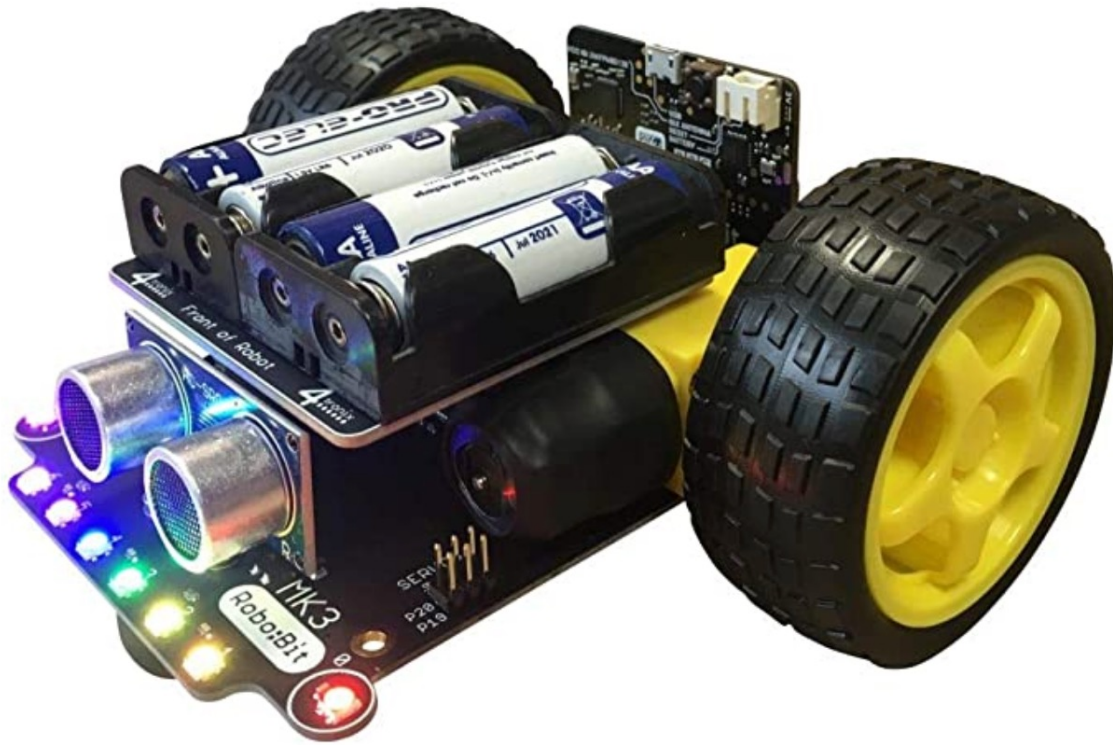
## Actividad 8: Luchador de sumo

- ❑ Entra en la clase de Microbit **8-Luchador de sumo**
- ❑ Ya estás listo para programar el luchador de sumo. Busca al oponente, détéctale, empújale, pero no te salgas tú solo.



# Mini-Taller de Robótica Móvil

## Programa un robot de sumo



Universidad  
de Alcalá



Departamento de  
**e**lectrónica

# STEM / STEAM

- **Términos STEM / STEAM**
  - **STEM (Science, Technology, Engineering, Maths).**
  - **STEAM (Science, Technology, Engineering, Art, Maths).**
- **Gran esfuerzo por parte de los gobiernos nacionales y autonómicos para impulsar el STEM/STEAM**





# Tecnología / Ingeniería

## □ Tecnología

### ■ Definiciones

- Conjunto de **teorías y de técnicas** que permiten el **aprovechamiento práctico del conocimiento científico**. - <https://dle.rae.es/tecnología>
- La **aplicación del conocimiento científico** a los objetivos prácticos de la vida humana o al cambio y manipulación del entorno por los humanos.  
<https://www.britannica.com/technology/technology>
- La tecnología es el **conjunto de conocimientos y técnicas que se aplican de manera ordenada** para alcanzar un determinado objetivo o resolver un problema.  
<https://economipedia.com/definiciones/tecnologia.html>
- La tecnología es la **suma de técnicas, habilidades, métodos y procesos** utilizados en la **producción de bienes o servicios** o en el logro de objetivos, como la investigación científica.  
<https://es.wikipedia.org/wiki/Tecnología>

# Tecnología / Ingeniería

## □ Ciencia y Tecnología

### ■ Ciencia

- Es un conjunto ordenado de conocimientos que se obtienen a través de la observación y la experimentación.

### ■ Tecnología

- Toma los conocimientos y técnicas provenientes de las diferentes ciencias y los utiliza para desarrollar bienes o servicios tangibles e intangibles que contribuyan al desarrollo humano.

<https://concepto.de/tecnologia/>

# Tecnología / Ingeniería

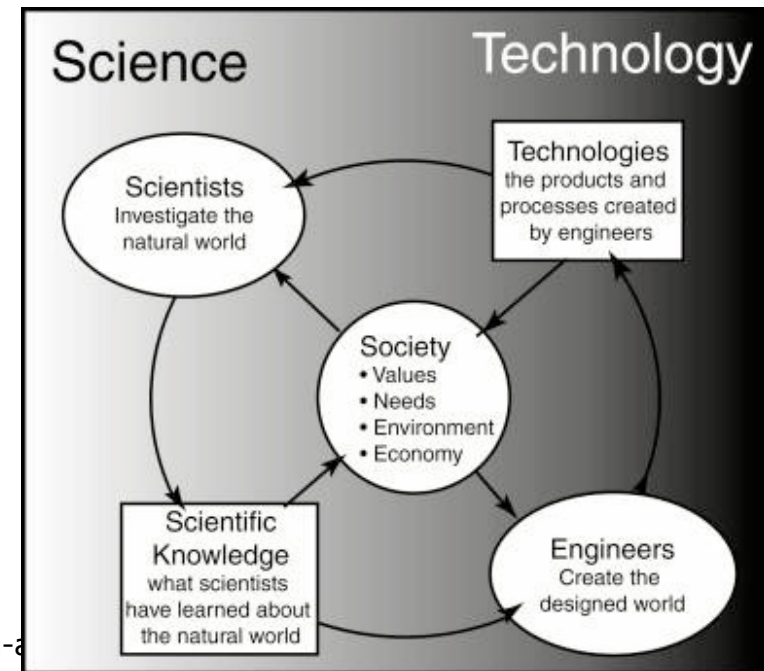
## □ Tecnología e Ingeniería

### ■ Ingeniería

- Es un **proceso de diseño** que combina el **conocimiento** de las propiedades de los materiales, los **modelos** que predicen cómo se comportan y el **pensamiento innovador** para crear soluciones a menudo novedosas para las necesidades humanas.
- La base de la ingeniería es el **diseño** de soluciones nuevas y el **proyecto de ingeniería**

### ■ Tecnología

- **Proceso de hacer cosas**, usando herramientas, materiales y habilidades de proceso para **convertir en realidad** física los planes de los ingenieros (así como los de otros profesionales del diseño, como artistas y arquitectos).



# Pensamiento computacional

## □ ¿Qué es el pensamiento computacional?

- Es el proceso de **reconocimiento de aspectos de la informática en el mundo que nos rodea, y aplicar herramientas y técnicas de la informática** para comprender y razonar sobre los sistemas y procesos tanto naturales como artificiales.
- Es pensar como lo haría un informático cuando nos enfrentamos a un problema.
- Características
  - **Descomposición:** consiste en el procedimiento por el cual un problema de mayor complejidad se desarticula en pequeñas series más manejables.
  - **Reconocimiento de patrones:** luego de la desarticulación del problema complejo, las pequeñas series son enfrentadas de forma individual de manera que puedan ser resueltas de forma similar a problemas frecuentados anteriormente.
  - **Abstracción:** Consiste en la omisión de información irrelevante al problema propuesto.
  - **Algoritmos:** se presentan pasos para la resolución de cada problema.

# Programación con Code.org

## □ ¿Qué es?

- Es una organización sin fines de lucro dedicada a ampliar el acceso a las ciencias de la computación en las escuelas y aumentar la participación de las mujeres jóvenes y estudiantes de otros grupos no suficientemente representados.
  - <https://code.org/international/about>
- Todos los recursos y tutoriales del plan de estudios son siempre de uso gratuito y están bajo licencias abiertas de Creative Commons, para que otros actores puedan utilizarlos con fines no comerciales.

# Programación con Code.org

## □ Tipos de cursos

### ■ Cursos exprés

- Combinan contenidos de otros cursos más extensos.

#### Curso Express para pre-lectores (2021)

De 4 a 8 años

Una introducción a las ciencias de la computación para prelectores, que combina lo mejor de nuestros cursos para educación preescolar y primer grado.



#### Curso Exprés (2021)

Edades: 9 a 18

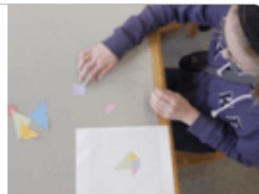
Una introducción a la ciencia de la computación: combina lo mejor de nuestro plan de estudios de primaria para estudiantes mayores.



#### Lecciones sin conexión

Desde 4 años

Si no tienes ordenadores, prueba estas lecciones sin conexión en tu aula.



# Programación con Code.org

## □ Tipos de cursos

### ■ Hora del Código

- Actividades pensadas para una hora.



#### Fiesta de Baile

Programa una fiesta de baile y compártela con tu familia y amigos. ¡Con música de Beyoncé, Harry Styles, Lizzo, Lil Nas X, Selena Gomez, la película de Disney «Encanto» y más!



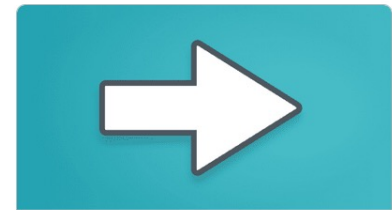
#### Minecraft

¡Usa tu creatividad y habilidades para resolver problemas para explorar y construir mundos bajo el agua con códigos!



#### Frozen

Vamos a usar código para unirnos a Anna y Elsa mientras exploran la magia y la belleza del hielo.



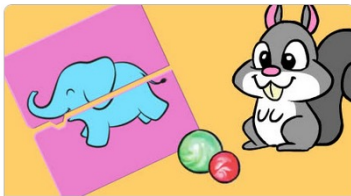
#### Ver más

Ver más tutoriales de la Hora del Código

# Programación con Code.org

## □ Tipos de cursos

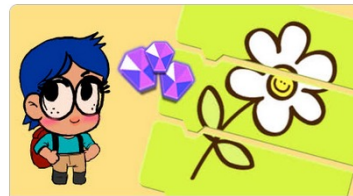
### ■ Fundamentos de informática para Escuelas Primarias



#### Curso A (2021)

De 4 a 7 años

Una introducción a la informática para prelectores.



#### Curso B (2021)

De 5 a 8 años

Una introducción a las ciencias de la computación para prelectores. Es similar al Curso A, pero con conceptos más avanzados.



#### Curso C (2021)

De 6 a 10 años

Aprende los fundamentos de la informática y crea tu propio arte, historias y juegos.



#### Curso D (2021)

Edades de 7 a 11 años

Aprende sobre sentencias condicionales y crea tu propio arte, historia o juego.



#### Curso E (2021)

De 8 a 12 años

Aprende los fundamentos de los sprites, los bucles anidados y las funciones.



#### Curso F (2021)

De 9 a 13 años

Aprende todos los conceptos de Fundamentos de Informática y crea tu propio arte, historia o juego.



# Programación con Code.org

## □ Tipos de cursos

### ■ Duración de los cursos

- Cada etapa tiene una duración aproximada de 40 minutos

### Cursos modernos

| Curso        | Edades         | Etapas y ejercicios                             |
|--------------|----------------|---|
| Curso C      | De 6 a 10 años | 9 etapas con 102 ejercicios.                    |
| Curso D      | De 7 a 11 años | 12 etapas con 138 ejercicios.                   |
| Curso E      | De 8 a 12 años | 8 etapas con 92 ejercicios y proyecto final.    |
| Curso F      | De 9 a 13 años | 10 etapas con 131 ejercicios y proyecto final.  |
| Curso rápido | De 9 a 18 años | 26 etapas largas con 298 ejercicios y proyecto. |

# Programación con Code.org

## □ Curso rápido (9 a 18 años)

### ■ Conceptos que se trabajan

- Secuencia de órdenes, Sprites (objetos de un juego), Eventos, Bucles, Condicionales, Funciones, Variables, Bucles "For"

### ■ Duración

- Programado: una media de 50 minutos por lección (se suele hacer más rápido). Unas 24 horas programadas

## Curso Exprés (2021)

Edades: 9 a 18

Una introducción a la ciencia de la computación: combina lo mejor de nuestro plan de estudios de primaria para estudiantes mayores.



Lección 1: 38 min

Lección 2: 50 min

Lección 3: 50 min

Lección 4: 40 min

Lección 5: 45 min

Lección 6: 50 min

Lección 7: 55 min

Lección 8: 60 min

Lección 9: 12 min

Lección 10: 45 min

Lección 11: 60 min

Lección 12: 55 min

Lección 13: 60 min

Lección 14: 45 min

Lección 15: 55 min

Lección 16: 60 min

Lección 17: 60 min

Lección 18: 60 min

Lección 19: 50 min

Lección 20: 60 min

Lección 21: 60 min

Lección 22: 55 min

Lección 23: 45 min

Lección 24: 40 min

Lección 25: 60 min

Lección 26: 60 min

Lección 27: 60 min

Lección 28: 45 min