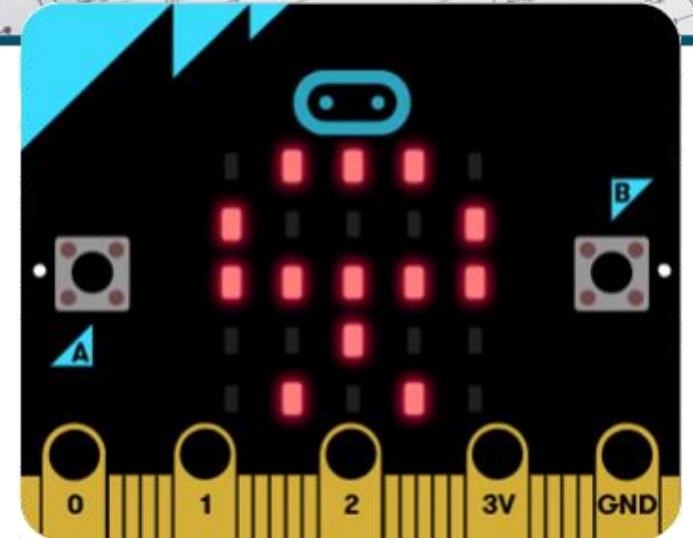
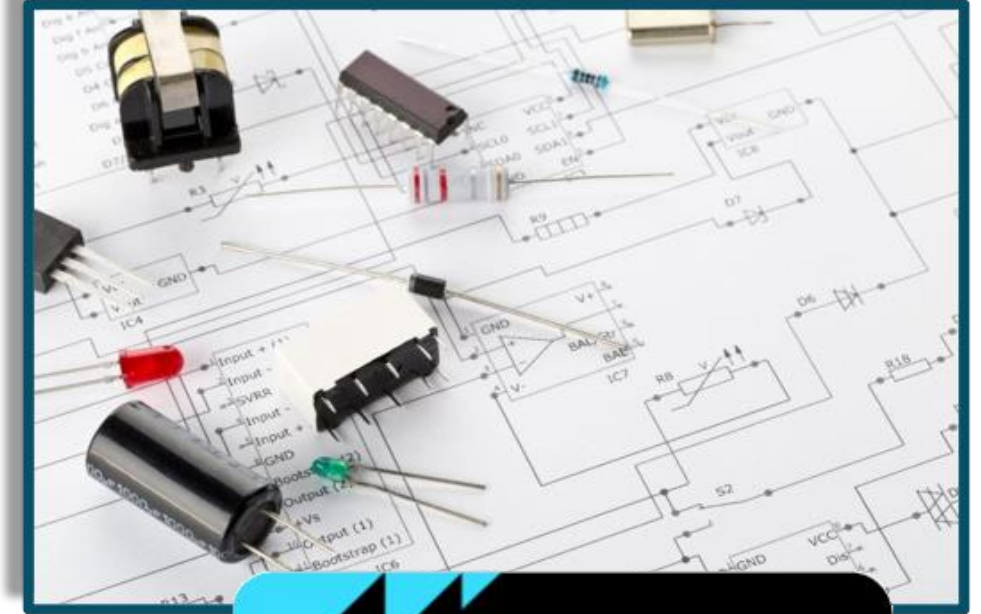


Programa Código Escuela 4.0

Actividades interdisciplinarias con robótica y pensamiento computacional



[Actividades interdisciplinarias con robótica y pensamiento computacional](#) © 2026 by [Jaime Temes Mendoza](#) is licensed under [CC BY-NC-SA 4.0](#)

Índice



- Introducción
- MRU y MRUA con Micro:bit y Maqueen.
- Plano inclinado.
- El juego de la célula.
- IA identificación de plantas.
- Crecimiento de semillas.
- Sistema respiratorio y circulatorio.
- Persiana inteligente.
- Girasol.

Introducción

La robótica y el pensamiento computacional están **incluidos en el currículo oficial** de varias asignaturas, como biología y matemáticas.

Además, **es una buena manera de captar el interés del alumnado y motivarlos**. De esta manera pueden aprender la asignatura desde otro ángulo y realizar proyectos interdisciplinarios que le ayuden a tener una comprensión más global de los procesos.



En este documento se explicarán brevemente los proyectos y se enlazarán los programas, pero no se proporciona una guía detallada, contacta con tu asesor si quieres llevarlos al aula.

En su mayoría **los proyectos se han realizado con micro:bit** y los componentes asociados de la dotación de CE 4.0, ya que usa programación por bloques. Si se desea son fácilmente trasladables a Arduino.

MRU y MRUA con Maqueen y Micro:Bit I

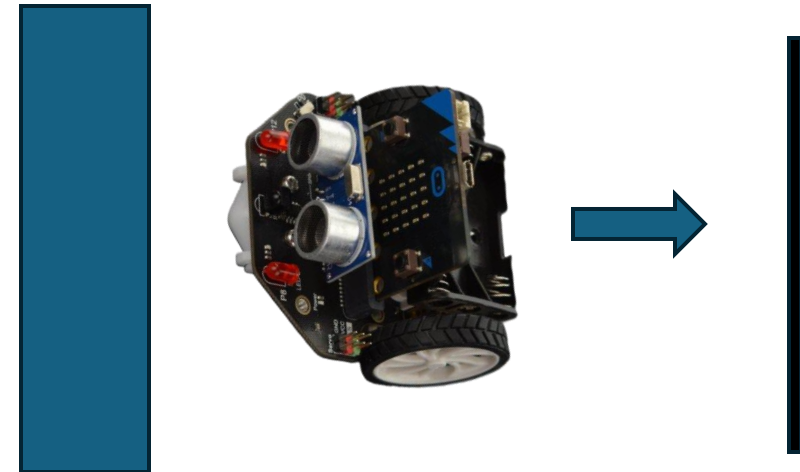
En esta práctica estudiaremos el movimiento rectilíneo uniforme y uniformemente acelerado utilizando la tarjeta Micro:Bit y el coche Maqueen con el sensor de ultrasonidos.

Programaremos el Maqueen para que **al presionar el botón A tomé la distancia inicial** a un objeto vertical que usaremos de referencia **y el tiempo inicial**, empezando después a moverse hacia atrás.

Cuando los sensores **siguelíneas lean dos negros** (línea situada detrás) el robot se parará y tomara las medidas de **distancia final y tiempo final**.

En el caso del **MRU** Maqueen se mueve siempre a la misma velocidad y **al presionar al botón a mostrará los datos de incremento de distancia e incremento de tiempo**.

Para el **MRUA** Maqueen va acelerando y **los incrementos de espacio y tiempo se almacenan en la placa usando la extensión datalogger**.



MRU y MRUA con Maqueen y Micro:Bit II

Es conveniente **tomar al menos una decena de datos**. Posteriormente se debe **calcular la velocidad o la aceleración media** según corresponda. Según prefiera cada profesor se puede hacer a mano, representando gráficas, usando hojas de cálculo,...

En la segunda parte de la práctica comprobaremos que nuestros datos y cálculos son correctos. Programaremos al robot para que recorra la distancia que deseemos cuando presionemos el botón. Para eso **utilizaremos las fórmulas del MRU/MRUA** junto con el valor obtenido de velocidad/aceleración **para calcular el tiempo durante el cual tienen que estar funcionando los motores del coche.**

Enlace a los programas:

- [MRU medición.](#)
- [MRU comprobación.](#)
- [MRUA medición.](#)
- [MRUA comprobación.](#)



Plano inclinado

Utilizaremos un cartón lo suficientemente grande para hacer una estructura sobre la que colocaremos dos sensores de infrarrojos. Luego programamos la micro:bit para que tome el tiempo inicial cuando soltemos la bola (primer infrarrojo) y el final cuando llegue al final (segundo infrarrojo).

Cuando construyamos la estructura es importante tener en cuenta el diámetro de la bola que vamos a usar, de manera que quepa, pero no vaya de un lado a otro.

Cuanto mayor sea la distancia entre los sensores, más fiable será la medida. Se puede utilizar una pieza como la de la imagen para evitar que la cabeza del sensor sobresalga del cartón y dificulte el movimiento de la bola.

El programa es muy sencillo, simplemente toma tiempo inicial/final cuando los sensores detectan la bola. En este caso **se ha utilizado la transmisión por línea, pero se puede modificar con datalogger o hacer que la toma de datos sea manual.**



[Enlace al programa.](#)

[Enlace al STL.](#)

El juego de la célula

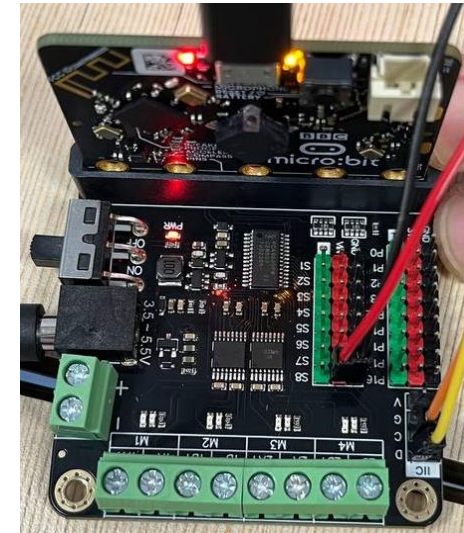
Esta práctica es un juego sencillo de preguntas y respuestas. Utilizaremos las listas de MakeCode (arreglos) para guardar las preguntas y sus respuestas y el display LCD del kit de Arduino para visualizarlas.

Programamos la placa para que al presionar el logo de micro:bit se muestre una pregunta al azar. Con el botón A vamos pasando las respuestas y con el B la seleccionamos para enviarla al programa y comprobar si es correcta. Al hacer un ruido alto repite la pregunta.

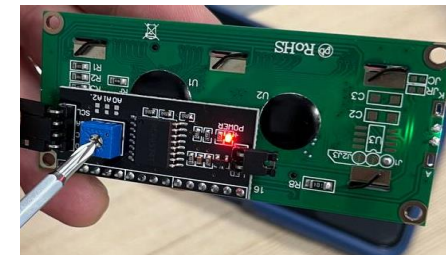
El juego es **fácilmente trasladable a cualquier materia**, incluso trabajar con frases y textos en lengua. Hay que tener en cuenta que **la LCD sólo visualiza 32 caracteres**, en el programa se ha utilizado un bucle para que vaya pasando los textos.

[Enlace al programa.](#)

[Actividades interdisciplinarias con robótica y pensamiento computacional](#) © 2026 by [Jaime Temes Mendoza](#) is licensed under [CC BY-NC-SA 4.0](#) 



Conexiones LCD-placa de expansión.



Potenciómetro para ajustar el brillo de la pantalla.

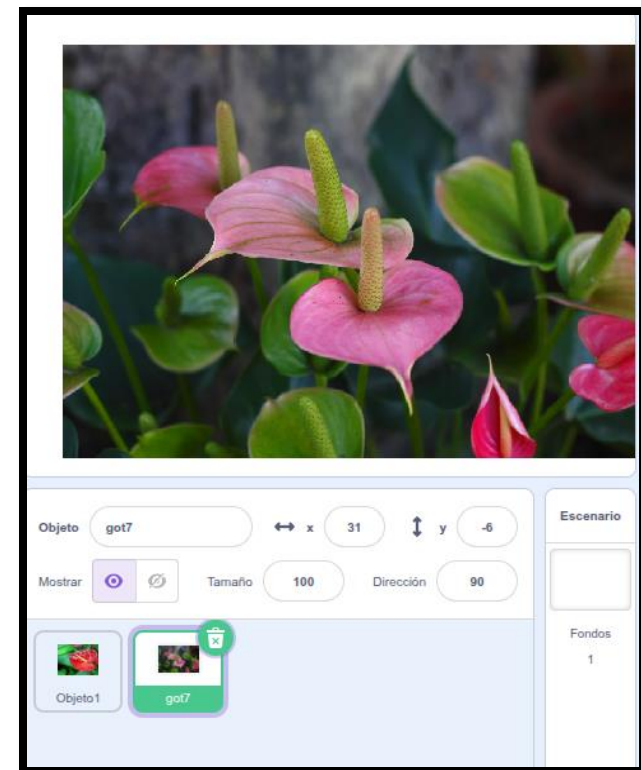
IA identificación de plantas

Utilizando la versión de [Scratch de MLlearning](#) podemos entrenar una IA que distinga entre distintas especies de plantas.

La parte más importante a la hora de entrenar la IA es la selección de imágenes que vamos a usar para cada planta. En el ejemplo la hemos entrenado con 6 imágenes de cada tipo y utilizado una séptima para comprobar el resultado.

Podemos hacer la práctica más interesante si es el propio alumnado el que hace las fotos con las que se entrenará el programa. Si estamos usando portátiles o dispositivos Android también tenemos la opción activar la cámara y que identifique el tipo de planta en tiempo real.

[Enlace a las imágenes y el programa.](#)



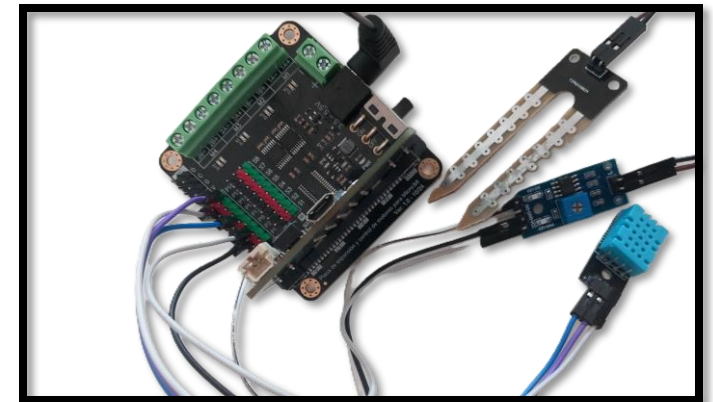
Crecimiento de semillas

En esta práctica vamos a monitorear las condiciones óptimas para el crecimiento de semillas. Se puede utilizar para hacer plantel o directamente para cultivar en el huerto.

Utilizaremos el sensor **DHT11** para medir la temperatura y humedad ambiental, el **sensor de humedad de suelo** y el **sensor de luz** de la placa. Hay que tener en cuenta que los valores del sensor de humedad de suelo oscilan entre 0 (máximo de humedad teórico) y 1023 (completamente seco).

En la programación se ha añadido la **toma de medidas a través de datalogger cada dos horas y al pulsar el botón A también de manera visual**.

El rábano es una de las semillas que menos tarda en germinar, tan sólo entre 3 y 7 días. Podemos preparar cuatro macetas con distintas condiciones de humedad de tierra, iluminación, temperatura (interior/exterior),... y observar cuál germina antes.

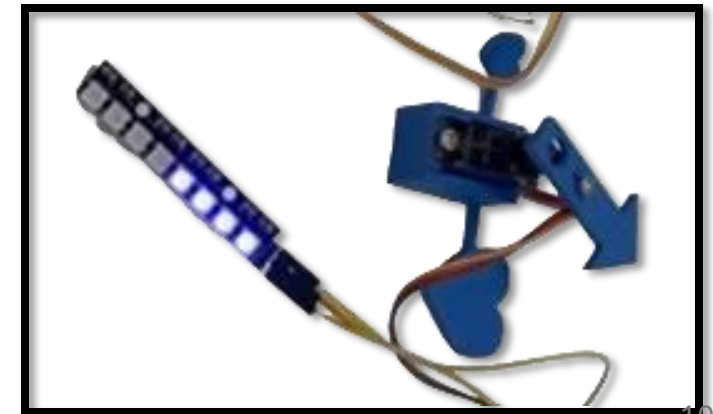
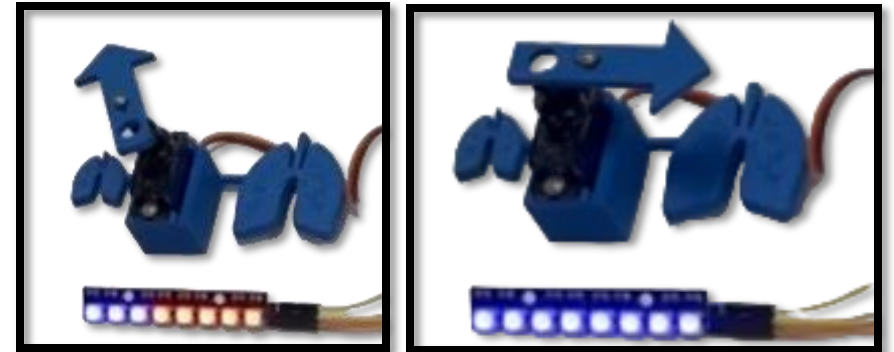


Sistema respiratorio y circulatorio I

Vamos a simular el sistema respiratorio y circulatorio usando servomotores de 180° y la tira de LEDs del kit. Para el sistema respiratorio el servo se irá moviendo simulando la compresión/expansión de los pulmones. Al expandir los pulmones se llenan de aire y los LED se van poniendo de color azul y al comprimir expulsa el aire y se ponen de color rojo.

Para el sistema respiratorio **el servo se irá moviendo simulando la compresión/expansión de los pulmones. Al expandir** los pulmones se llenan de aire y los LED se van poniendo de **color azul** y **al comprimir** expulsa el aire y se ponen de **color rojo**.

De igual manera **un servo simulará el movimiento del corazón** mientras que los **LEDs** irán cambiando de color, **azul para la sangre con oxígeno** y **rojo cuando vuelve sin**.



Sistema respiratorio y circulatorio II

Ambas programaciones se han realizado con los datos reales estándar de una persona sana en función del ritmo cardíaco medido en LPM (latidos por minuto). Los programas arrancan en 60 LPM y se puede aumentar y disminuir la frecuencia cardíaca.

Se pueden realizar ambas simulaciones de manera independiente: por un lado el respiratorio y por otro el circulatorio. También se adjuntan los programas para sincronizar ambos movimientos utilizando una tercera micro:bit. **Usaremos los botones A y B para aumentar/ disminuir la frecuencia cardíaca.** Cuando están sincronizados deberemos tocar el logotipo para encender/apagar.

Enlaces a programas:

- [Corazón independiente.](#)
- [Respiración independiente.](#)
- [Corazón sincronizado.](#)
- [Respiración sincronizado.](#)
- [Mando corazón y pulmón.](#)

Enlaces a STL:

- [Corazón servo.](#)
- [Pulmones servo.](#)
- [Flecha](#)

Persiana inteligente

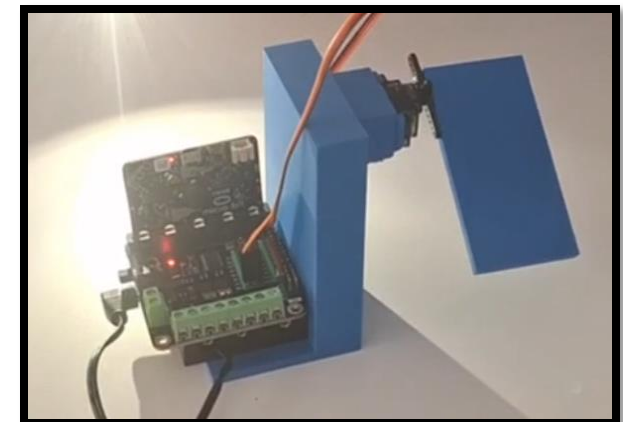
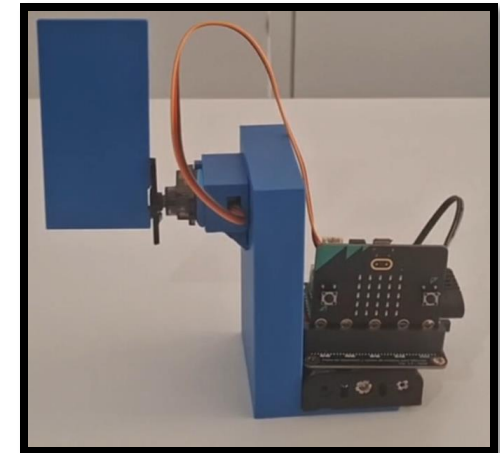
Con una programación muy simple podemos hacer que una persiana se abra o se cierre según la luz que incida sobre ella.

Este proyecto se puede implementar en las materias en las que se implemente domótica o también para mantener unas ciertas condiciones de iluminación sobre las plantas. Resultaría igualmente sencillo hacer que la persiana se abriera o cerrara según la temperatura o que activar un ventilador.

La programación es muy sencilla y consiste en hacer un mapeo, convertir de manera proporcional el valor obtenido por el sensor de luz de la micro:bit en el ángulo de posicionamiento del servomotor.

[Enlace al programa.](#)

[Enlace al STL.](#)



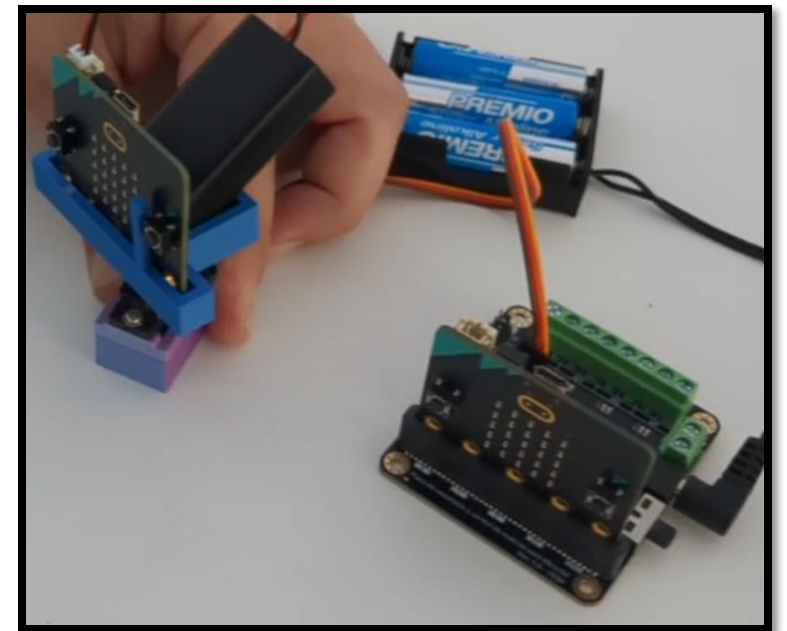
Girasol

Utilizaremos una micro:bit para tomar mediciones con el sensor de luz y otra para mover un servomotor de 180°. Con la primera calcularemos en qué ángulo ha detectado más luz y haremos que el servo se posicione para recibir la máxima iluminación.

Este proyecto se puede utilizar para posicionar placas solares, orientar el posicionamiento de plantas,... si en lugar del sensor de luz usamos el de sonido (micrófono) podremos averiguar la dirección del viento.

Al presionar el botón A el servo empieza a girar y la micro:bit hace las mediciones. Al presionar el botón B se posiciona en el punto en el que hay detectado más luz.

Aunque la programación es sencilla es necesario usar la radio de la microbit para que la tarjeta que está en el soporte le mande información sobre el posicionamiento del servo a la que se encuentra en la placa de expansión.



**Enlace al programa
[sensor y servo.](#)**

**Enlace al STL
[servo y micro:bit](#)**

[Actividades interdisciplinares con robótica y pensamiento computacional](#) © 2026 by [Jaime Temes Mendoza](#) is licensed under [CC BY-NC-SA 4.0](#)



¡Muchas gracias!