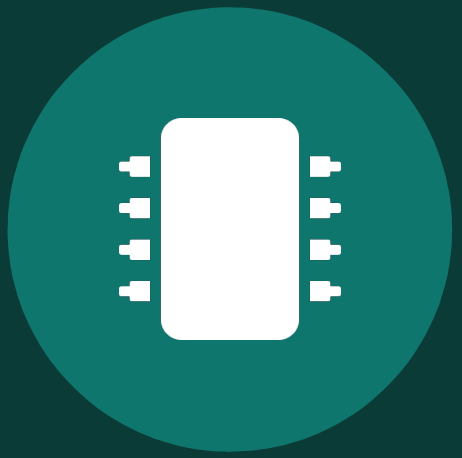
CLASE EXPRESS · CFIE DE LEÓN · MARTES, 20 DE OCTUBRE DE 2026

Learn to Code

Programar con micro:bit y bloques SAM en SAM Blockly



Pablo Núñez · 3 horas · para llevar al aula de 1.º y 2.º de ESO

Fabricotusideas.com/roboticaeducativa

Learn to Code · SAM Labs + micro:bit · Clase express · CFIE de León · CC BY-SA 4.0 · Iconos: Font Awesome Free (CC BY 4.0)

OBJETIVOS

Hoy salimos de aquí sabiendo…



1. Del cajón a funcionando

Poner en marcha el kit con Chromebook: firmware de la micro:bit, emparejado y proyecto SAM Blockly.



2. Crear actividades

Programar secuencias, eventos, bucles, condicionales y variables con sensores reales.



3. Llevarlo al aula

Esbozar una situación de aprendizaje para 1.º–2.º de ESO con el kit.

Learn to Code · SAM Labs + micro:bit · Clase express · CFIE de León · CC BY-SA 4.0 · Iconos: Font Awesome Free (CC BY 4.0)

AGENDA · 3 HORAS

La sesión de hoy

0:00–0:15	**Bienvenida**	15 min · Teoría
0:15–0:45	**Del cajón a funcionando**	30 min · Práctica guiada
0:45–1:30	**Primeros retos guiados (1–5)**	45 min · Práctica
1:30–1:40	Descanso	10 min · —
1:40–2:10	**Mini-proyecto como alumnado**	30 min · Práctica
2:10–2:50	**Llevarlo a mi aula**	40 min · Práctica
2:50–3:00	**Dudas, recursos y cuestionario**	10 min · Teoría

ENCAJE CURRICULAR

¿Por qué en 1.º y 2.º de ESO?

CD5

Competencia digital

Desarrollar soluciones tecnológicas con pensamiento computacional

STEM

Ciencia y tecnología

Medir, decidir con datos y construir prototipos

CyL

Decreto 39/2022

Tecnología y Digitalización: programación y proyectos

El kit es una excusa: lo importante es pensar, probar, equivocarse y mejorar.

Learn to Code · SAM Labs + micro:bit · Clase express · CFIE de León · CC BY-SA 4.0 · Iconos: Font Awesome Free (CC BY 4.0)

ANTES DE EMPAREJAR

Chromebook listo en 4 comprobaciones

ChromeOS 134 o posterior

Ajustes → Información de ChromeOS. Las versiones 123–133 fallan con el Bluetooth de los bloques

Bluetooth activado

En la bandeja del sistema, abajo a la derecha

Chrome actualizado

SAM Studio no conecta por Bluetooth en otros navegadores

Token a mano

El que comparte el docente desde SAM Studio

¿No puedes actualizar? En chrome://flags pon «Use Floss instead of BlueZ» en Disabled y reinicia.

Learn to Code · SAM Labs + micro:bit · Clase express · CFIE de León · CC BY-SA 4.0 · Iconos: Font Awesome Free (CC BY 4.0)

0:15

Del cajón a funcionando

30 minutos · Chromebook



Learn to Code · SAM Labs + micro:bit · Clase express · CFIE de León · CC BY-SA 4.0 · Iconos: Font Awesome Free (CC BY 4.0)


Classroom Size
Learn to Code Kit
featuring micro:bit
Flare Alert

¿Qué hay en cada kit?



Slider
Entrada: posición del slider



Sensor de luz
Entrada: cuánta luz hay



Luz RGB
Salida: cualquier color

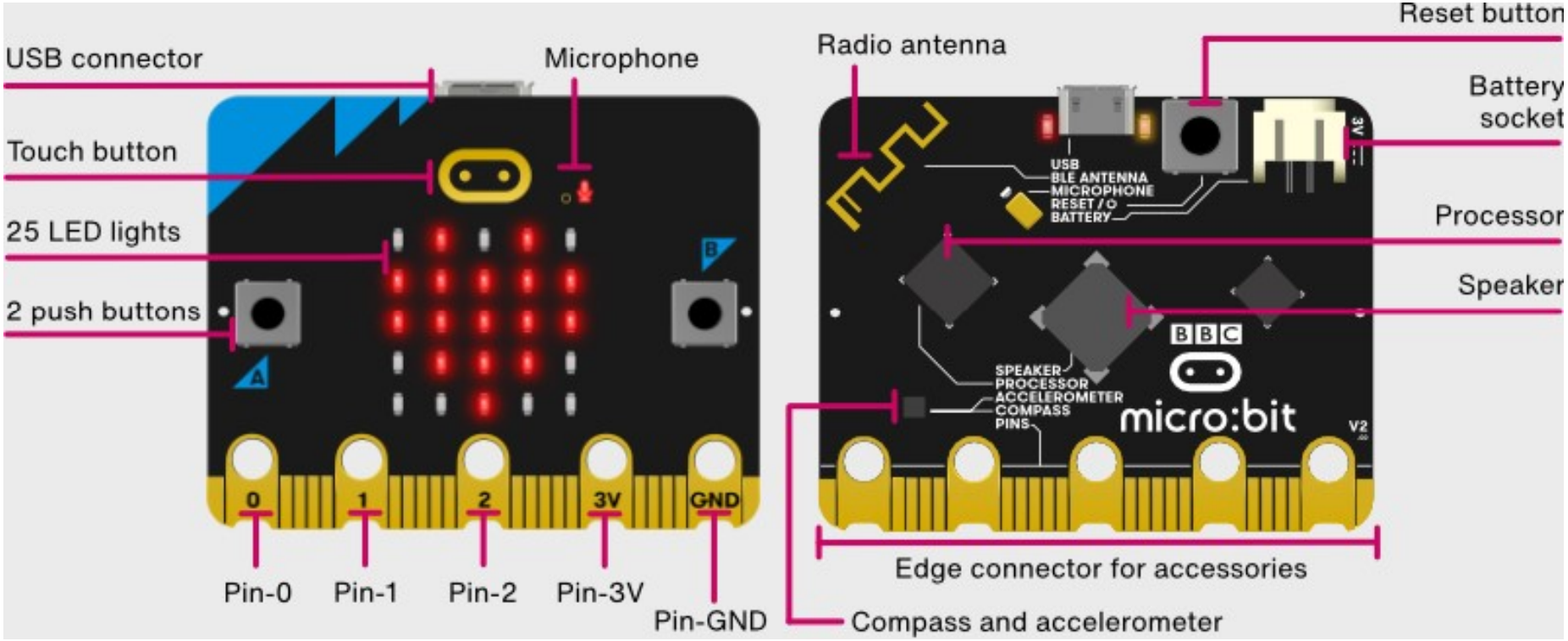


Microbit
Placa electrónica programable con pines abiertos.

Acelerómetro: Mide el movimiento y la aceleración en tres dimensiones (detecta inclinaciones hacia la izquierda, derecha, adelante, atrás, arriba o abajo, además de impactos o sacudidas).
Brújula (Magnetómetro): Detecta el norte magnético y mide la fuerza de los campos magnéticos cercanos.
Sensor de temperatura: Mide la temperatura ambiente (integrada en el procesador principal).
Sensor de luz ambiente: Utiliza los propios diodos de la matriz de LED de la placa para estimar la intensidad de la luz exterior.
Micrófono (solo en la versión micro:bit V2): Permite detectar niveles de sonido y ruido ambiental.
Sensor táctil (solo en la versión micro:bit V2): El logotipo dorado de la parte frontal funciona como un botón táctil capacitivo.
Botones físicos (A y B): Dos botones mecánicos en la placa que actúan como entradas de pulsación.
Antena de radio y Bluetooth: Permite enviar y recibir datos inalámbricos con otras placas micro:bit o dispositivos externos.

Materiales para un equipo: 1 microbit + cable usb + batery holder + 1 slider + 1 sensor de luz + luz rgb

El cerebro de la bestia: ¿Qué hay dentro de una Microbit?

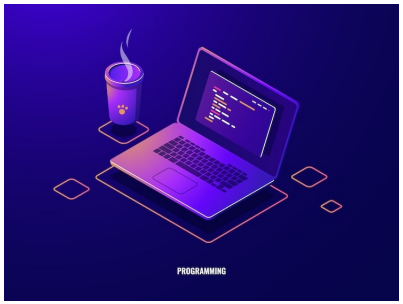


¿Cómo se puede programar la microbit?

SAM STUDIO

Microsc
MakeCo

https://makecode.microbit.org/



JavaScript, Python, Scratch (añadiendo una extensión) y Tickle (aplicación para iPad).

ENTRAR EN LA PLATAFORMA

Dos webs, dos usos

Contenidos · docente

www.samlabs.es/es-learntocode — **Learn to Code Portal**

- Entra con el correo del centro (34xxxxxx@educa.jcyl.es)
- Contraseña: la que facilita el CFIE o el coordinador TIC
- Lecciones en castellano: guía docente, diapositivas y ficha del alumnado

Programar · alumnado

studio.samlabs.com — **Activar token**

- El docente crea la clase en SAM Studio y comparte el token del estudiante (o su QR)
- El alumnado entra en studio.samlabs.com y activa el token, sin crear cuenta
- Siempre con Chrome (o Edge): Firefox no conecta por Bluetooth

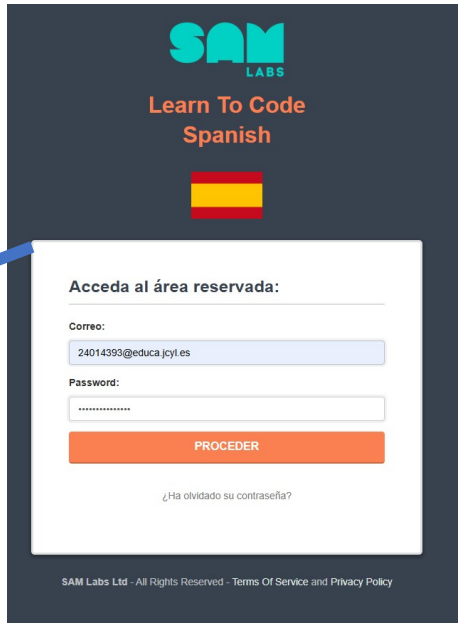
Learn to Code · SAM Labs + micro:bit · Clase express · CFIE de León · CC BY-SA 4.0 · Iconos: Font Awesome Free (CC BY 4.0)

Primera configuración de la plataforma LEARN TO CODE

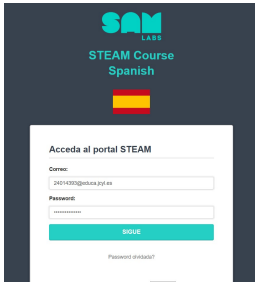
La plataforma de programación SÓLO funciona con Chrome o Chromebook

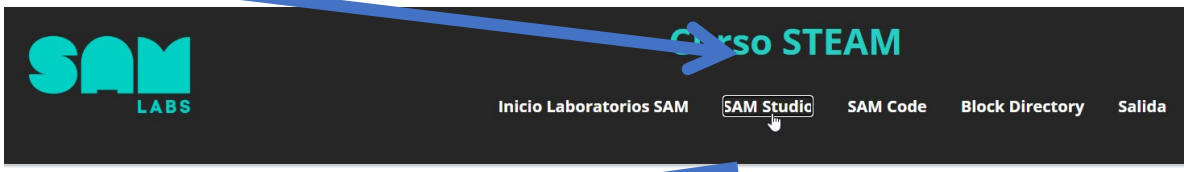
Entra en la web: https://www.samlabs.es/es-learntocode
Pon tu email de centro y la contraseña **SamlabsCyl2026!**

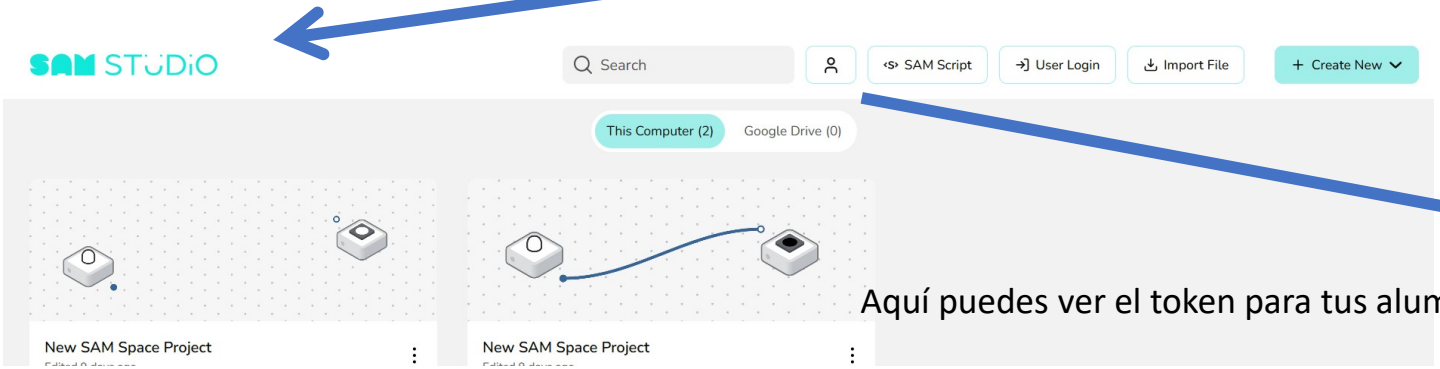




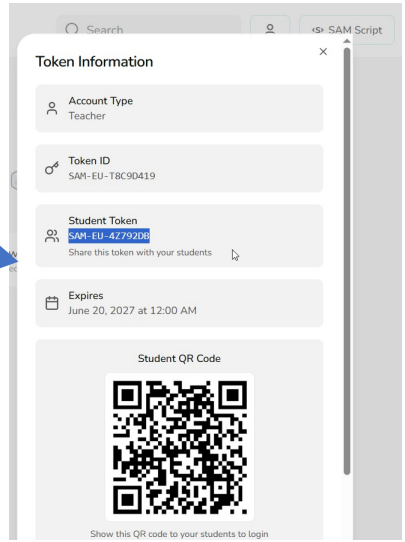
Entrar como profesor a la plataforma





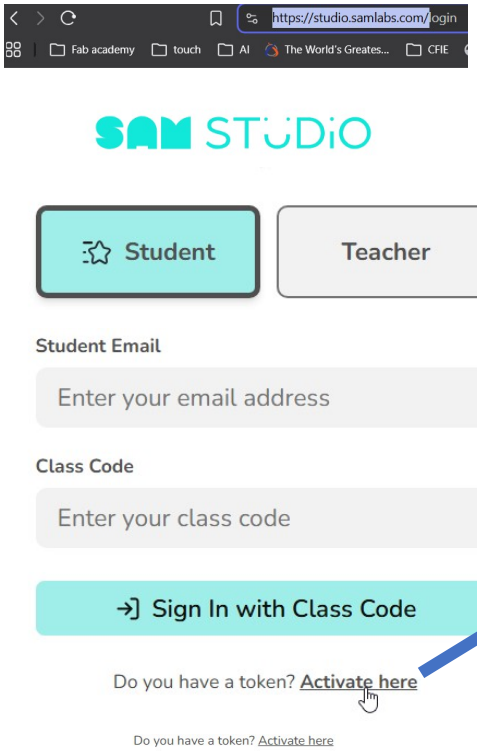


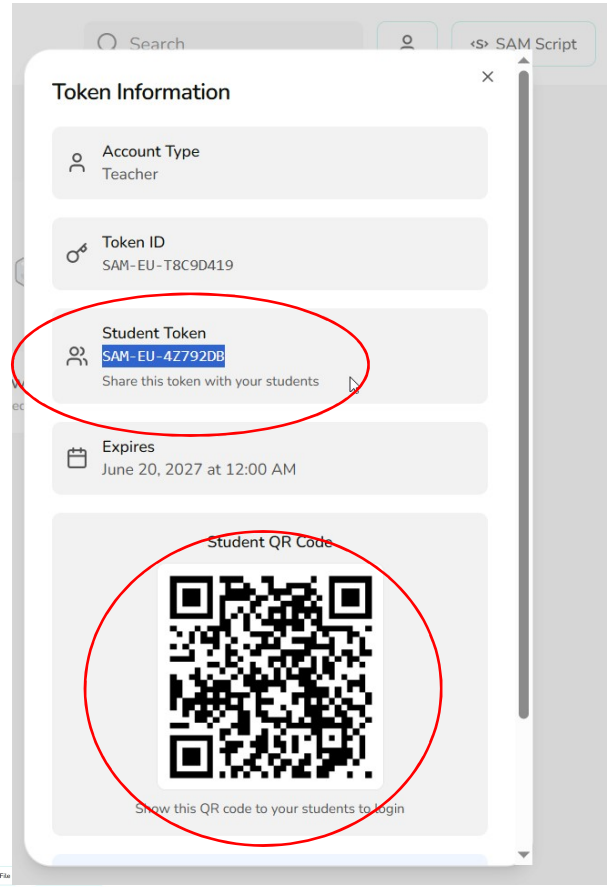
Aquí puedes ver el token para tus alumnos

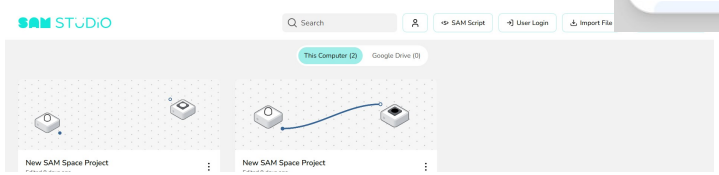


Entrada de alumnos en la plataforma

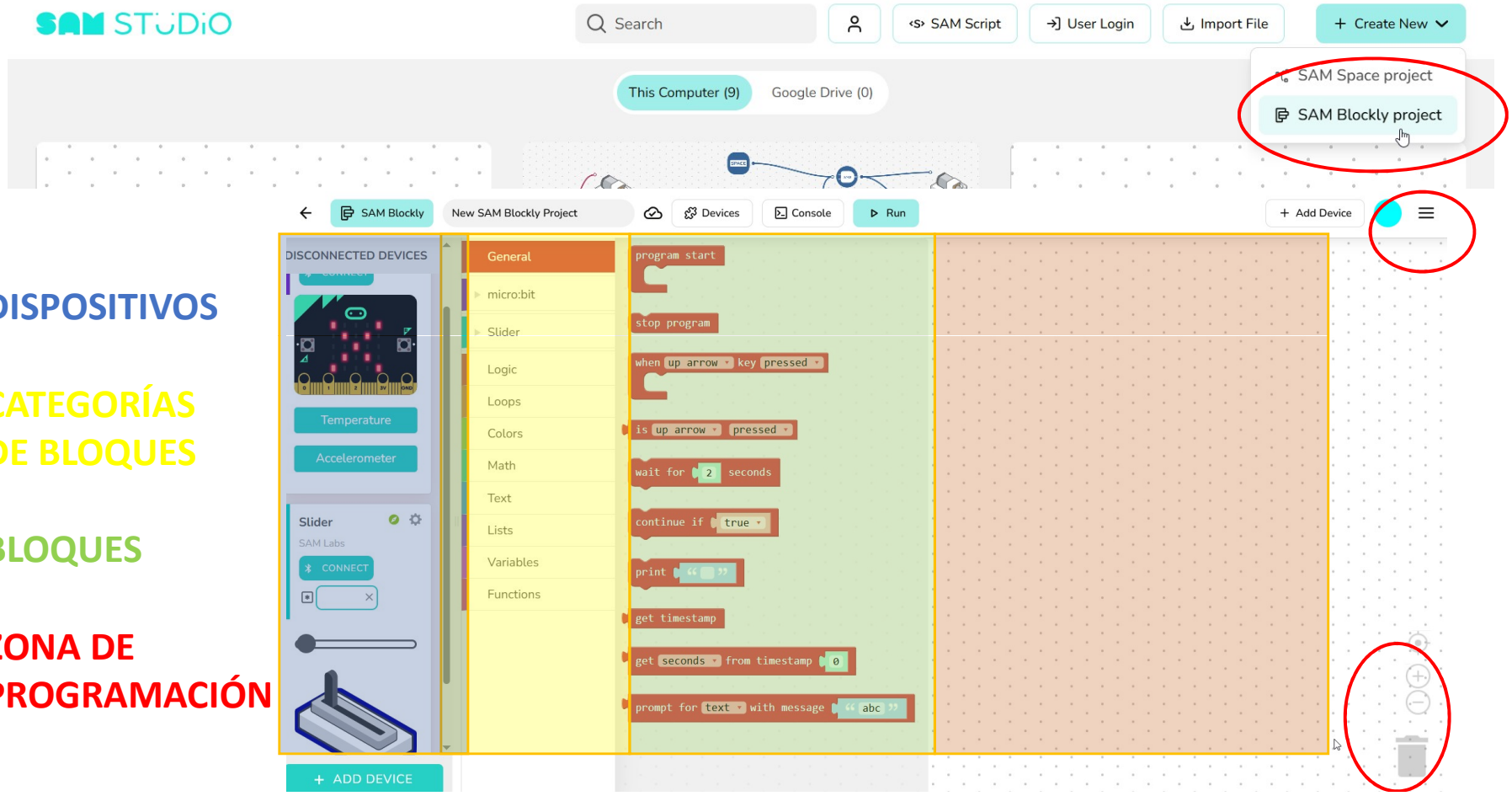
https://studio.samlabs.com/







El entorno de programación SAM Blockly



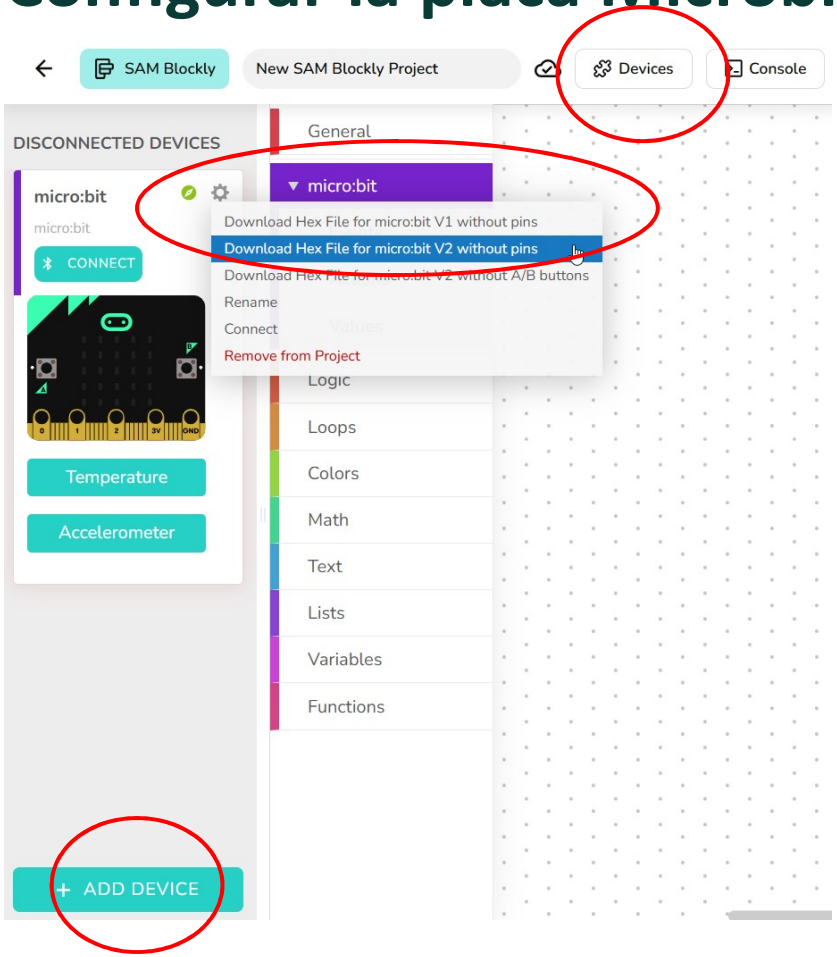
DISPOSITIVOS

CATEGORÍAS DE BLOQUES

BLOQUES

ZONA DE PROGRAMACIÓN

Configurar la placa Microbit (sólo una vez)

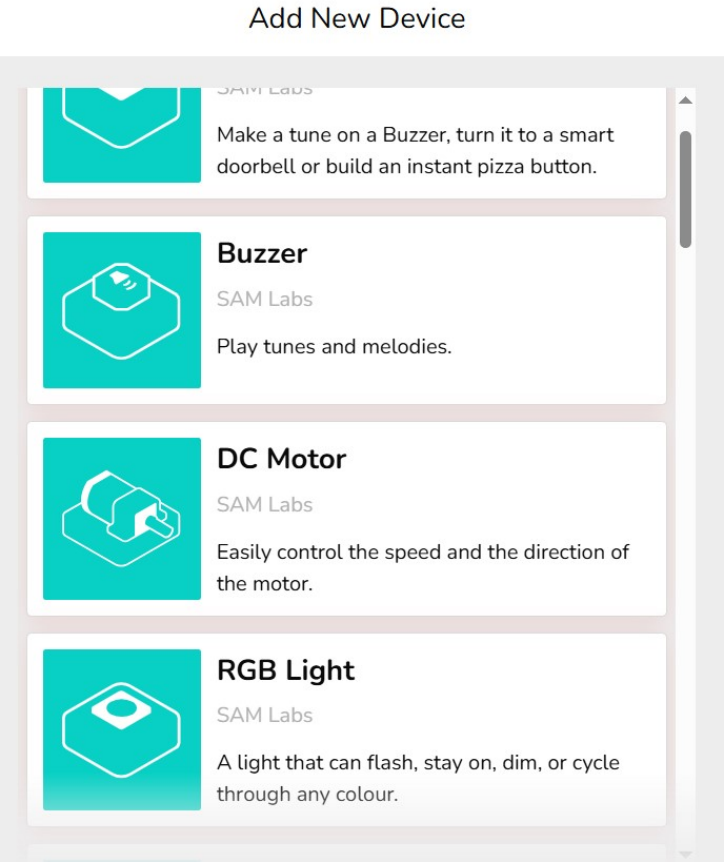


Hacer click en “Devices”
Add device
Elegir la microbit de la lista
Hacer click en la rueda de confuguración
Elegir “Download … microbit V2 without pins”
Descargar el fichero
Conectar la microbit al ordenador por USB
Copiar el fichero descargado dentro de la microbit.
Esperar a que se configure.
Aparecerá un texto en la pantalla de la microbit

Conectar por bluetooth a la microbit



También podemos añadir los modulos de SAM LABS



¿Qué es el hardware SAM?



El Light Sensor detecta la luz en la habitación y puede devolver un valor entre '0-100'.

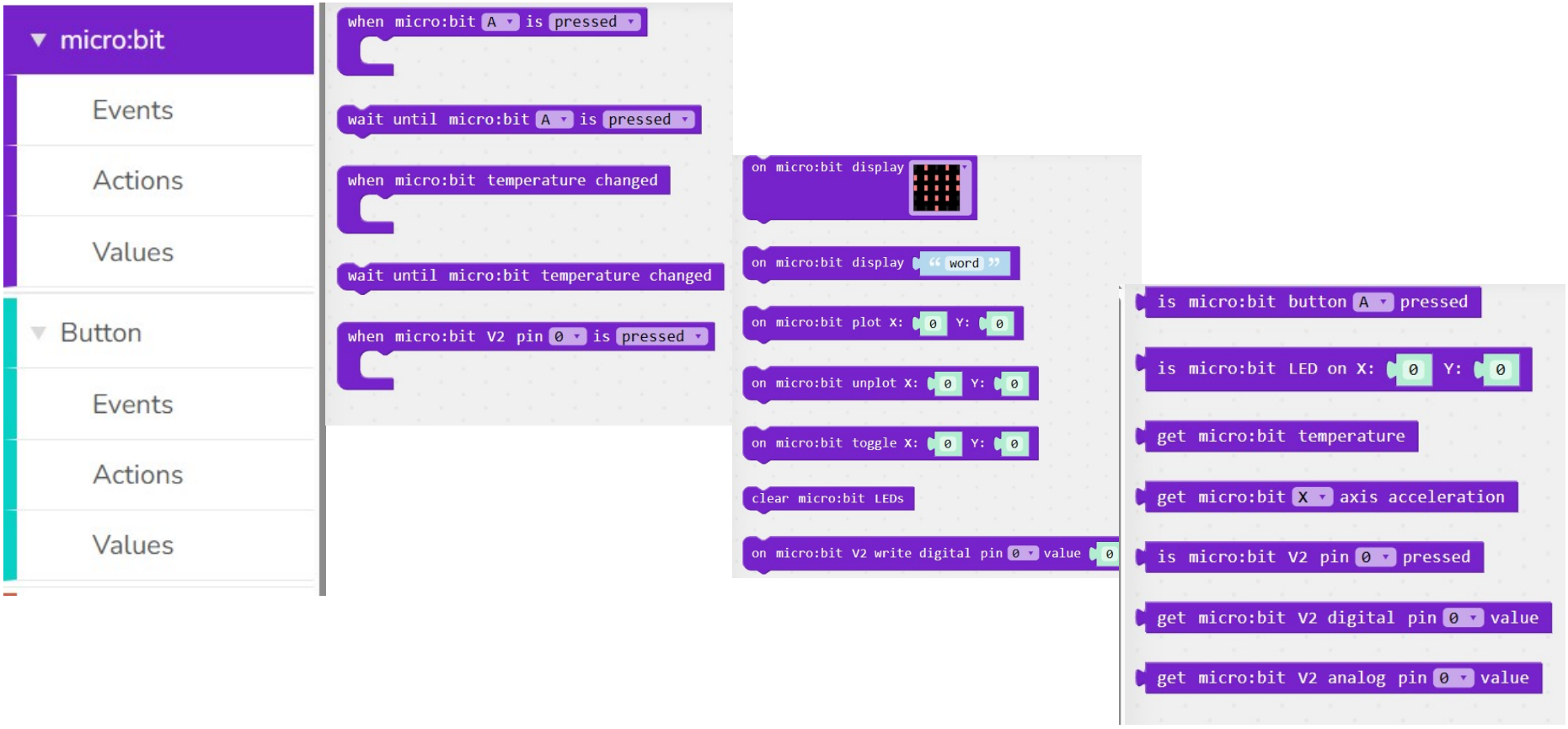


El Slider puede moverse de izquierda a derecha y devolver un valor entre '0-100'.



El Buzzer puede reproducir sonidos de diferentes tonos dentro de un rango de '0-100'.

Al conectar un dispositivo nos aparecerán nuevos bloques



ANTES DE EMPEZAR

Carga y encendido de los bloques

45 min

de carga dan unas 8 horas de uso

2–3 s

pulsando el botón sobre «SAM Labs» para encender

Rojo

luz roja: encendido y listo para emparejar

Cuidado con el conector micro-USB: entra en un solo sentido y se rompe si se fuerza. Usa los cables de carga múltiple del kit y deja todo cargando la víspera.

Learn to Code · SAM Labs + micro:bit · Clase express · CFIE de León · CC BY-SA 4.0 · Iconos: Font Awesome Free (CC BY 4.0)

PUESTA EN MARCHA

5 pasos para conectar



1 SAM Blockly

studio.samlabs.com → + Create New → SAM Blockly project

2 Firmware

Devices → micro:bit → engranaje → Download Hex (V2)

3 Copiar

App Archivos: arrastra el hex de Descargas a la unidad MICROBIT

4 Conectar

CONNECT → BBC micro:bit → Emparejar: «SAM LABS»

5 Bloques

Enciende (luz roja), CONNECT y ponles nombre: «Luz-07»

Si hay muchos bloques y no encuentras el tuyo: busca su ID de 4 dígitos o apaga los que no usas y empareja de uno en uno.

Learn to Code · SAM Labs + micro:bit · Clase express · CFIE de León · CC BY-SA 4.0 · Iconos: Font Awesome Free (CC BY 4.0)

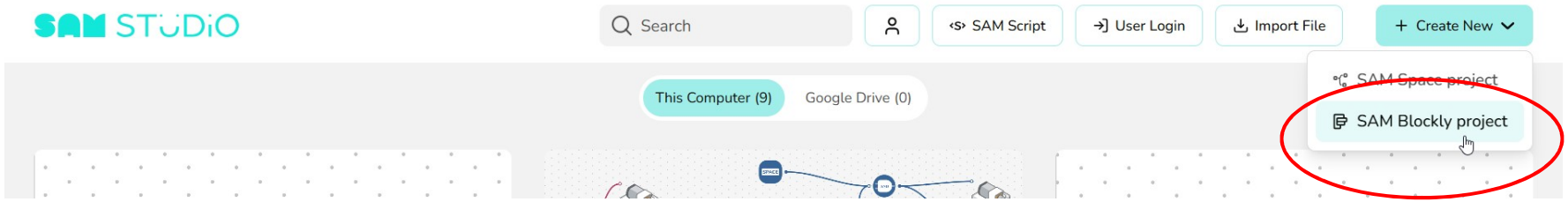
Testeando la microbit

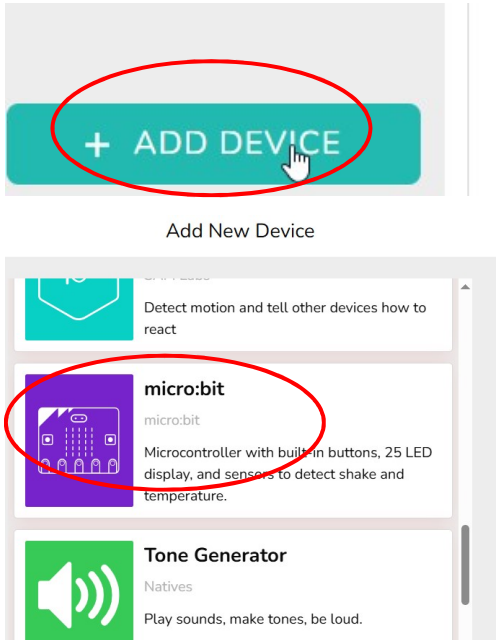
Primeros retos guiados

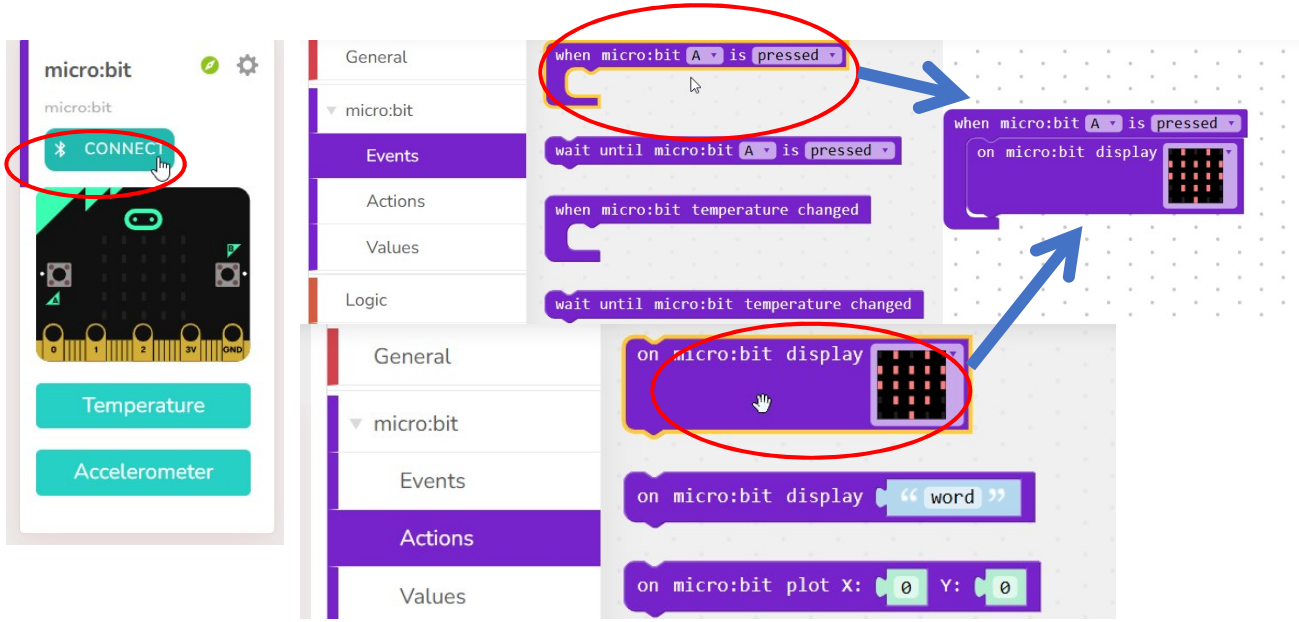


Learn to Code · SAM Labs + micro:bit · Clase express · CFIE de León · CC BY-SA 4.0 · Iconos: Font Awesome Free (CC BY 4.0)

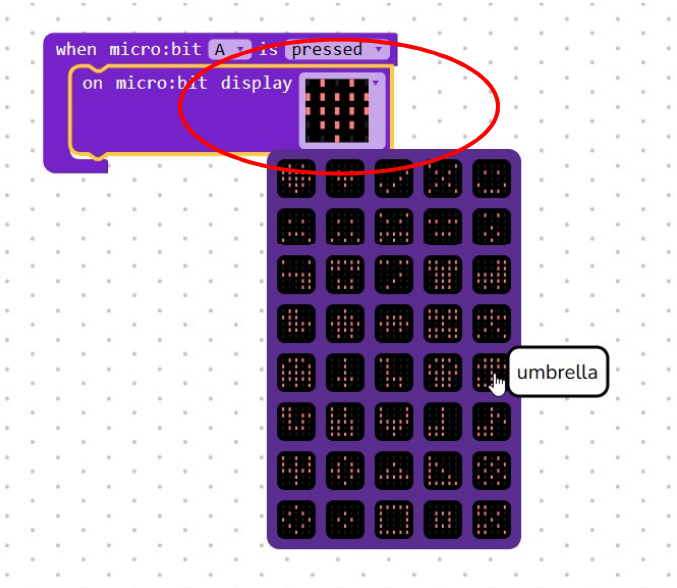
Prorgamar la pantalla de leds

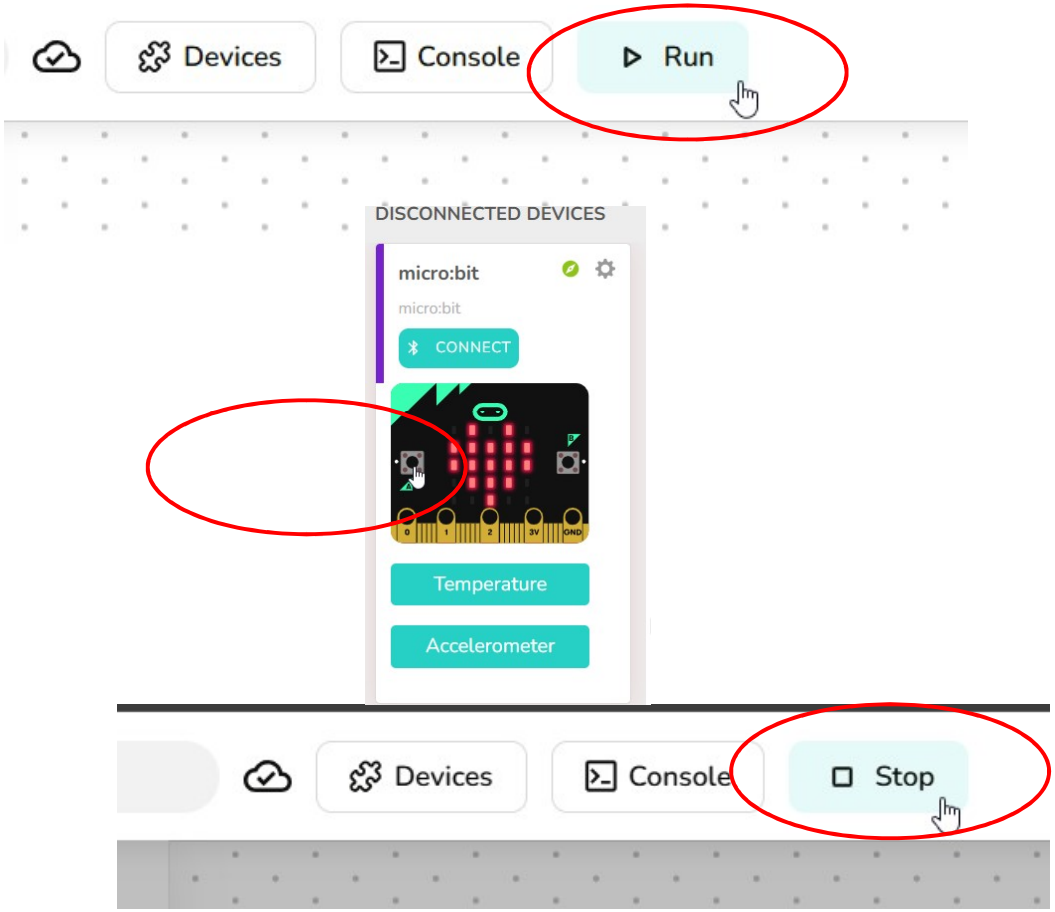




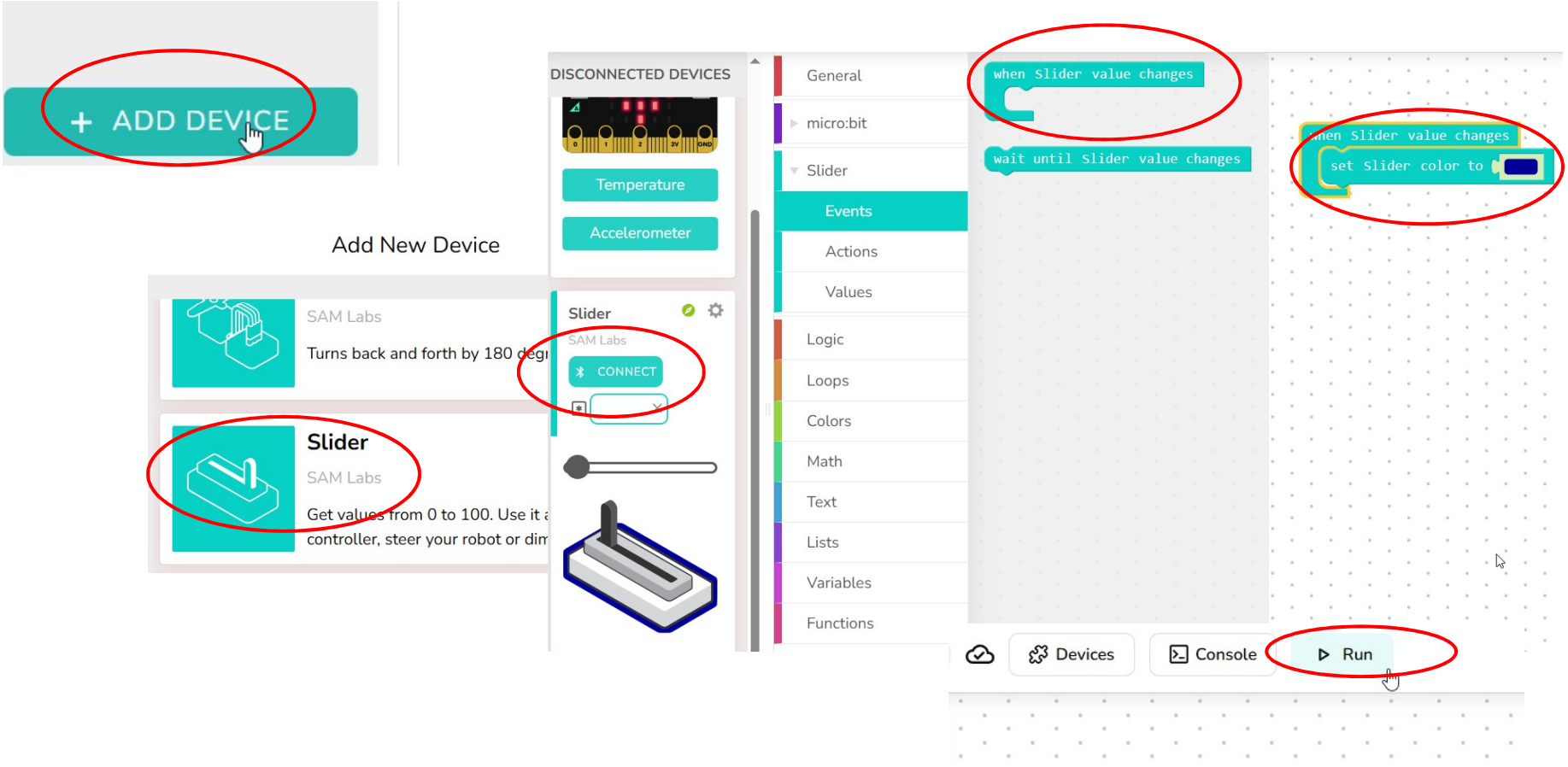


Programar la pantalla de leds

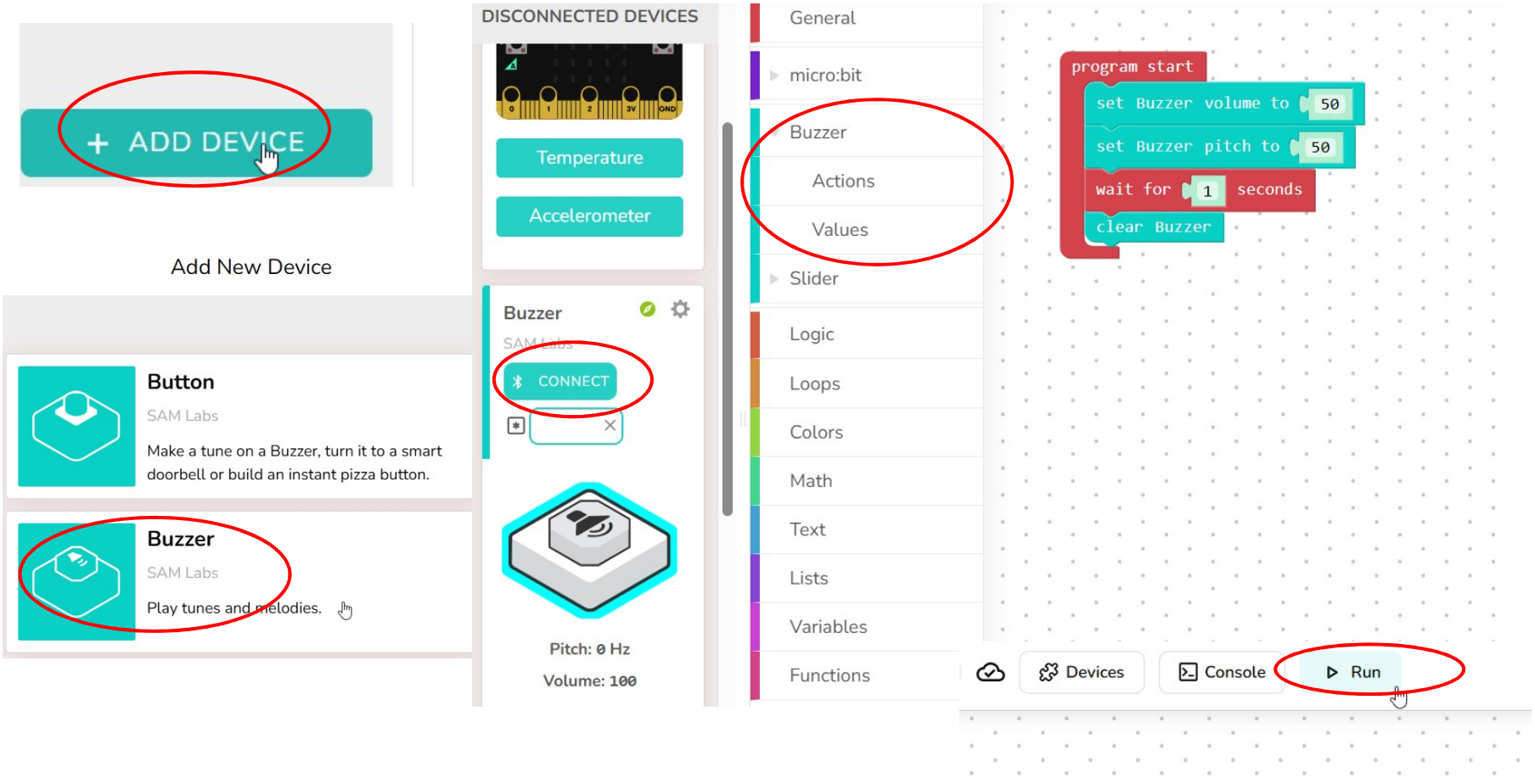




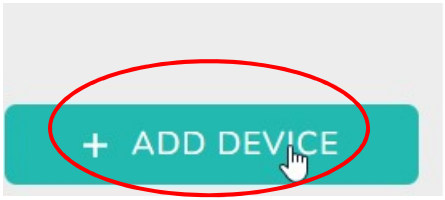
Programando el bloque SLIDER

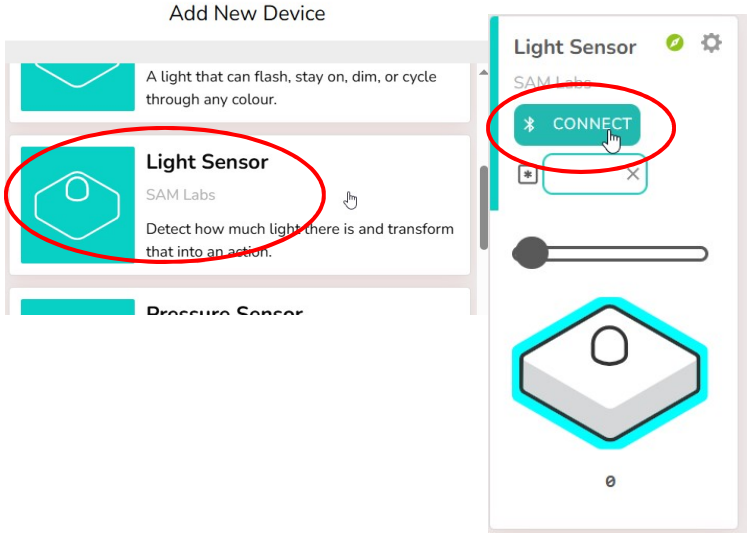


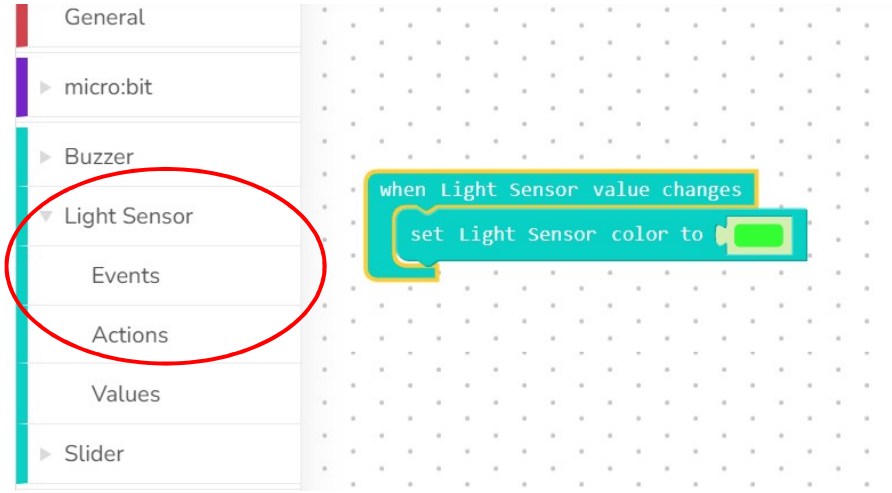
Programando el bloque BUZZER

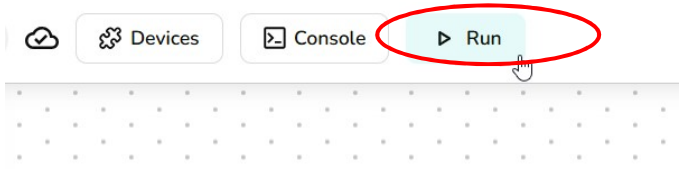


Programando el bloque SENSOR DE LUZ



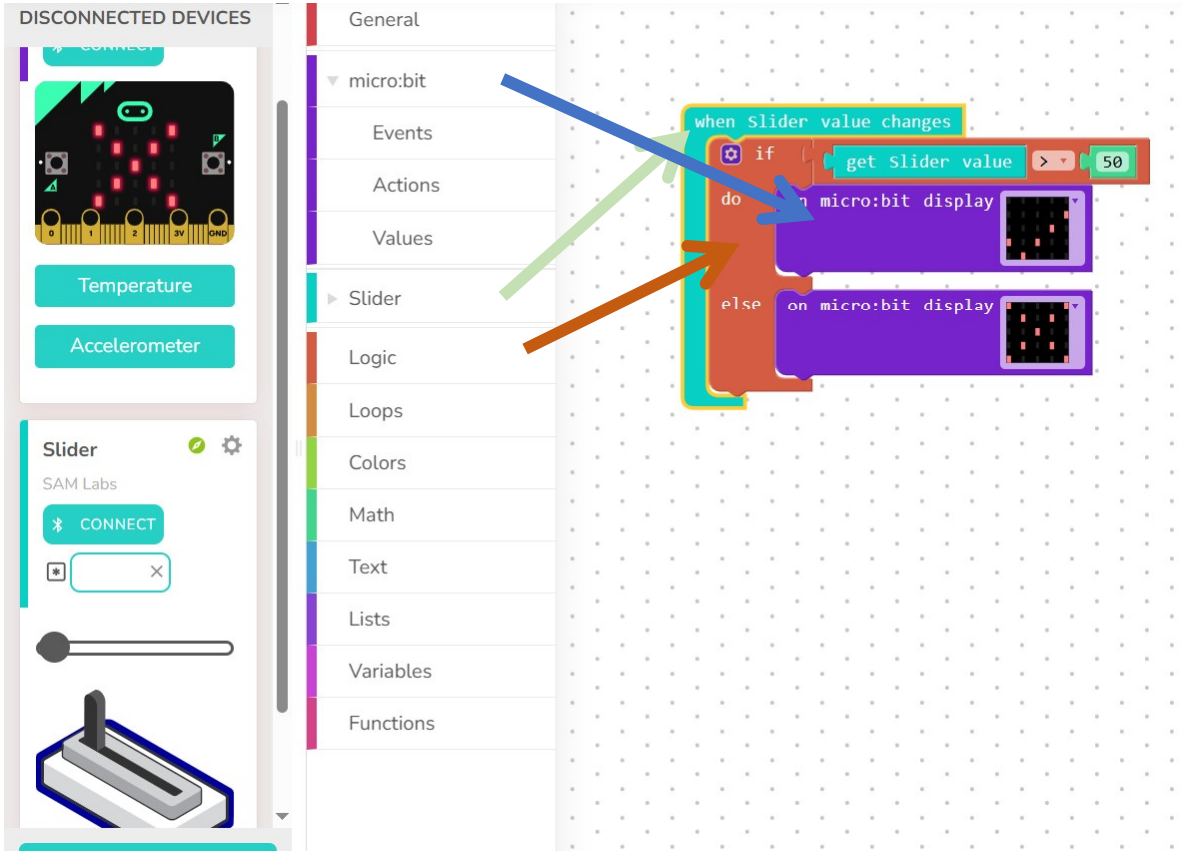






Interconectando componentes

Queremos conseguir que nuestro código cambie en el display de la microbit en función de la posición del slider:
V para valores >50
X para el resto



0:45

Primeros retos guiados

Retos 1–4 para todos y el 5 de ampliación. Cada reto, una idea de programación



MAPA DE RETOS

Del primer programa al mini-proyecto

1 Secuencia de arranque

Algoritmo y secuencia

micro:bit + timbre

2 Botones que responden

Eventos

micro:bit + timbre

3 El theremin de clase

Entrada → salida

Deslizador + timbre

4 Alarma de oscuridad

Bucle y condicional

Sensor de luz + micro:bit + timbre

5 Umbral a medida

Variables

Sensor de luz + deslizador + micro:bit

★ Farola inteligente

Integración

Todo el kit

Learn to Code · SAM Labs + micro:bit · Clase express · CFIE de León · CC BY-SA 4.0 · Iconos: Font Awesome Free (CC BY 4.0)

BÁSICO · ALGORITMO Y SECUENCIA

Reto 1 · Secuencia de arranque

Tu misión

Vas a programar la secuencia de encendido de un robot imaginario. Al empezar el programa, la micro:bit tiene que mostrar por la pantalla el texto “HOLA”, después muestra el icono de corazón“animado” y, al final, un pitido corto del BUZZER y en la pantalla una marca de «listo».

Pruébalo: ¿Se ve cada paso? Si todo pasa demasiado deprisa, cambia los tiempos de espera. Eso es depurar un algoritmo.

Bloques clave

- program start
- on micro:bit display ("texto")
- on micro:bit display [icono]
- wait for (1) seconds
- set Buzzer volume to (), set Buzzer pitch to () y clear Buzzer

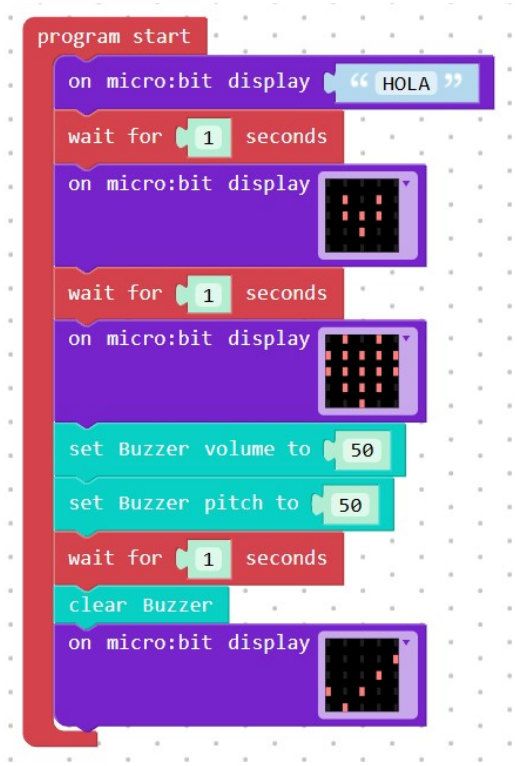
Material: micro:bit + timbre

Learn to Code · SAM Labs + micro:bit · Clase express · CFIE de León · CC BY-SA 4.0 · Iconos: Font Awesome Free (CC BY 4.0)

SECUENCIA DE ARRANQUE

Solución · Reto 1

```
program start
  on micro:bit display ("HOLA")
  wait for (1) seconds
  on micro:bit display [small diamond]
  wait for (0.5) seconds
  on micro:bit display [diamond]
  wait for (0.5) seconds
  set Buzzer volume to (50)
  set Buzzer pitch to (50)
  wait for (1) seconds
  clear Buzzer
  on micro:bit display [yes]
```



Claves

- Pon set Buzzer volume antes de set Buzzer pitch: así se evitan ruidos al apagar.
- Error típico: no poner esperas, así que solo se ve el último icono. Es una buena excusa para hablar del orden y el tiempo en un algoritmo.

Si te sobra tiempo: Crea una secuencia de apagado con el botón B.

Learn to Code · SAM Labs + micro:bit · Clase express · CFIE de León · CC BY-SA 4.0 · Iconos: Font Awesome Free (CC BY 4.0)

BÁSICO · EVENTOS

Reto 2 · Botones que responden

Tu misión

Haz que la micro:bit reaccione a los botones: con A muestra un corazón y suena un tono agudo en el buzzer; apretando B muestra una X y suena un tono grave en el buzzer.

Pruébalo: Pulsa A y B varias veces seguidas. ¿Qué ocurre si pulsas los dos a la vez?

Bloques clave

- when micro:bit [A] is [pressed]
- on micro:bit display [icono]
- set Buzzer pitch to ()
- wait for () seconds
- clear Buzzer

Material: micro:bit + timbre

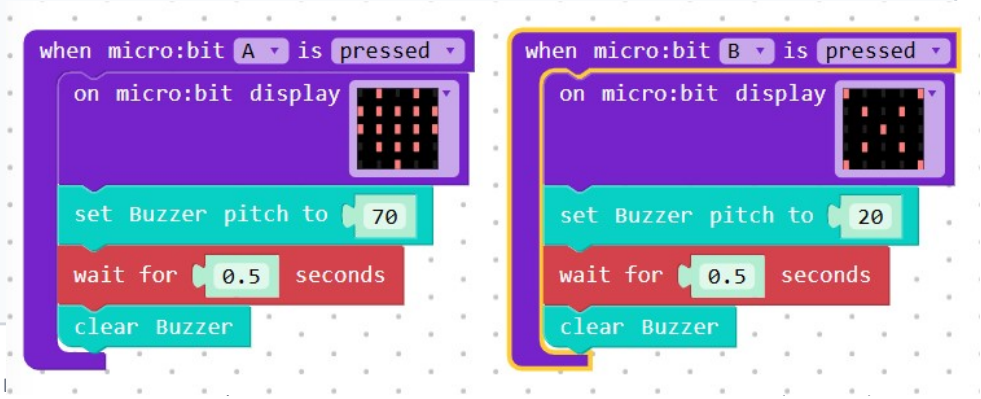
Learn to Code · SAM Labs + micro:bit · Clase express · CFIE de León · CC BY-SA 4.0 · Iconos: Font Awesome Free (CC BY 4.0)

BOTONES QUE RESPONDEN

Solución · Reto 2

```
when micro:bit [A] is [pressed]
  on micro:bit display [heart]
  set Buzzer pitch to (70)
  wait for (0.5) seconds
  clear Buzzer

when micro:bit [B] is [pressed]
  on micro:bit display [no]
  set Buzzer pitch to (20)
  wait for (0.5) seconds
  clear Buzzer
```



Claves

- Cada evento es un programa independiente dentro del mismo espacio de trabajo.
- El tono va de 0 (silencio) a 100 (más agudo).

Si te sobra tiempo: Convierte la micro:bit en un pulsador de concurso: A es «respuesta correcta» y B, «incorrecta».

BÁSICO · ENTRADA → SALIDA

Reto 3 · El theremin de clase

Tu misión

Construye un theremin, un instrumento que se toca sin tocarlo: al mover el deslizador, el buzzer cambia de tono en función de la posición del slider.

Pruébalo: El deslizador va de 0 a 100, igual que el tono del timbre. ¿Qué pasa en la posición 0?

Bloques clave

- when Slider value changes
- get Slider value
- set Buzzer pitch to ()
- set Buzzer volume to ()

Material: Deslizador + timbre

Learn to Code · SAM Labs + micro:bit · Clase express · CFIE de León · CC BY-SA 4.0 · Iconos: Font Awesome Free (CC BY 4.0)

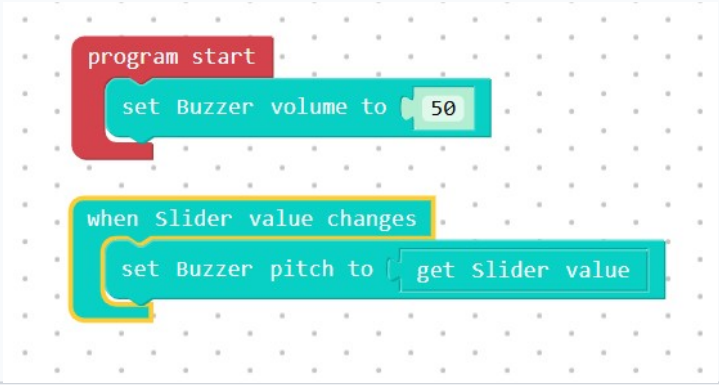
EL THEREMIN DE CLASE

Solución · Reto 3

```
program start
  set Buzzer volume to (50)

when Slider value changes
  set Buzzer pitch to (get Slider value)

when micro:bit [B] is [pressed]      ← ampliación
  clear Buzzer
```



Claves

- Idea clave: una entrada (deslizador) controla una salida (timbre). El valor viaja directamente de un bloque a otro.

Si te sobra tiempo: Añade la micro:bit: con el botón B el timbre se calla (clear Buzzer).

Learn to Code · SAM Labs + micro:bit · Clase express · CFIE de León · CC BY-SA 4.0 · Iconos: Font Awesome Free (CC BY 4.0)

MEDIO · BUCLE Y CONDICIONAL

Reto 4 · Alarma de oscuridad

Tu misión

Crea una alarma de oscuridad: si hay poca luz, la micro:bit muestra un aviso y el timbre suena. Si vuelve la luz, todo se calma.

Pruébalo: Tapa el sensor con la mano. Antes de programar, mira el valor del sensor con luz y tapado en el panel del dispositivo y elige un umbral entre los dos.

Bloques clave

- repeat forever
- if … do … else
- Comparador <
- get Light Sensor value
- on micro:bit display [icono]
- set Buzzer pitch to () y clear Buzzer

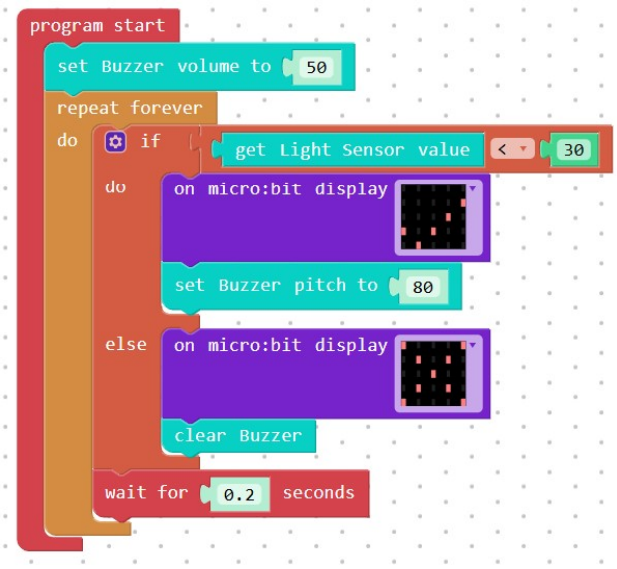
Material: Sensor de luz + micro:bit + timbre

Learn to Code · SAM Labs + micro:bit · Clase express · CFIE de León · CC BY-SA 4.0 · Iconos: Font Awesome Free (CC BY 4.0)

ALARMA DE OSCURIDAD

Solución · Reto 4

```
program start
  set Buzzer volume to (50)
  repeat forever
    if (get Light Sensor value) < (30)
      do   on micro:bit display [no]
           set Buzzer pitch to (80)
      else on micro:bit display [yes]
           clear Buzzer
    wait for (0.2) seconds
```



Claves

- El 30 es solo un ejemplo: cada aula tiene una luz distinta.
- La pequeña espera dentro del bucle evita saturar la conexión Bluetooth.

Si te sobra tiempo: Añade un tercer estado con else if: poca luz, un pitido suave; muy poca luz, un pitido fuerte.

Learn to Code · SAM Labs + micro:bit · Clase express · CFIE de León · CC BY-SA 4.0 · Iconos: Font Awesome Free (CC BY 4.0)

AMPLIACIÓN · VARIABLES

Reto 5 · Umbral a medida

Tu misión

Mejora la alarma: el umbral ya no está escrito en el programa, sino que lo elige el usuario con el deslizador y se guarda en una variable. Con el botón B, la micro:bit enseña el umbral actual por pantalla.

Pruébalo: Cambia el umbral con el deslizador y comprueba que la alarma salta antes o después.

Bloques clave

- Variables: set umbral to () y umbral
- when Slider value changes
- get Slider value
- when micro:bit [B] is [pressed]
- La alarma del reto 4

Material: Sensor de luz + deslizador + micro:bit

Learn to Code · SAM Labs + micro:bit · Clase express · CFIE de León · CC BY-SA 4.0 · Iconos: Font Awesome Free (CC BY 4.0)

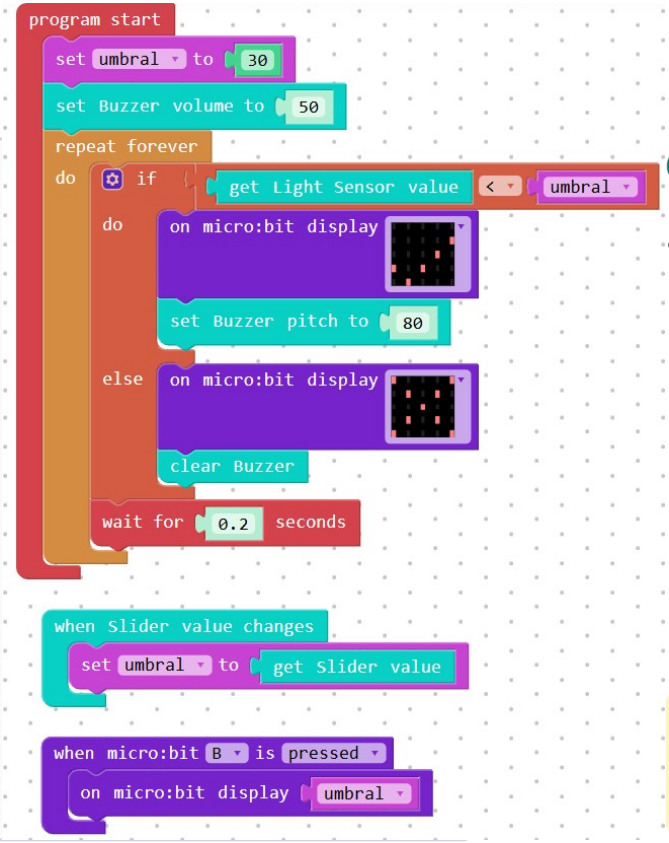
UMBRAL A MEDIDA

Solución · Reto 5

```
program start
  set umbral to (30)
  set Buzzer volume to (50)
  repeat forever
    if (get Light Sensor value) < (umbral)
      do   on micro:bit display [no]
           set Buzzer pitch to (80)
      else on micro:bit display [yes]
           clear Buzzer
    wait for (0.2) seconds

when Slider value changes
  set umbral to (get Slider value)

when micro:bit [B] is [pressed]
  on micro:bit display (umbral)
```



Claves

- Si el bloque de texto no acepta el número directamente, usa create text with o el bloque de conversión a texto.

Si te sobra tiempo: Usa el botón A para «capturar» la luz actual como umbral, en vez del deslizador.

Learn to Code · SAM Labs + micro:bit · Clase express · CFIE de León · CC BY-SA 4.0 · Iconos: Font Awesome Free (CC BY 4.0)



Descanso · 10 minutos

Los kits con poca batería, a cargar.

Learn to Code · SAM Labs + micro:bit · Clase express · CFIE de León · CC BY-SA 4.0 · Iconos: Font Awesome Free (CC BY 4.0)

1:40

Mini-proyecto

«Farola inteligente»: ahora sois vuestro alumnado



MINI-PROYECTO · 30 MIN

Farola inteligente

La misión

- Se enciende sola al oscurecer: la pantalla de la micro:bit es la luz
- El umbral se regula con el deslizador
- Un pitido corto al encenderse, solo una vez
- Extra: modo manual con el botón A
- 10 min construir + 20 min programar



Equipos de 2 o 3: cada uno con un rol. Al final, demostración de todos a la vez.

Learn to Code · SAM Labs + micro:bit · Clase express · CFIE de León · CC BY-SA 4.0 · Iconos: Font Awesome Free (CC BY 4.0)

MINI-PROYECTO

Solución · Farola inteligente

```
program start
  set umbral to (30)
  set encendida to (false)
  set Buzzer volume to (50)
  repeat forever
    if ((get Light Sensor value) < (umbral)) and (encendida = false)
      do   on micro:bit display [square]      ← «luz» de la farola
           set Buzzer pitch to (60)
           wait for (0.3) seconds
           clear Buzzer
           set encendida to (true)
    else if ((get Light Sensor value) >= (umbral)) and (encendida = true
)
      do   on micro:bit display [small square]  ← farola apagada
           set encendida to (false)
    wait for (0.2) seconds

when Slider value changes
  set umbral to (get Slider value)
```

Claves

- Usa el icono que más LED encienda como «farola encendida».
- Montaje sugerido: un tubo de cartón como poste, la micro:bit arriba mirando hacia abajo y el sensor de luz en la parte alta, sin que le dé la luz de la propia micro:bit.
- Si el sensor «ve» la luz de la micro:bit, la farola parpadea. Es un buen problema de diseño para el alumnado.

Learn to Code · SAM Labs + micro:bit · Clase express · CFIE de León · CC BY-SA 4.0 · Iconos: Font Awesome Free (CC BY 4.0)

2:10

Llevarlo a mi aula

Situación de aprendizaje modelo y la tuya · 40 min



Learn to Code · SAM Labs + micro:bit · Clase express · CFIE de León · CC BY-SA 4.0 · Iconos: Font Awesome Free (CC BY 4.0)

SITUACIÓN DE APRENDIZAJE MODELO

Ahorro energético en el instituto

«En el instituto hay luces encendidas en pasillos vacíos y a pleno día. ¿Podéis diseñar un sistema que encienda la luz solo cuando haga falta?»

Curso

1.º–2.º de ESO

Producto

Prototipo programado (farola o lámpara) e informe con los datos de luz medidos, presentado al equipo directivo

Áreas

Tecnología y Digitalización (principal), Matemáticas, Física y Química (2.º)

- 6 sesiones de 55 min · Equipos de 3 con roles rotativos (1 kit por equipo)
- Competencias: STEM1–STEM3 · CD5 · CPSAA3 · CE3

Learn to Code · SAM Labs + micro:bit · Clase express · CFIE de León · CC BY-SA 4.0 · Iconos: Font Awesome Free (CC BY 4.0)

SECUENCIA

6 sesiones: del reto a la presentación

1

Activación

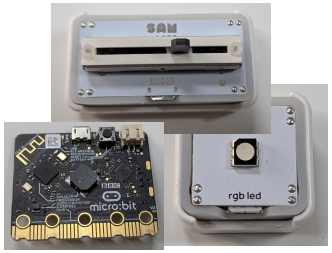
Recorrido por el centro: ¿dónde se malgasta luz? Algoritmo unplugged de una farola.



2

Exploración

Kit y retos 1–3: secuencia, eventos, entrada → salida.



3

Investigación

Medimos la luz con el sensor en aulas y pasillos a distintas horas. Tabla y gráfico.



4

Programación

Retos 4–5: alarma de oscuridad y umbral con una variable, decidido con los datos.

5

Construcción

Prototipo en cartón, programa final, pruebas y mejoras.



6

Comunicación

Presentación al equipo directivo con estimación del ahorro. Coevaluación.



Learn to Code · SAM Labs + micro:bit · Clase express · CFIE de León · CC BY-SA 4.0 · Iconos: Font Awesome Free (CC BY 4.0)

GESTIÓN DEL AULA

Que funcione en tu aula



Cargar la víspera

Bloques en carga y pilas nuevas en las micro:bit.



Flashear antes

Las micro:bit, con el hex de SAM ya cargado.



Numerar kits

Bloques, micro:bit y Chromebook con el mismo número.



Roles

Programador/a, ingeniero/a, científico/a de datos y portavoz.

Learn to Code · SAM Labs + micro:bit · Clase express · CFIE de León · CC BY-SA 4.0 · Iconos: Font Awesome Free (CC BY 4.0)

PREGUNTA GUÍA

¿Qué podrías medir, avisar o mover en tu materia?

Música

Theremin e instrumentos con el deslizador

Ed. Física

Cronómetros y señales con la micro:bit

Lengua

Historias interactivas con luz y sonido

Física y Química

Medir la luz y la energía

Matemáticas

Datos, umbrales y gráficos

Geografía

Estaciones meteorológicas sencillas

Learn to Code · SAM Labs + micro:bit · Clase express · CFIE de León · CC BY-SA 4.0 · Iconos: Font Awesome Free (CC BY 4.0)

TU SITUACIÓN DE APRENDIZAJE

Tu turno: 30 minutos

10 min Pensar

- Curso, área y reto real
- Producto final
- Actividad central: entradas → decisión → salidas

20 min Probar

- Programa la actividad central con el kit
- Anota qué falla y cómo lo arreglas
- Prepara 30 segundos para contarlo

Learn to Code · SAM Labs + micro:bit · Clase express · CFIE de León · CC BY-SA 4.0 · Iconos: Font Awesome Free (CC BY 4.0)

RECURSOS

Para seguir aprendiendo



Portal Learn to Code en castellano

samlabseurope.com: unidades de Secundaria con guía docente, diapositivas y ficha



CATEDU · micro:bit en el aula

Curso en castellano con situaciones de aprendizaje (CC BY-NC-SA)



Ayuda SAM Labs

support.samlabs.com · Bluetooth en Chromebook: go.samlabs.com/sambluetoothhelp



Materiales de este curso

Fichas, soluciones y plantilla · CC BY-SA 4.0 · Iconos: Font Awesome Free (CC BY 4.0)

¡Gracias!

Cuéntanos tu idea en 30 segundos y rellena el cuestionario.



Antes de irte: apaga, cuenta y guarda el kit.

Learn to Code · SAM Labs + micro:bit · Clase express · CFIE de León · CC BY-SA 4.0 · Iconos: Font Awesome Free (CC BY 4.0)