

<u>N.º PRÁCTICA</u>	<u>NOMBRE PRÁCTICA</u>
1	<u>Práctica 1: encender 1 led.</u>
2	<u>Práctica 2: encender 2 LED</u>
3	<u>Práctica 3: encender 2 LED 2</u>
4	<u>Práctica 4: encender 3 LED</u>
5	<u>Práctica 5: encender 3 leds como si fuera un semáforo</u>
6	<u>Práctica 6: encender los 5 leds</u>
7	<u>Práctica 7: encender los 5 LED, de forma que hagan un barrido</u>
8	<u>Práctica 8: práctica ldr,</u>
9	<u>Práctica 9: práctica servo.</u>
10	<u>Práctica 10: dos pulsadores</u>
11	<u>Práctica 11: un pulsador</u>
12	<u>Un pulsador 2</u>
13	<u>Un pulsador 3</u>
14	<u>Práctica 14: temperatura.</u>
15	<u>Práctica 15: mover un servo con potenciómetro.</u>
16	<u>Práctica 16: melodía</u>
17	<u>Práctica 17: relé.</u>
18	<u>Relé 2</u>
19	<u>Relé 3</u>
20	<u>Relé 4</u>
21	<u>Práctica 21: sensor distancia</u>
22	<u>Práctica 22: bluetooth encender un led.</u>
23	<u>Práctica 23: bluetooth y relé</u>
24	<u>Práctica 24: bluetooth -servo</u>
25	<u>Práctica 25: LCD</u>

Eduardo de la Torre

26	Práctica 26: LCD y LDR sensor de luz.
27	Práctica 27: LCD y pulsadores.
28	Práctica 28: Tarea LCD y sensor de temperatura.
29	Práctica 29: Tarea LCD y sensor de distancia.
30	Práctica 30. LCD, potenciómetro y servo.
31	Práctica 31. Tarea LCD, bluetooth y led.
32	Práctica 32: LCD y servo.
33	Práctica 33: Tarea controlador de motores. Coche.
34	Práctica 34: control del coche por bluetooth.
35	Práctica 35: Servo con deslizador app bluetooth
36	Práctica 36: ascensor servo.
37	Práctica 37: ventilador.
38	Prácticas 38: Tareas Tinkercard
39	Práctica 39: Crear app para manejar arduino con el móvil.
Microbit	
40	Dibujo con la extensión tortuga.
41	Dibujar tu nombre en la placa junto con dos iconos.
42	Agitar la placa micro bit
43	Botones
44	(LDR)
45	Temperatura
46	Simular el funcionamiento de 3 bombas
47	Sensor de voz,
48	brújula
49	juego
50	servomotor
51	Notas musicales tocando bananas o otra fruta.
52	Práctica para medir distancia mediante Maqueen.
53	LED de colores con Maqueen
54	Práctica para que Maqueen se mueva cuando le de cierta cantidad de luz.
55	Práctica para que Maqueen se mueva esquivando obstáculos.
56	Práctica para que Maqueen mueva un

Eduardo de la Torre

	<u>servomotor</u>
57	<u>Práctica para que Maqueen se mueva siguiendo una línea negra.</u>
58	<u>Práctica para que Maqueen se mueva siguiendo códigos de infrarrojos para televisor LG (opción 2 en la app).</u>
59	<u>Práctica para que Maqueen lea las señales de infrarrojo que se le manda.</u>
60	<u>Código morse</u>
61	<u>Semáforo hombre -mujer</u>
62	<u>Sensor de aparcamiento</u>
63	<u>Solicitar ayuda y encender lámpara con sensor de ldr a distancia (por radio)</u>

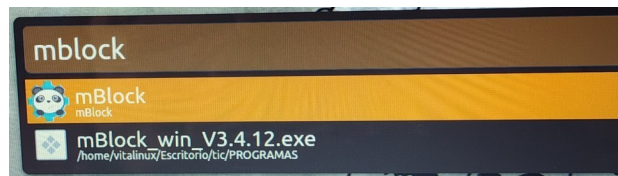
Eduardo de la Torre

Tutorial uso Mblock y arduino ide.

Paso 1: encendemos el ordenador y pulsamos a la vez en el teclado las teclas **control y espacio**.



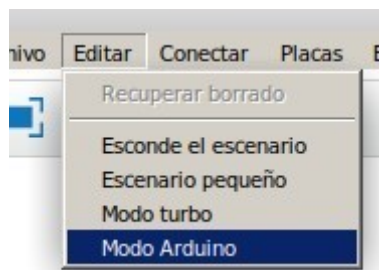
Paso 2: escribimos en la barra de buscador rápido que aparece la palabra **Mblock** para abrir dicho programa y pulsamos intro.



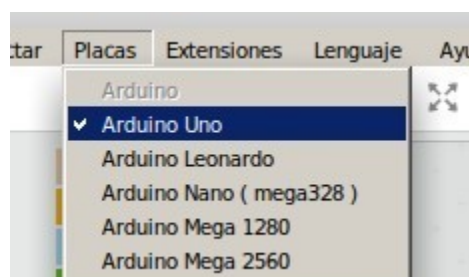
Paso 3: ponemos el idioma en español (picando en **language...español**) si no lo está.



Paso 4: ponemos si no lo está el programa en modo arduino picando en... **Editar Modo arduino**.



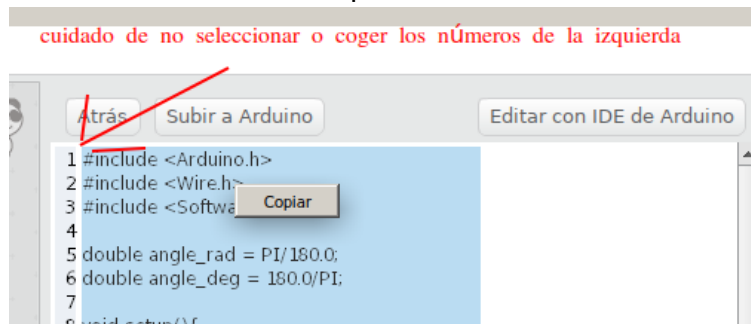
Paso 5: seleccionamos la placa arduino picando en... **Placas Arduino Uno**.



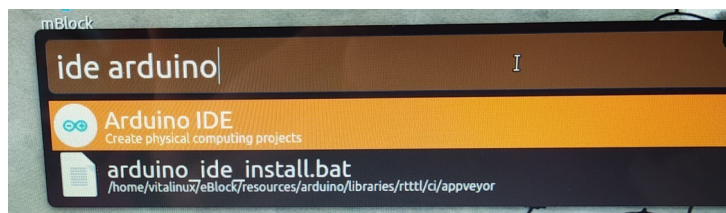
Paso 6. vamos arrastrando los bloques del programa a la zona de bloques, para ello debemos observar el color que tiene, (Para poder poner lo bloques naranjas hay que crear antes variables picando en....Datos y bloques -----Crear una variable).



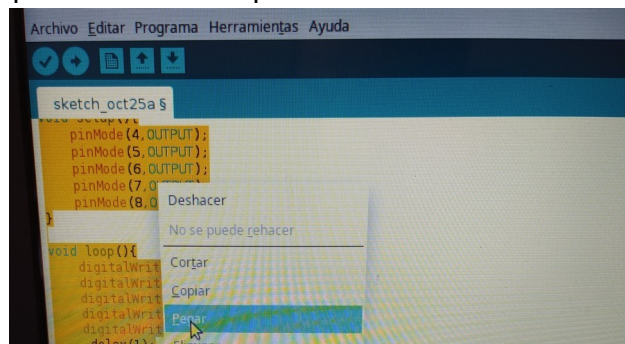
Paso 7: una vez puestos todos los bloques copiamos el código de la ventana de la derecha, teniendo cuidado de no copiar los números de las líneas.



Paso 8: pulsamos como al inicio las teclas **control y espacio** del teclado, escribimos en el buscador **arduino ide** y picamos **Intro** para entrar en el programa.

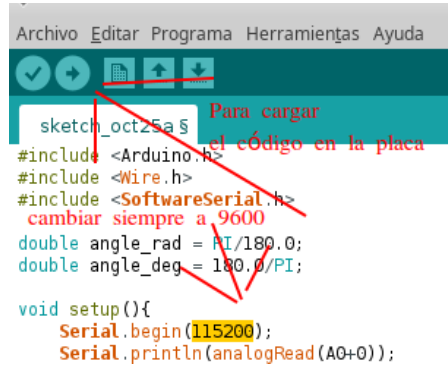


Paso 9: al abrirse el programa Arduino ide, borramos el código que aparece y pegamos el nuevo que habíamos copiado.



Eduardo de la Torre

Paso 10: cambiar la velocidad de comunicación con el puerto serie a **9600** (en el caso de que aparezca “según la práctica”) y picar sobre el icono de la flecha para cargar el código en la placa arduino.



The screenshot shows the Arduino IDE interface. The menu bar includes 'Archivo', 'Editar', 'Programa', 'Herramientas', and 'Ayuda'. Below the menu bar is a toolbar with several icons. A red arrow points from a text box to the upload button (a right-pointing arrow). The code editor displays the following code:

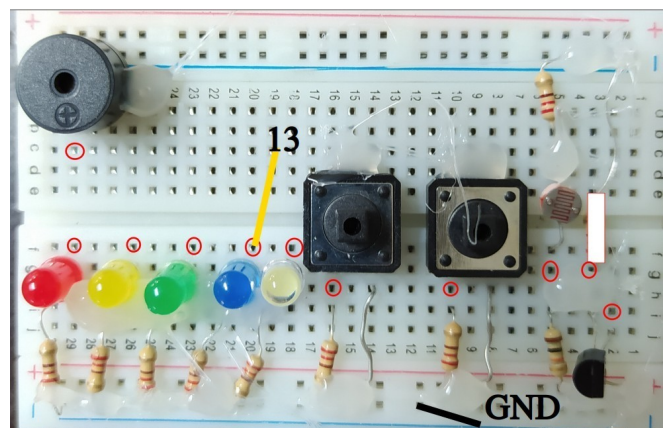
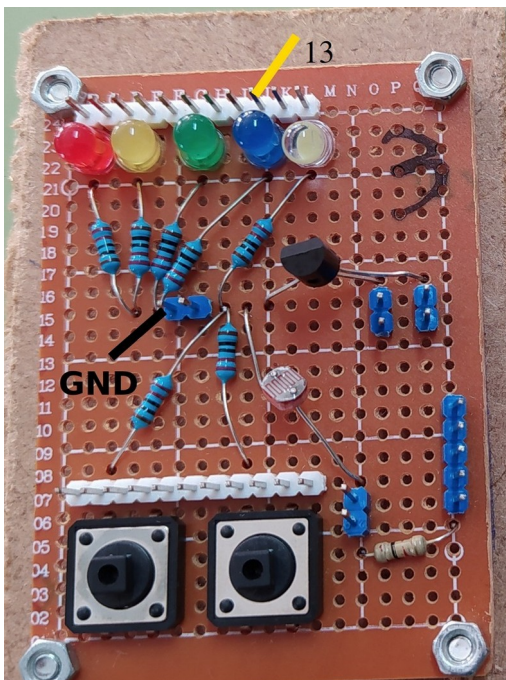
```
sketch_oct25a $
#include <Arduino.h>
#include <Wire.h>
#include <SoftwareSerial.h>
  cambiar siempre a 9600
double angle_rad = PI/180.0;
double angle_deg = 180.0/PI;

void setup(){
  Serial.begin(115200);
  Serial.println(analogRead(A0+0));
```

Práctica 1: encender 1 led.

Vamos a hacer que parpadee el led 13 cada 1 segundos.

```
Programa de Arduino
por siempre
  fijar salida pin digital 13 a ALTO
  esperar 1 segundos
  fijar salida pin digital 13 a BAJO
  esperar 1 segundos
```



Práctica 2: encender 2 LED, alternativamente, cuando te he encendido uno está pagado el otro.

Práctica 3: encender 2 LED, ambos deben encenderse y luego apagarse cada medio segundo.

Práctica 4: encender 3 LED, rojo, amarillo y verde, como si fuera un semáforo, primero se enciende el verde 2 segundos, luego al amarillo un segundo y por último rojo 2 segundos.

Práctica 5: encender 3 leds como si fuera un semáforo, pero en este caso, se enciende el verde 2 segundos, luego el amarillo parpadea tres veces cada medio segundo y por último el rojo se enciende dos segundos.

Práctica 6: encender los 5 leds, hacer que parpadee cada segundos.

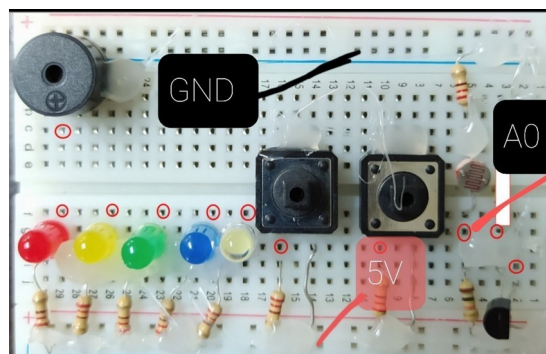
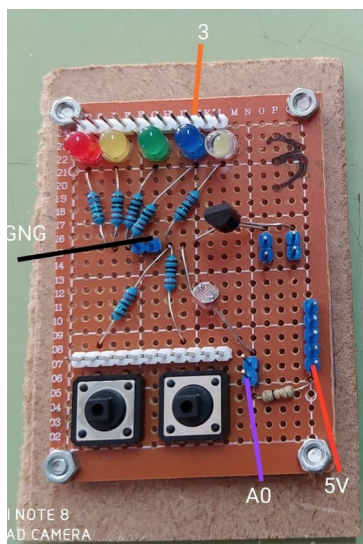
Práctica 7: encender los 5 LED, de forma que hagan un barrido de izquierda a derecha y de derecha izquierda, es decir, se enciende el rojo un segundo, luego el amarillo un segundo, luego el verde un segundo, luego el azul un segundo, luego el blanco un segundo, luego el azul un segundo, luego el verde un segundo y luego al amarillo un segundo.

Práctica 8: práctica ldr, sensor de luz o resistencia variable con la luminosidad.

Esta práctica se basa en encender un led cuando la luz que incide sobre la Ldr, supera un cierto nivel, debes ajustar ese nivel y función de la luminosidad de la clase.

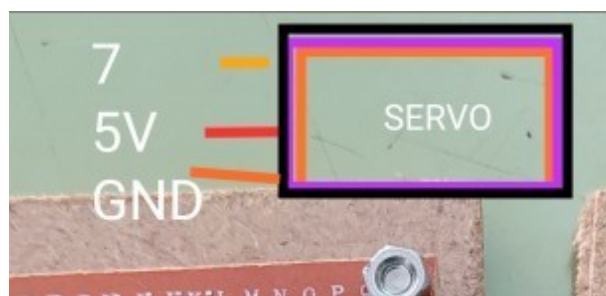
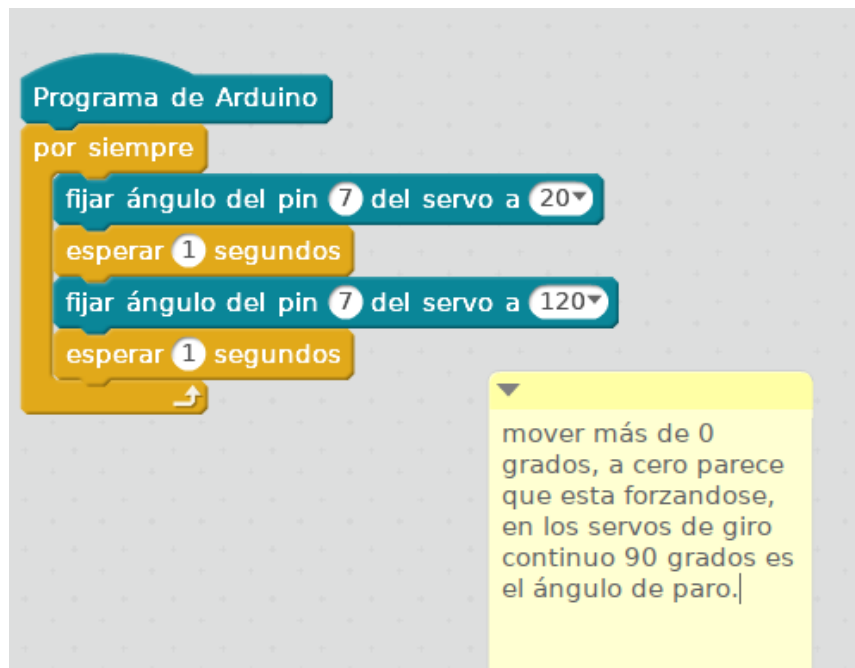
El LED tiene que encenderse cuando el valor de la señal A0 supere 300.

```
Programa de Arduino
por siempre
  escribir en el serial el texto leer pin analógico (A) 0
  si leer pin analógico (A) 0 > 500 entonces
    fijar salida pin digital 3 a ALTO
  si no
    fijar salida pin digital 3 a BAJO
```



Práctica 9: práctica servo.

Basándose en las imágenes , prueba el funcionamiento de un servo y cambia luego los datos para que el desplazamiento del servo vaya de 10 grados a 110 grados.

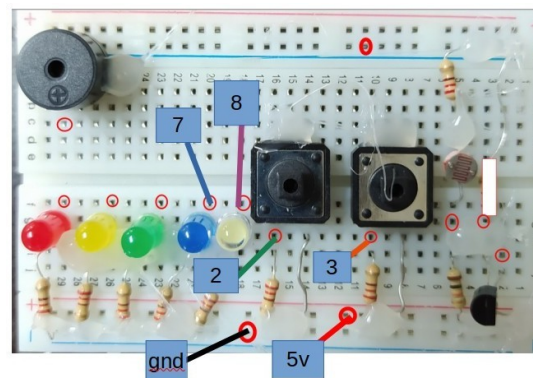
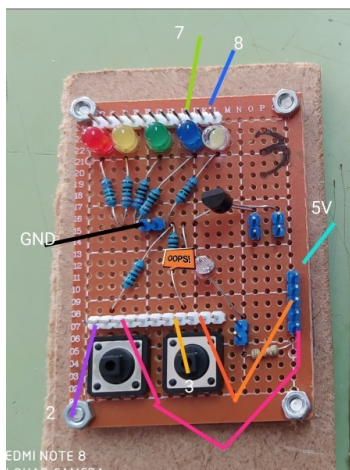


Práctica 10: dos pulsadores

Aquí tenéis que subirme captura del encendido de 2 LED con dos pulsadores, fijaros en la imagen que adjunto para saber cómo hacer el montaje en la placa.

```
Programa de Arduino
por siempre
  si leer pin digital 2 = 1 entonces
    fijar salida pin digital 7 a ALTO
  si no
    fijar salida pin digital 7 a BAJO
fin

Programa de Arduino
por siempre
  si leer pin digital 3 = 1 entonces
    fijar salida pin digital 8 a ALTO
  si no
    fijar salida pin digital 8 a BAJO
fin
```



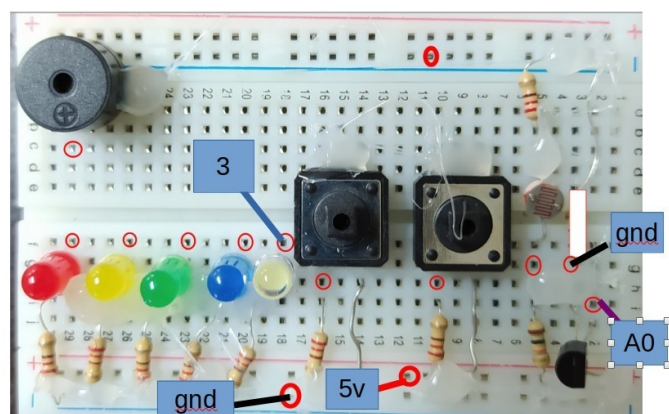
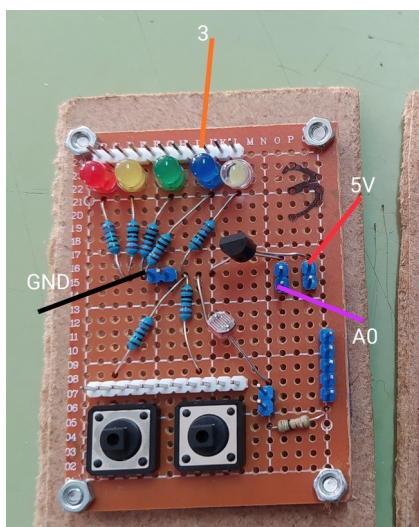
Práctica 11: Hacer una práctica en la que al pulsar un pulsador se enciendan tres leds al mismo tiempo.

Práctica 12: hacer una práctica en la que al pulsar un pulsador parpadeen dos leds cada segundo.

Práctica 13: Hacer una práctica en la que dos leds estén siempre parpadeando cada segundo hasta que presionamos un pulsador momento en el que se apagan

Práctica 14: temperatura.

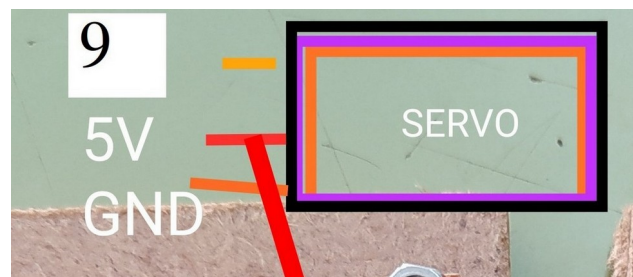
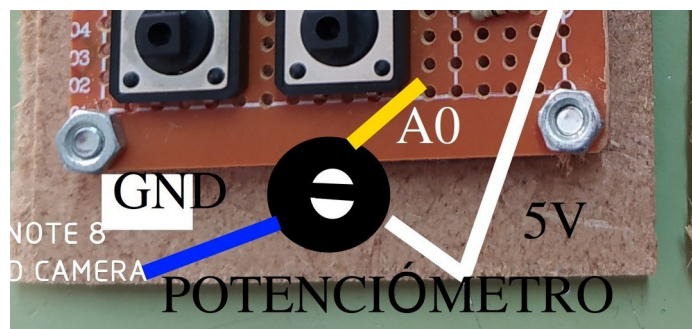
Aquí tenéis la programación del encendido de un LED (3) si la temperatura supera los 25 grados centígrados, dependiendo de la temperatura de la clase tendréis que cambiar la temperatura de corte, usar el terminal del puerto serie para ver como cambia el valor de la temperatura.



Práctica 15: mover un servo con potenciómetro.

Aquí debéis manejar 1 servo con un potenciómetro, con la entrada analógica A0. Hemos visto práctica de servo y de sensores que producen variación la entrada analógica A0, habría que mezclar ambas prácticas.

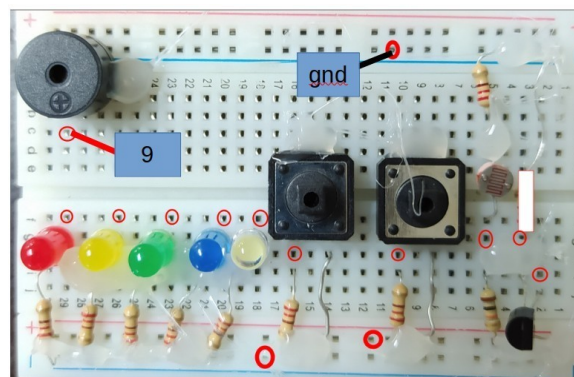
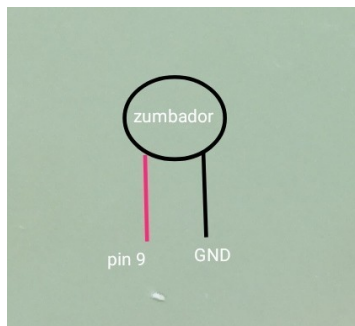
```
Programa de Arduino
por siempre
  escribir en el serial el texto leer pin analógico (A) 0
  fijar ángulo del pin 9 del servo a leer pin analógico (A) 0 / 6
```



Práctica 16: melodía

Tarea que consiste en sacar una melodía con un zumbador

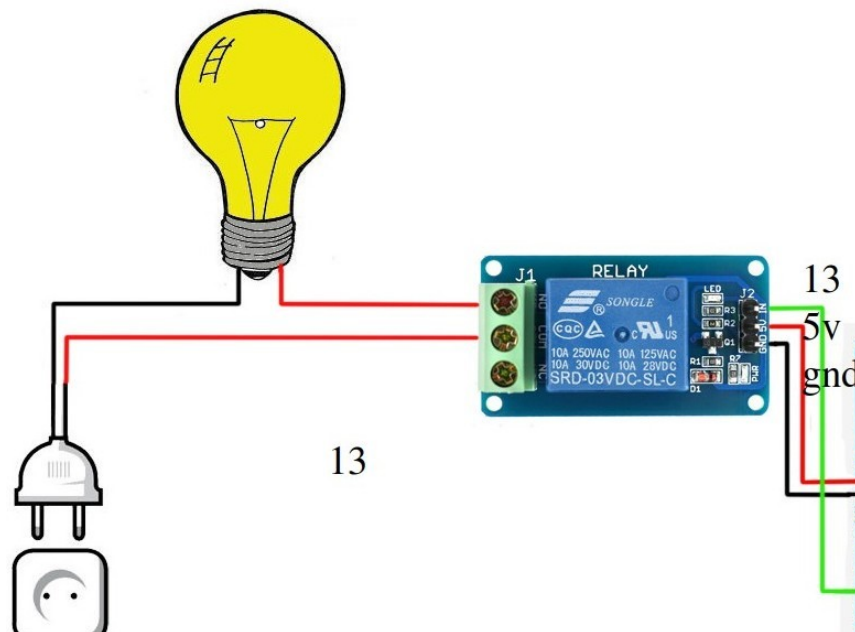
```
Programa de Arduino
fijar tiempo a 120
por siempre
  reproducir tono 9 en nota C7 beat tiempo * 1
  reproducir tono 9 en nota E6 beat tiempo * 0.25
  reproducir tono 9 en nota A6 beat tiempo * 0.25
  reproducir tono 9 en nota B6 beat tiempo * 0.25
  reproducir tono 9 en nota C7 beat tiempo * 0.25
  reproducir tono 9 en nota D7 beat tiempo * 0.33
  reproducir tono 9 en nota C7 beat tiempo * 0.33
  reproducir tono 9 en nota B7 beat tiempo * 0.33
  reproducir tono 9 en nota B6 beat tiempo * 0.33
  reproducir tono 9 en nota G6 beat tiempo * 0.33
  reproducir tono 9 en nota F6 beat tiempo * 0.75
  reproducir tono 9 en nota B6 beat tiempo * 0.75
  reproducir tono 9 en nota C7 beat tiempo * 1
esperar 1 segundos
```



Práctica 17: relé.

Aquí tenéis que manejar una bombilla con un relé (el relé es un componente que se utiliza para controlar generalmente un circuito de voltaje de 220v con otro de corriente continua de menos voltaje) , debéis hacer que se encienda y apague cada 3 segundos, os mando las conexiones, el código sería el mismo que para un led, pero las conexiones diferentes.

```
Programa de Arduino
por siempre
  fijar salida pin digital 13 a ALTO
  esperar 1 segundos
  fijar salida pin digital 13 a BAJO
  esperar 1 segundos
```



Práctica 18: Haz que la bombilla parpadee cada medio segundo.

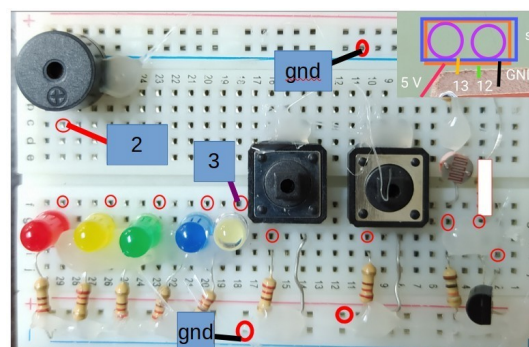
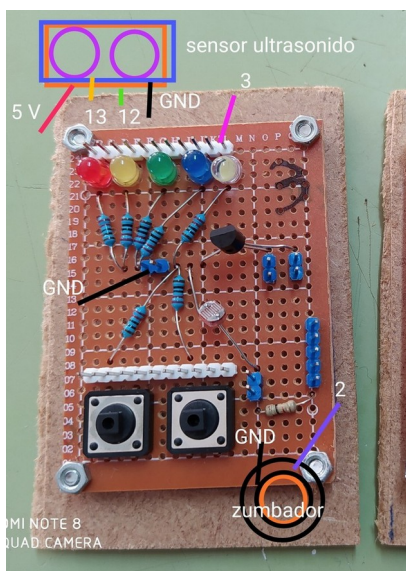
Práctica 19: Haz que la bombilla se encienda cuando pulsemos un pulsador.

Práctica 20: conseguir que la bombilla se encienda cuando la luz que llega a la ldr pase de cierto nivel.

Práctica 21: sensor distancia

El sensor de distancia que vamos a utilizar funciona mediante ultrasonido, manda una señal ultrasónica que rebota y luego vuelve al sensor, el tiempo de rebote se traduce en distancia. Tarea para encender led y zumbador, cuando la distancia sea menor de x cms.

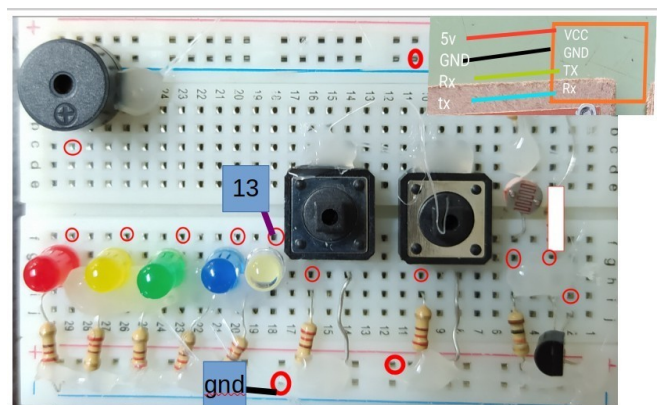
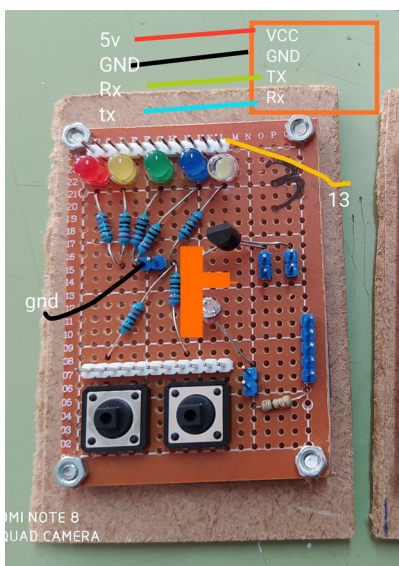
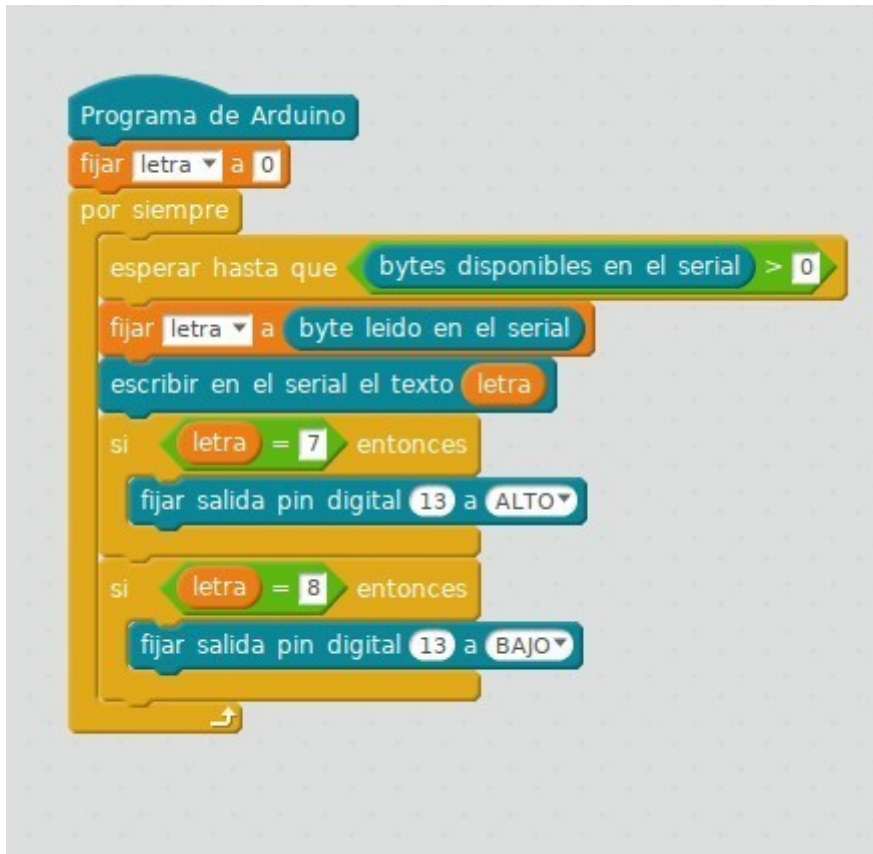
```
Programa de Arduino
por siempre
  escribir en el serial el texto lee el sensor ultrasónico trig pin 13 echo pin 12
  si lee el sensor ultrasónico trig pin 13 echo pin 12 < 50 entonces
    reproducir tono 2 en nota C4 beat Medio
    fijar salida pin digital 3 a ALTO
    esperar 1 segundos
  si no
    fijar salida pin digital 2 a BAJO
    fijar salida pin digital 3 a BAJO
    esperar 1 segundos
```



Práctica 22: bluetooth encender un led.

El bluetooth nos va a permitir comunicarnos mediante dicha vía con el arduino y controlarlo, también podemos representar sus valores en el monitor serial.

Tarea para manejar por bluetooth el arduino encendiendo un LED. (la aplicación para manejar el bluetooth la tenéis en la web).

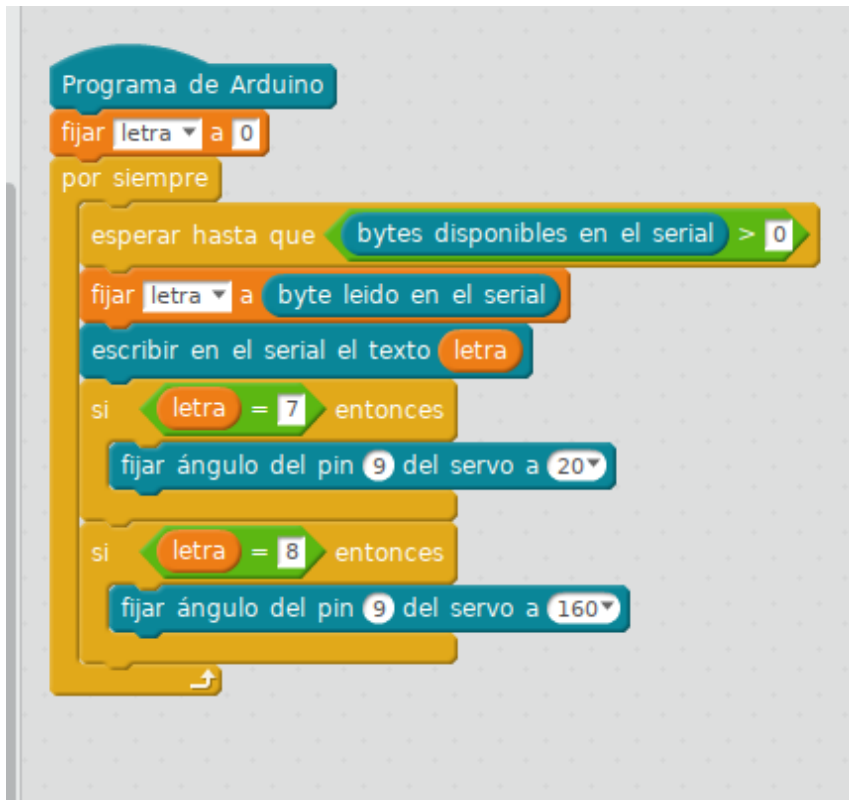


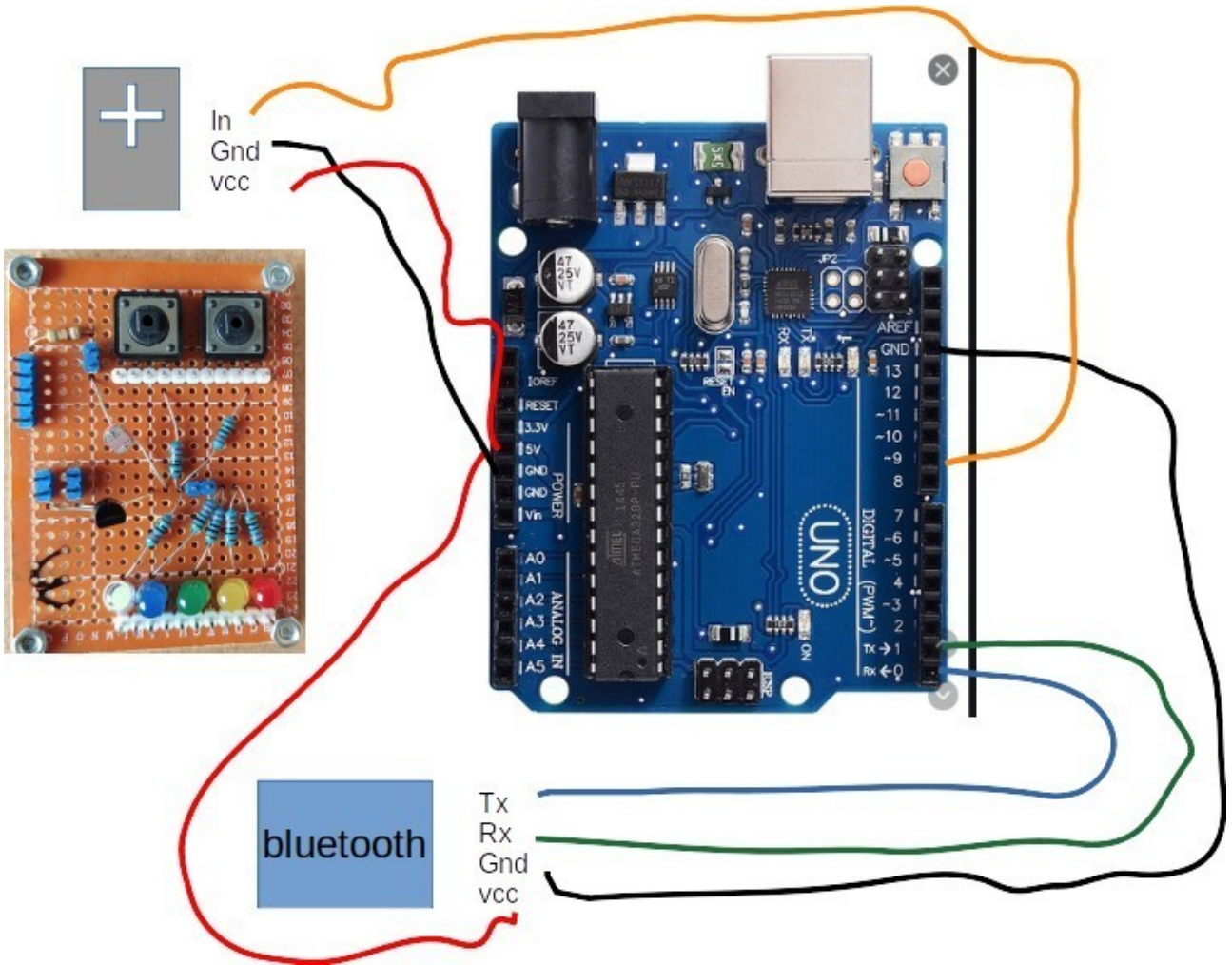
Práctica 23: bluetooth y relé

Debes encender con el móvil la bombilla de la tarea del relé.

Práctica 24: bluetooth -servo

Manejar un servo con el bluetooth de tal forma que picando el botón siete vaya a 20 grados y si picamos en el 8 vaya a 160 º.





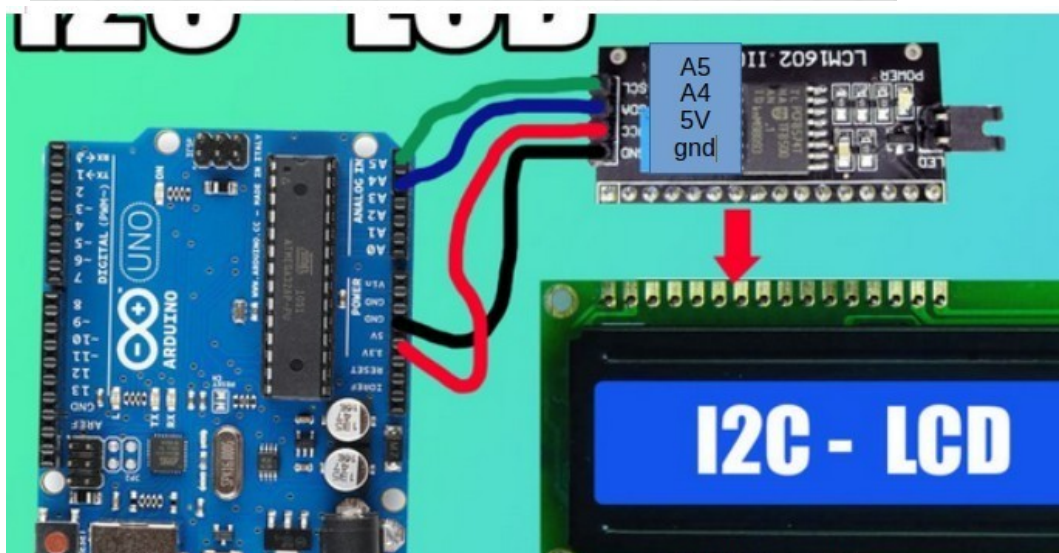
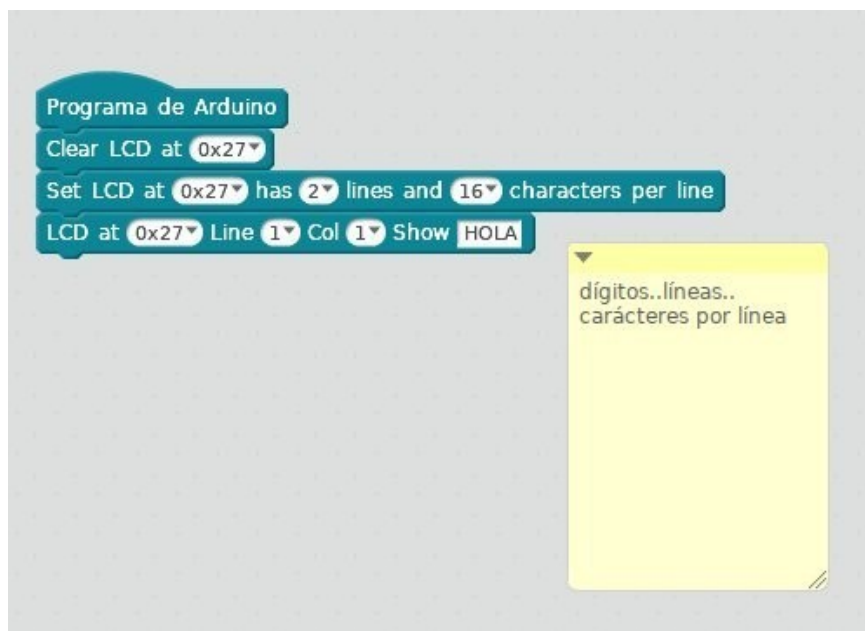
Práctica 25: LCD

Tarea para representar en una pantalla de LCD de 16*2 un texto, para la pantalla utilizaremos el componente o módulo I2C, que permite manejarla con sólo 4 cables, sin él tendríamos que utilizar muchos más y un potenciómetro.

Para utilizar los bloque de la LCD debemos instalar dos librerías si aún no lo está, tienes los archivos en la carpeta tic...programas...LCD.

Entrar dentro de mblock en extensiones... añadir extensión ... cambiar a archivo Zip seleccionar archivo librería mblock.zip

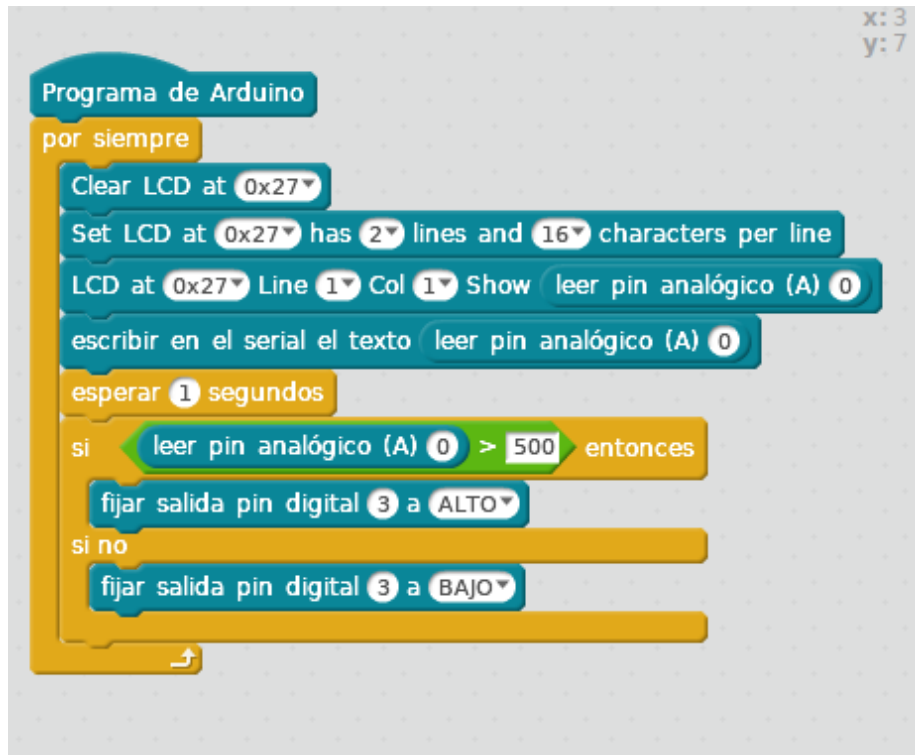
En Arduino añadir extensión seleccionar librería lyquidcrystal2c.zip



Eduardo de la Torre

Práctica 26: LCD y LDR sensor de luz.

Hacer la práctica del sensor de luz encendiendo un led y representar los datos en la LCD. (los datos de entrada de valor A0). También representarlos en el monitor serial.



Práctica 27: LCD y pulsadores.

Hacer la práctica del dos pulsadores y representar el cambio de estado 0 ...1 en la LCD o poner un texto de pulsado sin pulsar.

```
Programa de Arduino
por siempre
  Clear LCD at 0x27
  Set LCD at 0x27 has 2 lines and 16 characters per line
  LCD at 0x27 Line 1 Col 1 Show leer pin digital 7
  LCD at 0x27 Line 1 Col 2 Show leer pin digital 8
  escribir en el serial el texto leer pin digital 7
  escribir en el serial el texto leer pin digital 8
  esperar 0.5 segundos
  si leer pin digital 7 = 1 entonces
    fijar salida pin digital 2 a ALTO
  si no
    fijar salida pin digital 2 a BAJO
  si leer pin digital 8 = 1 entonces
    fijar salida pin digital 3 a ALTO
  si no
    fijar salida pin digital 3 a BAJO
```

The image shows a Scratch-style code editor for an Arduino program. The code is contained within a 'Programa de Arduino' block and is enclosed in a 'por siempre' (forever) loop. The code performs the following actions in sequence: 1. Clear the LCD at address 0x27. 2. Set the LCD at address 0x27 to have 2 lines and 16 characters per line. 3. Display 'leer pin digital 7' on the LCD at Line 1, Column 1. 4. Display 'leer pin digital 8' on the LCD at Line 1, Column 2. 5. Write 'leer pin digital 7' to the serial monitor. 6. Write 'leer pin digital 8' to the serial monitor. 7. Wait for 0.5 seconds. 8. Check if 'leer pin digital 7' is equal to 1. If yes, set digital pin 2 to HIGH (ALTO); if no, set it to LOW (BAJO). 9. Check if 'leer pin digital 8' is equal to 1. If yes, set digital pin 3 to HIGH (ALTO); if no, set it to LOW (BAJO). The code ends with a loop arrow.

Práctica 28: Tarea LCD y sensor de temperatura.

Hacer la práctica del sensor de temperatura y representar los valores de temperatura en la LCD.

Práctica 29: Tarea LCD y sensor de distancia.

Hacer la práctica del sensor de distancia y representar los valores de distancia en cm en la LCD.

Práctica 30. LCD, potenciómetro y servo.

Hacer la práctica del servo con potenciómetro y representar los valores en LCD.

Práctica 31. Tarea LCD, bluetooth y led.

Hacer la práctica del bluetooth y encendido de led y representar los valores en LCD. Cambio de estado de led (0 y 1 o texto: encendido y apagado) y entrada monitor serie del bluetooth.

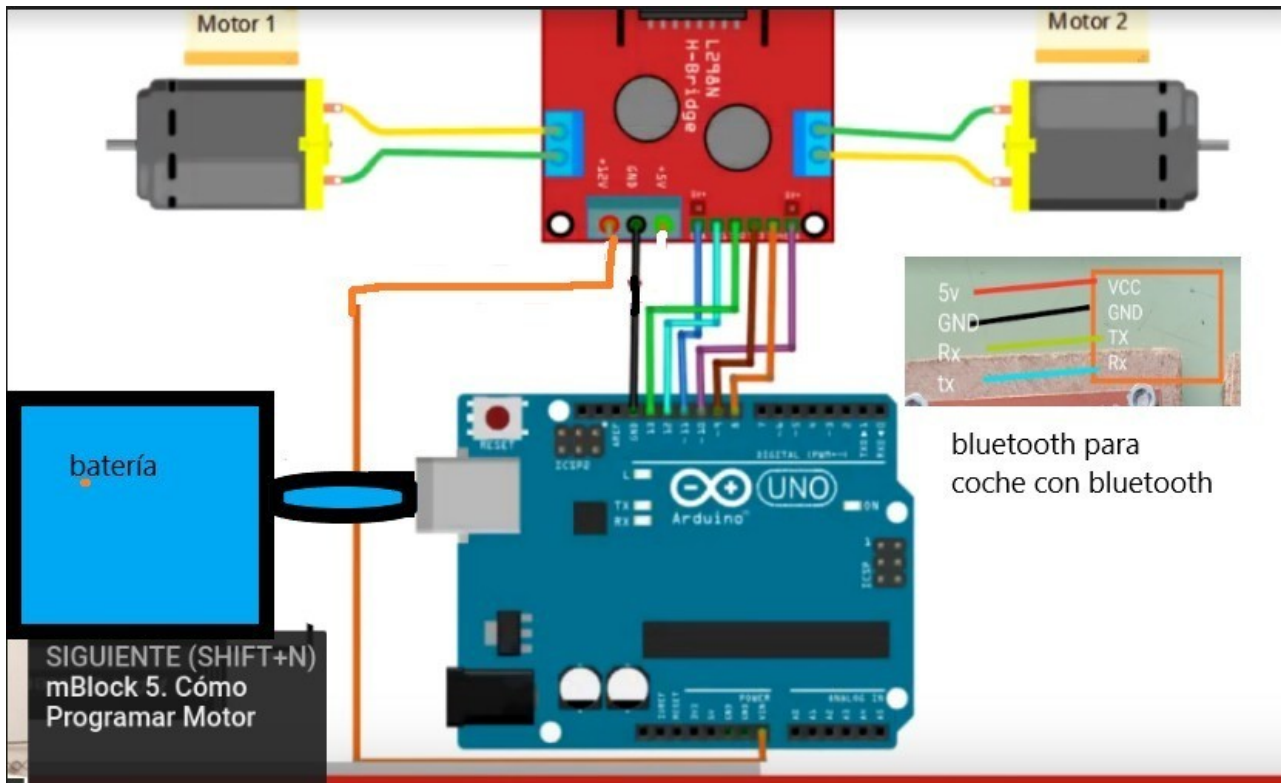
Práctica 32: LCD y servo.

Hacer la práctica del servo (mover por ejemplo de 10 a 110) y representar los datos de los ángulos en la LCD.

Práctica 33: Tarea controlador de motores. Coche.

Programar el funcionamiento de dos motores hacia delante un segundo y luego hacia atrás un segundo.

```
Programa de Arduino
fijar motor1 a 11
fijar IN1 a 12
fijar IN2 a 13
fijar motor2 a 10
fijar IN3 a 9
fijar IN4 a 8
fijar VELOCIDAD a 125
por siempre
  fijar pin PWM motor1 a VELOCIDAD
  fijar salida pin digital IN1 a BAJO
  fijar salida pin digital IN2 a ALTO
  fijar pin PWM motor2 a VELOCIDAD
  fijar salida pin digital IN3 a BAJO
  fijar salida pin digital IN4 a ALTO
  esperar 1 segundos
  fijar pin PWM motor1 a VELOCIDAD
  fijar salida pin digital IN1 a ALTO
  fijar salida pin digital IN2 a BAJO
  fijar pin PWM motor2 a VELOCIDAD
  fijar salida pin digital IN3 a ALTO
  fijar salida pin digital IN4 a BAJO
  esperar 1 segundos
```

Práctica 34: control del coche por bluetooth

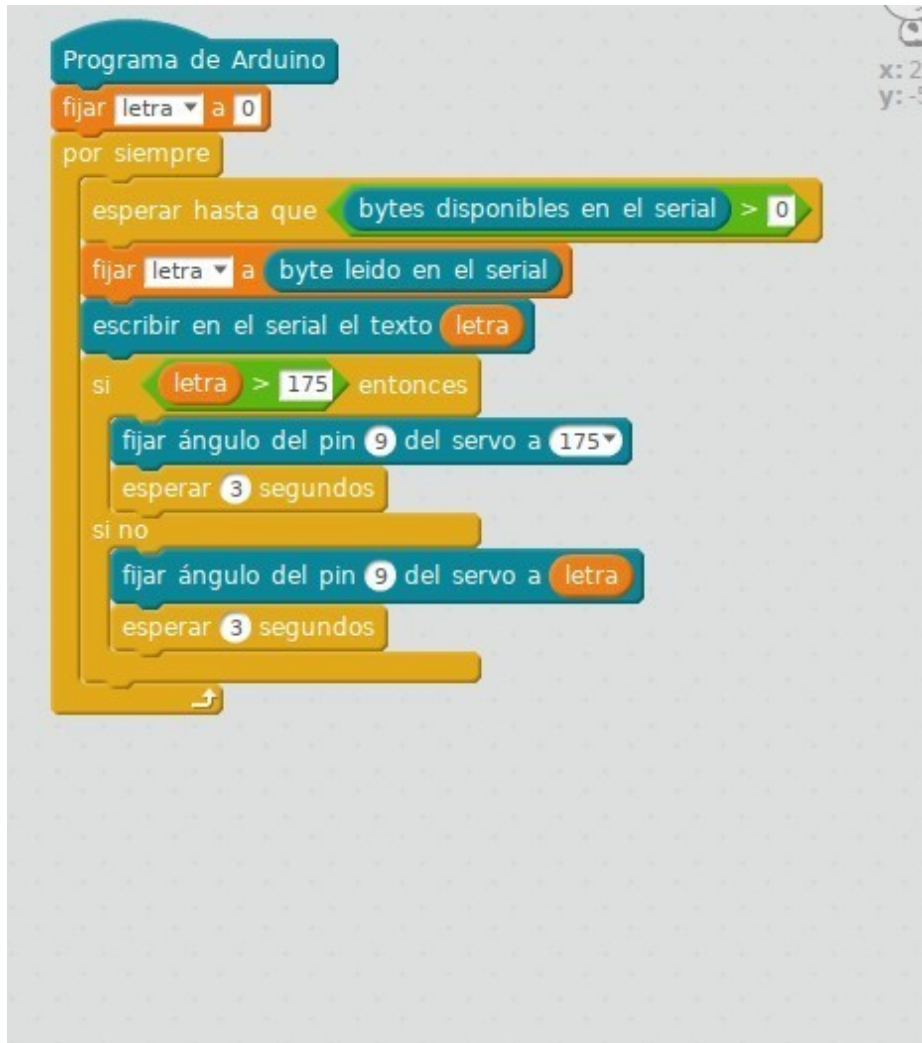
```
Programa de Arduino
fijar letra a 0
fijar motor1 a 11
fijar IN1 a 12
fijar IN2 a 13
fijar motor2 a 10
fijar IN3 a 9
fijar IN4 a 8
fijar VELOCIDAD a 125

por siempre
  esperar hasta que bytes disponibles en el serial > 0
  fijar letra a byte leído en el serial
  escribir en el serial el texto letra
  si letra = 0 entonces
    fijar pin PWM motor1 a VELOCIDAD
    fijar salida pin digital IN1 a BAJO
    fijar salida pin digital IN2 a ALTO
    fijar pin PWM motor2 a VELOCIDAD
    fijar salida pin digital IN3 a BAJO
    fijar salida pin digital IN4 a ALTO
  si letra = 1 entonces
    fijar pin PWM motor1 a VELOCIDAD
    fijar salida pin digital IN1 a ALTO
    fijar salida pin digital IN2 a BAJO
    fijar pin PWM motor2 a VELOCIDAD
    fijar salida pin digital IN3 a ALTO
    fijar salida pin digital IN4 a BAJO
  si letra = 2 entonces
    fijar pin PWM motor1 a VELOCIDAD
    fijar salida pin digital IN1 a BAJO
    fijar salida pin digital IN2 a BAJO
    fijar pin PWM motor2 a VELOCIDAD
    fijar salida pin digital IN3 a BAJO
    fijar salida pin digital IN4 a BAJO
  si letra = 3 entonces
    fijar pin PWM motor1 a VELOCIDAD
    fijar salida pin digital IN1 a BAJO
    fijar salida pin digital IN2 a ALTO
    fijar pin PWM motor2 a VELOCIDAD
    fijar salida pin digital IN3 a ALTO
    fijar salida pin digital IN4 a BAJO
```

Desconectar bluetooth
Antes de cargar y cambiar
9600

Práctica 35: Servo con deslizador app bluetooth

Se trata de desplazar el servo según los datos proporcionados por el deslizador de la aplicación de móvil.



Práctica 36: ascensor servo.

Vamos a mover el servo de giro continuo hacia derecha e izquierda para mover la cabina de un ascensor, pulsando un botón gira hacia la izquierda y pulsando otro a la derecha ,cada uno unos tres segundos, cada vez que pulsamos.



Práctica 37: ventilador.

Vamos a mover el servo de giro continuo hacia la derecha para que gire las aspas de un ventilador, pulsando un botón se inicia el giro y pulsando el otro se para.

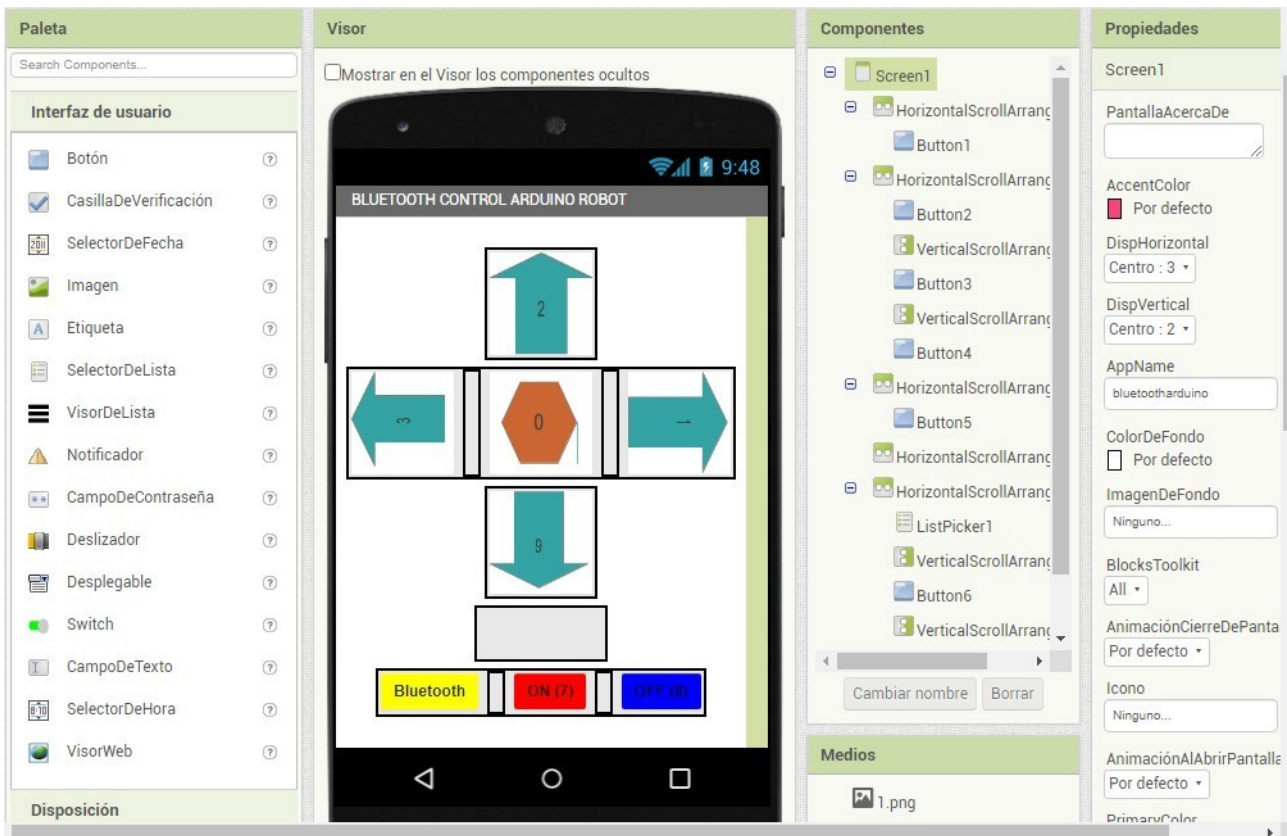


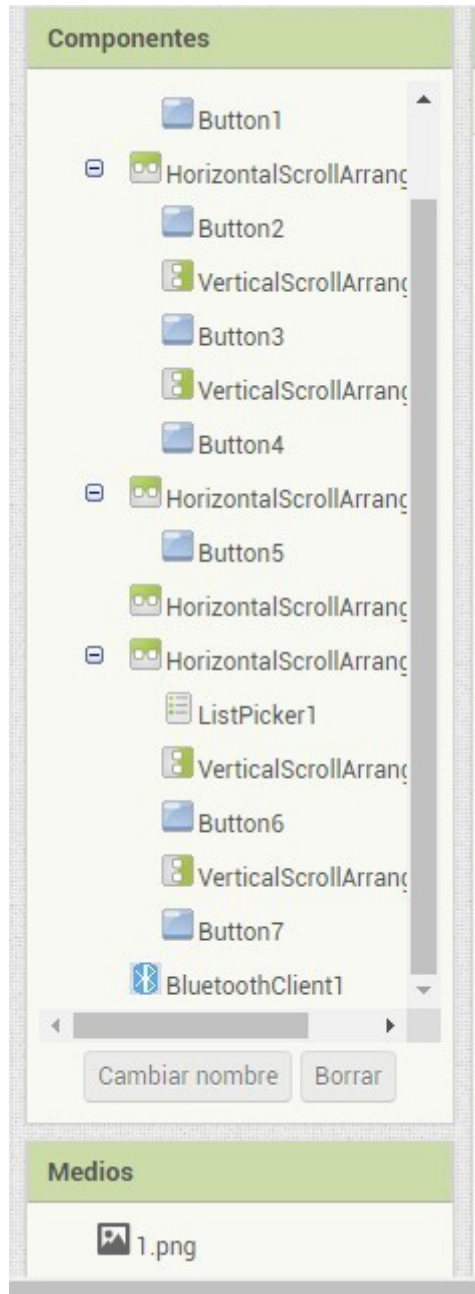
Prácticas 38: Tareas Tinkercard

Dentro de la carpeta tic..programas...tinkercard tenéis vídeos para realizar una serie de prácticas por tinkercard.

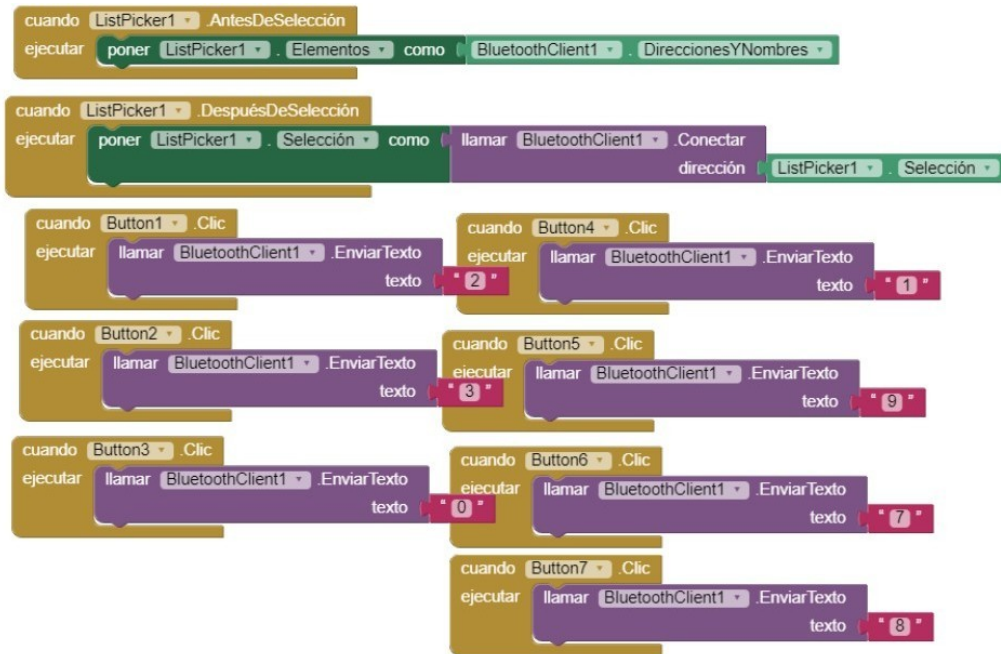
Práctica 39: Crear app para manejar arduino con el móvil.

Se trata de hacer vuestra app para manejar el arduino con app inventor, os dejo capturas de la mía, para que os fijéis y la personalizéis a vuestro gusto.





Eduardo de la Torre



Tutorial prácticas micro bit

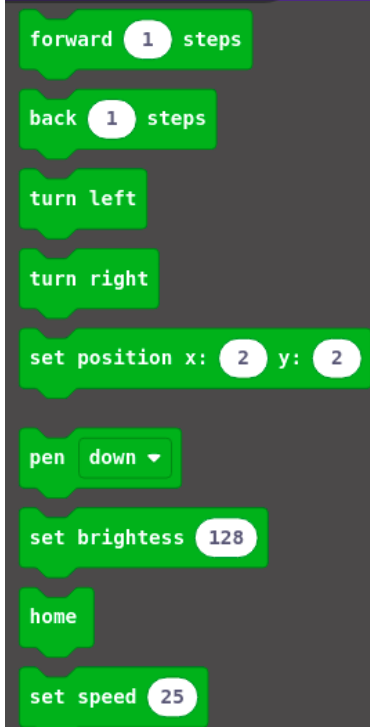
Práctica 40: Dibujo número 1 con la extensión tortuga. picamos en extensiones más extensiones y añadimos la extensión microturtle.



Luego dibujamos un cuadrado lo más grande posible de 5 x 5 píxels.

Resumen de comandos.

Eduardo de la Torre

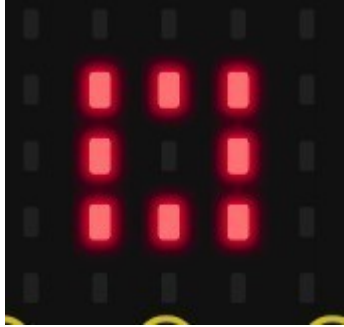
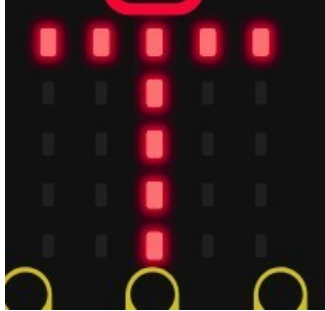
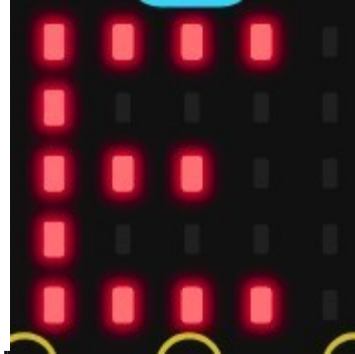
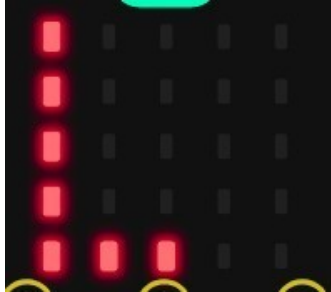
	<p>Hacia delante un paso.</p> <p>Hacia atrás un paso</p> <p>Girar a la izquierda</p> <p>Girar a la derecha</p> <p>Seleccionar la posición x2 y2</p> <p>Bajar lápiz o subir lápiz</p> <p>Seleccionar brillo</p> <p>Volver a casa</p> <p>Seleccionar velocidad.</p>
--	---



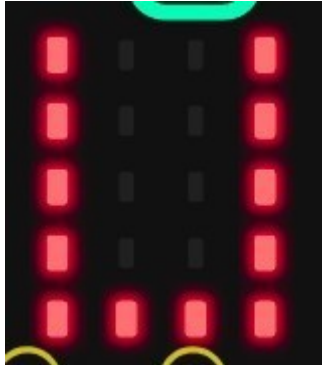
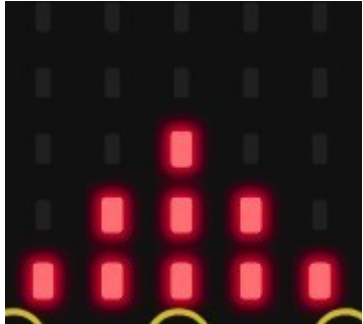
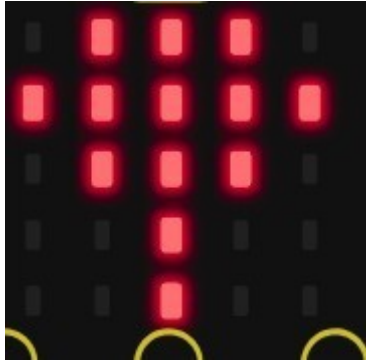
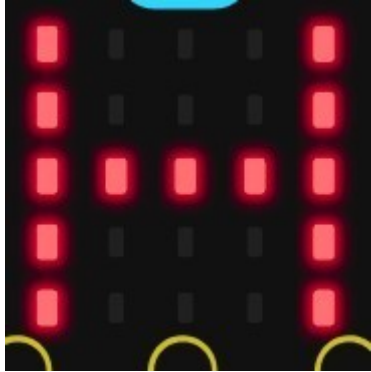
```
al iniciar
forward 2 steps
pen down
turn right
forward 2 steps
turn right
forward 4 steps
turn right
forward 4 steps
turn right
forward 4 steps
turn right
forward 2 steps
```

Eduardo de la Torre

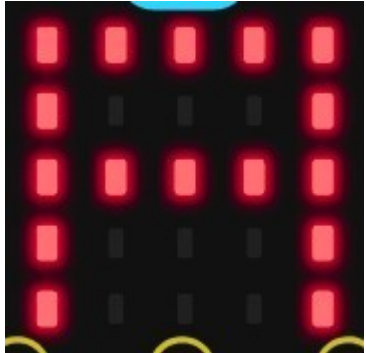
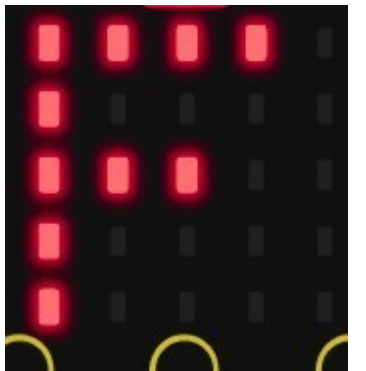
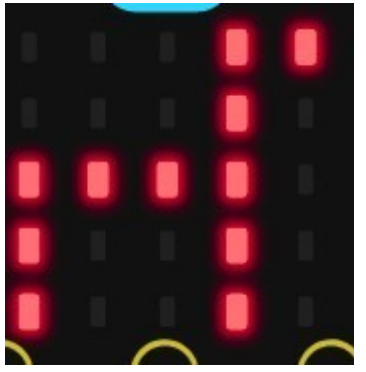
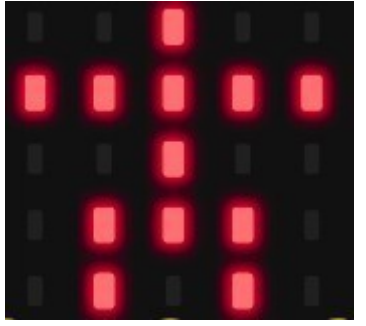
Haz las siguientes prácticas creando el código para el dibujo correspondiente.

Número de práctica	Dibujo.
1	 A 3x3 grid of red squares on a black background. The squares are arranged in three rows and three columns. At the bottom of the grid, there are three yellow semi-circles.
2	 A red shape resembling a 'T' on a grid. The top row consists of five red squares. Below this, there is a vertical column of five red squares. At the bottom of the grid, there are three yellow semi-circles.
3	 A red shape resembling a '3' on a grid. The top row consists of four red squares. The second row has one red square on the left. The third row consists of three red squares. The fourth row consists of four red squares. At the bottom of the grid, there are three yellow semi-circles.
4	 A red shape resembling a '4' on a grid. The first column consists of five red squares. The second column has one red square at the top. The third column has one red square at the top. The fourth column has one red square at the top. At the bottom of the grid, there are three yellow semi-circles.

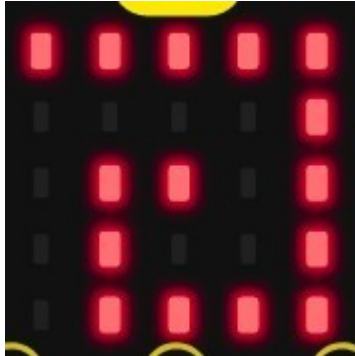
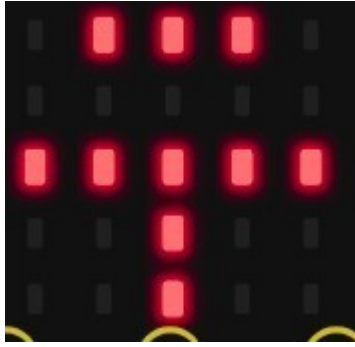
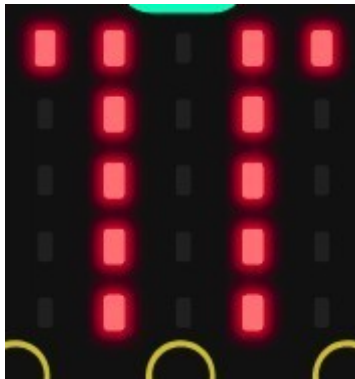
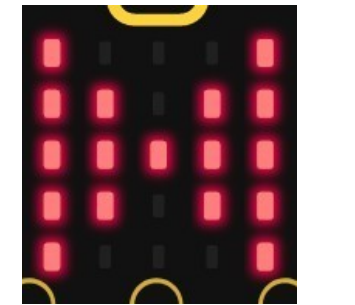
Eduardo de la Torre

5	
6	
7	
8	

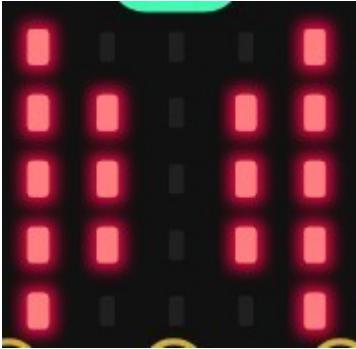
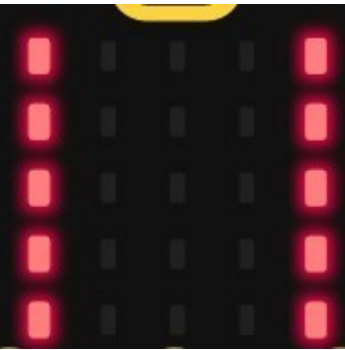
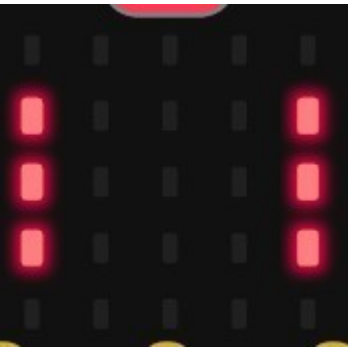
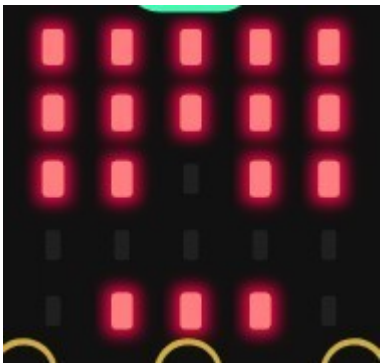
Eduardo de la Torre

9	
10	
11	
12	

Eduardo de la Torre

13	
14	
15	
16	

Eduardo de la Torre

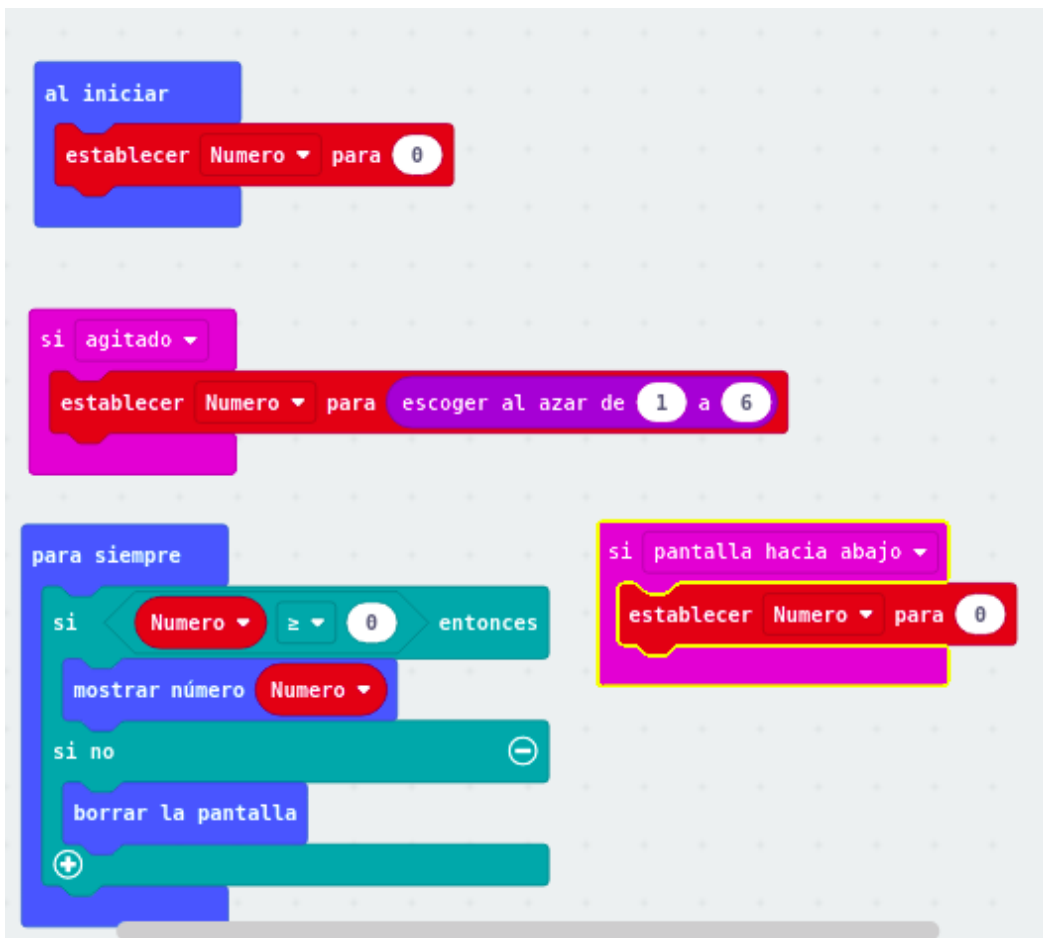
17	
18	
19	
20	

Práctica número 41 dibujar tu nombre en la placa junto con dos iconos.



Práctica número 42, en esta práctica al agitar la placa micro bit queremos conseguir que aparezca un número aleatorio entre 1 y 10, como si fuera un dado, en esta práctica hay que crear una variable picando en el bloque de variables, yo la he llamado número.

Eduardo de la Torre

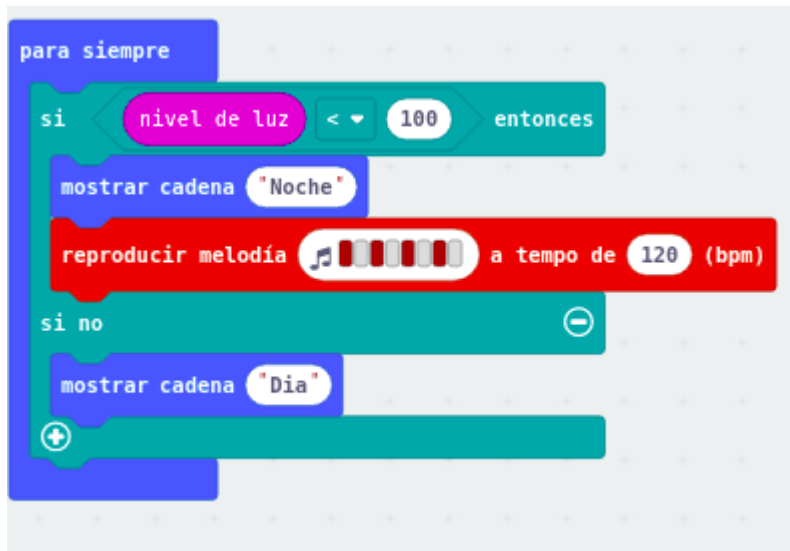


Práctica número 43, en ella al presionar el botón A se mostrará vuestro nombre, luego espera un segundo y sonará una melodía y si presionamos el botón B mostrar uno de vuestros apellidos espera un segundo y sonará otra melodía. Os muestro un código ejemplo, debéis modificarlo.

```
para siempre
  si botón A presionado entonces
    mostrar cadena "A"
    pausa (ms) 1000
    reproducir melodía [melodía A] a tempo de 120 (bpm)
    borrar la pantalla
  +
  si botón B presionado entonces
    mostrar cadena "B"
    pausa (ms) 1000
    reproducir melodía [melodía B] a tempo de 120 (bpm)
    borrar la pantalla
  +
```

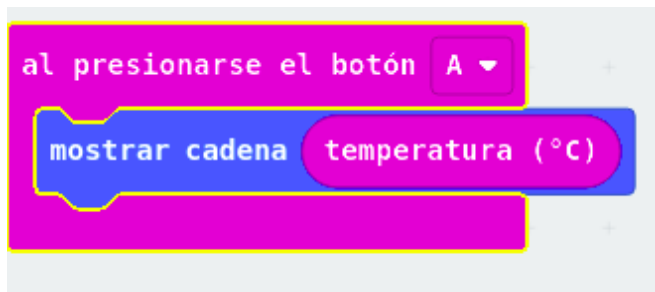
The image shows a Scratch script within a 'para siempre' (forever) loop. It contains two 'si' (if) blocks. The first 'si' block is triggered by 'botón A presionado' and contains: 'mostrar cadena "A"', 'pausa (ms) 1000', 'reproducir melodía' with a melody of 8 notes (red, orange, yellow, green, blue, purple, red, orange) at 120 bpm, and 'borrar la pantalla'. The second 'si' block is triggered by 'botón B presionado' and contains: 'mostrar cadena "B"', 'pausa (ms) 1000', 'reproducir melodía' with a melody of 8 notes (yellow, green, blue, purple, red, orange, yellow, green) at 120 bpm, and 'borrar la pantalla'. There are plus signs between the two 'si' blocks and at the end of the loop.

Práctica número 44 , en esta práctica utilizaremos sensor de luz (LDR) cuando la luz sea inferior a un nivel que nosotros cambiaremos a 50 lúmenes, la pantalla mostrará el icono de una luna (círculo) al mismo tiempo que suena una melodía y cuando es superior mostrará el icono de un sol ,os pongo un código de ejemplo para que lo cambiéis.



```
para siempre
si nivel de luz < 100 entonces
  mostrar cadena "Noche"
  reproducir melodía a tempo de 120 (bpm)
si no
  mostrar cadena "Dia"
```

Práctica número 45 ,en esta práctica vamos a utilizar el sensor de temperatura cuando presione el botón B en lugar de A debe mostrarse en la pantalla la temperatura a la que estamos.



```
al presionarse el botón A
  mostrar cadena temperatura (°C)
```

Práctica número 46, en esta práctica hemos simulado el funcionamiento de 3 bombas de agua representadas cada una de ellas por un LED, la primera bomba está encendida durante 4 segundos, la segunda durante dos y la tercera durante otros 4, debéis cambiar el código para que la primera bomba esté encendida 3 segundos, luego funcione la primera y la segunda 2 segundos y por último la tercera esté encendida 5 segundos.



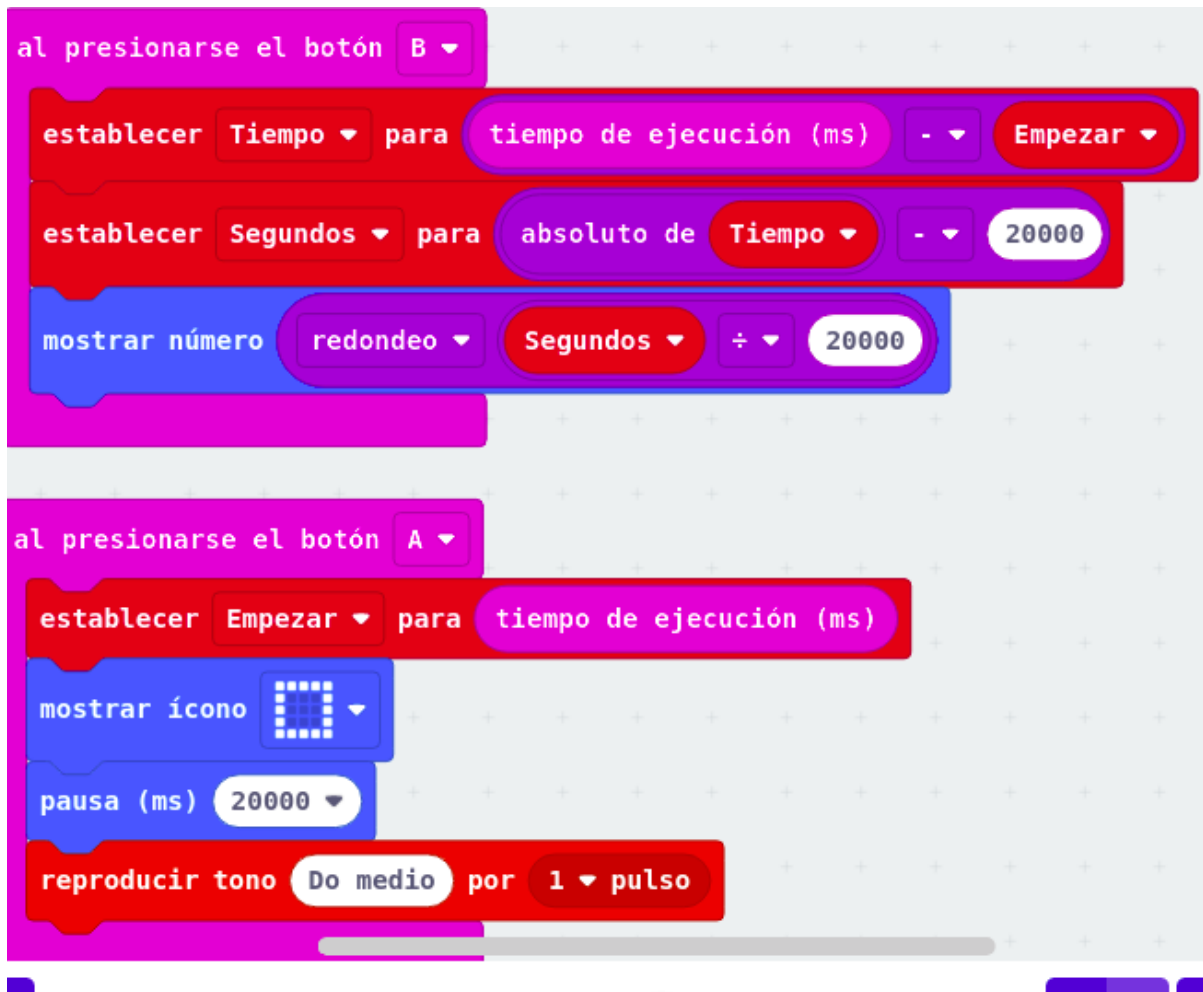
Práctica número 47, en esta práctica utilizaremos el sensor de voz, cuando detecte que hay sonido debe mostrar la imagen de una flecha hacia arriba , sonar una melodía y esperar 4 segundos, cuando detecte que el nivel de sonido es bajo mostrar una flecha hacia abajo.



Práctica número 48, en esta práctica vamos a hacer que la placa micro bit se convierta en una brújula, para ello directamente copiamos el código del dibujo, al inicio después de cargar el programa habrá que calibrarla girando la placa hasta completar el encendido de los LED de la misma.

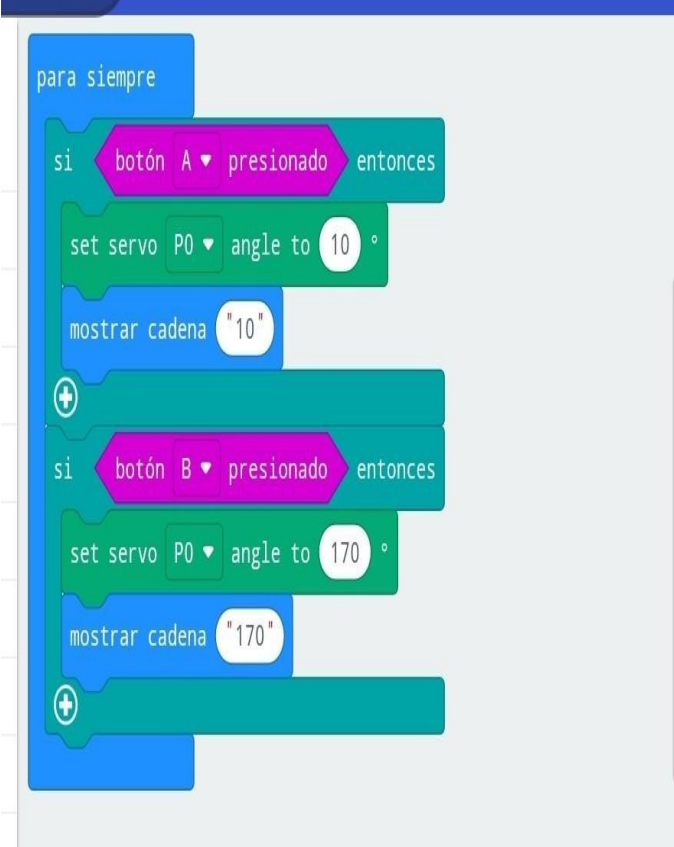
```
para siempre
  establecer degrees para dirección de la brújula (°)
  si degrees < 45 o degrees > 315 entonces
    mostrar cadena "N"
  si no, si degrees < 135 entonces
    mostrar cadena "E"
  si no, si degrees < 225 entonces
    mostrar cadena "S"
  si no
    mostrar cadena "0"
```

Práctica número 49, en esta práctica vamos a hacer un juego que consiste en calcular el transcurrir de una serie de segundos, para iniciar el cronómetro pulsamos en el botón A y para pararlo en el B, cuando pulsamos el botón B nos mostrará cuántos segundos de diferencia hay con respecto a los que hemos programado, debéis cambiar el código para que el conteo sea de 30 segundos y el icono que aparezca cuando pulsemos A sea diferente.



50

Práctica
para
mover con
la tarjeta
microbit
un
servomotor
.

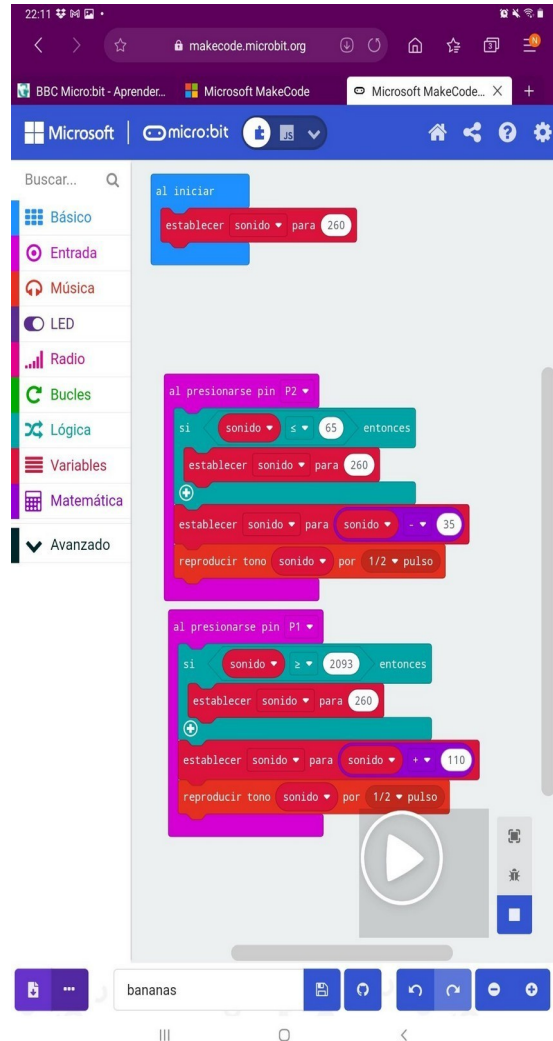


```
para siempre
si botón A presionado entonces
  set servo P0 angle to 10 °
  mostrar cadena "10"
+
si botón B presionado entonces
  set servo P0 angle to 170 °
  mostrar cadena "170"
+
```

The image shows a Scratch script for a Micro:bit. It starts with a 'para siempre' (forever) loop. Inside the loop, there are two 'if' blocks. The first 'if' block checks 'si botón A presionado entonces' (if button A is pressed then) and contains two blocks: 'set servo P0 angle to 10 °' and 'mostrar cadena "10"'. The second 'if' block checks 'si botón B presionado entonces' (if button B is pressed then) and contains two blocks: 'set servo P0 angle to 170 °' and 'mostrar cadena "170"'. There are also two empty 'if' blocks with plus signs, indicating where more code can be added.

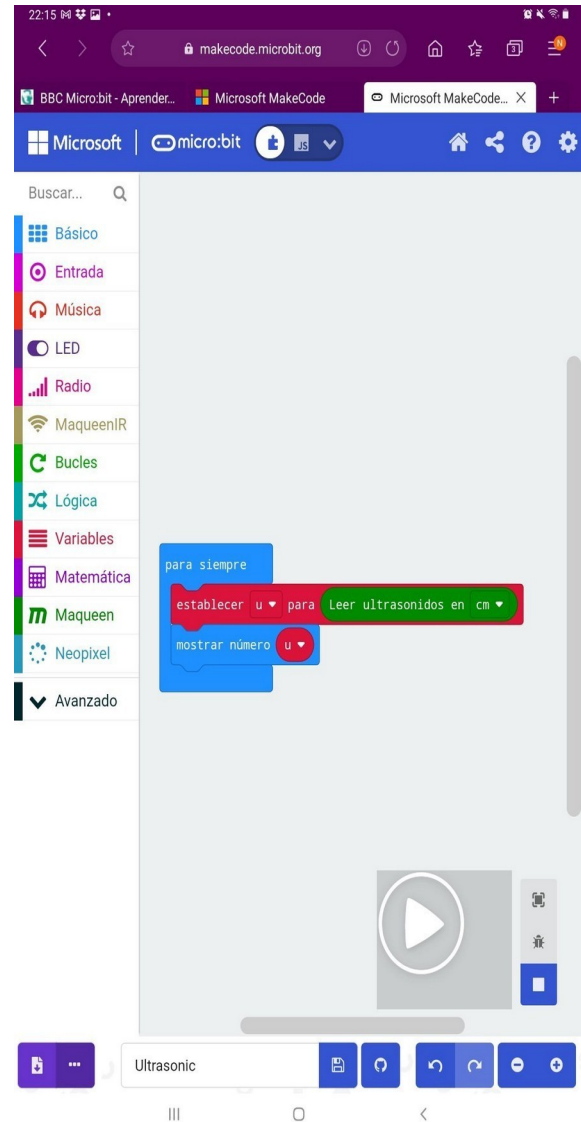
51

Práctica para hacer sonar notas musicales tocando bananas o otra fruta.



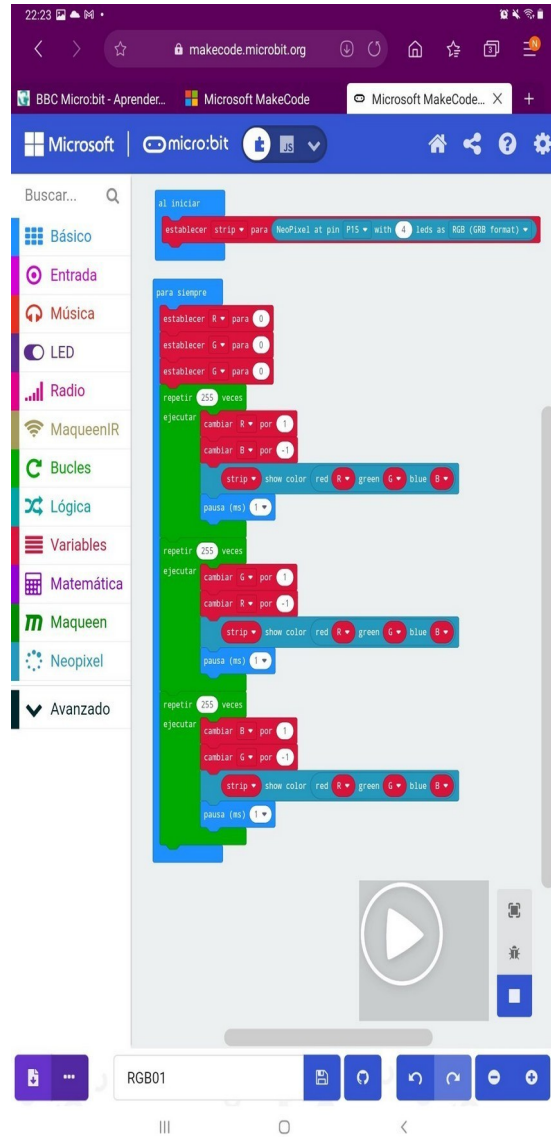
52

Práctica para medir distancia mediante Maqueen.



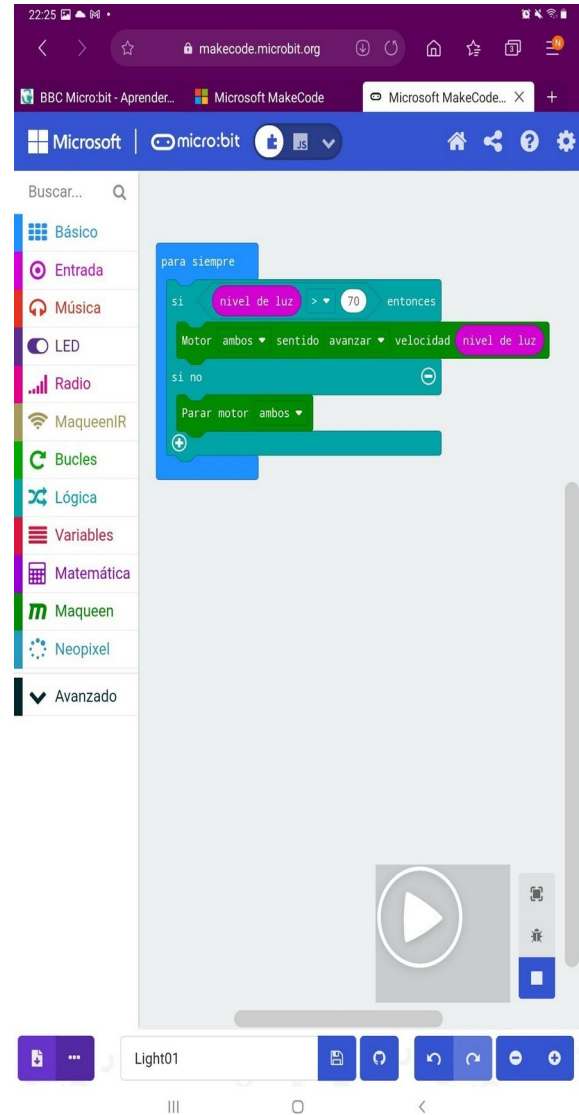
53

Práctica para mostrar LED de colores con Maqueen.



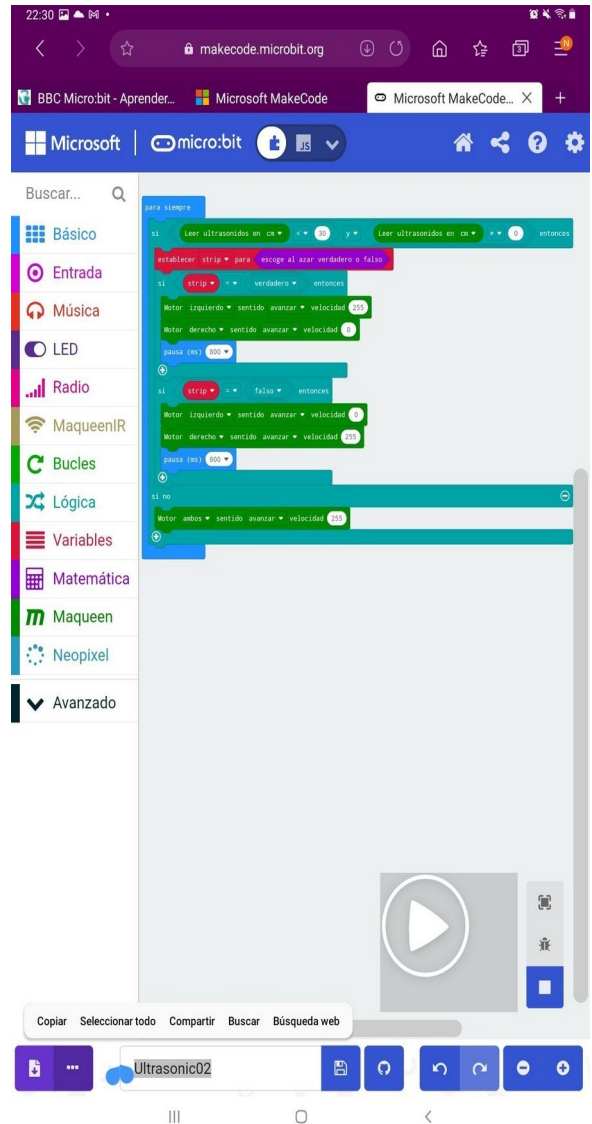
54

Práctica para que Maqueen se mueva cuando le de cierta cantidad de luz.



55

Práctica para que Maqueen se mueva esquivando obstáculos.



56

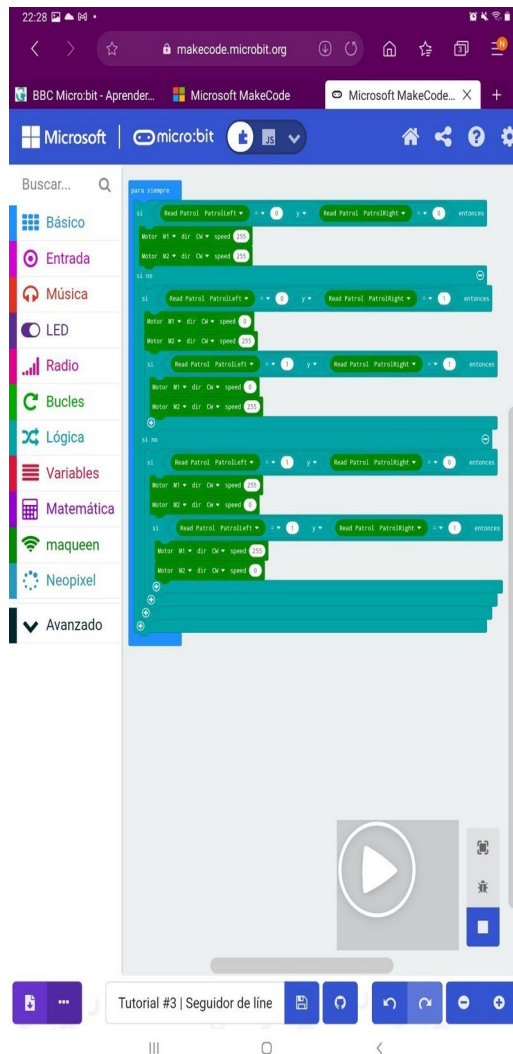
Práctica para que Maqueen mueva un servomotor.





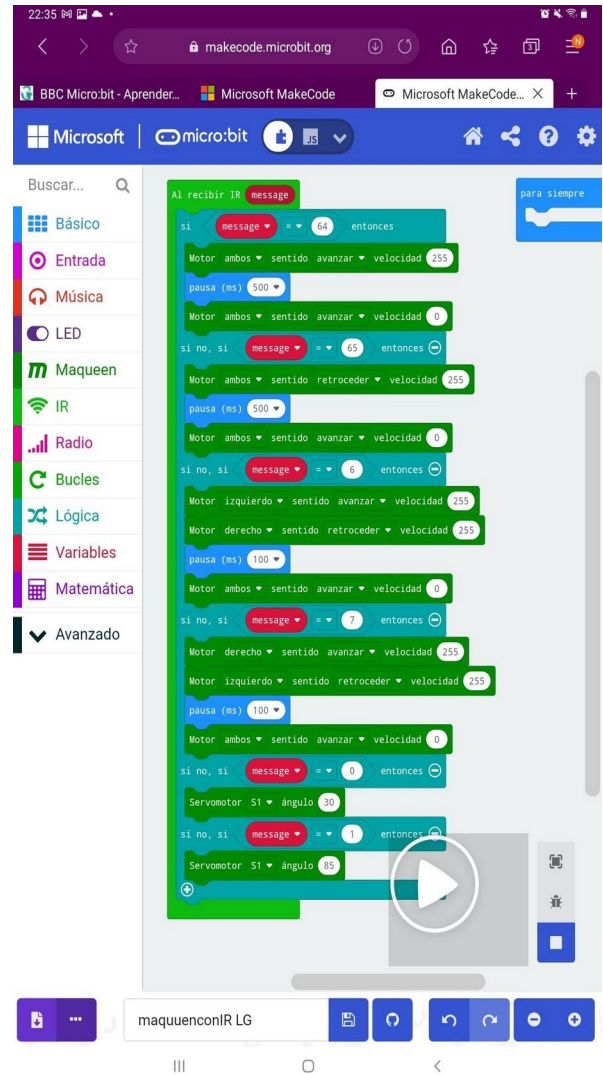
57

Práctica para que Maqueen se mueva siguiendo una línea negra.



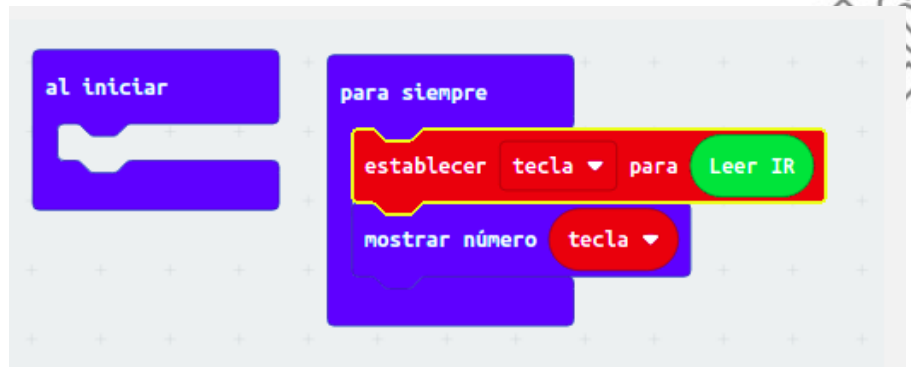
58

Práctica para que Maqueen se mueva siguiendo códigos de infrarrojos para televisor LG (opción 2 en la app).

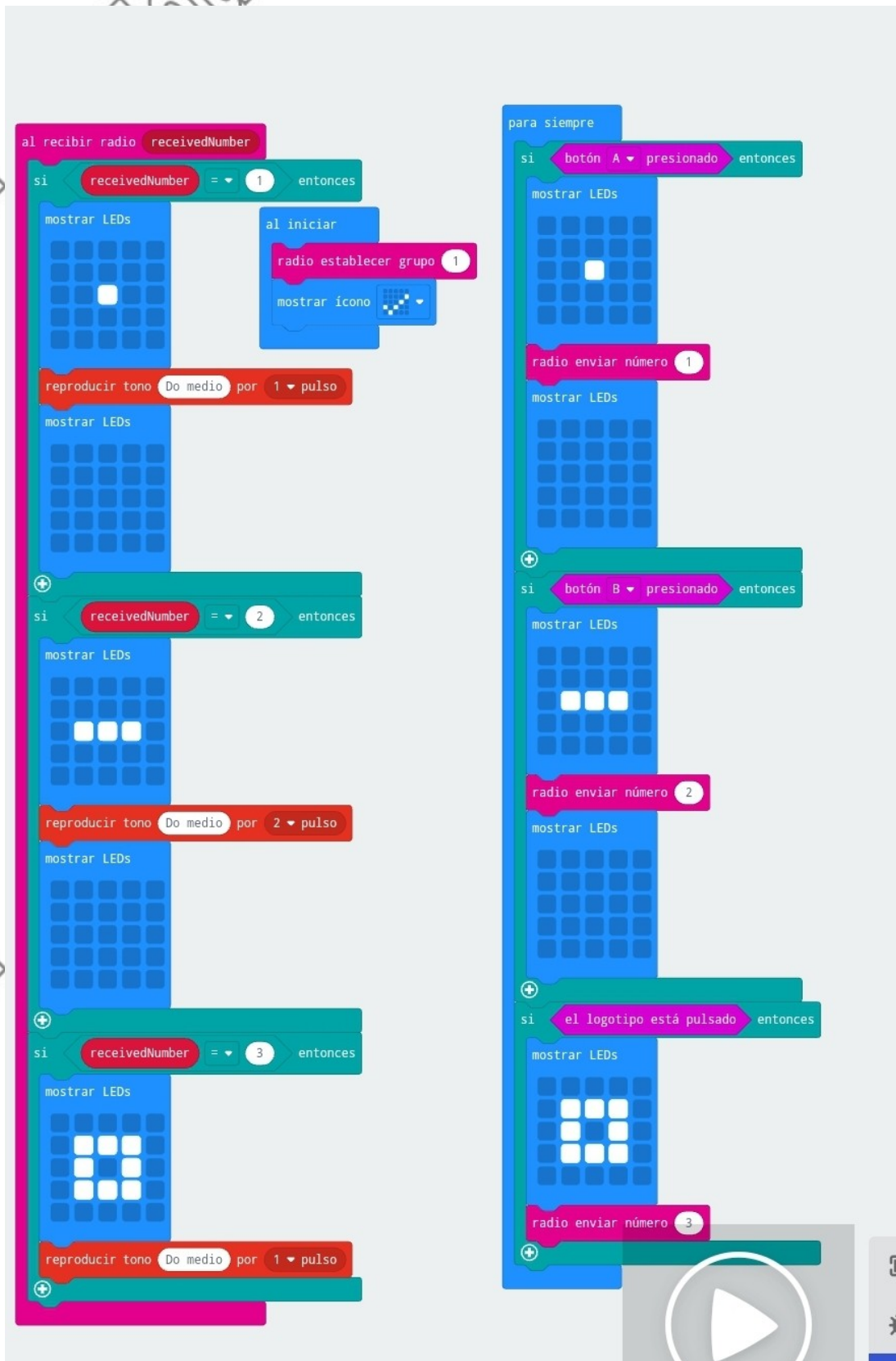


59

Práctica para que Maqueen lea las señales de infrarrojo que se le manda.



60 Código morse por radio



The image shows a Scratch script for a Morse code project. The script is organized into two main vertical columns of code blocks.

Left Column (Radio Reception Logic):

- al recibir radio** (when radio receives) block with a **receivedNumber** variable.
- si receivedNumber = 1 entonces** (if receivedNumber = 1 then) block:
 - mostrar LEDs** (show LEDs) block with a 1x1 grid.
 - al iniciar** (when started) block:
 - radio establecer grupo 1** (radio set group 1) block.
 - mostrar ícono** (show icon) block with a Morse code icon.
 - reproducir tono Do medio por 1 pulso** (play tone Do medium for 1 pulse) block.
 - mostrar LEDs** (show LEDs) block with a 1x1 grid.
- +** (add) block.
- si receivedNumber = 2 entonces** (if receivedNumber = 2 then) block:
 - mostrar LEDs** (show LEDs) block with a 1x2 grid.
 - reproducir tono Do medio por 2 pulso** (play tone Do medium for 2 pulse) block.
 - mostrar LEDs** (show LEDs) block with a 1x2 grid.
- +** (add) block.
- si receivedNumber = 3 entonces** (if receivedNumber = 3 then) block:
 - mostrar LEDs** (show LEDs) block with a 2x2 grid.
 - reproducir tono Do medio por 1 pulso** (play tone Do medium for 1 pulse) block.
- +** (add) block.

Right Column (Button Press Logic):

- para siempre** (forever) loop:
 - si botón A presionado entonces** (if button A pressed then) block:
 - mostrar LEDs** (show LEDs) block with a 1x1 grid.
 - radio enviar número 1** (radio send number 1) block.
 - mostrar LEDs** (show LEDs) block with a 1x1 grid.
 - +** (add) block.
 - si botón B presionado entonces** (if button B pressed then) block:
 - mostrar LEDs** (show LEDs) block with a 1x2 grid.
 - radio enviar número 2** (radio send number 2) block.
 - mostrar LEDs** (show LEDs) block with a 1x2 grid.
 - +** (add) block.
 - si el logotipo está pulsado entonces** (if logo is clicked then) block:
 - mostrar LEDs** (show LEDs) block with a 2x2 grid.
 - radio enviar número 3** (radio send number 3) block.
 - +** (add) block.

61 Semáforo hombre -mujer

The image shows a Scratch script for a traffic light simulation. The script is organized as follows:

- al iniciar** (when green flag clicked):
 - radio establecer grupo 21
- al recibir radio receivedNumber** (when I receive radio message):
 - si receivedNumber = 1 entonces** (if receivedNumber = 1 then):
 - repetir 4 veces (repeat 4 times):
 - ejecutar (do):
 - mostrar icono (show icon)
 - pausa (ms) 100 (wait 100 ms)
 - mostrar LEDs (show LEDs)
 - repetir 4 veces (repeat 4 times):
 - ejecutar (do):
 - mostrar icono (show icon)
 - pausa (ms) 100 (wait 100 ms)
 - reproducir tono Do medio por 1 pulso (play sound Do middle C for 1 pulse)
 - mostrar LEDs (show LEDs)
 - si receivedNumber = 2 entonces** (if receivedNumber = 2 then):
 - repetir 4 veces (repeat 4 times):
 - ejecutar (do):
 - mostrar LEDs (show LEDs)
 - pausa (ms) 100 (wait 100 ms)
 - mostrar LEDs (show LEDs)
 - repetir 4 veces (repeat 4 times):
 - ejecutar (do):
 - mostrar LEDs (show LEDs)
 - pausa (ms) 100 (wait 100 ms)
 - reproducir tono Do medio por 1 pulso (play sound Do middle C for 1 pulse)
 - mostrar LEDs (show LEDs)

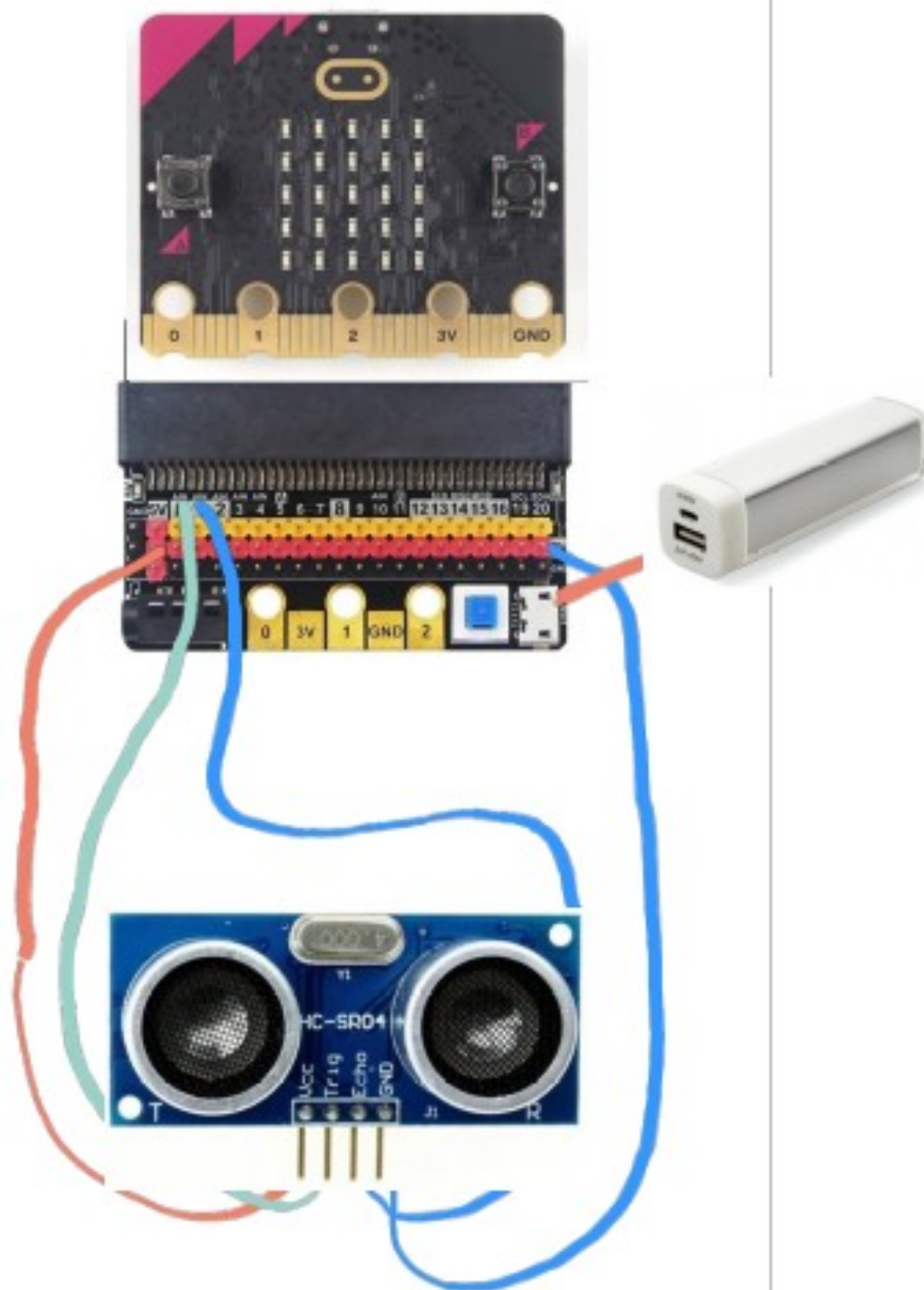
On the right side of the script area, there are two event blocks:

- al presionarse el botón A** (when button A is pressed):
 - radio enviar número 1 (send number 1 via radio)
- al presionarse el botón B** (when button B is pressed):
 - radio enviar número 2 (send number 2 via radio)

At the bottom right of the Scratch interface, there is a large play button icon.

62 Sensor de aparcamiento

```
para siempre
  mostrar ícono [ícono de sensor]
  si [botón A presionado] entonces
    repetir 6 veces
      ejecutar [mostrar LEDs]
```



63 Solicitar ayuda y encender lámpara con sensor de ldr a distancia (por radio)

The image shows a Scratch script for a radio-controlled LED lamp system. The script is organized into two main sections: a 'when green flag clicked' section and a 'when radio received' section.

When green flag clicked:

- al iniciar:** radio establecer grupo 1
- para siempre:**
 - mostrar icono (LED grid icon)
 - si botón B presionado entonces:
 - mostrar LEDs (LED grid)
 - radio enviar número 1
 - si nivel de luz < 30 entonces:
 - mostrar LEDs (LED grid)
 - radio enviar número 2
 - si nivel de luz > 30 entonces:
 - radio enviar número 3

When radio received:

- si receivedNumber = 1 entonces:
 - mostrar LEDs (LED grid)
 - reproducir melodia a tempo
- si receivedNumber = 2 entonces:
 - mostrar LEDs (LED grid)
 - escritura digital pin P1 a 1
- si receivedNumber = 3 entonces:
 - mostrar LEDs (LED grid)
 - escritura digital pin P1 a 0