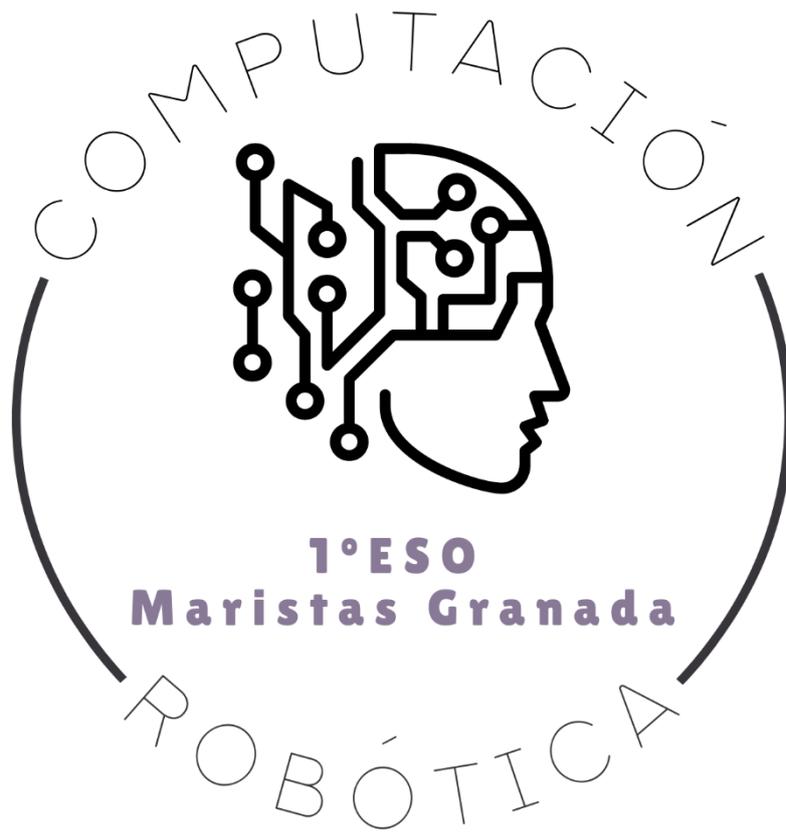


CURSO 2023-2024



RETO 13: MICRO:BIT Y DESPLAZAMIENTO DEL ROBOT MAQUEEN

COMPUTACIÓN Y ROBÓTICA 1ºESO

COLEGIO MARISTA LA INMACULADA
CALLE SÓCRATES, 8
18002 - GRANADA

Índice

0. Ubicación en la programación	2
1. Ubicación en la programación	3
2. Aprendemos practicando	4
2.1 Introducir comentarios en programas largos.....	4
2.2 Detectar obstáculos con maqueen. Nivel básico	4
2.3 Ampliar la detección de obstáculos y encender leds	6

0. Ubicación en la programación

Título: Reto 13. Micro:bit y desplazamiento del robot Maqueen.

Evaluación: Segunda.

Temporalidad: 1 sesión (2 horas consecutivas)

Índice de contenidos: Continuar aprendiendo y ganando habilidad en el entorno de programación por bloques. Creación de códigos de programación. Creación y uso de variables. Manejo de placa microcontroladora y de robot asociado a la placa.

Breve resumen de la situación: Seguimos programando en la placa micro:bit pero en esta sesión la placa la conectaremos con el robot Maqueen, el cual programaremos para que se desplace y evite obstáculos.

1. Ubicación en la programación

Repasamos la conexión de la placa con el robot Maqueen.

En esta sesión se trabajará por grupos, donde cada grupo tendrá un Maqueen para poder aplicar lo trabajado en clase. Al terminar cada programa el grupo deberá mostrar al profesor el resultado de lo configurado en su robot.

Todos los miembros del grupo deberán de participar en la elaboración del código y entenderlo, ya que el profesor le preguntará a un miembro aleatorio para que le explique lo programado. Además, como grupo deberán entregar una ficha con las impresiones de pantalla de cada programa donde aparezca el usuario, el código desarrollado que se pida y una breve explicación de dicho código.

La calificación final de la ficha de trabajo es la media de la calificación obtenida en cada uno de los retos.

Los retos se califican según los siguientes criterios:

- Están completos todos los retos y bien explicados.
- Están incluidas todas las impresiones de pantalla necesarias junto a un usuario.
- Se ejecuta correctamente el programa final creado y se muestra al profesor.

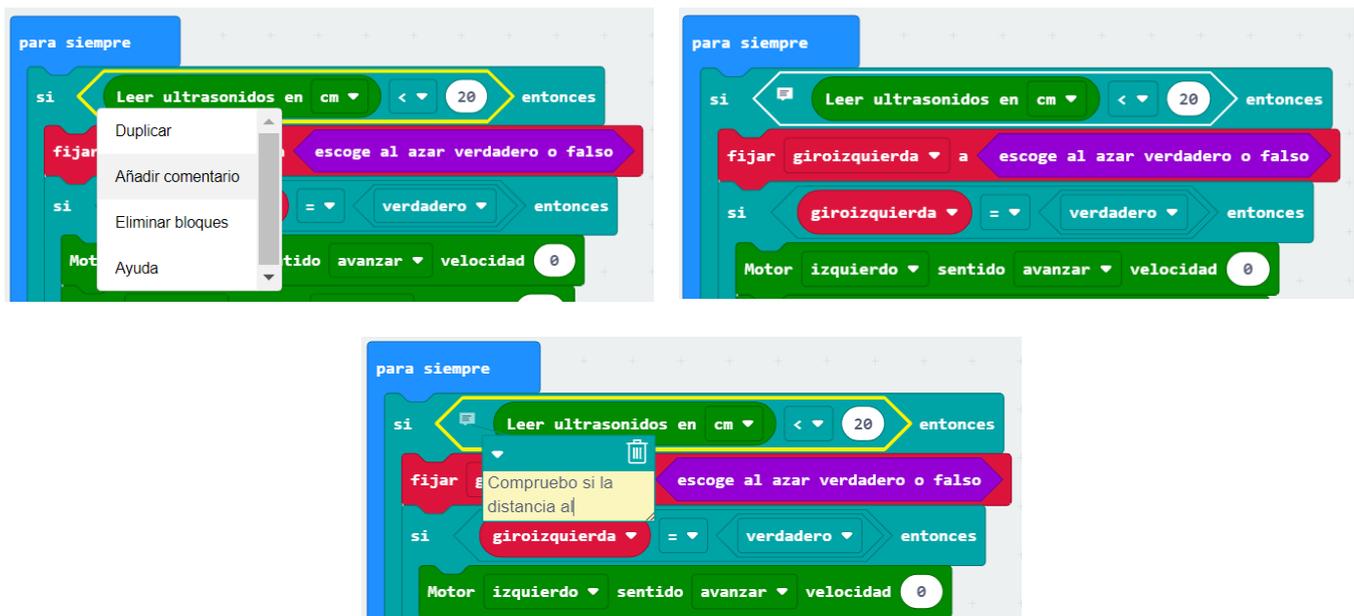
2. Aprendemos practicando

Anteriormente vimos como conectar la placa micro:bit con el robot Maqueen, descargar sus extensiones y como encender sus leds. Partiendo de ahí, en esta sesión vamos a seguir profundizando en dicho robot y vamos a hacer que se desplace evitando obstáculos próximos con los que podría chocar.

2.1 Introducir comentarios en programas largos

Cuando creamos un programa largo es probable que, al pasar los días y retomar el código, no nos acordemos bien de la lógica del programa. Para ello es muy recomendable introducir comentarios: texto escrito que explica qué estamos haciendo en un punto concreto del código. Esto nos facilitará el comprender bien lo programado y ayudará a elaborar programas más complejos.

Cuando introduces un bloque puedes pulsar sobre él con el botón derecho, elegir “Añadir comentario” y teclear en un cuadro de texto.



Acostúmbrate a comentar tus programas. Esto te dará un mayor conocimiento sobre tu código.

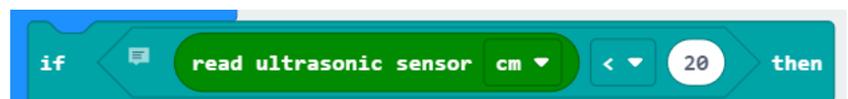
2.2 Detectar obstáculos con maqueen. Nivel básico

Creamos un proyecto nuevo e instalamos la extensión del robot maqueen.

Vamos a trabajar con el bloque “para siempre”, para garantizar que el código se ejecute constantemente. Recordamos que este bloque era un bucle que se repetía infinitamente.

Usando el sensor de ultrasonidos de la parte delantera, podemos introducir en el bloque “para siempre” un condicional que continuamente decida si la distancia detectada es inferior o no a 20 centímetros.

El sensor de ultrasonidos se encuentra en la extensión maqueen. Se introduce dentro del condicional “Si” para detectar si hay o no un obstáculo a menos de 20 cm.



Si el condicional se cumple, de manera aleatoria, giraremos a izquierda o a derecha . Si el condicional no se cumple, avanzaremos en línea recta durante un tiempo determinado. Generamos un aleatorio (True o False) con un comando del bloque de Matemáticas. Y almacenamos el resultado de la decisión aleatoria dentro de una variable, que llamaremos “giroizquierda”.

Si el valor aleatorio es True, la variable “giroizquierda” será True y decidiremos girar a la izquierda. Si el valor aleatorio es False, giraremos a la derecha.

La imagen de la derecha te muestra toda la lógica del condicional “Si”, cuando se cumple la condición marcada por el sensor de ultrasonidos.

Si maqueen detecta el obstáculo a menos de 20 cm, el programa entra dentro del condicional “Si”. Y genera un aleatorio (pick random) entre True o False.

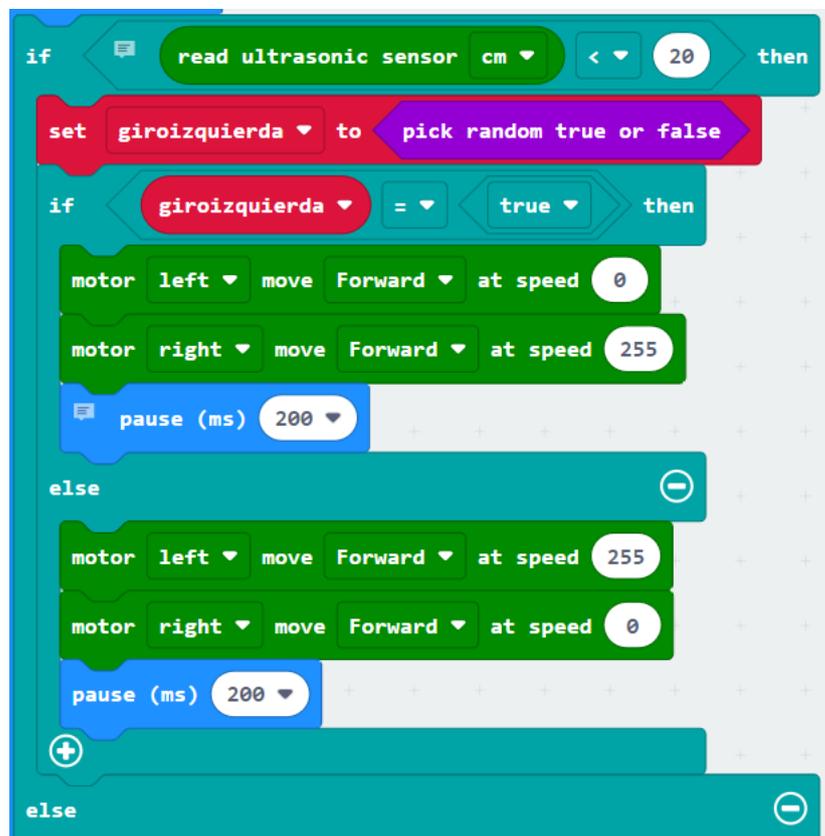
Si el aleatorio es True, la variable “giroizquierda” toma el valor True y entramos en el segundo condicional “Si”, que pone el motor izquierdo a 0 para que el coche gire a la izquierda.

Si el aleatorio es False, no entramos en el segundo condicional “Si”. Entramos en “Si No” (else). Y colocamos el motor derecho a 0 para que el coche gire a la derecha.

Fíjate que si el sensor no detecta ningún obstáculo por debajo de 20 cm, el condicional principal “Si” no se ejecuta. Y pasaríamos directamente a la parte final de “Si No”, situado en la parte inferior de la imagen de la derecha.



```
set giroizquierda to pick random true or false
```

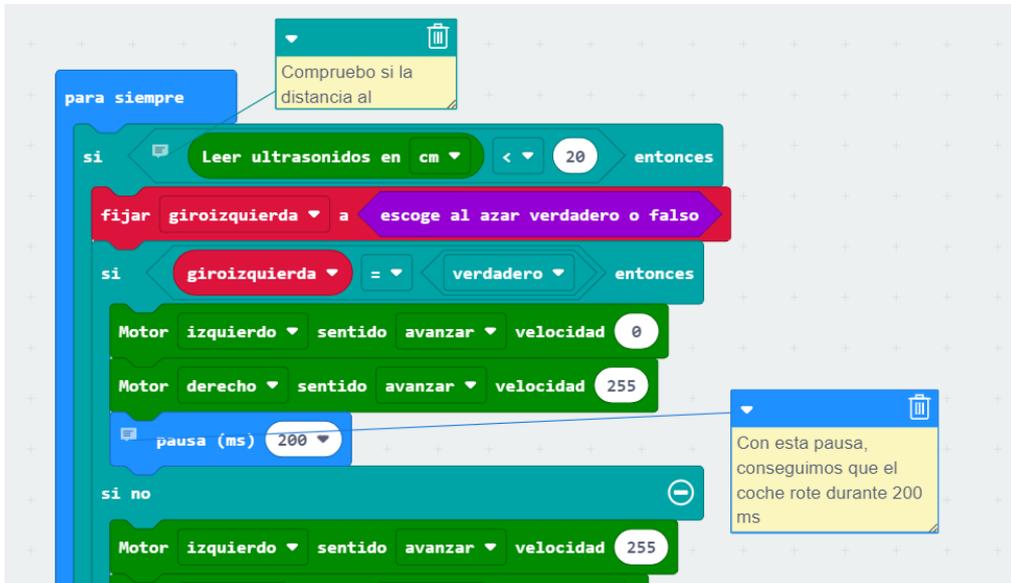


```

if read ultrasonic sensor cm < 20 then
  set giroizquierda to pick random true or false
  if giroizquierda = true then
    motor left move Forward at speed 0
    motor right move Forward at speed 255
    pause (ms) 200
  else
    motor left move Forward at speed 255
    motor right move Forward at speed 0
    pause (ms) 200
else
  
```

¿Qué hace la instrucción “Si No” del final? Simplemente indica que el coche avance en línea recta, ya que no se ha detectado ningún obstáculo. Y el código volvería a comenzar, ya que está contenido dentro del bloque “para siempre”.

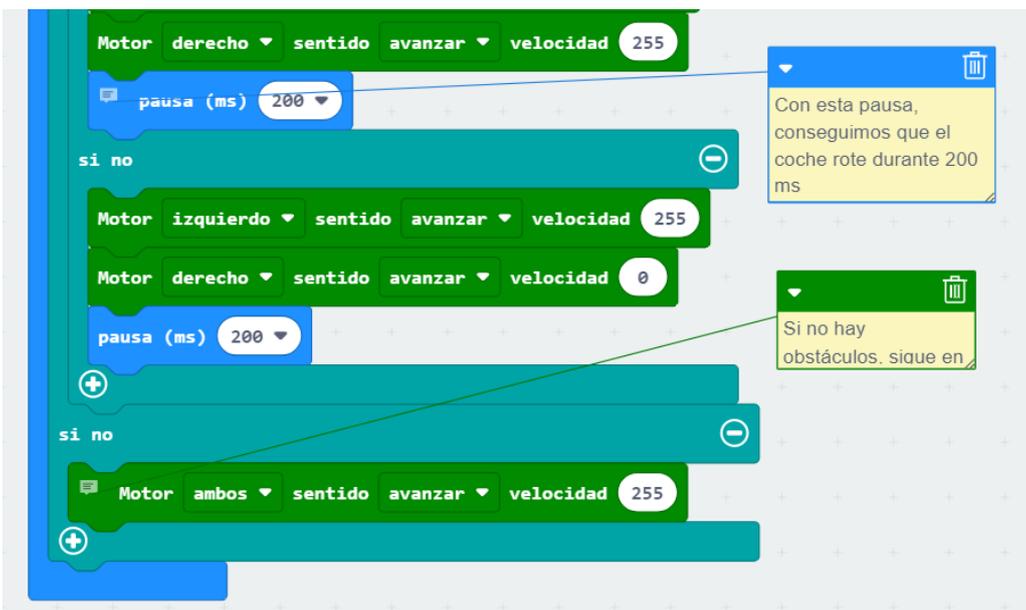
A continuación tienes dos imágenes con el código completo.



```

para siempre
  Compruebo si la distancia al
  si Leer ultrasonidos en cm < 20 entonces
    fijar giroizquierda a escoge al azar verdadero o falso
    si giroizquierda = verdadero entonces
      Motor izquierdo sentido avanzar velocidad 0
      Motor derecho sentido avanzar velocidad 255
      pausa (ms) 200
    si no
      Motor izquierdo sentido avanzar velocidad 255
  
```

Con esta pausa, conseguimos que el coche rote durante 200 ms



```

Motor derecho sentido avanzar velocidad 255
  pausa (ms) 200
  si no
    Motor izquierdo sentido avanzar velocidad 255
    Motor derecho sentido avanzar velocidad 0
    pausa (ms) 200
  si no
    Motor ambos sentido avanzar velocidad 255
  
```

Con esta pausa, conseguimos que el coche rote durante 200 ms

Si no hay obstáculos, sigue en

2.3 Ampliar la detección de obstáculos y encender leds

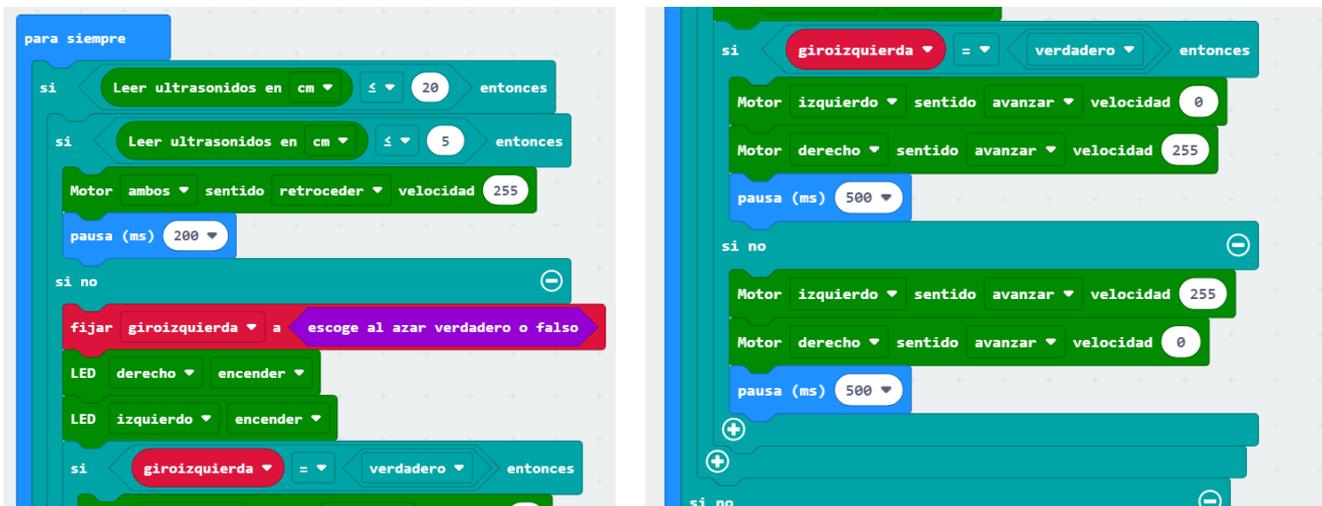
El código anterior tiene un serio problema: si el obstáculo es muy ancho (por ejemplo, una pared), maqueen gira al acercarse a la pared, pero tras cada pausa de 200 ms, vuelve a avanzar en línea recta, repite el bucle, y poco a poco termina estrellándose con el obstáculo.

Con el programa anterior conseguimos evitar objetos pequeños y muy separados entre sí (por ejemplo, botellas de plástico situadas en el suelo). Pero con una caja de zapatos o una mochila, lo normal es que maqueen termine impactando con el obstáculo si no mejoramos nuestro código.

¿Qué podemos hacer?

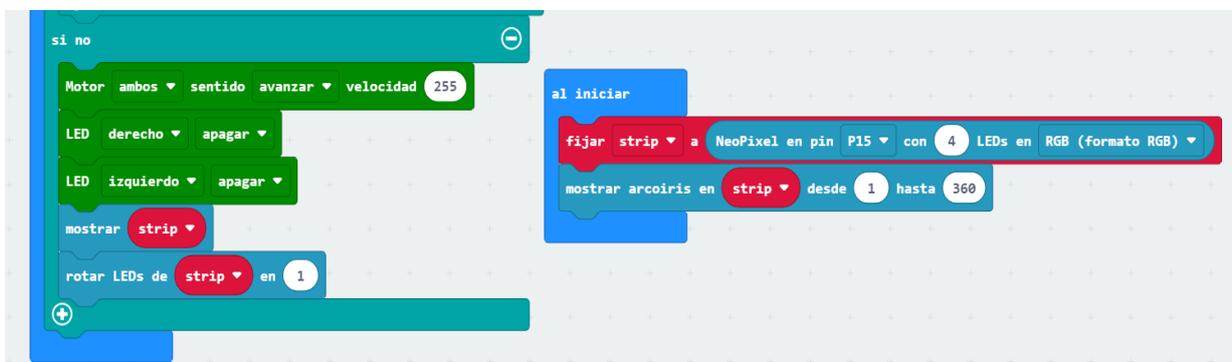
Si el coche no encuentra obstáculos, mantenemos alegremente su camino en línea recta. Pero si detecta un obstáculo a una distancia igual o inferior a 20 cm, **añadimos un segundo condicional más restrictivo, por si el obstáculo se encuentra a menos de 5 cm, que haga retroceder al coche durante 200 ms antes de volver a girar aleatoriamente a izquierda o derecha.** Además, cuando se detecte el obstáculo a menos de 20 cm, podemos aumentar el tiempo de giro de 200 ms a 500 ms, para superar los obstáculos más anchos.

También podemos encender los leds delanteros para indicar visualmente cuándo el coche ha detectado un obstáculo a menos de 20 cm. Así sabremos, visualmente, que es probable que se active el segundo condicional para distancias inferiores a 5 cm. Los leds de maqueen se controlan con una nueva extensión, que debes instalar: Neopixel.



The image shows two screenshots of Scratch code blocks. The left screenshot shows a 'para siempre' (forever) loop containing two 'si' (if) blocks. The first 'si' block checks 'Leer ultrasonidos en cm' (Read ultrasonics in cm) ≤ 20. If true, it sets 'Motor ambos' (Both motors) to 'sentido retroceder' (reverse) with 'velocidad 255' (speed 255) and adds a 'pausa (ms) 200' (pause 200 ms) block. The second 'si' block checks 'Leer ultrasonidos en cm' ≤ 5. If true, it sets 'Motor ambos' to 'sentido retroceder' with 'velocidad 255' and adds a 'pausa (ms) 200' block. After these blocks, there is a 'fijar giroizquierda' (set turn left) block set to 'escoge al azar verdadero o falso' (pick randomly true or false), followed by 'LED derecho' (right LED) and 'LED izquierdo' (left LED) both set to 'encender' (turn on). The right screenshot shows a 'si giroizquierda = verdadero' (if turn left = true) block. If true, it sets 'Motor izquierdo' (left motor) to 'sentido avanzar' (forward) with 'velocidad 0' (speed 0) and 'Motor derecho' (right motor) to 'sentido avanzar' with 'velocidad 255', followed by a 'pausa (ms) 500' block. If false, it sets 'Motor izquierdo' to 'sentido avanzar' with 'velocidad 255' and 'Motor derecho' to 'sentido avanzar' with 'velocidad 0', followed by a 'pausa (ms) 500' block.

Finalmente, cuando no detecte obstáculo alguno y avance en línea recta, vamos a encender en arcoíris los 4 RGB inferiores del coche, que son iniciados a los colores del arcoíris, al principio, con un bloque "al iniciar".



The image shows a screenshot of Scratch code blocks. On the left, there is a 'si no' (if not) block. Inside, it sets 'Motor ambos' (Both motors) to 'sentido avanzar' (forward) with 'velocidad 255' (speed 255), 'LED derecho' (right LED) to 'apagar' (turn off), and 'LED izquierdo' (left LED) to 'apagar'. It then adds 'mostrar strip' (show strip) and 'rotar LEDs de strip en 1' (rotate LEDs of strip by 1). On the right, there is an 'al iniciar' (when started) block. It sets 'fijar strip' (set strip) to 'NeoPixel en pin P15' (NeoPixel on pin P15) with '4 LEDs en RGB (formato RGB)' (4 LEDs in RGB format) and adds 'mostrar arcoiris en strip desde 1 hasta 360' (show rainbow on strip from 1 to 360).