

# JUEGO DE TRIGONOMETRÍA



## STEMJAM Teaching Guide

Developing make spaces to promote creativity  
around STEM in schools

Acronym: STEMJAM

Project no. 2016-1-ES01-KA201-025470

[www.stemjam.eu](http://www.stemjam.eu)



Co-funded by the  
Erasmus+ Programme  
of the European Union

# JUEGO DE TRIGONOMETRÍA

## RESUMEN

El objetivo es crear un juego para que los estudiantes midan distancias y calculen ángulos.

El profesor, utilizando tarjetas de diferentes colores, dibujará una figura geométrica en un tablero de papel, usando tarjetas rojas para los bordes y verdes para los vértices.

Los estudiantes tendrán que pasar por estas cifras con el mBot a través de solicitudes que el software planteará al usuario.

## OBJETIVOS DIDÁCTICOS

- ❖ Aplicar la fórmula:  $\text{Velocidad} = \text{Espacio} / \text{Tiempo}$ .
- ❖ Entender la fórmula de cálculo de los ángulos polígonos equiláteros.
- ❖ Conocer el triángulo escaleno y sus fórmulas.
- ❖ Conocer el área del rectángulo y sus fórmulas.

Materias STEM:      Ciencia       Tecnología       Ingeniería       Matemáticas

Nivel Educativo:      12-14 años       14-16 años

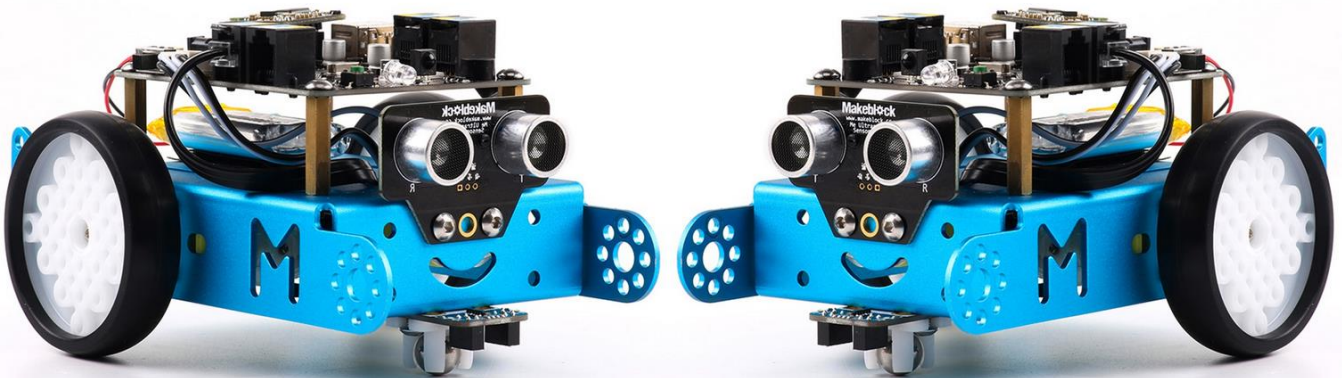
## PLANTEAMIENTO DEL PROBLEMA

Algunos estudiantes tienen dificultades para calcular ángulos y medir distancias, por lo que, mediante un juego que realizarán con los mBots, entrenarán los procedimientos que les ayudarán a realizar estos cálculos.



## LISTADO DE MATERIALES NECESARIOS

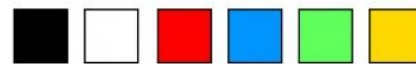
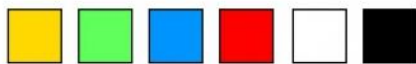
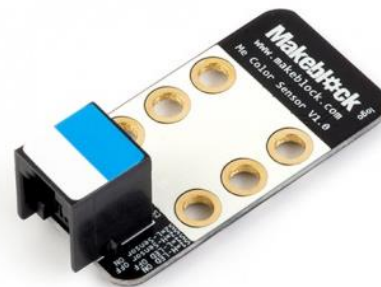
- (x2) mBot => Ref. 90054



- ❖ mBot Ranger (temporizador)



- ❖ (x2) Un Sensor de Color para cada robot:



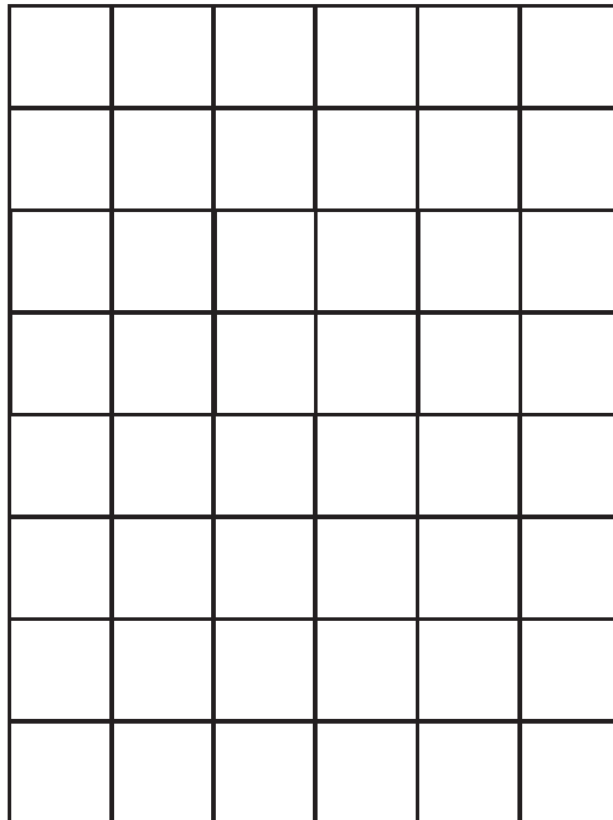
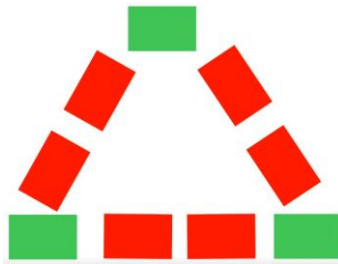
ELEMENT	ID	CABLE	AMOUNT	PORT 1			PORT 2			PORT 3				PORT 4				P.MOT1	P.MOT2
				Y	B	W	Y	B	W	Y	B	W	Bl	Y	B	W	Bl	W*	W*
Mbot Robot 2'4G			2																
Motor 1	W*																W*		
Motor 2	W*																W*		
Me RJ 25 adapter	Y																		
	B																		
	Bl																		
Mini Pan-Tilt kit																			
It has 2 servos.																			
We have to connect the servo to a RJ25 adapter																			
Mini Gripper																			
We have to connect the servo to a RJ25 adapter																			
Me 7-Segment serial display	B																		
Me Led Matrix 8x16	B	(1)	2										B						
Me Ultrasonic sensor	Y																		
Me Temperature Sensor - Waterproof	Y																		
Me Line Follower	B																		
Me Flame sensor	Bl																		
Me PIR Motion sensor	B																		
Me Sound sensor	Bl																		
Me Touch sensor	B																		
Mini Fan Pack	B																		
Me Color Sensor	B	(1)	2			B													
Me Temperature and Humidity sensor	Y																		
Me 130 Motor Fan Pack	B																		
RJ25 cables			4																
Structures and beams																			
Laptops																			
Attrezzo (not essential)																			

## DESCRIPCIÓN DE LA ACTIVIDAD

### Primera versión

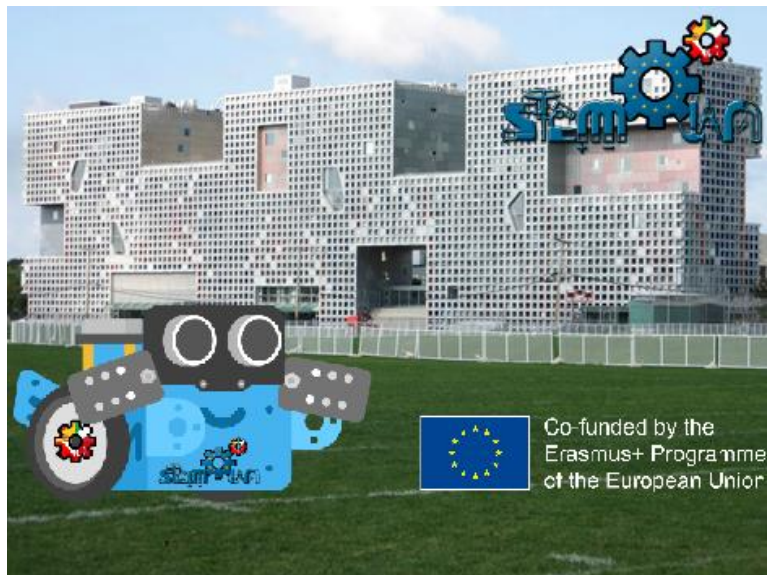
La actividad consiste en crear un juego para que los alumnos midan las distancias y calculen ángulos. El profesor dibujará una figura geométrica en un tablero de papel con tarjetas de diferentes colores, rojo para los bordes y verdes para los vértices y puntos de control. Luego, los estudiantes tendrán que pasar por estas figuras con el mBot mediante preguntas que el software planteará al usuario.

El primer paso es diseñar el tablero de papel (que puede encontrar en la carpeta de actividades) y las diferentes tarjetas:



El siguiente paso consiste en crear el software para el juego:

1. Diseño de las escenas para el juego:



2. Iniciamos la programación:

a. Creamos las variables que necesitaremos en nuestro software:

Variable Name	Annotation
Box1	Here, we display the different options that the user may be done
Box2	
Box3	
Checkpoints	For display the checkpoints that the user has got
Mistakes	For calculate the mistakes and hits
Puntuation	
advance	The CMs that mBot advance
degrees	The degrees that mBot turn right
mBot_SpeedBack	These variables are the speed that mBot advance or return
mBot_SpeedGo	
return	The CMs that mBot go backward

b. El código al inicio del juego es el siguiente:

```
when clicked
hide variable Box1
hide variable Box2
hide variable Box3
hide variable Checkpoints
set degrees to 0
set advance to 0
set return to 0
set mBot_SpeedGo to 100
set mBot_SpeedBack to -100
set Checkpoints to 3
set Box1 to
set Box2 to
set Box3 to
broadcast Game Start
```

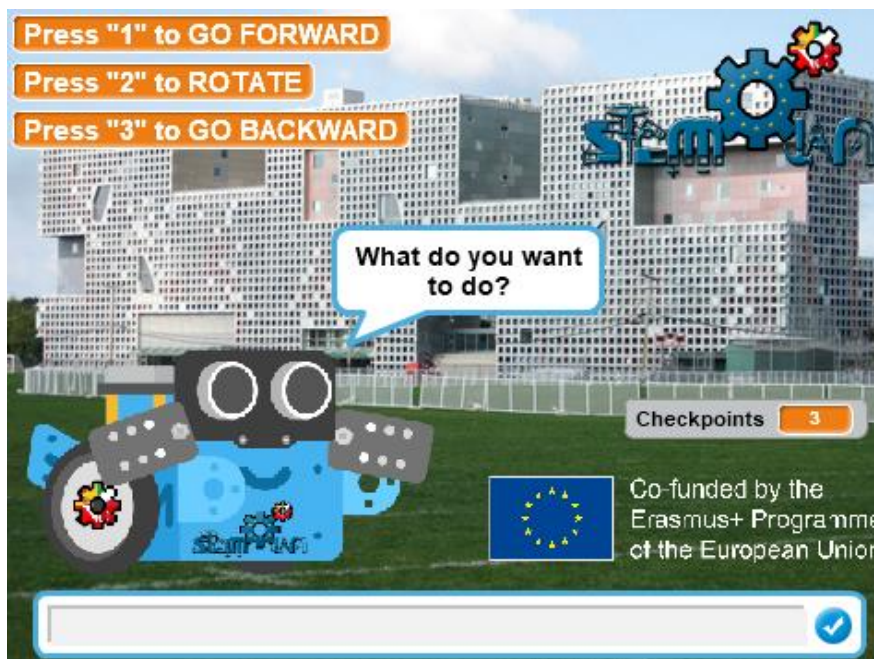
c. El código de la pantalla de "BIENVENIDA" es:

```
when I receive Game Start
say Hello! for 2 secs
say Welcome to the great MBOT trigonometry con for 3 secs
ask What is the name of your team and wait
repeat until not answer =
ask Please, insert the name of your tea and wait
say join Great answer for 2 secs
say Let's GO for 2 secs
decision
```



d. El software solicita al usuario la operación que desea realizar:

```
define decision
  show variable Box1
  show variable Box2
  show variable Box3
  show variable Checkpoints
  set Box1 to Press "1" to GO FORWARD
  set Box2 to Press "2" to ROTATE
  set Box3 to Press "3" to GO BACKWARD
  ask What do you want to do and wait
  if answer = 1 then
    advance
  if answer = 2 then
    rotate
  if answer = 3 then
    return
repeat until not answer = and answer = 1 or answer = 2 or answer = 3
  ask Please, insert the correct num and wait
  if answer = 1 then
    advance
  if answer = 2 then
    rotate
  if answer = 3 then
    return
```

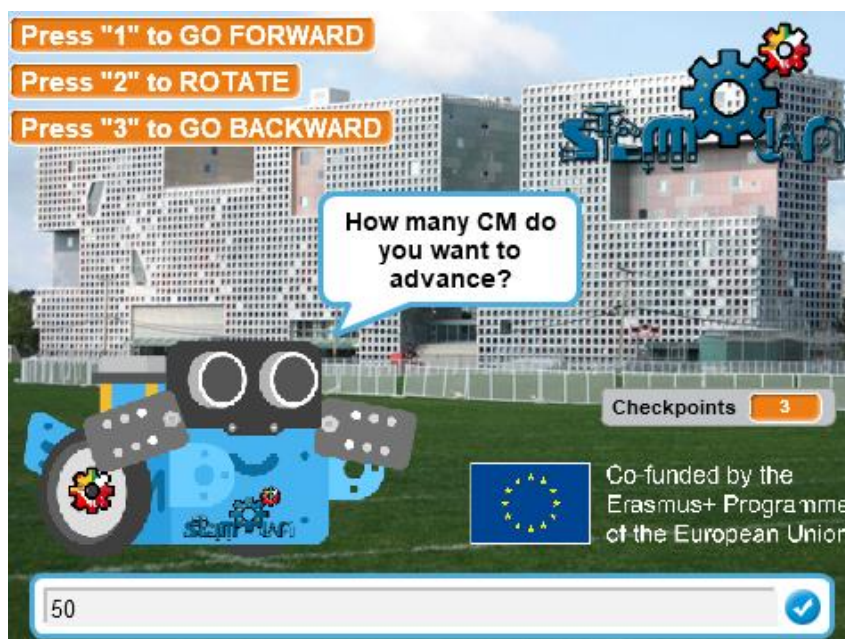




Los "checkpoints" indican los puntos de control que faltan por pasar.

e. Si selecciona la primera opción, "go forward" (avanzar), los códigos son los siguientes:

```
define advance
  ask How many CM do you want to advance and wait
  set advance to answer * 4.1 / 50
  say join Your mBot goes forward join answer cm for 2 secs
  set motor M1 speed mBot_SpeedGo
  set motor M2 speed mBot_SpeedGo
  if (ArduinoMode ONLY) colorsensor Port1 detected green then
    show drawing Port4 x: 0 y: 0 draw: [drawing]
    play tone on note C4 beat Half
    set Checkpoints to Checkpoints - 1
    if Checkpoints = 0 then
      say Good job!! You can return to the departure bd for 2 secs
      set Checkpoints to 3
    wait 2 secs
    show drawing Port4 x: 0 y: 0 draw: [drawing]
  wait advance secs
  run forward at speed 0
decision
```

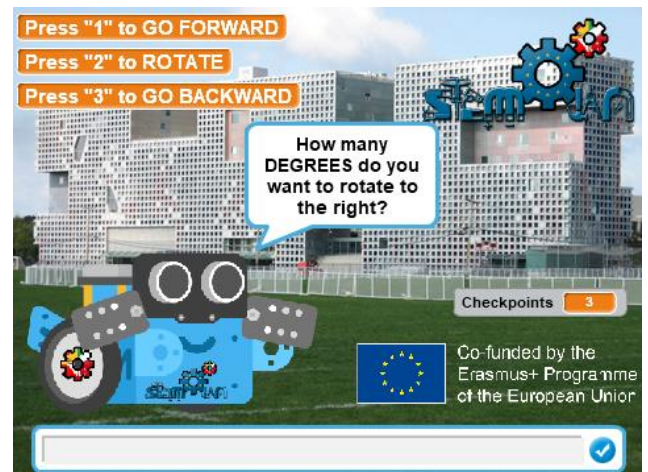


Cuando el mBot pasa sobre una tarjeta verde, el sensor de color lo detectará y mostrará una sonrisa.

```
if (ArduinoMode ONLY)colosensor Port1 detected green then
  show drawing Port4 x: 0 y: 0 draw: 😊
  play tone on note C4 beat Half
  set Checkpoints to Checkpoints - 1
  if Checkpoints = 0 then
    say Good job!! You can return to the departure bo for 2 secs
    set Checkpoints to 3
  wait 2 secs
  show drawing Port4 x: 0 y: 0 draw: 😊
```

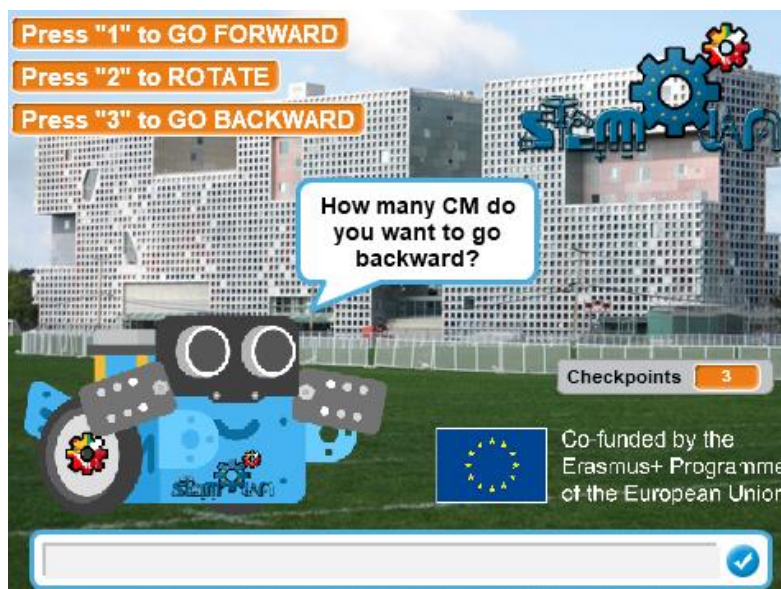
f. Y si el usuario selecciona la segunda opción, "rotate" (girar):

```
define rotate
  ask How many DEGREES do you want to rotate to the right and wait
  set degrees to answer * 0.77 / 90
  say join Your mBot rotates join answer degrees for 2 secs
  show drawing Port4 x: 0 y: 0 draw: ➡
  set motor M1 speed mBot_SpeedGo
  set motor M2 speed mBot_SpeedBack
  wait degrees secs
  run forward at speed 0
  if (ArduinoMode ONLY)colosensor Port1 detected green then
    show drawing Port4 x: 0 y: 0 draw: 😊
    play tone on note C4 beat Half
    set Checkpoints to Checkpoints - 1
    if Checkpoints = 0 then
      say Good job!! You can return to the departure bo for 2 secs
      set Checkpoints to 3
    wait 2 secs
    show drawing Port4 x: 0 y: 0 draw: 😊
  decision
```



Para el regreso, el código es muy similar al de la primera opción.

```
define return
ask How many CM do you want to go backward and wait
set return to answer * 4,1 / 50
say join Your mBot goes backward join answer cm for 2 secs
set motor M1 speed mBot_SpeedBack
set motor M2 speed mBot_SpeedBack
wait return secs
run forward at speed 0
if (ArduinoMode ONLY)colosensor Port1 detected green then
show drawing Port4 x: 0 y: 0 draw:
play tone on note C4 beat Half
set Checkpoints to Checkpoints - 1
if Checkpoints = 0 then
say Good job!! You can return to the departure bo for 2 secs
set Checkpoints to 3
wait 2 secs
show drawing Port4 x: 0 y: 0 draw:
decision
```



### 3. El código resultante sería:

```

define advance
  ask How many CM do you want to advance and wait
  set advance* to answer * 4.3 / 60
  say join Your mBot goes forward join answer cm for 2 secs
  set motor (M1) speed mBot_SpeedGo
  set motor (M2) speed mBot_SpeedGo
  if (Arduino Mode ONLY)color sensor (Port1) detected green? then
    show drawing (Port4) x: 0 y: 0 draw: 200
    play tone on note (C4) beat (Half)
    set Checkpoints* to Checkpoints - 1
    if Checkpoints = 0 then
      say Good job!! You can return to the departure L for 2 secs
      set Checkpoints* to 3
      wait 2 secs
      show drawing (Port4) x: 0 y: 0 draw: 200
    wait advance secs
    run forward at speed 0
  decision

define rotate
  ask How many DEGREES do you want to rotate to the L and wait
  set degrees* to answer * 0.77 / 60
  say join Your mBot rotates join answer degrees for 2 secs
  show drawing (Port4) x: 0 y: 0 draw: 200
  set motor (M1) speed mBot_SpeedGo
  set motor (M2) speed mBot_SpeedBack
  wait degrees secs
  run forward at speed 0
  if (Arduino Mode ONLY)color sensor (Port1) detected green? then
    show drawing (Port4) x: 0 y: 0 draw: 200
    play tone on note (C4) beat (Half)
    set Checkpoints* to Checkpoints - 1
    if Checkpoints = 0 then
      say Good job!! You can return to the departure L for 2 secs
      set Checkpoints* to 3
      wait 2 secs
      show drawing (Port4) x: 0 y: 0 draw: 200
    wait 2 secs
    show drawing (Port4) x: 0 y: 0 draw: 200
  decision

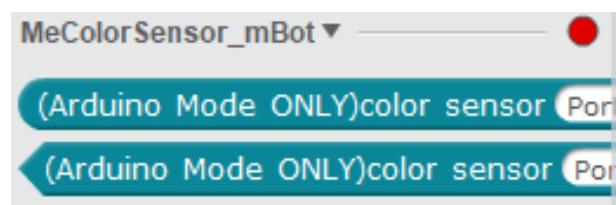
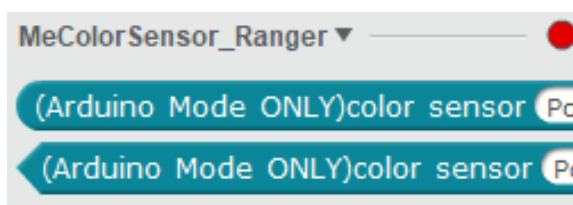
define return
  ask How many CM do you want to go backward and wait
  set return* to answer * 4.3 / 60
  say join Your mBot goes backward join answer cm for 2 secs
  set motor (M1) speed mBot_SpeedBack
  set motor (M2) speed mBot_SpeedBack
  wait return secs
  run forward at speed 0
  if (Arduino Mode ONLY)color sensor (Port1) detected green? then
    show drawing (Port4) x: 0 y: 0 draw: 200
    play tone on note (C4) beat (Half)
    set Checkpoints* to Checkpoints - 1
    if Checkpoints = 0 then
      say Good job!! You can return to the departure L for 2 secs
      set Checkpoints* to 3
      wait 2 secs
      show drawing (Port4) x: 0 y: 0 draw: 200
    wait 2 secs
    show drawing (Port4) x: 0 y: 0 draw: 200
  decision

define decision
  show variable Box1*
  show variable Box2*
  show variable Box3*
  show variable Checkpoints*
  set Box1* to Press "1" to GO FORWARD
  set Box2* to Press "2" to ROTATE
  set Box3* to Press "3" to GO BACKWARD
  ask What do you want to L and wait
  if answer = 1 then
    advance
  if answer = 2 then
    rotate
  if answer = 3 then
    return
  repeat until not answer = 1 and answer = 1 or answer = 2 or answer = 3
  ask Please, insert the correct numL and wait
  if answer = 1 then
    advance
  if answer = 2 then
    rotate
  if answer = 3 then
    return

when I receive Game Start
  say Hello! for 2 secs
  say Welcome to the great MBDT trigonometry cor for 2 secs
  ask What is the name of your team and wait
  repeat until not answer = 
  ask Please, insert the name of your L and wait
  say join Great answer for 2 secs
  say Let's GO! for 2 secs
  decision
  
```

Para instalar la librería en mBlock:

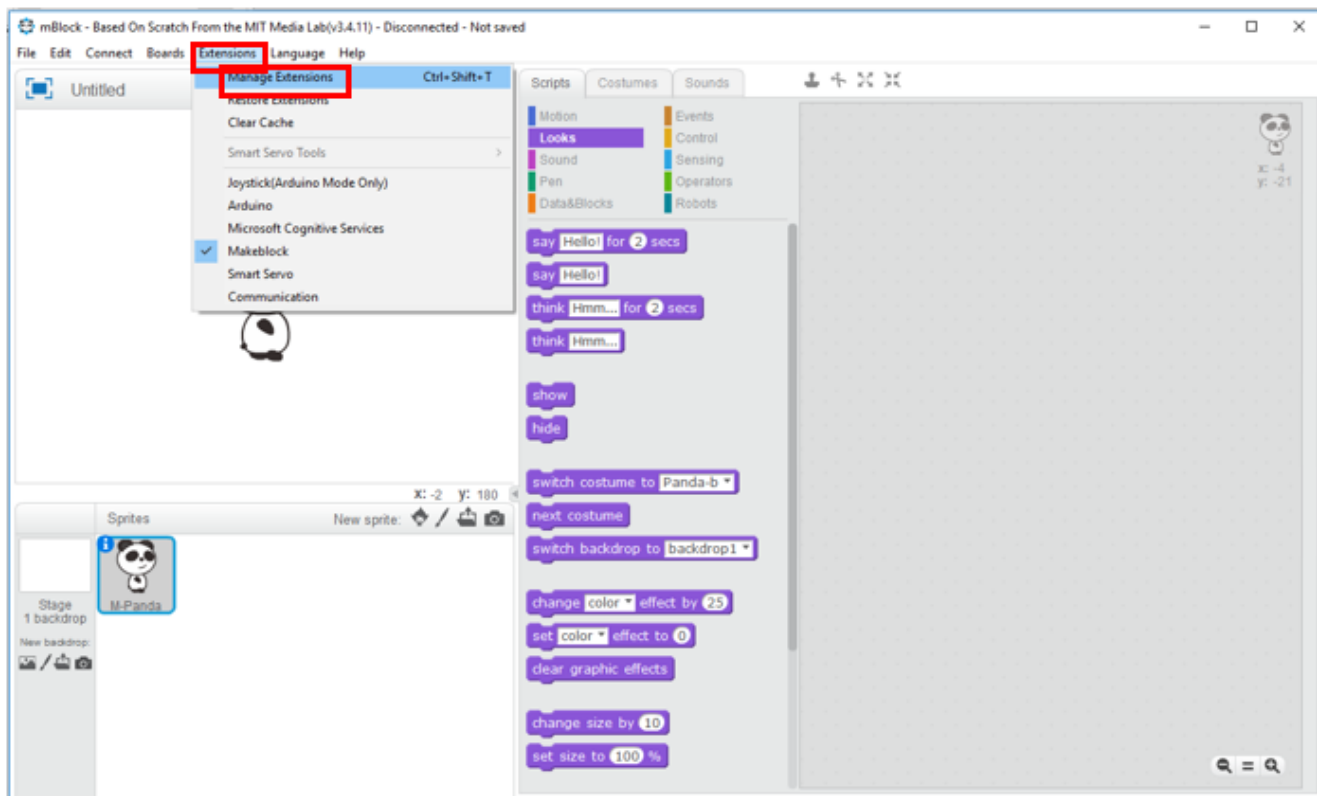
- Necesitamos instalar la librería en mBlock:



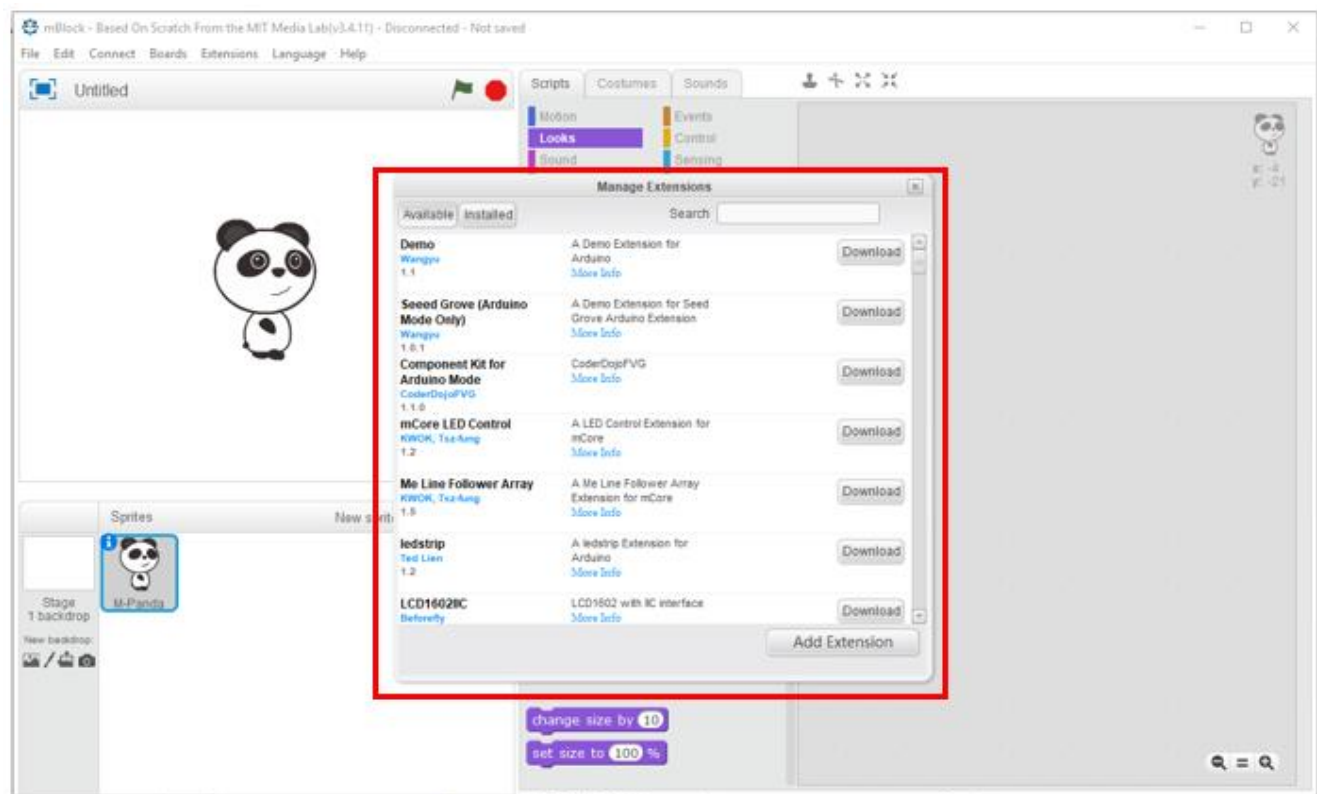
- Actualmente, sólo funciona si cargamos el código en la placa.

Para instalar la librería del sensor de color, seguiremos los siguientes pasos:

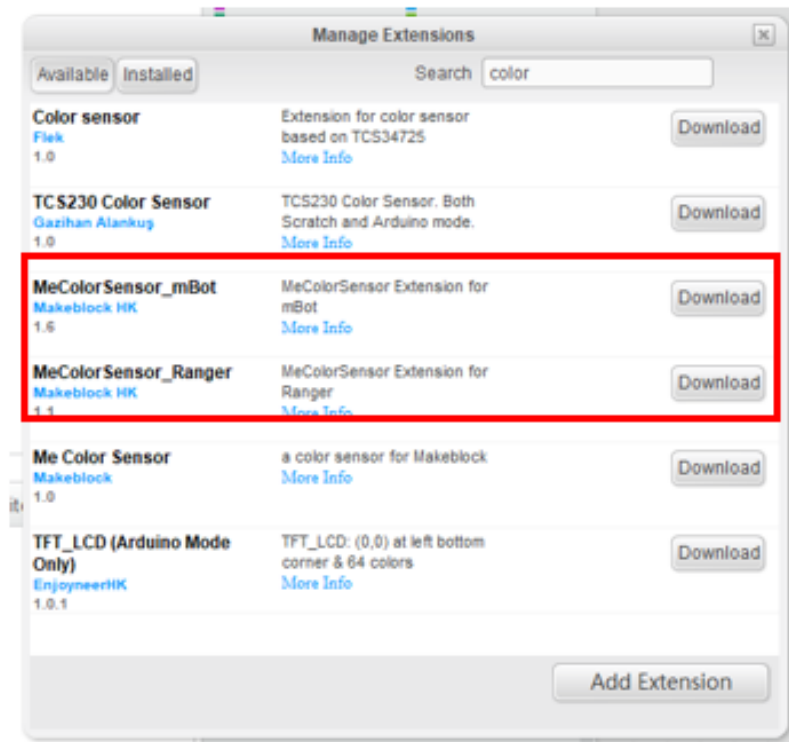
1. Inicie el software mBlock y vaya a "Extensiones" => "Administrar extensiones":



2. Aparecerá una nueva ventana:

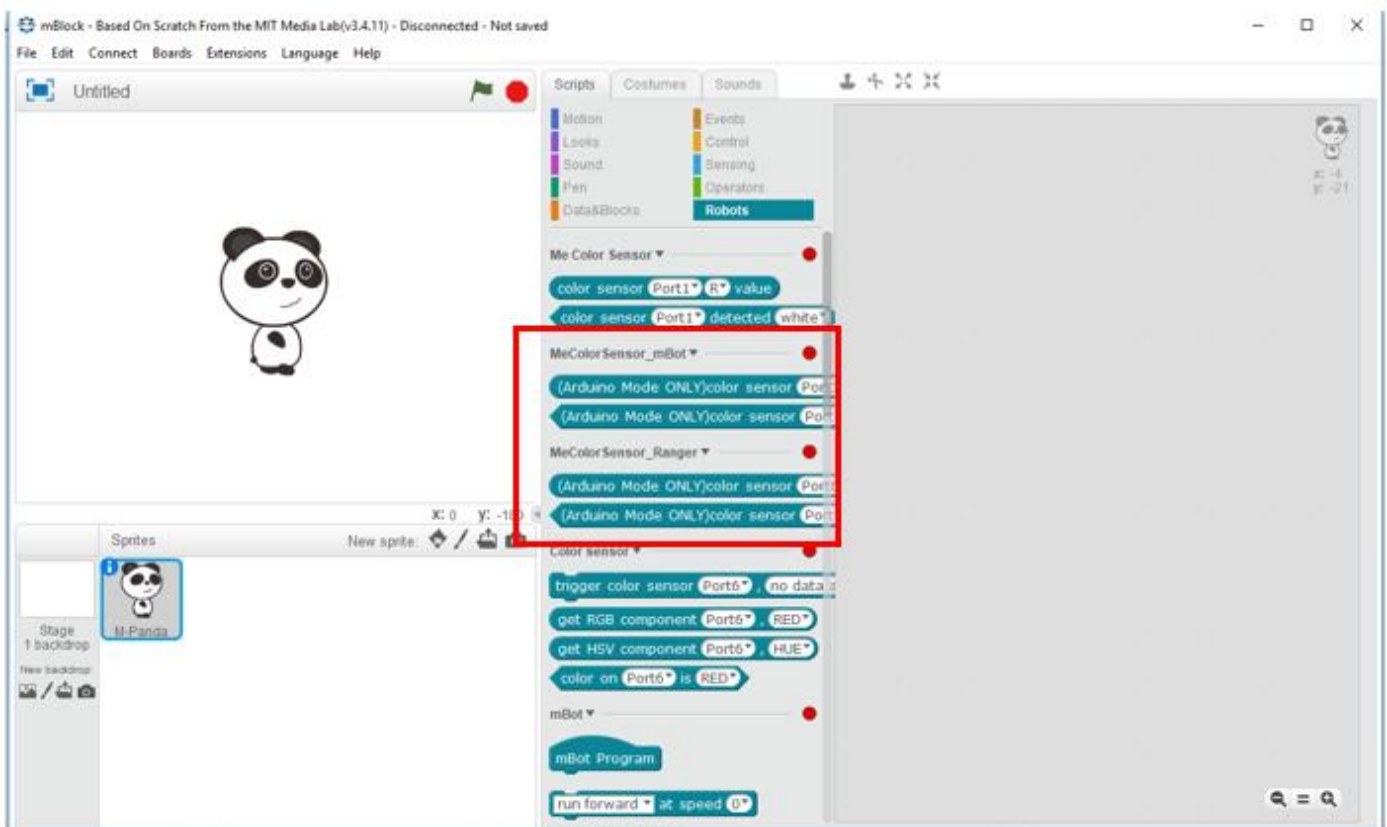


3. Introduzca la palabra "color" en el buscador:



4. Aparecerán "MeColorSensor\_mBot" y "MeColorSensor\_Ranger" will appear. Ahora, haga clic en "download".

5. Si todo salió bien, las bibliotecas instaladas aparecerán en la sección "robots".



Ahora, desarrollamos el código del temporizador para el mBot Ranger:



```

define Minute0
set led on board 1 red 0 green 255 blue 0

define Second30
set led on board 2 red 0 green 255 blue 0

define Minute1
set led on board 3 red 0 green 255 blue 0

define Minute1_30
set led on board 4 red 0 green 255 blue 0

define Minute2
set led on board 5 red 0 green 255 blue 0

define Minute2_30
set led on board 6 red 0 green 255 blue 0

define Minute3
set led on board 1 red 255 green 150 blue 0
set led on board 2 red 255 green 150 blue 0
set led on board 3 red 255 green 150 blue 0
set led on board 4 red 255 green 150 blue 0
set led on board 5 red 255 green 150 blue 0
set led on board 6 red 255 green 150 blue 0
set led on board 7 red 255 green 150 blue 0

define Minute3_30
set led on board 8 red 255 green 150 blue 0

define Minute4
set led on board 9 red 255 green 150 blue 0

define Minute4_30
set led on board 1 red 255 green 60 blue 0
set led on board 2 red 255 green 60 blue 0
set led on board 3 red 255 green 60 blue 0
set led on board 4 red 255 green 60 blue 0
set led on board 5 red 255 green 60 blue 0
set led on board 6 red 255 green 60 blue 0
set led on board 7 red 255 green 60 blue 0
set led on board 8 red 255 green 60 blue 0
set led on board 9 red 255 green 60 blue 0

define Minute5
set led on board 11 red 255 green 60 blue 0

define Minute5_30
repeat 10
set led on board all red 255 green 0 blue 0
wait 0.5 secs
set led on board all red 0 green 0 blue 0
wait 0.5 secs

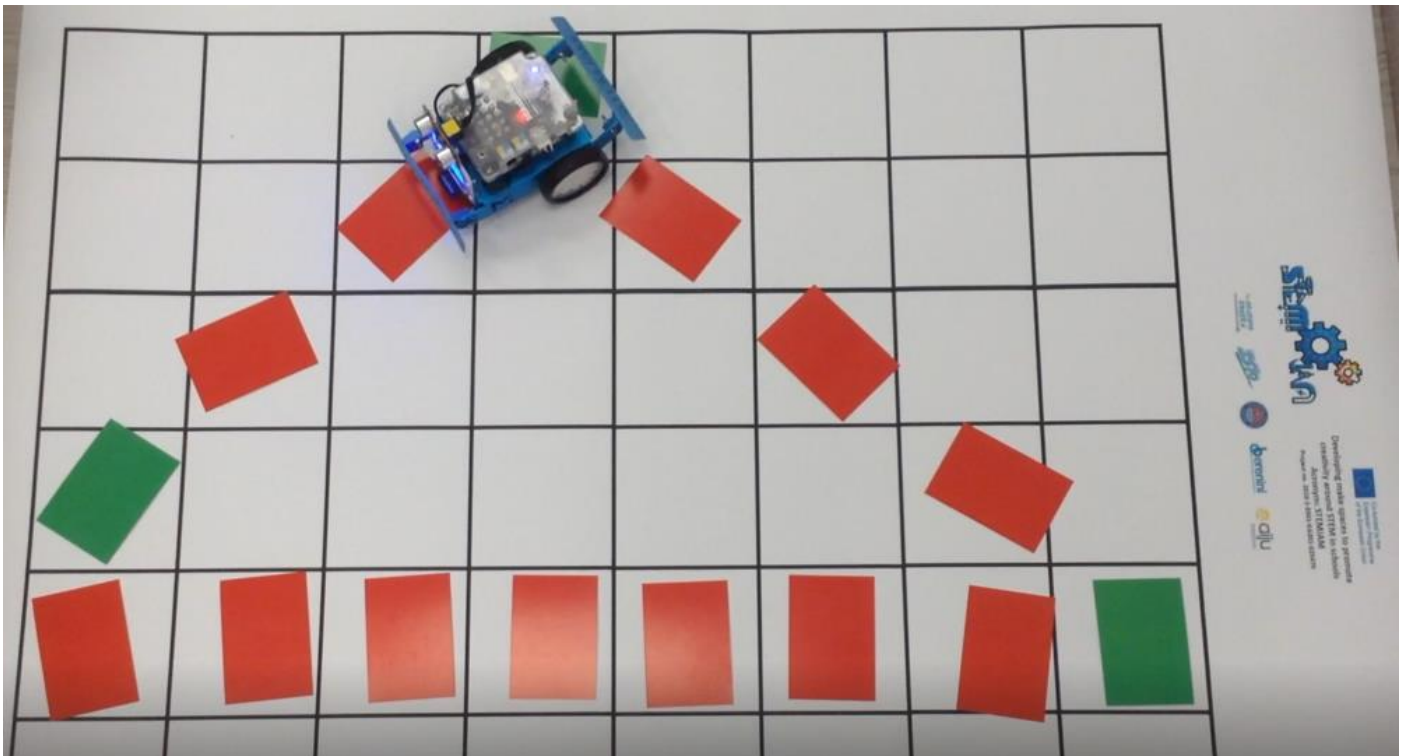
```

```

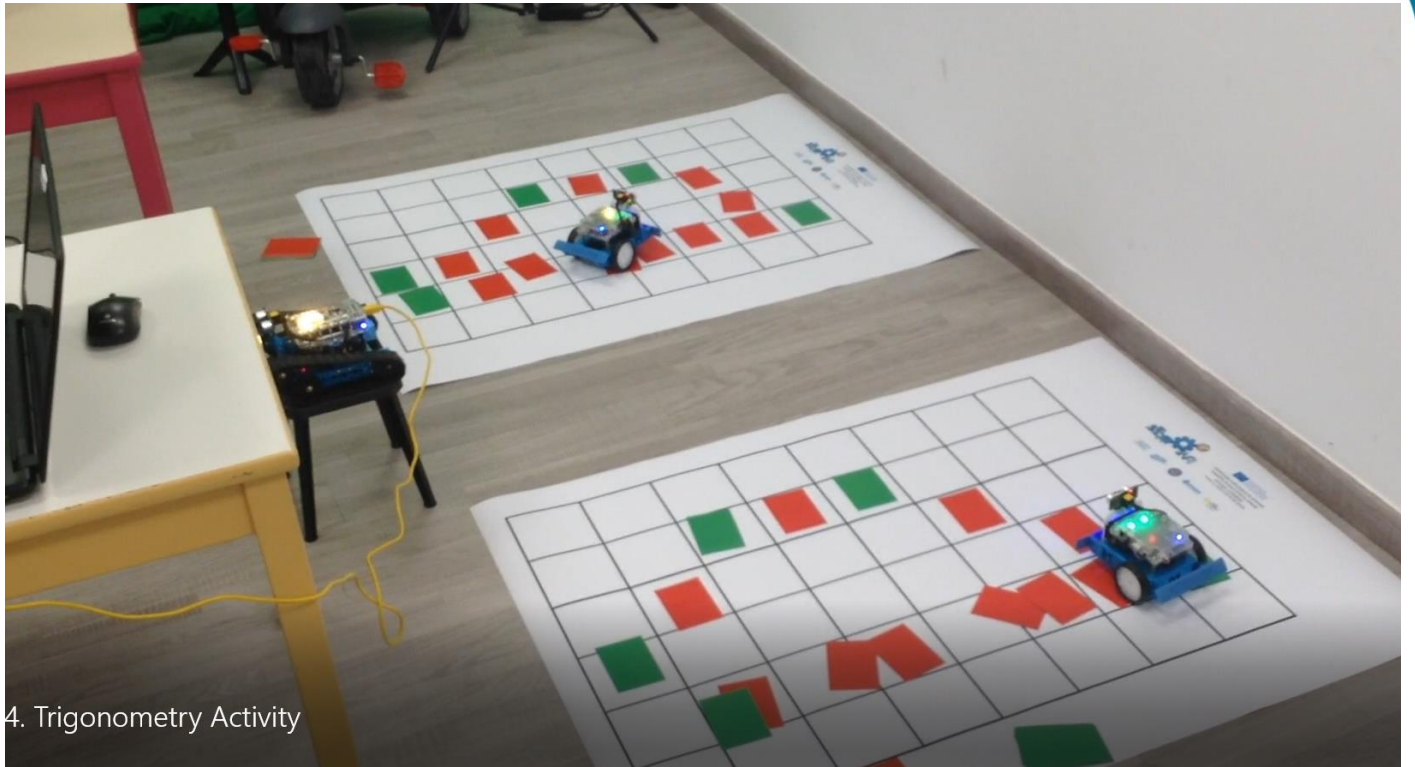
Auriga Program
forever
set led on board all red 0 green 0 blue 0
wait 1 secs
Minute0
wait 1 secs
Second30
wait 1 secs
Minute1
wait 1 secs
Minute1_30
wait 1 secs
Minute2
wait 1 secs
Minute2_30
wait 1 secs
Minute3
wait 1 secs
Minute3_30
wait 1 secs
Minute4
wait 1 secs
Minute4_30
wait 1 secs
Minute5
wait 1 secs
Minute5_30

```

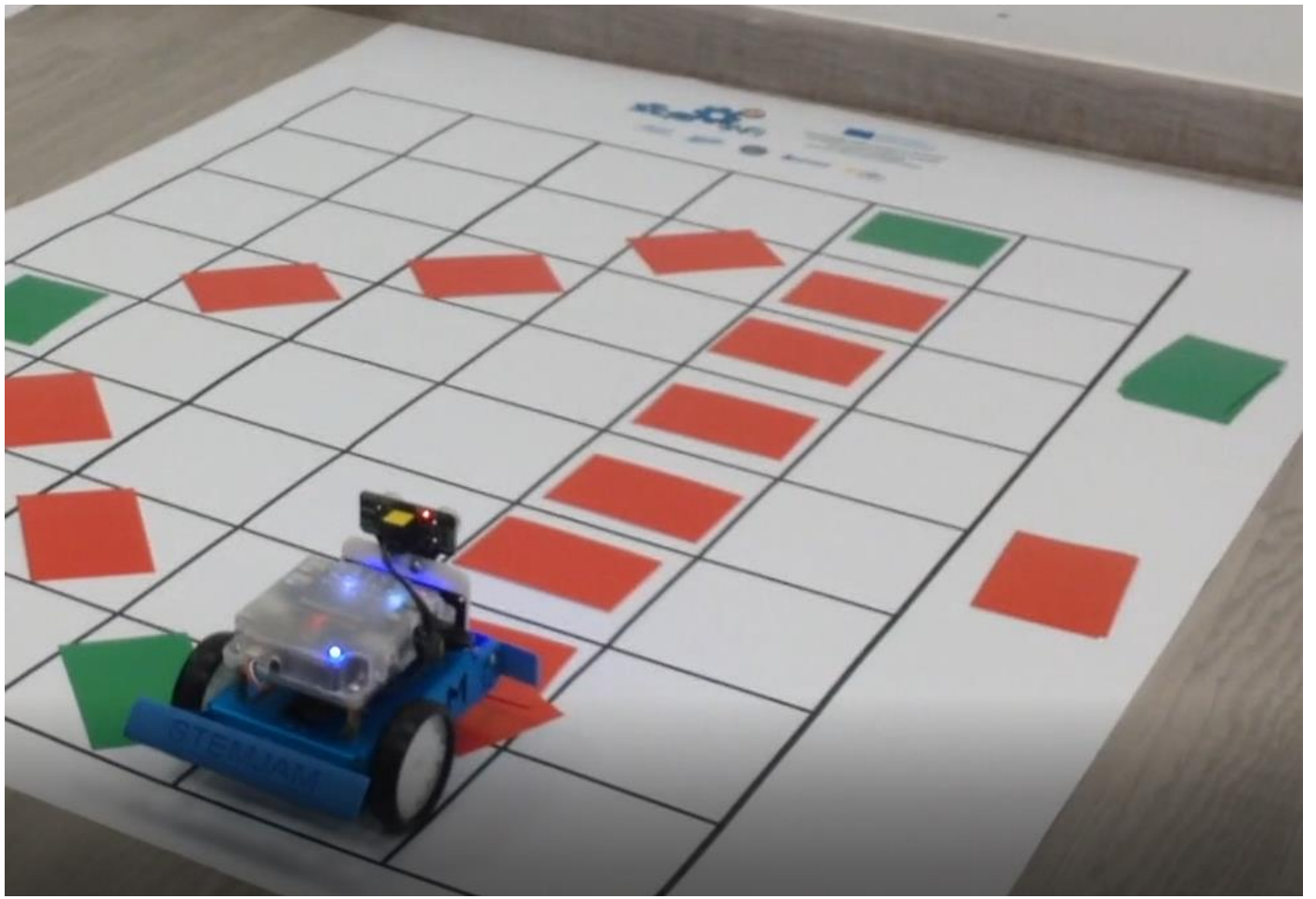
Para finalizar la descripción de la actividad, mostramos algunas imágenes de la misma:





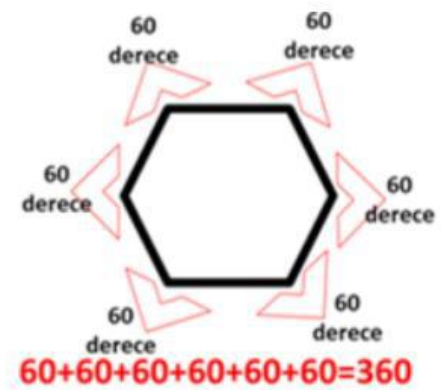
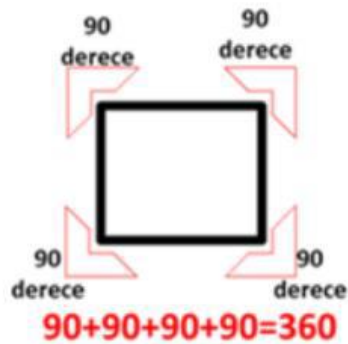
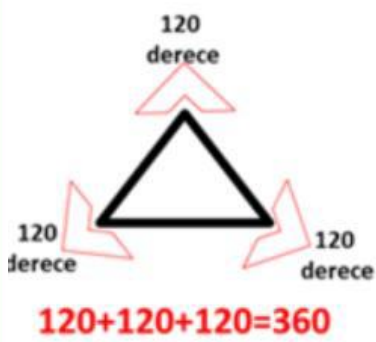


4. Trigonometry Activity



## Segunda Versión

Paso 1: Calculamos la distancia para mBot. Tomaremos el valor de velocidad como 100.



Paso 2: La programación de ángulo a eje, la mejora y la codificación de giro se realizan en el MBot.

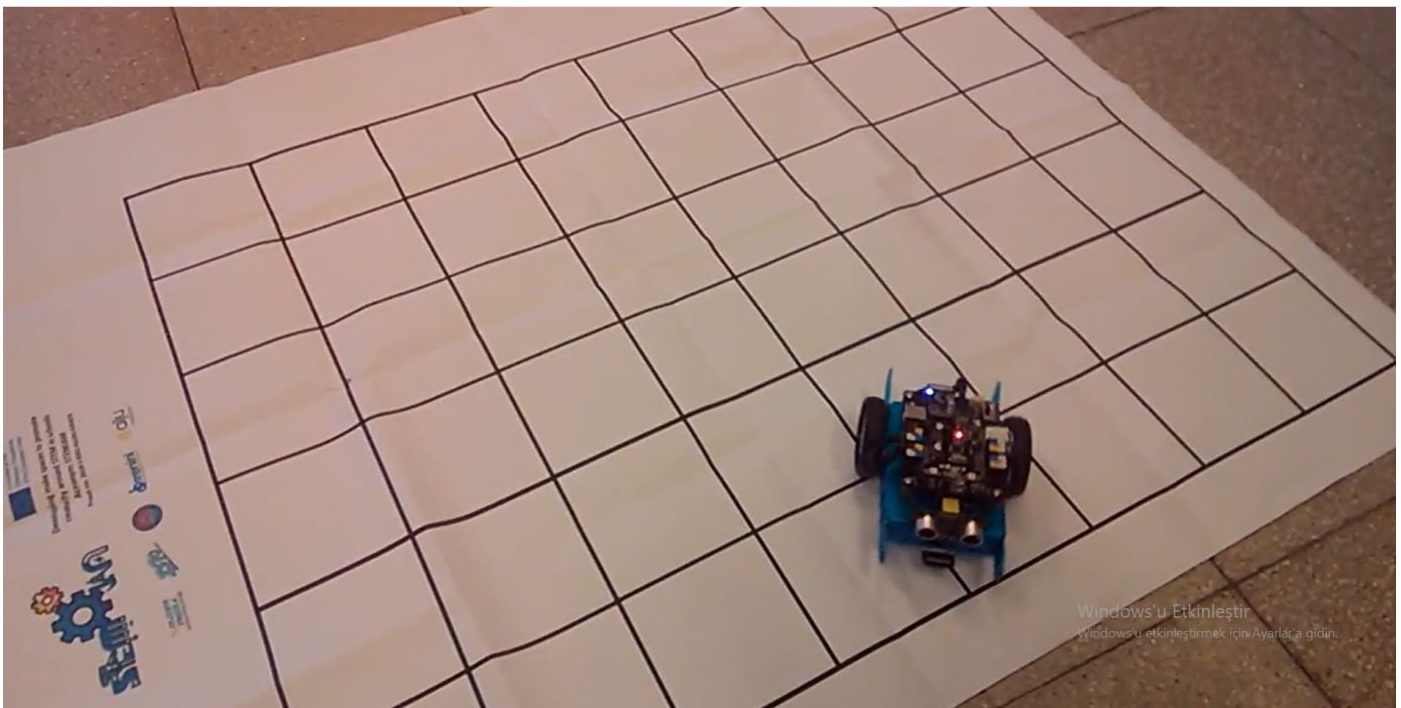
```
when I receive start
set numberofline to 0
ask How many gons there should there be? and wait
set numberofline to answer
ask How many cm per edge? and wait
repeat numberofline
  set go to answer * 0.75 / 50
  run forward at speed 100
  wait go secs
  set rotate to 360 / numberofline
  set motor M1 speed 100
  set motor M2 speed 0
  wait rotate * 1.2 / 90 secs
```

**Paso 3:** Escribimos fórmulas de cálculo de área y perímetro en mBot.

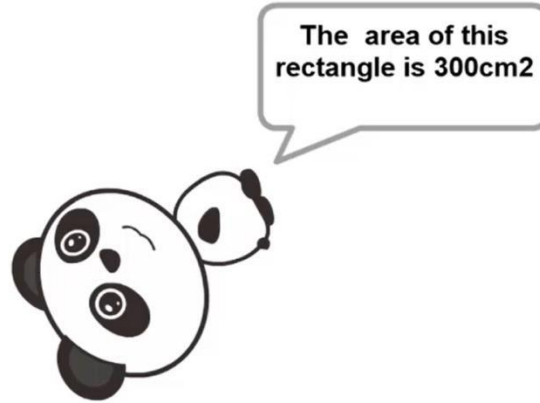
```
define Rectangle
ask Enter the length of the long side, and wait
set long edge to answer
ask Enter the length of the short side, and wait
set short edge to answer
say join The area of this rectangle is join long edge * short edge cm2 for 5 secs
say join The environment of this rectangle is join 2 * long edge + short edge cm for 5 secs

define Triangle
ask Enter the length of the first side, and wait
set line1_triangle to answer
ask Enter the length of the second side, and wait
set line2_triangle to answer
ask Enter the length of the third side, and wait
set line3triangle to answer
set s to line1_triangle + line2_triangle + line3triangle / 2
say join The area of this triangle is sqrt of s * s - line3triangle * s - line2_triangle * s - line1_triangle for 4 secs
say join The environment of this triangle is line1_triangle + line2_triangle + line3triangle for 4 secs
```

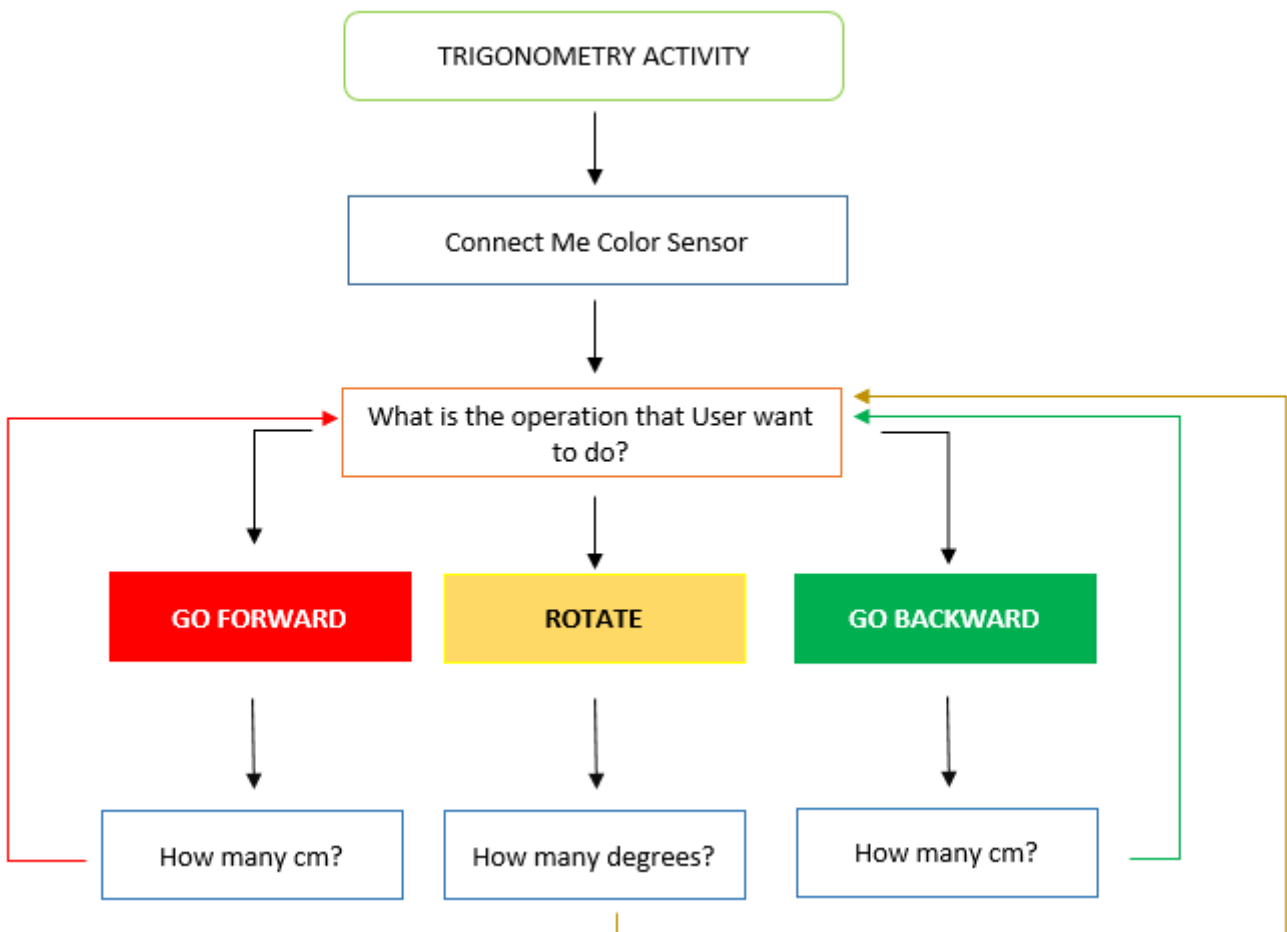
**Paso 4:** Realizamos movimientos geométricos mediante la cuadrícula.



Paso 5: El programa calcula el área del polígono y el perímetro y muestra el resultado en la pantalla.



### DIAGRAMA DE FLUJO



## EVALUACIÓN DEL ESTUDIANTE

Todos los estudiantes mejoraron sus conocimientos sobre formas geométricas y cálculo de áreas de polígonos.

## BIBLIOGRAFÍA

[https://www.makeblock.es/productos/sensor\\_color/](https://www.makeblock.es/productos/sensor_color/)

## MÁS INFORMACIÓN

### Primera Versión:

- 1)  $V = x / t$  y el tiempo y la distancia con mBot. La distancia se ajusta con el teclado, se formuló el tiempo / distancia y mBot se movió o realizó un giro. En esta actividad, se supone que el MBot toma un promedio de 12 cm en un segundo pero depende de la energía de la batería.
- 2) Colocamos el sensor de color en mBot, el color verde será para las esquinas y el rojo para los bordes.

### Segunda Versión:

- 1) Hemos trabajado geometría en esta actividad. Calculamos el ángulo exterior de un polígono equilátero. El usuario debe definir cuántos bordes y polígonos equiláteros deben preceder al MBot; y luego cuántos centímetros debe tener cada lado. El movimiento se obtiene de nuevo con la fórmula  $v = x / t$
- 2) En esta actividad, calculamos el área y la circunferencia de los cuerpos geométricos. Hemos definido la fórmula para cada objeto por separado.