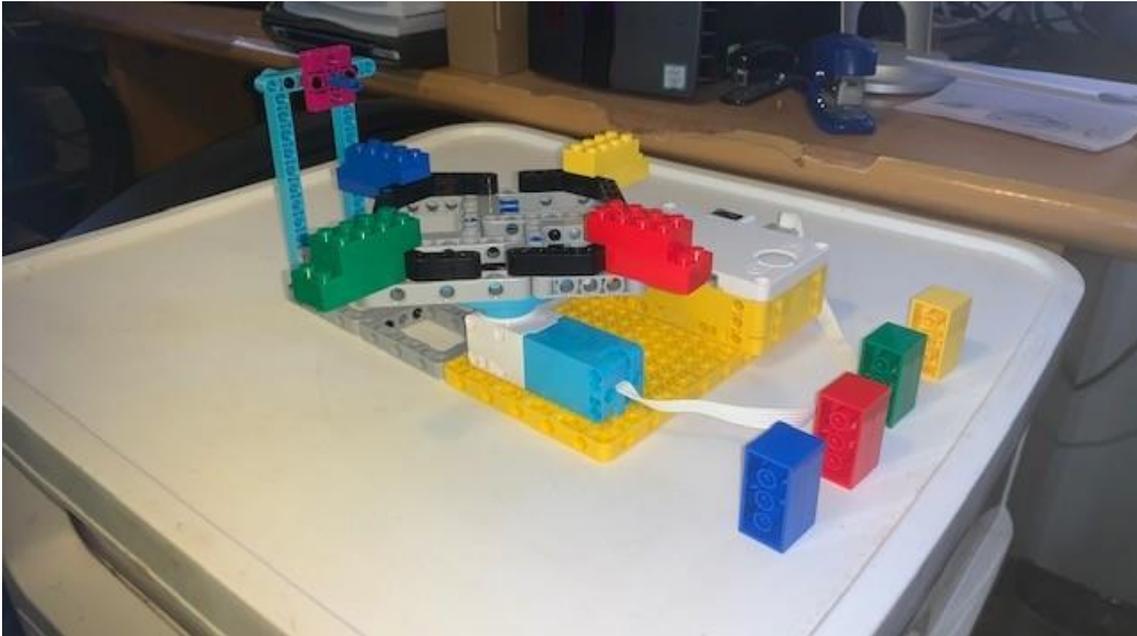


## Capítulo 27 - Memoriza los colores



Para este proyecto necesitamos un motor mediano y un sensor de color.

Vamos a crear un programa que nos pida 4 veces la lectura de color de 4 figuras, estas las podemos ir mostrando al sensor de color de forma aleatoria.

Vamos a crear un vector llamado colores para almacenar dichos colores, una vez ya ha leído los cuatro colores por mediación del motor mediano, nos tiene que mostrar el orden en que le mostramos al sensor de color dichos colores.

```
from spike import PrimeHub, LightMatrix, Button, StatusLight, ForceSensor, MotionSensor, Speaker, ColorSensor, App, DistanceSensor, Motor, MotorPair
from spike.control import wait_for_seconds, wait_until, Timer
from math import *
from spike.operator import not_equal_to
hub = PrimeHub()
Contador = 0
Colores = []
motor = Motor('F')
color = ColorSensor('A')
motor.set_default_speed(50)
motor.run_for_rotations(2)

for i in range(0,8):
    hub.speaker.beep(60,0.2)
    hub.light_matrix.show_image('ARROW_E')
    Colores.append(color.wait_for_new_color())
```

```

hub.light_matrix.off()

for i in range(0,4):
    Colores.remove(None)

for i in range(0,4):
    if Colores[i] == 'green':
        motor.run_to_position(0)
    if Colores[i] == 'blue':
        motor.run_to_position(270)
    if Colores[i] == 'yellow':
        motor.run_to_position(180)
    if Colores[i] == 'red':
        motor.run_to_position(90)
    wait_for_seconds(1)

```

Cuando hacemos un proyecto nuevo Python nos agrega el máximo de librerías, esto hace que el proyecto ocupe más memoria pero evitamos posible errores por no agregar todas las librerías que necesitamos.

Vamos a comentar el programa:

hub = PrimeHub() → Definimos un objeto llamado hub partiendo de la librería PrimeHub()

Contador = 0 → Definimos la variable Contador y la inicializamos a 0.

Colores = [] → Definimos un vector llamado Colores, no contiene ningún elemento.

Motor = Motor('F') → Le decimos que el motor está conectado en el puerto F.

Color = ColorSensor('A') → Le decimos que el sensor de color está conectado en el puerto A.

motor.set\_default\_speed(50) → configuramos la potencia del motor al 50%

motor.run\_for\_rotations(2) → El motor hace dos rotaciones.

for i in range(0,8) → Hacemos un bucle que se repite 8 veces.

hub.speaker.beep(60, 0.2) → Se reproducirá un sonido Nota 60 durante 0,2 segundos.

hub.light\_matrix.show\_image('ARROW\_E') → Muestra pixels Hub flecha hacia derecha.

Colores.append(color.wait\_for\_new\_color()) → Agrega un color en el vector Colores.

Hub.light\_matrix.off() → Apaga los pixeles de Hub.

### **Comentario:**

Cuando añadíamos los colores al vector Colores, este espera hasta un cambio de color, por este motivo de los 8 colores encontramos intercalados None, ningún color con el siguiente bucle eliminamos los colores None y nos quedará los colores que nos interesan.

```
for i in range(0,4):  
    Colores.remove(None)
```

En el siguiente bucle que se repite 4 veces le decimos que si Colores[i] 'i' posición del vector desde 0 hasta 3 tiene un determinado color el motor se posiciona a unos determinados grados.

```
for i in range(0,4):  
    if Colores[i] == 'green':  
        motor.run_to_position(0)  
    if Colores[i] == 'blue':  
        motor.run_to_position(270)  
    if Colores[i] == 'yellow':  
        motor.run_to_position(180)  
    if Colores[i] == 'red':  
        motor.run_to_position(90)  
    wait_for_seconds(1)
```

Si lo ejecutas verás que el resultado es el mismo.