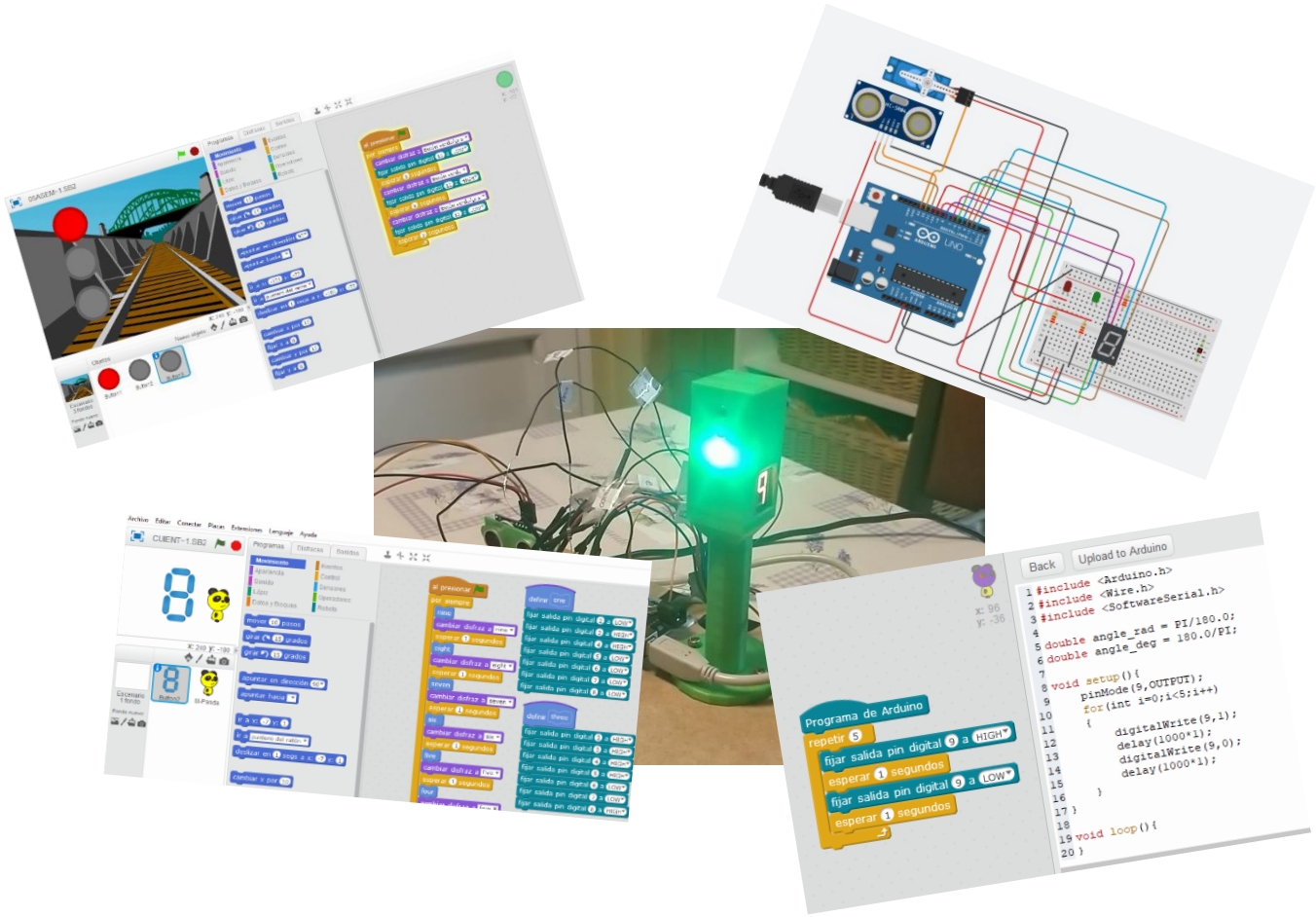
EJERCICIOS DE MBLOCK PARA APRENDER A PROGRAMAR CON

"EL ENTRENADOR DE PROGRAMACIÓN"

LA PLACA ARDUINO



DISPLAY							LEDES		ULTRASONIDOS		SERVOMOTOR
a	b	c	d	e	f	g	R	V	TRIGER	ECHO	
2	3	4	5	6	7	8	9	10	11	12	13

SANTIAGO PÉREZ ANTOLÍN

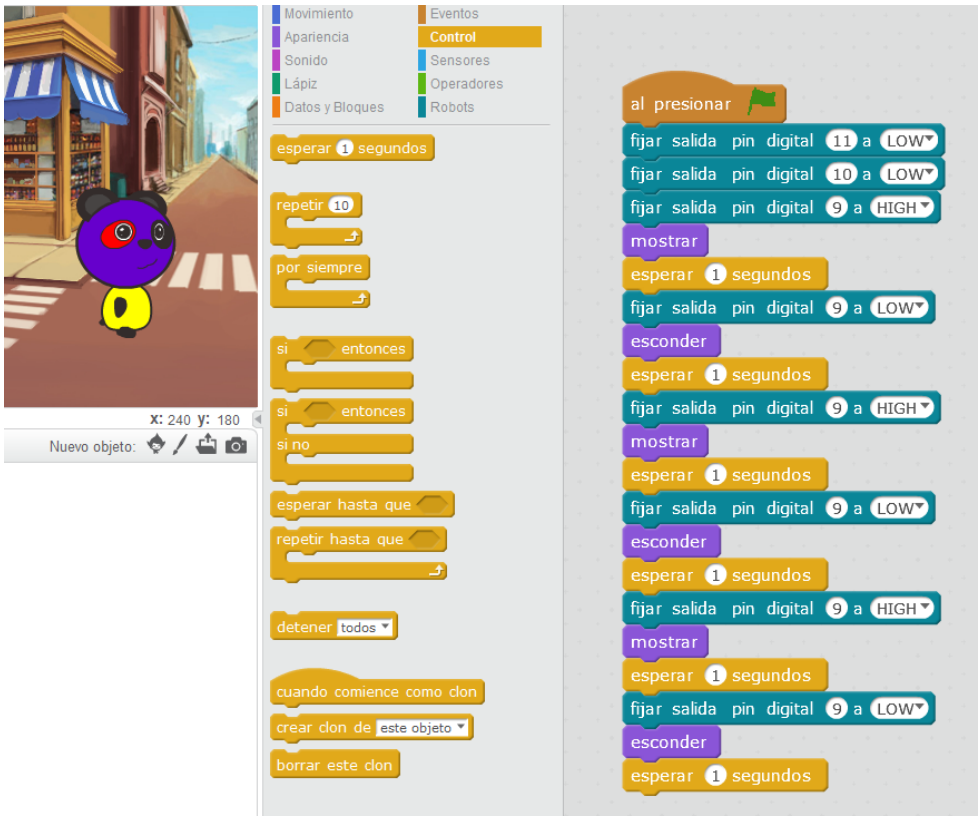
CEO BOECILLO

ÍNDICE

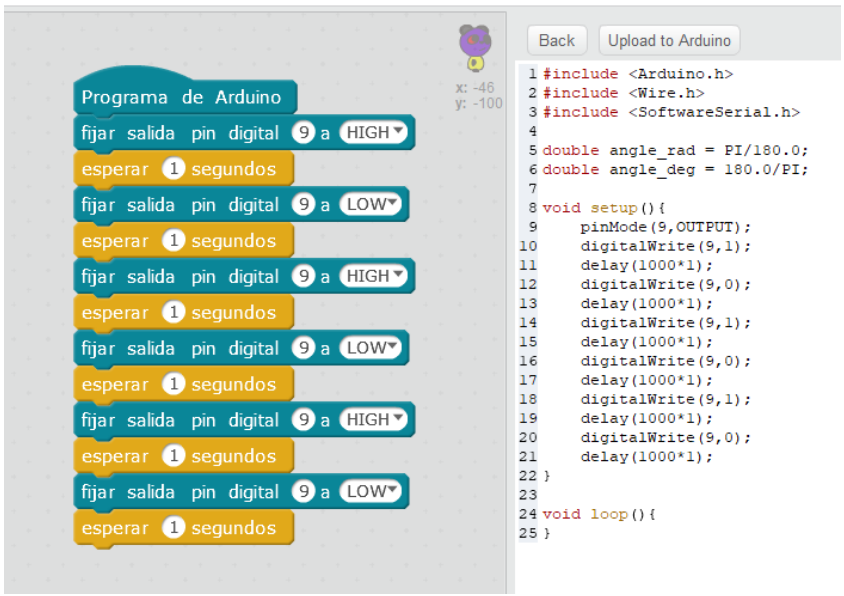
Contenido

CONEXIONES: Pin 9: ROJO Pin 10: AMARILLO Pin 11: VERDE

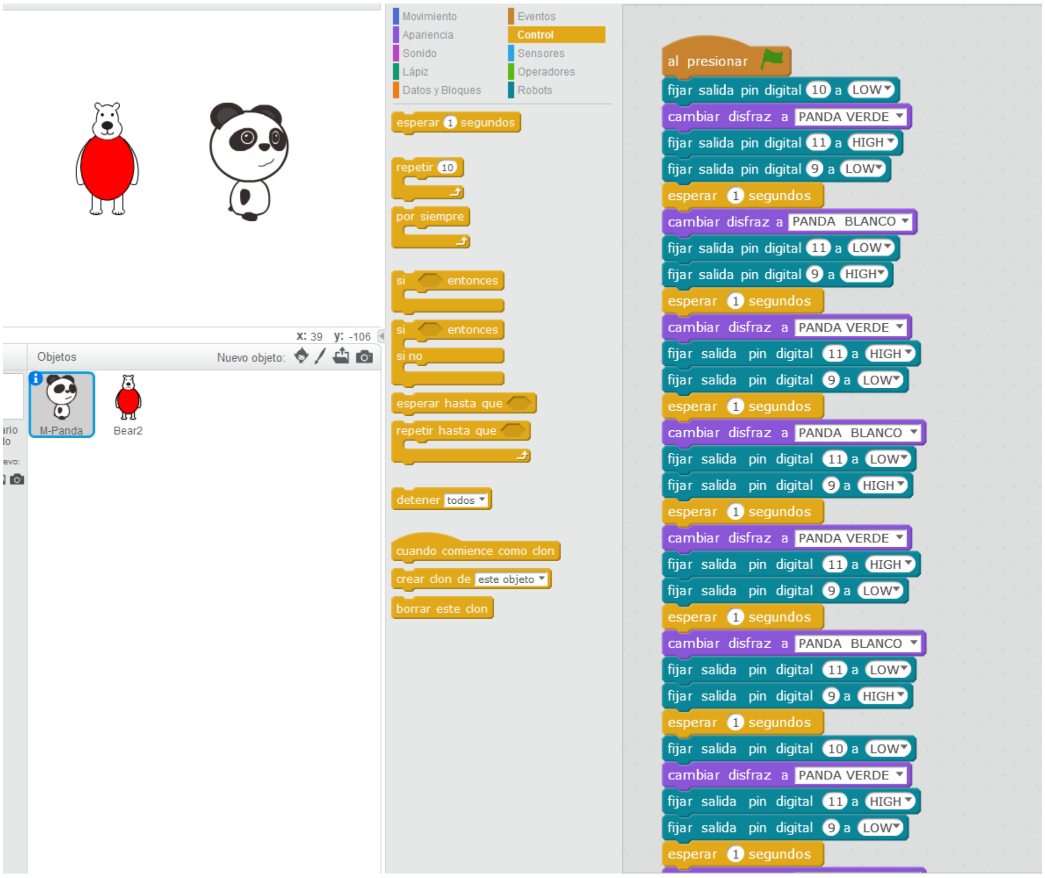
1.- Encender y apagar un LED tres veces.

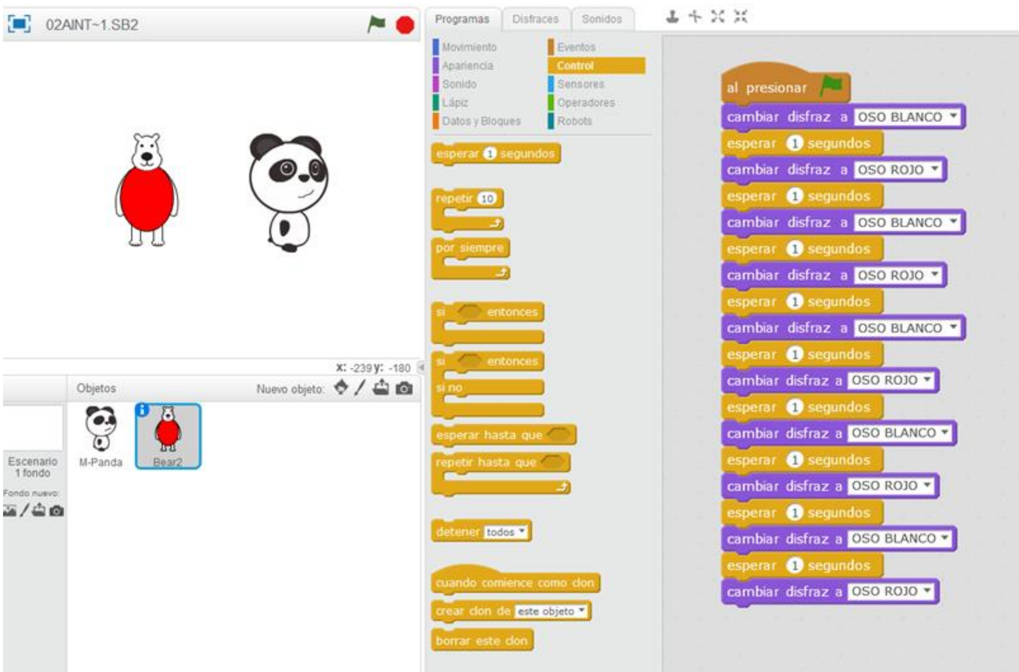


MODO ARDUINO

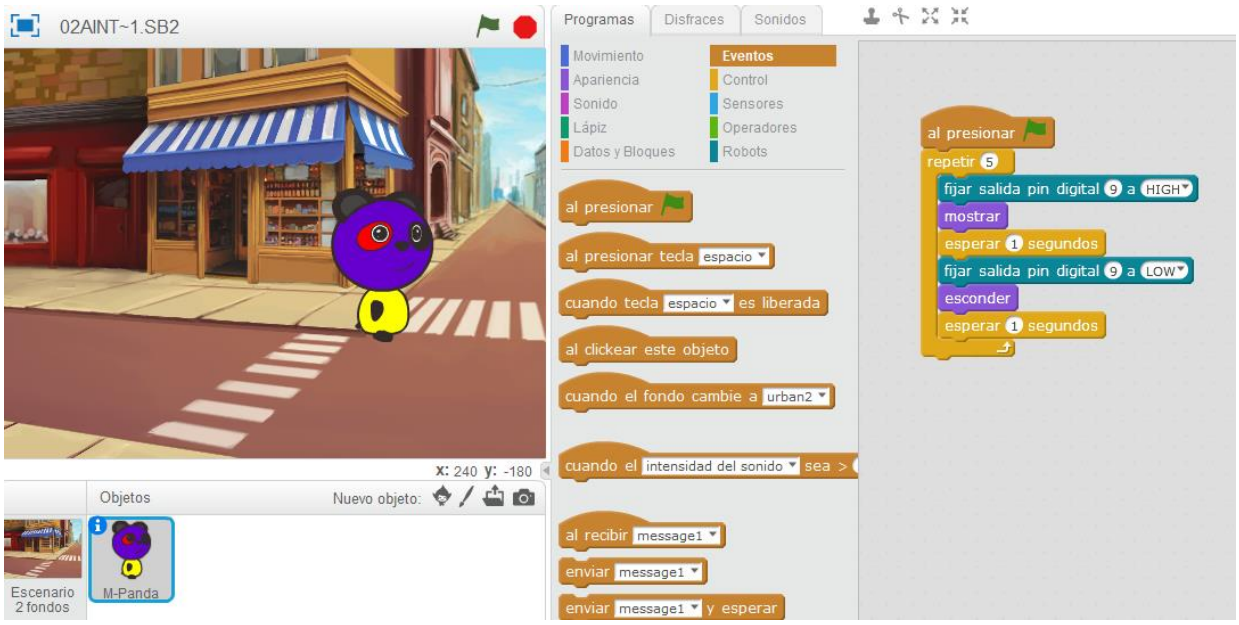


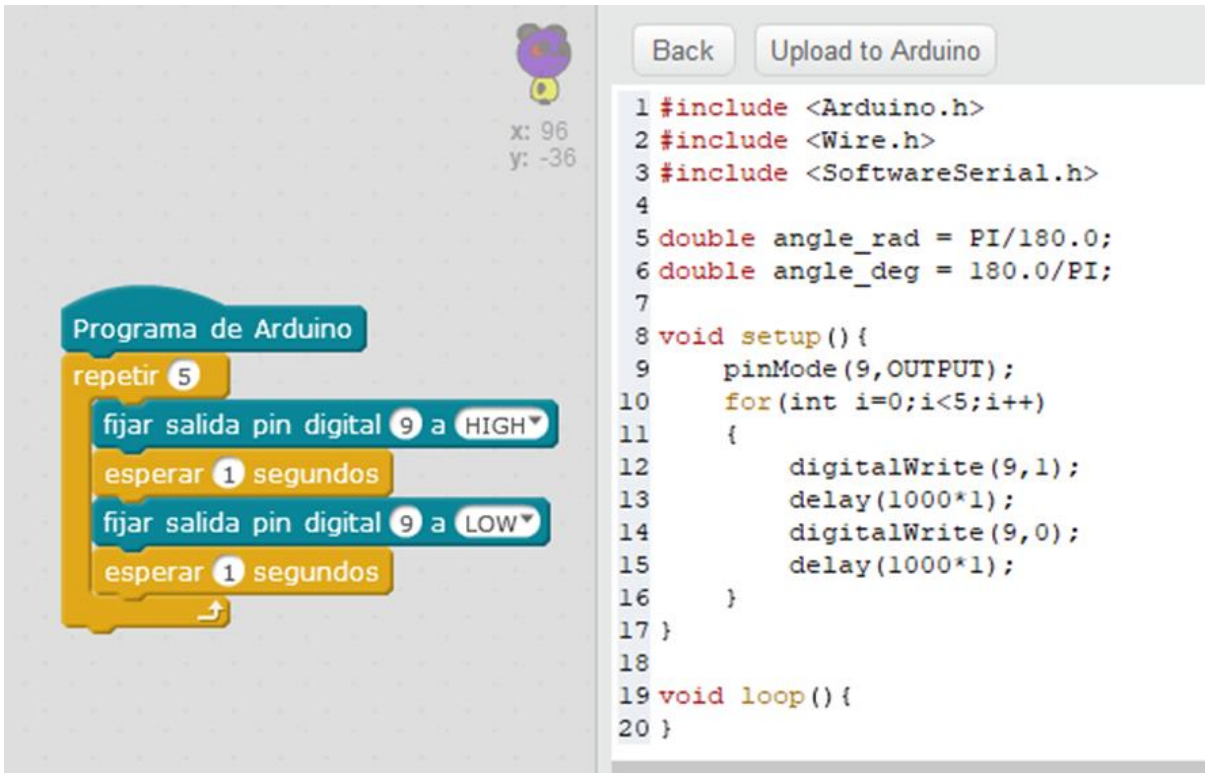
2.- Encender y apagar dos LEDs intermitentemente cinco veces.



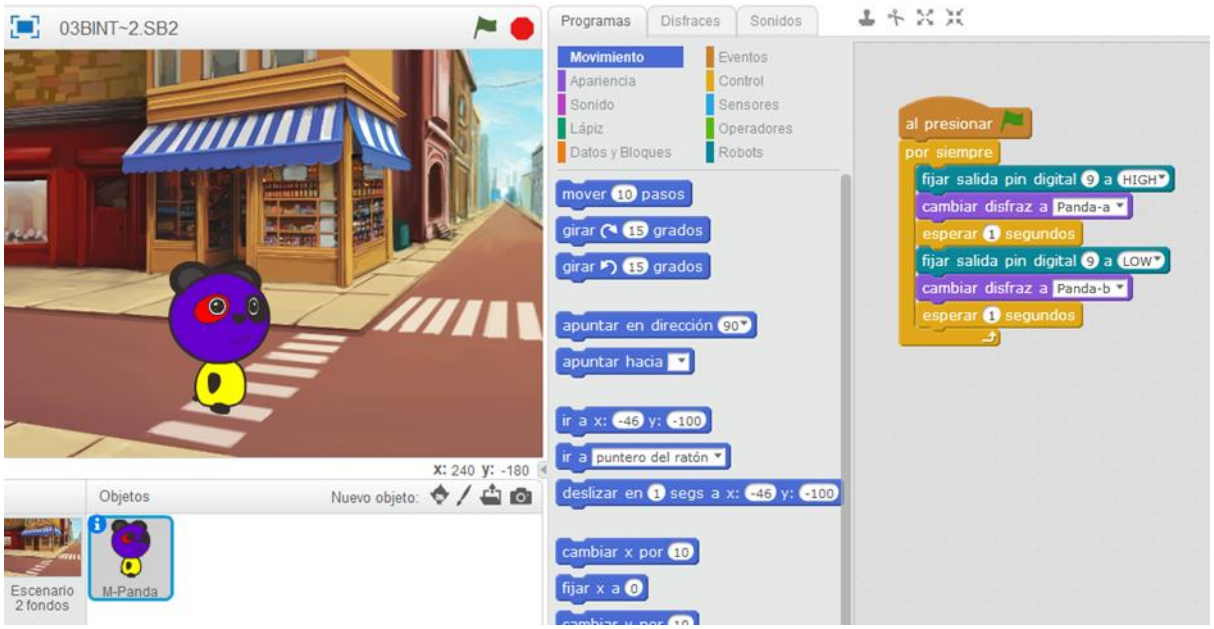


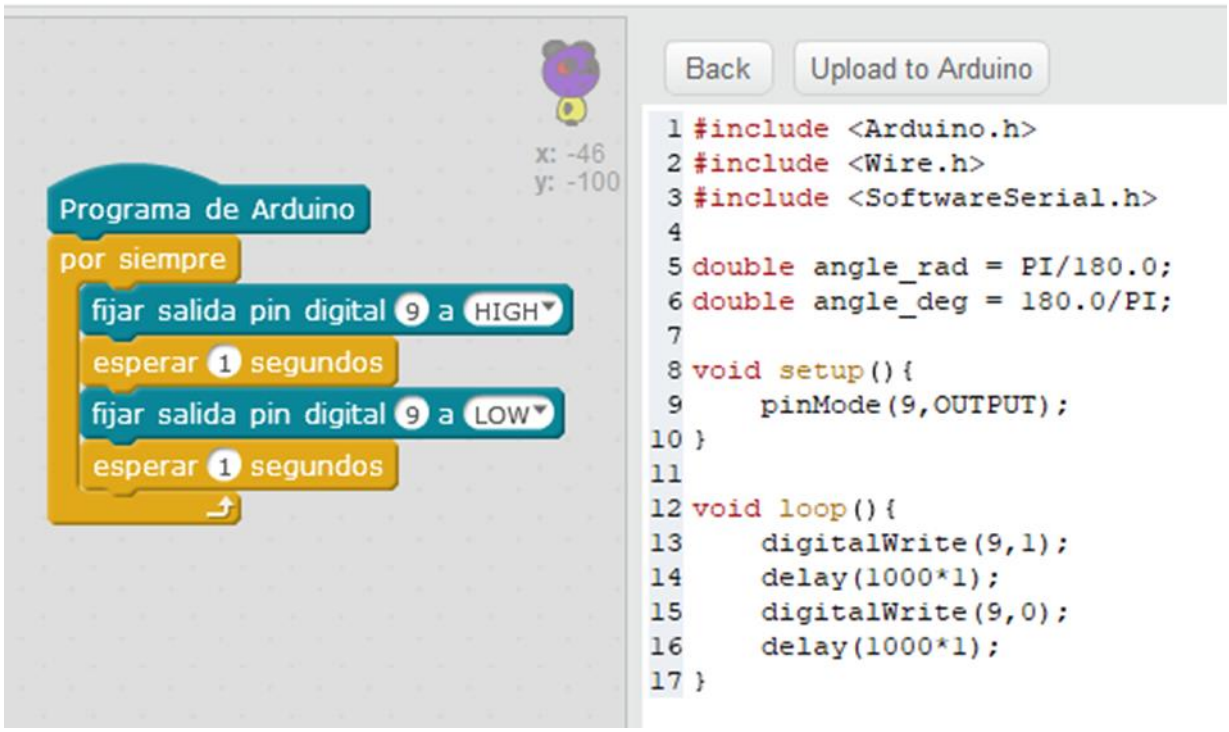
3.- Encender y apagar un LED cinco veces intermitentemente con la función FOR.





4.- Encender y apagar un LED todo el tiempo. Función LOOP.





5.- Programar el funcionamiento de un SEMÁFORO de tres LEDES.

A.- Creando tres objetos que simularán el funcionamiento de cada LED.

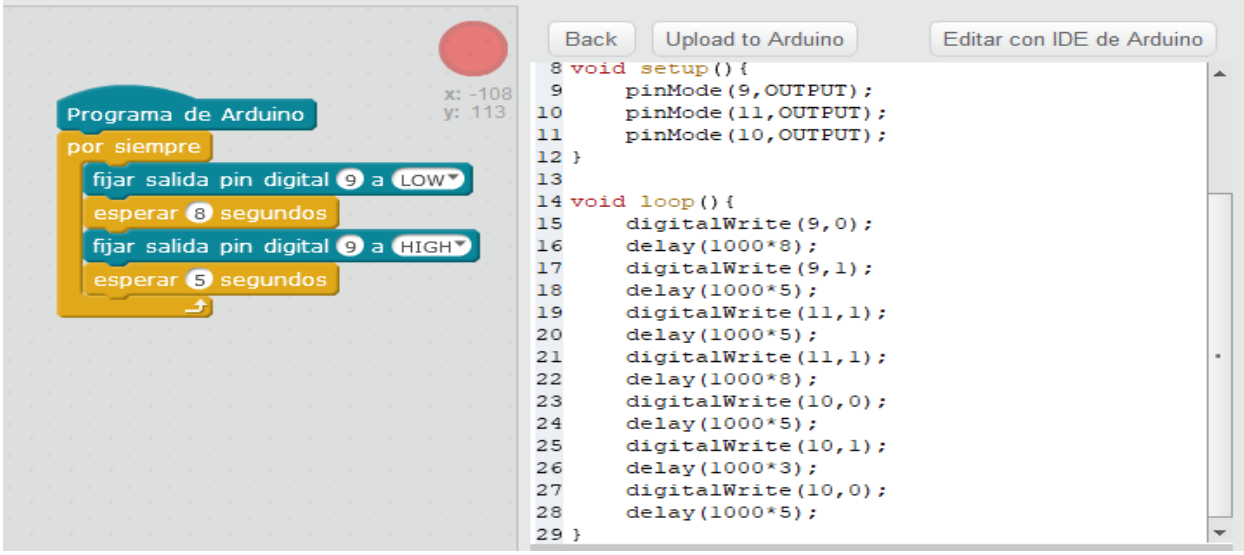
TIEMPO	COCHES		
	ROJO	NARANJA	VERDE
	PIN 9	PIN 10	PIN 11
5 segundos	HIGH	LOW	LOW
5 segundos	LOW	LOW	HIGH
3 segundos	LOW	HIGH	LOW

Ayuda: para un objeto la programación sería:

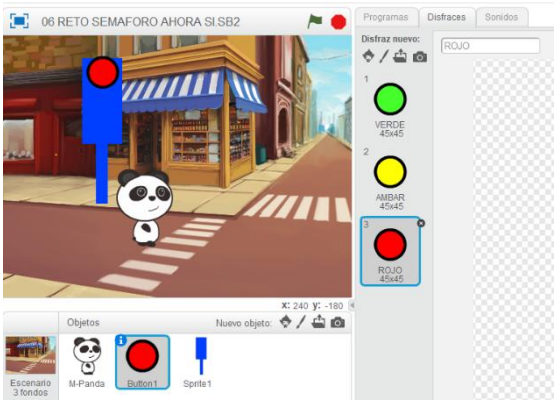
El código programado en ARDUINO sería.

Tendremos que eliminar de los tres objetos los bloques morados(apariencia).

Si vamos a: EDITAR \ MODO ARDUINO, nos aparecerá:

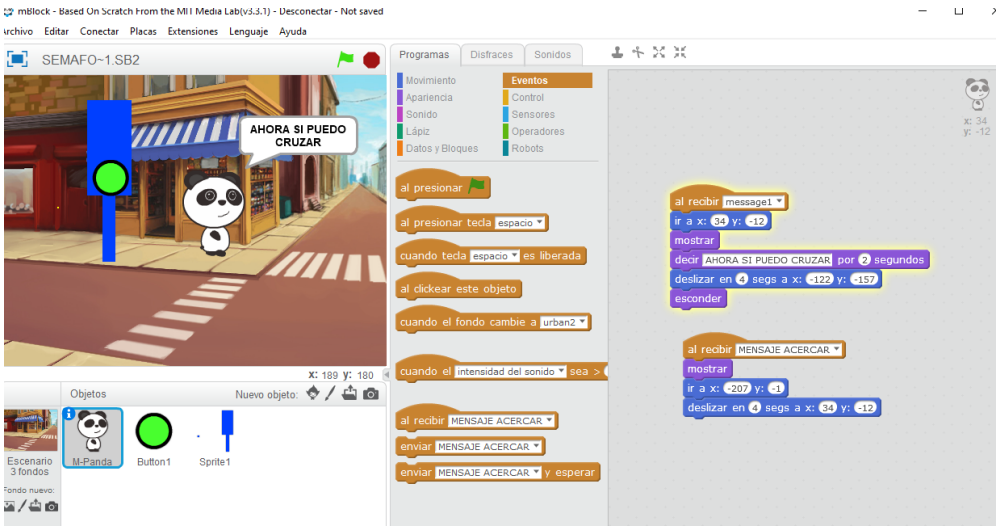


B.- Realiza el programa creando solo un objeto y a partir de él sus disfraces con los tres colores.



RETO: Crea un objeto (PANDA) y un escenario. El panda deberá cruzar la calle cuando el semáforo de peatones esté en verde y deberá esperar cuando esté en rojo. Añadir que haga un comentario.

Ayuda:



RETO: Programa un semáforo de 5 ledes (3 coches y 2 peatones).

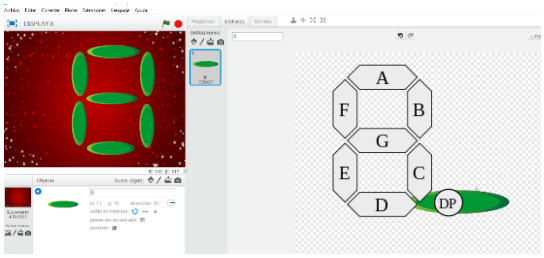
PROGRAMACIÓN SEGÚN TABLA

TIEMPO	COCHES			PEATONES	
	ROJO	NARANJA	VERDE	ROJO	VERDE
	PIN 9	PIN 10	PIN 11	PIN 12	PIN 13
5 segundos	HIGH	LOW	LOW	LOW	HIGH
5 segundos	LOW	LOW	HIGH	HIGH	LOW
3 segundos	LOW	HIGH	LOW	HIGH	LOW

6.- Programar la cuenta atrás de un display.

6.1.- Programación en paralelo:

A.- En Mblock diseña un objeto que se semeje a un display de siete segmentos, a continuación, duplica ese objeto seis veces y haz que cada objeto simule un led del display. Nombra los objetos con las letras correspondientes a las del display. También crea un fondo.



B.- Completa la tabla indicando el estado en el que tiene que estar cada led (Objeto) a lo largo de la cuenta atrás. (1 = encendido, 0 = apagado). (1 punto).

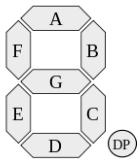
		Pin 2	Pin 3	Pin 4	Pin 5	Pin 6	Pin 7	Pin 8	
TIEMPO	VER	a	b	c	d	e	f	g	
0-1 seg	9	1	1	1	0	0	1	1	
1-2	8	1	1	1	1	1	1	1	
2-3	7	1	1	1	0	0	0	0	
3-4	6	1	0	1	1	1	1	1	
4-5	5	1	0	1	1	0	1	1	
5-6	4	1	1	0	0	0	1	1	
6-7	3	1	1	1	1	0	0	1	
7-8	2	1	1	0	1	1	0	1	
8-9	1	0	1	1	0	0	0	0	
9-10	0	1	1	1	1	1	1	0	

C.- Realiza la programación de los objetos para la cuenta atrás.

RETO: Después de la cuenta crea una animación en la que despegue un cohete de otro escenario, aterrice en otro escenario y salga otro objeto, de la nave, que deberá hacer un comentario.

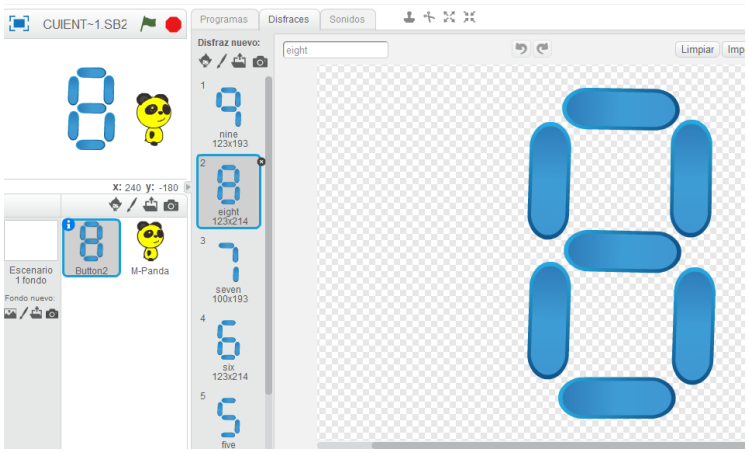
6.2.- Programación del display con un solo objeto y disfraces:

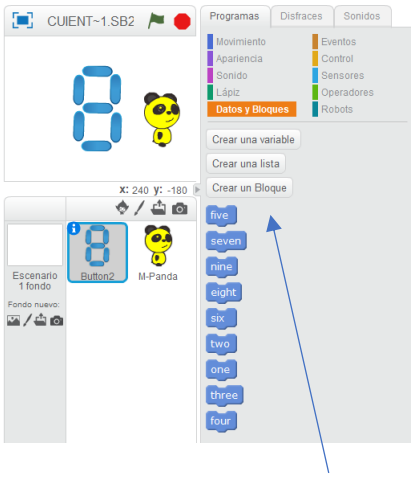
A.- Completa la tabla de los pines de conexión.



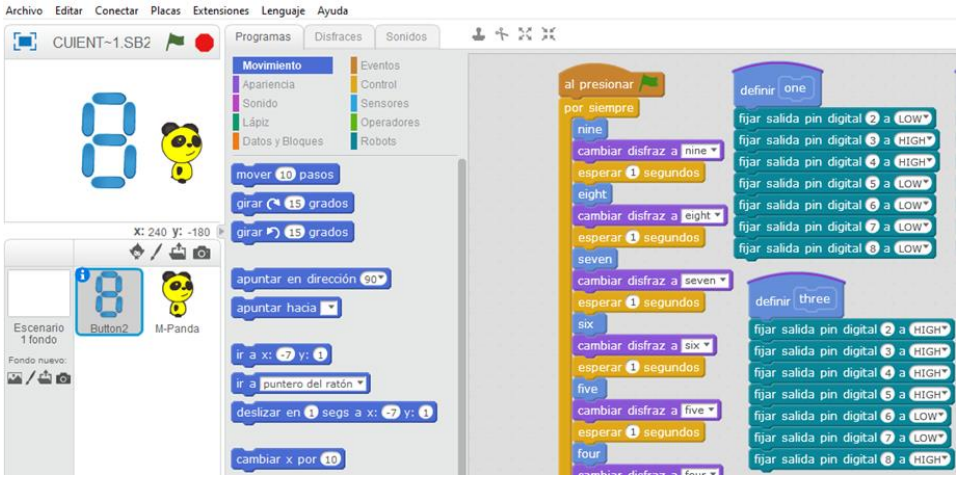
	Pin 2	Pin 3	Pin 4	Pin 5	Pin 6	Pin 7	Pin 8	
VER	a	b	c	d	e	f	g	
9								
8								
7								
6								
5								
4								
3								
2								
1								
0								

B.- Crea los disfraces.





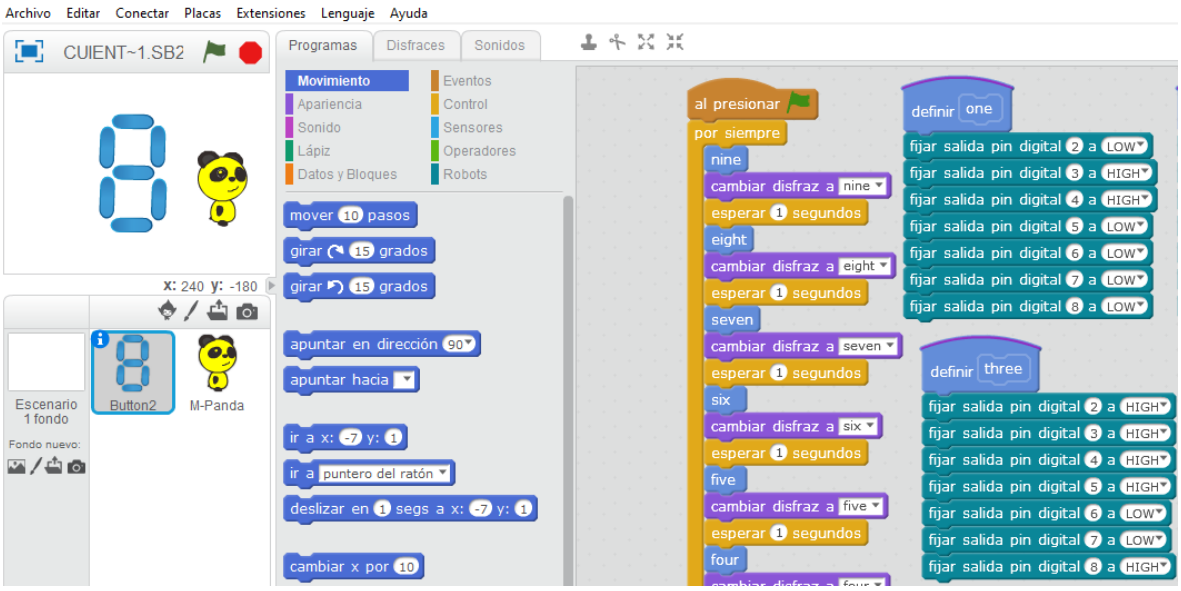
C.- Crea los Bloques de cada número y prográmalo para que funcione a la vez con Arduino.

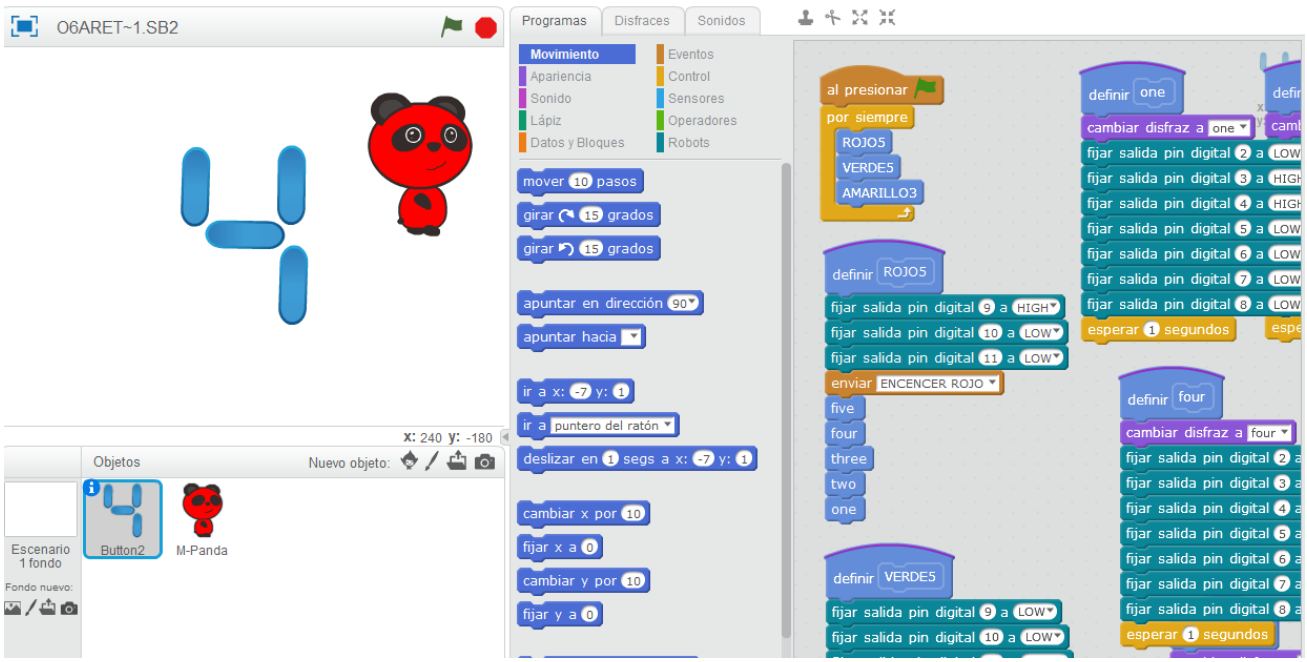


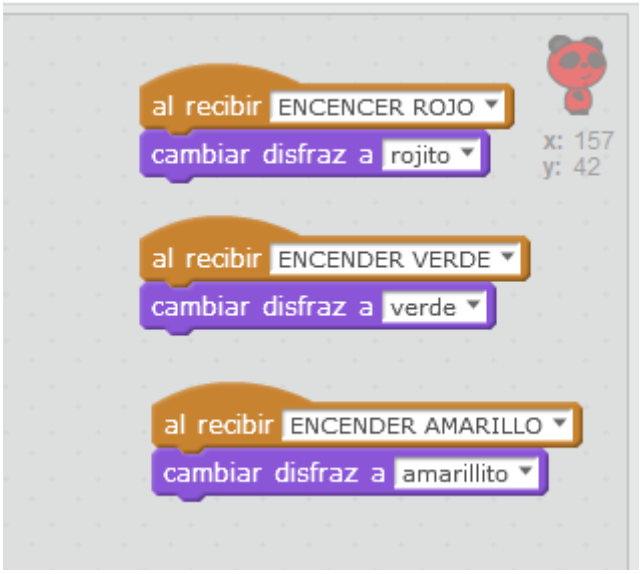
7.- Reto:

Realiza el programa de un semáforo en el que se indique mediante un DISPLAY la cuenta atrás de lo que dura cada color. Para ello una vez que tengas creado los bloques de cada número, crea tres bloques que contengan, cada uno, la programación de los LEDES y la cuenta atrás. Finalmente estos tres bloques los pondrás dentro de un POR SIEMPRE.

SOLUCIÓN:

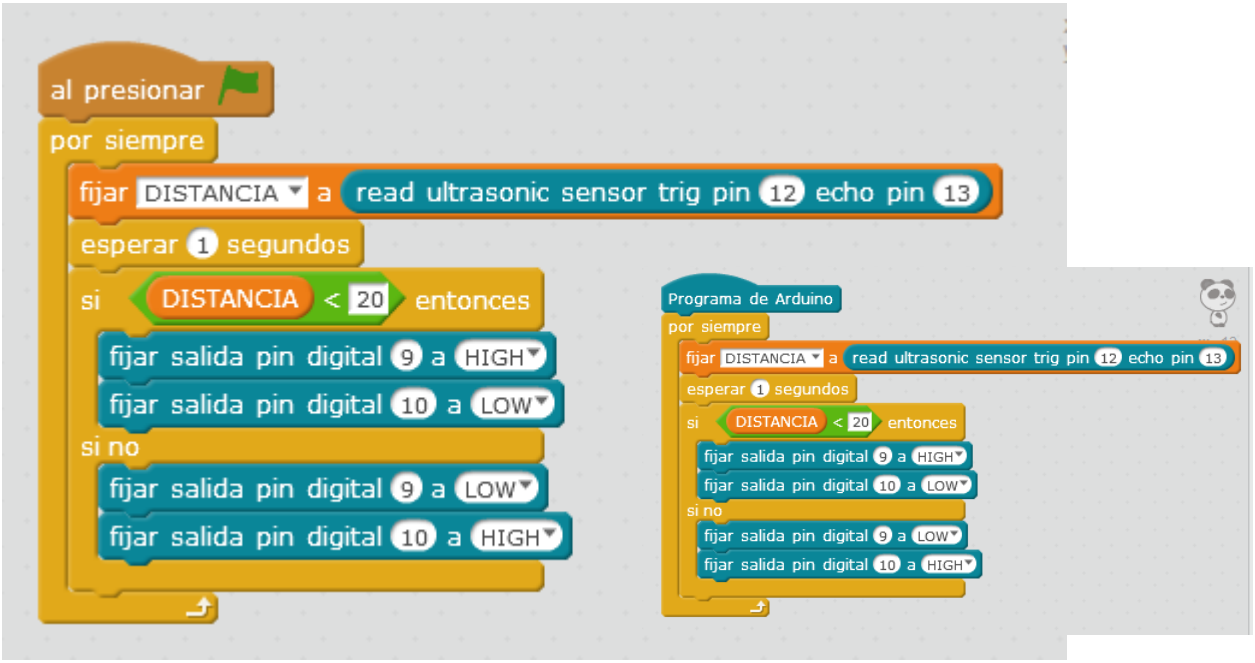





al recibir ENCENCER ROJO
cambiar disfraz a rojito
x: 157
y: 42
al recibir ENCENDER VERDE
cambiar disfraz a verde
al recibir ENCENDER AMARILLO
cambiar disfraz a amarillito

8 A.- Programar el SENSOR DE ULTRASONIDOS:

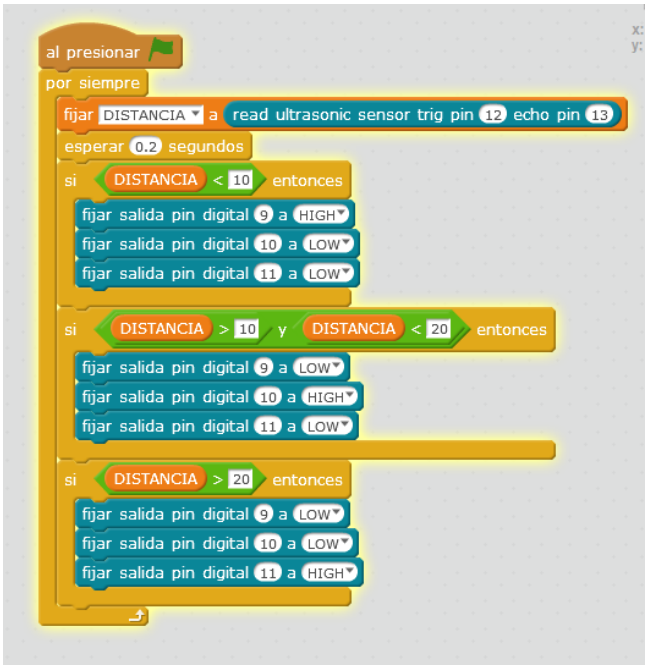
Para que en función de la distancia de un objeto a la barrera cambie el color de dos LEDES del semáforo. **IF…ELSE.** Tendrás que crear una variable (una caja en la que almacenarás los valores de lectura del ultrasonidos a la que llamarás “DISTANCIA”)

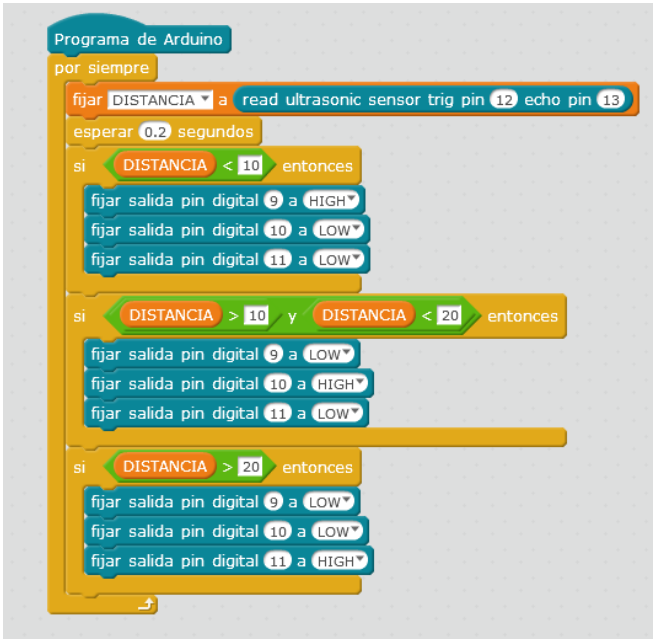



```
#include <Arduino.h>
#include <Wire.h>
#include <SoftwareSerial.h>

double angle_rad = PI/180.0;
double angle_deg = 180.0/PI;
double DISTANCIA;
float getDistance(int trig,int echo){
    pinMode(trig,OUTPUT);
    digitalWrite(trig,LOW);
    delayMicroseconds(2);
    digitalWrite(trig,HIGH);
    delayMicroseconds(10);
    pinMode(echo, INPUT);
    return pulseIn(echo,HIGH,30000)/58.0;
}

void setup(){
    pinMode(9,OUTPUT);
    pinMode(10,OUTPUT);
}
```

```
void loop(){
    DISTANCIA = getDistance(12,13);
    delay(1000*1);
    if((DISTANCIA) < (20)){
        digitalWrite(9,1);
        digitalWrite(10,0);
    }else{
        digitalWrite(9,0);
        digitalWrite(10,1);
    }
}
```

8 B.- Programar el ULTRASONIDOS para que en función de la distancia de un objeto a la barrera cambie el color de tres LEDES del semáforo. **IF , & y operadores de comparación.**





sketch_mar02a §

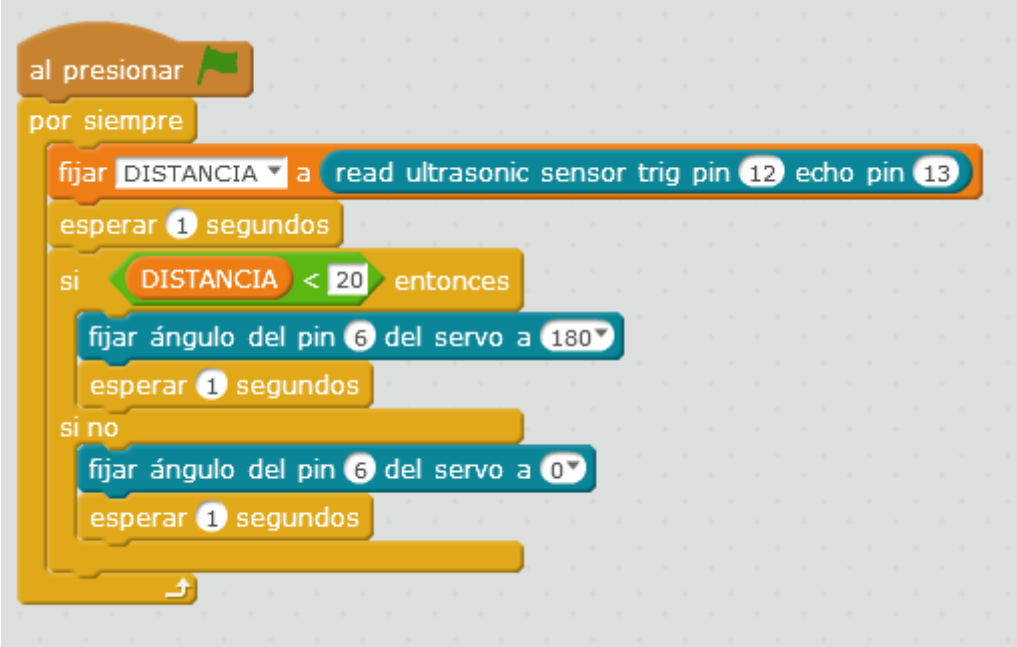
```
#include <Arduino.h>
#include <Wire.h>
#include <SoftwareSerial.h>

double angle_rad = PI/180.0;
double angle_deg = 180.0/PI;
double DISTANCIA;
float getDistance(int trig,int echo){
    pinMode(trig,OUTPUT);
    digitalWrite(trig,LOW);
    delayMicroseconds(2);
    digitalWrite(trig,HIGH);
    delayMicroseconds(10);
    pinMode(echo, INPUT);
    return pulseIn(echo,HIGH,30000)/58.0;
}

void setup(){
    pinMode(9,OUTPUT);
    pinMode(10,OUTPUT);
    pinMode(11,OUTPUT);
}
```

```
void loop(){
    DISTANCIA = getDistance(12,13);
    delay(1000*0.2);
    if((DISTANCIA) < (10)){
        digitalWrite(9,1);
        digitalWrite(10,0);
        digitalWrite(11,0);
    }
    if(((DISTANCIA) > (10)) & ((DISTANCIA) < (20))){
        digitalWrite(9,0);
        digitalWrite(10,1);
        digitalWrite(11,0);
    }
    if((DISTANCIA) > (20)){
        digitalWrite(9,0);
        digitalWrite(10,0);
        digitalWrite(11,1);
    }
}
```

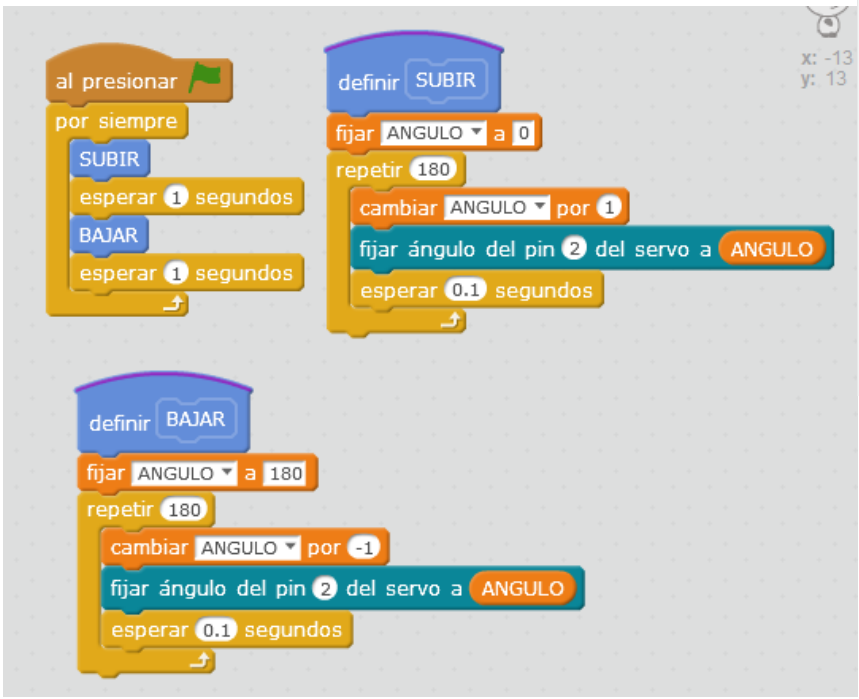
9.- Programa un SERVOMOTOR:

9.A.- Servomotor sube y baja utilizando una variable.

9.B.- Servomotor suba y baje en función de unas distancias.



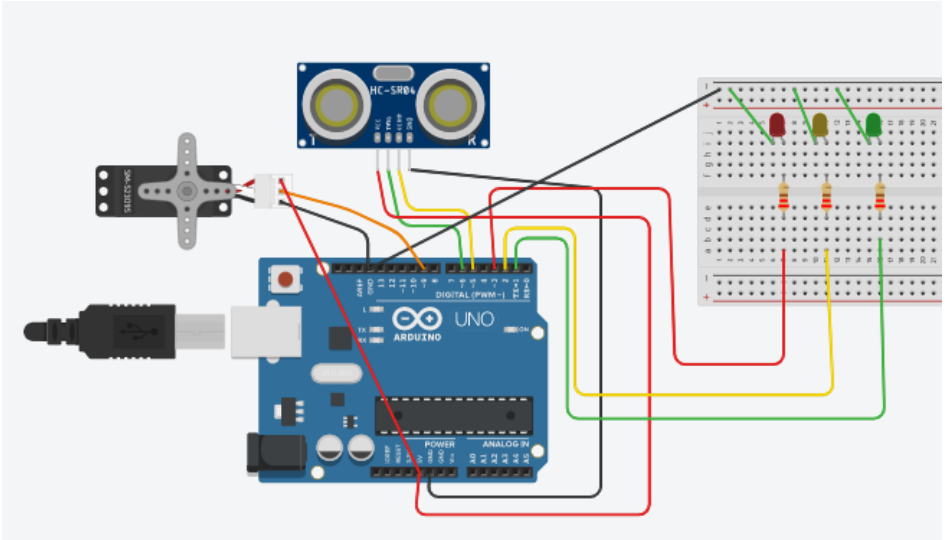
9.C.- Programa un SERVOMOTOR para que una barrera de aparcamiento suba y baje. FOR.




```
#include <Servo.h>

double angle_rad = PI/180.0;
double angle_deg = 180.0/PI;
void SUBIR();
double ANGULO;
Servo servo_2;
void BAJAR();

void SUBIR()
{
    ANGULO = 0;
    for(int i=0;i<180;i++)
    {
        ANGULO += 1;
        servo_2.write(ANGULO);
        delay(1000*0.1);
    }
}

void BAJAR()
{
    ANGULO = 180;
    for(int i=0;i<180;i++)
    {
        ANGULO += -1;
        servo_2.write(ANGULO);
        delay(1000*0.1);
    }
}

void setup(){
    servo_2.attach(2);
}

void loop(){
    SUBIR();
    delay(1000*1);
    BAJAR();
    delay(1000*1);
}
```

11.- Retos: Barrera de aparcamiento automática.

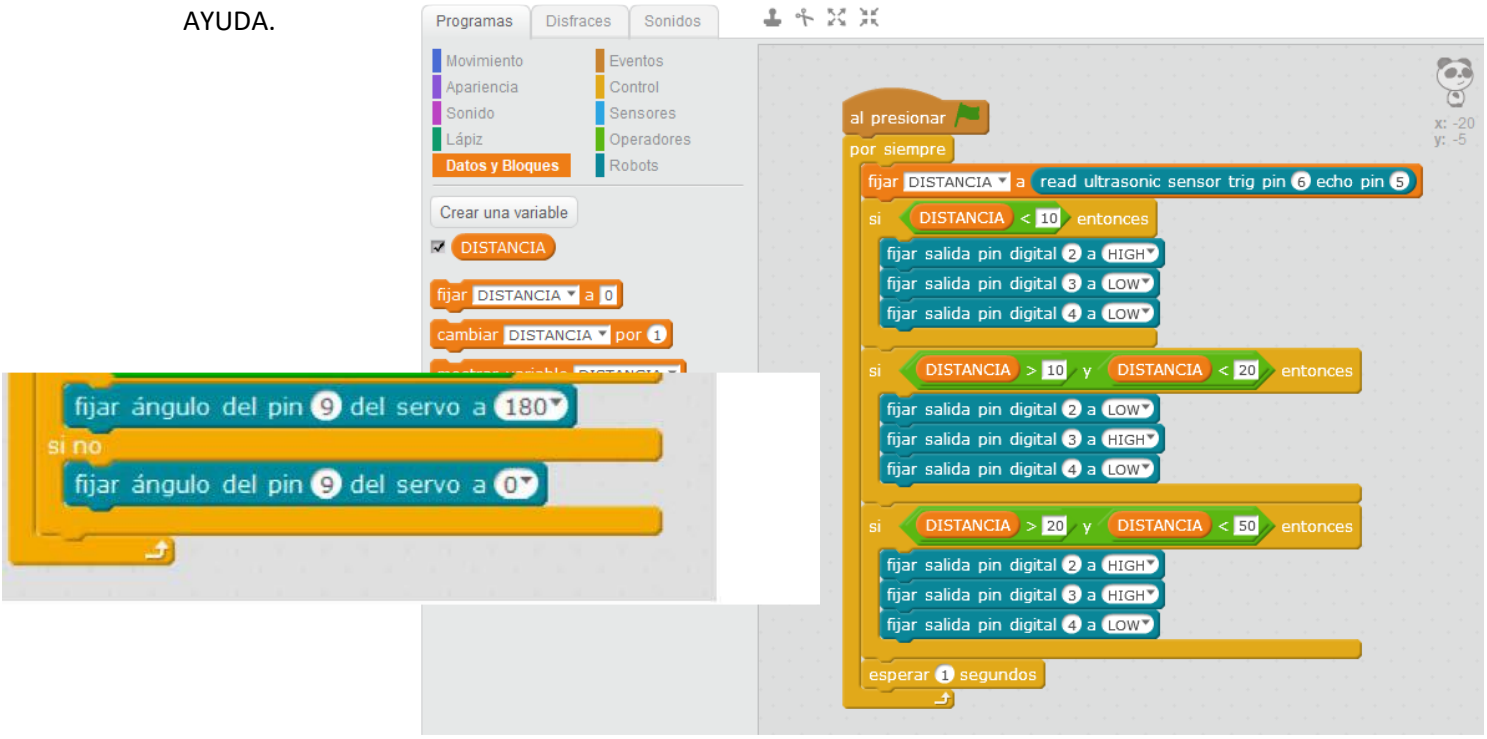
RETO A.- Realiza un Sketch en Mblock que simule el acceso a un parking en el que cuando no haya un objeto a menos de 10 cm de sensor de ultrasonidos el semáforo permanecerá en rojo y la barrera bajada y cuando esté a menos de 10 cm , el semáforo pasará a verde, se elevará una barrera y al cabo de 10 segundos se bajará la barrera y cambiará el semáforo a color rojo.(**NO UTILIZAR EL PIN 1**).



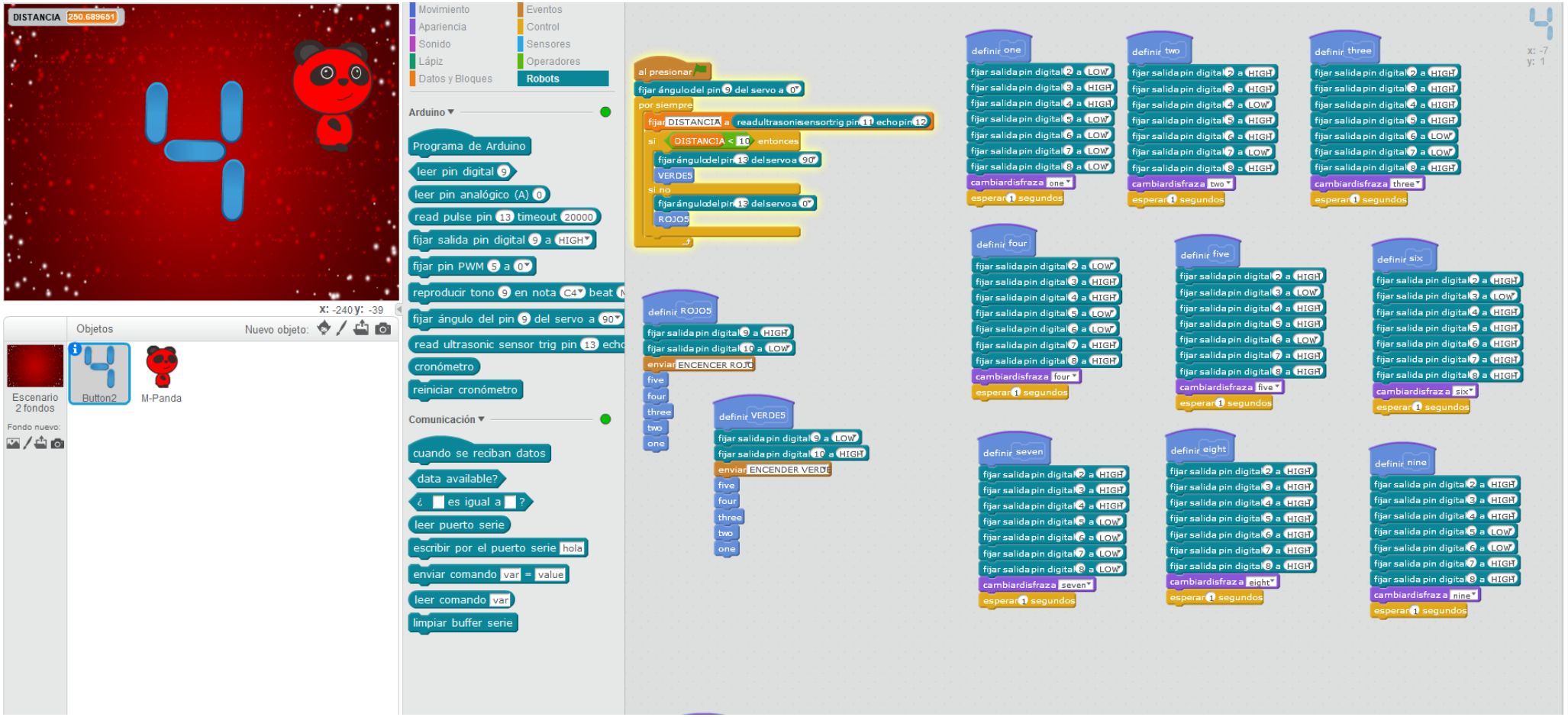
RETO B: A partir de la configuración de la tabla programa una barrera de estacionamiento en la que participen coordinados todos los elementos.

DISPLAY							LEDES		ULTRASONIDOS		SERVOMOTOR
a	b	c	d	e	f	g	R	V	TRIGER	ECHO	
2	3	4	5	6	7	8	9	10	11	12	13

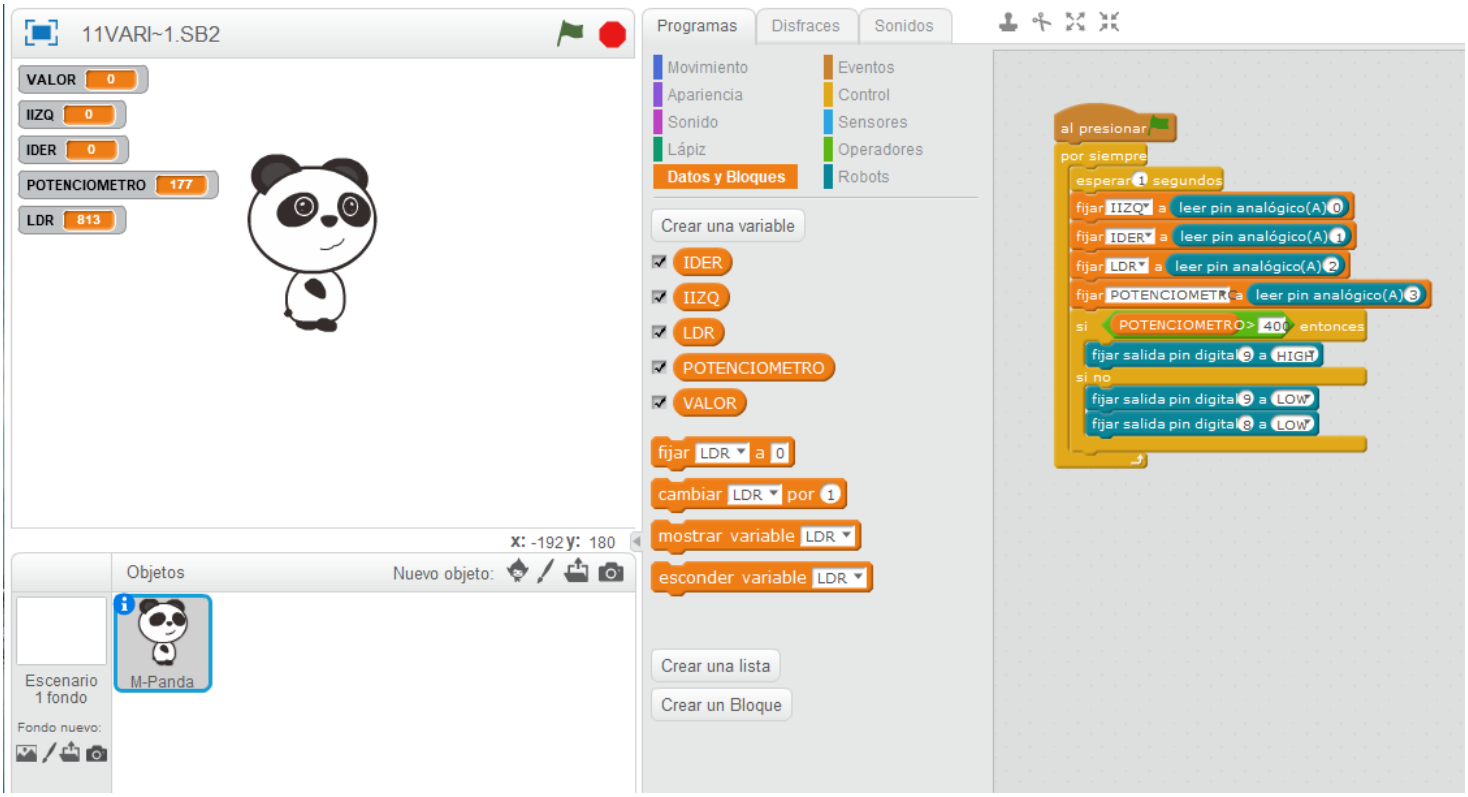
AYUDA.



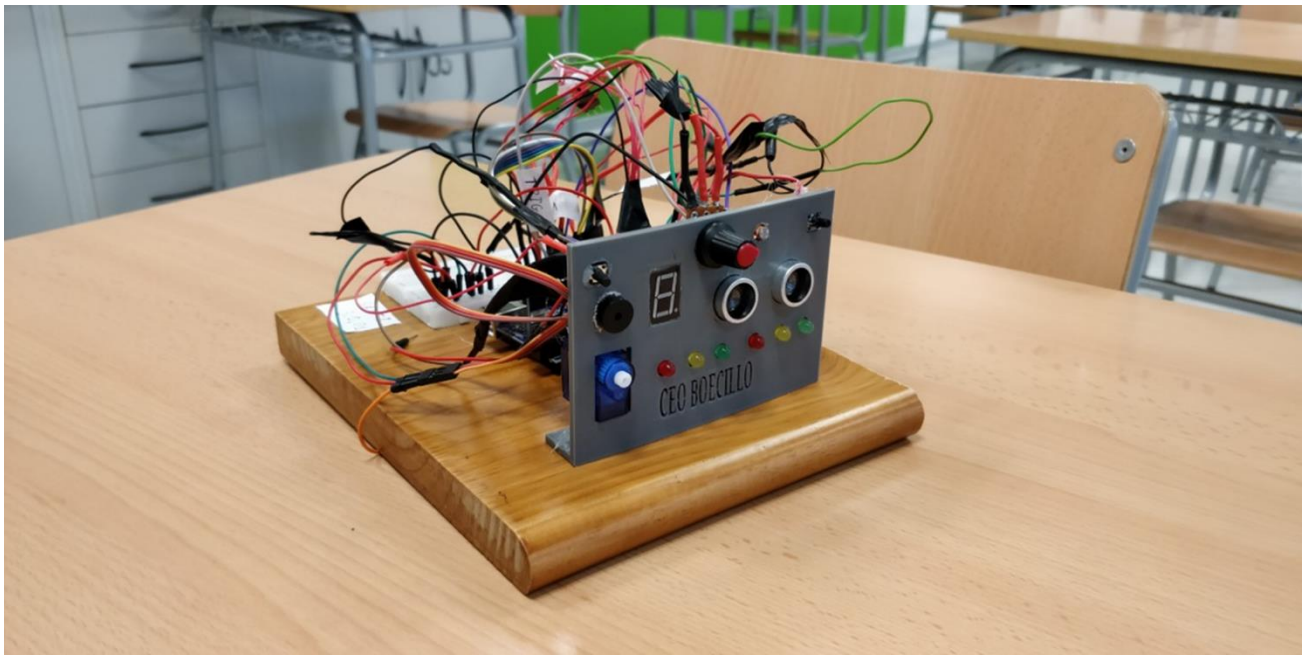
SOLUCIÓN RETO B.



12.- Entradas analógicas y digitales.

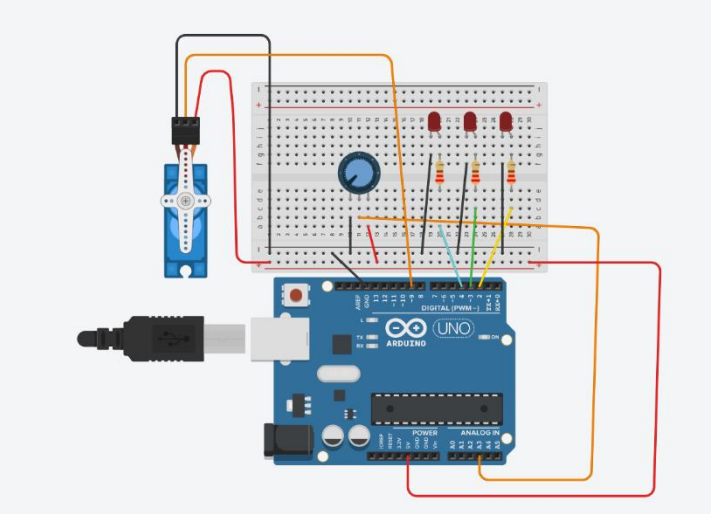


A partir de la comprensión de este contenido se pueden plantear ejercicios interactivos con servos, display, leds, etc.



13.- Control de un servomotor y encendido de 3 LEDs con un potenciómetro.

CONEXIONES

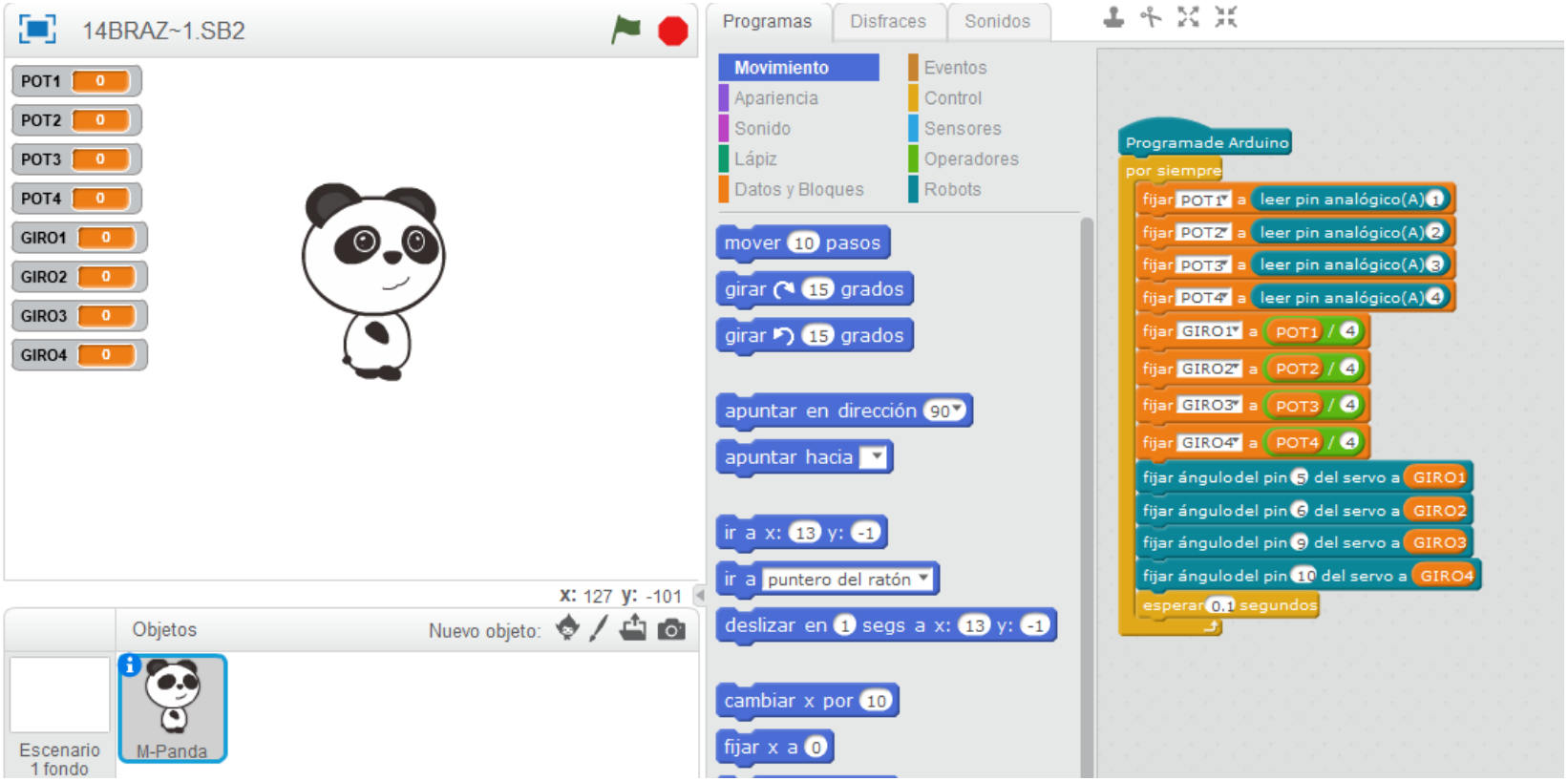


	SERVOM	POTENCIÓMETRO	LEDs		
PIN	9	A3	2	3	4

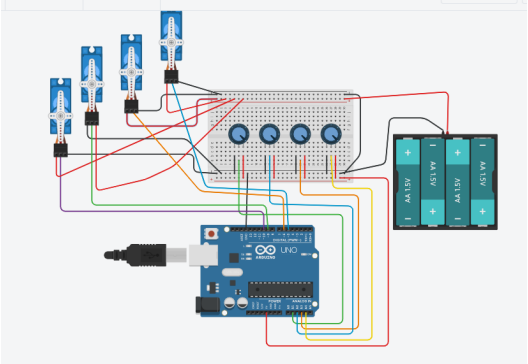
POTENCIOMETRO_LED_SERVO §

```
#include <Servo.h>
#define PINSERVO 9
#define POT A3
Servo myservo;
int giro = 0;
int valor=0;
#define LED2 2
#define LED3 3
#define LED4 4
void setup()
{
  Serial.begin(9600);
  myservo.attach(PINSERVO);
  pinMode(POT,INPUT);
  pinMode(LED2, OUTPUT);
   pinMode(LED3,OUTPUT);
   pinMode(LED4, OUTPUT);
 }
void loop()
{
valor=analogRead(POT);
giro=map(valor,0,1024,0,180);

myservo.write(giro);

  if(valor<300){
    digitalWrite(LED2,HIGH);
    digitalWrite(LED3,HIGH);
    digitalWrite(LED4,HIGH);
  }
  if(valor>300&&valor<600){
    digitalWrite(LED2,HIGH);
    digitalWrite(3,LOW);
    digitalWrite(4,LOW);
  }
  if(valor>600&& valor<800){
    digitalWrite(LED2,HIGH);
    digitalWrite(LED3,HIGH);
    digitalWrite(LED4,LOW);
  }
   if(valor>800){
    digitalWrite(LED2,HIGH);
    digitalWrite(LED3,HIGH);
    digitalWrite(LED4,HIGH);
   }
  delay(1000);
  Serial.print("El valor del poteciometro es= ");
  Serial.println(valor);
  Serial.print("malor giro= ");
  Serial.println(giro);
}
```

14.- Brazo robot. Control servos mediante potenciómetros.



14.- BRAZO ROBOT CONTROLADO POR CUATRO POTENCIÓMETROS. – MBLOCK.

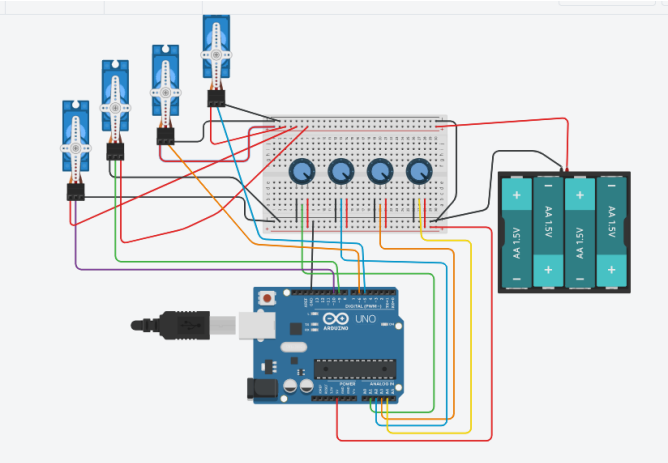


```
#include <Arduino.h>
#include <Wire.h>
#include <SoftwareSerial.h>

#include <Servo.h>

double angle_rad = PI/180.0;
double angle_deg = 180.0/PI;
double POT1;
double POT2;
double POT3;
double POT4;
double GIRO1;
double GIRO2;
double GIRO3;
double GIRO4;
Servo servo_5;
Servo servo_6;
Servo servo_9;
Servo servo_10;

```

```
void setup(){
    pinMode(A0+1,INPUT);
    pinMode(A0+2,INPUT);
    pinMode(A0+3,INPUT);
    pinMode(A0+4,INPUT);
    servo_5.attach(5);
    servo_6.attach(6);
    servo_9.attach(9);
    servo_10.attach(10);
}
```

```

void loop(){
    POT1 = analogRead(A0+1);
    POT2 = analogRead(A0+2);
    POT3 = analogRead(A0+3);
    POT4 = analogRead(A0+4);
    GIRO1 = (POT1) / (5.688);
    GIRO2 = (POT2) / (5.688);
    GIRO3 = (POT3) / (5.688);
    GIRO4 = (POT4) / (5.688);
    servo_5.write(GIRO1);
    servo_6.write(GIRO2);
    servo_9.write(GIRO3);
    servo_10.write(GIRO4);
    delay(1000*0.1);
}
```

14.- BRAZO ROBOT CONTROLADO POR CUATRO POTENCIÓMETROS - ARDUINO



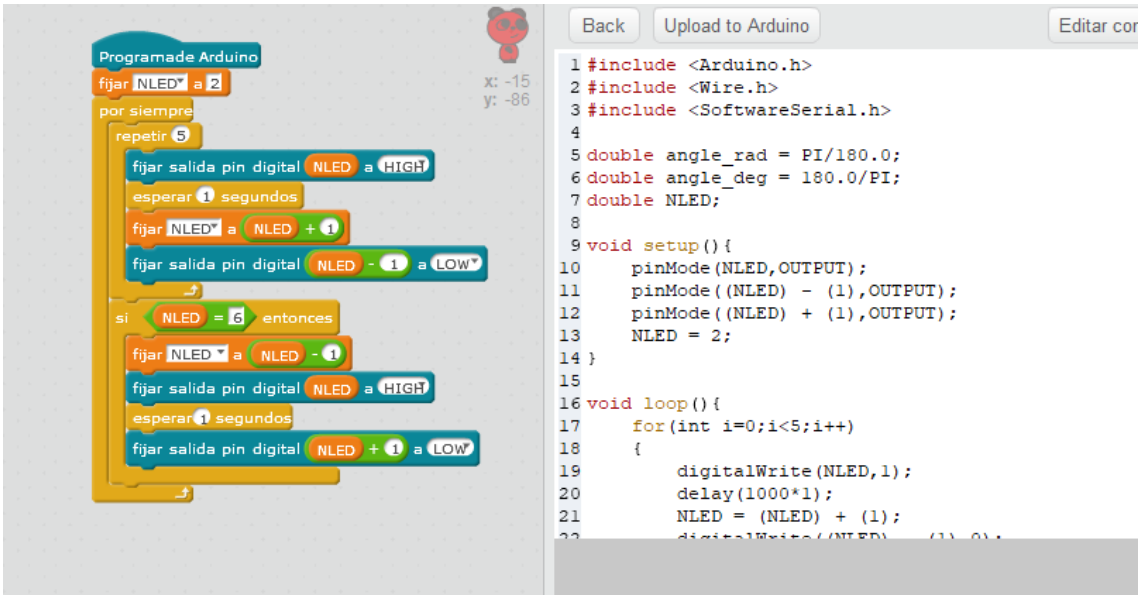
10_BRAZO_ROBOT_SAMUEL_4_SERVOS §

```
#include <Servo.h>

#define PINSERVO1 5
#define POT1 A1
Servo myservo1;

#define PINSERVO2 6
#define POT2 A2
Servo myservo2;

#define PINSERVO3 9
#define POT3 A3
Servo myservo3;

#define PINSERVO4 10
#define POT4 A4
Servo myservo4;

int giro1 = 0;
int valor1=0;
int giro2 = 0;
int valor2=0;
int giro3 = 0;
int valor3=0;
int giro4 = 0;
int valor4=0;

void setup()
{
  Serial.begin(9600);

   myservo1.attach(PINSERVO1);
   myservo2.attach(PINSERVO2);
   myservo3.attach(PINSERVO3);
   myservo4.attach(PINSERVO4);

    pinMode(POT1,INPUT);
    pinMode(POT2,INPUT);
    pinMode(POT3,INPUT);
    pinMode(POT4,INPUT);

 }
void loop()
{
valor1=analogRead(POT1);
giro1=map(valor1,0,1024,0,180);

valor2=analogRead(POT2);
giro2=map(valor1,0,1024,0,180);

valor3=analogRead(POT3);
giro3=map(valor3,0,1024,0,180);

valor4=analogRead(POT1);
giro4=map(valor1,0,1024,0,180);

  myservo1.write(giro1);
  myservo2.write(giro2);
  myservo3.write(giro3);
  myservo4.write(giro4);

  delay(1000);
  Serial.print("El valor del poteciometro 1 es= ");
  Serial.println(valor1);
  Serial.print("Valor giro1= ");
  Serial.println(giro1);

Serial.print("El valor del poteciometro 2 es= ");
  Serial.println(valor2);
  Serial.print("Valor giro1= ");
  Serial.println(giro2);

Serial.print("El valor del poteciometro 3 es= ");
  Serial.println(valor3);
  Serial.print("Valor giro1= ");
  Serial.println(giro3);

Serial.print("El valor del poteciometro 4 es= ");
  Serial.println(valor4);
  Serial.print("Valor giro4= ");
  Serial.println(giro4);
delay(500);

}
```

15.- COCHE FANTÁSTICO LEDS

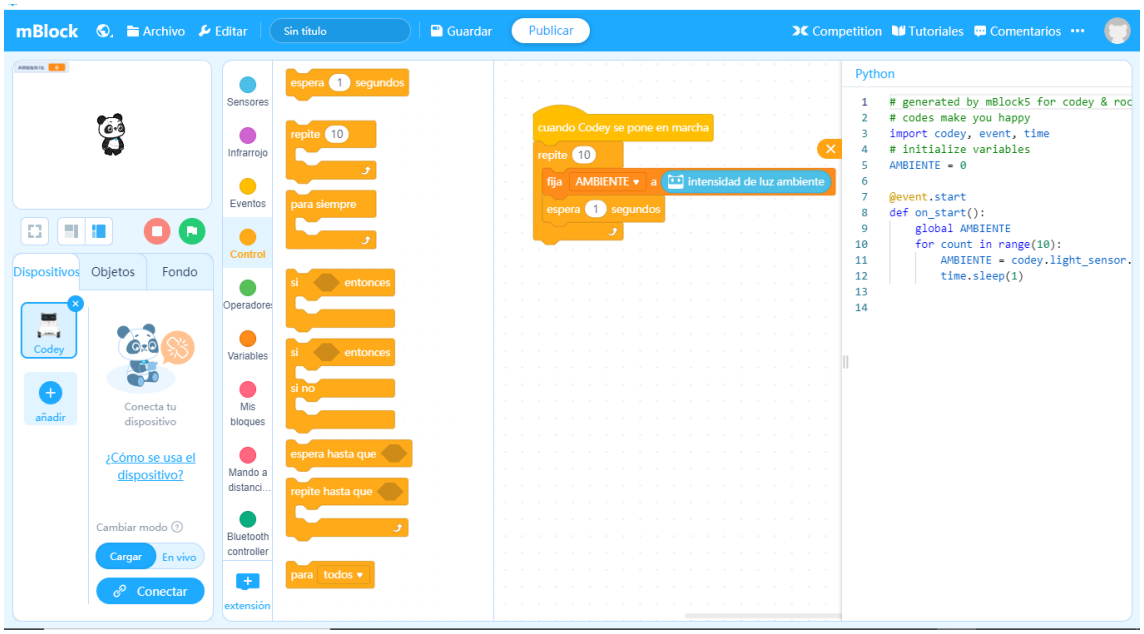
Programa una secuencia de encendido y apagado de 5 LEDs, de forma que semeje las luces del coche fantástico.

Ayuda:



16.- MBLOCK Y PYTHON

La nueva versión de Mblock permite el control de más dispositivos.



RETOS

A.- Programa un aparcamiento automático en el que intervengan: Un sensor de ultrasonidos, tres LEDes y un buzzer.

Funcionamiento: Al acercarse un objeto al sensor de ultrasonidos si está a más de 20 cm no se iluminará ningún LED ni sonará ningún sonido. Si el objeto está entre 20 y 15 cm se iluminará un LED y sonará un pitido cada 2 segundos. Si el objeto está entre 15 y 10 cm se iluminarán dos LEDes y sonará un pitido cada 1 segundos. Si el objeto está entre 10 y 15 cm se iluminarán tres LEDes y sonará un pitido cada 0,5 segundos.

B.- Programa un piano que funcione con pulsadores y un buzzer.

C.- Programa un display controlado por un potenciómetro.

D.- Crea un reto y resuélvelo.