



Dosis Web

<https://github.com/toplapbogota/dosisweb>

Arduino Firmata

<https://firmata.netlify.app>

```
1 // ...
2 // ...
3 // ...
4 // ...
5 // ...
6 // ...
7 // ...
8 // ...
9 // ...
10 // ...
11 // ...
12 // ...
13 // ...
14 // ...
15 // ...
16 // ...
17 // ...
18 // ...
19 // ...
20 // ...
21 // ...
22 // ...
23 // ...
24 // ...
25 // ...
26 // ...
27 // ...
28 // ...
29 // ...
30 // ...
31 // ...
32 // ...
33 // ...
34 // ...
35 // ...
36 // ...
37 // ...
38 // ...
39 // ...
40 // ...
41 // ...
42 // ...
43 // ...
44 // ...
45 // ...
46 // ...
47 // ...
48 // ...
49 // ...
50 // ...
51 // ...
52 // ...
53 // ...
54 // ...
55 // ...
56 // ...
57 // ...
58 // ...
59 // ...
60 // ...
61 // ...
62 // ...
63 // ...
64 // ...
65 // ...
66 // ...
67 // ...
68 // ...
69 // ...
70 // ...
71 // ...
72 // ...
73 // ...
74 // ...
75 // ...
76 // ...
77 // ...
78 // ...
79 // ...
80 // ...
81 // ...
82 // ...
83 // ...
84 // ...
85 // ...
86 // ...
87 // ...
88 // ...
89 // ...
90 // ...
91 // ...
92 // ...
93 // ...
94 // ...
95 // ...
96 // ...
97 // ...
98 // ...
99 // ...
100 // ...
```

DOSIS
WEB

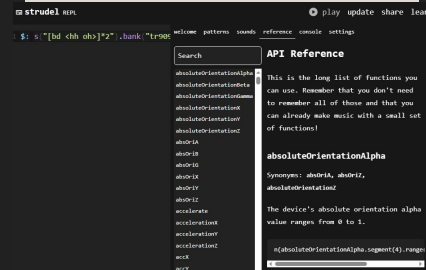
FIRMATA

OSC

OSC



<https://www.arduino.cc/>



<https://strudel.cc>

OSC mobile



GITHUB DOSIS WEB

<https://github.com/toplapbogota/dosisweb>

DOSIS WEB

LEDS

MOTORES DC

SERVO MOTORES

MOTORES PASO A PASO

SOLENOIDES

SONIDO

MICRÓFONOS DE CONTACTO

CONSOLA / INTERFAZ DE SONIDO

CABLEADO A MESA DE MEZCLA

DOSIS WEB

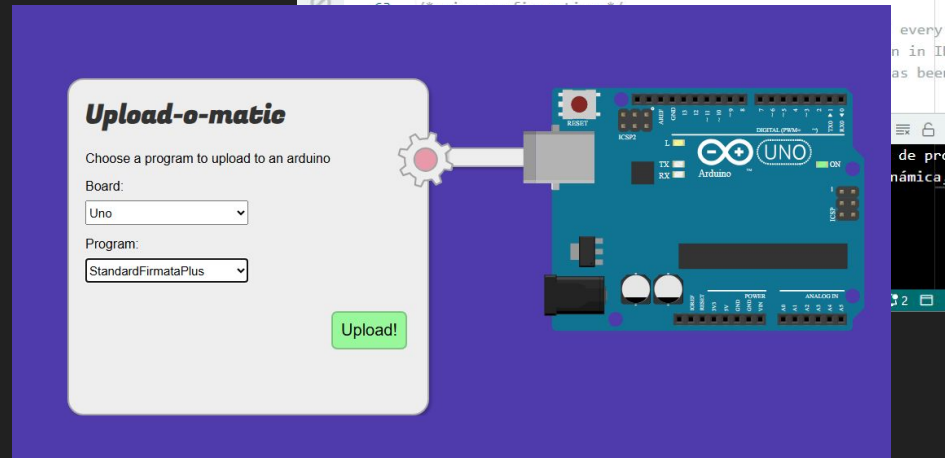
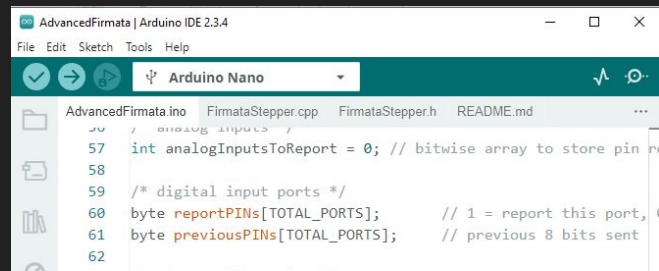


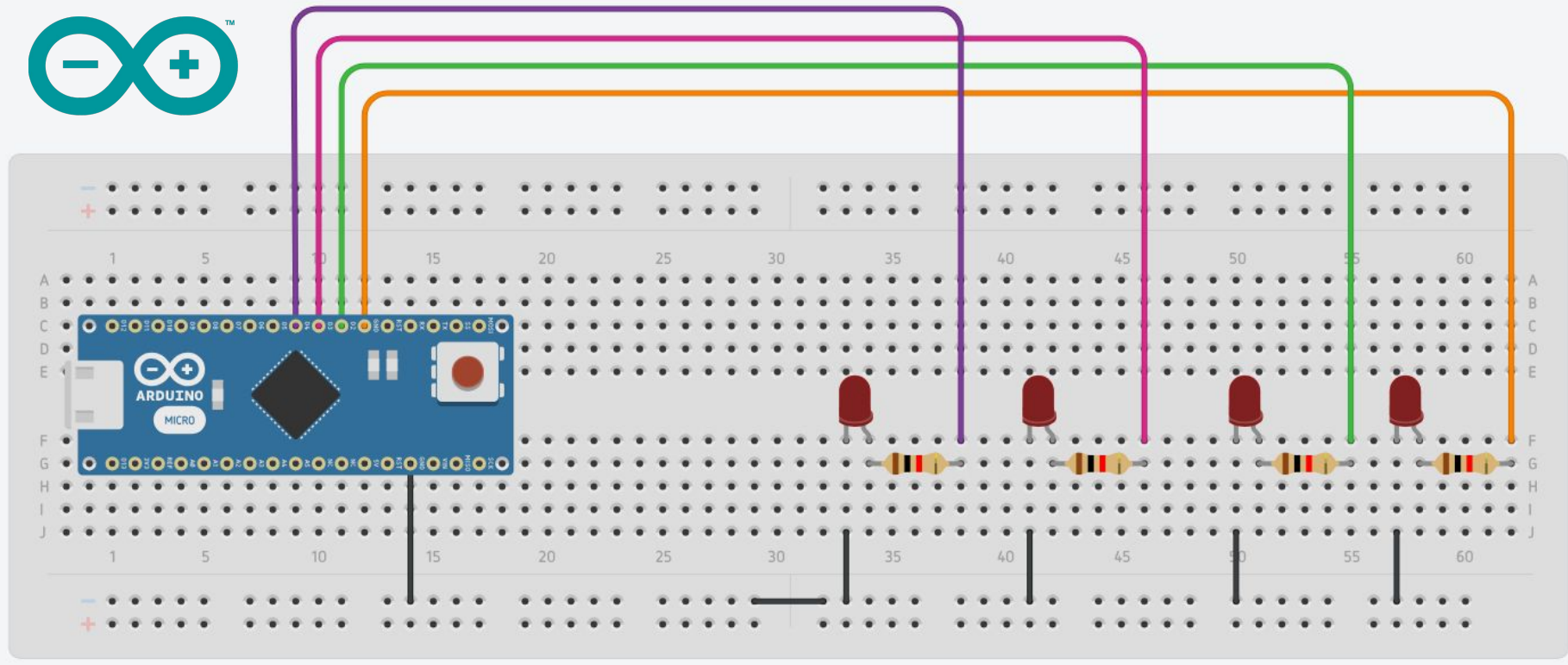
DOSIS WEB

```
1 |
2 |
3 | console.log("Escapar de la pantalla");
4 |
5 | toplap Bogotá :: CTRL+enter
6 |
7 |
8 | prender(13)
9 |
10 |
11 |
12 | apagar(13)
13 |
14 | dcmotor({pin1:0,pin2:7,pin3:8,dir:"derecha",vel:250,estado:2,modo:"descendente",tie
15 |
16 | apagarMotoresDC()
17 |
18 | teclado({estado:0,tiempo:3000})
19 |
20 | pararTeclado()
21 |
```

Controls
recargar
iniciar
encender la cámara

FIRMATA Plus





DOSIS WEB



CÓDIGOS:

```
//prende todos los pines de manera  
simultánea
```

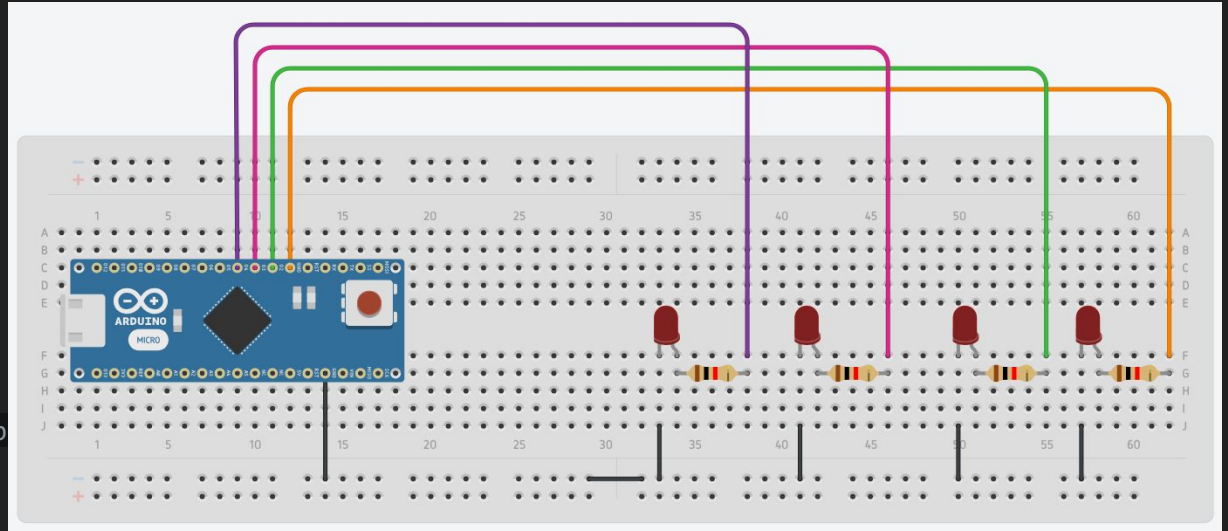
```
prender(2,3,4,5)
```

```
//apaga todos los pines de manera  
simultánea
```

```
apagar(2,3,4,5)
```

```
//prende y apaga consecutivamente cada p  
solo una vez
```

```
serie(2,3)
```



DOSIS WEB



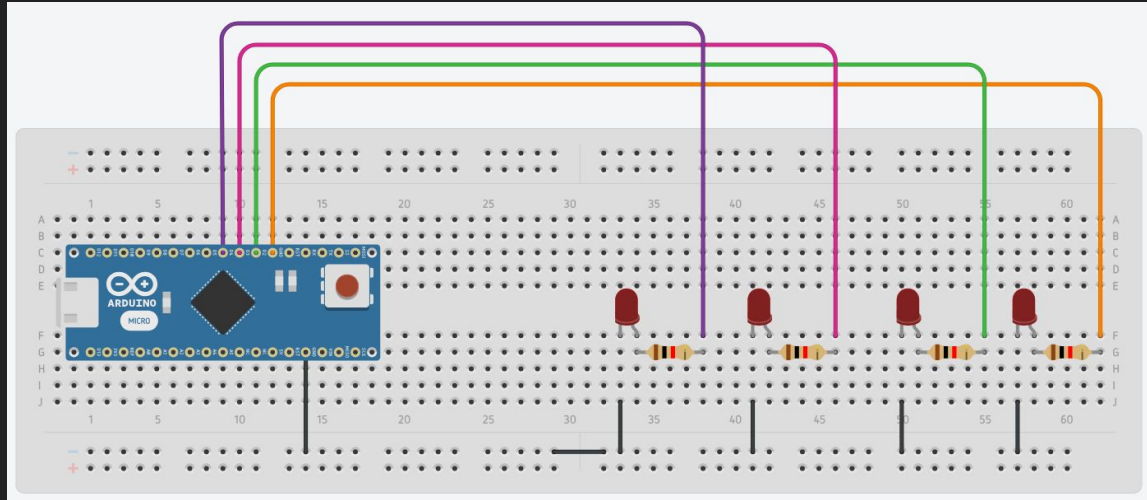
CÓDIGOS:

```
//prende y apaga consecutivamente cada pin y  
vuelve a empezar el ciclo
```

```
bucle(2,3,4,5)
```

```
//tiempo en milisegundos que activo cada pin  
cambiarIntervalo(500)
```

```
//para el bucle  
detenerBucle()
```



DOSIS WEB



CÓDIGOS:

```
//activa pines de manera simultanea, pero cada pin tiene un comportamiento en bucle de forma independiente
```

```
//inicio: define el momento en que se activa el pin
```

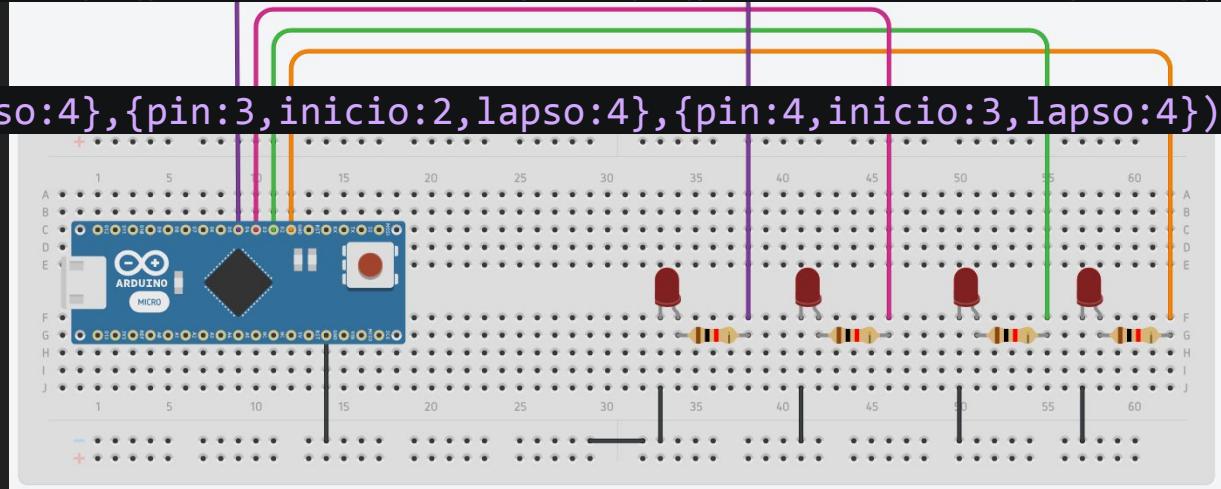
```
//lapso: es el tiempo que dura activo el pin
```

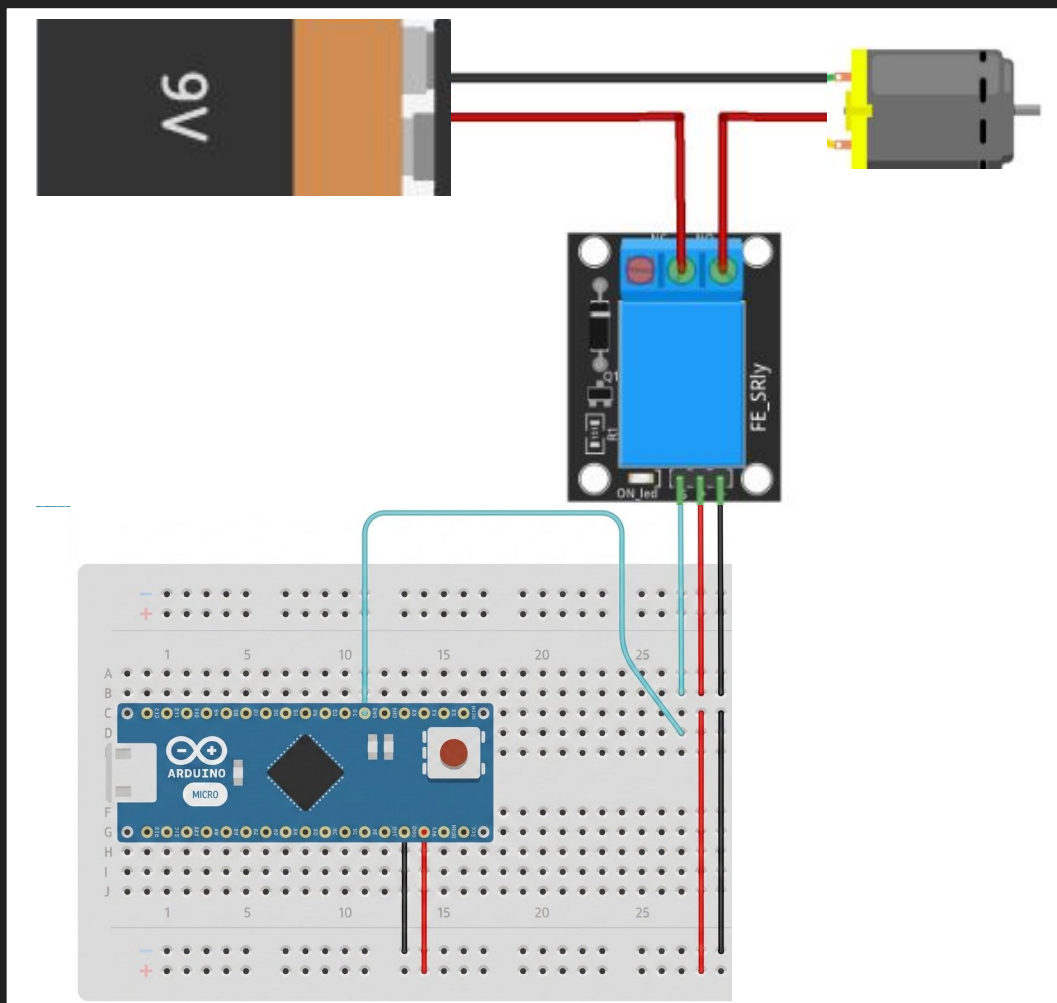
```
paralelo({pin:2,inicio:2,lapso:4},{pin:3,inicio:2,lapso:4},{pin:3,inicio:2,lapso:4})
```

```
paralelo({pin:2,inicio:1,lapso:4},{pin:3,inicio:2,lapso:4},{pin:4,inicio:3,lapso:4})
```

```
//apaga todos los pines
```

```
apagarTodo()
```





DOSIS WEB



CÓDIGOS:

```
//prende el pin y lo mantiene activo por un  
intervalo de tiempo
```

```
teclado({estado:0,tiempo:5000})
```

```
2222
```

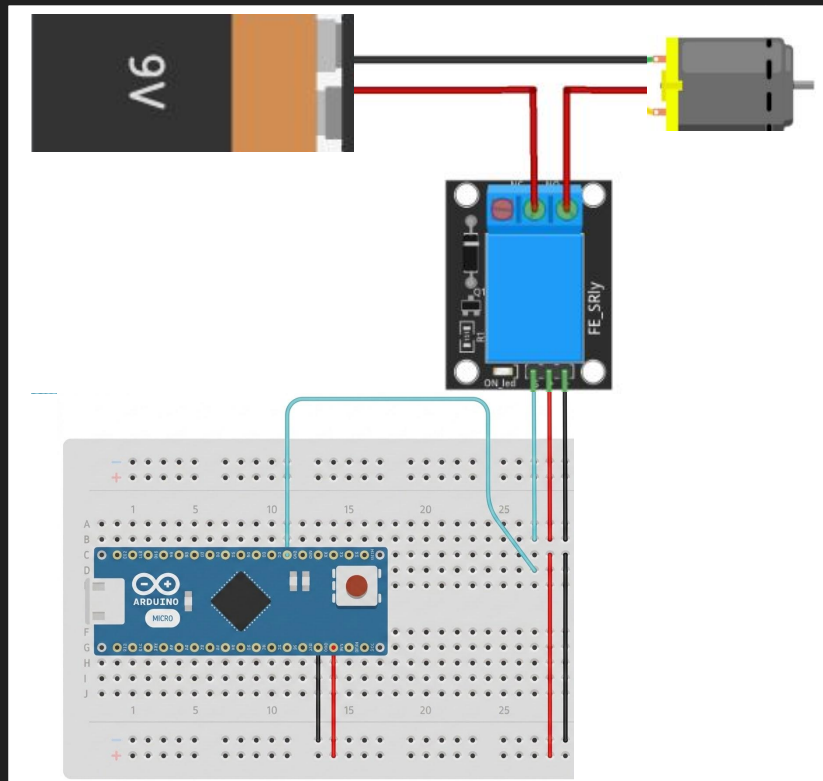
```
//activa el pin mientras se mantenga la tecla  
presionada
```

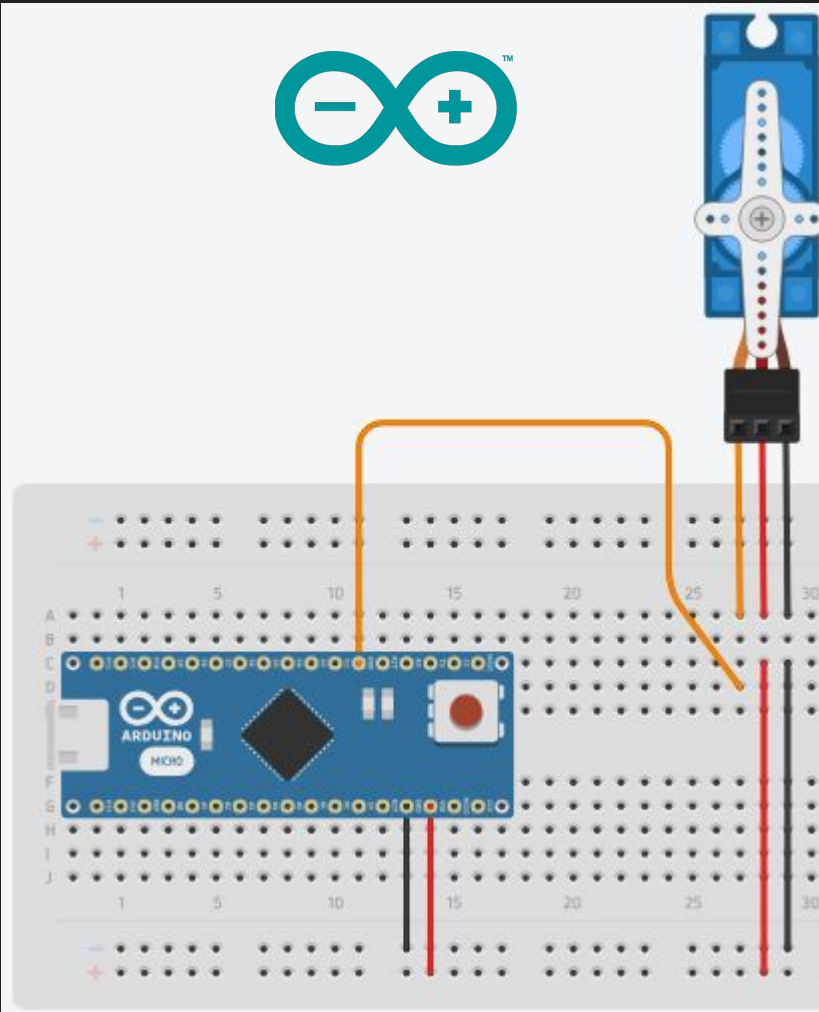
```
teclado({estado:1,tiempo:0})
```

```
222
```

```
//detiene el funcionamiento del teclado
```

```
pararTeclado();
```





Arduino con Standard Firmata*

DOSIS WEB



Arduino con Standard Firmata*

CÓDIGOS:

```
//start: angulo de inicio
```

```
//final: angulo hasta donde va a llegar
```

```
//estado 0: bucle del movimiento
```

```
servo({pin:2,estado:0, start:0, final:80, tiempo:0, pasos:0})
```

```
//estado 1: Genera el movimiento desde el start hasta el final en un tiempo definido.
```

```
//El final debe ser mayor al start.
```

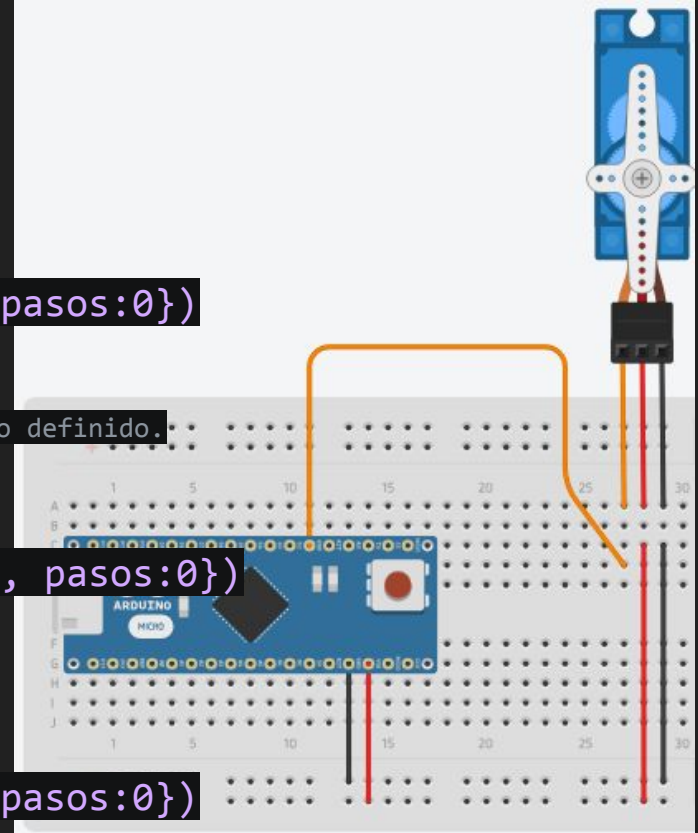
```
//Tiempo: que se demora realizando el movimiento
```

```
servo({pin:2,estado:1, start:20, final:80, tiempo:50, pasos:0})
```

```
//estado 2: Genera el movimiento hacia el ángulo final.
```

```
//No tiene en cuenta start, ni pasos, ni tiempo.
```

```
servo({pin:2,estado:2, start:0, final:80, tiempo:0, pasos:0})
```



DOSIS WEB



Arduino con Standard Firmata*

```
//estado 3: Genera el movimiento por pasos en un tiempo definido.
```

```
//servo({pin:2, estado:3, start:0, final:80, tiempo:10, pasos:5})
```

```
//estado 4: Bucle del movimiento con pausa.
```

```
//TiempoMov: tiempo que demora en iniciar el movimiento
```

```
//TiempoPausa: tiempo que demora en iniciar el bucle cuando llega al ángulo final
```

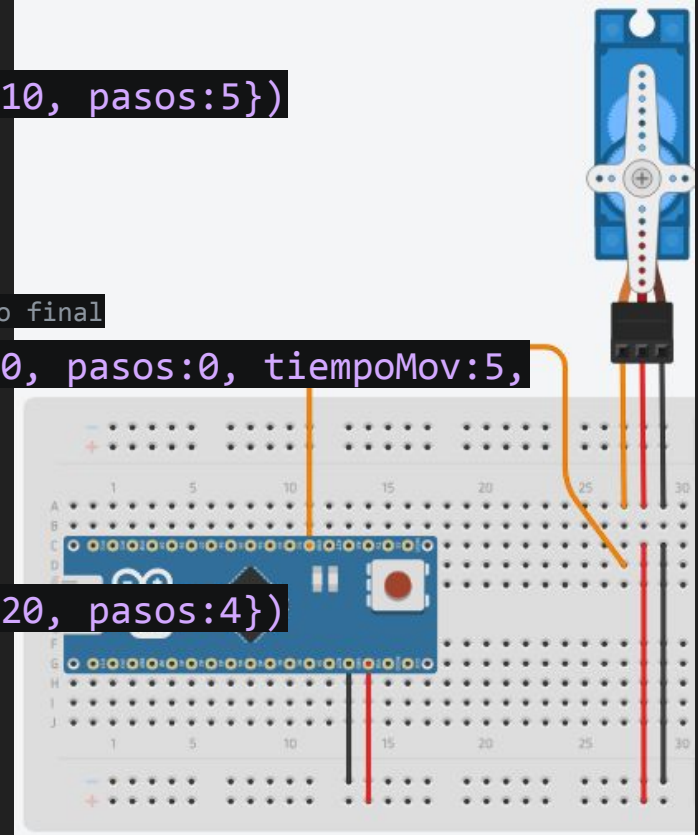
```
servo({pin:2, estado:4, start:40, final:160, tiempo:0, pasos:0, tiempoMov:5,  
tiempoPausa:2})
```

```
//Bucle de movimiento en un tiempo definido y movmimiento por pasos
```

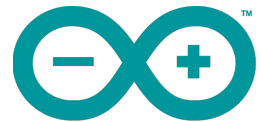
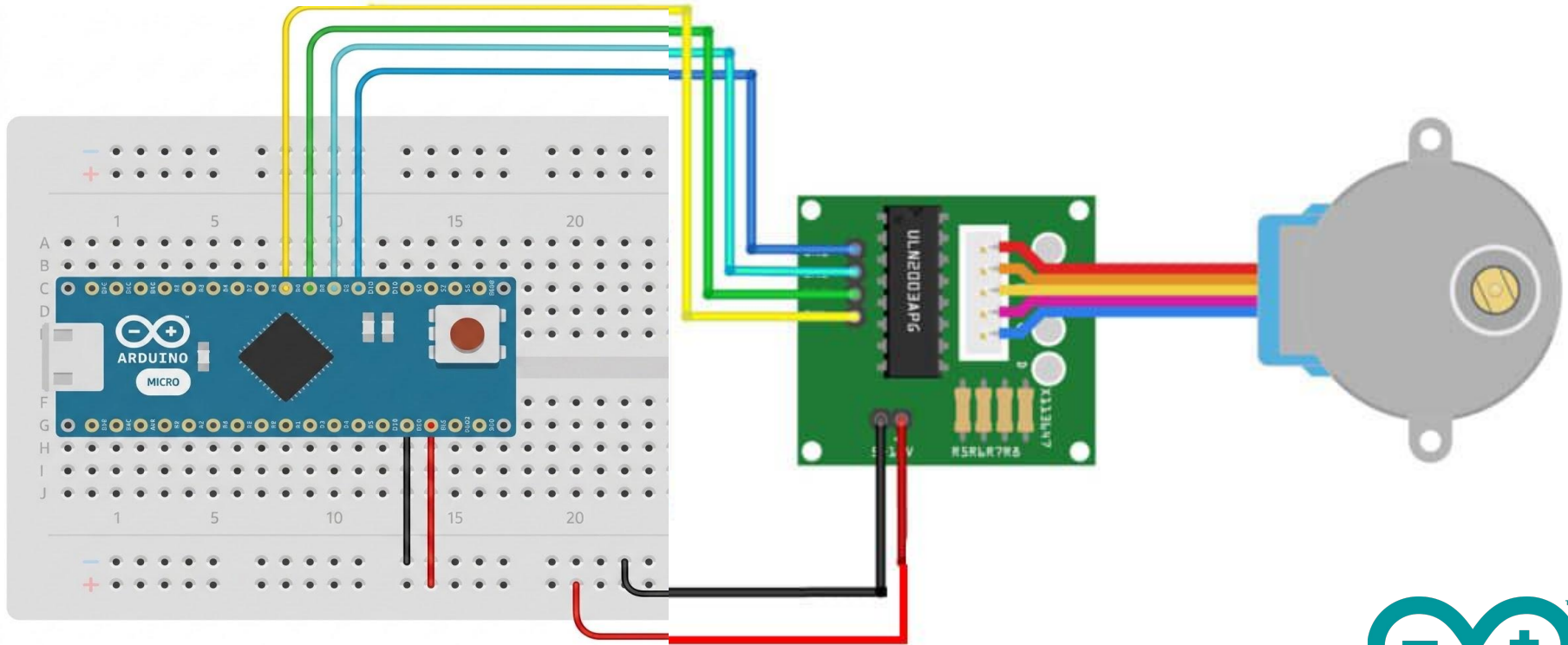
```
servo({pin:2, estado:5, start:40, final:160, tiempo:20, pasos:4})
```

```
//Detiene todos los Servomotores
```

```
apagarServos()
```



Arduino con Advance Firmata*



DOSIS WEB



Arduino con Advance Firmata*

```
//estado 0: Genera el movimiento en bucle.
```

```
//rpm: es la velocidad del giro (0 - 17)
```

```
//pasos: 2048 corresponde a una vuelta. 1024 media vuelta
```

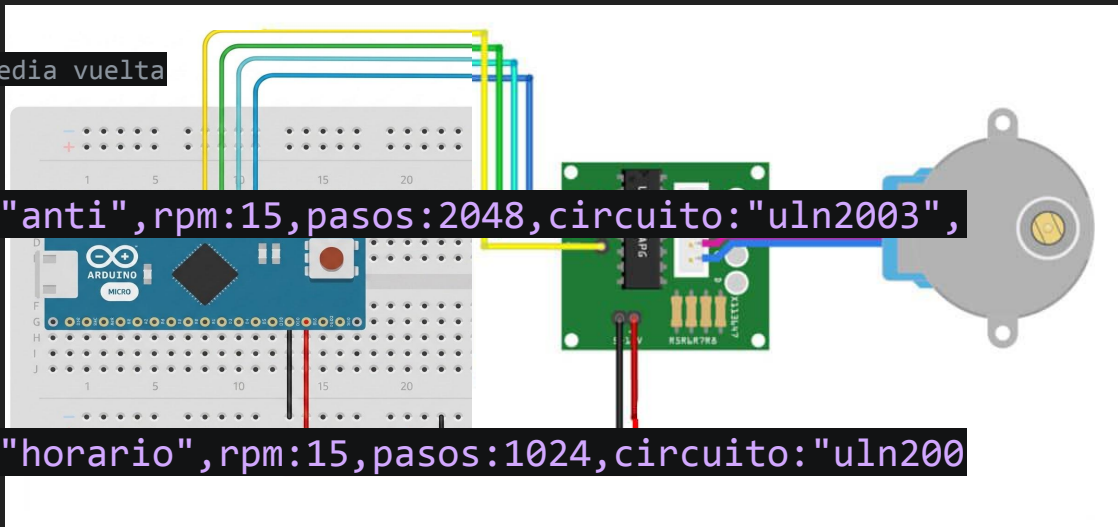
```
//sentido: "anti" giro en sentido anti horario
```

```
//sentido: "horario" giro en sentido horario
```

```
stepper({pines:[4,2,3,5],sentido:"anti",rpm:15,pasos:2048,circuito:"uIn2003",  
motor:"byj48",estado:0})
```

```
//estado 1: Avanza el número de pasos indicados
```

```
stepper({pines:[4,2,3,5],sentido:"horario",rpm:15,pasos:1024,circuito:"uIn200  
3",motor:"byj48",estado:1})
```



DOSIS WEB



Arduino con Advance Firmata*

```
//estado 2: Realiza UN solo bucle
```

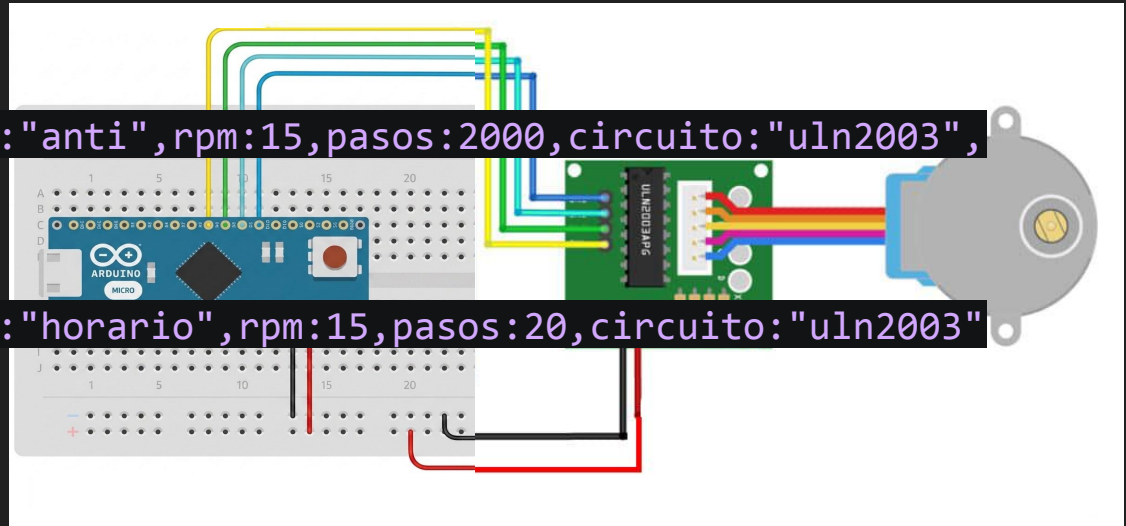
```
stepper({pines:[4,2,3,5],sentido:"anti",rpm:15,pasos:2000,circuito:"uIn2003",  
motor:"byj48",estado:2
```

```
//estado 3: Realiza el giro continuamente
```

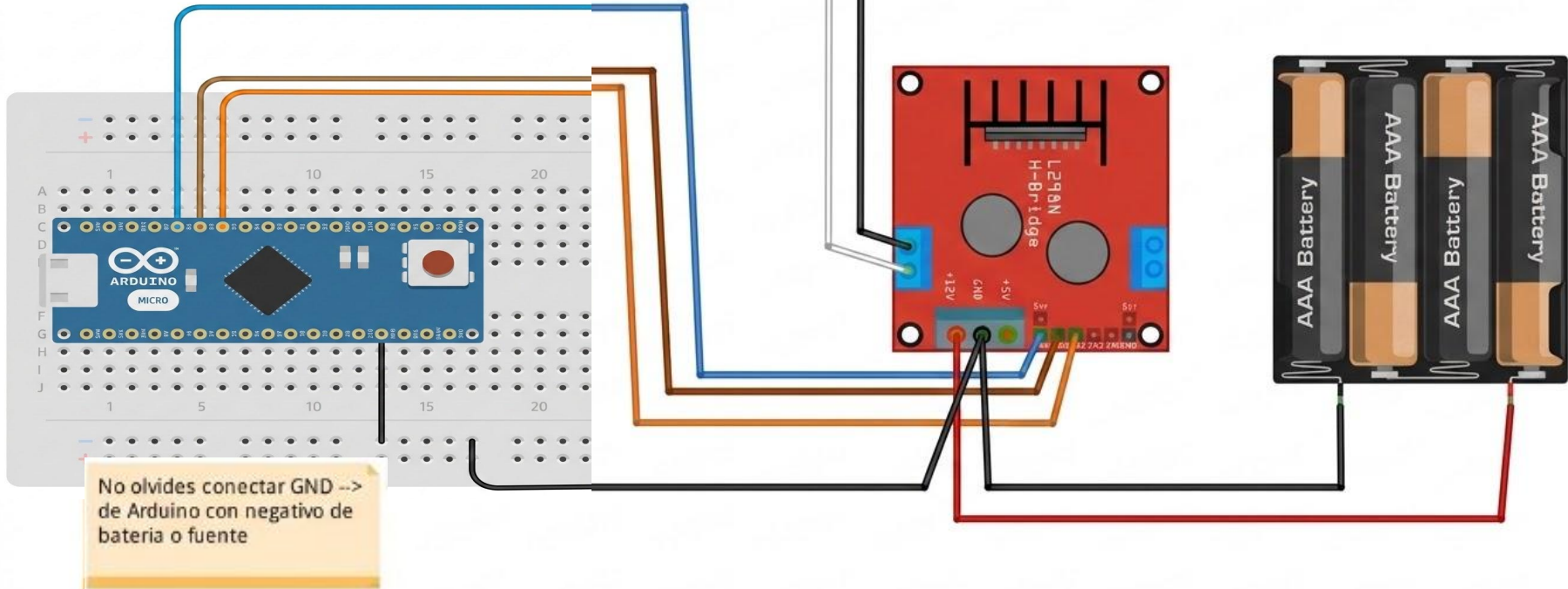
```
stepper({pines:[4,2,3,5],sentido:"horario",rpm:15,pasos:20,circuito:"uIn2003",  
,motor:"byj48",estado:3})
```

```
//Detiene todos los motores paso a paso
```

```
detenerStepper()
```



Módulo L298N
Motor DC de 3 a 6 V
Alimentación: - 4 pilas 1.5 V (o 1.2 V si son recargables)
- Fuente de 6, 7.5 V o 9 V



DOSIS WEB



Arduino con Standard Firmata*

```
//estado 0: Realiza el giro continuamente con la misma velocidad
```

```
//dir: izquierda o derecha, son convenciones
```

```
//el primer pin, pin1, es el de PWM, es con el que controla la velocidad
```

```
dcmotor({pin1:9,pin2:7,pin3:8,dir:"derecha",vel:150,estado:0})
```

```
//estado 1: Realiza el giro continuamente con velocidad ascendente o descendente
```

```
//modo: ascendente o descendente
```

```
//tiempo: tiempo en que se demora realizando el modo
```

```
dcmotor({pin1:9,pin2:7,pin3:8,dir:"izquierda",vel:150,estado:1,modo:  
"ascendente",tiempo:500})
```

Motor DC de 3 a 6 V

Alimentación: - 4 pilas 1.5 V (o 1.2 V si son recargables)

- Fuente de 6, 7.5 V o 9 V



de Arduino con negativo de
batería o fuente

DOSIS WEB



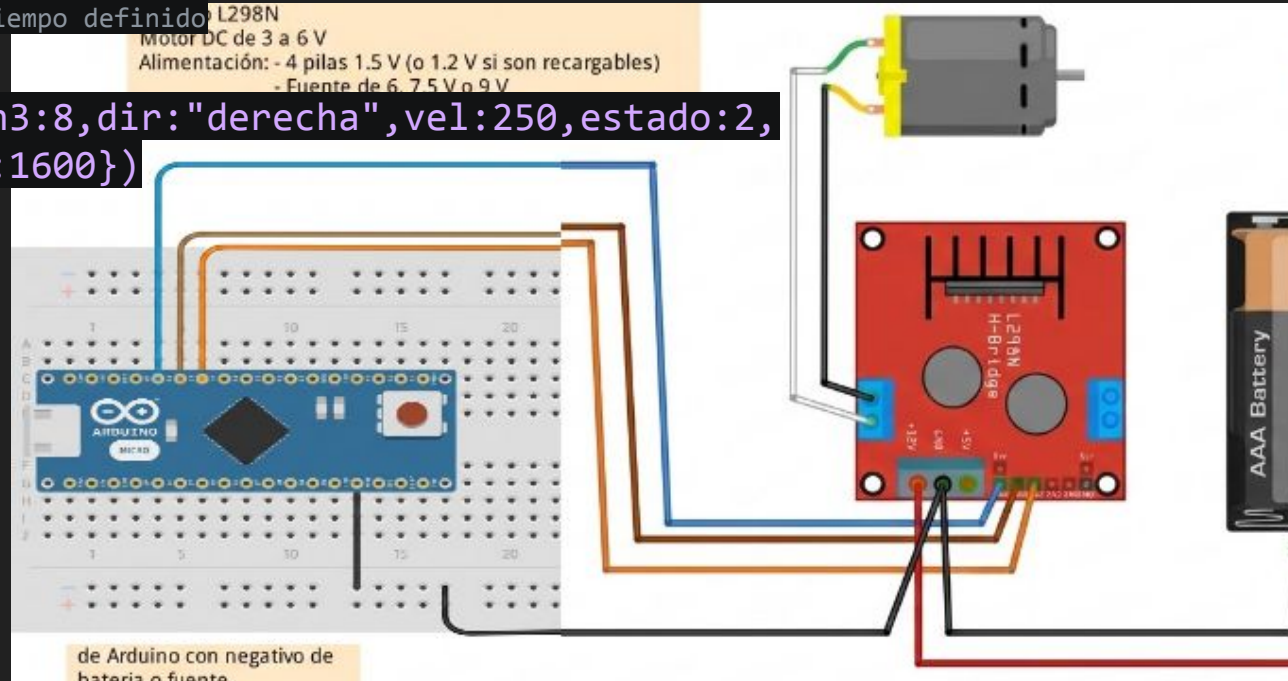
Arduino con Standard Firmata*

```
//estado 2: Realiza el giro por un tiempo definido
```

```
//tiempo: que dura activado el giro
```

```
dcmotor({pin1:9,pin2:7,pin3:8,dir:"derecha",vel:250,estado:2,  
modo:"descendente",tiempo:1600})
```

```
//Detener el giro de los motores DC  
apagarMotoresDC()
```



DOSIS WEB



```
//Enciende las cámaras web y muestra en la consola las cámaras que están conectadas al computador
```

```
listarCamaras()
```

```
//muestra la cámara en la pantalla
```

```
//x: ubicación horizontal (píxeles)
```

```
//y: ubicación vertical (píxeles)
```

```
//ancho: el ancho de la cámara en la pantalla (píxeles)
```

```
//alto: el alto de la cámara en la pantalla (píxeles)
```

```
webcam({camara:0,x:500,y:200,ancho:320,alto:240})
```

```
//Duplica la imagen de la cámara creada anteriormente
```

```
webcam({camara:0,duplicado:1,x:0,y:350,ancho:320,alto:240})
```

DOSIS WEB



```
//Borra la imagen de la cámara web en la pantalla
```

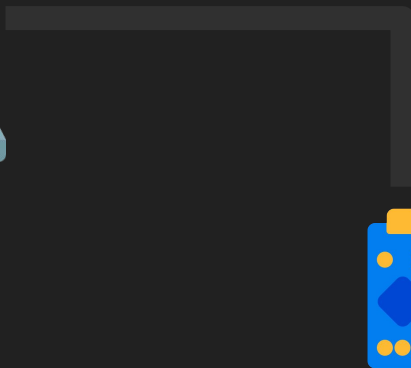
```
borrarWebcam({camara:0})
```

```
//Borra la imagen duplicada de la cámara web en la pantalla
```

```
borrarWebcam({camara:0,duplicado:1})
```



+ DOSIS
WEB +



//El celular y el pc deben estar
Conectados en la misma red

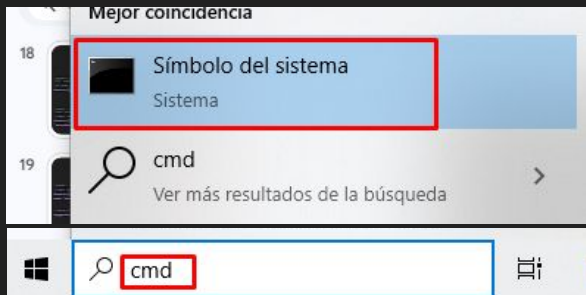
https://play.google.com/store/apps/details?id=com.ffsmultimedia.osccontroler&pcampaignid=web_share



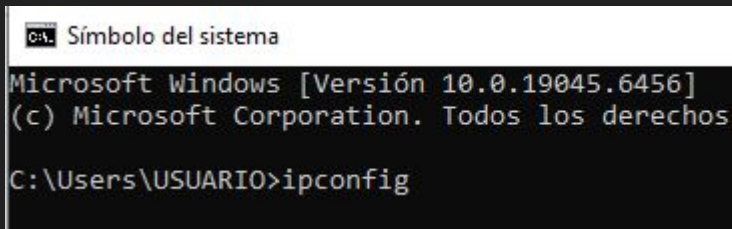
DOSIS
WEB



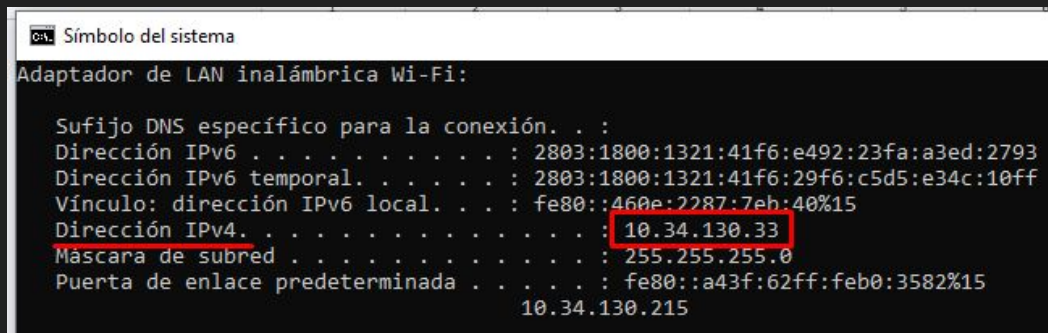
//1. Abrir la consola utilizando el comando CMD



//2. Escribir el comando: ipconfig

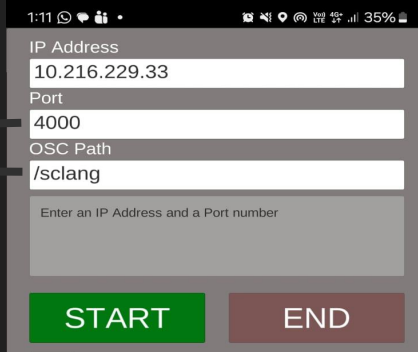


//3. Copiar el número de Dirección IPv4



//4. Pegar el número de Dirección IPv4

En la aplicación del celular





DOSIS
WEB



```
//Para iniciar OSC ("OSC Path" , Port)
```

```
activarOSC("/sclang",4000)
```

```
//Borra la imagen duplicada de la cámara web en la pantalla
```

```
oscEscuchar("button1","prender",13)
```

```
oscEscuchar("button2","apagar",13)
```

1:11 35%

IP Address
10.216.229.33

Port
4000

OSC Path
/sclang

Enter an IP Address and a Port number

START END

☐ Start automatically when app starts

CONTROLS

Setup

Note: You may need to toggle windowed/fullscreen mode after editing any text field

Switch to Windowed

Export Import

Quit Reset



DOSIS
WEB



```
oscEscuchar("button2","apagar",13)
```

```
oscEscuchar("button3","servo",{pin:2,estado:0, start:90, final:180, tiempo:0, pasos:0})
```

```
oscEscuchar("button2","apagarServos")
```

```
oscEscuchar("slider1","servoAnalogo",{pin:2, oscMin:0, oscMax:1, start:0, final:180})
```

```
oscEscuchar("slider2","servoAnalogo",{pin:3, oscMin:0, oscMax:1, start:0, final:180})
```

```
oscEscuchar("slider3","servoAnalogo",{pin:4, oscMin:0, oscMax:1, start:0, final:180})
```

HARDWARE HACKING ANÁLOGO



<https://vimeo.com/383910827?fl=pl&fe=sh>



<https://vimeo.com/361578410?fl=pl&fe=sh>

REFERENTES



<https://www.instagram.com/reels/Daxa2R0BCk4/>



<https://www.youtube.com/watch?v=zCIWpDyWEqE>

REFERENTES



<https://youtu.be/C5kRekXXLmk?si=2fNOPojOTT8qFinZ>



<https://www.youtube.com/watch?v=8i7NJayHuxc&list=PLkzEMII7nz0UH7OJAPefgruOylBns6cxC&index=127>