



Programación con MicroPython

Alexander López Parrado, PhD



UNIVERSIDAD
DEL QUINDÍO

**ARDUINO
DAY 2023**

March 25th
day.arduino.cc
[#ArduinoDay2023](https://twitter.com/ArduinoDay2023)

Contenido

- ❑ **Introducción**
 - *Instalación del firmware, herramientas software*
- ❑ **Acceso a periféricos**
 - *Entradas y salidas digitales, entradas analógicas y PWM*
- ❑ **Características avanzadas**
 - *Hilos, FSMs, Reducción consumo energético*
- ❑ **Envío de datos a plataformas en la nube**
 - *WiFi, Firebase.*



March 25th
day.arduino.cc
#ArduinoDay2023

INTRODUCCIÓN



March 25th
day.arduino.cc
[#ArduinoDay2023](https://twitter.com/ArduinoDay2023)



❑ MicroPython

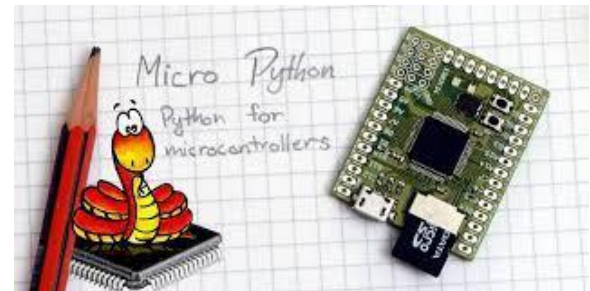
- *Implementación básica y eficiente de Python 3*
 - *Incluye un subconjunto pequeño de la biblioteca estándar de Python.*
 - *Optimizado para correr en microcontroladores y entornos restringidos.*
- ❑ El firmware (interprete de Python) debe ser almacenado en la memoria flash del microcontrolador.
- ❑ Incluye el REPL (*Read Evaluate Print Loop*) a través de una conexión serial (USB) o WEB.

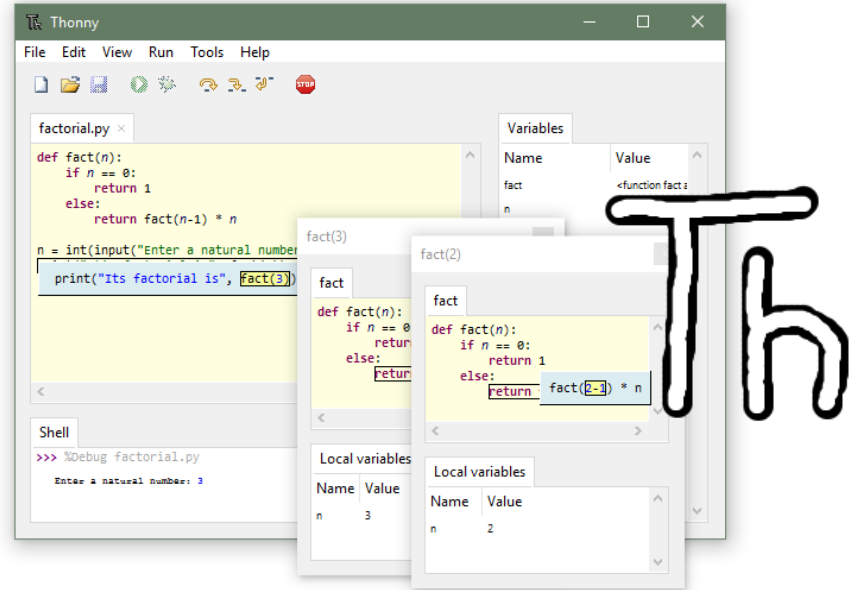
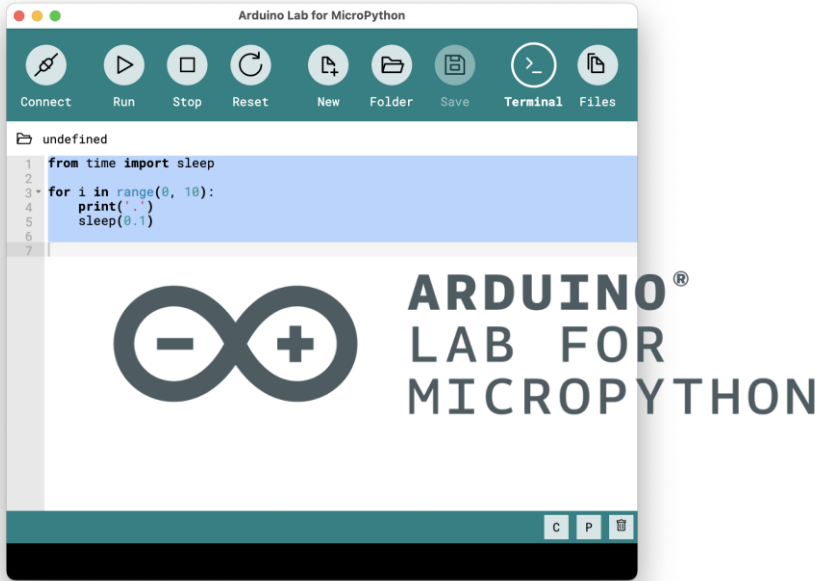


ARDUINO
DAY 2023



UNIVERSIDAD
DEL QUINDÍO





```
Thonny - MicroPython device :: /boot.py @ 13: 1
File Edit View Run Tools Help
[boot.py] x
1 # This file is executed on every boot (including wake-boot from deepsleep)
2 #import esp
3 #esp.osdebug(None)
4 #import webrepl
5 #webrepl.start()
6 from machine import deepsleep
7 from machine import Pin
8 from machine import RTC
9 from time import sleep
10
11 led = Pin (2, Pin.OUT)
12
13 rtc = RTC()
14 rtcData=rtc.memory()
15
16 if len(rtcData)==0:
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

Shell x
load:0x10070000,len:12071
ho 0 tail 12 room 4
load:0x40080400,len:4124
entry 0x40080680
1
Traceback (most recent call last):
  File "boot.py", line 37, in <module>
KeyboardInterrupt:
MicroPython v1.19.1 on 2022-06-18; ESP32 module with ESP32
Type "help()" for more information.

MicroPython v1.19.1 on 2022-06-18; ESP32 module with ESP32
Type "help()" for more information.
>>> |
```

Th

<https://thonny.org/>



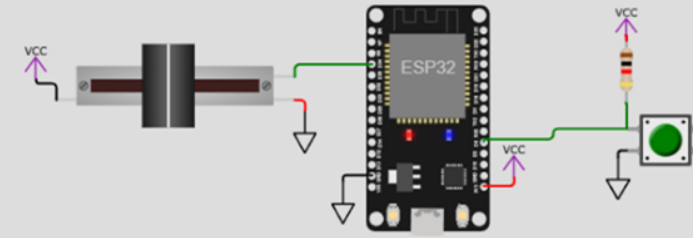
wokwi.com/projects/359847635338319873

WOKWI SAVE SHARE Arduino-Day-2023-Threads-FSM Docs

```
main.py diagram.json
30 if counter==0:
31     led_pin.value(not led_pin.value())
32
33 if state==0:
34     if button_pin.value()==0:
35         print('Button pressed')
36         state=1
37 elif state==1:
38     if button_pin.value()==1:
39         state=0
40         print('Button released')
41         buttonFlag=not(buttonFlag)
42
43
44
45 time.sleep_ms(delay)
46
47
48
49
50
51 buttonFlag=True
52
53
54 _thread.start_new_thread(button_thread,(10,500))
55 _thread.start_new_thread adc_thread,(1000,))
56
57
```

Simulation

00:17.233 54%



```
Type "help()" for more information.
>>> 1.81641
1.81641
Button pressed
Button released
Button pressed
Button released
1.81641
1.81641
Button pressed
Button released
```

WOKWI
Simulate IoT Projects in Your Browser

ARDUINO DAY 2023

UNIVERSIDAD DEL QUINDÍO

ACCESO A PERIFÉRICOS



March 25th
day.arduino.cc
[#ArduinoDay2023](https://twitter.com/ArduinoDay2023)

“Hola Mundo” en Microcontroladores

```
from machine import Pin
from utime import sleep_ms
```

```
ON_BOARD_PIN = 2
led_pin = Pin(ON_BOARD_PIN, Pin.OUT)
```

```
while True:
    led_pin.value(not led_pin.value())
    sleep_ms(1000)
```

Configuración
del pin

Ciclo infinito
haciendo
parpadeo del
led



ARDUINO
DAY 2023



Lectura de un pin digital

```
from machine import Pin
from utime import sleep_ms

LED = 2
BUTTON=0
led_pin = Pin(LED, Pin.OUT)
button_pin = Pin(BUTTON, Pin.IN)

while True:
    print(button_pin.value())
    led_pin.value(not led_pin.value())
    sleep_ms(1000)
```

Configuración
de los pines

Parpadeo
más lectura
de botón en
ciclo infinito



ARDUINO
DAY 2023



UNIVERSIDAD
DEL QUINDÍO

Lectura de voltaje de un pin analógico

```
from machine import Pin
from machine import ADC
from time import sleep

pin = Pin(34)
adc=ADC(pin)
adc.atten(ADC.ATTN_11DB)
while True:
    val1=adc.read()
    val2=3.3*val1/4095
    print('Raw value: ',val1,' and voltage: ',val2)
    sleep(1)
```

Configuración
del pin



Lectura de
voltaje en
ciclo infinito



ARDUINO
DAY 2023



UNIVERSIDAD
DEL QUINDÍO

PWM - Control de Servomotor

WOKWI SAVE SHARE Arduino-Day-2023-ADC-PWM Docs

```
main.py diagram.json  
1 from machine import Pin  
2 from machine import ADC  
3 from machine import PWM  
4 from time import sleep_ms  
5  
6  
7 servo = PWM(Pin(15), freq=50)  
8 pin = Pin(34)  
9 adc=ADC(pin)  
10 adc atten(ADC.ATTN_11DB)  
11 while True:  
12     val1=adc.read()  
13     pwm=51/4095*val1+51  
14     servo.duty(int(pwm))  
15     sleep_ms(100)  
16  
17  
18  
19
```

Simulation

00:14.421 99%



ARDUINO
DAY 2023



UNIVERSIDAD
DEL QUINDÍO

0 → 1ms
4095 → 2ms

$1\text{ms} \times 50\text{Hz} \times 1023 \cong 51$
 $2\text{ms} \times 50\text{Hz} \times 1023 \cong 102$

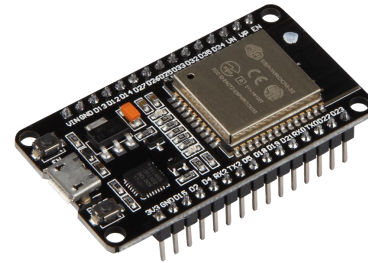
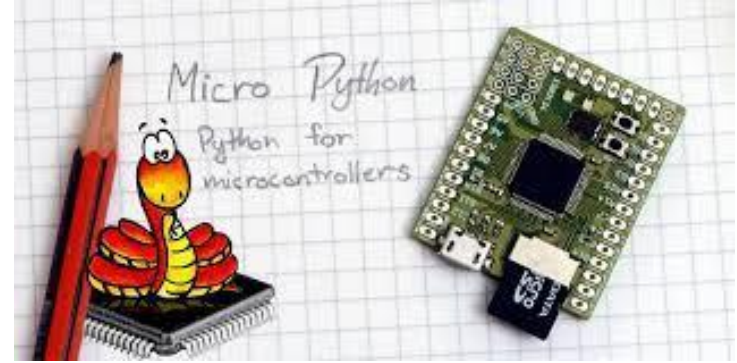
CARACTERÍSTICAS AVANZADAS



March 25th
day.arduino.cc
[#ArduinoDay2023](https://twitter.com/ArduinoDay2023)

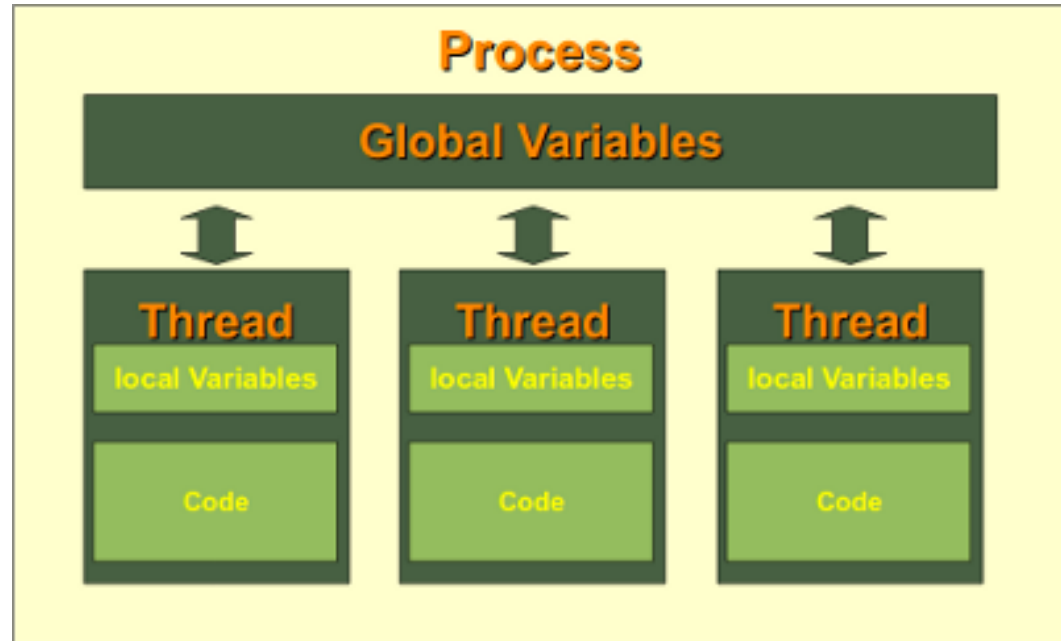
Hilos y FSMs en Micropython

- ❑ Threads
- ❑ Finite State Machines (FSMs)



Threads - Hilos

- ❑ Permiten concurrencia
- ❑ Ayudan organización de las tareas



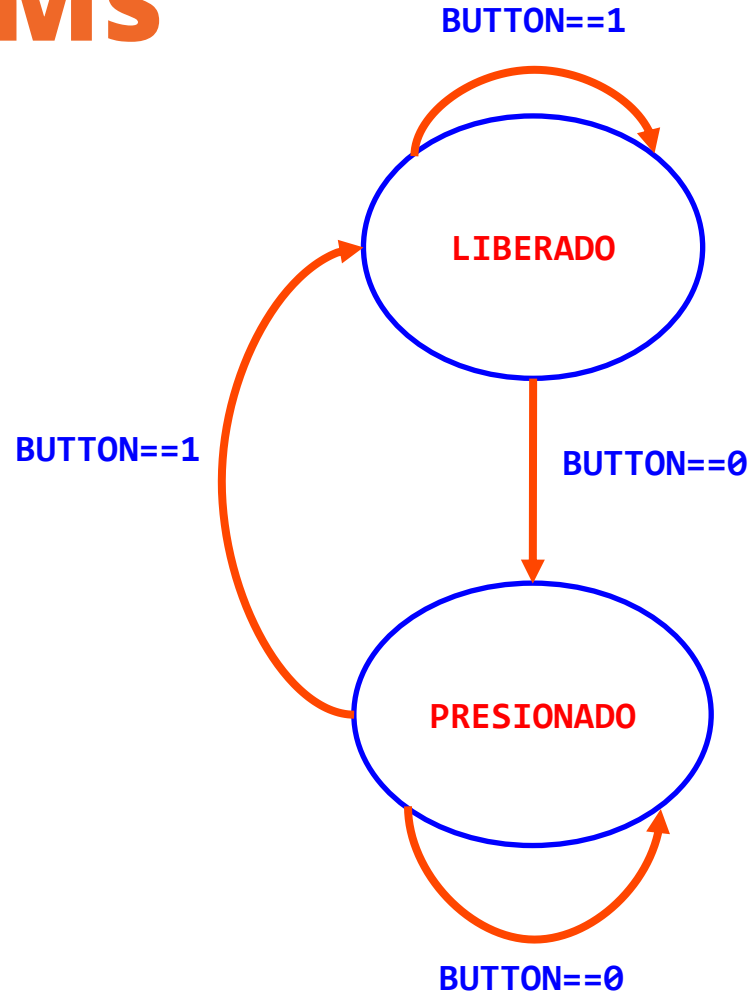
ARDUINO
DAY 2023



UNIVERSIDAD
DEL QUINDÍO

FSMs

- ❑ Mejora control de la aplicación
- ❑ Reduce la posibilidad de aparición de errores
- ❑ Facilita programación en tiempo real



ARDUINO
DAY 2023



UNIVERSIDAD
DEL QUINDÍO

REDUCCIÓN DEL CONSUMO ENERGÉTICO

- ❑ La CPU y el hardware de radio frecuencia usan la mayor parte de la energía
 - *Los dispositivos IoT devices envían datos periódicamente → Es una buena práctica usar los modos de bajo consumo (sleep modes) disponibles en algunos microcontroladores*
 - *Mejoran enormemente el desempeño en dispositivos energizados mediante baterías*
- ❑ Los modos de bajo consumo más usados en el ESP32 son *light sleep* y *deep sleep*

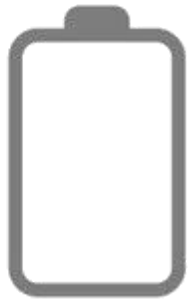


ARDUINO
DAY 2023

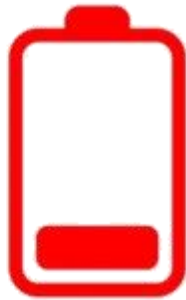


UNIVERSIDAD
DEL QUINDÍO





Active



Light Sleep



Modem Sleep



Deep Sleep



Hibernation



Turn Off

Power mode	Description		Power consumption	
Modem-sleep	The CPU is powered on.	240 MHz *	Dual-core chip(s)	30 mA ~ 68 mA
			Single-core chip(s)	N/A
		160 MHz *	Dual-core chip(s)	27 mA ~ 44 mA
			Single-core chip(s)	27 mA ~ 34 mA
		Normal speed: 80 MHz	Dual-core chip(s)	20 mA ~ 31 mA
			Single-core chip(s)	20 mA ~ 25 mA
Light-sleep	-		0.8 mA	
Deep-sleep	The ULP coprocessor is powered on.		150 μ A	
	ULP sensor-monitored pattern		100 μ A @1% duty	
	RTC timer + RTC memory		10 μ A	
Hibernation	RTC timer only		5 μ A	
Power off	CHIP_PU is set to low level, the chip is powered off.		1 μ A	



ARDUINO DAY 2023



UNIVERSIDAD DEL QUINDÍO

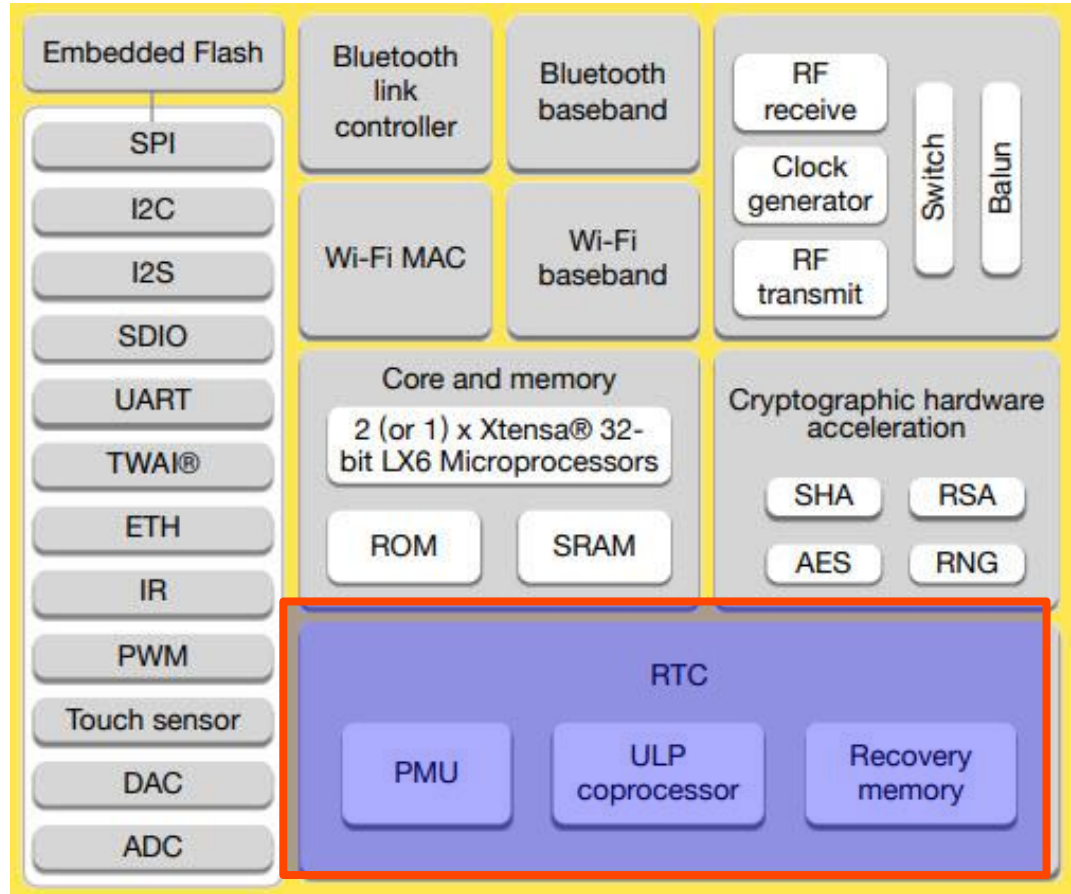
Modo Deep Sleep



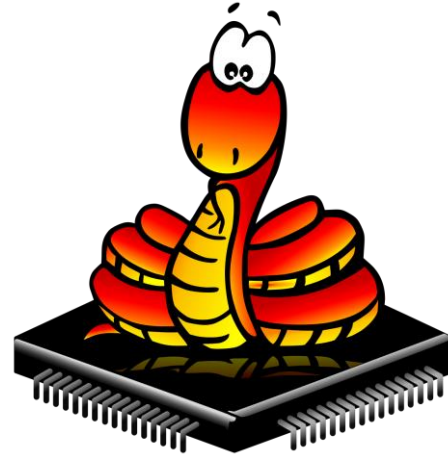
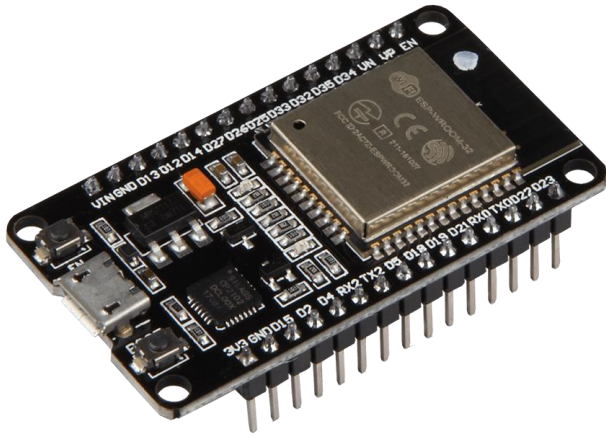
ARDUINO
DAY 2023



UNIVERSIDAD
DEL QUINDÍO



Modos *Deep sleep* y *light sleep*



ARDUINO
DAY 2023



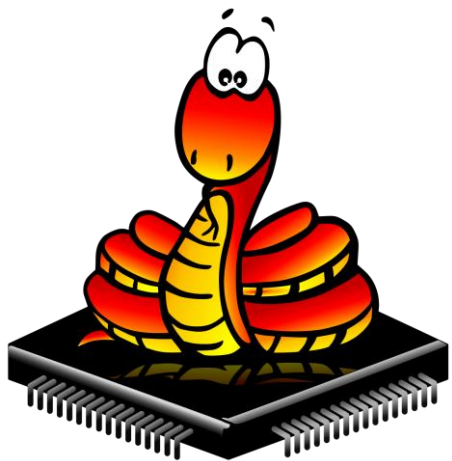
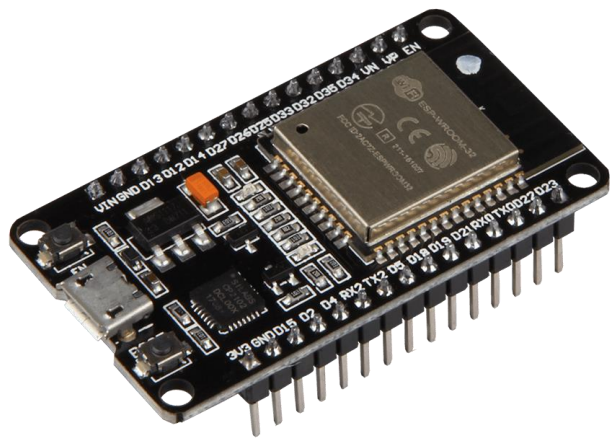
UNIVERSIDAD
DEL QUINDÍO

ENVÍO DE DATOS A PLATAFORMAS EN LA NUBE



March 25th
day.arduino.cc
[#ArduinoDay2023](https://twitter.com/ArduinoDay2023)

Enviar datos desde ESP32



Almacenamiento

- ❑ Firebase almacena la información en forma de documentos como objetos tipo JSON (*JavaScript Object Notation*)

```
{  
  "temp": 30,  
  "hum": 60  
}
```



ARDUINO
DAY 2023



UNIVERSIDAD
DEL QUINDÍO



Enviar datos desde ESP32

```
# Set URL for database
firebase.setURL("https://arduino-day-2023-default-rtdb.firebaseio.com/")

# Simulated temperature
temp=30
hum=60

while True:
    temp=temp+random.uniform(-0.1, 0.1)
    hum=hum+random.uniform(-0.1, 0.1)

    print('Sending data, temperature: ',temp,' humidity: ',hum)
    firebase.addTo("sensor_data",{ "temp":temp,"hum":hum,"timestamp": { ".sv" : "timestamp" } })
    time.sleep_ms(10000)
```



ARDUINO
DAY 2023



UNIVERSIDAD
DEL QUINDÍO

<https://github.com/ckoever/micropython-firebase-realtime-database>

Configurar índice en reglas

```
1 {  
2   "rules": {  
3     "sensor_data": {  
4       ".indexOn": ["timestamp"]  
5     },  
6     ".read": true,  
7     ".write": true  
8   }  
9 }
```

Acceder a los datos

❑ Obtener todos los datos

https://arduino-day-2023-default-rtdb.firebaseio.com/sensor_data.json

❑ Obtener datos en un rango

https://arduino-day-2023-default-rtdb.firebaseio.com/sensor_data.json?orderBy=%22timestamp%22&startAt=1679746192134&endAt=1679746202175



ARDUINO
DAY 2023



UNIVERSIDAD
DEL QUINDÍO



Alexander López-Parrado
parrado@uniquindio.edu.co ✨

Thank
you! ✨



UNIVERSIDAD
DEL QUINDÍO ✨

March 25th
week.arduino.cc
#ArduinoDay2023

