

Arduino IDE

- [¿Qué es Arduino IDE?](#)
- [Preparar Arduino IDE para Alvik: Instalar Placa ESP32 y librería Arduino Avik](#)
- [Preparar ALVIK para Arduino IDE: Modo Bootloader](#)
- [Arduino IDE sin IoT: Un pequeña danza](#)
- [Arduino IDE sin IoT: Evita obstáculos](#)
- [Arduino IDE sin IoT: Coche a control remoto](#)
- [Internet de las cosas IoT](#)
- [Arduino IDE con IoT: Escaneo Wifi](#)
- [Arduino IDE con IoT: Arduino Cloud](#)
- [Arduino IDE con IoT: Coche teledirigido](#)
- [GPIO del ESP32](#)
- [Arduino IDE con IoT: ESP32 + Sensores externos + IoT](#)
- [Predictor meteorológico](#)

¿Qué es Arduino IDE?

Necesitarás el **entorno de desarrollo Arduino IDE** (IDE, Integrated development environment) (aquí <https://www.arduino.cc/en/Main/Software> para descargarlo)

En Linux puede salir este mensaje "can't open device "/dev/ttyUSB0": Permission denied" donde 0 puede ser otro número, la solución [aquí](#)

Está constituido por un **editor de texto** para escribir el código, un **área de mensajes**, una barra de herramientas con botones para las funciones comunes, y una serie de menús.

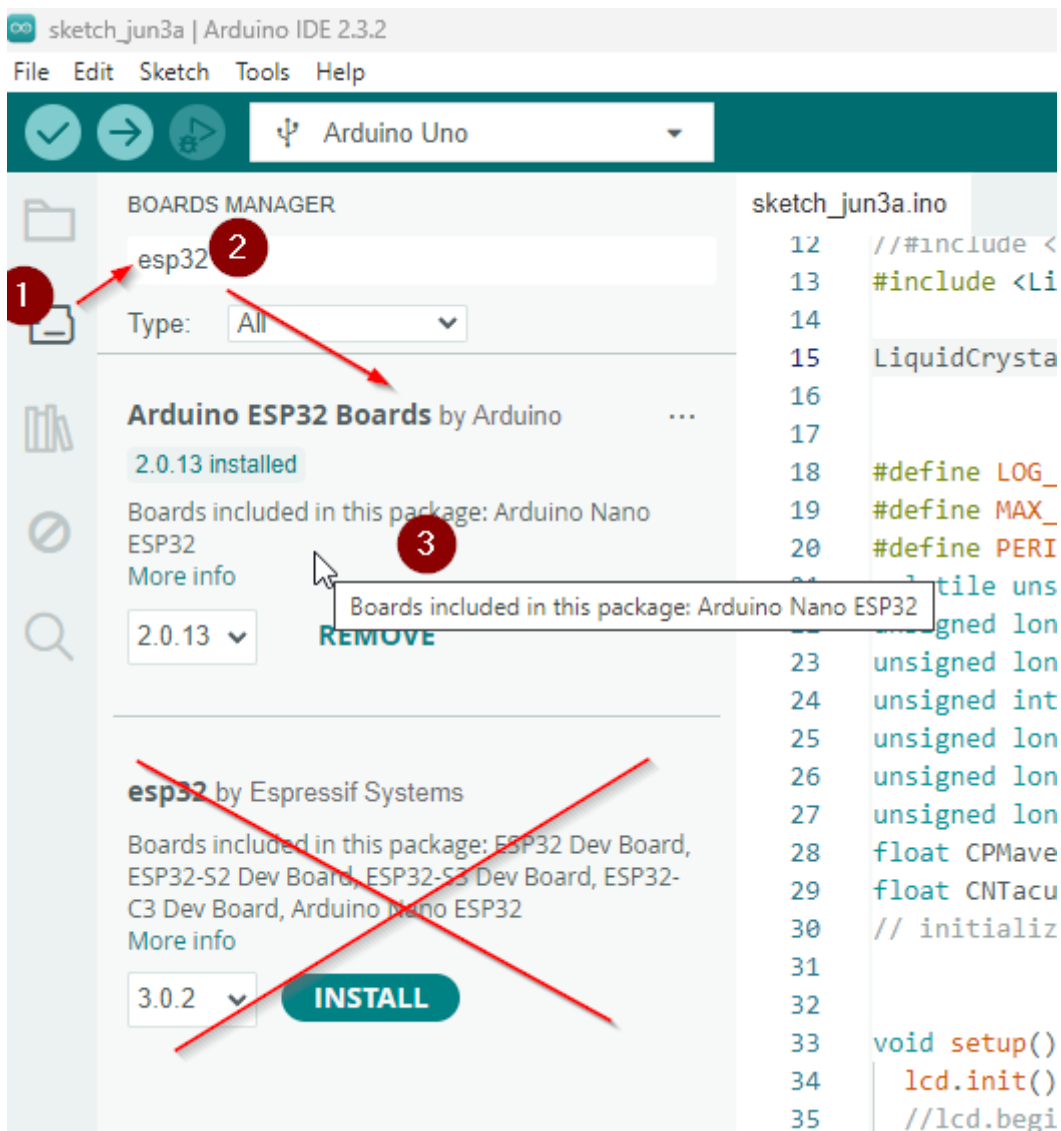
Arduino utiliza para escribir el código fuente o programa de aplicación lo que denomina "sketch" (programa). Estos programas son escritos en el editor de texto. Existe la posibilidad de cortar/pegar y buscar/remplazar texto.



Preparar Arduino IDE para Alvik: Instalar Placa ESP32 y libreria Arduino Avik

Instalar placa Arduino ESP32 Boards by Arduino

Lo primero que tenemos que hacer es instalar la placa Arduino ESP32 tal y como dice esta captura



O [este vídeo a partir de 9:30](#) (pongo el vídeo pues es interesante si quieres aprender más sobre Arduino ESP32)

<https://www.youtube.com/embed/R51tf66es9w?si=DAfiKpm7tBEcu7HVstart=570>

Instalar Librería Arduino_Alvik.h

Las funciones que tiene la librería son prácticamente las vistas en las APIs, ver

<https://libros.catedu.es/books/arduino-alvik/page/arduino-alvik-api>

Para ejecutarlo en el Arduino IDE tenemos que tener esta librería que es fácilmente instalable:

drive | Arduino IDE 2.3.2

File Edit Sketch Tools Help

Arduino Nano ESP32

LIBRARY MANAGER

Arduino_Alvik.h 2

Type: All

Topic: All

1 3

Arduino_Alvik by Arduino, Giovanni di Dio...

1.0.1 installed

Library to code Arduino Alvik robot This library is used to code Arduino Alvik, examples show y...
More info

1.0.1 **REMOVE**

Arduino_AlvikCarrier by Arduino, Giovanni di Dio...

Library and firmware for Arduino Alvik Carrier board This library is used to build the firmware used...
More info

1.0.1 **INSTALL**

```

1  /*
2  |
3  |
4  |
5  |
6  |
7  |
8  |
9  |
10 /*
11 |
12 #incl
13 |
14 Ardui
15 |
16 void·
17 ··alv
18 }
19 |
20 void·
21 ··alv
22 ··del
23 ··alv
24 ··del
25 }

```

Tal y como dice <https://docs.arduino.cc/tutorials/alvik/setting-alvik-arduino-ide/>

- **Arduino_Alvik:** Esta es la librería principal que usaremos en nuestros programas y contiene comandos de alto nivel para controlar el "cerebro" del Alvik, que es la placa Nano ESP32.

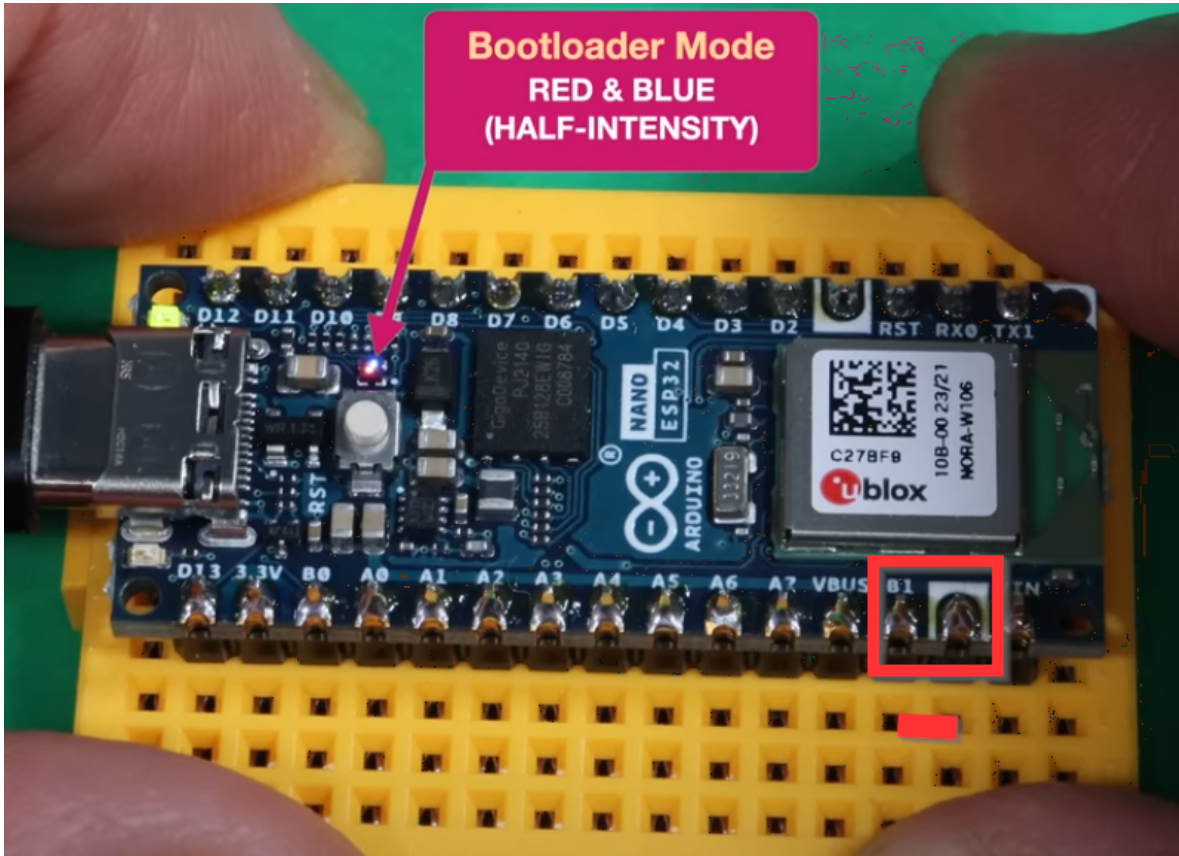
- **Arduino_AlvikCarrier:** Esta biblioteca está diseñada para la placa STM del dispositivo y resulta útil en situaciones donde se requiere un control más preciso de los comandos. Permite desarrollos más complejos, especialmente cuando se requiere una mayor integración con el hardware. Fuera del contexto de este curso.

Preparar ALVIK para Arduino IDE: Modo Bootloader

¿Qué es eso del Boodloader? Es un pequeño programa que esta en el microcontrolador (Arduino, ESP32, etc...) que permite que arranque la placa y espere las instrucciones del programa del usuario, digamos que es como un "pequeño sistema operativo de arranque" por ejemplo en el Arduino, se ejecuta en un poco de tiempo cuando arranca la placa o se resetea, y espera el programa IDE por el puerto USB, si llega (él comprueba que es un IDE y no otra cosa) lo almacena en un sitio de la memoria Flash y lo ejecuta, sino, pues ejecuta el que ya esta cargado. El bootloader hace que parpadee el led 13 de un Arduino UNO y se reserva un trozo de memoria para el Bootloader (en el Arduino UNO ocupa sólo 0.5K de los 32K que tiene disponibles el micro para ello). En nuestro caso el ESP32 Nano Arduino igual pero cuando cargamos el micropython nos cargamos ese bootloader por otro que tiene el compilador microPython. Lo de "quemar" o "flashear" el bootlader nos lo podemos encontrar en los cursos de Aularagón en el [Zigbee de domótica con Raspberry](#), o a la hora de [quemar el Nano Arduino como Arduino UNO en el curso de mClon](#)

Tal y como dice en <https://docs.arduino.cc/tutorials/alvik/setting-alvik-arduino-ide/>

1. cortocircuitar B1 y GND



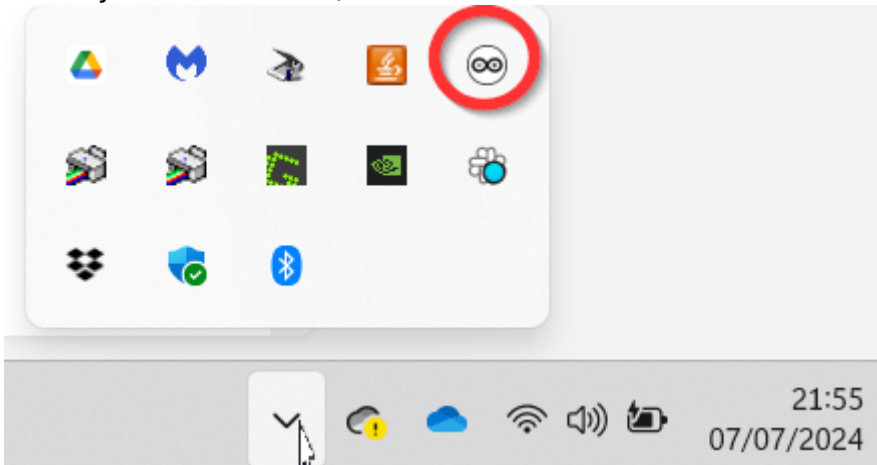
Yo lo hago con un destornillador y toco las dos puntas, con cuidado de no tocar nada más.

2. **Mientras** esta cortocircuitado **PULSA EL BOTÓN RESET**



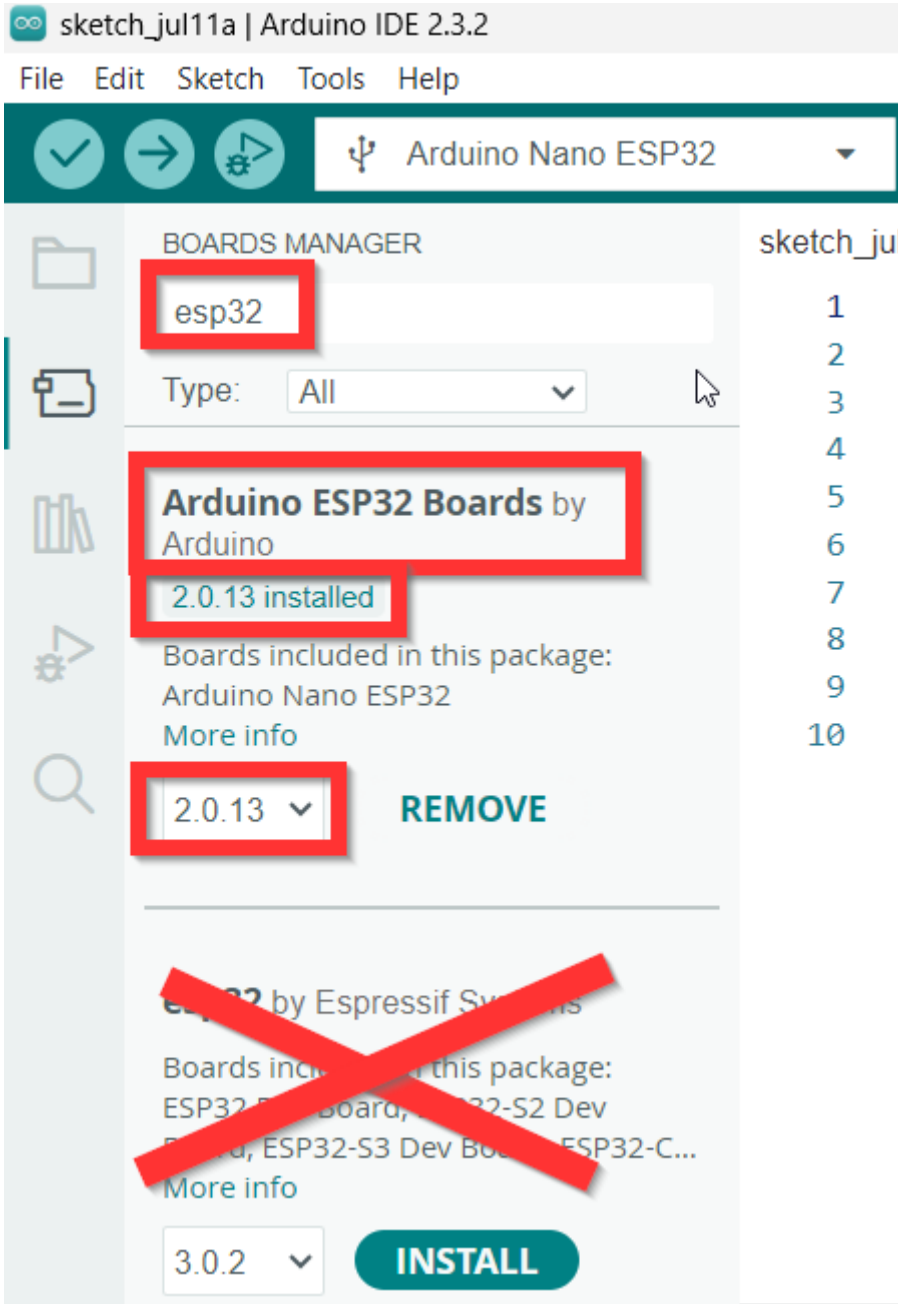
3. Soltar botón y cortocircuito pines B1 y GND y se queda el led de la placa ESP32 en color **púrpura** ver figura del paso 1, si no se queda púrpura, repetir el proceso **paciencia**

4. Abrir el programa **ARDUINO IDE PERO CERRAR OTROS**
(como el Arduino Create Agent que se queda en segundo plano abajo a la derecha , el MicroPython Installer....)

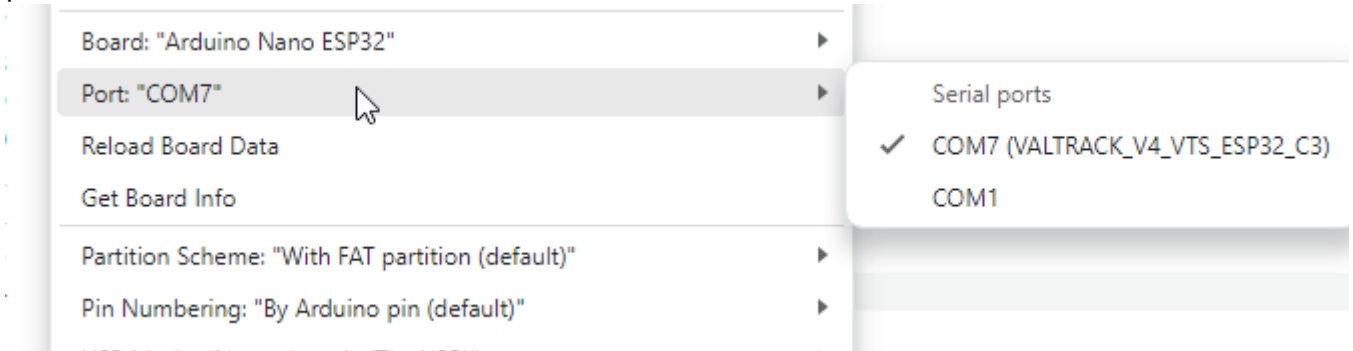




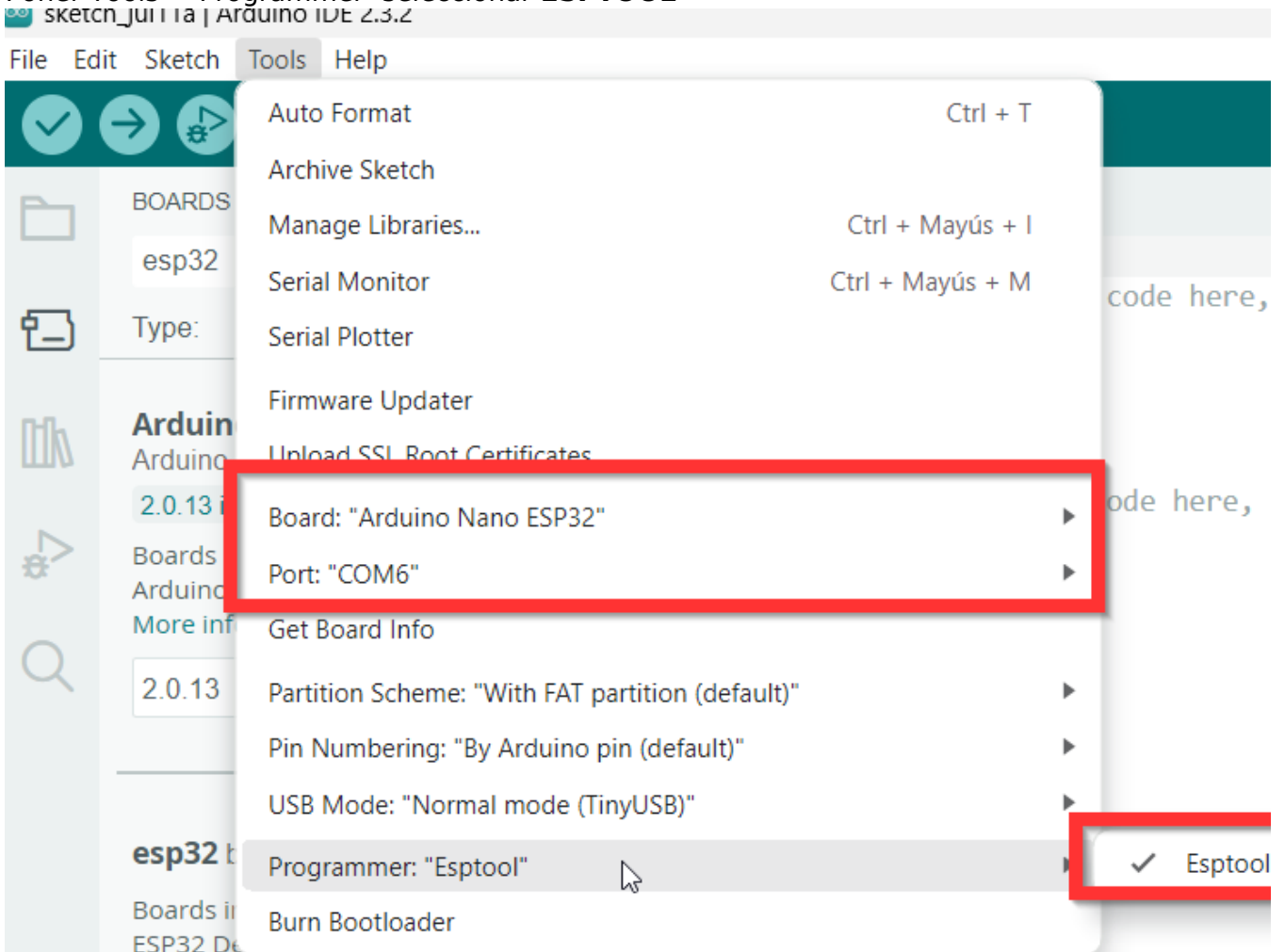
- 5. Asegurarse que esta instalado la placa **Arduino ESP32** correctamente al menos la versión 2.013 (ir a Tools-Boards-BoardManager) si hay una versión anterior, desinstalar el que hay (remove) e instalarlo de nuevo.

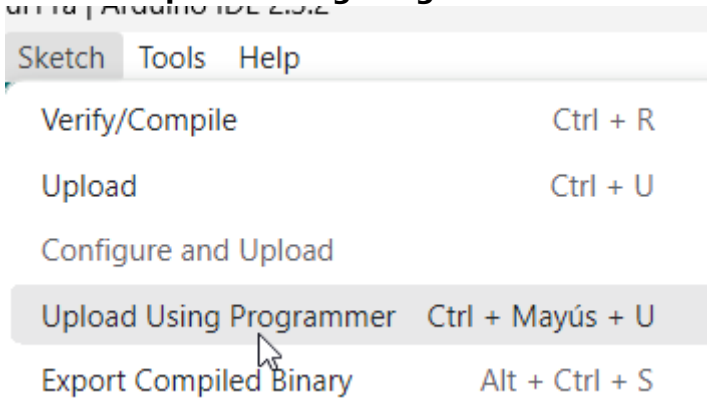


6. Ir a **Tools-Port** y seleccionar el puerto, habrá cambiado a uno nuevo, puede ser algo parecido a esto :



7. Ir a **Tools > Board - Arduino ESP32 Boards > Arduino Nano ESP32** (o esp32 > Arduino Nano ESP32)
8. Poner Tools-> Programmer- seleccionar **ESPTOOL**



9. Sketch > **Upload Using Programmer**

10. Cuando salga este mensaje pasar al paso 11

Leaving...
Hard resetting via RTS pin...

11. Apretar el botón **RESET** y ya puedes o ejecutar un programa en Arduino IDE o instalar MicroPython**Curiosidad:**

¿Por qué al poner la placa en modo Bootloader o cada vez que enciendo Arduino Alvik se enciende y se apaga el led RGB de al lado con los colores Rojo y Verde?

Solución <https://libros.catedu.es/books/arduino-alvik/page/gpio-del-esp32-ID7>

P: ¿No hay otra forma que no sea meter el destornillador y cortocircuitar los pines B1 y GND?

R: Sí, hemos elegido la anterior pues nos parece más rápida y sencilla pero en

https://github.com/arduino-libraries/Arduino_Alvik?tab=readme-ov-file#how-to-update-firmware-of-arduino-alvik-carrier tienes otro método utilizado el programa STM32 Cube Programmer

Arduino IDE sin IoT: Un pequeña danza

En la pagina https://www.arduino-libraries.info/libraries/arduino_alvik o desde https://github.com/arduino-libraries/Arduino_Alvik podemos descargarnos multitud de ejemplos de código escrito en Arduino IDE para manejar este robot

Una pequeña danza

Este sencillo programa hace mover el robot a una velocidad de 10 y va cambiando el giro de 45º a -45º cada segundo

```
#include "Arduino_Alvik.h"

Arduino_Alvik alvik;

void setup() {
  alvik.begin();
}

void loop() {
  alvik.drive(10, 45);
  delay(10000);
  alvik.drive(10, -45);
  delay(10000);
}
```

y da este error NO DEU dfu-util: No DFU capable USB device available Failed uploading: uploading error: exist status 74 ¿Por qué?

No has preparado convenientemente el ALVIK haz <https://libros.catedu.es/books/arduino-alvik/page/preparar-alvik-para-arduino-ide-modo-bootloader>



Resultado

<https://www.youtube.com/embed/1zG26a3Lf1g>

Arduino IDE sin IoT: Evita obstáculos

En la pagina https://www.arduino-libraries.info/libraries/arduino_alvik o desde https://github.com/arduino-libraries/Arduino_Alvik podemos descargarnos multitud de ejemplos de código escrito en Arduino IDE para manejar este robot

Evita obstáculos

Este sencillo programa hace mover el robot evitando obstáculos

```
#include "Arduino_Alvik.h"

Arduino_Alvik alvik;

void setup() {
  alvik.begin();
  delay(5000); // Waiting for the robot to setup
}

void loop() {
  float distance = 12.0;
  float degrees = 45.0;
  float speed = 10.0;

  float distance_l, distance_cl, distance_c, distance_r, distance_cr;

  alvik.get_distance(distance_l, distance_cl, distance_c, distance_r, distance_cr);
  delay(50);

  Serial.println(distance_c);

  if (distance_c < distance || distance_cl < distance || distance_cr < distance || distance_l
```



```
< distance || distance_r < distance) {  
  alvik.rotate(degrees);  
} else {  
  alvik.drive(speed, 0.0);  
}  
}
```

y da este error NO DEU dfu-util: No DFU capable USB device available Failed uploading: uploading error: exist status 74 ¿Por qué?

No has preparado convenientemente el ALVIK haz <https://libros.catedu.es/books/arduino-alvik/page/preparar-alvik-para-arduino-ide-modo-bootloader>

Resultado

<https://www.youtube.com/embed/QerJTBUlqSs>

Reto

Tienes muchos ejemplos en https://github.com/arduino-libraries/Arduino_Alvik/tree/main/examples

□□□□□□ □ A QUE ESPERAS? □□□□□

Arduino IDE sin IoT: Coche a control remoto

Más adelante verás un coche teledirigido con el móvil y usando internet.

Pero en este caso vamos a usar **DOS ARDUINOS ALVIKS** uno como receptor y otro con envío de órdenes

La fuente de este programa lo puedes encontrar en https://github.com/arduino-libraries/Arduino_Alvik/blob/main/examples/remote_control/remote_control.ino

Se graban los dos programas en los dos alviks, y al apretar OK uno es el que envía la órdenes y el otro le apretamos CANCEL y es el receptor

Es muy curioso que la comunicación lo hace via Wifi pero sin usar ningún punto de acceso exterior, sino entre ellos comunicación pareada (peer)

```
/*  
  This file is part of the Arduino_Alvik library.  
  
  Copyright (c) 2024 Arduino SA  
  
  This Source Code Form is subject to the terms of the Mozilla Public  
  License, v. 2.0. If a copy of the MPL was not distributed with this  
  file, You can obtain one at http://mozilla.org/MPL/2.0/.  
  
*/  
  
// This example shows how to interface 2 Alvik robots via ESPnow.  
// At startup, you can select if an Alvik is a trasmitter by pressing the "check button" or a  
// receiver by pressing "cancel button". Use arrows to move the robot.  
  
#include "Arduino_Alvik.h"  
#include <esp_now.h>
```

```
#include <WiFi.h>

Arduino_Alvik alvik;

uint8_t broadcastAddress[] = {0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF};
uint8_t myData;
esp_now_peer_info_t peerInfo;

int alvik_mode = -1; // 0 is receiver, 1 is sender

bool led_blink = false;

void setup() {
  Serial.begin(115200);
  while((!Serial)&&(millis())>3000);

  alvik.begin();

  WiFi.mode(WIFI_STA);
  if (esp_now_init() != ESP_OK) {
    Serial.println("Error initializing ESP-NOW");
    return;
  }

  while (alvik_mode == -1){
    if (alvik.get_touch_cancel()){
      alvik_mode = 0;
    }
    if (alvik.get_touch_ok()){
      alvik_mode = 1;
    }
  }
  if (alvik_mode == 0){
    esp_now_register_recv_cb(OnDataRecv);
```

```
}
else{
    memcpy(peerInfo.peer_addr, broadcastAddress, 6);
    peerInfo.channel = 0;
    peerInfo.encrypt = false;

    if (esp_now_add_peer(&peerInfo) != ESP_OK){
        Serial.println("Failed to add peer");
        return;
    }
}

void loop() {
    if (alvik_mode==0){
        alvik.left_led.set_color(led_blink, !led_blink, 0);
        alvik.right_led.set_color(!led_blink, led_blink, 0);
        delay(500);
    }
    else{
        if (alvik.get_touch_any()){
            if (alvik.get_touch_up()){
                myData = 'F';
                esp_now_send(broadcastAddress, (uint8_t *) &myData, sizeof(myData));
            }
            if (alvik.get_touch_down()){
                myData = 'B';
                esp_now_send(broadcastAddress, (uint8_t *) &myData, sizeof(myData));
            }
            if (alvik.get_touch_left()){
                myData = 'L';
                esp_now_send(broadcastAddress, (uint8_t *) &myData, sizeof(myData));
            }
            if (alvik.get_touch_right()){
                myData = 'R';
                esp_now_send(broadcastAddress, (uint8_t *) &myData, sizeof(myData));
            }
        }
    }
}
```

```

    }
    if (alvik.get_touch_center()){
        myData = 'S';
        esp_now_send(broadcastAddress, (uint8_t *) &myData, sizeof(myData));
    }
}
alvik.left_led.set_color(0, 0, led_blink);
alvik.right_led.set_color(0, 0, led_blink);
delay(100);
}
led_blink = !led_blink;
}

void OnDataRecv(const uint8_t * mac, const uint8_t *incomingData, int len){
    Serial.print(incomingData[0]);
    switch (incomingData[0]){
        case 'F':
            alvik.drive(7, 0);
            break;
        case 'B':
            alvik.drive(-7, 0);
            break;
        case 'L':
            alvik.drive(0, 45);
            break;
        case 'R':
            alvik.drive(0, -45);
            break;
        case 'S':
            alvik.brake();
            break;
    }
    Serial.println();
}
}

```

El resultado es espectacular



<https://www.youtube.com/embed/yzlXUp9fais>

Internet de las cosas IoT

El **Internet de las cosas** (Internet of Thing IoT) describe objetos físicos —o grupos de estos— con sensores, capacidad de procesamiento, software y otras tecnologías que se conectan e intercambian datos con otros dispositivos y sistemas a través de internet u otras redes de comunicación. El Internet de las cosas se ha considerado un término erróneo porque los dispositivos no necesitan estar conectados a la Internet pública. Sólo necesitan estar conectadas a una red y ser direccionables individualmente

[Fuente Wikipedia IoT Internet de las cosas CC-BY-SA](#)



[De Drawed by Wilgenbroed on FlickrTranslated by Prades97 CC BY-SA 3.0](#)

Estamos hablando de dispositivos que se conectan a internet de forma desatendida, por vía hardware (o mejor dicho firmware) a diferencia de un ordenador, tablet o móvil, donde tienes que configurar por software el dispositivo y hay un diálogo entre usuario y dispositivo sobre el uso de Internet (el software solicita tal página web, tales datos etc por voluntad del usuario o por diálogo con el usuario) Aquí los dispositivos están ya configurados de los datos que se comunican. Es decir "conectar y olvidar".

Piensa en la diferencia entre un enchufe inteligente y un ordenador, el primero es lo que se considera dentro de IoT

Desventajas: El acceso a Internet de dispositivos caseros puede generar problemas a nivel mundial:

- [el caso Mirai](#)
- [aspiradores que nos espían](#)

IoT en los cursos de Aularagón

- **Blynk:** lo que nos gusta de esta herramienta es que es casi "instantánea" o "síncrona". Esto es imprescindible con ciertos robots como el **Rover Marciano con Arduino**. Necesitamos que "gire" para evitar un obstáculo, no podemos esperar !!! Veremos con **BLYNK** un protocolo que entre el dispositivo electrónico (nuestro robot) y nosotros (en ordenador, en una APP en el móvil) la comunicación es instantánea, gracias a un servidor que hará de intermedio, que puede ser local (BLYNK LEGACY) o en Internet (BLYNK IoT).
 - **Blynk legacy** es la que se va a trabajar en
 - [Rover Marciano con Arduino](#)
 - [Arduinoblocks en el aula](#)
 - [ESP32 en el aula](#)
 - **Blynk IoT** es la que se va a trabajar con
 - [En ESP32 en el aula](#)
 - [En Smart Home ESP32](#)
- **ThinkSpeak y SmartioSpace**
 - [Smart Agriculture Kit para Micro:bit](#)
- **MQTT** El emisor envía datos, se almacenan en un servidor, y cuando puede, lo vuelca al cliente. Cliente y emisor pueden ser el dispositivo electrónico y nosotros o viceversa. Veremos que esto es lo que hace el protocolo **MQTT** y está tremendamente extendido por lo barato y fácil que es. Hace que los servidores no estén tan ocupados, por lo tanto hay varios proveedores que ofrecen este servicio gratuitamente. Hay robots como los que tienen la placa **TDR STEAM IMAGINA** que envía datos de temperatura, humedad, .. y pueden recibir datos pero no precisan de esta exigencia instantánea como un rover.
 - [ESP32 EN EL AULA](#)
 - [En Smart Home ESP32](#)
- **TELEGRAM**
 - [ESP32 EN EL AULA](#)
 - [En Smart Home ESP32](#)



- **Arduino cloud IoT**
 - [Arduino Alvik](#)
- **Cyberpi y mBot2**
 - [IoT con Cyberpi](#)

Arduino IDE con IoT: Escaneo Wifi

Desde <https://github.com/espressif/arduino-esp32/blob/master/libraries/WiFi/examples/WiFiScan/WiFiScan.ino> podemos encontrar este programa para escanear las redes wifi desde nuestro ESP32 Arduino

<https://app.arduino.cc/sketches/54b6f875-2961-4ec5-8a48-608d9dde5feb?view-mode=preview>

<https://app.arduino.cc/sketches/54b6f875-2961-4ec5-8a48-608d9dde5feb?view-mode=embed>

y da este error NO DEU dfu-util: No DFU capable USB device available Failed uploading: uploading error: exist status 74 ¿Por qué?

No has preparado convenientemente el ALVIK haz <https://libros.catedu.es/books/arduino-alvik/page/preparar-alvik-para-arduino-ide-modo-bootloader>

Instalando la librería Wifi.h

Te dará un error de compilación pues no tiene esta librería. Puedes descargar la versión última desde <https://www.arduino.cc/reference/en/libraries/wifi/>

[FUNCTIONS](#)[VARIABLES](#)[STRUCTURE](#)[LIBRARIES](#)[IOT CLOUD API](#)[GLOSSARY](#)

WiFi

Communication

Enables network connection (local and Internet) using the Arduino WiFi shield. With this library you can instantiate Servers, Clients and send/receive UDP packets through WiFi. The shield can connect either to open or encrypted networks (WEP, W The IP address can be assigned statically or through a DHCP. The library can also m. DNS.

[Go to repository](#)

Note: this library was retired and is no longer maintained.

Compatibility

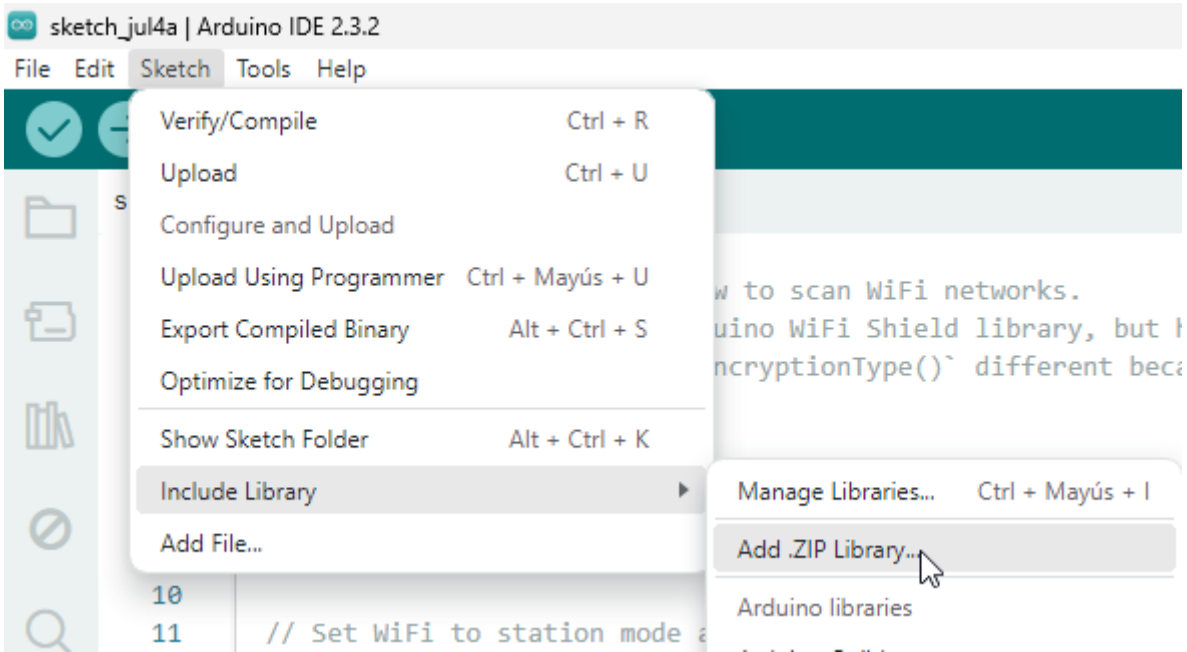
This library is compatible with **all** architectures so you should be able to use it on all Arduino boards.

Releases

To use this library, open the [Library Manager](#) in the Arduino IDE and install it from t

- 1.2.7 (latest)
- 1.2.6
- 1.2.5

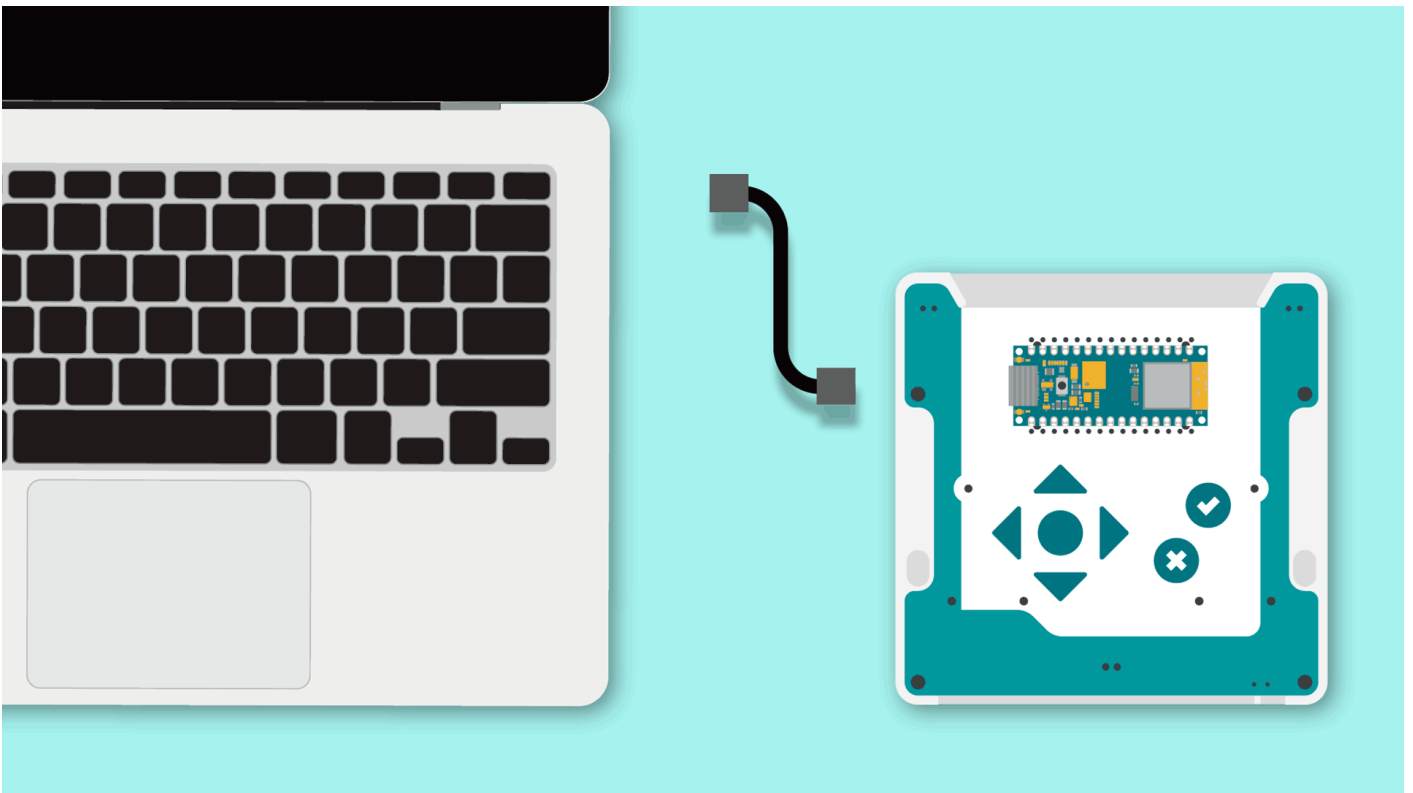
Una vez descargada (un fichero ZIP no lo descomprimas) en el editor Arduino IDE se instala desde este menú



Seleccionamos el fichero Zip que has descargado y ya tenemos la librería instalada

Compilamos

Antes de compilar CONECTAMOS NUESTRO ESP32

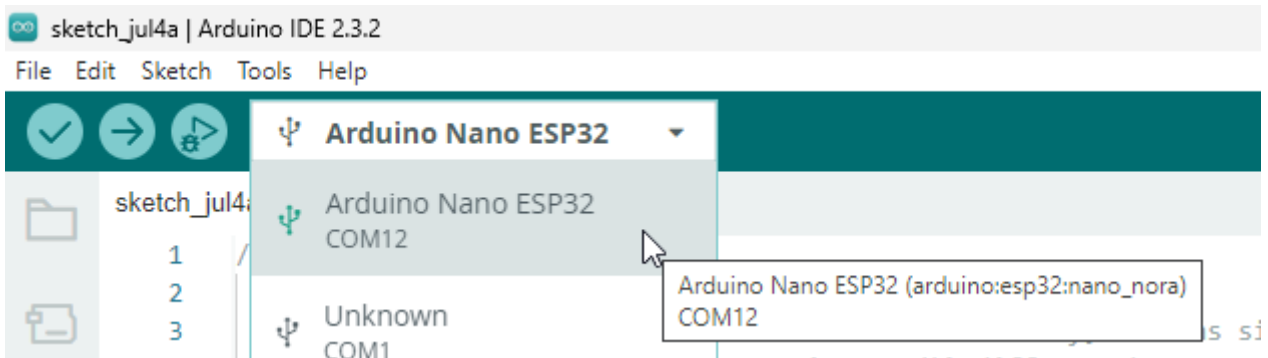


Licencia CC-BY-NC-SA origen <https://courses.arduino.cc/explore-robotics->

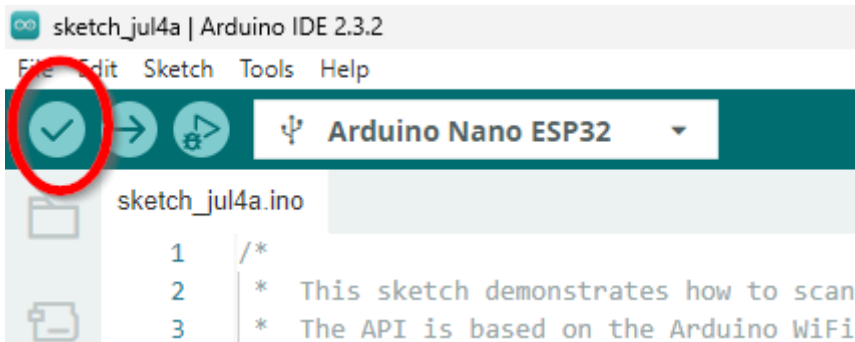
micropython/lessons/getting-started/

No hace falta encender el robot Arduino Alvik

Y seleccionamos la placa que ha reconocido

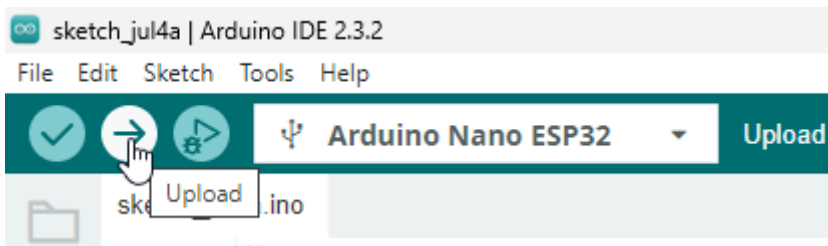


Y ya se puede compilar !!! no tiene que dar ningún fallo



Subirlo al ESP32

Pues si lo intentas subir

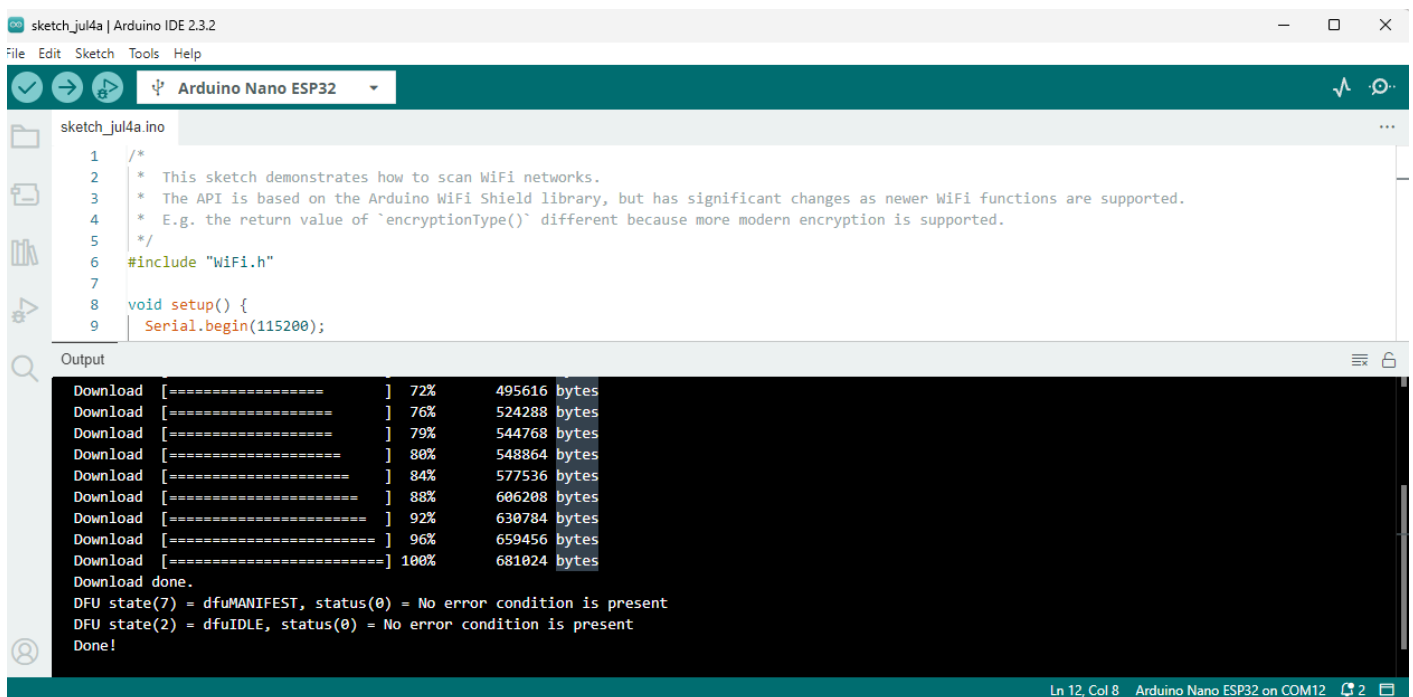


y da este error **NO DEU dfu-util: No DFU capable USB device available Failed uploading: uploading error: exist status 74 ¿Por qué?**

Lee <https://libros.catedu.es/books/arduino-alvik/page/modo-bootloader>

Resultado

Le damos a subir, y en la ventana de Output da como correcto



```
sketch_jul4a | Arduino IDE 2.3.2
File Edit Sketch Tools Help
Arduino Nano ESP32
sketch_jul4a.ino
1 /*
2  * This sketch demonstrates how to scan WiFi networks.
3  * The API is based on the Arduino WiFi Shield library, but has significant changes as newer WiFi functions are supported.
4  * E.g. the return value of `encryptionType()` different because more modern encryption is supported.
5  */
6 #include "WiFi.h"
7
8 void setup() {
9   Serial.begin(115200);
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
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
258
```

sketch_jul4a | Arduino IDE 2.3.2

File Edit Sketch Tools Help

Arduino Nano ESP32

```

1  /*
2  * This sketch demonstrates how to scan WiFi networks.
3  * The API is based on the Arduino WiFi Shield library, but has significant changes as new
4  * E.g. the return value of `encryptionType()` different because more modern encryption is
5  */
6  #include "WiFi.h"
7
8  void setup() {
9      Serial.begin(115200);

```

Output Serial Monitor x

Message (Enter to send message to 'Arduino Nano ESP32' on 'COM12')

```

Scan done
13 networks found
Nr | SSID | RSSI | CH | Encryption
1 | DIRECT-gd-EPSON-ET-4800 Series | -59 | 11 | WPA2
2 | catedu | -69 | 6 | WPA+WPA2
3 | INVITADOS_ARAGON | -75 | 1 | open
4 | COLABORADORES_ARAGON | -76 | 1 | WPA2-EAP
5 | EMBOU_76B6 | -86 | 5 | WPA+WPA2
6 | COLABORADORES_ARAGON | -87 | 6 | WPA2-EAP
7 | MIWIFI_F6Yh | -88 | 6 | WPA2
8 | INVITADOS_ARAGON | -88 | 6 | open
9 | COLABORADORES_ARAGON | -88 | 11 | WPA2-EAP

```

Dar previamente al botón de Reset

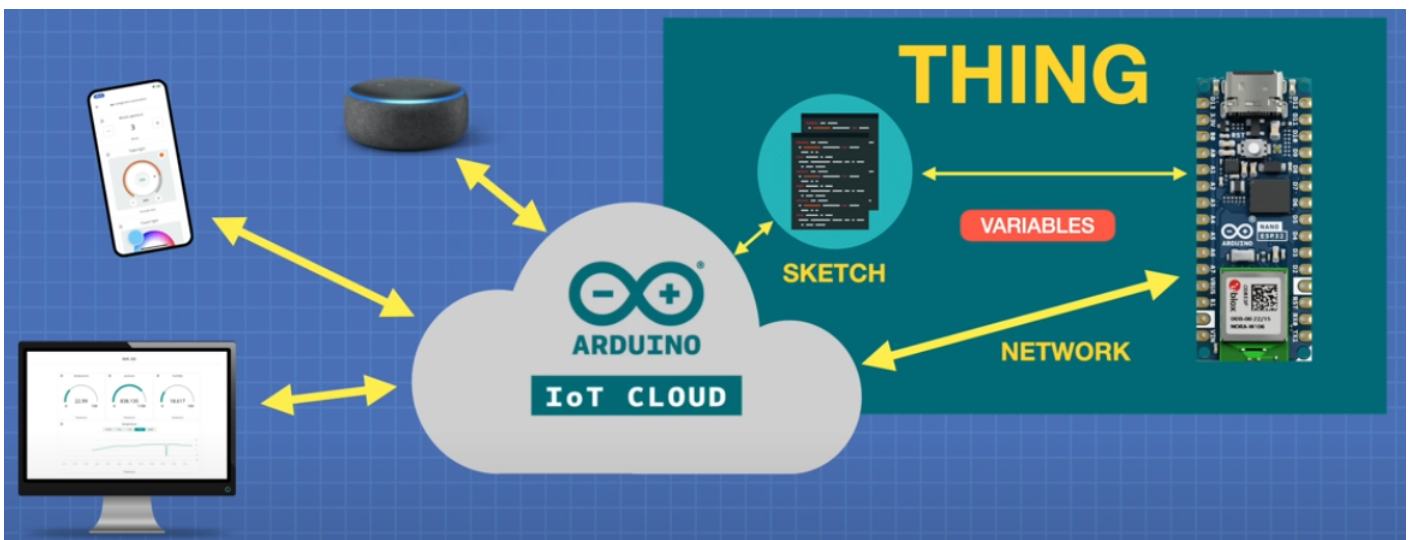
¿Puedo ahora ejecutar un programa en MicroPython?

No, tal y como dice aquí <https://libros.catedu.es/books/arduino-alvik/page/instalar-micropython> tienes que instalar el interpretador/compilador de Micropython dentro del ESP32, sino Arduino Lab for Micropython no se podrá conectar porque no lo encontrará.

Arduino IDE con IoT: Arduino Cloud

Esta plataforma <https://docs.arduino.cc/arduino-cloud/> nos permite conectar nuestras placas (Arduino v4, ESP32, et...) con un panel de control **Dashboard** y así controlarlos a distancia por Internet.

El mecanismo es sencillo, el ESP32 conectado por internet, pasa variables a un código (Sketch), a este conjunto se le llama **Thing**, y este se lo comunica a **IoT CLOUD** y la plataforma lo comunica a los paneles de control. **Dashboard** que se puede ver desde el PC o desde el móvil. El proceso también funciona al revés.



Extraído de [Youtube Exploring the Arduino Nano ESP32](#)

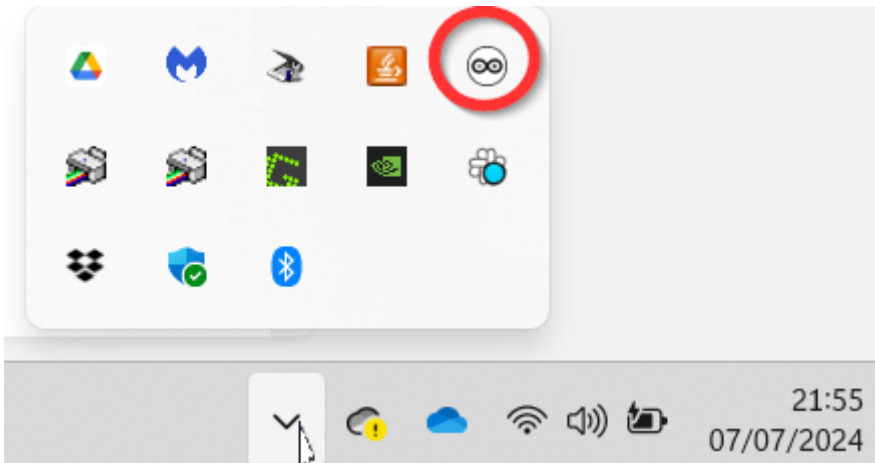
1. Creamos una cuenta en **Arduino Cloud**
2. Instalamos **Arduino Create Agent**
3. **Build the Thing** es decir preparamos nuestra placa ESP32 con el Sketch
 1. Creamos the device
 2. Creamos the thing
 3. Añadimos las variables
 4. Creamos el sketch y lo grabamos en el ESP32
4. Construimos un **Dashboard** o panel de control

PASO 1 LOGUEARSE EN ARDUINO CLOUD

En Plan permite una cuenta gratuita sólo se pueden 2 **things** ver <https://cloud.arduino.cc/plans>

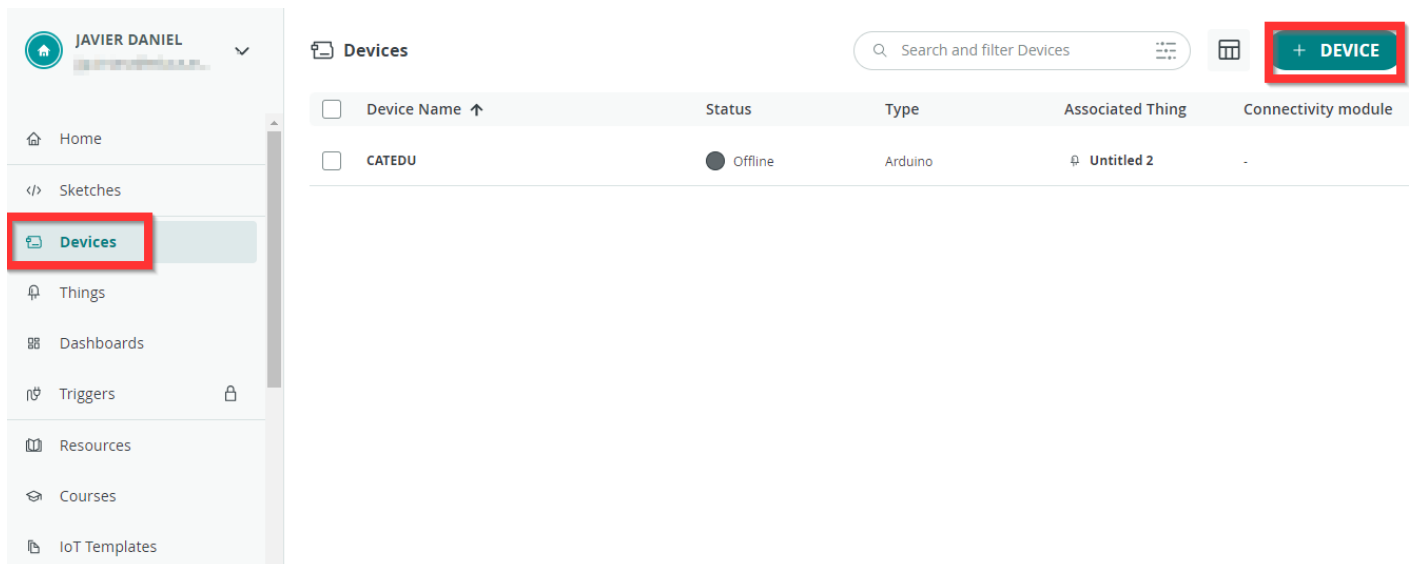
PASO 2 ADRUINO CREATE AGENT

Arduino Create Agent te lo puedes descargar desde <https://cloud.arduino.cc/download-agent>, se descarga, se ejecuta, hay que seguir los pasos, se queda en segundo plano en el PC y no tienes que preocuparte



PASO 3 Build the Thing: CREATE DEVICE

Primero añadimos un Device o placa en <https://app.arduino.cc/devices>



Elegimos placa Arduino

Setup Device

**RECOMMENDED****AUTOMATIC** Auto-sketch generation, online editing, easy upload**Arduino board** ⓘ

⌘ Arduino language (C++)

**Third party device** ⓘ

⌘ Arduino language (C++)

MANUAL Manual programming and upload, offline editing**Any Device** ⓘ

⌘ Python, MicroPython, JavaScript (NodeJS)

Si falla, ponemos la placa en modo Bootloader (ver qué es eso en

<https://libros.catedu.es/books/arduino-alvik/page/instalar-micropython>) y entonces detectará el puerto

Conectamos nuestro Arduino Alvik y saldrá un diálogo con un TOKEN on **Secret key** que lo guardaremos **ante todo no hacerlo público**



Make your device IoT-ready

To use this board you will need a Device ID and a Secret Key, please copy and save them or [download the PDF](#).

Also, keep in mind that this device authentication has a lower security level compared to other Arduino devices.

Device ID

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Secret Key



Secret key cannot be recovered

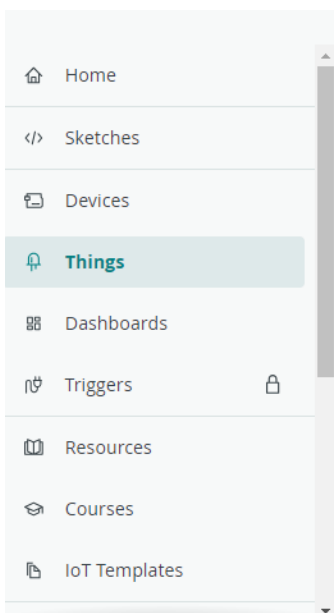
Please keep it safe, if you lose it you will have to delete and setup your device again.

I saved my device ID and Secret Key

CONTINUE

PASO 3 Build the Thing: CREATE THING

Una vez creada la placa, nos vamos a Thing, crear



Follow these simple steps to create your first Thing on Arduino Cloud



Create a Thing

A Thing is a virtual entity that lets you link your physical device to the Cloud: it includes variables, sketch and metadata



Associate Device and Network

Select the Device you want to use and enter your network credentials, so you can send and receive data remotely

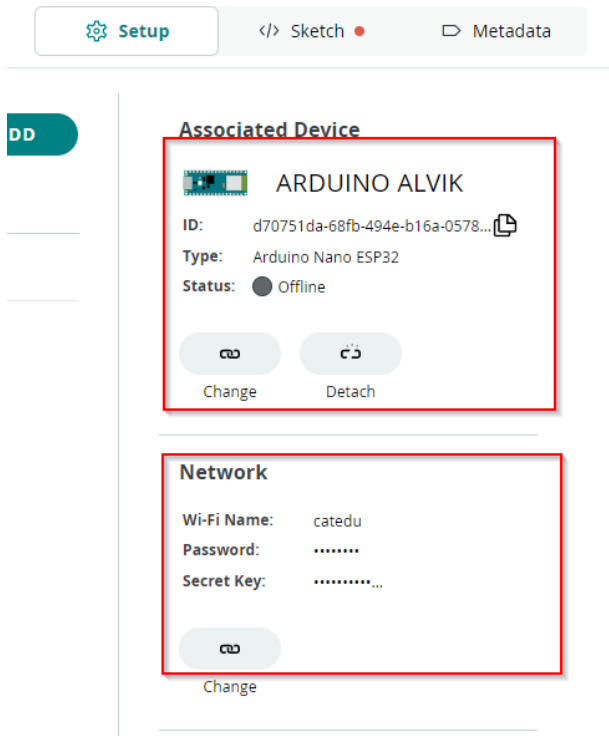


Start creating!

Easily add Cloud Variables that will be automatically included in the sketch, and that are used to exchange data with the Cloud

+ CREATE THING

Asociamos el Thing al Device, y le configuramos una red wifi (te pedirá el Secret Key)



PASO 3 Build the Thing: CREATE THING-VARIABLES

Luego añadimos variables, por ejemplo RGBverde que va a encender y apagar la luz verde, va a ser tipo Bool y Read&Write

RGBverde

Declaration

`bool` rGBverde

Type

Boolean

Variable Permission

Read & Write

Update Policy

On change

ID

72371e05-f94d-487a-bb77-f20b97f4dc18



Last Value

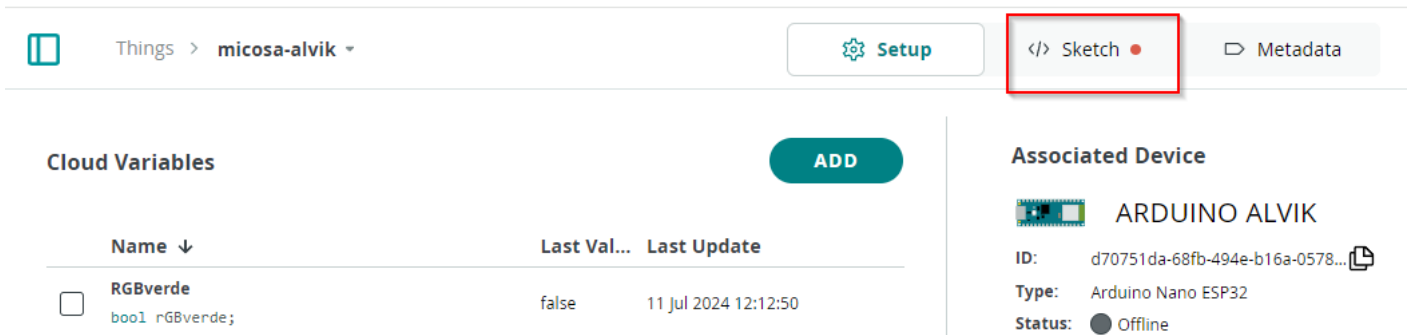
false

Last Update

11 Jul 2024 12:12:50


PASO 3 Build the Thing: CREATE THING-SKETCH

Dentro de Thinks nos vamos a SKETCH

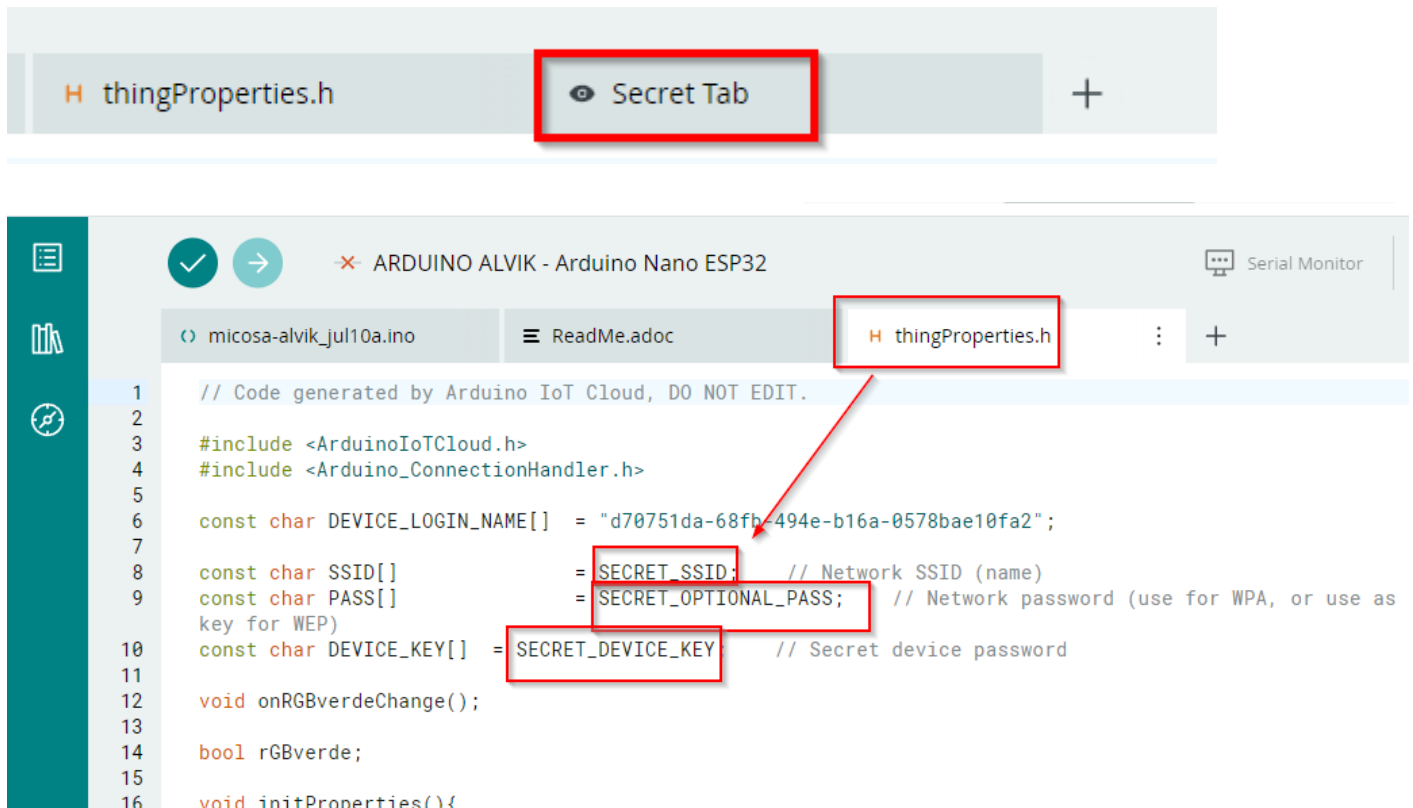


The screenshot shows the Things interface for a device named 'micosa-alvik'. The 'Sketch' tab is selected and highlighted with a red box. The interface includes a 'Setup' button, a table of Cloud Variables, and an 'Associated Device' section.

Name ↓	Last Val...	Last Update
<input type="checkbox"/> RGBverde <small>bool rGBverde;</small>	false	11 Jul 2024 12:12:50

Associated Device
 ARDUINO ALVIK
ID: d70751da-68fb-494e-b16a-0578...
Type: Arduino Nano ESP32
Status: ● Offline

y vemos que ha creado un código **thingProperties.h** que tiene que tener el SSID de la wifi, su contraseña y la palabra clave de nuestro ESP32, podemos ponerlo manualmente o nos fijamos y en **Secret Tab** estan ya puestos :

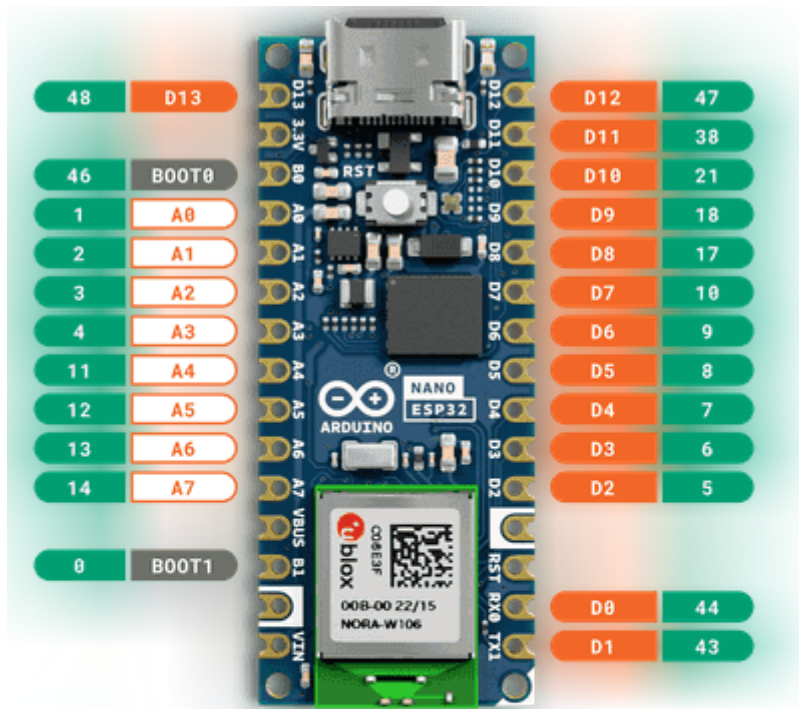


El otro script es el nombre que hemos creado en Thing y vemos que :

- LINEA 9 Esta declarada la variable que hemos añadido
- LINEA 16 Incluye la librería **thingProperties.h**
- LINEA 41 Añadimos en setup() la declaración que D13 SERÁ SALIDA **pinMode(D13,OUTPUT);**
- LINEA 60 AL 66 Añadimos en **onRGBverdeChange** una condicional, de tal manera que si la variable es cierta, que encienda el led y si es falsa que lo apague

¿Por qué es D13? ¿NO TENDRÍA QUE SER 48?

Eso ya lo hemos visto en <https://libros.catedu.es/books/arduino-alvik/page/parpadeo-led-esp32>



i ESP32 pin numbers

i Nano pin numbers

Fuente <https://docs.arduino.cc/tutorials/alvik/user-manual/>

```

/*
Sketch generated by the Arduino IoT Cloud Thing "Untitled"
https://create.arduino.cc/cloud/things/34a0aae1-c7b9-42ab-92d4-0e37bd51031f

Arduino IoT Cloud Variables description

The following variables are automatically generated and updated when changes are made to the
Thing

bool rGBverde;

```

Variables which are marked as READ/WRITE in the Cloud Thing will also have functions which are called when their values are changed from the Dashboard.

These functions are generated with the Thing and added at the end of this sketch.

```
*/  
  
#include "thingProperties.h"  
  
void setup() {  
  // Initialize serial and wait for port to open:  
  Serial.begin(9600);  
  // This delay gives the chance to wait for a Serial Monitor without blocking if none is  
found  
  delay(1500);  
  
  // Defined in thingProperties.h  
  initProperties();  
  
  // Connect to Arduino IoT Cloud  
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);  
  
  /*  
  The following function allows you to obtain more information  
  related to the state of network and IoT Cloud connection and errors  
  the higher number the more granular information you'll get.  
  The default is 0 (only errors).  
  Maximum is 4  
  */  
  setDebugMessageLevel(2);  
  ArduinoCloud.printDebugInfo();  
  
  /// MI CODIGO  
  pinMode(D13,OUTPUT);  
}  
  
void loop() {  
  ArduinoCloud.update();
```

```
// Your code here

}

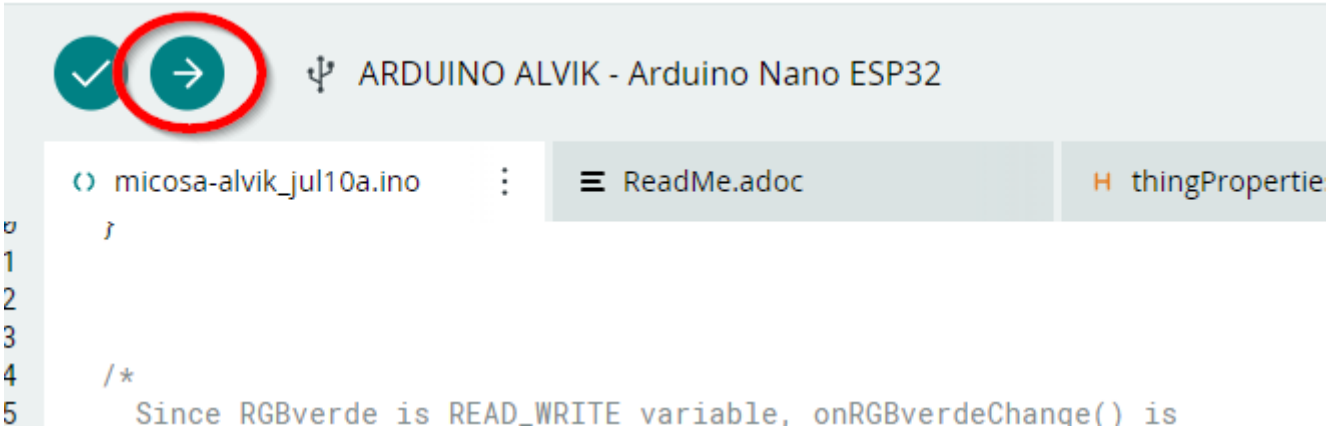
/*
  Since RGBverde is READ_WRITE variable, onRGBverdeChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onRGBverdeChange() {
  // Add your code here to act upon RGBverde change
  if (rGBverde){
    digitalWrite(D13,HIGH);

  }else{
    digitalWrite(D13,LOW);

  }
}

/*
  Since RGBrojo is READ_WRITE variable, onRGBrojoChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onRGBrojoChange() {
  // Add your code here to act upon RGBrojo change
}
```

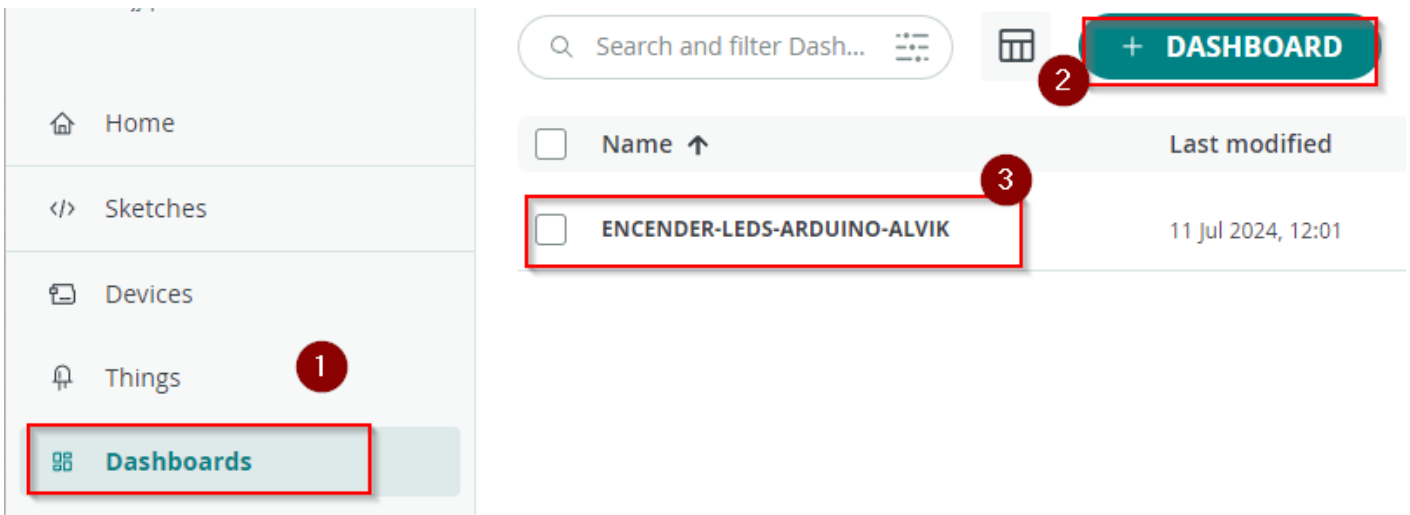
Lo subimos



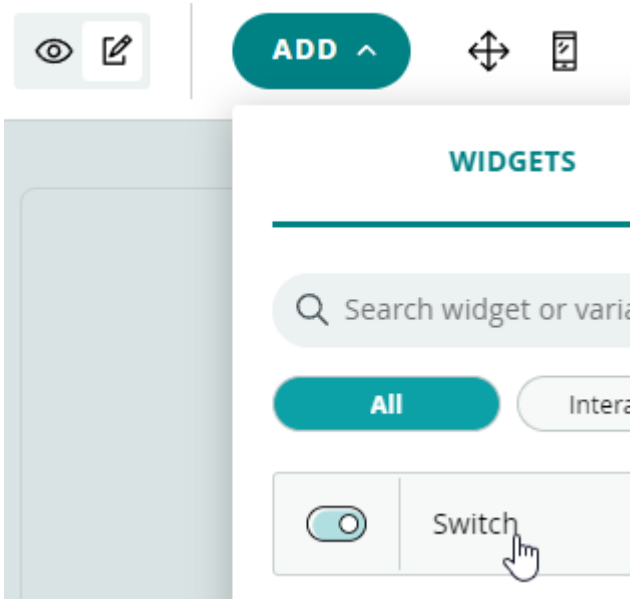
Ojo, tienes que tener el Arduino **Arduino Create Agent paso 2**

PASO 4 Dashboard

Creamos un panel de control



Y le añadimos un Switch asociado a la variable RGBverde



Podemos ver el dashboard en un teléfono móvil instalando la [APP Arduino IoT Cloud Remote](#)

Arduino IoT Cloud Remote

Arduino

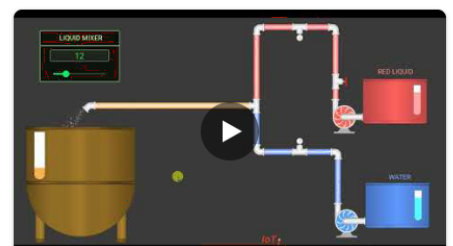
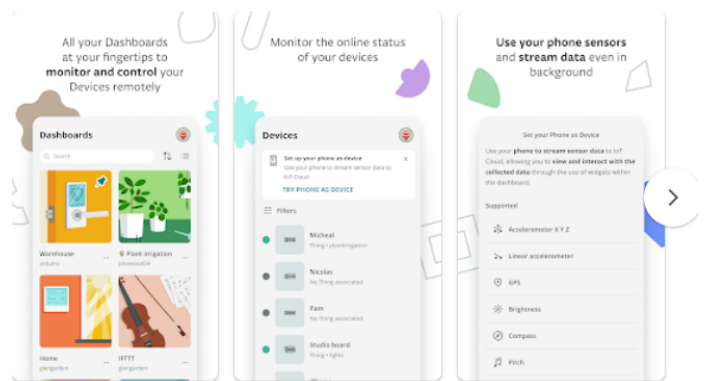
La aplicación Arduino para controlar sus proyectos de Internet de las cosas desde cualquier lugar.

3.8★
992 reseñas

100 mil+
Descargas

PEGI 3

Descargar



Al loguearse con tu cuenta, ya nos aparece el Dashboard

Resultado

<https://www.youtube.com/embed/YDVMybJtWUU>



Arduino IDE con IoT: Coche teledirigido

Aprovechamos [el programa que enciende y apaga un led por Arduino Cloud](#)

Variables

Le añadimos tres variables más :

1. **velocidad** tipo entero Read&Write
2. **giro** tipo entero Read&Write
3. **distancia** tipo float Read

Cloud Variables

ADD

Name ↓	Last Value	Last Update
<input type="checkbox"/> distancia float distancia;	0.3	11 Jul 2024 22:32:38
<input type="checkbox"/> giro int giro;	1	11 Jul 2024 22:32:31
<input type="checkbox"/> RGBverde bool rGBverde;	true	11 Jul 2024 22:31:00
<input type="checkbox"/> velocidad int velocidad;	0	11 Jul 2024 22:32:01

Sketch

En thingProperties.h añade **automáticamente** estas variables y funciones, **no tienes que añadirlas** :

```
void onGiroChange();
void onVelocidadChange();
void onRGBverdeChange();

float distancia;
int giro;
int velocidad;
```

```
bool rGBverde;
```

Pero en la función principal, nosotros vamos a poner el siguiente código :

- Línea 2 **#include "Arduino_Alvik.h"** para que incluya la librería de manejo del robot
- Línea 4 Creamos un objeto alvik **Arduino_Alvik alvik;**
- Línea 6 Creamos una variable tipo array de 5 elementos para almacenar las distancias que lee el sensor de distancia **float distances[5];**
- Línea 15 arrancamos el objeto alvik **alvik.begin();**
- Línea 41 que el alvik se mueva según la velocidad y el giro **alvik.drive(velocidad,giro);**
 - Es la instrucción principal y qué sencilla ☐☐
- Línea 42 leemos el array de distancias **alvik.get_distance(distances[0], distances[1], distances[2], distances[3], distances[4]);**
- Línea 43 de todas las distancias, sólo nos importa la 2 **distancia=distances[2];**

Nota: la instrucción 41 se han colocado dentro de loop() pero también se podría haber colocado dentro de onGiroChange();
onVelocidadChange();

```
#include "thingProperties.h"
#include "Arduino_Alvik.h"

Arduino_Alvik alvik;

float distances[5];

void setup() {
  // Initialize serial and wait for port to open:
  Serial.begin(9600);
  // This delay gives the chance to wait for a Serial Monitor without blocking if none is
  found
  delay(1500);

  alvik.begin();

  // Defined in thingProperties.h
  initProperties();
```

```
// Connect to Arduino IoT Cloud
ArduinoCloud.begin(ArduinoIoTPreferredConnection);

/*
  The following function allows you to obtain more information
  related to the state of network and IoT Cloud connection and errors
  the higher number the more granular information you'll get.
  The default is 0 (only errors).
  Maximum is 4
*/
setDebugMessageLevel(2);
ArduinoCloud.printDebugInfo();

/// MI CODIGO
pinMode(D13,OUTPUT);
}

void loop() {
  ArduinoCloud.update();
  // Your code here

  alvik.drive(velocidad,giro);
  alvik.get_distance(distances[0], distances[1], distances[2], distances[3], distances[4]);
  distancia=distances[2];

}

/*
  Since RGBverde is READ_WRITE variable, onRGBverdeChange() is
  executed every time a new value is received from IoT Cloud.
*/
```

```
void onRGBverdeChange() {
  // Add your code here to act upon RGBverde change
  if (rGBverde){
    digitalWrite(D13,HIGH);

  }else{
    digitalWrite(D13,LOW);

  }
}

/*
  Since RGBrojo is READ_WRITE variable, onRGBrojoChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onRGBrojoChange() {
  // Add your code here to act upon RGBrojo change
}

/*
  Since Velocidad is READ_WRITE variable, onVelocidadChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onVelocidadChange() {
  // Add your code here to act upon Velocidad change
}

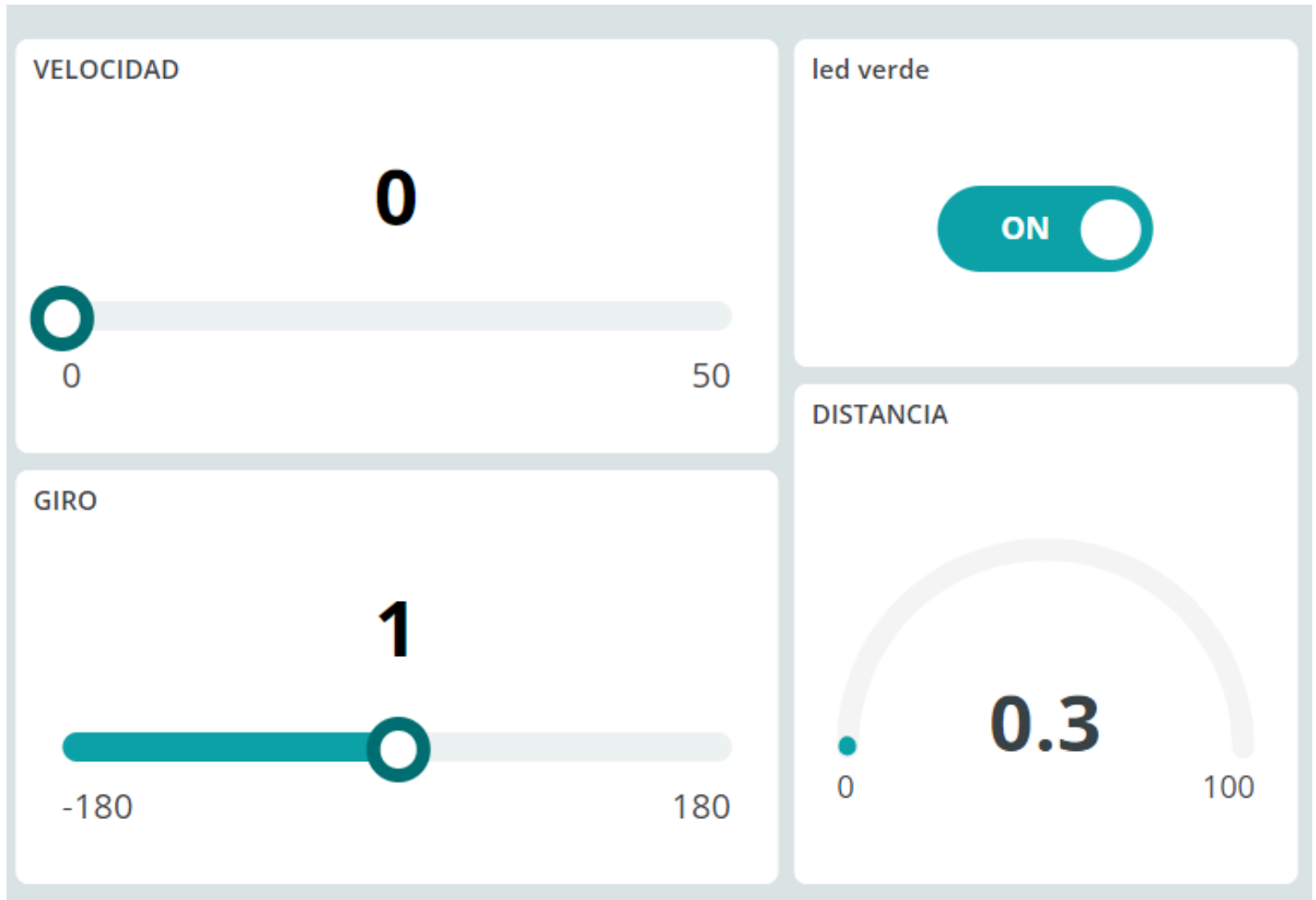
/*
  Since Giro is READ_WRITE variable, onGiroChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onGiroChange() {
  // Add your code here to act upon Giro change
}
```

Dashboard



Creemos un panel de control con:

- Un **slider** para velocidad de 0 a 50
- Un **slider** para el giro de -180 a +180
- Un **gauge** para distancia

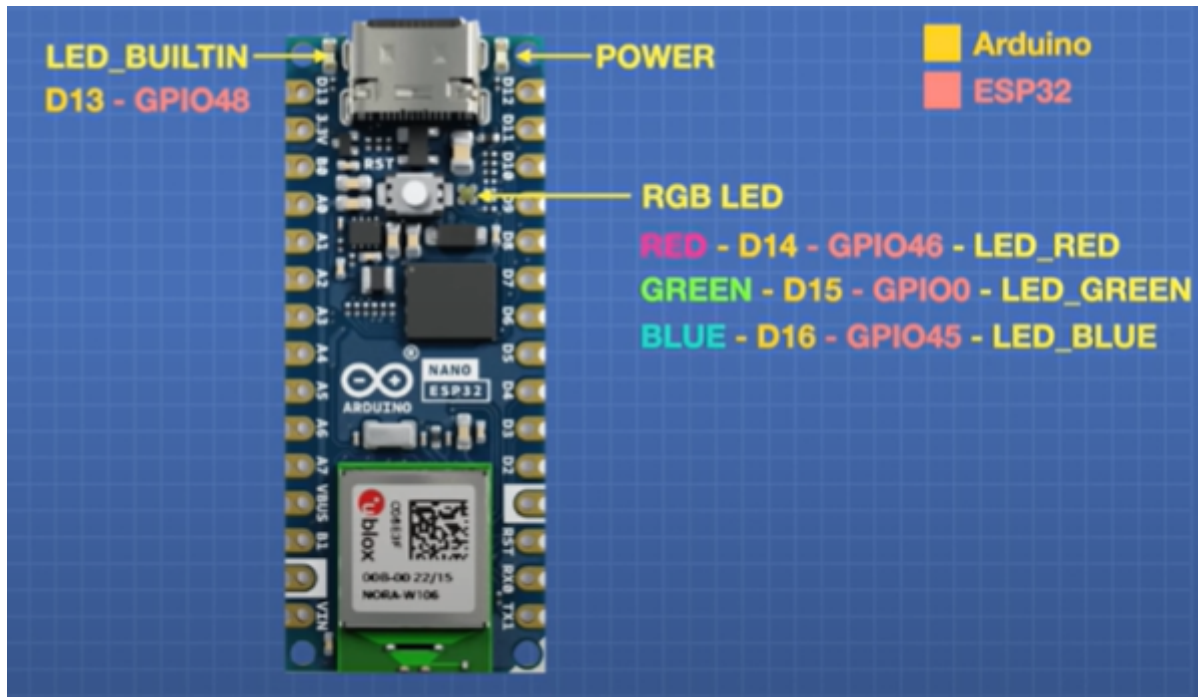


Resultado

<https://www.youtube.com/embed/5cWl3y3A3z8>

GPIO del ESP32

Mapa de los pines en el Arduino Nano ESP32

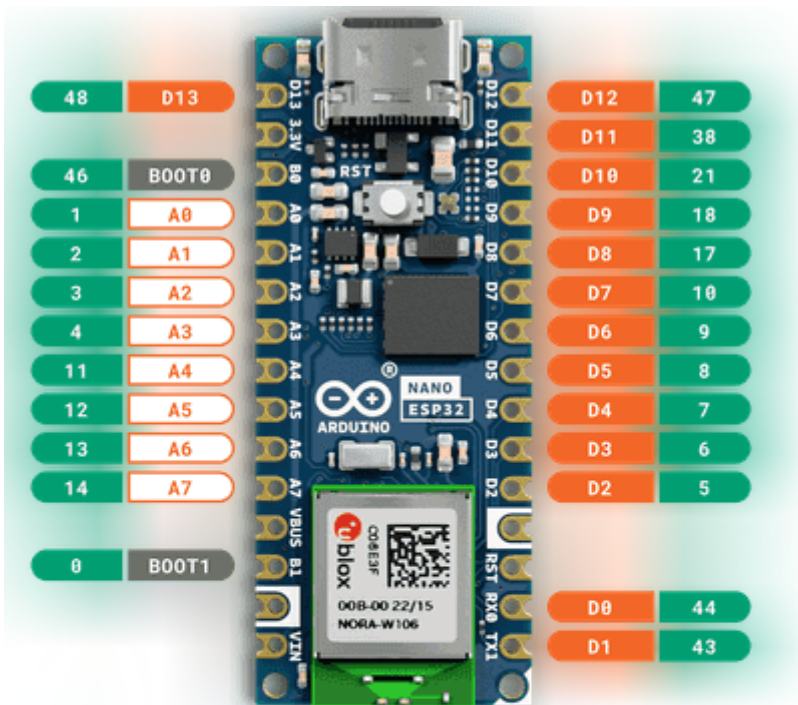


Extraído de [Youtube Exploring the Arduino Nano ESP32](#)

Como podemos observar, nuestro objetivo pues es el GPIO0

¿Dónde está físicamente los GPIO ?

Pues como podemos ver en este esquema el GPIO0 está en el pin BOOT1



i ESP32 pin numbers

i Nano pin numbers

Fuente CC-BY-SA <https://docs.arduino.cc/tutorials/alvik/user-manual/>

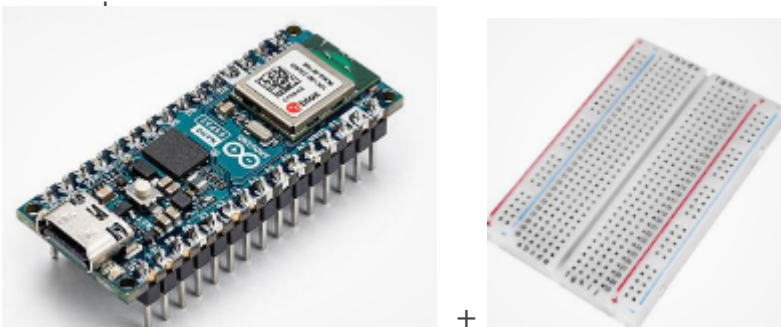
- SI USAMOS MICROPYTHON TENEMOS QUE USAR LAS VERDES
- SI USAMOS CÓDIGO ARDUINO IDE TENEMOS QUE USAR LAS ROJAS

Como puedes observar, si cortocircuitas B1 = GPIO = D15 con GND enciende el led RGB en color verde! esto pasa si Pones la placa en modo Bootloader.

Arduino IDE con IoT: ESP32 + Sensores externos + IoT

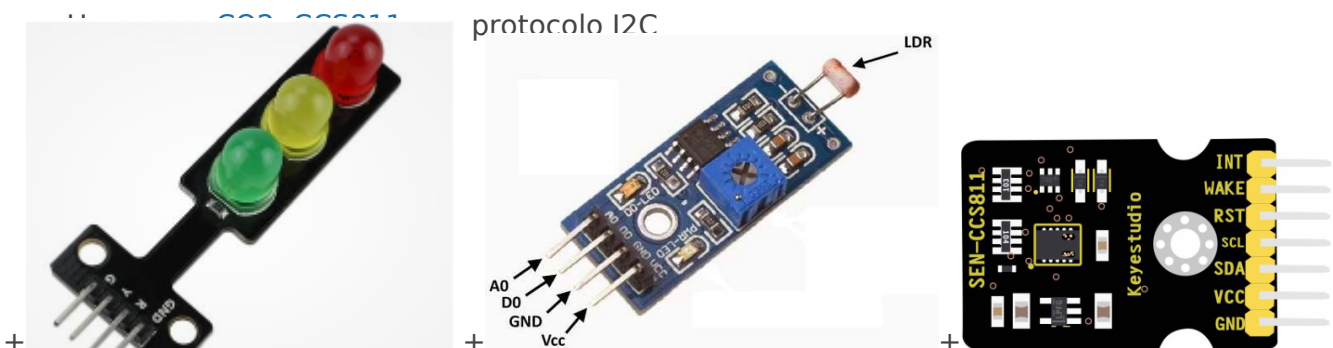
OBJETIVO

Ahora vamos a utilizar el ESP32 SIN EL ARDUINO ALVIK podemos sacar la placa microcontroladora y ponerlo en una placa *protoboard* y experimentar con sensores y actuadores estándares en el



Para ver varias posibilidades, vamos a ver estos sensores y actuadores (recomendamos ver estas páginas [actuadores](#) y [sensores](#))

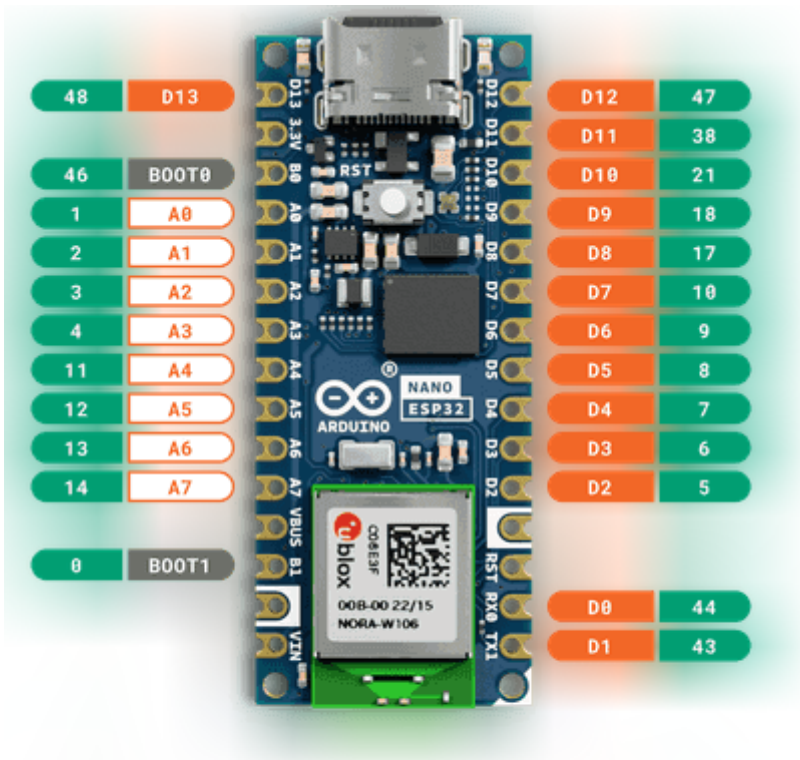
- Un led de salida simple, para practicar salida digital en mi caso voy a elegir este gracioso [semáforo](#)
- Un [sensor LDR](#) pero para practicar los dos tipos de señal,
 - con salida analógica
 - con salida digital.





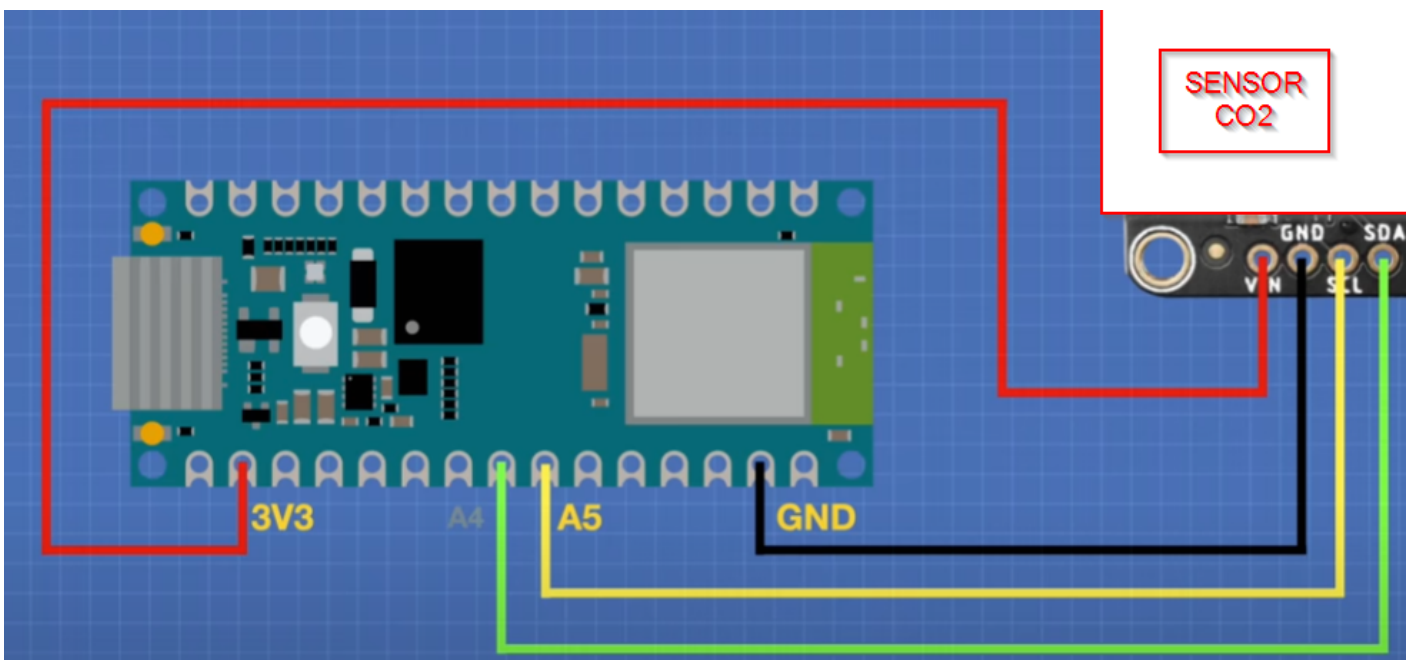
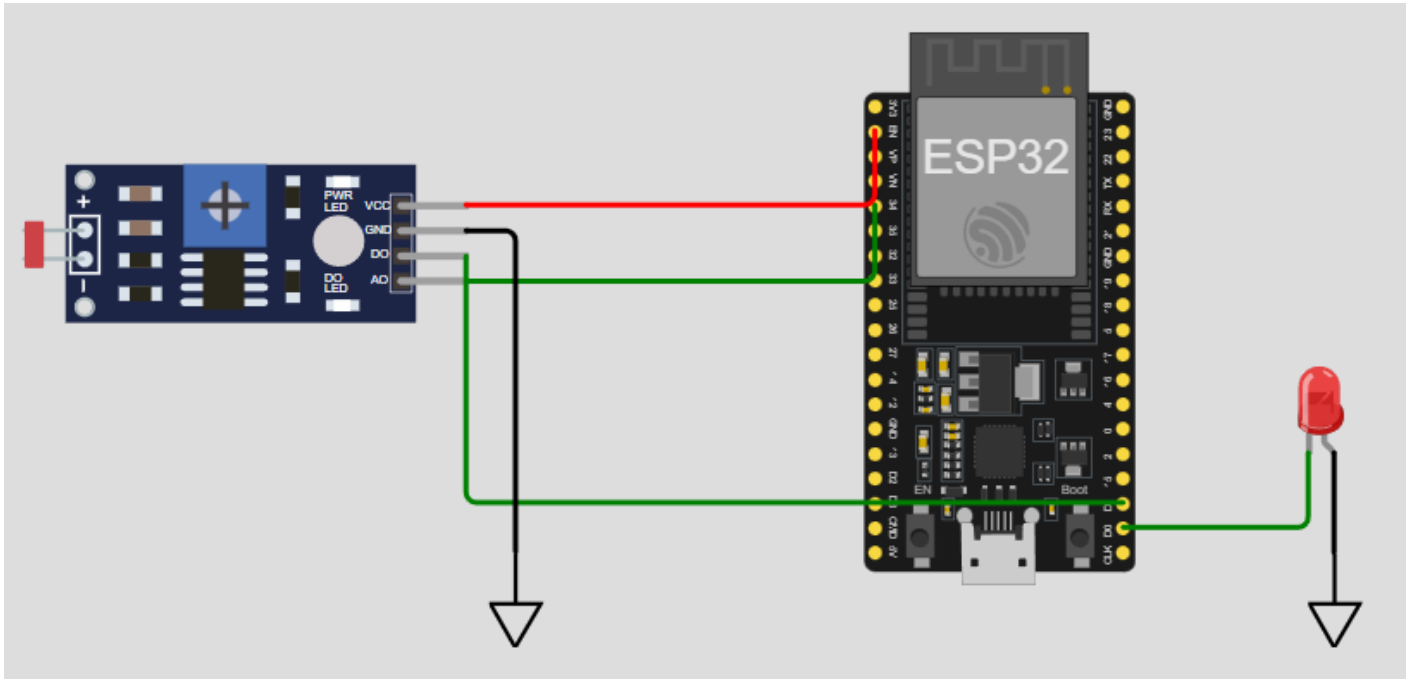
ESQUEMA DE CONEXIONES

- **SEMAFORO**
 - LED ROJO al D1 del ESP32
 - GND a GND
- **MODULO SENSOR LDR**
 - SEÑAL DIGITAL al D0 del ESP32
 - SEÑAL ANALÓGICA al A0 del ESP32
 - VCC a 3V3
 - GND A GND
- **MODULO SENSOR CO2**
 - SCL al pin A5 del ESP32
 - SDA al pin A4 del ESP32
 - PIN WAKE a GND
 - VCC a 3V3
 - GND A GND



i ESP32 pin numbers

i Nano pin numbers



DEVICES

Nos vamos a Arduino Cloud, y en **DEVICES** añadimos el ESP32 y obtenemos el TOKEN o palabra secreta (si has hecho la práctica anterior, no es necesario pues ya tenemos el TOKEN o palabra secreta) como es similar al caso anterior, no lo desarrollamos. (Nos pedirá también el SSID y la contraseña de la red wifi)

VARIABLES

Añadimos las siguientes **variables** :

- **CO2 tipo int y Read**
- **luz tipo int y Read**
- **luzdigital tipo bool y Read**
- **rojo tipo bool y Read&Write**

EL SKETCH -LIBRERIA CCS811

Primero añadiríamos la librería de keystudio <https://fs.keyestudio.com/KS0457> pero no lo permite Arduino Cloud, viendo las instrucciones, vemos que son las mismas que en los ejemplos de esta librería la de DF que es la que instalamos :

Things > DETECTOR CO2 ▾

Libraries
✕

Show libraries for:
DETECTORCO2 - ARDUINO NANO ESP32

All Libraries Custom Favorites

DFRobot_CCS811
3

Fary DFRobot <feng.yang@dfrobot.com>

DFRobot_CCS811 is the Sensor library of DFRobot

[More info](#)

Latest (1.0.1) ▾
INCLUDE

> Examples

SparkFun CCS811

SparkFun Arduino Library

SparkFun Electronics <sparkfun.com>

An Arduino library to drive the AMS CCS811 by I2C.

[More info](#)

```

71     lu:
72     if
73     (
74     )e:
75     (
76     }
77     lu:
78
79
80   }
81
82
83
84
85
86   /*
87     Since
88     execut
89   */
90   void on
91     // Ac
92   }
93

```

Console

esto provoca la incorporación de la línea 1 `#include <DFRobot_CCS811.h>`

EL SKETCH -EL CÓDIGO

- Tenemos las variables definidas en las líneas 10-13 :
 - **int cO2;**
 - **int luz;**
 - **bool luzdigital;**
 - **bool rojo;**

- Definimos una variable de tipo el sensor CCS811 en la línea 23 **DFRobot_CCS811 CCS811;**
- En Setup en las líneas 48-21 arrancamos ese sensor:
 - **while(CCS811.begin() != 0){**
 Serial.println("failed to init chip, please check if the chip connection is fine");
 delay(1000);
 }
- Definimos los pines digitales 0 y 1 como entrada y salida respectivamente:
 - **pinMode(1,OUTPUT);**
 pinMode(0,INPUT);
- En las líneas 60-70 que lea el CCS811 y la parte de CO2 que lo meta en la variable CO2 (línea 63)
 - **if(CCS811.checkDataReady() == true){**
 Serial.print("CO2: ");
 Serial.print(CCS811.getCO2PPM());
 cO2=CCS811.getCO2PPM();
 Serial.print("ppm, TVOC: ");
 Serial.print(CCS811.getTVOCPPB());
 Serial.println("ppb");
 } **else {**
 Serial.println("Data is not ready!");
 }
- En las línea 71 que luz sea la lectura del pin A0 **luz = analogRead(A0);**
- En las líneas 72-76 que según rojo se encienda o no el led
 - **if (rojo){**
 digitalWrite(1,HIGH);
 }else{
 digitalWrite(1,LOW);
 }
- En la línea 77 que luzdigital sea la lectura de la salida digital del sensor LDR
 - **luzdigital=digitalRead(0);**

```
#include <DFRobot_CCS811.h>
/*
Sketch generated by the Arduino IoT Cloud Thing "Untitled"
https://create.arduino.cc/cloud/things/17c10209-3874-430a-877c-c082ff7dd38d

Arduino IoT Cloud Variables description

The following variables are automatically generated and updated when changes are made to the
Thing

int c02;
int luz;
bool luzdigital;
bool rojo;

Variables which are marked as READ/WRITE in the Cloud Thing will also have functions
which are called when their values are changed from the Dashboard.

These functions are generated with the Thing and added at the end of this sketch.
*/

#include "thingProperties.h"

//DFRobot_CCS811 CCS811(&Wire, /*IIC_ADDRESS=*/0x5A);
DFRobot_CCS811 CCS811;

void setup() {
  // Initialize serial and wait for port to open:
  Serial.begin(9600);
  // This delay gives the chance to wait for a Serial Monitor without blocking if none is
found
  delay(1500);

  // Defined in thingProperties.h
  initProperties();

  // Connect to Arduino IoT Cloud
```

```
ArduinoCloud.begin(ArduinoIoTPreferredConnection);

/*
  The following function allows you to obtain more information
  related to the state of network and IoT Cloud connection and errors
  the higher number the more granular information you'll get.
  The default is 0 (only errors).
  Maximum is 4
*/
setDebugMessageLevel(2);
ArduinoCloud.printDebugInfo();

while(CCS811.begin() != 0){
  Serial.println("failed to init chip, please check if the chip connection is fine");
  delay(1000);
}
pinMode(1,OUTPUT);
pinMode(0,INPUT);
}

void loop() {
  ArduinoCloud.update();
  // Your code here
  if(CCS811.checkDataReady() == true){
    Serial.print("CO2: ");
    Serial.print(CCS811.getCO2PPM());
    c02=CCS811.getCO2PPM();
    Serial.print("ppm, TVOC: ");
    Serial.print(CCS811.getTVOCPPB());
    Serial.println("ppb");

  } else {
    Serial.println("Data is not ready!");
  }
}
```

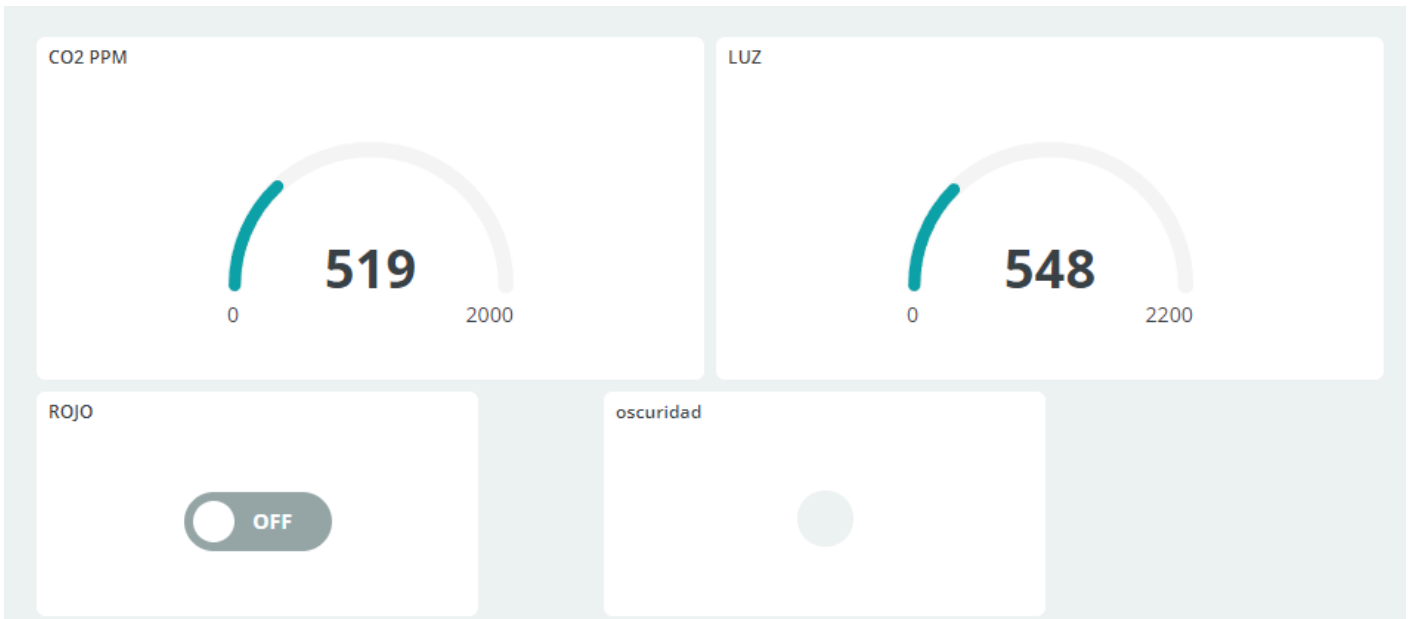
```
    luz = analogRead(A0);
    if (rojo){
        digitalWrite(1,HIGH);
    }else{
        digitalWrite(1,LOW);
    }
    luzdigital=digitalRead(0);

}

/*
  Since Rojo is READ_WRITE variable, onRojoChange() is
  executed every time a new value is received from IoT Cloud.
*/
void onRojoChange() {
    // Add your code here to act upon Rojo change
}
```

DASHBOARD

- Un gauge ligado a **CO2** desde 0 a 2000
- Un gauge ligado a **Luz** de 0 a 2.200
- Un Switch ligado a **rojo**
- Un led de oscuridad ligado a **luzdigital**



Alternativa : en vez de luz tendría que llamarse "oscuridad" que sea luz pero que vaya al revés

RESULTADO

<https://www.youtube.com/embed/WS02iGCux4U>

ALTERNATIVA: Que el semáforo visualice los niveles peligrosos de CO2, por ejemplo el umbral del amarillo 600-1.000

¿Te atreves a poner un servomotor?

Predictor meteorológico

Predictor meteorológico

Autor Mario Monteagudo Asesor digital Centro de Profesorado de Ejea de los Caballeros

Usando IDE Arduino y el sensor Bosch BME280 se puede medir la presión atmosférica, temperatura y humedad y usando un código predictivo **Algoritmo de Zambretti** que usa las diferencias de presiones cada 3 horas, puede hacer pronósticos. Para ello mide la presión cada 5 minutos y tienen en cuenta la altitud del lugar y la temperatura.

El ESP32 tendría que estar en el exterior y como hemos explicado, necesita 3 horas para hacer su primera predicción.

Este es el resultado

<https://www.youtube.com/embed/psGhrktsdQI?t=1s>

Y este es el código

```
#include "arduino_secrets.h"
//Predictor del tiempo mediante el algoritmo de Zambretti*****
//2024*Mario Monteagudo Alda*****mario.monteagudo@cpejea.es*****

#include "arduino_secrets.h"

#include <SimpleBME280.h> //Biblioteca usada para el sensor

const float ALTURA = 497.0; //Altitud del lugar

SimpleBME280 bme280; //Declaración de la instancia del sensor

float presiones[37]; //Presiones corregidas cincominutales de las 3 últimas horas
```

```

int zambretti;          //Número de Zambretti
float temperatura;     //°C
float humedad;         //Humedad relativa en %
float presion;         //Presión atmosférica en mbar
int contador = 0;      //Contador de número de medidas de presión cincominutales tomadas
bool encendido = false; //Bandera para intermitencia del LED

/*
  Arduino IoT Cloud Variables description

  The following variables are automatically generated and updated when changes are made to the
  Thing

  String boletin;
  float fiabilidad;
  float tendencia;

  Variables which are marked as READ/WRITE in the Cloud Thing will also have functions
  which are called when their values are changed from the Dashboard.

  These functions are generated with the Thing and added at the end of this sketch.
*/

#include "thingProperties.h"

void setup() {

  pinMode(LED_BUILTIN,OUTPUT);
  digitalWrite(LED_BUILTIN,HIGH);

  // Inicialización de la comunicación serie y espera
  Serial.begin(19200);
  delay(2000);

  // Defined in thingProperties.h
  initProperties();

```

```
// Connect to Arduino IoT Cloud and wait
ArduinoCloud.begin(ArduinoIoTPreferredConnection, false);
delay(2000);
/*
    The following function allows you to obtain more information
    related to the state of network and IoT Cloud connection and errors
    the higher number the more granular information you'll get.
    The default is 0 (only errors).
    Maximum is 4
*/
setDebugMessageLevel(2);
ArduinoCloud.printDebugInfo();

bme280.begin(); //Inicialización del sensor y espera
delay(2000);

//Primera lectura del sensor
bme280.update();
presion = bme280.getP();
temperatura = bme280.getT();

//Corrección de la presión según la altitud y temperatura y paso de Pa a mbar
presion = (presion * pow(1 - (0.0065 * ALTURA) / (temperatura + (0.0065 * ALTURA) +
273.15), -5.257 ))/100.0;
for (int i=0; i<=36; i++) { presiones[i]=presion;}

fiabilidad = 0;
}

void loop() {

    bme280.update();
    presion = bme280.getP();
    temperatura = bme280.getT();
    humedad = bme280.getH();
    presion = (presion * pow(1 - (0.0065 * ALTURA) / (temperatura + (0.0065 * ALTURA) +
```

```

273.15), -5.257 ))/100.0;

tendencia=presion-presiones[0];

for (int i=0; i<36; i++) { presiones[i]=presiones[i+1];}
presiones[36] = presion;

//Cálculo del número de Zambretti
if ((tendencia <= -1.6) & (presion > 985.0) & (presion < 1050.0)) {zambretti = round(127-
0.12*presion);}

else if ((tendencia >= 1.6) & (presion > 947.0) & (presion < 1030.0)) {zambretti =
round(185-0.16*presion);}

else if ((abs(tendencia) < 1.6) & (presion > 960.0) & (presion < 1033.0)) {zambretti =
round(144-0.13*presion);}

else {zambretti = 0;}

//Boletín con el pronóstico
boletin = "Temperatura: " + String(temperatura,1) + " °C" + "\n" +
"Humedad relativa: " + String(humedad,1) + " %" + "\n" +
"Presión: " + String(presion,0) + " mbar" + "\n" +
"Tendencia: " + String(tendencia,2) + " mbar" + "\n" +
pronostico(zambretti);

Serial.println(boletin + "\n");

//Espera de cinco minutos para la actualización
unsigned long tiempo = millis();
while ((millis()-tiempo) < 5*60*1000 )
{
  ArduinoCloud.update();
  delay(1000);
  encendido = !encendido;
  digitalWrite(LED_BUILTIN,encendido);
}

```

```
}

contador = min(36, contador + 1);

//Después de tres horas de medidas, la fiabilidad es del 100%
fiabilidad = map(contador, 0, 36, 0, 100);

}

//Cadena de texto con el pronóstico meteorológico*****

String pronostico(int z)
{
  switch (z)
  {
    case 1:
      return "Tiempo estable"; break;
    case 2:
      return "Buen tiempo"; break;
    case 3:
      return "Buen tiempo con ligera inestabilidad"; break;
    case 4:
      return "Buen tiempo evolucionando a chubascos"; break;
    case 5:
      return "Chubascos evolucionando a tiempo inestable"; break;
    case 6:
      return "Inestable evolucionando a lluvioso"; break;
    case 7:
      return "Intervalos de lluvia con empeoramiento"; break;
    case 8:
      return "Intervalos de lluvia evolucionando a gran inestabilidad"; break;
    case 9:
      return "Muy inestable con lluvia"; break;
    case 10 :
      return "Tiempo estable"; break;
    case 11:
```

```
    return "Buen tiempo"; break;
case 12:
    return "Buen tiempo con posibles chubascos"; break;
case 13:
    return "Buen tiempo con probables chubascos"; break;
case 14:
    return "Chubascos con intervalos despejados"; break;
case 15:
    return "Variable con algo de lluvia"; break;
case 16:
    return "Inestable con intervalos de lluvia"; break;
case 17:
    return "Lluvia a intervalos frecuentes"; break;
case 18:
    return "Muy inestable con lluvia"; break;
case 19:
    return "Tormentoso con lluvia abundante"; break;
case 20 :
    return "Tiempo estable"; break;
case 21:
    return "Buen tiempo"; break;
case 22:
    return "Evolucionando a buen tiempo"; break;
case 23:
    return "Buen tiempo evolucionando a mejor"; break;
case 24:
    return "Buen tiempo con posibles chubascos a primeras horas"; break;
case 25:
    return "Chubascos a primeras horas evolucionando a mejor"; break;
case 26:
    return "Variable evolucionando a mejor"; break;
case 27:
    return "Inestable a primeras horas evolucionando a estable"; break;
case 28:
    return "Inestable con probable mejoría"; break;
case 29:
```

```
    return "Inestable con breves intervalos de buen tiempo"; break;
case 30:
    return "Muy inestable con intervalos de buen tiempo"; break;
case 31:
    return "Tormentoso con probable mejoría"; break;
case 32:
    return "Tormentoso con mucha lluvia"; break;
default:
    return "Tiempo indeterminado";
}
}
```