

Sistema de riego automatico



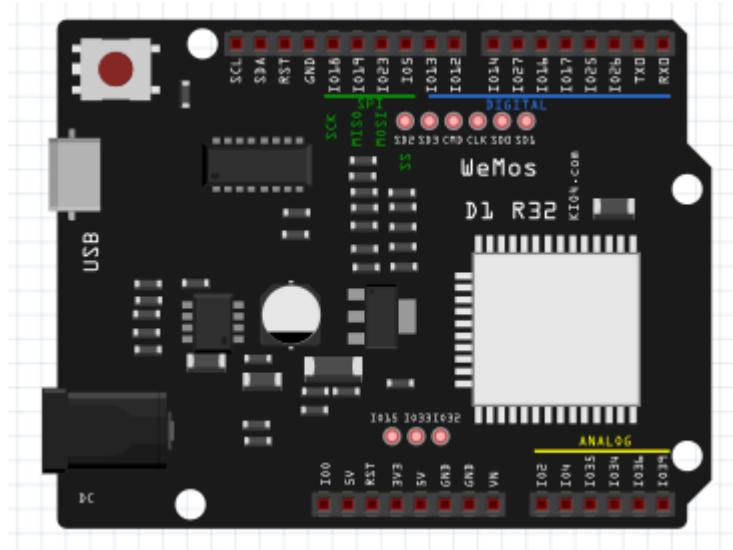
by @javacasm - José Antonio Vacas

<http://bit.ly/RiegoCruzRoja>

Vamos a realizar un sistema de riego automático (en función de la humedad del suelo) y que también podremos activar remotamente vía wifi

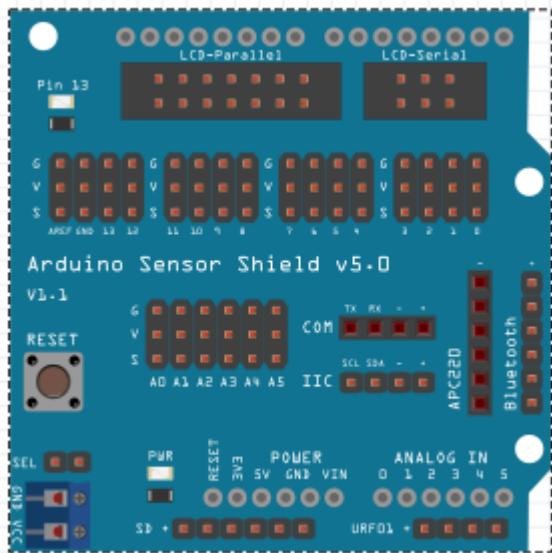
Hardware

Usaremos la placa ESP32 Wemos D1 R32, compatible con Arduino y que nos permite conexiones wifi y bluetooth



Este placa tiene el mismo formato que el conocido Arduino UNO y es compatible con esta placa a nivel de conexión.

Para facilitar la conexión de los sensores y actuadores usaremos una placa "Sensor Shield v5"



Que nos proporciona conexión para V, GND, Señal para cada sensor. Por ello usaremos sensores en formato módulo que nos facilita la conexión, puesto que los módulos tiene las mismas 3 conexiones: V, GND y S

Programación con bloques

Para programar usaremos una herramienta web de programación con bloques Arduinoblocks

Comenzaremos usando [ArduinoBlocks.com](https://www.ArduinoBlocks.com): una herramienta de programación con bloques ideal para iniciarse.

Instalamos [ArduinoBlock Connector](#) que nos permite sincronizar el programa en el estamos trabajando con la placa

ArduinoBlocks-Connector v5.1

Arduino Blocks
CONNECTOR 

ArduinoBlocks.com

```
bles. Maximum is 327680 bytes.

20:20:19> [ESP32 / ESP32-WROOM] [/dev/ttyUSB0] Compiling and uploading...
20:20:37> Sketch uses 211986 bytes (16%) of program storage space. Maximum is 1310720 bytes.
Global variables use 13824 bytes (4%) of dynamic memory, leaving 313856 bytes for local variables. Maximum is 327680 bytes.

20:21:03> !!! [Errno 9] Bad file descriptor
```

[Creamos nuestra cuenta](#) donde se guardarán nuestros proyectos:

The screenshot shows the registration form for ArduinoBlocks. The form fields are as follows:

- Correo electrónico
- Confirmación de correo electrónico
- Clave
- Confirmación de clave
- Nombre
- Apellidos
- País: SPAIN
- Ciudad

Below the form is a checkbox labeled "Recibir información y novedades por email" which is checked. At the bottom left is a "Captcha" field containing the text "vicedi". A large blue button at the bottom right says "Nuevo usuario".

Creando proyectos

Creamos proyecto personal con la placa ESP32 STEAMakers, que no es exactamente la nuestra pero sí es compatible

arduino blocks Buscar proyectos Proyectos ▾

Nuevo proyecto personal

Tipo de proyecto	
Nombre	UNO NANO / ATmega328 NANO / ATmega328 (new bootloader) MEGA / 2560 Leonardo UNO + Imagina TdRSTEAM 3dBot / Imagina Keyestudio EasyPlug Keyestudio KeyBot ESP32 STEAMakers
Descripción	ESP32 STEAMakers + Imagina TdR STEAM ESP32 STEAMakers + 3dBot ESP32 / WROOM ESP8266 / NodeMCU v2 ESP8266 / WeMos D1 Otto DIY / Nano Otto DIY / Nano (new bootloader)
Componentes	

Documentaremos la descripción del proyecto y los componentes que vamos a usar en cada montaje.

Encendiendo el LED de la placa

Usaremos el led que incluye la placa Wemos D1 que está conectado al pin 2

No necesitamos conectar nada

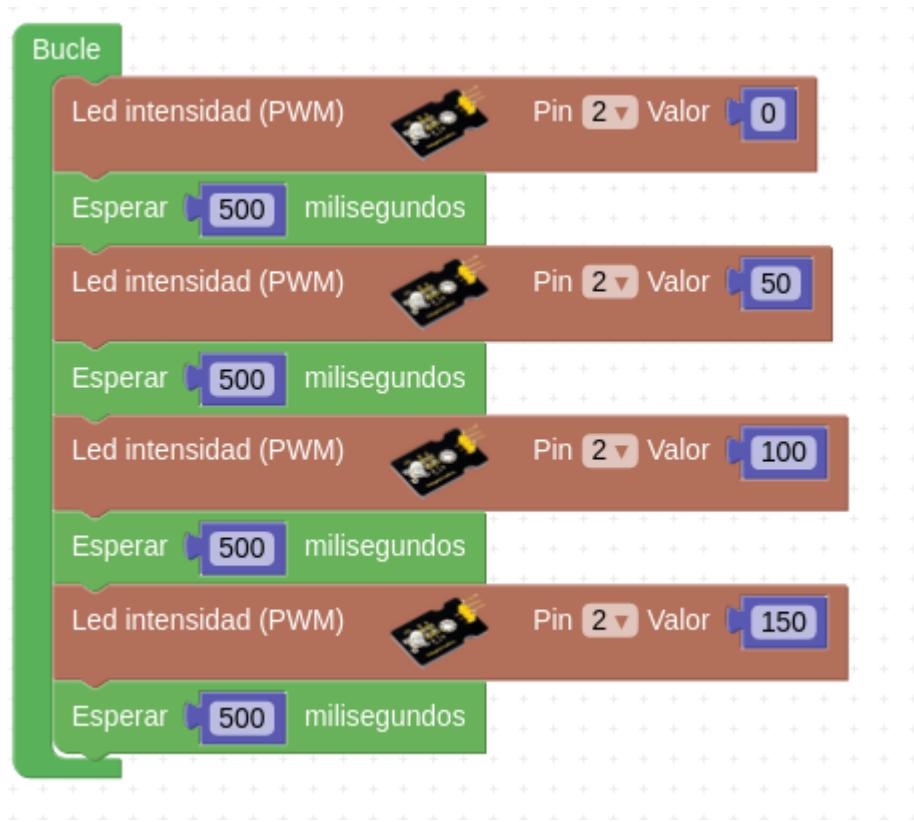


[Led 2 Steamakers](#)

Controlando el brillo de un led

Vamos a controlar ahora el brillo de un led analógicamente

led 2 PWM



Necesitamos:

- Wemos R32 D1
- Cable USB

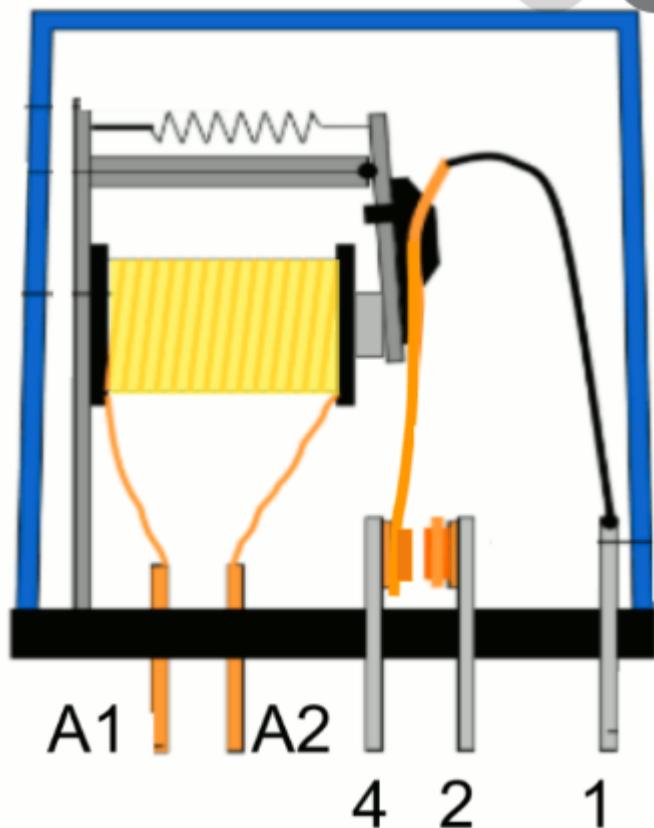
Veremos como el led va cambiando su brillo en 4 niveles.

Relé

Ahora vamos a conectar un relé para poder controlar dispositivos de más potencia

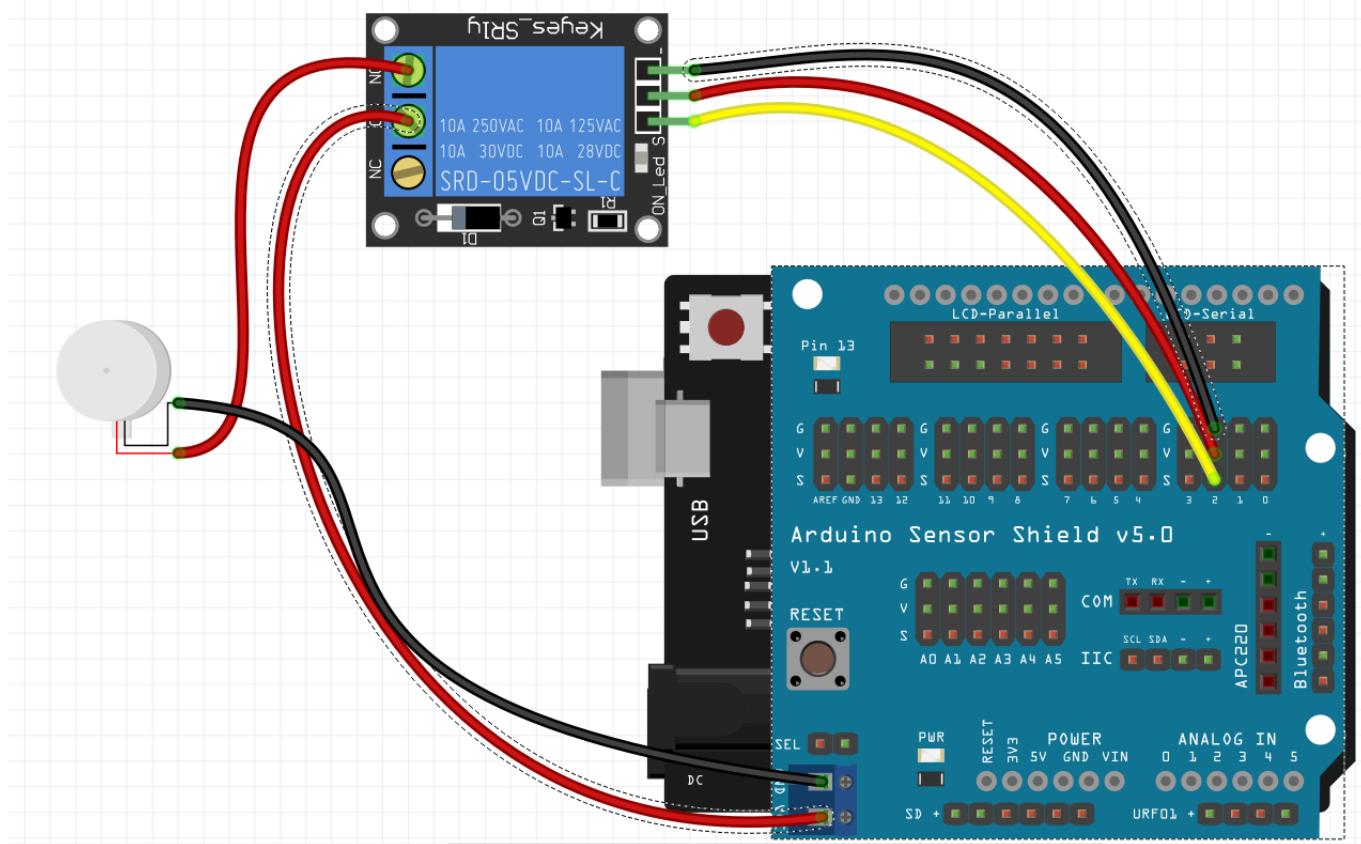
1. ¿Qué es un **relé**? un relé es un interruptor eléctrico que podemos accionar electrónicamente con una pequeña corriente y que funciona casi siempre electromecánicamente (un electroimán que atrae un

contacto eléctrico) de ahí el 'clic-clic' que hacen al funcionar.



2. Uso: los usaremos para controlar dispositivos que necesitan más potencia que nuestra placa y/o que funcionan a mayores voltajes.
3. Control: para activar/desactivar los relés sólo tenemos que activar/desactivar el pin de nuestra placa al que está conectado
4. Alimentación y Consumo: el control de varios relés requiere de una mayor potencia de la que puede suministrar el puerto USB, por lo que usaremos un alimentación más potente si es posible.
5. Puesto que las señales de control de las placas son de 3.3V pudiera ser que algunos relés no se activen correctamente, aunque la mayoría sí. Lo que es importante es que alimentemos el relé (patilla V o Vcc) con 5V.

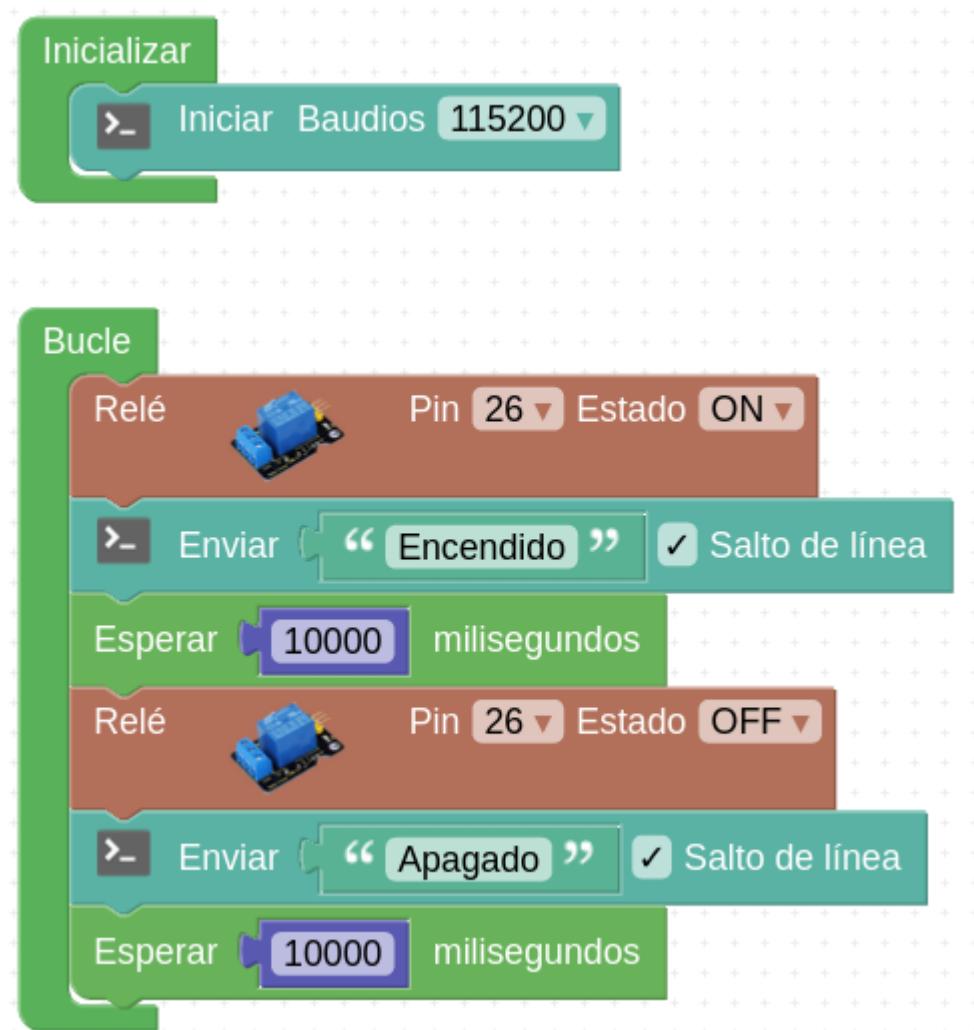
Conectamos el relé al pin 26



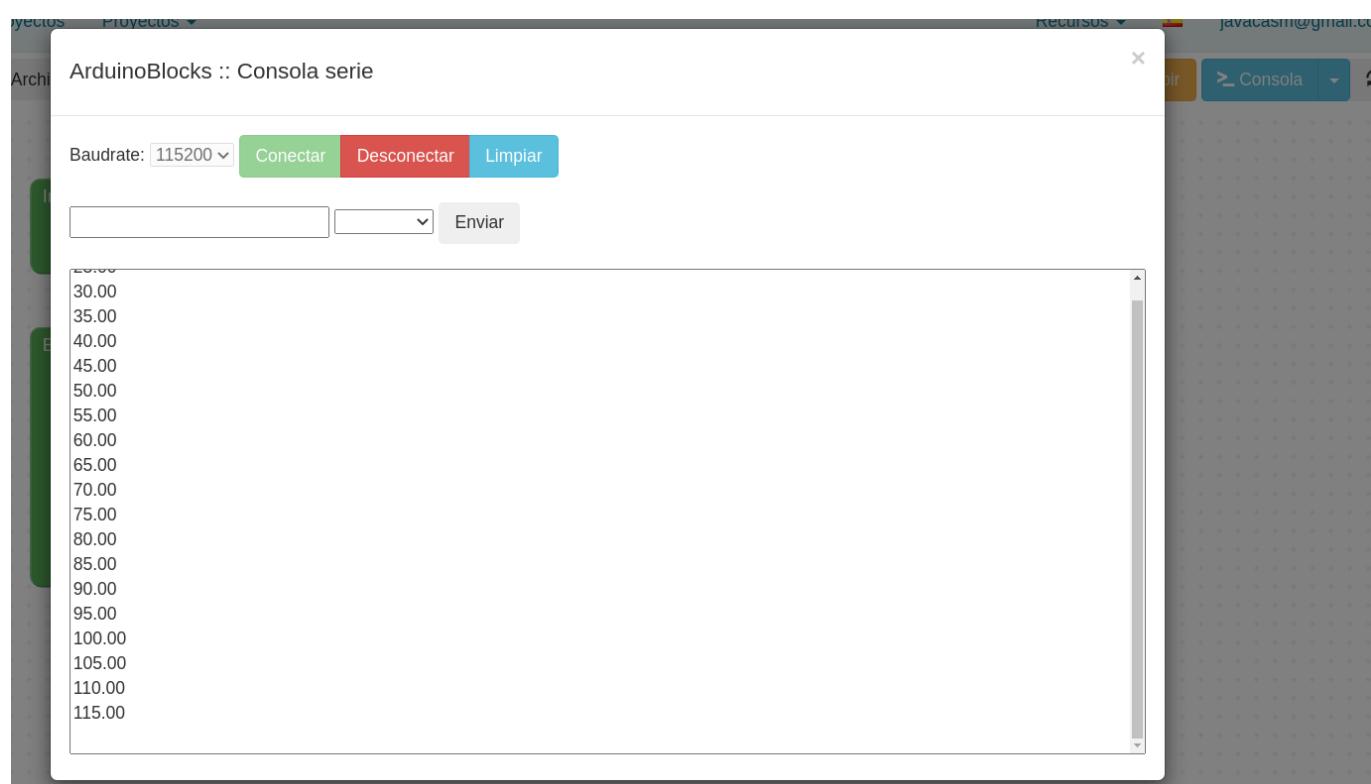
Necesitamos:

- Wemos R32 D1
- Sensor Shield
- Cable USB
- Relé

El [programa](#) será el siguiente



Activamos la consola, seleccionamos la velocidad adecuada y pulsamos "Conectar"

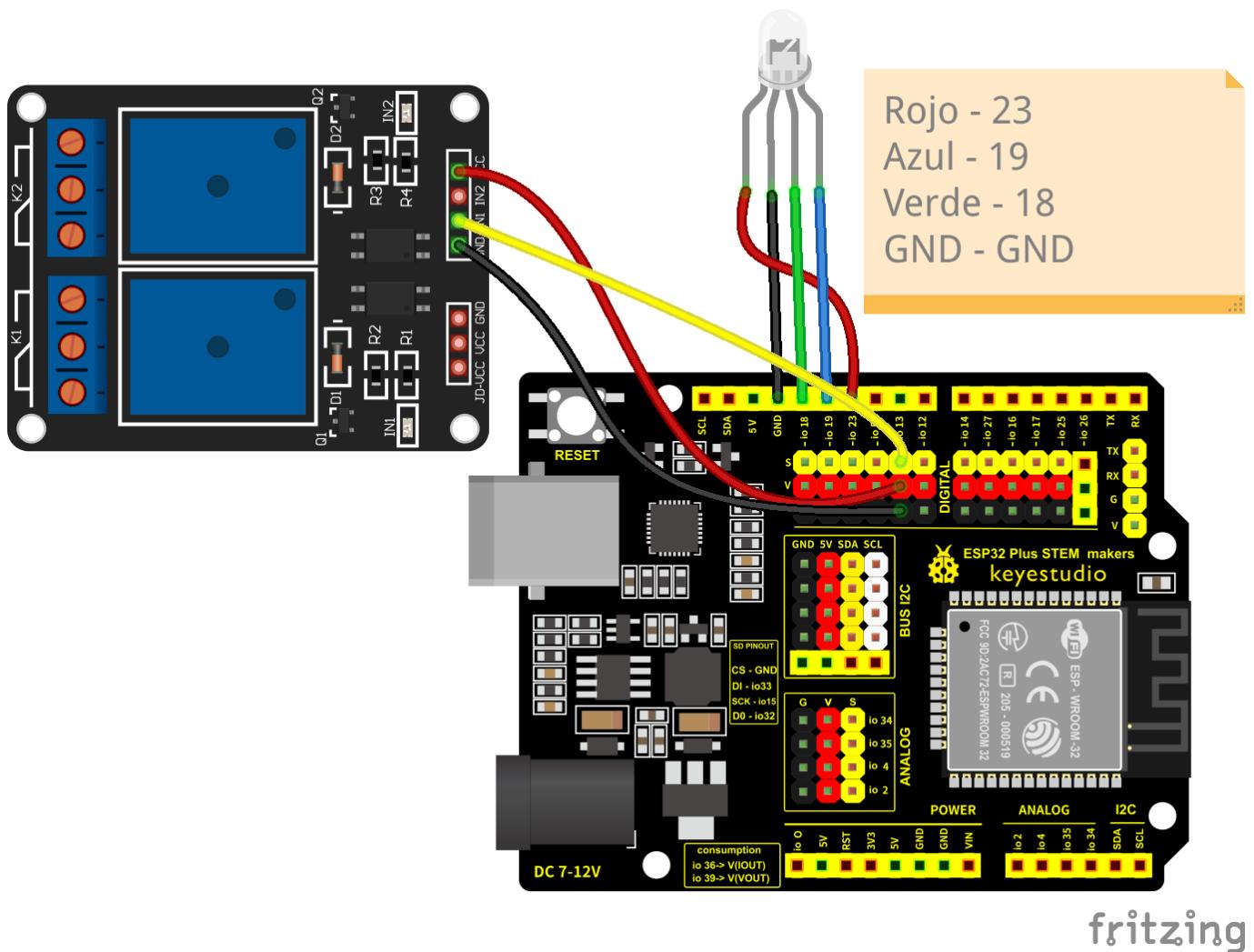


Controlando relés

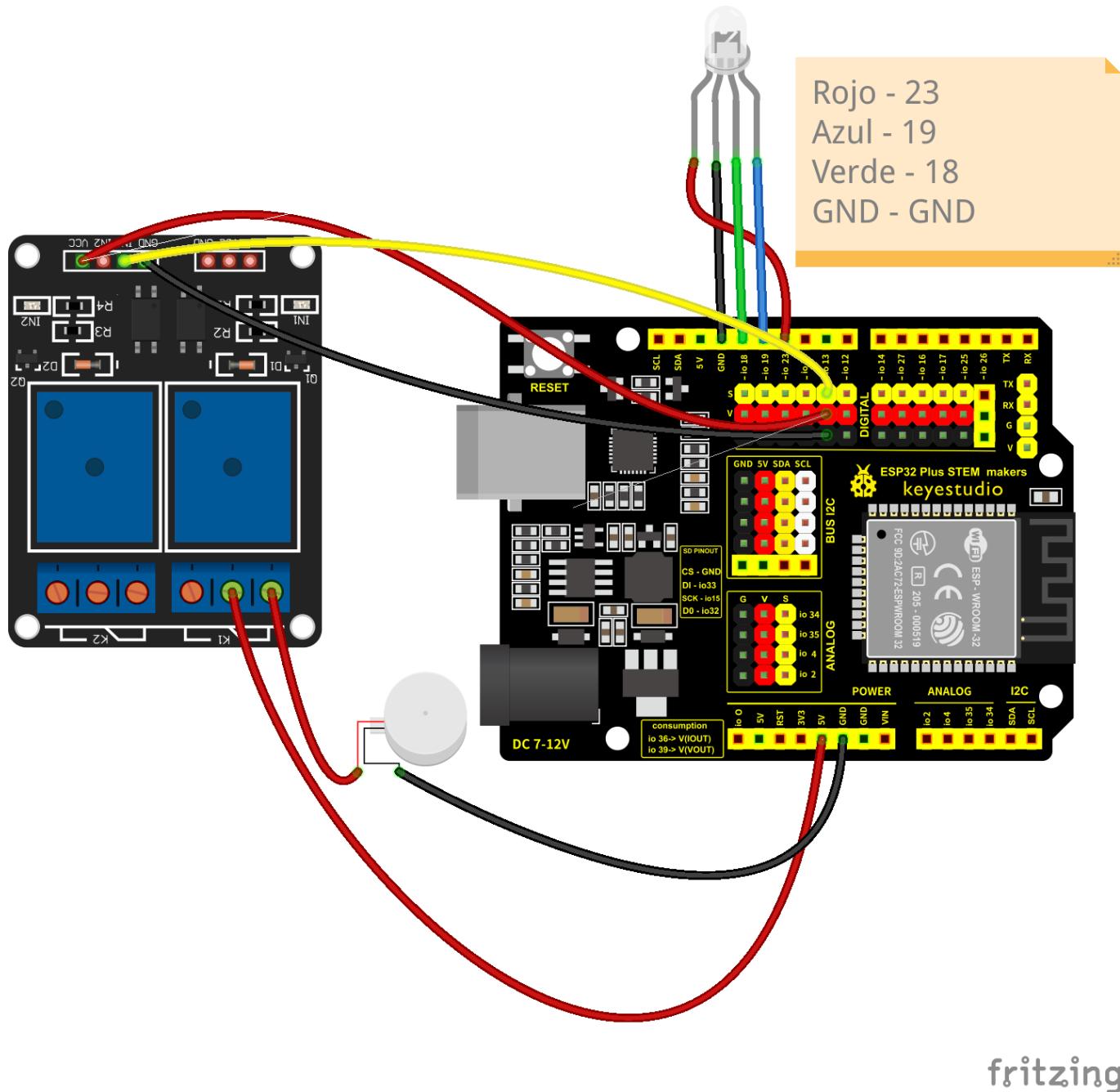
Una aplicación directa de lo que hemos visto activando leds, puede ser controlar un módulo con varios relés al mismo tiempo.



Como hemos dicho, un relé es un dispositivo que nos permite controlar un dispositivo conectado a la corriente con una salida de nuestro dispositivo y sus modestos 5 voltios. Es decir, el relé actúa como un interruptor electrónico y al activar la patilla que lo controla se cierra el contacto que deja pasar la corriente al dispositivo



En el montaje del ejemplo vemos como el relé 1 actúa como interruptor del circuito de alimentación de la bombilla. Al activar la salida 18, que controla el relé 1 (por estar conectada a IN1) cerrará la salida del relé y la alimentación llegará a la bombilla. Podemos poner la alimentación desde un enchufe o desde cualquier otra fuente de alimentación. Los relés funcionan como un interruptor eléctrico.



fritzing

Los dos circuitos eléctricos, de alta potencia (el motor) y el de baja potencia (la placa) están aislados entre sí. No obstante **hay que tener cuidado al manejar la parte de alto voltaje/potencia**

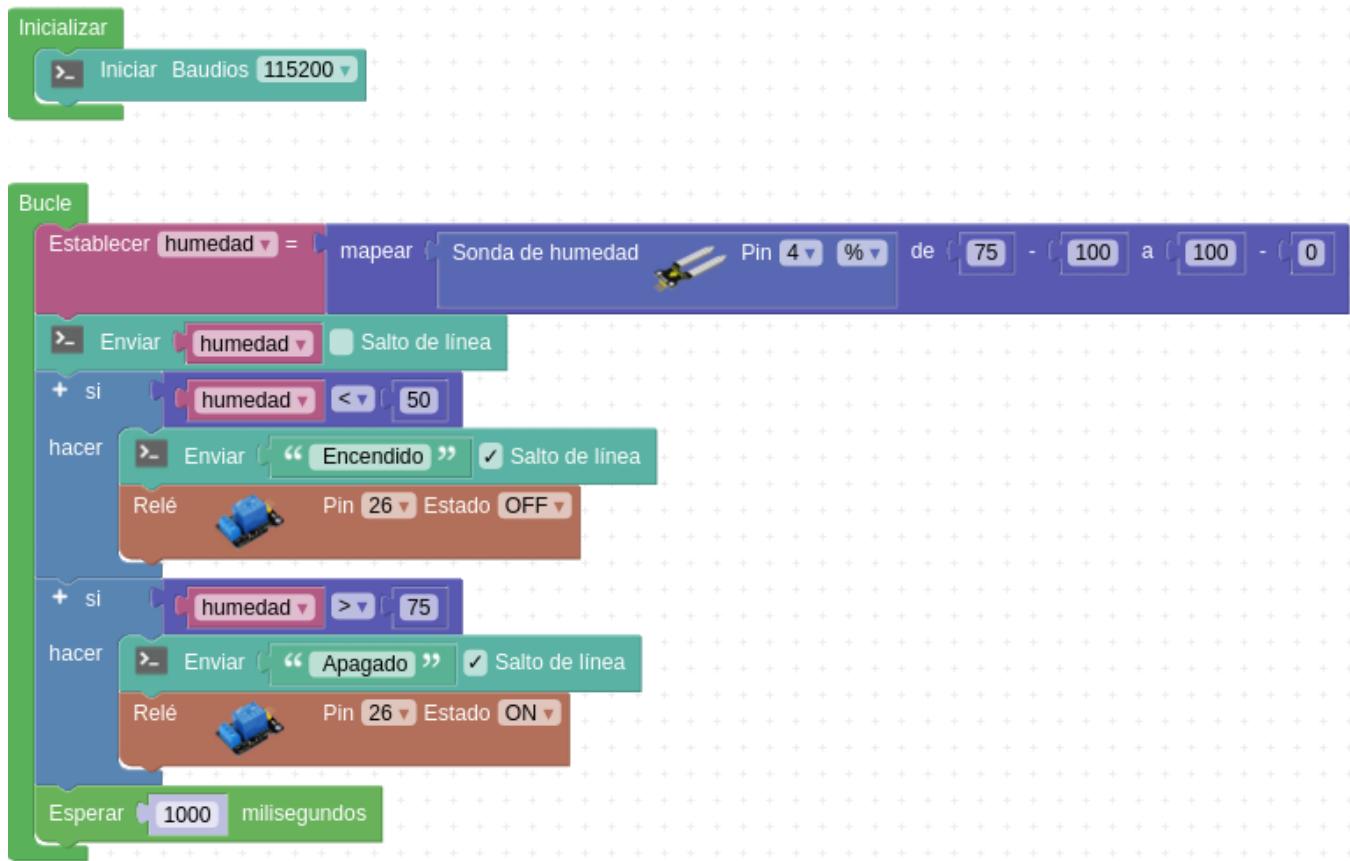
Sistema de riego

Podemos hacer distintas versiones:

- Tiempo: actúa como un temporizador, que se activará con cierta frecuencia. Tiene el inconveniente de que puede que reguemos sin necesidad, si ha llovido por ejemplo
- Humedad del suelo: medimos la humedad del suelo y por debajo o encima de un valor activamos o apagamos el riego
- Humedad y lluvia: detecta si llueve para en ese caso no activar el riego

Medida de humedad

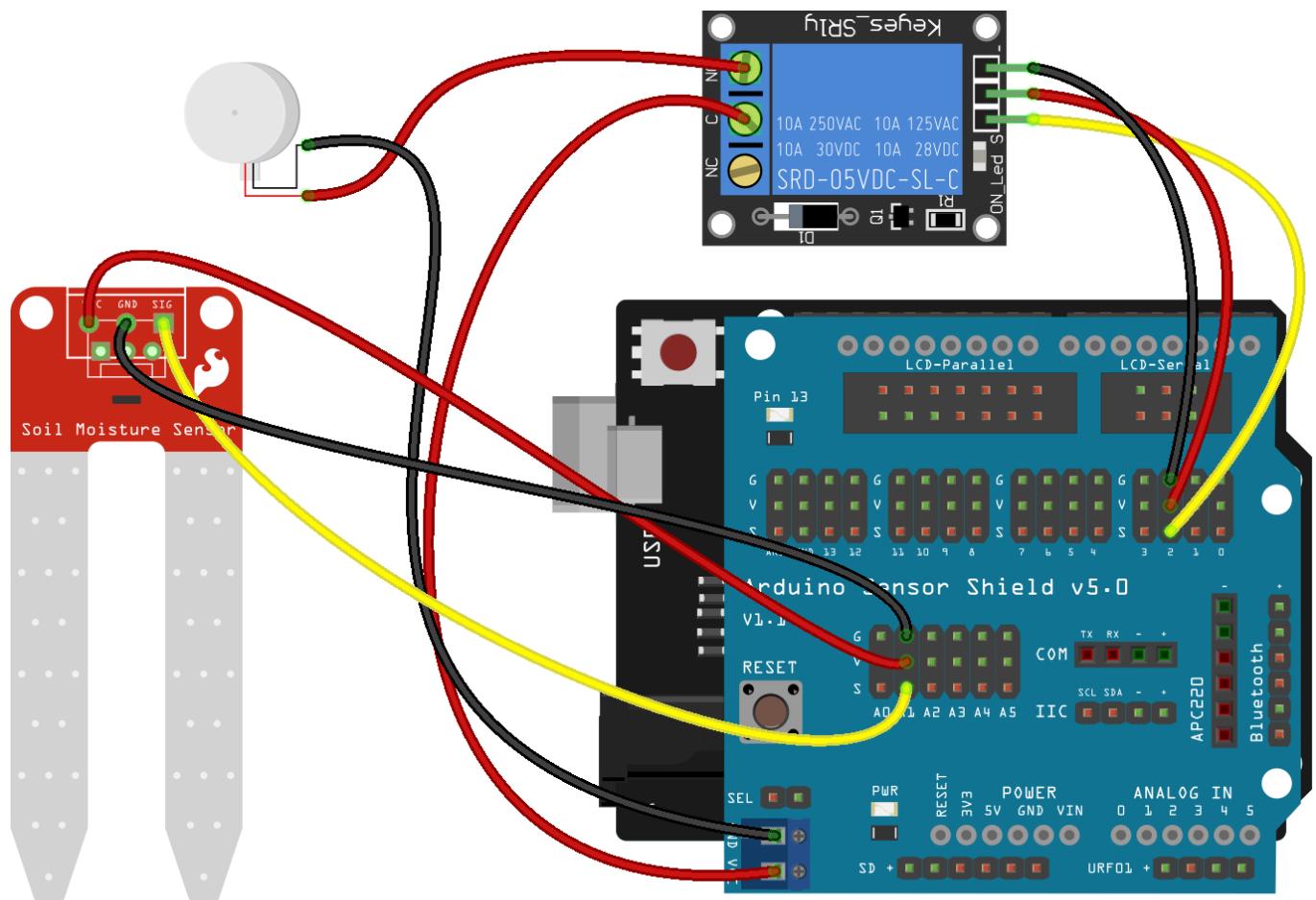
Medimos la humedad del suelo y por encima/debajo de un valor activamos/apagamos el riego.



Proyecto

Necesitamos:

- Wemos R1
- Cable USB
- Relé
- Bomba de riego
- Pila de 9v
- Sensor de humedad



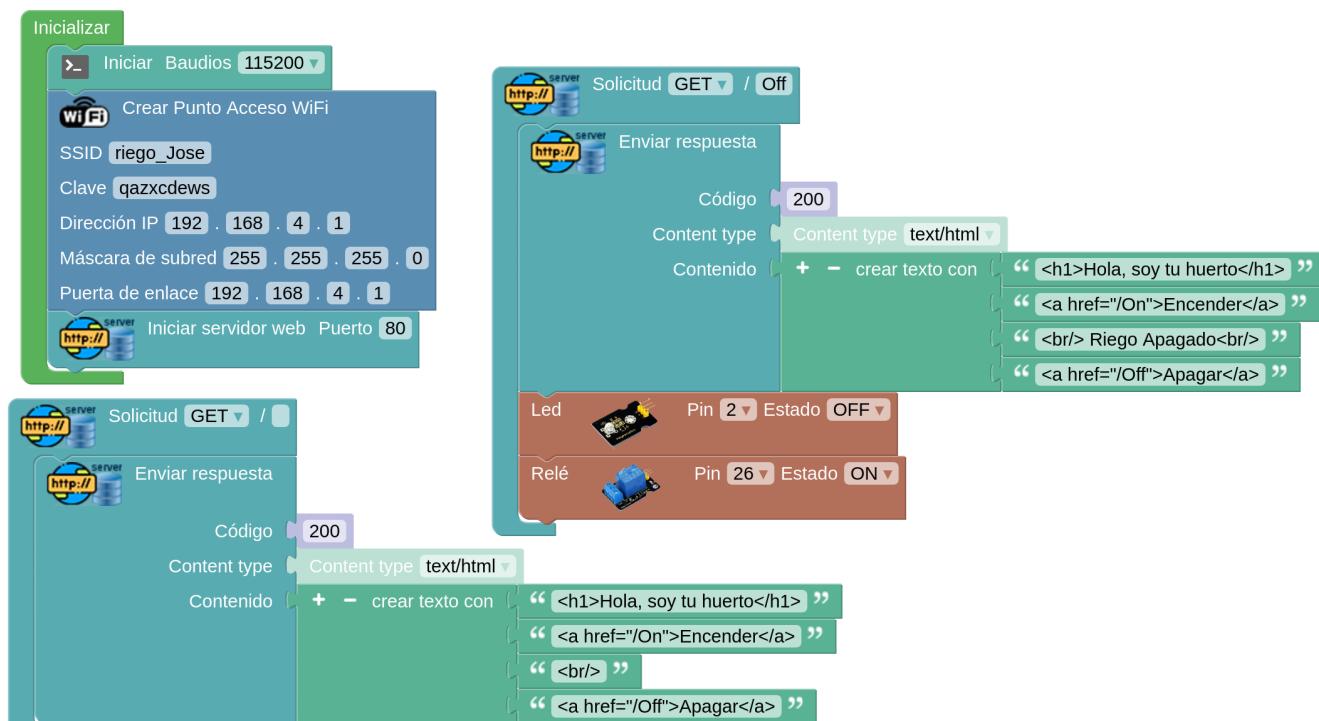
fritzing

Tenemos que determinar el valor de nuestro sensor que vamos a usar como umbral

Wifi

Creamos una red wifi a la que nos conectamos para poder controlar el riego

Proyecto wifi



Solicitud GET / On

```

    [Solicitud GET / On]
        [Enviar respuesta]
            [Código 200]
            [Content type text/html]
            [Contenido]
                + - crear texto con
                    " <h1>Hola, soy tu huerto</h1> "
                    " <a href="/On">Encender</a> "
                    " <br/>Riego Encendido<br/> "
                    " <a href="/Off">Apagar</a> "
        [Relé Pin 26 Estado ON]
        [Esperar 200 milisegundos]
        [Relé Pin 26 Estado OFF]
        [Led Pin 2 Estado OFF]

```

Bucle

```

    [Bucle]
        [Establecer humedad =]
            [mapear Sonda de humedad Pin 4 % de 60 a 100 a 100 a 0]
        [- Enviar humedad Salto de línea]
        [+ si humedad < 50]
            [hacer]
                [Relé Pin 26 Estado OFF]
                [Led Pin 2 Estado ON]
                [- Enviar "encendido" Salto de línea]
        [+ si humedad > 75]
            [hacer]
                [Relé Pin 26 Estado ON]
                [Led Pin 2 Estado OFF]
                [- Enviar "apagado" Salto de línea]
        [Esperar 100 milisegundos]

```

Iniciarizar

```

    [Crear Punto Acceso WiFi]
        [SSID riego_Jose]
        [Clave qazxcdews]
        [Dirección IP 192 . 168 . 4 . 1]
        [Máscara de subred 255 . 255 . 255 . 0]
        [Puerta de enlace 192 . 168 . 4 . 1]
    [Iniciar servidor web Puerto 80]

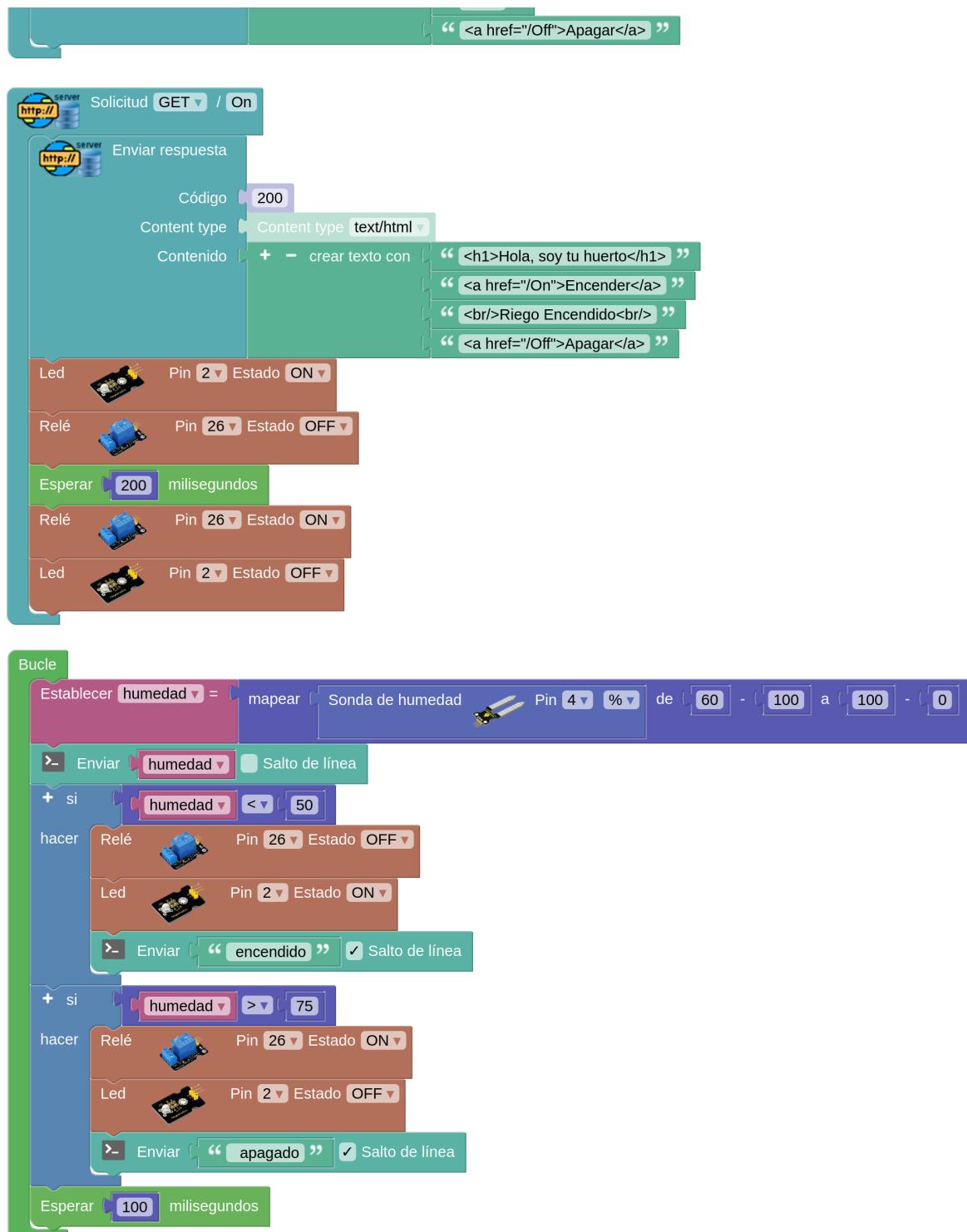
```

Solicitud GET / Off

```

    [Solicitud GET / Off]
        [Enviar respuesta]
            [Código 200]
            [Content type text/html]
            [Contenido]
                + - crear texto con
                    " <h1>Hola, soy tu huerto</h1> "
                    " <a href="/On">Encender</a> "
                    " <br/> Riego Apagado<br/> "
                    " <a href="/Off">Apagar</a> "
        [Relé Pin 26 Estado OFF]

```



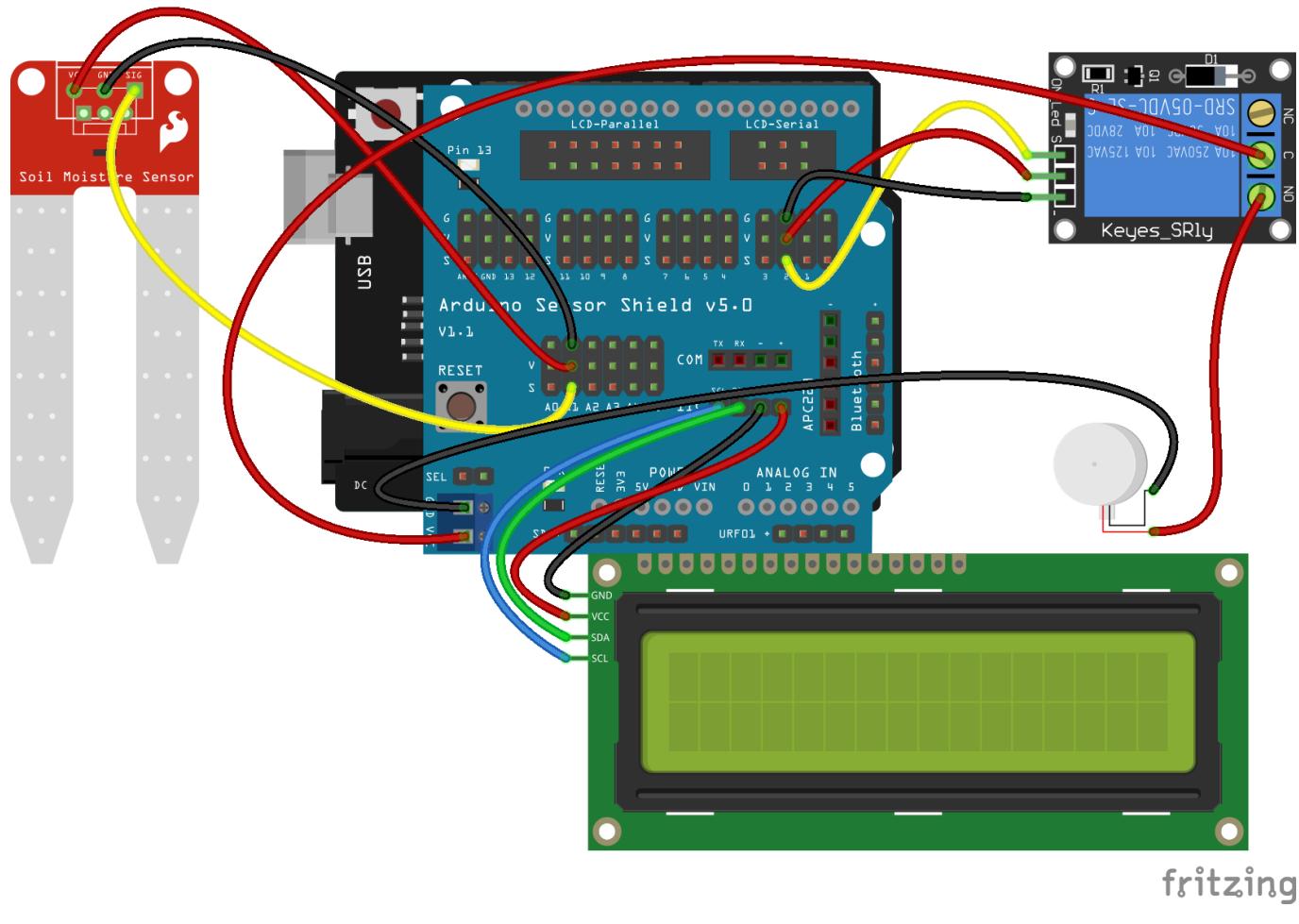
Para ver el resultado o para controlarlo tenemos que conectarnos al wifi y en un navegador abrir la dirección <http://192.168.4.1>

Desde ahí podemos activar o desactivar el riego

Pantalla LCD

Vamos a conectar una pantalla LCD (como las máquinas de vending)

La conexión es muy sencilla puesto que la LCD tiene las mismas 4 conexiones que la placa Sensor Shield



fritzing

- Wemos R32 D1
- relé - pin 26
- pila 9v
- cable usb
- bomba de riego - conectada al relé
- sensor de humedad - pin 4
- Pantalla LCD I2C

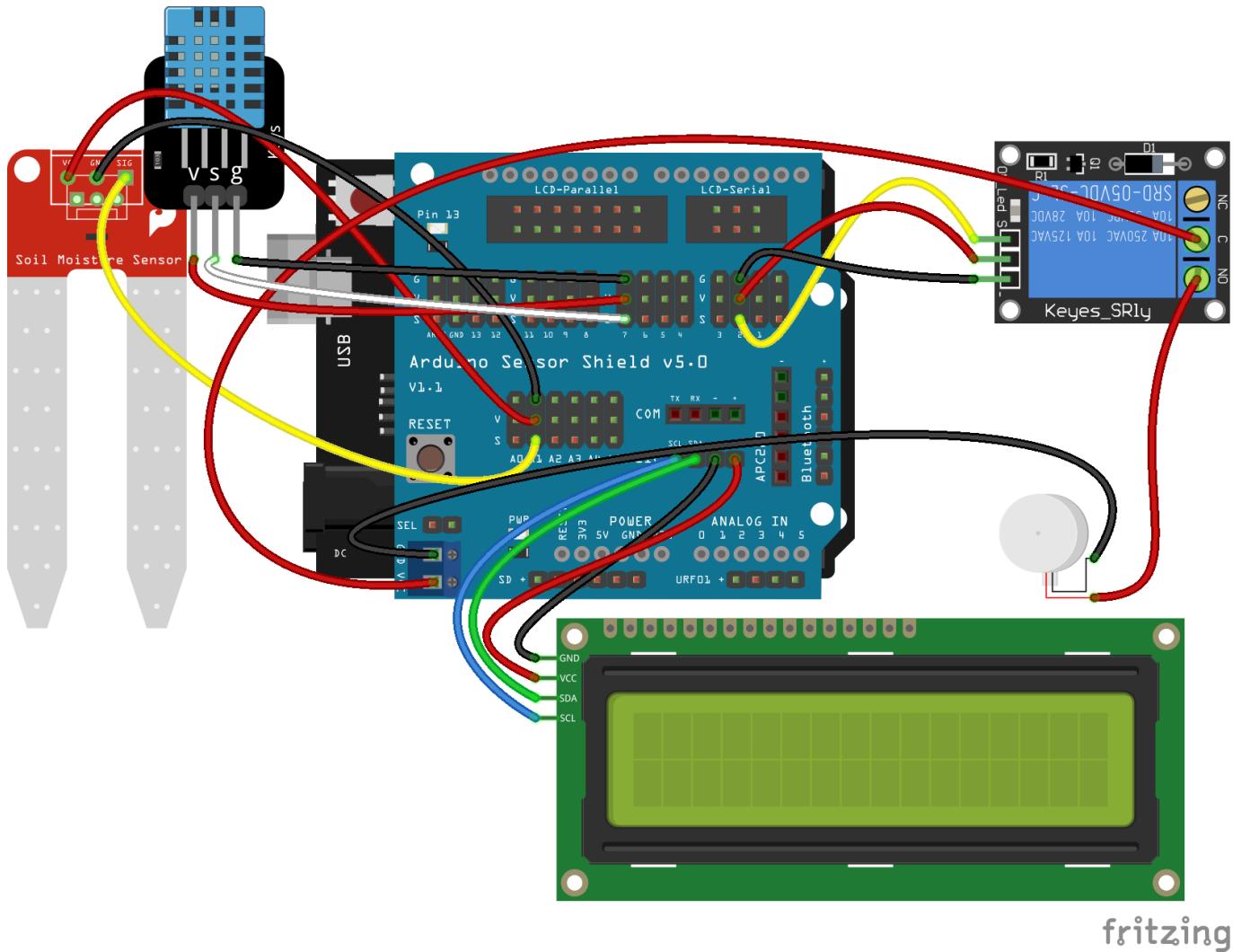
En el [programa](#) añadimos la configuración de la pantalla donde necesitamos saber el número de caracteres que tiene (2 filas y 16 columnas) y la dirección que es un número que nos da el fabricante (0x27 en la mayoría)

Mostraremos en la pantalla el valor de la humedad del suelo y el estado del riego



Riego + LCD + Sensor de humedad y temperatura DHT11

Vamos a añadir ahora un sensor de temperatura y humedad DHT11 conectado al pin 14

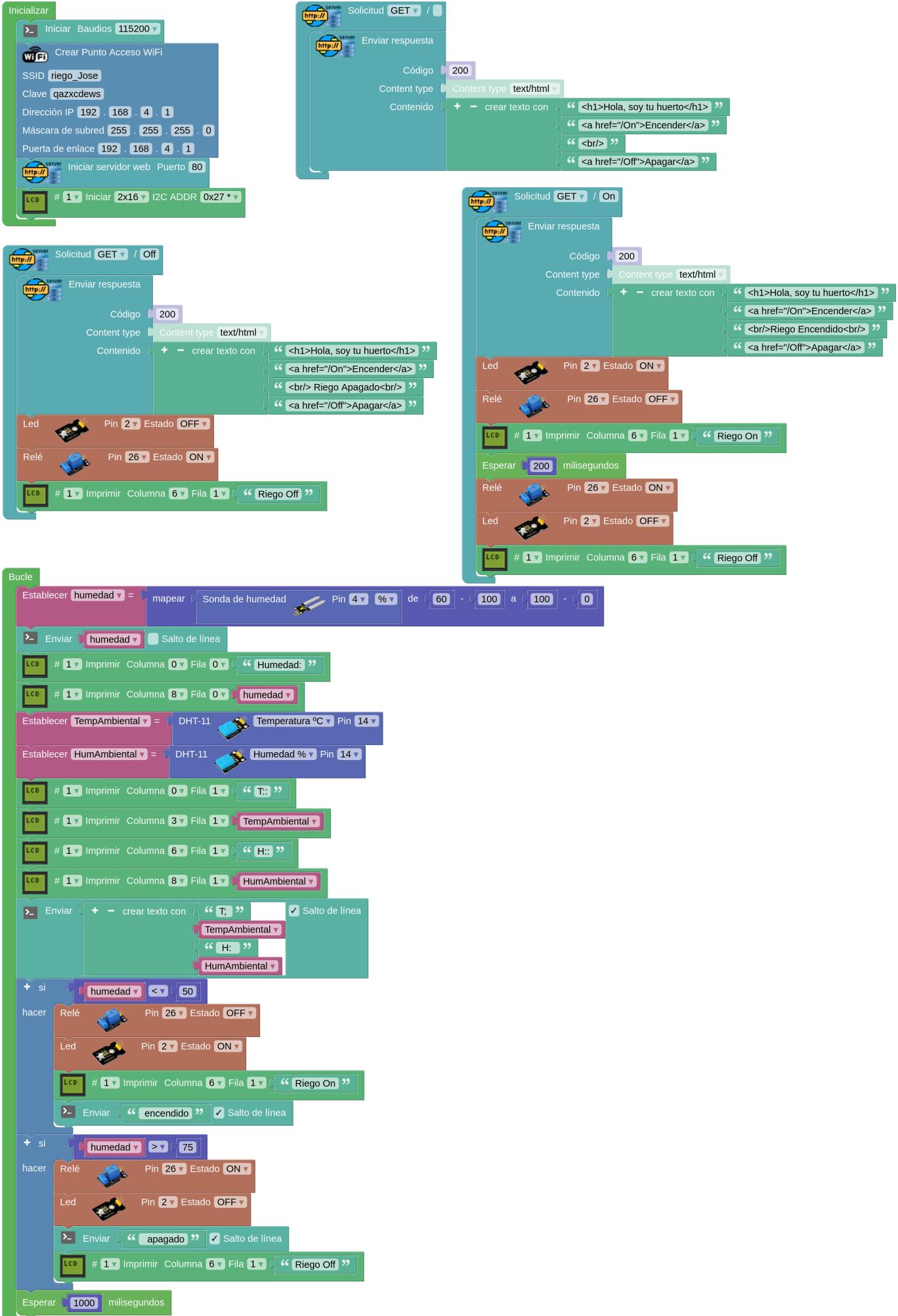


fritzing

- Wemos R32 D1
- relé - pin 26
- pila 9v
- cable usb
- bomba de riego - conectada al relé
- sensor de humedad - pin 4
- Pantalla LCD I₂C
- sensor DHT11 en pin 1

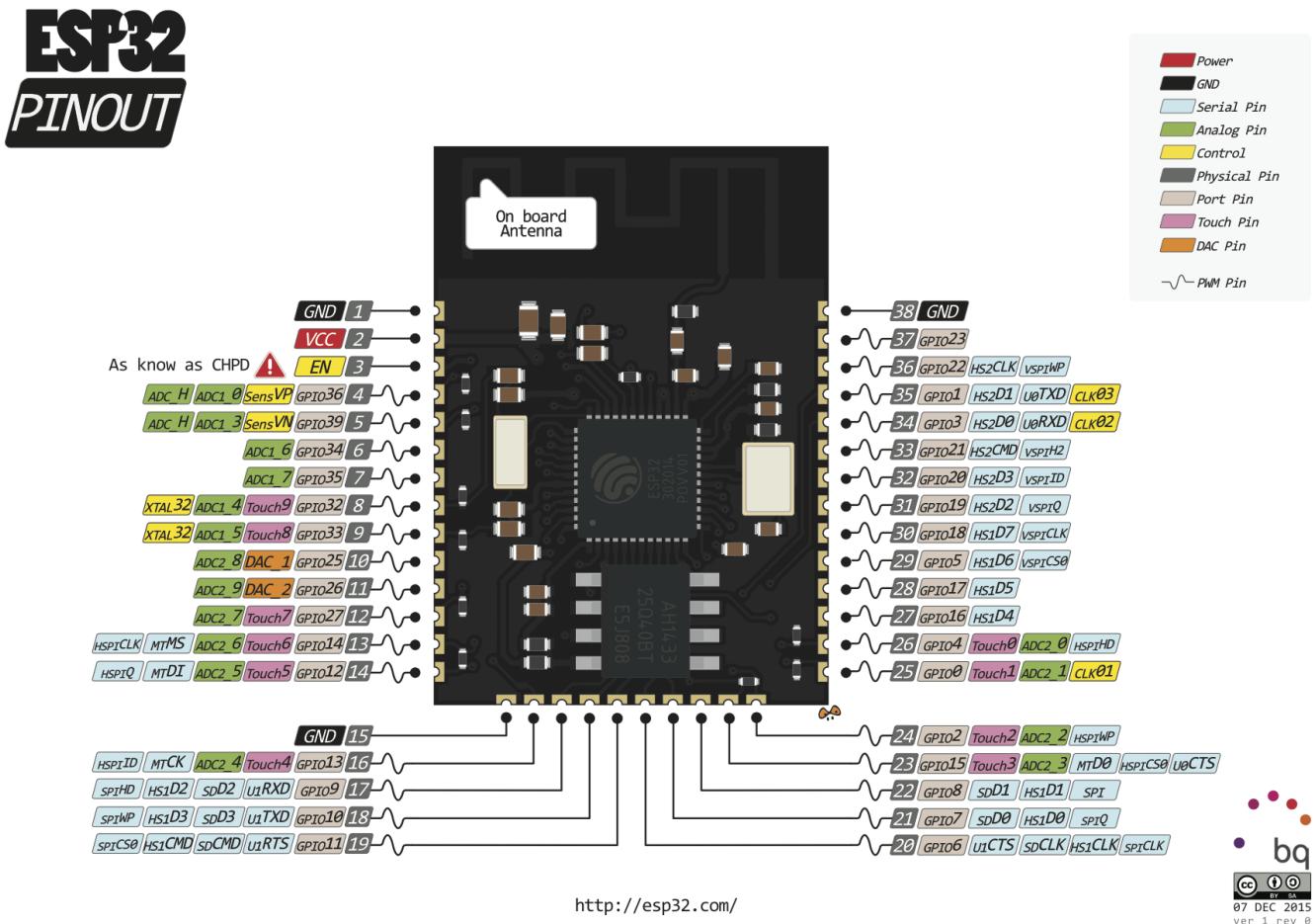
Mostraremos los datos de los sensores y el estado del riego/relé en la pantalla

Proyecto



Apéndice ESP32

Vamos a ver los detalles de la placa



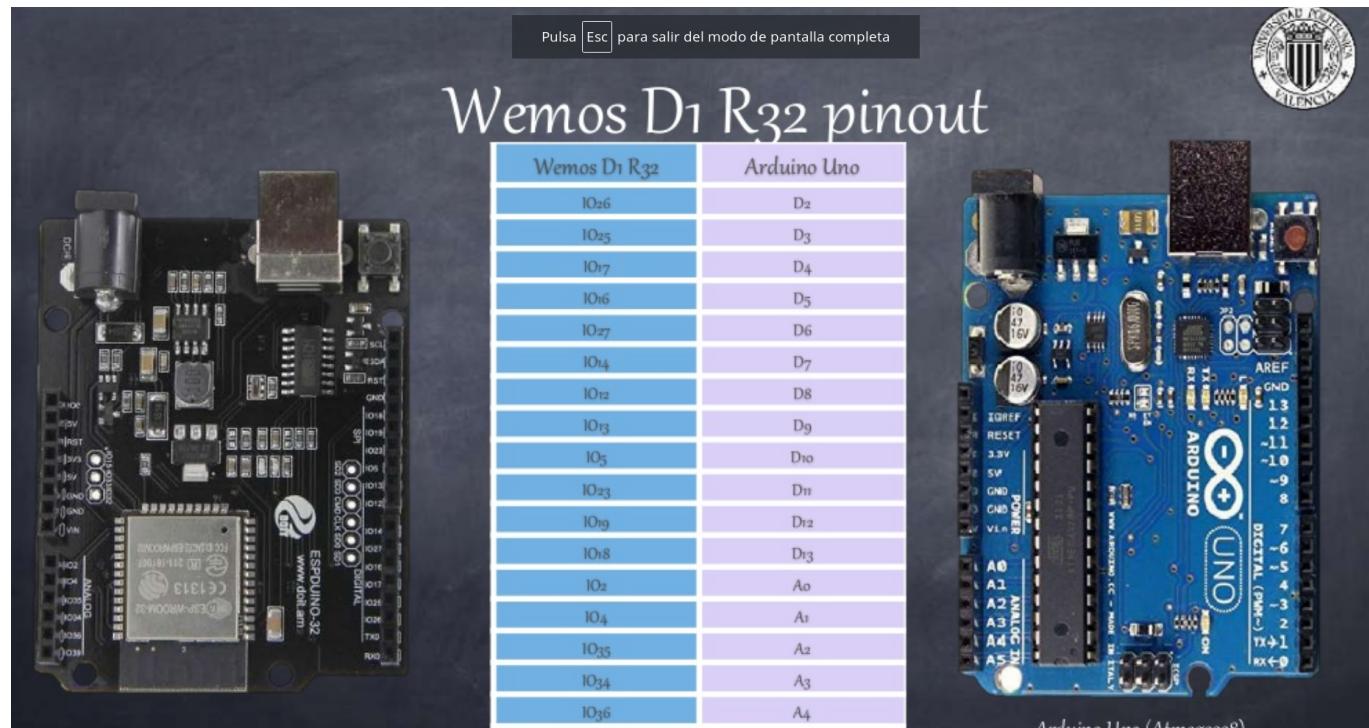
CPU: Xtensa dual-core 32-bit LX6 microprocessor 160 o 240 MHz hasta 600 DMIPS
 Ultra low power (ULP) co-processor
 Memoria: 520 KiB SRAM
 Wi-Fi: 802.11 b/g/n
 Bluetooth: v4.2 BR/EDR and BLE
 12-bit SAR ADC en 18 canales
 2 × 8-bit DACs
 10 × sensores capacitivos GPIOs
 4 × SPI
 2 × I²S interfaces
 2 × I²C interfaces
 3 × UART
 SD/SDIO/CE-ATA/MMC/eMMC
 SDIO/SPI slave
 Ethernet MAC con DMA
 CAN bus 2.0
 Controlador Infrarrojo (TX/RX, en 8 canales)
 Motor PWM
 LED PWM (up to 16 channels)
 Sensor de efecto Hall
 Ultra low power analog pre-amplifier

[Más detalles](#)

Correspondencia Arduino UNO - [Wemos D1 R32](#)

Alimentación: 5-12VDC
WiFi 802.11 b/g/n/e/i (802.11n hasta 150 Mbps)
Bluetooth v4.2 BR/EDR y BLE.
Compatible con shields de Arduino Uno.
6 Entrada analógica.
20 Entradas/Salidas digitales (3.3V) (con funciones PWM, interrupción).
Comunicación UART, SPI, I2C.
Tamaño: 68×53 mm
Conexión micro USB.
4Mbytes Flash Memoria 520Kb
Reloj: 240Mhz (un núcleo dedicado al procesar WiFi)
Temperatura: -40C+85C
Corriente: 250mA (max)
Corriente en modo ahorro: 0.15mA
Corriente de funcionamiento: 20mA (sin WiFi)

Pulsa Esc para salir del modo de pantalla completa



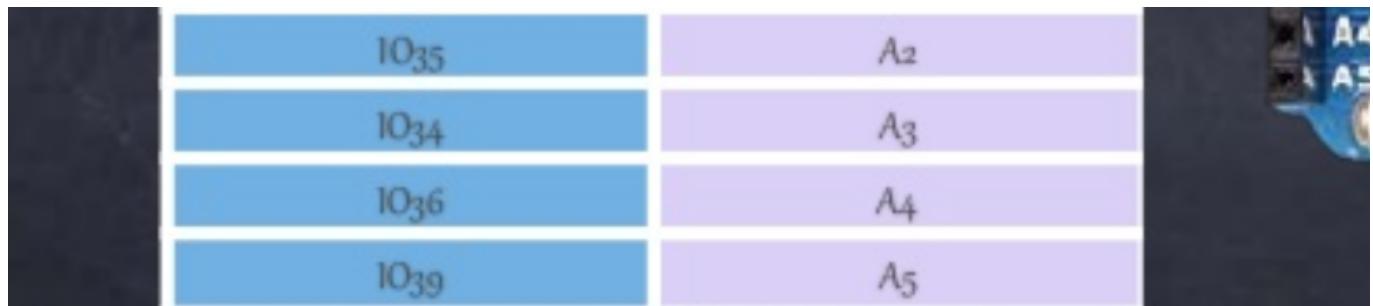
Wemos D1 R32 pinout

Wemos D1 R32	Arduino Uno
IO26	D2
IO25	D3
IO17	D4
IO16	D5
IO27	D6
IO14	D7
IO12	D8
IO13	D9
IO5	D10
IO23	D11
IO19	D12
IO8	D13
IO2	A0
IO4	A1
IO35	A2
IO34	A3
IO36	A4

Arduino Uno (Atmega328)

Wemos D1 R32 pinout

Wemos D1 R32	Arduino Uno
IO26	D2
IO25	D3
IO17	D4
IO16	D5
IO27	D6
IO14	D7
IO12	D8
IO13	D9
IO5	D10
IO23	D11
IO19	D12
IO8	D13
IO2	A0
IO4	A1

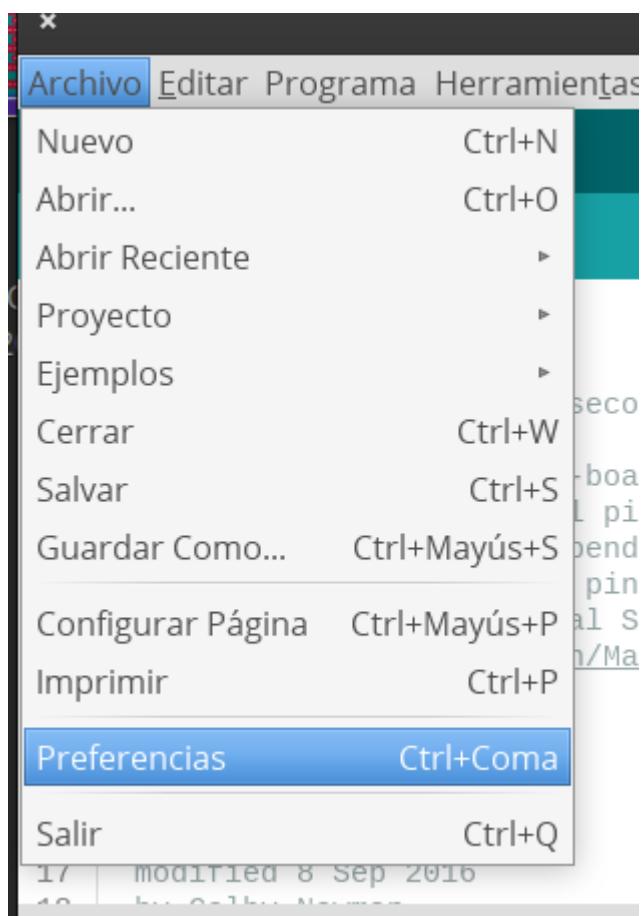


Imágenes cortesía de [Leopoldo Armesto Ángel](#), Lecturer at Universitat Politècnica de València (UPV)

Configuración del IDE de Arduino

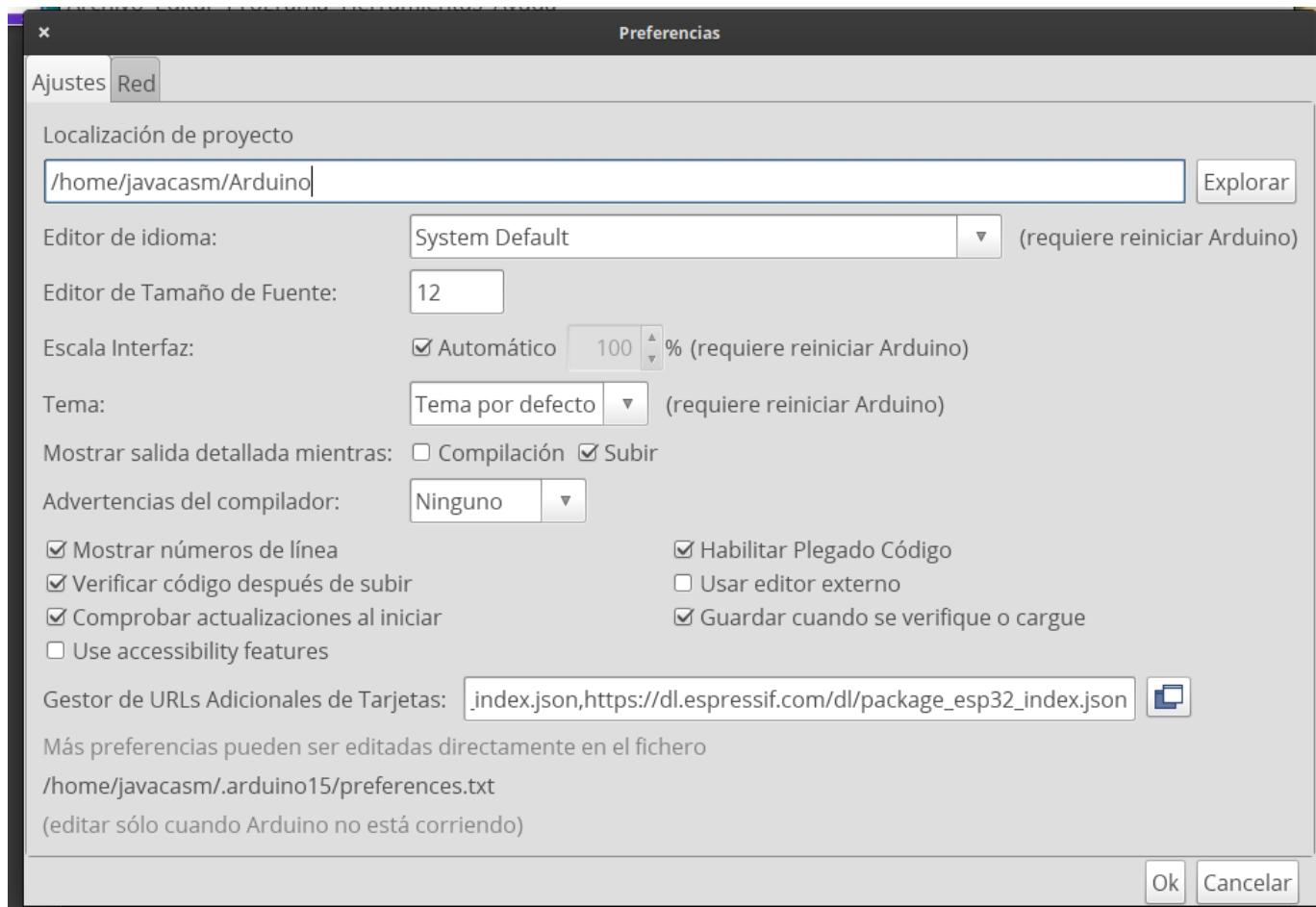
Si queremos usar el código de los programas tendremos que configurar el IDE de Arduino

En el menú Preferencias

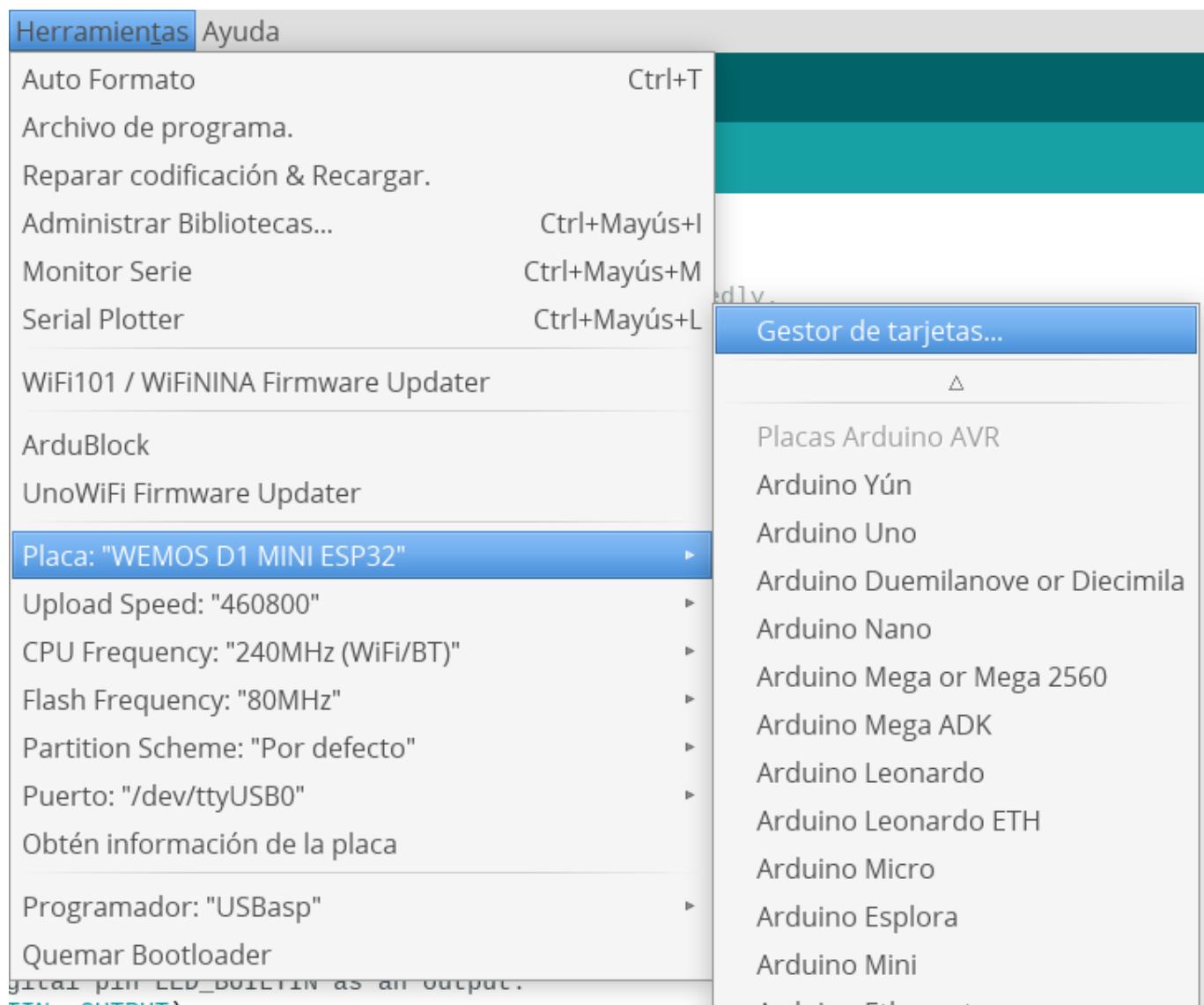


Añadimos la dirección del paquete de hardware de Espressif

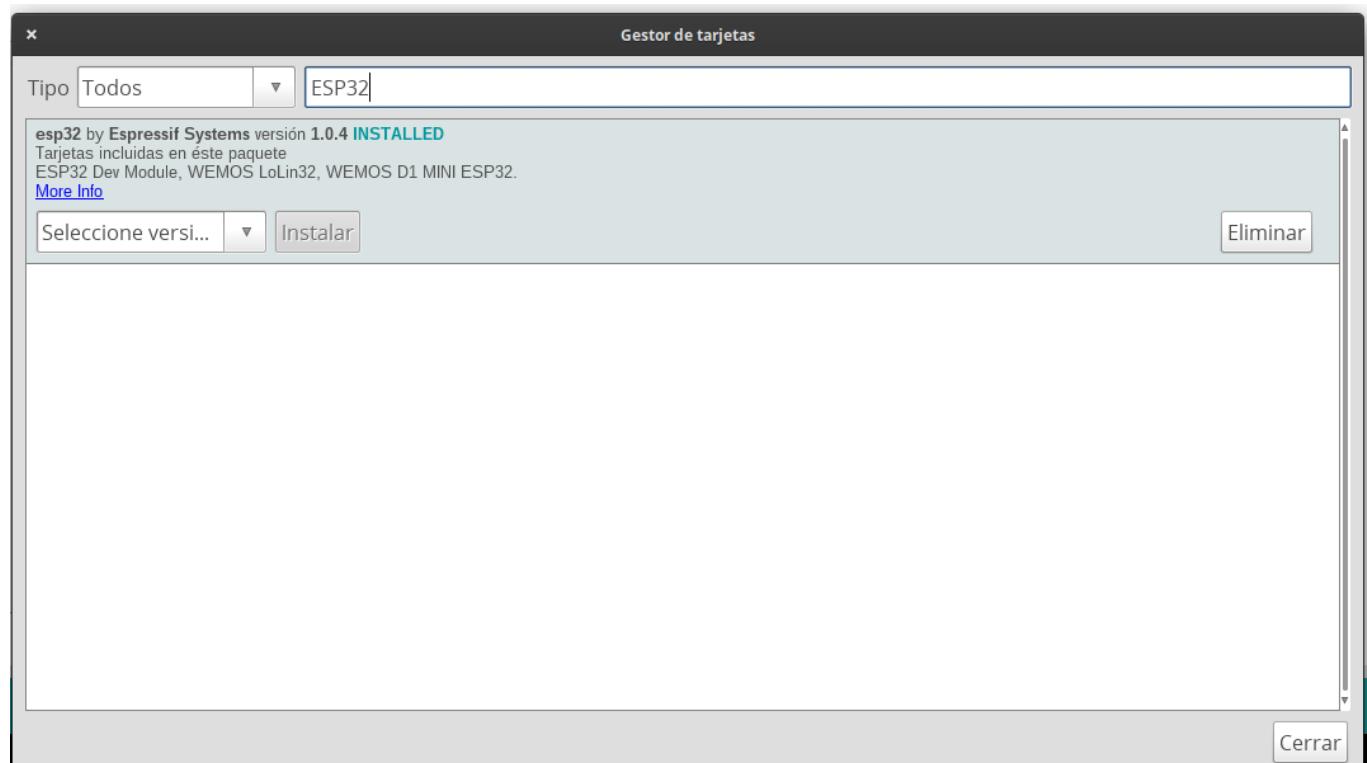
https://dl.espressif.com/dl/package_esp32_index.json



Ahora en gestor de tarjetas



buscamos **ESP32**



Y lo instalamos.

Ahora podremos seleccionar las placas de tipo **ESP32**

Aunque no viene nuestra placa, podemos seleccionar una compatible: **DOIT ESP32 DEVKIT V1**



Driver

[Drivers para CH341 en windows](#)