Proyecto 2ºASIR

Sistema IoT Cloud para la monitorización y visualización de parámetros ambientales

Por Méndez y Cuenca.

Índice

[**Introducción**](#_gjdgxs) **3**

[**Hardware**](#_30j0zll) **4**

[Wemos D1 mini pro](#_1fob9te) 5

[DHT22](#_3znysh7) 5

[MH-Z19B](#_2et92p0) 6

[Firmware](#_mu5fgc5cka0i) 9

[Instalación](#_cptfqxcvgnww) 10

[Dispositivos](#_6pf037gxxjty) 11

[Conexión al servidor](#_vh1eshwi21gh) 16

[Herramientas](#_npjtwzqfnbkw) 17

[**Software (Servidor)**](#_qyqn9byfmczb) **19**

[Creación](#_1t3h5sf) 19

[Mosquitto](#_4d34og8) 20

[Instalación](#_2s8eyo1) 20

[Configuración](#_17dp8vu) 20

[Pruebas](#_3rdcrjn) 21

[InfluxDB](#_26in1rg) 22

[Instalación](#_lnxbz9) 22

[Configuración](#_35nkun2) 23

[Telegraf](#_44sinio) 23

[Datos](#_2jxsxqh) 25

[Grafana](#_z337ya) 26

[Instalación](#_3j2qqm3) 26

[Configuración de gráficos](#_1y810tw) 26

[Alertas](#_4i7ojhp) 29

[**Bibliografía**](#_2xcytpi) **34**

# Introducción

El objetivo del proyecto es implementar una red IoT (Internet of Things) en diferentes aulas del instituto para recoger valores medioambientales como serían temperatura, humedad y nivel de CO2. El despliegue necesario para este proyecto se basa básicamente en el microcontrolador ESP8266 y varios sensores que recogerían valores medioambientales y que enviarán a través de Wifi. Estos datos son procesados por un broker MQTT y enviados para su almacenamiento a una base de datos a través de una aplicación llamada Telegraf. Para acceder a la información se usará Grafana, una herramienta que monitoriza y visualiza datos mediante representación con gráficas y tablas, con la finalidad de que la información representada sea más fácil de interpretar.

Un uso apropiado y útil de este proyecto de IoT, por la situación de pandemia actual, es el de medir los niveles de CO2 en las diferentes aulas, y si los niveles se superan en base a unos valores predefinidos, enviar alertas para que se actúe de forma inmediata.

La implementación de un medidor de CO2 (dióxido de carbono) consta de una una placa con un microcontrolador ESP8266, a la que están conectados los sensores, el del CO2, temperatura y humedad. Mediante esa placa lo que haremos es configurarla para que recoja los datos de los sensores de una forma periódica, cada minuto. Los sensores son clientes publicadores en un Broker MQTT, que recoge los datos para posteriormente enviarlos al agente Telegraf, y que a su vez inserta los valores recogidos en una base de datos temporal InfluxDB. Por último y mediante los datos de la base de datos temporal lo que se hará es generar unos gráficos mediante Grafana.

Una vez hecho todo el despliegue y visto que funciona correctamente le añadiremos un sistema de avisos mediante correo electrónico, esto es por ejemplo si se llega a una situación en la que hay demasiado CO2 en el aula, se pueda notificar y hacer el cambio de clase.

# **Hardware**

Necesitaremos una placa Wemos d1 mini pro, un sensor DHT22 (temperatura y humedad) y un sensor MHZ19B para CO2.

También añadiremos una pantalla LED para mostrar los valores a tiempo real

Estos componentes irán montados sobre una protoboard y conectado con jumpers a los pines.

Todo va dentro de una caja a medida hecha con impresora 3D cuyo código estará subido a github.

El modelo está hecho con tinkercad y pasado a código G que es el que interpreta la impresora con un programa llamado Slicer.

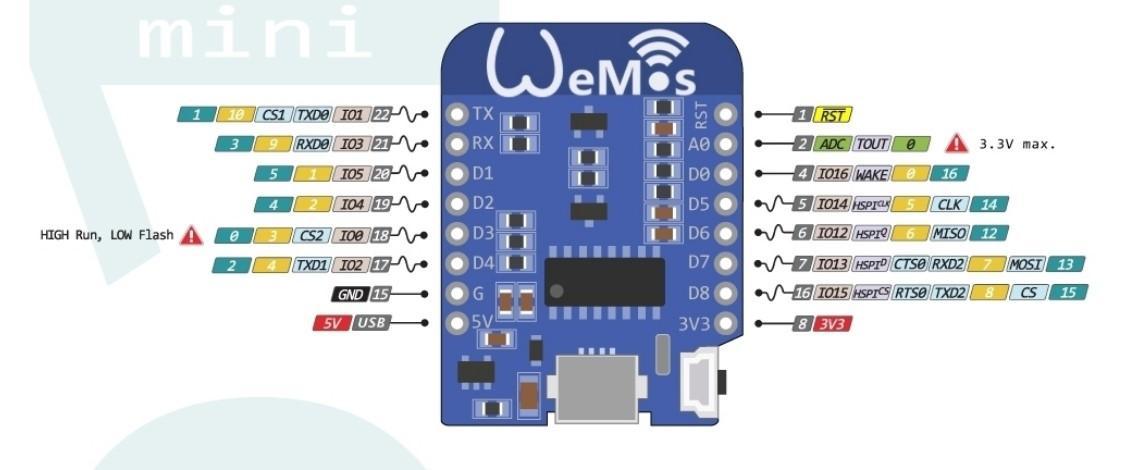
Se imprimió en un aula de nuestro centro, el Aula Ateca, que es una sala donde hay bastantes recursos para el uso de los alumnos.



## Wemos D1 mini pro

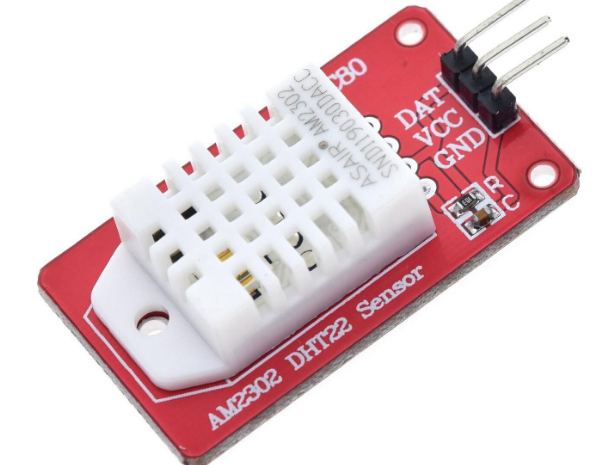
La placa que necesitamos para el proyecto es la Wemos D1 mini pro de 16 mb de memoria con el chip ESP8266, esta placa es una de las mejores que hay por ese rango de precios, entre 4 y 5 euros. Además con esta placa han trabajado unos compañeros míos realizando un proyecto y ellos también se decantaron por ella. Hemos elegido esta placa del fabricante Wemos frente a otros fabricantes como sería Olimex u Adafruit, en el caso de Olimex no posee ningún puerto USB y en cuanto a Adafruit sería una placa con más características de las que necesitamos.

En cuanto a las ventajas que tiene es que posee un puerto USB mediante el cual nos conectamos a ella para introducirle las librerías y demás información, tiene un botón de reset y la más importante de todas tiene 16 mb de memoria.



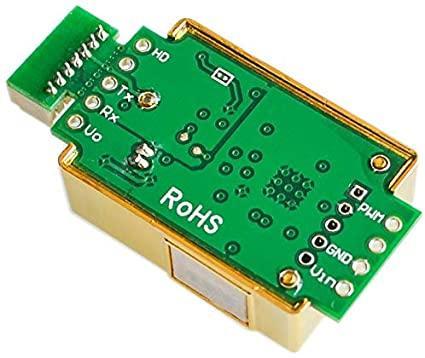
## DHT22

El DHT22 es un sensor que se utiliza para hacer mediciones de temperatura y de humedad, este sensor es del fabricante Adafruit. La ventaja más clara que he tenido en cuenta para decantarme por este y no por otro sensor es que este es digital, a diferencia de otros sensores como el LM35, este utiliza un pin digital mediante el cual se envía la información, esto hace que haya menos ruido, es decir que la información que recibamos sea más acorde con la realidad.



## MH-Z19B

El sensor de vamos a utilizar es el MH-Z19B, este sensor es de los mejores que hay calidad precio, hay uno que es un poco más caro el Senseair s8, este es un poco mejor que este sensor, pero por falta de tiempo a la hora de pedir los componentes, nos hemos decantando por el MH-Z19B que también es uno de los mejores sensores que hay actualmente en el mercado.



Definimos los pines

Comunes:

Tanto los sensores como la placa llevan 2 jumpers que van del pin GND a tierra , y del pin de 5V a corriente.

MHZ19B

Conectamos el pin RX a TX y viceversa

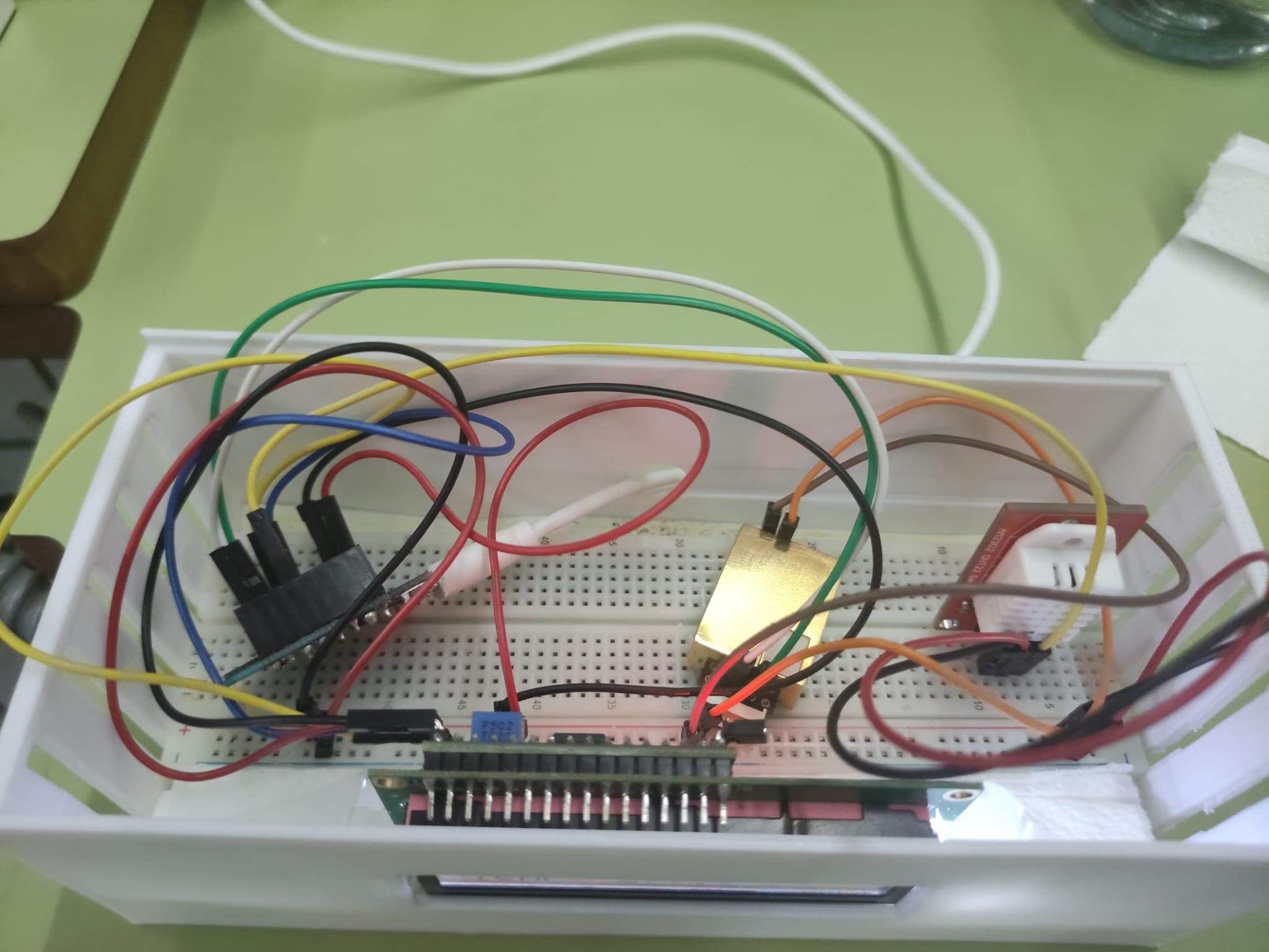
DHT22

Conectaremos el pin de Datos (dat) al pin D5.

Pantalla LED:

Lleva el pin SDA conectado a D2 y el pin SCL a D1

\*Excepto los del sensor de CO2 (MHZ19B), es indiferente en cual D estén conectados mientras se especifique en la configuración del firmware.\*



## **Firmware**

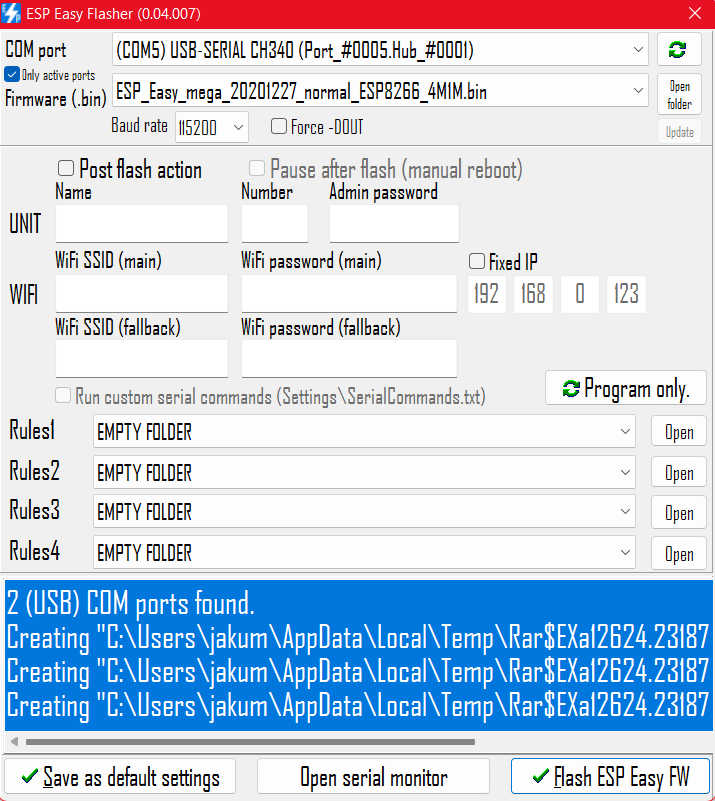
Utilizaremos el firmware ESP-Easy proporcionado por

<https://emariete.com/medidor-casero-co2/>

Lo primero que tienes que hacer es descargar al ordenador el programaqueluego cargarás en el Wemos D1 Mini y que administrará todo el funcionamiento.

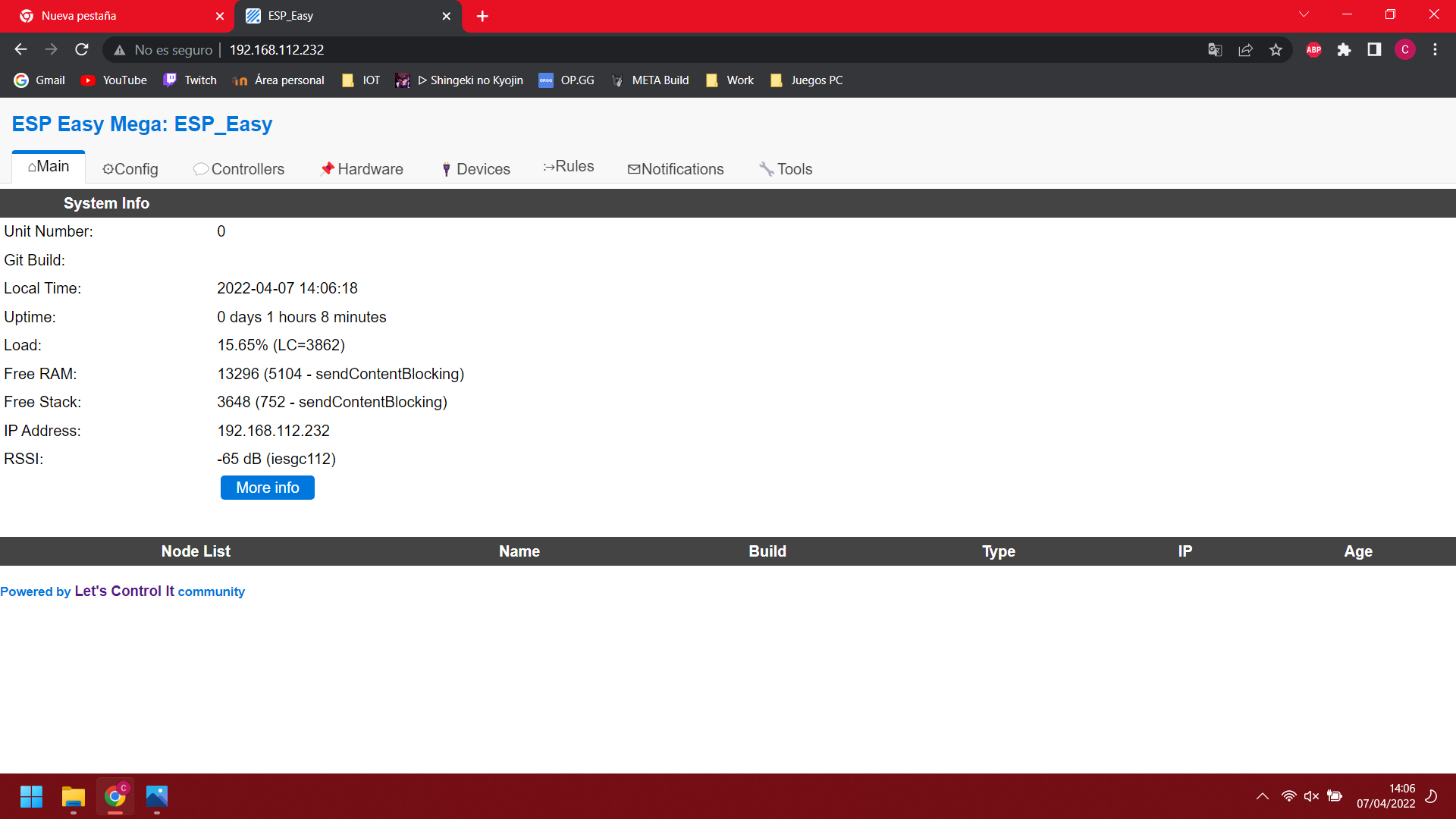
Hay que conectar la Wemos por USB al ordenador y ejecutar el programa. Esto creará un Punto de acceso Wifi temporal llamado ESP-EASY con la contraseña “configesp”. Hay que conectarse al mismo y nos pedirá una acción en el navegador, en este punto tendremos que configurar el WIFI al que se conectará la Wemos.

Es muy importante desconectar los pines TX t RX del sensor de CO2 al instalar el firmware ya que podría crear conflicto con el USB (desconéctalos temporalmente si ya los conectaste)

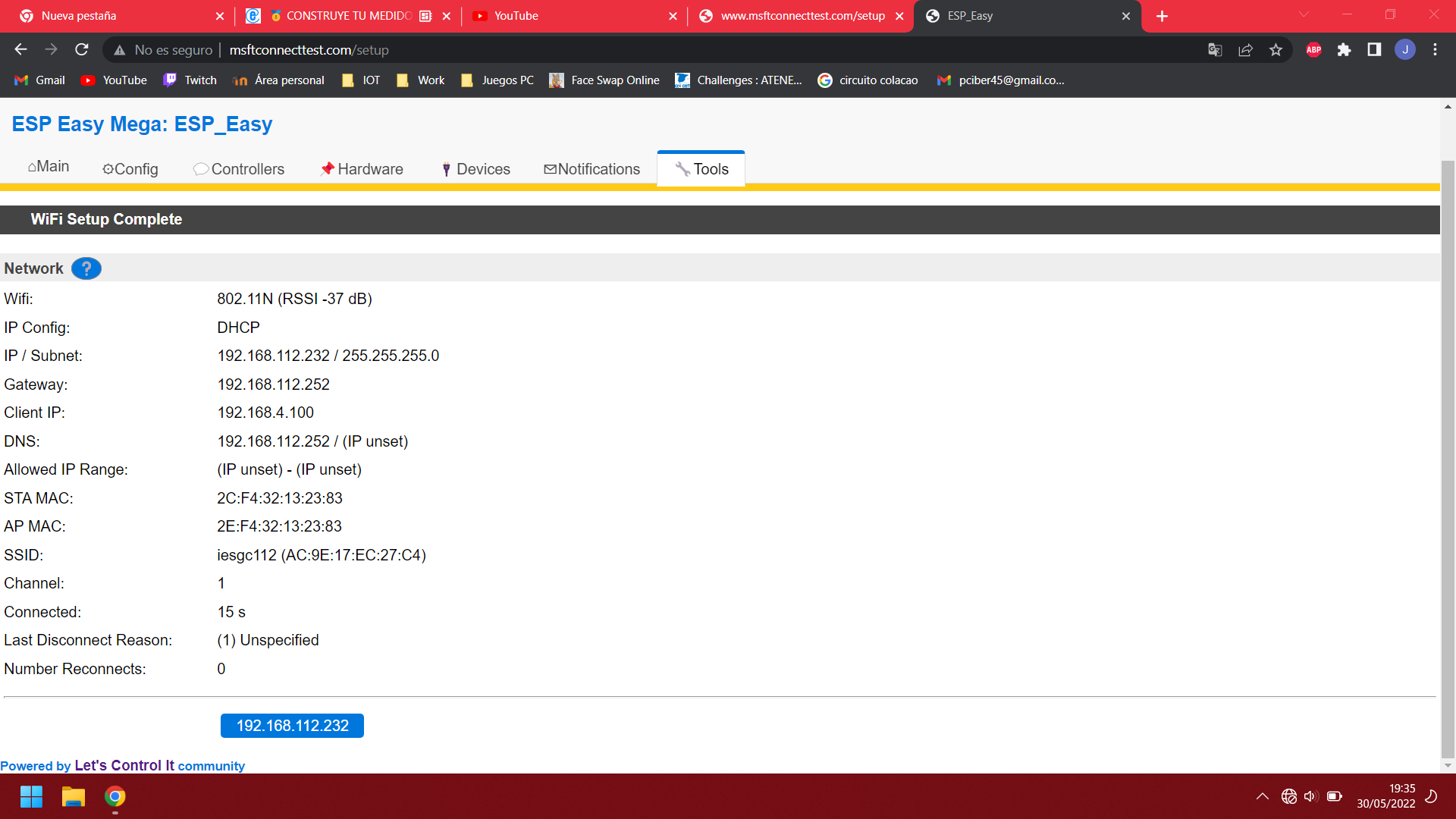


### Instalación

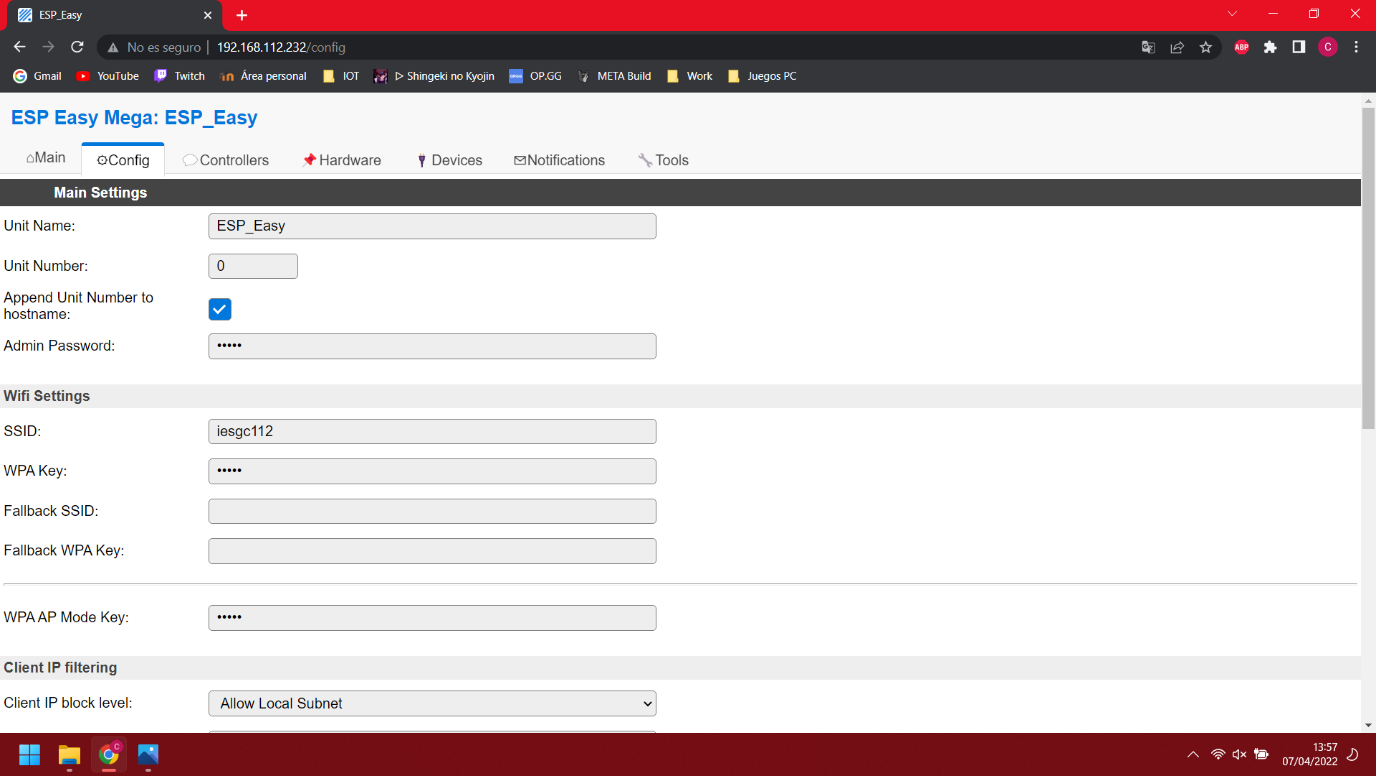
Definimos el wifi y su contraseña y ya entramos en la interfaz del programa. En este punto ya podemos volver a conectar nuestro WIFI y con sólo introducir la IP que nos proporciona esp-easy en el navegador ya podemos acceder a la interfaz de configuración.



En la pestaña “Tools” que analizaremos más adelante podemos ver más información de la red.

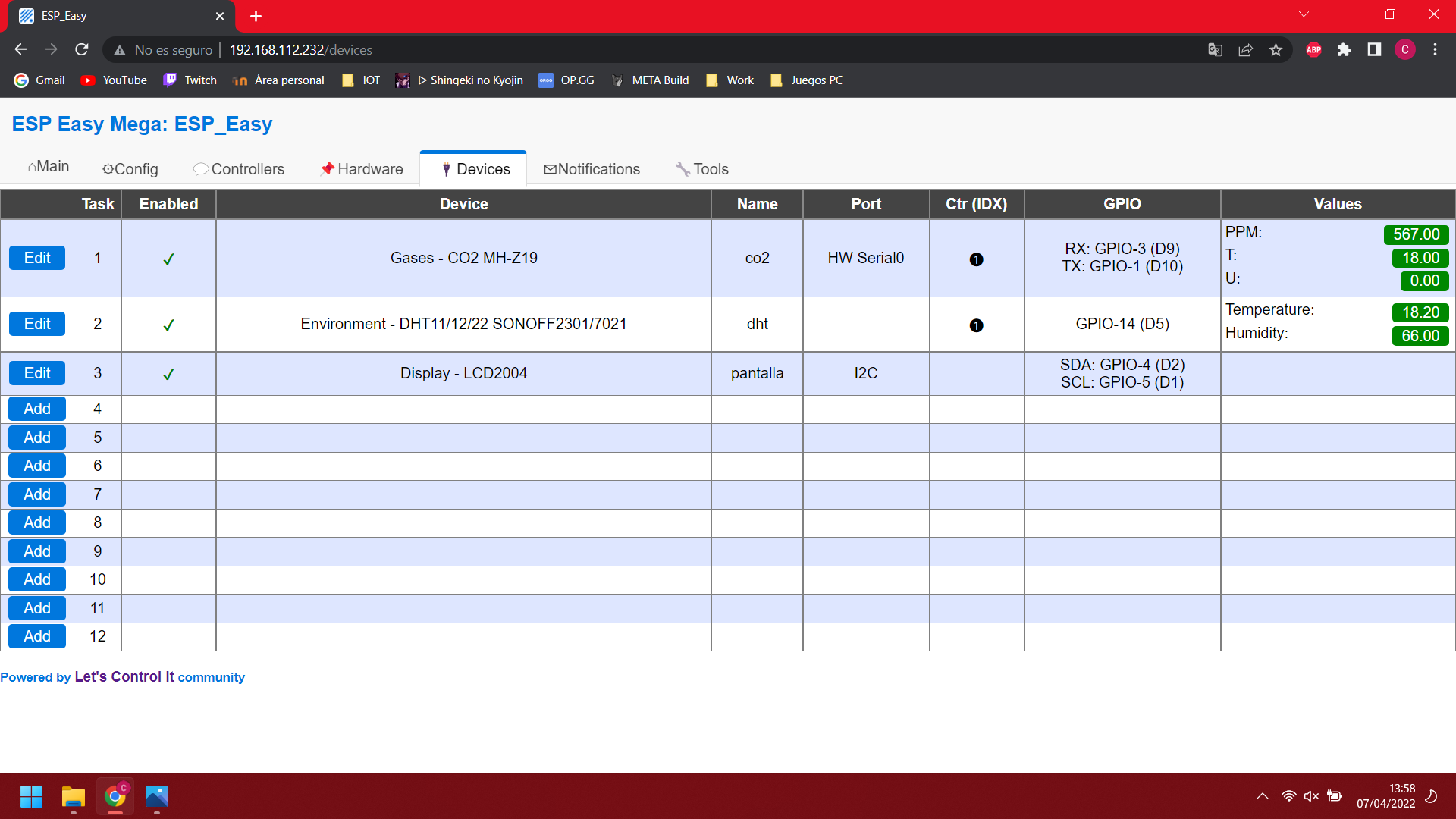


En “Config” podemos cambiar el Wifi, el nombre del dispositivo y la contraseña para acceder al mismo.



### Dispositivos

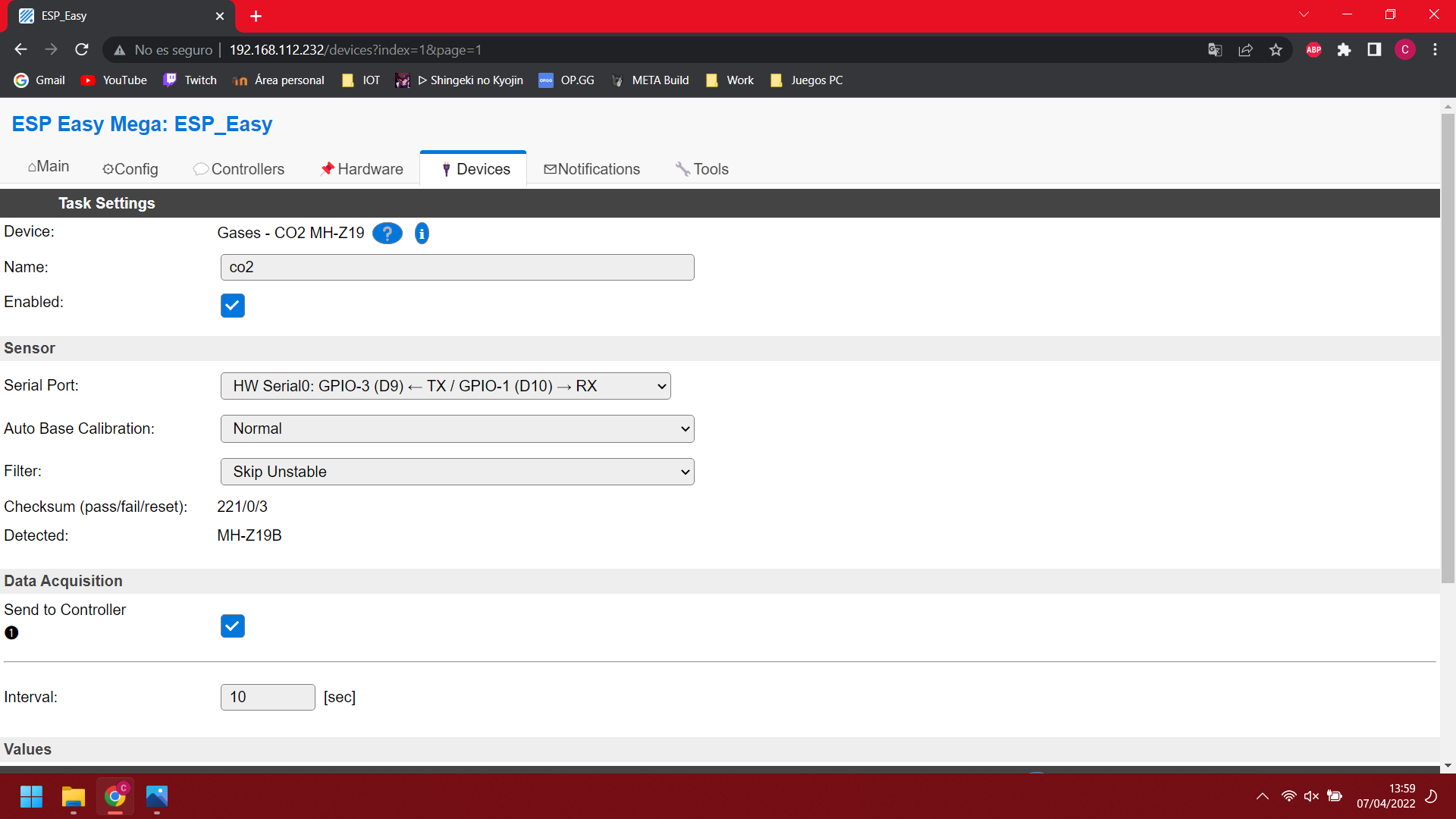
Lo primero es indicarle los dispositivos que están conectados, esto lo haremos desde la pestaña “Devices”.

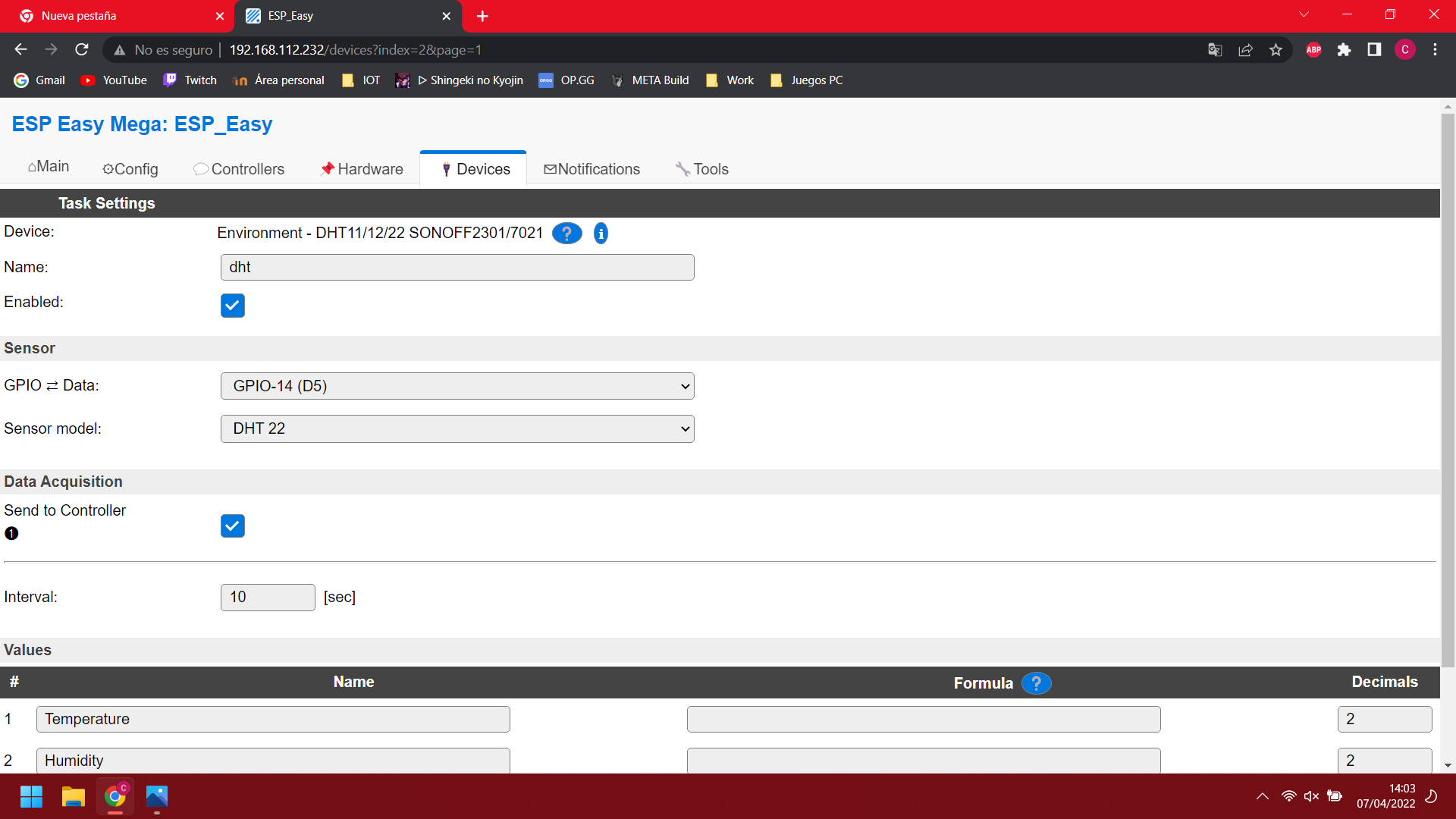


Esta es la ventana principal en la que nos muestra los dispositivos que tenemos conectados, podemos editar sus configuraciones o añadir uno nuevo.

Estas son las configuraciones de los 2 sensores.

Aquí les indicamos los pines a los que se conectan, el nombre visible, que estén habilitados y que manden los datos al controlador(esto último lo haremos a continuación)

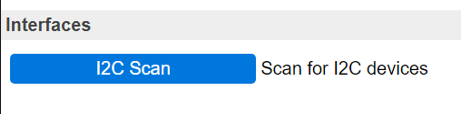


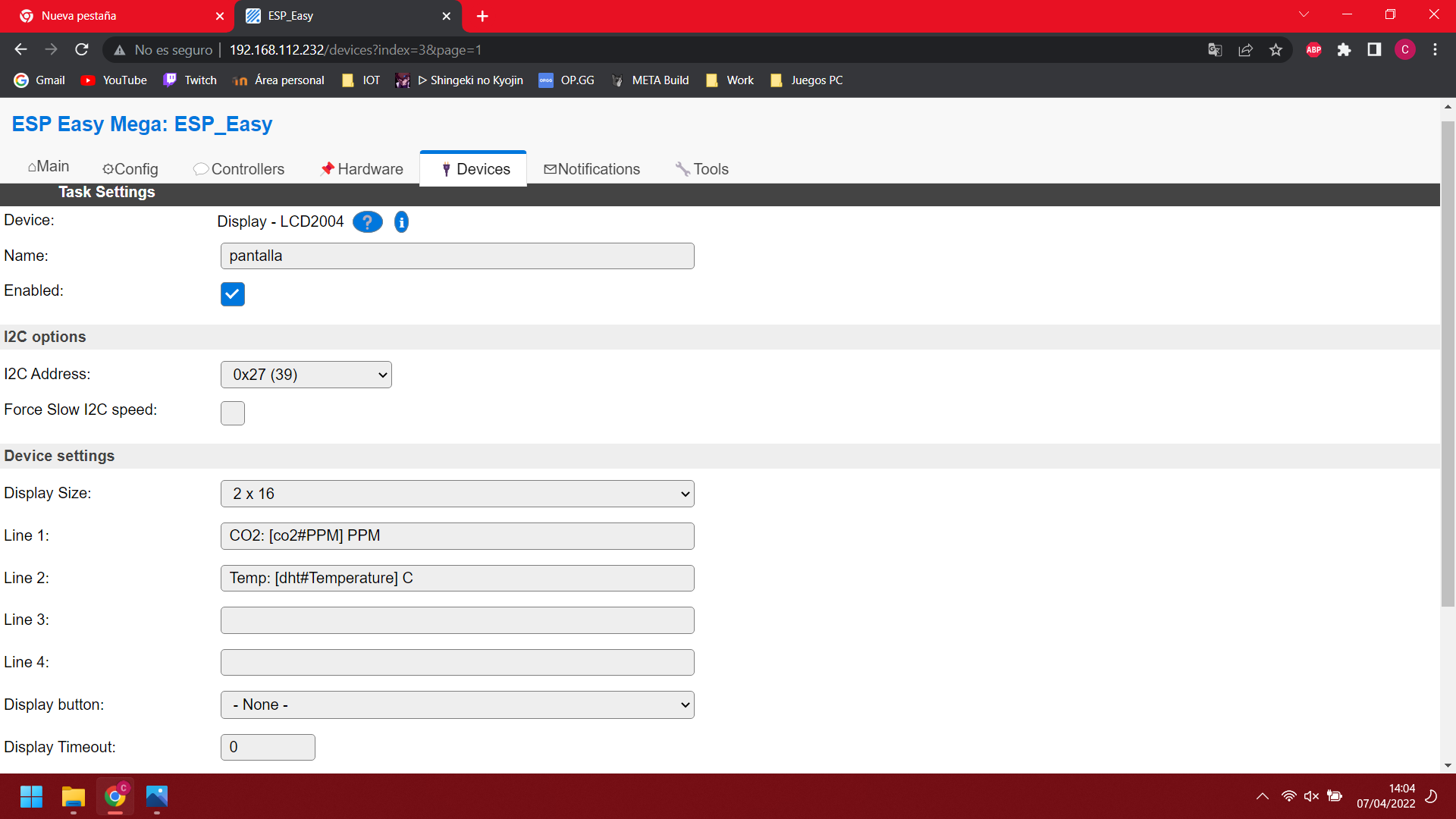


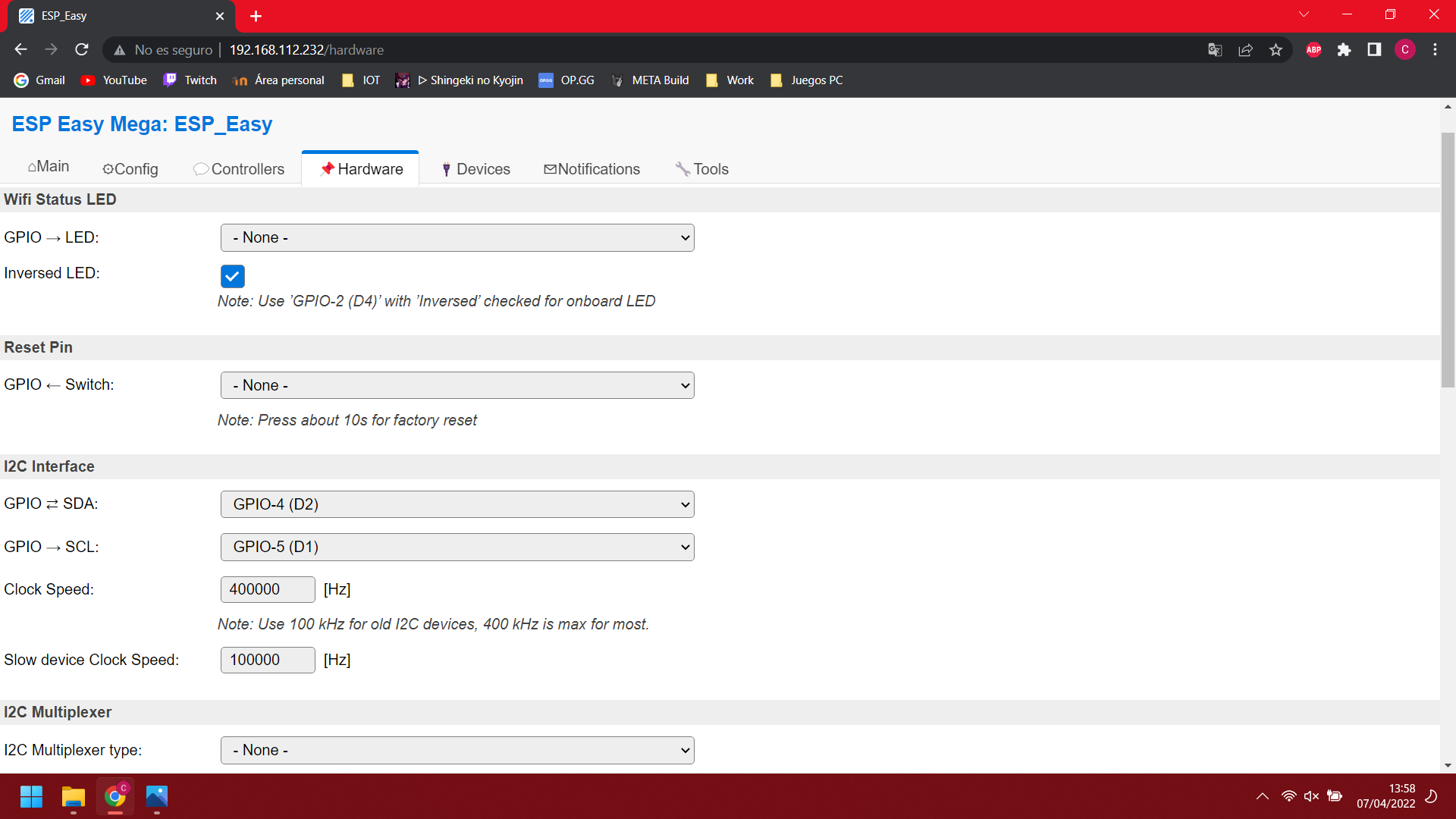
Para la pantalla indicamos el nombre del dispositivo, el tamaño y el texto que queremos que muestre que almacena en variables.

También hay que indicarle los pines en la pestaña “hardware” en I2C interface.

Es posible que no detecte la pantalla por lo que habría que buscarla desde “Tools” aquí:

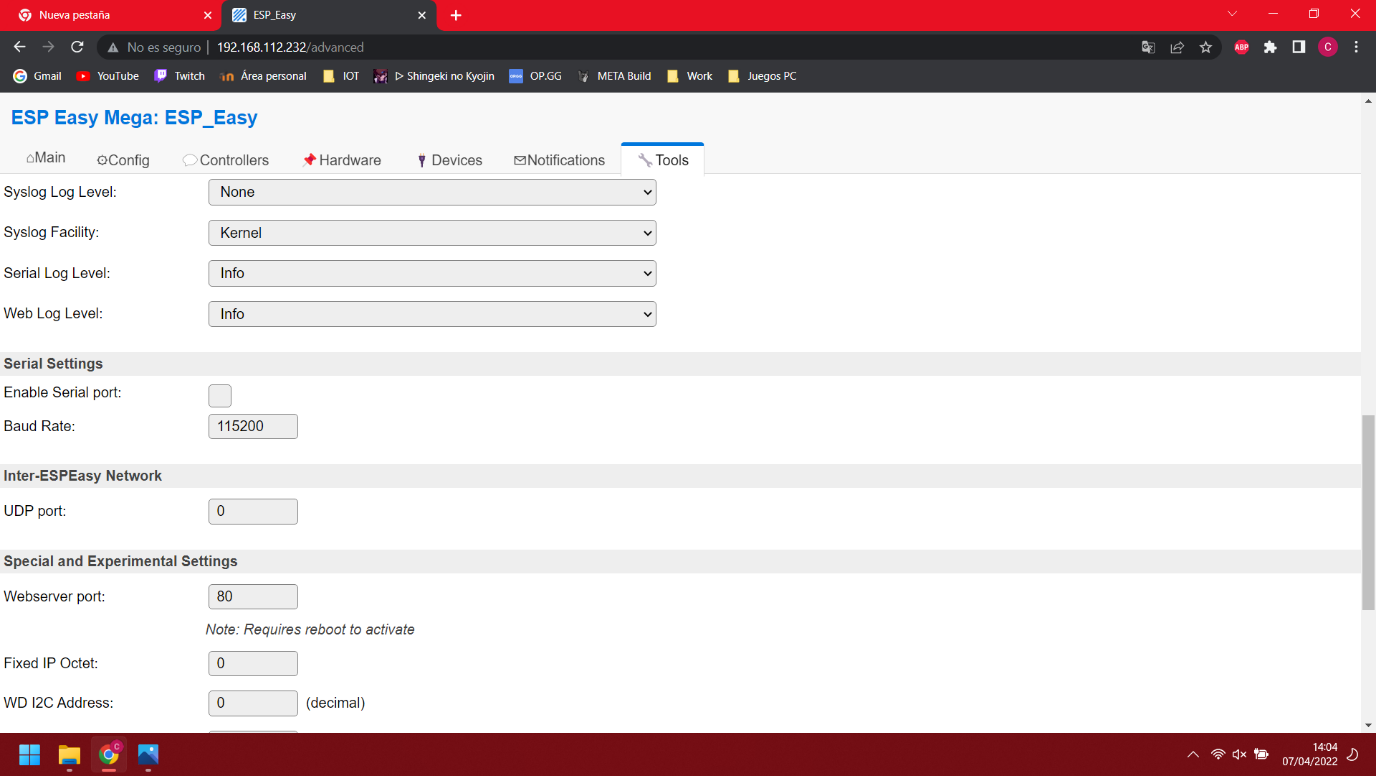




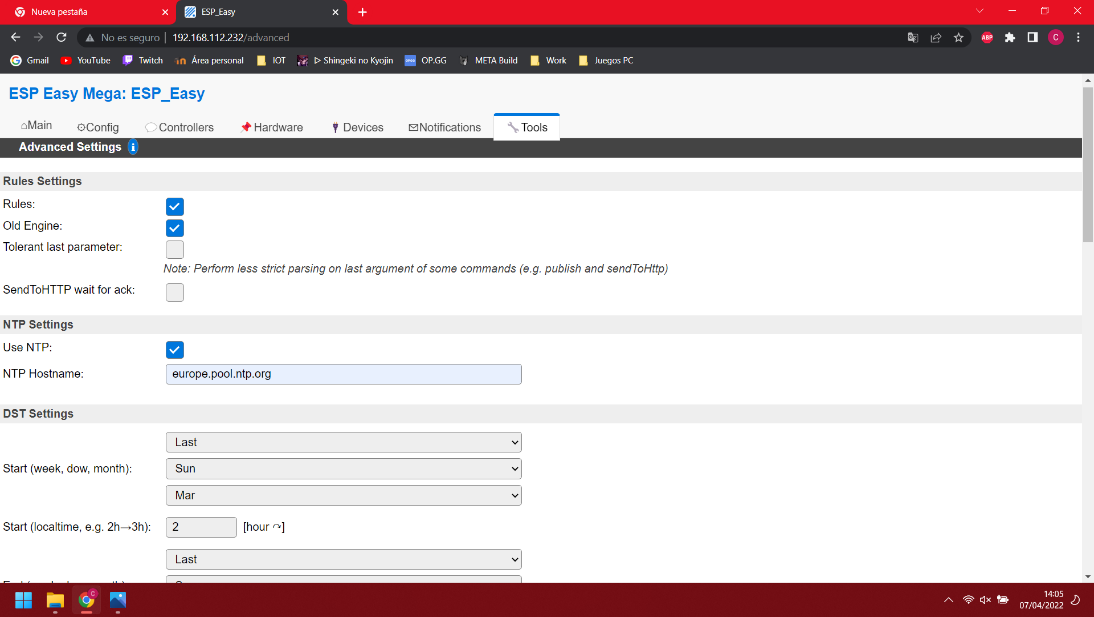


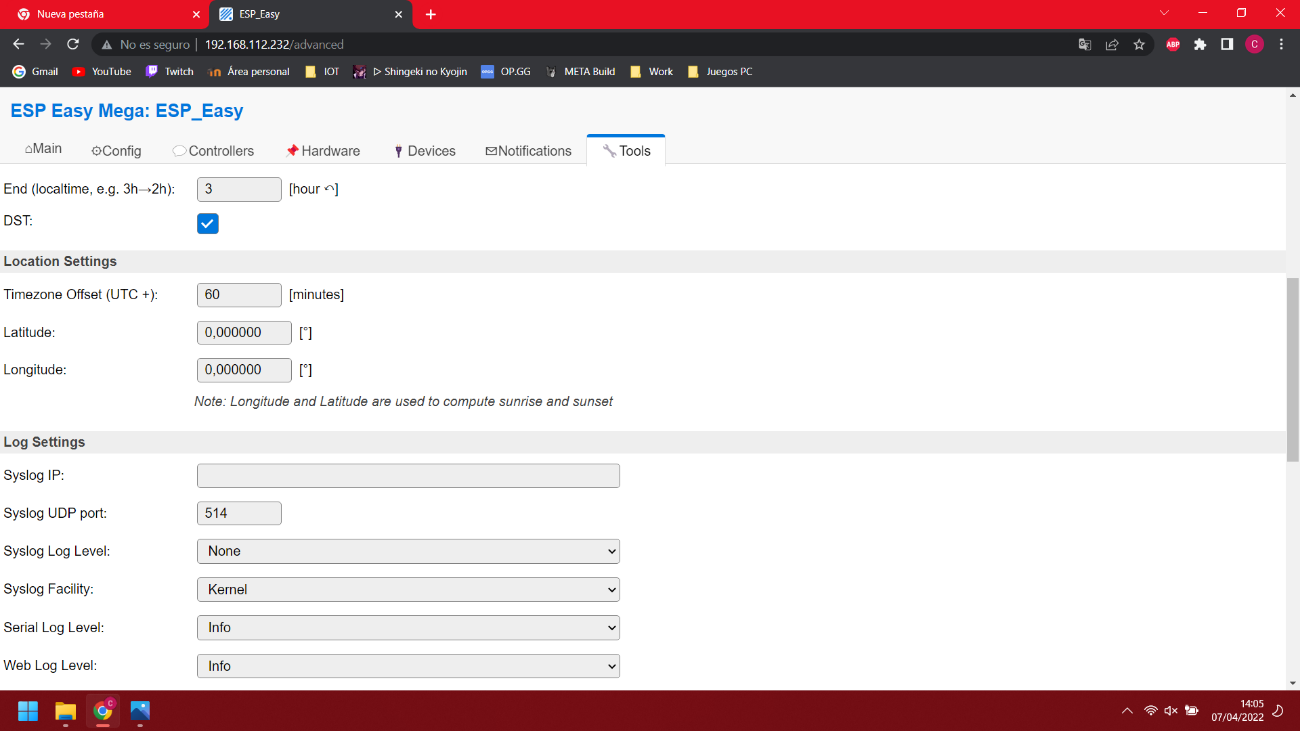
Hay que ajustar el Baud rate en advanced settings desde “Tools”





Ya que estamos aquí aprovechamos para ajustar la zona horaria del dispositivo con la siguiente configuración.



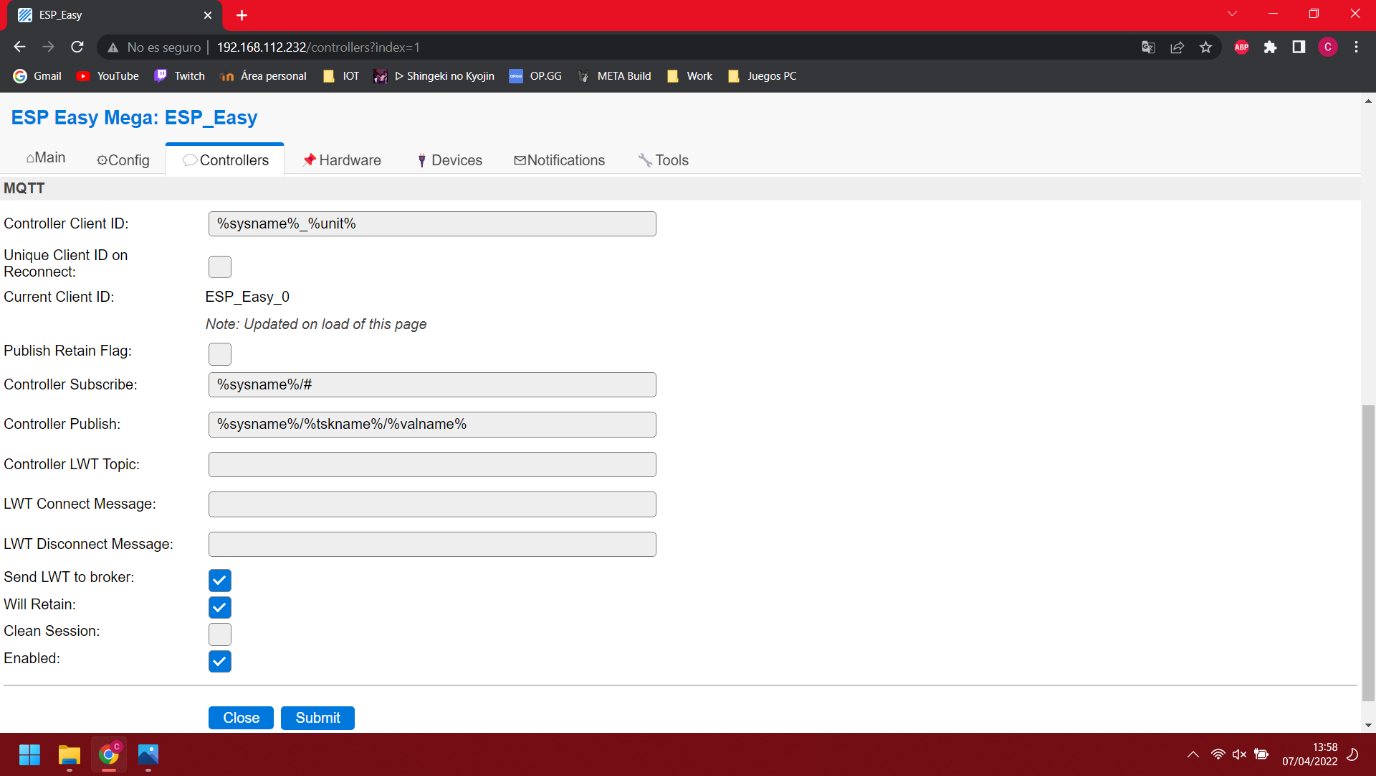
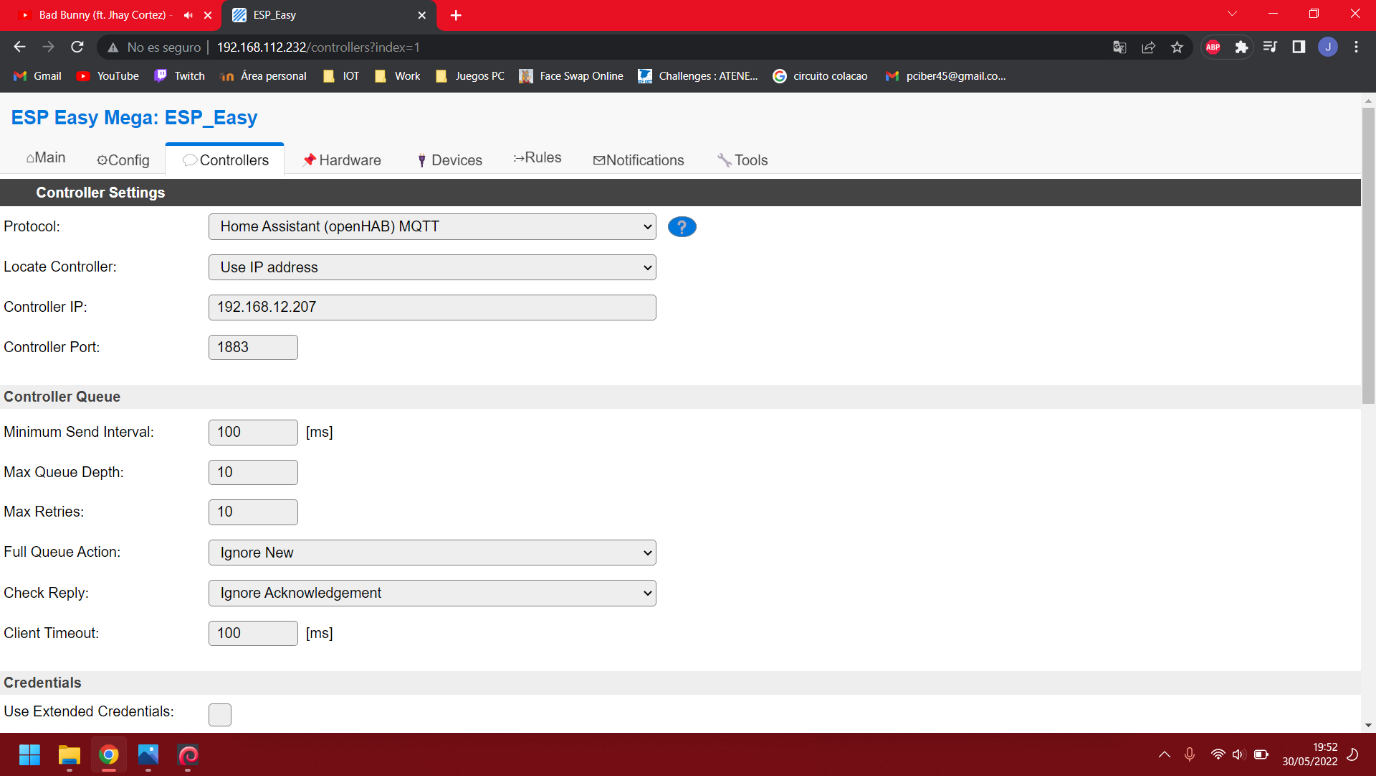


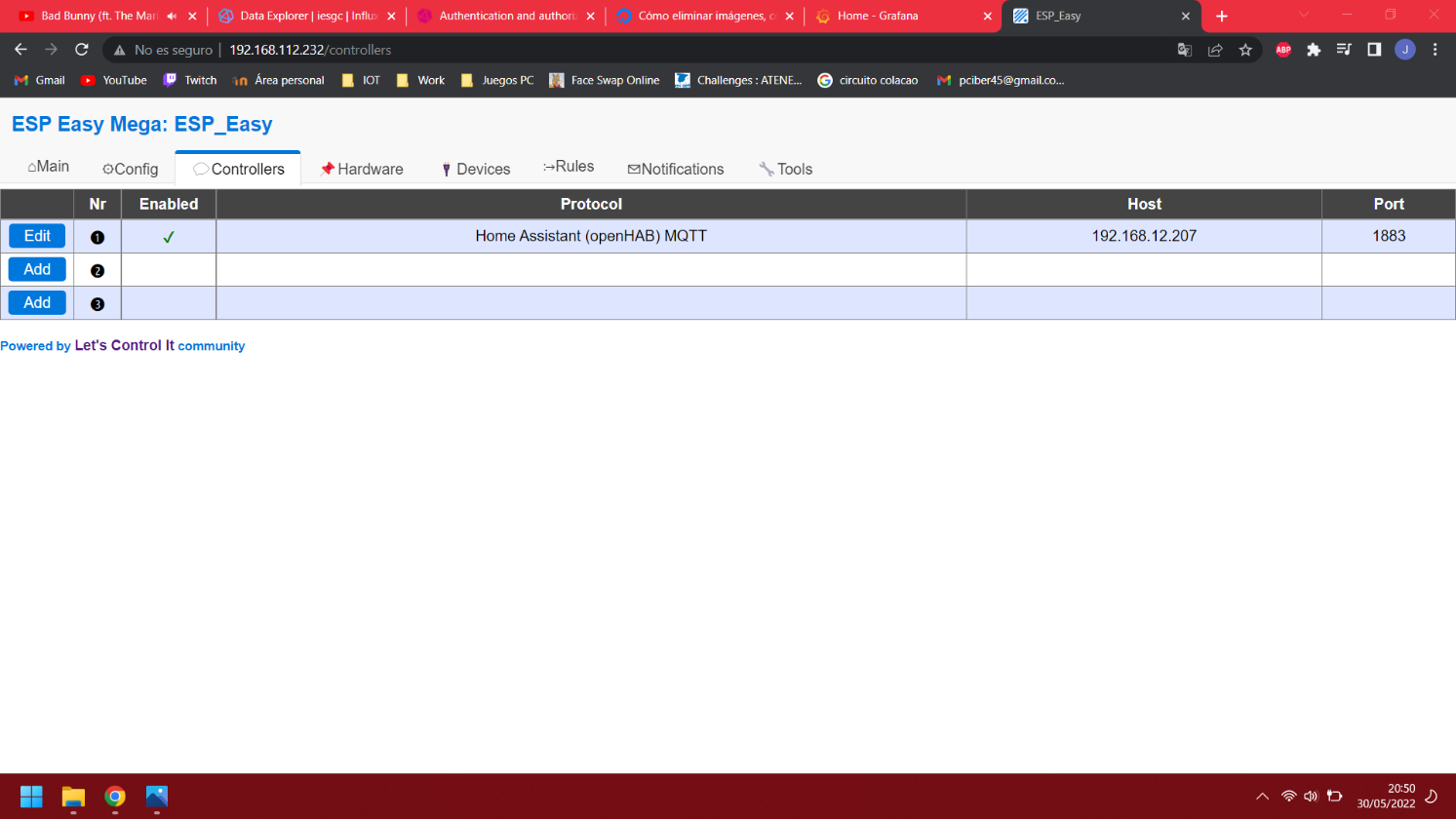
### Conexión al servidor

Ya están los dispositivos funcionando, sólo queda que envíen la información a nuestro servidor.

Vamos a la pestaña de “Controllers” y añadimos uno nuevo.

Tenemos que indicarle el protocolo, en este caso MQTT, la IP en la que está alojado el servidor y el puerto además de las otras opciones que necesitemos.





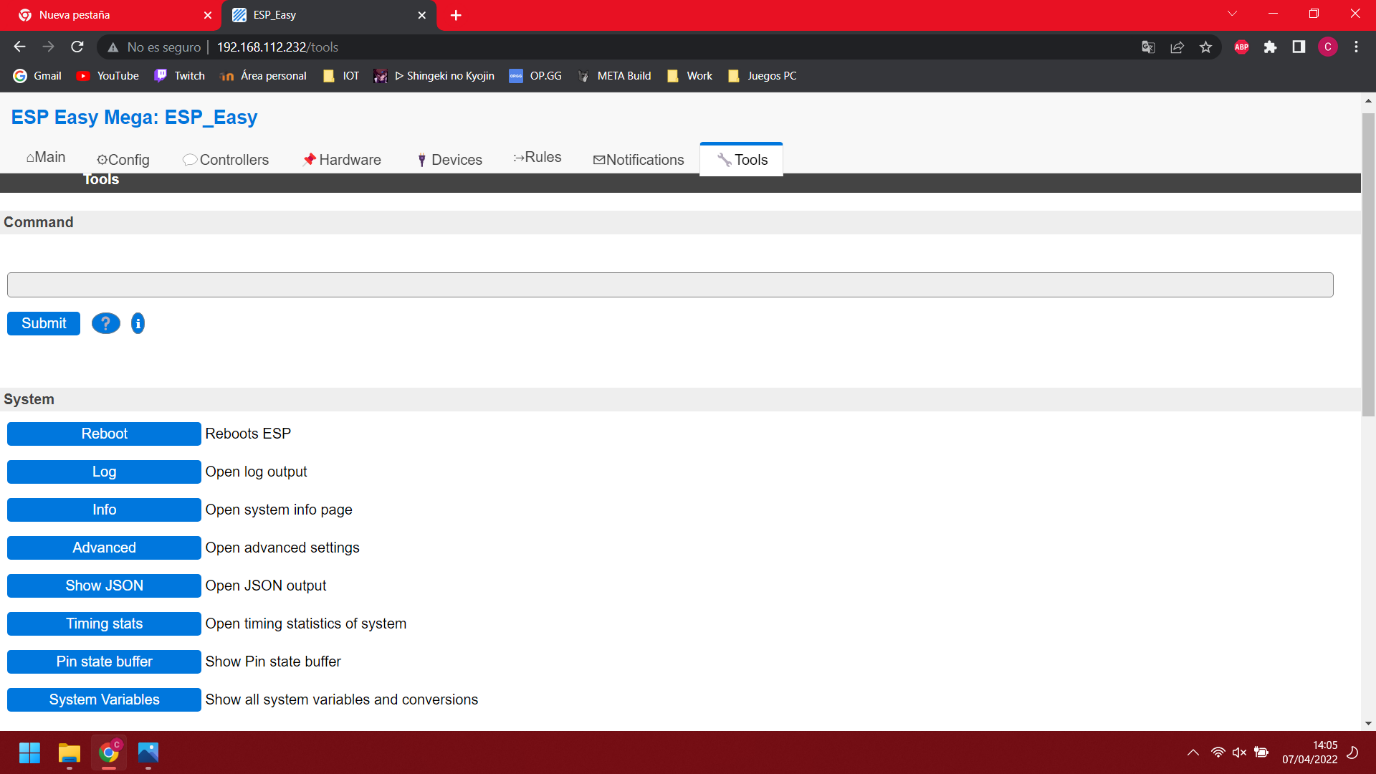
Ya lo tenemos configurado, sólo habría que marcar esta opción en cada uno de los dispositivos que queremos que envíen la información a nuestro servidor.

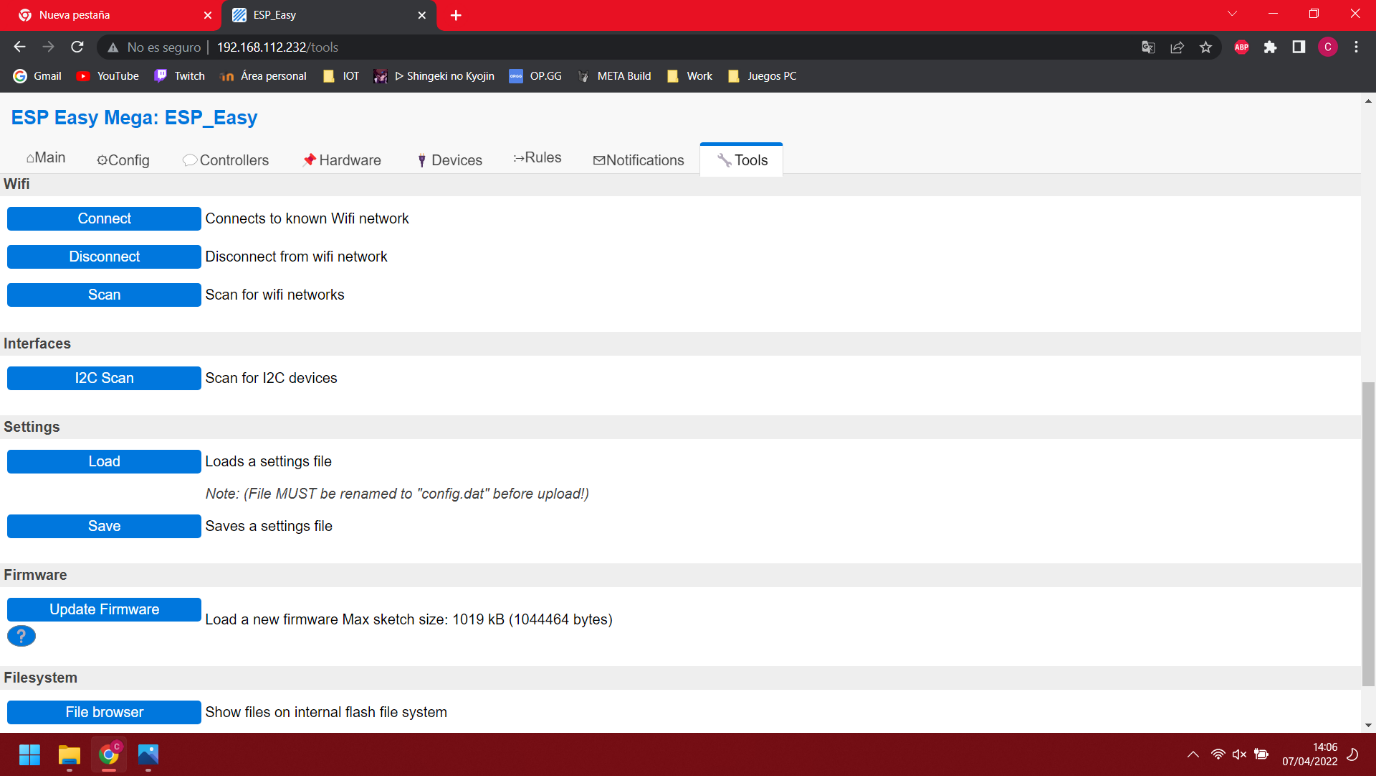


### Herramientas

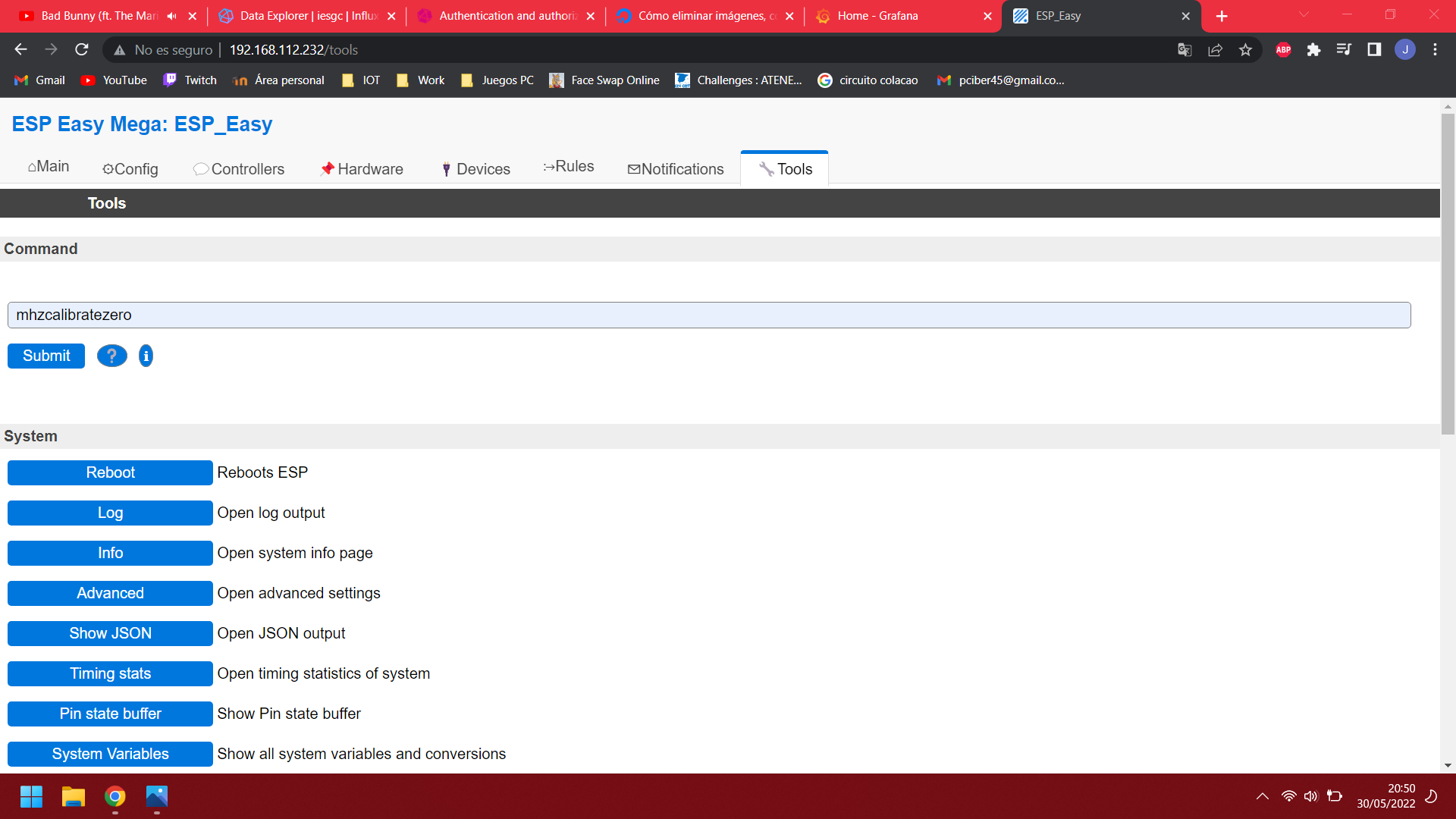
Sólo nos queda por ver la pestaña de “Tools”, ésta tiene múltiples opciones:

Entre ellas reiniciar el firmware, ver logs, obtener información, escanear wifi o dispositivos I2C como la pantalla etc.





También podemos introducir comandos para administrar nuestro entorno como por ejemplo éste para calibrar el co2.



Toda esta información podemos guardarla y cargarla con las opciones de Load y Save file.

Esto es muy útil para transportar nuestra configuración a otras instalaciones o para replicar la configuración en otro entorno.

# 

# **Software (Servidor)**

## El proyecto actualmente se encuentra montado sobre el servidor de nuestro centro, pero antes de hacerlo aquí, usamos AWS Educate como sitio de prueba y error, hasta que una vez conseguimos que todo funcionase correctamente y con fiabilidad ya lo cambiamos y lo subimos al servidor del instituto.

## Creación

Una vez la placa está completada toca la parte del servidor, este entorno está montado sobre el servidor del instituto en una máquina virtual de vmware, es un ubuntu server 20.04 con 2 GB de ram y dos núcleos y 20 GB de almacenamiento, no hace falta más ya que en sí el sistema de monitorización y de gráficos no consume muchos recursos.

Los puertos usados son los siguientes:

* 22. Este puerto es el usado por defecto para conectarte mediante el protocolo ssh.
* 25. Puerto usado por SMTP, usado para enviar emails en grafana.
* 80. Puerto usado por el protocolo HTTP, usado por la página que muestra en tiempo real los datos recogidos de la placa.
* 443. Puerto usado por HTTPS.
* 587. Puerto usado por SMTP, usado para enviar emails en grafana.
* 1883. Puerto usado para recibir datos a través de mosquitto.
* 3000. Puerto usado por grafana.
* 3306. Puerto usado para la base de datos.
* 8086. Puerto usado por influxDB.

Una vez terminado esto lanzamos la instancia, luego nos conectamos a ella y lanzamos un “apt update” para actualizar los repositorios.

Ya podemos iniciar la instalación de los programas necesarios.

## Mosquitto

Mosquitto es la herramienta que se encarga de recibir los datos de la placa. Este servicio utiliza el puerto 1883, que lo abrimos anteriormente.

### Instalación

Lo primero es lo primero, por lo que hay que instalarlo con el comando:

sudo apt install mosquitto

Hará que funcione la parte de recoger datos, también hay que instalar el cliente de mosquitto para mostrar los datos en la página web.

sudo apt install mosquitto-clients

Ahora hay que configurar para que mosquitto se inicie automáticamente:

sudo systemctl enable mosquitto.service

Hecho todo esto, el servicio ya estará funcionando, pero hay que tocar un archivo de la configuración para que pueda recibir datos.

### Configuración

El archivo de configuración se encuentra en la carpeta /etc/mosquitto.

Dentro de esta carpeta está el archivo de configuración principal y los certificados que permiten utilizar una conexión segura.

Modificamos el archivo mosquitto.conf.

Hay que añadir las siguientes líneas.

* listener 1883 Con esto se indica el puerto que utiliza.
* allow\_anonymous true Indicamos que puede recibir datos de cualquiera.
* protocol mqtt Indicamos el protocolo que usará el puerto. MQTT es el usado para recibir datos.

Ahora tenemos que reiniciar el servicio con el comando:

sudo systemctl restart mosquitto.service

### Pruebas

Una vez terminada la configuración podemos comprobar que recibe mensajes.

El comando mosquitto\_sub podemos suscribirnos a un “topic” el cual mostrará por la consola todos los mensajes recibidos.

sudo mosquitto\_sub -t “iesgrancapitan/112/#” es el comando que usaremos para mostrar los datos.



También podemos publicar datos manualmente con el comando

sudo mosquitto\_pub -h localhost -t “iesgrancapitan/112/temperatura” -m “20”

Sería algo así.

Enviamos el mensaje



Y con la suscripción vemos el mensaje.



## InfluxDB

Esta es la base de datos donde se albergarán los datos recogidos de mosquitto y luego grafana accederá a estos para el sistema de alertas.

### Instalación

Hay que tener en cuenta una cosa antes de instalar y es instalar la última versión ya que versiones antiguas no disponen de su propia lenguaje de consultas, en versiones anteriores usaba un lenguaje similar a SQl mientras en las últimas versiones puedes usar este lenguaje o su propio lenguaje, que es mucho mejor e intuitivo ya que puedes usar una herramienta similar al phpmyadmin.

Dicho lo anterior procedamos a la instalación.

La instalación se tiene que realizar sobre docker, por lo que hay que instalar docker con los siguientes comandos.

sudo apt install docker

Y también hay que instalar docker-compose para lanzar imágenes.

sudo apt install docker-compose

Una vez instalado tenemos que crear un archivo que se llamará docker-compose.yml, dentro de él añadiremos los parámetros necesarios para la instalación de influxdb. Esto es lo que hay que añadir.

influxdb:

image: influxdb:2.1.1

ports:

- 8086:8086

volumes:

- influxdb\_data:/var/lib/influxdb

environment:

- INFLUXDB\_DB=sensores

- INFLUXDB\_ADMIN\_USER=iesgc

- INFLUXDB\_ADMIN\_PASSWORD=proyecto

- INFLUXDB\_HTTP\_AUTH\_ENABLED=true

Ahora usamos el comando sudo docker-compose up -d que leerá el archivo y procederá a descargar la imagen 2.1.1 de influxdb y utilizará el puerto 8086, además de crear un usuario administrador.

Una vez terminado este proceso tenemos que usar el comando sudo docker ps

para mostrar todas las imágenes docker y así poder saber el ID de la de influxdb y así poder lanzarla. En mi caso la ID era “37b0f78a6b25”, por lo que para iniciarla hay que usar el siguiente comando:

sudo docker-compose up -d

Ahora podemos usar este comando para usar la imagen:

sudo docker exec -it 37b0f78a6b25 /bin/bash

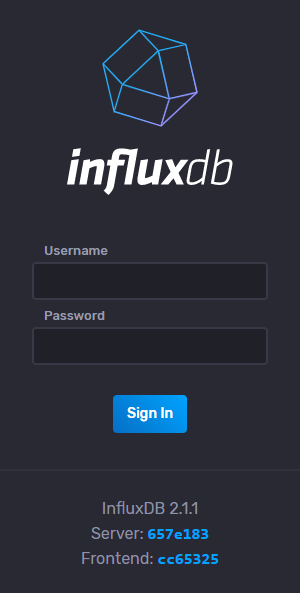
Esto abrirá una consola tal que así:



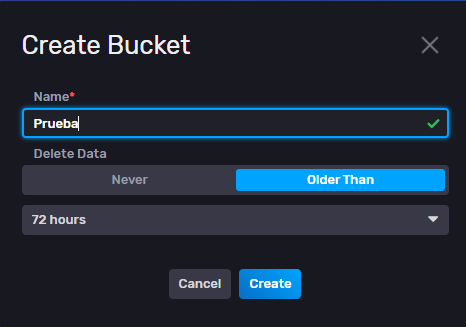
Donde podremos acceder a la configuración y administración por comandos, aunque lo recomendable es por su interfaz web.

### Configuración

### Hecho lo anterior nos vamos a un navegador web y ponemos ip\_servidor:8086 para acceder a la interfaz web, nos saldrá una página donde nos pedirá la información de logeo de la cuenta que se creó cuando se lanzó el compose up.



Entramos y en la parte izquierda vamos a data y lo primero de todo es crear un bucket, que es donde se recogen los datos.



Es recomendable habilitar el borrado de datos tras unos días para que no se llene de el almacenamiento. Ahora hay que instalar telegraf.

### Telegraf

Telegraf es la herramienta que se utiliza para conectar los datos obtenidos a través de mosquitto a influxdb y guardarlos ahí.

Como siempre lo primero es instalar el programa.

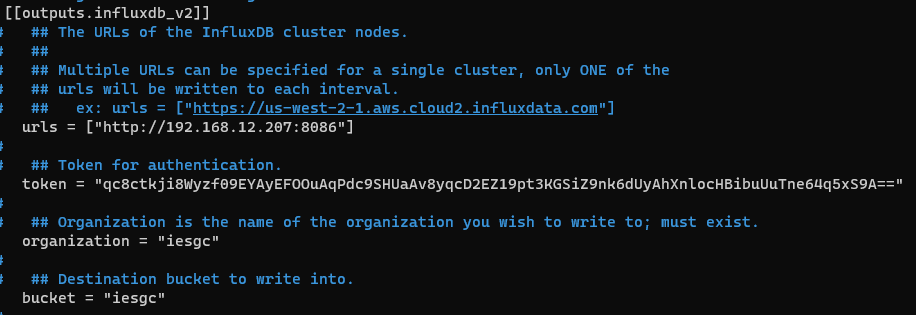
wget -qO- https://repos.influxdata.com/influxdb.key | sudo tee /etc/apt/trusted.gpg.d/influxdb.asc >/dev/null

source /etc/os-release

echo "deb https://repos.influxdata.com/${ID} ${VERSION\_CODENAME} stable" | sudo tee /etc/apt/sources.list.d/influxdb.list

sudo apt-get update && sudo apt-get install telegraf

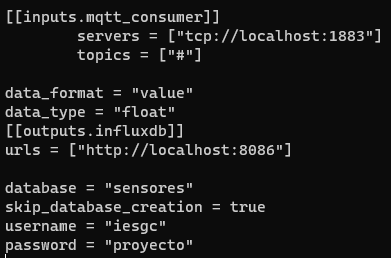
Una vez instalado nos vamos al archivo de configuración /etc/telegraf/telegraf.conf y lo dejamos tal que así.



Tenemos que indicar la url de influxdb junto a su puerto, poner el token que se obtiene en la parte de data de influxdb, api tokens y seleccionas el nombre del usuario que se creó ahí te saldrá una cadena de valores y la pegamos aquí.

Luego ponemos en organization y bucket el nombre del bucket que creamos.

Luego al fondo del archivo añadimos lo siguiente.

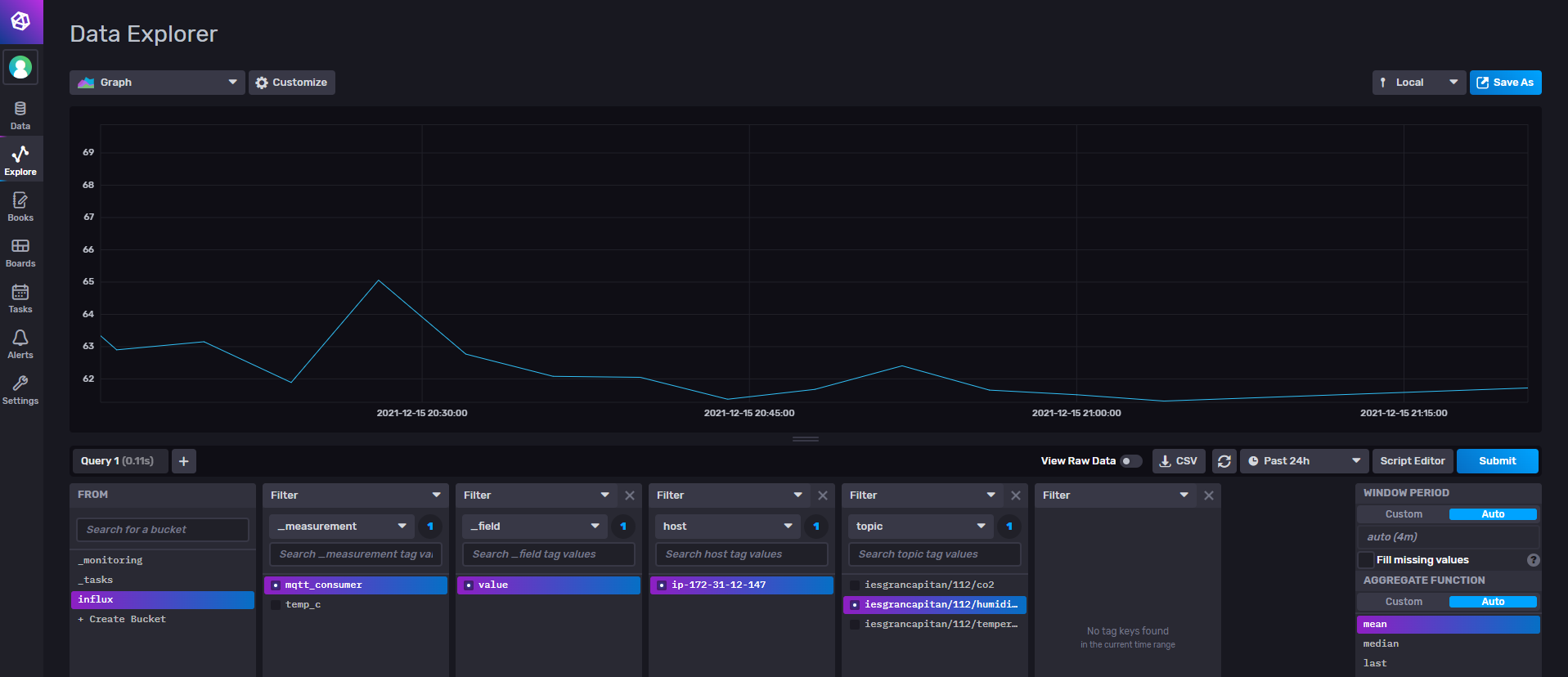


Con todo esto añadido recibirá los datos y luego los pasará a influxdb donde serán guardados. Reiniciamos el servicio y la parte de telegraf está terminada.

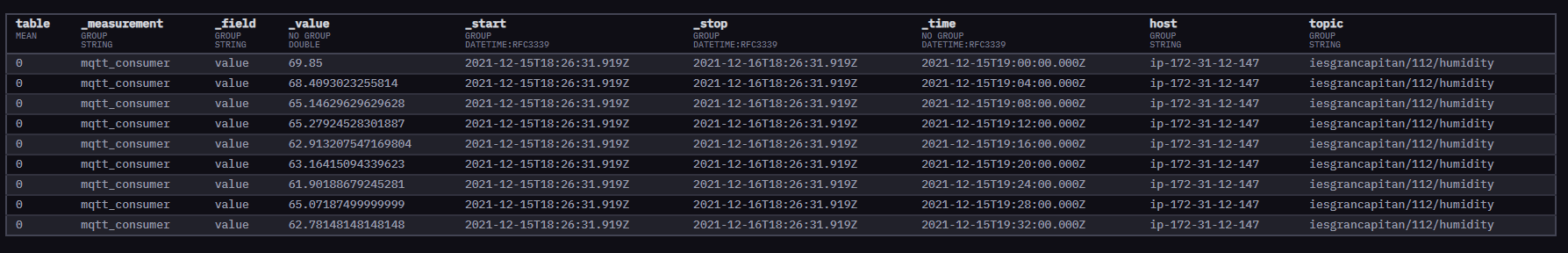
### Datos

Si todo está configurado como lo dicho anteriormente debería de guardarse los datos ya en influxdb. Para acceder a ellos, en la parte izquierda nos vamos a explore y ahí seleccionamos el bucket, que en mi caso es influx, y podemos elegir que ver, si el co2, temperatura, humedad o cualquier combinación de las tres.

En este ejemplo enseñaré la humedad que había el día 15 de las 20:15 a las 21:20.



También podemos ver los valores en una tabla.



Y también de muchas otras formas.

## Grafana

Esta es la herramienta encargada de leer los datos de influxdb y crear gráficos donde puedes configurar alertas para que envíe emails a las direcciones que uno desee.

### Instalación

Lo primero de todo es instalar grafana. Para esto hay que instalar unos paquetes básicos antes.

sudo apt-get install -y apt-transport-https

sudo apt-get install -y software-properties-common wget

Ahora obtenemos el repositorio de grafana.

wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add

echo "deb https://packages.grafana.com/enterprise/deb stable main" | sudo tee -a /etc/apt/sources.list.d/grafana.list

echo "deb https://packages.grafana.com/oss/deb stable main" | sudo tee -a /etc/apt/sources.list.d/grafana.list

Ahora actualizamos los repositorios y ya podemos instalar grafana.

sudo apt-get update

sudo apt-get install grafana

Hecho esto ya se instalará e iniciará automáticamente grafana, una cosa a tener en cuenta es que para reiniciar, iniciar, parar etc… hay que usar /etc/init.d/grafana-server

### Configuración de gráficos

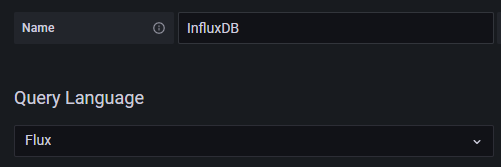
Una vez instalado nos vamos a un navegador y escribimos la siguiente URL ip-servidor:3000.

Una vez dentro nos pedirá que ingresemos, el usuario y contraseña inicial es admin admin, una vez entremos nos pedirá una contraseña nueva.

Ya dentro por fin, lo primero que tenemos que hacer es añadir un origen de datos.

Le damos a añadir data source y seleccionamos influxDB.

Le damos un nombre y MUY importante, seleccionamos en query language “flux” ya que es como estamos haciendo las consultas y está configurado.



Ahora ponemos la ip ( sirve con <http://localhost:8086> )



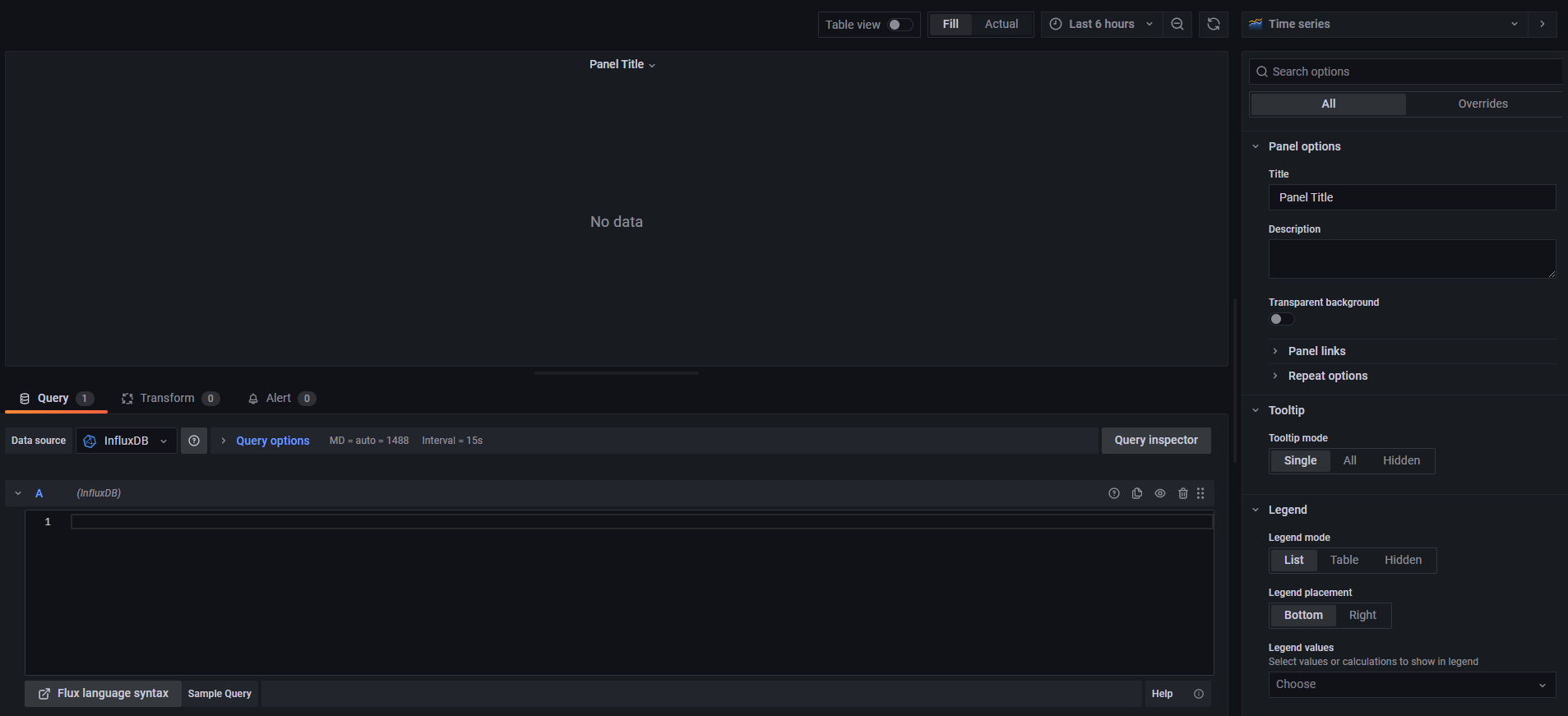
Y bajamos al final y donde pone InfluxDB Details ponemos la organización, el token que se obtiene de la misma forma que con telegraf y el bucket.



Le damos a save and test y si todo va bien nos deberá de decir que se ha encontrado 3 buckets.

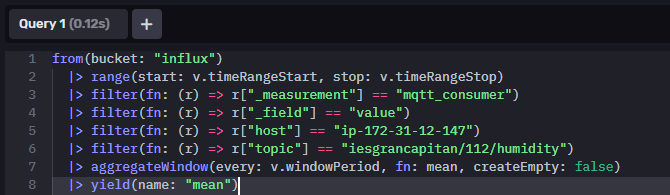
Ahora ya podemos crear los gráficos donde se crearán las alertas.

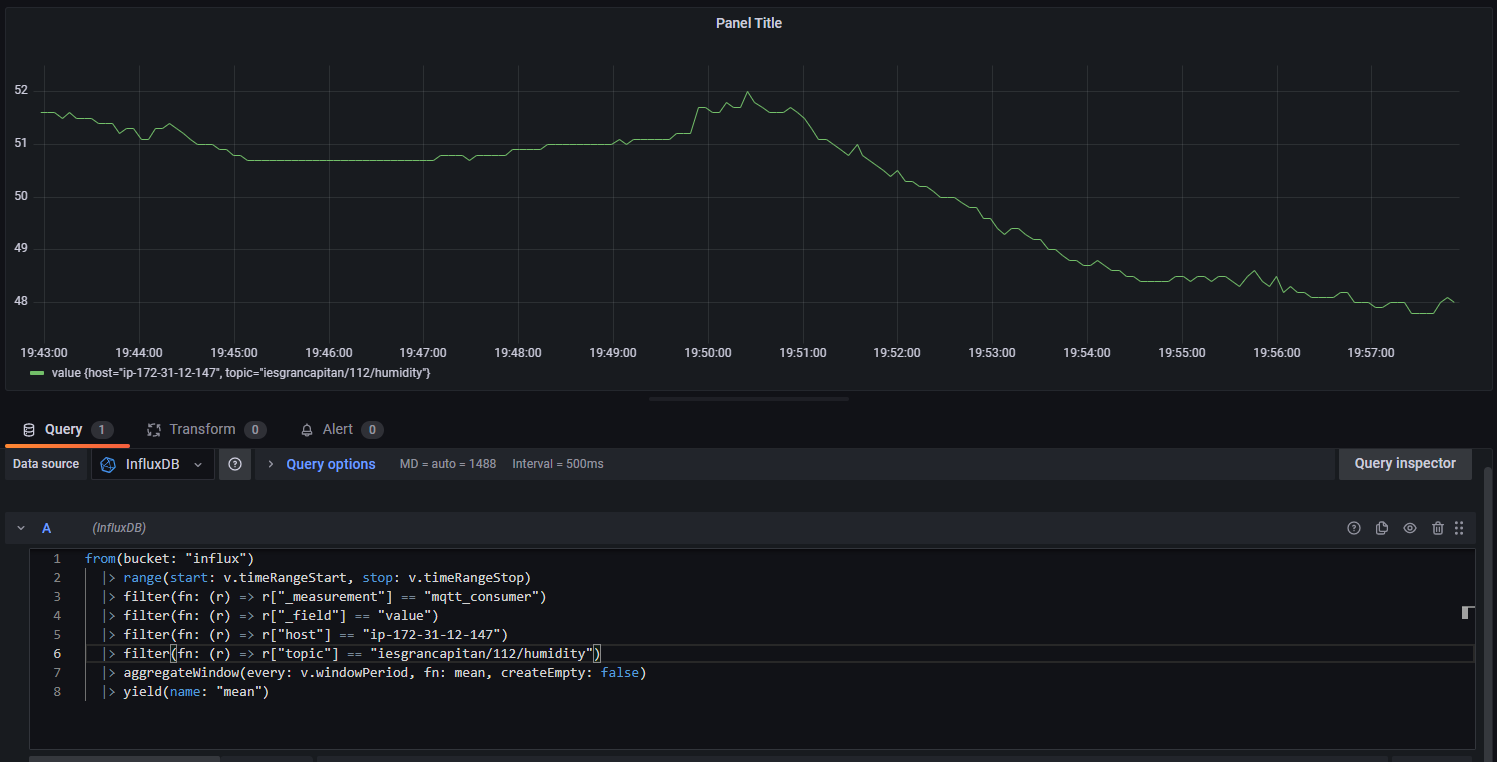
Para crear un gráfico solo hay que darle al más en la parte izquierda y luego a new panel, nos saldrá esto.



Aquí podremos escribir la consulta que usará para leer los datos.

Para obtenerla nos vamos a la web de influxdb y le damos a explore, seleccionamos el parámetro que queremos ver, en este caso va a ser humedad y le damos a submit y luego a script editor para ver la consulta.

Esto tendrá que salir, lo copiamos y lo pegamos en el editor de grafana y se tendrá que ver un gráfico.



Una vez hecho esto lo repetimos con el co2 y la temperatura. Nos quedará una dashboard tal que así.



### Alertas

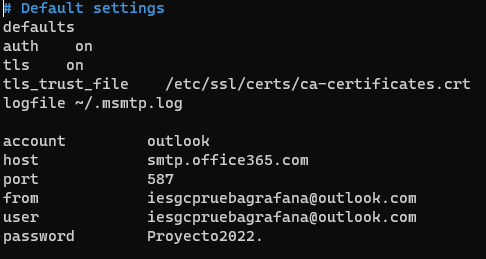
Las alertas están pensadas para que cada vez que el último valor se salga de un rango de valores o supere un valor máximo envíe un email a unos receptores previamente introducidos. También envía emails cuando hay un error, es decir no ha recibido ningún dato y cuando vuelve al funcionamiento normal y los valores vuelven al rango seguro.

Lo primero de todo es instalar ssmtp en el servidor, que es el gestor que se encargará de enviar los correos.

Su instalación es la siguiente:

sudo apt install msmtp

Luego creamos el archivo de configuración de msmtp al que llamaremos msmtprc e introducimos las siguientes líneas de código que también incluyen los datos del correo.



Y ahora le damos permisos 600 con chmod para que funcione correctamente

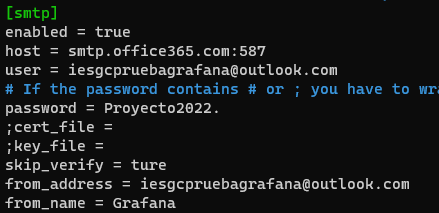
Y ahora para comprobar que todo está bien podemos enviar un correo de prueba con el siguiente comando.

echo -e "\nHOLA." | msmtp -a outlook a19romefr@iesgrancapitan.org



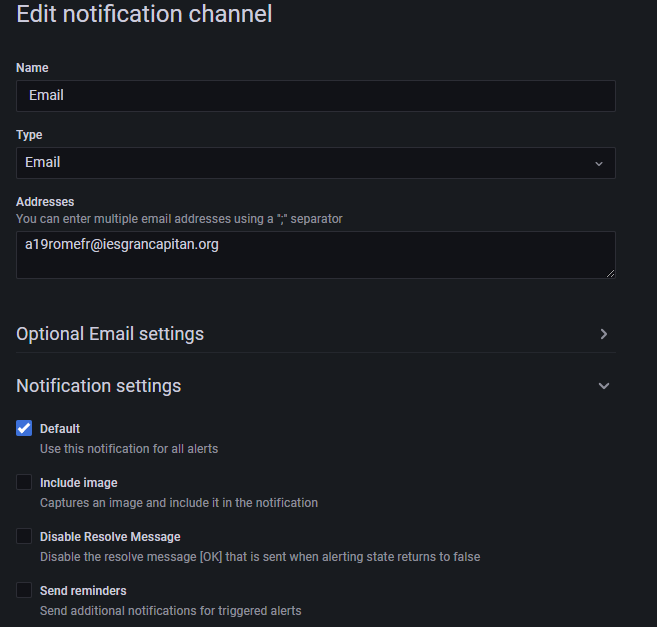
Hecho esto ya tenemos configurado el gestor, ahora tenemos que configurar grafana, nos vamos a su archivo de configuración /etc/grafana/grafana.ini

Una vez dentro nos vamos al apartado smtp e introducimos los datos del correo que usaremos.

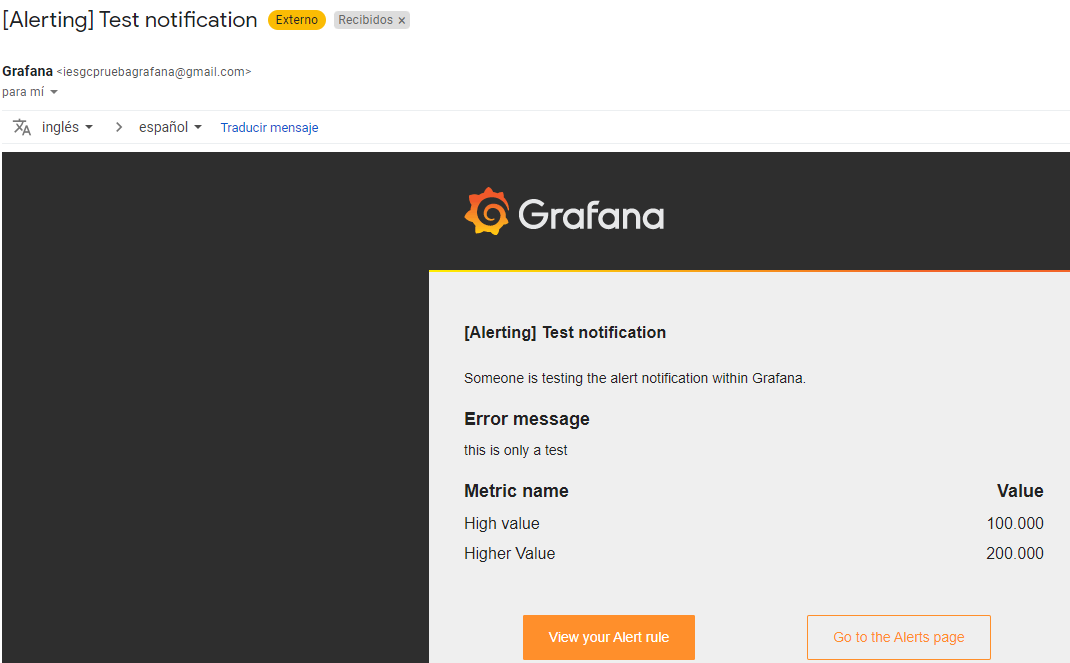


A continuación reiniciamos grafana mediante la orden /etc/init.d/grafana-server restart

Con esto ya grafana puede enviar correos. Pero antes hay que ir a la web de grafana y en la parte izquierda le damos a la campana, dentro de ahí seleccionamos notification channels y añadimos uno nuevo, le damos un nombre y seleccionamos email, también se puede elegir muchos tipos más. Y ponemos las direcciones de correo a las que queremos que lleguen los emails. Además le damos a notification settings y seleccionamos default para que sea por defecto siempre y nos ahorre tiempo a la hora de elegir.

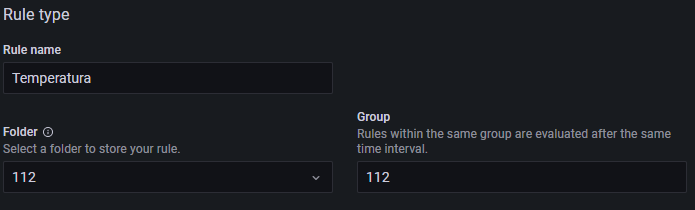


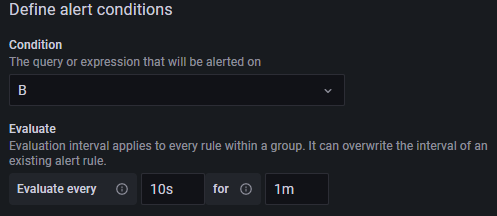
Si le damos a test, nos enviará un correo de prueba, si todo está bien nos debería de llegar un correo a la direcciones especificadas.



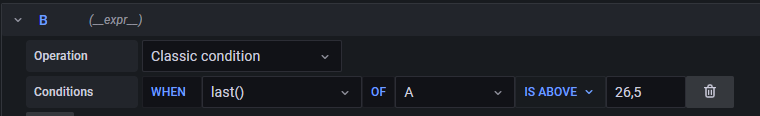
Ahora nos vamos al dashboard y creamos las alertas para cada gráfico.

Por ejemplo seleccionamos temperatura y le damos a alert, hay dentro le damos a crear una, le damos un nombre y un periodo de revisión de datos, la placa envía datos cada 10 segundos, por lo que he puesto que lo revise cada 10 segundos y si en un 1 minuto no ha vuelto a un dato dentro del rango envía la alerta.

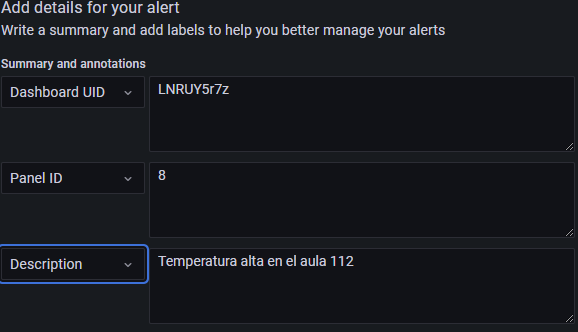




Luego en conditions seleccionamos last() en el valor que selecciona y “is outside range” a esto le ponemos que si supera 26,5 salte la alerta



Y al final en la notificación nos selecciona automáticamente la forma de envío ya que lo especificamos anteriormente y podemos escribir un mensaje.

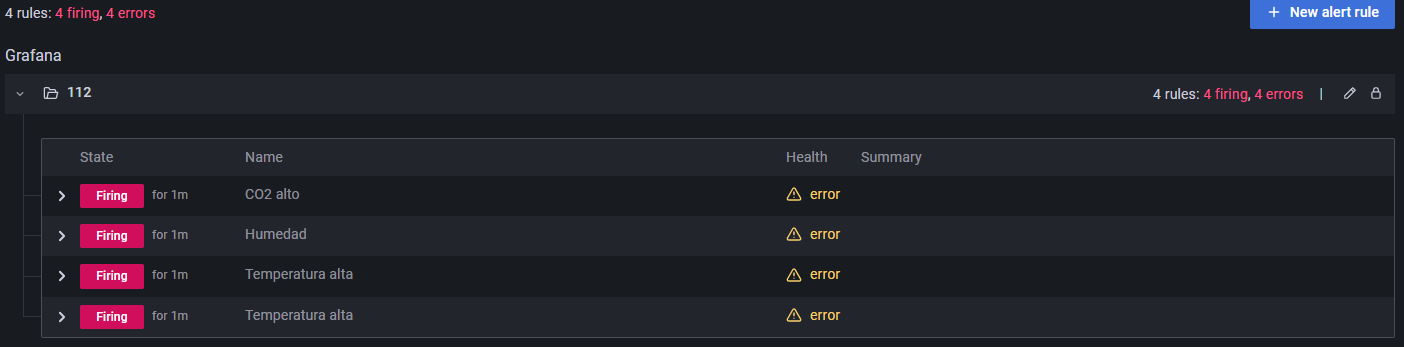


Una cosa a tener en cuenta es que las alertas son planas, es decir no puede identificar si la temperatura está por encima o por debajo, pero envía el valor por correo, así que la persona que lo lea puede saber si el valor está por debajo o por encima o si hay un error en la placa.

Ahora repetimos el mismo proceso en cada gráfico y ponemos en el de co2 que el valor sea superior a 1000 y en humedad que los valores estén fuera del rango de 29.5 y 70.5

Guardamos los cambios y ya estarían las alertas completadas

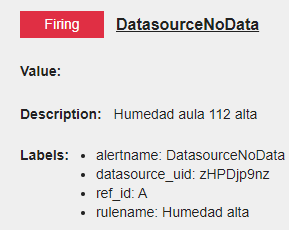
Para comprobar que todo ha ido bien nos podemos ir a la campana de la izquierda, donde saldrán todas las alertas que hay y grupos de alertas y el estado.



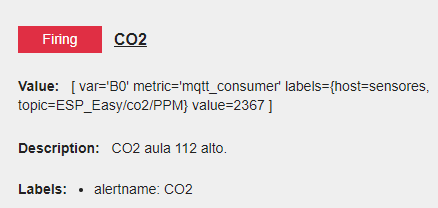
Si ahora en cualquier momento hay un error en la placa, es decir no envía, o se dan las condiciones para que se envíen alertas o vuelve a un estado normal nos llegará un correo diciéndonos si no hay datos, si está ok o si hay un aviso.



Este es un correo que nos indica que se ha arreglado la incidencia.



Correo de que no hay datos.



Correo de que los datos superan el límite establecido.

# Bibliografía

<https://docs.influxdata.com/influxdb/v2.1/>

<https://techexpert.tips/es/grafana-es/configuracion-de-la-notificacion-por-correo-electronico-de-grafana/>

<https://pypi.org/project/paho-mqtt/>

<https://www.youtube.com/watch?v=dk6895EwVbM&ab_channel=SteveCope>

https://emariete.com/medidor-casero-co2/

<https://www.letscontrolit.com/wiki/index.php?title=ESP_Easy_web_interface>

<https://github.com/josejuansanchez/iot-demo>

https://josejuansanchez.org/teaching/2021/02/18/iot-dashboard-sensores.html