Material:

* Raspberry Pi y sus periféricos (cable HDMI, teclado y ratón, cable de alimentación)
* Tarjeta SD (64 Gb)
* 2 ESP8266
* 1 DS18B20
* 1 BME280
* Protoboard
* Cables dupont
* Resistencia 4.7Ω

Software

* Raspbian <https://www.raspberrypi.org/downloads/raspbian/>
* Webmin <http://www.webmin.com/>
* InfluxDB <https://www.influxdata.com/>
* Grafana <https://grafana.com/>
* Blynk <https://blynk.io/>
* NODE-RED <https://nodered.org/>
* Mosquitto <https://mosquitto.org/>
* Arduino IDE <https://www.arduino.cc/en/Main/Software>
* Balena Etcher <https://www.balena.io/etcher>
* Win32 Image Writer <https://sourceforge.net/projects/win32diskimager/>
* SD Card Formatter <https://www.sdcard.org/downloads/formatter/>

## 1.- Presentación del proyecto

Con la publicación de la versión “Raspbian Buster” me propuse actualizar mi versión “Stretch”, después de revisar varias guías de actualización y varios intentos fallidos decidí realizar una instalación en blanco, es decir, desde cero.

Llevaba tiempo dándole vueltas a realizar un proyecto con una base de datos, que prometía ser la adecuada para proyectos de registro y análisis de estos datos (temperatura, humedad…) todo tipo de dato basado en un ***timestamp,*** INFLUXDB.

Con estos requisitos (Buster e InfluxDB) me plantee un proyecto de registro de temperatura, humedad, presión atmosférica, en diferentes habitaciones de la casa, monitorear las condiciones ambientales.

Para ello, iba a utilizar un sensor BME280 y otro DS18B20, montados en sendas **ESP8266**, enviando, a través de wifi, mediante **MQTT** a **Raspberry Pi**, aquí almacenar los datos en la BBDD de **InfluxDB** para después analizar los datos recogidos mediante **Grafana** y su ***Dashboard.***

## 2.- Instalación del sistema operativo

Nuestro primer paso será la instalación del sistema operativo en la tarjeta SD para poner a funcionar la raspberry pi, como ya he comentado, el objetivo es instalar la versión **Buster** de **Raspbian.**

Descargamos la imagen **ISO** del repositorio de la fundación Raspberry Pi, podemos ir mediante este [enlace](https://www.raspberrypi.org/downloads/raspbian/).

Una vez tenemos descargada la **ISO**, procedemos a instalarla en la SD mediante el programa **Balena Etcher.**

Ya vimos este proceso en el tutorial anterior: [**Servidor Web con Raspberry Pi y Arduino (I)**](https://www.punto3blog.ml/2019/01/servidor-web-con-raspberry-pi-y-arduino-1.html)**.**

Ponemos en nuestro ordenador la tarjeta SD mediante el lector de tarjetas, procedemos a dar formato con **SD Card Formatter**, ejecutamos **Balena Etcher**, nos pide seleccionar la imagen que vamos a copiar, nos pide el destino (nuestra SD) y procedemos a “***flashear***”.

Esperamos a que finalice el proceso y cerramos el programa.

Expulsamos la SD y listo.

Ahora introducimos la SD en la ranura de nuestra placa Raspberry Pi, conectamos cables, teclado, ratón… y damos alimentación.

La primera vez que arranca, nos solicita la contraseña del usuario por defecto "pi", el lenguaje del teclado, ubicación para el reloj, la configuración para la conexión WiFi...

una vez hemos terminado de configurar, comprobará las actualizaciones del sistema y nos pedirá que reiniciemos.

Mediante la opción del menú -> Preferencias -> Configuración Raspberry Pi -> Interfaces, o con el comando "raspi-config" podremos activar los diferentes servicios disponibles, recomiendo activar "SSH" y "VNC Server", así podremos acceder de forma remota a nuestra Raspberry.

Como siempre, es muy recomendable realizar copia de nuestra SD cada vez que afrontemos un proceso que suponga cambios en el sistema.

Creo que este es un buen momento para realizar una copia ya que acabamos de instalar **Raspbian** y es un buen punto de recuperación ante cualquier problema.

Para realizar la copia podemos utilizar **Win32 Image Writer.**

Seleccionamos el fichero IMG destino, si no existe lo creara, la unidad que queremos salvar, y como es la primera copia solo nos dará opción de "*Read*" Pulsamos en ella y nos realizara una copia de la tarjeta SD.

Por costumbre, suelo instalar “**Webmin**”, creo que es una buena herramienta de administrador, con ella podremos gestionar fácilmente, mediante una interface web, nuestro sistema.

Para elle, accedemos a consola, ya sea por SSH o desde el escritorio de Raspbian y tecleamos los siguientes comandos:

(Podemos verlo en [Webmin](http://www.webmin.com/deb.html))

**# wget http://prdownloads.sourceforge.net/webadmin/webmin\_1.941\_all.deb**

**# sudo dpkg --install webmin\_1.900\_all.deb**

es probable que nos solicite instalar algunas dependencias, entonces tecleamos:

**# sudo apt-get install perl libnet-ssleay-perl openssl libauthen-pam-perl libpam-runtime libio-pty-perl apt-show-versions python**

**# sudo apt --fix-broken install**

Abrimos una ventana del navegador, puede ser en local o desde otra máquina, y accedemos a la dirección:

https://localhost:10000 (desde local)

https://***ip\_de \_nuestra\_raspberry***:10000 (desde otra maquina)

nos mostrara la pantalla de *login,* accedemos con nuestro usuario de Raspberry.

Para configurar **Webmin**, tema, idioma…, accedemos al punto “***Webmin***” del menú de la parte izquierda de la pantalla.

Aquí os dejo el enlace a la documentación <https://doxfer.webmin.com/Webmin/Main_Page>

Y hasta aquí, esta primera entrada, en la siguiente, continuamos con la instalación del resto de programas en Raspberry

## 3.- Instalación de InfluxDB

Para la instalación de InfluxDB abrimos consola, bien en Raspberry bien por SSH y tecleamos los siguientes comandos:

# sudo apt-get update

# sudo ap-gett upgrade

# wget -qO- https://repos.influxdata.com/influxdb.key | sudo apt-key add -

# echo "deb https://repos.influxdata.com/debian buster stable" | sudo tee /etc/apt/sources.list.d/influxdb.list

# sudo apt update

# sudo apt install influxdb

# sudo systemctl unmask influxdb

# sudo systemctl enable influxdb

# sudo systemctl start influxdb

Con esto tendremos instalado InfluxDB, adicionalmente, si queréis probar InfluxDB, podemos instalar TELEGRAF, se trata de un ***Agente*** que recopila datos de nuestro sistema, creando una base de datos en InfluxDB.

Para instalarlo, debemos teclear en una consola:

# sudo apt-get install telegraf

# sudo nano /etc/telegraf/telegraf.conf

Y añadiremos la URL de nuestro servidor InfluxDB en la sección llamada outputs.influxdb, fijaros en la variable urls

# /etc/init.d/telegraf restart

Ahora, para poder analizar los datos recopilados por el ***agente*** TELEGRAF podemos instalar CHRONOGRAF, se trata de una interface grafica (modo web) para administrar InfluxDB y las bases de datos (en este caso TELEGRAF).

Para instalar, teclearemos:

# sudo apt-get update && sudo apt-get install chronograf

Para acceder a Chronograf teclearemos:

https://localhost:8888 (desde local)

https://***ip\_de \_nuestra\_raspberry***:8888 (desde otra maquina)

## 4.- Instalación de Grafana

El siguiente servicio que vamos a instalar es Grafana, para proceder teclearemos en una consola de comandos:

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

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

# sudo apt-get update

# sudo apt-get upgrade

# sudo apt-get install apt-transport-https

# sudo apt-get install grafana

# sudo systemctl daemon-reload

# sudo systemctl enable grafana.service

# sudo systemctl start grafana-server

Para comprobar como funciona y ya de paso, ver los datos que recopila el ***agente*** Telegraf, vamos a instalar un Dashboard, donde podemos ver todos los valores que estamos recopilando.

El Dashboard en concreto es: <https://grafana.com/grafana/dashboards/10578>

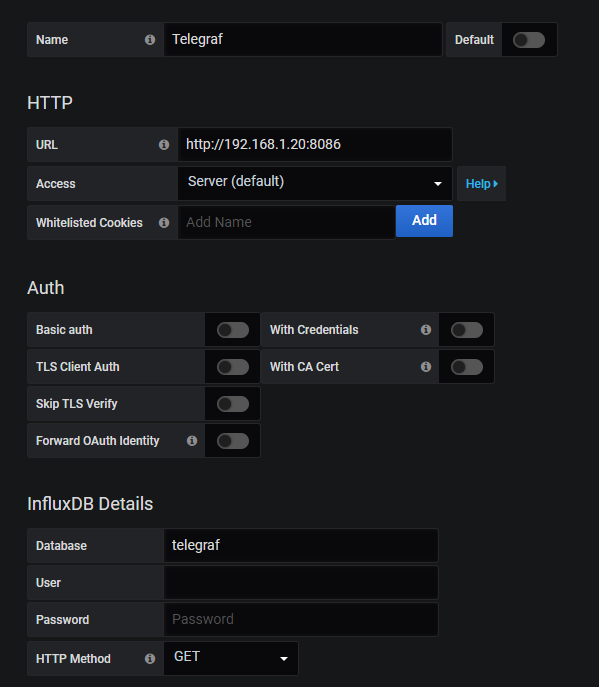
Para instalarlo, accedemos a Grafana y creamos un origen de datos para InfluxDB

Configuration -> Data Sources

Add data Source

Aquí buscamos el origen de datos para InfluxDB

Y pulsamos en Select



Debemos completar los campos correspondientes a:

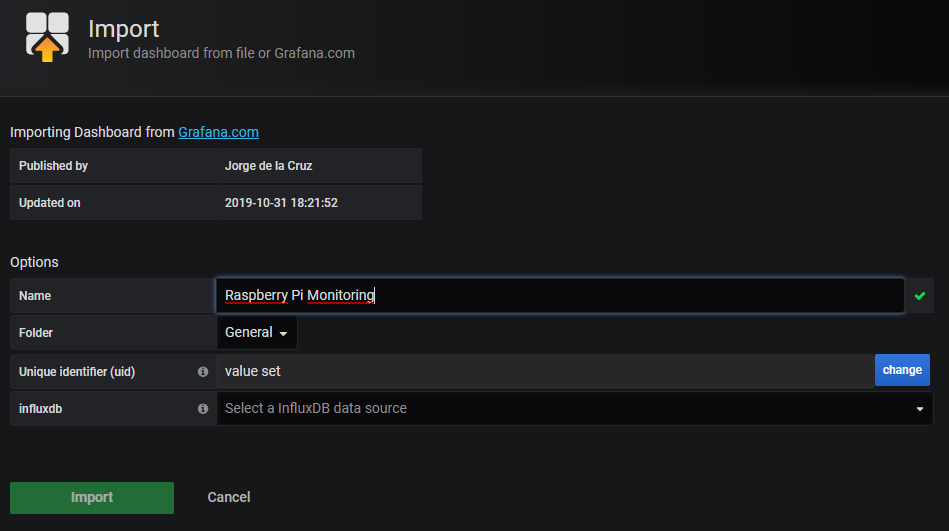
Name: descripción del origen de datos

URL: la dirección IP y el puerto de nuestro servidor InfluxDB

Database: nombre de la base de datos a la que queremos acceder

Y pulsamos en *Save & test*

Después, vamos a *Create* en el menú de **Grafana** y seleccionamos *Import*, en el campo “**Grafana.com Dashboard**” escribimos el código del Dashboard: **10578** y pulsamos en “***Load***”.



Nos pide:

Name: viene definido por defecto, podemos cambiarlo

Folder: carpeta de Grafana para guardar el Dashboard

Unique identifier (uid): debemos generarlo mediante el botón “*change*”

Influxdb: mediante el desplegable selecionamos el origen de datos que hemos creado en el paso anterior.

Pulsamos en “*Import*” para crear el Dashboard.

Por último, debemos modificar el fichero “*telegraf.conf*”

# cd /etc/telegraf

# sudo cp telegraf.conf telegra.ORIGINAL

# sudo nano /etc/telegraf/telegraf.conf

Y añadir:

#In order to monitor both Network interfaces, eth0 and wlan0, uncomment, or add the next:

[[inputs.net]]

[[inputs.netstat]]

[[inputs.file]]

files = ["/sys/class/thermal/thermal\_zone0/temp"]

name\_override = "cpu\_temperature"

data\_format = "value"

data\_type = "integer"

[[inputs.exec]]

commands = ["/opt/vc/bin/vcgencmd measure\_temp"]

name\_override = "gpu\_temperature"

data\_format = "grok"

grok\_patterns = ["%{NUMBER:value:float}"]

también tendremos que añadir el usuario telegraf al grupo de video, esto es para poder recopilar datos de la GPU.

En consola de comandos tecleamos:

# sudo usermod -G video telegraf

Ahora podemos acceder al Dashboard que hemos creado para revisar los datos que se está recopilando en **InfluDB** mediante el agente **Telegraf.**

## 5.- Instalación de Mosquitto

Para instalar Mosquitto en Raspberry Pi deberemos utilizar los siguientes comandos desde un a terminal:

# sudo wget http://repo.mosquitto.org/debian/mosquitto-repo.gpg.key

# sudo apt-key add mosquitto-repo.gpg.key

# cd /etc/apt/sources.list.d/

# sudo wget http://repo.mosquitto.org/debian/mosquitto-buster.list

# sudo apt-get update

# sudo apt-get install -y mosquitto mosquitto-clients

# sudo apt-get install -y python-pip

# sudo pip install paho-mqtt

# sudo systemctl status mosquitto

La configuración de mosquitto está el fichero /etc/mosquitto/mosquitto.conf. Recordar hacer copia del fichero antes de hacer alguna modificación

# cd /etc/mosquitto

# sudo cp mosquitto.conf mosquitto.ORIGINAL

Y editamos el fichero de configuración mediante:

# sudo nano mosquitto.conf

Añadimos (o remplazamos)

# Place your local configuration in /etc/mosquitto/conf.d/

#

# A full description of the configuration file is at

# /usr/share/doc/mosquitto/examples/mosquitto.conf.example

pid\_file /var/run/mosquitto.pid

persistence true

persistence\_location /var/lib/mosquitto/

log\_dest file /var/log/mosquitto/mosquitto.log

include\_dir /etc/mosquitto/conf.d

# Save all log in file

log\_type all

log\_timestamp true

allow\_anonymous false

password\_file /etc/mosquitto/passwd

port 1883

listener 9001

protocol websockets

# sudo systemctl restart mosquitto

# sudo systemctl status mosquitto

Para comprobar que está bien instalado abrimos dos terminales y en uno de ellos teclea:

# mosquitto\_sub -v -t "test/topic"

en el otro terminal:

# mosquitto\_pub -t "test/topic" -m "Hello World!"

Para configurar la seguridad en Mosquitto, definiremos un USUARIO y su contraseña mediante:

# sudo mosquitto\_passwd -c /etc/mosquitto/passwd USUARIO

Nos solicita la contraseña para este usuario que hemos definido, será el que, mas tarde, utilizaremos para hacer las conexiones en NODE-RED

Para publicar y suscribirse con usuario y contraseña usar en dos terminales:

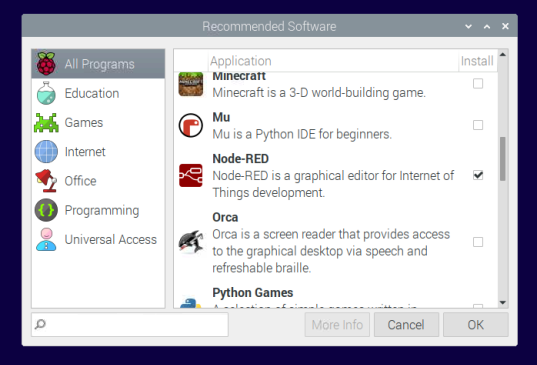
# mosquitto\_pub -d -t “test” -m “hola\_mundo” -u “USUARIO” -P "CONTRASEÑA"

# mosquitto\_sub -d -t “test” -u “USUARIO” -P “CONTRASEÑA”

## 6.- Instalación de NODE-RED

Para la instalación de NODE-RED en Raspberry Pi utilizaremos la utilidad de Software recomendado.

Accedemos al escritorio, y en el menú pulsamos “*preferencias”,* en este menú, *“Software recomendado”*

**

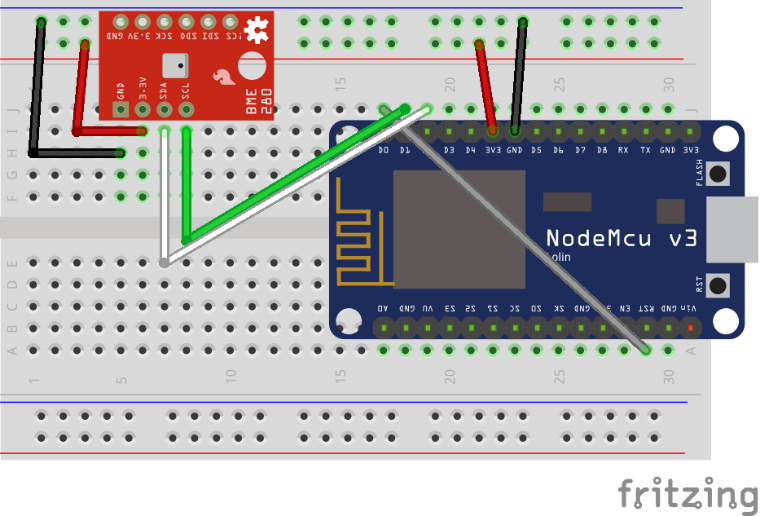
Aquí marcamos NODE-RED y pulsamos en el botón “*OK*”, esto realiza la instalación.

## 7.- Instalación de ESP8266 y sensores

Comenzamos con la instalación de las ESP8266 y los sensores, vamos a instalar un sensor BME280 y otro DS18B20, cada uno ira conectado a sendas tarjetas **ESP8266**, estas comunicarán mediante Wifi con Raspberry Pi donde serán, por una parte, almacenados en la base de datos de **InfluxDB**, de aquí sacaremos los datos para mostrar en **Grafana**, por otra parte, mostraremos en un dashboard de NODE-RED, y enviaremos a la plataforma de **Blynk** para poder acceder a ellos de forma remota.

### Sensor BME280

El esquema de conexión para el sensor BME280 es el siguiente:

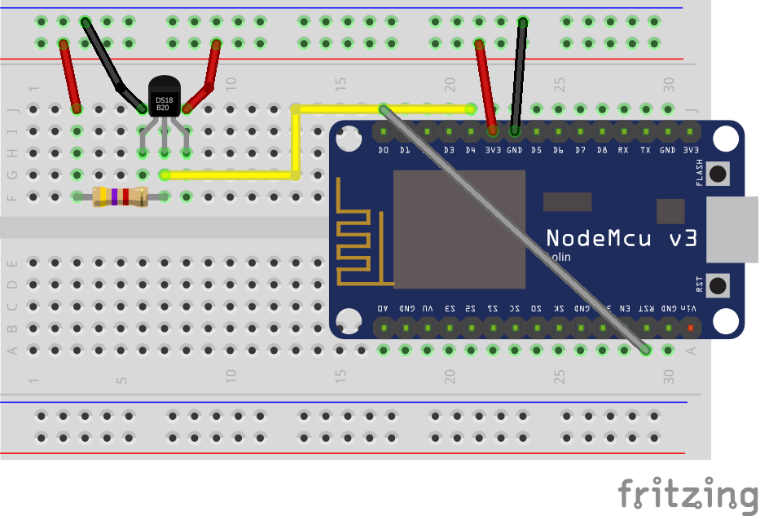


La comunicación de este sensor se realiza mediante **I2C**, conectamos **SDA** y **SCL** a los pines **SDA (D2)** y **SCL (D1)** de la placa.

También conectaremos un cable desde el pin **D0** al pin **RST** de la placa, con esto logramos que al poner en “*Deep Sleep*” podamos despertarla cada intervalo que definamos de “*sueño*”.

### Sensor DS18B20

El esquema de conexión para el sensor DS18B20 es el siguiente:



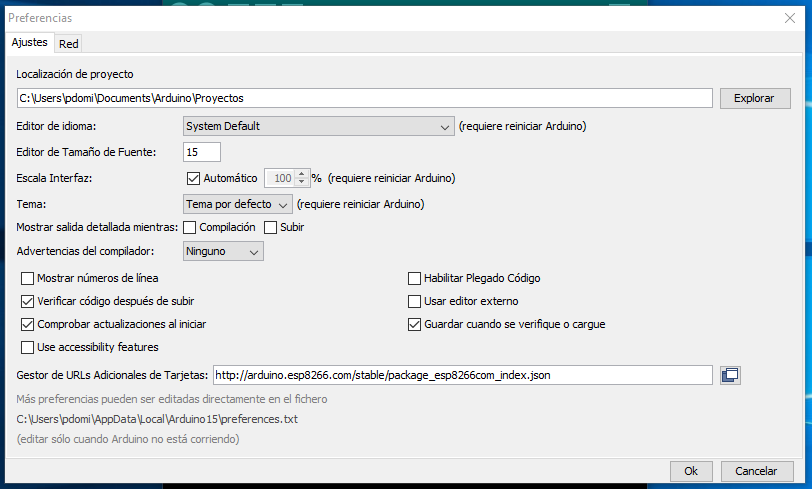
En este caso, el sensor comunica mediante *OneWire*, para ello conectaremos el pin central del sensor al pin **D4** de la placa, además, en el pin de datos del sensor pondremos una resistencia de **4.7Ω** a voltaje (**3v3**).

También conectaremos un cable desde el pin **D0** al pin **RST** de la placa, con esto logramos que al poner en “*Deep Sleep*” podamos despertarla cada intervalo que definamos de “*sueño*”.

### Sketch en Arduino IDE

Para poder comunicar, y así subir los sketches a la placa ESP8266, es necesario configurar Arduino IDE.

Para ello, ejecutamos Arduino IDE, accedemos a “*Archivo*”, “*Preferencias*”



En el campo “*Gestor de URLs adicionales de tarjetas*” añadimos: [**http://arduino.esp8266.com/stable/package\_esp8266com\_index.json**](http://arduino.esp8266.com/stable/package_esp8266com_index.json) y pulsamos en “OK”

Ahora debemos ir a “*Herramientas*” “Placa” “*Gestor de tarjetas*” y aquí instalar “*Esp8266 By ESP8266 Community*”.

Con esto tendremos instalada y disponible la Placa “NodeMCU (ESP-12E Module) que es la que utilizaremos.

Por otra parte, necesitaremos instalar varias librerías, en el caso del sketch para BME280:

* ESP8266WiFi
* PubSubClient
* Adafruit Unified Sensor
* Adafruit BME280 Library
* Streaming

En el caso del sketch para DS18B20:

* ESP8266WiFi
* PubSubClient
* OneWire
* DallasTemperature
* Streaming

Una vez incluidas las librerías, abrimos los sketches, podemos descárgalos desde [github](https://github.com/Pedroda0608/MQTT-NodeRED-InfluxDB-Grafana-Blynk), en la carpeta “ESP8266”.

Tendremos que rellenar las variables:

**Ssid**: Nombre de nuestra wifi

**Password**: contraseña de conexión de wifi

**mqtt\_server**: dirección IP de nuestra Raspberry

**mqtt\_username**: usuario de **mosquitto** (lo dimos de alta durante la instalación de **mosquitto**)

**mqtt\_password**: contraseña del usuario de **mosquitto**

También podremos definir el valor para “**SLEEP\_DELAY\_IN\_SECONDS**” está definido en 60 segundos, se trata del tiempo que permanecerá en modo “sueño” nuestra ESP8266, pasado este tiempo, despierta, comunica, envía y vuelve a modo “sueño”.

Conectamos, mediante un cable USB, la placa ESP8266 y subimos el sketch.

Podemos comprobar con el monitor serie el proceso de conexión, datos y envío.

## 8.- Configuración NODE-RED

### Preparación

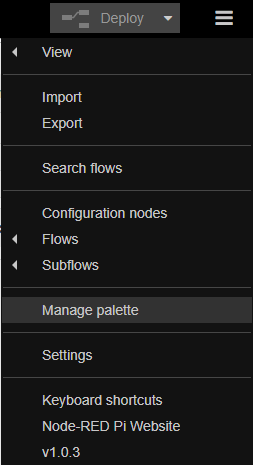
Vamos con la parametrización de NODE-RED, subiremos y configuraremos los “*Flows*” necesarios para recibir mediante MQTT los datos, mostrarlos en el Dashboard de NODE\_RED y almacenarlos en InfluxDB.

Vamos a necesitar importar lo que NODE-RED llama “*palette*”, no es más que grupos de nodos.

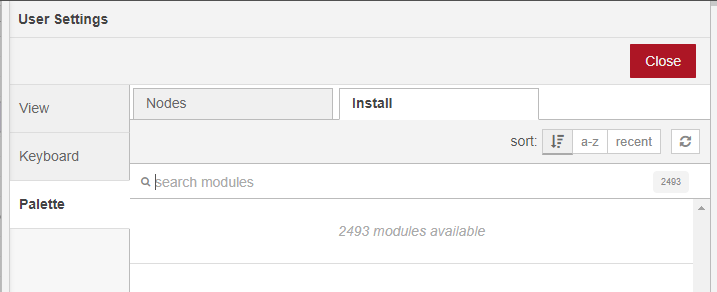
Las “*palette*” que importaremos son:

* node-red-contrib-blynk-ws
* node-red-contrib-influxdb
* node-red-contrib-moment
* node-red-dashboard

para importarlas, accedemos al menú de NODE-RED y seleccionamos “*Manage Palette*”



Vamos a la opción “*Install*” y en el cuadro “*Search modules*”



Escribimos el módulo a instalar, al seleccionarlo, pulsamos en el botón “*Install*”.

### Importar Flows

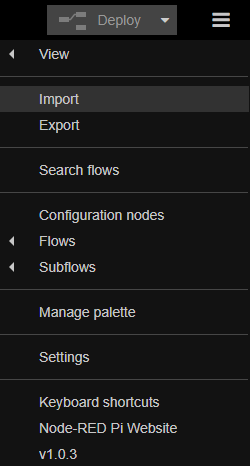
Descargamos los “Flows”de [github](https://github.com/Pedroda0608/MQTT-NodeRED-InfluxDB-Grafana-Blynk), en la carpeta NODE-RED, aquí encontraremos diferentes archivos que se corresponden cada uno de ellos con un Flow, hemos desarrollado los flows para que sean “modulables”, no es necesario importar todos para que sean funcionales independientemente.

Los flows son:

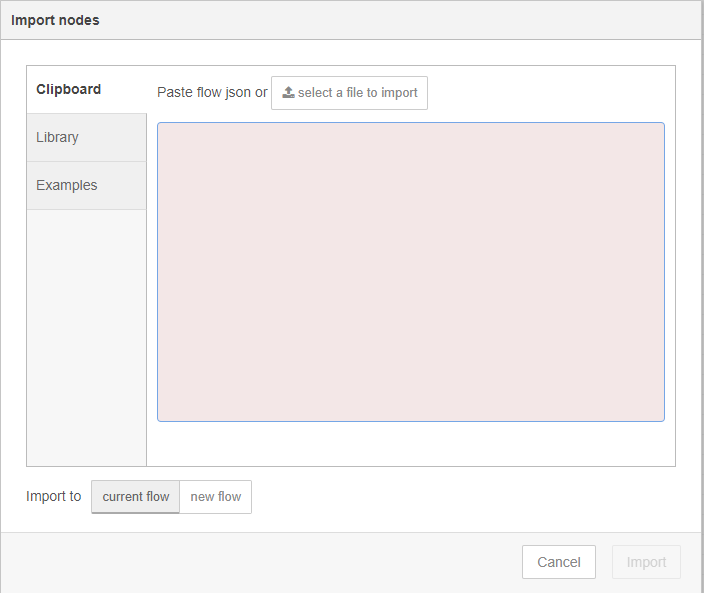
* **BME280-borrado de datos.json** Borrar los registros de BME280 de la bbdd.
* **BME280-comprobacion.json** Comprueba los datos recibidos por MQTT.
* **BME280-InfluxDB.json** Registra los datos en la bbdd.
* **BME280-Relojes.json** Muestra relojes de datos en el Dashboard.
* **BME280-seleccion datos.json** Muestra los datos registrados en la bbdd.
* **DS18B20.json** Flow con todos los procesos para DS18B20.
* **Graficas.json** Graficas en el Dashboard.
* **Envío\_Blynk.json** Envio de datos a Blynk.

Ahora vamos a importar los *flows* descargados.

Accedemos al menú de NODE-RED y pulsamos en “*Import*”



Nos aparece la pantalla de importación



Aquí podemos seleccionar, mediante el botón “*Select a file to import*” los ficheros descargados, de uno en uno, (también podemos abrir el fichero con **NotePad**, copiar y pegar en el cuadro de texto de Import).

Seleccionamos “New Flow” para que cree un Flow nuevo y pulsamos en el botón “*Import*”.

Una vez importados, deberemos revisar los parámetros de conexión.

Para Mqtt:

* En pestaña “connection” Server (IP del Servidor)
* En pestaña “Security” Username y Password (creados al instalar Mosquitto)

Para InfluxDB:

* Host (IP del servidor)

Para Blynk:

* Clave de autenticación (Auth\_Token generada al crear el proyecto en Blynk)

En los *Flows* que graban los datos en **InfluxDB**, hay un nodo de función donde se han definido las horas que realizamos registro de datos, aparte de que se envían datos a NODE-RED cada 60 segundos, para mostrar en el Dashboard de **NODE-RED** y en **Blynk**, recordad parámetro “*deep Sleep*” en Sketch de ESP8266, guardamos cada 6 horas (00:00, 06:00, 12:00 y 18:00).

Podemos cambiar estas horas, añadir nuevas o eliminar la función para registrar todos los datos recibidos.

## 9.- Creación Base de datos

Para crear la base de datos en InfluxDB, donde guardaremos los registros obtenidos, podemos hacerlo de varias formas, desde consola de comandos:

**# influx** (accedemos a la administración de InfluxDB)

**> CREATE DATABASE "mydb"**

**> exit**

También podemos hacerlo desde CHRONOGRAF, para ello accedemos a:

https://localhost:8888 (desde local)

https://***ip\_de \_nuestra\_raspberry***:8888 (desde otra maquina)

y en el apartado “**Explore**” añadimos una query y tecleamos:

**CREATE DATABASE " mydb"**

Por último, podemos ir al apartado “**InfluxDB Admin**”, seleccionamos la pestaña “**Databases**” y pulsamos en el botón **“+ Create Database**”, escribimos el nombre de la base de datos a crear (en nuestro caso mydb) y pulsamos en el botón de *check* para confirmar.

Podemos dar otro nombre a la base de datos, pero debemos cambiar los nodos de NODE-RED correspondientes a la grabación de datos en InfulxDB.

## 10.- Creación Dashboard en Grafana

Podemos encontrar, en la documentación de Grafana ayuda para el uso de InfluxDB en Grafana:

<https://grafana.com/docs/grafana/latest/features/datasources/influxdb/>

En líneas generales, los puntos a realizar son:

* Creación de un origen de datos que acceda a la base de datos creada para nuestro proyecto, ya vimos como realizarlo en la instalación del Dashboard para los datos obtenidos con TELEGRAF.
* Creación de un Dashboard para nuestro proyecto.
* Creación de Querys para obtener los datos a mostrar en el Dashboard
* Definición de la visualización de los datos para nuestro panel