

Activitat sm5a3.2: Instal·lació, configuració, comunicació i avaluació d'aplicacions utilitzant arquitectures heterogènies – Part II

1- Objectius

Aquesta pràctica té com a objectiu la instal·lació, execució i accés a una aplicació web en forma de contenidor Docker que s'executa sobre Raspberry Pi. S'haurà de:

- Crear una imatge de l'aplicació
- Executar múltiples instàncies de l'aplicació amb un nginx que faci de reverse proxy i balancejador de carrega.
- S'haurà d'accedir a l'aplicació des d'un altre equip de la xarxa.

2- Instal·lació, execució i accés a aplicacions dockeritzades

a) Instal·la el programari per treballar amb contenidors **Docker** sobre **Raspberry Pi** utilitzant la informació que trobaràs [aquí](#) a la secció **Install Docker on Raspberry Pi?**. Tingues en compte que:

- La instal·lació pot tenir una durada d'uns minuts. Pacència!!!!
- Has de fer un **logout** i **login** perquè sigui efectiva l'acció d'afegir el teu usuari de sistema al grup **docker**.

b) Dins de la **Raspberry Pi** crea un carpeta de nom **CostFabLlauBeg**. Entra a la carpeta i crea una carpeta de nom **app**.

c) Entra dins de la carpeta **app** i descarrega el fitxers d'una aplicació PHP que pots trobar al dipòsit <https://github.com/globproj2/CostFabLlauBeg.git> executant:

```
wget https://raw.githubusercontent.com/globproj2/CostFabLlauBeg/main/index.php
```

```
wget https://raw.githubusercontent.com/globproj2/CostFabLlauBeg/main/CostFabLlauBeg.php
```

d) Dins de la carpeta **CostFabLlauBeg**, crea un fitxer **Dockerfile** per poder crear una **nova imatge** de **Docker** amb les següents característiques:

- S'ha de crear a partir de la imatge ja existent de nom **php:8.4-apache**
- El directori de treball dels contenidors creats amb aquesta imatge serà per defecte **/var/www/html**.
- Ha de copiar el contingut de la carpeta **app** a la carpeta **/var/www/html** de la imatge.
- Exposa el port **80** dels contenidors creats amb aquesta imatge.

e) Crea una imatge de **Docker** que tingui el nom **llaunes** versió **1.0**. Això pot trigar uns minuts. Pacència!!!

f) Crea un contenidor de nom **llaunes** que s'executi en **2n terme**, amb accés a **terminal**, que exposi el port **80** intern del contenidor al port **8080** de la Raspberry Pi a partir de la imatge **llaunes** versió **1.0**.

h) Troba l'adreça IP de la Raspberry Pi.

i) Atura i deshabilita el servidor **Apache2** de la Raspberry Pi si està habilitat i actiu.

j) Accedeix a l'aplicació des del teu portàtil a partir de l'adreça IP de la teva Raspberry.

k) Crida'm per demostrar-me que funciona aquest apartat de l'activitat. Abans de cridar-me:

- Mostra la teva adreça IP amb l'ordre: **ip a show eth0**
- Accedeix a l'aplicació des de la teva màquina física i utilitzant l'adreça IP de la teva Raspberry Pi.
- Mostra les imatges de docker en la Raspberry PI amb l'ordre: **docker images**
- Mostra els contenidors en la Raspberry PI amb l'ordre: **docker ps -a**

NOTA: Si cal utilitzar **sudo** per treballar amb docker **NO S'ACCEPTARA EL LLIURAMENT**.

3- Instal·lació de node-red dockeritzat sobre Raspberry Pi

3.1- Què és node-red

Amb **node-red** és fàcil fer aplicacions web basades en el javascript i HTML però sense necessitat de tenir un coneixement gaire profund d'aquests llenguatges. Això és possible gràcies a l'utilització d'un blocs anomenats **nodes** que es poden interconnectar entre ells per fer un programa de manera ràpida i senzilla.

Els **nodes** s'agrupen dins d'un **Flow** (fluxe) per formar una aplicació. Normalment una aplicació està formada per un Flow.

Un cop creat el Flow s'ha de de fer un Deploy (Desplegament) perquè Node-RED pugui executar l'aplicació.

3.2- Instal·lació dockeritzada de node-red

a) Descarrega la imatge de node-red executant: **docker pull nodered/node-red**. Això pot trigar uns minuts. Pacència!!!

b) Crea una carpeta de nom **node-red** dins de la teva carpeta personal de la **Raspberry Pi**. Accedeix a la carpeta i crea una altra carpeta de nom **dades**.

c) Crea un contenidor executant **node-red** amb l'ordre:

```
docker run --name nr -i -t -d -p 1880:1880 -v dades:/data nodered/node-red
```

d) Comprova que s'ha obert el port **1800/tcp** de la Raspberry.

e) Accedeix a l'aplicació **node-red** des del teu portàtil a partir de l'adreça IP de la teva Raspberry.


g) Crida'm per demostrar-me que funciona aquest apartat de l'activitat. Abans de cridar-me:

- Mostra la teva adreça IP amb l'ordre: **ip a show eth0**
- Mostra les imatges de docker en la Raspberry PI amb l'ordre: **docker images**
- Mostra els contenidors en la Raspberry PI amb l'ordre: **docker ps -a**
- Accedeix a **node-red** des de la teva màquina física.

4 - Important un fluxe a Node-Red

a) Descarrega dins des del teu portàtil el fitxer [webserv.txt](#).

b) Esborra el fluxe **Flow1**. Crea un nou fluxe i canvia'l de nom. El nou nom serà **WebServer**.

c) Des del **Node-Red** ves a la icona de menú de la dreta  i escull **Import**. Copia i enganxa el codi que hi ha dins de **webserv.txt** i fes clic a **Import**. A continuació, fes un **Deploy** per despelgar el fluxe.

d) Des del teu equip físic connecta't a **http://ip_servidor_debian:1880/asix2** i comprova que tens una web disponible i accessible.

e) Fes clic a sobre del node **"Pàgina web"** i edita la pàgina web. Canvia el títol **"Hola ASIX2"** per **"Hola ASIX2 sm5act3.2"**. A continuació fes clic a **Done** i després fes un **Deploy** de l'aplicació.

f) Des del teu portàtil connecta't a **http://ip_servidor_debian:1880/asix2** i comprova la web actualitzada.

g) Crida'm per demostrar-me que funciona aquest apartat de l'activitat. Abans de cridar-me:

- Mostra la teva adreça IP amb l'ordre: **ip a show eth0**
- Accedeix a l'aplicació web des de la teva màquina física.

5 – Deshabilitat una aplicació

a) Fes clic a sobre de la pestanya **WebServer** i després, fes clic a **Disable Flow** per deshabilitar l'aplicació i després, fes un **Deploy**. Comprova que el servidor web deixa de funcionar i surt el missatge:

Can not GET /asix2

b) Fes clic a sobre de la pestanya **WebServer** i després, fes clic a **Enable Flow** per habilitar l'aplicació i després, fes un **Deploy**. Comprova que el servidor web funciona novament.

c) Crida'm per demostrar-me que pots habilitar/deshabilitar l'aplicació. Abans de cridar-me:


- Mostra la teva adreça IP amb l'ordre: **ip a show eth0**
- Accedeix a l'aplicació web des de la teva màquina física.


6 - Descarrega i instal·lació del mòdul Dashboard per Node-Red

a) a la icona de menú de la dreta  i escull Manage palette. A continuació:


- A **Install** → **search modules** i escriu: **node-red-dashboard**
- Instal·la el mòdul **node-red-dashboard** (està DEPRECATED però funciona correctament).



b) Comprova han aparegut nous nodes a la barra de l'esquerra dins de la secció **dashboard**. Si cal refresca el navegador o reinicia el docker de **node-red** fent un **stop** i un **start**.

c) Comprova que accedeixes a **Node-Red** i que també tens accés a la nova icona  a la part de superior de la dreta, juntament amb **info** i altres icones.

f) Comprova que si fas clic a la icona , pots accedir al **Dashboard** del projecte actual. Pel moment està buit.

g) Deshabilita el fluxe **servidor_web** i fes un **Deploy** per actualitzar el seu estat.

h) Descarrega el fitxer [projecte2.json](#) dins del teu portàtil. Des del **Node-Red** ves a la icona de menú  i **Import**. Després escull **select a file to import** i selecciona el fitxer **projecte2.json**. A continuació, fes clic a **Import**. Finalment, selecciona la nova pestanya **Projecte2_Dashboard** i fes un **Deploy** de l'aplicació.

i) Amb el fluxe **Projecte2_Dashboard** seleccionat, fes clic a sobre de la icona  i després a  per accedir al **Dashboard** de **Projecte2_Dashboard**.

j) Comprova que els controls i els indicadors creats funcionen correctament.

k) Modifica el node **ctrl_llums** perquè doni valors entre **0** i **1000**.

l) Modifica el node **ind_llums** per que mostri valors entre **0** i **1000**. La zona **verda** anirà de **0** a **400**, la zona **groga** de **401** a **600** i la **vermella** de **601** a **1000**.

m) Fes un **Deploy** de l'aplicació i comprova que funcionen els nous valors.

n) Si funciona, **deshabilita** el fluxe i fes un nou **Deploy** per desar l'estat de l'aplicació.

o) Crida'm per demostrar-me que pots habilitar/deshabilitar l'aplicació. Abans de cridar-me:

- Mostra la teva adreça IP amb l'ordre: **ip a show eth0**
- Accedeix a l'aplicació web des de la teva màquina física.

7- Treballant amb Node-Red i el sensor bme280/bmp280

a) Connecta la pantalla **OLED** i el sensor **BME280** a la **Raspberry PI** utilitzant el mòdul adaptador. A continuació, comprova que la Raspberry PI detecta els 2 dispositius executant:

```
i2cdetect -y 1
```

Si tot funciona bé, s'haurien de detectar els 2 dispositius a les adreces **0x3c** i **0x76**.

b) Atura i esborra el contenidor **nr** que vas crear a l'apartat 3.


c) Inicia un nou contenidor amb **Node-Red** utilitzant ara la següent ordre:

```
docker run --name nr -i -t -d -p 1880:1880 -v dades:/data --device /dev/i2c-1 --user node-red:994 nodored/node-red
```

Aquesta nova ordre té dues opcions:

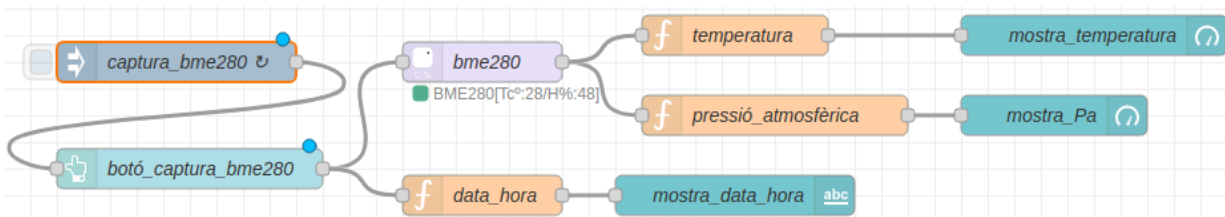
- `--user node-red:994`
- `--device /dev/i2c-1`

amb les quals node-red tindrà accés a les dades del sensor **BME280**.

d) Accedeix a **Node-Red** i selecciona **Menú de Node-RED**  i després l'opció "Manage Palette". A la secció "Install" demana la instal·lació del node **node-red-contrib-bme280** (pot trigar un parell de minuts).

c) Deshabilita tots els Flows existents i crea un nou fluxe de nom **bme280** i després fes un **Deploy**.

d) Crea una aplicació com aquesta:



e) El node de tipus **bme280** s'haurà de configurar amb les següents propietats:

- Nom: bme280
- Bus: 1
- I2C Address: 0x76 o 0x77 (en funció del valor trobat amb l'ordre **i2cdetect -y 1**)
- Topic: bme280

f) El node de tipus **inject** ha de tenir les següents propietats:

- Name: captura_bme280
- msg.payload → millisencs since epoch
- Habilitat → inject once after 0.1 seconds, then
- Repeat: interval
- every: 60 seconds

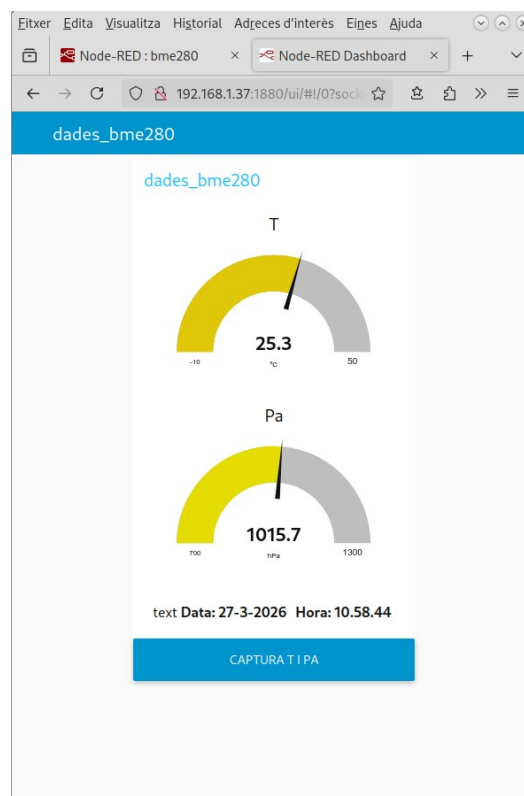
m) Configura el node de tipus **text de Dashboard** connectat a la funció **data_hora** així:

- Group: Afegeix-lo a [dades_bme280] dades_bme280
- Label → en blanc
- Layout → Escull la 2a opció de la primera fila que permet visualitzar el text centrat.
- Name: mostra_data_hora
- es clic a Done

n) Fes un **Deploy** de l'aplicació. Ara el servidor Node-RED té el programa salvat i en execució esperant que algú es connecti al fluxe Fluxe bme280 per visualitzar les dades.

o) Comprova que l'estat del node **bme280** és **Ready**.

p) Des del teu portàtil i amb el teu navegador accedeix al Dashboard de l'aplicació utilitzant l'URL **http://ip_raspberry:1880/ui**, i comprova que pots veure el mateix fluxe. Hauràs de veure una finestra similar a aquesta:



q) Comprova que la temperatura, data, hora i pressió atmosfèrica canvien i s'actualitzen:

- A cada minut
- Fent click a sobre del botó CAPTURA T i PA
- Fent click a sobre del **node** tipus **inject** de nom **captura_bme280** que es troba a la finestra de desenvolupament **Fluxe_codi_bme280**.

s) Si tot ha anat bé, exporta el **current flow complet formatat** a un fitxer **JSON local** de nom **bme280.json** (no cal afegir extensió, s'escriu automàticament).

- Deshabilita l'aplicació i fes un **Deploy** per desar el seu estat.
- Atura **Node-RED** correctament executant: **docker stop nr**

t) Crida'm per demostrar-me que pots accedir a l'aplicació web des del teu portàtil i que funciona correctament.

8- Treballant amb Node-RED i telegram - Configuració inicial i proves de funcionament

a) Deshabilita l'aplicació **bme280** i la resta de Flows.

b) Instal·la el mòdul **node-red-contrib-telegrambot**. Pot trigar uns minuts. Pacència!!

c) Obre aquesta pàgina: <https://www.sohamkamani.com/blog/2016/09/21/making-a-telegram-bot/>

c) Obre el teu telegram i la secció de cerca escriu **@BotFather** per trobar el **bot** de nom **BotFather** que és el pare de tots els **bot** de **telegram**. Un **bot** és una aplicació que es comporta com un usuari de **telegram**. Es pot fer una aplicació d'aquest tipus amb **node-RED**.

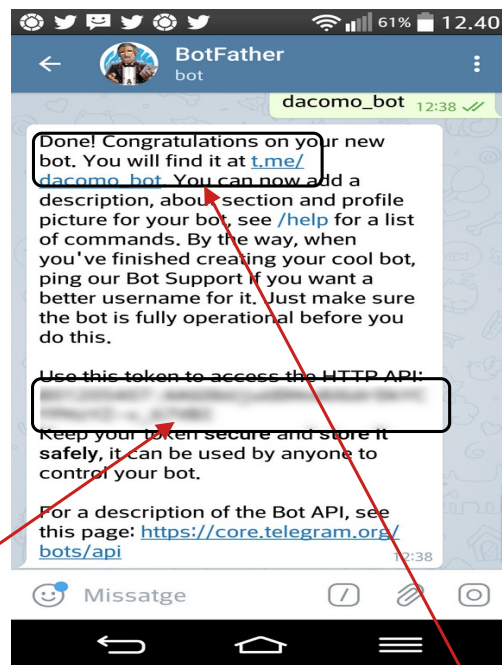
d) Dins del chat amb **@BotFather**, envia els missatges:

/newbot

i hauré d'escollir:

- Un nom pel **bot** → **sm5grpXX_bot** a on **XX** és el vostre número de grup.
- Un nom d'usuari pel bot → **sm5grpXX_bot** a **XX** és el vostre número de grup i el **_bot** del final és molt important.

e) Mira que en el darrer missatge enviat per **@BotFather** hi ha un codi anomenat **token**. Fes una captura de pantalla amb el teu mòbil per no perdre'l:



El **token** correspon a la zona difuminada.

Aquest és el teu **nom d'usuari** del bot.

f) Ara fes click a sobre de l'enllaç **t.me/ sm5grpXX_bot** (recordant que **XX** és el teu número de grup) que trobaràs al principi del mateix missatge per accedir per primer vegada al **chat** del teu **bot**.

g) Afegeix un nou **Flow** de nom **chat_telegram_sm5grpXX_bot** (a on **XX** és el teu número de grup).

h) Afegeix un node de **telegram** de tipus **sender**:

- Afegeix un nou bot de telegram amb aquestes característiques:
 - Boot-Name: **sm5grpXX_bot**
 - Token: Escriu el token del teu bot que hauràs rebut durant la seva creació
 - Fes un **Add**.
- Nom del node: **envia_sm5grpXX_bot**
- Fes clic a **Done**.

i) Afegeix un node de **telegram** de tipus **receiver**:

- Bot → selecciona **sm5grpXX_bot**
- Nom del node: **rep_sm5grpXX_bot**
- Download Directory: **/usr/src/node-red**
- Fes un **Done**.

j) Connecta l'**entrada** del node de **telegram sender** a la primera **sortida** del node **telegram receiver**.

k) selecciona el nodes de tipus dashboard **gauge**. Edita'l:

- Group → clic a la icona per afegir un nou grup (icona amb el signe +):
 - Name: **telegram_sm5grpXX_bot**
 - Tab (pestanya) → clic a la icona per afegir un nou tab (icona amb el signe +):
 - Name: **telegram_sm5grpXX_bot**
 - Fes clic a **Add**
 - Fes clic a **Add**
- Type: **Gauge**
- Label: **Velocitat de la xarxa**
- Units: **Mbps**
- Range → mín=0 a max=1000
- Name: **telegram_sm5grpXX_bot_bot**
- Fes clic a **Done**

m) Afegeix un node tipus **funció** i de nom **ObtencioValor** amb el contingut:

```
msg.payload=msg.payload.content;  
return msg;
```

i fes un **Done**.

n) Connecta la primera **sortida** del node de **telegram receiver** al l'**entrada** de la funció **ObtencioValor**.

o) Connecta la sortida de la funció **ObtencioValor** a l'entrada del node de tipus **gauge**.

p) Fes un **Deploy**.

q) Comprova que els node de telegram canvien a l'estat "**connected**".

r) Des del teu portàtil i amb el teu navegador accedeix al Dashboard de l'aplicació utilitzant l'URL **http://ip_raspberry:1880/ui**.

s) Escriu **/start** al teu **bot** de telegram per començar les comunicacions.

t) Envia valors entre **0** i **1000** des del bot de telegram i comprova que:

- Canvia l'indicador del dashboard.
- Dins del chat es replica el valor enviat.

u) Crida'm per demostrar-me que pots accedir a l'aplicació web des del teu portàtil i que funciona correctament.

9- Obtenció del Chat ID del Bot de Telegram

a) Deshabilita la resta de Flows. Crea un nou Flow de nom **ChatIdTelegram**.

b) Afegeix un node de tipus **telegram receiver**:

- Selecciona el bot de l'apartat anterior
- Download Directory: **en blanc**
- Nom del node: **rep_chatid_sm5grpXX_bot**
- Fes un **Done**.

c) Afegeix un node de tipus un **funció** de nom **ObtencioChatId** amb el contingut:

```
msg.payload=msg.payload.chatId;
return msg;
```

i després fes un **Done**.

d) Afegeix un node tipus **debug** que:

- Doni com a sortida el **msg.payload**
- El nom sigui **mostra_ChatId**

i després fes un **Done**.

e) Connecta la primera **sortida** del node de **telegram receiver** al **entrada** de la funció **ObtencioChatId**.

f) Connecta la sortida d'**ObtencioChatId** a l'**entrada** de **mostra_ChatId**.

g) Fes un **Deploy** del programa i després selecciona l'eina "**Debug messages**". Esborra qualsevol missatge anterior.

h) Escriu un missatge des del teu **bot** i comprova que pots obtenir el teu **ChatId**. Apunta-te'l.

i) Crida'm per mostrar-me el teu ChatId.

10- Mostrant la temperatura i Pressió atmosfèrica dins del Bot de Telegram

a) Si has pogut trobar el teu **ChatId**, deshabilita el Flow **ChatIdTelegram**.

b) Habilita i accedeix al Flow **bme280** que vas crear a l'apartat 7.

c) Afegeix un node de tipus **funció** i fes que el seu nom sigui **msg_temp_bot**. Connecta la seva entrada a la sortida del node de tipus **funció** i de nom **temperatura**. Fes que tingui el següent codi:

```
msg.payload = {
  "content": "temp: " + msg.payload + " °C",
  "chatId": "xxxxxxxx",
  "type": "message"
}
return msg;
```

A on és **xxxxxxxx** el l'identificador del chat associat al teu bot de telegram que vas trobar a l'apartat anterior.

Finalment fes un **Done**.

c) Afegeix un node de tipus **funció** i fes que el seu nom sigui **msg_Pa_bot**. Connecta la seva entrada a la sortida del node de tipus **funció** i de nom **pressió_atmosferica**. Fes que tingui el següent codi:

```
msg.payload = {
  "content": "Pa: "+msg.payload+" hPa",
  "chatId": "xxxxxxxx",
  "type": "message"
}
return msg;
```

i després fes un **Done**.

d) Afegeix un node de tipus **delay**. Connecta la seva entrada a la sortida del node **msg_Pa_bot**. Configura'l de la següent manera:

- Action:
 - **Delay each message**
 - **Fixed delay**
- For **10 Miliseconds**
- Name del node: **delay_Pa**
- Fes un **Done**.
-

e) Afegeix un node de tipus **telegram sender** amb la configuració:

- Bot → selecciona el teu **bot**
- Nom del node: **envia_dades_bot**

i després fes un **Done**.

f) Connecta l'entrada del node **envia_dades_bot** a la sortida dels següents 2 nodes:

- La funció **msg_temp_bot**
- El delay **delay_Pa**

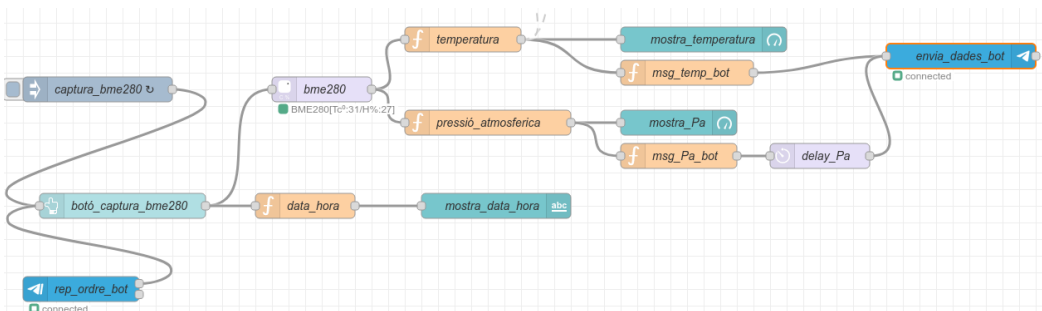
g) Afegeix un node de tipus **telegram command** amb la configuració:

- Bot → selecciona el teu **bot**
- Nom del node: **rep_ordre_bot**
- Command: **/envia**

i després fes un **Done**.

h) Connecta la primera sortida del node **rep_ordre_bot** a l'entrada del botó **botó_captura_bme280**.

i) **Ajuda:** Exemple de com pot quedar del connexionat de l'aplicació:



j) Fes un **Deploy**.

k) Des del teu portàtil i amb el teu navegador accedeix al Dashboard de l'aplicació utilitzant l'URL http://ip_raspberry:1880/ui.

l) Comprova que:

- El node bme280 està "Ready".
- Els nodes de telegram estan "connected".
- Cada minut:
 - Reps un missatge en el teu bot de telegram amb la temperatura i pressió atmosfèrica.
 - Canvia en el dashboard els indicador de data, hora, temperatura i pressió atmosfèrica.
- Que si envies el missatge /**envia** des del bot:
 - Reps un missatge en el teu bot de telegram amb la temperatura i pressió atmosfèrica.
 - Canvia en el dashboard els indicador de data, hora, temperatura i pressió atmosfèrica.
- Si des del Dashboard prems el botó de Captura T i Pa:
 - Reps un missatge en el teu bot de telegram amb la temperatura i pressió atmosfèrica.
 - Canvia en el dashboard els indicador de data, hora, temperatura i pressió atmosfèrica

m) Crida'm per demostrar-me que pots accedir a l'aplicació web des del teu portàtil i que funciona correctament d'acord amb les indicacions de l'apartat anterior.

11- Mostrant la temperatura via MQTT

a) Afegeix un node de tipus **funció** i fes que el seu nom sigui **mqtt_temp**. Connecta la seva entrada a la sortida del node de tipus **funció** i de nom **temperatura**. Fes que tingui el següent codi:

```
msg.payload = msg.payload + "°C";
return msg;
```

Finalment fes un **Done**.

b) Afegeix un node de tipus **mqtt out** i fes que la seva configuració sigui aquesta:

- Server:
 - Name: **mqtt_temp_sm5grpXX**
 - Server: **broker.hivemq.com** (broker públic vàlid per fer proves)
 - Port: **1883**
 - Connected automatically → **Seleccionat**
 - Protocol: **MQTT V3.1.1**
 - Session → Use clean session → **Seleccionat**
 - Fes un **Add**.
- Topic: **mqtt_temp_sm5grpXX**
- Nom del node: **mqtt_temp_sm5grpXX_bot**
- Fes clic a **Done**.

c) Connecta la sortida del node de la funció **temperatura** a l'entrada del node de la funció **mqtt_temp**.

d) Connecta la sortida del node de la funció **mqtt_temp** a l'entrada del node **mqtt_temp_envia**.

e) Fes un **Deploy**. Comprova que el node **mqtt_temp_envia** està "connected".

f) Amb l'aplicació de mòbil **MyMQTT** connectat al servidor **broker.hivemq.com** i fes una subscripció al topic **mqtt_temp_sm5grpXX**.

g) Crida'm per demostrar-me que pots accedir a la temperatura via MQTT des del teu mòbil.