

## Pràctica 2: Aplicacions d'explotació del sistema

### PART 1 - Instal·lació, execució i accés a aplicacions dockeritzades sobre Raspberry Pi: Instal·lació de node-red

#### 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 **How to install Docker on Raspberry Pi?**. Recorda que has de fer un logout i login perquè sigui efectiva l'acció d'afegir el teu usuari de sistema al grup **docker**. Instal·la a continuació el paquet **docker-compose**.

b) Dins de la **Raspberry Pi** crea un carpeta de nom **m05uf2pr2a**. 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 **m05uf2pr2a**, 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:7.4-apache**
- Ha de copiar el contingut de la carpeta **app** a la carpeta **/var/www/html** de la imatge.
- El directori de treball dels contenidors creats amb aquesta imatge serà per defecte **/var/www/html**.
- Exposa el port **80** dels contenidors creats amb aquesta imatge.

e) Crea una imatge de **Docker** que tingui el nom **llaunes** versió **1.0**.

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 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.

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

a) Descarrega la imatge de node-red executant:

```
docker pull nodered/node-red
```

b) Crea una carpeta de nom **node-red**. Accedeix a la carpeta i crea una carpeta de nom **dades**.

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

```
docker run -itd -p 1880:1880 -v dades:/data --name dockerNodeRed nodered/node-red
```

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

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

#### LLIURAMENT DE LA PART 1

a) Comprovació que el contenidor està en marxa.

b) Comprovació d'accés a l'aplicació des del portàtil amb l'adreça IP de la Raspberry P.

d) Accés a node-red de la Raspberry Pi.

e) Data de lliurament (100%): **31-01-2025** (posteriorment 70%)

## PART 2 - Desplegant aplicacions d'explotació del sistema amb node-red sobre Raspberry Pi

### 1- Breu introducció a 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 fer un **Deploy** (Desplegament) perquè Node-RED pugui executar l'aplicació.

### 2 - Accions bàsiques amb Node-Red

a) Inicia un contenidor amb Node-Red amb la següent ordre:

```
docker run -itd -p 1880:1880 -v dades:/data --device /dev/i2c-1 --user node-red:994 --name dockerNodeRed nodered/node-red
```

b) Amb la informació dins del següent enllaç:


<http://developer.opto22.com/nodered/general/getting-started/node-red-hello-world/>

crea dins **Flow1** un fluxe (flow) que injecti el missatge "**Hola món!!!!**" cada **2** segons.

c) Fes clic a sobre de **Flow1** i canvia el nom del fluxe a **fluxe\_holamón\_v1**.

- d) Fes clic a sobre de **fluxe\_holamón\_v1** i després, a la part inferior esquerra, fes clic a **Enabled** per deshabilitar el fluxe. Comprova que deixa **fluxe\_holamón\_v1** de funcionar.
- e) Torna a habilitar **fluxe\_holamón\_v1** i després a la part superior de la dreta fes clic a sobre de **Delete**. Comprova que no pots esborrar-lo perquè és l'únic fluxe que hi ha en **Node-Red**.
- f) Fes clic a el botó **+** i comprova que es crea un nou fluxe de nom Flow2. Canvia el nom a **fluxe\_Holamón\_v2**.
- g) Selecciona els 2 nodes i la connexió que hi ha dins de **fluxe\_holamón\_v1**, copia'ls fent **Ctrl+c** i enganxa'ls dins de **fluxe\_holamón\_v2** fent **Ctrl+v**. Fes un **Deploy** per desar el fluxe i posar-lo en marxa.
- h) Torna a seleccionar **fluxe\_holamón\_v1** i intenta esborrar-lo novament. Comprova que ara s'esborra.

### **3 - Important un fluxe a Node-Red**

- a) Deshabilita **fluxe\_holamón\_v2** i fes un **Deploy** per actualitzar el seu estat.
- b) Descarrega des del teu equip el fitxer [webserv.txt](#).
- c) Crea un nou fluxe i canvia'l de nom. El nou nom serà **servidor\_web**.
- d) 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**.
- e) Fes un **Deploy** per desar el fluxe i posar-lo en marxa.
- f) Des del teu equip físic connecta't a **http://ip\_servidor\_debian:1880/asix2** i comprova que tens una web disponible i accessible.
- g) Fes clic a sobre del node "Pàgina web" i edita la pàgina web. Canvia el títol "Hola ASIX2" per "Hola ASIX2 SM5A2". Canvia "Hola classe d'ASIX2!" per "Hola classe d'ASIX2 SM5A2!". Fes un **Deploy** per actualitzar el fluxe i reiniciar-lo.
- h) Des del teu equip físic connecta't a **http://ip\_servidor\_debian:1880/asix2** i comprova la web s'ha actualitzat.

### **4 - Descarrega i instal·lació del mòdul Dashboard per Node-Red**

- a) Accedeix des de la Raspberry al bash del docker **dockerNodeRed**:

```
docker exec -it dockerNodeRed bash
```

i un cop dins del contenidor, executa la següent ordre per descarregar el mòdul **Dashboard** de **Node-Red**:

```
npm i node-red-dashboard
```

- b) Encara dins del contenidor, accedeix a **node\_modules**, que és el directori a on s'ha descarregat el mòdul, i instal·la el mòdul descarregat. Executa:

```
cd node_modules  
cd node-red-dashboard  
npm install
```

**NOTA 1:** No et preocupis pels warnings, la instal·lació funcionarà.

**NOTA 2:** Pot trigar una estona perquè és un mòdul important i d'una mida força gran.

c) Reinicia **Node-Red**. Surt del node **dockerNodeRed** i executa les següents ordres:

```
docker stop dockerNodeRed
docker start dockerNodeRed
```

d) Accedeix amb el navegador a **http://adreça\_ip:1880**. Comprova que pots accedir a **Node-Red** i que també, a la part de dalt de la dreta, juntament amb info i altres icones també tens accés a la nova icona:



i també si fas clic a aquesta icona, pots accedir amb el navegador al **Dashboard** fent clic a la icona:



e) Deshabilita el fluxe **servidor\_web** i fes un **Deploy** per actualitzar el seu estat.

f) Descarrega el fitxer [projecte2.json](#). 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**. Ara selecciona la nova pestanya **Projecte2\_Dashboard** i fes un **Deploy** de l'aplicació.

g) Amb el fluxe **Projecte2\_Dashboard** seleccionat, fes clic a sobre de la icona i després a per accedir al **Dashboard** de **Projecte2\_Dashboard**.

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

i) Modifica el node **ctrl\_llums** perquè doni valors entre **0 i 1000**.

j) 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**.

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

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

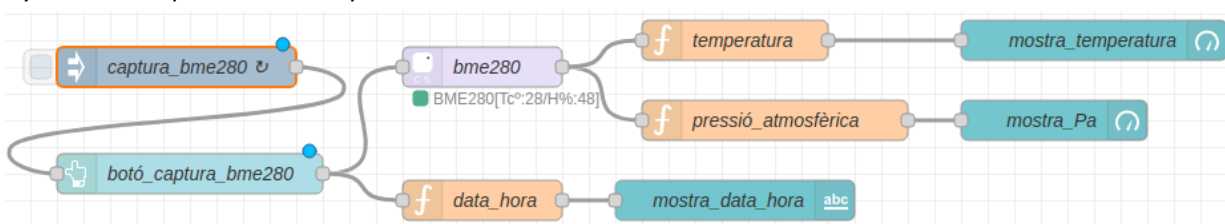
#### 4- Treballant amb Node-Red i el sensor bme280/bmp280

a) Connecta el sensor BME280 a la Raspberry Pi tal i com ho vas fer a la pràctica [eh5a1pr2](#) i comprova que el sistema ho detecta. Llegeix la pàgina **6** de la pràctica indicada.

b) 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 **Fluxe\_codi\_bme280** i després fes un **Deploy**.

d) Crea una aplicació com aquesta:





l) Configura el node de tipus **Gauge de Dashboard** connectat a la funció **pressió\_atmosfèrica** amb les següents propietats:

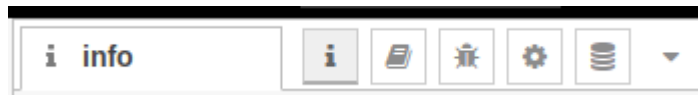
- Group → Afegeix-lo a [Fluxe\_bme280] dades\_bme280
- Type: Gauge
- Label: Pa
- Units: hPa
- Range:
  - Mín: 700
  - max: 1300
- Name: mostra\_Pa
- Fes clic a Done

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

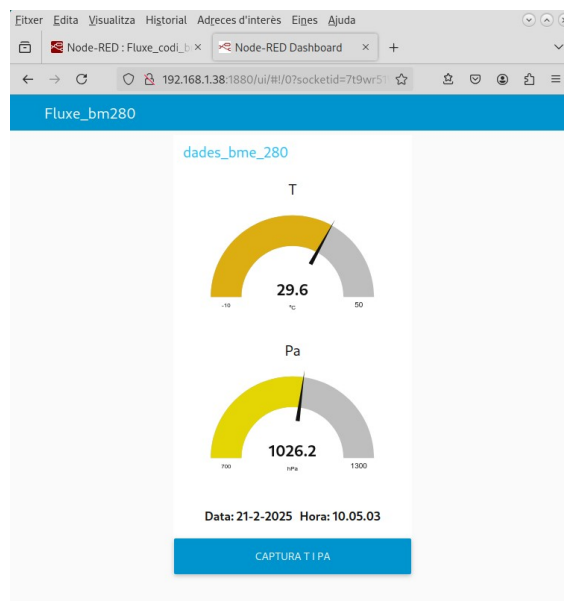
- Group: Afegeix-lo a [Fluxe\_bme280] dades\_bme280
- Label → en blanc
- Layout → Escull la 2a opció de la primera fila que permet visualitzar el text centrat.
- Name: mostra\_data\_hora
- es clic a Done

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) Accedeix al **Dashboard** de l'aplicació. A la dreta de la finestra trobaràs això:



p) Fes clic al triangle. A continuació selecciona:  Dashboard. Després fes clic a  i 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**.

r) Finalment, des del teu portàtil i amb el teu navegador accedeix al Dashboard de l'aplicació utilitzant l'URL [http://ip\\_raspberry:1880/ui](http://ip_raspberry:1880/ui), i comprova que pots veure el mateix fluxe.

s) Si tot ha anat bé i s'ha corregit la pràctica:

- Exporta el **flow complet** a un fitxer JSON 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 dockerNodeRed`.
- Atura correctament el sistema operatiu de la **Raspberry Pi 3+**.
- Desmunta correctament tot el maquinari connectat a la **Raspberry Pi 3+**.

## LLIURAMENT DE LA PART 2

a) Comprovació del funcionament de l'aplicació de visualització de temperatura, pressió atmosfèrica, data i hora. Això vol dir que:

- Els indicadors marquen la temperatura i pressió atmosfèrica dins dels marges indicats i amb les etiquetes i unitats demanades.
- L'aplicació mostra dia i hora.
- El botó CAPTURA T I PA funciona (com a mínim, canvia l'hora).

b) Data de lliurament: **21-02-2025** pel 100%. Després un 70%.

## PART 3 – Afegint missatgeria amb Telegram sobre l'aplicació de monitoratge de temperatura i Pressió relativa amb node-red dockeritzat i sensor BME280 sobre Raspberry Pi

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

a) Instal·la el mòdul **node-red-contrib-telegrambot**.

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

c) Obre el teu telegram i la secció de busqueda escriu **@botfather** per trobar el **bot** de nom **botfather** que és el pare de tots els usuaris **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**.

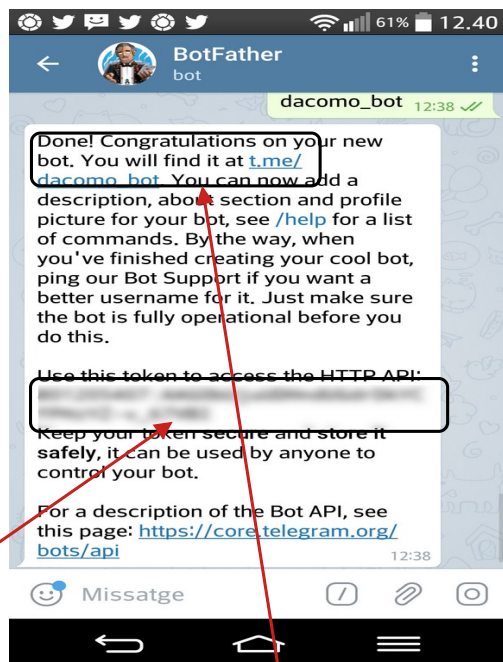
d) Dins del chat amb **@botfather**, envia els missatges:

**/newbot**

i haurem d'escollir:

- Un nom pel **bot** → **eh5grupXX\_bot** a on **XX** és el vostre número de grup.
- Un nom d'usuari pel bot → **eh5grupXX\_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.

f) Ara fes click a sobre de l'enllaç **t.me/ eh5grupXX\_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) Crea un nou **Flow** de nom **chat\_telegram\_grupXX\_bot** (a on XX és el teu número de grup).

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

a) Afegeix un nou bot de telegram amb aquestes característiques:

Bot-Name: **eh5grupXX\_bot**

Token: *Escriu el token del teu bot que hauràs rebut durant la seva creació*

i fes un **Add**.

b) Nom del node: **envia\_eh5grupXX\_bot**

i després fes un **Done**.

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

a) Bot → selecciona **eh5grupXX\_bot**

c) Nom del node: **rep\_eh5grupXX\_bot**

b) Download Directory: **/usr/src/node-red**

i després fes un **Done**.

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

k) Deshabilita la resta de Flows.



l) 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\_eh5grupXX\_bot**
  - Tab (pestanya) → clic a la icona per afegir un nou tab (icona amb el signe +):
    - Name: **tab\_telegram\_eh5grupXX\_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: **eh5grupXX\_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) Obre el **Dashboard** del **Flow** clicant a sobre de 

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.

## 2- 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\_eh5grupXX\_bot**

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

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

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

a) **Habilita**, fes un **Deploy** i accedeix al Flow **Fluxe\_codi\_bme280** que vas crear a l'apartat 2.

b) 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.pressure_hPa.toFixed(1)+" 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**

b) For **10 Milliseconds**

c) Name del node: **delay\_Pa**

i després 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**.

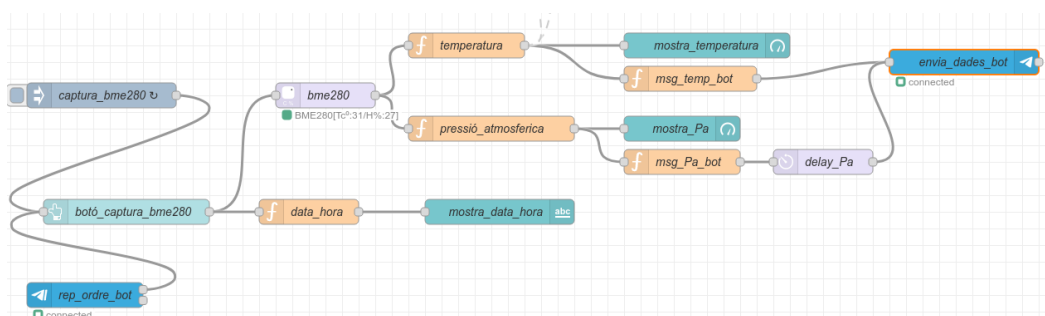
h) Connecta la primera sortida del node **rep\_ordre\_bot** a l'entrada del botó **botó\_captura\_bme280**.

i) Fes un **Deploy**.

j) Comprova que:

- Cada minut:
  - Repts 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:
  - Repts 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:
  - Repts 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

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



### LLIURAMENT DE LA PART 3

- a) Comprovació del funcionament de l'aplicació d'acord amb les indicacions del punt j) de l'apartat 3  
b) Data de lliurament: **28-02-2025** pel 100%. Després un 70%.