

Node-RED

M09UF3
M15UF1

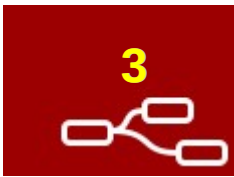


Node-RED és una eina de programari desenvolupada per l'empresa IBM que permet connectar dispositius físics en l'anomenat internet de les coses (IdC o IoT).

<https://nodered.org/>

Node-RED

Documentació



Molts exemples basats en la guia de programació de Node RED.

<http://noderedguide.com/>

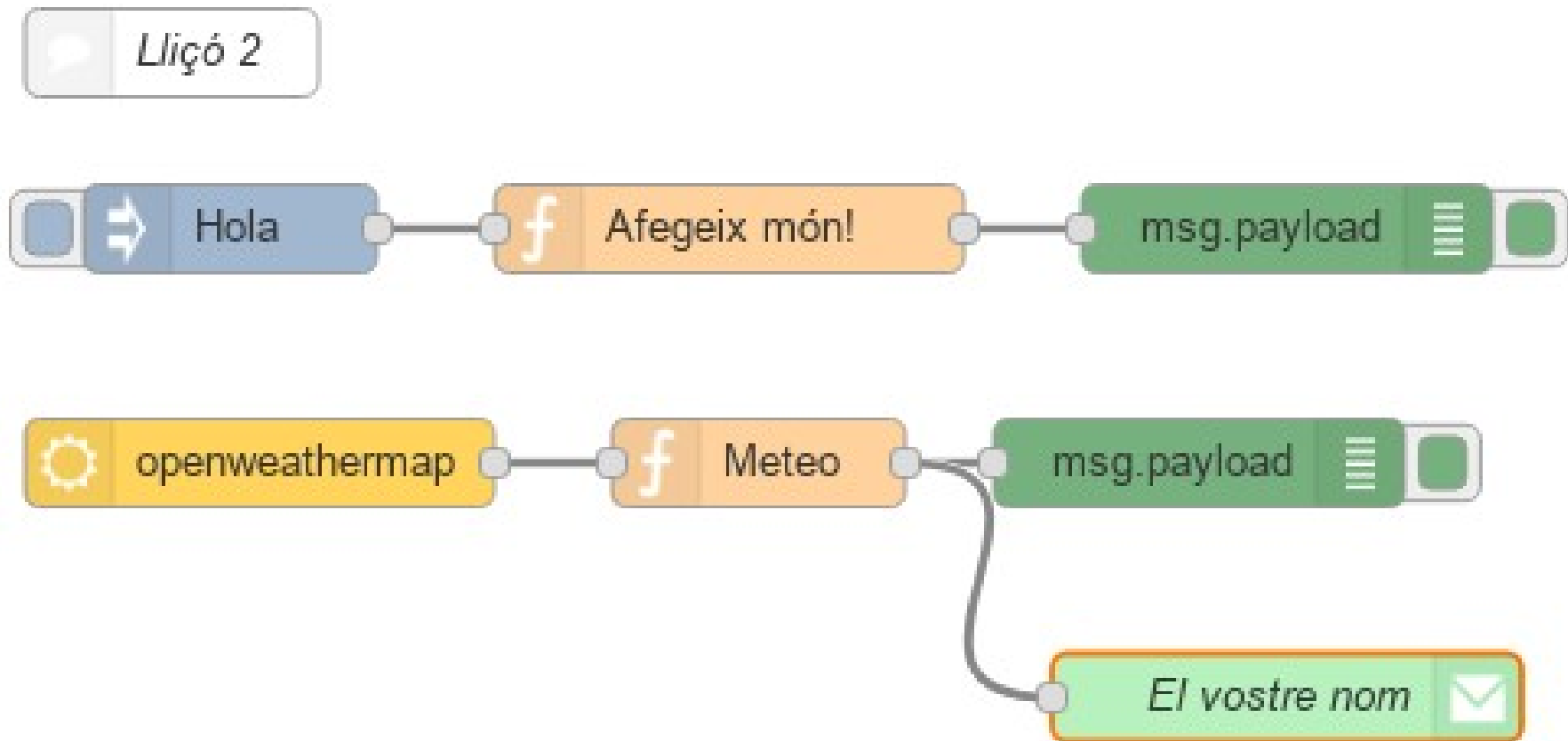
Node RED Programming Guide

Programming the IoT

Home

The node red guide is a series of lectures designed to get you started with Node-RED. Starting from the very basics, it introduces you to how Node-RED works and how to get the most out of its built in nodes. Each lecture consists of a series of hands-on examples that take you from basic to intermediate Node-RED programming in easy to follow steps.

Conèixer el temps i trametre-ho per correu



Node-RED

Node comentari

A screenshot of the Node-RED web interface. The top bar shows the Node-RED logo and the text 'Node-RED'. Below it, there's a search bar for nodes and a tab for 'Flow 1'. The main workspace contains several nodes: a 'comment' node with the text 'Lliçó 2', a 'msg' node with the text 'Hola', and an 'openweathermap' node. The 'comment' node is selected, and an 'Edit comment node' dialog is open. The dialog has three buttons: 'Delete', 'Cancel', and 'Done'. It contains a 'Title' field with the text 'Lliçó 2' and a 'Body' text area with the following text:

```
1 Aquest és un exemple de la segona lliçó de node  
2  
3 http://noderedguide.com/node-red-lecture-2-bui
```

Node-RED

Node d'injecció

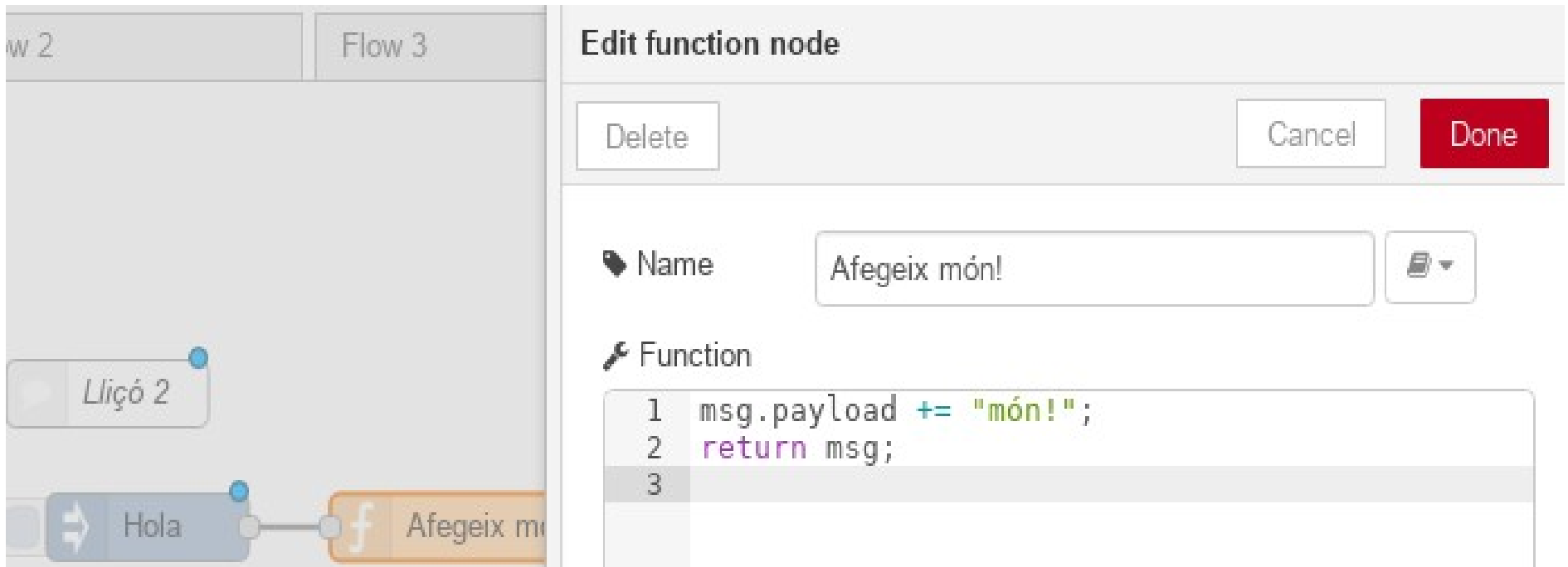


The screenshot shows the Node-RED interface. On the left, a sidebar lists various nodes under the 'input' category, including 'inject', 'catch', 'status', 'link', 'mqtt', and 'http'. The main workspace displays a flow with three nodes: 'Lliçó 2', 'Hola', and 'openweather'. The 'Edit inject node' configuration panel is open, showing the following settings:

- Delete**: Button
- Cancel**: Button
- Done**: Button
- Payload**:
- Topic**:
- Repeat**:
- Inject once at start?**:
- Name**:

Node-RED

La funció Afegeix món!

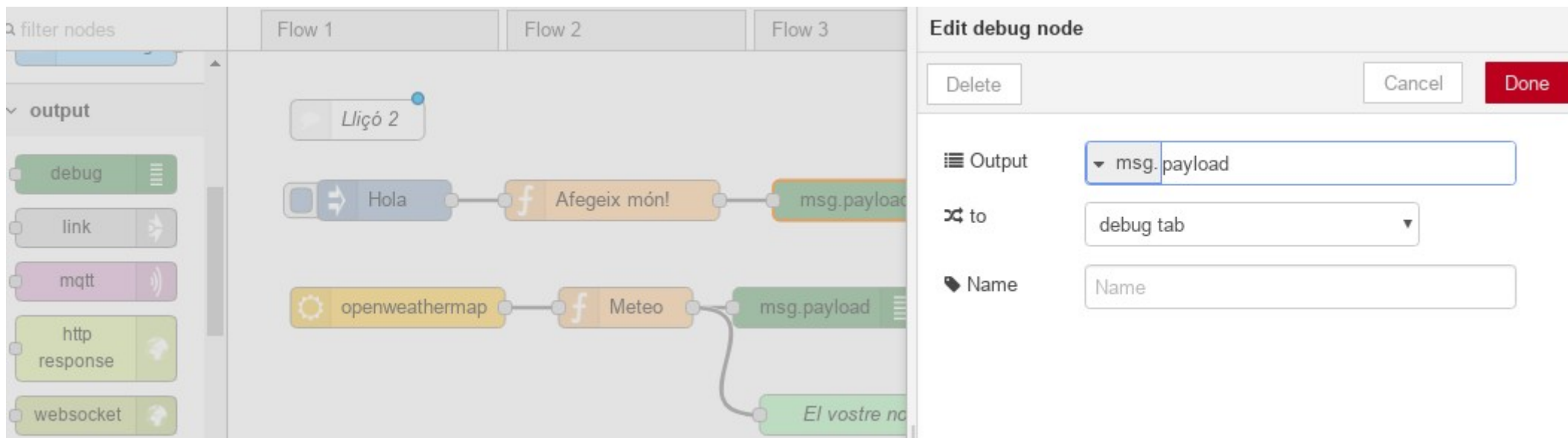


The screenshot shows the Node-RED interface. On the left, a flow named 'Flow 3' contains a message node with the text 'Hola' and a function node labeled 'Afegeix món!'. The function node is selected, and the 'Edit function node' panel is open on the right. The panel has three buttons: 'Delete', 'Cancel', and 'Done'. The 'Name' field contains 'Afegeix món!' and the 'Function' field contains the following JavaScript code:

```
1 msg.payload += "món!";  
2 return msg;  
3
```

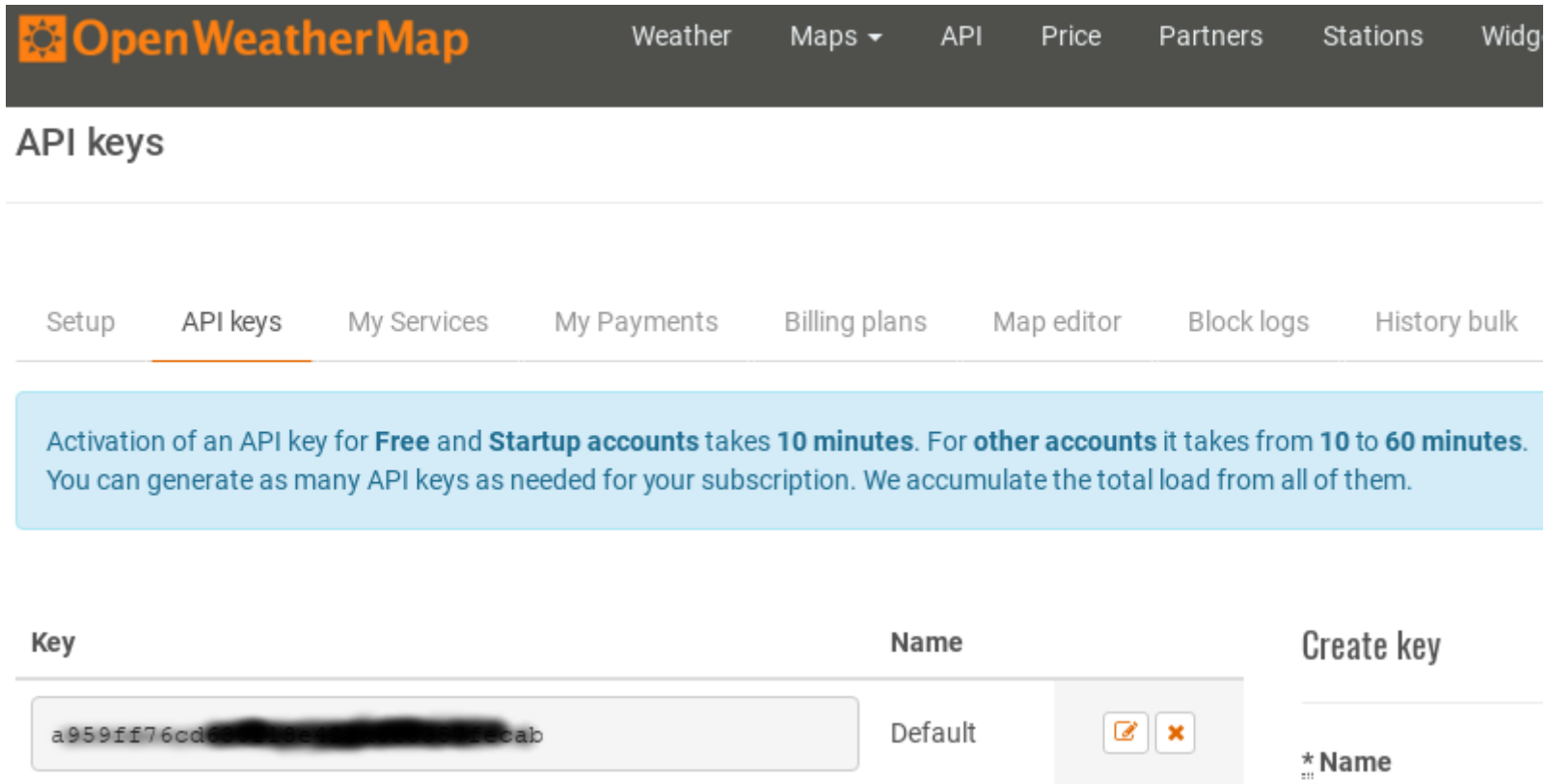
Node-RED

Node de depuració




The screenshot displays the Node-RED web interface. On the left, a sidebar shows various nodes under the 'output' category, including 'debug', 'link', 'mqtt', 'http response', and 'websocket'. The main workspace shows a flow with three nodes: a 'Hola' node, an 'Afegix món!' function node, and a 'msg.payload' node. Below this, another flow starts with an 'openweathermap' node, followed by a 'Meteo' function node, which connects to two 'msg.payload' nodes. One 'msg.payload' node is connected to a 'debug' node. The right-hand panel, titled 'Edit debug node', contains configuration options: 'Output' set to 'msg.payload', 'to' set to 'debug tab', and a 'Name' field.

Aconseguíu les claus de l'API d'OpenWeatherMap



The screenshot shows the OpenWeatherMap website's API keys management interface. At the top, there is a navigation bar with the OpenWeatherMap logo and links for Weather, Maps, API, Price, Partners, Stations, and Widgets. Below the navigation bar, the page title is "API keys". A secondary navigation bar contains links for Setup, API keys (which is underlined), My Services, My Payments, Billing plans, Map editor, Block logs, and History bulk. A light blue informational box states: "Activation of an API key for **Free** and **Startup accounts** takes **10 minutes**. For **other accounts** it takes from **10 to 60 minutes**. You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them." Below this, there is a table with columns for Key, Name, and Create key. The first row shows a key value (partially obscured), the name "Default", and a "Create key" button with a pencil icon and a close icon. A small asterisk and "Name" label are visible below the button.

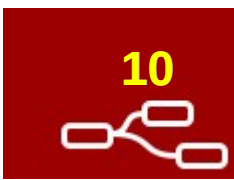
Key	Name	Create key
a959ff76cd...cab	Default	

Claus: https://home.openweathermap.org/api_keys

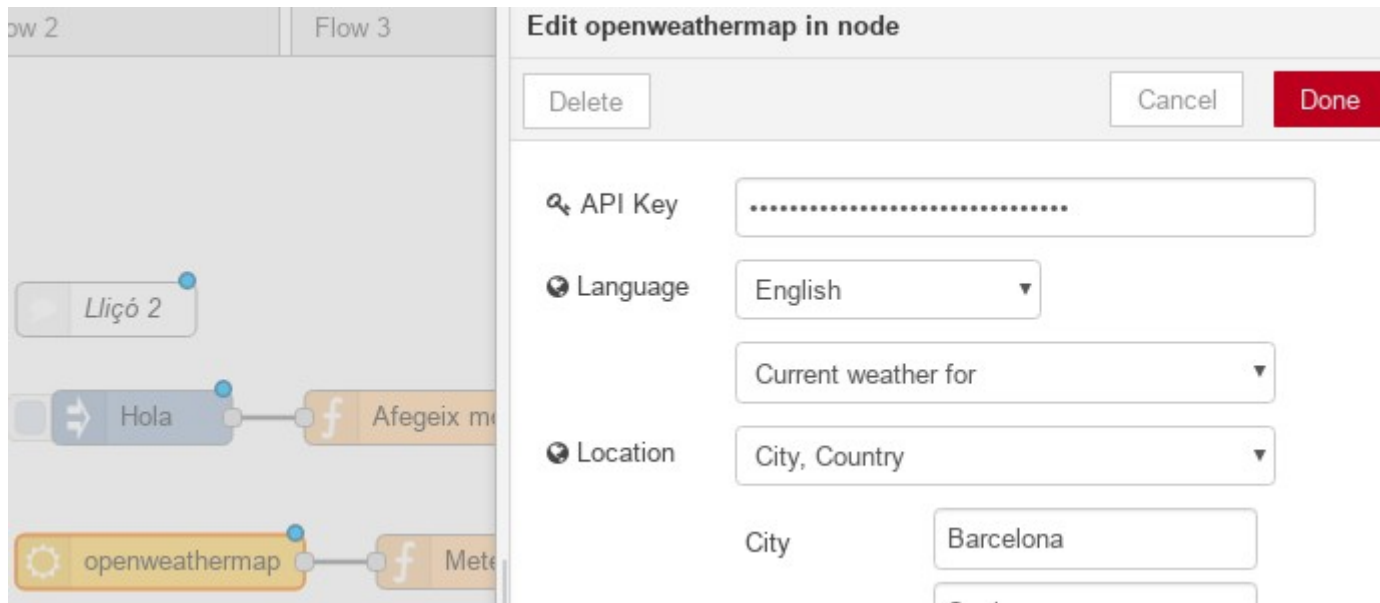
https://binefa.cat/IoT/nodeRed/02_nodeRed_01.txt



Node-RED



Introduïu la clau de l'API d'OpenWeatherMap



Clau: https://home.openweathermap.org/api_keys

https://binefa.cat/IoT/nodeRed/02_nodeRed_01.txt


Node-RED

La funció Meteo

Window 3

Edit function node


Delete Cancel Done

Name 

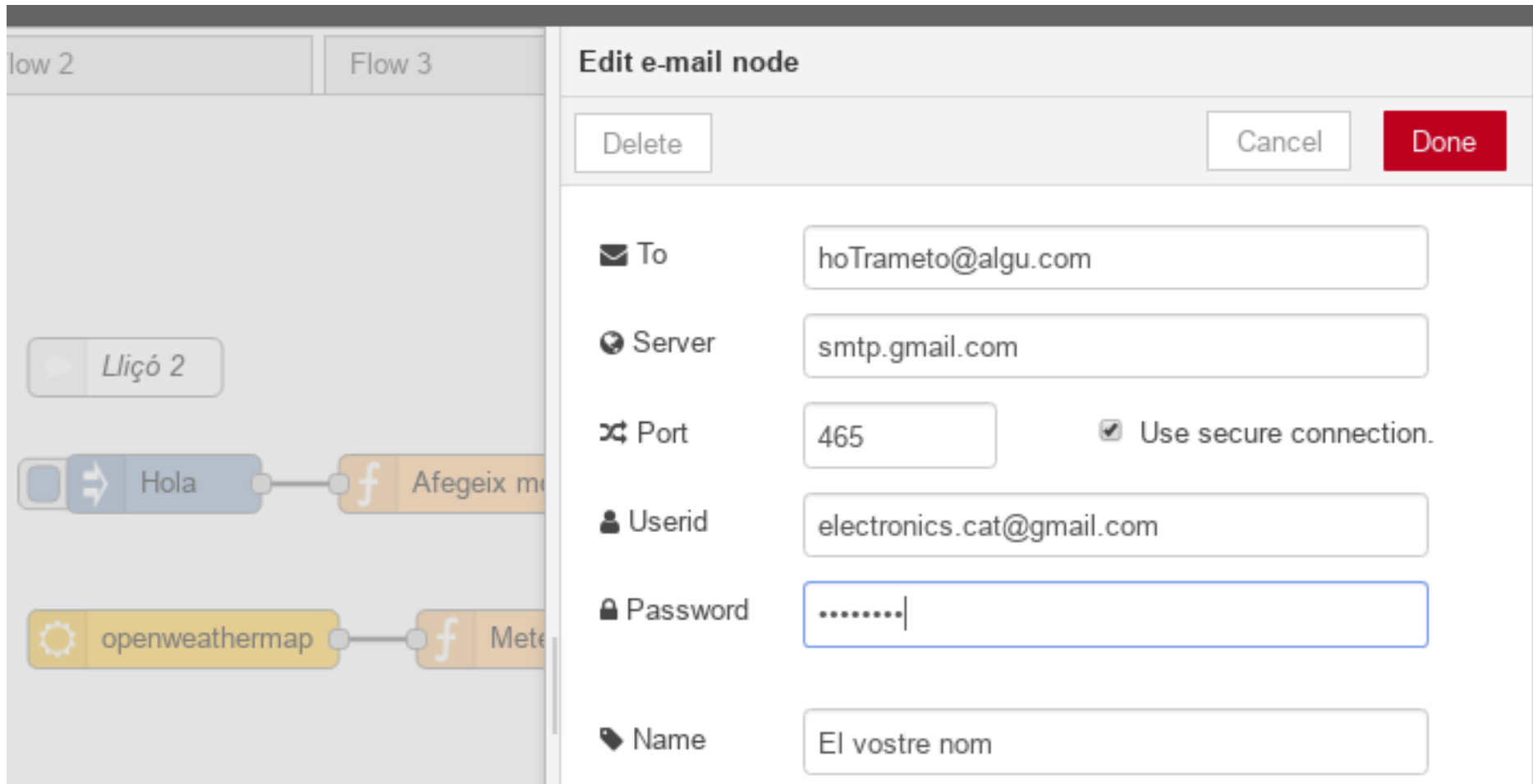
Function

```
1 - if (msg.payload.weather === "Clear") {
2     msg.payload = "Clear skies ahead today!";
3     return msg;
4 - }else{
5     msg.payload = msg.payload.weather;
6     return msg;
7 - }
8 return null;
9
```

Afegeix m...

 Meteo

Configuració del node de correu



The screenshot shows the Node-RED interface with the 'Edit e-mail node' configuration panel open. The panel includes the following fields and options:

- Delete** button
- Cancel** button
- Done** button
- To**: hoTrameto@algu.com
- Server**: smtp.gmail.com
- Port**: 465 Use secure connection.
- Userid**: electronics.cat@gmail.com
- Password**: [masked]
- Name**: El vostre nom

The background shows a flow with nodes: 'Lliçó 2', 'Hola', 'Afegix m...', 'openweathermap', and 'Mete...'.



Conèixer el temps i trametre-ho per correu

The screenshot shows the Node-RED interface with two flows and a debug console. The top bar shows 'Flow 1', 'Flow 2', 'Flow 3', and 'Lliçó 3'. The first flow, 'Lliçó 2', consists of a 'Hola' node, an 'Afegix món!' function node, and a 'msg.payload' output node. The second flow, 'Lliçó 3', consists of an 'openweathermap' node, a 'Meteo' function node, a 'msg.payload' output node, and an 'El vostre nom' node. The debug console shows two messages: one with payload 'Hola món!' and another with payload 'Clouds'.

```
27/3/2018, 19:19:05 node: 278d1e4.ce97ce2
msg.payload : string[9]
"Hola món!"

27/3/2018, 19:19:24 node: e7ad5d5b.2f381
msg.payload : string[8]
"Clouds"
```



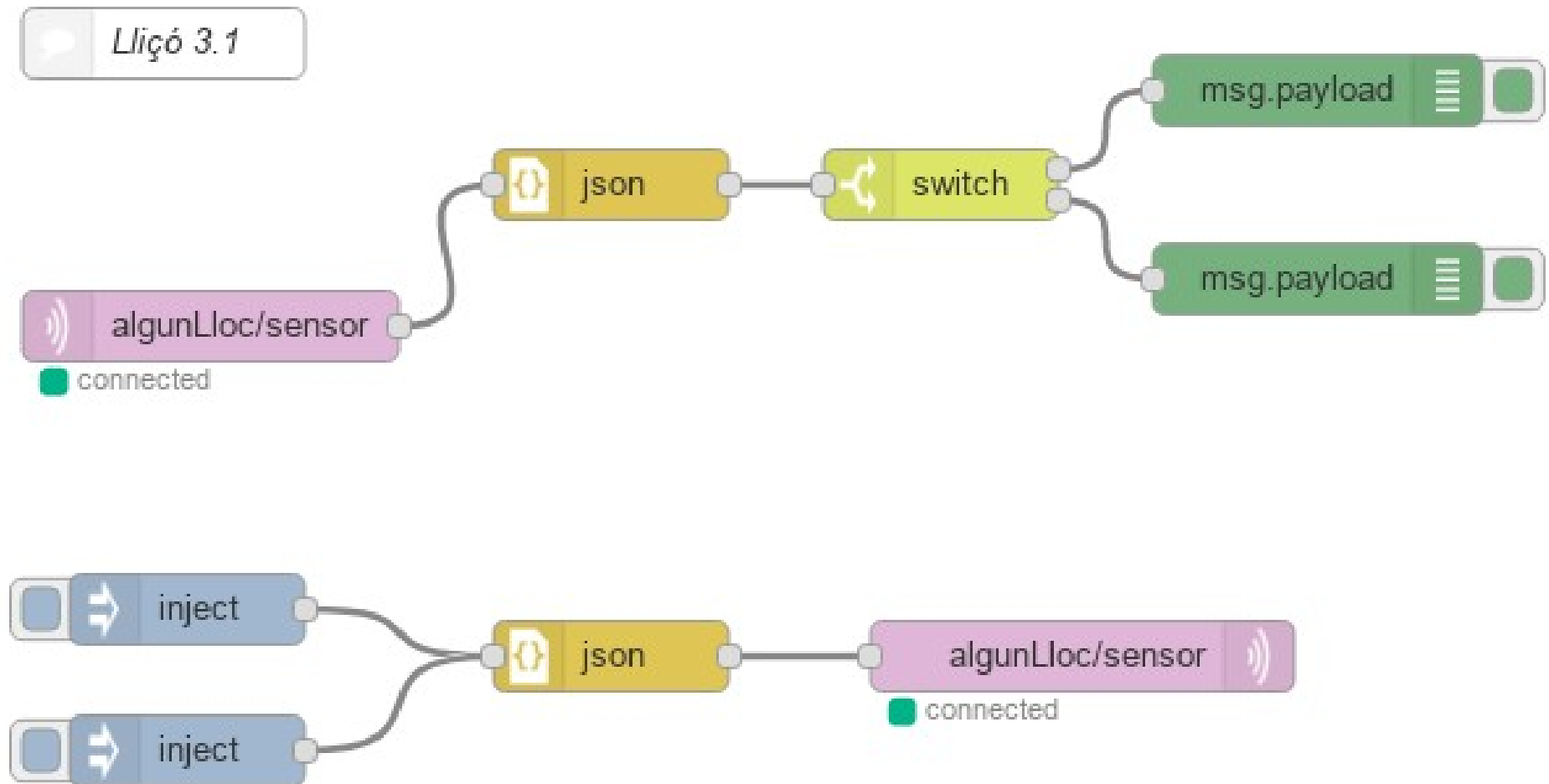
Node-RED



Conèixer el temps i trametre-ho per correu
Recepció del correu

The screenshot shows a Gmail interface. At the top, there is a Google search bar and a Gmail logo with a dropdown arrow. Below the search bar, there are navigation buttons: a back arrow, a plus sign, and an exclamation mark. A yellow tooltip points to the plus sign button with the text "Fes clic aquí per". Below the navigation buttons, there is a red button labeled "REDACTA". To the right of the redacted area, there is a section titled "Current Weather Information" with a gear icon and the email address "electronics.cat@gmail.com". Below the email address, there is a dropdown menu labeled "per a usuari" and the word "Clouds".

https://binefa.cat/IoT/nodeRed/02_nodeRed_01.txt



Node-RED

Tramesa de JSON a missatges MQTT Nodes d'injecció



Edit inject node

Delete Cancel Done

✉ Payload

Edit inject node

Delete Cancel Done

✉ Payload

Tramesa de JSON a missatges MQTT Node JSON



Edit json node

Delete Cancel Done

Name

El node JSON trameta un objecte JavaScript a partir de la cadena rebuda o trameta una cadena en format JSON si rep un objecte JavaScript.

Tramesa de JSON a missatges MQTT Node per a trametre MQTT



Edit mqtt out node

Delete Cancel Done

Server

Topic

QoS Retain

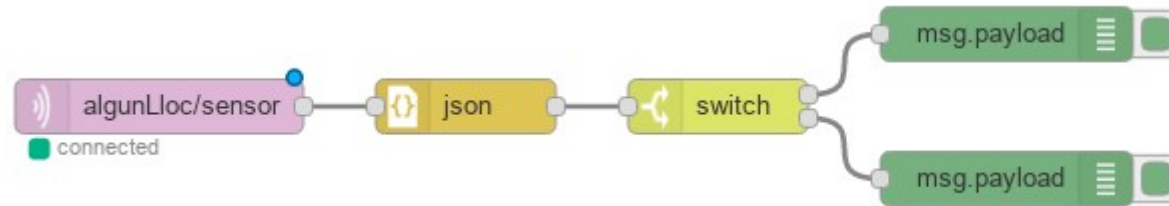
Name

Tip: Leave topic, qos or retain blank if you want to set them via msg properties.

El port habitual per a MQTT no segur és el 1883.
En aquest exemple es fa servir un mediador (*broker*) personalitzat al port 1888.

https://binefa.cat/IoT/nodeRed/03_nodeRed_01.txt

Recepció JSON de missatges MQTT Node per a rebre MQTT



Edit mqtt in node

Delete Cancel Done

Server

Topic

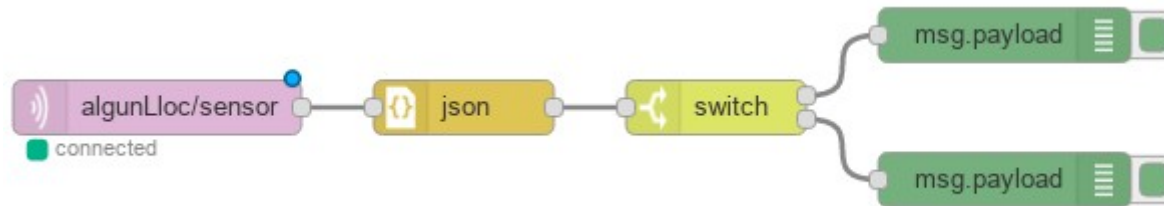
QoS

Name

El port habitual per a MQTT no segur és el 1883.
En aquest exemple es fa servir un mediador (*broker*) personalitzat al port 1888.

https://binefa.cat/IoT/nodeRed/03_nodeRed_01.txt

Recepció JSON de missatges MQTT Node JSON



Edit json node

Delete Cancel Done

Name

El node JSON tramet un objecte JavaScript a partir de la cadena rebuda o tramet una cadena en format JSON si rep un objecte JavaScript.

Recepció JSON de missatges MQTT Node de commutació (*switch*)

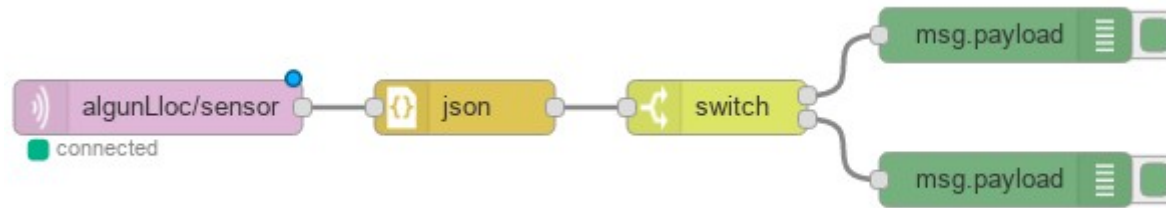


The image shows a Node-RED flow and the configuration panel for a switch node. The flow consists of three nodes: a purple MQTT sensor node labeled 'algunLloc/sensor' with a 'connected' status, a yellow JSON node labeled 'json', and a green switch node. The switch node is connected to two 'msg.payload' nodes. The configuration panel for the switch node is open, showing the following settings:

- Edit switch node** (Title)
- Delete** (Button)
- Cancel** (Button)
- Done** (Button)
- Name**: Name (Text input)
- Property**: msg.payload.bValor (Dropdown menu)
- Rules**:
 - Rule 1: is true → 1 (Dropdown menu, arrow, and close button)
 - Rule 2: is false → 2 (Dropdown menu, arrow, and close button)
- + add** (Button)
- checking all rules** (Dropdown menu)

https://binefa.cat/IoT/nodeRed/03_nodeRed_01.txt

Recepció JSON de missatges MQTT Node de depuració (*debug*)



Edit debug node

Delete Cancel Done

Output

to

Name

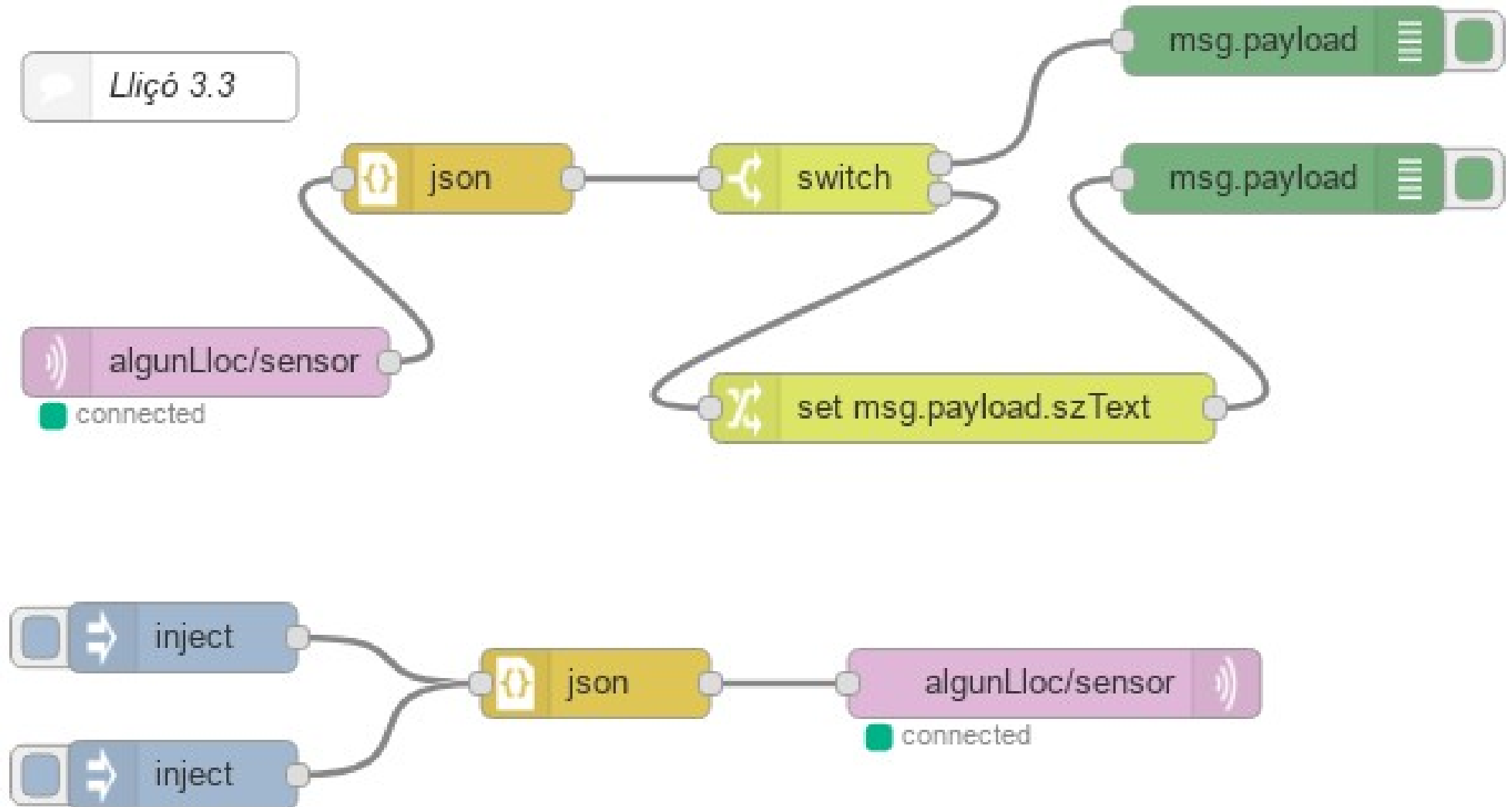
Deploy

info debug

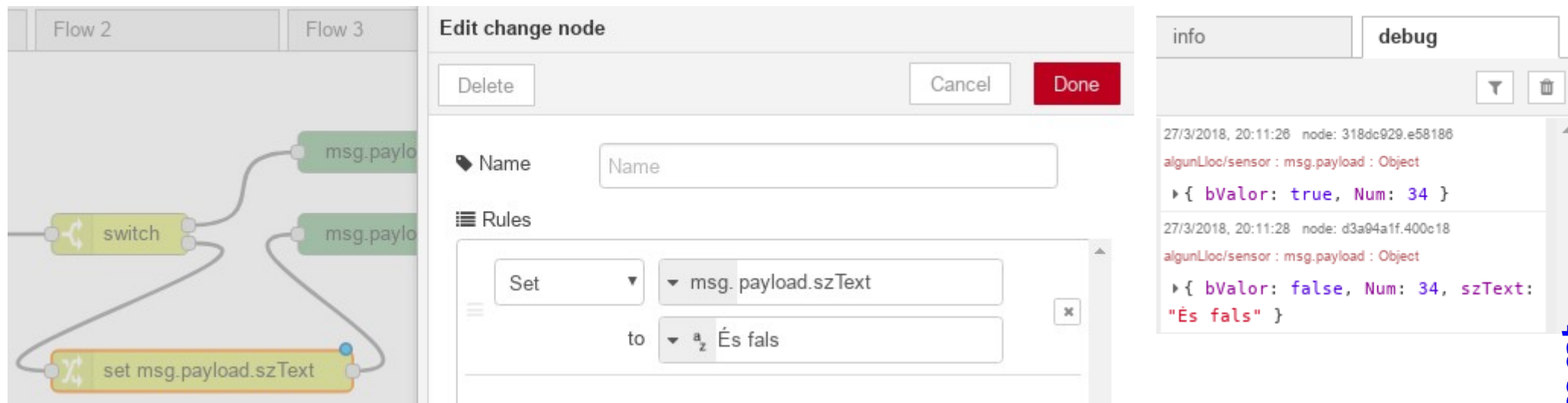
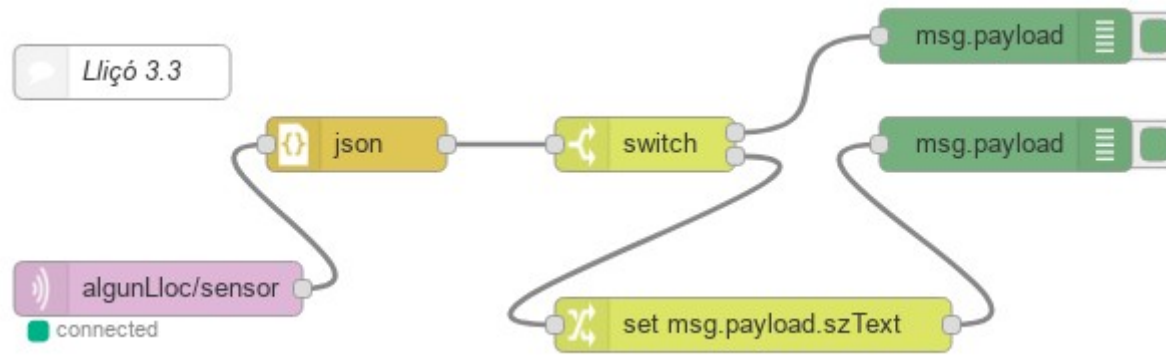
27/3/2018, 20:04:20 node: b19283fe.15de7
algunLloc/sensor : msg.payload : Object
▶ { bValor: true, Num: 34 }

27/3/2018, 20:04:23 node: 448583a8.548acc
algunLloc/sensor : msg.payload : Object
▶ { bValor: false, Num: 34 }

Recepció JSON de missatges MQTT Ús del node de canvi (*change*)



Recepció JSON de missatges MQTT Ús del node de canvi (*change*)



The screenshot shows the Node-RED interface. On the left, a flow diagram is visible with a 'switch' node and a 'set msg.payload.szText' node. The 'Edit change node' dialog is open, showing the 'Name' field and the 'Rules' section. The rule is configured as follows:

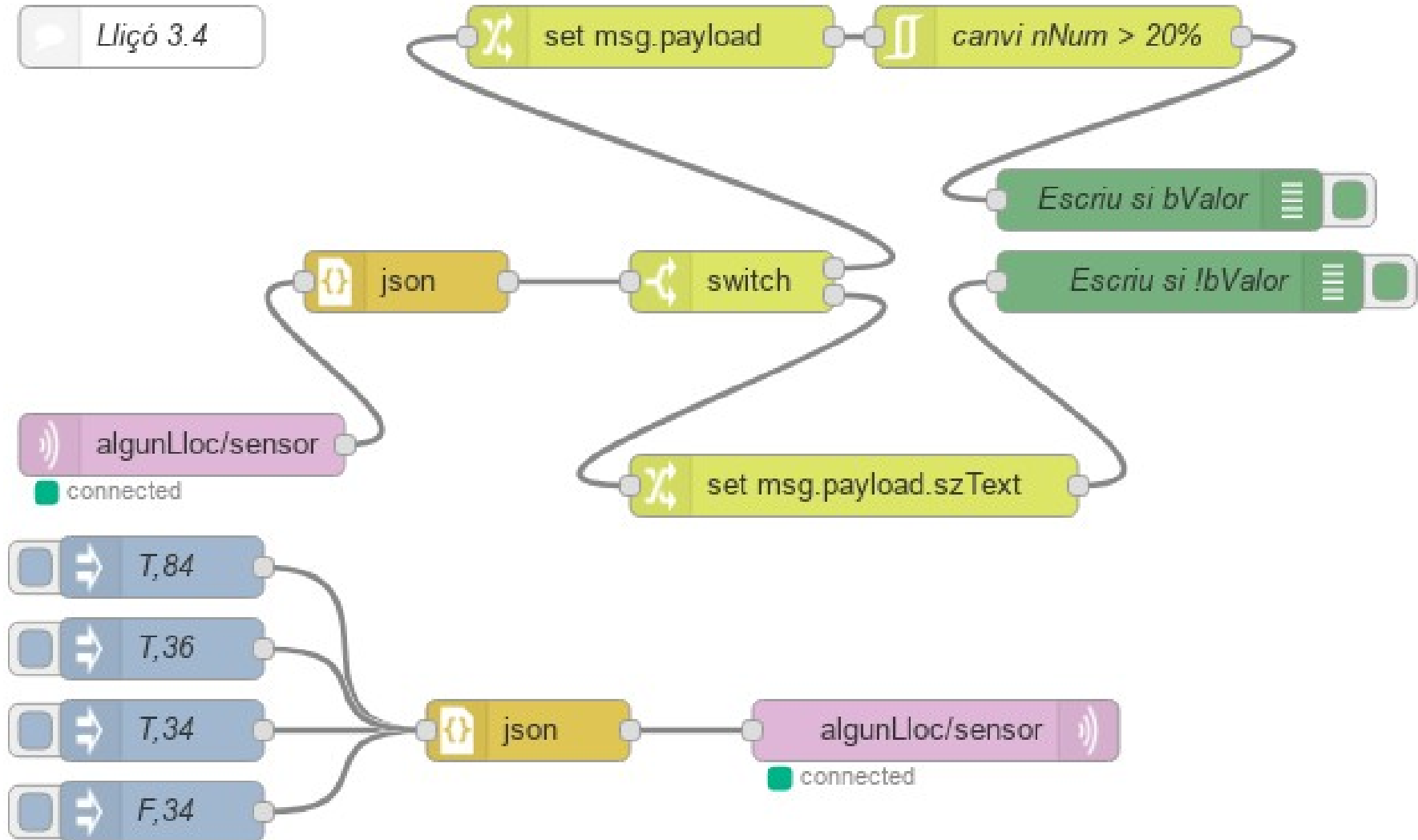
```
Set msg.payload.szText to a_2 És fals
```

The debug console on the right shows two messages:

```
27/3/2018, 20:11:26 node: 318dc929.e58186
algunLloc/sensor : msg.payload : Object
  ▶ { bValor: true, Num: 34 }

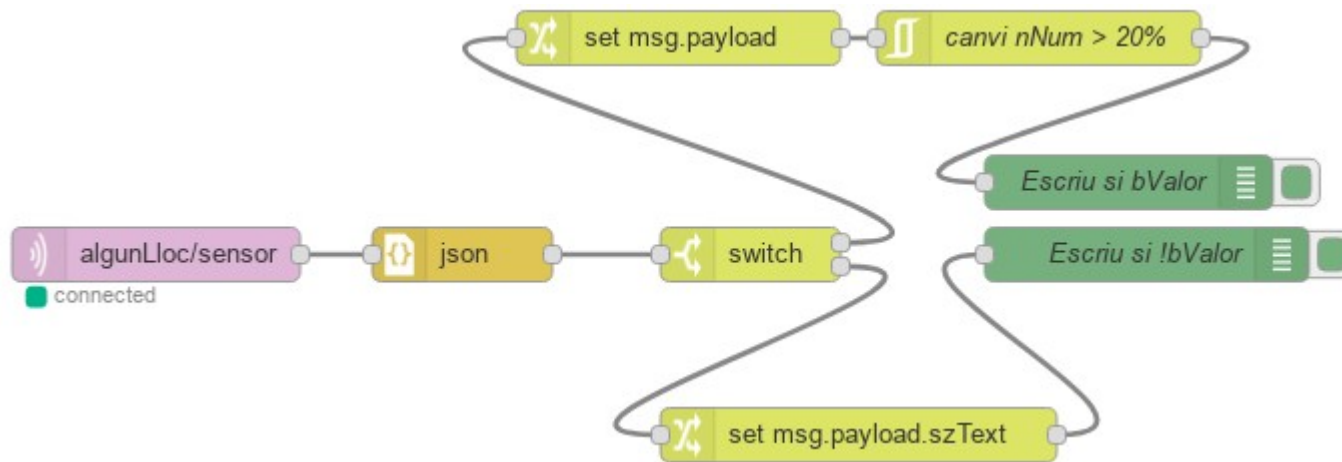
27/3/2018, 20:11:28 node: d3a94a1f.400c18
algunLloc/sensor : msg.payload : Object
  ▶ { bValor: false, Num: 34, szText: "És fals" }
```


Recepció JSON de missatges MQTT Ús del node rbe (*Report By Exception*)

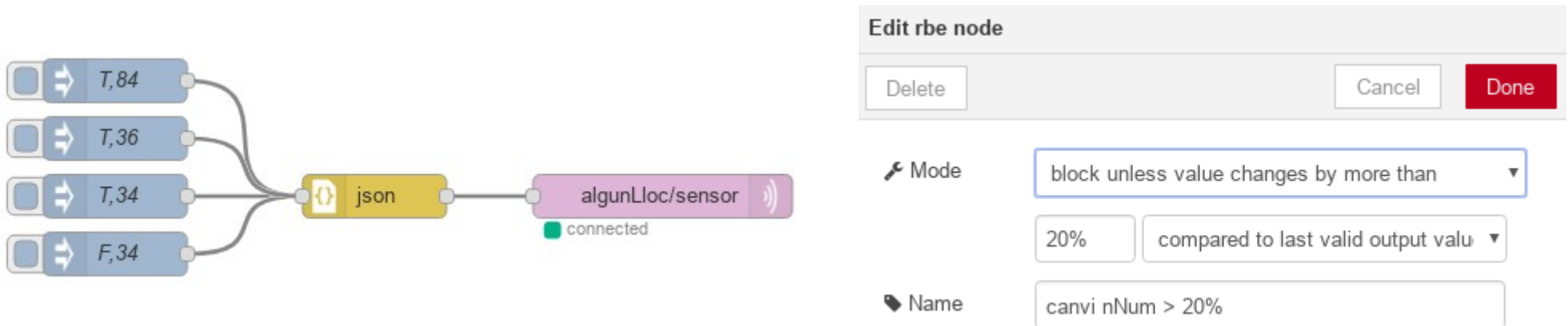


https://binefa.cat/IoT/nodeRed/03_nodeRed_04.txt

Recepció JSON de missatges MQTT Ús del node rbe (*Report By Exception*)

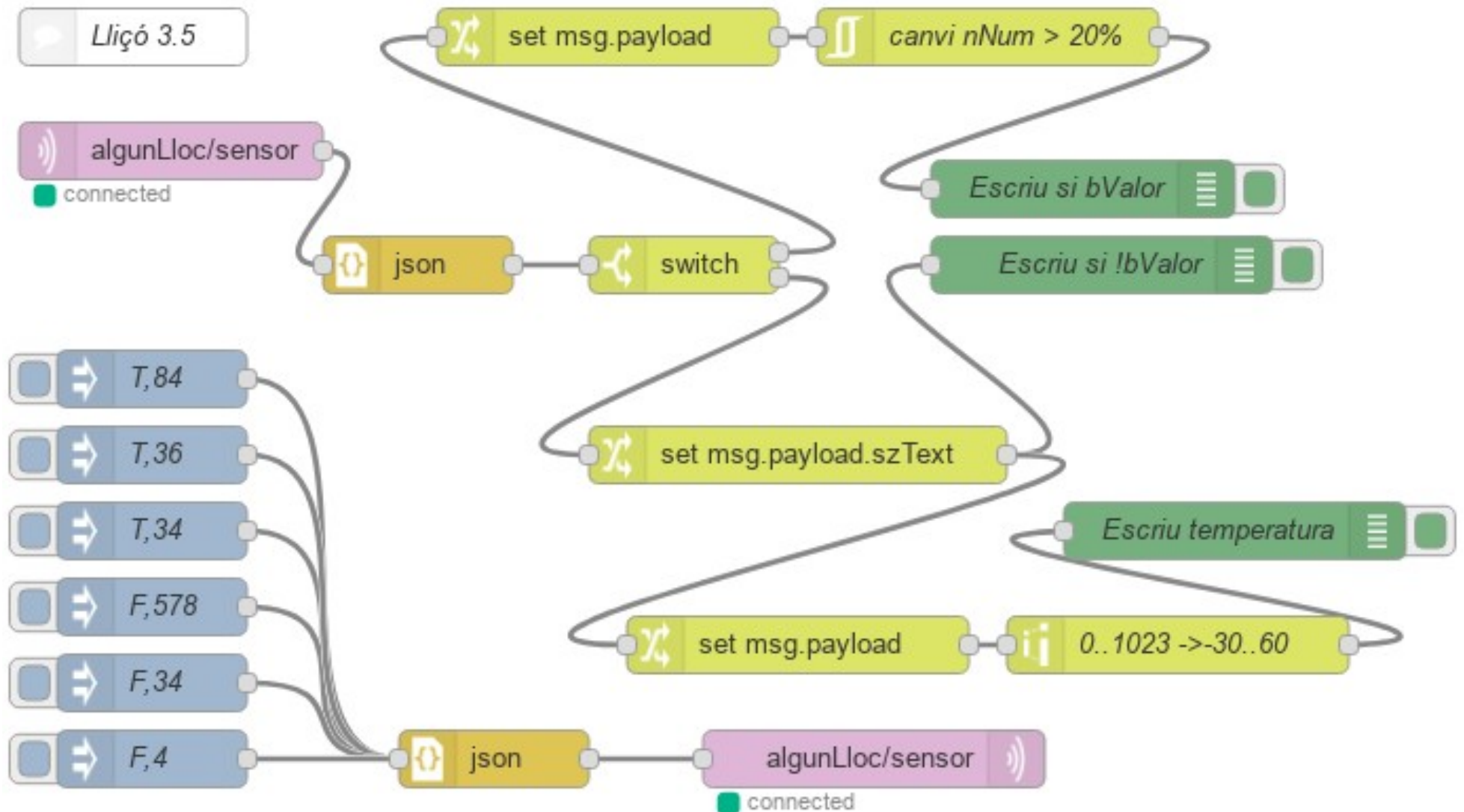


El node rbe trameta la informació que li entra si es compleix la condició. En aquest cas són variacions superiors al 20%.



https://binefa.cat/IoT/nodeRed/03_nodeRed_04.txt

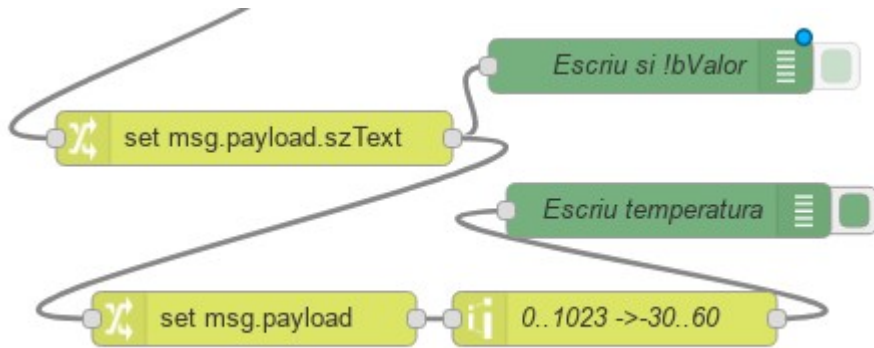
Recepció JSON de missatges MQTT Ús del node ventall (range)





Node-RED

Recepció JSON de missatges MQTT Ús del node ventall (range)



Edit range node

Delete Cancel **Done**

Action: Scale msg.payload

Map the input range:

from: to:

to the result range:

from: to:

Round result to the nearest integer?

Name:

Tip: This node ONLY works with numbers.

El node ventall transforma un ventall de valors en un altre. En aquest cas pot rebre un número entre 0 i 1023 i ho transforma proporcionalment a un valor entre -30 i +60.

- T,84
- T,36
- T,34
- F,578
- F,34
- F,4

https://binefa.cat/IoT/nodeRed/03_nodeRed_05.txt

Node-RED


Sòcols web (*websockets*)

Lliçó 3.7

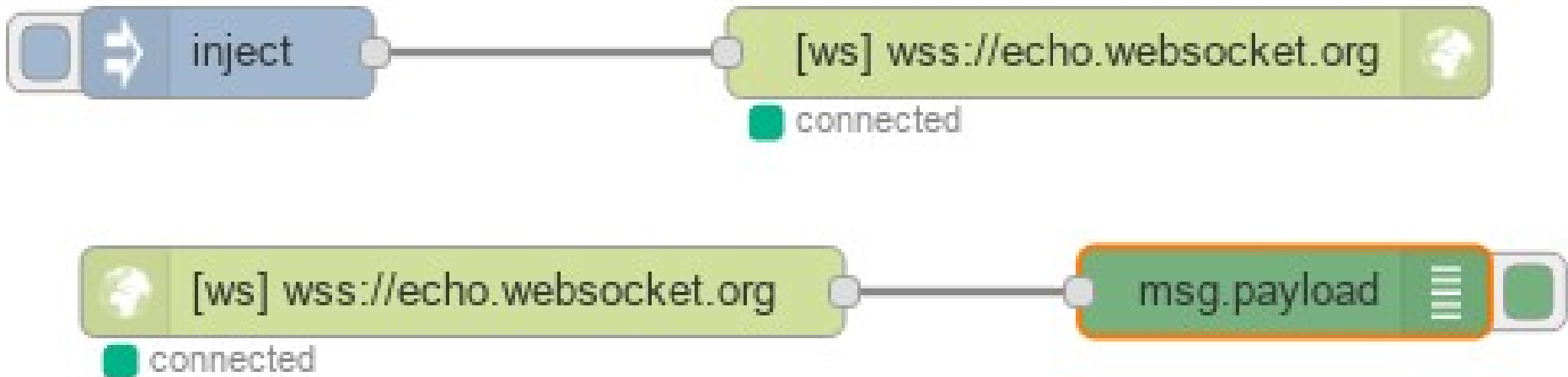
Edit websocket out node

Delete Cancel Done

Type

URL 

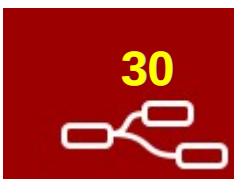
Name





Node-RED

Sòcols TCP



Servidor ESP32/ESP8266 i client Node-RED

Lliçó 3.8



```
/dev/ttyUSBO
```

```
.....  
WiFi connected with IP address:  
192.168.1.171  
new client  
client sent: Des del Node-RED
```

Edit inject node

Delete

✉ Payload

Edit tcp out node

Delete Cancel Done

Type port
at host

Close connection after each message is sent?
 Decode Base64 message?

Name

Codi Node-RED: https://binefa.cat/loT/nodeRed/03_nodeRed_08.txt

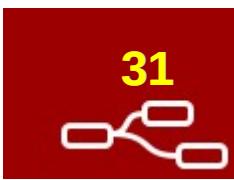
Codi d'Arduino IDE per a l'ESP32: https://binefa.cat/loT/esp32/esp32_tcpServer_00/esp32_tcpServer_00.ino

Codi d'Arduino IDE per a l'ESP8266: https://binefa.cat/loT/esp8266/esp8266_tcpServer_00/esp8266_tcpServer_00.ino

Font: <http://www.iotsharing.com/2017/05/tcp-udp-ip-with-esp32.html>



Node-RED



Sòcols TCP

Client ESP32/ESP8266 i servidor Node-RED

Lliçó 3.9

tcp:8088
0 connections

msg.payload

Edit tcp in node

Delete Cancel Done

Type Listen on port 8088

Output stream of String payload(s)
delimited by

Topic Topic

Name Name

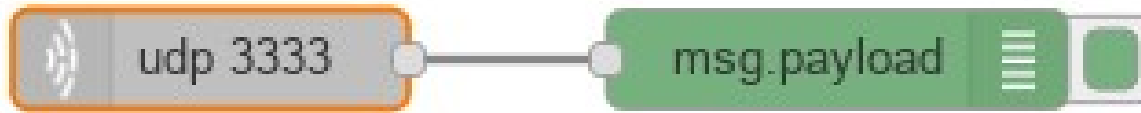
Codi Node-RED: https://binefa.cat/IoT/nodeRed/03_nodeRed_09.txt

Codi d'Arduino IDE per a l'ESP32: https://binefa.cat/IoT/esp32/esp32_tcpClient_00/esp32_tcpClient_00.ino

Codi d'Arduino IDE per a l'ESP8266: https://binefa.cat/IoT/esp8266/esp8266_tcpClient_00/esp8266_tcpClient_00.ino

Font: <http://www.iotsharing.com/2017/05/tcp-udp-ip-with-esp32.html>

Lliçó 3.10



Edit udp in node

Delete Cancel Done

Listen for

on Port using

Output

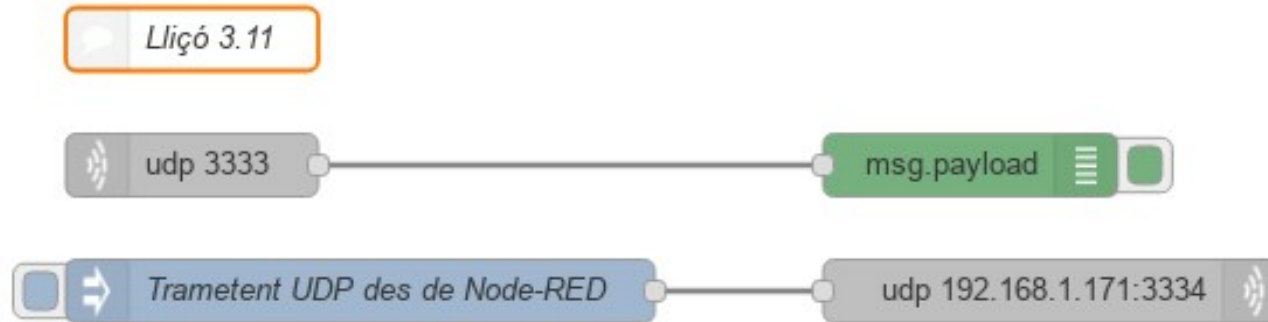
Name

Tip: Make sure your firewall will allow the data in.

Codi Node-RED: https://binefa.cat/IoT/nodeRed/03_nodeRed_10.txt

Codi d'Arduino IDE per a l'ESP32: https://binefa.cat/IoT/esp32/esp32_udpClient_00/esp32_udpClient_00.ino

Codi d'Arduino IDE per a l'ESP8266: https://binefa.cat/IoT/esp8266/esp8266_udpClient_01/esp8266_udpClient_01.ino



Edit udp out node

Delete Cancel Done

Send a to port

Address

Decode Base64 encoded payload?

Name

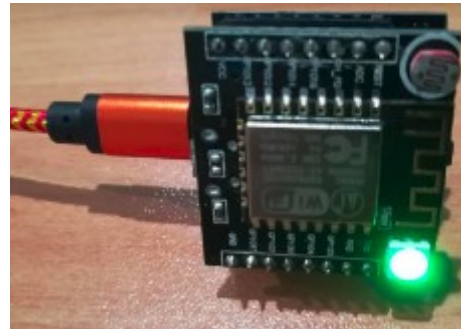
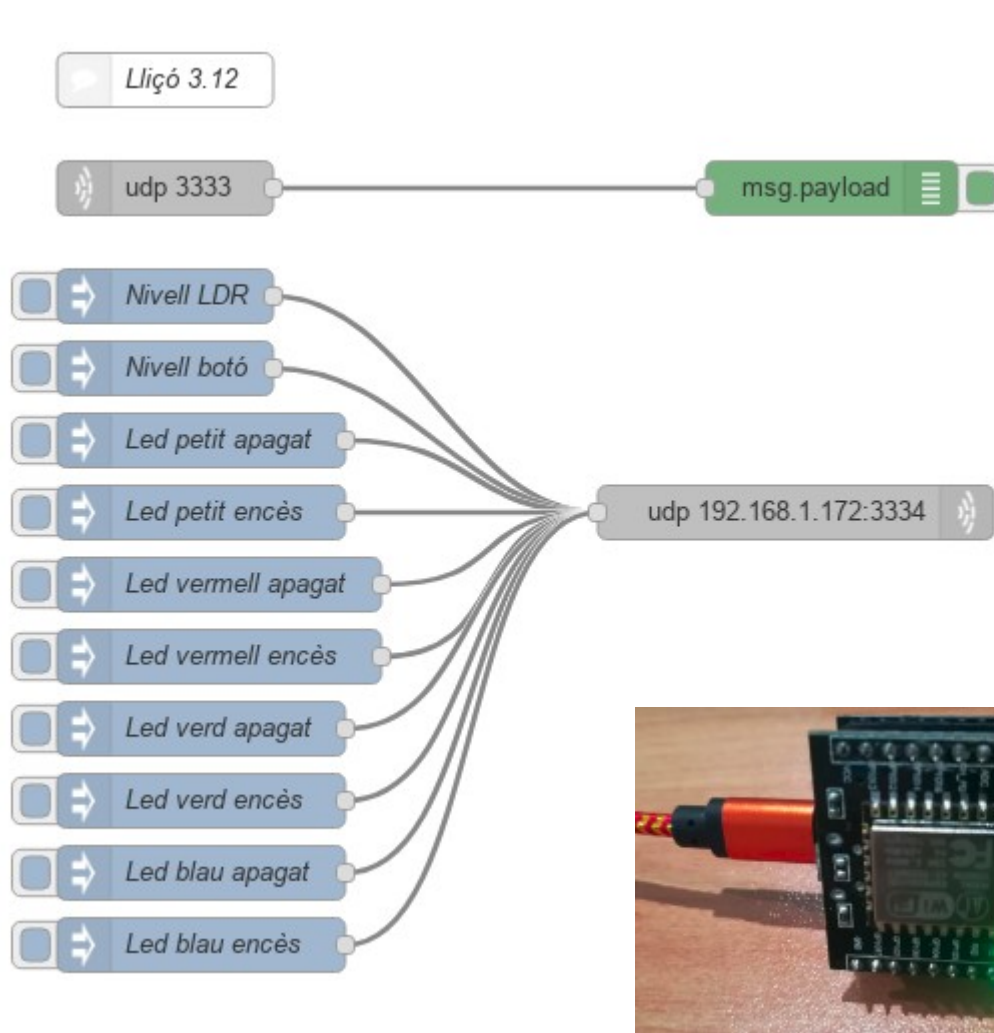
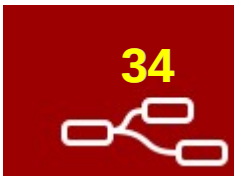
Tip: leave address and port blank if you want to set using `msg.ip` and `msg.port`.



Node-RED

Sòcols UDP

Pseudoservidor ESP32/ESP8266 i client Node-RED



Edit inject node

Delete Cancel Done

✉ Payload

☰ Topic

🔄 Repeat

Inject once at start?

📌 Name

```

if (szMsg == "12l" || szMsg == "13L") {
  digitalWrite(GPI012, LOW);
  Serial.println("GPI012 LOW -> Green OFF");
  bUnderstood = true;
  vSendUdp("OK", udpPortTx);
}
if (szMsg == "15h" || szMsg == "15H") {
  digitalWrite(GPI015, HIGH);
  Serial.println("GPI015 HIGH -> Red ON");
  bUnderstood = true;
  vSendUdp("OK", udpPortTx);
}
if (szMsg == "15l" || szMsg == "15L") {
  digitalWrite(GPI015, LOW);
  Serial.println("GPI015 LOW -> Red OFF");
  bUnderstood = true;
  vSendUdp("OK", udpPortTx);
}

```

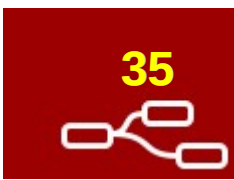
Codi Node-RED: https://binefa.cat/IoT/nodeRed/03_nodeRed_12.txt

Codi d'Arduino IDE per a l'ESP8266: https://binefa.cat/IoT/esp8266/esp8266_udpServer_01/esp8266_udpServer_01.ino



Node-RED

LoRaWAN



Nodes 'ttn device' i 'ttn message'

websocket

tcp

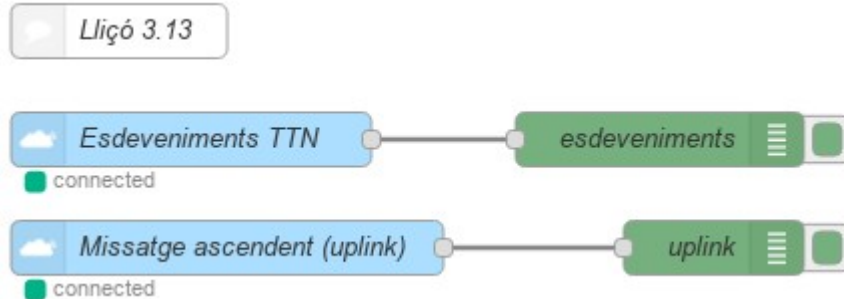
udp

ttn device

ttn message

output

debug



Edit debug node

Delete Cancel Done

Output complete msg object

to debug tab

Name esdeveniments

Edit ttn device node

Delete Cancel Done

Name Esdeveniments TTN

App formacio-ttn

Device ID

Event activations

Edit ttn message node

Delete Cancel Done

Name Missatge ascendent (uplink)

App formacio-ttn

Device ID

Field

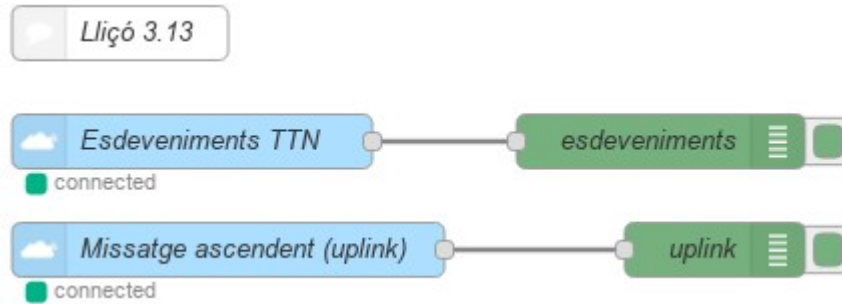
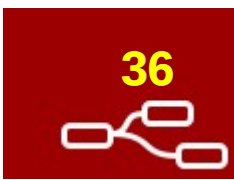
Codi Node-RED: https://binefa.cat/loT/nodeRed/03_nodeRed_13.txt

Codi d'Arduino IDE per a l'ESP32: https://binefa.cat/loT/esp32/lorawan_ttnEsp32_rfm95_01_multichannel-llico_3_13.tar.gz



Node-RED

LoRaWAN App ID



Applications > formacio-ttn

Overview

ttn device > Edit ttn app node

Delete

Cancel

Update

App ID

Region or Broker

Access Key

APPLICATION OVERVIEW

Application ID **formacio-ttn**

Description Formació sobre The Things Network

Created 7 days ago

Handler ttn-handler-eu (current handler)

Codi Node-RED: https://binefa.cat/loT/nodeRed/03_nodeRed_13.txt

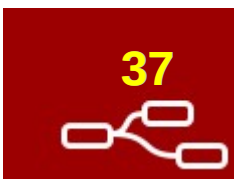
Codi d'Arduino IDE per a l'ESP32: https://binefa.cat/loT/esp32/lorawan_ttnEsp32_rfm95_01_multichannel-llico_3_13.tar.gz



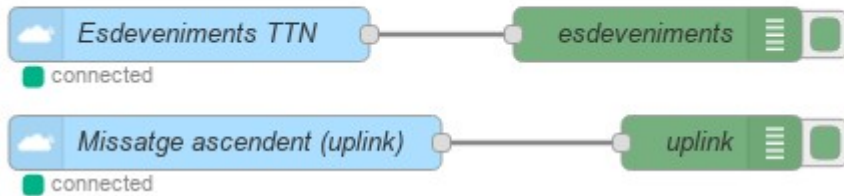
Node-RED

LoRaWAN

Access Key



Lliçó 3.13



ttn device > Edit ttn app node

Delete Cancel Update

App ID formacio-ttn

Region or Broker eu

Access Key

ACCESS KEYS

manage keys

default key

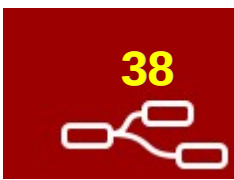
devices messages

..... base64

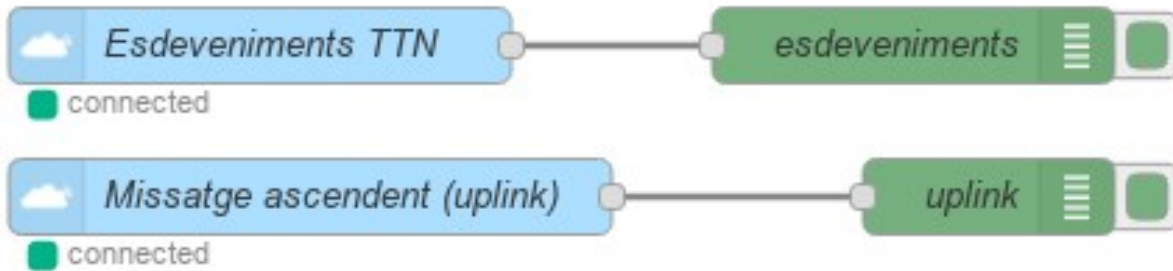


Node-RED

LoRaWAN



Lliçó 3.13



```

/dev/ttyUSBO

[RFM95] Packet queued
160030409: [RFM95] EV_TXCOMPLETE (includes waiting for RX windows)
TXRX_ACK confirmed UP frame was acked

TX complete .....
0 0

[RFM95] Packet queued
161409360: [RFM95] EV_TXCOMPLETE (includes waiting for RX windows)
TXRX_ACK confirmed UP frame was acked

TX complete .....
0 0

 Desplaçament automàtic
Ambdós NL & CR

```

```

29/3/2018, 20:59:56 node: uplink
msg: Object
▶ { app_id: "formacio-ttn", dev_id: "ttgo-01", hardware_serial:
"6543617430303031", port: 1, counter: 38 _ }

29/3/2018, 21:00:58 node: uplink
msg: Object
▼ object
  app_id: "formacio-ttn"
  dev_id: "ttgo-01"
  hardware_serial: "6543617430303031"
  port: 1
  counter: 39
▼ payload_raw: buffer[4] raw
  0: 0
  1: 0
  2: 0
  3: 39
▼ metadata: object
  time: "2018-03-29T19:00:58.270307023Z"
  frequency: 867.9
  modulation: "LORA"
  data_rate: "SF7BW125"
  airtime: 51456000
  coding_rate: "4/5"
▼ gateways: array[1]
  ▼ 0: object
    gtw_id: "eui-b827ebfffe875de4"
    timestamp: 4290676260
    time: "2018-03-29T19:00:58.247289Z"
    channel: 7
    rssi: -48
    snr: 9.5

```

Codi Node-RED: https://binefa.cat/loT/nodeRed/03_nodeRed_13.txt

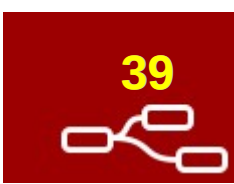
Codi d'Arduino IDE per a l'ESP32: https://binefa.cat/loT/esp32/lorawan_ttnEsp32_rfm95_01_multichannel-llico_3_13.tar.gz



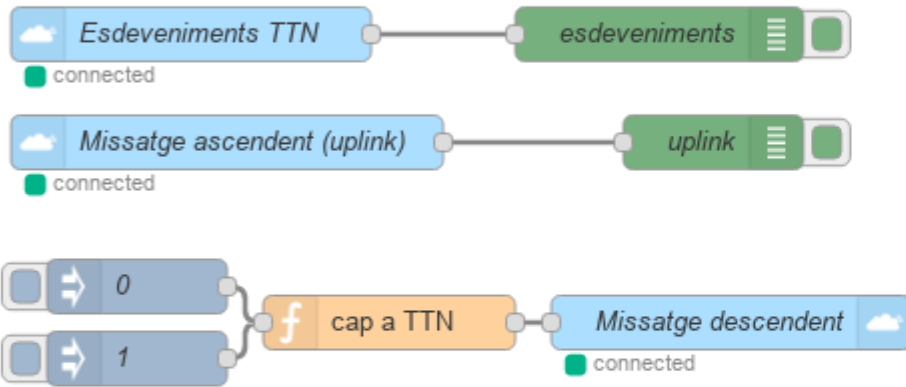
Node-RED

LoRaWAN

Missatge descendent (*downLink*)



Lliçó 3.14



```

29/3/2018, 21:30:28 node: uplink
msg : Object
  > { app_id: "formacio-ttn", dev_id: "ttgo-01", hardware_serial:
    "6543617430303031", port: 1, counter: 74 ... }

29/3/2018, 21:30:42 node: uplink
msg : Object
  > { app_id: "formacio-ttn", dev_id: "ttgo-01", hardware_serial:
    "6543617430303031", port: 1, counter: 71 ... }
  
```

/dev/ttyUSBO

```

TX complete .....
1 1
51
[RFM95] Received 1 bytes of payload
51
[RFM95] Packet queued
252489096: [RFM95] EV_TXCOMPLETE (includes waiting for RX windows)
TXRX_ACK confirmed UP frame was acked

TX complete .....
1 1
71
[RFM95] Received 1 bytes of payload
71
  
```

Desplaçament automàtic Ambdós NL & CR

```

return {
  dev_id: msg.dev_id,
  port: msg.port,
  schedule: "replace",
  confirmed: false,
  payload: Buffer.from([msg.payload ? 0x71 : 0x51])
}
  
```

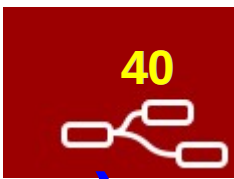
0101 0000	80	50	P	0111 0000	112	70	p
0101 0001	81	51	Q	0111 0001	113	71	q
0101 0010	82	52	R	0111 0010	114	72	r

Codi Node-RED: https://binefa.cat/loT/nodeRed/03_nodeRed_14.txt

Codi d'Arduino IDE per a l'ESP32: https://binefa.cat/loT/esp32/lorawan_ttnEsp32_rfm95_01_multichannel-llico_3_13.tar.gz



Node-RED



LoRaWAN / Missatge descendent (*downlink*)



Applications > formacio-ttn > Devices > ttgo-01 > Data

APPLICATION DATA

Filters: uplink downlink activation ack error

	time	counter	port	
▲	21:34:49	79	1	payload: 0000 00 4E
▲	21:33:47	78	1	payload: 0000 00 4D
▲	21:32:45	77	1	payload: 0000 00 4C
▲	21:31:43	76	1	payload: 0000 00 4B
▼	21:30:42		1	payload: 71
▲	21:30:42	75	1	payload: 0000 00 4A
▼	21:30:39		1	<i>scheduled</i> payload: 71
▼	21:30:26		1	payload: 51
▲	21:30:26	74	1	payload: 0000 00 49
▼	21:30:17		1	<i>scheduled</i> payload: 51
▲	21:29:45	73	1	payload: 0000 00 48

Codi Node-RED: https://binefa.cat/loT/nodeRed/03_nodeRed_14.txt

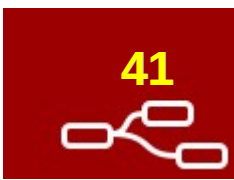
Codi d'Arduino IDE per a l'ESP32: https://binefa.cat/loT/esp32/lorawan_ttnEsp32_rfm95_01_multichannel-llico_3_13.tar.gz



Node-RED

LoRaWAN

Missatge descendent (downlink)



Edit function node

Delete Cancel Done

Name

Function

```

1 - return {
2   dev_id: msg.dev_id,
3   port: msg.port,
4   schedule: "replace",
5   confirmed: false,
6   payload: Buffer.from([msg.payload ? 0x71 : 0x51])
7 - }
  
```

Edit inject node

Delete Cancel Done

Payload

Topic

Repeat

Inject once at start?

Name

0101 0000	80	50	P	0111 0000	112	70	p
0101 0001	81	51	Q	0111 0001	113	71	q
0101 0010	82	52	R	0111 0010	114	72	r

DEVICE OVERVIEW

Application ID formacio-ttn

Device ID ttgo-01

Description Placa TTGO-01

Activation Method ABP

Edit ttn send node

Delete Cancel Done

Name

App

Device ID

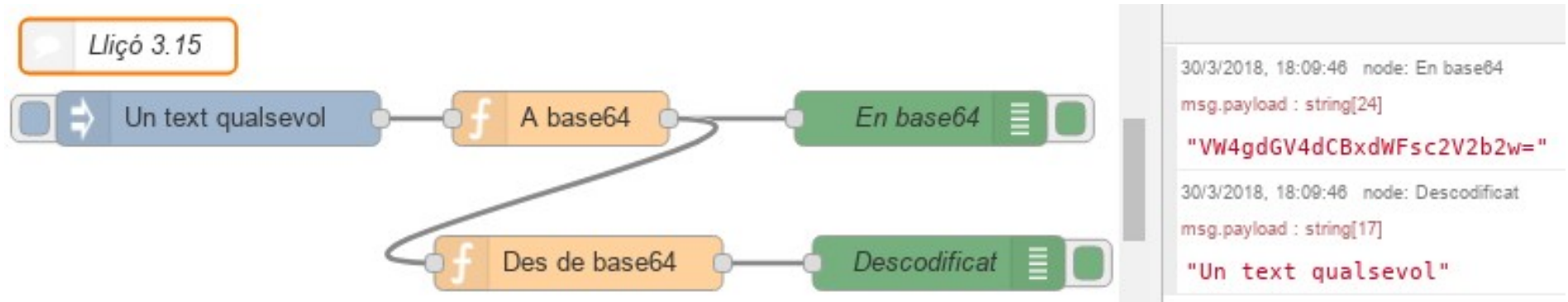
Port

Codi Node-RED: https://binefa.cat/loT/nodeRed/03_nodeRed_14.txt

Codi d'Arduino IDE per a l'ESP32: https://binefa.cat/loT/esp32/lorawan_ttnEsp32_rfm95_01_multichannel-llico_3_13.tar.gz

Node-RED

Codificació i descodificació en base64



Edit function node

Delete Cancel Done

Name A base64

Function

```
1 var b = new Buffer(msg.payload);
2 msg.payload = b.toString('base64');
3
4 return msg;
```

Edit function node

Delete Cancel Done

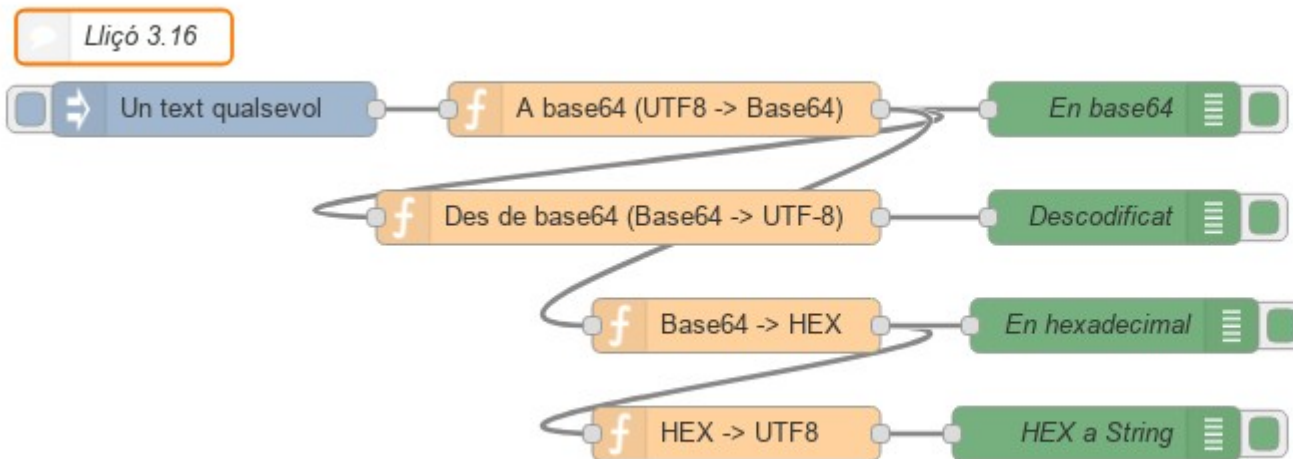
Name Des de base64

Function

```
1 var b = new Buffer(msg.payload, 'base64');
2 msg.payload = b.toString();
3
4 return msg;
```

Codificació i descodificació en base64

Presentació en hexadecimal



```

30/3/2018, 18:35:45 node: En base64
msg.payload : string[24]
"VW4gdGV4dCBxdWFsc2V2b2w="

30/3/2018, 18:35:45 node: Descodificat
msg.payload : string[17]
"Un text qualsevol"

30/3/2018, 18:35:45 node: HEX a String
msg.payload : string[17]
"Un text qualsevol"

30/3/2018, 18:35:45 node: En hexadecimal
msg.payload : string[34]
"556e20746557874207175616c7365576666c"
  
```

Edit function node

Delete Cancel Done

Name Base64 -> HEX

Function

```

1 var b = new Buffer(msg.payload, 'base64')
2 msg.payload = b.toString('hex');
3
4 return msg;
  
```

Edit function node

Delete Cancel Done

Name HEX -> UTF8

Function

```

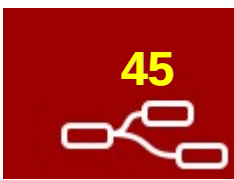
1 var b = new Buffer(msg.payload, 'hex')
2 msg.payload = b.toString('utf8');
3
4 return msg;
  
```




Node-RED

LoRaWAN

TTN fent servir MQTT



Edit mqtt in node

Delete Cancel Done

Server: eu.thethings.network:1883

Topic: formacio-ttn/devices/ttgo-01/up

QoS: 2

Name: Name

Edit mqtt out node

Delete Cancel Done

Server: eu.thethings.network:1883

Topic: formacio-ttn/devices/ttgo-01/down

QoS: Retain

Name: Name

Tip: Leave topic, qos or retain blank if you want to set them via msg properties.

mqtt in > **Edit mqtt-broker node**

Delete Cancel Update

Connection Security Birth Message Will Message

Server: eu.thethings.network Port: 1883

Enable secure (SSL/TLS) connection

Client ID: Leave blank for auto generated

Keep alive time (s): 60 Use clean session

Use legacy MQTT 3.1 support

mqtt in > **Edit mqtt-broker node**

Delete Cancel Update

Connection Security Birth Message Will Message

Username: formacio-ttn

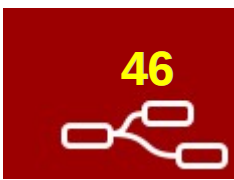
Password:



Node-RED

LoRaWAN

TTN fent servir MQTT



Edit function node

Delete Cancel Done

Name Repartidor

Function

```
1 var keys = Object.keys(msg.payload);
2
3 var msgs = keys.map(function(key) {
4   return { topic: key, payload: msg.payload[key] };
5 });
6 return [msgs];
```

Edit function node

Delete Cancel Done

Name (UTF8 -> hex -> Base64)

Function

```
1 var b = new Buffer(msg.payload);
2 msg.payload = b.toString('base64');
3
4 return msg;
```

Edit switch node

Delete Cancel Done

Name

Property msg.topic

- == a_z hardware_serial → 1 x
- == a_z port → 2 x
- == a_z counter → 3 x
- == a_z payload_raw → 4 x
- == a_z metadata → 5 x

Edit function node

Delete Cancel Done

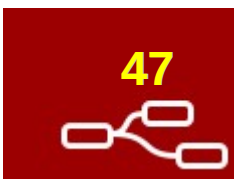
Name {"payload_raw": msg.payload }

Function

```
1 msg.payload = { "payload_raw": msg.payload };
2
3 return msg;
```

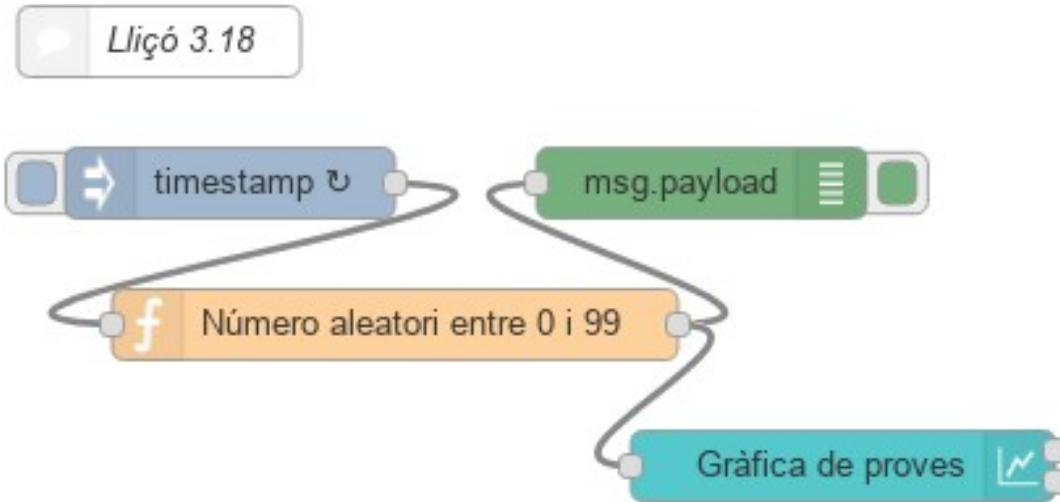



Node-RED



Panel·ls de control (dashboards) Visualització de n·umeros aleatoris a una gr·afica

```
jordi@debianJB:~/node-red$ npm i node-red-dashboard
```

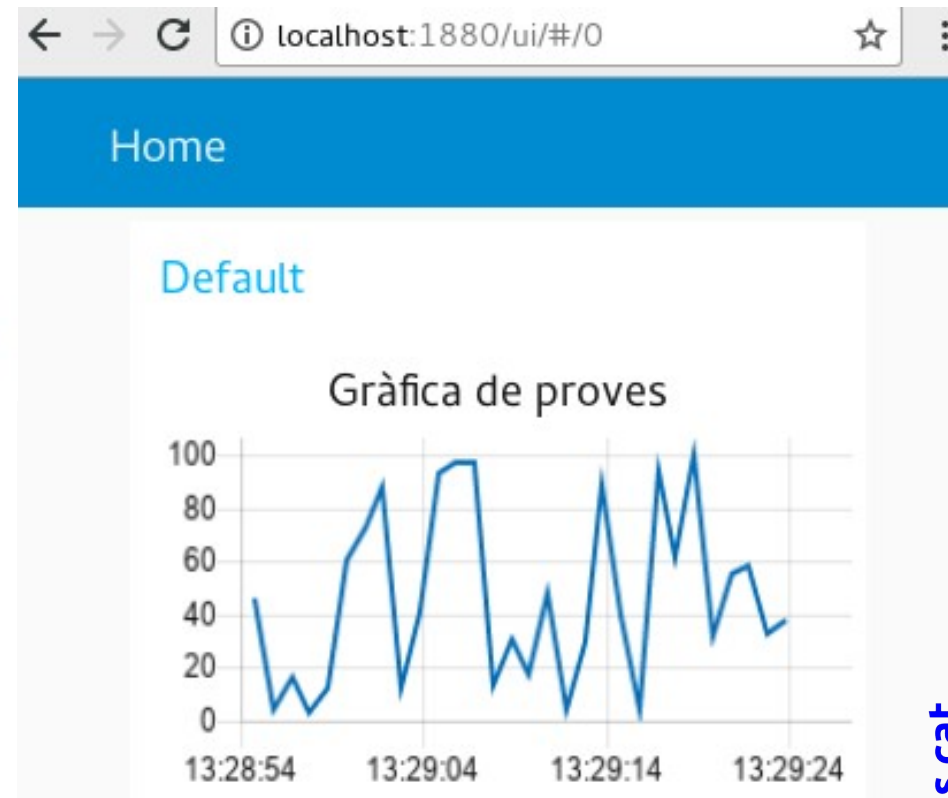


Edit function node

Delete Cancel Done

Name: N·umero aleatori entre 0 i 99

```
1 msg.payload = Math.round(Math.random()*100);  
2  
3 return msg;
```

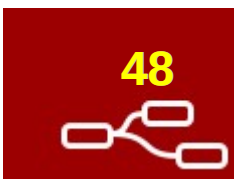


Codi Node-RED: https://binefa.cat/loT/nodeRed/03_nodeRed_18.txt

Font: <http://developers.sensetecnic.com/article/a-node-red-dashboard-using-node-red-dashboard/>



Node-RED



Panel·ls de control (dashboards) Visualització de n·umeros aleatoris a una gr·afica

The image shows the Node-RED interface with several panels open. On the left, the 'Edit chart node' panel is active, showing configuration for a 'Line chart' node. The chart is titled 'Gr·afica de proves' and is part of the 'Default [Home]' dashboard. The X-axis is labeled 'HH:mm:ss' and shows a time range from 13:28:54 to 13:29:24. The Y-axis ranges from 0 to 100. The chart displays a fluctuating blue line representing random data points.

In the center, the 'Edit inject node' panel is open, showing configuration for an inject node. The payload is set to 'timestamp', the topic is empty, and the repeat interval is set to '1 seconds'. The 'Inject once at start?' checkbox is unchecked.

At the bottom, a flow diagram shows the following nodes: a 'timestamp' node, a 'msg.payload' node, a 'Número aleatori entre 0 i 99' node, and a 'Gr·afica de proves' chart node. The flow is: timestamp -> msg.payload -> Número aleatori entre 0 i 99 -> Gr·afica de proves.

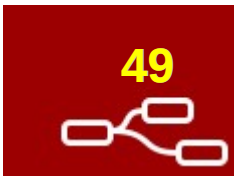
On the right, a browser window shows the dashboard at localhost:1880/ui/#/0. The dashboard has a blue header 'Home' and a chart titled 'Gr·afica de proves' with the same data as the 'Edit chart node' panel.

Codi Node-RED: https://binefa.cat/loT/nodeRed/03_nodeRed_18.txt

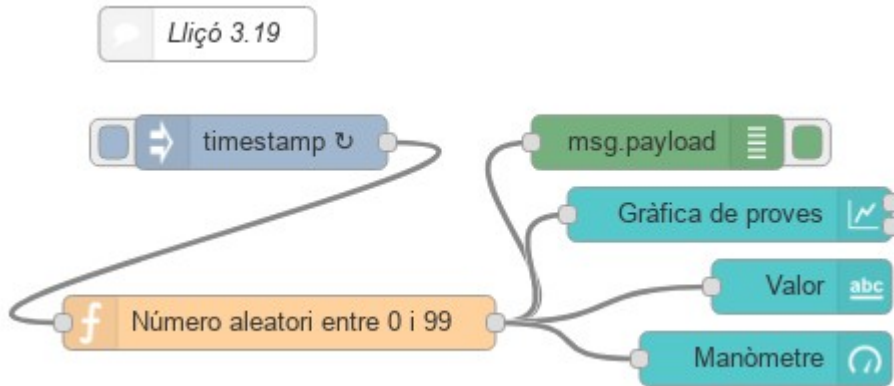
Font: <http://developers.sensetecnic.com/article/a-node-red-dashboard-using-node-red-dashboard/>



Node-RED



Panells de control (*dashboards*) Visualització a una gràfica, manòmetre i text



Edit text node

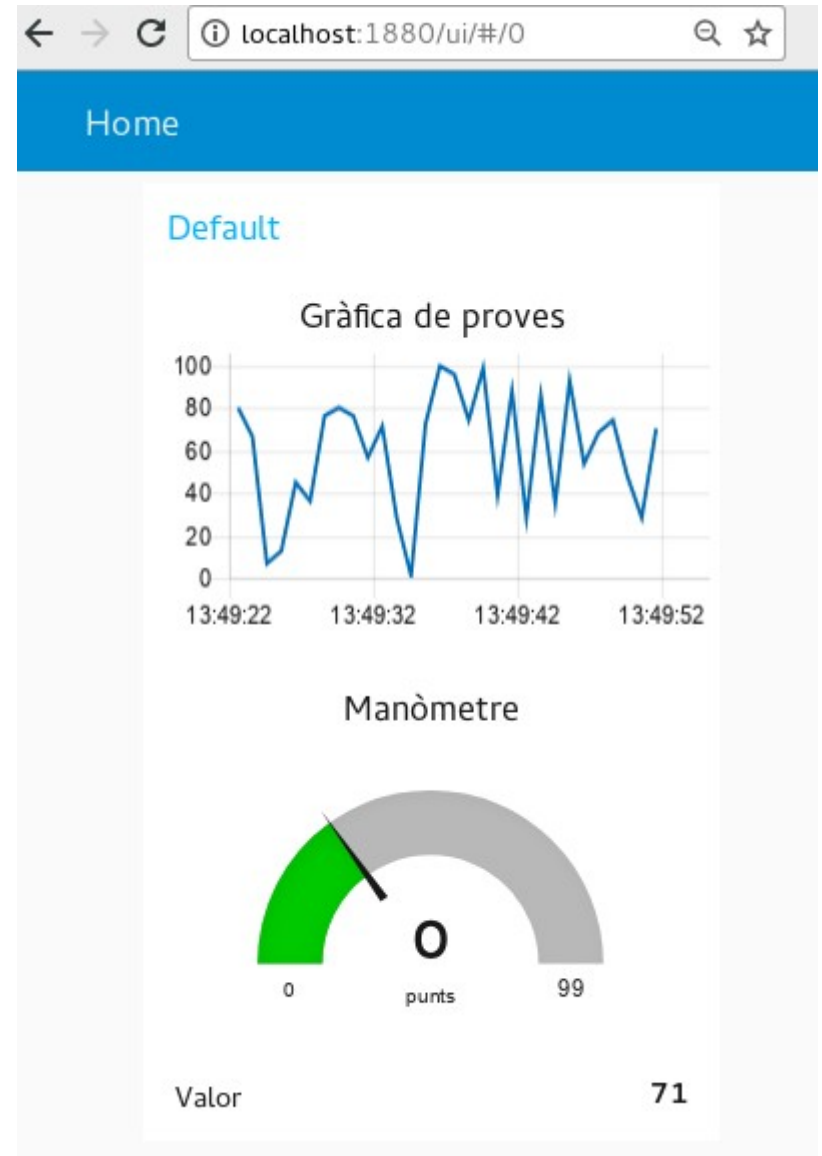
Delete Cancel Done

Group: Default [Home]

Size: auto

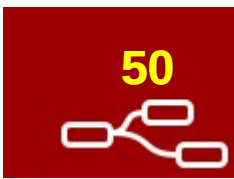
Label: Valor

Value format: {{msg.payload}}





Node-RED



Panel·ls de control (dashboards) Visualització a una gràfica, manòmetre i text

Edit gauge node

Delete Cancel Done

Group Default [Home]

Size auto

Type Gauge

Label Manòmetre

Value format {{msg.value}}

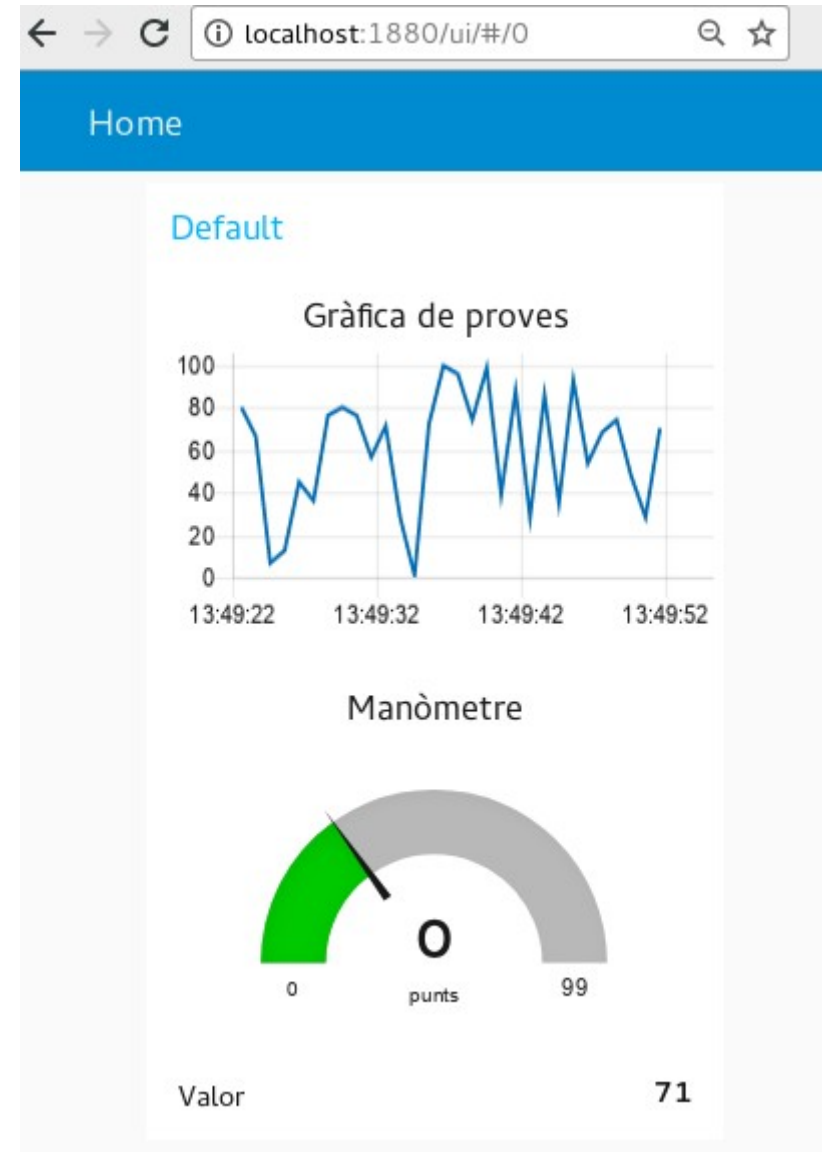
Units punts

Range min 0 max 99


Colour gradient

Sectors 0 ... 30 ... 75 ... 99

Name



Panel·ls de control (*dashboards*)



The image displays the Node-RED interface. At the top, there are tabs for 'Lliçó 3', 'info', 'debug', and 'dashboard'. The main workspace shows a flow with the following components:

- Lliçó 3.20**: A label for the current lesson.
- timestamp**: A node that provides a timestamp.
- Número aleatori entre 0 i 99**: A function node that generates a random number between 0 and 99.
- msg.payload**: A message node that receives data from the random number node.
- Gràfica de proves**: A chart node that displays a line graph of test scores over time.
- Valor**: A text node that displays the current value of the random number.
- Manòmetre**: A gauge node that displays the current value of the random number as a percentage of the range (0-100).
- Dònut**: A donut chart node that displays the current value of the random number as a percentage of the range (0-100).
- Brúixola**: A compass node that displays the current value of the random number as a percentage of the range (0-100).
- Nivell**: A gauge node that displays the current value of the random number as a percentage of the range (0-100).

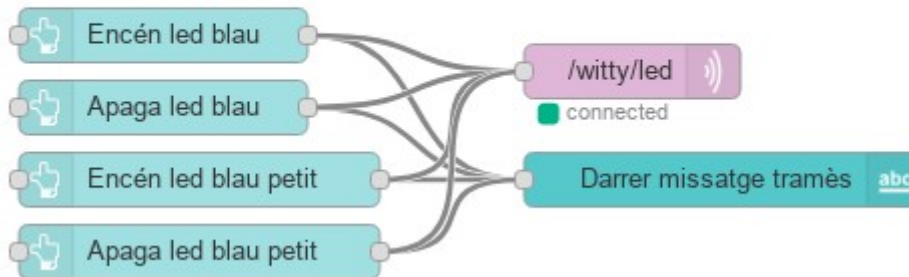
The dashboard, titled 'Visualització', is divided into three sections:

- Principal**: Contains a line graph titled 'Gràfica de proves' and a text node titled 'Valor' showing the value 33.
- Segona**: Contains a gauge titled 'Nivell' showing 37 punts, a compass titled 'Brúixola' showing 33 punts, and a donut chart titled 'Manòmetre' showing 33 punts.
- Tercera**: Contains a gauge titled 'Nivell' showing 33 punts and a donut chart titled 'Dònut' showing 33 punts.

The right sidebar shows the 'Layout' and 'Theme' settings, and the 'Tabs & Links' section, which lists the dashboard components under the 'Visualització' tab.

Panel·ls de control (*dashboards*)

Lliçó 3.21

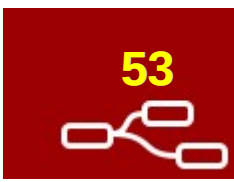


```
void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    // Attempt to connect
    if (client.connect("ESP8266 Client")) {
      Serial.println("connected");
      // ... and subscribe to topic
      client.subscribe("/witty/led");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      // Wait 5 seconds before retrying
      delay(5000);
    }
  }
}
```





Node-RED



Panel·ls de control (*dashboards*)

Codi Node-RED: https://binefa.cat/IoT/nodeRed/03_nodeRed_11.txt

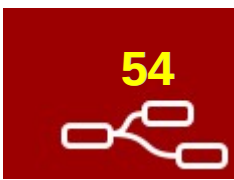
Codi d'Arduino IDE per a l'ESP32: https://binefa.cat/IoT/esp32/esp32_udpServer_00/esp32_udpServer_00.ino

Codi d'Arduino IDE per a l'ESP8266:



Node-RED

Telnet / sèrie



Codi Node-RED: https://binefa.cat/IoT/nodeRed/03_nodeRed_11.txt

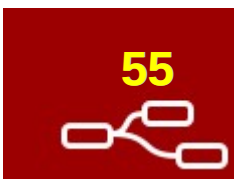
Codi d'Arduino IDE per a l'ESP32: https://binefa.cat/IoT/esp32/esp32_udpServer_00/esp32_udpServer_00.ino

Codi d'Arduino IDE per a l'ESP8266:



Node-RED

ESPnow



Codi Node-RED: https://binefa.cat/IoT/nodeRed/03_nodeRed_11.txt

Codi d'Arduino IDE per a l'ESP32: https://binefa.cat/IoT/esp32/esp32_udpServer_00/esp32_udpServer_00.ino

Codi d'Arduino IDE per a l'ESP8266: