



THE THINGS
NETWORK

LMiC e INTEGRACIONES

juanfelixmateos@gmail.com

Diciembre 2018

LMiC

- LoraMAC-in-C → LoraWAN-in-C
- Stack LoRaWAN para el entorno Arduino
 - Nos permite utilizar módulos basados en el SX1276 mediante SPI para conectar a TTN
 - Desarrollada por IBM
 - Adaptada al entorno Arduino por Matthijs Kooijman
 - <https://github.com/matthijskooijman/arduino-lmic>



LMiC: Configuración pines

```
// Pin mapping para los nodos de TTN_Madrid
const lmic_pinmap lmic_pins = {
    .nss = 10,
    .rxtx = LMIC_UNUSED_PIN,
    .rst = 9,
    .dio = {2, 7, 8},
};
```

LMiC: Reducir el tamaño

- Para reducir el tamaño podemos deshabilitar (descomentando) las opciones ping y beacon en el archivo **config.h**, que se utilizan en los dispositivos de clase B (Beacons)
- También podemos usar un DEBUG_LEVEL 0 (ver diapositiva siguiente)

```
40 // Uncomment this to disable all code related to joining
41 // #define DISABLE_JOIN
42 // Uncomment this to disable all code related to ping
43 #define DISABLE_PING
44 // Uncomment this to disable all code related to beacon tracking.
45 // Requires ping to be disabled too
46 #define DISABLE_BEACONS
```


LMiC: Depuración

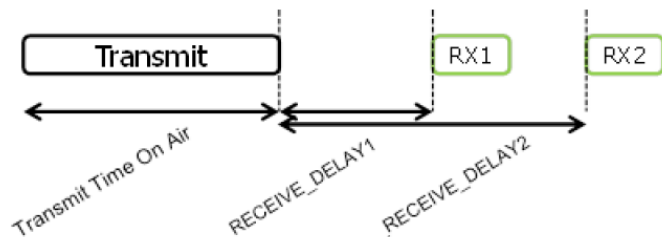
- En el archivo **config.h**
 - Aumentar el DEBUG_LEVEL a 2
 - Descomentar el LMIC_PRINTF_TO

```
23 // Set this to 1 to enable some basic debug output (using printf) about
24 // RF settings used during transmission and reception. Set to 2 to
25 // enable more verbose output. Make sure that printf is actually
26 // configured (e.g. on AVR it is not by default), otherwise using it can
27 // cause crashing.
28 #define LMIC_DEBUG_LEVEL 2
29
30 // Enable this to allow using printf() to print to the given serial port
31 // (printf object). This can be easy for debugging. The
32 // current implementation only works on AVR, though.
33 #define LMIC_PRINTF_TO Serial
```

DESCOMENTAR

LMiC: Downlink RX2 en SF9

- Los downlink RX2 de TTN usan un SF9 (en lugar del SF12 que establece el estándar)
 - Si usamos ABP**, es nuestra responsabilidad establecer la configuración de los downlink en el RX2, que en el caso de TTN utilizan **SF9** y **869.525Hz**
 - En OTAA se configura automáticamente en la fase de JOIN



```
214 // TTN uses SF9 for its RX2 window.
215 LMIC.dn2Dr = DR_SF9;
216
217 // Set data rate and transmit power for uplink (note: txpow seems to be ignored by the library)
218 LMIC_setDrTxpow(DR_SF7,14);
219
220 // Start job
221 do_send(&sendjob);
```

EU863-870

Uplink:

- 868.1 - SF7BW125 to SF12BW125
- 868.3 - SF7BW125 to SF12BW125 and SF7BW250
- 868.5 - SF7BW125 to SF12BW125
- 867.1 - SF7BW125 to SF12BW125
- 867.3 - SF7BW125 to SF12BW125
- 867.5 - SF7BW125 to SF12BW125
- 867.7 - SF7BW125 to SF12BW125
- 867.9 - SF7BW125 to SF12BW125
- 868.8 - FSK

Downlink:

- Uplink channels 1-9 (RX1)
- 869.525 - SF9BW125 (RX2 downlink only)

Note that The Things Network uses the non-standard SF9BW125 data rate for RX2 in Europe. If your devices use OTAA, this will be configured automatically when they join. If your devices use ABP, you'll need to program this RX2 data rate into the devices in order to make them work with TTN.

LMiC: Downlink problema de precisión del oscilador

- El oscilador de algunos Arduino no tiene suficiente precisión para gestionar los tiempos que requieren los downlink.
 - Tener en cuenta que para poder unirse por OTAA es necesario que funcionen los downlink
 - Puede corregirse introduciendo la siguiente línea en el setup
`LMIC_setClockError(MAX_CLOCK_ERROR * 1 / 100);`
 - O incluso:
`LMIC_setClockError(MAX_CLOCK_ERROR * 10 / 100);`

LMiC: ¿Cómo funciona?

- Modelo de programación basado en eventos
 - **onEvent**(ev_t ev)
- Se prepara una tarea de envío que se ejecuta inmediatamente
 - **LMiC_setTxData2**(1, mydata, sizeof(mydata)-1, 0);
- Cuando se complete la tarea anterior, se programa una nueva tarea para que se ejecute en un instante determinado
 - **os_setTimedCallback**(&sendjob, os_getTime()+sec2osticks(TX_INTERVAL), do_send);
- No se respetará ese instante si no cumple el duty cycle de la banda ISM

LMiC: Duty Cycle 1/2

- <https://docs.google.com/spreadsheets/d/1voGAtQAjC1qBmaVuP1ApNKs1ekgUjavHuVQIXyYSvNc/edit#gid=0>
 - File>Make a copy

Formulas taken from AN1200.13 (LoRa Modem Designers Guide)

http://www.semtech.com/images/datasheet/LoraDesignGuide_STD.pdf

No warranty of correctness!

Input

Payload size	13 bytes	Total payload, including (at least) 13 bytes of LoRaWAN header if used
Spread factor	SF9	SF12-SF6. Higher means more range and better reception, and more airtime
Explicit header	yes	no header (1) or with header (0). This is the low-level header that indicates coding rate, payload length and payload CRC presence and can be left out if both sides have these parameters fixed
Low DR optimize	no	disabled (0) or enabled (1), intended to correct for clock drift at SF11 and SF12
Coding rate	4 / 5	4/5 - 4/8. This is the error correction coding. Higher values mean more overhead.
Preamble symbols	8	8 for all regions defined in LoRaWAN 1.0, can be different using plain LoRa
Bandwidth	125kHz	Typically 125, sometimes 250 or 500

Output

Tsym	4.096	ms
Tpreamble	50.176	ms
payloadSymbNb	28	number of symbols
Tpayload	114.688	ms
Tpacket	164.864	ms

Duty cycle

	Time between packet subsequent starts	
0.10%	164.864	s
1%	16.4864	s
10%	1.64864	s

LMiC: Duty Cycle 2/2

- Los duty cycle están regulados por la sección 7.2.3 del estándar ETSI EN300.220:
 - g (863.0 – 868.0 MHz): 1%
 - g1 (868.0 – 868.6 MHz): 1%
 - g2 (868.7 – 869.2 MHz): 0.1%
 - g3 (869.4 – 869.65 MHz): 10% → Tiene sentido usarla para downlink (1 gateway para muchos nodos)
 - g4 (869.7 – 870.0 MHz): 1%
- Existe una política de uso responsable en TTN que establece:
 - 30 segundos de uplink por día y nodo
 - 10 mensajes de downlink al día

LMiC: Ejemplos de la biblioteca 1/4

- En setup ()
 - `do_send(&sendjob);`
- En `do_send(osjob_t* j)`
 - **LMiC_setTxData2**(1, mydata, sizeof(mydata)-1, 0);
 - Primer parámetro: puerto/port
 - Último parámetro: ¿confirmar upload?
- En `loop()`
 - **os_runloop_once**();
 - Se encarga de llevar un control del tiempo (timer) y ejecutar las acciones que correspondan en cada instante.

LMiC: Ejemplos de la biblioteca 2/4

- En `on_event(ev_t ev)`
case `EV_TXCOMPLETE`:
 `Serial.println(F("EV_TXCOMPLETE (includes waiting for RX windows)"));`
 if (`LMIC.txrxFlags & TXXRX_ACK`)
 `Serial.println(F("Received ack"));`
 if (`LMIC.dataLen`) {
 `Serial.println(F("Received "));`
 `Serial.println(LMIC.dataLen);`
 `Serial.println(F(" bytes of payload"));`
 }
 // Schedule next transmission
 `os_setTimedCallback`(&sendjob, `os_getTime()+sec2osticks(TX_INTERVAL)`, `do_send`);
 break;

LMiC: Ejemplos de la biblioteca 3/4

- ABP

```
RXMODE_RSSI
478: engineUpdate, opmode=0x808
540: link data pending
590: Considering band 0, which is available at 464
73: is available at 0
10: 3
11: is available at 464
14: 0
16: is available at 464
196: (channel duty limit)
2254: Ready for uplink
3223: Updating info for TX at 539, airtime will be 3856. Setting available time for band 1 to 3831169024
3542: TXMODE, freq=867100000, len=26, SF=7, BW=125, CR=4/5, IH=0
Packet queued
7139: irq: dio: 0x0 flags: 0x8
7223: Scheduled job 0x37e, cb 0xbfe ASAP
7312: Running job 0x37e, cb 0xbfe, deadline 0
7461: Scheduled job 0x37e, cb 0xa33 at 69608
69613: Running job 0x37e, cb 0xa33, deadline 69608
69785: RXMODE_SINGLE, freq=867100000, SF=7, BW=125, CR=4/5, IH=0
69870: irq: dio: 0x1 flags: 0x80
69982: Scheduled job 0x37e, cb 0x199a ASAP
70212: Running job 0x37e, cb 0x199a, deadline 0
70484: Scheduled job 0x37e, cb 0xa5f at 132396
132403: Running job 0x37e, cb 0xa5f, deadline 132396
132581: RXMODE_SINGLE, freq=869525000, SF=9, BW=125, CR=4/5, IH=0
133749: irq: dio: 0x1 flags: 0x80
133840: Scheduled job 0x37e, cb 0x19b1 ASAP
133934: Running job 0x37e, cb 0x19b1, deadline 0
134122: EV_TXCOMPLETE (includes waiting for RX windows)
134442: Scheduled job 0x260, cb 0x41c at 3884437
134713: engineUpdate, opmode=0x900
```

TIEMPO EN TICKS
ARDUINO UTILIZA
62500
TICKS POR
SEGUNDO

ENVÍO DEL PRIMER MENSAJE

ABRIR LA VENTANA RX1 EN 1 SEGUNDO

ABRIR LA VENTANA RX2 EN 1 SEGUNDO

SIGUIENTE ENVÍO EN 60 SEGUNDOS

LMiC: Ejemplos de la biblioteca 4/4

- OTAA

```
Starting
RXMODE_RSSI
445: engineUpdate, opmode=0x8
521: Scheduled job 0x37e, cb 0x132b ASAP
Packet queued
622: Running job 0x37e, cb 0x132b, deadline 0
832: EV_JOINING
927: engineUpdate, opmode=0xc
1097: Uplink join pending
1242: Airtime available at 165947 (previously determined)
1569: Uplink delayed until 165947
1765: Scheduled job 0x37e, cb 0x155a at 165822
165829: Running job 0x37e, cb 0x155a, deadline 165822
165943: engineUpdate, opmode=0xc
166011: Uplink join pending
166124: Airtime available at 165947 (previously determined)
166462: Ready for uplink
167135: Updating info for TX at 166010, airtime will be 3856. Setting available time for band 0 to 1593442304
167484: TXMODE, freq=868300000, len=23, SF=7, BW=125, CR=4/5, IH=0
170957: irq: dio: 0x0 flags: 0x8
171045: Scheduled job 0x37e, cb 0xb8c ASAP
171140: Running job 0x37e, cb 0xb8c, deadline 0
171323: Scheduled job 0x37e, cb 0x9ba at 480322
480328: Running job 0x37e, cb 0x9ba, deadline 480322
480506: RXMODE_SINGLE, freq=868300000, SF=7, BW=125, CR=4/5, IH=0
487027: irq: dio: 0x0 flags: 0x40
487150: Scheduled job 0x37e, cb 0x153b ASAP
487246: Running job 0x37e, cb 0x153b, deadline 0
488231: Setup channel, idx=3, freq=867100000
488333: Setup channel, idx=4, freq=867300000
488437: Setup channel, idx=5, freq=867500000
488641: Setup channel, idx=6, freq=867700000
488894: Setup channel, idx=7, freq=867900000
489512: EV_JOINED
```

ENVÍO DE LA PETICIÓN DE JOINING

EL RX2 PARA JOIN SE PRODUCE A LOS 5 SEGUNDOS

CONFIGURACIÓN DE LOS CANALES CON LA INFORMACIÓN RECIBIDA EN EL ACCEPT DEL JOINING

LMiC: ¿Cómo hacer envíos inmediatos
(no programados)?

- Cambiamos...
 - **os_setTimedCallback**(&sendjob,
os_getTime()+sec2osticks(TX_INTERVAL), do_send);
- por...
 - **os_setCallback** (&sendjob, do_send);
- ATENCIÓN: El duty cycle sigue aplicándose

LMiC & Low-Power: Envíos periódicos 1/2

- En lugar de programar un envío, hacemos envíos inmediatos y ponemos el uC a dormir tras completarse cada uno de ellos.

Low-Power by Rocket Scream Electronics Versión 1.6.0 **INSTALLED**
Lightweight power management library Lightweight power management library
[More info](#)

```
--  
32 #include <lmic.h>  
33 #include <hal/hal.h>  
34 #include <SPI.h>  
35 #include "LowPower.h"  
--
```

LMiC & Low-Power: Envíos periódicos 2/2

- En el evento EV_TXCOMPLETE

```
104     case EV_TXCOMPLETE:
105         Serial.println(F("EV_TXCOMPLETE (includes waiting for RX windows)"));
106         if (LMIC.txrxFlags & TXRX_ACK)
107             Serial.println(F("Received ack"));
108         if (LMIC.dataLen) {
109             Serial.println(F("Received "));
110             Serial.println(LMIC.dataLen);
111             Serial.println(F(" bytes of payload"));
112         }
113         // Schedule next transmission
114         //os_setTimedCallback(&sendjob, os_getTime()+sec2osticks(TX_INTERVAL), do_send);
115         Serial.println("Me voy a echar una siestecita");
116         delay(1000); // Para que terminen de imprimirse los mensajes en el terminal
117         for (byte contador = 0; contador < 8; contador++) {
118             LowPower.powerDown(SLEEP_8S, ADC_OFF, BOD_OFF);
119         }
120         os_setCallback (&sendjob, do_send);
121         break;
122     case EV_LOST_TSYNC:
```

LMiC & Low-Power: Envíos por interrupción 1/3

- Creamos una variable bandera para indicar si hay un envío en curso
- En el setup() anulamos el envío inicial y ponemos el pin 3 en modo INPUT_PULLUP

```
31  
32 #include <lmic.h>  
33 #include <hal/hal.h>  
34 #include <SPI.h>  
35 #include "LowPower.h"  
36 boolean envioEnCurso = false;  
37
```

```
224 // Start job  
225 //cb_send(&sendjob);  
226 LMIC_setClockError(MAX_CLOCK_ERROR * 1 / 100);  
227 //Conecto en el pin 3 un interruptor a GND normalmente abierto  
228 pinMode(3, INPUT_PULLUP);  
229 }
```

LMiC & Low-Power: Envíos por interrupción 2/3

- En el evento EV_TXCOMPLETE, anulamos el envío periódico, e indicamos que el envío ha finalizado

```
105 case EV_TXCOMPLETE:
106     Serial.println(F("EV_TXCOMPLETE (includes waiting for RX windows)"));
107     if (LMIC.txrxFlags & TXRX_ACK)
108         Serial.println(F("Received ack"));
109     if (LMIC.dataLen) {
110         Serial.println(F("Received "));
111         Serial.println(LMIC.dataLen);
112         Serial.println(F(" bytes of payload"));
113     }
114     // Schedule next transmission
115     //os_setTimedCallback(&sendjob, os_getTime()+sec2osticks(TX_INTERVAL), do_send);
116     envioConcluido=true;
117     break;
```

LMiC & Low-Power: Envíos por interrupción 3/3

- En el loop() creamos la lógica de la interrupción

```
230 void enviarMensaje() {
231     os_setCallback (&sendjob, do_send);
232 }
233 void loop() {
234     attachInterrupt(digitalPinToInterrupt(3), enviarMensaje, FALLING);
235     Serial.println("Me voy a dormir hasta que ocurra algo");
236     delay(1000);
237     LowPower.powerDown(SLEEP_FOREVER, ADC_OFF, BOD_ON);
238     // Deshabilitamos las interrupciones al despertarnos
239     detachInterrupt(0);
240     envioEnCurso=true;
241     //Esperamos a que concluya el envío
242     while(envioEnCurso){
243         os_runloop_once();
244     }
245 }
```


LMiC & Low-Power: Puerta abierta/cerrada 1/2

- Cambiamos la variable `my_data` a un array de 51 elementos
- En `do_send()` detectamos la longitud real del mensaje

```
58 //static uint8_t mydata[] = "Hello, world!";  
59 uint8_t mydata[51];  
60 static osjob_t sendjob;  
..
```

```
141 void do_send(osjob_t* j) {  
142     // Check if there is not a current TX/RX job running  
143     if (LMIC.opmode & OP_TXRXPEND) {  
144         Serial.println(F("OP_TXRXPEND, not sending"));  
145     } else {  
146         // Prepare upstream data transmission at the next possible time.  
147         //LMIC_setTxData2(1, mydata, sizeof(mydata) - 1, 0);  
148         LMIC_setTxData2(1, mydata, strlen((char *)mydata), 0);  
149         Serial.println(F("Packet queued"));  
150     }  
151     // Next TX is scheduled after TX_COMPLETE event.  
152 }
```

LMiC & Low-Power: Puerta abierta/cerrada 2/2

- Cambios en loop()

```
234 void loop() {  
235   strcpy((char *) mydata, "Puerta abierta");  
236   attachInterrupt(digitalPinToInterrupt(3), enviarMensaje, FALLING);  
237   Serial.println("Me voy a dormir hasta que se abra la puerta");  
238   delay(100);  
239   LowPower.powerDown(SLEEP_FOREVER, ADC_OFF, BOD_ON);  
240   //Se ha abierto la puerta  
241   // Deshabilitamos las interrupciones al despertarnos  
242   detachInterrupt(0);  
243   envioEnCurso = true;  
244   //Esperamos a que concluya el envío  
245   while (envioEnCurso) {  
246     os_runloop_once();  
247   }
```

```
248   strcpy((char *) mydata, "Puerta cerrada");  
249   if (digitalRead(3) == HIGH) {  
250     do_send(&sendjob);  
251   } else {  
252     attachInterrupt(digitalPinToInterrupt(3), enviarMensaje, RISING);  
253     Serial.println("Me voy a dormir hasta que se cierre la puerta");  
254     delay(100);  
255     LowPower.powerDown(SLEEP_FOREVER, ADC_OFF, BOD_ON);  
256   }  
257   //Se ha cerrado la puerta  
258   // Deshabilitamos las interrupciones al despertarnos  
259   detachInterrupt(0);  
260   envioEnCurso = true;  
261   //Esperamos a que concluya el envío  
262   while (envioEnCurso) {  
263     os_runloop_once();  
264   }  
265 }
```

¿Qué son las integraciones?

- Tus datos son tus datos: Si no te encargas de ellos se pierden
 - ¿Cómo me encargo de ellos?
 - Con integraciones para:
 - Generar avisos
 - Almacenarlos
 - Modificar el estado de actuadores

¿Qué aspecto tienen los datos?

- Payload
- RAW
- Formateada
- Metadata

The screenshot shows the 'Data' view for the application 'jfmateos_taller_integraciones'. The interface includes filters for 'uplink', 'downlink', 'activation', 'ack', and 'error'. The selected message is an uplink from device 'jfmateos_nodo_v3' at 18:46:05. The message details are as follows:

- Uplink**: The 'Payload' is shown as a hexadecimal string: 01 67 00 C8 02 73 03 A8 03 68 58 04 02 01 38 05 66 00 06 01 01.
- Fields**: A JSON object containing sensor data:

```
{  "analog_in_4": 3.12,  "barometric_pressure_2": 93.6,  "digital_out_6": 1,  "presence_5": 0,  "relative_humidity_3": 44,  "temperature_1": 20}
```
- Metadata**: A JSON object containing message and gateway information:

```
{  "time": "2018-12-13T17:46:05.459148513Z",  "frequency": 868.1,  "modulation": "LORA",  "data_rate": "SF7Bw125",  "coding_rate": "4/5",  "gateways": [    {      "gtw_id": "eui-2c3ae8ffff225d1f",      "timestamp": 2342586533,      "time": "",      "channel": 0,      "rssi": -51,      "snr": 6,      "latitude": 40.37384,      "longitude": -3.7321    },    {      "gtw_id": "eui-b827ebffff8d9794",      "timestamp": 1288370683,      "time": "2018-12-13T17:46:05.440009Z",      "channel": 0,      "rssi": -60,      "snr": 10,      "rf_chain": 1,      "latitude": 40.37384,      "longitude": -3.7321    }  ]}
```

¿Cómo formateamos el Payload?

- Podemos usar nuestro propio esquema...
 - ...pero Cayenne lpp (low power payload) es muy popular y logra cierto nivel de estandarización basado en:
 - IPSO Alliance Smart Objects Guidelines

Cayenne LPP: Canal

- Cada sensor/actuador está identificado por un número de canal del 0 al 64

Payload

01 67 00 C8 02 73 03 A8 03 68 58 04 02 01 38 05 66 00 06 01 01

Canal 1 Canal 2 ...

```
{
  "analog_in_4": 3.12,
  "barometric_pressure_2": 93.6,
  "digital_out_6": 1,
  "presence_5": 0,
  "relative_humidity_3": 44,
  "temperature_1": 20
}
```


- Cada sensor/actuador tiene asociado un tipo de dato

The diagram illustrates the process of interpreting a temperature sensor reading from a JSON payload. It features a light blue background with a white box containing a JSON snippet. A red circle highlights the value "00C8" in the "temperature_1" field. A blue callout box points to this value, containing the text: "Temperatura con resolución 0.1°C con signo y MSB" and the calculation "0x00C8=200=20.0C". Another blue callout box points to the "temperature_1" field label, containing the text "Sensor de temperatura". A third blue callout box points to the "Canal 1" label, which is also highlighted with a red circle. The JSON snippet is as follows:

```
{
  "analog_in_4": 3,
  "barometric_press": 1013,
  "digital_out_6": 1,
  "presence_5": 0,
  "relative_humidity_3": 44,
  "temperature_1": 20
}
```

Cayenne LPP: Biblioteca para Arduino 1/2

```
#include <CayenneLPP.h>
```

```
CayenneLPP lpp(uint8_t size);
```

- Reserva el buffer para el payload
- El tamaño máximo varía en función de la frecuencia y el spread factor (un valor seguro es 51 bytes)

```
lpp.reset();
```

```
lpp.addTemperature(1, 22.5);
```

```
lpp.addBarometricPressure(2, 1073.21);
```

```
lpp.addGPS(3, 52.37365, 4.88650, 2);
```

```
LMIC_setTxData2(1, lpp.getBuffer(), lpp.getSize(),  
0);
```

CayenneLPP by Electronic Cats Versión 1.0.1 **INSTALLED**
CayenneLPP Arduino Library. Compatible with Cayenne Low Power Payload.
[More info](#)

Seleccione vers... ▼

Instalar

```
uint8_t addDigitalInput(uint8_t channel, uint8_t value);  
uint8_t addDigitalOutput(uint8_t channel, uint8_t value);
```

```
uint8_t addAnalogInput(uint8_t channel, float value);  
uint8_t addAnalogOutput(uint8_t channel, float value);
```

```
uint8_t addLuminosity(uint8_t channel, uint16_t lux);  
uint8_t addPresence(uint8_t channel, uint8_t value);  
uint8_t addTemperature(uint8_t channel, float celsius);  
uint8_t addRelativeHumidity(uint8_t channel, float rh);  
uint8_t addAccelerometer(uint8_t channel, float x, float y, float z);  
uint8_t addBarometricPressure(uint8_t channel, float hpa);  
uint8_t addGyrometer(uint8_t channel, float x, float y, float z);  
uint8_t addGPS(uint8_t channel, float latitude, float longitude, float meters);
```

Cayenne LPP: Biblioteca para Arduino 2/2

- Partiendo del ejemplo
 - _04_ttn-abp_madrid_envio_cada_60_segundos_lowpower
- Incluir la biblioteca y crear el objeto lpp
- Modificar do_send()

```
32 #include <lmic.h>
33 #include <hal/hal.h>
34 #include <SPI.h>
35 #include "LowPower.h"
36 #include <CayenneLPP.h>
37 CayenneLPP lpp(51);
```

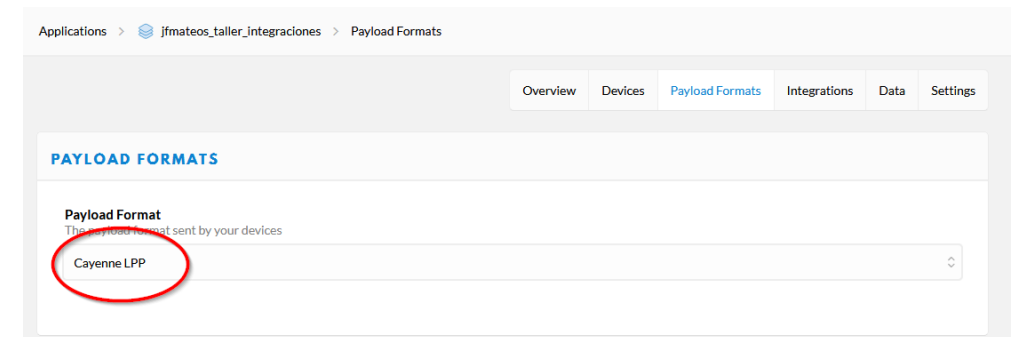
```
147 void do_send(osjob_t* j) {
148     // Check if there is not a current TX/RX job running
149     if (LMIC.opmode & OP_TXRXPEND) {
150         Serial.println(F("OP_TXRXPEND, not sending"));
151     } else {
152         // Prepare upstream data transmission at the next possible time.
153         //LMIC_setTxData2(1, mydata, sizeof(mydata)-1, 0);
154         lpp.reset();
155         lpp.addTemperature(1, 22.5);
156         lpp.addBarometricPressure(2, 1073.21);
157         lpp.addGPS(3, 52.37365, 4.88650, 2);
158         LMIC_setTxData2(1, lpp.getBuffer(), lpp.getSize(), 0);
159
160         Serial.println(F("Packet queued"));
161     }
162     // Next TX is scheduled after TX_COMPLETE event.
163 }
```

Cayenne LPP: Formato de payload

- Activamos el formato de payload Cayenne lpp para extraer los campos de la carga de pago
- De la carga de pago
 - 016700E1027329EC038807FDD800BEE10000C8
- Se extraen los campos

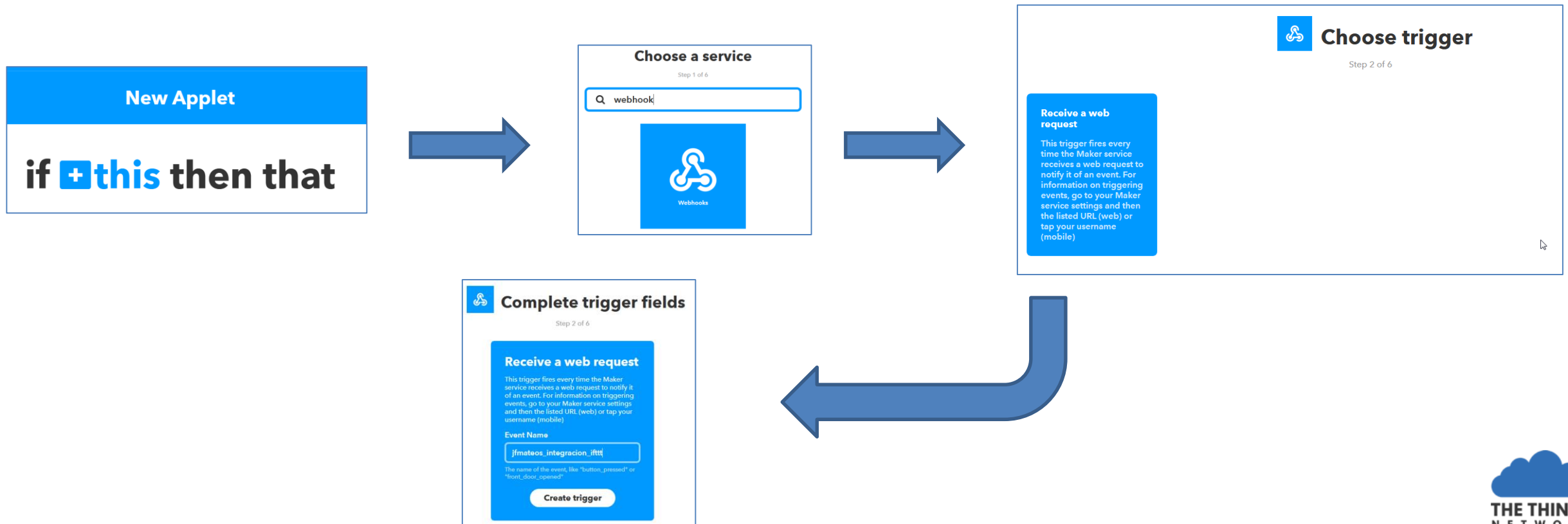
Fields

```
{
  "barometric_pressure_2": 1073.2,
  "gps_3": {
    "altitude": 2,
    "latitude": 52.3736,
    "longitude": 4.8865
  },
  "temperature_1": 22.5
}
```



IFTTT: Crear un applet 1/3

- Definimos la sección **this** como un webhook




IFTTT: Crear un applet 2/3

- Definimos la sección **that** para que nos envíe una notificación

Choose action service
Step 3 of 6

Q notifications



Notifications



Choose action
Step 4 of 6

Send a notification from the IFTTT app
This action will send a notification to your devices from the IFTTT app.

Send a rich notification from the IFTTT app
This action will send a rich notification to your devices from the IFTTT app. Rich notifications may include a title, image, and link that opens in a browser or installed app.



Complete action fields
Step 5 of 6

Send a notification from the IFTTT app
This action will send a notification to your devices from the IFTTT app.

Message

`"{{EventName}}":
{{OccurredAt}}: Temperatura:
{{Value1}} - Presion: {{Value2}}`

Add ingredient

Create action



Review and finish
Step 6 of 6

If Maker Event "jfmateos_integracion_ifttt", then Send a notification from the IFTTT app

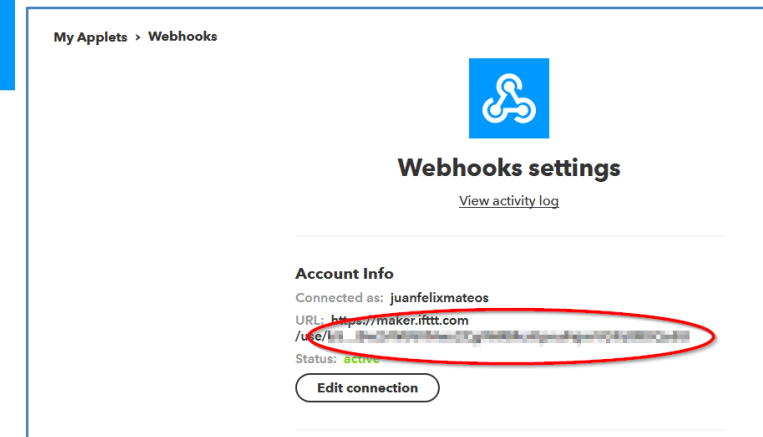
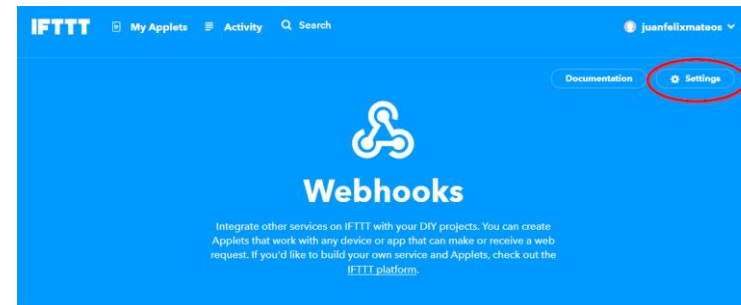
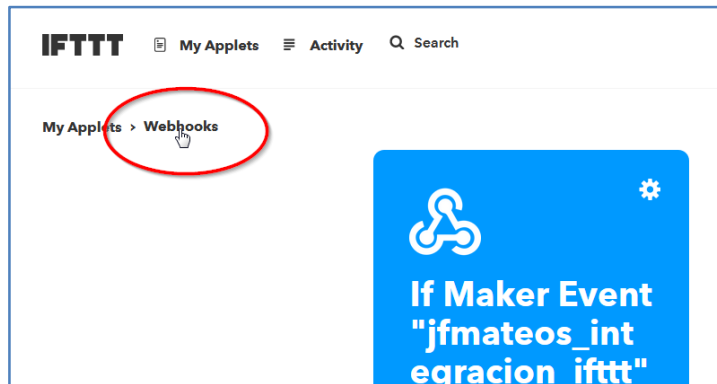
by juanfelmateos

works with

Finish

IFTTT: Crear un applet 3/3

- Obtener la clave/llave del webhook



IFTTT: Activar la integración en TTN

- Utilizar el nombre del evento y la clave de IFTTT

Applications > jfmateos_taller_integraciones > Integrations

Process ID
The unique identifier of the new integration process

jfmateos_mi_integracion_ifttt

Event Name
The event name of your IFTTT recipe

jfmateos_integracion_ifttt

Key
Your key

...

Value 1
Payload field name to send as value 1

temperature_1

Value 2
Payload field name to send as value 2

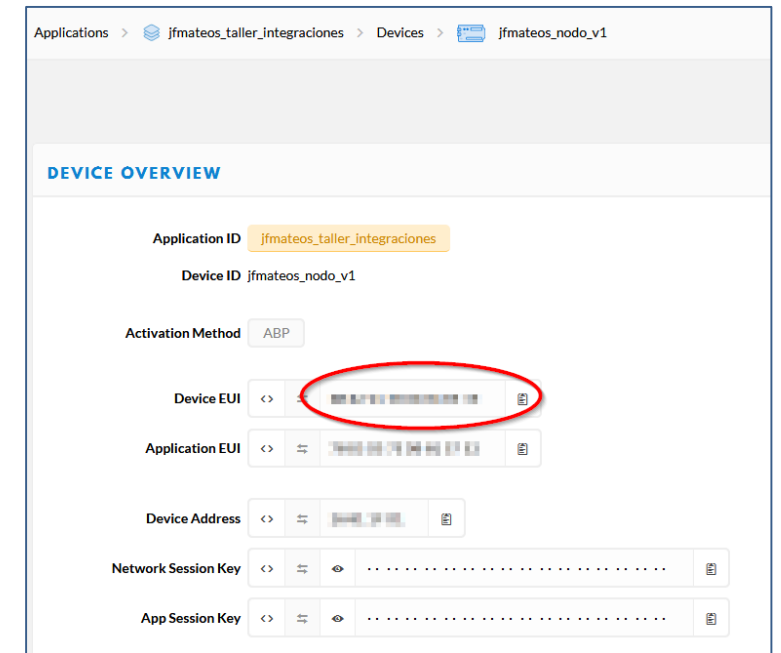
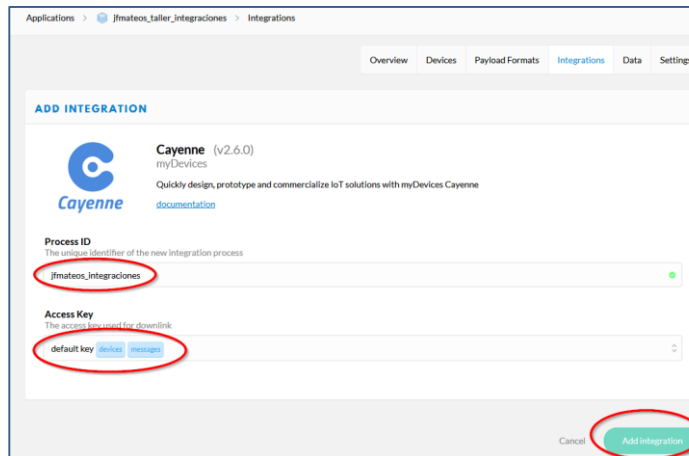
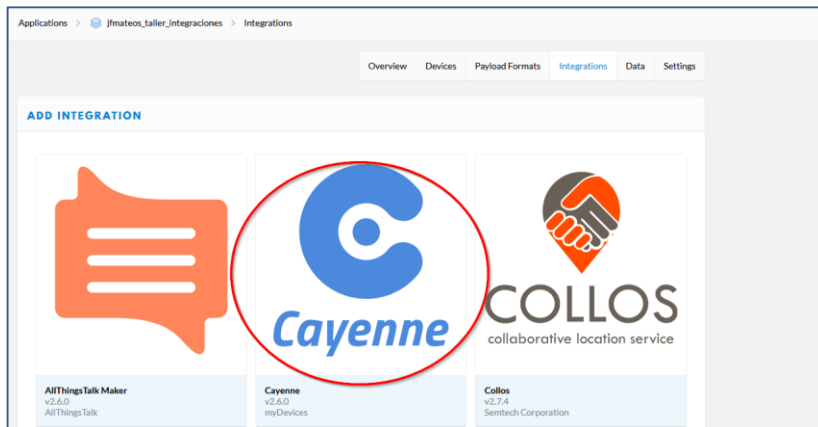
barometric_pressure_2

Value 3
Payload field name to send as value 3

Cancel Add integration

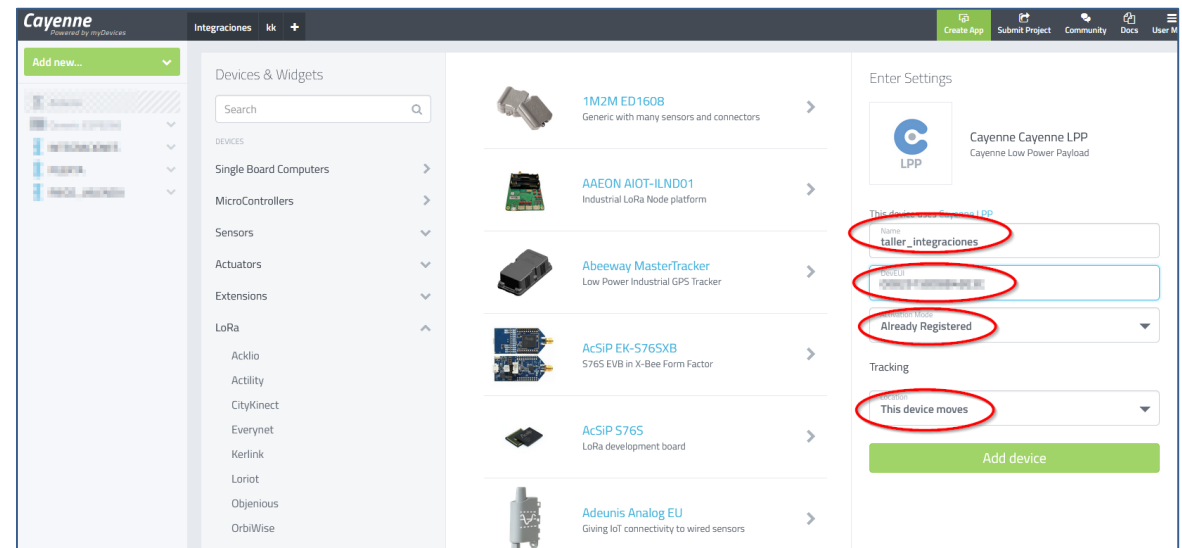
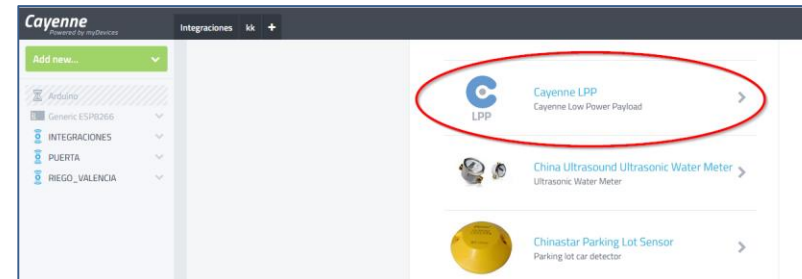
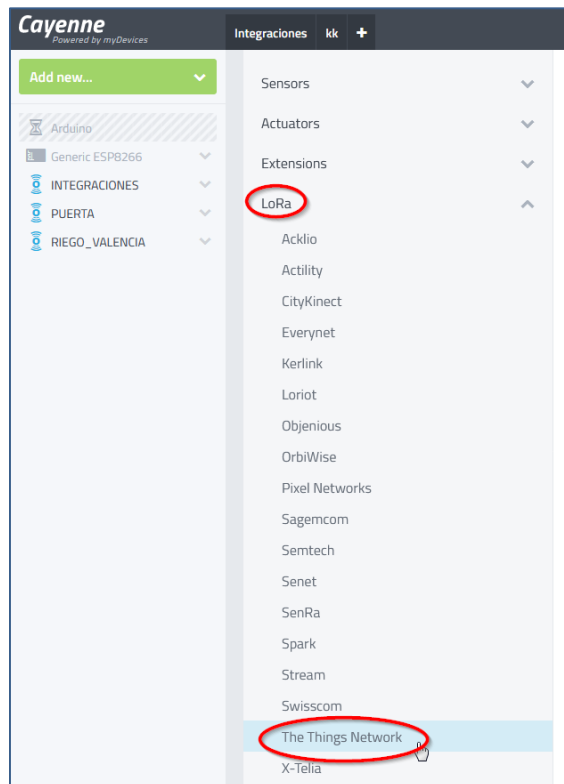
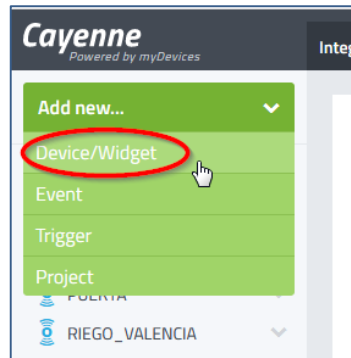
CAYENNE

- Activamos la integración en TTN y tomamos el EUI del dispositivo

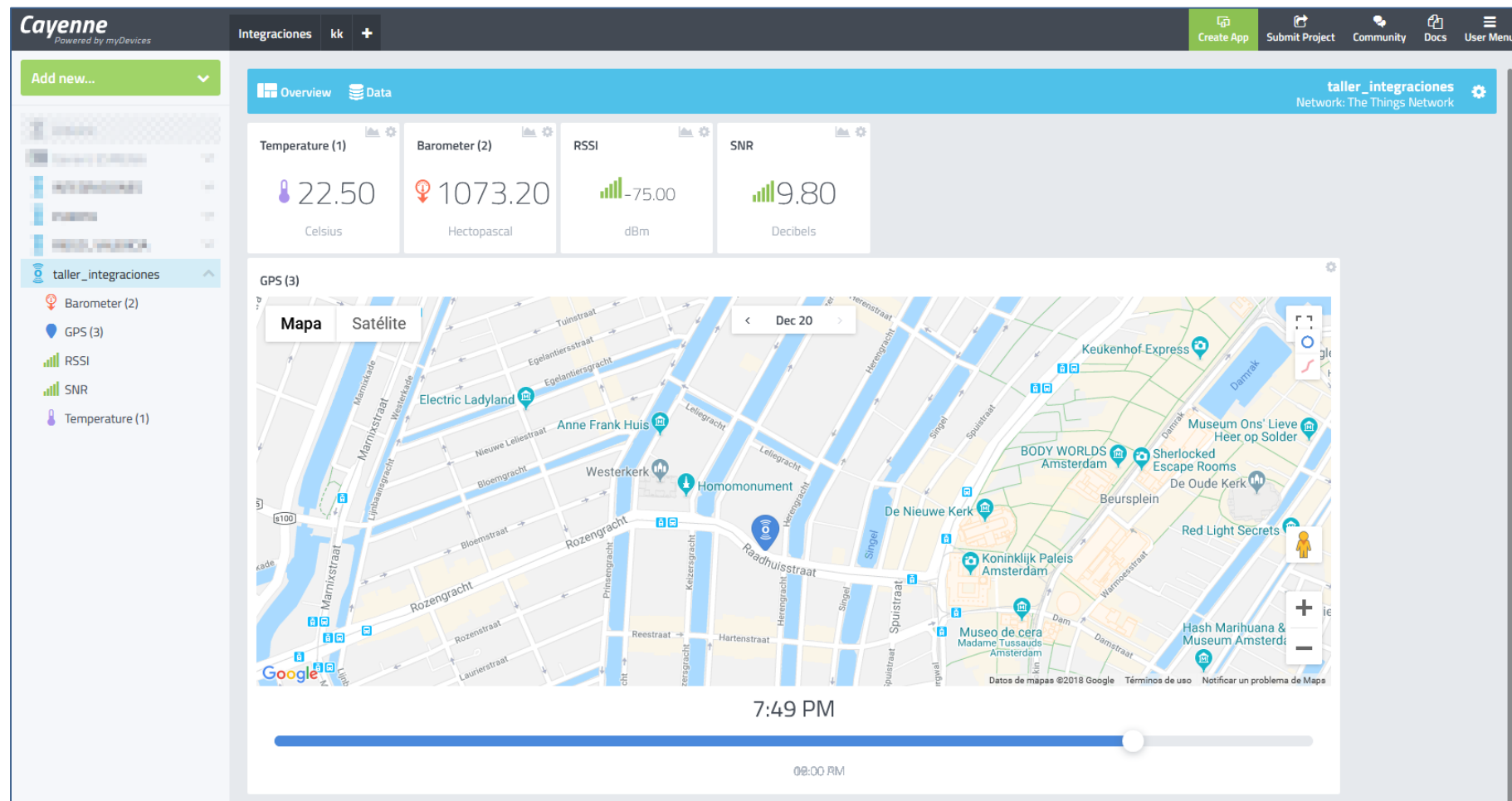


CAYENNE

- Registramos el device en Cayenne

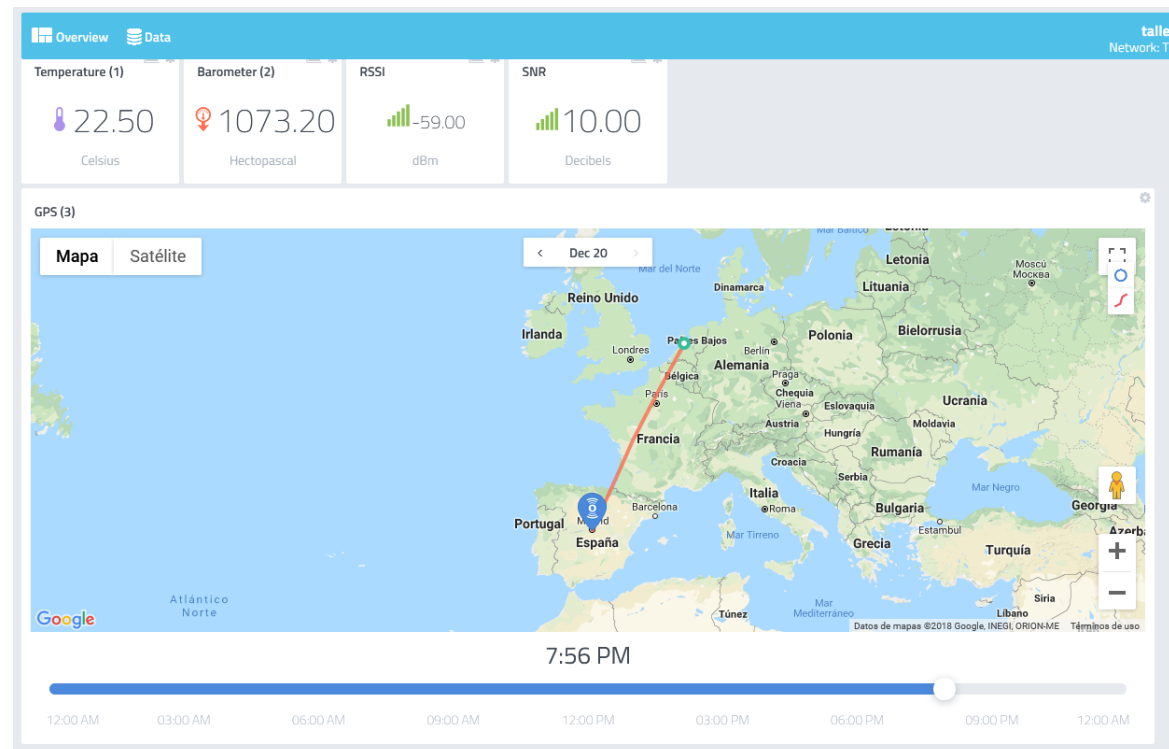


CAYENNE



CAYENNE

- Simular un uplink con esta carga de pago para cambiar de Amsterdam a Madrid
 - 016700E1027329EC0388062A88FF6FB80000C8



CAYENNE: Downlink

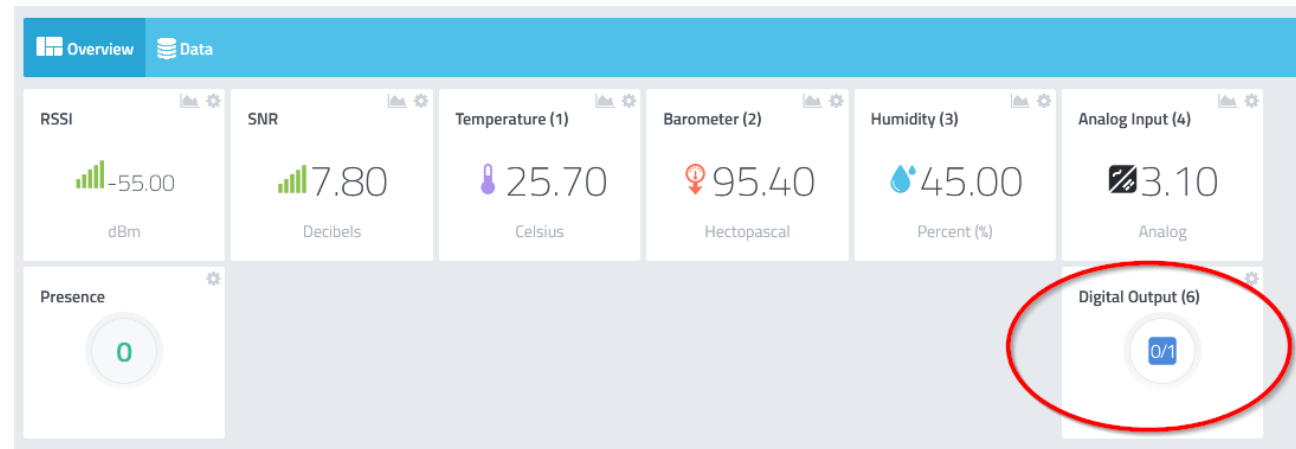
- Con los método addDigitalOutput() y addAnalogOutput() podemos generar downlinks desde Cayenne
 - Los downlink de Cayenne se envían con solicitud de ACK

```
lpp.addDigitalOutput(6, 1);    // channel 6, set digital output high
```

```
case EV_TXCOMPLETE:
    Serial.println(F("EV_TXCOMPLETE (includes waiting for RX windows)"));
    if (LMIC.txrxFlags & TXRX_ACK)
        Serial.println(F("Received ack"));

    if (LMIC.dataLen) {
        Serial.println(F("Received "));
        Serial.println(LMIC.dataLen);
        Serial.println(F(" bytes of payload"));
        for (int i = 0; i < LMIC.dataLen; i++) {
            Serial.println(LMIC.frame[LMIC.dataBeg + i]);
        }
        if (LMIC.frame[LMIC.dataBeg + 2] == 100) {
            digitalWrite(6, HIGH);
        } else {
            digitalWrite(6, LOW);
        }
    }
}
```

digitalOutput
devuelve 0 o 100



Data Storage

- La integración Data Storage almacena nuestros datos de la última semana




Applications > jfmateos_taller_integraciones > Integrations > main

Overview Devices Payload Formats **Integrations** Data Settings

INTEGRATION OVERVIEW

Status ● Running

Integration info [go to platform](#)

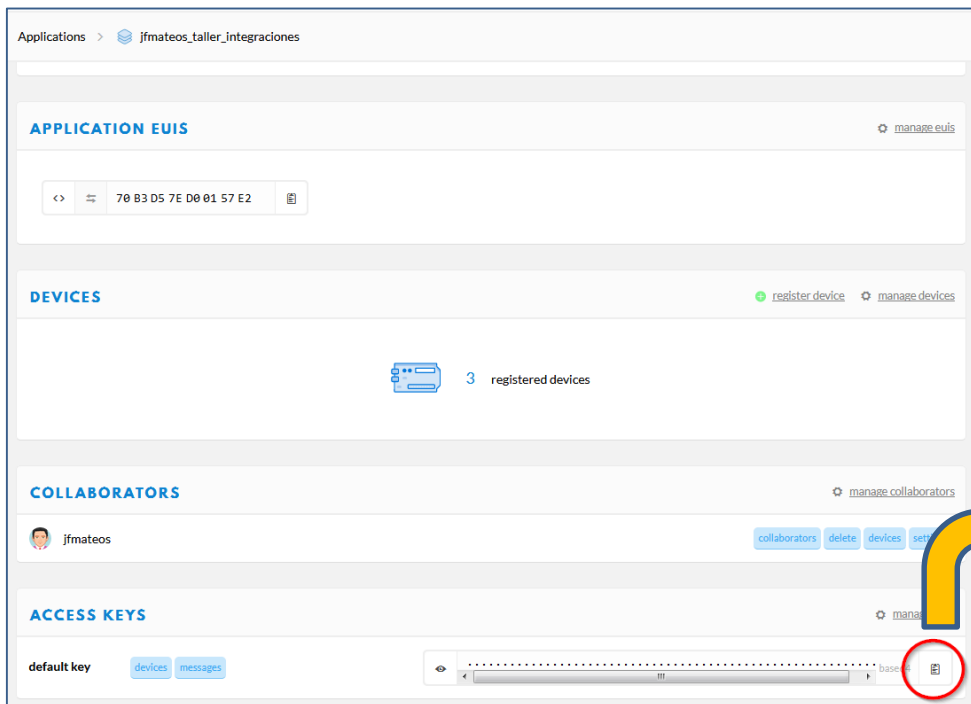
Platform  Data Storage (v2.0.1)

Author The Things Industries B.V.

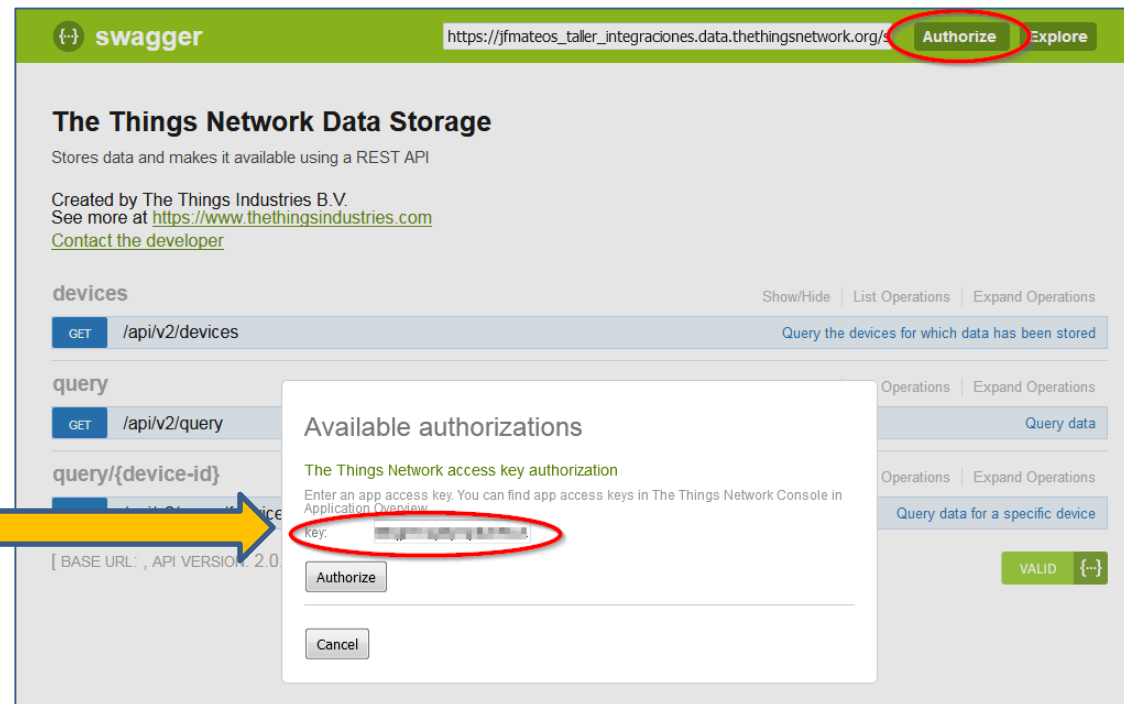
Description Stores data and makes it available through an API. Your data is stored for seven days.

Data Storage

- Usamos la Access Key para acreditarnos en la REST API de Swagger



The screenshot shows the 'Applications' page for 'jfmateos_taller_integraciones'. It includes sections for 'APPLICATION EUIs', 'DEVICES' (3 registered devices), 'COLLABORATORS', and 'ACCESS KEYS'. In the 'ACCESS KEYS' section, a 'default key' is displayed with a 'copy' icon circled in red.



The screenshot shows the Swagger REST API interface for 'The Things Network Data Storage'. The URL is 'https://jfmateos_taller_integraciones.data.thethingsnetwork.org/'. The 'Authorize' button is circled in red. A modal window titled 'Available authorizations' is open, showing 'The Things Network access key authorization' with a red circle around the 'key' input field.

Data Storage

- Probamos el query de todos los datos

query Show/Hide | List Operations | Expand Operations

GET `/api/v2/query` Query data

Implementation Notes
Query the data for all devices

Response Class (Status 200)
An array of data

Model | **Example Value**

```
[
  {
    "device_id": "string",
    "raw": "string",
    "time": "string",
    "field1": "string",
    "field2": "string"
  }
]
```

Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
last	<input type="text"/>	Duration on which we want to get the data (default 1h). Pass 30s for the last 30 seconds, 1h for the last hour, 2d for the last 48 hours, etc	query	string

Try it out! [Hide Response](#)

Curl

```
curl -X GET --header 'Accept: application/json' --header 'Authorization: ...'
```

Request URL

```
https://jfmateos_taller_integraciones.data.thethingsnetwork.org/api/v2/query
```

Response Body

```
{
  "presence_5": 0,
  "raw": "AWcA8gJzA7oDaGUEAgE4BWYABgEBBwMu4A==",
  "relative_humidity_3": 50.5,
  "temperature_1": 24.2,
  "time": "2018-12-20T20:33:55.561273224Z"
},
{
  "analog_in_4": null,
  "analog_out_7": null,
  "barometric_pressure_2": 1073.2,
  "device_id": "jfmateos_nodo_v1",
  "digital_out_6": null,
  "gps_3": "map[latitude:52.3736 longitude:4.8865 altitude:2]",
  "presence_5": null,
  "raw": "AWcA4QJzKewD1Af92AC+4QAAYA==",
  "relative_humidity_3": null,
  "temperature_1": 22.5,
  "time": "2018-12-20T20:34:40.972988337Z"
},
{
  "connection": "keep-alive",
  "content-type": "application/json",
  "date": "Thu, 20 Dec 2018 21:21:52 GMT",
  "server": "nginx/1.13.7",
  "transfer-encoding": "chunked"
}
```

Response Code

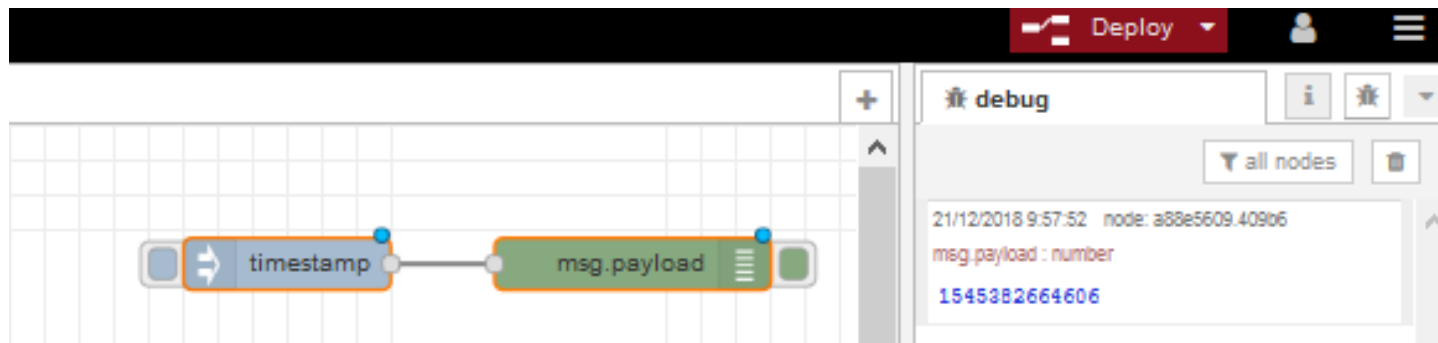
```
200
```

Response Headers

```
{
  "connection": "keep-alive",
  "content-type": "application/json",
  "date": "Thu, 20 Dec 2018 21:21:52 GMT",
  "server": "nginx/1.13.7",
  "transfer-encoding": "chunked"
}
```

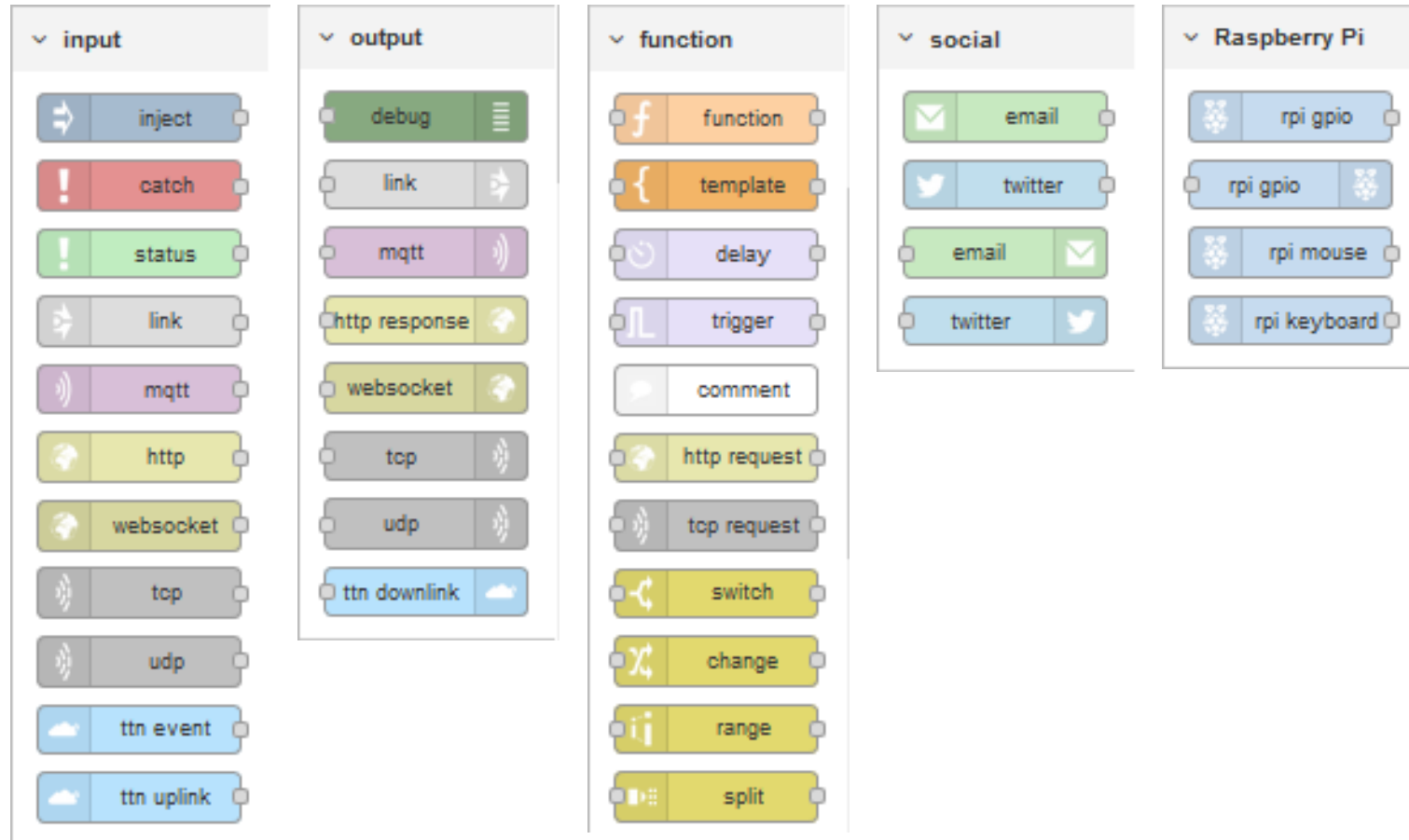
Node-RED

- Desarrollado por IBM en 2013 → Open Source
- Objetivo: Facilitar la construcción de sistemas IoT
- Programación visual basada en **nodos** y **flujos**
- Los nodos "consumen" y "produce" **mensajes**
- Un mensaje es un objeto JavaScript, que contiene al menos la propiedad **payload**
 - `msg={payload:"juanfe"};`
 - Adicionalmente, por defecto, todos los mensajes poseen las propiedades **topic** y **_msgid**



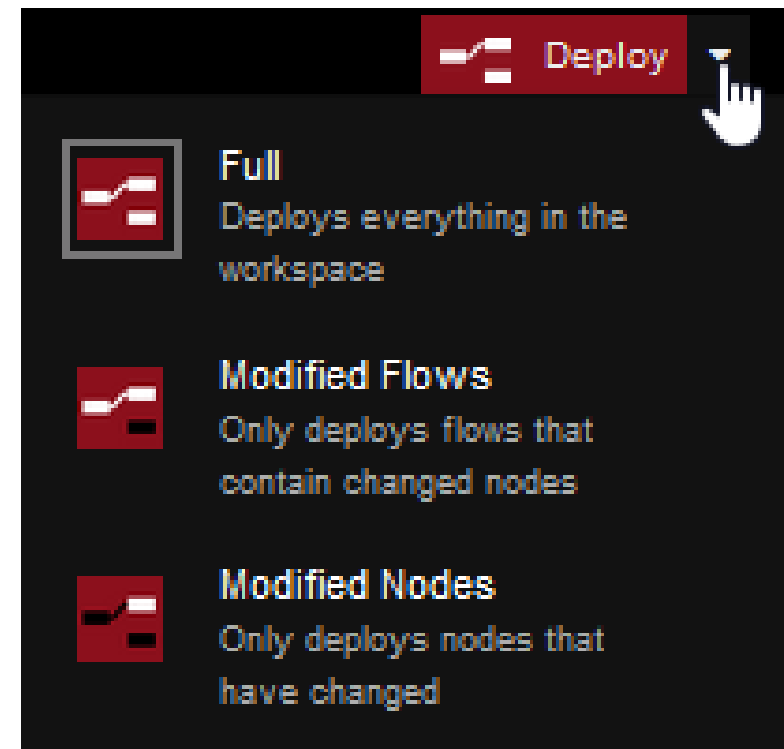
Node-RED: Tipos de nodos

- Entrada
- Salida
- Función
- Otros



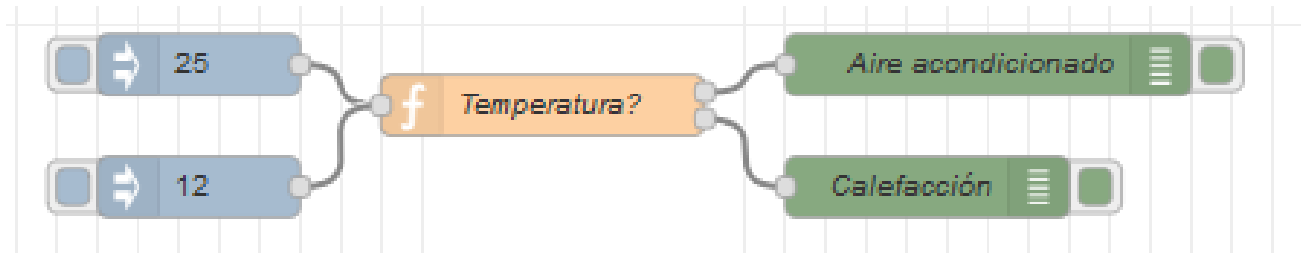
Node-RED: Nodos y deployment

- Deployment/Despliegue = Poner en funcionamiento



Node-RED: Ejemplo básico

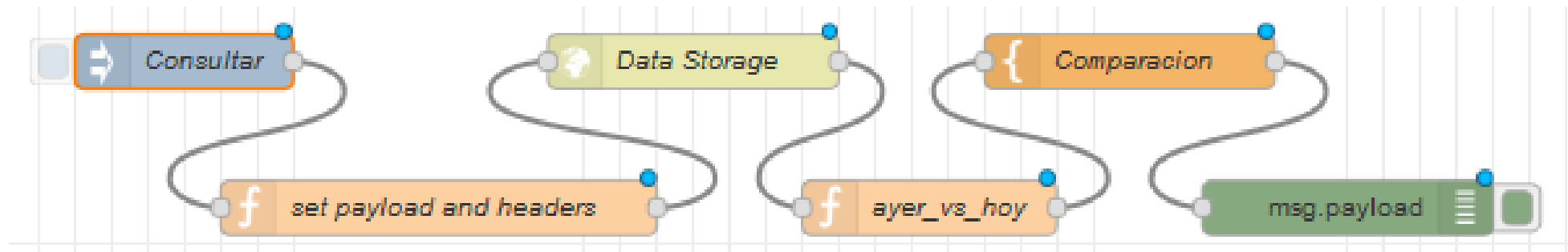
- En función del valor recibido, redirigimos la ejecución hacia un nodo u otro



```
node properties
Name
Temperatura?
Function
1 if(msg.payload>22){
2   msg.payload="Hace calor";
3   return [msg,null];
4 }else{
5   msg.payload="Hace frío";
6   return [null,msg];
7 }
```

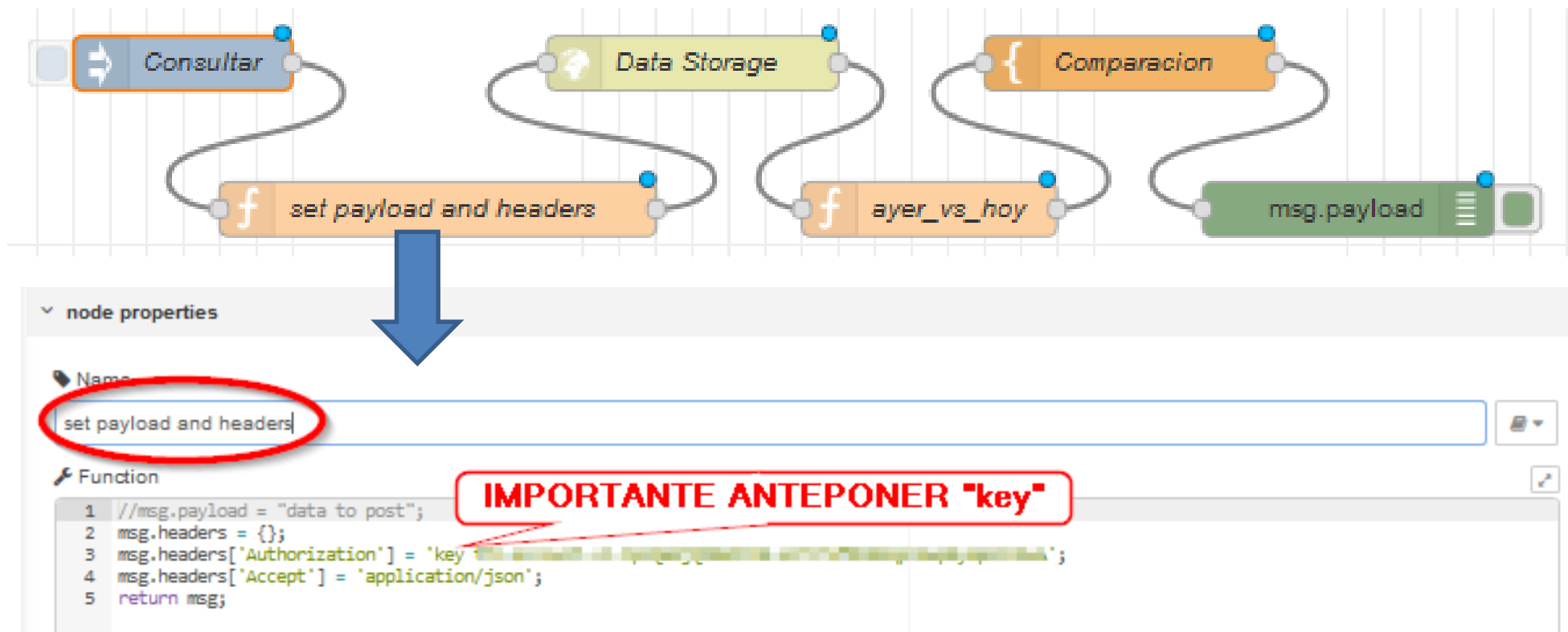
Node-RED & Data Storage

- Vamos a comparar la temperatura actual con la de hace 24 horas



Node-RED & Data Storage

- Vamos a comparar la temperatura actual con la de hace 24 horas



Node-RED & Data Storage

- Solicitamos los registros de las últimas 24 horas



Edit http request node

Delete Cancel Done

node properties

Method GET

URL https://jfmateos_taller_integraciones.data.thethingsnetwork.org/api/v2/query?last=48h

☐ Enable secure (SSL/TLS) connection

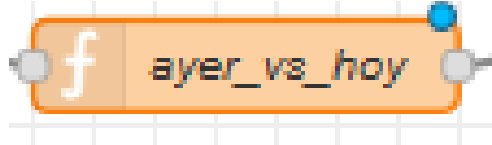
☐ Use basic authentication

Return a parsed JSON object

Name Data Storage

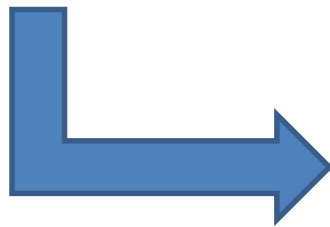
Tip: If the JSON parse fails the fetched string is returned as-is.

Node-RED & Data Storage



```
var i,registros,hoy,respuesta;respuesta={};
respuesta.payload={};
hoy=Date.now();
registros=msg.payload.length;
for(i=0;i<registros;i++){
    if(Date.parse(msg.payload[i].time) > (hoy - (24 * 60 * 60 * 1000))){
        respuesta.payload.temperatura_ayer=msg.payload[i].temperature_1;
        respuesta.payload.timestamp_ayer=Date.parse(msg.payload[i].time);
        respuesta.payload.temperatura_hoy=msg.payload[registros-1].temperature_1;
        respuesta.payload.timestamp_hoy=Date.parse(msg.payload[registros-1].time);
        respuesta.payload.intervalo=Math.floor((respuesta.payload.timestamp_hoy-
        respuesta.payload.timestamp_ayer)/(1000*60));
        respuesta.payload.deltaT=(respuesta.payload.temperatura_hoy -
        respuesta.payload.temperatura_ayer).toFixed(2);
        return respuesta;
    }
}
```

Node-RED & Data Storage



node properties

Name
Comparacion

☒ Set property msg. payload

☐ Format Mustache template

☒ Template Syntax Highlight: mustache

```
1 En los últimos {{payload.intervalo}} minutos, la temperatura ha variado {{payload.deltaT}}
2
```

Node-RED & ThingSpeak

- Creamos en ThinkSpeak un canal con 2 campos llamados Temperatura y Presión

ThingSpeak™

Channels ▾

Apps ▾

Community

Support ▾

integraciones_ttn

Channel ID: **659918**
Author: [jfmateos_tspk7](#)
Access: Private

[Private View](#) [Public View](#) [Channel Settings](#) [Sharing](#) [API Keys](#) [Data](#)

Channel Settings

Percentage complete

30%

Channel ID

659918

Name

integraciones_ttn

Description

Field 1

Temperatura

☒

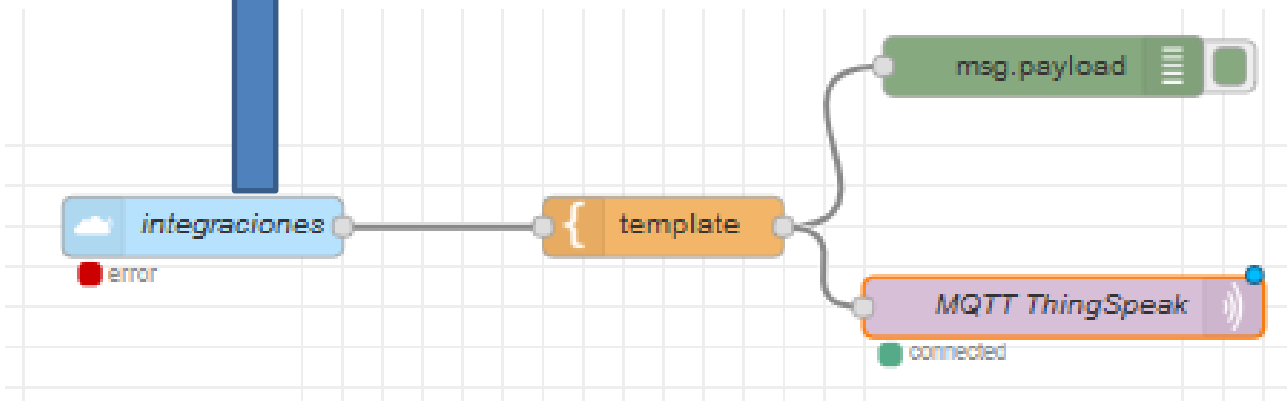
Field 2

Presion

☒

Node-RED & ThingSpeak

- Creamos en Node-RED el siguiente flujo



Edit ttn uplink node

Delete Cancel Done

node properties

Name integraciones

App jfmateos_taller_integraciones

Device ID

Field

Edit ttn uplink node > Edit ttn app node

Delete Cancel Update

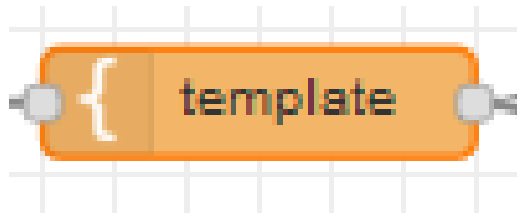
App ID jfmateos_taller_integraciones

Access Key

Discovery address discovery.thethingsnetwork.org:1900

Node-RED & ThingSpeak

- Usamos un nodo Template para crear el formato de carga de pago que requiere ThingSpeak



Dialog box titled "Edit template node" with buttons "Delete", "Cancel", and "Done".

node properties

Name
Name

☒ Set property msg. payload

Format
Mustache template

Template
Syntax Highlight: mustache

```
1 field1={{payload.temperature_1}}&field2={{payload.barometric_pressure_2}}
```

Node-RED & ThingSpeak

- Obtenemos nuestra key MQTT de ThingSpeak en Account > My Profile

ThingSpeak™ Channels Apps Community Support Commercial Use How to Buy Account Sign Out

My Profile

MathWorks Account settings

MathWorks Account Email	[redacted]
User ID	[redacted]
Password	*****

[Edit MathWorks Account settings](#)

[Edit MathWorks community information](#)

ThingSpeak settings

Time Zone	UTC Change Time Zone
User API Key	[redacted] Generate New API Key
MQTT API Key	[redacted] Generate New MQTT API Key

Help

MathWorks Account settings:

- MathWorks Account email and password are needed to Sign In to ThingSpeak.
- User ID is displayed as the author of your public channels.
- If you edit your MathWorks Account settings, you will need to sign out of ThingSpeak and log back in.

ThingSpeak Settings:

- Time Zone is used when displaying data in your charts, and when scheduling your ThingSpeak apps.
- User API key is required to create and manage channels using the REST API.
- MQTT API key is required to subscribe to channel topics using the MQTT API.

API Requests

Get Channel List

```
GET https://api.thingspeak.com/channels.json?api_key=[redacted]
```

Create a Channel

```
POST https://api.thingspeak.com/channels.json
api_key=[redacted]
name=My New Channel
```

Clear a Channel Feed

```
DELETE https://api.thingspeak.com/channels/CHANNEL_ID/feeds.json
api_key=[redacted]
```

Delete a Channel

```
DELETE https://api.thingspeak.com/channels/CHANNEL_ID
api_key=3UFx8YUPJY1T2DEL
```

Node-RED & ThingSpeak

- Configuramos el nodo MQTT output con los datos del bróker de ThingSpeak

Edit mqtt out node

Delete Cancel Done

node properties

Server thingspeak

Topic channels/ /publish/

QoS Retain

Name MQTT ThingSpeak

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

Edit mqtt out node > Edit mqtt-broker node

Delete Cancel Update

Name thingspeak

Connection Security Messages

Server mqtt.thingspeak.com 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

Edit mqtt out node > Edit mqtt-broker node

Delete Cancel Update

Name thingspeak

Connection Security Messages

Username jfmateos

Password

Vale cualquier nombre

Key MQTT

Node-RED & ThingSpeak

- Configuramos el Topic con el número del canal y su key de escritura

ThingSpeak™ Channels Apps Community Support

integraciones

Channel ID:

Author:

Access: Private

Private View Public View Channel Settings Sharing API Keys

Write API Key

Key

Generate New Write API Key

Edit mqtt out node

Delete

Topic channels/3/publish/

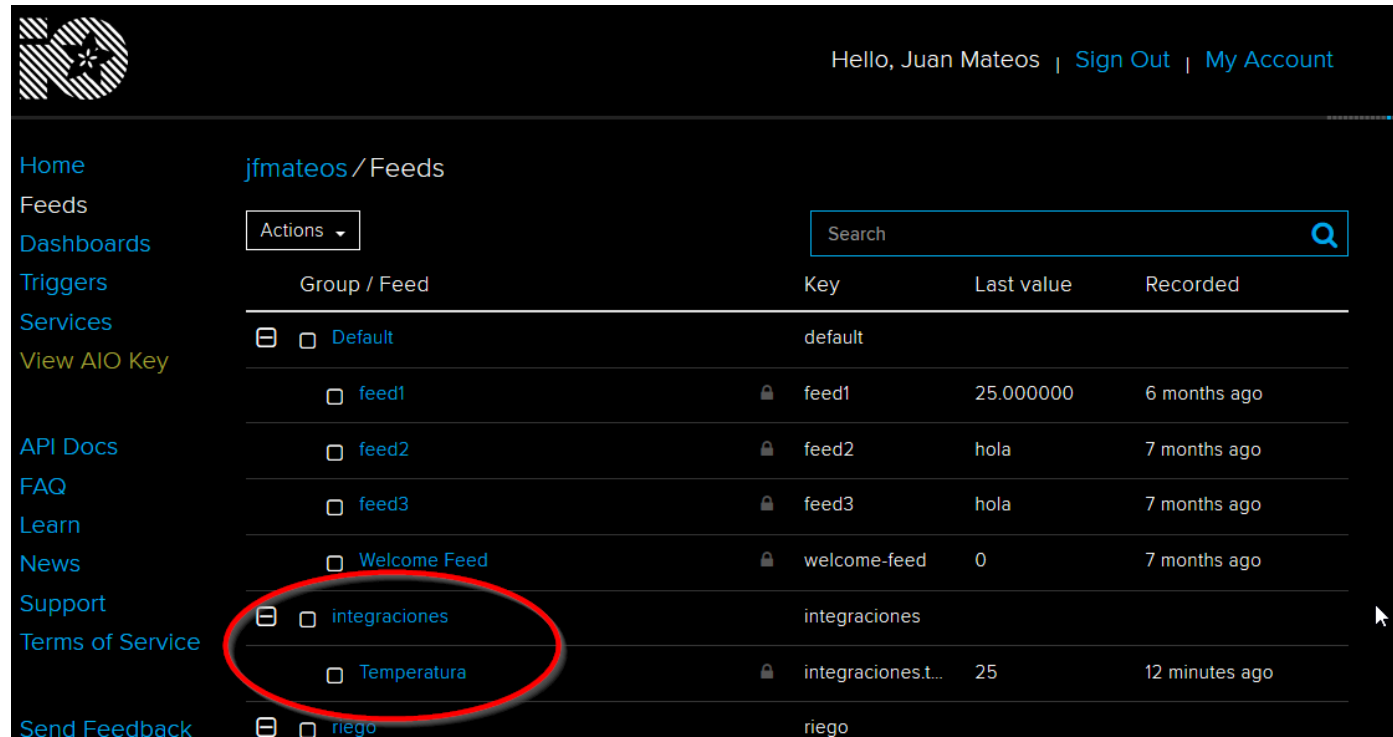
QoS Retain

Name MQTT ThingSpeak

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

Node-RED & Adafruit.io con MQTT

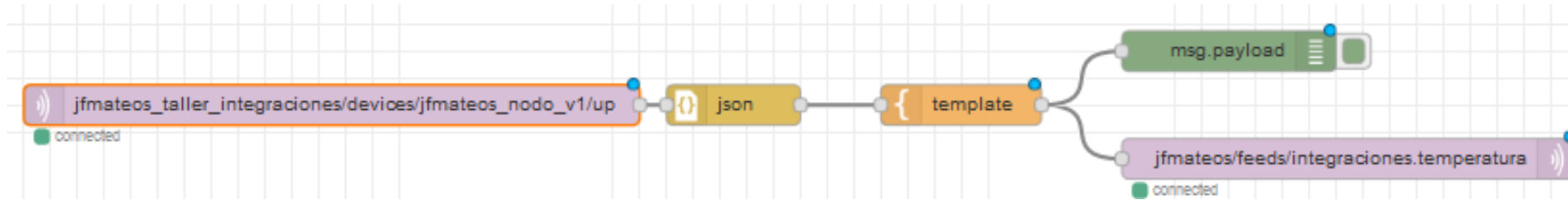
- Creamos un grupo
Integraciones con un **feed**
Temperatura en adafruit.io



Adafruit.io interface showing the 'jfmateos/Feeds' page. The page displays a list of feeds under the 'integraciones' group, including 'Default', 'feed1', 'feed2', 'feed3', 'Welcome Feed', 'integraciones', and 'Temperatura'. The 'Temperatura' feed is highlighted with a red circle, showing a last value of 25 recorded 12 minutes ago.

Group / Feed	Key	Last value	Recorded
Default	default		
feed1	feed1	25.000000	6 months ago
feed2	feed2	hola	7 months ago
feed3	feed3	hola	7 months ago
Welcome Feed	welcome-feed	0	7 months ago
integraciones	integraciones		
Temperatura	integraciones.t...	25	12 minutes ago
riego	riego		

Node-RED & Adafruit.io con MQTT



Edit mqtt in node > Edit mqtt-broker node

Delete Cancel Update

Name thethingsnetwork

Connection Security Messages

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

Edit mqtt in node > Edit mqtt-broker node

Delete Cancel Update

Name thethingsnetwork

Connection Security

Username jfmateos_taller_integraciones

Password

Edit mqtt in node

Delete Cancel Done

node properties

Server thethingsnetwork

Topic jfmateos_taller_integraciones/devices/jfmateos_nodo_v1/up

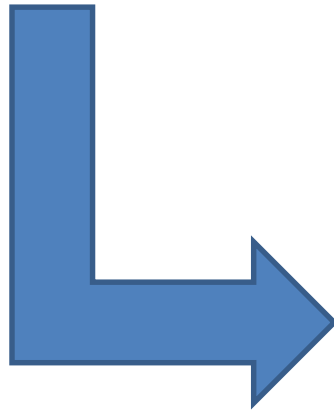
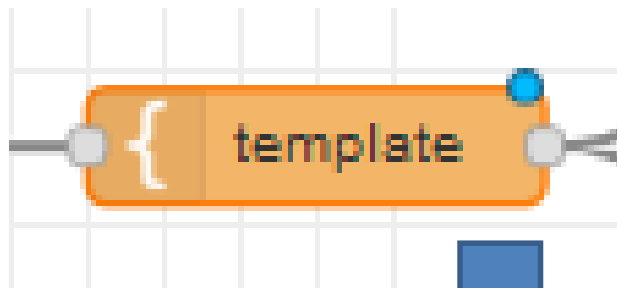
ID de aplicación

Id de dispositivo

ID de la Aplicación

Access key de la Aplicación

Node-RED & Adafruit.io con MQTT





Edit template node

DeleteCancelDone

node properties

Name




☒ Set property

▼ msg.payload

</> Format

Mustache template

▼

 Template

Syntax Highlight:

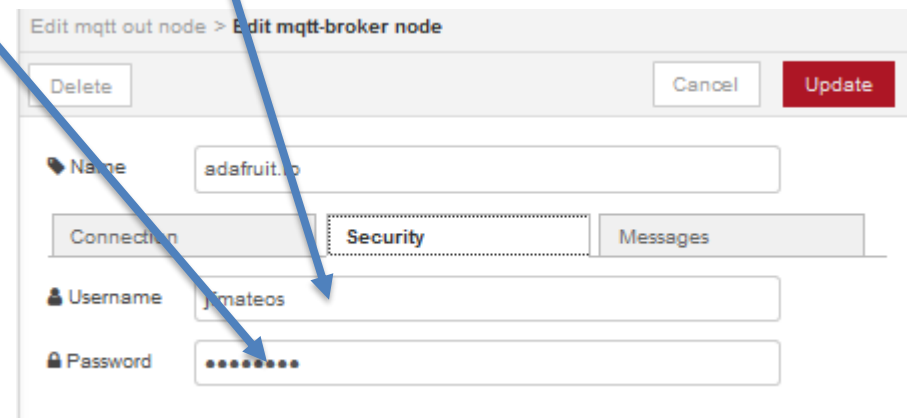
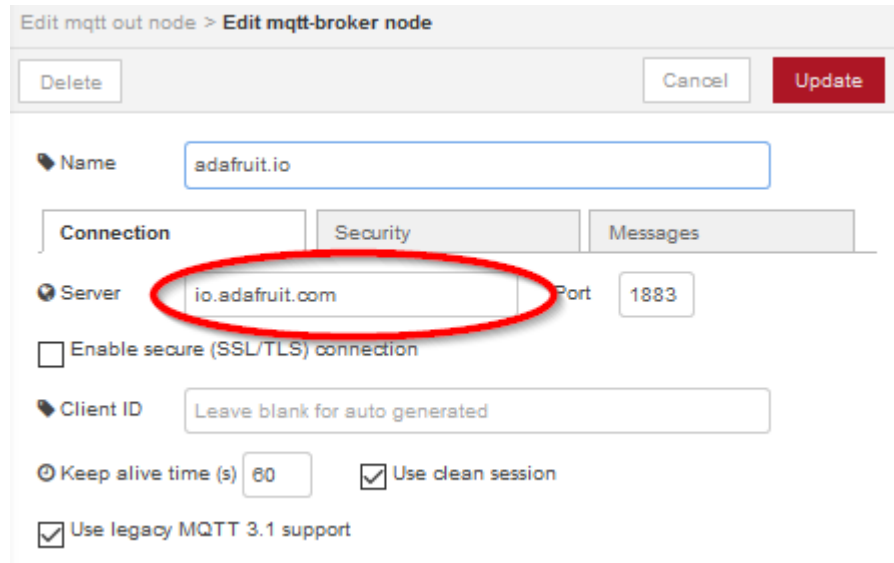
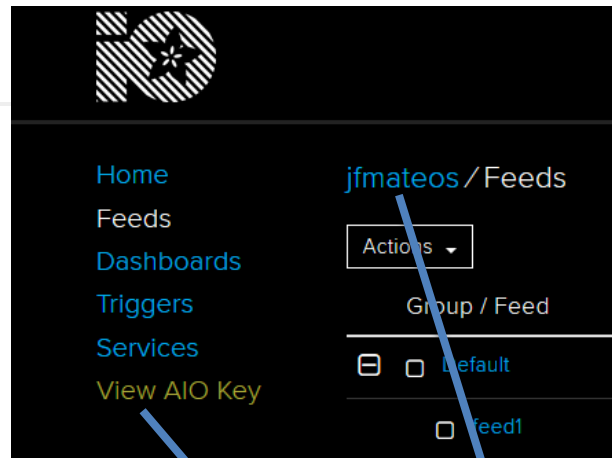
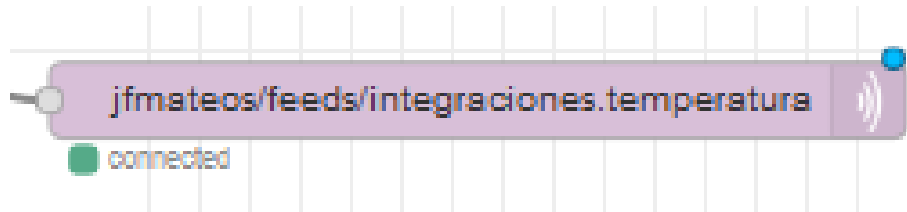
mustache

▼

1

```
{{payload.payload_fields.temperature_1}}
```

Node-RED & Adafruit.io con MQTT



TTN → IFTTT → Adafruit.io (sin Node-RED)

Configure



If Maker Event
"jfmateos_mi_integracion_ift
tt", then Send data to
integraciones - Temperatura
feed

98/140

[View activity log](#)

Receive notifications when
this Applet runs



Receive a web request

This trigger fires every time the
Maker service receives a web
request to notify it of an event. For
information on triggering events, go
to your Maker service settings and
then the listed URL (web) or tap your
username (mobile)

Event Name

jfmateos_mi_integracion_ift
t

The name of the event, like
"button_pressed" or "front_door_opened"



Send data to Adafruit IO

This Action will send data to a feed
in your Adafruit IO account.

Feed name

integraciones - Tempe ▼

The name of the feed to save data to.

Data to save

Value1

The data to be
saved to your feed.

Add ingredient

Save

HTTP Integration

- Podemos utilizar RequestBin para analizar el formato que utiliza TTN

 RequestBin

<http://requestbin.fullcontact.com/>

http://requestbin.fullcontact.com

POST /12yoxgr1

</> application/json

556 bytes

14s ago

From 52.169.225.45, 70.132.20.159

FORM/POST PARAMETERS

None

HEADERS

Connection: close
Total-Route-Time: 0
X-Request-Id: 65e495b3-6e7f-479a-81ad-012b2ff6abee
Cloudfront-Viewer-Country: IE
Host: requestbin.fullcontact.com
Content-Type: application/json
Cloudfront-Forwarded-Proto: http
X-Amz-Cf-Id: WzdfnfGfp7_Lu6L3vYkaUH-QHJ6Zd-OIRZnZ5FSc9-qz4Afs-26M1A==
User-Agent: http-ttn/2.6.0
Cloudfront-Is-Tablet-Viewer: false
Cloudfront-Is-Mobile-Viewer: false
Via: 1.1 f781469e78b7a441c6f692b1629e1519.cloudfront.net (CloudFront), 1.1 vegur
Accept-Encoding: gzip
Connect-Time: 0
Cloudfront-Is-Smarttv-Viewer: false
Content-Length: 556
Op-Viewer: true

RAW BODY

```
ie": "2018-12-21T13:42:54.028105122Z"}, "downlink_url": "https://integrations.thethingsnetwork.org/ttn-eu/api/v2/down/jfmateos_taller_integraciones/jfmateos_http_integration?key=ttn-accour
```

downlink_url

HTTP Integration

Por ejemplo, podemos crear un Trigger de tipo Reactive en Adafruit.io para enviar una petición a un webhook de IFTTT, en el que formatearemos los datos (la carga de pago debe ir en base64), y se los mandaremos a la dirección del downlink de TTN.

```
{ "dev_id": "jfmateos_nodo_v1",  
  "port": 1, "confirmed": false,  
  "payload_raw": "AQIDBA==" }
```

Edit Trigger

If

led_downlink

▼

Is

equal to

▼

Comparison Value or Fe...

▼

1

Then


send a webhook message ...

▼

<https://maker.ifttt.com/trig>

Cancel

Save



Make a web request

This action will make a web request to a publicly accessible URL. NOTE: Requests may be rate limited.

URL

```
https://integrations.thethingnetwork.org/ttn-eu/api/v2/down/jfmateos_taller_integraciones/jfmateos_http_integration?key=ttn-account-v2-jfmateos@thethingnetwork.org
```

Surround any text with "<<>>" to escape the content

POST

▼

The method of the request e.g. GET, POST, DELETE

Content Type (optional)

application/json

Optional

Body (optional)

```
[{"dev_id":
"jfmatsco_node_v1", "port":
1, "confirmed": false,
"payload_raw":
"AGIDBA=="}]
```

Surround any text with `"<code>"` to escape the content

Add ingredients

Save