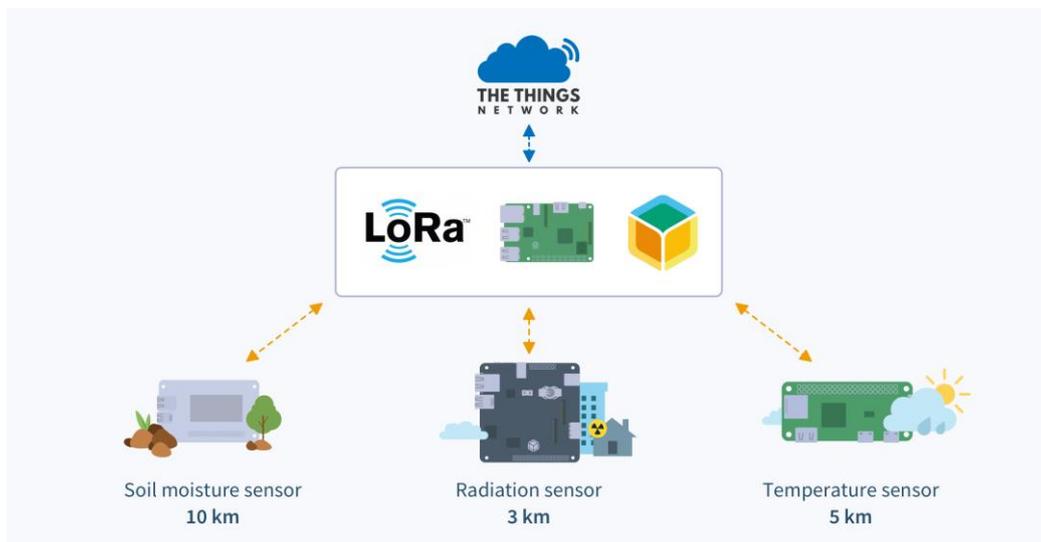


CREACIÓN DE NODOS LORA CON PLACAS DE DESARROLLO, ALTA EN THE THINGS NETWORK Y SINCRONIZACIÓN CON BALENAOS



Autor: Daniel Rodríguez Moya

Agradecimientos: Piarina Cañizo Otero, Alejandro Juan García y Marc Pous

Fecha: 18 de Julio de 2023

ÍNDICE DE CONTENIDOS

INTRODUCCIÓN.....	1
PRIMER PASO: CREACIÓN DE UN NODO LORA	3
SEGUNDO PASO - A: ALTA DEL NODO LORA EN EL SERVIDOR DE THE THINGS NETWORK - PROTOTIPO	4
SEGUNDO PASO - B: ALTA DEL NODO LORA EN EL SERVIDOR DE THE THINGS NETWORK - COMERCIAL	9
TERCER PASO: CONFIGURACIÓN A REALIZAR EN EL STACK DE MING PARA EL PROCESADO DE DATOS DEL NODO LORA.....	13
CONFIGURACIÓN DEL CONTAINER NODE-RED.....	13
CONFIGURACIÓN DEL NODO MQTT IN.....	15
CONFIGURACIÓN DEL NODO FUNCTION	18
CONFIGURACION DEL NODO INFLUXDB OUT.....	19
CONFIGURACIÓN DEL CONTAINER GRAFANA	22
CUARTO PASO: COMPARTIR PÚBLICAMENTE UN DASHBOARD.....	24

ÍNDICE DE ILUSTRACIONES

Ilustración 1: Logo de LoRa Alliance	1
Ilustración 2: MikroTik wAP LR8 kit	2
Ilustración 3: Diagrama de arquitectura de la infraestructura LoRa implementada en MediaLab explicado por Marc Pous	2
Ilustración 4: Despiece y ensamblado del hardware del nodo LoRa	4
Ilustración 5: Menú inicial de TTN	5
Ilustración 6: Menú de aplicaciones	5
Ilustración 7: Menú de la aplicación de ejemplo	6
Ilustración 8: Campos rellenados para registrar un prototipo basado en la placa Cubecell en registrar dispositivo.....	7
Ilustración 9: Menú del dispositivo recién registrado.....	7
Ilustración 10: “Payload formatters” para el proyecto de ejemplo	8
Ilustración 11: “Activation information” en formato array de bytes	8
Ilustración 12: Archivo ttnparams.h con los campos de “Activation information” de TTN rellenados...9	9
Ilustración 13: Recepción de los datos en TTN captados por el prototipo desarrollado	9
Ilustración 14: Dragino LHT65-E1	10
Ilustración 15: Ejemplo de código “QR” en la caja de un nodo Dragino	10
Ilustración 16: Menú “Manually” para el registro de un nodo LoRa	11
Ilustración 17: Página web de Dragino para el sensor LHT65-E1	11
Ilustración 18: Página “Documents -> User Manual”	12
Ilustración 19: Lista de versiones del documento “UserManual”	12
Ilustración 20: “Decoder” para TTN en el documento “UserManual”	12
Ilustración 21: Datos del nodo Dragino LHT65-E1 siendo recibidos en TTN	12
Ilustración 22: Ventana emergente de “projects” en Node-RED	13
Ilustración 23: Creación y asignación de nombre a un nuevo flujo.....	14

Ilustración 24: Nodos de Node-RED dispuestos pero sin conectarse aún.....	14
Ilustración 25: Ventana emergente principal del nodo mqtt in en la que se debe rellenar el campo de “Topic” y, después, hacer clic en el botón del lápiz	15
Ilustración 26: Ventana “Connection” en la que se debe introducir el campo “Server” desde la integración MQTT de TTN y, después, acceder a “Security”	15
Ilustración 27: Menú de “API Key” de TTN en la que se tendrá que generar una nueva key y copiarla al protapapeles ya que será la única vez que esté disponible su acceso	16
Ilustración 28: Menú “Security” en el que se debe introducir la “Application ID” de TTN como “Username” y la “API Key” del portapapeles en “Password”	17
Ilustración 29: Nodo mqtt in conectado correctamente a la integración MQTT de TTN.....	17
Ilustración 30: Información recibida por MQTT en Node-RED y mostrada por el primer nodo “debug”	18
Ilustración 31: Variables filtradas tras el nodo “function”	19
Ilustración 32: Menú “Data Sources” de Grafana.....	19
Ilustración 33: Configuración para una nueva base de datos de nombre "medialab_db" que funcionará como predeterminada.....	20
Ilustración 34: Ventana emergente principal del nodo “influxdb out”	21
Ilustración 35: Campos del apartado “Server” del nodo “influxdb out”	21
Ilustración 36: Flujo de Node-RED para el ejemplo del tutorial completamente configurado.....	22
Ilustración 37: Menú “Dashboards -> Manage” de Grafana.....	22
Ilustración 38: Nuevo “dashboard”.....	23
Ilustración 39: Campos de “Query” rellenos para graficar la variable “battery”	23
Ilustración 40: “Dashboard” de Grafana completamente configurado	24
Ilustración 41: “Dashboard” Ejemplo Tutorial guardado y añadido a la lista	24
Ilustración 42: Franja temporal configurada para mostrar los últimos 30 minutos	25
Ilustración 43: Código “QR” para acceder al “dashboard” Ejemplo Tutorial.....	25

ÍNDICE DE FIGURAS

Figura 1: Código del nodo “function” para el filtrado de las variables del sensor	18
---	----

INTRODUCCIÓN

Este tutorial desarrollará una explicación paso a paso en la creación de nodos LoRa, el alta en The Things Network y su sincronización en el servicio de procesamiento de datos de IoT de balenaOS. Cabe destacar que este documento es el principal dentro de una colección de tres tutoriales que dan cuenta, además, de la instalación de un GateWay LoRa y de la implementación de balenaOS y el *stack* de MING con títulos, respectivamente, *Configuración y registro en The Things Network de un GateWay LoRa e Instalación de balenaOS y MING Stack en una Raspberry PI 3 para acelerar aplicaciones de IoT con LoRa*.

Antes de comenzar, es importante entender la red de comunicaciones a emplear, LoRaWAN, cuyo nombre proviene de *Long Range* y se engloba entre las tecnologías LPWAN o *Low Power Wide Area Network*. Administrada por la LoRa Alliance, que certifica a todo fabricante que desee trabajar con esta tecnología, se sustenta en los siguientes pilares:

- Alta tolerancia a las interferencias
- Alta sensibilidad para recibir datos (-168 dB)
- Basado en modulación *chirp*
- Bajo consumo (dispositivos con autonomía de hasta 10 años con una carga de batería)
- Largo alcance (10 a 20 km)
- Baja transferencia de datos (hasta 255 bytes)
- Conexión punto a punto
- Frecuencia de trabajo en Europa 868 MHz

En resumidas cuentas, si la aplicación de IoT que se desea implementar requiere alejarse del punto de conexión y la transferencia de datos va a ser limitada, además de que el acceso a una red eléctrica pueda ser una complicación, LoRa resulta una elección muy oportuna.



Ilustración 1: Logo de LoRa Alliance

Por otra parte, es de especial interés dar a conocer el componente electrónico fundamental para implementar una red red LoRa: el *GateWay*. Este dispositivo se encarga de recibir o *escuchar* todo aquello que publican los nodos, que es el nombre que reciben todos los dispositivos de medición y adquisición de datos. En el caso de la infraestructura LoRa implementada en MediaLab Universidad de Oviedo, el modelo elegido ha sido el MikroTik LR8 kit:



Ilustración 2: MikroTik wAP LR8 kit

Una vez éste recibe la información, carga los datos en internet. Por tanto, el *GateWay* debe conectarse a un router de internet por cable ethernet.

Una vez en internet, es fundamental que el usuario registre los datos captados en alguna parte. Siguiendo con las elecciones de MediaLab, se escogió la plataforma de The Things Network (TTN), una red de servidores que permite conectar *GateWays* y registrar los datos por los diferentes nodos LoRa. Además, da soporte a integraciones como MQTT, la cual será una pieza clave más adelante en este tutorial.

Finalmente, y debido a la falta de un servicio flexible y gratuito de persistencia y procesamiento de datos desde TTN, se debe crear un dispositivo con el software de balenaOS y el *stack* de MING para guardar y graficar los datos captados.

El diagrama de arquitectura de la infraestructura es el siguiente:

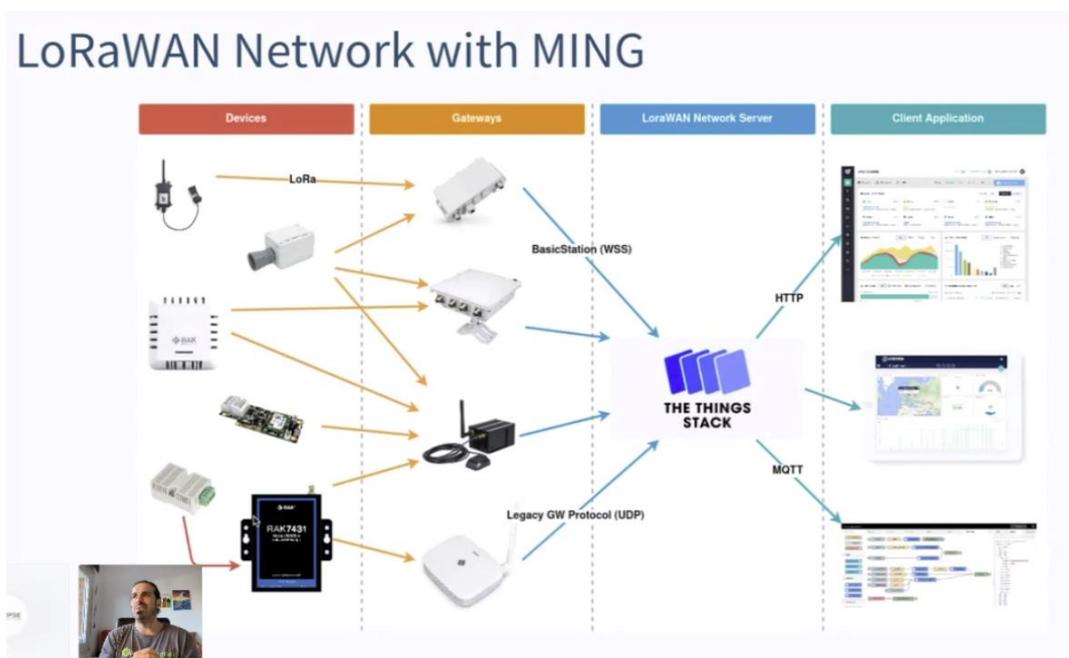


Ilustración 3: Diagrama de arquitectura de la infraestructura LoRa implementada en MediaLab explicado por Marc Pous

Por todo ello, el material del que se deberá disponer es el siguiente:

- Placa de desarrollo Heltec Cubecell Dev-Board HTCC-AB01
- Componentes electrónicos que se quieran conectar a la placa de desarrollo
- *GateWay* LoRa Mikrotik Wap LR8 kit
- Router de internet
- 2 cables de ethernet
- Raspberry Pi 3
- Una microSD
- Una cuenta en The Things Network
- Una cuenta en balena

PRIMER PASO: CREACIÓN DE UN NODO LORA

Este primer paso ha tomado nota de las instrucciones dadas en el blog de ALEX CORVIS, en la entrada *Review: HelTec CubeCell Dev-Board HTCC-AB01 + Proyecto de Estación Meteorológica*: [Review: HelTec CubeCell DevBoard + Proyecto Estación Meteorológica](#) 🔗💧 (alexcorvis.cc)

Brevemente, lo primero es desarrollar un programa para el microcontrolador Heltec Cubecell HTCC-AB01. Esta placa de desarrollo se convierte en una aliada genial para aprender a prototipar aplicaciones con LoRaWAN ya que cuenta con su propio emisor LoRa y *Sleep Mode* para reducir al mínimo el consumo de batería. Se recomienda el uso de la extensión de PlatformIO para VS Code a la hora de escribir los programas, ya que supone una herramienta mucho más versátil que el propio Arduino IDE.

En este caso, se ha seguido la guía del blog para hacer una pequeña estación meteorológica con un sensor I2C BME280 que capta humedad, presión y temperatura. Como añadido, se ha conectado una batería de 1800 mAh directamente al puerto para baterías de la placa de desarrollo que deberá dar una autonomía suficiente para varios meses con una sola carga y se ha diseñado una carcasa en impresión 3D. Una vez se cablea y se ensambla el proyecto, disponible en el GitHub de Alex ([GitHub-alexcorvis84/mini-lora-weatherstation: A mini battery powered LoRa weather station to measure temperature, humidity and barometric pressure.](#)), se observa el siguiente resultado:

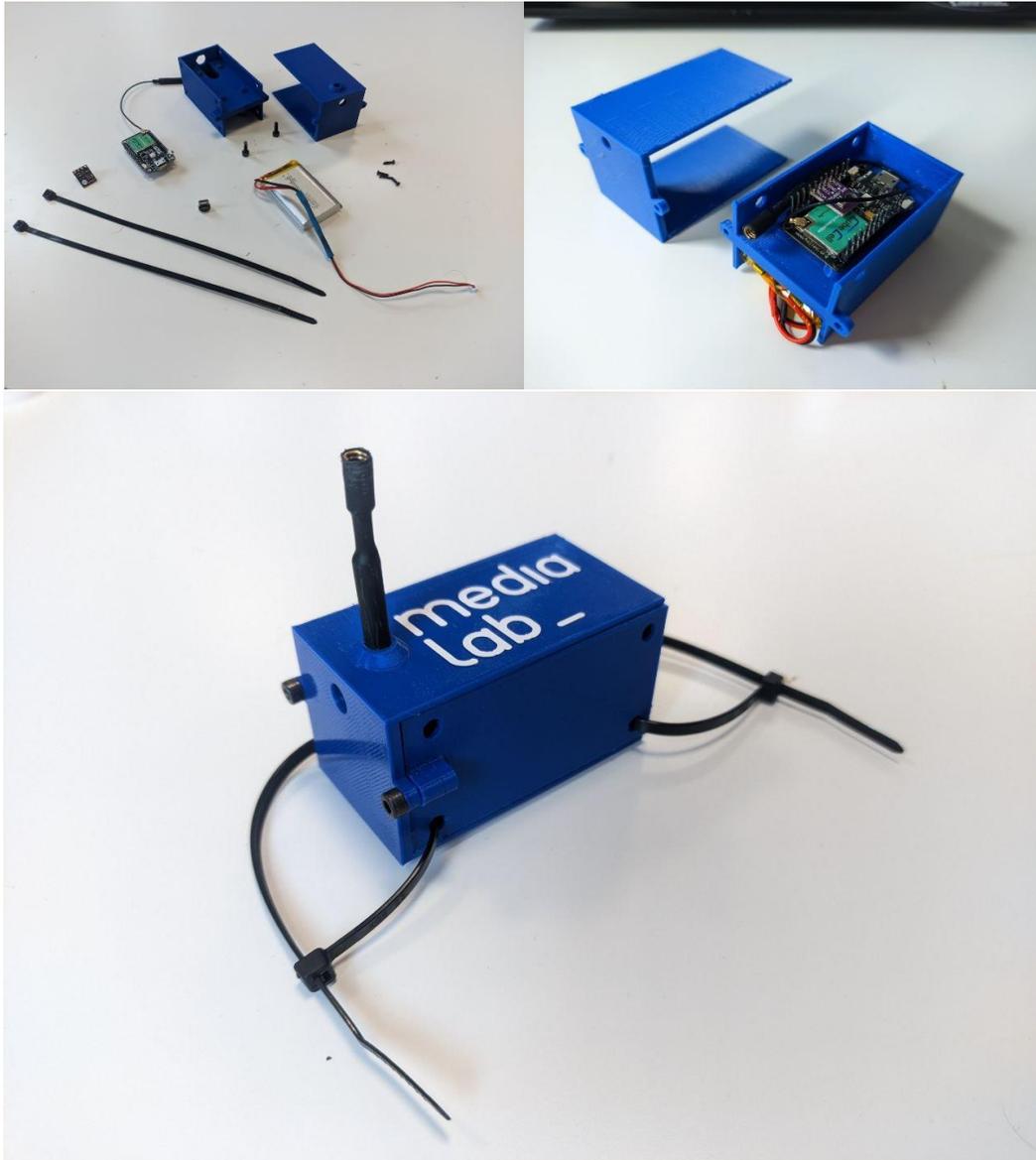


Ilustración 4: Despiece y ensamblado del hardware del nodo LoRa

SEGUNDO PASO - A: ALTA DEL NODO LORA EN EL SERVIDOR DE THE THINGS NETWORK - PROTOTIPO

Este segundo paso ha tomado nota de las instrucciones dadas en el TFM de Piarina Cañizo Otero sobre *Estudio y comparativa de redes IoT desplegadas sobre LoRa y 5G: Estudio de redes*: [Microsoft Word - TFM MINGTELE Memoria Piarina Cañizo Otero 2022 Junio.docx \(uniovi.es\)](#).

En este momento, se debe abrir la consola de TTN para la región europea (<https://eu1.cloud.thethings.network/>) e iniciar sesión con la cuenta de MediaLab (pedir las credenciales a Ramón). Una vez dentro y teniendo en cuenta que ya se ha realizado previamente la configuración y sincronización entre el servidor y el *GateWay* (consúltese la guía *Configuración y registro en The Things Network de un GateWay LoRa*), se debe acceder al menú de aplicaciones haciendo clic en *Go to applications*:

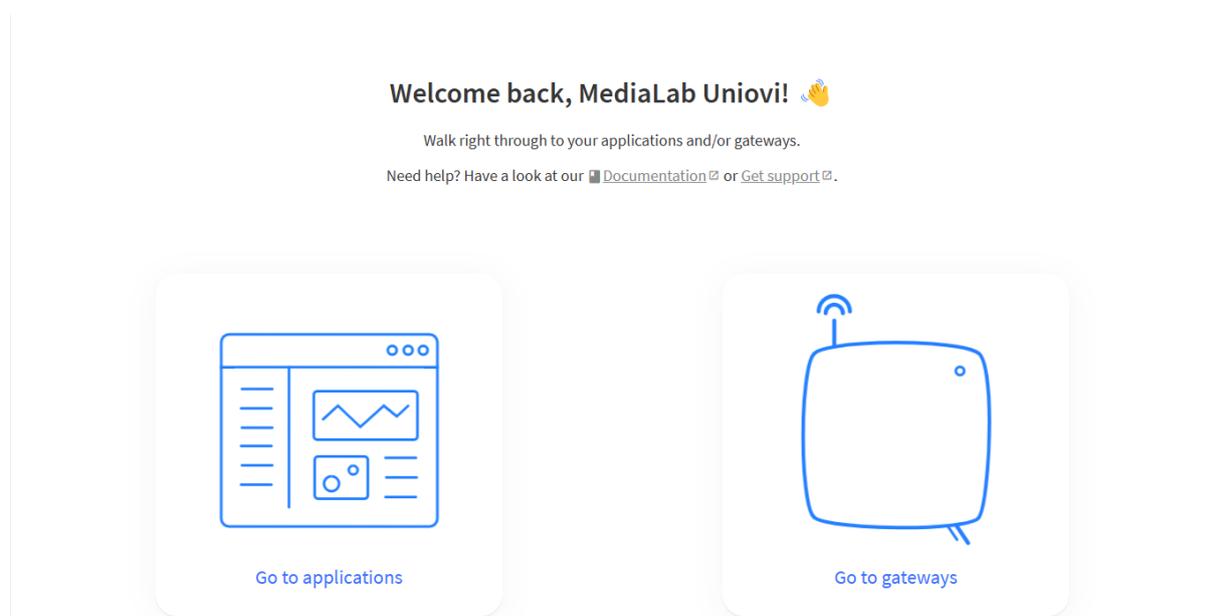


Ilustración 5: Menú inicial de TTN

Applications (14)

Search [+ Create application](#)

ID	Name	End devices	Created at
vaquer-tracker-2023	Crotal geolocalizador de ganado VaquerTracker	3	Jun 15, 2023
cubecell-ccs811-uo265941	Cubecell CCS811 UO265941	1	May 8, 2023
cubecell-bmp280-uo265941	Cubecell BMP280 UO265941	1	Apr 19, 2023
lgt-92-medialab	GPS LGT-92 DRAGINO	1	Feb 24, 2023
lht65-e1-uo265941	SENSOR LHT65 DRAGINO	1	Feb 24, 2023
dispositivo-cueli	dispositivo-cueli	1	Dec 19, 2022
vaquertracker	Vaquer Tracker 1.0	2	Jul 12, 2022
vp2-medialab	VP2_MediaLab	1	May 30, 2022
testmedialab	test	7	Mar 14, 2022

Ilustración 6: Menú de aplicaciones

Desde el menú de aplicaciones, lo primero es crear una nueva aplicación haciendo clic sobre el botón **+ Create application** (parte superior derecha en la *ilustración 6*). Una aplicación representa un conjunto de dispositivos o nodos al que el usuario le quiere dar un fin relacionado.

Una vez en el menú de creación de una nueva aplicación, se rellenan los campos y se hace clic en **Create application**. Tras ello, aparecerá el menú de la aplicación con la información que se acaba de aportar. El siguiente paso es añadir un *end device*, que representa el nodo que acabamos de desarrollar con la placa Cubecell. Para ello, se hace clic en el botón **+ Register end device** situado en la esquina inferior derecha:

The screenshot shows the 'ejemplo tutorial' application page in the The Things Network console. The left sidebar contains a navigation menu with options: Overview, End devices, Live data, Payload formatters, Integrations, Collaborators, API keys, and General settings. The main content area displays the application details for 'ejemplo tutorial' (ID: ejemplo-tutorial). It shows 'No recent activity' and '0 End devices'. The 'General information' section includes fields for Application ID, Created at, and Last updated at. The 'Live data' section shows a single log entry: '18:07:52 ejemplo-tu... Create application'. At the bottom, there are buttons for 'Import end devices' and 'Register end device'.

Ilustración 7: Menú de la aplicación de ejemplo

Una vez se accede al menú de para registrar un dispositivo y, como se trata de un prototipo, se deben generar las credenciales *JoinEUI*, *DevEUI* y *AppKey* en *Provisioning information* y rellenar los campos de la siguiente manera:

The screenshot shows the 'Register end device' form in the The Things Network console. The form is titled 'Register end device' and includes a QR code scanner option. The 'End device type' section has two radio buttons: 'Select the end device in the LoRaWAN Device Repository' (selected) and 'Enter end device specifics manually'. Below this, there are dropdown menus for 'End device brand' (HelTec AutoMation), 'Model' (HTCC-AB01(Class A ...)), 'Hardware Ver.' (Unknown ..), 'Firmware Ver.' (1.0), and 'Profile (Region)' (EU_863_870). A detailed description of the 'HTCC-AB01(Class A OTAA)' device is provided, including a small image of the board and links to the 'Product website' and 'Data sheet'.

ejemplo tutorial

Frequency plan [ⓘ] *
Europe 863-870 MHz (SF9 for RX2 - recommended)

Provisioning information

JoinEUI [ⓘ] *
00 01 11 22 23 33 44 45
This end device can be registered on the network

DevEUI [ⓘ] *
70 B3 D5 7E D0 05 FA 1F 1/50 used

AppKey [ⓘ] *
09 10 93 FB 50 23 39 2A 30 63 20 A0 F0 E0 B4 63

End device ID [ⓘ] *
ejemplo-tutorial
This value is automatically prefilled using the DevEUI

After registration

View registered end device
 Register another end device of this type

Ilustración 8: Campos rellenados para registrar un prototipo basado en la placa Cubecell en registrar dispositivo

Tras el correcto registro, aparece el menú propio del dispositivo:

ejemplo tutorial

Overview

End devices

Live data

Payload formatters

Integrations

Collaborators

API keys

General settings

Hide sidebar

Applications > ejemplo tutorial > ejemplo-tutorial > End devices

ejemplo-tutorial
ID: ejemplo-tutorial

↑ n/a ↓ n/a No activity yet [ⓘ]

Overview Live data Messaging Location Payload formatters General settings

General information

End device ID ejemplo-tutorial

Frequency plan Europe 863-870 MHz (SF9 for RX2 - recommended)

LoRaWAN version LoRaWAN Specification 1.0.2

Regional Parameters version RP001 Regional Parameters 1.0.2 revision B

Created at Jul 18, 2023 23:52:24

Hardware

Brand heltec

Model cubecell-dev-board-class-a-otaa

Hardware version _unknown_hw_version_

Firmware version 1.0

Activation information

AppEUI 00 01 11 22 23 33 44 45

DevEUI 70 B3 D5 7E D0 05 FA 1F

AppKey

Live data

23:52:24 Create end device

See all activity --

Location

Change location settings --

No location information available

Ilustración 9: Menú del dispositivo recién registrado

En él, se debe configurar un aspecto fundamental, los *Payload formatters*. Para ello, se debe acceder a la pestaña con el mismo nombre debajo del propio nombre del dispositivo, como se puede ver en la *ilustración 9*. Se debe de escoger, en el caso del proyecto de ejemplo, *Custom Javascript formatter* y, en él, se debe pegar el siguiente código que forma parte del proyecto de PlatformIO:

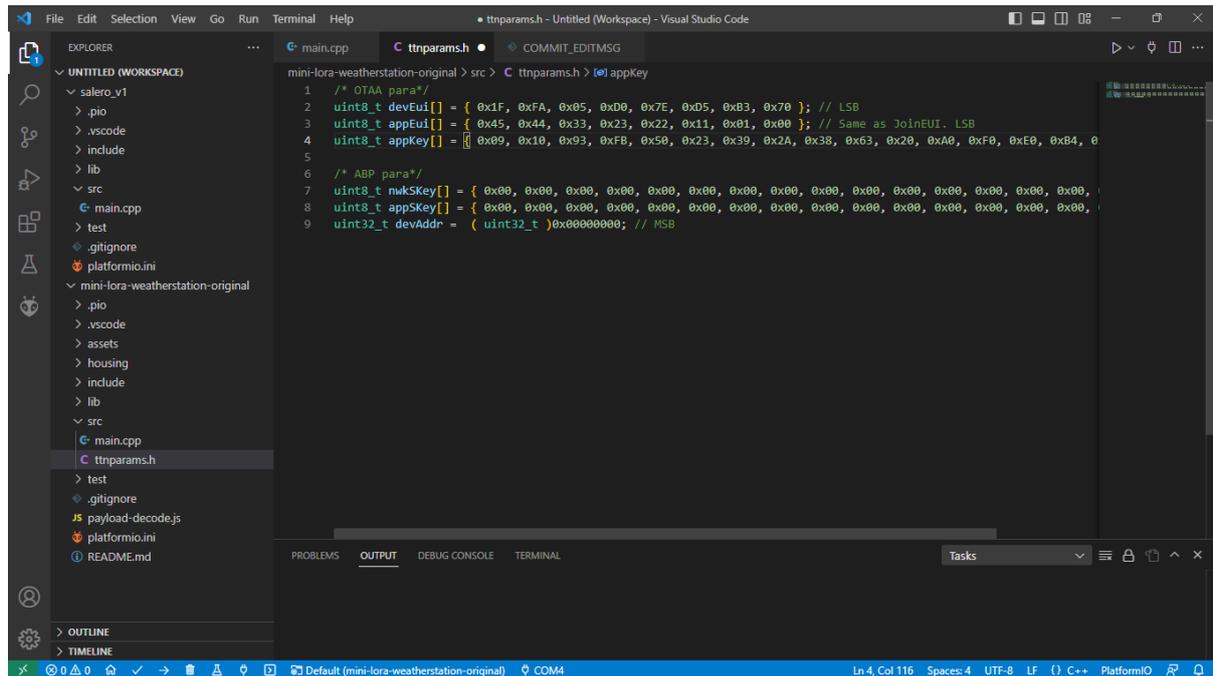
Ilustración 10: "Payload formatters" para el proyecto de ejemplo

Una vez realizado lo anterior, se debe hacer clic en *Save changes*. Explicado de forma superficial, este paso es necesario ya que, debido a las limitaciones en el volumen de envío de datos por LoRa, en el código de VS Code del proyecto se debe realizar una descomposición a bytes para optimizar; por lo que en TTN y gracias a los *payload formatters*, se reconstruyen las variables para que vuelvan a poder a ser leídas de igual forma que en local por el usuario antes de ser enviadas desde el código de VS Code a TTN.

Finalizando la configuración del nodo, se debe volver a la pestaña de *Overview*, a la izquierda de *Payload formatters*, y convertir a array de bytes de tipo *lsb* para la *AppEUI* y la *DevEUI* y *msb* para la *AppKey* los campos del apartado de *Activation information*.

Ilustración 11: "Activation information" en formato array de bytes

Cada una de ellas debe ser copiada respectivamente en el archivo del proyecto en VS Code de nombre *ttnparams.h* en la sección *OTAA*.



```

1  /* OTAA para*/
2  uint8_t devEui[] = { 0x1F, 0xFA, 0x05, 0x00, 0x7E, 0x05, 0xB3, 0x70 }; // LSB
3  uint8_t appEui[] = { 0x45, 0x44, 0x33, 0x23, 0x22, 0x11, 0x01, 0x00 }; // Same as JoinEUI. LSB
4  uint8_t appKey[] = { 0x09, 0x10, 0x93, 0xFB, 0x50, 0x23, 0x39, 0x2A, 0x38, 0x63, 0x20, 0xA0, 0xF0, 0xE0, 0xB4, 0x00 };
5
6  /* ABP para*/
7  uint8_t nwkSKey[] = { 0x00, 0x00 };
8  uint8_t appSKey[] = { 0x00, 0x00 };
9  uint32_t devAddr = ( uint32_t )0x00000000; // MSB

```

Ilustración 12: Archivo *ttnparams.h* con los campos de "Activation information" de TTN rellenos

Es ahora cuando se debe flashear el proyecto en la placa desde PlatformIO.

Con ésto completado y accediendo a la pestaña *Live data*, situada donde las anteriormente mencionadas, se puede observar que el envío de datos está siendo según lo previsto:

Time	Type	Data preview
↑ 18:13:42	Forward uplink data message	DevAddr: 26 08 08 E6 < > Payload: { battery: 3.758, humidity: 60, pressure: 1019.95, temperature: 25.23 }
↑ 18:13:42	Successfully processed data message	DevAddr: 26 08 08 E6 < >
↑ 18:13:18	Forward uplink data message	DevAddr: 26 08 08 E6 < > Payload: { battery: 3.758, humidity: 60, pressure: 1019.91, temperature: 25.21 }
↑ 18:13:18	Successfully processed data message	DevAddr: 26 08 08 E6 < >
↑ 18:12:39	Forward uplink data message	DevAddr: 26 08 08 E6 < > Payload: { battery: 3.758, humidity: 60, pressure: 1019.89, temperature: 25.18 }
↑ 18:12:39	Successfully processed data message	DevAddr: 26 08 08 E6 < >
↑ 18:12:07	Forward uplink data message	DevAddr: 26 08 08 E6 < > Payload: { battery: 3.758, humidity: 60, pressure: 1019.91, temperature: 25.17 }
↑ 18:12:07	Successfully processed data message	DevAddr: 26 08 08 E6 < >

Ilustración 13: Recepción de los datos en TTN captados por el prototipo desarrollado

SEGUNDO PASO - B: ALTA DEL NODO LORA EN EL SERVIDOR DE THE THINGS NETWORK - COMERCIAL

Con el proceso sabido de dar de alta un nodo prototipado en una placa de desarrollo en TTN, hacerlo con uno comercial es un proceso mucho más sencillo. Para este ejemplo, se ha elegido el sensor de temperatura y humedad LHT65-E1 del fabricante Dragino:



Ilustración 14: Dragino LHT65-E1

Para darlo de alta en una cuenta de TTN, se debe repetir el proceso del paso alternativo para prototipos, pero, simplemente, escaneando el código QR que incluye en su caja:



Ilustración 15: Ejemplo de código "QR" en la caja de un nodo Dragino

Cabe mencionarse que, como se puede observar en la *ilustración 15*, también se puede dar de alta desde la pestaña *Manually* introduciendo la información correspondiente al plan de frecuencias de la región europea, versión de LoRaWAN y credenciales correspondientes:

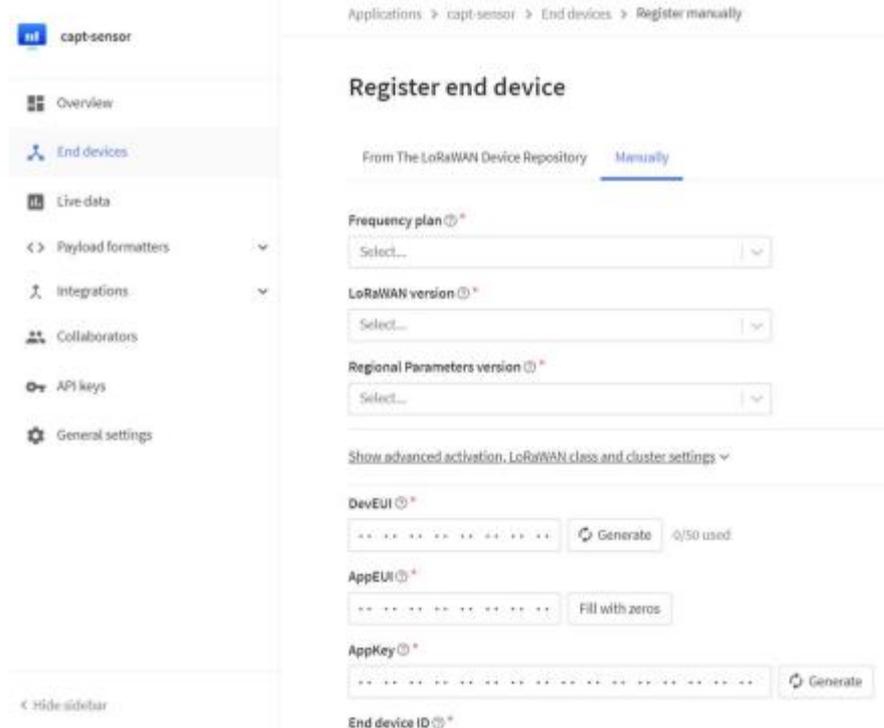


Ilustración 16: Menú "Manually" para el registro de un nodo LoRa

A continuación, se debe introducir el *payload formatter* al igual que con el prototipo con la diferencia de que, con un nodo comercial de Dragino, se puede copiar desde la documentación aportada en la página web del producto ([LHT65 LoRaWAN Temperature & Humidity Sensor \(dragino.com\)](https://www.dragino.com/en/products/temperature-humidity-sensor/lht65)):

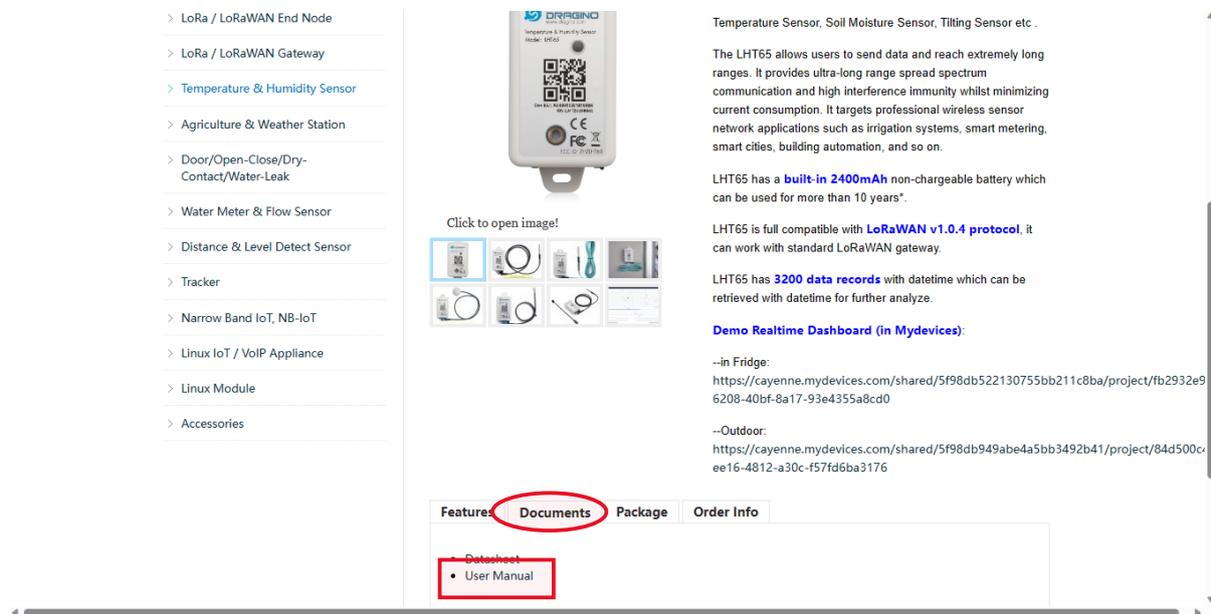


Ilustración 17: Página web de Dragino para el sensor LHT65-E1

File	Size	Modified
Parent Directory		
Certificate	15.9 MB	2021-Oct-05
Firmware	17.7 MB	2023-May-26
payload_decode	11.1 KB	2022-Mar-24
Test_Report	5.6 MB	2020-Nov-21
UserManual	11.1 MB	2022-Jan-26
datasheet_LHT65_LoRaWAN_Temperature_Humidity_Sensor.pdf	457.7 KB	2022-Feb-08

1 Files - 5 Folders Total size: 50.8 MB

Ilustración 18: Página "Documents -> User Manual"

File	Size	Modified
Parent Directory		
DRAGINO_LHT65_AT_Commands_v1.0.pdf	559.2 KB	2019-Jun-26
LHT65_Temperature_Humidity_Sensor_UserManual_v1.3.pdf	1.5 MB	2019-Dec-29
LHT65_Temperature_Humidity_Sensor_UserManual_v1.7.3.pdf	2.3 MB	2020-Jul-15
LHT65_Temperature_Humidity_Sensor_UserManual_v1.8.3.pdf	2.4 MB	2021-May-10
LHT65_Temperature_Humidity_Sensor_UserManual_v1.8.4.pdf	2.4 MB	2021-Jul-14
LHT65_Temperature_Humidity_Sensor_UserManual_v1.8.5.pdf	2.0 MB	2022-Jan-26

6 Files - 0 Folders Total size: 11.1 MB

Ilustración 19: Lista de versiones del documento "UserManual"

```
function Decoder(bytes, port) {
  // Decode an uplink message from a buffer
  // (array) of bytes to an object of fields.
  var value=(bytes[0]<<8 | bytes[1]) & 0x3FFF;
  var batV=value/1000;//Battery,units:V

  value=bytes[2]<<8 | bytes[3];
  if(bytes[2] & 0x80)
  {value |= 0xFFFF0000;}
  var temp_SHT=(value/100).toFixed(2);//SHT20,temperature,units:°C

  value=bytes[4]<<8 | bytes[5];
  var hum_SHT=(value/10).toFixed(1);//SHT20,Humidity,units:%

  value=bytes[7]<<8 | bytes[8];
  if(bytes[7] & 0x80)
  {value |= 0xFFFF0000;}
  var temp_DS=(value/100).toFixed(2);//DS18B20,temperature,units:°C

  return {
    BatV:batV,
    TempC_DS:temp_DS,
    TempC_SHT:temp_SHT,
    Hum_SHT:hum_SHT
  };
}
```

LHT65 Temperature & Humidity sensor 10 / 40

Ilustración 20: "Decoder" para TTN en el documento "UserManual"

Acudiendo a la pestaña *Live data* de TTN, se observa que ya están llegando los datos de las distintas mediciones:

787 ↓ 34 • Last activity 3 minutes ago

Overview **Live data** Messaging Location Payload formatters General settings

Time	Type	Data preview
09:55:48	Forward uplink data message	DevAddr: 26 0B 67 1D Payload: { BatV: 3.064, Hum_SHT: 55.4, TempC_DS: 23.56, TempC_SHT: 23.76 }
09:55:48	Successfully processed dat...	DevAddr: 26 0B 67 1D

Ilustración 21: Datos del nodo Dragino LHT65-E1 siendo recibidos en TTN

TERCER PASO: CONFIGURACIÓN A REALIZAR EN EL STACK DE MING PARA EL PROCESADO DE DATOS DEL NODO LORA

Una vez más, se recuerda que, ya que en este paso se hará uso en profundidad de los servicios ofrecidos por el stack MING instalado en balenaOS gestionado desde balenaCloud, se recomienda una lectura al tutorial pertinente a esta materia: *Instalación de balenaOS y MING Stack en una Raspberry Pi 3 para acelerar aplicaciones de IoT con LoRa*.

De este modo, se recomienda empezar a trabajar con dos ventanas de navegador abiertas a la vez ya que se copiarán en el Node-RED instalado en balenaOS una serie de elementos que vienen dados en TTN para la conexión por MQTT.

Con ello, se debe acceder al *dashboard* de balenaCloud, entrar en la flota de *espacio-medialab* y elegir el dispositivo *RaspPi-MediaLab*. Una vez aquí, se accede al hipervínculo del apartado de *PUBLIC DEVICE URL*, que abrirá una pestaña nueva en el navegador con Node-RED en la que se debe iniciar sesión con las credenciales asignadas en el balenaCloud en el apartado de *Device Variables*.

CONFIGURACIÓN DEL CONTAINER NODE-RED

Una vez abra Node-RED, se hace clic en *Not right now* en la siguiente ventana emergente:

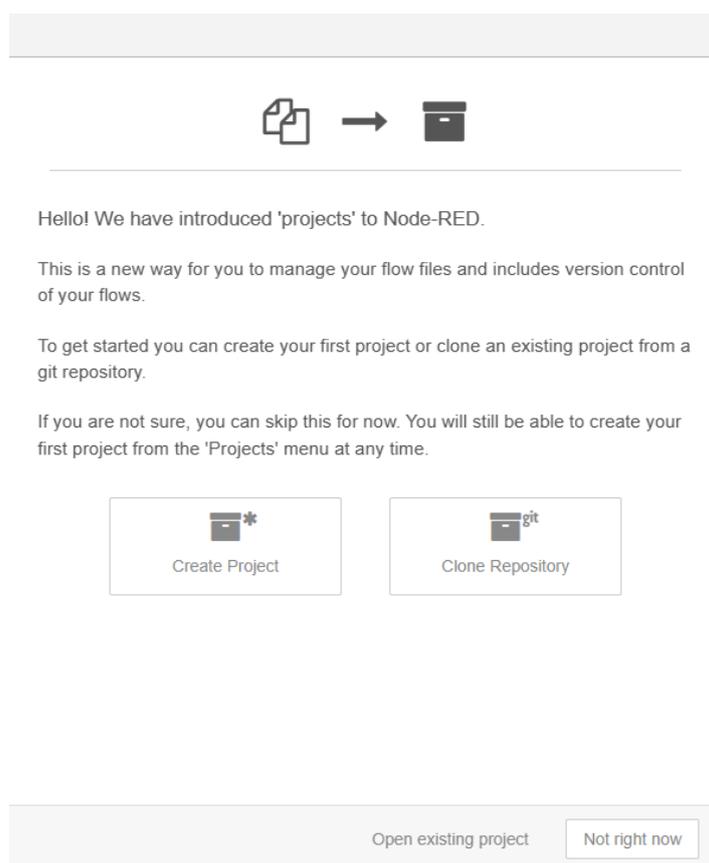


Ilustración 22: Ventana emergente de "projects" en Node-RED

Esto reconducirá al menú de flujos, en el que se debe hacer clic arriba a la derecha en el símbolo de *+* para crear un nuevo flujo y donde se le asignará un nombre, en este caso *Ejemplo Tutorial*, haciendo clic en la pestaña que aparece:

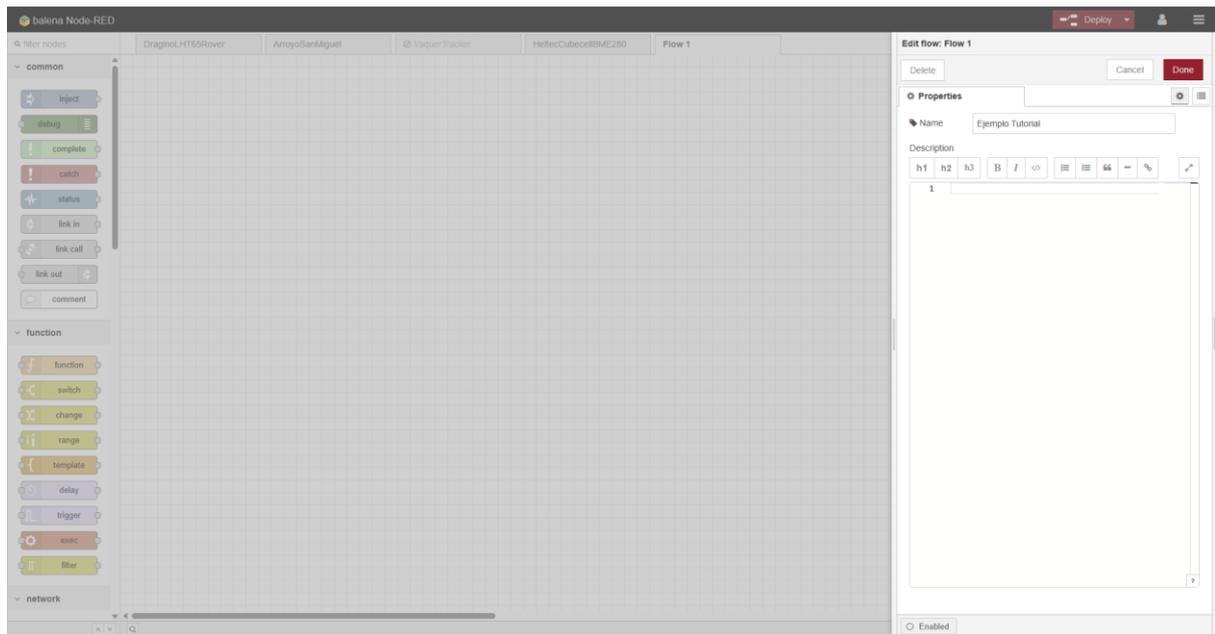


Ilustración 23: Creación y asignación de nombre a un nuevo flujo

Es ahora cuando, desde el menú vertical izquierdo, se deben de arrastrar y colocar, de izquierda a derecha, los siguientes nodos:

- *mqtt in*
- *comment*
- *function*
- *debug* (dos nodos)
- *influxdb out*

Se disponen de la siguiente manera:

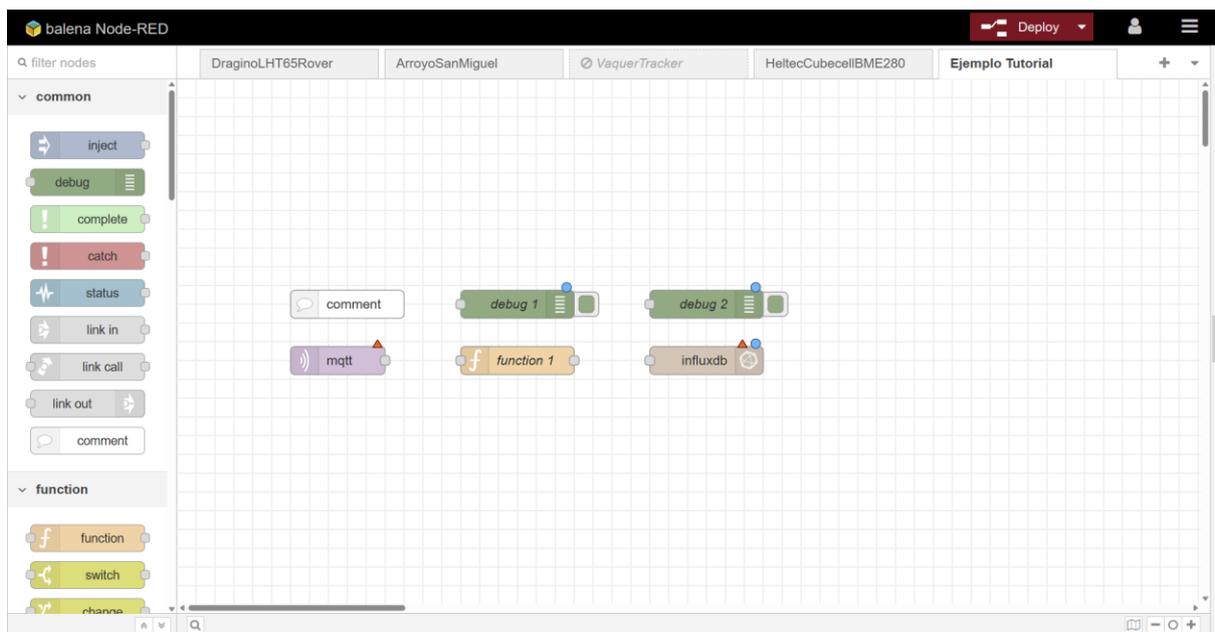


Ilustración 24: Nodos de Node-RED dispuestos pero sin conectarse aún

CONFIGURACIÓN DEL NODO MQTT IN

El primer nodo que se debe configurar es *mqtt in*. Por ello se hace doble clic en él y, siguiendo las siguientes ilustraciones, se rellenan los campos:

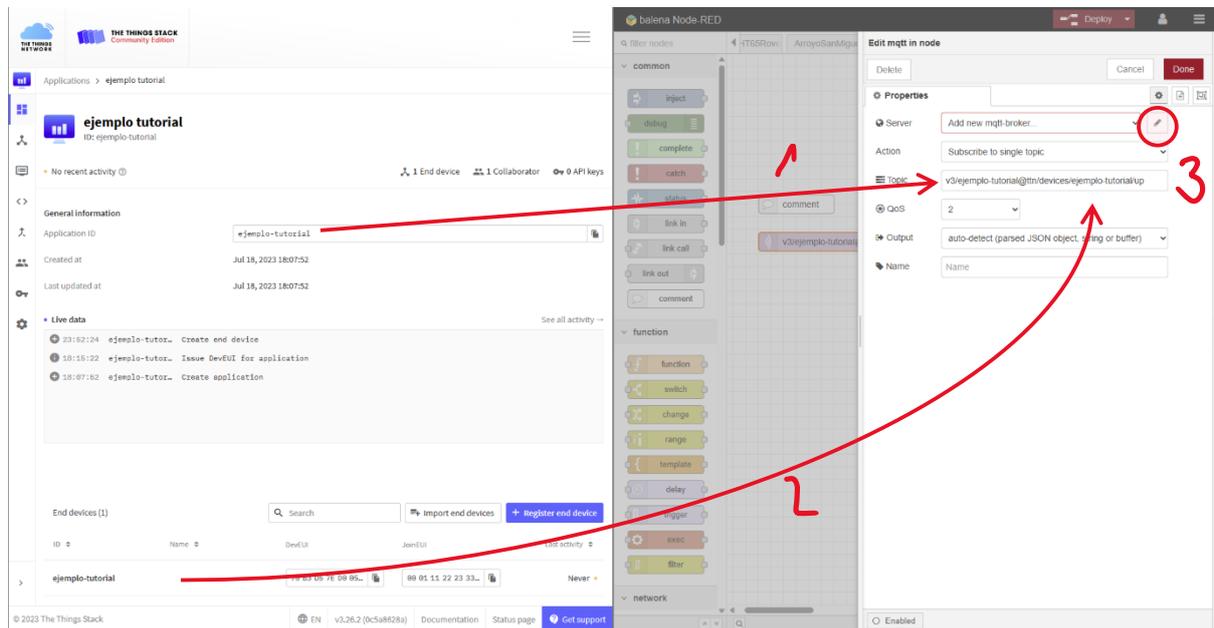


Ilustración 25: Ventana emergente principal del nodo *mqtt in* en la que se debe rellenar el campo de "Topic" y, después, hacer clic en el botón del lápiz

Como aclaración, *Topic* se rellena de la siguiente forma: *v3/"Application ID"@ttn/devices/"End device ID"/up*.

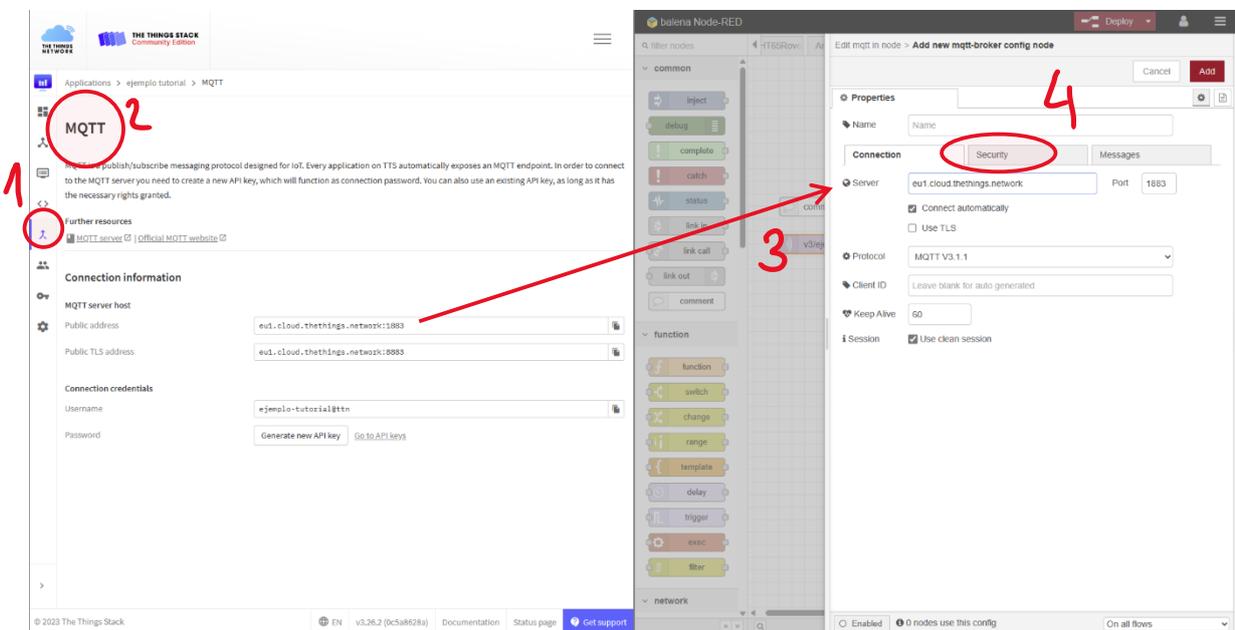


Ilustración 26: Ventana "Connection" en la que se debe introducir el campo "Server" desde la integración MQTT de TTN y, después, acceder a "Security"

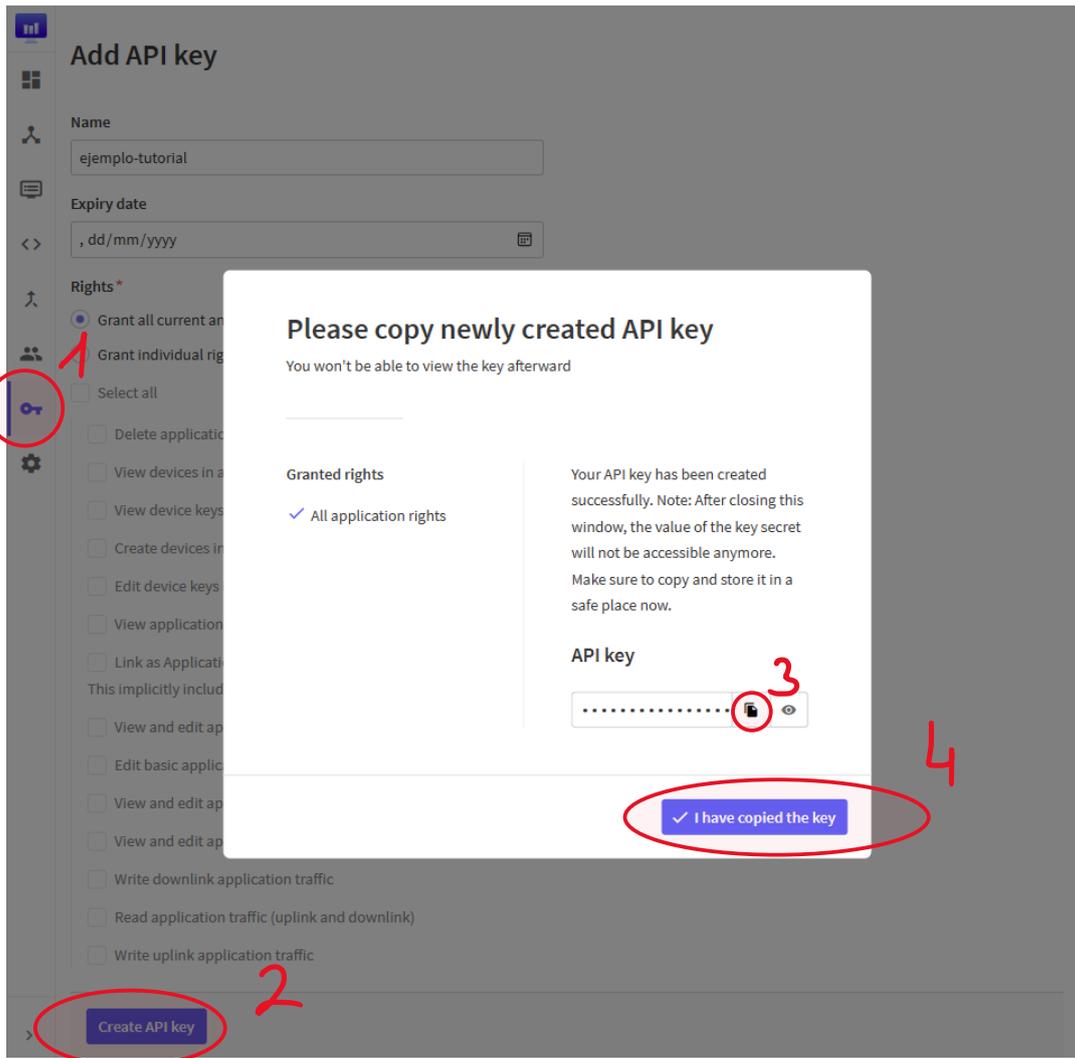


Ilustración 27: Menú de "API Key" de TTN en la que se tendrá que generar una nueva key y copiarla al protapapeles ya que será la única vez que esté disponible su acceso

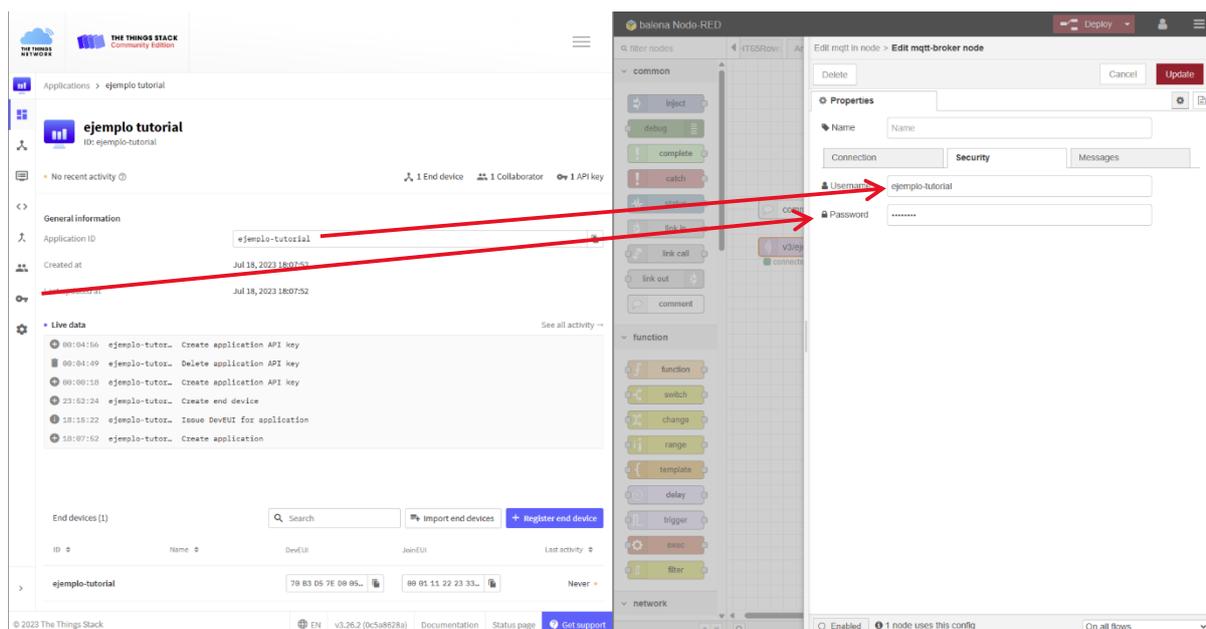


Ilustración 28: Menú “Security” en el que se debe introducir la “Application ID” de TTN como “Username” y la “API Key” del portapapeles en “Password”

Una vez se tienen todos los campos rellenos, se hace clic en el botón rojo de *Update* arriba a la derecha, después al de *Done* y, finalmente, a *Deploy*. Si todo se ha hecho correctamente, debería de aparecer un cuadrado verde y la palabra *connected* debajo del nodo *mqtt in*:



Ilustración 29: Nodo *mqtt in* conectado correctamente a la integración MQTT de TTN

Como añadido a la configuración de este nodo, se recomienda hacer doble clic en el nodo *comment*, cambiar su nombre a *API Key Ejemplo Tutorial* y pegar la *key* generada para no perderla.

Para comprobar que la información se esté recibiendo de forma correcta, se conecta el nodo *mqtt in* al primer nodo *debug* y, haciendo clic en las opciones de *debug* situadas en la parte superior derecha de la interfaz de Node-RED para seleccionar que solo se imprima la información de dicho nodo, se observa cómo va llegando la información:



Ilustración 30: Información recibida por MQTT en Node-RED y mostrada por el primer nodo "debug"

CONFIGURACIÓN DEL NODO FUNCTION

Observándose la información recibida por el nodo anterior, es importante enfatizar un problema: la Raspberry Pi 3 cuenta con una memoria que, aunque generosa para los datos que se van a almacenar, no es infinita, por tanto, se deben de guardar únicamente aquellos valores provenientes de las variables que sean de interés. Para ello, observando lo que se imprime en el *debug*, se debe seguir la siguiente ruta para identificar las variables relacionadas con las mediciones del sensor BME280 conectado a la Cubecell: *object* -> *uplink_message* -> *decoded_payload*. Prestándose atención, aquí se ubican las variables *battery*, *humidity*, *pressure* y *temperature* que se envían desde TTN. Para quedarse únicamente con ellas, se debe escribir el siguiente código en el bloque de función:

```

var bateria = msg.payload.uplink_message.decoded_payload.battery;
var humedad = msg.payload.uplink_message.decoded_payload.humidity;
var presion = msg.payload.uplink_message.decoded_payload.pressure;
var temperatura = msg.payload.uplink_message.decoded_payload.temperature;

msg.payload = { battery: bateria, humidity: humedad, pressure: presion,
temperature: temperatura };

return msg;

```

Figura 1: Código del nodo "function" para el filtrado de las variables del sensor

Conectándose ahora también el nodo *mqtt in* al nodo de función y éste último al segundo nodo *debug*, se observa lo siguiente:

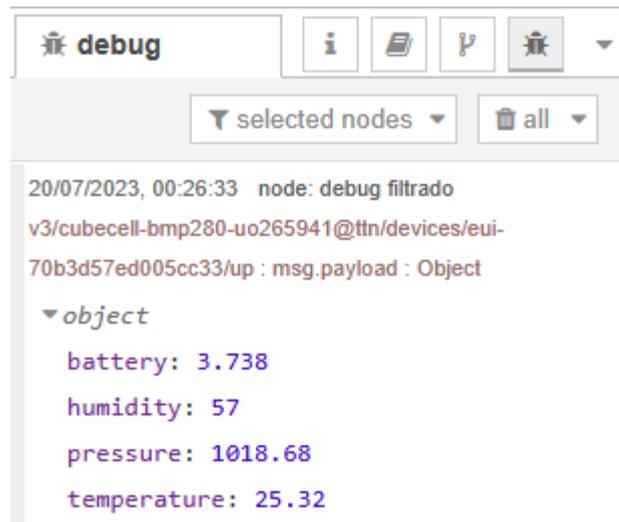


Ilustración 31: Variables filtradas tras el nodo "function"

Gracias al nodo *function* se ha desechado lo innecesario para guardar únicamente lo relacionado con las variables del sensor.

CONFIGURACION DEL NODO INFLUXDB OUT

Finalmente, el último nodo a configurar es *influxdb out*. Este nodo se encarga de establecer conexión con el *container* de InfluxDB y almacenar la información de las variables filtradas en la base de datos.

Lo primero es acceder al servicio de Grafana, ya que es desde aquí donde se crean y configuran también las bases de datos. Su URL es la siguiente: <http://4f566df1fed52c6e7fd5f661f64ae3eb.balena-devices.com:8080>.

Una vez se introducen las credenciales (Usuario: *admin*, Contraseña: *m3di4l4b*), aparece el menú principal de Grafana. Aquí se debe acceder al menú *Configuration* -> *Data Sources*:

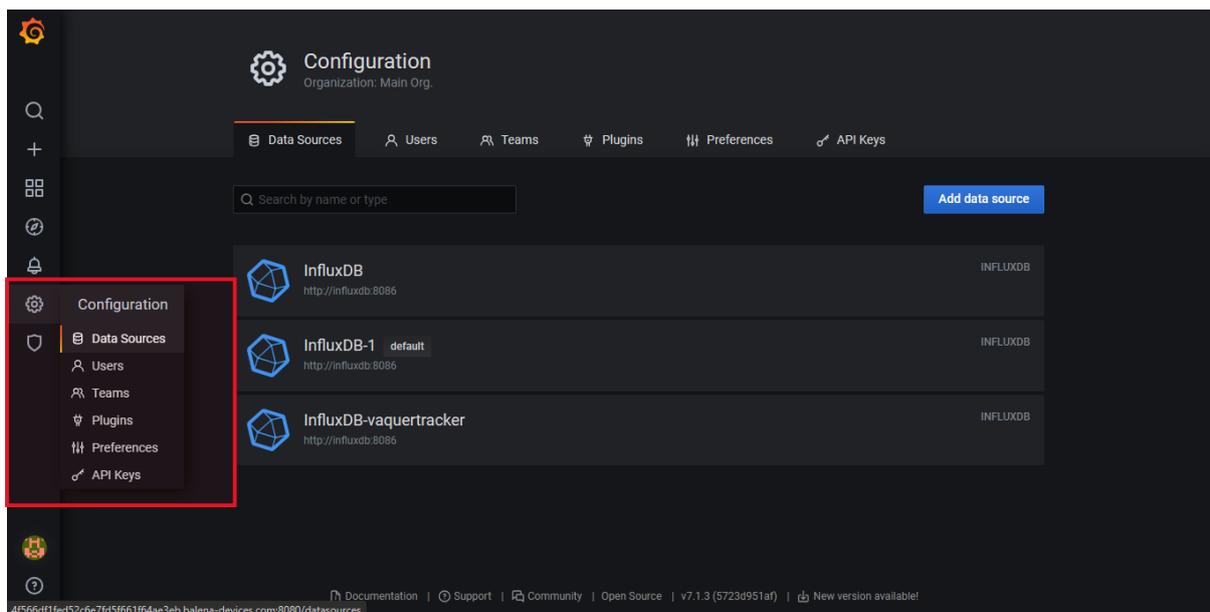


Ilustración 32: Menú "Data Sources" de Grafana

Ahora, se debe hacer clic en el botón *Add data source* y configurar una nueva base de datos de InfluxDB y establecerla como predeterminada o *Default*:

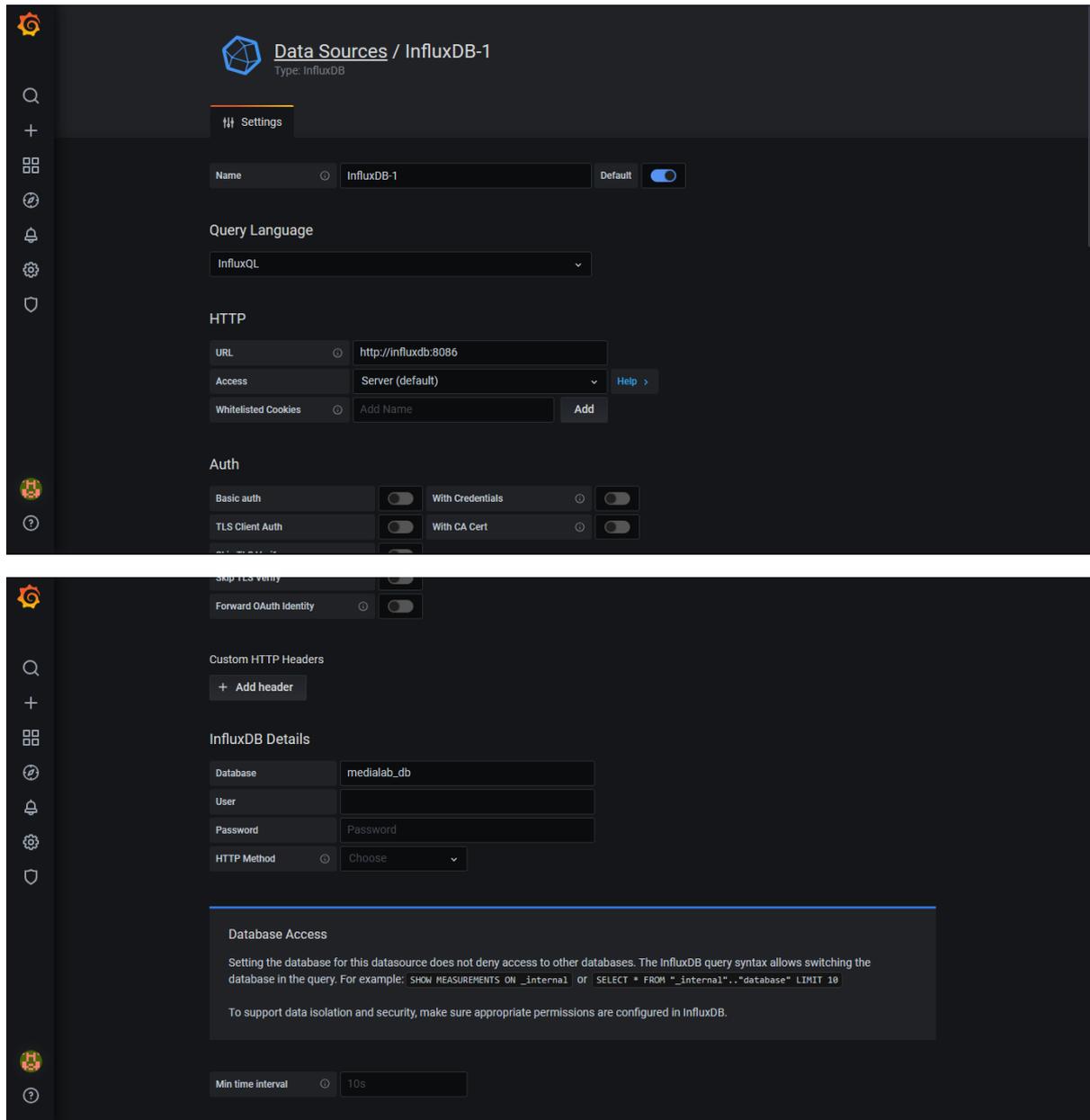


Ilustración 33: Configuración para una nueva base de datos de nombre "medialab_db" que funcionará como predeterminada

Habiéndose creado la base de datos, se vuelve a Node-RED para la configuración del nodo. Haciendo doble clic sobre él, se rellenan los campos de la ventana emergente principal:

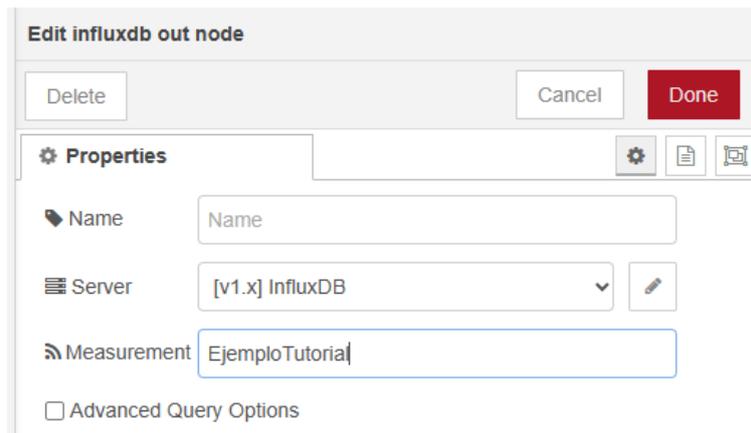


Ilustración 34: Ventana emergente principal del nodo "influxdb out"

Measurement representa la etiqueta bajo la que se recogerán los datos de este nodo en la base de datos *medialab_db*. A continuación, se hace clic en el botón del lápiz del apartado *Server* en el que se rellenan los siguientes campos:

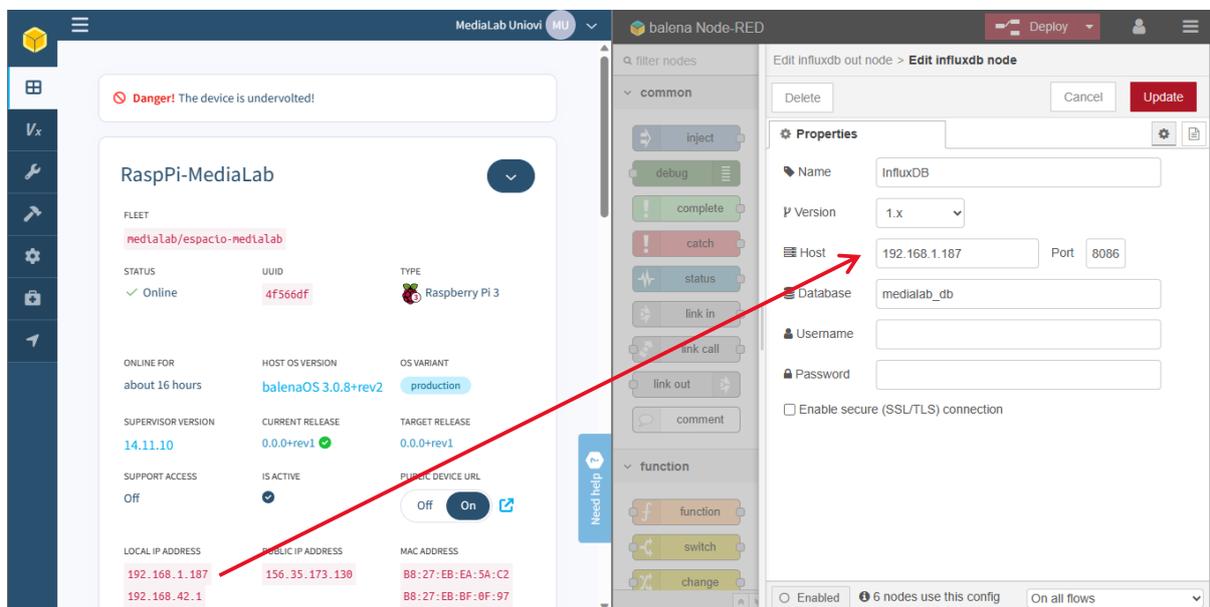


Ilustración 35: Campos del apartado "Server" del nodo "influxdb out"

Una vez se ha completado este paso, se hace clic en *Update* y *Done*. Se comprueba que se han unido todos los nodos y, opcionalmente, se les asigna nombres para facilitar su comprensión. Se recuerda aplicar los cambios haciendo clic en *Deploy*:

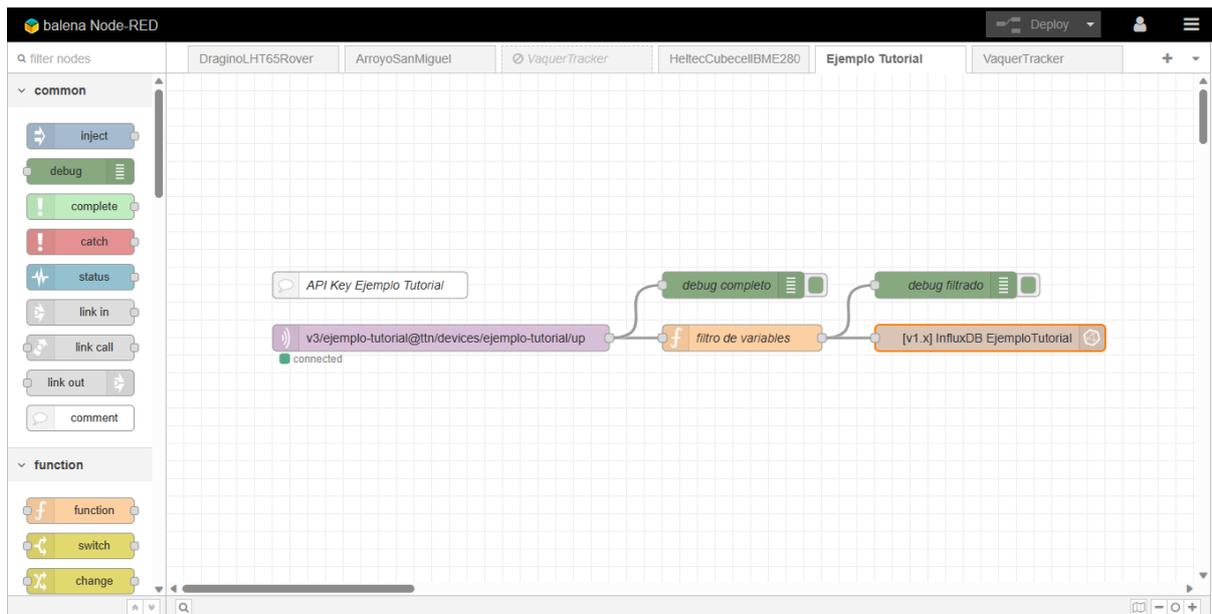


Ilustración 36: Flujo de Node-RED para el ejemplo del tutorial completamente configurado

CONFIGURACIÓN DEL CONTAINER GRAFANA

En este último servicio del *stack* de MING se explicará cómo crear el elemento final sobre el que el usuario pueda ver de forma gráfica y actualizado en tiempo real los datos de las variables recogidas por su sensor LoRa. Para ello, se accede de nuevo a Grafana, esta vez al apartado del menú lateral *Dashboards -> Manage*:

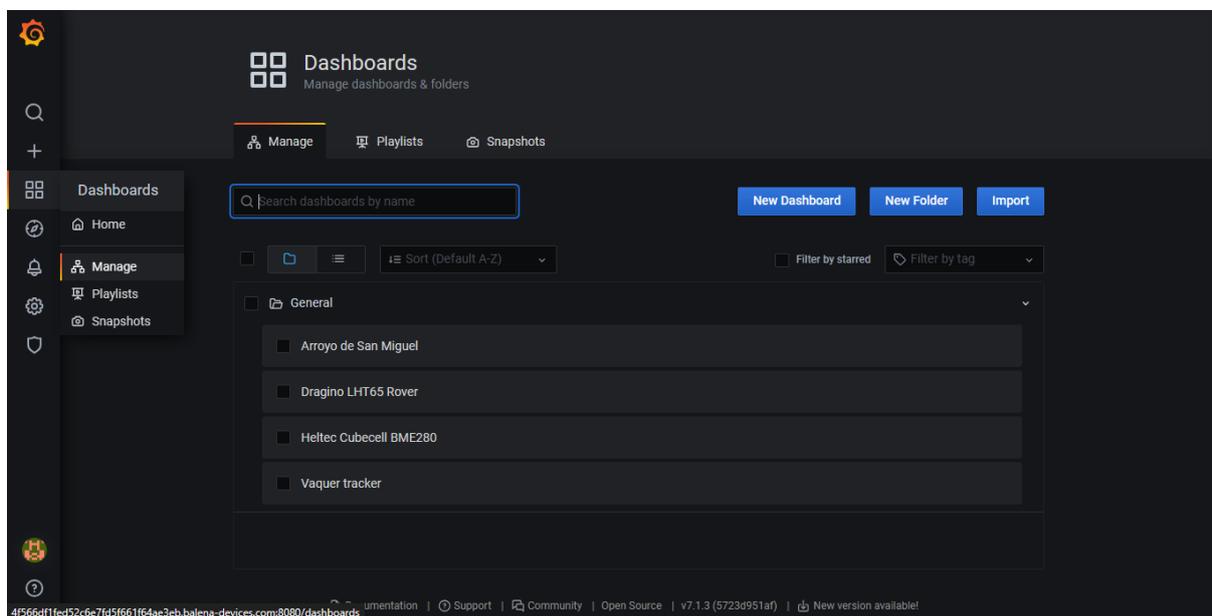


Ilustración 37: Menú "Dashboards -> Manage" de Grafana

Haciendo clic en el botón *New Dashboard*, se lanzará el editor de *dashboards* y, sobre el botón de *Add new panel* o el del diagrama de barras con un signo de suma naranja, se creará el primer gráfico:

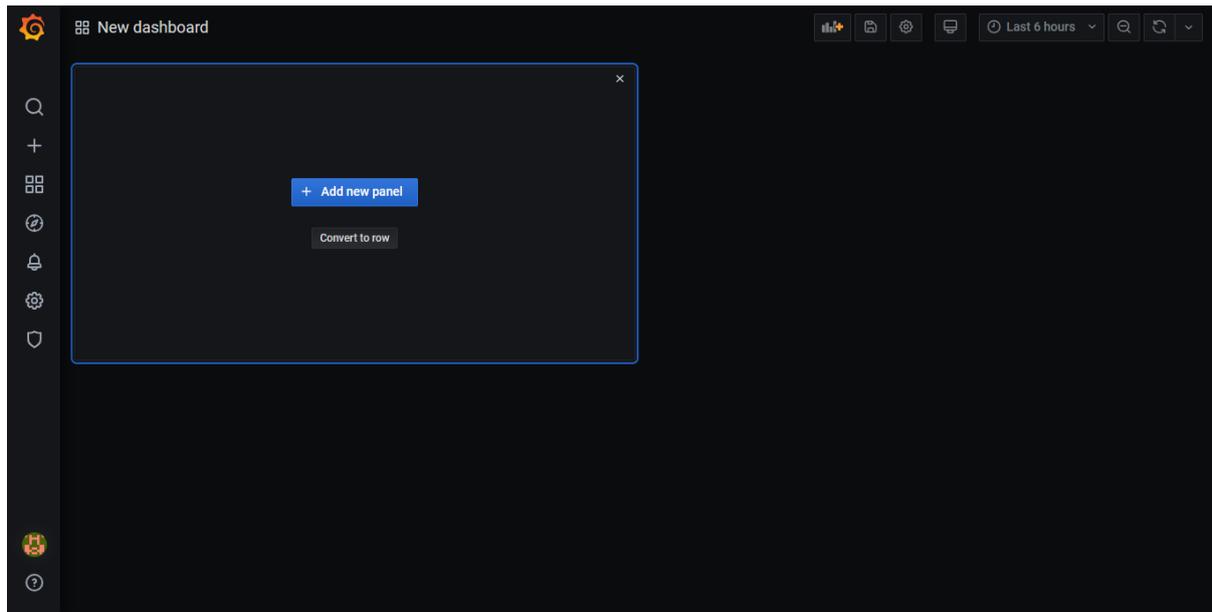


Ilustración 38: Nuevo "dashboard"

A continuación, se deben rellenar los campos del apartado *Query* según la siguiente captura que, para el ejemplo, se cogió el *field* de *battery*, aunque se recuerda que se dispone de otros (*temperature*, *pressure* o *humidity*):

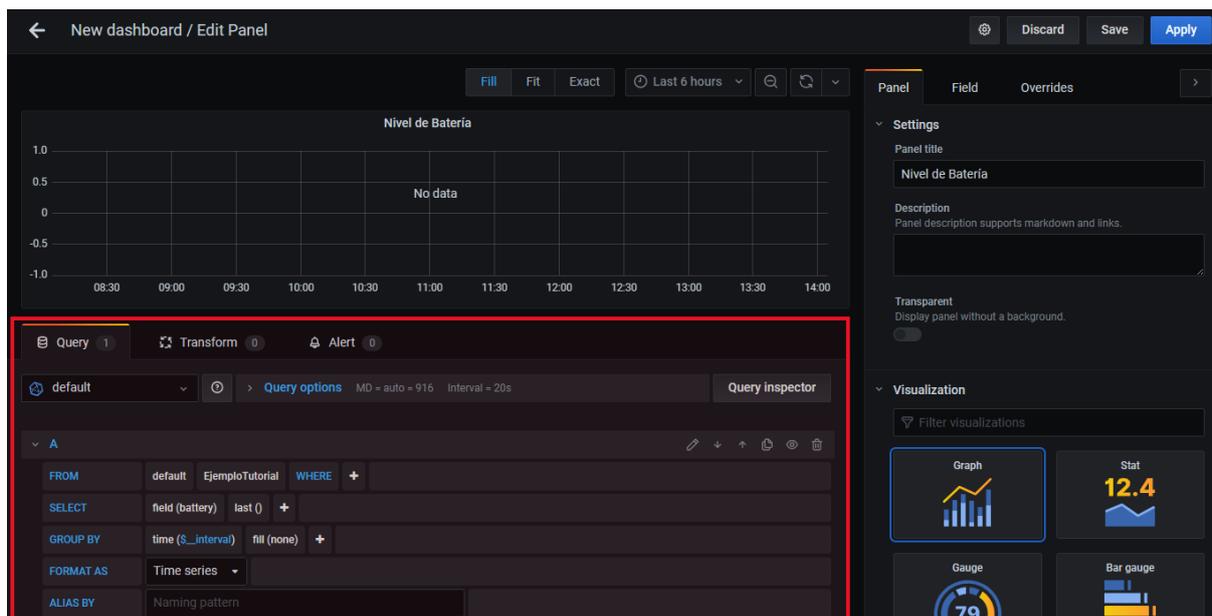


Ilustración 39: Campos de "Query" rellenados para graficar la variable "battery"

Lo siguiente, y a libre gusto del usuario, es configurar los campos de *Panel*, *Field* y *Overrides* de la parte superior derecha de la interfaz. Una vez se complete la configuración, se debe hacer clic en *Save* y *Apply* para ver cómo se visualiza el gráfico de la variable. En este momento, ya se debería ver cómo se van graficando los datos:



Ilustración 40: “Dashboard” de Grafana completamente configurado

Una vez se haya finalizado la configuración y se desee guardar los cambios, se hace clic en el botón con el icono del disquete y aparecerá una ventana emergente en la que se debe dar un nombre al *dashboard* y, volviendo al menú *Dashboards* -> *Manage*, aparecerá entre los que se vayan creando para cada uno de los nodos:

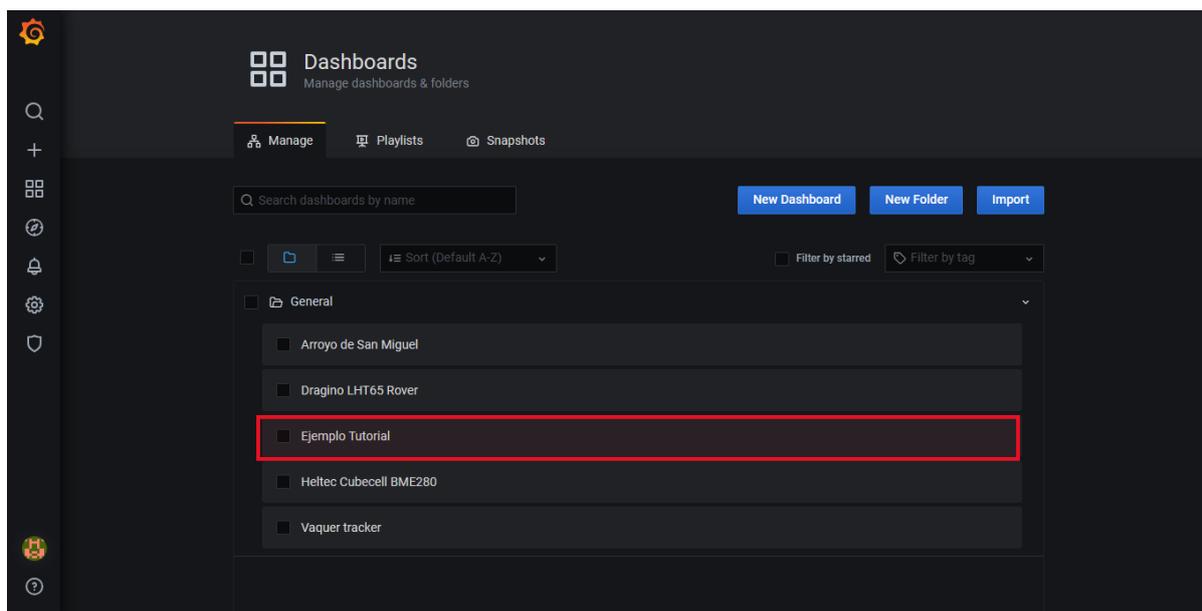


Ilustración 41: “Dashboard” Ejemplo Tutorial guardado y añadido a la lista

En caso de querer editarse cualquier elemento, tan solo es necesario hacer clic sobre él y modificarlo a gusto.

CUARTO PASO: COMPARTIR PÚBLICAMENTE UN DASHBOARD

Como último paso de este tutorial, es interesante resaltar el interés de crear una dirección URL que, con tan solo un clic, una persona completamente ajena a los conocimientos dados en el tutorial pueda acceder a un *dashboard*. Para ello, se debe acceder al *dashboard* de interés y ajustar para qué franja

temporal se quiere compartir. El ajuste se encuentra en la parte superior derecha de la interfaz. Para este caso, se compartirá para los últimos 30 minutos:

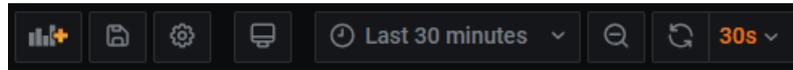


Ilustración 42: Franja temporal configurada para mostrar los últimos 30 minutos

Una vez se ha hecho esto, tan solo se debe copiar la URL que aparece en el navegador. Como ejemplo final, el de este caso resulta ser el siguiente: [Ejemplo Tutorial - Grafana \(4f566df1fed52c6e7fd5f661f64ae3eb.balena-devices.com\)](https://ejemplo-tutorial-grafana-4f566df1fed52c6e7fd5f661f64ae3eb.balena-devices.com). Adicionalmente, en el documento *Listado de nodos LoRa para la adquisición de datos en MediaLab Universidad de Oviedo*, se hace un repaso a todos los nodos que han sido desarrollados en la cátedra y se adjuntan los links correspondientes a sus paneles.

Sin embargo, algunos navegadores ofrecen una opción también muy interesante: generar un código QR escaneable desde la cámara de un teléfono móvil:

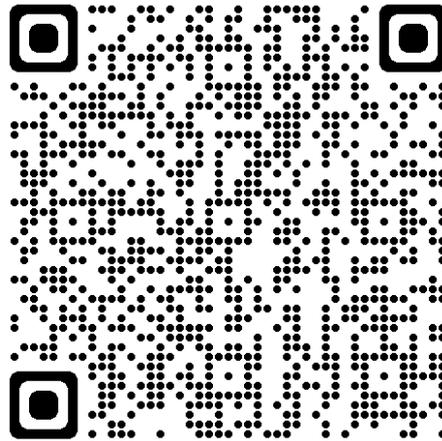


Ilustración 43: Código "QR" para acceder al "dashboard" Ejemplo Tutorial