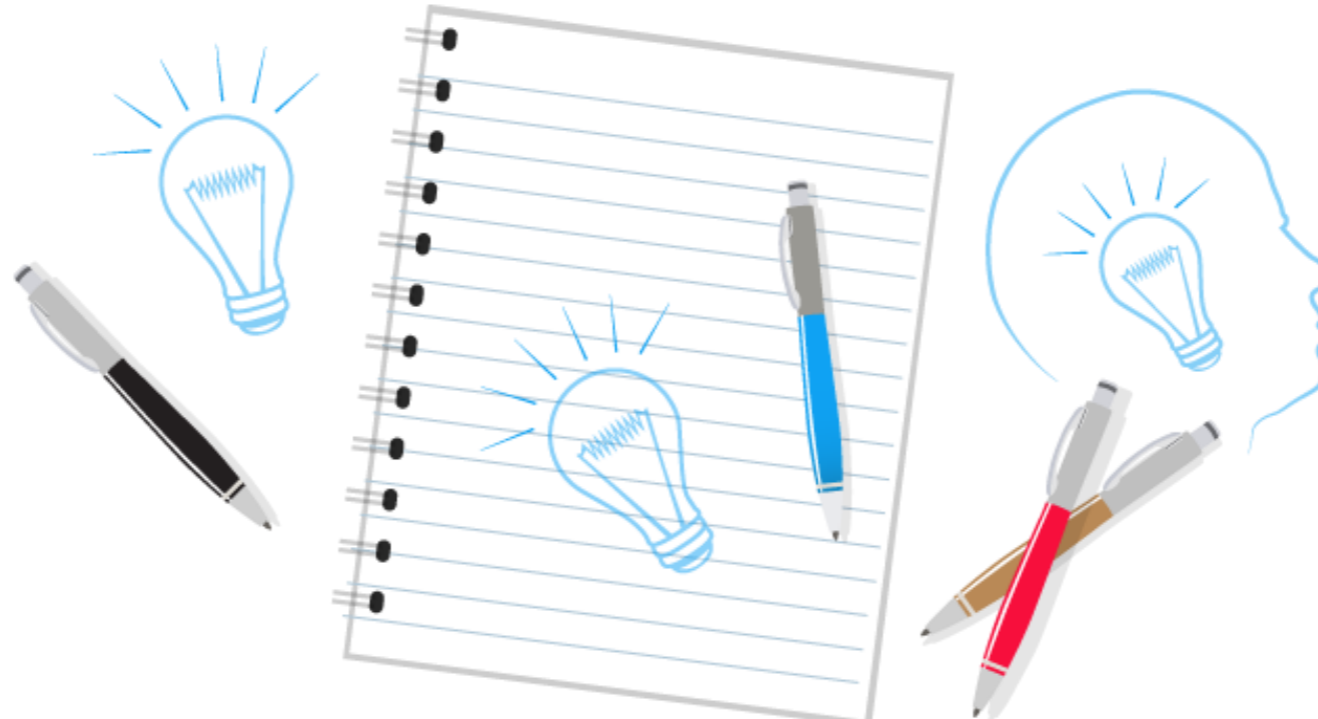


CAPÍTULO 1. Internet of Things

v.1.1 MARZO 2024

Ricardo Moraleda Gareta

[Director departamento de software de GDO Software]



Internet of Things

v.1.1 MARZO 2024



IOT

SIEMENS



Siemens IOT 2020

Node.JS

Node-Red

Flows

Dash boards

JSON

Modbus TCP

SCADA



MQTT

Ubidots



Dispositivo IOT



SIMATIC IOT 2020 / 2040 **SIEMENS**

Gateway Ethernet fabricado por Siemens y basado en Arduino.
Sistema Operativo: Yocto Linux 2.1

5 ED de 24 V y 2 SD, 2 EA entre 0 y 10 V o entre 0 y 20 mA

Programable con lenguajes de alto nivel (Java, C++, Python,
JavaScript: Node.js / Node-RED)

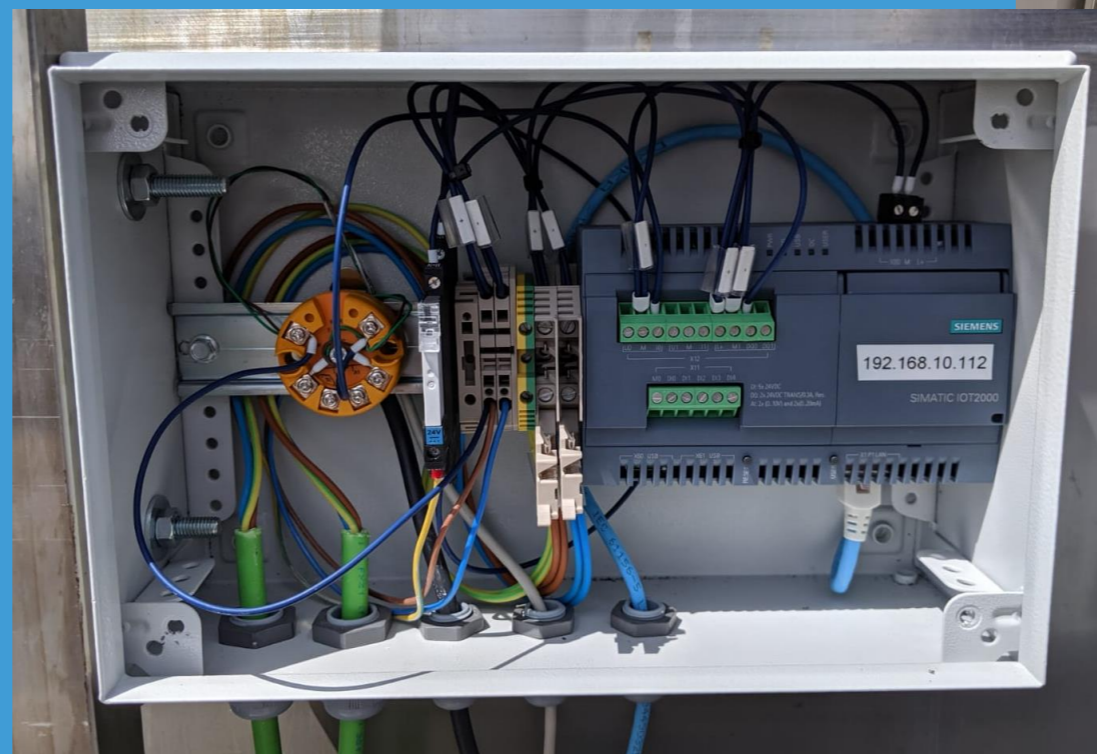
Para profundizar cómo instalar y configurar la SD y el software, seguid el siguiente enlace (ENERO 2018):
<https://drive.google.com/file/d/18NCgvOZVMuQlij5iruu0gIL-xa4y5eON/view>

IOT 2020 en instalación real

IOT por cada depósito de vino controlando:

- Sonda T^o PT100 (4-20 mA)
- Relé válvula frío

Instalaciones de 24 y 64 IOTs en red Ethernet





Módulos para PC



Node.js



Entorno de ejecución para JavaScript

Paso 1. Instalar Node.js

Lo primero que vamos a hacer, es instalar Node.js. Para ello, descargaremos desde su página web <https://nodejs.org/en/> la version para Windows. En este ejemplo, descargaremos e instalaremos la version LTS 10.15.2

Comprobaremos que está bien instalado usando el Command Prompt: `node --version ; npm --version`

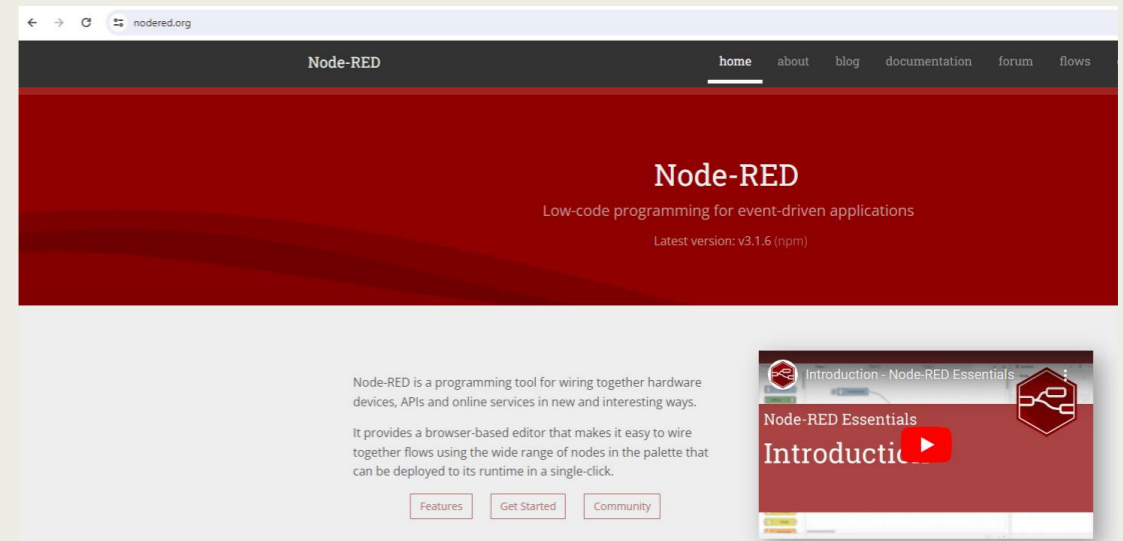
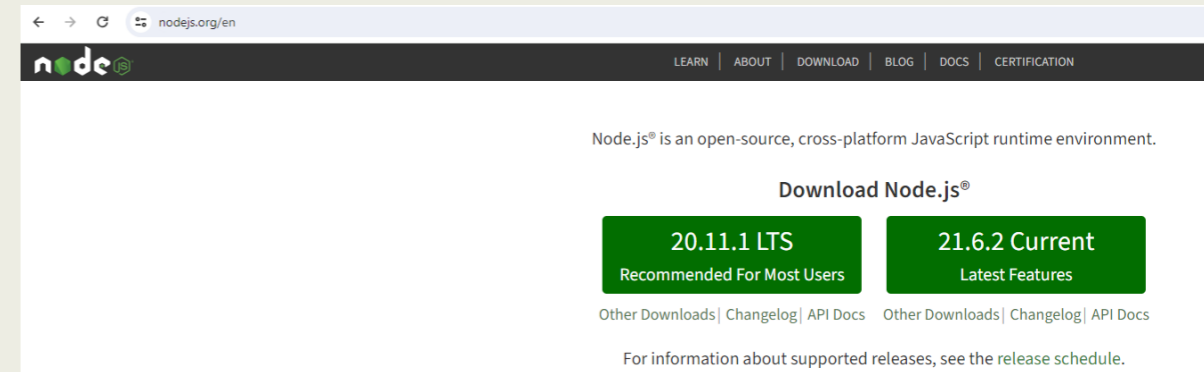
Node-Red



Paso 2. Instalar Node-Red

Una vez instalado node.js pasaremos a instalar Node-Red

Para ello, nuevamente desde el Command Prompt ejecutaremos: `npm install -g --unsafe-perm node-red`





Node-Red



Node-Red

Herramienta de desarrollo basada en flujo para la programación visual desarrollada originalmente por IBM para conectar dispositivos de hardware, API y servicios en línea como parte de Internet of Things.

<http://localhost:1880>

The screenshot displays the Node-RED web interface in a browser window. The address bar shows the URL `localhost:1880/#flow/e8c35ae7.b7d968`. The interface includes a left sidebar with a 'filter nodes' search bar and a list of nodes categorized into 'input' and 'output'. The main workspace shows a flow named 'Flow 1' with several nodes: an inject node with the text 'Bienvenida:Hola mundo!', two function nodes labeled 'Sumando A' and 'Sumando B', a function node labeled 'Resultado', and a message node. Below this, there is a more complex flow involving an 'increment' node, a 'start' node, and a 'msg payload' node. A terminal window at the bottom shows the Node-RED startup logs, including the version information and the server URL `http://127.0.0.1:1880/`.

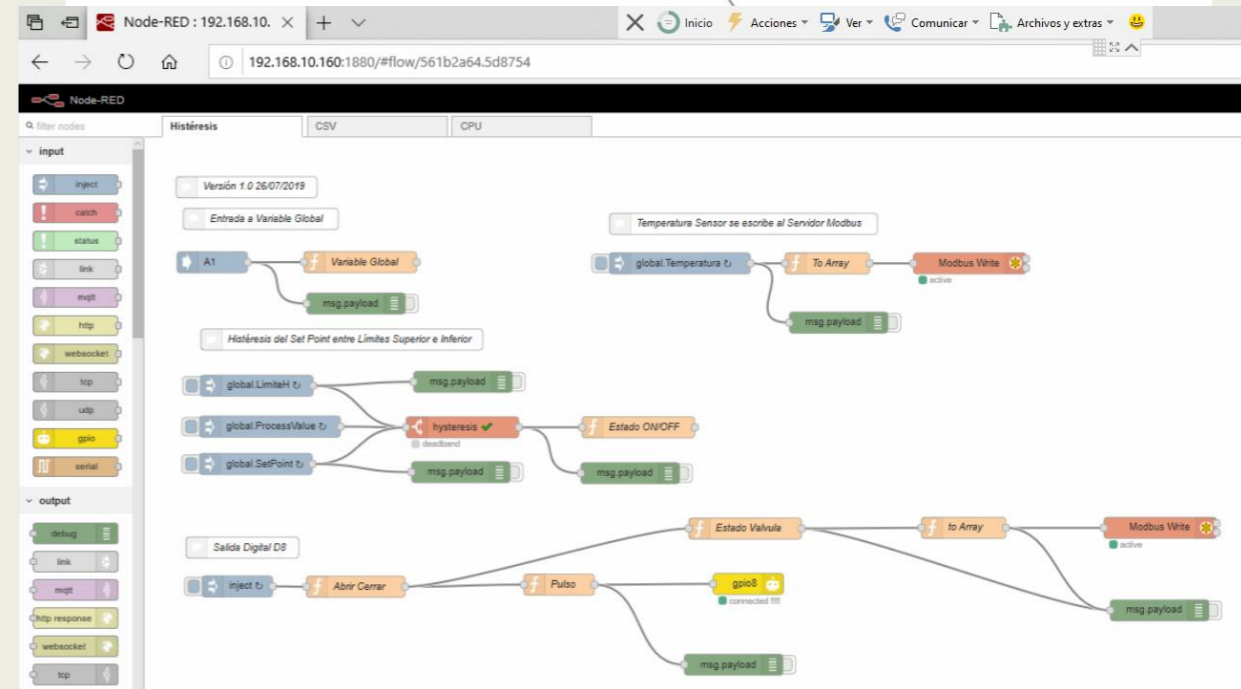
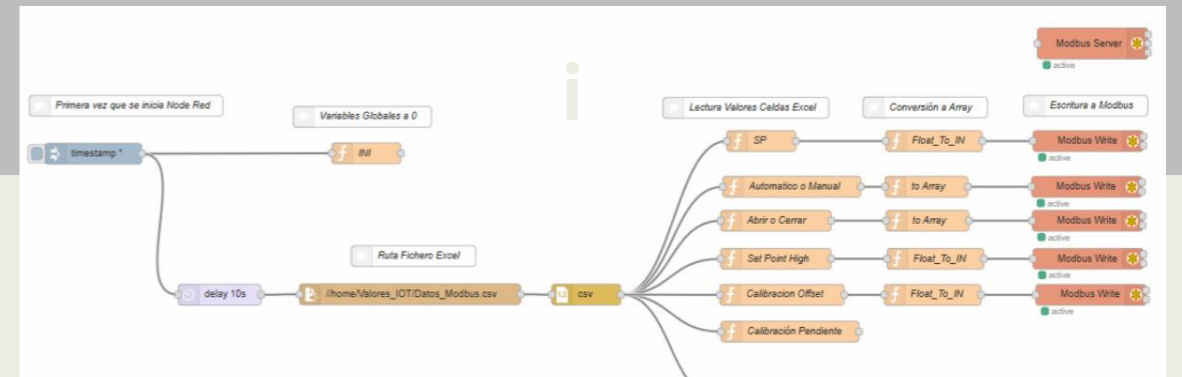
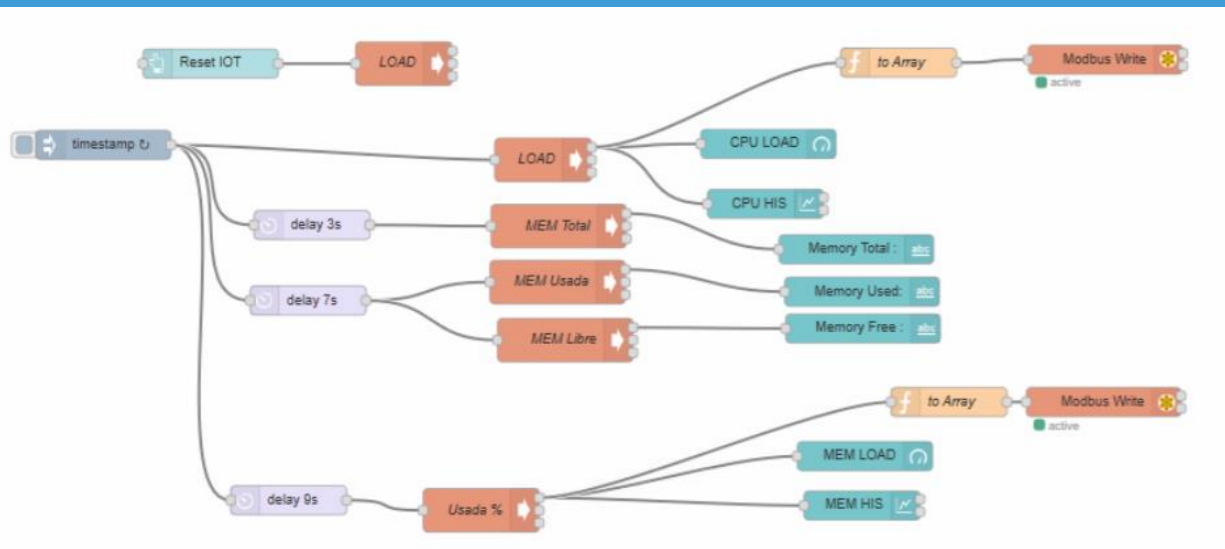


Node-Red (flujos)



Ejemplos de flujos

- Control de proceso
- Escritura en fichero local CSV
- Publicación en dashboard





Node-Red (Dashboard)



<http://localhost:1880/ui>

Estado del Tanque **Automático**

Consigna en Manual **Abrir**

Temperatura Tanque



SetPoint SP **SP 17.50**

Límite Superior **SPH 1.0**

Valor Actual Calibración Pendiente **0.98**

Valor Actual Calibración Offset **-1.7**

Histórico Tanque



Valor Calibración Pendiente

CPU_IOT

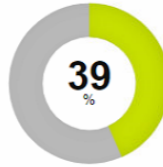
RESET IOT

Memory Total : 491.376 MB

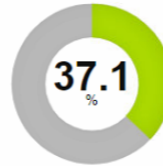
Memory Used: 182.504 MB

Memory Free : 309 MB

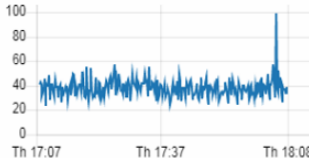
CPU LOAD




Memoria Usada



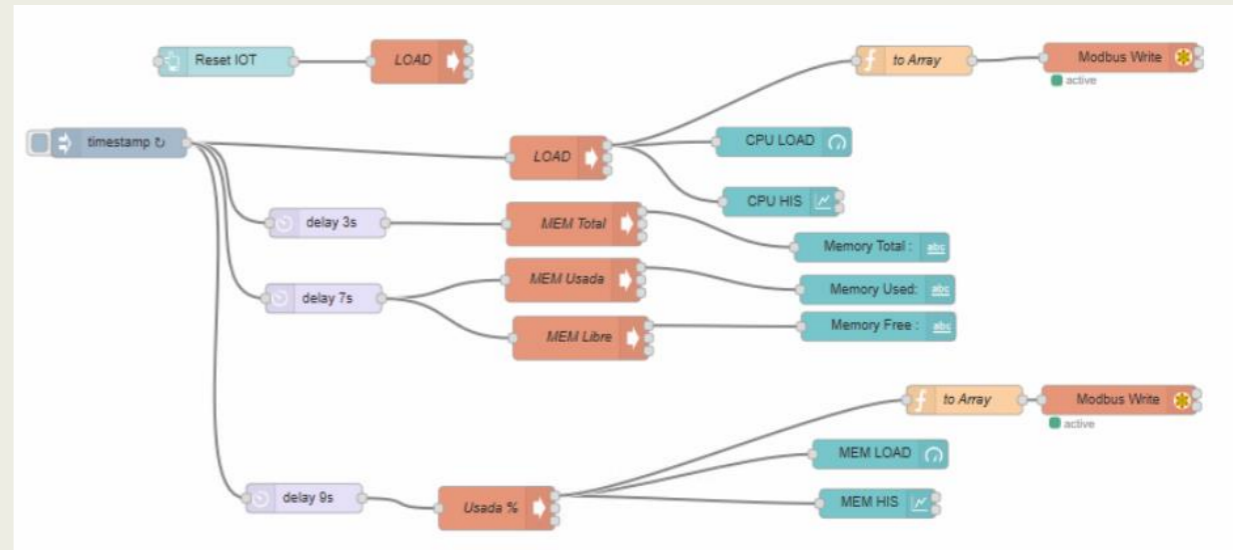
CPU HIS



MEM HIS



Flujo





Node-Red (JSON)

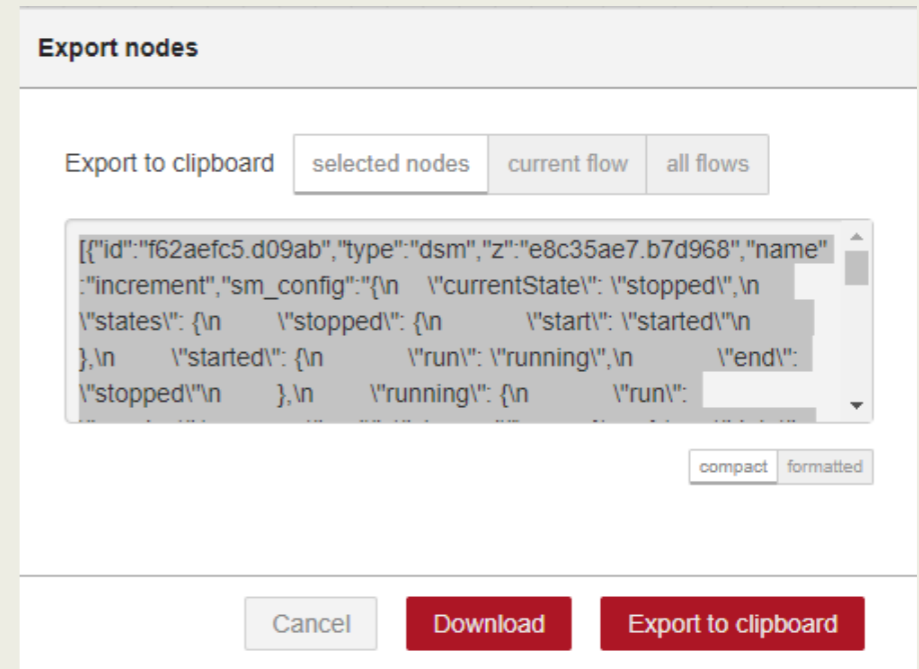


Flujos en formato JSON



Acrónimo de JavaScript Object Notation, «notación de objeto de JavaScript») es un formato de texto sencillo para el intercambio de datos.

Exportación de los flujos



- Objetos: Son colecciones no ordenadas de pares de la forma <nombre>:<valor> separados por comas y puestas entre llaves. El nombre tiene que ser una cadena y entre ellas. El valor puede ser de cualquier tipo. Ejemplo:

```
{"departamento":8,"nombredpto":"Ventas","director":"juan rodriguez","empleados":[{"nombre":"Pedro","apellido":"Fernandez"}, {"nombre":"Jacinto","apellido":"Benavente"}]}
```



Node-Red (Modbus TCP)

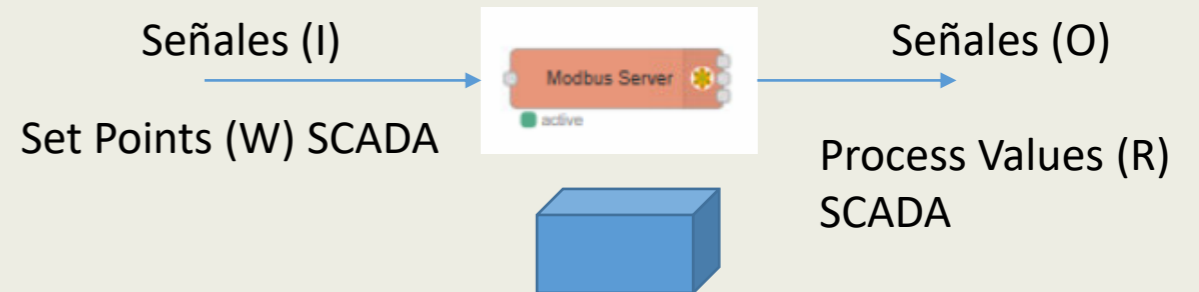
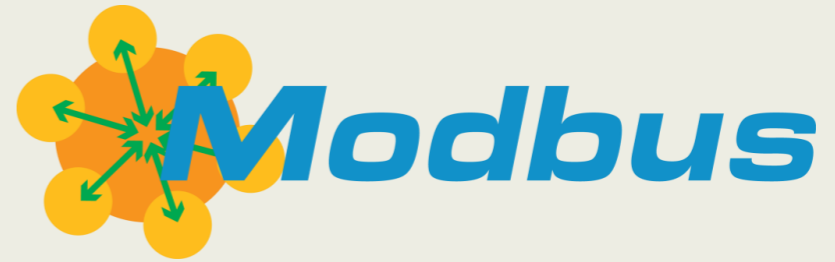


Intercambio de datos por Modbus TCP

Control desde SCADA de múltiples IOTs distribuidos.

SCADA programado con Visual C# .NET.

Nodo Modbus Server





SCADA (Modbus TCP)



Control

Comunicación OK
Habilitado

ID: 18
Descripción: 253
Zona: FILA2
IP: 192.168.10.110

Manual (0) / Auto (1)

Controlador Temperatura
PV: 18.1
Set Point: 17.5

Limite Superior SP (°C): 1
Calib. sonda (°C): -1.7

D. Inicial: 1081 g/L D. Actual: 1075 g/L
CHARD ECO CP
CHCP190819

DEN_MAX	DEN_MIN	SP
1083	1059	17.5
1058	1016	14.5
1015	1005	16.5
1004	992	17.5

MENSAJE	DESCENSO_DENSIDAD
OXIGEN 5 mg	20
NUTRICIO + ...	22
CLARIFICACIÓ	30

FECHA DEN DIF TEMP

21/08/2019 9:17	1081		
22/08/2019 8:30	1079		14.0
23/08/2019 8:25	1077	2	13.8
24/08/2019 9:42	1075	2	15.6

Alarmas = 2

18/09/2019 23:46:12

FECHA_INICIO	DEPOSITO	TEMPERATURA	ESTADO_ALARMA	MENSAJE
18/09/2019 23:45:41	247	26.1	UNACK_ALM	TEMPERATURA FUERA DE MARGEN LO/HI
18/09/2019 18:14:39	248	20.4	UNACK_ALM	TEMPERATURA FUERA DE MARGEN LO/HI



MQTT



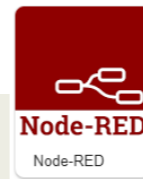
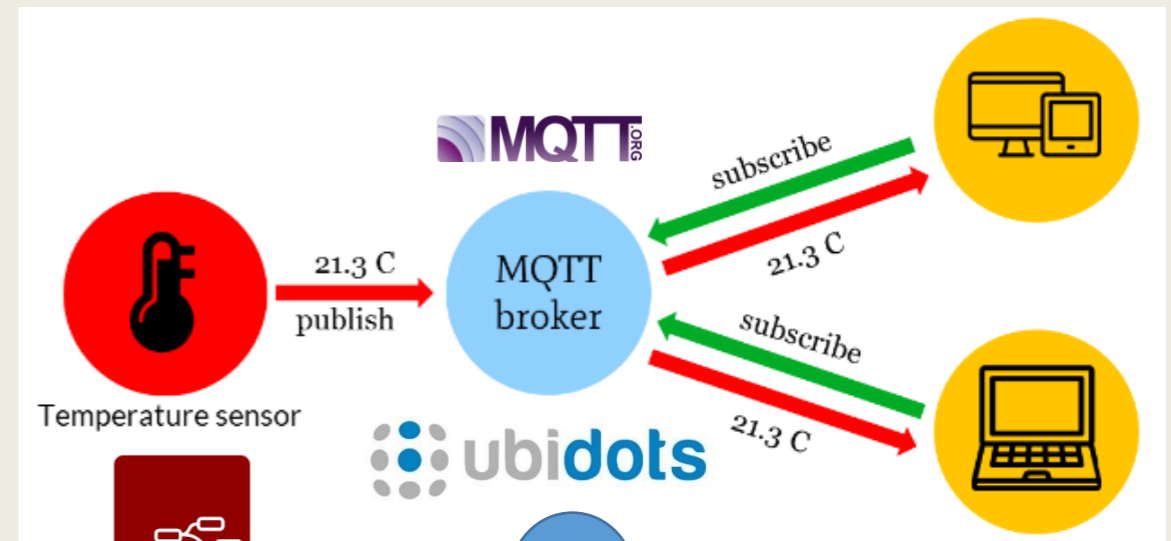
Message Queue Telemetry Transport)

Protocolo de mensajería basado en publicación-suscripción estándar ISO.

Funciona sobre TCP / IP.

Está diseñado para conexiones con ubicaciones remotas de "huella de código pequeña" o el ancho de banda de la red es limitado.

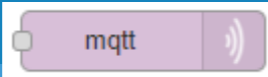
Broker. Recolector de datos del **publicador** y servidor a **suscriptores**



Publicación-Suscripción datos cloud



1 Publicar MQTT en NodeRed



Edit mqtt out node

Delete Cancel Done

Properties

Server: Ubidots

Topic: /v1.6/devices/plc_1200

QoS: [] Retain: []

Name: Name

Edit mqtt out node > Edit mqtt-broker node

Delete Cancel Update

Properties

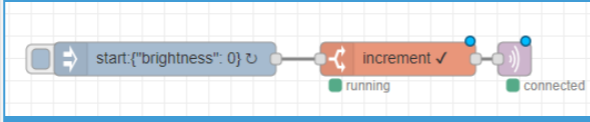
Name: Ubidots

Connection: Security Messages

Username: BBFF-cEuGo50WfUuNZYokdMGPKW1SnDH1jBb

Password: []

Cuenta creada en Ubidots (broker)



PUBLICACIÓN
Device: plc_1200
Variable: msg.payload.brightness

Suscribir MQTT en NodeRed



Edit mqtt in node

Delete Cancel Done

Properties

Server: Ubidots

Topic: /v1.6/devices/plc_1200/brightness

QoS: 2

Output: auto-detect (string or buffer)

Name: Name

SUSCRIPCIÓN
Device: plc_1200
Variable: brightness

```
graph LR; A["MQTT In (/v1.6/devices/plc_1200/brightness)"] --> B["msg.payload"]
```

debug

Time	Node ID	Running	msg.payload.brightness
28/8/2019 16:32:24	node: 301bc5df.14d6ea	running	30
28/8/2019 16:32:25	node: 301bc5df.14d6ea	running	40
28/8/2019 16:32:26	node: 301bc5df.14d6ea	running	50
28/8/2019 16:32:27	node: 301bc5df.14d6ea	running	60




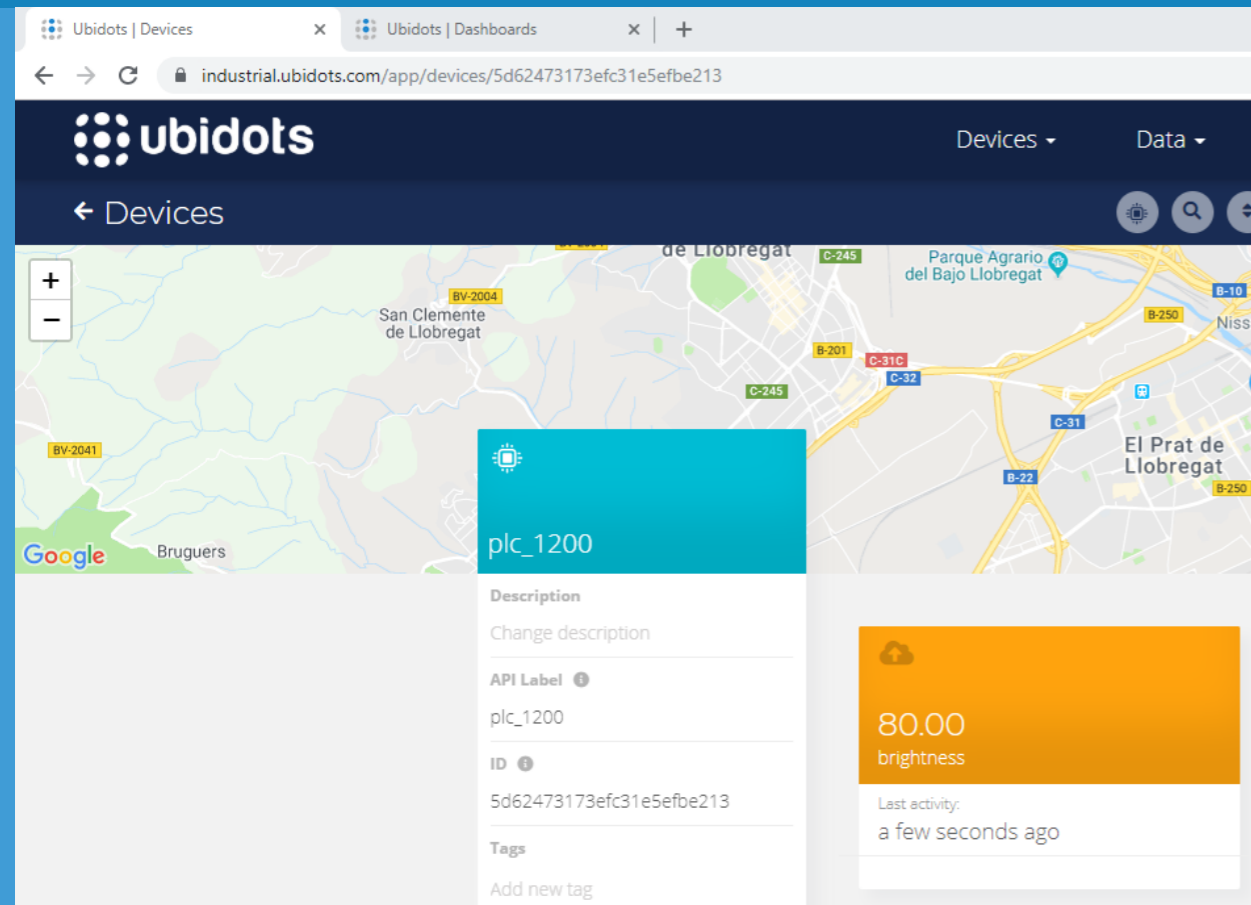
Broker datos cloud



2 Ubidots (MQTT broker)

Publicación de puntos (dots) en la nube desde Node-Red

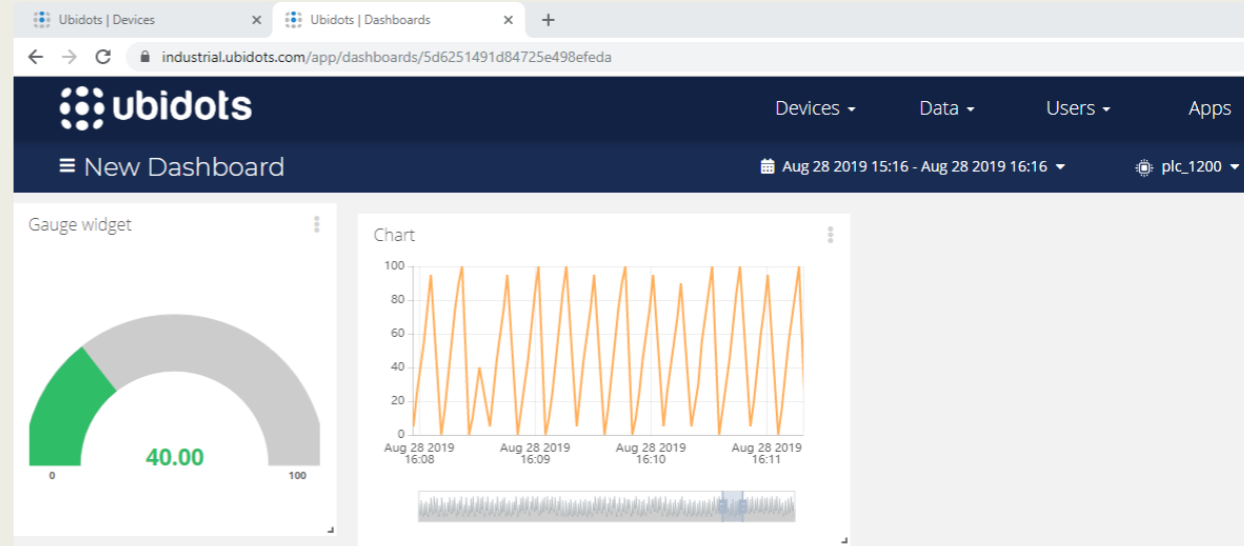
Utilizamos protocolo **MQTT** (Message Queue Telemetry Transport) 



The screenshot shows the Ubidots web interface for a device named 'plc_1200'. The browser address bar shows 'industrial.ubidots.com/app/devices/5d62473173efc31e5efbe213'. The device details include:

- Description:** Change description
- API Label:** plc_1200
- ID:** 5d62473173efc31e5efbe213
- Tags:** Add new tag

On the right, a gauge widget displays a value of 40.00, and a chart widget shows a line graph with a value of 80.00 labeled as 'brightness'. The last activity is noted as 'a few seconds ago'.



The screenshot shows a Ubidots dashboard for the device 'plc_1200'. The browser address bar shows 'industrial.ubidots.com/app/dashboards/5d6251491d84725e498efeda'. The dashboard features:

- Gauge widget:** A semi-circular gauge showing a value of 40.00.
- Chart widget:** A line chart showing a periodic signal over time, with a value of 80.00 indicated.

The chart x-axis shows timestamps from Aug 28 2019 16:08 to 16:11.

Internet of Things

v.1.1 MARZO 2024



<https://www.linkedin.com/in/ricardo-moraleda-gareta-9421099>

<https://www.linkedin.com/company/gdo-electric1996/>

RICARDO MORALEDA GARETA