

# Internet de las Cosas (IoT)

Lic. Ismael P. Rodríguez  
[ismael@lidi.info.unlp.edu.ar](mailto:ismael@lidi.info.unlp.edu.ar)



Cloud Computing. Cloud Robotics. 2025  
Facultad de Informática - Universidad Nacional de La Plata





III-LIDI



# Agenda

- **Introducción: ¿Qué es Internet de las cosas?**
- Tecnologías de comunicación: NFC, RFID, Bluetooth 4.0, WiFi
- Tecnología M2M: Protocolo MQTT
- AWS – IoT
- Node-RED
- Laboratorio: “How-To”
- Consultas



# ¿Qué es Internet de las cosas?

**¿Internet of Thing = Machine to Machine?**

**¿IoT = M2M?**

M2M: es la comunicación Pto. a Pto., de una máquina o dispositivo con una máquina o dispositivo remoto. Esta comunicación se realiza por medio de las redes telefónicas tradicionales o móviles.

IoT: es mucho más que M2M, la comunicación se realiza vía redes basadas en IP, enviando los datos a Internet/Cloud.



# ¿Qué es Internet de las cosas?

Es un concepto que se refiere a la interconexión digital de objetos cotidianos por medio de internet. Alternativamente, Internet de las cosas es el punto en el tiempo en el que se conectarían a internet más “cosas u objetos” que personas. Si los objetos de la vida cotidiana tuvieran incorporadas etiquetas de radio, podrían ser identificados y gestionados por otros equipos, de la misma manera que si lo fuesen por seres humanos.



# ¿Qué es Internet de las cosas?

Existen dispositivos/”cosas” que son: sensores o actuadores

- Sensores: Un sensor es un dispositivo que está capacitado para detectar acciones o estímulos externos y responder en consecuencia. Estos pueden transformar las magnitudes físicas o químicas en magnitudes eléctricas. Ej.: un sensor de temperatura.
- Actuadores: Un actuador es un dispositivo capaz de transformar esta energía eléctrica en la activación de un proceso con la finalidad de generar un efecto sobre un proceso automatizado.



5 6 7 8 9



# ¿Qué es Internet de las cosas?

Definición de Internet of Things - IoT:

- Es un modelo/paradigma que abarca la interconexión de sensores y actuadores (objetos inteligentes) con el fin de lograr un ambiente inteligente, con un alto grado de captura autónoma de datos, transferencia de eventos e interoperabilidad.

Definición menos formal:

- Es una red de objetos “inteligentes” que utilizan “tecnologías” para comunicarse entre ellos y el entorno que los rodea.



III-LIDI



# Agenda

- Introducción: ¿Qué es Internet de las cosas?
- **Tecnologías de comunicación: NFC, RFID, Bluetooth 4.0, WiFi**
- Tecnología M2M: Protocolo MQTT
- AWS – IoT
- Node-RED
- Laboratorio: “How-To”
- Consultas





III-LIDI



# Tecnología de comunicación

## NFC – Near Fiel Communication

- Es una tecnología inalámbrica de corto alcance y alta frecuencia que permite el intercambio de datos entre dispositivos.
- Tasa de transferencia puede alcanzar los 424 kbit/s.
- Se utiliza para comunicación instantánea: identificación y validación de equipos, personas, Tarjeta de Crédito, etc.
- Distancia máxima 20 cm.





III-LIDI



# Tecnología de comunicación

## RFID – Radio Frequency IDentification

- El identificador por radiofrecuencia, es un sistema de almacenamiento y recuperación de datos remoto que usa dispositivos denominados etiquetas, tarjetas y tags RFID. El propósito fundamental de la tecnología RFID es transmitir la identidad de un objeto (similar a un número de serie único) mediante ondas de radio. Las tecnologías RFID se agrupan dentro de las denominadas Auto ID (automatización automática).





III-LIDI



# Tecnología de comunicación

## RFID – Radio Frequency IDentification





III-LIDI



# Tecnología de comunicación

## Bluetooth 4.0

- es una especificación industrial para Redes Inalámbricas de Área Personal (WPAN) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2,4 GHz. Los principales objetivos que se pretenden conseguir con esta norma son:
  - Facilitar las comunicaciones entre equipos móviles.
  - Eliminar los cables y conectores entre éstos.
  - Ofrecer la posibilidad de crear pequeñas redes inalámbricas y facilitar la sincronización de datos entre equipos personales.



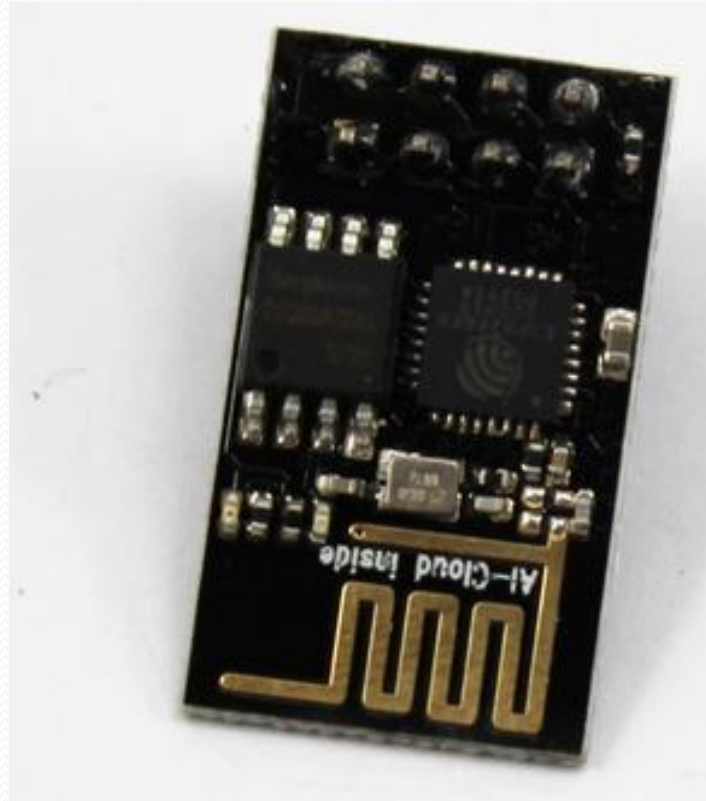
III-LIDI



# Tecnología de comunicación

## WiFi

- Módulo Arduino ESP8266





# Agenda

- Introducción: ¿Qué es Internet de las cosas?
- Tecnologías de comunicación: NFC, RFID, Bluetooth 4.0, WiFi
- **Tecnología M2M: Protocolo MQTT**
- AWS – IoT
- Node-RED
- Laboratorio: “How-To”
- Consultas



III-LIDI



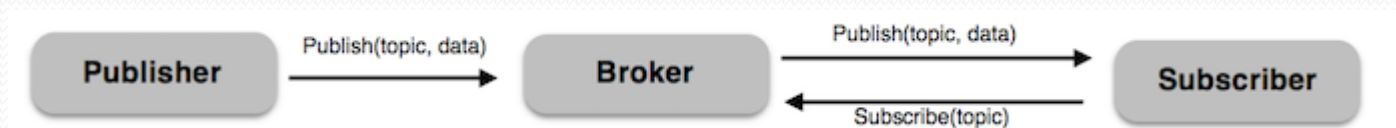
# Protocolo MQTT

- MQTT – Message Queuing Telemetry Transport.

WEB: <http://mqtt.org>

- Es un protocolo de conectividad para M2M y IoT, de publicación/suscripción ligero, diseñado para reducir al mínimo los requisitos de ancho de banda de red y los requisitos de recursos.

Ej: Sensor/temperatura/sala1



- Ofrece comunicación segura con TLS.





III-LIDI



# Protocolo MQTT

- Ofrece 3 niveles de calidad de servicio:
  - **QoS 0: “A lo sumo una vez la entrega”**: Los mensajes se entregan en base a los mejores esfuerzos de la red de Protocolo Internet subyacente. Se puede producir pérdida de mensajes. Esta calidad de servicio con la comunicación de datos de sensores ambientales, por ejemplo, no importa si una lectura individual se pierde, si la siguiente se publica poco después.
  - **QoS 1: “Al menos una vez la entrega”**: Se asegura que los mensajes llegan, pero se pueden producir duplicados.
  - **QoS 2: “Exactamente una vez la entrega”**: Se asegura que los mensajes llegan exactamente una sola vez. Es recomendable utilizar esta calidad de servicio con sistemas de facturación, por ejemplo. Los mensajes duplicados o perdidos pueden provocar un problema o generar cargos incorrectos.
- AWS IoT utiliza MQTT v.3.1.1 con QoS 0 y 1.





III-LIDI



# Protocolo MQTT

- Desde el 29-10-2014 es un estándar abierto OASIS

WEB: [https://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=mqtt](https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=mqtt)

Esto es muy importante, dado que puedo configurar un dispositivo, independientemente del sistema/plataforma de Broker de mensajes provisto por un Cloud Público o Privado.



III-LIDI



# Protocolo MQTT

- Existen Brokers comerciales y gratuitos:

- HiveMQ (Comercial): <http://www.hivemq.com/>



- CloudMQTT (Comercial): <https://www.cloudmqtt.com/>



- Mosquitto (Open Source): <http://mosquitto.org/>



Mosquitto

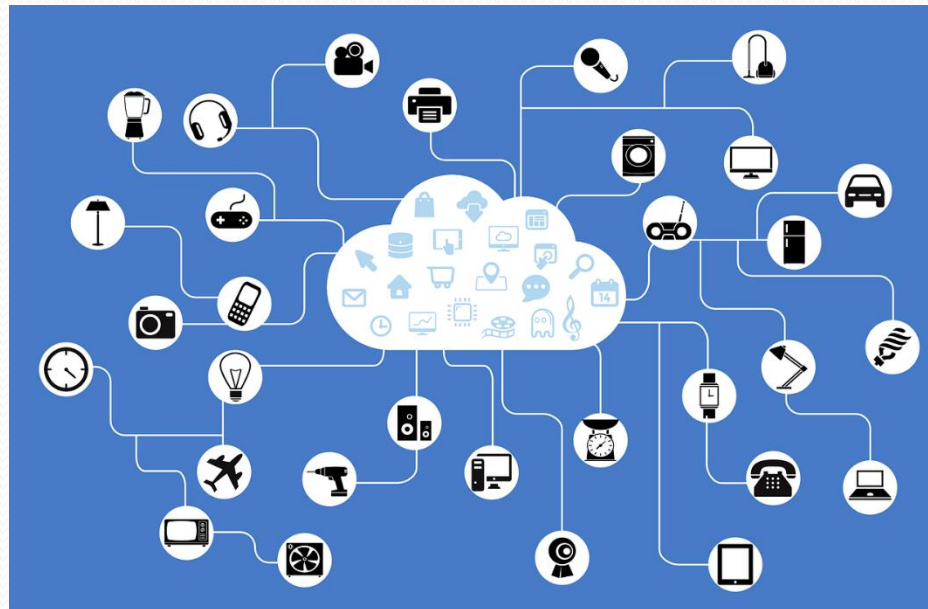
An Open Source MQTT v3.1 Broker



# Agenda

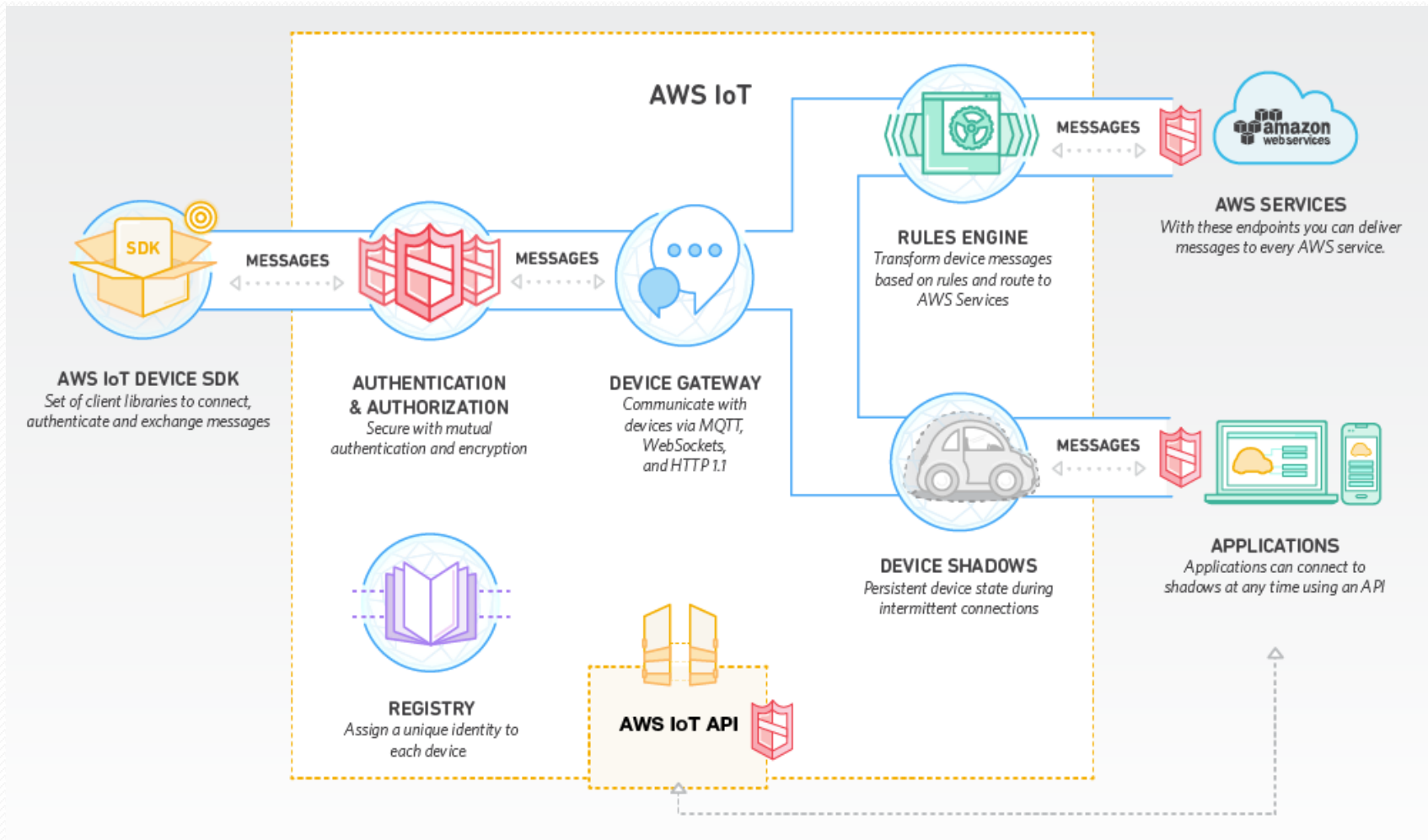
- Introducción: ¿Qué es Internet de las cosas?
- Tecnologías de comunicación: NFC, RFID, Bluetooth 4.0, WiFi
- Tecnología M2M: Protocolo MQTT
- **AWS - IoT**
- Node-RED
- Laboratorio: “How-To”
- Consultas

Es una plataforma que permite interconectar dispositivos de forma fácil y segura, con el fin de intercambiar datos y procesar los mismos por medio de aplicaciones en la nube.





# AWS IoT - Componentes





# AWS IoT - Componentes

- Broker de Mensajes (Comunicación vía Protocolos MQTT, HTTP o WebSockets).
- Gateway para dispositivos (Permite el intercambio de mensajes utilizando un modelo de publicación o suscripción sobre un tema/tópico concreto).
- Autenticación y autorización (Permite autenticación mutua y cifrado en todos los puntos de conexión. La autenticación puede ser la de AWS “SigV4” o basada en certificado X.509)
- Registro de dispositivos (Establece una identidad única para cada dispositivos, permitiendo definir la forma de conexión).





# AWS IoT – Regiones Disponibles

## Precios por región

Región	Precio
EE.UU. Este (Norte de Virginia)	5 USD por millón de mensajes
EE.UU. Este (Ohio)	5 USD por millón de mensajes
EE.UU. Oeste (Oregón)	5 USD por millón de mensajes
UE (Irlanda)	5 USD por millón de mensajes
UE (Frankfurt)	5 USD por millón de mensajes
UE (Londres)	5 USD por millón de mensajes
Asia Pacífico (Sídney)	6 USD por millón de mensajes
Asia Pacífico (Seúl)	6 USD por millón de mensajes
Asia Pacífico (Tokio)	8 USD por millón de mensajes
Asia Pacífico (Singapur)	8 USD por millón de mensajes





# Agenda

- Introducción: ¿Qué es Internet de las cosas?
- Tecnologías de comunicación: NFC, RFID, Bluetooth 4.0, WiFi
- Tecnología M2M: Protocolo MQTT
- AWS - IoT
- **Node-RED**
- Laboratorio: “How-To”
- Consultas





# Node-RED



The screenshot displays the Node-RED web interface. On the left, the 'input' category is expanded, showing various nodes like inject, catch, status, link, mqtt, http, websocket, tcp, udp, aws mqtt, AR.Drone, and Watson IoT. The central workspace shows a flow named 'Flow 1' with an 'AWS MQTT' node (status: connected) connected to a 'msg.payload' node, which is then connected to an 'LED YELLOW' node. The right sidebar has tabs for 'info' and 'debug'. The 'debug' tab is active, showing two messages:

```
2017-8-28 11:22:25 node: 37ef5333.a3c29c  
LED : msg.payload : string[1]  
"1"  
  
2017-8-28 11:22:37 node: 37ef5333.a3c29c  
LED : msg.payload : string[1]  
"0"
```

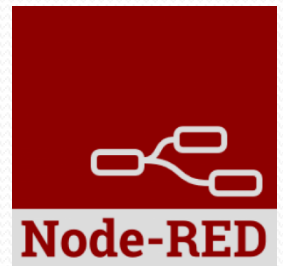


III-LIDI



# Node-RED

- Node-RED es una herramienta de programación visual desarrollada por IBM, que desde 2016 ofreció la herramienta como proyecto OpenSource.
- Node-RED, muestra visualmente las relaciones y funciones, y permite al usuario programar sin tener que escribir código.
- Node-RED es un editor de flujo basado en WEB, donde se puede añadir o eliminar nodos y conectarlos entre sí con el fin de hacer que se comuniquen entre ellos.



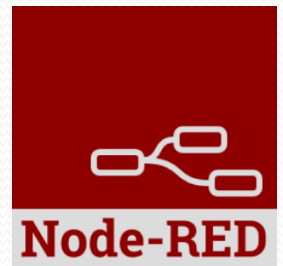


III-LIDI



# Node-RED

- En Node-RED, cada nodo es uno de los siguientes dos tipos: un nodo de inyección o un nodo de función.
- Los nodos de inyección producen un mensaje sin necesidad de entrada y lanzan el mensaje al siguiente nodo conectado a éste.
- Los nodos de función, por el contrario, tienen una entrada y realizan algún trabajo en él. Con una gran cantidad de estos nodos para elegir, Node-Red hace que el conectar los dispositivos de hardware, APIs y servicios en línea sea más fácil.

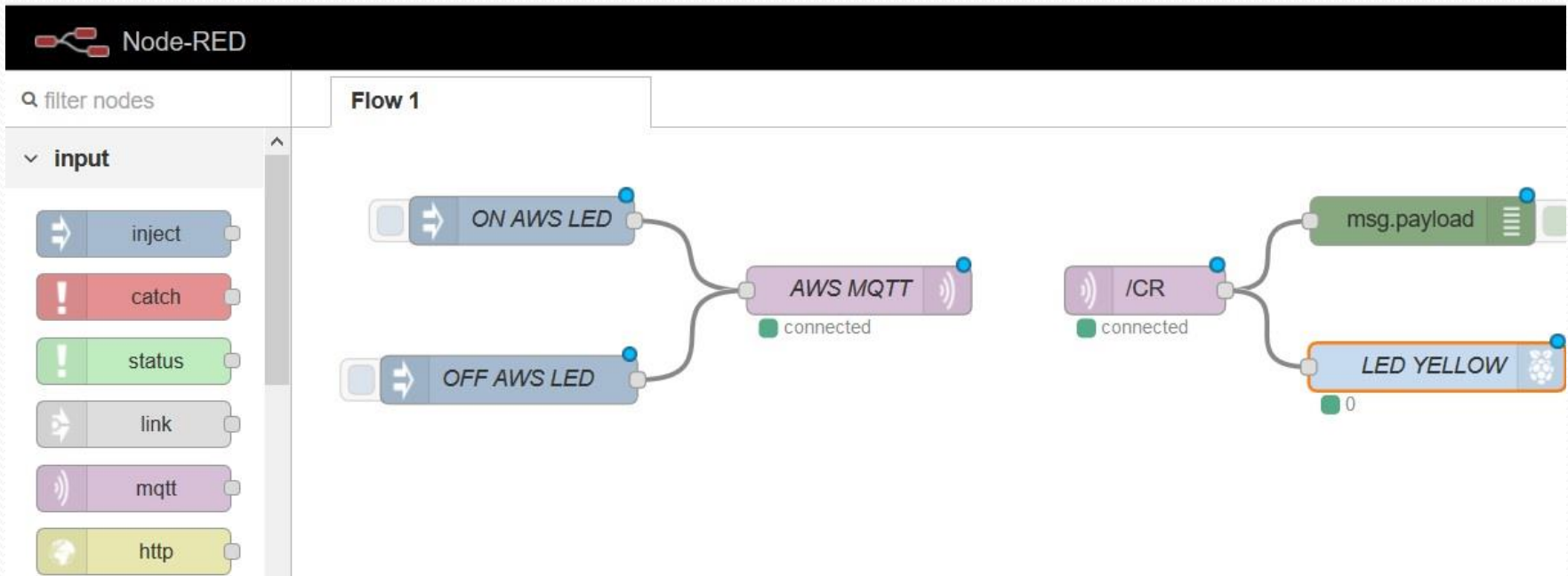




III-LIDI

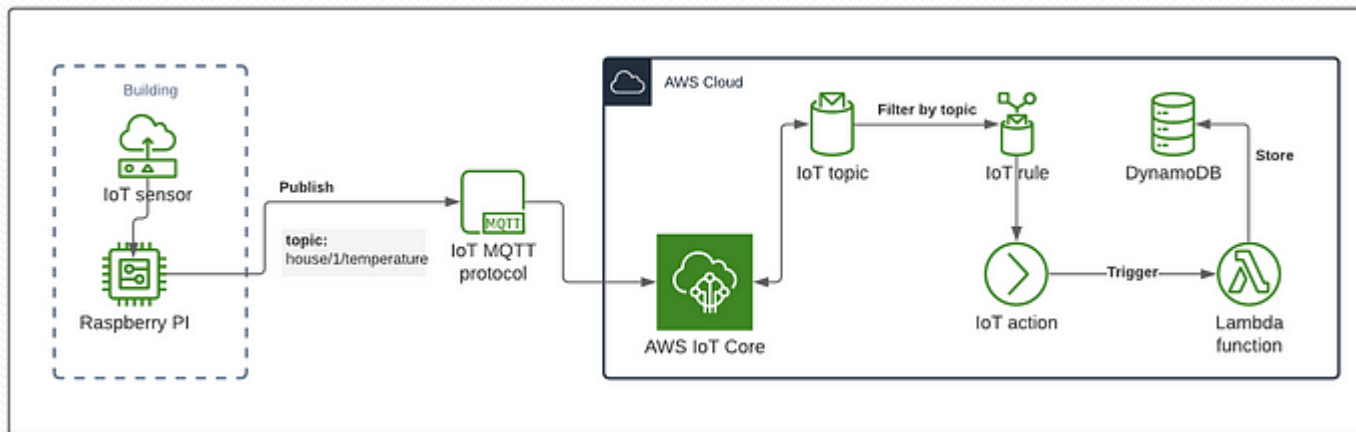


# Node-RED





# Ejemplo: Raspberry + AWS IoT





# Node.JS



- En Node-RED, el flujo creado es almacenado utilizando JSON
- Node-RED está basado en Node.js.
- Node.js es un entorno de ejecución para JavaScript construido con el motor de JavaScript V8 de Chrome. Node.js usa un modelo de operaciones E/S sin bloqueo y orientado a eventos, que lo hace liviano y eficiente. El administrador de paquetes de Node.js, npm, es el gestor mas grande de librerías de código abierto en el mundo.

<https://nodejs.org/dist/latest-v8.x/>





III-LIDI



# Agenda

- Introducción: ¿Qué es Internet de las cosas?
- Tecnologías de comunicación: NFC, RFID, Bluetooth 4.0, WiFi
- Tecnología M2M: Protocolo MQTT
- AWS - IoT
- Node-RED
- **Laboratorio: “how-to”**
- Consultas





III-LIDI



# Laboratorio: “how-to”

- 1) Lanzar una instancia t3.micro con Debian 13.
- 2) Conectarse vía SSH e instalar node-red:

```
# sudo apt update  
# sudo apt install snapd  
# sudo snap install core  
# sudo snap install node-red
```





III-LIDI



# Laboratorio: “how-to”

## 3) Verificar la instalación de Node-RED.

```
# sudo apt install net-tools  
# sudo netstat -na | grep 1880
```

```
admin@ip-172-31-9-35:~$ sudo netstat -na | grep 1880  
tcp        0      0 0.0.0.0:1880          0.0.0.0:*             LISTEN  
tcp        0      0 172.31.9.35:1880     163.10.22.55:59397    ESTABLISHED  
admin@ip-172-31-9-35:~$
```

## 4) Agregar regla de entrada para el puerto TCP 1880.

## 5) Verificar acceso a Node-RED desde un navegador WEB.



III-LIDI



# Laboratorio: “how-to”

Accedemos al dashboard de Node-RED:





# Laboratorio: “how-to”

Accedemos a la consola de AWS, al servicio “IoT Core”:

## AWS IoT

Conecte, pruebe y administre de forma segura los dispositivos de IoT.

AWS IoT admite miles de millones de dispositivos y billones de mensajes, y puede procesar y dirigir estos mensajes a puntos de enlace de AWS y a otros dispositivos de manera fiable y segura.

### Introducción a AWS IoT

La conexión rápida le guía a través de la conexión de un dispositivo en unos 15 minutos. Registrará su primer dispositivo y verá cómo el dispositivo envía mensajes MQTT a AWS IoT.

Conectar dispositivo





III-LIDI



# Laboratorio: “how-to”

AWS IoT > Conexión > Conectar un dispositivo

Step 1

**Preparar su dispositivo**

Step 2

Registrar y proteger su dispositivo

Step 3

Elegir plataforma y SDK

Step 4

Descargar kit de conexión

Step 5

Ejecutar kit de conexión

## Preparar su dispositivo [Información](#)

### Cómo funciona



En este asistente, crearemos un recurso de objeto en AWS IoT. Un recurso de objeto es una representación digital de un dispositivo físico o una entidad lógica.



Un recurso de objeto utiliza certificados a fin de proteger la comunicación entre el dispositivo y AWS IoT. Las políticas de AWS IoT controlan el acceso a los recursos de AWS IoT. Este asistente crea el certificado y la política para su dispositivo.



Cuando un dispositivo se conecta a AWS IoT, las políticas habilitan suscribirse y publicar mensajes MQTT con el agente de mensajes de AWS IoT.



III-LIDI



# Laboratorio: “how-to”

## Propiedades del objeto

☒ Crear un nuevo objeto

☐ Elegir un objeto existente

Nombre del objeto

Enter\_name

Ingrese un nombre único que contenga únicamente letras, números, guiones, dos puntos o guiones bajos. Un nombre de objeto no puede contener espacios.

## Configuraciones adicionales

Puede utilizar estas configuraciones para agregar detalles que le ayuden a organizar, administrar y buscar sus objetos.

- Tipo de objeto: *opcional*
- Atributos de objeto que permiten búsquedas : *opcional*
- Grupos de objetos: *opcional*
- Grupo de facturación : *opcional*



### Certificado y política para el dispositivo

El dispositivo requiere un certificado de dispositivo único a fin de autenticar de forma segura la identidad en AWS IoT y una política de AWS IoT que lo autorice a enviar y recibir mensajes. Crearemos estos recursos para su dispositivo de forma automática. Puede revisar y editar las propiedades más adelante, si es necesario.

Cancelar

Anterior

Siguiente



# Laboratorio: “how-to”

## Plataforma y SDK

Elija el SO de la plataforma y el SDK de dispositivos de AWS IoT que desee utilizar para su dispositivo.

### Sistema operativo de la plataforma del dispositivo

Este es el sistema operativo instalado en el dispositivo que se conectará a AWS.

☒ **Linux/macOS**

Versión de Linux: any

Versión de macOS: 10.13+

☐ **Windows**

Versión 10

### SDK para dispositivos con AWS IoT

Elija un SDK de dispositivo que esté en un idioma compatible con su dispositivo.

☒ **Node.js**

Versión 10+

Se necesita la instalación de Node.js y npm

☐ **Python**

Versión 3.6+

Se necesita la instalación de Python y Git

☐ **Java**

Versión 8

Se necesita la instalación de Java JDK, Maven y Git

Cancelar

Anterior

Siguiente



# Laboratorio: “how-to”

## Kit de conexión

Certificado

IsmaelTest.cert.pem

Script para enviar y recibir mensajes

start.sh

Clave privada

IsmaelTest.private.key

Política

IsmaelTest-Policy

[Ver política](#)

SDK para dispositivos con AWS IoT

Node.js

## Descargar

Sí lo ejecuta desde un navegador en el dispositivo, después de descargar el kit de conexión, estará en la carpeta de descarga del navegador.

Sí no lo está ejecutando desde un navegador en su dispositivo, tendrá que transferir el kit de conexión desde la carpeta de descarga de su navegador a su dispositivo utilizando el método que probó cuando preparó su dispositivo en el paso 1.

 **Descargar kit de conexión**



III-LIDI



# Laboratorio: “how-to”

Nombre	Tamaño	Comprimido	Tipo	Modificado	CRC32
..			Carpeta de archivos		
IsmaelTest.cert.pem	1.220	1.220	Archivo PEM	5/6/2023 17:37	8D399195
IsmaelTest.private.key	1.679	1.679	Archivo KEY	5/6/2023 17:37	E4C66943
IsmaelTest.public.key	451	451	Archivo KEY	5/6/2023 17:37	42FC3716
IsmaelTest-Policy	1.024	1.024	Archivo	5/6/2023 17:37	49112B01
start.sh	1.031	1.031	Archivo SH	5/6/2023 17:37	13B655C6





III-LIDI



# Laboratorio: “how-to”

## IsmaelTest-Policy



Esta es la política que hemos creado para el kit de conexión. Las políticas de AWS IoT autorizan el acceso a los recursos y acciones de AWS IoT.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Publish",
        "iot:Receive"
      ],
      "Resource": [
        "arn:aws:iot:us-east-2:920804707122:topic/sdk/test/java",
        "arn:aws:iot:us-east-2:920804707122:topic/sdk/test/python",
        "arn:aws:iot:us-east-2:920804707122:topic/sdk/test/js"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Subscribe"
      ],
      "Resource": [
        "arn:aws:iot:us-east-2:920804707122:topicfilter/sdk/test/java",
        "arn:aws:iot:us-east-2:920804707122:topicfilter/sdk/test/python",
        "arn:aws:iot:us-east-2:920804707122:topicfilter/sdk/test/js"
      ]
    }
  ]
}
```

 Copiar

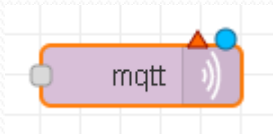
Aceptar



III-LIDI



# Laboratorio: “how-to”



### Edit mqtt out node

DeleteCancelDone

#### Properties

Server

Add new mqtt-broker...

Topic

IsmaelTopic

QoS

0

Retain

Name

Name

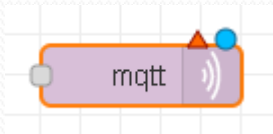
Tip: Leave topic, qos or retain blank if you want to set them via msg properties.



III-LIDI



# Laboratorio: “how-to”



Edit mqtt out node > **Add new mqtt-broker config node**

Cancel Add

**Properties**

Name

**Connection** Security Messages

Server  Port

☒ Connect automatically

☒ Use TLS

Protocol

Client ID

Keep Alive

Session ☒ Use clean session



III-LIDI



# AWS IoT – Node-Red

Properties

Name

Connection

Security

Messages

Server  Port

☒ Enable secure (SSL/TLS) connection

TLS Configuration

Client ID

Keep alive time (s)  ☒ Use clean session




☐ Use legacy MQTT 3.1 support




III-LIDI




# AWS IoT – Node-Red


 **Properties**  


☐ Use key and certificates from local files


 Certificate

 Upload


38ce78ab92-certificate.pe...



 Private Key


 Upload


38ce78ab92-private.pem.key




Passphrase

private key passphrase (optional)


 CA Certificate

 Upload


AmazonRootCA1.pem.txt



☒ Verify server certificate

 Server Name

for use with SNI

 Name

AWSIoT



III-LIDI



# AWS IoT – Node-Red

MQTT client Info Connected as iotconsole-1603394890493-1

Subscriptions

[Subscribe to a topic](#)  
[Publish to a topic](#)

TopicVM1

TopicVM1

Publish

Specify a topic and a message to publish with a QoS of 0.

Publish to topic

```
1 {}
2 {}
3 {}
```

Publish

Specify a topic and a message to publish with a QoS of 0.

Publish to topic

```
1 {}
2 {}
3 {}
```

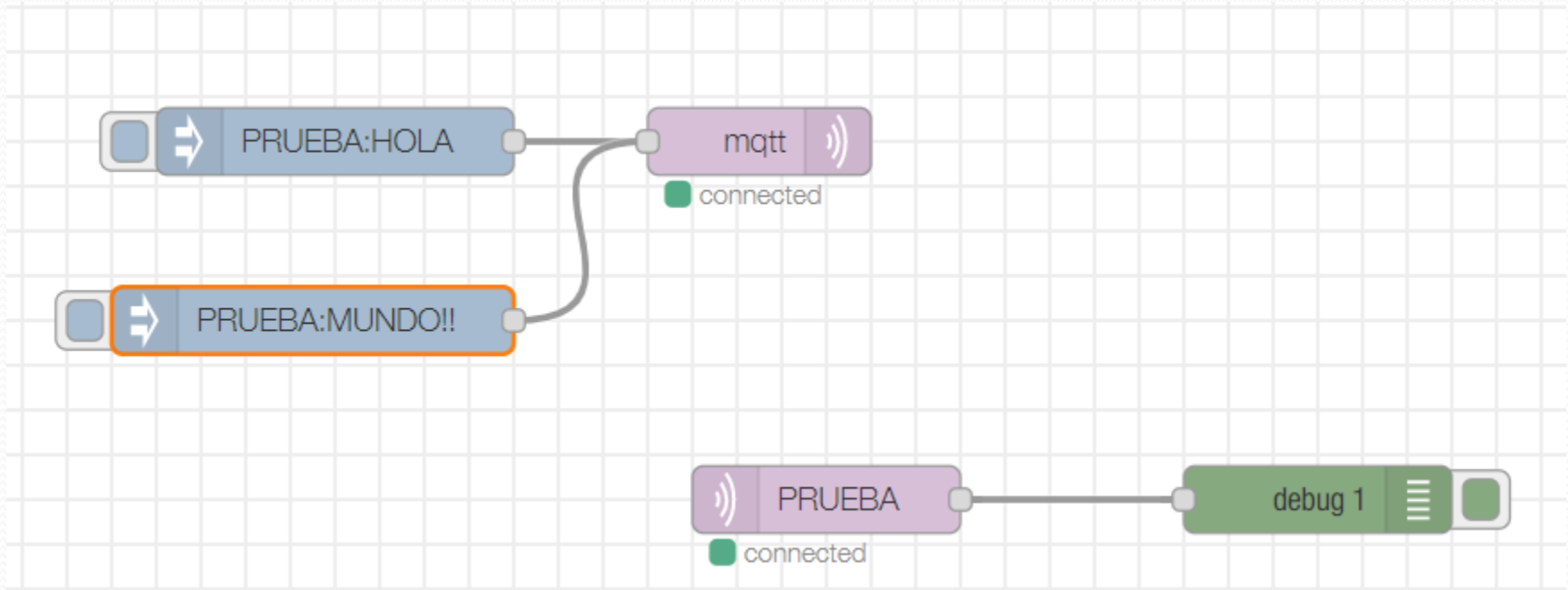
TopicVM1	October 22, 2020, 16:39:56 (UTC-0300)	Export Hide
TopicVM1	October 22, 2020, 16:39:55 (UTC-0300)	Export Hide
1		
TopicVM1	October 22, 2020, 16:39:53 (UTC-0300)	Export Hide
TopicVM1	October 22, 2020, 16:39:49 (UTC-0300)	Export Hide
1		



III-LIDI



# AWS IoT – Node-Red



14/9/2022, 15:45:58 node: debug 1

PRUEBA : msg.payload : string[4]

"HOLA"

14/9/2022, 15:46:00 node: debug 1

PRUEBA : msg.payload : string[7]

"MUNDO! !"



III-LIDI



# Laboratorio: “how-to”

**B) Instalaremos MOSQUITTO MQTT en Debian 13.**

```
# sudo apt install mosquitto  
mosquitto-clients -y
```





III-LIDI



# Laboratorio: “how-to”

**B-2) Editamos el archivo de configuración de mosquitto:**

```
# sudo nano /etc/mosquitto/mosquitto.conf
```

```
listener 1883
```

```
allow_anonymous false
```

```
password_file /etc/mosquitto/passwd
```

```
# sudo mosquitto_passwd -c  
/etc/mosquitto/passwd nombre_de_usuario
```



III-LIDI



# Laboratorio: “how-to”

**B-3) Reiniciamos el servicio de mosquito:**

```
# sudo systemctl restart mosquito
```

**B-4) Verificamos el servicio de mosquito:**

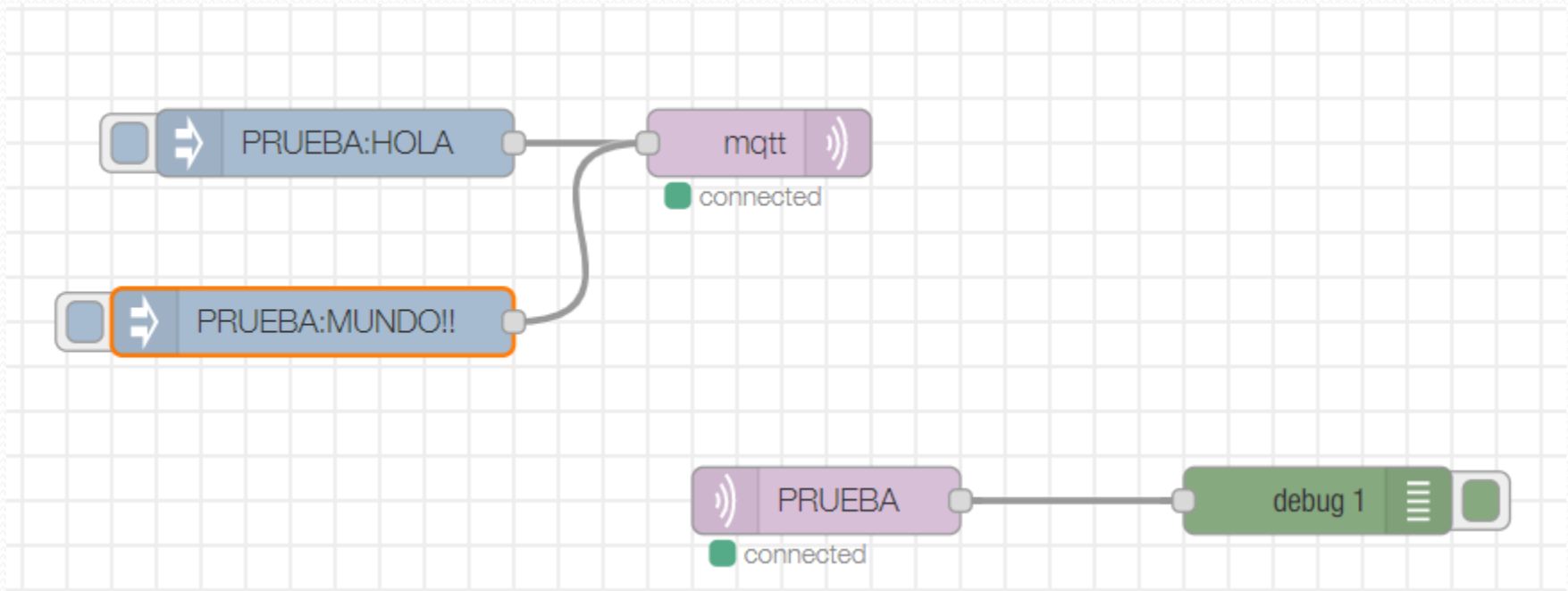
```
# sudo systemctl status mosquito
```



III-LIDI



# Mosquitto MQTT – Node-Red



14/9/2022, 15:45:58 node: debug 1

PRUEBA : msg.payload : string[4]

"HOLA"

14/9/2022, 15:46:00 node: debug 1

PRUEBA : msg.payload : string[7]

"MUNDO!!"



# Agenda

- Introducción: ¿Qué es Internet de las cosas?
- Tecnologías de comunicación: NFC, RFID, Bluetooth 4.0, WiFi
- Tecnología M2M: Protocolo MQTT
- AWS - IoT
- Node-RED
- Laboratorio: “How-To”
- **Consultas**