

# DISEÑO, DESARROLLO Y PROTOTIPADO DE UN GATEWAY M2M MULTIPROTOCOLO PARA APLICACIONES IOT

Aarón Cancio Gómez

Grado en Ingeniería Informática

Arduino

**ANTONI MORELL PÉREZ**  
**PERE Tuset PEIRÓ**

JUNIO DE 2018

# ESTRUCTURA DE CONTENIDOS

## INTRODUCCIÓN

### RESUMEN DEL TRABAJO

El presente trabajo trata sobre el diseño, desarrollo y prototipado de un Gateway M2M multiprotocolo para aplicaciones IoT basado en Arduino, Raspberry PI y el uso de software libre. El objetivo que se persigue es permitir la interconexión, comunicación bidireccional y adquisición de información procedente de nodos con diferentes protocolos de comunicación mediante el uso de conectores, gestionar la persistencia de la información en una base de datos local y permitir la interoperabilidad mediante el uso de una API REST y la publicación de los datos en un servicio en la nube

### VIDEO DE DEMOSTRACIÓN

<https://vimeo.com/user86270229/tfg-demostracion-acancio>

INTRODUCCIÓN

01

TECNOLOGÍA Y  
ARQUITECTURA

02

DESARROLLO Y  
PROTOTIPADO

03

PRODUCTO FINAL  
Y CONCLUSIONES

04

# OBJETIVOS Y PLANIFICACIÓN

## INTRODUCCIÓN

### OBJETIVOS DEL PROYECTO

El objetivo principal es diseñar, desarrollar y prototipar un sistema que permita la comunicación e interconexión con diferentes nodos mediante múltiples dispositivos y protocolos. Como objetivos secundarios

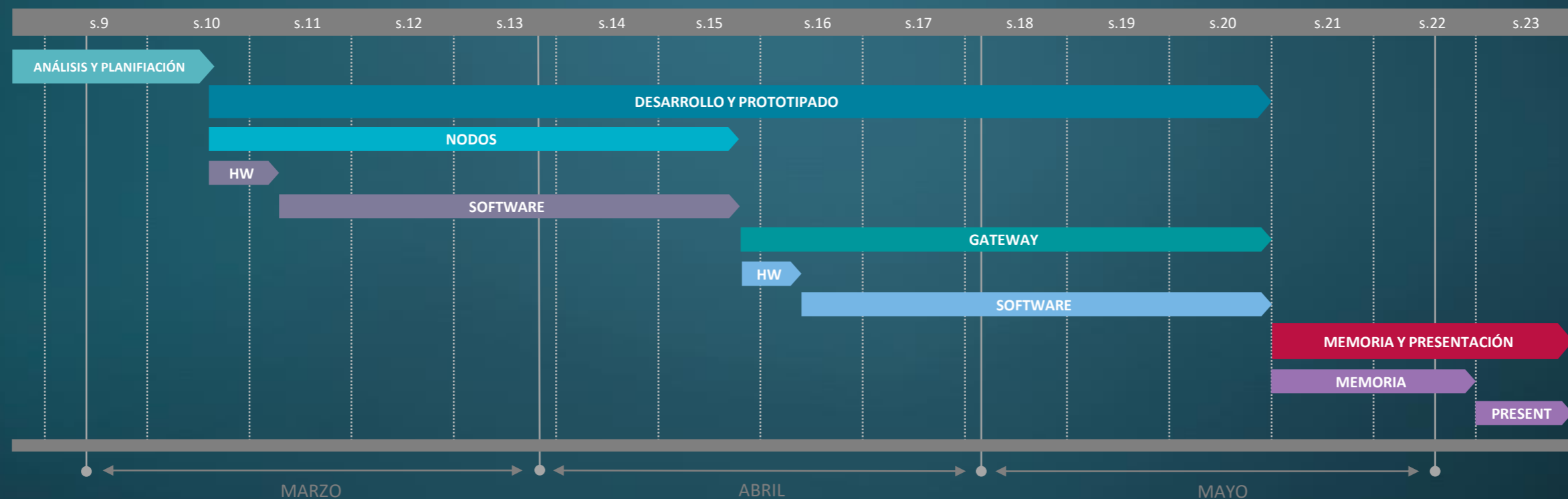
- Que ofrezca servicios de persistencia e interoperabilidad.
- Que esté basado en tecnologías libres y abiertas
- Que su arquitectura sea modular y su funcionamiento transparente

### OBJETIVOS PERSONALES

El objetivo principal es profundizar en el conocimiento de diferentes tecnologías soluciones tecnológicas, lenguajes y aplicaciones. Como objetivos secundarios:

- Ampliar el conocimiento de electrónica y señales
- Afianzar los conocimientos adquiridos hasta ahora
- Conocer las tendencias tecnológicas actuales y sondear las futuras

## PLANIFICACIÓN



# INTERNET OF THINGS Y MACHINE 2 MACHINE

TECNOLOGÍA Y ARQUITECTURA

A pesar de ser conceptos cercanos, M2M e IoT representan contextos y filosofías distintas de aproximarse a la comunicación entre máquinas.



## MACHINE 2 MACHINE

- Redes de nodos en sistemas aislados y acotados
- Consumo de la información local y restringida
- Sistemas homogéneos en toda la red
- Muy implantado en industria y redes de telemetría



## NUESTRA SOLUCIÓN

- Alta flexibilidad y funcionalidad a bajo coste
- Consumo de información local, público o mixto
- Sistemas heterogéneos interoperables
- Filosofía Do It Yourself, pequeñas empresas



## INTERNET OF THINGS

- Objetos en lugar de nodos, todo está conectado
- Consumo de información público o masivo
- Sistemas heterogéneos, gran cantidad de información
- Big Data, Cloud Computing, nuevas formas de negocio

# REQUISITOS DE LA SOLUCIÓN

## TECNOLOGÍA Y ARQUITECTURA

La comunicación entre Gateway y nodos ha de tener carácter bidireccional y poder establecerse mediante diferentes protocolos

El sistema ha de implementar una ontología que identifique a las entidades de manera jerárquica y unívoca dentro de la misma red.

Los nodos y el Gateway tienen que poder llevar a cabo las mismas acciones independientemente de su protocolo o modelo de comunicación

Los nodos deben poder informar periódicamente al Gateway del valor sus parámetros y poder responder a peticiones de lectura o asignación

El Gateway debe poder descubrir nodos conectados, recibir publicaciones y enviar peticiones de lectura o asignación a los nodos



Se ha de proporcionar una API REST con capacidad para leer y actualizar entidades y sea una interfaz única sobre el Gateway y los nodos

Se debe poder almacenar las lecturas publicadas así como la estructura jerárquica de la red en una base de datos relacional

Las lecturas publicadas por los nodos también se han de almacenar en una base de datos no relacional sin periodo límite

Se han de poder publicar las lecturas un servicio en la nube especializado en IoT que ofrezca capacidades de visualización

La incorporación de un nuevo nodo no ha de requerir de ninguna configuración en el Gateway y la comunicación debe ser automática

# PROTOSCOLOS DE COMUNICACIÓN

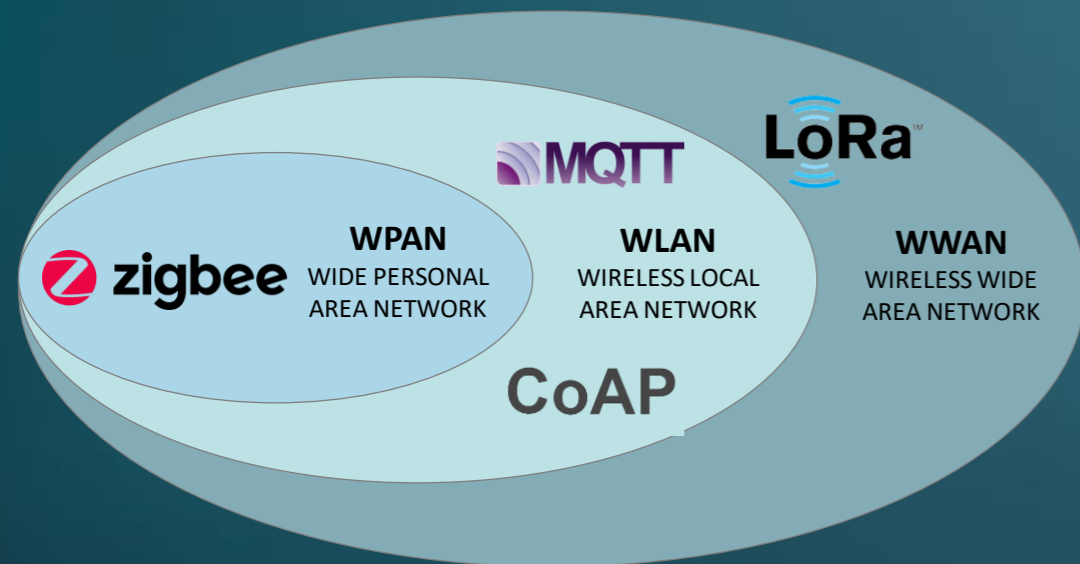
TECNOLOGÍA Y ARQUITECTURA

Los protocolos elegidos para la implementación son algunos de los más populares en su ámbito y rango de actuación.



## UN PROTOCOLO PARA CADA NECESIDAD

Cada uno de los protocolos está pensado para un contexto local distinto, diferenciado específicamente por la amplitud física de la red



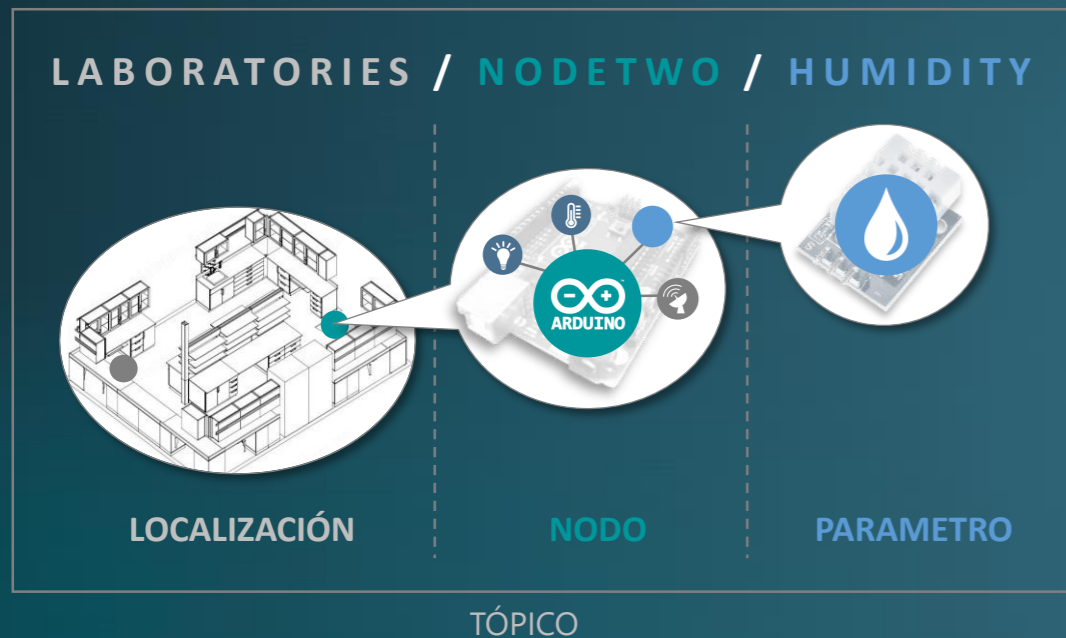
## PROTOSCOLOS EN EL MODELO OSI

No todos los protocolos actúan sobre todas las capas del modelo OSI. En algunos, como en LoRa, no usamos el stack completo

CAPA OSI	LoRa	ZigBEE	CoAP	MQTT
APLICACIÓN	Aplicación			
PRESENTACIÓN	LoRaWan	ZigBee	CoAP	MQTT
SESIÓN			UDP	TCP
TRANSPORTE			IPv4, IPv6	
RED	MAC			
ENLACE A DATOS	IEEE 802.15.4g	IEEE 802.15.4	IEEE 802.11	
FÍSICA	IEEE 802.11			

### ONTOLOGÍA JERÁRQUICA

Se define una jerarquía de entidades independiente del protocolo basada en las de las MQTT y COAP



### FORMATO DE MENSAJES

Se define una jerarquía de entidades independiente del protocolo basada en las de las MQTT y COAP

<pre>{   "COMMAND": "RESPOND",   "TOPIC": "GREENHOUSE/TWO/HUMIDITY",   "VALUE": 65,   "ID": "1D462Y530983G2" }</pre>	<pre>{   "VALUE": 65,   "ID": "1D462Y530983G2" }</pre>
ZIGBEE Y LORA	MQTT Y COAP

### ACCIONES Y COMANDOS

Se define una jerarquía de entidades independiente del protocolo basada en las de las MQTT y COAP

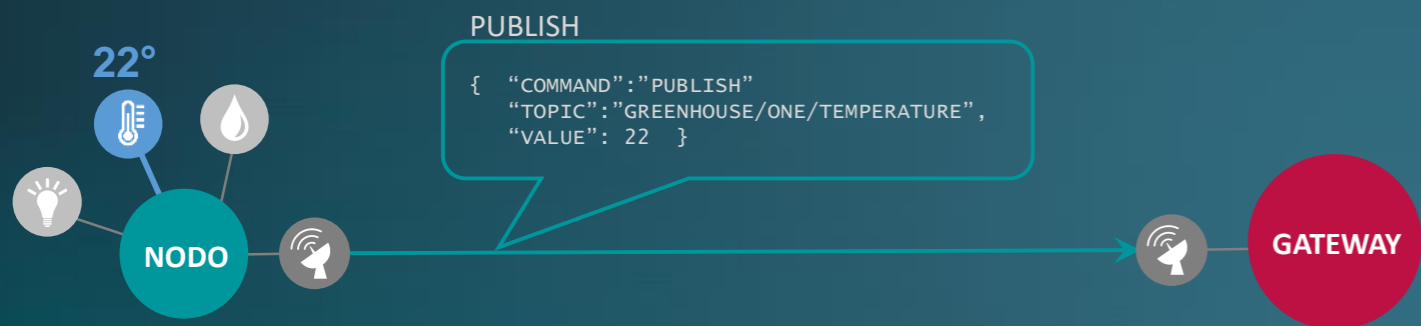


# PUBLICACIÓN, SOLICITUDES Y RESPUESTAS

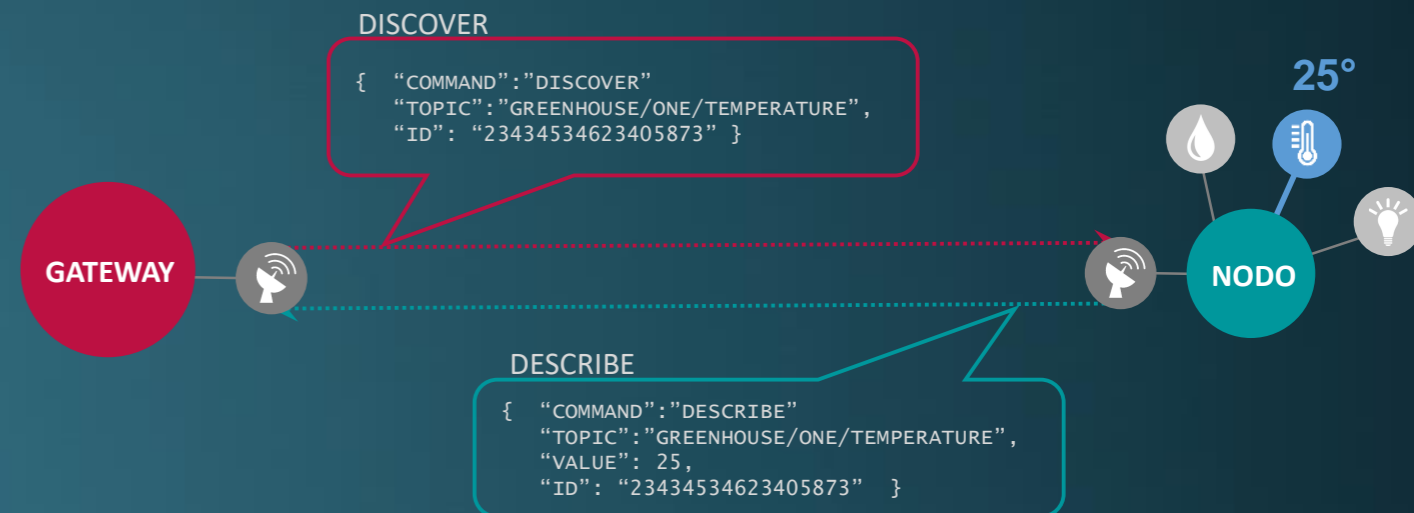
TECNOLOGÍA Y ARQUITECTURA

Se establece un modelo comunicación de tipo solicitud y respuesta pero incluyendo en cierta forma la publicación por parte de los nodos y la suscripción automática del Gateway

## PUBLICACIÓN y SUSCRIPCIÓN AUTOMÁTICA



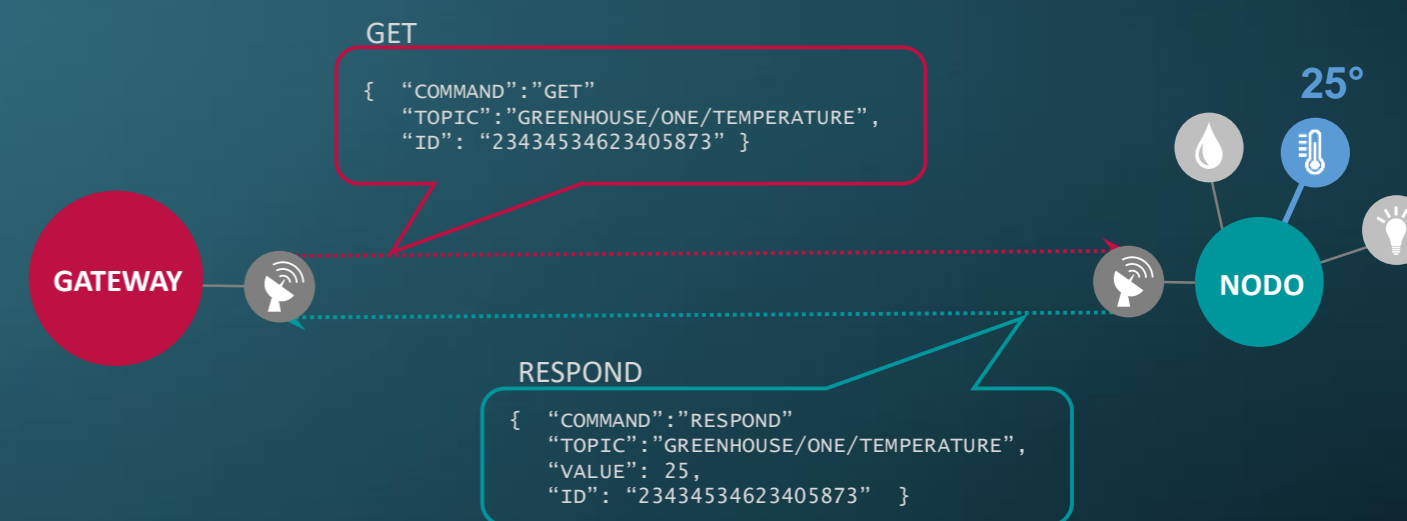
## SOLICITUD DISCOVER Y RESPUESTA DESCRIBE



## SOLICITUD SET Y RESPUESTA ACK



## SOLICITUD GET Y RESPUESTA RESPOND

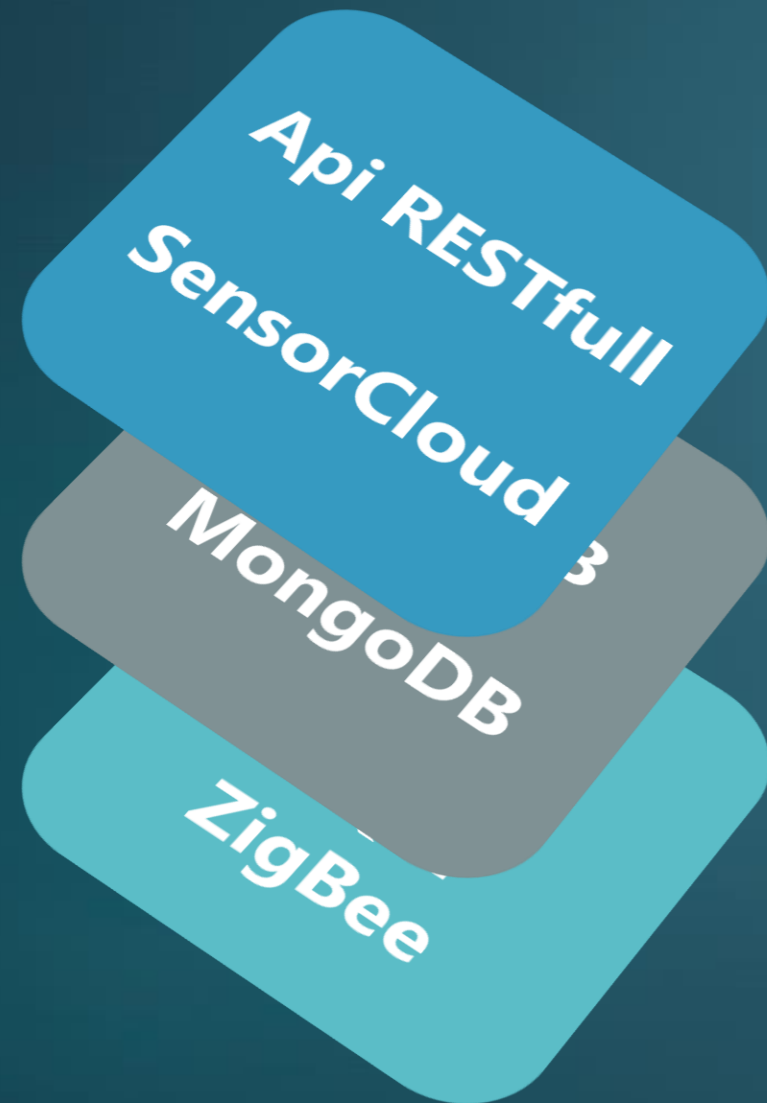




# ARQUITECTURA EN CAPAS DE LA SOLUCIÓN

## TECNOLOGÍA Y ARQUITECTURA

Para afrontar el desarrollo de la solución se plantea un diseño en capas que aglutinen las diferentes funciones y responsabilidades apilándose unas sobre otras



### INTEROPERABILIDAD

Capacidad de comunicarse con diferentes dispositivos

### PERSISTENCIA

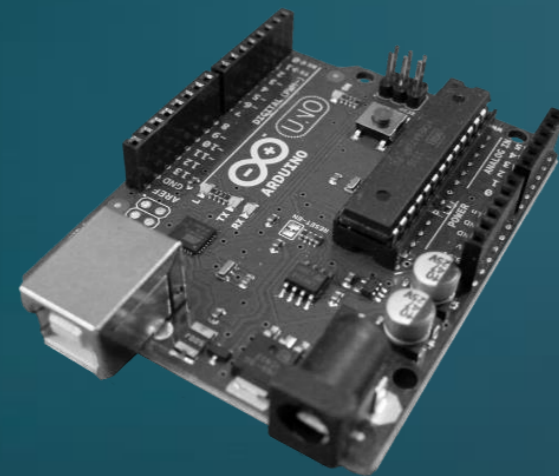
Capacidad de preservar los datos para su consume posterior

### ADQUISICIÓN

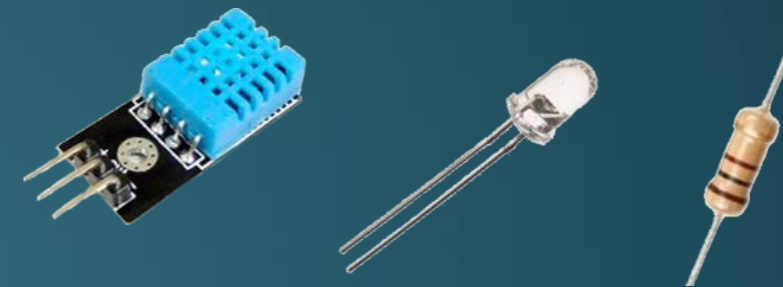
Capacidad de adquirir información de diferentes dispositivos



### DIAGRAMA DE CONEXIÓN



### HARDWARE COMÚN



#### SENSORES

Para medir la temperatura y la humedad se utilizará el sensor DHT11, un sensor digital de bajo coste que ofrece ambas lecturas con una respuesta rápida. La precisión del sensor no es muy elevada pero para el propósito del proyecto cumple perfectamente su función.

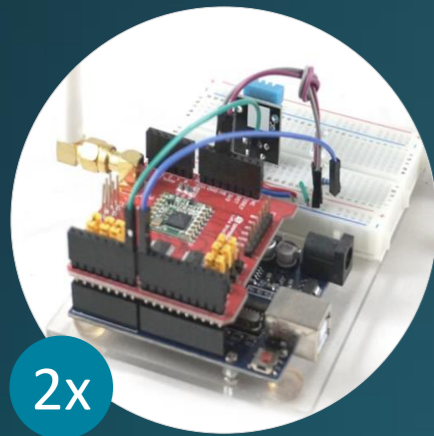
#### ACTUADOR

Un sencillo LED de dos terminales conectado en serie a una resistencia asume el rol de un actuador (como podría ser un relé o una electrobomba).

# HARDWARE DE LOS NODOS II

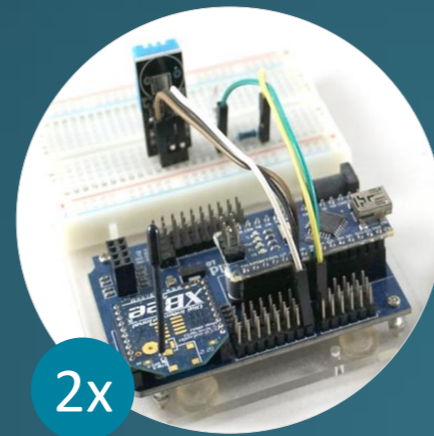
DESARROLLO Y PROTOTIPADO

El entorno de prototipado está formado por ocho nodos, dos por cada protocolo, muy similares entre sí. Los nodos están formados por una placa Arduino diferente para cada tipo de nodo, una interfaz de comunicaciones, un sensor y un actuador, todo ello montado sobre una protoboard.



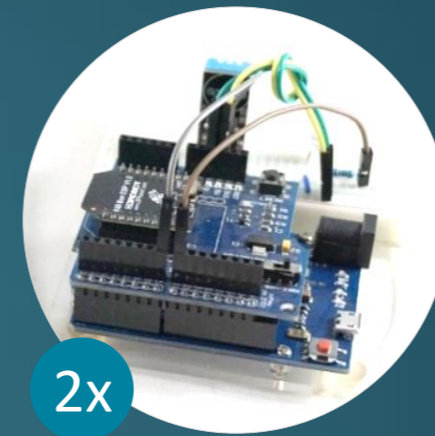
## Nodos LoRa

Placa Arduino Uno  
Escudo Dragino LoRa Shield



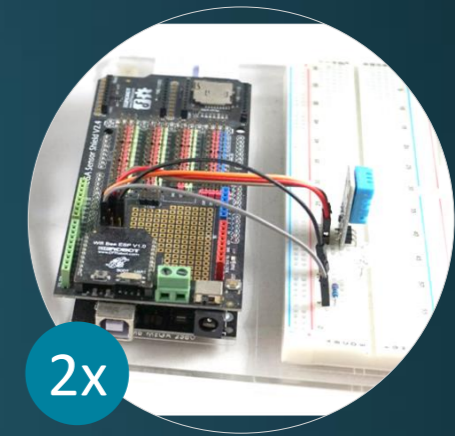
## Nodos ZigBee

Placa Arduino Nano  
Escudo Nano I/O Shield  
Digi Xbee S2



## Nodos MQTT

Placa Arduino Leonardo  
Escudo Bee  
EXP8266 WifiBee



## Nodos CoAP

Placa Arduino Mega 2560  
Escudo Bee  
EXP8266 WifiBee



ESP8266  
WIFI BEE



DIGI  
XBEE S2



NANO I/O  
SHIELD



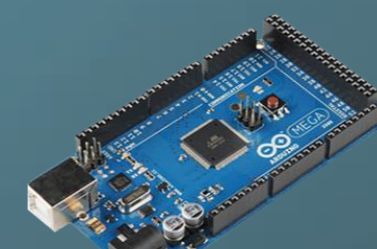
ARDUINO  
UNO



ARDUINO  
NANO



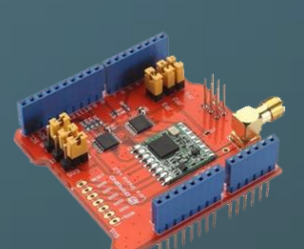
ARDUINO  
LEONARDO



ARDUINO  
MEGA 2560



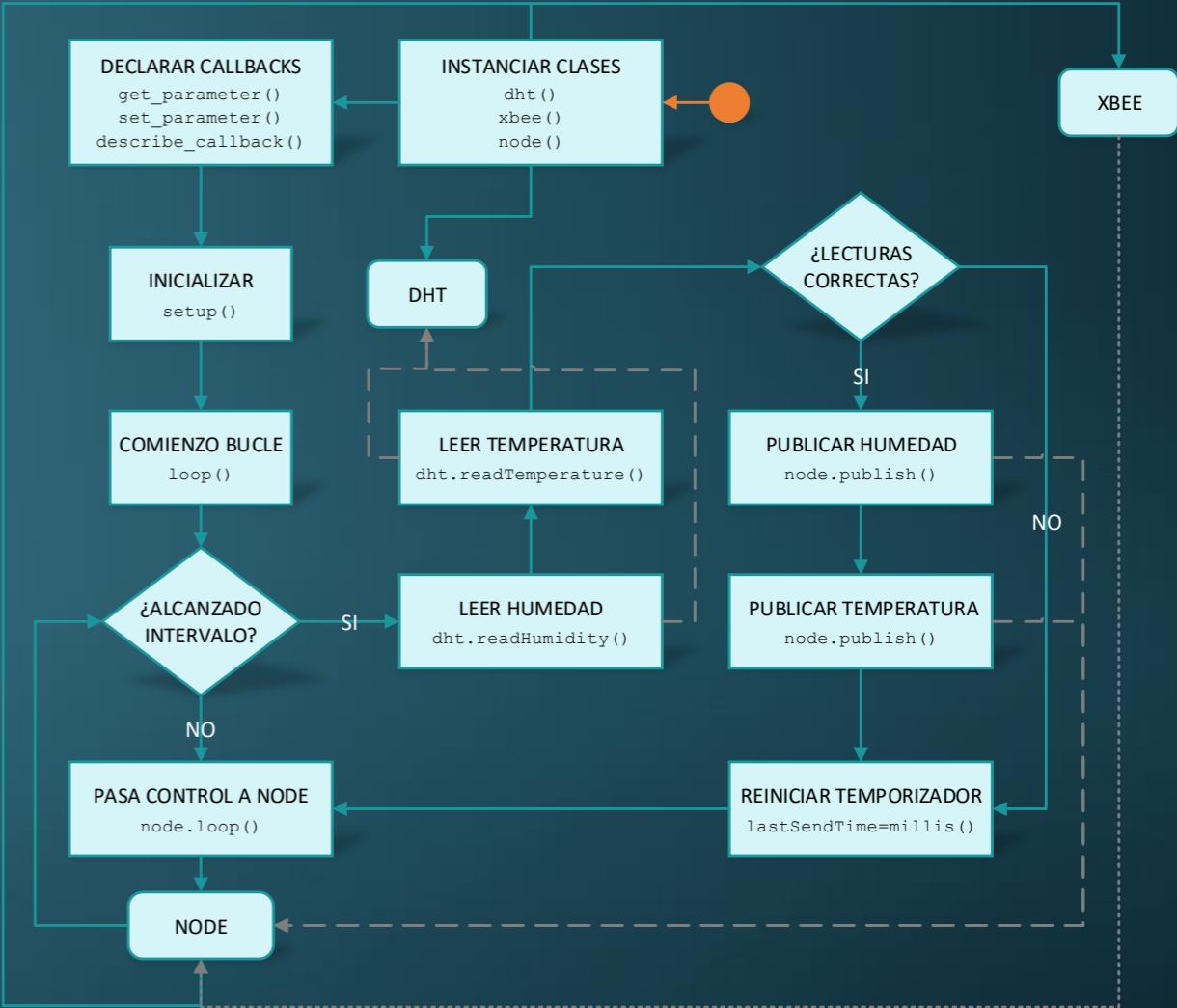
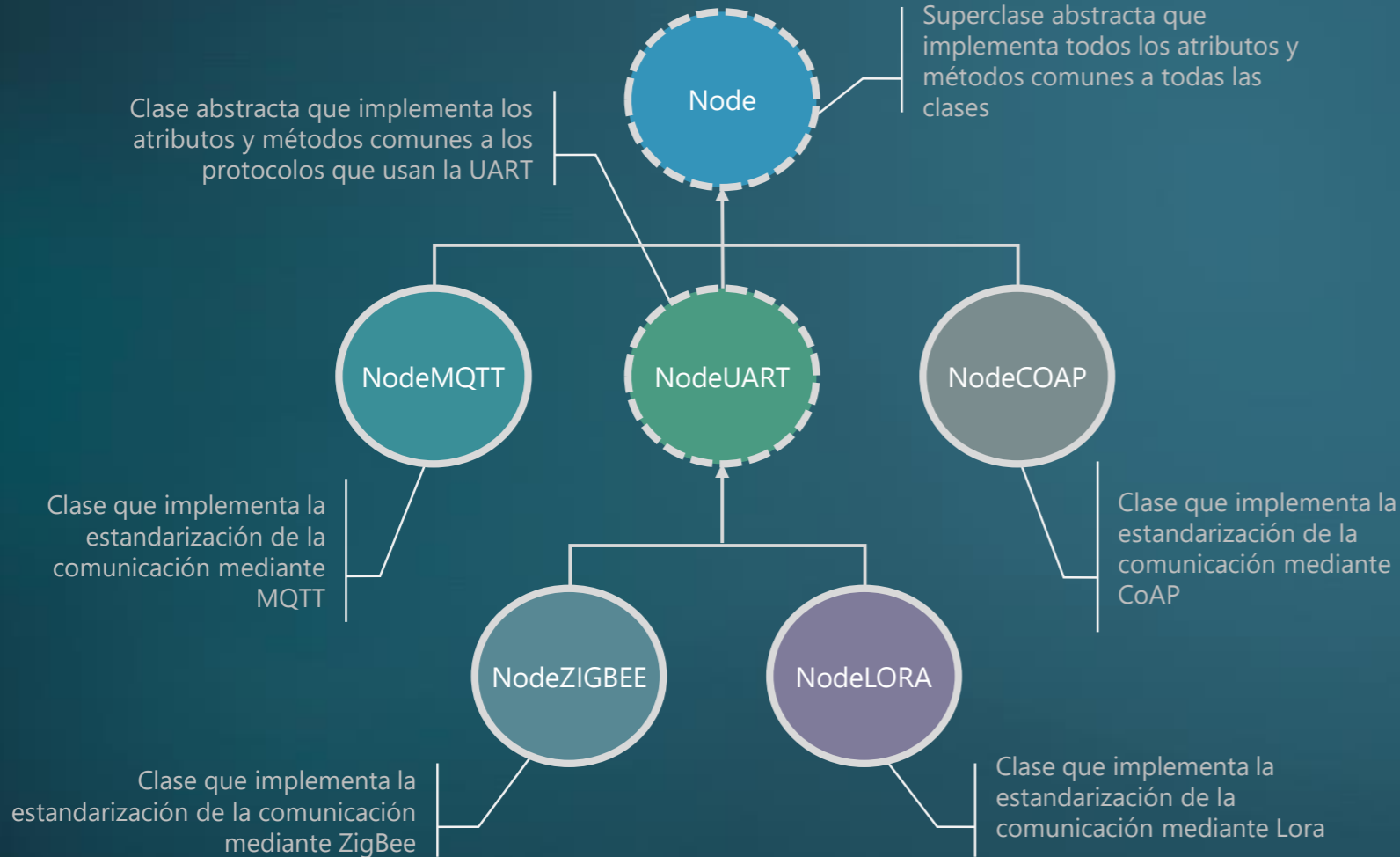
ARDUINO  
UNO



DRAGINO  
LORA SHIELD

## LIBRERÍA ARDUINO NODE

Para este proyecto se desarrolla una librería Arduino en lenguaje C++ que implementa los métodos encargados de gestionar los diferentes comandos y acciones que representan el núcleo de la funcionalidad del sistema: el modelo de intercambio de mensajes común a todos los nodos así como métodos de codificación y decodificación y otros métodos auxiliares

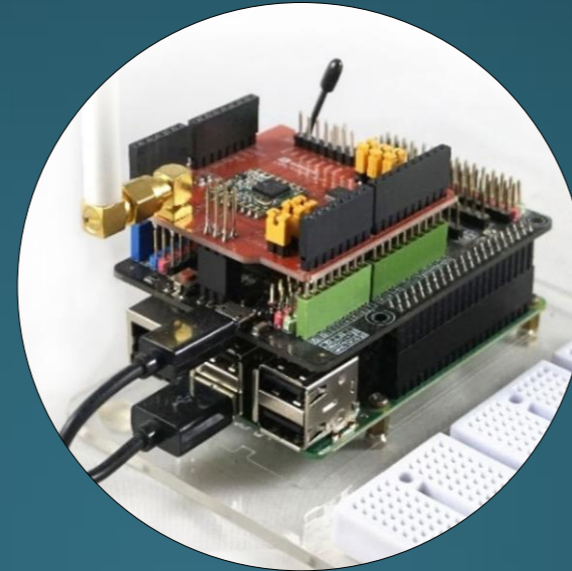
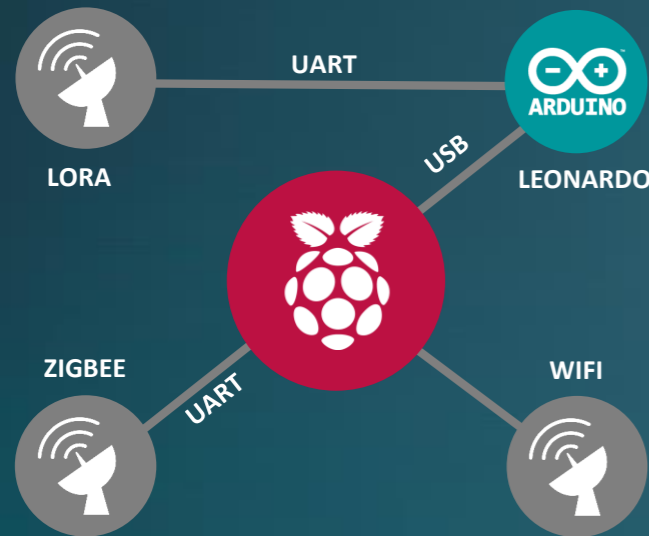


# HARDWARE DEL GATEWAY

DESARROLLO Y PROTOTIPADO

12

## DIAGRAMA DE CONEXIÓN



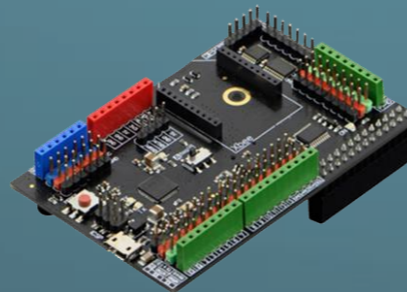
## Gateway

SBC Raspberry PI 3 B  
Escudo de expansión Arduino  
Draguino Lorashield  
Digi XBEE S2

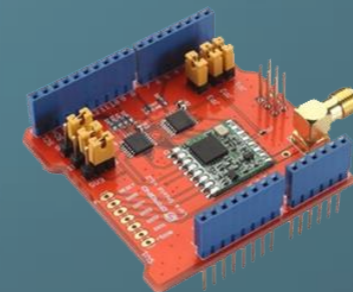
- La Raspberry Pi 3B lleva WiFi integrado para la comunicación mediante MQTT y CoAP además de la conexión a Internet
- Se utiliza un escudo de expansión con una doble finalidad
  - Conectar el módulo ZigBee a la UART de la Raspberry mediante el GPIO
  - Ofrecer una placa completa Arduino Leonardo sobre la que conectar el Draguino Lora Shield
- Usar un Arduino como intermediario para la conexión LoRA es un compromiso de diseño basado en el hardware disponible



RASPERRY  
PI 3B



ESCUDO DE  
ESPANSIÓN ARUIDNO



DRAGUINO  
LORA SHIELD

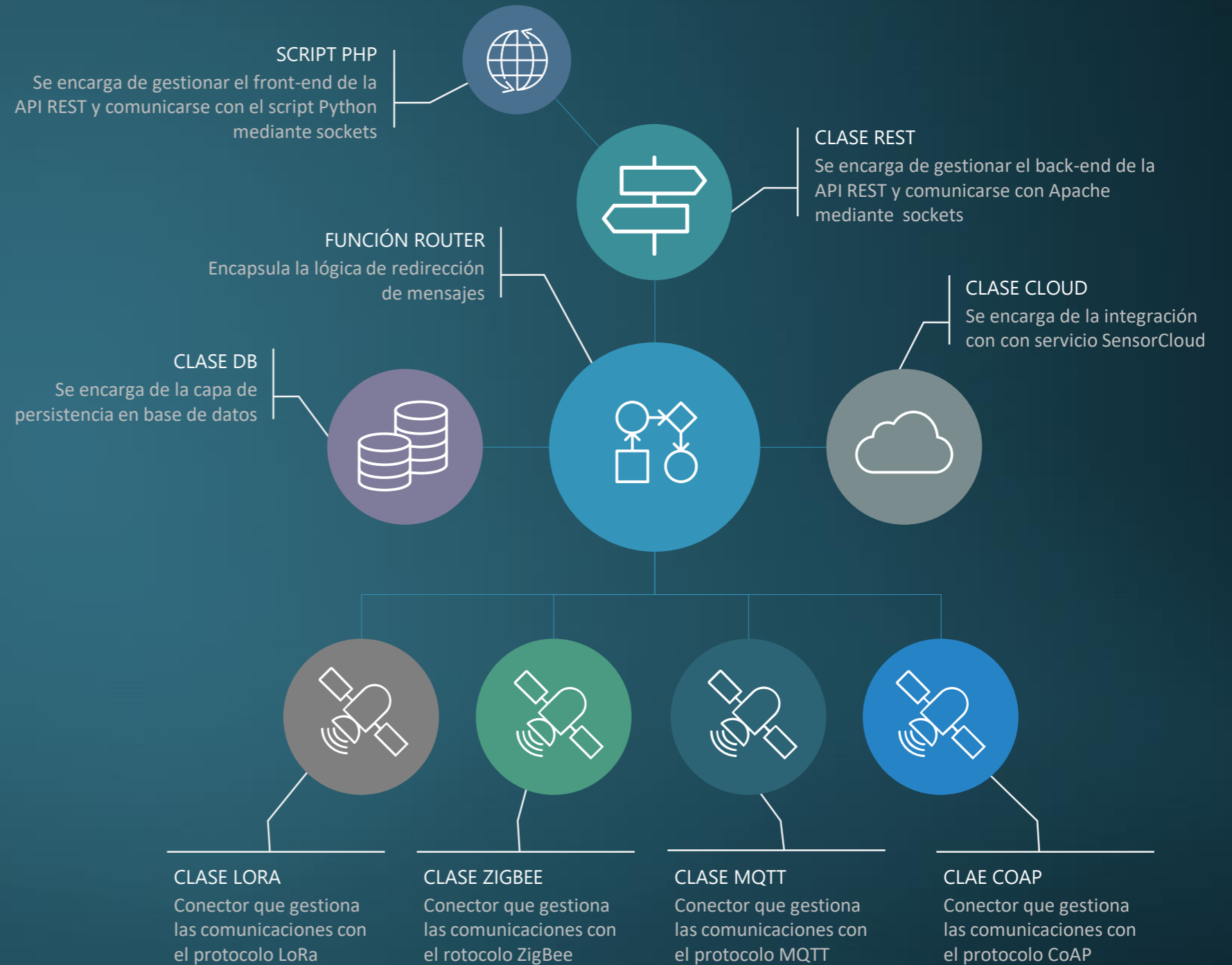


DIGI  
XBEE SB2

# SOFTWARE DEL GATEWAY

DESARROLLO Y PROTOTIPADO

La aplicación principal del Gateway es un Script en Python que gestiona toda la lógica de la comunicación así como las capas de servicio



# CAPA DE ADQUISICIÓN

DESARROLLO Y PROTOTIPADO

Cada uno de los cuatro conectores correspondientes a los protocolos tiene especificidades causadas por el propio modelo de comunicación o las librerías que se utilizan para gestionarlo pero todos los conectores comparten los cuatro métodos fundamentales







### BASE DE DATOS MONGODB

Como base de datos no relacional hemos elegido MongoDB

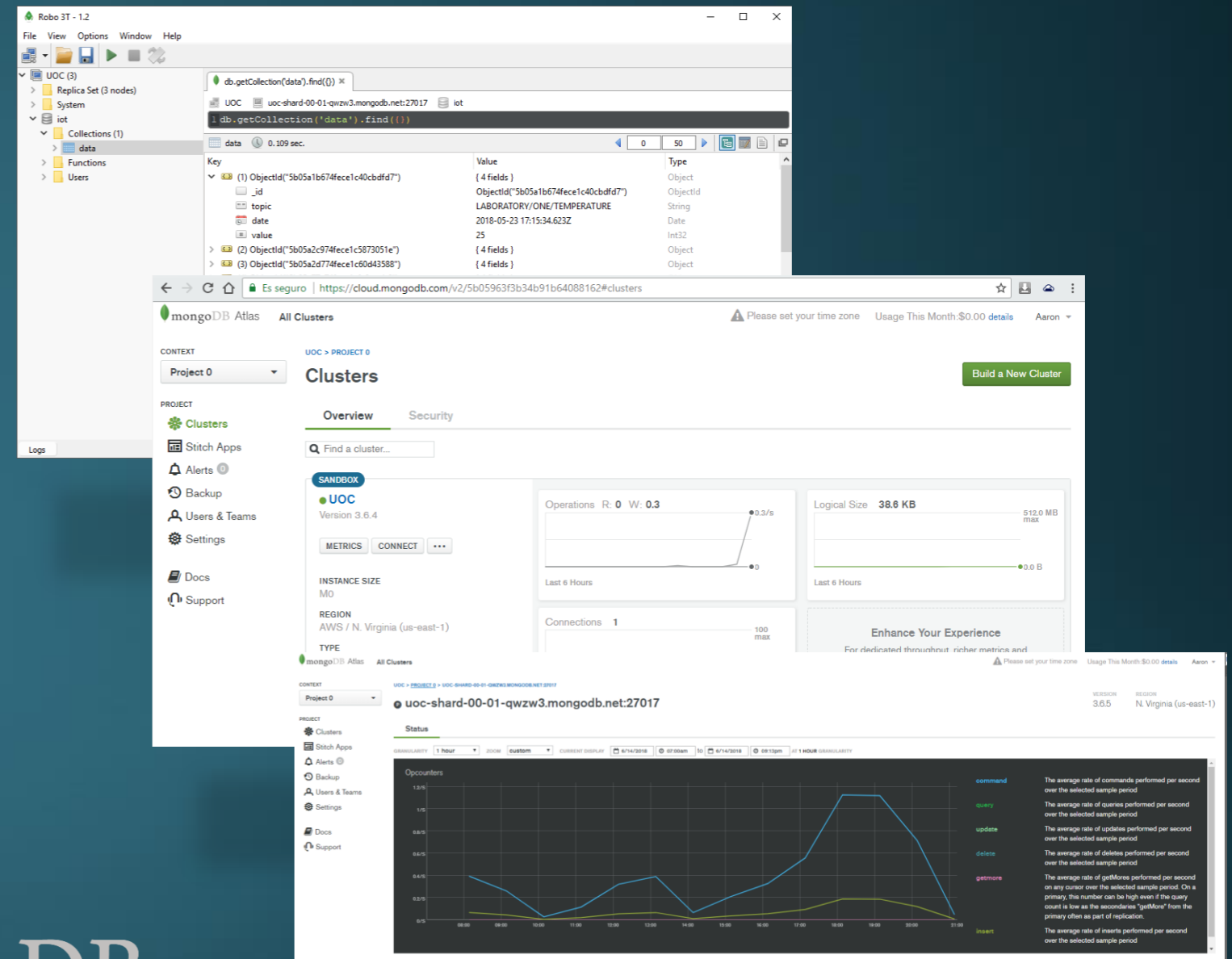
En lugar de ejecutar una instancia en local hemos optado por utilizar una instancia en la nube montada sobre un cluster que proporciona de manera gratuita (para algunos escenarios) el servicio MongoDB Atlas

### COLECCIONES Y DOCUMENTOS

MongoDB utiliza un formato de JSON binario para almacenar los documentos (equivalente a registros en SQL) en colecciones (equivalente a tablas en SQL)

- Almacenamos los documentos en la colección **iot**
- La estructura de campos del documento es la siguiente

```
{  
  "TOPIC": "<topic>",  
  "VALUE": <value>,  
  "DATETIME": <timestamp>  
}
```



# CAPA DE INTEROPERABILIDAD · SENSORCLOUD

## DESARROLLO Y PROTOTIPADO

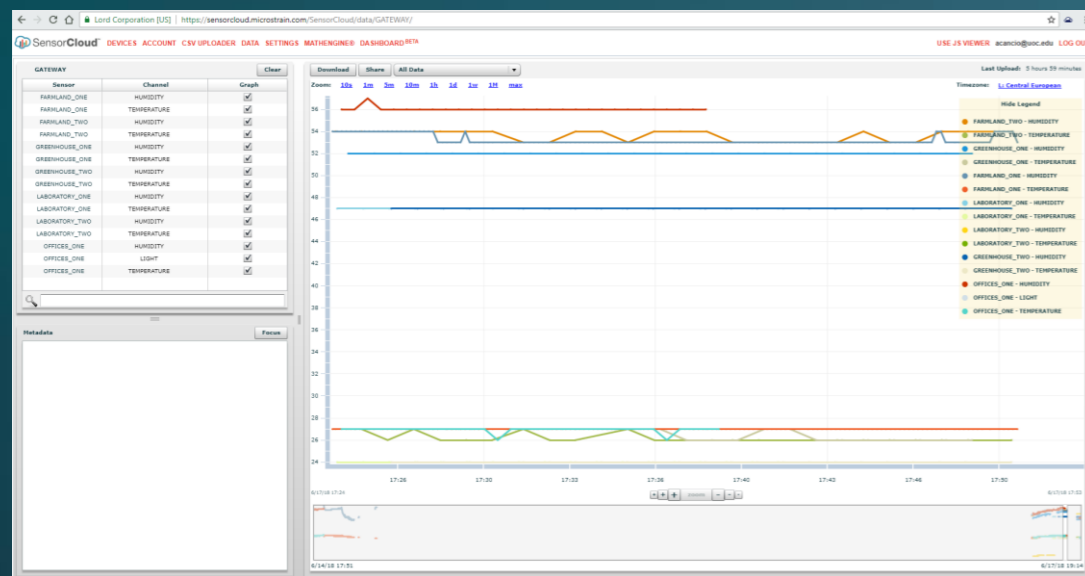
En la capa de interoperabilidad se ha implementado la integración con un servicio en la nube especializado en IoT denominado SensorCloud. Este servicio ofrece almacenamiento, visualización, explotación de datos, alarmas y gestión remota y funciones matemáticas avanzadas



- La integración con SensorCloud se realiza mediante una REST Api
- La autenticación se lleva a cabo mediante clave y token
- Toda lectura publicada por los nodos se informa al servicio Cloud

### EQUIVALENCIA ENTRE ENTIDADES

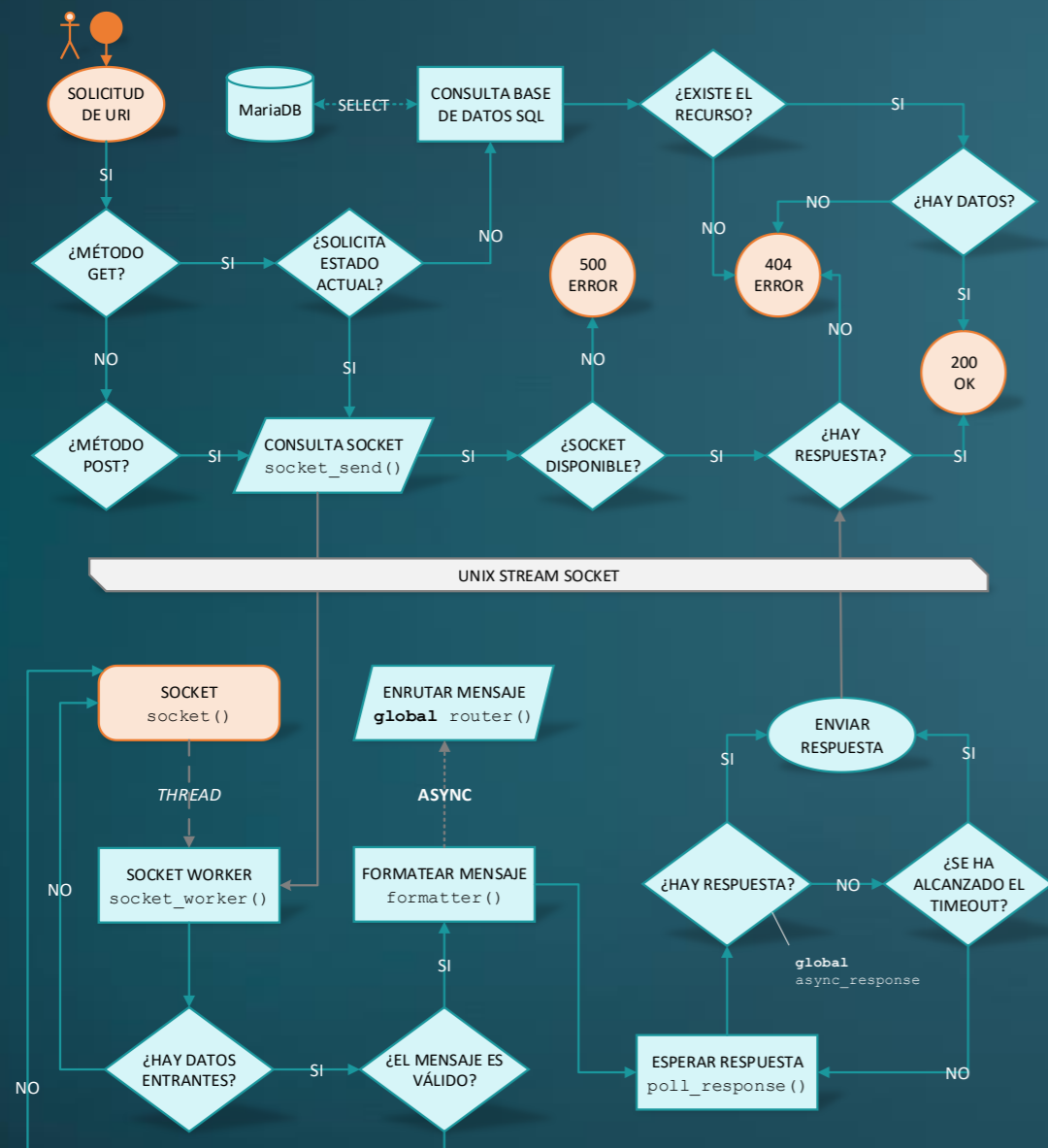
ONTOLOGÍA	NIVEL 1	NIVEL 2	NIVEL 3
SENSORCLOUD	DEVICE	SENSOR	CHANNEL
GATEWAY	GATEWAY	LOCATION + NODE	PARAMETER



# CAPA DE INTEROPERABILIDAD · REST API

## DESARROLLO Y PROTOTIPADO

En la capa de servicios de interoperabilidad hemos creado un servicio web basado en arquitectura REST que proporciona una API para la consulta y actualización de recursos



La funcionalidad y responsabilidad de la API REST está dividida en dos partes:

- Un front-end gestionado por un script PHP corriendo sobre un servidor Apache
- Un back-end gestionado por la aplicación en Python

La comunicación entre ambas se realiza por medio de stream sockets de Unix.

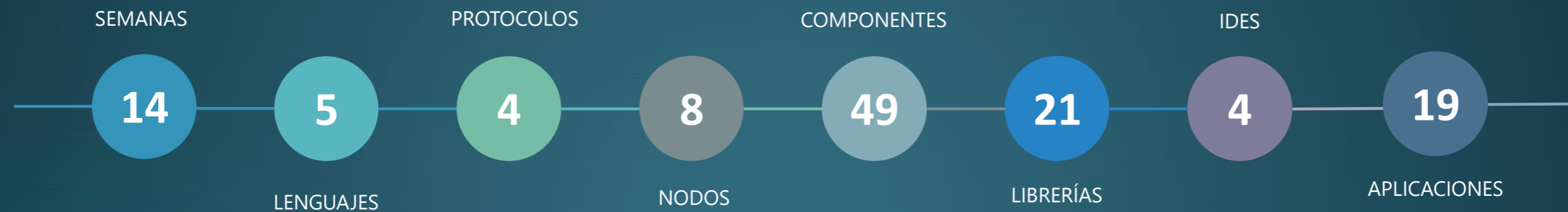
## OPERACIONES, MÉTODOS Y URIS DE LA API

OPERACIÓN	METODO	URI	PARAMETRO
DISCOVER	GET	/API/V1	
DISCOVER	GET	/API/V1/<LOCATION>	
DISCOVER	GET	/API/V1/<LOCATION>/<NODE>	
READ LAST	GET	/API/V1/<LOCATION>/<NODE>/<PARAMETER>	
READ NUM	GET	/API/V1/<LOCATION>/<NODE>/<PARAMETER>/<1..1000>	
READ NOW	GET	/API/V1/<LOCATION>/<NODE>/<PARAMETER>/STATE	
UPDATE	POST	/API/V1/<LOCATION>/<NODE>/<PARAMETER>/	:VALUE

# PRODUCTO FINAL

## RESULTADOS Y VALORACIÓN

El resultado final del proyecto es un Gateway plenamente funcional capaz de interactuar con los nodos del entorno de prototipado en los cuatro protocolos ofreciendo además los servicios de las capas de persistencia e interoperabilidad de forma eficaz y sin ningún impacto en el rendimiento del Gateway



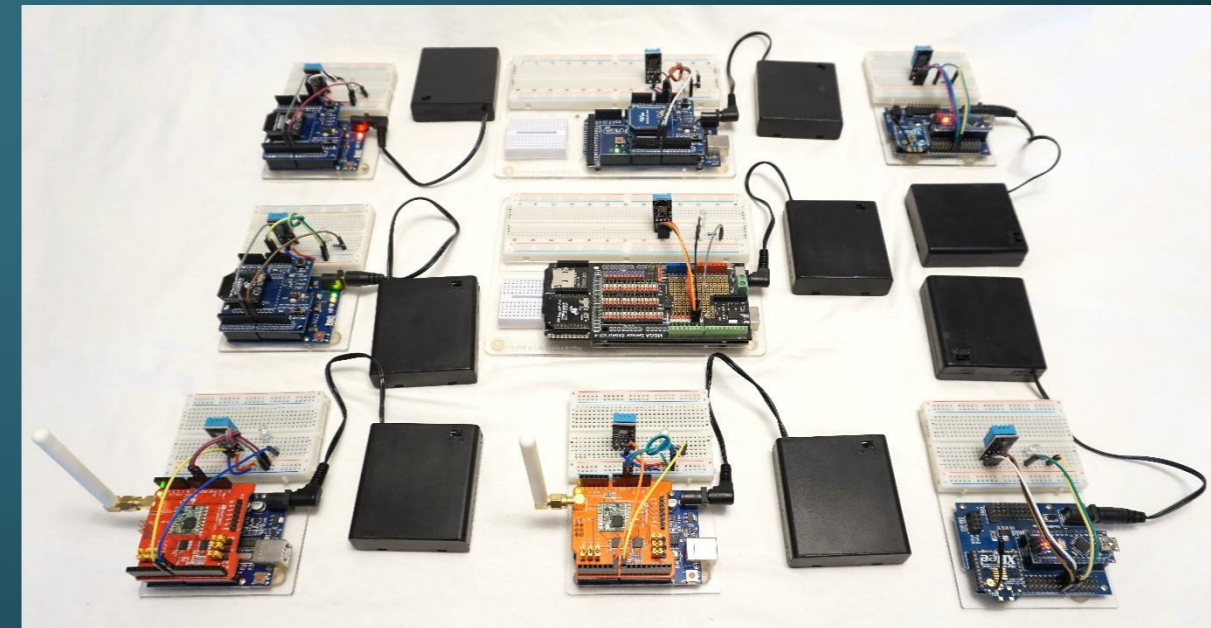
## GATEWAY



## SOFTWARE

- Script Gateway.py
- Script Gateway.php
- Script Gateway.sql
  - Librería Node
- 4 Sketches de Nodo
- 1 Sketch del Gateway

## NODOS



Aarón Cancio Gómez

Grado en Ingeniería Informática  
Arduino

Barcelona, Junio 2018