

PRESENTACIÓN FINAL SISTEMA EMBEBIDOS DISTRIBUIDOS

PABLO SLAVKIN, GONZALO LAVIGNA

PROFESORES

LEONARDO CARDUCCI
SEBASTIAN GARCÍA MARRA
FEDERICO ZACCHIGNA

FECHA

18/10/2019

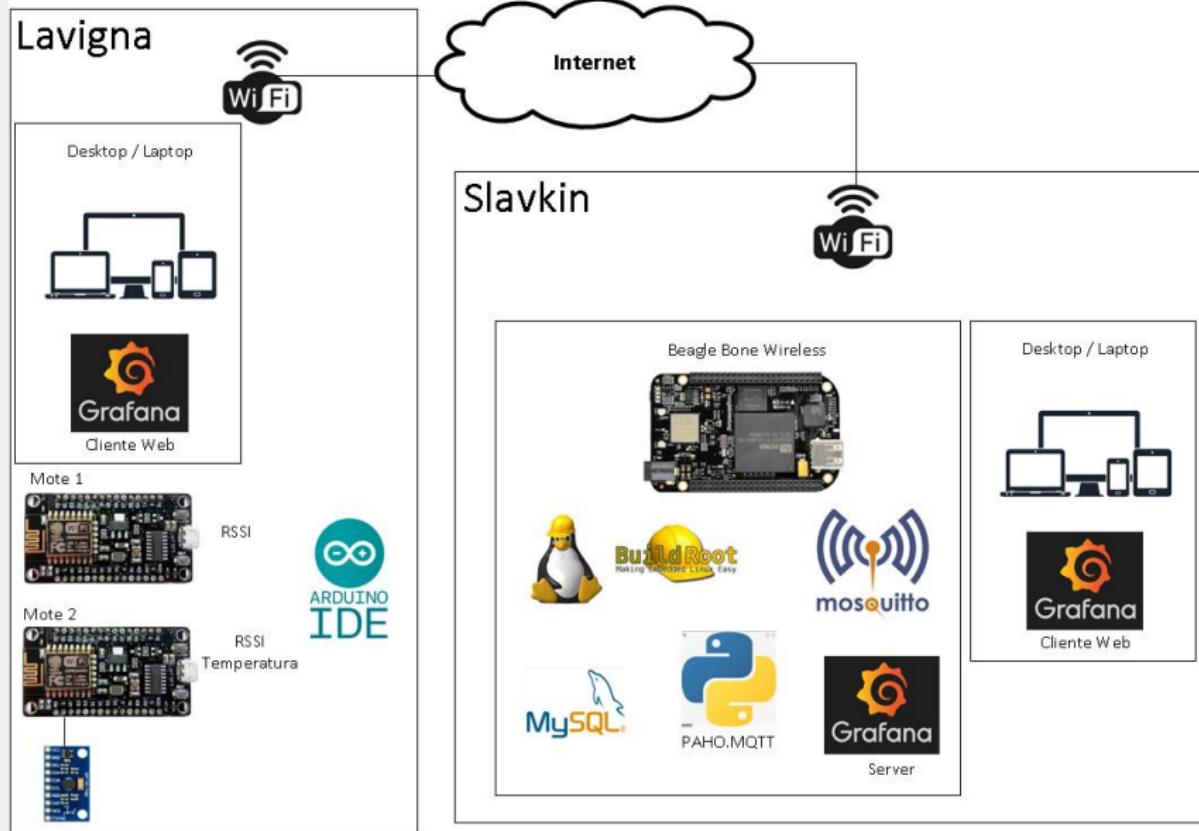
MOTIVACIÓN



■ Generar un ejemplo práctico con las siguientes características

- ▶ Instanciar broker MQTT en una SBC.
- ▶ Comunicación entre 2 redes wifi a través de la nube.
- ▶ Utilización de bases de datos.
- ▶ Utilización de un visor de telemetría.
- ▶ Implementación con distintas plataformas de HW.

BUILDROOT-MYSQL-GRAFANA-BBB WIRELESS



SECUENCIA DE DATOS 1



```
snprintf(msg, 50, "%ld", WiFi.RSSI());  
client.publish("mote1/rssi", msg);
```



IP pública



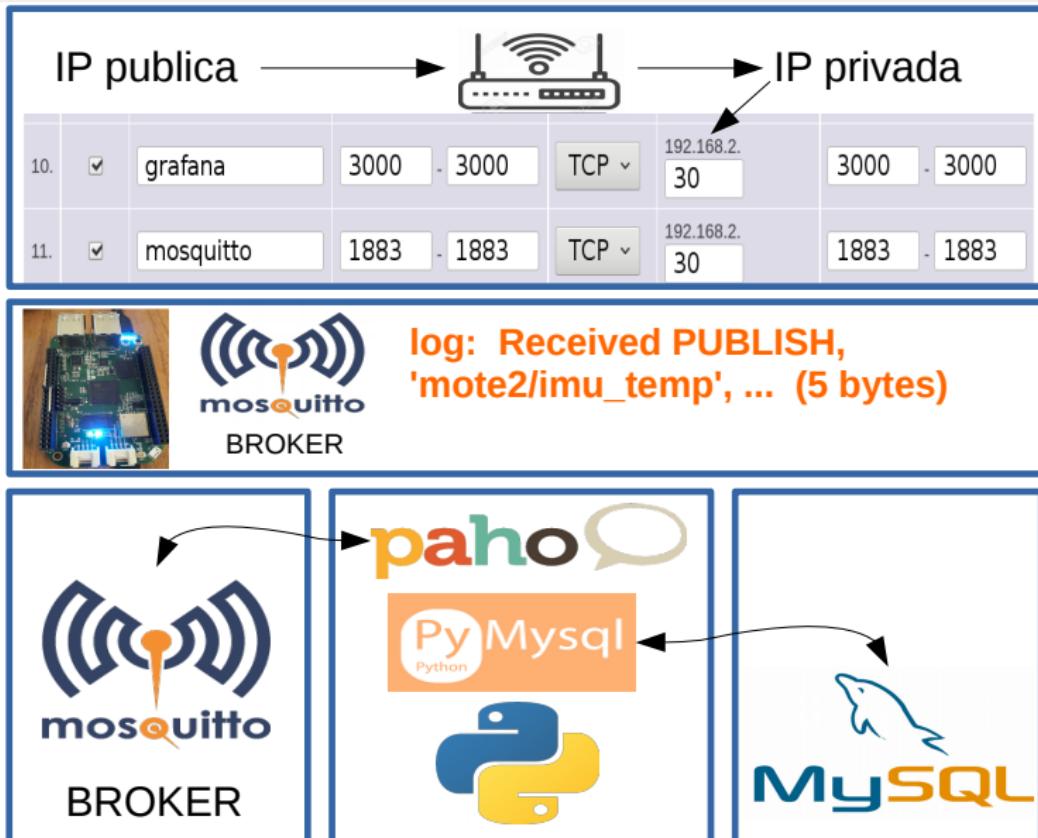
IP privada

10.	<input checked="" type="checkbox"/>	grafana	3000	-	3000	TCP	▼	192.168.2. 30	3000	-	3000
11.	<input checked="" type="checkbox"/>	mosquitto	1883	-	1883	TCP	▼	192.168.2. 30	1883	-	1883

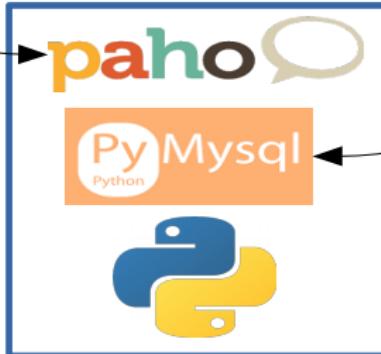


```
log: Received PUBLISH, 'mote2 imu_temp', ... (5 bytes)  
Topic= mote2 imu_temp msg= 29.47  
mysql table mote2_imu_temp value 29.47
```

SECUENCIA DE DATOS 2



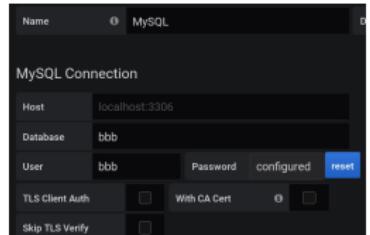
SECUENCIA DE DATOS 3



```
import pymysql  
import paho.mqtt.client  
  
def on_message(message):  
    mydb("INSERT INTO {} VALUES {}".format(message.topic,  
msg.payload)
```



FROM	mote2_imu_temp	Time
SELECT	Column: v	+
WHERE	Macro: \${_timeFilter}	+



SECUENCIA DE DATOS 4

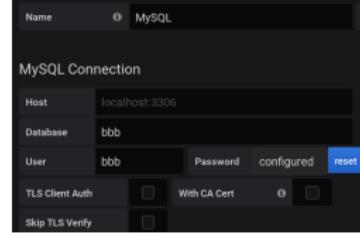


```
import pymysql
import paho.mqtt.client

def on_message(message):
    mydb("INSERT INTO {} VALUES {}" message.topic,
msg.payload)
```



SERVER



```
MariaDB [bbb]> show tables;
+-----+
| Tables_in_bbb |
+-----+
motel_alive
motel_rssi
mote2_alive
mote2_imu_temp
mote2_rssi
+-----+
5 rows in set (0.002 sec)

MariaDB [bbb]> describe motel_rssi;
+-----+
| Field | Type      | Null | Key | Default
+-----+
| v     | float     | NO   |     | NULL
| date  | timestamp | NO   |     | current_timestamp()
+-----+
2 rows in set (0.005 sec)
```

```
MariaDB [bbb]> show tables;
+-----+
| Tables_in_bbb |
+-----+
motel_alive
motel_rssi
mote2_alive
mote2_imu_temp
mote2_rssi
+-----+
5 rows in set (0.002 sec)
```

SECUENCIA DE DATOS 5



MySQL

```
MariaDB [bbb]> describe motel_rssi;
```

Field	Type	Null	Key	Default
v	float	NO		NULL
date	timestamp	NO		current_timestamp()

```
2 rows in set (0.005 sec)
```

```
MariaDB [bbb]> show tables;
```

Tables_in_bbb
motel_alive
motel_rssi
mote2_alive
mote2_imu_temp
mote2_rssi

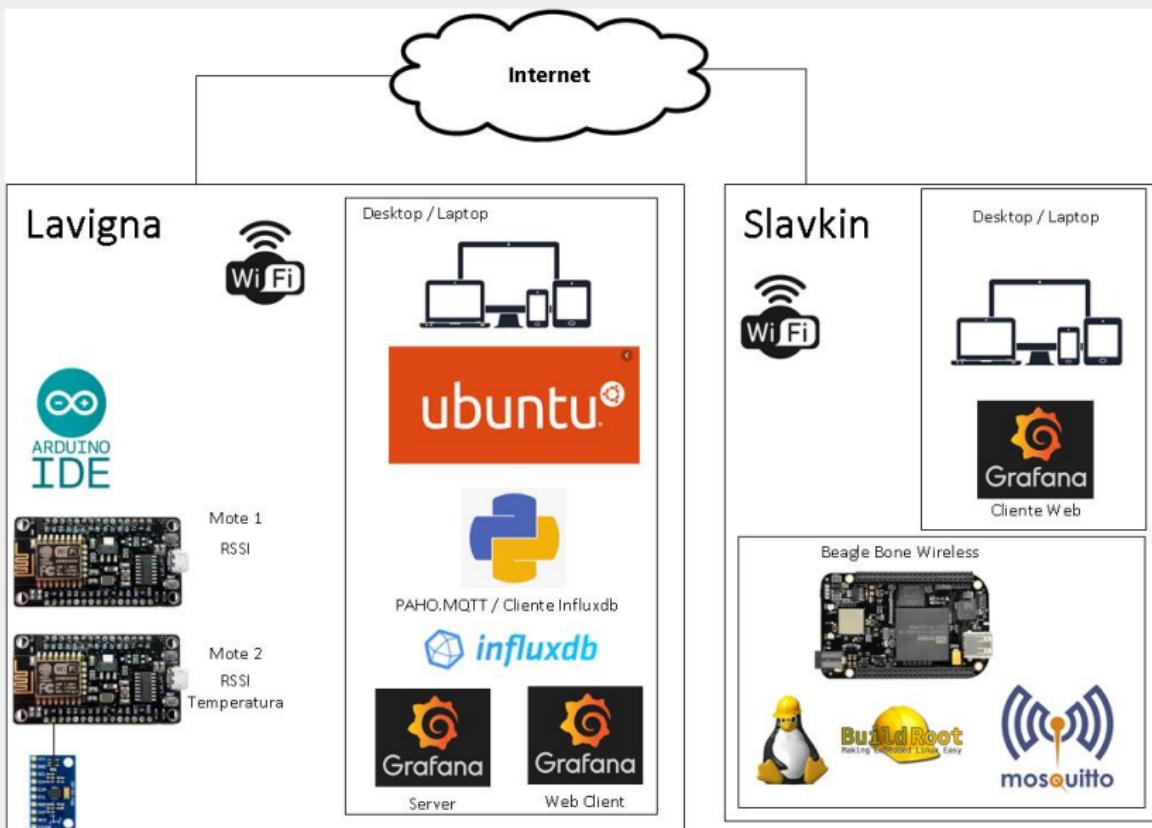
```
5 rows in set (0.002 sec)
```



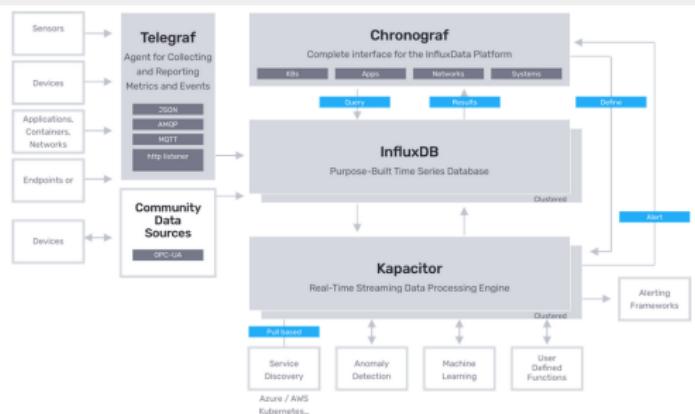
CLIENT



INFLUXDB-GRAFANA-DESKTOP



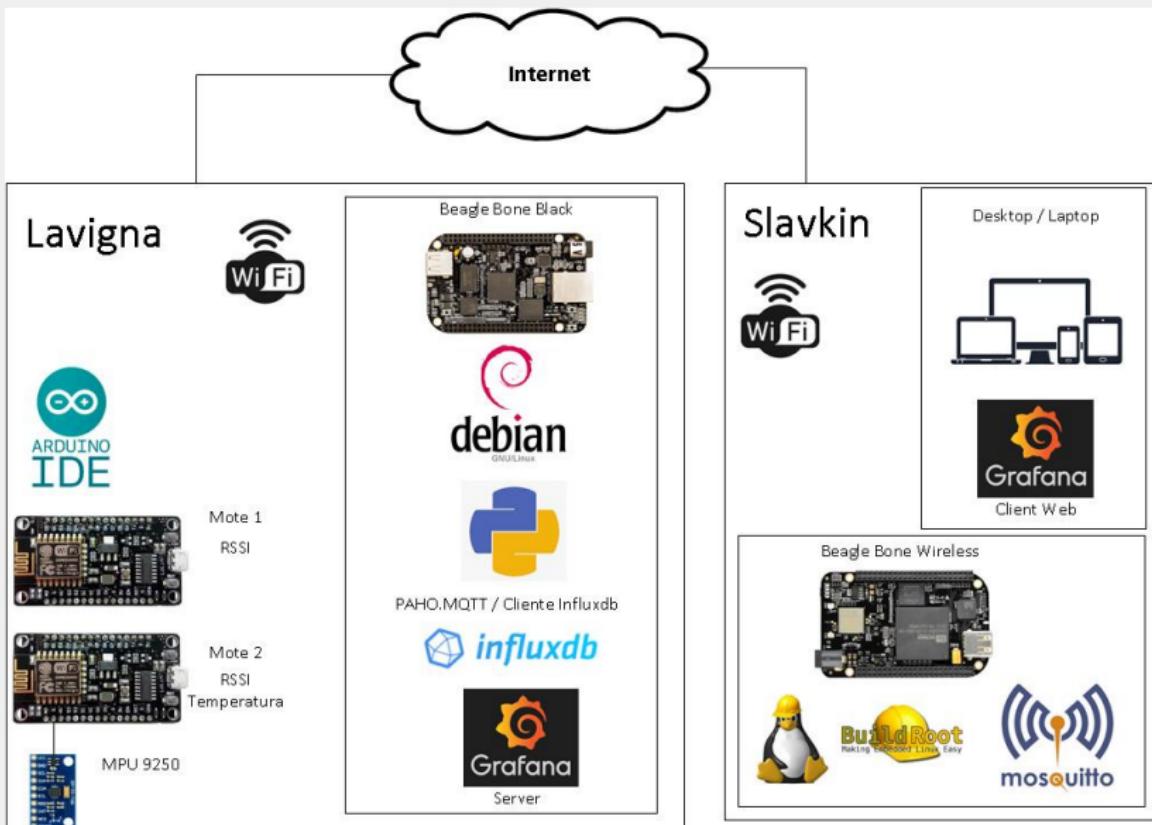
INFLUXDB-GRAFANA-DESKTOP



■ Características InfluxDB:

- ▶ Open source
- ▶ Diseñada para manejar métricas, eventos o mediciones que son time-stamped.
- ▶ Queries similares a SQL, especializadas para obtener rangos de tiempos.
- ▶ API via HTTP puerto 8086 o JSON.
- ▶ Es parte del entorno TICS.

INFLUXDB-GRAFANA-BBB



GRAFANA



GRAFANA



GRAFANA



CONCLUSIONES - MOTES

ESP:

- Económico y asequible
- Bibliotecas disponibles para MQTT
- Bajo consumo y dimensiones reducidas
- Fácilmente escalable
- Compatible con herramientas de Arduino
- Limitada potencia de calculo

Begale:

- Costo razonable
- Funcionamiento como mote/mqtt/webserver
- Herramientas de programación open y standard
- Herramientas de programación open
- Bajo consumo de CPU
- Simple y ágil

CONCLUSIONES - BASES DE DATOS

MySQL :

- Organización en tablas
- Posibilidad de lindeo de tablas
- Estable y configurable
- Consumo de CPU intensivo
- Poco apropiada para embebidos

influxDB:

- Organización por hash tags
- Bajo consumo de CPU
- Simple y ágil
- Configuración limitada
- Queries optimizadas para adquisición por slots temporales

TRABAJOS FUTUROS

- Investigar opciones con Telegraf para evitar scripts intermedios
- Investigar opciones con Cronograf como alternativa a Grafana
- Analizar el uso de Dockers
- Implementar aplicación para publicar alarmas por el broker.
Ej: temp1>10 mosquitto_pub -m apagar_calefactor

PREGUNTAS

