

# Memoria Práctica 2

## Comunicaciones

Leopoldo Cadavid Piñero

Febrero 2022

# Índice

1. Introducción	3
2. Sesión 1: Ubidots	3
3. Sesión 2: Introducción al entorno de Shelly	5
4. Sesión 3: Uso del protocolo HTTP Request	6
5. Sesión 4: Uso del MQTT	9
6. Ampliaciones finales	12

## 1. Introducción

En la siguiente práctica continuaremos con el aprendizaje distintos tipos de plataformas que nos permitan realizar aplicaciones del tipo IOT, así como el reforzamiento con herramientas como Nodered o la introducción a nuevos tipos de protocolos de comunicaciones como lo puede ser el MQTT.

## 2. Sesión 1: Ubidots

El objetivo en esta sesión es instalar y familiarizarse con la plataforma Ubidots. Ubidots es una plataforma de gestión de módulos enfocada para trabajar el campo de *Internet of Things*. En la práctica lo que haremos será loguear en la página y utilizar la API para trabajar desde Nodered.

Lo primero será registrarse en la página y crear lo que se llama en la plataforma "Devices". Obtenemos un token para comunicarnos con Nodered, y nos vamos a crear el siguiente Nodo:

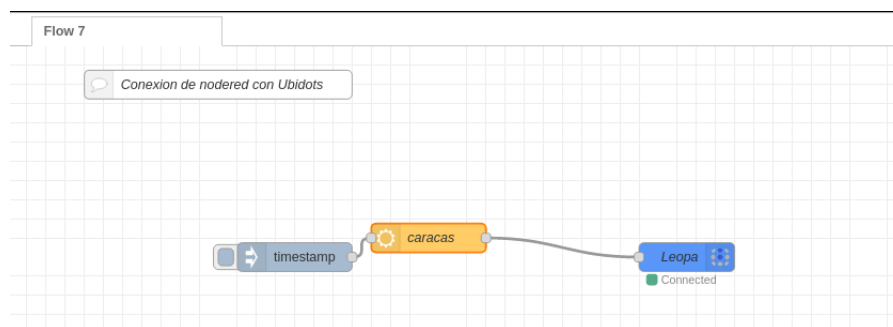


Figura 1: Nodo básico de Conexión entre Ubidots, Nodered y OpenWeather

Si vamos a Ubidots, veremos lo siguiente:

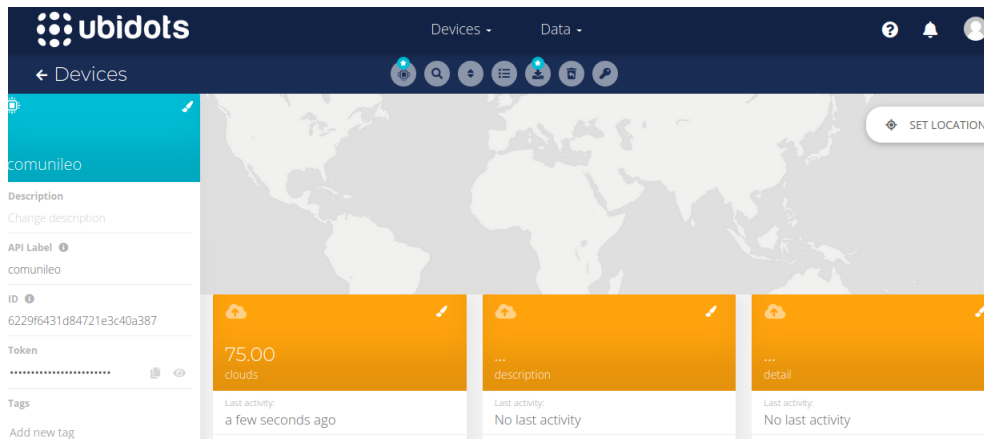


Figura 2: Resultado de la variable de Ubidots

En la anterior imagen vemos cómo en Ubidots hemos recibido varias de las variables que nos proporciona el nodo de OpenWeather.

Ahora, a modo de ampliación, lo que haremos es filtrar la información que enviamos desde Nodered, para que sólo reciba la variable temperatura, y además para que esta se encuentre en centígrados. Para ello debemos escribir una función que recoja y envíe específicamente ese parámetro en el formato necesario para que lo lea la API de Ubidots. El nodo sería el siguiente:

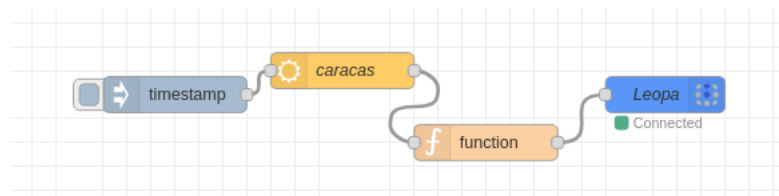


Figura 3: Nodo de ampliación de Ubidots

Finalmente, con el valor de la temperatura que recogemos en Ubidots, creamos un dashboard para ejemplificar el potencial de la plataforma en cuanto a gestión de información y automatización. El resultado sería el siguiente:

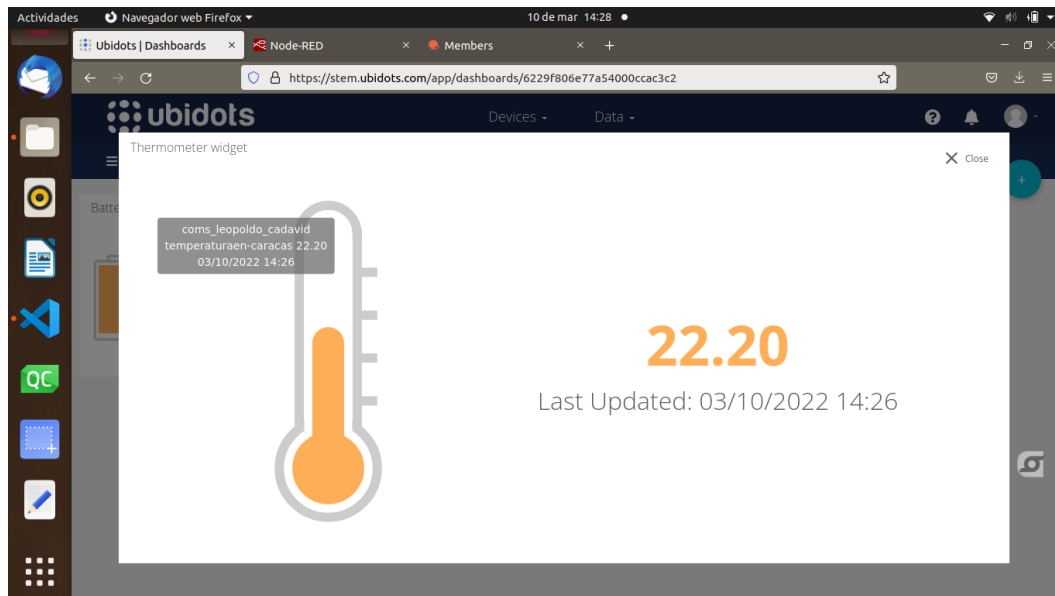


Figura 4: Nodo de ampliacion de Ubidots

### 3. Sesión 2: Introducción al entorno de Shelly

En esta sesión lo que haremos será iniciar el proceso de automatización de una sencilla instalación, usando protocolos de comunicación vistos en la teoría.

Lo que haremos en esta primera sesión será principalmente seguir los pasos que realiza el compañero para conectarse a la instalación.

Los pasos seguidos en clase han sido:

- Conectarse a la red del router de la instalación que es la subred de control
- Meterse en la interfaz de shelly
- Desde aquí podremos conectarnos a los dispositivos del sistema y probar como podemos dar instrucciones y recibir info
- **Aconsejable:** decirle al shelly a que red se tiene que conectar. Decimos a shelly que se conecte a la red que nosotros queremos (esto se graba en su memoria).
- Para nuestro caso: **Nombre de la red:** Cudy-081c, **IP:** 192.168.10.100 (rango x.x.10.(0/255)), **Gateway (opcional):**.

- Si hemos introducido todo bien, nuestro shell estará conectado en la red del router.
- Necesitamos saber las peticiones get para acceder a los componentes.
- En clase el compañero se ha conectado a la ip comentada y ha encendido y apagado el relé de la instalación.

No se llegaron a especificar más objetivos para esta sesión.

## 4. Sesión 3: Uso del protocolo HTTP Request

En esta sesión proseguiremos con el aprendizaje de protocolos sobre la instalación y para ello haremos uso de las peticiones por http, los bloques *HTTP Request* del Nodered.

El profesor nos da especificaciones y seguimos los siguientes pasos:

- **Red:** Cudy-081c , **Clave:** Comunica2022
- Nos conectamos a la IP: 192.168.10.100
- **Peticiones:** nos metemos en Nodered y con el nodo *http request* con el método **get** y url **192.168.10.110/relay/0?turn=off**. Con esto, nos conectaremos al relé de la estación de clase y podemos encender o apagar el relé según consideremos.

Primero, veamos el entorno de shelly al estar conectados a la red de Cudy:



Figura 5: Vista de la interfaz Shelly

Ahora veamos los nodos implementados en Nodered para realizar la conexión de este con el Shelly.

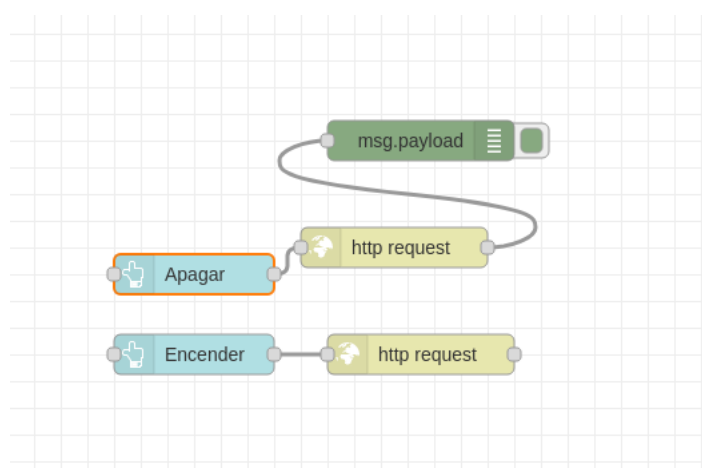


Figura 6: Nodos con protocolo HTTP request

Y el resultado en la UI de Nodered:

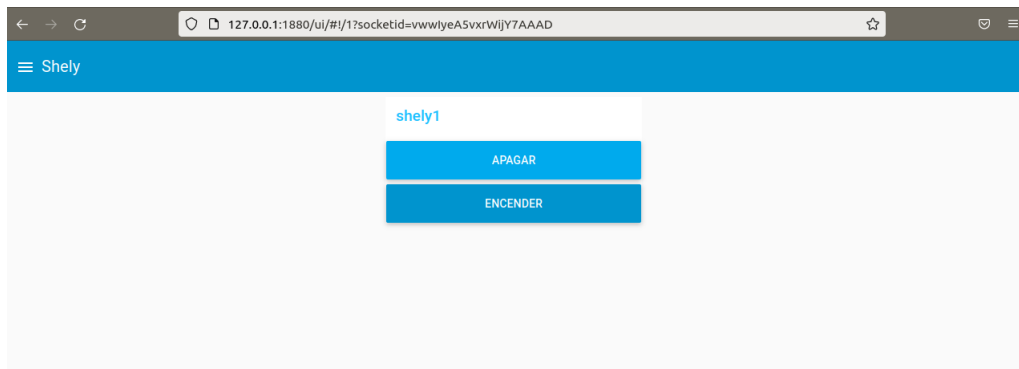


Figura 7: Interfaz de los botones

Adicionalmente lo que se ha hecho es introducir una interfaz para visualizar el estado de nuestro relé en tiempo real. Esto lo hacemos con un código que trate el mensaje que recibimos del estado del relé, que puede ser "On" u "Off". El nodo y los resultados son los siguientes:

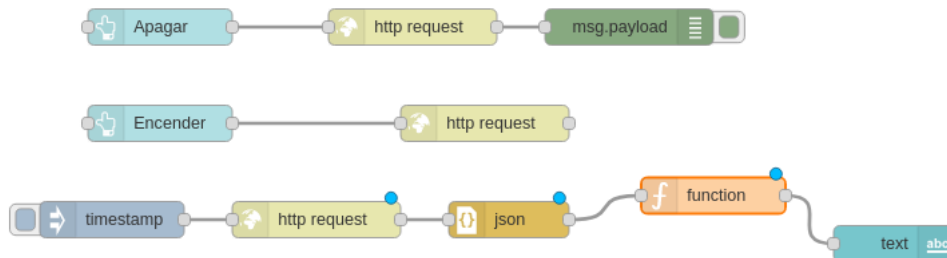


Figura 8: Botones y lectura del relé



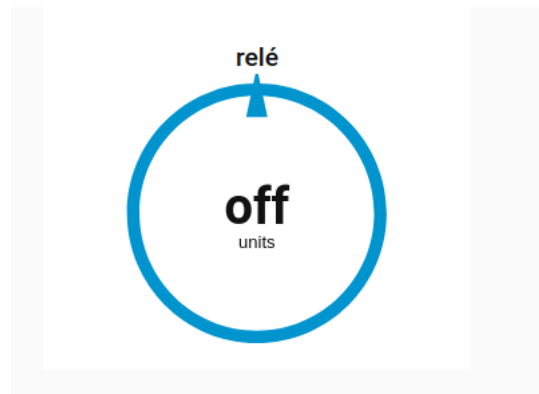


Figura 9: Visualización del estado del relé, en tiempo real

## 5. Sesión 4: Uso del MQTT

La idea de esta sesión es cambiar el uso de http requests, por el uso de nodos mqtt, para que la carga de las conexiones recaiga sobre el broker, en este caso la raspberry, en vez de en el shelly. Este protocolo resulta interesante porque permite un mayor tráfico de información sobre la instalación, dejando que un sistema más potente se encargue de publicar la información a otros suscriptores que la necesiten, pero sin que la propia Shelly deba saturarse por la carga de trabajo.

Algunos pasos seguidos en la sesión:

- creamos nodo mqtt
- Debemos configurar info del broker: Clave: comunicacionepass; IP: 192.16810.144
- configuramos el topic según los intereses

El siguiente paso de la sesión será ver como podemos ver el topic del shelly donde se estan publicando los datos que nos interesen. Veamos el flow de Nodered implementado para tal proposito:

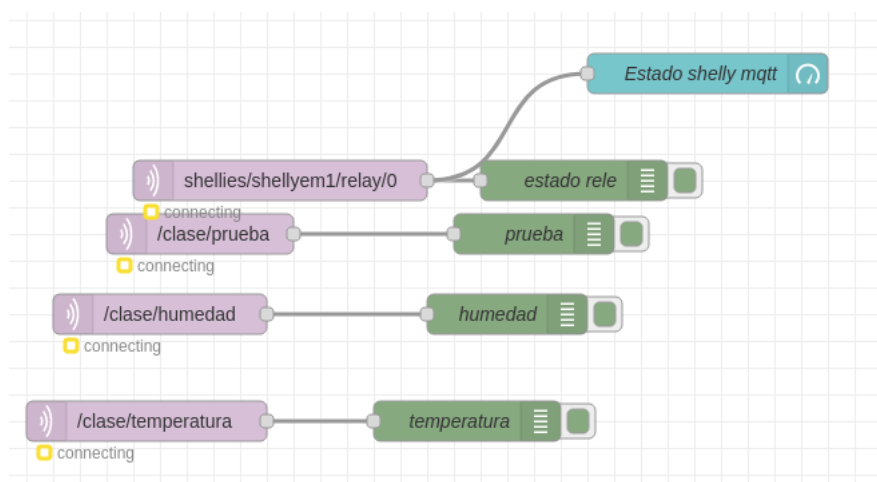


Figura 10: Shelly con protocolo MQTT

Vemos en la anterior imagen cómo estamos accediendo a los datos de: humedad, temperatura, o estado del relé de nuestra instalación, todo suscribiéndonos a los topics correspondientes. Algunas lecturas vistas desde el debugger:

```

shellies/shellyem1/relay/0 : msg.payload : string[4]
"on"

31/3/2022, 14:18:32 node: 95aa5f4300817c01
shellies/shellyem1/relay/0 : msg.payload : string[3]
"off"

31/3/2022, 14:18:37 node: 95aa5f4300817c01
shellies/shellyem1/relay/0 : msg.payload : string[13]
"1648729115752"

31/3/2022, 14:18:39 node: 95aa5f4300817c01
clase/humedad : msg.payload : string[5]
"75.6%"

31/3/2022, 14:18:39 node: 95aa5f4300817c01
clase/humedad : msg.payload : string[5]
"75.6%"

```

Figura 11: Resultados de la lectura de los topics

31/3/2022, 14:19:08 95aa5f4300817c01

/clase/humedad : msg.payload : string[4]

**"47.9"**

---

31/3/2022, 14:19:08 95aa5f4300817c01

/clase/temperatura : msg.payload : string[4]

**"19.0"**

Figura 12: Resultados de la lectura de los topics

A modo de añadido, se ha introducido la interfaz de encendido y apagado de botones que se utilizó en el apartado anterior, pero adaptado al protocolo MQTT.

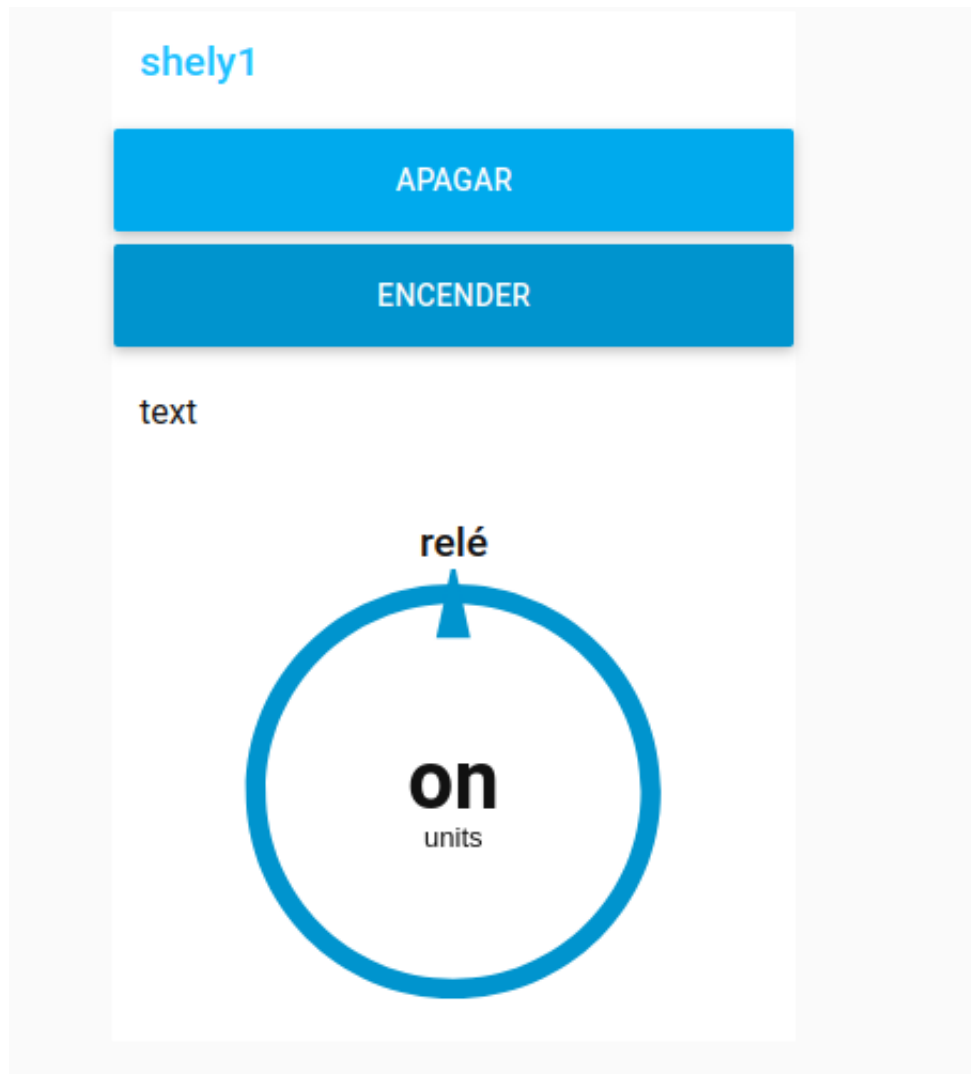


Figura 13: UI conectada a los topics MQTT

## 6. Ampliaciones finales

Finalmente, a modo de ampliación final, siguiendo con lo visto en la anterior práctica, se ha desarrollado un flow de Nodered, que se conecte a la Api de telegram para que podamos encender y apagar el relé desde la aplicación, además, podremos consultar la temperatura de la estación desde el móvil.

El flow desarrollado ha sido el siguiente:

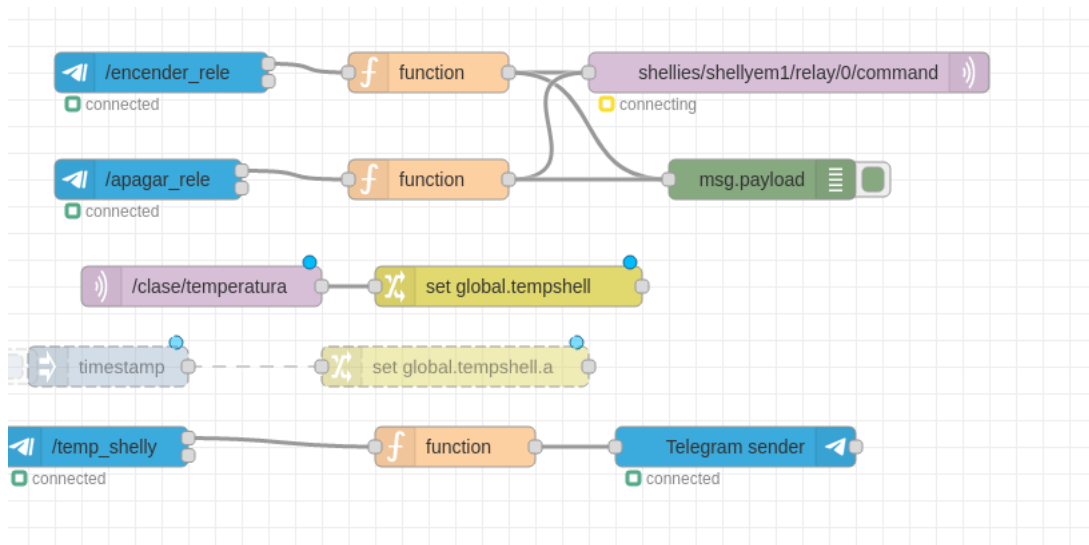


Figura 14: Ampliación usando API de Telegram en Nodered

Los comandos creados han sido `/encender_rele`, `/apagar_rele` y `/temp_shelly`. Para encender y apagar el relé, simplemente esperamos a recibir el comando, en cuyo caso, vamos al topic `"shellies/shellyem1/relay/0/command"` y publicamos el estado que corresponda usando `"On"` u `"Off"`.

Para la lectura de la temperatura, la estrategia es conectarse al topic de shelly que corresponde y recibir constantemente la información, la cual guardamos en una variable global a la que accederemos desde otro nodo, cuando se requiera publicar esta información. Así evitaremos estar publicando constantemente el valor de la temperatura en Telegram, pero nos aseguraremos de tener el valor más actual posible una vez haya sido pedido este.

Es importante mencionar que, aunque se ha implementado el nodo y realizado algunas pruebas, resulta físicamente imposible realizar una prueba con la instalación de la cual disponemos en el laboratorio, esto se debe a que, para poder ver lo que publica Shelly, debemos estar conectados al router del laboratorio, lo cual nos imposibilita estar conectados al wifi y poder realizar la conexión con Telegram. Para poder realizar algo parecido, deberíamos conseguir conectar el Shelly a internet, como indicaba el profesor en la sesión 2. Para ello, habría que inciar a Shelly una red para conectarse y esta quedará grabada en su memoria.

Ahora bien, realizando algunas emulaciones podemos ver si al menos la lógica por la cuál se ha implementado el flow esta bien encaminada. Por ejemplo,

si escribimos los comandos para encender y apagar el relé en Telegram, y posteriormente visualizamos mediante el debugger lo que se publicaría al topic vemos lo siguiente:

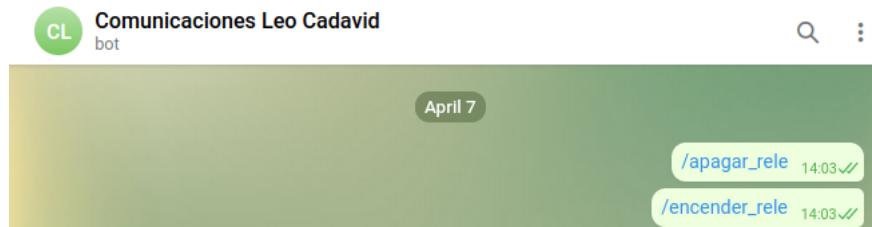


Figura 15: Comandos dados al bot en Telegram



Figura 16: información que se publica al topic que enciende y apaga el relé

Como observamos, en caso de encontrarse conectado a la vez a Shelly y a Telegram, los mensajes correctos se estarían publicando al topic, permitiéndonos encender y apagar el relé desde la aplicación.

En el caso de revisar la temperatura, podemos sustituir el mensaje a recibir del topic, por un timestamp con un número prefijado a modo de simulación de la temperatura. Si hacemos esto, y ejecutamos el comando en Telegram podemos ver lo siguiente:

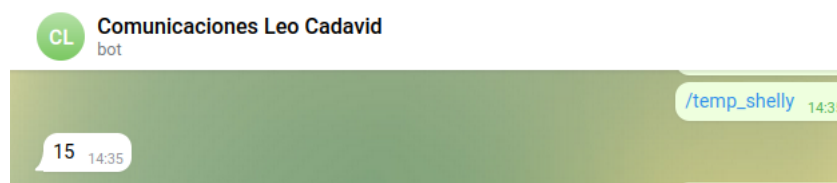


Figura 17: Temperatura simulada de la instalación

Vemos que si llegamos a recibir la información, y en caso de estar conectados a Shelly, solo haría falta cambiar el timestamp por un nodo *mqtt in*.