*PROYECTO INFORMÁTICA INDUSTRIAL*

**INSTALACIÓN DOMÓTICA CON**

**ESP 8266**

**Leonardo García Guillen**

**Juan José García Quiles**

**Javier López Soria**

**Frank Leonardo Luque Lineros**

**Grado en Ingeniería Electrónica, Robótica y Mecatrónica**

**5 de Enero de 2021**

Contenido

[1. INTRODUCCIÓN Y OBJETIVOS 3](#_Toc61196990)

[2. DISEÑO HARDWARE 3](#_Toc61196991)

[3. DISEÑO SOFTWARE 5](#_Toc61196992)

[3.1. NodeRed 5](#_Toc61196993)

[3.1.1. Introducción 5](#_Toc61196994)

[3.1.2. Nodos MQTT 5](#_Toc61196995)

[3.1.3. Nodos Telegram 5](#_Toc61196996)

[3.1.4. DashBoard 5](#_Toc61196997)

[3.2. Arduino 5](#_Toc61196998)

[3.2.1. Introducción 5](#_Toc61196999)

[3.2.2. Estructura 5](#_Toc61197000)

[4. RESULTADO Y CONCLUSIONES 5](#_Toc61197001)

[5. MANUAL DE USUARIO 5](#_Toc61197002)

[6. LISTA DE FICHEROS 5](#_Toc61197003)

# INTRODUCCIÓN Y OBJETIVOS

En el siglo XXI se ha desarrollado la robótica a una velocidad increíble, mucho más rápido de lo que la gente de hace unos años podría haber imaginado. hasta hace relativamente poco tener una casa “inteligente” era algo perteneciente al género de ficción pero en pleno 2021 la domótica es algo muy común y sencillo de aplicar en cualquier casa.

Se va a utilizar las herramientas y programas que hemos aprendido durante el curso para configurar una casa con diversas funcionalidades domóticas. Se va a hacer uso de NodeRed para crear flujos de datos, donde recogeremos datos de una base de datos y enviaremos y recogeremos información por MQTT y también para representar los datos mediante un “DashBoard”. Para la base de datos vamos a usar MongoDB y para la programación de la ESP8266 se utilizará Arduino.

Nuestro objetivo es crear un bot de Telegram que nos permita a través de comandos sencillos y claros poder controlar y recibir información de varias habitaciones de nuestra casa. Trataremos de regular la luz de las habitaciones, pudiendo elegir el nivel de intensidad; Podremos ver el estado de las luces, la humedad, la temperatura; También podremos consultar la predicción del tiempo y cambiar el tiempo de recogida de datos.

# DISEÑO HARDWARE

El hardware donde vamos a meter el programa Arduino es una placa ESP8266, Este es un chip “low-cost” con Wi-Fi. Tiene 16 pines GPIO, 1 conversor ADC de 10 bits y un pin GPIO2 dedicado a UART. Podemos ver la configuración de pines en la figura 1.

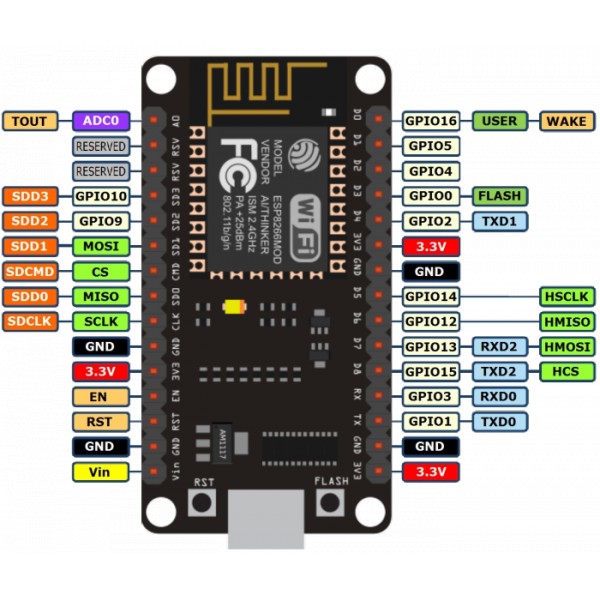


Figura 1. ESP8266

Los CHIP ID de cada ESP8266 y en que habitación se encuentra:

- EL ESP145 corresponde a la cocina.

- EL ESP47 corresponde al salón.

- EL ES1122489 corresponde al baño.

- EL ESP230 corresponde a la habitación.

además de eso vamos a tener distintos sensores que son; el DHT-11, MQ-2 y el VL53L0X.

El sensor DHT-11 está conectado en todas las ESP y lo conectamos en el D1 la entrada digital y lo alimentamos con 3.3V y también lo conectamos a la tierra de la placa. Este sensor lo que hace es coger la temperatura (ºC) y la humedad (%). En la figura 2 vemos el sensor DHT-11.

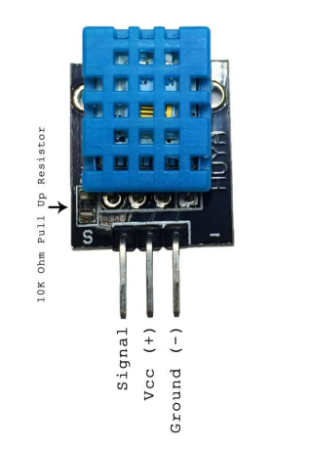


Figura 2. Sensor DHT-11

El sensor MQ-2 solo está conectado en las placas de la cocina, salón y baño y lo conectamos a la única entrada analógica que es la A0. El sensor lo conectamos por la salida analógica y lo alimentamos con 3.3V y lo metemos a la tierra de la placa. Aquí nos da los valores del humo, CO y LPG, todo en ppm.



Figura 3. Sensor MQ-2

El sensor VL53L0X solo está conectado en la placa del salón y lo conectamos a la placa. Podemos verlo en la figura 4. Lo vamos a utilizar para comprobar si una puerta está abierta, observando cuando se reduce la distancia del infrarrojo. Lo vamos a utilizar para comprobar el estado de una puerta (si está abierta o cerrada), observando si la distancia medida supera un umbral específico. El modo de conexión del sensor con la ESP8266 es muy sencillo; el pin Vcc se debe conectar a una fuente de tensión de 5V, el problema es que el valor de tensión máximo que alcanza un pin de la ESP es de 3.3V (razón por la cual a veces se dan problemas de inicialización). Otro de los pines que se deben conectar es el GND, uniéndolo a cualquier pin GND de nuestra ESP. Finalmente, los pines SCL y SDA se han conectado a los correspondientes de la placa, estos son, los GPIO5 y GPIO4 respectivamente.

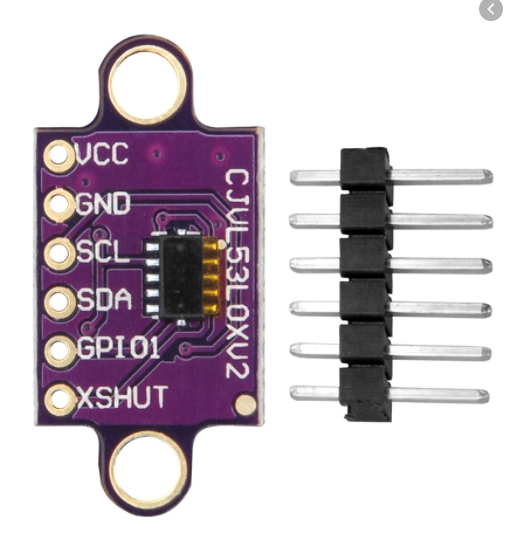


Figura 4. Sensor VL53L0X

# DISEÑO SOFTWARE

## NodeRed

### Introducción

En este trabajo se ha utilizado NodeRed, una herramienta de programación basada en flujos y nodos. En NodeRed vamos a configurar la comuniación entre los diferentes programas que vamos a usar: (Telegram,arduino,MongoDB(base de datos)).

La mayoría de flujos tienen una estructura parecida: En primer lugar un nodo de telegram para recibir el comando, después una serie de nodos internos de nodered que procesan la información, normalmente sacada de la base de datos que esta a su vez lo ha recibido por MQTT y finalmente la envía por telegram.

### Recogida de datos

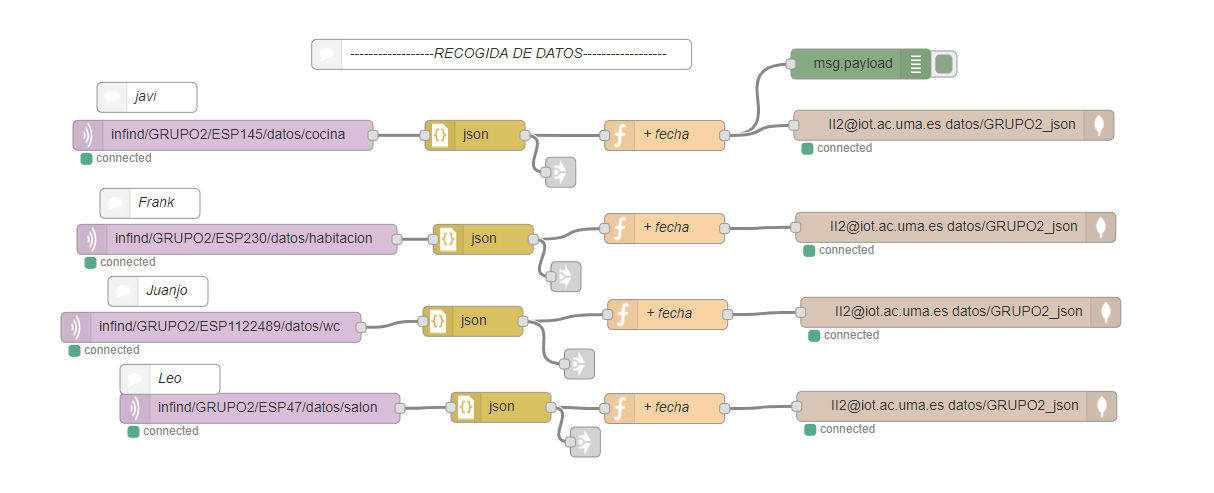
La comunicación se ha realizado por MQTT para conectar las diferentes ESP8266. Hemos recogido datos con arduino y lo hemos **publicado** en un topic. Estos datos se han recogido por NodeRed, que tiene nodos que se **suscriben** a los topics anteriores. La información que se recibe por MQTT está formateada en JSON y se tendrá que convertir a “string” para trabajar con ella. Cada habitación cuenta con un topic por donde se ha enviado la información, a esta le hemos añadido la fecha y se ha guardado en la base de datos. Ver figura 5. Los datos que llegan se formatean en JSON y posteriormente se le añade la fecha. Estos datos se almacenan finalmente en mongoDB.

Figura 5. NodeRed recogida de datos

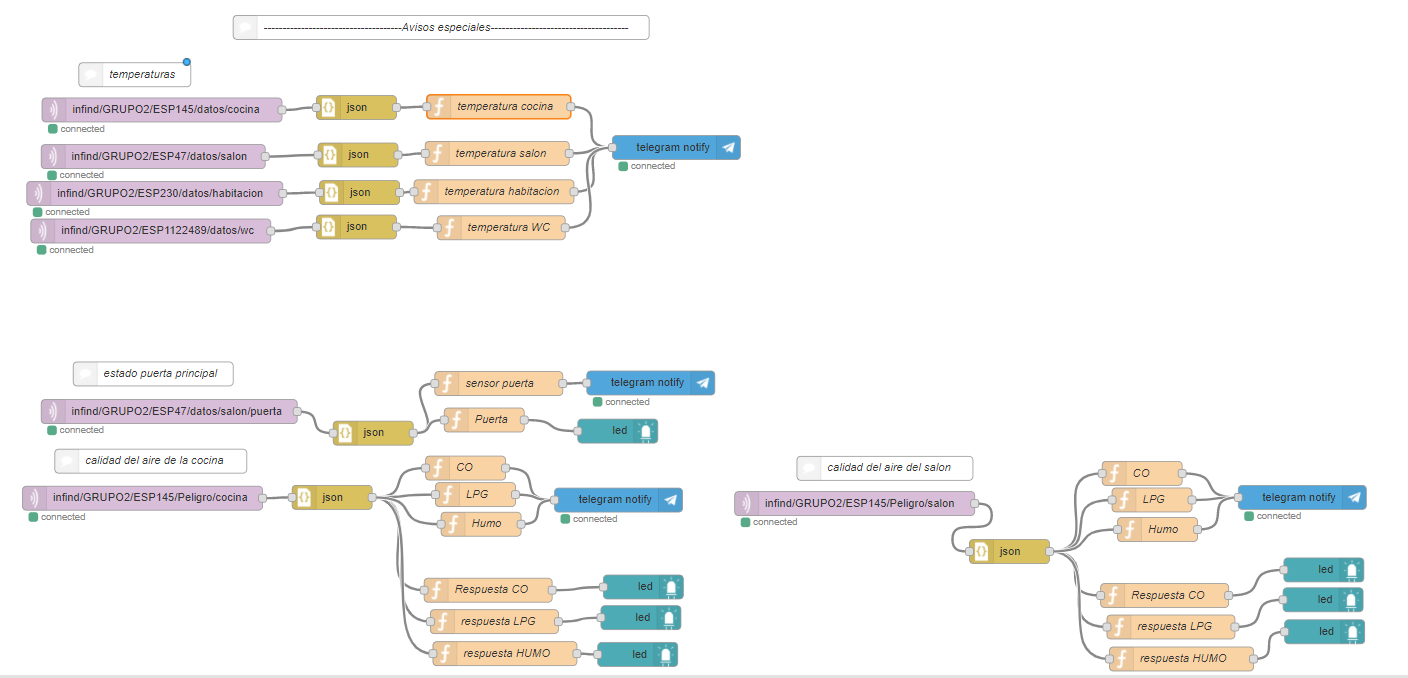
Tenemos otros topics que recogen datos adicionales, como los gases contaminantes que hay en la cocina, como vemos en la figura 6. Cada flujo de temperatura tiene una función cuyo objetivo es detectar la temperatura máxima asignada por una variable global que podemos modificar. En el caso del detector de la puerta recibe la distancia que hay de una pared a otra y si es mas corta que cierto valor entonces detectamos “Puerta abierta”. En el caso de los gases recibimos niveles de peligro (0,1,2) y el nodeRed avisa cada vez que aumenta este nivel.

Figura 6. NodeRed Avisos especiales

### Envió del nivel de luminosidad del LED

Cuando usamos telegram para cambiar la intensidad del LED tenemos que enviar por MQTT el nivel elegido, para que la ESP8266 la recoja y actualice el valor de la intensidad. Enviamos estos datos por el topic infind/GRUPO2/ESPXX/led/cmd/YY, cambiando XX por el ChipId y la YY por la habitación que queramos. Podemos observarlo en la figura 7.

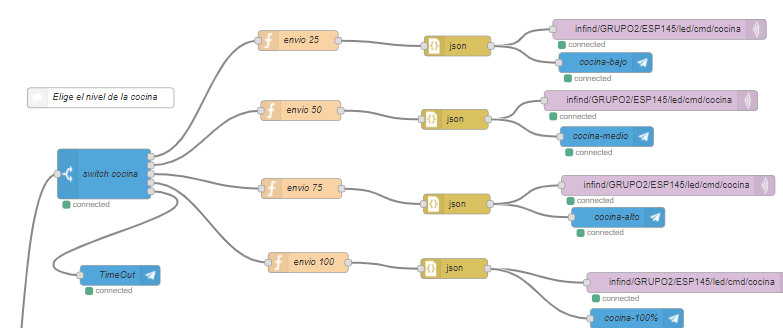
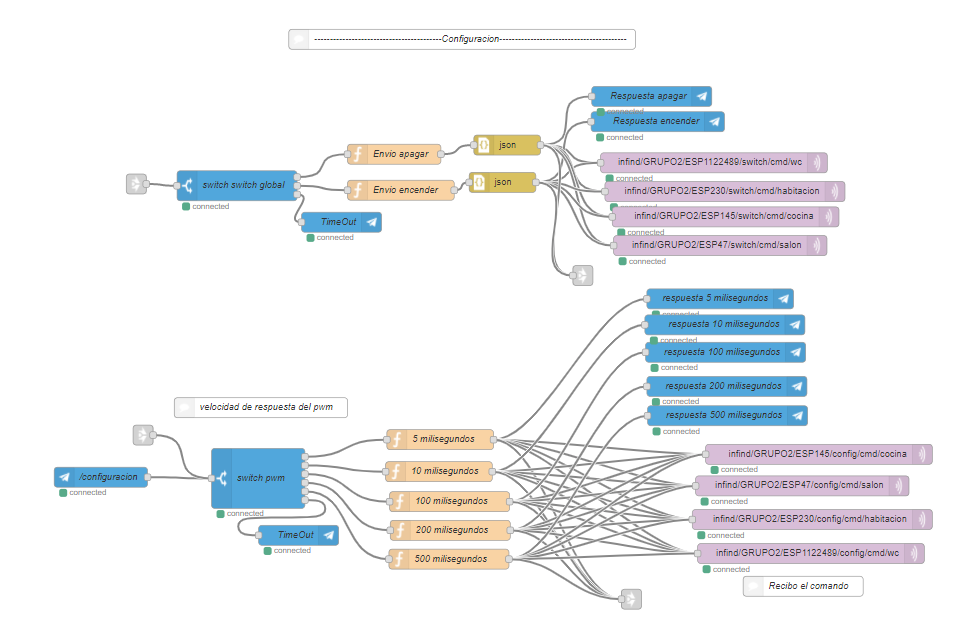
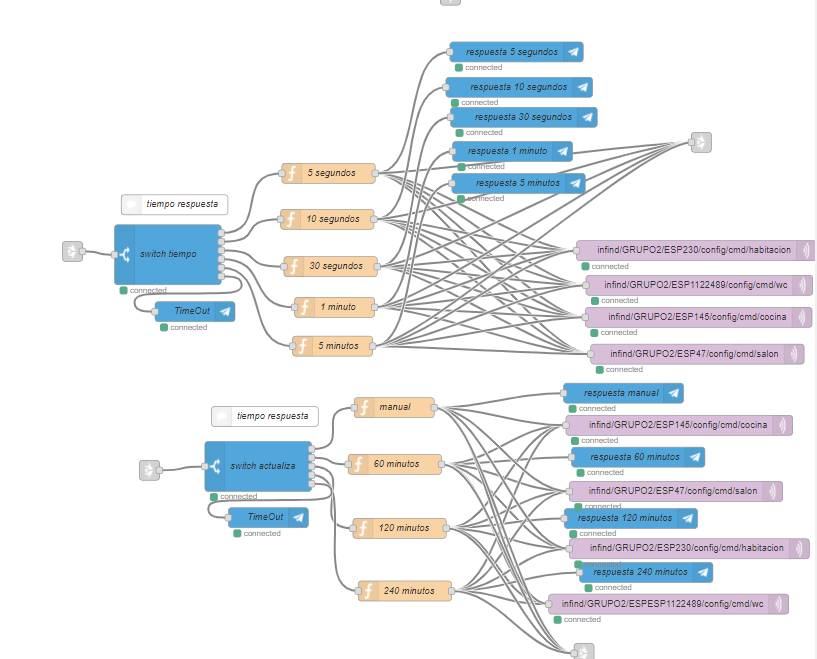


Figura 7. Envio nivel LED

Tenemos una estructura para cada habitación, pero es exactamente igual.

También contamos con un topic de configuración, que recoge varios valores y los lleva a “infind/GRUPO2/ESPXX/config/cmd/YY”. Los datos que se recogen son la velocidad de PWM, el tiempo de recogida de datos, el tiempo de actualizacion, el estado al que queremos el switch y el estado que queremos el led. Todos estos parámetro se podrán ajustar por telegram. En la figura 8 podemos ver este nodo MQTT.





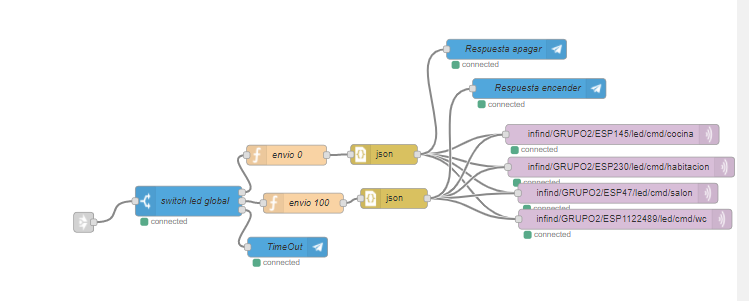


Figura 8. MQTT de configuracion

### Actualización FOTA y predicción

Para poder realizar actualizaciones FOTA se ha creado un flujo cuyo resultado se envía al topic que se ve en la figura 9.

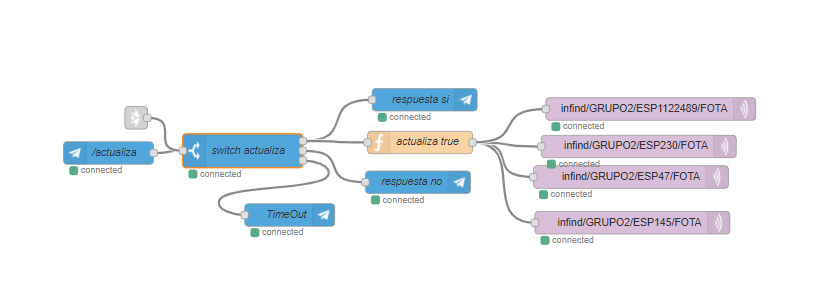


Figura 9. Flujo FOTA

Otra función extra que tiene nuestro bot es la de consultar el tiempo atmosférico de hoy, mañana o pasado. Este flujo consulta al AEMET y posteriormente construye una respuesta en JSON que será enviada por Telegram. Ver figura 10

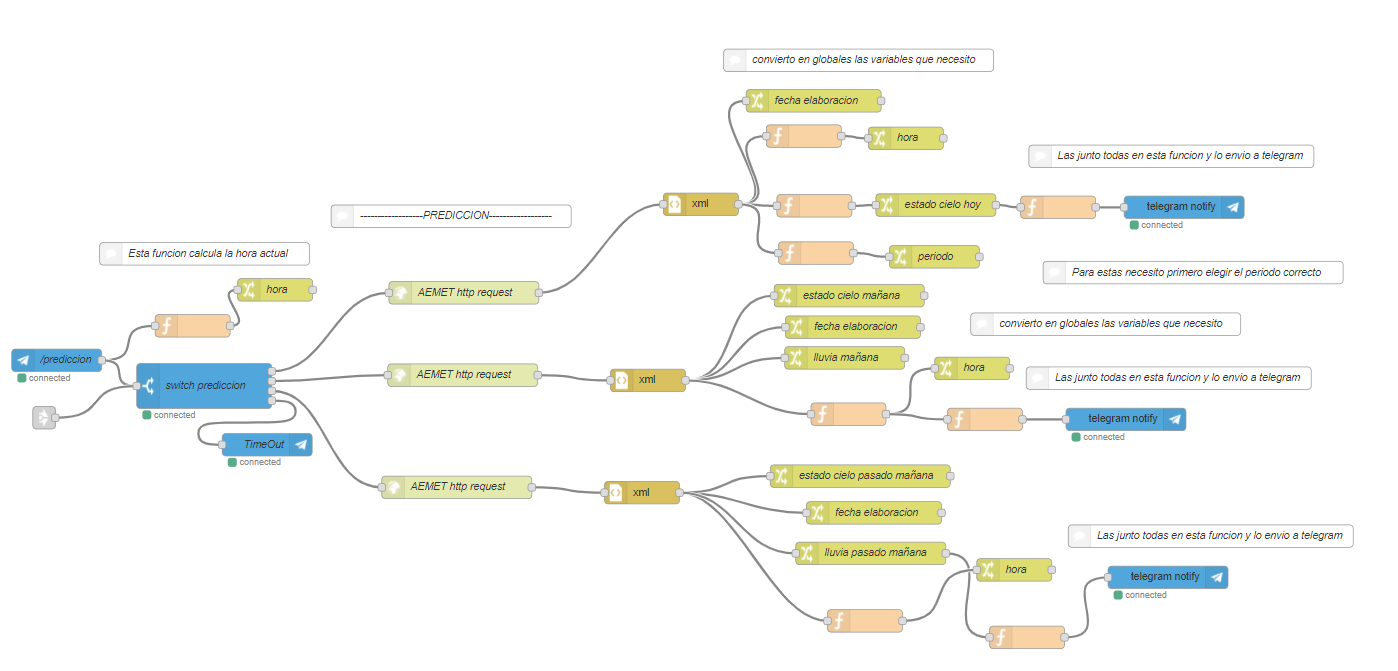


Figura 10. Prediccion tiempo

### Consultar estado

La función que más se usará de este bot será la de consultar estado. Este flujo genera una consulta a la base de datos para posteriormente sacar la información deseada. Posteriorment una función tomará los datos y los preparará para presentarlos en telegram.

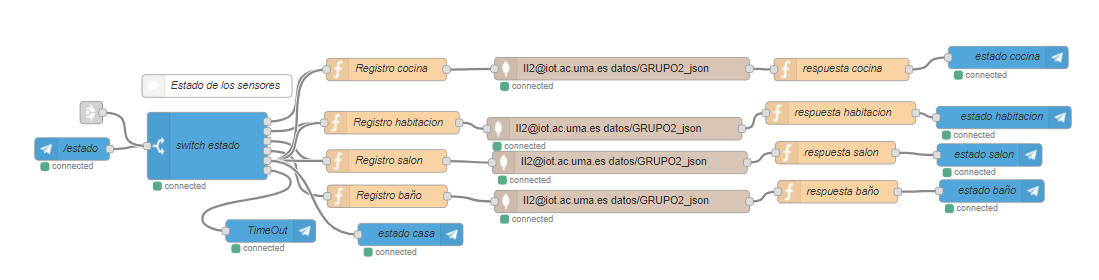


Figura 11. Flujo de consulta de estado

El formato de la consulta tendrá el aspecto de la figura 12.

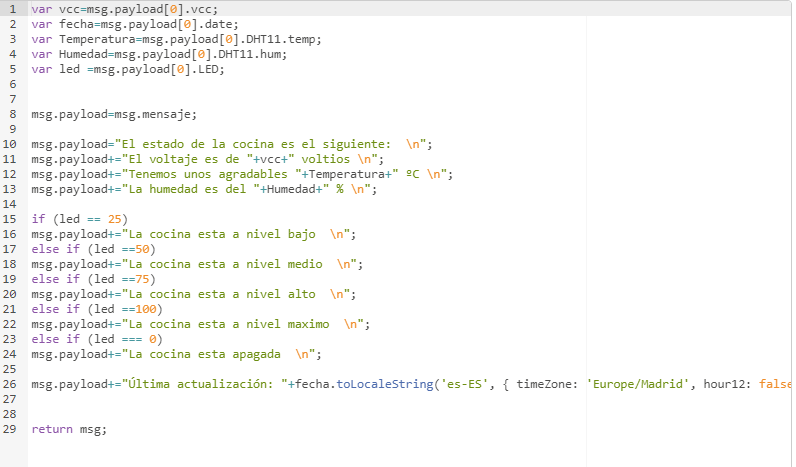


Figura 12. Formato /consulta

### DashBoard

## Arduino

### Introducción

### Estructura

## Telegram

### Introducción

La herramienta que vamos a usar para hacer las consultas y las modificaciones es Telegram. El objetivo es hacer una interfaz fácil de utilizar y entender para poder usar la aplicación sin necesidad de conocer el código que hay detrás. Para hacer esto se han usado los nodos de Telegram que ofrece NodeRed. En el punto 3.1 ya vimos estos elementos de telegram, en este punto veremos en detalle cómo funciona.

### Funcionamiento

Lo primero que tenemos que hacer es enviar un mensaje de bienvenida, como vemos en la figura 13. Al escribir /start recibimos un mensaje que nos indica las multiples opciones que tenemos, también tenemos la opción de ver un mensaje tutorial que nos indicará cómo usar cada comando.

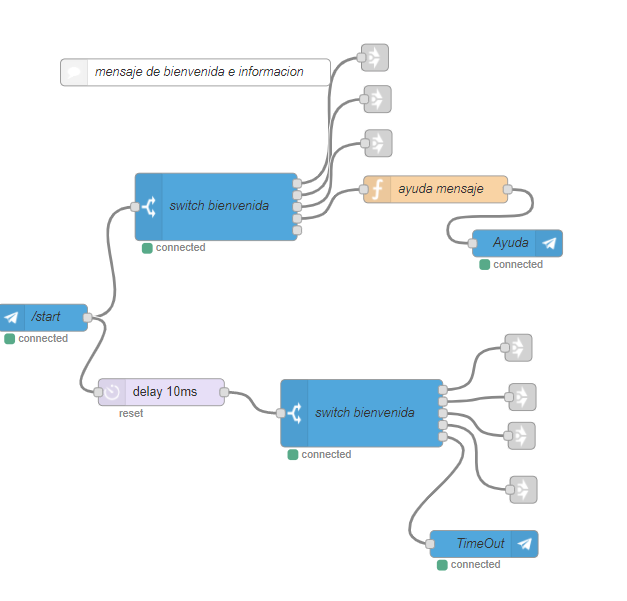


Figura 13. Bienvenida telegram

En la figura 14 podemos ver cómo lo veriamos en telegram. Lo primero que haremos si es la primera vez que se utiliza el bot es ver el mensaje tutorial.

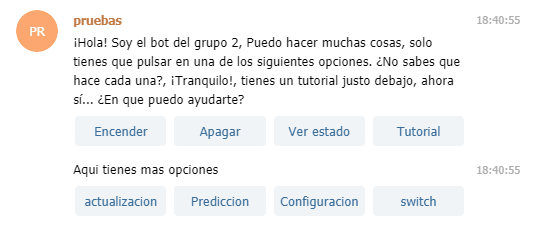


Figura 14. Bienvenida (telegram)

El mensaje tutorial nos resume todos los comandos que hay y que hace cada uno. En la figura 15 vemos el texto que aparece al clicar en “Tutorial “.

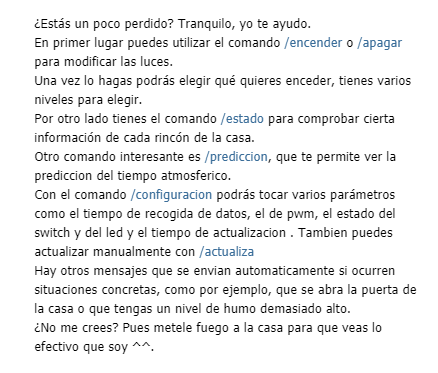


Figura 15. Tutorial telegram

Los comandos encender/apagar y ver estado ya se explicaron antes, en el caso del estado el resultado se vería tal y como vemos en la figura 16. Hemos clicado en “todo” para ver todos los estados junto con su fecha de adquisición.

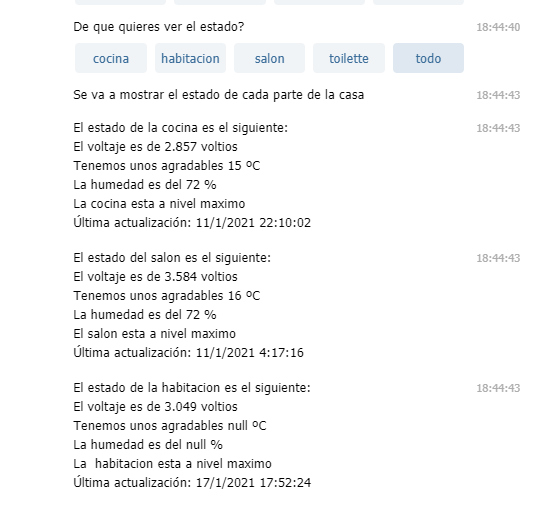


Figura 16. Estado telegram

Para el flujo de predicción veriamos algo como en la figura 17.

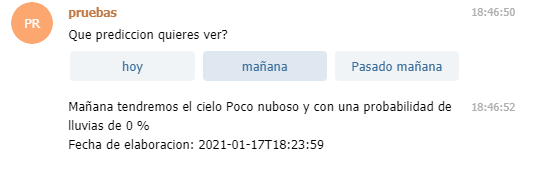


Figura 17. Prediccion telegram

Si usamos el comando /configuracion saldrán una serie de preguntas que iremos contestando con los datos que queremos. Entre ellas está la posibilidad de elegir actualización manual.

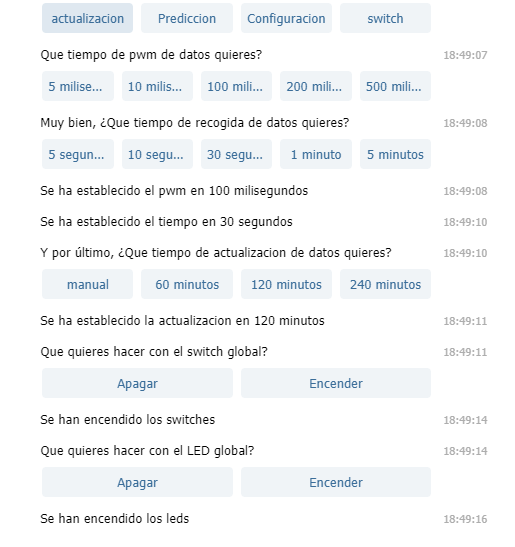


Figura 18. configuracion telegram

# RESULTADO Y CONCLUSIONES

# MANUAL DE USUARIO

# LISTA DE FICHEROS