



UNIVERSIDAD DE MÁLAGA

# INSTALACIÓN DOMÓTICA CON ESP8266

PROYECTO FINAL DE INFORMÁTICA INDUSTRIAL

GRADO EN INGENIERÍA ELECTRÓNICA, ROBÓTICA Y MECATRÓNICA

Autores: Leonardo García Guillén  
Juan José García Quiles  
Javier López Soria  
Frank Leonardo Luque Lineros

Tutores: Dr. Andrés Rodríguez  
D. Rubén Delgado Escaño

## TABLA DE CONTENIDO

1. INTRODUCCIÓN Y OBJETIVOS .....	1
2. DISEÑO DE HARDWARE.....	1
2.1. Esquemas de conexión.....	4
3. DISEÑO DE SOFTWARE.....	5
3.1. Librerías y topics.....	5
3.1.1. Librerías.....	5
3.1.2. Flujo de programa .....	6
3.1.3. Topics .....	7
3.2. Node-RED .....	8
3.2.1. Introducción .....	8
3.2.2. Recogida de datos .....	8
3.2.3. Envío de nivel de luminosidad del led.....	9
3.2.4. Actualización FOTA y predicción .....	11
3.2.5. Consultar estado .....	13
3.2.6. Nodos para ajustar variables globales .....	14
3.2.7. Servicios adicionales.....	15
3.2.8. Dashboard .....	16
3.2.9. Google Home.....	20
3.3. Arduino.....	23
3.3.1. Introducción .....	23
3.3.2. Inicialización y declaración de estructuras.....	24
3.3.3. Interrupciones .....	24
3.3.4. Funciones para la serialización de datos JSON.....	24
3.3.5. Conexión.....	24
3.3.6. Función de recepción de datos por MQTT .....	24
3.3.7. Setup y OTA.....	25
3.3.8. Función loop.....	25
3.4. Telegram.....	25
3.4.1. Introducción .....	25

3.4.2.	Funcionamiento .....	25
3.5.	Robustez.....	29
3.5.1.	Topics .....	29
3.5.2.	Inicializaciones.....	30
3.5.3.	Sensores .....	30
3.5.4.	Anti-rebotes .....	31
3.6.	Google Home.....	31
3.6.1.	Introducción .....	31
3.6.2.	Dispositivos con Nora .....	31
4.	MANUAL DE USUARIO.....	35
4.1.	Dashboard del Node-RED.....	35
4.1.1.	¿Cómo puedo ver los datos actuales?.....	35
4.1.2.	¿Cómo puedo seleccionar el área de la vivienda que quiero controlar? .....	38
4.1.3.	¿Cómo puedo cambiar los valores máximos y mínimos de la alarma? .....	40
4.1.4.	¿Cómo puedo ver un registro histórico de los datos recogidos? .....	42
4.1.5.	¿Cómo puedo descargar los datos recuperados por una consulta en formato CSV?.....	43
4.2.	Conexión con el asistente de Google .....	43
4.2.1.	¿Qué dispositivos puedo tener?.....	43
4.2.2.	¿De dónde son los dispositivos? .....	44
4.2.3.	¿Cómo miro los datos de los dispositivos? .....	44
4.2.4.	¿Qué significa cada cosa en los dispositivos y como le doy comandos? .....	45
4.2.5.	¿El Google Home puede darme datos?.....	48
4.2.6.	¿Puedo cambiar los valores de los dispositivos? .....	48
4.3.	Bot de Telegram .....	49
4.3.1.	¿Cómo puedo tener acceso al bot de Telegram?.....	49
4.3.2.	¿Qué comandos puedo usar con el bot?.....	50
4.3.3.	¿Cómo puedo ver información de la vivienda desde el bot de Telegram?.....	50
4.3.4.	¿Qué tipo de configuración puedo hacer con el bot de Telegram?.....	52
4.3.5.	¿Qué puedo controlar desde el bot de Telegram? .....	54
5.	RESULTADOS Y CONCLUSIONES .....	57
6.	LISTA DE FICHEROS.....	57

## 1. 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 Node-RED 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.

## 2. DISEÑO DE 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.

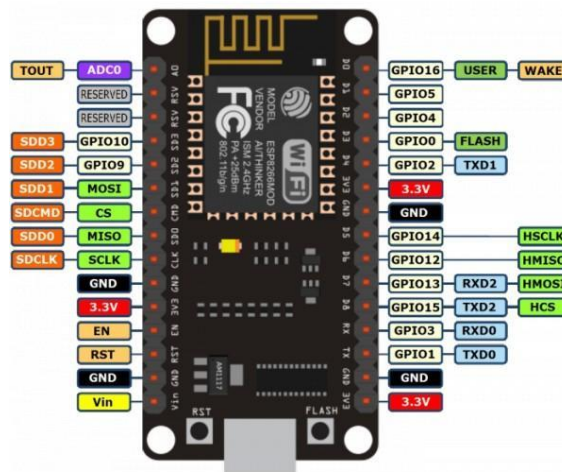


Figura 1. ESP8266

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

- El ESP5826449 corresponde a la cocina.
- El ESP5821487 corresponde al salón.
- El ESP1122489 corresponde al baño.
- El ESP8789990 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 (°C) y la humedad (%). En la Figura 2 vemos el sensor DHT-11.

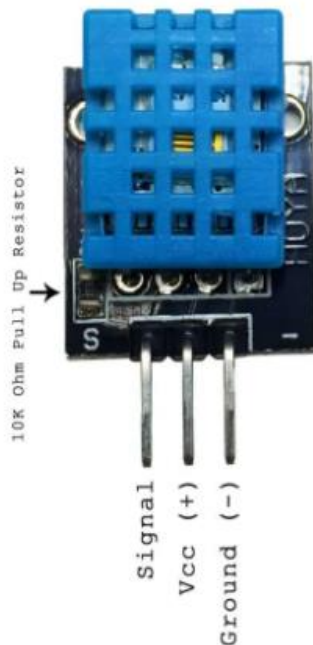


Figura 2. Sensor DHT11

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. Véase Figura 3.



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.

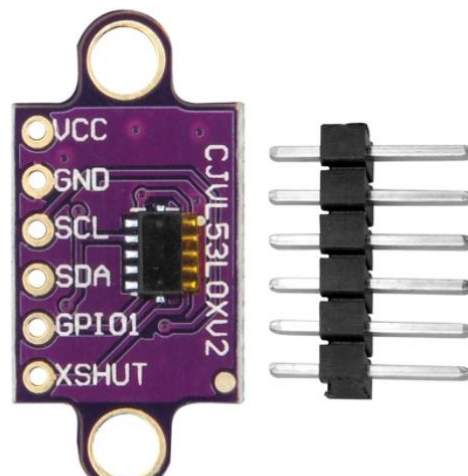


Figura 4. Sensor VL53L0X

## 2.1. Esquemas de conexión

En esta sección se contemplará el interconectado hardware entre la ESP8266 con los distintos sensores de los que disponemos. Para todas las ESP se ha tenido en cuenta el mismo conexionado, evitando así cambiar las inicializaciones de pines dentro del código Arduino. Los pines vienen identificados y descritos en el apartado anterior. Tenemos los conexionados en las Figuras 5,6 y 7.

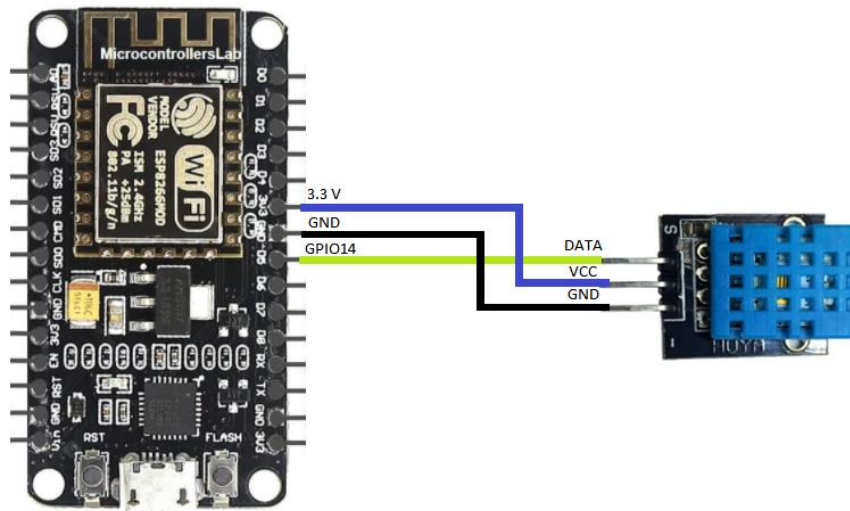


Figura 6. Conexionado ESP8266 - DHT11

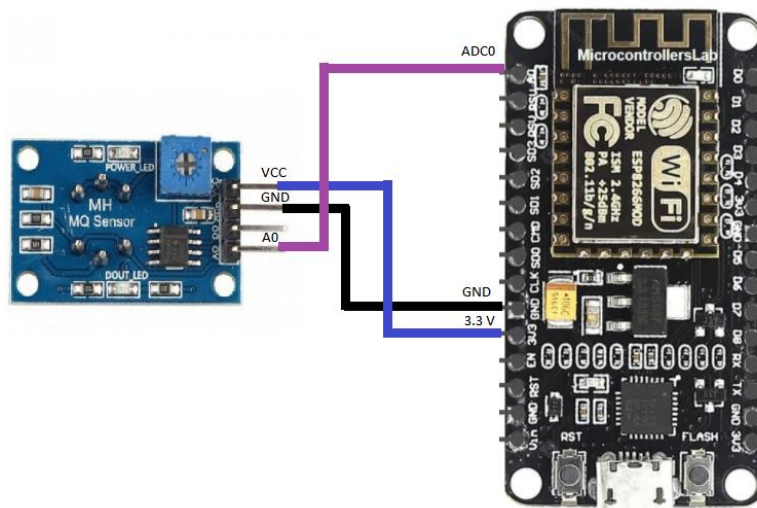


Figura 5. Conexionado ESP8266 - MQ2



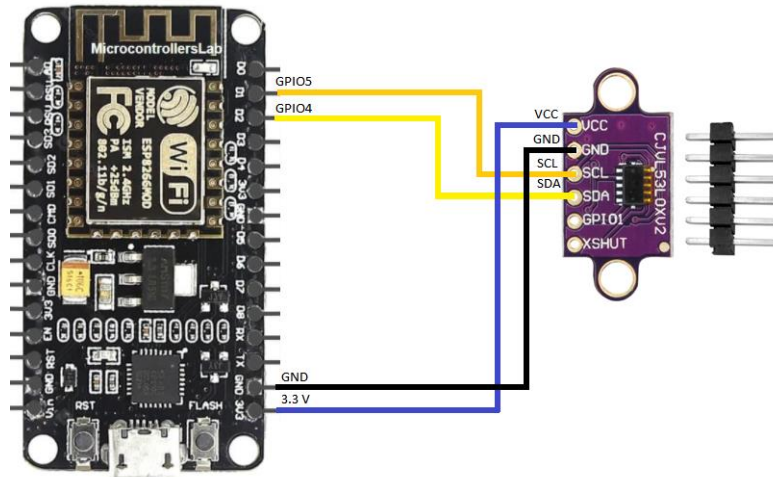


Figura 7. Conexión de ESP8266 - VL53L0X

### 3. DISEÑO DE SOFTWARE

#### 3.1. Librerías y topics

##### 3.1.1. Librerías

Para la realización tanto de flujos en Node-RED como uso de sensores y estructuras en Arduino, ha sido necesario requerir de diversas librerías. La descripción de estas se expone a continuación:

- ESP8266WiFi.h: Permite la conexión vía Wifi y, por tanto, envío de mensajes por MQTT.
- ESP8266httpUpdate.h: Permite la búsqueda de actualizaciones en un servidor OTA.
- PubSubClient.h: Permite la mensajería MQTT, tanto la suscripción como la publicación de topics.
- ArduinoJson.h: Permite entender y construir estructuras en JSON.
- ArduinoJson.h: Permite entender y construir estructuras en JSON.
- DHTesp.h: Permite reconocer y usar el sensor DHT11.
- Adafruit\_VL53L0X.h: Permite reconocer y usar el sensor láser VL53L0X.
- MQ2LiB.h: Permite reconocer y usar el sensor de calidad de aire MQ2.
- Node-red-contrib-nora: Permite usar la herramienta Google Home en Node-RED.
- Node-red-contrib-telegrambot-Home: Permite conectar el resto de flujo con Telegram.
- Node-red-tools/node-red-contrib-mongodb: Permite el uso de los nodos MongoDB en Node-RED.



### 3.1.2. Flujo de programa

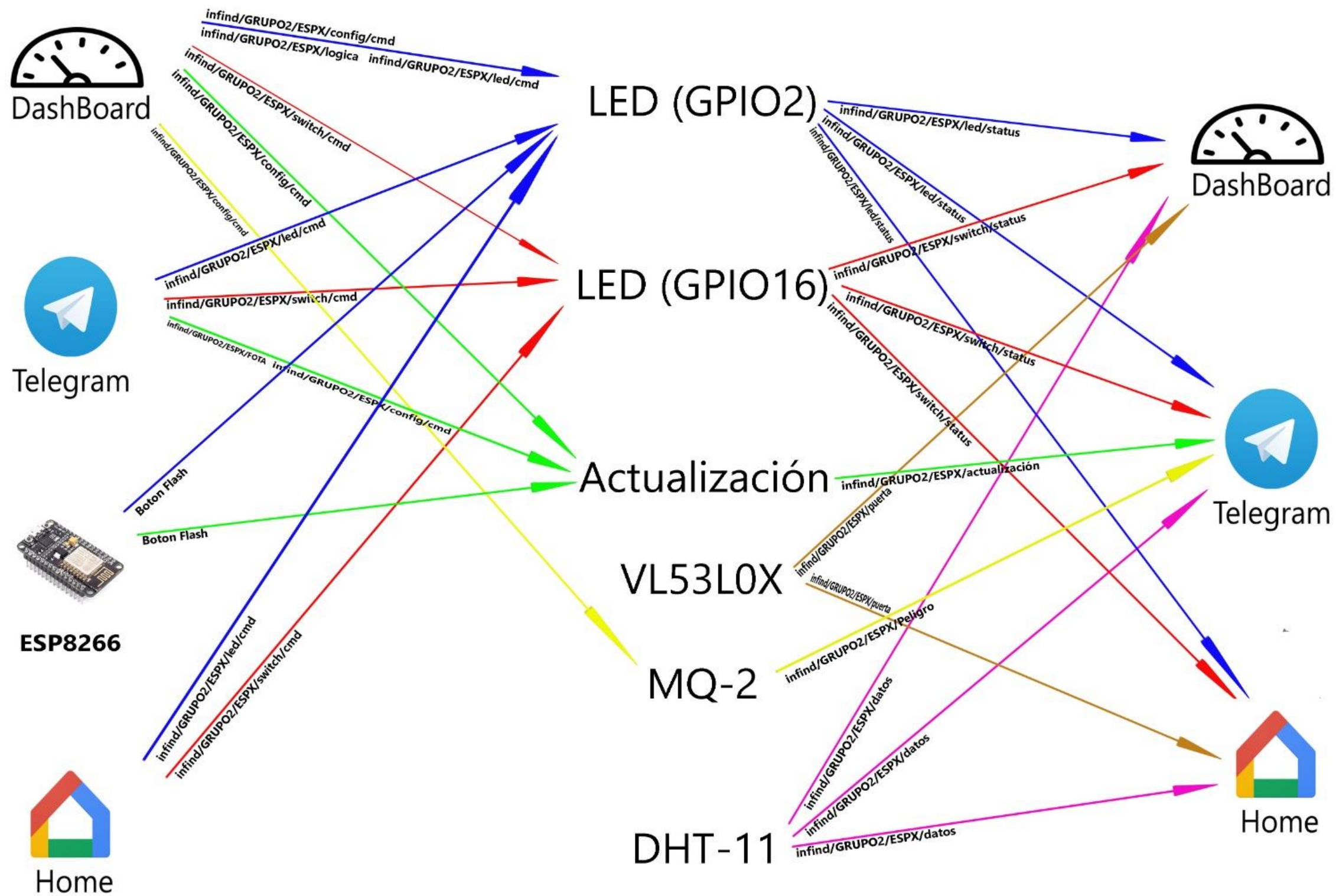


Figura 8. Esquema Global

### 3.1.3. Topics

Los distintos nodos que componen el sistema global se comunican entre sí basándose en subscripciones y publicaciones de mensajes formalizados en JSON con campos de interés. Entre ellos se encuentra la información de los sensores, búsqueda de actualizaciones, velocidad de envío de datos o de cambio de la intensidad en los leds, etc. De esta forma, se expone a continuación un listado de todos los topics que han sido usados para el proyecto.

Topic	Raíces de estructura JSON
PUBLICACIONES (ARDUINO)	
<code>infind/GRUPO2/ESPX/datos</code>	<code>{"CHIPID", "Uptime", "Vcc", "DHT11": {"temp", "hum"}, "LED", "SWITCH", "Wifi": {"SSID", "IP", "RSSI"}}</code>
<code>infind/GRUPO2/ESPX/conexion</code>	<code>{"CHIPID", "online"}</code>
<code>infind/GRUPO2/ESPX/led/status</code>	<code>{"CHIPID", "led", "origen"}</code>
<code>infind/GRUPO2/ESPX/switch/status</code>	<code>{"CHIPID", "switch", "origen"}</code>
<i><code>infind/GRUPO2/ESPX/actualizacion</code></i>	<i><code>{"actualiza"}</code></i>
<i><code>infind/GRUPO2/ESPX/puerta</code></i>	<i><code>{"CHIPID", "Estado", "VL53LOX"}</code></i>
<i><code>infind/GRUPO2/ESPX/Peligro</code></i>	<i><code>{"CHIPID", "CO", "LPG", "Humo"}</code></i>
SUBSCRIPCIONES (ARDUINO)	
<code>infind/GRUPO2/ESPX/FOTA</code>	<code>{"actualiza"}</code>
<code>infind/GRUPO2/ESPX/led/cmd</code>	<code>{"level"}</code>
<code>infind/GRUPO2/ESPX/switch/cmd</code>	DATOS
<code>infind/GRUPO2/ESPX/config/cmd</code>	<code>{"envia", "actualiza", "velocidad", "LED", "SWITCH"}</code>
<i><code>infind/GRUPO2/ESPX/logica</code></i>	<i><code>{"logica"}</code></i>

Tabla 1. Topics

Como bien se puede apreciar en la tabla anterior, aparte de los topics propuestos en las especificaciones del proyecto, se han creado hasta cuatro nuevos (pueden verse en un *color distintivo y en cursiva*), tres de ellos como publicaciones y el restante como subscripción.

- *`infind/GRUPO2/ESPX/actualizacion`*: Publica de la forma `{"actualiza": false}` en el momento en que alguna de las placas ESP buscan alguna actualización. Una vez la encuentran y se actualizan, o bien ya contaba con la última versión del programa (acabe la función de actualización siguiendo un método secuencial), se publicará tomando valor booleano `{"actualiza": true}`.
- *`infind/GRUPO2/ESPX/puerta`*: Publica de la forma `{"CHIPID": "ESPX", "Estado": "estado inicialización sensor", "VL53LOX": distancia}`. Para el campo "Estado" se pueden dar dos opciones, o bien "Fallo al iniciar" en caso de que el sensor láser haya tenido algún problema a la hora de inicializarse, o bien "VL53LOX activo" si por el contrario ha sido inicializado correctamente. El campo restante, "VL53LOX" hace saber la distancia marcada por el sensor láser (en caso de haber un fallo al inicializar esta distancia tomará el valor -1).
- *`infind/GRUPO2/ESPX/Peligro`*: Publica de la forma `{"CHIPID": "ESPX", "CO": PeligroCO, "LPG": PeligroLPG, "Humo": PeligroHumo}`. Cada uno de los campos (a excepción de *CHIPID*) muestra un valor (números enteros 0, 1, 2 o 3) que corresponde con un grado de Peligrosidad en el ambiente, bien sea por monóxido de carbono, gas natural o humo respectivamente. A mayor Peligrosidad, mayor entero asignado.
- *`infind/GRUPO2/ESPX/logica`*: A través de la subscripción a este topic con forma `{"logica": positiva/negativa}`, es posible cambiar el estado de la iluminación del led y del switch. De

forma predeterminada, estos GPIO2 y GPIO16 son modificados con lógica negativa, de forma que en caso de contar con consigna “level”: 0, ambos se apagarán, y con “level”: 100 ambos estarán encendidos a su nivel máximo. Se trata por tanto de una lógica negativa ya que a la hora de trabajar en Arduino un valor digital LOW implica apagar, HIGH implica encender y para el PWM se trabaja, aunque analógicamente, pero de forma similar. En contraposición, si se elige una lógica positiva, para el PWM el valor será el inverso, de forma que con una consigna “level”: 0, el led (GPIO2) se encenderá a su nivel máximo, y con una consigna “level”: 100, dicho led se apagará. Con el switch (GPIO16) ocurre lo mismo, ante consigna nula, el switch estará encendido, y ante consigna unitaria, este estaría apagado.

## 3.2. Node-RED

### 3.2.1. Introducción

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

La mayoría de los flujos tienen una estructura parecida: En primer lugar, un nodo de Telegram para recibir el comando, después una serie de nodos internos de Node-RED 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.

### 3.2.2. 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 como se ve en la Figura 9.

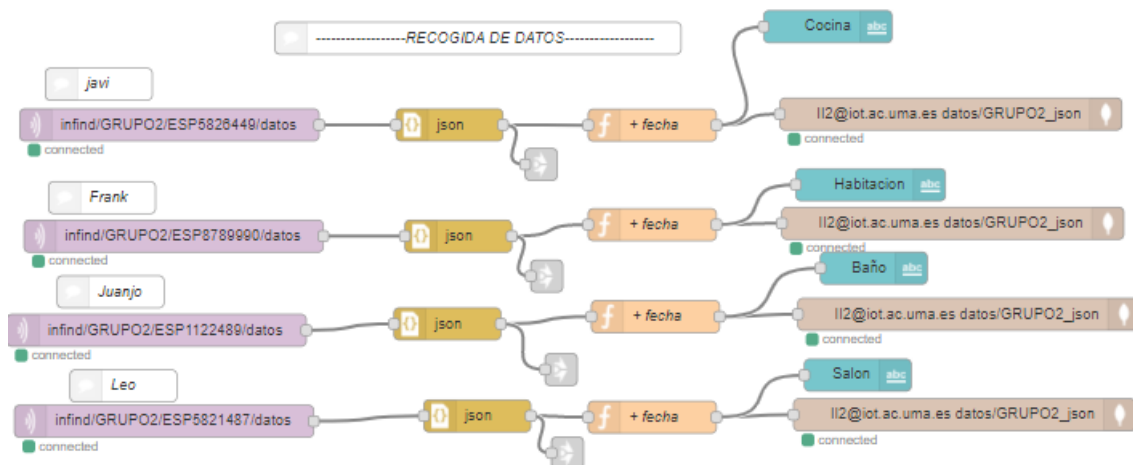


Figura 9. Flujo de recogida de datos

Node-RED, 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. Los datos que llegan se formatean en JSON y posteriormente se le añade la fecha. Estos datos se almacenan finalmente en mongoDB.

Tenemos otros topics que recogen datos adicionales, como los gases contaminantes que hay en la cocina, como vemos en la Figura 10. 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 más corta que cierto valor entonces detectamos “Puerta abierta”. En el caso de los gases recibimos niveles de peligro (0,1,2,3) y el Node-RED avisa cada vez que aumenta este nivel. También aparecerán avisos cuando se supere la temperatura o la humedad en el dashboard.

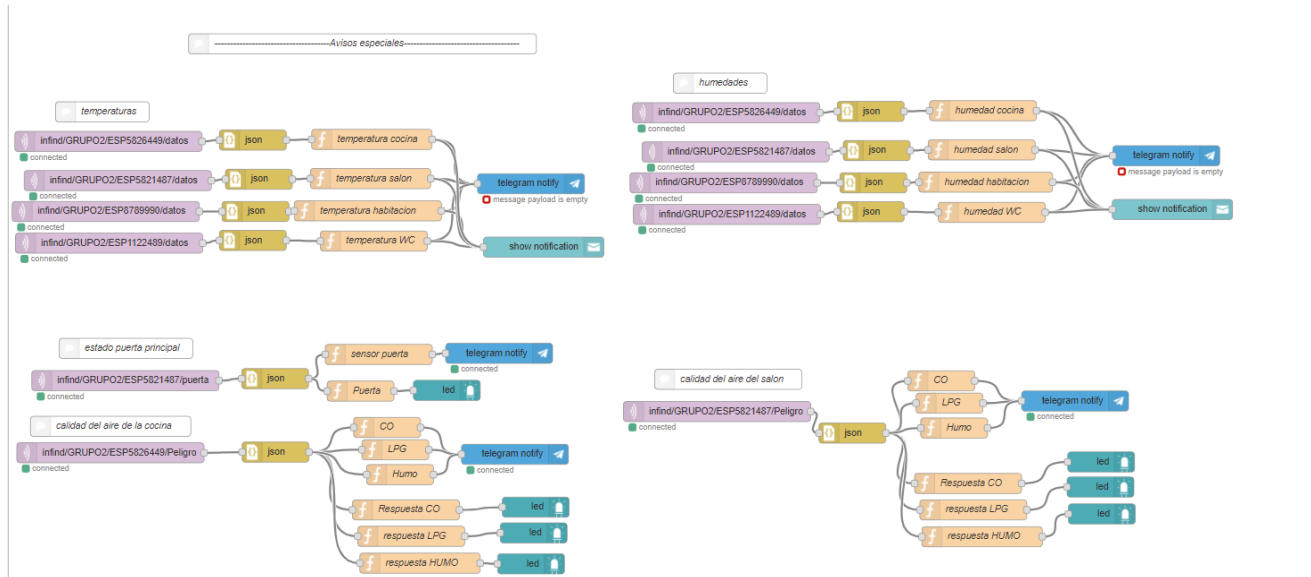


Figura 10. Flujo de avisos especiales

### 3.2.3. Envío de 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/ESPX/led/cmd`, cambiando X por el CHIPID propio de cada placa. Podemos observarlo en la Figura 11.

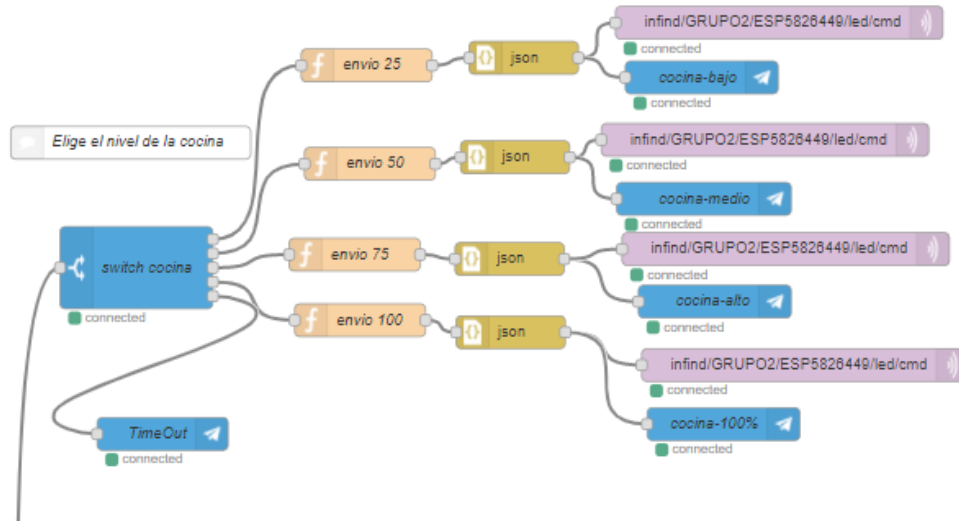
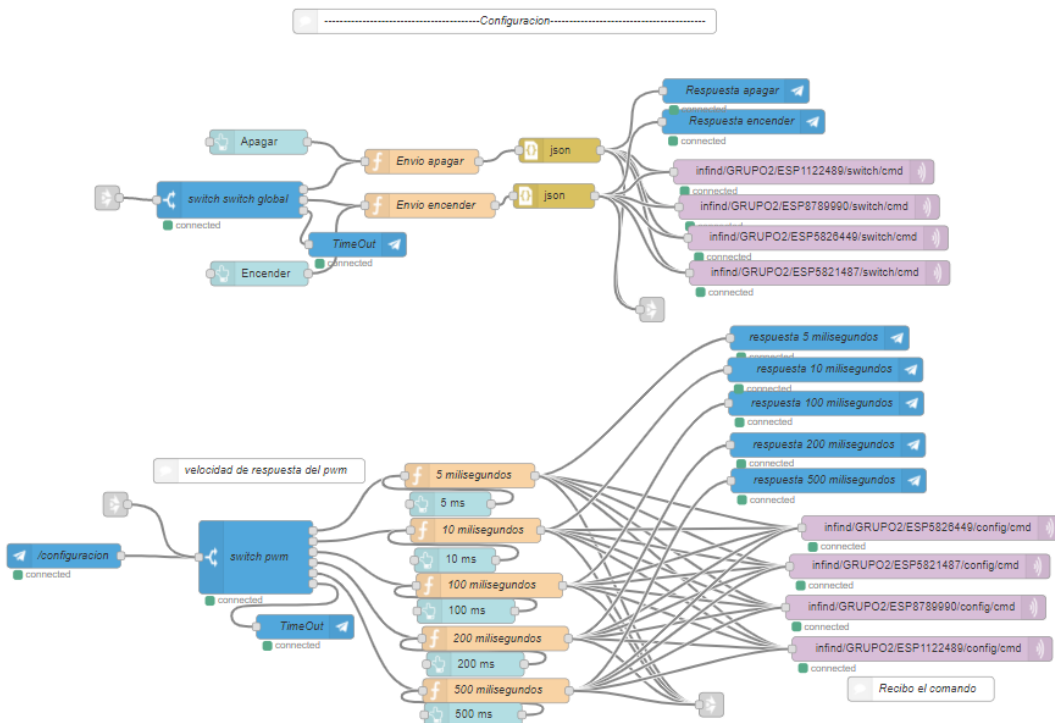


Figura 11. Flujo switch cocina

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/ESPX/config/cmd”. Los datos que se recogen son la velocidad de PWM, el tiempo de recogida de datos, el tiempo de actualización, el estado al que queremos el switch y el estado que queremos el led. Todos estos parámetros se podrán ajustar por Telegram. En la Figura 12 podemos ver este nodo MQTT.





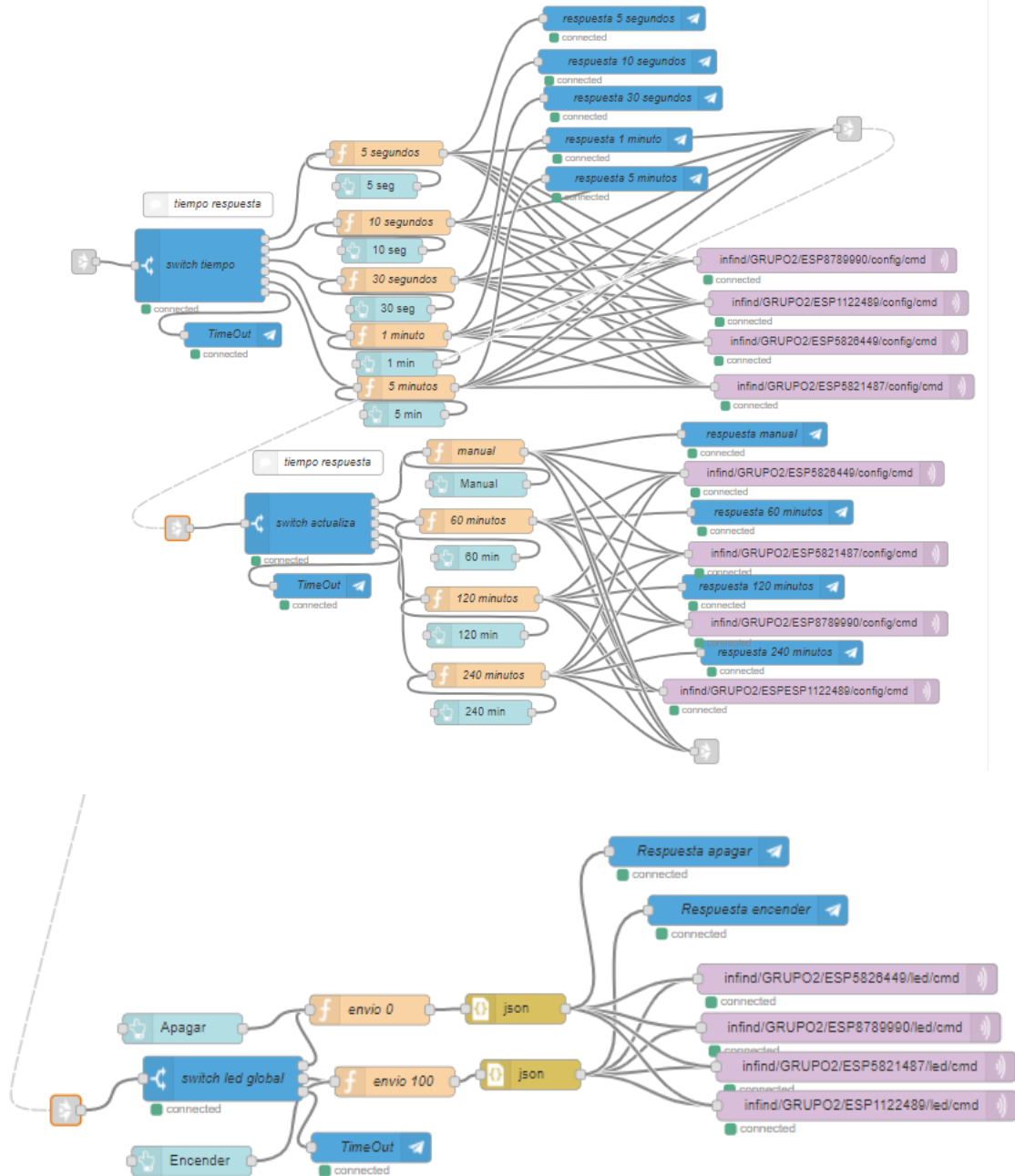


Figura 12. Flujo de configuración

### 3.2.4. 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 13. Otro topic avisará por Telegram durante y después de la actualización.

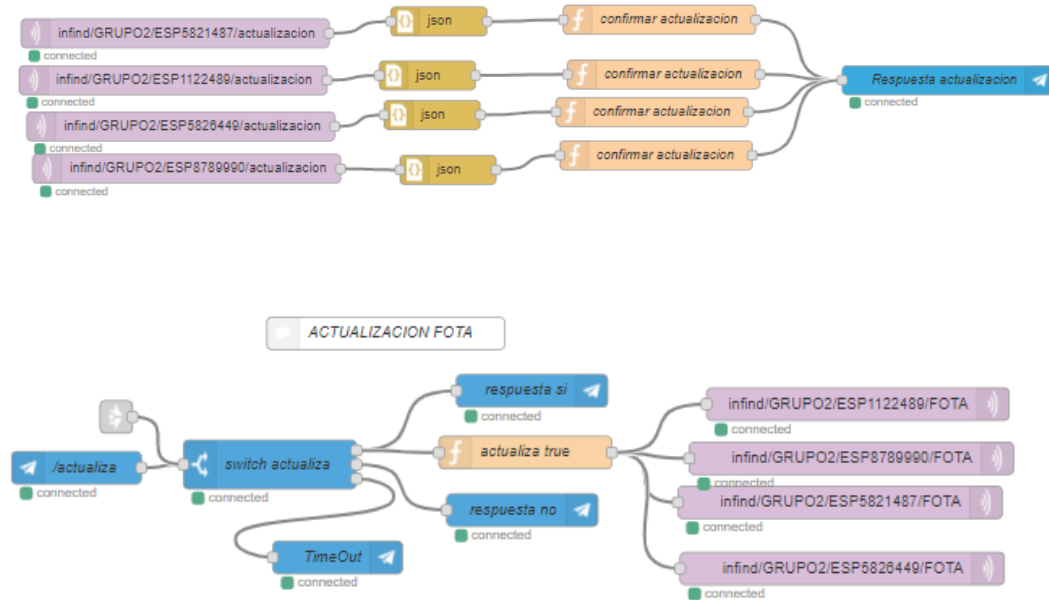


Figura 13. Flujo actualización

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 14.

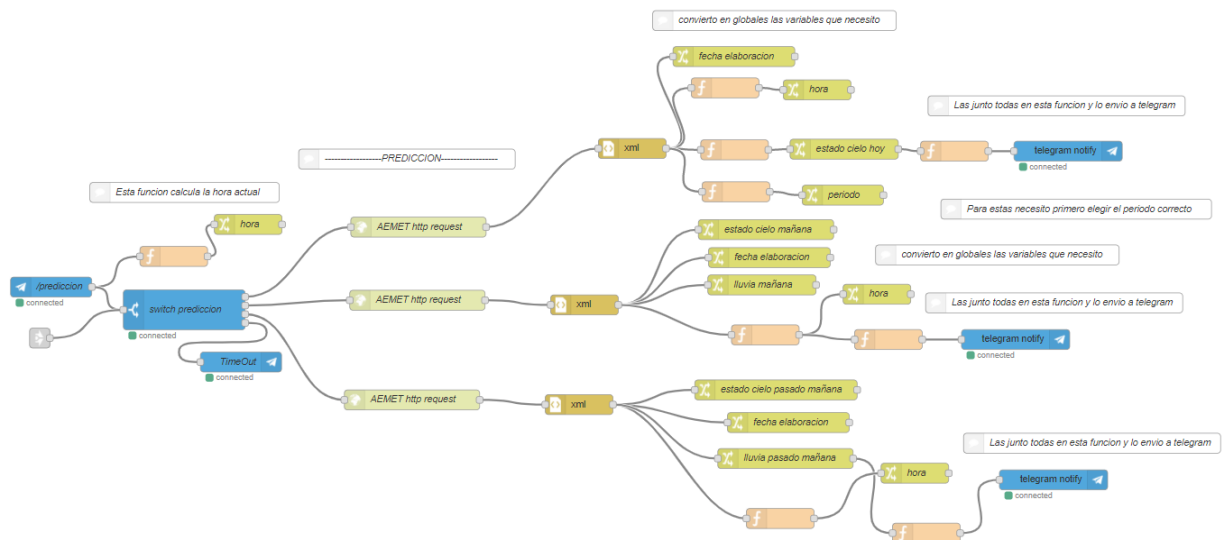


Figura 14. Tiempo atmosférico



### 3.2.5. 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. Posteriormente una función tomará los datos y los preparará para presentarlos en Telegram. Véase Figura 15.

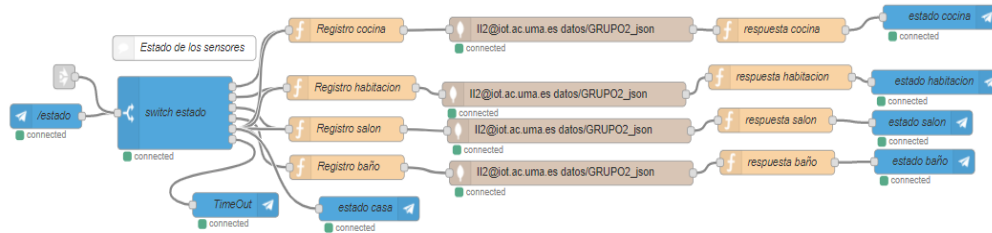


Figura 15. Flujo de consulta de estados

El formato de la consulta tendrá el aspecto de la Figura 16.

```

1 var vcc=msg.payload[0].vcc;
2 var fecha=msg.payload[0].date;
3 var Temperatura=msg.payload[0].DHT11.temp;
4 var Humedad=msg.payload[0].DHT11.hum;
5 var led =msg.payload[0].LED;
6
7
8 msg.payload=msg.mensaje;
9
10 msg.payload="El estado de la cocina es el siguiente: \n";
11 msg.payload+="El voltaje es de "+vcc+" voltios \n";
12 msg.payload+="Tenemos unos agradables "+Temperatura+" °C \n";
13 msg.payload+="La humedad es del "+Humedad+" % \n";
14
15 if (led == 25)
16 msg.payload+="La cocina esta a nivel bajo \n";
17 else if (led ==50)
18 msg.payload+="La cocina esta a nivel medio \n";
19 else if (led ==75)
20 msg.payload+="La cocina esta a nivel alto \n";
21 else if (led ==100)
22 msg.payload+="La cocina esta a nivel maximo \n";
23 else if (led === 0)
24 msg.payload+="La cocina esta apagada \n";
25
26 msg.payload+="Última actualización: "+fecha.toLocaleString('es-ES', { timeZone: 'Europe/Madrid', hour12: false
27
28
29 return msg;

```

Figura 16. Formato respuesta

### 3.2.6. Nodos para ajustar variables globales

Se han usado otras preguntas de Telegram para ajustar algunos valores, por ejemplo, las temperaturas límite para que la alarma nos avise.

Se ha implementado un limitador para la temperatura máxima y límite, se pueden elegir valores concretos desde Telegram y cualquier valor desde el Dashboard. La temperatura elegida se transformará a variable global y actuarán en las funciones de “avisos especiales” que vimos antes. En la Figura 17 podemos ver este flujo.

Además de la temperatura, la humedad también puede ser ajustada. El flujo es equivalente al anterior; elegimos una humedad inferior/superior y esta se pasa a una global para ser usada en “avisos especiales”. Veremos este flujo en la Figura 18.

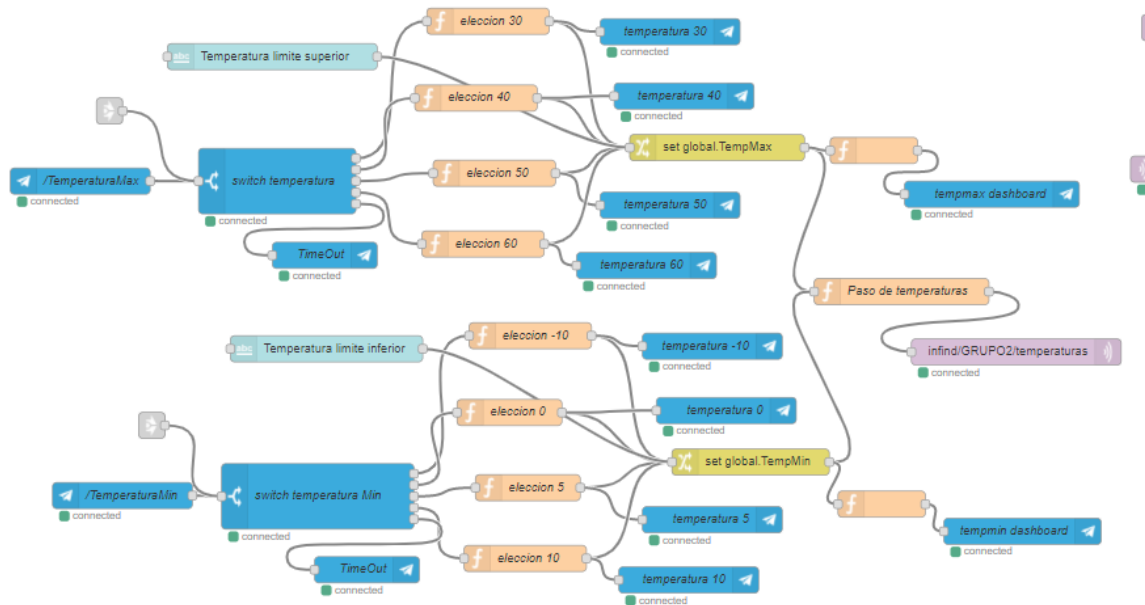


Figura 17. Ajuste temperaturas límite

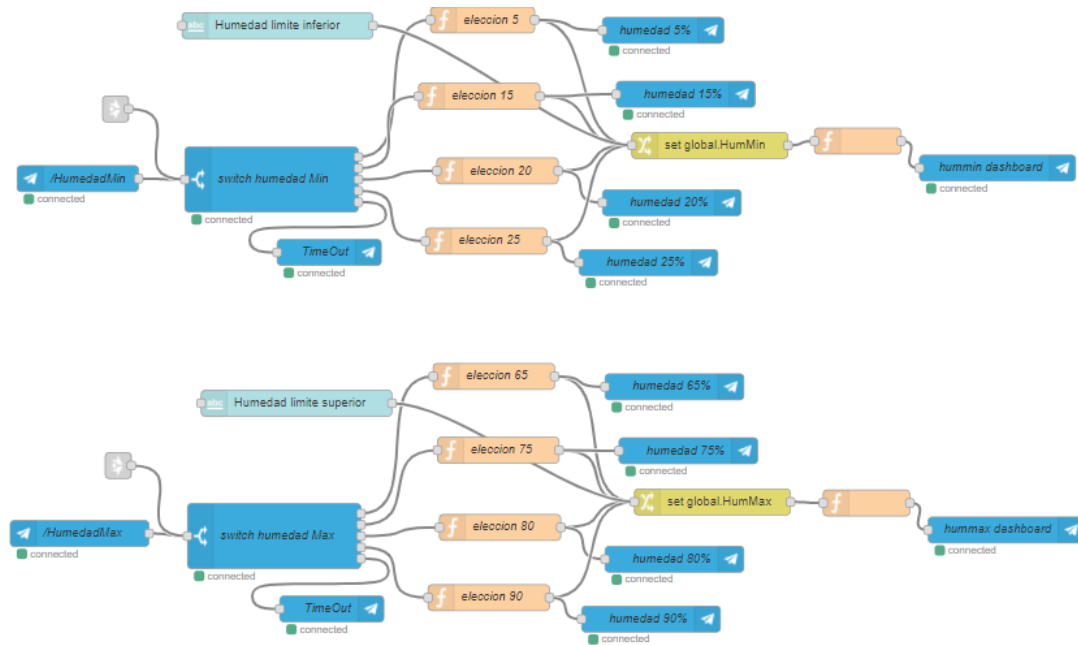


Figura 18. Ajuste humedades

### 3.2.7. Servicios adicionales

Se han agregado nodos que ofrecen más funcionalidad a nuestro bot. En primer lugar, podemos encender y apagar el led de la placa (el que no es PWM) mediante un switch desde el Dashboard o mediante un mensaje por Telegram. Véase Figura 19.

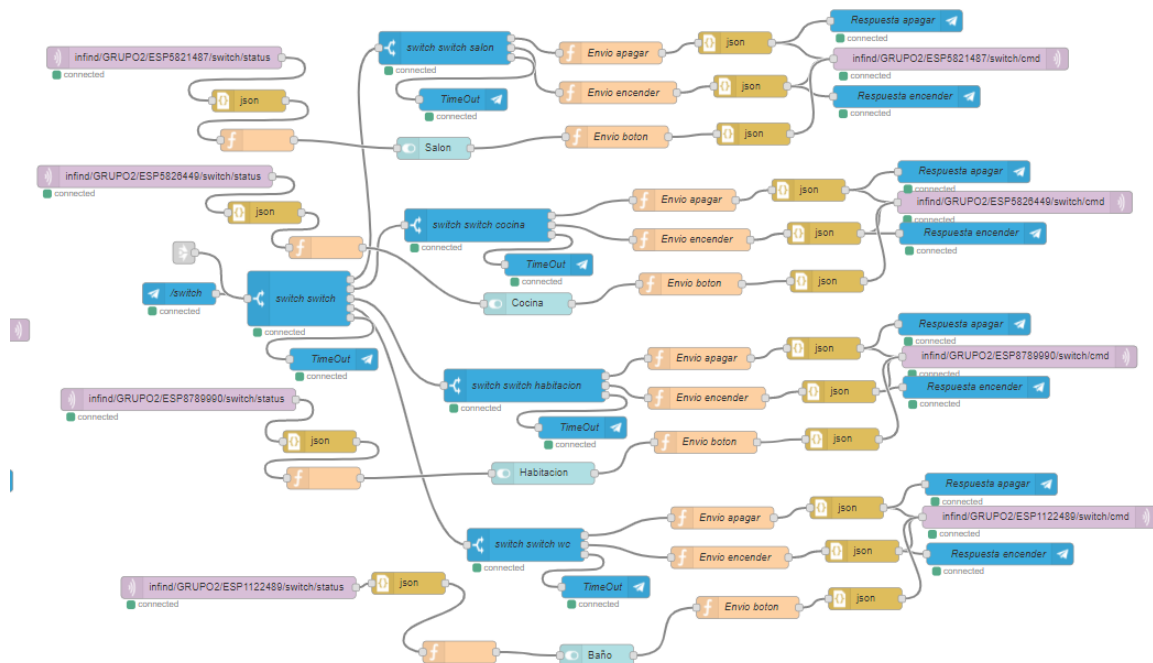


Figura 19. Flujo de switch individual

Adicionalmente podemos cambiar la lógica del PWM y del switch (ver apartado de Arduino). Se ha implementado con un flujo simple que recoge la respuesta que demos por Telegram y la manda por un topic. Véase Figura 20.

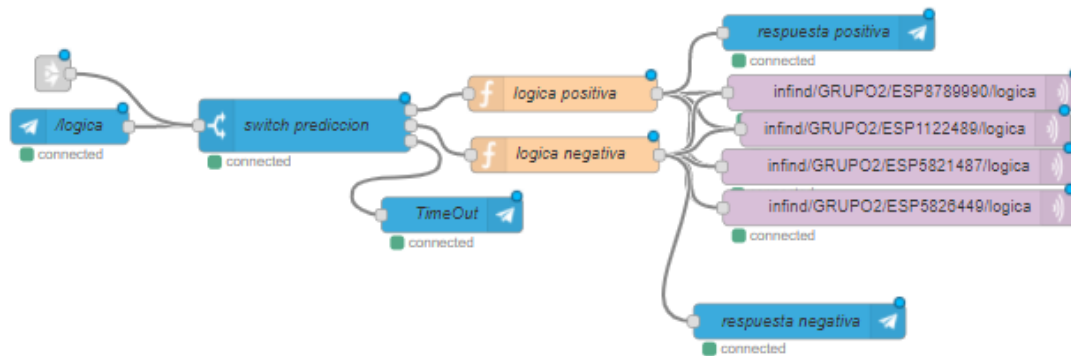


Figura 20. flujo de lógica

### 3.2.8. Dashboard

Otra parte esencial del Node-RED es su Dashboard, una interfaz que nos permite interactuar con los flujos. Además, se puede personalizar para obtener un estilo más personalizado.

Para montar el Dashboard son necesarios nodos especiales de Node-RED característicos por su color azul claro. Se ha dividido el Dashboard en 4 pestañas: panel de control, descarga Excel, configuración e históricos. No vamos a entrar en detalle sobre su funcionamiento en este apartado porque ya tenemos el manual de usuario más adelante.

El bloque Panel de control contiene información de las diferentes partes de la casa. Cuenta con displays para la humedad, temperatura, iluminación, además de gráficas que muestran la evolución en tiempo real de estos datos. También tiene un apartado para controlar la iluminación PWM y el switch. Cuenta con información sobre la fecha de la última actualización. En esta pantalla se muestran los avisos especiales como el peligro por humo o si la puerta está abierta o cerrada. En la Figura 21 podemos ver esta pantalla.

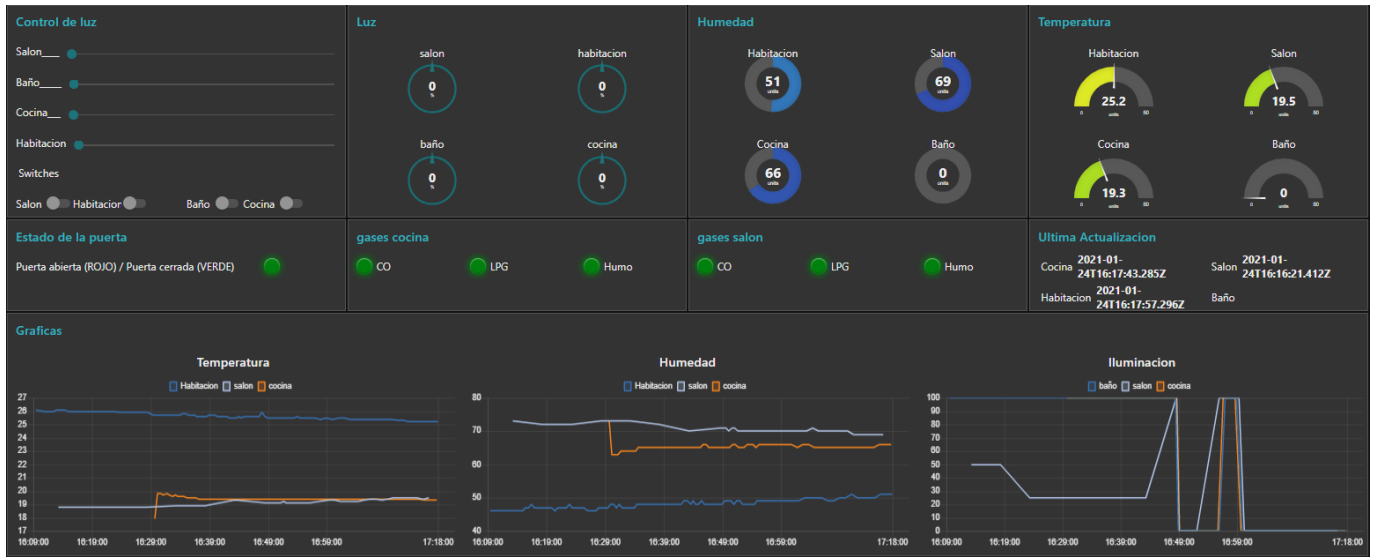


Figura 21. Interfaz panel de control

Hay múltiples nodos en todo el Node-RED que envían información al panel de control. El principal lo vemos en la Figura 22, donde se muestran los indicadores y los “sliders”.

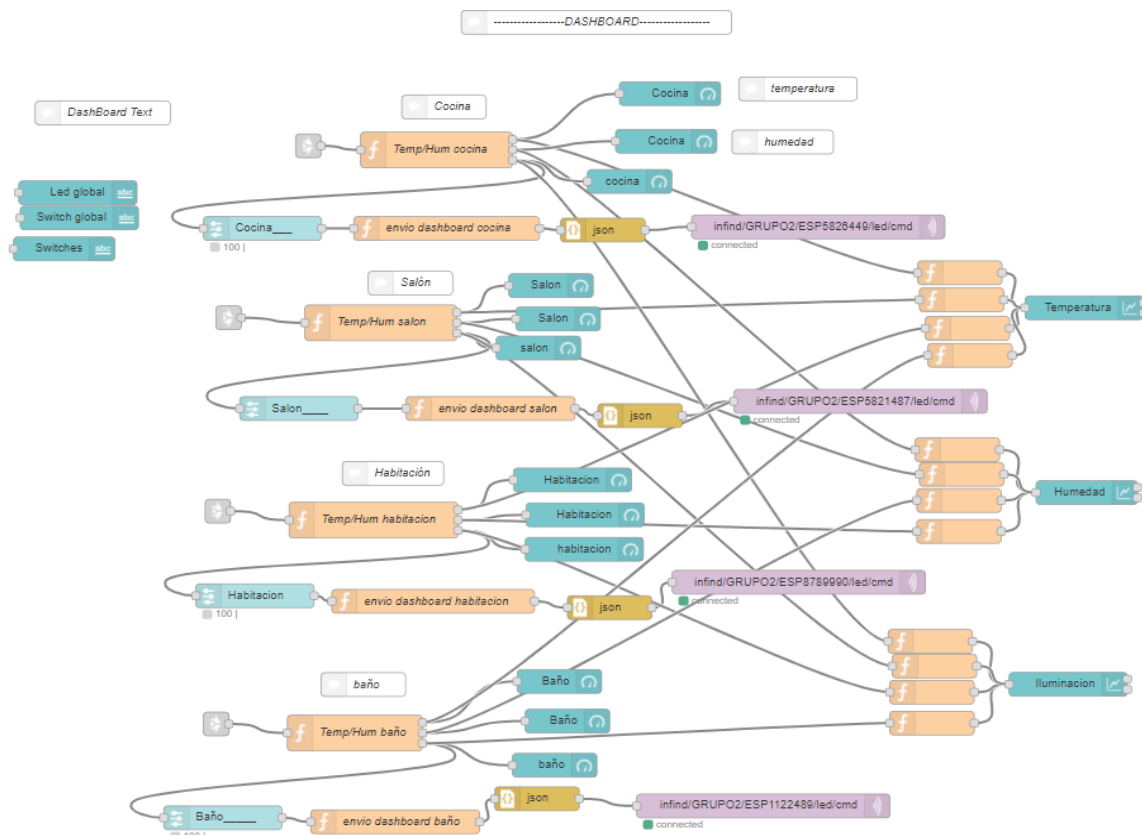


Figura 22. Flujo panel Dashboard

A continuación, tenemos la segunda pantalla más importante, la de configuración, en la Figura 23.

<b>Tiempo de pwm</b>			<b>Switch y Led</b>		
5 MS	10 MS	100 MS	Switch global	ENCENDER	APAGAR
200 MS		500 MS	Led global	ENCENDER	APAGAR
<b>Recogida de datos</b>			<b>Actualización de datos</b>		
5 SEG	10 SEG	30 SEG	MANUAL	60 MIN	
1 MIN		5 MIN	120 MIN	240 MIN	
<b>Temperaturas límite</b>					
Temperatura limite inferior			Temperatura limite superior		
<b>Humedades</b>					
Humedad limite inferior			Humedad limite superior		

Figura 23. Panel de configuración

En este bloque podemos ajustar algunos parámetros como se puede ver en la imagen. Los nodos se encuentran muy repartidos y se pueden apreciar en Figuras anteriores.

Veamos ahora el panel de descarga por Excel. Este flujo nos permite seleccionar un rango de fechas y descargar ciertos datos de ese rango. Se ha decidió recoger la temperatura, iluminación y humedad, además del topic y la fecha. En la Figura 24 vemos cómo se vería esta pantalla y en la Figura 25 el flujo que lo forma.

**Consulta de datos**

Fecha inicio 23 ene. 2021

Fecha final 23 ene. 2021

Descarga registros en CSV (EXCEL)

separador decimal = coma

Figura 24. Descarga Excel

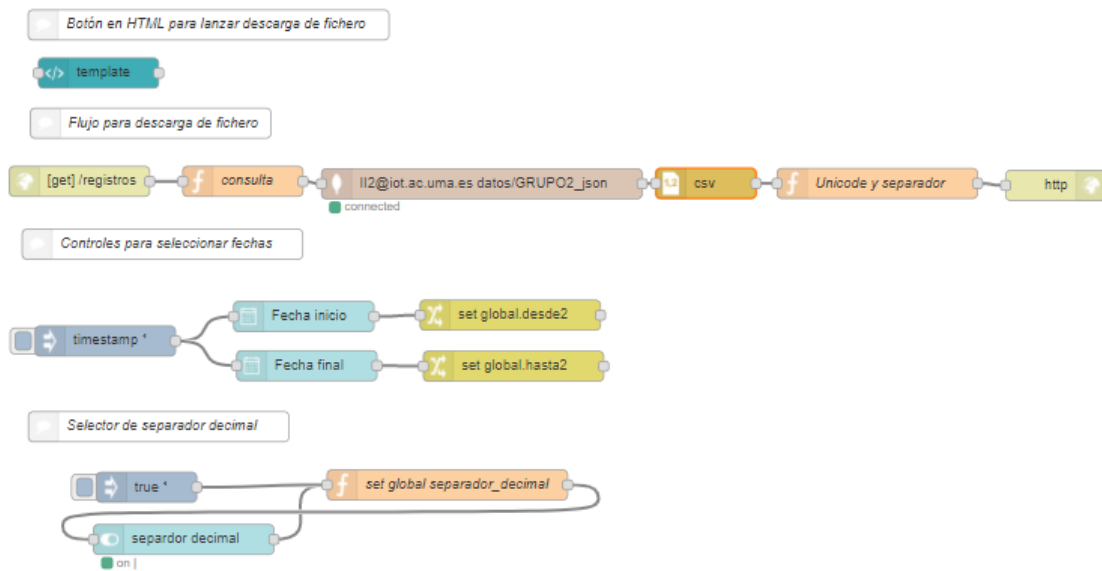


Figura 25. Flujo descarga Excel

Ya solo nos falta ver la pestaña de Históricos. En esta pestaña podemos elegir un rango de fechas y podremos ver una gráfica con la temperatura media, máxima y mínima. Tenemos una gráfica por parte de la casa. En la Figura 26 solo veremos una de ellas. En la Figura 27 vemos el flujo de Node-RED que nos permite tener esta pantalla (uno de los cuatro). Tomamos un par de variables de fechas y la introducimos en una función que nos busca todos los datos pedidos y nos devuelve sus medias, mínimos y máximos. Estos datos los llevamos a una gráfica del Dashboard.

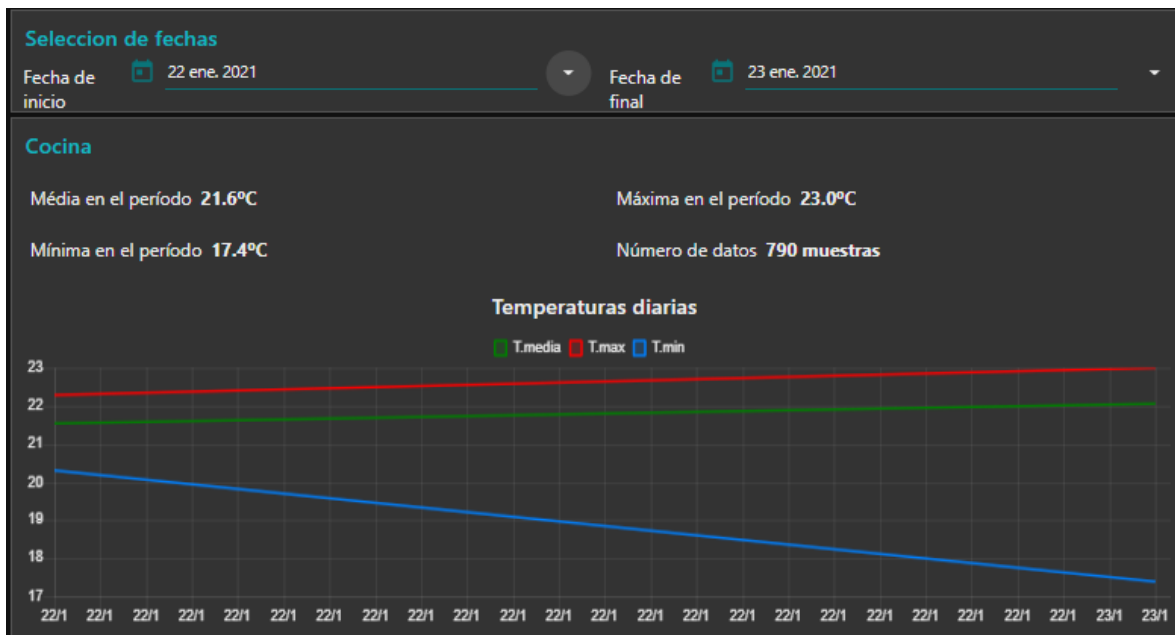


Figura 26. Histórica temperatura



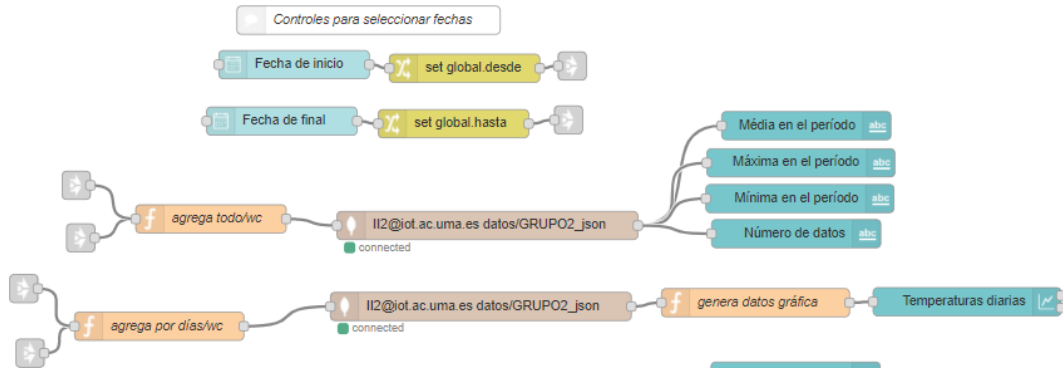


Figura 27. Flujos históricos

### 3.2.9. Google Home

Para que el Google Home detecte dispositivos usamos la librería de Nora en el Node-RED. Para todos los nodos de esta librería tenemos que crear una Config que se lo ponemos a todos los nodos, para la Config aparte de decirle un nombre tenemos que ponerle un token, para conseguirlo lo cogemos desde la página de Nora donde tendremos que iniciar sesión para obtenerlo.

Para dispositivos que sean leds que se pueda cambiar el brillo a gusto, se usa el nodo Light. En el nodo light tenemos que meter el Config que hemos creado anteriormente y ponerle un nombre, que es el nombre que muestra en la aplicación de Home. El interior de uno de los nodos se muestra en la Figura 28.

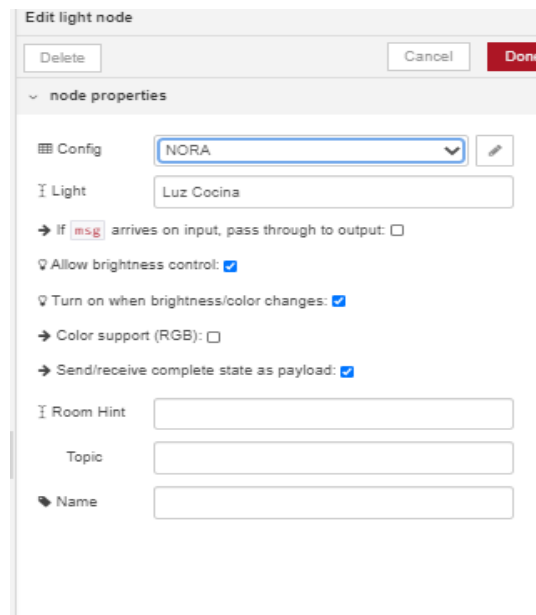


Figura 28. Nodo led Google Home

En esta parte cogemos los datos del topic status del led, este le ponemos un payload que tenga: “online” que siempre será true, un “on” que tendrá true o false si el brillo es distinto de cero o no para apagarlo o encenderlo y por último “brightness” que es el valor del brillo. Después cuando salen datos del nodo Nora que es cuando se modifica desde Google Home, se le mete una función que dependiendo del valor del “On” lo pone en 0 o en el valor del brillo que nos saca del mismo modo que en la función de entrada. El esquema general se ve en la Figura 29, esto se tendrá que repetir para cada placa.

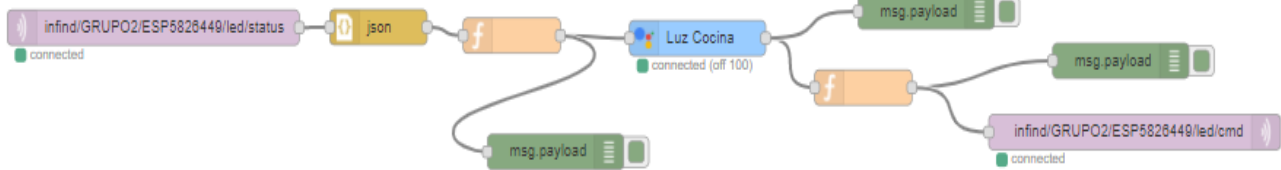


Figura 29. Flujo Node-RED Google Home

Ahora para crear un dispositivo para los leds que solo tienen encendido y apagado, para eso usamos el nodo switch.

En el nodo switch tenemos que meter el config que hemos creado anteriormente y ponerle un nombre, que es el nombre que muestra en la aplicación de Home. También le ponemos en los payload de entrada para el on y off le ponemos la configuración que queremos, que es el valor del switch que lo cambiamos a “level” para pasarlo directamente al cmd. El interior de uno de los nodos se muestra en la Figura 30.

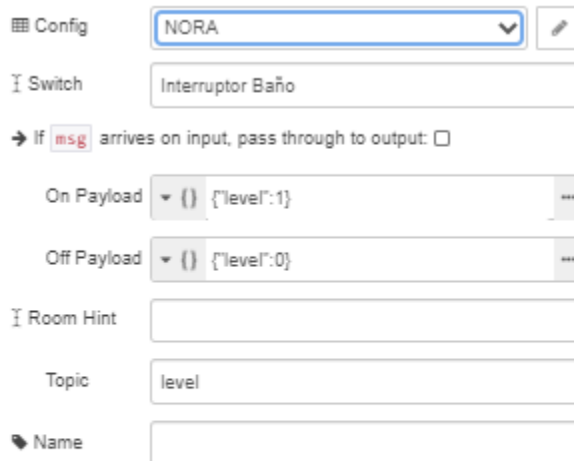


Figura 30. Nodo switch Google Home

El valor que nos da el nodo se le puede enviar directamente al topic cmd del switch y para meterle el valor al nodo Nora, se desde el topic status del switch el valor del switch cambiándole de nombre a "level". El esquema general se ve en la Figura 31, esto se tendrá que repetir para cada placa.

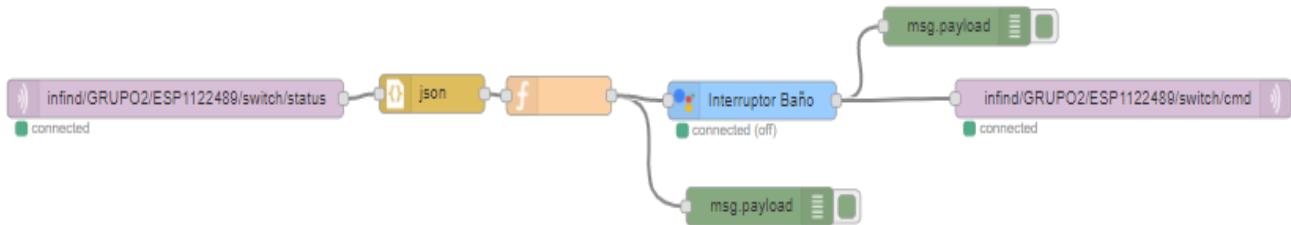


Figura 31. Flujo switch Google home

Por último, para saber los datos de temperatura y humedad le hemos metido el nodo Nora de thermostat. En este nodo tenemos que meter el nombre y el config como en los anteriores y marcarle que nos dé en grados centígrados y que tenga modo heat y cool. Véase Figura 32.

**Config**

**Thermostat**

→ If **msg** arrives on input, pass through to output: ☐

Temperature Unit: ☒ C ☐ F

**Supported modes**

<input type="checkbox"/> off	<input checked="" type="checkbox"/> heat	<input checked="" type="checkbox"/> cool
<input type="checkbox"/> heatcool	<input type="checkbox"/> on	<input type="checkbox"/> auto
<input type="checkbox"/> fan-only	<input type="checkbox"/> purifier	<input type="checkbox"/> eco
<input type="checkbox"/> dry		

Buffer range celsius

Indicates if the device operates using one-way (checked) or two-way communication: ☒

Required if the device supports query-only execution: ☐

**Room Hint**

**Topic**

**Name**

Figura 32. Nodo termostato

Para este nodo le cogemos desde el topic de datos y le pasamos el valor de la temperatura y la humedad. Aparte de darnos eso datos hacemos que se ponga en modo heat si se pasa un límite inferior de la temperatura y que se ponga en modo cool si se pasa en otro límite superior. Estos limites se cogen de un topic que nos da los dos valores (que se establecen en el Dashboard) y

lo convertimos en variables globales para tenerlo en todos los distintos nodos, ya que usan los mismos límites. Para enviarle los datos se tiene que poner el payload: “mode” indicamos el modo que será “heat”, “cool” o “Off” dependiendo si se ha alcanzado los límites o no, “setpoint” que lo ponemos a 22 que es la temperatura a la que se enciende la calefacción o el aire acondicionado, “setpointLow” que pasamos el límite inferior, “setpointHigh” que nos dice el límite superior, “humidity” que decimos el nivel de humedad que hemos cogido del topic datos y “temperature” que es el valor de la temperatura que hemos sacado del topic datos. Desde el nodo Nora no sacamos nada a ningún sitio, solo nos dice si ha cambiado algo desde la aplicación. El esquema general se ve en la Figura 33, esto se tendrá que repetir para cada placa, excepto por lo de las variables globales que son para todas.

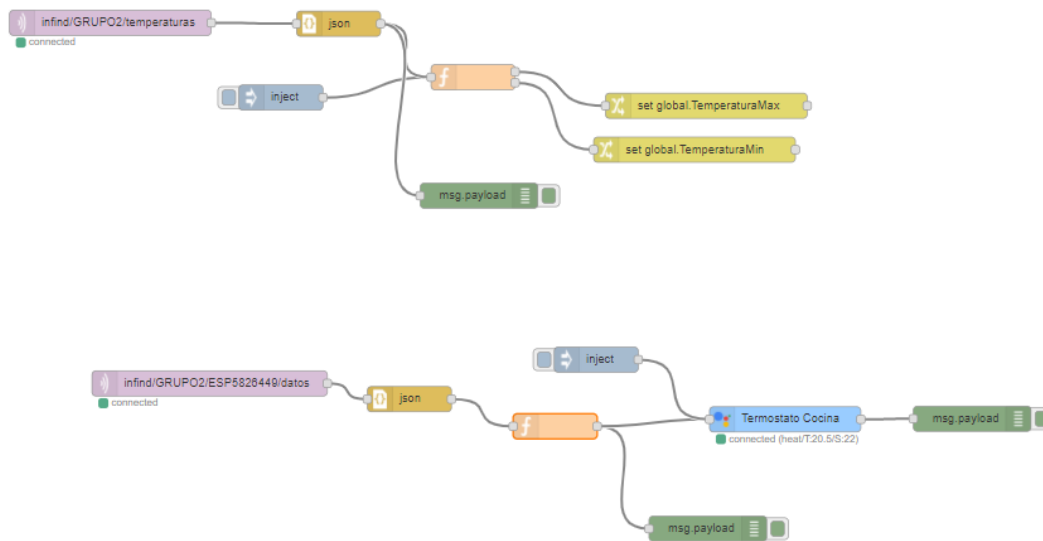


Figura 33. Flujo termostato

### 3.3. Arduino

#### 3.3.1. Introducción

En este trabajo se ha utilizado Arduino IDE, una herramienta de programación basada en código C++. A través del código programado en Arduino, se ha podido usar la placa ESP8266 con varias de las funciones que nos permite implementar. Entre ellas está el control de la iluminación de los leds por método directo (a través del botón flash o GPIO0) o bien por MQTT desde Telegram o el Dashboard; permite el envío de datos estructurados formalizados en JSON, cuyos datos van desde el estado de los leds o de los sensores hasta configuraciones propias del WiFi o de la propia placa como son la SSId, IP, o el CHIPID.

De esta forma, el usuario podrá conocer todos estos valores, que serán enviados por MQTT o bien de forma periódica, como son los datos globales, o bien cada vez que el usuario realiza cambios en alguno de los leds, o busca alguna actualización, entre otras.

### **3.3.2. Inicialización y declaración de estructuras**

A continuación, se define los tipos de datos para cada estructura JSON, en donde se especifican todos los datos que se van a enviar por cada estructura y el tipo al que pertenecen (float, bool, long...). En todos los topics se envían datos con una estructura JSON.

Tras esto, se produce la declaración de las variables globales que se utilizarán en varias funciones del código. Se han especificado como 'volatile' aquellas que tienen que ver con interrupciones (ya que pueden ser modificadas tanto dentro del código como desde fuera). De esta forma, se declaran las constantes y variables y en algunos casos se les asigna un valor inicial. Así mismo, en esta sección se han creado las estructuras de todos los topics que se usarán, tanto para publicar como para suscribir.

### **3.3.3. Interrupciones**

Posteriormente, se encuentra la Rutina de Tratamiento para la Interrupciones (RTI). Esta función nos permitirá utilizar el botón flash (GPIO0) que tendrá distintas utilidades dentro del código. Al ser un botón físico, ha sido necesario implementar un filtro anti-rebotes que elimina los pulsos con tamaño inferior a 50 ms.

### **3.3.4. Funciones para la serialización de datos JSON**

Posterior a esto se definen todas las funciones (serializa\_JSON) como conversión a String de las estructuras JSON para que puedan ser interpretados. Dentro de cada función se crean todos los grupos y subgrupos en JSON que almacenarán datos concretos. Estos datos serán publicados posteriormente por MQTT.

### **3.3.5. Conexión**

Seguidamente aparece la conexión a internet. En esta parte se establece la conexión WiFi.

Detrás de la conexión a internet está la conexión a MQTT. En esta función nos permite conectar como clientes (en función del CHIPID de cada placa) a un servidor determinado, en este caso es necesario introducir usuario y contraseña, especificando también un mensaje de últimas voluntades en caso de desconexión. Una vez se establezca la conexión procedemos a subscribirnos a los topics correspondientes; estos están el de actualización FOTA, configuración completa del dispositivo o bien los valores deseados del led o el switch.

### **3.3.6. Función de recepción de datos por MQTT**

A continuación, se desarrolla la función Callback, donde se toman los distintos datos que son enviados por MQTT y a los que estemos previamente suscritos (esta subscripción se establece en la función de conexión MQTT). De esta forma, cada vez que se reciba un dato de un topic específico, se asigna el valor de la raíz a una variable con la que se trabajará según la funcionalidad requerida

(buscar actualizaciones, encendido o apagado del led (GPIO2) o switch (GPIO16), cambio en el envío de datos, velocidad de cambio en el PWM...). En caso de que alguno de los mensajes recibidos pretenda variar la intensidad del led, se activará una señal que permitirá su modificación como PWM más adelante en la función principal loop ().

### **3.3.7. Setup y OTA**

Posteriormente se encuentra el setup, donde se inicializan los pines a usar en la ESP8266, se hace una llamada a cada una de las funciones anteriormente descritas, y se asigna a cada placa un identificador que permite inicializar unos sensores u otros, ya que no todas las estancias de la vivienda (o placas ESP8266) cuentan con los mismos sensores.

En penúltimo lugar, se desarrollan todas las funciones que serán usadas por la actualización OTA, entre ellas citar especialmente la del progreso de actualización en caso de que la firma OTA no coincida con la de la última actualización disponible.

### **3.3.8. Función loop**

Finalmente se desarrolla el loop. Se trata de una función que se repetirá en bucle continuamente. Una vez se haya establecido conexión MQTT comienza el bucle. En esta sección se realizan distintas tareas, entre ellas está el estudio de las interrupciones y las respectivas pulsaciones al botón flash (esto es, si se ha pulsado una, dos o más veces o bien se ha mantenido pulsado un cierto tiempo) y su posterior consecuencia (encendido o apagado del led o búsqueda de actualizaciones); cambio de la intensidad del led (GPIO2) como PWM (aumentando en pasos de 1% hasta alcanzar el valor deseado) búsqueda de actualizaciones o incluso publicación de datos por MQTT (datos globales y sensores láser y MQ2). Para el caso del envío de datos globales, la publicación se producirá cada cierto tiempo de forma periódica; en contraposición, el sensor láser y el de calidad del aire publicarán únicamente cuando la medida tomada supere un umbral determinado.

## **3.4. Telegram**

### **3.4.1. 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 Node-RED. En el punto 3.1 ya vimos estos elementos de Telegram, en este punto veremos en detalle cómo funciona.

### **3.4.2. Funcionamiento**

Lo primero que tenemos que hacer es enviar un mensaje de bienvenida, como vemos en la Figura 34. Al escribir /start recibimos un mensaje que nos indica las múltiples opciones que tenemos, también tenemos la opción de ver un mensaje tutorial que nos indicará cómo usar cada comando.

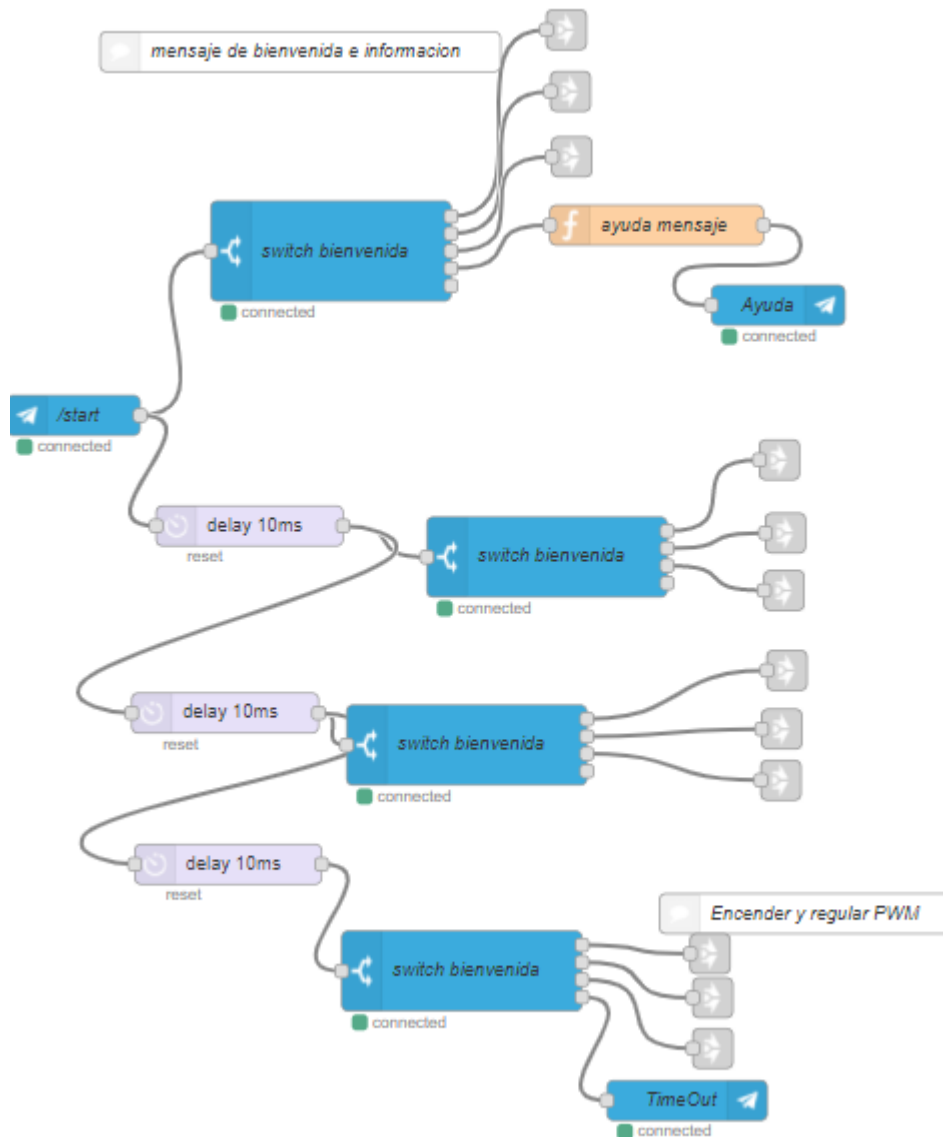


Figura 34. Flujo de bienvenida

En la Figura 35 podemos ver cómo lo veríamos en Telegram. Lo primero que haremos si es la primera vez que se utiliza el bot es ver el mensaje tutorial.



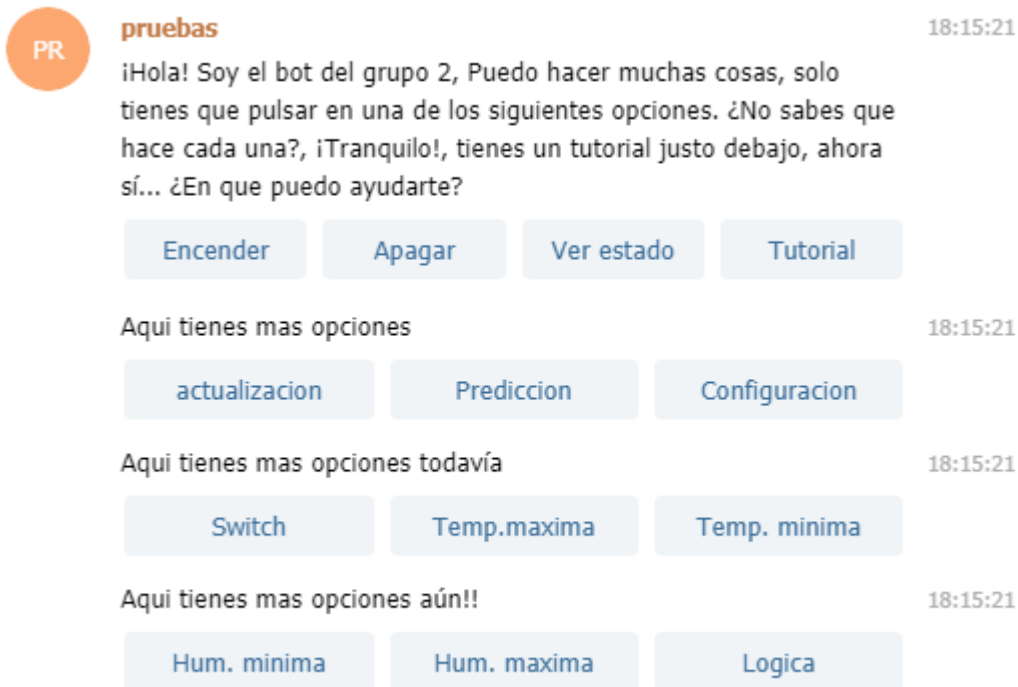


Figura 35. Telegram, bienvenida

El mensaje tutorial nos resume todos los comandos que hay y que hace cada uno. En la Figura 36 vemos el texto que aparece al clicar en “Tutorial”.

¿Estás un poco perdido? Tranquilo, yo te ayudo. 18:15:49  
En primer lugar puedes utilizar el comando `/encender` o `/apagar` para modificar las luces.  
Una vez lo hagas podrás elegir qué quieres encender, tienes varios niveles para elegir.  
Por otro lado tienes el comando `/estado` para comprobar cierta información de cada rincón de la casa.  
Otro comando interesante es `/prediccion`, que te permite ver la prediccion del tiempo atmosferico.  
Con el comando `/configuracion` podrás tocar varios parámetros como el tiempo de recogida de datos, el de pwm, el estado del switch y del led y el tiempo de actualizacion . Tambien puedes actualizar manualmente con `/actualiza`  
Si escribes `/TempMax` o `/TempMin` podrás cambiar la temperatura a la que quieres los avisos  
Con `/HumMax` y `/HumMin` puedes regular los avisos para la humedad, con `/logica` puedes cambiar la lógica de los LEDs  
Hay otros mensajes que se envian automaticamente si ocurren situaciones concretas, como por ejemplo, que se abra la puerta de la casa o que tengas un nivel de humo demasiado alto.  
¿No me crees? Pues mete fuego a la casa para que veas lo efectivo que soy ^^.

Figura 36. 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 37. Hemos clicado en “cocina” para ver el estado de la cocina.

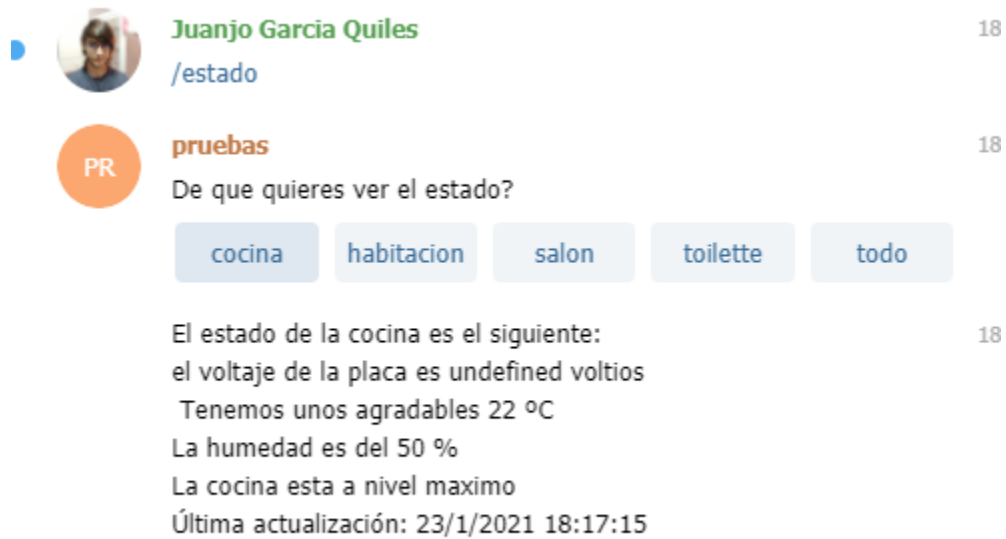


Figura 37. Estado Telegram

Para el flujo de predicción veríamos algo como en la Figura 38.

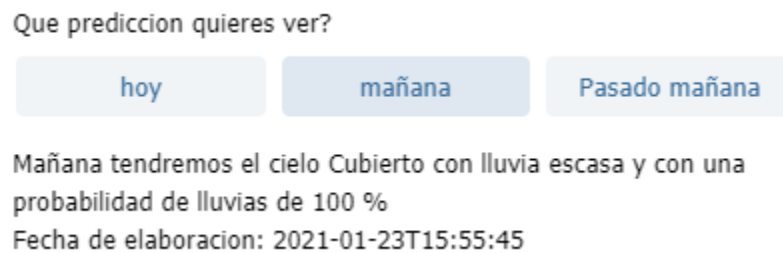


Figura 38. Telegram, predicción

Si usamos el comando /configuración saldrán una serie de preguntas que iremos contestando con los datos que queremos. Entre ellas está la posibilidad de elegir actualización manual. Véase la Figura 39.

actualizacion	Prediccion	Configuracion	switch
Que tiempo de pwm de datos quieres?			18:49:07
5 milise...	10 milis...	100 mili...	200 mili...
Muy bien, ¿Que tiempo de recogida de datos quieres?			18:49:08
5 segun...	10 segun...	30 segun...	1 minuto
Se ha establecido el pwm en 100 milisegundos			18:49:08
Se ha establecido el tiempo en 30 segundos			18:49:10
Y por último, ¿Que tiempo de actualizacion de datos quieres?			18:49:10
manual	60 minutos	120 minutos	240 minutos
Se ha establecido la actualizacion en 120 minutos			18:49:11
Que quieres hacer con el switch global?			18:49:11
Apagar			Encender
Se han encendido los switches			18:49:14
Que quieres hacer con el LED global?			18:49:14
Apagar			Encender
Se han encendido los leds			18:49:16

Figura 39. configuración Telegram

El resto de las funciones se basan en una única pregunta con respuesta, se verá con detalle en el manual de usuario. Se puede encender y apagar el led por PWM, actualizar las placas, apagar los switches de manera individual, cambiar las temperaturas y humedades límite y la lógica.

### 3.5. Robustez

Por lo que respecta a los aspectos robustos de nuestro sistema, se han incluido en código C++ (Arduino) algunas técnicas que nos aseguran un correcto funcionamiento global en caso de que ocurra alguna interferencia o fallo de cualquier tipo. Algunas de estas técnicas son las siguientes:

#### 3.5.1. Topics

Para comprobar que se ha recibido cierta información desde algún sensor o bien conocer el estado de conexión/desconexión, se han definido una serie de estructuras en Arduino que nos permiten estar informados cada cierto tiempo. Algunas de estas son:

- Conexión WiFi. Gracias al topic infind/GRUPO2/ESPX/conexion, somos conscientes del valor del último estado de conexión (conectado o desconectado). Teniendo en cuenta los mensajes de últimas voluntades, el sistema es capaz de notificar al usuario el estado de desconexión. En caso de no ser capaz de conectarse a la WiFi, se volverá a intentar una conexión cada 5 segundos, hasta conseguir el objetivo.
- Sensor láser. Se ha realizado un programa que permite avisar al usuario en caso de que no se haya conseguido una correcta inicialización del sensor. Esto se produce a través

del topic infind/GRUPO2/ESPX/puerta. Como ya se ha comentado, en caso de que se haya producido un fallo, se hace saber con un comentario específico, dando como valor de distancia un -1. En caso contrario, se hace saber al usuario que la puerta del salón está abierta o cerrada, estudiando la distancia medida y siendo esta enviada por MQTT. La decisión final de que esté abierta o cerrada la puerta se hará sin embargo en Node-RED.

- Valor led y switch. Cada vez que se selecciona un valor determinado del led o del switch, se envía por el topic infind/GRUPO2/ESPX/led/status o bien infind/GRUPO2/ESPX/switch/status el valor real que toma en cada instante el led o el switch respectivamente. Con esto, si se selecciona el nivel de intensidad 100 para el led, se podrá comprobar la progresión para cada 1% de intensidad que está tomando el led (en el caso del switch ocurre lo mismo, aunque solo para el estado apagado o encendido ya que no cuenta con regulación por PWM).
- Búsqueda de actualizaciones. La búsqueda de nuevas actualizaciones se puede producir de tres formas distintas (cada vez que se inicia la placa ESP, si se mantiene más de dos segundos pulsado el botón flash o bien por MQTT especificándolo directamente o cada vez que cumple un cierto tiempo (de forma periódica). Es por ello por lo que hemos formalizado otro topic (infind/GRUPO2/ESPX/actualizacion) que hace saber al usuario que se va a producir una búsqueda de actualizaciones al igual que el momento en que se actualiza. También hace constancia del método en que se ha buscado la actualización, de forma que el usuario sea consciente de los cambios que se hayan podido producir por parte de él mismo o de otro usuario.

### 3.5.2. Inicializaciones

Inicialmente, en caso de intentar emplear el código con una placa ESP32 se hace saber al usuario que el funcionamiento no será válido, de forma que se produzca un error y se seleccione así la ESP8266.

### 3.5.3. Sensores

Se cuenta con un total de 4 placas ESP8266, cada una con un número de sensores distintos. Para evitar inicializar todos los sensores, se ha asignado a cada placa un nombre identificador en función del CHIPID de la ESP que corresponda. Las posibles opciones son las siguientes:

- El que cuente con la asignación de “salon” podrá usar los 3 sensores ya comentados (DHT11, VL53L0X y MQ2).
- Si es asignado como “cocina”, los sensores de los que se dispondrá serán el de temperatura y humedad (DHT11) y el de calidad del aire (MQ2).
- Si es asignado como “dormitorio” o “wc”, el único sensor del que se hará uso será el DHT11.
- En caso de detectar otra placa que no sea de los 4 integrantes del grupo a priori será tomada como “habitación”, cuyo único sensor disponible será el DHT11.

#### **3.5.4. Anti-rebotes**

partir de un sencillo condicionante, se evitan las falsas pulsaciones que se puedan producir cuando se active el botón flash (GPIO0). El código es un simple 'if' que permite evitar los pulsos de menos de 50 ms.

### **3.6. Google Home**

#### **3.6.1. Introducción**

En este trabajo se ha usado la aplicación Home para controlar distintas cosas de la placa igual que conocer información sobre esta. Además, se le tiene conectada a la casa que hemos creado un dispositivo Google Home para controlar la aplicación con comando de voz.

#### **3.6.2. Dispositivos con Nora**

Para meter distintos dispositivos, en el que la información que maneja se envía desde o hacia Node-RED se usa la biblioteca Nora, que es la que hemos usado en Node-RED. Para meterla necesitamos configurar un dispositivo y darle a Funciona con Google y buscamos Nora, en este nos pedirá iniciar sesión, que usamos la misma cuenta con la que hemos cogido el token en la parte de Node-RED. Al añadirlo todos los nodos de Nora que se tengan en el Node-RED saldrán como dispositivos con sus respectivos nombres. Tenemos que meter los dispositivos en la habitación correspondiente. Cuando ya está configurado debería salir lo que aparece en la Figura 40.

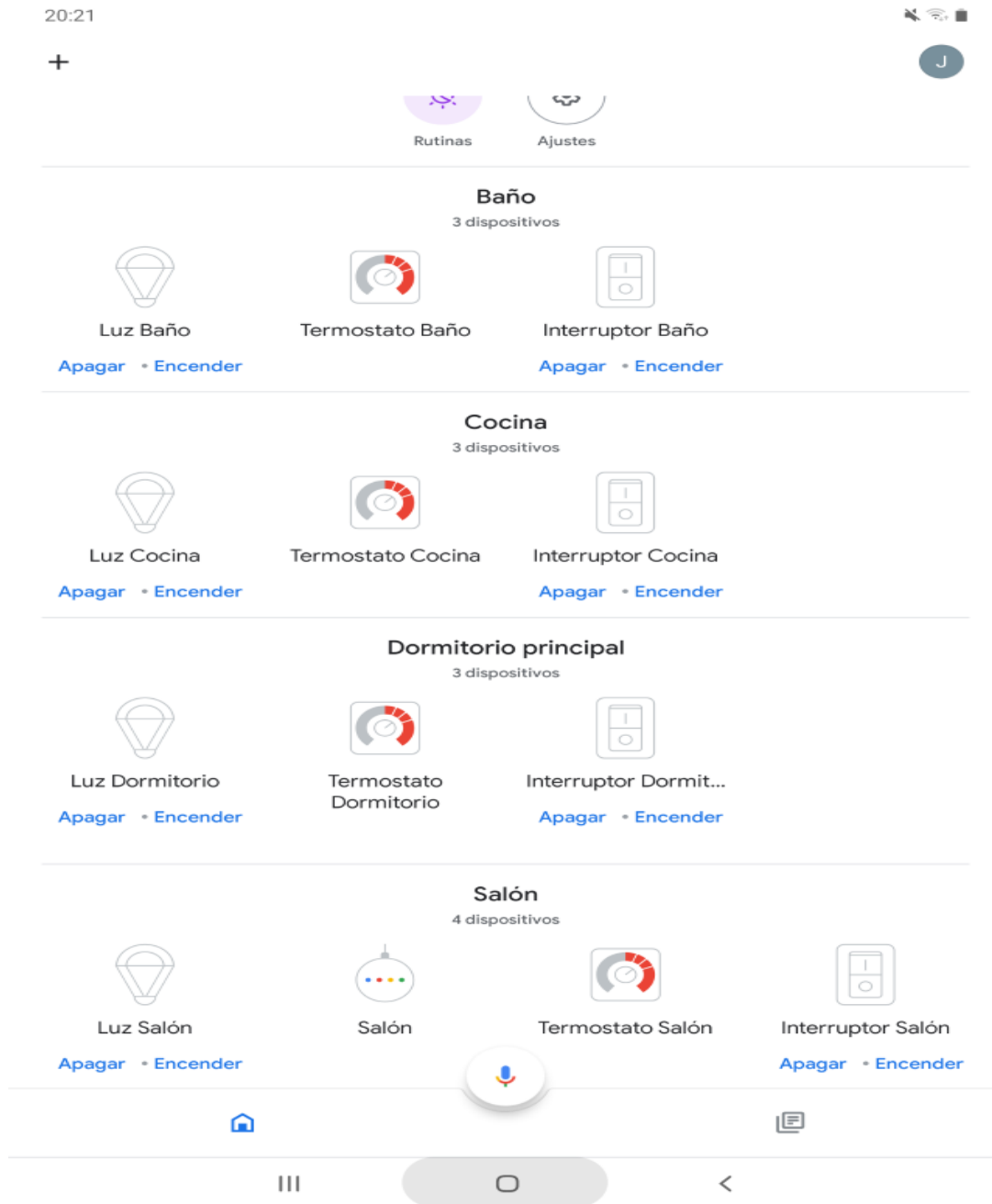


Figura 40. Interfaz de Home

Ahora vamos a explicar los distintos dispositivos, que siempre nos dirán el estado en el que están.

- Los dispositivos Bombillas, que son los leds con PWM. En este dispositivo nos sale un deslizador donde podemos indicar el porcentaje de brillo, además tenemos un botón para apagarlo y encenderlo. Nosotros le enviamos el estado del led, igual que él nos envía datos cuando lo cambia él, esto se hace desde Node-RED. Esto lo hemos puesto para cuando se cambie el brillo se encienda. La interfaz se puede ver en la Figura 41.

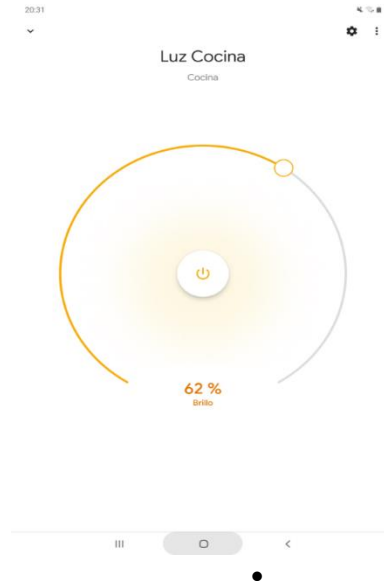


Figura 41. Interfaz de iluminación

- Los dispositivos de interruptor, que es el led que solo solo tiene encendido y apagado, la interfaz solo tiene un botón para apagarlo o encenderlo. Nosotros le enviamos el estado del led, igual que él nos envía datos cuando lo cambia él, esto se hace desde Node-RED. La interfaz se ve en la Figura 42.



Figura 42. Interfaz del Interruptor

- Ahora tenemos el termostato que es para saber la temperatura y la humedad. La interfaz tiene un deslizador para poner la temperatura del aparato, igual que un botón de encendido



y apagado, que el encendido es poniendo un modo (en nuestro caso hemos metido el modo calor y frio), esto no lo vamos a controlar desde aquí ya que se apagará o encenderá a 22°C, eligiendo el modo de encendido, cuando se haya superado temperatura límites (superior e inferior) que hayamos puesto en el Dashboard. Lo que nos importa de aquí es que marca la temperatura en grados centígrados y el porcentaje de humedad encima del botón y debajo del deslizador. Los datos que coge igual que el modo se hace desde el Node-RED. La interfaz se puede ver en la Figura 43.



Figura 43. Interfaz termostato

## 4. MANUAL DE USUARIO

Este manual le permitirá aprender a utilizar todas las funcionalidades básicas de la aplicación.

### 4.1. Dashboard del Node-RED

La interfaz de la aplicación viene establecida en Node-RED y cuenta con diferentes tabs, con los que se pueden organizar la información y controlar diferentes parámetros de la vivienda.

#### 4.1.1. ¿Cómo puedo ver los datos actuales?

Mediante los diferentes tabs de la aplicación, acceda a “Panel de control”, en el cual podrá ver de forma general gráficas, información y controles básicos de cada área de la vivienda. Ver Figura 44 y 45.

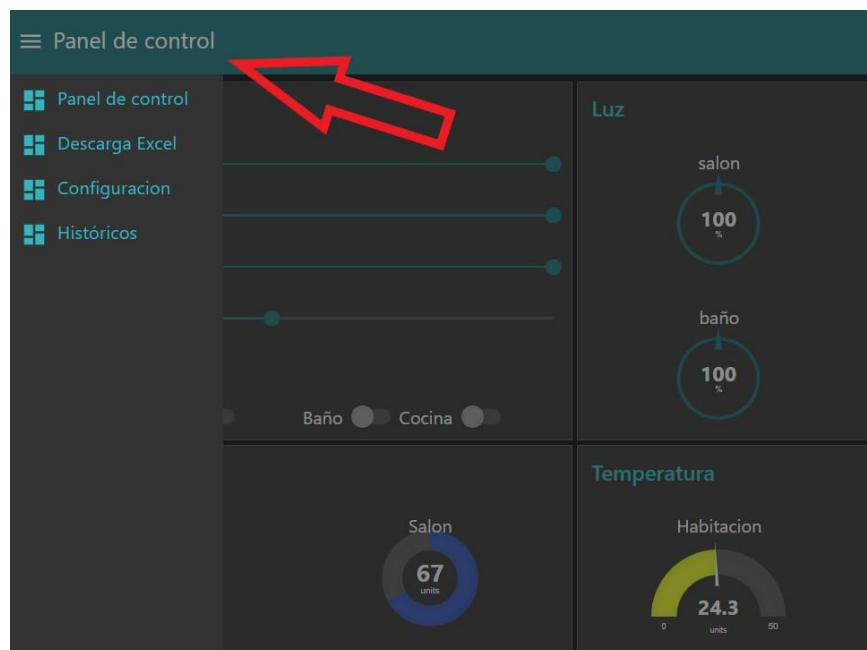


Figura 44. Selector de TABS

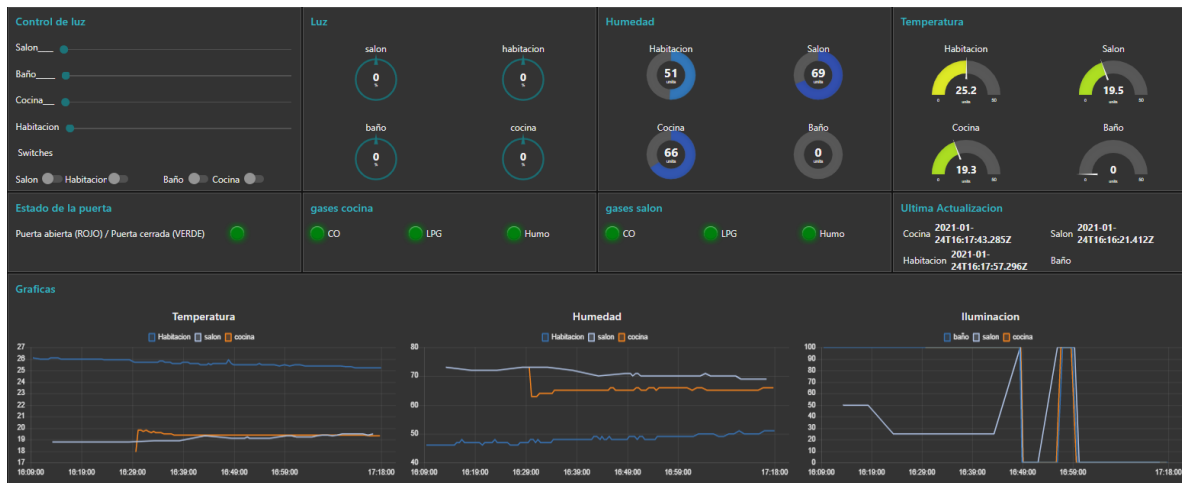


Figura 45. Panel de Control

En la sección de “Gráficas”, en la parte inferior de la pantalla, puede ver el cambio que ha tenido la humedad, temperatura e iluminación de todas las áreas de la vivienda a lo largo del tiempo. Ver Figura 46.



Figura 46. Panel de control (Gráficas)

Por otro lado, en “Última actualización” puede ver información importante como la última actualización de las diferentes áreas de la vivienda. Ver Figura 47.

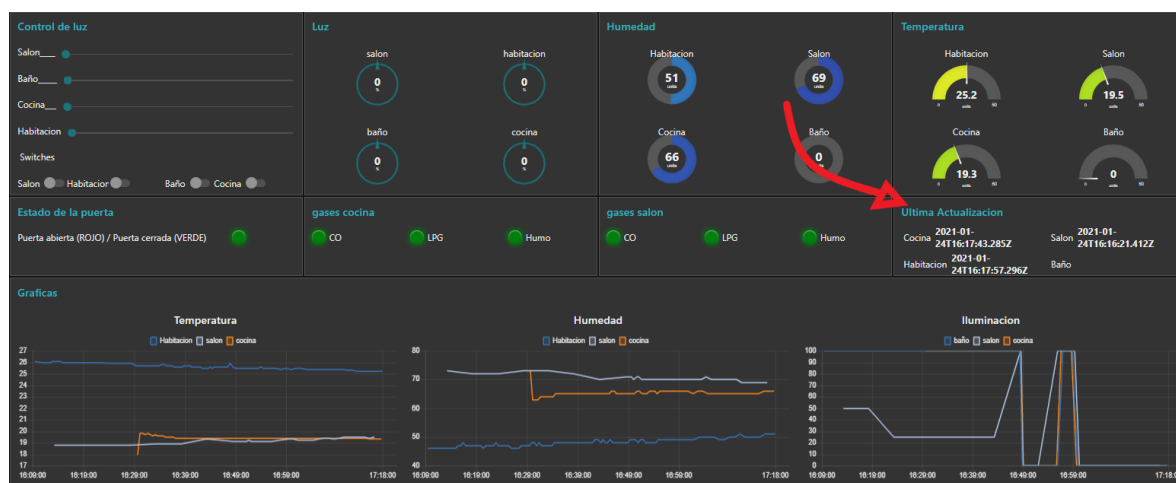


Figura 47. Panel de Control (Última actualización)

En esta ventana también puede ver los niveles de CO<sub>2</sub>, Humo y Gas natural (LPG) que hay en la cocina y el salón, indicado con unos leds el nivel, siendo verde, amarillo, naranja y rojo los niveles de peligrosidad de estos. Además del estado de la puerta, verde para cerrada y rojo para abierta. Ver Figura 48.



Figura 48. Panel de Control (Estado puerta y Gases)

Desde el panel de control también puede configurar las opciones básicas de esta aplicación como pueden ser: los switches y leds independientes de las diferentes áreas.

En la parte superior de la pantalla, podrá ver indicadores de temperatura, humedad e iluminación actuales. Ver Figura 49.



Figura 49. Panel de Control (Indicadores)

#### 4.1.2. ¿Cómo puedo seleccionar el área de la vivienda que quiero controlar?

En el recuadro de control dentro del selector de tabs “panel de control” se puede interactuar con los diferentes switches y leds de cada área de la vivienda. Para modificar la intensidad del led a controlar, arrastre con el cursor el slider, podrá modificar el valor entre un intervalo de 0 a 100.

Para cambiar el estado de los switches, con un clic en el switch es posible cambiar el estado independiente de cada área. Ver Figura 50.



Figura 50. Panel de Control (Control de luz y switch)

Para acceder a un control más específico de los parámetros de la aplicación, acceda al selector de tabs, en el inicio, a continuación, seleccione “configuración”. Ver Figura 51 y 52.

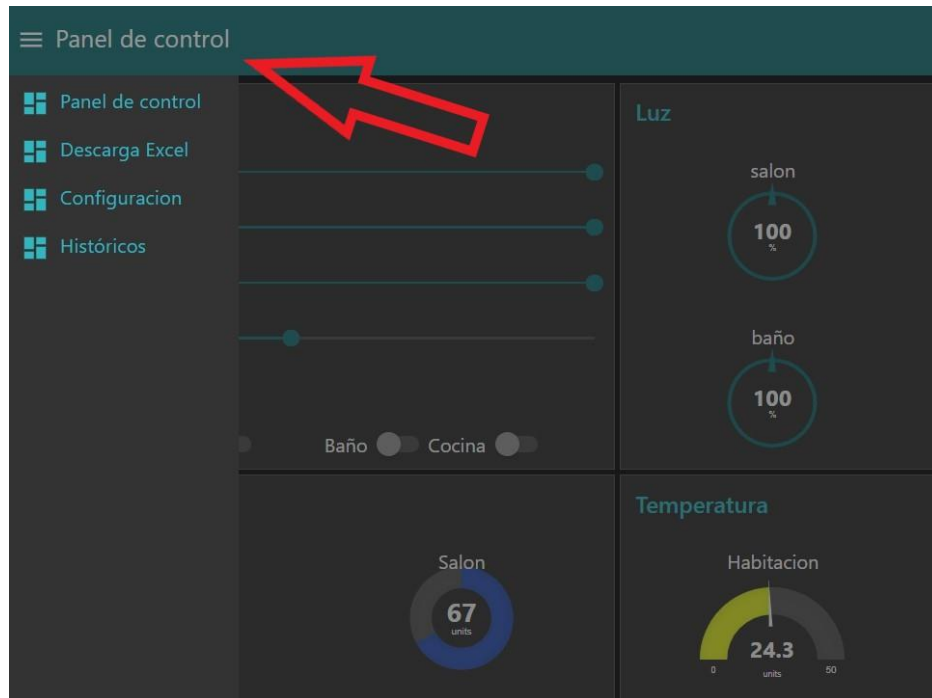


Figura 51. Selector de TABS

<b>Tiempo de pwm</b>			<b>Switch y Led</b>	
5 MS	10 MS	100 MS	Switch global	ENCENDER APAGAR
200 MS		500 MS	Led global	ENCENDER APAGAR
<b>Recogida de datos</b>			<b>Actualización de datos</b>	
5 SEG	10 SEG	30 SEG	MANUAL	60 MIN
1 MIN		5 MIN	120 MIN	240 MIN
<b>Temperaturas límite</b>				
Temperatura límite inferior			Temperatura límite superior	
<b>Humedades</b>				
Humedad límite inferior			Humedad límite superior	

Figura 52. Configuración

En esta ventana podrá modificar haciendo clic con el ratón que tiempo de recogida de datos desea, tiempo de PWM y tiempo de actualización de datos. Además, también puede escoger la opción “manual” en el recuadro de *Actualización de datos*.

Por otra parte, en el recuadro de *switch y Led*, puede encender y apagar todos los switches y leds de la vivienda de forma síncrona.

#### 4.1.3. ¿Cómo puedo cambiar los valores máximos y mínimos de la alarma?

Para cambiar los parámetros de la alarma, mediante el selector de tabs, seleccione configuración, y a continuación, en la parte inferior de la pantalla introduzca los límites de temperatura y humedad, siendo temperatura un valor en grados centígrados y la humedad un porcentaje. Ver Figura 53 y 54.

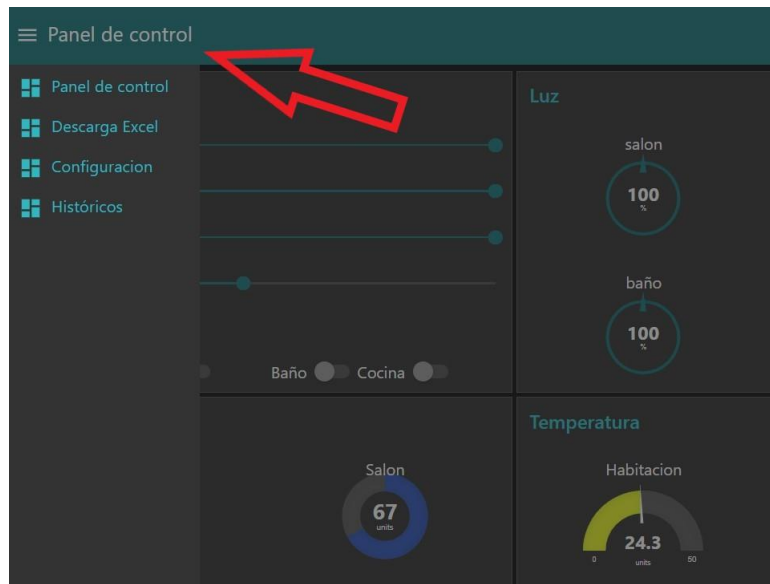


Figura 53. Selector de TABS

<b>Tiempo de pwm</b>			<b>Switch y Led</b>	
5 MS	10 MS	100 MS	Switch global	ENCENDER APAGAR
200 MS	500 MS		Led global	ENCENDER APAGAR
<b>Recogida de datos</b>			<b>Actualizacion de datos</b>	
5 SEG	1 SEG	30 SEG	MANUAL	60 MIN
1 MIN	5 MIN		120 MIN	240 MIN
<b>Temperaturas límite</b>				
Temperatura limite inferior			Temperatura limite superior	
<b>Humedades</b>				
Humedad limite inferior			Humedad limite superior	

Figura 54. Configuración (Alarma)



#### 4.1.4. ¿Cómo puedo ver un registro histórico de los datos recogidos?

Para acceder a un registro histórico de la vivienda, seleccione con el cursor en el selector de tabs *Históricos*, a continuación, en la esquina superior izquierda de la pantalla podrá ver un recuadro llamado “Selección de Fechas”, mediante dos desplegables podrá escoger la fecha de inicio y final que desea consultar. Ver Figura 55 y 56. También vemos la notificación de temperatura máxima.

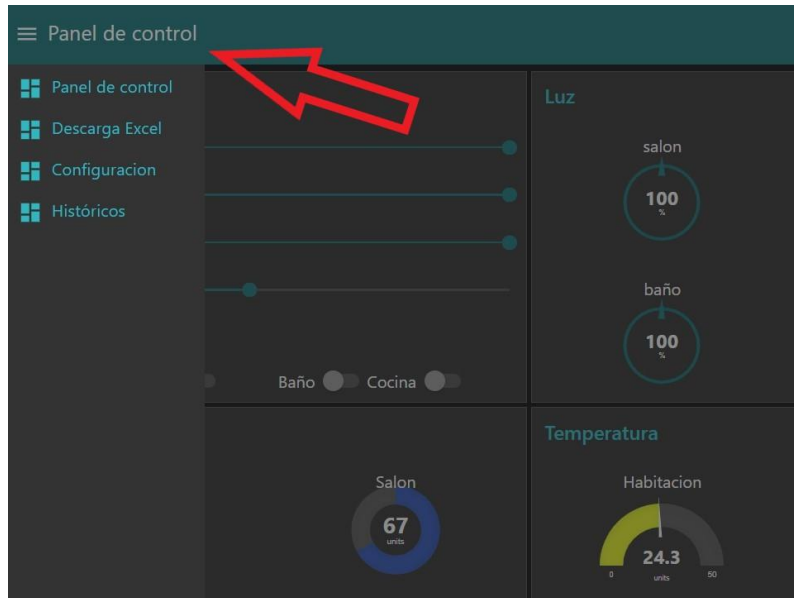


Figura 54. Selector de TABS

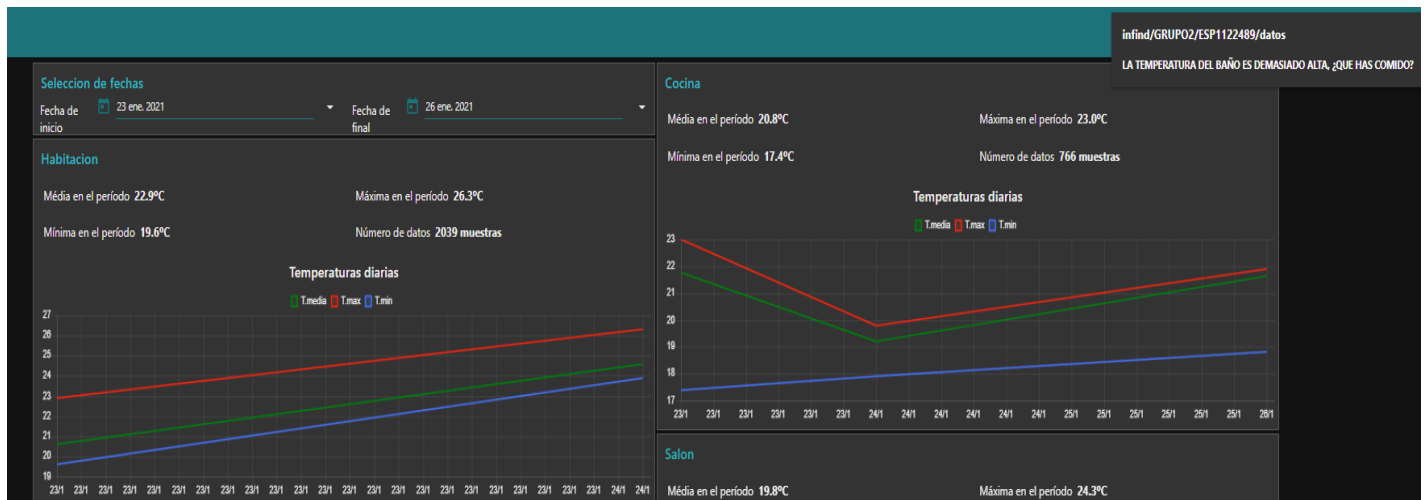


Figura 55. Histórico

#### 4.1.5. ¿Cómo puedo descargar los datos recuperados por una consulta en formato CSV?

Para hacer una descarga de una consulta, acceda a la ventana Descarga Excel, a continuación, en la parte superior derecha de la pantalla, en el recuadro “Consulta de datos a descargar” podrá seleccionar mediante dos desplegables la fecha de inicio y fin para su posterior descarga. Una vez seleccionada el rango de fechas a descargar, arrastre el cursor hasta el botón “Descarga registros en CVS (Excel)” y con un clic se descargará un fichero con los datos en el rango seleccionado. Ver Figura 56.

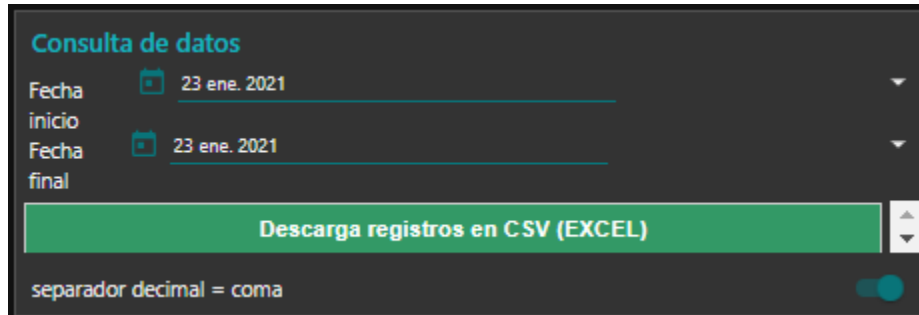


Figura 556. Descarga Excel

## 4.2. Conexión con el asistente de Google

### 4.2.1. ¿Qué dispositivos puedo tener?

Desde la aplicación Home se puede tener distintos dispositivos, estos son:

- Para los leds que se puede controlar el valor del brillo, este dispositivo se llama bombilla en Home. Desde la app se puede encender o apagar la luz y controlar el porcentaje de brillo del led, también se puede ver el estado de este. En nuestro caso, se conecta los leds del GPIO2 y el formato del nombre es: luz y el nombre de la habitación (ej. Luz Cocina).
- Para leds que solo tiene apagado y encendido, este dispositivo se llama interruptor en Home. Desde la aplicación se puede encender y apagar la luz, igual que ver el estado de este. En nuestro caso, se conecta los leds del GPIO16 y el formato del nombre es: switch y el nombre de la habitación (ej. Luz Cocina).
- Para el dispositivo termóstato. En este dispositivo se puede conocer el valor de la temperatura en grados y el porcentaje de humedad. Además, estos dispositivos se pueden conectar con un sistema de calefacción o de aire acondicionado, entonces cuando se llegue a la temperatura mínima o máxima (que se define en el Dashboard), se activa respectivamente el aparato conectado, también se puede hacer de manera individual. En nuestro caso, se conecta con el sensor DHT11 y el formato del nombre es: Termostato y el nombre de la habitación (ej. Luz Cocina).

#### 4.2.2. ¿De dónde son los dispositivos?

Desde la aplicación nos dice dónde están los dispositivos como se puede ver en la Figura 57.

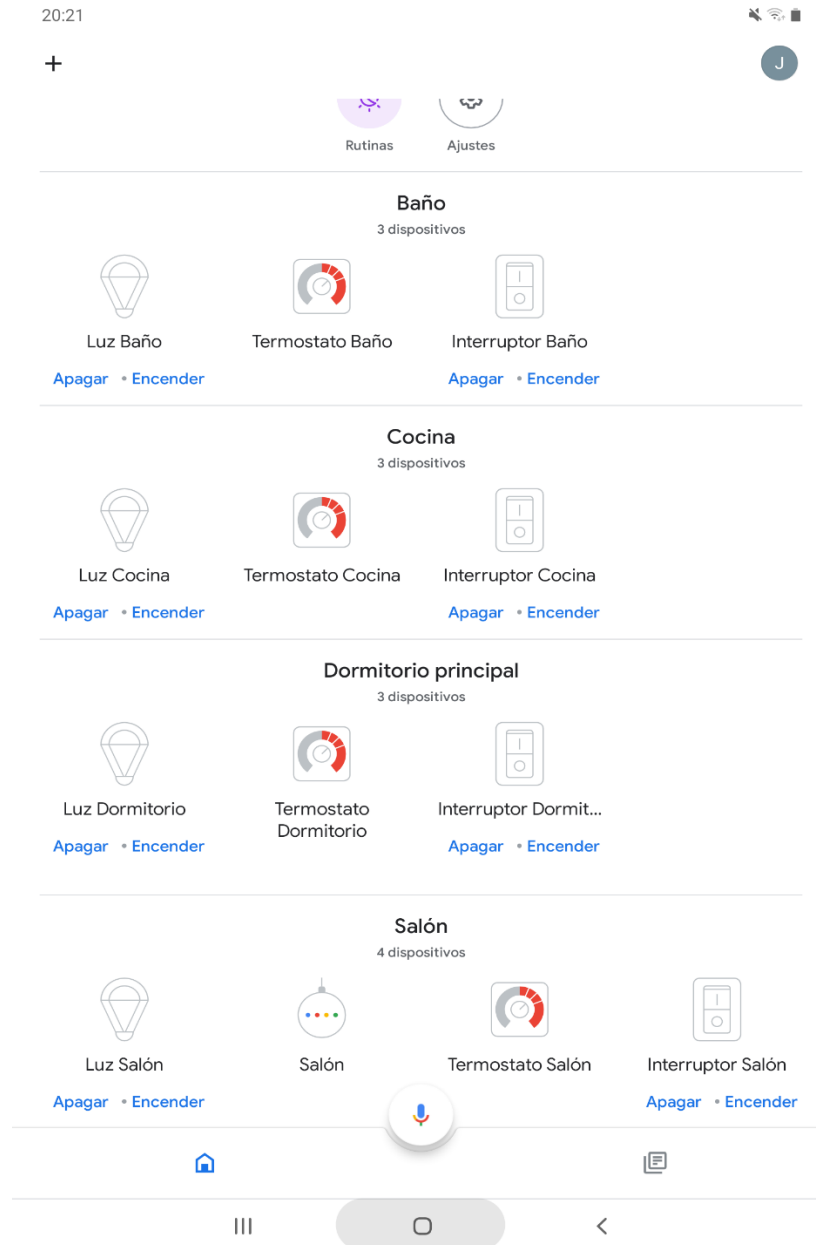


Figura 57. Interfaz Home

#### 4.2.3. ¿Cómo miro los datos de los dispositivos?

Para saber los datos de los dispositivos solo tenemos que pulsar sobre él.

Un posible problema es que puede que desde fuera no salga actualizado por lo que tendremos que cargar la pantalla arrastrando la pantalla hacia abajo cargándose de nuevo.

#### 4.2.4. ¿Qué significa cada cosa en los dispositivos y como le doy comandos?

- Los dispositivos Bombillas. En este dispositivo nos sale un deslizador donde podemos indicar el porcentaje de brillo, además tenemos un botón para apagarlo y encenderlo. El dispositivo está configurado para cuando se cambie el brillo se encienda. Donde se corta el círculo que forma el deslizador se muestra el porcentaje de brillo y cuando se encuentra en amarillo en vez de gris este encendido. La interfaz se puede ver en la Figura 58.

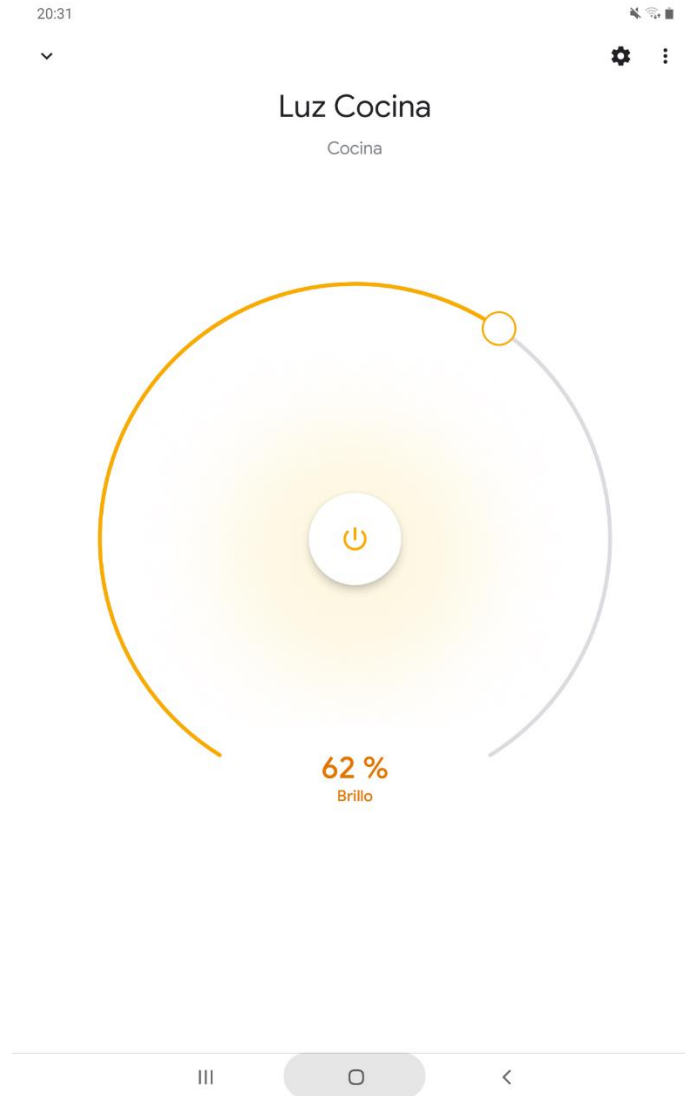


Figura 58. Interfaz Bombilla

- Los dispositivos de interruptor. La interfaz solo tiene un botón para apagarlo o encenderlo. Si este encendido se muestra un poco de azul alrededor del botón, si no sale nada esta apagado. La interfaz se ve en la Figura 59.

20:33



## Interruptor Dormitorio

Dormitorio principal



Figura 59. Interfaz Termostato

- Ahora tenemos el termostato. En este encima del botón encendido tendremos el valor de la temperatura en grados centígrados y el porcentaje de la humedad. También en medio del deslizador nos saldrá el estado del aparato que tengamos conectado en la habitación, este se cambia automáticamente con la temperatura límite que hemos puesto en DashBoard. La interfaz se ve en la Figura 60.

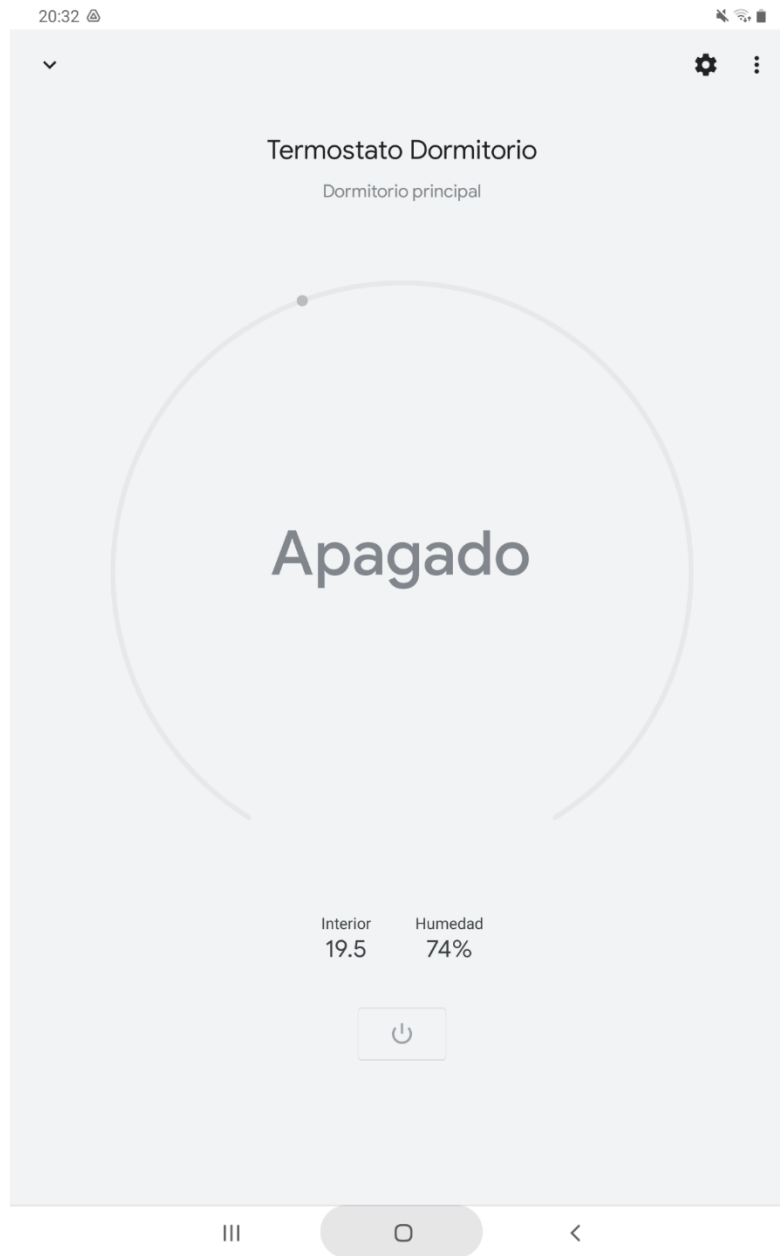


Figura 60. Interfaz Termostato

#### 4.2.5. ¿El Google Home puede darme datos?

Si tienes un dispositivo Google Home conectado en tu casa y tienes distintos dispositivos, se le puede preguntar sobre los distintos sensores, diciéndole “Ok Google” después lo que le quieres preguntar.

- Para el led que se le puede controlar el brillo, se le puede preguntar el brillo en el que está puesto, para ello se le tiene que decir:

“Ok Google dime el brillo de la (Nombre del dispositivo)”.

Ej. *“Ok Google dime el brillo de la Luz Cocina”.*

- Para el dispositivo termóstato, si se quiere saber la temperatura:

“Ok Google dime la temperatura del (Nombre del dispositivo)”. Esto te dará también el estado del aparato que tengas conectado.

Ej. *“Ok Google dime la temperatura del Termóstato Cocina”.*

Si se quiere saber la humedad:

“Ok Google dime la humedad del (Nombre del dispositivo)”.

Ej. *“Ok Google dime la humedad del Termóstato Cocina”.*

#### 4.2.6. ¿Puedo cambiar los valores de los dispositivos?

Si tienes un dispositivo Google Home conectado en tu casa y tienes distintos dispositivos, se le puede preguntar sobre los distintos sensores, diciéndole “Ok Google” después lo que quieres que haga.

- Para el led que se le puede controlar el brillo, se le puede cambiar el brillo en el que está puesto, para ello se le tiene que decir:

“Ok Google cambia el brillo de (Nombre del dispositivo) a X”.

Ej. *“Ok Google cambia el brillo de Luz Cocina a 25”. Esto cambia el brillo a 25%.”*

También se le puede decir que lo apague o encienda:

“Ok Google enciende/apaga la (Nombre del dispositivo)”

Ej. *“Ok Google apaga la Luz Cocina”. Esto apaga la Luz Cocina.*

- Para el led que solo tiene encendido y apagado, para controlar el estado se le tiene que decir:

“Ok Google enciende/apaga el (Nombre del dispositivo)”

Ej. *“Ok Google apaga el interruptor”*. Esto apaga la Interruptor Cocina.

- Para el dispositivo termóstato, se puede poner la calefacción o el aire acondicionado, para ello se le tiene que decir:

“Ok Google pon en modo calor/frio el (Nombre del dispositivo) a X”.

Ej. *“Ok Google pon en modo calor el Termóstato Cocina a 21”*. Esto te lo pone en modo calefacción y lo pone a 21°C.

### 4.3. Bot de Telegram

Mediante la aplicación Telegram, también es posible ver el estado de la vivienda y controlar sus diferentes parámetros.

#### 4.3.1. ¿Cómo puedo tener acceso al bot de Telegram?

Para poder usar el bot de Telegram lo único que necesita es estar en el grupo de Telegram. Una vez ahí, usando comandos puede ver el estado de las diferentes áreas de la vivienda.



#### 4.3.2. ¿Qué comandos puedo usar con el bot?

Con el comando `/start` se inicia el bot, una vez iniciado se muestran por pantalla tres botones, *encender*, *apagar*, *ver estado* y *tutorial*. Para una explicación detallada de los comandos que puede usar arrastre el cursor hasta el botón *tutorial* y a continuación el bot le mostrará por pantalla una breve explicación de la funcionalidad de cada comando. Ver Figura 61.



Figura 61. Telegram Bienvenida (Tutorial)

#### 4.3.3. ¿Cómo puedo ver información de la vivienda desde el bot de Telegram?

El bot de Telegram está pensado como un semejante de la interfaz de Node-RED, por lo tanto, cumple las mismas funciones.

Una vez iniciado el bot con el comando `/start`, arrastre el cursor hasta el botón “Ver estado”, una vez pulsado se mostrará por pantalla botones indicando el área de la vivienda del que desea obtener información sobre el estado en el que se encuentra. También es posible acceder a esta área con el comando `/estado`. Ver Figura 62.

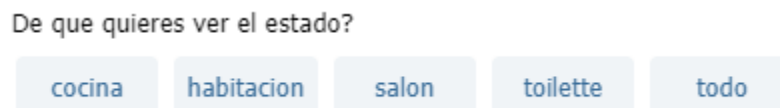


Figura 62. Telegram (Estado)

Una vez seleccionada el área a consultar, se mostrará por pantalla el voltaje de la placa, temperatura, humedad, iluminación y última actualización del área seleccionada. Ver Figura 63.

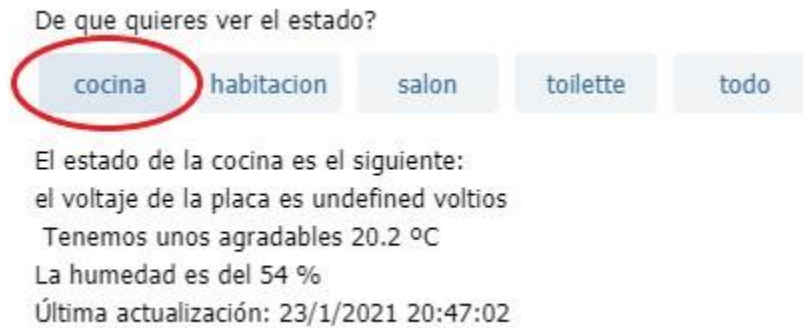


Figura 63. Telegram (Estado cocina)

Por otra parte, también puede ver información sobre un pronóstico del tiempo haciendo clic en el botón “Predicción”. Otra forma de acceder a esta área es con el comando `/prediccion`. Ver Figura 64.

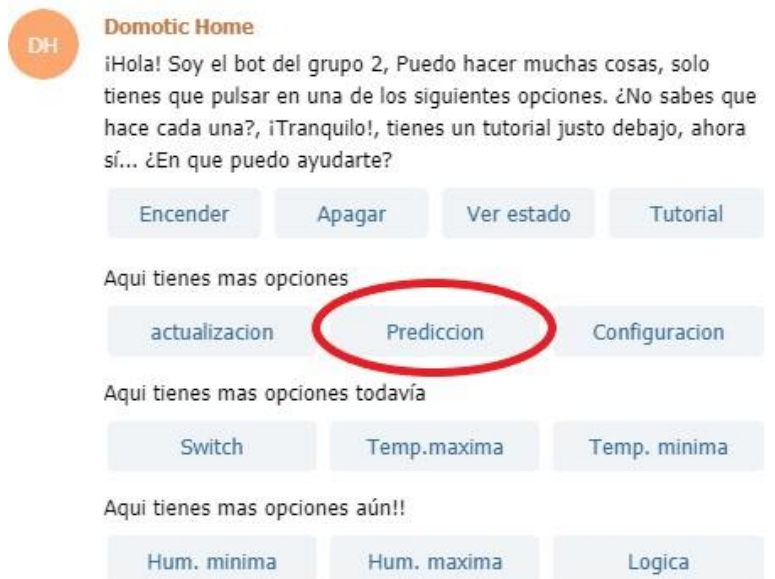


Figura 64. Telegram (Predicción)

Una vez pulsado el botón predicción, se mostrará por pantalla las opciones disponibles a consultar. Ver Figura 65.

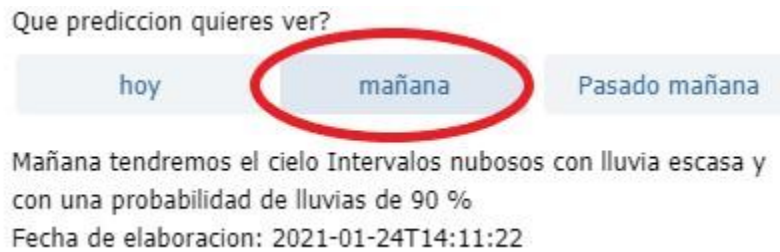


Figura 65. Telegram (Predicción Mañana)

#### 4.3.4. ¿Qué tipo de configuración puedo hacer con el bot de Telegram?

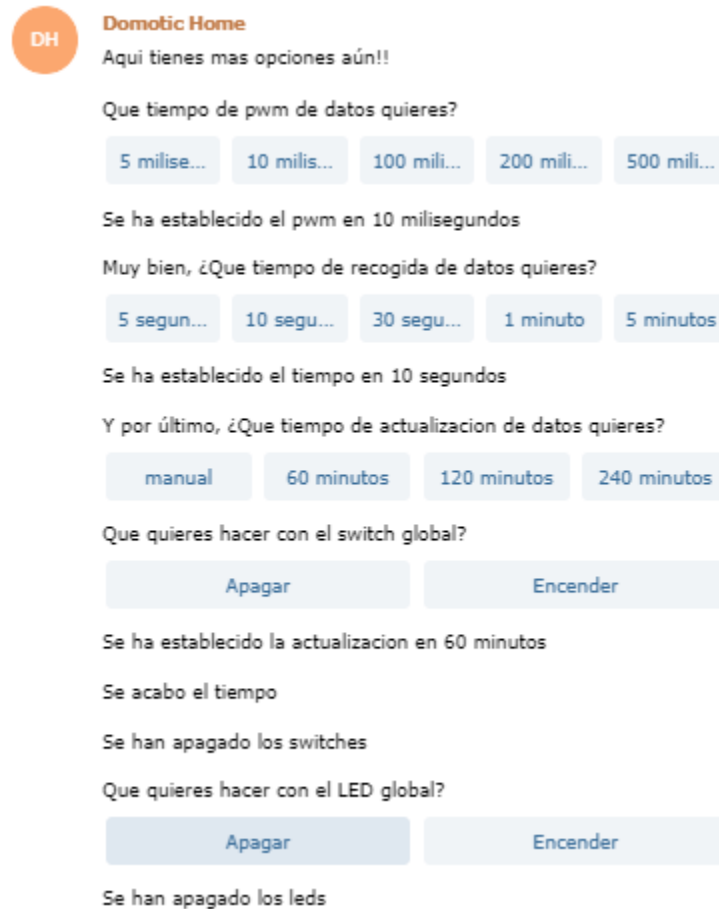
Para acceder a una configuración de parámetros de la vivienda desde el bot de Telegram, una vez iniciado el bot con el comando `/start`, arrastre el cursor hasta el botón “Configuración” y una vez pulsado se mostrará por pantalla una secuencia de parámetros a seleccionar. Podemos ver el inicio del bot en la Figura 66. Los parámetros a configurar son:

- Tiempo PWM.
- Tiempo de recogida de datos.
- Tiempo de actualización de datos.
- Encendido y apagado de todos los leds (LED Global).
- Encendido y apagado de todos los switches (switch Global).



Figura 66. Telegram Bienvenido (Configuración)

Una vez pulsado el botón de configuración, se mostrará secuencialmente por pantalla todas las configuraciones con las opciones disponibles. Ver Figura 67.



**DH** **Domotic Home**  
Aquí tienes mas opciones aún!!

Que tiempo de pwm de datos quieres?

5 milise... 10 milis... 100 mili... 200 mili... 500 mili...

Se ha establecido el pwm en 10 milisegundos

Muy bien, ¿Que tiempo de recogida de datos quieres?

5 segun... 10 segun... 30 segun... 1 minuto 5 minutos

Se ha establecido el tiempo en 10 segundos

Y por último, ¿Que tiempo de actualizacion de datos quieres?

manual 60 minutos 120 minutos 240 minutos

Que quieres hacer con el switch global?

Apagar Encender

Se ha establecido la actualizacion en 60 minutos

Se acabo el tiempo

Se han apagado los switches

Que quieres hacer con el LED global?

Apagar Encender

Se han apagado los leds

Figura 67. Telegram (Configuración)

Para configurar la alarma por temperatura o humedad no es necesario acceder a “Configuración”, desde el menú de inicio, después del comando `/start`, pulse con el cursor el botón “Temp. Máxima” o “Temp. Mínima” para cambiar el rango de la alarma por temperatura. Lo mismo puede hacer con el rango de la alarma por humedad. Ver Figura 68.

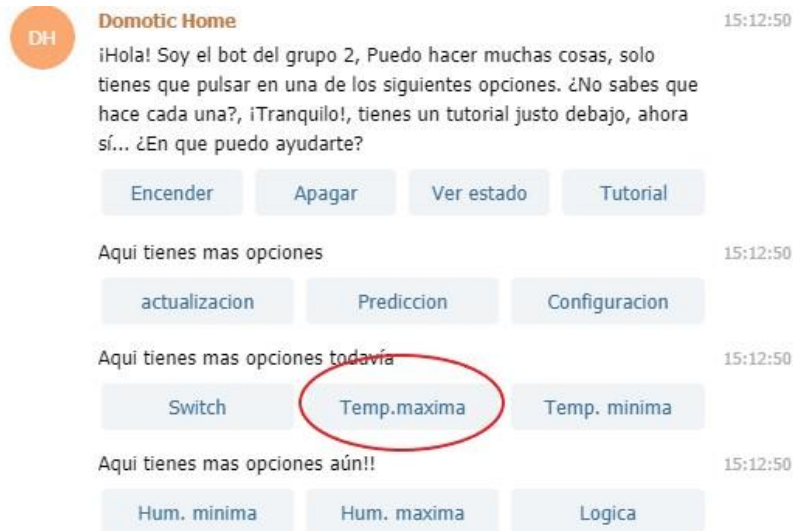


Figura 68. Telegram Bienvenida (Temp.Máxima)

Una vez seleccionado que parámetro de la alarma que desea modificar, se mostrará por pantalla las opciones disponibles. En el caso de la temperatura máxima para que se active la alarma se mostrarían las siguientes opciones. Ver Figura 69.

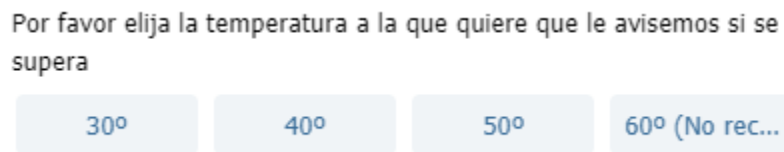


Figura 69. Temp.Máxima

#### 4.3.5. ¿Qué puedo controlar desde el bot de Telegram?

Desde el bot de Telegram puede controlar el apagado, encendido e iluminación de los leds de las diferentes áreas de la vivienda de manera global o individual. Además, también puede apagar y encender los switches de manera global o individual.

Para controlar estos parámetros desde el bot, una vez iniciado con el comando `/start`, arrastre el cursor hasta el botón “Encender”. Ver Figura 70.

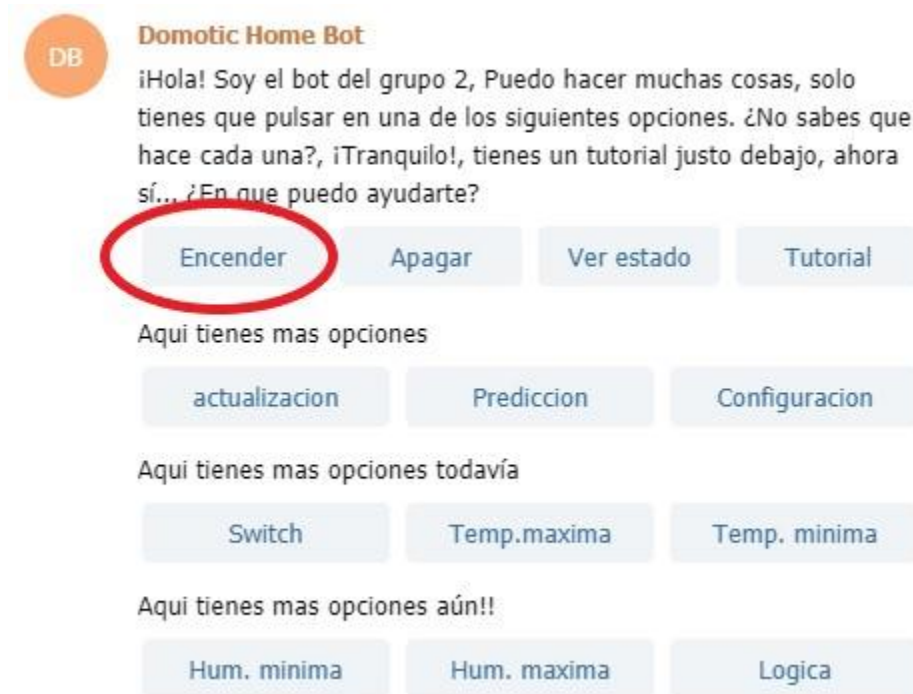


Figura 70. Telegram Bienvenida (Encender)

Una vez pulsado podrá elegir el área, se mostrará por pantalla las opciones de encendido que tiene disponible. Ver Figura 71.



Figura 71. Telegram (Encender cocina)

Para encender o apagar los switches, en el menú principal seleccione con el cursor la opción “Switches”. Ver Figura 72.

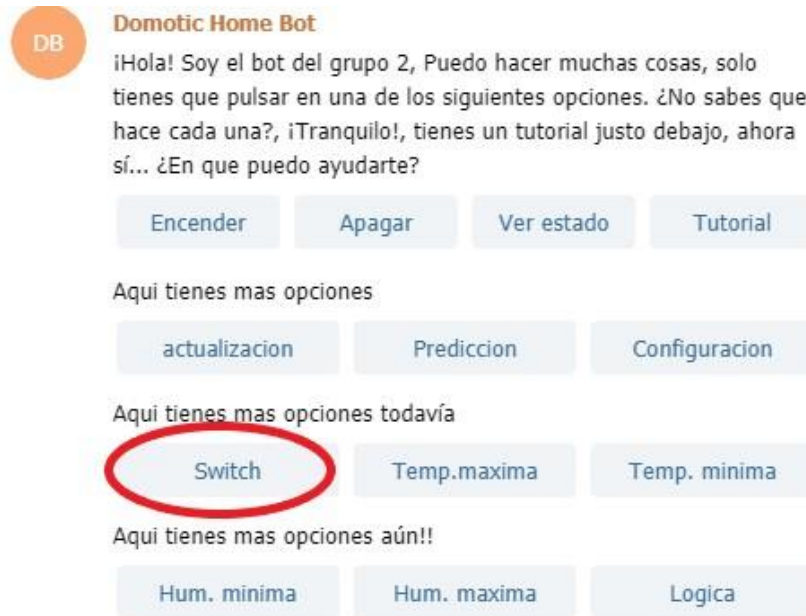


Figura 72. Telegram Bienvenida (Switch)

Una vez pulsado, se mostrará por pantalla que área de la vivienda se desea controlar, y las opciones disponibles. Ver Figura 73.

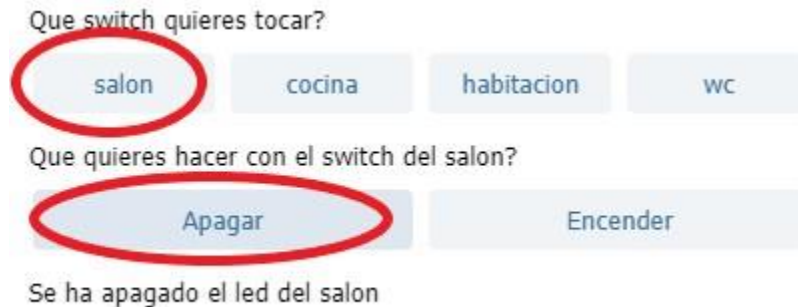


Figura 73. Telegram Switch (Salón)



## 5. RESULTADOS Y CONCLUSIONES

Durante el trayecto de este proyecto hemos podido aprender diversos aspectos relacionados con la plataforma de Node-RED y la programación en Arduino. Al haber enfocado el proyecto en la parte domótica de una vivienda, hemos podido comprender más el funcionamiento e importancia que puede tener la tecnología e informática en el uso diario. La domótica ofrece muchas ventajas, aparte de la comodidad, pudimos implementar seguridad, siendo como un ejemplo el control y detección mediante sensores de fugas de gas en la vivienda.

Este proyecto nos ha permitido tener una noción más cercana del mundo laboral, hemos tenido que desarrollar un enfoque más comercial poniéndonos en la piel del usuario final para así poder diseñar la plataforma con la que este se encontraría, tanto en Node-RED como en Telegram.

Por último, la experiencia de trabajar en equipo en este proyecto nos ha ayudado a tener una visión más clara de cómo distribuir funciones, planificar objetivos y llevar un registro de las actividades realizadas con ayuda de plataformas como GitHub y Trello.

## 6. LISTA DE FICHEROS

### 6.1. Ficheros Arduino

- *“Grupo2\_Arduino.ino”*, en este archivo está el código Arduino que se le programa a las placas ESP8266. El código es el mismo para todos excepto que hay que cambiar el WiFi y la contraseña, igual que la dirección IP propia de cada usuario.

### 6.2. Ficheros Node-RED

- *“GRUPO2\_NodeRed.json”*, en este archivo esta las conexiones del Arduino con el Dashboard y el Telegram, es el fichero principal de Node-RED.
- *“GRUPO2\_NodeRed\_Home.json”*, aquí está el Node-RED que conecta el Arduino con la aplicación Home.