

INSTALACIÓN DOMÓTICA CON ESP8266

PROYECTO FINAL DE INFORMÁTICA INDUSTRIAL

GRADO EN INGENIERÍA ELECTRÓNICA, ROBÓTICA Y MECATRÓNICA

CURSO ACADÉMICO

2020 - 2021

Autores: Frank Leonardo Luque Lineros

Javier López Soria

Juan José García Quiles

Leonardo García Guillen

Tutores: Dr. Andrés Rodríguez

D. Rubén Delgado Escaño

**TABLA DE CONTENIDO**

[**1.** **INTRODUCCIÓN Y OBJETIVOS** 1](#_Toc62242248)

[**2.** **DISEÑO DE HARDWARE** 1](#_Toc62242249)

[**2.1.** **Esquemas de conexión** 4](#_Toc62242250)

[**3.** **DISEÑO DE SOFTWARE** 5](#_Toc62242251)

[**3.1.** **NodeRed** 5](#_Toc62242252)

[**3.1.1.** **Introducción** 5](#_Toc62242253)

[**3.1.6.** **Nodos MQTT** 10](#_Toc62242254)

[**3.1.7.** **Nodos Telegram** 10](#_Toc62242255)

[**3.1.8.** **DashBoard** 11](#_Toc62242256)

[**3.2.** **Arduino** 11](#_Toc62242257)

[**3.2.1.** **Introducción** 11](#_Toc62242258)

[**3.2.2.** **Estructura** 11](#_Toc62242259)

[**4.** **RESULTADOS Y CONCLUSIONES** 15](#_Toc62242260)

[**5.** **MANUAL DE USUARIO** 15](#_Toc62242261)

[**5.1.** **Interfaz del NodeRed** 15](#_Toc62242262)

[**5.1.1.** **¿Cómo puedo ver los datos actuales?** 16](#_Toc62242263)

[**5.1.2.** **¿Cómo puedo seleccionar el área de la vivienda que quiero controlar?** 16](#_Toc62242264)

[**6.** **LISTA DE FICHEROS** 19](#_Toc62242265)

# **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.

1. **DISEÑO DE HARDWARE**

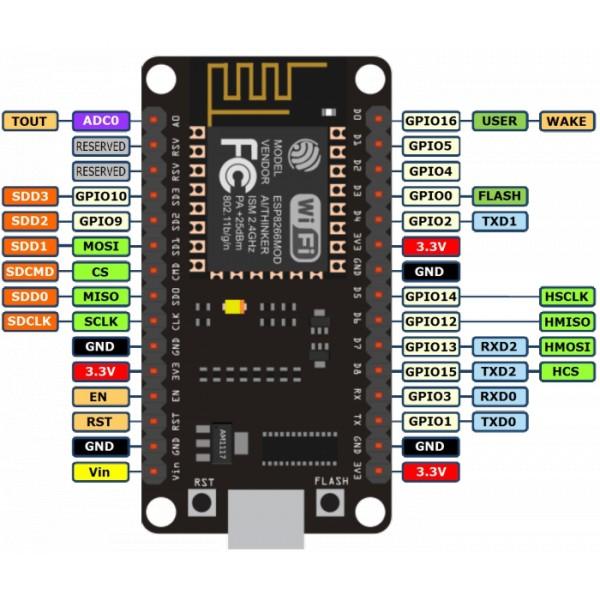
El hardware donde vamos a meter el programa Arduino es una placa ESP8266, Este es un chip “low-cost” con Wi-Fi. Tiene 16 pines GPIO, 1 conversor ADC de 10 bits y un pin GPIO2 dedicado a UART. Podemos ver la configuración de pines en la figura 1.

Figura 1. ESP8266

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

* EL ESP145 corresponde a la cocina.
* EL ESP47 corresponde al salón.
* EL ESPX corresponde al baño.
* EL ESPX corresponde a la habitación.

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

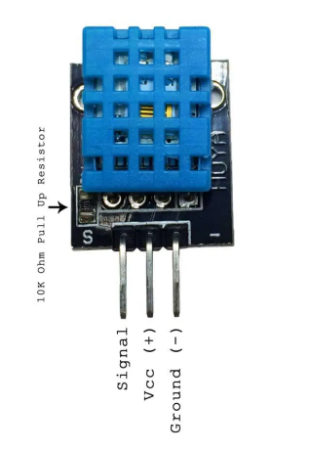
El sensor DHT-11 está conectado en todas las ESP y lo conectamos en el D1 la entrada digital y lo alimentamos con 3.3V y también lo conectamos a la tierra de la placa. Este sensor lo que hace es coger la temperatura (ºC) y la humedad (%). En la figura 2 vemos el sensor DHT-11.

Figura 2. Sensor 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.

Figura 3. Sensor MQ-2

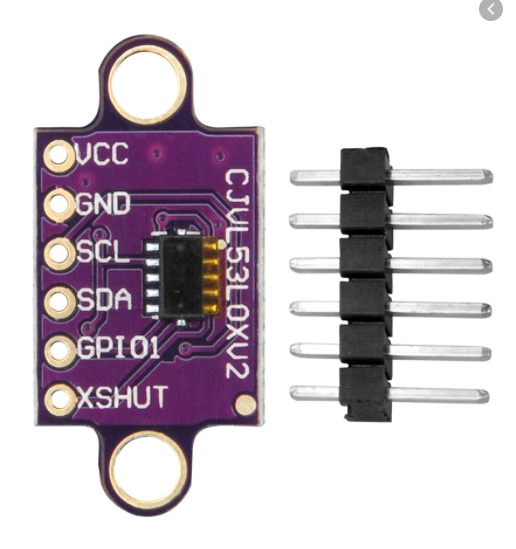
El sensor VL53L0X solo está conectado en la placa del salón y lo conectamos a la placa. Podemos verlo en la figura 4. Lo vamos a utilizar para comprobar si una puerta está abierta, observando cuando se reduce la distancia del infrarrojo. Lo vamos a utilizar para comprobar el estado de una puerta (si 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

* 1. **Esquemas de conexión**

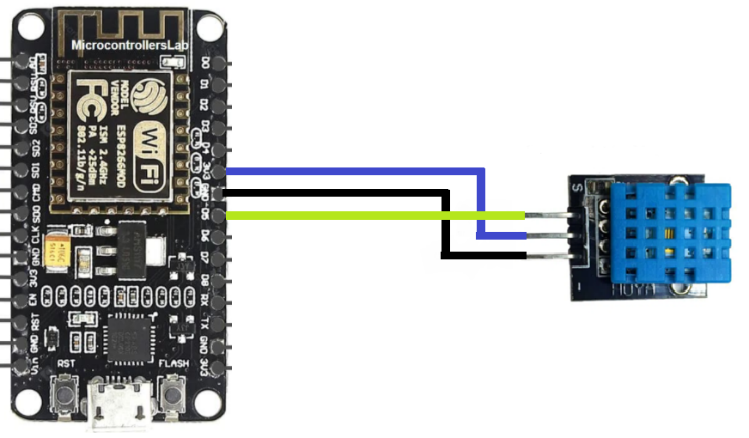
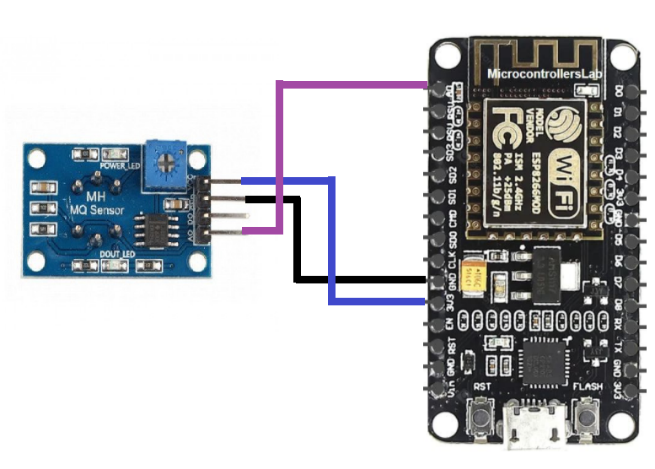
En esta sección se contemplará el interconectado hardware entre la ESP8266 con los distintos sensores de los que disponemos. Para todas las ESP se ha tenido en cuenta el mismo conexionado, evitando así cambiar las inicializaciones de pines dentro del código Arduino. Los pines vienen identificados y descritos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

Figura 5. Conexionado ESP8266 - DHT11

Figura 6. Conexionado ESP8266 – MQ2

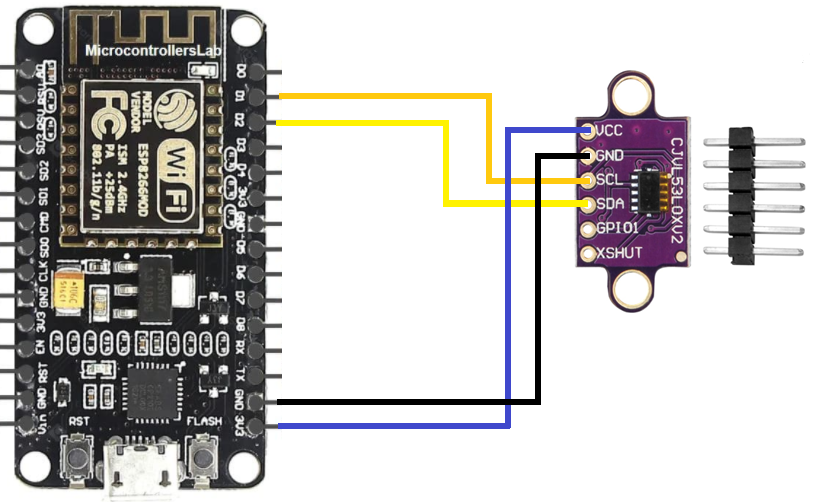


Figura 7. Conexionado ESP8266 – VL53L0X

1. **DISEÑO DE SOFTWARE**

(Texto)

* 1. **NodeRed**

(Más texto)

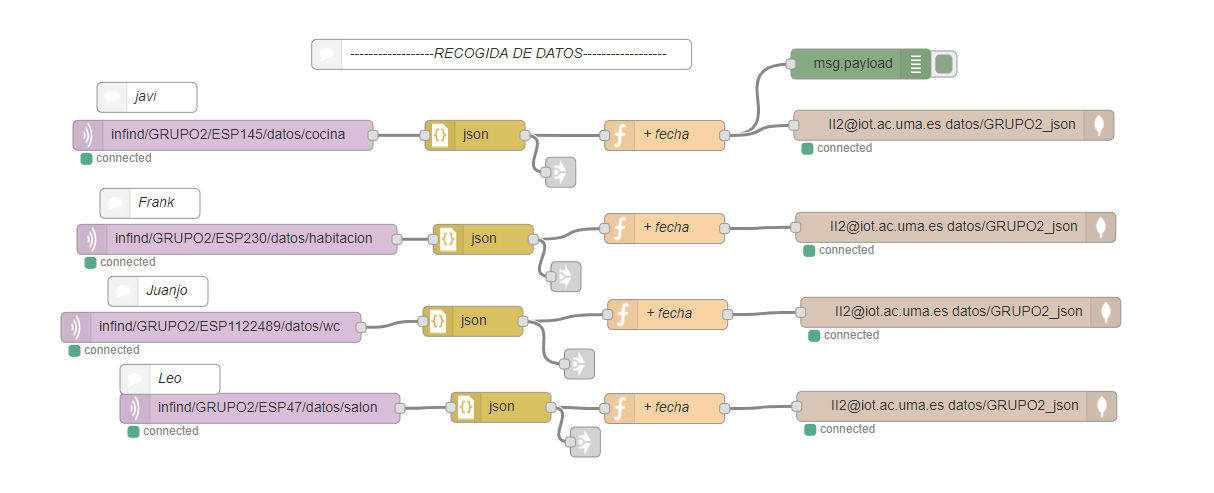
* + 1. **Introducción**

En este trabajo se ha utilizado NodeRed, una herramienta de programación basada en flujos y nodos. En NodeRed 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 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.

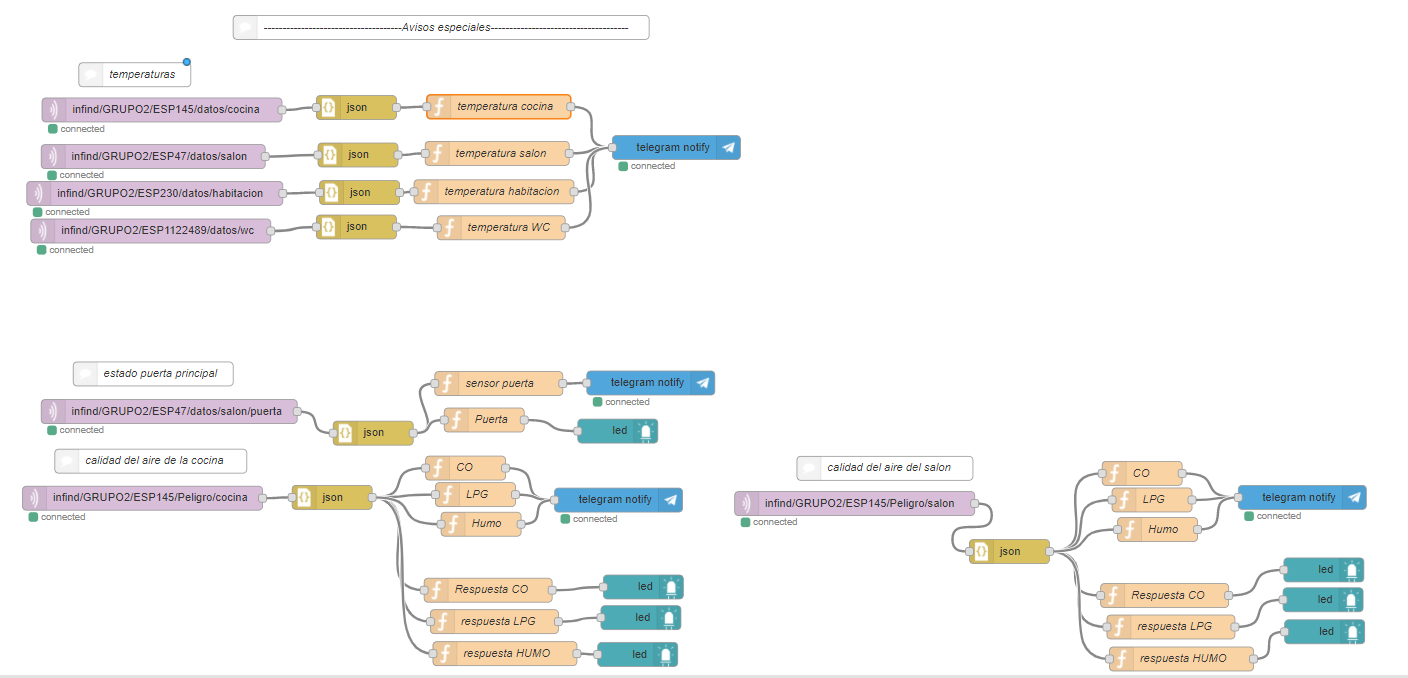
* + 1. **Recogida de datos**

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:



*Figura8. NodeRed recogida de datos*

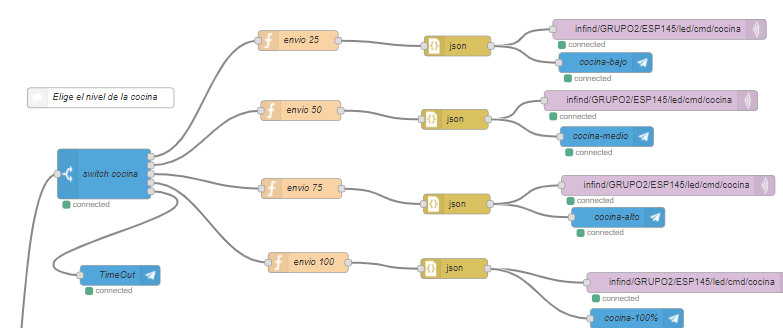
NodeRed, que tiene nodos que se **suscriben** a los topics anteriores. La información que se recibe por MQTT está formateada en JSON y se tendrá que convertir a “string” para trabajar con ella. Cada habitación cuenta con un topic por donde se ha enviado la información, a esta le hemos añadido la fecha 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.

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 más 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 9. NodeRed Avisos especiales*

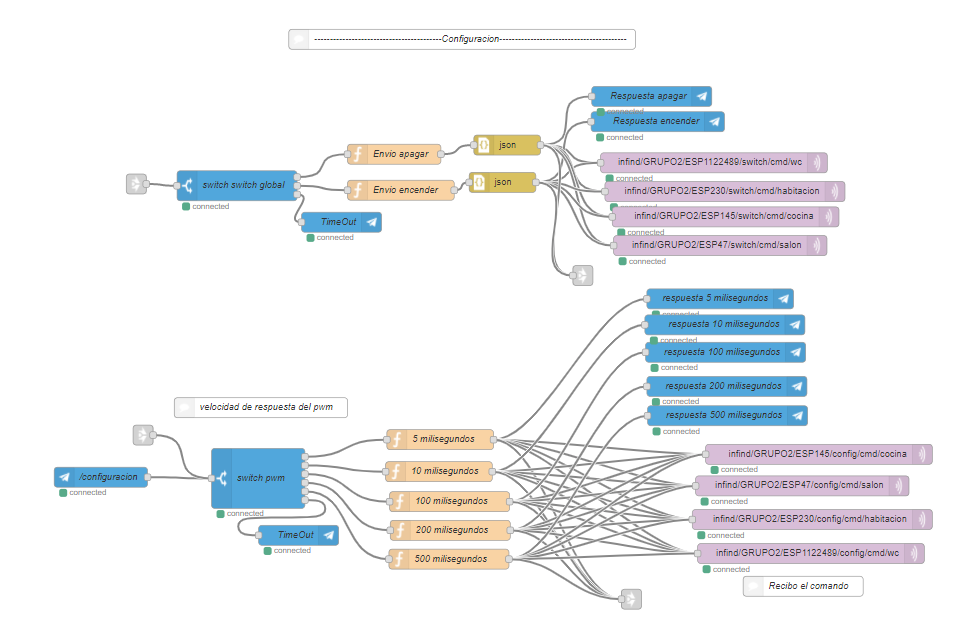
* + 1. **Envío de nivel de luminosidad del LED**

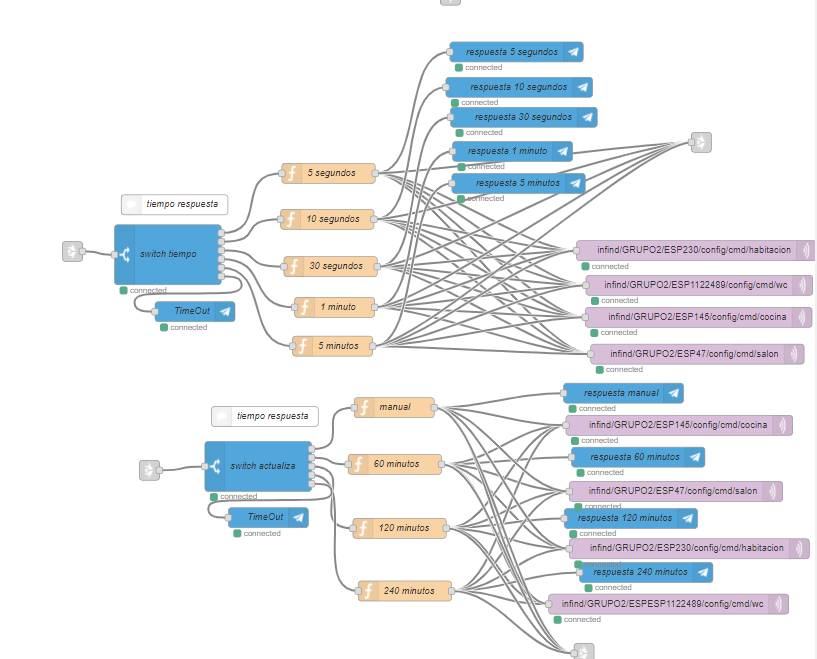
Cuando usamos Telegram para cambiar la intensidad del LED tenemos que enviar por MQTT el nivel elegido, para que la ESP8266 la recoja y actualice el valor de la intensidad. Enviamos estos datos por el topic infind/GRUPO2/ESPXX/led/cmd/YY, cambiando XX por el ChipId y la YY por la habitación que queramos.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

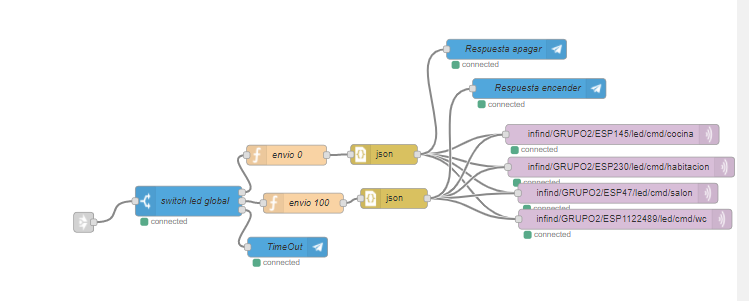
*Figura 10. Envío nivel LED*

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 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 11 podemos ver este nodo MQTT.







*Figura 11. MQTT de configuración*

* + 1. **Actualización FOTA y predicción**

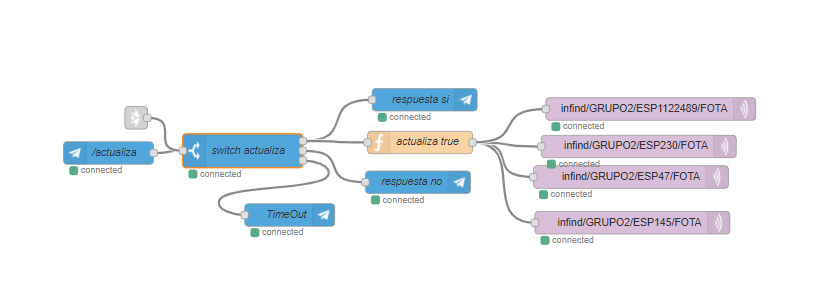
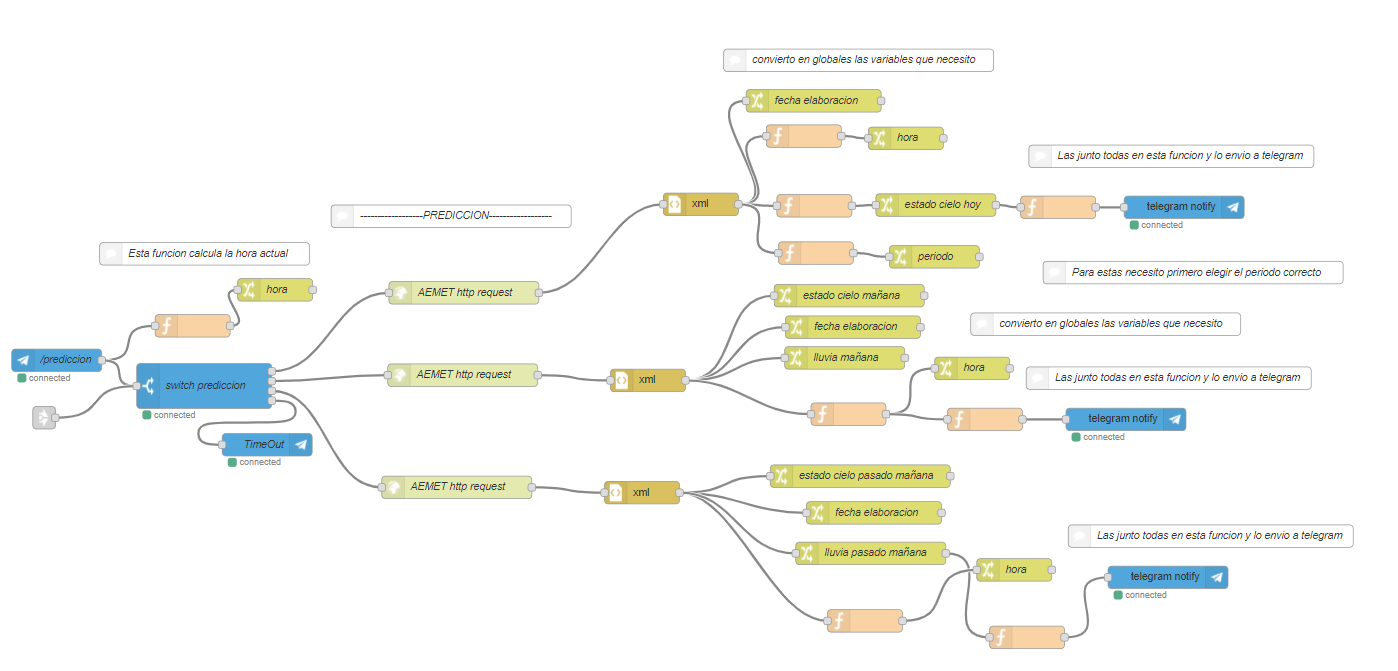
Para poder realizar actualizaciones FOTA se ha creado un flujo cuyo resultado se envía al topic que se ve en la figura 12.

Figura 12. 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 13

*Figura 13. Predicción tiempo*

* + 1. **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.

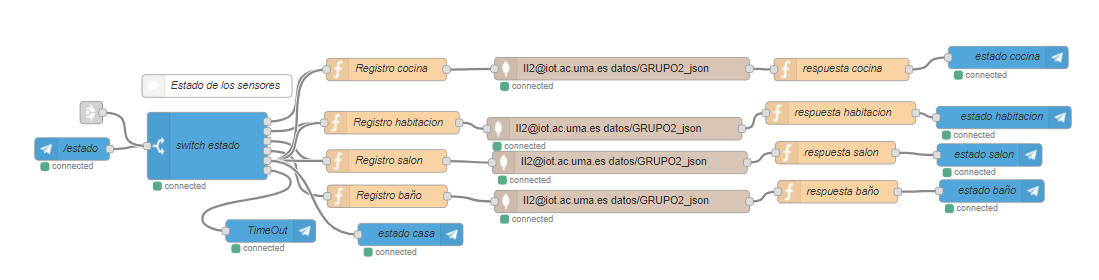
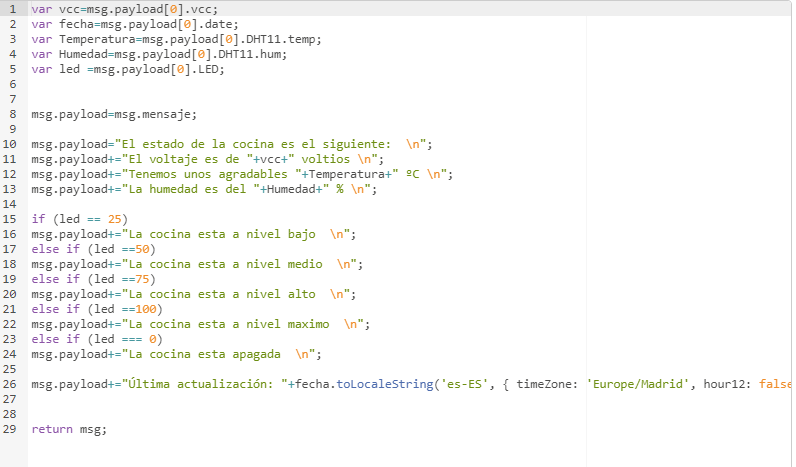


Figura14. Flujo de consulta de estado

El formato de la consulta tendrá el aspecto de la figura 15.

*Figura 15. Formato /consulta*

* + 1. **Nodos MQTT**

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.

*(INSERTAR IMAGEN RECOGIDA DE DATOS)*

Tenemos otros topics que recogen datos adicionales, como los gases contaminantes que hay en la cocina.

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.

* + 1. **Nodos Telegram**

(Texto)

* + 1. **DashBoard**

(Texto uwu)

* 1. **Arduino**

(TEXTOoo)

* + 1. **Introducción**

En este trabajo se ha utilizado Arduino IDE, una herramienta de programación basada en código c++. En Arduino IDE vamos a meterle el programa a la placa ESP8266 por lo que tendrá una conexión MQTT y internet, recoger datos de los distintos sensores, controlar sus LEDS y utilizar los distintos botones.

* + 1. **Estructura**

Al principio del Arduino tenemos la inicialización donde se incluyen y se definen los paquetes o expresiones para el correcto funcionamiento y envío/recepción de datos. Entre las expresiones se incluye; el sensor laser, el DHT11, el MQ-2, la recogida de tensión y los datos para la actualización

Después se define los tipos de datos para cada estructura JSON, en donde tenemos que decir todos los datos que se van a enviar por cada estructura y el tipo que son. En todos los topics se envía datos con una estructura JSON.

Luego viene la declaración de las variables globales que se utilizaran en varias funciones del código, donde le decimos de que tipo es y el valor inicial que queremos.

A continuación, tenemos la función para las interrupciones, esta función nos deja utilizar el botón flash que tendrá distintas utilidades dentro del código. También tenemos un filtro anti-rebotes que elimina los pulsos de menos de 50 ms.

Más adelante tenemos la conversión a string de las estructuras JSON para que puedan ser interpretados, aquí decimos el nombre con el que queremos que se marque los valores y creamos grupos para meter los distintos datos.

Seguidamente tenemos la conexión a internet, que es una función donde nos conectamos a internet, que siempre volveremos a ella si nos desconectamos. Además, detrás de esto está la conexión a MQTT, donde también volveremos si da fallos la conexión y no saldremos de ninguna de estas funciones hasta que se corrija. En la función de la conexión a MQTT, tenemos que poner los topics a los que tenemos que subscribirnos y poner el usuario y contraseña del servidor donde enviamos los datos por los topics.

A continuación, tenemos el callback donde cogemos los distintos datos que nos envían los distintos topics a los que nos hemos subscrito, que nos meteremos aquí cada ver que se reciba algo de ahí y ya en la función mira en que topic le ha llegado.

Después tenemos el setup donde configuramos los distintos pines que vamos a utilizar y todas las cosas que se necesita hacer al principio, como la actualización.

Por último, tenemos el loop, que es una función que se repetirá todo el rato. Aquí es donde vamos a llamar a las distintas funciones cuando se necesite utilizarlas, enviar los datos de los sensores y utilizar las interrupciones.

* 1. **Telegram**
     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 NodeRed. En el punto 3.1 ya vimos estos elementos de Telegram, en este punto veremos en detalle cómo funciona.

* + 1. **Funcionamiento**

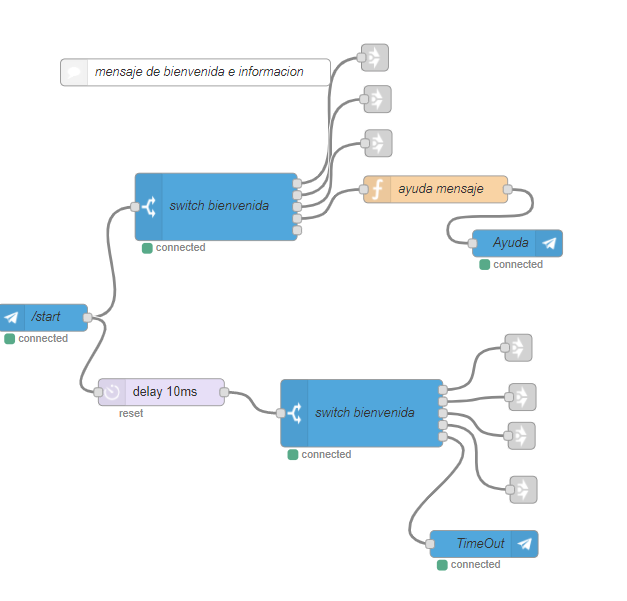
Lo primero que tenemos que hacer es enviar un mensaje de bienvenida, como vemos en la figura 13. Al escribir /start recibimos un mensaje que nos indica las múltiples opciones que tenemos, también tenemos la opción de ver un mensaje tutorial que nos indicará cómo usar cada comando.

Figura 16. Bienvenida Telegram

En la figura 17 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.

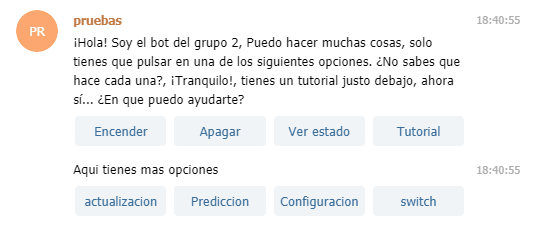


Figura 17. Bienvenida (Telegram)

El mensaje tutorial nos resume todos los comandos que hay y que hace cada uno. En la figura 18 vemos el texto que aparece al clicar en “Tutorial “.

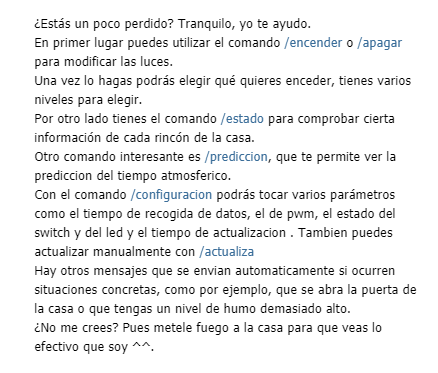


Figura 18. Tutorial Telegram

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 19. Hemos clicado en “todo” para ver todos los estados junto con su fecha de adquisición.

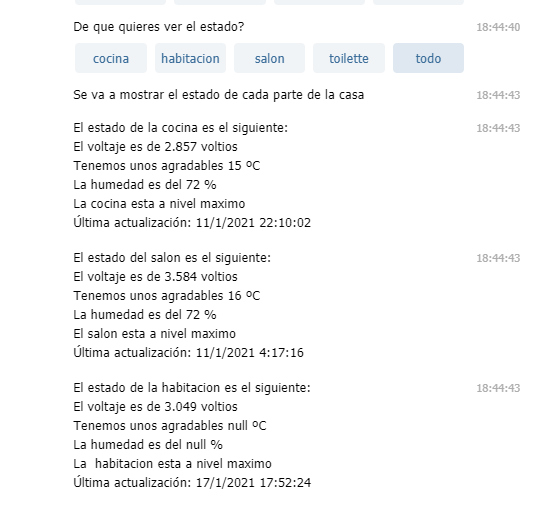


Figura 19. Estado Telegram

Para el flujo de predicción veríamos algo como en la figura 20.

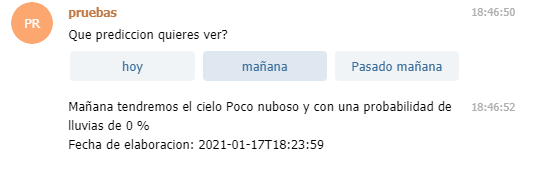
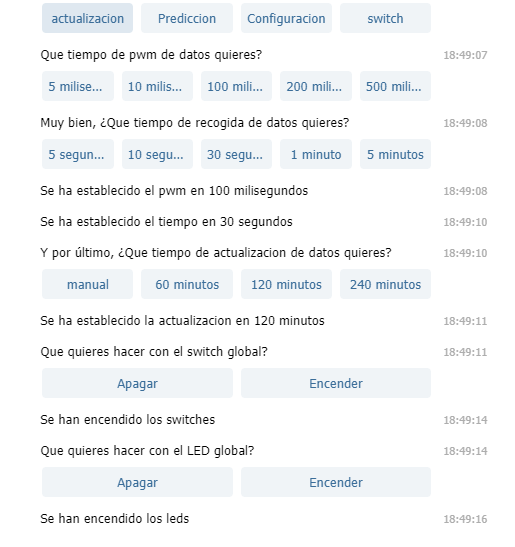


Figura 20. Predicción Telegram

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.

*Figura 21. configuración Telegram*

1. **RESULTADOS Y CONCLUSIONES**

(Text)

1. **MANUAL DE USUARIO**

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

* 1. **Interfaz del NodeRed**

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

* + 1. **¿Cómo puedo ver los datos actuales?**

*(imagen panel de los tabs)*

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

En el recuadro “Uso del área” puede ver una estadística del uso que han tenido las diferentes áreas de la vivienda, que tan frecuentemente son utilizadas respecto a las otras.

En la sección de “Gráficas” puede ver el cambio que ha tenido la humedad y temperatura de todas las áreas de la vivienda a lo largo del tiempo.

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

En esta ventana también puede ver los niveles de C02, Humo y Gas natural (LPG) que hay en la cocina y el salón, indicado con led 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.

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.

* + 1. **¿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.

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”.

*(imagen del selector de tabs)*

En esta ventana podrás modificar haciendo clic con el ratón que tiempo de recogida de datos deseas, tiempo de pwm y tiempo de actualización de datos. Además, también puedes escoger la opción manual en el recuadro de la actualización de datos.

Por otra parte, en el recuadro de “Switch y Led”, puedes encender y apagar todos los switches y leds de la vivienda de forma simultánea.

* + 1. **¿Es posible 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.

*(imagen zoom de configuración para ver temperatura y humedad)*

* 1. **Conexión con el asistente de Google**

* + 1. **¿Qué dispositivos puedo tener?**

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

* Para el Leds que se puede controlar el valor del brillo. 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. 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 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).

* + 1. **¿De dónde son los dispositivos?**

Desde la aplicación tenemos que asignar a cada dispositivo una habitación diferente.

* + 1. **¿Cómo miro los dispositivos?**

Para saber los datos de los dispositivos y cambiarlo, solo tenemos que meterte en la aplicación y seleccionar el dispositivo, en el dispositivo te saldrá toda la información que da el dispositivo y las distintas funciones que tiene.

* + 1. **¿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 la 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”

* + 1. **¿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 la 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) a”

Ej. “Ok Google apaga la Luz Cocina”, esto apaga la Luz Cocina.

* Para la Led que solo tiene encendido y apagado, para controlar el estado se le tiene que decir:

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

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 21ºC.

1. **LISTA DE FICHEROS**