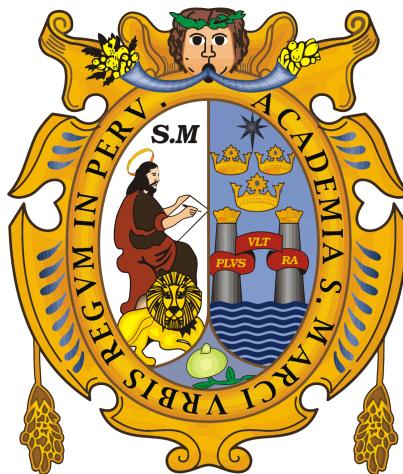


UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
(Universidad del Perú, DECANA DE AMÉRICA)

FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
E.P. DE INGENIERÍA DE SISTEMAS E INFORMÁTICA



Computación Física

**Sistema Integral de Automatización y Seguridad para Hogares
IoT utilizando comunicación MQTT y Cloud Computing**

Profesora:

Yessica Rosas Cueva

Integrantes:

Ayma Pumainca Juan Fidel
Monzon Seguerra Patrick
Navarro Plaza Steven Manuel
Quispe Masias Caleb

Lima Perú

2023

1. Introducción.....	3
a. Revisión del estado del arte.....	3
a.1. Door Security System for Home Monitoring Based on ESP32.....	3
a.2. Design and Build a Home Security System based on an ESP32 Cam Microcontroller with Telegram Notification.....	3
b. Planteamiento del problema.....	3
c. Objetivos.....	4
Objetivo general:.....	4
Objetivos específicos:.....	4
2. Marco teórico.....	5
Mqtt (Publicador Suscriptor).....	5
Node Red.....	6
Broker Mosquitto.....	6
Computing Cloud.....	7
3. Componentes del sistema.....	8
Sistema.....	8
Descripción de los dispositivos utilizados.....	9
Microcontroladores.....	9
-ESP8266.....	9
-Arduino Mega.....	9
Sensores.....	10
-FC28.....	10
-DHT22.....	10
-HC-SR04.....	10
-LDR.....	11
Actuadores.....	11
-Motor DC.....	11
-Modulo Relay 2.....	12
-Bomba de agua DC-1020.....	12
4. Implementación del sistema.....	13
Requerimientos funcionales.....	13
Requerimientos no funcionales.....	13
Diagrama de flujo del proyecto.....	14
Diagrama de comunicación.....	15
Desarrollo del Sistema.....	16
Amazon Web Services (AWS).....	16
Node-Red.....	17
5. Resultados.....	19
6. Conclusiones.....	20
7. Bibliografía.....	20
8. Anexos.....	21
Código diseñado para el arduino Mega 2560.....	21
Código diseñado para el ESP8266.....	25
Código JSON del Node Red.....	33

1. Introducción

a. Revisión del estado del arte

a.1 Door Security System for Home Monitoring Based on ESP32

El artículo propone un sistema de seguridad para puertas basado en la tecnología de Internet de las cosas (IoT) y una aplicación Android llamada "Door Security System". El objetivo principal de este sistema es aumentar la seguridad en una casa al monitorear el estado de la puerta, controlarla y prevenir posibles intrusos.

El sistema utiliza el protocolo de comunicación MQTT en la nube para establecer la conexión entre el teléfono inteligente y el sistema de cerradura de la puerta. Se implementa un sensor PIR en la cerradura de la puerta para detectar movimientos cercanos a la misma, y se instala un sensor táctil en el picaporte para reconocer la presencia de una mano humana. Utiliza sensores y tecnología de comunicación para mejorar la seguridad de las puertas de una casa, brindando a los ocupantes una forma conveniente de monitorear y controlar el estado de la puerta, así como una alerta temprana en caso de intrusión.

a.2 Design and Build a Home Security System based on an ESP32 Cam Microcontroller with Telegram Notification

Este estudio tiene como objetivo crear un dispositivo capaz de aumentar la seguridad de una habitación en el hogar utilizando el ESP32 cam como microcontrolador y el sensor PIR para detectar movimientos al cruzar el área del sensor, así como el sensor de incendios como detector de fuego. En este estudio, el ESP32 cam se utiliza como el cerebro principal del sistema, el cual leerá los datos del sensor de incendios y del sensor PIR, y controlará de forma remota la cerradura de la puerta para abrir y cerrarla. Los datos obtenidos de los sensores serán enviados al servidor a través del ESP32 cam.

Los resultados de las pruebas han sido satisfactorios y se ajustan al sistema diseñado, permitiendo que las imágenes capturadas y las notificaciones de incendio sean enviadas a la aplicación de Telegram con un porcentaje de éxito del 100%.

a.3 Internet of Things Based Home Monitoring and Device Control Using Esp32

Este artículo presenta un sistema de monitorización y control doméstico flexible y fiable de bajo coste con seguridad adicional utilizando ESP32, con conectividad IP a través de Wi-Fi local para acceder y controlar dispositivos por usuario formal de forma remota utilizando la aplicación de teléfono inteligente android. Este sistema es un servidor autónomo y utiliza IoT para controlar los dispositivos deseados por los humanos, desde máquinas industrializadas hasta bienes de usuario. La monitorización del hogar y el sistema de control de dispositivos no sólo se refiere a

la disminución de los esfuerzos humanos, sino también al ahorro de energía y tiempo. Para demostrar la eficacia y viabilidad de este sistema, se presenta un sistema de monitorización del hogar mediante el uso de módulo ESP32. Esto ayuda al usuario a controlar diversas condiciones en el hogar como la temperatura ambiente, fugas de gas, los niveles de agua en el tanque y la detección de personas y el control de diversos aparatos como la luz, ventilador, motor, perilla de gas y tomar decisiones basadas en la retroalimentación de los sensores de forma remota.

b. Planteamiento del problema

En la actualidad, los propietarios de viviendas enfrentan diversos desafíos en términos de comodidad, eficiencia y seguridad. La falta de un control centralizado de los dispositivos y sistemas del hogar dificulta la gestión eficiente de las tareas diarias y el monitoreo en tiempo real. Además, la seguridad del hogar se ve comprometida debido a la dificultad para supervisar y gestionar la seguridad de forma adecuada. Otro problema que se observa es el consumo energético innecesario debido a la falta de automatización. Los propietarios de viviendas no cuentan con un sistema que les permita optimizar el uso de energía, lo que resulta en un mayor gasto y una huella ecológica negativa.

En este contexto, se identifica una brecha entre las necesidades de los propietarios de viviendas y las soluciones tradicionales disponibles en el mercado. Los sistemas de automatización y seguridad actuales no están integrados, lo que limita la capacidad de gestionar eficientemente el hogar y el monitoreo en tiempo real.

Por tanto, es necesario desarrollar e implementar un sistema de IoT que aborde estas problemáticas y permita a los propietarios de viviendas mejorar la comodidad, eficiencia y seguridad en sus hogares. Los objetivos específicos del proyecto incluyen el diseño e implementación de un sistema centralizado de control para dispositivos domésticos, la integración de sensores y cámaras para mejorar la seguridad y el monitoreo en tiempo real, así como la automatización de tareas para optimizar el consumo de energía.

Al lograr estos objetivos, se espera obtener beneficios significativos para los propietarios de viviendas, como una mayor comodidad y conveniencia en la gestión diaria, una mayor seguridad mediante el monitoreo en tiempo real y una reducción del consumo de energía. En última instancia, este proyecto contribuirá a mejorar la calidad de vida en el hogar, brindando un entorno más eficiente, seguro y confortable para los residentes.

c. Objetivos

Objetivo general:

Desarrollar e implementar un sistema de automatización y seguridad para una casa utilizando tecnología de Internet de las cosas (IoT), con el fin de mejorar la comodidad, eficiencia y seguridad del hogar.

Objetivos específicos:

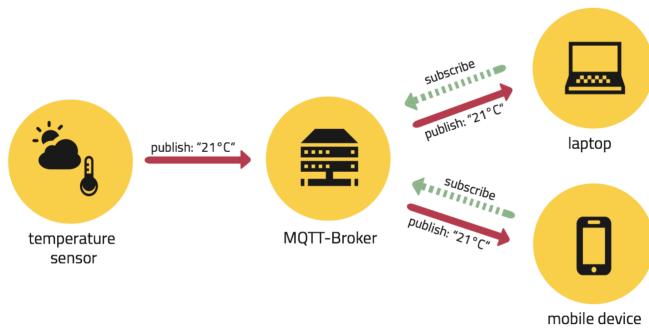
- Identificar y seleccionar los dispositivos y sensores adecuados para la automatización y seguridad del hogar, considerando aspectos como iluminación, climatización, cerraduras, detección de movimiento y alarmas.
- Desarrollar y programar un sistema centralizado que permita controlar y monitorear los dispositivos y sensores de manera remota, a través de un dashboard.
- Establecer reglas y automatizaciones personalizadas para optimizar el uso de energía, como el control de iluminación y climatización en función de la presencia de personas en las habitaciones o la detección de apertura o cierre de puertas y ventanas.
- Integrar un sistema de vigilancia y detección de intrusos utilizando sensores de movimiento y/o sensores ultrasónicos, enviando notificaciones en tiempo real al propietario en caso de eventos sospechosos.
- Establecer una conexión bidireccional MQTT entre los dispositivos y el servidor de aplicaciones en la Cloud Computing, permitiendo el intercambio de mensajes en tiempo real.
- Realizar pruebas del sistema de automatización y seguridad, simulando diferentes situaciones y escenarios para garantizar su funcionamiento correcto y confiable.
- Documentar y compartir los resultados del proyecto mediante un informe, con el objetivo de contribuir al conocimiento y la difusión de sistemas de automatización y seguridad del hogar basados en IoT.

2. Marco teórico

Mqtt (Publicador Suscriptor)

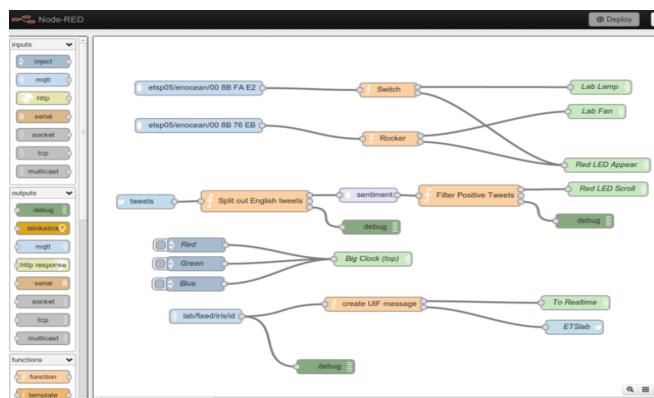
MQTT es un protocolo de mensajería ligero diseñado para la comunicación eficiente entre dispositivos de IoT. Se destaca por su bajo consumo de ancho de banda y recursos, así como su facilidad de implementación. Utiliza un modelo de publicación/suscripción, donde los dispositivos publican mensajes en temas y los interesados se suscriben a estos temas para recibirlos. Ofrece tres niveles de calidad de servicio (QoS) para garantizar la entrega confiable de mensajes.

La topología de MQTT es flexible, permitiendo la conexión a un broker centralizado o la comunicación directa entre dispositivos. Proporciona opciones de seguridad como autenticación y encriptación de las comunicaciones. Es compatible con múltiples plataformas y lenguajes de programación, lo que facilita su integración en diferentes sistemas y dispositivos.



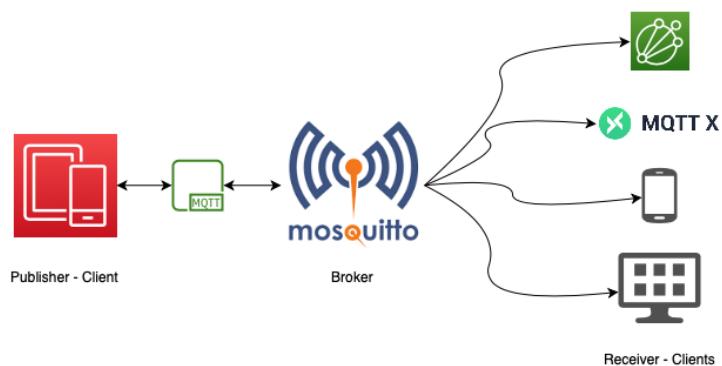
Node Red

Node-RED es una plataforma de programación visual para IoT que simplifica el desarrollo de aplicaciones y flujos de trabajo. Con su interfaz gráfica intuitiva, permite a los usuarios conectar nodos predefinidos para automatizar tareas y facilitar la comunicación entre dispositivos y servicios. Destaca por su integración con MQTT, lo que facilita la interacción con dispositivos IoT. Además, cuenta con una amplia biblioteca de nodos y es flexible y escalable, adaptándose a diferentes entornos de implementación. En resumen, Node-RED es una herramienta poderosa para crear soluciones IoT rápidamente, desde la monitorización de dispositivos hasta la gestión de datos y la domótica.



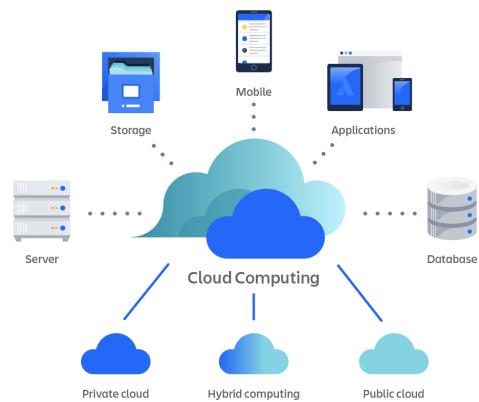
Broker Mosquitto

Mosquitto es un popular software de código abierto que implementa un broker MQTT. Actúa como un intermediario entre los dispositivos que utilizan el protocolo MQTT, permitiendo la publicación y suscripción de mensajes. Mosquitto es conocido por ser ligero, fácil de configurar y altamente escalable, lo que lo hace adecuado para su uso en entornos de IoT con recursos limitados. Proporciona funciones de seguridad, como autenticación y cifrado, para proteger las comunicaciones MQTT.



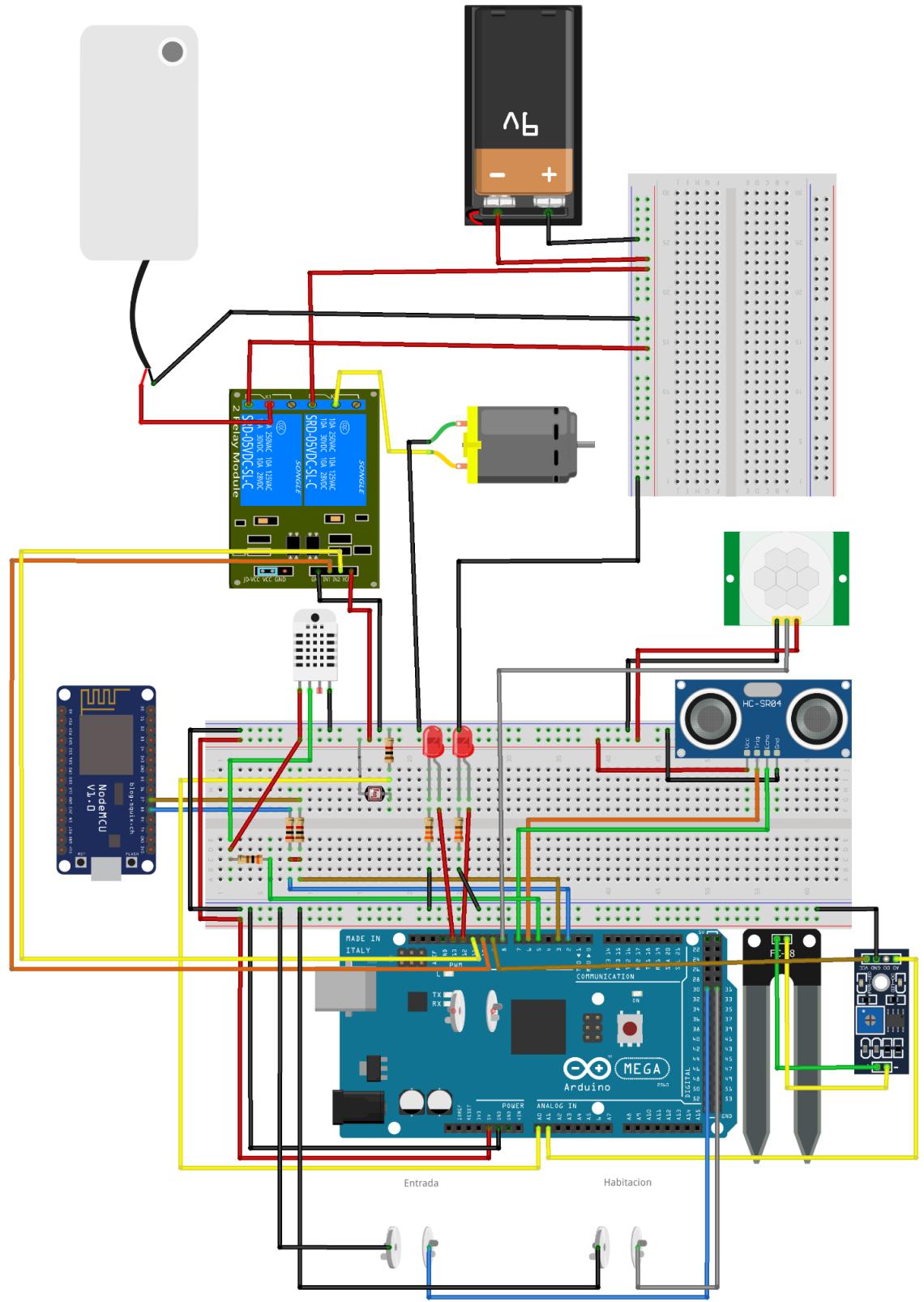
Computing Cloud

La computación en la nube, también conocida como cloud computing, es un modelo de entrega de servicios de computación a través de Internet. Permite a los usuarios acceder a recursos informáticos, como servidores, almacenamiento, bases de datos y software, a través de una red de servidores remotos. En lugar de mantener infraestructuras locales costosas, los usuarios pueden aprovechar la escalabilidad y flexibilidad de la nube para satisfacer sus necesidades de cómputo de manera más eficiente. La computación en la nube ofrece beneficios como la disponibilidad bajo demanda, la flexibilidad de escalar recursos según sea necesario y la externalización de la administración de la infraestructura. En resumen, la computación en la nube es un modelo de entrega de servicios informáticos a través de Internet que proporciona acceso flexible y escalable a recursos informáticos.



3. Componentes del sistema

Sistema



fritzing

Descripción de los dispositivos utilizados

Microcontroladores

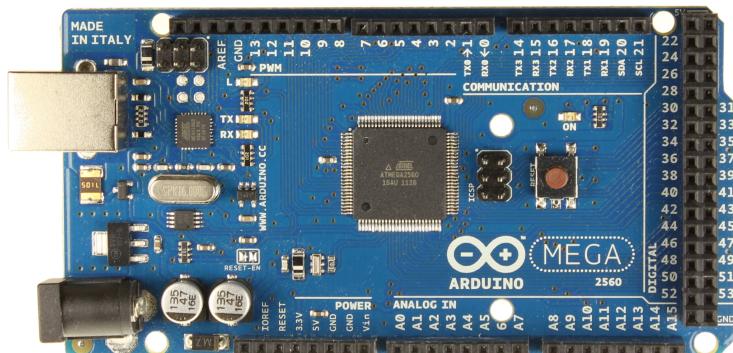
-ESP8266

El ESP8266, desarrollado por Espressif Systems, es un microcontrolador económico y de bajo consumo de energía que integra un chip Wi-Fi, permitiendo realizar tareas de comunicación a través de Internet. Se utiliza comúnmente en proyectos de Internet de las cosas (IoT) debido a su versatilidad y facilidad de programación. Gracias a sus puertos GPIO, permite la conexión de sensores y actuadores, y su tamaño compacto lo hace ideal para aplicaciones de prototipos y proyectos de automatización. Aunque han surgido versiones mejoradas como el ESP32, el ESP8266 sigue siendo popular debido a su asequibilidad y disponibilidad en el mercado.



-Arduino Mega

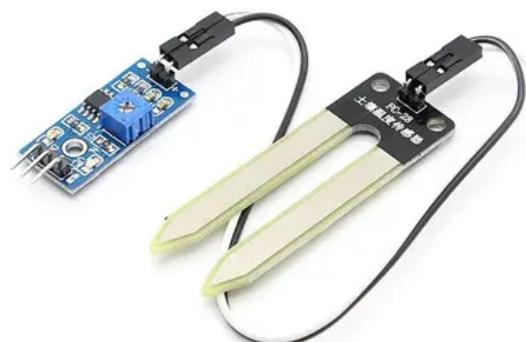
El Arduino Mega es una placa de desarrollo basada en el microcontrolador ATmega2560. Es una versión ampliada del Arduino UNO, con mayor cantidad de pines de entrada/salida y más memoria, lo que lo hace adecuado para proyectos más complejos. El Arduino Mega es compatible con el entorno de programación de Arduino, lo que permite a los usuarios escribir y cargar código fácilmente.



Sensores

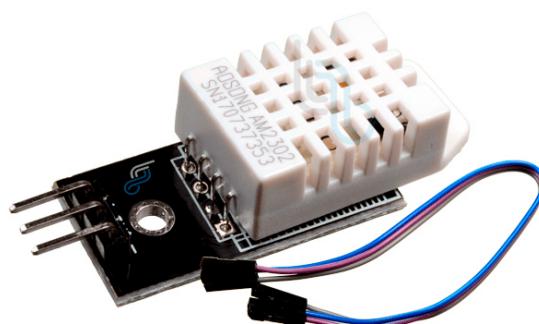
-FC28

El sensor FC28, también conocido como sensor de humedad del suelo, será utilizado para medir la humedad en el suelo. Este sensor permite detectar la cantidad de agua presente en el suelo y proporciona una lectura analógica que puede ser interpretada por el ESP8262. Esto permitirá monitorizar la humedad del suelo en tiempo real, lo que resulta útil en aplicaciones agrícolas para controlar el riego de manera automática o para detectar condiciones de sequía o exceso de humedad.



-DHT22

El sensor DHT22, o sensor de temperatura y humedad, permitirá medir tanto la temperatura ambiente como la humedad relativa del entorno. El DHT22 ofrece una alta precisión y estabilidad en las mediciones, y su interfaz de comunicación es compatible con el ESP8262. La información obtenida de este sensor puede ser utilizada en aplicaciones de control ambiental, como la activación automática de sistemas de climatización, la monitorización de condiciones de cultivo o la detección de condiciones ambientales adversas.



-HC-SR04

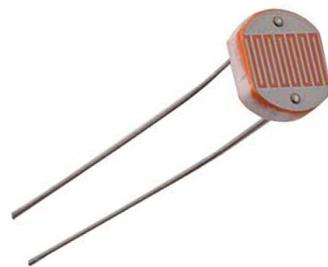
El sensor HC-SR04, conocido como sensor ultrasónico de distancia, será empleado para medir la distancia entre el sensor y un objeto. Este sensor emite pulsos ultrasónicos y mide el tiempo que tarda en recibir el eco de vuelta, permitiendo calcular la distancia de manera precisa. Con la información obtenida de este sensor,

se pueden implementar aplicaciones como sistemas de detección de obstáculos, control de acceso basado en proximidad o robots autónomos que eviten colisiones.



-LDR

El sensor LDR, o fotoresistor, será utilizado para medir la intensidad de la luz en un entorno. El LDR varía su resistencia eléctrica en función de la cantidad de luz incidente, permitiendo obtener una lectura analógica que puede ser procesada por el ESP8262. Este sensor se utiliza en aplicaciones de iluminación automatizada, monitoreo de condiciones de luminosidad, detección de presencia lumínica o control de sistemas de iluminación basados en la cantidad de luz ambiental.



Actuadores

-Motor DC

El motor DC es un tipo de actuador que convierte la energía eléctrica en movimiento mecánico. Se puede controlar la velocidad y dirección del motor DC mediante señales de control proporcionadas por el microcontrolador ESP8262. Este actuador es versátil y se utiliza en una amplia variedad de aplicaciones, como robots móviles, sistemas de control de posicionamiento y dispositivos automatizados.



www.rambal.com

-Modulo Relay 2

El módulo de relé de 2 canales es un dispositivo que actúa como un interruptor controlado por señales eléctricas. Permite el control de dispositivos de mayor potencia que no pueden ser manejados directamente por el microcontrolador. El módulo de relé de 2 canales cuenta con dos relés independientes que pueden ser activados o desactivados mediante señales enviadas desde el ESP8262. Es ampliamente utilizado para controlar electrodomésticos, sistemas de iluminación y otros dispositivos de alta potencia.



-Bomba de agua DC-1020

La bomba de agua DC-1020 es un actuador específico diseñado para el bombeo de agua. Esta bomba, alimentada por corriente continua, es capaz de generar presión y caudal para mover el agua a través de tuberías u otros sistemas de distribución. La bomba de agua DC-1020 puede ser controlada por el microcontrolador ESP8262 mediante la conexión de los cables de alimentación y control adecuados.



4. Implementación del sistema

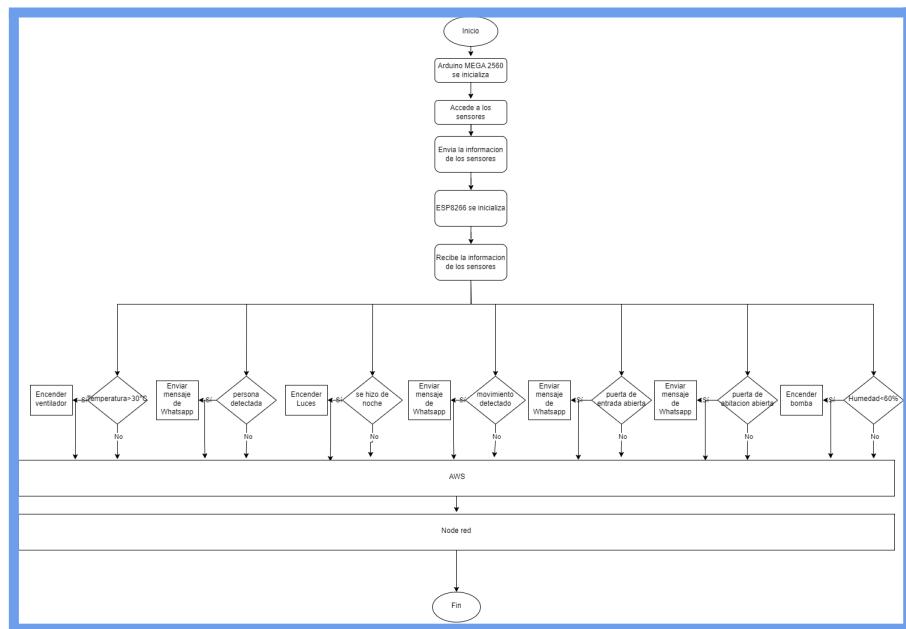
Requerimientos funcionales

- El sistema mostrará en tiempo real los valores de los sensores, usa el Dashboard de Node-red.
- El sistema se encargará de enviar una notificación por Whatsapp cada vez que el sensor de movimiento o sensores magnéticos detecten alguna anomalía.
- El sistema permitirá al usuario encender focos, encender ventiladores o encender la bomba automática cada vez que desee.
- El sistema encenderá los ventiladores cada vez que la temperatura del ambiente supere el umbral determinado por el usuario.
- El sistema realizará el riego de macetas cuando el sensor FC-28 registre una humedad menor al 60%.
- El sistema activará los focos cuando el sensor de luz esté por debajo de los valores determinados por el usuario.
- El sistema enviará una notificación por voz cada vez que el sensor ultrasónico ubicado en la puerta registre la presencia de una persona en un rango de 20 cm.
- El sistema almacenará la información recibida por los sensores.

Requerimientos no funcionales

- El sistema no demorará más de 3 segundos entre la orden recibida por el Dashboard de Node Red y los actuadores.
- El sistema funcionará siempre que se conecte a internet.
- El sistema es escalable, lo que significa que podrá añadirse nuevos sensores y más actuadores.

Diagrama de flujo del proyecto

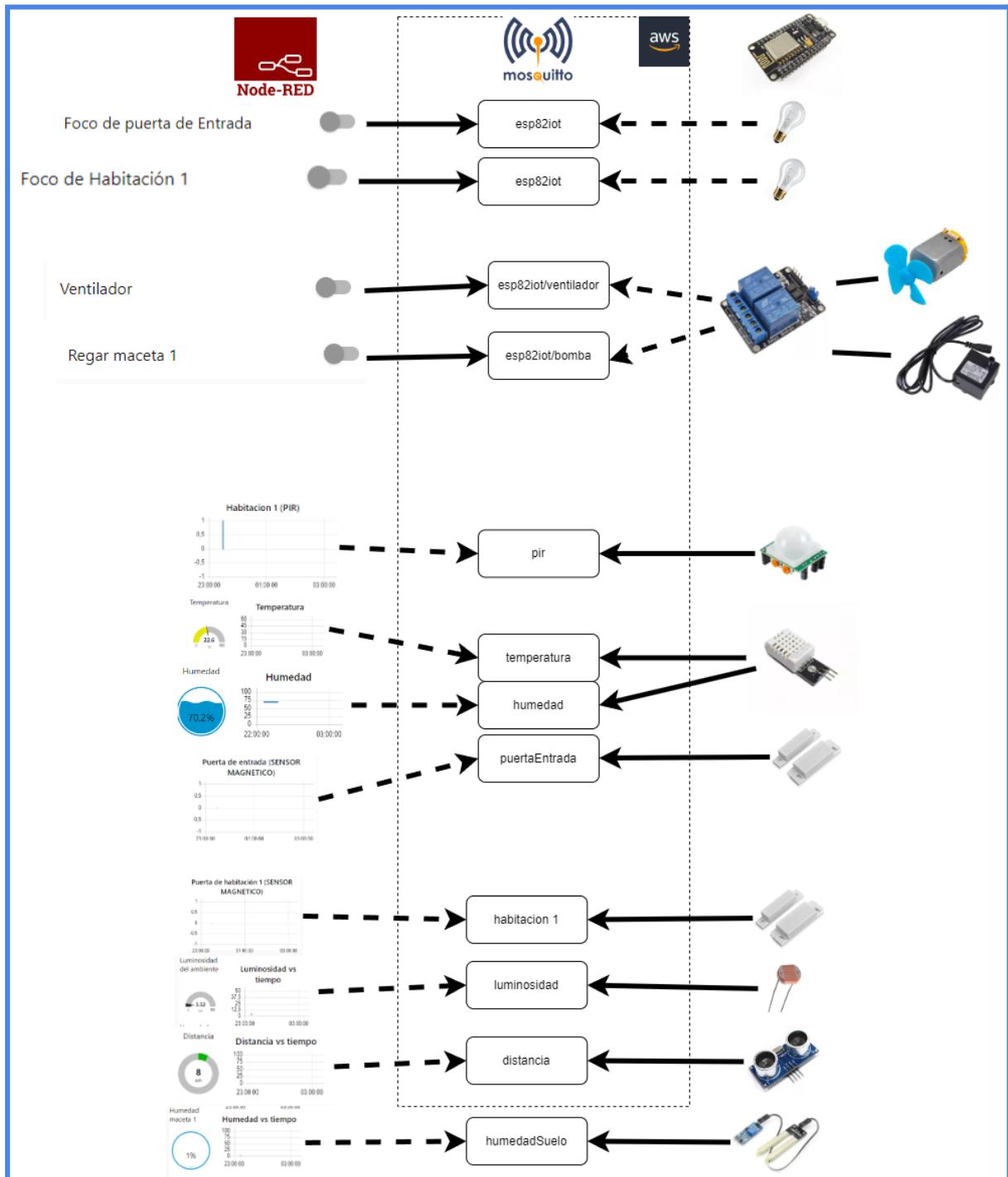


El diagrama de flujo muestra la secuencia de tareas que el sistema realiza, primero el Arduino mega se encargará de tomar los datos de todos los sensores y hacer la transformación necesaria para enviarla al ESP8266, este una vez lo tenga realizará una serie de acciones en relación a los datos que se le envía.

Los datos de los sensores serán enviados mediante el protocolo MQTT al broker situado en AWS y luego gestionado por Node Red.

La información es almacenada en la computadora de forma local con ayuda de Node Red.

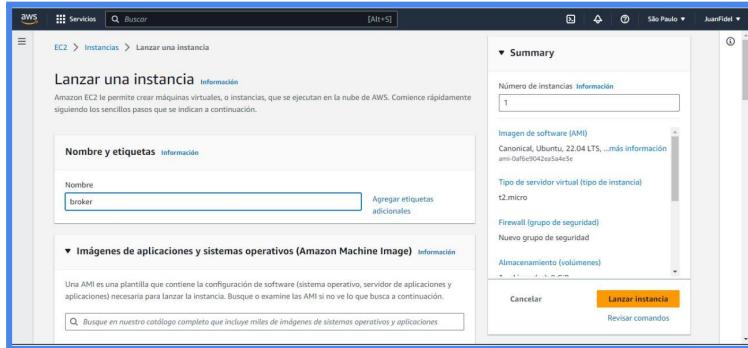
Diagrama de comunicación



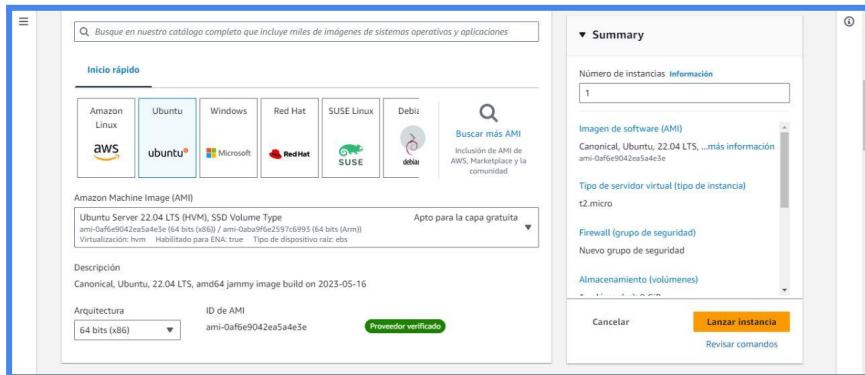
Desarrollo del Sistema

Amazon Web Services (AWS)

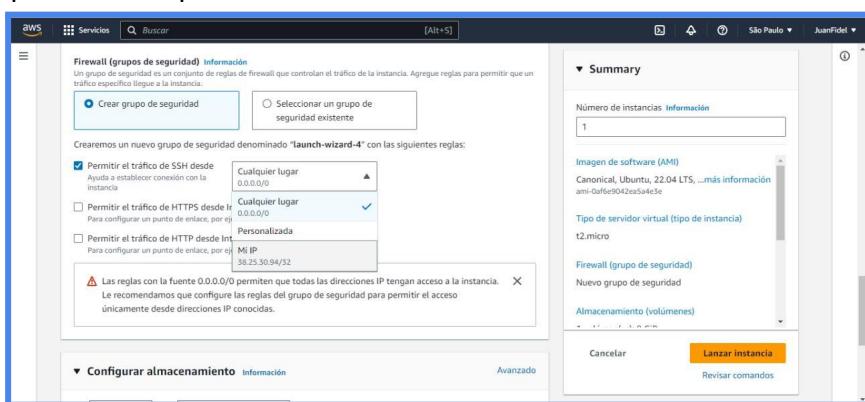
Para tener un broker es necesario la utilización de Mosquitto, para ello vamos a hacer uso de una instancia en la nube. El broker se va crear en una máquina virtual contenida en Amazon Web Service.



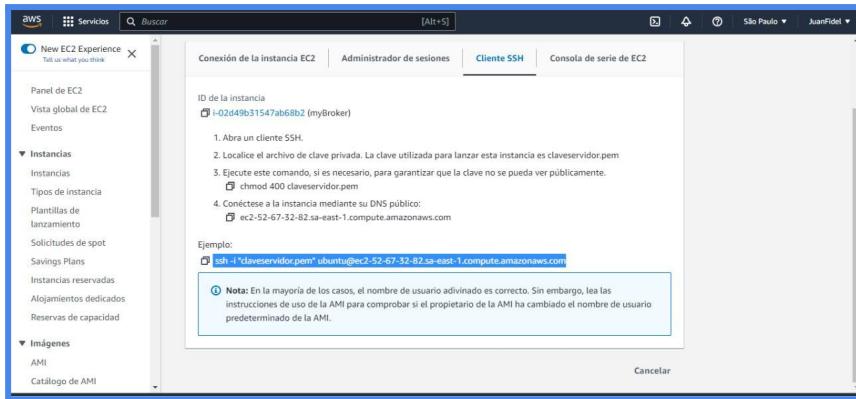
Se creará con el sistema operativo Ubuntu.



También se configurará el firewall de modo que obtengamos los puertos por defecto que usa Mosquitto.



Finalmente nos conectamos por consola a nuestra máquina virtual.



A partir de aquí instalamos Mosquitto y con eso tenemos nuestro broker en AWS.

```
Expanded Security Maintenance for Applications is not enabled.

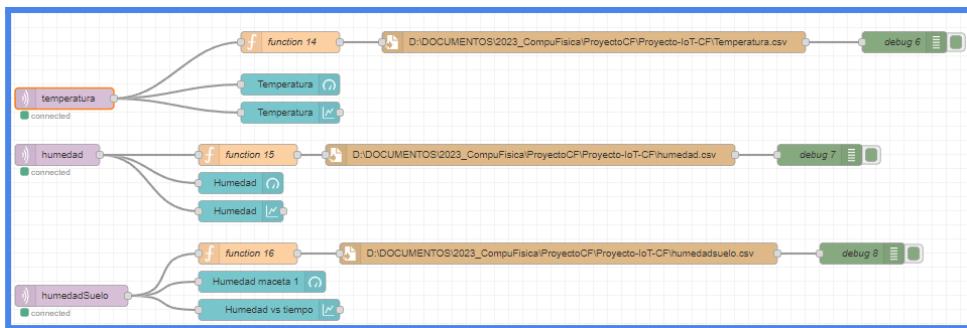
62 updates can be applied immediately.
16 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

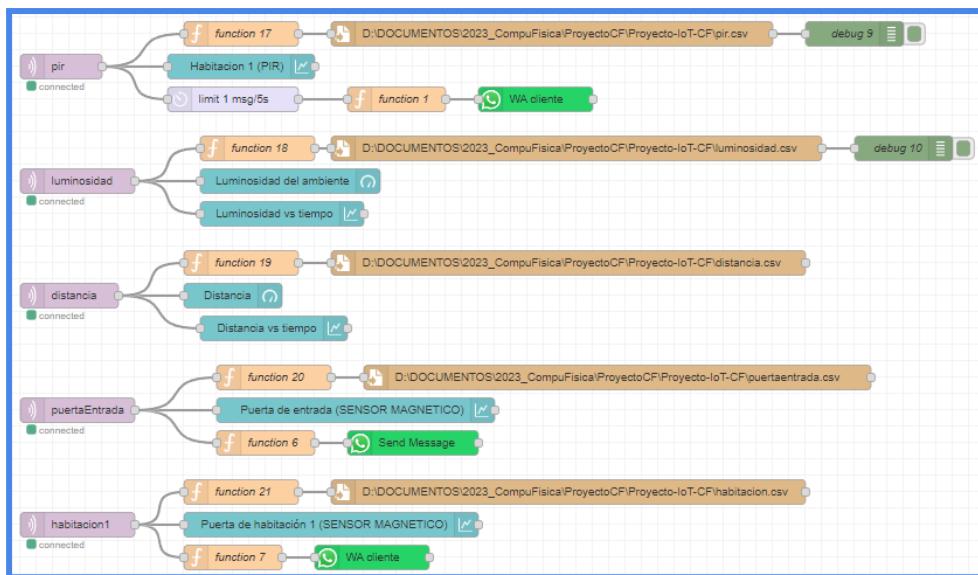
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Sun May 21 02:48:22 2023 from 38.25.16.86
ubuntu@ip-172-31-32-215:~$ |
```

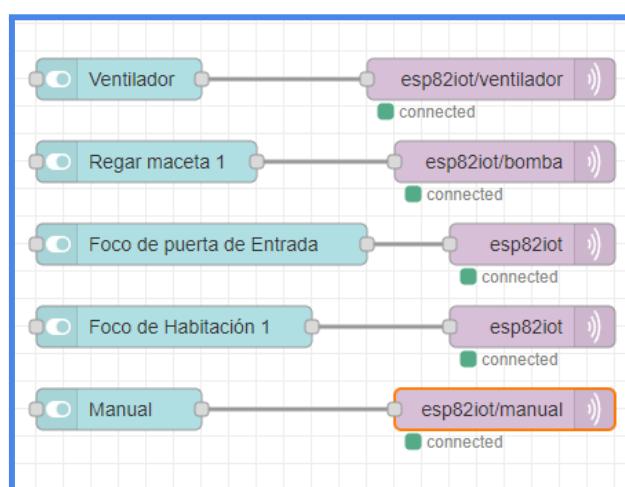
Node-Red

Sensores

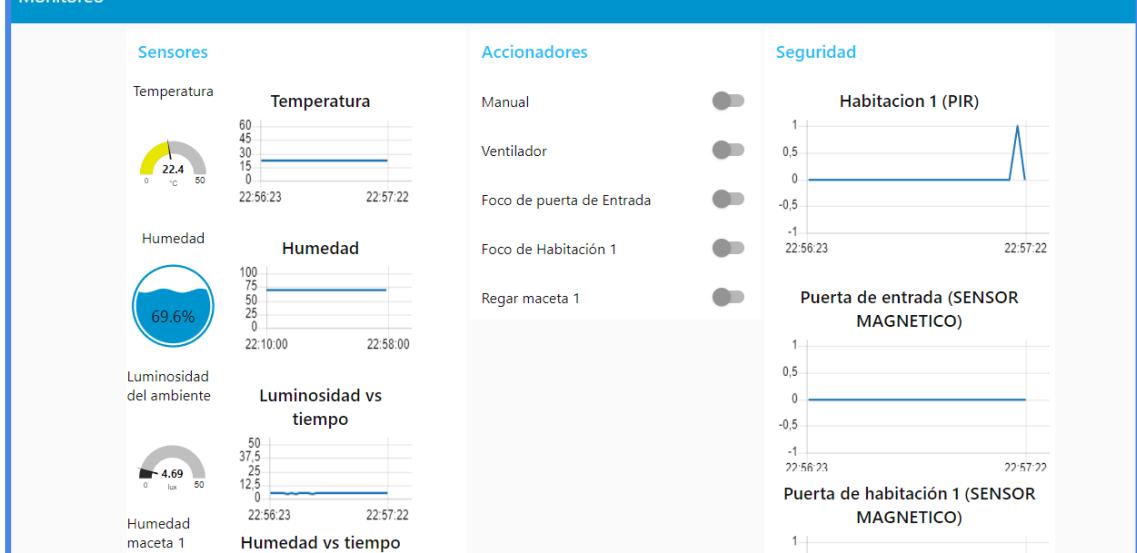


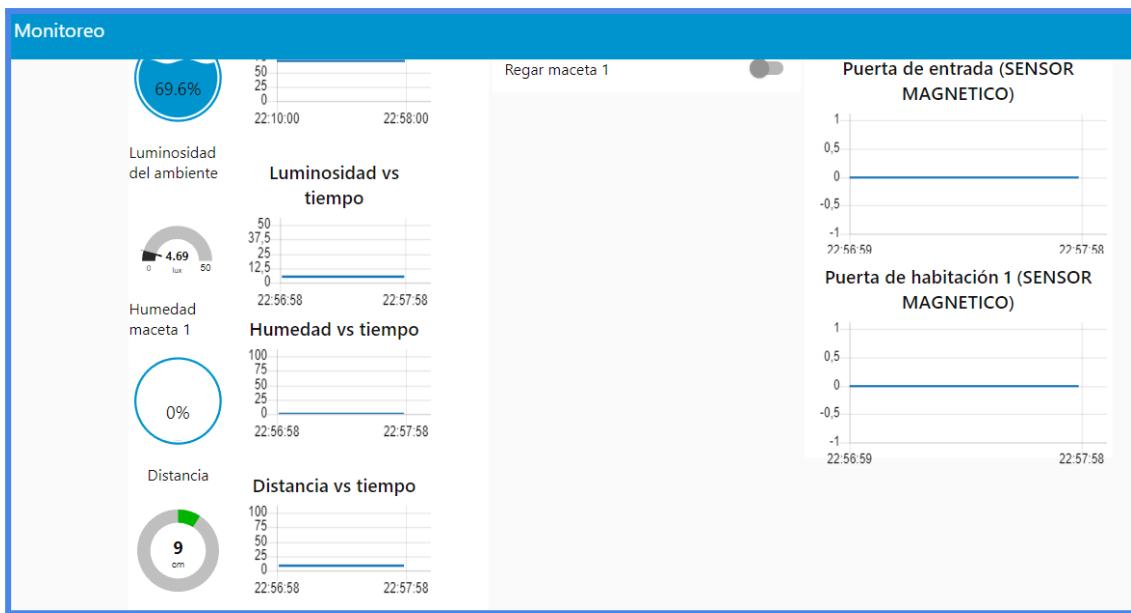


Actuadores



Monitoreo





5. Resultados

Se registraron las pruebas ejecutadas en las dos reuniones presenciales realizadas los días 23/06/203 y 07/07/2023.

Fecha	Descripción	Fallo	Estado
23/06/2023	Prueba de sensor PIR	Falla energética	En revisión ▾
23/06/2023	Prueba de Modulo Relay	-	Aprobada ▾
23/06/2023	Prueba de sensor HC-SR04	Medida inadecuada	En revisión ▾
23/06/2023	Sensor DHT22	-	Aprobada ▾
23/06/2023	Sensor FC28	Modo de Pin	Aprobada ▾
07/07/2023	Sensor Magnético	-	Aprobada ▾
07/07/2023	Motor DC	-	Aprobada ▾
07/07/2023	Bomba de agua	Adaptador inadecuado	En revisión ▾
07/07/2023	Prueba de sensor PIR	-	Aprobada ▾
07/07/2023	Flujo node red	-	Aprobada ▾
07/07/2023	Prueba de sensor HC-SR04	Módulo defectuoso	Aprobada ▾

6. Conclusiones

- El sistema de automatización permite a los propietarios controlar de manera centralizada los dispositivos y sistemas del hogar, lo que simplifica las tareas diarias y proporciona una mayor comodidad. La capacidad de programar y ajustar automáticamente diversas configuraciones según las necesidades individuales mejora la experiencia de vivir en el hogar.
- La automatización de tareas y la optimización del consumo de energía contribuyen a una mayor eficiencia energética en el hogar. El sistema de IoT permite programar y controlar el encendido y apagado de dispositivos según la demanda real, lo que conduce a un uso más eficiente de la energía y, en última instancia, a la reducción de los costos asociados.
- La implementación de un sistema de IoT en el hogar demuestra una adopción avanzada de tecnología y una adaptación a las tendencias actuales. Los propietarios de viviendas se benefician al tener un hogar conectado que aprovecha al máximo las ventajas de la automatización y la seguridad inteligente.

7. Bibliografía

<https://www.sciencedirect.com/science/article/pii/S1877050919311378>

<https://jartel.polinema.ac.id/index.php/jartel/article/view/296/189>

https://www.researchgate.net/profile/Rajendra-Prasad-Ch/publication/334226986_Internet_of_Things_Based_Home_Monitoring_and_Device_Control_Using_Esp32/links/5d1de7fe458515c11c0ff074/Internet-of-Things-Based-Home-Monitoring-and-Device-Control-Using-Esp32.pdf

<https://github.com/PfisterDaniel/node-red-contrib-whatsapp-cmb/blob/main/README.md#create-api-key>

8. Anexos

Código diseñado para el arduino Mega 2560

```
/*ZONA DE INCLUSIÓN DE LIBRERIAS*/
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <SoftwareSerial.h>
#include <NewPing.h>

String data="";

/*DEFINICIÓN DE PINES*/
#define rxPin 2
//Pines para el envío se Arduino Uno a NodeMCU.
#define txPin 3

//Sensores magneticos
#define sensorMagPin1 40      //Puerta de entrada
#define sensorMagPin2 45      //Habitacion 1
int valMag1=0;
int valMag2=0;

//Sensor de humedad DHT22
#define DHT_PIN 5 // Se define el pin para conectar el sensor
DHT22
#define DHT_TIPO DHT22 //Definimos el modelo del sensor DHT22
#define LED 16

//Sensor LDR
#define LDR_PIN A0

//Sensor FC-28
#define sensorFCPin A1 // Sensor FC-28 pin analogico
#define pinfc28V 9 //control de alimentacion
```

```

int humedadSuelo = 0;
float valHumsuelo;

//PIR
int pinPir = 8;
int val = 0; //valor que recibimos del pir
int pirEstado = LOW; //estado inicial del pir, no hay
movimiento

//pines SensorUltrasonico
#define TRIG_PIN 6      // trigger
#define ECO_PIN 7      // echo
unsigned long tiempo, distancia;
NewPing sonar(TRIG_PIN, ECO_PIN);

/*DECLARACIÓN DE VARIABLES GLOBALES*/
unsigned int valor_medido;
//Valor obtenido al leer por A0.
int ledPin = 10;
//Pin de conexión del led de prueba.
float LR;
//Variable para almacenar la luminosidad relativa.

/*PROTOTIPADO*/
SoftwareSerial Trans(rxPin, txPin);           //Definimos
puerto serie virtual. Pines para Rx y Tx.
DHT dht(DHT_PIN, DHT_TIPO);                  //Nombramos
y configuramos el sensor para poder inicializarlo.

/*Variables DHT22*/
float t;
float h;

long tdhtx = 0;

void setup()
{
    pinMode(ledPin, OUTPUT);
    pinMode (txPin , OUTPUT);
    Serial.begin(9600);
    Trans.begin(9600);                         //Velocidad de
envío para la transmisión al NodeMCU.
    Serial.println("Gestionador de sensores!");
    //Configuramos el los pines FC-28
    pinMode(sensorFCPin, INPUT);
}

```

```

pinMode(pinfc28V, OUTPUT);
pinMode(sensorMagPin1, INPUT);
pinMode(sensorMagPin2, INPUT);
dht.begin(); //Arrancamos
el sensor.
}

void loop()
{
    long tiempo = millis();
//messageData="";

    sensorUltrasonico();
    sensorPir();
    sensorMagnetico();
    if (tiempo - tdhtx > 2000) { //Envia cada 2 segundos
        tdhtx = tiempo;

        sensorDHT();
        sensorLDR();
        sensorFC28();

        data= "T" + String(t);
        data = data + "H" + String(h);
        data = data + "L" + String(LR);
        data = data + "VHS" + String(valHumsuelo);
        data = data + "U" + String(distancia);
        data = data + "Mov" + String(val);
        data = data + "Mag1" + String(valMag1);
        data = data + "Mag2" + String(valMag2);

        Trans.println(data);
        Serial.println(data);

    }
}

void sensorDHT() {
    //Enviando humedad y temperatura
    t = dht.readTemperature();
    h = dht.readHumidity();
    if (isnan(h) || isnan(t)) {
        Serial.println(F("Failed to read from DHT sensor!"));
        return;
    }
}

```

```

}

void sensorLDR() {
    valor_medido = analogRead(LDR_PIN);
    //Valor de luminosidad medido en A0
    LR = (valor_medido * 100.0) / 1023.0;
    //Luminosidad medida en tanto por ciento
    if (isnan(LR)) {
        Serial.println(F("Failed to read from LDR sensor!"));
        return;
    }
}

void sensorFC28() {

    //Enviando humedad y temperatura
    //Aplique energía al sensor de humedad del suelo
    digitalWrite(pinfc28V, HIGH);
    delay(10); // espera de 10 milisegundos
    humedadSuelo = analogRead(sensorFCPin);
    // Apague el sensor para reducir la corrosión del metal
    //con el tiempo
    digitalWrite(pinfc28V, LOW);
    //Convertir el valor en porcentaje
    valHumsuelo = map(humedadSuelo, 1023, 0, 0, 100);
    if (isnan(valHumsuelo)) {
        Serial.println(F("Failed to read from FC-28 sensor!"));
        return;
    }
}
//Metodo para leer los sensores magneticos
void sensorMagnetico()
{
    //Sensor magnetico de puerta de entrada
    valMag1 = digitalRead(sensorMagPin1);
    valMag2 = digitalRead(sensorMagPin2);
}

void sensorPir()
{
    //Sensor de Movimiento
    val = digitalRead(pinPir);
    // if (val == HIGH) { //si está activado
    //     digitalWrite(pinLED,HIGH);//encender LED movimiento
    //     if (pirEstado == LOW) { //si previamente estaba
apagado
    //         Serial.println("Sensor activado");
    //         pirEstado = HIGH;
}

```

```

        //      }
        //    } else { //si está desactivado
        //      //digitalWrite(pinLED,LOW);
        //      if (pirEstado == HIGH) { //si previamente estaba
encendido
        //        Serial.println("Sensor apagado");
        //        pirEstado = LOW;
        //      }
        //    }
      }

void sensorUltrasonico() {

  tiempo = sonar.ping_median(5); //tiempo promedio de ir y
venir del trigger
  distancia = tiempo / US_ROUNDTRIP_CM;

  if (isnan(distancia)) {
    Serial.println(F("Failed to read from
sensorUltrasonico!"));
    return;
  }
  delay(50);
}

```

Código diseñado para el ESP8266

```

//Librerias
#include <SoftwareSerial.h>
#include <ESP8266WiFi.h>
#include <PubSubClient.h>

//Serial para la comunicacion con el arduino Mega 2560
SoftwareSerial SerialESP8266(13, 15); // RX, TX

//variables para la captura del mensaje del Arduino Mega por
monitor serial
const int lonbuffer = 12;
// longitud del buffer
char buffer[lonbuffer];
// buffer para almacenar el comando
float luminosidad;
// valor del tercer parametro

boolean manual = false;

```

```

float t;
float h;
float l;
float valHumsuelo;
int distancia;
int movimiento;
int sensorMgn1;
int sensorMgn2;

//Pines de los focos
#define greenPin D4
#define redPin D3

//Pin del ventilador
#define ventiladorPin D5

//Pin del ventilador
#define bombaPin D6

//Configuracion del user
const char* ssid = "TP_Ayma";
const char* password = "09089327";

char data_temp[12] = "";
char data_humi[12] = "";
char data_humiSuelo[12] = "";
char data_digital[12] = "";
char data_dist[12] = "";
char data_lumi[12] = "";
char data_Mag1[12] = "";
char data_Mag2[12] = "";

//IP del servidor BROKER
//const char *mqtt_broker = "192.168.0.110";
const char *mqtt_broker = "18.231.187.97";
//const char *mqtt_broker = "15.229.78.220";
const int mqtt_port = 1883;
//const char* mqtt_server="test.mosquitto.org";

/*
 // MQTT Broker
 const char *mqtt_broker = "broker.emqx.io";
 const char *mqtt_username = "emqx";
 const char *mqtt_password = "publico";
 const int mqtt_port = 1883;
*/

```

```

//Inicializamos el nodeMCU para wifi y Mqtt
WiFiClient esp82;
PubSubClient client(esp82);

long lastMsg = 0;
long tdhtx = 0;
char msg[50];
int value = 0;

//*****Funciones para Mqtt*****
//-----Configuración Wifi-----
void config_wifi() {
    //Conectandose a una red Wifi
    Serial.println("Conectandose a la red ...");
    Serial.println(ssid);
    WiFi.begin(ssid, password);

    // Mientras el ESP82 no se conecte al AP:
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }

    Serial.println("");
    Serial.println("ESP82 conectado, su IP es: ");
    Serial.println(WiFi.localIP());
}

//-----Lectura de datos que llegan despues de
//suscribir-----
void callback(String topic, byte* message, unsigned int
length) {
    //Serial.println("Mensaje que llega del topic: ");
    Serial.print("Mensaje que llega del topic[");
    Serial.print(topic);
    Serial.print("] ");
    //Serial.println(". Message: ");
    String messageData = "";

    for (int i = 0; i < length; i++) {
        //Serial.println((char)message[i]);
        Serial.print((char)message[i]);
        messageData += (char)message[i];
    }
    Serial.println();
    Serial.print("Tamaño del mensaje :");
    Serial.println(length);
    Serial.println();
}

```

```

Serial.println("-----");
Serial.println(messageData);

if (String(topic) == "esp82iot/manual") {
    if (messageData == "On") {
        manual = true;
    } else {
        manual = false;
    }
}

//FOCOS DE LA PUERTA DE ENTRADA Y HABITACIÓN 1
if (String(topic) == "esp82iot" && manual) {
    if (messageData == "redOn") {
        Serial.println("red On");
        digitalWrite(redPin, HIGH);
        //Foco de entrada
    } else if (messageData == "redOff") {
        Serial.println("red Off");
        digitalWrite(redPin, LOW);
    }
    if (messageData == "greenOn") {
        Serial.println("green On");
        digitalWrite(greenPin, HIGH);
    } else if (messageData == "greenOff") {
        Serial.println("greenOff");
        digitalWrite(greenPin, LOW);
    }
}
//VENTILADOR
if (String(topic) == "esp82iot/ventilador" && manual) {
    if (messageData == "On") {
        manual = true;
        Serial.println("ventilador On");
        digitalWrite(ventiladorPin, LOW);
    } else if (messageData == "Off") {
        Serial.println("ventilador off");
        digitalWrite(ventiladorPin, HIGH);
    }
}
//BOMBA DE AGUA
if (String(topic) == "esp82iot/bomba" && manual) {
    if (messageData == "On") {
        Serial.println("bomba On");
        digitalWrite(bombaPin, LOW);
    } else if (messageData == "Off") {
        Serial.println("bomba off");
        digitalWrite(bombaPin, HIGH);
    }
}

```

```

        }

    }

//-----Conexión a broker/
Suscribir-----
void reconnect() {
    while (!client.connected()) {
        Serial.println("Iniciando conexión con Broker...");
        String clienteId = "ESP82";
        if (client.connect(clienteId.c_str())) {
            //if (client.connect(clienteId.c_str(), mqtt_username,
mqtt_password)) {
                client.subscribe("esp82iot");//topic para suscribir
                //client.subscribe("esp82iot/");//topic para suscribir
                client.subscribe("esp82iot/ventilador");
                client.subscribe("esp82iot/bomba");
                client.subscribe("esp82iot/manual");//modo manual
                Serial.println("Conexión exitosa");
            }
        } else {
            Serial.println("Failed, rc=");
            Serial.print(client.state());
            Serial.print(" esperando 3 segundos");
            delay(3000);
        }
    }
}

void setup()
{
    Serial.begin(9600);
    SerialESP8266.begin(9600);
    config_wifi();
    pinMode(greenPin, OUTPUT);
    pinMode(redPin, OUTPUT);
    pinMode(ventiladorPin, OUTPUT);
    digitalWrite(ventiladorPin, HIGH);
    pinMode(bombaPin, OUTPUT);
    digitalWrite(bombaPin, HIGH);
    //client.setServer(mqtt_broker,1883);//Conexión al
servidor/Broker
    client.setServer(mqtt_broker, mqtt_port);
    client.setCallback(callback);//Llamada a los callback para
ver si recibo mensajes del broker
}

```

```

void loop()
{
    if (!client.connected()) {
        reconnect();
    }
    client.loop();

    checkSerialCom();
    if (!manual) {
        encendido_luzAuto(l);
        encendidoRiegoAuto(valHumsuelo);
        encendidoVentiladorAuto(t);
    }
}

//////////Metodo para la lectura del monitor serial
void checkSerialCom() {
    if (SerialESP8266.available() > 0)
    {
        //parseo de los valores de los sensores
        SerialESP8266.readBytesUntil('T', buffer, lonbuffer);
        float t = SerialESP8266.parseFloat();

        SerialESP8266.readBytesUntil('H', buffer, lonbuffer);
        float h = SerialESP8266.parseFloat();

        SerialESP8266.readBytesUntil('L', buffer, lonbuffer);
        float l = SerialESP8266.parseFloat();

        SerialESP8266.readBytesUntil('VHS', buffer, lonbuffer);
        float valHumsuelo = SerialESP8266.parseFloat();

        SerialESP8266.readBytesUntil('U', buffer, lonbuffer);
        int distancia = SerialESP8266.parseInt();

        SerialESP8266.readBytesUntil('Mov', buffer, lonbuffer);
        int movimiento = SerialESP8266.parseInt();

        if (movimiento == 1) {
            client.publish("pir", "1"); //el topic se llama pir
            //client.publish("pir", mov); //el topic se llama pir
        } else {
            client.publish("pir", "0"); //el topic se llama pir
        }

        SerialESP8266.readBytesUntil('Mag1', buffer, lonbuffer);
        int sensorMgn1 = SerialESP8266.parseInt();
    }
}

```

```

SerialESP8266.readBytesUntil('Mag2', buffer, lonbuffer);
int sensorMgn2 = SerialESP8266.parseInt();

//Publicar los datos en el respectivo topico
sprintf(data_temp, "%3.2f", t); //dar formato a un numero
entero, flotante, double, etc a String (3 enteros.2 decimales
flotantes)
client.publish("temperatura", data_temp); //el topic se
llama temperatura
Serial.print("Publish message temperatura: ");
Serial.println(data_temp);//msg

sprintf(data_humi, "%3.2f", h); //dar formato a un numero
entero, flotante, double, etc a String (3 enteros.2 decimales
flotantes)
client.publish("humedad", data_humi); //el topic se llama
humedad
Serial.print("Publish message humedad: ");
Serial.println(data_humi);//msg

sprintf(data_lumi, "%3.2f", l); //dar formato a un numero
entero, flotante, double, etc a String (3 enteros.2 decimales
flotantes)
client.publish("luminosidad", data_lumi); //el topic se
llama luminosidad
Serial.print("Publish message luminosidad: ");
Serial.println(data_lumi);//msg

sprintf(data_humiSuelo, "%3.2f", valHumsuelo); //dar
formato a un numero entero, flotante, double, etc a String (3
enteros.2 decimales flotantes)
client.publish("humedadSuelo", data_humiSuelo); //el topic
se llama humedadSuelo
Serial.print("Publish message humedadSuelo: ");
Serial.println(data_humiSuelo);//msg

sprintf(data_dist, "%u", distancia); //dar formato a un
numero entero, flotante, double, etc a String (3 enteros.2
decimales flotantes)
client.publish("distancia", data_dist); //el topic se
llama distanciaObj
Serial.print("Publish message distanciaObj: ");
Serial.print(data_dist);//msg
Serial.println(" cm");

Serial.print("Publish message movimiento: ");
Serial.println(movimiento);//msg

```

```

        sprintf(data_Mag1, "%d", sensorMgn1); //dar formato a un
        numero entero, flotante, double, etc a String (3 enteros.2
        decimales flotantes)
        client.publish("puertaEntrada", data_Mag1); //el topic se
        llama puertaEntrada
        Serial.print("Publish message magnetico 1: ");
        Serial.println(data_Mag1);//msg

        sprintf(data_Mag2, "%d", sensorMgn2); //dar formato a un
        numero entero, flotante, double, etc a String (3 enteros.2
        decimales flotantes)
        client.publish("habitacion1", data_Mag2);

        Serial.print("Publish message magnetico 2: ");
        Serial.println(data_Mag2);//msg

    }

}

//////////Metodos para el trabajo automatico
void encendido_luzAuto(float lumi) {
    if (lumi <= 20.0) {
        //Puerta de entrada
        digitalWrite(redPin, HIGH);
    } else {
        digitalWrite(redPin, LOW);
    }
}

void encendidoRiegoAuto(float humedadSuel) {
    if (humedadSuel <= 50.0) {
        digitalWrite(bombaPin, HIGH);
        delay(1000);
        digitalWrite(bombaPin, LOW);
    }
}

void encendidoVentiladorAuto(float temperatura) {
    if (temperatura > 30.0) { //&& humedad<40
        digitalWrite(ventiladorPin, HIGH);
    } else {
        digitalWrite(ventiladorPin, LOW);
    }
}

```

Código JSON del Node Red

```
[  
  {  
    "id": "cb69e07ed1033f63",  
    "type": "tab",  
    "label": "Flow 3",  
    "disabled": false,  
    "info": "",  
    "env": []  
  },  
  {  
    "id": "7d1bade6b8475eb",  
    "type": "mqtt in",  
    "z": "cb69e07ed1033f63",  
    "name": "",  
    "topic": "distancia",  
    "qos": "2",  
    "datatype": "json",  
    "broker": "171132c7.ece67d",  
    "nl": false,  
    "rap": true,  
    "rh": 0,  
    "inputs": 0,  
    "x": 480,  
    "y": 960,  
    "wires": [  
      [  
        "6da2f8e9bd575003",  
        "2be2b592790d563f",  
        "3cc38ca40c78f65d",  
        "fe2e3c5eeb581346"  
      ]  
    ]  
  },  
  {  
    "id": "6da2f8e9bd575003",  
    "type": "ui_gauge",  
    "z": "cb69e07ed1033f63",  
    "name": "",  
    "group": "624cdb54f883b70c",  
    "order": 23,  
    "width": "2",  
    "height": "3",  
    "gtype": "donut",  
    "title": "Distancia",  
    "label": "cm",  
    "format": "{{value}}"  
  }
```

```
        "min": 0,
        "max": "100",
        "colors": [
            "#00b500",
            "#e6e600",
            "#ca3838"
        ],
        "seg1": "33",
        "seg2": "66",
        "diff": false,
        "className": "",
        "x": 680,
        "y": 960,
        "wires": []
    },
    {
        "id": "74131948b1659792",
        "type": "mqtt in",
        "z": "cb69e07ed1033f63",
        "name": "",
        "topic": "pir",
        "qos": "2",
        "datatype": "json",
        "broker": "171132c7.ece67d",
        "nl": false,
        "rap": true,
        "rh": 0,
        "inputs": 0,
        "x": 470,
        "y": 680,
        "wires": [
            [
                "b76520f9e61543aa",
                "35fbdc6a669ff36b",
                "08c505b304f6bfb2"
            ]
        ]
    },
    {
        "id": "2dc005ce4959b0ee",
        "type": "ui_switch",
        "z": "cb69e07ed1033f63",
        "name": "",
        "label": "regar maceta 1",
        "tooltip": "",
        "group": "0816e6b02af5af5",
        "order": 5,
        "width": 0,
```

```
"height": 0,
"passthru": true,
"decouple": "false",
"topic": "motor_agua",
"topicType": "msg",
"style": "",
"onvalue": "On",
"onvalueType": "str",
"onicon": "",
"oncolor": "",
"offvalue": "Off",
"offvalueType": "str",
"officon": "",
"offcolor": "",
"animate": false,
"className": "",
"x": 1860,
"y": 540,
"wires": [
  [
    "c148e4b42add851a"
  ]
],
},
{
  "id": "c148e4b42add851a",
  "type": "mqtt out",
  "z": "cb69e07ed1033f63",
  "name": "",
  "topic": "esp82iot/bomba",
  "qos": "",
  "retain": "",
  "respTopic": "",
  "contentType": "",
  "userProps": "",
  "correl": "",
  "expiry": "",
  "broker": "171132c7.ece67d",
  "x": 2120,
  "y": 540,
  "wires": []
},
{
  "id": "176bc7eb7b74c599",
  "type": "mqtt in",
  "z": "cb69e07ed1033f63",
  "name": "",
  "topic": "temperatura",
```

```
"qos": "2",
"datatype": "json",
"broker": "171132c7.ece67d",
"nl": false,
"rap": true,
"rh": 0,
"inputs": 0,
"x": 490,
"y": 280,
"wires": [
  [
    "0b88d9438adaad86",
    "8a664556916e2fd7",
    "3d3e41bb830fd339"
  ]
],
{
  "id": "0b88d9438adaad86",
  "type": "ui_gauge",
  "z": "cb69e07ed1033f63",
  "name": "",
  "group": "624cdb54f883b70c",
  "order": 1,
  "width": "2",
  "height": "3",
  "gtype": "gage",
  "title": "Temperatura",
  "label": "°C",
  "format": "{{value}}",
  "min": 0,
  "max": "50",
  "colors": [
    "#00b500",
    "#e6e600",
    "#ca3838"
  ],
  "seg1": "20",
  "seg2": "30",
  "diff": false,
  "className": "",
  "x": 810,
  "y": 260,
  "wires": []
},
{
  "id": "c88e3ff2947e2ccd",
  "type": "ui_gauge",
```

```
"z": "cb69e07ed1033f63",
"name": "",
"group": "624cdb54f883b70c",
"order": 2,
"width": "2",
"height": "3",
"gtype": "wave",
"title": "Humedad",
"label": "%",
"format": "{{value}}",
"min": 0,
"max": "100",
"colors": [
    "#00b500",
    "#e6e600",
    "#ca3838"
],
"seg1": "30",
"seg2": "70",
"diff": false,
"className": "",
"x": 740,
"y": 400,
"wires": []
},
{
"id": "f6dd087843fd99c8",
"type": "mqqt in",
"z": "cb69e07ed1033f63",
"name": "",
"topic": "humedad",
"qos": "2",
"datatype": "json",
"broker": "171132c7.ece67d",
"nl": false,
"rap": true,
"rh": 0,
"inputs": 0,
"x": 480,
"y": 360,
"wires": [
[
    "c88e3ff2947e2ccd",
    "791d980553550108",
    "17fc1daf8395fb94"
]
]
},
{

```



```
"type": "ui_chart",
"z": "cb69e07ed1033f63",
"name": "",
"group": "624cdb54f883b70c",
"order": 1,
"width": "4",
"height": "3",
"label": "Temperatura",
"chartType": "line",
"legend": "false",
"xformat": "HH:mm:ss",
"interpolate": "linear",
"nodata": "",
"dot": false,
"ymin": "0",
"ymax": "60",
"removeOlder": 1,
"removeOlderPoints": "",
"removeOlderUnit": "60",
"cutout": 0,
"useOneColor": false,
"useUTC": false,
"colors": [
    "#1f77b4",
    "#aec7e8",
    "#ff7f0e",
    "#2ca02c",
    "#98df8a",
    "#d62728",
    "#ff9896",
    "#9467bd",
    "#c5b0d5"
],
"outputs": 1,
"useDifferentColor": false,
"className": "",
"x": 810,
"y": 300,
"wires": [
    []
]
},
{
"id": "791d980553550108",
"type": "ui_chart",
"z": "cb69e07ed1033f63",
"name": "",
"group": "624cdb54f883b70c",
```

```
"order": 2,
"width": "4",
"height": "3",
"label": "Humedad",
"chartType": "line",
"legend": "false",
"xformat": "HH:mm:ss",
"interpolate": "linear",
"nodata": "",
"dot": false,
"ymin": "0",
"ymax": "100",
"removeOlder": 1,
"removeOlderPoints": "",
"removeOlderUnit": "3600",
"cutout": 0,
"useOneColor": false,
"useUTC": false,
"colors": [
    "#1f77b4",
    "#aec7e8",
    "#ff7f0e",
    "#2ca02c",
    "#98df8a",
    "#d62728",
    "#ff9896",
    "#9467bd",
    "#c5b0d5"
],
"outputs": 1,
"useDifferentColor": false,
"className": "",
"x": 740,
"y": 440,
"wires": [
    []
]
},
{
"id": "b76520f9e61543aa",
"type": "ui_chart",
"z": "cb69e07ed1033f63",
"name": "",
"group": "0b7f7e36c9b309cf",
"order": 18,
"width": "0",
"height": "0",
"label": "Habitacion 1 (PIR)",
```

```
        "chartType": "line",
        "legend": "false",
        "xformat": "HH:mm:ss",
        "interpolate": "linear",
        "nodata": "",
        "dot": false,
        "ymin": "-1",
        "ymax": "1",
        "removeOlder": 1,
        "removeOlderPoints": "",
        "removeOlderUnit": "60",
        "cutout": 0,
        "useOneColor": false,
        "useUTC": false,
        "colors": [
            "#1f77b4",
            "#aec7e8",
            "#ff7f0e",
            "#2ca02c",
            "#98df8a",
            "#d62728",
            "#ff9896",
            "#9467bd",
            "#c5b0d5"
        ],
        "outputs": 1,
        "useDifferentColor": false,
        "className": "",
        "x": 690,
        "y": 680,
        "wires": [
            []
        ]
    },
{
    "id": "17539bd3482dd3a9",
    "type": "mqtt out",
    "z": "cb69e07ed1033f63",
    "name": "",
    "topic": "esp82iot/ventilador",
    "qos": "",
    "retain": "",
    "respTopic": "",
    "contentType": "",
    "userProps": "",
    "correl": "",
    "expiry": "",
    "broker": "171132c7.ece67d",
```

```
        "x": 2110,
        "y": 480,
        "wires": []
    },
    {
        "id": "6f1bb839adf5c382",
        "type": "ui_switch",
        "z": "cb69e07ed1033f63",
        "name": "",
        "label": "Ventilador",
        "tooltip": "",
        "group": "0816e6b02af5af5",
        "order": 2,
        "width": 0,
        "height": 0,
        "passthru": true,
        "decouple": "false",
        "topic": "esp82iot/ventilador",
        "topicType": "msg",
        "style": "",
        "onvalue": "On",
        "onvalueType": "str",
        "onicon": "",
        "oncolor": "",
        "offvalue": "Off",
        "offvalueType": "str",
        "officon": "",
        "offcolor": "",
        "animate": false,
        "className": "",
        "x": 1840,
        "y": 480,
        "wires": [
            [
                "17539bd3482dd3a9"
            ]
        ]
    },
    {
        "id": "742e4ff4bfb49b15",
        "type": "mqtt out",
        "z": "cb69e07ed1033f63",
        "name": "",
        "topic": "esp82iot",
        "qos": "",
        "retain": "",
        "respTopic": "",
        "contentType": ""
    }
]
```

```
"userProps": "",  
"correl": "",  
"expiry": "",  
"broker": "171132c7.ece67d",  
"x": 2140,  
"y": 600,  
"wires": []  
},  
{  
    "id": "58e21b79967b97d6",  
    "type": "mqtt out",  
    "z": "cb69e07ed1033f63",  
    "name": "",  
    "topic": "esp82iot",  
    "qos": "",  
    "retain": "",  
    "respTopic": "",  
    "contentType": "",  
    "userProps": "",  
    "correl": "",  
    "expiry": "",  
    "broker": "171132c7.ece67d",  
    "x": 2140,  
    "y": 660,  
    "wires": []  
},  
{  
    "id": "e32a4f42a1515c34",  
    "type": "ui_switch",  
    "z": "cb69e07ed1033f63",  
    "name": "",  
    "label": "Foco de puerta de Entrada",  
    "tooltip": "",  
    "group": "0816e6b02af5af5",  
    "order": 3,  
    "width": 0,  
    "height": 0,  
    "passthru": true,  
    "decouple": "false",  
    "topic": "topic",  
    "topicType": "msg",  
    "style": "",  
    "onvalue": "redOn",  
    "onvalueType": "str",  
    "onicon": "",  
    "oncolor": "",  
    "offvalue": "redOff",  
    "offvalueType": "str",  
}
```

```
        "officon": "",  
        "offcolor": "",  
        "animate": false,  
        "className": "",  
        "x": 1900,  
        "y": 600,  
        "wires": [  
            [  
                "742e4ff4bfb49b15"  
            ]  
        ]  
    },  
    {  
        "id": "3ad856ec31a51984",  
        "type": "ui_switch",  
        "z": "cb69e07ed1033f63",  
        "name": "",  
        "label": "Foco de Habitación 1",  
        "tooltip": "",  
        "group": "0816e6b02af5af5",  
        "order": 4,  
        "width": 0,  
        "height": 0,  
        "passthru": true,  
        "decouple": "false",  
        "topic": "topic",  
        "topicType": "msg",  
        "style": "",  
        "onvalue": "greenOn",  
        "onvalueType": "str",  
        "onicon": "",  
        "oncolor": "",  
        "offvalue": "greenOff",  
        "offvalueType": "str",  
        "officon": "",  
        "offcolor": "",  
        "animate": false,  
        "className": "",  
        "x": 1880,  
        "y": 660,  
        "wires": [  
            [  
                "58e21b79967b97d6"  
            ]  
        ]  
    },  
    {  
        "id": "fe9d97b706f8fa54",  
        "type": "ui_text",  
        "z": "cb69e07ed1033f63",  
        "name": "",  
        "label": "Foco de Habitación 2",  
        "tooltip": "",  
        "group": "0816e6b02af5af5",  
        "order": 5,  
        "width": 0,  
        "height": 0,  
        "passthru": true,  
        "decouple": "false",  
        "topic": "topic",  
        "topicType": "msg",  
        "style": "",  
        "onvalue": "greenOn",  
        "onvalueType": "str",  
        "onicon": "",  
        "oncolor": "",  
        "offvalue": "greenOff",  
        "offvalueType": "str",  
        "officon": "",  
        "offcolor": "",  
        "animate": false,  
        "className": "",  
        "x": 1880,  
        "y": 720,  
        "wires": [  
            [  
                "58e21b79967b97d6"  
            ]  
        ]  
    },  
    {  
        "id": "3a0a23c53a5a0a54",  
        "type": "ui_text",  
        "z": "cb69e07ed1033f63",  
        "name": "",  
        "label": "Foco de Habitación 3",  
        "tooltip": "",  
        "group": "0816e6b02af5af5",  
        "order": 6,  
        "width": 0,  
        "height": 0,  
        "passthru": true,  
        "decouple": "false",  
        "topic": "topic",  
        "topicType": "msg",  
        "style": "",  
        "onvalue": "greenOn",  
        "onvalueType": "str",  
        "onicon": "",  
        "oncolor": "",  
        "offvalue": "greenOff",  
        "offvalueType": "str",  
        "officon": "",  
        "offcolor": "",  
        "animate": false,  
        "className": "",  
        "x": 1880,  
        "y": 780,  
        "wires": [  
            [  
                "58e21b79967b97d6"  
            ]  
        ]  
    }]
```

```

    "type": "function",
    "z": "cb69e07ed1033f63",
    "name": "function 1",
    "func": "if (msg.payload == 1) {\n      var d = new\nDate();\n      var provincia = \"Lima\"\n      var distrito =\n\"SJL\"\n      var sitio = \"Hogar\"\n      var id\n      var day =\nd.getDate()\n      var month = d.getMonth()\n      month = month +\n1;\n      var year = d.getFullYear()\n      var hours = d.getHours()\nvar minutes = d.getMinutes()\n      var seconds = d.getSeconds()\n\nif (day <= 9)\n      day = \"0\" + day;\n      if (month <= 9)\nmonth = \"0\" + month;\n      if (hours <= 9)\nhours = \"0\" +\nhours;\n      if (minutes <= 9)\nminutes = \"0\" +\nminutes;\n      if (seconds <= 9)\nseconds = \"0\" +\nseconds;\n\n      var date = day + \"/\" + month + \"/\" + year\nvar time = hours + \":\" + minutes + \":\" + seconds\n\nmsg.payload =\n      \"IoT dispositivo de movimiento\"\n      +\n\"=====\"\n      +\n\"ACTIVADO, MOVIMIENTO DETECTADO EN HABITACION 1!!!\"\n      +\n\"=====\"\n      +\n\"Fecha: \" + date + \"\\n\" +\n      \"Hora: \" + time +\n\"\\n\" +\n      \"Provincia: \" + provincia + \"\\n\" +\n\"Distrito: \" + distrito + \"\\n\" +\n      \"Sitio: \" + sitio\n+ \"\\n\"\n      return msg;\n}\nreturn null;\n",
    "outputs": 1,
    "noerr": 0,
    "initialize": "",
    "finalize": "",
    "libs": [],
    "x": 880,
    "y": 720,
    "wires": [
      [
        "5173a0063d6cd1ba"
      ]
    ]
  },
  {
    "id": "5173a0063d6cd1ba",
    "type": "node-red-contrib-whatsapp-cmb-send-message",
    "z": "cb69e07ed1033f63",
    "name": "WA cliente",
    "account": "990c808eb3798486",
    "text": "payload",
    "inputtypemessage": "msg",
    "rejectssl": false,
    "x": 1050,
    "y": 720,
    "wires": [

```

```
        []
    ],
},
{
  "id": "35fbdc6a669ff36b",
  "type": "delay",
  "z": "cb69e07ed1033f63",
  "name": "",
  "pauseType": "rate",
  "timeout": "5",
  "timeoutUnits": "seconds",
  "rate": "1",
  "nbRateUnits": "5",
  "rateUnits": "second",
  "randomFirst": "1",
  "randomLast": "5",
  "randomUnits": "seconds",
  "drop": true,
  "allowrate": true,
  "outputs": 1,
  "x": 680,
  "y": 720,
  "wires": [
    [
      "fe9d97b706f8fa54"
    ]
  ]
},
{
  "id": "e10170b42cdba5ca",
  "type": "mqqt in",
  "z": "cb69e07ed1033f63",
  "name": "",
  "topic": "puertaEntrada",
  "qos": "2",
  "datatype": "auto-detect",
  "broker": "171132c7.ece67d",
  "nl": false,
  "rap": true,
  "rh": 0,
  "inputs": 0,
  "x": 490,
  "y": 1100,
  "wires": [
    [
      "406831e306939e43",
      "7e627a62ba3a986b",
      "191e01f86718530a"
    ]
  ]
},
```

```
        ]
    ]
},
{
  "id": "8b3bb4688b978cb3",
  "type": "mqtt in",
  "z": "cb69e07ed1033f63",
  "name": "",
  "topic": "habitacion1",
  "qos": "2",
  "datatype": "auto-detect",
  "broker": "171132c7.ece67d",
  "nl": false,
  "rap": true,
  "rh": 0,
  "inputs": 0,
  "x": 490,
  "y": 1240,
  "wires": [
    [
      "2b9cf656b70e1c16",
      "fbb9196b4de4a453",
      "0a14cc909792750b"
    ]
  ]
},
{
  "id": "7e627a62ba3a986b",
  "type": "ui_chart",
  "z": "cb69e07ed1033f63",
  "name": "",
  "group": "0b7f7e36c9b309cf",
  "order": 19,
  "width": "0",
  "height": "0",
  "label": "Puerta de entrada (SENSOR MAGNETICO)",
  "chartType": "line",
  "legend": "false",
  "xformat": "HH:mm:ss",
  "interpolate": "linear",
  "nodata": "",
  "dot": false,
  "ymin": "-1",
  "ymax": "1",
  "removeOlder": 1,
  "removeOlderPoints": "",
  "removeOlderUnit": "60",
  "cutout": 0,
```

```
"useOneColor": false,
"useUTC": false,
"colors": [
    "#1f77b4",
    "#aec7e8",
    "#ff7f0e",
    "#2ca02c",
    "#98df8a",
    "#d62728",
    "#ff9896",
    "#9467bd",
    "#c5b0d5"
],
"outputs": 1,
"useDifferentColor": false,
"className": "",
"x": 830,
"y": 1100,
"wires": [
    []
]
},
{
    "id": "fbb9196b4de4a453",
    "type": "ui_chart",
    "z": "cb69e07ed1033f63",
    "name": "",
    "group": "0b7f7e36c9b309cf",
    "order": 20,
    "width": "0",
    "height": "0",
    "label": "Puerta de habitación 1 (SENSOR MAGNETICO)",
    "chartType": "line",
    "legend": "false",
    "xformat": "HH:mm:ss",
    "interpolate": "linear",
    "nodata": "",
    "dot": false,
    "ymin": "-1",
    "ymax": "1",
    "removeOlder": 1,
    "removeOlderPoints": "",
    "removeOlderUnit": "60",
    "cutout": 0,
    "useOneColor": false,
    "useUTC": false,
    "colors": [
        "#1f77b4",
```

```
        "#aec7e8",
        "#ff7f0e",
        "#2ca02c",
        "#98df8a",
        "#d62728",
        "#ff9896",
        "#9467bd",
        "#c5b0d5"
    ],
    "outputs": 1,
    "useDifferentColor": false,
    "className": "",
    "x": 800,
    "y": 1240,
    "wires": [
        []
    ]
},
{
    "id": "20d435c70631d2fc",
    "type": "node-red-contrib-whatsapp-cmb-send-message",
    "z": "cb69e07ed1033f63",
    "name": "",
    "account": "990c808eb3798486",
    "text": "payload",
    "inputtypemessage": "msg",
    "rejectssl": false,
    "x": 900,
    "y": 1140,
    "wires": [
        []
    ]
},
{
    "id": "5c69115c91618156",
    "type": "node-red-contrib-whatsapp-cmb-send-message",
    "z": "cb69e07ed1033f63",
    "name": "WA cliente",
    "account": "990c808eb3798486",
    "text": "payload",
    "inputtypemessage": "msg",
    "rejectssl": false,
    "x": 850,
    "y": 1280,
    "wires": [
        []
    ]
},
```

```

{
  "id": "406831e306939e43",
  "type": "function",
  "z": "cb69e07ed1033f63",
  "name": "function 6",
  "func": "if (msg.payload == 1) {\n      var d = new\nDate();\n      var provincia = \"Lima\"\n      var distrito =\n\"SJL\"\n      var sitio = \"Hogar\"\n      var id\n      var day =\nd.getDate()\n      var month = d.getMonth()\n      month = month +\n1;\n      var year = d.getFullYear()\n      var hours = d.getHours()\nvar minutes = d.getMinutes()\n      var seconds = d.getSeconds()\n\nif (day <= 9)\n      day = \"0\" + day;\n      if (month <= 9)\nmonth = \"0\" + month;\n      if (hours <= 9)\nhours = \"0\" +\nhours;\n      if (minutes <= 9)\nminutes = \"0\" +\nminutes;\n      if (seconds <= 9)\nseconds = \"0\" +\nseconds;\n\n      var date = day + \"/\" + month + \"/\" + year\nvar time = hours + \":\" + minutes + \":\" + seconds\n\nmsg.payload =\n      \"IoT sensor magnetico\"\n      +\n\"=====\"\n\n\"ACTIVADO, LA PUERTA DE LA ENTRADA HA SIDO ABIERTA!!!\"\n\n\"=====\"\n\n\"Fecha: \" + date + \"\\n\" +\n      \"Hora: \" + time +\n\"\\n\" +\n      \"Provincia: \" + provincia + \"\\n\" +\n\"Distrito: \" + distrito + \"\\n\" +\n      \"Sitio: \" + sitio\n+ \"\\n\"\n      return msg;\n\n      return null;\n\n      \"outputs\": 1,\n      \"noerr\": 0,\n      \"initialize\": \"\",\n      \"finalize\": \"\",\n      \"libs\": [],\n      \"x\": 720,\n      \"y\": 1140,\n      \"wires\": [\n        [\n          \"20d435c70631d2fc\"\n        ]\n      ]\n    },\n    {\n      \"id\": \"2b9cf656b70e1c16\",\n      \"type\": \"function\",\n      \"z\": \"cb69e07ed1033f63\",\n      \"name\": \"function 7\",\n      \"func\": \"if (msg.payload == 1) {\n      var d = new\nDate();\n      var provincia = \"Lima\"\n      var distrito =\n\"SJL\"\n      var sitio = \"Hogar\"\n      var id\n      var day =\nd.getDate()\n      var month = d.getMonth()\n      month = month +\n1;\n      var year = d.getFullYear()\n      var hours = d.getHours()\n\nif (day <= 9)\n      day = \"0\" + day;\n      if (month <= 9)\nmonth = \"0\" + month;\n      if (hours <= 9)\nhours = \"0\" +\nhours;\n      if (minutes <= 9)\nminutes = \"0\" +\nminutes;\n      if (seconds <= 9)\nseconds = \"0\" +\nseconds;\n\n      var date = day + \"/\" + month + \"/\" + year\nvar time = hours + \":\" + minutes + \":\" + seconds\n\nmsg.payload =\n      \"IoT sensor magnetico\"\n      +\n\"=====\"\n\n\"ACTIVADO, LA PUERTA DE LA ENTRADA HA SIDO ABIERTA!!!\"\n\n\"=====\"\n\n\"Fecha: \" + date + \"\\n\" +\n      \"Hora: \" + time +\n\"\\n\" +\n      \"Provincia: \" + provincia + \"\\n\" +\n\"Distrito: \" + distrito + \"\\n\" +\n      \"Sitio: \" + sitio\n+ \"\\n\"\n      return msg;\n\n      return null;\n\n      \"outputs\": 1,\n      \"noerr\": 0,\n      \"initialize\": \"\",\n      \"finalize\": \"\",\n      \"libs\": [],\n      \"x\": 720,\n      \"y\": 1140,\n      \"wires\": [\n        [\n          \"20d435c70631d2fc\"\n        ]\n      ]\n    }\n  ]\n}
```

```

var minutes = d.getMinutes()\n      var seconds = d.getSeconds()\n\n
if (day <= 9)\n          day = \"0\" + day;\n          if (month <= 9)\n
month = \"0\" + month;\n          if (hours <= 9)\n              hours = \"0\" +\n+ hours;\n          if (minutes <= 9)\n              minutes = \"0\" +\nminutes;\n          if (seconds <= 9)\n              seconds = \"0\" +\nseconds;\n\n      var date = day + \"/\" + month + \"/\" + year\n\n
var time = hours + \":\" + minutes + \":\" + seconds\n\n
msg.payload =\n          \"IoT sensor magnetico\"\n          +\n          \"=====\"\n          +\n          \"ACTIVADO, LA PUERTA DE LA ENTRADA HABITACION 1 HA \\\n          +\n          \"SIDO ABIERTA!!!\"\n          +\n          \"=====\"\n          +\n          \"Fecha:\n          \" + date + \"\\n\"\n          +\n          \"Hora: \" + time + \"\\n\"\n          +\n          \"Provincia: \" + provincia + \"\\n\"\n          +\n          \"Distrito: \" +\ndistrito + \"\\n\"\n          +\n          \"Sito: \" + sitio + \"\\n\"\n\nreturn msg;\n}\n\nreturn null;\n",
        "outputs": 1,
        "noerr": 0,
        "initialize": "",
        "finalize": "",
        "libs": [],
        "x": 680,
        "y": 1280,
        "wires": [
            [
                "5c69115c91618156"
            ]
        ]
    },
    {
        "id": "1cdf46798927d1c0",
        "type": "ui_gauge",
        "z": "cb69e07ed1033f63",
        "name": "",
        "group": "624cdb54f883b70c",
        "order": 19,
        "width": "2",
        "height": "3",
        "gtype": "gage",
        "title": "Luminosidad del ambiente",
        "label": "lux",
        "format": "{{value}}",
        "min": 0,
        "max": "50",
        "colors": [
            "#292929",
            "#e6e600",
            "#ca3838"
        ]
    }
}

```

```
],
  "seg1": "15",
  "seg2": "30",
  "diff": false,
  "className": "",
  "x": 750,
  "y": 820,
  "wires": []
},
{
  "id": "d527f109740300f9",
  "type": "ui_chart",
  "z": "cb69e07ed1033f63",
  "name": "",
  "group": "624cdb54f883b70c",
  "order": 20,
  "width": "4",
  "height": "3",
  "label": "Luminosidad vs tiempo",
  "chartType": "line",
  "legend": "false",
  "xformat": "HH:mm:ss",
  "interpolate": "linear",
  "nodata": "",
  "dot": false,
  "ymin": "0",
  "ymax": "50",
  "removeOlder": 1,
  "removeOlderPoints": "",
  "removeOlderUnit": "60",
  "cutout": 0,
  "useOneColor": false,
  "useUTC": false,
  "colors": [
    "#1f77b4",
    "#aec7e8",
    "#ff7f0e",
    "#2ca02c",
    "#98df8a",
    "#d62728",
    "#ff9896",
    "#9467bd",
    "#c5b0d5"
  ],
  "outputs": 1,
  "useDifferentColor": false,
  "className": "",
  "x": 740,
```

```
"y": 860,
"wires": [
    []
]
},
{
    "id": "b4ca4fef4f8a958c",
    "type": "ui_chart",
    "z": "cb69e07ed1033f63",
    "name": "",
    "group": "624cdb54f883b70c",
    "order": 22,
    "width": "4",
    "height": "3",
    "label": "Humedad vs tiempo",
    "chartType": "line",
    "legend": "false",
    "xformat": "HH:mm:ss",
    "interpolate": "linear",
    "nodata": "",
    "dot": false,
    "ymin": "0",
    "ymax": "100",
    "removeOlder": 1,
    "removeOlderPoints": "",
    "removeOlderUnit": "60",
    "cutout": 0,
    "useOneColor": false,
    "useUTC": false,
    "colors": [
        "#1f77b4",
        "#aec7e8",
        "#ff7f0e",
        "#2ca02c",
        "#98df8a",
        "#d62728",
        "#ff9896",
        "#9467bd",
        "#c5b0d5"
    ],
    "outputs": 1,
    "useDifferentColor": false,
    "className": "",
    "x": 780,
    "y": 580,
    "wires": [
        []
]
}
```

```
},
{
  "id": "03178388431cccfb",
  "type": "ui_gauge",
  "z": "cb69e07ed1033f63",
  "name": "",
  "group": "624cdb54f883b70c",
  "order": 21,
  "width": "2",
  "height": "3",
  "gtype": "wave",
  "title": "Humedad maceta 1",
  "label": "%",
  "format": "{{value}}",
  "min": 0,
  "max": "100",
  "colors": [
    "#00b500",
    "#e6e600",
    "#ca3838"
  ],
  "seg1": "",
  "seg2": "",
  "diff": false,
  "className": "",
  "x": 770,
  "y": 540,
  "wires": []
},
{
  "id": "2be2b592790d563f",
  "type": "ui_chart",
  "z": "cb69e07ed1033f63",
  "name": "",
  "group": "624cdb54f883b70c",
  "order": 24,
  "width": "4",
  "height": "3",
  "label": "Distancia vs tiempo",
  "chartType": "line",
  "legend": "false",
  "xformat": "HH:mm:ss",
  "interpolate": "linear",
  "nodata": "",
  "dot": false,
  "ymin": "0",
  "ymax": "100",
  "removeOlder": 1,
```

```
"removeOlderPoints": "",  
"removeOlderUnit": "60",  
"cutout": 0,  
"useOneColor": false,  
"useUTC": false,  
"colors": [  
    "#1f77b4",  
    "#aec7e8",  
    "#ff7f0e",  
    "#2ca02c",  
    "#98df8a",  
    "#d62728",  
    "#ff9896",  
    "#9467bd",  
    "#c5b0d5"  
,  
    ],  
    "outputs": 1,  
    "useDifferentColor": false,  
    "className": "",  
    "x": 730,  
    "y": 1000,  
    "wires": [  
        []  
    ]  
},  
{  
    "id": "d980da34163ebf27",  
    "type": "ui_switch",  
    "z": "cb69e07ed1033f63",  
    "name": "",  
    "label": "Manual",  
    "tooltip": "",  
    "group": "0816e6b02af5af5",  
    "order": 1,  
    "width": 0,  
    "height": 0,  
    "passthru": true,  
    "decouple": "false",  
    "topic": "topic",  
    "topicType": "msg",  
    "style": "",  
    "onvalue": "On",  
    "onvalueType": "str",  
    "onicon": "",  
    "oncolor": "",  
    "offvalue": "Off",  
    "offvalueType": "str",  
    "officon": "",
```

```
        "offcolor": "",  
        "animate": false,  
        "className": "",  
        "x": 1840,  
        "y": 720,  
        "wires": [  
            [  
                "d22abb196bbd773a"  
            ]  
        ]  
    },  
    {  
        "id": "d22abb196bbd773a",  
        "type": "mqtt out",  
        "z": "cb69e07ed1033f63",  
        "name": "",  
        "topic": "esp82iot/manual",  
        "qos": "",  
        "retain": "",  
        "respTopic": "",  
        "contentType": "",  
        "userProps": "",  
        "correl": "",  
        "expiry": "",  
        "broker": "171132c7.ece67d",  
        "x": 2120,  
        "y": 720,  
        "wires": []  
    },  
    {  
        "id": "3d3e41bb830fd339",  
        "type": "function",  
        "z": "cb69e07ed1033f63",  
        "name": "function 14",  
        "func": "var d = new Date();\nvar date =  
d.getFullYear() + "/" + (d.getMonth() + 1) + "/" + d.getDate() + ";" + d.g  
etHours() + ":" + d.getMinutes() + ":" + d.getSeconds();\nmsg.payload  
= msg.payload + ";" + date\nreturn msg;",  
        "outputs": 1,  
        "noerr": 0,  
        "initialize": "",  
        "finalize": "",  
        "libs": [],  
        "x": 810,  
        "y": 200,  
        "wires": [  
            [  
                "1001c6f11b1f2a1f"  
            ]  
        ]  
    }]
```

```
        ]
    ]
},
{
    "id": "1001c6f11b1f2a1f",
    "type": "file",
    "z": "cb69e07ed1033f63",
    "name": "",
    "filename": "E:\\UNIVERSIDAD\\2023-I\\COMPUTACIÓN  
FÍSICA\\ESP8266\\Proyecto personal\\Proyecto Computación  
Física\\Mediciones\\Temperatura.csv",
    "filenameType": "str",
    "appendNewline": true,
    "createDir": true,
    "overwriteFile": "false",
    "encoding": "none",
    "x": 1400,
    "y": 200,
    "wires": [
        [
            "297de050d0295a37"
        ]
    ]
},
{
    "id": "297de050d0295a37",
    "type": "debug",
    "z": "cb69e07ed1033f63",
    "name": "debug 6",
    "active": true,
    "tosidebar": true,
    "console": false,
    "tostatus": false,
    "complete": "true",
    "targetType": "full",
    "statusVal": "",
    "statusType": "auto",
    "x": 1620,
    "y": 260,
    "wires": []
},
{
    "id": "17fc1daf8395fb94",
    "type": "function",
    "z": "cb69e07ed1033f63",
    "name": "function 15",
    "func": "var d = new Date();\nvar date = d.getFullYear() +  
\"/\" + (d.getMonth() + 1) + \"/\" + d.getDate() + \";\" +  
\"\"
```

```
d.getHours() + \":\" + d.getMinutes() + \":\" +
d.getSeconds();\nmsg.payload = msg.payload + \";\" + date\nreturn
msg;",
    "outputs": 1,
    "noerr": 0,
    "initialize": "",
    "finalize": "",
    "libs": [],
    "x": 750,
    "y": 360,
    "wires": [
        [
            "eac8c0006226569f"
        ]
    ]
},
{
    "id": "eac8c0006226569f",
    "type": "file",
    "z": "cb69e07ed1033f63",
    "name": "",
    "filename": "E:\\UNIVERSIDAD\\2023-I\\COMPUTACIÓN
FÍSICA\\ESP8266\\Proyecto personal\\Proyecto Computación
Física\\Mediciones\\humedad.csv",
    "filenameType": "str",
    "appendNewline": true,
    "createDir": true,
    "overwriteFile": "false",
    "encoding": "none",
    "x": 1310,
    "y": 360,
    "wires": [
        [
            "be3d9c01e7efd195"
        ]
    ]
},
{
    "id": "be3d9c01e7efd195",
    "type": "debug",
    "z": "cb69e07ed1033f63",
    "name": "debug 7",
    "active": true,
    "tosidebar": true,
    "console": false,
    "tostatus": false,
    "complete": "payload",
    "targetType": "msg",
```

```

    "statusVal": "",
    "statusType": "auto",
    "x": 1560,
    "y": 360,
    "wires": []
},
{
    "id": "419ea4e090a7d9a4",
    "type": "function",
    "z": "cb69e07ed1033f63",
    "name": "function 16",
    "func": "if (msg.payload==2) \\n      msg.payload=\"Activar\nmotor de agua\"\nvar d = new Date();\\nvar date = d.getFullYear() +\n\"/\" + (d.getMonth() + 1) + \"/\" + d.getDate() + \";\" +\nd.getHours() + \":\" + d.getMinutes() + \":\" +\nd.getSeconds();\\nmsg.payload = msg.payload + \";\" + date\\nreturn\nmsg;",
    "outputs": 1,
    "noerr": 0,
    "initialize": "",
    "finalize": "",
    "libs": [],
    "x": 750,
    "y": 500,
    "wires": [
        [
            "5865db722dbd06de"
        ]
    ]
},
{
    "id": "5865db722dbd06de",
    "type": "file",
    "z": "cb69e07ed1033f63",
    "name": "",
    "filename": "E:\\UNIVERSIDAD\\2023-I\\COMPUTACIÓN\nFÍSICA\\ESP8266\\Proyecto personal\\Proyecto Computación\nFísica\\Mediciones\\humedadsuelo.csv",
    "filenameType": "str",
    "appendNewline": true,
    "createDir": true,
    "overwriteFile": "false",
    "encoding": "none",
    "x": 1350,
    "y": 500,
    "wires": [
        [
            "d545e45e51c01a00"
        ]
    ]
}

```

```
        ],
    ],
},
{
    "id": "d545e45e51c01a00",
    "type": "debug",
    "z": "cb69e07ed1033f63",
    "name": "debug 8",
    "active": true,
    "tosidebar": true,
    "console": false,
    "tostatus": false,
    "complete": "payload",
    "targetType": "msg",
    "statusVal": "",
    "statusType": "auto",
    "x": 1620,
    "y": 500,
    "wires": []
},
{
    "id": "5943a6859f78996f",
    "type": "file",
    "z": "cb69e07ed1033f63",
    "name": "",
    "filename": "E:\\UNIVERSIDAD\\2023-I\\COMPUTACIÓN FÍSICA\\ESP8266\\Proyecto personal\\Proyecto Computación Física\\Mediciones\\pir.csv",
    "filenameType": "str",
    "appendNewline": true,
    "createDir": true,
    "overwriteFile": "false",
    "encoding": "none",
    "x": 1230,
    "y": 640,
    "wires": [
        [
            "b535b1ab0cfcad72"
        ]
    ]
},
{
    "id": "08c505b304f6fb2",
    "type": "function",
    "z": "cb69e07ed1033f63",
    "name": "function 17",
    "func": "var d = new Date();\nvar date = d.getFullYear() + \"/\" + (d.getMonth() + 1) + \"/\" + d.getDate() + \";\" + "
}
```

```
d.getHours() + \":\:\" + d.getMinutes() + \":\:\" +
d.getSeconds();\nmsg.payload = msg.payload + \";\:\" + date\nreturn
msg;",
    "outputs": 1,
    "noerr": 0,
    "initialize": "",
    "finalize": "",
    "libs": [],
    "x": 690,
    "y": 640,
    "wires": [
        [
            "5943a6859f78996f"
        ]
    ]
},
{
    "id": "b535b1ab0cfcad72",
    "type": "debug",
    "z": "cb69e07ed1033f63",
    "name": "debug 9",
    "active": true,
    "tosidebar": true,
    "console": false,
    "tostatus": false,
    "complete": "payload",
    "targetType": "msg",
    "statusVal": "",
    "statusType": "auto",
    "x": 1640,
    "y": 720,
    "wires": []
},
{
    "id": "870af814b65e95b7",
    "type": "function",
    "z": "cb69e07ed1033f63",
    "name": "function 18",
    "func": "var d = new Date();\nvar date = d.getFullYear() +
\"/\\" + (d.getMonth() + 1) + \"/\\\" + d.getDate() + \";\\\" +
d.getHours() + \":\:\" + d.getMinutes() + \":\:\" +
d.getSeconds();\nmsg.payload = msg.payload + \";\:\" + date\nreturn
msg;",
    "outputs": 1,
    "noerr": 0,
    "initialize": "",
    "finalize": "",
    "libs": []
```

```
        "x": 710,
        "y": 780,
        "wires": [
            [
                "6ab76dd136644f1e"
            ]
        ]
    },
{
    "id": "6ab76dd136644f1e",
    "type": "file",
    "z": "cb69e07ed1033f63",
    "name": "",
    "filename": "E:\\UNIVERSIDAD\\2023-I\\COMPUTACIÓN FÍSICA\\ESP8266\\Proyecto personal\\Proyecto Computación Física\\Mediciones\\luminosidad.csv",
    "filenameType": "str",
    "appendNewline": true,
    "createDir": true,
    "overwriteFile": "false",
    "encoding": "none",
    "x": 1260,
    "y": 780,
    "wires": [
        [
            "4614427b9680eceb"
        ]
    ]
},
{
    "id": "4614427b9680eceb",
    "type": "debug",
    "z": "cb69e07ed1033f63",
    "name": "debug 10",
    "active": true,
    "tosidebar": true,
    "console": false,
    "tostatus": false,
    "complete": "payload",
    "targetType": "msg",
    "statusVal": "",
    "statusType": "auto",
    "x": 1500,
    "y": 780,
    "wires": []
},
{
    "id": "3cc38ca40c78f65d",
```

```
        "type": "function",
        "z": "cb69e07ed1033f63",
        "name": "function 19",
        "func": "var d = new Date();\nvar date = d.getFullYear() +
\"/\\" + (d.getMonth() + 1) + \"/\" + d.getDate() + \";\" +
d.getHours() + \":\" + d.getMinutes() + \":\" +
d.getSeconds();\nmsg.payload = msg.payload + \";\" + date\nreturn
msg;",
        "outputs": 1,
        "noerr": 0,
        "initialize": "",
        "finalize": "",
        "libs": [],
        "x": 690,
        "y": 920,
        "wires": [
            [
                "1f37c273ffed3e7b"
            ]
        ]
    },
    {
        "id": "1f37c273ffed3e7b",
        "type": "file",
        "z": "cb69e07ed1033f63",
        "name": "",
        "filename": "E:\\UNIVERSIDAD\\2023-I\\COMPUTACIÓN
FÍSICA\\ESP8266\\Proyecto personal\\Proyecto Computación
Física\\Mediciones\\distancia.csv",
        "filenameType": "str",
        "appendNewline": true,
        "createDir": true,
        "overwriteFile": "false",
        "encoding": "none",
        "x": 1250,
        "y": 920,
        "wires": [
            []
        ]
    },
    {
        "id": "191e01f86718530a",
        "type": "function",
        "z": "cb69e07ed1033f63",
        "name": "function 20",
        "func": "var d = new Date();\nvar date = d.getFullYear() +
\"/\\" + (d.getMonth() + 1) + \"/\" + d.getDate() + \";\" +
d.getHours() + \":\" + d.getMinutes() + \":\" +
```

```
d.getSeconds();\nmsg.payload = msg.payload + \";\" + date\nreturn\nmsg;,\n    \"outputs\": 1,\n    \"noerr\": 0,\n    \"initialize\": \"\",\n    \"finalize\": \"\",\n    \"libs\": [],\n    \"x\": 730,\n    \"y\": 1060,\n    \"wires\": [\n        [\n            \"caled1a7ac690ac4\"\n        ]\n    ]\n},\n{\n    \"id\": \"caled1a7ac690ac4\",,\n    \"type\": \"file\",,\n    \"z\": \"cb69e07ed1033f63\",,\n    \"name\": \"\",,\n    \"filename\": \"E:\\UNIVERSIDAD\\2023-I\\COMPUTACIÓN\nFÍSICA\\ESP8266\\Proyecto personal\\Proyecto Computación\nFísica\\Mediciones\\puertaentrada.csv\",,\n    \"filenameType\": \"str\",,\n    \"appendNewline\": true,,\n    \"createDir\": true,,\n    \"overwriteFile\": \"false\",,\n    \"encoding\": \"none\",,\n    \"x\": 1310,,\n    \"y\": 1060,,\n    \"wires\": [\n        []\n    ]\n},\n{\n    \"id\": \"a530280ffe8fccc7\",,\n    \"type\": \"file\",,\n    \"z\": \"cb69e07ed1033f63\",,\n    \"name\": \"\",,\n    \"filename\": \"E:\\UNIVERSIDAD\\2023-I\\COMPUTACIÓN\nFÍSICA\\ESP8266\\Proyecto personal\\Proyecto Computación\nFísica\\Mediciones\\habitacion.csv\",,\n    \"filenameType\": \"str\",,\n    \"appendNewline\": true,,\n    \"createDir\": true,,\n    \"overwriteFile\": \"false\",,\n    \"encoding\": \"none\",,\n    \"x\": 1260,,
```

```
        "y": 1200,
        "wires": [
            []
        ]
    },
{
    "id": "0a14cc909792750b",
    "type": "function",
    "z": "cb69e07ed1033f63",
    "name": "function 21",
    "func": "var d = new Date();\nvar date = d.getFullYear() +\n\"/\" + (d.getMonth() + 1) + \"/\" + d.getDate() + \";\" +\nd.getHours() + \":\" + d.getMinutes() + \":\" +\nd.getSeconds();\nmsg.payload = msg.payload + \";\" + date\nreturn\nmsg;",
    "outputs": 1,
    "noerr": 0,
    "initialize": "",
    "finalize": "",
    "libs": [],
    "x": 690,
    "y": 1200,
    "wires": [
        [
            "a530280ffe8fccc7"
        ]
    ]
},
{
    "id": "39a2fa354e9af781",
    "type": "ui_audio",
    "z": "cb69e07ed1033f63",
    "name": "",
    "group": "316f83b78a6c2eb8",
    "voice": "Microsoft Helena - Spanish (Spain)",
    "always": true,
    "x": 1040,
    "y": 1380,
    "wires": []
},
{
    "id": "da3663f8f72f9051",
    "type": "function",
    "z": "cb69e07ed1033f63",
    "name": "function 22",
    "func": "if(msg.payload < 4){\n    msg.payload = \"Se\nregistra presencia en puerta de entrada\"\n}\nreturn msg;",
    "outputs": 1,
```

```
        "noerr": 0,
        "initialize": "",
        "finalize": "",
        "libs": [],
        "x": 870,
        "y": 1380,
        "wires": [
          [
            "39a2fa354e9af781"
          ]
        ]
      },
      {
        "id": "fe2e3c5eeb581346",
        "type": "delay",
        "z": "cb69e07ed1033f63",
        "name": "",
        "pauseType": "rate",
        "timeout": "5",
        "timeoutUnits": "seconds",
        "rate": "1",
        "nbRateUnits": "5",
        "rateUnits": "second",
        "randomFirst": "1",
        "randomLast": "5",
        "randomUnits": "seconds",
        "drop": true,
        "allowrate": true,
        "outputs": 1,
        "x": 680,
        "y": 1380,
        "wires": [
          [
            "da3663f8f72f9051"
          ]
        ]
      },
      {
        "id": "171132c7.ece67d",
        "type": "mqtt-broker",
        "broker": "localhost",
        "port": "1883",
        "clientid": "",
        "usetls": false,
        "compatmode": true,
        "keepalive": "60",
        "cleansession": true,
        "birthTopic": ""
      }
```

```
        "birthQos": "0",
        "birthPayload": "",
        "willTopic": "",
        "willQos": "0",
        "willPayload": ""

    },
    {
        "id": "624cdb54f883b70c",
        "type": "ui_group",
        "name": "Sensores",
        "tab": "c64becea02cea741",
        "order": 1,
        "disp": true,
        "width": "7",
        "collapse": false,
        "className": ""

    },
    {
        "id": "0816e6b02af5af5",
        "type": "ui_group",
        "name": "Accionadores",
        "tab": "c64becea02cea741",
        "order": 2,
        "disp": true,
        "width": "6",
        "collapse": false,
        "className": ""

    },
    {
        "id": "0b7f7e36c9b309cf",
        "type": "ui_group",
        "name": "Seguridad",
        "tab": "c64becea02cea741",
        "order": 3,
        "disp": true,
        "width": "6",
        "collapse": false,
        "className": ""

    },
    {
        "id": "990c808eb3798486",
        "type": "node-red-contrib-whatsapp-cmb-account",
        "name": "WA Cliente"

    },
    {
        "id": "316f83b78a6c2eb8",
        "type": "ui_group",
        "name": "Accionadores",
```

```
        "tab": "ae18cf62a80c4eb9",
        "order": 2,
        "disp": true,
        "width": "6",
        "collapse": false,
        "className": ""
    },
{
    "id": "c64becea02cea741",
    "type": "ui_tab",
    "name": "Monitoreo",
    "icon": "dashboard",
    "disabled": false,
    "hidden": false
},
{
    "id": "ae18cf62a80c4eb9",
    "type": "ui_tab",
    "name": "Monitoreo",
    "icon": "dashboard",
    "disabled": false,
    "hidden": false
}
]
```

Criterio de evaluación

Los cambios en el repositorio de github pasada la hora de entrega, descontarán 5 puntos.

Los grupos que envíen su informe final posterior a la hora de entrega, descontarán 5 puntos.

Las notas máximas serán:

20 trabajo satisfactorio, entregado en el plazo, y que utilice node-red

13 trabajo satisfactorio, entregado en el plazo, que no utilice node-red (puede utilizar alguna otra plataforma vista en clase).

Presentación en clase

Cada grupo tendrá un máximo de 15 minutos para presentar su proyecto, tiempo en el cual se debe explicar lo más resaltante de su trabajo y, debe incluir un video sobre el funcionamiento de la aplicación.

La presentación estará a cargo de un estudiante del grupo, elegido al azar.