



SISTEMA INTEGRAL DE AUTOMATIZACIÓN Y SEGURIDAD PARA HOGARES IOT UTILIZANDO COMUNICACIÓN MQTT Y CLOUD COMPUTING

Asignatura: Computación Física

Docente: Jessica Rosas Cuevas



Integrantes – G1

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

TABLA DE CONTENIDOS:

CRITERIOS

01

UTILIDAD

03

ESPECIFICACIÓN

02

COMPLEJIDAD

04

RESULTADOS



01

UTILIDAD

ODS aplicados al proyecto

ODS7: Energía asequible y no contaminante

ODS9: Industria, innovación e infraestructura

ODS11: Ciudades y comunidades sostenibles

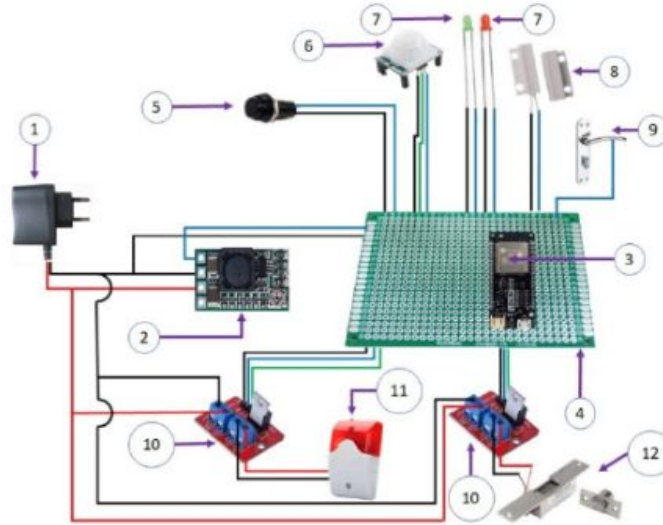


Estado del arte

Objetivo

Aumentar la seguridad en una casa al monitorear y controlar el estado de la puerta para prevenir posibles intrusos.

1. Door Security System for Home Monitoring Based on ESP32



<https://doi.org/10.1016/j.procs.2019.08.218>

Autores

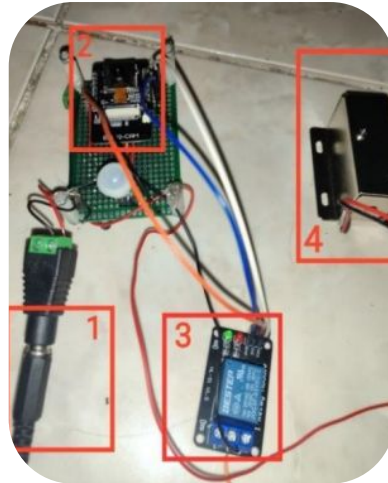
Andreas, Cornelio Revelivan
Aldawira, Handhika
Wiratama Putra, Novita
Hanafiah, Surya Surjarwo,
Aswin Wibisurya.

Estado del arte

Objetivo

Crear un dispositivo capaz de aumentar la seguridad de una habitación utilizando: ESP32 cam, sensor PIR y sensor de incendio.

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



<https://doi.org/10.33795/jartel.v12i2.296>

Autores

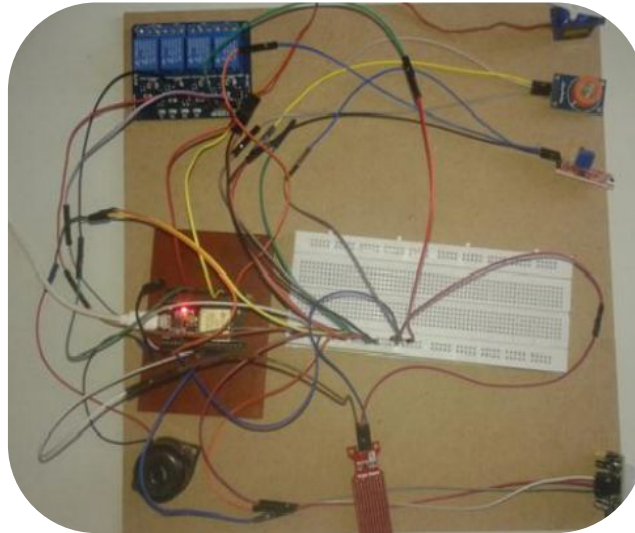
Filantropi Yusuf Aji
Cahyono, Nugroho Suharto,
Lis Diana Mustafa.

Estado del arte

Objetivo

Desarrollar un sistema de bajo costo, flexible y fiable de monitoreo en el hogar con un sistema de control con seguridad adicional utilizando ESP32.

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



https://www.researchgate.net/publication/334226986_Internet_of_Things_Based_Home_Monitoring_and_Device_Control_Using_Esp32

Autores

V. Pravalika, Ch. Rajendra
Prasa

Planteamiento del problema

Los propietarios de viviendas enfrentan desafíos en términos de comodidad, eficiencia y seguridad debido a la falta de control centralizado de los dispositivos. Además, la falta de automatización conduce a un consumo energético innecesario. Por lo tanto, es necesario desarrollar un sistema IoT que aborde estos problemas y mejore la comodidad, eficiencia y seguridad en los hogares.



OBJETIVO DEL PROYECTO

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.

OBJETIVO DEL PROYECTO

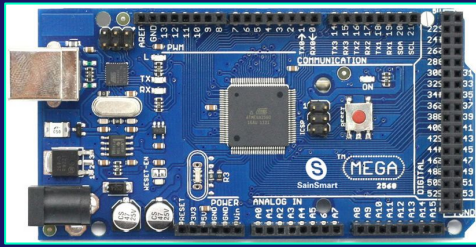
Objetivos específicos:

- 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.



02

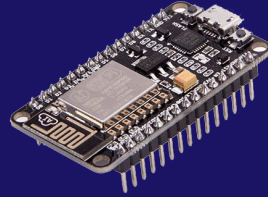
COMPLEJIDAD



TECNOLOGÍAS USADAS



HARDWARE



1. NODEMCU ESP8266
2. Arduino Mega
3. PIR
4. Sensor magnetico
5. Modulo Relay
6. LDR
7. Sensor Ultrasonico
8. Motor DC
9. Sensor FC-28
10. DHT22
11. Laptop
12. Computador

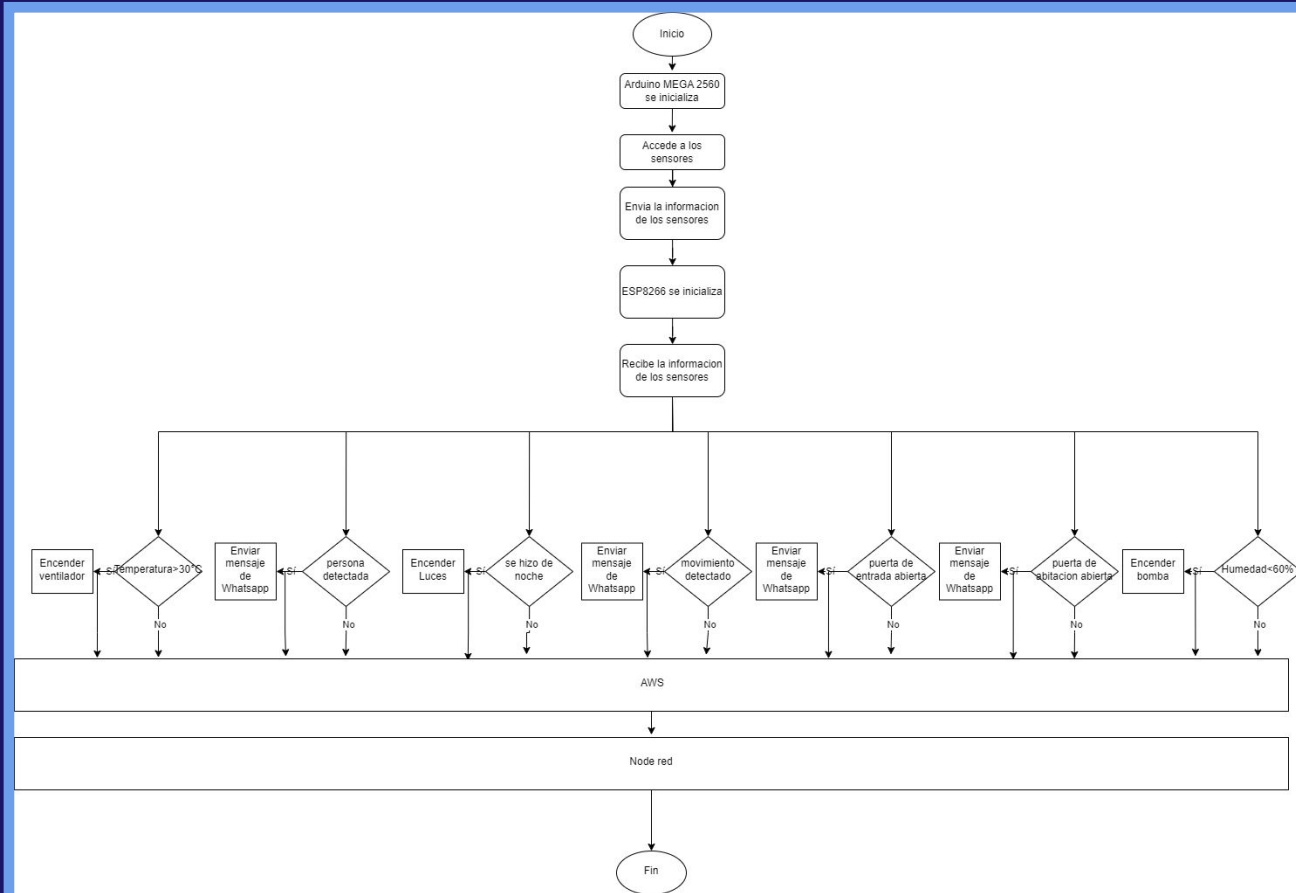
SOFTWARE

1. IDE Arduino
2. Mosquitto
3. Fritzing
4. SO Ubuntu consola
5. AWS
6. Whatsapp
7. Node Red



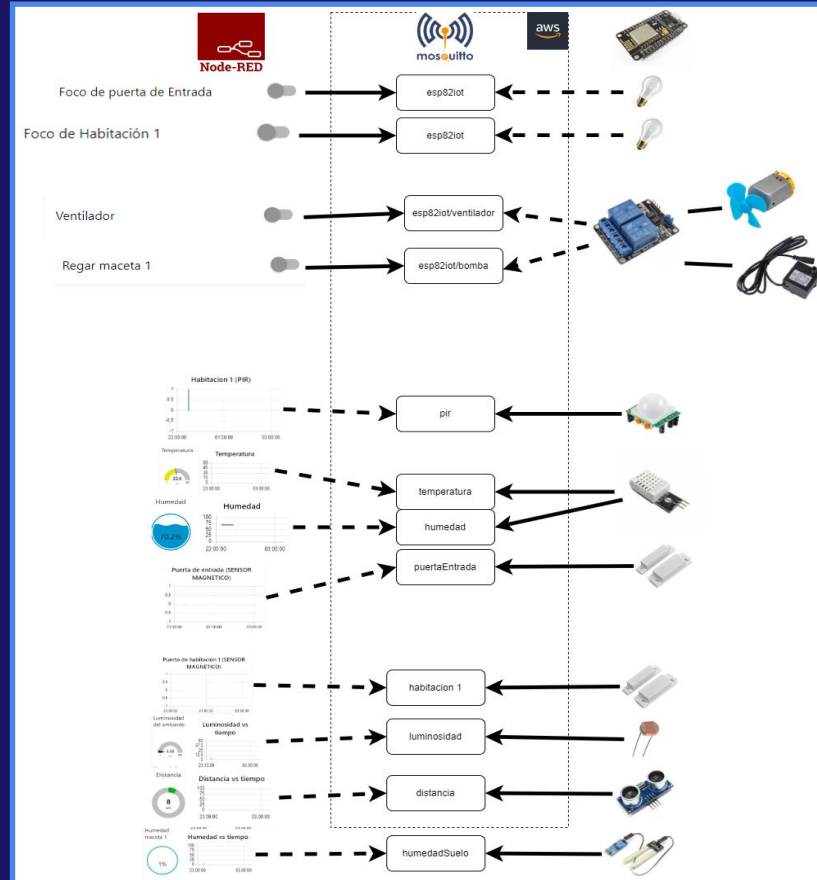
SOLUCIÓN PROPUESTA

DIAGRAMA DE FLUJO

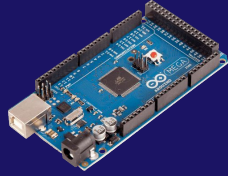


SOLUCIÓN PROPUESTA

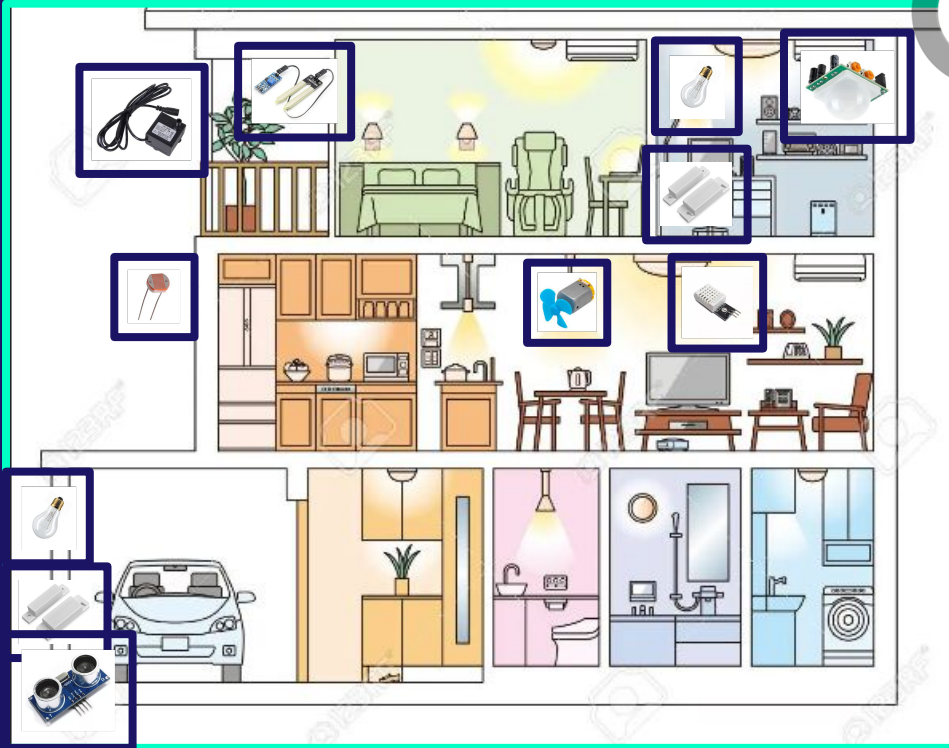
COMUNICACIÓN MQTT



SOLUCIÓN PROPUESTA



Node-RED





03

ESPECIFICACIÓN

REQUERIMIENTOS DEL SISTEMA



Monitoreo y control remoto

El sistema debe permitir el monitoreo y control remoto de diversos dispositivos y sistemas en el hogar





Seguridad del hogar

El sistema debe incluir funcionalidades de seguridad para proteger el hogar y sus ocupantes como la capacidad de enviar notificaciones de alerta



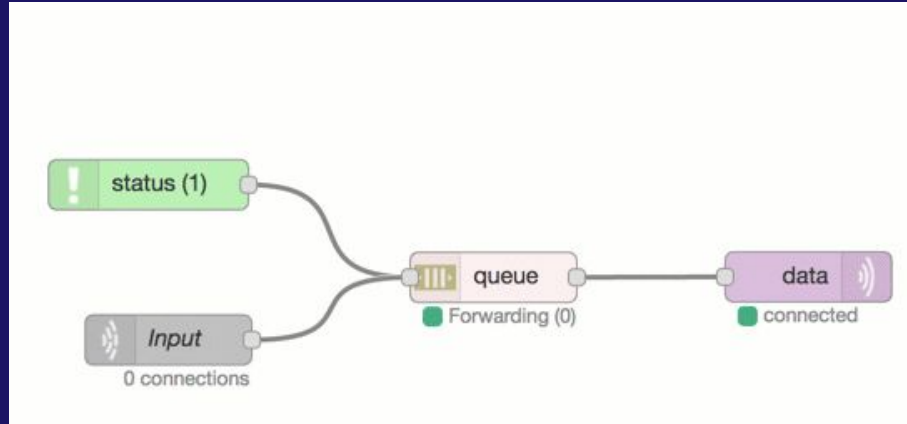
Eficiencia energética

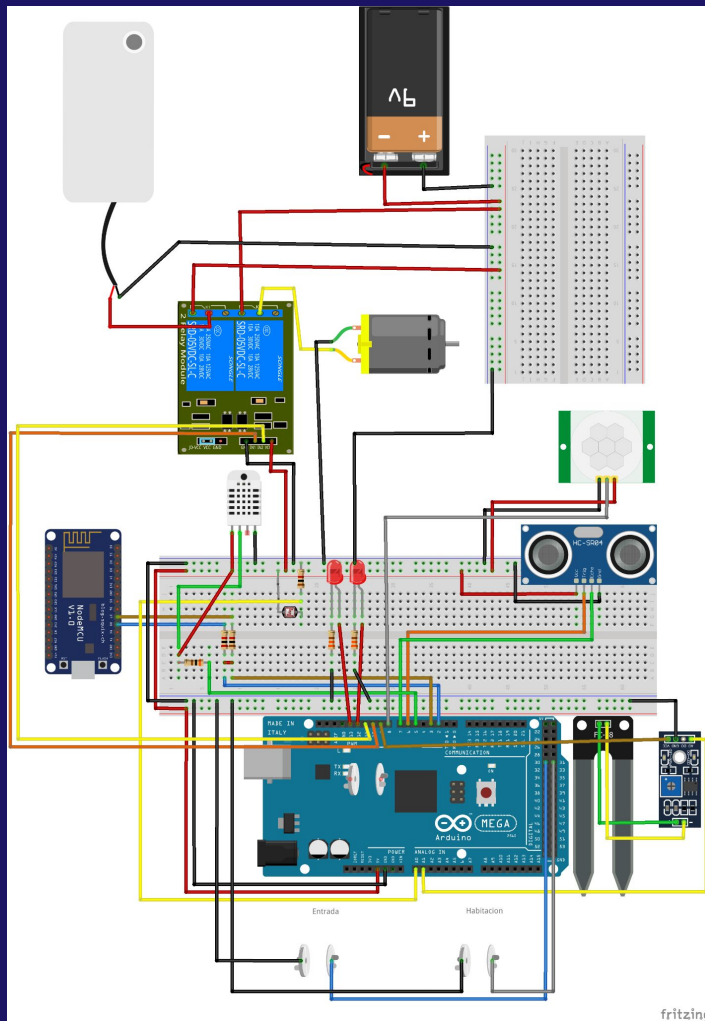
El sistema debe tener la capacidad de monitorear y controlar el consumo de energía en el hogar.



Integración con Node-RED

El sistema debe estar integrado con la plataforma Node-RED para facilitar la configuración y automatización de las diferentes funciones.



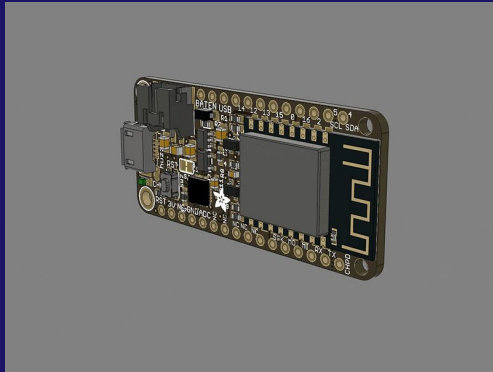


ARQUITECTURA DEL SISTEMA

MICROCONTROLADORES

ESP8266

Módulo de Wi-Fi altamente integrado y de bajo costo utilizado para la conectividad inalámbrica en proyectos de Internet de las cosas (IoT).



Arduino Mega

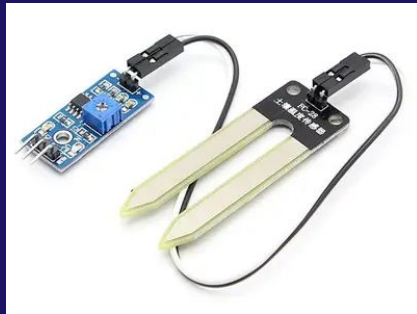
Placa de desarrollo basada en microcontrolador ATmega2560, con mayor cantidad de pines y recursos.



SENSORES

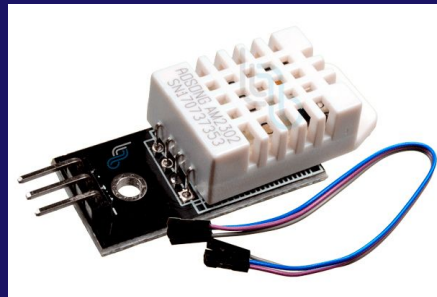
FC28

Módulo de detección de humedad del suelo.



DHT22

Dispositivo que combina un sensor de temperatura y humedad



SENSORES

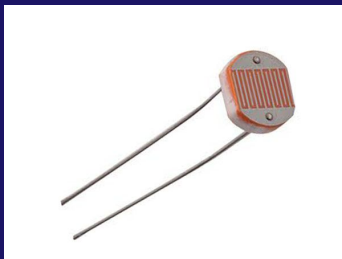
HC-SR04

Módulo ultrasónico utilizado para medir distancias mediante la emisión y recepción de ondas sonoras.



LDR

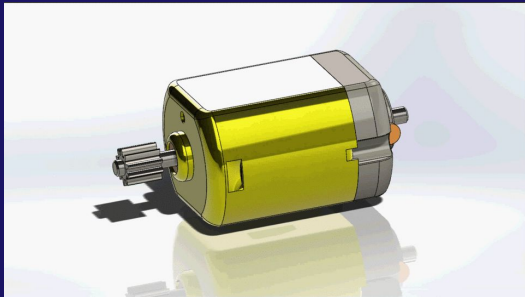
Componente que varía su resistencia eléctrica en función de la intensidad de la luz incidente.



ACTUADORES

MOTOR DC

Tipo de motor eléctrico que convierte la energía eléctrica en movimiento rotatorio mediante el uso de corriente continua

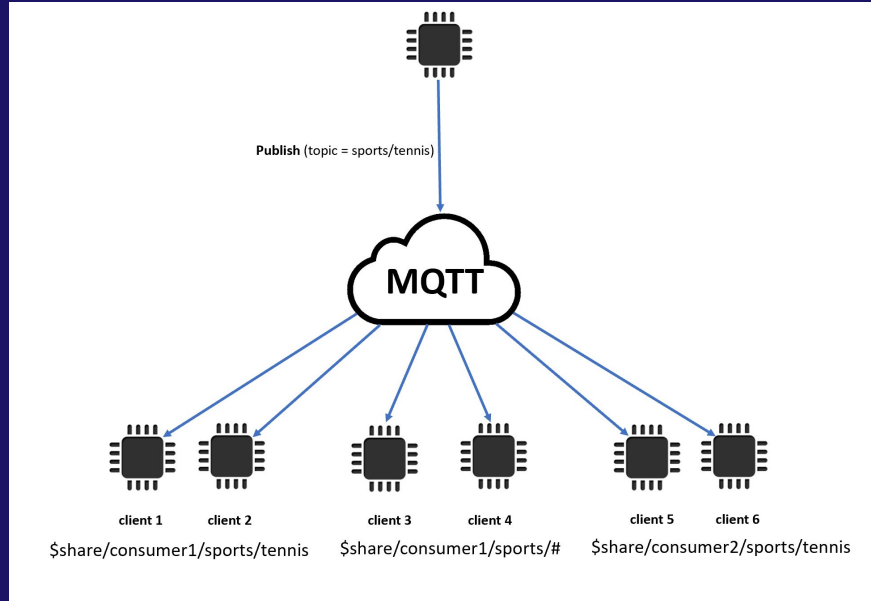


BOMBA AGUA DC-1020

Bomba accionada por un motor DC que se utiliza para transportar y circular agua en sistemas de riego



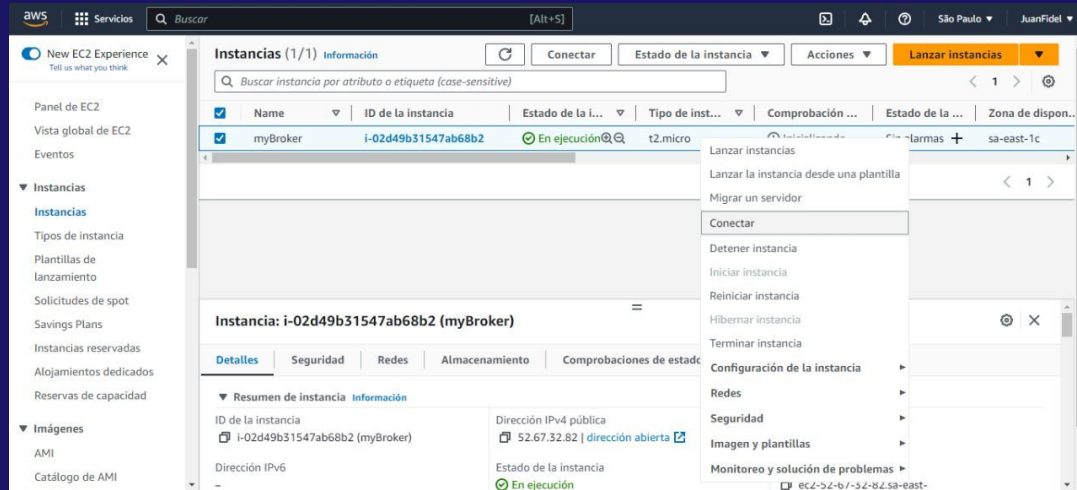
COMUNICACIÓN DEL SISTEMA



MQTT

Utiliza un modelo de publicación / suscripción en el que los dispositivos pueden publicar mensajes en temas específicos y suscribirse a temas para recibir mensajes relevantes

PLATAFORMAS DEL SISTEMA



AWS EC2

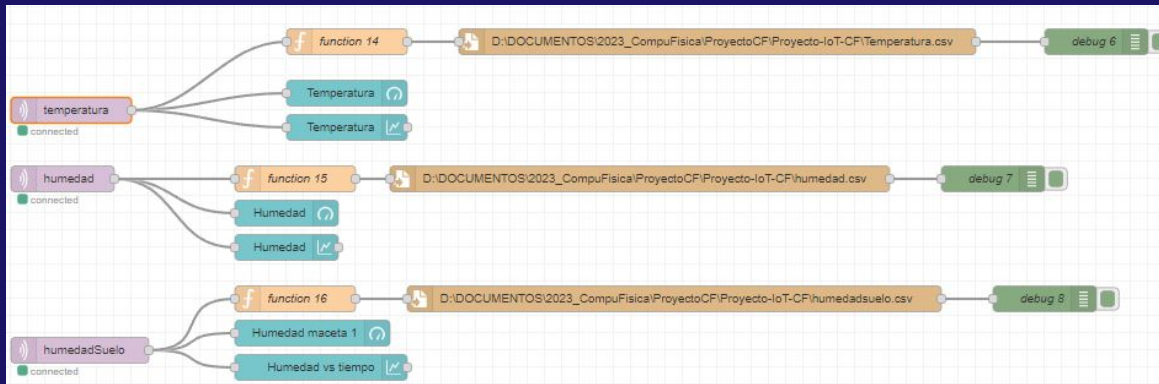
EC2 permite a los usuarios lanzar y administrar máquinas virtuales (instancias) en la nube, lo que les permite ejecutar aplicaciones y cargas de trabajo de manera flexible y eficiente.

PLATAFORMAS DEL SISTEMA



Node-Red

Entorno de programación visual basado en nodos para conectar dispositivos, servicios y API de diferentes plataformas y crear flujos de trabajo de forma visual.



PLATAFORMAS DEL SISTEMA

Monitoreo

Sensores

Temperatura



Temperatura



Humedad



Humedad



Luminosidad
del ambiente



Luminosidad vs
tiempo



Humedad
maceta 1

Humedad vs tiempo

Accionadores

Manual



Ventilador



Foco de puerta de Entrada



Foco de Habitación 1



Regar maceta 1



Seguridad

Habitacion 1 (PIR)



Puerta de entrada (SENSOR
MAGNETICO)



Puerta de habitación 1 (SENSOR
MAGNETICO)



Monitoreo

69.6%

Luminosidad
del ambiente

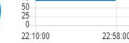
4.69
lux

Humedad
maceta 1

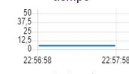
0%

Distancia

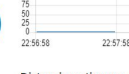
9
cm



Luminosidad vs
tiempo



Humedad vs tiempo



Distancia vs tiempo

Regar maceta 1

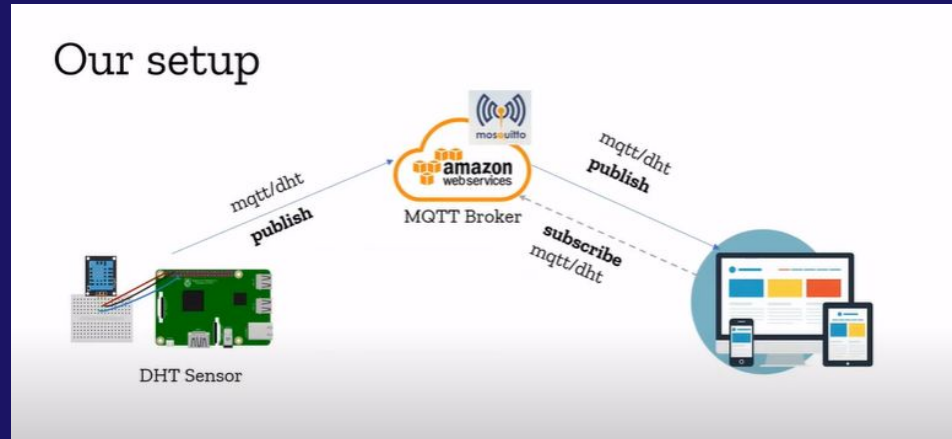
Puerta de entrada (SENSOR
MAGNETICO)



Puerta de habitación 1 (SENSOR
MAGNETICO)



PLATAFORMAS DEL SISTEMA



Mosquitto

Broker de mensajes MQTT de código abierto, que proporciona una plataforma de mensajería ligera y de bajo consumo de energía para la comunicación entre dispositivos en redes IoT.



04

RESULTADOS

PRUEBAS

Fecha	Descripción	Fallo	Estado
23/06/2023	Prueba de sensor PIR	Falla energética	En revis... ▾
23/06/2023	Prueba de Modulo Relay	-	Aprobada ▾
23/06/2023	Prueba de sensor HC-SR04	Medida inadecuada	En revis... ▾
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 revis... ▾
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 ▾



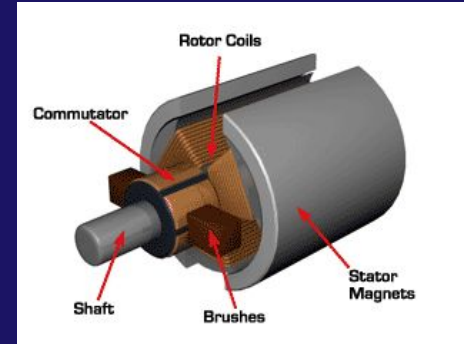
RECOLECCIÓN DE DATOS

	A	B	C
1	Luminosidad	Fecha	Hora
2	4.3	09/07/2023	19:04:39
3	4.4	09/07/2023	19:04:41
4	4.4	09/07/2023	19:04:44
5	4.3	09/07/2023	19:04:46
6	4.3	09/07/2023	19:04:48
7	4.3	09/07/2023	19:04:50
8	4.3	09/07/2023	19:04:52
9	4.4	09/07/2023	19:04:54
10	4.3	09/07/2023	19:04:56

- distancia.csv
- habitacion.csv
- humedad.csv
- humedadsuelo.csv
- luminosidad.csv
- pir.csv
- puertaentrada.csv
- Temperatura.csv

FUNCIONAMIENTO: VENTILADOR

```
1 // VENTILADOR
2 if (String(topic) == "esp82iot/ventilador" && manual){
3     if (messageData == "On"){
4         manual = true;
5         Serial.println("ventilador On");
6         digitalWrite(ventiladorPin, LOW);
7     }
8     else if (messageData == "Off"){
9         Serial.println("ventilador off");
10        digitalWrite(ventiladorPin, HIGH);
11    }
12 }
13
14 void encendidoVentiladorAuto(float temperatura){
15     if (temperatura > 30.0){ //&& humedad<40
16         digitalWrite(ventiladorPin, HIGH);
17     }
18     else{
19         digitalWrite(ventiladorPin, LOW);
20     }
21 }
```



FUNCIONAMIENTO: BOMBA DE AGUA

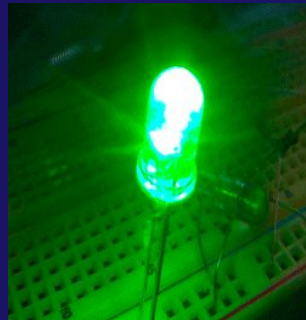


```
1  //BOMBA DE AGUA
2  if (String(topic) == "esp82iot/bomba" && manual) {
3      if (messageData == "On") {
4          Serial.println("bomba On");
5          digitalWrite(bombaPin, LOW);
6      } else if (messageData == "Off") {
7          Serial.println("bomba off");
8          digitalWrite(bombaPin, HIGH);
9      }
10 }
11
12 void encendidoRiegoAuto(float humedadSuel) {
13     if (humedadSuel <= 50.0) {
14         digitalWrite(bombaPin, HIGH);
15         delay(1000);
16         digitalWrite(bombaPin, LOW);
17     }
18 }
```



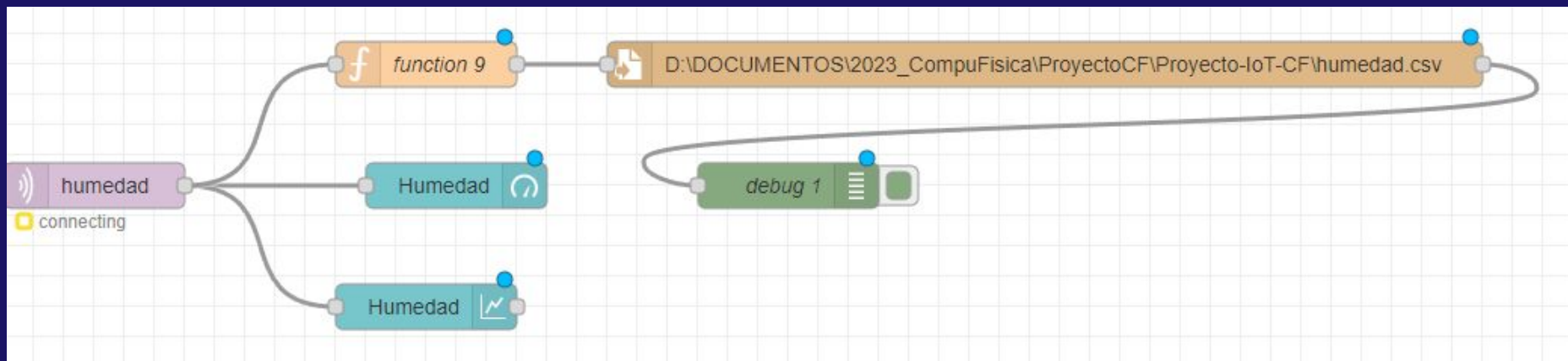
FUNCIONAMIENTO: LUCES

```
1 //FOCOS DE LA PUERTA DE ENTRADA Y HABITACIÓN 1
2 if (String(topic) == "esp82iot" && manual) {
3   if (messageData == "redOn") {
4     Serial.println("red On");
5     digitalWrite(redPin, HIGH);
6     //Foco de entrada
7   } else if (messageData == "redOff") {
8     Serial.println("red Off");
9     digitalWrite(redPin, LOW);
10  }
11  if (messageData == "greenOn") {
12    Serial.println("green On");
13    digitalWrite(greenPin, HIGH);
14  } else if (messageData == "greenOff") {
15    Serial.println("greenOff");
16    digitalWrite(greenPin, LOW);
17  }
18 }
```

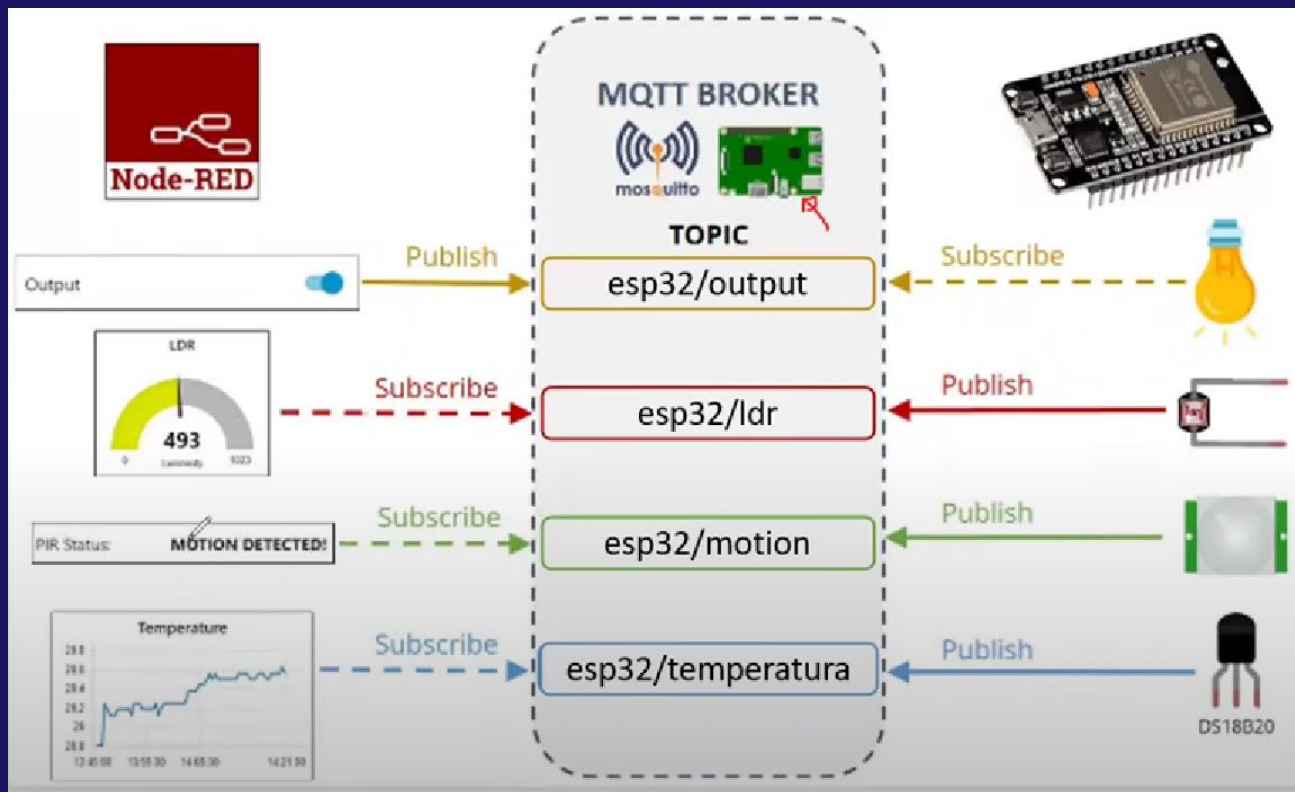


```
1 void encendido_luzAuto(float lumi) {
2   if (lumi <= 20.0) {
3     //Puerta de entrada
4     digitalWrite(redPin, HIGH);
5   } else {
6     digitalWrite(redPin, LOW);
7   }
8 }
```

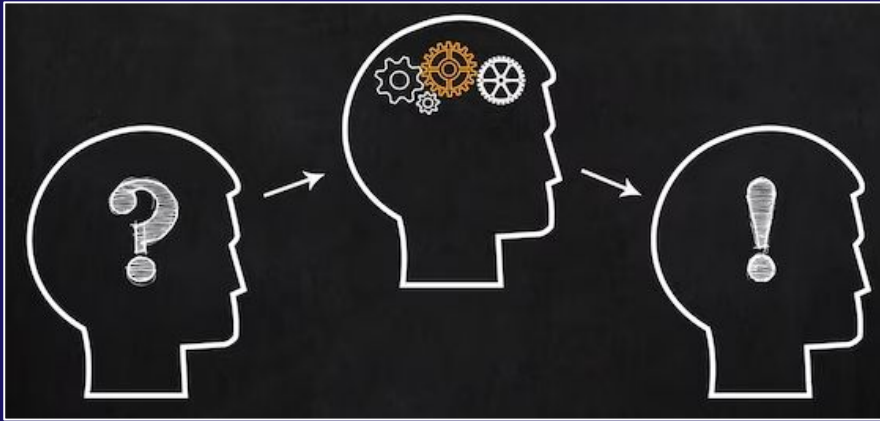
FUNCIONAMIENTO: ALMACENAMIENTO



FUNCIONAMIENTO: RED



LECCIONES APRENDIDAS



- Programación de sensores y actuadores IoT.
- Configuración de flujos de trabajo en Node Red.
- Publicación y suscripción de dispositivos a broker Mosquito.
- Almacenamiento de datos usando archivos csv.

GRACIAS