Centro de Estudios de Postgrado

Máster en Ingeniería Geomática y Geoinformación

UNIVERSIDAD DE JAÉN
Centro de Estudios de Postgrado

Trabajo Fin de Máster

GEOPORTAL PARA LA MONITORIZACIÓN DE ESPACIOS USANDO REDES DE SENSORES Y ARQUITECTURAS ORIENTADAS A SERVICIOS

Alumno: Conejo Gangkofner, Daniel

Tutor: Prof. D. Manuel A. Ureña Cámara
Prof. D. Pablo Cano Marchal

Dpto: Ingeniería Cartográfica, Geodésica y Fotogrametría.

Ingeniería Electrónica y Automática.

Diciembre, 2019

Título:	**Geoportal para la monitorización de espacios usando redes de sensores y arquitecturas orientadas a servicios.**
Autor:	**Daniel Conejo Gangkofner.**
Tutores:	**Manuel A. Ureña** **Pablo Cano**
Departamento:	**Ingeniería Cartográfica, Geodésica y Fotogrametría.** **Ingeniería Electrónica y Automática.**
Fecha:	**16 de diciembre de 2019**

Resumen

Este Trabajo Fin de Master tiene como objetivo captar, almacenar y analizar los datos provenientes de una red de sensores.

Para ello se ha diseñado e instalado una red de sensores que tiene como objetivo la medición y obtención de un conjunto de parámetros ambientales. Usando arquitectura basada en servicios, estos datos serán recopilados en un servidor que, a su vez hará de servicio de mapas (Cliente web local).
Se hará un análisis de la información extraída de los datos recopilados.

Finalmente, se desarrollará un cliente que permita la visualización de los datos captados en tiempo real de forma local. Este cliente tendrá adicionalmente diferentes servicios implementados en el servidor.

Palabras clave: Arduino, Node-Red, MQTT, MySQL, Sensores, Cliente Web, Monitorización de Interiores

Título:	**Geoportal para la monitorización de espacios usando redes de sensores y arquitecturas orientadas a servicios.**
Autor:	**Daniel Conejo Gangkofner.**
Tutores:	**Manuel A. Ureña** **Pablo Cano**
Departamento:	**Ingeniería Cartográfica, Geodésica y Fotogrametría.** **Ingeniería Electrónica y Automática.**
Fecha:	**16 December 2019**

Abstract

This Master's Final Project aims to capture, store and analyze data from a sensor network.

To this end, a sensor network that aims to measure and obtain a set of environmental parameters has been designed and installed. Through the use of service-based architecture, this data will be collected on a server that, in turn, will serve as a map service(LocalWebClient).
An analysis of the information extracted from the collected data will be made.

Finally, a client that allows the visualization of the data collected in real time locally will be developed. This client will additionally have different services implemented on the server.

Keywords: Arduino, Node-Red, MQTT, MySQL, Sensors, Web Client, Interior Monitoring

ÍNDICE

Capítulo 1 INTRODUCCIÓN

1.1 MOTIVACIÓN

La idea que hay detrás de este Trabajo Fin de Master surge de los conocimientos adquiridos en el master de Ingeniería Geomática y Geoinformación y busca monitorizar elementos ambientales usando una arquitectura basada en servicios.

Cada vez más, las principales ciudades de todo el mundo luchan por transformarse en espacios más innovadores a través de las Tecnologías de la Información y las Comunicaciones (TIC).
Su utilización no solo supone mejoras en la provisión de los servicios, sino que allana el camino para convertir nuestras ciudades en verdaderas “Smart Cities”.

Con la creciente importancia de las “Smart Cities” y su transformación en espacios innovadores, la sensorización y monitorización de elementos se ha convertido en uno de los elementos claves e inevitables de las ciudades inteligentes.

1.2 OBJETIVOS Y JUSTIFICACIÓN

El objetivo principal de este trabajo es el desarrollar un sistema que sea capaz de capturar, gestionar, analizar y visualizar parámetros ambientales tal y como se podría encontrar y usar en una “Smart City” que quiera aumentar la eficiencia y la rentabilidad de sus servicios.

- **Diseño de una red de sensores para la monitorización de interiores.**

Se estudiará la distribución ideal de cada uno de los sensores en la zona de estudio para perfeccionar la captura de datos y aumentar así la rentabilidad y eficiencia del servicio.

- **Almacenamiento y análisis de la información.**

La información recibida desde el sistema de captura se almacenará en una base de datos y al mismo tiempo se enviará al cliente Web.

- **Visualización de la información.**

Finalmente el cliente web tendrá diferentes elementos que me permitirán una visualización y monitorización de los datos captados en tiempo real.

Capítulo 2 ANTECEDENTES

2.1 ARDUINO

Arduino es una plataforma de computación física de software y hardware.
Ambos componentes son de código abierto. El hardware consta de una placa de I / O (Input/Output)simple con un micro controlador y entradas y salidas analógicas y digitales.

El entorno de desarrollo se basa en el Processing y debería facilitar a los usuarios técnicamente menos experimentados el acceso a la programación y los micro controladores. La programación en sí se realiza en un lenguaje de programación similar a C o C ++, con detalles técnicos como archivos de encabezado en gran parte ocultos para los usuarios y bibliotecas extensas y ejemplos que simplifican la programación.

Arduino se puede usar para controlar objetos interactivos independientes o para interactuar con aplicaciones de software en computadoras (como Adobe Flash, Processing, Max / MSP, Pure Data, SuperCollider, varios lenguajes de scripting, Terminal, vvvv, etc.).

Figura 1: Logo Arduino

HARDWARE:

El hardware de una placa Arduino típica se basa en un microcontrolador Atmel AVR de la serie megaAVR, como el ATmega328.

Todas las placas funcionan con USB (5V) o una fuente de alimentación externa (7-12V) y tienen un oscilador de cristal de 16 MHz. También hay variantes con un voltaje de suministro de 3.3 V y variantes con reloj desviado. Las extensiones también se pueden usar para programar otros microcontroladores, como el ESP8266, a través del IDE de Arduino.

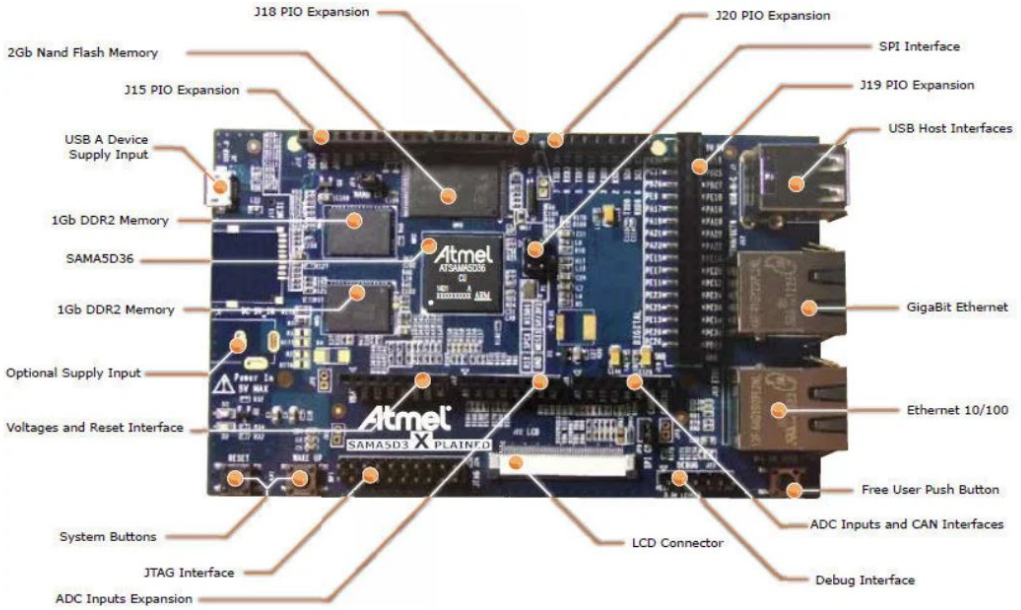


Figura 2: Microcontrolador

SOFTWARE:

Arduino viene con su propio entorno de desarrollo integrado (IDE) basado en Wiring IDE. Es una aplicación Java que está disponible para las plataformas comunes Windows, Linux y macOS de forma gratuita.

Para un programa funcional, es suficiente definir dos funciones:

setup (): se llama una vez al iniciar el programa (ya sea después de la transferencia a la placa o después de presionar el botón de reinicio), por ejemplo, para definir los pines como entrada o salida.

loop (): pasará una y otra vez siempre que la placa Arduino esté encendida.

Aquí hay un ejemplo de un programa (en la dicción Arduino: Sketch) que parpadea un LED conectado a la placa Arduino:

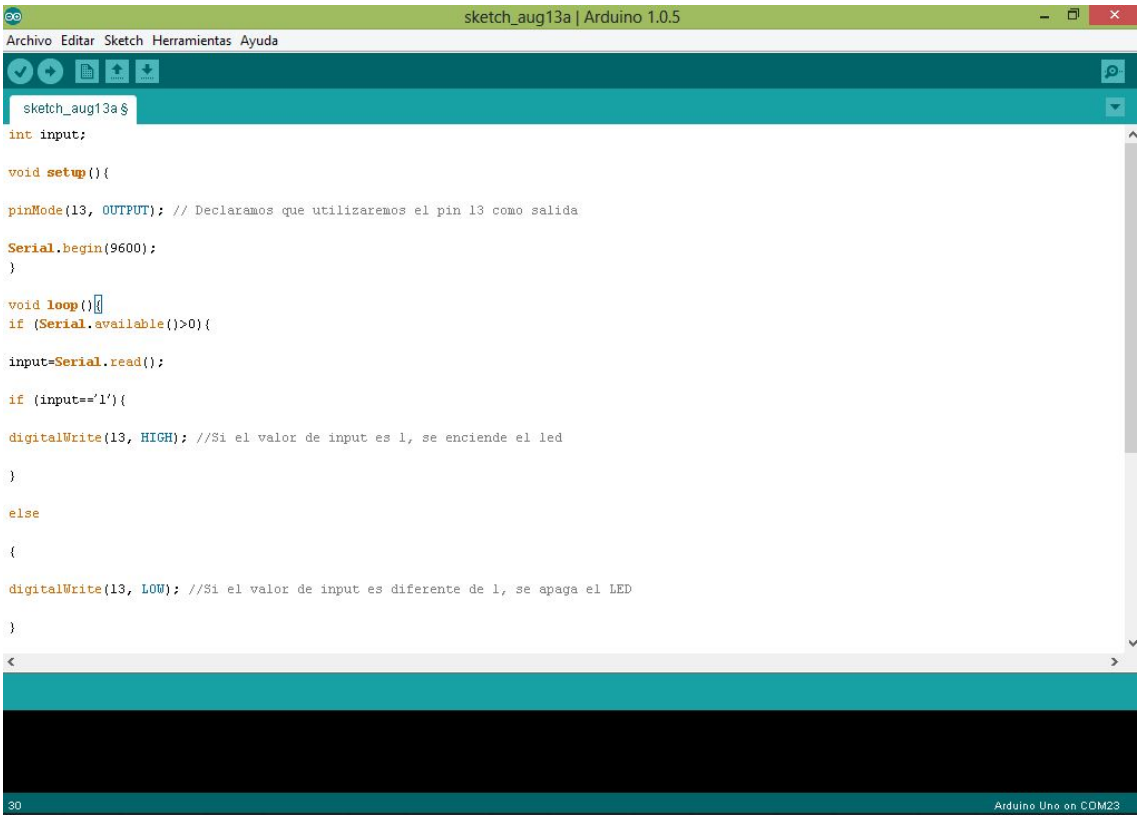


Figura 3: IDE (Integrated Development Enviroment) Arduino

2.2 MQTT

Message Queuing Telemetry Transport (MQTT) es un protocolo de mensajería abierto para la comunicación de máquina a máquina (M2M) que permite la transmisión de datos de telemetría como mensajes entre dispositivos, a pesar de las grandes demoras o las redes limitadas. Los dispositivos correspondientes van desde sensores y actuadores, teléfonos móviles, sistemas integrados en vehículos o computadoras portátiles hasta computadoras completamente desarrolladas.

El protocolo MQTT también se conoce con nombres antiguos como "WebSphere MQTT" (WMQTT), "Protocolo SCADA" o "Protocolo de dispositivo SCADA MQ Integrator" (MQIsdp).

La Autoridad de Números Asignados de Internet (IANA) reserva los puertos 1883 y 8883 para MQTT. Los mensajes MQTT pueden encriptarse usando el protocolo TLS.

Es interesante que un servidor MQTT ("intermediario") contenga toda la situación de datos de sus socios de comunicación y, por lo tanto, pueda usarse como una base de datos de estado. Por lo tanto, es posible conectar dispositivos MQTT pequeños y de bajo rendimiento a un intermediario MQTT, por lo que los dispositivos recopilan datos y / o reciben comandos, mientras que una imagen de situación compleja surge solo en el intermediario MQTT y puede ser evaluada aquí o por un socio de comunicación eficiente. De esta manera, las intervenciones pueden transmitirse desde una o más instancias poderosas al agente MQTT y distribuirse a los dispositivos individuales. Esto hace que MQTT sea muy adecuado para soluciones de automatización y se usa ampliamente en IoT debido a su facilidad de uso.

Figura 4: Logo MQTT

PROTOCOLO:

MQTT es un protocolo cliente-servidor. Los clientes envían mensajes al servidor ("intermediario") después de la conexión con un tema que clasifica jerárquicamente el mensaje; por ejemplo,
cocina / refrigerador / temperatura o
automóvil / bicicleta / 3 / presión de aire.
Los clientes pueden suscribirse a estos temas, con el servidor reenvía los mensajes recibidos a los suscriptores apropiados.

Los mensajes siempre consisten en un tema y el contenido del mensaje. Los mensajes se envían con una calidad de servicio definible: como máximo una vez (el mensaje se envía una vez y puede que no llegue a una desconexión), al menos una vez y exactamente una vez (esto asegura que el mensaje llegue una única vez, incluso si se interrumpe la conexión). Además, el servidor puede recibir instrucciones mediante el indicador de retención para almacenar en caché el mensaje de este tema. Los clientes que se suscriben a este tema primero reciben el mensaje en caché.

Al establecer una conexión, los clientes pueden definir una "última voluntad" en forma de mensaje. Si se pierde la conexión con el cliente, este mensaje se publica y se envía a los suscriptores correspondientes.

MQTT se usa comúnmente sobre TCP y tiene un encabezado de 2 bytes. El primer byte contiene el tipo de mensaje (4 bits), la calidad del servicio (2 bits) y un indicador de retención.

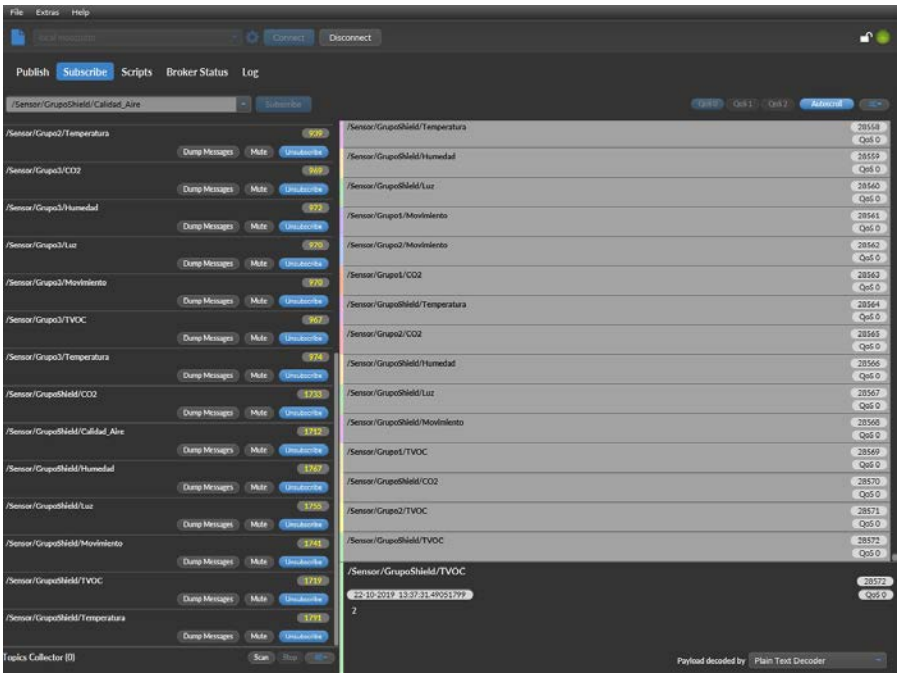


Figura 5: Software MQTT

2.3 NODE-RED

Figura 6: Logo Node Red

Node-RED es una herramienta de desarrollo gráfico desarrollada por IBM. El software permite implementar aplicaciones en el área de Internet de las cosas con un principio modular simple. Los bloques de funciones individuales se conectan arrastrando conexiones. Una gran selección de componentes incluidos cubre la mayoría de los servicios y tecnologías más comunes.

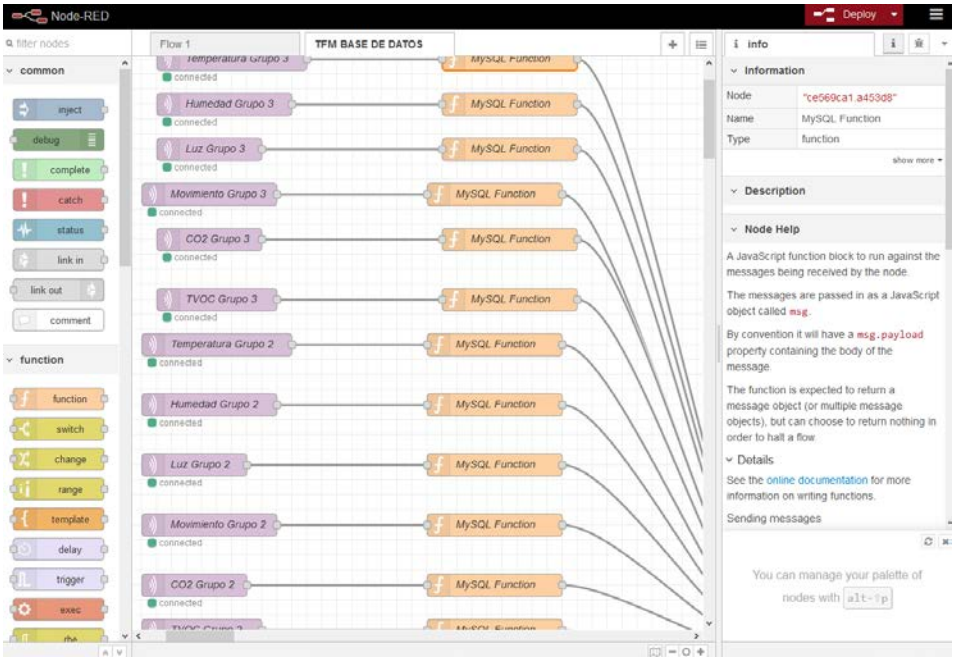
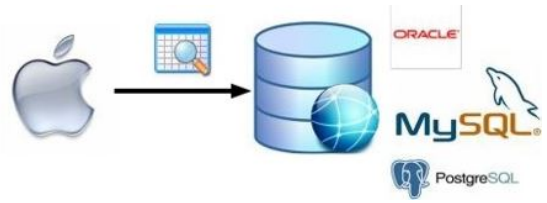


Figura 7: Software NodeRed

Con Node-RED, se pueden conectar diferentes nodos de entrada, salida y procesamiento en un solo flujo. Esto le permite controlar y controlar diferentes cosas. Cada nodo tiene una tarea definida y única. Si los datos se transmiten a un nodo, el nodo puede procesar estos datos y reenviarlos al siguiente nodo.



2.4 BASE DE DATOS

Una base de datos, también llamada sistema de base de datos, es un sistema para la gestión electrónica de los mismos. La tarea esencial de una base de datos es almacenar grandes volúmenes de información de manera eficiente, sin contradicciones y de forma permanente, así como proporcionar los subconjuntos necesarios en diferentes formas de presentación basadas en las necesidades para los usuarios y los programas de aplicación.

Figura 8: Logo MySQL

2.4.1 MySQL

MySQL es uno de los sistemas de gestión de bases de datos relacionales más populares del mundo. Está disponible como software de código abierto, así como una versión comercial para varios sistemas operativos y constituye la base de muchos sitios web dinámicos.

Una de las aplicaciones más extendidas con MySQL es el almacenamiento de datos para servicios en red.

Estructura del sistema de almacenamiento utilizado.

MySQL proporciona un servidor de base de datos que nos permitirá almacenar la información de los sensores.
El puerto predeterminado para el servidor MySQL es 3306 en el Protocolo de control de transmisión (TCP).

Se pueden crear varias bases de datos en el sistema de gestión de bases de datos, el servidor MySQL. Se pueden crear varias tablas en una base de datos. En la práctica, MySQL crea una carpeta en el disco duro para cada base de datos, que almacena archivos para la estructura y los datos de las tablas individuales. El formato exacto de estos archivos depende del motor de almacenamiento utilizado para cada tabla.

Las tablas pueden ser cada una de un tipo diferente. El tipo de tabla determina qué motor de almacenamiento (subsistema de almacenamiento) usar para las solicitudes a una tabla. Cada tabla puede contener columnas donde se pueden almacenar datos de un tipo de datos específico (por ejemplo, enteros o cadenas). El tamaño máximo de las tablas solo está limitado por el sistema operativo.

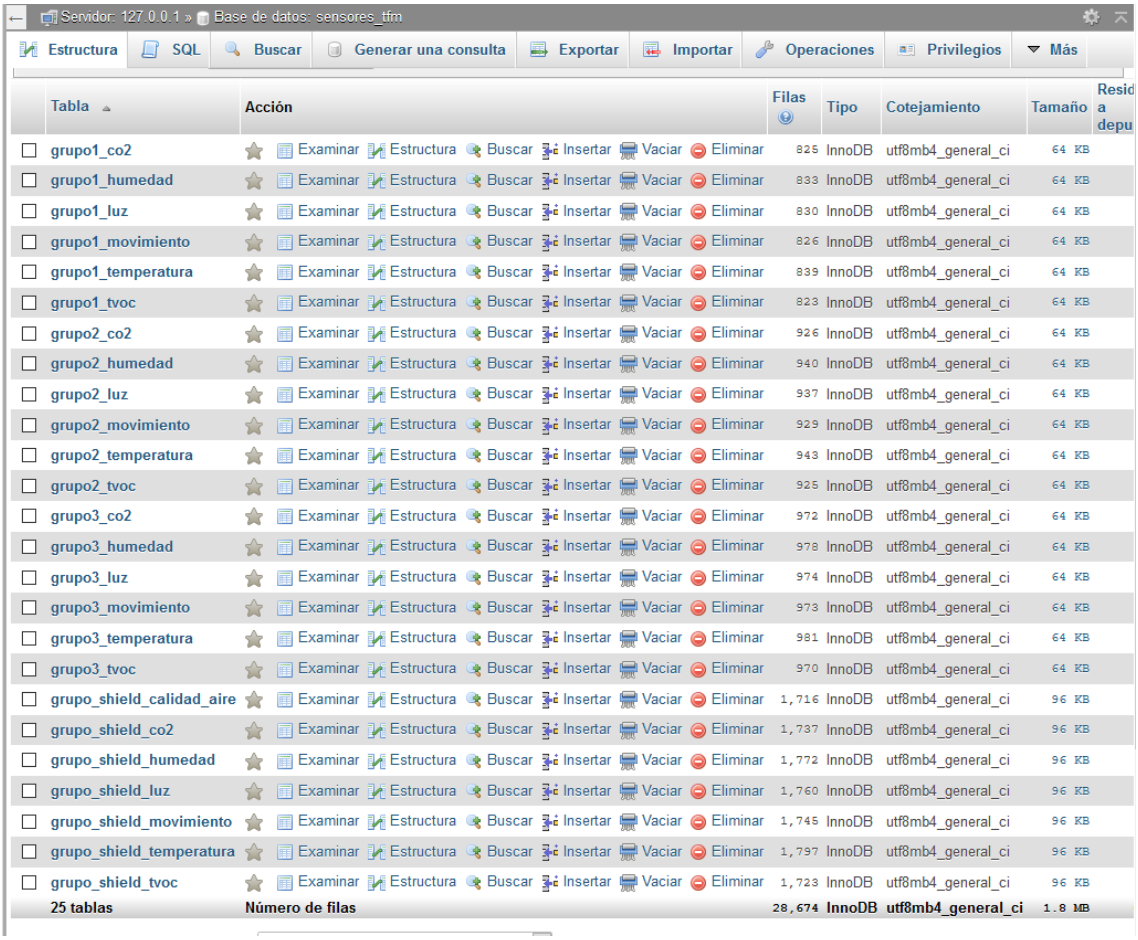


Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Resid a depu
grupo1_co2	Examinar Estructura Buscar Insertar Vaciar Eliminar	825	InnoDB	utf8mb4_general_ci	64 KB	
grupo1_humedad	Examinar Estructura Buscar Insertar Vaciar Eliminar	833	InnoDB	utf8mb4_general_ci	64 KB	
grupo1_luz	Examinar Estructura Buscar Insertar Vaciar Eliminar	830	InnoDB	utf8mb4_general_ci	64 KB	
grupo1_movimiento	Examinar Estructura Buscar Insertar Vaciar Eliminar	826	InnoDB	utf8mb4_general_ci	64 KB	
grupo1_temperatura	Examinar Estructura Buscar Insertar Vaciar Eliminar	839	InnoDB	utf8mb4_general_ci	64 KB	
grupo1_tvoc	Examinar Estructura Buscar Insertar Vaciar Eliminar	823	InnoDB	utf8mb4_general_ci	64 KB	
grupo2_co2	Examinar Estructura Buscar Insertar Vaciar Eliminar	926	InnoDB	utf8mb4_general_ci	64 KB	
grupo2_humedad	Examinar Estructura Buscar Insertar Vaciar Eliminar	940	InnoDB	utf8mb4_general_ci	64 KB	
grupo2_luz	Examinar Estructura Buscar Insertar Vaciar Eliminar	937	InnoDB	utf8mb4_general_ci	64 KB	
grupo2_movimiento	Examinar Estructura Buscar Insertar Vaciar Eliminar	929	InnoDB	utf8mb4_general_ci	64 KB	
grupo2_temperatura	Examinar Estructura Buscar Insertar Vaciar Eliminar	943	InnoDB	utf8mb4_general_ci	64 KB	
grupo2_tvoc	Examinar Estructura Buscar Insertar Vaciar Eliminar	925	InnoDB	utf8mb4_general_ci	64 KB	
grupo3_co2	Examinar Estructura Buscar Insertar Vaciar Eliminar	972	InnoDB	utf8mb4_general_ci	64 KB	
grupo3_humedad	Examinar Estructura Buscar Insertar Vaciar Eliminar	978	InnoDB	utf8mb4_general_ci	64 KB	
grupo3_luz	Examinar Estructura Buscar Insertar Vaciar Eliminar	974	InnoDB	utf8mb4_general_ci	64 KB	
grupo3_movimiento	Examinar Estructura Buscar Insertar Vaciar Eliminar	973	InnoDB	utf8mb4_general_ci	64 KB	
grupo3_temperatura	Examinar Estructura Buscar Insertar Vaciar Eliminar	981	InnoDB	utf8mb4_general_ci	64 KB	
grupo3_tvoc	Examinar Estructura Buscar Insertar Vaciar Eliminar	970	InnoDB	utf8mb4_general_ci	64 KB	
grupo_shield_calidad_aire	Examinar Estructura Buscar Insertar Vaciar Eliminar	1,716	InnoDB	utf8mb4_general_ci	96 KB	
grupo_shield_co2	Examinar Estructura Buscar Insertar Vaciar Eliminar	1,737	InnoDB	utf8mb4_general_ci	96 KB	
grupo_shield_humedad	Examinar Estructura Buscar Insertar Vaciar Eliminar	1,772	InnoDB	utf8mb4_general_ci	96 KB	
grupo_shield_luz	Examinar Estructura Buscar Insertar Vaciar Eliminar	1,760	InnoDB	utf8mb4_general_ci	96 KB	
grupo_shield_movimiento	Examinar Estructura Buscar Insertar Vaciar Eliminar	1,745	InnoDB	utf8mb4_general_ci	96 KB	
grupo_shield_temperatura	Examinar Estructura Buscar Insertar Vaciar Eliminar	1,797	InnoDB	utf8mb4_general_ci	96 KB	
grupo_shield_tvoc	Examinar Estructura Buscar Insertar Vaciar Eliminar	1,723	InnoDB	utf8mb4_general_ci	96 KB	
25 tablas	Número de filas	28,674	InnoDB	utf8mb4_general_ci	1.8 MB	

Figura 9: Todas las tablas en la Base de Datos que se han creado para este TFM

Un cliente puede enviar solicitudes de bases de datos a un servidor MySQL. Este es responsable de manejar cada solicitud de la manera más eficiente posible. Primero, se consulta el caché de consultas y, si el resultado no existe, la consulta se analiza, optimiza y finalmente se ejecuta.

2.5 CLIENTE WEB

En el entorno cliente-servidor de la World Wide Web, los clientes web controlan los procesos. Son los usuarios del servicio los que envían solicitudes a los servidores web y posteriormente reciben una respuesta.

Por lo tanto, un cliente web envía una solicitud a un servidor web utilizando el HTTP, con lo cual este último le devuelve un documento HTML. HTML es el lenguaje de descripción de página en la WWW que permite la realización de hipervínculos. Esto facilita la navegación entre servidores de todo el mundo. El cliente web interpreta las instrucciones HTML del documento recibido y presenta la página web al usuario.

Para proporcionar páginas web dinámicas en el cliente, no es necesario llamar primero a un programa en el servidor, como es el caso de los programas CGI. Por ejemplo, el servidor puede devolver un applet de Java al que se hace referencia en un documento HTML en respuesta a una solicitud. Un applet de Java es un programa codificado de Java que se ejecuta en el cliente. Sin embargo, esto debe tener un intérprete de Java, que es el caso en todos los navegadores más nuevos. Netscape proporcionó otra posibilidad a través del desarrollo del lenguaje de programación JavaScript. Las declaraciones de JavaScript se incrustan en el código HTML de un documento y, por lo tanto, se transfieren al cliente junto con el documento. El navegador Netscape interpreta las declaraciones de JavaScript y presenta el contenido generado. Ambos mecanismos proporcionan una funcionalidad compleja para hacer que las páginas HTML sean dinámicas.

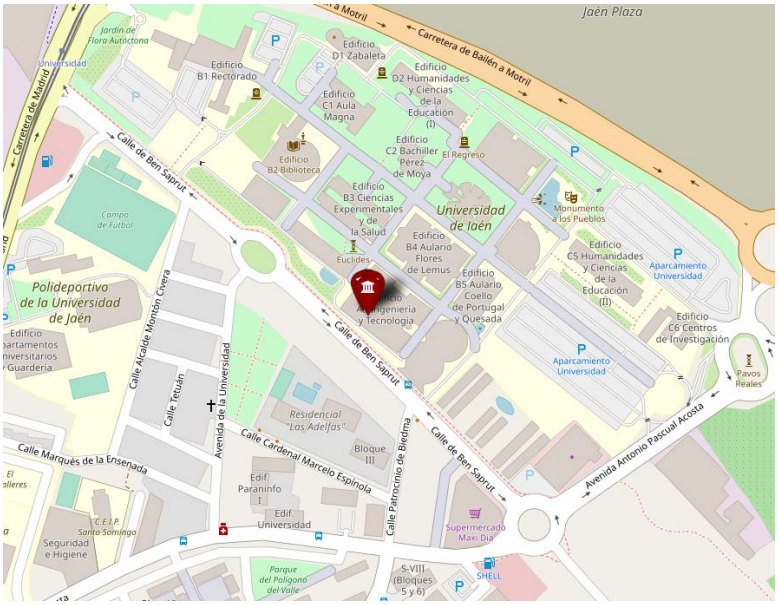


Figura 10: Cliente Web

Capítulo 3 ZONA DE ESTUDIO

3.1 AULA 352

Debido a que el objetivo del presente trabajo es la monitorización de interiores con diferentes sensores, se ha elegido un aula de la universidad de Jaén para realizar el montaje .
Concretamente se trata del aula 352 en la tercera planta de la escuela de Ingeniería y Arquitectura. Se ha considerado este aula un entorno adecuado para montar la red de sensores, realizar pruebas y tomar datos representativos.
A continuación se muestra la información del catastro de dicho edificio.

Figura 11: Información Catastro

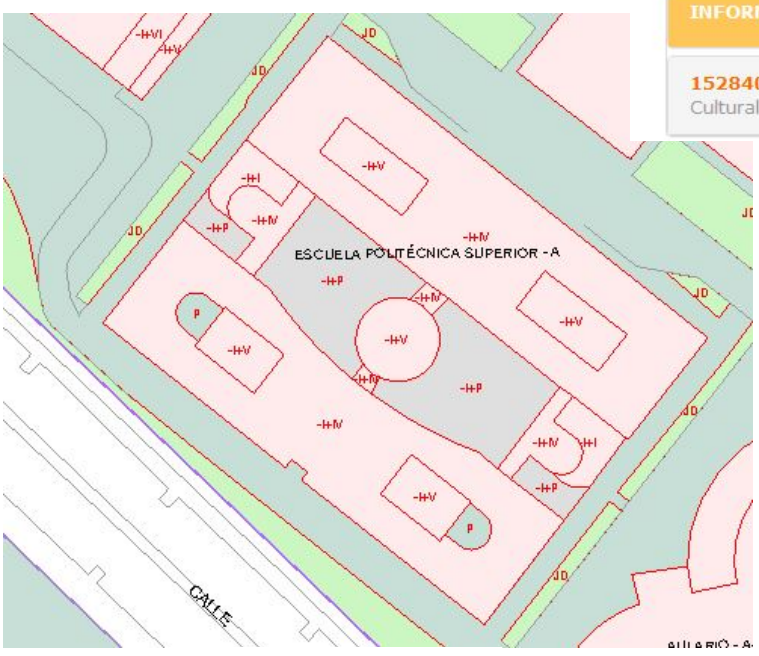


Figura 12: Plano Catastro

Capítulo 4 MATERIAL Y METODO

Este apartado tiene como objetivo resumir el conjunto de materiales necesarios que se han usado para el correcto desarrollo del trabajo.

4.1 SENSORES

De cada uno de los diferentes sensores existen 4 unidades excepto del sensor de calidad de aire, del cual solo se ha usado uno.

4.1.1 SENSOR DE MOVIMIENTO – SENSOR PIR SR505

Los sensores infrarrojos pasivos (PIR) son dispositivos para la detección de movimiento. Son baratos, pequeños, de baja potencia, y fáciles de usar. Por esta razón son frecuentemente usados en juguetes, aplicaciones domóticas o sistemas de seguridad.

Los sensores PIR se basan en la medición de la radiación infrarroja. Todos los cuerpos (vivos o no) emiten una cierta cantidad de energía infrarroja, mayor cuanto mayor es su temperatura. Los dispositivos PIR disponen de un sensor eléctrico capaz de captar esta radiación y convertirla en una señal eléctrica.

En realidad cada sensor está dividido en dos campos y dispone de un circuito eléctrico que compensa ambas mediciones. Si ambos campos reciben la misma cantidad de infrarrojos la señal eléctrica resultante es nula. Por el contrario, si los dos campos realizan una medición diferente, se genera una señal eléctrica.

De esta forma, si un objeto atraviesa uno de los campos se genera una señal eléctrica diferencial, que es captada por el sensor, y se emite una señal digital.

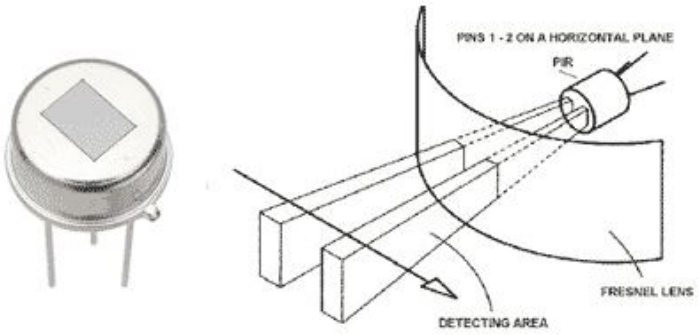


Figura 13: Campo visión Sensor PIR

El otro elemento restante para que todo funcione es la óptica del sensor. Básicamente es una cúpula de plástico formada por lentes de fresnel, que divide el espacio en zonas, y enfoca la radiación infrarroja a cada uno de los campos del PIR.

De esta manera, cada uno de los sensores capta un promedio de la radiación infrarroja del entorno. Cuando un objeto entra en el rango del sensor, alguna de las zonas marcadas por la óptica recibirá una cantidad distinta de radiación, que será captado por uno de los campos del sensor PIR, disparando la alarma.

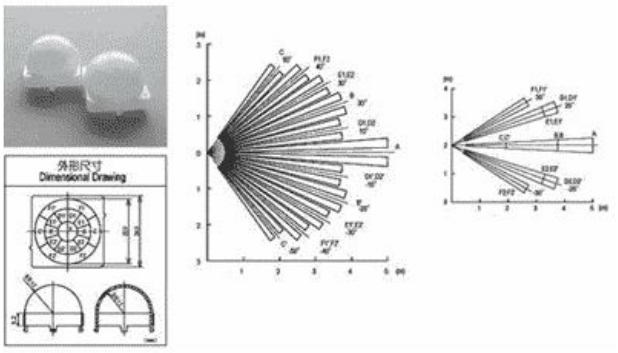


Figura 14: Rango del sensor PIR

Descripción:

- Tensión de funcionamiento: DC4.5-20V
- Corriente en reposo: <60uA
- Nivel de salida: Gao 3.3V / Low 0V
- Disparador: disparador de reutilizable (por defecto)
- Tiempo de retardo: Los 8S por defecto + -30%
- Dimensiones de la placa: 10 * 23mm
- ángulo de la inducción: <100 grados de ángulo de cono
- Distancia de detección: 3 metros
- Temperatura de trabajo: -20 a +80 grados
- Sensor de la lente Dimensiones: Diámetro: 10 mm

Figura 15: Sensor Movimiento

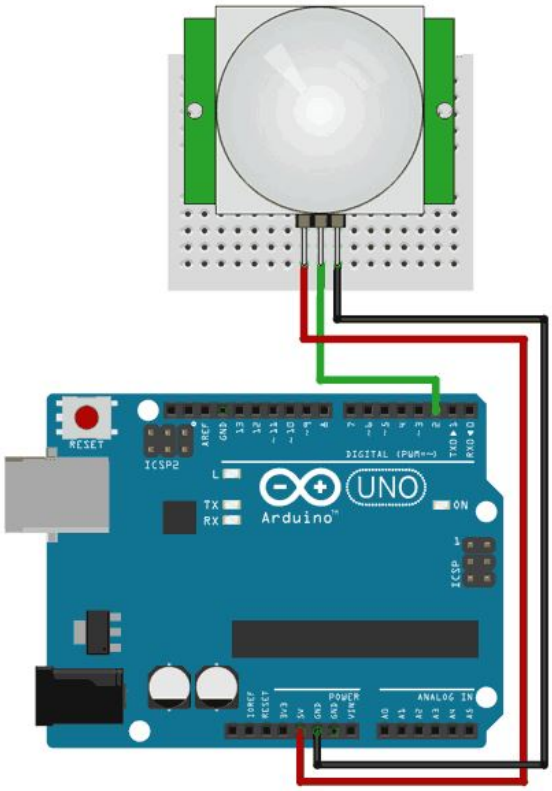


Figura 16: Esquema Microcontrolador + Sensor

4.1.2 SENSOR DE CALIDAD DE AIRE – GROVE AIR QUALITY SENSOR v1.3

Este sensor está diseñado para un monitoreo integral de la condición del aire interior. Responde a una amplia gama de gases nocivos, como monóxido de carbono, alcohol, acetona, diluyente, formaldehído, etc. Debido al mecanismo de medición, este sensor no puede generar datos específicos para describir las concentraciones de gases objetivo cuantitativamente. Pero sigue siendo lo suficientemente competente como para usarse en aplicaciones que requieren solo resultados cualitativos, como pulverizadores de actualización automática y sistemas de ciclado automático de aire.

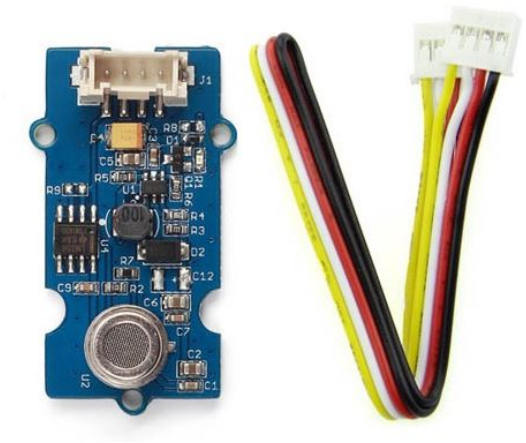


Figura 17: Sensor Calidad de Aire

Descripción:

- Dimensiones: 40mm x 20 mm x 15 mm
- Peso: 9g
- Bateria: Excluida
- Sensor Winsen MP503
- Potencia 5V o 3.3V

Debido a su alto coste se ha optado por comprar solo uno.

4.1.3 SENSOR DE LUZ – BH1750

El BH1750 es un sensor digital del nivel de luz que puede ser conectado con facilidad a un autómata o procesador como Arduino para formar un luxómetro.

A diferencia de otros sistemas de medición del nivel de luz, como por ejemplo las resistencias LDR, la respuesta espectral del BH1750 está diseñada para ser similar a la del ojo humano, por lo que son capaces de proporcionar la medición de lux.

Recordamos que los luxes son la unidad del sistema internacional para la iluminancia. La iluminancia es la relación entre el flujo luminoso (la cantidad de luz emitida por una fuente de luz) y la superficie en la que se mide.

A modo de referencia, para mostrar un orden de magnitud de luxes, en la siguiente tabla se muestran algunos ejemplos típicos de iluminancia.

Situación	Luxes
Noche	0.001-0.02
Luna llena	0.2-0.6
Día nublado, en interior	5-50
Día nublado, en exterior	50-500
Día soleado, en interior	100-1000
Bajo luz directa del sol	100.000
Habitación, salón	150-300
Mesa oficina/lectura	500-700
Supermercados/exposiciones	750-1000
Mesas dibujo/trabajo	1000-1500

Figura 18: Tabla Luz

El BH1750 tiene un amplio rango de medición ajustable desde los 0.11 a 100000 lux, por lo que es capaz de medir en casi cualquier situación de iluminación. Incorpora un Conversor Analogico-Digital (ADC) de 16bits que proporciona una resolución de 65535 niveles.

El sensor BH1750 tiene una baja influencia al espectro infrarrojo, rechazo al ruido de 50/60 Hz (luz artificial) y alta independencia del origen de la fuente de luz (luz natural, halógenos, LED, incandescencia).

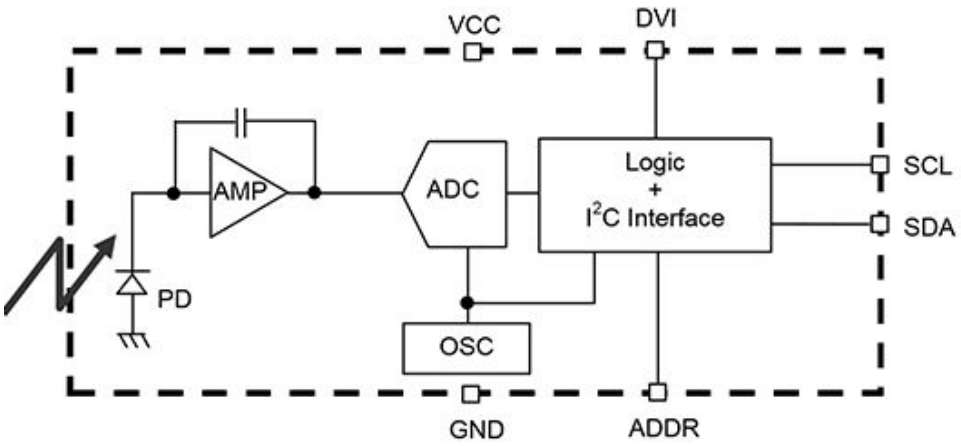


Figura 19: Circuito eléctrico Sensor Luz

La comunicación se realiza a través del bus I2C, por lo que es sencillo obtener los datos medidos. La tensión de alimentación es de bajo voltaje entre 2.4 a 3.6V.

Frecuentemente se encuentran integrados en módulos que incorporan la electrónica necesaria para conectarla de forma sencilla a un Arduino**.** En la mayoría de los módulos, esto incluye un regulador de voltaje que permite alimentar directamente a 5V.

El BH1750 se emplea, principalmente, para regular la retroiluminación de LCDs y Keypads en dispositivos móviles, así como para regular la iluminación en cámaras digitales.

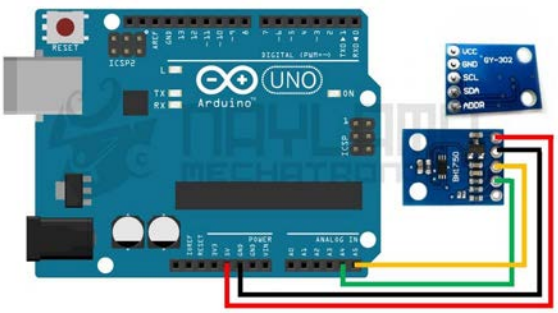


Figura 20: Esquema Microcontrolador + Sensor

4.1.4 SENSOR DE TEMPERATURA Y HUMEDAD – DHT22

El DHT11 y el DHT22 (o AM2302) son dos modelos de una misma familia de sensores, que permiten realizar la medición simultánea de temperatura y humedad.

Estos sensores disponen de un procesador interno que realiza el proceso de medición, proporcionando la medición mediante una señal digital, por lo que resulta muy sencillo obtener la medición desde un microprocesador como Arduino.

Ambos sensores presentan un encapsulado de plástico similar. Podemos distinguir ambos modelos por el color del mismo. El DHT11 presenta una carcasa azul, mientras que en el caso del sensor DHT22 el exterior es blanco.

De ambos modelos, el DHT11 es el modelo más básico, y cuenta peores características técnicas. El DHT22 es el modelo superior pero, por contra, tiene un precio superior.

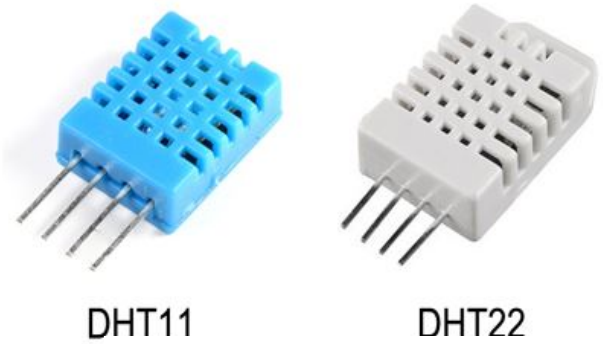


Figura 21: Sensor Humedad y Temperatura

Las características del DHT11 son realmente escasas, especialmente en rango de medición y precisión.

- Medición de temperatura entre 0 a 50, con una precisión de 2ºC
- Medición de humedad entre 20 a 80%, con precisión del 5%.
- Frecuencia de muestreo de 1 muestras por segundo (1 Hz)

El DHT11 es un sensor muy limitado que podemos usar con fines de formación, pruebas, o en proyectos que realmente no requieran una medición precisa.

Por el contrario, el modelo DHT22 tiene unas características mucho más aceptables.

- Medición de temperatura entre -40 a 125, con una precisión de 0.5ºC
- Medición de humedad entre 0 a 100%, con precisión del 2-5%.
- Frecuencia de muestreo de 2 muestras por segundo (2 Hz)

EL DHT22 (sin llegar a ser en absoluto un sensor de alta precisión) tiene unas características aceptables para que sea posible emplearlo en proyectos reales de monitorización o registro, que requieran una precisión media.

En este trabajo se ha usado el modelo DHT22.

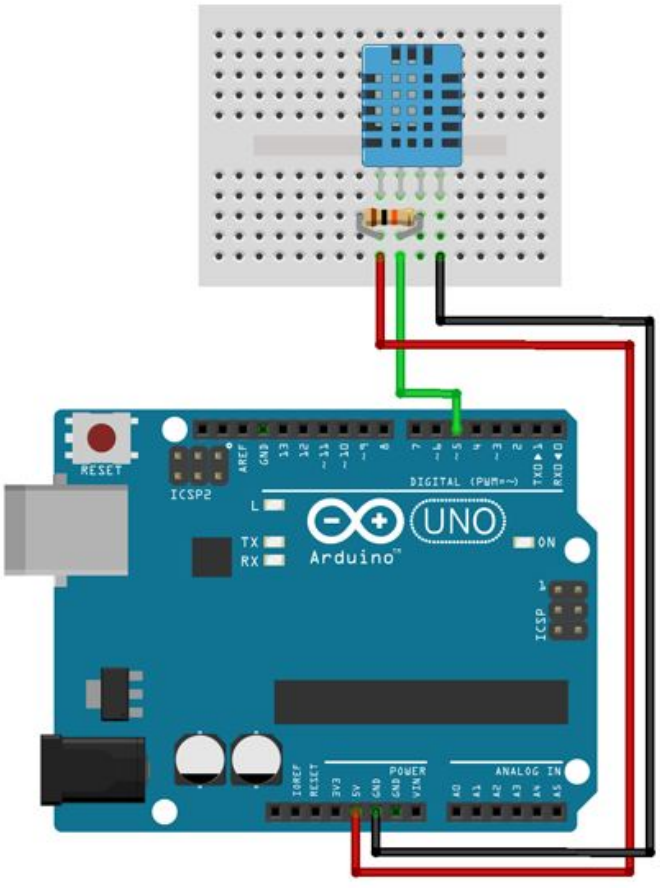


Figura 22: Esquema Microcontrolador + Sensor

4.1.5 SENSOR DE GASES – CCS811

El sensor de calidad del aire CCS811 es un sensor de gas digital de potencia ultra baja que integra un sensor de gas MOX (óxido de metal) para detectar una amplia gama de VOC (compuestos orgánicos volátiles) para el monitoreo de la calidad del aire interior con una MCU (unidad de microcontrolador) integrada. MCU consta de ADC (convertidor de analógico al digital) y la interfaz I2C. Se basa en una tecnología de micro placas únicas de *ams* que brinda soluciones altamente confiables para sensores de gas, con bajo consumo de energía.

Figura 23: Sensor de Gases

Descripción:

- Voltaje de funcionamiento: 1.8V a 3.3V
- Interfaz: I2C
- Dirección predeterminada I2c: 0x5B
- Rango de medición de TVOC: 0 a 1187 ppb (partes por mil millones)
- Rango de medición de eCO2: 400 a 8192 ppm (partes por millón)
- Modo de funcionamiento: 5 diferentes
- Baja potencia optimizada
- Conexión opcional para sensor de temperatura NTC
- Tamaño: 18 x 18 x 2.5 mm
- Peso: 1.5g

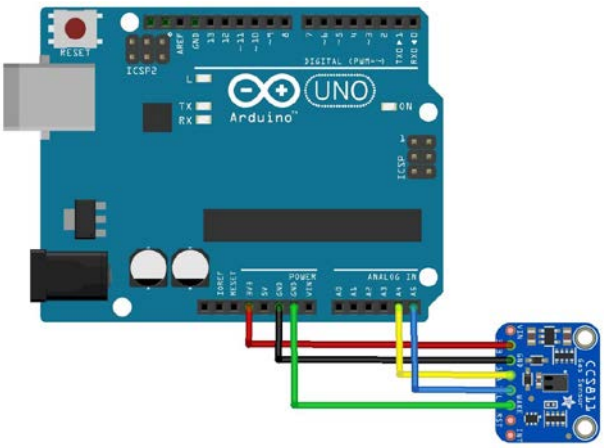


Figura 24: Esquema Microcontrolador + Sensor

4.2 MICROCONTROLADORES

Se han usado dos tipos diferentes de micocontroladores con el objetivo de probar diferentes tecnologías de comunicación.

4.2.1 ARDUINO WIFI Rev2

El Arduino Uno WiFi Rev2 es un Arduino Uno con módulo WiFi integrado. La placa se basa en el Microchip MEGA4809 con módulos WiFi ESP32 u-blox NINA-W13 integrados. El módulo NINA-W13 es un SoC autónomo con protocolo TCP / IP integrado que puede proporcionar acceso a una red WiFi.

El Arduino Uno WiFi Rev2 se programa utilizando el software Arduino (IDE), nuestro entorno de desarrollo integrado común a todas nuestras placas y que se ejecuta tanto en línea como fuera de línea.

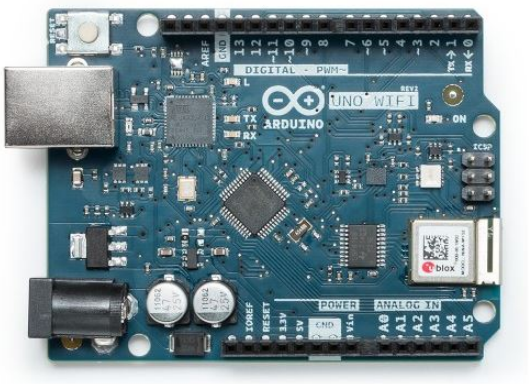


Figura 25: Microcontrolador Arriba (Rev2)

Figura 26: Microcontrolador Abajo (Rev2)

Descripción:

- Microcontrolador: ATMEGA4809
- Voltaje de Operación: 5V
- Voltaje de entrada (recomendado) : 7 - 12V
- Voltaje de entrada (límite) : 6 - 12V
- Los pines de E / S digitales: 14 - 5 proporcionan salida PWM
- Pines de E / S digitales PWM: 5

- Pines de entrada analógica: 6
- Corriente DC por I / O Pin: 20 mA
- Corriente DC para 3.3V Pin : 50 mA
- Memoria Flash: 48 KB (ATMEGA4809)
- SRAM: 6.144 B (ATMEGA4809)
- EEPROM: 256 Bytes (ATMEGA4809)
- Velocidad de reloj: 16 MHz
- LED_BUILTIN: 25
- Longitud: 68.6 mm
- Ancho: 53.4 mm
- Peso: 25 g

4.2.2 ARDUINO UNO

El componente principal es un microcontrolador de ATMEL, el ATMEGA 328P. En él está el software del cargador de arranque Arduino que le permite ejecutar programas.

El Arduino UNO tiene 14 pines de E / S digitales, seis de los cuales se pueden usar como canales PWM (permitiendo, por ejemplo, la atenuación de los LED). Además, hay seis pines de entrada analógica disponibles.

La placa maneja la comunicación a través de UART, SPI e I2C (TWI).

Está conectado al ordenador a través de USB para el intercambio de datos y la programación. Dado que el microcontrolador no puede comunicarse directamente a través de USB, hay un segundo microcontrolador (ATmega8U2) en el Arduino UNO que hace la "traducción".

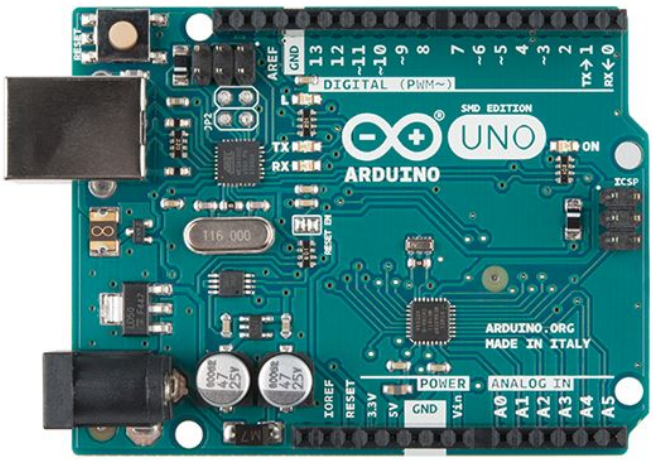


Figura 27: Microcontrolador Arriba (UNO)

Figura 28: Microcontrolador Abajo (UNO)

Descripción:

- Microcontrolador: ATMEGA4809
- Voltaje de Operación: 5V
- Voltaje de entrada (recomendado) : 7 - 12V
- Voltaje de entrada (límite) : 6 - 12V
- Los pines de E / S digitales: 14 - 5 proporcionan salida PWM
- Pines de E / S digitales PWM: 6
- Pines de entrada analógica: 6
- Corriente DC por I / O Pin: 20 mA
- Corriente DC para 3.3V Pin : 50 mA
- Memoria Flash: 32 KB (ATMEGA328P)
- SRAM: 2 KB (ATMEGA328P)
- EEPROM: 1 KB (ATMEGA328P)
- Velocidad de reloj: 16 MHz
- LED_BUILTIN: 13
- Longitud: 68.6 mm
- Ancho: 53.4 mm
- Peso: 25 g

4.2.3 ARDUINO UNO WIFI SHIELD

El Arduino WiFi Shield permite que una placa Arduino se conecte a Internet utilizando la especificación inalámbrica 802.11 (WiFi). Se basa en el sistema de LAN inalámbrica HDG204 802.11b / g incluido en el paquete. Un AT32UC3 proporciona una pila de red (IP) capaz de TCP y UDP. El escudo de WiFi se conecta a una placa Arduino utilizando encabezados largos de alambre que se extienden a través del escudo. Esto mantiene el diseño del pin intacto y permite apilar otro escudo en la parte superior.

El WiFi Shield puede conectarse a redes inalámbricas que funcionan de acuerdo con las especificaciones 802.11by 802.11g.

Existe una ranura para tarjeta micro-SD integrada, que se puede usar para almacenar archivos para servir en la red. Es compatible con Arduino Uno y Mega. Se puede acceder al lector de tarjetas microSD incorporado a través de la Biblioteca SD. Al trabajar con esta biblioteca, SS está en el Pin 4.

Arduino se comunica con el procesador del escudo Wifi y la tarjeta SD utilizando el bus SPI (a través del encabezado ICSP). Esto está en los pines digitales 11, 12 y 13 en el Uno y los pines 50, 51 y 52 en el Mega. En ambas placas, el pin 10 se usa para seleccionar el HDG204 y el pin 4 para la tarjeta SD. Estos pines no se pueden usar para E / S generales. En el Mega, el pin SS del hardware, 53, no se usa para seleccionar el HDG204 o la tarjeta SD, pero debe mantenerse como salida, de lo contrario la interfaz SPI no funcionará.

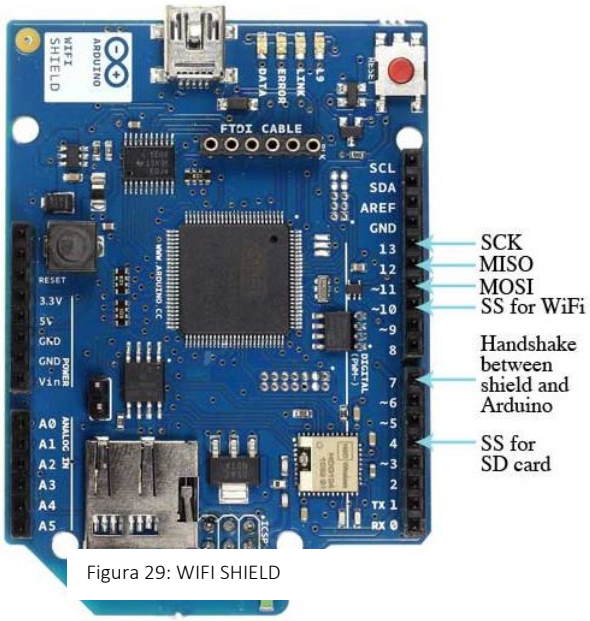


Figura 29: WIFI SHIELD

El pin digital 7 se utiliza como un intermediario entre el escudo WiFi y el Arduino, y por ello no debe usarse.

El escudo puede conectarse a redes encriptadas que usan encriptación WPA2 Personal o WEP. También se puede conectar a redes abiertas. Una red debe transmitir su SSID para que el escudo pueda conectarse.

4.3 CARGADOR

Para asegurar el suministro de energía de los micocontroladores (y por lo tanto de los sensores) se ha utilizado un cargador para cada uno de los cuatro grupos de sensores.

Descripción:

- compatible con Arduino Uno REV 3
- Salida: 9V 1000mA
- Conector (exterior): 5,5mm x 2,1mm
- Compatible 9V 500mA , 1000mA

Figura 30: Cargador

Capítulo 5 METODOLOGIA

En el siguiente esquema se muestra el camino que recorre la información captada por los sensores hasta llegar a la base de datos y al cliente web.

- Los sensores captan la información a través de Arduino
- La información es enviada al protocolo de suscripción y publicación de MQTT
- Posteriormente la información es recibida en el entorno de programación visual de Node Red
- Una vez llegada a Node Red se envía a través de las “function Nodes” a la base de datos MySQL
- Paralelamente se crea el cliente Web desde Node Red

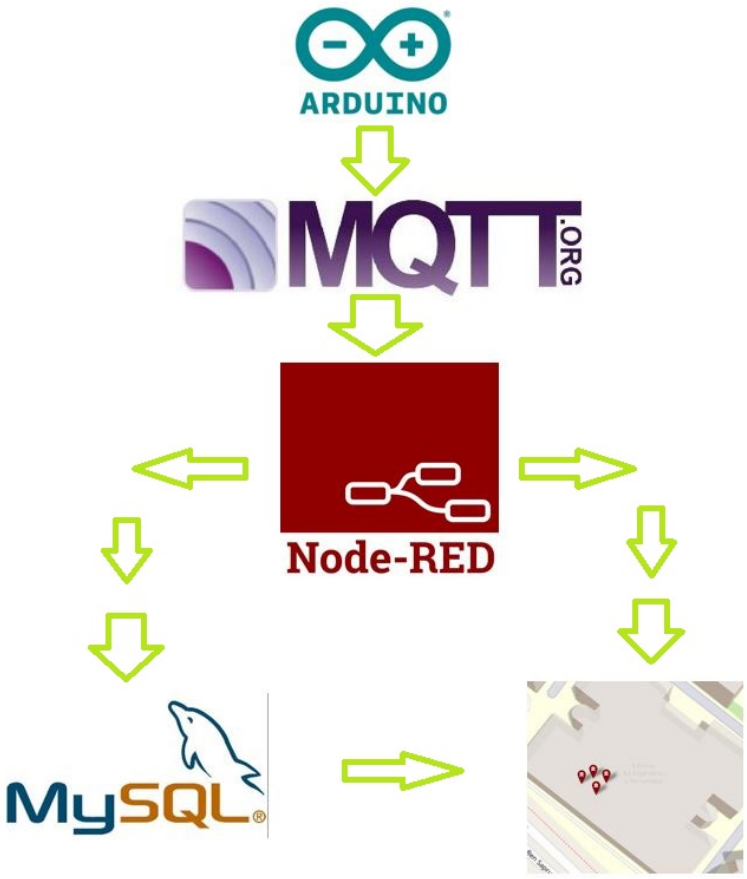


Figura 31: Camino de los datos

5.1 ARDUINO

En este trabajo se han usado 4 códigos principales, uno para cada Grupo de sensores. Cada uno de los grupos consta de las siguientes piezas:

➔ Grupo 1:

- Sensores: DHT22, CCS811, BH1750, PIR SR505
- Microcontroladores: Arduino WiFi Rev2
- Cargador

➔ Grupo 2:

- Sensores: DHT22, CCS811, BH1750, PIR SR505
- Microcontroladores: Arduino WiFi Rev2
- Cargador

➔ Grupo 3:

- Sensores: DHT22, CCS811, BH1750, PIR SR505
- Microcontroladores: Arduino WiFi Rev2
- Cargador

➔ Grupo Shield:

- Sensores: DHT22, CCS811, BH1750, PIR SR505, Air Quality Sensor v1.3
- Microcontroladores: Arduino Uno, Arduino Uno Wifi Shield
- Cargador

Como se puede observar, todos los grupos (excepto el Grupo Shield) usan las mismas piezas. El grupo shield usa otro microcontrolador, que necesita un Shield adicional para conectarse a la red wifi que se ha usado para este proyecto. Aparte del microcontrolador, el grupo shield también contiene el único sensor de Calidad de Aire.

Por lo tanto el código de los grupos 1, 2 y 3 será muy similar, variando solo el código del grupo shield.

5.1.1 CODIGO GRUPO 1, 2 y 3

Como ya se ha explicado en el apartado de "Material y Método" la estructura que sigue el código de Arduino es bastante simple.

Está formado por las dos funciones básicas "void setup" que se ejecuta solo una vez y la función "void loop" que se repite una y otra vez.

LIBRERIAS:

En el mundo de Arduino una librería es una colección de funciones que se incluyen de una manera muy sencilla y explicita en nuestro "Sketch" y que proporciona una cierta funcionalidad especifica.

#include <ArduinoMqttClient.h>

Esta biblioteca proporciona un cliente para realizar mensajes simples de publicación / suscripción con un servidor que admita MQTT.

#include <WiFiNINA.h>

Esta librería nos permite usar la función Wifi del microcontrolador (Arduino Wifi Rev2).

#include <DHT.h>

Esta librería nos permite la lectura del sensor DHT22 sin tener que preocuparnos el protocolo de comunicación entre el Arduino y los dos sensores.

#include <Wire.h>

Esta librería nos permite la comunicación con dispositivos 12C / TWI

#include <BH1750.h>

Esta librería nos permite leer el sensor de luz digital BH1750.

#include "Adafruit_CCS811.h"

Esta librería nos permite leer el sensor de gases CCS811.

CONSTANTES:

```
#define  SIGNAL_PIN 2        // Se asigna el pin 2 a la lectura del sensor de movimiento

#define  DHTPIN 5            // Se asigna el pin 5 a la lectura del sensor de Humedad y Temperatura

char ssid[] = "Cartografia";     //     Nombre de la red Wi-Fi a la que nos queremos conectar

char pass[] = "C@rtoU119";  // Contraseña de la red Wi-Fi de la red a la que nos queremos conectar.

int  port  = 1883;  // se asigna el puerto


const char* topicName = "/Sensor/Grupo1/Temperatura";

const char* topicName2 = "/Sensor/Grupo1/Humedad";

const char* topicName3 = "/Sensor/Grupo1/Luz";

const char* topicName4 = "/Sensor/Grupo1/Movimiento";

const char* topicName5 = "/Sensor/Grupo1/CO2";

const char* topicName6 = "/Sensor/Grupo1/TVOC";
```

Aquí se declaran los tópicos a los que se va a suscribir. Esta sería la única parte del código que va a variar en los en los Grupos 1 , 2 y 3, ya que cada uno de los grupos tendrá otros tópicos a los que se va a suscribir para mandar la información.

Un ejemplo para el Grupo 1 seria:
`const char* topicName = "/Sensor/Grupo1/Temperatura";`

Un ejemplo para el Grupo 2 seria:
`const char* topicName = "/Sensor/Grupo2/Temperatura";`

Un ejemplo para el Grupo 3 seria:
`const char* topicName = "/Sensor/Grupo3/Temperatura";`

VOID SETUP:

A continuación se ve el código de la función "Void Setup" del Grupo 1, Grupo 2 y Grupo 3.

```
void setup() {
  Serial.begin(9600);        //Inicializa el serial y espera a que se
abra el puerto
  while (!Serial) {          // espera a que se conecte el puerto
serie. Necesario solo para puerto USB nativo
  }
  Serial.print("Attempting to connect to WPA SSID: ");    // intenta
conectar a la red wifi
  Serial.println(ssid);
  while (WiFi.begin(ssid, pass) != WL_CONNECTED) {      //fallo,
vuelve a intentar
    Serial.print(".");
    delay(5000);
  }

  Serial.println("You're connected to the network");   // Se devuelve
por el serial ese mensaje
  Serial.println();
  Serial.print("Attempting to connect to the MQTT broker: "); // Se
devuelve por el serial ese mensaje
  Serial.println(server);

  if (!mqttClient.connect(server, port)) {                //prueba
conectarse al servidor MQTT
    Serial.print("MQTT connection failed! Error code = ");  // Se
devuelve por el serial ese mensaje de Error
    Serial.println(mqttClient.connectError());

    while (1);
  }
  Serial.println("You're connected to the MQTT broker!");   // Se
devuelve por el serial ese mensaje de Exito
  Serial.println();
   dht.begin();                                             // Se
inicializa el sensor DHT
  Wire.begin();                                             // Se
inicializa la comunicacion 12c
  lightMeter.begin();                                       // Se
inicializa el sensor BHT1750
  pinMode(SIGNAL_PIN, INPUT);

  if(!ccs.begin()){
 Serial.println("Failed to start sensor! Please check your wiring.");
// Se devuelve por el serial ese mensaje de Error
  while(1);
 }
while(!ccs.available());
float temp = ccs.calculateTemperature();  // Se ha inicializado
correctamente el sensor CSS811 y se procede a calcular la temperatura
ccs.setTempOffset(temp - 25.0);

}
```

VOID LOOP:

A continuación se ve el código de la función “Void Loop” del Grupo 1, Grupo 2 y Grupo 3.

```
void loop() {                    // sondear nuevos mensajes MQTT y
enviar mensajes vivos

  mqttClient.poll();

    float temp = dht.readTemperature();  // Se declara la variable
"temp" y se le asigna el valor leido por el sensor de temperatura
    char buffer[10];                     // Se define el numero de
caracteres que puede tomar la variable
    dtostrf(temp,0, 0, buffer);          // Se convierte el valor de
la variable "temp" en un String
    mqttClient.beginMessage(topicName);  // Se inicializa el mensaje y
se elige el topico al que sera mandado
    mqttClient.print(buffer);            // Se manda el valor que ha
llegado desde el sensor de Temperatura
    mqttClient.endMessage();           // Se cierra el mensaje
delay(60000);                         // Se espera 6 segundos

    float hum = dht.readHumidity();  // Se declara la variable "hum" y
se le asigna el valor leido por el sensor de humedad
    dtostrf(hum,0, 0, buffer);          // Se convierte el valor de la
variable "temp" en un String
    mqttClient.beginMessage(topicName2); // Se inicializa el mensaje y
se elige el topico al que sera mandado
    mqttClient.print(buffer);           // Se manda el valor que ha
llegado desde el sensor de Humedad
    mqttClient.endMessage();            // Se cierra el mensaje
delay(60000);                           // Se espera 6 segundos

    float lux = lightMeter.readLightLevel();  // Se declara la
variable "lux" y se le asigna el valor leido por el sensor de Luz
    dtostrf(lux,0, 0, buffer);          // Se convierte el valor de la
variable "lux" en un String
    mqttClient.beginMessage(topicName3); // Se inicializa el mensaje y
se elige el topico al que sera mandado
    mqttClient.print(buffer);         // Se manda el valor que ha
llegado desde el sensor de Luz
    mqttClient.endMessage();         // Se cierra el mensaje
delay(60000);                        // Se espera 6 segundos


if (digitalRead(SIGNAL_PIN) == HIGH) {   // Se comprueba si el pin 2
ha registrado una entrada de datos
    mqttClient.beginMessage(topicName4); // Se inicializa el mensaje y
se elige el topico al que sera mandado
```

```
    mqttClient.print("1");            // Se manda el valor "1" que
significa que se ha detectado un movimiento
    mqttClient.endMessage();       // Se cierra el mensaje
  } else{                         // en caso contrario
    mqttClient.beginMessage(topicName4); // Se inicializa el mensaje y
se elige el topico al que sera mandado
    mqttClient.print("0");         // Se manda el valor "0" que
significa que NO se ha detectado ningun movimiento
    mqttClient.endMessage();       // Se cierra el mensaje
    }
delay(60000);                     // Se espera 6 segundos

if (ccs.available()) {                      // si hay un dato del
sensor CSS811 disponible se procede a calcular la temperatura
   float temp = ccs.calculateTemperature();
  if(!ccs.readData()){              // se comprueba si el sensor de
gases ha leido algun dato
    mqttClient.beginMessage(topicName5); // Se inicializa el mensaje y
se elige el topico al que sera mandado
    mqttClient.print(ccs.geteCO2());    // Se manda el valor de CO2
que ha llegado desde el sensor de gases
    mqttClient.endMessage();       // Se cierra el mensaje

  } else{                          // en caso contrario
    mqttClient.beginMessage(topicName5); // Se inicializa el mensaje y
se elige el topico al que sera mandado
    mqttClient.print("Error");     // Se manda el mensaje "Error"
    mqttClient.endMessage();  // Se cierra el mensaje
    }
delay(60000);                // Se espera 6 segundos
}

if (ccs.available()) {                      // si hay un dato del
sensor CSS811 disponible se procede a calcular la temperatura
   float temp = ccs.calculateTemperature();
  if(!ccs.readData()){                  // se comprueba si el sensor
de gases ha leido algun dato
    mqttClient.beginMessage(topicName6); // Se inicializa el mensaje y
se elige el topico al que sera mandado
    mqttClient.print(ccs.getTVOC());  // Se manda el valor de TVOC que
ha llegado desde el sensor de gases
    mqttClient.endMessage();       // Se cierra el mensaje

  } else{                                // en caso contrario
    mqttClient.beginMessage(topicName6); // Se inicializa el mensaje y
se elige el topico al que sera mandado
    mqttClient.print("Error");     // Se manda el mensaje "Error"
    mqttClient.endMessage();     // Se cierra el mensaje
    }
delay(60000);                   // Se espera 6 segundos
}
}
```

5.1.2 CODIGO GRUPO SHIELD

El código del grupo shield tiene algunas variaciones con respecto al código anterior ya que cambia la manera de conectarse al MQTT y además incluye el sensor de Calidad de Aire.

LIBRERIAS:

Este código incluye tres librerías que el código anterior no tenía. Dos para la conexión al MQTT y una para el sensor de calidad de Aire.

```
#include <WiFi.h>
```

Esta librería permite que una placa de Arduino que además disponga de un Arduino Wifi Shield (como es el caso) se conecte a internet.

```
#include <PubSubClient.h>
```

Una librería que brinda soporte para MQTT

```
#include"AirQuality.h"
```

Esta librería nos permite usar el sensor de Calidad de Aire.

CONSTANTES:

Las constantes se mantienen bastante similares en su conjunto. La única variación se da en los tópicos a los que se van a suscribir.

```
const char* topicName = "/Sensor/ GrupoShield /Temperatura";

const char* topicName2 = "/Sensor/ GrupoShield /Humedad";

const char* topicName3 = "/Sensor/ GrupoShield /Luz";

const char* topicName4 = "/Sensor/ GrupoShield /Movimiento";

const char* topicName5 = "/Sensor/ GrupoShield /CO2";

const char* topicName6 = "/Sensor/ GrupoShield /TVOC";
```

```
const char* topicName6 = "/Sensor/ GrupoShield/Calidad_Aire";
```

VOID SETUP:

Hay dos cambios en la función “setup” en comparación con el código anterior.

El código para conectarse a la red wifi varia ligeramente y en segundo lugar se inicializa el Sensor de Calidad de Aire.

```
Serial.println("Attempting to connect to WPA network...");
// Se devuelve este mensaje de intento de conexion por el serial
  status = WiFi.begin(ssid, pass);// Se intenta conectar a la red wifi

  if ( status != WL_CONNECTED) {
// En caso de no conseguir conectarse a la red el codigo se para aqui
    Serial.println("Couldn't get a wifi connection");
 // Se devuelve este mensaje de Error por el serial
    while(true);
  }
  else {
    Serial.println("Connected to network");
 // En caso de exito se devuelve este mensaje
  }
```

VOID LOOP:

En la función loop del código del Grupo Shield varían principalmente dos cosas.

- La manera de suscribirse y publicar un mensaje at raves de MQTT

```
client.publish("/Sensor/GrupoShield/Temperatura", buffer);
```

Como se puede ver se podría considerar más fácil ya que solo se necesita una línea de código y no tres como en el caso de los Arduino WiFi Rev2.

- Se añade la parte de código del sensor de Calidad de Aire.

```
   current_quality=airqualitysensor.slope();
    if (current_quality >= 0)
// si se recibe un dato valido entra en la siguiente funcion
    {
        if (current_quality==0)      // si el valor recibido es cero
            client.publish("/Sensor/GrupoShield/Calidad_Aire", "4");
  // Se publica el valor "4" en este topico
        else if (current_quality==1)    // si el valor recibido es uno
            client.publish("/Sensor/GrupoShield/Calidad_Aire", "3");
  // Se publica el valor "3" en este topico
        else if (current_quality==2)    // si el valor recibido es dos
            client.publish("/Sensor/GrupoShield/Calidad_Aire", "2");
  // Se publica el valor "2" en este topico
        else if (current_quality ==3)  // si el valor recibido es tres
            client.publish("/Sensor/GrupoShield/Calidad_Aire", "1");
  // Se publica el valor "1" en este topico
    }
```

- Un valor de 4 es el grado de contaminación más alto que mide el sensor de calidad de aire. 4 = Alta contaminación! Señal de fuerza activa!

- Un valor de 3 es el segundo grado de contaminación más alto que mide el sensor de calidad de aire. 3 = Alta contaminación!

- Un valor de 2 es el segundo grado de contaminación más bajo que mide el sensor de calidad de aire. 2 = Baja contaminación!

- Un valor de 1 es el grado de contaminación más bajo que mide el sensor de calidad de aire. 1 = Aire Fresco

5.1.3 ESQUEMA Y CIRCUITO ELECTRICO

A continuación se muestra el esquema de montaje de todos los sensores con el microcontrolador y el esquema eléctrico.

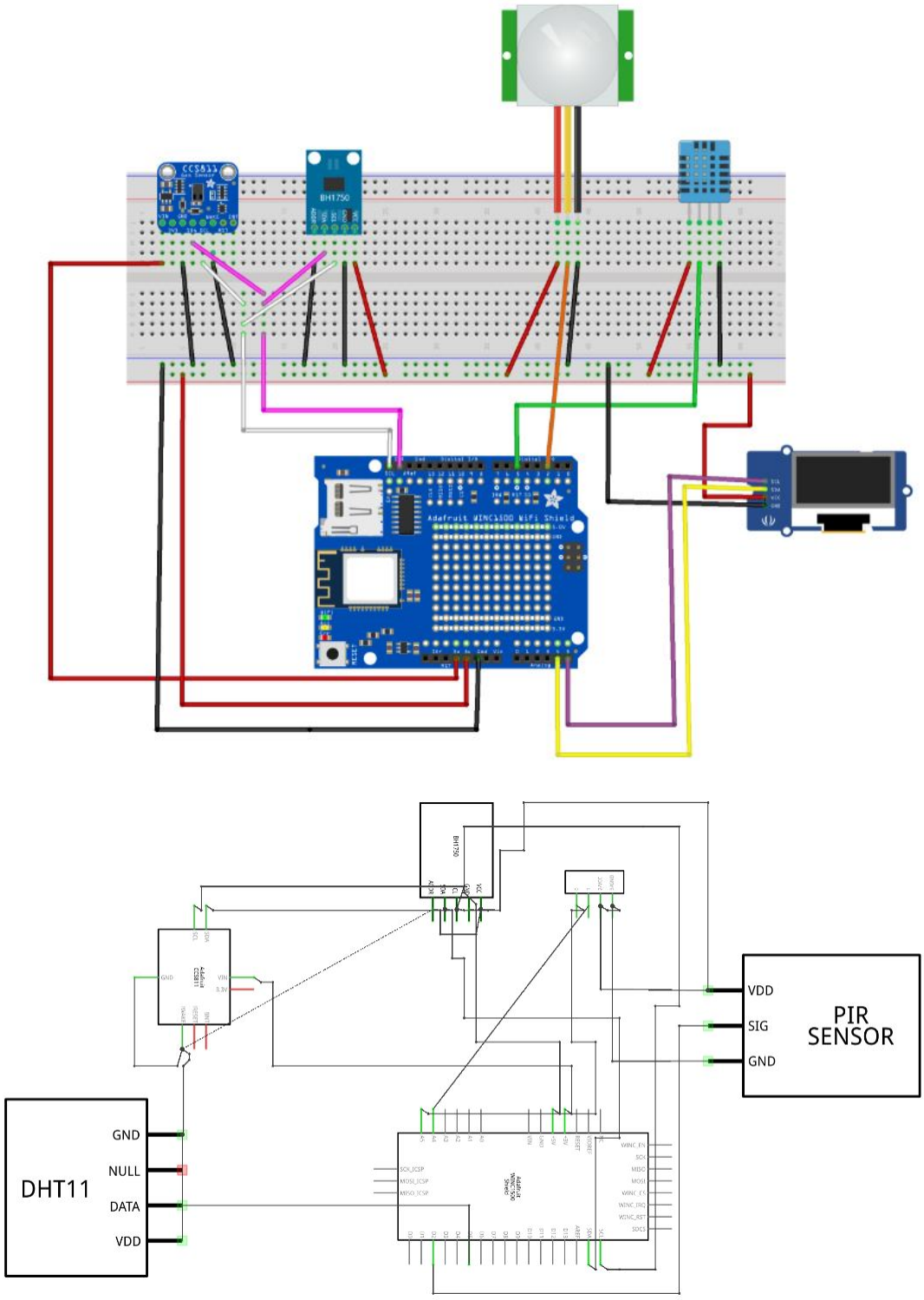


Figura 32: Esquema y Circuito Eléctrico Microcontrolador + Sensores

5.2 MQTT

El uso del bróker es bastante sencillo. Una vez instalado todo correctamente se procede con las configuraciones básicas para establecer una conexión. Para ello principalmente se introduce la IP del ordenador local en el que se está usando MQTT.

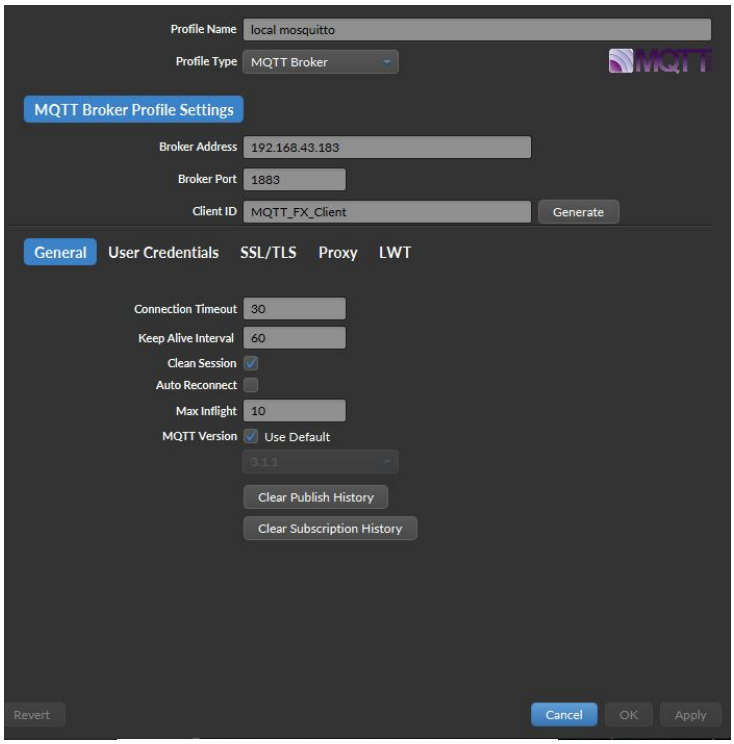


Figura 33: Configuración MQTT

Una vez establecida la conexión solo queda suscribirse a los tópicos establecidos en el código Arduino para recibir los datos de los sensores.

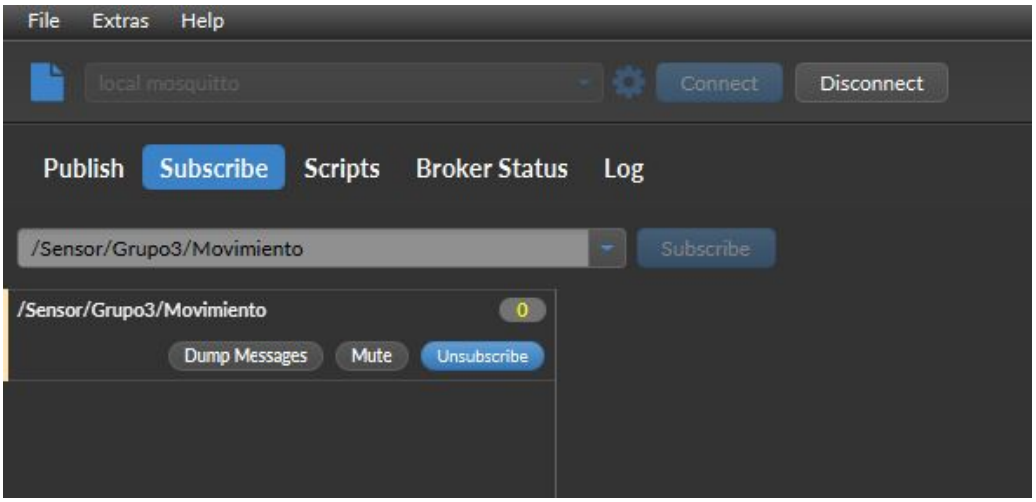


Figura 34: Suscribir a un Tópico en MQTT

Finalmente suscrito a todos los tópicos la apariencia seria la siguiente.

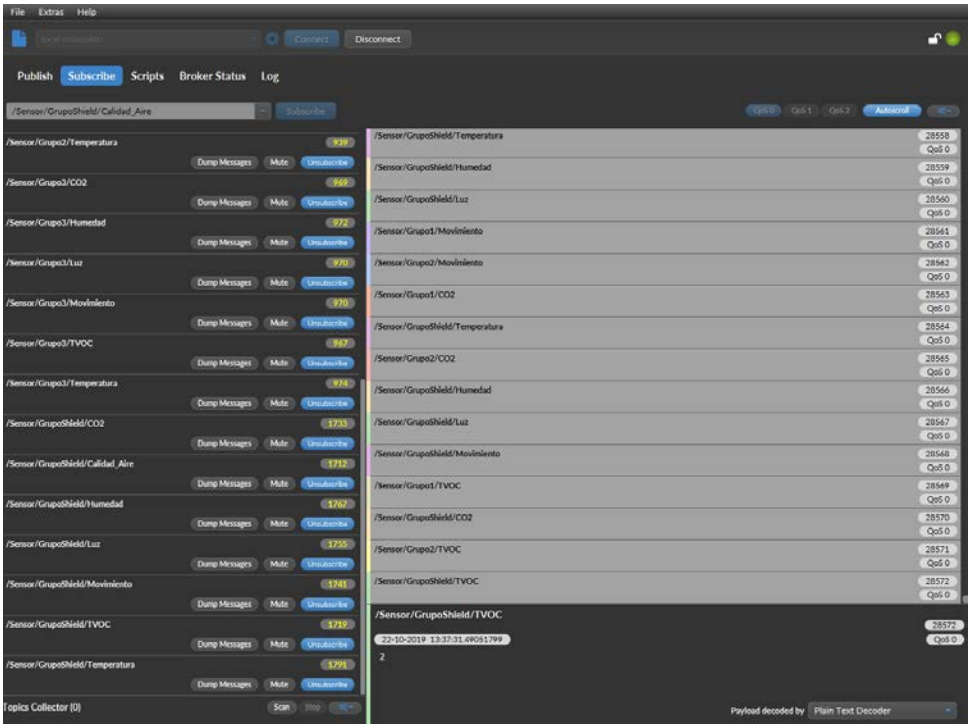


Figura 35: Listado de Tópicos suscritos

5.3 NODE RED

Como ya se ha explicado en el apartado de 2.3, Node Red es una herramienta de desarrollo basada en flujo para programación visual para conectar dispositivos de hardware, API y servicios en línea como parte de Internet de las cosas.

En este trabajo se ha usado Node Red como punto central de todos los elementos.

- Recibe los datos del MQTT
- Manda los datos entrantes de MQTT a la base de Datos (MySQL)
- Se crea el cliente web y el panel de monitorización, y se representan los datos entrantes en tiempo real.

5.3.1 NODE RED - MQTT

Para recibir los datos a través de MQTT es necesario crear un nodo MQTT. En dicho nodo únicamente hay que configurar la IP del servidor del que se vayan a recibir los datos y después suscribirse al tópico correspondiente.

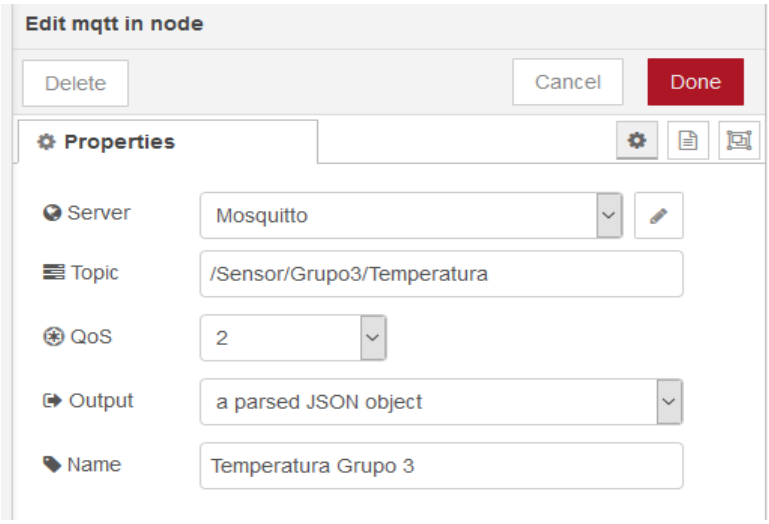


Figura 36: Configuración del nodo MQTT en NodeRed

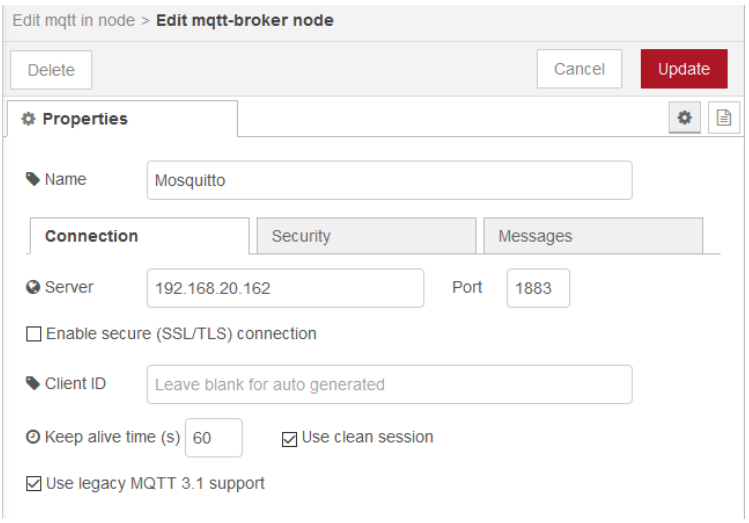


Figura 37: Configuración del nodo MQTT en Node Red

Adicionalmente se puede elegir el tipo de “Output” que tendrá el dato.

5.3.2 NODE RED - MySQL

Para insertar ahora el valor de entrada en la base de datos se necesitan otros dos nodos.

- Un nodo “function” en el que se coge el valor y se inserta en la tabla de las base de datos creada anteriormente

```
2
3 msg.topic = "INSERT INTO grupo3_temperatura ( `Temperatura`) VALUES ( "+ msg.payload+");"
4 return msg;
```

Figura 38: Código Node Red para insertar valor en la base de Datos

- Un nodo “mySQL” para conectarse con la base de datos.

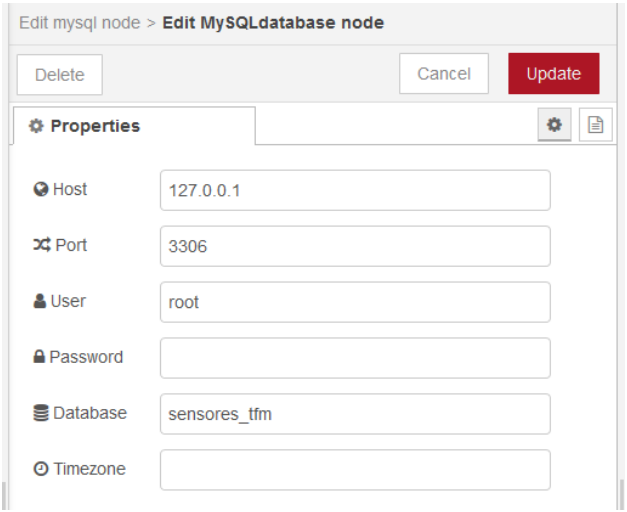


Figura 39: Configuración del nodo MySQL en NodeRed

Para establecer la conexión únicamente se necesita la IP del servidor de la base de Datos, el puerto, el usuario y el nombre propio que se le ha dado a la base de Datos.

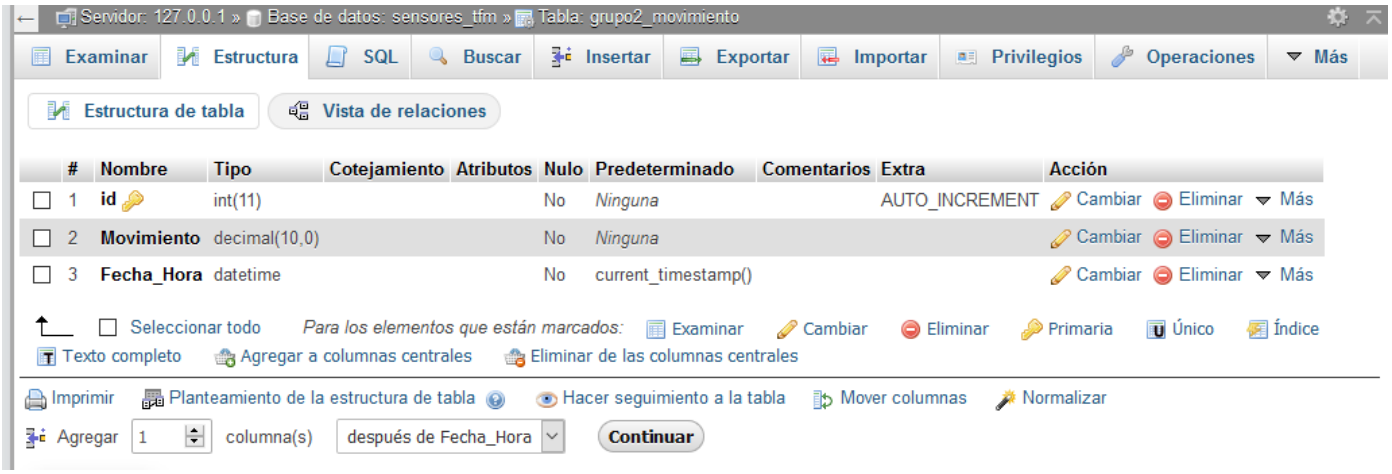


Figura 40: Vista de los campos de una tabla en la base de Datos

Con estos tres nodos unidos ya se podría recibir un dato a través de MQTT para después mandarlo y guardarlo en la base de Datos.

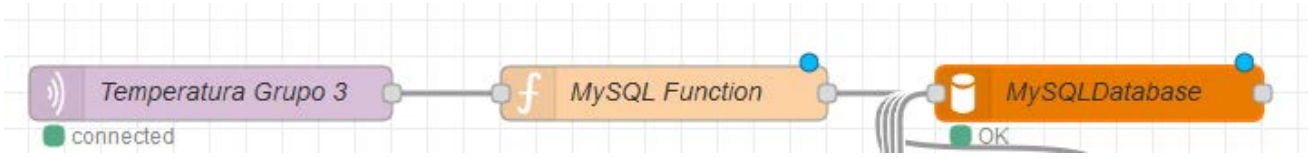


Figura 41: Nodos en Node Red para enviar datos a MySQl

Habría que crear estos nodos para guardar cada uno de los datos entrantes de todos los sensores que se están usando en este proyecto.

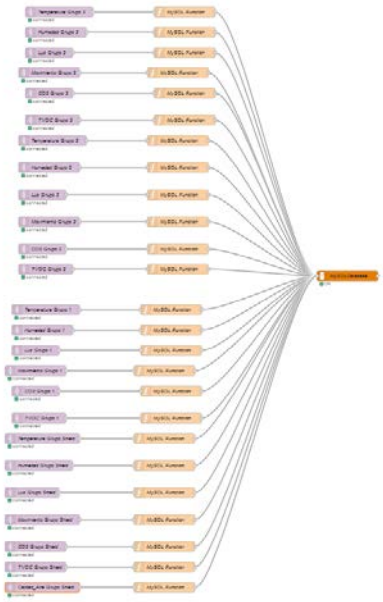


Figura 42: Todos los nodos usados para enviar los datos a MySQl

Para este trabajo finalmente se han creado 25 tablas diferentes en la base de datos.

- 6 Tablas para el Grupo 1
- 6 Tablas para el Grupo 2
- 6 Tablas para el Grupo 3
- 7 Tablas para el Grupo Shield

Hay 4 tablas para los sensores de temperatura, 4 tablas para los sensores de humedad, 4 tablas para los sensores de luz, 4 tablas para los sensores de movimiento, 4 tablas para los sensores de CO2, 4 tablas para los sensores de TVOC y 1 tabla para el sensor de calidad de aire.

Cada una de las tablas está formada por 3 columnas. La primera columna le asigna automáticamente una ID a cada dato entrante. La segunda columna contiene el valor captado por el sensor correspondiente. La tercera columna crea un valor de fecha y hora actual para cada uno de los datos que se almacena en esa tabla. La estructura de esta tabla se puede observar en la *Figura 40*.

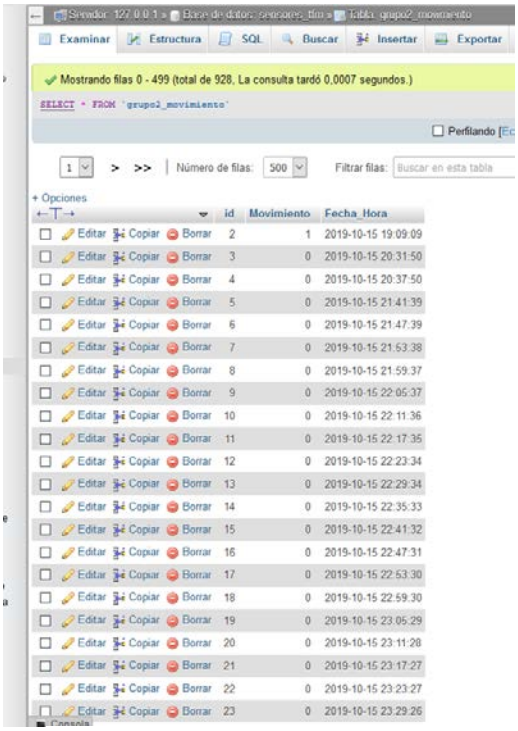


Figura 91: Tabla con datos de Movimiento del Grupo 2

5.3.3 NODE RED – CLIENTE WEB

Entre muchas de las cosas que ofrece Node Red encontramos las herramientas necesarias para crear un cliente web.
El objetivo de este cliente web es que cada uno de los cuatro grupos aparezca con su propio icono, y que cuando se clique encima aparezcan los valores en tiempo real que están tomando todos los sensores. La localización en el mapa seria exactamente la localización que tiene en la realidad.

Los nodos y pasos necesarios serían los siguientes.

1. Se necesita crear un nodo de tipo “function” para guardar las variables con los datos entrantes de MQTT.

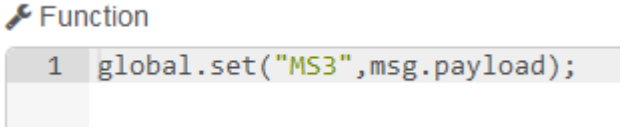


Figura 43: Función en Node Red para guardar variable

2. Una vez que todas las variables estén guardadas se crea otro nodo de tipo "*function*" para llamar a todas las variables declaradas anteriormente. Adicionalmente se añade una variable para la fecha y la hora que se va renovando conforme entran los datos.

```
var d = new Date();
var u = global.get("MS23");
var h= global.get("MS25");
var luz= global.get("MS27");
var co = global.get("MS29");
var tvoc =global.get("MS31");
var mo =global.get("MS33");
var ai =global.get("MS34");
var t = d.getTime();

msg.payload={lat:37.787235, lon:    -3.777897,name:"Grupo Shield", icon:"university","Fecha y Hora":d,
"Temperatura":u,  "Humedad":h,  "Luz":luz,  "CO2":co,  "TVOC":tvoc,"Movimiento":mo,"Calidad de Aire":ai};

return msg;
```

Figura 44: Función en Node Red para crear un punto con los datos recibidos de los sensores en el cliente web

3. El último paso sería crear un nodo de tipo "inject" para que los datos se vayan actualizando. En este caso se le ha dado un tiempo de 10 segundos. Finalmente se conecta todo con el nodo del Cliente Web.

Figura 45: Conjunto de nodos usados para el cliente web

Finalmente obtendríamos nuestro cliente web con los cuatro grupos de sensores y los datos de los sensores en tiempo real.
Se puede acceder al Cliente Web de forma local a través del siguiente enlace http://localhost:1880/worldmap/

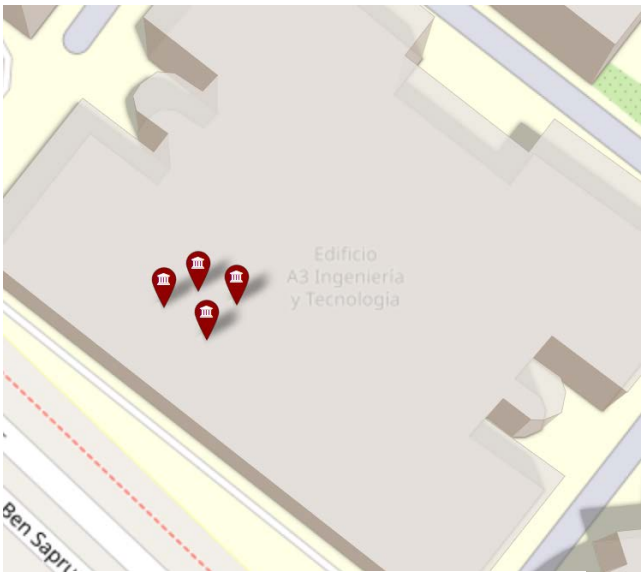


Figura 46: Cliente Web con los 4 puntos

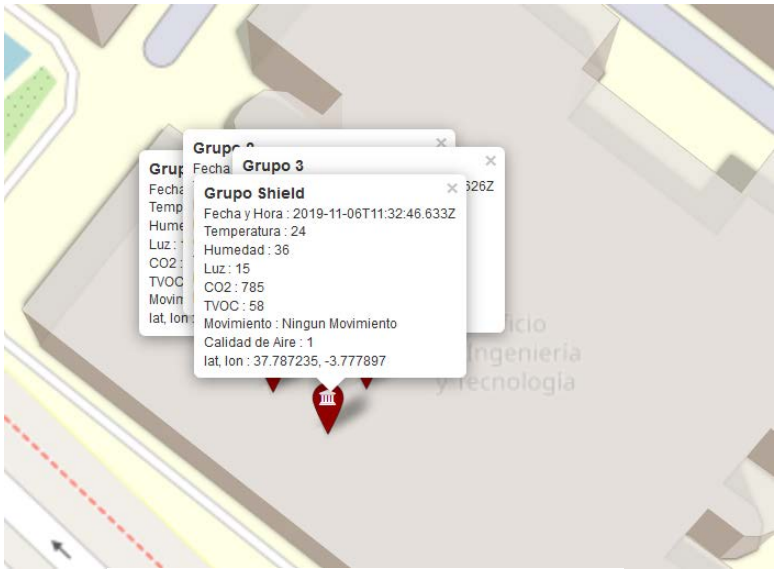


Figura 47: Cliente Web con Información de los 4 puntos

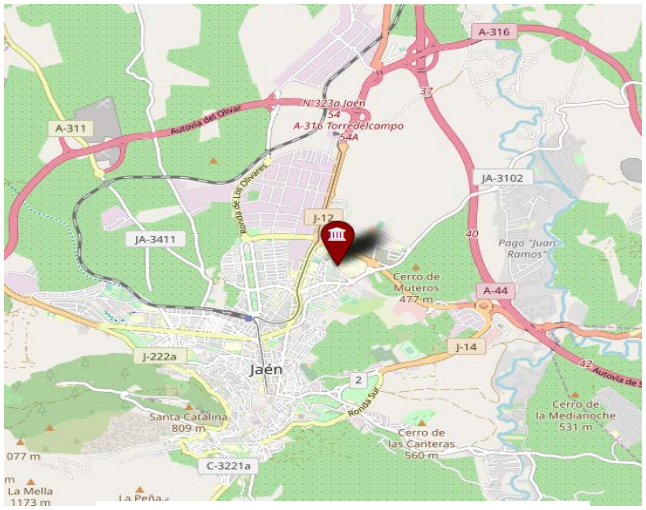


Figura 48: Cliente Web con fondo Open Street Map

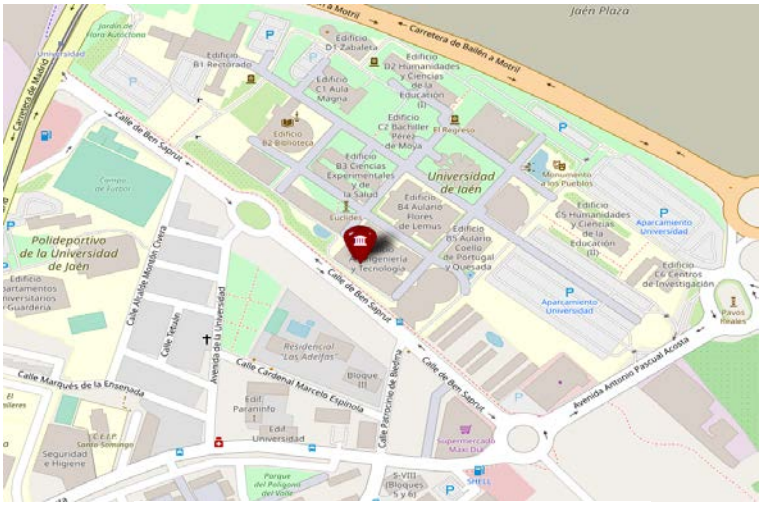


Figura 49: Cliente Web con fondo Open Street Map

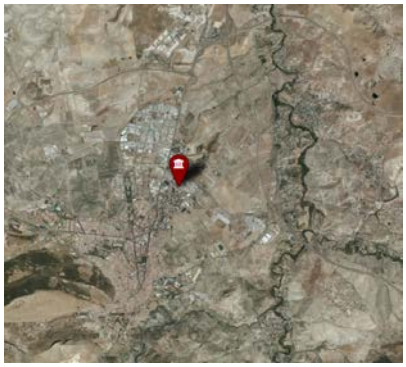


Figura 50: Cliente Web con fondo ESRI SATELITE

Figura 51: Cliente Web con fondo ESRI SATELITE

5.3.4 NODE RED – PANEL DE CONTROL

Aparte de un cliente web, se ha creado un panel de control para poder visualizar la información de una forma más gráfica. En dicho panel aparecen gráficos que representan el valor medio de los cuatro grupos de cada uno de los sensores en la ultima hora.

Para crear cada uno de estos gráficos se han seguido los siguientes pasos.

1. Crear un nodo "function" que llama a las variables que hemos almacenado como se ha explicado posteriormente. En la misma función se calcula el valor medio y se devuelve el valor.

Function

```
1  var msg1=global.get("MS3");
2  var msg2=global.get("MS4");
3  var msg3=global.get("MS22");
4  var msg4=global.get("MS23");
5
6
7  var MS5=(msg2+msg1+msg3+msg4)/(4);
8
9
10 msg.payload = MS5;
11
12
13 return msg;
```

Figura 52: función Node Red para llamar variables y hacer calculo

2. En el segundo paso se crean los nodos de tipo "Dashboard" para representar el valor que se está devolviendo.

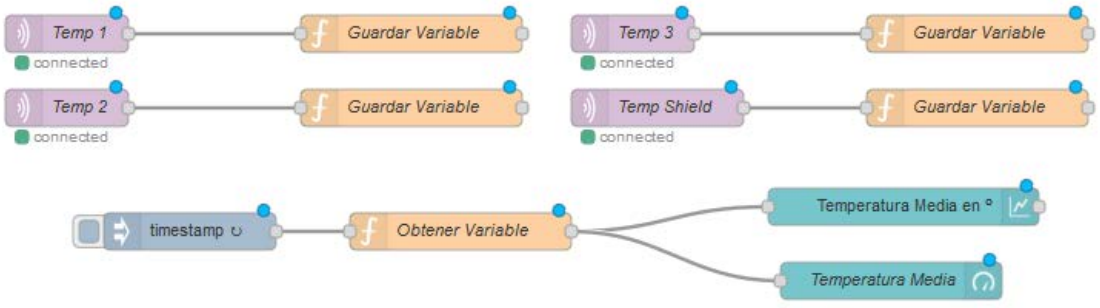


Figura 53: Nodos usados para el Panel de Control

Finalmente obtendríamos nuestro Panel de Control con los cuatro grupos de sensores y la media calculada de los datos de los sensores en tiempo real.

Se puede acceder al Panel de Control de forma local a través del siguiente enlace http://localhost:1880/ui/#!/0

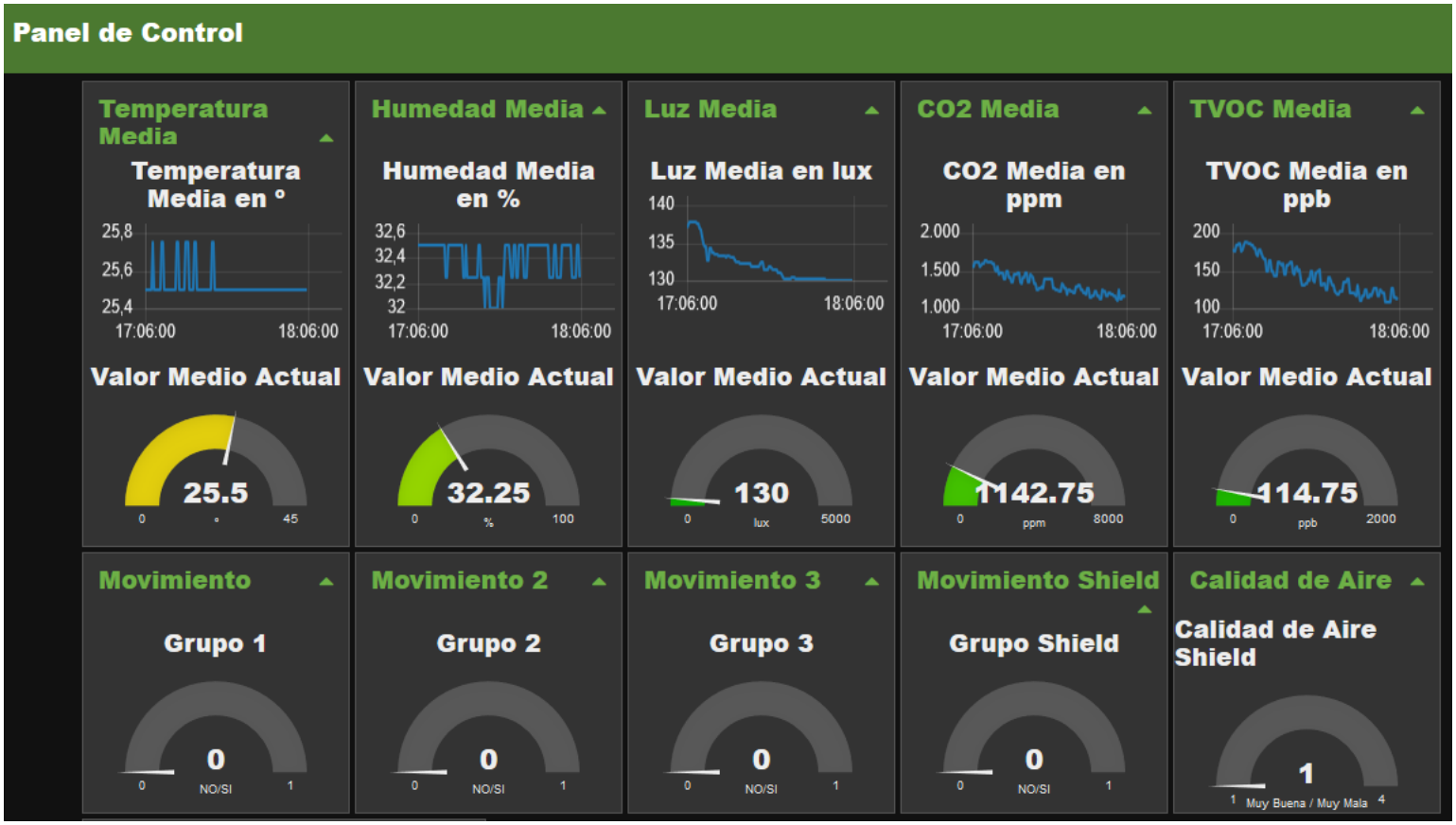


Figura 54: Vista del Panel de Control creado con Node Red

En la parte inferior se ha insertado un gráfico para ver de manera rápida si se ha detectado un movimiento. Un “0” equivaldría a ningún Movimiento detectado y un “1” significa que si hay movimiento.

Figura 55: Ningún Movimiento

Figura 56: Movimiento Detectado

5.3.5 NODE RED – AULA 352

En esta parte del cliente se ha tratado de reconstruir el aula 352 de la facultad de Ingeniería de la UJA, lugar donde esta colocados los sensores.

Los nodos y pasos necesarios serían los siguientes.

1. Lo primero ha sido reconstruir el aula 352. Para ello se ha usado el programa "SmartDraw". *SmartDraw* es una herramienta de diagrama utilizada para hacer diagramas de flujo, organigramas, mapas mentales, diagramas de proyectos y otras imágenes comerciales.

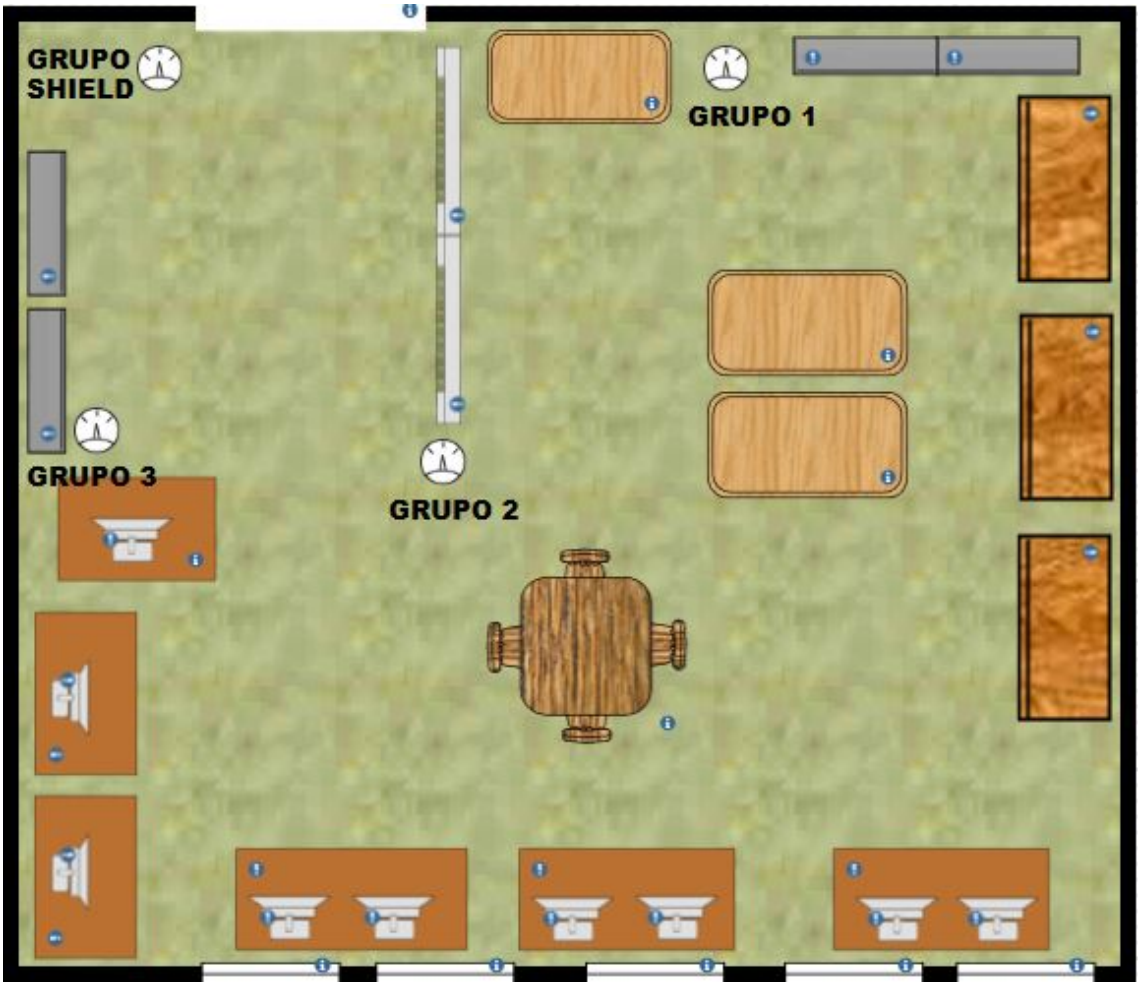


Figura 57: Plano del Aula 352

Una vez creado el plano virtual del aula se han marcado las posiciones exactas de los cuatro grupos de sensores.

Para poder visualizar los datos en tiempo real y el plano se necesitan dos nodos principales.

1. Un nodo de tipo “function” para llamar todas las variables globales de un grupo y crear una salida de todos los datos como un solo Output.

```
var msg1=global.get("MS22");
var msg2=global.get("MS24");
var msg3=global.get("MS26");
var msg4=global.get("MS28");
var msg5=global.get("MS30");
var msg6=global.get("MS32");



msg.payload = "Temperatura: "+msg1+"º"+", "+" Humedad: "+msg2+"%"+", "+" Luz: "+msg3+"lux , "
+msg6+", "+" CO2: "+msg4+" ppm, TVOC: "+msg5+" ppb ";

return  msg;
```

Figura 58: Función en Node Red para llamar variables y crear un Output

2. Un segundo nodo de tipo “Template” del grupo dashboard. Este nodo se usa para llamar al plano, establecer la altura y el ancho y para colocar la salida de datos en el sitio elegido.

```
<div>
<img src="http://localhost:1880/ui/Floor5.png" height="700" width="700"><br/>


  <div ng-repeat="el in msg.payload" style="left: {{20}}px; position: absolute; top: {{60}}px;">

        <br />
        <div style="color: {{el.color}}; font-size: 10px; padding-top: 2px; text-align: center;">
            {{el.text}}</div>
                <div style="color: {{el.color}}; font-size: 25px; padding-top: 2px; text-align: center;">
            {{msg.payload}}</div>

        </div>
</div>
```

Figura 59: Código Node Red para crear el plano del aula en el visor web

Finalmente le añadimos un “inject” que se ejecuta cada 5 segundos y el flujo entero quedaría de la siguiente forma.

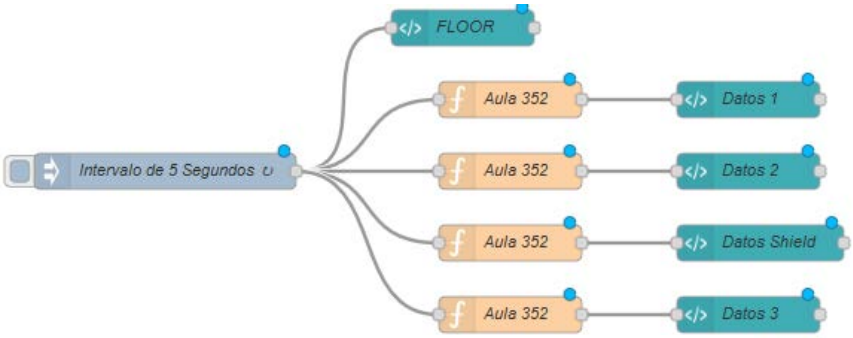


Figura 60: Nodos usados para el Aula

El producto final sería el plano del aula 352 junto con los datos de todos los sensores que se actualizan cada 5 segundos.

Figura 61: Visualización del plano del aula con los datos de los sensores en el visor web

Capítulo 6 RESULTADOS

Para hacer un análisis de los datos captados por estos sensores se ha elegido un periodo representativo de cuatro días que transcurre entre las fechas del 15.10.2019 y el 18.10.2019. Se trata de un martes, miércoles, jueves y viernes, siendo el viernes de especial interés, ya que al ser día festivo, el espacio universitario estuvo vacío. Esto nos dio la opción, por ejemplo, de comprobar si alguno de los sensores de movimiento capto algún movimiento durante un día no lectivo.

La cantidad de datos captados y almacenados durante estas fechas suman un total de 28922.

Algunos de los valores tomados por los sensores, como por ejemplo los de Temperatura, se han comparado con datos climáticos de la plataforma libre, "Open Data" de AEMET. Esta plataforma nos da la opción de elegir un periodo de tiempo y una estación meteorológica para visualizar los datos climáticos más básicos que se pueden conseguir de forma gratuita.

La estación meteorológica más cercana a la universidad con datos disponibles para dichas fechas es la estación "Jaén" con los siguientes datos:

Provincia	JAEN
Indicativo	5270B
Altitud	580
Latitud	374639N
Longitud	034832W
Inicio Datos	15.10.2019
Fin Datos	18.10.2019

La distancia en línea aérea a la facultad de Ingeniera de la Universidad de Jaén es de 2,94 km.

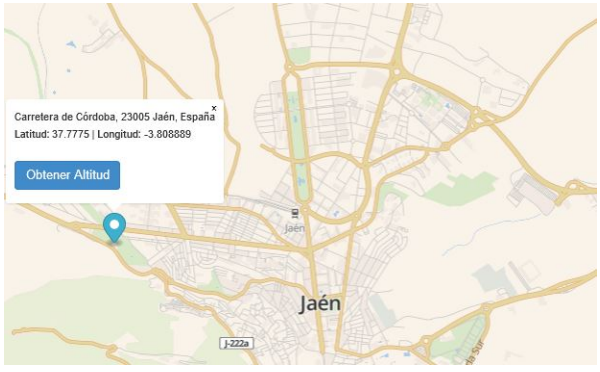


Figura 62: Localización de la estación Meteorológica

Figura 63: Distancia de la estación Meteorológica a la UJA

6.1 TEMPERATURA

Para analizar los datos de temperatura adecuadamente se han exportado los valores captados de la base de datos a Excel, para poder crear un gráfico para cada uno de los 4 sensores de temperatura. Posteriormente se han comparado estos valores con datos climáticos de la plataforma libre, "Open Data" de AEMET.

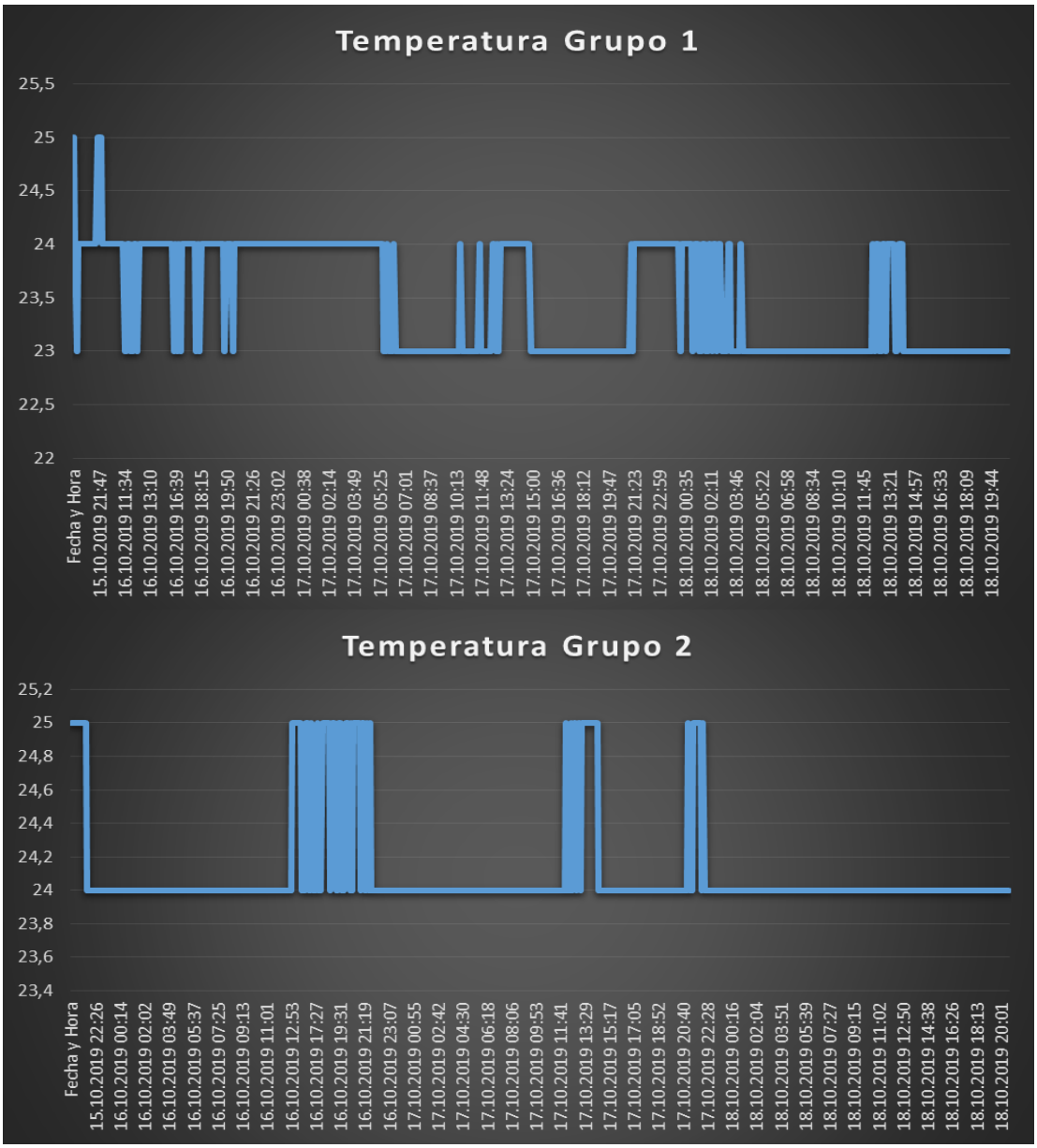


Figura 64: Grafico de Temperatura del Grupo 1

Figura 65: Grafico de Temperatura del Grupo 2

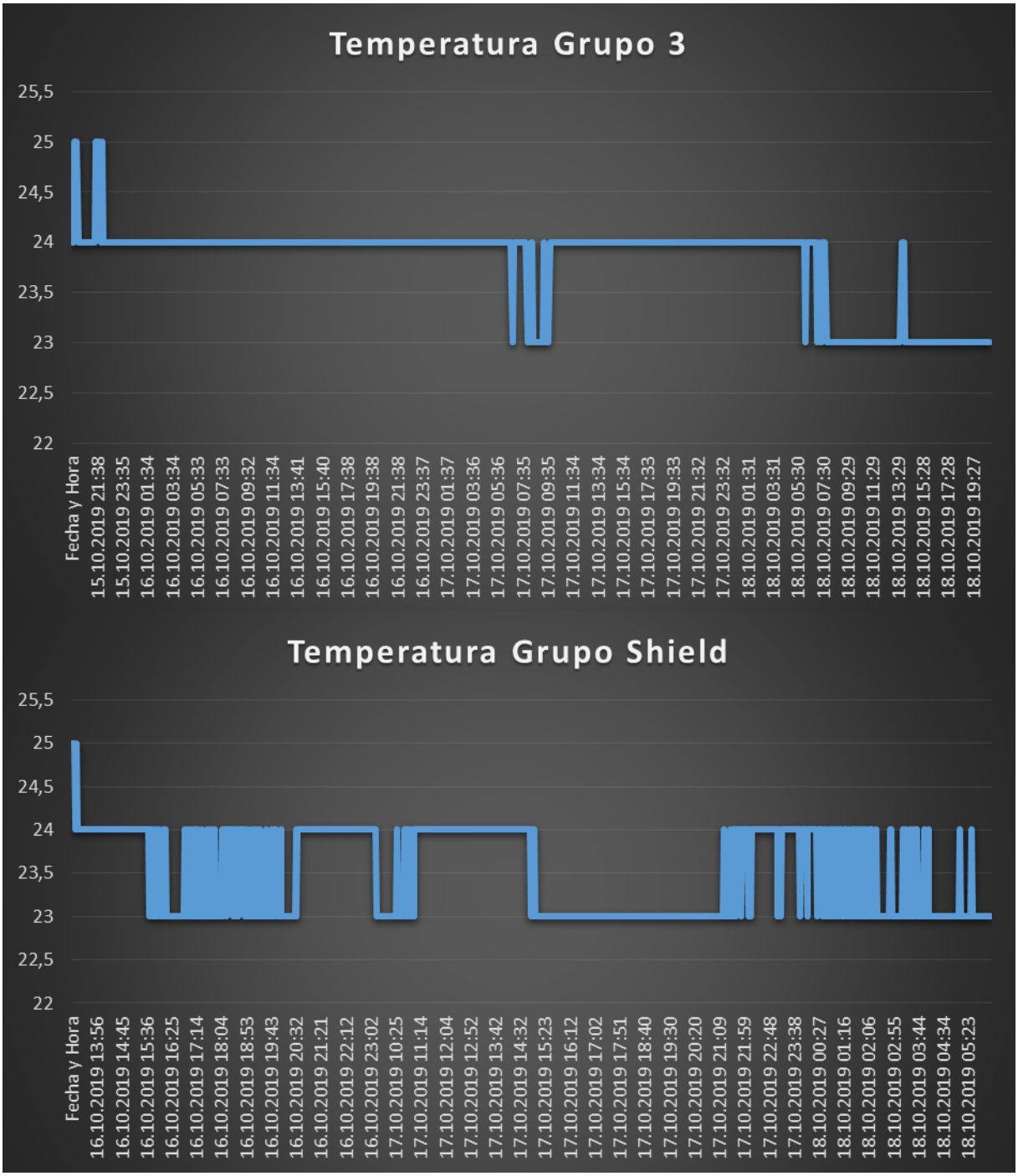


Figura 66: Grafico de Temperatura del Grupo 3

Figura 67: Grafico de Temperatura del Grupo Shield

Como se puede comprobar en los gráficos todos los sensores han tomado los mismos valores para este periodo de tiempo. Todos los valores se sitúan entre 23° y 25°.
Únicamente el sensor del grupo 2 tiene un valor mínimo de 24°. Esto se debe probablemente a que este grupo de sensores está colocado prácticamente debajo del aire acondicionado del cual sale aire caliente constantemente. Se trata de un aula con las ventanas cerradas las 24 horas del día por lo que el calor se mantiene bastante

bien. Por este motivo la variación de temperatura por las noches es mínima con una bajada de unos 1° o 2°.

Como se puede ver en el grafico inferior, la temperatura media de Jaén para estos días varía entre 13,6° y 18°. En cambio en el aula hay una temperatura media de 24°. La dificultad a la hora de querer comparar estos datos con los datos captados por AEMET tiene dos razones fundamentales. La primera es que al tratarse de una habitación de interior partimos de una temperatura más elevada. Si además tenemos en cuenta que las ventanas y la puerta están prácticamente cerradas 24 horas, entonces el intercambio de temperatura es mínimo.
En segundo lugar, el aula dispone de aire acondicionado que los días 15,16 y 17 estaba encendido a una temperatura media de 21°-23°.

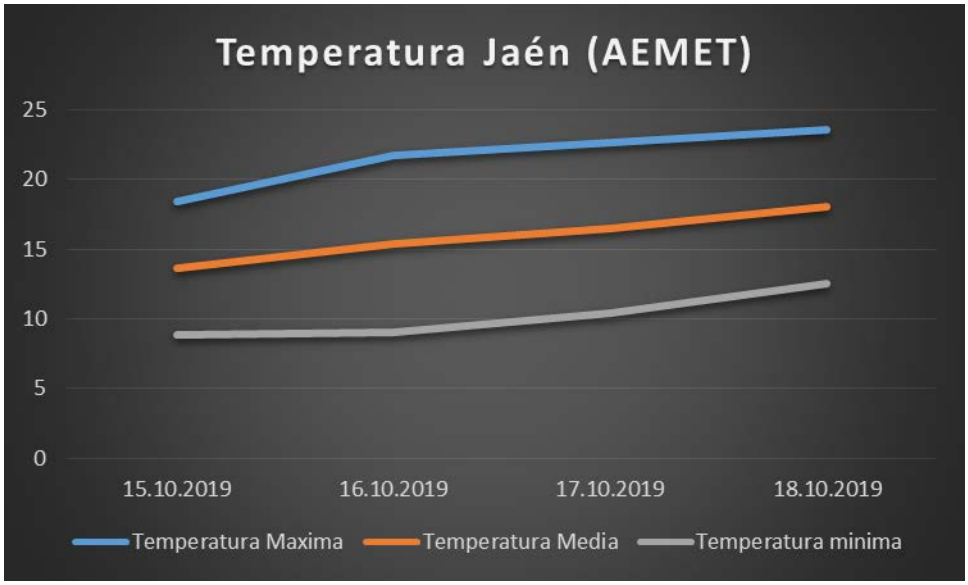


Figura 68: Grafico de Temperatura con datos de AEMET

6.2 HUMEDAD

Para analizar los datos de humedad adecuadamente se han exportado los valores captados de la base de datos a Excel, para poder crear un gráfico para cada uno de los 4 sensores de humedad. Igual que en el caso de la temperatura se ha querido comparar estos valores con datos climáticos de la plataforma libre, “Open Data” de AEMET, pero al no existir los datos de humedad en esa base de datos libre, no se ha podido realizar esta comparación.

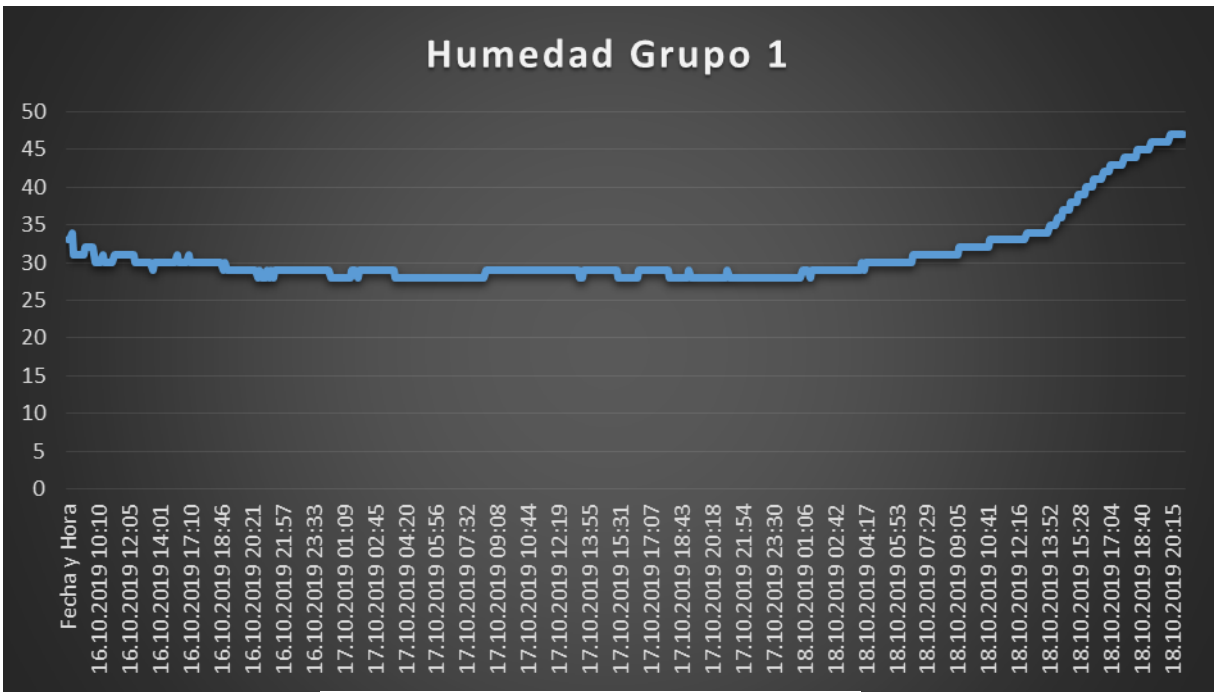


Figura 69: Grafico de Humedad del Grupo 1

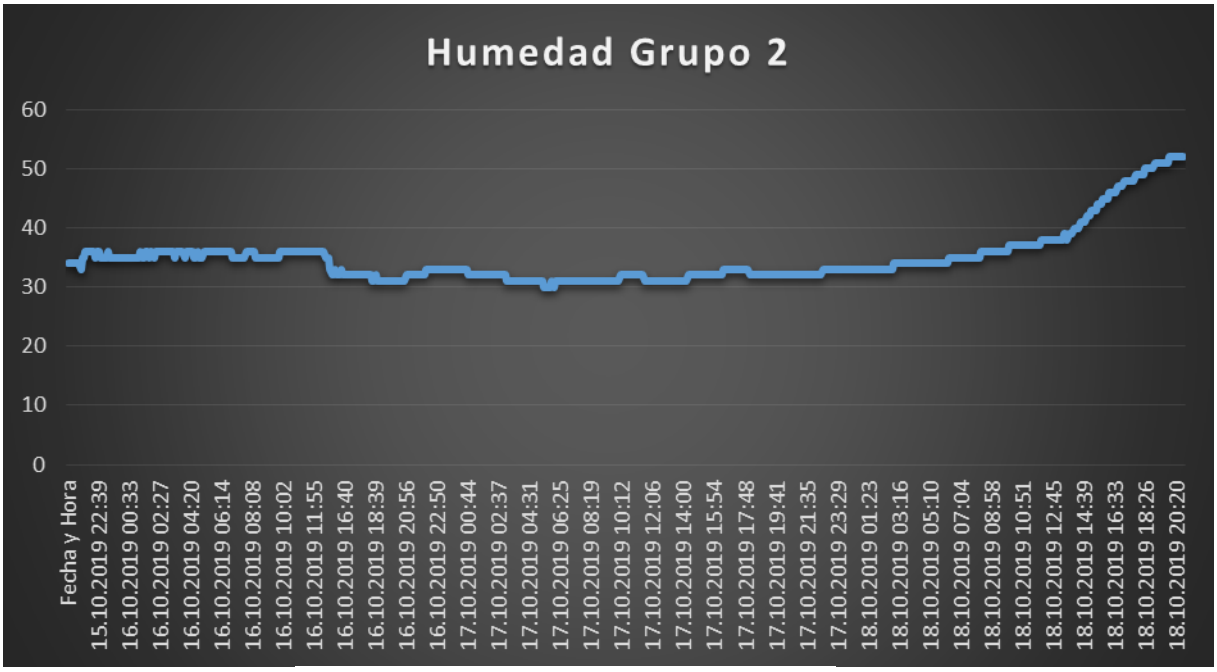


Figura 70: Grafico de Humedad del Grupo 2

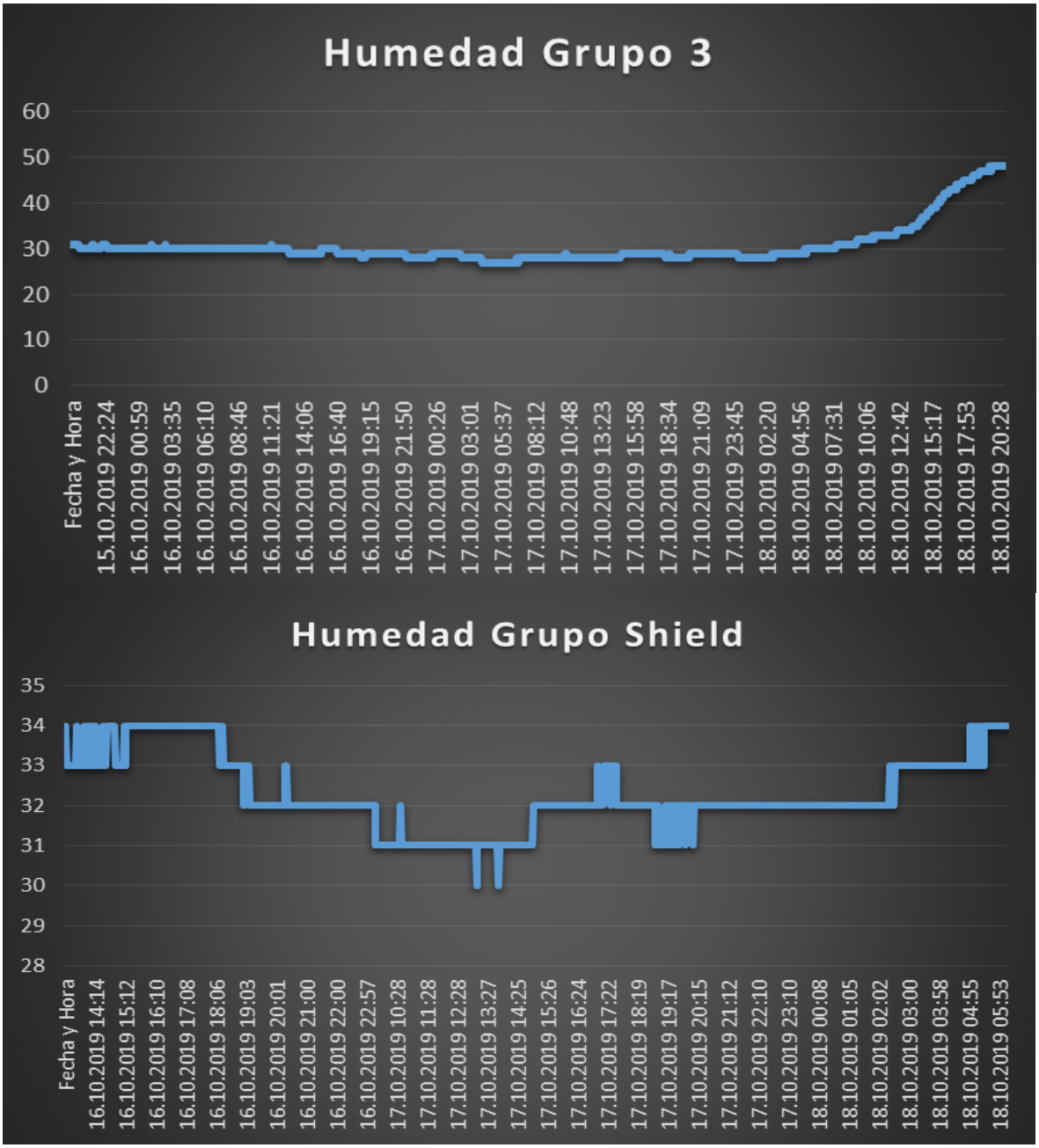


Figura 71: Grafico de Humedad del Grupo 3

Figura 72: Grafico de Humedad del Grupo Shield

Como se puede observar en los gráficos, los 4 sensores han medido un valor medio aproximado de 35% de humedad. Un valor que se ha mantenido durante los tres primeros días y que ha sufrido un notable ascenso el cuatro día. Ha empezado a subir a partir de las 15:00 hasta llegar a un 50% de humedad. Este valor se encuentra al límite ya que el porcentaje saludable recomendado para una habitación de este tipo oscila entre el 40% y el 60% de humedad. El valor bajo recogido se debe al aire acondicionado, que estuvo encendido todos los días salvo el viernes 18, pues era festivo. Para confirmar que se debe al aire acondicionado se puede observar como empieza a subir la humedad el día en el que está cerrado la universidad.

6.3 LUZ

Para analizar los datos de luz adecuadamente se han exportado los valores captados de la base de datos a Excel, para poder crear un gráfico para cada uno de los 4 sensores de luz. La unidad de medida de estos datos es el lux.

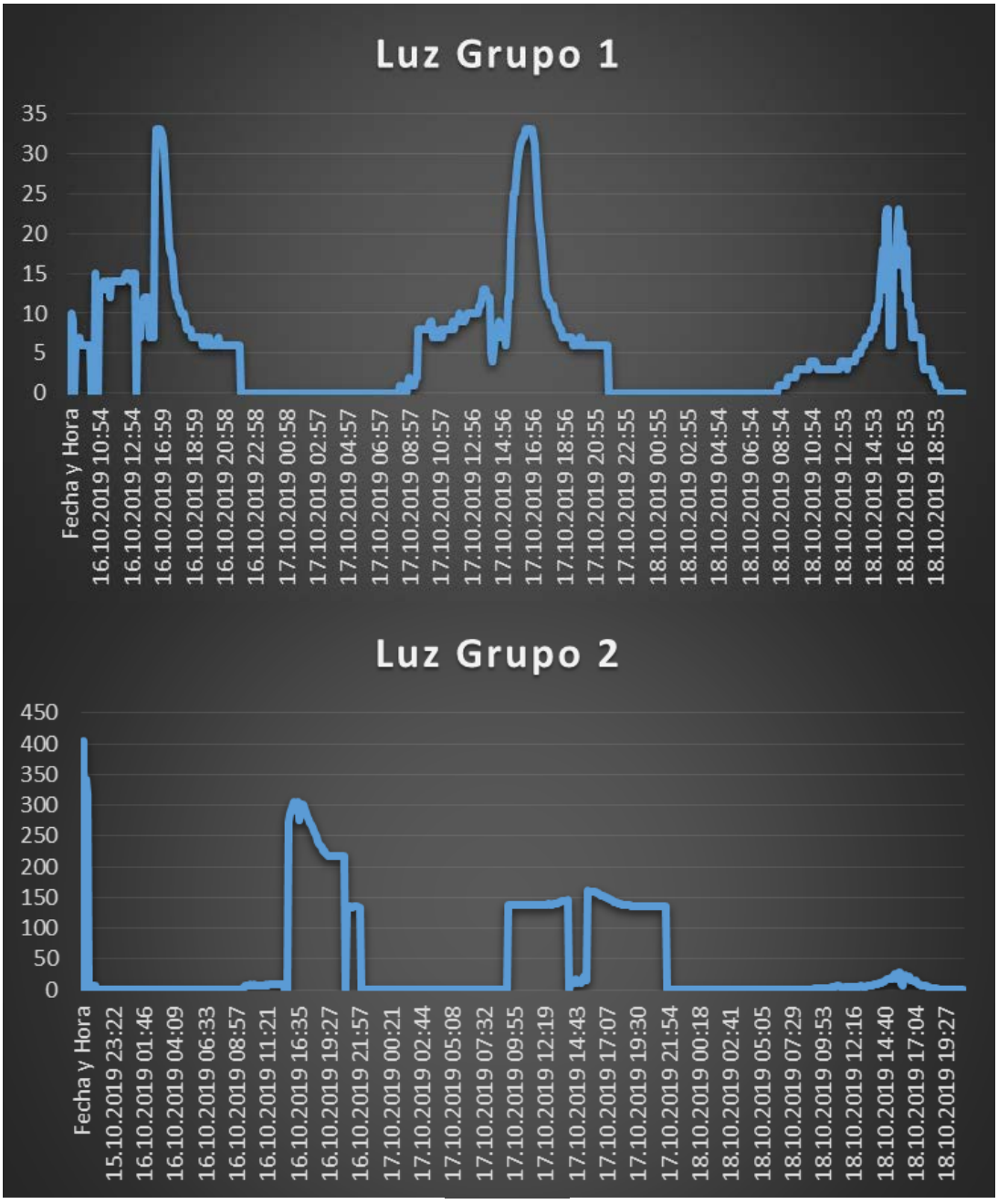


Figura 73: Grafico de Luz del Grupo 1

Figura 74: Grafico de Luz del Grupo 2

Figura 75: Grafico de Luz del Grupo 3

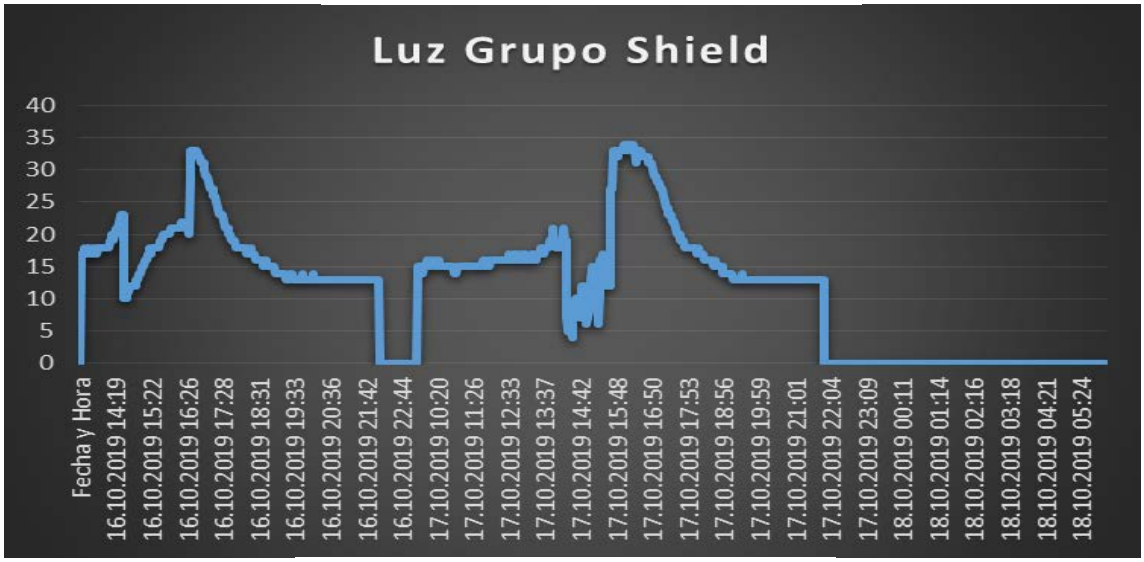


Figura 76: Grafico de Luz del Grupo Shield

Estos datos nos hablan de la cantidad de luz que han medido los sensores a lo largo de los cuatro días. En los cuatro sensores se puede observar muy bien que aproximadamente a las 9 de la mañana se ha encendido la luz del aula y sobre las 22:00 se ha apagado, siendo esa mi hora de llegada y salida esos días, del aula. También se puede apreciar como el día 18 (festivo) la luz es mucho más baja que los demás días ya que nadie ha encendido las luces en el aula. El grupo 1 es el único que ha medido unos valores algo más altos el día 18 (festivo) por la tarde, pudiéndose justificar esto con su localización debajo de una ventana que da al pasillo, dando lugar así a que le caiga más luz que a los demás sensores, que tienen una posición más central en el aula.

6.4 MOVIMIENTO

Para analizar los datos de movimiento adecuadamente se han exportado los valores captados de la base de datos a Excel, para poder crear un gráfico para cada uno de los 4 sensores de movimiento. En el grafico solo se verán representados dos valores numéricos: un “1” significa movimiento detectado y un “0” significa que **no** se ha detectado ningún movimiento.

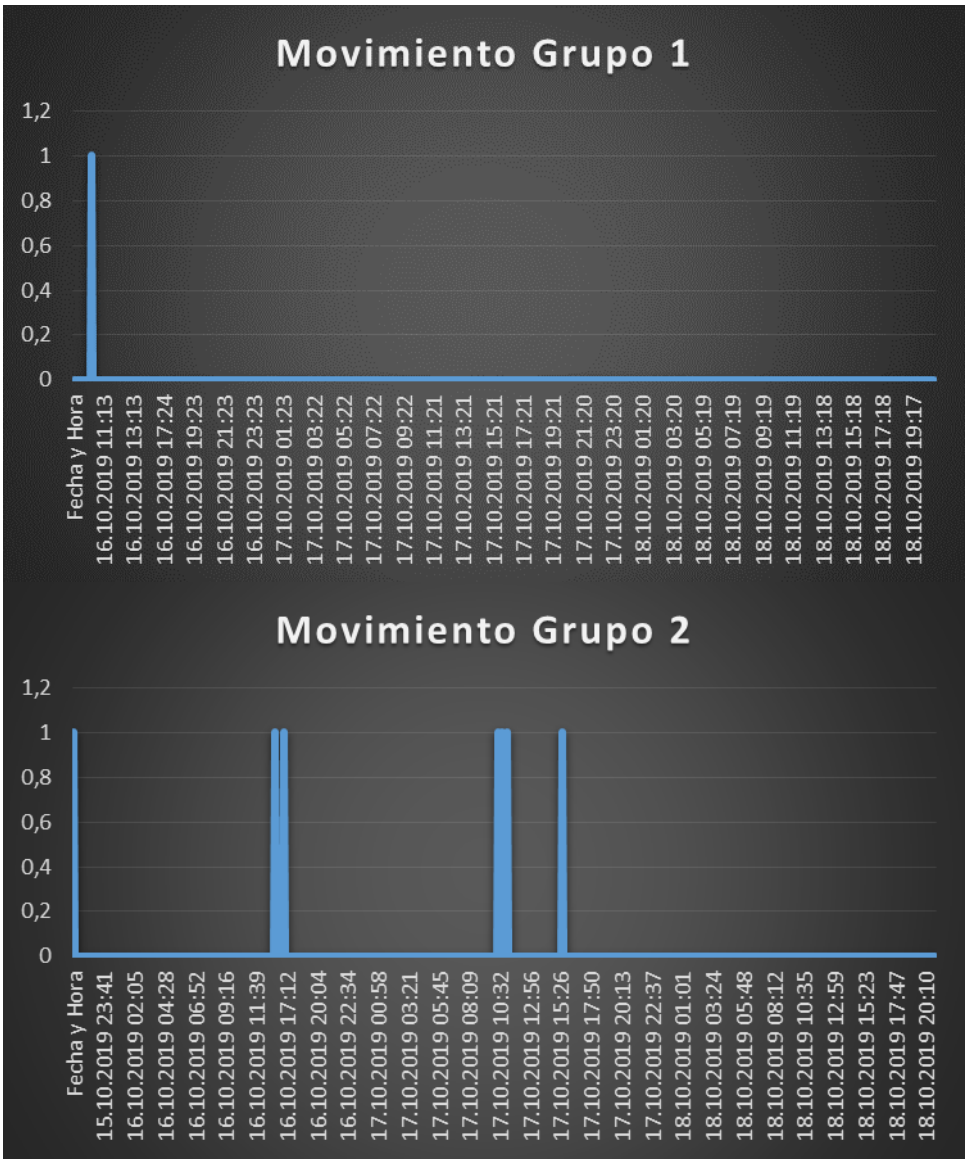


Figura 77: Grafico de Movimiento del Grupo 1

Figura 78: Grafico de Movimiento del Grupo 2

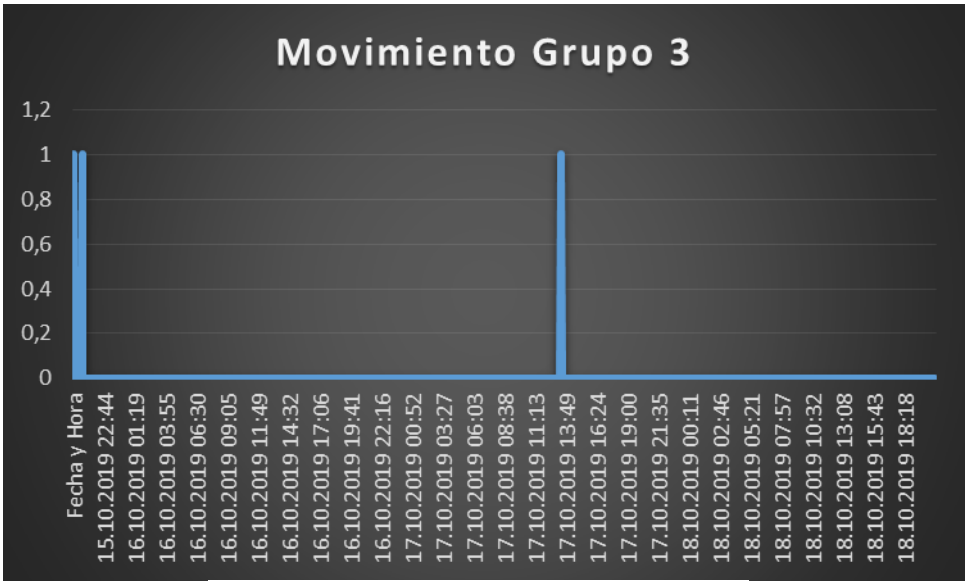


Figura 79: Grafico de Movimiento del Grupo 3

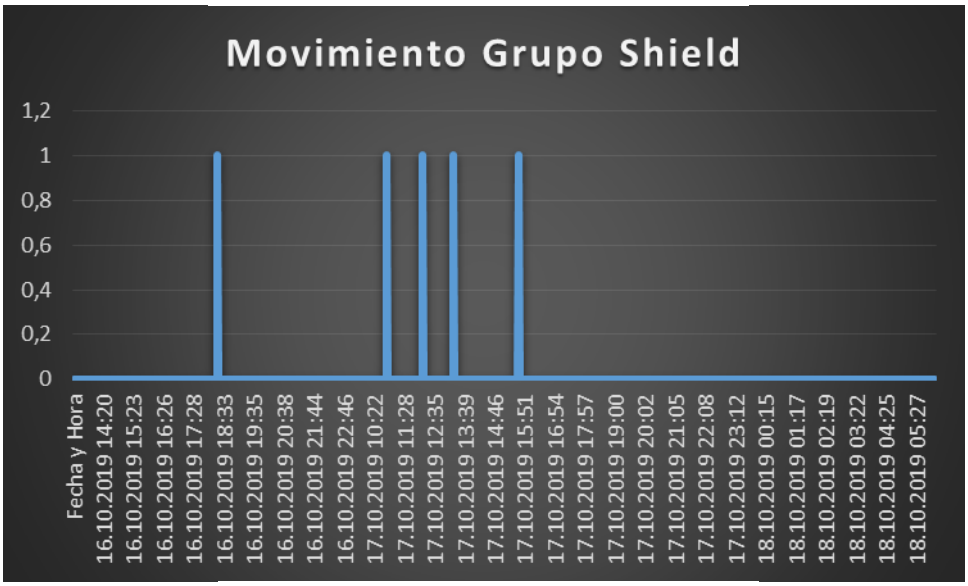


Figura 80: Grafico de Movimiento del Grupo Shield

Lo primero que se puede destacar de estos resultados es que se ha detectado poco movimiento. Esto tiene dos razones principales.

La primera simplemente es que en general yo era la única persona que trabajaba en ese aula y al estar prácticamente todo el día sentado, se ha detectado poco movimiento.

La segunda razón es el código de Arduino que se ha utilizado para captar los movimientos. Se ha incluido la función para captar el movimiento en el loop que se ha

usado para los demás sensores que como ya se ha explicado en el apartado de Metodología tiene un *delay* de 6 segundos entre cada sensor. Por lo tanto se tarda unos 40 segundos aproximados en comprobar si hay algún movimiento en ese momento. Si se ha producido un movimiento en ese periodo de tiempo no se ha guardado como tal. Por lo tanto el movimiento tenía que coincidir más o menos con el momento en el que se estaba ejecutando esa parte del código.

El Grupo Shield es el que estaba colocado justo al lado de la puerta por lo que es el sensor que más movimientos ha detectado. El sensor del grupo "2" también se encuentra prácticamente en el centro del aula por lo que ha detectado la mayoría de movimientos. El que prácticamente nunca ha detectado ningún movimiento es el sensor del grupo 1 ya que se encuentra en una parte del aula por la que nunca me he movido y además tiene un panel enfrente lo que lo limita en gran medida a la hora de captar movimiento.

6.5 CO2

Para analizar los datos de CO2 adecuadamente se han exportado los valores captados de la base de datos a Excel, para poder crear un gráfico para cada uno de los 4 sensores de CO2. La unidad de medida del CO2 es en ppm(partes por millón).

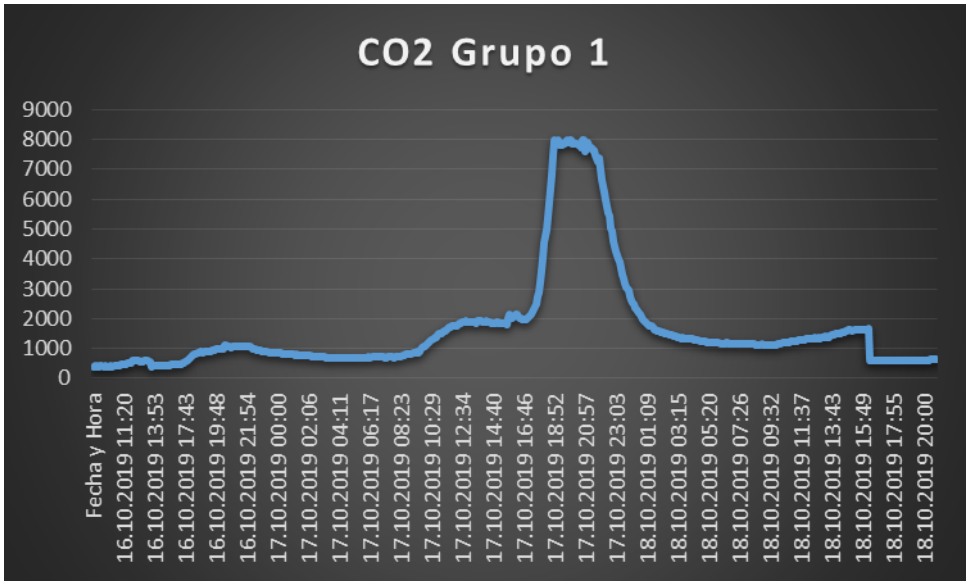


Figura 81: Grafico de CO2 del Grupo 1

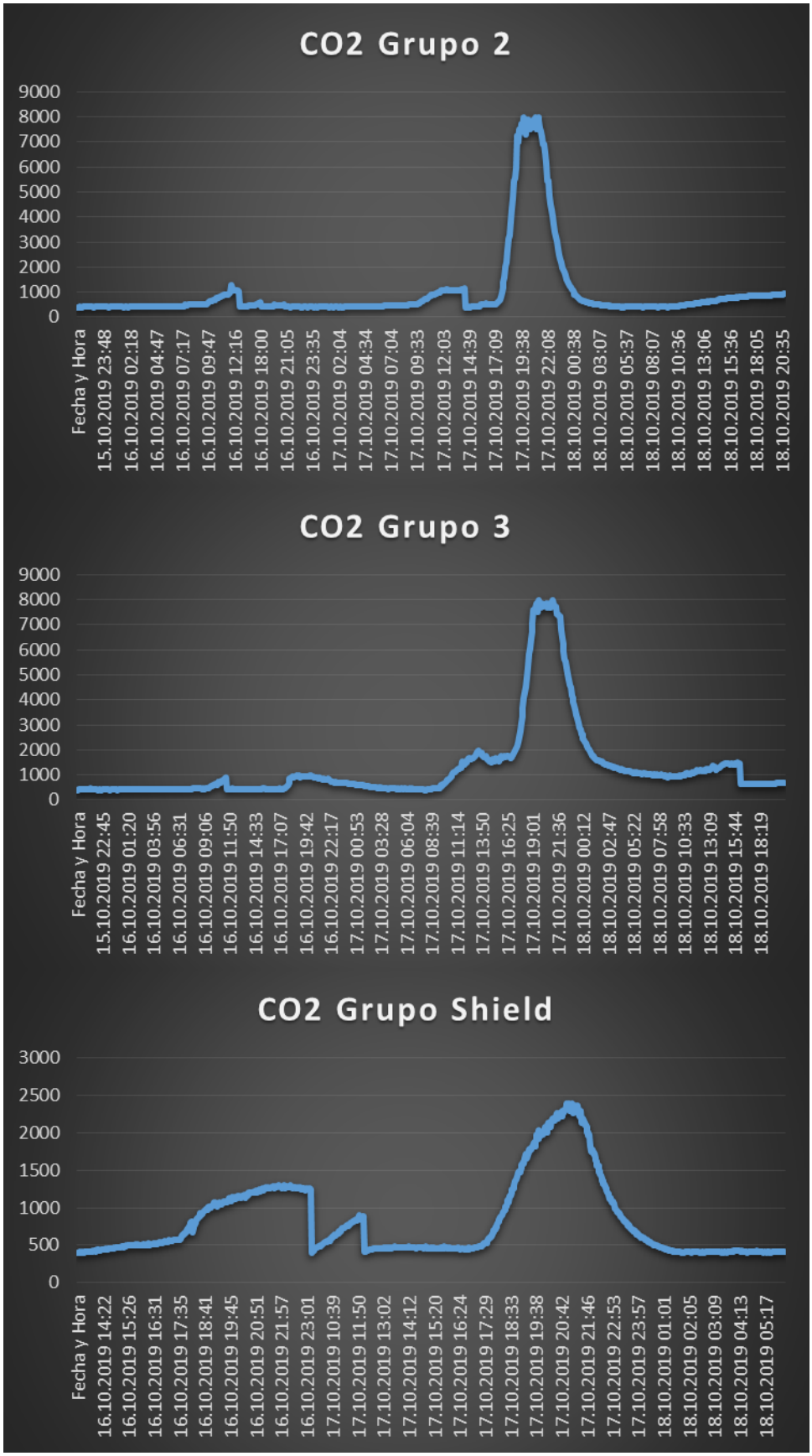


Figura 82: Grafico de CO2 del Grupo 2

Figura 83: Grafico de CO2 del Grupo 3

Figura 84: Grafico de CO2 del Grupo Shield

Lo primero que llama la atención de estos datos son los valores máximos que alcanzan los sensores, siendo hasta 8000 ppm en algunos casos. Un valor normal, recomendable y saludable para una habitación suele ser de entre unos 400 ppm y 800 ppm. La razón principal para esto es la sensibilidad de este sensor al CO2. Al tratarse de un sensor con un coste aproximado de unos 0,40¢, no tiene comparación con un sensor industrial que podría dar resultados más fiables y realistas.

Aun así, al igual que en los datos de luz, en estos valores se pueden ver los días y las horas en las que hubo personas en el aula. Los valores suben considerablemente por la tarde a partir de las 16:00 hasta las 22:00 cuando empieza a bajar de nuevo. Esto coincide con mis horas en el aula. Por la noche se mantiene un valor bajo constante de unos 500ppm. El jueves 17 por la tarde es cuando se ha producido ese pico alto en los gráficos y coincide con la presencia de otro compañero durante toda la tarde en el aula.

6.6 TVOC

El TVOC del inglés "Total volatile organic compounds". El TVOC es una agrupación de una amplia gama de compuestos químicos orgánicos para simplificar la presentación de informes cuando estos están presentes en el ambiente del aire o las emisiones.

Las personas pasan el 90% de su tiempo en interiores donde las concentraciones de contaminantes gaseosos son significativamente más altas que en exteriores. El uso generalizado de nuevos productos y materiales de construcción, así como el aislamiento mejorado para la eficiencia energética, ha resultado en un aumento de las concentraciones de compuestos orgánicos volátiles (COV). Estos compuestos orgánicos volátiles se originan principalmente de pinturas y solventes, alfombras y muebles, y agentes de limpieza, y también son emitidos por humanos. Los niveles elevados de VOC pueden tener un impacto negativo en el bienestar, la comodidad y las habilidades cognitivas. La exposición a altos niveles de COV puede evitarse o reducirse significativamente mediante la ventilación regular, la purificación del aire y la eliminación de fuentes fuertes de COV. El concepto Total VOC (TVOC) se ha establecido como un método práctico y rentable para inspeccionar los ambientes interiores en busca de contaminación.

Para analizar los datos de TVOC adecuadamente se han exportado los valores captados de la base de datos a Excel, para poder crear un gráfico para cada uno de los 4 sensores de TVOC. La unidad de medida del TVOC es en ppb (partes por billón).

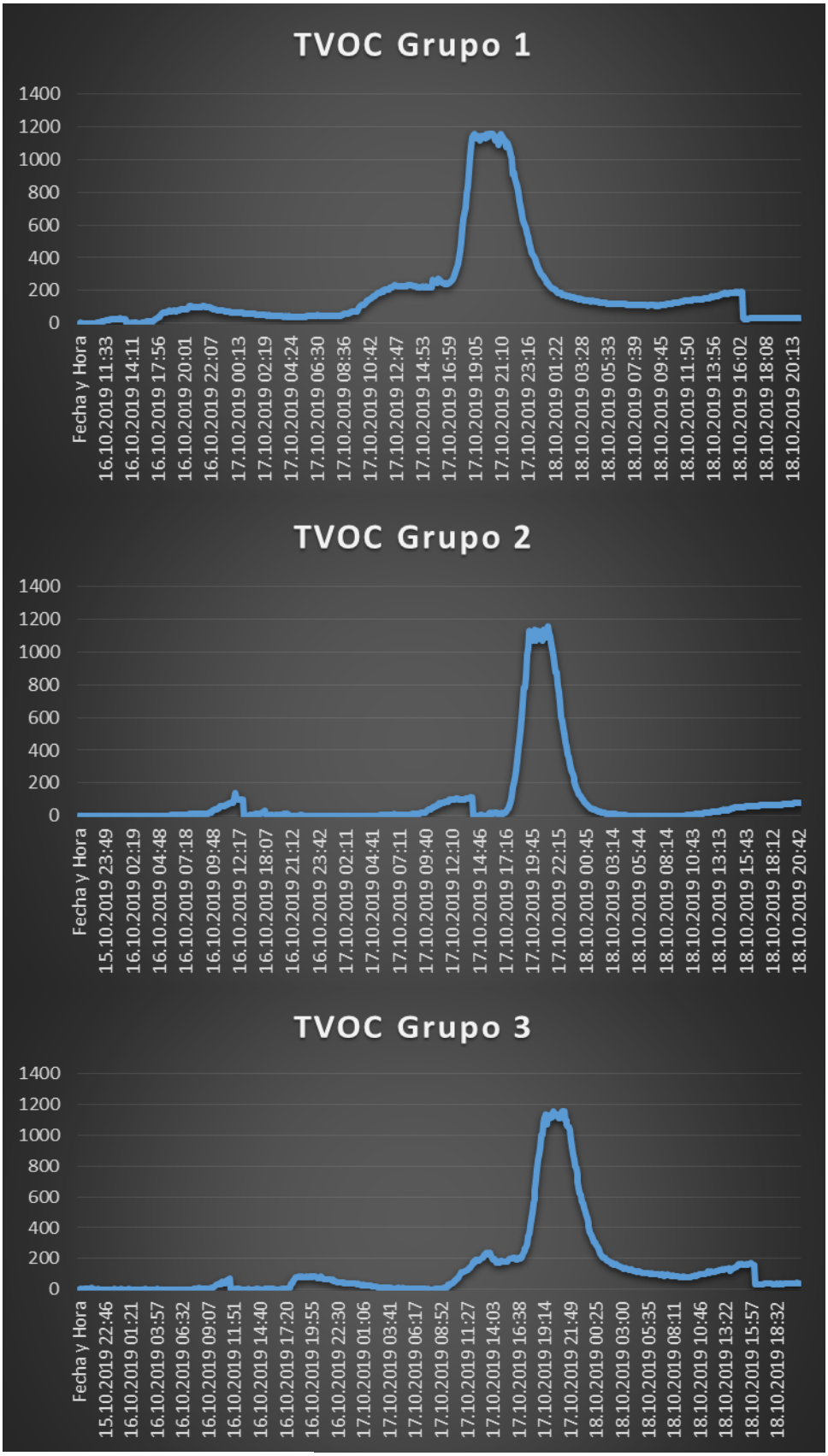


Figura 85: Grafico de TVOC del Grupo 1

Figura 86: Grafico de TVOC del Grupo 2

Figura 87: Grafico de TVOC del Grupo 3

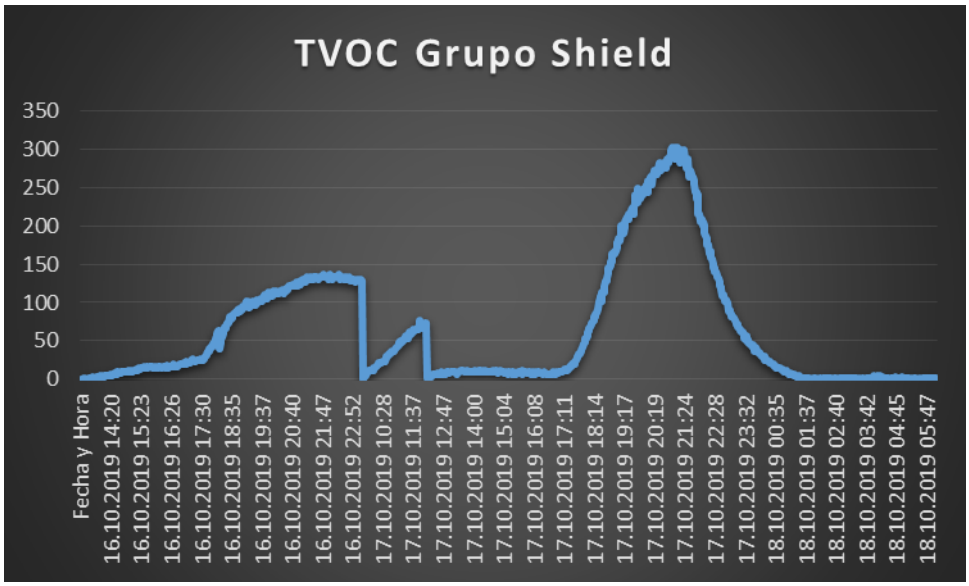


Figura 88: Grafico de TVOC del Grupo Shield

A igual que los datos de CO2, estos datos de TVOC son medidos por el sensor de gases CCS811. Si comparamos los gráficos del CO2 con los gráficos de TVOC vemos que estos gases van mano en mano. Cuando se produce una subida o bajada en uno, en el otro también se da. Lo más destacable es la subida del jueves por la tarde que se puede observar de la misma forma en los datos TVOC. Para el análisis de los datos se ha usado una tabla con niveles críticos de TVOC creada por la agencia federal alemana de medio ambiente.

Level	Hygienic Rating	Recommendation	Exposure Limit	TVOC [ppb]
5 Unhealty	Situation not acceptable	Use only if unavoidable / Intense ventilation necessary	hours	2200 – 5500
4 Poor	Major objections	Intensified ventilation / airing necessary Search for sources	< 1 month	660 – 2200
3 Moderate	Some objections	Intensified ventilation / airing recommended Search for sources	< 12 months	220 – 660
2 Good	No relevant objections	Ventilation / airing recommended	no limit	65 – 220
1 Excellent	No objections	Target value	no limit	0 – 65

Figura 89: Tabla con Niveles de TVOC

Los valores captados por los sensores se sitúan entre los 0ppb y 200ppb lo cual es considerado un nivel moderado. Tal y como se puede apreciar en el gráfico, el jueves 17, ha tenido lugar una subida de los niveles de TVOC, alcanzando un valor moderado. En estas ocasiones, la agencia federal alemana de medio ambiente recomienda descubrir las fuentes de dichos valores y ventilar la habitación.

6.7 CALIDAD DE AIRE

Para analizar los datos de calidad de aire adecuadamente se han exportado los valores captados de la base de datos a Excel, para poder crear un gráfico para el único sensor de este tipo que se ha usado. Como ya se ha explicado en el apartado 4 de metodología este sensor tiene 4 valores para medir la calidad de aire.

- Un valor de 4 es el grado de contaminación más alto que mide el sensor de calidad de aire. 4 = Alta contaminación! Señal de fuerza activa!
- Un valor de 3 es el segundo grado de contaminación más alto que mide el sensor de calidad de aire. 3 = Alta contaminación!
- Un valor de 2 es el segundo grado de contaminación más bajo que mide el sensor de calidad de aire. 2 = Baja contaminación!
- Un valor de 1 es el grado de contaminación más bajo que mide el sensor de calidad de aire. 1 = Aire Fresco

Figura 90: Grafico de Calidad de Aire del Grupo Shield

Como se puede ver en el gráfico, todos los días y a todas horas del periodo de estudio se ha medido un valor de "1" y por lo tanto una calidad de "Aire Fresco", el mejor nivel de calidad que mide este sensor. Se han hecho pruebas para manipular el aire alrededor del sensor para comprobar que sea capaz de medir otros niveles de calidad y

efectivamente se consiguió también un nivel “2” equivalente a una baja contaminación.

Después de los resultados de TVOC, esto nos vuelve a confirmar que el aula en el que se han realizado las pruebas, a pesar de tener las ventanas cerradas 24 horas cerradas y no tener apenas ventilación, dispone de una buena calidad de Aire.

Capítulo 7 CONCLUSIONES

Este trabajo fin de master tenía como objetivo el desarrollo de un geoportal para la monitorización de espacios usando redes de sensores y arquitecturas orientadas a servicios.
Se ha mostrado como montar un servicio, cosa imprescindible en el IOT (Internet de las cosas), partiendo de un software como Arduino para programar los sensores.
Se ha usado una herramientas como MQTT que se caracteriza por su simple funcionamiento a la hora de recibir y enviar datos. Se ha visto el potencial de una plataforma como lo es Node-Red que nos permite recibir datos, reenviarlos a una base de datos, crear clientes web o paneles de control, y solo requiere unos básicos conocimientos de programación.
Se han usado una serie de sensores que en algunos casos vinieron en dos piezas y fue necesario soldarlos. En general han tenido un coste muy bajo pero aun así han dado buenos y fiables resultados.
He descubierto que a la hora de captar datos con un sensor durante varios días es inevitable el uso de un cargador ya que las baterías se agotan al cabo de unas horas.

Otra curiosidad a destacar en este trabajo es la importancia del cableado. Han surgido algunos problemas de conexión, principalmente con el sensor de gases (CCS811), que no mandaba los datos captados al microcontrolador si el cableado usado era demasiado largo. Al cambiar y probar con cables más cortos estando este prácticamente estirado, se han resuelto estos problemas.

En cuanto a los sensores no podemos olvidar que en el mercado hay sensores mucho más potentes y con una mayor precisión que los usados en este trabajo y que si el estudio que se quiera realizar es de tipo profesional probablemente se deberían usar otros sensores.

A la hora de analizar los resultados, la disponibilidad de una mayor variedad de datos climáticos (como por ejemplo, de humedad) para el periodo de estudio hubiera sido interesante, dando de esta manera opción a un análisis más profundo.

Viendo los resultados de Co2 tenía mis dudas en cuanto a la fiabilidad de sensor pero viendo que los datos medidos de TVOC por el mismo sensor se sitúan dentro de los umbrales que ha definido la agencia federal alemana de medio ambiente, me han quitado de dudas.

Ha sido un trabajo interesante donde he podido poner a prueba los conocimientos obtenidos en asignaturas del master como servicios web, redes de sensores y aplicaciones geomaticas.

Capítulo 8 BIBLIOGRAFIA

AEMET. (s.f.). *datosclima*. Obtenido de Base de datos Meteorológica: https://datosclima.es/

Arduino. (s.f.). Obtenido de https://www.arduino.cc/

Company, S. (2019). Obtenido de Indoor Air Quality and Volatile Organic Compounds: https://www.repcomsrl.com/wp-content/uploads/2017/06/Environmental_Sensing_VOC_Product_Brochure_EN.pdf

Cope, S. (s.f.). *Steves Internet Guide*. Obtenido de http://www.steves-internet-guide.com/

IGN. (s.f.). Obtenido de Iberpix: https://www.ign.es/iberpix2/visor/

Khatri, P. (18 de Abril de 2018). *Circuitdigest*. Obtenido de https://circuitdigest.com/microcontroller-projects/tvoc-co2-measurement-using-aduino-and-ccs811-air-quality-sensor

Llamas, L. (24 de Julio de 2015). *Luisllama*. Obtenido de https://www.luisllamas.es/detector-de-movimiento-con-arduino-y-sensor-pir/

meteoclimatic. (s.f.). Obtenido de https://www.meteoclimatic.net/index

MQTT. (s.f.). Obtenido de http://mqtt.org/

MySQL. (s.f.). Obtenido de https://www.mysql.com/

Node Red Guide. (s.f.). Obtenido de http://noderedguide.com/

Node-Red. (s.f.). Obtenido de https://nodered.org/

seeedstudio. (s.f.). Obtenido de http://wiki.seeedstudio.com/Grove-Air_Quality_Sensor_v1.3/

Capítulo 9 ANEXOS

9.1 CODIGOS ARDUINO

9.1.1 GRUPO 1

```
#include <ArduinoMqttClient.h>
#include <WiFiNINA.h>
#include <DHT.h>
#include <Wire.h>
#include <BH1750.h>
#include "Adafruit_CCS811.h"
#define SIGNAL_PIN 2
#define DHTPIN 5
#define DHTTYPE DHT11

Adafruit_CCS811 ccs;

char ssid[] = "Cartografia";
char pass[] = "C@rtoU119";

BH1750 lightMeter;
DHT dht(DHTPIN, DHTTYPE);
WiFiClient wifiClient;
MqttClient mqttClient(wifiClient);

int port = 1883;
const char* topicName = "/Sensor/Grupo1/Temperatura";
const char* topicName2 = "/Sensor/Grupo1/Humedad";
const char* topicName3 = "/Sensor/Grupo1/Luz";
const char* topicName4 = "/Sensor/Grupo1/Movimiento";
const char* topicName5 = "/Sensor/Grupo1/CO2";
const char* topicName6 = "/Sensor/Grupo1/TVOC";
IPAddress server(192, 168, 20, 162);
const long interval = 1000;
unsigned long previousMillis = 0;
int count = 0;
void reconnect() {
  while (!mqttClient.connected()) {
    Serial.print("Attempting MQTT connection...");
    if (mqttClient.connect("arduinoClient")) {
      Serial.println("connected");
    } else {
      Serial.print("failed, rc=");
```

```
      Serial.println(" try again in 5 seconds");
      delay(5000);
    }
  }
}

void setup() {
  Serial.begin(9600);
  while (!Serial) {
  }

  Serial.print("Attempting to connect to WPA SSID: ");
  Serial.println(ssid);
  while (WiFi.begin(ssid, pass) != WL_CONNECTED) {
    Serial.print(".");
    delay(5000);
  }

  Serial.println("You're connected to the network");
  Serial.println();
  Serial.print("Attempting to connect to the MQTT broker: ");
  Serial.println(server);

  if (!mqttClient.connect(server, port)) {
    Serial.print("MQTT connection failed! Error code = ");
    Serial.println(mqttClient.connectError());

    while (1);
  }

  Serial.println("You're connected to the MQTT broker!");
  Serial.println();
   dht.begin();
  Wire.begin();
  lightMeter.begin();
  pinMode(SIGNAL_PIN, INPUT);


  if(!ccs.begin()){
 Serial.println("Failed to start sensor! Please check your wiring.");
  while(1);
 }
while(!ccs.available());
float temp = ccs.calculateTemperature();
ccs.setTempOffset(temp - 25.0);

}
void loop() {

  mqttClient.poll();

    float temp = dht.readTemperature();
    char buffer[10];
    dtostrf(temp,0, 0, buffer);
    mqttClient.beginMessage(topicName);
    mqttClient.print(buffer);
    mqttClient.endMessage();
delay(60000);

    float hum = dht.readHumidity();
    dtostrf(hum,0, 0, buffer);
```

```
    mqttClient.beginMessage(topicName2);
    mqttClient.print(buffer);
    mqttClient.endMessage();
delay(60000);

    float lux = lightMeter.readLightLevel();
    dtostrf(lux,0, 0, buffer);
    mqttClient.beginMessage(topicName3);
    mqttClient.print(buffer);
    mqttClient.endMessage();
delay(60000);


if (digitalRead(SIGNAL_PIN) == HIGH) {
    mqttClient.beginMessage(topicName4);
    mqttClient.print("1");
    mqttClient.endMessage();
  } else{
    mqttClient.beginMessage(topicName4);
    mqttClient.print("0");
    mqttClient.endMessage();
    }
delay(60000);

if (ccs.available()) {
   float temp = ccs.calculateTemperature();
  if(!ccs.readData()){
    mqttClient.beginMessage(topicName5);
    mqttClient.print(ccs.geteCO2());
    mqttClient.endMessage();

  } else{
    mqttClient.beginMessage(topicName5);
    mqttClient.print("Error");
    mqttClient.endMessage();
    }
delay(60000);
}

if (ccs.available()) {
   float temp = ccs.calculateTemperature();
  if(!ccs.readData()){
    mqttClient.beginMessage(topicName6);
    mqttClient.print(ccs.getTVOC());
    mqttClient.endMessage();

  } else{
    mqttClient.beginMessage(topicName6);
    mqttClient.print("Error");
    mqttClient.endMessage();
    }
delay(60000);
}
}
```

9.1.2 GRUPO 2

```
#include <ArduinoMqttClient.h>
#include <WiFiNINA.h>
#include <DHT.h>
#include <Wire.h>
#include <BH1750.h>
#include "Adafruit_CCS811.h"
#define SIGNAL_PIN 2
#define DHTPIN 5
#define DHTTYPE DHT11
Adafruit_CCS811 ccs;

char ssid[] = "Cartografia";
char pass[] = "C@rtoU119";

BH1750 lightMeter;
DHT dht(DHTPIN, DHTTYPE);
WiFiClient wifiClient;
MqttClient mqttClient(wifiClient);
int        port     = 1883;

const char* topicName = "/Sensor/Grupo2/Temperatura";
const char* topicName2 = "/Sensor/Grupo2/Humedad";
const char* topicName3 = "/Sensor/Grupo2/Luz";
const char* topicName4 = "/Sensor/Grupo2/Movimiento";
const char* topicName5 = "/Sensor/Grupo2/CO2";
const char* topicName6 = "/Sensor/Grupo2/TVOC";

IPAddress server(192, 168, 20, 162);
const long interval = 1000;
unsigned long previousMillis = 0;
int count = 0;
void reconnect() {
  while (!mqttClient.connected()) {
    Serial.print("Attempting MQTT connection...");
    if (mqttClient.connect("arduinoClient")) {
      Serial.println("connected");
    } else {
      Serial.print("failed, rc=");
      Serial.println(" try again in 5 seconds");
      delay(5000);
    }
  }
}



void setup() {
  Serial.begin(9600);
```

```
  while (!Serial) {
    ;
  }

  Serial.print("Attempting to connect to WPA SSID: ");
  Serial.println(ssid);
  while (WiFi.begin(ssid, pass) != WL_CONNECTED) {
    // failed, retry
    Serial.print(".");
    delay(5000);
  }
  Serial.println("You're connected to the network");
  Serial.println();
  Serial.print("Attempting to connect to the MQTT broker: ");
  Serial.println(server);

  if (!mqttClient.connect(server, port)) {
    Serial.print("MQTT connection failed! Error code = ");
    Serial.println(mqttClient.connectError());
    while (1);
  }
  Serial.println("You're connected to the MQTT broker!");
  Serial.println();
   dht.begin();
  Wire.begin();
  lightMeter.begin();
  pinMode(SIGNAL_PIN, INPUT);


  if(!ccs.begin()){
 Serial.println("Failed to start sensor! Please check your wiring.");
  while(1);
 }
while(!ccs.available());
float temp = ccs.calculateTemperature();
ccs.setTempOffset(temp - 25.0);

}

void loop() {

  mqttClient.poll();

    float temp = dht.readTemperature();
    char buffer[10];
    dtostrf(temp,0, 0, buffer);
    mqttClient.beginMessage(topicName);
    mqttClient.print(buffer);
    mqttClient.endMessage();
delay(60000);

    float hum = dht.readHumidity();
    dtostrf(hum,0, 0, buffer);
    mqttClient.beginMessage(topicName2);
    mqttClient.print(buffer);
    mqttClient.endMessage();
delay(60000);

    float lux = lightMeter.readLightLevel();
    dtostrf(lux,0, 0, buffer);
    mqttClient.beginMessage(topicName3);
```

```
    mqttClient.print(buffer);
    mqttClient.endMessage();
delay(60000);


if (digitalRead(SIGNAL_PIN) == HIGH) {
    mqttClient.beginMessage(topicName4);
    mqttClient.print("1");
    mqttClient.endMessage();
  } else{
    mqttClient.beginMessage(topicName4);
    mqttClient.print("0");
    mqttClient.endMessage();
    }
delay(60000);

if (ccs.available()) {
   float temp = ccs.calculateTemperature();
  if(!ccs.readData()){
    mqttClient.beginMessage(topicName5);
    mqttClient.print(ccs.geteCO2());
    mqttClient.endMessage();

  } else{
    mqttClient.beginMessage(topicName5);
    mqttClient.print("Error");
    mqttClient.endMessage();
    }
delay(60000);
}

if (ccs.available()) {
   float temp = ccs.calculateTemperature();
  if(!ccs.readData()){
    mqttClient.beginMessage(topicName6);
    mqttClient.print(ccs.getTVOC());
//    mqttClient.print("ppb");
    mqttClient.endMessage();

  } else{
    mqttClient.beginMessage(topicName6);
    mqttClient.print("Error");
    mqttClient.endMessage();
    }
delay(60000);
}
}
```

9.1.3 GRUPO 3

```
#include <ArduinoMqttClient.h>
#include <WiFiNINA.h>
#include <DHT.h>
#include <Wire.h>
#include <BH1750.h>
#include "Adafruit_CCS811.h"
#define SIGNAL_PIN 2
#define DHTPIN 5
#define DHTTYPE DHT11

Adafruit_CCS811 ccs;

char ssid[] = "Cartografia";
char pass[] = "C@rtoU119";

BH1750 lightMeter;
DHT dht(DHTPIN, DHTTYPE);
WiFiClient wifiClient;
MqttClient mqttClient(wifiClient);

int port = 1883;
const char* topicName = "/Sensor/Grupo3/Temperatura";
const char* topicName2 = "/Sensor/Grupo3/Humedad";
const char* topicName3 = "/Sensor/Grupo3/Luz";
const char* topicName4 = "/Sensor/Grupo3/Movimiento";
const char* topicName5 = "/Sensor/Grupo3/CO2";
const char* topicName6 = "/Sensor/Grupo3/TVOC";

IPAddress server(192, 168, 20, 162);

const long interval = 1000;
unsigned long previousMillis = 0;

int count = 0;

void reconnect() {
  while (!mqttClient.connected()) {
    Serial.print("Attempting MQTT connection...");
    if (mqttClient.connect("arduinoClient")) {
      Serial.println("connected");
    } else {
      Serial.print("failed, rc=");
      Serial.println(" try again in 5 seconds");
      delay(5000);
    }
  }
}
```

```
void setup() {
  Serial.begin(9600);
  while (!Serial) {
    ;
  }
  Serial.print("Attempting to connect to WPA SSID: ");
  Serial.println(ssid);
  while (WiFi.begin(ssid, pass) != WL_CONNECTED) {
    Serial.print(".");
    delay(5000);
  }

  Serial.println("You're connected to the network");
  Serial.println();
  Serial.print("Attempting to connect to the MQTT broker: ");
  Serial.println(server);

  if (!mqttClient.connect(server, port)) {
    Serial.print("MQTT connection failed! Error code = ");
    Serial.println(mqttClient.connectError());

    while (1);
  }

  Serial.println("You're connected to the MQTT broker!");
  Serial.println();
   dht.begin();
  Wire.begin();
  lightMeter.begin();
  pinMode(SIGNAL_PIN, INPUT);


  if(!ccs.begin()){
 Serial.println("Failed to start sensor! Please check your wiring.");
  while(1);
 }
while(!ccs.available());
float temp = ccs.calculateTemperature();
ccs.setTempOffset(temp - 25.0);

}

void loop() {

  mqttClient.poll();

    float temp = dht.readTemperature();
    char buffer[10];
    dtostrf(temp,0, 0, buffer);
    mqttClient.beginMessage(topicName);
    mqttClient.print(buffer);
    mqttClient.endMessage();
delay(60000);

    float hum = dht.readHumidity();
    dtostrf(hum,0, 0, buffer);
    mqttClient.beginMessage(topicName2);
    mqttClient.print(buffer);
    mqttClient.endMessage();
delay(60000);
```

```
    float lux = lightMeter.readLightLevel();
    dtostrf(lux,0, 0, buffer);
    mqttClient.beginMessage(topicName3);
    mqttClient.print(buffer);
    mqttClient.endMessage();
delay(60000);

if (digitalRead(SIGNAL_PIN) == HIGH) {
    mqttClient.beginMessage(topicName4);
    mqttClient.print("1");
    mqttClient.endMessage();
  } else{
    mqttClient.beginMessage(topicName4);
    mqttClient.print("0");
    mqttClient.endMessage();
    }
delay(60000);

if (ccs.available()) {
   float temp = ccs.calculateTemperature();
  if(!ccs.readData()){
    mqttClient.beginMessage(topicName5);
    mqttClient.print(ccs.geteCO2());
    mqttClient.endMessage();

  } else{
    mqttClient.beginMessage(topicName5);
    mqttClient.print("Error");
    mqttClient.endMessage();
    }
delay(60000);
}

if (ccs.available()) {
   float temp = ccs.calculateTemperature();
  if(!ccs.readData()){
    mqttClient.beginMessage(topicName6);
    mqttClient.print(ccs.getTVOC());
    mqttClient.endMessage();

  } else{
    mqttClient.beginMessage(topicName6);
    mqttClient.print("Error");
    mqttClient.endMessage();
    }
delay(60000);
}
}
```

9.1.4 GRUPO SHIELD

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <DHT.h>
#include <Wire.h>
#include <BH1750.h>
#include "Adafruit_CCS811.h"
#include"AirQuality.h"
#include"Arduino.h"

#define DHTPIN 5
#define DHTTYPE DHT11
#define SIGNAL_PIN 2
Adafruit_CCS811 ccs;
AirQuality airqualitysensor;
int current_quality =-1;

char ssid[] = "Cartografia";
char pass[] = "C@rtoU119";
int status = WL_IDLE_STATUS;

IPAddress server(192, 168, 20, 162);

const char* topicName = "/Sensor/GrupoShield/Temperatura";
const char* topicName2 = "/Sensor/GrupoShield/Humedad";
const char* topicName3 = "/Sensor/GrupoShield/Luz";
const char* topicName4 = "/Sensor/GrupoShield/Movimiento";
const char* topicName5 = "/Sensor/GrupoShield/CO2";
const char* topicName6 = "/Sensor/GrupoShield/TVOC";
const char* topicName7 = "/Sensor/GrupoShield/Calidad_Aire";

BH1750 lightMeter;
DHT dht(DHTPIN, DHTTYPE);
WiFiClient wifiClient;
PubSubClient client(wifiClient);

const int lightPin = 8;
void reconnect() {
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    if (client.connect("arduinoClient")) {
      Serial.println("connected");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      delay(5000);
    }
  }
}
void setup()
{
```

```
Serial.println("Attempting to connect to WPA network...");
  status = WiFi.begin(ssid, pass);

  if ( status != WL_CONNECTED) {
    Serial.println("Couldn't get a wifi connection");
    while(true);
  }
  else {
    Serial.println("Connected to network");
  }

  Serial.begin(9600);

airqualitysensor.init(A0);

  client.setServer(server, 1883);
  dht.begin();
  Wire.begin();
  lightMeter.begin();
  pinMode(SIGNAL_PIN, INPUT);
    Serial.println("CCS811 test");
  if(!ccs.begin()){
 Serial.println("Failed to start sensor! Please check your wiring.");
  while(1);
 }
while(!ccs.available());
float tempp = ccs.calculateTemperature();
ccs.setTempOffset(tempp - 25.0);
}

void loop()
{
  if (!client.connected()) {
    Serial.print("Connecting ...\n");
    client.connect("Arduino Client");
  }
  else {
    // Envio
    float temp = dht.readTemperature();
    float hum = dht.readHumidity();
    float lux = lightMeter.readLightLevel();
    float CO2 = ccs.geteCO2();
    float TVOC = ccs.getTVOC();
    char buffer[10];
    dtostrf(temp,0, 0, buffer);
    client.publish("/Sensor/GrupoShield/Temperatura", buffer);

    delay(15000);
    dtostrf(hum,0, 0, buffer);
    client.publish("/Sensor/GrupoShield/Humedad", buffer);
   delay(15000);
       dtostrf(lux,0, 0, buffer);
    client.publish("/Sensor/GrupoShield/Luz", buffer);
   delay(15000);
if (digitalRead(SIGNAL_PIN) == HIGH) {
  client.publish("/Sensor/GrupoShield/Movimiento", "1");
  } else{
    client.publish("/Sensor/GrupoShield/Movimiento", "0");
    }
 delay(15000);
```

```
if(ccs.available()){
 float tempp = ccs.calculateTemperature();
  if(!ccs.readData()){
    float CO2 = (ccs.geteCO2());
    float TVOC = (ccs.getTVOC());
    char buffer[10];
   dtostrf(CO2,0, 0, buffer);
   client.publish("/Sensor/GrupoShield/CO2",buffer);
   delay(15000);
   dtostrf(TVOC,0, 0, buffer);
   client.publish("/Sensor/GrupoShield/TVOC", buffer);
   delay(15000);}
  else{
   Serial.println("ERROR!");
  while(1);
  }}
    current_quality=airqualitysensor.slope();
    if (current_quality >= 0)
    {
        if (current_quality==0)
            client.publish("/Sensor/GrupoShield/Calidad_Aire", "4");
        else if (current_quality==1)
            client.publish("/Sensor/GrupoShield/Calidad_Aire", "3");
        else if (current_quality==2)
            client.publish("/Sensor/GrupoShield/Calidad_Aire", "2");
        else if (current_quality ==3)
            client.publish("/Sensor/GrupoShield/Calidad_Aire", "1");
    }}
  delay(5000);
  client.loop();}
ISR(TIMER2_OVF_vect)
{
    if(airqualitysensor.counter==122)
    {
        airqualitysensor.last_vol=airqualitysensor.first_vol;
        airqualitysensor.first_vol=analogRead(A0);
        airqualitysensor.counter=0;
        airqualitysensor.timer_index=1;
        PORTB=PORTB^0x20;
    }
    else   {
        airqualitysensor.counter++;
   }}
```

9.2 CODIGO NODE RED

[{"id":"89502d50.a538e","type":"tab","label":"CLIENTE WEB","disabled":false,"info":""},{"id":"20d6d10a.dddf5e","type":"ui_template","z":"89502d50.a538e","group":"b6df9849.2ee1a8","name":"Aula 352","order":6,"width":0,"height":0,"format":" \n<img src=\"http://localhost:1880/ui/Floor2.png\" height=\"700\" width=\"700\">
\n\n\n <div ng-repeat=\"el in msg.payload\" style=\"left: {{170}}px; position: absolute; top: {{350}}px;\">\n \n
\n <div style=\"color: {{el.color}}; font-size: 10px; padding-top: 2px; text-align: center;\">\n {{el.text}} \n <div style=\"color: {{el.color}}; font-size: 25px; padding-top: 2px; text-align: center;\">\n {{msg.payload}} \n \n \n ","storeOutMessages":true,"fwdInMessages":true,"templateScope":"local","x":880,"y":400,"wires":[[]]},{"id":"1e6744c1.d1e9eb","type":"ui_template","z":"89502d50.a538e","group":"b6df9849.2ee1a8","name":"Aula 352","order":0,"width":0,"height":0,"format":" \n<img src=\"http://localhost:1880/ui/FloorShield.png\" height=\"700\" width=\"700\">
\n\n\n <div ng-repeat=\"el in msg.payload\" style=\"left: {{20}}px; position: absolute; top: {{60}}px;\">\n \n
\n <div style=\"color: {{el.color}}; font-size: 10px; padding-top: 2px; text-align: center;\">\n {{el.text}} \n <div style=\"color: {{el.color}}; font-size: 25px; padding-top: 2px; text-align: center;\">\n {{msg.payload}} \n \n \n ","storeOutMessages":true,"fwdInMessages":true,"templateScope":"local","x":880,"y":460,"wires":[[]]},{"id":"75e07708.7e4478","type":"mqtt in","z":"89502d50.a538e","name":"Todo Grupo 1","topic":"/Sensor/Grupo1/Alles","qos":"2","datatype":"auto","broker":"e67338bd.c1a978","x":270,"y":340,"wires":[["5386dfa.48a042"]]},{"id":"f596a0b9.3ec9e8","type":"ui_template","z":"89502d50.a538e","group":"7391748f.f51ce4","name":"Aula 352","order":0,"width":0,"height":0,"format":" \n<img src=\"http://localhost:1880/ui/Floor1.png\" height=\"700\" width=\"700\">
\n\n\n <div ng-repeat=\"el in msg.payload\" style=\"left: {{410}}px; position: absolute; top: {{60}}px;\">\n \n
\n <div style=\"color: {{el.color}}; font-size: 10px; padding-top: 2px; text-align: center;\">\n {{el.text}} \n <div style=\"color: {{el.color}}; font-size: 25px; padding-top: 2px; text-align: center;\">\n {{msg.payload}} \n \n \n ","storeOutMessages":true,"fwdInMessages":true,"templateScope":"local","x":880,"y":340,"wires":[[]]},{"id":"5386dfa.48a042","type":"function","z":"89502d50.a538e","name":"Aula 352","func":"return {\n \"topic\": msg.topic,\n \"payload\": {\n \"Grupo 1\": msg.payload \n \n }\n};\n","outputs":1,"noerr":0,"x":600,"y":340,"wires":[["f596a0b9.3ec9e8"]]},{"id":"b0921c3b.af55b","type":"mqtt in","z":"89502d50.a538e","name":"Todo Grupo 2","topic":"/Sensor/Grupo2/Alles","qos":"2","datatype":"auto","broker":"e67338bd.c1a978","x":270,"y":400,"wires":[["aa5fd63b.04327"]]},{"id":"aa5fd63b.04327","type":"function","z":"89502d50.a538e","name":"Aula 352","func":"return {\n \"topic\": msg.topic,\n \"payload\": {\n \"Grupo 2\": msg.payload \n \n }\n};\n","outputs":1,"noerr":0,"x":600,"y":400,"wires":[["20d6d10a.dddf5e"]]},{"id":"1b965204.a749d6","type":"mqtt in","z":"89502d50.a538e","name":"Todo Grupo Shield","topic":"/Sensor/GrupoShield/Humedad","qos":"2","datatype":"auto","broker":"e67338bd.c1a978","x":290,"y":460,"wires":[["8f758e38.6c24e8"]]},{"id":"8f758e38.6c24e8","type":"function","z":"89502d50.a538e","name":"Aula 352","func":"var msg1=global.get(\"MS23\");\nvar msg2=global.get(\"MS25\");\nvar msg3=global.get(\"MS27\");\nvar msg4=global.get(\"MS29\");\nvar msg5=global.get(\"MS31\");\nvar msg6=global.get(\"MS33\");\nvar msg7=global.get(\"MS34\");\n\n\nmsg.payload = \"Temperatura:\"+msg1+\"º\"+\", \"+\" Humedad: \"+msg2+\"%\"+\", \"+\" Luz: \"+msg3+\" lux \"+msg6+\" .....\"+\" CO2: \"+msg4+\" ppm, TVOC: \"+msg5+\" ppb \";\n\nreturn {\n \"topic\": msg.topic,\n \"payload\": {\n \"Grupo Shield\": msg.payload\n \n }\n};\n","outputs":1,"noerr":0,"x":600,"y":460,"wires":[["1e6744c1.d1e9eb"]]},{"id":"6fb17ee7.848b68","type":"mqtt in","z":"89502d50.a538e","name":"Todo Grupo 3","topic":"/Sensor/Grupo3/Alles","qos":"2","datatype":"auto","broker":"e67338bd.c1a978","x":270,"y":520,"wires":[["a52cfa86.ed80c8"]]},{"id":"a52cfa86.ed80c8","type":"function","z":"89502d50.a538e","name":"Aula 352","func":"return {\n \"topic\": msg.topic,\n \"payload\": {\n \"Grupo 3\": msg.payload \n \n }\n};\n","outputs":1,"noerr":0,"x":600,"y":520,"wires":[["a781a3c9.d0678"]]},{"id":"a781a3c9.d0678","type":"ui_template","z":"89502d50.a538e","group":"7391748f.f51ce4","name":"Aula 352","order":0,"width":"0","height":"0","format":" \n<img src=\"http://localhost:1880/ui/Floor.png\" height=\"700\" width=\"700\">
\n\n\n <div ng-repeat=\"el in msg.payload\" style=\"left: {{3}}px; position: absolute; top: {{325}}px;\">\n \n
\n <div style=\"color: {{el.color}}; font-size: 10px; padding-top: 2px; text-align: center;\">\n {{el.text}} \n <div style=\"color: {{el.color}}; font-size: 25px; padding-top: 2px; text-align: center;\">\n {{msg.payload}} \n \n \n \n","storeOutMessages":true,"fwdInMessages":true,"templateScope":"local","x":880,"y":520,"wires":[[]]},{"id":"780d9833.67e9","type":"function","z":"89502d50.a538e","name":"Guardar Variable","func":"global.set(\"MS3\",msg.payload);","outputs":1,"noerr":0,"x":530,"y":600,"wires":[[]]},{"id":"f5cc66ca.56b738","type":"function","z":"89502d50.a538e","name":"Obtener Variable","func":"var msg1=global.get(\"MS3\");\nvar msg2=global.get(\"MS4\");\nvar msg3=global.get(\"MS22\");\nvar msg4=global.get(\"MS23\");\n\n\nvar

MS5=(msg2+msg1+msg3+msg4)/(4);\n\n\nmsg.payload = MS5;\n\n\nreturn msg;","outputs":1,"noerr":0,"x":650,"y":760,"wires":[["53e7339.626e2cc","9c3027f3.ed4c08"]]},{"id":"5fd96cba.381664","type":"inject","z":"89502d50.a538e","name":"","topic":"","payload":"","payloadType":"date","repeat":"30","crontab":"","once":false,"onceDelay":0.1,"x":430,"y":760,"wires":[["f5cc66ca.56b738"]]},{"id":"a7e6acc.731eed","type":"function","z":"89502d50.a538e","name":"Guardar Variable","func":"global.set(\"MS4\",msg.payload);","outputs":1,"noerr":0,"x":530,"y":660,"wires":[[]]},{"id":"4884f7d4.2f23f8","type":"mqtt in","z":"89502d50.a538e","name":"Temp 1","topic":"/Sensor/Grupo1/Temperatura","qos":"2","datatype":"json","broker":"e67338bd.c1a978","x":250,"y":600,"wires":[["780d9833.67e9"]]},{"id":"4eea97e5.98a648","type":"mqtt in","z":"89502d50.a538e","name":"Temp 2","topic":"/Sensor/Grupo2/Temperatura","qos":"2","datatype":"json","broker":"e67338bd.c1a978","x":250,"y":660,"wires":[["a7e6acc.731eed"]]},{"id":"afc6b1aa.4bff7","type":"function","z":"89502d50.a538e","name":"Guardar Variable","func":"global.set(\"MS6\",msg.payload);","outputs":1,"noerr":0,"x":650,"y":860,"wires":[[]]},{"id":"65f29e90.cd25a8","type":"function","z":"89502d50.a538e","name":"Obtener Variable","func":"var msg1=global.get(\"MS6\");\nvar msg2=global.get(\"MS7\");\nvar msg3=global.get(\"MS24\");\nvar msg4=global.get(\"MS25\");\n\n\nvar MS8=(msg2+msg1+msg3+msg4)/(4);\n\n\nmsg.payload = MS8;\n\n\nreturn msg;","outputs":1,"noerr":0,"x":670,"y":1000,"wires":[["c7193496.bd042","51475cbc.8f7d7c"]]},{"id":"1728d941.43e1c7","type":"inject","z":"89502d50.a538e","name":"","topic":"","payload":"","payloadType":"date","repeat":"30","crontab":"","once":false,"onceDelay":0.1,"x":450,"y":1000,"wires":[["65f29e90.cd25a8"]]},{"id":"3ab39cf7.0a23ac","type":"function","z":"89502d50.a538e","name":"Guardar Variable","func":"global.set(\"MS7\",msg.payload);","outputs":1,"noerr":0,"x":650,"y":920,"wires":[[]]},{"id":"857bcd1b.465478","type":"mqtt in","z":"89502d50.a538e","name":"Humedad 1","topic":"/Sensor/Grupo1/Humedad","qos":"2","datatype":"json","broker":"e67338bd.c1a978","x":390,"y":860,"wires":[["afc6b1aa.4bff7"]]},{"id":"c7193496.bd042","type":"ui_chart","z":"89502d50.a538e","name":"","group":"2a744fe7.fb3e18","order":0,"width":0,"height":0,"label":"Humedad Media en %","chartType":"line","legend":"true","xformat":"HH:mm:ss","interpolate":"linear","nodata":"","dot":false,"ymin":"","ymax":"","removeOlder":1,"removeOlderPoints":"","removeOlderUnit":"3600","cutout":0,"useOneColor":false,"colors":["#1f77b4","#aec7e8","#ff7f0e","#2ca02c","#98df8a","#d62728","#ff9896","#9467bd","#c5b0d5"],"useOldStyle":false,"outputs":1,"x":980,"y":1000,"wires":[[]]},{"id":"51475cbc.8f7d7c","type":"ui_gauge","z":"89502d50.a538e","name":"Humedad Media","group":"2a744fe7.fb3e18","order":0,"width":0,"height":0,"gtype":"gage","title":"","label":"%","format":"{{value}}","min":0,"max":"100","colors":["#00b500","#e6e600","#ca3838"],"seg1":"","seg2":"","x":940,"y":1060,"wires":[]},{"id":"4051d6ce.445558","type":"mqtt in","z":"89502d50.a538e","name":"Humedad 2","topic":"/Sensor/Grupo2/Humedad","qos":"2","datatype":"json","broker":"e67338bd.c1a978","x":390,"y":920,"wires":[["3ab39cf7.0a23ac"]]},{"id":"53e7339.626e2cc","type":"ui_chart","z":"89502d50.a538e","name":"","group":"420c050c.649c64","order":1,"width":0,"height":0,"label":"Temperatura Media en ºC","chartType":"line","legend":"false","xformat":"HH:mm:ss","interpolate":"linear","nodata":"","dot":false,"ymin":"","ymax":"","removeOlder":1,"removeOlderPoints":"","removeOlderUnit":"3600","cutout":0,"useOneColor":false,"colors":["#1f77b4","#aec7e8","#ff7f0e","#2ca02c","#98df8a","#d62728","#ff9896","#9467bd","#c5b0d5"],"useOldStyle":false,"outputs":1,"x":1010,"y":740,"wires":[[]]},{"id":"9c3027f3.ed4c08","type":"ui_gauge","z":"89502d50.a538e","name":"Temperatura Media","group":"420c050c.649c64","order":2,"width":0,"height":0,"gtype":"gage","title":"gauge","label":"º","format":"{{value}}","min":0,"max":"45","colors":["#00b500","#e6e600","#ca3838"],"seg1":"","seg2":"","x":1000,"y":800,"wires":[]},{"id":"c6beb785.d793a","type":"function","z":"89502d50.a538e","name":"Guardar Variable","func":"global.set(\"MS9\",msg.payload);","outputs":1,"noerr":0,"x":650,"y":1180,"wires":[[]]},{"id":"edce587c.133518","type":"function","z":"89502d50.a538e","name":"Obtener Variable","func":"var msg1=global.get(\"MS9\");\nvar msg2=global.get(\"MS10\");\nvar msg3=global.get(\"MS26\");\nvar msg4=global.get(\"MS27\");\n\n\nvar MS11=(msg2+msg1+msg3+msg4)/(4);\n\n\nmsg.payload = MS11;\n\n\nreturn msg;","outputs":1,"noerr":0,"x":670,"y":1320,"wires":[["c157013a.39274","af925814.c736b"]]},{"id":"616fdd1b.56ea14","type":"inject","z":"89502d50.a538e","name":"","topic":"","payload":"","payloadType":"date","repeat":"30","crontab":"","once":false,"onceDelay":0.1,"x":450,"y":1320,"wires":[["edce587c.133518"]]},{"id":"34726c11.bde23c","type":"function","z":"89502d50.a538e","name":"Guardar Variable","func":"global.set(\"MS10\",msg.payload);","outputs":1,"noerr":0,"x":650,"y":1240,"wires":[[]]},{"id":"626a1878.037eb","type":"mqtt in","z":"89502d50.a538e","name":"Luz 1","topic":"/Sensor/Grupo1/Luz","qos":"2","datatype":"json","broker":"e67338bd.c1a978","x":370,"y":1180,"wires":[["c6beb785.d793a"]]},{"id":"266b6778.dc8b08","type":"mqtt in","z":"89502d50.a538e","name":"Luz 2","topic":"/Sensor/Grupo2/Luz","qos":"2","datatype":"json","broker":"e67338bd.c1a978","x":370,"y":1240,"wires":[["34726c11.bde23c"]]},{"id":"c157013a.39274","type":"ui_chart","z":"89502d50.a538e","name":"","group":"f33d3453.29001","order":1,"width":0,"height":0,"label":"Luz Media en lux","chartType":"line","legend":"false","xformat":"HH:mm:ss","interpolate":"linear","nodata":"","dot":false,"ymin":"","ymax":"","removeOlder":1,"removeOlderPoints":"","removeOlderUnit":"3600","cutout":0,"useOneColor":false,"colors":["#1f77b4","#aec7e8","#ff7f0e","#2ca02c","#98df8a","#d62728","#ff9896","#9467bd","#c5b0d5"],"useOldStyle":false,"outputs":1,"x":980,"y":1300,"wires":[[]]},{"id":"af925814.c736b","type":"ui_gauge","z":"89502d50.a538e","name":"Luz Media","group":"f33d3453.29001","order":2,"width":0,"height":0,"gtype":"gage","title":"gauge","label":"lux","format":"{{value}}","min":0,"max":"5000","colors":["#00b500","#e6e600","#ca3838"],"seg1":"","seg2":"","x":970,"y":1380,"wires":[]},{"id":"4030acb5.4584f4","type":"function","z":"89502d50.a538e","name":"Guardar Variable","func":"global.set(\"MS12\",msg.payload);","outputs":1,"noerr":0,"x":630,"y":1480,"wires":[[]]},{"id":"4fa4f553.654414","type":"function","z":"89502d50.a538e","name":"Obtener Variable","func":"var msg1=global.get(\"MS13\");\nvar msg2=global.get(\"MS12\");\nvar msg3=global.get(\"MS28\");\nvar msg4=global.get(\"MS29\");\n\n\nvar

MS14=(msg2+msg1+msg3+msg4)/(4);\n\n\nmsg.payload = MS14;\n\n\nreturn msg;","outputs":1,"noerr":0,"x":650,"y":1620,"wires":[["7632bf21.d002","22a181b5.31b126"]]},{"id":"f66bb4dd.63585","type":"inject","z":"89502d50.a538e","name":"","topic":"","payload":"","payloadType":"date","repeat":"30","crontab":"","once":false,"onceDelay":0.1,"x":430,"y":1620,"wires":[["4fa4f553.654414"]]},{"id":"d386776a.99cdf","type":"function","z":"89502d50.a538e","name":"Guardar Variable","func":"global.set(\"MS13\",msg.payload);","outputs":1,"noerr":0,"x":630,"y":1540,"wires":[[]]},{"id":"4d6aaab6.6e274c","type":"mqtt in","z":"89502d50.a538e","name":"CO2 1","topic":"/Sensor/Grupo1/CO2","qos":"2","datatype":"json","broker":"e67338bd.c1a978","x":350,"y":1480,"wires":[["4030acb5.4584f4"]]},{"id":"2e8cd31b.75d4dc","type":"mqtt in","z":"89502d50.a538e","name":"CO2 2","topic":"/Sensor/Grupo2/CO2","qos":"2","datatype":"json","broker":"e67338bd.c1a978","x":350,"y":1540,"wires":[["d386776a.99cdf"]]},{"id":"64b573f5.0a721c","type":"function","z":"89502d50.a538e","name":"Guardar Variable","func":"global.set(\"MS15\",msg.payload);","outputs":1,"noerr":0,"x":610,"y":1740,"wires":[[]]},{"id":"87e264cb.eb9728","type":"function","z":"89502d50.a538e","name":"Obtener Variable","func":"var msg1=global.get(\"MS15\");\nvar msg2=global.get(\"MS16\");\nvar msg3=global.get(\"MS30\");\nvar msg4=global.get(\"MS31\");\n\n\nvar MS17=(msg2+msg1+msg3+msg4)/(4);\n\n\nmsg.payload = MS17;\n\n\nreturn msg;","outputs":1,"noerr":0,"x":630,"y":1880,"wires":[["103fc722.062961","efccd9a2.bf8f"]]},{"id":"24b90bb4.42a71c","type":"inject","z":"89502d50.a538e","name":"","topic":"","payload":"","payloadType":"date","repeat":"30","crontab":"","once":false,"onceDelay":0.1,"x":410,"y":1880,"wires":[["87e264cb.eb9728"]]},{"id":"74b3d013.80464","type":"function","z":"89502d50.a538e","name":"Guardar Variable","func":"global.set(\"MS16\",msg.payload);","outputs":1,"noerr":0,"x":610,"y":1800,"wires":[[]]},{"id":"1dcb852f.1235ab","type":"mqtt in","z":"89502d50.a538e","name":"TVOC 1","topic":"/Sensor/Grupo1/TVOC","qos":"2","datatype":"json","broker":"e67338bd.c1a978","x":340,"y":1740,"wires":[["64b573f5.0a721c"]]},{"id":"b4dd0aa8.2ecba8","type":"mqtt in","z":"89502d50.a538e","name":"TVOC 2","topic":"/Sensor/Grupo2/TVOC","qos":"2","datatype":"json","broker":"e67338bd.c1a978","x":340,"y":1800,"wires":[["74b3d013.80464"]]},{"id":"7632bf21.d002","type":"ui_chart","z":"89502d50.a538e","name":"CO2 Media","group":"f3f6bec.a1c66c","order":1,"width":0,"height":0,"label":"CO2 Media en ppm","chartType":"line","legend":"false","xformat":"HH:mm:ss","interpolate":"linear","nodata":"","dot":false,"ymin":"","ymax":"","removeOlder":1,"removeOlderPoints":"","removeOlderUnit":"3600","cutout":0,"useOneColor":false,"colors":["#1f77b4","#aec7e8","#ff7f0e","#2ca02c","#98df8a","#d62728","#ff9896","#9467bd","#c5b0d5"],"useOldStyle":false,"outputs":1,"x":1000,"y":1600,"wires":[[]]},{"id":"22a181b5.31b126","type":"ui_gauge","z":"89502d50.a538e","name":"CO2 Media","group":"f3f6bec.a1c66c","order":2,"width":0,"height":0,"gtype":"gage","title":"gauge","label":"ppm","format":"{{value}}","min":0,"max":"8000","colors":["#00b500","#e6e600","#ca3838"],"seg1":"","seg2":"","x":1000,"y":1680,"wires":[]},{"id":"103fc722.062961","type":"ui_chart","z":"89502d50.a538e","name":"TVOC Media","group":"a80e4fd6.1b3bd","order":2,"width":0,"height":0,"label":"TVOC Media en ppb","chartType":"line","legend":"false","xformat":"HH:mm:ss","interpolate":"linear","nodata":"","dot":false,"ymin":"","ymax":"","removeOlder":1,"removeOlderPoints":"","removeOlderUnit":"3600","cutout":0,"useOneColor":false,"colors":["#1f77b4","#aec7e8","#ff7f0e","#2ca02c","#98df8a","#d62728","#ff9896","#9467bd","#c5b0d5"],"useOldStyle":false,"outputs":1,"x":960,"y":1840,"wires":[[]]},{"id":"efccd9a2.bf8f","type":"ui_gauge","z":"89502d50.a538e","name":"TVOC Media","group":"a80e4fd6.1b3bd","order":3,"width":0,"height":0,"gtype":"gage","title":"gauge","label":"ppb","format":"{{value}}","min":0,"max":"2000","colors":["#00b500","#e6e600","#ca3838"],"seg1":"","seg2":"","x":960,"y":1920,"wires":[]},{"id":"89ceb42e.14c","type":"function","z":"89502d50.a538e","name":"Grupo 1","func":"var d = new Date();\nvar u = global.get(\"MS3\");\nvar h= global.get(\"MS6\");\nvar luz= global.get(\"MS9\");\nvar co = global.get(\"MS12\");\nvar tvoc =global.get(\"MS15\");\nvar mo =global.get(\"MS18\");\nvar t = d.getTime();\n\nmsg.payload={lat:37.787245, lon:-3.777928,name:\"Grupo 1\", icon:\"university\",\"Fecha y Hora\":d, \"Temperatura\":u, \"Humedad\":h, \"Luz\":luz, \"CO2\":co, \"TVOC\":tvoc,\"Movimiento\":mo};\n\nreturn msg;","outputs":1,"noerr":0,"x":640,"y":60,"wires":[["a10bb197.5de8b8"]]},{"id":"a10bb197.5de8b8","type":"worldmap","z":"89502d50.a538e","name":"Mapa","lat":"37.7875684143","lon":"-3.7775702029","zoom":"","layer":"OSM","cluster":"","maxage":"","usermenu":"hide","panit":"true","hiderightclick":"false","coords":"none","path":"","x":850,"y":140,"wires":[]},{"id":"188a29e5.eb9eee","type":"inject","z":"89502d50.a538e","name":"Intervalo de 10 segundos","topic":"","payload":"","payloadType":"date","repeat":"10","crontab":"","once":false,"onceDelay":0.1,"x":400,"y":60,"wires":[["89ceb42e.14c"]]},{"id":"9f90bfda.52863","type":"function","z":"89502d50.a538e","name":"Grupo 2","func":"var d = new Date();\nvar u = global.get(\"MS4\");\nvar h= global.get(\"MS7\");\nvar luz= global.get(\"MS10\");\nvar co = global.get(\"MS13\");\nvar tvoc =global.get(\"MS16\");\nvar mo =global.get(\"MS20\");\nvar t = d.getTime();\n\nmsg.payload={lat:37.787273, lon: -3.777837,name:\"Grupo 2\", icon:\"university\",\"Fecha y Hora\":d, \"Temperatura\":u, \"Humedad\":h, \"Luz\":luz, \"CO2\":co, \"TVOC\":tvoc,\"Movimiento\":mo};\n\nreturn msg;","outputs":1,"noerr":0,"x":640,"y":120,"wires":[["a10bb197.5de8b8"]]},{"id":"47fa55e1.2752fc","type":"inject","z":"89502d50.a538e","name":"Intervalo de 10 segundos","topic":"","payload":"","payloadType":"date","repeat":"10","crontab":"","once":true,"onceDelay":0.1,"x":400,"y":120,"wires":[["9f90bfda.52863"]]},{"id":"e0ff99cf.d4a6c","type":"function","z":"89502d50.a538e","name":"Grupo 3","func":"var d = new Date();\nvar u = global.get(\"MS22\");\nvar h= global.get(\"MS24\");\nvar luz= global.get(\"MS26\");\nvar co = global.get(\"MS28\");\nvar tvoc =global.get(\"MS30\");\nvar mo =global.get(\"MS32\");\nvar t = d.getTime();\n\nmsg.payload={lat:37.787252, lon: -3.777792,name:\"Grupo 3\", icon:\"university\",\"Fecha y Hora\":d, \"Temperatura\":u, \"Humedad\":h, \"Luz\":luz, \"CO2\":co, \"TVOC\":tvoc,\"Movimiento\":mo};\n\nreturn msg;","outputs":1,"noerr":0,"x":640,"y":180,"wires":[["a10bb197.5de8b8"]]},{"id":"12d872.89d24f8e","type":"function","z":"89502d50.a538e","name":"Grupo Shield","func":"var d = new Date();\nvar u = global.get(\"MS23\");\nvar h=

global.get(\"MS25\");\nvar luz= global.get(\"MS27\");\nvar co = global.get(\"MS29\");\nvar tvoc =global.get(\"MS31\");\nvar mo =global.get(\"MS33\");\nvar ai =global.get(\"MS34\");\nvar t = d.getTime();\n\nmsg.payload={lat:37.787175, lon: -3.777823,name:\"Grupo Shield\", icon:\"university\",\"Fecha y Hora\":d, \"Temperatura\":u, \"Humedad\":h, \"Luz\":luz, \"CO2\":co, \"TVOC\":tvoc,\"Movimiento\":mo,\"Calidad de Aire\":ai};\n\nreturn msg;","outputs":1,"noerr":0,"x":650,"y":240,"wires":[["a10bb197.5de8b8"]]},{"id":"9e29ff63.654e78","type":"inject","z":"89502d50.a538e","name":"Intervalo de 10 segundos","topic":"","payload":"","payloadType":"date","repeat":"10","crontab":"","once":true,"onceDelay":0.1,"x":400,"y":180,"wires":[["e0ff99cf.d4a6c"]]},{"id":"3edab2f1.04799e","type":"inject","z":"89502d50.a538e","name":"Intervalo de 10 segundos","topic":"","payload":"","payloadType":"date","repeat":"10","crontab":"","once":true,"onceDelay":0.1,"x":400,"y":240,"wires":[["12d872.89d24f8e"]]},{"id":"8b840a31.a3158","type":"ui_template","z":"89502d50.a538e","group":"45dcf39f.70f49c","name":"Localizacion de los Sensores","order":0,"width":0,"height":0,"format":" \n Localización Actual de todos los Sensores\n ","storeOutMessages":true,"fwdInMessages":true,"templateScope":"local","x":1080,"y":80,"wires":[[]]},{"id":"f2148824.0a844","type":"ui_gauge","z":"89502d50.a538e","name":"","group":"258e2d90.885d1a","order":1,"width":0,"height":0,"gtype":"gage","title":"Grupo 1","label":"NO/SI","format":"{{value}}","min":0,"max":"1","colors":["#00b500","#e6e600","#ca3838"],"seg1":"","seg2":"","x":620,"y":2000,"wires":[]},{"id":"1b6110dc.493b8f","type":"mqtt in","z":"89502d50.a538e","name":"Movimiento 1","topic":"/Sensor/Grupo1/Movimiento","qos":"2","datatype":"json","broker":"e67338bd.c1a978","x":350,"y":2000,"wires":[["f2148824.0a844"]]},{"id":"991c594a.f8bf5","type":"mqtt in","z":"89502d50.a538e","name":"Movimiento 2","topic":"/Sensor/Grupo2/Movimiento","qos":"2","datatype":"json","broker":"e67338bd.c1a978","x":350,"y":2120,"wires":[["7556aae7.856c84"]]},{"id":"7556aae7.856c84","type":"ui_gauge","z":"89502d50.a538e","name":"","group":"258e2d90.885d1a","order":2,"width":0,"height":0,"gtype":"gage","title":"Grupo 2","label":"NO/SI","format":"{{value}}","min":0,"max":"1","colors":["#00b500","#e6e600","#ca3838"],"seg1":"","seg2":"","x":620,"y":2120,"wires":[]},{"id":"96ba13de.870f1","type":"function","z":"89502d50.a538e","name":"Guardar Variable","func":"global.set(\"MS18\",msg.payload);","outputs":1,"noerr":0,"x":650,"y":2060,"wires":[[]]},{"id":"ee3360d4.84067","type":"function","z":"89502d50.a538e","name":"Guardar Variable","func":"global.set(\"MS20\",msg.payload);","outputs":1,"noerr":0,"x":650,"y":2180,"wires":[[]]},{"id":"83c680b0.4a8408","type":"function","z":"89502d50.a538e","name":"Guardar Variable","func":"global.set(\"MS22\",msg.payload);","outputs":1,"noerr":0,"x":1070,"y":600,"wires":[[]]},{"id":"cd5df4a4.443ee","type":"function","z":"89502d50.a538e","name":"Guardar Variable","func":"global.set(\"MS23\",msg.payload);","outputs":1,"noerr":0,"x":1070,"y":660,"wires":[[]]},{"id":"7096ff7b.7a832","type":"mqtt in","z":"89502d50.a538e","name":"Temp 3","topic":"/Sensor/Grupo3/Temperatura","qos":"2","datatype":"json","broker":"e67338bd.c1a978","x":790,"y":600,"wires":[["83c680b0.4a8408"]]},{"id":"721db94f.e0475","type":"mqtt in","z":"89502d50.a538e","name":"Temp Shield","topic":"/Sensor/GrupoShield/Temperatura","qos":"2","datatype":"json","broker":"e67338bd.c1a978","x":810,"y":660,"wires":[["cd5df4a4.443ee"]]},{"id":"4a226509.b1d75c","type":"function","z":"89502d50.a538e","name":"Guardar Variable","func":"global.set(\"MS24\",msg.payload);","outputs":1,"noerr":0,"x":1130,"y":860,"wires":[[]]},{"id":"211c1440.01a284","type":"function","z":"89502d50.a538e","name":"Guardar Variable","func":"global.set(\"MS25\",msg.payload);","outputs":1,"noerr":0,"x":1130,"y":920,"wires":[[]]},{"id":"5440c316.ccc194","type":"mqtt in","z":"89502d50.a538e","name":"Humedad 3","topic":"/Sensor/Grupo3/Humedad","qos":"2","datatype":"json","broker":"e67338bd.c1a978","x":870,"y":860,"wires":[["4a226509.b1d75c"]]},{"id":"92b49fb5.f985f8","type":"mqtt in","z":"89502d50.a538e","name":"Humedad Shield","topic":"/Sensor/GrupoShield/Humedad","qos":"2","datatype":"json","broker":"e67338bd.c1a978","x":880,"y":920,"wires":[["211c1440.01a284"]]},{"id":"716dd6a5.2714d","type":"function","z":"89502d50.a538e","name":"Guardar Variable","func":"global.set(\"MS26\",msg.payload);","outputs":1,"noerr":0,"x":1130,"y":1180,"wires":[[]]},{"id":"b1d07bae.38c","type":"function","z":"89502d50.a538e","name":"Guardar Variable","func":"global.set(\"MS27\",msg.payload);","outputs":1,"noerr":0,"x":1130,"y":1240,"wires":[[]]},{"id":"1225f302.367165","type":"mqtt in","z":"89502d50.a538e","name":"Luz 3","topic":"/Sensor/Grupo3/Luz","qos":"2","datatype":"json","broker":"e67338bd.c1a978","x":850,"y":1180,"wires":[["716dd6a5.2714d"]]},{"id":"869c2c39.0aa95","type":"mqtt in","z":"89502d50.a538e","name":"Luz Shield","topic":"/Sensor/GrupoShield/Luz","qos":"2","datatype":"json","broker":"e67338bd.c1a978","x":860,"y":1240,"wires":[["b1d07bae.38c"]]},{"id":"16ec96fe.611209","type":"function","z":"89502d50.a538e","name":"Guardar Variable","func":"global.set(\"MS28\",msg.payload);","outputs":1,"noerr":0,"x":1130,"y":1480,"wires":[[]]},{"id":"ee1ca2ce.c82b6","type":"function","z":"89502d50.a538e","name":"Guardar Variable","func":"global.set(\"MS29\",msg.payload);","outputs":1,"noerr":0,"x":1130,"y":1540,"wires":[[]]},{"id":"437656a8.c40fc8","type":"mqtt in","z":"89502d50.a538e","name":"CO2 3","topic":"/Sensor/Grupo3/CO2","qos":"2","datatype":"json","broker":"e67338bd.c1a978","x":850,"y":1480,"wires":[["16ec96fe.611209"]]},{"id":"703b7f44.a2f47","type":"mqtt in","z":"89502d50.a538e","name":"CO2 Shield","topic":"/Sensor/GrupoShield/CO2","qos":"2","datatype":"json","broker":"e67338bd.c1a978","x":870,"y":1540,"wires":[["ee1ca2ce.c82b6"]]},{"id":"77e390b9.1ee2a","type":"function","z":"89502d50.a538e","name":"Guardar Variable","func":"global.set(\"MS30\",msg.payload);","outputs":1,"noerr":0,"x":1090,"y":1740,"wires":[[]]},{"id":"98b6653e.c2eb9","type":"function","z":"89502d50.a538e","name":"Guardar Variable","func":"global.set(\"MS31\",msg.payload);","outputs":1,"noerr":0,"x":1090,"y":1800,"wires":[[]]},{"id":"84261320.a8797","type":"mqtt in","z":"89502d50.a538e","name":"TVOC 3","topic":"/Sensor/Grupo3/TVOC","qos":"2","datatype":"json","broker":"e67338bd.c1a978","x":820,"y":1740,"wires":[["77e390b9.1ee2a"]]},{"id":"4de06695.7b486","type":"mqtt in","z":"89502d50.a538e","name":"TVOC

Shield","topic":"/Sensor/GrupoShield/TVOC","qos":"2","datatype":"json","broker":"e67338bd.c1a978","x":830,"y":1800,"wires":[["98b6653e.c2eb9"]]},{"id":"f1250756.d8603","type":"mqtt in","z":"89502d50.a538e","name":"Movimiento 3","topic":"/Sensor/Grupo3/Movimiento","qos":"2","datatype":"json","broker":"e67338bd.c1a978","x":350,"y":2240,"wires":[["6eceb2af.115344"]]},{"id":"6a3c90c6.ea0518","type":"mqtt in","z":"89502d50.a538e","name":"Movimiento Shield","topic":"/Sensor/GrupoShield/Movimiento","qos":"2","datatype":"json","broker":"e67338bd.c1a978","x":370,"y":2360,"wires":[["3b520cf9.e7f1fc"]]},{"id":"cac96c9d.96a7","type":"function","z":"89502d50.a538e","name":"Guardar Variable","func":"global.set(\"MS32\",msg.payload);","outputs":1,"noerr":0,"x":650,"y":2300,"wires":[[]]},{"id":"1e4e75db.43594a","type":"function","z":"89502d50.a538e","name":"Guardar Variable","func":"global.set(\"MS33\",msg.payload);","outputs":1,"noerr":0,"x":650,"y":2420,"wires":[[]]},{"id":"6eceb2af.115344","type":"ui_gauge","z":"89502d50.a538e","name":"Grupo 3","group":"258e2d90.885d1a","order":3,"width":0,"height":0,"gtype":"gage","title":"Grupo 3","label":"NO/SI","format":"{{value}}","min":0,"max":"1","colors":["#00b500","#e6e600","#ca3838"],"seg1":"","seg2":"","x":620,"y":2240,"wires":[]},{"id":"3b520cf9.e7f1fc","type":"ui_gauge","z":"89502d50.a538e","name":"Grupo Shield","group":"258e2d90.885d1a","order":4,"width":0,"height":0,"gtype":"gage","title":"Grupo Shield","label":"NO/SI","format":"{{value}}","min":0,"max":"1","colors":["#00b500","#e6e600","#ca3838"],"seg1":"","seg2":"","x":630,"y":2360,"wires":[]},{"id":"d61be05f.7b79e8","type":"mqtt in","z":"89502d50.a538e","name":"Calidad de Aire Shield","topic":"/Sensor/GrupoShield/Calidad_Aire","qos":"2","datatype":"json","broker":"e67338bd.c1a978","x":320,"y":2520,"wires":[["e6f38b17.cac01","443afcb8.001834"]]},{"id":"443afcb8.001834","type":"ui_gauge","z":"89502d50.a538e","name":"Calidad de Aire Shield ","group":"e91caabd.310a28","order":0,"width":0,"height":0,"gtype":"gage","title":"Calidad de Aire Shield ","label":"Muy Buena / Muy Mala","format":"{{value}}","min":"1","max":"4","colors":["#00b500","#e6e600","#ca3838"],"seg1":"","seg2":"","x":880,"y":2520,"wires":[]},{"id":"e6f38b17.cac01","type":"function","z":"89502d50.a538e","name":"Guardar Variable","func":"global.set(\"MS34\",msg.payload);","outputs":1,"noerr":0,"x":610,"y":2600,"wires":[[]]},{"id":"fbc8daa.4aca228","type":"mqtt in","z":"89502d50.a538e","name":"Movimiento Shield 2","topic":"/Sensor/GrupoShield/Movimiento/2","qos":"2","datatype":"utf8","broker":"e67338bd.c1a978","x":370,"y":2420,"wires":[["1e4e75db.43594a"]]},{"id":"7aa5fc32.93a7a4","type":"mqtt in","z":"89502d50.a538e","name":"Movimiento 33","topic":"/Sensor/Grupo3/Movimiento/33","qos":"2","datatype":"utf8","broker":"e67338bd.c1a978","x":360,"y":2300,"wires":[["cac96c9d.96a7"]]},{"id":"98e6c1ba.e7bc08","type":"mqtt in","z":"89502d50.a538e","name":"Movimiento 22","topic":"/Sensor/Grupo2/Movimiento/22","qos":"2","datatype":"utf8","broker":"e67338bd.c1a978","x":360,"y":2180,"wires":[["ee3360d4.84067"]]},{"id":"fd88c88e.54cea8","type":"mqtt in","z":"89502d50.a538e","name":"Movimiento 11","topic":"/Sensor/Grupo1/Movimiento/11","qos":"2","datatype":"utf8","broker":"e67338bd.c1a978","x":350,"y":2060,"wires":[["96ba13de.870f1"]]},{"id":"b6df9849.2ee1a8","type":"ui_group","z":"","name":"Grupo 2 + Grupo SHIELD","tab":"13f6d20a.ff5cde","order":3,"disp":true,"width":"14","collapse":true},{"id":"e67338bd.c1a978","type":"mqtt-broker","z":"","name":"Mosquitto","broker":"192.168.43.183","port":"1883","clientid":"","usetls":false,"compatmode":true,"keepalive":"60","cleansession":true,"birthTopic":"","birthQos":"0","birthPayload":"","closeTopic":"","closeQos":"0","closePayload":"","willTopic":"","willQos":"0","willPayload":""},{"id":"7391748f.f51ce4","type":"ui_group","z":"","name":"Grupo 1 + Grupo 3","tab":"13f6d20a.ff5cde","disp":true,"width":"14","collapse":true},{"id":"2a744fe7.fb3e18","type":"ui_group","z":"","name":"Humedad Media Aula 352","tab":"416426ef.928fe8","disp":true,"width":"8","collapse":true},{"id":"420c050c.649c64","type":"ui_group","z":"","name":"Temperatura Media Aula 352","tab":"416426ef.928fe8","order":2,"disp":true,"width":"8","collapse":true},{"id":"f33d3453.29001","type":"ui_group","z":"","name":"Luz Media Aula 352","tab":"416426ef.928fe8","disp":true,"width":"8","collapse":true},{"id":"f3f6bec.a1c66c","type":"ui_group","z":"","name":"CO2 Media Aula 352","tab":"416426ef.928fe8","disp":true,"width":"5","collapse":true},{"id":"a80e4fd6.1b3bd","type":"ui_group","z":"","name":"TVOC Media Aula 352","tab":"416426ef.928fe8","order":2,"disp":true,"width":"5","collapse":true},{"id":"45dcf39f.70f49c","type":"ui_group","z":"","name":"Mapa","tab":"416426ef.928fe8","disp":true,"width":"6","collapse":true},{"id":"258e2d90.885d1a","type":"ui_group","z":"","name":"Movimiento","tab":"416426ef.928fe8","order":5,"disp":true,"width":"6","collapse":true},{"id":"e91caabd.310a28","type":"ui_group","z":"","name":"Calidad de Aire","tab":"416426ef.928fe8","order":2,"disp":true,"width":"4","collapse":true},{"id":"13f6d20a.ff5cde","type":"ui_tab","z":"","name":"AULA 352","icon":"dashboard","disabled":false,"hidden":false},{"id":"416426ef.928fe8","type":"ui_tab","z":"","name":"Panel de Control","icon":"dashboard","order":2,"disabled":false,"hidden":false}]