



Universidad de  
Oviedo



# **ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN**

## **MÁSTER UNIVERSITARIO EN INGENIERÍA INDUSTRIAL**

### **ÁREA DE TECNOLOGÍA ELECTRÓNICA**

#### **SISTEMA DE MONITORIZACIÓN Y CONTROL DE LA CALIDAD DEL AIRE DE UN LOCAL MEDIANTE SENSORES INALÁMBRICOS Y COMUNICACIONES VÍA MQTT CON PLATAFORMA IoT**

**D. SALGUEIRO LLAVE, Marcos**

**TUTOR: D. QUINTANA BARCIA, Pablo José  
COTUTOR: D. RICO SECADES, Manuel**

**FECHA: JUNIO 2021**



## **Agradecimientos**

---

Quiero empezar este agradecimiento por mis dos tutores, Pablo José Quintana Barcia y Manuel Rico Secades, sin ellos no sé si hubiera llegado a las mismas soluciones, pero estoy seguro de que el camino hasta llegar a ellas hubiese sido diferente, y por supuesto no tan gratificante. Han estado atentos, pero no agobiantes, lo que es de agradecer. Guardaré un gran recuerdo de esta singular pareja de tutores.

Además, aprovechando que este trabajo de fin de máster me da la posibilidad de hacerlo, me gustaría agradecer a todas las personas que he conocido, ya muchos de ellos puedo catalogar de amigos, con las que me he divertido y que me han ayudado a llegar hasta la meta de este arduo periplo universitario, no me querría dejar a nadie, pero ellos bien saben quiénes son.

Por último, el agradecimiento más especial para mi familia, a mis padres que han estado día tras día animándome y apoyándome hasta terminar esta etapa universitaria que tanto esfuerzo a costado culminar. Simplemente gracias.

## Resumen

---

El auge de la tecnología “Internet of Things” (IoT), o internet de las cosas, en castellano, está permitiendo la interconexión de cualquier dispositivo electrónico entre sí, vía internet, instalado en cualquier objeto que se pueda imaginar. Aprovechando esta tecnología se ha diseñado para este trabajo de fin de máster la monitorización de la calidad del aire de una estancia mediante la regulación del caudal de aire que se introduce.

El trabajo consistirá, por un lado, en un cálculo fluidodinámico, para determinar el caudal de aire en función de los niveles de concentración de CO<sub>2</sub> de la estancia, así como diversos cálculos que permitirán la selección de los equipos de ventilación. Por otro lado, se realizarán una serie de comunicaciones entre los dispositivos sensorios que se situarán en la estancia y los dispositivos de actuación empleados para regular la velocidad de giro del motor del ventilador, situados en los equipos de ventilación que se instalen. Para el control de estas comunicaciones se diseñarán dos aplicaciones, la primera de ellas para la monitorización de la calidad del aire y, la segunda, empleada para la gestión del intercambio de información entre los diferentes dispositivos electrónicos utilizados.

## **Abstract**

---

The rise of the technology "Internet of Things" (IoT), or Internet of Things, in Spanish, is allowing the interconnection of any electronic device with each other, via the Internet, installed in any object you can imagine. Taking advantage of this technology has been designed for this final work master monitoring the air quality of a room by regulating the air flow that is introduced.

The work will consist, on the one hand of a fluid dynamic calculation to determine the air flow rate according to the CO<sub>2</sub> concentration levels of the room, as well as various calculations that will allow the selection of ventilation equipment. In addition, a series of communications shall be made between the sensory devices to be in the room and the actuating devices used to regulate the motor speed of the fan, located in the ventilation equipment to be installed. For the control of these communications two applications will be designed, the first of them for the monitoring of the quality of the air and, the second, used for the management of the exchange of information between the different electronic devices used.

## Índice general

|  |      |
|--|------|
| Índice de tablas.....                                      | iii  |
| Índice de figuras.....                                     | iv   |
| Índice de ecuaciones.....                                  | vii  |
| Lista de acrónimos.....                                    | viii |
| 1. Introducción y objetivos.....                           | 1    |
| 1.1. Presentación y justificación del proyecto.....        | 1    |
| 1.2. Objetivos.....  | 2    |
| 1.3. Estructura del proyecto.....                          | 3    |
| 2. Normativa.....  | 6    |
| 3. Contextualización y descripción de la estancia.....     | 8    |
| 3.1. Qué es IoT e Industria 4.0.....                       | 8    |
| 3.2. Estancia objeto de estudio.....                       | 10   |
| 4. Ventilación.....  | 14   |
| 4.1. Calidad del aire exterior.....                        | 15   |
| 4.2. Calidad del aire interior.....                        | 16   |
| 4.3. Cálculo de la ventilación.....                        | 16   |
| 4.3.1. Caudal de ventilación.....                          | 17   |
| 4.3.2. Presiones y pérdidas de carga en los conductos..... | 18   |
| 4.3.3. Potencia específica de ventilación.....             | 21   |
| 4.4. Unidad de tratamiento de aire.....                    | 23   |
| 4.4.1. Filtros.....  | 25   |
| 4.4.2. Recuperador de calor.....                           | 27   |
| 4.4.3. Ventiladores.....                                   | 28   |
| 4.5. Difusores y rejillas.....                             | 30   |
| 5. Sistema de adquisición de datos.....                    | 32   |
| 5.1. Elección del microcontrolador.....                    | 33   |
| 5.2. Placa de desarrollo.....                              | 35   |
| 5.3. Bus I <sup>2</sup> C.....                             | 37   |
| 5.4. Sensores disponibles en el mercado.....               | 38   |
| 5.4.1. Sensor MG811.....                                   | 39   |
| 5.4.2. Sensor SGP30.....                                   | 39   |
| 5.4.3. Sensor CCS811.....                                  | 39   |
| 5.4.4. Sensor Si7021.....                                  | 40   |
| 5.4.5. Sensor DHT22.....                                   | 40   |
| 5.4.6. Sensor BMP280.....                                  | 41   |
| 5.5. Módulos de sensores comerciales.....                  | 41   |

---

## ÍNDICE GENERAL

---

|   |    |
|---|----|
| 5.5.1. Módulo con sensores CCS811, Si7021 y BMP280.....                       | 41 |
| 5.5.2. Módulo SCD30 .....   | 42 |
| 5.6. Equipos comerciales de monitorización de la calidad del aire .....       | 42 |
| 5.6.1. Equipo de monitorización de aire Nanoenvi IAQ.....                     | 42 |
| 5.7. Selección de los sensores.....   | 43 |
| 5.8. Conexión entre el SCD30 y el NodeMCU.....                                | 45 |
| 5.8.1. Direccionamiento de la información del protocolo I <sup>2</sup> C..... | 46 |
| 6. Comunicación entre sistemas.....   | 48 |
| 6.1. Protocolo de comunicación.....   | 49 |
| 6.1.1. LoRaWAN .....  | 49 |
| 6.1.2. ZigBee.....  | 51 |
| 6.2. Message Queue Telemetry Transport.....                                   | 52 |
| 6.2.1. Bróker.....  | 54 |
| 6.3. Arduino.....   | 55 |
| 6.4. Aplicación con Arduino.....  | 56 |
| 6.4.1. Librerías .....  | 57 |
| 6.4.2. Variables globales y prototipos de las funciones .....                 | 59 |
| 6.4.3. Setup.....   | 64 |
| 6.4.4. Programa principal .....   | 65 |
| 7. Sistemas de gestión y actuación.....                                       | 67 |
| 7.1. Sistema de gestión .....   | 67 |
| 7.1.1. Librerías .....  | 67 |
| 7.1.2. Variables .....  | 68 |
| 7.1.3. Funciones .....  | 68 |
| 7.1.4. Programa principal.....  | 74 |
| 7.1.5. Interfaz de la aplicación .....  | 74 |
| 7.2. Sistema de actuación.....  | 76 |
| 7.2.1. Protocolo Modbus .....   | 76 |
| 7.2.2. Variador de frecuencia .....   | 77 |
| 8. Conclusiones y trabajos futuros .....                                      | 80 |
| 8.1. Conclusiones.....  | 80 |
| 8.2. Trabajos futuros .....   | 81 |
| 9. Bibliografía .....   | 83 |
| Anexo A .....   | 92 |
| Anexo B .....   | 98 |

## Índice de tablas

|  |    |
|--|----|
| Tabla 2.1.- Nomenclatura y título de las normas del proyecto.....                      | 7  |
| Tabla 4.1.- Pérdidas de carga máxima para cada equipo del sistema de ventilación. .... | 19 |
| Tabla 4.2.- Resumen de las presiones en la red de conductos.....                       | 21 |
| Tabla 4.3.- Cálculo de las potencias a caudal máximo. ....                             | 23 |
| Tabla 4.4.- Categoría de los ventiladores del sistema de ventilación.....              | 23 |
| Tabla 4.5.- Correspondencia entre norma UNE EN 16980 y norma UNE EN 779. ....          | 26 |
| Tabla 5.1.- Función de cada pin del NodeMCU utilizado para el proyecto. ....           | 37 |
| Tabla 5.2.- Nomenclatura y funcionalidad de cada pin del módulo SCD30.....             | 44 |



## Índice de figuras

|  |    |
|--|----|
| Figura 3.1.- Internet of Things (imagen obtenida bajo licencia “Creative Commons” de la web [6]).....                      | 9  |
| Figura 3.2.- Revoluciones industriales (imagen obtenida bajo licencia “Creative Commons” de la web [10])......             | 10 |
| Figura 3.3.- Edificio Polivalente del Campus Universitario de Viesques (imagen obtenida de la web [11])......              | 10 |
| Figura 3.4.- Ubicación del salón de actos en la segunda planta del edificio (imagen obtenida de la web [12])......         | 11 |
| Figura 3.5.- Croquis del salón de actos. ....  | 12 |
| Figura 3.6.- Conferencia en el salón de actos del edificio polivalente (imagen obtenida de la web del Comercio [13])...... | 13 |
| Figura 4.1.- Punto de funcionamiento del ventilador de impulsión [19].....   | 22 |
| Figura 4.2.- Punto de funcionamiento del ventilador de extracción [20]......   | 22 |
| Figura 4.3.- Esquema de la UTA del proyecto.....   | 25 |
| Figura 4.4.- Prefiltro Bio Bag 60 (imagen obtenida de la web de Venfilter [24]). ....                                      | 27 |
| Figura 4.5.- Filtro Bag-glass 80 (imagen obtenida de la web de Venfilter [25]). ....                                       | 27 |
| Figura 4.6.- Recuperador de calor de placas de SystemAir (imagen obtenida de su catálogo [27])<br>.....                    | 28 |
| Figura 4.7.- Ventilador centrífugo BVC-M 15/15 1,5 kW 950 de Casals (imagen obtenida su web [19])......                    | 29 |
| Figura 4.8.- Ventilador centrífugo BSTR-M 22/11 1,1 kW 550 de Casals (imagen obtenida de su web [20])......                | 30 |
| Figura 4.9.- Difusor de tobera WDA-R-175 de Schako (imagen obtenida de la web de Schako [29])......                        | 30 |
| Figura 4.10.- Rejilla de extracción E-LO/AD de Euroclima (imagen obtenida de la web de Euroclima [31]). ....               | 31 |
| Figura 5.1. Módulo del microcontrolador ESP8266. ....  | 34 |
| Figura 5.2.- Esquema del microcontrolador ESP8266 (imagen obtenida del “datasheet” del microcontrolador [39])......        | 35 |
| Figura 5.3.- Placa de desarrollo NodeMCU con microcontrolador ESP8266.....   | 35 |
| Figura 5.4.- Pines del NodeMCU.....  | 36 |
| Figura 5.5.- Arquitectura del bus I2C. ....  | 37 |
| Figura 5.6.- Ejemplo de comunicación maestro-esclavo en el bus I2C. ....   | 38 |
| Figura 5.7.- Sensor MG811 para mediciones de CO2 (imagen obtenida del “datasheet” del sensor [47])......                   | 39 |
| Figura 5.8.- Sensor SGP30 para mediciones de CO2 y VOC (imagen obtenida del “datasheet” del sensor [48])......             | 39 |
| Figura 5.9.- Sensor CCS811 para mediciones de CO2 y VOC (imagen obtenida del “datasheet” del sensor [49])......            | 40 |

## ÍNDICE DE FIGURAS

---

|  |    |
|--|----|
| Figura 5.10.- Sensor Si7021 para mediciones de temperatura y humedad relativa (imagen obtenida del “datasheet” del sensor [50]).               | 40 |
| Figura 5.11.- Sensor de temperatura y humedad relativa DHT22 (imagen obtenida de una web especializada [52]).                                  | 40 |
| Figura 5.12.- Sensor de presión y temperatura BMP280 (imagen obtenida del “datasheet” del sensor [53]).  | 41 |
| Figura 5.13.- Módulo de sensores BMP280, CCS811 y Si7021 (imagen obtenida de la web de Aliexpress [54]).                                       | 41 |
| Figura 5.14.- Módulo SCD30 para medición de CO2, humedad relativa y temperatura (imagen obtenida del “datasheet” del módulo de sensores [55]). | 42 |
| Figura 5.15.- Equipo Nanoenvi IAQ para la monitorización de la calidad del aire (imagen obtenida de la web del fabricante [56]).               | 43 |
| □Figura 5.16.- Esquema y pines del módulo SCD30 (esquema obtenido del “datasheet” de este módulo [55]).  | 44 |
| Figura 5.17.- Conexiones entre el NodeMCU y el módulo SCD30.   | 45 |
| Figura 5.18.- Flujograma de lectura y envío de datos del módulo de sensores al microcontrolador.   | 47 |
| Figura 5.19.- Ejemplo de almacenamiento de datos en el registro (imagen obtenida del “interface description” del sensor [59]).                 | 47 |
| Figura 6.1. Protocolo TCP/IP.  | 49 |
| Figura 6.2.- Protocolo de comunicación LoRaWAN.  | 50 |
| Figura 6.3. Funcionamiento de LoRaWAN con sensores y actuadores.   | 50 |
| Figura 6.4. Protocolo de comunicación ZigBee.  | 51 |
| Figura 6.5.- Estructura del protocolo MQTT.  | 53 |
| Figura 6.6.- Estructura de los mensajes del protocolo MQTT.  | 53 |
| Figura 6.7.- Bróker MQTT.  | 54 |
| Figura 6.8.- Bróker utilizado en el proyecto.  | 55 |
| Figura 6.9.- Arduino IDE.  | 55 |
| Figura 6.10.- Librerías y certificado de la aplicación en Arduino.   | 58 |
| Figura 6.11.- Certificado de encriptación TLS.   | 58 |
| Figura 6.12.- Creación del cliente MQTT seguro que cumple con el certificado.  | 58 |
| Figura 6.13.- Objeto y variables para la medición de los parámetros ambientales.   | 59 |
| Figura 6.14.- Creación de variables para el protocolo MQTT.  | 60 |
| Figura 6.15.- Función para la conexión WiFi.   | 60 |
| Figura 6.16.- Función para la conexión segura con el bróker.   | 61 |
| Figura 6.17.- Función de verificación de la conexión con TLS.  | 61 |
| Figura 6.18.- Función para la conexión del cliente con el bróker.  | 62 |
| Figura 6.19.- Función para la medición y el almacenamiento de los niveles de CO2 en la estancia.   | 62 |
| Figura 6.20.- Función para la medición y el almacenamiento de la temperatura en la estancia.   | 62 |

---

## ÍNDICE DE FIGURAS

---

|   |    |
|---|----|
| Figura 6.21.- Función para la medición y el almacenamiento de la humedad relativa en la estancia.....                 | 63 |
| Figura 6.22.- Función para la creación del mensaje que se publicará en el bróker.....                                 | 63 |
| Figura 6.23.- Función para la publicación del mensaje en el bróker bajo el correspondiente “topic”.....               | 64 |
| Figura 6.24.- Configuración de la velocidad de subida, la conexión I2C y la autocalibración del módulo.....           | 64 |
| Figura 6.25.- Conexión del microcontrolador a la red WiFi y creación del cliente MQTT seguro.....                     | 65 |
| Figura 6.26.- Comprobación del estado de la conexión cliente-bróker.....  | 65 |
| Figura 6.27.- Recogida de los datos de medición en variables, generación del mensaje y subrutina de espera.....       | 66 |
| Figura 6.28.- Publicación del mensaje en el bróker.....   | 66 |
| Figura 7.1.- Librerías de la aplicación del sistema de gestión.....   | 68 |
| Figura 7.2.- IP del bróker, creación de cliente seguro, variable auxiliar y “topics” para suscribirse y publicar..... | 68 |
| Figura 7.3.- Función para la conexión del cliente MQTT con el bróker.....   | 69 |
| Figura 7.4.- Función para la desconexión del cliente MQTT con el bróker.....  | 69 |
| Figura 7.5.- Suscripción del cliente MQTT al “topic”.....   | 70 |
| Figura 7.6.- Desuscripción del cliente MQTT del “topic”.....  | 70 |
| Figura 7.7.- Recogida de datos y llamamiento a funciones de almacenamiento, interpretación y publicación.....         | 71 |
| Figura 7.8.- Agregación en base de datos de los parámetros ambientales medidos.....                                   | 72 |
| Figura 7.9.- Conexión con base de datos, creación de ésta en caso de que no existiera.....                            | 72 |
| Figura 7.10.- Instrucción ordenada por el sistema de gestión para el de actuación.....                                | 73 |
| Figura 7.11.- Acciones que puede ordenar el sistema de gestión.....   | 73 |
| Figura 7.12.- Publicación de la orden en el bróker.....   | 74 |
| Figura 7.13.- Rutina de “callback” tras la recepción del mensaje.....   | 74 |
| Figura 7.14.- Interfaz gráfica de la aplicación.....  | 75 |
| Figura 7.15.- Empleo habitual de los modos operativos en el protocolo Modbus.....                                     | 77 |
| Figura 7.16.- Esquema de funcionamiento de un variador de frecuencia.....   | 77 |
| Figura 7.17.- Diagrama de un variador de frecuencia.....  | 78 |
| Figura 7.18.- Variador de frecuencia modelo ACH480 de ABB (imagen obtenida del catálogo de ABB [78]).....             | 79 |

## Índice de ecuaciones

|                    |    |
|--------------------|----|
| Ecuación 4.1.....  | 17 |
| Ecuación 4.2.....  | 17 |
| Ecuación 4.3.....  | 17 |
| Ecuación 4.4.....  | 18 |
| Ecuación 4.5.....  | 18 |
| Ecuación 4.6.....  | 19 |
| Ecuación 4.7.....  | 19 |
| Ecuación 4.8.....  | 20 |
| Ecuación 4.9.....  | 20 |
| Ecuación 4.10..... | 21 |
| Ecuación 4.11..... | 22 |
| Ecuación 4.12..... | 23 |

## Lista de acrónimos

|                  |   |
|------------------|---|
| ACK              | “Acknowledgement” (Acuse de recibo)   |
| $\Delta P_p$     | Pérdidas de carga primarias en los conductos, Pa                              |
| $\Delta P_T$     | Variación de presión total, Pa  |
| $C_{h,ext}$      | Concentración de CO <sub>2</sub> en el exterior del recinto, ppm              |
| $C_{h,int}$      | Concentración de CO <sub>2</sub> en el interior de la estancia, ppm           |
| CISC             | “Complex Instruction Set Computer”  |
| CRC              | “Cyclic Redundancy Check”   |
| CSS              | “Chirp Spread Spectrum”   |
| D                | Diámetro del conducto de ventilación, mm                                      |
| $\varepsilon$    | Rugosidad de la pared del conducto de ventilación, mm                         |
| $\varepsilon_v$  | Efectividad de la ventilación   |
| f                | Factor de fricción  |
| $G_h$            | Carga de CO <sub>2</sub> debida a los ocupantes presentes en la estancia, l/h |
| GPIO             | “General Purpose Input/Output” (Entrada/Salida de propósito general)          |
| HVAC             | “Heating, Ventilating and Air Conditioning”                                   |
| IDAE             | Instituto para la Diversificación y el Ahorro de la Energía                   |
| IDE              | “Integrated Development Enviroment” (Entorno de desarrollo integrado)         |
| IEEE             | “Institute of Electrical and Electronics Engineers”                           |
| I <sup>2</sup> C | “Inter-Integrated Circuit”  |
| IoT              | “Internet of Things” (Internet de las cosas)                                  |
| IP               | “Internet Protocol” (Protocolo de Internet)                                   |
| L                | Longitud del conducto de ventilación, m                                       |
| LLSB             | “Least Least Significant Bit”   |
| LMSB             | “Least Most Significant Bit”  |
| $L_w$            | Potencia sonora, dB   |
| M2M              | “Machine to Machine”  |
| MLSB             | “Most Least Significant Bit”  |
| MMSB             | “Most Most Significant Bit”   |
| MQTT             | “Message Queue Telemetry Transport”   |
| $\mu$            | Viscosidad dinámica del fluido, Pa s  |
| OASIS            | “Organization for the Advancement of Structured Information Standards”        |
| PCB              | “Printed Circuit Board” (Placa de circuito impreso)                           |
| $p_d$            | Presión dinámica, Pa  |
| $\dot{P}_f$      | Potencia que se ha de suministrar al fluido, W                                |

---

## LISTA DE ACRÓNIMOS

---

|                  |   |
|------------------|---|
| PLC              | “Programmable Logic Controller” (Controlador lógico programable)      |
| PM <sub>x</sub>  | Partículas en suspensión, ppm o ppb                                   |
| Ė <sub>v</sub>   | Potencia consumida por el ventilador, W                               |
| PWM              | “Pulse Width Modulation” (Modulación por Ancho de Pulso)              |
| QoS              | “Quality of Service” (Calidad del Servicio)                           |
| Q <sub>v</sub>   | Caudal de ventilación, m <sup>3</sup> /h                              |
| R                | “Read”  |
| η                | Rendimiento de la ventilación   |
| Re               | Número de Reynolds  |
| RISC             | “Reduced Instruction Set Computer”                                    |
| RITE             | Reglamento de Instalaciones Térmicas en los Edificios                 |
| ρ                | Densidad del aire, kg/m <sup>3</sup>                                  |
| S                | Sección del conducto de ventilación, m <sup>2</sup>                   |
| SCL              | “Serial Clock” (Señal de reloj)                                       |
| SDA              | “Serial Data” (Dato a transmitir)                                     |
| SoC              | “System on a chip”  |
| SPI              | “Serial Peripheral Interface”   |
| TCP              | “Transmission Control Protocol” (Protocolo de Control de Transmisión) |
| TLS              | “Transport Layer Security”  |
| UTA              | Unidad de Tratamiento de Aire   |
| V                | Velocidad media del fluido, m/s                                       |
| VOCs             | Compuestos orgánicos volátiles en el aire, ppb                        |
| W                | “Write”   |
| W <sub>esp</sub> | Potencia específica del sistema de ventilación, W/(m <sup>3</sup> /s) |



# 1. Introducción y objetivos

## 1.1. Presentación y justificación del proyecto

La idea de este trabajo de fin de máster nace de la actual crisis sanitaria ocasionada por el virus SARS-Cov-2, conocido popularmente como COVID-19 o Coronavirus, debido a las nuevas medidas que se han tomado en el ámbito de la ventilación de espacios cerrados. El control de la calidad del aire en este tipo de espacios es un tema que concierne especialmente a las autoridades sanitarias. Este control es el punto de partida de este proyecto.

A esta idea se le ha sumado, la tendencia que existe actualmente a interconectar digitalmente todo tipo de objetos con internet. Este concepto se conoce como “Internet of Things” (IoT) o Internet de las cosas, en castellano, y se basa en una comunicación inalámbrica entre dispositivos electrónicos. En los hogares ya se tiene la posibilidad de conectar dispositivos como enchufes o lámparas, entre sí, gracias a la nube. Sin embargo, esta tecnología no es exclusiva a los aparatos domésticos. En la industria también se emplea este concepto y se interconectan todos los elementos para conseguir un control que abarca desde recogida de datos hasta las ordenes de actuación. Dicho control se logra cuando todos los elementos que forman parte de un determinado proceso industrial se pueden comunicar entre ellos, empleando esta tecnología.

Juntando estas dos ideas, la ventilación de espacios cerrados y la tecnología IoT, se ha creado, para este proyecto, la monitorización y el control de una estancia en función de la calidad del aire que albergue en su interior. Para lograr aunar estas dos ideas se planifican una serie de objetivos, como se verá más adelante en este mismo capítulo.

La estancia escogida para llevar a cabo el proyecto será el salón de actos del edificio polivalente de la Escuela Politécnica de Ingeniería de Gijón. La elección de esta estancia es debida al conocimiento de las dimensiones y el aforo de esta, ya que esta información será crucial para el desarrollo del proyecto.

Los equipos de ventilación actuales permiten una regulación del caudal de aire con gran precisión, empleando un variador de frecuencia que actúa sobre el motor del ventilador modificando su velocidad de giro. En este proyecto se contará con tres subsistemas interconectados mediante comunicaciones inalámbricas: por un lado, estará el sistema de adquisición de datos, que enviará las medidas realizadas al sistema de gestión. Este sistema guardará la información recibida en una base de datos y tomará decisiones de actuación sobre el ventilador en función de las medidas tomadas. Por último, el tercero de los sistemas recibirá esta decisión y actuará sobre el ventilador en consecuencia.

La decisión que ha de tomar el sistema de gestión se realizará en función de los niveles de concentración del CO<sub>2</sub> que haya en el interior de la estancia en cada momento, no pudiendo superarse unos máximos estipulados por la normativa. Estos niveles de concentración van a ser uno de los parámetros ambientales medidos por el sistema de adquisición de datos. El resto de los parámetros analizados serán aquellos que sean relevantes para el bienestar térmico de los ocupantes de la estancia.

Como se ha comentado en los párrafos anteriores, la comunicación será inalámbrica, lo que significa que no habrá ningún tipo de cableado que interconecte los diferentes subsistemas que conforman el proyecto. En este caso el protocolo de comunicación que se ha elegido es el “Message Queue Telemetry Transport”, más conocido por sus siglas MQTT.

Hoy en día ya existen multitud de opciones para realizar comunicaciones inalámbricas, no obstante, el protocolo MQTT ofrece una serie de ventajas, que será explicadas a lo largo de este proyecto, que lo hacen idóneo para esta monitorización de la calidad del aire. Así mismo, se



---

## 1. INTRODUCCIÓN Y OBJETIVOS

---

verán en este documento otras alternativas igualmente válidas para realizar estas comunicaciones pero que se irán descartando por diversos motivos.

Este tipo de monitorización de la ventilación, e incluso la climatización, para grandes espacios cerrados está cada vez más extendido, ya que su instalación automatiza el control de la calidad del aire, actuando y avisando de las posibles incidencias que pudieran acaecer durante su funcionamiento. Además, la implementación de esta monitorización repercutirá en un ahorro energético y económico para sus usufructuarios. Es por ello lógico, que grandes empresas como Ikea recurran a esta solución para sus tiendas.

### 1.2. Objetivos

Los objetivos del proyecto estarán enfocados en dos grandes grupos, el primero de ellos será realizar el cálculo de la ventilación que sea necesaria en la estancia, mientras que el segundo consistirá en diseñar y conectar inalámbricamente los tres sistemas con los que contará la instalación.

#### I. Primer objetivo: Cálculo de la ventilación

El primero de estos objetivos será el cálculo de la ventilación necesaria para la estancia, con el que se podrán definir los equipos del sistema de ventilación que se habrán de instalar. Este cálculo ha de cumplir con una serie de requisitos técnicos comunes para cualquier tipo de instalaciones de ventilación, los cuales aparecen recogidos en el *Reglamento de Instalaciones Térmicas en los Edificios*, denominado habitualmente según sus siglas RITE [1].

El primer paso del cálculo será categorizar la calidad del aire que habrá de tenerse en el interior de la estancia, será función de su uso, así como el existente en área de Gijón dónde se halla la estancia, la zona de Viesques. Una vez categorizadas estas dos calidades ya se podrá realizar el cálculo de la ventilación propiamente dicho, empezando por el caudal de ventilación que se deberá suministrar a la estancia, éste será función de los niveles máximos de concentración del CO<sub>2</sub> que pueda albergar en su interior.

No obstante, para la selección de equipos, no será suficiente, únicamente, con el valor del caudal de aire, sino que será necesario calcular las pérdidas de carga en las que infiere la instalación de ventilación, estas van a ser debidas a los conductos y a los equipos de filtrado, intercambio de calor, difusores y rejillas que habrá que colocar para alcanzar la calidad de aire interior buscada.

Sin embargo, el RITE [1], exigirá una potencia específica que categorizará todo el sistema de ventilación, esta se obtendrá de la división entre las pérdidas de carga y el caudal de ventilación. Con el resultado de esta potencia se habrá completado el cálculo de la instalación, el cual era el primero de los objetivos del proyecto.

Finalizado el cálculo se seleccionarán los equipos de ventilación que deberían instalar para el correcto funcionamiento del sistema de ventilación.

#### II. Segundo objetivo: Diseño de los sistemas y su comunicación inalámbrica

El segundo de los objetivos de la instalación consiste en el diseño de los diferentes sistemas que harán posible la monitorización de la calidad del aire y la regulación del ventilador. A su vez, para cumplimentar el objetivo, se mostrará cómo se va a llevar a cabo la comunicación inalámbrica entre ellos mediante el protocolo MQTT.

El protocolo de comunicación MQTT será el que ayude a conectar entre sí todos los sistemas. Cada sistema será un cliente que se suscribirá a uno o varios “topics” para recibir y/o publicar información. El bróker MQTT se encargará de procesar todos estos mensajes que viajarán por la red. Se ha avanzado el comportamiento que empleará este protocolo de comunicación para entender mejor como se van a diseñar los sistemas. La descripción detallada

---

## 1. INTRODUCCIÓN Y OBJETIVOS

---

y contextualizada de este protocolo aparecerá en el subapartado 6.2 del Capítulo 6 del documento.

El primer sistema que se va a diseñar será el de adquisición de datos, este sistema deberá ser capaz de realizar las mediciones ambientales de la estancia y comunicarse mediante el protocolo MQTT con el bróker donde publicará los resultados. Para la toma de datos se emplearán una serie de sensores, estos una vez realizada la medición han de poder enviarla al microcontrolador que gestione este sistema cuando se lo solicite, para que éste a su vez pueda publicarlo en el bróker MQTT.

El segundo de estos sistemas será de gestión, encargado del procesamiento de los datos ambientales recogidos. Este sistema deberá estar suscrito al “topic” que contenga la información de los datos ambientales publicada desde el sistema de adquisición en el bróker MQTT. Una vez obtenida, realizará dos funciones, la primera de ellas será almacenar en una base de datos los valores de las mediciones junto con la fecha y hora en la que han tenido lugar, a modo de repositorio de incidencias. La segunda de estas funciones será la de interpretación de estos valores, fundamentándola en los máximos indicados por el RITE [1] de niveles de concentración de CO<sub>2</sub> para estancias similares al salón de actos objeto de estudio. La interpretación de los valores indicará que acción ha de acometerse sobre el sistema de actuación que controla el motor del ventilador. La acción que se decida ejercer se publicará de nuevo en el bróker para que el sistema de actuación pueda recibirla y obrar a consecuencia.

El tercero, y último, de los sistemas será el de actuación. Este sistema deberá ser capaz de leer la orden emitida desde el sistema de gestión y modificar la velocidad de giro del motor del ventilador en función de la orden leída. Para ello este sistema deberá estar suscrito a la información que haya publicado el sistema de gestión en el bróker MQTT. Finalmente, un variador de frecuencia se encargará de realizar esta modificación de la velocidad de giro.

Concluida la explicación del diseño que deberá implementarse en el sistema de actuación, quedan definidos los objetivos del proyecto. A continuación, se realizará un breve resumen de cada uno de los capítulos que conforman este documento, con los que se pretende alcanzar cada uno de estos objetivos.

### 1.3. Estructura del proyecto

Este proyecto se encuentra dividido en ocho capítulos, además de la bibliografía. Los tres primeros capítulos están dedicados a introducción, base normativa y contextualización del proyecto. En el cuarto capítulo se realizará el cálculo fluidodinámico de la instalación. En los capítulos quinto, sexto y séptimo se explicarán cada uno de los tres sistemas que llevarán a cabo la monitorización y las comunicaciones entre ellos mediante el protocolo MQTT. El octavo, y último, capítulo estará dedicado a las conclusiones del proyecto y a los trabajos futuros que deban efectuarse para conseguir implementarlo en la realidad.

El **Capítulo 1** está referido a la introducción del proyecto, como se ha visto en él aparece la idea que inspiró el proyecto, la justificación y los objetivos que deberán haberse llevado a cabo a la conclusión de éste.

El **Capítulo 2** corresponderá a la normativa en la cual está basado el cálculo de la ventilación del proyecto, como se ha visto anteriormente esta base técnica compete al RITE [1], para los requisitos que ha de cumplir la instalación, y a la Guía Técnica IDAE número 10, titulada *Selección de equipos de transporte de fluido*, para el procedimiento de cálculo de la instalación. No obstante, para determinadas operaciones se recurrirá a normativa más específica, la cual también aparecerá reflejada en este capítulo.

El **Capítulo 3** contextualizará el proyecto en el marco del IoT y describirá la estancia objeto de estudio. Al inicio del capítulo se realizará un breve resumen de los conceptos de IoT e Industria 4.0 tan boyantes en nuestros días. Seguidamente se contextualizará el proyecto en este marco del IoT, viendo la influencia que tienen las comunicaciones inalámbricas en la

monitorización de dispositivos. La segunda parte del capítulo estará dedicada a la descripción de la estancia donde se instalará el sistema de ventilación, se verán su ubicación, tanto del edificio en la ciudad de Gijón como de la estancia dentro del mismo, sus dimensiones básicas, superficie y altura de techo, su aforo, y, por último, su uso más habitual como sala de conferencias para diferentes actos de la escuela.

El **Capítulo 4** estará dedicado a la parte más técnica del proyecto, el cálculo de la ventilación. En este capítulo se verá el procedimiento de cálculo que se seguirá para obtener el caudal de ventilación, las pérdidas de carga de la instalación y la potencia específica con las que poder seleccionar los equipos de la instalación, así como categorizar la propia instalación, requisito obligatorio indicado por el RITE [1]. Este procedimiento de cálculo se iniciará con la identificación de la calidad del aire exterior en la zona de Gijón donde se encuentra la estancia. Proseguirá con la calidad del aire interior en ésta, que es función de su uso como salón de actos. Obtenidas las calidades del aire exterior e interior se realizará el cálculo fluidodinámico de la instalación, mediante el cual se espera lograr los valores del caudal de ventilación, pérdidas de carga y potencias de la instalación. La última parte del capítulo está destinada a la selección de los equipos de ventilación, en ella se elegirán los equipos que más se ajusten a los cálculos realizados, viéndose para varios de estos equipos algunas de las alternativas existentes en el mercado.

El **Capítulo 5** mostrará el diseño del primero de los tres sistemas empleados para la monitorización de la instalación, el sistema de adquisición de datos. En la primera parte del capítulo se seleccionarán los diferentes elementos que va a conformar el sistema. El primero de ellos será el microcontrolador encargado de la gestión de todo el sistema, deberá ser capaz de conectarse inalámbricamente mediante el protocolo MQTT con el bróker, para poder publicar en él, y a su vez solicitar los sensores periódicamente los datos ambientales. El segundo de los elementos será la placa de desarrollo que albergará, entre otros módulos, el microcontrolador, y que actuará de enlace físico entre éste y los sensores. El tercer elemento del sistema lo van a formar los sensores ambientales encargados de la toma de medidas, se realizará un amplio estudio de mercado para seleccionar los más adecuados para la instalación. Finalmente, se verá cómo se llevará a cabo la conexión física entre la placa de desarrollo y los sensores, así como la comunicación mediante el protocolo I<sup>2</sup>C, “Inter-Integrated Circuit”, entre los sensores y el microcontrolador.

El **Capítulo 6** se dedicará a la comunicación entre los tres sistemas de la instalación. En primer lugar, se explicará lo que es un protocolo de comunicación y se verán dos opciones existentes en el mercado con las que se podrían intercomunicar los sistemas. A continuación, ya se comenzará a describir y desarrollar el protocolo de comunicación MQTT, que se empleará para las aplicaciones que se realizarán, comentando sus características del protocolo y sus ventajas. Seguidamente se explicará su funcionamiento, mediante clientes que podrán estar suscritos a la información o publicarla o incluso ambas opciones, y cuál es la estructura de los mensajes que son enviados mediante este protocolo. El siguiente punto del capítulo se centrará en el funcionamiento el nodo central o bróker MQTT, que actúa como intermediario de los mensajes, y cómo será realizada esta comunicación, es decir, que sistemas publican la información y cuales están suscritos a ella. Para finalizar el capítulo se documentará la aplicación, en el programa Arduino, que se creará para la publicación de los mensajes, con los datos ambientales, en el bróker desde el sistema de adquisición de datos, que ejercerá como cliente del protocolo. Dicha aplicación deberá ser capaz de recoger los datos del sensor, crear un cliente MQTT, conectarse y publicar los datos en el bróker. La documentación de esta aplicación estará enfocada en cada una de las zonas que la compondrán: librerías, variables globales y prototipos de las funciones, configuración y el programa principal.

El **Capítulo 7** estará dedicado a los dos últimos sistemas, el de gestión y el de actuación. El primero de ellos se explicará a través de la documentación de la aplicación en Python diseñada para tal fin. Las funciones del sistema de gestión, como se ha visto en los

---

## 1. INTRODUCCIÓN Y OBJETIVOS

---

objetivos, consistirán en almacenar en una base de datos los valores de las mediciones, realizadas por el sistema de adquisición, e interpretar esos valores para emitir la orden correspondiente al sistema de actuación. Para lograrlo la aplicación deberá conectarse con el bróker MQTT, crear un cliente que se suscriba a la información publicada por el sistema de adquisición de datos. Una vez recibido el mensaje, almacenará los datos en la base de datos, indicando la fecha y la hora a la que han sido tomados, y, con esos mismos datos, leerlos e interpretarlos para emitir la orden. La publicación de esta orden en el bróker requerirá que el cliente MQTT además de suscriptor sea publicador de información, por lo que así se hará. El otro sistema, el de actuación, no se creará para él ninguna aplicación que lo gestione y, únicamente, se describirán los dos elementos que lo formarán y la comunicación entre ellos. El primero de ellos será la placa de desarrollo con el microcontrolador, que coincidirá con la del sistema de adquisición de datos. El segundo elemento será el variador de frecuencia que se deberá encargar de proporcionar el nivel de tensión adecuada al motor del ventilador en cada momento para así controlar la velocidad de giro de éste. Además, para finalizar el capítulo, se verá el protocolo de comunicación que se emplea entre estos dos elementos, en este caso será el protocolo Modbus, debido a las características propias que incorpora.

El **Capítulo 8** será el capítulo final del proyecto, en él se discutirán las conclusiones que se han sacado del proyecto y se verá si se han alcanzado los objetivos propuestos al inicio. A su vez, en la segunda parte de este capítulo se engloban los trabajos futuros que habrían de hacerse para poder hacer realidad esta instalación. Dichos trabajos futuros competarán al estudio detallado del salón de actos, que no se ha podido realizar, la compra de todos los equipos de ventilación, así como un pliego de condiciones para su futura instalación, los sensores, el variador de frecuencia y los buses de comunicación. Una vez realizada esta compra será más fácil ajustar los programas creados a la realidad. Y finalmente, la configuración, pruebas y puesta en servicio de toda la instalación.

## 2. Normativa

En este capítulo dedicado a la normativa se comentarán cuáles van a ser las normas principales que se seguirán en el desarrollo del proyecto. Así mismo, se mencionarán el resto de las normas secundarias, pero necesarias, para la confección de este. Cabe anunciar que la normativa utilizada compete al cálculo de la ventilación, Capítulo 4, ya que los sistemas y los protocolos de comunicación que se estudiarán no están recogidos en ninguna normativa actualmente.

La base del cálculo de ventilación será RITE [1]. Este documento establece las condiciones que deben cumplir las instalaciones destinadas a atender la demanda de bienestar térmico e higiene a través de las instalaciones de calefacción, climatización y agua caliente sanitaria, para conseguir un uso racional de la energía [2]. Por lo tanto, en él se recogen los requisitos que deberá cumplir la instalación del proyecto.

Los requisitos del RITE que debe cumplir el cálculo de la instalación serán, en un primer momento, la categorización de la calidad del aire tanto exterior como interior de la estancia, en función de los datos recogidos por las estaciones de medición de los ayuntamientos, para el aire exterior, y en función del uso que tenga la estancia para el aire interior. A su vez, indica también el límite máximo del nivel de concentración de CO<sub>2</sub> que pueda albergar dicha estancia. Además, una vez finalizado el cálculo, establece la categorización del sistema de ventilación en función de su potencia específica.

Por otro lado, para la selección de los equipos que compondrán el sistema de ventilación, el RITE establece unos valores máximos de pérdidas de carga para cada uno de ellos [1]. A su vez, para los algunos equipos ofrece una serie de criterios que habrán de cumplirse en la instalación. Estos criterios competen a los filtros y al recuperador de calor.

Del primero de ellos, los filtros, el RITE establece la clase de filtro que será necesario instalar en función de la categoría de la calidad de aire interior y exterior. A su vez, si el sistema de ventilación incorpora un recuperador de calor el RITE exige la instalación un prefiltro antes del este.

Del segundo de ellos, el recuperador de calor, el RITE establece su obligatoria instalación si se superan los 0,5 m<sup>3</sup>/s de caudal de ventilación, en el caso del proyecto como se calculará en el subapartado 4.3, si se superará dicho valor, por lo que su instalación será obligatoria. También en función de las horas de funcionamiento del recuperador y el caudal de ventilación el RITE establece la eficiencia mínima y la máxima pérdida de presión de este equipo.

Una vez vista la base normativa del proyecto, se pasará a comentar la guía técnica que se seguirá para realizar el procedimiento de cálculo de la ventilación en el proyecto. Esta guía técnica es la Guía Técnica IDAE número 10, titulada *Selección de equipos de transporte de fluido* [3], en ella se explica todo el procedimiento de cálculo desde el dimensionamiento de los conductos hasta la potencia que será necesaria entregar al fluido, pasando por todas las pérdidas de carga que haya en la instalación. No obstante, el cálculo del caudal de ventilación se realiza gracias a otra guía técnica, en este caso la Guía Técnica IDAE número 17, titulada *Instalaciones de climatización con equipos autónomos* [4].

A continuación, en la Tabla 2.1 se muestran el resto de las normas que se van a considerar para el cálculo de ventilación. Las normas que formen parte del RITE o de una guía técnica no se incluirán, ya que se incluye la guía que las alberga, así como las normas que excedan del cálculo de la ventilación, como pueden ser normas específicas de los equipos de la instalación.

## 2. NORMATIVA

| Nomenclatura            | Año  | Título de la norma  |
|-------------------------|------|---|
| Guía Técnica IDAE nº 9  | 2012 | Ahorro y recuperación de energía en instalaciones de climatización.   |
| Guía Técnica IDAE nº 12 | 2010 | Condiciones climáticas exteriores de proyecto.  |
| Real Decreto 102        | 2011 | Relativo a la mejora de la calidad del aire.  |
| UNE 100-011-91          | 1991 | Climatización. La ventilación para una calidad aceptable del aire en la climatización de los locales.   |
| UNE EN 779 (ya anulada) | 2013 | Filtros de aire utilizados en ventilación general para eliminación de partículas. Determinación de las prestaciones de los filtros.   |
| UNE-EN ISO 16000-26     | 2014 | Aire de interiores. Parte 26: Estrategia de muestreo del dióxido de carbono (CO <sub>2</sub> ).   |
| UNE-EN 16798-3          | 2018 | Eficiencia energética de los edificios. Ventilación de los edificios. Parte 3: Para edificios no residenciales. Requisitos de eficiencia para los sistemas de ventilación y climatización (Módulos M5-1, M5-4). |
| UNE-EN ISO 16890-1      | 2017 | Filtros de aire utilizados en ventilación general. Parte 1: Especificaciones técnicas, requisitos y clasificación según eficiencia basado en la materia particulada (PM).                                       |
| UNE 77233               | 1996 | Calidad del aire. Tratamiento de datos de temperatura, presión y humedad.   |
| VDI 2081-1              | 2019 | “Noise generation and noise reduction in air conditioning”.   |

Tabla 2.1.- Nomenclatura y título de las normas del proyecto.

### 3. Contextualización y descripción de la estancia

En este capítulo se expondrá la contextualización del proyecto en el marco del IoT y se describirá la estancia objeto de estudio.

En la primera parte de este capítulo se explicarán los conceptos de IoT e Industria 4.0, que como se verá a lo largo del mismo van de la mano. Del IoT se verá el amplio rango de posibilidades que ofrece desde el ámbito doméstico al industrial gracias al desarrollo que están teniendo las nuevas tecnologías de las redes de banda ancha que están permitiendo la interconexión inalámbrica de millones de dispositivos.

Este concepto llevado a la industria es lo que ha desembocado en una nueva revolución industrial. A lo largo del capítulo se verá cuando fue promulgado este concepto de Industria 4.0 por primera vez, en que está consistiendo y que ventajas ofrece esta revolución 4.0 al sector industrial.

Como se verá a lo largo del proyecto, éste se basa en el uso del IoT, ya que se empleará un protocolo de comunicación en el cual un dispositivo enviará la información inalámbricamente a un nodo central, y éste la tendrá reenviará, de la misma manera, a otro dispositivo de actuación. Como se enseñará a continuación esta interconexión entre los dispositivos es en lo que se fundamenta el IoT.

En la segunda parte del capítulo se describirá la estancia donde se instalará el sistema de ventilación.

En primer lugar, se ubicará la estancia dentro del edificio docente al que pertenece, comentando, a su vez, la ciudad donde se encuentra dicho edificio, que un factor determinante a la hora de conocer la calidad del aire exterior, como se verá en el subapartado 4.1 del proyecto.

Ubicada la estancia se procederá a conocer sus dimensiones básicas, como son la superficie de la sala, la altura del techo y un pequeño croquis que mostrará la morfología de la estancia y la longitud de cada una de sus paredes. Como es lógico, las dimensiones de la estancia son básicas para posteriormente dimensionar el sistema de ventilación, el cual se describe en profundidad en el Capítulo 4.

Finalmente, conocidas las dimensiones de la estancia se pasará a identificar el uso que se le da a la misma, el número de horas al año que se utiliza para tales fines, así como su aforo máximo. Estos tres datos son relevantes para calcular tanto la generación máxima de CO<sub>2</sub> por parte de los ocupantes de la sala como el caudal de ventilación que será necesario suministrarle. Por otro lado, el número de horas al año de funcionamiento del sistema de ventilación será necesario para el cálculo de su recuperador de calor. Dichos cálculos se realizan en los subapartados 4.2, 4.3.1 y 4.4.2, respectivamente, del siguiente capítulo.

#### 3.1. Qué es IoT e Industria 4.0

El Internet of Things (IoT) es la interconexión digital entre dispositivos electrónicos enlazados a través de un nodo central, teniendo como objetivo final la conexión de objetos físicos con internet permitiendo el intercambio de datos e información entre ellos [5]. Estos objetos pueden ser muy variados, desde un frigorífico hasta sofisticadas herramientas industriales, por lo que el alcance de esta tecnología, tanto a nivel doméstico como industrial, es muy elevado.





### 3. CONTEXTUALIZACIÓN Y DESCRIPCIÓN DE LA ESTANCIA

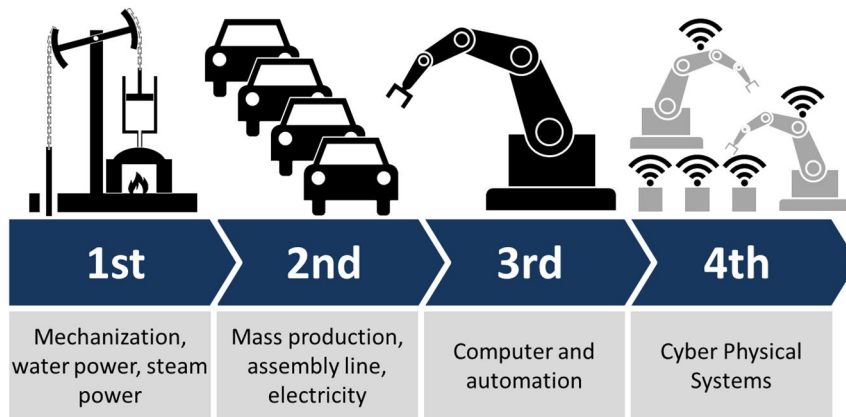


Figura 3.2.- Revoluciones industriales (imagen obtenida bajo licencia "Creative Commons" de la web [10]).

Si bien este concepto no está tan ligado a este proyecto en concreto como el de IoT, no se podía dejar de mencionar ya que muchas empresas incorporan los sistemas de ventilación como un elemento más de su IoT. Además, muchas compañías dedicadas a la ventilación, climatización y control de la calidad del aire ofrecen sistemas que se basan en los principios y cálculos de este proyecto.

### 3.2. Estancia objeto de estudio

La estancia objeto de estudio forma parte del Edificio Polivalente del Campus Universitario de Viesques. Este edificio está ubicado en la zona de Viesques de Gijón. En él pueden encontrarse diversas estancias destinadas principalmente a la docencia, como son aulas y laboratorios, no obstante, en el edificio también se pueden encontrar salas dedicadas al personal administrativo y de dirección de la Escuela Politécnica, así como los despachos de un gran número de docentes. Además de las estancias anteriormente mencionadas, el edificio alberga un salón de actos que será la sala para estudiar. Finalmente, como no podía ser de otra forma, tratándose de un edificio universitario cuenta con una cafetería y un comedor donde docentes y estudiantes pueden reponer fuerzas. En la Figura 3.3 se muestra una imagen aérea de este edificio.



Figura 3.3.- Edificio Polivalente del Campus Universitario de Viesques (imagen obtenida de la web [11]).

### 3. CONTEXTUALIZACIÓN Y DESCRIPCIÓN DE LA ESTANCIA

Como se ha comentado al inicio de este capítulo, la situación geográfica del edificio tiene una clara incidencia en los contaminantes que puedan aparecer en el aire exterior que deberá ser tratado. Este análisis de la calidad del aire exterior de la zona de Viesques en Gijón se realiza en el subapartado 4.1 del siguiente capítulo.

El salón de actos que será la estancia objeto de estudio se halla en la segunda planta del edificio, exactamente en el módulo 0, enfrente de los laboratorios de Química con los que cuenta el edificio, y se puede acceder al salón desde dos puertas situadas en el vestíbulo principal al que se llega directamente tras subir las escaleras desde la primera planta. La ubicación exacta en su planta que ocupa el salón de actos puede verse en el plano de la Figura 3.4 resaltado en color azul claro con borde amarillo.

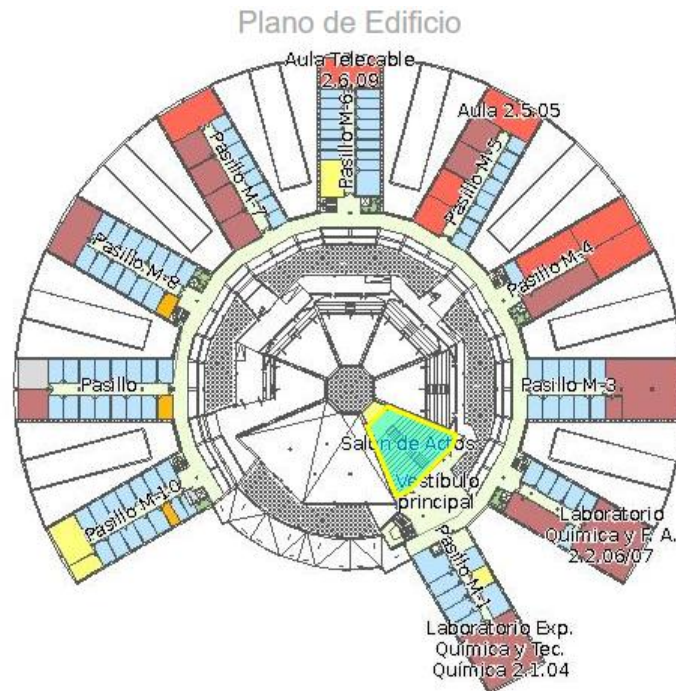


Figura 3.4.- Ubicación del salón de actos en la segunda planta del edificio (imagen obtenida de la web [12]).

Una vez ubicada de la estancia, se procederá a conocer sus dimensiones. El salón de actos tiene una superficie de 180,92 m<sup>2</sup> y una altura de techo estimada (no se ha podido realizar la toma de esta medida debido a la pandemia) de 4 m desde su punto más bajo, ya que la estancia es escalonada. La morfología y las dimensiones de las paredes de la estancia aparecen en la Figura 3.5 que corresponde a un croquis del salón, las cotas aparecen en metros.

### 3. CONTEXTUALIZACIÓN Y DESCRIPCIÓN DE LA ESTANCIA

---

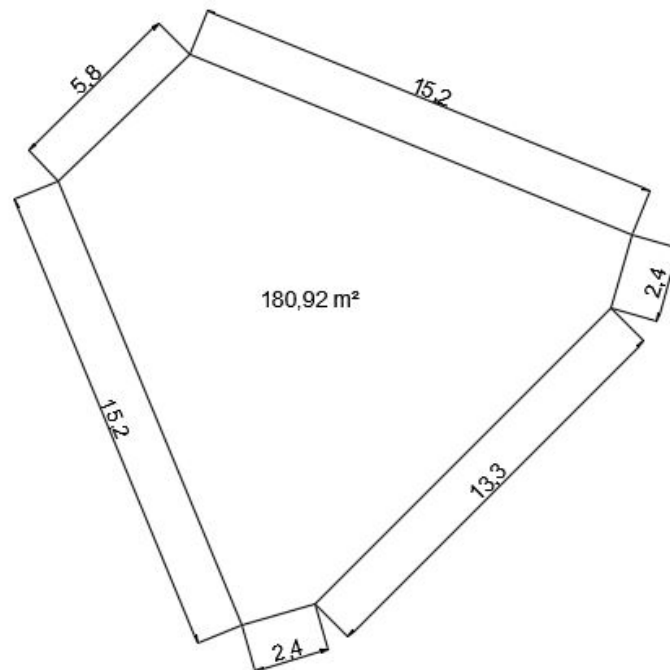


Figura 3.5.- Croquis del salón de actos.

El salón de actos tiene un aforo máximo de 231 personas y su uso habitual es como sala de conferencias, además la estancia acoge multitud de actos universitarios relacionados con jornadas de acogida, entrega de reconocimientos, etc. Uno de estos ejemplos se muestra en la imagen de la Figura 3.6, en la que aparece el director de la Escuela Politécnica, Juan Carlos Campo, durante las jornadas de acogida de este curso 2020/21.

Este tipo de actos se realizan con cierta frecuencia a lo largo del curso académico, el factor de uso de la estancia, medido según las horas de funcionamiento del sistema de ventilación, tal y como se comentó al inicio de este capítulo, también ha de ser tenido en cuenta a la hora del cálculo del recuperador de calor que será necesario incorporar en dicho sistema como se verá en el subapartado 4.4.2 del capítulo siguiente.

Los asistentes a las conferencias se encontrarán sentados en las butacas destinadas a tal fin, así como los conferenciantes cuentan con sillas donde pueden sentarse a dar su discurso, este punto, que parece irrelevante a primera vista, influye en la tasa metabólica de cada uno de ellos, que tiene relación, entre otras cosas, con la actividad que estén desarrollando y como la estén desarrollando, al hallarse sentados la tasa metabólica por ocupante es inferior a si estuvieran realizando otro tipo de actividad que requiriera movimiento [1]. Esto está íntimamente ligado con la producción de CO<sub>2</sub> de cada uno de los ocupantes, ya que a menor actividad menor generación de CO<sub>2</sub>.

### 3. CONTEXTUALIZACIÓN Y DESCRIPCIÓN DE LA ESTANCIA

---



*Figura 3.6.- Conferencia en el salón de actos del edificio polivalente (imagen obtenida de la web del Comercio [13]).*

Finalmente, como se había comentado al inicio de este capítulo, siempre es necesario conocer el uso habitual de una estancia para determinar la calidad de aire interior que deberá haber en la misma, tal y como aparece recogido en la normativa vigente. A su vez, conocer el aforo permite calcular tanto la generación de  $\text{CO}_2$  por parte de los ocupantes de la estancia como calcular el caudal de ventilación que ha de suministrarse para un correcto bienestar térmico de sus ocupantes. Estos dos cálculos, la determinación de la calidad de aire interior y el caudal de ventilación máximo se llevará a cabo en los subapartados 4.2 y 4.3.1 del siguiente capítulo.

### 4. Ventilación

En el presente capítulo se desarrollarán los cálculos concernientes al sistema de ventilación de la estancia objeto de estudio. Estos cálculos incluirán el estudio fluidodinámico del aire en dicho sistema, así como la selección de los equipos de ventilación adecuados para cumplir las exigencias técnicas que indique la normativa.

Los dos primeros subapartados del capítulo versarán sobre la calidad del aire, el primero de ellos categorizará la existente en el exterior del recinto, mientras el segundo lo hará sobre la que será necesaria conseguir en el interior de la estancia.

Como la sala de estudio se encuentra en Gijón, se analizarán los niveles de calidad del aire en esta ciudad, los cuales son recogidos de las diferentes estaciones de medición con las que cuenta el Ayuntamiento repartidos a lo largo de la ciudad, poniendo especial interés en los niveles de CO<sub>2</sub>, debido a que la monitorización del sistema de ventilación de la estancia se realizará en función a estos niveles de concentración.

La calidad del aire interior está determinada por el uso que vaya a tener la estancia, como se ha visto en el capítulo anterior la estancia se utilizará como sala de conferencias. El uso de la estancia categoriza, según la normativa existente [1], la calidad del aire que debe tener y, en función de esta categorización, los niveles de CO<sub>2</sub> máximo que puede albergar para conservar el bienestar de sus ocupantes.

En el siguiente subapartado del capítulo se realizará el cálculo fluidodinámico del sistema de ventilación. Este cálculo se compone de tres partes correlativas:

a) El caudal de ventilación: se calcula a partir de las concentraciones de CO<sub>2</sub> en la estancia. Para ello, ha de tenerse en cuenta la diferencia entre los niveles de concentración máxima entre el exterior y el interior de la estancia, la generación de CO<sub>2</sub> por parte de los ocupantes de la estancia y la eficiencia de la ventilación.

b) Las pérdidas de carga o presión: se empezarán dimensionando los conductos de ventilación, este cálculo será función de la velocidad del aire a través de ellos y del caudal de ventilación. A continuación, se calcularán las pérdidas de carga del fluido a lo largo de los conductos de ventilación, existirá un circuito dedicado a la impulsión de aire y otro a la extracción o retorno de este desde el interior de la estancia. Estas pérdidas podrán ser dinámicas y estáticas, como se verá a lo largo del subapartado. A estas pérdidas se les sumarán las que provocan los equipos instalados en el sistema de ventilación tales como, recuperador de calor, filtros, rejillas y difusores. La suma de todas estas pérdidas de carga serán las que deban afrontar los ventiladores.

c) La potencia y el rendimiento: se calculará la potencia que se debe suministrar al fluido y el rendimiento de la ventilación teniendo en cuenta la potencia que consumirá el ventilador. Con esta última potencia y el caudal de ventilación será posible calcular la potencia específica que categorizará la instalación según dicta la normativa [1].

A partir del caudal de ventilación y las pérdidas de carga, se obtiene un punto de funcionamiento, con él se buscará en los catálogos de ventiladores los más adecuados para la instalación, obteniéndose así la potencia consumida por el ventilador comentada en este último punto.

Una vez realizado el cálculo fluidodinámico, se procederá a dar una solución para el sistema de ventilación, ésta va a pasar por una selección de equipos adecuada a las necesidades fluidodinámicas y normativas que se plantean.

La UTA albergará todos los equipos destinados al tratamiento de aire, desde el filtrado, la impulsión y el retorno del aire y la recuperación de calor entre las dos corrientes de aire. Esta

UTA estará formada por módulos, cada uno de ellos con una función específica para el tratamiento del aire.

El mercado de UTAs ofrece múltiples soluciones que ya incluyen todos los equipos necesarios para el sistema de ventilación del proyecto, no obstante, con el fin de adecuarse a las necesidades específicas de cada equipo se ha decidido realizar una selección individual de cada uno de ellos, su unión formará la UTA del proyecto.

El primero de estos equipos que se va a analizar será el de filtrado, se comentarán los requisitos existentes que han de tener estos filtros en función de la normativa y las calidades del aire exterior e interior. Se destaca un cambio en la norma reciente que implica una variación en la nomenclatura de los tipos de filtros, debido a que esta modificación aún no aparece recogida en el RITE [1], se ha decidido exponer cuales serán estos cambios con el fin de aclarar la selección de los filtros. Finalizado este apunte se seleccionan los filtros adecuados para el proyecto cumpliendo con la normativa actual, mencionando sus características.

El segundo equipo corresponde al recuperador de calor, encargado de aprovechar el calor procedente del interior de la estancia para calentar el aire proveniente del exterior. La instalación de este recuperador es obligatoria debido a que se supera un cierto mínimo de caudal de ventilación establecido por la normativa como se verá a lo largo del capítulo. En su correspondiente subapartado se comentarán los tipos de recuperadores de calor más habituales para los sistemas de ventilación y se escogerá el más adecuado para la instalación, comentando sus características.

El tercer, y último, equipo que formará parte de la UTA serán los ventiladores, los cuales se encargan de impulsar el aire a través de los conductos. En su correspondiente subapartado se hablará de los diferentes tipos de ventiladores existentes para este tipo de soluciones, la clasificación que tienen en función de las pérdidas de carga que debe vencer, pudiendo ser de baja, media o alta presión, y, finalmente, se selecciona el ventilador más adecuado para el circuito de impulsión y para el de retorno.

Un aspecto que se va a tener en cuenta a la hora de seleccionar el ventilador adecuado es que el motor de éste ha de poder modificar su régimen de giro mediante un variador de frecuencia, ya que la monitorización de esta velocidad de giro va a ser realizada a partir de la concentración de CO<sub>2</sub>.

Con los ventiladores quedarían definidos todos los equipos necesarios para el tratamiento del aire del sistema de ventilación, no obstante, será necesario la instalación de unas unidades terminales en los conductos. En el circuito de impulsión estas unidades terminales serán unos difusores de aire que permitan propagar el aire a lo largo de toda la estancia, sin que se vea afectado el confort térmico de los ocupantes por las corrientes de aire que puedan provocar. En el caso del circuito de retorno esta unidad terminal será una rejilla situada a la entrada del aire interior desde la estancia en dicho circuito, se emplea principalmente para evitar que partículas u objetos de gran tamaño puedan dañar los equipos del sistema de ventilación.

### 4.1. Calidad del aire exterior

La calidad del aire exterior (ODA) se determina a partir de las mediciones diarias y anuales de unos determinados compuestos según recoge la Directiva Europea 2008/50/CE y que su transposición a la legislación española se ha realizado mediante el Real Decreto 102/2011, el cual establece los compuestos que han de medirse en las estaciones de control y vigilancia ambiental. Las mediciones que realizan son de las partículas en suspensión PM<sub>10</sub> y PM<sub>2,5</sub> y de los contaminantes dióxido de azufre, SO<sub>2</sub>, dióxido de nitrógeno, NO<sub>2</sub>, monóxido de carbono, CO, ozono, O<sub>3</sub>, y BTX (benceno, tolueno y xileno).

La ciudad de Gijón se caracteriza por encontrarse en un entorno industrial, por lo tanto, el aire de la ciudad puede contener partículas de estos compuestos, los cuales no es deseable que entren en el ambiente interior del salón de actos durante la ventilación.

---

## 4. VENTILACIÓN

---

Los últimos datos anuales recogidos por el Ayuntamiento de Gijón datan del 2016, en ellos se observa que ninguna estación ha superado los umbrales mínimos establecidos por el Real Decreto 102/2011, por lo que se puede decir que la calidad del aire en Gijón es muy buena [14] y por lo tanto se puede aceptar un nivel de ODA 1, que según establece el RITE es aire puro que puede contener partículas sólidas de forma temporal [1].

La determinación del nivel de calidad del aire exterior será clave a la hora de escoger los filtros con los que contará la unidad de tratamiento del aire (UTA) que se debe instalar para la ventilación del salón de actos.

Por otra parte, la concentración de CO<sub>2</sub> en el aire exterior de Gijón, al tratarse de una ciudad, el RITE establece un valor de 400 ppm [1]. Este valor se utilizará próximamente para el cálculo del caudal de aire que se debe introducir en la estancia en función de la concentración del CO<sub>2</sub> en el interior en cada momento.

### 4.2. Calidad del aire interior

El RITE [1] establece los criterios de calidad del aire interior para todo tipo de locales, en este apartado se seguirán sus directrices para determinar esta calidad, gracias a la cual se conocerán los valores límite de concentraciones de CO<sub>2</sub> en la estancia para llevar a cabo la regulación del sistema de ventilación.

La calidad del aire en el interior de los locales está ordenada por categorías en función del uso que vaya a tener el local. En el caso del salón de actos del presente proyecto se encuentra en una categoría 3 (IDA 3). Esta categoría considera que la calidad del aire interior deber ser media [1].

La consideración de categoría 3 para la calidad del aire interior permite determinar la máxima concentración de CO<sub>2</sub> que puede haber presente en la estancia. En este caso la concentración máxima debe ser de 800 ppm superior a la concentración que haya en el aire exterior, establecida por normativa en el subapartado 4.1 con un valor de 400 ppm. Por lo tanto, la concentración máxima es de 1200 ppm para el interior del salón de actos [1].

### 4.3. Cálculo de la ventilación

Una vez conocida la ubicación, las características de la estancia objeto del proyecto, así como del edificio que la alberga, la calidad de aire exterior en la zona y la calidad del aire interior que se necesita en el salón, se va a realizar el cálculo del sistema de ventilación.

En los siguientes subapartados se realizará el cálculo fluidodinámico del sistema de ventilación, caudal de ventilación y presiones y pérdidas de carga en los conductos, obteniéndose así el punto de funcionamiento de la instalación, con el que se podrá determinar la potencia necesaria en el ventilador, y la potencia específica de la misma, necesaria para su categorización, según la normativa vigente.

El primer paso consistirá en calcular el caudal de ventilación de diseño para el salón, se utilizarán las calidades del aire, el aforo y la tasa de emisión de CO<sub>2</sub> de las personas que allí se encuentren.

Posteriormente se calcularán las pérdidas de carga o presión a lo largo de los conductos, a las que se sumarán las pérdidas de cada uno de los equipos que forma parte de la instalación, filtros, recuperador de calor, rejillas y difusores.

Una vez obtenidos los valores para el caudal de ventilación y para las pérdidas de presión del circuito de ventilación, las cuales serán estáticas, se podrá seleccionar el ventilador más adecuado para la instalación. Éste, además de proporcionar el caudal de diseño y vencer las pérdidas de carga, ha de contar con un variador de frecuencia que regule su velocidad de giro para los momentos en los que se exceda el límite de concentración de CO<sub>2</sub> en el interior del salón.

Finalmente, con los valores del caudal y las pérdidas de carga se realizará el cálculo de las potencias, estas potencias incluyen la que será necesaria suministrar al fluido para que pueda circular por de los conductos y salga a través de los difusores a la velocidad y presión adecuadas, y la potencia específica de la instalación para su categorización.

### 4.3.1. Caudal de ventilación

El caudal de ventilación asegura unos niveles de concentraciones de CO<sub>2</sub> adecuados dentro de la estancia, evitando así, en todo momento, encontrarse con niveles superiores al máximo estipulado en el subapartado 4.2. El caudal será regulable mediante un variador de frecuencia que controla la velocidad de giro del ventilador, no obstante, para el diseño de la instalación y la selección del ventilador se debe emplear el caudal máximo que ha de suministrarse.

El cálculo de este caudal máximo va a depender de la concentración de CO<sub>2</sub> en el aire exterior, la concentración máxima de CO<sub>2</sub> en el aire interior, la generación de CO<sub>2</sub> de los ocupantes del salón y la efectividad de la ventilación, mediante la Ecuación (4.1) [4].

$$Q_v = \frac{G_h}{C_{h,int} - C_{h,ext}} \cdot \frac{1}{\varepsilon_v} \cdot 10^6 \frac{l}{s} \quad (4.1)$$

Donde  $Q_v$  es el caudal de ventilación,  $G_h$  es la carga de CO<sub>2</sub> que se va a generar en el salón debido a los ocupantes,  $C_{h,int} - C_{h,ext}$  corresponde a las concentraciones de CO<sub>2</sub> en el interior y exterior, su diferencia está establecida en el RITE y es de 800 ppm para las calidades de aire interior y exterior del proyecto [1], y  $\varepsilon_v$  es la efectividad de la ventilación.

La carga de CO<sub>2</sub> generada es la correspondiente a las 231 personas de aforo máximo que tiene el salón de actos. La norma UNE 100-011-91 indica que los ocupantes de este tipo de estancias (salones de actos, auditorios, salas de conferencias, etc) producen 19 l/h de CO<sub>2</sub> [4], por lo tanto, realizando el sencillo cálculo como se muestra en la Ecuación (4.2), se obtiene que el conjunto de los ocupantes genera 4389 l/h de CO<sub>2</sub>.

$$G_h = 231 \text{ personas} \cdot 19 \frac{l}{h \cdot \text{persona}} = 4389 \frac{l}{h} \quad (4.2)$$

La efectividad de la ventilación depende de la posición de los difusores de impulsión y las rejillas de extracción y de la temperatura del aire impulsado. Este parámetro es difícil de calcular a priori, por lo tanto, a de estimarse. Los valores mostrados en la norma UNE-EN 13779 para las características de la sala de estudio se encuentran en el rango de 0,9 a 1 [4], englobando funcionamiento en invierno y en verano, por ello para estar en el lado de la seguridad se fija un valor de 0,9 para la efectividad de la ventilación.

Una vez obtenidos todos los parámetros de la Ecuación (4.1), del caudal de ventilación, se procede en la Ecuación (4.3) a resolverla.

$$Q_v = \frac{4389}{800} \cdot \frac{1}{0.9} \cdot \frac{10^6}{3600} = 1693 \frac{l}{s} \quad (4.3)$$

Este resultado indica el caudal máximo de ventilación necesario si el salón de actos se encuentra completamente lleno, sin embargo, esto no siempre será así. En ese momento entraría en juego la regulación del caudal, el regulador permitirá la modificación del caudal entre el 15 y 100% del máximo que se acaba de calcular [3].

La simplificación que se va a realizar consiste en igualar el caudal impulsado al extraído. Esta simplificación permitirá una mejor comparativa entre los resultados de la impulsión y la extracción. En un caso real estos dos caudales pueden ser iguales o diferentes, pero en el proyecto estudiado no ha lugar hacer diferenciación entre ellos.



Un último detalle del caudal de ventilación son las unidades, normalmente en los catálogos de los ventiladores este caudal aparece en unidades superiores ( $\text{m}^3/\text{h}$ ). Debido a este motivo se realiza una sencilla conversión de unidades que permite pasar el caudal de ventilación a dichas unidades, por lo tanto, el caudal que se empleará para seleccionar el ventilador va a ser de  $6096 \text{ m}^3/\text{h}$ .

### 4.3.2. Presiones y pérdidas de carga en los conductos

El cálculo de la presión va a permitir escoger el ventilador más adecuado para las características de la instalación. Cabe recordar que el ventilador ha de suministrar energía suficiente al fluido para poder ventilar la estancia, por lo tanto, el ventilador deberá vencer una serie de pérdidas acaecidas a lo largo de la red de conductos, así como las provocadas por los equipos instalados.

La presión total es la suma de la presión estática y la presión dinámica, la primera de ellas es la que el ventilador debe vencer y será igual a la suma de todas las pérdidas del circuito, mientras que la segunda depende exclusivamente de las propiedades del fluido, su densidad y su velocidad a la salida del ventilador. Por lo tanto, a continuación, se calculará la presión dinámica y posteriormente se calcularán las pérdidas de carga de la instalación para determinar la presión estática. Con los valores de estas presiones se hallarán la potencia a suministrar al fluido y la potencia consumida por el ventilador, para obtener la potencia específica que es necesario justificar en toda instalación según el RITE [1].

En primer lugar, se deberá fijar una velocidad de paso del aire a través de los conductos. Esta velocidad debe ser tal, que el aire a su paso por los conductos de ventilación no se supere una determinada potencia sonora de 35 dB, establecido para salas de conferencias en la norma VDI 2081-1. Por ello se selecciona una velocidad baja para el paso del aire a través de los conductos, esta velocidad será de 4 m/s.

Una vez fijada esta velocidad de paso se calcula el diámetro que deben tener los conductos de ventilación. Este diámetro se calcula mediante la Ecuación (4.4) [3], dando como resultado un diámetro de 734 mm. Un diámetro de tales dimensiones no se encuentra estandarizado en el mercado, por lo tanto, se escogerá un diámetro normalizado de 750 mm, esto provocará una ligera disminución de la velocidad del aire a través de los conductos con respecto a la fijada al inicio del subapartado.

$$D = \left( \frac{4 \cdot Q_v}{\pi \cdot V} \right)^{1/2} \quad (4.4)$$

Donde  $Q_v$  es el caudal de ventilación en  $\text{m}^3/\text{s}$  y  $V$  la velocidad media del aire por los conductos de ventilación.

La nueva velocidad, para el diámetro normalizado de 750 mm, se obtiene también de la Ecuación (4.4) pero en este caso despejando la velocidad. Entonces la velocidad real del aire a través de los conductos es de 3,83 m/s.

Con los valores del diámetro normalizado, 750 mm, y la velocidad real del aire por los conductos, 3,83 m/s, es posible esclarecer la potencia sonora que tendrá el aire del sistema de ventilación mediante la Ecuación (4.5) que aparece recogida en la norma VDI 2081 para conductos rectos metálicos [15].

$$L_w = 7 + 50 \cdot \log V + 10 \cdot \log S \quad (4.5)$$

Donde  $V$  es la velocidad del aire en m/s y  $S$  la sección del conducto de aire de ventilación en  $\text{m}^2$ , obtenida a partir del diámetro del conducto.

La potencia sonora obtenida es de 32,63 dB, inferior al máximo permitido por la normativa para salas de conferencias, por lo tanto, la velocidad real del aire a través de los conductos será de 3,83 m/s y un diámetro de los conductos de 750 mm.

#### 4. VENTILACIÓN

Una vez determinada la velocidad del fluido a lo largo de los conductos es posible calcular la primera de las presiones, la dinámica, ésta depende de la densidad del fluido, en este caso el aire, y la velocidad del fluido recién calculada según la Ecuación (4.6) [3].

$$p_D = \frac{1}{2} \cdot \rho \cdot V^2 \quad (4.6)$$

Donde la densidad del aire,  $\rho$ , es 1,21 kg/m<sup>3</sup> [3] y la velocidad del fluido es 3,83 m/s. Con estos valores de densidad y velocidad el resultado de la presión dinámica es 8,89 Pa, se estima que la velocidad a través de los conductos será igual tanto para la impulsión como para la extracción, por ende, la presión dinámica también será igual.

Las pérdidas de carga de una red de conductos dependen de la fricción del aire con las paredes del conducto, de los elementos de difusión y rejillas de extracción, y de los equipos que conforman el sistema de ventilación, filtros y recuperadores de calor. Las pérdidas de carga debidas a los elementos de la instalación de ventilación son establecidas por el fabricante, por ello en los subapartados que van desde el 4.4.1 hasta el 4.5, donde se escogerán estos elementos se especifica la pérdida de carga de cada uno de ellos. No obstante, el RITE recoge unos valores máximos que han de tener cada uno de estos elementos [1]. Las pérdidas de carga máxima recogidas en la norma y las propias de los elementos seleccionados del sistema de ventilación se muestran en la Tabla 4.1. Mientras que las pérdidas de carga en los conductos se calcularán como se desarrolla a continuación, y serán equivalentes a la presión estática que debe suministrar el ventilador.

| Elemento               | Circuito   | Pérdida de carga           |                            |
|------------------------|------------|----------------------------|----------------------------|
|                        |            | Norma                      | Seleccionado               |
| Filtros                | Impulsión  | Depende del tipo de filtro | Depende del tipo de filtro |
|                        | Extracción |                            | 98 Pa                      |
| Recuperadores de calor | Impulsión  | 120 Pa                     | 32,5 Pa                    |
|                        | Extracción |                            | 31,9 Pa                    |
| Elementos de difusión  | Impulsión  | 200 Pa                     | 10 Pa                      |
| Rejillas de retorno    | Extracción | 40 Pa                      | 20,6 Pa                    |

*Tabla 4.1.- Pérdidas de carga máxima para cada equipo del sistema de ventilación.*

Una breve aclaración para el caso de los filtros, tal y como aparece en esta Tabla 4.1 la pérdida de carga en los filtros del circuito de impulsión depende del tipo de filtro, esto es debido a que la impulsión contará con dos filtros, uno de los cuales se denomina prefiltro mientras que el otro es un filtro normal. Estos dos filtros tienen pérdidas de carga distintas, de ahí esa dependencia del tipo de filtro para clarificarla. No obstante, en el subapartado 4.4.1 se especifican las características y el uso para cada uno de estos filtros.

El siguiente paso será calcular las pérdidas de carga provocadas por los conductos, éstas pueden dividirse en pérdidas primarias y en pérdidas secundarias. Las primarias son debidas a la viscosidad del fluido, tamaño y forma geométrica del conjunto y la rugosidad de las paredes del conducto, mientras que las secundarias se deben a los accesorios con los que contará la red de conductos [3].

Las pérdidas primarias se calculan en función de la fricción del aire sobre las paredes del conducto de ventilación como se puede ver en la Ecuación (4.7) [3].

$$\Delta P_p = f \cdot \frac{L}{D} \cdot \frac{1}{2} \cdot \rho \cdot V^2 \quad (4.7)$$

---

#### 4. VENTILACIÓN

---

Donde  $f$  es el factor de fricción,  $L$  la longitud de los conductos,  $D$  el diámetro de los conductos,  $\rho$  es la densidad del aire y  $V$  la velocidad del aire en los conductos.

El cálculo del factor de fricción se puede realizar de manera iterativa mediante la Ecuación de Colebrook-White [16]. No obstante, existen otras ecuaciones que permiten hallar el resultado asumiendo un pequeño error, en este proyecto se va a emplear la Ecuación de Barr, Ecuación (4.8) [17].

$$\frac{1}{\sqrt{f}} = -2 \cdot \log \left( \frac{\varepsilon}{3.7 \cdot D} + \frac{5.1286}{Re^{0.89}} \right) \quad (4.8)$$

Donde  $\varepsilon$  es la rugosidad de las paredes, es función del material de fabricación de los conductos, los conductos van a ser de chapa de acero galvanizado con una rugosidad de 0,09 mm. Y  $Re$  es el número de Reynolds, que se calcula según la Ecuación (4.9) [16].

$$Re = \frac{\rho \cdot V \cdot D}{\mu} \quad (4.9)$$

La velocidad del aire y el diámetro de los conductos ya fueron calculados, mientras que la densidad,  $\rho$ , y la viscosidad dinámica del fluido,  $\mu$ , se estiman en  $1,21 \text{ kg/m}^3$  y  $1,80 \cdot 10^{-5} \text{ Pa}\cdot\text{s}$  [3] respectivamente, para una temperatura de  $20 \text{ }^\circ\text{C}$  y al nivel del mar [18].

El valor del número de Reynolds resultante es de 193000, este resultado elevado indica una baja influencia de la viscosidad en el flujo, por lo tanto, puede suponerse la hipótesis de flujo ideal.

Con el resultado del número de Reynolds ya es posible calcular el factor de fricción mediante la Ecuación (4.8), dando como resultado 0,0167.

El valor del factor de fricción permite calcular las pérdidas primarias en los conductos. Debe tenerse en cuenta que la longitud de los conductos de ventilación es de 20 m para la impulsión de aire y de 18 m para el retorno. A partir de los valores calculados ya se pueden determinar las pérdidas de carga primarias en los conductos, mediante la Ecuación (4.7), que serán de 38,73 Pa para la impulsión y de 34,86 Pa para la extracción.

Las pérdidas de carga secundarias en los conductos de ventilación se producen por los cambios de dirección del fluido acaecidos en los accesorios que conforman la red de conductos. Para el cálculo se estima que son un 30% con respecto a las primarias [3], por lo tanto, las pérdidas de carga secundaria van a ser de 11,62 Pa en la impulsión y de 10,46 Pa en la extracción de aire.

Finalmente, las pérdidas de carga totales en los conductos serán la suma de las pérdidas primarias y las secundarias, en este caso 50,35 Pa para la impulsión de aire y 45,32 Pa para la extracción.

Las pérdidas de carga totales serán la suma de todas las pérdidas anteriores. Ha de tenerse en cuenta que los sistemas de impulsión y extracción de aire están constituidos por diferentes elementos, los únicos elementos comunes son el recuperador de calor y el filtro, el resto difieren entre sí, para la impulsión se añaden las pérdidas de carga provocadas por el prefiltro y por los difusores de aire, mientras que para la extracción se añade la rejilla de retorno a los elementos comunes. Con todo ello, las pérdidas de carga en la impulsión se elevan hasta los 455,96 Pa y para la extracción hasta los 314,42 Pa. Estas serán, a su vez, las presiones estáticas requeridas en los ventiladores de impulsión y extracción respectivamente.

Un último apunte, en el campo de las presiones, es la presión total, que será requerida para el cálculo de potencias del siguiente subapartado, esta presión es la suma de las pérdidas de carga a lo largo de la instalación y la presión dinámica. Nuevamente se obtendrán dos resultados para la presión total, el primero de ellos será la presión total en el circuito de impulsión, de valor

## 4. VENTILACIÓN

464,85 Pa, y el segundo para el circuito de retorno o extracción, de valor 323,30 Pa. En la Tabla 4.2 se muestra un resumen de los resultados de las presiones calculadas en este subapartado.

| Circuito   | Presión           |           |
|------------|-------------------|-----------|
|            | Tipo              | Valor     |
| Impulsión  | Dinámica          | 8,89 Pa   |
|            | Pérdidas de carga | 455,96 Pa |
|            | Total             | 464,85 Pa |
| Extracción | Dinámica          | 8,89 Pa   |
|            | Pérdidas de carga | 314,42 Pa |
|            | Total             | 323,30 Pa |

Tabla 4.2.- Resumen de las presiones en la red de conductos.

Con los valores de la presión total se da por finalizado el cálculo de las presiones del sistema de ventilación, el siguiente subapartado estará dedicado al cálculo de potencias, con el fin de hallar la potencia específica de la instalación exigida por la normativa.

### 4.3.3. Potencia específica de ventilación

En este subapartado se desarrollarán los pasos necesarios para obtener la potencia específica de la instalación de ventilación, con el fin de determinar la categoría en la que se encuentra el sistema de ventilación estudiado.

El primer paso será calcular la potencia que se debe suministrar al fluido por parte del ventilador para que el aire avance a la velocidad establecida, en el subapartado 4.3.2, a través de los conductos de ventilación. Esta potencia es el resultado de multiplicar el caudal de ventilación requerido por la presión total tal y como aparece en la Ecuación (4.10) [3].

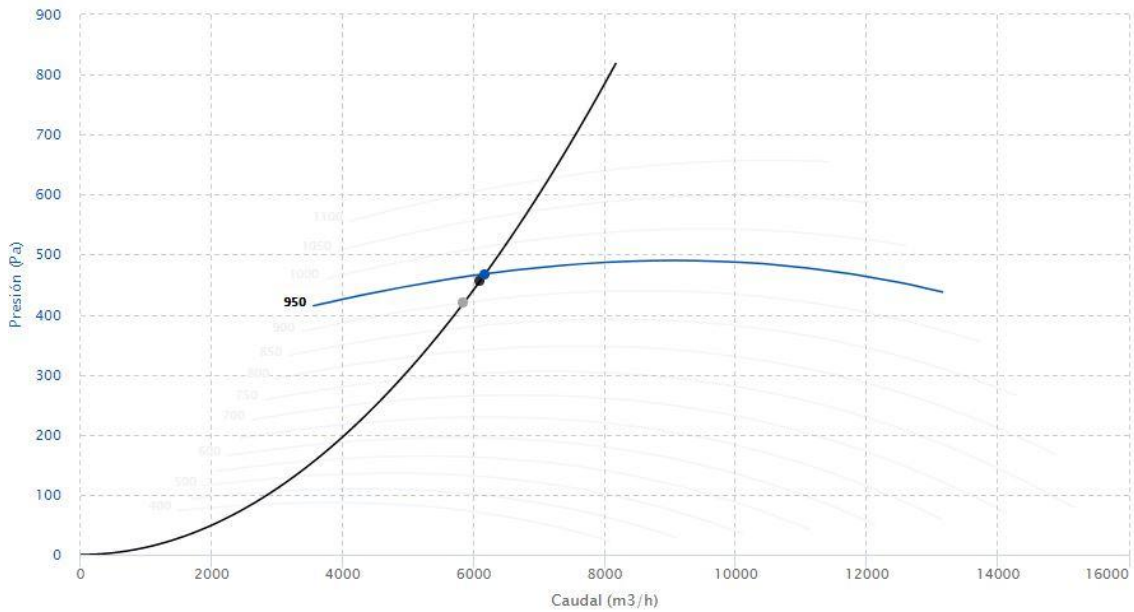
$$\dot{P}_f = Q_v \cdot \Delta P_T \quad (4.10)$$

Donde  $Q_v$  es el caudal de ventilación y  $\Delta P_T$  es la variación de presión total, corresponde a la diferencia de presiones entre la atmosférica y la total, por lo tanto, al ser en términos relativos, coincide con la presión total.

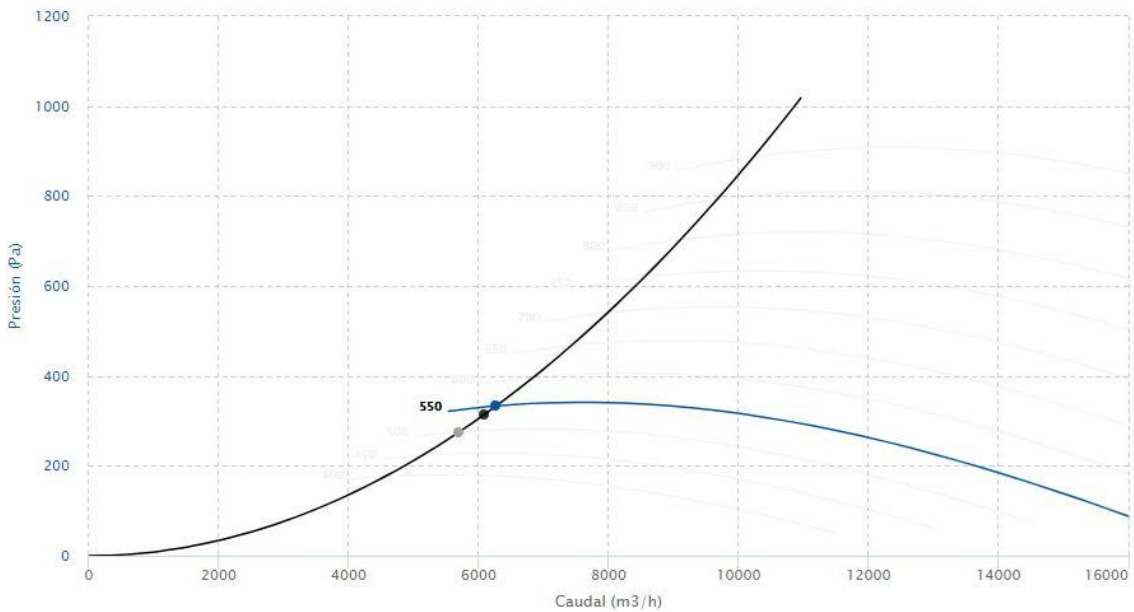
El siguiente paso corresponde a la selección del ventilador a partir de su curva característica con los valores de caudal y presión estática. El caudal de ventilación calculado será el caudal nominal de trabajo del ventilador en función de los niveles de concentración de  $CO_2$  en el aire de la estancia permitidos por la normativa [1], pues, cabe recordar que al actuar sobre el variador de frecuencia del ventilador lo que se consigue es modificar la velocidad de giro del ventilador [3], y esto permitirá el aumento o disminución del caudal, desde un mínimo hasta un máximo, ofreciendo diferentes puntos nominales de funcionamiento para un mismo ventilador. A su vez, las potencias que se deben suministrar al fluido serán diferentes en cada punto de funcionamiento ya que son función del caudal.

En el subapartado 4.4.3 se comentará pormenorizadamente la elección y las características de los ventiladores para la impulsión y la extracción de aire. En el catálogo del fabricante se comprueba la potencia que debe tener el motor para aportar dicho caudal y vencer las pérdidas de carga. En el caso de la impulsión el ventilador escogido consume una potencia máxima de 1,5 kW, mientras que en el caso de la extracción de aire la potencia máxima que puede consumir es de 1,1 kW. En los siguientes dos gráficos, Figura 4.1 y Figura 4.2, se muestra como se ha realizado la selección de los ventiladores a partir de su punto de funcionamiento, con las curvas resistentes de cada circuito y característica del ventilador, el primero de ellos corresponde a la selección del ventilador de impulsión y el segundo al de extracción.

## 4. VENTILACIÓN



*Figura 4.1.- Punto de funcionamiento del ventilador de impulsión [19].*



*Figura 4.2.- Punto de funcionamiento del ventilador de extracción [20].*

Una vez conocidas la ecuación que rige la potencia aportada al fluido, véase la Ecuación (4.10), y la potencia consumida por los ventiladores de impulsión y extracción, se calcula el rendimiento de la ventilación mediante la Ecuación (4.11) [3].

$$\eta = \frac{\dot{P}_f}{\dot{P}_v} \quad (4.11)$$

Donde  $\dot{P}_f$  es la potencia aportada al fluido y  $\dot{P}_v$  es la potencia consumida por el ventilador, el rendimiento será relativo al punto de funcionamiento del ventilador en función de las necesidades de ventilación de la estancia.

En la Tabla 4.3 se muestra el cálculo de potencias realizado para el punto de funcionamiento a caudal máximo, lo que implica los mayores niveles de CO<sub>2</sub> en el salón.

#### 4. VENTILACIÓN

| Circuito   | Potencia |           | Rendimiento |
|------------|----------|-----------|-------------|
|            | Aportada | Consumida |             |
| Impulsión  | 787,13 W | 1500 W    | 52%         |
| Extracción | 547,45 W | 1100 W    | 50%         |

Tabla 4.3.- Cálculo de las potencias a caudal máximo.

Después de hallarse la potencia consumida por el ventilador ya es posible el cálculo de la potencia específica del sistema de ventilación, necesario para establecer la categoría a la que pertenece el mismo [1], mediante la división entre la mayor potencia consumida por el ventilador y el caudal máximo, tal y como aparece en la Ecuación (4.12) [3]. Este cálculo ha de realizarse tanto para el ventilador situado en el circuito de impulsión como para el ventilador situado en el de extracción.

$$W_{esp} = \frac{\dot{P}_v}{Q_v} \quad (4.12)$$

Donde  $\dot{P}_v$  es la potencia consumida por el ventilador y  $Q_v$  el caudal máximo de la instalación.

La potencia específica y la categoría de cada ventilador aparecen reflejados en la Tabla 4.4, se observa que ambos cumplen con el resultado esperado siendo superior en el caso de la impulsión que en de la extracción.

| Circuito   | Potencia específica          | Categoría |
|------------|------------------------------|-----------|
| Impulsión  | 885,85 W/(m <sup>3</sup> /s) | SFP 3     |
| Extracción | 649,62 W/(m <sup>3</sup> /s) | SFP 2     |

Tabla 4.4.- Categoría de los ventiladores del sistema de ventilación.

Una vez hallada la potencia específica de cada ventilador y la categoría a la que pertenecen, el sistema de ventilación queda definido numéricamente, en los siguientes subapartados de este capítulo se procederá a estudiar detalladamente cada equipo que compone la instalación.

#### 4.4. Unidad de tratamiento de aire

La unidad de tratamiento de aire (UTA) es la encargada proporcionar el caudal de aire adecuado, y en condiciones óptimas, al salón de actos. El aire proveniente del exterior a su paso por la UTA sufre una serie de modificaciones ambientales para poder ser impulsado hacia el interior de la estancia.

Las unidades de tratamiento de aire constan de una serie de módulos, instalados de forma conjunta, que permiten el tratamiento del aire proveniente del exterior y proporcionan al fluido la energía necesaria para alcanzar las condiciones de impulsión y expulsión adecuadas para la ventilación de la estancia.

Existirán en la UTA dos circuitos uno para la impulsión de aire y otro para el retorno del aire de la estancia, estos circuitos tendrán como nexo común el recuperador de calor dónde se realiza el intercambio térmico entre las dos corrientes de aire.

El recuperador de calor, común a ambos circuitos, emplea parte del calor proveniente del aire de retorno de la sala en calentar el aire proveniente directamente del exterior del

---

## 4. VENTILACIÓN

---

exterior. Su instalación es obligatoria debido a que el caudal de ventilación es superior a  $0.5 \text{ m}^3/\text{s}$  [1], tal y como se ha calculado en el subapartado 4.3.1.

### I. Circuito de impulsión

El primer módulo del circuito de impulsión, antes de llegar al recuperador de calor partiendo desde el exterior, contendrá el prefiltro encargado de evitar que las partículas de mayor tamaño entren en el sistema de ventilación y puedan dañar el resto de los equipos. La utilización del prefiltro se vuelve obligatoria al contar la instalación con recuperador de calor.

Una vez superado el recuperador de calor, el aire del circuito de impulsión pasará por un segundo filtro, más restrictivo que el prefiltro inicial, que evitará la entrada de las partículas más finas al interior de la sala, lo que podría ocasionar molestias a los ocupantes y daños en el ventilador.

El último elemento en el circuito de impulsión será el ventilador de impulsión encargado de proporcionar el caudal de ventilación calculado en el subapartado 4.3.1 de  $6096 \text{ m}^3/\text{h}$ , y que además debe vencer las pérdidas de carga del circuito de impulsión calculadas en el subapartado 4.3.2 que ascendían hasta los  $464,85 \text{ Pa}$ .

### II. Circuito de extracción o retorno

El circuito de retorno se va a analizar siguiendo el flujo de aire desde el interior de la estancia y su salida al exterior. El aire proveniente del interior de la sala deberá pasar por un filtro que evite la entrada al sistema de ventilación de cualquier tipo de partícula que se haya podido generar en la estancia, con el fin de evitar problemas en los equipos de extracción.

El aire de retorno, una vez filtrado, será impulsado por el ventilador de retorno. Este ventilador ha de ser capaz de extraer de la sala el mismo caudal de ventilación que le era impulsado,  $6096 \text{ m}^3/\text{h}$ , y, a su vez, vencer las pérdidas de carga del circuito de retorno calculadas en el 4.3.2 en  $323,30 \text{ Pa}$ .

El primer módulo, situado en la entrada del aire exterior, debe ser de filtrado, en este caso ha de tratarse de un filtro que evite la entrada hacia el resto del sistema de ventilación de las partículas de mayor tamaño que hubiera en el aire. Este primer módulo se denomina prefiltro y es obligatoria su instalación debido a que el sistema de ventilación contará con recuperador de calor, como se comentará más adelante.

Finalmente, el aire de retorno atraviesa el recuperador de calor produciéndose el intercambio térmico y es expulsado hacia el exterior.

La Figura 4.3 esquematiza la posición que ocupan cada uno de los elementos aquí mencionados en la UTA. El esquema se ha realizado con el programa SystemAirCAD desarrollado por la empresa SystemAir. Esta una compañía internacional sueca de fabricación y distribución de soluciones HVAC, que lleva trabajando en el sector desde el año 1974 [21], ofrece esta herramienta que permite la personalización de la UTA en función de las características propias de cada instalación.

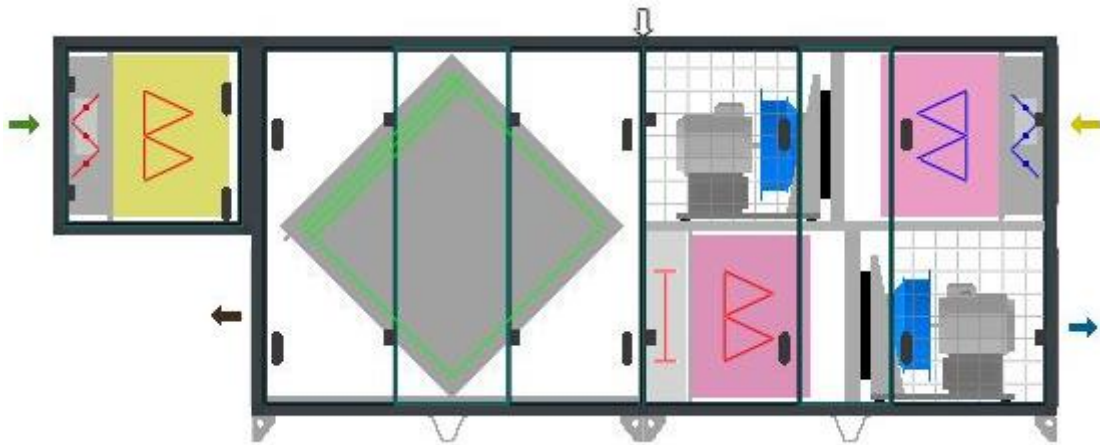


Figura 4.3.- Esquema de la UTA del proyecto.

Cabe decir, a su vez, que el esquema mostrado es una de las múltiples configuraciones o disposiciones que pueden tener los elementos en la UTA, ya que existen ocasiones que incluyen más, como es el caso de baterías para la climatización, o menos si el sistema no requiere recuperador de calor, por poner algunos ejemplos.

En los siguientes subapartados se explica pormenorizadamente cada uno de los elementos que van a conformar la UTA y que se han mencionado a lo largo de este capítulo. Comentar, también, que el mercado ofrece UTAs con todos los módulos necesarios, pero para la realización de este proyecto se ha decidido seleccionar los elementos que la forman por separado para escoger el equipo más ajustado a las necesidades de caudal y pérdidas de carga calculadas, tal y como se ha comentado al inicio del capítulo.

### 4.4.1. Filtros

Los filtros en un sistema de ventilación permiten introducir y extraer el aire en una estancia con la calidad adecuada, libre de partículas y compuestos que puedan ser nocivos para las personas. Conocidas las calidades del aire exterior e interior, ODA 1 e IDA 3 respectivamente, se pueden seleccionar los filtros adecuados para la instalación. Cabe destacar el cambio reciente en la normativa de los filtros, quedando anulada la norma UNE EN 779 en favor de la norma UNE EN 16890, durante el año 2017. En la norma anulada, aún presente en el RITE, se especificaba la clase de filtro, mientras que la norma actual los agrupa por grupos de filtros en función de su capacidad de retención. En la Tabla 4.5 se muestra la correspondencia entre las dos normas [22]. Los tipos de filtros necesarios en la instalación se van a explicar en función de la norma anulada debido a que es la recogida en el RITE, no obstante, los filtros seleccionados sí cumplirán con la nueva legislación.



#### 4. VENTILACIÓN

|                         |        | UNE EN 16890 | Porcentaje de retención | UNE EN 779 |
|-------------------------|--------|--------------|-------------------------|------------|
| Filtros de polvo grueso | Grueso |              | 40%                     | G1         |
|                         |        |              | 50%                     | G2         |
|                         |        |              | > 60%                   | G3         |
|                         |        |              | > 60%                   | G4         |
| Filtros de polvo finos  | ePM1   |              | < 20%                   |            |
|                         |        |              | < 40%                   | M5         |
|                         |        |              | ≥ 50%                   |            |
|                         | ePM2.5 |              | < 40%                   |            |
|                         |        |              | 50-60%                  | M6         |
|                         |        |              | > 60%                   |            |
|                         | ePM10  |              | 50-70%                  |            |
|                         |        |              | > 65%                   | F7         |
|                         |        |              | > 80%                   |            |
|                         | ePM1   |              | 70-80%                  |            |
|                         |        |              | > 80%                   | F8         |
|                         |        |              | > 90%                   |            |
| ePM2.5                  |        | > 80%        |                         |            |
|                         |        | > 90%        | F9                      |            |
|                         |        | > 95%        |                         |            |

*Tabla 4.5.- Correspondencia entre norma UNE EN 16980 y norma UNE EN 779.*

El RITE indica la necesidad de incorporar prefiltros en la UTA si la instalación cuenta con recuperador de calor [1], como muestra el subapartado 4.4.2 el sistema de ventilación descrito debe llevarlo, por lo tanto, en primer lugar, se han de seleccionar los prefiltros, que serán instalados antes del ventilador y estarán en contacto directo con el aire exterior. La normativa indica que estos prefiltros deben ser de la clase F6 o M6 (corresponden al mismo tipo de filtros) [1].

Existen diferentes tipos de filtros que cumplirían con la normativa, pero para esta clase de filtrado lo más habitual, debido a su precio, son los filtros de bolsa sintéticos, los cuales están fabricados con fibras sintéticas y cuentan con una forma cónica que favorece el flujo de aire a través de ellos [23].

El prefiltro seleccionado es el modelo Bio Bag 60 de la marca Venfilter, Figura 4.4, el cual cumple con la clase M6 de la norma UNE EN 779, incluso su eficiencia en ePM10 es superior al 70%. El modelo está fabricado con fibras de polipropileno que incorporan un prefiltro sintético, las bolsas tienen forma cónica y cuenta con un marco de acero galvanizado [24]. El modelo permite elegir las dimensiones y el número de bolsas que llevará instaladas, para el proyecto se ha seleccionado el tamaño de 592x592x500 mm y ocho bolsas que permite el paso de 3400 m<sup>3</sup>/h y tiene una pérdida de carga inicial de 63 Pa [24]. Por lo tanto, han de instalarse dos prefiltros de este modelo para permitir el paso del caudal de aire calculado de 6096 m<sup>3</sup>/h.



Figura 4.4.- Prefiltro Bio Bag 60 (imagen obtenida de la web de Venfilter [24]).

A continuación del ventilador se debe instalar el filtro que cumpla con los requisitos de calidad del aire interior (IDA 3) en función de la calidad del aire exterior (ODA 1). Según dictamina el RITE, para tales calidades, ese filtro debe ser de clase F7 [1]. Esta clase de filtro es la misma que se instalará antes del ventilador de extracción, para evitar acumulaciones de partículas provenientes del interior de la sala ya sea en el ventilador o en el recuperador de calor.

Nuevamente, como en el caso de los prefiltros, los filtros de bolsa destacan por su relación calidad-precio. En este caso se ha seleccionado un filtro de bolsas de fibra de vidrio, se construyen igual que los filtros convencionales sintéticos, pero están fabricados con fibra de vidrio gruesa y fina entretrejida, permitiendo un mayor porcentaje de retención de partículas que los convencionales [25].

El filtro elegido es el modelo Bag-glass 80, Figura 4.5, que está desarrollado por la compañía Venfilter. Este modelo cumple con la clase F7 de la norma UNE EN 779, incluso supera para partículas en eficiencia a lo establecido por la norma ya que supera el 70% de retención en partículas ePM1. Las bolsas están fabricadas en fibra de vidrio, son, también, de forma cónica para mejorar el flujo de aire a través de ellas y están encajadas en un marco de acero galvanizado. Las dimensiones de este modelo serán 592x592x635 mm y llevan instaladas ocho bolsas, permitiendo el paso de 3400 m<sup>3</sup>/h y con una pérdida de carga inicial de 98 Pa [25]. Nuevamente, como en el caso de los prefiltros, será necesaria la instalación de dos de estos filtros tanto para el circuito de impulsión como para el de retorno.



Figura 4.5.- Filtro Bag-glass 80 (imagen obtenida de la web de Venfilter [25]).

### 4.4.2. Recuperador de calor

El recuperador de calor se encarga de calentar el aire proveniente del exterior con el aire que es extraído del interior de la estancia para poder ser introducido en unas condiciones termohigrólicas mejores y así, consumir menos energía en climatizar ese aire exterior. El RITE establece la obligatoriedad de su instalación si el caudal de ventilación supera los 0,5 m<sup>3</sup>/s [1], en el presente caso es de 1,69 m<sup>3</sup>/s, como se calculó en el subapartado 4.3.1 de los cálculos de ventilación.

## 4. VENTILACIÓN

---

Los recuperadores de calor son, al fin y al cabo, intercambiadores térmicos en los cuales por un lado pasará el aire exterior y por el otro el aire interior intercambiando calor entre ellos. Los tipos de intercambiadores más habituales son los de placas y los rotativos.

Los intercambiadores de placas están constituidos por una carcasa (abierta en los extremos) que, en su interior cuenta con una estructura celular. Son empleados cuando se requiere una gran superficie de intercambio y poco espacio. Sin embargo, su característica más notable, en comparación con los rotativos, es que nunca se encuentran en contacto los dos fluidos [26].

Los intercambiadores rotativos están formados por una carcasa que contiene una rueda o tambor en su interior, construida con un material permeable al aire, que se encuentra girando. Las dos corrientes de aire necesitan estar en contacto para intercambiar el calor, por ello es muy importante el filtrado previo en este tipo de recuperador y la sección de purga que incorporan [26].

La eficiencia mínima y la máxima pérdida de presión en los recuperadores de calor la establece el RITE en función de las horas de funcionamiento y del caudal de ventilación de la instalación [1]. Las horas de funcionamiento a lo largo del año serán inferiores a 2000 h, ya que en el salón de actos estudiado no se ofrecen conferencias consecutivas ni, incluso, diarias. Y el caudal de ventilación que se ha calculado en el subapartado 4.3.1 es de 1.69 m<sup>3</sup>/s. Por lo tanto, la eficiencia mínima del recuperador ha de ser de 44% y la máxima pérdida de presión de 120 Pa, según indica la normativa [1].

El recuperador de calor escogido es de placas, está diseñado por la empresa SystemAir y fabricado en aluminio [27]. Este recuperador tiene una eficiencia del 56,6% y unas pérdidas de carga de 32,5 Pa para el circuito de impulsión de aire y de 31,9 Pa para el de extracción. Como se puede observar cumple perfectamente con los requisitos marcados por la normativa. Debido a la temperatura media de Gijón en verano es similar a la del interior de la estancia, 23,5 °C en el exterior [18] y 21 °C en el interior, el recuperador de calor no trabajará durante esos meses de verano. Una imagen de este tipo de recuperadores de calor se muestra en la Figura 4.6.



*Figura 4.6.- Recuperador de calor de placas de SystemAir (imagen obtenida de su catálogo [27])*

### 4.4.3. Ventiladores

Los ventiladores se emplean para transportar el aire a través de conductos. En el sistema de ventilación se utilizan para la impulsión y extracción de aire hacia y desde el interior del salón de actos.

Los dos tipos más habituales de ventiladores que se suelen instalar son los axiales y los centrífugos, deben su nombre a la dirección del flujo de aire a través de ellos. Los ventiladores

#### 4. VENTILACIÓN

---

axiales se suelen emplear en sistemas de ventilación que presenten pequeñas pérdidas de carga, mientras que los centrífugos son los más utilizados cuando se requieran mayores presiones y la instalación cuente con pérdidas de carga superiores [3].

En el caso del sistema de ventilación estudiado, éste cuenta con filtros y recuperadores de calor que provocan mayores pérdidas de carga que los conductos. Por ello, el ventilador centrífugo será la opción escogida para la instalación, tanto para la impulsión como para la extracción, ya que permitirá vencer todas las pérdidas de carga que va a haber a lo largo del circuito de ventilación [3].

Dentro de los ventiladores centrífugos se diferencian entre sí en función de la inclinación de los álabes, éstos pueden estar inclinados hacia atrás, ser rectos o estar inclinados hacia delante, en función de la cantidad de partículas que hubiera en el aire [3]. Como la instalación a estudiar no trabaja en entornos polvorientos el ventilador centrífugo escogido tendrá los álabes inclinados hacia atrás.

El último criterio de clasificación de ventiladores centrífugos es a partir de la presión estática del circuito, esto distingue ventiladores de baja presión, hasta 50 mmca, de media presión, entre 100 y 200 mmca, y de alta presión, pueden superar los 300 mmca [3]. La presión estática que se ha calculado en el subapartado 4.3.2 para el circuito de impulsión es de 455,96 Pa, equivalentes a 46,48 mmca, por lo tanto, este ventilador se clasificará entre los de baja presión. Mientras que la presión estática calculada, en ese mismo subapartado, para el circuito de extracción ha sido de 314,41 Pa, equivalente a 32,05 mmca, por ello, este segundo ventilador también se engloba en la categoría de de baja presión.

A todo lo anterior cabe añadir que el ventilador ha de proporcionar el caudal de ventilación adecuado en cada momento, por ello debe ser capaz de funcionar mediante un variador de frecuencia que regule su velocidad y, además, no varíe, significativamente, su rendimiento con respecto al de diseño al realizar esta regulación. Por ello, los motores de los ventiladores que se vayan a seleccionar serán trifásicos ya que este tipo de motores son los habituales para el control mediante el variador [3].

En el caso de la impulsión de aire el modelo de ventilador centrífugo BVC-M 15/15 1,5 kW 950 de baja presión fabricado por la empresa Casals cumple con los criterios de selección mencionados en este subapartado. Dicho ventilador consigue, según su curva característica, un caudal de 6175,08 m<sup>3</sup>/h, puede vencer una presión estática de hasta 467,91 Pa y tiene una eficiencia para el punto de funcionamiento del proyecto de 51,52% [19], por lo que se ajusta perfectamente a las exigencias. Se puede ver una imagen de este modelo de ventilador en la Figura 4.7.



Figura 4.7.- Ventilador centrífugo BVC-M 15/15 1,5 kW 950 de Casals (imagen obtenida su web [19]).

---

## 4. VENTILACIÓN

---

Por otro lado, para el caso de la extracción del aire interior de la estancia se ha seleccionado un ventilador centrífugo de baja presión para cumplir con las exigencias, el modelo elegido es el BSTR-M 22/11 1,1 kW 550 fabricado, también, por la empresa Casals. Este modelo, al entrar en su curva característica, se obtiene un caudal cercano al máximo necesario de 6270,03 m<sup>3</sup>/h, permite vencer unas pérdidas de carga estáticas de 333,24 Pa y una eficiencia para el punto de funcionamiento del proyecto del 56,69% [20], ajustándose así a los valores requeridos. La Figura 4.8 muestra cómo sería físicamente este modelo de ventilador.



Figura 4.8.- Ventilador centrífugo BSTR-M 22/11 1,1 kW 550 de Casals (imagen obtenida de su web [20]).

### 4.5. Difusores y rejillas

Los difusores son los elementos terminales del circuito de impulsión, deben ser capaces de liberar al ambiente interior el caudal de aire necesario para la ventilación de la estancia. Estos estarán colocados sobre los conductos de tal forma que permitan la expulsión del aire sin provocar corrientes que incomoden a los ocupantes de la sala [1].

Los difusores de tobera WDA-R-175 de la marca Schako, Figura 4.9, permiten la difusión del aire aplicando una con un caudal por unidad de 1100 m<sup>3</sup>/h por lo que sería necesaria la instalación de seis toberas para cumplir con los requisitos de ventilación calculados. Dichas toberas tienen un diámetro de salida de aire de 175 mm, permitiendo una gran difusión de aire a baja velocidad, alcanzando los 15 metros con una velocidad terminal del aire de 0.8 m/s. Aunque estas toberas permiten regulación de apertura, al controlarse la velocidad de giro del motor no va a ser necesaria esta regulación, por lo que se considera que permanentemente estarán abiertas al 100%, para este porcentaje de apertura las pérdidas de carga que inducen son muy pequeñas de 10 Pa por tobera. La tobera está fabricada con aluminio lacado en color blanco [28].



Figura 4.9.- Difusor de tobera WDA-R-175 de Schako (imagen obtenida de la web de Schako [29]).

Las rejillas de extracción permiten el paso del aire interior de la estancia hacia el circuito de extracción [1]. La pérdida de carga que provocan estas rejillas ha de ser la menor

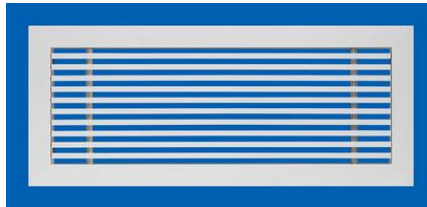
---

#### 4. VENTILACIÓN

---

posible y, a su vez, contar con las dimensiones adecuadas para poder evacuar el caudal de aire de extracción requerido.

Las rejillas E-LO/AD de la compañía Euroclima, Figura 4.10, permiten este cometido. Las rejillas de este modelo que se seleccionan son de 400 mm de altura y 800 mm de longitud, cada rejilla permite la extracción de 3250 m<sup>3</sup>/h de aire, por lo que van a ser necesarias dos rejillas para extraer el caudal de aire calculado en este proyecto de 6096 m<sup>3</sup>/h. A su vez, permiten la entrada de aire con una velocidad de hasta 5,7 m/s y tienen unas pérdidas unitarias de 20,6 Pa. Están fabricadas de aluminio extruido con acabado anodizado blanco [30].



*Figura 4.10.- Rejilla de extracción E-LO/AD de Euroclima (imagen obtenida de la web de Euroclima [31]).*

Finalmente, la UTA puede contar con unas rejillas o compuertas a la entrada y salida del aire, cuyo objetivo principal es evitar la entrada de animales o grandes partículas que hubiera en el aire exterior y que podrían ocasionar una avería en alguno de los módulos de ésta.

## 5. Sistema de adquisición de datos

En este capítulo se detallará el sistema de adquisición de datos ambientales que se utilizará para la monitorización del sistema de ventilación, se abordará cuáles son los elementos que lo componen y cómo se realiza la interconexión entre ellos. Este sistema de adquisición de datos consistirá en un microcontrolador que obtendrá periódicamente información de los sensores. La comunicación entre el microcontrolador y los sensores se puede realizar de varias maneras. Una de ellas es utilizando un conversor analógico-digital. Otra bastante común (y que será la escogida en este proyecto) es el protocolo I<sup>2</sup>C.

En primer lugar, se expondrán los microcontroladores, cuyas funciones principales serán solicitar el dato al sensor y enviarlo, una vez recibido, al bróker del protocolo de comunicación MQTT. Para la elección del microcontrolador adecuado se empezará comentando los componentes más habituales que suelen albergar y a partir de ellos su categorización básica. Una vez conocidas las características comunes de los microcontroladores se verán las posibilidades que ofrece el mercado de los microcontroladores y se procederá a la selección del más adecuado en función de una serie de parámetros, que van desde su capacidad hasta su precio. Después, una vez seleccionado el microcontrolador, se detallan todas sus características, incluyendo los componentes intrínsecos que incorpora, así como cada una de las conexiones disponibles en el microcontrolador a través de sus pines.

En el siguiente punto se analiza la placa de desarrollo, que se empleará principalmente como interconexión entre el microcontrolador y el sensor y, como intermediario en la alimentación eléctrica de ambos. Por ello, simplificando el circuito electrónico, se escogerá una placa de desarrollo que incorpore el microcontrolador elegido como uno de sus componentes. Consecuencia de lo anterior, será en este caso la placa la que ha de conectarse, a través de sus pines, físicamente con el sensor mediante el protocolo I<sup>2</sup>C. Además, como último apunte, la placa tiene que ser reconocida por el programa Arduino, que como se verá al final de este capítulo, será el utilizado para realizar la aplicación de monitorización de la calidad del aire. Conocidos estos requerimientos, se seleccionará la placa que los aglutine todos, se estudiará pormenorizadamente el resto de sus componentes y se verán sus características intrínsecas.

Después de analizar la placa de desarrollo se estudiará el bus I<sup>2</sup>C, en concreto las características propias de su protocolo de comunicación, también denominado I<sup>2</sup>C. Este protocolo se emplea para transferir la información en el sistema de adquisición de datos entre los sensores y la placa de desarrollo. Debido a la elección de este protocolo los componentes que se vayan a seleccionar en este capítulo han de ser compatibles con él.

A continuación, se va a realizar un breve estudio de mercado con las diferentes opciones existentes para la medición de la calidad del aire interior, desde sensores unitarios, pasando por los módulos de sensores hasta llegar a equipos más complejos de monitorización de la calidad del aire. Una vez obtenida la información básica de cada uno de ellos se escogerá el más conveniente para este proyecto, en función de los parámetros que se desee controlar, su rango de medición, la precisión del sensor y el coste de éste. Del sensor seleccionado se expondrán todas sus características y se estudiarán los pines que incluye, identificando cuales de ellos van a ser necesarios para realizar la conexión I<sup>2</sup>C. El subapartado se cerrará mencionando las reglas de posicionamiento de este tipo de sensores ambientales sobre las paredes de la estancia, dicha posición como se verá se halla a la altura de las vías respiratorias de los ocupantes de la sala.

Finalmente, se explicará la conexión física que se va a realizar entre el sensor elegido y la placa de desarrollo, es decir, los pines de ambos elementos que será necesario enlazar para llevar a cabo la conexión entre ellos. Esta conexión no sólo deberá incluir la concerniente al bus I<sup>2</sup>C, para la transmisión de los datos recogidos por el sensor, sino que se además se debe tener en cuenta que la alimentación de éste también será llevada a cabo desde la placa. Además, se explicará cómo se realiza el direccionamiento de la información del protocolo I<sup>2</sup>C entre estos

dos elementos. Se mostrará el registro donde se almacenan los datos y la posición en él de los valores de las mediciones que hayan sido realizadas.

### 5.1. Elección del microcontrolador

Un microcontrolador es un circuito integrado programable que ejecuta las órdenes programadas en su memoria. Estos microcontroladores suelen albergar una serie de componentes para llevar a cabo las órdenes programadas, algunos de los componentes que se pueden encontrar en un microcontrolador son [32]:

- Memorias: son necesarias para almacenar las órdenes, variables y funciones que se deban ejecutar en el microcontrolador.
- Periféricos de entrada-salida: para la interacción del microcontrolador con los diferentes elementos externos con los que ha de conectarse.
- Temporizadores: es necesario el control temporal a la hora de realizar las órdenes.
- Interrupciones: capaces de activar o paralizar las órdenes en caso de que suceda un evento.
- Convertidores ADC: realizan conversiones analógico-digitales en el microcontrolador.
- Módulo WiFi: para la conexión a internet del microcontrolador. Este módulo no es habitual encontrarlo en los microcontroladores lo que provoca que la elección del microcontrolador se estreche bastante.

Los microcontroladores pueden dividirse en categorías en función de su memoria, su número de bits, el tipo de arquitectura y su conjunto de instrucciones [33].

- 1) Según su memoria se pueden categorizar en:
  - a. De memoria externa: cuando el microcontrolador no forma parte de los bloques funcionales del chip.
  - b. De memoria incorporada o incrustada: cuando el microcontrolador sí forma parte de los bloques funcionales del chip.
- 2) Según el número de bits:
  - a. De 8 bits: ejecuta operaciones aritméticas y lógicas, empleado para baja potencia por su facilidad de uso [34].
  - b. De 16 bits: mayor rendimiento y memoria que el de 8 bits, empleado habitualmente en aplicaciones de consumo e IoT [35].
  - c. De 32 bits: son requeridos para redes y comunicaciones, para operaciones matemáticas de punto flotante, y aritmética avanzada. Además, es necesaria una arquitectura más compleja que las dos opciones anteriores [35].
- 3) Según su arquitectura:
  - a. Von Neumann: en esta arquitectura la memoria para datos y programa es única. La longitud de palabra de las instrucciones está limitada a la de los datos, esto provoca gran cantidad de ciclos máquina para la ejecución de programas complejos ya que el acceso a la instrucción y a los datos debe hacerse de manera alternativa y no simultáneamente [36].
  - b. Harvard: en esta arquitectura la memoria de datos y la memoria del programa se encuentran separadas, esto permite utilizar diferentes longitudes de palabra para la instrucción y para el dato y trabajar en paralelo [37].
- 4) Según su conjunto de instrucciones:
  - a. CISC: “Complex Instruction Set Computer”, utiliza un gran conjunto de complejas instrucciones de lenguaje de máquina [38].



---

## 5. SISTEMA DE ADQUISICIÓN DE DATOS

---

- b. RISC: “Reduced Instruction Set Computer”, utiliza un conjunto reducido de instrucciones más sencillas, lo que reduce el tiempo máquina [38].

El mercado de microcontroladores ofrece una amplia gama de fabricantes y modelos para todo tipo de utilidades. Algunas de las compañías más fuertes en este sector son Microchip, Texas Instruments, Freescale, Motorola, ST o Intel, como se puede ver este sector está dominado por empresas norteamericanas, ya que es el lugar donde más desarrollo histórico ha tenido esta tecnología.

El microcontrolador que se vaya a escoger ha de interactuar con los sensores solicitándoles el dato con la medición que hayan recogido. Esta solicitud del dato se programará para que sea enviado al microcontrolador periódicamente, estableciéndose así una monitorización continua de la calidad de aire interior en la estancia. Los datos recogidos por el microcontrolador deberán ser enviados hasta un sistema de gestión, que, como se verá en el Capítulo 6, esta labor depende de un bróker que los interconecte. Para que los datos puedan llegar al sistema de gestión, el microcontrolador seleccionado deberá contar con un módulo WiFi incorporado. El motivo es que se ha escogido utilizar un protocolo de comunicaciones inalámbrico mediante redes WiFi para comunicar al microcontrolador con el bróker, explicado en el subapartado 6.1. El microcontrolador, a su vez, se encontrará embebido en una placa de desarrollo que se encargará de conectarlo con el resto de los elementos que van a formar parte del circuito físico.

El microcontrolador seleccionado es el modelo ESP8266, mostrado en la Figura 5.1, desarrollado por la empresa Espressif. Dicho microcontrolador forma parte de la familia de módulos WiFi basados en procesadores ARM [39]. Se ha buscado, para el proyecto, una placa de desarrollo en el mercado que incorpore todo lo necesario para programar y utilizar este microcontrolador (diodos de protección, regulador de tensión y grabador), esta placa es conocida como NodeMCU y se hablará de ella en detalle en el siguiente subapartado.

La CPU o procesador con el que cuenta el microcontrolador es Tensilica L106 de 32 bits y arquitectura RISC, lo que le permite una velocidad de trabajo máxima de 160 MHz, aunque su velocidad estándar de trabajo es de 80 MHz siendo esta la frecuencia a la que el microcontrolador puede ejecutar las operaciones. Además del procesador, el microcontrolador lleva incorporada una memoria RAM de 50 kB para un pequeño almacenamiento de datos con fácil acceso a las mismas. A su vez, cuando se requiera cargar programas externos el microcontrolador incorpora una tercera memoria SPI flash de 4 MB para tal cometido. La tensión de alimentación que soporta está entre los 2,5 y los 3,6 V, mientras que el consumo de corriente es de 80 mA durante su funcionamiento [39].

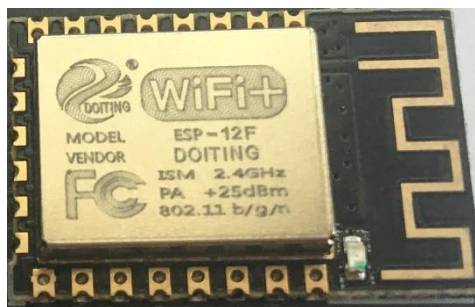


Figura 5.1. Módulo del microcontrolador ESP8266.

Este tipo de microcontroladores son SoC (“System on a chip”), lo cual quiere decir que llevan integrados varios de los módulos que pueda contener el aparato electrónico en un único circuito integrado o chip [40]. El microcontrolador ESP8622 incorpora, entre otros, un módulo WiFi 802.11 b/g/n de 2,4 GHz que permite su conexión a internet [39], en el caso del presente

## 5. SISTEMA DE ADQUISICIÓN DE DATOS

proyecto este módulo servirá para la interconexión inalámbrica entre el microcontrolador y el bróker.

A todo lo anterior, cabe añadir, que cuenta con una entrada analógica de 10 bits, 17 pines de entrada y salida GPIO de propósito general y un convertidor analógico-digital de 1 V como tensión máxima de entrada [39]. En la Figura 5.2 se puede ver el esquema del microcontrolador con la situación de los pines en él.

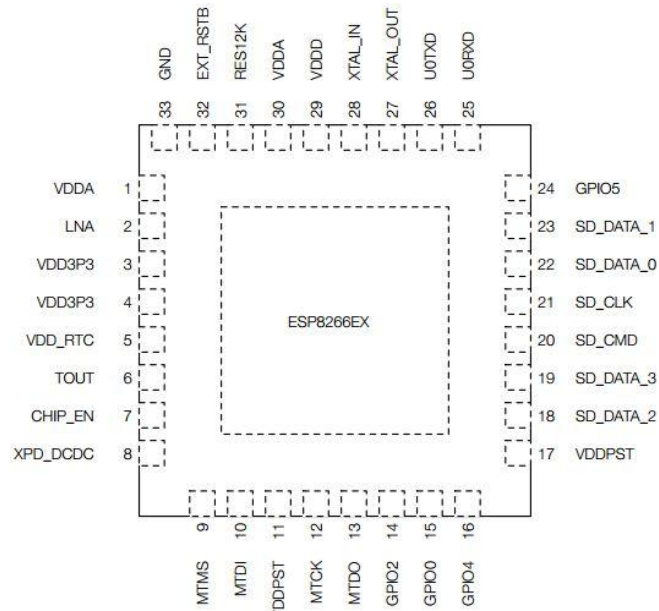


Figura 5.2.- Esquema del microcontrolador ESP8266 (imagen obtenida del "datasheet" del microcontrolador [39]).

### 5.2. Placa de desarrollo

La placa de desarrollo que se ha elegido es la denominada NodeMCU la cual lleva instalado el microcontrolador ESP8266, primer requisito indispensable para su elección. Esta placa integra la electrónica necesaria y la tecnología *Plug & Play*, por lo que no necesitan más que un cable USB y un ordenador para funcionar [41]. Además, actualmente, permite la integración con el IDE de Arduino [42], que será el utilizado para programar el microcontrolador en este proyecto. La Figura 5.3 muestra la morfología de esta placa de desarrollo.



Figura 5.3.- Placa de desarrollo NodeMCU con microcontrolador ESP8266.

Esta placa incluye además del microcontrolador una serie de módulos adicionales, que, aunque no realizan tantas funciones como el microcontrolador resultan necesarios para el correcto funcionamiento de la conexión con el módulo de sensores., ya que alguno de ellos va a

## 5. SISTEMA DE ADQUISICIÓN DE DATOS

estar implicado en la conexión con el módulo de sensores. Estos módulos incluidos en el NodeMCU van a ser [41]:

- Puerto USB-TTL: permite alimentar al NodeMCU así como programarlo desde un ordenador.
- Chip CH340G: es el conversor serie USB necesario para convertir el puerto USB en útil para la placa de desarrollo, gracias a él se puede programar, monitorizar y controlar el NodeMCU.
- Regulador de voltaje AMS1117: permite una salida de hasta 1 A y una regulación de tensión entre 1,5 y 5 V, con una diferencia entrada/salida de menos de 1 V, escogida mediante resistencias externas [43]. Se utilizará para alimentar el sensor al que esté conectado y al microcontrolador, ambos a 3,3 V.
- Botón de flash: se encuentra conectado a la salida 0 y en ocasiones permite la carga del firmware al NodeMCU, no será necesaria su utilización para este proyecto.
- Botón de reset: restablece el sistema del NodeMCU en caso de aparecer un evento indeseado en el mismo.

Los pines de conexión que incluye el NodeMCU serán los que aparecen en la Figura 5.4, la gran mayoría son coincidentes a los del microcontrolador, no obstante, en la Tabla 5.1 se especifican el resto de los pines que vienen incluidos en la placa y no coinciden con los del microcontrolador, junto con los que van a ser necesarios para la conexión con el bus I<sup>2</sup>C, ya que existe en algunos casos un cambio de nomenclatura que es necesario mencionar para entender la aplicación de Arduino creada en el subapartado 6.4.

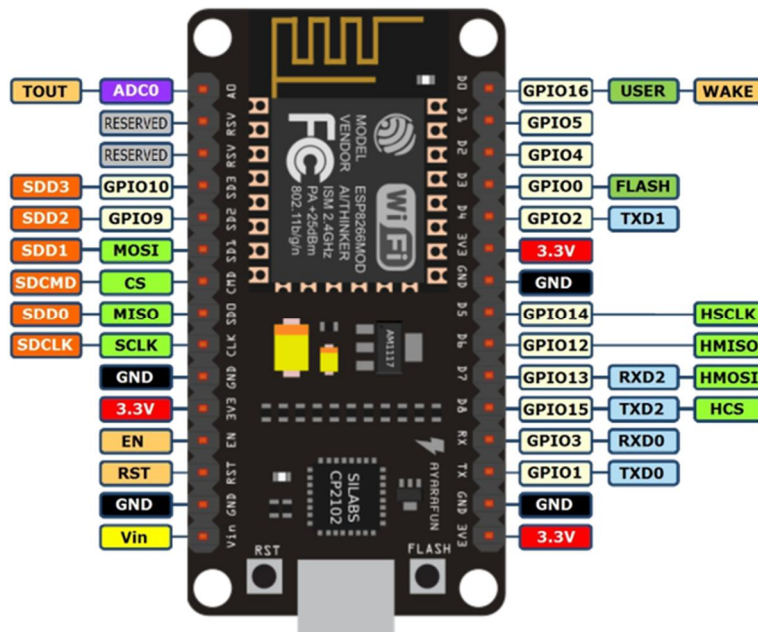


Figura 5.4.- Pines del NodeMCU.

| Nomenclatura del Pin | Funcionalidad                                |
|----------------------|--|
| A0                   | Entrada analógica                            |
| D6                   | SCL del bus I <sup>2</sup> C                 |
| D7                   | SDA del bus I <sup>2</sup> C                 |
| GND                  | Conexión a masa                              |
| 3.3V                 | Tensión de alimentación del microcontrolador |

Tabla 5.1.- Función de cada pin del NodeMCU utilizado para el proyecto.

La conexión física entre el NodeMCU y el sensor, como se ha comentado al inicio de este capítulo, se va a realizar mediante el bus I<sup>2</sup>C. Esto es debido a que el NodeMCU únicamente dispone de una entrada analógica, pero está limitada a un voltio, si recibiese una tensión mayor se dañaría [41]. Por ello, será necesario la conexión mediante el bus I<sup>2</sup>C a través de dos de las entradas digitales que incorpora la placa para dicho cometido.

### 5.3. Bus I<sup>2</sup>C

El bus I<sup>2</sup>C debe su nombre al protocolo síncrono de comunicación “Inter-Integrated Circuit” (I<sup>2</sup>C) que utilizan estos buses serie de transmisión de datos y han sido desarrollados por Phillips desde la década de 1980 [44].

Este tipo de buses utilizan dos cables, uno para el dato a transmitir, denominado Serial Data (SDA), y otro para crear la señal de reloj, llamado Serial Clock (SCL) [45]. Además de las dos líneas de transmisión de información, la de datos y la señal de reloj, es necesaria una tercera línea que actuara como referencia (GND). Excepto esta última línea de referencia, las otras dos utilizadas para la transmisión de datos son de drenador abierto, por lo que necesitan resistencias de “pull-up”. La velocidad de transmisión estándar es de 100 kbits/s, aunque permite velocidades de hasta 3,4 Mbits/s [45].

La arquitectura del bus I<sup>2</sup>C corresponde con la de maestro-esclavo, el maestro es el que inicia la comunicación con el esclavo solicitándole el dato o enviéndoselo. Por el contrario, los esclavos no pueden iniciar la comunicación con el maestro ni comunicarse entre sí [44]. En el caso del proyecto la función de maestro será ejercida por el microcontrolador y la de esclavo por los sensores ambientales. En la Figura 5.5 se puede ver un esquema de la arquitectura de un bus I<sup>2</sup>C.

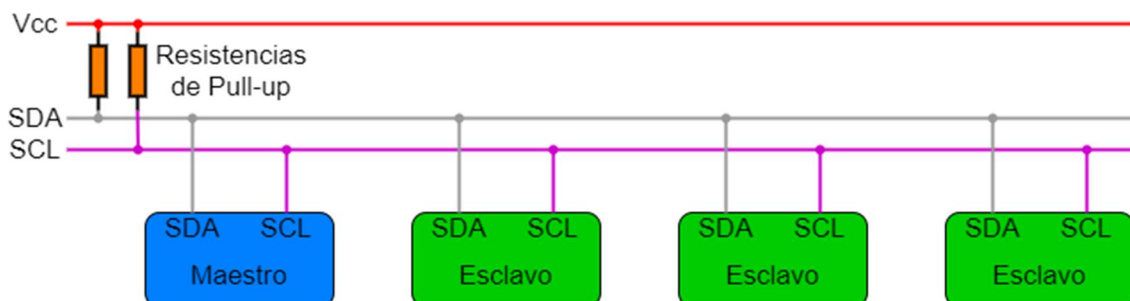


Figura 5.5.- Arquitectura del bus I<sup>2</sup>C.

La comunicación del bus I<sup>2</sup>C consta de siete bits de direccionamiento del esclavo con el que se ha de comunicar y un bit para indicar si se quiere recibir o enviar información (R/W), dicho bit será “alto” cuando deba enviar el dato al maestro y “bajo” cuando deba recibirlo.

Existen, a su vez, los bits de validación (ACK) que son generados por el esclavo al escribir y por el maestro al leer la información recibida [45]. Y por último lo más importante con lo que ha de contar la comunicación I<sup>2</sup>C son los ocho bits de datos que contendrán la información a transferir entre el maestro y el esclavo. Estos datos también contarán con su propio bit de validación, que actúa como acuse de recibo de la información enviada. Para el caso del presente proyecto, estos bits serán los datos ambientales recogidos por los sensores. La Figura 5.6 muestra un ejemplo de comunicación del bus I<sup>2</sup>C.

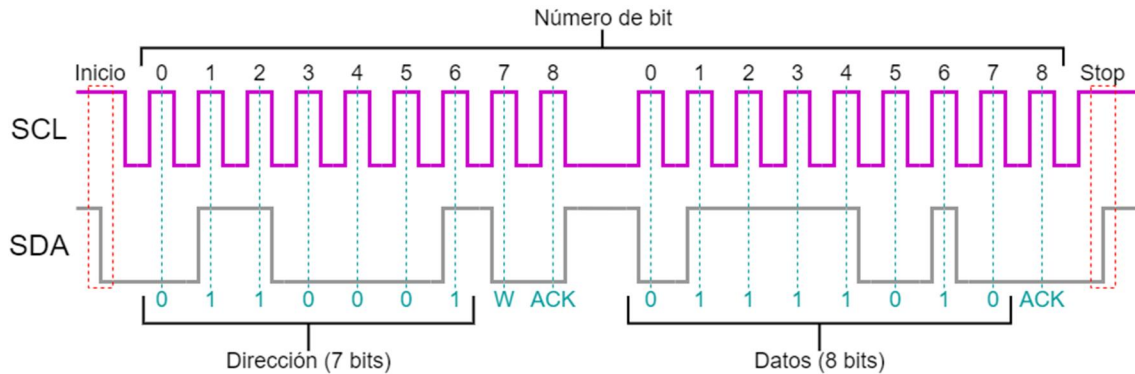


Figura 5.6.- Ejemplo de comunicación maestro-esclavo en el bus I<sup>2</sup>C.

La elección de este protocolo de comunicación es debida a su facilidad de uso, la alta velocidad de comunicación que ofrece y, por último, a que las distancias entre los diferentes dispositivos del sistema de adquisición van a ser cortas.

Los sensores que se escogerán en el siguiente subapartado, 5.4, deberán ser capaces de comunicarse mediante de esta tecnología a través de este tipo de buses, ya que actualmente existe una estandarización de esta tecnología en los aparatos electrónicos modernos [45], lo que hace muy interesante el empleo de esta tecnología para este proyecto.

### 5.4. Sensores disponibles en el mercado

Los sensores van a realizar la medición de diferentes parámetros de la calidad del aire ambiental en el interior de la estancia para poder regular, a partir de los datos recogidos, la velocidad de giro del ventilador. En este subapartado se realizará un estudio de los tipos de sensores existentes en el mercado, se analizarán los más interesantes para el caso de estudio y se comentará la situación que tendrán en la sala según la normativa vigente.

En el mercado existe una gran variedad de sensores, tanto analógicos como digitales, que permiten la medición de parámetros ambientales para determinar la calidad del aire en una estancia. El salón de actos de estudio está dedicado a impartir conferencias y ubicado en un edificio universitario, por lo tanto, la exposición a ambientes nocivos está descartada, por ello, los parámetros ambientales más interesantes a estudiar, y que determinarán la calidad del aire interior, son los siguientes:

- Niveles de concentración de CO<sub>2</sub> en el aire, es el parámetro utilizado para realizar la regulación, tal y como está recogido en el RITE [1].
- Temperatura, parámetro significativo en el bienestar térmico de los ocupantes de la estancia.
- Humedad relativa, al igual que en el caso de la temperatura, este parámetro influye en el bienestar térmico de los ocupantes.

Como se ha mencionado al inicio de este capítulo la comunicación entre el sensor y el microcontrolador se va a realizar a través del bus I<sup>2</sup>C. Debido a que el bus I<sup>2</sup>C conlleva un protocolo de comunicación específico, como se explicará en el subapartado 5.3, todos los sensores analizados han de ser compatibles con este protocolo.

A continuación, se realizará un breve estudio para conocer las diferentes opciones que ofrece el mercado para la medición de la calidad del aire en el interior de las salas y otros parámetros ambientales como puede ser la presión atmosférica o barométrica.

### 5.4.1. Sensor MG811

El primer sensor que se va a analizar es el sensor MG811, comercializado por Winsen, es un sensor analógico que mide los niveles de concentración de CO<sub>2</sub> en el aire. Este sensor especializado tiene un rango de medición entre 350 y 10000 partes por millón (ppm), su tensión de alimentación es 5 V, no obstante, el sensor necesita ser precalentado para funcionar correctamente, este precalentamiento se realiza a 6 V [46]. Para esta tensión de alimentación de alimentación de 5 V su consumo es de 140 mA [47]. La Figura 5.7 muestra cómo es este sensor.



Figura 5.7.- Sensor MG811 para mediciones de CO<sub>2</sub> (imagen obtenida del “datasheet” del sensor [47]).

### 5.4.2. Sensor SGP30

El siguiente sensor para analizar es el sensor SGP30, desarrollado por Adafruit, el cual permite la medición de la calidad del aire en el interior de estancias, en concreto la concentración de CO<sub>2</sub> y los compuestos orgánicos volátiles. Este sensor es digital y puede medir la concentración de CO<sub>2</sub> en un rango entre 400 y 60000 ppm, mientras que la de VOCs entre 0 y 60000 partes por billón (ppb). Su tensión de alimentación está comprendida en el rango entre 3 y 5 V y la corriente de alimentación cuando el sensor se encuentra realizando mediciones es de 50 mA [48]. En la Figura 5.8 se muestra una imagen de este sensor.

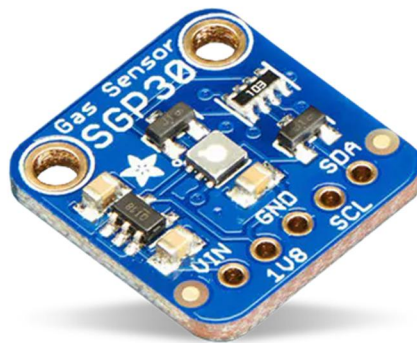


Figura 5.8.- Sensor SGP30 para mediciones de CO<sub>2</sub> y VOC (imagen obtenida del “datasheet” del sensor [48]).

### 5.4.3. Sensor CCS811

El sensor CCS811, desarrollado por AMS, permite la medición de la concentración del CO<sub>2</sub> y VOCs en el ambiente, el primero en un rango entre 400 y 8192 ppm y el segundo entre 0 y 1187 ppb. La tensión de alimentación se ha de establecerse entre 1,8 y 3,6 V y la corriente de alimentación es de 30 mA cuando realiza las mediciones [49]. En la Figura 5.9 se muestra cómo sería este sensor.



Figura 5.9.- Sensor CCS811 para mediciones de CO<sub>2</sub> y VOC (imagen obtenida del “datasheet” del sensor [49]).

### 5.4.4. Sensor Si7021

El sensor Si7021 permite la medición de estos dos factores ambientales, la temperatura y la humedad relativa. El rango de medición de la temperatura abarca desde los -40 a los 125 °C, mientras que la humedad relativa se mide en todo el rango posible, desde 0 hasta el 100%. La tensión de alimentación se encontrará entre los 1,9 y los 3,6 V, y el consumo de corriente máximo mientras realiza las mediciones será de 4 mA [50]. En la Figura 5.10 se puede ver una imagen de este sensor.



Figura 5.10.- Sensor Si7021 para mediciones de temperatura y humedad relativa (imagen obtenida del “datasheet” del sensor [50]).

### 5.4.5. Sensor DHT22

El sensor DHT22 realiza la medición de temperatura y humedad relativa en los rangos de -40 a 80 °C y de 0 a 100% respectivamente. La tensión de alimentación permitida se establece entre 3,3 y 6 V, mientras que la corriente consumida está comprendida entre 1 y 1,5 mA para dicho rango de tensiones [51]. En la Figura 5.11 se observa una imagen de este sensor.

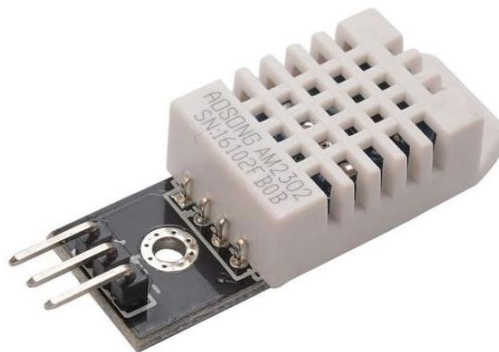


Figura 5.11.- Sensor de temperatura y humedad relativa DHT22 (imagen obtenida de una web especializada [52]).

### 5.4.6. Sensor BMP280

El sensor BMP280 realiza mediciones de presión atmosférica y temperatura, con unos rangos de medida de 300 a 1100 hPa y de -40 a 85 °C, respectivamente. Este sensor, también digital, tiene una tensión de alimentación mínima de 1,71 V y máxima de 3,6 V, mientras que la corriente máxima consumida está establecida en 1,12 mA durante la medición con la tensión máxima [53]. En la Figura 5.12 puede verse la imagen de este sensor.



Figura 5.12.- Sensor de presión y temperatura BMP280 (imagen obtenida del "datasheet" del sensor [53]).

## 5.5. Módulos de sensores comerciales

En el mercado, además de aparecer los sensores individualmente, se pueden encontrar diferentes sensores instalados conjuntamente que permiten realizar sus mediciones simultáneamente, estos sensores agrupados reciben el nombre de módulos. En el presente trabajo se ha decidido que además de monitorizar la concentración de CO<sub>2</sub> en el ambiente, se monitorice la temperatura y la humedad del salón de actos. Al ser una estancia ubicada dentro de un edificio dedicado a la enseñanza no se espera encontrar sólidos o gases en el aire del ambiente que perjudiquen la salud de las personas que allí se encuentren, ni una variación significativa de la presión atmosférica interior.

### 5.5.1. Módulo con sensores CCS811, Si7021 y BMP280

El primer módulo que se va a analizar incluye tres de los sensores vistos anteriormente, el CCS811 para la medición del nivel de concentración de CO<sub>2</sub>, el Si7021 para las mediciones de humedad relativa y temperatura y además incluye el sensor BMP280 con el cual se realizarían las mediciones de presión atmosférica, además de permitir una segunda medición de la temperatura. Este es un módulo no oficial, no existe ningún distribuidor oficial que lo comercialice, no obstante, se puede encontrar fácilmente en webs de comercio online, tales como Aliexpress, Amazon o Ebay. En la Figura 5.13 se observa este módulo con los tres sensores incorporados.



Figura 5.13.- Módulo de sensores BMP280, CCS811 y Si7021 (imagen obtenida de la web de Aliexpress [54]).



### 5.5.2. Módulo SCD30

El siguiente módulo para analizar es el SCD30 desarrollado por la compañía Sensirion, el cual incorpora también tres sensores para realizar las mediciones de humedad relativa, temperatura y concentración de CO<sub>2</sub> en el aire. Los sensores que incorpora este módulo para la medición de humedad y temperatura permiten el rango completo de 0 hasta el 100% para la humedad y, en cuanto a la temperatura, su rango incluye desde los -40 hasta los 70 °C. Mientras que el sensor que incorpora para la medición del CO<sub>2</sub> es el NDIR, el cual permite la medición de dicho compuesto desde los 400 ppm hasta los 10000 ppm. La tensión de alimentación para este módulo puede oscilar entre los 3,3 y los 5,5 V, mientras que la máxima corriente que va a necesitar cuando está realizando la toma de datos es de 75 mA [55]. Este módulo no está comercializado directamente por Sensirion sino que cuenta con distribuidores locales en función del continente, para Europa los distribuidores son Farnell, RS Components y An Avnet Company. La Figura 5.14 presenta una imagen de este módulo.



Figura 5.14.- Módulo SCD30 para medición de CO<sub>2</sub>, humedad relativa y temperatura (imagen obtenida del "datasheet" del módulo de sensores [55]).

## 5.6. Equipos comerciales de monitorización de la calidad del aire

Finalmente, existen equipos comerciales que permiten la monitorización continua de la calidad del aire en interiores. Este tipo de equipos incluyen, además de los sensores, unos sistemas de alerta luminosos indicativo de que uno de los parámetros monitorizados sobrepasa un cierto valor límite.

### 5.6.1. Equipo de monitorización de aire Nanoenvi IAQ

El equipo de monitorización de la calidad del aire Nanoenvi IAQ, desarrollado por la compañía Envira IoT, permite la monitorización de los siguientes parámetros ambientales, y se incluyen además sus rangos de medición [56]:

- Monóxido de carbono (CO):  $\pm 100$  ppm de error.
- Compuestos orgánicos volátiles (VOCs): entre 0 a 60000 ppb.
- Materia particulada (PM<sub>1,5</sub>, PM<sub>2,5</sub>, PM<sub>4</sub> y PM<sub>10</sub>): entre 10 y 100 ppm.
- Dióxido de carbono (CO<sub>2</sub>): entre 0 y 40000 ppm.
- Temperatura: entre los -40 y 70 °C.
- Humedad relativa: del 0 al 100%.
- Presión barométrica: entre 50 y 115 kPa.

Además, este equipo permite el almacenamiento de los datos recogidos en su memoria interna. Por el contrario, deben estar conectados a una fuente de corriente alterna de 240 V y requieren de una corriente de 180 mA en el momento de la toma de medidas [56]. En la Figura 5.15 se muestra un ejemplo de este equipo, como se puede observar cambia significativamente el aspecto exterior con respecto al resto de opciones anteriores.



Figura 5.15.- Equipo Nanoenvi IAQ para la monitorización de la calidad del aire (imagen obtenida de la web del fabricante [56]).

### 5.7. Selección de los sensores

Los equipos de monitorización de la calidad del aire son los aparatos más complejos vistos hasta el momento, pero también los más costosos, esto unido al gran número de mediciones de diferentes parámetros ambientales que no son relevantes para la sala de estudio, debido a su ubicación y utilización, es el motivo por el cual para este proyecto de estudio se descartan como la opción más viable. Por ello, en este caso las opciones más accesibles serán los módulos de sensores ya que se pretenden realizar diferentes mediciones ambientales y resulta ventajoso que ya hayan aparecido estos conjuntos de sensores en el mercado. Otra de las ventajas que incluyen estos módulos, los estudiados en el proyecto, es que todos sus sensores son digitales por lo que no será necesario el empleo de un convertidor analógico-digital para la lectura del dato. A su vez, estos módulos permiten la su implementación y control mediante el programa Arduino que será el utilizado en este proyecto.

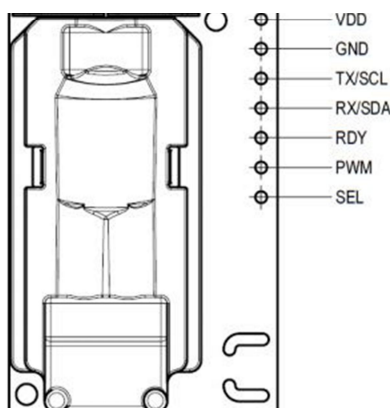
El módulo seleccionado para el proyecto es el SCD30 (ver Figura 5.14) desarrollado, fabricado y distribuido por la empresa Sensirion. Este módulo, como se comentó anteriormente, incorpora sensores ambientales de concentración del CO<sub>2</sub>, humedad relativa y temperatura. La elección de este módulo es debida a la mejor precisión de sus sensores en comparación al resto de opciones estudiadas, además su precio no es elevado. Las características de cada uno de estos sensores se especifican a continuación [55]:

- Sensor de concentración del CO<sub>2</sub> NDIR:
  - Tensión de alimentación: entre 3,3 y 5,5 V.
  - Consumo de corriente durante mediciones: 75 mA.
  - Rango de medición de la concentración del CO<sub>2</sub>: entre 400 y 10000 ppm.
  - Precisión de la medición:  $\pm 30$  ppm.
  - Temperatura ambiental de operación: entre 0 y 50 °C.
  - Humedad relativa de trabajo para no condensación: entre 0 y 95%.
- Sensor de humedad relativa:
  - Tensión de alimentación: entre 3,3 y 5,5 V.
  - Consumo de corriente durante mediciones: 75 mA.
  - Rango de medición de la humedad relativa: entre 0 y 100%.
  - Precisión de la medición:  $\pm 3\%$ .
  - Temperatura ambiental de operación: entre 0 y 50 °C.
  - Humedad relativa de trabajo para no condensación: no se condensa a ningún porcentaje de humedad relativa.
- Sensor de temperatura:
  - Tensión de alimentación: entre 3,3 y 5,5 V.
  - Consumo de corriente durante mediciones: 75 mA.
  - Rango de medición de la temperatura: entre -40 y 70 °C

## 5. SISTEMA DE ADQUISICIÓN DE DATOS

- Precisión de la medición:  $\pm 0,4$  °C.
- Temperatura ambiental de operación: entre -40 y 70 °C.
- Humedad relativa de trabajo para no condensación: no se condensa a ningún porcentaje de humedad relativa.

El módulo SCD30 dispone de siete pines de conexión, como se muestra en la Figura 5.16, destinados principalmente a la transmisión de información al sistema de gestión. El módulo permite el envío de esta información de tres maneras: las dos primeras son digitales, una mediante el protocolo I<sup>2</sup>C (la utilizada en este proyecto), y la otra empleando el protocolo Modbus, mientras que la tercera es mediante señal PWM. La funcionalidad de cada uno de los pines del módulo aparece reflejada en la Tabla 5.2 [55].



• Figura 5.16.- Esquema y pines del módulo SCD30 (esquema obtenido del “datasheet” de este módulo [55]).

| Nomenclatura del Pin | Funcionalidad  |
|----------------------|--|
| VDD                  | Alimentación del módulo  |
| GND                  | Conexión a tierra  |
| SCL                  | Señal de reloj, incluye resistencias de pull-up                                  |
| SDA                  | De comunicaciones, incluye resistencias de pull-up                               |
| RDY                  | Se activa cuando la medición ya ha sido realizada                                |
| PWM                  | Salida PWM con el valor de la concentración de CO <sub>2</sub>                   |
| SEL                  | Pin de selección de interfaz, se conecta a tierra para conexión I <sup>2</sup> C |

Tabla 5.2.- Nomenclatura y funcionalidad de cada pin del módulo SCD30.

El módulo elegido deberá instalarse en pared interior a una altura aproximada entre uno y dos metros sobre el nivel del suelo, coincidiendo con la entrada de las vías respiratorias de los ocupantes de la estancia. Además, se deben situar al menos medio metro alejado de puertas y ventanas, a un metro de las esquinas de la estancia y relativamente cerca de las rejillas de expulsión de aire, un mínimo de medio metro, siempre que sea posible [57].

### 5.8. Conexión entre el SCD30 y el NodeMCU

Una vez seleccionado el módulo de sensores, en este caso el SCD30, se procede a detallar como realiza las conexiones dicho módulo con el resto de los elementos presentes en la instalación electrónica.

Esta conexión se va a realizar mediante el bus I<sup>2</sup>C, el cual permite el envío de información, en este caso el valor de la medición, al microcontrolador de la placa de desarrollo cuando el dato le sea solicitado. Por lo tanto, el módulo ha de contar con una serie de pines destinados a este fin, serán los pines denominados SCL y SDA (en el subapartado 5.3 se especifica el significado de esta nomenclatura y su funcionamiento).

Por otro lado, el módulo necesita ser alimentado por una fuente de tensión y corriente externa que le proporcione la energía necesaria para que los sensores del módulo puedan realizar las mediciones ambientales. En este caso la alimentación del módulo será llevada a cabo desde el mismo NodeMCU a través del pin de 3.3 V por el que estarán también conectados. Como se ha visto en el subapartado 5.2, el NodeMCU incorpora un regulador de voltaje para tal fin, este regulador de voltaje permitía hasta 1 A de corriente de salida, el módulo SCD30 únicamente consume 75 mA mientras se encuentra realizando las mediciones por lo que no habrá problemas de suministro eléctrico. Existirá, por tanto, un pin de conexión que sirva como alimentación al módulo, denominado VDD en la Tabla 5.2.

El esquema de la conexión física entre los pines del módulo y los del NodeMCU, Figura 5.17, se ha realizado mediante el programa Fritzing, el cual permite interconectar los pines de ambos elementos de una forma muy visual. Este programa es de código abierto y accesible para cualquier persona, ofreciendo una herramienta de software para la creación de prototipos útiles tanto para la enseñanza como para el diseño y fabricación de PCBs para las empresas [58].

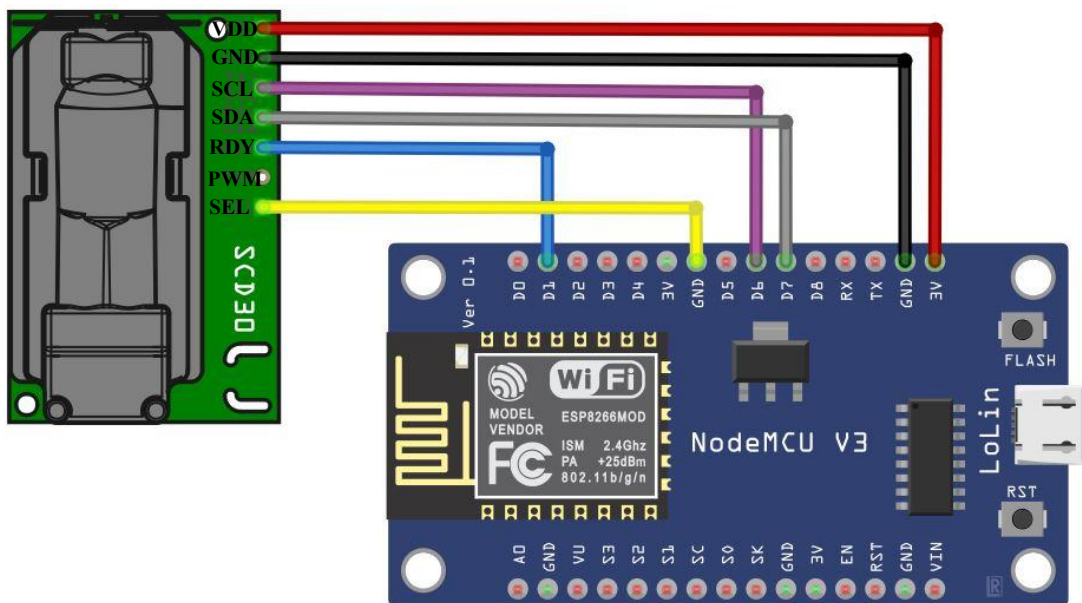


Figura 5.17.- Conexiones entre el NodeMCU y el módulo SCD30.

Como se ha comentado al inicio de este subapartado la alimentación del módulo será desde el propio NodeMCU, para ello, se empleará una de las salidas a 3.3 V con las que cuenta el NodeMCU, fijándose así la tensión de alimentación para la toma de medidas de los sensores del módulo en esos 3.3 V, que entrarán al módulo a través del pin VDD. Esta alimentación está representada en la Figura 5.17 mediante el cable rojo.

La conexión mediante el bus I<sup>2</sup>C requiere un mínimo de cuatro líneas, una para la señal de reloj, otra para la transmisión de la información (en el subapartado 5.3 se explica

pormenorizadamente la transmisión de información utilizando este bus I<sup>2</sup>C), una tercera que actuará como referencia o masa y la de alimentación antes ya mencionada, por lo tanto, el módulo empleará los pines denominados SCL, SDA y GND, respectivamente. Estos pines estarán unidos a través del bus con los pines D6, D7 y GND, los dos primeros son los específicos del NodeMCU para llevar a cabo la conexión I<sup>2</sup>C. Como se verá en el subapartado 5.3, las líneas conectadas a los pines SCL y SDA necesitan resistencias de “pull-up” para su funcionamiento, no obstante, como se ha mencionado en la Tabla 5.2, en el módulo de sensores elegido ya se encuentran instaladas de serie. En la Figura 5.17 estos pines están unidos mediante los cables morado, gris y negro, respectivamente.

Cuando la medición está realizada se activa el pin denominado RDY, en ese momento el dato ya puede ser enviado cuando lo solicite el sistema de gestión. Existen dos opciones de actuación con este pin, la primera de ellas es no conectarlo al NodeMCU, esto implica que a través del bus I<sup>2</sup>C se pregunte al sensor correspondiente si la medida está realizada. Y la segunda opción, por la que se ha optado en este proyecto, es conectar el pin RDY a uno de los pines de entrada/salida digitales de propósito general disponibles en el NodeMCU, en este caso se ha decidido el D1, con ello se consigue que sea el microcontrolador de la placa el que observe ese pin, y cuando lo encuentre activo recoja la medida sin necesidad de preguntar al sensor. En la Figura 5.17 esta conexión es llevada a cabo a través del cable azul.

El empleo de la conexión I<sup>2</sup>C implica, que el pin PWM que permite la salida analógica de la medición no vaya a utilizarse y pueda dejarse sin conectar o flotante. A su vez, el “datasheet” del módulo indica que el pin denominado SEL ha de conectarse a tierra cuando se emplea I<sup>2</sup>C [55], como es el caso. En la esquematización de la Figura 5.17 esta conexión a tierra del pin SEL se realiza con el cable amarillo.

### 5.8.1. Direccionamiento de la información del protocolo I<sup>2</sup>C

Una vez visto como se comunican físicamente mediante el bus I<sup>2</sup>C, se pasará a ver como realiza el protocolo I<sup>2</sup>C el intercambio de información entre los dos elementos, el módulo de sensores y el microcontrolador del NodeMCU.

La comunicación en este protocolo es maestro-esclavo, como se ha visto en el subapartado 5.3, la función de maestro está ejercida por el microcontrolador que solicita los datos al módulo de sensores, que actuará como esclavo.

El envío de la información tiene lugar tal y como aparece en el flujograma de la Figura 5.18. En primer lugar, ha de configurarse la dirección en I<sup>2</sup>C del módulo de sensores, finalizado este paso el módulo ya estaría en condiciones de enviar los datos. Dicho envío no se realiza hasta que el pin RDY del módulo se activa, lo que indica que los sensores del módulo ya han realizado la medición, por lo tanto, es posible la lectura de estos datos por parte del microcontrolador.

## 5. SISTEMA DE ADQUISICIÓN DE DATOS

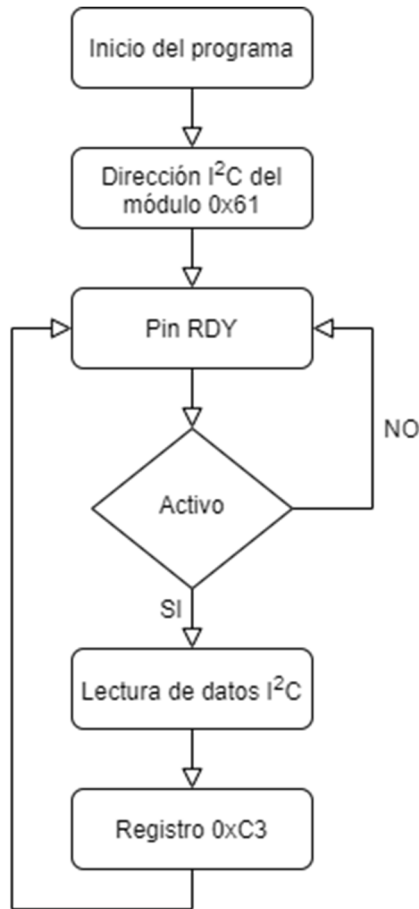


Figura 5.18.- Flujograma de lectura y envío de datos del módulo de sensores al microcontrolador.

Al solicitar el microcontrolador la lectura de los datos, estos se encuentran almacenados en el registro 0xC3 del módulo de sensores [59]. En ese registro aparecerán 18 bytes con los valores de las tres mediciones realizadas, así como un aviso de error en caso de que hubiera habido un problema a la hora de realizar la medición. Los primeros seis bytes corresponden al dato del nivel de CO<sub>2</sub>, los siguientes seis serán para el valor de la temperatura y los último seis para el de humedad relativa. Un ejemplo del almacenamiento en el registro de estos 18 bytes puede verse en la Figura 5.19, en este caso los datos obtenidos de la medición son de 439 ppm para el nivel de concentración de CO<sub>2</sub>, de 27,2 °C para la temperatura y de 48,8% para humedad relativa.

| Start | Read Header | CO2 MMSB | CO2 MLSB | CRC  | CO2 LMSB | CO2 LLSB | CRC  |
|-------|-------------|----------|----------|------|----------|----------|------|
| Start | 0xC3        | 0x43     | 0xDB     | 0xCB | 0x8C     | 0x2E     | 0x8F |

| T MMSB | T MLSB | CRC  | T LMSB | T LLSB | CRC  |
|--------|--------|------|--------|--------|------|
| 0x41   | 0xD9   | 0x70 | 0xE7   | 0xFF   | 0xF5 |

| RH MMSB | RH MLSB | CRC  | RH LMSB | RH LLSB | CRC  | Stop |
|---------|---------|------|---------|---------|------|------|
| 0x42    | 0x43    | 0xBF | 0x3A    | 0x1B    | 0x74 | Stop |

Figura 5.19.- Ejemplo de almacenamiento de datos en el registro (imagen obtenida del "interface description" del sensor [59]).

## 6. Comunicación entre sistemas

Una vez visto en el capítulo anterior los diferentes elementos que formarán el hardware del sistema de adquisición de datos, en este capítulo se abordará todo lo concerniente al software de éste explicándose los protocolos de comunicación utilizados para la conexión y, para finalizar el capítulo, se realizará una aplicación práctica de estos protocolos mediante el programa Arduino.

La primera parte de este capítulo estará centrada en conocer diferentes protocolos de comunicación que existen en la actualidad para llevar a cabo la tarea de transmisión de información. Entre ellos se escogerá el más adecuado para el proyecto en función de las características propias de cada uno.

Una vez seleccionado el protocolo de comunicación, que en este caso será el MQTT, éste se desarrollará exponiendo cómo comunica el sistema de adquisición de datos con el sistema de gestión para transferir información entre ellos. El protocolo de comunicación transmitirá el dato con la medición ambiental como un mensaje desde el sistema de adquisición hasta el sistema de gestión, que es el encargado de leer ese mensaje e interpretarlo para poder monitorizar la velocidad de giro del motor del ventilador. Para ello, este sistema de gestión volverá a transmitir un mensaje a través de este protocolo haciéndolo llegar al sistema de actuación, cuya función consiste únicamente en variar el régimen de giro del motor, en caso de ser necesario. Estos dos últimos sistemas se verán en el Capítulo 7.

Los siguientes subapartados estarán destinados a estudiar los diferentes elementos del software que conforman este protocolo MQTT, su estructura. Como los sensores, el microcontrolador y la placa de desarrollo ya fueron vistos en el capítulo anterior, para éste se estudiará en profundidad el bróker o nodo central de este protocolo.

El bróker, es el mediador del protocolo MQTT entre el sistema de adquisición de datos y el sistema de gestión del que depende el actuador. El bróker recibirá el mensaje con el dato de la medición ambiental, publicado en él bajo un cierto “topic” por el sistema de adquisición de datos, y será el encargado de transmitirlo al sistema de gestión mediante la publicación de un mensaje bajo otro cierto “topic”. Una vez el dato haya sido evaluado por el sistema de gestión éste enviará un nuevo mensaje bajo otro “topic” al bróker que lo transmitirá al sistema de actuación que este suscrito al “topic” que lo recibirá y actuará en consecuencia.

A continuación, una vez vistos descrito el bróker del MQTT, se procederá a realizar la aplicación que comunique el sistema de adquisición con el bróker. Para llevar a cabo esta aplicación se utilizará el programa Arduino que emplea lenguaje Wiring para su código.

El primer paso será conocer el programa Arduino y comentar el porqué de la selección de dicho programa para elaborar la aplicación de este proyecto. La aplicación utilizará el puerto USB-TTL que incorpora el NodeMCU para subir las instrucciones al microcontrolador desde un ordenador.

Finalmente, se comentará como se va a llevar a cabo la aplicación con Arduino argumentando las zonas que la compondrán y se incluirá en cada línea de código una pequeña aclaración de la funcionalidad de esa línea para el resultado conjunto. Al acabar la aplicación, ésta debe ser capaz de recoger el dato del sensor periódicamente, convertirlo en un mensaje que pueda ser enviado al bróker y enviarlo al bróker bajo un determinado “topic”. Conseguido esto, el bróker deberá publicar el “topic” con el mensaje para que el sistema de gestión suscrito a él lo pueda leer.

### 6.1. Protocolo de comunicación

Un protocolo de comunicación es un conjunto de pautas que posibilitan la comunicación entre diferentes elementos que forman parte de un sistema para que intercambien información. Estos protocolos determinan cual es la semántica y la sintaxis que deben emplearse en dicho proceso comunicativo, incluyen, también, ciertas reglas que permitan la recuperación de ciertos datos que se hayan podido perder en el intercambio de información [60].

Seguramente, al pensar en un protocolo de comunicación el primero en venir a la mente es el más universal y conocido de todos, el protocolo de comunicación de internet, denominado TCP/IP [61], cuyas siglas corresponden a “Transmission Control Protocol” e “Internet Protocol”, respectivamente. La actuación conjunta de estos dos protocolos posibilita el acceso al enlace entre los equipos que acceden a la red [60], Figura 6.1. No obstante, para este proyecto, aunque las conexiones inalámbricas sean realizadas gracias a este protocolo TCP/IP, el tipo de protocolo que se estudiará en profundidad corresponde a la forma de transmisión de la información entre los diferentes elementos que estén interconectados entre sí, desde los sensores hasta los actuadores.



Figura 6.1. Protocolo TCP/IP.

En este proyecto el protocolo de comunicación debe ser capaz de transmitir el dato que le vaya a llegar desde el sistema de adquisición al sistema de gestión que procederá a actuar en consecuencia al valor del dato recibido, ya que se encuentra en comunicación con el actuador que monitoriza la velocidad de giro del ventilador.

El mercado ofrece diferentes posibilidades para realizar esta comunicación, a continuación, en los siguientes subapartados, se verán las características de algunos de estos protocolos existentes, escogiéndose el más adecuado para el proyecto en función de las necesidades y los equipos que conforman la instalación.

Los protocolos que se van a estudiar serán el LoRaWAN, el ZigBee y el MQTT como representación de los protocolos de comunicación para IoT. De los dos primeros se analizarán sus características, mientras que el MQTT, al ser el elegido para el proyecto, se analizará enfocándose en la aplicación práctica del proyecto.

#### 6.1.1. LoRaWAN

LoRaWAN es un protocolo de red que utiliza la tecnología LoRa para para comunicar y administrar dispositivos LoRa, se compone de dos partes principalmente: “gateways”, antenas, y nodos, los primeros son los encargados de recibir y enviar información a los nodos y los segundos, son los dispositivos finales que envían y reciben información hacia el “gateway” [62], Figura 6.2. Las principales características de este protocolo son [63]:

- Topología en estrella [62].
- Conexiones bidireccionales con encriptación de extremo a extremo (AES 128 [62]).



## 6. COMUNICACIÓN ENTRE SISTEMAS

- Bajo consumo de energía.
- Largo alcance de comunicación (de 10 a 20 km).
- Conexiones de infinidad de sensores y equipos a redes públicas o privadas (hasta un millón de nodos en red).
- Baja velocidad de datos.
- Baja frecuencia de transmisión, movilidad y servicios de localización.
- Interoperabilidad de redes LoRaWAN en todo el mundo.

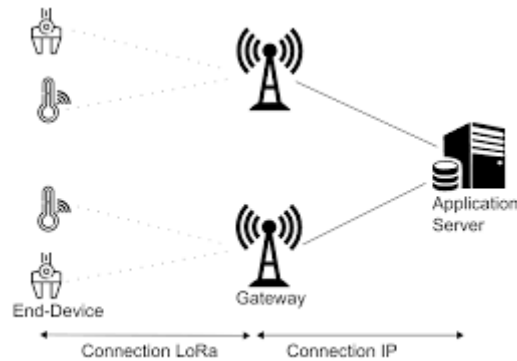


Figura 6.2.- Protocolo de comunicación LoRaWAN.

LoRa es un tipo de modulación en radiofrecuencia patentado por Semtech, esta tecnología de modulación se denomina “Chirp Spread Spectrum”, CSS, y se lleva utilizando en comunicaciones espaciales y militares desde hace décadas [63]. Las principales ventajas de la modulación LoRa son [62]:

- Alta tolerancia a las interferencias.
- Alta sensibilidad para recibir datos (-168 dB).
- Basado en modulación “chirp”.
- Conexión punto a punto.
- Frecuencias de trabajo: 915 MHz en América, 868 MHz en Europa y 433 MHz en Asia.

Las aplicaciones más comunes de la tecnología LoRa son conexiones a grandes distancias y para redes de IoT que empleen sensores sin disponibilidad de conexión a la red eléctrica. Siendo, por tanto, habitual su uso en ciudades inteligentes, lugares con poca cobertura y para construir redes privadas de sensores y actuadores [63], Figura 6.3.

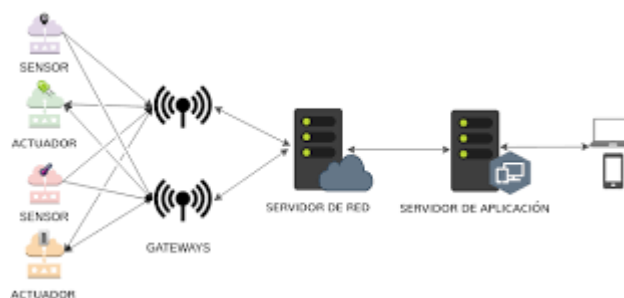


Figura 6.3. Funcionamiento de LoRaWAN con sensores y actuadores.

En este proyecto los sensores van a alimentarse eléctricamente a través del regulador de voltaje AMS1117 que incorpora el NodeMCU, además se contará con buena cobertura de red, por lo tanto, no se utilizará este protocolo en el proyecto.

### 6.1.2. ZigBee

ZigBee es un protocolo de comunicación inalámbrico de alto nivel desarrollado por ZigBee Alliance que emplea ondas de baja energía. Este protocolo se basa en el estándar IEEE 802.15.4 aprobado en el 2004 para las capas de baja comunicación [64]. Los componentes necesarios que emplea este protocolo, Figura 6.4, son [65]:

- Controlador: encargado de organizar las rutas, sólo puede existir uno por red.
- Router: su función es enlazar dos o más dispositivos que no estén conectados entre sí.
- Dispositivos finales: corresponden a los sensores y actuadores de la red.

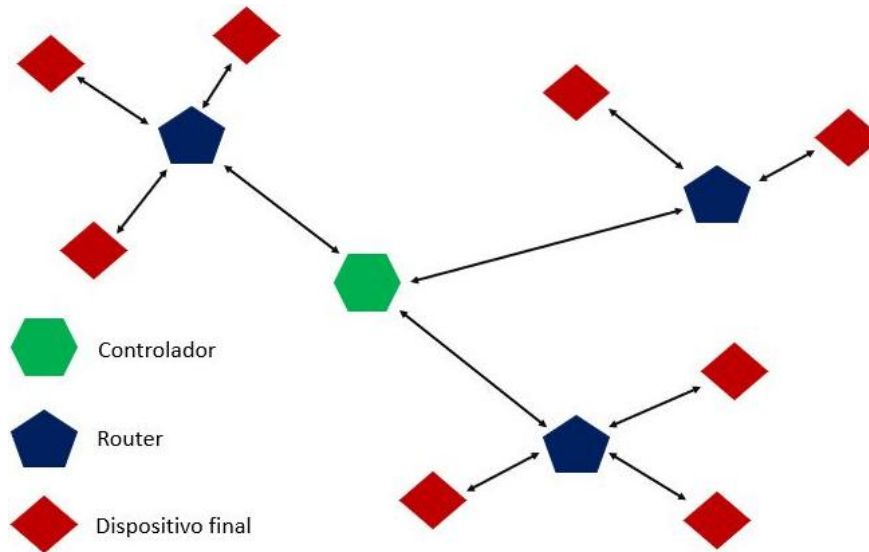


Figura 6.4. Protocolo de comunicación ZigBee.

Las características principales de este protocolo de comunicación son su bajo consumo y transferencia de datos. Además, emplea habitualmente una topología de tipo malla, puede adoptar una topología en estrella o en árbol en caso de ser necesario [65], y tiene un tamaño reducido [64]. Otras de sus características son [66]:

- Tamaño de red: hasta 65000 nodos.
- Banda de frecuencia de 2,4 GHz.
- Velocidad de datos de 250 kbits/s.
- Encriptación de extremo a extremo (AES 128).
- Comunicación hasta 300 m.
- Cada router puede admitir hasta 240 dispositivos finales.
- Frecuencia de trabajo: 868 MHz para Europa y 915 MHz para Estados Unidos [65].

Las principales ventajas de este protocolo frente a dos de sus competidores habituales, la tecnología WiFi y la tecnología Bluetooth, es el menor consumo y un hardware más reducido, lo que reduce también sus costes, que el empleado por la tecnología WiFi, y, frente a la tecnología Bluetooth, ofrece un mayor alcance y número de nodos que se pueden incorporar a una red [64]. Además, su topología tipo malla permite la reorganización de la red en caso de que algún nodo falle sin mayor afectación para el conjunto de la red [65].

Las aplicaciones habituales para esta tecnología están referidas al IoT, tanto a nivel doméstico como industrial, permitiendo que los equipos se comuniquen entre ellos y no directamente al WiFi lo que podría saturar la red [67].

Pese a que este protocolo sería perfectamente válido para llevar a cabo una solución de IoT, se ha decidido el empleo de otro protocolo de comunicación para el trabajo (el MQTT), aún más sencillo y liviano, ya que no habrá problemas de saturación de la red, que va a permitir su implementación mediante el programa Arduino, que será el empleado para escribir el programa de instrucciones que deberá subirse al microcontrolador.

### 6.2. Message Queue Telemetry Transport

El Message Queue Telemetry Transport (MQTT) es un protocolo de comunicación machine-to-machine (M2M) desarrollado originariamente por IBM durante la década de los 90 (aunque actualmente es un estándar abierto de OASIS) para soluciones de IoT construido sobre la pila TCP/IP [68].

Está diseñado como un transporte de mensajería de publicación/suscripción extremadamente liviano que conecta dispositivos remotos, principalmente sensores, con una huella de código pequeña y un ancho de banda de red mínimo. La arquitectura MQTT sigue una topología de estrella con un nodo central que actúa como servidor o bróker, y permite la conexión de hasta 10000 suscriptores o clientes. Dicho bróker se encarga de la gestión de la red y la transmisión de los mensajes. Hoy en día, MQTT se utiliza en una amplia variedad de industrias, como la automotriz, la fabricación, las telecomunicaciones, el petróleo y el gas, etc [69].

Además de las ya mencionadas, el protocolo MQTT incorpora una serie de características particulares, dichas características son [69]:

- Comunicaciones bidireccionales entre el bróker y el dispositivo, ya sea cliente o suscriptor.
- MQTT puede escalarse para conectarse a millones de dispositivos.
- Ofrece tres niveles de Calidad de Servicio (QoS) ante fallos [70]:
  - Nivel 0: “At most once”, el mensaje se envía una única vez, pero puede perderse ante fallos que ocurran. Se emplea cuando el mensaje no es crítico.
  - Nivel 1: “At least once”, el mensaje se envía las veces que sea necesario hasta que se garantice la entrega al destinatario. Éste puede recibir varias veces el mismo mensaje.
  - Nivel 2: “Exactly once”, se garantiza la entrega del mensaje una única vez. Se emplea cuando se necesita mayor fiabilidad en la entrega.
- Cifrado de mensajes mediante TLS y autenticación de clientes mediante protocolos de autenticación modernos.

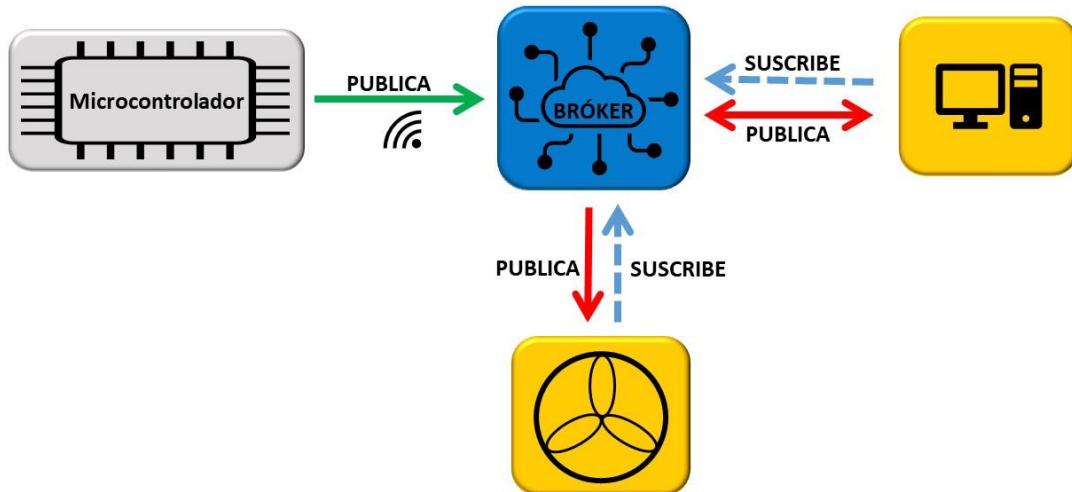
En base a estas características el protocolo MQTT tiene una serie de ventajas frente a sus competidores para IoT [70], estas ventajas son:

- Su escalabilidad, lo que le permite la conexión de millones de dispositivos.
- El desacoplamiento entre dispositivos, reduciendo su dependencia.
- Asincronismo en la lectura, recepción y respuesta a los mensajes entre las partes.
- Sencillez en los elementos que conforman el protocolo.
- Ligereza, ya que no consume demasiados recursos.
- Eficiencia energética de los equipos al no necesitar un gran ancho de banda.
- Seguridad y calidad, para una mayor fiabilidad y robustez en las comunicaciones.

El servicio de mensajería publicación/suscripción se puede asemejar al ámbito periodístico, en el cual un reportero envía un reportaje a una central de noticias y ésta la difunde a los diferentes medios de comunicación que estén suscritos a ella. En el caso de la estructura

## 6. COMUNICACIÓN ENTRE SISTEMAS

MQTT del proyecto son los sensores los que envían los datos de sus mediciones al bróker, a través del microcontrolador, que se encargará de enviarlos a los diferentes actuadores que estén suscritos a cada “topic”, permitiendo la monitorización de los equipos en función de los valores recogidos por los sensores. La Figura 6.5 esquematiza la estructura de este protocolo MQTT.



*Figura 6.5.- Estructura del protocolo MQTT.*

Los mensajes del protocolo MQTT tienen una pequeña huella de código, por eso es un protocolo ligero. La estructura de cada uno de estos mensajes consta de una cabecera fija, otra cabecera variable (que es opcional) y el contenido real del mensaje. La cabecera fija ocupa entre 2 y 5 bytes, de 1 a 4 bytes para codificar su longitud y otro más para el control, y es obligatoria en todos los mensajes. La cabecera variable se emplea en determinados mensajes, no obstante, no se utilizará para el proyecto y tiene una longitud indeterminada. El contenido del mensaje puede llegar a ocupar hasta 256 MB de información [71]. La Figura 6.6 muestra la estructura que tienen todos los mensajes de este protocolo MQTT.



*Figura 6.6.- Estructura de los mensajes del protocolo MQTT.*

La estructura física del protocolo MQTT constará por tanto de diferentes elementos que permitan la comunicación entre los sensores que realizan las mediciones y los actuadores que monitorizan la velocidad de giro del motor, estos elementos serán:

- Sensores: realizarán la toma de datos ambientales de la estancia cada cierto periodo temporal.
- Bus I<sup>2</sup>C: su función será la de transmisión de información entre el sensor y la placa de desarrollo.
- Microcontrolador: es el enlace entre el bróker y los sensores, solicitará el dato a los sensores y enviará un mensaje con dicho dato al bróker bajo un cierto “topic”.

- Placa de desarrollo: se encarga de alimentar tanto al microcontrolador como al sensor, e interconecta el microcontrolador con el resto del circuito físico que forman el sensor y el bus I<sup>2</sup>C.
- Bróker o nodo central: es el canalizador de toda la información, se encarga de recibir los mensajes enviados por el microcontrolador y el sistema de gestión y publicarlos con el “topic” correspondiente para que los sistemas suscritos a esos “topics” los reciban, ya sea el sistema de gestión o el de actuación
- Sistema de actuación: se trata en este caso de un sistema que incluya un variador de frecuencia para monitorizar el ventilador del sistema de ventilación. El valor del nivel de concentración del CO<sub>2</sub> en el ambiente será el que rijan como se ha de modificar la velocidad de giro del motor del ventilador.

Tanto los sensores, como el microcontrolador, la placa de desarrollo y el bus I<sup>2</sup>C están ampliamente analizados en el Capítulo 5, mientras que los sistemas de gestión y actuación se estudiarán en el Capítulo 7, por lo tanto, en este capítulo se analizará el nodo central o bróker del protocolo MQTT que es la base de este protocolo de comunicación.

### 6.2.1. Bróker

El bróker del MQTT es el intermediario entre el sistema de adquisición de datos, el sistema de gestión y el sistema de actuación. Su funcionamiento se basa en recibir la información procedente del cliente bajo un cierto “topic”, en forma de mensaje, y remitirla a los suscriptores que estén adscritos a dicho “topic” [42]. La Figura 6.7 representa el esquema de funcionamiento de un bróker.



Figura 6.7.- Bróker MQTT.

En este proyecto el papel de cliente será ejercido por el sistema de adquisición de datos, el sensor del sistema recogerá el dato ambiental y lo enviará a través del bus I<sup>2</sup>C hasta el microcontrolador del NodeMCU, el cual gracias a su módulo WiFi envía el dato bajo el “topic” correspondiente al bróker.

Una vez que el dato se halla en poder del bróker, éste lo publica a todos los suscriptores que estén suscritos al “topic” bajo el cual llegó el dato al bróker. El proyecto contará con dos suscriptores, el primero de ellos será un sistema de gestión encargado de almacenar el dato albergado en el mensaje del “topic”, evaluarlo y publicar otro mensaje bajo otro cierto “topic” con el resultado de la evaluación para que el sistema de actuación opere según esta evaluación. El segundo suscriptor, por tanto, consistirá en el sistema de actuación que estará suscrito al “topic” con el que haya publicado el sistema de gestión, recibirá la información y actuará en consecuencia.

Los sistemas de gestión y actuación se verán en el Capítulo 7, el primer sistema se detallará en profundidad, mientras que el segundo se comentará de manera más simplificada debido a la imposibilidad de configuración de los equipos de actuación, que quedarán como trabajos futuros para el proyecto.

Un aspecto que se ha tenido en cuenta a la hora de seleccionar los equipos es que han de ser capaces de conectarse a la red, esto es debido a que el bróker con el que deben interactuar es inalámbrico lo que obliga tanto al sistema de adquisición de datos como a los actuadores a

contar con algún tipo de módulo WiFi para tal fin [42]. En este caso como se ha visto en el subapartado 5.1 el módulo WiFi del sistema de adquisición estaba integrado en el microcontrolador, mientras que como se verá el subapartado 7.2.2 el variador de frecuencia seleccionado también contará con uno. Por supuesto, también la base de datos que lleve el registro de todos los datos estará instalada en un ordenador con acceso a internet.

En la actualidad existen multitud brókers de código abierto que guardan entre sí, aunque programados en diferentes lenguajes como son C, Java o Javascript, para aplicaciones diversas en función de las necesidades de cada proyecto. En el caso del presente proyecto, el bróker que se ha seleccionado es el denominado Mosquitto desarrollado por Eclipse y distribuido bajo licencia EPL/EDL. Este bróker es de código abierto, está programado en C, es multiplataforma, liviano, apto para servidores de poca potencia y está diseñado especialmente para aplicaciones del protocolo de comunicación MQTT [72].



Figura 6.8.- Bróker utilizado en el proyecto.

### 6.3. Arduino

Arduino es una plataforma de desarrollo y creación electrónica de código abierto basada en un hardware y un software libre. El hardware consistirá en una placa electrónica que contará con un microcontrolador reprogramable y una serie de pines hembra, permitiendo la interconexión entre estos últimos y los sensores con los que deberá comunicarse. El software será de libre creación por parte de los desarrolladores y será implementado a través del entorno de desarrollo de Arduino, denominado Arduino IDE, mediante lenguaje de programación Wiring [73]. La Figura 6.9.- Arduino IDE. muestra el aspecto que tiene este entorno de desarrollo al abrirse un nuevo proyecto.

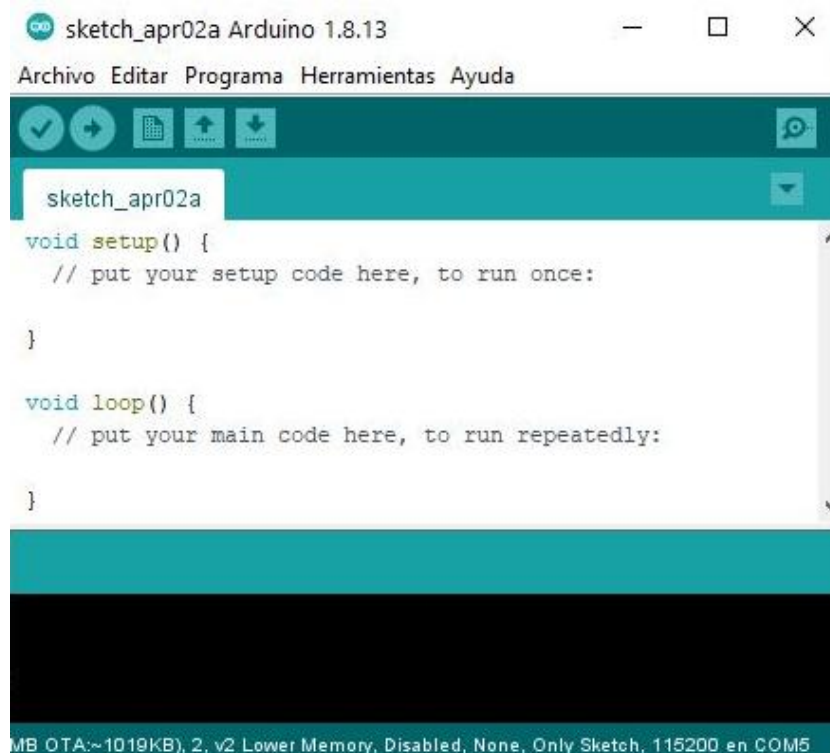


Figura 6.9.- Arduino IDE.

El proyecto Arduino nació en el Instituto de Diseño de Ivrea, Italia, en el año 2005 con el fin de facilitar el aprendizaje y uso de la electrónica a los estudiantes del instituto, principalmente rebajando los costes de las placas electrónicas, ya que hasta aquella época las existentes les resultaban excesivamente caras. Posteriormente, al cerrar el instituto se vieron en la tesitura de poder perder todo el proyecto Arduino, por lo que decidieron liberarlo y abrirlo al mundo para que todos, sean cuales sean sus conocimientos técnicos, pudieran participar del proyecto [73].

La ventaja principal de Arduino, y por la cual se utilizará para el proyecto, es que es libre, lo que reduce el coste ostensiblemente si lo comparamos con otros softwares de pago. Además, existen multitud de placas de desarrollo oficiales y no oficiales que permiten trabajar con el entorno de Arduino, como el NodeMCU escogido para el proyecto que es una de las no oficiales que permite su implementación con este entorno. También cuenta con una gran comunidad de usuarios lo que conlleva que haya un gran número de librerías y programas ya existentes para prácticamente cualquier aplicación, lo que facilita enormemente la tarea al desarrollador. Finalmente, decir de este programa que es multiplataforma, puede instalarse en los sistemas operativos de Windows, Mac OS y Linux, y que el lenguaje de programación Wiring que implementa es de sencilla comprensión, ya que está basado en C++ [73].

El hardware utilizado para el proyecto es la placa de desarrollo NodeMCU, ampliamente comentada en el subapartado 5.2, la cual incorpora el microcontrolador ESP8266 para realizar las instrucciones que se le implementarán mediante código. No obstante, no basta sólo con conectar el NodeMCU al ordenador para subirle las instrucciones que se vayan a escribir en el Arduino IDE, sino que este último ha de configurarse para poder realizar esta exportación de instrucciones y que el microcontrolador del NodeMCU las pueda interpretar. Completada esta configuración la aplicación quedará lista a la espera de la introducción de las instrucciones.

### 6.4. Aplicación con Arduino

En este subapartado se desarrollará el código de programación para la monitorización de los sensores en lenguaje Wiring. El programa resultante será subido al microcontrolador ESP8266 del NodeMCU, que lo ejecutará periódicamente.

La aplicación debe de ser capaz de leer el dato ambiental obtenido por el sensor, crear una función que lo almacene en una variable, crear la conexión I<sup>2</sup>C a través de los pines del SDA y del SCL, crear la conexión WiFi para conectarse con el bróker y enviarle la cabecera, los “topics”, para, posteriormente, incluir bajo cada “topic” la variable con el dato recogido por el sensor correspondiente, lo que formará el contenido del mensaje que se enviará al bróker. Finalmente, la monitorización se consigue mediante un bucle continuo que realice, si bien no las conexiones, ya que éstas deberán producirse una única vez (debido a que el sistema queda conectado hasta que se dé por terminada la conexión), si el resto de las acciones con el fin de recoger periódicamente los datos ambientales y enviárselos, cuando sea requerido, al bróker bajo el “topic” proporcionado.

La estructura que va a seguir el código de la aplicación será la denominada modular o por zonas, cada una de ellas con una finalidad distinta e interconectadas entre sí a través de las variables y funciones que se van a crear a lo largo del programa. Las zonas con las que contará la aplicación del proyecto serán:

- Librerías: serán el conjunto de funciones predefinidas e implementadas en el entorno de desarrollo que se usarán en la aplicación del proyecto. En esta zona se incluirá también el certificado de seguridad TLS para los mensajes.
- Variables globales y prototipos de las funciones: en esta zona se incluirán las variables y funciones que se utilizarán en la aplicación. No obstante, por simplificar el programa principal las funciones se han creado en otros archivos

adjuntos y serán llamadas por el programa principal cuando se las requiera. Por otro lado, las variables van a recoger la siguiente información:

- Los datos ambientales recogidos por los sensores.
  - La nomenclatura y la contraseña de la red WiFi a la que va a conectarse el microcontrolador.
  - La dirección IP que tendrá el bróker.
  - La cadena de caracteres primaria para la identificación del cliente, que en este caso será el microcontrolador.
  - La cabecera, obligatoria, del mensaje que se enviará al bróker.
  - Se guardará un espacio en memoria para almacenar los futuros mensajes que se enviarán al bróker.
  - El “topic” según el cual se publicará el mensaje en el bróker.
- **Setup:** es la zona de configuración de los equipos. En este caso se establecerá la comunicación I<sup>2</sup>C con el sensor a través de los pines SDA y SCL del bus I<sup>2</sup>C en la placa de desarrollo. A su vez, en esta zona, se selecciona la velocidad de subida del programa, para el monitor serie y se crea una breve rutina de calibración del sensor en función de la altura a la que se halle. En esta zona tiene lugar, también el establecimiento de la conexión WiFi. Y, finalmente, se establece la comunicación con el bróker por parte del cliente, el microcontrolador.
  - **Programa principal:** el programa principal recogerá periódicamente los datos ambientales almacenados en las variables, los pasará a una cadena de caracteres, le añadirá a esa cadena la cabecera correspondiente y, el mensaje generado, lo publicará bajo su “topic” correspondiente en el bróker.

### 6.4.1. Librerías

La primera zona corresponde a las librerías que incorpora la aplicación. Las cuales como se ha mencionado anteriormente son un conjunto de funciones predefinidas destinadas a una funcionalidad específica. Estas librerías se incluyen al principio de la aplicación, en la primera zona, y podrán ser empleadas en cualquier momento a lo largo de la misma.

La aplicación que aquí se desarrolla va a contar con seis librerías, cada una de ellas con un cometido concreto que incluye desde el control del módulo de sensores hasta la comunicación entre los elementos del sistema de adquisición, y, de este sistema, con el bróker. Además de las seis librerías en esta zona se incluye el certificado de seguridad TLS que permite la encriptación de los mensajes enviados al bróker. En la Figura 6.10 se muestra como han de codificarse estas librerías y el certificado. A continuación, se especifica el uso de cada una de ellas:

- **string:** librería que permite trabajar con cadenas de caracteres.
- **stdlib:** librería empleada para transformar las variables numéricas a cadena de caracteres.
- **Wire:** librería para la comunicación I<sup>2</sup>C.
- **SparkFun\_SCD30\_Arduino\_Library:** librería para el control del módulo de sensores SCD30.
- **ESP8266WiFi:** librería para poder operar con el módulo WiFi del microcontrolador.
- **PubSubClient:** librería de publicación y suscripción del MQTT, para poder trabajar con este protocolo de comunicación.



## 6. COMUNICACIÓN ENTRE SISTEMAS

```
//Zona 1: Librerías

#include <string.h>
#include <stdlib.h>
#include <Wire.h>
#include <SparkFun_SCD30_Arduino_Library.h>
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include "Certificado_TLS.h"
```

Figura 6.10.- Librerías y certificado de la aplicación en Arduino.

La creación del certificado no será objeto de estudio del proyecto, no obstante, su utilización en la aplicación resulta fundamental para la encriptación del mensaje. Además, el archivo que contiene el certificado se incluye la creación de la primera variable, como es el cliente que se comunicará con el bróker a través de los mensajes encriptados mediante TLS. Por ello, en la Figura 6.11 recoge la codificación de este certificado y, en la Figura 6.12, se carga el certificado y se crea el cliente MQTT que cumpla con él.

```
const char caCert[] PROGMEM = R"EOF(
-----BEGIN CERTIFICATE-----
MIID9TCCAt2gAwIBAgIUJDMHEXyf6WUuWYv9fTHrztntNe2MwDQYJKoZIhvcNAQEL
BQAwYkxkCzAJBgNVBAYTAkVTMREwDwYDVQQIDAhBc3RlcmlhczEOMAwGA1UEBwwF
R2lqb24xZDZANBgNVBAoMB1VOSU9WSTEOMAwGA1UECwwFQ0UzSTIxFDASBgNVBAMM
CONFM0kyUm9vdENBMSAwHgYJKoZIhvcNAQkBFhFlbG9wZXpjQHVuaW92aS51czAe
Fw0yMTA1MTQwNzA4NTVaFw0zMTA1MTIwNzA4NTVaMIGJMQswCQYDVQQGEwJFUzER
MA8GA1UECAwIQXN0dXJpYXN0dXJpYXN0dXJpYXN0dXJpYXN0dXJpYXN0dXJpYXN0
MA8GA1UECwwFQ0UzSTIxFDASBgNVBAMMCONFM0kyUm9vdENBMSAwHgYJKoZIhvcNAQk
BhFlbG9wZXpjQHVuaW92aS51czAeFw0yMTA1MTQwNzA4NTVaFw0zMTA1MTIwNzA4
NTVaMIGJMQswCQYDVQQGEwJFUzERMA8GA1UECAwIQXN0dXJpYXN0dXJpYXN0dXJp
YXN0dXJpYXN0dXJpYXN0dXJpYXN0dXJpYXN0dXJpYXN0dXJpYXN0dXJpYXN0dXJp
VkkxZDZANBgNVBAoMB1VOSU9WSTEOMAwGA1UECwwFQ0UzSTIxFDASBgNVBAMM
CONFM0kyUm9vdENBMSAwHgYJKoZIhvcNAQkBFhFlbG9wZXpjQHVuaW92aS51czAe
SIB3DQEJARYRZm9vdENBMSAwHgYJKoZIhvcNAQkBFhFlbG9wZXpjQHVuaW92aS51
DwAwggEKAoIBAQCZ4mECT81JtTmJG/f0da92U1jXBhSelwnGnZO+mmP8ajpbTd+Q
elQkEyRa10IsEPfkjhzliMeLDWx43lncFJHV5zk+DIzcn9YHPgMVakcEqQQdTodC
F3zLtyAlHSv6IHB62i5vC7KCOGIh/GDo+w37xAhxXzg71sbR/bOS1TCyW03umSDD
NbQwzo3sy8M9KgAI0Pr3nKd9rPLAILWaUBbQs0Dim2zsdpeFrlAgwkG5TaAlIggY
7fjEli+jp5WY986BkXCGCoG+gLFempbIhhr/BaDRKToe6QBffLgRkKqNHHZLNeDS
kjSS2HA5I+KW+tmWey79Y8/BhSOaZI5o15NBAGMBAAGjUzBRMB0GA1UdDgQWBBSO
/sDMYvALUh2eJn2Wv6+zUns7ETAfBgNVHSMEGDAWgBSO/sDMYvALUh2eJn2Wv6+z
Uns7ETAfBgNVHRMBAf8EBTADAQH/MA0GCSqGSIb3DQEBCwUAA4IBAQA7cSa3ZZ+1
PK7+kdwRwIOxfj/oEId+sBW23jtnLJlpaKjtqVuRdcSOFJSqb7W7IT+PkI98WmCI
aUaI85z5miWbVHVyQu3JgCy3sEpuW3xeGzJHE5pcqyEkrmjsVEoBcqK5QggyOd6q
avF37IV4SPYsLw6IHiFh8rVPTR06RZmH5Cmww3aew8qgmwal/bIUTpev6xDawKZS
bFT93cu/+09XWBjkBN/nuXgYYV08DJUrAcocw+f+8x12dg9MYR1ZLFAGuetipthH
QvYRmyT0doWuOWFTmWis30J2Pp8ME3/KMNNKC2dWAOwCwWu12zJJPEpusTjtpFxE
4Wc8V3P14+gz
-----END CERTIFICATE-----
)EOF";
```

Figura 6.11.- Certificado de encriptación TLS.

```
const uint8_t mqttCertFingerprint[] =
{0x99, 0x14, 0x5D, 0x64, 0x77, 0xF1, 0x4C, 0x84, 0xB4, 0x4B, 0xAE, 0x22, 0x80, 0xE7,
 0xDD, 0x86, 0xAF, 0x74, 0x9C, 0xB1};
X509List caCertX509(caCert);
WiFiClientSecure MQTT_cliente;
PubSubClient mqttClient(MQTT_cliente);
```

Figura 6.12.- Creación del cliente MQTT seguro que cumple con el certificado.

### 6.4.2. Variables globales y prototipos de las funciones

La segunda zona corresponde a variables globales y prototipos de las funciones. Como se ha comentado al principio del subapartado los prototipos de las funciones se encontrarán separados del resto del programa para facilitar la accesibilidad a éste, por lo que se explicarán, individualmente, una vez se hayan expuesto las variables.

#### I. Variables globales

La zona de variables se subdivide en dos, por un lado aparecerán las variables necesarias para almacenar los datos que vayan a recoger los sensores, así como la creación de un objeto que gobierne el módulo de sensores, y por otro lado se encuentran las variables que van a almacenar información relativa a la comunicación entre el sistema de adquisición y el bróker, tales como usuario y contraseña del WiFi al que se conecte el sistema, la dirección IP del bróker al que enviar la información, o las cadenas de caracteres que almacenarán los mensajes que se envíen al bróker.

En primer lugar, se comenzará comentando el objeto que gobernará el módulo de sensores. En este caso gracias a la librería “SparkFun\_SCD30\_Arduino\_Library” se crea un objeto SCD30 de nombre “airSensor”, el cual tiene como principal función la de permitir obtener los datos de las mediciones cada dos segundos, no obstante, además de su uso principal se empleará para verificar el estado del pin RDY y para la calibración del sensor, como se irá viendo a lo largo de la documentación de esta aplicación.

A continuación, se crean las variables que almacenarán los datos que se vayan a medir y se inicializan. La designación de este nombre al objeto SCD30 y la creación de las variables aparece reflejado en la Figura 6.13.

```
//Zona 2: Variables globales y prototipos de las funciones

//Módulo de sensores
SCD30 airSensor;
uint16_t NivelCO2=0;
float Temp=0;
float HumRel=0;
```

*Figura 6.13.- Objeto y variables para la medición de los parámetros ambientales.*

La segunda parte de esta zona está formada por el conjunto de variables necesarias para llevar a cabo la comunicación MQTT.

Las primeras variables que se crearán serán el nombre y la contraseña de la red WiFi a la cual se va a conectar el microcontrolador. Seguidamente, la dirección IP del bróker a la cual hay que enviar los datos desde el microcontrolador.

A continuación, se crea el inicio del nombre del cliente que se conectará al WiFi y publicará la información en el bróker con los datos ambientales recogidos por los sensores. Sin embargo, esta información ha de publicarse en el bróker bajo una cierta cabecera. Por lo tanto, la siguiente variable que se crea es la cabecera del futuro mensaje que se publicará en el bróker MQTT. Tras la cabecera del mensaje se creará el “string” que contendrá el propio mensaje que se vaya a publicar en el bróker. Este “string” se inicializa con cero caracteres, pero se les asigna un tamaño máximo de 50 caracteres en memoria. Este mensaje albergará los datos ambientales recogidos, vistos en la Figura 6.13, a los que se le añadirán sus unidades y la cabecera del mensaje. Este proceso de creación del mensaje se realizará mediante concatenación de cadenas como se verá más adelante.

En este momento entra en juego otro aspecto, y es que, aunque el mensaje sea enviado bajo una cierta cabecera, el bróker lo publicará para sus suscriptores con otro título, lo que se ha ido llamando a lo largo del proyecto “topic”. Por ello, para finalizar esta zona de variables del

protocolo MQTT se creará el “topic” bajo el cual se van a publicar los mensajes en el bróker. Todo este proceso de creación de las variables del protocolo MQTT aparece representado en las líneas de código de la Figura 6.14.

```
//MQTT
//Nomenclatura de la red WiFi y contraseña
const char* ssid="WiFi_TFM";
const char* password="*****";

//Dirección IP del bróker al cual se conecta
const char* IP_mqtt_server="156.35.154.1";

//Se crea el inicio del cliente WiFi para conectarse al WiFi
String clientID="Marcos_";

//Cabeceras de los mensajes que se enviarán al bróker MQTT
const char* cabecera_MQTT="Datos ambientales: ";

//Se crean los strings dónde se escribirá el mensaje para el bróker
char MQTT_Datos[50]={0};

//"Topic" para el bróker MQTT
const char* topic_datos="TFM/datos";
```

Figura 6.14.- Creación de variables para el protocolo MQTT.

### II. Prototipos de las funciones

Una vez vistas las variables globales que se han de crear para llevar a cabo la toma de datos y la comunicación MQTT, se expondrán cada una de las funciones que hacen posible que esto pueda llevarse a cabo.

#### i. Función para la conexión WiFi

Esta función va a permitir al microcontrolador conectarse a la red WiFi dónde se encuentre. Para ello, en primer lugar, se envían a esta función las variables globales con el usuario y la contraseña WiFi a la que ha de conectarse. Seguidamente se crea una pequeña subrutina para comprobar si se ha establecido la conexión WiFi. Las líneas de código que implementan esta función aparecen reflejadas en la Figura 6.15.

```
void conectar_WiFi()
{
    WiFi.begin(ssid,password);
    Serial.print("Conectando");
    while (WiFi.status() != WL_CONNECTED)
    {
        delay(500);
    }
    //Se informa cuando la conexión haya sido exitosa
    Serial.println();
    Serial.print("Conectado a direccion IP: ");
    Serial.println(WiFi.localIP());
}
```

Figura 6.15.- Función para la conexión WiFi.

### ii. Función para el protocolo de seguridad de la conexión con el bróker

Como se ha comentado al inicio de la aplicación la conexión con el bróker ha de realizarse de manera segura, para ello se emplea la encriptación del mensaje mediante el certificado de seguridad TLS. Esta función, por tanto, creará el cliente seguro del protocolo MQTT, e indicará el puerto al que ha de conectarse con el bróker para realizar esta conexión segura, el 8883. Las líneas de código que recogen esta función aparecen representadas en la Figura 6.16.

```
void conectar_broker_MQTT()
{
    MQTT_cliente.setTrustAnchors(scaCertX509);
    MQTT_cliente.allowSelfSignedCerts();
    MQTT_cliente.setFingerprint(mqttCertFingerprint);
    verifytls();
    //Ajustes del broker MQTT
    mqttClient.setServer(IP_mqtt_server, 8883);
    clientID += String(random(0xffff), HEX);
    Serial.print("MQTT-ClientID: ");
    Serial.println(clientID);
}
```

Figura 6.16.- Función para la conexión segura con el bróker.

### iii. Función que verifique la conexión mediante TLS

Como se ha podido ver en las líneas de código de la Figura 6.16 éstas incluían una función denominada “verifytls” encargada de verificar que se haya implementado correctamente el protocolo TLS de seguridad. Para ello, comprueba que el cliente se ha conectado a la dirección IP del bróker a través del puerto 8883. En la Figura 6.17 se muestra, en líneas de código, la rutina que permiten realizar esta verificación.

```
bool verifytls()
{
    bool success = false;
    Serial.print("Verificando las conexiones TLS con: ");
    Serial.println(IP_mqtt_server);
    success = MQTT_cliente.connect(IP_mqtt_server, 8883);
    if (success)
        Serial.println("Conexión realizada: Certificado y huella validas.");
    else
        Serial.println("La conexión ha fallado");
    return (success);
}
```

Figura 6.17.- Función de verificación de la conexión con TLS.

### iv. Función de conexión del cliente con el bróker

Esta función comunicará el cliente con el bróker. Se crea una rutina hasta que la conexión tenga éxito, y en caso de que no haya sido exitosa la conexión se mostraría el fallo que ha tenido lugar. Las líneas de código que implementan esta conexión aparecen en la Figura 6.18.

```
void connect()
{
  //Bucle hasta que tenga éxito la conexión
  while (!mqttClient.connected())
  {
    if (mqttClient.connect(clientID.c_str()))
    {
      Serial.println("Conectado");
    }
    else
    {
      Serial.print("Fallo, rc=");
      Serial.print(mqttClient.state());
      delay(5000);
    }
  }
}
```

Figura 6.18.- Función para la conexión del cliente con el bróker.

v. Función de medición del nivel de CO<sub>2</sub>

Esta función se encarga de recoger el dato con el nivel de CO<sub>2</sub> del sensor y almacenarlo en su variable correspondiente. Sólo se recogerá el dato cuando el pin RDY del módulo de sensores esté activo, por ello a la función se le envía el objeto SCD30 creado en la zona de variables, así como la variable global para almacenarlo. En la Figura 6.19 se muestran las líneas de código necesarias para implementar esta función.

```
uint16_t LeerNivelCO2(SCD30 airSensor){
  if (airSensor.dataAvailable())
  {
    NivelCO2=airSensor.getCO2();
  }
  return NivelCO2;
}
```

Figura 6.19.- Función para la medición y el almacenamiento de los niveles de CO<sub>2</sub> en la estancia.

vi. Función de medición de la temperatura

La segunda de las funciones que se ha implementado en la aplicación sigue la misma línea que la de su predecesora simplemente que ahora cambiando el dato ambiental y la variable global que lo almacena, en este caso todo estará relacionado con la temperatura. En la Figura 6.20 se observan estas conocidas líneas de código con los cambios introducidos.

```
float LeerTemp(SCD30 airSensor){
  if (airSensor.dataAvailable())
  {
    Temp=airSensor.getTemperature();
  }
  return Temp;
}
```

Figura 6.20.- Función para la medición y el almacenamiento de la temperatura en la estancia.

vii. Función para la medición de la humedad relativa

La última de las funciones para la medición de datos ambientales corresponde al valor de la humedad relativa. Se emplean, nuevamente las mismas líneas que para las dos funciones anteriores sólo que utilizando el objeto y las variables globales oportunas para esta medición. En la Figura 6.21 se muestran estas líneas de código.

```
float LeerHum(SCD30 airSensor){  
  
    if (airSensor.dataAvailable())  
    {  
        HumRel=airSensor.getHumidity();  
    }  
    return HumRel;  
}
```

Figura 6.21.- Función para la medición y el almacenamiento de la humedad relativa en la estancia.

viii. Función generar mensaje de datos ambientales

Esta función utiliza los datos de medición almacenados en las variables de concentración de CO<sub>2</sub>, temperatura y humedad relativa, para crear el mensaje que se publicará, posteriormente, en el bróker. En primer lugar, se crean las variables locales tipo “string” que almacenarán los datos ambientales en ese formato, ya que el mensaje ha de ser publicado como una cadena de caracteres. Por ello, posteriormente, se transformarán los datos de la medición en una cadena de caracteres. Una vez transformado cada valor de la medición a “string”, éstos ya podrán ser añadidos al mensaje mediante concatenación de “strings”.

Como se ha visto en la zona de variables globales, se ha creado una variable de tipo “string” que recogerá el mensaje a publicar en el bróker. Es importante el vaciado o la inicialización de este “string” para evitar posibles errores de solapamiento con el nuevo mensaje que se desee publicar.

Finalmente, el mensaje es creado mediante concatenación, se comenzará añadiendo la cabecera del mensaje y seguidamente los “strings” creados con los datos de medición y sus unidades correspondientes, “ppm” para el nivel de concentración de CO<sub>2</sub>, “°C” para la temperatura y “%” para la humedad relativa. Todo este proceso de creación del mensaje aparece recogido en líneas de código en la Figura 6.22.

```
void GenerarMensaje(int NivelCO2, float Temp, float HumRel){  
    char string_NivelCO2[5];  
    char string_Temp[5];  
    char string_HumRel[5];  
    sprintf(string_NivelCO2,"%05i",NivelCO2);  
    MQTT_Datos[0]='\0';  
    strcpy(MQTT_Datos, cabecera_MQTT);  
    strcat(MQTT_Datos, string_NivelCO2);  
    strcat(MQTT_Datos, " ppm, ");  
    dtostrf(Temp, 5, 2, string_Temp);  
    strcat(MQTT_Datos, string_Temp);  
    strcat(MQTT_Datos, " °C y ");  
    dtostrf(HumRel, 5, 1, string_HumRel);  
    strcat(MQTT_Datos, string_HumRel);  
    strcat(MQTT_Datos, " %");  
}
```

Figura 6.22.- Función para la creación del mensaje que se publicará en el bróker.

### ix. Función cliente del protocolo MQTT

Esta función está encargada de publicar el mensaje, recién creado, en el bróker bajo el “topic” declarado en la zona de variables globales. Las líneas de código que implementan esta función aparecen reflejadas en la Figura 6.23.

```
void clienteMQTT(char* MQTT_Datos)
{
    mqttClient.publish(topic_datos, MQTT_Datos);
}
```

Figura 6.23.- Función para la publicación del mensaje en el bróker bajo el correspondiente “topic”.

### 6.4.3. Setup

En esta tercera zona se configurará tanto el módulo de sensores, como la conexión I<sup>2</sup>C entre dicho módulo y el microcontrolador. A su vez, se realizan las dos primeras llamadas a funciones, en este caso serán a la función de conexión del microcontrolador con la red WiFi, función “i” del subapartado anterior, y a la función que establece el protocolo de seguridad con el bróker, función “ii” del subapartado anterior.

En primer lugar, para la configuración de estos equipos se ha optado por fijar la velocidad de subida de la información al microcontrolador en 115200 baudios. Posteriormente se inicializa la conexión I<sup>2</sup>C entre el módulo y el microcontrolador a través de sus pines correspondientes.

A continuación, se hace uso de una funcionalidad incluida en la librería “SparkFun\_SCD30\_Arduino\_Library” del módulo de sensores, que no es otra que la autocalibración de dicho módulo con respecto a la altura sobre el nivel del mar en la que se vaya a ubicar. El establecimiento de esta velocidad de subida, el inicio de la conexión I<sup>2</sup>C y la autocalibración del módulo aparecen recogidos en líneas de código en la Figura 6.24.

```
//Zona 3: Setup

void setup() {

    Serial.begin(115200); //Se fija la velocidad de subida
    Serial.println("TFM utilizando el módulo SCD30"); //Título de las mediciones
    Wire.begin(D6,D7); //Se arranca la conexión a través de los puertos para I2C

    if (airSensor.begin(Wire, true) == false) //Se comprueba que se ha establecido la conexión I2C
    {
        Serial.println("Sensor no detectado. Por favor compruébalo");
        while (1)
            ;
    }

    //El módulo permite autocalibrarse en función de la altitud a la que deba realizar las mediciones
    Serial.print("Autocalibracion ha resultado ser ");
    if (airSensor.getAutoSelfCalibration() == true)
        Serial.println("correcta");
    else
        Serial.println("incorrecta");
}
```

Figura 6.24.- Configuración de la velocidad de subida, la conexión I<sup>2</sup>C y la autocalibración del módulo.

Por otro lado, en esta misma zona, se conecta el microcontrolador a la red WiFi, donde se encuentre situado, mediante la llamada a la función de conexión WiFi, ya vista en el subapartado 6.4.2, en la cual se incluyen la identificación de la red y su contraseña, necesarios para establecer la conexión.

Para finalizar esta zona de configuración, se llamará a la función que implementa el protocolo de conexión segura con el bróker. La creación de un cliente MQTT seguro, como se

ha visto también en el subapartado 6.4.2, es necesario realizarla, además de con el protocolo de seguridad, con la dirección IP del bróker y a través del puerto 8883 del bróker. Estas dos llamadas a funciones, tanto la conexión WiFi como la creación del cliente MQTT seguro aparecen reflejas en líneas de código en la Figura 6.25.

```
//Configuración de la conexión WiFi y creación del cliente seguro MQTT
conectar_WiFi();
conectar_broker_MQTT();
```

*Figura 6.25.- Conexión del microcontrolador a la red WiFi y creación del cliente MQTT seguro.*

### 6.4.4. Programa principal

La cuarta y última zona corresponde al programa principal, las líneas de código que se escriban esta parte del programa se repetirán cíclicamente, es decir, una vez que se completen todos los comandos se vuelve al primero.

El programa principal comenzará con dos comprobaciones, la primera de ellas corresponde al estado en el que se encuentra la conexión del cliente con el bróker y, la segunda permite conocer si se encuentra activo el pin RDY del módulo de sensores.

La subrutina que se ha creado para conocer el estado de la conexión del cliente con el bróker realiza un llamamiento a la función de conexión con el bróker, vista con el indicativo “iv” en el listado de funciones creadas en el subapartado 6.4.2, en caso de que no se haya establecido aún la conexión. Esta comprobación aparece en líneas de código en la Figura 6.26.

```
//Zona 4: Programa principal

void loop() {

    //Se comprueba la conexión con el bróker para enviar los datos
    if (!mqttClient.connected()){connect();}
```

*Figura 6.26.- Comprobación del estado de la conexión cliente-bróker.*

La segunda comprobación se realiza mediante una subrutina encargada de esperar a que el pin RDY se encuentre activo, lo que indica que los sensores del módulo ya han terminado de realizar las mediciones y se puede proceder a su lectura.

A continuación, una vez realizadas las comprobaciones anteriores, se procede a la recogida de los datos de mediciones realizada por los sensores y a la generación del mensaje, que se publicará en el bróker.

La lectura de estos datos se realiza mediante las funciones correspondientes creadas, para tal fin, en el subapartado 6.4.2. Estas funciones eran la “v”, “vi” y “vii” del listado de funciones ese subapartado, siendo, respectivamente, de lectura de los niveles de CO<sub>2</sub>, de temperatura y de humedad relativa. En ellas se hacía uso del objeto SCD30, que se pasa a la función como parámetro, creado para leer esos datos y posteriormente se almacenaban en una variable.

La generación del mensaje que se va a enviar al bróker se realiza, nuevamente, con el llamado a una función, en este caso la vista en el apartado “viii” del listado de funciones del subapartado 6.4.2. Esta función se encargaba de transformar los datos a cadenas de caracteres, es por ello, que se le pasa las variables de los datos como parámetros de la función, añadirle la cabecera y agrupar el conjunto recién creado bajo un “string” que conformaría dicho mensaje. A su vez, esta función borraba el mensaje anterior para evitar posibles errores. En la Figura 6.27 se muestran las líneas de código que realizan el llamamiento a estas funciones, tanto las de lectura de datos como las de generación del mensaje, y la subrutina de espera en caso de que no se hayan realizado todavía.



## 6. COMUNICACIÓN ENTRE SISTEMAS

---

```
if (airSensor.dataAvailable()==HIGH){
  //Se recogen los datos de la medición y se crea el mensaje
  NivelCO2=LeerNivelCO2(airSensor);
  Temp=LeerTemp(airSensor);
  HumRel=LeerHum(airSensor);
  GenerarMensaje(NivelCO2, Temp, HumRel);
}
else
  delay(2000);
```

Figura 6.27.- Recogida de los datos de medición en variables, generación del mensaje y subrutina de espera.

Los últimos comandos que se van a implementar en el programa principal son los correspondientes a la publicación del mensaje en el bróker por parte del cliente bajo el correspondiente “topic”. Una vez publicado, se esperan dos segundos a que los sensores vuelvan a realizar las mediciones y se avisa al bróker que el cliente sigue activo, aunque no publique nada en ese lapso de tiempo.

La publicación de los mensajes en el bróker por parte del cliente se realiza mediante el llamamiento a la última de las funciones vista en el listado del subapartado 6.4.2, la “ix”. En ella el cliente publica en el bróker el mensaje bajo el correspondiente “topic”. Para ello, ha de pasarse por parámetro a la función el mensaje creado. Las líneas de código que incluyen el llamamiento a esta función, así como la espera de un segundo y la comunicación del cliente al bróker indicándole que permanece activo, están representadas en la Figura 6.28.

```
clienteMQTT(MQTT_Datos);

delay(1000);

mqttClient.loop();
```

Figura 6.28.- Publicación del mensaje en el bróker.

La aplicación completa aparece en el Anexo A, donde se podrán ver todas las líneas de código que la componen aquí comentadas, junto con las líneas que incluyen la verificación de este código.

## 7. Sistemas de gestión y actuación

En este capítulo se describirán los dos sistemas restantes, el de gestión y el de actuación. El primero de ellos generará la orden que ha de cumplir el segundo, consiguiéndose así la regulación del motor del ventilador.

La comunicación entre ellos será, como se ha visto a durante todo el proyecto, mediante el protocolo MQTT. En él el sistema de gestión deberá publicar la orden bajo un determinado “topic” y el sistema de actuación debe ser capaz de recibir esa orden suscribiéndose a dicho “topic”.

El primero de los sistemas que se va a analizar será el de gestión, este sistema estará suscrito al “topic” con los datos de la medición publicados por el sistema de adquisición. Una vez recibido el mensaje tendrá que interpretarlo y publicar la instrucción en otro “topic” para que el sistema de actuación pueda recibirla.

### 7.1. Sistema de gestión

El sistema de gestión tendrá dos funciones principales, la primera de ellas de almacenamiento de datos y la segunda de interpretación de éstos para poder emitir la orden.

Este sistema estará realizado íntegramente en software, se creará una aplicación mediante el programa Python que contendrá las líneas de código necesarias para llevar a cabo las dos funciones.

En primer lugar, el programa ha de poder suscribirse a los “topics” bajo los que ha publicado las mediciones el sistema de adquisición de datos.

Una vez logrado este primer paso, la aplicación guardará estos datos en una base de datos, datándolos, consiguiendo así llevar un registro de los valores diarios obtenidos y las posibles incidencias que pudiera haber surgido durante el funcionamiento.

El siguiente paso, en paralelo con el registro de los datos, será la interpretación de éstos. Para ello, el sistema de gestión debe contemplar todos los escenarios posibles y dirimir una orden que se ha de publicar para que llegue al sistema de actuación. En el caso de este proyecto únicamente se han considerado un número determinado de escenarios, debido a que, como se comentará en el siguiente subapartado 7.2, no se dispone del variador de frecuencia, por lo que se desconoce el formato de publicación de las órdenes para que éste pueda interpretarlo, quedando la comprobación de todos los escenarios posibles como trabajos futuros.

A continuación, se irán detallando cada uno de los componentes de la aplicación creada en Python. Estos componentes serán las librerías, las variables, las funciones, el programa principal (únicamente empleado para llamar a la función que debe recibir el mensaje) y, por último, la interfaz gráfica de esta aplicación, cuyas líneas de código no se comentarán, pero se incluirán en el Anexo B de este proyecto.

#### 7.1.1. Librerías

En esta primera zona se escribirán todas las librerías que se utilizarán para la realización de la aplicación, así como las funciones de otros módulos que habrán de ser llamadas para que el programa funcione. Cabe destacar la librería “paho.mqtt.client” empleada para la comunicación con el protocolo MQTT y “ssl” para trabajar con el cliente seguro. El resto se utilizan para la interfaz de la aplicación, obtener la fecha y hora a cada instante, generar valores aleatorios y trabajar con “strings”. Así como la librería “tkinter” para la creación de la interfaz gráfica de la aplicación. La última corresponde a una función de otro módulo empleada para almacenar los datos en la base de datos. En la Figura 7.1 se muestra con líneas de código esta zona de librerías.

```
#ZONA: Librerías
from tkinter import font
from tkinter.font import BOLD
import paho.mqtt.client as mqtt
import time
import ssl
from tkinter import *
import random
import string
from BDD_SG import AnadirDatos
```

Figura 7.1.-Librerías de la aplicación del sistema de gestión.

### 7.1.2. Variables

A continuación, se mostrarán las variables correspondientes a la dirección IP del bróker, se creará el cliente MQTT seguro, la variable auxiliar para las órdenes y se implementarán los “topics”, tanto a los que el sistema de gestión deba suscribirse, para recibir los mensajes del sistema de adquisición de datos, como bajo el cual publicará la orden. El desarrollo de estas variables está escrito en líneas de código en la Figura 7.2.

```
#ZONA: Variables globales y prototipos de las funciones
#Dirección IP del bróker
IP_mqtt_server="156.35.154.1"

#Creación del cliente seguro
cliente_MQTT = 'Marcos_' + ''.join(random.choice(string.ascii_uppercase + string.digits)
                                   for x in range(8))
mqttc = mqtt.Client(cliente_MQTT)
mqttc.tls_set("cacert.pem", tls_version=ssl.PROTOCOL_TLSv1)
mqttc.tls_insecure_set(True)

#Variables para las órdenes
nivelAnt=0

#"Topic" al que se suscribe
topic_Datos="TFM/datos"

#"Topic" para publicar
topic_act="TFM/orden"
```

Figura 7.2.- IP del bróker, creación de cliente seguro, variable auxiliar y “topics” para suscribirse y publicar.

### 7.1.3. Funciones

En este subapartado se explicarán todas las funciones que alberga la aplicación, son las que van a realizar las labores de conexión y desconexión con el bróker, suscripción y desuscripción a los “topics”, recepción de los mensajes, almacenamiento en base de datos, interpretación de los datos y publicación de mensajes en el “topic” asignado.

#### I. Conexión y desconexión del bróker

La conexión con el bróker será necesaria para que el sistema de gestión pueda suscribirse al “topic” que alberga el mensaje con los datos de las mediciones, así como publicar los mensajes con las órdenes bajo el otro “topic” correspondiente.

Esta función conecta el cliente MQTT con la dirección IP del bróker a través del puerto 8883. Una vez lo ha realizado, el cliente se queda esperando a suscribirse a los “topics” o a

publicar en ellos. Las líneas de código que muestran esta función aparecen representadas en la Figura 7.3.

```
#Función para la conexión con el bróker
def ConectarBroker():
    try:
        IP_broker=IP_mqtt_server
        puerto=8883
        timeout_seg=60
        mqttc.connect(IP_broker, puerto,timeout_seg)
        time.sleep(1)
        mqttc.loop_start()
        time.sleep(0.1)
        mqttc.loop_start()
        cuadroMensaje.insert(0, "Conectado al bróker con IP: "+IP_broker)
        cuadroPublicacion.insert(0, "Conectado al bróker con IP: "+IP_broker)
        cuadroMensaje.insert(0, "-----")
        cuadroPublicacion.insert(0, "-----")
    except:
        cuadroMensaje.insert(0, "-----")
        cuadroPublicacion.insert(0, "-----")
        cuadroMensaje.insert(0, "No se ha podido conectar con el bróker")
        cuadroPublicacion.insert(0, "No se ha podido conectar con el bróker")
```

Figura 7.3.- Función para la conexión del cliente MQTT con el bróker.

La siguiente función de desconexión al bróker está ligada a la anterior. En ella la aplicación realiza dos acciones, por un lado, se desuscribe del “topic” y, por otro, desconecta al cliente MQTT del bróker. En la Figura 7.4 se pueden ver las líneas de código que se emplean para la realización de esta función.

```
#Función para la desconexión con el bróker
def DesconectarBroker():
    time.sleep(0.1)
    mqttc.unsubscribe(topic_Datos)
    mqttc.disconnect()
    mqttc.loop_stop()
    time.sleep(1)
    cuadroMensaje.insert(0, "-----")
    cuadroPublicacion.insert(0, "-----")
    cuadroMensaje.insert(0, "Desuscribiendo del topic")
    cuadroMensaje.insert(0, "Desconectando del bróker")
    cuadroPublicacion.insert(0, "Desuscribiendo del topic")
    cuadroPublicacion.insert(0, "Desconectando del bróker")
    cuadroMensaje.insert(0, "-----")
    cuadroPublicacion.insert(0, "-----")
```

Figura 7.4.- Función para la desconexión del cliente MQTT con el bróker.

## II. Suscripción y desuscripción a “topics”

Estas funciones son similares a las de conexión y desconexión con el bróker, sólo que estas se encargan de suscribirse o desuscribirse al “topic”.

La función de suscribirse genera la suscripción del cliente MQTT al “topic”. La implementación de esta función aparece en líneas de código en la Figura 7.5.

```
#Función para suscribirse al "topic"
def SuscribirTopic():
    time.sleep(1)
    mqttc.subscribe(topic_Datos)
    mqttc.loop_start()
    cuadroMensaje.insert(0, "Suscrito al topic")
    cuadroMensaje.insert(0, "-----")
```

*Figura 7.5.- Suscripción del cliente MQTT al "topic".*

De la misma forma, pero a la inversa, actúa la función para desuscribirse del "topic". Se desuscribe al cliente MQTT del "topic". Las líneas de código de esta función aparecen en la Figura 7.6.

```
#Función para desuscribirse al "topic"
def DesuscribirTopic():
    time.sleep(0.1)
    mqttc.unsubscribe(topic_Datos)
    mqttc.loop_start()
    cuadroMensaje.insert(0, "-----")
    cuadroMensaje.insert(0, "Desuscribiendo del topic")
    cuadroMensaje.insert(0, "-----")
```

*Figura 7.6.-Desuscripción del cliente MQTT del "topic".*

### III. Recepción de los mensajes con los datos ambientales

Esta función es la que se encarga de recoger los mensajes a los que el cliente MQTT está suscrito.

En primer lugar, comprueba que hay mensaje, no obstante, como se verá en el programa principal esta función sólo actúa cuando se recibe un mensaje, por lo que esta comprobación es a modo de seguridad del funcionamiento correcto de la aplicación. Cuando se recibe un mensaje, la aplicación recoge la fecha y la hora a la que ha sido recibido el mensaje y recoge del mensaje los datos correspondientes a los niveles de concentración de CO<sub>2</sub>, de temperatura y de humedad relativa.

De cada uno de ellos extrae el dato en forma de cadena de caracteres para después transformarlo a tipo numérico, ya sea entero o decimal, y almacenarlo en su variable correspondiente.

Una vez guardados los datos en una variable, la variable que alberga el nivel de concentración de CO<sub>2</sub> se compara con el valor auxiliar, que se irá renovando para cada mensaje con el valor anterior de concentración que haya habido. Se crea una breve subrutina para comparar este valor anterior con el nuevo, si son distintos se llama a la función que almacenará las variables en la base de datos, que se comentará a continuación en este mismo subapartado. Consecutivamente las variables que contengan los datos de concentración de CO<sub>2</sub>, temperatura y la auxiliar se pasarán a otra función que se encargará de interpretarlos, esta función se verá más adelante. Esa interpretación se almacenará en una variable de tipo "string" y se pasará como parámetro a la función publicar, que como se verá posteriormente se encargará de publicarla en el bróker. Y, finalmente se esperan dos segundos que es el tiempo que tarda el módulo de sensores en realizar cada nueva medición. Mientras que si son iguales no será necesario emitir ninguna orden sino que simplemente se añadirán los valores de la concentración, temperatura y humedad relativa a la base de datos. Las líneas de código que implementan esta función aparecen en la Figura 7.7.

```
#Función para la recepción de los datos
def RecibirDatos (mosq, userdata, msg):
    if msg.payload != 0:
        fecha_tupla=time.localtime()
        Fecha=time.strftime("%d/%m/%Y",fecha_tupla)
        Hora=time.strftime("%H:%M:%S",fecha_tupla)
        cuadroMensaje.insert(0, "Mensaje: "+str(msg.payload.decode('utf-8')))
        SnivelCO2=str(msg.payload)
        SdatoCO2=SnivelCO2[20:26]
        global nivelCO2
        nivelCO2=int(SdatoCO2)
        Stemp=str(msg.payload)
        SdatoTemp=Stemp[31:37]
        global Temp
        Temp=float(SdatoTemp)
        Shum=str(msg.payload)
        SdatoHum=Shum[50:56]
        global HumRel
        HumRel=float(SdatoHum)
        global nivelAnt
        if nivelAnt!=nivelCO2:
            AnadirDatos(Fecha, Hora, nivelCO2, Temp, HumRel)
            orden=ordenVentilador(nivelCO2, Temp, nivelAnt)
            publicar(orden)
            time.sleep(2)
            nivelAnt=nivelCO2
        else:
            AnadirDatos(Fecha, Hora, nivelCO2, Temp, HumRel)
            time.sleep(2)
            nivelAnt=nivelCO2
    else:
        print("No había mensaje")
```

Figura 7.7.- Recogida de datos y llamamiento a funciones de almacenamiento, interpretación y publicación.

#### IV. Creación de la base de datos y agregación de registros

En la función anterior se ha visto que se llamaba a una función que añadía registros a la base de datos. Ahora en esta función que añade registros se mostrará que antes comprueba si existe una base de datos ya creada dónde pueda almacenarlos. Tanto la creación como la agregación de registros emplea lenguaje SQLite. Además, será necesaria la incorporación de una librería para poder trabajar con este tipo de bases de datos.

Lo primero que realiza la función que agrega registros en la base de datos es contactar con una base de datos que, si no está creada, se crea en ese momento. Después emplea un puntero que recorrerá los registros, utilizará los datos que le paso la función de recogida de datos y los almacenará en su lugar correspondiente. Finalizada la agregación de los registros, la comunicación con la base de datos debe ser concluida. Las líneas de código que implementan esta función aparecen en la Figura 7.8.

```
import sqlite3
#Función para añadir los datos en los campos
def AnadirDatos(Fecha, Hora, nivelCO2, Temp, HumRel):
    ConexionBDD()
    conexionBDD=sqlite3.connect("Aplicación")
    cursor=conexionBDD.cursor()
    datos=Fecha, Hora, nivelCO2, Temp, HumRel
    cursor.execute("INSERT INTO REGISTRO VALUES(?,?,?,?,?)", datos)
    conexionBDD.commit()
```

Figura 7.8.- Agregación en base de datos de los parámetros ambientales medidos.

La función de conexión que es llamada desde esta última conecta con la base de datos, o la crea en caso de que no existiera, y genera también un puntero para recorrerla. La Figura 7.9 muestra cómo son las líneas de código de esta función de conexión con la base de datos.

```
#Función para la creación y conexión con la BDD
def ConexionBDD():
    conexionBDD=sqlite3.connect("Aplicación")
    cursor=conexionBDD.cursor()
    try:
        cursor.execute('''
        #CREATE TABLE REGISTRO(
        #Fecha CHAR(10),
        #Hora CHAR(8),
        #Nivel CO2 INTEGER,
        #Temperatura DECIMAL(4,2),
        #Humedad DECIMAL(5,1)
        ''')
    except:
        print("Base de datos ya creada")
```

Figura 7.9.- Conexión con base de datos, creación de ésta en caso de que no existiera.

## V. Interpretación de los datos

Esta función está compuesta por un subconjunto de muchas de ellas, las cuales corresponden a una instrucción concreta que ordenará el sistema de gestión para que realice el de actuación. Esta interpretación de los datos está particularizada a la concentración de CO<sub>2</sub> que hubiera sido medida, además de un caso extremo en el que entra en juego la medición que se haya hecho de la temperatura. Por ello, en función a este valor del CO<sub>2</sub> se ordenará una acción u otra al sistema de actuación. Al no disponerse del módulo de sensores ni del variador de frecuencia no se va a poder analizar cómo deberían ser estas órdenes, por lo que se han creado diferentes acciones ante situaciones de funcionamiento habituales esperadas. Las líneas de código que recogen esta función aparecen en la Figura 7.10, mientras que las acciones contempladas se encuentran en la siguiente Figura 7.11.

```
#Funciones para el sistema de gestión
def ordenVentilador(nivelCO2,Temp, nivelAnt):
    if ((nivelCO2<600) & (Temp<=60)):
        if nivelCO2<nivelAnt:
            accion=apagaVentilador()
        else:
            accion=ventiladorApagado()
    elif ((nivelCO2>=600) & (nivelCO2<=615) & (Temp<=60)):
        accion=encenderVentilador()
    elif ((nivelCO2>615) & (nivelCO2<=1200) & (Temp<=60)):
        if nivelCO2<nivelAnt:
            accion=reduceVelocidad()
        else:
            accion=aumentaVelocidad()
    elif ((nivelCO2>1200) & (Temp<=60)):
        accion=velocidadMaxima()
    else:
        cuadroMensaje.insert(0, "Está habiendo un incendio, apagando ventilador")
        DesuscribirTopic()
        DesconectarBroker()
    return accion
```

Figura 7.10.- Instrucción ordenada por el sistema de gestión para el de actuación.

```
def ventiladorApagado():
    vApagado=("El ventilador se encuentra apagado")
    return vApagado

def apagaVentilador():
    vApagando=("Apaga el ventilador")
    return vApagando

def encenderVentilador():
    eVentilador=("Arranca el ventilador")
    return eVentilador

def reduceVelocidad():
    rVelocidad=("Reduce la velocidad de giro")
    return rVelocidad

def aumentaVelocidad():
    aVelocidad=("Aumenta la velocidad de giro")
    return aVelocidad

def velocidadMaxima():
    vMax=("Ventilador a máxima velocidad")
    return vMax
```

Figura 7.11.- Acciones que puede ordenar el sistema de gestión.



## VI. Publicación de las órdenes

Una vez conocida la orden que ha de enviarse al sistema de actuación, ésta se publicará en forma de mensaje bajo un cierto “topic”.

La función publica con el cliente MQTT creado al inicio de la aplicación bajo el “topic” de publicar, asignado en el subapartado 7.1.2 correspondiente a las variables, el mensaje con la orden dictada por el sistema de gestión. Las líneas de código que desarrollan esta función son las que aparecen en la Figura 7.12.

```
#Función para publicar la orden en el bróker
def publicar(orden):
    time.sleep(0.1)
    mqttc.publish(topic_act,orden)
    cuadroPublicacion.insert(0, "-----")
    cuadroPublicacion.insert(0, "Publicando mensaje")
    cuadroPublicacion.insert(0, "Mensaje publicado: "+orden)
```

Figura 7.12.- Publicación de la orden en el bróker.

### 7.1.4. Programa principal

El programa principal de esta aplicación simplemente consiste en acudir a la función de recepción de mensajes cuando el sistema de adquisición de datos publique el mensaje en el “topic” al cual está suscrito el sistema de gestión, actuando como rutina de “callback”. La línea de código que implementa esta llamada está reflejada en la Figura 7.13.

```
#ZONA: Programa principal
#Se define la rutina Callback cuando se reciban los datos por MQTT
mqttc.on_message=RecibirDatos
```

Figura 7.13.- Rutina de “callback” tras la recepción del mensaje.

### 7.1.5. Interfaz de la aplicación

El último paso será la creación de una interfaz gráfica para esta aplicación que permita interactuar con ella. Esta interfaz ha sido creada con la librería “tkinter”, como se ha comentado en el subapartado 7.1.1 de este capítulo, que permite la creación de interfaces gráficas de una manera sencilla. La interfaz cuenta con una serie de botones que permitan la conexión y desconexión con el bróker, así como para la suscripción y desuscripción del “topic”. Mientras que la publicación de la orden para el sistema de actuación se realiza de manera automática, sin opción de interacción.

Por último, la interfaz contará con una serie de cuadros de texto donde se mostrarán diversas informaciones dependiendo del cuadro. Esta información corresponde con la dirección IP del bróker, los “topics”, tanto al que se suscribe el sistema como con el que publica. Además de otros dos cuadros que permitirán ver lo que está sucediendo en la aplicación, si se conecta o desconecta del bróker, si se suscribe o desuscribe al “topic”, y los mensajes con los datos recibidos y con la orden publicada. La Figura 7.14 muestra el diseño de esta interfaz.



Figura 7.14.- Interfaz gráfica de la aplicación.

## 7.2. Sistema de actuación

El sistema de actuación es el encargado de producir una variación en la velocidad de giro del motor, regulando así el caudal de ventilación que se proporciona a la instalación en función de la indicación que le haya hecho el sistema de gestión. La comunicación entre el sistema de gestión y el de actuación se realiza de nuevo mediante el protocolo MQTT a través del bróker, tal y como sucedía anteriormente entre el sistema de actuación y de gestión.

Este sistema de actuación va a estar compuesto por dos elementos principales, una placa de desarrollo y un variador de frecuencia. La interconexión de estos dos elementos se realizará mediante el bus Modbus que emplea el protocolo de comunicación con su mismo nombre.

El primero de ellos, la placa de desarrollo se encontrará suscrita al “topic” bajo el cual publique el sistema de gestión la modificación oportuna que deba hacerse en la velocidad de giro del motor. La placa de desarrollo que se utilizaría sería la misma que se ha empleado en el sistema de adquisición, el NodeMCU, ya que su microcontrolador puede conectarse a internet y, además, permite la comunicación mediante este protocolo Modbus. Las características de esta placa de desarrollo aparecen perfectamente documentadas en el subapartado 5.2 de este proyecto, por lo tanto, no se repetirán.

Del protocolo Modbus y del variador de frecuencia se hablará más detenidamente en los dos siguientes subapartados. No obstante, la conexión entre el NodeMCU y el variador de frecuencia mediante este protocolo no forma parte de los objetivos del proyecto y, por lo tanto, está fuera de su alcance, quedando como trabajo futuro.

### 7.2.1. Protocolo Modbus

El protocolo Modbus es un protocolo de comunicación basado en una arquitectura maestro-esclavo o cliente-servidor. Este protocolo permite una comunicación fiable entre dispositivos de automatización y de campo [74], lo que le hace ser la opción más interesante para la comunicación entre los dos elementos del sistema de actuación de este proyecto.

Este protocolo fue diseñado por Modicon en 1979 para su uso en los controladores lógicos programables (PCLs). Este protocolo garantiza una transmisión de datos Ethernet ultrarrápida y, al tener una estructura de datos independiente del fabricante, permite la comunicación entre dispositivos de distintos fabricantes [74].

En una red que implemente el estándar de este protocolo puede haber hasta 247 esclavos controlados por el maestro. Estos esclavos sólo podrán mandar la información cuando se lo solicite el maestro, y éste, a su vez, puede escribir en ellos. La comunicación es siempre iniciada por el maestro, una vez que se establece el maestro procede a enviar el mensaje al esclavo que actúa en consecuencia y le responde [75].

Los modos operativos con los que puede trabajar el protocolo Modbus en función de la capa en la que se encuentre son los siguientes [74]:

- Modbus TCP: comunicación Ethernet basada en el modelo cliente-servidor. Tiene como requisito que los nodos han de encontrarse en el mismo intervalo de direcciones IP. Utiliza el puerto estándar 502 para la comunicación [75].
- Modbus RTU: comunicación maestro-esclavo de serie a través de los puertos RS-232 o RS-485, dependiendo de la longitud a la que se sitúen los esclavos del maestro, siendo de 15 m para el primer puerto y de 1200 m para el segundo.
- Modbus ASCII: es similar al RTU, pero empleando caracteres ASCII con 4 bits de datos cada uno. Actualmente se encuentra en desuso [75].

En la Figura 7.15 se muestra en un esquema el empleo habitual de los modos operativos anteriormente comentados en el protocolo Modbus. En ella se puede ver como los elementos de

las capas más altas que requieren Ethernet utilizan el modo TCP, mientras que las capas más bajas se conectan mediante el modo RTU.

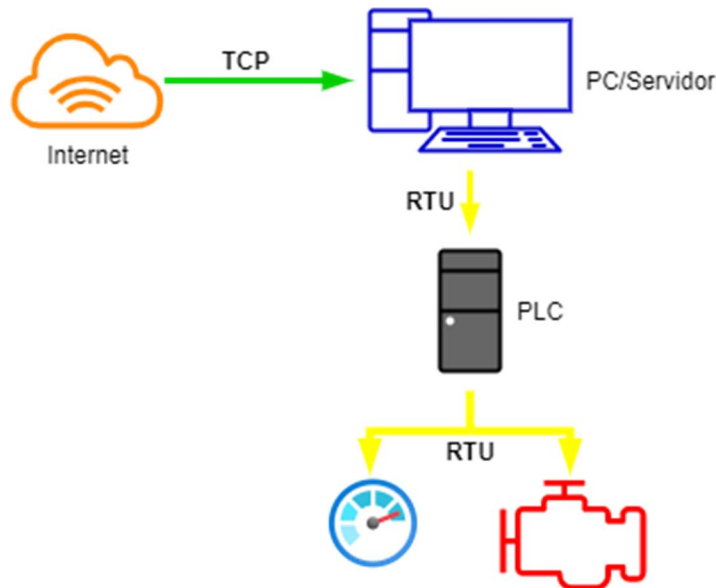


Figura 7.15.- Empleo habitual de los modos operativos en el protocolo Modbus.

Las ventajas principales con las que cuenta este protocolo de comunicación son [75]:

- Arquitectura simple: la arquitectura maestro-esclavo ofrece una gran flexibilidad operativa. Además, el modo TCP ofrece una velocidad de datos desde 10 hasta 100 Mbits/s para las topologías de anillo o estrella habituales.
- Fácil integración con las instalaciones industriales: utiliza conexiones de puerto serie estándares tanto para el modo TCP (RJ45) como para RTU (RS-485).
- Alta fiabilidad: este protocolo ofrece una alta fiabilidad al incorporar comprobaciones de errores que cubren la totalidad del mensaje.

En vista de todo lo comentado en este subapartado, el empleo de este protocolo de comunicación entre el NodeMCU y el variador de frecuencia queda justificado gracias a la simpleza de su arquitectura, su velocidad de transferencia de datos y su contrastado uso en la industria para el control y la monitorización de actuadores.

### 7.2.2. Variador de frecuencia

El variador de frecuencia es el elemento que proporciona al motor eléctrico del ventilador un determinado nivel de tensión que actúa directamente sobre la velocidad de giro del motor de corriente alterna [76], provocando, al producirse esta perturbación, que el caudal de aire que proporciona el ventilador se vea afectado según requiera la situación. El funcionamiento de un variador de frecuencia aparece recogido esquemáticamente en la Figura 7.16.

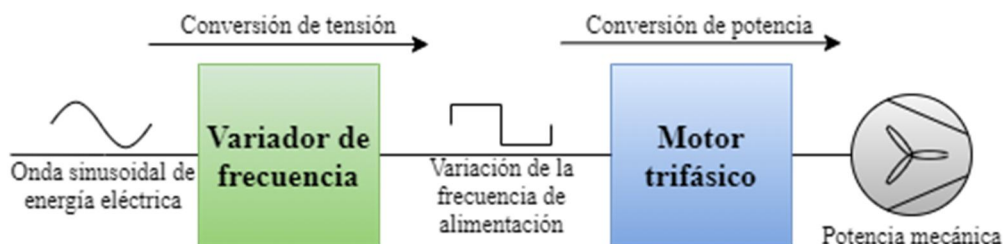


Figura 7.16.- Esquema de funcionamiento de un variador de frecuencia.

Aunque en la actualidad se estima que menos del 10% de los motores eléctricos a nivel mundial están equipados con variadores de frecuencia, está comprobado que estos permiten reducir el consumo energético de las instalaciones, así como las emisiones de CO<sub>2</sub>. El variador de frecuencia permite que la velocidad de giro del motor coincida exactamente con la que se necesita en cada momento, ahí radica la clave de la reducción del consumo energético que se sitúa entre el 20 y el 70% en función de la instalación [77]. El control de la velocidad de giro está a su vez ligado con la potencia proporcionada por el motor, esto conlleva, al reducirse la potencia en ciertos periodos de tiempo (con la estancia vacía, por ejemplo), alargar la vida útil de los equipos, debido a que se deterioraran menos. Además, el variador no va a necesitar prácticamente mantenimiento [76].

El variador de frecuencia consta, intrínsecamente, de un rectificador que transforma la corriente alterna en corriente continua, unos condensadores que suavizan la forma de onda de la corriente eléctrica resultante al cargarse de la corriente continua y, por último, el inversor que vuelve a transformar la corriente continua en alterna ya regulada que es la que recibe el motor eléctrico del ventilador [76]. En la Figura 7.17 se muestra el diagrama de un variador de frecuencia con los tres elementos mencionados en este párrafo.

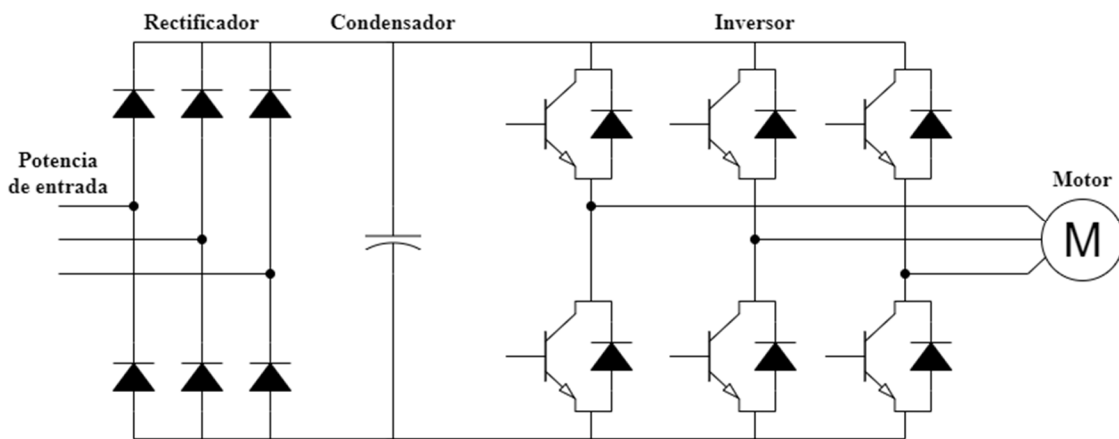


Figura 7.17.- Diagrama de un variador de frecuencia.

En el mercado existe gran variedad de variadores de frecuencia, no obstante, el variador que ha de seleccionarse debe poder contar con un protocolo de comunicación Modbus para su control. Además, debe de ser capaz de operar con el motor trifásico que acciona los ventiladores, cuyas potencias, cabe recordar del subapartado 4.3.3, eran de 1,5 kW para la impulsión de aire y de 1,1 kW para la extracción.

La placa de desarrollo le hará llegar el mensaje a través del protocolo Modbus indicándole al variador el nivel de tensión que ha de proporcionar al motor. Esto va a permitir un control absoluto sobre el caudal de ventilación que se va a proporcionar.

Con todo ello el variador de frecuencia seleccionado, la instalación deberá contar con dos de ellos uno para la impulsión y otro para la extracción, es el modelo ACH480 de la compañía ABB, Figura 7.18, este modelo cumple con los principales requisitos exigidos tanto de comunicación Modbus, como de motor, ya que pueden trabajar con los motores trifásicos mencionados en el párrafo anterior. Otras características incorporadas de serie en este modelo son las siguientes [78]:

- Pueden trabajar con frecuencias de la red eléctrica entre 48 y 63 Hz. En España la frecuencia de red son 50 Hz, que es la utilizada en este proyecto.
- La frecuencia de conexión al motor llega hasta los 599 Hz, permitiendo la monitorización de la velocidad de giro de este.
- El modo de control del motor permite tanto control escalar como vectorial.

## 7. SISTEMAS DE GESTIÓN Y ACTUACIÓN

---

- Los motores que puede gobernar podrán ser síncronos, asíncronos y de imán permanente.
- La temperatura de trabajo está comprendida entre los -40 y los 70 °C.
- La humedad relativa de trabajo para evitar condensaciones puede oscilar entre el 5 y el 95%.
- Cuenta con un grado de protección IP20, que evita la entrada de polvo al interior del variador.
- Dos entradas y dos salidas analógicas que se podrán seleccionar para corriente o tensión.
- Una señal de tensión que puede variar entre 0 y 10 V.
- Una señal de corriente que va desde los 0 a los 20 mA.
- El variador incluye seis salidas digitales.
- Los protocolos de comunicación que incluye son:
  - BACnet MS/TP.
  - Modbus RTU.
  - N2.
- Dispone de dos de los siguientes puertos de conexión plug-in:
  - BACnet/IP.
  - Modbus TCP.
  - PROFINET I/O.
  - EtherNet/IP.
  - EtherCAT.
  - EtherNet POWERLINK.



Figura 7.18.- Variador de frecuencia modelo ACH480 de ABB (imagen obtenida del catálogo de ABB [78]).

Una vez seleccionado y documentado el variador de frecuencia, los elementos del sistema de actuación quedarían definidos, y con él todos los sistemas con los que contará la instalación de ventilación para su monitorización y control.

# 8. Conclusiones y trabajos futuros

En este capítulo se verán las conclusiones y se analizarán los trabajos futuros que han de realizarse para poder implementar este tipo de monitorización de la calidad del aire interior en el salón de actos objeto de estudio.

En el subapartado de conclusiones se comprobará si se han conseguido alcanzar los dos objetivos fijados al inicio de este proyecto (véase subapartado 1.2) y se analizará que pasos se han seguido para lograrlos.

En el subapartado de trabajos futuros se explicarán las limitaciones que tiene el proyecto, actualmente, y que acciones será conveniente realizar para poder implementar, en un futuro, este proyecto en la realidad.

## 8.1. Conclusiones

En el presente proyecto se ha visto como llevar a cabo la monitorización de un sistema de ventilación en función de la calidad del aire interior de un salón de actos.

El proyecto tenía dos objetivos claros:

- Realización de un cálculo fluidodinámico.
- Diseño de los tres sistemas, de adquisición de datos, de gestión y de actuación. Además de mostrar cómo se realiza el protocolo de comunicación MQTT entre ellos.

Del primero de los objetivos, el cálculo fluidodinámico, se pretendía obtener los valores del caudal de ventilación necesario para la estancia, las pérdidas de carga que generaba la instalación y la potencia específica, resultados indispensables en toda instalación de ventilación. Para la consecución de este objetivo se han seguido las directrices indicadas por el RITE [1] y la Guía Técnica número 10, titulada *Selección de equipos de transporte de fluido* [3], para el procedimiento de cálculo. Una vez realizado el cálculo se ha proseguido con la elección de los equipos de ventilación que mejor se ajustan a la instalación. Por lo tanto, con la selección de los equipos realizada, se puede concluir que el objetivo ha sido cumplimentado y el cálculo fluidodinámico queda definido.

Del segundo de ellos, se ha diseñado cada uno de los tres sistemas que conforman la instalación, adquisición de datos, gestión y actuación, y se ha explicado cómo se realiza la comunicación MQTT entre ellos. El diseño de los sistemas competía a los elementos que lo formaban, así como la comunicación entre cada uno de estos elementos.

Los elementos del sistema de adquisición de datos fueron el módulo de sensores y el NodeMCU. El módulo permitía realizar la medición de los tres parámetros ambientales elegidos para el proyecto, la concentración de CO<sub>2</sub> en el ambiente, la temperatura y la humedad relativa. Mientras que el NodeMCU incorporaba un microcontrolador con el cual se conectaba a internet para enviar mensajes al sistema de gestión a través de un bróker MQTT. Estos dos elementos se conectaban entre sí mediante otro protocolo de comunicación, el I<sup>2</sup>C, en el cual el papel del maestro era ejercido por el microcontrolador que solicitaba los datos al módulo de sensores, que actuaba como esclavo. Posteriormente, el microcontrolador publicaba esos datos en el bróker. Para realizar estas acciones, tanto la toma de datos por parte de los sensores como la publicación de los datos en el bróker, se ha creado una aplicación en el programa Arduino que, haciendo uso de una serie de librerías, las consigue efectuar.

El segundo de los sistemas, el de gestión, realizaba dos funciones, por un lado, se suscribía a los datos publicados en el bróker por el microcontrolador del sistema de adquisición de datos y los almacenaba en una base de datos, datando cuando habían tenido lugar, y, por otro lado, interpretaba estos datos para publicar en el bróker la orden que debía llevar a cabo el

sistema de actuación. Nuevamente, para este sistema se ha creado una aplicación, en este caso en el programa Python, que se encargaba de estas dos funciones.

El tercero de los sistemas correspondía al de actuación y estaba formado por un NodeMCU y un variador de frecuencia. El microcontrolador del NodeMCU debía de suscribirse al “topic” en el que se publican las órdenes de arranque, paro y velocidad de funcionamiento del motor del ventilador, que hubiera publicado el sistema de gestión, para posteriormente interpretarla y comunicarle al variador de frecuencia cómo debía actuar en consecuencia, y éste, a su vez, proporcionaría el nivel de tensión adecuado al motor en función de la orden recibida. La comunicación entre estos dos elementos se realizaba mediante el protocolo Modbus, que también emplea una estructura maestro-esclavo para su funcionamiento.

Finalmente, la comunicación entre estos tres sistemas se llevaba a cabo mediante el protocolo de comunicación MQTT, que sigue un servicio de mensajería de publicación-suscripción, en el cual un dispositivo publica un mensaje bajo un cierto “topic” en el bróker MQTT haciendo que otro dispositivo suscrito a ese mismo “topic” pueda leerlo.

Comprobado el funcionamiento de las dos aplicaciones, tanto la diseñada en Arduino como la de Python, se puede asegurar que el segundo de los objetivos también se ha alcanzado, ya que al lograr la plena funcionalidad de las aplicaciones se ha conseguido la monitorización de la calidad del aire interior del salón que se buscaba.

Una vez cumplidos los objetivos se comentarán los trabajos futuros que deberían realizarse en el proyecto para llevarlo a la práctica en el salón de actos objeto de estudio.

### 8.2. Trabajos futuros

Los trabajos futuros que se deberían llevar a cabo en este proyecto, para poder implementarlo en la realidad, competen principalmente a la compra de los equipos de ventilación y de los elementos los diferentes sistemas, en especial los de adquisición de datos y de actuación. Sin embargo, existen diversos aspectos del proyecto que se han tenido que suponer debido al desconocimiento de los datos y que se comentarán a continuación.

En primer lugar, se debería realizar un estudio detallado tanto del salón de actos como del Edificio Polivalente, debido a que únicamente se conocían la superficie del salón y su aforo. En el caso del salón sería necesario el conocimiento de la altura del techo de la estancia, lo que incidiría sobre elección de la velocidad del aire a través de los conjuntos y en la selección de los difusores. Por otro lado, se desconocía, a la hora de la realización del proyecto, si era viable la colocación de una UTA en el tejado del edificio, la situación de esta y la distancia que hubiera entre ella y el salón determinaría la longitud de los conductos, así como los accesorios (codos, bifurcaciones, etc) que habrían de instalarse tras el diseño de la red de conductos.

Finalizado el diseño de la red de conductos se deberían recalcular las pérdidas de carga, de igual forma que se ha hecho en el subapartado 4.3.2, y comprobar si fueran válidos los ventiladores seleccionados. No obstante, no se esperan grandes cambios a este respecto.

Diseñada la red de conductos y comprobada la validez de los ventiladores elegidos, el siguiente paso sería comprar todos los equipos de ventilación y los conductos y accesorios de dicha red. Para su instalación será necesario la creación de un pliego de condiciones en el que se atestigüe el procedimiento y la posición que ha de ocupar cada uno de estos equipos en el sistema de ventilación cumpliendo la normativa vigente.

Dos de los tres sistemas, los de adquisición y actuación, empleados para la monitorización padecen el mismo problema que el sistema de ventilación, la carencia de algunos de los elementos seleccionados, por su elevado coste, para poder realizar las pruebas de funcionalidad pertinentes que se deberían haber llevado a cabo durante la elaboración del proyecto. Por lo tanto, las pruebas de las aplicaciones creadas han sido realizada suponiéndose



---

## 8. CONCLUSIONES Y TRABAJOS FUTUROS

---

los datos, para el sistema de adquisición, y las órdenes que se encomiendan, para el de actuación.

En este caso únicamente se ha podido utilizar el NodeMCU, el resto de los elementos de estos dos sistemas no han estado disponibles. Es por ello, que se ha confiado en la documentación del desarrollador del módulo de sensores para diseñar el código de la aplicación en Arduino en el momento de la toma de datos y en el de su calibración. Así mismo, se han estudiado las conexiones, tanto físicas como de protocolo de comunicación, mediante el bus I<sup>2</sup>C entre el NodeMCU y el módulo de sensores, aunque tampoco se ha podido contar con este bus durante el proceso de elaboración del proyecto.

De forma similar ha ocurrido con el variador de frecuencia del sistema de actuación, en esta cuestión, además, se carecía del conocimiento de la conexión entre el variador y el motor del ventilador, no obstante, se ha comprobado la validez la compatibilidad de ambos en caso de una futura compra conjunta. El bus Modbus de comunicación entre el NodeMCU y el variador de frecuencia ocurre lo mismo que en los casos anteriores, se carecía de él. Además, el establecimiento de la conexión a través del empleo de este bus de comunicación se encuentra fuera del alcance del proyecto, y ha quedado como trabajo futuro, al igual que sucedía con la conexión entre el variador y el motor.

Por lo tanto, estos dos sistemas habría que probarlos una vez comprados todos sus elementos. Se prevé que no aparezcan errores durante la instalación y puesta en marcha del módulo de sensores, ya que se han seguido las directrices dadas por el desarrollador. Mientras que las conexiones entre los elementos del sistema de actuación, y entre éste y el motor, deban estudiarse en profundidad para su futura puesta en funcionamiento.

Cumplimentados estos trabajos, se espera que sea posible configurar todos los equipos, probar su funcionamiento individualizado y realizar la puesta en marcha de toda la instalación en conjunto, para monitorizar la calidad del aire interior del salón.

## 9. Bibliografía

- Ministerio de la Presidencia, «Reglamento de Instalaciones Térmicas en los Edificios,» [1] Madrid, 2007.
- Instituto para la Diversificación y Ahorro de la Energía, «IDAE,» [En línea]. Available: [2] <https://www.idae.es/tecnologias/eficiencia-energetica/edificacion/reglamento-de-instalaciones-termicas-de-los-edificios>. [Último acceso: 17 Mayo 2021].
- ATECYR, «Guía Técnica 10: Selección de equipos de transporte de fluido,» IDAE, [3] Madrid, 2012.
- ATECYR, «Guía Técnica 17: Instalaciones de climatización con equipos autónomos,» [4] IDAE, Madrid, 2012.
- J. Wang, M. Lim, C. Wang y M. Tseng, «The evolution of the Internet of Things (IoT) over the past 20 years,» *Computers & Industrial Engineering*, vol. 155, 2021. [5]
- j. (. d. usuario), «Pixabay,» 25 Mayo 2015. [En línea]. Available: [6] <https://pixabay.com/es/vectors/red-iot-internet-de-las-cosas-782707/>. [Último acceso: 21 Marzo 2021].
- OCU, «OCU,» 31 Agosto 2020. [En línea]. Available: [7] <https://www.ocu.org/tecnologia/internet-telefonía/informe/tecnologia-5g#:~:text=EI%20G%20permite%20que%20haya,Menor%20latencia..> [Último acceso: 22 Marzo 2021].
- Deloitte, «Deloitte,» [En línea]. Available: [8] <https://www2.deloitte.com/es/es/pages/manufacturing/articles/que-es-la-industria-4.0.html>. [Último acceso: 22 Marzo 2021].
- ISOTools, «ISOTools,» 18 Julio 2018. [En línea]. Available: [9] <https://www.isotools.org/2018/07/12/industria-4-0-que-debemos-saber/>. [Último acceso: 22 Marzo 2021].
- Wikipedia, «Wikipedia,» 14 Diciembre 2014. [En línea]. Available: [10] [https://es.wikipedia.org/wiki/Industria\\_4.0](https://es.wikipedia.org/wiki/Industria_4.0). [Último acceso: 14 Mayo 2021].
- Escuela Politécnica de Ingeniería de Gijón, «Epigijon,» [En línea]. Available: [11] <https://epigijon.uniovi.es/laescuela/edificios/polivalente>. [Último acceso: 1 Mayo 2021].
- Universidad de Oviedo, «GISUniovi,» [En línea]. Available: [12] <https://gis.uniovi.es/GISUniovi/MostrarDetalle.do?idDetalle=229&filtroPlanta=P2&tipoDetalle=dPlanta>. [Último acceso: 1 Mayo 2021].
- G. Maese y J. C. Tuero, «El Comercio,» 4 Septiembre 2020. [En línea]. Available: [13] <https://www.elcomercio.es/asturias/universidad/incertidumbre-curso-debe-20200904000804-ntvo.html>. [Último acceso: 15 Febrero 2021].
- Servicio de Protección del Medioambiente del Ayuntamiento de Gijón, «Calidad del Aire en Gijón,» Gijón, 2016. [14]
- A. Vegas Casado, «Difusión de aire, acústica y confort,» *El Instalador*, nº 459, pp. 18-35, [15] 2009.

---

## LISTA DE ACRÓNIMOS

---

- F. C. McQuiston, P. J. D. y S. J. D., Heating, Ventilating, and Air Conditioning, John Wiley & Sons, Inc., 2005.
- FEXMEX, «FEXMEX Fluid Equipment Supply,» 22 Enero 2020. [En línea]. Available: [17] <http://www.fesmex.com.mx/articulo/perdidas-por-friccion-en-tuberias>. [Último acceso: 15 Marzo 2021].
- ATECYR, «Guía Técnica 12: Condiciones climáticas exteriores de proyecto,» IDAE, [18] Madrid, 2010.
- Casals, «Casals,» 2021. [En línea]. Available: [19] <https://www.casals.com/es/fanware/50/fans/15230-bvc-m-15-15?flow=6096&pressure=46.4991&temperature=20&height=0>. [Último acceso: 10 Mayo 2021].
- Casals, «Casals,» 2021. [En línea]. Available: [20] <https://www.casals.com/es/fanware/50/fans/15199-bstr-m-22-11?flow=6096&pressure=32.1211&temperature=20&height=0>. [Último acceso: 10 Mayo 2021].
- SystemAir, «SystemAir,» [En línea]. Available: <https://www.systemair.com/es/quienes-somos/>. [Último acceso: 10 Mayo 2021].
- Venfilter, «Venfilter,» [En línea]. Available: <https://www.venfilter.es/normativa/guia-comparativa-de-normas-para-la-clasificacion-de-los-filtros-de-aire>. [Último acceso: 10 Abril 2021].
- Ald Air Industrial Filtration, «Grupo Cartés Industria,» [En línea]. Available: [23] [https://www.filtrosartes.es/media/descargas/informacion-tecnica/FILTROS\\_BOLSA\\_.pdf](https://www.filtrosartes.es/media/descargas/informacion-tecnica/FILTROS_BOLSA_.pdf). [Último acceso: 16 Abril 2021].
- Venfilter, «Venfilter,» [En línea]. Available: <https://www.venfilter.es/producto/bio-bag-60>. [24] [Último acceso: 6 Marzo 2021].
- Venfilter, «Venfilter,» [En línea]. Available: <https://www.venfilter.es/producto/bag-glass-80-galvanizado>. [25] [Último acceso: 6 Marzo 2021].
- ATECYR, «Guía Técnica 9: Ahorro y recuperación de energía en instalaciones de climatización,» IDAE, Madrid, 2012.
- SystemAir, «SystemAir,» 2020. [En línea]. Available: [27] [https://shop.systemair.com/upload/assets/GENIOX\\_CATALOGUE\\_GLOBAL\\_2020\\_E2602.PDF?090d73d5](https://shop.systemair.com/upload/assets/GENIOX_CATALOGUE_GLOBAL_2020_E2602.PDF?090d73d5). [Último acceso: 10 Mayo 2021].
- Schako, «Schako,» 18 Febrero 2021. [En línea]. Available: [https://schako.com/wp-content/uploads/wda\\_es.pdf](https://schako.com/wp-content/uploads/wda_es.pdf). [28] [Último acceso: 9 Abril 2021].
- Schako, «Schako,» [En línea]. Available: <https://schako.com/es/schako-products/tobera-de-gran-alcance-wda/>. [29] [Último acceso: 9 Abril 2021].
- Euroclima, «Euroclima,» [En línea]. Available: [30] <http://www.euroclima.es/uploads/document/576/E-LOAD.pdf>. [Último acceso: 9 Abril 2021].
- Euroclima, «Euroclima,» [En línea]. Available: <http://www.euroclima.es/familias/1/sub-familias/5/productos/17>. [31] [Último acceso: 9 Abril 2021].
- D. Ibrahim, «Chapter 1 - Microcomputer systems,» de *Arm-Based Microcontroller*
-

- [32] *Multitasking Projects*, Newnes, 2020, pp. 1-12.
- Electrónica Lugo, «Electrónica Lugo,» 2021. [En línea]. Available:  
[33] <https://electronicalugo.com/conceptos-basicos-tipos-y-aplicaciones-de-microcontroladores/>. [Último acceso: 17 Abril 2021].
- B. Giovino, «Digi-Key,» 21 Mayo 2019. [En línea]. Available:  
[34] <https://www.digikey.es/es/blog/the-8-bit-microcontroller-is-still-a-great-option-for-low-power-and-ease-of-use>. [Último acceso: 17 Abril 2021].
- B. Giovino, «Digi-Key,» 24 Julio 2019. [En línea]. Available:  
[35] <https://www.digikey.es/articles/take-advantage-of-16-bit-mcu-performance-and-low-power>. [Último acceso: 17 Abril 2021].
- R. Shankar y E. B. Fernandez, «Chapter 13 - Extended Von Neumann Architectures,» de  
[36] *VLSI and Computer Architecture*, San Diego, Academic Press, Inc., 1989, pp. 385-396.
- R. Shankar y E. B. Fernandez, «Chapter 15 - Harvard Architectures,» de *VLSI and*  
[37] *Computer Architecture*, San Diego, Academic Press, Inc., 1989, pp. 405-408.
- E. Conrad, S. Misenar y J. Feldman, «Chapter 4 - Domain 3: Security Engineering  
[38] (Engineering and Management of Security),» de *CISSP Study Guide*, Waltham, Syngress, 2016, pp. 103-217.
- Espressif IOT Team, «Espressif,» 2020. [En línea]. Available:  
[39] [https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf). [Último acceso: 26 Marzo 2021].
- M. J. Bellido, *Introducción al Diseño de SoC (Systems On Chip)*, Sevilla, 2017.  
[40]
- NodeMCU Team, «NodeMCU,» 2018. [En línea]. Available:  
[41] [https://www.nodemcu.com/index\\_en.html#fr\\_54745c8bd775ef4b99000011](https://www.nodemcu.com/index_en.html#fr_54745c8bd775ef4b99000011). [Último acceso: 26 Marzo 2021].
- M. Kashyap, V. Sharma y N. Gupta, «Taking MQTT and NodeMcu to IOT:  
[42] Communication in Internet of Things,» *Procedia Computer Science*, vol. 132, pp. 1611-1618, 2018.
- Advanced Monolithic Systems, Inc., «Advanced Monolithic Systems, Inc.,» [En línea].  
[43] Available: <http://www.advanced-monolithic.com/pdf/ds1117.pdf>. [Último acceso: 19 Abril 2021].
- D. Eisenreich y B. DeMuth, «The I2C Bus,» de *Designing Embedded Internet Devices*,  
[44] Newnes, 2002, pp. 433-466.
- A. Mihaela, A. Enache, A. Negut, A. Macarie y G. Brezeanu, «An improved digital output  
[45] buffer for a digitar temperature sensor with I2C high speed interface,» *Solid State Electronic Letters*, vol. 1, nº 2, pp. 147-151, 2020.
- Kim, «Sandbox Electronics,» 3 Febrero 2014. [En línea]. Available:  
[46] <https://sandboxelectronics.com/?p=147>. [Último acceso: 23 Abril 2021].
- Winsen, «Zhengzhou Winsen Electronics Technology Co., Ltd,» 10 Marzo 2015. [En  
[47] línea]. Available: <https://datasheetspdf.com/pdf/1415579/Winsen/MG811/1>. [Último acceso: 16 Febrero 2021].
- Adafruit, «Mouser,» 17 Enero 2018. [En línea]. Available:
-

- [48] <https://www.mouser.es/pdfdocs/adafruit-sgp30-gas-tvoc-eco2-mox-sensor.pdf>. [Último acceso: 16 Febrero 2021].
- Adafruit, «Adafruit,» 30 Diciembre 2020. [En línea]. Available: [https://cdn-](https://cdn-learn.adafruit.com/downloads/pdf/adafruit-ccs811-air-quality-sensor.pdf)
- [49] [learn.adafruit.com/downloads/pdf/adafruit-ccs811-air-quality-sensor.pdf](https://cdn-learn.adafruit.com/downloads/pdf/adafruit-ccs811-air-quality-sensor.pdf). [Último acceso: 9 Abril 2021].
- Silicon Laboratories, «Silicon Labs,» 2016. [En línea]. Available:
- [50] <https://www.silabs.com/documents/public/data-sheets/Si7021-A20.pdf>. [Último acceso: 17 Febrero 2021].
- Aosong Electronics Co.,Ltd, «Sparkfun,» [En línea]. Available:
- [51] <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>. [Último acceso: 15 Marzo 2021].
- HwLibre, «Hardwarelibre,» [En línea]. Available:
- [52] [https://www.hwlibre.com/dht22/#Pinout\\_caracteristicas\\_y\\_datasheet](https://www.hwlibre.com/dht22/#Pinout_caracteristicas_y_datasheet). [Último acceso: 15 Marzo 2021].
- Bosch Sensortec, «Adafruit,» 5 Mayo 2015. [En línea]. Available: [https://cdn-](https://cdn-shop.adafruit.com/datasheets/BST-BMP280-DS001-11.pdf)
- [53] [shop.adafruit.com/datasheets/BST-BMP280-DS001-11.pdf](https://cdn-shop.adafruit.com/datasheets/BST-BMP280-DS001-11.pdf). [Último acceso: 15 Marzo 2021].
- Aliexpress, «Aliexpress,» [En línea]. Available:
- [54] [https://es.aliexpress.com/item/32869139805.html?src=google&albch=shopping&acnt=439-079-4345&isdl=y&slnk=&plac=&mtctp=&albbt=Google\\_7\\_shopping&aff\\_platform=google&aff\\_short\\_key=UneMJZVf&gclsrc=aw.ds&&albagn=888888&&ds\\_e\\_adid=438858099973&ds\\_e\\_matchtype=&ds](https://es.aliexpress.com/item/32869139805.html?src=google&albch=shopping&acnt=439-079-4345&isdl=y&slnk=&plac=&mtctp=&albbt=Google_7_shopping&aff_platform=google&aff_short_key=UneMJZVf&gclsrc=aw.ds&&albagn=888888&&ds_e_adid=438858099973&ds_e_matchtype=&ds). [Último acceso: 19 Abril 2021].
- Sensirion, «Sensirion,» Mayo 2020. [En línea]. Available:
- [55] [https://www.sensirion.com/fileadmin/user\\_upload/customers/sensirion/Dokumente/9.5\\_CO2/Sensirion\\_CO2\\_Sensors\\_SCD30\\_Datasheet.pdf](https://www.sensirion.com/fileadmin/user_upload/customers/sensirion/Dokumente/9.5_CO2/Sensirion_CO2_Sensors_SCD30_Datasheet.pdf). [Último acceso: 15 Marzo 2021].
- Envira IoT, «Envira IoT,» 2018. [En línea]. Available: <https://enviraiot.es/nanoenvi-iaq/>.
- [56] [Último acceso: 15 Marzo 2021].
- Protronix, «Care For Air,» [En línea]. Available:
- [57] <https://www.careforair.eu/en/recommendations-for-placement-of-air-quality-sensors/>. [Último acceso: 5 Marzo 2021].
- Fritzing, «Fritzing,» [En línea]. Available: <https://fritzing.org/>. [Último acceso: 20 Abril 2021].
- [58] 2021].
- Sensirion, «Sensirion,» Mayo 2020. [En línea]. Available:
- [59] [https://www.sensirion.com/fileadmin/user\\_upload/customers/sensirion/Dokumente/9.5\\_CO2/Sensirion\\_CO2\\_Sensors\\_SCD30\\_Interface\\_Description.pdf](https://www.sensirion.com/fileadmin/user_upload/customers/sensirion/Dokumente/9.5_CO2/Sensirion_CO2_Sensors_SCD30_Interface_Description.pdf). [Último acceso: 15 Mayo 2021].
- J. Pérez y A. Gardey, «Deficiencia.de,» 2015. [En línea]. Available:
- [60] <https://definicion.de/protocolo-de-comunicacion/>. [Último acceso: 26 Abril 2021].
- L. F. Pedraza-Martínez, C. A. Hernández-Suárez y D. A. López-Sarmiento, «Sistema de comunicación TCP/IP para el control de una intersección de tráfico vehicular,» *Ingeniería Investigación y Tecnología*, vol. XIV, n° 4, pp. 583-594, 2013.
- [61] 2013.
- Sabas, «Medium,» 25 Septiembre 2017. [En línea]. Available:

- [62] <https://medium.com/beelan/haciendo-iot-con-lora-cap%C3%ADtulo-1-qu%C3%A9-es-lora-y-lorawan-8c08d44208e8>. [Último acceso: 26 Abril 2021].
- CatSensors, «CatSensors,» [En línea]. Available:
- [63] <https://www.catsensors.com/es/lorawan/tecnologia-lora-y-lorawan>. [Último acceso: 26 Abril 2021].
- Venco, «Venco,» 3 Diciembre 2020. [En línea]. Available: <https://www.vencoel.com/que-es-zigbee-como-funciona-y-caracteristicas-principales/>. [Último acceso: 27 Abril 2021].
- [64] Domótica en casa, «Domótica en casa,» 2018. [En línea]. Available: <https://domoticaencasa.es/que-es-zigbee/>. [Último acceso: 27 Abril 2021].
- ZigBee Alliance, «ZigBee Alliance,» 2020. [En línea]. Available: <https://zigbeealliance.org/es/soluci%C3%B3n/Zigbee/>. [Último acceso: 27 Abril 2021].
- [66] E. Rodríguez de Luis, «Xataka,» 5 Marzo 2019. [En línea]. Available: <https://www.xataka.com/seleccion/zigbee-z-wave-que-que-se-diferencian-que-marcas-domotica-compatibles>. [Último acceso: 27 Abril 2021].
- [67] M. Yuan, «IBM developer,» 10 Octubre 2017. [En línea]. Available: <https://developer.ibm.com/es/technologies/iot/articles/iot-mqtt-why-good-for-iot/#:~:text=Originariamente%2C%20MQTT%20fue%20inventado%20y,comunicaci%C3%B3n%20as%C3%ADncrona%20entre%20las%20partes..> [Último acceso: 17 Mayo 2021].
- [68] MQTT.org, «MQTT.org,» 2020. [En línea]. Available: <https://mqtt.org/>. [Último acceso: 24 Marzo 2021].
- [69] Isaac, «Hardwarelibre,» [En línea]. Available: <https://www.hwlibre.com/mqtt/>. [Último acceso: 28 Abril 2021].
- [70] J. E. Crespo, «Aprendiendo Arduino,» [En línea]. Available: <https://aprendiendoarduino.wordpress.com/tag/seguridad-mqtt/>. [Último acceso: 17 Mayo 2021].
- [71] Mosquitto.org, «Mosquitto,» [En línea]. Available: <https://mosquitto.org/>. [Último acceso: 29 Marzo 2021].
- [72] Arduino, «Arduino,» [En línea]. Available: <https://arduino.cl/que-es-arduino/>. [Último acceso: 1 Abril 2021].
- [73] Wago, «Wago,» 28 Julio 2020. [En línea]. Available: <https://www.wago.com/es/modbus>. [Último acceso: 2 Junio 2021].
- [74] Aula 21, «Aula 21 Centro de formación técnica para la industria,» [En línea]. Available: <https://www.cursosaula21.com/modbus-que-es-y-como-funciona/>. [Último acceso: 2 Junio 2021].
- [75] S&P, «Soler & Palau,» 7 Enero 2020. [En línea]. Available: <https://www.solerpalau.com/es-es/blog/variador-de-frecuencia/>. [Último acceso: 30 Marzo 2021].
- [76] ABB, «ABB,» 2021. [En línea]. Available: <https://new.abb.com/drives/es/que-es-un-variador>. [Último acceso: 30 Marzo 2021].
- [77] ABB, «ABB,» 2 Septiembre 2019. [En línea]. Available: <https://search.abb.com/library/Download.aspx?DocumentID=3AUA0000231852&Langua>
-

## LISTA DE ACRÓNIMOS

---

- [78] geCode=en&DocumentPartId=&Action=Launch. [Último acceso: 31 Marzo 2021].
- M. J. Berenguer Subils y F. Bernal Domínguez, «NTP 549: El dióxido de carbono en la
- [79] evaluación de la calidad del aire interior,» Instituto Nacional de Seguridad e Higiene en el Trabajo, 2000.
- Fanware, «Casals,» 2021. [En línea]. Available:
- [80] <https://www.casals.com/es/fanware/50/fans/mtrl-atex>. [Último acceso: 31 Marzo 2021].
- Fanware, «Casals,» 2021. [En línea]. Available:
- [81] <https://www.casals.com/es/fanware/50/fans/15199-bstr-m-22-11?flow=6096&pressure=32.9695&temperature=20&height=0>. [Último acceso: 31 Marzo 2021].





# **ANEXOS**



## Anexo A

En este anexo se mostrará la aplicación creada en el programa Arduino para la recogida de los datos ambientales por parte del sistema de adquisición de datos. Esta aplicación, ya descrita en el subapartado 6.4 del proyecto, se encarga de realizar la medición de los tres parámetros ambientales, concentración de CO<sub>2</sub>, temperatura y humedad relativa, por medio del módulo de sensores SCD30 y publicarla en el bróker MQTT bajo un determinado “topic” gracias a la conexión a internet que ofrece el microcontrolador del NodeMCU. A continuación, se verán nuevamente las líneas de código que incluye esta aplicación comenzando por las zonas principales, y dejando para el final, cada una de las funciones que serán llamadas durante el funcionamiento del programa principal, así como el certificado de seguridad necesario para crear el cliente seguro TLS.

```
//Zona 1: Librerías

#include <string.h>
#include <stdlib.h>
#include <Wire.h>
#include <SparkFun_SCD30_Arduino_Library.h>
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include "Certificado_TLS.h"

//Zona 2: Variables globales y prototipos de las funciones

//Módulo de sensores
SCD30 airSensor;
uint16_t NivelCO2=0;
float Temp=0;
float HumRel=0;
```

---

## ANEXO A

---

```
//MQTT
//Nomenclatura de la red WiFi y contraseña
const char* ssid="WiFi_TFM";
const char* password="*****";

//Dirección IP del bróker al cual se conecta
const char* IP_mqtt_server="156.35.154.1";

//Se crea el inicio del cliente WiFi para conectarse al WiFi
String clientID="Marcos_";

//Cabeceras de los mensajes que se enviarán al bróker MQTT
const char* cabecera_MQTT="Datos ambientales: ";

//Se crean los strings dónde se escribirá el mensaje para el bróker
char MQTT_Datos[50]={0};

//"Topic" para el bróker MQTT
const char* topic_datos="TFM/datos";
//Zona 3: Setup

void setup() {

  Serial.begin(115200); //Se fija la velocidad de subida
  Serial.println("TFM utilizando el módulo SCD30"); //Título de las mediciones
  Wire.begin(D6,D7); //Se arranca la conexión a través de los puertos para I2C

  if (airSensor.begin(Wire, true) == false) //Se comprueba que se ha establecido la conexión I2C
  {
    Serial.println("Sensor no detectado. Por favor compruébalo");
    while (1)
      ;
  }

  //El módulo permite autocalibrarse en función de la altitud a la que deba realizar las mediciones
  Serial.print("Autocalibracion ha resultado ser ");
  if (airSensor.getAutoSelfCalibration() == true)
    Serial.println("correcta");
  else
    Serial.println("incorrecta");
  //Configuración de la conexión WiFi y creación del cliente seguro MQTT
  conectar_WiFi();
  conectar_broker_MQTT();
  //Zona 4: Programa principal

void loop() {

  //Se comprueba la conexión con el bróker para enviar los datos
  if (!mqttClient.connected()){connect();}
```

## ANEXO A

```
if (airSensor.dataAvailable()==HIGH){
    //Se recogen los datos de la medición y se crea el mensaje
    NivelCO2=LeerNivelCO2(airSensor);
    Temp=LeerTemp(airSensor);
    HumRel=LeerHum(airSensor);
    GenerarMensaje(NivelCO2, Temp, HumRel);
}
else
    delay(2000);

clienteMQTT(MQTT_Datos);

delay(1000);

mqttClient.loop();

Serial.print("Mensaje enviado: ");
Serial.println(MQTT_Datos);
```

Con la comprobación del mensaje enviado se llega al final del programa principal, a continuación, se incluyen el certificado de seguridad TLS y las funciones de la aplicación.

```
const char caCert[] PROGMEM = R"EOF(
-----BEGIN CERTIFICATE-----
MIID9TCCAt2gAwIBAgIUJDMHEXyF6WUuWYv9fTHrztntNe2MwDQYJKoZIhvcNAQEL
BQAwYkxkCzAJBgNVBAYTAKVITREwDwYDVQQIDAhBc3RlcmlhczEOMAwGA1UEBwwF
R2lqb24xZDZANBgNVBAoMB1VOSU9WSTEOMAwGA1UECwwFQ0UzSTIxFDASBgNVBAMM
CONFM0kyUm9vdENBMSAwHgYJKoZIhvcNAQkBFhFlbG9wZXpjQHVuaW92aS51czAe
Fw0yMTA1MTQwNzA4NTVaFw0zMTA1MTIwNzA4NTVaMIGJMQswCQYDVQQGEwJFUzER
MA8GA1UECAwIQXN0dXJpYkxkCzAJBgNVBACMBUdpam9uMQ8wDQYDVQQKDAZVTk1P
VkkxXjAJBgNVBAsMBUNFM0kyMRQwEgYDVQQDDAtDRlNjM1Jvb3RDQTEgMB4GCSqG
SIb3DQEJARYRZkxkCzAJBgNVBAMTMBUdpam9uMQ8wDQYDVQBAQAA4IB
DwAwggEKAoIBAQCZ4mECT81JtmJG/f0da92U1jXBhSelwnGnZO+mmP8ajpbTd+Q
e1QkEyRa10IsEPFkjhzliMeLDWx43lncFJHV5zk+DIzcn9YHPgMVakcEqQQdTodC
F3zLtyALHSv6IHB62i5vC7KC0GIh/GDo+w37xAhxXzg7l1sbR/bOS1TCyW03umSDD
NbQwzo3sy8M9KgAI0pr3nKd9rPLAILWaUBbQs0Dim2zsdpeFr1AgwkG5TaAlIggY
7fjEli+jp5WY986BkXCGCoG+gLFempbIhhr/BaDRKToe6QBFfLgRkKqNHHZLNeDS
kjSS2HA5I+KW+tmWey79Y8/BhSOaZI5o15NBAGMBAAGjUzBRMB0GA1UdDgQWBBSO
/sDMYvALUh2eJn2Wv6+zUns7ETAfBgNVHSMEGDAWgBSO/sDMYvALUh2eJn2Wv6+z
Uns7ETAPBgNVHRMBAf8EBTADAQH/MA0GCSqGSIb3DQEBCwUAA4IBAQA7cSa3ZZ+1
PK7+kdwRwIOxfj/oEId+sBW23jtNLJlpaKjtgVuRdcSOFJSqb7W7IT+PkI98WmCI
aUaI85z5miWbVHVyQu3JgCy3sEpuW3xeGzJHE5pcqyEkrmjsVEoBcqK5QgjjyOd6q
avF37IV4SPYsLw6IHiFh8rvpTRO6RZmH5Cmwv3aew8qgmwal/bIUTpev6xDawKZS
bFT93cu/+09XWBJkBN/nuXgYV08DJUrAcocw+f+8x12dg9MYR1ZLFAGuetipthH
QvYRmyT0doWuOWFTmWiS30J2Pp8ME3/KMNKNC2dWAOWcWwul2zJJPEpusTjtpFxE
4Wc8V3P14+gz
-----END CERTIFICATE-----
)EOF";

const uint8_t mqttCertFingerprint[] =
{0x99, 0x14, 0x5D, 0x64, 0x77, 0xF1, 0x4C, 0x84, 0xB4, 0x4B, 0xAE, 0x22, 0x80, 0xE7,
 0xDD, 0x86, 0xAF, 0x74, 0x9C, 0xB1};

X509List caCertX509(caCert);
WiFiClientSecure MQTT_cliente;
PubSubClient mqttClient(MQTT_cliente);
```

```
void conectar_WiFi()
{
    WiFi.begin(ssid,password);
    Serial.print("Conectando");
    while (WiFi.status() != WL_CONNECTED)
    {
        delay(500);
    }
    //Se informa cuando la conexión haya sido exitosa
    Serial.println();
    Serial.print("Conectado a direccion IP: ");
    Serial.println(WiFi.localIP());
}

void conectar_broker_MQTT()
{
    MQTT_cliente.setTrustAnchors(scaCertX509);
    MQTT_cliente.allowSelfSignedCerts();
    MQTT_cliente.setFingerprint(mqttCertFingerprint);
    verifytls();
    //Ajustes del broker MQTT
    mqttClient.setServer(IP_mqtt_server,8883);
    clientID += String(random(0xffff), HEX);
    Serial.print("MQTT-ClientID: ");
    Serial.println(clientID);
}

bool verifytls()
{
    bool success = false;
    Serial.print("Verificando las conexiones TLS con: ");
    Serial.println(IP_mqtt_server);
    success = MQTT_cliente.connect(IP_mqtt_server, 8883);
    if (success)
        Serial.println("Conexión realizada: Certificado y huella validas.");
    else
        Serial.println("La conexión ha fallado");
    return (success);
}
```

```
void connect()
{
  //Bucle hasta que tenga éxito la conexión
  while (!mqttClient.connected())
  {
    if (mqttClient.connect(clientID.c_str()))
    {
      Serial.println("Conectado");
    }
    else
    {
      Serial.print("Fallo, rc=");
      Serial.print(mqttClient.state());
      delay(5000);
    }
  }
}

uint16_t LeerNivelCO2(SCD30 airSensor){

  if (airSensor.dataAvailable())
  {
    NivelCO2=airSensor.getCO2();
  }
  return NivelCO2;
}

float LeerTemp(SCD30 airSensor){

  if (airSensor.dataAvailable())
  {
    Temp=airSensor.getTemperature();
  }
  return Temp;
}

float LeerHum(SCD30 airSensor){

  if (airSensor.dataAvailable())
  {
    HumRel=airSensor.getHumidity();
  }
  return HumRel;
}
```

## ANEXO A

---

```
void GenerarMensaje(int NivelCO2, float Temp, float HumRel){
    char string_NivelCO2[5];
    char string_Temp[5];
    char string_HumRel[5];
    sprintf(string_NivelCO2,"%05i",NivelCO2);
    MQTT_Datos[0]='\0';
    strcpy(MQTT_Datos, cabecera_MQTT);
    strcat(MQTT_Datos, string_NivelCO2);
    strcat(MQTT_Datos, " ppm, ");
    dtostrf(Temp, 5, 2, string_Temp);
    strcat(MQTT_Datos, string_Temp);
    strcat(MQTT_Datos, " °C y ");
    dtostrf(HumRel, 5, 1, string_HumRel);
    strcat(MQTT_Datos, string_HumRel);
    strcat(MQTT_Datos, " %");
}
void clienteMQTT(char* MQTT_Datos)
{
    mqttClient.publish(topic_datos, MQTT_Datos);
}
```



## Anexo B

En este segundo anexo se verá la aplicación creada para el sistema de gestión que se encarga de la recepción de los mensajes, su decodificación, el almacenamiento en una base de datos de los resultados de las mediciones y la interpretación de los datos ambientales para proporcionar la instrucción que ha de publicar en el bróker para que llegue al sistema de actuación. Nuevamente, como en el caso anterior se iniciará desde las librerías, después las variables globales y el programa principal, posteriormente, aparecerán las funciones construidas y, finalmente, la creación de la interfaz gráfica. A continuación, se mostrarán las líneas de código ya comentadas en el subapartado 7.1, además de las necesarias para la creación de la interfaz gráfica de la aplicación.

```
#ZONA: Librerías
from tkinter import font
from tkinter.font import BOLD
import paho.mqtt.client as mqtt
import time
import ssl
from tkinter import *
import random
import string
from BDD_SG import AnadirDatos
#ZONA: Variables globales y prototipos de las funciones
#Dirección IP del bróker
IP_mqtt_server="156.35.154.1"

#Creación del cliente seguro
cliente_MQTT = 'Marcos_' + ''.join(random.choice(string.ascii_uppercase + string.digits)
                                   for x in range(8))
mqttc = mqtt.Client(cliente_MQTT)
mqttc.tls_set("cacert.pem", tls_version=ssl.PROTOCOL_TLSv1)
mqttc.tls_insecure_set(True)

#Variables para las órdenes
nivelAnt=0

#"Topic" al que se suscribe
topic_Datos="TFM/datos"

#"Topic" para publicar
topic_act="TFM/orden"

#ZONA: Programa principal
#Se define la rutina Callback cuando se reciban los datos por MQTT
mqttc.on_message=RecibirDatos
```

```
#Función para la conexión con el bróker
```

```
def ConectarBroker():  
    try:  
        IP_broker=IP_mqtt_server  
        puerto=8883  
        timeout_seg=60  
        mqttc.connect(IP_broker, puerto,timeout_seg)  
        time.sleep(1)  
        mqttc.loop_start()  
        time.sleep(0.1)  
        mqttc.loop_start()  
        cuadroMensaje.insert(0, "Conectado al bróker con IP: "+IP_broker)  
        cuadroPublicacion.insert(0, "Conectado al bróker con IP: "+IP_broker)  
        cuadroMensaje.insert(0, "-----")  
        cuadroPublicacion.insert(0, "-----")  
    except:  
        cuadroMensaje.insert(0, "-----")  
        cuadroPublicacion.insert(0, "-----")  
        cuadroMensaje.insert(0, "No se ha podido conectar con el bróker")  
        cuadroPublicacion.insert(0, "No se ha podido conectar con el bróker")
```

```
#Función para la desconexión con el bróker
```

```
def DesconectarBroker():  
    time.sleep(0.1)  
    mqttc.unsubscribe(topic_Datos)  
    mqttc.disconnect()  
    mqttc.loop_stop()  
    time.sleep(1)  
    cuadroMensaje.insert(0, "-----")  
    cuadroPublicacion.insert(0, "-----")  
    cuadroMensaje.insert(0, "Desuscribiendo del topic")  
    cuadroMensaje.insert(0, "Desconectando del bróker")  
    cuadroPublicacion.insert(0, "Desuscribiendo del topic")  
    cuadroPublicacion.insert(0, "Desconectando del bróker")  
    cuadroMensaje.insert(0, "-----")  
    cuadroPublicacion.insert(0, "-----")
```

```
#Función para suscribirse al "topic"
```

```
def SuscribirTopic():  
    time.sleep(1)  
    mqttc.subscribe(topic_Datos)  
    mqttc.loop_start()  
    cuadroMensaje.insert(0, "Suscrito al topic")  
    cuadroMensaje.insert(0, "-----")
```

```
#Función para desuscribirse al "topic"
def DesuscribirTopic():
    time.sleep(0.1)
    mqttc.unsubscribe(topic_Datos)
    mqttc.loop_start()
    cuadroMensaje.insert(0, "-----")
    cuadroMensaje.insert(0, "Desuscribiendo del topic")
    cuadroMensaje.insert(0, "-----")

#Función para la recepción de los datos
def RecibirDatos (mosq, userdata, msg):
    if msg.payload != 0:
        fecha_tupla=time.localtime()
        Fecha=time.strftime("%d/%m/%Y",fecha_tupla)
        Hora=time.strftime("%H:%M:%S",fecha_tupla)
        cuadroMensaje.insert(0, "Mensaje: "+str(msg.payload.decode('utf-8')))
        SnivelCO2=str(msg.payload)
        SdatoCO2=SnivelCO2[20:26]
        global nivelCO2
        nivelCO2=int(SdatoCO2)
        Stemp=str(msg.payload)
        SdatoTemp=Stemp[31:37]
        global Temp
        Temp=float(SdatoTemp)
        Shum=str(msg.payload)
        SdatoHum=Shum[50:56]
        global HumRel
        HumRel=float(SdatoHum)
        global nivelAnt
        if nivelAnt!=nivelCO2:
            AnadirDatos(Fecha, Hora, nivelCO2, Temp, HumRel)
            orden=ordenVentilador(nivelCO2, Temp, nivelAnt)
            publicar(orden)
            time.sleep(2)
            nivelAnt=nivelCO2
        else:
            AnadirDatos(Fecha, Hora, nivelCO2, Temp, HumRel)
            time.sleep(2)
            nivelAnt=nivelCO2
    else:
        print("No había mensaje")
```

```
import sqlite3
#Función para añadir los datos en los campos
def AnadirDatos(Fecha, Hora, nivelCO2, Temp, HumRel):
    ConexionBDD()
    conexionBDD=sqlite3.connect("Aplicación")
    cursor=conexionBDD.cursor()
    datos=Fecha, Hora, nivelCO2, Temp, HumRel
    cursor.execute("INSERT INTO REGISTRO VALUES(?,?,?,?,?)", datos)
    conexionBDD.commit()
#Función para la creación y conexión con la BDD
def ConexionBDD():
    conexionBDD=sqlite3.connect("Aplicación")
    cursor=conexionBDD.cursor()
    try:
        cursor.execute('''
        #CREATE TABLE REGISTRO(
        #Fecha CHAR(10),
        #Hora CHAR(8),
        #Nivel CO2 INTEGER,
        #Temperatura DECIMAL(4,2),
        #Humedad DECIMAL(5,1))
        ''')
    except:
        print("Base de datos ya creada")
#Funciones para el sistema de gestión
def ordenVentilador(nivelCO2,Temp, nivelAnt):
    if ((nivelCO2<600) & (Temp<=60)):
        if nivelCO2<nivelAnt:
            accion=apagaVentilador()
        else:
            accion=ventiladorApagado()
    elif ((nivelCO2>=600) & (nivelCO2<=615) & (Temp<=60)):
        accion=encenderVentilador()
    elif ((nivelCO2>615) & (nivelCO2<=1200) & (Temp<=60)):
        if nivelCO2<nivelAnt:
            accion=reduceVelocidad()
        else:
            accion=aumentaVelocidad()
    elif ((nivelCO2>1200) & (Temp<=60)):
        accion=velocidadMaxima()
    else:
        cuadroMensaje.insert(0, "Está habiendo un incendio, apagando ventilador")
        DesuscribirTopic()
        DesconectarBroker()
    return accion
```

```
def ventiladorApagado():
    vApagado=("El ventilador se encuentra apagado")
    return vApagado

def apagaVentilador():
    vApagando=("Apaga el ventilador")
    return vApagando

def encenderVentilador():
    eVentilador=("Arranca el ventilador")
    return eVentilador

def reduceVelocidad():
    rVelocidad=("Reduce la velocidad de giro")
    return rVelocidad

def aumentaVelocidad():
    aVelocidad=("Aumenta la velocidad de giro")
    return aVelocidad

def velocidadMaxima():
    vMax=("Ventilador a máxima velocidad")
    return vMax
#Función para publicar la orden en el bróker
def publicar(orden):
    time.sleep(0.1)
    mqttc.publish(topic_act,orden)
    cuadroPublicacion.insert(0, "-----")
    cuadroPublicacion.insert(0, "Publicando mensaje")
    cuadroPublicacion.insert(0, "Mensaje publicado: "+orden)
```

A continuación, se mostrarán las líneas de código que se han utilizado para la creación de la interfaz gráfica de la aplicación del sistema de gestión.

```
#ZONA 4: Entorno de la aplicación
#Creación de la ventana
ventana=Tk()
ventana.title("Sistema de gestión")
ventana.iconbitmap("Logo epi.ico")
ventana.geometry('480x600')
```

```
#Creación de labels
#Suscripción a mensaje
titulo1Label=Label(ventana, text="Conectarse al bróker")
titulo1Label.grid(row=0, column=1, columnspan=2, pady=10, sticky="nesw")
titulo1Label.config(font=("Calibri", 14, "bold"))

ipBrokerLabel=Label(ventana, text="IP del bróker:")
ipBrokerLabel.grid(row=1, column=1, sticky="nesw", padx=10)
ipBrokerLabel.config(font=("Calibri", 10))

titulo2Label=Label(ventana, text="Suscribirse al topic")
titulo2Label.grid(row=3, column=0, columnspan=2, pady=10, sticky="nesw")
titulo2Label.config(font=("Calibri", 14, "bold"))

suscribirseDatos=Label(ventana, text="Topic datos:")
suscribirseDatos.grid(row=4, column=0, sticky="nesw", padx=10)
suscribirseDatos.config(font=("Calibri", 10))

titulo3Label=Label(ventana, text="Mensaje recibido")
titulo3Label.grid(row=8, column=1, columnspan=2, pady=10, sticky="nesw")
titulo3Label.config(font=("Calibri", 14, "bold"))
#Publicación de mensaje
titulo4Label=Label(ventana, text="Publicar bajo el topic")
titulo4Label.grid(row=3, column=2, columnspan=2, pady=10, sticky="nesw")
titulo4Label.config(font=("Calibri", 14, "bold"))

publicarOrdenLabel=Label(ventana, text="Topic publicar:")
publicarOrdenLabel.grid(row=4, column=2, sticky="nesw", padx=10)
publicarOrdenLabel.config(font=("Calibri", 10))

titulo5Label=Label(ventana, text="Mensaje publicado")
titulo5Label.grid(row=10, column=1, columnspan=2, pady=10, sticky="nesw")
titulo5Label.config(font=("Calibri", 14, "bold"))
```

```
#Creación de los entry
#Suscripción a mensaje
cuadroIP=StringVar()
cuadroIP.set(IP_mqtt_server)
cuadroIP=Entry(ventana, textvariable=cuadroIP)
cuadroIP.grid(row=1, column=2, sticky="nesw")

topicDatos=StringVar()
topicDatos.set(topic_Datos)
cuadroTemp=Entry(ventana, textvariable=topicDatos)
cuadroTemp.grid(row=4, column=1, sticky="nesw")

panel_informativo=Scrollbar(ventana)
cuadroMensaje=Listbox(ventana, yscrollcommand=panel_informativo.set)
cuadroMensaje.grid(row=9, column=0, columnspan=4)
cuadroMensaje.config(width=60, height=8)
panel_informativo.config(command=cuadroMensaje.yview)

#Publicación de mensaje
topicAct=StringVar()
topicAct.set(topic_act)
cuadroAct=Entry(ventana, textvariable=topicAct)
cuadroAct.grid(row=4, column=3, sticky="nesw")

panel_publicacion=Scrollbar(ventana)
cuadroPublicacion=Listbox(ventana, yscrollcommand=panel_publicacion.set)
cuadroPublicacion.grid(row=11, column=0, columnspan=4)
cuadroPublicacion.config(width=60, height=8)
panel_publicacion.config(command=cuadroPublicacion.yview)
#Creación de los botones
botonConectar=Button(ventana, text="Conectar", command=ConectarBroker)
botonConectar.grid(row=2, column=1, pady=5)

botonDesconectar=Button(ventana, text="Desconectar", command=DesconectarBroker)
botonDesconectar.grid(row=2, column=2, pady=5)

botonSuscribir=Button(ventana, text="Suscribir", command=SuscribirTopic)
botonSuscribir.grid(row=6, column=0, pady=5)

botonDesuscribir=Button(ventana, text="Cancelar suscripción", command=DesuscribirTopic)
botonDesuscribir.grid(row=6, column=1, pady=5)

ventana.mainloop()
```

