



ugr

Universidad
de Granada

TRABAJO FIN DE GRADO
INGENIERÍA EN INGENIERÍA INFORMÁTICA

Desarrollo de un sistema automatizado de gestión de conservación preventiva de obras de arte

**Sistema de monitorización para ayudar a la conservación preventiva del
patrimonio cultural**

Autor

Gadi Macías Sánchez

Director

Jesús González Peñalver



Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Granada, 30 junio de 2023

Desarrollo de un sistema automatizado de gestión de conservación preventiva de Arte: Sistema de monitorización para ayudar a la conservación preventiva del patrimonio cultural

Palabras clave: Programación de sistemas empujados, obras de arte, conservación preventiva, sensores, Arduino, MQTT, Node-RED, conservación preventiva, IoT.

Resumen

El objetivo de este proyecto es realizar el diseño y el desarrollo de crear un sistema monitorizado para alertar y prevenir el deterioro de obras de arte, ya sean de patrimonio cultural, obras que se encuentren en museos, fachadas... aunque este proyecto se centrará en salas de museo. El objetivo principal será crear un sistema monitorizado encargado de recoger información sobre el estado de los parámetros para tener en cuenta para evitar el deterioro de las obras de arte. Teniendo en cuenta los datos obtenidos se podrá tener información del estado en tiempo real de los parámetros que deterioran las obras y ayudar a la toma de decisiones.

Development of Automated Preventive Conservation Management System for Art: Monitoring System Assist in the Preventive Conservation of Cultural

Keywords: Embedded systems programming, artwork, preventive conservation, sensors, Arduino, MQTT, Node-RED, preventive conservation, IoT.

Abstract

The objective of this project is to design and develop a system to alert and prevent the deterioration of artworks, whether they are part of cultural heritage, exhibited in museums, or displayed on facades. The main goal is to create a monitored system that collects information about the parameters that need to be considered to prevent the deterioration of artworks. By considering these collected data, real-time information about the conditions that affect the artworks can be obtained, thereby aiding in decision-making processes.

Yo, **Gadi Macías Sánchez**, alumno de la titulación Ingeniería Informática de la **Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación de la Universidad de Granada**, con DNI 44064048G, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Grado en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: Gadi Macías Sánchez

Granada a 30 de junio de 2023.

D. Jesús González Valdepeñas, Profesor del alumno Gadi Macías Sánchez del Departamento Arquitectura y Tecnología de Computadores de la Universidad de Granada.

Informan:

Que el presente trabajo, titulado *Desarrollo de un sistema automatizado de gestión de conservación preventiva de arte, Sistema de monitorización para ayudar a la conservación preventiva de arte del proyecto*, ha sido realizado bajo su supervisión por **Gadi Macías Sánchez**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Granada a 30 de junio de 2023.

Los directores:

Jesús González Peñalver

Agradecimientos

A mi tutor, por ayudarme y aconsejarme en todo momento. A mis padres y mi hermana por confiar en mí, darme ánimos y ayudarme a llegar al final. A Lucía por ser mi mayor apoyo siempre que lo necesito.

ÍNDICE

Capítulo 1. Objetivos, estructura del proyecto.....	17
1.1 Objetivos	17
1.2 Motivación	18
1.3 Estructura del trabajo	18
Capítulo 2. Introducción a la conservación preventiva.....	20
2.1 Antecedentes, justificación de la conservación preventiva.....	20
2.2 Conservación preventiva.....	21
2.3 Estudio del escenario y valores meteorológicos relevantes para la conservación preventiva tenidos en cuenta para este proyecto.....	23
Capítulo 3. Especificación de los requisitos	26
3.1 Descripción de las necesidades a cubrir.....	26
3.2 Reglas de negocio.....	27
3.3 Requisitos	27
3.3.1 Requisitos funcionales	27
3.3.2 Requisitos no funcionales.....	28
3.4 Casos de uso del sistema.....	28
3.5 Metodología de diseño	33
Capítulo 4. Gestión de riesgos, planificación y presupuesto	36
4.1 Análisis de riesgos	36
4.2 Diagrama de Gantt.....	37
4.3 Planificación temporal y presupuesto.....	39
4.3.1 Presupuesto Hardware	41
4.3.2 Presupuesto Software	42
Capítulo 5. Diseño del sistema.....	43
5.1 Estudio y especificación formal	43
5.1.1 Especificación Hardware.....	43
5.1.2 Especificación Software.....	45
5.2 Lógica del sistema	49
Capítulo 6. Desarrollo del sistema	51

6.1 Software y protocolos utilizados	51
6.1.1 Software Arduino IDE	51
6.1.2 Node-RED	51
6.1.3 Protocolos de comunicación	54
6.1.3.1 Protocolo MQTT	55
6.1.4 Protocolo I2C	57
6.2 Hardware utilizado	58
6.2.1 Hardware WeMos D1 R2 Wifi	58
6.2.2 Hardware Elegoo Uno R3	59
6.2.3 Sensores utilizados	60
6.3 Esquemático del circuito de salas de museo	61
6.4 Estudio previo	64
Capítulo 7. Parte Experimental y Resultados.....	66
7.1 Pruebas y resultados	66
7.1.1 Evaluación de los sensores y conexión Wifi	66
7.1.2 Evaluación de Node-RED	70
Capítulo 8. Conclusiones	80
8.1 Valoración personal	80
8.2 Limitaciones y futuras mejoras	80
8.3 Conclusiones	81
Capítulo 9. Anexos	83
9.1 Manual de usuario de Node-RED	83
9.1.1 Instalación de Node-RED	83
9.1.2 Interfaz de Node-RED	83
9.1.3 Creación del flujo	85
9.1.4 Ejecución y depuración del flujo	85
9.1.5 Paleta de Nodos	86
9.1.6 Importación y Exportación de flujos	86
Capítulo 10. Bibliografía.....	88

ÍNDICE DE FIGURAS

Capítulo 3

Figura 3.1 Diagrama de Casos de uso del sistema.....	29
--	----

Capítulo 4

Figura 4.1 Diagrama de Gantt.....	38
-----------------------------------	----

Capítulo 5

Figura 5.1 Diagrama UML de la estructura planteada para el sistema.....	50
---	----

Capítulo 6

Figura 6.1 Node-RED editor de flujo basado en el navegador.....	53
Figura 6.2 DashBoard de Node-RED. Medidor de Temperatura, Humedad.....	54
Figura 6.3 Ejemplo de comunicación del protocolo MQTT.....	55
Figura 6.4 Esquemático 1 del circuito utilizado con Paint3D.....	62

Capítulo 7

Figura 7.1. Casos de Uso para comprobar el correcto funcionamiento de la lectura de Sensores.....	67
Figura 7.2 Casos de uso de la Verificación de la conexión Wifi.....	68
Figura 7.3 Visualización de los datos obtenidos por el puerto serie de Arduino.....	69
Figura 7.4. Casos de Uso para el correcto funcionamiento de Node-RED.....	70
Figura 7.5 Flujo del sistema en Node-RED.....	73
Figura 7.6. Datos recibidos en Node-RED.....	74
Figura 7.7 Visualización de los datos en la Interfaz de Usuario.....	75
Figura 7.8 Datos almacenados en BBDD.....	76
Figura 7.9. Flujo del sistema de alertas en Node-RED.....	77
Figura 7.10. Casos de Uso para verificar el correcto funcionamiento del sistema de Alertas.....	77
Figura 7.11. Sistema de Alertas mediante bot de Telegram.....	79

Capítulo 9

Figura 9.1 Barra de herramientas de Node-RED.....	84
Figura 9.2 Editor de flujos de Node-RED.....	84
Figura 9.3 Panel de propiedades de Node-RED.....	85
Figura 9.4 Nodos de depuración en Node-RED.....	86
Figura 9.5 Exportar un flujo en Node-RED.....	87

ÍNDICE DE TABLAS

Capítulo 3

Tabla 3.1 Explicación Caso de Uso “Iniciar Aplicación”	29
Tabla 3.2 Explicación Caso de Uso “Lectura de Sensores”	30
Tabla 3.3 Explicación Caso de Uso “Visualización de los datos obtenidos”	31
Tabla 3.4 Explicación Caso de Uso “Recibimiento de alertas”	32

Capítulo 4

Tabla 4.1 Análisis de Riesgos	36
Tabla 4.2 Fase de Estudio	39
Tabla 4.3 Fase de Desarrollo	40
Tabla 4.4 Fase de Pruebas	40
Tabla 4.5 Fase de Documentación	40
Tabla 4.6 Coste Hardware	41
Tabla 4.7 Coste Software	42

Capítulo 5

Tabla 5.1 Tabla comparativa entre las diferentes opciones de placa hardware estudiadas	44
Tabla 5.2 Tabla comparativa entre las diferentes herramientas Software estudiadas	47
Tabla 5.3 Ventajas de MySQL Workbench favorables a nuestro proyecto	47
Tabla 5.4 Tabla comparativa entre las posibles soluciones para la implementación del Sistema de Alertas	48

Capítulo 6

Tabla 6.1 Direcciones de trabajo del sensor BH1750	60
Tabla 6.2 Resolución del sensor BH1750	60

Capítulo 7

Tabla 7.1 Explicación Caso de Uso “Correcta lectura de los sensores Aplicación”	68
Tabla 7.2 Explicación Caso de Uso “Verificación de la conexión Wifi”	69
Tabla 7.3 Explicación Caso de Uso “Correcto recibimiento de los datos”	68
Tabla 7.4 Explicación Caso de Uso “Correcta visualización de los datos”	69

Capítulo 1. Objetivos, estructura del proyecto.

1.1 Objetivos

El objetivo final del proyecto es el desarrollo de un prototipo de un sistema monitorizado capaz de medir los valores para la correcta conservación de obras de arte y patrimonio cultural. Para ello, se van a simular dos salas de un museo. Donde en cada sala habrá distintos tipos de obras de arte dependiendo del material del que estén contruidos, ya que, a la hora de crear un sistema monitorizado para la conversación de obras de arte influye el tipo de material para las condiciones meteorológicas que debe cumplir la sala donde se almacenen. Dependiendo de los datos obtenidos de los valores medidos, se generarán una serie de señales que avisarán en caso de superar los umbrales previamente definidos, para evitar así daños en las obras.

Para llegar al objetivo final hemos marcado una serie de objetivos que desarrollándolos hemos podido finalizar este proyecto.

- Investigar sobre la conservación preventiva de arte, sobre todo de los valores que se deben de tener en cuenta para la correcta monitorización del sistema.
- Analizar los diferentes dispositivos hardware y software que se pueden utilizar para realizar este proyecto.
- Investigar sobre los diferentes protocolos de comunicación que se pueden implementar en función de los dispositivos seleccionados.
- Saber administrar una base de datos para poder almacenar los datos leídos de los diferentes sensores.
- Aprender a integrar las diferentes herramientas software y hardware que se utilicen en este proyecto.
- Desarrollar un sistema software de visualización de los datos leídos para poder ayudar a tener las condiciones meteorológicas controladas.
- Desarrollar un sistema de alertas para avisar al personal responsable de alguna anomalía en el sistema monitorizado como pueden ser cambios en los valores meteorológicos leídos o alertas por un mal funcionamiento.

1.2 Motivación

La motivación detrás de este proyecto surge de la pasión por el arte, que ha tenido el alumno durante años, combinada con la experiencia en prácticas extracurriculares en la empresa NTT DATA. Durante estas prácticas, se tuvo la oportunidad de trabajar en proyectos que involucraban la extracción y manipulación de datos de una red de sensores. Fue entonces cuando surgió la idea de combinar la informática, específicamente el desarrollo de sistemas empujados para crear sistemas monitorizados con el arte, concretamente la conservación preventiva para llevar a cabo este proyecto.

Otra de las razones principales que impulsaron al alumno a realizar este proyecto fue la escasa información disponible en Internet sobre cómo se monitorizan actualmente estos sistemas en los museos de arte. En el pasado, los museos solían utilizar registradores de datos autónomos para medir y registrar las condiciones ambientales. Estos dispositivos se programaban para tomar mediciones periódicas y almacenar los datos en su memoria interna. Con el auge de la tecnología IoT (Internet of Things), al alumno le pareció interesante investigar sobre cómo configurar una red IoT para ofrecer otra solución a estos sistemas de monitorización.

Además, surgió la idea de innovar en los sistemas de alerta dedicados a los avisos en este tipo de sistemas. Hasta el momento, no se han encontrado sistemas implementados que innoven utilizando redes sociales, como pueden ser un bot en alguna red social. Esto proporcionaría ventajas como la facilidad de uso para el personal y la utilización de software libre sin ningún costo adicional.

1.3 Estructura del trabajo

Para llevar a cabo este trabajo se han analizado consecutivamente todos los pasos que se han seguido en su diseño y desarrollo. Para que cualquier persona que esté leyendo esto sea capaz de entender el trabajo realizado. El trabajo se divide en los siguientes capítulos:

En el Capítulo 1, se establecen los objetivos del trabajo y se presenta la motivación que impulsa el proyecto. Además, se describe la estructura del trabajo y cómo se desarrollará cada capítulo.

En el Capítulo 2, se introduce el concepto de conservación preventiva, justificando su importancia y analizando antecedentes relacionados. También se estudia el escenario y se identifican los valores meteorológicos relevantes para la conservación preventiva que

se tendrán en cuenta en el proyecto.

El Capítulo 3 se centra en la especificación de los requisitos del sistema. Se describen las necesidades que se pretenden cubrir, se establecen las reglas de negocio y se detallan los requisitos funcionales y no funcionales del sistema. Además, se presentan los casos de uso y se explica la metodología de diseño utilizada.

En el Capítulo 4, se aborda la gestión de riesgos, la planificación y el presupuesto del proyecto. Se realiza un análisis de riesgos, se elabora un diagrama de Gantt para planificar las tareas y se establece un presupuesto tanto para el hardware como para el software necesario.

El Capítulo 5 se dedica al diseño del sistema, donde se lleva a cabo un estudio y una especificación formal. Se detallan las especificaciones tanto del hardware como del software, y se analiza la lógica del sistema.

En el Capítulo 6, se describe el desarrollo del sistema, incluyendo el software y los protocolos utilizados. Se mencionan las herramientas como Arduino IDE y Node-RED, así como los protocolos de comunicación empleados. También se presenta el hardware utilizado, los sensores y se muestra un esquemático del circuito de las salas del museo.

El Capítulo 7 se dedica a la parte experimental y los resultados obtenidos. Se realizan pruebas y se evalúan los sensores, la conexión Wi-Fi y el funcionamiento de Node-RED.

En el Capítulo 8, se presentan las conclusiones del trabajo, incluyendo una valoración personal, las limitaciones encontradas y las futuras mejoras propuestas.

El Capítulo 9 contiene los anexos, donde se encuentra el manual de usuario de Node-RED, que detalla la instalación, la interfaz, la creación y depuración de flujos, así como la importación y exportación de flujos.

Finalmente, en el Capítulo 10 se lista la bibliografía utilizada en el trabajo.

Capítulo 2. Introducción a la conservación preventiva

2.1 Antecedentes, justificación de la conservación preventiva

Según Manfred Koller, “Desde la antigüedad, y más específicamente desde la época medieval, el renacimiento o el barroco, se aplicaban prácticas para la conservación de edificios, pinturas murales, esculturas y pinturas con un enfoque de prevención del deterioro.” (1994: 274).^[1] Se trataba de garantizar la salvación de la materia que lo compone y su perdurabilidad a través del tiempo. Antiguamente y durante siglos sólo se restauraba y casi no se conservaba ningún tipo de patrimonio, por lo que los propios artistas eran los encargados de la restauración en obras de arte. Según Herráez y Rodríguez “Dos aspectos han favorecido especialmente a una evolución en la conservación del patrimonio histórico: el respeto a la autenticidad o integridad de las obras, y la incorporación de las ciencias experimentales y utilización del método científico en las intervenciones.” (1999: 1).^[2] Durante el siglo XVIII se presenta un cambio en el modo de entender la conservación del patrimonio histórico. Las consecuencias de este cambio de pensamiento fueron los importantes descubrimientos arqueológicos como Pompeya, Palatino de Roma y Egipto. Debido a estos descubrimientos aumentaron las colecciones de los museos, ya que, se llenaron de excavaciones arqueológicas y con graves problemas de conservación. Otro factor que hay que tener en cuenta fue la creciente contaminación provocada por el proceso de industrialización y desarrollo urbano. Los altos valores de humedad, iluminación artificial, aire contaminado perjudicaron el patrimonio cultural.

Durante el siglo XIX los problemas de conservación pasaron a intentar buscar soluciones que pudiese resolver la aplicación de las ciencias experimentales. No es hasta 1905, principios del siglo XX cuando se crea el Laboratorio del Museo Británico y el Laboratorio del museo de Berlín, dos acontecimientos fundamentales para la historia de la conservación preventiva de arte no obstante no es hasta casi mitad del siglo XX cuando

los principios de la conservación preventiva no se empiezan a institucionalizar.

Actualmente, existe personal cualificado en conservación y restauración. Empresas que se encargan de la conservación preventiva y de implementar sistemas de monitorización que se adecuen a los diferentes clientes, como pueden ser, museos, exposiciones al aire libre... Con el paso de los años, el uso de las tecnologías se ha integrado casi que, en cualquier ámbito, por lo que no iba a ser menos la implementación en el estudio de la conservación preventiva del patrimonio cultural.

2.2 Conservación preventiva

Como se ha mencionado en el apartado anterior antiguamente el concepto de conservación preventiva era algo que prácticamente era irrelevante ya que las herramientas eran escasas para poder mantener las condiciones meteorológicas necesarias, por lo que para poder hablar de la conservación preventiva antes del siglo XX se debe analizar más el concepto de restauración, ya que anterior a este siglo, como se ha comentado, los métodos de conservación eran escasos debido a que se estaban empezando a investigar métodos científicos y experimentales que favorecieran a la conservación del patrimonio. Hasta entonces eran los propios artistas los encargados de restaurar y “conservar” el patrimonio cultural. Al no tenerse en cuenta el concepto de conservación como se conoce hoy día cabe de esperar que las conservaciones de las obras artísticas tampoco se llevaran a cabo de manera exitosa conservando la integridad de éstas como ocurre en la actualidad, por ende, cabe mencionar que otras muchas obras que podrían considerarse menores en valor histórico ni se tenían en cuenta para ser conservadas en perfecto estado. Puesto que, era el mismo autor de la obra el encargado de conservar la autenticidad de ésta, la única medida que se tomaba en caso de deterioración de las obras era la restauración.

No fue hasta el siglo XX cuando la conservación se formaliza como una acción reconocida en Instituciones cuando se habla de preservar el patrimonio cultural que hasta ahora se conoce.

Antes de hablar de la creación y evolución de la conservación preventiva cabría hacer un inciso para hablar sobre la evolución del concepto de “patrimonio cultural”. Se trata de un concepto bastante joven, ya que antiguamente cuando se hablaba de obras de arte no se hacía uso de esta palabra para englobar así a los objetos que se conceden con un valor y un significado articular y distintivo. El problema que traía esta definición es que se trata de un concepto relativo, que se fue construyendo mediante un proceso de

atribución de valores que estaban influenciadas por la historia, entre otras influencias. En definitiva, según Josué Lluïl Peñalba

La idea de patrimonio ha ido evolucionando a lo largo de los siglos desde un planteamiento particularista, centrado en la propiedad privada y el disfrute individual, hacia una creciente difusión de los monumentos y las obras de arte como ejemplos modélicos de la cultura nacional y símbolos de la identidad colectiva. (Lluïl, 2005: 180).^[3]

En cuanto a la evolución de la conservación preventiva en España, no es también hasta el siglo XX cuando se imponen una serie de criterios donde la conservación preventiva pasa a tener mucha más repercusión que la restauración. Es decir, los objetivos perseguidos por la restauración desde entonces y hasta la actualidad pasan a “devolver a la obra su legibilidad desde un punto de vista estético, pero salvaguardando todos sus valores documentales genuinos intactos, sin eliminar generalmente aportaciones de otras épocas por el testimonio que suponen del quehacer humano” (Macarrón, 2021: 192) ^[4]. Fue en este siglo cuando se tienen en cuenta los principios éticos cuando se habla de restauración-conservación. En ellos se habla de “anteponer la conservación a la restauración”.

En 1961 se crea en España el Instituto Central de Conservación y Restauración con la finalidad de “dotar al país de un Organismo que con las garantías científicas y técnicas necesarias pueda atender a la tarea de restauración y conservación de las obras y objetos.” (Ruiz, 1995: 38).^[5] Lo que esto significó que las reformas del sistema para la protección del patrimonio cultural histórico español deben ser comprendidas ahora con respecto al propio Instituto de Conservación y Restauración (ICROA). Al principio, esta institución funcionaba paralelamente a otras como la encargada de restaurar libros y documentos (Centro Nacional de Restauración de Libros y Documentos) entre otras. No es hasta 1971 cuando el ICROA centraliza las tareas de Conservación y Restauración de los bienes muebles pertenecientes al Patrimonio Histórico Español.

En este mismo siglo, en 1977 se crea el Ministerio de Cultura y el ICROA pasa a llamarse Instituto de Conservación y Restauración de Bienes Culturales en 1985 y pasa a constituirse como “un órgano moderno de la administración del Estado especializado en la protección, conservación y restauración de los Bienes Culturales y agrupando los organismos hasta ahora dispersos especializados en la Conservación del Patrimonio” (Ruiz, 1995: 39)^[6] donde se incluyen el ICROA, el CECOMI (Centro de Conservación y Microfilmación Documental y Bibliográfica), Subdirecciones Generales de Restauración de Monumentos y Arqueología.

2.3 Estudio del escenario y valores meteorológicos relevantes para la conservación preventiva tenidos en cuenta para este proyecto

En este apartado, se describirán los escenarios a tener en cuenta durante la realización del proyecto. Para este proyecto debido al presupuesto, y el tiempo del que se dispone para montar todo se ha decantado por implementar la siguiente lógica dentro del proyecto.

Se va a monitorizar dos salas diferentes de un museo, teniendo en cuenta los distintos valores meteorológicos que deben tener cada sala. Esto dependerá del patrimonio cultural que haya en su interior.

Para ello, según el Instituto de Conservación y Restauración de Bienes Culturales los parámetros para tener en cuenta a la hora de hacer mediciones para monitorizar salas serían:

-Humedad Relativa (HR): Se deben de tener en cuenta unos límites mínimos y máximos de humedad relativa ya que el patrimonio cultural expuesto a humedades extremas puede deteriorarse y dañarse con facilidad, corriendo el riesgo de que la obra se pierda por completo. Para ello, “el límite mínimo: tomando como referencia la respuesta de los materiales orgánicos ante las oscilaciones de la HR del aire se puede establecer un límite mínimo de un 45 por 100 (45%) de HR. El límite máximo: Está condicionado por el favorecimiento de la proliferación de microorganismos y otros organismos biodegradadores a determinados niveles de HR, por lo que este límite debe fijarse en el 65 por 100 (65%).” (Herráez & Rodríguez, 1989: 9).^[7]

También es importante tener en cuenta las oscilaciones por lo que se define “un rango óptimo para la conservación de la mayoría de los objetos y colecciones, ya sea en condiciones de exposición o de almacenamiento, puede ser 55 ± 5 por 100 de HR en un rango de temperatura de 18 ± 2 °C, con un límite máximo de oscilaciones diarias de ± 3 por 100.” (Herráez & Rodríguez, 1989: 9)^[8].

-Temperatura: En cuanto a la temperatura del aire en el interior de un museo, en condiciones normales, es muy importante tener en cuenta que también que las temperaturas deben de acomodarse al confort del personal del museo y de los visitantes, por lo que “un rango óptimo aconsejado por distintos especialistas es de 18 ± 2 °C, con una fluctuación diaria máxima de 1,5 °C.” (Herráez & Rodríguez, 1989: 9)^[9].

-Iluminación: La iluminación debe de acomodarse a unos niveles en los que se puedan

apreciar los objetos del interior de la sala. Sin embargo, se debe tener en cuenta el poder que tiene la luz para la degradación del patrimonio cultural, por lo que, el Instituto de Conservación y Restauración de Bienes Culturales diferencia entre tres tipos diferentes de valores de iluminación dependiendo de la sensibilidad del material del patrimonio cultural. “Para tratar de minimizar este tipo de deterioro se han adoptado unos niveles de iluminación de referencia basados en estudios científicos, para exhibición de objetos y colecciones.” (Herráez & Rodríguez, 1989: 12) ^[10].

-Sensibilidad alta: Esta sensibilidad se debe de tener en cuenta para objetos especialmente vulnerables como acuarelas, tejidos, materiales teñidos, pigmentos procedentes de sustancias animales o vegetales, grabados en color, dibujos, fotografías en color, pergaminos, colecciones de ciencias naturales. Para ello el máximo de lux debe de ser de 50 lx.

-Sensibilidad media: Para objetos de sensibilidad media como grabados en blanco y negro, fotografías y material de archivo, materiales orgánicos no pintados, policromías, pinturas al óleo y acrílicas, materiales pintados y lacados, marfil. La iluminación medida en lux no debe de superar el umbral 150-200 lux.

-Sensibilidad baja: Objetos como cerámicas, porcelana, vidrio. La iluminación máxima debe de ser de 300 lux.

También se tienen en cuenta otro tipo de objetos insensibles a la luz como los constituidos por piedra o metales ya que, “no sufren fotodegradación, aunque los efectos térmicos ocasionados por la radiación IR y la acción de la radiación UV pueden afectarles.” (Michalski, 1987: 104) ^[11]

-Contaminación: El tema de los contaminantes del aire requiere tenerse en cuenta en sitios donde se conserven las obras de arte. En este caso es necesario una buena ventilación para eliminar el aire mediante filtros físicos. Para ello, se deben de tener en cuenta contaminantes gaseosos como CO, CO₂, SO₂, NO₂, O₃, etc. Una vez detectados deberá de usarse un filtro de carbono activo o realizar un lavado del aire.

En cuanto al edificio y su equipamiento influyen las condiciones ambientales en el interior. Es importante que el edificio esté dotado para soportar climas extremos, con un buen aislamiento adecuado, para así poder mantener la temperatura y HR necesaria dentro de los límites aconsejados. Para la iluminación, es aconsejable que las salas de exhibición carezcan de iluminación natural. Si esto no fuera posible, el Instituto de Conservación y Restauración de Bienes Culturales aconseja que “las ventanas y lucernarios deberán

dotarse de filtros UV y vidrios que limiten la iluminación transmitida.” (Herráez & Rodríguez, 1989: 13) ^[12].

Capítulo 3. Especificación de los requisitos

3.1 Descripción de las necesidades a cubrir

La necesidad más básica por cubrir ha sido tener que investigar para poder encontrar la manera de poder realizarlo de la manera óptima y teniendo en cuenta los recursos disponibles. No ha sido una tarea fácil y se ha tenido que investigar y aprender conceptos de otras materias como la conservación preventiva de arte para así poder realizar un proyecto correcto y adecuado a las necesidades reales de los museos.

En el estudio de la descripción de necesidades se ha tenido en cuenta que antes de iniciar la fase de desarrollo debemos de analizar con detalles que componentes usar y qué software instalar para su posterior uso, por lo que también ha sido necesario investigar sobre diferentes formas de implementar este proyecto.

En cuanto a las necesidades que se quieren cubrir para poder hacer las respectivas mediciones meteorológicas y climáticas de los espacios dónde se encuentran las obras tenemos la necesidad de tener controlado los diferentes parámetros meteorológicos que pueden llegar a deteriorar los bienes artísticos. Por otro lado, también se quiere enviar un aviso al personal responsable cuando se superen alguno de los umbrales establecidos para los diferentes parámetros estudiados. Por lo que, las necesidades a satisfaceres se resumen en:

- Obtener información actualizada de los diferentes parámetros meteorológicos y mostrarlos en tiempo real en un software especializado para ello, como puede ser una interfaz web.
- Enviar un aviso al dispositivo supervisado por el personal responsable.

A la hora de crear la interfaz web, se debe cumplir facilitar al personal responsable la interacción con el servicio y para ello debe satisfacerse lo siguiente:

- Diseñar una interfaz sencilla y accesible para mejorar las experiencias de usuario.

- Mostrar todos los valores en tiempo real que puedan ser relevantes para mantener el sistema monitorizado actualizado, como pueden ser la hora y fecha y los diferentes valores meteorológicos.
- Diseñar una interfaz legible con un formato adecuado. Se debe utilizar una combinación de colores que facilite la lectura

3.2 Reglas de negocio

Según las necesidades establecidas se deben plantear unas normas básicas que definan los diferentes aspectos del diseño y desarrollo del sistema:

RN-01: En la implementación y pruebas del proyecto, tanto en la parte software como hardware se utilizarán recursos libres de derecho de autor.

RN-02: Los responsables podrán acceder a toda la información sobre los valores recogidos por los diferentes sensores.

RN-03: El uso del servicio será gratuito para todos los responsables de este.

3.3 Requisitos

3.3.1 Requisitos funcionales

En el desarrollo de cualquier sistema o aplicación es esencial establecer un conjunto claro de los requisitos funcionales que definan las funcionalidades y características que el producto final debe cumplir. Éstos describen cómo debe comportarse el sistema, qué acciones debe realizar y qué resultados debe producir. A continuación, se presentarán detalladamente los requisitos funcionales que han sido identificados para este proyecto, brindando una visión clara y precisa de los objetivos que se buscan alcanzar:

1. Obtener un diseño válido que funcione.
2. Construir un circuito electrónico que conecte los sensores, con los dispositivos hardware que se utilizarán para su correcta lectura.
3. Construir un sistema autónomo que sea capaz de tomar decisiones y realizar los ajustes automáticos en función de los datos y las condiciones meteorológicas detectadas.
4. Poder generar valores de los sensores a partir del circuito construido.

5. Poder almacenar estos valores por si posteriormente pudiesen ser útiles.
6. Mostrar los valores de los sensores generados para su posterior interpretación.
7. Detectar anomalías en el ambiente que generen alertas para el personal responsable.

3.3.2 Requisitos no funcionales

Estos requisitos se centran en aspectos claves como la usabilidad, funcionalidad y otros atributos que influyen en la calidad y experiencia del usuario. En este apartado, se detallarán los requisitos no funcionales identificados para este proyecto para cumplir con los objetivos y las necesidades del usuario.

1. Sistema autónomo: El sistema debe ser capaz de operar de manera independiente, sin intervención constante de usuarios o administradores.
2. Sistema escalable: El sistema debe de ser capaz de manejar un aumento en la cantidad de datos y usuarios sin degradar el rendimiento.
3. Bajo coste: El sistema debe de tener un coste bajo optimizando los recursos y utilizar eficientemente los recursos disponibles durante todo el ciclo de vida del proyecto y la operación del sistema.
4. Seguridad y fiabilidad: El sistema debe proteger los datos meteorológicos y la privacidad de los usuarios.
5. Estabilidad y funcionabilidad: El sistema debe de ser estable y fácil de mantener en funcionamiento.
6. Rendimiento: El sistema debe ser capaz de procesar y mostrar los datos meteorológicos en tiempo real sin retrasos significativos.

3.4 Casos de uso del sistema

Lo primero que se ha realizado una vez que se han tenido todos estos conceptos claros ha sido un esquema general de los componentes necesarios y su función en el sistema.

Material y actores involucrados en el sistema

1. Sistema Hardware: Las placas hardware deberán de ser las encargadas de leer los diferentes sensores y mandar por sus correspondientes protocolos los datos para su posterior manipulación en el servidor.

2. Servidor: El servidor leerá los valores recibidos de los sensores y realizará las funciones necesarias para acabar almacenando los datos y a su vez mostrarlos en la interfaz gráfica.
3. Base de datos: Se almacenarán los datos recibidos desde el servidor para tener un control de los valores almacenados con sus correspondientes fechas de lectura.
4. Personal responsable del correcto funcionamiento del sistema: El personal del museo será el responsable de recibir las alertas causadas por las anomalías del sistema ya sea un fallo en los componentes hardware o un fallo en algún otro punto del sistema. Si se detectan anomalías se enviarán avisos a través de la API al personal responsable.

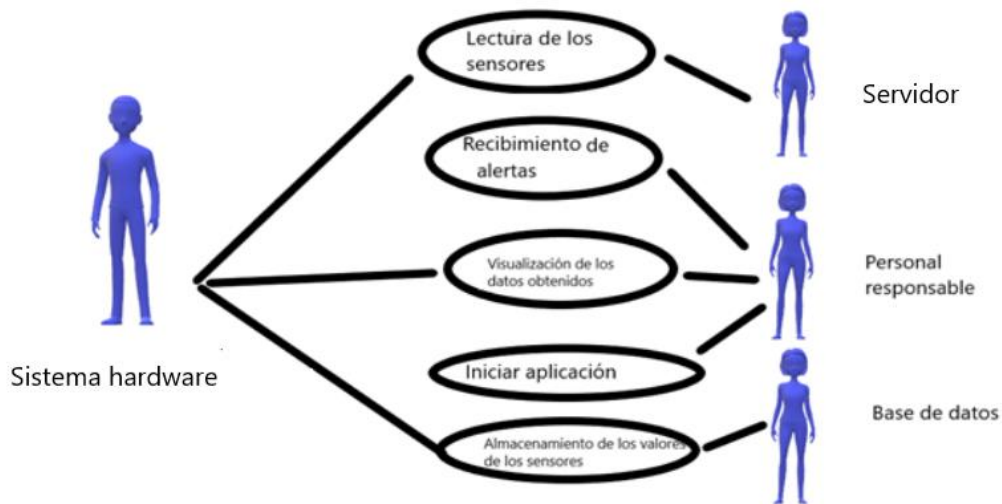


Figura 3.1 Diagrama de Casos de uso del sistema

A continuación, describiremos detalladamente los diferentes casos de uso del sistema, proporcionando un análisis detallado de cada uno de ellos. Durante este proceso, exploraremos meticulosamente las múltiples formas en las que el sistema puede ser utilizado, examinando sus aplicaciones prácticas en distintos escenarios y contextos.

Nombre del caso	Iniciar aplicación
Actores	1. Personal responsable

Dependencias	-
Propósito	Leer los diferentes valores que proporcionan los sensores para tener medidas fiables dentro del sistema.
Resumen	El personal responsable debe iniciar la aplicación para poder realizar el proceso de visualización de los datos obtenidos.
Curso normal básico	<ol style="list-style-type: none"> 1. Se leen los valores 2. El personal responsable inicia la aplicación
Cursos alternativos	La aplicación da fallos y se repite la acción hasta que se inicia
Observaciones	-

Tabla 3.1 Explicación Caso de Uso “Iniciar Aplicación”

Nombre del caso	Lectura de los sensores
Actores	<ol style="list-style-type: none"> 1. Sistema hardware 2. Servidor
Dependencias	CU-Iniciar aplicación
Propósito	Leer los valores captados de los sensores
Resumen	Una vez que se hayan configurado e inicializado los sensores, deben de leerse los valores producidos por estos.
Curso normal básico	<ol style="list-style-type: none"> 1. Configurar e inicializar los sensores. 2. Leer estos valores para su posterior envío.
Cursos alternativos	En caso de fallo en la lectura de los sensores generar alertas.
Observaciones	-

Tabla 3.2 Explicación Caso de Uso “Lectura de los sensores”

Nombre del caso	Visualización de los datos obtenidos
Actores	<ol style="list-style-type: none"> 1. Sistema hardware 2. Personal responsable
Dependencias	CU-Iniciar aplicación, CU-Lectura de sensores
Propósito	Conseguir visualizar los datos leídos de los sensores.
Resumen	El usuario al iniciar la aplicación desde el navegador visualizará la interfaz de usuario donde se encuentran gráficas que muestran los diferentes valores de los sensores.
Curso normal básico	<ol style="list-style-type: none"> 1. El usuario inicia la aplicación desde el navegador. 2. La placa hardware envía los datos captados por los sensores 3. Se generan los datos y se visualizan en la interfaz de usuario.
Curso alternativo	Se generan alertas si los datos no se llegan a visualizar
Observaciones	-

Tabla 3.3 Explicación Caso de Uso “Visualización de los datos obtenidos”

Nombre del caso	Almacenamiento de los valores de los sensores
Actores	<ol style="list-style-type: none"> 1. Base de datos (BBDD) 2. Sistema hardware 3. Servidor
Dependencias	CU-Iniciar aplicación
Propósito	Almacenar los datos en BBDD.
Resumen	Una vez que se hayan leído los valores captados de los sensores se debe procesar

	hasta completar su correcto almacenamiento en la base de datos.
Curso normal básico	<ol style="list-style-type: none"> 1. Recibir los diferentes valores de los sensores que llegan desde Arduino. 2. Procesar estos valores de manera que estén aptos para ser almacenados 3. Realizar un almacenamiento desde el servidor hacia la BBDD.
Curso alternativo	El almacenamiento da fallo y se envía una alerta
Observaciones	El sistema debe de almacenar siempre los valores leídos.

Tabla 3.4 Explicación Caso de Uso “Almacenamiento de los valores de los sensores”

Nombre del caso	Recibimiento de alertas
Actores	<ol style="list-style-type: none"> 1. Personal responsable 2. Servidor
Dependencias	CU-Iniciar aplicacion, CU-Lectura de Sensores, CU- Almacenamiento de los valores de los sensores
Propósito	Generar alertas cuando se produzcan anomalías
Resumen	Una vez almacenados los datos se deben de procesar por una lógica de alertas del Sistema para así poder captar anomalías
Curso normal básico	<ol style="list-style-type: none"> 1. Una vez almacenados los datos se analizan de manera que se tienen en cuenta si superan o no los umbrales establecidos. 2. Si se superan los umbrales

	<p>establecidos se generan alertas que se enviarán a través de la API.</p> <p>3. El personal responsable recibirá un mensaje alertando sobre la/s anomalía/s.</p>
Cursos alternativos	No se produce ninguna anomalía en la última lectura.
Observaciones	-

Tabla 3.5 Explicación Caso de Uso “Recibimiento de alertas”

3.5 Metodología de diseño

Durante el proceso de selección de la metodología de diseño para este proyecto, se ha llevado a cabo un estudio exhaustivo de las diversas metodologías ampliamente utilizadas por las empresas en proyectos similares. Se han considerado metodologías como el Desarrollo Rápido de Aplicaciones (RAD) y las metodologías en cascada. El objetivo de este análisis ha sido identificar la metodología más adecuada que se ajuste a las necesidades específicas de este proyecto, teniendo en cuenta factores como la eficiencia en la entrega, la gestión de riesgos y la capacidad de adaptación en los requisitos.

Las metodologías de diseño rápido de aplicaciones, como el enfoque RAD, se caracterizan por su énfasis en la entrega rápida de prototipos y la participación de los usuarios en el proceso de desarrollo. Si bien estas metodologías pueden ser efectivas en muchos casos, en nuestro proyecto específico pueden presentar desafíos adicionales.

En primer lugar, la entrega de prototipos rápidos puede no ser adecuada para nuestro proyecto, ya que requiere una planificación cuidadosa y una validación rigurosa de los requisitos antes de proceder a la implementación. Debido a la complejidad y la naturaleza crítica del proyecto, es esencial garantizar una base sólida y una comprensión clara de los requisitos antes de avanzar en el desarrollo.

Además, las metodologías RAD a menudo implican una mayor participación de los usuarios en el proceso de desarrollo. Si bien la retroalimentación de los usuarios es valiosa, en nuestro proyecto puede ser más adecuado contar con una participación más limitada de los usuarios y dejar la toma de decisiones clave en manos de expertos técnicos y profesionales del dominio. Esto garantizará una mayor coherencia y un enfoque más

eficiente para alcanzar los objetivos del proyecto.

Por otro lado, las metodologías en cascada, que siguen una secuencia lineal y ordenada de etapas de desarrollo, pueden presentar limitaciones en un proyecto como el nuestro. Dado que los requisitos pueden evolucionar y cambiar a medida que se avanza en el desarrollo, es importante contar con un enfoque más flexible que permita la adaptación y la incorporación de cambios a lo largo del tiempo. Las metodologías en cascada pueden resultar inflexibles en este sentido, lo que podría llevar a dificultades para abordar los cambios en los requisitos y, en última instancia, afectar negativamente la calidad y la eficiencia del proyecto.

En resumen, aunque las metodologías de diseño rápido de aplicaciones y en cascada son ampliamente utilizadas en otro tipo de proyectos para nuestro proyecto específico se requiere un enfoque más estructurado y adaptable. Necesitamos garantizar una planificación y validación rigurosas de los requisitos, minimizando la participación excesiva de los usuarios y permitiendo una mayor flexibilidad para adaptarse a los cambios. De esta manera, podremos gestionar de manera eficaz los riesgos y alcanzar los objetivos del proyecto de manera satisfactoria.

Finalmente, tras este exhaustivo estudio sobre las diferentes metodologías se ha decantado por seguir una metodología ágil Scrum ya que este tipo de metodologías se crean como respuesta a las limitaciones de las técnicas de gestión estructuradas tradicionalmente como pueden ser las metodologías en cascada. A continuación, se exponen una serie de ventajas que favorecen al proyecto presente:

- Las metodologías Ágil como Scrum permiten adaptar la forma de trabajo a las condiciones del proyecto, por ende, se consigue flexibilidad e inmediatez para poder amoldar el desarrollo a circunstancias que vayan surgiendo.
- El poder revisar y adaptar el desarrollo software y hardware de nuestro sistema permite tener un mayor control sobre el trabajo, por lo que se reducen los tiempos y costes del proyecto.
- En estos métodos el diseño e implementación del proyecto son las actividades centrales del proceso el resto se van integrando en torno a estas.
- Este tipo de metodologías se enfocan en la calidad y mejora continua del proyecto ya que debido a las revisiones constantes son flexibles a cambio. Para nuestro sistema este tipo de metodologías permitirá identificar y corregir

problemas a medida que surgen, mejorando la calidad del sistema a lo largo del tiempo.

A continuación, se detallan los Sprint desarrollados, coste en tiempos y una breve descripción de su contenido:

1. 1º sprint (3 semanas): Análisis y Diseño del sistema. Se han analizado los requisitos del cliente, aplicaciones similares del mercado y trabajos que habían sido realizados por otras empresas para realizar las mediciones de los valores meteorológicos óptimos que ayuden a la conservación preventiva de las obras de arte en museos. Posteriormente se ha creado y diseñado los diferentes diagramas UML para el desarrollo del sistema.
2. 2º sprint (3 semanas): Desarrollo del sistema hardware. Se montó el circuito necesario para la interconexión entre las placas hardware y los sensores. Una vez hecho esto se creó el código que leyera los valores emitidos por los sensores.
3. 3º sprint (3 semanas): Desarrollo del sistema software. Se montó el código y flujo del sistema que permitiese una correcta visualización en la Interfaz de usuario y un correcto almacenado de los valores en base de datos.
4. 4º sprint (2 semanas): Pruebas y corrección de errores. Se ha procedido a realizar pruebas y correcciones del sistema desarrollado.
5. 5º sprint (1 semana): Despliegue del sistema y puesta en producción. Se ha generado la documentación necesaria para su puesta en marcha en los servidores y su integración con el código desarrollado.

Capítulo 4. Gestión de riesgos, planificación y presupuesto

4.1 Análisis de riesgos

En esta sección se presentan los diferentes tipos de riesgos que surgieron durante la ejecución del proyecto, así como las soluciones propuestas para abordarlos y la probabilidad estimada de que ocurran.

Tipo de Riesgo	Probabilidad de aparición	Soluciones
Errores en el diseño	Bajo	Rehacer el diseño. Investigar más para obtener más información.
Fallos en el montaje del circuito	Medio	Cambiar disposición de componentes hardware (cables, sensores...)
Fallos en el hardware	Medio	Sustituir componentes hardware (cables,sensores...)
Fallos de programación software	Alto	Hacer debug para dar con los problemas. Verificar cada proceso aislado.
Software malicioso	Bajo	Cortafuegos, antivirus.

Tabla 4.1 Análisis de Riesgos

4.2 Diagrama de Gantt

En esta sección, se presentará un diagrama de Gantt que proporcionará una representación visual detallada de la progresión temporal de las diversas etapas del proyecto, desde su inicio hasta el final. El objetivo principal de este diagrama es mostrar de manera clara y concisa la secuencia de tareas y su duración estimada, lo que facilita la coordinación y el seguimiento del proyecto. Cada etapa se representará como una barra horizontal en el diagrama, y su longitud corresponderá a su duración prevista en horas. El siguiente diagrama también será acompañado de una línea de tiempo vertical que muestra los períodos de tiempo en intervalos regulares, lo que permitirá una mejor comprensión de la duración de cada etapa en relación con el tiempo total del proyecto.

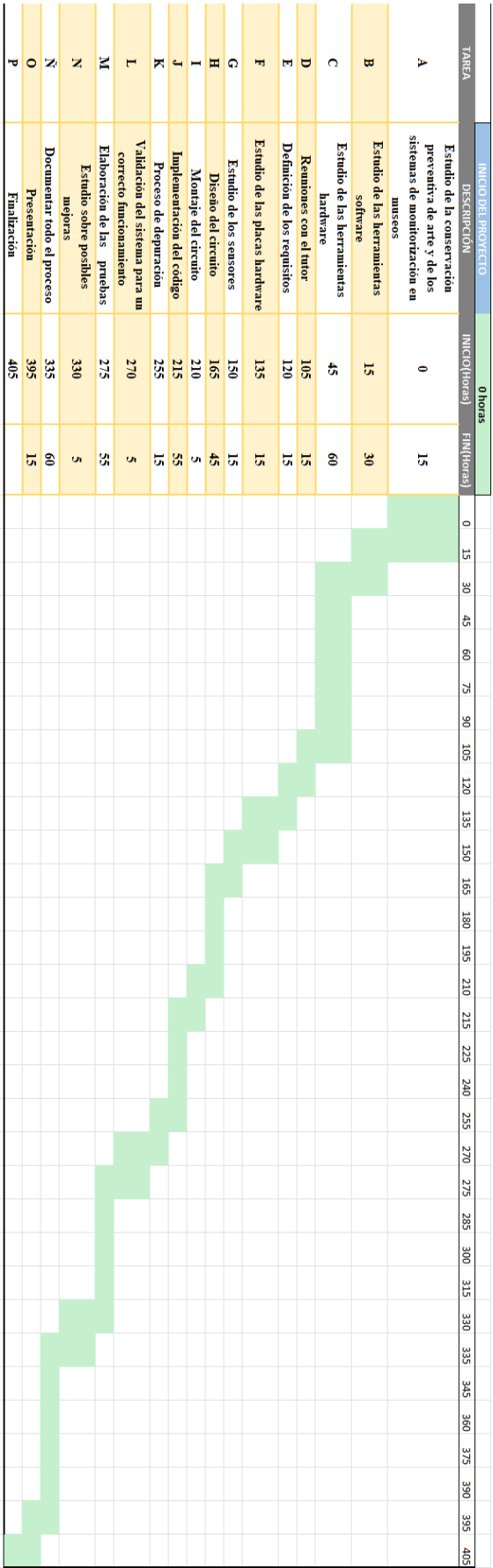


Figura 4.1 Diagrama de Gantt

4.3 Planificación temporal y presupuesto

Una vez asignado el tutor de este proyecto y la idea de éste en octubre de 2022 se comienza a investigar e indagar sobre ideas, hacer estudios preliminares sobre la viabilidad para así poder descartar y modificar propuestas hasta que finalmente se llega a la idea adecuada ajustada a los objetivos deseados sobre febrero de 2023.

La fecha para realizar la entrega fue en la convocatoria de junio de 2023. Teniendo 4 meses para así entregar un prototipo del sistema final que cumpla todos los requisitos propuestos.

El Trabajo de Fin de Grado consta de 12 créditos, equivalente a aproximadamente 25-20 horas de trabajo por crédito. En total en torno a las 360 horas para dedicarlas a la elaboración del proyecto, redacción de la memoria y muestra de pruebas finales. A continuación, basado en el total de horas se describe un análisis temporal de la realización del proyecto por etapas:

1.Fase de Estudio

<i>Fases de Estudio</i>	<i>Horas</i>
<i>Investigación sobre Conservación Preventiva y Sistemas de Monitorización en Museos.</i>	15
<i>Estudio de las herramientas software</i>	35
<i>Estudio de las herramientas hardware</i>	50
<i>Reuniones con el tutor</i>	7
<i>Total</i>	107

Tabla 4.2 Fase de Estudio

2. Fase de Desarrollo

<i>Fases de Desarrollo</i>	<i>Horas</i>
<i>Definición de los requisitos.</i>	15
<i>Análisis</i>	35
<i>Diseño</i>	40

<i>Montaje del circuito</i>	8
<i>Implementación del proyecto</i>	70
<i>Depuración del proyecto</i>	6
<i>Total</i>	174

Tabla 4.3 Fase de Desarrollo

3. Fase de Pruebas

<i>Fases de Pruebas</i>	Horas
<i>Pruebas Software</i>	50
<i>Pruebas Hardware</i>	20
<i>Depuración de errores</i>	10
<i>Validación</i>	4
<i>Posibles mejoras</i>	6
<i>Total</i>	90

Tabla 4.4 Fase de Pruebas

4. Fase de Documentación

<i>Fases de Documentación</i>	Horas
<i>Elaboración de la memoria</i>	60
<i>Grabación de video</i>	3
<i>Presentación</i>	10
<i>Total</i>	73

Tabla 4.5 Fase de Documentación

El tiempo total que se ha empleado en la elaboración del proyecto ha sido de 405 horas.

A continuación, presentaremos una estimación del coste total del proyecto. Se analizarán los aspectos adicionales necesarios para obtener un producto final. Se proporcionará una descripción detallada del presupuesto para el desarrollo de software y hardware, incluyendo el costo por hora de trabajo realizado.

Además, se presentará un presupuesto para el lanzamiento del producto final, que incluirá las etapas de diseño, implementación y desarrollo, fabricación, distribución, marketing y lanzamiento al mercado.

4.3.1 Presupuesto Hardware

Este presupuesto está en base a las placas seleccionadas, indicando su precio por unidad y el coste total (en euros) de los diferentes productos utilizados.

Hardware	Número de unidades	Precio por unidad(euros)	Coste total (euros)
Placa ELEGOO UNO R3	1	19,99	19,99
Placa Wemos D1 R2	1	11,99	11,99
Cable USB	1	4,99	4,99
Cable microUSB	1	4,90	4,90
Sensor DHT22	2	2,65	5,3
Sensor BH1750	2	3,05	6,10
Sensor MQ-135	2	4,35	8,7
Protoboard	1	4,99	4,99
Cables	25	1,50	1,50
Total			68,46

Tabla 4.6 Coste Hardware

El presupuesto detalla los componentes seleccionados, incluyendo el número de unidades, el precio por unidad y el costo total en euros de los diferentes productos utilizados. Se ha considerado la adquisición de placas como Elego Uno R3 y la Wemos D1 R2, junto con cables USB y microUSB, así como sensores como el DHT22, BH1750 y MQ-135. También se incluyen una protoboard y cables adicionales. En resumen, el costo total del presupuesto asciende a 68,46 euros.

4.3.2 Presupuesto Software

A este presupuesto Software se incluye la mano de obra para poder obtener el sistema final.

Concepto	Número de horas	Coste por hora (euros)	Coste total (euros)
Mano de obra:			
Arduino IDE	10	20	200
Node-RED	25	20	500
Montaje y aprendizaje	30	20	600
Total	65	60	1300

Tabla 4.7 Coste por Software

En conclusión, haber producido este sistema de monitorización supone un coste de 1300 euros. Es importante mencionar que este costo no se considera una pérdida, ya que el trabajo realizado podría haber generado beneficios si se hubiera desarrollado para terceros.

Capítulo 5. Diseño del sistema

5.1 Estudio y especificación formal

Este apartado trata sobre el proceso del estudio de las diferentes opciones estudiadas. Primero, se debe tener en cuenta qué tipos de placas hardware de desarrollo utilizar para posteriormente elegir los demás elementos hardware que sean compatibles, puesto que será primordial en el desarrollo del proyecto.

5.1.1 Especificación Hardware

Dada la limitación del presupuesto y acorde a tener que hacer mediciones simulando dos salas de museo, para la realización de este proyecto necesitaremos dos placas hardware. Por tanto, cada placa debe contar con:

- Entradas y salidas digitales/analógicas suficientes para los diferentes sensores
- Capacidad para enviar los datos recolectados al sistema
- Microcontrolador con dimensiones adecuadas
- Memoria y capacidad de procesamiento suficientes para la implementación del sistema diseñado.
- Sistema de alimentación adecuado para el sistema diseñado
- Facilidad de programación y soporte

Acorde a los medios y presupuesto que tenemos disponible se han estudiado los posibles usos de la placa hardware Arduino Uno, Elegoo Uno R3, WeMos D1 Wifi, Arduino Mega 2560, Arduino Mega ADK. Para que el trabajo sea más completo y se puedan estudiar diferentes formas de diseñar el mismo sistema se elegirán dos placas distintas. En este trabajo se analizarán los modelos Arduino Uno, Elegoo uno R3, WeMos D1 Wifi y Arduino Mega 2560 ya que para las necesidades que se abarcan cumplen con las prestaciones necesarias.

Características	Arduino Uno	Elegoo Uno R3	Wemos D1 Wifi	Arduino Mega 2560
Entradas/Salidas digitales/analógicas	26 GPIO	14 GPIO	11 GPIO	54 digitales, 16 analógicas
Compatibilidad con sensores	✓	✓	✓	✓
Capacidad de enviar datos al sistema	✓	✓	✓	✓
Sistema de alimentación adecuado	5V	5V	3.3V/5V	5V
Memoria y capacidad de procesamiento	2 KB RAM, 32 MB Flash	2 KB RAM, 32 KB Flash	4 MB Flash	8 KB RAM, 256 KB Flash
Facilidad de instalación de librerías	✓	✓	✓	✓
Precio	29,28 ¹	35,00 ²	4,90 ³	48,40 ⁴

Tabla 5.1 Tabla comparativa para las diferentes opciones de placas hardware estudiadas

Finalmente, se ha llegado a la conclusión de que una de las placas utilizadas sea Elegoo Uno R3, ya que el alumno aparte de tener esta placa también incluía cables y sensores que serán útiles para la realización de las pruebas y el producto final de este proyecto. Además, se ha demostrado que encaja con las prestaciones necesarias para este proyecto. La segunda placa que utilizaremos para este proyecto será la WeMos D1 Wifi, ya que, aparte de cumplir con las prestaciones necesarias para este proyecto consta de un

¹ URL con el precio: <https://www.amazon.es/Arduino-UNO-A000066-microcontrolador-ATmega328/dp/B008GRTSV6>

² URL del KIT de iniciación: <https://www.elegoo.com/en-es/products/elegoo-uno-project-super-starter-kit>

³ URL con el precio: <https://tienda.bricogeek.com/arduino-compatibles/1218-wemos-d1-wifi-esp8266.html>

⁴ URL: <https://store-usa.arduino.cc/products/arduino-mega-2560-rev3>

módulo wifi con el que podremos simular un escenario más real de un sistema monitorizado. Las elecciones de estas placas aparte del precio más asequible que tienen son la cantidad suficientes de pines para el proyecto presente y una capacidad de procesamiento y memoria suficientes. Finalmente, se ha llegado a esta conclusión porque una de las placas puede usar una comunicación serie como sería el uso de la placa Elegoo Uno R3, y otra de las placas usará una comunicación Wifi, por lo que, se abarcarán más soluciones posibles y reales a este proyecto. Cada placa monitorizará una sala de museo.

Una vez tomada la decisión de que placas usar se debe hacer un estudio sobre los diferentes sensores a utilizar. Como se ha visto en apartados anteriores los valores para tener en cuenta para este tipo de estudios son los siguientes aquí mencionados:

-Temperatura y humedad: Para cubrir las necesidades de temperatura y humedad se dispondrá de un sensor DHT22 de temperatura y humedad. Ya que, aparte de tener los requerimientos necesarios este sensor nos permite obtener información de la temperatura como de la humedad relativa. Se estudió la posibilidad de usar el sensor DHT11 pero ofrecía mediciones de menores calidades por un precio similar.

-Luminosidad: Para medir los valores de luminosidad en el patrimonio cultural se necesita un sensor con un rango de 50-200 lux. Para ello, se dispone de un sensor LDR de Arduino, y un BH1750. Se ha elegido el sensor BH1750, ya que a diferencia de los LDR éste es capaz de proporcionar la medición en lux, medida más acorde a las necesidades de este proyecto. En cuanto a calidad el sensor BH1750 ofrece mediciones de calidad que se ajusta a nuestro presupuesto.

-Contaminación: Para evitar la contaminación en los museos o salas donde se expongan las obras de arte se debe de mantener con una humedad del 70-80%. También se hace uso de sensores que miden los niveles de CO2. Para ello se dispone de sensores MQ-7 y MQ135. Se utilizará el sensor MQ-135 ya que aparte de medir el nivel de CO2 en una sala es capaz de detectar compuestos químicos como el amoníaco y los niveles de humo. Se ha escogido el MQ135 por ser más sensible y ser más utilizado para medir la calidad del aire en salas y oficinas.

En conclusión, ajustándose al presupuesto del que se dispone se utilizarán los componentes hardware mencionados arriba para integrar en el sistema monitorizado.

5.1.2 Especificación Software

Dentro de las especificaciones software debemos de tener en cuenta que tendremos que utilizar softwares que sean compatibles con los elementos hardware que se han

estudiado en el apartado anterior. También se deben de tener en cuenta las siguientes especificaciones para realizar un sistema software de calidad para el usuario.

- Interfaz de usuario simple que ofrezca las funcionalidades descritas
- Framework adecuado para el diseño y funcionamiento de medición y almacenamiento de datos.
- Software suficiente para gestión de datos
- Conectividad entre la base de datos y el software utilizado para la medición de los valores
- Lenguaje de programación adecuado, con librerías específicas que faciliten la comunicación entre el software y hardware utilizados.
- Protocolos de comunicación para poder enviar los valores de los sensores a las placas hardware seleccionadas.
- Implementar medidas adecuadas para proteger la comunicación entre las placas seleccionadas y el software.

Uno de los principales motivos por los que se ha decantado el alumno por realizar este proyecto ha sido intentar innovar sobre los sistemas monitorizados frente a los que ya existían para la conservación preventiva. Indagando sobre los sistemas monitorizados actuales se han estudiado las posibles maneras de implementar nuestro diseño de manera innovadora. A continuación, se expondrán las diferentes alternativas y se justificará por la solución aportada. Antes de realizar una comparación sobre las posibles herramientas utilizadas para este proyecto debemos de definir las diferentes herramientas que nos harán falta en función de los requisitos expuestos anteriormente y el sistema a implementar.

Lo primero de todo necesitaremos una aplicación software donde implementar la lógica del proyecto. Es decir, donde se incluirá todo el código fuente que implementará el sistema. Después de realizar un estudio sobre las diferentes herramientas software de entornos de programación más utilizadas para este tipo de proyectos se ha decidido comparar las aplicaciones más comunes para la realización del código fuente de este tipo de proyectos como puede ser: Eclipse y Visual Studio Code frente a Node-RED. Esta última herramienta mencionada se descubrió en una asignatura del curso actual, siendo una herramienta muy utilizada recientemente para proyectos de este estilo. También debemos de tener en cuenta que deberá de usarse el software Arduino IDE para almacenar el código fuente que gestionará los diferentes sensores.

Características	Eclipse	Visual Studio Code	Node-RED
Enfoque	IDE (Entorno de desarrollo integrado)	Editor de código fuente	Específico para IoT, plataforma de programación visual
Comunidad y soporte	X	✓	✓
Programación Visual	X	X	✓
Curva de aprendizaje	Mayor curva de aprendizaje	Mayor curva de aprendizaje	Curva de aprendizaje más leve
Lenguajes admitidos	Java, C/C++, Python, PHP y más	Java, C/C++, Python, PHP y más	JavaScript, JSON
Compatibilidad	Multiplataforma (Windows, macOS, Linux)	Multiplataforma (Windows, macOS, Linux)	Multiplataforma (Windows, macOS, Linux)

Tabla 5.2 Tabla comparativa entre las diferentes herramientas Software estudiadas

Lo siguiente que debemos de comparar será el sistema donde se almacenarán los datos leídos de los sensores como puede ser una aplicación de gestión de bases de datos del tipo MySQL WorkBench o MariaDB. El alumno consta de conocimientos sobre MySQL Workbench por lo que, será el sistema utilizado ya que consta de las siguientes características compatibles con los requisitos del proyecto:

Características	MySQL Workbench
Licencia	Comercial y gratuita (en 47omplet Community Edition)
Soporte de bases de datos	MySQL
Interfaz de usuario	Interfaz gráfica 47omplete y robusta
Comunidad y soporte	Gran comunidad y soporte activo

Tabla 5.3 Ventajas de MySQL Workbench favorables a nuestro proyecto

Por último, debemos de analizar qué software utilizar para el sistema de alertas sabiendo qué uno de los motivos de la realización de este proyecto era la innovación con redes sociales para este tipo de sistemas de alertas.

Se pretende realizar una comparación entre diferentes redes sociales que puedan servir para crear este tipo de alertas. Se han comparado las aplicaciones Twitter y Telegram para la realización de este sistema de alertas.

Características	Bot de Telegram	Bot de Twitter
Acceso y alcance	A través de la aplicación Telegram	A través de la aplicación Twitter
Funcionalidades	Envío de mensajes, comandos, multimedia, etc.	Publicación de Tweets, seguimiento de hashtags, retweets, etc.
Interacción con los usuarios	Conversaciones directas con los usuarios	Respuestas y menciones a usuarios
Integración con Servicios	Integración con servicios de terceros mediante API	Integración con servicios de terceros mediante API

Tabla 5.4 Tabla comparativa entre las posibles soluciones para la implementación del Sistema de Alertas

Después de realizar este estudio de las diferentes herramientas que se pueden llegar a utilizar se ha decidido decantarse por Node-RED debido a que tiene un enfoque específico para aplicaciones IoT proporcionando una interfaz de usuario fácil de usar.

Aparte de esto, Node-RED tiene todos los requisitos necesarios para la realización de nuestro proyecto porque ofrece una interfaz donde se pueden integrar las demás aplicaciones software como pueden ser Arduino IDE, MySQL Workbench. Indagando en esta aplicación se ha llegado a la conclusión de que será todo un reto para el alumno aprender a utilizar este tipo de aplicación ya que, se trata de una programación basada en flujo. Siendo este tipo de programación algo nuevo para el alumno.

En cuanto al sistema de alertas teniendo en cuenta el estudio realizado sobre las diferentes redes sociales posibles a utilizar la aplicación de Telegram beneficiará más a nuestro sistema final ya que se trata de una conversación directa con el responsable del museo mientras que en Twitter las notificaciones serían menos directas, teniendo que acceder al usuario de Twitter para saber qué anomalías han podido producirse en el sistema.

5.2 Lógica del sistema

A continuación, en la figura 5.1 se presenta un diagrama UML que ofrece una representación visual de los distintos caminos y estados que pueden experimentar el sistema a lo largo de su ciclo de vida. Este diagrama tiene como objetivo proporcionar una comprensión clara y concisa de las diferentes etapas que presenta el sistema durante la ejecución del proyecto. Los caminos representan las acciones, decisiones y eventos específicos que pueden tener lugar en cada etapa. Al seguir estos caminos, se establece un flujo de trabajo que guía las actividades que deben llevarse a cabo en cada etapa del sistema.

Como podemos observar en la Figura 5.1 el sistema consta de 5 etapas diferenciadas *“Inicio del sistema”, “Lectura de Sensores”, “Comunicación con Node-RED”, “Visualización en Dashboard de los valores recibidos”, “Almacenamiento de los datos en BBDD”, “Aviso al personal mediante el sistema de alertas”*. Estas etapas aseguran un correcto funcionamiento del sistema desde el Inicio del sistema hasta el aviso al personal mediante el sistema de alertas cuando se produjesen.

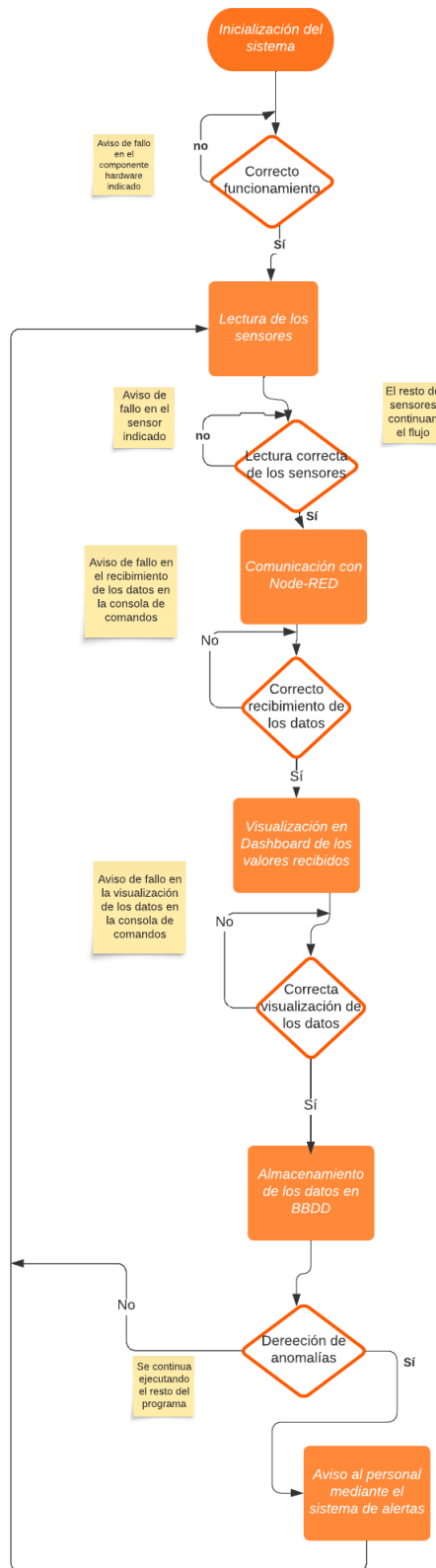


Figura 5.1 Esquema de la estructura planteada para el sistema

Capítulo 6. Desarrollo del sistema

6.1 Software y protocolos utilizados

6.1.1 Software Arduino IDE

Arduino tiene una plataforma de código abierto para la creación de prototipos basados en hardware y software de código abierto. El entorno de programación para Arduino es multiplataforma, por lo que se puede instalar y ejecutar tanto en Windows, Mac OS y Linux.

El entorno de programación de Arduino es un programa informático que se le conoce como IDE (Integrated Development Environment o Entorno de Desarrollo Integrado). Consta de un editor de código con lenguaje JAVA, aunque admite los lenguajes C y C++ utilizando reglas especiales de estructuración de códigos. También consta con un compilador, depurador y un constructor de interfaz gráfica. El IDE se instala en el ordenador y sobre él se escriben los programas para que el microcontrolador de Arduino los ejecute. ^[13]

La versatilidad y la infinidad de posibilidades que ofrece Arduino hace que sea una de las herramientas de programación más completas actualmente en el mercado. Por lo que, es uno de los motivos para el uso de ésta en este proyecto.

6.1.2 Node-RED

Node-RED es una herramienta de programación visual, definida también como un editor de flujo basado en el navegador. Está basado en la programación basada en flujos, por lo que permite desarrollar aplicaciones y automatizaciones de manera visual. En lugar de tener código tradicional, se utilizan nodos interconectados en un flujo gráfico para definir la lógica y el comportamiento de la aplicación. ^[14]

Node-RED se ha consolidado como framework open-source para la gestión y transformación de datos en tiempo real en entornos de Industria, IOT, Marketing Digital

o sistemas de Inteligencia Artificial entre otros. Esta herramienta de programación conecta varios dispositivos a la vez, tanto de hardware como de servicios de Internet. Es muy útil para los equipos dedicados a trabajo industrial o como prueba de soluciones para equipos de planta que necesiten comunicarse entre sí. Es un motor de flujos que cuenta con un enfoque IoT, y esta condición permite definir flujos de servicios a través de protocolos como MQTT. [15]

La principal ventaja de Node-RED es que al estar basado en una programación que está basada en flujos facilita la programación, ya que presta una buena representación visual mediante sus nodos del programa haciendo que el usuario tenga más facilidad a la hora de conocer el flujo del programa y por ende, su funcionamiento. Esta herramienta facilita la programación, pero es imprescindible tener una base de programación para poder interconectar los nodos, darle funcionalidad a través del lenguaje JavaScript (lenguaje de programación ampliamente utilizado para el desarrollo de aplicaciones web y aplicaciones relacionadas con servidores) y permitir un flujo adecuado del programa. Node-RED está basado en JavaScript.

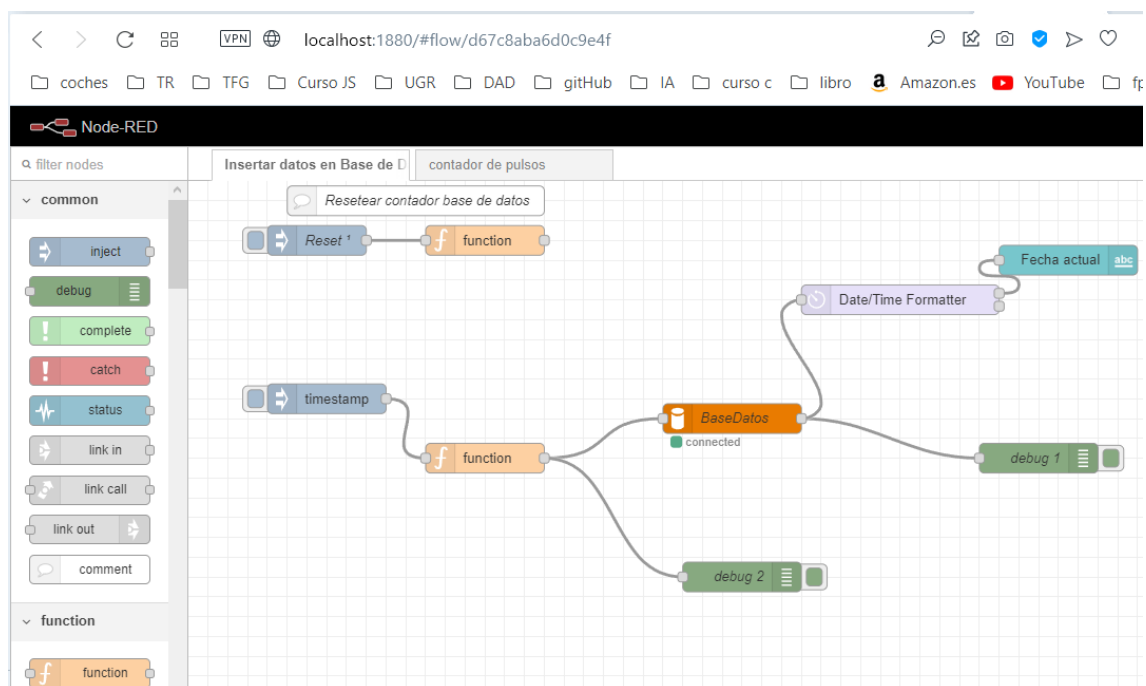


Figura 6.1 Node-RED editor de flujo basado en el navegador

Otra de las ventajas de Node-RED es que está construido en Node.js de manera que se destaca su “modelo no-bloqueante” y su “orientación a eventos asíncronos”. Gracias a estas características de Node.js el runtime de Node-RED es muy ligero con un bajo consumo de recursos, constando, además, de una gran capacidad para poder ejecutarse en los dispositivos y sistemas operativos en los que se pueda correr Node.js

Cabe destacar también como ventaja que Node-RED puede desplegarse en dispositivos de bajo coste como podría ser Raspberry Pi pudiendo ser utilizado en dispositivos actuadores o sensores en una red IOT. Por otro lado, Node-RED también tiene la capacidad de correr localmente en ordenadores y servidores locales con sistemas operativos como “Windows”, “Linux” o “MAC”. Esto permite una distribución de la capacidad de tratamiento de datos y eventos a través de la red.

Como se comentó en el apartado anterior, otro de los motivos por los cuales se ha decidido usar Node-RED es por hacer uso de sus nodos “Dashboard” ya que, como su nombre indica, este nodo nos permite crear un dashboard desde el flujo en Node-RED. Para poder visualizar esta Dashboard basta con introducir el siguiente URL: http://IP_NodeRED:1880/ui cuando tengamos los diferentes nodo configurados como se explicará más adelante. Este tipo de nodos simplificará la visualización de los diferentes valores medidos de los sensores. [16]

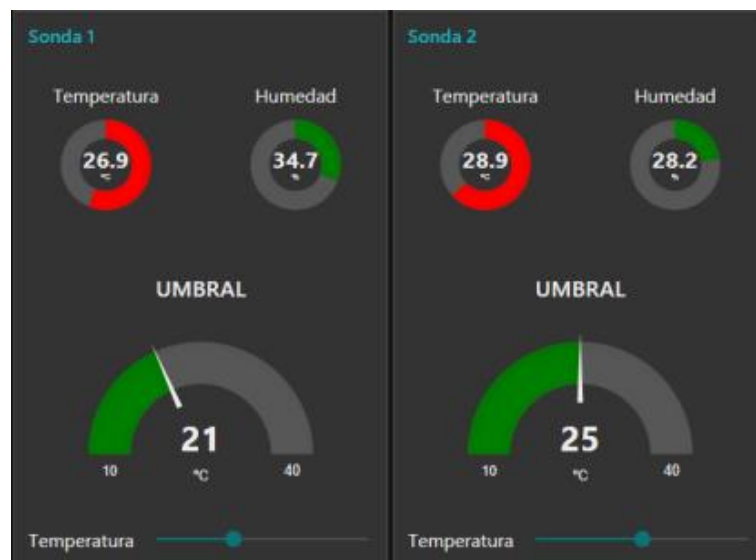


Figura 6.2 DashBoard de Node-RED. Medidor de Temperatura, Humedad

6.1.3 Protocolos de comunicación

A la hora de decantarse por un protocolo de comunicación, se han tenido en cuenta los protocolos MQTT y HTTP. Tras un periodo de información sobre ambos protocolos se ha decantado por el uso de MQTT ya que, en comparación con HTTP, MQTT es uno de los principales pilares en IoT gracias a su sencillez, ligereza y una serie de ventajas. A continuación, se mencionan algunas ventajas a la hora de utilizar MQTT frente a HTTP para IoT:

-Eficiencia y ancho de banda: MQTT es un protocolo ligero y eficiente diseñado específicamente para redes de baja potencia y ancho de banda limitado. Utiliza un modelo de suscripción/publishing, lo que esto significa que los dispositivos pueden enviar y recibir mensajes de forma asíncrona, evitando la necesidad de realizar solicitudes y respuestas constantemente. En comparación, HTTP establece una conexión para cada solicitud, lo que puede generar una mayor sobrecarga en términos de ancho de banda y recursos.

-Consumo de energía: MQTT es eficiente en cuanto al consumo de energía debido al bajo ancho de banda. Esto hace que para dispositivos IoT con recursos limitados, como sensores. HTTP, por otro lado, requiere una conexión establecida para cada solicitud, lo que puede resultar un consumo de energía más alto.

-Escalabilidad: MQTT es altamente escalable y puede manejar grandes volúmenes de mensajes y dispositivos conectados. El modelo mencionado antes de suscripción/publicación está basado en un bróker centralizado, lo que facilita la distribución de mensajes a múltiples suscriptores. En cambio, HTTP se basa en solicitudes y respuestas directas entre el cliente y el servidor, lo que puede generar cuellos de botella y limitaciones en términos de escalabilidad en entornos con gran cantidad de dispositivos conectados.

-Conexiones persistentes: MQTT permite mantener conexiones persistentes entre los clientes y el bróker, lo que proporciona una comunicación bidireccional eficiente y en tiempo real. HTTP, en cambio, sigue un modelo de solicitud/respuesta, lo que puede introducir retrasos haciendo que la comunicación en tiempo real se vea perjudicada.

6.1.3.1 Protocolo MQTT

El protocolo MQTT en este proyecto se utilizará para enviar los valores de los sensores desde Arduino IDE al servidor de Node-RED para así posteriormente poder manejar los datos.

MQTT son las siglas MQ Telemetry Transport. Es un protocolo de comunicación M2M (machine-to-machine). Está basado en la pila TCP/IP como base para la comunicación. Es un servicio de mensajería con patrón publicador/suscriptor (pub-sub) donde los dispositivos o aplicaciones se comunican a través de un intermediario centralizado llamado bróker. [17]

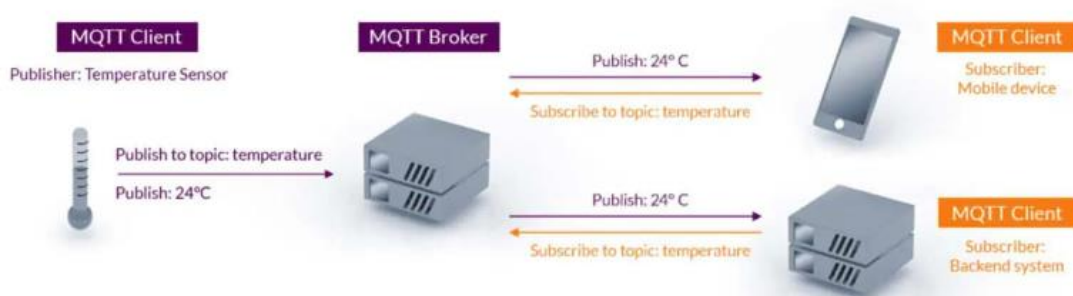


Figura 6.3 Ejemplo de comunicación del protocolo MQTT

MQTT consta de dos entidades funcionales: Broker MQTT y el cliente MQTT. Cualquier dispositivo o aplicación en una red IoT puede convertirse en un cliente MQTT. Los clientes publican o reciben mensajes a través del Broker. El bróker es el sistema back-end que coordina los mensajes entre los diferentes clientes. Las responsabilidades del bróker incluyen recibir y filtrar mensajes, identificar a los clientes suscritos a cada canal y reenviar los mensajes.

Para este proyecto se ha decidido utilizar Mosquitto (Broker MQTT), es un software de código abierto y gratuito que actúa como bróker entre los clientes de MQTT. Mosquitto es ligero y compatible con todo tipo de dispositivos, desde computadoras hasta servidores. [18]

MQTT minimiza sus transmisiones con un tamaño de mensaje pequeño, pero bien definido. Cada mensaje tiene un encabezado fijo de 2 bytes. La carga útil del mensaje es de 256MB. Lo que hace, que sea más ligero y útil para este tipo de proyectos.

También cabe mencionar los tres niveles diferentes de calidad de servicio (QoS) que implementa el protocolo, ya que permite a los diseñadores de redes elegir entre minimizar

la transmisión de datos y maximizar la fiabilidad. Dependiendo de la fiabilidad o la cantidad de banda de ancha y recursos de los que dispongamos se pueden distinguir los siguientes niveles:

-QoS 0: En este nivel, no hay garantía de la entrega del mensaje. El mensaje se envía una sola vez desde el publicador al bróker MQTT sin confirmación de entrega o reintentos. Ofrece menor fiabilidad, pero también eficiencia en términos de ancho de banda y recursos.

-QoS 1: En este nivel, el mensaje se entregará al menos una vez al suscriptor. El publicador envía el mensaje al bróker MQTT, que lo almacena temporalmente y se encarga de retransmitirlo al suscriptor. Si el suscriptor no confirma la recepción del mensaje, el bróker lo reenviará hasta que se confirme la entrega. Esto asegura que el mensaje se entregue al menos una vez, pero puede suponer duplicación de mensajes si hay reintentos.

-QoS 2: En este nivel, el mensaje se entregará una vez y también se garantiza que el mensaje se ha entregado con éxito. Al igual que en el nivel QoS 1, el mensaje se almacena temporalmente en el bróker MQTT. En este nivel, a diferencia del nivel QoS 1, se utiliza un proceso adicional de intercambio de confirmaciones entre el publicador, el bróker y el suscriptor para evitar la duplicación de mensajes. Esto asegura la entrega única y sin duplicación, aunque introduce una mayor sobrecarga y latencia en la comunicación.

Teniendo en cuenta la importancia de los valores meteorológicos a medir en este proyecto el nivel de seguridad será QoS 2.

A la hora de publicar o reenviar mensajes (en el caso del Broker) se debe de tener en cuenta los “topic”. Los topic son estructuras en una jerarquía que utilizan el carácter barra (/) como separador. Esta estructura se asemeja a la organización de un árbol de directorios en un sistema de archivos en un ordenador. Por ejemplo, como se ve en la Figura 4, el topic sería “temperature”, en el caso de que tuviésemos más sensores podríamos definir más topic jerárquicamente, por ejemplo, podríamos tener la siguiente estructura: “DatosSensores/temperatura”. Esto permite a un suscriptor especificar que solo desea recibir datos de clientes que publiquen valores relacionados con la temperatura. Alternativamente, dentro del topic “DatosSensores” podemos tener más subtemas como por ejemplo “DatosSensores/Humedad”. Los temas se pueden crear automáticamente si no existiesen a la hora de enviar mensajes desde el cliente al bróker.

MQTT también permite nombres de usuario y contraseñas para que un cliente establezca la conexión con el bróker. El problema es que, para mantener la claridad del

protocolo, los nombres de usuario y las contraseñas se transmiten en texto sin cifrar. Por otra parte, MQTT para solventar esto permite la implementación de los protocolos de seguridad SSL/TLS. Lamentablemente, esto añade una sobrecarga sustancial a las comunicaciones, pero asegura la fiabilidad de los datos.

6.1.4 Protocolo I2C

El sensor BH1750 utiliza el protocolo I2C para establecer la comunicación con las placas hardware, ya que el sensor BH1750 está diseñado con dos pines específicos, SCL (línea de reloj) y SDA (línea de datos), que son propios de este protocolo. Gracias a esta compatibilidad, la comunicación entre el sensor BH1750 y las placas hardware se puede realizar de manera eficiente y confiable utilizando el protocolo I2C. A continuación se describe el protocolo utilizado por estos sensores.

“El protocolo I2C es un estándar de comunicación serie que permite la transferencia de datos entre dispositivos electrónicos. Fue desarrollado por Philips Semiconductor (ahora NXP Semiconductors) en la década de 1980 y se ha convertido en uno de los protocolos más ampliamente utilizados en la industria.” (González, 2017: 203) ^[18]

Este protocolo utiliza dos líneas de comunicación: la línea de reloj (SCL) y la línea de datos (SDA). Estas líneas son bidireccionales y se comparten entre todos los dispositivos conectados en un bus I2C.

El dispositivo maestro, que generalmente es un microcontrolador, controla la comunicación en el bus I2C. El maestro inicia y detiene las transmisiones de datos y actúa como el control central del sistema. Los dispositivos esclavos, como sensores o periféricos, responden a las solicitudes del maestro y proporcionan los datos requeridos. La comunicación en el bus I2C se basa en el principio de direccionamiento. Cada dispositivo esclavo en el bus tiene una dirección única que se utiliza para identificarlo durante la comunicación. El maestro envía una señal de inicio en el bus seguida de la dirección del dispositivo al que desea comunicarse. Si el dispositivo esclavo tiene la dirección correcta, responde al maestro y se establece la comunicación entre ellos.

Durante la transferencia de datos, el maestro genera pulsos de reloj en la línea SCL para sincronizar la transmisión. En cada pulso de reloj, el estado del bit de datos en la línea SDA es leído o escrito. La transmisión de datos se realiza en paquetes de 8 bits, con el bit más significativo (MSB) transmitido primero.

El protocolo I2C también admite operaciones de lectura y escritura. En una operación de

escritura, el maestro envía datos al dispositivo esclavo, mientras que, en una operación de lectura, el maestro solicita datos al dispositivo esclavo. Para cambiar entre la escritura y la lectura, se utiliza un bit de dirección adicional llamado bit R/W (Read/Write) que se coloca inmediatamente después de la dirección del dispositivo.

Este protocolo también permite la conexión de múltiples dispositivos en el mismo bus, utilizando direcciones únicas para cada dispositivo. Esto facilita la expansión del sistema mediante la adición de nuevos dispositivos sin requerir un gran número de pines de E/S en el microcontrolador.

En resumen, el protocolo I2C es una interfaz de comunicación serial ampliamente utilizado que permite la transferencia eficiente de datos entre dispositivos. Su diseño simple y su capacidad para conectar múltiples dispositivos en el mismo bus lo hacen ideal para aplicaciones que requieren comunicación entre diferentes componentes electrónicos.

6.2 Hardware utilizado

6.2.1 Hardware WeMos D1 R2 Wifi

Para la realización del proyecto como otro de los elementos principales tenemos la placa, WeMos D1 R2. Siendo una alternativa más barata que Arduino Uno Wifi. Esta placa low cost es compatible con Arduino IDE incluyendo también un módulo Wifi como Arduino Uno Wifi. Esta placa será la encargada de recibir y procesar la información de las entradas de una de las salas virtuales creadas en nuestro museo, como también enviar las salidas procesadas y establecer la conexión vía Wifi para comunicarnos con el servidor a través de MQTT. Para poder utilizarla se hará uso de un cable microUSB conectado al ordenador de manera que la placa recibirá 5V de alimentación.

WeMos D1 R2 consta de diferentes modelos que se han fabricado con un propósito diferente y unas características variadas (como el tamaño físico, número de pines E/S, modelo del microcontrolador, etc). Independientemente de la placa WeMos todas pertenecen a la familia de microcontroladores ESP8266, por lo que comparten la mayoría de las características software, como arquitectura, librerías y documentación. En este caso WeMos es un microcontrolador parecido a Arduino UNO Wifi siendo éste una alternativa más barata.

Otra de las herramientas hardware que necesitaremos serán los componentes

necesarios para interconectar los sensores con las placas hardware seleccionadas. Para la realización del circuito electrónico necesitaremos una placa PCB. Una placa electrónica o PCB (Printed Circuit Board o Placa de Circuito Impreso). Es una superficie plana fabricada con un material no conductor, que a su vez consta de distintas capas de material conductor. Hasta el momento, es la forma más estable de construir un circuito electrónico ya que es una manera compacta y sencilla de crear. De esta manera, el usuario final no se debe preocupar por las conexiones eléctricas que necesite el microcontrolador para funcionar y puede empezar a directamente a desarrollar las aplicaciones electrónicas necesarias.

6.2.2 Hardware Elegoo Uno R3

Para la otra sala se va a utilizar la placa Elegoo uno R3. Esta placa también es compatible con Arduino IDE y también será la encargada de recibir y procesar la información que se mostrará en la otra sala creada en nuestro museo, como también enviar las salidas procesadas por el puerto serie de Arduino hacia el servidor de Node-RED. Se podría haber instalado una extensión Wifi o Ethernet como es ‘‘Ethernet Shield W5100’’ pero por establecer otro tipo de conexión y hacer más enriquecido el proyecto con diferentes tipos de conexiones entre el servidor y Arduino se ha decidido establecer conexión por el puerto serie.

La placa Elegoo UNO R3 es una versión low cost que imita a Arduino Uno R3, consta de un chip CH340C y un chip SMD ATmega328p, que garantizan más memoria y velocidades de transferencias más rápidas.

ATmega328p consta de un total de 32 KB de memoria flash, 2 KB de SRAM y 1 KB de EEPROM. La placa también está equipada con:

- 14 pines de entrada/salida digital (SCL, SDA, IOREF)
- 8 entradas analógicas
- Voltaje de entrada 7-12V
- Corriente de 5V de 500MA
- Corriente de 3.3V de 50 MA
- 14 pines de E/S digitales (6 salidas PWM)
- 1 Reset-Pin

- Frecuencia de reloj de 16 Mhz

6.2.3 Sensores utilizados

Para medir los valores de luminosidad utilizaremos el sensor BH1750. Se trata de un sensor digital que nos entrega valores de medición en Lux (lumen / m²) siendo esta la medida estándar para el nivel de iluminación y la que necesitaremos para poder medir con más exactitud los niveles de luminosidad. Su rango de precisión es de 1 – 65535 Lux (configurable).

El módulo consta de un regulador interno de 3.3V pudiendo usarse una alimentación de 5V sin problemas. Tiene una interfaz de comunicación I2C que hace que sea compatible con la mayoría de los microcontroladores. El módulo consta también de un pin para establecer la dirección de trabajo:

Pin ADDR	Dirección I2C
ADDR=HIGH(5V)	Dirección 0x5C
ADDR = LOW (GND)	Dirección 0x23

Tabla 6.1 Direcciones de trabajo del sensor BH1750

Si no conectamos el pin ADDR la dirección será 0x23 ya que, internamente el módulo tiene una resistencia a GND.

También podemos configurar la resolución del sensor viéndose afectada la velocidad de medición:

MODO	RESOLUCIÓN	TIEMPO DE MEDICIÓN
High resolution Mode2	0.5 lx	120 ms
High resolution Mode	1 lx	120 ms
Low resolution Mode	4 lux	16 ms

Tabla 6.2 Resolución del sensor BH1750

Estas configuraciones se subdividen en dos:

-Modo continuous: Se utilizará cuando necesitemos medir constantemente.

-Modo one time: El módulo se apaga después de realizar una medición, luego tendremos que volver a configurar.

Para nuestro proyecto se utilizará el modo continuos.

Para medir los valores de temperatura y humedad constantes utilizaremos el sensor DHT22 que muestra las siguientes características:

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

Se utilizará este sensor ya que es un sensor más preciso y ofrece un mayor rango de lectura de datos en comparación con el DHT11 (otra de las opciones estudiadas).

Para medir la contaminación haremos uso de sensores que midan los niveles de CO2 aparte de tener en cuenta la humedad, ya que para evitar una mayor contaminación la humedad debe estar entre el 70-80%. Se llega a la conclusión de que para medir el nivel de CO2 utilizaremos el sensor MQ135, ya que es un sensor más sensible y es más asequible por el usuario para medir la calidad del aire en espacios cerrados.

Este sensor cuenta con las siguientes características:

- Voltaje de operación: 5V.
- Corriente de operación: 150 mA.
- Potencia de consumo: 800 mW.
- Tiempo de precalentamiento: 20 segundos.
- Resistencia de carga: Potenciómetro (Ajustable)
- Detección de partes por millón: 10ppm-1000 ppm.
- Concentración detectable: Amoniaco, sulfuro, benceno, humo.

6.3 Esquemático del circuito de salas de museo

Antes de comenzar con las fases de prueba del sistema, se presenta el esquemático del circuito realizado para nuestro sistema. En esta descripción detallada, se destacarán las conexiones y configuraciones específicas utilizadas. Además, se abordarán las características y propiedades de los diferentes componentes, así como su interacción en el circuito global, de manera que se pueda comprender plenamente cómo se logra el objetivo final de este sistema eléctrico.

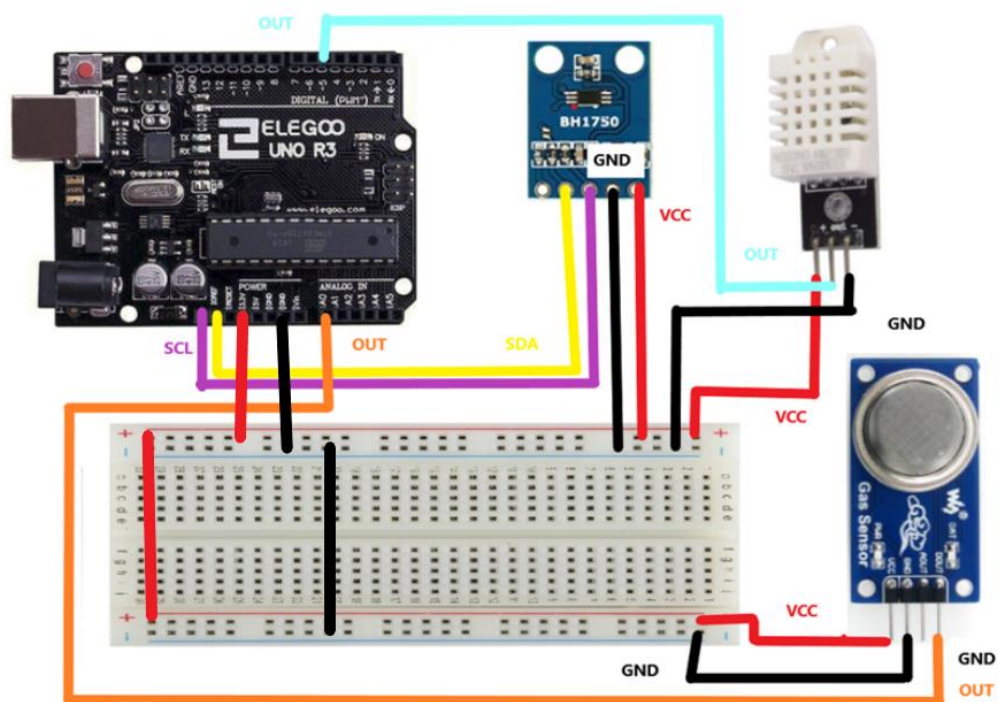


Figura 6.4 Esquemático 1 del circuito utilizado con Paint3D

En la figura 6.4 se muestra las conexiones entre la placa Elegoo Uno R3 y los diferentes sensores encargados de leer los valores de temperatura, humedad, iluminación y calidad de aire de una de las salas. A continuación, se describirán los componentes necesarios para su interconexión.

1. Conexión a tierra y alimentación: Para asegurar el correcto funcionamiento de los sensores es necesario establecer conexiones de tierra y alimentación. Se debe conectar el pin VCC de cada sensor a la fuente de alimentación de 5V de la placa, mientras que el pin GND debe conectarse a la conexión a tierra de la placa. Para facilitar estas conexiones, se puede utilizar una placa PCB que permita interconectar todos los sensores sin dificultad.

2. Sensor BH1750: Para conectar el sensor BH1750 a la placa hardware, se requieren cuatro cables. Los pines de salida del sensor, SCL (línea de reloj) y SDA (línea de datos), deben conectarse a los pines de E/S digitales de la placa designados para esta función específica.

3. Sensor DHT22: Para conectar el sensor DHT22 a la placa hardware, se necesitan tres cables. El pin de salida (out) del sensor se conecta a uno de los pines de E/S digitales de la placa, en este proyecto utilizaremos el pin D5 pudiendo utilizarse cualquier otro.

4. Sensor MQ135: Para conectar el sensor MQ135, se requieren tres cables. El pin de salida analógica (out) del sensor se conecta al pin de entrada analógica correspondiente en la placa.

Estas conexiones permitirán establecer la comunicación y transferencia de datos entre los sensores y la placa, lo que facilitará la adquisición y procesamiento de información por parte del microcontrolador o sistema utilizado en nuestro proyecto.

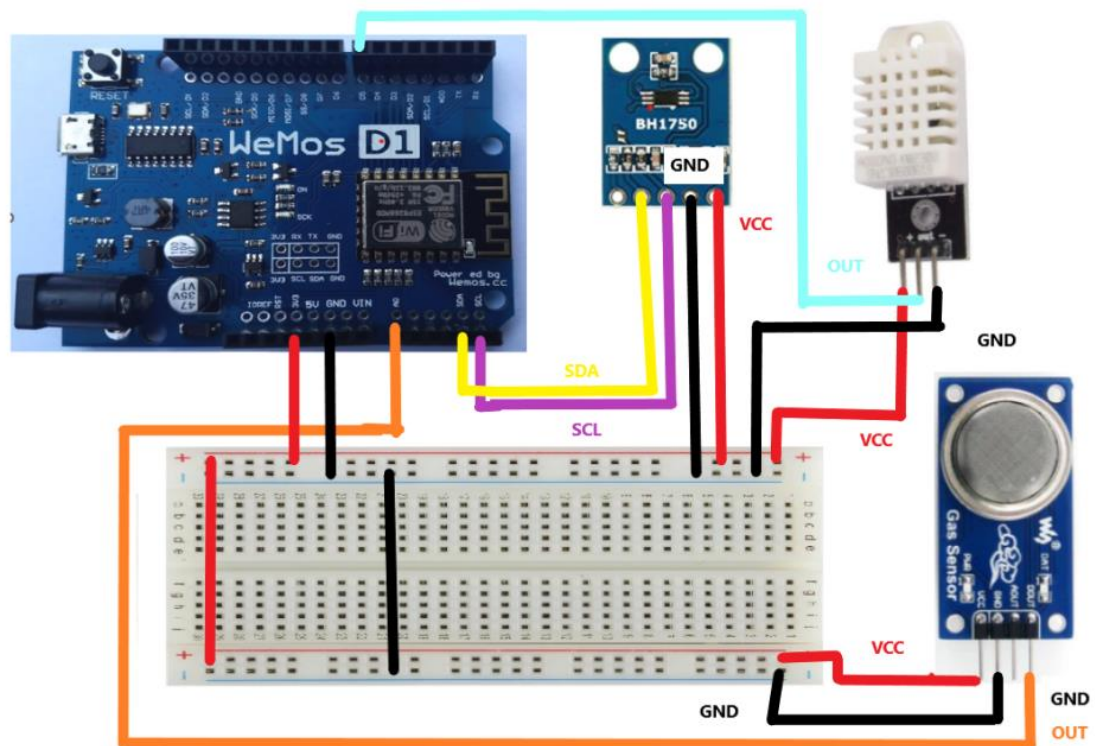


Figura 6.5 Esquemático 2 del circuito utilizado con Paint3D

En la Figura 6.5 se ilustran las interconexiones entre la placa Wemos D1 Wifi y los diversos sensores encargados de medir los valores de temperatura, humedad, iluminación y calidad del aire en una de las salas. Es notable que la configuración de la circuitería es idéntica, por lo que se considera que los componentes están explicados de manera detallada en el apartado que describe la Figura 6.5.

En la siguiente figura 6.6 se presenta el resultado final del montaje del circuito electrónico. Esta imagen muestra cómo se han conectado todos los componentes y elementos del circuito, reflejando la configuración y disposición física del mismo. El resultado final del montaje es crucial para asegurar el correcto funcionamiento del circuito y su capacidad para cumplir con los objetivos y funcionalidades previstas.

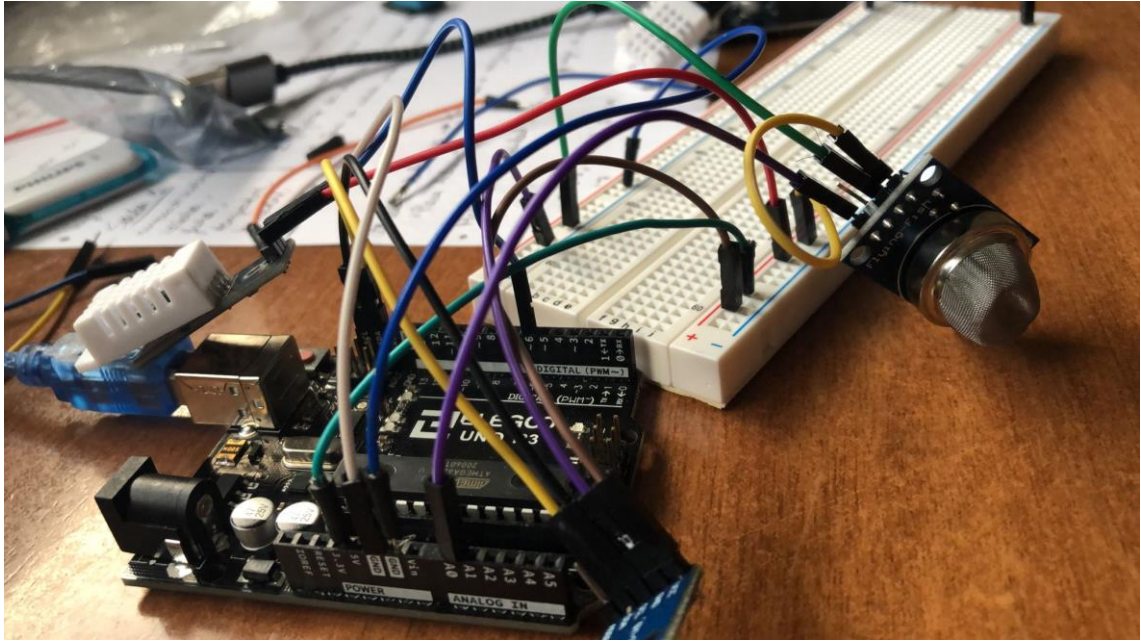


Figura 6.6 Comunicación del circuito con Arduino

6.4 Estudio previo.

Antes de poder empezar con el desarrollo del proyecto, se ha tenido que realizar un estudio previo de NodeRED, repasar conceptos de Arduino y el uso de las librerías que van a ser usadas en este proyecto.

Para familiarizarse con NodeRED y con su entorno de programación ha sido de gran ayuda los cursos online gratuitos de Youtube “*Node-RED Curso básico I*”^[19] y “*Node-RED Curso básico II*”^[20] impartido por “Mecatrónica MADE”. Ya que, en el primer tutorial se explican los fundamentos teóricos de Node-RED y el uso de los diferentes nodos comúnmente utilizados. En general se imparten unas bases básicas para aprender a utilizar Node-RED en el mundo IOT. En el segundo tutorial se trabaja con Arduino/ESP como cliente suscriptor, utilizando la herramienta Node-RED como Cliente publicador y comunica ambos clientes bajo el Broker Mosquitto. Este segundo tutorial ha servido para adquirir los conceptos de programación que se aplican para simular una de las salas del museo donde se ha utilizado la placa WeMos D1 con el protocolo MQTT y la conexión con el Broker utilizado.

En cuanto a Arduino, en el grado se han impartido varias asignaturas donde se ha utilizado Arduino IDE, por lo que, se ha tenido que repasar los fundamentos e instalar las nuevas librerías que se utilizarán en el proyecto.

Para poder crear una Dashboard que cumpla con los requisitos establecidos en nuestro proyecto ha sido de gran ayuda el tutorial impartido por MecatronicsMADE “*Curso Node-RED Dashboard (Básico I)*” [21]. Gracias a este tutorial se han adquirido los conocimientos básicos para la creación de una Dashboard básica donde posteriormente el alumno ha ido investigando para poder llegar a crear una Dashboard más elaborada.

Capítulo 7. Parte Experimental y Resultados

7.1 Pruebas y resultados

Este proyecto se llevó a cabo con uno de los objetivos de recolectar datos precisos y confiables sobre diversas variables ambientales que aseguraran la correcta conservación de las obras de arte. Para evaluar la eficiencia y rendimiento del sistema, se realizaron pruebas exhaustivas que abarcaron diferentes condiciones y escenarios. Durante estas pruebas, se recopilaron mediciones continuas y se compararon con referencias de alta precisión para garantizar la calidad de los resultados obtenidos. Los resultados obtenidos demostraron la capacidad del sistema de sensores para monitorear de manera precisa y estable las variables de interés, lo que valida su idoneidad para aplicaciones en tiempo real y proporciona una base sólida para su implementación en entornos prácticos.

7.1.1 Evaluación de los sensores y conexión Wifi

En esta sección se explicará el proceso que se llevó a cabo para la verificación del correcto funcionamiento de los sensores en ambas placas utilizadas. Para lograr esto, se realizaron pruebas en diferentes salas del domicilio del alumno para así verificar el correcto funcionamiento de los sensores. Para estas pruebas se utilizó el puerto serie de Arduino IDE, donde se muestran los datos recibidos o las alertas correspondientes en función de los posibles fallos en los componentes hardware. Las lecturas de los datos se realizaron cada minuto para asegurar las condiciones meteorológicas estables dentro de las salas. Estas pruebas fueron fundamentales para asegurar que los sensores están capturando y transmitiendo la información de manera adecuada, y permitieron identificar cualquier problema o mal funcionamiento que pudo surgir. El uso del puerto serie en Arduino IDE facilitó la visualización en tiempo real de los datos y alertas, lo que facilitó el proceso de depuración y diagnóstico de posibles fallas en los sensores.

A continuación, se mostrarán y detallarán los diferentes casos de uso para verificar el correcto funcionamiento del sistema durante estas pruebas.

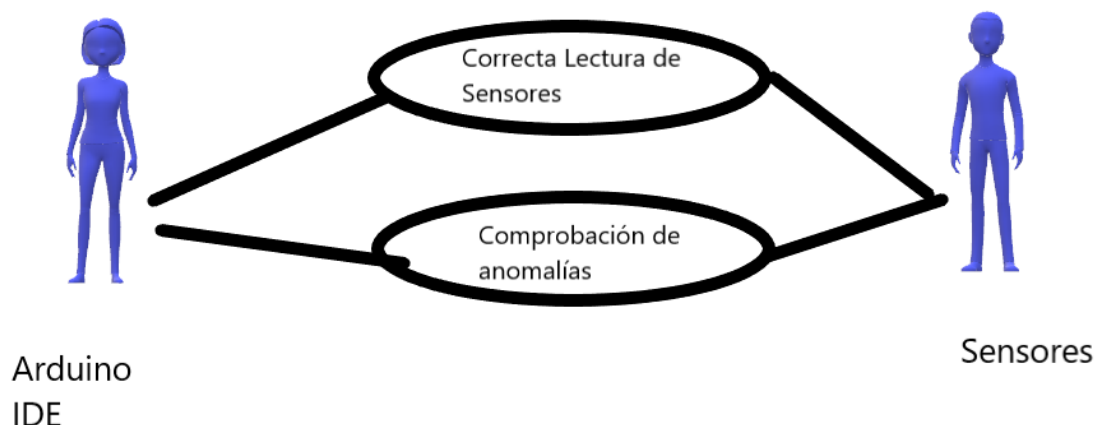


Figura 7.1. Casos de Uso para comprobar el correcto funcionamiento de la lectura de sensores

Nombre del caso	Correcta lectura de Sensores
Actores	1. Arduino IDE 2. Sensores
Dependencias	-
Propósito	Comprobar el correcto funcionamiento de la lectura de los datos
Resumen	Una vez que se realizó las pruebas convenientes de que los sensores se configuraban y visualizaban correctamente se llevó a cabo una serie de pruebas para verificar la correcta lectura de los sensores
Curso normal básico	<ol style="list-style-type: none"> 1. Recibir los diferentes valores de los sensores que llegan desde las placas hardware a través de los puertos usb. 2. Procesar estos valores de manera que estén aptos para ser leídos y posteriormente enviados a Node-RED 3. Asegurarse de realizar una correcta visualización de los datos

	leídos a través del Puerto serie de Arduino. De manera que fuese mas visual detectar cualquier problema.
Curso alternativo	La visualización da fallo y se detecta una anomalía en las lecturas
Observaciones	El sistema debe de visualizar siempre los valores leídos.

Tabla 7.1 Explicación Caso de Uso “Correcta la lectura de los sensores”

Después de confirmar la lectura correcta de los sensores, se llevaron a cabo pruebas para verificar la conexión Wifi-adeuada de una de las placas. Se utilizó un código desarrollado específicamente para asegurar la correcta conexión del dispositivo a Internet. Esto es importante porque la conexión Wifi puede tener un impacto en el sistema, ya que podría impedir el envío de los datos recopilados por los sensores.



Figura 7.2 Casos de uso de la Verificación de la conexión Wifi

Nombre del caso	Verificación de la conexión Wifi
Actores	1. Arduino IDE 2. Placa Elegoo Uno
Dependencias	-
Propósito	Comprobar la correcta conexión wifi del sistema
Resumen	Se realiza una comprobación de la correcta conexión wifi a través del código

	Arduino desarrollado. Verificando que se comprueba la correcta conexión del Sistema a Internet.
Curso normal básico	<ol style="list-style-type: none"> 1. Una vez ejecutado el Código Arduino, se verifica que las credenciales introducidas son las correctas para establecer la conexión Wifi. 2. Procesar estas credenciales para establecer la conexión Wifi. 3. Asegurarse de realizar una correcta configuración de las credenciales mostrando por el Puerto serie de Arduino un mensaje de éxito en la conexión.
Curso alternativo	La visualización da fallo y se detecta una anomalía en la conexión Wifi.
Observaciones	El sistema debe de visualizar siempre el éxito de la conexión.

Tabla 7.2 Explicación Caso de Uso “Verificación de la conexión wifi”

A continuación, se muestra en la Figura 6.3 como se visualizarían los datos leídos de los sensores y la correcta conexión Wifi del sistema.

```

.....
Conexión WiFi establecida
Dirección IP: 192.168.0.155
Iniciando sensores...

Conectando al broker MQTT...
Conexión MQTT establecida
Temperatura: 31.00 °C
Humedad: 55.00 %
CO2: 491.00 ppm, CO: 1186.00 ppm
Iluminación: 18
Conectando al broker MQTT...
Conexión MQTT establecida
Temperatura: 31.00 °C
Humedad: 55.00 %
CO2: 498.00 ppm, CO: 1196.00 ppm
Iluminación: 17

```

Figura 7.3 Visualización de los datos obtenidos por el puerto serie de Arduino

En caso de fallo de lectura o inicialización de los sensores se mostraron avisos por el mismo puerto serie, haciendo más fácil la lectura de los posibles fallos que pudieron surgir. Para asegurar un correcto funcionamiento de este sistema de avisos se aseguró que el código Arduino era capaz de detectar anomalías en la lectura o inicialización de los sensores.

7.6.2 Evaluación de Node-RED

En este apartado se describirán las pruebas realizadas en Node-RED para asegurar una correcta visualización y almacenamiento de los datos, como también el correcto recibimiento de los datos desde Arduino IDE. Para ello se realizaron pruebas en ambos flujos de las salas simuladas. A continuación, se describen los diferentes casos de uso que comprueban el correcto funcionamiento de Node-RED.

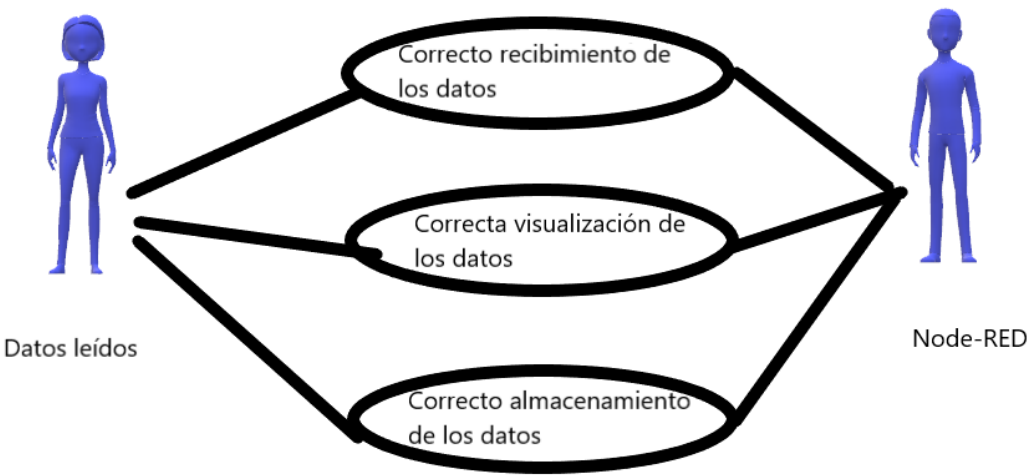


Figura 7.4. Casos de Uso para el correcto funcionamiento de Node-RED

Nombre del caso	Correcta recibimiento de los datos
Actores	1. Datos leídos 2. Node-RED
Dependencias	CU- Correcta Lectura de los sensores, CU- Correcta Conexión Wifi
Propósito	Comprobar el correcto recibimiento de los

	datos leídos por los sensores desde Arduino IDE
Resumen	Se realiza una comprobación del correcto recibimiento de los datos a través Node-RED haciendo uso del Sistema de alertas montado. Verificando que se comprueba la correcta lectura de los datos.
Curso normal básico	<ol style="list-style-type: none"> 1. Se hace uso del Sistema de alertas montado entre Node-RED y Telegram para asegurar que los datos se reciben correctamente. 2. Se procesan los datos en el para su posterior visualización. 3. Asegurarse de realizar una correcta configuración de las credenciales enviando los datos a Interfaz Gráfica asegurando el éxito de las lecturas.
Curso alternativo	La visualización da fallo y se detecta una anomalía en el recibimiento de los datos, que producirá una alerta.
Observaciones	El sistema debe de visualizar siempre el éxito de la lectura.

Tabla 7.3 Explicación Caso de Uso “Correcto recibimiento de los datos”

Nombre del caso	Correcta visualización de los datos
Actores	<ol style="list-style-type: none"> 3. Datos leídos 4. Node-RED
Dependencias	CU- Correcta Lectura de los sensores, CU- Correcta Conexión Wifi
Propósito	Comprobar la correcta visualización de los datos leídos por los sensores desde

	Arduino IDE.
Resumen	Se realiza una comprobación de la correcta visualización de los datos a través Node-RED haciendo uso de la Interfaz Gráfica. Verificando que se comprueba la correcta visualización de los datos.
Curso normal básico	<p>4. Se hace uso de la Interfaz Gráfica de Usuario para visualizar los datos recibidos a través de Node-RED.</p> <p>5. Asegurarse de realizar una correcta visualización de los datos, mostrando fallo en el caso de que se produzca una incorrecta visualización de los fallos.</p>
Curso alternativo	La visualización da fallo y se detecta una anomalía en la visualización de los datos, que producirá una alerta.
Observaciones	El sistema debe de visualizar siempre los datos recibidos.

Tabla 7.4 Explicación Caso de Uso “Correcta visualización de los datos”

Nombre del caso	Correcto almacenamiento de los datos
Actores	<p>5. Datos leídos</p> <p>6. Node-RED</p>
Dependencias	CU- Correcta Lectura de los sensores, CU- Correcta Conexión Wifi, CU-Correcta visualización de los datos
Propósito	Comprobar el correcto almacenamiento de los datos leídos y visualizados por Node-RED.
Resumen	Se realiza una comprobación del correcto almacenamiento de los datos a través

	Node-RED.
Curso normal básico	<p>6. Se hace uso del MySQL para asegurar el correcto almacenamiento de los datos. Mostrando mensajes de fallo en el caso de que no se almacenen correctamente.</p> <p>7. Se almacenan los datos para posible uso posteriormente.</p>
Curso alternativo	El almacenamiento da fallo y se detecta una anomalía en el almacenamiento de los datos, que producirá una alerta.
Observaciones	El sistema debe de almacenar los datos siempre.

Tabla 7.5 Explicación Caso de Uso “Correcto almacenamiento de los datos”

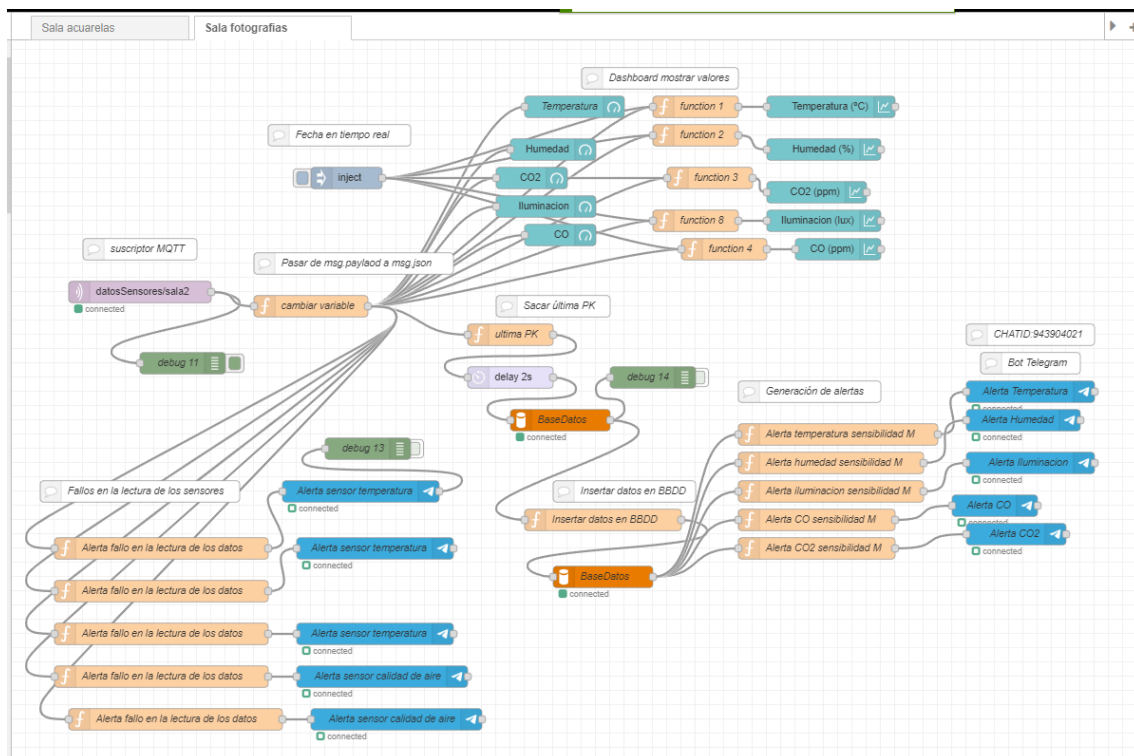


Figura 7.5. Flujo del sistema en Node-RED

La figura 7.5 representa el flujo de Node-RED desde la recepción de datos desde Arduino IDE hasta su almacenamiento en la base de datos (BDD) y la posible detección de anomalías. Con el fin de garantizar un funcionamiento adecuado de este flujo, se llevaron a cabo pruebas divididas en cuatro fases:

1. Fase de lectura de datos desde Arduino: Para verificar la correcta recepción de los datos provenientes de Arduino IDE, se utilizaron nodos de depuración (debug) que mostraban los datos recibidos. También se tuvo en cuenta la alerta al personal responsable enviando mensajes al canal de Telegram destinado para el aviso de estas anomalías. Se realizaron pruebas manteniendo los sensores capturando datos durante períodos prolongados (alrededor de dos días) para observar el formato de los datos y asegurar su correcto funcionamiento a lo largo del tiempo. Esto permitió descartar posibles problemas que podrían conducir a un mal funcionamiento del sistema si los sensores se dejaban en funcionamiento durante largos períodos.

Es importante destacar que, para la sala simulada donde se reciben los datos a través del protocolo MQTT, se llevaron a cabo pruebas algo distintas con ciertas particularidades debido a la forma en que Node-RED utiliza algunos nodos. Fue necesario tener en cuenta que algunos nodos desarrollados por usuarios de la comunidad especifican que la variable en la que se almacenará la respuesta de dichos nodos es "msg.payload". Esta variable proporcionada por Node-RED ayuda a garantizar la integridad de los datos al utilizar la plataforma. Sin embargo, se tuvo que tener precaución al utilizar más de un nodo que utilice "msg.payload" como almacenamiento de los datos procesados, ya que fue necesario realizar cambios en las variables.

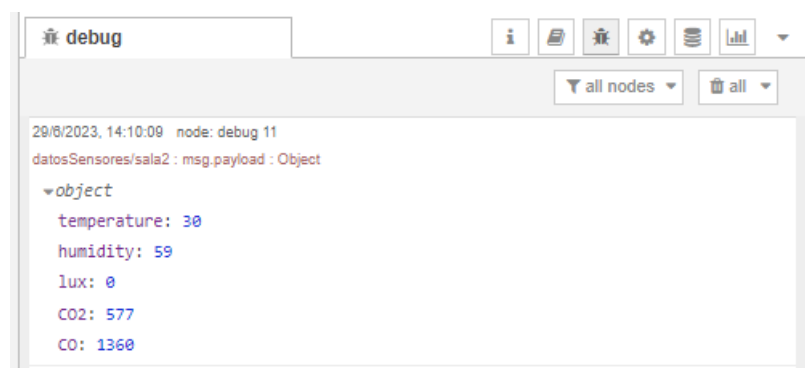


Figura 7.6. Datos recibidos en Node-RED

2. Fase de visualización de los datos en la Interfaz de Usuario. Durante esta fase de pruebas, se consideraron posibles fallos relacionados con la visualización de los

datos, como la configuración incorrecta de los nodos en Node-RED. Es importante destacar que, gracias a la programación basada en flujos en Node-RED, se puede garantizar una visualización correcta de los datos, sólo asegurándonos de descartar problemas de configuración, como se hizo en esta fase. Este único foco de pruebas se debe a que, en la fase anterior de pruebas, se abordaron y descartaron errores potenciales relacionados con los sensores y su lectura en etapas anteriores.

Durante esta fase de pruebas, se llevaron a cabo manipulaciones en las condiciones meteorológicas de la sala de pruebas con el objetivo de evaluar su impacto en la visualización de gráficas y manómetros que representan los diferentes valores capturados. Esta interfaz de usuario presenta gráficas que muestran, en función del tiempo, tanto los valores de la última medición como los valores registrados a lo largo del tiempo. Estas representaciones visuales facilitan el seguimiento y mantenimiento adecuado de las salas.

Durante las pruebas, se ajustaron los valores asociados a diferentes colores para proporcionar una representación visual más clara y detallada de las condiciones meteorológicas. Este enfoque permitió tener un control más exhaustivo de las condiciones antes de que se generen alertas. La utilización de colores específicos ayudó a identificar rápidamente posibles anomalías o situaciones críticas, lo que contribuyó a una gestión más eficiente y precisa de las condiciones ambientales.

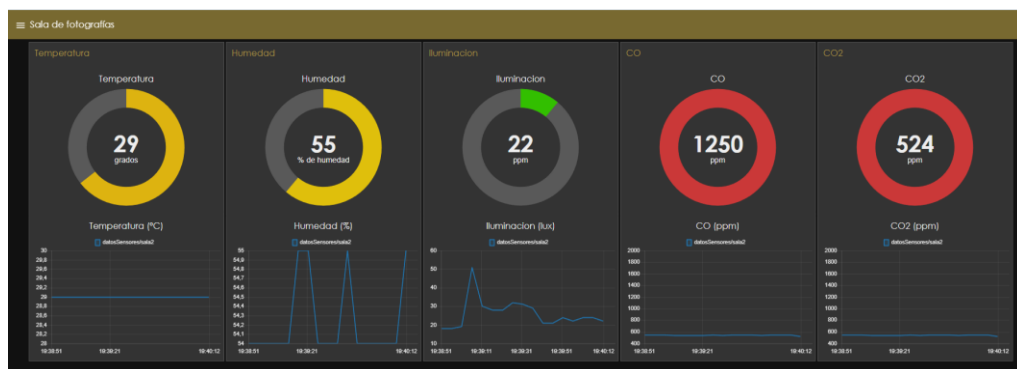
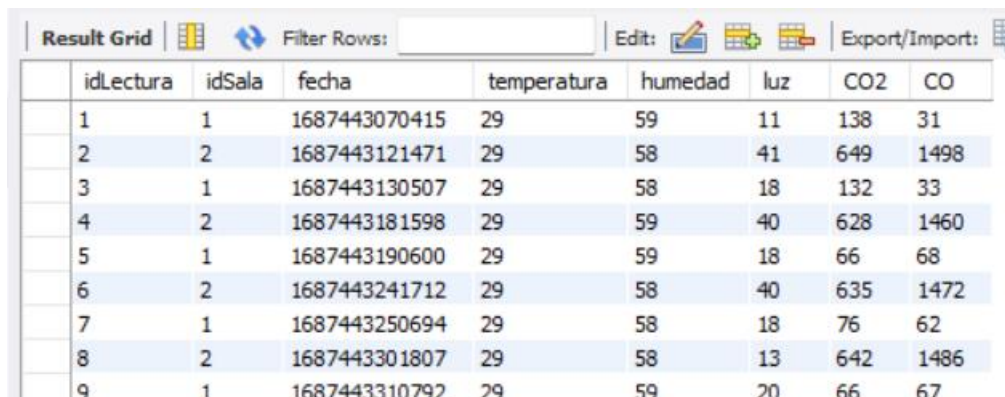


Figura 7.7 Visualización de los datos en la Interfaz de Usuario

- 3 Fase de pruebas de almacenamiento de datos en BBDD. Durante esta fase de pruebas, se realizaron pruebas prolongadas en el tiempo con el objetivo de identificar posibles errores en el almacenamiento de datos. Se tuvieron en cuenta diversos fallos potenciales, como un almacenamiento incorrecto de los datos o errores en el formato de estos. Es fundamental garantizar un almacenamiento preciso y confiable de los datos capturados para asegurar la

integridad de la información.

Durante estas pruebas, surgieron dificultades en el almacenamiento de datos procedentes de ambas salas, ya que se utiliza la misma base de datos, pero se requería identificar la procedencia de cada una de las salas mediante un id de sala.



	idLectura	idSala	fecha	temperatura	humedad	luz	CO2	CO
	1	1	1687443070415	29	59	11	138	31
	2	2	1687443121471	29	58	41	649	1498
	3	1	1687443130507	29	58	18	132	33
	4	2	1687443181598	29	59	40	628	1460
	5	1	1687443190600	29	59	18	66	68
	6	2	1687443241712	29	58	40	635	1472
	7	1	1687443250694	29	58	18	76	62
	8	2	1687443301807	29	58	13	642	1486
	9	1	1687443310792	29	59	20	66	67

Figura 7.8 Datos almacenados en BBDD

Para abordar este desafío, se implementó una solución ajustando el tiempo de las mediciones. En lugar de tomar mediciones simultáneas, se estableció un intervalo de 2 segundos entre las mediciones de ambas salas. De esta manera, se logró evitar conflictos en el almacenamiento de datos, permitiendo que se registren correctamente y en tiempo real sin superponerse.

Esta solución garantizó que los datos capturados de cada sala se almacenen adecuadamente, asegurando su correcta identificación y evitando cualquier pérdida o sobrescritura accidental. Además, el ajuste en el tiempo de las mediciones no afectó significativamente la frecuencia de captura de datos, lo que permitió mantener un seguimiento constante y actualizado de las condiciones de ambas salas.

4 Fase de pruebas del Sistema de alertas. A la hora de garantizar un correcto funcionamiento del sistema de alertas se llevó el sistema a producir mediciones que preocupasen a la conservación preventiva de las obras de arte situadas en las simulaciones de las diferentes salas. Para así asegurar el correcto funcionamiento del sistema de alertas.

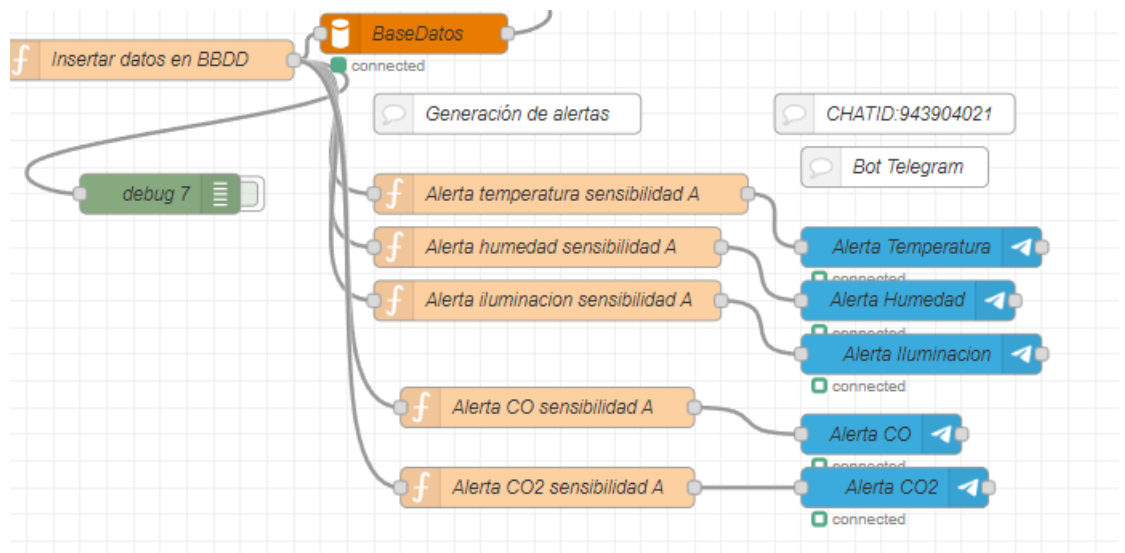


Figura 7.9. Flujo del sistema de alertas en Node-RED

En la Figura 7.9 se puede apreciar que se realizó una verificación del flujo en Node-RED para asegurar su correcto funcionamiento en el envío de notificaciones a través de Telegram a través de los nodos debug. Con el fin de comprobar esta funcionalidad de alertas del sistema, se utilizó el dispositivo móvil del alumno configurado para recibir las alertas generadas por los valores meteorológicos capturados.

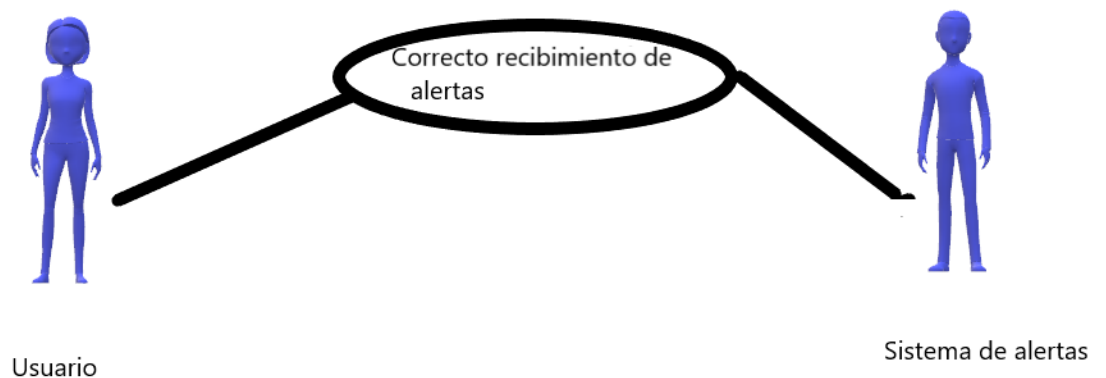


Figura 7.10. Casos de Uso para verificar el correcto funcionamiento del sistema de alertas

Nombre del caso	Correcto recibimiento de alertas
Actores	7. Usuario 8. Sistema de alertas
Dependencias	CU- Correcta Lectura de los sensores, CU-

	Correcta Conexión Wifi, CU-Correcta visualización de los datos, CU-Correcto almacenamiento de los datos
Propósito	Comprobar el correcto recibimiento de las alertas producidas.
Resumen	Se realiza una comprobación del correcto recibimiento de las alertas a través de Node-RED
Curso normal básico	<p>8. Se expone el Sistema a valores meteorológicos fuera del rango de valores óptimos para producir las alertas.</p> <p>9. Se verifica por Node-RED el correcto procesamiento y envío de las alertas al personal responsable</p>
Curso alternativo	El Sistema no produce ninguna alerta porque no existen anomalías
Observaciones	El sistema debe de enviar alertas siempre que los valores meteorológicos superen los umbrales definidos.

Tabla 7.6 Explicación Caso de Uso “Correcto recibimiento de alertas”

Durante esta etapa de pruebas, se llevaron a cabo diferentes escenarios para evaluar la recepción y visualización de las alertas en el dispositivo móvil. Estos escenarios incluyeron la simulación de diversas condiciones meteorológicas y la generación intencionada de alertas correspondientes, como pudo ser un fallo en la lectura de los sensores o simular ambientes meteorológicos que estuvieran fuera de los umbrales establecidos. De esta manera, se pudo verificar el correcto funcionamiento del sistema de alertas y su capacidad para enviar notificaciones en tiempo real a través de la aplicación de Telegram en el dispositivo móvil del alumno.



AlertasMuseoTFG
bot

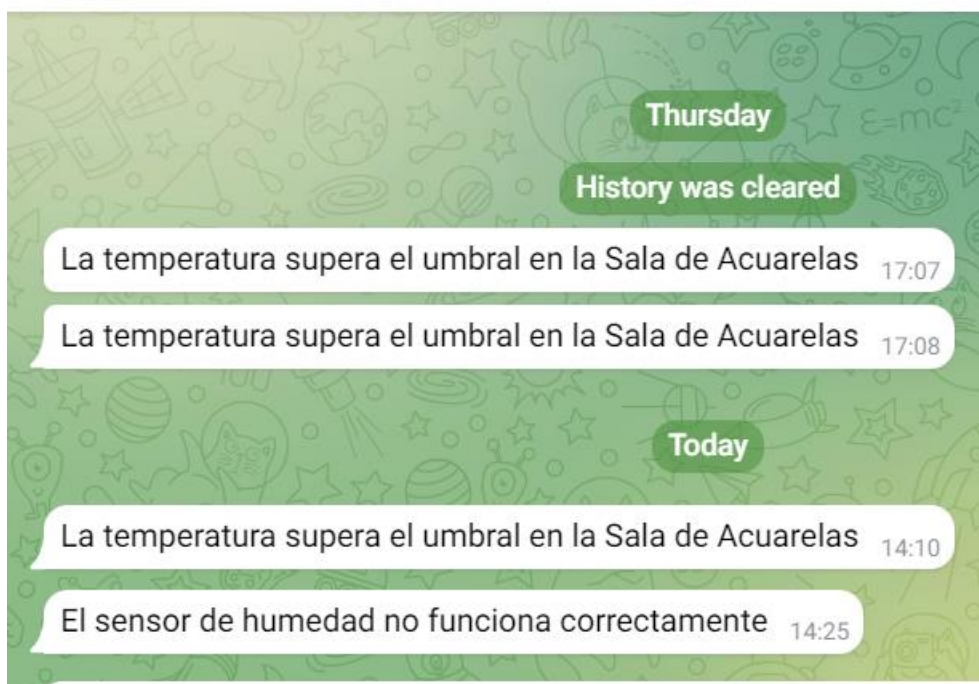


Figura 7.11. Sistema de Alertas mediante bot de Telegram

Capítulo 8. Conclusiones

8.1 Valoración personal

La realización de este proyecto ha sido un proceso lento pero muy gratificante, ya que finalmente se han podido cumplir los objetivos propuestos, consiguiendo combinar el hardware con el software sin problemas mayores, pudiendo así darle solución.

La etapa inicial del proyecto, una vez que tuve claro lo que quería hacer fue la más difícil debido a la fase de aprendizaje, que abarcó la mayor parte del desarrollo del proyecto. Adquirir los conocimientos sobre Node-RED para la lectura de las señales a través de los sensores y luego integrarlo con los nodos de mysql que a su vez se conectaban con la base de datos MySQL, aparte de aprender sobre el protocolo MQTT e integrarlo con Arduino IDE y con el nodo MQTT de Node-RED, han sido los procesos de aprendizaje más costosos para mí, ya que implicaba aprender conceptos sobre la programación basada en flujos de Node-RED que no conocía.

Sin embargo, todo el tiempo dedicado al desarrollo del proyecto ha sido muy beneficioso. Ha mejorado mi capacidad de aprendizaje y me ha permitido aplicar los conocimientos adquiridos durante mi paso por la carrera universitaria. Además, me ha brindado la confianza necesaria para abordar un problema desde el inicio hasta su final, superando los errores que han surgido en el camino.

Finalmente, estoy muy contento de haber llegado hasta aquí ya que, en un punto de la realización del proyecto pensé que con el resto de las asignaturas y el trabajo que realizo fuera de la escuela me sería imposible llegar a finalizarlo.

8.2 Limitaciones y futuras mejoras

Una vez completado el proyecto, es posible identificar las siguientes limitaciones que podrían mejorarse en el futuro mediante una mayor inversión económica en el mismo. A continuación, se detallarán estas limitaciones y las posibles áreas de mejora que podrían abordarse mediante una mayor inversión financiera en el proyecto.

La principal limitación se encuentra la limitación presupuestaria del alumno, ya que, se ha tenido que trabajar dentro un límite financiero bajo y estricto. Por lo que esto ha afectado a adquirir componentes y sensores de mayor calidad, lo que resulta en

mediciones menos precisas o confiables que con equipos más costosos. Aparte de afectar a los componentes utilizados en este trabajo el límite financiero también ha afectado al escenario descrito, ya que, si se contase con más placas o más sensores podríamos simular un museo completo, exposiciones al aire libre...etc. y hacer así un proyecto que se ajuste más a la realidad.

En consecuencia, de haber utilizado placas Arduino de bajo coste debemos de tener en cuenta que tendremos limitaciones de energía o de escala, ya que, el uso de dos placas Arduino puede presentar desafíos si se necesita ampliar el sistema a más salas futuras. En cuanto a la energía, como fuentes de alimentación se mencionan los puertos USB del ordenador alimentados a 5V. Actualmente este proyecto depende de la duración y estabilidad de la alimentación del pc, sería recomendable considerar el uso de fuentes de alimentación alternativas.

Presentando mejoras futuras podrían utilizarse sensores inalámbricos de mayor coste ya que, son fáciles de instalar puesto que no requieren de conexiones físicas. Esto simplificaría el proceso de construcción del circuito y reducción del tiempo del proceso de implementación para la configuración del sistema. Los sensores inalámbricos permiten una mayor flexibilidad y escalabilidad en comparación con los sistemas cableados pudiendo así ubicar los sensores en diferentes ubicaciones sin restricciones de cableados como se tiene actualmente. Esto facilitaría la monitorización en áreas de difícil acceso. Además, a largo plazo sería una mejora para agregar más sensores sin la necesidad de realizar nuevas instalaciones de cables.

8.3 Conclusiones

Basándonos en el estudio que se ha realizado sobre el estudio de conservación preventiva de arte y el análisis de hardware, se exponen las siguientes conclusiones:

- Se expone durante el desarrollo de este proyecto cómo la conservación preventiva es muy necesaria para poder mantener las obras que existen actualmente en nuestro patrimonio, para ello, una forma fácil y eficaz es construir sistemas monitorizados para poder detectar anomalías en el ambiente de los museos.
- El desarrollo de este proyecto está orientado a construir un sistema monitorizado final que se considere un producto útil para cubrir la poca

información sobre monitorización de sistemas asociados a la conservación preventiva del patrimonio cultural y arte.

- Se puede ver como la combinación de Node-RED (Software) y placas Arduino (Hardware) dan solución a problemas reales para dar solución a la monitorización de valores meteorológicos que puedan estropear el patrimonio cultural en museos. La investigación y aprendizaje de Node-RED hace que se demuestre lo eficaz que es esta aplicación para proyectos IoT. De manera que el uso de Node-RED supone una mejora para los sistemas de monitorización aportando un bajo consumo y velocidades de transmisión en tiempo real.

- Se exponen mejoras fáciles y asequibles para crear sistemas de alerta como lo son los bot de Telegram. Las ventajas del uso de redes sociales como Telegram suponen una comunicación instantánea y en tiempo real con el personal responsable. El acceso a Telegram es desde cualquier lugar, ya que, puede recibir alertas en un dispositivo móvil. Finalmente, otra de las ventajas de Telegram es la integración con otros servicios y plataformas, lo que amplía las posibilidades de gestión y análisis de los datos recopilados.

En resumen, el estudio de este trabajo y su realización subrayan la importancia de la conservación preventiva proponiendo un sistema monitorizado como solución útil destacando el uso de Node-RED y resaltando las ventajas de utilizar las redes sociales para el sistema de alertas.

Capítulo 9. Anexos

9.1 Manual de usuario de Node-RED

Este manual de usuario pretende introducir al usuario a la herramienta de Node-RED para comprender el uso de ésta antes de mostrar las pruebas realizadas durante la realización del sistema.

9.1.1 Instalación de Node-RED

Como se comentó en el apartado donde se describía la herramienta los pasos para instalar Node-RED serían los siguientes:

1. Descargar e instalar Node-RED siguiendo las instrucciones específicas para tu sistema operativo desde el sitio oficial de Node-RED <https://nodered.org>
2. Iniciar Node-RED ejecutando el comando correspondiente en la línea de comandos “node-red” o utilizando el método de inicio específico de tu sistema.

9.1.2 Interfaz de Node-RED

Para acceder a la interfaz de Node-RED debemos de ingresar la siguiente URL en el navegador web: <http://localhost:1880>.

La interfaz de Node-RED consta de tres áreas principales:

1. Barra de herramientas: contiene botones para crear, guardar y desplegar flujos, así como acceder a la paleta de nodos y a la configuración del entorno.

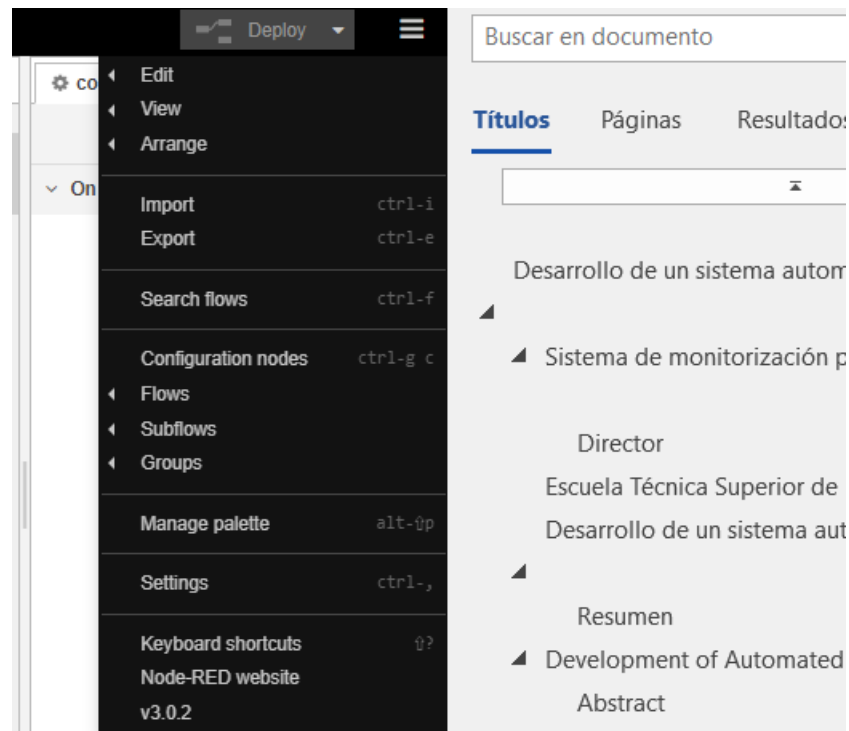


Figura 9.1 Barra de herramientas de Node-RED

2. Editor de flujos: Aquí es donde puedes crear y editar flujos de trabajo conectando nodos arrastrándolos desde la paleta y uniéndolos entre sí.

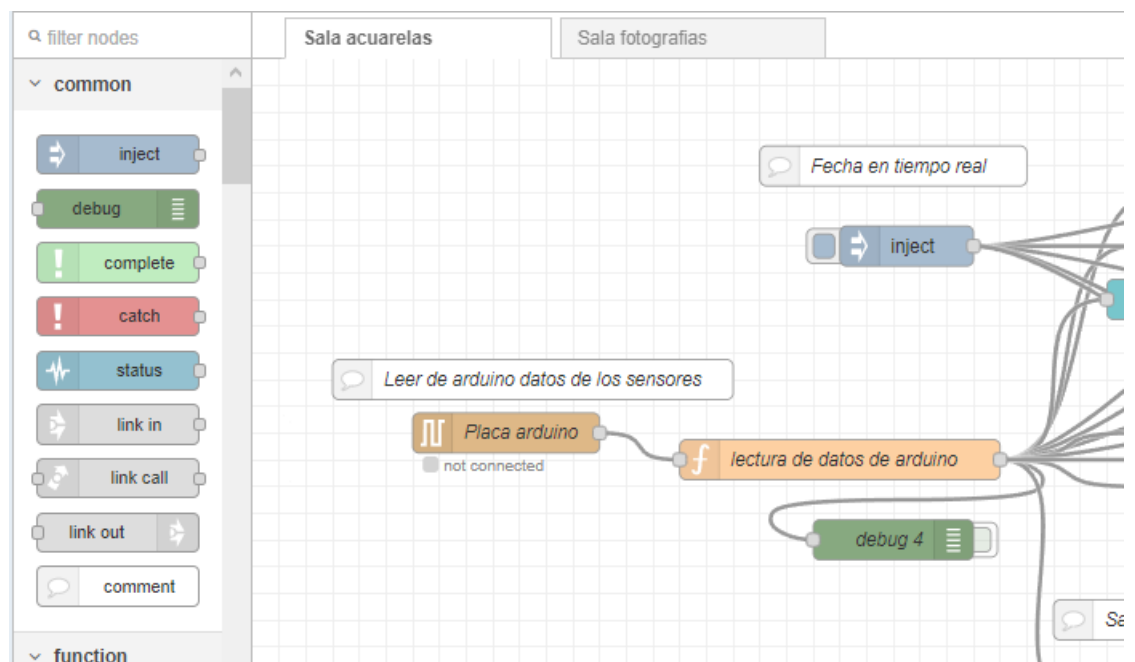


Figura 9.2 Editor de flujos de Node-RED

3. Panel de propiedades: Muestra las propiedades y configuraciones de los nodos seleccionados en el editor de flujos.

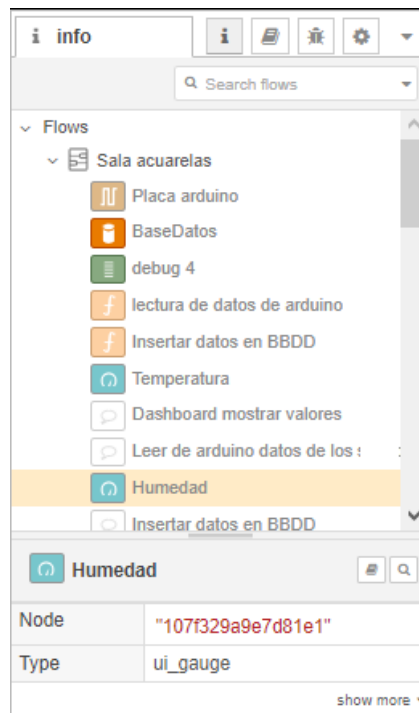


Figura 9.3 Panel de propiedades de Node-RED

9.1.3 Creación del flujo

Para poder crear un flujo en Node-RED debemos de seguir los siguientes pasos:

1. Arrastrar los nodos desde la paleta y soltarlo en el editor de flujos.
2. Conectar los nodos arrastrando un cable desde la salida de un nodo hasta la entrada de otro nodo.
3. Configurar los nodos seleccionándolos y estableciendo sus propiedades en el panel de propiedades.
4. Personaliza tu flujo agregando más nodos desde la paleta y conectándolos según las necesidades del programa.
5. Guardar el flujo haciendo click en “Guardar” en la barra de herramientas.

9.1.4 Ejecución y depuración del flujo

A continuación, se describen los pasos para poder desplegar el flujo de un programa.

1. Para ejecutar un flujo, debemos de hacer click en el botón “Desplegar” en la barra de herramientas. Esto activará el flujo y comenzará a procesar los datos.
2. Si se detectan errores o problemas en el flujo, Node-RED resaltará los nodos con problemas en rojo. Se puede hacer click en los nodos para ver más detalles sobre el error

3. Se pueden utilizar los nodos de depuración para inspeccionar los datos que fluyen a través del flujo. Se pueden agregar nodos de depuración y ver los mensajes de depuración en la pestaña “Debug” en la barra lateral derecha.

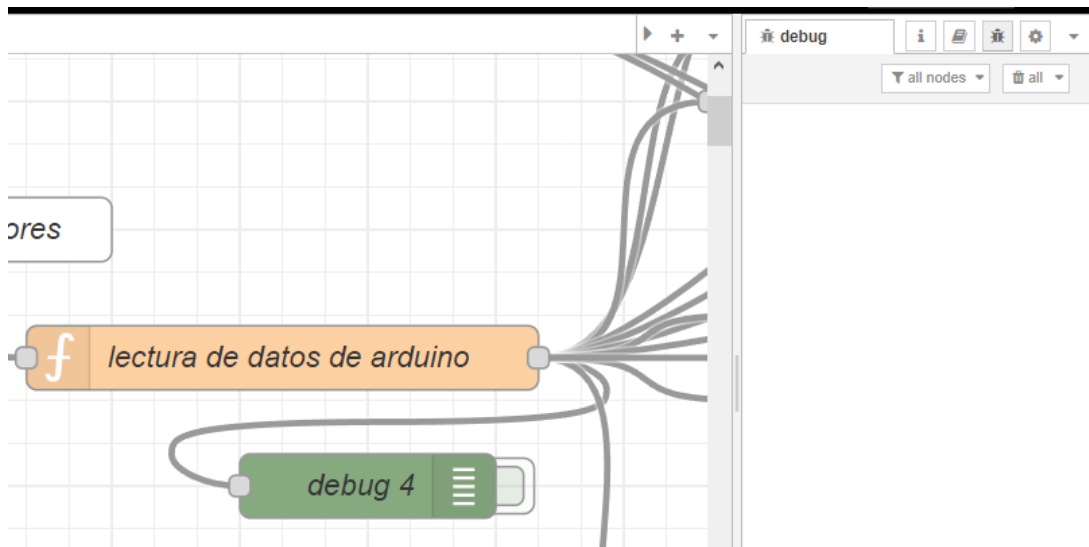


Figura 9.4 Nodos de depuración en Node-RED

9.1.5 Paleta de Nodos

Node-RED consta de una paleta de nodos con una amplia gama de nodos predefinidos y otros contribuidos por la comunidad que se pueden utilizar sin problemas en el flujo.

1. Explorar la paleta para buscar nodos específicos o utilizar la barra de búsqueda para encontrar nodos por nombre o función.

9.1.6 Importación y Exportación de flujos

Node-RED consta con la función de importar y exportar flujos en formato JSON para poder compartirlos o transferirlos entre instancias.

1. Para importar un flujo, hacer click en el menú de tres barras en la esquina superior derecha de la interfaz, selecciona “Import” y carga el archivo JSON del flujo.
2. Para exportar un flujo, hacer click en el menú de tres barras, seleccionar “Export” y elegir el formato de exportación deseado.

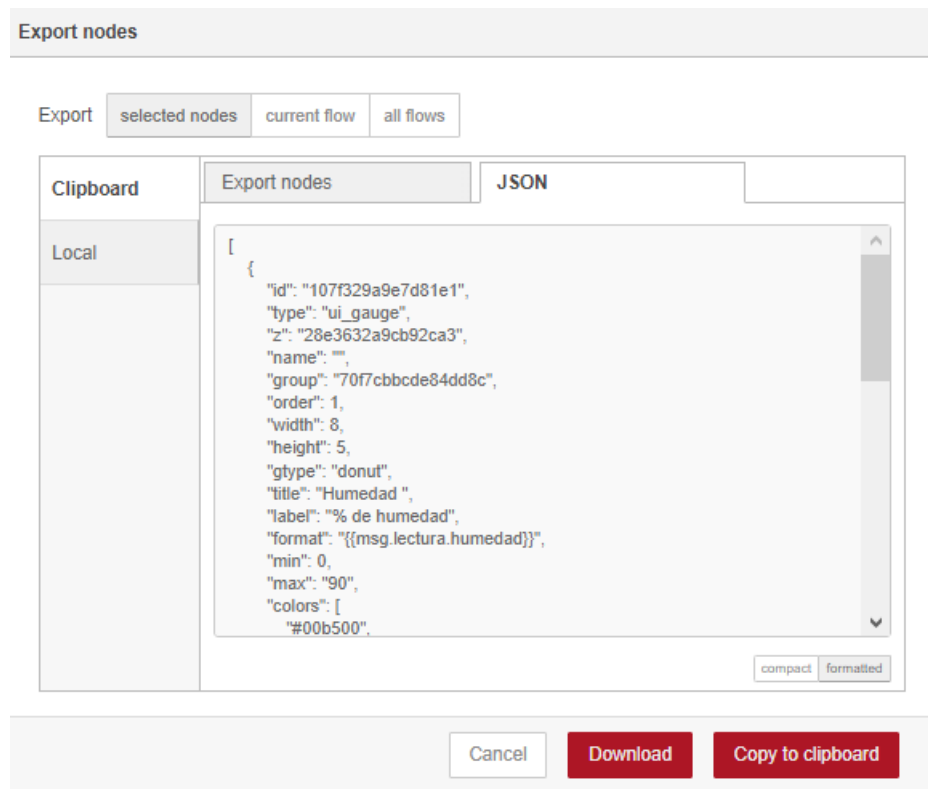


Figura 9.5 Exportar un flujo en Node-RED

Capítulo 10. Bibliografía

- [1] Herráez J.A. & Rodríguez Lorite, M.A. (1989). *Manual para el uso de aparatos y toma de datos de las condiciones ambientales en museos*. Ministerio de Cultura de Madrid.
- [2] Herráez J.A. & Rodríguez Lorite, M.A. (1999). La Conservación Preventiva de las Obras de Arte. *Arbor*. “Conservación del Patrimonio artístico”. Nº 164, 645.
- [3] Llull Peñalba, J. (2005). Evolución del concepto y de la significación social del patrimonio cultural. *Arte, Individuo y Sociedad*, 17, 175-204.
- [4] Macarrón Miguel, A.M. (2021). *Historia de la conservación y la restauración*. Tecnos.
- [5] (1987). *Damage to Museum Objects by Visible Radiation and Ultraviolet Radiation. Conference on Lighting in Museums, Galleries and Historic Houses. The Museums Association, United Kingdom Institute for Conservation and Group of Designers and Interpreters in Museums*. Bristol 9/10th april.
- [6] Ruiz de Lacanal, M.D. (1995). El Instituto de Conservación y Restauración de Bienes Culturales y la formación del conservador-restaurador. *Boletín del Instituto Andaluz del Patrimonio Histórico*, Año nº 3, Nº 10, 38-45.
- [7] Ruiz de Lacanal, M.D. (1995). El Instituto de Conservación y Restauración de Bienes Culturales y la formación del conservador-restaurador. *Boletín del Instituto Andaluz del Patrimonio Histórico*, Año nº 3, Nº 10, 38-45.
- [8] Ruiz de Lacanal, M.D. (1995). El Instituto de Conservación y Restauración de Bienes Culturales y la formación del conservador-restaurador. *Boletín del Instituto Andaluz del Patrimonio Histórico*, Año nº 3, Nº 10, 38-45.
- [9] (1994). *Learning from the History of Preventive Conservation. Preventive Conservation, Practice, Theory and Research*. IIC Ottawa Congress, 12-16 September 1994.
- [10] (1994). *Learning from the History of Preventive Conservation. Preventive Conservation, Practice, Theory and Research*. IIC Ottawa Congress, 12-16 September 1994.
- [11] Ruiz de Lacanal, M.D. (1995). El Instituto de Conservación y Restauración de Bienes Culturales y la formación del conservador-restaurador. *Boletín del Instituto Andaluz del Patrimonio Histórico*, Año nº 3, Nº 10, 38-45.
- [12] Ruiz de Lacanal, M.D. (1995). El Instituto de Conservación y Restauración de Bienes Culturales y la formación del conservador-restaurador. *Boletín del Instituto Andaluz del Patrimonio Histórico*, Año nº 3, Nº 10, 38-45.
- [13] Página oficial de Arduino. **Software Arduino**. URL: <https://www.arduino.cc/en/software>
- [14] Página oficial de Node-RED. **Node-RED**. URL: <https://nodered.org>
- [15] Rodríguez, J. (2021). ¿Qué es Node-RED y para qué sirve? URL: <https://blog.gruposinelec.com/actualidad/que-es-node-red-y-para-que-sirve/>
- [16] Aprendiendo Arduino. **Dashboard en Node-RED**. URL: <https://aprendiendoarduino.wordpress.com/2020/03/10/dashboard-en-node-red/>
- [17] Página oficial MQTT. **MQTT**. URL: <https://mqtt.org>
- [18] Página oficial Mosquitto. **Eclipse Mosquitto**. URL: <https://mosquitto.org>
- [18] Gonzalez Cortés, C. A. (2017). *Aprendiendo Electrónica*. Editorial: Prisma C.E.M.

[19] Curso impartido por Mecatrónica MADE. **Node-RED Curso básico I.** URL:
<https://www.youtube.com/watch?v=20RvO080yQ0&list=PLcmn4IteQPYqcZuzR7VLkQsFeAN0fPsPN>

[20] Curso impartido por Mecatrónica MADE. **Node-RED Curso básico II.** URL:
https://www.youtube.com/watch?v=AOgtD4kY_vY&list=PLcmn4IteQPYrMOJO6rCerzIcbaGYdUPz6

[21] Curso impartido por Mecatrónica MADE. **Node-RED Dashboard Básico I.** URL:
<https://www.youtube.com/watch?v=l9nx2regmr4&list=PLcmn4IteQPYo22Zqah1YEyafVImGdWm4b>