



universidad
de león



**Escuela de Ingenierías
Industrial, Informática y Aeroespacial**

**GRADO EN INGENIERÍA EN ELECTRÓNICA INDUSTRIAL
Y AUTOMÁTICA**

Trabajo de Fin de Grado

**SISTEMA DE MONITORIZACIÓN PARA GRANJA DE GANADO
OVINO**

MONITORING SYSTEM FOR SHEEP FARM

Autor: Víctor Canseco Pascual
Tutor: Ángela Díez Díez

(Julio, 2022)

UNIVERSIDAD DE LEÓN
Escuela de Ingenierías Industrial, Informática y
Aeroespacial

GRADO EN INGENIERÍA EN ELECTRÓNICA INDUSTRIAL Y
AUTOMÁTICA
Trabajo de Fin de Grado

ALUMNO: Víctor Canseco Pascual

TUTOR: Ángela Díez Díez

TÍTULO: Sistema de monitorización para granja de ganado ovino

TITLE: Monitoring system for sheep farm

CONVOCATORIA: Julio, 2022

RESUMEN:

Diseño y fabricación de una maqueta basada en una nave ya existente a la cual se desean instalar diferentes sensores para monitorizar parámetros de interés para el ganadero como son temperatura, humedad, amoníaco o ruido ambiente. Los datos instantáneos serán mostrados por pantalla para una rápida previsualización y luego serán enviados a una página web para hacer un registro de la evolución de dichos datos, además permitirá la consulta desde otro dispositivo u otra ubicación. Los valores también serán analizados para llevar a cabo otras acciones como abrir ventanas o activar ventiladores para regular temperaturas o por necesidad de renovar el aire; cortar el suministro de agua en caso de detectarse un caudal elevado en la tubería o activar una alarma visual y auditiva para hacer saber al usuario que ocurre algún problema. Para lograr los objetivos se investigará sobre las condiciones de bienestar animal y se hará un estudio de lo ya existente para lograrlo, se compararán las opciones ofrecidas para llevarlo a cabo de una manera más sencilla y de bajo coste.

ABSTRACT:

Design and production of a mockup based on an existent sheep farm in which many sensors would be installed to monitor different parameters of interest to the rancher like temperature, humidity, ammonia or ambient noise. Instant values would be shown on screen for a quick preview and later sent to a webpage, which allow us to register the evolution over time, this also could be used to consult the information on another device or in another location. Data would also be used to take other actions like opening the windows or trigger fans to regulate temperature or renovate the air, stop water supply if there is a high flow or activate a visual and auditory alarm to let the user know there is a problem. To achieve the objectives information about animal welfare will be investigated. Also a study about what is installed nowadays in farms is needed and what could be used in the project to make it simple and low cost.

Palabras clave: Monitorización, control, sensores, prototipo, ovino.

Firma del alumno:

VºBº Tutor/es:

Índices

ÍNDICE DE CONTENIDO

1	Memoria	9
1.1	Antecedentes.....	9
1.2	Objetivo	9
1.3	Requisitos del sistema	9
1.3.1	Condiciones de bienestar	9
1.3.2	Requisitos de operatividad.....	15
1.4	Alcance	17
1.5	Análisis de lo existente	19
1.5.1	Ámbito industrial.....	19
1.5.2	Prototipo	22
1.6	Descripción del sistema	28
1.6.1	Hardware	28
1.6.1.1	Sensores	29
1.6.1.2	Actuadores	34
1.6.1.3	HMI	38
1.6.1.4	Controlador.....	41
1.6.1.5	Alimentación	44
1.6.2	Software.....	45
1.6.2.1	Entorno de programación	45
1.6.2.2	Diseño.....	45
1.7	Instalación existente	46
2	Pruebas y calibraciones	47
3	Diseño de la maqueta.....	56
4	Mejoras y conclusiones	69
5	Normativa y referencias	71

5.1 Bibliografía.....	71
5.2 Normativa.....	79
5.3 Glosario de términos.....	79
6 ANEXOS.....	82
6.1 Anexo I: Hojas de datos	82
6.1.1 Hojas de datos de sensores	82
6.1.2 Hojas de datos de actuadores.....	85
6.1.3 Hojas de datos de HMI.....	87
6.1.4 Hojas de datos de procesador	88
6.1.5 Hojas de datos de alimentación	89
6.2 Anexo II: Planos	90
6.2.1 Planos de la instalación.....	90
6.2.2 Planos piezas 3D.....	93
6.3 Anexo III: Flujogramas	94
6.4 Anexo IV: Programación.....	105
6.4.1 Código principal	105
6.4.2 Códigos de error	123

ÍNDICE DE FIGURAS

Figura 1.1: Relación de la productividad con la calidad del aire (Fuente: [6]).	12
Figura 1.2: Efectos negativos en función de la concentración de NH ₃ en ppm (Fuente: [6]).	13
Figura 1.3: Emisiones de amoniaco por sectores (Fuente: [9]).	13
Figura 1.4: Equivalencia nivel dB con ejemplo real (Fuente: [11]).	14
Figura 1.5: Temperaturas máximas y mínimas en León (Fuente: [12]).	15
Figura 1.6: Datos temperatura en Valle de Mansilla a fecha 31/12/2021, hora local 12:00.	16
Figura 1.7: Grados de protección IP (Fuente: [18]).	17
Figura 1.8: Visualización de datos dol-sensors (Fuente: [25]).	20
Figura 1.9: Comparativa modelos Arduino (Fuente: [47]).	28
Figura 1.10: Conexiones DHT22	30
Figura 1.11: Conexiones MQ-137	31
Figura 1.12: Conexiones MQ-135	32
Figura 1.13: Conexiones MAX9814	33
Figura 1.14: Funcionamiento del sensor de efecto Hall (Fuente: [54]).	33
Figura 1.15: Conexiones YF-S201	34
Figura 1.16: L298N sentido de giro (Fuente: [57]).	35
Figura 1.17: Puente-H (Fuente: [57]).	35
Figura 1.18: Conexiones driver L298N	36
Figura 1.19: Conexiones SG90	37
Figura 1.20: Conexión Buzzer	37
Figura 1.21: Conexión relé	38
Figura 1.22: Conexiones LDC y controlador I2C	39
Figura 1.23: Pines del ESP8266-ESP01 (Fuente: [68]).	40
Figura 1.24: Conexiones ESP8266-ESP01	41
Figura 1.25: Pines Arduino nano (Fuente: [70]).	42
Figura 1.26: Pines Arduino nano, detallado (Fuente: [71]).	43
Figura 1.27: Conexiones Arduino nano	44
Figura 2.1: Plotter voltios salida MAX9814	48
Figura 2.2: Gráfico relación Ruido-Voltaje para MAX9814	49
Figura 2.3: Calibración YF-S201	50
Figura 2.4: Ejemplo de cómo se presentan los datos en WebPlotDigitalizer (Fuente: [84]).	51
Figura 2.5: Prueba de envío de datos a ThingSpeak, se muestran temperatura y humedad con dos tipos de visualizaciones.	54
Figura 2.6: Comandos AT para ESP8266 del ejemplo (Fuente: [90]).	55
Figura 3.1: Piezas principales en su posición correspondiente antes del pegado.	57

Figura 3.2: Imagen cenital de la maqueta.....	58
Figura 3.3: Servomotor cerrado y ventanas.....	59
Figura 3.4: Sensores de gases y temperatura y humedad (con ampliación).....	59
Figura 3.5: Electroválvula y caudalímetro.....	60
Figura 3.6: Detalle de los elementos en la protoboard.....	60
Figura 3.7: Inicio del programa visto desde el monitor serial.....	61
Figura 3.8: Monitor serial, apertura de ventanas.....	61
Figura 3.9: Servomotor abriendo las ventanas.....	62
Figura 3.10: Monitor serial, apertura de ventanas e inicio del primer motor al 50%.....	62
Figura 3.11: Apreciación de los ventiladores.....	63
Figura 3.12: Datos mostrados en la LCD durante el funcionamiento.....	63
Figura 3.13: Datos enviados a ThingSpeak.....	64
Figura 3.14: Widgets extra ThingSpeak.....	65
Figura 3.15: Mensaje de error y led de alarma encendido.....	66
Figura 3.16: PCB vista en el programa.....	67
Figura 3.17: Formato de salida en capas de la PCB.....	67
Figura 4.1: Diagrama de Gantt del proyecto.....	70
Figura 5.1: Regulación del ancho de pulso (Fuente: [56]).....	80
Figura 6.1: Flujograma de medición de temperatura.....	94
Figura 6.2: Flujograma de medición de humedad.....	95
Figura 6.3: Flujograma de medición de amoniaco.....	96
Figura 6.4: Flujograma de medición de dióxido de carbono.....	97
Figura 6.5: Flujograma función aireación.....	98
Figura 6.6: Flujograma función ventilación. Parte 1.....	99
Figura 6.7: Flujograma función ventilación. Parte 2.....	100
Figura 6.8: Flujograma de medición de caudal y corte del suministro.....	101
Figura 6.9: Flujograma de medición de ruido ambiente.....	102
Figura 6.10: Flujograma función imprimir.....	103
Figura 6.11: Flujograma función ThingSpeak (mandar datos a la web).....	103
Figura 6.12: Flujograma de funcionamiento general.....	104

ÍNDICE DE TABLAS

Tabla 1.1: TIH para los efectos del estrés térmico en rumiantes lecheros (Fuente: [3]).	11
Tabla 1.2: Estrés térmico según el ritmo respiratorio (en respiraciones por minuto) (Fuente: [3]).	11
Tabla 1.3: Comparativa modelos DOL (Fuente: [18]).	19
Tabla 1.4: Comparación de los principales servomotores (Fuente:[45]).	26
Tabla 1.5: Pines necesarios.	27
Tabla 2.1: Tabla relación Ruido-Voltaje para MAX9814	48
Tabla 2.2: Valor de Rs obtenido a partir de la gráfica del MQ-137.	51
Tabla 2.3: Valor de Rs obtenido a partir de la gráfica del MQ-135.	52
Tabla 3.1: Dimensiones de las piezas principales que conforman la nave.	56
Tabla 3.2: Consumos aproximados	58
Tabla 3.3: Precio componentes del sistema.	68
Tabla 6.1: Sensor de temperatura y humedad (Fuente: [48]).	82
Tabla 6.2: Sensor de amoniaco (Fuente: [49]).	83
Tabla 6.3: Sensor calidad aire (Fuente: [50]).	83
Tabla 6.4: Sensor de Ruido ambiente (Fuente: [51]).	84
Tabla 6.5: Sensor caudal (Fuente: [52]).	84
Tabla 6.6: Driver motores (Fuente: [55]).	85
Tabla 6.7: Ventilador (Fuente: [58]).	85
Tabla 6.8: Servomotor (Fuente: [59]).	86
Tabla 6.9: Buzzer (Fuente: [60]).	86
Tabla 6.10: Pantalla LCD y controlador I2C (Fuentes: [63] [64]).	87
Tabla 6.11: ESP8266 (Fuente: [68]).	88
Tabla 6.12: Arduino nano (Fuente: [70]).	88
Tabla 6.13: Conversor voltaje para protoboard (Fuente: [71]).	89
Tabla 6.14: Regulador de tensión de 5V (Fuente: [72]).	89
Tabla 6.15: Códigos de error en el programa	123

1 Memoria

1.1 Antecedentes

Con la mejora de las tecnologías y las nuevas recomendaciones de bienestar animal se busca conseguir mejoras en la salud del ganado y en el resultado del producto final que se puede ofrecer al consumidor. Es por ello que entran en juego sistemas para medir y controlar las variables tanto del entorno como en algunas ocasiones del propio animal.

1.2 Objetivo

El objetivo que se busca en este proyecto es la creación de una maqueta y prototipo capaces de:

- Representar a escala (reducida) 1:50 la nave ya dispuesta.
- Monitorizar distintas variables clave en este tipo de entornos y poder mostrarlas a un tercer usuario mediante un display y/o página web.
- Realizar el control sobre accionamientos en función de las variables medidas, preferiblemente de forma automática para permitir que el usuario final sea prescindible en tareas livianas.
- Avisar al usuario en caso de que el sistema no pueda llevar a cabo sus funciones ya sea por un error en los distintos elementos o porque no pueda controlar sus salidas.

Con ello el ganadero tendrá la información de su instalación de una manera rápida y visual, además de no tener que estar controlando manualmente las variables. Los animales tendrán un mayor nivel de confort, beneficiando su bienestar general y su producción.

1.3 Requisitos del sistema

1.3.1 Condiciones de bienestar

Para llevar a cabo este proyecto cuyo fin es la mejora del bienestar animal primero se debe definir este término:

“Un animal está en buenas condiciones de bienestar si está sano, cómodo, bien alimentado, en seguridad, puede expresar formas innatas de comportamiento y si no padece sensaciones desagradables de dolor, miedo o desasosiego”. [1]

En base a lo que en la definición se cita y, siendo un proyecto consistente en la monitorización y realización de un pequeño control de manera automática sobre el recinto, se puede comprobar que afecta a los términos de comodidad y desasosiego.

Una vez ya enmarcado el ámbito del proyecto deben definirse las condiciones en que el animal, en nuestro caso ovino adulto (mayor a 6 meses), se debe encontrar para cumplir con los estándares de comodidad [2]:

- Espacio, acceso al alimento y al agua: los metros disponibles por animal, además del número de comederos y bebederos y su localización se encuentra controlado ya por el propio ganadero y no siendo objeto del proyecto.
- Temperatura óptima de confort: se encuentra entre 10 y 15°C, siendo los extremos de temperatura a evitar los inferiores a -8 y superiores a 30°C. Valores fuera del rango de confort acarrearán estrés térmico, problemas de ingesta, baja sociabilidad y disputas con otros individuos y problemas de salud de diferente severidad, llegando incluso a la muerte del animal.
- Humedad: aunque en la página no se cita ningún rango de confort su relación con la temperatura es importante a la hora de mantener la comodidad en el ambiente.
- Ventilación: se recomiendan velocidades entre 0.5 a 1.8m/s, necesario para la regulación de temperatura y humedad, así como la expulsión de los gases del desarrollo de los animales (principalmente metano y amoníaco) y la introducción de aire limpio y oxigenado.
- Luz solar: regula el ciclo de los animales, tampoco es objeto del proyecto.
- Niveles de ruido: la maquinaria, el sistema de ventilación y los distintos trabajos realizados pueden alterar al ganado y producir problemas de manera parecida a la temperatura. Se recomienda mantener unos niveles de ruido no elevados.

Ampliando los apartados de temperatura y humedad se encuentra un estudio cuyo fin es la mejora del bienestar animal, llevado a cabo entre la Universidad Autónoma de Barcelona y distintas granjas de Castilla y León [92], en base a otros estudios ya realizados previamente en el sector, se procedió a evaluar los perjuicios ocasionados por calor. Para ello se hace uso del índice temperatura-humedad dado por la fórmula $THI = 0.8 * T + RH * (T - 14.41) + 46.4$ [3], siendo T la temperatura en °C y RH la humedad relativa en %. Con los resultados obtenidos se ha llegado a la tabla siguiente:

Tabla 1.1: TIH para los efectos del estrés térmico en rumiantes lecheros (Fuente: [3]).

		Temperatura (°C)																						
		0	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44
Humedad relativa (%)	10	44	46	48	49	51	53	55	57	58	60	62	64	66	67	69	71	73	75	76	78	80	82	84
	20	43	45	47	49	51	53	55	57	59	61	63	65	67	69	71	73	75	77	79	81	83	85	87
	30	41	43	45	48	50	52	54	56	59	61	63	65	67	70	72	74	76	78	81	83	85	87	89
	40	40	42	44	47	49	52	54	56	59	61	64	66	68	71	73	76	78	80	83	85	88	90	92
	50	38	41	43	46	49	51	54	56	59	62	64	67	69	72	75	77	80	82	85	88	90	93	95
	60	37	40	42	45	48	51	54	56	59	62	65	68	70	73	76	79	82	84	87	90	93	96	98
	70	35	38	41	44	47	50	53	56	59	62	65	68	71	74	77	80	83	86	89	92	95	98	101
	80	34	37	40	43	47	50	53	56	59	63	66	69	72	75	79	82	85	88	91	95	98	101	
	90	32	36	39	43	46	49	53	56	60	63	66	70	73	77	80	83	87	90	94	97	100		
	100	31	35	38	42	45	49	53	56	60	63	67	71	74	78	81	85	89	92	96	99			

Termoneutralidad
 Efectos leves
 Efectos moderados
 Efectos severos
 Fatal

Se puede observar que los valores a conseguir son los indicados bajo el color verde y cuyo límite se encuentra en THI=67, es decir la zona de termoneutralidad. El sistema a desarrollar debe ser capaz de realizar las mediciones y actuar para conseguir mantener el ambiente en estas condiciones, mediante la ventilación en mayor o menor medida. Para valores superiores ya entramos en las zonas que pueden producir efectos adversos y que, cifras superiores a THI=90, producen efectos de carácter fatídico.

Esta tabla, elaborada más en base al vacuno lechero, puede ser extrapolada hacia los sectores ovino o caprino sin mucho problema, teniendo en consideración que las diferentes razas dentro de cada grupo pueden presentar mayor o menor resistencia frente al calor. Es por ello que esta tabla debe ser tomada como referencia.

En el estudio también se indica que el estrés por las temperaturas puede ser cuantificado en función del ritmo respiratorio. Se muestra tabla como ampliación de la anterior y para mostrar distintos métodos para las mediciones, sin efectos sobre el proyecto a realizar.

Tabla 1.2: Estrés térmico según el ritmo respiratorio (en respiraciones por minuto) (Fuente: [3]).

Nivel		Vacuno	Caprino	Ovino
Basal		20	25-30	
Normalidad		<40		
Estrés	Leve	40-60		
	Moderado	60-80		
	Severo	80-120		
Fatal		>150	>200	

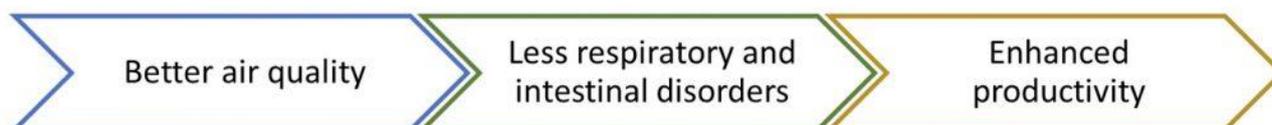
En el apartado de ventilación se indica su función reguladora de temperatura y humedad y su función de renovación del aire, expulsando gases nocivos. Uno de los gases más conocidos es el metano (CH_4) por su implicación en el efecto invernadero, sin embargo, resulta más interesante el estudio del amoníaco (NH_3), también procedente del proceso digestivo. Comparando rápidamente ambos gases:

- El metano es incoloro e inodoro y no es tóxico, aunque puede ser asfixiante al desplazar el oxígeno del ambiente, es un gas extremadamente inflamable [4].
- El amoníaco es también incoloro, pero con un olor muy penetrante. Resulta corrosivo y reacciona violentamente con ácidos, oxidantes fuertes y halógenos. Es un gas tóxico, en altas concentraciones puede producir quemaduras y daños en la piel y el sistema respiratorio. En concentraciones menores causa irritación [5].

En vista de que el amoníaco resulta ser un gas que puede producir molestias en bajas concentraciones y empeore significativamente la calidad del aire será en el cual más se centre el proyecto y sobre el que se produzca de manera directa la acción de ventilar, resultando los otros gases expulsados también.

Sus efectos en los animales de la instalación son mayores al tratarse de su lugar de estancia y estando expuestos de manera continuada, destacando sus efectos sobre las mucosas y los sistemas inmunitario y digestivo en los que se reducen las defensas y la absorción de nutrientes respectivamente [6]. Como es de esperar, todo ello provoca una disminución del rendimiento y producción de la granja.

Figura 1.1: Relación de la productividad con la calidad del aire (Fuente: [6]).



Se recomienda que las cantidades de amoníaco en el aire, medidas normalmente en ppm (partes por millón) y a nivel de suelo considerando la altura del ganado, sean muestreadas en varios puntos y de forma recurrente a lo largo del día, siendo cuidadosos en la mañana ya que la ventilación en las horas nocturnas suele ser escasa coincidiendo con las temperaturas más frescas y agravándose en los meses invernales [6].

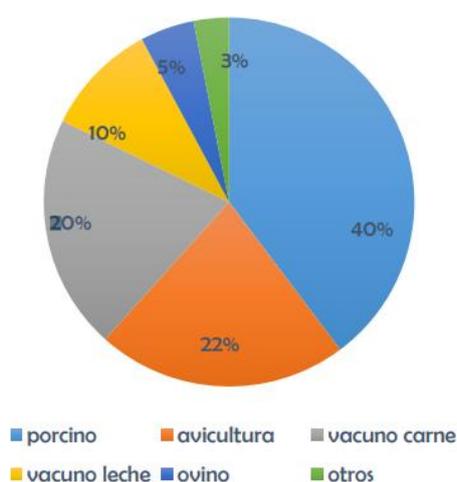
Figura 1.2: Efectos negativos en función de la concentración de NH₃ en ppm (Fuente: [6]).

- 10 ppm – lesions in the pulmonary surface
- 20 ppm – higher susceptibility to respiratory diseases (odour can be detected by humans)
- 50 ppm – reduced growth by 5%
- 100 ppm – ocular and respiratory problems + reduced growth by 15%

Se comprueba que por encima de 10ppm ya se pueden producir lesiones y que por encima de 20ppm ya puede ser detectado por el olfato humano. La normativa que se puede encontrar para estos casos es escasa, en el BOE 2007/43/CE sólo se refleja en instalaciones avícolas en las cuales se debe conseguir que la concentración no sea superior a las 20ppm [7]. Idealmente se deben buscar unos niveles casi imperceptibles para lograr las mejores condiciones de bienestar, de manera más realista se intentará no llegar al límite de producción de lesiones.

La mitigación de la concentración de amoniaco en granjas de ovino puede ser llevada a cabo mediante la sola ventilación sin necesidad de introducir cambios en la alimentación o modificaciones del alojamiento como sí se están aplicando en otras especies ganaderas. Esto es debido al porcentaje de emisiones a la atmósfera que representa el sector ovino en comparación al resto de sectores y su previsión de futuro [8] [9].

Figura 1.3: Emisiones de amoniaco por sectores (Fuente: [9]).



Es importante no olvidar que el aire cuenta con más gases en su composición, cuyas concentraciones determinarán la calidad del mismo, y su renovación resulta imprescindible para aportar oxígeno y mantener el confort.

Para finalizar contamos con los niveles de ruido ambiente. En la web de Oviespaña [2] se menciona que por un aumento de 45dB a 90dB se redujeron los tiempos de ingesta, se produjo un incremento del comportamiento inactivo de las ovejas y se mostraron signos de sufrimiento. Si bien parece que el valor inicial fue duplicado se debe recordar que está indicado en decibelios y su relación no es lineal sino logarítmica, es por ello que el clásico ejemplo que se realiza es que la potencia se duplica al pasar de 3dB a 6dB.

Para realizar esta comparación se va a usar una calculadora online que convierta la cifra en dB a su relación equivalente en potencia [10]. Empezando por el ejemplo vemos que 3db equivalen a 2mW y que para 6dB serían 4mW. Si introducimos 45dB como resultado nos aparece 31.623W y para el valor de 90dB su valor se eleva hasta 1 000 000W o 1MW. Si se realiza la división entre ambos valores se comprueba que el valor inicial ha sido multiplicado por 31 622.55. Esto nos indica que, aunque el aumento de los decibelios pueda no parecer gran cosa, debe tenerse precaución con lo que realmente los datos están mostrando.

Figura 1.4: Equivalencia nivel dB con ejemplo real (Fuente: [11]).



En la Figura 1.4 se puede observar que el nivel de ruido recomendado por la Organización Mundial de la Salud está en 55dB, situado en un ambiente poco ruidoso. Para el nivel de ruido de nuestras instalaciones lo más normal es que nos encontremos en el nivel superior correspondiente a un ambiente ruidoso como es el caso durante las distintas tareas a realizar en el día y que comprenden los horarios de ordeño, el momento de reposición de los comederos y otras tareas de movimiento del ganado. En algunas otras ocasiones, podremos esperar que el nivel se eleve hasta un ambiente molesto, eso sin contar cuando se realicen tareas con maquinaria agrícola.

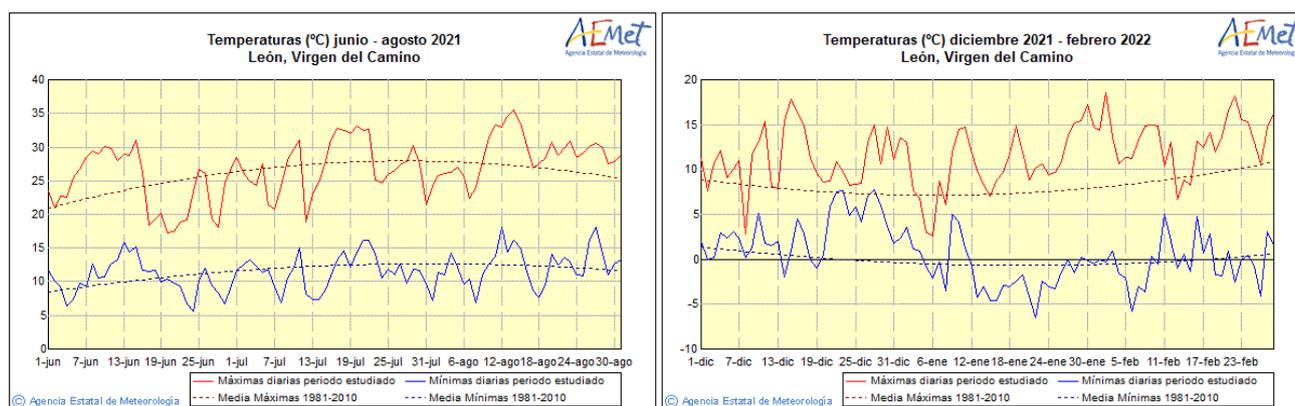
Todas estas recomendaciones no son sólo aplicables para los animales, en las instalaciones muchas veces también se encontrarán trabajadores que deberán estar en un ambiente laboral adecuado en el que, aunque las horas que allí se encuentren no sean tantas, su estado de salud no se vea comprometido y, de igual manera, se encuentren en un ambiente agradable y cómodo.

1.3.2 Requisitos de operatividad

Por último, se deberán analizar las condiciones climáticas de la zona en que se encuentran las instalaciones y que determinarán las condiciones de trabajo de los aparatos a instalar, permitiendo cierto margen al ser instalados posteriormente bajo techo y en un ambiente normalmente más cálido que el exterior.

Se toman de referencia dos páginas web, los primeros datos provienen de la Agencia Estatal de Meteorología (AEMet) [12] contando el verano 2021 para las temperaturas máximas y el invierno 2021/2022 para las mínimas.

Figura 1.5: Temperaturas máximas y mínimas en León (Fuente: [12]).

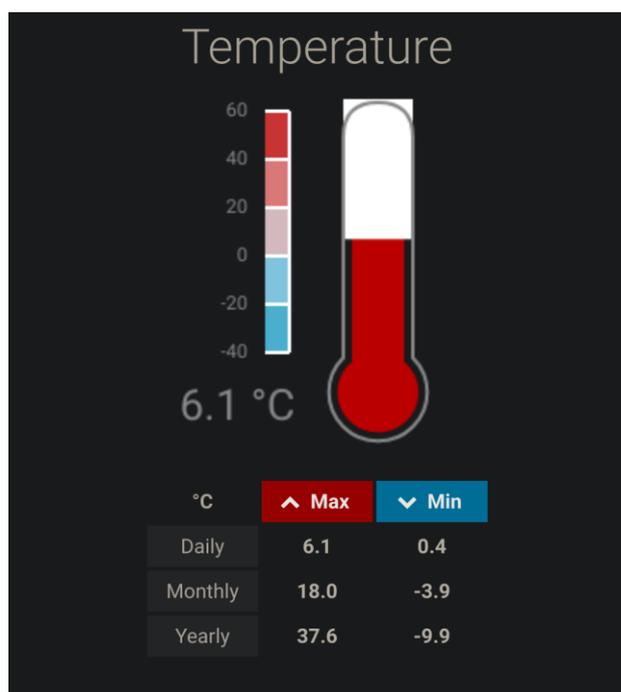


En la Figura 1.5 se puede observar que la máxima contabilizada se sitúa en agosto y su valor es ligeramente superior a 35°C mientras que la mínima ocurre en enero y es inferior a -5°C.

Los segundos datos se toman de la página Weather Spark [13]. En el transcurso del año la temperatura sufre variaciones entre los 0°C y los 27°C y rara vez es inferior a -4°C o superior a 32°C, situación que, como se ve en el párrafo anterior, sí pueden ocurrir.

Si tomamos otra referencia, en este caso los datos de una estación meteorológica en la propia localidad que son publicados en la página web de Weathercloud [14] y cuyos registros pueden consultarse para el día, mes y año actual.

Figura 1.6: Datos temperatura en Valle de Mansilla a fecha 31/12/2021, hora local 12:00



Para el año 2021 las mediciones de temperatura máxima fueron de 37.6°C, valor bastante aproximado a lo ofrecido en las gráficas de AEMet de la *Figura 1.5* para la ciudad de León. En cuanto a las temperaturas mínimas tomadas en León eran ligeramente menores a -5°C, mientras que en lo medido en Valle de Mansilla casi se alcanzan los -10°C.

En base a los datos proporcionados para el año 2021 y tomando en cuenta las vivencias en el pueblo para años anteriores en las que la temperatura ha llegado a disminuir de -13°C se determina: el rango operativo de los dispositivos a instalar deberá ser igual o superior a los valores de -15°C y de 40°C.

Otra de las adversidades a las que se debe hacer frente es al polvo en suspensión que se puede encontrar en el ambiente y que puede estropear los dispositivos a colocar. Es importante a la hora de elegir en caso de hacer una instalación real que los equipos cuenten con grado de protección IP [15] según la norma UNE-EN 60529:2018 [17].

Figura 1.7: Grados de protección IP (Fuente: [18]).

PROTECCIÓN ANTE CUERPOS SÓLIDOS			PROTECCIÓN CONTRA EL AGUA	
	Sin protección	0	Sin protección	
	Protección contra objetos con diámetro superior a 50mm	1	Protección ante un goteo vertical	
	Protección contra objetos con diámetro superior a 12mm	2	Protección contra goteo con inclinación de 15°	
	Protección contra objetos con diámetro superior a 2.5mm	3	Protección ante pulverización	
	Protección contra objetos con diámetro superior a 1mm	4	Protección ante salpicaduras	
	Protección ante el polvo	5	Protección ante chorros de agua	
	Protección totalmente estanco ante el polvo	6	Protección ante chorros continuos de agua	
		7	Protección contra inmersiones temporales	
		8	Protección contra inmersiones permanentes	

En el caso que nos ocupa buscaremos aquellos cuyo índice sea superior al IP5X para evitar el polvo y siendo recomendable que posea también nivel de protección frente a las salpicaduras, resultando en índices superiores al IP54.

1.4 Alcance

Para lograr los objetivos contaremos con diversos sensores para medir las siguientes variables:

- Temperatura: el ganado deberá encontrarse en un ambiente tal que su metabolismo sea el adecuado y su producción y calidad de vida no se vean reducidas, así como evitar llegar a temperaturas extremas que pueden resultar fatales (como se aprecia en la *Tabla 1*). Las medidas deberán realizarse de una manera continuada, pero sin necesidad de ser intervalos muy cortos ya que no conlleva un peligro inminente. A ser posible se tomará la medida en varios puntos para poder comparar las distintas zonas de la instalación. En función de lo medido se deberán aplicar soluciones de ventilación para mitigar sus efectos.

- Humedad: en conjunto con la temperatura y como se ve en la *Tabla 1* será necesario mantener los niveles en un rango de confort. Se aplica el mismo criterio de medida que para la temperatura.
- Amoníaco (NH₃): como se ha visto constituye un importante indicador de la calidad del lugar, unos niveles elevados pueden advertir de faltas nutricionales en la dieta o un inadecuado estado de la cama de los animales, necesaria ventilación e incluso limpieza realizada por el ganadero para no acarrear problemas mayores. Deberán ser medidas recurrentes, preferiblemente a nivel de suelo y en varios puntos de las instalaciones.
- Calidad del aire: se usará un sensor de distintos tipos de gases, aunque también sea capaz de medir el amoníaco lo hará de manera menos específica ya que tendremos un sensor destinado a ello y más en conjunto con los demás gases para evaluar el ambiente general del recinto. Al ser un sensor de gases sus criterios de instalación serán iguales a los del amoníaco
- Ruido ambiente: se medirá el nivel de ruido, además de producir estrés en los animales y molestias a los trabajadores si allí se encuentran, un ambiente bullicioso es también indicador de algún problema dentro del rebaño que deberá ser solventado por el ganadero.
- Caudal: el control sobre el abastecimiento de agua suministrado permitirá la detección de fugas o roturas en el suministro o bebederos, evitando la anegación de la zona afectada y la incomodidad del ganado. También permite reducir el gasto de agua que se podría producir si el ganadero no observase pronto la fuga y los posteriores trabajos de limpieza.

Con la información obtenida por los sensores y tras ser procesada gestionaremos los actuadores oportunos, que en nuestro caso serían:

- Motor: su función será la de abrir o cerrar las ventanas o aperturas en la nave para controlar la ventilación de manera pasiva.
- Ventiladores: se encargarán de mover el aire de manera forzada cuando la apertura de las ventanas no sea suficiente, permitiendo controlar los niveles de temperatura y humedad y renovar el aire de la estancia.
- Válvula: realizará el corte al suministro de agua en caso de que el caudal medido sea más elevado de lo normal causado por fugas y/o roturas.
- Alarma y luz de emergencia: ante un error en el programa, la imposibilidad de que el sistema pueda llevar a cabo sus funciones o alguna otra anomalía se notificará al usuario mediante una señal acústica y/o visual.

La información podrá ser consultada por el usuario en:

- Pantalla: mostrará in situ los valores instantáneos ya procesados de los sensores, así como información de relevancia tales como los códigos de error.
- Página web: pudiendo ser visualizada desde otro lugar y dispositivo proporcionará gráficas sobre la evolución de las variables al ir recopilando los datos.

1.5 Análisis de lo existente

Se estudiarán algunos sensores ya usados actualmente y se compararán sus ventajas y desventajas, al final se verá también lo disponible para realizar el proyecto a una escala menor.

1.5.1 Ámbito industrial

Se realiza una búsqueda rápida acerca de sensores usados en granjas, uno de los resultados arrojados es la página de dol-sensors [18] en la que se encuentran diferentes modelos de sensores, ya diseñados y contruidos pensando en instalaciones con animales y que, por tanto, deberían cumplir con las necesidades del alcance, se resumen en la tabla siguiente los datos más relevantes para su comparación:

Tabla 1.3: Comparativa modelos DOL (Fuente: [18]).

Modelo	Medida de:	Rango de medida	Rango operativo	Grado de protección	Referencia
DOL 15	Temperatura	-30 a 60°C	-30 a 60°C	IP54	[19]
DOL 104	RH	0-100%RH	-40 a 60°C	IP67	[20]
DOL 114	Temperatura	-40 a 60°C	-40 a 60°C	IP68	[21]
	RH	0-100%RH			
DOL 19	CO2	0-10 000 ppm	-20 a 60°C	IP54	[22]
DOL 53	NH3	0-100 ppm	0 a 50°C	IP65	[23]

*Al momento de elaborar la tabla la web oficial no se encontraba disponible, los datos recabados son de otra página proveedora.

Las especificaciones se ajustan a los criterios que se habían considerado necesarios para el grado de protección y las condiciones térmicas de operatividad, a excepción del sensor DOL 53, que no trabaja en temperaturas negativas. Estos sensores necesitan de un procesador al que conectarse y que pueda interpretar los datos que le llegan; afortunadamente los sensores ofrecen varias señales de salida tanto en voltaje como en corriente, facilitando su conexionado.

Para estos sensores mencionados sería necesario contar con una unidad de procesamiento que nos permita la comunicación con los sensores, interpretar los datos que éstos le envían y poder así tratarlos, enviarlos o mostrarlos. En el caso de los sensores DOL contamos con el iDOL 63 [24], una caja que nos permite alimentar los sensores, recibir sus datos y enviarlos por la red mediante cable ethernet del cual también se alimenta a 48V. Los datos pueden ser consultados mediante la app de OmniFarm [25], que a su vez permite conectar varios iDOL 63, cada uno con sus sensores. La app permite una lectura

del valor actual de temperatura, humedad y CO2 en su versión básica, así como una alarma en la temperatura. En su versión extendida permite ya todos los sensores como el de NH3 y un mayor ajuste de las alarmas, además permite ver el historial de registros de las últimas 24h mediante una gráfica.

Figura 1.8: Visualización de datos dol-sensors (Fuente: [25]).



En este caso se ha visto que los equipos van cableados, estos presentan ventajas y desventajas frente a otros sistemas que hacen uso de la tecnología inalámbrica y que pueden, por ejemplo, sufrir interferencias. Los sistemas cableados pueden resultar ineficientes si el cable no es instalado correctamente, en las naves pueden ser susceptibles de ser mordidos por el ganado o pisados por maquinaria.

Otro punto importante es el de su modularidad, los sistemas modulares permiten que su instalación se lleve a cabo de manera más rápida, eligiendo lo que el cliente necesite, permitiendo colocar cada elemento en el lugar deseado y pudiendo expandirse en un futuro. Esto también permite sustituir las partes que hayan podido sufrir una rotura o la posibilidad de reemplazarse por un modelo más moderno.

También encontramos la empresa EnviraIoT y su sistema Nanoenvi® FARM [26] que según su propia definición es “una solución completa para la monitorización de las condiciones ambientales en las granjas, facilitando un mayor crecimiento saludable de los animales, reduciendo los riesgos de enfermedades y mejorando, con ello, la calidad de los productos y el rendimiento global de la explotación”. Se destaca dentro de esta definición: monitorización, crecimiento saludable, mejora de calidad y rendimiento; palabras clave para este proyecto.

Su solución se plantea como una red de sensores distribuidos por las instalaciones en puntos relevantes, algunos de estos sensores incluyen los que se han elegido como importantes, tal es el caso de los de temperatura y humedad, concentraciones de gases potencialmente peligrosos, hidrocarburos y bencenos, ruido, etc. Para su aislamiento en estos entornos cuentan con IP66. Su sistema de comunicación se basa en redes inalámbricas y los datos son gestionados por su plataforma IoT [27] donde pueden ser visualizados.

Los dos sistemas que se han mencionado anteriormente se han especializado para introducirse en explotaciones ganaderas, aunque generalmente son explotaciones avícolas o porcinas las que montan estos dispositivos. Su uso se encuentra en auge en los sectores vacuno y ovino, aunque en bastante menor medida para este último. Es por ello que se encuentra poca información al respecto.

Si nos alejamos de las búsquedas en específico para el sector ovino, podemos encontrar más opciones y variedad en el sector industrial, como es el caso de Monnit y su gama de productos [28] en la que encontramos sensores de varias categorías y dentro de ellas diferentes sensores a considerar.

En el apartado de temperatura encontramos sensores para entornos estándar y extremos, siendo el estándar el comprendido entre -40°C y $+125^{\circ}\text{C}$ y que ya cumple nuestras necesidades. Si acudimos al datasheet [29] de estos sensores nos encontramos con diferentes modelos que abarcan un mayor o menor rango, dejando descartado el modelo más básico al no cumplir con las temperaturas mínimas de operación. Al final de la hoja se nos indica también que los modelos de “grado comercial” no deben ser usados en condiciones de polvo y humedad elevado, por lo que la opción a elegir serían los de grado industrial que ya presentan cierto grado de protección contra los elementos, el cual no viene indicado.

Para los otros apartados como la de humedad y gases nos encontramos con opciones bastante similares, resultando las opciones de grado industrial las más convenientes para lo que se busca. Todos estos sensores disponen de comunicación inalámbrica propia que afirma tener un alcance superior a la de otras plataformas y a la propia red wifi. Para usar este potencial se precisa usar uno de sus gateways [30], disponibles con conexión ethernet o redes móviles para su envío a una plataforma IoT mediante software propietario.

Para encontrar más variedad de productos se puede acudir a páginas que agrupen diferentes marcas, un ejemplo es DirectIndustry [31], esta web permite la búsqueda por diferentes categorías y filtrar según las condiciones de trabajo, tipo de conexiones, nivel de protección, etc. La desventaja de esta búsqueda frente a lo visto anteriormente consiste en que algunos sistemas pueden presentar incompatibilidades entre ellos, resultando en una instalación lenta e incluso ineficaz.

1.5.2 Prototipo

Recuperando los objetivos del proyecto nos encontramos que para llevar a cabo el prototipo y maqueta a una escala menor se debe abandonar el ámbito industrial y acudir a los denominados movimientos DIY y maker [32] [33]. Para comenzar con ello es necesaria una plataforma sobre la que trabajar: una de las más usadas en el mundo es Arduino [34], que destaca por su sencillez y facilidad para encontrar información y proyectos de otros usuarios. Otro de sus competidores es Raspberry Pi [35] que de igual manera ofrece diferentes productos para adecuarse a distintos proyectos y cuenta con mucho soporte por parte de la comunidad. En base a la experiencia previa se usará para este proyecto Arduino.

Una vez definida la placa sobre la que se va a realizar el proyecto, de la cual se analizará posteriormente la mejor opción de las que se ofertan en función de entradas y salidas requeridas, es necesario realizar una búsqueda de los sensores compatibles y que suelen usarse, además de que cumplan con lo descrito en el alcance. El sensor elegido en cada categoría será detallado en la posterior descripción del sistema.

Elección del sensor de temperatura

Se accede a una página en que se comparan varios sensores para Arduino [36].

-LM35: uno de los sensores más conocidos y baratos, calibrado en °C, salida analógica. Mide temperaturas entre 2 y 150 con alimentación positiva entre 4-30V, para temperaturas inferiores es necesaria alimentación negativa. Rango total de -55°C a 150°C. Tiene una precisión de $\pm 0.5^\circ\text{C}$ y el tiempo de respuesta es de 4min. Su precio para un lote de 10 puede encontrarse desde 4€ en vendedores internacionales, llegando a encontrarse desde 2€ la unidad en vendedores nacionales.

-TMP36: muy parecido al anterior, sin el inconveniente de la alimentación negativa. Voltaje de operación de 2.7 a 5.5V y rango de medición de -40°C a 150°C con una precisión de $\pm 2^\circ\text{C}$. Su tiempo de respuesta es de 8min. Su precio ronda los 2€ la unidad en vendedores internacionales.

-TC74: sensor digital, comunicación mediante I2C [37]. Tensión de alimentación entre 2.7 a 5.5V y medición de temperatura de -40°C a 125°C. Su precisión oscila entre ± 2 y 3°C . Se toman hasta 8 muestras por segundo. Se puede encontrar con precios superiores a los 5€.

-DHT11: sensor de temperatura y humedad. Salida digital. Voltaje de 3 a 5.5V y medición de 0°C a 50°C con precisión de $\pm 2^\circ\text{C}$. Muestreo de la temperatura cada 2s. Su precio parte de menos de 2€ sin la placa con la resistencia pull-up y de poco más de 2€ con la placa para su conexión directa.

-DHT22: modelo superior al 11. Lectura de temperaturas negativas, rango de -40°C a 80°C con precisión de $\pm 0.5^{\circ}\text{C}$. Muestras cada 2s. Es más caro que su modelo inferior, rondando los 4€ para el modelo sin placa y superando los 5€ si viene con ella.

-SHT15: temperatura (de -40°C a $123,8^{\circ}\text{C} \pm 0.3^{\circ}\text{C}$) y humedad. Protocolo I2C. Muestreo por segundo. Resulta un sensor bastante más caro que los otros, llegándose a encontrar por precios superiores a los 30€.

Aparte de los mencionados en la página hay otras opciones como:

-SHT31: comunicación digital mediante protocolo I2C, tensión de alimentación de 2.4 a 5.5V, medición de temperatura de -40°C a 125°C con precisión de 0.2°C y humedad. Modelo superior al SHT15 y precio similar para su versión oficial. Otras versiones pueden encontrarse en torno a los 5-10€.

-BME680: protocolo I2C, voltaje de 3.3 a 5V. Mediciones de temperatura (-40°C a $123,8^{\circ}\text{C} \pm 1^{\circ}\text{C}$), humedad, presión y gas. Su precio ronda los 10€ para vendedores internacionales y los 30€ para envíos nacionales.

Para la elección final se descartan el LM35 por la necesidad de alimentación negativa y el DHT11 al no medir temperaturas bajo 0. El TMP36 y el TC74 también son descartados ya que los restantes permiten medir la humedad. Entre los restantes, teniendo en cuenta que el sistema no necesita realizar medidas en intervalos cortos, sus características y precios; destacan el SHT31 y el DHT22, siendo este último el ya utilizado en otros proyectos tanto en clase como personales, su elección queda, por tanto, asegurada. Su rango de mediciones es adecuado al proyecto y cuenta con una gran precisión.

Elección del sensor de humedad

En el párrafo anterior ya se ha elegido un sensor capaz de detectar la humedad relativa en el ambiente, aun así, se detallan los términos relevantes a la humedad y se verá qué más opciones se ofertan y puedan resultar interesantes.

-DHT22: 0-100% RH precisión de $\pm 2\%$.

-SHT31: 0-100% RH precisión de $\pm 2\%$.

-BME680: 0-100% RH precisión de $\pm 3\%$.

-HR202: medición de humedad exclusivamente de 20-95% RH en entornos entre $0-60^{\circ}\text{C}$, alimentación de 1.5V en AC [39]. Su precio ronda el euro.

Muchos de los sensores de humedad relativa son los que también miden la temperatura. El sensor HR202 de sólo humedad no vale para nuestro sistema ya que no cumple el rango operativo de temperaturas, además de no cubrir el 100% de la humedad relativa.

Elección del sensor de amoniaco

La elección de un sensor de NH₃ para la plataforma de Arduino no es difícil ya que en el mercado se encuentra, principalmente, una sola oferta. Este sensor es el MQ-137, destacar su alimentación de 5V posible en AC o DC y su necesidad de precalentamiento para conseguir valores precisos. Si este sensor viene ya integrado en una PCB facilita el trabajo de conexionado. Su precio con la placa parte de los 16€.

Elección del sensor de calidad del aire

En este apartado, a diferencia del anterior, se buscarán sensores que detecten más de un elemento en el aire.

-CCS811: voltaje de alimentación de 1.8 a 3.6V y temperaturas de operatividad entre -40°C y 85°C, comunicación mediante I2C. Cuenta con cinco modos de operación que modifican su consumo y su tiempo de muestreo de 250ms, 1s, 10s y 60s. Puede medir concentraciones de CO₂ entre 400 a 8192ppm y TCOV (Total Compuestos Orgánicos Volátiles) entre 0 a 1187ppb (partes por billón) [40]. Su precio comienza en 11€ y puede elevarse hasta superar los 30€ en vendedores locales.

-SGP30: alimentación de 1.62 a 1.98V, comunicación mediante I2C, funcional entre -40°C y 85°C. Mediciones de etanol e hidrógeno (H₂) de 0-100ppm, TCOV de 0-60000ppb y CO₂ de 400-60000ppm. Su resolución varía entre los distintos gases y concentraciones, siendo mayor el error a mayores concentraciones. Su tiempo de medida también cambia, siendo de 1s para el etanol y el H₂ y de 25ms para el CO₂ y TCOV [40]. Su precio varía entre 7-26€ dependiendo del vendedor y envío.

-MQ-135: similar al 137, alimentación a 5V en AC o DC y necesidad de precalentamiento, detección de NH₃ entre 10-300ppm, NO_x, alcohol entre 10-300ppm, benceno entre 10-1000ppm, humo y CO₂. Su temperatura operativa abarca desde -10°C a 45°C y hasta el 95%RH y las concentraciones de oxígeno pueden variar la sensibilidad siendo la óptima de 21%. Se destaca su poco degradamiento con el tiempo. Su precio varía entre 2-4€.

Como elección final se ha optado por el MQ-135 ya que se dispone para la medida de NH₃ de un sensor de la misma familia, además de que este también es capaz de detectar amoniaco a diferencia de sus competidores. Destaca también su menor precio y la capacidad de detección de otros gases y humo.

Elección del sensor de ruido ambiente

-KY-038: alimentación del módulo a 5V, salidas analógica y digital con umbral de disparo regulable mediante el potenciómetro integrado. Micrófono de alta sensibilidad [42]. Es un sensor muy barato que ronda los 0.50€.

-MAX4466: alimentación entre 2.4-5.5V. Módulo con amplificación, aunque no la suficiente para un altavoz. Ganancia ajustable mediante potenciómetro. Salida analógica [43]. Su precio se sitúa en torno a 1.3€.

-MAX9812: modelo L alimentado entre 2.7-3.6V, modelo H entre 4.5-5.5V. Salida analógica. Ganancia fija a 20dB [44]. Se puede encontrar entre 3-4€.

-MAX9814: voltaje de funcionamiento de 2.7-5.5V, configuraciones de ganancia de 40, 50 y 60 dB con control automático, tiempo de ataque programable, rango operativo de temperaturas entre -40°C a 85°C. Salida analógica. Su precio parte de los 4€ para vendedores internacionales y puede llegar hasta los 18€ en algunos vendedores nacionales.

Se elige finalmente el MAX9814 por ser el módulo más avanzado, con control sobre la ganancia.

Elección del sensor de caudal

Para el sensor de caudal nos encontramos que la diferencia entre los modelos es su conexión de rosca, su funcionamiento es el mismo. Se elige el modelo YF-S201 de ½". Su tensión de alimentación se encuentra entre 5-18V, su salida es digital y su rango operativo de -25°C a 80°C.

Elección del motor de apertura de las ventanas

La operación de abrir o cerrar las ventanas puede llevarse a cabo mediante varios tipos de motor y diferentes formas de control. Para el empleo de motores convencionales pueden ser necesarios sistemas de control, necesidad de engranajes multiplicadores o reductores para mover la carga y finales de carrera para detener el movimiento, siendo este caso más adecuado para una implantación real del sistema su escala normal. Otra opción es la del uso de motores paso a paso, de ahora en adelante PaP, en los que también se hace necesario el uso de un sistema de control auxiliar y, en ocasiones, finales de carrera si el motor no es capaz de devolver su información de posición, son también una buena opción de cara a un sistema real.

La opción más idónea para el prototipo y que puede ser controlado desde el propio Arduino consiste en el uso de un servomotor, que ya cuenta con el sistema de control, el grupo reductor y el motor integrado dentro del encapsulamiento. Dentro de esta categoría hay una gran variedad de modelos cuya

diferencia principal es la cantidad de par que pueden proporcionar. La siguiente tabla proporciona una comparación entre los principales servomotores usados en Arduino.

Tabla 1.4: Comparación de los principales servomotores (Fuente:[45]).

Modelo	Tensión de alimentación	Par	Precio aproximado
SG90	3V-7.2V	1-1.6 Kg/cm	2-3€
SG5010	4.8V-6V	3.5-6.5 Kg/cm	4-14€
MG995	4.8V – 7.2V	8.5-10 kg/cm	4-7€
MG946	4.8V – 7.2V	10.5-14 Kg/cm	6-17€
RDS3115	5V – 7.2V	13.5- 15Kg/cm	8-20€

El servomotor que se usará será el SG90 al ya disponer de él de proyectos anteriores.

Elección del display

Para su uso en Arduino encontramos gran variedad de pantallas, desde LDC a OLED de mayor o menor tamaño y resolución. Para este proyecto en que la pantalla no mostrará muchos cambios el OLED presenta una desventaja, es más susceptible de sufrir el llamado efecto “*burn-in*” [45], esto consiste en la degradación de los LEDs que componen la pantalla y que en situaciones de imágenes estáticas es cuando más se agrava. Esto no es exclusivo de los paneles OLED, sucede en todas las pantallas, pero la tecnología usada en los primeros es más susceptible al efecto.

En este proyecto se usará una LCD modelo 2004, que indica 20 caracteres y 4 filas. Su tamaño permitirá presentar los datos sin necesidad de cambiar entre dos visualizaciones.

La conexión de la pantalla se hará mediante protocolo I2C gracias a un módulo adicional, simplificando su uso y cableado.

Nota aclarativa: los precios indicados para los elementos se han consultado a fecha de redacción de esta parte y pueden sufrir modificaciones a consultas posteriores.

Elección del microcontrolador

Una vez hecha la selección de los componentes que conformarán el proyecto se deben analizar las conexiones necesarias para la elección del controlador. A las opciones ya elegidas se sumarán más elementos, los cuales no precisan de comparación entre modelos ya que su uso está bien diferenciado o no hay otro competidor.

En este momento no se diferencia entre entrada o salida, sólo se realiza el recuento de los pines necesarios. Los elementos que pueden hacer uso indistinto de pines digitales o analógicos como el caso del led o el buzzer se han colocado de manera que no queden en uso muchos pines del mismo tipo.

Tabla 1.5: Pines necesarios.

Sensor	Pines necesarios	Tipo de pin	
		Analógico	Digital
DHT22A	1		1
DHT22B	1		1
MQ-137	1	1	
MQ-135	1	1	
MAX9814	1	1	
YF-S201	1		1
SG90	1		1
Pantalla - I2C	2	2	
Driver motores	4		4
Relé electroválvula	1		1
ESP8266	2		2
Buzzer	1	1	
Led	1	1	
TOTAL	18	7	11

Figura 1.9: Comparativa modelos Arduino (Fuente: [47]).

	Arduino Uno	Arduino Mega2560	Arduino Leonardo	Arduino Due	Arduino ADK	Arduino Nano	Arduino Pro Mini	Arduino Esplora
								
Microcontrolador	ATmega328	ATmega2560	ATmega32u4	AT91SAM3X8E	ATmega2560	ATmega168 (versão 2.x) ou ATmega328 (versão 3.x)	ATmega168	ATmega32u4
Portas digitais	14	54	20	54	54	14	14	-
Portas PWM	6	15	7	12	15	6	6	-
Portas analógicas	6	16	12	12	16	8	8	-
Memória	32 K (0,5 K usado pelo bootloader)	256 K (8 K usados pelo bootloader)	32 K (4 K usados pelo bootloader)	512 K disponível para aplicações	256 K (8 K usados pelo bootloader)	16 K (ATmega168) ou 32K (ATmega328), 2 K usados pelo bootloader	16 K (2k usados pelo bootloader)	32 K (4 K usados pelo bootloader)
Clock	16 Mhz	16 Mhz	16 Mhz	84 Mhz	16 Mhz	16 Mhz	8 Mhz (modelo 3.3v) ou 16 Mhz (modelo 5v)	16 Mhz
Conexão	USB	USB	Micro USB	Micro USB	USB	USB Mini-B	Serial / Módulo USB externo	Micro USB
Conector para alimentação externa	Sim	Sim	Sim	Sim	Sim	Não	Não	Não
Tensão de operação	5v	5v	5v	3.3v	5v	5v	3.3v ou 5v, dependendo do modelo	5v
Corrente máxima portas E/S	40 mA	40 mA	40 mA	130 mA	40 mA	40 mA	40 mA	-
Alimentação	7 - 12 Vdc	7 - 12 Vdc	3.35 - 12 V (modelo 3.3v), ou 5 - 12 V (modelo 5v)	5v				

Se puede comprobar que cualquiera de los modelos de la imagen sería válido (se excluye el esplora), para el Arduino Uno que ofrece 6 pines analógicos y en la *Tabla 1.5* se han contabilizado 7, podría cambiarse el LED de alarma a un pin digital.

El modelo final a usar, al igual que en alguna de las anteriores elecciones, es el Arduino nano debido a que ya se encuentra disponible de proyectos anteriores.

1.6 Descripción del sistema

En este apartado se desarrollarán los elementos que conformarán nuestro sistema y que han sido elegidos anteriormente.

1.6.1 Hardware

Las hojas de datos de los siguientes dispositivos pueden encontrarse en el *Anexo 6.1*.

1.6.1.1 Sensores

Sensor de temperatura y humedad – DHT22 [48]

Como ya se ha visto este sensor nos permite la medición de dos variables que guardan una estrecha relación en el confort. Acudiendo a su datasheet se pueden comprobar sus principales características entre las que se destaca:

- Gran precisión.
- Señal digital ya calibrada.
- Estabilidad a largo plazo.
- Gran distancia de transmisión (hasta 100m).
- Bajo consumo.

Su elemento sensitivo está conectado directamente a un procesador de 8-bit lo que gracias a ello y a su calibración de fábrica permite obtener las señales de temperatura y humedad a su salida digital.

Dentro de las especificaciones técnicas se nos indica otro número de modelo que es el AM2302. Necesita de una alimentación de 3.3-5.5V en DC y su consumo durante las mediciones varía entre 1-1.5mA mientras que en reposo su máximo no supera los 50uA.

En lo relativo a su capacidad de medida su rango de temperatura abarca de -40°C hasta los 80°C con un error de ± 0.5 y una resolución de 0.1°C, mientras que para la humedad abarca el 100% del rango con un error de entre $\pm 2\%$ pudiendo llegar a un máximo de 5% en algún caso, si resolución es idéntica a la de la temperatura con 0.1%. El elemento sensible para realizar estas medidas es un condensador de polímero húmedo.

Para su conexión se ha optado por un módulo ya instalado en PCB que incluye una resistencia pull-up entre la señal de salida y la alimentación. A pesar de ello y por recomendaciones de ejemplos vistos se añadirá también una resistencia de 10K Ω entre el pin de alimentación y el de señal.

Se usarán dos sensores A y B, se tomará la medida de ambos y se hará una media, su conexión usará un pin digital para cada sensor y será la siguiente:

Figura 1.10: Conexiones DHT22.



Sensor de amoníaco NH₃ – MQ-137 [49]

Las características que se destacan son:

- Respuesta rápida.
- Alta sensibilidad.
- Larga vida útil.
- Estabilidad.

La alimentación del sensor se puede llevar a cabo mediante corriente continua o corriente alterna, ambas a 5V, usaremos corriente continua ya que es más fácil de obtener este voltaje. El módulo necesita de precalentamiento de por lo menos 24h de su resistencia interna, de valor 31Ω con tolerancia de $\pm 5\%$, para la cual el consumo puede llegar hasta los 800mW.

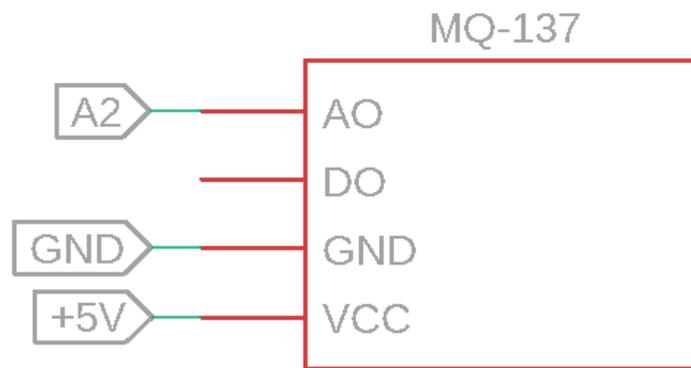
Las condiciones de operatividad van desde -10°C hasta 45°C para la temperatura y hasta el 95% de RH, los niveles de oxígeno para una lectura lo más precisa posible deben estar en el 21%, con un mínimo de 2% para poder realizar mediciones.

Su rango de detección de amoníaco va desde 5 hasta 200ppm para un valor de resistencia de $900\text{K}\Omega$ - $4900\text{K}\Omega$. El calor y humedad pueden afectar su valor.

Es preciso para el uso de este sensor contar con una resistencia limitante. El módulo adquirido ya cuenta con ella.

Se usará una entrada analógica para leer su valor, su conexión resultante es:

Figura 1.11: Conexiones MQ-137.



Sensor de calidad del aire – MQ-135 [50]

Siendo éste un sensor de la misma familia que el sensor usado para el amoniaco se puede esperar que características principales sean similares. Su principio de funcionamiento se basa en la medición de la conductividad de su elemento semiconductor, en este caso SnO₂. La conductividad es menor en aire limpio y va aumentando con las concentraciones de gases. Los gases que puede medir son NH₃, alcohol, benceno, humo y CO₂. Midiendo esta variación se obtiene el valor a la salida, que debe ser convertido a un parámetro entendible para nosotros, como ppm. Destaca por:

- Buena sensibilidad en un amplio rango.
- Respuesta rápida.
- Larga vida útil.
- Bajo coste y simplicidad.

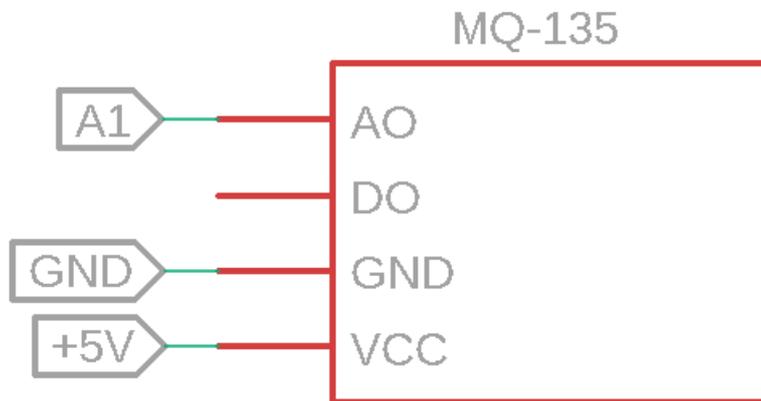
Los parámetros de alimentación son iguales a los del MQ-137, se debe alimentar a 5V y se usará corriente continua. Su resistencia para precalentamiento en este caso es de $33\Omega \pm 5\%$ para una temperatura ambiente y su consumo sigue siendo inferior a 800mW. Se recomienda un precalentamiento superior a 24h.

Su rango operativo se mantiene, las temperaturas que soporta van desde -10°C hasta 45°C y humedad relativa menor al 95%. La concentración de oxígeno que asegura mediciones precisas es de 21% y el mínimo para su funcionamiento es del 2%.

Su rango de medida es de 10-300ppm para el NH₃ y el alcohol y de 10-1000ppm para el benceno. Su resistencia de medida varía entre 30K Ω hasta 200K Ω y puede sufrir variaciones con la temperatura y humedad.

Para realizar las mediciones se necesita una resistencia limitante, la cual ya viene instalada en el módulo.

Figura 1.12: Conexiones MQ-135.



Sensor de ruido ambiente – MAX9814 [51]

Este sensor consiste en un micrófono amplificado con sesgo en bajo ruido, sus características a destacar son:

- Control automático de ganancia (AGC).
- Tres ajustes de ganancia de 40dB, 50dB y 60dB.
- Tiempo de ataque y liberación programables.

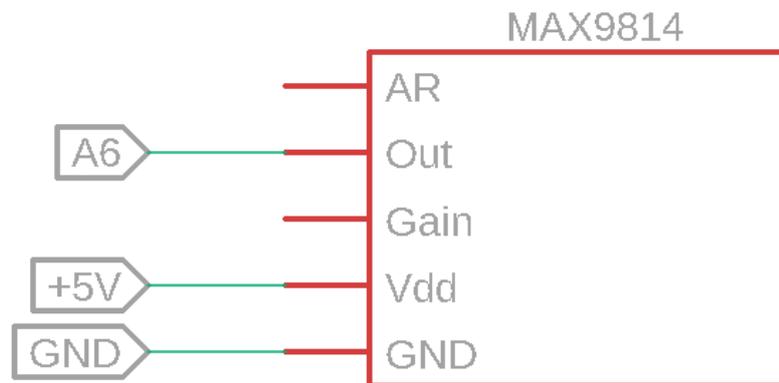
El preamplificador de bajo ruido tiene una ganancia fija de 12dB, un amplificador variable de ganancia entre 0 y 20dB que depende del voltaje de salida y del control automático de ganancia. Esto permite conseguir a su salida los ajustes de 40, 50 o 60dB. Estos ajustes se consiguen cableando el pin de ganancia a Vdd para 40dB y a GND para 50dB, dejar el pin sin conectar proporciona 60dB

Los tiempos de ataque y liberación permiten modificar la compresión del sonido, sin embargo, para este proyecto no serán necesarios.

Las características eléctricas de este modelo definen una tensión de alimentación entre 2.7 y 5.5V y cuyo consumo de corriente nominal es de 3.1mA, pudiendo llegar hasta un máximo de 6mA.

Se usará un pin analógico para su lectura. Su conexión final es:

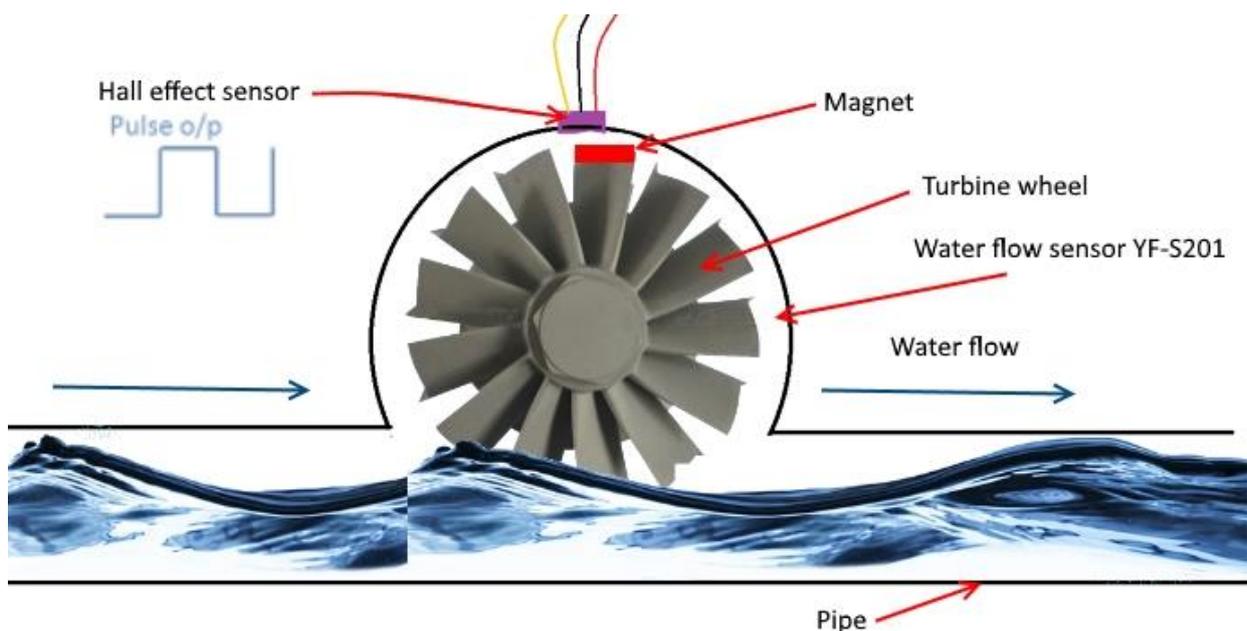
Figura 1.13: Conexiones MAX9814.



Sensor de caudal – YF-S201 [52]

Este sensor consiste en un cuerpo de plástico con roscas de $\frac{1}{2}$ " en sus extremos para la conexión al suministro de agua. En su interior se encuentra una turbina, un imán y un sensor de efecto Hall. Este sensor basa su funcionamiento en el *'fenómeno físico resultante de la aparición de un campo eléctrico por separación de cargas en el interior de un conductor por el que circula un campo magnético'* [53]. En nuestro caso esto ocurre porque en la turbina se encuentra un imán y en la carcasa el sensor, de esta manera al circular el agua y mover la turbina se mueve el imán, que pasa cerca del sensor y es contabilizado. Un ejemplo muy sencillo es el del cuentakilómetros de una bicicleta.

Figura 1.14: Funcionamiento del sensor de efecto Hall (Fuente: [54]).



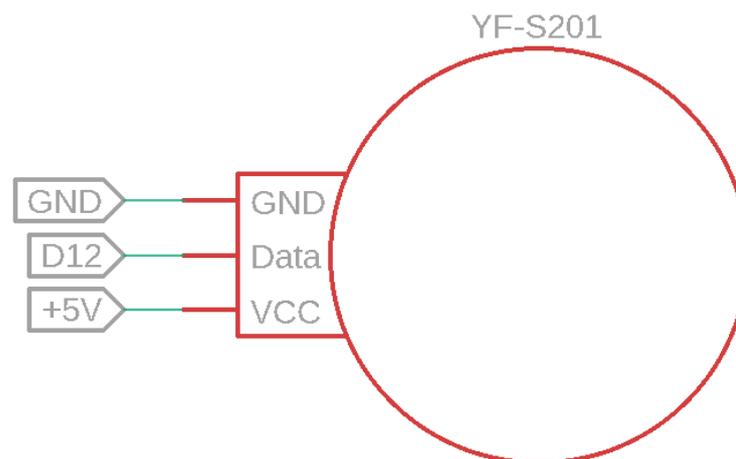
Para hacer funcionar el sensor se debe proporcionar una tensión normal de alimentación de entre 5 y 18V, pudiendo soportar entre 4.5 y 24V. El consumo máximo de corriente para el valor normal inferior de tensión es de 15mA.

Puede medir un flujo de entre 1 a 30L/min con una presión no superior a 2MPa y una temperatura máxima del líquido de 120°C.

Las condiciones ambientales para su instalación son de una temperatura menor a 80°C, no se indica temperatura mínima, pero dado que va a circular agua en su interior y ésta se expande al congelarse es recomendable que no suceda, ya que podría romper la carcasa o la turbina.

Su lectura se realiza mediante un pin digital, ya que su onda obtenida es cuadrada y se deberán contar los pulsos.

Figura 1.15: Conexiones YF-S201.



1.6.1.2 Actuadores

Driver para motores de DC – L298N [55]

El control de los motores de corriente continua puede realizarse de dos maneras principales. Se podría hacer uso de un relé permitiendo un accionamiento de todo o nada o, como en este caso, hacer uso de un driver que permite modificar su velocidad mediante una señal PWM [56] aplicada en los pines de activación (enable) ENA y ENB. El driver también permite modificar su sentido de giro o frenarlo en función de si la señal que llega a los pines IN1-IN4 es alta o baja como muestra la imagen adjunta.

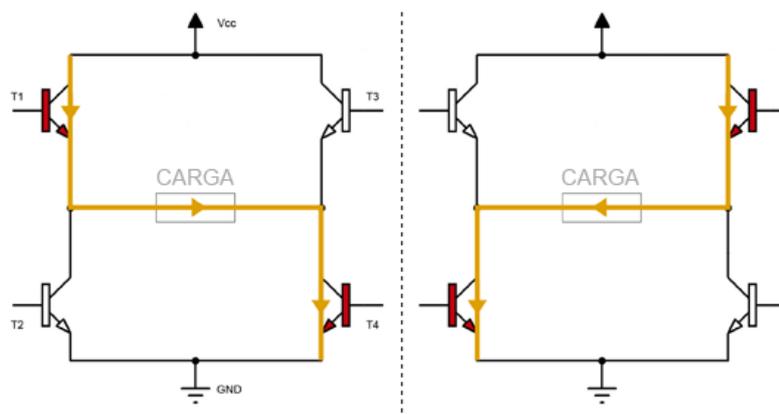
Figura 1.16: L298N sentido de giro (Fuente: [57]).

	Adelante	Atrás	Freno
IN1 (o IN3)	HIGH	LOW	LOW
IN2 (o IN4)	LOW	HIGH	LOW

El driver funciona mediante el llamado puente-H, consistente en 4 transistores que conectados diagonalmente hacen circular la corriente en un sentido u otro para permitir cambiar el giro.

Su uso en este proyecto no precisa de cambios en el sentido de giro ni freno, por lo que los pines se asociarán 2 a 2.

Figura 1.17: Puente-H (Fuente: [57]).



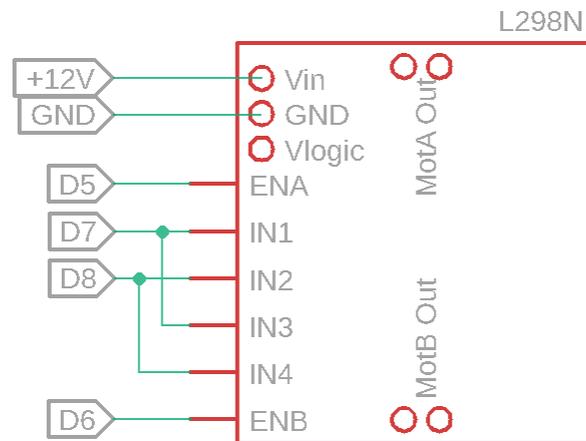
El módulo en el que viene integrado hacer que sus conexiones sean más simples. El módulo admite una entrada de entre 5V a 35V para Vin. Si la tensión es igual o inferior a 12V y el jumper que se encuentra justo encima está cerrado el módulo se alimentará mediante el regulado incluido a 5V y se obtendrá en el pin Vlogic esos 5V de salida. Para tensiones superiores a 12V es necesario abrir el jumper para evitar quemarlo y alimentar la electrónica del módulo mediante el Vlogic a 5V.

Los motores a conectar en los laterales de la placa deben tener un consumo máximo de 2A cada uno.

El consumo del propio circuito se encuentra entre 0-36mA aproximadamente.

Para su control y, en consecuencia, el de los motores se usarán 4 pines digitales. Dos serán necesarios para controlar la velocidad de cada motor respectivamente y los otros dos para el sentido de giro.

Figura 1.18: Conexiones driver L298N.



Ventilador DC 12V [58]

Para representar los ventiladores se ha optado por un modelo genérico a 12V de tamaño reducido, usado normalmente para refrigeración de pequeños componentes o en extrusores de impresoras 3D.

Aunque su indicativo sea de 12V el ventilador admite tensiones desde 4.5V hasta 13.2V y de ello dependerá su velocidad. El consumo máximo es de 0.6W o 0.05A.

Otros datos que pueden resultar útiles son su velocidad de $5000 \pm 15\%$ rpm, su flujo a presión cero de 0.847CFM o su presión a flujo cero de 1.15mmH₂O.

Su conexión no representa una problemática ya que son de dos cables los cuales van directamente al driver. Al querer que los ventiladores giren hacia el mismo sentido y queda bloqueado en la programación en caso de que giren en sentidos opuestos se deberán permutar los cables.

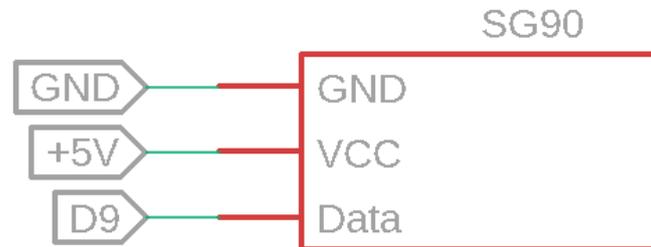
Servomotor – SG90 [59]

Un servomotor no es más que un motor al que se le ha añadido el driver y el grupo reductor y que tiene la capacidad de moverse hasta una posición y mantenerla. En este caso la posición que puede adoptar se encuentra entre 0° y 180°, limitada por su construcción. Se determina mediante un potenciómetro integrado.

Se debe alimentar entre 4.8 y 6V. Su consumo depende de su momento de trabajo siendo lo más bajo en reposo de 10mA, entre 100 y 250mA en movimiento y hasta 360mA cuando se encuentra fijando una posición.

El par máximo que puede ejercer es de 1.7kg/cm y su velocidad de movimiento 0.12s/60°.

Figura 1.19: Conexiones SG90.



Buzzer [60]

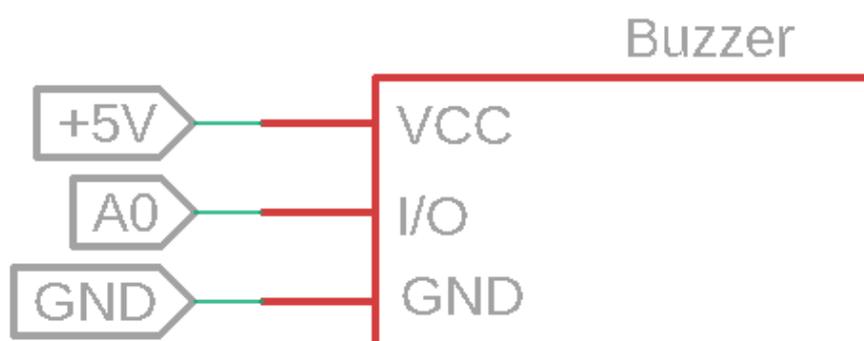
El buzzer o zumbador es un pequeño altavoz que representará la alarma del sistema, como muchos otros elementos viene ya montado en una PCB acompañado de un transistor para poder alimentarlo y transmitir los datos por separado, no siendo realmente necesario debido a su pequeño consumo, aunque permite liberar esa carga del microcontrolador.

Necesita alimentación entre 4 y 8V, para el valor medio de 6V su gasto de corriente es inferior a 30mA.

Su rango operativo permite usarlo entre -25°C hasta 80°C.

Su frecuencia de resonancia es de 2300±300MHz. Y su tono es continuo, aunque como veremos más adelante en la programación esto puede ser cambiado.

Figura 1.20: Conexión Buzzer.

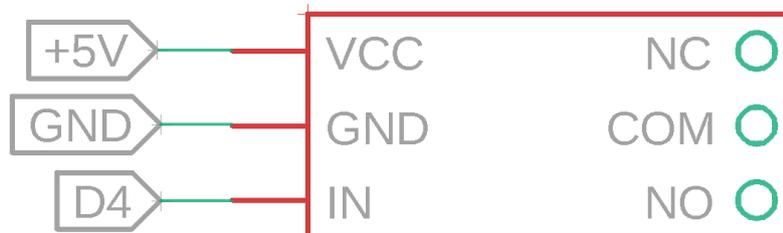


Relé [61] y electroválvula

Si se desea controlar algún elemento de mayor voltaje o potencia mediante una pequeña señal como es la del Arduino es necesario incorporar un relé. De manera simplificada consiste en un interruptor que será activado mediante una señal digital. Al activarse el relé proporcionará alimentación a una electroválvula genérica que cerrará la tubería de agua.

Se debe alimentar el módulo a 5V. A su salida posee tres conexiones con capacidad de 250VAC o 30VDC, una normalmente abierta (NO) hasta 10A, otra normalmente cerrada (NC) hasta 5A y el común (COM).

Figura 1.21: Conexionado relé.



1.6.1.3 HMI

Pantalla LCD – 2004A y controlador I2C – F113 [63] [64]

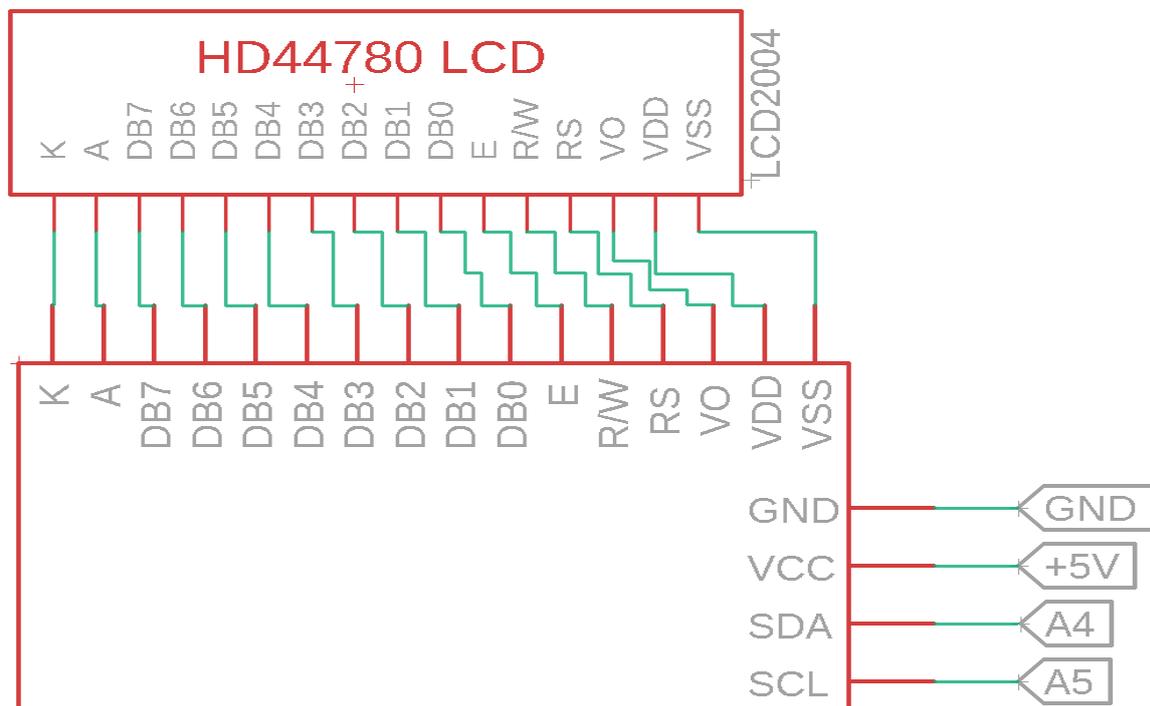
Ya se ha mencionado en la elección de la pantalla la ventaja del uso de una con tecnología LCD frente a una OLED en nuestro caso, debido al efecto *Burn-in*. Una pantalla LCD denominada así por ser de cristal líquido y que ha sido usada desde hace bastante tiempo hace uso de las propiedades de la luz polarizada para mostrar los caracteres, haciendo uso de filtros mientras recibe una iluminación de fondo [65].

La alimentación de la luz de fondo se realiza a 5V, usados también para el nivel lógico.

Soporta temperaturas de entre -10°C a 60°C.

Para controlar la pantalla de una manera más sencilla y evitar tanto cableado se suele recurrir a un controlador I2C, protocolo ya en el glosario y que hace uso de los pines exclusivos A4-SDA y A5-SDL.

Figura 1.22: Conexiones LDC y controlador I2C.



Módulo WiFi – ESP8266-ESP01 [66]

Necesario para poder añadir la comunicación inalámbrica a nuestro proyecto, en este caso mediante la red WiFi de 2.4GHz, y poder enviar los datos para su visualización online y por ello se encuentra dentro de este apartado de HMI. Dentro de los modelos que podemos encontrar [67] se ha optado directamente por el ESP01 ya que es el que se suele encontrar más fácilmente y muchos ejemplos lo usan.

Este modelo cuenta con todo lo necesario en un módulo compacto, incluida la antena. Puede funcionar independientemente o como esclavo de otro microcontrolador mediante comandos AT enviados por el puerto de comunicaciones serie.

Sus características de red permiten su conexión según los protocolos 802.11 b/g/n para la frecuencia ya mencionada de 2.4GHz hasta una velocidad de 72.2Mbps. Admite seguridad WPA / WPA2 y cifrado WEP /TKIP /AES y protocolos IPv4, TCP, UDP o HTTP.

Su alimentación debe ser entre 2.5 y 3.6V siendo lo típico el uso de 3.3V. La intensidad promedio requerida para su funcionamiento es de 80mA.

Las temperaturas de operación soportadas comprenden desde -40°C hasta los 125°C.

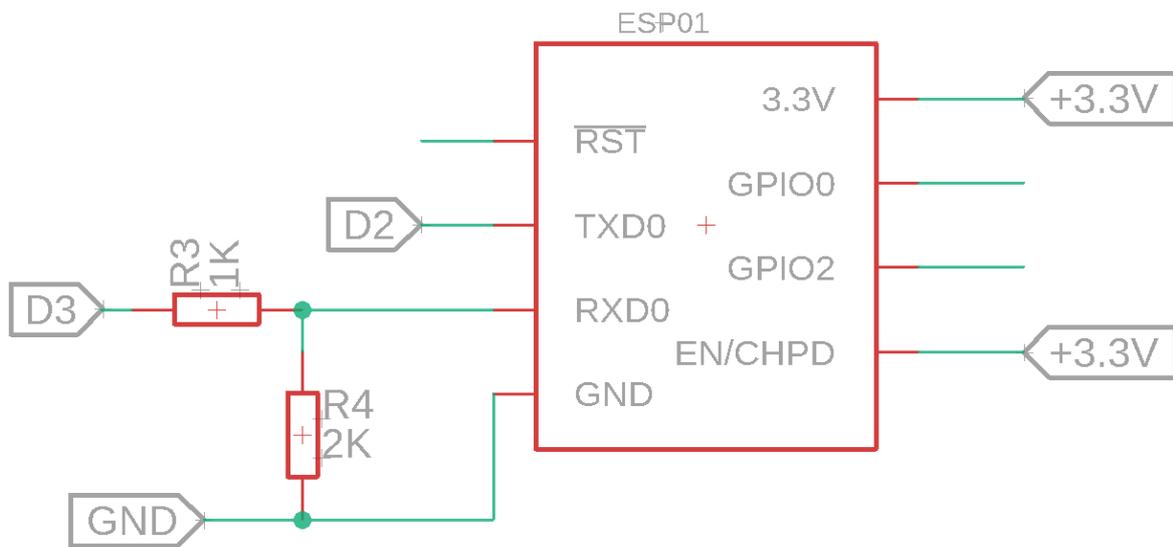
Figura 1.23: Pines del ESP8266-ESP01 (Fuente: [68]).



- GND es la toma de tierra.
- GPIO2 es una entrada salida de propósito general. Es el pin digital número 2.
- GPIO0 es una entrada salida de propósito general. Es el pin digital número 0.
- RXD es el pin por donde se van a recibir los datos del puerto serie. Trabaja a 3,3 V. También se puede utilizar como pin digital GPIO: sería el número 3.
- TXD es el pin por donde se van a transmitir los datos del puerto serie. Trabaja a 3,3 V. También se puede utilizar como pin digital GPIO: sería el número 1.
- CH_PD pin para apagar y encender el ESP-01: si lo ponemos a 0 V (LOW) se apaga, y a 3,3 V (HIGH) se enciende.
- RESET pin para resetear el ESP-01: si lo ponemos a 0 V (LOW) se resetea.
- Vcc es por donde alimentamos el ESP-01. Funciona a 3,3 V y admite un máximo de 3,6 V. La corriente suministrada debe ser mayor que 200 mA [68].

En el caso del pin RXD, en el cual se indica que es por donde recibe los datos y que trabaja a 3.3V, es necesario realizar un divisor de tensión a la salida del pin de 5V de Arduino.

Figura 1.24: Conexiones ESP8266-ESP01.



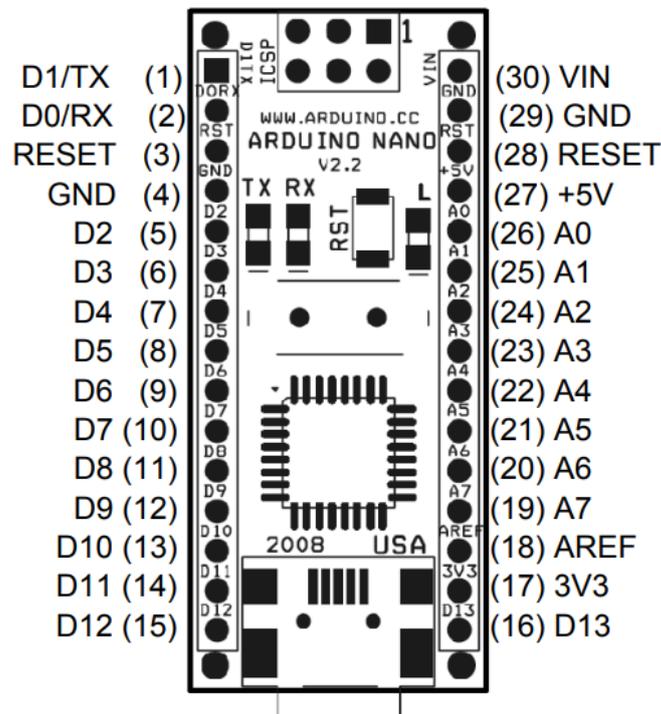
1.6.1.4 Controlador

El “cerebro” del sistema será el Arduino nano, basado en el microcontrolador de 8-bit de Atmel ATmega328p [69]. Algunas de sus principales características son:

- Arquitectura RISC (Reduced Instruction Set Computing)
- Memoria flash de 32Kb.
- Memoria EEPROM de 1Kb.
- 2Kb de SRAM.
- Frecuencia de trabajo de 16MHz.
- 2 temporizadores/contadores de 8-bit.
- 1 temporizador/contador de 16-bit.
- Conversores analógico-digital.
- Compatible con interfaces maestro/esclavo.
- Compatible con protocolo I2C.
- Funciones de interrupción y reseteo.

Una vez ya integrado en la placa se nos proporciona una variedad de conexiones disponibles. Comenzando por sus pines situados en su periferia:

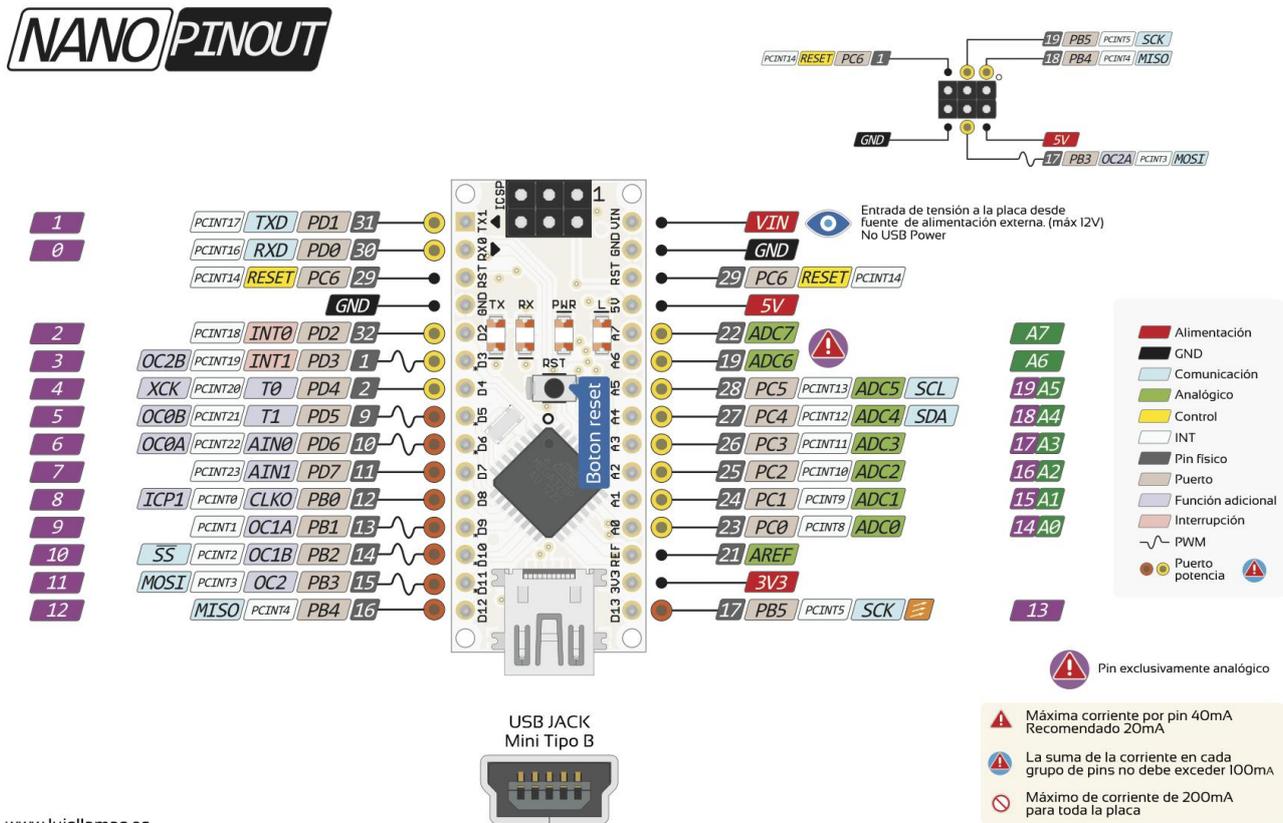
Figura 1.25: Pines Arduino nano (Fuente: [70]).



- 1-2, 5-16 Pines digitales de entrada/salida, denominación letra D seguido número de pin.
- 19-26 Pines analógicos de entrada, denominación letra A seguido número de pin.
- 3, 28 Pines de Reset cuando se conectan a tensión 0, denominación RST
- 18 Pin de referencia para el convertor analógico-digital (ADC), denominación AREF.
- 4, 29 Pines de tierra, potencial 0, denominación GND.
- 27 Salida (regulador interno) o entrada de +5V, denominación +5V.
- 17 Salida de +3.3V, denominación 3V3.
- 30 Entrada alimentación (7-12V), denominación VIN.

Esto sería lo básico de los pines del Arduino nano, si se acude a otras fuentes podemos encontrar imágenes con más detalles sobre cada pin y el módulo.

Figura 1.26: Pines Arduino nano, detallado (Fuente: [71]).



www.luisllamas.es

En esta imagen ya se puede ver que el Arduino cuenta con USB mini para su alimentación y conexión al ordenador para programarlo y/o visualizar los datos del puerto serie.

Para algunos puertos digitales se cuenta con señal modulada PWM, denotada mediante un símbolo ondulatorio.

Los pines analógicos 1-5 pueden actuar como digitales, 6 y 7 son exclusivamente analógicos.

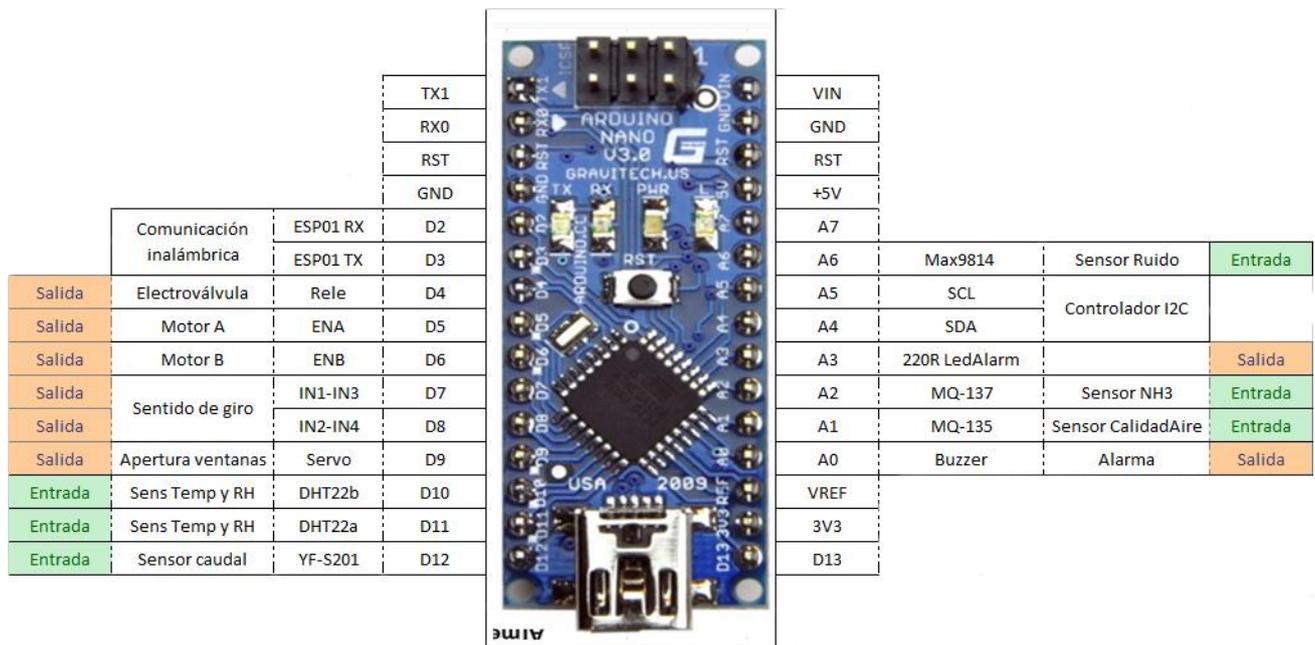
Se observa también un botón físico de reset y LEDs indicadores de funcionamiento.

Volviendo a sus características ya se ha visto que su voltaje de operación es de 5V, que puede alimentarse mediante USB, aplicando una tensión de 5V al puerto +5V o aplicando entre 7-12V al puerto VIN. Su consumo es de 19mA. Cuenta con salidas de 5V (también en los pines de salida) y 3.3V, pudiendo suministrar una intensidad en cada uno de 40mA hasta un total en conjunto de 200mA.

Su rango operativo empieza en -40°C y llega hasta los 85°C.

Una vez conectados todos los elementos que conforman el sistema y asignadas sus entradas y salidas el conexionado final del Arduino resulta en:

Figura 1.27: Conexiones Arduino nano.



1.6.1.5 Alimentación

Fuente de alimentación para protoboard – MB108 [72]

La tensión necesaria para los elementos vistos normalmente es de 5V y en algún caso de 3.3V. Para lograr esto es necesario un conversor de voltaje que para Arduino se puede encontrar fácilmente con conexionado para protoboard y ya nos suministra ambos voltajes requeridos, seleccionables mediante los jumper en sus extremos.

El módulo puede ser alimentado mediante un USB de tipo A de 5V o mediante el conector Barrel Jack entre 6.5 y 12V, siendo entonces el USB una salida de 5V. Permite una corriente máxima de salida de 700mA y cuenta con interruptor de encendido.

Regulador de tensión –LM7805 [73]

Para algunos módulos que necesiten más potencia (como los sensores de gases) y para no sobrecargar la fuente MB108 se usarán estos reguladores de tensión a 5V.

En la entrada admiten voltajes entre 7 y 25V, y a la salida pueden proporcionar hasta 1.5A.

Alimentación general

Para suministrar tensión a los módulos conversores que a su vez alimentan al sistema se usarán fuentes de 12V conectadas a la red o baterías de 12V.

1.6.2 Software

1.6.2.1 Entorno de programación

El código que se ejecutará en el Arduino y permitirá recolectar los datos de los sensores, procesarlos, actuar en consecuencia y mostrarlos al usuario estará escrito, aunque en lenguaje propio, basado en C++. Los programas de los que se dispone para escribir el código son:

Arduino IDE

El IDE [74] propio de Arduino [75] proporciona las funciones más básicas para programar como el editor de código, depurador para los errores y el monitor serial. Permite compilar y cargar el código a una gran variedad de modelos de Arduino y otras placas de manera sencilla.

Visual Studio Code + PlatformIO

Este programa [76] desarrollado por Microsoft ya ofrece funcionalidades como el autocompletado, marcado de errores, integración con Git y desarrollo remoto o colaborativo; es una herramienta más completa a la cual se pueden añadir extensiones que mejoren nuestra productividad como colorear los pares de paréntesis y llaves para diferenciar el de apertura y el de cierre o colorear las funciones, datos y su tipo, etc. Esto permite tener un código más visual, ordenado y entendible. Otra extensión interesante permite analizar nuestra escritura y llevar a cabo un autocompletado más elaborado gracias a funciones de Inteligencia Artificial, pudiendo completar líneas enteras que resultan repetitivas.

Sumado al programa se cuenta con la extensión de PlatformIO [77] que añade la funcionalidad para programar dispositivo embebidos como Arduino o el ESP8266. El programa detecta automáticamente el puerto al que se encuentra conectado el Arduino.

1.6.2.2 Diseño

Para el diseño de los planos, conexionado y diseño 3D se ha hecho uso de los programas de Autodesk.

AutoCad

Software de diseño asistido por ordenador para realizar los planos a escala de las instalaciones y sus medidas.

Eagle

Software para el diseño de PCB, aunque en este proyecto no sea necesaria su realización ya que los sensores se encuentran repartidos por la instalación. Se ha usado para mostrar los conexiones de los sensores.

Fusion360

Software de diseño 3D para elaborar algunas piezas del proyecto, destinadas a su posterior impresión

Para la redacción, tablas, gráficos y flujogramas se han usado programas de Microsoft.

Word

Redacción del proyecto.

Excel

Creación de las tablas, aunque pueden ser elaboradas desde el propio Word resulta más fácil y sencillo su uso, además permite funciones avanzadas.

Creación de diagramas de Gantt que permitan visualizar el tiempo de dedicación a las distintas tareas que componen este proyecto.

Visio

Realización de flujogramas para entender el funcionamiento del código de una manera más visual.

1.7 Instalación existente

El proyecto toma como base una nave ya construida. Las instalaciones han sufrido ligeras modificaciones desde sus planos originales, de los cuales se dispone y que se tomarán como referencia para trazarlos de nuevo, esta vez en formato digital, con el uso de AutoCad.

El diseño de la nave actual cuenta con varios ventanales y aperturas para ventilación pasiva y un par de ventiladores que proporcionan la ventilación forzada en los meses más cálidos y los cuales se van a mantener.

La nave cuenta con un suministro de agua procedente desde el lateral izquierdo (tomando como referencia un observador situado de cara a las puertas) desde donde se distribuye al resto.

Los planos pueden ser consultados en el Anexo II.

2 Pruebas y calibraciones

Cada sensor será puesto a prueba de manera individual para comprender su funcionamiento y analizar los datos que devuelve. Posteriormente se explicará su funcionamiento dentro de la programación final.

Sensor de temperatura y humedad

El sensor elegido viene ya calibrado para obtener a su salida la temperatura en grados centígrados y la humedad relativa en porcentaje. Para su uso se precisa de una librería, usada para los sensores de la familia DHT.

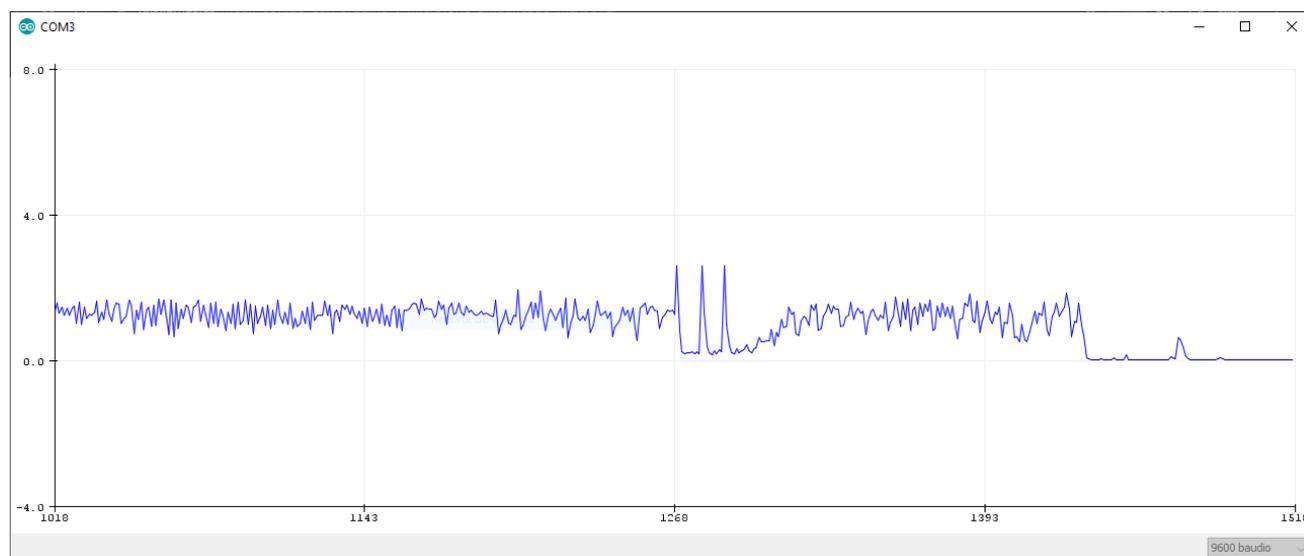
Considerando el ejemplo [78] se comprueba que para ello se incluye la librería DHT.h primero, posteriormente se define el modelo del sensor y el pin al que está conectado. En el setup se debe inicializar el sensor. En el loop, para leer la temperatura y humedad, mediante un comando read de la librería se obtienen directamente los valores que se alojan en variables. También se comprueba si no ha devuelto valores, de manera que se pueda ver si hay un error en el sensor.

Sensor de ruido ambiente

Se toma como base el código para un sensor de la misma familia, el MAX9812 [43]. En este caso no se usa librería, lo primero es definir el pin al que va conectado el sensor (se recuerda que es analógico) y una ventana de muestreo que corresponde a 50ms o 20Hz. Para medir durante esta ventana es necesario contar el tiempo de programa, en milisegundos, y definir un máximo de señal en 0 y un mínimo en 1024, el equivalente al rango de señal que puede recibir la entrada de Arduino. Una vez tomadas las medidas durante este intervalo y actualizado el máximo y mínimo su diferencia nos da el valor de pico a pico, que luego es convertido a voltios.

Una visualización rápida del funcionamiento del sensor se puede obtener usando la función de plotter del Arduino IDE. En el que se ven las variaciones en el ruido hablando un poco, los picos altos al dar una palmada que luego descienden rápidamente y el valor casi 0 al estar en silencio.

Figura 2.1: Plotter voltios salida MAX9814.



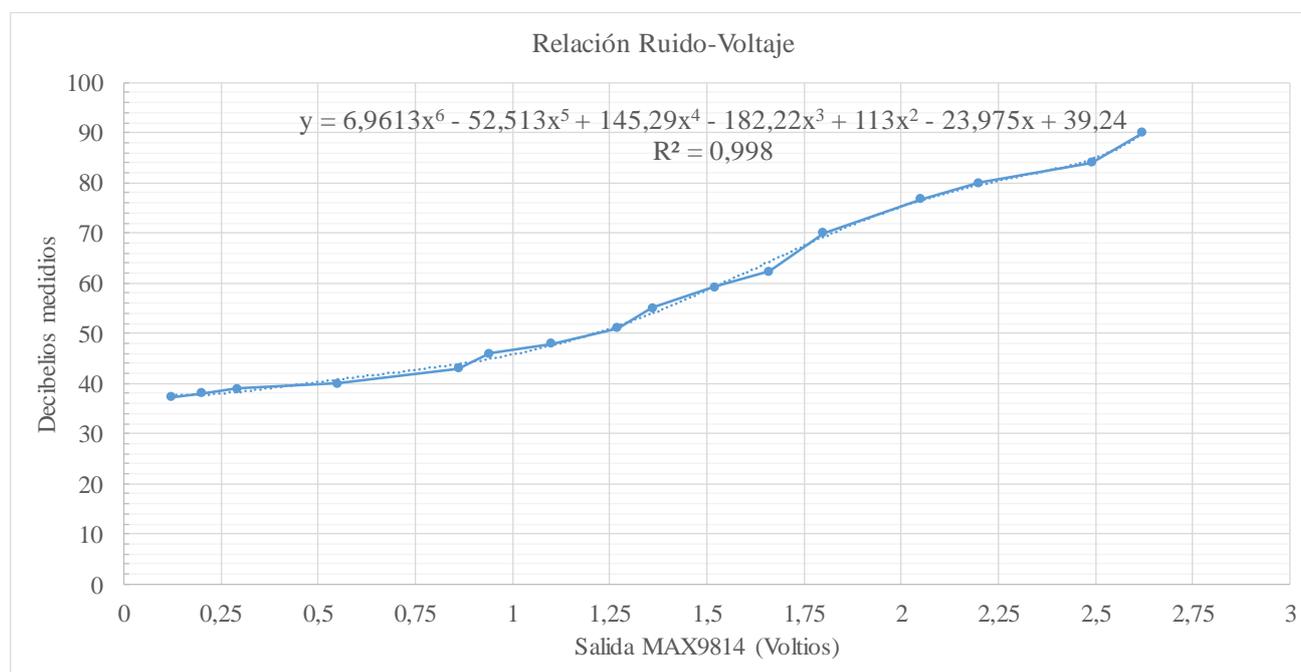
Para este sensor si es necesaria calibración si queremos obtener a la salida un valor en decibelios, para ello se conecta al Arduino y se ejecuta su código. Para diferentes valores de sonido producidos se miden mediante una aplicación para móvil, en este caso y tras haber visualizado una comparativa sobre una aplicación que sea bastante precisa, se usará Decibel X [79]. Los valores en dB de la aplicación se mostrarán en una tabla frente a los valores en voltios obtenidos del Arduino,

Tabla 2.1: Tabla relación Ruido-Voltaje para MAX9814.

Salida MAX9814 (V)	Nivel ruido (dB)
0,12	37,3
0,20	38,0
0,29	38,9
0,55	40,0
0,86	43,0
0,94	46,0
1,10	47,9
1,27	51,0
1,36	55,0
1,52	59,2
1,66	62,4
1,80	70,0
2,05	76,8
2,20	80,0
2,49	84,0
2,62	90,0

A partir de la tabla ya se puede elaborar un gráfico de dispersión.

Figura 2.2: Gráfico relación Ruido-Voltaje para MAX9814.



Para los puntos obtenidos se busca una línea de tendencia que tenga el mejor ajuste posible. Probando las opciones que ofrece Excel se obtiene el mejor ajuste para un polinomio de sexto grado como se puede observar en el gráfico superior.

Para obtener en el código el valor final en decibelios basta con introducir el polinomio, siendo 'y' de la función el valor en decibelios que queremos obtener y 'x' el valor en voltios que se obtiene con anterioridad. Destacar que para poder realizar el cálculo se usa la función pow.

Sensor de caudal

Para el caudalímetro también es necesaria una pequeña calibración. El código [80] nos permite conocer el número de pulsos obtenidos al hacer pasar un volumen determinado por el sensor, para ello se define el pin al que está conectado y una variable de conteo de pulsos. En el setup se incluye una interrupción para el pin que ejecuta la función de contar los pulsos y en el flanco de subida. En el loop sólo se lleva a cabo el mostrar el número de pulsos por el monitor serie.

Tras realizar 5 pruebas, vertiendo agua previamente medida su capacidad en litros, a través del sensor se obtiene:

Figura 2.3: Calibración YF-S201.

Litros	Pulsos	K
1	439	7,32
2	854	7,12
0,5	214	7,13
0,5	222	7,40
1	429	7,15

El valor de K es calculado mediante la expresión $K = \frac{n^{\circ} \text{ de pulsos}}{\text{volumen} * 60}$.

Se realiza el promedio entre los valores obtenidos de K, dando como resultado que la constante para nuestro sensor es de 7.22.

Como lo que queremos es medir el caudal en litros por minuto en el código [81] para su uso se define un intervalo de medida y la constante calculada. De igual manera se usa la interrupción y luego se obtiene la frecuencia de pulsos para el intervalo de medida. Conociendo la frecuencia y la constante se obtiene el valor en L/min mediante una simple división.

Sensores MQ-137 y MQ-135

Al tratarse de sensores de la misma familia su proceso de calibración es similar. En la página [82] que se ha tomado ya anteriormente como referencia para otros sensores y elementos nos aparecen ejemplos de código para realizar lecturas tanto digital (con el potenciómetro integrado en la placa para modificar el umbral de disparo) como analógica (para medir el voltaje resultante a la salida), sin embargo, para conocer la concentración de gases es necesario relacionar el voltaje con la concentración como es el caso del último de los ejemplos, el cual ya viene completo.

El procedimiento separando los pasos se puede encontrar en este video [83] y cuyo ejemplo se puede descargar, de esta manera tras realizar varias medidas nos devuelve el valor de R0.

Para ambos ejemplos es necesario, en base al datasheet de cada sensor, deducir la resistencia del sensor en el aire limpio. La gráfica obtenida, en escala logarítmica, se presenta como una imagen y para hallar los valores haremos uso de la página WebPlotDigitalizer [84].

Tras realizar un recorte para obtener solo la gráfica y subirla a la página web se nos indica que señalemos 4 puntos, dos para cada eje, y que se indique su valor, así como una casilla que marcar si se trata de escala logarítmica como es el caso. Lo siguiente consiste en marcar el punto que queremos que se nos indique su valor, en este caso al tratarse de una línea recta horizontal el valor que nos

interesa es el del eje 'y' y por ello se designan varios puntos sobre la misma línea. Después de analizarlos la página nos deja ver ambas coordenadas de cada punto y exportarlos a formato csv.

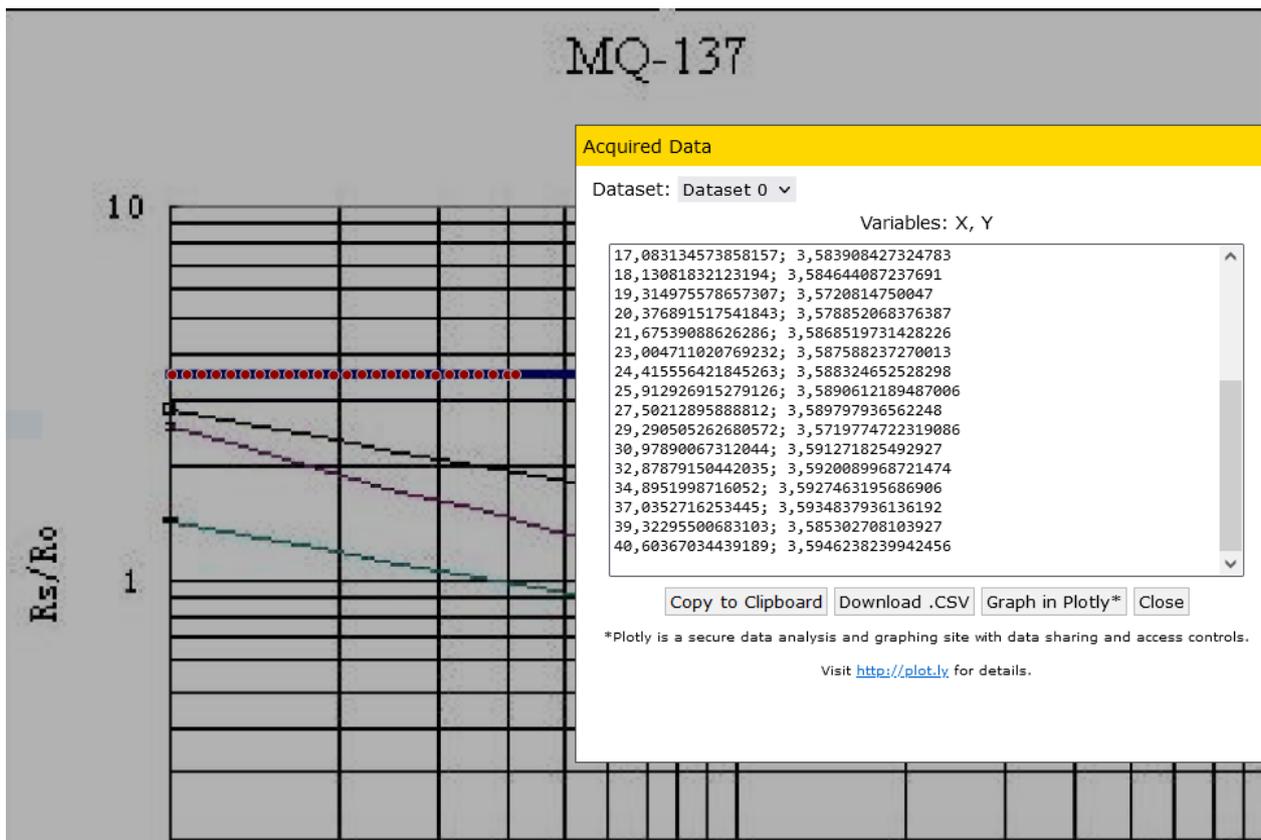
De los valores obtenidos se hace un promedio y se aproxima a dos decimales, que será el valor a introducir en el ejemplo para ejecutarlo en el Arduino para obtener R0.

Tabla 2.2: Valor de Rs obtenido a partir de la gráfica del MQ-137.

Valores de Y (MQ-137)				
3,57283522	3,5809673	3,58464409	3,58832465	3,592009
3,57802858	3,58170235	3,57208148	3,58906122	3,59274632
3,57876303	3,58243756	3,57885207	3,58979794	3,59348379
3,57949764	3,58317292	3,58685197	3,57197747	3,58530271
3,58023239	3,58390843	3,58758824	3,59127183	3,59462382

Promedio:	3,58
------------------	------

Figura 2.4: Ejemplo de cómo se presentan los datos en WebPlotDigitalizer (Fuente: [84]).



Observando la gráfica del otro sensor vemos que la línea del aire es bastante similar al ya calculado, permitiendo facilitar la calibración al no tener que cambiar el parámetro. Destacar que esto no es así

para todos los sensores, ejemplificando con el caso del MQ-2 y MQ-3 que tienen un valor aproximado de 10 y 60 respectivamente.

Tabla 2.3: Valor de Rs obtenido a partir de la gráfica del MQ-135.

Valores de Y (MQ-135)				
3,58347212	3,58347212	3,58347212	3,57321526	3,58347212
3,58347212	3,58347212	3,58347212	3,58347212	3,58347212
3,58347212	3,58347212	3,58347212	3,58347212	3,57435346
3,58347212	3,58347212	3,58347212	3,58347212	3,58347212
3,58347212	3,58347212	3,58347212	3,58347212	3,58347212
Promedio:				3,58

Para los actuadores no será necesaria una calibración, así que se limitará a probar su funcionamiento.

Driver motor

El controlador de los motores permite el control de la velocidad y el sentido de giro, su ejemplo de funcionamiento es además bastante sencillo. Una vez declarados cada pin que se usará lo único a realizar es una escritura analógica (PWM) del valor de velocidad del motor (de 0 a 255) para ENA o ENB y configurar las entradas IN1-4 en función del sentido de giro deseado como se ve en la *Figura 1.16*. Una apreciación es que algunos motores no permiten el cambio del sentido de giro, si se da el caso también puede ser necesario intercambiar las conexiones de los pines del motor con el driver.

Ejemplos se pueden encontrar en muchos sitios, acudiendo a la página de confianza tenemos un ejemplo que hace girar motores en un sentido y luego en el otro a la velocidad que indiquemos previamente [85].

Servomotor

Su uso se simplifica enormemente con el uso de la librería pertinente, permitiendo que tras declarar el servo como una variable y adjudicarla a un pin se pueda escribir la posición directamente, en grados, en él [86].

Buzzer

El zumbador es otro elemento muy sencillo de implementar, poniendo el pin al que se encuentra conectado en un valor alto ya permite hacerlo sonar.

Si se hace uso de la función Tone se puede conseguir modificar la frecuencia del sonido. Esta función usa el timer 2 de Arduino por lo que no se podrán usar los pines 3 y 11 como PWM [87].

Como últimos elementos de prueba se encuentran las interfaces de visualización de los datos.

Pantalla LCD con controlador I2C

Gracias al uso del controlador las conexiones e instrucciones para la pantalla resultan mucho más sencillas. Para su uso también se contará con las librerías Wire.h y LiquidCrystal_I2C.h encargadas de permitir la comunicación entre el Arduino y el módulo mediante dicho protocolo.

El código para usar la pantalla, como la mayoría de códigos que hacen uso de librerías, resulta sencillo, pero puede llegar a ocupar bastantes líneas y llegar a ser engorroso. El primer paso es declarar la dirección y tamaño de la pantalla a usar (en nuestro caso la dirección es por defecto 0x27 y es una pantalla de 20, 4). Lo siguiente consiste en inicializar la pantalla y la retroiluminación, después de ello sólo es necesario familiarizarse con los comandos, resultando un proceso tan sencillo como situar el cursor en la pantalla de manera matricial y dar la orden de escribir los datos [88].

Módulo WiFi

El ESP8266 puede ser programado de diferentes maneras, muchos de ellos vienen incluidos con otro módulo que permite su conexión mediante USB al PC, normalmente estos módulos requieren soldar un botón o interruptor entre los pines de GND y GPIO 0 (ver *Figura 1.23*) para poder hacer que el módulo entre en modo programación. La programación también puede llevarse a cabo mediante el uso de un Arduino haciendo las conexiones necesarias.

De igual manera el ESP puede venir con distintas versiones o modelos de firmware. Para la opción que se ha escogido que nos permita el envío de datos a ThingSpeak es necesario contar con un firmware que admita los comandos AT (más sobre esto posteriormente). Este proceso se encuentra detallado en la web de KIO4 [89]: conexiones a realizar, modo de programación, etc. De manera más sencilla y entendible, este proceso consiste en descargar y ejecutar un archivo, sin necesidad de realizar programación. Posteriormente se nos muestran algunos ejemplos de códigos para probar el módulo.

El código final que se prueba para asegurar su funcionamiento se usa en conjunto con el sensor de temperatura y humedad. El vídeo del cual proviene [90] explica todo el proceso necesario: registro en la web de ThingSpeak [91], creación de canales, clave para el envío, conexiones, explicación comandos AT y explicación del código.

Los comandos AT son instrucciones para comunicarse con el ESP, en este ejemplo se usan algunos de ellos como puede verse en la *Figura 2.6*. Para los comandos enviados también se recibe una respuesta.

Continuando al código aparece primero la librería `SoftwareSerial.h`, la cual nos permite el uso de otros pines que no sean el 0 y el 1 para la comunicación serial, de esta manera se nos permite seguir haciendo uso del monitor serial para leer los valores del resto de elementos mediante conexión USB. También se cuenta con la librería `DHT.h` para el sensor de temperatura como ya se ha visto. Se deben definir los parámetros de la red WiFi a la que se encuentra conectado y el `apiKey` obtenido de ThingSpeak.

En el setup se debe inicializar la comunicación serial, se configura el ESP como cliente y se le pasa la información de la red para que se conecte. El tiempo que tarda en configurarse emplea una función de tiempo de respuesta, consistente en leer los datos seriales del ESP para guardarlos y poder imprimirlos.

En el loop se leen el valor de temperatura y el de humedad, esos datos se pasan a la función para mandarlo a ThingSpeak, mediante una conexión TCP a la dirección IP de ThingSpeak y puerto 80. Para mandar los datos se crea una cadena de caracteres, en este caso para enviar dos valores. Luego se aplica un delay grande para que al módulo le dé margen para la comunicación.

Tras realizar las conexiones y comprobar que los datos se envían este es el resultado obtenido para las dos variables. Los datos pueden ser presentados de varias maneras. La página posee la posibilidad de exportar los datos para su análisis con Matlab.

Figura 2.5: Prueba de envío de datos a ThingSpeak, se muestran temperatura y humedad con distintos tipos de visualizaciones.



Figura 2.6: Comandos AT para ESP8266 del ejemplo (Fuente: [90]).

Comandos AT ESP8266

Comando	Respuesta	Función
AT - Probar iniciación correcta		
AT	OK	Prueba si el módulo responde correctamente
AT+RST - Reinicia el módulo		
AT+RST	OK	Resetéa el módulo
AT+CWMODE - Modo Wifi		
AT+CWMODE=?	+CWMODE:(1-3) OK	Lista los modos válidos
AT+CWMODE?	+CWMODE:modo OK	Pregunta en que modo AP esta actualmente el módulo
AT+CWMODE=modo	OK	Establece el módulo en el modo dado 1 = Modo estación (cliente) 2 = Modo AP (huésped) 3 = Modo AP + Estación (modo dual)
AT+CWLAP - Lista APs disponibles		
AT+CIPSTART - Establece una conexión TCP o registra un puerto UDP e inicia la conexión		
AT+CIPSTART=type,addr,port	OK	Empieza una conexión como cliente (en modo conexión única) type: puede ser "TCP" o "UDP" addr: String que contenga la dirección IP remota port: String que contenga el puerto remoto
AT+CIPSTART=id,type,addr,port	OK	Empieza una conexión como cliente (En modo conexión múltiple) id: ID de la conexión (1-4)
AT+CIPSTART=?	[+CIPSTART:(id)("type"),("ip address"),(port)] OK	Lista los posibles comandos
AT + CIPCLOSE - Cierra la conexión TCP o UDP		
AT+CIPCLOSE=?	OK	
AT+CIPCLOSE=id	OK	Cierra la conexión TCP o UDP con el ID "id" en modo conexión múltiple
AT+CIPCLOSE	OK	Cierra la conexión TCP o UDP para modo de conexión única
AT+CIPSEND - Enviar datos		
AT+CIPSEND=?	OK	
AT+CIPSEND=length	SEND OK	Establece la longitud de datos a enviarse (máximo 2048). Para un envío normal (modo conexión única)
AT+CIPSEND=id.length	SEND OK	Establece la longitud de datos a enviarse en la conexión número "id". Para un envío normal (modo conexión múltiple)
AT+CIPSEND		Envía datos sin adornos cada 20ms. El módulo retorna ">" después ejecutar el comando, si se recibe el comando "+++" se regresa al modo comando.

3 Diseño de la maqueta

La maqueta se realiza a escala 1:50, una escala algo grande para lo que se suele encontrar en maquetas pero que de esta manera facilita la instalación de los distintos sensores y permite una mejor apreciación de estos.

Para su construcción se dispone de un tablero de madera de 120x80x1cm que servirá como base de la construcción de la propia nave, formada por piezas recortadas de otro tablero, esta vez de grosor 0.3cm. Para las cubiertas como el arco de la nave o el tejado se usará metacrilato o láminas de plástico transparente para permitir ver su interior. Como soportes adicionales, para marcar la curvatura de la nave y las columnas de soporte de los frontales se usarán varillas de madera.

Se comienza sobre el tablero base marcando las líneas exteriores tomando el *Plano 1* como referencia. Posteriormente se cortan las piezas del tablero más fino, es necesario tomar las medidas de los planos 1 y 2, obteniendo las medidas reales se procede a realizar la escala, obteniendo la tabla siguiente:

Tabla 3.1: Dimensiones de las piezas principales que conforman la nave.

PANEL	UNIDADES	DIMENSIONES REALES (m)		DIMENSIONES MAQUETA 1:50 (cm)	
Arco nave	1	22,8	39,9	45,6	79,8
Tapa nave frontal	1	3,95	15,9	7,9	31,8
Tapa nave trasera	1	3,2	14,7	6,4	29,4
Pared nave	2	1,4	39,9	2,8	79,8
Pared derecha cobertizo	1	5	4,38	10	8,7
Pared izquierda cobertizo	1	7	5	14	10
Pared fondo cobertizo	1	15,96	5	31,9	10
Techo cobertizo	1	7,08	18,1	14,1	36,2

Figura 3.1: Piezas principales en su posición correspondiente antes del pegado.



Todas las piezas se cortan de manera rectangular, realizando posteriormente cortes para dar la curvatura o caída necesarias. Posteriormente se colocan en su posición adheridas mediante pegamento para maquetas, empezando desde el interior para no obstaculizarse al pegar la pieza siguiente.

Antes de colocar la cubierta se procede con la colocación y cableado de componentes. La normativa aplicable a este apartado viene reflejada en el Reglamento Electrotécnico para Baja Tensión, conocido por sus siglas REBT [93], acogiéndose al *ITC-BT-36 INSTALACIONES A MUY BAJA TENSIÓN* que dicta grosso modo: “*Estas instalaciones comprenden aquellas cuya tensión nominal no excede 50V en alterna o 75V en continua*”.

Para realizar las conexiones de la maqueta se emplean cables Dupont o jumper en las conexiones cercanas y cable de altavoces de sección 1.5mm para las conexiones más lejanas. Para su conexión con los módulos el cable puede ir soldado al patillaje directamente, sujeto mediante un tornillo o soldado su extremo para su uso como un jumper y así evitar el calentamiento del componente al soldarlo. Se hará uso de la protoboard para facilitar el conexionado.

Para alimentar el circuito se puede hacer uso de una fuente de alimentación directa a la red o de una batería de 12V, permitiendo su funcionamiento sin depender de la red y pudiendo mantener su funcionamiento ante un corte. Para la carga de la batería se puede hacer uso de una placa solar que la cargue durante el día. La transformación de la tensión entrante a la necesaria para los módulos se hace de 3 maneras distintas:

- Placa mb102 para protoboard (*Tabla 6.13*), salidas de 5V y 3.3V hasta 700mA.
- Regulador LM7805 (*Tabla 6.14*), salida de 5V hasta 1.5A.
- Driver de motores (*Tabla 6.6*), posee su propio regulador para alimentarse a 5V.

Para no exigir mucha corriente a la placa de la protoboard esta es la distribución:

Tabla 3.2: Consumos aproximados

Elemento	Consumo (mA)	Unidades	Total (mA)	PLACA	REGULADOR
Temp y hume	1,5	2	3	3	
Gases	160	2	320		320
Ruido	6	1	6	6	
Caudal	15	1	15	15	
Servo	360	1	360		360
Buzzer	30	1	30	30	
LCD	200	1	200	200	
Wifi	100	1	100	100	
TOTAL (mA)			1034	354	680

En el caso del regulador LM7805 se acopla una lámina metálica y un ventilador de tipo turbina para asegurar su correcta refrigeración durante el período de calentamiento de los sensores de gases.

Figura 3.2: Imagen cenital de la maqueta.



Descripción del sistema (izquierda a derecha): servomotor para las ventanas, ventilador A, sensor de amoníaco (arriba) y de dióxido de carbono (abajo), sensores de temperatura, electroválvula, ventilador B, caudalímetro, regulador de 5V, protoboard y LDC.

Imágenes más detalladas se muestran a continuación.

Figura 3.3: Servomotor cerrado y ventanas.



Figura 3.4: Sensores de gases y temperatura y humedad (con ampliación).

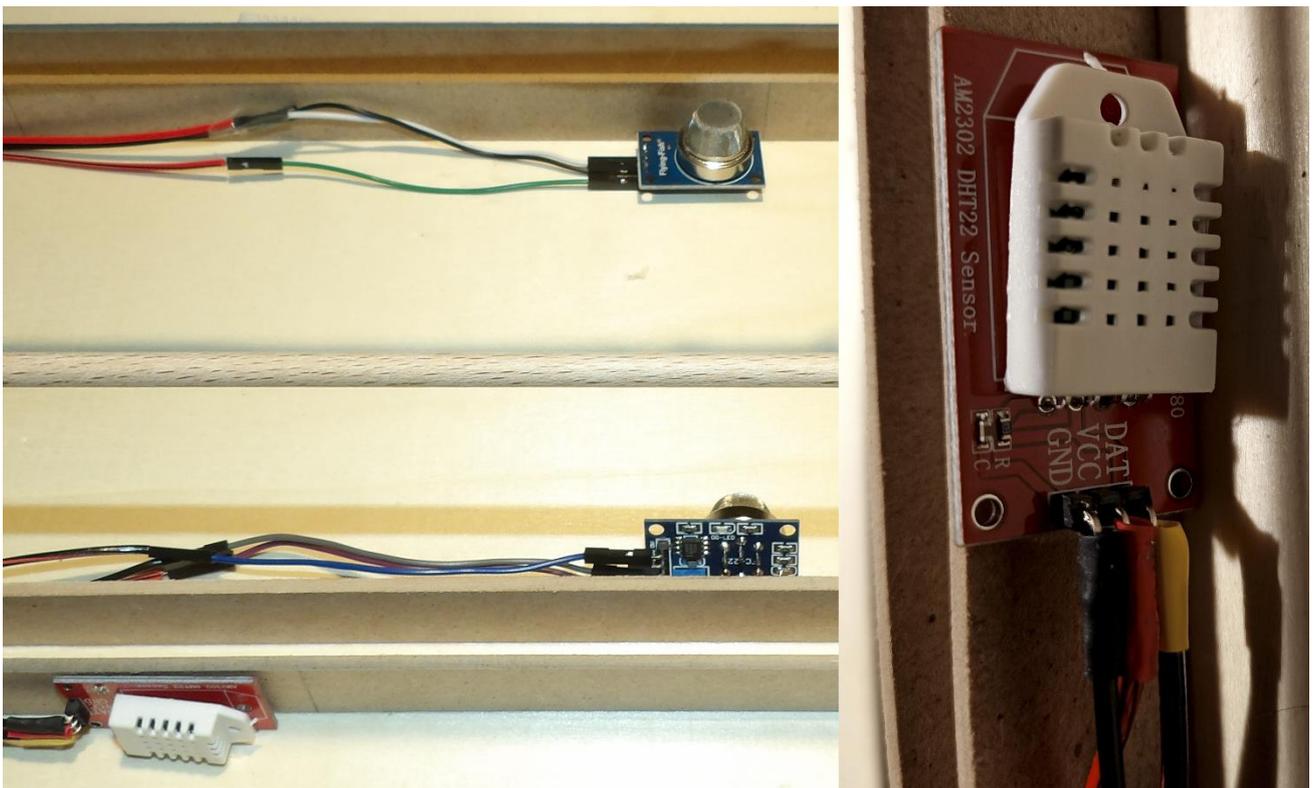
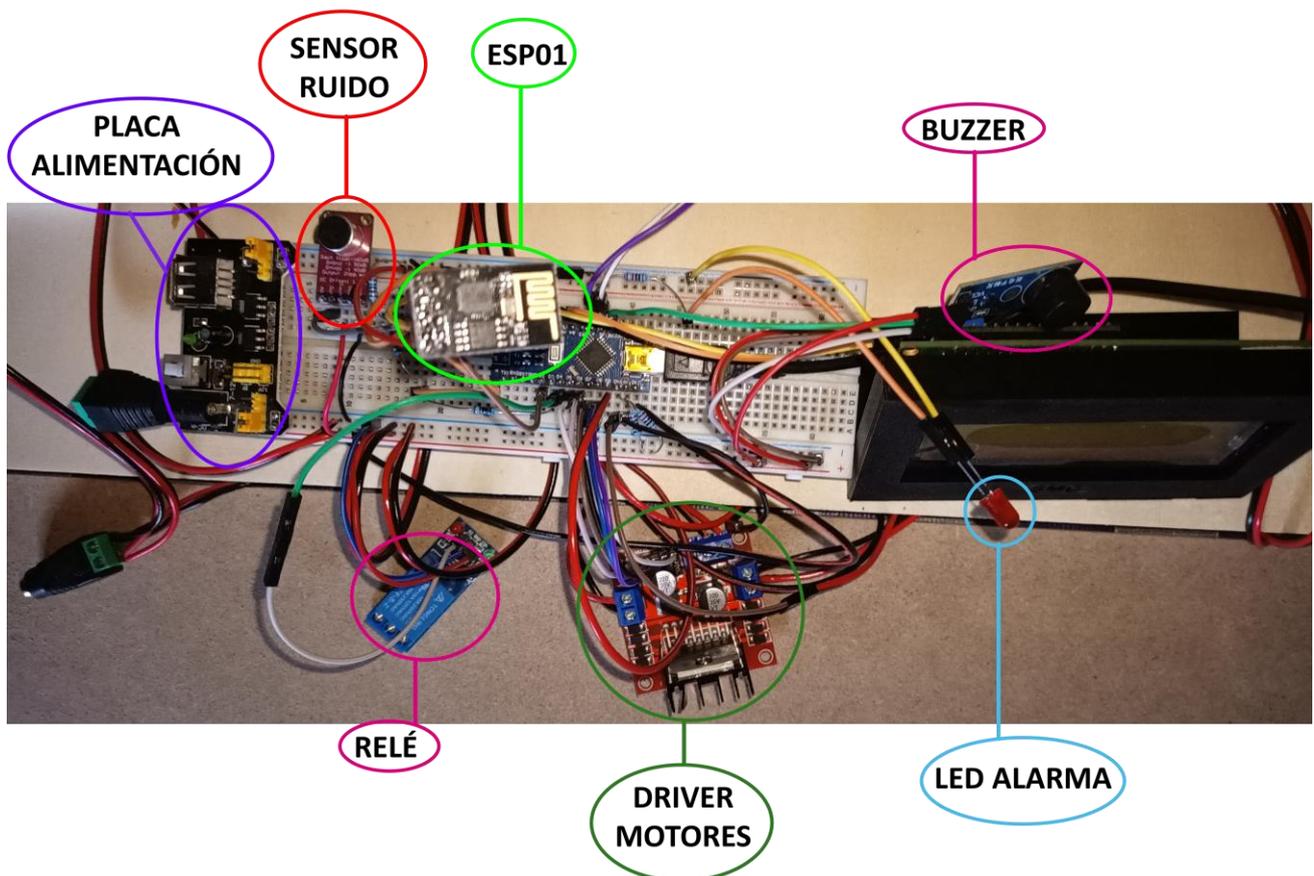


Figura 3.5: Electroválvula y caudalímetro.



Figura 3.6: Detalle de los elementos en la protoboard.



Finalmente se conecta la fuente de alimentación general y se alimenta el Arduino mediante USB para así también comprobar por el monitor serial el funcionamiento del programa (el funcionamiento puede encontrarse de manera simplificada en el *Anexo III*. Imágenes de su funcionamiento pueden verse a continuación.

Figura 3.7: Inicio del programa visto desde el monitor serial.

```
COM3
Iniciando
AT+CWMODE=1

OK
AT+CWJAP="Nombre red","Clave red"
Setup completed
Temperature: 22.10 *C
Humidity: 55.25 %      Contador: 1
Amoniac: 17 ppm
CO2: 233 ppm Caudal: 0.00 l/min
Ruido: 38.72 dB
AT+CIPSTART="TCP","184.106.153.149",80
AT+CIPSEND=118
GET /update?api_key=VE3F7J6QKDYCT69D&field1=22.10&field2=55.25&field3=17.00&field4=233.00&field5=0.00&field6=38.72

Temperature: 22.15 *C
Humidity: 55.25 %      Contador: 2
Amoniac: 19 ppm
CO2: 212 ppm Caudal: 0.00 l/min
Ruido: 39.19 dB
AT+CIPSTART="TCP","184.106.153.149",80
AT+CIPSEND=118
GET /update?api_key=VE3F7J6QKDYCT69D&field1=22.15&field2=55.25&field3=19.00&field4=212.00&field5=0.00&field6=39.19
```

Aquí se muestran los dos primeros ciclos del programa en funcionamiento, al principio se puede ver la configuración del módulo wifi como cliente, se le transmiten el nombre y la clave de la red y cuando acaba su configuración nos devuelve un Setup completed.

A continuación, se nos muestran los valores de las diferentes variables y el valor de contador de la temperatura. También se aprecia que el valor de caudal se mantiene a 0, esto es porque se ha preferido no involucrar líquidos cerca de la maqueta.

Figura 3.8: Monitor serial, apertura de ventanas.

```
Temperature: 22.20 *C
Humidity: 55.25 %      Contador: 5
Amoniac: 18 ppm
CO2: 202 ppm Caudal: 0.00 l/min
Ruido: 40.73 dB
AT+CIPSTART="TCP","184.106.153.149",80
AT+CIPSEND=118
GET /update?api_key=VE3F7J6QKDYCT69D&field1=22.20&field2=55.25&field3=18.00&field4=202.00&field5=0.00&field6=40.73

Abrir ventanas
Temperature: 22.20 *C
Humidity: 55.20 %      Contador: 6
Amoniac: 19 ppm
CO2: 198 ppm Caudal: 0.00 l/min
Ruido: 38.76 dB
AT+CIPSTART="TCP","184.106.153.149",80
AT+CIPSEND=118
GET /update?api_key=VE3F7J6QKDYCT69D&field1=22.20&field2=55.20&field3=19.00&field4=198.00&field5=0.00&field6=38.76
```

En la imagen superior se aprecia que, debido a la temperatura de 22.2°C, el contador ha ido aumentando hasta ser mayor que 5, momento en que el sistema decide que es momento de abrir las ventanas. Al ser la primera inicialización el contador seguirá aumentando hasta estabilizarse.

La apertura de las ventanas se produce mediante el brazo del servomotor que las empuja hacia arriba.

Figura 3.9: Servomotor abriendo las ventanas.



Figura 3.10: Monitor serial, apertura de ventanas e inicio del primer motor al 50%.

```
Abrir ventanas
Motor 1
Temperature: 22.35 *C
Humidity: 55.00 %      Contador: 15
Amoniaco: 17 ppm
CO2: 200 ppm Caudal: 0.00 l/min
Ruido: 39.43 dB
AT+CIPSTART="TCP", "184.106.153.149", 80
AT+CIPSEND=118
GET /update?api_key=VE3F7J6QKDYCT69D&field1=22.35&field2=55.00&field3=17.00&field4=200.00&field5=0.00&field6=39.43
```

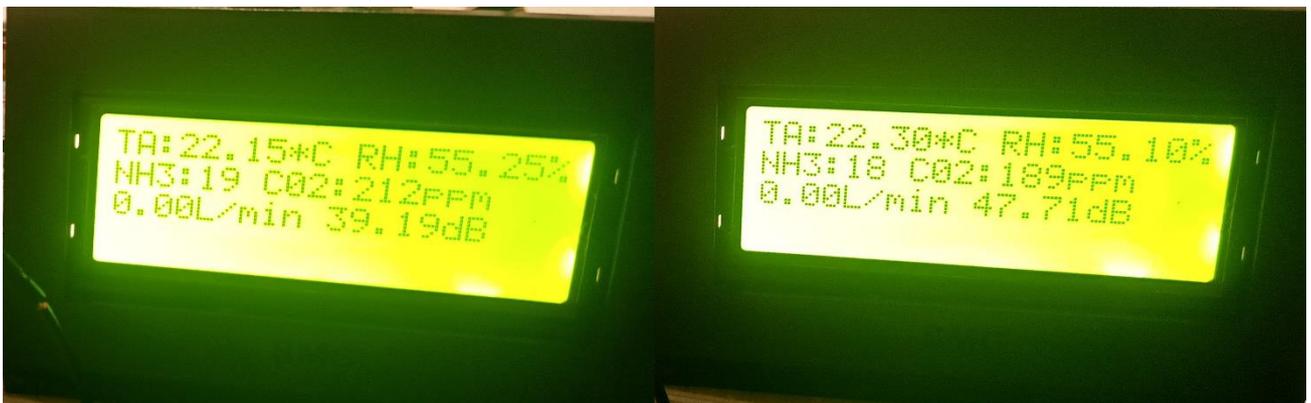
El poder ver si el ventilador se encuentra funcionando resulta difícil plasmarlo en imagen. Prestando atención a la figura siguiente el ventilador de la izquierda, siendo el ventilador A en la maqueta y situado más atrás en la nave, no se pueden ver las palas, cosa que sí sucede en el del lado derecho y que se encuentra parado.

Figura 3.11: Apreciación de los ventiladores



Para fines de configuración y comprobaciones sí que resulta útil el uso del monitor serial, mientras que para el usuario resulta de más utilidad presentar los datos sobre la LCD.

Figura 3.12: Datos mostrados en la LCD durante el funcionamiento.



A esta información mostrada sería también interesante añadir en la última fila la variable cuentaAlarma, indicándonos si se ha producido un error momentáneo que haya hecho saltar la alarma pero que posteriormente se haya solventado sin necesidad de acudir. Esta última línea puede servir para otra información que pueda ser requerida, permitiendo cierto grado de personalización en el sistema, y por supuesto, no se está limitado a sólo 4 líneas de información.

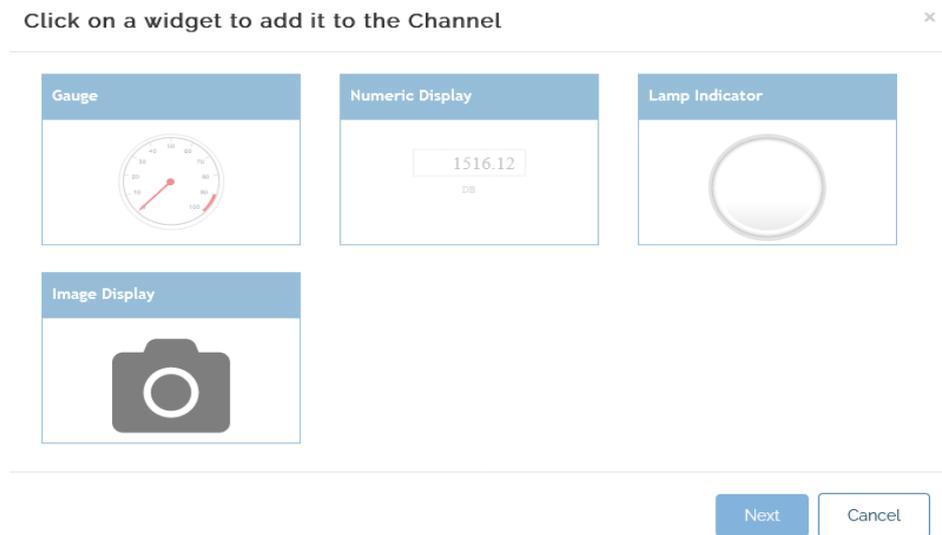
El otro método de visualización de la información es mediante la página de ThingSpeak. En su arranque el sistema no presenta los valores reales, es necesario dejarlo en funcionamiento un tiempo en el que las variables se estabilicen. Los picos presentados en las variables de los gases pueden deberse a que se les ha dejado poco tiempo de calentamiento y debería ser extendido. Al final de algunas de las gráficas se encuentran grandes picos, resultantes de haberles movido antes de su desconexión.

Figura 3.13: Datos enviados a ThingSpeak.



De igual manera que con los datos en el display se pueden agregar más gráficas en la misma página como la alarma, siendo en este caso oportuno otro widget ofrecido en ThingSpeak, el de indicador de luz, permitiendo que se encienda si el valor que recibe es superior al que le indiquemos en la configuración.

Figura 3.14: Widgets extra ThingSpeak.



Para comprobar si la alarma cumple con su función se procede a desconectar uno de los sensores de temperatura, al ser uno de los cables más accesibles, resultando en obtener a los pocos segundos el error correspondiente, seguido del sonido de alarma y el led parpadeante en rojo.

Al contemplar el mensaje en pantalla se puede apreciar un error en la programación al momento de imprimir los errores. Éstos están hechos para que el mensaje aparezca en la fila 4, sin embargo, para conseguir esto se debe colocar en el código que sea en la línea 3, ya que se empieza el conteo en el 0.

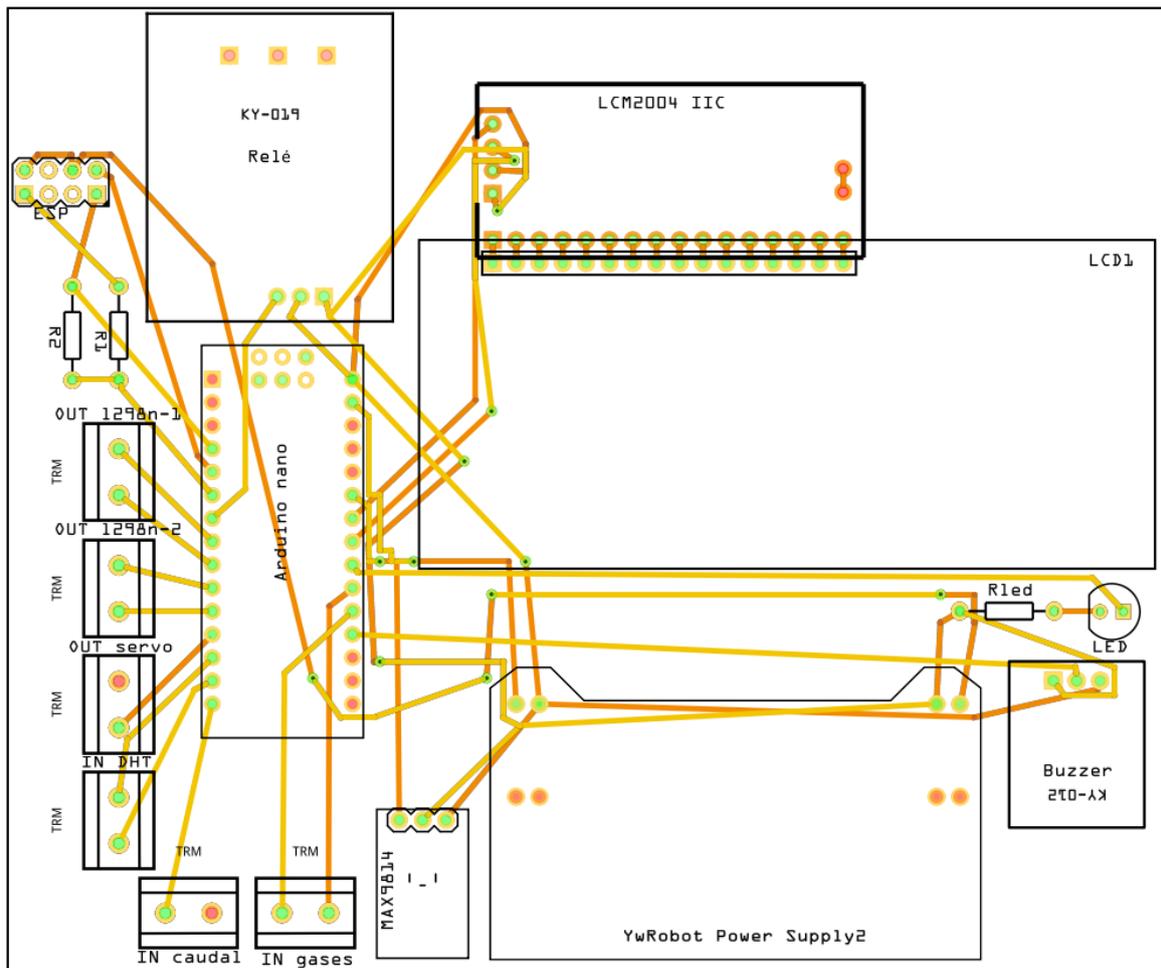
Figura 3.15: Mensaje de error y led de alarma encendido.



En la maqueta actual, siguiendo la idea de la modularidad como estándar en este tipo de instalaciones, cada elemento se encuentra separado del resto; sin embargo, es posible realizar una agrupación de algunos de ellos en una misma placa central a la que el resto de módulos puedan conectarse fácilmente. Los dispositivos que pueden agruparse en una PCB serían: el procesador, el display y su controlador, el módulo de comunicaciones Wifi, el led indicador, las resistencias, el relé, el sensor de ruido ambiente y el zumbador de la alarma (teniendo precaución que estos dos elementos no entren en conflicto entre sí).

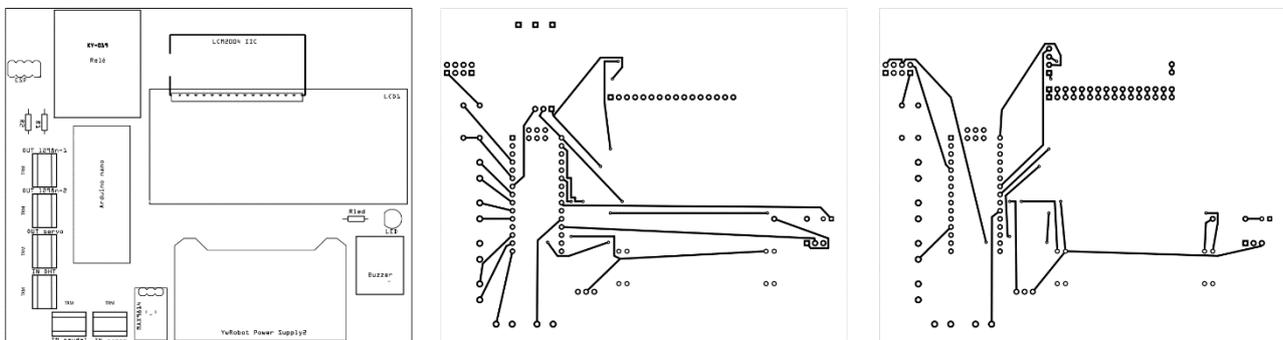
Aunque la PCB puede ser realizada con Eagle como se ha hecho para los conexionados, debido a una incompatibilidad con ciertas footprints se ha optado por hacer uso del programa Fritzing [93], un programa más enfocado al ámbito educativo y que por ello resulta muy visual, además de poder encontrarse multitud de componentes creados por los usuarios. Tras el conexionado de los componentes antes mencionados en la pestaña protoboard se pasa a la pestaña de PCB, en la que se puede proceder a ordenar los componentes, pudiendo elegir un tamaño de placa mayor o menor. El programa nos ofrece las opciones de enrutado automático o manual, trabajando en dos caras. En nuestro caso con la opción auto el resultado en vista de programa sería:

Figura 3.16: PCB vista en el programa.



En opciones de exportación para producción el programa proporciona una serie de archivos con su diseño separado en capas.

Figura 3.17: Formato de salida en capas de la PCB.



De izquierda a derecha se pueden ver componentes, capa superior y capa inferior de las pistas de cobre.

La norma más usada en el diseño de PCB es la IPC 2221 [95], más específicamente la 2221B, que puede ser usada en conjunto con otras normas. La norma nos define los pasos y elementos a considerar a la hora de proceder con la creación del circuito impreso. Atendiendo a los materiales se deberá tener en cuenta su finalidad y condiciones ambientales del lugar de instalación, si va a estar sometida a esfuerzos o vibraciones, etc. Por supuesto al ser un circuito deberá cumplir condiciones como distancias de separación en función del voltaje para evitar arcos eléctricos o cumplir con un ancho determinado para manejar la corriente, tampoco hay que olvidarse de las frecuencias.

Como mención final al desarrollo de la maqueta se puede analizar el coste aproximado de ésta, empezando con los componentes nos encontramos:

Tabla 3.3: Precio componentes del sistema.

Elemento	Unidades	Precio ud	Envío	Precio total	Observaciones
Protoboard	1	3,35 €	0,00 €	3,35 €	Conexión sencillo de las pruebas
LM7805	2	0,32 €	0,00 €	0,63 €	Regulador de tensión 5V para los sensores MQ
Ventilador 2510	2	2,75 €	0,00 €	5,50 €	
Fuente de alimentación 3-24V 3A	1	12,21 €	0,00 €	12,21 €	Alimentación del sistema
Módulo relé	1	2,50 €	0,00 €	2,50 €	Activación de la electroválvula
MQ-137	1	15,82 €	2,91 €	18,73 €	Medición de amoniaco
MQ-135	1	1,32 €	0,70 €	2,02 €	Medición de la calidad del aire
Buzzer	1	0,33 €	0,58 €	0,91 €	Sonido alarma
LCD2004+I2C	1	4,84 €	0,00 €	4,84 €	Pantalla
ESP8266-ESP01	1	1,81 €	0,00 €	1,81 €	Módulo de comunicación Wifi
L298N	1	1,41 €	0,00 €	1,41 €	Controlador motores DC
YF-S201	1	3,43 €	0,00 €	3,43 €	Sensor caudal
MB102	1	0,88 €	0,60 €	1,48 €	Convertor voltaje para protoboard
DHT22	2	2,55 €	0,84 €	6,78 €	Sensor temperatura y humedad
Adaptador Jack	3	0,47 €	1,00 €	4,41 €	Conexión rápido
Electroválvula	1	3,42 €	1,96 €	5,38 €	Cierre del suministro de agua
Arduino nano	1	5,21 €	1,51 €	6,72 €	Unidad de procesamiento
SG-90	1	1,92 €	0,00 €	1,92 €	Cierre/apertura ventanas

TOTAL	84,03
--------------	-------

Sumado a ese total, el precio del cable y jumpers usados, que se encuentra en torno a los 10€, además del precio de los tableros, varillas y plásticos de la estructura que suman 15€ más, el precio final de la maqueta se situaría en torno a los 100€. Para una instalación real con sensores especializados el presupuesto aumenta considerablemente.

4 Mejoras y conclusiones

La primera mejora posible, dado que se trata de un sistema que pretende funcionar continuamente en operación y alimentado con una batería, sería contar con algún tipo de ahorro de energía, en Arduino esto puede ser logrado mediante el modo sleep [96].

El procesador repite el mismo ciclo una y otra vez, en nuestro caso al tratarse de variables que fluctúan lentamente no necesitan un intervalo de medida tan corto. Para solventar este problema el modo sleep se encarga de suspender la ejecución del código hasta ser despertado nuevamente por hardware o software.

En el Arduino nano, más concretamente en su procesador Atmega328P, se encuentran seis modos con estos consumos: Idle (15mA), ADC Noise Reduction (6.5mA), Power-save (1.62mA), Power-down (0.36mA), Standby (0.84mA), Extended Standby (1.62mA).

Como de costumbre para facilitar el uso de Arduino se encuentra disponible una librería que nos permite “dormir” el Arduino, esta librería es la Low-Power [97]. Si por un error se deja el procesador dormido permanentemente la solución consiste en reprogramarlo con el ejemplo del IDE del led parpadeando.

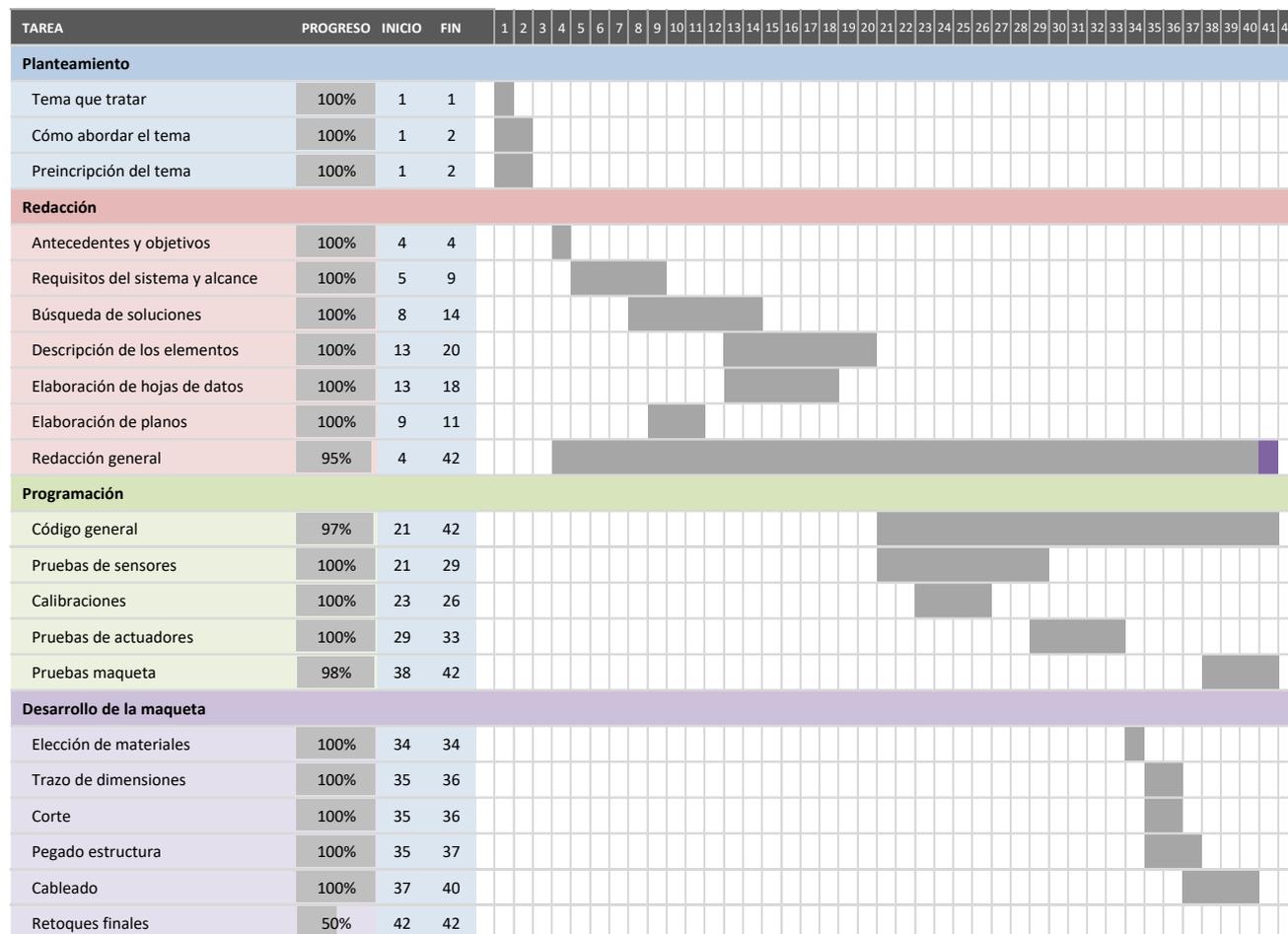
Con la librería se hace uso del Watch Dog Timer, una interrupción periódica que se dispara al llegar el contador a 0. Los tiempos aceptados van desde los 15ms a los 8s y un modo para siempre. Un ejemplo de cómo sería activar este sistema: “*LowPower.powerDown(SLEEP_8S, ADC_OFF, BOD_OFF);*” se le indica que debe dormir 8s, se apaga el conversor Analógico-Digital y el circuito de Brown Out Detection, usado para detectar niveles bajos de tensión, útil para cuando la batería que lo alimenta se va agotando y así proteger sus circuitos.

Para conseguir tiempos de suspensión mayores se debe hacer uso del modo sleep *forever* y despertarlo mediante una interrupción externa, bien sea un pulsador o un reloj.

Otra mejora de cara al usuario final sería la creación de una app que permita su visualización en el Smartphone del usuario sin tener que entrar en la página web, siendo además más confidencial.

El tiempo del proyecto se muestra en un diagrama de Gantt. Dado que el proyecto se ha llevado a cabo de manera discontinua tanto en días como en horas de trabajo se hace una aproximación a lo que serían días trabajados durante una jornada normalizada de 8h y se muestran seguidos.

Figura 4.1: Diagrama de Gantt del proyecto.



Teniendo en cuenta todo el trabajo realizado a lo largo del proyecto, podemos concluir que se ha obtenido una mejor perspectiva de los requisitos de bienestar animal que serán de gran interés, tanto para ganaderos como para consumidores, en los años venideros. Por otro lado, los sistemas de gestión, análisis, monitorización, automatizaciones, etc. desempeñarán también un gran papel en la mejora de la producción y calidad, lo que a su vez pueda resultar en un beneficio económico para los productores.

5 Normativa y referencias

5.1 Bibliografía

- [1] R. Muñoz, «Bienestar animal», Spei Domus, 2014. [En línea]. Disponible en: <https://www.mapa.gob.es/es/ganaderia/temas/produccion-y-mercados-ganaderos/bienestanimal/>. [Accedido: 18-jun-2022].
- [2] Redacción oviespana.com, «Las instalaciones de ovino deben asegurar unas condiciones ambientales y de bienestar que permitan a los animales comer, beber, expresar su comportamiento natural y descansar de forma apropiada - Oviespaña», 2020. [En línea]. Disponible en: <https://www.oviespana.com/Articulos/297503-instalaciones-ovino-deben-asegurar-condiciones-ambientales-bienestar-permitan-animales.html>. [Accedido: 18-jun-2022].
- [3] «Temperature Humidity Index: what you need to know about it». [En línea]. Disponible en: <https://www.pericoli.com/EN/news/120/Temperature-Humidity-Index-what-you-need-to-know-about-it.html>. [Accedido: 18-jun-2022].
- [4] Ministerio de Agricultura Alimentación y Medio Ambiente, «CH4 (Metano) | PRTR España», 2007. [En línea]. Disponible en: <http://www.prtr-es.es/CH4-metano,15588,11,2007.html>. [Accedido: 18-jun-2022].
- [5] PRTR ESPAÑA, «NH3 (Amoniac) | PRTR España», 2007, 2007. [En línea]. Disponible en: <http://www.prtr-es.es/NH3-amoniaco,15593,11,2007.html>. [Accedido: 18-jun-2022].
- [6] «Keys to reduce ammonia levels in the farm and improve animal welfare». [En línea]. Disponible en: <https://www.veterinariadigital.com/en/articulos/keys-to-reduce-ammonia-levels-in-the-farm-and-improve-animal-welfare-and-performance/>. [Accedido: 18-jun-2022].
- [9] E. Y. Sector Agrario La Calidad Del Aire y M. Josefa Lueso Sordo, «GANADERÍA Y CALIDAD DEL AIRE», 2019.
- [10] «Calculadora de conversión de dBm a mW», RapidTables I RT. [En línea]. Disponible en: https://www.rapidtables.org/convert/power/dBm_to_mW.html. [Accedido: 18-jun-2022].
- [11]I. Colbrand, «¿A qué equivalen los diferentes niveles de decibelios?», Allpe, 2021. [En línea]. Disponible en: <http://www.allpe.com/acustica/ingenieria-acustica/mediciones-acusticas/a-que-equivalen-los-diferentes-niveles-de-decibelios/>. [Accedido: 18-jun-2022].

- [12] «Análisis estacional: León, Virgen del Camino - Otoño 2021 - Temperatura - Agencia Estatal de Meteorología - AEMET. Gobierno de España». [En línea]. Disponible en: http://www.aemet.es/es/serviciosclimaticos/vigilancia_clima/analisis_estacional?w=1&l=2661&datos=temp. [Accedido: 18-jun-2022].
- [13] «Análisis estacional: León, Virgen del Camino - Otoño 2021 - Temperatura - Agencia Estatal de Meteorología - AEMET. Gobierno de España». [En línea]. Disponible en: http://www.aemet.es/es/serviciosclimaticos/vigilancia_clima/analisis_estacional?w=1&l=2661&datos=temp. [Accedido: 18-jun-2022].
- [14] «Valle de Mansilla - Weathercloud | Red mundial de estaciones meteorológicas». [En línea]. Disponible en: <https://app.weathercloud.net/d6726304567#current>. [Accedido: 18-jun-2022].
- [15] matelec, «Grados de Protección IP», Grados de protección IP, 2019. [En línea]. Disponible en: https://ingemecanica.com/tutoriales/grado_proteccion_ip.html. [Accedido: 18-jun-2022].
- [17] «Grados de protección IP». [En línea]. Disponible en: <https://elblogenergia.com/decoracion/iluminacion/grados-proteccion-ip>. [Accedido: 18-jun-2022].
- [18] «Productos - Spanish». [En línea]. Disponible en: <https://www.dol-sensors.com/es/productos/>. [Accedido: 18-jun-2022].
- [19] dol sensors, «DOL 15 temperature sensor».
- [20] dol sensors, «DOL 104 humidity sensor (0-3/0-5/0-10 V)».
- [21] dol sensors, «DOL 114 temperature and humidity sensor (0-5/0-10 V)».
- [22] dol sensors, «DOL 19 carbon dioxide sensor».
- [23] dol sensors, «DOL 53 Ammonia Sensor (NH 3) Climate sensor for measuring ammonia level in livestock houses».
- [24] «iDOL 63 User Guide - dol sensors». [En línea]. Disponible en: <https://www.dol-sensors.com/idol-63-user-guide/>. [Accedido: 18-jun-2022].
- [25] «OmniFarm User Guide - DOL Sensors». [En línea]. Disponible en: https://www.dol-sensors.com/omnifarm-user-guide/#omnifarm_app. [Accedido: 18-jun-2022].
- [26] Envira IOT, «Sensor para explotaciones ganaderas Nanoenvi Farm», 2020. [En línea]. Disponible en: <https://enviraiot.es/nanoenvi-farm/>. [Accedido: 18-jun-2022].

- [27] M. Gracia, «¿Qué es IoT (Internet Of Things)? | Deloitte España», January 8, 2019. [En línea]. Disponible en: <https://www2.deloitte.com/es/es/pages/technology/articles/IoT-internet-of-things.html%0Ahttps://www2.deloitte.com/es/es/pages/technology/articles/IoT-internet-of-things.html#>. [Accedido: 18-jun-2022].
- [28] «IoT Remote Monitoring Products and Solutions». [En línea]. Disponible en: <https://www.monnit.com/products/>. [Accedido: 18-jun-2022].
- [29] M. Corp, «Remote Monitoring for Business Wireless Temperature Sensors General Description».
- [30] «Ethernet & Cellular Gateways for Wireless IoT Sensors». [En línea]. Disponible en: <https://www.monnit.com/products/gateways/>. [Accedido: 18-jun-2022].
- [31] «DirectIndustry - El marketplace B2B de la industria: sensores, automatismos, motores, bombas, manipulación, embalajes, etc.» [En línea]. Disponible en: <https://www.directindustry.es/>. [Accedido: 18-jun-2022].
- [32] «La cultura del DIY es algo más que el Craft - El invernadero creativo». [En línea]. Disponible en: <https://www.elinvernaderocreativo.com/la-cultura-del-diy-es-algo-mas-que-el-craft/>. [Accedido: 18-jun-2022].
- [33] «Movimiento “maker”: más que “do it yourself” - Fundación Aquae». [En línea]. Disponible en: <https://www.fundacionaquae.org/wiki/movimiento-maker-mucho-mas-que-do-it-yourself-2/>. [Accedido: 18-jun-2022].
- [34] <https://www.arduino.cc/>, «Arduino - Home», Hardware, 2016. [En línea]. Disponible en: <https://www.arduino.cc/>. [Accedido: 18-jun-2022].
- [35] «Learn with the Raspberry Pi Foundation». [En línea]. Disponible en: <https://www.raspberrypi.org/>. [Accedido: 18-jun-2022].
- [36] L. del Valle Hernández, «Sensor de temperatura, escoge el mejor para tus proyectos con Arduino», ProgramarFácil.com, 2017. [En línea]. Disponible en: <https://programarfácil.com/podcast/82-escoger-mejor-sensor-temperatura-arduino/>. [Accedido: 18-jun-2022].
- [37] «Descripción y funcionamiento del Bus I2C | Robots Didácticos», Robots Didácticos, 2018. [En línea]. Disponible en: <http://robots-argentina.com.ar/didactica/descripcion-y-funcionamiento-del-bus-i2c/>. [Accedido: 18-jun-2022].

- [38] «HR202 Sensor de humedad - Geek Factory». [En línea]. Disponible en: <https://www.geekfactory.mx/tienda/sensores/hr202-sensor-de-humedad/>. [Accedido: 18-jun-2022].
- [39] ams AG, «ams Datasheet CCS811 Ultra-Low Power Digital Gas Sensor for Monitoring Indoor Air Quality».
- [40] Sensirion AG, «Datasheet SGP30 Sensirion Gas Platform», p. 16, 2018.
- [41] «Sensor de Sonido analógico y digital para arduino», 2018. [En línea]. Disponible en: <https://naylampmechatronics.com/sensores-luz-y-sonido/34-sensor-de-sonido-ky-038.html>. [Accedido: 18-jun-2022].
- [42] «Módulo Micrófono con Amplificador MAX4466». [En línea]. Disponible en: <https://naylampmechatronics.com/sensores-luz-y-sonido/390-modulo-microfono-con-amplificador-max4466.html>. [Accedido: 18-jun-2022].
- [43] L. Llamas, «Medir sonido con Arduino y micrófono amplificado MAX9812», 2016. [En línea]. Disponible en: <https://www.luisllamas.es/medir-sonido-con-arduino-y-microfono-amplificado-max9812/>. [Accedido: 18-jun-2022].
- [44] Naylamp Mechatronics, «Tutorial uso de servomotores con arduino», Naylamp Mechatronics, 2017. [En línea]. Disponible en: https://naylampmechatronics.com/blog/33_Tutorial-uso-de-servomotores-con-arduino-.html. [Accedido: 18-jun-2022].
- [45] «Burn-in en monitor OLED: un problema a tener en cuenta». [En línea]. Disponible en: <https://hardzone.es/tutoriales/componentes/burn-in-monitor-oled-problemas/>. [Accedido: 18-jun-2022].
- [46] A. S. L. Geovani, «Tipos De Placas Arduino». [En línea]. Disponible en: <https://programacionenjavaups.wordpress.com/tipos-de-placas-arduino/>. [Accedido: 18-jun-2022].
- [47] «Arduino - Modelos más usados - MBrobotics». [En línea]. Disponible en: <https://mbrobotics.es/blog/tipos-de-arduinos/>. [Accedido: 18-jun-2022].
- [48] T. Liu, «Digital Humidity and Temperature sensor», Adfruit, pp. 1-5, 2016.
- [49] «MQ-137». [En línea]. Disponible en: https://components101.com/sites/default/files/component_datasheet/MQ-137_Sensor_Datasheet.pdf. [Accedido: 18-jun-2022].

- [50] «MQ-135». [En línea]. Disponible en: https://www.electronicoscaldas.com/datasheet/MQ-135_Hanwei.pdf. [Accedido: 18-jun-2022].
- [51] M. Integrated, «Microphone Amplifier with AGC and Low-Noise Microphone Bias MAX9814 Microphone Amplifier with AGC and Low-Noise Microphone Bias Absolute Maximum Ratings», Datasheet, 2009. [En línea]. Disponible en: <http://datasheets.maximintegrated.com/en/ds/MAX9814.pdf>. [Accedido: 18-jun-2022].
- [52] «YF-S201». [En línea]. Disponible en: https://components101.com/sites/default/files/component_datasheet/YF-S201-Datasheet.pdf. [Accedido: 18-jun-2022].
- [53] «Sensor de efecto Hall: todo lo que necesitas saber para tus proyectos con Arduino | Hardware libre». [En línea]. Disponible en: <https://www.hwlibre.com/sensor-de-efecto-hall/>. [Accedido: 18-jun-2022].
- [54] Acoptex, «Basics: Project 071a Water flow sensor YF-S201», Acoptex, 2018. [En línea]. Disponible en: <http://acoptex.com/project/359/basics-project-071a-water-flow-sensor-yf-s201-at-acoptexcom/#sthash.Z1ziNWXC.KLxvhFV8.dpbs>. [Accedido: 18-jun-2022].
- [55] [1] R. Rittenberry, «Hands-on technology: L298N Dual H-Bridge Motor Driver», Occup Health Saf, 2005. [En línea]. Disponible en: [http://www.handsontec.com/dataspecs/L298N Motor Driver.pdf](http://www.handsontec.com/dataspecs/L298N%20Motor%20Driver.pdf). [Accedido: 18-jun-2022].
- [56] «PWM Arduino, Pulse Width Modulation, Qué es. | Descubrearduino.com». [En línea]. Disponible en: <https://descubrearduino.com/pwm/>. [Accedido: 18-jun-2022].
- [57] L. Llamas, «Controlar motores de corriente continua con Arduino y L298N», Luisllamas.Es, 2016. [En línea]. Disponible en: <https://www.luisllamas.es/arduino-motor-corriente-continua-l298n/>. [Accedido: 18-jun-2022].
- [58] S. J. Town y D. Guan, «AFB02512LAYBF». [En línea]. Disponible en: <https://www.deltafan.com/Download/Spec/AFB02512LAYBF.pdf>. [Accedido: 18-jun-2022].
- [59] ProtoSupplies, «Servo Motor Micro SG90 - ProtoSupplies». [En línea]. Disponible en: <https://protosupplies.com/product/servo-motor-micro-sg90/>. [Accedido: 18-jun-2022].
- [60] «Buzzer». [En línea]. Disponible en: [https://components101.com/sites/default/files/component_datasheet/Buzzer Datasheet.pdf](https://components101.com/sites/default/files/component_datasheet/Buzzer%20Datasheet.pdf). [Accedido: 18-jun-2022].

- [61] Naylamp Mechatronics SAC, «Módulo Relay 1CH 5VDC», 2021. [En línea]. Disponible en: <https://naylampmechatronics.com/drivers/297-modulo-relay-1-canal-5vdc.html>. [Accedido: 18-jun-2022].
- [62] Admin, «¿Qué es un sistema HMI y para qué sirve? | AUTYCOM», Autycom, 2021. [En línea]. Disponible en: <https://www.autycom.com/que-es-un-sistema-hmi/>. [Accedido: 18-jun-2022].
- [63] Tinsharp, «Specification for Approval APPROVED BY CHECKED BY PREPARED BY», Adafuit website, pp. 1-18, 2021.
- [64] Handson Technology, «Handson Technology User Guide I2C Serial Interface 20x4 LCD Module», Handson Tech, 2021.
- [65] «LCD con Arduino texto en movimiento paso a paso». [En línea]. Disponible en: <https://programarfacil.com/blog/arduino-blog/texto-en-movimiento-en-un-lcd-con-arduino/>. [Accedido: 18-jun-2022].
- [66] «ESP8266EX Datasheet». [En línea]. Disponible en: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf. [Accedido: 18-jun-2022].
- [67] Prometec, «Modelos ESP8266 | Tienda y Tutoriales Arduino». [En línea]. Disponible en: <https://www.prometec.net/modelos-esp8266/>. [Accedido: 18-jun-2022].
- [68] CodigoElectronica, «ESP8266 esp01 datasheet | CodigoElectronica». [En línea]. Disponible en: <http://codigoelectronica.com/blog/esp8266-esp01-datasheet#especificaciones>. [Accedido: 18-jun-2022].
- [69] ATMEL, «ATmega328P». [En línea]. Disponible en: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf.
- [70] «Arduino Nano (V2.3) User Manual Released under the Creative Commons Attribution Share-Alike 2.5 License». [En línea]. Disponible en: <https://www.arduino.cc/en/uploads/Main/ArduinoNanoManual23.pdf>. [Accedido: 18-jun-2022].
- [71] L. Llamas, «Esquema De Patillaje (Pinout) De Arduino Uno, Nano, Mini Y Mega», 2015. [En línea]. Disponible en: <https://www.luisllamas.es/esquema-de-patillaje-de-arduino-pinout/>. [Accedido: 18-jun-2022].

- [72] «Handson Technology». [En línea]. Disponible en: <https://www.handsontec.com/dataspecs/mb102-ps.pdf>. [Accedido: 18-jun-2022].
- [73] T. I. Incorporated., «µA7800 SERIES POSITIVE-VOLTAGE REGULATORS POST OFFICE BOX 655303 • DALLAS, TEXAS 75265 D 3-Terminal Regulators D Output Current up to 1.5 A D Internal Thermal-Overload Protection D High Power-Dissipation Capability D Internal Short-Circuit Current Limitin», 1976. [En línea]. Disponible en: <https://www.sparkfun.com/datasheets/Components/LM7805.pdf>. [Accedido: 18-jun-2022].
- [74] MIDDLEWARE, «El concepto de IDE», Red Hat, Inc., 2017. [En línea]. Disponible en: <https://www.redhat.com/es/topics/middleware/what-is-ide>. [Accedido: 18-jun-2022].
- [75] Arduino, «Software | Arduino», Arduino.cc, 2021. [En línea]. Disponible en: <https://www.arduino.cc/en/software>. [Accedido: 18-jun-2022].
- [76] Visual Studio Code - Code Editing. Redefined
<https://code.visualstudio.com/>
- [77] PlatformIO, «PlatformIO is a professional collaborative platform for embedded development», PlatformIO, 2022. [En línea]. Disponible en: https://platformio.org/?utm_source=platformio&utm_medium=piohome%0Ahttps://platformio.org. [Accedido: 18-jun-2022].
- [78] L. Llamas, «Medir temperatura y humedad con Arduino y DHT11 o DHT22», Ingeniería, informática y diseño, 2016. [En línea]. Disponible en: <https://www.luisllamas.es/arduino-dht11-dht22/>. [Accedido: 18-jun-2022].
- [79] «Decibel X: dBA Sonómetro Pro - Aplicaciones en Google Play». [En línea]. Disponible en: <https://play.google.com/store/apps/details?id=com.skypaw.decibel&gl=US>. [Accedido: 18-jun-2022].
- [80] C. Arzuza, D. Chavarro, y D. Palacios, «Transductor de flujo de agua YF-S201 - YouTube». [En línea]. Disponible en: <https://www.youtube.com/watch?v=8Os665VJvwY>. [Accedido: 18-jun-2022].
- [81] «Medidor de flujo - caudalimetro - Arduino - Robotica - YouTube». [En línea]. Disponible en: https://www.youtube.com/watch?app=desktop&v=tJ7d_T-BqI8. [Accedido: 18-jun-2022].
- [82] L. Llamas, «Detector de gases con Arduino y la familia de sensores MQ», Ingeniería, informática y diseño, 2016. [En línea]. Disponible en: <https://www.luisllamas.es/arduino-detector-gas-mq/>. [Accedido: 18-jun-2022].

- [83] «SERIE SENSORES Y MODULOS #17: SENSOR GAS MQ - CALCULO DE PPM - CALIBRACION - MQ2 - YouTube». [En línea]. Disponible en: <https://www.youtube.com/watch?v=ihxuKCD-zFw>. [Accedido: 18-jun-2022].
- [84] Ankit Rohatgi, «WebPlotDigitizer - Copyright 2010-2017 Ankit Rohatgi», 2017. [En línea]. Disponible en: <http://arohatgi.info/WebPlotDigitizer/app/>? [Accedido: 18-jun-2022].
- [85] L. Llamas, «Controlar motores de corriente continua con Arduino y L298N», Luisllamas.Es, 2016. [En línea]. Disponible en: <https://www.luisllamas.es/arduino-motor-corriente-continua-l298n/>. [Accedido: 18-jun-2022].
- [86] L. DEL VALLE, «Servomotor con Arduino tutorial de programación paso a paso», 2016. [En línea]. Disponible en: <https://programarfacil.com/tutoriales/fragmentos/servomotor-con-arduino/>. [Accedido: 18-jun-2022].
- [87] L. Llamas, «Reproducir sonidos con Arduino y un buzzer pasivo o altavoz», Reproducir sonidos con Arduino y un buzzer pasivo o altavoz, 2016. [En línea]. Disponible en: <https://www.luisllamas.es/reproducir-sonidos-arduino-buzzer-pasivo-altavoz/>. [Accedido: 18-jun-2022].
- [88] L. Llamas, «Conectar Un Display Lcd Hitachi a Arduino Por Bus I2C», 2016. [En línea]. Disponible en: <https://www.luisllamas.es/arduino-lcd-i2c/>. [Accedido: 18-jun-2022].
- [89] «ESP8266-01 Firmware Install Putty SSH comandos AT». [En línea]. Disponible en: <http://kio4.com/arduino/57modulowifi.htm>. [Accedido: 18-jun-2022].
- [90] «17. Arduino, ESP8266 y ThingSpeak: envío de datos de un DHT11 a una plataforma IoT. - YouTube». [En línea]. Disponible en: <https://www.youtube.com/watch?v=C5PpMGVVD7c>. [Accedido: 18-jun-2022].
- [91] MathWorks, «IoT Analytics - ThingSpeak Internet of Things», Thingspeak, 2022. [En línea]. Disponible en: <https://thingspeak.com/>. [Accedido: 18-jun-2022].
- [92] U. A. de B. UAB y I. T. A. de C. y L. ITACyL, «MEJORA DEL BIENESTAR Y LA EFICACIA PRODUCTIVA EN GRANJAS DE OVINO LECHERO MEDIANTE EL CONTROL DE LOS EFECTOS MEDIOAMBIENTALES», 2022.
- [94] «Fritzing». [En línea]. Disponible en: <https://fritzing.org/>. [Accedido: 28-jun-2022].

[96] «Arduino y el modo Sleep | Tienda y Tutoriales Arduino». [En línea]. Disponible en: <https://www.prometec.net/el-modo-sleep-en-arduino/>. [Accedido: 30-jun-2022].

[97] Rocket Scream Electronics, «rocketscream/Low-Power: Low Power Library for Arduino». [En línea]. Disponible en: <https://github.com/rocketscream/Low-Power>. [Accedido: 30-jun-2022].

5.2 Normativa

[7] Consejo de la Unión Europea, «Directiva/2007/43/CE del Consejo de 28 de junio de 2007 por la que se establecen las disposiciones mínimas para la protección de los pollos destinados a la producción de carne», Diario Oficial de la Unión Europea, 2007. [En línea]. Disponible en: <https://www.boe.es/buscar/doc.php?id=DOUE-L-2007-81253>. [Accedido: 18-jun-2022].

[8] «BOE.es - BOE-A-2003-17811 Resolución de 11 de septiembre de 2003, de la Secretaría General de Medio Ambiente, por la que se dispone la publicación del Acuerdo de 25 de julio de 2003, del Consejo de Ministros, por el que se aprueba el Programa nacional de reducción progresiva de emisiones nacionales de dióxido de azufre (SO₂), óxidos de nitrógeno (NO_x), compuestos orgánicos volátiles (COV) y amoníaco (NH₃).» [En línea]. Disponible en: https://www.boe.es/diario_boe/txt.php?id=BOE-A-2003-17811. [Accedido: 18-jun-2022].

[16] AENOR, «UNE-EN 60529:2018 Grados de protección proporcionados por las ...», 2018. [En línea]. Disponible en: <https://www.une.org/encuentra-tu-norma/busca-tu-norma/norma?c=N0060015>. [Accedido: 18-jun-2022].

[93] «BOE.es - Reglamento electrotécnico para baja tensión e ITC». [En línea]. Disponible en: https://www.boe.es/biblioteca_juridica/codigos/codigo.php?modo=2&id=326_Reglamento_electrotecnico_para_baja_tension_e_ITC. [Accedido: 27-jun-2022].

[95] Aldeltatec, «IPC 2221 para Diseño de PCB como usarla y actualización.», Aldeltatec. [En línea]. Disponible en: <https://www.aldeatec.com/blog-diseno-con-normas-y-certificaciones/norma-ipc-2221b-para-diseno-de-pcb-actualizacion/>. [Accedido: 28-jun-2022].

5.3 Glosario de términos

-RH (Relative Humidity – Humedad Relativa): relación entre la cantidad de humedad en un lugar y la cantidad de humedad si el mismo lugar estuviera completamente saturado.

-THI (Temperature Humidity Index): medida que indica el efecto combinado de la temperatura ambiente y la humedad relativa, usado para cuantificar el estrés térmico. [3]

-Grado de Protección IP (Ingress Protection): código formado por las letras IP seguido, normalmente, de dos cifras numéricas que indican el grado de protección de la envolvente encargada de proteger algún elemento o dispositivo de influencias externas. La primera cifra (0-6) hace referencia a la protección contra la penetración de cuerpos sólidos extraños (como el polvo). La segunda cifra (0-8) determina la protección contra el agua [15].

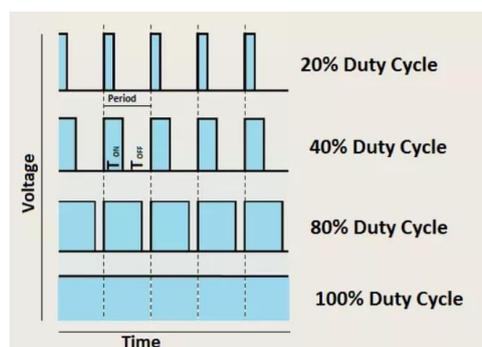
-IoT (Internet of Things): se define como la interconexión de dispositivos y objetos a través de una red, normalmente Internet, en la que pueden comunicarse e interactuar unos con otros sin necesidad de intervención humana [25].

-Movimiento DIY (Do It Yourself – Hazlo Tú Mismo) y movimiento maker: son movimientos que promueven la realización de cosas por uno mismo, en sus orígenes su función era la de reciclar o reusar materiales y ahorrar dinero. Con el tiempo y el incremento de tecnologías más accesibles para el público como las impresoras en 3D y las placas de desarrollo estos movimientos se han ido popularizando y extendiendo, llegando al ámbito educativo y de ocio. El movimiento maker se basa en los principios de crear, innovar, compartir, jugar, participar y cambiar [32] [33].

-Protocolo I2C (Inter-Integrated Circuit): protocolo de comunicación serial síncrona entre circuitos integrados. Es un sistema maestro/esclavo. Consta de dos cables, uno para transmisión/recepción de información (SDA – System Data) y otro para la señal de reloj (SCL – System Clock) [37].

-PWM (Pulse Width Modulation – Modulación de Ancho de Pulso): los pines digitales en Arduino ofrecen nivel alto a 5V y nivel bajo a 0V mediante una señal cuadrada. La modificación del ancho del pulso, es decir, cuánto tiempo está a nivel alto permite poder cambiar la luminosidad de un led, el control del ángulo de giro de un servomotor o la velocidad de un motor.

Figura 5.1: Regulación del ancho de pulso (Fuente: [56]).



-IDE (Integrated Development Enviroment – Entorno de Desarrollo Integrado): es un software para el diseño de aplicaciones que combina herramientas para desarrolladores con una interfaz de usuario gráfica (GUI). Generalmente cuenta con un editor de código capaz de resaltar la sintaxis, autocompletado y comprobación de errores; automatización de compilaciones locales para automatizar tareas sencillas y repetitivas, transcripción a código binario para la máquina y ejecución de pruebas; y depurador para probar programas y mostrar errores [73].

-HMI Human-Machine Interface o Interfaz Humano-Máquina: es el sistema intermediario entre el operario y el ordenador, su función es la de mostrar la información, normalmente en tiempo real de los parámetros de un proceso [62].

-Normas IPC: *“son una herramienta basada en la experiencia y en la industria, guía de las mejores prácticas de diseño, fabricación, ensamble, inspección de circuitos impresos o PCB (Printed Circuit Board), cables, conectores, insumos de soldadura, pruebas y muchos elementos más. Al aplicarlas ayudan a alcanzar altos niveles de calidad. También ayudan a optimizar el proceso completo de manufactura, reducir el tiempo de lanzamiento al mercado, y obtener un producto con mejor desempeño.”* [95].

6 ANEXOS

6.1 Anexo I: Hojas de datos

6.1.1 Hojas de datos de sensores

Tabla 6.1: Sensor de temperatura y humedad (Fuente: [48]).

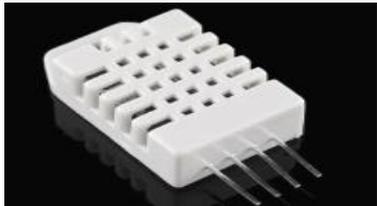
HOJA DE DATOS DE SENSOR			
Parámetros generales			
Uso	Medir temperatura y humedad		
Modelo	DHT22 / AM2302		
Tipo de sensor	Condensador de humedad de polímero		
Señal de salida	Digital		
Características			
	Mínimo	Máximo	Unidades
Tensión de alimentación	3,3	5,5	Voltios (V) - DC
Consumo en reposo	40	50	Microamperios (uA)
Consumo nominal	1	1,5	Miliamperios (mA)
Tiempo de lectura	2	2	Segundos (s)
Rango operativo	0	100	Porcentaje humedad (%)
	-40	80	Grados Celsius (°C)
Resolución	0,1	0,1	-
Precisión	-1	+1	Porcentaje (%)
	-0,2	+0,2	Grados Celsius (°C)
Dimensiones (sin placa)			
Largo	Alto	Ancho	Unidades
33,6	15,1	7	Milímetros (mm)

Tabla 6.2: Sensor de amoniaco (Fuente: [49]).

HOJA DE DATOS DE SENSOR				
Parámetros generales				
Uso	Medir nivel amoniaco			
Modelo	MQ-137			
Tipo de sensor	Semiconductor			
Señal de salida	Analógica y digital			
Características				
	Mínimo	Máximo	Unidades	
Tensión de alimentación	5	5	Voltios (V) - DC / AC	
Tensión calentador	5	5	Voltios (V) - DC / AC	
Consumo nominal		800	Milivatios (mW)	
Tiempo de calentamiento	24		Horas (h)	
Rango operativo		95	Porcentaje humedad (%)	
	-10	45	Grados Celsius (°C)	
Detección de concentración	5	200	Partes por millón (ppm) - NH3	
Dimensiones (sin placa)				
Largo	Alto	Ancho	Unidades	
19	23	19	Milímetros (mm)	

Tabla 6.3: Sensor calidad aire (Fuente: [50]).

HOJA DE DATOS DE SENSOR				
Parámetros generales				
Uso	Medir la calidad del aire			
Modelo	MQ-135			
Tipo de sensor	Semiconductor			
Señal de salida	Analógica y digital			
Características				
	Mínimo	Máximo	Unidades	
Tensión de alimentación	5	5	Voltios (V) - DC / AC	
Tensión calentador	5	5	Voltios (V) - DC / AC	
Consumo nominal		800	Milivatios (mW)	
Tiempo de calentamiento	24		Horas (h)	
Rango operativo		95	Porcentaje humedad (%)	
	-10	45	Grados Celsius (°C)	
Detección de concentración	10	300	Partes por millón (ppm) - NH3	
	10	1000	Partes por millón (ppm) - Benceno	
	10	300	Partes por millón (ppm) - Alcohol	
Dimensiones (sin placa)				
Largo	Alto	Ancho	Unidades	
19	23	19	Milímetros (mm)	

Tabla 6.4: Sensor de Ruido ambiente (Fuente: [51]).

HOJA DE DATOS DE SENSOR				
Parámetros generales				
Uso	Medir nivel de ruido			
Modelo	MAX9814			
Tipo de sensor	Micrófono			
Señal de salida	Analógica			
Características				
	Mínimo	Máximo	Unidades	
Tensión de alimentación	2,7	5,5	Voltios (V) - DC	
Consumo nominal		6	Miliamperios (mA)	
Ganancia	40 - 50 - 60		Decibelios (dB)	
Tiempo de lectura	2	2	Segundos (s)	
Rango operativo	-40	85	Grados Celsius (°C)	
Salida	20	20	Kilohercios (kHz)	
	430	430	Microvoltios (μV) - AC	
Dimensiones (con placa)				
Largo	Alto	Ancho	Unidades	
25	14	16	Milímetros (mm)	

Tabla 6.5: Sensor caudal (Fuente: [52]).

HOJA DE DATOS DE SENSOR				
Parámetros generales				
Uso	Medir caudal de agua			
Modelo	YF-S201			
Tipo de sensor	Efecto Hall			
Señal de salida	Digital			
Características				
	Mínimo	Máximo	Unidades	
Tensión de alimentación	4,5	24	Voltios (V) - DC	
Consumo nominal a 5V	15	15	Miliamperios (mA)	
Tasa de flujo	1	30	Litros por minuto	
Temperatura de operación		80	Grados Celsius (°C)	
Temperatura del líquido		120	Grados Celsius (°C)	
Presión		1,75	Megapascuales (Mpa)	
Conexión rosca	1/2	1/2	Pulgadas (")	
Dimensiones				
Largo	Alto	Ancho	Unidades	
62	35	34,8	Milímetros (mm)	

6.1.2 Hojas de datos de actuadores

Tabla 6.6: Driver motores (Fuente: [55]).

HOJA DE DATOS DE ACTUADOR				
Parámetros generales				
Uso	Accionar y regular motores DC			
Modelo	L298N			
Señal de control	Digital (PWM velocidad)			
Características				
	Mínimo	Máximo	Unidades	
Tensión de alimentación con jumper	5	12	Voltios (V) - DC	
Tensión de alimentación sin jumper	12	35	Voltios (V) - DC	
Tensión salida con jumper activo	5	5	Voltios (V) - DC	
Consumo nominal	0	36	Miliamperios (mA)	
Corriente máxima al motor		2	Amperios (A)	
Potencia a 75°C		20	Vatios (W)	
Temperatura almacenamiento	-25	130	Grados Celsius (°C)	
Dimensiones				
Largo	Alto	Ancho	Unidades	
43	43	29	Milímetros (mm)	

Tabla 6.7: Ventilador (Fuente: [58]).

HOJA DE DATOS DE ACTUADOR				
Parámetros generales				
Uso	Ventilación			
Modelo	AFB02512LAYBF			
Señal de control	Controlado por driver L298N			
Tipo de rodamiento	Hidráulico			
Características				
	Mínimo	Máximo	Unidades	
Tensión de alimentación	4,5	13,2	Voltios (V) - DC	
Consumo de corriente	0,03	0,05	Amperios (A)	
Potencia		0,6	Vatios (W)	
Caudal	0,67	0,67	Pies cúbicos por minuto (CFM)	
	0,024	0,019	Metros cúbicos por minuto (CFM)	
Velocidad	0	5000	Revoluciones por minuto (rpm)	
Presión	0,045	0,032	Pulgada de columna de agua	
Ruido	18	20	Decibelios (dB)	
Dimensiones				
Largo	Alto	Ancho	Unidades	
25	25	10	Milímetros (mm)	

Tabla 6.8: Servomotor (Fuente: [59]).

HOJA DE DATOS DE ACTUADOR			
Parámetros generales			
Uso	Apertura ventanas		
Modelo	SG-90		
Señal de control	Digital (PWM)		
Características			
	Mínimo	Máximo	Unidades
Tensión de alimentación	4,8	6	Voltios (V) - DC
Corriente en reposo		10	Miliamperios (mA)
Corriente en movimiento	100	250	Miliamperios (mA)
Corriente en posición		360	Miliamperios (mA)
Ángulo de rotación	0	180	Grados (°)
Par		1,7	Kilos por centímetro (kg*cm)
Velocidad		0,12/60	Segundos / Grados (s/°)
		83.3	Revoluciones minuto (rpm)
Dimensiones			
Largo	Alto	Ancho	Unidades
32	13	31	Milímetros (mm)

Tabla 6.9: Buzzer (Fuente: [60]).

HOJA DE DATOS DE ACTUADOR			
Parámetros generales			
Uso	Sonido de alarma		
Modelo	MH-FMD		
Señal de control	Digital		
Características			
	Mínimo	Máximo	Unidades
Tensión de alimentación	4	8	Voltios (V) - DC
Consumo de corriente	0	30	Miliamperios (mA)
Potencia sonora		≥85	Decibelios (dB)
Frecuencia de resonancia	2000	2600	Hercios (Hz)
Rango operativo	-25	80	Grados Celsius (°C)
Dimensiones (con placa)			
Largo	Alto	Ancho	Unidades
38	13	10	Milímetros (mm)

6.1.3 Hojas de datos de HMI

Tabla 6.10: Pantalla LCD y controlador I2C (Fuentes: [63] [64]).

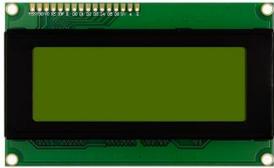
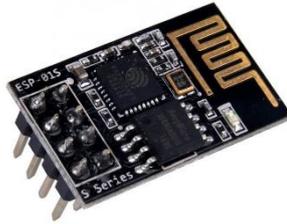
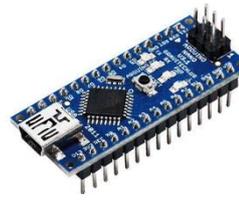
HOJA DE DATOS DE HMI			
Parámetros generales LCD			
Uso	Visualización de datos		
Modelo	LCD 2004A		
Controlador	HD44780		
Señal de control	Proveniente del adaptador I2C		
Número de caracteres	20 caracteres x 4 líneas		
Características			
	Mínimo	Máximo	Unidades
Tensión de alimentación	5	5	Voltios (V) - DC
Consumo de corriente	0	200	Miliamperios (mA)
Rango operativo	-10	60	Grados Celsius (°C)
		90	Porcentaje humedad (%)
Dimensiones			
Largo	Alto	Ancho	Unidades
98	60	13,4	Milímetros (mm)
Parámetros generales controlador I2C			
Uso	Comunicación entre el microcontrolador y el display		
Modelo	FC-113		
Señal de control	Analógica (SCL Y SDA)		
Características			
	Mínimo	Máximo	Unidades
Tensión de alimentación	5	5	Voltios (V) - DC
Dimensiones			
Largo	Alto	Ancho	Unidades
53	19	15	Milímetros (mm)

Tabla 6.11: ESP8266 (Fuente: [68]).

HOJA DE DATOS DE HMI				
Parámetros generales				
Uso	Envío de los datos a la página web			
Modelo	ESP8266 - ESP01			
Señal de control	Digital RX0 TX1			
Protocolos WiFi admitidos	802,11 b/g/n			
Seguridad	WPA/WPA2			
Cifrado	WEP/TKIP/AES			
Características				
	Mínimo	Máximo	Unidades	
Tensión de alimentación	2,5	3,6	Voltios (V) - DC	
Consumo de corriente	0	80	Miliamperios (mA)	
Frecuencia	2,4		Gigahercios (GHz)	
Rango operativo	-40	125	Grados Celsius (°C)	
Velocidad de bits		72,2	Megabits por segundo (Mbps)	
Dimensiones				
Largo	Alto	Ancho	Unidades	
98	60	13,4	Milímetros (mm)	

6.1.4 Hojas de datos de procesador

Tabla 6.12: Arduino nano (Fuente: [70]).

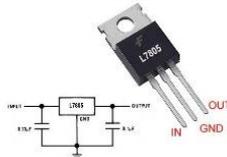
HOJA DE DATOS DE PROCESADOR				
Parámetros generales				
Modelo	Arduino Nano - ATmega328P			
Pines analógicos	6			
Pines digitales	14 (6 salida PWM)			
Memoria flash	32 kilobytes			
Características				
	Mínimo	Máximo	Unidades	
Tensión alimentación	7	12	Voltios (V) - DC	
Tensión operación	5	5	Voltios (V) - DC	
Tensión de salida	3,3 y 5		Voltios (V) - DC	
Consumo de corriente	0	19	Miliamperios (mA)	
Corriente de salida I/O	0	40	Miliamperios (mA)	
Corriente de salida	0	200	Miliamperios (mA)	
Frecuencia reloj		16	Megahercios (MHz)	
Rango operativo	-40	85	Grados Celsius (°C)	
Dimensiones				
Largo	Alto	Ancho	Unidades	
43	43	29	Milímetros (mm)	

6.1.5 Hojas de datos de alimentación

Tabla 6.13: Conversor voltaje para protoboard (Fuente: [71]).

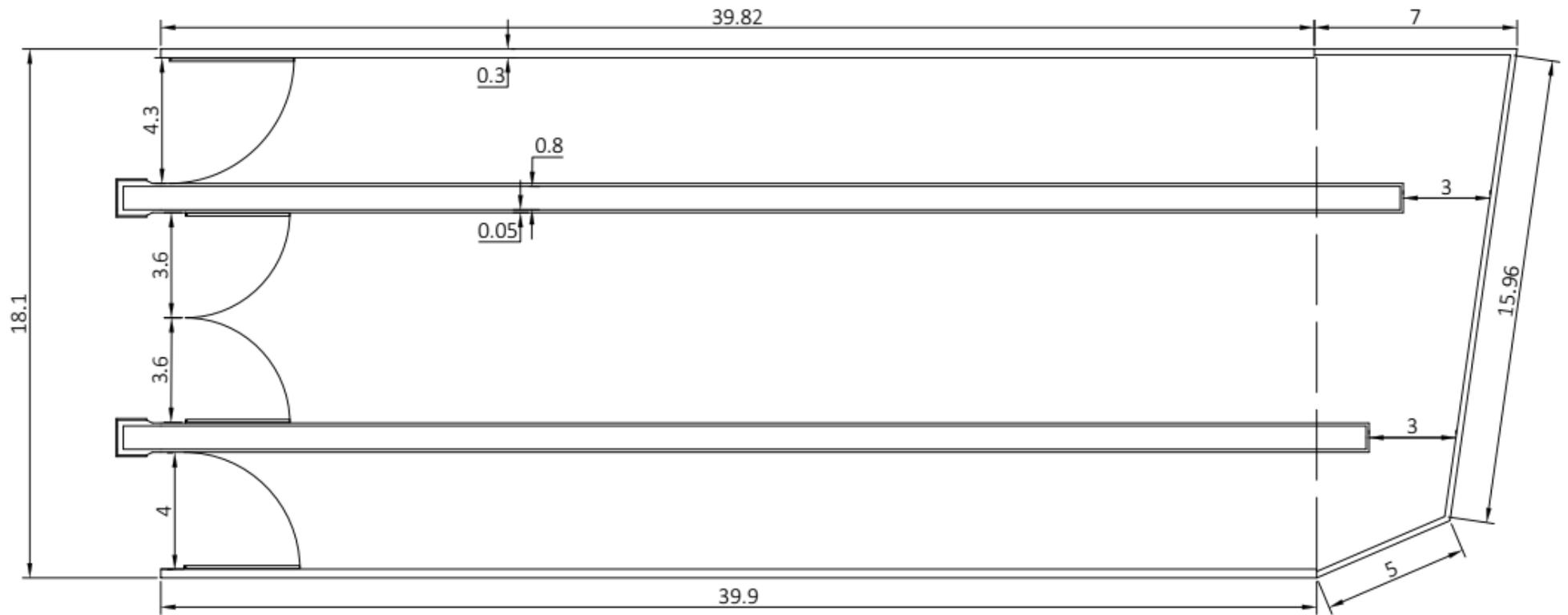
HOJA DE DATOS DE ALIMENTACIÓN				
Parámetros generales				
Uso	Convertir el voltaje de la fuente			
Modelo	mb102			
Conectores de entrada	USB tipo A y DC barrel jack			
Características				
	Mínimo	Máximo	Unidades	
Tensión de entrada	6,5	12	Voltios (V) - DC	
Tensión de salida	3,3 y 5		Voltios (V) - DC	
Corriente salida	0	700	Miliamperios (mA)	
Dimensiones				
	Largo	Alto	Ancho	Unidades
	53	32	24	Milímetros (mm)

Tabla 6.14: Regulador de tensión de 5V (Fuente: [72]).

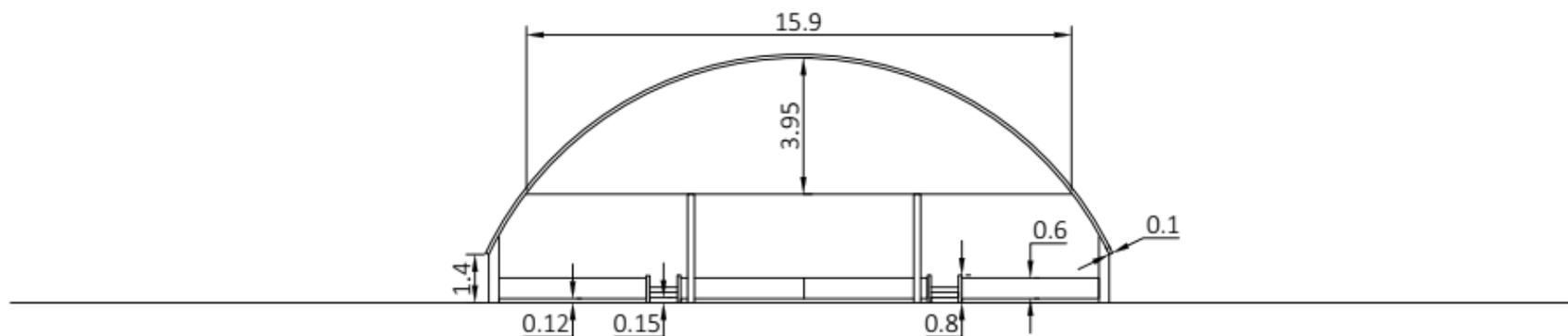
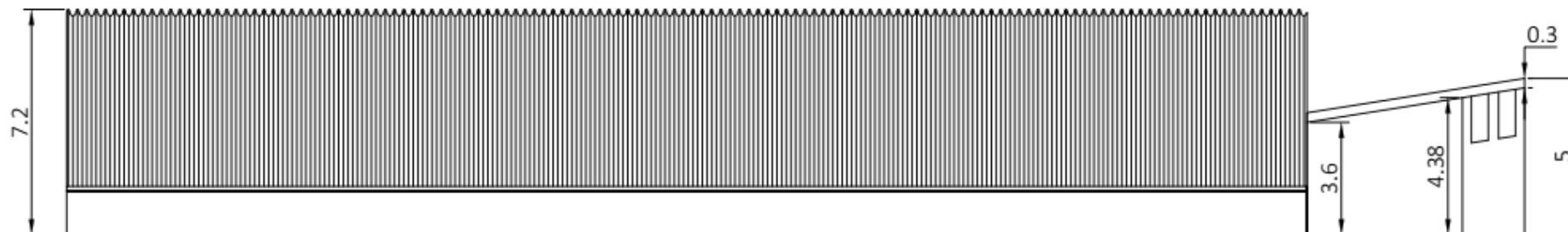
HOJA DE DATOS DE ALIMENTACIÓN				
Parámetros generales				
Uso	Convertir el voltaje de la fuente			
Modelo	LM7805			
Tipo de encapsulado	TO220			
Características				
	Mínimo	Máximo	Unidades	
Tensión de entrada	7	35	Voltios (V) - DC	
Tensión de salida	4,8	5,2	Voltios (V) - DC	
Corriente salida	0	1,5	Amperios (A)	
Dimensiones				
	Largo	Alto	Ancho	Unidades
	30	10	4	Milímetros (mm)

6.2 Anexo II: Planos

6.2.1 Planos de la instalación

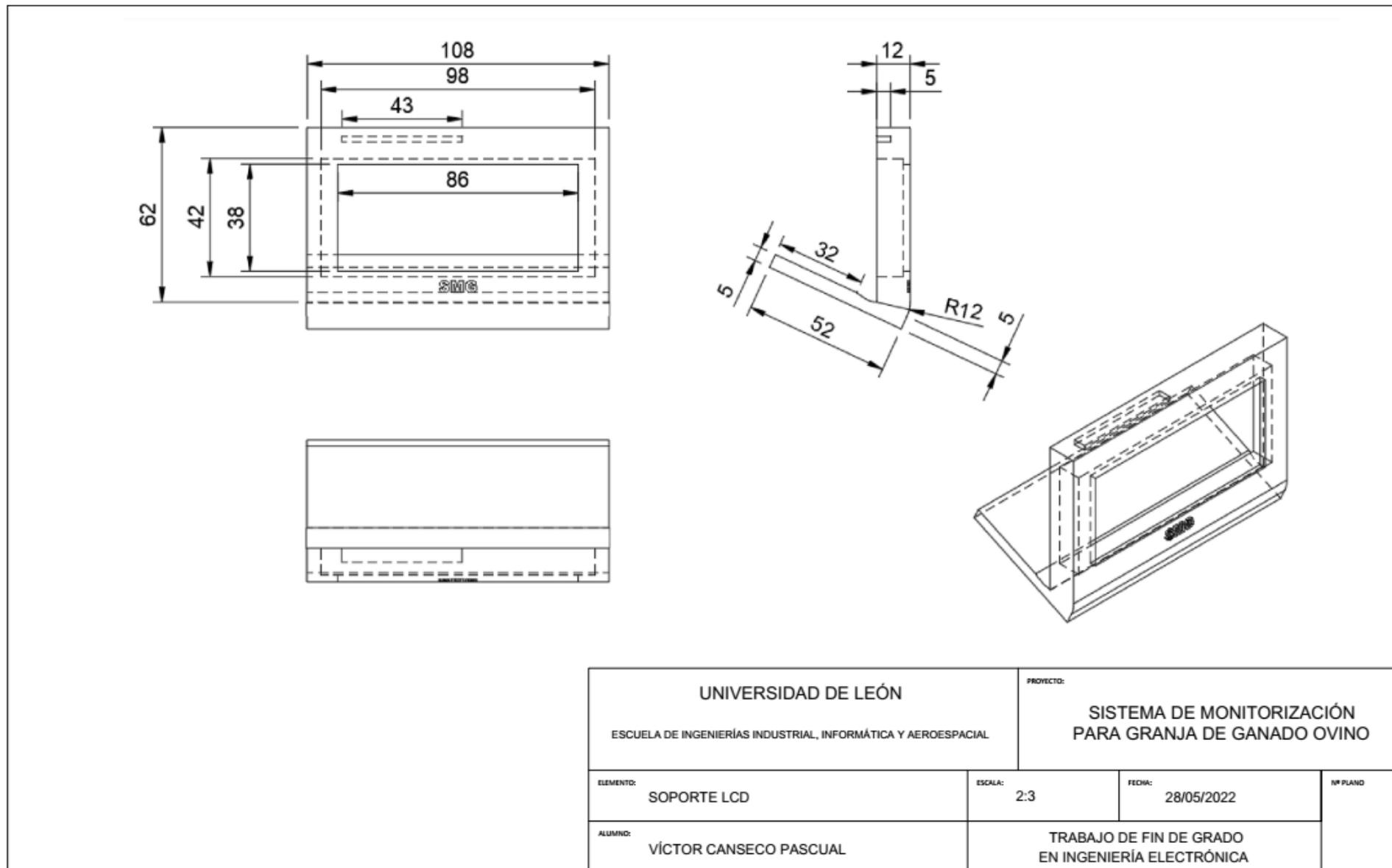


<p>UNIVERSIDAD DE LEÓN</p> <p>ESCUELA DE INGENIERÍAS INDUSTRIAL, INFORMÁTICA Y AEROESPACIAL</p>		<p>PROYECTO:</p> <p>SISTEMA DE MONITORIZACIÓN PARA GRANJA DE GANADO OVINO</p>	
<p>ELEMENTO:</p> <p>PLANO DE LA NAVE</p>	<p>ESCALA:</p> <p>1:200</p>	<p>FECHA:</p> <p>28/05/2022</p>	<p>Nº PLANO</p> <p>1</p>
<p>ALUMNO:</p> <p>VÍCTOR CANSECO PASCUAL</p>	<p>TRABAJO DE FIN DE GRADO EN INGENIERÍA ELECTRÓNICA</p>		



UNIVERSIDAD DE LEÓN ESCUELA DE INGENIERÍAS INDUSTRIAL, INFORMÁTICA Y AEROSPAZIAL		PROYECTO: SISTEMA DE MONITORIZACIÓN PARA GRANJA DE GANADO OVINO	
ELEMENTO: PLANO DE LA NAVE	ESCALA: 1:200	FECHA: 28/05/2022	Nº PLANO: <div style="font-size: 2em; font-weight: bold; text-align: center;">2</div>
ALUMNO: VÍCTOR CANSECO PASCUAL	TRABAJO DE FIN DE GRADO EN INGENIERÍA ELECTRÓNICA		

6.2.2 Planos piezas 3D



6.3 Anexo III: Flujogramas

Figura 6.1: Flujograma de medición de temperatura.

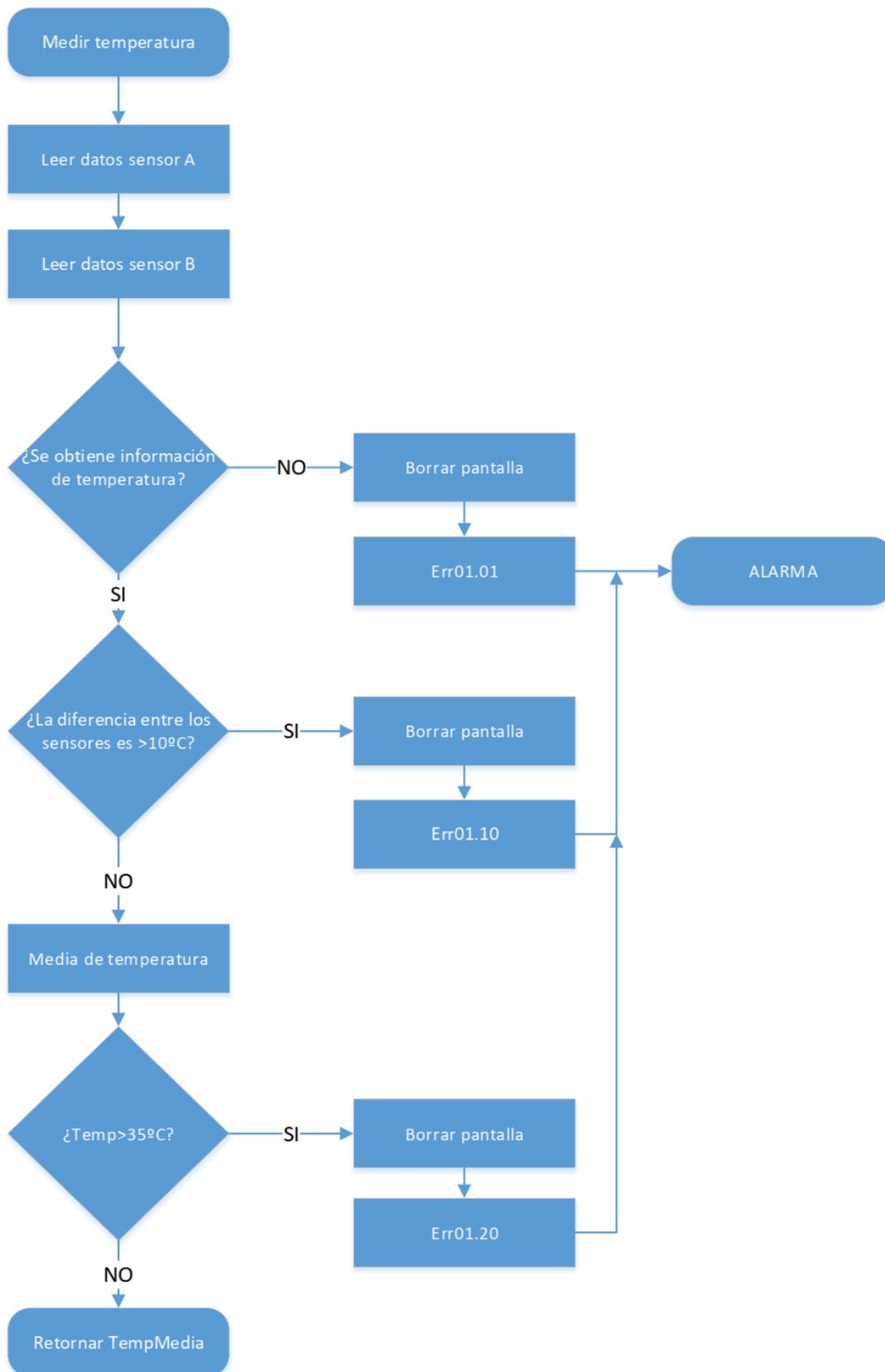


Figura 6.2: Flujo de medición de humedad.

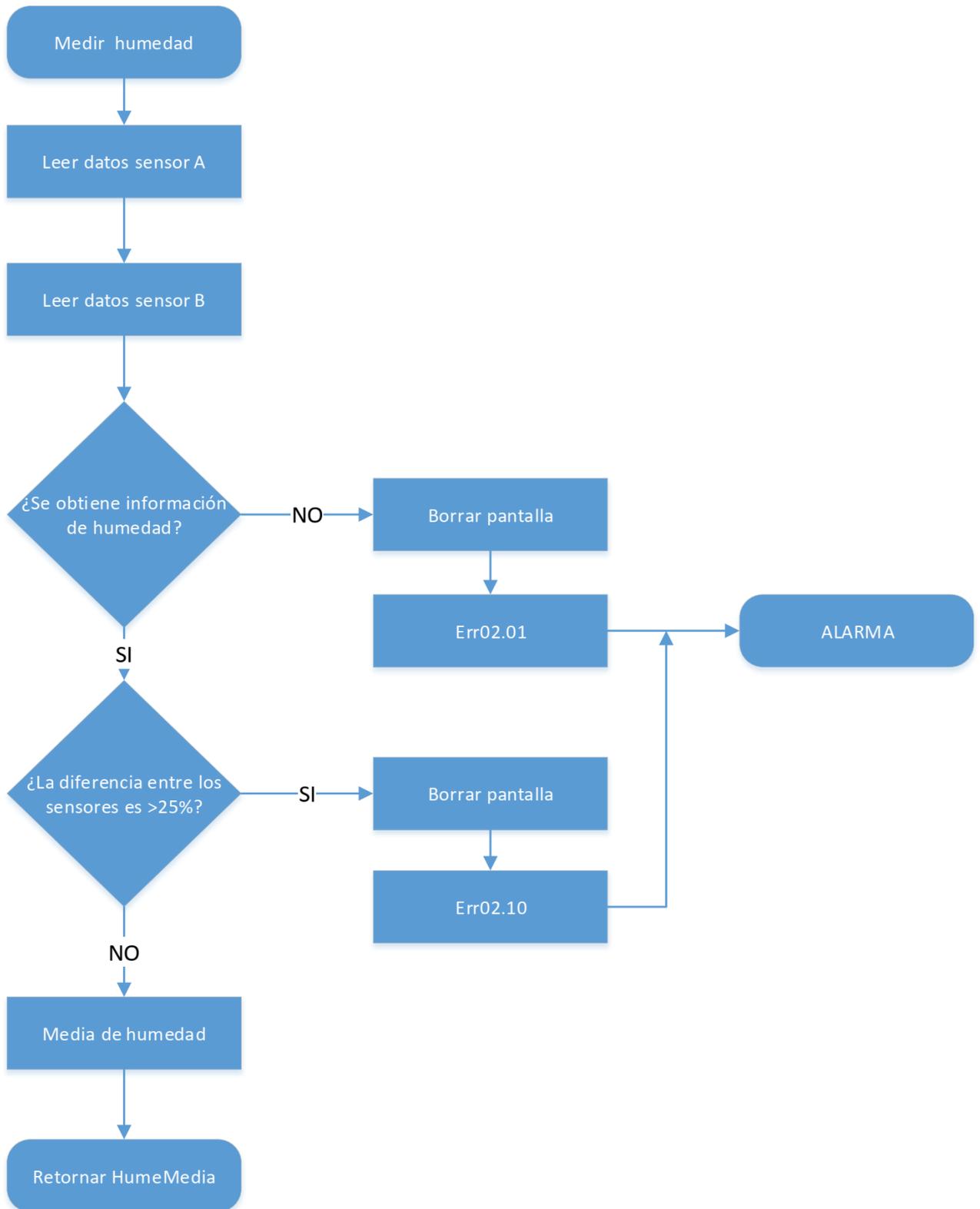


Figura 6.3: Flujo de medición de amoniaco.

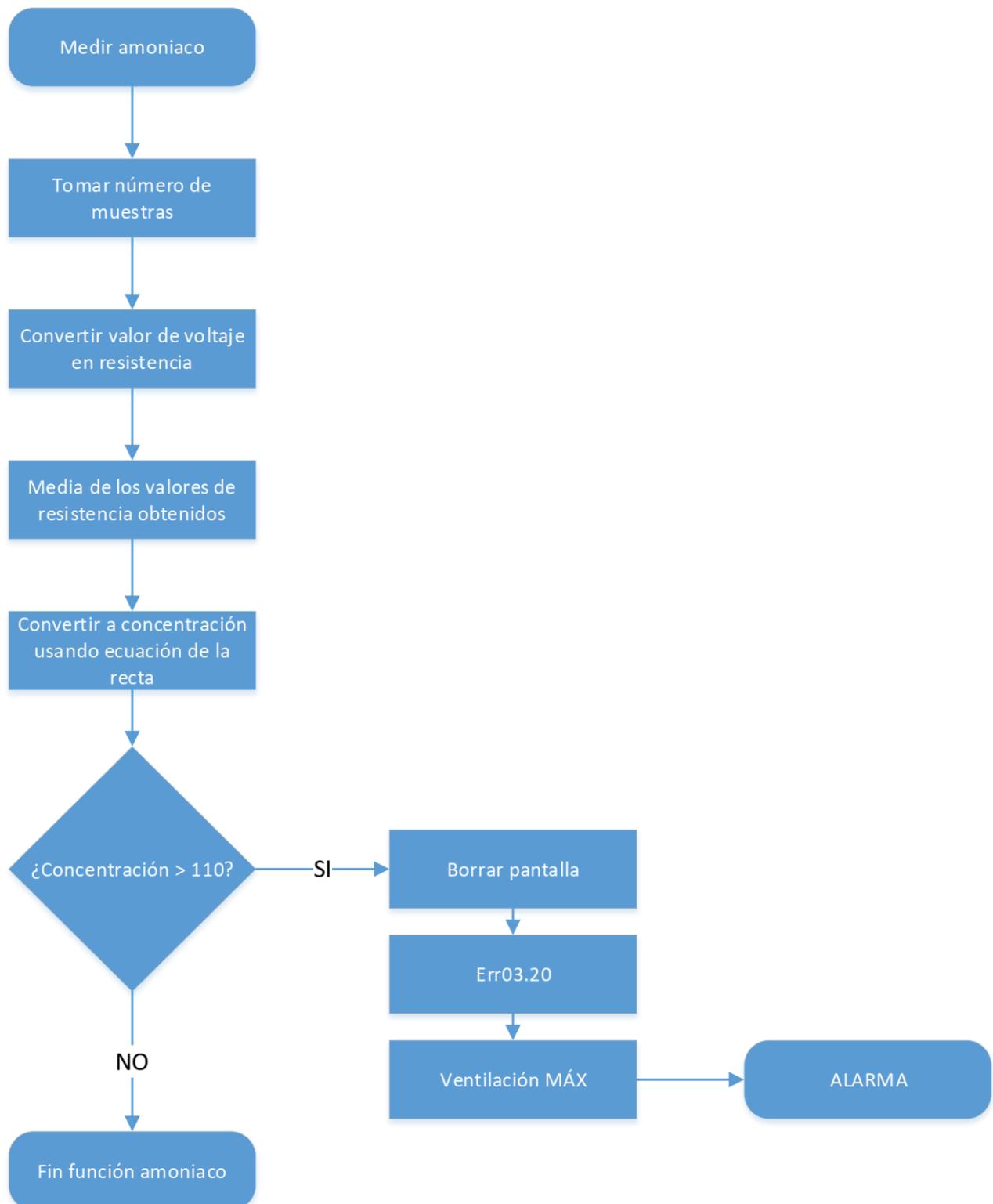


Figura 6.4: Flujo de medición de dióxido de carbono.

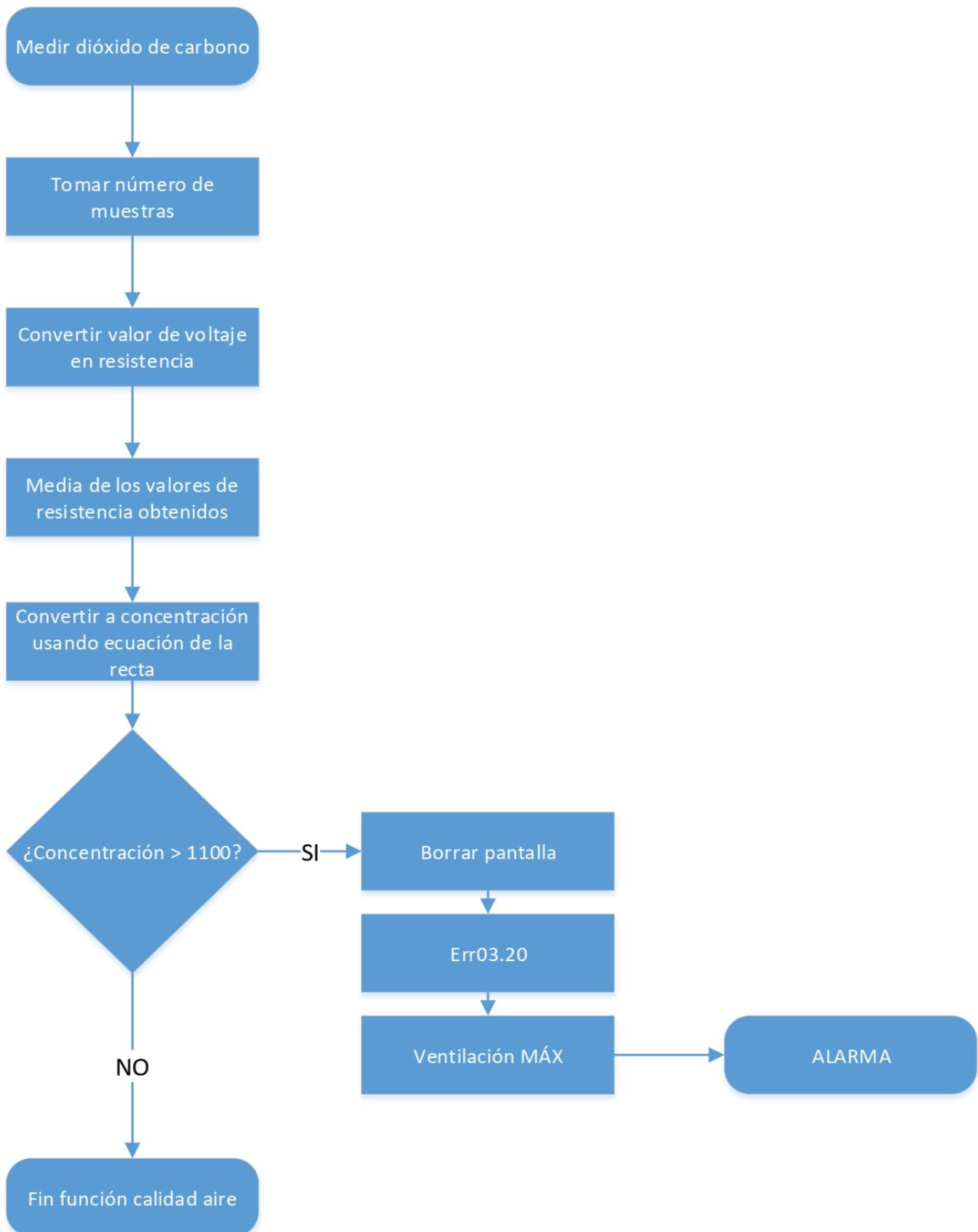


Figura 6.5: Flujoograma función aireación.

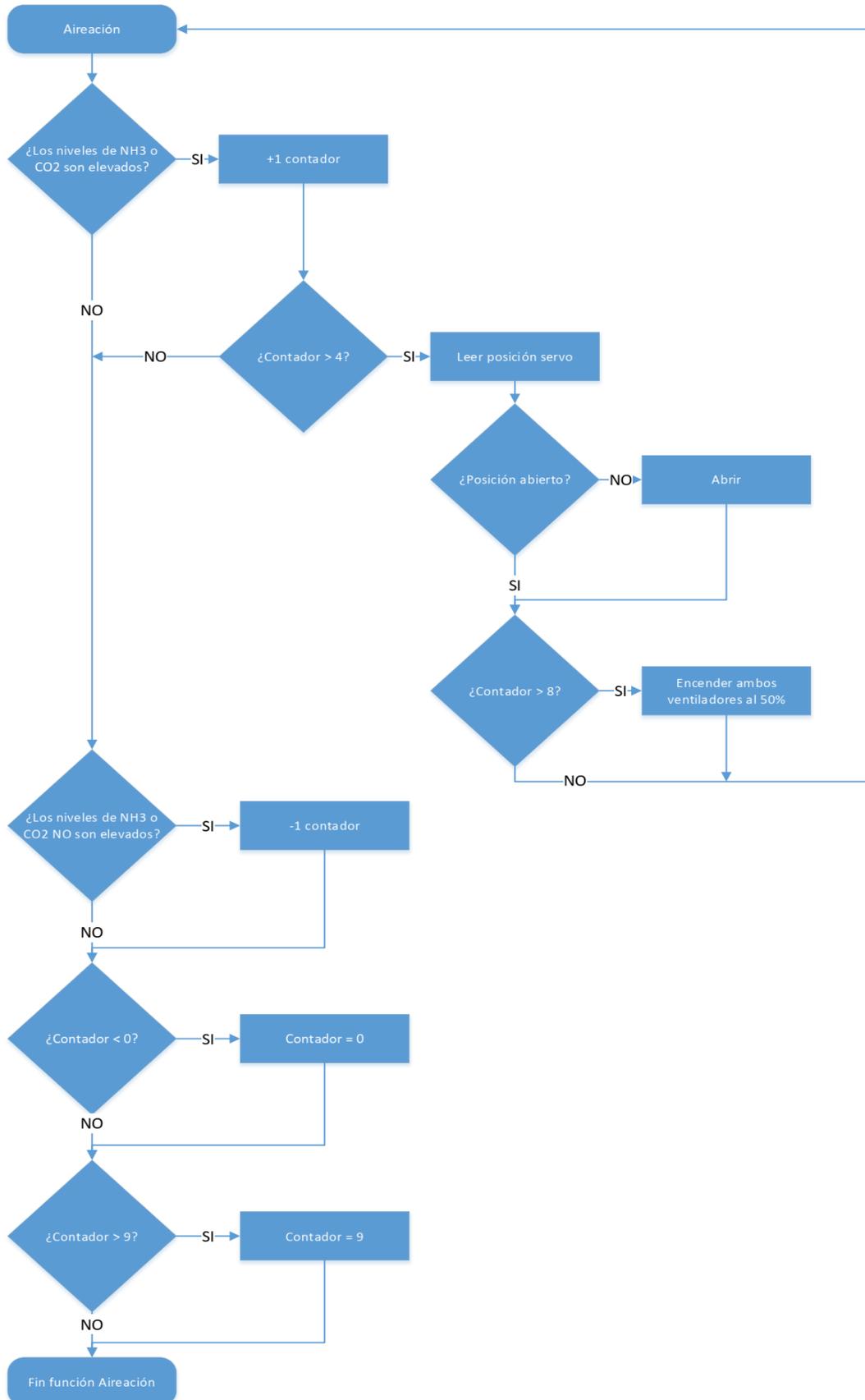


Figura 6.6: Flujograma función ventilación. Parte 1.

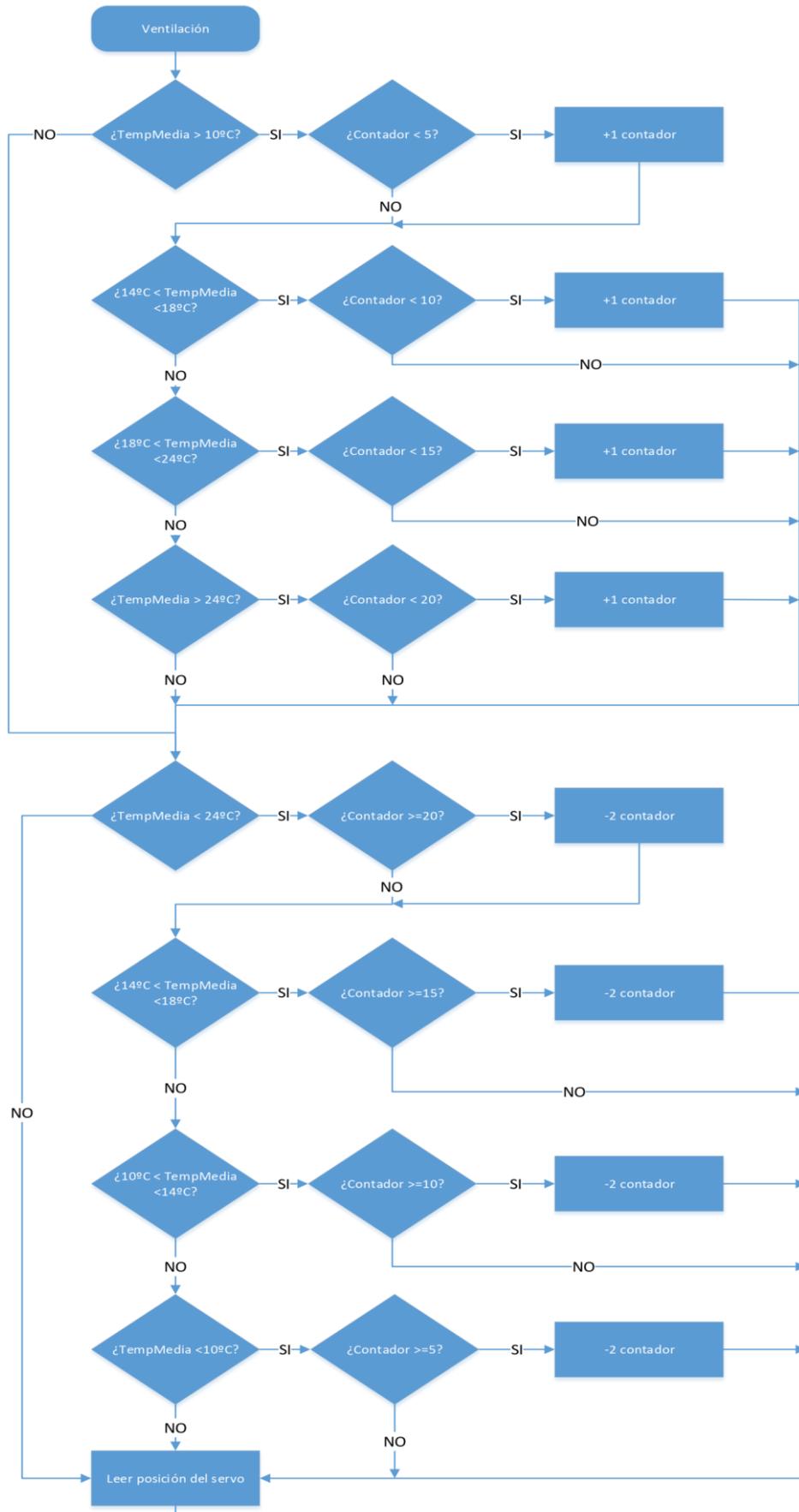


Figura 6.7: Flujograma función ventilación. Parte 2.

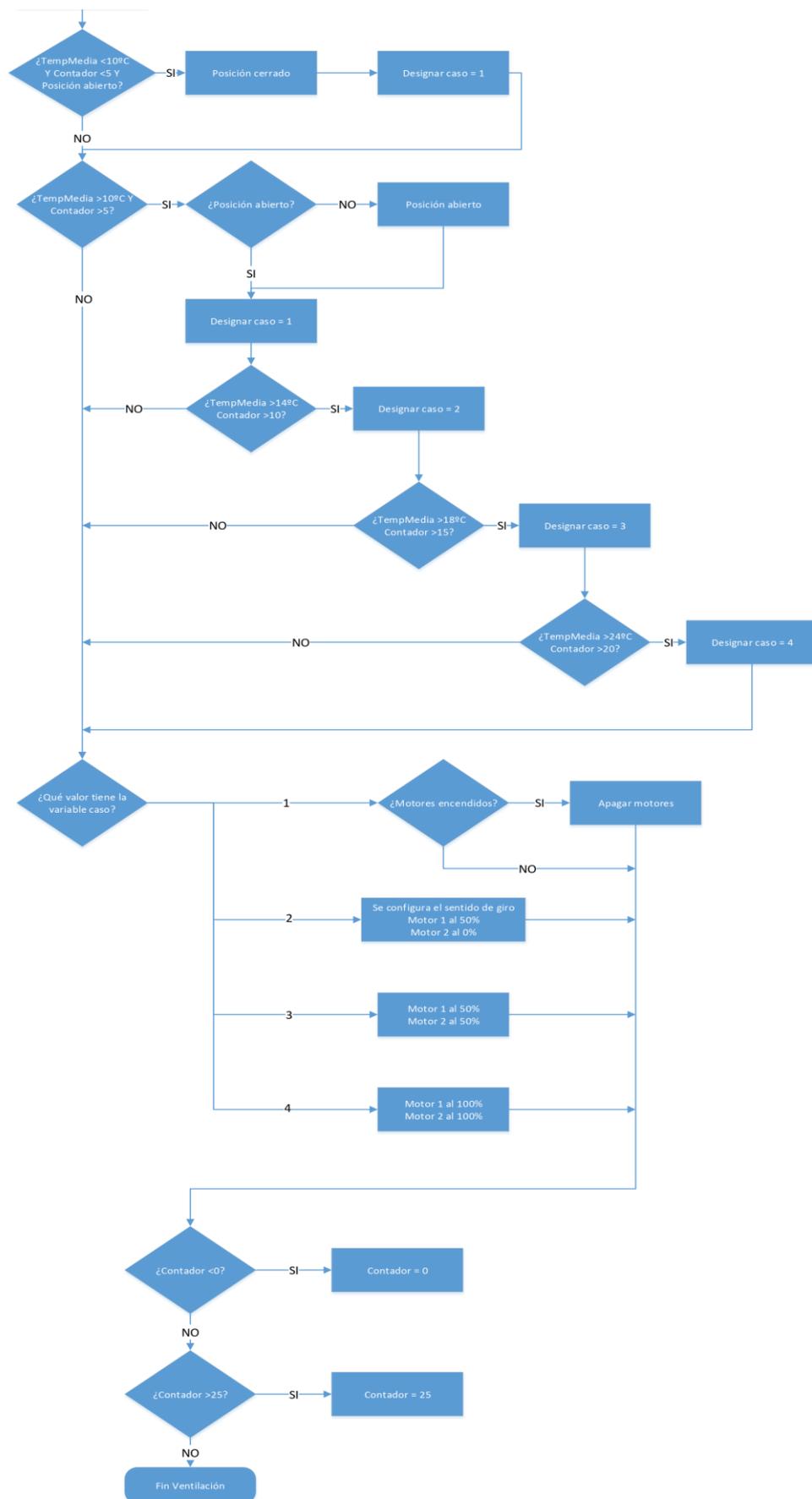


Figura 6.8: Flujoograma de medición de caudal y corte del suministro.

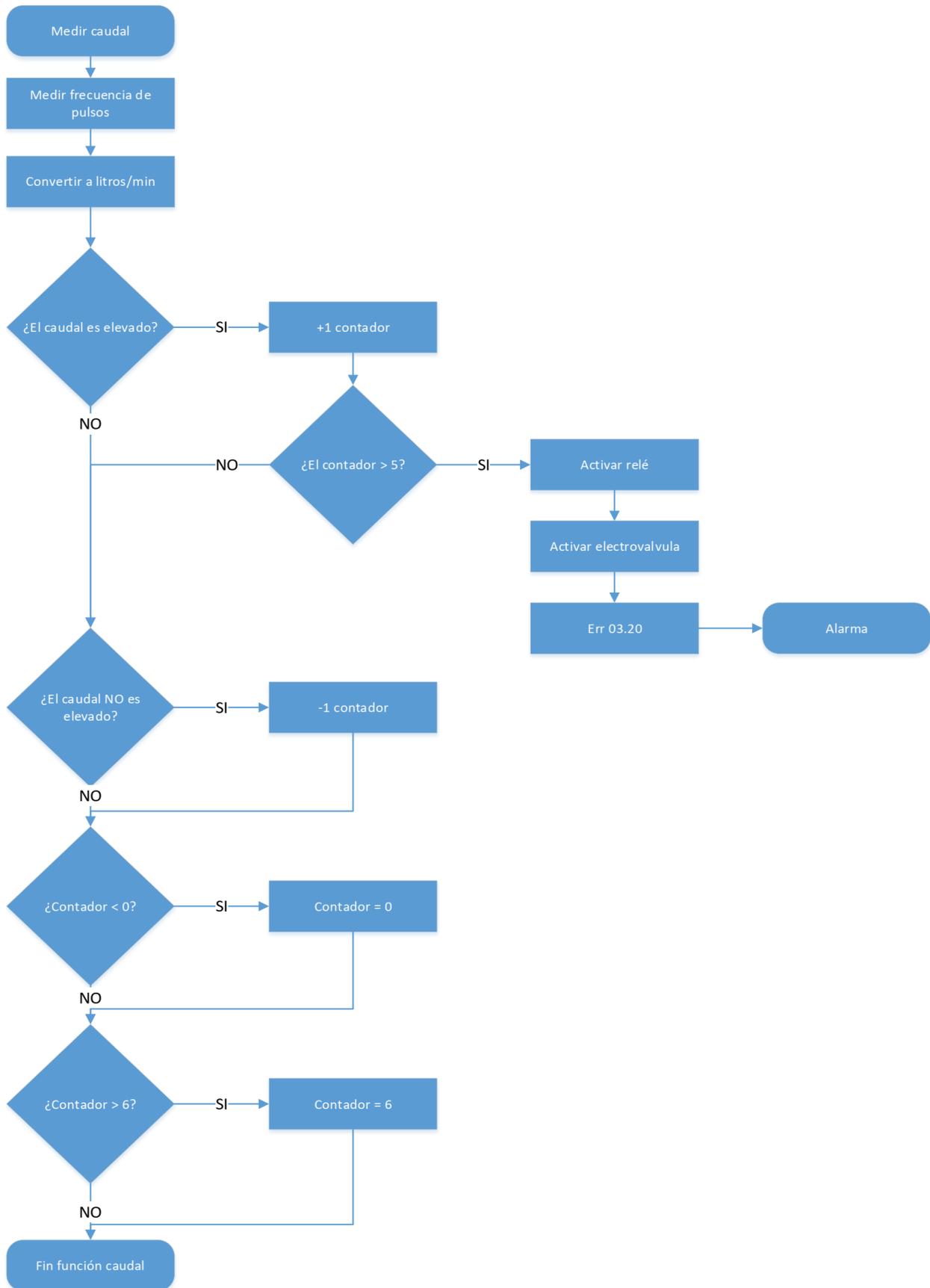


Figura 6.9: Flujograma de medición de ruido ambiente.

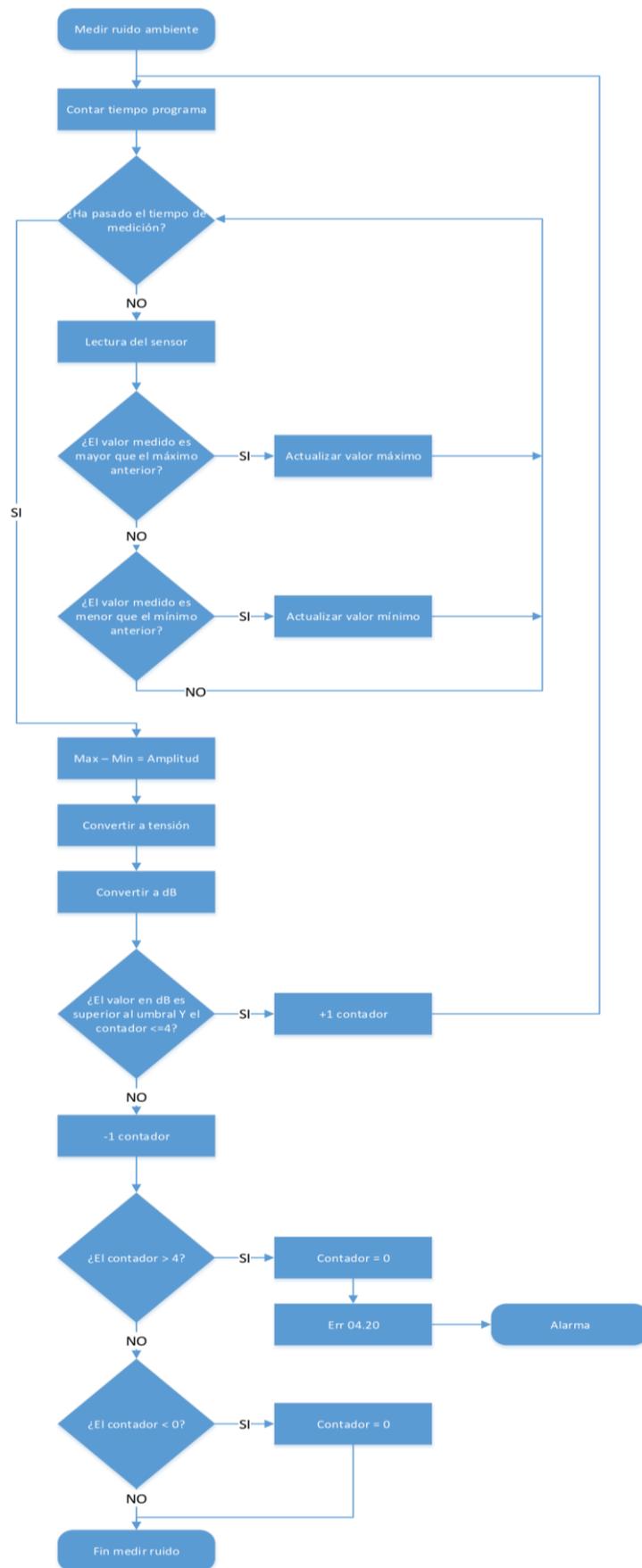


Figura 6.10: Flujograma función imprimir.

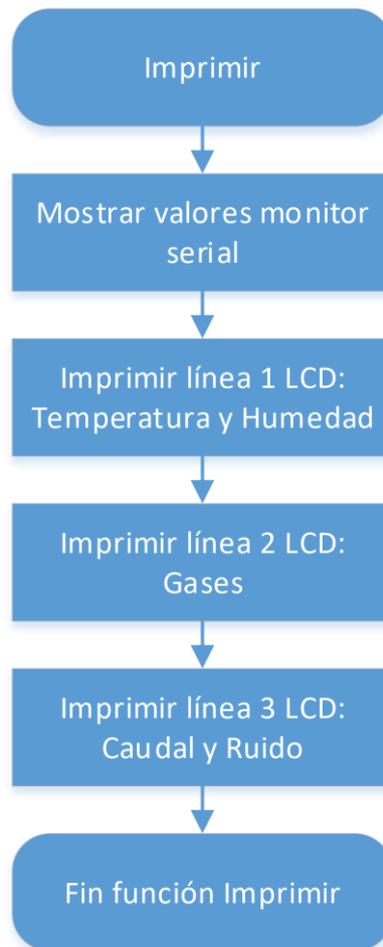
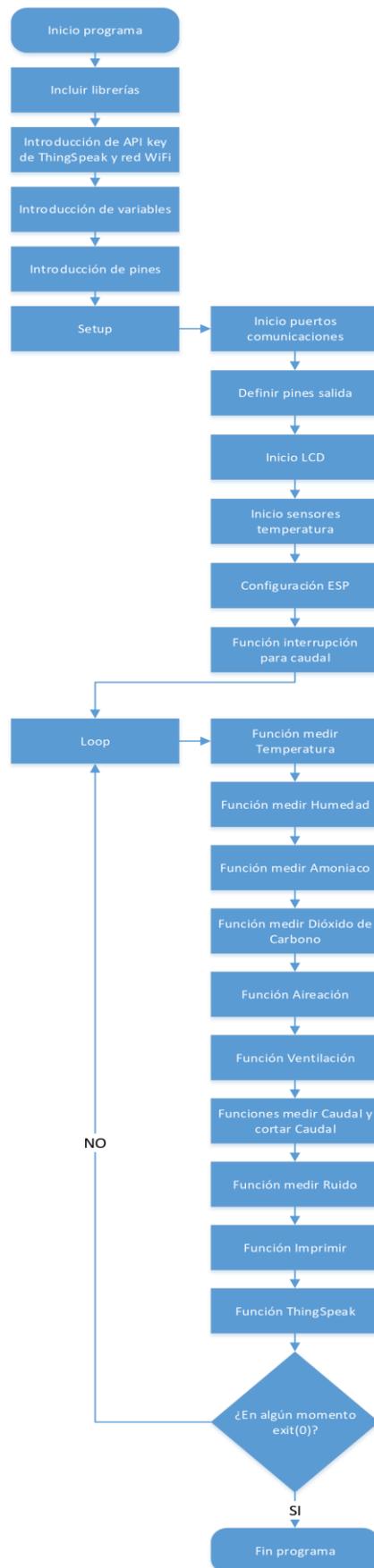


Figura 6.11: Flujograma función ThingSpeak (mandar datos a la web).



Figura 6.12: Flujo de funcionamiento general.



6.4 Anexo IV: Programación

6.4.1 Código principal

```

#include <Arduino.h>
#include <DHT.h> //librería sensores de temperatura y humedad
#include <math.h>
#include <LiquidCrystal_I2C.h> //librería pantalla
#include <Wire.h> //uso de los pines SCL y SDA del controlador I2C de la pantalla
#include <Servo.h> //librería para controlar el servomotor
#include <SoftwareSerial.h> //librería para usar otros pines que no sean el 0 y el 1 para comunicación serial
SoftwareSerial espSerial = SoftwareSerial(2,3); // arduino RX pin=2 arduino TX pin=3 por tanto el pin TX del ESP01 va al pin RX del arduino

Servo servoMot; //se hace llamar a nuestro servo servoMot
LiquidCrystal_I2C lcd(0x27, 20, 4); // Inicia el LCD en la dirección 0x27, con 20 caracteres y 4 líneas

String apiKey = "VE3F7J6QKDYCT69D"; //API KEY WRITE de ThingSpeak, número personal obtenido de la web
String ssid = "NOMBREred"; // Nombre de la red
String password = "CLAVEred"; // Clave de la red

#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321
//VARIABLES -----
//temperatura y humedad
float tempMedia = 0;
int caso = 0; //se usa para la activación de la ventilación
int cuentaTemp = 0; //este contador se usará para ver si la temperatura supera un umbral
float humeMedia = 0;

//servomotor
int pos = 0; //variable para leer la posición del servo

//caudalímetro
const int measureInterval = 2500; //intervalo de medida del caudal
volatile int pulseConter; //contador de los pulsos de la señal de caudalímetro
const float factorK = 7.22; //factor de corrección del sensor calculado experimentalmente (consultar calibraciones y pruebas)
float litros = 0;

```

```

int cuentaCaudal = 0; //contador para controlar si un caudal
elevado puede suponer una rotura en el sistema de agua

//ruido
const int sampleWindow = 50; //variable para el sensorRuido.
Ancho ventana en mS (50 mS = 20Hz)
double dB = 0;
int cuentadB = 0;

//alarma
int cuentaAlarma = 0; //variable para ver las veces que se
dispara la alarma

//gases
const int RL_VALUE = 1; // Resistencia RL del modulo en Kilo
ohms
const float R0 = 3.58; // Resistencia R0 del sensor en Kilo
ohms
const int READ_SAMPLE_INTERVAL = 100; // Tiempo entre
muestras
const int READ_SAMPLE_TIMES = 5; // Numero muestras

int partesNH3 = 0; //concentración de amoniacó en ppm
int partesCO2 = 0; //concentración de dióxido de carbono en ppm

int cuentaAire = 0;

//PINES -----
const int sensorCaudal = 12; //pin del caudalímetro
const int DHT1Pin = 10; //se define el pin al que está
conectado el primer sensor DHT
const int DHT2Pin = 11; //y lo mismo para el segundo
const int pinENA = 5; //pin activar motor 1
const int pinENB = 6; //pin activar motor 2
const int pinIN1 = 7; //IN1=IN3 pin dirección del motor
const int pinIN2 = 8; //IN2=IN4 dirección del motor junto con
IN1
const int rele = 4;
const int buzzer = A0; //se asigna el pin analógico 0 al buzzer
const int MQ135 = A1; // Pin del sensor MQ-135
const int MQ137 = A2; // Pin del sensor MQ-137
const int LED = A3;
const int sensorRuido = A6;

```

```

DHT dht1(DHT1Pin, DHTTYPE); //se conforma el pin con el tipo
de sensor
DHT dht2(DHT2Pin, DHTTYPE);

//Funcion de interrupción para contar los pulsos del
caudalímetro
void ISRCountPulse()
{
    pulseConter++;
}

//Funciones necesarias para el ESP01 para poder enviar los datos
a ThingSpeak
void tiempoRespuesta(int waitTime){ //función usada para que el
ESP01 tenga tiempo de configurarse
    unsigned long t=millis();
    char c;
    while ((t+waitTime)>millis()){
        if (espSerial.available()){
            c=espSerial.read();
            Serial.print(c);
        }
    }
}

boolean thingSpeakWrite(float value1, float value2, float
value3, float value4, float value5, float value6){
    String cmd = "AT+CIPSTART=\"TCP\",\"; //
Conexión TCP
    cmd += "184.106.153.149"; //
Dirección de api.thingspeak.com
    cmd += "\",80";
    espSerial.println(cmd); //Mandar datos seriales al ESP8266
    Serial.println(cmd);
    if(espSerial.find("Error")){
        Serial.println("AT+CIPSTART error");
        return false;
    }
    String getStr = "GET /update?api_key="; // Prepara el String
GET a mandar
    getStr += apiKey;
    getStr += "&field1=";

```

```

    getStr += String(value1); //convertir de flotante a
String    valor de temperatura
    getStr += "&field2=";
    getStr += String(value2); //valor de humedad
    getStr += "&field3=";
    getStr += String(value3); //valor de NH3
    getStr += "&field4=";
    getStr += String(value4); //valor de CO2
    getStr += "&field5=";
    getStr += String(value5); //valor de caudal
    getStr += "&field6=";
    getStr += String(value6); //valor de ruido
    // ...
    getStr += "\r\n\r\n";
    // enviar la longitud de la cadena
    cmd = "AT+CIPSEND=";
    cmd += String(getStr.length());
    espSerial.println(cmd);
    Serial.println(cmd);
    delay(100);
    if(espSerial.find(">")){//si Ok envíe la cadena de GET
        espSerial.print(getStr);
        Serial.print(getStr);
    }
    else{
        espSerial.println("AT+CIPCLOSE");
        // Que el usuario sepa que no se están enviando datos
        Serial.println("AT+CIPCLOSE");
        return false;
    }

    return true;
}
//Acaba el envío de datos a ThingSpeak

```

```

void setup() {
  Serial.begin(9600); //se inicia el monitor serial 9600
  baudios
  espSerial.begin(9600); // Habilita la comunicación serial
  con el ESP01
  Serial.println("Iniciando");
  //pinMode
  pinMode(buzzer,OUTPUT); //se define el pin A8 del buzzer como
  salida
  pinMode(LED,OUTPUT);
  pinMode(pinIN1, OUTPUT);
  pinMode(pinIN2, OUTPUT);
  pinMode(pinENA, OUTPUT);
  pinMode(pinENB, OUTPUT);

  servoMot.attach(9); //asignamos el pin 9 al servo
  servoMot.write(0); //colocamos el servo en la pos 0
  //Inicio LDC
  lcd.init (); //se inicia la pantalla LCD
  lcd.backlight(); //se inicia la luz de la pantalla
  lcd.setCursor(0, 1); //se coloca el cursor de la LCD en la
  línea 1, posición 0
  lcd.print("INICIANDO SISTEMA");
  delay(1000);
  lcd.setCursor(0, 2);
  lcd.print("Bienvenido");
  delay(1000);
  lcd.clear();

  //Inicio DHT
  dht1.begin(); //se inician los sensores de temperatura y
  humedad
  dht2.begin();

  //Configuración del ESP01 mediante comandos AT
  espSerial.println("AT+CWMODE=1"); // Configure el ESP8266
  como cliente
  tiempoRespuesta(1000);
  espSerial.println("AT+CWJAP=\""+ssid+"\", \""+password+"\"");
  // Configure el SSID y password para conexión WIFI
  tiempoRespuesta(5000);
  Serial.println("Setup completed");
}

```

```

    attachInterrupt(digitalPinToInterrupt(sensorCaudal),
    ISRCountPulse, RISING); //Función de interrupción para el
    caudalímetro con el flanco de subida
}

void alarma(){
    Serial.println("Ha ocurrido un error en el sistema");
    lcd.setCursor(0, 1); //se coloca el cursor de la LCD en la
    línea 1, posición 0
    lcd.print("AVISO Ha ocurrido un");
    lcd.setCursor(0, 2);
    lcd.print("error en el sistema!");
    for (int i = 0; i <5; i++) {
        tone(buzzer, 1000); //se hace sonar el buzzer con
    frecuencia 1000
        digitalWrite(LED,HIGH);
        delay(300);
        digitalWrite(LED,LOW);
        delay(300);
        digitalWrite(LED,HIGH);
        delay(400);
        tone(buzzer, 2000); //se hace sonar el buzzer con
    frecuencia 2000
        digitalWrite(LED,LOW);
        delay(300);
        digitalWrite(LED,HIGH);
        delay(300);
        digitalWrite(LED,LOW);
        delay(400);
        noTone(buzzer); //se apaga el buzzer
    }

    delay(5000);
    lcd.clear();
    if(cuentaAlarma>3){ //si se ha disparado la alarma varias
    veces se entra en bucle de alarma, sino se vuelve a comprobar el
    sistema
        alarma(); //vuelve a empezar la función
    }
    cuentaAlarma++;
//exit(0); //esto termina el programa
}

```

```

float leeTemp() { //función que se encarga de medir las dos temp
y hacer la media
    delay(2000);
    float tempS1 = dht1.readTemperature(); //se lee la
temperatura en los dos sensores
    float tempS2 = dht2.readTemperature();
    delay(250);
    if (isnan(tempS1) || isnan(tempS2)) { //se comprueba si los
sensores están bien
        Serial.println("No se puede obtener información de
temperatura");
        lcd.clear(); //borramos pantalla
        lcd.setCursor(0, 4);
        lcd.print("Err01.01"); //escribimos código error en
pantalla (consultar error en documento adjunto)
        delay(500);
        alarma();
    }
    float tempERROR = abs(tempS1 - tempS2); //se calcula la
diferencia entre las dos temperaturas
    if (tempERROR > 10){ //si la diferencia de temperaturas
es grande puede haber un error en los sensores y se avisa al
usuario
        Serial.println("Diferencia de temperaturas elevada");
        lcd.clear(); //borramos pantalla
        lcd.setCursor(0, 4);
        lcd.print("Err01.10"); //escribimos código error en
pantalla (consultar error en documento adjunto)
        delay(500);
        alarma();
    }
    tempMedia = (tempS1+tempS2)/2; //se hace una media entre
ambos
    if (tempMedia > 35){
        Serial.println("Temperatura muy elevada");
        lcd.clear(); //borramos pantalla
        lcd.setCursor(0, 4);
        lcd.print("Err01.20"); //escribimos código error en
pantalla (consultar error en documento adjunto)
        delay(500);
        ventMAX(); //se manda a función de ventilación máxima y
luego, desde allí, a alarma
    }
}

```

```
}
return tempMedia;
}

float leeHume(){ //se lee la humedad de los sensores
    delay(2000);
    float humeS1 = dht1.readHumidity();
    float humeS2 = dht2.readHumidity();
    delay(250);
    if (isnan(humeS1) || isnan(humeS2)) { //se comprueba si los
sensores están bien
        Serial.println("No se puede obtener información de
humedad");
        lcd.clear();
        lcd.setCursor(0, 4);
        lcd.print("Err02.01"); //escribimos código error en
pantalla (consultar error en documento adjunto)
        delay(500);
        alarma();
    }
    float humeERROR= abs(humeS1 - humeS2);
    if (humeERROR > 25){
        Serial.println("Diferencia de humedad elevada");
        lcd.clear();
        lcd.setCursor(4, 0);
        lcd.print("Err02.10"); //escribimos código error en
pantalla (consultar error en documento adjunto)
        delay(500);
        alarma();
    }
    humeMedia = (humeS1+humeS2)/2;
    return humeMedia;
}
```

```

float getMQResistance(int raw_adc) //función que convierte el
valor recibido de voltaje de los sensores de gases en el valor
de resistencia
{
    return (((float)RL_VALUE / 1000.0*(1023 - raw_adc) /
raw_adc));
}

void amoniaco(){ //función para determinar el nivel de amoniaco
en el ambiente mediante el sensor MQ-137
    const float A137 = -0.2733679;
    const float B137 = 0.4354262;
    float rs1 = 0;
    for (int i = 0;i<READ_SAMPLE_TIMES;i++) {
        rs1 += getMQResistance(analogRead(MQ137));
        delay(READ_SAMPLE_INTERVAL);
    }
    rs1 = rs1 / READ_SAMPLE_TIMES; //se hace la media de muestras
tomadas
    partesNH3 = pow(10, B137 + A137 * log(rs1/R0)); //se calcula
la concentración en ppm

    if (partesNH3 > 110){
        Serial.println("Nivel de amoniaco elevado");
        lcd.clear();
        lcd.setCursor(0, 4);
        lcd.print("Err03.20"); //escribimos código error en
pantalla (consultar error en documento adjunto)
        delay(250);
        ventMAX();
    }
}

void ventMAX(){ //ventiladores a máxima potencia y redirección a
alarma
    digitalWrite(pinIN1, HIGH); //se configura el sentido de giro
de ambos motores y se los hace funcionar al máximo
    digitalWrite(pinIN2, LOW);
    analogWrite(pinENA, 255);
    analogWrite(pinENB, 255);

    alarma();
}

```

```

void calidadAire(){ //función para determinar el nivel de CO2 en
el ambiente mediante el sensor MQ-135
  const float A135 = -0.334262527; //Introducimos los valores de
A y B para calcular la recta de calibración de ambos sensores
  const float B135 = 0.692602609;
  float rs2 = 0;
  for (int i = 0;i<READ_SAMPLE_TIMES;i++) {
    rs2 += getMQResistance(analogRead(MQ135));
    delay(READ_SAMPLE_INTERVAL);
  }
  rs2 = rs2 / READ_SAMPLE_TIMES; //se hace la media de muestras
tomadas
  partesCO2 = pow(10, B135 + A135 * log(rs2/R0)); //se calcula
la concentración en ppm

  if (partesCO2 > 1100){
    Serial.println("Nivel de CO2 elevado");
    lcd.clear();
    lcd.setCursor(0, 4);
    lcd.print("Err04.20"); //escribimos código error en
pantalla (consultar error en documento adjunto)
    delay(250);
    ventMAX();
  }
}

void ventilacion (){
  if(tempMedia>10){ //cuando la temp está por encima de 10
grados se entra a las condiciones
    if(cuentaTemp<5){ //si el contador tiene un valor inferior
a 5 se incrementa en 1
      cuentaTemp++;
    }
    else if(tempMedia>14 && tempMedia<18){ //cuando la
temperatura ya supera los 14 grados pero no los 18
      if(cuentaTemp<10){ //y si el contador es inferior a 10
se incrementa en 1
        cuentaTemp++;
      }
    }
    else if(tempMedia>18 && tempMedia<24){ //se repite el
apartado anterior pero para valores superiores

```

```
        if(cuentaTemp<15){
            cuentaTemp++;
        }
    }
    else if(tempMedia>24){ //para el último tramo de
temperatura
        if(cuentaTemp<20){
            cuentaTemp++;
        }
    }
}
if(tempMedia<24){ //si la temperatura baja
    if(cuentaTemp>=20){ //y el contador está en un valor alto
se resta
        cuentaTemp = cuentaTemp-2; //restando 2 hacemos que
al bajar no quede en el límite y pueda haber menos oscilaciones
    }
    else if(tempMedia>14 && tempMedia<18){ //a medida que la
temperatura baja el contador baja también
        if(cuentaTemp>=15){
            cuentaTemp = cuentaTemp-2;
        }
    }
    else if(tempMedia>10 && tempMedia<14){
        if(cuentaTemp>=10){
            cuentaTemp = cuentaTemp-2;
        }
    }
    else if(tempMedia<10){
        if(cuentaTemp>=5){
            cuentaTemp = cuentaTemp-2;
        }
    }
}
pos = servoMot.read(); //se lee la posición del servo
if(tempMedia<10 && cuentaTemp<5 && pos!=0){ //en caso de
temperatura baja y ventana abierta se cierra
    servoMot.write(0);
    caso=1; //se manda apagar motores
}
```

```

    if(tempMedia>10 && cuentaTemp>4){ //si la temperatura ha sido
superior al limite más de 5 veces abrimos ventanas
        Serial.println(" Abrir ventanas ");
        pos = servoMot.read();
        if(pos!=90){ //si el servo no está abierto, se abre. Las
ventanas permanecerán abiertas en los siguientes casos
            servoMot.write(90);
        }
        caso=1;
        if(tempMedia>14 && cuentaTemp>9){ //según vaya
aumentando la temperatura y el contador vamos accionando más
ventilación
            Serial.println(" Motor 1 al 50 ");
            caso=2;
            if(tempMedia>18 && cuentaTemp>14){
                Serial.println(" Motor 2 al 50 ");
                caso=3;
                if(tempMedia>24 && cuentaTemp>19){
                    Serial.println(" Motores 1 y 2 al 100 ");
                    caso=4;
                }
            }
        }
    }
    switch (caso){
        case 1: //abrir/cerrar ventanas con anterioridad y/o
apagar motores
            if(pinENA!=0 || pinENB!=0){ //si los motores están
encendidos se apagan
                analogWrite(pinENA, 0);
                analogWrite(pinENB, 0);
                digitalWrite(pinIN1, LOW); //se pone el pin en bajo
para reducir el consumo
            }
            break;
        case 2: //motor 1 al 50%
            digitalWrite(pinIN1, HIGH); //se configura el sentido
de giro de ambos motores
            digitalWrite(pinIN2, LOW);
            if(pinENA!=130 || pinENB!=0){
                analogWrite(pinENA, 130); //se acciona el motor en A
al 50% (255/2≈130)

```

```

        analogWrite(pinENB, 0); //se mantiene apagado el
motor B al bajar en la escala
    }
    break;
    case 3: //motor 2 al 50%
        if(pinENA!=130 || pinENB!=130){
            analogWrite(pinENA, 130); //se mantiene la velocidad
en el motor A, así se reduce también la velocidad si viene del
siguiente if
            analogWrite(pinENB, 130);
        }
        break;
    case 4: //motor 1 y 2 al 100%
        if(pinENA!=255 || pinENB!=255){
            analogWrite(pinENA, 255);
            analogWrite(pinENB, 255);
        }
        break;
    }

    if(cuentaTemp<0){ //con esto se asegura que el contador no
baje de 0
        cuentaTemp=0;
    }
    if(cuentaTemp>25){ //con esto se asegura que el contador no
suba de 25
        cuentaTemp=25;
    }
} //acaba función ventilación

void aireacion() { //función para abrir las ventanas o activar
ventiladores sin contar la temperatura

    if(partesNH3>20 || partesCO2 >100){
        cuentaAire++;
        if (cuentaAire>4){
            pos = servoMot.read(); //se lee la posición del servo
            if(pos!=90){ //si el servo no está abierto, se abre.
                servoMot.write(90);
            }
        }
        if (cuentaAire>8){ //si abriendo las ventanas no se regula
se activa ventilación forzada al 50%

```

```

        digitalWrite(pinIN1, HIGH); //se configura el sentido de
giro de ambos motores
        digitalWrite(pinIN2, LOW);
        if(pinENA!=130 || pinENB!=130){
            analogWrite(pinENA, 130);
            analogWrite(pinENB, 130);
        }
    }
    aireacion(); //se comienza de nuevo la función hasta que se
hayan regulado los niveles de gases
}
}
if (partesNH3<20 && partesCO2<100){
    cuentaAire--;
}

if(cuentaAire<0){ //con esto se asegura que el contador no
baje de 0
    cuentaAire=0;
}
if(cuentaAire>9){ //con esto se asegura que el contador no
suba de 9
    cuentaAire=9;
}
}

float GetFrequency() //se calcula la frecuencia de los pulsos
del sensor de efecto Hall
{
    pulseConter = 0;

    interrupts();
    delay(measureInterval);
    noInterrupts();

    return (float)pulseConter * 1000 / measureInterval;
}

```

```

void corteCaudal(){ //función que si detecta que el caudal es
elevado corta el suministro
  if(litros>30){
    cuentaCaudal++;
    if(cuentaCaudal>5){
      digitalWrite(rele,HIGH); //se activa el rele que a su vez
cierra la electrovalvula
      Serial.println("Caudal elevado");
      lcd.clear();
      lcd.setCursor(0, 4);
      lcd.print("Err05.20"); //escribimos código error en
pantalla (consultar error en documento adjunto)
      delay(500);
      alarma();
    }
  }
  else if(litros<30){
    cuentaCaudal--;
  }
  if(cuentaCaudal<0){
    cuentaCaudal = 0;
  }
  if(cuentaCaudal<6){
    cuentaCaudal = 6;
  }
}

void ruido(){
  unsigned long startMillis= millis();

  unsigned int signalMax = 0;
  unsigned int signalMin = 1024;

  // Recopilar durante la ventana
  unsigned int sample;
  while (millis() - startMillis < sampleWindow)
  {
    sample = analogRead(sensorRuido);
    if (sample < 1024)
    {
      if (sample > signalMax)
      {
        signalMax = sample; // Actualizar máximo

```

```

    }
    else if (sample < signalMin)
    {
        signalMin = sample; // Actualizar mínimo
    }
}
}
unsigned int peakToPeak = signalMax - signalMin; // Amplitud
del sonido
double volts = (peakToPeak * 5.0) / 1024; // Convertir a
tensión
//Serial.println(volts);
//se convierten los voltios a decibelios mediante el
polinomio resultante de la calibración del sensor (ver
apartado)
dB = 6.9613*pow(volts,6)-
52.513*pow(volts,5)+145.29*pow(volts,4)-
182.22*pow(volts,3)+113*pow(volts,2)-23.975*pow(volts,1)+39.24;
//Serial.println(dB);
delay(250);

//se crea una condicional para si el valor supera un umbral
de ruido
if (cuentadB>4){
//si el umbral ha sido superado varias veces se notifica al
usuario el ruido elevado
    Serial.println("Ruido elevado");
    lcd.clear();
    lcd.setCursor(0, 4);
    lcd.print("Err06.20"); //escribimos código error en
pantalla (consultar error en documento adjunto)
    delay(500);

    cuentadB = 0;
    alarma();
}
if(dB>89 && cuentadB<=4){
    cuentadB++;
    ruido(); //si los decibelios son superiores al umbral se
añade 1 al contador y se vuelve a mirar el valor
}
    cuentadB--; //si en el anterior if la función no vuelve a
comenzar el contador baja

```

```

    if(cuentadB < 0){
        cuentadB = 0;
    }
}

void imprimir(){
    Serial.print("Temperature: ");
    Serial.print(tempMedia);
    Serial.println(" *C ");
    Serial.print("Humidity: ");
    Serial.print(humeMedia);
    Serial.print(" %\t");
    //Serial.print("Contador: ");
    //Serial.println(cuentaTemp);
    delay(100);
    Serial.print("Amoniaco: ");
    Serial.print(partesNH3);
    Serial.println(" ppm ");
    Serial.print("CO2: ");
    Serial.print(partesCO2);
    Serial.print(" ppm ");
    delay(100);
    Serial.print("Caudal: ");
    Serial.print(litros);
    Serial.println(" l/min ");
    Serial.print("Ruido: ");
    Serial.print(dB);
    Serial.print(" dB ");
    delay(100);

    delay(500);
    lcd.setCursor(0, 0);
    lcd.print("TA:"); //Temperatura Ambiente
    lcd.print(tempMedia);
    lcd.print("*C ");
    lcd.print("RH:"); //del inglés Relative Humidity
    lcd.print(humeMedia);
    lcd.print("%");
    delay(100);
    lcd.setCursor(0, 1);
    lcd.print("NH3:"); //Amoniaco
    lcd.print(partesNH3);
    lcd.print("-");

```

```

        lcd.print("CO2:"); //Dióxido de carbono
        lcd.print(partesCO2);
        lcd.print("ppm");
    delay(100);
        lcd.setCursor(0, 2);
        lcd.print(litros);
        lcd.print("L/min");
        lcd.print(dB);
        lcd.print("dB");
    }

void loop() {
    tempMedia = leeTemp(); //se manda leer la temperatura de
    ambos sensores
    delay(100);
    humeMedia = leeHume();
    delay(100);
    amoniaco();
    delay(100);
    calidadAire();
    delay(100);
    aireacion();
    delay(100);
    ventilacion();
    delay(250);

    float frequency = GetFrequency(); // obtener frecuencia de
    caudal en Hz
    litros = frequency / factorK; // calcular caudal L/min
    corteCaudal();
    delay(100);
    ruido();
    delay(100);
    imprimir(); //se imprimen los valores
    delay(500);
    thingSpeakWrite(tempMedia, humeMedia, partesNH3, partesCO2,
    litros, dB); //Envía los valores a la función para escribir en
    ThingSpeak
    delay(16000); //ThingSpeak necesita al menos 15s para actualizar
}

```

6.4.2 Códigos de error

Tabla 6.15: Códigos de error en el programa.

Código	Descripción
01.01	Error sensor temperatura
01.10	Diferencia de temperaturas elevadas
01.20	Temperatura muy elevada
02.01	Error sensor humedad
02.10	Diferencia de humedades elevadas
03.20	Niveles de amoniaco elevados
04.20	Niveles de CO2 elevados
05.20	Caudal elevado
06.20	Ruido elevado