Trabajo Fin de Grado
Ingeniería de las Tecnologías de Telecomunicación
Sistemas Electrónicos

Sistema electrónico para la supervisión remota de parámetros de un acuario

Autor: Julia León Jiménez

Tutor: Juan de la Cruz García Ortega

Departamento de Ingeniería Electrónica
Área de tecnología electrónica
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2019

Trabajo Fin de Grado
Ingeniería de las Tecnologías de Telecomunicación

Sistema electrónico para la supervisión remota de parámetros de un acuario

Autor:
Julia León Jiménez

Tutor:
Juan de la Cruz García Ortega
Profesor titular

Departamento de Ingeniería Electrónica
Área de tecnología electrónica
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, 2019

Trabajo Fin de Grado: Sistema electrónico para la supervisión remota de parámetros de un acuario

Autor: Julia León Jiménez

Tutor: Juan de la Cruz García Ortega

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2019

El Secretario del Tribunal

A todo el que se ha cruzado en mi camino durante estos años

Agradecimientos

A mis profesores, por enseñarme todo lo que sé de esta bonita materia. En especial a Juan García, por su paciencia, constancia y enseñanza para que este proyecto salga adelante.

A mi gran familia, por creer siempre en mi, por aguantar conmigo los momentos difíciles de esta carrera y por celebrar mis triunfos como propios. A mis abuelos, por enseñarme gran parte de los valores que tengo hoy.

A mis referentes de vida, mis padres y mis hermanas María y Ángela, por enseñarme a ver siempre el lado positivo de las cosas y ser mi sostén. A Roscón, compañero fiel y silencioso de estudio.

A mis amigos, los que siempre estarán y los que estuvieron de paso, que me han acompañado durante esta carrera de fondo. Gracias por compartir cada momento conmigo, se que sin vosotros no hubiera llegado hasta aquí.

Resumen

En nuestro día a día tenemos presente todo tipo de dispositivos electrónicos que facilitan las comunicaciones, tanto con otras personas como con otros dispositivos electrónicos.

Por otra parte, actualmente, en un gran número de hogares hay mascotas, donde destacan, entre otros, los peces. Son muchos los hogares en los que encontramos un acuario. Éstos presentan en el agua contenedora parámetros que es necesario controlar de forma regular. Normalmente estos parámetros se controlan de manera manual, siendo necesaria la presencia de una persona.

El principal objetivo de este trabajo es diseñar un sistema que permita supervisar los parámetros más importantes del acuario de manera remota, a través de la red móvil o WiFi.

A lo largo del proyecto se llevará a cabo el desarrollo teórico y la programación necesarios para el diseño del sistema, además del montaje final del prototipo. Por último, se realizarán unas pruebas que muestren el correcto funcionamiento del sistema.

Abstract

In our daily life, we are in constant contact with every type of electronic devices which facilitate communication, both with other people and with other computerized gadgets.

On the other hand, there are a lot of people today who have a pet in their homes and fishes are one of the most common companion animals. There are a lot of homes in which we can find an aquarium. The water of these aquariums may have some parameters that should be monitored on a regular basis. These parameters are usually controlled manually, being the presence of a person necessary.

The main aim of this project is to design a system which allows the most important parameters to be monitorized remotely, that is through a mobile network or a Wi-Fi connection.

The theoretical basis will be held in this work, as well as the programming, so essential for the system design. Besides, the final assembly of the prototype will be shown. Finally, the proper functioning of the system will be tested.

Índice

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

1 INTRODUCCIÓN

No hay mejor práctica que una buena teoría.

- Kurt Lewin -

En este primer capítulo se realizará un breve análisis sobre la motivación para llevar a cabo este proyecto así como el objetivo de éste. Por otro lado, se detallará la estructura del presente documento.

1.1 Motivación

La presencia de Internet en nuestra rutina diaria nos ha llevado a controlar numerosas funciones a través de él, desde las luces o persianas de los hogares hasta las cuentas bancarias. Por ello, cada vez vemos más sencillo el uso de aplicaciones que nos faciliten y complementen el trabajo que tienen algunos sistemas, como por ejemplo el que vamos a realizar en este proyecto.

Este se encargará de la supervisión de parámetros de un acuario, función que podremos realizar gracias a las comunicaciones.

Por otro lado, gran parte de los sistemas fabricados para la supervisión de acuarios tienen unas prestaciones avanzadas que generalmente no se pretenden controlar en un acuario que no pertenece a una piscifactoría, este tipo de sistemas presentan un coste muy elevado.

Las razones anteriores han llevado a la realización de este proyecto, que permitirá la supervisión y futuro control remoto de los parámetros más importantes de estos ecosistemas.

1.2 Objetivo

El objetivo fundamental del proyecto es la realización de un sistema que se encargue de la supervisión de parámetros de un acuario tales como la temperatura, el nivel de agua y el estado de la red eléctrica, con el fin de obtener una notificación vía SMS que permita saber si el sistema sufre alguna anomalía en alguno de estos parámetros.

El alcanzar este objetivo pasa por una serie de fases que han de cumplirse previamente al objetivo fundamental, numeradas a continuación:

- ✓ Entender las condiciones normales a las que debe estar sometido un acuario para tener un perfecto ecosistema y establecer unos requerimientos a cumplir por el sistema de vigilancia.
- ✓ Comprender cómo se produce la interacción de Arduino con sensores externos, para la obtención de datos de los parámetros definidos.
- ✓ Realizar la elección correcta de los sensores necesarios para el sistema y realizar un estudio sobre su funcionamiento.
- ✓ Estudiar la conexión de Arduino con un servidor web a través de Wi-Fi.
- ✓ Programación del funcionamiento del sistema mediante el software IDE de Arduino.

- ✓ Construcción del prototipo del sistema.
- ✓ Realización de pruebas que confirmen el correcto funcionamiento del sistema diseñando, cumpliendo con los requerimientos establecidos en la primera fase.

Una vez concluidos estos pasos, se habrá obtenido el sistema completo.

1.3 Estructura

El presente proyecto constará de un total de 10 capítulos.

El primero de ellos, como hemos visto, trata una breve introducción del sistema a diseñar, así como los objetivos que debemos cumplir para obtener el proyecto deseado.

El segundo capítulo contemplará una breve historia sobre los acuarios y los parámetros que lo regulan. Por otro lado, se estudiará el estado del arte de los sistemas de supervisión, entendido como todos aquellos desarrollos realizados con el mismo fin que nuestro sistema, que son base teórica sobre la que se sustenta el presente proyecto.

El tercer capítulo analizará los parámetros más importantes a medir en un acuario y las condiciones que deben cumplir estos para tener un sistema estable y equilibrado para la supervivencia de los peces.

En el cuarto capítulo se abordará la filosofía que sigue un sistema de supervisión general y se describirá la solución elegida para realizar la supervisión de los acuarios de este proyecto, explicando el papel que jugará cada elemento dentro del sistema completo.

En el quinto se seleccionarán los componentes necesarios para llevar a cabo el esquema mostrado en el apartado anterior. Se elegirá el microcontrolador, el módulo necesario para la conexión remota con el sistema y los sensores que se utilizarán para medir los parámetros del agua deseados. Para proceder a la selección del mejor candidato para el prototipo final, se analizarán las características de diferentes dispositivos que realicen la misma función.

El sexto capítulo explicará el desarrollo software llevado a cabo para hacer posible la funcionalidad del sistema. En primer lugar, se explicará en detalle el software de programación utilizado, qué es un servidor web y cuál se ha usado en el proyecto. En segundo lugar, se presentarán mediante diagramas de flujo los sketch o programas que se han utilizado.

En un séptimo capítulo se tendrán las pruebas realizadas al prototipo para comprobar su correcto funcionamiento. Se someterá el sistema a tres situaciones anómalas distintas y se corregirán para comprobar que se deja de enviar la alarma de detección de fallo.

Para finalizar, en los dos últimos capítulos se expondrán las conclusiones del desarrollo de todo el proyecto, así como posibles mejoras para proyectos futuros.

2 ESTADO DEL ARTE

Un acuario es un pequeño o gran recipiente que contiene agua salada o dulce y permite mantener un ecosistema cerrado e independiente del entorno natural donde pueden vivir peces y otros animales acuáticos.

2.1 Historia

El concepto de acuario nació en el siglo XVIII, cuando las personas más acomodadas de cada país comenzaron a coleccionar conchas, plantas y animales marinos para el disfrute propio y de los invitados. Además, había la necesidad de transportar estos animales y plantas desde su lugar de origen hasta el lugar donde iban a exponerse. En estos años, los acuarios eran recipientes pequeños metálicos donde se incrustaban los cristales, por lo que la visión de su interior era muy reducida. El paso a los acuarios modernos se produjo en el año 1838 por Nathaniel Bagshaw Ward, que añadió más agua que tierra en estos recipientes.

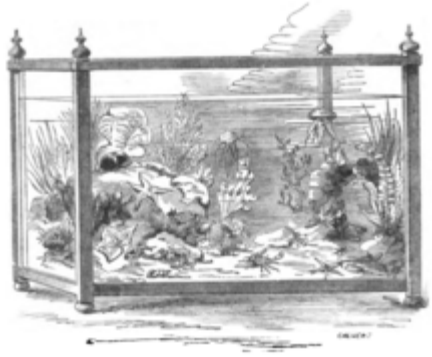


Figura 1: Acuario marino, año 1860

A pesar de que los animales y plantas más demandados eran los marinos, este tipo de acuarios tardó más en desarrollarse, ya que se desconocían los medios marinos y no eran capaces de recrearlos y mantenerlos. Estos acuarios vivieron su gran avance después de la Segunda Guerra Mundial, con el desarrollo del buceo, forma por la cual se comenzaron a conocer los fondos marinos.

2.2 Clasificación

Existen diversos tipos de acuarios según su finalidad: acuario de reproducción, acuario de biotopo, acuario comunitario... En el presente proyecto nos centraremos en este último, donde pueden convivir peces y plantas de distintas especies, ya que son los ecosistemas más demandados para uso doméstico.

Por otro lado, a su vez, los acuarios se clasifican en función de la salinidad del agua que contiene, dividiéndose en tres grupos:

- Acuario de agua dulce: Concentración salina <0.5%.
- Acuario de agua salada: Concentración salina 5% - 18%.
- Acuario de agua salobre: Concentración salina 0.5% - 5%.

En este proyecto nos centraremos en los acuarios de agua dulce, ya que son los más comunes debido a su reducido tamaño y fácil mantenimiento.

Figura 2: Acuario de agua dulce

2.3 Sistemas desarrollados

Los acuarios están presentes en gran parte de los hogares del mundo, ya que existe el concepto de que los peces son animales independientes que no necesitan demasiada atención para su cuidado, pero la realidad es que necesitan unos parámetros bien regulados.

A lo largo de los años, se han creado sistemas comerciales y sistemas de software libre para el control de parámetros de un acuario. En este proyecto vamos a destacar dos de estos sistemas: Reef Angel Aquarium Controller y Aquatronica.

2.3.1 Reef Angel Aquarium Controller

Este sistema de control ha sido creado por la empresa irlandesa Reef Angel, asentada en Dublín. Consta de un controlador que monitoriza numerosos parámetros del acuario: pH, temperatura, luz, calcio... y nos permite controlarlos de manera remota. Este controlador puede complementarse con diferentes accesorios o expansiones, que permiten controlar otros parámetros como la salinidad, la humedad o el nivel de agua.

Es un sistema muy avanzado y cómodo de utilizar, pero tiene un costo muy elevado, rondando los 500€ aquellos que poseen las funciones más simples y pudiendo alcanzar más de 1000€ aquellos sistemas que presten todos los avances logrados en el control de parámetros.

Figura 3: Reef Angel Controller

2.3.2 Aquatronica

Aquatronica es un sistema que garantiza automatización y seguridad en el control de acuarios. El poseedor del sistema puede controlar todos los dispositivos eléctricos y electrónicos conectados al acuario mediante un simple controlador. El sistema de control se compone de 4 unidades principales: unidad de control, unidad de potencia, interfaces/módulos y electrodos y otros.

Cada sistema de Aquatronica es capaz de controlar hasta 30 sensores diferentes, incluso en distintos acuarios, permitiendo también la conexión remota.

Entre los sensores de medición cabe destacar los más importantes:

- Electrodo PH: Mide el nivel de pH en el agua de manera continua.
- Sensor de oxígeno disuelto: Mide el nivel de oxígeno disuelto de forma continua en el agua.
- Sensor de nivel: Controla el nivel de agua del acuario, permitiendo rellenar el agua gracias al programa de controller.
- Sensor de temperatura: Mide la temperatura del agua de manera continua.
- Sensor de luz: Mide las condiciones de luz del acuario, teniendo dos valores: noche y día.

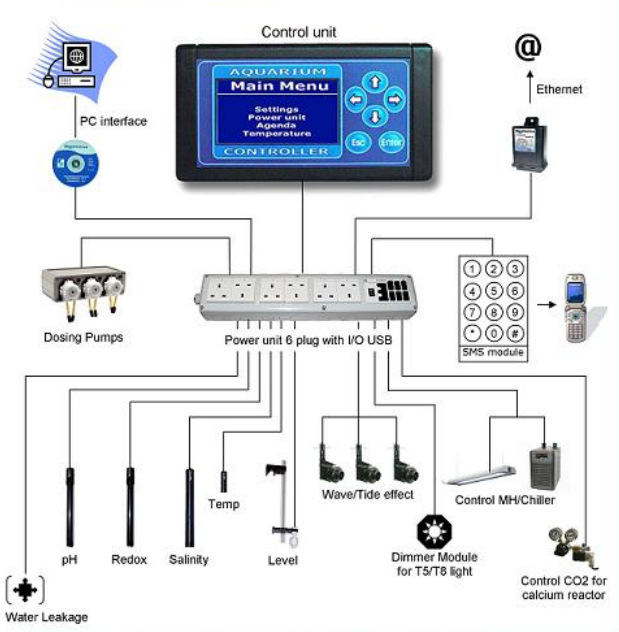


Figura 4: Aquatronica Controller

Como en el caso de Reef Angel, a pesar de ser un sistema completo con unas prestaciones fantásticas, su principal incoveniente es el precio, siendo 650€ para el kit más básico de control y 1009€ la versión deluxe.

3 PARÁMETROS DEL AGUA A SUPERVISAR

El presente capítulo analiza los parámetros más importantes a supervisar en un acuario, así como las condiciones que deben cumplir para formar un buen ecosistema.

De los parámetros analizados en este capítulo, el sistema físico a implementar en el proyecto final supervisará el nivel de agua, la temperatura y, a través de una cámara web, se obtendrán imágenes de la pecera en directo.

3.1 Temperatura

La temperatura del acuario es un factor muy importante para la supervivencia y salud de los peces, se debe tener en cuenta que el oxígeno del agua disminuye al aumentar la temperatura. Por tanto, hay que garantizar que este parámetro esté comprendido en un rango adecuado.

Por otro lado, al variar la temperatura del agua, varía la temperatura corporal de los peces, ya que no poseen mecanismos de termorregulación. Por ello, hay que mantener la temperatura dentro de un rango en función del tipo de peces que formen el acuario.

Según el tipo de agua que requieran los seres vivos del acuario, se puede distinguir:

Tabla 1: Rango de temperatura en función del tipo de agua

Agua	Rango de Temperatura
Fría	[0,18] ºC
Tropical/Templada	[18,28] ºC

3.1.1 Unidad de medida de la temperatura

Para medir este parámetro existen distintos tipos de unidades de medida, las tres más usuales son:

Grados Kelvin (K): Es una medida absoluta, ya que parte del cero absoluto. Es la temperatura teórica más baja posible. Utilizada en el Sistema Internacional de Unidades desde 1954. Fue creada por William Thomson, sobre la base de grados Celsius, marca el cero absoluto en -273’15ºC.

Grados Celsius (ºC): Es una medida relativa, ya que se compara con un proceso fisicoquímico establecido que siempre se produce a la misma temperatura. Para definir los grados Celsius o centígrados, se establece, en una atmósfera de presión, el punto de congelación del agua a 0ºC y el punto de ebullición del agua a 100ºC, dividiendo la escala en 100 partes iguales en las que cada una corresponde a 1 grado. Esta escala fue propuesta en 1742 por el físico y astrónomo sueco Anders Celsius.

Grados Fahrenheit (ºF): Es una medida relativa de la temperatura. Para definir los grados Fahrenheit, se estalece la temperatura de congelación del agua a 32ºF y la de ebullición del agua a 212°F.

En la siguiente imagen (*Figura 5*) se observa la comparación entre estas tres medidas de la temperatura.

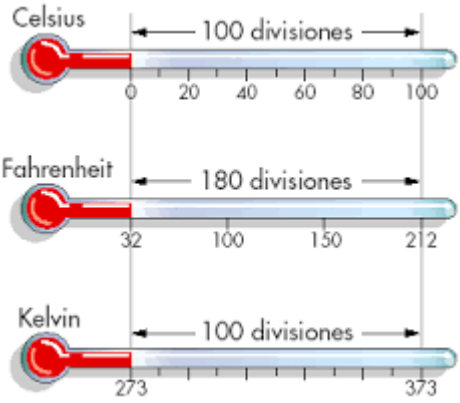


Figura 5: Comparación ºC - ºF - ºK

En el caso del sistema de supervisión de acuario desarrollado, es conveniente saber la presición necesaria en decimales de la temperatura medida. Tanto los peces de agua dulce como los de agua salada se mantienen a una temperatura aproximada de 24 ºC, sin embargo, los de agua salada son más sensibles a cambios ligeros de temperatura. Sabiendo esto, se determina que es necesario contar con una medida de temperatura con al menos un decimal. Para ello, se requiere un sensor de temperatura digital de al menos 8 bits de precisión.

3.2 PH

El potencial de hidrógeno o pH es una medida de acidez, neutralidad o alcalinidad de una solución acuosa. Indica la cantidad de iones de hidrógeno que existen en el agua.

Al ser logarítmica la escala en la que se mide, un pH de 5 es diez veces más ácido que un pH de 6. Por tanto, se debe controlar el pH de manera regular, ya que un pequeño cambio puede provocar estrés en los peces.

Tabla 2: Tipo de agua en función del nivel de pH

Nivel de pH	Agua
pH < 7	Ácida
pH = 7	Neutra
pH > 7	Alcalina

El rango usual de pH que debe haber en un acuario para tener unas condiciones adecuadas es [7.8,8.5], aunque el rango óptimo es [8.1,8.3], más complicado de mantener. Por tanto, la precisión necesaria es, al menos, de un decimal.

3.2.1 Unidad de medida del pH

Se mide como el logaritmo negativo en base 10 de la actividad de los iones de hidrógeno, correspondiéndose con la siguiente fórmula:

$$pH = -log_{10}(a_{H^+})$$

Para entender este fenómeno de iones de hidrógenos liberados en el agua, se debe realizar un breve resumen del proceso de electrólisis del agua. Éste se corresponde con la descomposición del agua en los gases hidrógeno y oxígeno, gracias a una corriente eléctrica continua aplicada entre dos electrodos, obteniéndose así una reacción de oxidación – reducción, en la que se ceden y captar electrones.

3.3 Dureza

La dureza permite conocer la mineralización del agua, es decir, la cantidad de sales minerales que hay disueltas en ella, mayoritariamente está determinada por los iones de calcio y magnesio, al ser éstos los más abundantes. Esta medida está relacionada con la conductividad, ya que midiendo este parámetro se puede obtener una buena aproximación de la dureza.

La dureza está determinada por la cantidad de carbonato de calcio disuelto en el agua, constituyendo aproximadamente el 90% de las sales disueltas.

Tabla 3: Dureza del agua según la zona a la que pertenece

Zona de Agua	Dureza
Zona lluviosa	Baja
Zona calcárea	Alta
Zona floresta	Baja

La dureza se divide en dos grupos principales:

- Dureza permanente (GH): Relativa a la presencia de iones de calcio y magnesio en el agua.
- Dureza temporal (KH): Relativa a la presencia de iones de carbonato y bicarbonato en el agua.

Por otro lado, el agua se puede clasificar según la dureza que presenta:

Tabla 4: Tipo de agua según dureza de ésta

Dureza (ppm)	Agua
0 – 50	Muy blanda
50 – 100	Blanda
100 – 150	Poco dura
150 – 300	Dura
> 300	Muy dura

3.3.1 Unidad de medida de la dureza

La unidad de medida utilizada de forma más usual para la medida de la dureza del agua es el grado hidrométrico francés (ºHF). El cálculo de este parámetro se realiza con la siguiente fórmula:

$$\frac{\frac{mg}{l}\ Ca\ x\ 2.5\ +\ \frac{mg}{l}\ Mg\ x\ 4.2}{10}$$

Otra de las formas más comunes de obtener la medida de la dureza de un acuario es a partir de la conductividad eléctrica, relacionada directamente con las sales o iones que están disueltos en el medio acuoso, en la siguiente siguiente gráfica (*Figura 6*) podemos observar la relación entre ambas magnitudes.

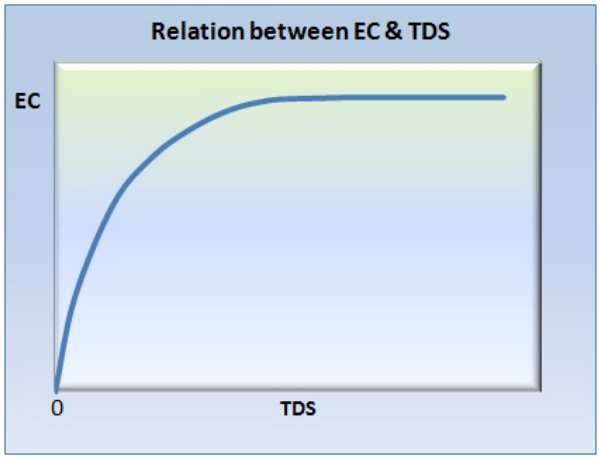


Figura 6: Relación entre la Conductividad Eléctrica y el Total de Sólidos Disueltos

A medida que aumenta la cantidad de sólidos disueltos, aumenta la conductividad eléctrica, sin embargo, cuando la concentración de sólidos llega a un nivel determinado, ya no está directamente relacionado con la conductividad. La conversión del TDS a la conductividad eléctrica puede ser realizada mediante la siguiente ecuación:

$$TDS\ (ppm)\ =\ 0.64\ x\ EC\ (\mu S/cm)\ =\ 640\ x\ EC\ (dS/m)$$

donde 1 ppm = 1 mg/L es la unidad de medida para sólidos disueltos.

Para un acuario, la dureza ideal se encuentra entre la dureza blanda y la poco dura.

3.4 Nivel de agua

El nivel de agua del sistema es uno de los parámetros más importantes a supervisar, debido a que un elevado descenso o un gran aumento de éste puede llevar a la pérdida de todos los seres vivos que alberga el acuario. Podemos diferenciar un nivel adecuado para dos casos:

- Acuario sin tapa: Hay que tener en cuenta que el acuario no debe estar demasiado lleno, ya que, si se produce un cambio brusco de temperatura, los peces pueden saltar. Por tanto, se ha decidido que un nivel idóneo será 6-8 cm por debajo del máximo nivel de la pecera.
- Acuario con tapa: El nivel apto de este tipo de acuarios se corresponde con el máximo que permita, ya que no hay posibilidad de pérdida de seres vivos por salto o desbordamiento.

4 ESQUEMA GENERAL DEL SISTEMA DE SUPERVISIÓN

4.1 Filosofía de un Sistema de Supervisión

El objetivo principal de los sistemas de supervisión es poder monitorizar los procesos de forma que se consiga un correcto funcionamiento incluso en situaciones anómalas.

Los sistemas de suspervisión son capaces de automatizar tareas como la detección de fallos, el análisis de datos o la recogida de estos. Para que esto sea posible, es necesario disponer de la mayor cantidad de información, por ello, el presente sistema cuenta con sensores que recogen información del agua de los acuarios. Estos sensores pueden ser análogicos o digitales. En el caso de los dispositivos análogicos es necesario usar un convertidor a digital para que el microcontrolador pueda procesar la información.

Por otro lado, además de los dispositivos que recojan la información, también es necesaria una herramienta capaz de mostrar al usuario final los fallos detectados en la recogida de estos datos. Por ello, se cuenta en el presente proyecto con un servidor web.

4.2 Descripción General del Sistema de Supervisión

El sistema completo constará de los bloques mostrados en la *Figura 7*.

En primer lugar, será gobernado por un **microcontrolador**, encargado de examinar las señales de entrada dadas por los sensores incorporados: sensor de temperatura, sensor de nivel de agua y relé, con el fin de detectar anomalías en el comportamiento de estos dispositivos y enviar los datos recogidos al módulo Wi-Fi.

Por otro lado, el sistema se complementará con un **teléfono móvil** o un **ordenador** a través de los cuales se supervisarán los parámetros de los acuarios, visitando el servidor web donde se encuentran albergadas las alarmas de comportamiento incorrecto detectadas por parte del microcontrolador.

Para hacer posible la comunicación entre el sistema incorporado en el acuario y el servidor MQTT al que se accede a través del teléfono móvil u ordenador, se hará uso de un **módulo WiFi**. Este módulo recibirá los datos por parte del microcontrolador y se encargará de procesarlos y enviarlos al servidor MQTT correspondiente a través de internet.

Por último, se contará con una **batería auxiliar** que comenzará a funcionar en caso de que se produzca un corto en la red eléctrica y ésta deje de alimentar el sistema. Esta batería se cargará continuamente al estar conectada a la red, por lo que se garantiza que, si la red eléctrica fallase, el sistema no se apagará. Para avisar al usuario de que el sistema está siendo alimentado por la batería debido a un fallo en la red, el sistema contará con un relé conectado a ésta, que cuando conmute enviará una señal al microcontrolador y este, como con las señales de los demás sensores, se encargará de enviarla al módulo Wi-Fi para publicar un mensaje en el servidor web.

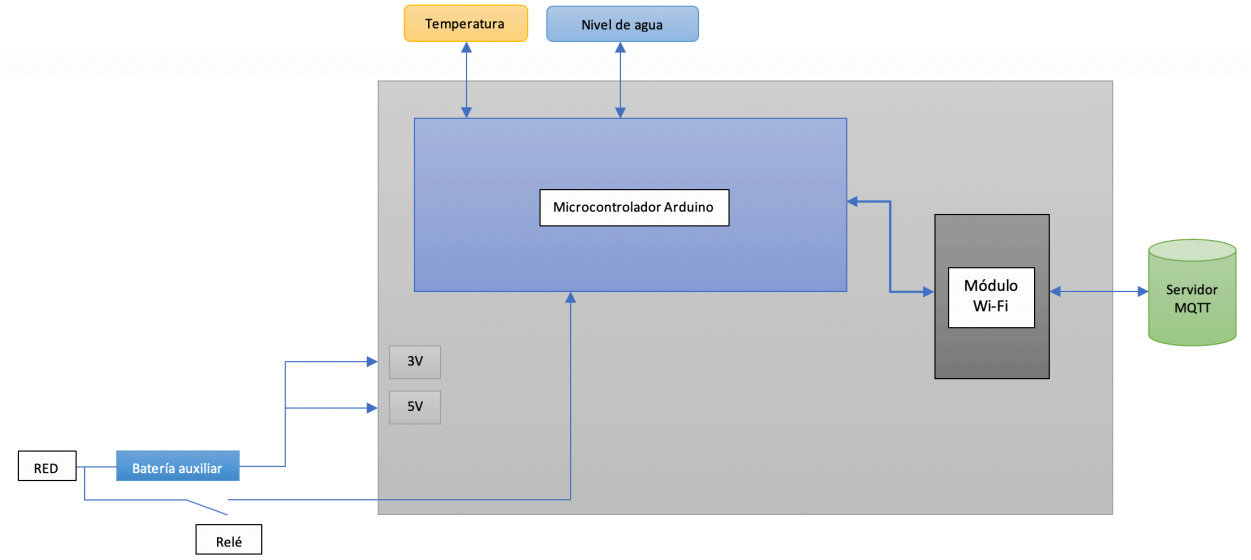


Figura 7: Esquema general del sistema de supervisión

5 ESTRUCTURA HARDWARE

El capítulo 5 se centrará en el análisis de la estructura hardware del sistema, es decir, el análisis de cada uno de los componentes elegidos para llevar acabo las funciones requeridas por el sistema.

5.1 Microcontrolador

5.1.1 ¿Qué es un microcontrolador?

El componente principal e indispensable para que el sistema funcione es el microcontrolador, encargado de responder a las señales de entrada dada por los sensores, automatizar procedimientos y procesar información. Para ello, el microcontrolador está compuesto, como mínimo, por tres elementos principales:

- Microprocesador: Encargado de ejecutar programas mediante instrucciones programadas en un lenguaje de bajo nivel. Está formado por:
  - ALU: Unidad Aritmeticológica, su función principal es realizar operaciones lógicas, aritméticas y miscelaneas.
  - Unidad de control: Conjunto de sistemas digitales secuenciales encargados de la distribución de la lógica de las señales.
  - Registros: Son las memorias principales de los procesadores.
- Memoria: El microcontrolador consta de 3 tipos de memoria: FLASH, RAM y EEPROM.
- Periféricos: Encargados de la comunicación con otros dispositivos, de leer sensores, habilitar o deshabilitar actuadores.

Para este sistema se necesitará un microcontrolador que cumpla con los siguientes **requisitos**:

- Entradas: Deberá contar con, al menos, 10 entradas.
- Salidas: Deberá contar con, al menos, 3 salidas.
- Convertidores Analógico/Digitales: Deberá contar con, al menos, 1 CAD.
- Canales TX – RX: Deberá contar con, al menos, 3 canales TX – RX.

Podemos encontrar una familia de microcontroladores que lidera el mercado gracias a su relación calidad/precio, estos son los microcontroladores Arduino. Por ello, se analizarán los distintos componentes que forman esta familia de microcontroladores para así poder elegir el que más se adecue al sistema a desarrollar.

5.1.2 Arduino

Arduino es una plataforma tanto de software como de hardware libre que diseña placas de desarrollo hardware para su uso en proyectos que requieran controlar y supervisar objetos del mundo real. Se desarrolló como una herramienta didáctica en el Interaction Design Institute Ivrea de Italia, en el año 2003. El objetivo principal del desarrollo de esta herramienta fue facilitar la creación de dispositivos por parte de estudiantes y reducir el coste de estos.

En cuanto a software, Arduino cuenta con un entorno de desarrollo integrado (IDE) particular, formado por un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Desde este IDE, que implementa el lenguaje de programación de Arduino, se pueden desarrollar aplicaciones que se ejecutarán en la placa hardware, ya que el programa incluye la opción de carga en la memoria flash de ésta.

Existen múltiples placas de microcontroladores. A continuación, se analizarán algunos de ellos:

- Arduino UNO: Es la placa más extendida de Arduino, al ser la primera en comercializarse. Se puede hablar de la placa básica de esta familia, ya que la mayoría de sus sucesoras implementan las características de esta.

Figura 8: Placa Arduino UNO

Tabla 5: Características Arduino UNO

Microcontrolador	ATmega328
Alimentación (V)	5 V
Pines I/O Digitales	14
Pines Analógicos	6
Memorias FLASH / SRAM / EEPROM	32KB / 2KB / 1KB

- Arduino PRO: Placa con el mismo microcontrolador que el Arduino UNO, con una alimentación menos exigente, pero, a su vez, con menos prestaciones en cuanto a memoria.

Figura 9: Placa Arduino PRO

Tabla 6: Características Arduino PRO

Microcontrolador	ATmega328
Alimentación (V)	3.3 V
Pines I/O Digitales	14
Pines Analógicos	6
Memorias FLASH / SRAM / EEPROM	16KB / 1KB / 512B

- Arduino Due: Destaca frente al resto de placas de Arduino por su potente velocidad de trabajo (84 MHz), su memoria RAM (96 KB) y su memoria flash (512 KB), otorgando grandes prestaciones para proyectos de alto redimiento.

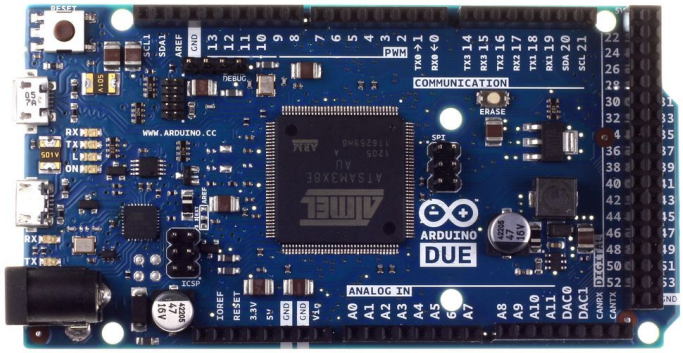


Figura 10: Placa Arduino DUE

Tabla 7: Características Arduino DUE

Microcontrolador	AT91SAM3X8E
Alimentación (V)	3.3 V
Pines I/O Digitales	54
Pines Analógicos	12
Memorias FLASH / SRAM	512 KB / 96 KB
Velocidad del reloj	84 MHz

- Arduino MEGA 2560: Es un modelo superior al Arduino UNO, cuenta con un mayor número de pines digitales y analógicos, además de presentar mayor memoria.

Figura 11: Placa Arduino MEGA

Tabla 8: Características Arduino MEGA

Microcontrolador	ATmega2560
Alimentación (V)	5 V
Pines I/O Digitales	54
Pines Analógicos	16
Memorias FLASH / SRAM / EEPROM	256 KB / 8 KB / 4 KB
Velocidad del reloj	16 MHz

- Arduino YÚN: Es una placa similar al Arduino UNO, con la cualidad de incluir soporte WiFi, Ethernet y tarjeta microSD sin necesidad de shields.

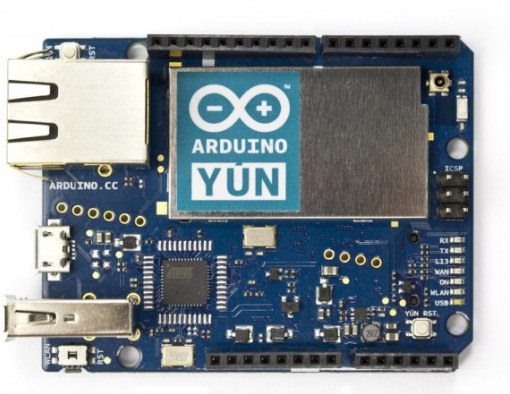


Figura 12: Placa Arduino YÚN

Tabla 9: Características Arduino YÚN

Microcontrolador	ATmega32u4
Alimentación (V)	5 V
Pines I/O Digitales	20
Pines Analógicos	12
Memorias FLASH / SRAM / EEPROM	32 KB / 2.5 KB / 1KB
Velocidad del reloj	16 MHz
Ethernet / WiFi	Sí

- Arduino Mega ADK: Es una placa basada en el Arduino Mega 2560 visto anteriormente, que incorpora una interfaz preparada para conectarse a dispositivos Android a través de USB.

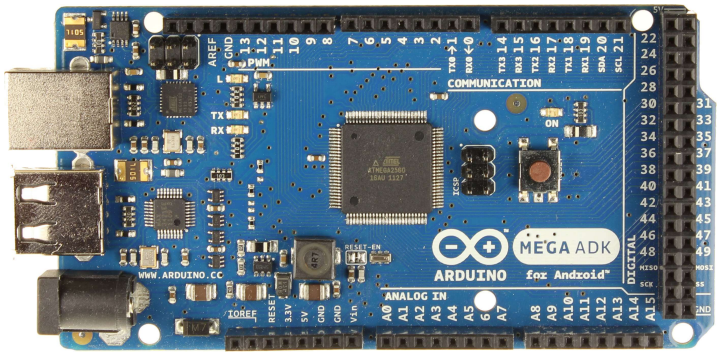


Figura 13: Placa Arduino MEGA ADK

Tabla 10: Características Arduino MEGA ADK

Microcontrolador	ATmega2560
Alimentación (V)	5 V
Pines I/O Digitales	54
Pines Analógicos	16
Memorias FLASH / SRAM / EEPROM	256 KB / 8 KB / 4 KB
Velocidad del reloj	16 MHz

- Arduino LILYPAD: Placa de Arduino diseñada para su integración en prendas.

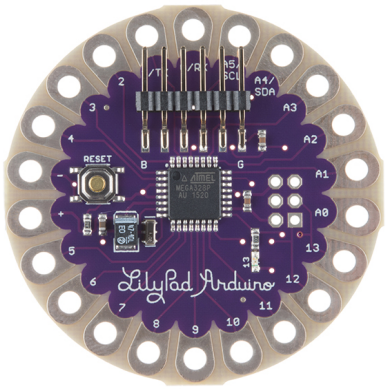


Figura 14: Placa Arduino LILYPAD

Tabla 11: Características Arduino LILYPAD

Microcontrolador	ATmega168V
Alimentación (V)	2.7 – 5.5 V
Pines I/O Digitales	14
Pines Analógicos	6
Memorias FLASH / SRAM / EEPROM	16 KB / 1 KB / 512 B
Velocidad del reloj	8 MHz

Una vez conocidas estas placas y sus características, y basándose en los requisitos que se analizan en el apartado 5.1, el microcontrolador Arduino seleccionado para el desarrollo del sistema de supervisión será el Arduino MEGA 2560.

5.2 Módulo Wi-Fi

Para hacer posible la transmisión de datos, se requiere que el sistema cuente con la conocida conexión Wi-Fi (Wireless Fidelity).

5.2.1 ¿Qué es Wi-Fi?

W-iFi es una red inalámbrica que permite la conexión entre dispositivos electrónicos. Nació en 1999 de la unión de varias empresas para formar la WECA (Wireless Compatibility Alliance), actualmente llamado Alianza Wi-Fi, cuyo objetivo principal es fomentar la tecnología inalámbrica y garantizar la compatibilidad de dispositivos. Esta compatibilidad e interoperabilidad entre equipos se forjó bajo la norma IEEE 802.11b en el año 2000.

5.2.2 Shields Wi-Fi para Arduino

La función de envío de datos vía internet se realizará a través de un módulo Wi-Fi conectado al microcontrolador del sistema.

En el mercado de Arduino podemos encontrar, principalmente, los siguientes módulos Wi-Fi:

- Shield NodeMCU WiFi ESP8266: Circuito integrado que permite obtener conexión Wi-Fi y que

soporta el protocolo TCP/IP.

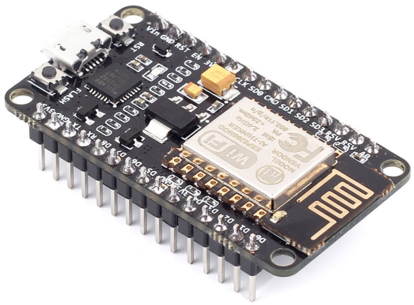


Figura 15: Shield Wi-Fi ESP-01 8266

Tabla 12: Características ESP-01 8266

Voltaje de operación	3 – 3.6 V
Temperatura de operación	[-40, 125] ºC
Tamaño del procesador	32 bits
Memoria	512 kB – 1 MB
Pines	10 GPIO, I2C, 1-WIRE
Estándar Wi-Fi	802.11b/n/g
Protocolos soportados	IPv4, TCP / UDP / HTTP / FTP

Se analiza sólo esta opción ya que es la más extendida en el mundo de Arduino, estando incluida dentro del microcontrolador Arduino UNO Wi-Fi. Su popularidad se debe a su bajo coste, bajo consumo y buenas prestaciones.

5.3 Sensores

5.3.1 Sensor de nivel de agua

Los sensores de nivel de agua compatibles con Arduino que lideran el mercado son, principalmente, dos. Las diferencias entre ambos son notables y se analizan a continuación:

- Boya de nivel: Actúan como un interruptor, proporcionando una entrada digital para Arduino.

Figura 16: Boya de nivel

Tabla 13: Características boya de nivel

Tipo de sensor	Digital
Voltaje de alimentación	5 V
Rango de temperatura de funcionamiento	[-10,85] ºC
Número de pines	2

- Sensor de nivel: Ofrece una señal analógica proporcional a la cantidad de agua que detecta, cuando se producen pequeñas modificaciones.

Figura 17: Sensor de nivel de agua

Tabla 14: Características sensor de nivel

Tipo de sensor	Analógico
Voltaje de alimentación	3.3 – 5 V
Corriente	< 20 mA
Rango de sensibilidad a la humedad	10% – 90%
Número de pines	3

- Sensor de ultrasonido: Permite medir la distancia entre el sensor y el agua, con ello se puede saber si el acuario se ha llenado.

Figura 18: Sensor de ultrasonido

Tabla 15: Características sensor de ultrasonido

Tipo de sensor	Digital
Voltaje de alimentación	5 V
Corriente	< 2 mA
Distancia de detección	2 – 400 cm
Número de pines	5

Finalmente, se decide hacer uso de las boyas de nivel, ya que para los requisitos del sistema son los más económicos y sencillos, pudiendo detectar un aumento/disminución del sistema en el rango de temperaturas al que va a estar sometido el sistema completo en un instante.

5.3.2 Sensor de temperatura

Para la medida de la temperatura del agua se utilizará un sensor de temperatura compatible con Arduino. A continuación, se analizarán los parámetros más importantes de los sensores más populares en el mercado de Arduino, con el fin de seleccionar el que mejor se adapte a los requerimientos del proyecto.

- Sensor de temperatura LM35: Este sensor analógico está calibrado en grados centígrados, por lo que no es necesario el uso de un conversor.

  El principal problema de este sensor es que para medir temperaturas en el rango [-55, 2] ºC es necesario aplicar un voltaje negativo, lo que es imposible usando Arduino, por lo que se descarta su elección. Además de no ser sumergible.

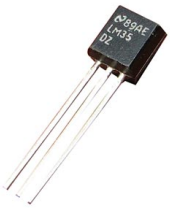


Figura 19: Sensor de temperatura LM35

Tabla 16: Características LM35

Tensión de alimentación	4 – 30 V
Rango de valores	[-55, 150] ºC
Precisión	± 0.5 ºC
Precio	0.70 €

- Sensor de temperatura TMP36: Se diferencia del sensor anterior, principalmente, en dos aspectos: no es necesaria la administración de una alimentación negativa y la precisión de la medida es superior en ± 2 ºC.

  El principal problema de este sensor es su precisión y su precio, habiendo en el mercado sensores con

mejor prestación y menor precio, por lo que se descarta su elección. Además de no ser sumergible.

Figura 20: Sensor de temperatura TMP36

Tabla 17: Características TMP36

Tensión de alimentación	2.7 – 5.5 V
Rango de valores	[-40, 50] ºC
Precisión	± 2 ºC
Precio	2.5 €

- Sensor de temperatura DHT22: El sensor DHT22 mide tanto temperatura como humedad.

  El inconveniente más destacable de este sensor para el sistema de supervisión es su precio, habiendo en el mercado sensores con mejores prestaciones y precio menor. Además de no ser sumergible.

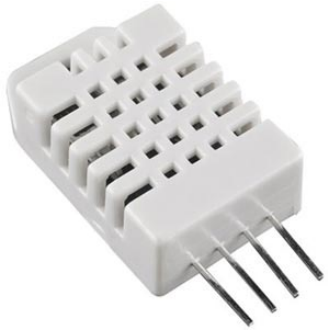


Figura 21: Sensor de temperatura DHT22

Tabla 18: Características DHT22

Tensión de alimentación	3.3 – 6 V
Rango de valores	[-40, 80] ºC
Precisión	± 0.5 ºC
Precio	4.50 €

- Sensor de temperatura DS18B20: La ventaja principal de este sensor es que cuenta con una sonda sumergible en líquido, por lo que es ideal para el sistema de supervisión. Por otro lado, para el rango de temperaturas requerido, el sensor tiene una precisión de ± 0.5 ºC.

  Otra de las ventajas de este sensor es que utiliza el protocolo 1-Wire, el cual permite conectar varios sensores DS18B20 con un solo cable, así solo será necesario usar un pin de Arduino.

  Además, el precio de este sensor es inferior a los anteriores, lo que hace más atractiva su compra.

Figura 22: Sensor de temperatura DS18B20

Tabla 19: Características DS18B20

Tensión de alimentación	3 – 5.5 V
Rango de valores	[-55, 125] ºC
Precisión	± 0.5 ºC – ± 2 ºC
Precio	0.98 €

Debido a todas las ventajas que presenta, se decide hacer uso de sensores de temperatura DS18B20, económicos en precio y recursos del microcontrolador Arduino.

5.3.3 Relé de alimentación

Para informar al usuario de que se ha producido un corte en la red eléctrica, el sistema contará con un relé que se encuentra conectado a esta. Cuando se produce un corte, el relé conmutará y el microntrolador recibirá, en uno de sus pines digitales, una señal a 0. Esta información será recogida y procesada por el sistema con el fin de publicar en el servidor web un mensaje de aviso.

En el sistema general presentado en el punto 4, se puede observar la posición del relé respecto al sistema.

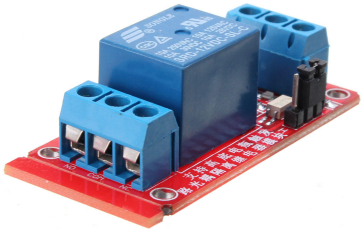


Figura 23: Relé de alimentación 220V

5.4 Batería auxiliar

Para la supervivencia de los animales marinos que residen en los diferentes acuarios, es imprescindible contar con una batería auxiliar que se active en caso de corte en la red eléctrica que alimenta el sistema de supervisión.

Se encuentran distintos modos de alimentar el microcontrolador Arduino, núcleo del sistema.

- Pila de 9V: Una de las opciones más utilizadas consiste en aplicar una tensión de 9V a través del jack de Arduino. Su precio no es asequible a largo plazo, ya que no son baterías recargables.

Figura 24: Pila 9V

- 5 baterías recargables AA de 1.2V: La opción de utilizar 5 baterías recargables es similar a la anterior, con la diferencia de ser rentables a largo plazo si se cuenta con un cargador.

Figura 25: Baterías recargables AA de 1.2 V

- Batería recargable USB: Las baterías USB proporcionan 5V regulados a su salida, lo que hace innecesario el uso de un regulador de voltaje.

Figura 26: Batería recargable USB

Para el montaje final, se ha decidido hacer uso de una batería recargable USB, conectada directamente al Arduino y al módulo de Wi-Fi.

5.5 Ordenador

El elemento auxiliar del sistema, encargado, entre otras cosas, de tener conexión a internet para poder hacer consultas al servidor web sobre los mensajes publicados, será un ordenador.

Por otro lado, éste tendrá la labor de depurar código en el caso de fallo software del sistema.

Para poder llevar a cabo los dos puntos anteriores se requiere que el ordenador cuente con un sistema actualizado, conexión a internet vía cable o Wi-Fi y con la versión más reciente del software de programación especificado, en este caso IDE de Arduino.

6 ESTRUCTURA SOFTWARE

6.1 IDE Arduino

La programación de las funcionalidades que se van a implementar en el dispositivo Arduino se llevará a cabo en una aplicación distribuida por el propio fabricante de los microcontroladores Arduino. Esta aplicación, disponible para los sistemas operativos más habituales (macOS, Windows, Linux), puede obtenerse mediante la página web oficial de Arduino (www.arduino.cc) o en las tiendas oficiales de estos sistemas operativos.

Los archivos generados en esta aplicación, cuya extensión es ino, se denominan sketch. Un microcontrolador o módulo compatible con Arduino puede albergar un único sketch en su sistema.

En cuanto la aplicación se abre, aparecerá el cuerpo principal de un sketch de Arduino, que consta de dos funciones principales:

- void setup(): Función que recoge la inicialización de todos los objetos y variables del sistema así como las funcionalidades que se quieren ejecutar una sola vez.
- void loop(): Función bucle en la que se desarrollará la parte del programa que se ejecuta de manera continua.

A estas dos funciones principales, se pueden sumar todas las secundarias que el desarrollador considere oportunas para su proyecto.

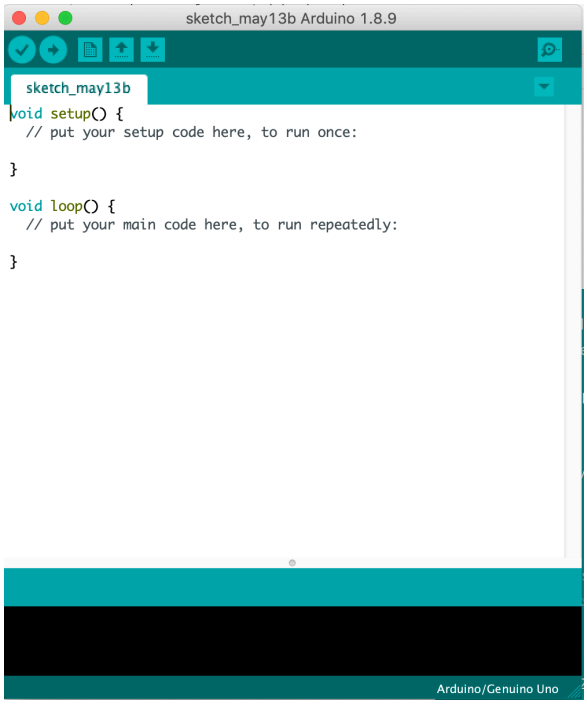


Figura 27: Pantalla principal IDE Arduino

La aplicación consta de dos barras de menú principales:

- Barra de menú extendida: Se corresponde con la barra de menú común que presentan la mayoría de los programas de edición existentes. Desde ella se pueden configurar todos los parámetros

relacionados con el sistema tales como librerías o tipo de módulo conectado, compilar un programa o verificar si el código desarrollado presenta errores de lenguaje.

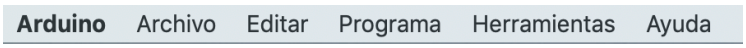


Figura 28: Barra de menú extendida IDE Arduino

- Barra de menú de acciones principales: La aplicación presenta una barra de iconos que contiene las acciones principales más usadas a la hora de desarrollar un sketch. Estas acciones son, de izquierda a derecha de la imagen, verificar, subir sketch al módulo, nuevo sketch, abrir sketch, salvar sketch y, por último, a la derecha del todo, la apertura del monitor serie.

Figura 29: Barra de menú acciones principales IDE Arduino

6.2 Servidor MQTT (Message Queue Telemetry Transport)

6.2.1 Protocolo MQTT

El protocolo conocido como protocolo MQTT se utiliza para la comunicación M2M (Machine to Machine) en el IoT (Internet of Things) o internet de las cosas. Es uno de los protocolos más utilizados debido a que requiere de muy pocos recursos para realizar la comunicación, en nuestro caso, entre el Arduino y un servidor web.

La estructucta o topología de este protocolo es en estrella. Cuenta con un nodo central, conocido como **broker**, que se encarga de la transmisión de paquetes entre los **clientes** y de la gestión de la red.

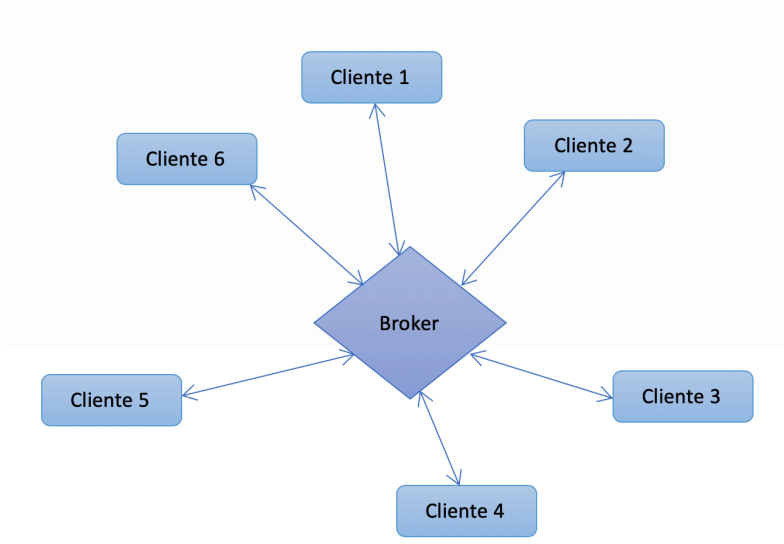


Figura 30: Topología protocolo MQTT

Para que la comunicación broker – cliente sea posible, es necesario hacer uso de un elemento del mensaje conocido como **topic**, al que tanto cliente como broker deben estar suscritos. Los topics cuentan con una arquitectura jerárquica en la que de uno de ellos pueden colgar otros, pudiendo suscribirse el cliente en el nivel que desee.

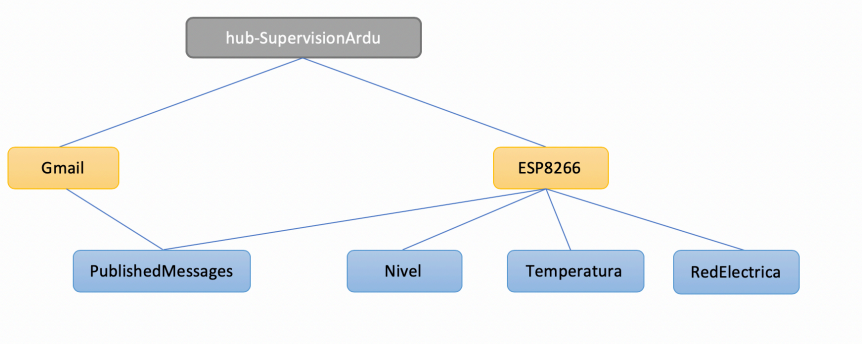


Figura 31: Topología del sistema

Este protocolo contiene una serie de mensajes principiales necesarios para poner en funcionamiento un servidor MQTT:

- CONNECT: Mensaje que se encarga de establecer una conexión entre el broker y el cliente.
- CONNACK: Mensaje del protocolo MQTT que recibe el cliente una vez que la conexión se ha establecido.
- SUBSCRIBE: Mensaje que envía el cliente al broker para indicar que quiere recibir los mensajes de uno o varios topics abiertos en el servidor.
- SUBACK: Mensaje que recibe el cliente por parte del broker cuando ha aceptado una petición de suscripción.

Una vez establecida la conexión cliente – broker y la suscripción a los diferentes topics se ha completado con éxito, el cliente también puede publicar en los topics que le corresponden con el mensaje PUBLISH.

6.2.2 Servidor web

Para el almacenamiento y consulta de los datos recogidos por los sensores conectados al microcontrolador se ha decidido contar con un servidor MQTT.

Un servidor web es un espacio en la nube en la que un cliente puede alojar los datos que considere y obtenerlos o visitarlos en cualquier momento en el que cuente con conexión a internet.

En este caso, se ha hecho uso de un servidor web MQTT ya implementado, conocido como *MyQttHub*. Este servidor cuenta con múltiples opciones de configuración que se analizarán a continuación.

6.2.2.1 Servidor web MyQttHub

En la página principal de la web se muestran los dispositivos del dominio que se tiene asociado al usuario/contraseña que se ha identificado, en este caso, al dominio *SupervisionArdu* se tienen asociados dos dispositivos o devices. El primero de ellos se creó con la intención de tener un dispositivo administrador que siempre estuviera asociado, el segundo de ellos es el dispositivo *ESP8266 Client*, utilizado en el presente proyecto para llevar a cabo toda la comunicación a través del protocol MQTT.

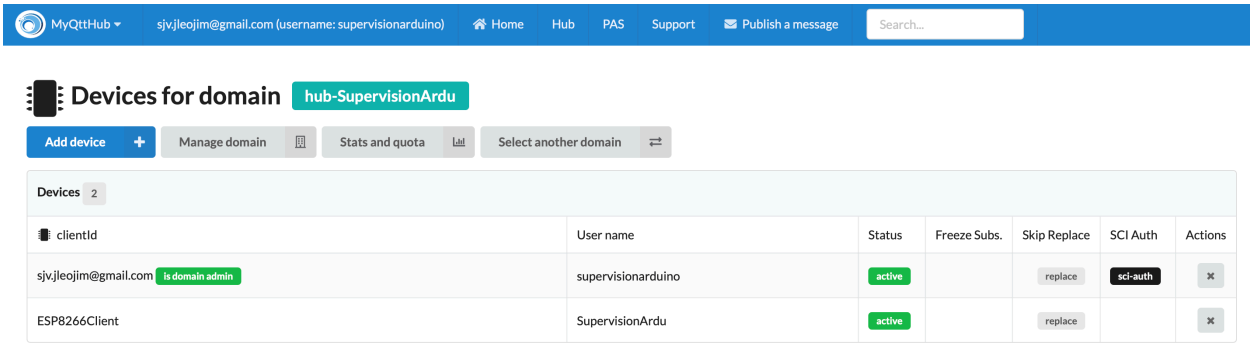


Figura 32: Página principal MyQttHub

En la esquina superior izquierda se despliega un menú a partir del cual se puede acceder a todas las prestaciones y configuraciones que ofrece el servidor.

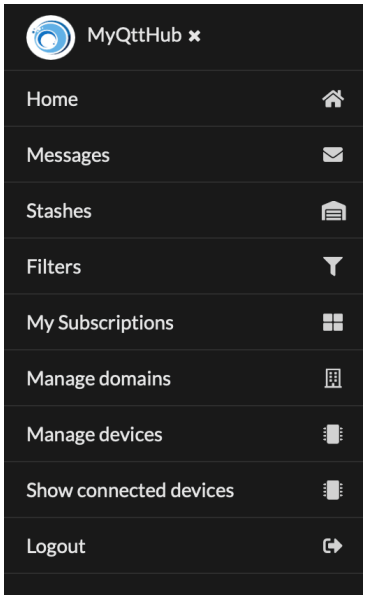


Figura 33: Menú MyQttHub

- Messages: En este apartado se muestran los mensajes publicados en todos los topics del servidor. Permite mostrar las últimas 24 horas, los últimos 7 días y el registro completo, limpiar el registro y descargarlo.

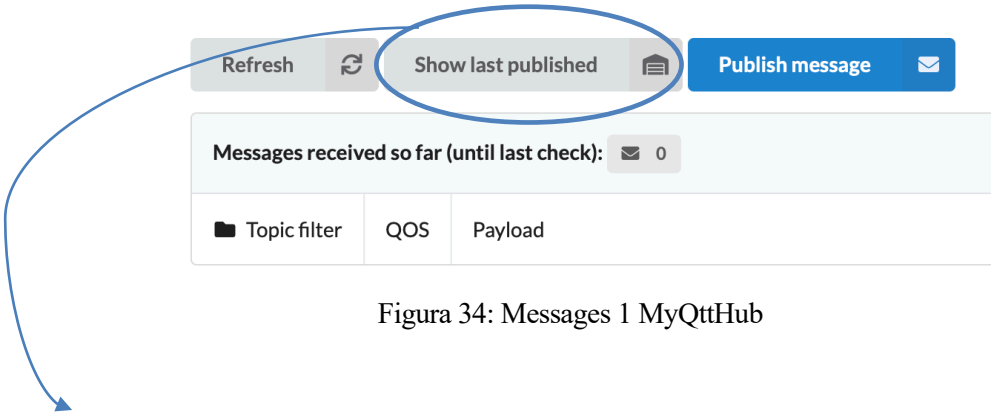


Figura 34: Messages 1 MyQttHub

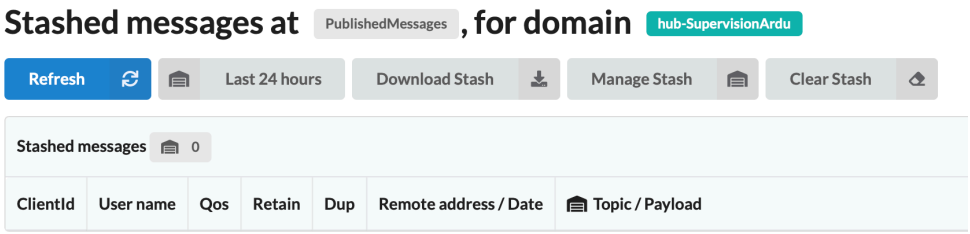


Figura 35: Messages 2 MyQttHub

- Stashes: Se muestran los mensajes publicados separados por topics, quedando en manos del usuario el topic que desea consultar. En este sistema, se cuenta con dos topics, uno referido a cambios anómalos en la temperatura de los acuarios y otro referido al nivel de agua de éstos.

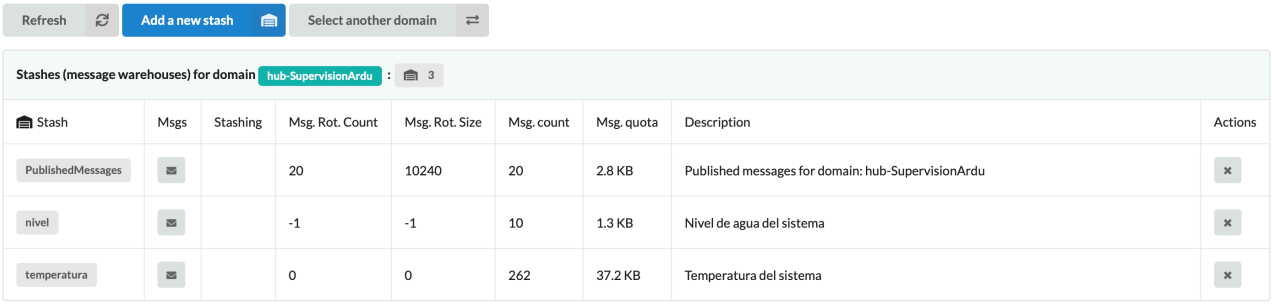


Figura 36: Stashes MyQttHub

- Filters: Para que en el apartado anterior sea posible la clasificación de mensajes según el topic al que pertenezcan, es necesario aplicar filtros. A la hora de publicar un mensaje en el servidor, el cliente debe indicar la ruta en la que quiere publicarlo, por ejemplo, si el mensaje se debe a una temperatura no deseada, el cliente debe indicar que desea publicar el mensaje en la ruta temperatura/Temperatura

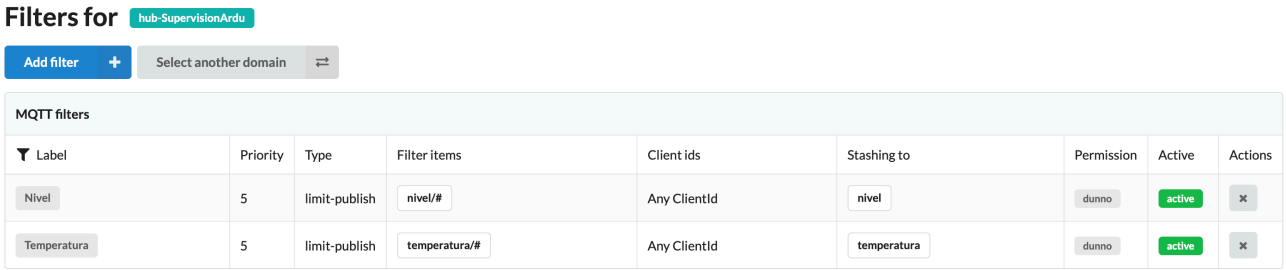


Figura 37: Filters MyQttHub

- My Subscriptions: Se muestran los topics a los que se encuentra suscrito el dispositivo.
- Manage domains: Se muestran los dominios activos en el servidor web. En este caso, el dominio activo es *hub-SupervisionArdu*.
- Manage devices: Se muestran los dispositivos vinculados al usuario.
- Show connected devices: En este apartado se muestran los dispositivos conectados al servidor.

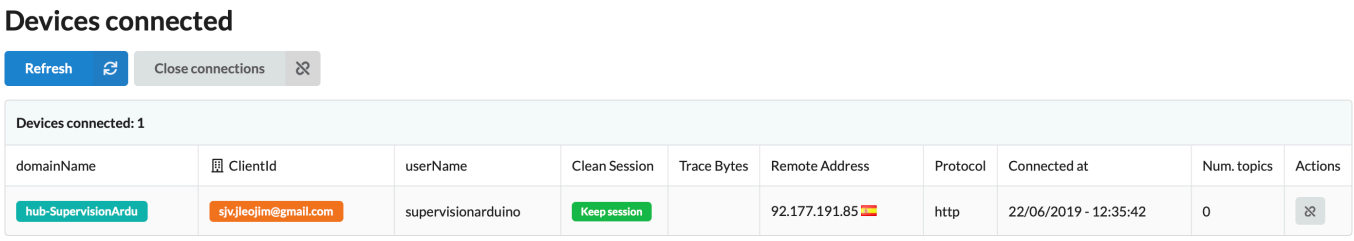


Figura 38: Dispositivos conectados MyQttHub

- Logout: Cierre de la sesión.

6.3 Programas de control

En el caso del sistema desarrollado en este proyecto, se cuenta con dos módulos principales, el microcontrolador Arduino Mega y el módulo Wi-Fi nodeMCU ESP8266, cada uno de ellos se controlará con un sketch diferente.

6.3.1 Sketch de control Arduino Mega

La recogida de datos de los sensores así como el envío de éstos al módulo WiFi se llevarán a cabo a través del microcontrolador Arduino Mega, el cual tiene cargado en su sistema un sketch que se analizará a continuación.

6.3.1.1 Librerías y declaración de variables, constantes e instancias

Para el correcto funcionamiento del sketch, se hace uso de las siguientes librerías, de las cuales se usan las funciones especificadas en cada tabla.

6.3.1.1.1 Librería SoftwareSerial.h

Encargada de proporcionar las funciones necesarias para abrir el puerto de comunicación serie y escribir en él.

Tabla 20: Funciones librería SoftwareSerial

Función	Utilidad
begin	Abrir el puerto serie
println	Escribir por párrafos en la salida del puerto serie

Dado que los sensores de temperatura utilizados en el proyecto cuentan con la tecnología 1-wire, la cual permite colocar varios sensores en un mismo bus, utilizando un solo pin de datos, se necesitará la librería de este protocol así como la específica de este sensor.

6.3.1.1.2 Librería DallasTemperature.h

A través de las funciones implementadas en esta librería es posible realizar lecturas y configuraciones de los sensores de temperatura.

Tabla 21: Funciones librería DallasTemperature

Función	Utilidad
requestTemperatures	Leer las temperaturas de los sensores
getTempCByIndex	Obtener la temperatura en ºC del sensor x

6.3.1.1.3 Librería OneWire

Librería en la que se encuentra implementado todo el protocolo 1-wire necesario para el funcionamiento de los sensores de temperatura.

6.3.1.1.4 Declaración de variables, constantes e instancias

A continuación se muestran las variables, constantes e instancias declaradas en el programa principal.

En primer lugar, se define el pin de datos al que van a estar conectados los sensores de temperatura (*pinDatosDQ*), al tratarse de sensores que utilizan el protocolo 1-wire, solo será necesario definer un pin de datos.

En segundo lugar, se declaran los objetos necesarios para el desarrollo del programa, es decir, el bus 1-wire y el objeto DallasTemperature del tipo del sensor de temperatura de nuestro sistema.

Por último, se definen las variables str (cadena de texto que va a ser enviada al puerto serie) y nivel0, nivel1 y nivel2, que son los pines donde están conectados los sensores de nivel de agua.

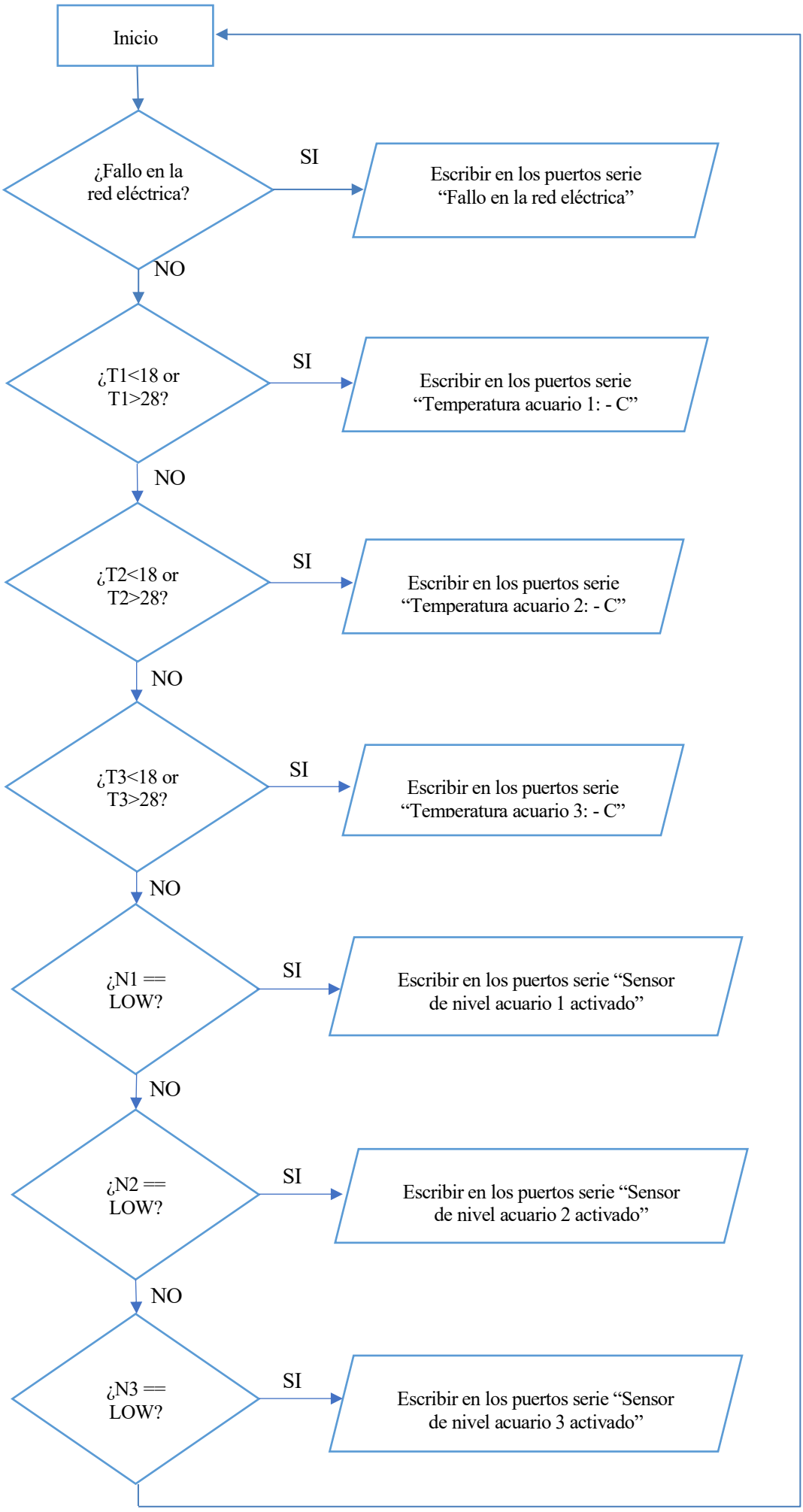
```
// Pin donde se conecta el bus 1-Wire
const int pinDatosDQ = 9;

// Objetos
OneWire oneWireObjeto(pinDatosDQ);
DallasTemperature sensorDS18B20(&oneWireObjeto);

// Variables
String str;
int nivel0 = 35;
int nivel1 = 37;
int nivel2 = 39;
```

Figura 39: Variables, constantes e instancias sketch principal

6.3.1.2 Diagrama de flujo del programa principal

En primer lugar, se inicializan tanto la comunicación serie necesaria para enviar los datos recogidos al módulo Wi-Fi como la comunicación del protocolo 1-wire necesaria para la jerarquía maestro-esclavo de los sensores de temperatura. Por otro lado, se definen los pines donde están conectados los sensores de nivel como pines de entrada.

En segundo lugar, en el bucle loop, se lleva a cabo la comprobación de los tres sensores de temperatura, si alguno de ellos no se encuentra dentro del rango de temperatura adecuado para el acuario, se enviará un mensaje de aviso al módulo Wi-Fi, escribiendo en el puerto serie el mensaje que se desea publicar en el servidor web. Tras esta comprobación, se realiza la de los tres sensores de nivel, si alguno de ellos se ha activado debido a un descenso del nivel de agua, se escribirá en el puerto serie el mensaje que se desea publicar en el servidor web.


Inicio
¿Fallo en la red eléctrica?
SI
Escribir en los puertos serie "Fallo en la red eléctrica"
NO
¿T1<18 or T1>28?
SI
Escribir en los puertos serie "Temperatura acuario 1: - C"
NO
¿T2<18 or T2>28?
SI
Escribir en los puertos serie "Temperatura acuario 2: - C"
NO
¿T3<18 or T3>28?
SI
Escribir en los puertos serie "Temperatura acuario 3: - C"
NO
¿N1 == LOW?
SI
Escribir en los puertos serie "Sensor de nivel acuario 1 activado"
NO
¿N2 == LOW?
SI
Escribir en los puertos serie "Sensor de nivel acuario 2 activado"
NO
¿N3 == LOW?
SI
Escribir en los puertos serie "Sensor de nivel acuario 3 activado"

6.3.2 Sketch de control nodeMCU ESP8266

Como se explicó en el punto 6.1, un sketch de Arduino contiene dos funciones principales, las cuales se van a analizar para el sketch de inicialización y control del modulo Wi-Fi.

6.3.2.1 Librerías y declaración de variables

Para el funcionamiento del programa, se han necesitado funciones definidas en librerías independientes, las cuales se van a detallar a continuación.

6.3.2.1.1 Librería ESP8266.h

Librería necesaria para la conexión a internet a través del módulo Wi-Fi ESP8266.

Tabla 22: Funciones librería ESP8266

Función	Utilidad
begin	Establecer conexión con la red WiFi
status	Comprobar si la conexión de ha establecido con éxito

6.3.2.1.2 Librería SoftwareSerial.h:

Encargada de proporcionar las funciones necesarias para abrir el puerto de comunicación serie y escribir en él.

Tabla 23: Funciones librería SoftwareSerial

Función	Utilidad
begin	Abrir el puerto serie
println	Escribir por párrafos en la salida del puerto serie
available	Comprobar si el puerto está disponible
read	Leer por caracteres lo escrito en el puerto serie

6.3.2.1.3 Librería PubSubClient.h:

Para poder hacer que el módulo WiFi funcione como un cliente MQTT, capaz de publicar y recibir mensajes de un topic que se encuentra en un servidor MQTT, se requiere esta librería.

Tabla 24: Funciones librería PubSubClient

Función	Utilidad
setServer	Acceder a la URL y puerto donde se aloja el servidor MQTT

setCallback	Recibir mensajes de los diferentes topics
connected	Comprobar conexión al servidor
connect	Conexión al servidor MQTT (ID, user, password)
loop	Mantener activa la conexión del cliente MQTT a la web
publish	Publicar en un topic del servidor MQTT

6.3.2.1.4 Declaración de variables, constantes e instancias

Antes de nada, se definen las variables y constantes que van a ser utilizadas a lo largo del programa.

En primer lugar, se definen las credenciales del Wi-Fi y las credenciales del servidor MQTT para poder establecer la conexión en el momento indicado en el código.

- Credenciales Wi-Fi:
  - const char* wid: Nombre de la red Wi-Fi a la que se va a conectar el módulo.
  - const char* password: Contraseña de la red Wi-Fi a la que se va a conectar el módulo.
- Credenciales servidor MQTT:
  - const char* MQTTServer: URL de la página web donde se encuentra el servidor.
  - const int MQTTPort: Puerto por el que se va a realizar la transmisión y recepción de datos.
  - const char* MQTTId: Identificador del servidor donde se va a realizar el intercambio de datos.
  - const char* MQTTUser: Usuario del servidor.
  - const char* MQTTPsswrd: Contraseña del servidor.

En segundo lugar, se definen las variables *mensaje* e *i*, además de declarar los siguientes objetos:

- ESPClient: Objeto tipo ESP8266 a través del cual se usan las funciones necesarias para establecer la conexión a internet.
- client: Objeto tipo PubSubClient usado para la conexión al servidor web MQTT y para hacer posible la transmisión de datos entre Arduino y el servidor.

```
//Credenciales del WiFi
const char* wid = "Leonidos";
const char* password = "27304323B";
//Credenciales del sevidor MQTT
const char* MQTTServer = "node02.myqtthub.com";
const int MQTTPort = 1883;
const char* MQTTId = "ESP8266Client";
const char* MQTTUser = "SupervisionArdu";
const char* MQTTPsswrd = "9oJAxIFk-SOad4I5n";

int i = 0;
char mensaje[100];

WiFiClient ESPClient;
PubSubClient client(ESPClient);
```

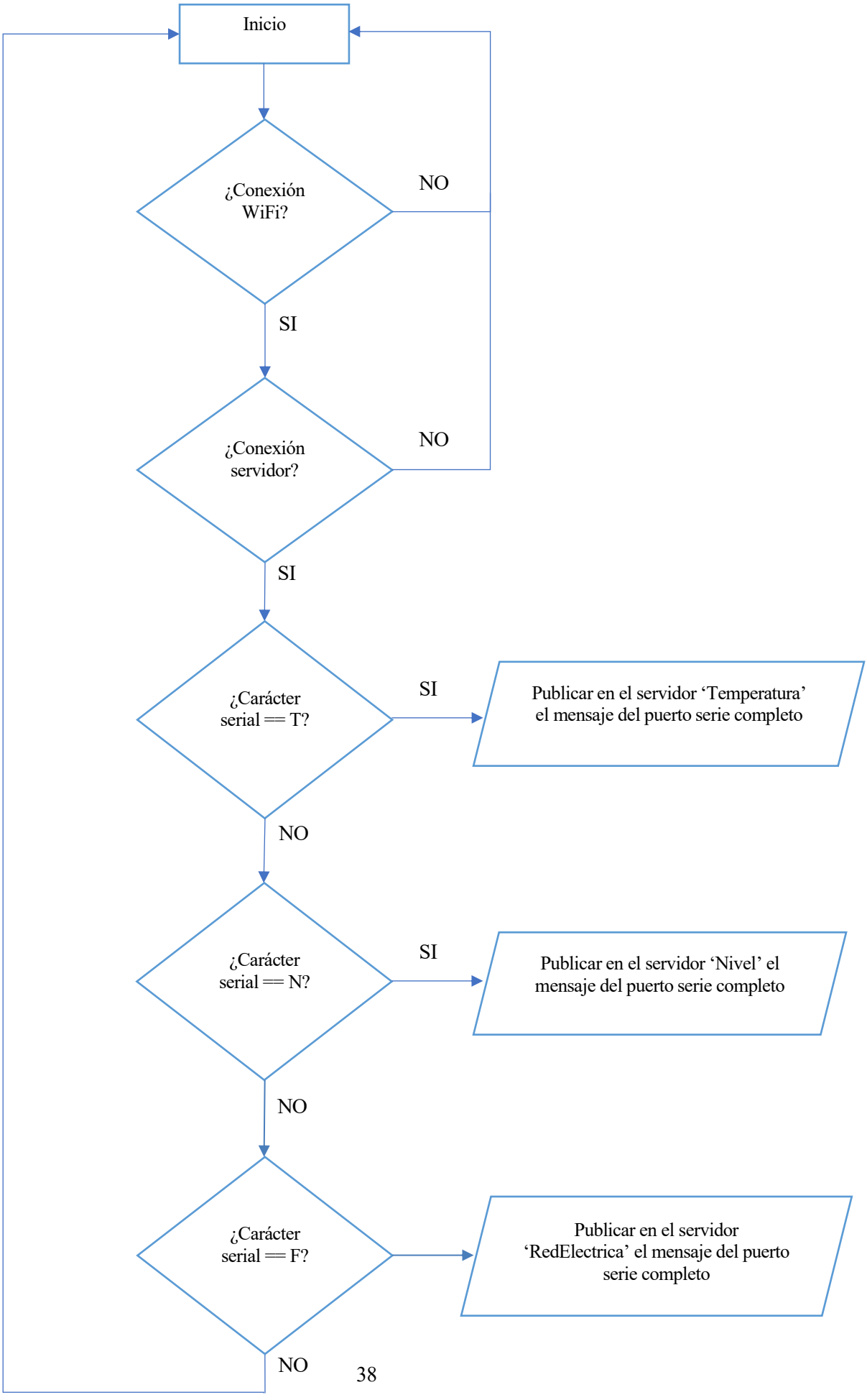
Figura 40: Variables, constantes e instancias del sketch MQTT

6.3.2.2 Diagrama de flujo del programa MQTT

En primer lugar, se inicializan el puerto serie y la comunicación Wi-Fi. Se establecen la conexión tanto a la red Wi-Fi configurada como al servidor web MQTT.

En segundo lugar, en el bucle loop, se realiza una lectura del puerto serie, guardando en la variable *mensaje* los caracteres recibidos hasta el siguiente caracter fin de línea (\n). A continuación, se diferencia entre los mensajes que comienzan por la letra T, pertenecientes a cambios en la temperatura, y los mensajes que comienzan por la letra S, pertenecientes a los cambios en el nivel de agua del acuario. Una vez identificado el tipo de mensaje, este se publica en el servidor web, dentro del topic (temperatura o nivel) que le corresponda.

Para complementar este funcionamiento, se ha incluido una función *callback*, encargada de mostrar en el puerto serie los mensajes que se publican en los topics en los que el cliente está suscrito. Esta función no se utiliza en el presente sistema, ya que no se pretende tener los puertos de comunicación abiertos en modo lectura.


Inicio
¿Conexión WiFi?
NO
SI
¿Conexión servidor?
NO
SI
¿Carácter serial == T?
SI
Publicar en el servidor 'Temperatura' el mensaje del puerto serie completo
NO
¿Carácter serial == N?
SI
Publicar en el servidor 'Nivel' el mensaje del puerto serie completo
NO
¿Carácter serial == F?
Publicar en el servidor 'RedElectrica' el mensaje del puerto serie completo
NO

7 PRUEBAS PROTOTIPO

Para concluir con el análisis del funcionamiento y comportamiento del sistema, se incluyen 3 pruebas realizadas con el prototipo, ocasionando diferentes escenarios donde más de un sensor se encuentra fuera de rango.

El montaje final del prototipo se puede observar en la siguiente imagen:

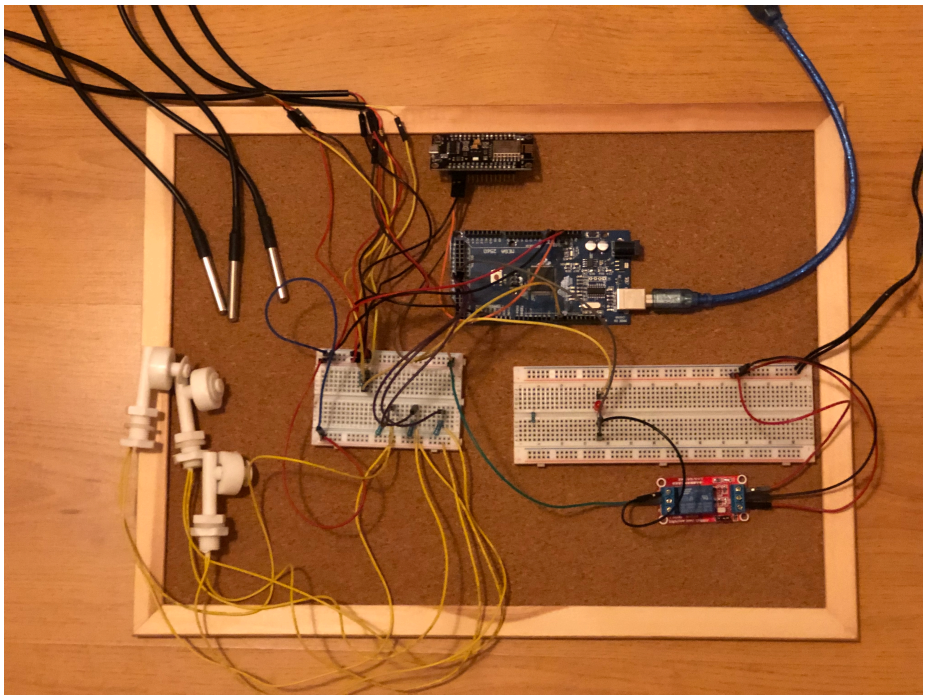


Figura 41: Montaje final del prototipo

7.1 Prueba 1. Sensores de temperatura y nivel acuario 3 activados

En esta primera prueba realizada, se provoca un descenso de la temperatura del acuario así como de su nivel.

Como se analizó en el punto 3, el acuario debe tener una temperatura comprendida en el rango entre 18 ºC y 28 ºC, si la temperatura del acuario es menor que 18 ºC o mayor que 28 ºC, el sistema enviará un mensaje de aviso al servidor web MQTT, en particular, publicará este mensaje en el topic *Temperatura*, aunque también se podrá encontrar en el apartado *PublishedMessages*.

Por otro lado, cuando el nivel de agua se encuentra por debajo del nivel deseado, haciendo que el sensor de nivel se desactive, también se llevará a cabo el proceso de envío del aviso, en este caso se publicará en el topic *Nivel*, además de publicarse en *PublishedMessages*.

Este mensaje publicado en el servidor se inserta en una tabla que cuenta con las siguientes columnas:

- ClientId: Identificador del cliente que está publicando el mensaje en el servidor.
- User name: Nombre del usuario registrado al que pertenece el cliente.
- QoS: Nivel de calidad de servicio.
- Retain: Conservación de mensajes.

- Dup: Duplicación de mensajes
- Remote address/Date: IP desde la que se ha enviado el mensaje y fecha/hora del momento de la publicación.
- Topic/Payload: Topic al que pertenece el mensaje publicado y texto del mensaje que se ha publicado.
- Actions: El servidor da dos opciones, volver a enviar el mensaje o borrarlo del servidor.

En primer lugar, se vacía el acuario número 3, haciendo que se active el sensor de nivel de este acuario. A continuación, se llena de agua fría sin llegar al nivel deseado, por lo que además de la alarma del sensor de nivel ya activada, se activa también el sensor de temperatura, indicando que ésta es de 12.75 ºC (*Figura 43*).

Para que el sistema salga de este estado, se termina de llenar al acuario 3 con agua templada para alcanzar la temperatura adecuada.

Las temperaturas de los acuarios 1 y 2 también se encuentran por encima del rango y ha llegado el aviso al servidor, pero este caso se tratará en el siguiente apartado.

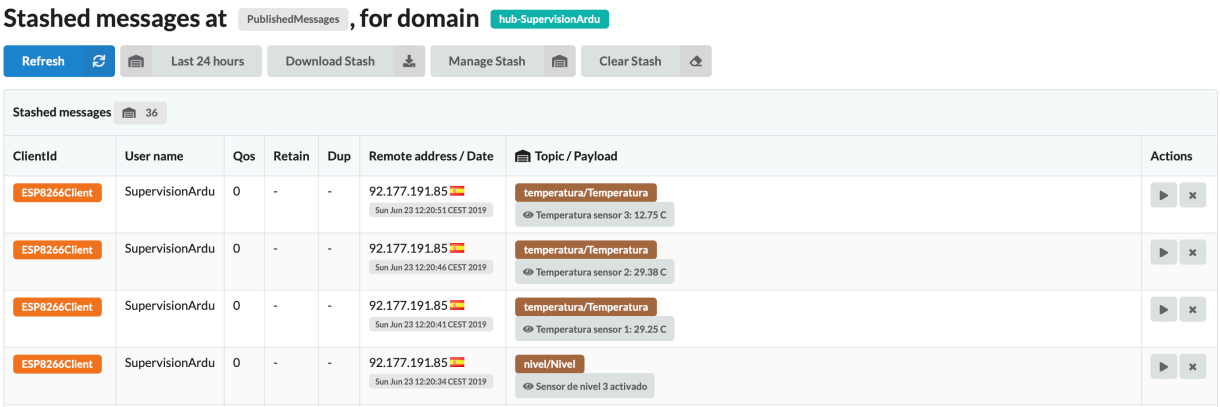


Figura 42: Prueba 1. Sensores de temperatura y nivel acuario 3 activados

7.2 Prueba 2. Sensores de temperatura acuarios 1 y 2 activados

En la segunda prueba, se provoca el aumento de la temperatura en los acuarios 1 y 2.

Como se puede observar en la figura, los acuarios 1 y 2 tienen una temperatura de 29.19 ºC y 29.31 ºC respectivamente, al estar fuera de rango, el sistema ha enviado al servidor el mensaje de aviso, siendo publicado en la fecha y hora indicadas en el mensaje.

Posteriormente, se ha rellenado el acuario con agua fría para bajar la temperatura y el mensaje ha dejado de enviarse al servidor, ya que se ha desactivado la alarma al volver a estar la temperatura del agua dentro del rango permitido.

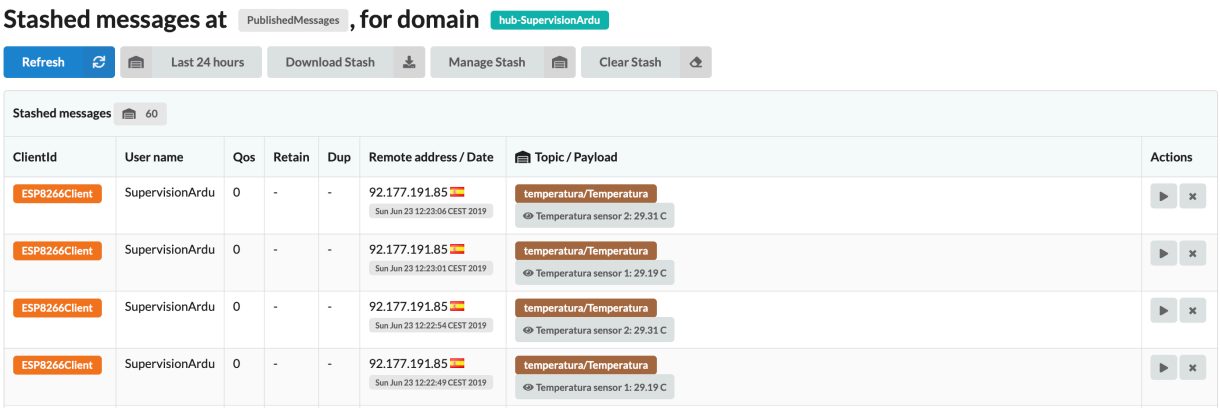


Figura 43: Prueba 2. Sensores de temperatura acuarios 1 y 2 activados

7.3 Prueba 3. Sensores de nivel acuario 1 y 3 activados

En la tercera, se procede al vaciado de los acuarios 1 y 3, para provocar que las alarmas que avisan de que los niveles son inferiores al deseado de activen.

Como se observa en la *Figura 45*, mientras no se reponga el agua perdida en el acuario, el sistema continuará publicando en el topic *Nivel* del servidor el mensaje de que el sensor de nivel se encuentra activado. Una vez que el nivel de agua sube al deseado, el sistema desactivará la alarma y dejará de enviar el mensaje al servidor.

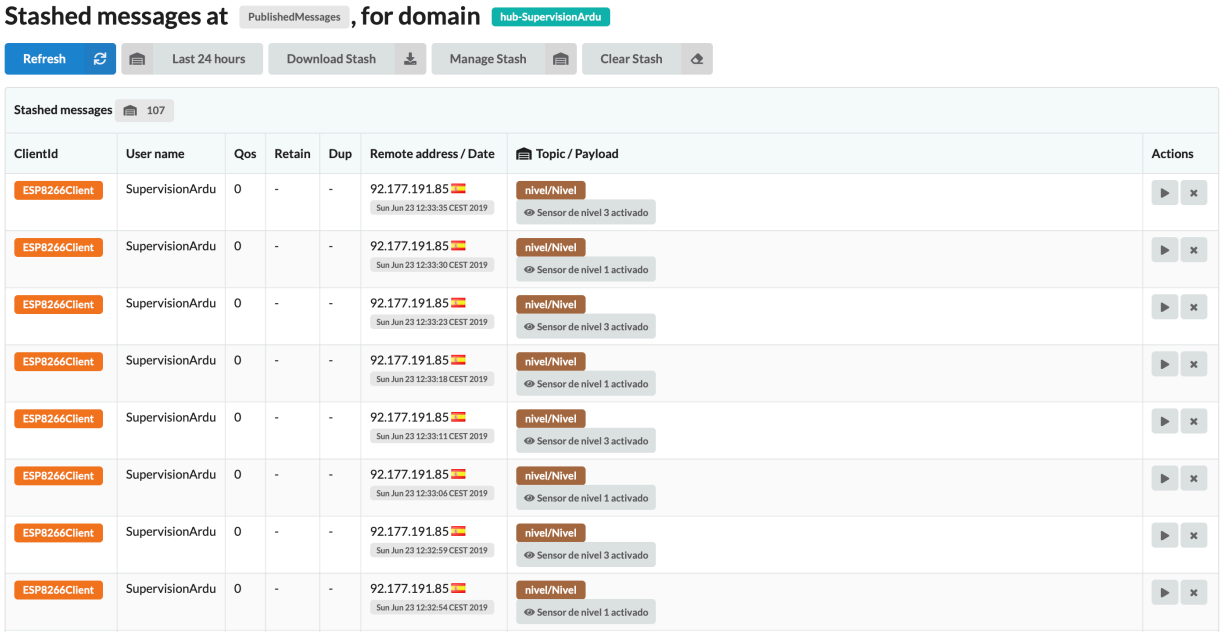


Figura 44: Prueba 3. Sensores de nivel acuarios 1 y 3 activados

7.4 Prueba 4. Fallo en la red eléctrica

Como se explica en el apartado 4, el sistema cuenta con un relé a partir del cual se obtiene la información necesaria para hacer llegar al usuario el aviso de que la red eléctrica ha fallado.

Para llevar a cabo esta última prueba, partiendo del sistema conectado y recogiendo información de manera normal, se procede a quitar la alimentación. Como se puede observar en la *Figura 45*, a las 13:58:17 se produce el fallo en la red, y el sistema añade a la información de error de los sensores de temperatura, el error en la alimentación.

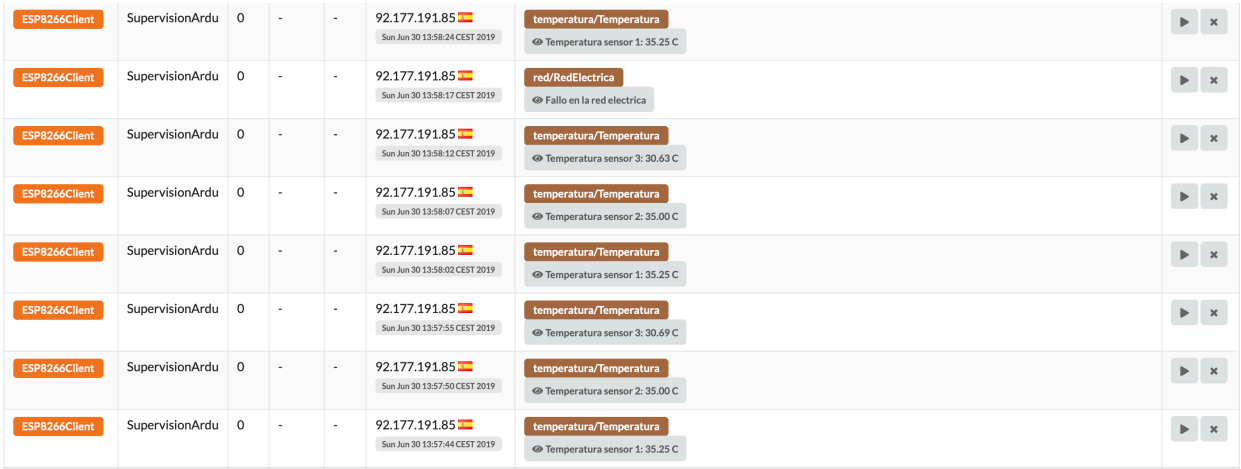


Figura 45: Prueba 4. Fallo en la red eléctrica

8 CONCLUSIONES

Tras el análisis del sistema completo, tanto en hardware como en software, se puede concluir que se ha logrado diseñar un sistema de supervisión de parámetros de un acuario con conexión a un servidor web MQTT con éxito.

En el punto 2 se mostraban algunos sistemas ya diseñados con un precio muy alto, en este proyecto se ha conseguido mejorar las prestaciones de estos sistemas, añadiendo la posibilidad de consulta en un servidor web, bajando el precio cantidades notables. Para mostrarlo de manera más visual, en la siguiente tabla se observan el precio de los componentes elegidos así como el precio final del sistema, sin contar con el aumento que la mano de obra supone.

Tabla 25: Presupuesto sistema completo

Componente	Precio	Componente	Precio
Arduino MEGA	15.30 €	Módulo Wi-Fi	7.69 €
Sensor DS80B20 (x3)	3.49 €	Kit Arduino	2.54 €
Sensor de nivel (x3)	2.77 €	120 cables Arduino	7.09 €
Relé	3.5 €		

Precio total del sistema de supervisión: **54.9 €**

A nivel personal, el desarrollo del proyecto me ha permitido conocer el mundo de Arduino, el cual siempre me ha llamado la atención y no había tenido oportunidad de aprender. Por otro lado, he aprendido el funcionamiento de los servidores web y lo complejo que puede llegar a ser crear uno. En general, el sentimiento después de finalizar este proyecto es extraordinario, me ha proporcionado aprendizaje no solo tecnológico y de comunicaciones, también de organización y redacción de proyectos.

9 MEJORAS FUTURAS

Sobretodo en la tecnología, los sistemas avanzan y se actualizan conforme cambian las necesidades de los usuarios del producto. Este sistema no iba a ser menos, hay mejoras que se pueden aplicar en futuras versiones del proyecto.

En cuanto a hardware, relacionado con los sensores, se pueden añadir tantos como parámetros se quieran medir. Al ser el microcontrolador del sistema un Arduino MEGA que cuenta con las mejores prestaciones en cuanto a pines y memoria, en un futuro pueden añadirse los sensores e incluso actuadores que el usuario desee.

En lo relacionado con software, el desarrollo de una aplicación propia en la que se registren los mensajes de alerta como lo hacen en el servidor web sería un punto positivo, tanto económicamente, ya que aumentaría el precio del sistema por tener más exclusividad, como tecnológicamente la empresa que lo desarrolle, al aprender nuevos lenguajes de programación en Android/iOS.

Por otro lado, se podría añadir al sistema, el control de los parámetros que se supervisan, añadiendo actuadores.

10 ANEXOS

10.1 Código Arduino de los programas utilizados

10.1.1 Programa principal

```
#include <SoftwareSerial.h>
#include <OneWire.h>
#include <DallasTemperature.h>

// Pin donde se conecta el bus 1-Wire
const int pinDatosDQ = 9;

// Objetos
OneWire oneWireObjeto(pinDatosDQ);
DallasTemperature sensorDS18B20(&oneWireObjeto);

// Variables
String str;
int nivel0 = 35;
int nivel1 = 37;
int nivel2 = 39;
int relay = 41;

void setup() {
    // Iniciamos la comunicacion serie
    Serial.begin(115200);
    Serial1.begin(115200);

    // Iniciamos el bus 1-Wire
    sensorDS18B20.begin();
    delay(2000);

    //Se definen los pines de los sensores de nivel como entradas
    pinMode(nivel0, INPUT);
    pinMode(nivel1, INPUT);
    pinMode(nivel2, INPUT);
    pinMode(relay, INPUT);
}


void loop() {
    // Mandamos comandos para toma de temperatura a los sensores
    delay(1000);
    sensorDS18B20.requestTemperatures();

    if (sensorDS18B20.getTempCByIndex(0) < 18 or sensorDS18B20.getTempCByIndex(0) > 28) {
      str = String("Temperatura sensor 1: ")+String(sensorDS18B20.getTempCByIndex(0))+String(" C");
      Serial.println(str);
      Serial1.println(str);
      delay(5000);
    }

    if (sensorDS18B20.getTempCByIndex(1) < 18 or sensorDS18B20.getTempCByIndex(1) > 28) {
      str = String("Temperatura sensor 2: ")+String(sensorDS18B20.getTempCByIndex(1))+String(" C");
      Serial.println(str);
      Serial1.println(str);
      delay(5000);
    }

    if (sensorDS18B20.getTempCByIndex(2) < 18 or sensorDS18B20.getTempCByIndex(2) > 28) {
      str = String("Temperatura sensor 3: ")+String(sensorDS18B20.getTempCByIndex(2))+String(" C");
      Serial.println(str);
      Serial1.println(str);
      delay(5000);
    }

    if (digitalRead(nivel0) == LOW) {
     Serial.println("Sensor de nivel 1 activado");
     Serial1.println("Sensor de nivel 1 activado");
     delay(5000);
    }
```

```
  if (digitalRead(nivel1) == LOW) {
   Serial.println("Sensor de nivel 2 activado");
   Serial1.println("Sensor de nivel 2 activado");
   delay(5000);
  }

  if (digitalRead(nivel2) == LOW) {
   Serial.println("Sensor de nivel 3 activado");
   Serial1.println("Sensor de nivel 3 activado");
   delay(5000);
  }

  if (digitalRead(relay) == LOW) {
   Serial.println("Fallo en la red electrica");
   Serial1.println("Fallo en la red electrica");
   delay(5000);
  }
}
```

10.1.2 Programa MQTT

```
#include <ESP8266WiFi.h>
#include <SoftwareSerial.h>
#include <PubSubClient.h>

//Credenciales del WiFi
const char* wid = "Leonidos";
const char* password = "27304323B";

//Credenciales del sevidor MQTT
const char* MQTTServer = "node02.myqtthub.com";
const int MQTTPort = 1883;
const char* MQTTId = "ESP8266Client";
const char* MQTTUser = "SupervisionArdu";
const char* MQTTPsswrd = "9oJAxIFk-SOad4I5n";

// Objetos
WiFiClient ESPClient;
PubSubClient client(ESPClient);

// Variables
int i = 0;
char mensaje[100];

void setup() {
    // Iniciamos la comunicación serie
    Serial.begin(115200);

    while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port only
    }

    WiFi.begin(wid,password);

    while (WiFi.status() != WL_CONNECTED){
      delay(500);
    }


    Serial.println("Conectado a la red WiFi");

    client.setServer(MQTTServer, MQTTPort);
    client.setCallback(callback);

    while(!client.connected()){
      if (client.connect(MQTTId, MQTTUser, MQTTPsswrd)){
        Serial.println("Conectado al servidor MQTT");
      }
      else {
        Serial.println("Conexión al servidor MQTT fallida");
        delay(2500);
      }
    }
}

void loop() {
  client.loop();
  if (Serial.available() and client.connected()) {
    while (Serial.available()) {
      char character = Serial.read();
      char first_char;
      if (character != '\n') {
        if (i == 0){
          first_char = character;
        }
        mensaje[i] = character;
        i++;
      } else {
          mensaje[i] = '\0';
          Serial.println((const char*)mensaje);
          i = 0;
          if (first_char == 'T') {
            Serial.println("Publica alarma temperatura");
            client.publish("temperatura/Temperatura", (const char*)mensaje);
```

```
          } else if (first_char == 'S'){
            Serial.println("Publica alarma sensor nivel");
            client.publish("nivel/Nivel", (const char*)mensaje);
          } else if (first_char == 'F'){
            Serial.println("Publica alarma fallo red electrica");
            client.publish("red/RedElectrica", (const char*)mensaje);
          }
        }
      }
    }
}

void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Se encuentra en el topic: ");
  Serial.println(topic);

  Serial.print("Mensaje: ");
  for (int i = 0; i < length; i++) {
    Serial.print((char)payload[i]);
  }
  Serial.print("\n");
}
```

REFERENCIAS

[1] Unidades de la temperatura

http://www.pce-iberica.es/medidor-detalles-tecnicos/unidades-temperatura.htm

[2] Temperatura de un acuario

http://www.mascotapedia.com/como-puedo-mantener-una-temperatura-adecuada-en-mi-acuario/

[3] pH

https://es.wikipedia.org/wiki/Hidr%C3%B3geno

http://www.elacuarista.com/secciones/quimica1_agua.htm

[4] Conductividad eléctrica

http://www.smart-fertilizer.com/es/articles/electrical-conductivity

http://www.infoagro.com/instrumentos_medida/doc_conductividad_electrica.asp?k=53

[5] Microcontrolador

https://hetpro-store.com/TUTORIALES/microcontrolador/

[6] Qué es Arduino

http://arduino.cl/que-es-arduino/

[7] Arduino MEGA

https://comohacer.eu/analisis-comparativo-placas-arduino-oficiales-compatibles/#Arduino_Mega

[8] Tipos de Arduino

https://hacedores.com/cuantos-tipos-diferentes-de-arduino-hay/

[9] Wi-Fi en Arduino

https://aprendiendoarduino.wordpress.com/2016/11/12/wifi-en-arduino/

[10] Alimentación de Arduino

https://www.luisllamas.es/alimentar-arduino-baterias/

[11] Programación de Arduino

https://www.prometec.net/condicionales-botones/

http://manueldelgadocrespo.blogspot.com/p/description-text-strings-can-be.html

[12] Aquatronica y Reef Angel Aquarium Controller

https://www.aquatronica.com/resources/downloads/ita/ALTRO/CATALOGHI/CATALOGO%20GENERAL%20V02_ES.pdf

http://www.reefangel.com

[13] Qué es MQTT

https://geekytheory.com/que-es-mqtt