



**Universidad**  
Zaragoza

## Trabajo Fin de Grado

Solución IoT para gestión de red de abastecimiento  
hídrico.

IoT solution for water supply network management.

Autor/es

**Alejandro Cencerrado Tello**

Director/es

**José Félix Serna Fortea**

Escuela Universitaria Politécnica de Teruel

2022



Escuela Universitaria  
Politécnica - Teruel  
Universidad Zaragoza

## DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD TRABAJO FIN DE GRADO Y FIN DE MASTER

A-PRCPT-002-07 V2

D./D<sup>a</sup> Alejandro Cencerrado Tello

con DNI/NIE: 73107857B en aplicación de lo dispuesto en el artículo 14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declara que el Trabajo Fin de Grado en Ingeniería Informática

Con Título: Solución IoT para la gestión de una red de abastecimiento hídrico.

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Teruel, a 27 de Noviembre de 2022

Fdo.: Alejandro Cencerrado Tello



# “Solución IoT para gestión de red de abastecimiento hídrico”

## Resumen

En el presente trabajo se ha desarrollado el diseño e implementación de un sistema informático IoT (Internet of Things), dedicado a la gestión de una red de abastecimiento hídrico, alimentada por energía proveniente de paneles solares y con nodos situados en lugares geográficamente aislados.

Estará formado por tres componentes: una aplicación web multiusuario que mostrará los datos relevantes de la red y permitirá su gestión; un prototipo físico a pequeña escala y un sistema de comunicaciones entre los elementos utilizando la especificación LoRa.

Palabras Clave: IoT, MQTT, LoRa, panel fotovoltaico, abastecimiento hídrico.

# “IoT solution for water supply network management”

## ABSTRACT

In the present project, the design and implementation of an IoT (Internet of Things) computer system has been developed, dedicated to the management of a water supply network, powered by energy from solar panels and with nodes located in geographically isolated places.

It will be made up of three components: a multi-user web application that will display the relevant network data and allow its management; a small-scale physical prototype and communication system between the elements using the LoRa specification.

Keywords: IoT, MQTT, LoRa, Solar Pannel, water supply.

## TABLA DE CONTENIDO

<b>LISTADO DE FIGURAS</b> .....	<b>C</b>
<b>LISTADO DE TABLAS</b> .....	<b>C</b>
<b>1. INTRODUCCIÓN</b> .....	<b>1</b>
1.1 MOTIVACIÓN .....	1
1.1.2 ¿Por qué LoRa? .....	2
1.2 OBJETIVOS .....	2
1.3 PROBLEMAS ABORDADOS .....	2
1.4 RESULTADOS PREVISTOS .....	3
1.5 VISIÓN GENERAL DEL DOCUMENTO .....	3
1.6 PROPÓSITO .....	3
1.7 ÁMBITO DEL SISTEMA .....	4
1.8 DEFINICIONES Y ABREVIATURAS .....	4
1.8.1 Definiciones .....	4
1.8.2 Abreviaturas .....	8
<b>2. ESPECIFICACIÓN DE REQUISITOS</b> .....	<b>8</b>
2.1 INTRODUCCIÓN .....	8
2.2 DESCRIPCIÓN GENERAL .....	8
2.2.1 Perspectiva y funciones del producto .....	8
2.2.3 Características de los usuarios .....	9
2.2.4 Casos de Uso .....	10
2.2.5 Restricciones .....	12
2.3 REQUISITOS .....	12
2.3.1 Requisitos de Interfaces Externas .....	12
2.3.2 Requisitos Funcionales .....	13
2.3.3 Requisitos De Rendimiento .....	17
2.3.4 Otros Requisitos .....	17
<b>3. PLANIFICACIÓN FINAL Y ANÁLISIS DE ESFUERZO</b> .....	<b>17</b>
3.1 DIAGRAMA DE GANTT .....	17
3.2 ANÁLISIS DE ESFUERZO .....	19
<b>4. DISEÑO</b> .....	<b>19</b>
4.1 DISEÑO ARQUITECTÓNICO .....	20
4.1.1 Elementos .....	20
4.1.2 Comunicaciones .....	23
4.2 DISEÑO SOFTWARE .....	28
4.2.1 Diagramas de Clases .....	28
4.2.4 Base de Datos .....	31
4.3 DISEÑO HARDWARE .....	31
4.3.1 Nodo Emisor .....	31
4.3.2 Nodo Puente .....	37
<b>5. IMPLEMENTACIÓN</b> .....	<b>37</b>

5.1 APLICACIÓN WEB “HYDRO MANAGER” .....	37
<b>6. PROTOTIPO - DEMOSTRADOR .....</b>	<b>41</b>
6.1 PRUEBAS DE COBERTURA LORA.....	41
6.2 PROTOTIPO NODO EMISOR .....	42
6.2.1 <i>Instalación de Energía Fotovoltaica:</i> .....	43
6.2.2 <i>Sensores:</i> .....	43
6.2.3 <i>Depósitos</i> .....	44
6.2.4 <i>Programación desarrollada</i> .....	44
6.3 PROTOTIPO NODO PUENTE .....	46
6.3.1 <i>Programación Desarrollada</i> .....	46
6.4 BROKER MQTT .....	48
<b>7. CONCLUSIONES .....</b>	<b>49</b>
<b>8. BIBLIOGRAFÍA .....</b>	<b>50</b>

## LISTADO DE FIGURAS

ILUSTRACIÓN 1: EJEMPLO DE BOMBEO SOLAR .....	1
ILUSTRACIÓN 2: DIAGRAMA DE CASOS DE USO. ....	10
ILUSTRACIÓN 3: DIAGRAMA GANTT MINIATURA .....	19
ILUSTRACIÓN 4: DISEÑO ARQUITECTÓNICO. ....	20
ILUSTRACIÓN 5: DIAGRAMA DE ORGANIZACIÓN DE TOPICS. ....	23
ILUSTRACIÓN 6: MINIATURA VISTA GENERAL DC. ....	28
ILUSTRACIÓN 7: DIAGRAMA CLASES SKETCH PUENTE. ....	30
ILUSTRACIÓN 8: DIAGRAMA CLASES SKETCH EMISOR. ....	30
ILUSTRACIÓN 9: DISEÑO RELACIONAL.....	31
ILUSTRACIÓN 10: PÁGINA DE LOGIN. ....	37
ILUSTRACIÓN 11: FORMULARIOS DE REGISTRO Y DE CONTRASEÑA OLVIDADA.....	38
ILUSTRACIÓN 12: PÁGINA DE CONTACTO. ....	38
ILUSTRACIÓN 13: PANEL DE CONTROL .....	39
ILUSTRACIÓN 14: MUESTRA DE ESTADO DE NODO Y GRÁFICAS.....	39
ILUSTRACIÓN 15: ESTADO ACTUAL DE DEPÓSITO.....	39
ILUSTRACIÓN 16: BUSCADOR HISTÓRICO.....	40
ILUSTRACIÓN 17: GRÁFICAS HISTÓRICO .....	40
ILUSTRACIÓN 18: TABLAS HISTÓRICO .....	40
ILUSTRACIÓN 19: DIAGRAMA DEL CABLEADO. ....	42
ILUSTRACIÓN 20: PROTOTIPO EMISOR.....	43
ILUSTRACIÓN 21: DISPENSADORES Y SENSOR VOLUMEN TAPA. ....	44
ILUSTRACIÓN 22: WEMOS TTGO ESP32 PUENTE. ....	46
ILUSTRACIÓN 23: RASPBERRY PI 3 B+ NODO PUENTE. ....	48

## LISTADO DE TABLAS

TABLA 1: CARACTERÍSTICAS DE USUARIO ROL "PROPIETARIO" .....	9
TABLA 2: CARACTERÍSTICAS DE USUARIO ROL "ADMINISTRADOR" .....	10
TABLA 3: GANTT RESUMEN DEL PROYECTO .....	17
TABLA 4: PLANIFICACIÓN - RESUMEN DE TAREAS .....	18
TABLA 5: ANÁLISIS DE ESFUERZO .....	19

## 1. INTRODUCCIÓN

El uso de IoT (Internet of Things o Internet de las Cosas) está transformando el mundo de la tecnología por completo y a una gran velocidad.

Este concepto hace referencia a la interconexión digital de dispositivos y objetos a través de una red (bien sea una privada o Internet, la red de redes). Esta interconexión y el flujo de datos entre los dispositivos hace que éstos sean capaces de compartir información y realizar acciones de forma autónoma y automatizada.

Respecto al tipo de objetos o dispositivos podría ser cualquiera, desde sensores y dispositivos mecánicos hasta objetos cotidianos como puede ser un frigorífico, el calzado o la ropa.

Internet ha evolucionado rápidamente y esto ha permitido que IoT sea ya una realidad y no sólo una visión de futuro. La fama de esta tecnología radica principalmente en todas las aplicaciones y posibilidades que nos proporciona tanto para mejorar tanto la vida cotidiana de las personas como los entornos empresariales, donde ya se está implantando desde hace algún tiempo.

Las aplicaciones son casi infinitas, a continuación, se van a describir algunos ejemplos para dar visibilidad de alguna de ellas: Desde sensores para gestionar el tráfico, la energía o la contaminación, espejos inteligentes que detectan la prenda y nos muestran tallas disponibles y colores, hasta máquinas que fabrican componentes en la industria, drones que ayudan en agricultura y lucha contra incendios o viñedos conectados que permiten establecer criterios para el cuidado de la viña, seleccionar el tiempo óptimo para la recogida y elaboración del vino.

[1] [2] [3]

### 1.1 MOTIVACIÓN

El auge del IoT y la zona geográfica en la que nos encontramos (Teruel) hace que existan grandes oportunidades de desarrollo de las técnicas y tecnologías utilizadas en el medio rural.

Analizando el sector agropecuario podemos encontrar gran cantidad de sistemas alimentados mediante energía solar y que almacenan la energía generada en baterías. Otra de las características de este sector es que se compone de instalaciones situadas en zonas aisladas, pueblos, o zonas de campo en las que no es posible conseguir cobertura de telefonía móvil y mucho menos conexión a internet. Asimismo, la gran mayoría de las instalaciones agropecuarias utilizan una red de depósitos distribuidos por las instalaciones, empleados para tareas como abastecer los bebederos de una determinada nave o el riego de diferentes huertos.

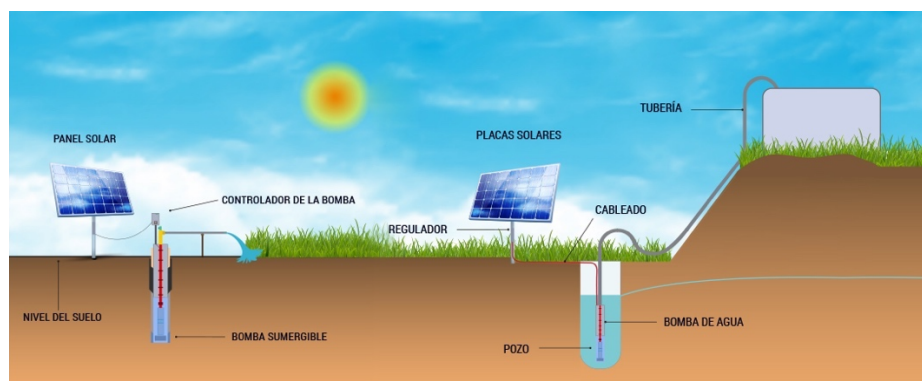


Ilustración 1: Ejemplo de bombeo solar

El problema de las soluciones utilizadas actualmente es el uso de sistemas de comunicaciones con alto coste de instalación y consumo de energía eléctrica.

El principal objetivo de mi proyecto es crear un sistema informático innovador que se pueda implantar en un entorno real del medio rural, por ello se decide crear un sistema que resuelva todas las necesidades nombradas anteriormente y que utilizará un sistema de comunicaciones con tecnología LoRa.

### 1.1.2 ¿Por qué LoRa?

Entre las principales ventajas del uso de LoRa sobre otras tecnologías cabría destacar:

- Alta tolerancia a las interferencias.
- Alta sensibilidad para recibir datos (-168dB).
- Bajo Consumo (hasta 10 años con una batería).
- Largo alcance, de 10 a 20km, esto unido al bajo consumo la coloca al frente de otras tecnologías que requieren una cantidad de energía mucho mayor para alcanzar dichas distancias.

[4]

## 1.2 OBJETIVOS

A continuación, se muestran los objetivos definidos.

### Objetivo Principal

El objetivo principal de este proyecto es la demostración de todos los conocimientos adquiridos durante el grado de Ingeniería informática, y la aplicación de éstos en la realización del análisis, diseño y desarrollo de un sistema informático IoT (Internet of Things), que permita gestionar el funcionamiento de una red de abastecimiento hídrica utilizada en instalaciones agropecuarias, alimentada por energía proveniente de paneles solares en una instalación geográficamente aislada y comunicado por redes de largo alcance y bajo consumo.

### Objetivos Específicos

Los objetivos específicos serán los siguientes:

- La gestión de su alimentación eléctrica mediante energía solar.
- Creación de un sistema de comunicaciones óptimo, que permita su integración con un back-end remoto.
- La construcción de una maqueta que permita hacer una validación técnica de todos los aspectos anteriores.

## 1.3 PROBLEMAS ABORDADOS

En este proyecto trabajaré por primera vez con microcontroladores, así como con el entorno de desarrollo Arduino o tecnologías como LoRa. Por otra parte, afianzaré conocimientos adquiridos a lo largo del grado como son el desarrollo con lenguajes como Java, HTML, CSS, Javascript y el uso de Bases de Datos.

Algunas de tecnologías a utilizar para el desarrollo de este proyecto serán las siguientes: J2EE, MQTT, I2C, LoRa, microcontrolador ESP32 y panel fotovoltaico.

## 1.4 RESULTADOS PREVISTOS

Los resultados que se esperan conseguir son los siguientes:

- Mostrar el volumen existente en cada uno de los depósitos de agua que compongan la red.
- Activar de manera automática o manual el llenado de los depósitos que lo requieran.
- Mostrar el histórico de la energía eléctrica generada por los paneles solares, así como la entrada de energía actualizada en cada momento y una comparación con la intensidad de luz solar.
- Facilitar información del consumo eléctrico actual.
- Visualización del estado de las baterías.
- Muestra del histórico e información actual de temperatura y presión en la localización de cada uno de los depósitos.
- Establecer una comunicación con el backend utilizando la especificación LoRa.
- Construcción de una maqueta, a modo de demostrador tecnológico.

## 1.5 VISIÓN GENERAL DEL DOCUMENTO

El presente documento se compone de los apartados descritos a continuación:

- **Especificación de Requisitos:** este apartado contiene todo lo referente a la fase de análisis y estudio de las acciones que debe permitir realizar el sistema para satisfacer las necesidades del usuario, para ello se ha seguido el estándar *IEEE 830-1998 - IEEE Recommended Practice for Software Requirements Specifications*.
- **Planificación final y análisis de esfuerzo:** apartado en el que se encuentra un diagrama de Gantt con la planificación final del proyecto y esfuerzo asociado a cada grupo de tareas.
- **Diseño:** apartado que contiene el diseño arquitectónico de los elementos que forman el sistema, un diseño detallado del software, así como el diseño de como se almacenarán los datos necesarios en la base de datos y como se realizan las comunicaciones.
- **Implementación:** Capítulo en el que se narran algunos detalles del desarrollo de los diferentes componentes Software y Hardware del sistema y sus comunicaciones.
- **Prototipo - demostrador:** apartado destinado a todo lo referente a la fase de creación de un prototipo a pequeña escala que nos permita realizar las pruebas pertinentes en el sistema.
- **Conclusiones:** apartado en el que se desarrollan las conclusiones sobre el proyecto y los resultados.
- **Bibliografía:** referencias bibliográficas utilizadas a lo largo del desarrollo del proyecto.
- **Anexos:** contendrán toda la información complementaria al documento principal que no ha tenido cabida en él al exceder el límite de páginas establecido en 50.

## 1.6 PROPÓSITO

El producto a obtener será “**Hydro Manager**”, un sistema informático cuyo propósito será el de encargarse de la gestión y control de una red de abastecimiento hídrico utilizada en una instalación del sector agropecuario.

## 1.7 ÁMBITO DEL SISTEMA

Como ya se ha comentado en apartados anteriores el sistema deberá estar preparado para instalarse en un entorno rural o agrícola.

Sus características son:

- Zonas aisladas.
- Difícil, costoso o incluso imposible acceso a red eléctrica y cobertura móvil.
- Medio agresivo: altas temperaturas, humedad, polvo, fuertes vientos...
- Uso de depósitos de agua.
- Analizando el sector agropecuario podemos encontrar gran cantidad de sistemas alimentados mediante energía solar y que almacenan la energía generada en baterías.

Todas estas características nos llevan a escoger LoRa como sistema de comunicaciones aprovechando así su bajo consumo, y largo alcance en relación a consumo.

El sistema estará compuesto por tres partes bien diferenciadas:

- Una aplicación web multiusuario que deberá mostrar todos los datos relevantes de la empresa en la que trabaje el usuario.
- Dispositivos con sensores y actuadores situados en cada uno de los depósitos.
- Sistema de comunicaciones entre todos los elementos que componen el sistema utilizando la especificación LoRa.

## 1.8 DEFINICIONES Y ABREVIATURAS

### 1.8.1 Definiciones

#### 1) Del negocio:

- a) **Nodo:** En el contexto de este proyecto hablaremos de nodo como cada uno de los puntos que componen una red de abastecimiento hídrico.
- b) **Nodo emisor:** Todo aquel punto de la red de abastecimiento en la que se instale el software encargado de generar las lecturas del estado de esa localización y de sus depósitos.
- c) **Nodo puente:** Dispositivo encargado de hacer de intermediario entre los mensajes que le llegan de los nodos emisores que componen la red y el exterior (Internet).
- d) **Depósito:** depósito o tanque de agua, es un elemento muy importante en instalaciones de abastecimiento hídrico e instalaciones agropecuarias.
- e) **Batería:** Es un dispositivo que almacena energía eléctrica, mediante el uso de procedimientos electroquímicos. Luego, dicha energía, es devuelta en casi toda su totalidad.
- f) **Energía Solar Fotovoltaica:** *“transforma de manera directa la luz solar en electricidad empleando una tecnología basada en el efecto fotovoltaico. Al incidir la radiación del sol sobre una de las caras de una célula fotoeléctrica (que conforman los paneles) se produce una diferencia de potencial eléctrico entre ambas caras que hace que los electrones salten de un lugar a otro, generando así corriente eléctrica.”* [5]

- g) **Panel Fotovoltaico / Panal solar:** Un módulo fotovoltaico, es un panel que está formado por un conjunto de células solares, que se encargan de convertir directamente en electricidad los fotones que provienen de la luz del sol.

La producción de corriente, depende de la irradiación (nivel de iluminación), de modo que, cuanto más sea la luz captada, mayor será la intensidad eléctrica a través de la célula.

- h) **Bomba hidráulica:** también conocida como bomba de agua, es una máquina que transforma la energía con la que es accionada, en energía del fluido, al incrementar la energía se aumenta la presión permitiendo así mover el líquido de una zona de menor presión a otra zona de mayor presión.

## 2) Del sistema:

- a) **Rol:** Función que alguien o algo desempeña en el sistema.
- b) **Sesión:** Una sesión es un conjunto de interacciones que tienen lugar en su sitio web en un periodo determinado. Finalizan pasado un periodo de tiempo de inactividad o al cerrar el navegador web. Es utilizada para mantener un estado asociado a un usuario en el lado del cliente, para ello se utilizan Cookies.
- c) **Iniciar Sesión:** Acción de comenzar una sesión de un usuario específico, permitiendo así identificarse frente a un sistema o servicio.
- d) **Login:** Véase "Iniciar Sesión".
- e) **Logout:** es la acción de terminar una sesión de un usuario específico, finalizando así el uso de un servicio o sistema. Es la acción inversa de iniciar sesión.
- f) **Registrarse:** Proceso por el cual un usuario se da de alta en el sistema como empleado de una determinada empresa.
- g) **Requerimiento:** Función o característica que debe poseer el producto software.
- h) **Requisito:** Véase "Requerimiento".
- i) **Interfaz de Usuario:** Medio que permite la comunicación entre usuario y sistema.
- j) **Conexión:** Comunicación entre dos entes que tienen características similares de comunicación.
- k) **Password:** Contraseña o clave para autenticar el ingreso a un lugar o sitio.

## 3) De Tecnología:

- a) **Framework:** es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.
- b) **Bootstrap:** Framework o biblioteca multiplataforma de código abierto para diseño de sitios y aplicaciones web.
- c) **Sistema Gestor de Base de Datos:** Son un tipo de software muy específico dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.
- d) **Base de Datos:** Conjunto de datos que pertenecen al mismo contexto, almacenados sistemáticamente para su posterior uso.

- e) **SGBD**: Sistema gestor de Base de Datos, también conocido en inglés como DBMS. Software destinado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan.
- f) **Enlace**: Referencia en un documento de hipertexto a otro documento o recurso.
- g) **Aplicación**: Programa informático diseñado para facilitar al usuario la realización de un determinado tipo de trabajo.
- h) **H2**: Sistema de gestión de base de datos relacional desarrollado en Java.
- i) **Protocolo**: Señal mediante la cual se reconoce que puede tener lugar la comunicación o la transferencia de información.
- j) **Servidor**: Computadora conectada a una red que pone sus recursos a disposición del resto de los integrantes de la red. Suele utilizarse para mantener datos centralizados o para gestionar recursos compartidos.
- k) **Periférico**: En informática, término utilizado para dispositivos, como unidades de disco, impresoras, módem que están conectados a un ordenador o computadora y son controlados por su microprocesador.
- l) **LoRa**: Es una técnica de modulación de espectro expandido derivada de la tecnología de espectro expandido de chirp (CSS). Es una plataforma inalámbrica de largo alcance y bajo consumo que se ha convertido en la tecnología de facto para las redes de Internet de las cosas (IoT) en todo el mundo. [6]
- m) **Payara**: Payara Server es un servidor de aplicaciones de código abierto derivado de GlassFish Server Open Source Edition. Fue creado en 2014 por C2B2 Consulting como un reemplazo del GlassFish después de que Oracle anunciara que estaba descontinuoando el soporte comercial para GlassFish. En comparación con GlassFish, Payara Server se lanza con más frecuencia trimestralmente con correcciones de errores, parches y mejoras adicionales. Desde abril de 2016 Payara Server está patrocinado por Payara Services Ltd, que es responsable del desarrollo continuo y la coordinación de las contribuciones de la comunidad. [7]
- n) **Java**: es un lenguaje de programación de propósito general, concurrente, orientado a objetos, que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo, el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. [8]
- o) **Web Service**: Un servicio web (en inglés, Web Service o Web services) es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios Web [9]
- p) **Socket**: Los sockets son un mecanismo que nos permite establecer un enlace entre dos programas que se ejecutan independientes el uno del otro (generalmente un programa cliente y un programa servidor). [10]

- q) **MQTT:** (Message Queue Telemetry Transport) es un protocolo de transporte de mensajes Cliente/Servidor basado en publicaciones y suscripciones a los denominados “tópicos”.

Cada vez que un mensaje es publicado será recibido por el resto de dispositivos adheridos a un tópico del protocolo. [11] Broker MQTT, también conocido como agente, está en el corazón de cualquier protocolo de publicación / suscripción. El agente es responsable de recibir todos los mensajes, filtrar los mensajes, determinar quién está suscrito a cada mensaje y enviar el mensaje a estos clientes suscritos. También tiene las sesiones de todos los clientes persistentes, incluidas las suscripciones y los mensajes perdidos. Otra responsabilidad del Broker es la autenticación y autorización de los clientes. [12]

- r) **IoT:** El Internet de las Cosas (IoT) hace referencia a una red de objetos de uso cotidiano, interconectados entre sí. Se trata de una nueva tendencia que surge tras el desarrollo del Smartphone, y que afecta a diferentes sectores, como la industria, la sanidad, la administración pública... [13]

- s) **Un mapa de sitio web:** (o mapa de sitio o mapa web) es una lista de las páginas de un sitio web accesibles por parte de los buscadores y los usuarios. Esto ayuda a los visitantes y a los bots de los motores de búsqueda a hallar las páginas de un sitio web. [14]

- t) **Arduino:** es una plataforma de creación de electrónica de código abierto, la cual está basada en hardware y software libre, flexible y fácil de utilizar para los creadores y desarrolladores. Esta plataforma permite crear diferentes tipos de microordenadores de una sola placa a los que la comunidad de creadores puede darles diferentes tipos de uso. [15]

- u) **MVC:** Modelo Vista Controlador. Es un patrón de arquitectura software que consiste en separar las diferentes responsabilidades de una aplicación: los datos, la interfaz de usuario y la lógica de control; en tres tipos de componentes a los que llama: Modelos, Vistas y Controladores.

- v) **UML:** UML son las siglas de “Unified Modeling Language” o “Lenguaje Unificado de Modelado”. Se trata de un estándar que se ha adoptado a nivel internacional por numerosos organismos y empresas para crear esquemas, diagramas y documentación relativa a los desarrollos de software (programas informáticos). [16]

- w) **IEEE:** Instituto de Ingeniería Eléctrica y Electrónica, es una asociación mundial de ingenieros dedicada a la normalización y el desarrollo en áreas técnicas.

- x) **I2C:** significa Circuito Interintegrado (Por sus siglas en Inglés Inter-Integrated Circuit) es un protocolo de comunicación serial desarrollado por Phillips Semiconductors allá por la década de los 80s. Básicamente se creó para poder comunicar varios chips al mismo tiempo dentro de los televisores. El I2C toma e integra lo mejor de los protocolos SPI y UART. Con el I2C podemos tener a varios maestros controlando uno o múltiples esclavos. Esto puede ser de gran ayuda cuando se van a utilizar varios microcontroladores para almacenar un registro de datos hacia una sola memoria o cuando se va a mostrar información en una sola pantalla. [17]

- y) **JSON:** (JavaScript Object Notation) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript. Es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos. [18]

- z) **Singleton**: es un patrón de diseño creacional que nos permite asegurarnos de que una clase tenga una única instancia, a la vez que proporciona un punto de acceso global a dicha instancia. [19]

## 1.8.2 Abreviaturas

- **RQF**: Requisito Funcional.
- **RQNF**: Requisito No Funcional.
- **RQ\_I**: Requisitos de Interfaz.
- **CAR**: Característica.
- **BD**: Utilizado para referirse a la Base de Datos.
- **CU**: Caso de uso. Representa una de las acciones que puede llevar a cabo el usuario.
- **ERS**: Especificación de Requisitos de Software.
- **ECTS**: European Credit Transfer and Accumulation System (ECTS) (Sistema Europeo de Transferencia y Acumulación de Créditos en idioma español).
- **CRUD**: Acrónimo de las funciones básicas en base de datos: Create, Read, Update, Delete.

## 2. ESPECIFICACIÓN DE REQUISITOS

### 2.1 INTRODUCCIÓN

El presente apartado representa la Especificación de Requisitos de Software (ERS) de este sistema. Su estructura sigue las directrices referidas en el estándar *IEEE Recommended Practice for Software Requirements Specifications IEEE 830-1998*. [20]

### 2.2 DESCRIPCIÓN GENERAL

#### 2.2.1 Perspectiva y funciones del producto

El sistema “**Hydro Manager**” es independiente de otros sistemas y se subdivide en diferentes módulos encargados de gestionar las diferentes tareas. Uno de los principales será la aplicación web llamada “Hydro Manager”, accesible desde cualquier navegador, ya sea desde un ordenador o cualquier dispositivo móvil.

El sistema permitirá al usuario conocer el volumen existente en cada uno de los depósitos de agua que compongan la red, activar el llenado de los depósitos de manera manual o automática, visualizar un histórico de la energía eléctrica generada, así como la entrada de energía actualizada en cada momento y una comparación con la intensidad de luz solar, conocer el consumo eléctrico actual y el estado de las baterías y por último mostrar un histórico de temperaturas y presión en la localización de cada uno de los depósitos.

#### Funciones de la aplicación Web “Hydro Manager”

1. **Inicio de sesión**: con acceso a las funcionalidades de la aplicación web.
2. **Cierre de sesión o logout**: el sistema permitirá cerrar sesión en todo momento.
3. **Registro de usuarios**. El sistema admitirá el registro de usuarios con el rol usuario.

4. **Visualización de red de depósitos.** El sistema permitirá el seguimiento con gráficas y tablas de la información actualizada de cada nodo que compone la red e histórico de:
  - 4.1. Volumen actual en cada depósito.
  - 4.2. Estado de la bomba asignada a dicho depósito.
  - 4.3. Temperatura y presión atmosférica.
  - 4.4. Entrada y consumo de energía.
  - 4.5. Luz solar: indicando la existencia o no.
5. **Control de llenado de depósitos:** la aplicación web permitirá la activación y desactivación de la bomba de llenado de cada depósito.
6. **Mi Cuenta:** El sitio web tendrá una sección mi cuenta en la que se mostrarán los datos del usuario permitiendo también su modificación.
  - 6.1. Nombre.
  - 6.2. Email.
  - 6.3. Contraseña: No se mostrará, pero se permitirá su modificación.
  - 6.4. Organización.
7. **Contacto:** la aplicación web contará con formulario e información de contacto con la empresa desarrolladora del sistema.
8. **Mapa del sitio:** el sistema contará con una página con una gráfica o lista del mapa del sitio web que servirá como ayuda para los usuarios y para la indexación de los buscadores.

### 2.2.3 Características de los usuarios

La aplicación web tendrá dos tipos de usuario o rol.

Para tener acceso a la funcionalidad de la aplicación web se debe iniciar sesión y cada usuario de rol “propietario” tendrá acceso sólo a la red de depósitos de su organización.

<b>Tipo de Usuario</b>	Propietario
<b>Formación</b>	Bajo nivel de conocimientos técnicos
<b>Habilidades</b>	Manejo de ordenador o dispositivo móvil
<b>Actividades</b>	<ul style="list-style-type: none"> <li>• Login.</li> <li>• Visualización de datos y gráficas.</li> <li>• Activación o desactivación de bombas de agua.</li> <li>• Modificación de la configuración de su empresa.</li> </ul>

Tabla 1: Características de usuario rol “Propietario”

<b>Tipo de Usuario</b>	Administrador
<b>Formación</b>	Alto nivel de conocimientos técnicos
<b>Habilidades</b>	Manejo de ordenador o dispositivo móvil
<b>Actividades</b>	<ul style="list-style-type: none"> <li>• Gestión de la base de datos:</li> <li>• CRUD de Usuarios.</li> <li>• CRUD de Empresas.</li> <li>• CRUD de Configuraciones.</li> <li>• CRUD de Nodos.</li> <li>• CRUD de Depósitos.</li> <li>• CRUD de Registros de Nodo.</li> <li>• CRUD de Registros de Depósito.</li> </ul>

Tabla 2: Características de usuario rol "Administrador"

## 2.2.4 Casos de Uso

A continuación, se muestra el diagrama de casos de uso y una descripción más en detalle de cada uno de los casos.

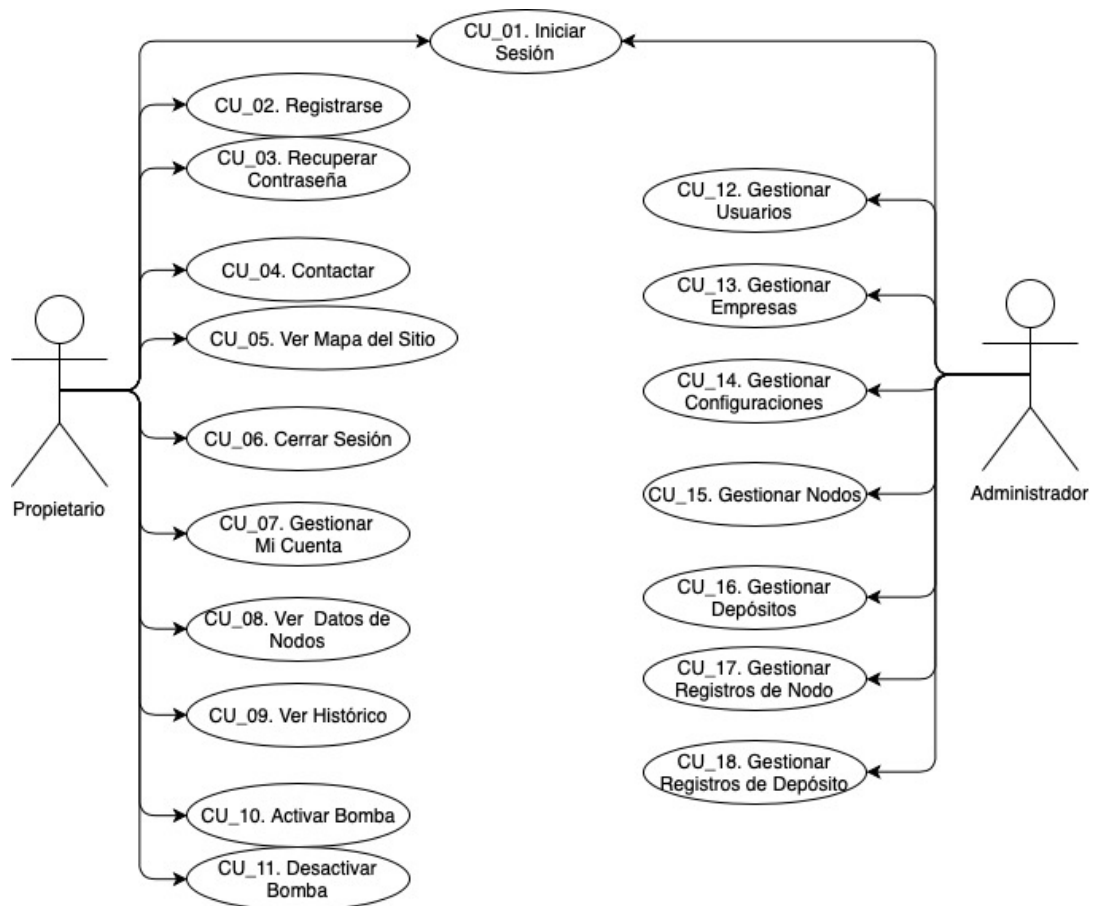


Ilustración 2: Diagrama de casos de uso.

**CU\_01. Iniciar Sesión:** Para acceder a las funcionalidades más importantes de la aplicación es necesario que el usuario se identifique.

**CU\_02. Registrarse:** El usuario tiene la posibilidad de registrarse, pero no podrá iniciar sesión hasta que el Administrador de la Base de datos verifique su identidad.

**CU\_03. Recuperar Contraseña:** Se enviará un email al usuario con una nueva contraseña temporal con la que acceder y cambiar la contraseña a su gusto.

**CU\_04. Contactar:** El usuario podrá acceder a formulario e información de contacto.

**CU\_05. Ver Mapa del Sitio.**

**CU\_06. Cerrar Sesión:** Una vez haya iniciado sesión el usuario podrá cerrar sesión en cualquier momento.

**CU\_07. Gestionar Mi Cuenta:** El usuario puede acceder a la sección “mi cuenta” y en ella visualizar sus datos personales (Empresa, nombre, email, empresa,) y modificar los datos de usuario y la configuración de la empresa. El usuario podrá modificar su nombre de usuario y contraseña en la sección “Mi Cuenta”.

**CU\_08. Ver Datos de Nodos (Depósitos):** El usuario visualizará un panel con gráficas e información actualizada de cada uno de los nodos que compongan la red de depósitos.

**CU\_09. Ver Histórico:** Tras seleccionar un nodo el usuario podrá conocer su histórico en detalle.

**CU\_10. Activar Bomba:** El usuario autenticado podrá activar la bomba encargada de llenar el depósito seleccionado.

**CU\_11. Desactivar Bomba:** El usuario podrá desactivar la bomba de uno de los depósitos.

**CU\_12. Gestionar Usuarios.** El usuario “Administrador” podrá crear, leer, actualizar y borrar usuarios desde el panel de administración.

**CU\_13. Gestionar Empresas.** Administrador podrá crear, leer, actualizar y borrar las empresas que se almacenen en la base de datos.

**CU\_14. Gestionar Configuraciones.** El administrador podrá crear, leer, actualizar y borrar la configuración de cada empresa.

**CU\_15. Gestionar Nodos.** El usuario “Administrador” podrá gestionar los nodos de cada empresa.

**CU\_16. Gestionar Depósitos.** El usuario “Administrador” podrá crear, leer, actualizar y borrar depósitos desde el panel de administración.

**CU\_17. Gestionar Registros de Nodo.** El usuario “Administrador” podrá crear, leer, actualizar y borrar registros del estado de un nodo desde el panel de administración.

**CU\_18. Gestionar Registros de Depósito.** El Administrador podrá gestionar los registros que indican el estado de los depósitos de la aplicación desde el panel de administración.

## 2.2.5 Restricciones

1. Los nodos emisores deben de utilizar algún **sistema de comunicación diferente al uso de cobertura móvil o WIFI** ya que en ocasiones no es posible el uso de ésta.
2. El sistema deberá estar alimentado con **baterías cargadas por energía solar** puesto que en la mayoría de los casos no se tiene acceso a la red eléctrica comercial.
3. El puente LoRa y el contenedor de aplicaciones deberán tener conexión a internet.
4. Normativa LoRa:
  - a. Según normativa de LoRa, en Europa se debe utilizar la frecuencia de **868MHz**.
  - b. Potencia máxima de emisión: 25 mW.
  - c. **Regla del 1%:** No podemos transmitir más que el 1% del tiempo. Es decir, si por ejemplo enviar un paquete nos lleva 100 ms, tendremos que mantenernos a la espera al menos 900 ms. De esta manera dejamos el canal libre para otros posibles usuarios. Como norma general, si no se realizan más de un envío cada 3 minutos estaremos cumpliendo la normativa.

## 2.3 REQUISITOS

### 2.3.1 Requisitos de Interfaces Externas

A continuación, se van a definir los requisitos de interfaces externas diferenciándolo entre los 3 programas a desarrollar: Aplicación Web “Hydro Manager”, Nodo Emisor y Nodo Puente.

#### Aplicación Web “Hydro Manager”

##### Interfaces de Usuario

- RQ\_I 1.** La interfaz con el usuario consistirá en una aplicación o página web compuesta por un conjunto de ventanas, botones, gráficas y tablas diseñadas y desarrollada específicamente para el sistema.
- RQ\_I 2.** La interfaz será accesible y sencilla adecuándose así a las características y conocimientos de los usuarios, que deberán saber en todo momento en que sección se encuentran.
- RQ\_I 3.** La interfaz web debe ser responsive y con una estructura homogénea y balanceada con 4 niveles de profundidad a lo sumo.
- RQ\_I 4.** La interfaz debe disponer de formulario de Login en el que se solicitará al usuario la introducción de usuario y contraseña.

##### Interfaces Software

- RQ\_I 5.** El usuario deberá utilizar un navegador Web que soporte HTML5 y Javascript para mostrar de manera correcta las gráficas y estilos.

##### Interfaces de Comunicación

- RQ\_I 6.** La comunicación entre el contenedor de aplicaciones y el cliente (navegador web) se realizará a través del protocolo HTTP.
- RQ\_I 7.** El contenedor de aplicaciones web deberá hacer uso de MQTT para la publicación en el bróker MQTT de las acciones realizadas por el usuario.

#### Nodo Emisor

## Interfaces Hardware

---

- RQ\_I 18.** Relé / actuador: El software deberá ser capaz de activar, desactivar y conocer el estado de los relés o válvulas que tenga conectados.
- RQ\_I 19.** El programa deberá ser capaz de interpretar los datos obtenidos por los sensores conectados, obteniendo así datos como temperatura, presión, volumen de llenado de un depósito, voltaje y amperaje.

## Interfaces de Comunicación

---

- RQ\_I 10.** La comunicación entre el nodo emisor y el nodo puente se realizará a través de LoRa.

## Nodo Puente

---

## Interfaces de Comunicación

---

- RQ\_I 11.** La comunicación entre el nodo puente y los nodos emisores será a través de LoRa.
- RQ\_I 12.** El software del nodo puente hará uso de MQTT tanto para la publicación en el bróker MQTT de los datos recibidos por los nodos emisores como para recibir los mensajes del contenedor de aplicaciones.

## 2.3.2 Requisitos Funcionales

---

A continuación, se van a definir los requisitos funcionales de manera separada para los 3 programas a desarrollar: Aplicación Web “Hydro Manager”, Nodo Emisor y Nodo Puente.

## Aplicación Web “Hydro Manager”

---

- RQF 1.** Formulario de Registro. El sistema dispondrá de un enlace en el menú superior que llevará a un formulario donde los usuarios podrán registrarse en la aplicación web rellorando los siguientes datos:
- Email.
  - Usuario.
  - Nombre completo.
  - Contraseña.
  - Organización.
- RQF 2.** Registro de usuarios. El sistema permitirá registrarse como nuevo usuario, dicho usuario no tendrá acceso hasta que el administrador de base de datos verifique la autenticidad de su identidad y organización.
- RQF 3.** Sólo podrá existir un usuario con el mismo nombre de usuario.
- RQF 4.** No será posible la existencia de dos usuarios con el mismo email.
- RQF 5.** Formulario de Login: El sistema dispondrá de un enlace en el menú superior que llevará a un formulario de Login con el que el usuario podrá iniciar sesión.
- RQF 6.** En caso de no recordar la contraseña el usuario tendrá ocasión de pedir la recuperación de su cuenta, el sistema generará un nuevo password aleatorio, lo asignará al usuario y le enviará un email con dicha contraseña.

- RQF 7.** La información requerida durante el proceso de inicio de sesión de un usuario será nombre de usuario o email y contraseña.
- RQF 8.** Autenticación del Usuario. Los usuarios deberán identificarse para acceder a toda la funcionalidad del sistema.
- RQF 9.** Tras la validación del inicio de sesión, el sistema almacenará en sesión el nombre de usuario, nombre completo y su organización.
- RQF 10.** Un usuario sólo tendrá acceso a los datos correspondientes a su organización o empresa.
- RQF 11.** En el caso de que el usuario autenticado tenga el rol “Administrador” el sistema redireccionará al usuario al panel de administración.
- RQF 12.** La aplicación dispondrá de un enlace en el menú superior que aparecerá cuando se haya iniciado sesión. Se mostrará el nombre del usuario y un botón de “Cerrar Sesión”.
- RQF 13.** El usuario dispondrá de acceso a los apartados:
- a. Página de Inicio.
  - b. Iniciar Sesión / Registrarse / Cerrar sesión. Esta opción dependerá de si el usuario ha iniciado sesión o no.
  - c. Mi Cuenta (Cuando el usuario haya iniciado sesión).
  - d. Contacto.
- RQF 14.** El menú superior contendrá el logotipo de la aplicación.
- RQF 15.** El logotipo de la aplicación será un enlace que dará acceso a la página de inicio de la aplicación.
- RQF 16.** El usuario podrá acceder la sección “Mi Cuenta” en la que podrá visualizar siguientes datos:
- a. Nombre de Usuario.
  - b. Email.
  - c. Nombre completo.
  - d. Contraseña.
  - e. Empresa (Nombre, Descripción, Dirección, y Localidad).
  - f. Configuración de la empresa.
- RQF 17.** Se podrá modificar el nombre de usuario dentro del apartado “Mi Cuenta”.
- RQF 18.** Se permitirá modificar la contraseña a través de un formulario en la página “Mi Cuenta”.
- RQF 19.** El sistema permitirá modificar el nombre completo del usuario.
- RQF 20.** Dentro de la página “Mi Cuenta” se podrá modificar el email del usuario actual.

- RQF 21.** El sistema mostrará un panel de administración para los usuarios con rol “Propietario” con registros de todos los nodos que compongan la red. Una vez listados se podrá ver:
- a. Información de su empresa u organización.
  - b. Mapa con ubicación de todos los nodos de su organización.
  - c. Nombre del nodo.
  - d. Fecha de última actualización.
  - e. Temperatura.
  - f. Presión atmosférica.
  - g. Luz Solar.
  - h. Voltaje y Amperaje.
  - i. Depósitos (Puede contener uno o varios).
    - i. Volumen actual.
    - ii. Botón activar bomba.
    - iii. Estado.
      - 1. Activado (Llenándose).
      - 2. Desactivado (En reposo o consumiendo).
    - iv. Fecha última actualización.
- RQF 22.** El usuario podrá seleccionar un nodo y ver el histórico detallado vinculado al nodo seleccionado. Para ello se utilizarán tablas y gráficas con valores y fechas.
- RQF 23.** En el apartado histórico se podrá seleccionar la cantidad o fecha de registros a mostrar y usar en las gráficas.
- RQF 24.** La página se actualizará automáticamente en caso de existir nuevos datos.
- RQF 25.** La aplicación web contará con pie de página y este tendrá enlace a las páginas de “Mapa del sitio”.
- RQF 26.** La página “Contacto” será accesible sin iniciar sesión, mostrará información del desarrollador de la aplicación, un formulario de contacto y otros medios de contacto.
- RQF 27.** El formulario de contacto enviará un email con la duda al correo [info.hydromanager@gmail.com](mailto:info.hydromanager@gmail.com)
- RQF 28.** La página “Mapa del Sitio” será accesible sin iniciar sesión y mostrará una gráfica del mapa web.
- RQF 29.** Todos los formularios contarán con control de errores y notificación de éstos al usuario.
- RQF 30.** El usuario con el rol “Administrador” tendrá acceso a un panel de administración en el que podrá crear, leer, actualizar y borrar registros de todas las tablas de la base de datos permitiendo así una administración más sencilla.

## Nodo Emisor

---

- RQF 31.** El nodo emisor contará con sensores que detectarán y enviarán cada cierto tiempo configurado la información actualizada a su nodo puente.
- RQF 32.** Activación automática de la bomba de agua. El sistema activará automáticamente el llenado del depósito de agua de un nodo al descender de un volumen configurado.
- RQF 33.** Desactivado automático de la bomba de agua. El sistema desactivará la bomba de agua al alcanzar un volumen máximo configurado por el usuario.
- RQF 34.** Desactivado automático en caso de error. El sistema desactivará automáticamente el llenado del depósito en caso de lecturas erróneas del volumen o en caso de detectar que no aumenta el volumen a pesar de tener activada la bomba.
- RQF 35.** El nodo deberá estar siempre a la escucha de las ordenes que le envíe el nodo puente vía LoRa.
- RQF 36.** El nodo emisor recibirá la configuración del nodo puente al iniciar por primera vez y conseguir conectar con él.
- RQF 37.** El nodo deberá tener una configuración por defecto para el caso en el que no consiga conectar con el nodo puente.
- RQF 38.** El nodo emisor deberá estar a la escucha de posibles cambios en la configuración enviados por el Puente a través de Lora.
- RQF 39.** Todos los nodos emisores que estén dentro de una misma organización o empresa tendrán la misma configuración.
- RQF 40.** Los parámetros con posibilidad de configuración para los nodos de cada organización son:
- Frecuencia de envío de información.
  - Volumen de activación de llenado. (Ejemplo: 15%, Cuando el depósito se encuentre al 15% se activará la bomba).
  - Volumen de desactivación de llenado. (Ejemplo: 95%, Cuando el depósito se encuentre al 95% se desactivará la bomba).

## Nodo Puente

---

- RQF 41.** El nodo puente validará la información recibida de los nodos emisores antes de trabajar con ella.
- RQF 42.** El nodo puente publicará en los topics MQTT correspondientes los datos recibidos de los nodos emisores en caso de llegar en el formato adecuado.
- RQF 43.** El nodo puente enviará a los nodos emisores la configuración que deben tener éstos cuando la soliciten.
- RQF 44.** El nodo puente hará llegar a todos los nodos emisores de una empresa su nueva configuración en caso de ser modificada.
- RQF 45.** En nodo puente hará llegar al nodo adecuado la solicitud de activación o desactivación de una de sus bombas.

### 2.3.3 Requisitos De Rendimiento

- RQNF 1.** El sistema deberá ser confiable, teniendo una disponibilidad de 24 horas los 7 días de la semana.
- RQNF 2.** El sistema deberá ser escalable y poder gestionar más de una empresa, más de un nodo por empresa y más de un depósito por nodo.

### 2.3.4 Otros Requisitos

#### De Seguridad

- RQNF 3.** Sólo se permitirá el acceso a los datos y utilidades del sistema a aquellos usuarios que inicien sesión.
- RQNF 4.** Se usará un sistema de cifrado para almacenar las contraseñas de los usuarios en la base de datos. El algoritmo de encriptación utilizado será sha512.
- RQNF 5.** Únicamente tendrá acceso a la base de datos el usuario administrador de base de datos accediendo a ella con su usuario y contraseña.
- RQNF 6.** El sistema deberá garantizar la seguridad de la información almacenada.

#### De Usabilidad

- RQNF 7.** La aplicación web deberá seguir las normas de usabilidad de contenido web estipulado según el estándar WCAG 2.1 de W3C. [21]

## 3. PLANIFICACIÓN FINAL Y ANÁLISIS DE ESFUERZO

En este capítulo se va a mostrar la planificación final del proyecto combinado con un análisis del esfuerzo desglosando las tareas y mostrando un cálculo aproximado de horas dedicadas a cada una de ellas. Se realiza un diagrama Gantt con la planificación final del proyecto.

### 3.1 DIAGRAMA DE GANTT

Se añaden y organizan todas las tareas para obtener un cálculo aproximado del tiempo total destinado y de la fecha de finalización del proyecto.

Para el cálculo del número de días de duración de cada tarea en el diagrama de Gantt se ha tenido en cuenta que cada día se destinan entre 2 horas y 2 horas y media de trabajo ya que este TFG se realiza compaginándolo con la vida laboral, en caso de solaparse varias tareas se repartiría ese tiempo entre ellas. Al cálculo de horas también se le sumarán algunas horas extra de ocasiones en las que se trabaja durante un fin de semana o más horas diarias de las habituales.

<b>Encargado del proyecto</b>	Alejandro Cencerrado Tello
<b>Fecha de Inicio y fin del proyecto</b>	14 Febrero 2022 - 21 Noviembre 2022
<b>Nº de Tareas</b>	34
<b>Recursos</b>	1

Tabla 3: Gantt Resumen del Proyecto

Nombre	Fecha de inicio	Fecha de fin	Duración
Definición de tema del proyecto	17/01/22	20/01/22	4
Introducción	21/01/22	1/02/22	8
Motivación	21/01/22	24/01/22	2
Objetivos	25/01/22	26/01/22	2
Problemas Abordados	27/01/22	27/01/22	1
Resultados Previstos	28/01/22	28/01/22	1
Visión General del Documento	31/01/22	1/02/22	2
Ámbito	31/01/22	31/01/22	1
Definiciones y Abreviaturas	31/01/22	1/02/22	2
Fase de Análisis	2/02/22	28/02/22	19
Especificación de Requisitos	2/02/22	16/02/22	11
Casos de uso	17/02/22	21/02/22	3
Planificación	22/02/22	28/02/22	5
Diseño	1/03/22	31/03/22	23
Diseño Arquitectónico	1/03/22	7/03/22	5
Diseño Hardware	8/03/22	15/03/22	6
Diseño Software	16/03/22	31/03/22	12
Diagrama de Clases	16/03/22	31/03/22	12
BD - Diagrama E/R	16/03/22	21/03/22	4
Implementación de Software	1/04/22	19/08/22	100
Nodo Emisor	1/04/22	19/08/22	100
Nodo Punte	1/04/22	7/07/22	70
Broker MQTT	1/04/22	8/04/22	6
Base de Datos	1/04/22	22/04/22	16
Aplicación web	1/04/22	29/07/22	86
Pruebas	2/05/22	31/10/22	129
Pruebas de cobertura LoRa	2/05/22	3/05/22	2
Contrucción del prototipo	22/08/22	7/10/22	35
Pruebas Finales	10/10/22	31/10/22	15
Conclusiones y Trabajos Futuros	2/11/22	3/11/22	2
Memoria	21/01/22	31/03/22	50
Correcciones Finales Memoria	4/11/22	24/11/22	15
Depósito	21/11/22	21/11/22	0
Fin Plazo Depósito	2/12/22	2/12/22	0

Tabla 4: Planificación - Resumen de Tareas

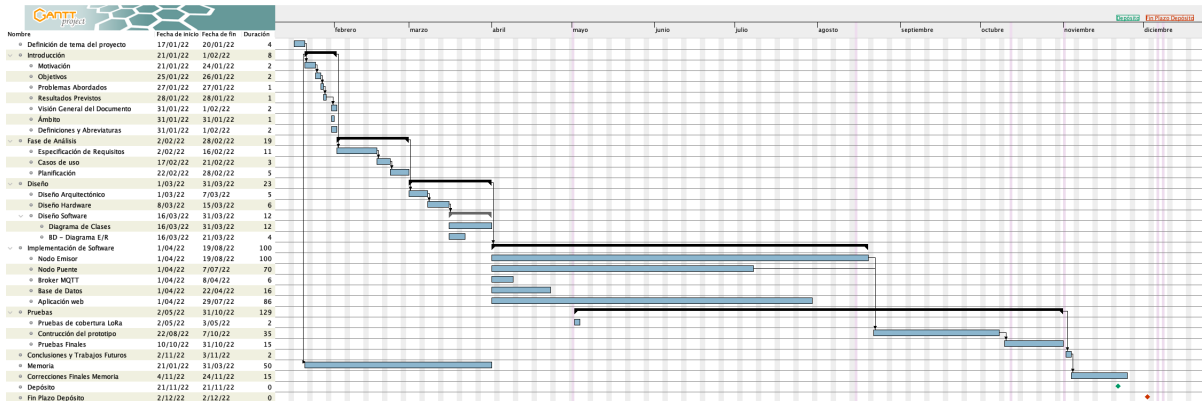


Ilustración 3: Diagrama Gantt miniatura

(Diagrama Gantt a mayor escala disponible en Anexo 1: Planificación, Ilustración 1)

### 3.2 ANÁLISIS DE ESFUERZO

Listado de tareas con el cálculo aproximado de las horas destinadas a cada una de ellas.

TAREA	HORAS
Definición del tema del proyecto y presentación de la propuesta	8
Introducción de la memoria	16
Fase de Análisis	40
Fase de diseño	46
Implementación del Software	
o Desarrollo para Nodo Emisor	100
o Desarrollo para Nodo Puente	50
o Instalación y configuración de Broker MQTT	6
o Instalación y configuración de Base de Datos	20
o Aplicación Web “Hydro Manager”	70
Pruebas	
o Prueba de cobertura LoRa	4
o Construcción del prototipo	80
o Pruebas finales de todo el sistema	30
Correcciones finales de la memoria	35
<b>TOTAL</b>	<b>505 horas</b>

Tabla 5: Análisis de esfuerzo

## 4. DISEÑO

En este apartado se muestran todas las decisiones tomadas en la fase de diseño del sistema, dichas decisiones se han realizado en base a los resultados de la fase de análisis anterior.

## 4.1 DISEÑO ARQUITECTÓNICO

A continuación, se van a definir los diferentes elementos o niveles que componen el sistema y la manera en la que se comunican entre ellos.

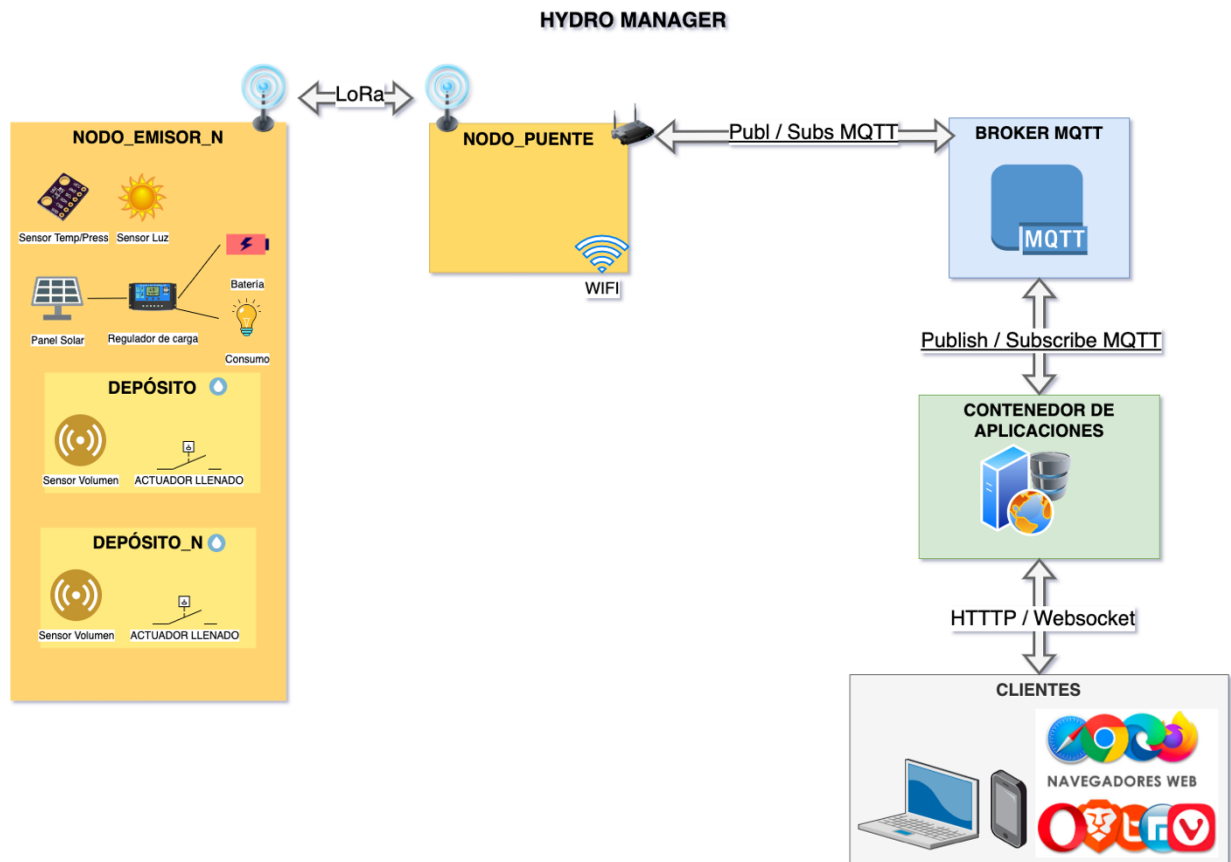


Ilustración 4: Diseño Arquitectónico.

### 4.1.1 Elementos

Como se puede apreciar en el diagrama nos encontramos ante un sistema de 5 capas:

- A. Nodo Emisor.
- B. Nodo Puente.
- C. Broker MQTT.
- D. Contenedor de aplicaciones Payara.
- E. Clientes, usuarios de la aplicación web “HydroManager” a través del navegador web.

Cada capa sólo se comunica con sus capas contiguas.

#### A - Nodo Emisor

Dispositivo con todos los sensores instalados, puede contener uno o varios depósitos y tendrá un relé actuador para cada uno de ellos que se encargará de activar el llenado o abrir la válvula correspondiente. El nodo emisor envía al Gateway cada cierto tiempo los datos con el estado actual (batería, carga solar, agua en el depósito, temperatura ambiente, presión...). El sistema podrá contener varios nodos emisores componiendo así una red.

Programado con el entorno de desarrollo Arduino.

El nodo emisor contará con:

- LoRa.
  - Chip LoRa.
  - Antena.
- Sensor de temperatura y presión atmosférica.
- Sensor de luz solar.
- Instalación de energía fotovoltaica.
  - Uno o varios paneles solares.
  - Regulador de carga.
  - Baterías.
- Sensores de voltaje e intensidad.
- Alimentación por energía fotovoltaica.
- Uno o varios depósitos que tendrán:
  - Sensor de cantidad de líquido en el depósito (volumen).
  - Actuador que active la bomba de agua o válvula para su llenado.

El emisor enviará por LoRa al puente la información leída por los sensores periódicamente según el tiempo configurado o al cambiar el estado del actuador.

También permanecerá a la escucha para actuar en caso de recibir mensajes del Gateway o puente LoRa.

## B - Nodo Puente

---

Puente de nodos. Dispositivo que recibe los mensajes de los nodos y los envía al servidor. Programado con el entorno de desarrollo Arduino.

Es el encargado de hacer de intermediario entre los mensajes que le llegan de los nodos que componen la red y el exterior (Internet) y por tanto el servidor.

Este componente dispone de conexión WIFI y LoRa.

Se encarga de las siguientes 3 funciones:

1. Escucha de mensajes LoRa.
2. Carga de configuración.
3. Publicación

### **Escucha de Mensajes LoRa**

El nodo puente se mantiene a la escucha de los mensajes que le puedan llegar de los nodos de la red a través de LoRa y también permanece a la escucha de los cambios en los topics MQTT del bróker situado en el servidor.

En caso de cambiar el topic de configuración de alguna empresa el puente debe notificarlo a todos los nodos que pertenezcan a ella.

En caso de cambiar de estado algún topic relacionado con un actuador el puente debe notificarlo al nodo correspondiente que deberá actuar en consecuencia.

## Carga de Configuración

El puente conecta con el servidor y lee los topics correspondientes a la configuración de su empresa, en esta configuración se encuentra el tiempo en segundos que deben esperar los nodos para enviar un nuevo registro.

El nodo puente enviará dicha configuración a un nodo emisor cuando se ponga en marcha y lo solicite y cada vez que haya algún cambio en la configuración.

## Publicación MQTT

Todos los mensajes recibidos de los nodos emisores son publicados en el bróker MQTT situado en el servidor, más adelante, en el apartado comunicaciones se mostrará la estructura seguida para los topics y el diseño de los mensajes MQTT publicados.

### C - Broker MQTT Mosquitto.

---

Instalado en una Raspberry Pi.

Contendrá:

- Topics de configuración de los nodos de cada empresa.
- Topics de estado de cada nodo.
  - Temperatura, presión, voltajes...
- Topic de estado de cada depósito.
  - Porcentaje.
  - Estado del interruptor de bomba/válvula, activo/desactivado.

El contenedor de aplicaciones contará con una aplicación a la escucha de los cambios en todos los topics. Todo cambio quedará registrado en la base de datos en su tabla correspondiente.

### D - Servidor o Contenedor de Aplicaciones

---

La parte del contenedor de aplicaciones esta compuesta por un Servidor Payara 5 encargado de contener la aplicación web “Hydro Manager” y la base de datos utilizada por la propia aplicación. Interactúa con el cliente y con el Broker MQTT.

#### Contenedor de aplicaciones

Contenedor o servidor de aplicaciones Payara 5 encargado de realizar las consultas en la BD y servir la página web al cliente con la información actualizada.

Por cada cliente que inicie sesión se abrirá un socket servidor-cliente por el que se transferirán los nuevos registros en JSON para que se vayan actualizando, permitiendo así una IU asíncrona en la app web.

#### Base de Datos.

Base de datos H2 utilizada para almacenar todos los datos de histórico del sistema, usuarios, empresas, nodos, depósitos y configuraciones. Se encuentra alojada en el propio servidor de aplicaciones Payara 5.

Se utilizará JPA. Diagrama relacional disponible en el apartado 4.2.2 Base de Datos, de este mismo documento.

## E - Clientes de la Aplicación Web “Hydro Manager”

Cualquier usuario que acceda al sistema a través de un navegador web, ya sea accediendo desde un ordenador o desde cualquier dispositivo móvil. En él se mostrarán las tablas y gráficos con la información útil (temperatura, voltajes, volumen...) de toda la red de depósitos. El usuario podrá activar el llenado alguno de los depósitos si lo desea, ver información de su empresa o cambiar la configuración.

En caso de existir nuevos datos la página web deberá actualizarse automáticamente añadiendo dichos datos a las tablas o gráficas que en ese momento se estén mostrando.

Para ello el sistema abre un socket que conecta el cliente con el servidor, en caso de existir nuevos registros el cliente los recibe por socket en formato JSON y por medio de Javascript y JQuery se actualiza la vista sin requerir ninguna acción del usuario.

### 4.1.2 Comunicaciones

- A. MQTT
- B. LoRa
- C. WebSocket.

#### A – MQTT

En las comunicaciones MQTT se utilizan 4 formatos de mensaje diferente entre el nodo puente y la aplicación web “Hydro Manager” que se publicarán en sus correspondientes topics de la siguiente forma.

#### Topics

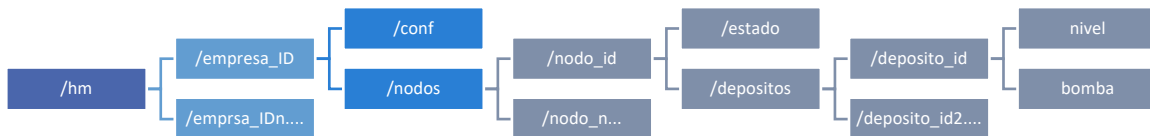


Ilustración 5: Diagrama de organización de Topics.

El topic principal o raíz de la aplicación será “/hm” y bajo él cada empresa tendrá un subtopic que corresponderá con el identificador que tengan en la base de datos.

Cada empresa tendrá dos subtopics: “/depositos” en el que se publicará un mensaje con los parámetros de configuración actuales y “/nodos” bajo el que recaerán todos los nodos que formen la red de dicha empresa y serán identificados por su id.

El subtopic de un nodo contendrá los subtopics: “/estado” en el que se publicará el mensaje con el estado actual del nodo y “/depositos” bajo el que recaerán el o los depósitos que tenga el nodo seleccionado.

Y por último el subtopic de un depósito contendrá el subtopic “/nivel” con el mensaje con el estado actual del depósito y “/bomba” con el momento actual del actuador o relé.

## Listado de topics:

---

1. /hm/empresa\_id/depositos (Ejemplo: /hm/1/depositos)
2. /hm/empresa\_id/nodos/nodo\_id/estado (Ejemplo: /hm/1/nodos/1/estado)
3. /hm/empresa\_id/nodos/nodo\_id/depositos/deposito\_id/nivel (Ejemplo: /hm/1/nodos/1/depositos/1/nivel)
4. /hm/empresa\_id/nodos/nodo\_id/depositos/deposito\_id/bomba (Ejemplo: /hm/1/nodos/1/depositos/1/bomba)

## Mensajes

---

Formato de los mensajes MQTT utilizados y ejemplos.

### Configuración de Empresa

---

Ejemplo → `{"frecuencia":180,"activacion":15,"desactivacion":90,"empresa":1}`

JSON publicado en el topic `"/hm/empresa_id/configuración"` (Ejemplo: `/hm/1/configuracion`).

En este caso es la aplicación web la encargada de publicar en dicho topic cuando se modifica la configuración de una empresa.

En este mensaje se envían todos los datos necesarios para establecer la configuración de una empresa incluida la id de dicha empresa.

Siguiendo los datos del ejemplo la configuración haría referencia a la empresa con id 1, la frecuencia de envío de datos deberá ser cada 3 minutos, el llenado de un depósito comenzará cuando el nivel sea inferior al 15% y se detendrá automáticamente al llegar al 90%.

### Estado de Nodo

---

Ejemplo `{"tm":1,"n":1,"e":1,"t":23.52,"p":109805.4,"ls":1,"vp":12.28,"ap":0.30,"vb":12.28,"ab":0.30,"vs":12.28,"as":0.30}`

JSON publicado en el topic `"/hm/empresa_id/nodos/nodo_id/estado"` (Ejemplo: `/hm/1/nodos/1/estado`)

Todos los datos referentes a un nodo obtenidos a partir de los sensores, se mantiene el mismo mensaje recibido por LoRa del emisor.

- tm: tipo de mensaje. (1 para estado nodo y 2 para estado depósito)
- n: id del nodo.
- e: id de la empresa a la que pertenece el nodo.
- t: Temperatura en °C con 2 decimales.
- p: Presión en Pa.
- ls: luz solar, si hay o no luz.
- vp: voltaje paneles con 2 decimales.
- ap: amperios paneles con 4 decimales.
- vb: voltaje batería con 2 decimales.
- ab: amperios batería con 4 decimales.
- vs: voltaje salida con 2 decimales.
- as: amperios salida con 4 decimales.

La aplicación permanece suscrita a estos mensajes y se encarga de almacenar un nuevo registro de estado de nodo con dichos datos y de comunicarlo por websocket a los usuarios.

## Estado Depósito

---

Ejemplo: `{"tm":2,"n":1,"e":1,"d":1,"nivel":25,"estaActivo":1}`

JSON publicado en el topic `"/hm/empresa_id/nodos/nodo_id/depositos/deposito_id/nivel"`

(Ejemplo: `/hm/1/nodos/1/depositos/1/nivel`)

Se mantiene el mismo mensaje ya generado por el emisor y recibido por LoRa.

- tm: tipo de mensaje. (1 para estado nodo y 2 para estado depósito)
- n: id del nodo.
- e: empresa a la que pertenece el nodo.
- d: id del depósito.
- nivel: porcentaje de nivel del depósito con 2 decimales.
- estaActivo: indica si el llenado del depósito se encuentra activo en ese momento.

La aplicación web está suscrita a estos mensajes y se encarga de dar de alta el nuevo registro de estado de depósito en la base de datos y de enviarlo por websocket a todos los usuarios conectados.

## Estado Bomba/Actuador

---

Ejemplo: `{1}`

Sólo almacena el estado del actuador con un booleano.

JSON publicado en el topic `"/hm/empresa_id/nodos/nodo_id/depositos/deposito_id/bomba"`

(Ejemplo: `/hm/1/nodos/1/depositos/1/bomba`).

Tanto el servidor de aplicaciones como el puente publican este tipo de mensajes.

- El puente genera este mensaje al recibir por LoRa un mensaje de estado de depósito.
- El servidor de aplicaciones publica dicho mensaje si un usuario activa/desactiva un actuador desde el sitio web. El puente permanece suscrito a dicho topic para enviar la orden al nodo objetivo.

## B - LoRa

---

En las comunicaciones LoRa se envían diferentes mensajes en ambas direcciones.

Tipos de mensaje:

1. Estado de nodo.
2. Estado depósito.
3. Configuración / Solicitud de configuración.
4. Activación de Relé o Bomba.

Para reducir el número de mensajes LoRa no se envía una comunicación en concreto para el estado del relé ya que dicha información también se encuentra en estado de depósito.

Para limitar el tiempo de uso del canal los mensajes de estado se enviarán de manera periódica cumpliendo el tiempo de la configuración de la empresa, alternando entre el tipo de estado y limitando dicho intervalo a un mínimo de 180s.

---

## De Nodo Emisor a Nodo Puente

---

Mensajes de estado obtenidos a partir de las lecturas de los sensores, cambios en algún actuador o solicitud de configuración.

### Estado Nodo

---

Ejemplo:

```
{"tm":1,"n":1,"e":1,"t":23.52,"p":109805.4,"ls":1,"vp":12.28,"ap":0.30,"vb":12.28,"ab":0.30,"vs":12.28,"as":0.30}
```

Datos referentes al estado de un nodo obtenidos a partir de los sensores.

### Estado Deposito

---

Ejemplo: {"tm":2,"n":1,"e":1,"d":1,"nivel":25,"estaActivo":1}

A partir de este mensaje el puente generará uno de tipo estado bomba para el actuador.

### Solicitud de Configuración

---

Ejemplo: {"tm":3,"n":1,"e":1}

Enviado por el emisor durante el arranque para solicitar la configuración actual de su empresa.

- tm: tipo de mensaje. (3 para configuración)
- n: id del nodo.
- e: empresa a la que pertenece el nodo.

---

## De Nodo Puente a Nodo Emisor

---

### Configuración

---

Ejemplo: {"tm":3,"e":1,"frecuencia":30,"activacion":30,"desactivacion":90}

- tm: tipo de mensaje. (1 para estado nodo, 2 para estado depósito, 3 configuración, 4 actuador)
- e: empresa a la que pertenece el nodo.
- activacion: porcentaje de volumen mínimo por el que se debe activar el llenado.
- desactivacion: porcentaje de volumen del momento en que debe desactivarse el llenado.

Enviado al nodo que lo solicite o a todos los nodos de la empresa en caso de tratarse de una actualización de configuración. Cuando se envía a todos los nodos se hace a sus direcciones exactas, no por broadcast, de esta forma será más seguro.

### Activación de Relé

---

Ejemplo: {"tm":4,"n":1,"e":1,"d":1,"activarRele":1}

- tm: tipo de mensaje. (1 para estado nodo, 2 para estado depósito, 3 configuración, 4 actuador)
- e: empresa a la que pertenece el nodo.
- n: id del nodo al que pertenece el depósito.
- d: id del depósito.
- activarRele: 1|0 (1 para activar).

Enviado al nodo concreto para que active o desactive el relé o válvula deseado.

## C - WebSocket

---

Cuando el usuario accede a la web se inicia un websocket cliente-servidor y éste permanece a la escucha tanto de eventos generados por el usuario en la web como de mensajes enviados del servidor al usuario.

Es muy útil para actualizar dinámicamente la página de manera asíncrona cuando llegan nuevos registros de estado, tanto en el panel de control como en histórico.

El socket se inicia en las páginas en las que el usuario ya está logueado y por tanto se puede obtener su empresa.

La aplicación web agrupa las sesiones de conexiones al socket por la empresa a la que pertenezca el usuario, de manera que sólo se le enviarán registros de nodos pertenecientes a su empresa.

### Cliente -> Servidor

---

#### Activación / Desactivación de interruptor

---

Cuando el usuario pulsa en un botón de activación o desactivación de un actuador se envía un mensaje al servidor a través del websocket.

Ejemplo: `{"tipo": "INT", "idNodo": 1, "idEmpresa": 1, "idDeposito": 1, "valor": 1}`

- tipo: "INT" será el tipo utilizado para el interruptor o actuador.
- idNodo
- idEmpresa
- idDeposito
- valor: 1 | 0.

Cuando el servidor recibe este tipo de mensajes se encarga de publicar en el topic MQTT correspondiente el cambio para que el nodo puente actúe. `/hm/1/nodos/1/depositos/1/bomba {1}`

### Servidor -> Cliente

---

Cuando se genera un nuevo registro de estado de nodo o de nivel de depósito el servidor envía por websocket dicho registro a todos los usuarios conectados de la empresa a la que pertenezca el registro.

El sitio web con ayuda de Javascript y JQuery se encargará de añadir esos nuevos registros a las tablas y gráficas mostradas sin necesidad de recargar la página.

#### Estado Nodo

---

Ejemplo: `{"vPanelSolar":12.62,"ampPanelSolar":1.0E-4,"tipo":"estado_nodo","vSalida":12.62,"ampBateria":1.0E-4,"fecha_hora":"2022-11-16 12:54:37.647","vBateria":12.62,"temperatura":25.71,"luzSolar":false,"n":1,"presion":98544.86,"ampSalida":1.0E-4}`

- tipo: tipo de mensaje. ("estado\_nodo")
- n: id del nodo.
- temperatura: Temperatura en °C.
- presion: Presión en Pa.
- luzSolar: luz solar, si hay o no luz.
- vBateria: voltaje batería.
- ampBateria: amperios batería.
- vPanelSolar: voltaje paneles.



### **Paquete hm.config**

Contiene la clase **ParamConfiguracion** utilizada para almacenar los parámetros de configuración más importantes de todo el programa, de esta forma será mucho más sencillo modificar estos ajustes sin la necesidad de ir a los diferentes ficheros en los que se utilice dicho parámetro. (Anexo 2, Ilustración 2)

### **Paquete hm.entity**

Contiene todas las entidades de los datos con los que va a trabajar la aplicación y que van a ser almacenados en la base de datos utilizando persistencia. (Anexo 2, Ilustración 3)

### **Paquete hm.db**

Paquete que contiene todas las fachadas de las clases persistentes mostradas en hm.entity

(Anexo 2, Ilustración 4)

### **Paquete hm.webSocket**

(Anexo 2, Ilustración 5) Dicho paquete contendrá la clase **WebSocketManager**, esta clase es la encargada de gestionar los websockets cliente-servidor con los usuarios que inician sesión en la web. Las sesiones se almacenan en el HashMap sessionsPorEmpresa agrupadas según la empresa a la que pertenece cada usuario.

Hace uso de la clase Interruptor que explicaremos más adelante, cuando un usuario actúa sobre alguno de los botones de activación/desactivación de bomba de agua se llama a la función onMessage, por otro lado, se utilizará la función sendJsonString para el envío de nuevos registros, ya sea de estado nodo o estado depósito desde el servidor hacia los clientes que corresponda a través del websocket.

### **Paquete hm.mqtt**

Paquete contenedor de todas las clases relacionadas con la parte MQTT del programa. (Anexo 2, Ilustración 6).

**MQTTManager**, Singleton encargado de conectar con el bróker MQTT, suscribirse a los topics necesarios y publicar.

Clase **Interruptor**, es un Singleton que hace uso de la clase MQTTManager para la publicación MQTT en el topic correspondiente a un interruptor en el caso de que deba cambiar de estado tras la acción de un usuario en la aplicación. (Ejemplo: /hm/nodos/1/depositos/1/bomba Mensaje: {1})

**Starter**, Singleton que también hace uso de la clase MQTTManager, en este caso se encarga de suscribirse a todos los topics de la aplicación ("/hm/#"), utiliza un objeto de la clase MQTTListener para permanecer a la escucha de la llegada de nuevos mensajes a los topics suscritos y gestiona lo que se debe hacer en caso de tratarse de nuevos mensajes en los topics de registros de estado nodo o de depósito.

**MQTTListener** implementa MQTTCallback y por tanto es la clase encargada de gestionar lo que se debe hacer en caso de recibir un nuevo mensaje en los topics a los que se esté suscrito.

### **Paquete hm.control**

Contiene todos los servlets encargados de la lógica de aplicación con funciones como iniciar sesión, obtener registros, nodos, empresas... (Anexo 2, Ilustración 7)

Diagrama de clases tanto del sketch para el nodo emisor como del nodo receptor o puente con los atributos y funciones más relevantes. \*Explicación del código disponible en los apartados 6.2 *Prototipo Nodo Emisor* y 6.3 *Prototipo Nodo Puente*.

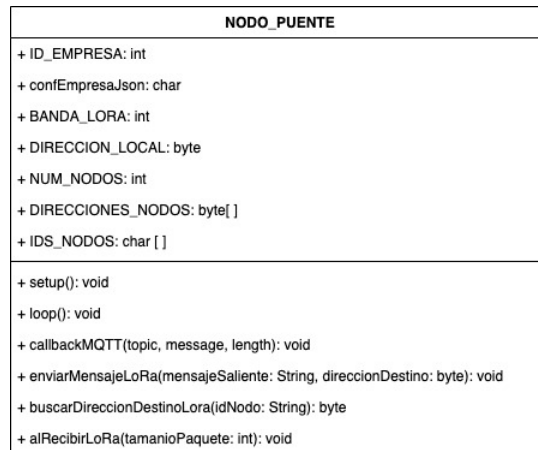


Ilustración 7: Diagrama Clases Sketch Puente.

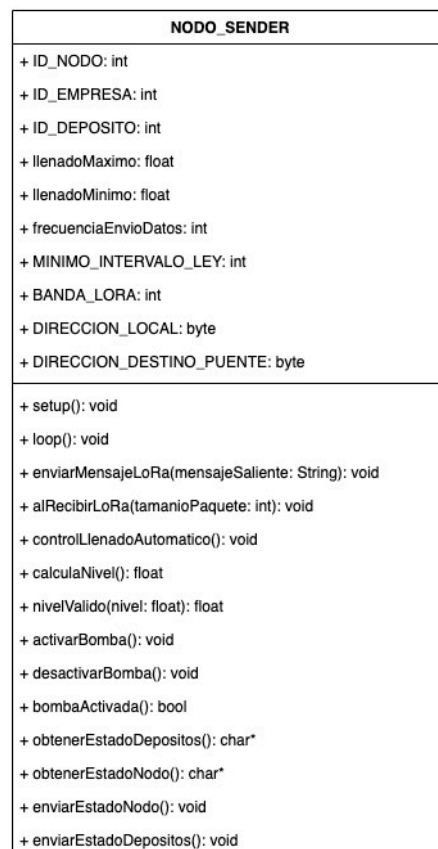


Ilustración 8: Diagrama Clases Sketch Emisor.

## 4.2.4 Base de Datos

### Diagrama Relacional

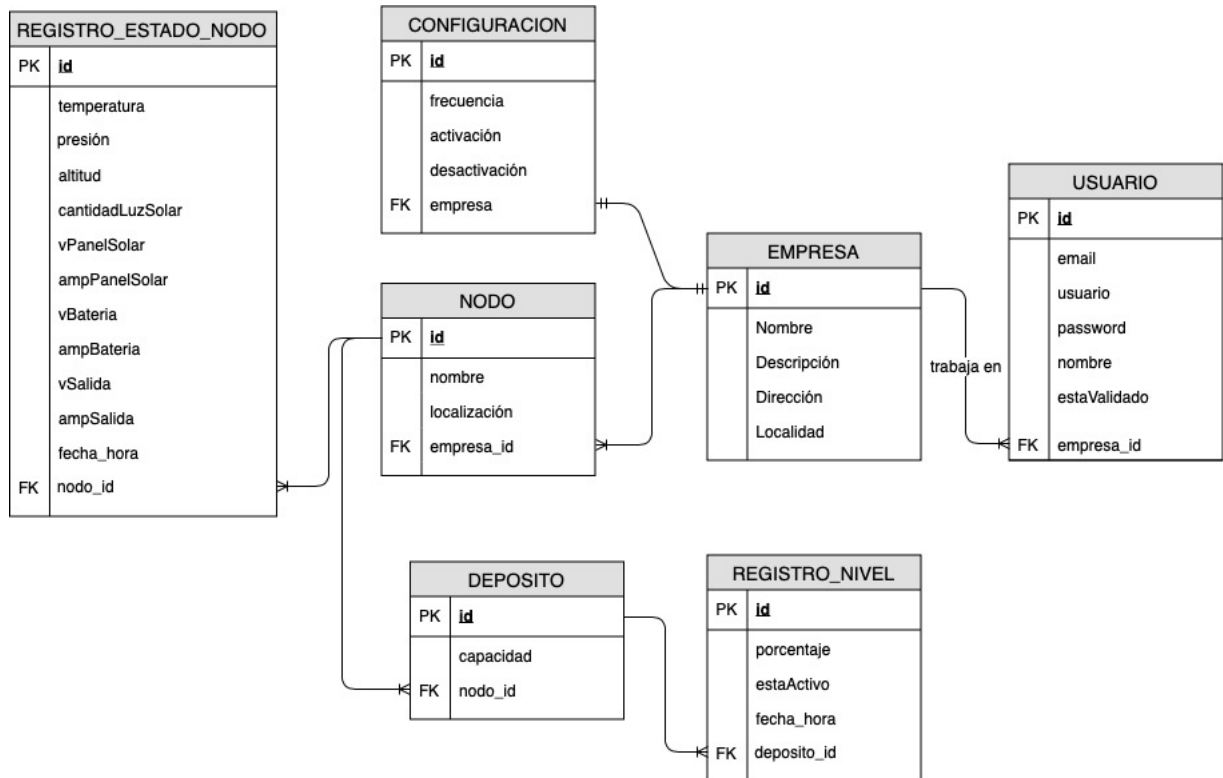


Ilustración 9: Diseño Relacional

## 4.3 DISEÑO HARDWARE

Apartado destinado a la definición y listado de todos los componentes hardware escogidos para la implementación del sistema.

### 4.3.1 Nodo Emisor

#### TTGO LORA32 868/915Mhz ESP32 con pantalla.

(Anexo 4, Ilustración 10.)

#### Descripción:

Placa basada en el popular chip **ESP32** de Espressif. El modulo ESP32 es compatible con Arduino y ofrece un chip híbrido Bluetooth/Wifi con conectividad inalámbrica 802.11 b/g/n que podemos utilizar de una manera muy sencilla.

Además, esta placa integra: una pantalla OLED de 0,96" que nos permitirá mostrar algunos datos importantes en el momento como pueden ser los mensajes emitidos o las lecturas de los sensores, y un módulo transceptor inalámbrico LoRa **SX1276** que como su nombre indica será el encargado de emitir y recibir los mensajes LoRa.

### Características:

- Chip ESP32.
- Bluetooth.
- Wifi 802.11 b/g/n.
- Posibilidad de alimentación por batería de litio.
- Conexión MicroUSB tanto para alimentación como para programación.
- Voltaje de funcionamiento: 3,3 V a 7V.
- Rango de temperatura: -40 ° C a + 90 ° C.
- Tarifas de datos: 150 Mbps @ 11n HT40., 72 Mbps 11 g 54 Mbps @, @ HT20 11n, 11 Mbps a 11b.
- Pantalla OLED 0,96'' 128x64 píxeles (SSD1306).
- Antena de 2dBi con rosca SMA.

**Precio:** 27,40€

**Link de compra:** [LILYGO TTGO LORA32 915Mhz - Banggood](#)

### Batería 12V

---

(Anexo 4, Ilustración 11)

### Descripción:

Batería de 12V y una capacidad de 6800mAh utilizada para la alimentación de todos los componentes del nodo emisor. Cuenta con 2 conexiones, una de entrada en la que podemos conectar el cargador incluido y otra de salida.

### Parámetros técnicos:

- Voltaje nominal: 12 V
- Voltaje de entrada: 12,6 V
- Voltaje de salida: 9 V-12,6 V
- Capacidad de la batería: 6800 mAh (6.8Ah)

Incluye cargador 12,6 V 1A.

**Precio:** 11,20€

**Link de compra:** [Batería 12V 6800mAh - Aliexpress](#)

### Panel Solar 12V

---

(Anexo 4, Ilustración 12)

### Descripción:

Panel Solar de 12V con una potencia máxima de carga de 5,5W. El panel incluye juego de pinzas.

### Características:

- Potencia máxima 5,5W.
- Voltaje en circuito abierto: 13,5V.
- Voltaje en funcionamiento: 12V.

- Corriente de cortocircuito 0,5 (A).
- Corriente de funcionamiento: 0 – 0,035(A).
- Dimensiones: 21 x 16,5 x 0,8 cm.

**Precio:** 13,03€

**Link de compra:** [Panel Solar 12V 5,5W - Aliexpress](#)

### Regulador de Carga

---

(Anexo 4, Ilustración 13)

#### Descripción:

Regulador de carga PWM de 10A. Un regulador de carga solar es el encargado de controlar el flujo energético que circula entre el campo fotovoltaico y las baterías de la instalación. Trabaja en función de la intensidad y voltaje a la que se inyecta energía en las baterías, por lo que se trata de un dispositivo preparado para controlar constantemente el nivel de carga de los acumuladores y optimizar la vida útil de los mismos.

En el mercado destacan los reguladores de carga MPPT y los reguladores de carga PWM.

Los reguladores MPPT maximizan su corriente a la salida a baterías, pues son capaces a adaptar la entrada fotovoltaica a la tensión de baterías. Esto le permite hacer funcionar a la placa en su punto de máxima potencia para obtener la máxima eficiencia de producción en cada instante, por ello hablamos de reguladores MPPT maximizadores.

Un regulador PWM únicamente puede ser utilizado si la tensión de las placas solares y las baterías es la misma, se caracterizan por tener un precio inferior a los reguladores MPPT y es por ello que esta será la tipología escogida para la construcción del prototipo.

#### Características:

- Voltaje de Batería: 12/24V automático.
- Corriente de carga: 10A.
- Entrada máxima: 50V (Baterías de 24V) o 25V (Baterías de 12V).
- Salida USB 3A.
- Consumo: <10mA.

**Precio:** 5,43€

**Link de compra:** [Regulador PWM 10A - Aliexpress](#)

### Módulo Relé de 4 Canales

---

(Anexo 4, Ilustración 14)

#### Descripción:

Módulo formado por 4 relés (4 canales independientes) especial para prototipado rápido con Arduino u otros controladores. Muy fácil de conectar y programar, sin necesidad de librerías específicas de control, sólo necesita el voltaje de salida de un pin (TTL- 5 V) de una placa Arduino o similar para controlar su funcionamiento. Con LED indicador de estado.

#### Características:

- 4 canales.
- Entrada 5V.
- Tensión máxima en AC: 250V/10A por canal.
- Tensión máxima en DC: 30V/10A por canal.

**Precio:** 2,21€

**Link de compra:** [Relé 5V 4 Canales - Banggood](#)

#### Bomba de Agua

---

(Anexo 4, Ilustración 15)

#### Descripción:

Mini bomba de agua sumergible ultra silenciosa y de bajo consumo con una potencia de 240l/hora y unas dimensiones reducidas que la hacen perfecta para el prototipo.

#### Características:

- Flujo de 240l / hora.
- Tensión nominal: DC12V.
- Corriente nominal: 375MA
- Consumo: 4,2W.
- Diámetro de entrada y salida 8mm.
- Altura máxima 300cm.
- Temperatura máxima del líquido 60°C
- Tamaño: 54 (L) x37 (W) x42 (H) mm
- Vida útil: 20000 horas
- Ruido: ≤40DB
- A prueba de agua: IP68
- Resistencia al calor: 0-100 °

**Precio:** 4,10€

**Link de compra:** [Bomba de agua sumergible - Aliexpress](#)

#### Sensor de BMP280

---

(Anexo 4, Ilustración 16)

#### Descripción:

Sensor de temperatura y presión barométrica BMP280 de alta precisión y de bajo consumo de energía. Basado en tecnología BOSCH piezo-resistiva con gran robustez EMC, alta precisión y linealidad, así como con estabilidad a largo plazo. Se ha diseñado para ser conectado directamente a un microcontrolador a través de I2C o SPI. El BMP280 es la evolución del BMP180, presenta mejoras en la resolución de presión y temperatura, es más preciso y cuenta con un menor consumo de energía. Es un sensor muy utilizado en Drones y sistemas de monitorización de clima o salud/fitness.

### Características:

- Voltaje de Operación: 1.8V - 3.3V DC.
- Interfaz de comunicación: I2C o SPI (3.3V).
- Presión:
  - Rango operativo: 300 a 1100 hPa (0.3-1.1bar).
  - Resolución: 0.16 Pa.
  - Precisión absoluta: 1 hPa.
  - Rango de altura medible: 0-9100 metros.
  - Precisión absoluta relativa de  $\pm 0.12$  hPa.
- Temperatura:
  - Temperatura de funcionamiento de -40 °C a +85 °C.
  - Resolución: 0.01°C.
  - Precisión: 1°C.
- Frecuencia de Muestreo: 157 Hz (máx.)
- Tiempo de medición promedio de 5.5 milisegundos

**Precio:** 1€

**Link de compra:** [BMP 280 - Ebay](#)

---

### Sensor de Corriente INA 219

(Anexo 4, Ilustración 17)

### Descripción:

El INA219 es un sensor fabricado por Texas Instruments que permite realizar medidas de tensión, intensidad y potencia en circuitos electrónicos.

El INA219 puede medir tensiones de 0V a 26V, pudiendo elegir entre dos escalas de 16V y 32V, con una precisión máxima de  $\pm 0.5$ .

La ganancia de la medición corriente es ajustable por software, lo que influye en el rango y precisión que podemos obtener. Al mínimo, puede medir corrientes de hasta  $\pm 3.2$ A con una precisión de 0.8mA. Al máximo, un rango de  $\pm 400$ mA con precisión de 0.1mA.

La alimentación del módulo es compatible con 3V3 y 5V. La comunicación se realiza a través de I2C, por lo que es muy sencillo conectarlo a un procesador como Arduino.

### Características:

- Alimentación: 3,3 y 5V DC.
- Medición de tensión: 0 - 26V.
- Medición de corriente:  $\pm 3.2$ A
- Interfaz de comunicación: I2C.

**Precio:** 1,08€

**Link de compra:** [Sensor - INA 219 - Aliexpress](#)

## Sensor Ultrasonidos HC – SR04

---

(Anexo 4, Ilustración 18)

### Descripción:

El sensor HC-SR04 ultrasonidos para determinar la distancia de un objeto en un rango de 2 a 450 cm. Destaca por su pequeño tamaño, bajo consumo energético, buena precisión y excelente precio. Es el más utilizado dentro de los sensores de tipo ultrasonido debido a la cantidad de información y proyectos disponibles en la web.

El sensor HC-SR04 posee dos transductores: un emisor y un receptor piezoeléctricos, además de la electrónica necesaria para su operación.

El funcionamiento del sensor es el siguiente: el emisor piezoeléctrico emite 8 pulsos de ultrasonido(40KHz) luego de recibir la orden en el pin TRIG, las ondas de sonido viajan en el aire y rebotan al encontrar un objeto, el sonido de rebote es detectado por el receptor piezoeléctrico, luego el pin ECHO cambia a Alto (5V) por un tiempo igual al que demoró la onda desde que fue emitida hasta que fue detectada, el tiempo del pulso ECO es medido por el microcontrolador y así se puede calcular la distancia al objeto. El funcionamiento del sensor no se ve afectado por la luz solar o material de color negro (aunque los materiales blandos acústicamente como tela o lana pueden llegar a ser difíciles de detectar).

### Características:

- Alimentación: 5V DC.
- Frecuencia de trabajo: 40KHz.
- Consumo (suspendido): <2mA.
- Consumo (trabajo): 15mA.
- Ángulo efectivo: < 15°.
- Distancia: 2cm a 400cm.
- Resolución: 3mm.

**Precio:** 1,10€

**Link de compra:** [Sensor ultrasónico distancia - Aliexpress](#)

## Sensor de Luz Solar

---

(Anexo 4, Ilustración 19)

### Descripción:

Sensor LDR o fotorresistencia, es un resistor cuya resistencia disminuye al aumentar la intensidad de luz que incide sobre él, dicho resistor está hecho de sulfuro de Cadmio. La salida será positiva siempre que la intensidad de luz supere el umbral configurado. El módulo tiene dos leds, uno nos indica si el dispositivo está alimentado y el segundo nos revela la señal de salida. Este tipo de sensores son muy utilizados para instalaciones de jardinería y señalización en calzadas.

### Características:

- Voltaje de funcionamiento: 3,3 – 5 V.
- Salida Digital: 0|1.
- Dimensiones: 3,2cm x 1,4cm.

- Chip comprobador de voltaje LM393.

**Precio:** 1,84€

**Link de compra:** [Sensor LDR - Aliexpress](#)

### 4.3.2 Nodo Puente

Para construir un nodo puente sólo se requiere un componente hardware, la placa Wemos TTGO LORA32 868/915Mhz ESP32 con pantalla ya descrita en el apartado anterior.

Dicha placa ya cuenta con un chip híbrido Bluetooth/Wifi con conectividad inalámbrica 802.11 b/g/n y un módulo transceptor inalámbrico LoRa **SX1276** que como su nombre indica será el encargado de emitir y recibir los mensajes LoRa.

Será alimentada por energía eléctrica y por tanto no será necesaria ninguna batería adicional.

## 5. IMPLEMENTACIÓN

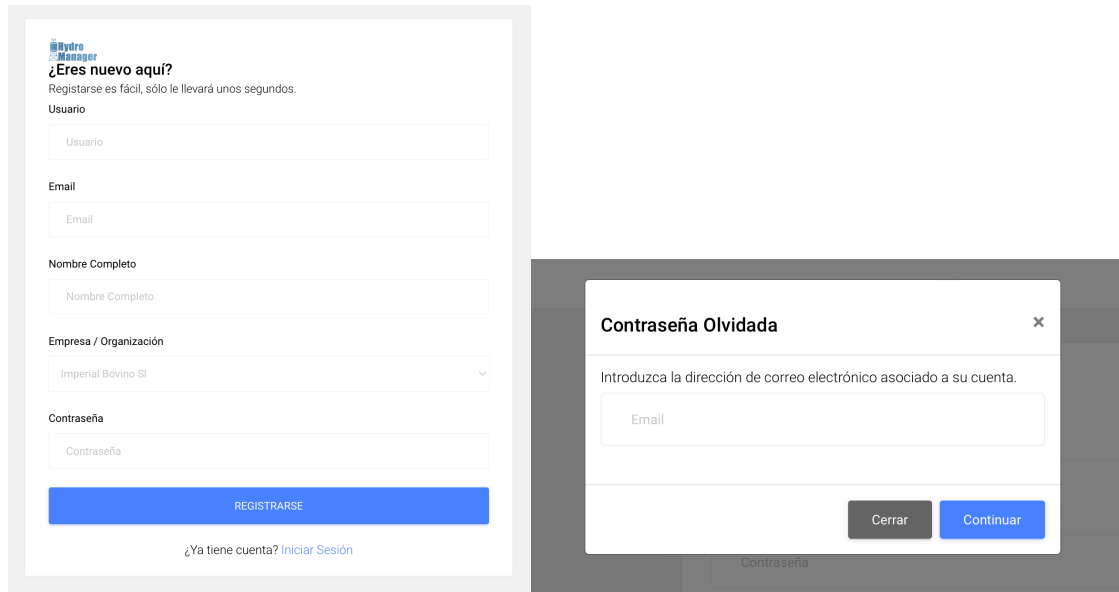
### 5.1 APLICACIÓN WEB “HYDRO MANAGER”

La primera página a la que accede el usuario por defecto será la de Inicio de sesión, en dicho formulario podrá acceder al formulario de registro, recuperar la contraseña en caso de haberla olvidado o por supuesto iniciar sesión introduciendo nombre de usuario o correo y contraseña.

**Ilustración 10: Página de Login.**

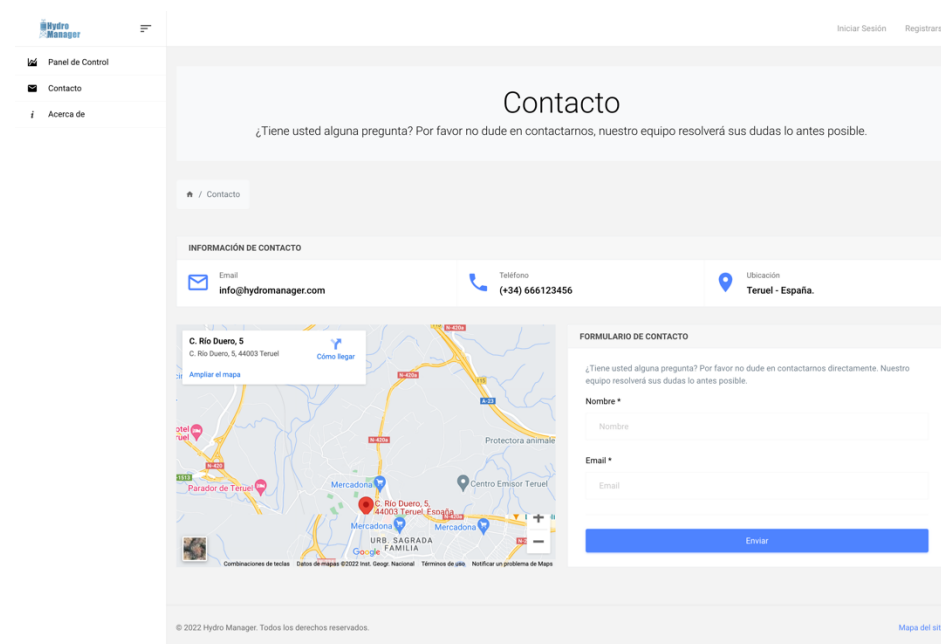
Los usuarios podrán registrarse con el rol “propietario” a través del formulario indicando en él los datos solicitados y seleccionando su empresa dentro del listado de empresas disponibles, estos usuarios quedarán pendientes de validación por parte del administrador y hasta entonces no podrán tener acceso.

En caso de no recordar la contraseña podrá recuperarse introduciendo la dirección de correo de su cuenta, esto invocará al Servlet PassOlvidada encargado de comprobar que el email introducido existe, en caso de ser así se genera una nueva contraseña aleatoria de 10 caracteres y la envía por email a la dirección del usuario.

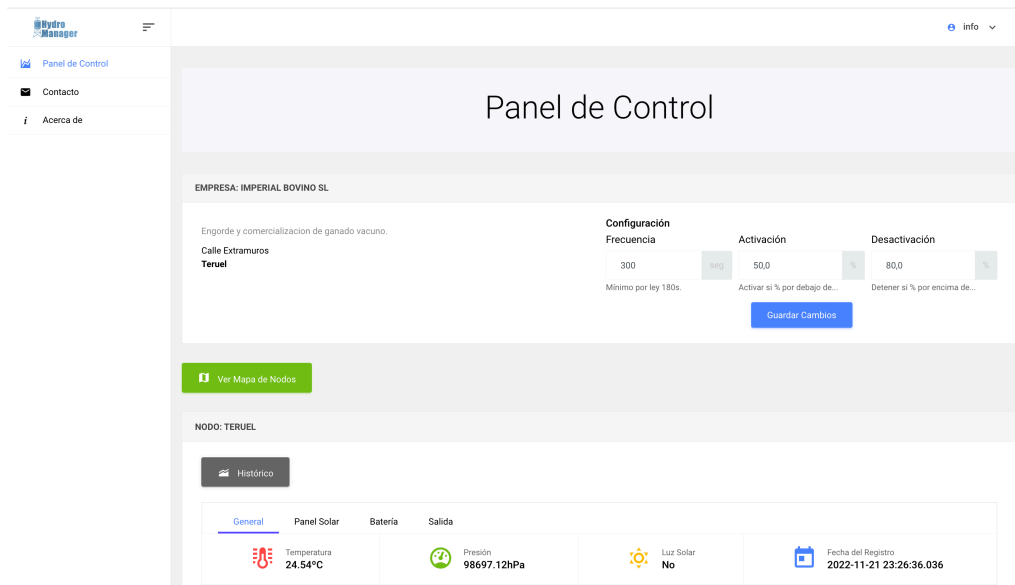


**Ilustración 11: Formularios de registro y de contraseña olvidada.**

La página de contacto y “Acerca de” son accesibles sin la necesidad de iniciar sesión. En la página de contacto se encuentra un formulario con el que el usuario podrá enviar una duda al correo de contacto de HydroManager [info.hydromanager@gmail.com](mailto:info.hydromanager@gmail.com), dicho formulario es totalmente funcional con el uso del servlet EnviarEmailContacto.

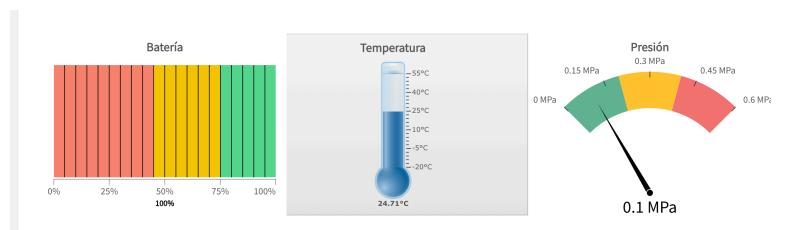


**Ilustración 12: Página de contacto.**



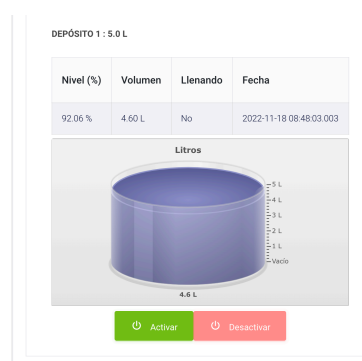
**Ilustración 13: Panel de control**

Una vez el usuario inicia sesión ya consigue acceso al panel de control con información de su empresa, su configuración actual y la posibilidad de actualizarla, el listado de nodos con su estado actual y todos sus depósitos. Esta página también mostrará un mapa de google con la ubicación de cada uno de los nodos.



**Ilustración 14: Muestra de estado de nodo y gráficas**

Como se puede apreciar se muestra toda la información recibida del nodo tanto de manera numérica como con gráficas para que sea mucho más visible. Toda esta información se actualiza automáticamente en cuanto se reciban nuevos datos sin necesidad de recargar la página.



**Ilustración 15: Estado actual de depósito**

En el caso de los depósitos se muestra cada uno de ellos con su estado, gráfica y la posibilidad de activar o desactivar la bomba de manera manual.

Con cada nodo se mostrará un botón “Histórico” que cargará el histórico del nodo con dicha id.

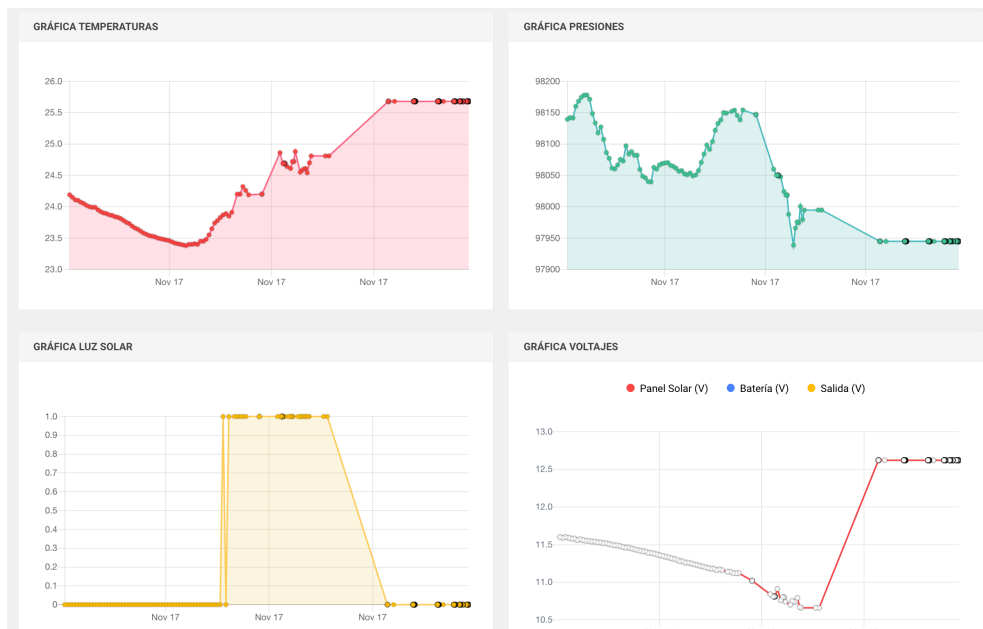
Por defecto se mostrarán gráficas y tablas con los datos de los últimos 50 registros obtenidos, pero se pueden obtener más datos a través del buscador, permite buscar por cantidad de registros recientes (50 últimos, 100, 150, 200, 250, 300, registros de este mes, mes pasado) u obtener todos los registros de una fecha en concreto. Como sucede en el caso de la página principal aquí también se actualizarán las gráficas y tablas con la llegada de un nuevo registro en el caso de estar mostrando los registros más recientes o los de la fecha actual.

🏠 / Nodo: 1 / Histórico

**BUSCADOR:**  
Mostrando gráficas en base a todos los registros del día 2022-11-15.

Por nº de registros: Últimos 50  Por fecha: 15/11/2022

**Ilustración 16: Buscador histórico**



**Ilustración 17: Gráficas histórico**

Mostrar	Entradas								
10	Entradas								
°C	Presión	Luz Solar	Panel (V)	Panel (A)	Batería (V)	Batería (A)	Salida (V)	Salida (A)	Fecha
25.21	98428.19	Si	1.02	-0.0001	1.02	-0.0001	1.02	-0.0001	2022-11-15 21:45:41.041
25.21	98428.19	Si	1.02	-0.0001	1.02	-0.0001	1.02	-0.0001	2022-11-15 21:45:35.035
25.21	98428.19	Si	1.02	-0.0001	1.02	-0.0001	1.02	-0.0001	2022-11-15 21:45:30.030
25.21	98428.19	Si	1.02	-0.0001	1.02	-0.0001	1.02	-0.0001	2022-11-15 21:45:24.024
25.21	98428.19	Si	1.02	-0.0001	1.02	-0.0001	1.02	-0.0001	2022-11-15 21:45:19.019
25.21	98428.19	Si	1.02	-0.0001	1.02	-0.0001	1.02	-0.0001	2022-11-15 21:45:14.014
25.21	98428.19	Si	1.02	-0.0001	1.02	-0.0001	1.02	-0.0001	2022-11-15 21:45:10.010
25.21	98428.19	Si	1.02	-0.0001	1.02	-0.0001	1.02	-0.0001	2022-11-15 21:45:04.004
25.21	98428.19	Si	1.02	-0.0001	1.02	-0.0001	1.02	-0.0001	2022-11-15 21:44:59.059
25.21	98428.19	Si	1.02	-0.0001	1.02	-0.0001	1.02	-0.0001	2022-11-15 21:44:54.054

Mostrando 1 a 10 de 437 Entradas

Anterior 1 2 3 4 5 ... 44 Siguiente

Mostrar	Entradas		
10	Entradas		
Nivel (%)	Volumen	Llenando	Fecha
90.24	4.51	No	2022-11-15 21:38:51.051
90.24	4.51	No	2022-11-15 21:37:58.058
90.24	4.51	No	2022-11-15 21:13:56.056
83.56	4.18	No	2022-11-15 21:07:55.055
83.44	4.17	No	2022-11-15 21:01:55.055
83.44	4.17	No	2022-11-15 20:55:54.054
83.56	4.18	No	2022-11-15 20:49:53.053
83.56	4.18	No	2022-11-15 20:43:53.053
85.26	4.26	No	2022-11-15 20:37:52.052
85.26	4.26	No	2022-11-15 20:31:51.051

Mostrando 1 a 10 de 338 Entradas

Anterior 1 2 3 4 5 ... 34 Siguiente

**Ilustración 18: Tablas histórico**

## 6. PROTOTIPO - DEMOSTRADOR

### 6.1 PRUEBAS DE COBERTURA LORA

Para la realización de esta prueba se utilizan las dos placas TTGO LoRa32 utilizadas para la construcción del prototipo, una de ellas hará de emisor y la otra de puente.

#### Emisor:

Será el encargado de generar y transmitir vía LoRa mensajes de prueba cada 15 segundos con el mensaje “Hola LoRa”, y un número que irá aumentando.

#### Nodo Puente:

Recibe dichos mensajes por LoRa, se conecta a internet vía WIFI y los publica por MQTT en el servidor de la universidad (IP: 155.210.68.154), en el topic: /stw/act/lora/msg.

El formato de la publicación es el siguiente:

```
"String pubString = "{Msg: "+ incoming + ",ID:" + incomingMsgId + ", RSSI:" + LoRa.packetRssi() + "}";
```

De esta manera podremos ver el mensaje ("Hola Lora nº"), el ID y la cobertura RSSI (A mayor valor peor cobertura).

#### Visionado de mensajes

Para ver los mensajes recibidos se utiliza la APP Android MQTT Dash, esta aplicación nos permite conectarnos al bróker MQTT instalado en el servidor de la universidad y subscribirnos al Topic utilizado, de manera que podemos ver el último mensaje recibido y la hora de recepción desde el móvil.

Cuando ya se tienen todos los componentes configurados y funcionando correctamente se procede a realizar la prueba.

#### Preparación

Para ello se coloca el receptor o puente en una ventana de la segunda planta de la vivienda y se conecta a la red eléctrica y al wifi del domicilio.

Por otra parte, el emisor irá conectado a un powerbank para su alimentación y será con este dispositivo con el que nos iremos moviendo ya que no requiere conexión a internet.

#### Resultados

A medida que nos alejamos vemos como aumenta el RSSI. Debemos ir fijándonos en que el contador y el id son números consecutivos y que se actualiza el mensaje cada 15 segundos. Teniendo todo esto en cuenta todo funciona correctamente hasta llegar al punto 1.

Como se puede ver en la imagen (Anexo 5, Ilustración 20) al seguir caminando hay varias viviendas que interfieren en la señal y es en este momento donde se pierde la señal o se ven mensajes erróneos. La distancia alcanzada hasta este punto en línea recta son **489,93m**.

(Anexo 5, Ilustración 21) Sigo avanzando por el camino y llego a un punto más alejado **594,11m en el** que no interfieren los edificios y la señal vuelve a ser correcta.

Sigo andando por los caminos próximos a la zona, en algunos momentos se pierde la conexión, pero al llegar al punto 3, una zona elevada, vuelven a recibirse mensajes con números consecutivos y con el formato adecuado alcanzando así un máximo de **651,35m** de distancia en línea recta en campo abierto. (Anexo 5, Ilustración 22)

## 6.2 PROTOTIPO NODO EMISOR

El prototipo físico construido se encargará de simular a pequeña escala la estructura de un nodo con todos los sensores y un depósito, estará dividido de la siguiente manera:

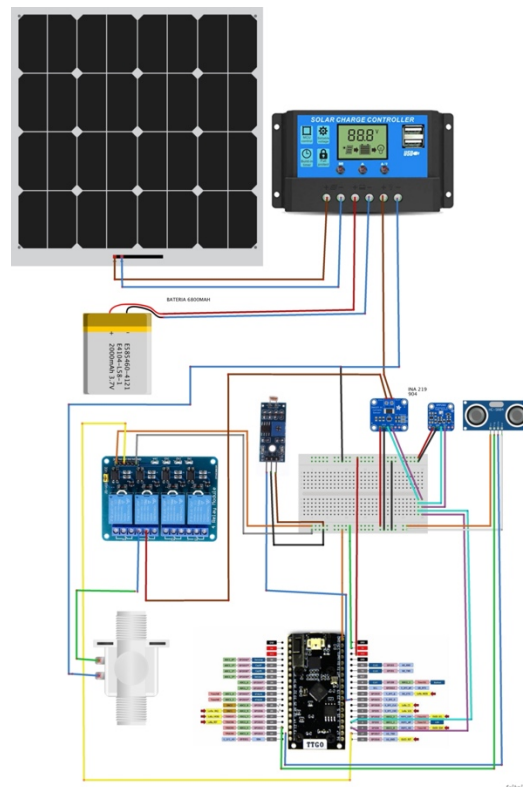


Ilustración 19: Diagrama del cableado.

\*Diagrama a mayor escala disponible en (Anexo 5, Ilustración 23)

Para conectar los elementos del prototipo se utilizan diferentes tipos de cables, una protoboard y conectores para facilitar la conexión o desconexión de la batería, la bomba de agua y el panel solar.

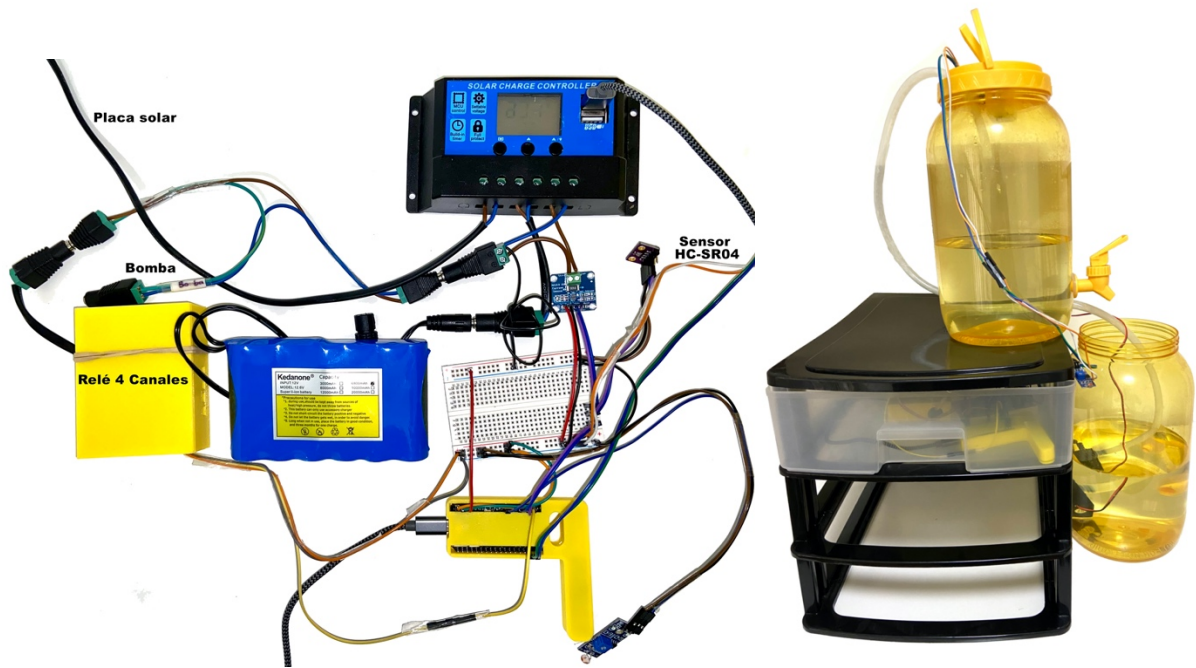


Ilustración 20: Prototipo emisor.

Imágenes a mayor escala en (Anexo 5, Ilustración 25, 26 y 27)

### 6.2.1 Instalación de Energía Fotovoltaica:

Por un lado, se encuentra la instalación de energía fotovoltaica compuesta por un panel solar de 12v, una batería de 12v y 6800mAh y el regulador de carga PWM que cuenta con 2 puertos USB de 5V que nos serán útiles para la alimentación de la placa TTGO del emisor e incluso del puente si deseamos probarlos en la misma ubicación.

Al utilizar un regulador de carga PWM todo el circuito formado por placa, batería y salida comparten la misma tensión e intensidad y por tanto con colocar un sensor de voltaje será suficiente, para poder medir en los 3 puntos en este caso sería necesario cerrar el circuito en cada punto y leer.

En un caso real, con dispositivos más grandes, voltajes distintos entre las placas y las baterías y regulador de carga MPPT sí podremos leer la diferente corriente en los 3 puntos.

### 6.2.2 Sensores:

En cuanto a sensores contará con cuatro en total, los mismos comentados durante la fase de diseño hardware, dos de ellos, el sensor de temperatura BMP280 y el sensor de voltaje INA219 utilizan interfaz I2C mientras que los otros dos no.

En cuanto a alimentación, el sensor de temperatura se alimentará del puerto de 3.3V de la placa y el resto lo harán del de 5V.

Se decide colocar el sensor de voltaje en la salida del regulador, como ya se ha comentado podría colocarse en cualquiera de los 3 puntos del circuito de 12v.

### 6.2.3 Depósitos

---

Para la simulación del llenado del depósito se utilizan dos dispensadores con grifo, no representará el pozo del que se extrae el agua instalando en él la bomba de agua con una manguera y el otro hará las funciones del depósito principal del que leemos el volumen actual colocando el sensor de distancia en su tapa.

Para vaciar el depósito principal se utilizará el grifo que incluye vertiendo el agua en el pozo de nuevo.



Ilustración 21: Dispensadores y Sensor volumen tapa.

### 6.2.4 Programación desarrollada

---

Se desarrolla en Arduino el sketch llamado HydroM\_Nodo\_Sender y se carga en la placa del emisor.

El programa contará con parámetros de configuración que se utilizarán a lo largo del programa para facilitar la modificación de este con simplemente revisar las primeras líneas.

Estos parámetros permitirán modificar, por ejemplo, la id del nodo, id de empresa, los sensores activos o la activación de debug (mostrará más mensajes por el puerto serie).

Tras cargar las librerías e inicializar las variables necesarias se lanza la función de setup().

#### Setup()

---

En ella se inicia el puerto serie, los pines a usar, se activa LoRa, los sensores, el relé (y se desactiva) y por último se envía un mensaje LoRa de solicitud de configuración de la empresa hacia el nodo puente.

Si se completa toda la configuración correctamente se inicia la función loop().

#### Loop()

---

Esta función se ejecuta un número infinito de veces y por tanto contiene el código que se ejecutará continuamente.

En este punto el nodo emisor ya habrá solicitado la configuración actual de la empresa compuesta de la frecuencia de envío de datos, el valor para activar el llenado y el valor máximo para desactivarlo, pero hasta la recepción del mensaje LoRa contará con una configuración por defecto para poder funcionar de manera autónoma en caso de no conectar con el puente durante el arranque o tardar éste en responder.

El trabajo que realiza la función loop se divide en 3 funciones.

1. Control de llenado automático del depósito.
2. Envío periódico de estado de nodo o de depósito.
3. Escucha de la llegada de mensajes LoRa.

### Control de llenado automático del depósito.

Se encarga de llevar la lógica de control del nivel de depósito para gestionar el llenado automático y parado de la bomba si supera el nivel configurado o no bombea agua después de varios intentos.

El programa cuenta con el parámetro de configuración TIEMPO\_LECTURA\_DEPOSITO que permite ajustar cada cuanto tiempo se leerá el volumen del depósito, si el nivel leído es válido continúa con la lógica, si no lo es y el relé está activo, lo desactiva por seguridad y envía el estado actual al puente vía LoRa.

Para calcular el nivel o porcentaje de llenado del depósito se utiliza la lectura del sensor de ultrasonidos obteniendo la distancia del sensor al nivel del agua, conociendo además por configuración cual sería la distancia en caso de estar lleno y cual la del 0%.

En caso de leerse un nivel válido (valor comprendido entre 0 y 100%) se comprueba el estado de la bomba, si está desactivada y el depósito está por debajo del nivel mínimo configurado para activar el llenado (llenadoMinimo) se activa la bomba y se envía este nuevo estado de depósito al puente.

En caso de haber leído nivel válido y tener el relé desactivado se comprueba si el nivel actual del depósito ya ha llegado al nivel máximo deseado y en caso de ser así se desactiva la bomba y se envía el nuevo estado de depósito, más adelante se explicará como se realiza el envío de estados.

Para evitar que la bomba permanezca activa tratando de extraer agua de un “pozo vacío” o con el nivel por debajo de su entrada, se añade una lógica de seguridad, si la bomba está activa y el nivel actual es igual que el anterior se anota un intento de llenado y en caso de llegar a un máximo configurado llamado INTENTOS\_MAXIMOS, se desactiva la bomba por seguridad y se envía ese cambio de estado al puente. Para que no se produzcan más reintentos hasta pasado un tiempo (TIEMPO\_ESPERA\_REINTENTO\_LLENADO), se almacena ese momento actual y se comprobará antes de volver a activar la bomba automáticamente de nuevo.

### Envío periódico de estado de nodo o de depósito.

Es en esta función donde se gestiona el envío periódico, según el intervalo configurado para cada empresa (frecuenciaEnvioDatos) del estado del nodo y del depósito, y por tanto es aquí donde se originan gran parte de los registros que luego acabarán almacenándose en la base de datos y mostrándose en la aplicación web.

Para limitar el uso del canal se irá alternando entre los dos registros de estado existentes, si el último enviado es de tipo “estadoNodo” el siguiente será de depósito.

Para la generación del estado del depósito se realiza la lectura del estado actual del Pin asignado al relé y del % de volumen actual con los datos obtenidos por el sensor de ultrasonidos, con estos datos se genera el mensaje con estructura JSON válida y se envía por LoRa al nodo Puente (Ejemplo: {“tm”:2,“n”:1,“e”:1,“d”:1,“nivel”:53,“estaActivo”:1} ).

Para la construcción del estado del nodo se realizan lecturas en los otros tres sensores, de presión y temperatura en el sensor con el BMP280, de voltaje y amperaje con el sensor INA 219 y de presencia de luz solar con el sensor LDR. Una vez se han obtenido todas las lecturas se genera el mensaje similar a JSON y se envía por LoRa al puente. Como ya se ha comentado, el sensor INA219 asignará sus valores leídos a batería, placa solar y salida.

(Ejemplo: {"tm":1,"n":1,"e":1,"t":22.10,"p":109805.4,"ls":1,"vp":12.62,"ap":0.30,"vb":12.62,"ab":0.30,"vs":12.62,"as":0.30})

### Escucha de la llegada de mensajes LoRa.

Parte encargada de la escucha de la posible llegada de mensajes LoRa al nodo emisor.

Tanto para la recepción como para el envío se hace uso de la librería LoRa.h.

Se lee el receptor, emisor, id del mensaje de entrada, longitud del mensaje y por último el contenido completo del mensaje.

Tras leer toda la información del mensaje se comprueba si la longitud es correcta, el destino es la dirección LoRa del nodo o se trata de un mensaje broadcast y el formato del mensaje es un JSON deserializable.

Si se cumple todo esto se deserializa el JSON recibido para obtener así el tipo de mensaje, la empresa y la id del nodo destino, de esta forma nos aseguramos una vez más de que el mensaje es aceptable para este nodo comparándolo con los parámetros ID\_EMPRESA e ID\_NODO configurados.

El nodo emisor podrá recibir mensajes de dos tipos, tipo relé (4) y tipo configuración (3). Si el mensaje recibido es de tipo relé se procederá a actualizar el estado del relé que activa la bomba y en caso de ser mensaje de tipo configuración se actualizarán los parámetros de configuración de empresa (llenadoMaximo, llenadoMinimo y frecuenciaEnvioDatos).

Ejemplo de mensajes: {"tm":3,"e":1,"frecuencia":30,"activacion":30,"desactivacion":90} y {"tm":4,"n":1,"e":1,"d":1,"activarRelé":1}

## 6.3 PROTOTIPO NODO PUENTE

Como ya se comenta en el diseño hardware este prototipo sólo requerirá de la propia placa y la instalación del programa desarrollado para que haga sus funciones.

La placa ya cuenta con antena Wifi y LoRa que permitirá tanto recibir los mensajes del emisor o emisores como suscribirse al bróker MQTT.

Este nodo ya estaría instalado en un punto con internet y energía eléctrica, por tanto, para las pruebas podemos conectarlo a la propia red eléctrica doméstica.

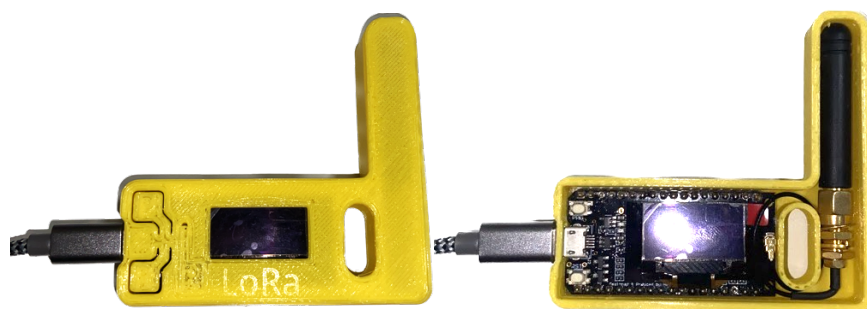


Ilustración 22: Wemos TTGO ESP32 Puente.

### 6.3.1 Programación Desarrollada

Se desarrolla en Arduino el sketch llamado HydroM\_Puennte y se carga en la placa del receptor o puente.

El programa, al igual que en el emisor, contará con parámetros de configuración para facilitar la modificación de este con simplemente revisar las primeras líneas.

Estos parámetros permitirán modificar, entre otros, la id de empresa, el modo debug, el listado de nodos y sus direcciones LoRa. A pesar de que actualmente sólo existirá un nodo emisor ya se añade la lógica para almacenar las direcciones LoRa de varios nodos y escoger a cuál de ellos se envía el mensaje.

Tras cargar las librerías e inicializar las variables necesarias se lanza la función de `setup()`.

## Setup()

---

Se inicia el puerto serie, los pines a usar, se activa LoRa, se realiza la conexión WiFi y MQTT y se suscribe a todos los topics de la empresa ("/hm/1/#") y se asigna como callback la función `callbackMQTT` para gestionar como debe actual el nodo puente en caso de recibir algún mensaje nuevo en alguno de los topics.

Si se completa toda la configuración con éxito se inicia la función `loop()`.

## Loop()

---

Contiene el código que se ejecutará en bucle infinitamente.

La función del puente es permanecer a la escucha de la recepción de mensajes LoRa o mensajes MQTT en los topics a los que se haya suscrito.

### Escucha de la llegada de mensajes LoRa.

---

Tanto para la recepción como para en envío se hace uso de la librería `LoRa.h`.

Como sucede en el emisor, cuando llega un mensaje LoRa se lee el receptor, emisor, id del mensaje de entrada, longitud del mensaje y por último el contenido completo del mensaje.

Tras leer toda la información del mensaje se comprueba si la longitud es correcta, el destino es la dirección LoRa del nodo o se trata de un mensaje broadcast y el formato del mensaje es un JSON deserializable, es importante ya que los mensajes utilizados por el programa y el servidor de aplicaciones siempre deberán ser JSON.

Si se cumple todo esto se deserializa el JSON recibido para obtener así el tipo de mensaje y la empresa, utilizaremos la id de empresa para asegurarnos de que el mensaje es aceptable para este nodo puente comparándolo con el parámetro `ID_EMPRESA`.

Se podrán recibir mensajes de 3 tipos por LoRa, estado nodo (1), estado depósito (2) y tipo configuración (3).

Si el mensaje recibido es un estado de nodo se genera el string con el topic MQTT al que deberá ir destinado utilizando la id del nodo origen y la id de empresa a la que pertenecen ambos y se publica en él el mensaje completo recibido. El topic para este prototipo en el que hay sólo un nodo, su id es 1 y el id de empresa es 1 sería: `"/hm/1/nodos/1/estado"`.

Si el mensaje es un nuevo estado de depósito se hará algo muy similar al caso de estado nodo, se generará el string con el topic destino `"/hm/1/nodos/1/depositos/1/nivel"` y se publicará en él, pero además de esto se genera un segundo mensaje que sólo contendrá el valor del estado actual del relé, extrayendo del JSON el campo "estaActivo" que publicará en el topic de tipo bomba o relé `"/hm/1/nodos/1/depositos/1/bomba"`.

En el caso de recibir un mensaje LoRa de solicitud de configuración de un nodo emisor cuando arranque, lo que hacemos es obtener la dirección LoRa del nodo emisor y enviarle de vuelta un mensaje LoRa con la configuración actual de la empresa en formato JSON.

## Escucha de mensajes MQTT

Función `callbackMQTT()`. El nodo puente permanecerá suscrito a todos los topics con información relevante de su empresa, por tanto, recibirá mensajes de topics muy distintos y deberá realizar funciones distintas según el topic.

Lo primero que hará será dividir el topic en partes para conocer datos como el tipo de mensaje, el nodo, empresa, id de depósito, etc. Para el tipo utilizaremos la última parte del topic, los tipos que trataremos serán “bomba” y “configuracion”.

Si el mensaje recibido viene del topic “bomba” (“/hm/1/nodos/1/depósitos/1/bomba”) se tratará de un cambio en el estado del relé generado por la aplicación web a causa de la acción de un usuario sobre el botón Activar/Desactivar bomba y por tanto lo que se deberá hacer será enviar dicha acción al nodo destino para que cambie el estado de su relé. Se genera el mensaje en el formato necesario, se obtiene la dirección LoRa de destino y se envía al nodo emisor. Ejemplo de mensaje enviado al Sender: `{"tm":4,"n":1,"e":1,"d":1,"activarRele":1}`

Si el mensaje es en un topic de configuración (“/hm/1/nodos/1/configuracion”) se trata de una actualización en alguno de los parámetros de configuración de la empresa a la que pertenece el puente y esta modificación se habrá realizado a través de la aplicación web. Se toma el mensaje recibido en el topic, se deserializa, se añade el tipo de mensaje 3 (“tm”:3) para que el Sender lo comprenda y se envía dicho mensaje por LoRa a todos los nodos del listado de nodos. `{"tm":3,"e":1,"frecuencia":30,"activacion":30,"desactivacion":90}`

## 6.4 BROKER MQTT

Para realizar las funciones de bróker MQTT y que éste sea independiente del servidor de aplicaciones payara se opta por utilizar una Raspberry Pi 3 B+.

Se le configura una dirección IP fija, se instala el Bróker de código abierto Mosquitto y se configura para que se inicie dicho servicio durante el arranque, de esta manera cuando queramos iniciar las simulaciones sólo será necesario encender el dispositivo y esperar a que esté disponible.



Ilustración 23: Raspberry Pi 3 B+ Nodo Puente.

## 7. CONCLUSIONES

Tras finalizar el desarrollo del proyecto y la implementación del prototipo podemos concluir que:

1. El bajo coste de los elementos necesarios para realizar el prototipo lo convierte en una opción viable, accesible y fácilmente escalable para su traspaso a una red real de depósitos.
2. El auge del IoT sumado al bajo coste ya mencionado hace que existan grandes oportunidades de desarrollo de las técnicas y tecnologías utilizadas en el medio rural.
3. El protocolo LoRa satisface las necesidades más importantes del IoT para aplicaciones que requieran de largo alcance y muy baja potencia.
4. Aunque la llegada del 5G a todo el territorio español parezca inminente, la realidad es que gran parte del medio rural no espera su llegada a corto plazo, en muchos lugares todavía no ha llegado la generación anterior e incluso la cobertura es escasa.

## 8. BIBLIOGRAFÍA

- [1] A. Molina, «marketing4ecommerce,» 10 Diciembre 2015. [En línea]. Available: <https://marketing4ecommerce.net/el-mercado-iot-prospecta-su-crecimiento-hacia-el-2019/>. [Último acceso: Abril 2022].
- [2] «The New Now - IoT acelera su ritmo de crecimiento en todos los sectores,» Noviembre 2017. [En línea]. Available: <https://www.thenewnow.es/negocio/iot-acelera-ritmo-crecimiento/>. [Último acceso: Abril 2022].
- [3] M. Gracia, «IoT - Internet Of Things - Deloitte,» [En línea]. Available: <https://www2.deloitte.com/es/es/pages/technology/articles/loT-internet-of-things.html>. [Último acceso: Julio 2022].
- [4] «AlfaIoT - ¿Qué es Lora?,» [En línea]. Available: <https://www.alfaiot.com/blog/ultimas-noticias-1/post/que-es-lora-2>. [Último acceso: Abril 2022].
- [5] «Acciona - Energías Renovables,» Acciona, [En línea]. Available: <https://www.acciona.com/es/energias-renovables/energia-solar/fotovoltaica/>. [Último acceso: Abril 2022].
- [6] SEMTECH, «Semtech - What is LoRa?,» [En línea]. Available: <https://www.semtech.com/lora/what-is-lora>. [Último acceso: Abril 2022].
- [7] Wikipedia, «Wikipedia - Payara Server,» Marzo 2019. [En línea]. Available: [https://en.wikipedia.org/wiki/Payara\\_Server](https://en.wikipedia.org/wiki/Payara_Server). [Último acceso: Abril 2022].
- [8] Wikipedia, «Wikipedia - Java (lenguaje de programación),» 2019. [En línea]. Available: [https://es.wikipedia.org/wiki/Java\\_\(lenguaje\\_de\\_programación\)](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programación)). [Último acceso: Abril 2022].
- [9] Wikipedia, «Wikipedia - Servicio Web,» 2019. [En línea]. Available: [https://es.wikipedia.org/wiki/Servicio\\_web](https://es.wikipedia.org/wiki/Servicio_web). [Último acceso: Abril 2022].
- [10] J. D. M. González, «ProgramarYa - Curso de Java Avanzado,» Abril 2018. [En línea]. Available: <https://www.programarya.com/Cursos-Avanzados/Java/Sockets>. [Último acceso: Abril 2022].
- [11] TST, «TST - MQTT,» [En línea]. Available: <http://www.tst-sistemas.es/mqtt/>. [Último acceso: Abril 2022].
- [12] T. H. Team, «HiveMQ - MQTT Essentials Part 3.,» 26 Enero 2015. [En línea]. Available: <https://www.hivemq.com/blog/mqtt-essentials-part-3-client-broker-connection-establishment/>. [Último acceso: Abril 2022].
- [13] CIC, «¿Qué es el Internet de las Cosas y cuáles son sus aplicaciones?,» 14 Enero 2019. [En línea]. Available: <https://www.cic.es/iot-sus-aplicaciones/>. [Último acceso: Abril 2022].
- [14] Wikipedia, «Wikipedia - Mapa de Sitio Web,» 27 Septiembre 2017. [En línea]. Available: [https://es.wikipedia.org/wiki/Mapa\\_de\\_sitio\\_web](https://es.wikipedia.org/wiki/Mapa_de_sitio_web). [Último acceso: Abril 2022].

- [15] Y. FM, «Xataka - Qué es Arduino, cómo funciona y qué puedes hacer con uno,» 3 Agosto 2018. [En línea]. Available: <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>. [Último acceso: Abril 2022].
- [16] C. Krall, «Aprenderaprogramar - ¿Qué es y para qué sirve UML?,» [En línea]. Available: [https://www.aprenderaprogramar.com/index.php?option=com\\_content&view=article&id=688:i-que-es-y-para-que-sirve-uml-versiones-de-uml-lenguaje-unificado-de-modelado-tipos-de-diagramas-uml&catid=46&Itemid=163](https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=688:i-que-es-y-para-que-sirve-uml-versiones-de-uml-lenguaje-unificado-de-modelado-tipos-de-diagramas-uml&catid=46&Itemid=163). [Último acceso: Abril 2022].
- [17] 4 Febrero 2017. [En línea]. Available: <https://teslabem.com/nivel-intermedio/fundamentos/>. [Último acceso: 12 10 2022].
- [18] «JSON.org,» [En línea]. Available: <https://www.json.org/json-es.html>. [Último acceso: 22 11 2022].
- [19] «Refactoring Guru,» Noviembre 2022. [En línea]. Available: <https://refactoring.guru/es/design-patterns/singleton>. [Último acceso: Noviembre 2022].
- [20] IEEE, «IEEE 830-1998 - IEEE Recommended Practice for Software Requirements Specifications,» 2009. [En línea]. Available: <https://standards.ieee.org/standard/830-1998.html>. [Último acceso: Abril 2022].
- [21] W. W. C. A. G. (. 2.1, «W3C Web Content Accessibility Guidelines (WCAG) 2.1 W3C Recommendation 05 June 2018,» 2018. [En línea]. Available: <https://www.w3.org/TR/WCAG21/>. [Último acceso: Abril 2022].
- [22] D. BOE-A-2017-542, «BOE,» 2017. [En línea]. Available: [https://www.boe.es/diario\\_boe/txt.php?id=BOE-A-2017-542](https://www.boe.es/diario_boe/txt.php?id=BOE-A-2017-542). [Último acceso: Abril 2022].