

Sistema hidropónico a través de IoT

Hydroponic system through IoT

Trabajo Fin de Grado
INGENIERÍA INFORMÁTICA
2024 - 2025



VNiVERSIDAD
D SALAMANCA

Autor:

Javier Cabo Correa

Tutores:

Sergio García Sánchez
Alicia García Holgado

D. Sergio García Sánchez, profesor del Departamento de Física Aplicada, y Dña.
Alicia García Holgado, profesora del Departamento de Informática y Automática de la
Universidad de Salamanca

CERTIFICAN:

Que el trabajo titulado “Sistema hidropónico a través de IoT” ha sido realizado por D. Javier Cabo
Correa, con DNI ****9954D y constituye la memoria del trabajo realizado para la superación de la
asignatura Trabajo de Fin de Grado de la Titulación Grado en Ingeniería Informática de esta
Universidad.

Y para que así conste a todos los efectos oportunos.

En Salamanca, a 2 de julio de 2025

D. Sergio García Sánchez

Dpto. Física Aplicada

Universidad de Salamanca

Dña. Alicia García Holgado

Dpto. Informática y Automática

Universidad de Salamanca

Resumen

Este Trabajo de Fin de Grado presenta el diseño e implementación de un sistema hidropónico inteligente basado en tecnologías de Internet de las Cosas (IoT), con el objetivo de monitorizar y optimizar el cultivo sin suelo a través de sensores, automatización y una plataforma web de control. El sistema permite la medición en tiempo real de variables clave como la temperatura, el nivel de agua, la conductividad eléctrica (TDS) y el pH, a través de un microcontrolador Arduino. Estos datos son enviados mediante red local a un servidor que almacena la información en una base de datos MySQL, activa mecanismos automáticos (como bomba de agua) y genera alarmas cuando se detectan condiciones críticas.

Además, se ha desarrollado una plataforma web accesible desde cualquier navegador para visualizar los datos históricos, configurar parámetros críticos para cada sensor, gestionar usuarios y personalizar las condiciones de cultivo en función del tipo de planta. La aplicación está diseñada con tecnologías web como JavaScript y PHP, e integra funcionalidades como notificaciones automáticas por correo ante eventos relevantes.

El sistema busca facilitar la automatización y supervisión remota de cultivos hidropónicos, ofreciendo una solución modular, ampliable y orientada tanto a usuarios principiantes como avanzados, con especial atención a la sostenibilidad, el bajo coste y la facilidad de uso.

Palabras clave: Hidroponía, IoT, Arduino, automatización, sensores, agricultura inteligente, plataforma web.

Abstract

This Final Degree Project presents the design and implementation of an intelligent hydroponic system based on Internet of Things (IoT) technologies, aimed at monitoring and optimizing soilless cultivation through sensors, automation, and a web-based control platform. The system enables real-time measurement of key variables such as temperature, water level, electrical conductivity (TDS), and pH using an Arduino microcontroller. These data are transmitted over a local network to a server that stores the information in a MySQL database, triggers automatic mechanisms (such as water pump), and generates alerts when critical conditions are detected.

In addition, a web platform has been developed, accessible from any browser, to visualize historical data, configure sensor thresholds, manage users, and customize cultivation conditions based on plant type. The application is built using web technologies such as JavaScript, and PHP, and includes features like automated email notifications for relevant events.

The system aims to facilitate the automation and remote supervision of hydroponic crops, offering a modular, scalable solution targeted at both beginner and advanced users, with a focus on sustainability, low cost, and ease of use.

Keywords: Hydroponics, IoT, Arduino, automation, sensors, smart agriculture, web platform.

Contenido

1. Introducción	10
2. Análisis de mercado	11
3. Objetivos	12
3.1 Objetivo general	13
3.2 Objetivos específicos	13
3.3 Objetivos técnicos	13
3.4 Objetivos personales	14
4. Estado del arte y fundamentos teóricos	14
4.1 Introducción a la hidroponía	14
4.2 Evolución tecnológica en hidroponía: automatización e IoT	15
4.3 Unidades estándar y variables monitorizadas	15
4.4 Proyectos relacionados y soluciones actuales	15
4.4.1 Proyecto OpenLanuza	15
4.4.2 Sistema automatizado de Kyle Gabriel	16
4.4.3 Proyecto de Víctor Muñoz Rebollar (UPC)	16
4.4.4 Soluciones comerciales: NIDO PRO y Rocket24	16
4.5 Conclusiones del estado del arte	17
5. Normas y referencias	18
5.1 Metodología de desarrollo utilizada	18
5.2 Wireframes	19
5.3 Métricas	21
6. Restricciones y alcance	23
6.1 Restricciones	23
6.3 Alcance del proyecto	23
7. Herramientas y tecnologías utilizadas	24
7.1 Hardware	24
7.1.1 Arduino UNO R4 WiFi (WiFiS3)	24
7.1.2 Sensores de bajo coste (TDS, pH, temperatura, nivel de agua)	24
7.1.3 Pantalla LCD I2C (HD44780)	24
7.2 Software y tecnologías de desarrollo	25
7.2.1 Arduino IDE	25
7.2.2 Python	25
7.2.3 PHP	26
7.2.4 HTML	26
7.2.6 JavaScript	27
7.2.7 Bootstrap	27
7.2.8 MySQL	28
7.2.9 Visual Studio Code	28
7.3 Herramientas de diseño y modelado	29
7.3.1 MySQL Workbench	29
7.3.2 Visual Paradigm Online	29
7.3.3 Excalidraw	30

7.3.4 EZEstimate	30
7.3.5 Microsoft Visio	31
7.3.6 Microsoft Project	31
7.5 Herramientas de montaje y prototipado físico	32
7.5.1 Sierra de pelo	32
7.5.2 Taladro eléctrico	32
7.5.3 Papel de lija	32
7.5.4 Pistola de silicona	33
7.5.5 Destornillador manual y tornillos	33
7.5.6 Soldador eléctrico con control de temperatura	34
7.5.7 Pasta para soldar (flux)	34
8. Estudio de alternativas y viabilidad	35
8.1 Evaluación de herramientas y tecnologías	35
8.1.1 Microcontrolador	35
8.1.1.1 Opción elegida	35
8.1.1.2 Opciones descartadas	35
8.1.2 Lenguaje backend	36
8.1.2.1 Opción elegida	36
8.1.2.2 Opciones descartadas	36
8.1.3 Base de datos	37
8.1.3.1 Opción elegida	37
8.1.3.2 Opciones descartadas	37
8.1.4 Herramientas de diseño	37
8.2 Viabilidad económica	38
8.3 Posibles vías de monetización	39
9. Descripción general de la solución	40
9.1 Arquitectura del sistema	40
9.2 Componentes principales	41
9.2.1 Sensores	41
9.2.2 Microcontrolador (Arduino UNO R4 WiFi)	42
9.2.3 Actuadores	43
9.2.4 Backend	44
9.2.5 Base de datos	45
9.2.6 Plataforma web (frontend)	46
9.3 Comunicación y funcionamiento general	49
10. Análisis del sistema	51
10.1 Objetivos del sistema	51
10.2 Actores del sistema	52
10.3 Requisitos del sistema	53
10.3.1 Requisitos no funcionales	53
10.3.2 Requisitos de información	54
10.3.3 Requisitos funcionales	55
11. Seguridad del sistema	57
12. Análisis de riesgos	58

13. Estimación del esfuerzo y planificación temporal	59
14. Conclusiones y trabajo futuro	62
15. Referencias	63

Índice de tablas

Tabla 1: Viabilidad económica	38
Tabla 2: Requisitos no funcionales	53
Tabla 3: Requisitos de información	54
Tabla 4: Requisitos funcionales	56

Índice de figuras

Figura 1: Aerospring Indoor Garden	11
Figura 2: LetPot	12
Figura 3: Arduino Proyecto OpenLanuza	16
Figura 4: Pantalla de login	19
Figura 5: Pantalla principal	20
Figura 6: Pantalla de configuración de usuarios	20
Figura 7: Pantalla de configuración de plantas	21
Figura 8: Arduino UNO R4 WiFi	24
Figura 9: Arduino IDE	25
Figura 10: Python	25
Figura 11: PHP	26
Figura 12: HTML	26
Figura 13: JavaScript	27
Figura 14: Bootstrap	27
Figura 15: MySQL	28
Figura 16: Visual Studio Code	28
Figura 17: MySQL Workbench	29
Figura 18: Visual Paradigm Online	29
Figura 19: Excalidraw	30
Figura 20: EZEstimate	30
Figura 21: Microsoft Visio	31
Figura 22: Microsoft Project	31
Figura 23: Sierra de pelo	32
Figura 23: taladro y papel de lija	33
Figura 23: Pistola de silicona	33
Figura 23: Soldador eléctrico	34
Figura 23: Pasta para soldar	34
Figura 24: Sistema Hidropónico	40
Figura 25: Algunos sensores	41
Figura 26: Arduino	42
Figura 27: Bomba de agua	43
Figura 28: Pantalla LCD	43
Figura 29: Base de datos	45
Figura 30: Logotipo	46
Figura 31: Pantalla panel de control	47
Figura 32: Pantalla histórico	47
Figura 33: Pantalla configuración de usuarios	48
Figura 34: Diagrama de arquitectura del sistema	50
Figura 35: Planificación temporal	59
Figura 36: Diagrama Gantt 1	60
Figura 37: Diagrama Gantt 2	61

1. Introducción

En la actualidad, el crecimiento demográfico, el cambio climático y la creciente escasez de recursos naturales han llevado a una preocupación global por la sostenibilidad y la eficiencia en la producción de alimentos [1]. En este contexto, la agricultura inteligente emerge como una alternativa innovadora que combina tecnología y sostenibilidad para optimizar los procesos agrícolas. Esta nueva visión incorpora sensores, automatización y análisis de datos con el objetivo de hacer más eficientes los cultivos, reducir el consumo de agua y energía, y anticipar problemas antes de que se conviertan en fallos críticos del sistema.

Una de las técnicas que más relevancia ha ganado en este ámbito es el cultivo hidropónico, que permite el crecimiento de plantas sin necesidad de suelo, utilizando soluciones minerales disueltas en agua. Este tipo de cultivo resulta especialmente interesante en entornos urbanos o en lugares donde el acceso a tierra fértil es limitado. La integración de tecnologías del Internet de las Cosas (IoT) en sistemas hidropónicos permite monitorizar y controlar las condiciones del cultivo en tiempo real, aumentando así la eficiencia y facilitando la gestión automatizada del proceso [2].

El presente Trabajo Fin de Grado (TFG) tiene como objetivo el diseño e implementación de un sistema inteligente para el control y supervisión de un cultivo hidropónico, apoyado en una arquitectura IoT. Para ello, se ha desarrollado un sistema completo que incluye sensores físicos (temperatura, nivel de agua, pH, entre otros), un microcontrolador Arduino, una plataforma web de monitorización, y una base de datos donde se almacena toda la información recogida y tratada. Además, se han incluido mecanismos automáticos de actuación y notificación que permiten reaccionar ante situaciones anómalas detectadas por los sensores.

A lo largo del desarrollo del proyecto, se ha buscado crear una solución modular, adaptable y accesible, tanto desde el punto de vista del *hardware* como del *software*. El sistema permite una gestión flexible de los parámetros críticos del cultivo, así como la definición de alarmas y la configuración de distintos tipos de plantas con sus rangos de valores óptimos. El trabajo combina conocimientos de electrónica, programación embebida, desarrollo web y bases de datos, integrándolos en una propuesta funcional, escalable y con potencial de aplicación real.

En este documento se detallan los aspectos más relevantes del desarrollo del sistema, incluyendo tanto el diseño físico y electrónico como la implementación del *software* embebido y de la plataforma web. La memoria se estructura en varios apartados que incluyen una revisión del estado del arte, los objetivos y metodología seguidos, el análisis y diseño del sistema, así como su implementación, validación y conclusiones finales. Además, se incorporan una serie de anexos donde se recogen diagramas, esquemas de *hardware*, fragmentos de código relevantes y capturas del sistema en funcionamiento, con el fin de complementar la información técnica que se encuentra en el cuerpo principal del trabajo.

2. Análisis de mercado

En los últimos años, el interés por los cultivos hidropónicos ha crecido considerablemente debido a su capacidad para optimizar el uso del agua, el espacio y los recursos agrícolas, especialmente en entornos urbanos o con limitaciones hídricas [3]. Este crecimiento ha impulsado el desarrollo de sistemas comerciales que permiten controlar y automatizar algunos procesos del cultivo hidropónico, con el objetivo de facilitar su gestión a usuarios domésticos, investigadores o pequeños agricultores.

Existen múltiples kits de cultivo en el mercado que permiten implementar sistemas hidropónicos de forma relativamente sencilla. Por ejemplo, productos como Aerospring Indoor Garden¹ o Rise Gardens² permiten cultivar vegetales con control parcial de parámetros como la iluminación, el flujo de agua o la ventilación, ofreciendo interfaces visuales intuitivas mediante aplicaciones móviles conectadas a la nube. Sin embargo, su coste suele ser elevado, rondando entre los 600 € y 1.000 €, lo que los hace inaccesibles para una parte importante de la población interesada en este tipo de soluciones [4].



Figura 1: Aerospring Indoor Garden

Otras alternativas como el sistema LetPot [5] están enfocadas a usuarios domésticos con poca experiencia. Estos dispositivos ofrecen funciones como el riego automático y alertas visuales cuando el nivel del agua o los nutrientes es bajo. Sin embargo, tienen limitaciones importantes: su capacidad de personalización es muy baja, no permiten ajustar parámetros específicos del cultivo, y no ofrecen funciones de control remoto o registro histórico de datos.

¹ Sistema hidropónico vertical de interior que permite cultivar múltiples plantas en poco espacio.
<https://aerospringhydroponics.com/collections/system>

² Jardín hidropónico inteligente para el hogar, modular y conectado a una app.
<https://risegardens.com/>



Figura 2: LetPot

En el ámbito más técnico o de investigación, algunos productos permiten la monitorización avanzada de parámetros como pH, TDS (Total Dissolved Solids), temperatura del agua o humedad ambiental. Sin embargo, muchos de estos sistemas requieren conocimientos previos en programación, electrónica o configuración de redes, lo que limita su adopción entre personas no especializadas. Además, en muchos casos se trata de soluciones cerradas, con componentes propietarios que dificultan la ampliación o modificación del sistema según las necesidades del usuario.

En este contexto, surge la necesidad de desarrollar una solución abierta, accesible y económica, que permita al usuario controlar y supervisar un sistema hidropónico desde una plataforma web. Esta solución debe facilitar la conexión con sensores físicos (como los de pH, TDS o nivel de agua), registrar datos en tiempo real, generar alarmas ante valores críticos, activar actuadores automáticamente y mantener un histórico de mediciones para su análisis. Además, debe ser modular y extensible, permitiendo futuras integraciones con nuevas funcionalidades o tecnologías, como la predicción de riegos mediante inteligencia artificial, sistemas de alertas inteligentes o dashboards visuales personalizados.

El proyecto desarrollado en este trabajo pretende cubrir ese vacío, proporcionando una plataforma de control basada en tecnologías de código abierto, con un diseño personalizable y adaptable tanto a necesidades educativas como productivas.

3. Objetivos

Este proyecto tiene como finalidad desarrollar una solución tecnológica que permita la monitorización y control de un sistema hidropónico doméstico a través de Internet, combinando el uso de sensores físicos con una plataforma web que actúe como interfaz de supervisión y gestión del cultivo. Este proyecto surge de la necesidad de contar con una herramienta de bajo coste, adaptable y de fácil uso para usuarios sin conocimientos

técnicos avanzados, que quieran iniciarse o mejorar sus cultivos mediante técnicas de agricultura inteligente.

3.1 Objetivo general

El objetivo principal del proyecto es diseñar e implementar un sistema hidropónico inteligente, conectado mediante tecnologías IoT, que permita la supervisión remota del estado del cultivo a través de una plataforma web accesible y funcional. Este sistema integrará sensores físicos, un controlador Arduino, una base de datos y un entorno web interactivo que permita visualizar los valores, generar alarmas, activar respuestas automáticas y gestionar configuraciones del sistema.

3.2 Objetivos específicos

Para alcanzar el objetivo general se han planteado los siguientes objetivos específicos, orientados a la implementación progresiva y validación del sistema:

- Desarrollar un circuito funcional que integre sensores físicos de temperatura, pH, TDS y nivel de agua, garantizando su correcta lectura mediante un microcontrolador compatible (Arduino).
- Implementar un script en lenguaje Python capaz de recibir y procesar los datos enviados por el microcontrolador, almacenarlos en una base de datos y verificar si se cumplen las condiciones para activar una alarma o una actuación automática.
- Diseñar una base de datos relacional eficiente (MySQL) que permita registrar la información de sensores, usuarios, alarmas, actuaciones y configuraciones del sistema.
- Crear una interfaz web funcional y visualmente clara (mediante PHP, JavaScript y Bootstrap), que muestre en tiempo real los valores registrados, el estado del sistema, y que permita al usuario administrar alarmas, configuraciones y rangos de cultivo por tipo de planta.
- Incorporar un sistema de autenticación de usuarios, gestión de roles y visualización de logs para mejorar la seguridad y trazabilidad de las acciones realizadas.

3.3 Objetivos técnicos

A lo largo del desarrollo del proyecto han surgido una serie de retos técnicos que han dado lugar a aprendizajes.

Uno de los primeros desafíos fue lograr una comunicación estable y continua entre el dispositivo Arduino y el script en Python, utilizando el protocolo HTTP y el intercambio de datos en formato XML. De este modo se consigue asegurar que los datos de los sensores lleguen correctamente al sistema.

Otro aspecto importante fue la sincronización de la base de datos con los cambios realizados desde la interfaz web. En particular, fue necesario implementar una lógica que permitiera aplicar de forma automática los rangos óptimos según la planta seleccionada por el usuario, lo que implicó una cuidadosa coordinación entre el backend y la lógica de configuración.

También supuso un reto diseñar y optimizar el sistema de alarmas y actuaciones automáticas. El objetivo era que el sistema fuera capaz de detectar situaciones anómalas, como un aumento excesivo de la temperatura del agua, y responder automáticamente, por ejemplo, aumentando el caudal de la bomba. También debe tener en cuenta otros factores críticos, como una disminución del nivel de agua en el tanque o la ausencia de agua en la zona de riego, detectables mediante sensores de nivel. En esos casos, el sistema debe emitir alertas o actuar para evitar que las plantas queden sin hidratación.

Por último, se trabajó especialmente en mantener el código modular y fácil de mantener. Esta decisión permitirá en el futuro incorporar nuevas funcionalidades, como dashboards externos para visualización avanzada o incluso algoritmos de inteligencia artificial para una gestión más inteligente del cultivo.

3.4 Objetivos personales

Además de los objetivos mencionados, este proyecto tiene un enfoque personal que busca consolidar las habilidades conseguidas a lo largo del grado universitario y afrontar nuevos retos de aprendizaje. La realización de este TFG supone una oportunidad para poner en práctica todos los conocimientos adquiridos durante los años de formación, desde la programación y la electrónica hasta el diseño de bases de datos.

A lo largo de este proyecto he podido aprender y dominar nuevas tecnologías relacionadas con el IoT y los sistemas embebidos, ampliando así mis competencias técnicas y la capacidad de adaptación a distintos entornos de desarrollo. Al mismo tiempo, he querido innovar explorando una nueva solución práctica que pueda aportar valor real a un problema concreto y así mejorar la calidad de vida o la eficiencia de los procesos agrícolas.

Por último, este TFG supone un paso importante para prepararme de cara al entorno laboral, al poner a prueba la capacidad de organizarme, trabajar de forma autónoma y resolver problemas reales de manera efectiva. En definitiva, con proyectos de este estilo busco crecer como profesional y como persona, aprendiendo de cada etapa del proyecto y sentando unas bases sólidas para los retos que puedan venir en el futuro.

4. Estado del arte y fundamentos teóricos

4.1 Introducción a la hidroponía

La hidroponía es una técnica agrícola que permite cultivar plantas sin suelo, utilizando soluciones acuosas enriquecidas con nutrientes esenciales. Aunque esta técnica se

remonta a civilizaciones antiguas, su desarrollo moderno comenzó a mediados del siglo XX y ha cobrado especial relevancia en el contexto actual, donde la sostenibilidad y el aprovechamiento eficiente de recursos son prioritarios. Frente a los métodos tradicionales, la hidroponía ofrece ventajas como el ahorro de agua, la reducción del uso de pesticidas y la posibilidad de cultivar en entornos urbanos o no aptos para la agricultura convencional [6].

4.2 Evolución tecnológica en hidroponía: automatización e IoT

La incorporación de la tecnología a la hidroponía ha permitido avanzar hacia sistemas más precisos, escalables y automatizados. Gracias a plataformas como Arduino, Raspberry Pi o ESP32, es posible integrar sensores que monitorean variables críticas como temperatura, pH, TDS (Total Dissolved Solids), y nivel de agua. Asimismo, el uso de actuadores como bombas peristálticas o relés permite realizar ajustes automáticos para mantener el sistema en condiciones óptimas [7].

En paralelo, el auge del IoT ha abierto nuevas posibilidades de conexión y control remoto. Un sistema hidropónico conectado puede transmitir datos en tiempo real a una aplicación web o móvil, activar mecanismos automáticos de riego o ventilación y generar alertas ante condiciones fuera de rango. Esta digitalización impulsa una forma de agricultura más inteligente, accesible y basada en datos [8].

4.3 Unidades estándar y variables monitorizadas

En la mayoría de sistemas hidropónicos inteligentes se trabaja con una serie de parámetros críticos cuya monitorización resulta esencial para la salud del cultivo:

- Temperatura del agua: entre 18 °C y 26 °C como rango óptimo.
- pH del agua/solución nutritiva: habitualmente entre 5.5 y 7, dependiendo del tipo de planta.
- TDS (conductividad eléctrica): mide la concentración de nutrientes en el agua. Los rangos ideales varían según el cultivo, pero suelen estar entre 600 y 900 ppm (partes por millón).
- Nivel de agua/solución nutritiva: su control es fundamental para evitar fallos en el riego o daños a la bomba.

El uso de sensores para estas variables, combinado con umbrales de referencia, permite automatizar el control del entorno de cultivo y realizar ajustes en tiempo real.

4.4 Proyectos relacionados y soluciones actuales

4.4.1 Proyecto OpenLanuza

El proyecto OpenLanuza, desarrollado en el colegio Juan de Lanuza (Zaragoza), es un sistema modular de cultivo hidropónico que emplea sensores de temperatura, humedad, pH y conductividad, junto con actuadores como bombas y luces de crecimiento. Aunque su enfoque es educativo y formativo, su estructura técnica comparte similitudes con el presente trabajo, especialmente en lo relativo a la monitorización ambiental y la automatización de cultivos [9].

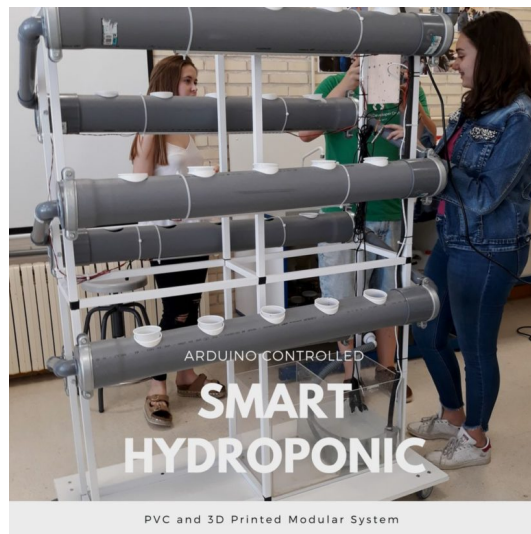


Figura 3: Arduino Proyecto OpenLanuza

4.4.2 Sistema automatizado de Kyle Gabriel

Kyle Gabriel, microbiólogo e ingeniero, desarrolló un sistema hidropónico inteligente utilizando una Raspberry Pi como núcleo central. Su propuesta incorpora sensores avanzados, como los de temperatura, humedad, pH, TDS, CO₂ y control de caudal, y destaca tanto por su alto nivel de integración como por el respaldo de una comunidad técnica muy activa. Sin embargo, su complejidad técnica y el coste asociado lo hacen más adecuado para usuarios con conocimientos avanzados.

En cambio, el sistema desarrollado en este proyecto apuesta por una alternativa más sencilla y accesible. Gracias al uso de Arduino, una arquitectura modular y una lógica clara, se busca facilitar su adopción por parte de personas sin formación técnica avanzada, sin renunciar a la funcionalidad ni a la capacidad de crecimiento del sistema [10].

4.4.3 Proyecto de Víctor Muñoz Rebollar (UPC)

Este proyecto, desarrollado en la Universidad Politécnica de Cataluña, implementa un sistema de riego hidropónico automatizado usando Arduino y sensores de pH, conductividad, oxígeno disuelto y temperatura. Su estructura es más sencilla que otros ejemplos, pero persigue los mismos objetivos de eficiencia y automatización. En este TFG

se propone una evolución de dicho planteamiento mediante la incorporación de una plataforma web y la posibilidad de modificar dinámicamente los parámetros [11].

4.4.4 Soluciones comerciales: NIDO PRO y Rocket24

El sistema NIDO PRO, de la empresa Agrotec, es una solución comercial para cultivo vertical con control automático de nutrientes, iluminación y riego. Incluye aplicación móvil, sensores y una interfaz amigable, aunque no ofrece tantas opciones de personalización como un sistema abierto como el desarrollado en este TFG [12].

Por su parte, el sistema NFT Rocket24 de Plantawa está orientado al cultivo doméstico, permitiendo cultivar hasta 24 plantas en espacios reducidos. Aunque su facilidad de uso lo convierte en una opción atractiva, carece de control automático por sensores, monitorización remota o personalización avanzada [13].

4.5 Conclusiones del estado del arte

A partir de la revisión de trabajos anteriores y soluciones existentes, se observa una tendencia clara hacia la integración de automatización, sensorización e interfaces gráficas en los sistemas de cultivo hidropónico. Sin embargo, muchas de estas propuestas presentan limitaciones, ya sea por su enfoque educativo, su complejidad técnica o su rigidez. El presente trabajo busca posicionarse como una solución intermedia suficientemente avanzada como para ofrecer funcionalidad real y automatización completa, pero lo bastante accesible como para que cualquier usuario interesado pueda adaptarla, personalizarla y ampliarla.

5. Normas y referencias

5.1 Metodología de desarrollo utilizada

Durante el desarrollo de este proyecto se ha adoptado una metodología ágil basada en el enfoque Scrum, adaptado a un contexto individual. La elección de esta metodología se debe a su flexibilidad, su enfoque iterativo y su capacidad para facilitar el control y la planificación en proyectos complejos, incluso cuando son llevados a cabo por una sola persona.

Scrum permite dividir el trabajo en bloques temporales (sprints) centrados en la entrega incremental de funcionalidades. En este proyecto, se definieron iteraciones semanales en las que se abordaban tareas concretas como el montaje del circuito físico, la programación de sensores, el desarrollo del backend o la interfaz web. Cada semana se planificaban objetivos específicos, y al final de cada ciclo se revisaban los avances y se reajustaban las prioridades en función de los problemas detectados o nuevas ideas surgidas.

Este enfoque permitió mantener una evolución continua del sistema, detectar errores con antelación y reajustar tanto el diseño como la funcionalidad de forma progresiva. La metodología también se complementó con el uso de herramientas de planificación y estimación como EZEstimate, que ayudaron a distribuir adecuadamente las tareas a lo largo del calendario disponible para el TFG.

5.2 Wireframes

Durante las primeras fases del desarrollo del sistema se diseñaron varios *wireframe* de la interfaz web utilizando la herramienta Excalidraw. Esta aplicación de dibujo colaborativo permite la creación rápida de bocetos con una estética “a mano alzada”, ideal para validar ideas de diseño antes de implementar el sistema definitivo.

El objetivo principal de estos prototipos fue definir la estructura general de la plataforma web, así como los elementos clave de interacción con el usuario. Gracias a estos diseños fue posible alinear la funcionalidad de la interfaz con los requisitos del sistema, facilitando la toma de decisiones sobre distribución visual, campos requeridos y navegación entre secciones.

A continuación, se muestran algunos de los *wireframes* más representativos:

- Pantalla de login: Se diseñó un formulario simple de acceso al sistema, donde el usuario introduce su nombre de usuario y contraseña. Esta pantalla sirvió como base para implementar el sistema de autenticación y gestión de roles.

The wireframe shows a rectangular box representing the login screen. At the top, the title 'Pantalla de login' is written in a handwritten style. Below the title, the screen is divided into two main sections by a vertical line. The left section is empty. The right section contains two input fields. The first field is preceded by the label 'Nombre de usuario' and the second by 'Contraseña'. Both labels and input fields are in a handwritten style.

Figura 4: Pantalla de login

- Pantalla principal del sistema: Esta vista incluye los datos más importantes recogidos por los sensores (temperatura, pH, TDS, nivel de agua) y el estado de la bomba de agua. También se representa un menú lateral que permite acceder a otras secciones del sistema. El objetivo de esta vista era mostrar la información de forma clara y centralizada.

Pantalla principal

Menú lateral

Datos sensores y actuadores

Sensor temperatura

Sensor nivel de agua 1

Sensor nivel de agua 2

Sensor pH

Sensor conductividad

Bomba de agua

Figura 5: Pantalla principal

- Pantalla de configuración de usuarios: Se diseñó un formulario para añadir o modificar usuarios, incluyendo campos como el nombre, correo electrónico y el rol asignado. Esta pantalla ayudó a estructurar la lógica de permisos dentro de la aplicación.

Pantalla de configuración de usuarios o plantas

Configuración usuario

Nombre

Correo electrónico asignado

Rol

...

Figura 6: Pantalla de configuración de usuarios

- Pantalla de configuración de plantas: Incluye campos para definir los valores límite asociados a distintos parámetros (temperatura, pH, niveles de agua). Esta configuración permite que el sistema reaccione automáticamente ante condiciones fuera del rango óptimo.

El prototipo de la pantalla de configuración de plantas se divide en dos secciones principales. La sección superior, titulada 'Pantalla de configuración de usuarios o plantas', contiene un sub-título 'Configuración plantas'. Debajo de este, se encuentran cuatro campos de entrada de texto organizados en una cuadrícula de 2x2. Los encabezados de los campos son: 'Valor límite temperatura' (arriba a la izquierda), 'Valor límite nivel de agua 1' (arriba a la derecha), 'Valor límite nivel de agua 2' (abajo a la izquierda) y 'Valor límite pH agua' (abajo a la derecha). Cada campo tiene un botón de 'X' en su esquina superior derecha para borrar el contenido. Debajo de estos campos, hay tres puntos negados '...' que indican la posibilidad de configurar más parámetros.

Figura 7: Pantalla de configuración de plantas

Estos prototipos me sirvieron como base para la creación final de la plataforma web que se puede observar con detalle en el documento Anexo 6: Manual de usuario.

5.3 Métricas

Para evaluar el correcto funcionamiento del sistema, se han definido una serie de métricas que permiten comprobar si el comportamiento general del prototipo se ajusta a los objetivos planteados. Estas métricas se han validado mediante pruebas técnicas con el sistema montado y también con pruebas de usuario de la plataforma web.

Las métricas consideradas han sido las siguientes:

- Precisión y comportamiento de los sensores: al tratarse de sensores de bajo coste, la precisión es limitada y depende en gran medida de la calibración manual.
En particular:
 - El sensor de pH incluye un potenciómetro de ajuste que se ha calibrado manualmente utilizando soluciones de referencia y un destornillador de precisión. Aunque puede presentar desviaciones, los valores se mantienen dentro de un rango aceptable para su uso doméstico.
 - Los sensores de nivel de agua son altamente sensibles y pueden reaccionar a pequeñas variaciones de contacto con el agua. Se han colocado en el sistema en zonas concretas para trabajar más adecuadamente.
 - El sensor de TDS requiere un tiempo considerable para estabilizarse tras su inmersión en el agua. Durante las pruebas, se detectó que tarda varios segundos en ofrecer una lectura fiable, lo que implica que no es apto para lecturas instantáneas.

- Fiabilidad del sistema de actuación automática: Se validó que, cuando un sensor detecta una condición crítica (como bajo nivel de agua o valores de pH fuera de rango), el sistema es capaz de generar automáticamente una alarma y activar la bomba de agua si así está configurado. Estas pruebas se realizaron en un entorno de laboratorio controlado.
- Evaluación de la interfaz web mediante prueba de usuarios: Para comprobar la usabilidad de la plataforma, se llevó a cabo una prueba con tres perfiles de usuario con diferentes niveles de experiencia técnica. Se les pidió realizar tareas comunes como iniciar sesión, modificar los valores límite de sensores o revisar datos históricos. En general, todos los usuarios fueron capaces de completar las tareas sin asistencia externa, lo cual sugiere que la interfaz es clara, funcional y comprensible para el público objetivo.

Estas métricas han servido como base para validar el sistema y justificar que cumple su propósito principal en un entorno doméstico o educativo. Los resultados completos de estas pruebas se detallan más adelante.

6. Restricciones y alcance

6.1 Restricciones

El sistema presenta una serie de limitaciones técnicas y operativas, definidas tanto por el alcance del proyecto como por las decisiones de diseño tomadas:

- *Hardware* limitado: Se ha optado por una arquitectura basada en Arduino con sensores de bajo coste (pH, TDS, temperatura, nivel), lo que implica precisión limitada respecto a sistemas más profesionales.
- Sin batería de respaldo: El sistema depende de una fuente de alimentación continua; en caso de corte eléctrico, dejará de funcionar.
- Procesamiento local mínimo: Toda la lógica avanzada se implementa en el backend, ya que el microcontrolador tiene recursos limitados.

6.3 Alcance del proyecto

Este proyecto tiene un alcance bien delimitado que permite validar la funcionalidad del sistema sin exceder los recursos disponibles:

- Se cubre la monitorización y actuación automatizada sobre un sistema hidropónico doméstico, enfocado a plantas de hoja como la lechuga.
- El sistema permite visualizar datos en tiempo real, generar alertas por valores críticos, activar una bomba de forma automática y mantener un registro histórico de mediciones.
- Se ha desarrollado una plataforma web funcional con control de acceso, configuración de plantas y sensores, y gestión de usuarios.
- El proyecto incluye el desarrollo completo del *hardware*, backend y frontend, así como el montaje físico del sistema.

Cabe destacar que el sistema desarrollado corresponde a un prototipo funcional orientado a validar una prueba de concepto. Su diseño responde a un enfoque de bajo coste y fácil montaje, pensado para usuarios individuales que deseen implementar un pequeño sistema hidropónico doméstico. Aunque el prototipo demuestra la viabilidad técnica del sistema, su explotación a mayor escala requeriría procesos de producción estandarizados, mejoras en la durabilidad de componentes y consideraciones adicionales en materia de seguridad eléctrica y mantenimiento. En su forma actual, se trata de una solución adecuada para entornos domésticos, accesible para usuarios con conocimientos básicos de electrónica y jardinería.

7. Herramientas y tecnologías utilizadas

Durante el desarrollo del proyecto se han empleado varias herramientas y tecnologías, tanto en *hardware* como en *software*, que han permitido crear una solución funcional y escalable. A continuación, se detallan todas ellas, junto con una breve justificación de su elección frente a alternativas existentes.

7.1 Hardware

7.1.1 Arduino UNO R4 WiFi (WiFiS3)

Microcontrolador de 32 bits con conectividad WiFi integrada. Ha sido elegido por su compatibilidad con múltiples sensores, facilidad de programación y amplio soporte en la comunidad. A diferencia de otras alternativas como Raspberry Pi o ESP32, ofrece un equilibrio adecuado entre simplicidad y funcionalidad para tareas de sensorización básica y envío de datos.

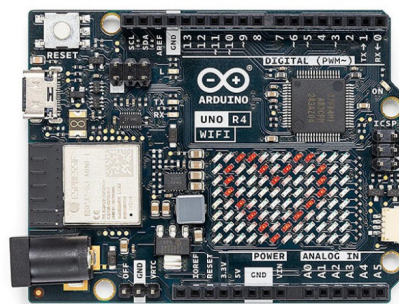


Figura 8: Arduino UNO R4 WiFi

7.1.2 Sensores de bajo coste (TDS, pH, temperatura, nivel de agua)

Se han seleccionado sensores analógicos compatibles con Arduino, con el objetivo de mantener el sistema económico y accesible. Aunque presentan menor precisión que sensores industriales, su rendimiento es suficiente para un entorno doméstico o educativo.

7.1.3 Pantalla LCD I2C (HD44780)

Permite mostrar información local en tiempo real, como el estado de los sensores. Su uso mejora la autonomía del sistema en caso de caída del servidor, y su integración con Arduino es sencilla gracias a las librerías disponibles.

7.2 Software y tecnologías de desarrollo

7.2.1 Arduino IDE

Arduino IDE es un entorno de desarrollo integrado especialmente diseñado para programar microcontroladores compatibles con la plataforma Arduino. Proporciona una interfaz sencilla para escribir, compilar y cargar código directamente a la placa, además de permitir el uso de librerías específicas para la gestión de sensores, actuadores y comunicación de datos. Es utilizado en proyectos de electrónica, robótica y sistemas embebidos. Se eligió por ser el estándar oficial para placas Arduino, con soporte nativo de bibliotecas y una interfaz sencilla y eficiente.



Figura 9: Arduino IDE

7.2.2 Python

Python es un lenguaje de programación de alto nivel, multiplataforma y de código abierto. Se caracteriza por su sintaxis sencilla y clara, lo que lo convierte en una herramienta accesible para desarrolladores de distintos niveles. Su versatilidad permite abordar tareas como análisis de datos, desarrollo de scripts automatizados, gestión de servidores o creación de aplicaciones web. Python dispone de un ecosistema muy amplio de librerías que facilitan el desarrollo de soluciones técnicas robustas.



Figura 10: Python

7.2.3 PHP

PHP es un lenguaje de programación del lado del servidor ampliamente utilizado para el desarrollo web. Permite generar contenido dinámico, gestionar sesiones de usuario y acceder a bases de datos. Es especialmente adecuado para la creación de páginas web que requieren interacción con el backend, siendo compatible con la mayoría de servidores y sistemas operativos actuales.



Figura 11: PHP

7.2.4 HTML

HTML es el lenguaje estándar de marcado utilizado para estructurar páginas web. Define la jerarquía y organización del contenido mediante etiquetas, permitiendo representar elementos como encabezados, párrafos, enlaces, listas, formularios y más. Constituye la base sobre la cual se construyen las interfaces web modernas.



Figura 12: HTML

7.2.6 JavaScript

JavaScript es un lenguaje de programación de alto nivel que se ejecuta principalmente en el navegador del cliente. Se utiliza para dar interactividad a las páginas web mediante el control de eventos, la manipulación dinámica del DOM (Document Object Model), validaciones en formularios, animaciones y otras funcionalidades. Junto con HTML, forma el núcleo del desarrollo web front-end.



Figura 13: JavaScript

7.2.7 Bootstrap

Bootstrap es un framework de código abierto para el desarrollo front-end que proporciona una amplia colección de componentes visuales predefinidos como botones, formularios, menús y tablas. Está basado en HTML, CSS y JavaScript, y facilita la creación de interfaces web que se adaptan automáticamente a diferentes dispositivos y tamaños de pantalla.

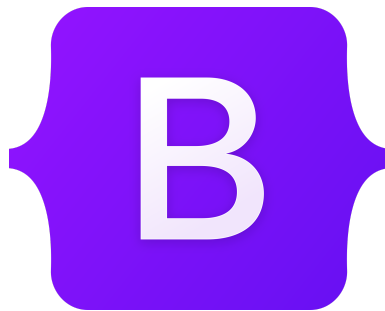


Figura 14: Bootstrap

7.2.8 MySQL

MySQL es un sistema de gestión de bases de datos relacional de código abierto, ampliamente utilizado en aplicaciones web y sistemas de información por su rendimiento, fiabilidad y compatibilidad multiplataforma. Permite almacenar, consultar y administrar grandes volúmenes de datos estructurados mediante el uso del lenguaje SQL. En este proyecto, ha sido empleado para gestionar y organizar la información relativa a usuarios, sensores, configuraciones, registros históricos y eventos del sistema, garantizando un acceso eficiente y seguro a los datos.



Figura 15: MySQL

7.2.9 Visual Studio Code

Visual Studio Code es un editor de código fuente desarrollado por Microsoft, multiplataforma y de código abierto. Ofrece soporte para diferentes lenguajes de programación, integración con sistemas de control de versiones como Git, y un sistema de extensiones que permite personalizar el entorno de trabajo. Se eligió por su ligereza, extensibilidad, integración con sistemas de control de versiones y compatibilidad con múltiples lenguajes.

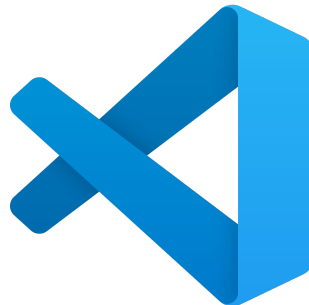


Figura 16: Visual Studio Code

7.3 Herramientas de diseño y modelado

7.3.1 MySQL Workbench

MySQL Workbench es una herramienta visual oficial de MySQL para el diseño, modelado y administración de bases de datos. Dispone de un editor gráfico que permite construir modelos entidad-relación de forma intuitiva, facilitando la definición de tablas, claves primarias/foráneas y relaciones entre entidades. Ha sido utilizado para diseñar y validar el modelo lógico de la base de datos antes de su implementación definitiva, asegurando una estructura coherente y normalizada.



Figura 17: MySQL Workbench

7.3.2 Visual Paradigm Online

Visual Paradigm es una herramienta de modelado orientada a la creación de diagramas UML y otros tipos de representaciones visuales. Permite diseñar casos de uso, clases, actividades, secuencia y otros elementos útiles en el análisis y diseño de sistemas. También facilita el trabajo colaborativo y la documentación de proyectos. Se eligió por su entorno web accesible, variedad de plantillas y facilidad para exportar resultados en formato imagen o PDF.



Figura 18: Visual Paradigm Online

7.3.3 Excalidraw

Excalidraw es una herramienta online de dibujo colaborativo que simula una pizarra digital. Su estética "hecha a mano" y enfoque minimalista la convierte en una opción ideal para el diseño de prototipos rápidos. Se ha utilizado para hacer bocetos, validar ideas de diseño y tomar decisiones tempranas sobre la disposición visual sin necesidad de una herramienta de diseño avanzada.



Figura 19: Excalidraw

7.3.4 EZEstimate

EZEstimate es una herramienta especializada para la estimación de esfuerzo en proyectos de *software*. Permite calcular de forma estructurada la duración aproximada de un proyecto considerando factores como la complejidad técnica, el entorno de desarrollo, la cantidad de actores y casos de uso. En este Trabajo de Fin de Grado se utilizó para obtener una estimación objetiva del tiempo necesario para completar el desarrollo, apoyando la planificación temporal y la asignación de recursos de manera fundamentada.



Figura 20: EZEstimate

7.3.5 Microsoft Visio

Microsoft Visio es una herramienta de diagramación profesional que permite crear esquemas técnicos, diagramas de flujo, modelos UML y representaciones visuales complejas. Su interfaz intuitiva y su integración con el ecosistema de Microsoft Office facilitan el diseño estructurado de modelos de datos, procesos y arquitecturas. En este TFG se utilizó para representar mediante diagramas la arquitectura del sistema, el flujo principal del Arduino y los procesos del backend Python.



Figura 21: Microsoft Visio

7.3.6 Microsoft Project

Microsoft Project es una herramienta de gestión de proyectos utilizada para planificar tareas, establecer cronogramas, asignar recursos y realizar seguimiento del progreso. Ofrece una visión estructurada y temporal de todas las fases de un proyecto, con funciones de desglose jerárquico, dependencias entre tareas y control de hitos clave. En este proyecto se empleó para elaborar el cronograma detallado del desarrollo, organizando las distintas iteraciones, reuniones y entregas, y facilitando una planificación realista acorde con los recursos disponibles y las restricciones académicas del calendario.



Figura 22: Microsoft Project

7.5 Herramientas de montaje y prototipado físico

Además del desarrollo *software* y electrónico, el proyecto requirió la construcción física de un sistema hidropónico funcional, lo cual implicó el uso de herramientas manuales y eléctricas para el montaje, la fijación y el acabado del prototipo.

7.5.1 Sierra de pelo

Se utilizó una sierra de pelo para realizar cortes precisos en materiales plásticos, especialmente en los tubos de PVC que conforman la estructura del sistema hidropónico. Esta herramienta permitió ajustar las dimensiones y realizar aberturas específicas, como el orificio en la tapa de la caja de conexiones para que se vea mejor la pantalla LCD, donde se integran los sensores con la placa y el microcontrolador. Además, se utilizaron hojas de mayor grosor cuando fue necesario cortar secciones de PVC más resistentes o de mayor diámetro.



Figura 23: Sierra de pelo

7.5.2 Taladro eléctrico

Se empleó para crear agujeros circulares de tamaño adecuado en los tubos de PVC donde posteriormente se insertarían los recipientes de las plantas. Se utilizó una corona perforadora de diámetro acorde al vaso, con un taladro inalámbrico de 20 V.

7.5.3 Papel de lija

Tras los cortes y perforaciones, se lijaron los bordes para evitar imperfecciones y mejorar la estética del acabado. También ayudó a evitar que se dañaran los recipientes de las plantas.



Figura 23: taladro y papel de lija

7.5.4 Pistola de silicona

Se utilizó para fijar componentes en el interior de la caja del sistema y para sellar uniones entre los distintos trozos de tubo empleados en la construcción del sistema hidropónico. Su aplicación resulta rápida, resistente y, en caso necesario, reversible, permitiendo desmontar los elementos sin causar daños.



Figura 23: Pistola de silicona

7.5.5 Destornillador manual y tornillos

Se utilizaron herramientas básicas como destornilladores de diferentes tipos para asegurar la tapa y el montaje final de la caja que contiene el Arduino, el sistema de cableado, las conexiones de los sensores y el LCD. Se utilizó una caja de montaje estándar, perforada manualmente y adaptada a las dimensiones de los componentes.

7.5.6 Soldador eléctrico con control de temperatura

Se utilizó para soldar las conexiones entre sensores, cables y placas de prototipado. Este modelo específico permite regular la temperatura para evitar dañar componentes sensibles.



Figura 23: Soldador eléctrico

7.5.7 Pasta para soldar (flux)

Se aplicó durante el proceso de soldadura para evitar oxidación, mejorar la conductividad y facilitar la adhesión del estaño. Fue especialmente útil tras detectar problemas de oxidación en conexiones tras exposiciones prolongadas a humedad y altas temperaturas.



Figura 23: Pasta para soldar

8. Estudio de alternativas y viabilidad

8.1 Evaluación de herramientas y tecnologías

Durante el desarrollo del proyecto, se consideraron múltiples herramientas y tecnologías tanto a nivel de *hardware* como de *software*. A continuación, se presentan las principales decisiones adoptadas junto con una justificación comparativa frente a otras alternativas disponibles.

8.1.1 Microcontrolador

Durante la fase inicial del proyecto se valoraron varias opciones en cuanto al microcontrolador que serviría de núcleo para la sensorización y envío de datos del sistema hidropónico. La elección de la placa era crítica, ya que debía garantizar compatibilidad con múltiples sensores analógicos, estabilidad en la transmisión de datos y facilidad de programación. A continuación, se resumen las principales alternativas analizadas y la justificación de la decisión adoptada.

8.1.1.1 Opción elegida

Finalmente, se optó por el uso de Arduino UNO R4 WiFi, una versión moderna de la clásica placa Arduino que incorpora conectividad inalámbrica. Esta opción ofrecía una excelente relación entre simplicidad de uso, bajo coste y funcionalidad integrada, lo que la hacía especialmente adecuada para un proyecto de estas características. Además, su compatibilidad con una amplia gama de sensores analógicos y digitales facilitó la integración con los módulos de temperatura, pH, TDS y nivel de agua.

También influyó la experiencia previa adquirida con esta plataforma en una asignatura del grado, lo que permitió reducir la curva de aprendizaje y centrarse en la lógica de aplicación en lugar de resolver problemas de bajo nivel.

8.1.1.2 Opciones descartadas

El microcontrolador ESP32 fue considerado como posible alternativa, ya que cuenta con mayor capacidad de procesamiento, conectividad dual (WiFi y Bluetooth), y más pines GPIO. Sin embargo, su uso implicaba una mayor complejidad de programación, especialmente en lo relativo a la configuración de redes, gestión de interrupciones y control de periféricos.

Además, algunas librerías utilizadas para los sensores presentaban problemas de compatibilidad o requerían adaptaciones específicas para funcionar correctamente en el entorno del ESP32. Por ello, se descartó esta opción al no ofrecer una ventaja decisiva respecto a Arduino en el contexto de un sistema hidropónico doméstico y controlado.

También se valoró el uso de una Raspberry Pi, especialmente por su capacidad para ejecutar sistemas operativos completos y lenguajes de alto nivel, así como por su conectividad avanzada. Sin embargo, en este caso, su uso habría implicado un incremento significativo en el coste, el consumo energético y la complejidad general del sistema.

Dado que las tareas a realizar eran esencialmente de lectura de sensores y envío periódico de datos, se consideró que utilizar una Raspberry Pi suponía una sobrecarga innecesaria, tanto a nivel técnico como económico.

8.1.2 Lenguaje backend

8.1.2.1 Opción elegida

Para la implementación del backend del sistema, se eligió Python como lenguaje principal, debido a su sintaxis clara, facilidad de aprendizaje, y su amplio ecosistema de librerías orientadas al desarrollo rápido de aplicaciones. Python resultó especialmente útil para la creación de scripts de automatización, manejo de peticiones HTTP y comunicación con el microcontrolador a través de protocolos simples como HTTP/XML.

Desde un punto de vista personal, el uso de Python también representó un reto y una oportunidad de aprendizaje. Aunque había trabajado con este lenguaje en algunas ocasiones, no lo había explorado tan en profundidad como me hubiera gustado. Este proyecto permitió reforzar mis conocimientos en desarrollo backend, lógica de negocio, y automatización de procesos, ganando confianza en su aplicación práctica dentro de un entorno real.

8.1.2.2 Opciones descartadas

R fue considerada brevemente como alternativa, especialmente por su potencia en análisis de datos y procesamiento estadístico. No obstante, se descartó por no estar orientado a la construcción de backends en tiempo real, ni contar con un ecosistema tan robusto para tareas de automatización, gestión de servidores o manejo eficiente de peticiones HTTP. Su uso habría supuesto una curva de integración más elevada, una mayor dependencia de paquetes externos poco optimizados para entornos embebidos, y una menor flexibilidad para mantener una arquitectura modular y escalable.

PHP, por su parte, fue utilizado exclusivamente en el desarrollo de la interfaz web, donde demostró ser eficiente en la generación dinámica de páginas y la gestión de sesiones de usuario. Sin embargo, no se utilizó como lenguaje de backend para la lógica del sistema por tratarse de un lenguaje más orientado al desarrollo web que al manejo estructurado de tareas en segundo plano o procesos automatizados. Sus limitaciones como lenguaje de propósito general lo hacían menos adecuado para este tipo de implementación en comparación con Python.

8.1.3 Base de datos

8.1.3.1 Opción elegida

Para la base de datos, se seleccionó MySQL, un sistema de gestión de bases de datos relacional consolidado en entornos web y empresariales. MySQL destaca por su madurez, rendimiento estable y extenso ecosistema de herramientas, entre las que se encuentra MySQL Workbench, utilizada en este proyecto para el diseño lógico del modelo entidad-relación (EER). Esta base de datos permite gestionar múltiples usuarios con distintos niveles de acceso, soporta transacciones seguras y asegura la integridad referencial mediante el uso de claves primarias y foráneas. Su elección es adecuada para un sistema con múltiples entidades relacionadas, como sensores, usuarios, plantas, alarmas o configuraciones personalizadas, donde se requiere una estructura relacional robusta y consultas SQL eficientes.

8.1.3.2 Opciones descartadas

MariaDB, aunque es una bifurcación directa de MySQL y completamente compatible a nivel de comandos y estructuras, fue descartada principalmente porque MySQL ofrece un mayor respaldo comercial, actualizaciones más regulares y una mejor integración con herramientas oficiales como MySQL Workbench. Se optó por la versión original y más extendida del sistema.

MongoDB, como solución NoSQL orientada a documentos, también fue considerada en fases iniciales. Sin embargo, fue descartada al requerir el sistema un modelo de datos claramente relacional, con dependencias estructuradas entre entidades. La naturaleza flexible y desestructurada de MongoDB hubiera dificultado la implementación de restricciones de integridad, así como la ejecución de consultas complejas con múltiples uniones y relaciones. Además, su adopción habría supuesto una mayor curva de aprendizaje sin aportar beneficios significativos para este tipo de aplicación.

8.1.4 Herramientas de diseño

Se utilizaron combinadamente para diferentes tareas: prototipado rápido (Excalidraw), diagramas UML (Visual Paradigm) y esquemas técnicos (Microsoft Visio). Otras herramientas como Figma o Lucidchart fueron consideradas pero desestimadas por ser más complejas o requerir licencias de pago.

8.2 Viabilidad económica

Este proyecto ha sido concebido con un enfoque de bajo coste, accesible y replicable. El coste total del sistema prototipo ha sido de unos 125 euros aproximadamente, incluyendo sensores, microcontrolador, materiales plásticos y elementos de montaje.

Componente	Coste (€)
Arduino UNO R4 WiFi	34,90
Sensor pH	11,70
Sensor TDS	3,10
Sensor temperatura DS18B20	1,20
Sensores de nivel de agua	0,60 x 2
Bomba de agua + mosfet	4,50
Caja y materiales de montaje	25,00
Otros (cables, LCD, placas)	35,00
Grana planta	1,00
Soporte germinación	0,50 por planta
Nutriente	14 € por cada 132 litros de agua
Total aproximado	125,00

Tabla 1: Viabilidad económica

Este coste podría reducirse en futuras versiones del proyecto mediante la integración de sensores multifunción o la reutilización de componentes ya existentes. Sin embargo, es importante señalar que, durante el desarrollo, fue necesario comprar varios tipos de sensores, ya que algunos se deterioraron con el uso continuado durante las pruebas.

Además del coste del *hardware*, es importante considerar el coste estimado del servidor necesario para alojar y mantener la aplicación web. En este proyecto, al tratarse de una solución funcional de pequeño alcance, bastaría con un servicio de alojamiento básico, con un coste aproximado de 80 a 100 euros anuales. Esta cifra incluye el funcionamiento del backend, la base de datos y el acceso a través del navegador. Si se optara por añadir un dominio personalizado, el coste podría incrementarse ligeramente.

8.3 Posibles vías de monetización

Aunque este TFG no tiene fines comerciales directos, se han identificado algunas posibilidades futuras de monetización si el sistema se evoluciona o escala:

- Versión kit educativo: Un paquete con piezas, instrucciones y plataforma preconfigurada para centros educativos interesados en enseñar agricultura e IoT de forma práctica.
- Licencia freemium de plataforma web: Ofrecer acceso básico gratuito con opción de características avanzadas bajo suscripción (por ejemplo: visualización de gráficos históricos, IA para predicción, control móvil remoto, etc.).
- Servicio de consultoría e instalación: Implementar el sistema personalizado para pequeños agricultores o usuarios domésticos sin conocimientos técnicos.
- Venta del sistema completo preensamblado: Con interfaz plug-and-play para usuarios sin necesidad de montaje físico ni configuración técnica.

Estas alternativas podrían explorarse en trabajos futuros o en colaboración con terceros interesados.

9. Descripción general de la solución

9.1 Arquitectura del sistema

La solución desarrollada consiste en un sistema hidropónico inteligente basado en una arquitectura distribuida que combina *hardware* embebido, backend de procesamiento y una plataforma web como interfaz de usuario. El sistema ha sido diseñado siguiendo un enfoque modular y escalable, que permite tanto el crecimiento futuro del sistema como su adaptación a diferentes necesidades de cultivo.

La arquitectura se compone de tres niveles principales:

- Nivel físico (sensores y actuadores): Encargado de la captación de datos y activación de respuestas automáticas mediante un microcontrolador.
- Nivel lógico (backend): Donde se recibe, procesa y almacena la información, además de gestionar alarmas, lógica de automatización y configuraciones del sistema.
- Nivel de presentación (frontend web): Donde el usuario puede visualizar datos, configurar parámetros, gestionar usuarios y supervisar el estado del sistema en tiempo real.



Figura 24: Sistema Hidropónico

9.2 Componentes principales

9.2.1 Sensores

El sistema incorpora los siguientes sensores analógicos:

- Sensor de pH: Mide la acidez o alcalinidad del agua. Es calibrable mediante un potenciómetro.
- Sensor de TDS: Mide la cantidad de sólidos disueltos en el agua (nutrientes). Necesita unos segundos de estabilización.
- Sensor de temperatura (DS18B20): Evalúa la temperatura del agua del cultivo.
- Sensores de nivel de agua (x2): Detectan niveles mínimo y máximo del depósito. Son especialmente sensibles al contacto.

Estos sensores permiten una monitorización continua y personalizada de las condiciones del cultivo.

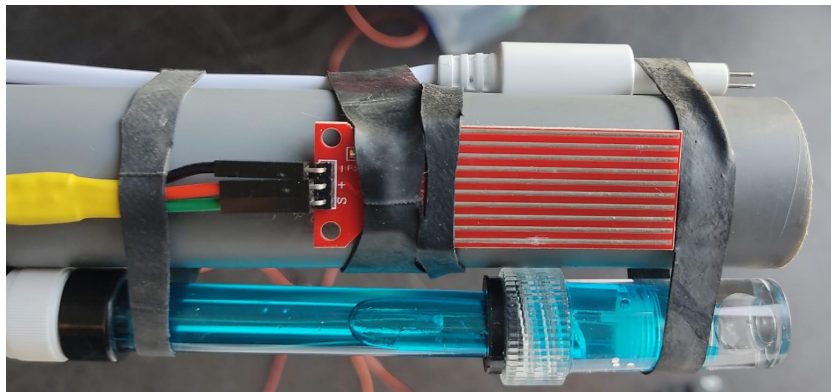


Figura 25: Algunos sensores

Todos los sensores del sistema se pueden apreciar en el Anexo 7: Montaje físico del sistema.

9.2.2 Microcontrolador (Arduino UNO R4 WiFi)

Este dispositivo centraliza la lectura de sensores y la comunicación con el sistema backend. Está programado para tomar lecturas periódicas, mostrarlas en una pantalla LCD y enviarlas al servidor mediante peticiones HTTP en formato XML.

Además, en función de los datos recogidos y de las configuraciones definidas por el usuario, puede activar la bomba de agua si los valores están fuera del rango deseado.

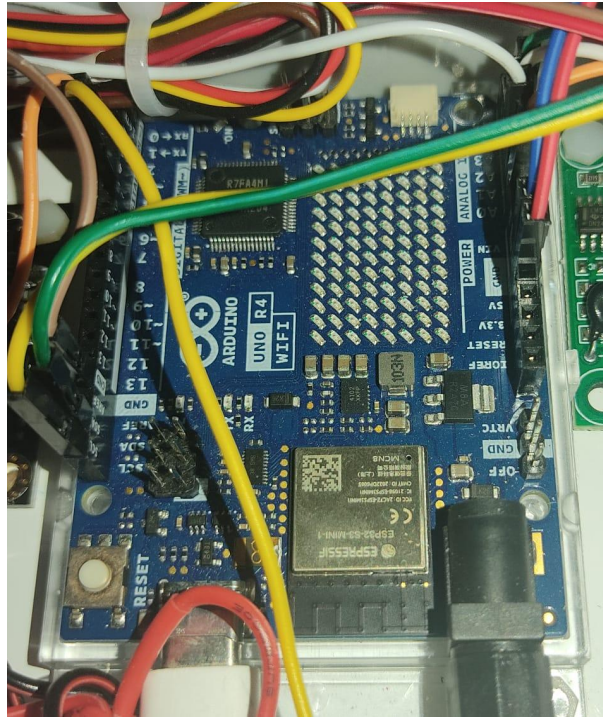


Figura 26: Arduino

El montaje del microcontrolador Arduino, junto con los sensores, se puede encontrar en el Anexo 7: Montaje físico del sistema.

9.2.3 Actuadores

Realmente el actuador principal del sistema es la bomba de agua, pero también debe considerarse la pantalla LCD como un elemento activo, ya que forma parte de la interacción con el usuario y contribuye al funcionamiento general del sistema.

- Bomba de agua: Controlada mediante un mosfet. Puede activarse automáticamente según las condiciones detectadas en el sistema.
- Pantalla LCD I2C: Muestra valores locales en tiempo real, útil en caso de pérdida de conectividad.

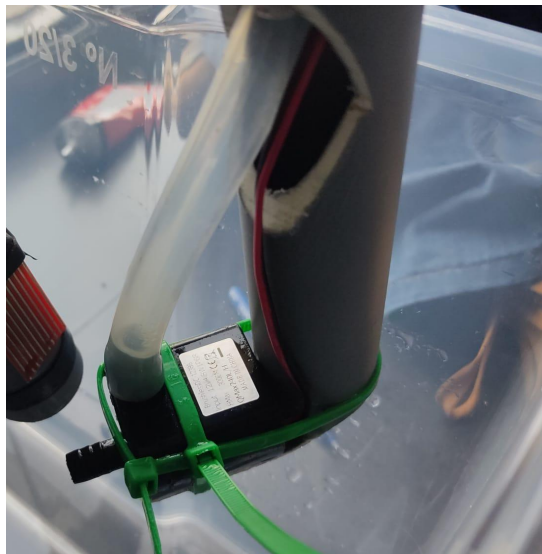


Figura 27: Bomba de agua

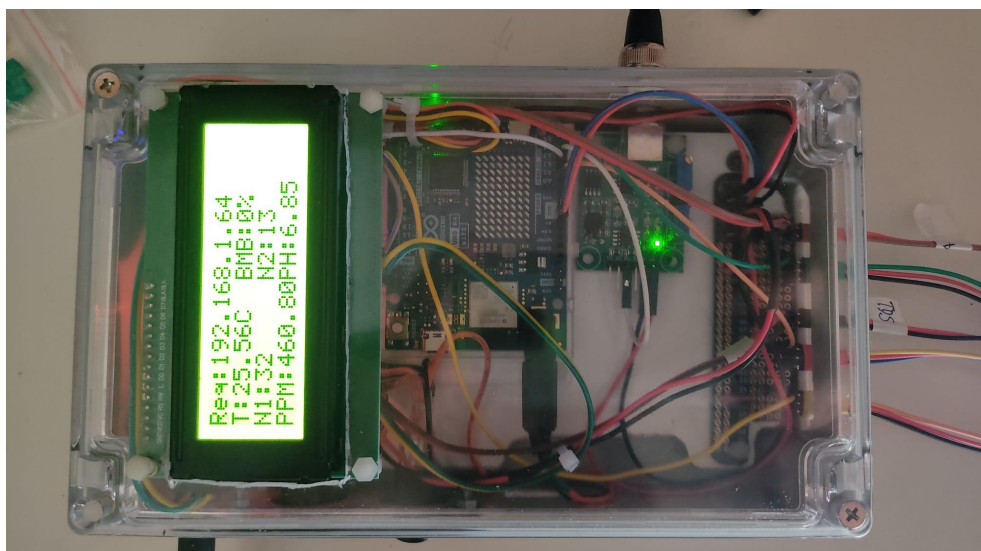


Figura 28: Pantalla LCD

Todos los componentes físicos del sistema, al igual que todo el proceso de fabricación se encuentran detallados en el anexo Anexo 7: Montaje físico del sistema.

9.2.4 Backend

El backend ha sido desarrollado en Python y funciona como un servidor intermedio. Sus principales funciones son:

- Recibir los datos enviados por el Arduino.
- Analizar si los valores superan umbrales críticos definidos.
- Registrar los datos en la base de datos relacional (MySQL).
- Generar alarmas automáticas y activar mecanismos de respuesta si es necesario.

Este backend se mantiene activo en segundo plano, siendo ligero, eficiente y fácil de extender.

El flujo general del sistema, junto con información de los archivos del backend y el frontend se encuentra mejor detallado en el anexo Anexo 5: Documentación técnica.

9.2.5 Base de datos

El modelo relacional implementado en MySQL permite almacenar:

- Lecturas de sensores históricas.
- Alarmas generadas y acciones automáticas.
- Usuarios y roles.
- Configuraciones específicas por tipo de planta.

El diseño de la base de datos fue modelado previamente en MySQL Workbench y puede consultarse en detalle en el Anexo 2: Análisis y diseño del sistema.

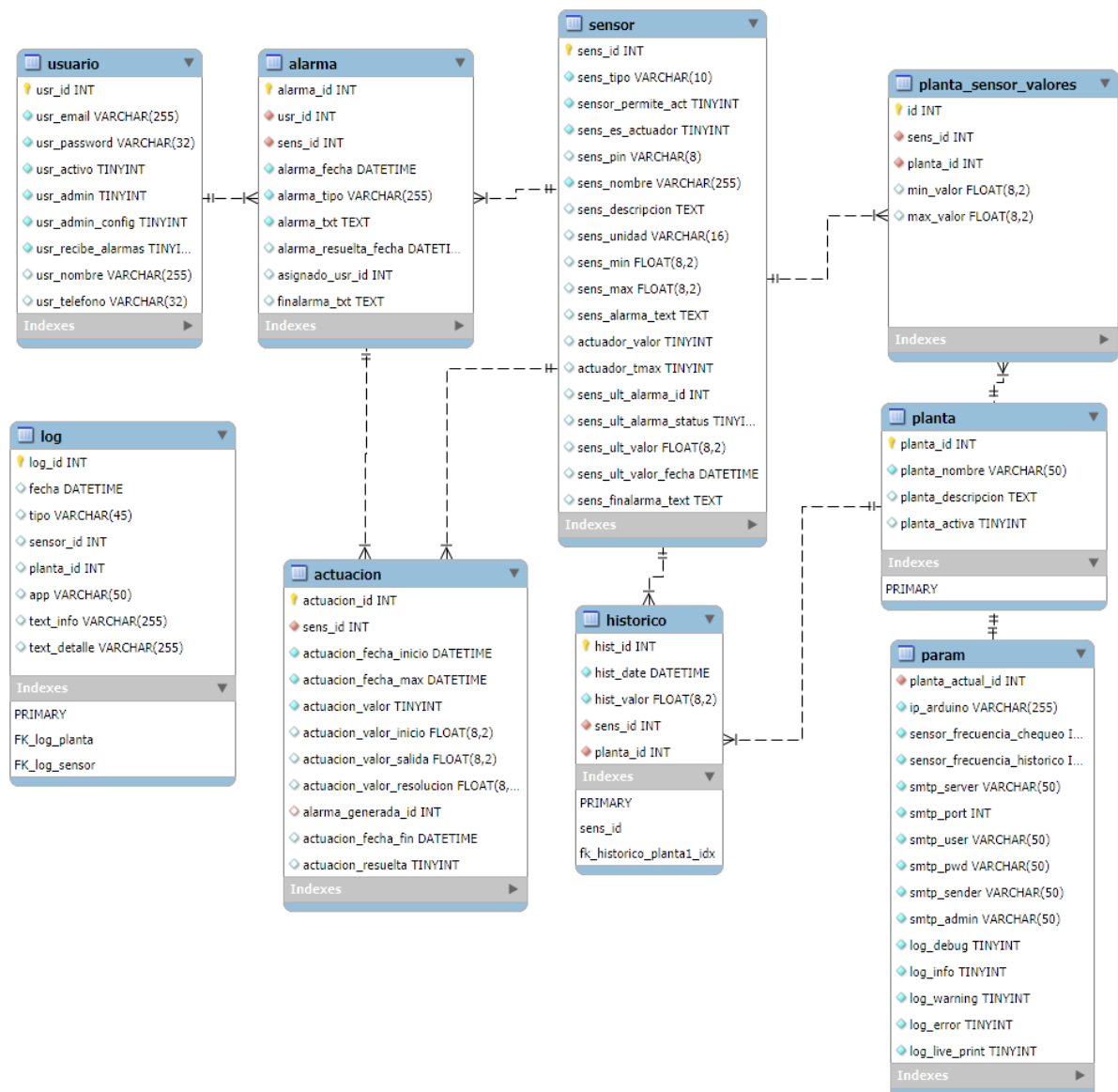


Figura 29: Base de datos

9.2.6 Plataforma web (frontend)

Desarrollada con PHP y JavaScript, esta plataforma permite:

- Visualizar en tiempo real los datos de sensores.
- Consultar el histórico de mediciones.
- Configurar alarmas, parámetros críticos y plantas.
- Gestionar usuarios y niveles de acceso.
- Activar manualmente la bomba o reiniciar el sistema.

La plataforma es responsive gracias a Bootstrap, lo que facilita su uso desde móviles y tablets.

Se añadió un logotipo que representa una gota de agua fusionada con una hoja verde seleccionado desde un repositorio de recursos libres.

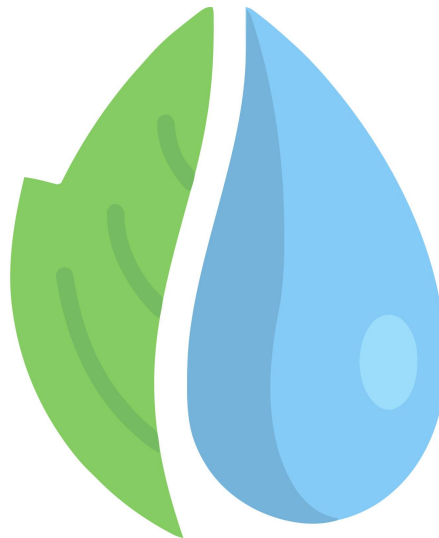


Figura 30: Logotipo

Como nombre de la aplicación se puso “Hidroponify”, un nombre breve y reconocible que fusiona el concepto de hidroponía con la terminación tecnológica “-ify”.

El flujo general del sistema, junto con información de los archivos del backend y el frontend se encuentra mejor detallado en el anexo Anexo 5: Documentación técnica.

Todas las vistas de la plataforma web, con mayor detalle, se encuentran en el Anexo 6: Manual de usuario.



Menu

Panel de control

Administración

Historico

Desconexión

Sistema hidropónico cultivando LECHUGA



Temperatura
DS18B20
23.25 °C

27/06/2025
10:40:45



Nivel de Agua 1
37 mm

27/06/2025
10:40:45



Nivel de Agua 2
1 mm

27/06/2025
10:40:46



Calidad del Agua
(TDS)
521.6 ppm

27/06/2025
10:40:46



Bomba de Agua
0 %

27/06/2025
10:40:46



Sensor PH (0-14)
6.92 pH

27/06/2025
10:40:46

Alarmas pendientes

Fecha	Sensor	Tipo	Alarma
27/06/2025 09:35:29	Nivel de Agua 2	CRÍTICA	ALERTA: Nivel de agua insuficiente en el tubo de circulación

Alarmas resueltas

Figura 31: Pantalla panel de control



Menu

Panel de control

Administración

Historico

Desconexión

Historico

Valores historicos de sensores

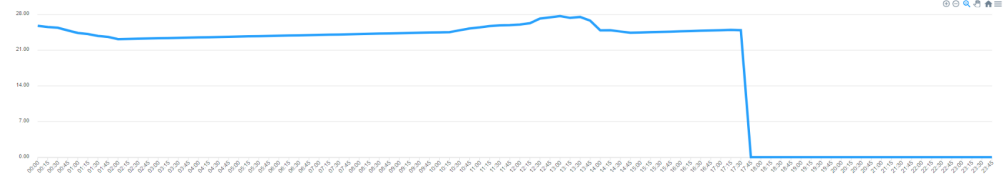
Fecha historico

22/06/2025

Intervalo en min (ej: 15)

15min

Temperatura DS18B20



Nivel de Agua 1



Figura 32: Pantalla histórico

Menu

🏠 Panel de control

⚙️ Administración ▾

General

Sensores

Usuarios

Plantas

Log

📊 Historico

🔌 Desconexión

Configuración de usuarios

Usuarios registrados

+ Nuevo Usuario

Ver todos los usuarios

ID	Nombre	Email	Teléfono
1	ADMINISTRADOR DEL SISTEMA	admin@arduino.com	555 55 55 66
3	Alberto	alberto@arduino.com	555 55 55 66
4	Miempresa	miempresa128@gmail.com	555 55 55 55

Aplicar

Resetear

Editar usuario ADMINISTRADOR DEL SISTEMA

Nombre

ADMINISTRADOR DEL SISTEMA

Teléfono

555 55 55 66

☒ Activo

☒ Administrador

☒ Recibe alarmas

Email

admin@arduino.com

Password

Figura 33: Pantalla configuración de usuarios

9.3 Comunicación y funcionamiento general

El flujo general del sistema es el siguiente:

1. Arduino toma lecturas de los sensores cada cierto tiempo.
2. Los datos se muestran en pantalla LCD y se envían al servidor vía WiFi mediante HTTP.
3. El backend recibe la información, la guarda en base de datos y verifica si es necesario actuar.
4. Si algún parámetro está fuera de los rangos definidos, se genera una alarma o se activa la bomba automáticamente.
5. El usuario puede consultar toda la información y modificar configuraciones desde la plataforma web.

Este proceso garantiza una automatización básica con supervisión continua por parte del usuario.

El flujo general del sistema, junto con información de los archivos del backend y el frontend se encuentra mejor detallado en el anexo Anexo 5: Documentación técnica.

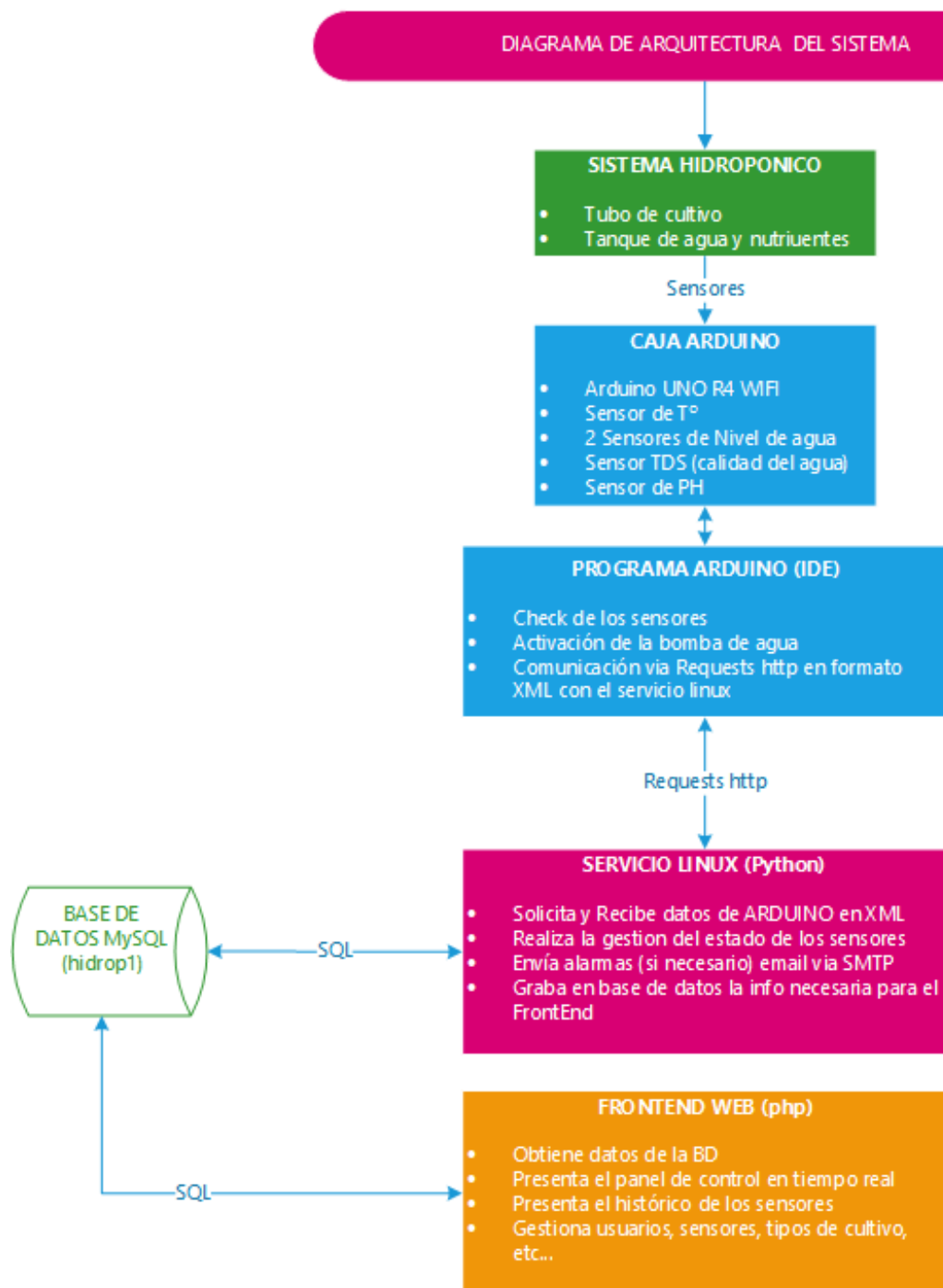


Figura 34: Diagrama de arquitectura del sistema

10. Análisis del sistema

10.1 Objetivos del sistema

Los objetivos principales del sistema hidropónico inteligente basado en IoT son los siguientes:

- OBJ-001. Permitir el registro, autenticación y gestión de usuarios con distintos roles (usuario y administrador).
- OBJ-002. Monitorizar en tiempo real parámetros ambientales relevantes para el cultivo hidropónico (temperatura, pH, nivel de agua, conductividad eléctrica).
- OBJ-003. Almacenar y visualizar los datos históricos obtenidos por los sensores del sistema.
- OBJ-004. Configurar límites críticos para cada parámetro, personalizables por planta, con generación automática de alertas cuando se superen.
- OBJ-005. Automatizar la actuación del sistema (bomba de agua) en función de las condiciones medidas y configuradas.
- OBJ-006. Facilitar una interfaz web accesible y clara que permita a los usuarios gestionar la información y supervisar el sistema sin conocimientos técnicos avanzados.

10.2 Actores del sistema

- **Usuario registrado**
 - Puede iniciar sesión y consultar los valores actuales e históricos de los sensores.
 - Puede visualizar las configuraciones de cultivo (valores críticos por planta).
 - Puede visualizar las configuraciones de usuarios.
 - Puede visualizar el registro de logs del sistema.
 - Puede recibir notificaciones y alarmas.
- **Administrador**
 - Tiene acceso a todas las funciones de usuario registrado.
 - Puede crear, editar o desactivar otros usuarios.
 - Puede crear, editar o desactivar otros cultivos.
 - Puede modificar directamente los parámetros del sistema y configuraciones globales.
 - Puede consultar el registro de logs del sistema.
- **Sistema de sensorización (*hardware embebido*)**
 - Actor no humano que representa la interacción entre el microcontrolador y los sensores.
 - Envía datos periódicos al servidor backend.
 - Ejecuta acciones automáticas (ej. activar bomba) cuando se supera un umbral.
- **Backend del sistema**
 - Procesa la lógica de negocio.
 - Almacena y gestiona la base de datos.
 - Genera alarmas, registros y notificaciones según las condiciones del sistema.

10.3 Requisitos del sistema

Para garantizar que el sistema hidropónico inteligente cumpla con los objetivos definidos, se han identificado distintos tipos de requisitos. Estos se clasifican en tres grupos: funcionales, no funcionales e informativos. A continuación se detallan cada uno de ellos.

10.3.1 Requisitos no funcionales

Los requisitos no funcionales especifican los criterios de calidad que debe cumplir el sistema, como rendimiento, seguridad, disponibilidad o escalabilidad. No describen funcionalidades concretas, sino propiedades generales que el sistema debe garantizar para ser aceptable.

ID	Requisito no funcional	Descripción breve
RNF-001	Usabilidad	La plataforma debe contar con una interfaz clara e intuitiva, accesible desde distintos navegadores.
RNF-002	Rendimiento	La carga de datos en tiempo real debe tardar pocos segundos bajo condiciones normales.
RNF-003	Portabilidad	El sistema debe ser capaz de ejecutarse en múltiples sistemas operativos sin cambios.
RNF-004	Mantenibilidad	El código debe estar estructurado de forma modular para permitir actualizaciones o mejoras futuras.
RNF-005	Fiabilidad del sistema físico	Ante fallos puntuales de sensores, el sistema debe continuar funcionando con los restantes.
RNF-006	Compatibilidad técnica	Los componentes <i>hardware</i> deben ser compatibles entre sí.
RNF-007	Accesibilidad básica	La interfaz debe ser legible para personas sin conocimientos técnicos.

Tabla 2: Requisitos no funcionales

Las especificaciones completas se encuentran detalladas en el Anexo 1: Especificaciones del sistema.

10.3.2 Requisitos de información

Los requisitos de información describen los datos que el sistema debe almacenar, gestionar y utilizar. Esta información es crítica para garantizar la trazabilidad, el control y la personalización del sistema.

ID	Requisito de información	Descripción breve
IRQ-001	Alertas y notificaciones	Mensajes generados cuando un sensor detecta valores fuera de lo normal.
IRQ-002	Plantas y parámetros asociados	Información de las plantas y sus rangos de valores óptimos.
IRQ-003	Sensor de nivel de agua	Lecturas periódicas de nivel en cada punto del sistema.
IRQ-004	Sensor de pH	Lecturas de pH del agua.
IRQ-005	Sensor de temperatura del agua	Lecturas de temperatura del agua desde un sensor de temperatura DS18B20.
IRQ-006	Sensor de conductividad	Datos sobre TDS en el agua para estimar nutrientes.
IRQ-007	Datos de usuario	Nombre, correo electrónico, contraseña,, rol (usuario o administrador).
IRQ-008	Configuración de cultivos	Rangos de temperatura, pH, TDS y nivel de agua específicos por cultivo.

Tabla 3: Requisitos de información

Las especificaciones completas se encuentran detalladas en el Anexo 1: Especificaciones del sistema.

10.3.3 Requisitos funcionales

Los requisitos funcionales especifican las capacidades que el sistema debe ofrecer a los usuarios y administradores para cumplir con los objetivos planteados. Estas funcionalidades se han agrupado y representado mediante casos de uso, que describen las acciones que cada actor puede realizar al interactuar con la plataforma.

Estos casos de uso han sido agrupados en cuatro bloques temáticos principales:

- Gestión de usuarios.
- Gestión del sistema y cultivos.
- Visualización de datos.
- Notificaciones y alertas.

Cada caso de uso refleja una funcionalidad concreta que puede ser ejecutada por uno o varios actores del sistema. Se han agrupado en bloques temáticos relacionados con la gestión de usuarios, la administración del sistema y cultivos, la visualización de datos y la gestión de alertas y notificaciones.

ID	Caso de uso	Descripción breve
CU-001	Iniciar sesión	Permite a los usuarios autenticarse en la plataforma.
CU-002	Visualizar usuarios	Visualización de los usuarios registrados en el sistema.
CU-003	Crear usuario	El administrador puede añadir nuevos usuarios.
CU-004	Editar usuario	Modificación de datos de un usuario existente por el administrador.
CU-005	Desactivar usuario	Permite al administrador deshabilitar el acceso de un usuario.
CU-006	Crear/Añadir cultivo	Añadir una nueva configuración de planta al sistema.
CU-007	Editar cultivo	Modificar la información de un cultivo existente.
CU-008	Desactivar cultivo	Inhabilitar temporalmente un cultivo del sistema.
CU-009	Configurar sensores y actuador	Definir los límites de los sensores y comportamiento del actuador.
CU-010	Editar configuración general del sistema	Modificar opciones globales del sistema como el umbral de actuación.
CU-011	Consultar datos de sensores en tiempo real	Visualización instantánea de los valores que miden los sensores.

CU-012	Consultar históricos de sensores	Acceso a registros de datos pasados de los sensores.
CU-013	Visualizar configuración de cultivos	Ver los valores límite de los sensores configurados por cada planta.
CU-014	Visualizar configuración del sistema	Consultar parámetros globales del sistema.
CU-015	Visualizar logs	Acceso al historial de eventos del sistema (acciones, errores).
CU-016	Recibir notificaciones del sistema	Recepción automática de alertas por correo cuando se detectan anomalías.
CU-017	Consultar historial de alertas	Visualización de todas las alarmas generadas por el sistema.
CU-018	Cerrar sesión	Permite a los usuarios desconectarse de la plataforma.

Tabla 4: Requisitos funcionales

Todas las funcionalidades quedan reflejadas mediante los casos de uso del sistema, organizados por tipo de gestión (usuarios, sistema, sensores, etc.). Las especificaciones completas se encuentran detalladas en el Anexo 1: Especificaciones del sistema.

11. Seguridad del sistema

En el desarrollo de un sistema inteligente basado en IoT, incluso en entornos cerrados o de uso limitado, la seguridad continúa siendo un factor importante. En este proyecto, la arquitectura se ha diseñado para operar en red local, sin exposición directa a internet, lo que proporciona una protección natural frente a ataques externos y permite un mayor control sobre el entorno de ejecución. Esta decisión arquitectónica reduce la superficie de ataque y simplifica las medidas de protección necesarias, tal y como se explica con mayor detalle en el Anexo 4: Plan de Seguridad.

A nivel lógico, el sistema gestiona dos tipos de usuarios: administrador y usuario registrado, con diferentes niveles de acceso. El control de acceso se realiza mediante credenciales validadas en base de datos, y la interfaz está adaptada a cada rol. Si bien no se implementan mecanismos avanzados como el cifrado de contraseñas o autenticación mediante tokens³, esto se justifica por tratarse de un entorno cerrado y sin usuarios concurrentes.

Desde el punto de vista físico, se ha prestado atención a la protección del *hardware*. Los elementos electrónicos están organizados en una caja cerrada y se han utilizado etiquetas para reducir errores y riesgos por manipulación accidental.

En cuanto al desarrollo, se han aplicado buenas prácticas como la gestión de roles en el backend y el uso de logs de actividad para el seguimiento de acciones.

Para una descripción más exhaustiva de las medidas y enfoques adoptados, puede consultarse el Anexo 4: Plan de Seguridad.

³ Método de verificación en el que, tras iniciar sesión, el servidor genera un token (una cadena única) que se entrega al usuario y se usa para validar sus futuras peticiones sin necesidad de enviar las credenciales cada vez.

12. Análisis de riesgos

En el desarrollo de cualquier sistema tecnológico, especialmente aquellos que integran *hardware*, *software* embebido y plataformas web, es esencial realizar un análisis de riesgos que permita anticipar posibles problemas y diseñar medidas preventivas. Este proyecto, al abarcar múltiples capas (sistema físico, sensores, microcontrolador, backend y frontend web), está expuesto a riesgos tanto técnicos como operativos. Por ello, se ha llevado a cabo un análisis DAFO [14] que permite identificar fortalezas, debilidades, amenazas y oportunidades vinculadas al sistema desarrollado.

Entre las principales **fortalezas** del sistema desarrollado destaca su bajo coste económico, lo que permite su implementación en entornos domésticos, educativos o experimentales sin necesidad de realizar grandes inversiones. Gracias al uso de componentes accesibles como el microcontrolador Arduino UNO R4 WiFi, sensores comunes (pH, TDS, temperatura, nivel de agua) y una estructura física sencilla, se ha conseguido construir un sistema funcional por un precio aproximado de 70–90 euros, como se detalla en el apartado de viabilidad económica.

Otra de las fortalezas más relevantes es la arquitectura modular y extensible del sistema, tanto a nivel físico como lógico. La plataforma está diseñada para permitir el reemplazo, ampliación o mejora de sensores sin rediseñar el conjunto completo, lo que facilita su mantenimiento, evolución o adaptación a distintos tipos de cultivo.

Sin embargo, el sistema también presenta algunas **debilidades** propias de su carácter prototipo. La más relevante es la limitación en la seguridad avanzada, ya que no se incluyen mecanismos como el cifrado de contraseñas o la protección frente a ataques CSRF⁴ (*Cross-Site Request Forgery*) debido a que el entorno está restringido a una red local sin exposición pública. Además, el sistema depende de ciertos componentes económicos que, si bien reducen el coste, pueden tener una durabilidad limitada, como ocurrió durante las fases de prueba con sensores que debieron ser reemplazados.

Entre las **oportunidades** más relevantes se encuentra la posibilidad de escalar la solución mediante la integración de nuevos sensores, o incluso el uso de técnicas de predicción y optimización basadas en inteligencia artificial. También existe potencial para su despliegue en entornos educativos, ya que el sistema utiliza tecnologías de código abierto y es fácilmente replicable. Además, el auge de la agricultura urbana y los huertos inteligentes supone un contexto favorable para la expansión de este tipo de soluciones tecnológicas.

Por último, también se deben contemplar algunas **amenazas** externas. La obsolescencia tecnológica⁵ puede afectar tanto al *hardware* como a las dependencias del *software* si no se mantiene actualizado. Asimismo, en caso de querer abrir el sistema a Internet o escalar su uso a múltiples usuarios concurrentes, se requeriría una revisión profunda de la arquitectura y un refuerzo de sus medidas de seguridad.

⁴ Tipo de ataque en el que un atacante engaña a un usuario autenticado para que ejecute acciones no deseadas en una aplicación web en la que está logueado, aprovechando su sesión activa.

⁵ Proceso por el cual un componente, sistema o tecnología queda desactualizado o ineficiente frente a nuevas alternativas, perdiendo soporte, compatibilidad o utilidad con el tiempo.

13. Estimación del esfuerzo y planificación temporal

Durante el desarrollo del proyecto se ha llevado a cabo una estimación detallada del esfuerzo requerido, basada en la métrica Use Case Points (UCP) [15]. Esta metodología permite estimar el tamaño y la carga de trabajo de un proyecto *software* a partir de los casos de uso definidos en su análisis funcional [16]. En el Anexo 3: Estimación del tamaño y esfuerzo se detalla el proceso completo seguido, incluyendo los factores técnicos y ambientales considerados, el cálculo de puntos de caso de uso, y la conversión de estos en horas de desarrollo estimadas.

La planificación temporal se ha elaborado mediante la herramienta Microsoft Project, permitiendo definir tareas, iteraciones, dependencias y periodos de pausa. En las siguiente figuras se muestran una versión reducida del cronograma del proyecto, con la secuencia de tareas principales desde noviembre de 2024 hasta julio de 2025, y el diagrama de Gantt resultante.

	▷ Inicio	48 días	mié 20/11/24	vie 24/01/25
	Hito fin Inicio	0 días	vie 24/01/25	vie 24/01/25
	▷ Diseño del sistema	15 días	lun 27/01/25	vie 14/02/25
	Hito fin Diseño del sistema	0 días	vie 14/02/25	vie 14/02/25
	▷ Montaje físico y electrónica	21 días	lun 17/02/25	lun 17/03/25
	Hito fin Montaje físico y electrónica	0 días	lun 17/03/25	lun 17/03/25
	▷ Backend y base de datos	20 días	mar 18/03/25	lun 14/04/25
	Hito fin Backend y base de datos	0 días	lun 14/04/25	lun 14/04/25
	▷ Frontend	19 días	mar 15/04/25	vie 09/05/25
	Hito fin Frontend	0 días	vie 09/05/25	vie 09/05/25
	▷ Integración, pruebas y cultivo	10 días	lun 12/05/25	vie 23/05/25
	Hito fin Integración, pruebas y cultivo	0 días	vie 23/05/25	vie 23/05/25
	▷ Documentación y entrega	17 días	jue 12/06/25	vie 04/07/25
	Hito fin Documentación y entrega	0 días	vie 04/07/25	vie 04/07/25

Figura 35: Planificación temporal

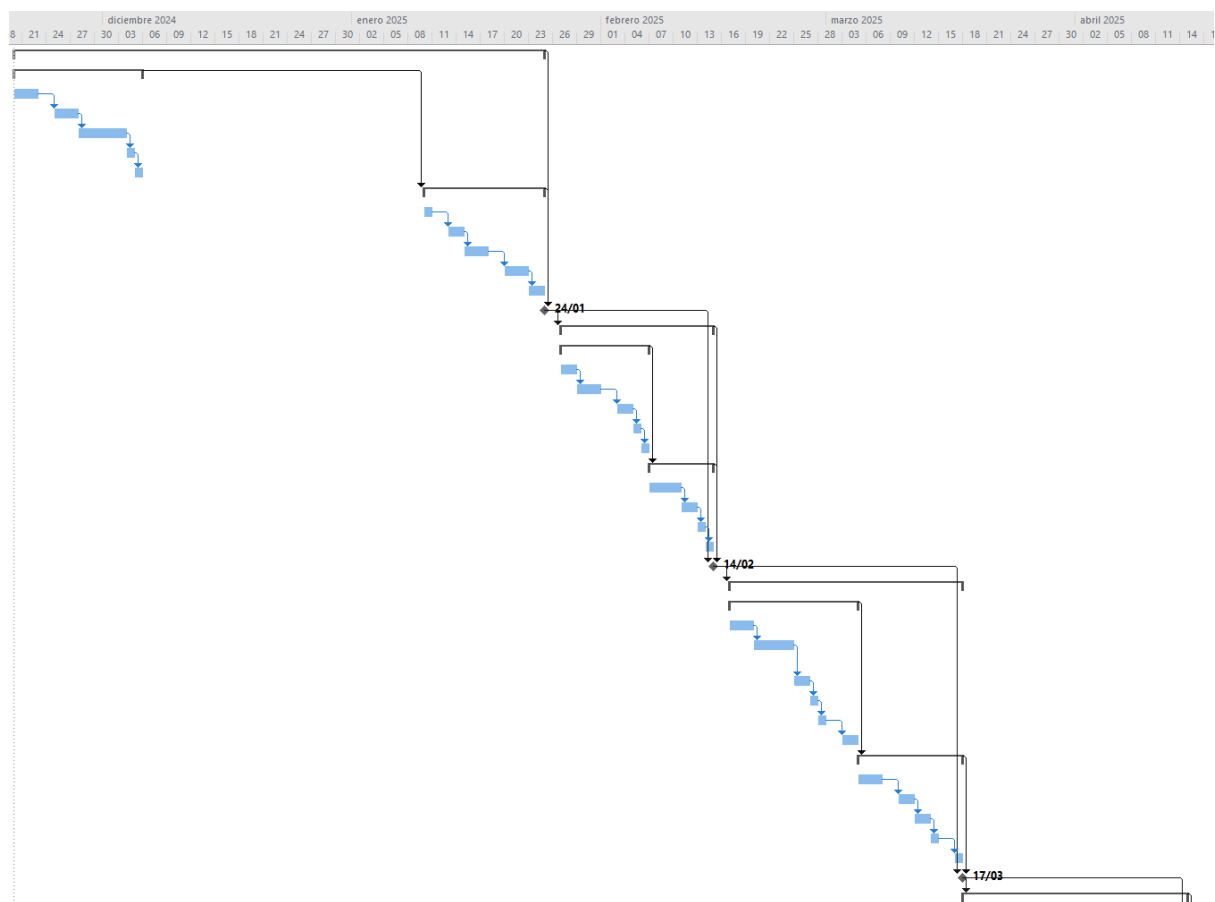


Figura 36: Diagrama Gantt 1

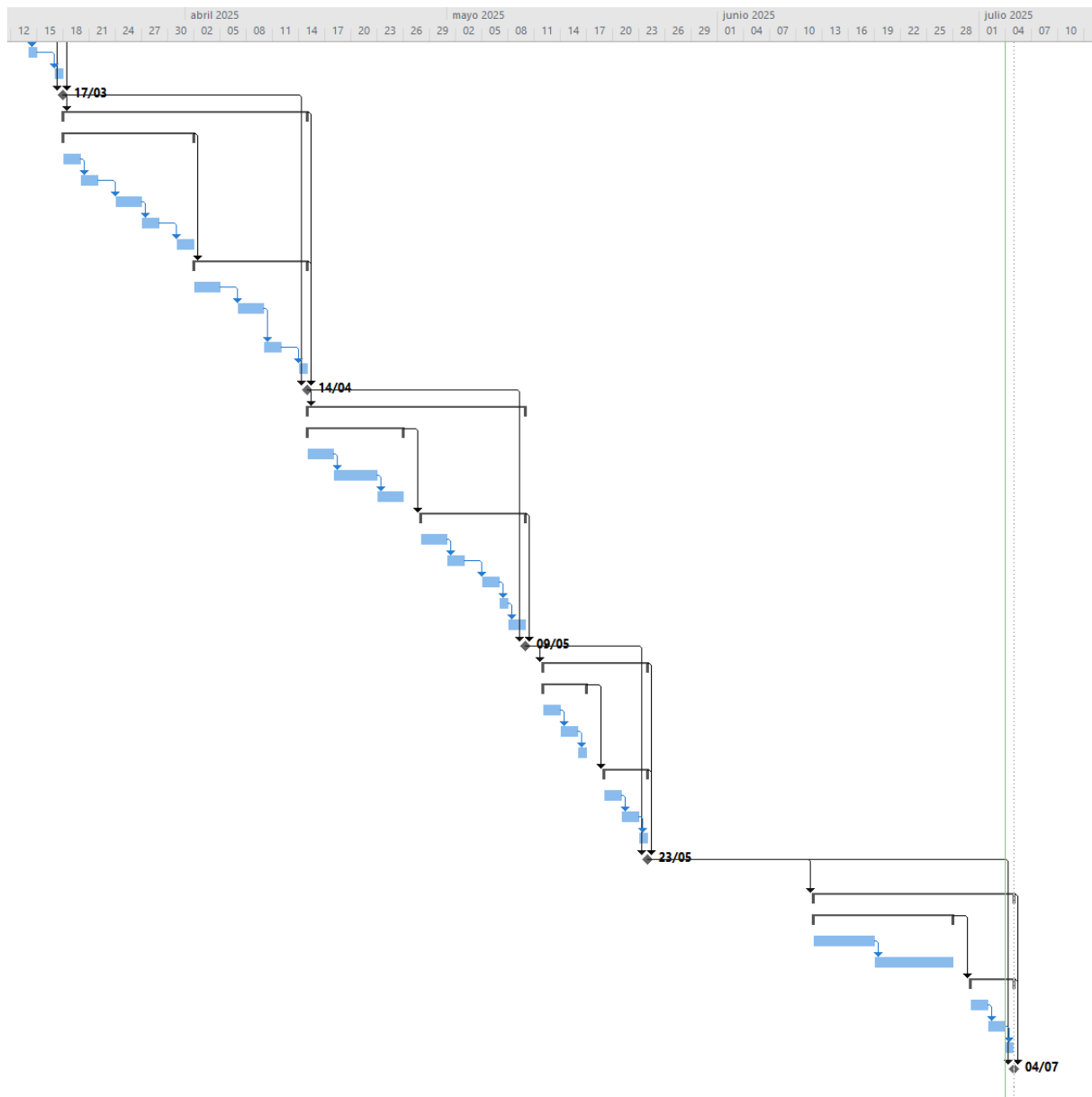


Figura 37: Diagrama Gantt 2

Cabe destacar que durante el desarrollo del proyecto hubo dos periodos de pausa, tal como se ha mencionado anteriormente, motivados por la dedicación plena a la preparación de exámenes y trabajos finales correspondientes a un máster universitario.

La versión completa de la estimación UCP y la planificación temporal se encuentra documentada en el Anexo 3: Estimación del tamaño y esfuerzo.

14. Conclusiones y trabajo futuro

Este proyecto ha permitido el diseño y desarrollo de un sistema inteligente de monitorización y control para un cultivo hidropónico, combinando *hardware* de bajo coste, sensores específicos, un microcontrolador programado en Arduino y una plataforma web desarrollada en tecnologías abiertas. El proyecto ha conseguido integrar todos estos elementos en una solución funcional y escalable, capaz de registrar parámetros del entorno de cultivo, emitir alertas y activar acciones automáticas para su mantenimiento.

La arquitectura implementada, tanto a nivel físico como lógico, permite una separación clara de responsabilidades entre el sistema embebido y la plataforma *web*, facilitando así el mantenimiento y futuras ampliaciones.

Asimismo, se han cumplido los objetivos principales marcados al inicio del proyecto: el sistema es adaptable a distintos tipos de plantas, configurable, y ofrece una experiencia clara tanto para administradores como para usuarios registrados.

No obstante, existen algunas limitaciones que podrían abordarse en futuras versiones. Por ejemplo, mejorar la precisión y durabilidad de los sensores utilizados mediante la incorporación de modelos industriales o sensores multifunción integrados. También sería interesante habilitar la gestión remota mediante acceso seguro desde fuera de la red local, incorporando mecanismos de autenticación reforzada. Por otro lado, la inclusión de algoritmos de análisis predictivo mediante técnicas de inteligencia artificial podría aportar valor añadido, permitiendo anticipar necesidades de riego o detectar patrones anómalos en los datos históricos.

Finalmente, desde una perspectiva más aplicada, el sistema tiene un gran potencial para ser utilizado en entornos educativos o demostrativos, donde estudiantes o usuarios sin experiencia técnica puedan comprender el funcionamiento de un ecosistema tecnológico completo aplicado a la agricultura inteligente.

15. Referencias

1. Food and Agriculture Organization of the United Nations (FAO). The Future of Food and Agriculture – Trends and Challenges. FAO, 2017. <https://www.fao.org/3/i6583e/i6583e.pdf>
2. Barranco, I. O., Ibarra, K. J. C., Peña, L. A. M., & Cuadra, A. N. (2024). SISTEMA IOT HIDROPONICO. www.anfei.mx. <https://www.anfei.mx/revista/index.php/revista/article/view/1005>
3. Aguayo, H. (2025, 8 febrero). El crecimiento del mercado de hidroponía: innovación y oportunidades de inversión. <https://www.linkedin.com/pulse/el-crecimiento-del-mercado-de-hidropon%C3%ADa-innovaci%C3%B3n-y-aguayo-rivera-1jmle/>
4. Home | Aerospring Hydroponics. (s. f.). Aerospring Hydroponics. <https://aerospringhydroponics.com/>
5. LetPot's garden. (s. f.). LetPot® indoor Garden, Hydroponic Smart Planter with App Control. LetPot's Garden. <https://letpot.com/>
6. colaboradores de Wikipedia. (2025, 16 mayo). Hidroponía. Wikipedia, la Enciclopedia Libre. <https://es.wikipedia.org/wiki/Hidropon%C3%ADa>
- ECOagricultor. (2021, 23 mayo). Hidroponía: Qué es y cuáles son las Ventajas y Desventajas de los Cultivos Hidropónicos. ECOagricultor. <https://www.ecoagricultor.com/hidroponia-que-es-y-ventajas-desventajas-cultivos-hidroponicos/>
- Iberdrola. (2021, 22 abril). QUÉ ES LA HIDROPONÍA y SUS VENTAJAS. Iberdrola. <https://www.iberdrola.com/sostenibilidad/que-es-hidroponia-y-ventajas>
7. Arduino.cc. “What is Arduino?” Disponible en: <https://www.arduino.cc/en/Guide/Introduction>
8. IoT in Agriculture: Improving Efficiency and Sustainability – Azeosoft. (s. f.). <https://azeosoft.com/blog/iot-in-agriculture-improving-efficiency-and-sustainability/#:~:text=Answer%3A%20IoT%20technology%20benefits%20agriculture%20by%20enabling%20precision.improve%20productivity%2C%20and%20enhance%20sustainability%20in%20farming%20practices.>
9. Ruiz, C. (2018, 13 noviembre). Cultivo hidropónico inteligente controlado por Arduino – Open Lanuza. <https://openlanuza.com/cultivo-hidroponico-inteligente-controlado-por-arduino/>
10. Projects | Kyle Gabriel. (s. f.). <https://kylegabriel.com/projects/>

Kyle Gabriel. (2020, 3 junio). Build an Automated Hydroponic System [Vídeo]. YouTube. <https://www.youtube.com/watch?v=nyqykZK2Ev4>

11. fbustos@yc.cl. (2025, 10 febrero). Diseño de un sistema de riego hidropónico automático asistido por un hardware arduino - PortalFruticola.com. Portal Frutícola. <https://www.portalfruticola.com/noticias/2018/01/10/disenio-de-un-sistema-de-riego-hidroponico-automatico-asistido-por-un-hardware-arduino/>
12. NIDO PRO. Agrotec. Disponible en: <https://www.agrotec.com/producto/nidopro-sistema-hidroponico-inteligente-cultivo-vertical/>
13. Rocket24 NFT. Plantawa. Disponible en: <https://plantawa.com/sistema-de-cultivo-hidroponico-nft-rocket24.html>
14. Infoautonomos. (2024, 16 septiembre). Analisis DAFO: qué es y ejemplo. <https://www.infoautonomos.com/plan-de-negocio/analisis-dafo/>
15. Estimación por Puntos de Caso de Uso (el famoso Karner). (2010, 5 noviembre). Un Poco de Java. <https://unpocodejava.com/2010/11/06/estimacion-por-puntos-de-caso-de-uso-el-famoso-karner/>
16. Estimación de software basada en puntos de casos de uso. Ejemplo práctico. (2011, 22 julio). Monografias.com. <https://www.monografias.com/trabajos87/estimacion-software-basada-puntos/estimacion-software-basada-puntos>