**NORMAS BÁSICAS DE ESTILO Y FORMATO**

El texto del trabajo se escribirá preferentemente en español. Se admitirá el idioma inglés, siempre que el trabajo esté avalado por un convenio internacional (Erasmus u otros) o si la Comisión delegada de Trabajos Fin de Grado o la Comisión Académica del Máster, según el caso, informe favorablemente a partir de una solicitud previa del alumno o alumna.

**PROYECTOS DE INGENIERÍA**

Se seguirán las consideraciones indicadas en la norma UNE 157001:2004 en cuanto a requisitos formales de carácter general con qué deben redactarse los proyectos.

Se utilizará la carátula incluida en este anexo.

Serán elementos de obligada aparición los siguientes:

**Resumen.** Entre 300 y 500 palabras como referencia.

**Documento Nº 1. Índice**

**Documento Nº 2. Memoria.** Tiene como misión justificar las soluciones adoptadas, su adecuación a la normativa legal aplicable y, conjuntamente con los planos y el pliego de condiciones, debe describir de forma unívoca el objeto del Proyecto.

**Documento Nº 3. Anexos**

Este documento debe contener los anejos necesarios (según proceda en cada caso) correspondientes a:

* **Documentación de partida.** Este Anexo debe incluir aquellos documentos que se han tenido en cuenta para establecer los requisitos de diseño.
* **Cálculos**. Este Anexo o Anexos tienen como misión justificar las fórmulas aplicadas, las soluciones adoptadas y, conjuntamente con los documentos planos y el pliego de condiciones, debe describir de forma unívoca el objeto del Proyecto.

Debe contener las hipótesis de partida, los criterios y procedimientos de cálculo, así como los resultados finales base del dimensionado o comprobación de los distintos elementos que constituyen el objeto del Proyecto.

* Anexos de aplicación en función del ámbito del Proyecto, son por ejemplo:
  + Seguridad (prevención de incendios, sanidad, radiaciones, pública concurrencia, etc.).
  + Medio ambiente (acústica, residuos, emisiones, etc.).
  + Eficiencia energética.
  + Emplazamiento del proyecto, Geotécnicos, Hidráulicos, Hidrológicos, Pluviométricos, etc.
  + Gestión de residuos.
  + Otros.
* **Estudios con entidad propia.** Este documento debe contener todos aquellos estudios que deban incluirse en el Proyecto por exigencias legales. Debe comprender, entre otros y sin carácter limitativo, los relativos a:
  + Estudio Básico de Seguridad y Salud o Estudio de Seguridad y Salud, según corresponda.
  + Estudio de Impacto Ambiental. Se realizará según la tipología prevista en la legislación vigente que aplique.

Cada anexo debe contener la justificación del cumplimiento de la normativa legal vigente aplicable y, si procede, de las fórmulas aplicadas para el cálculo.

* **Otros documentos que justifiquen y aclaren conceptos expresados en el Proyecto.** Se pueden incluir:
  + Catálogos de los elementos constitutivos del objeto del Proyecto.
  + Listados.
  + Información en soportes lógicos, magnéticos, ópticos o cualquier otro.
  + Maquetas o modelos.
  + Otros documentos que se juzguen necesarios.

**Documento Nº 4. Planos.** Se deberá tener en cuenta lo indicado en las normas referencias en el apartado 8.2 de la norma UNE 157001:2014.

**Documento Nº 5. Pliego de Condiciones.** El pliego de condiciones es uno de los documentos que constituyen el Proyecto y tiene como misión establecer las condiciones técnicas, económicas, administrativas, facultativas y legales para que el objeto del Proyecto pueda materializarse en las condiciones especificadas, evitando posibles interpretaciones diferentes de las deseadas.

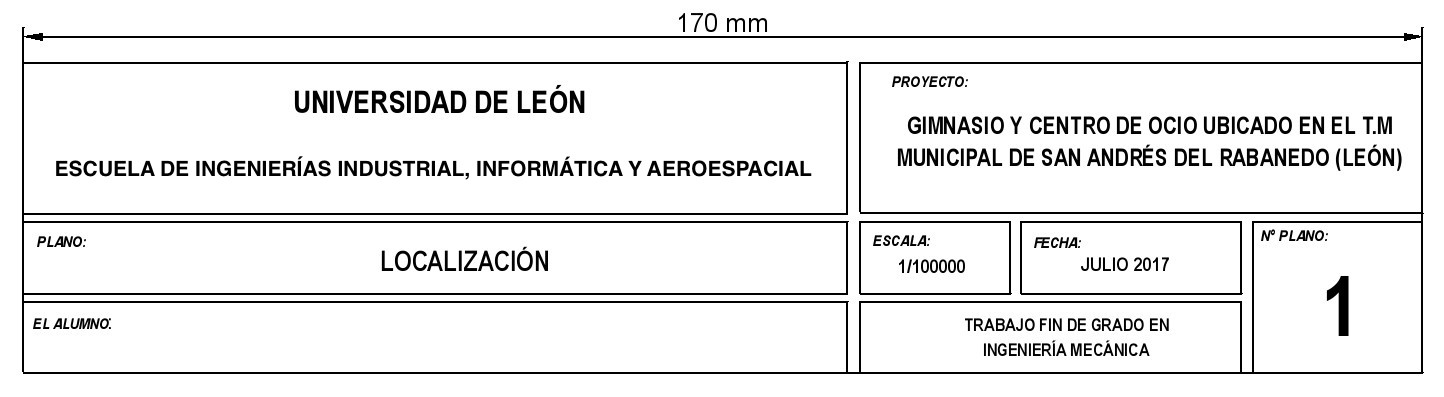
**Documento Nº 6. Mediciones.** Debe contener un listado completo de las partidas de obra que configuran la totalidad del proyecto. Debe servir de base para la realización del Presupuesto.

**Documento Nº 7. Presupuesto.**

MODELO DE CAJETÍN PARA PLANOS

Anchura máxima 170 mm, y altura variable. No habría límites si el plano no se doblase.

Situación en ángulo inferior derecho del área de dibujo del plano, con el mismo grosor que el recuadro que marca la zona efectiva de dibujo.



**ESTUDIOS TÉCNICOS, ORGANIZATIVOS Y ECONÓMICOS, Y TRABAJOS TEÓRICO-EXPERIMENTALES**

Se utilizará la carátula incluida en este anexo. El tamaño de papel será A4 y los márgenes del documento serán 3 cm en la parte superior y a la izquierda y 2.5 cm en la parte inferior y a la derecha.

Se utilizarán encabezados y pies de página. En el pie de página irá el nombre del alumno-autor del trabajo y en el encabezado irá la numeración de las hojas, indicándose el total.

Para la numeración de los capítulos, apartados y anexos se seguirán las recomendaciones de la norma UNE 50132.

Cada capítulo iniciará una página nueva. Los títulos de los capítulos utilizarán una fuente de tamaño 28 puntos en negrita y llevarán la primera letra en mayúscula, mientras que el resto serán minúsculas. El interlineado (si el título ocupa más de una línea) será sencillo. Después del título irá una línea en blanco de tamaño 12 puntos.

Para los títulos de los apartados, se utilizará un tamaño de 14 puntos, letra negrita y mayúscula. Los títulos de los apartados y sub-apartados irán precedidos del número del apartado. Debajo del título de los apartados se dejará una línea en blanco.

Para el cuerpo del texto se utilizará un tamaño de 12 puntos, justificado a izquierda y derecha y con interlineado igual a 1.5 líneas, sin espacio adicional entre párrafos.

Las figuras y tablas se intercalarán en el texto, centradas y con un pie de figura o encabezado de tabla. Las figuras y tablas llevarán una línea antes y otra después, y la forma de nombrarlas en el texto será “figura #.#” o “tabla #.#”, donde el primer número es el del capítulo y el segundo es el número de orden de la figura o de la tabla dentro de ese capítulo. El texto al pie de la figura o en el encabezado de la tabla estará en escrito con un tamaño de 12 puntos, y deberá señalar lo que se representa de una forma clara. Ejemplo:

Figura 4.1. Relación entre … (Fuente: ….)

Tabla 4.1. Tabla con resultados de…(Fuente: ….)

Si el texto requiere la inclusión de ecuaciones, el formato de las mismas es libre. Se numerarán según el formato (#.#) alineado al margen derecho junto a la ecuación y se citarán en el cuerpo del texto como “ecuación #.#”, donde el primer número es el capítulo y el segundo el número de la ecuación. Ejemplo:

… (4.1)

Al final del cuerpo del trabajo debe incluirse una lista de todas las fuentes bibliográficas en las que se basa éste, y debe hacerse referencia a cada uno de los elementos de esta lista en todos los lugares adecuados dentro del texto. A pesar de que la norma UNE 50135 incluye cuestiones relativas al formato de las referencias bibliográficas, se usarán prioritariamente los estilos de referenciación:

* HARVARD (Autor-Fecha)
* IEEE (Numérico)

En todo caso, se podrá usar cualquier otro formato siempre que esté normalizado.

Serán elementos de obligada aparición los siguientes:

**Resumen.** Entre 300 y 500 palabras, en español.

**Abstract.** Entre 300 y 500 palabras, en inglés.

**Índice de contenidos**

**Índice de figuras**

**Índice de cuadros y tablas**

**Glosario de signos, símbolos, unidades, abreviaturas, acrónimos o términos** (si hubiera)

**Cuerpo del trabajo (recomendación):**

* Introducción
* Estado del arte
* Núcleo del trabajo con ilustraciones esenciales y tablas (objetivos, materiales y métodos,…)
* Resultados
* Conclusiones y recomendaciones
* Agradecimientos, si los hubiere

**Lista de referencias bibliográficas**

**Anexos**

****

Escuela de Ingenierías

Industrial, Informática y Aeroespacial

**GRADO EN INGENIERÍA INFORMÁTICA**

Trabajo de Fin de Grado

Aplicación web para la gestión de cultivos hidropónicos

Web application for hydroponic crop management

Autor: José Antonio López Pérez

Tutores: Rubén Rodríguez, Jesús Fernández

(Julio, 2025)

|  |  |
| --- | --- |
| UNIVERSIDAD DE LEÓN  Escuela de Ingenierías Industrial, Informática y Aeroespacial  GRADO EN INGENIERÍA INFORMÁTICA  Trabajo de Fin de Grado | |
| **ALUMNO:** José Antonio López Pérez | |
| **TUTOR:** Rubén Rodríguez, Jesús Fernández | |
| **TÍTULO:** Aplicación web de gestión de cultivos hidropónicos | |
| **TITLE:** Web application for hydroponic crop management | |
| **CONVOCATORIA:** Julio, 2025 | |
| **RESUMEN:**  En un contexto de creciente interés por los sistemas de cultivo sostenible y la optimización de recursos, la hidroponía se presenta como una alternativa eficiente para la producción de alimentos en espacios reducidos y con un menor consumo de agua. Sin embargo, uno de los principales retos de estos sistemas es la necesidad de un monitoreo constante y una gestión precisa de variables como la temperatura, la humedad o los niveles de nutrientes. Automatizar y facilitar este control resulta fundamental para asegurar la viabilidad del cultivo y mejorar su rendimiento.  Este Trabajo de Fin de Grado tiene como objetivo principal el diseño y desarrollo de una aplicación web dedicada a la gestión de módulos hidropónicos inteligentes. La aplicación permite a los usuarios monitorizar en tiempo real los datos capturados por distintos sensores instalados en el sistema, además de programar acciones automáticas basadas en condiciones específicas o preferencias del usuario. De este modo, es posible optimizar el mantenimiento del cultivo mediante la automatización de tareas como el riego, la ventilación o el ajuste de la iluminación. La plataforma está organizada en diferentes secciones que ofrecen funcionalidades como la visualización gráfica de los datos registrados, el estado actual del sistema, recomendaciones personalizadas para mejorar el cultivo y paneles de configuración para adaptar el funcionamiento del módulo a las necesidades concretas del usuario.  Desde el punto de vista técnico, el sistema sigue una arquitectura basada en microservicios virtualizados mediante contenedores Docker, lo que garantiza su escalabilidad, portabilidad y facilidad de despliegue. Además, el proyecto incluye el diseño y la construcción de un módulo hidropónico físico, equipado con una serie de sensores y componentes electrónicos gestionados a través de una estructura fabricada mediante impresión 3D. Gracias a esta combinación de hardware y software, se ofrece una solución integral para el control inteligente de cultivos hidropónicos. | |
| **ABSTRACT:**  In a context of growing interest in sustainable farming systems and resource optimization, hydroponics emerges as an efficient alternative for food production in reduced spaces and with lower water consumption. However, one of the main challenges of these systems is the need for constant monitoring and precise management of variables such as temperature, humidity, and nutrient levels. Automating and facilitating this control isessential to ensure the viability of the crop and improve its performance.  This Final Degree Project aims to design and develop a web application dedicated to the management of intelligent hydroponic modules. The application allows users to monitor in real time the data captured by different sensors installed in the system, as well as to schedule automatic actions based on specific conditions or user preferences. In this way, it is possible to optimize crop maintenance through the automation of tasks such as irrigation, ventilation, or lighting adjustment. The platform is organized into different sections that offer functionalities such as graphical visualization of recorded data, the status of the system, personalized recommendations to improve the crop, and configuration panels to adapt the module's operation to the user's specific needs.  From a technical point of view, the system follows an architecture based on microservices virtualized through Docker containers, which ensures scalability, portability, and ease of deployment. Additionally, the project includes the design and construction of a physical hydroponic module, equipped with a series of sensors and electronic components managed through a structure manufactured by 3D printing. Thanks to this combination of hardware and software, a comprehensive solution is offered for the intelligent control of hydroponic crops. | |
| **Palabras clave:** Hidroponía, Automatización, Monitoreo, Sensores, Aplicación web, Microservicios, Docker, Impresión 3D, Optimización de recursos, Control inteligente | |
| **Firma del alumno:** | **VºBº Tutor/es:** |

Tabla de contenido

[Índice de figuras 9](#_Toc192241343)

[Índice de tablas 10](#_Toc192241344)

[Glosario de términos 10](#_Toc192241345)

[Introducción 11](#_Toc192241346)

[Estudio del problema 14](#_Toc192241347)

[El contexto del problema 14](#_Toc192241348)

[El estado de la cuestión 15](#_Toc192241349)

[La definición del problema 15](#_Toc192241350)

[Gestión de proyecto software 16](#_Toc192241351)

[2.1 Alcance del proyecto 16](#_Toc192241352)

[2.1.1 Definición del proyecto 16](#_Toc192241353)

[2.1.2 estimación de tareas y recursos 17](#_Toc192241354)

[2.1.3 Presupuesto 18](#_Toc192241355)

[2.2 Plan de trabajo 19](#_Toc192241356)

[2.2.1 Identificación de tareas 19](#_Toc192241357)

[2.2.2. Asignación de tareas y estimación 20](#_Toc192241358)

[2.2.3. Planificación de tareas 2.3. Gestión de recursos 20](#_Toc192241359)

[2.3.1. Especificación de recursos 20](#_Toc192241360)

[2.3.2. Asignación de recursos 20](#_Toc192241361)

[2.4. Gestión de riesgos 20](#_Toc192241362)

[2.4.1. Identificación de riesgos 20](#_Toc192241363)

[Solución 21](#_Toc192241364)

[3.1. Descripción de la solución 3.2. El proceso de desarrollo 21](#_Toc192241365)

[3.2.1. Análisis 3.2.2. Diseño 21](#_Toc192241366)

[3.3. El producto del desarrollo 21](#_Toc192241367)

[Evaluación 21](#_Toc192241368)

[4.1. Proceso de evaluación 21](#_Toc192241369)

[4.1.1. Forma de evaluación 21](#_Toc192241370)

[4.1.2. Casos de prueba 4.2. Análisis de resultados Conclusión 21](#_Toc192241371)

[Lista de referencias 21](#_Toc192241372)

[Anexo 1 - Seguimiento de Proyecto 21](#_Toc192241373)

[Anexo 2 - Manual de usuario 21](#_Toc192241374)

# Índice de figuras

[Ilustración 1 8](#_Toc190859315)

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.

Ilustración 1

# Índice de tablas

# Glosario de términos

**React**: Framework utilizado para la creación de aplicaciones web.

**API**: Conjunto de definiciones y protocolos que permiten que dos aplicaciones se comuniquen entre sí.

**Backend**: Parte de una aplicación que gestiona la lógica, la base de datos y el procesamiento de datos en el servidor.

**MariaDB**: Sistema de gestión de bases de datos relacional, derivado de MySQL, utilizado para almacenar y gestionar datos.

**DaisyUI**: Plugin de Tailwind CSS que proporciona componentes predefinidos y estilos personalizables para interfaces web.

**Docker**: Plataforma que permite crear, distribuir y ejecutar aplicaciones dentro de contenedores ligeros y portables.

**CSS**: Lenguaje de hojas de estilo usado para describir la presentación y diseño de documentos HTML.

**Arduino**: Plataforma de hardware libre basada en una placa con microcontrolador y un entorno de desarrollo para crear proyectos electrónicos.

**Software**: Conjunto de programas, instrucciones y reglas informáticas que permiten ejecutar tareas en un dispositivo.

**Hardware**: Conjunto de componentes físicos y tangibles de un sistema informático o electrónico

# Introducción

Planteamiento del problema

Este proyecto plantea una solución enfocada en el desarrollo de un producto software completo destinado a la gestión y automatización de módulos hidropónicos inteligentes. El objetivo principal es permitir a los usuarios visualizar la información recogida por los sensores instalados en el sistema y facilitar la programación de acciones automáticas que optimicen el mantenimiento del cultivo. A través del procesamiento y presentación adecuada de los datos capturados (como temperatura, humedad o electro conductividad del agua), se busca garantizar un control preciso que permita mejorar el rendimiento y la viabilidad de los cultivos hidropónicos.

Con el desarrollo del proyecto, se pretende analizar y aplicar metodologías ágiles durante la gestión del proceso, empleando un enfoque iterativo e incremental. Gracias a este modelo, el producto podrá ir incorporando nuevas funcionalidades de manera progresiva a lo largo de diferentes iteraciones, permitiendo que sea operativo desde las primeras fases y favoreciendo la obtención de feedback por parte de los usuarios. Esto facilita la detección temprana de errores y reduce los costes asociados a posibles cambios, además de asegurar una evolución continua del sistema a medida que se desarrollan nuevas versiones.

En cuanto a la arquitectura, se ha optado por un diseño basado en microservicios virtualizados mediante contenedores Docker, lo que permite dividir el sistema en diferentes componentes independientes, cada uno encargado de una funcionalidad específica, garantizando flexibilidad, escalabilidad y facilidad de mantenimiento. La comunicación entre estos microservicios es fundamental y se realiza a través de peticiones bien definidas que permiten integrar de manera eficiente todas las partes del sistema.

El producto final integra tanto el desarrollo de la aplicación web, que proporciona al usuario acceso a los datos y configuración del módulo hidropónico, como el diseño físico del propio módulo, fabricado mediante impresión 3D y equipado con sensores y componentes electrónicos. Gracias a esta combinación de hardware y software, se ofrece una solución completa que facilita el control inteligente y automatizado de cultivos hidropónicos, proporcionando una alternativa eficiente y sostenible para la producción de alimentos.

Objetivos

El objetivo principal del proyecto es desarrollar una aplicación web que permita la gestión y automatización de módulos hidropónicos inteligentes, facilitando a los usuarios la monitorización de datos en tiempo real y la programación de acciones automáticas para optimizar el mantenimiento del cultivo.

Además del objetivo principal, hay una serie de objetivos específicos:

* Comprender qué es un módulo hidropónico inteligente y cuáles son sus principales componentes.
* Comprender la arquitectura basada en microservicios y su aplicación en sistemas de gestión de datos IoT.
* Conocer los procedimientos y métodos para desarrollar una arquitectura basada en microservicios mediante Docker.
* Comprender el funcionamiento y la integración de sensores en sistemas de monitoreo ambiental (temperatura, humedad, nutrientes).
* Conocer los medios de comunicación entre los microservicios del sistema y cómo garantizar su correcta interacción.
* Evaluar la experiencia del usuario en la aplicación web, analizando la usabilidad y accesibilidad de las funcionalidades ofrecidas.
* Integrar hardware y software de forma eficiente para ofrecer una solución completa de gestión de cultivos hidropónicos.
* Conocer las ventajas que aporta el uso de un bus de datos a la hora de evitar pérdidas de información.
* Desarrollar un módulo hidropónico diseñado específicamente para la aplicación.
* Analizar los datos obtenidos y realizar recomendaciones para el usuario.

Metodología

Para el desarrollo de este proyecto se ha seguido una metodología ágil e incremental, gestionando el ciclo de vida del software a través de la plataforma GitHub como herramienta principal para el control de versiones y seguimiento del progreso. El trabajo se ha dividido en dos grandes etapas de aproximadamente tres meses cada una.

Durante la primera etapa, el objetivo principal fue obtener una primera versión del prototipo funcional, enfocada en desarrollar un Producto Mínimo Viable (MVP) que incluyera las características básicas necesarias para validar el funcionamiento del sistema. En esta fase inicial se priorizó la integración de los componentes esenciales del sistema, como la captura de datos desde sensores, la visualización básica de información en la aplicación web y la ejecución de acciones automáticas simples.

La planificación del desarrollo se organizó en iteraciones de dos semanas, al final de las cuales se incorporaba al proyecto al menos una nueva funcionalidad que permitiera avanzar de forma continua y medible. Además, esta estructura permitió realizar pruebas parciales y corregir posibles errores antes de continuar con nuevas implementaciones.

Una vez validado el prototipo inicial, se llevó a cabo una segunda etapa, también de tres meses, centrada en el perfeccionamiento del sistema y el desarrollo de la versión final. En esta fase se añadieron mejoras tanto a nivel funcional como visual, incluyendo la optimización de la interfaz de usuario, la ampliación de las opciones de configuración y la integración completa de todas las funcionalidades planificadas para el producto final.

El uso de esta metodología ha permitido mantener un flujo de trabajo flexible y adaptativo, facilitando la incorporación de mejoras continuas y la resolución temprana de incidencias, lo que ha contribuido a obtener un sistema estable y operativo al finalizar cada ciclo de desarrollo.

Estructura del trabajo

Este trabajo se organiza en varios apartados que permiten guiar al lector a través de todas las fases del desarrollo del proyecto, desde su concepción hasta su evaluación final:

**Capítulo 1: Análisis del problema**

En esta sección se contextualiza la problemática que da origen al proyecto, detallando las necesidades detectadas y los objetivos que se persiguen. También se analizan las alternativas existentes y se justifican las tecnologías seleccionadas para abordar la solución.

**Capítulo 2: Planificación y gestión del proyecto**

Aquí se describe cómo se ha organizado el desarrollo del sistema, incluyendo la definición del alcance del trabajo, la distribución y calendarización de tareas, así como los recursos empleados a lo largo del proceso. Se detallan las estrategias seguidas para garantizar una gestión eficiente y controlada del proyecto.

**Capítulo 3: Desarrollo de la solución**

En este apartado se profundiza en la solución implementada, explicando de manera detallada el diseño del sistema, sus principales componentes y la arquitectura adoptada. Se justifica cada decisión técnica tomada y se describe cómo estas decisiones permiten cumplir con los objetivos planteados.

**Capítulo 4: Validación y resultados**

Esta sección recoge el proceso llevado a cabo para verificar el correcto funcionamiento del sistema, así como las pruebas realizadas. Además, se presentan los resultados obtenidos y se analiza hasta qué punto se han alcanzado los objetivos iniciales.

**Capítulo 5: Conclusiones y futuras mejoras**

Finalmente, se realiza una reflexión global sobre el trabajo desarrollado. Se exponen los principales retos y dificultades encontrados durante el proyecto, se valoran los resultados alcanzados y se proponen posibles líneas de mejora y evolución futura del sistema.

**Anexo 1: Registro del desarrollo**

Este anexo recoge de forma cronológica la evolución del proyecto, mostrando cómo ha progresado a lo largo del tiempo y los hitos más relevantes alcanzados.

**Anexo 2: Guía de usuario**

Incluye instrucciones claras y detalladas para facilitar al usuario la utilización de la aplicación, desde su puesta en marcha hasta el manejo de sus principales funcionalidades.

Capítulo 1

# Estudio del problema

En esta sección se realiza el estudio del contexto del problema, su estado en la actualidad y se define sus aspectos clave.

## El contexto del problema

El Trabajo de Fin de Grado tiene como objetivo el desarrollo de GreenhouseIOT, una aplicación web orientada a la gestión y supervisión de datos generados por un módulo hidropónico. La idea principal surge como una oportunidad para consolidar conocimientos en tecnologías actuales del ámbito del desarrollo web, además de obtener experiencia práctica en la creación de soluciones reales con proyección profesional.

El proyecto propone una herramienta capaz de monitorizar y almacenar los datos obtenidos a partir de sensores instalados en sistemas hidropónicos, permitiendo así un control preciso de variables críticas como temperatura, humedad o nivel de nutrientes. Todo ello está pensado para facilitar el seguimiento de estos cultivos en tiempo real desde una plataforma web intuitiva y accesible.

Gracias al uso de tecnologías modernas como React para el frontend, FastAPI para la lógica del backend, MariaDB como sistema de almacenamiento, junto con Kafka para el envío de datos y Docker para la contenerización de servicios, la aplicación busca ofrecer una solución completa para el seguimiento local de instalaciones hidropónicas.

## El estado de la cuestión

El auge de la agricultura inteligente y, en particular, de los sistemas hidropónicos, ha impulsado el desarrollo de soluciones tecnológicas orientadas al monitoreo y gestión de cultivos mediante sensores IoT. A nivel industrial, existen plataformas robustas que ofrecen servicios avanzados de análisis de datos, control remoto y optimización de recursos, aunque muchas de estas soluciones están orientadas a explotaciones a gran escala y suponen altos costes de implementación.

En cambio, GreenhouseIOT surge como una alternativa pensada para instalaciones locales o de pequeña escala, proporcionando una plataforma sencilla, adaptable y sin grandes requerimientos técnicos para el usuario final. Mientras que otras herramientas del mercado priorizan la integración de sistemas complejos y centralizados, este proyecto apuesta por una arquitectura distribuida mediante microservicios, facilitando así su mantenimiento, escalabilidad y despliegue en diferentes entornos.

## La definición del problema

En base al análisis previo, el problema principal que aborda este proyecto es la falta de soluciones accesibles y adaptadas a pequeñas instalaciones hidropónicas que permitan gestionar y visualizar de manera eficiente los datos recogidos por sensores IoT.

Con GreenhouseIOT se pretende cubrir las siguientes necesidades:

* Monitorización continua de parámetros ambientales relevantes (temperatura, humedad, nivel de agua, etc.).
* Almacenamiento ordenado y seguro de los datos generados por los sensores.
* Visualización clara e intuitiva de los datos históricos y en tiempo real desde una interfaz web.
* Comunicación eficiente entre los diferentes módulos del sistema.
* Facilitar la escalabilidad del sistema mediante una arquitectura basada en contenedores (Docker) y microservicios.
* Además, uno de los objetivos clave de la aplicación es ofrecer una experiencia de usuario sencilla, que permita tanto la instalación como la gestión del sistema sin necesidad de conocimientos avanzados en tecnología o programación. A partir de una plataforma web accesible desde cualquier dispositivo, el usuario puede supervisar su módulo hidropónico, realizar consultas de los registros históricos y recibir recomendaciones basadas en los datos almacenados.

Con todo ello, GreenhouseIOT pretende ser una herramienta útil y práctica para impulsar el uso de la agricultura inteligente en proyectos locales y domésticos.

Capítulo 2

# Gestión de proyecto software

## 2.1 Alcance del proyecto

En este apartado se describen las tareas necesarias para llevar a cabo el desarrollo de GreenhouseIOT, así como los recursos requeridos, la estimación de costes y los posibles riesgos asociados. La finalidad del proyecto es crear una aplicación web que permita monitorizar y gestionar los datos de sensores instalados en un módulo hidropónico, facilitando su visualización, análisis y almacenamiento de manera sencilla y eficiente para el usuario final.

La aplicación busca ofrecer una herramienta de apoyo a pequeñas instalaciones hidropónicas, proporcionando información en tiempo real y registros históricos que permitan optimizar el cultivo y anticipar posibles problemas en el entorno controlado.

### 2.1.1 Definición del proyecto

El proyecto consiste en el desarrollo de una plataforma web que permita a los usuarios supervisar y gestionar los datos de su módulo hidropónico. La aplicación incluirá funciones como:

* Visualización en tiempo real de datos ambientales (temperatura, humedad, nivel de nutrientes, etc.).
* Consulta de registros históricos y gráficas comparativas.
* Alertas ante valores críticos o anómalos detectados.
* Gestión del almacenamiento de datos mediante una base de datos relacional.
* Panel de administración básico para supervisar el sistema y verificar el estado de los distintos módulos.

La arquitectura estará dividida en 3 capas: origen de datos, persistencia y explotación, desarrolladas de manera paralela y coordinada, asegurando una integración continua y modular para facilitar su mantenimiento y escalabilidad.

La tecnología elegida para la capa de explotación es React, que permitirá diseñar una interfaz interactiva y dinámica para el usuario, en la capa de persistencia encontramos la base de datos y un contenedor desarrollado con Fastapi que permite interactuar con ella. Por último, en la capa de origen de datos encontramos un servicio de Python que se encarga de preguntar a los módulos hidropónicos los datos de forma periódica y prepara esos datos para ser transportados por Kafka hacia la capa de persistencia.

Además, todo el entorno se desplegará utilizando contenedores Docker, asegurando así la portabilidad y estabilidad de la aplicación, independientemente del entorno de ejecución.

El sistema está diseñado para su funcionamiento a nivel local, pensado para pequeñas instalaciones hidropónicas que requieran una solución asequible, funcional y sencilla de implementar.

### 2.1.2 estimación de tareas y recursos

**Tareas**

El desarrollo del proyecto se organizará en tareas semanales, dentro de una metodología incremental que permitirá ir incorporando nuevas funcionalidades progresivamente. A lo largo del desarrollo se realizarán reuniones periódicas para revisar los avances, valorar posibles mejoras y ajustar las prioridades.

Entre las tareas principales previstas se encuentran:

* Diseño de la arquitectura de la aplicación y configuración inicial del entorno.
* Desarrollo del backend con FastAPI.
* Desarrollo del frontend con React.
* Integración de la base de datos MariaDB.
* Configuración y manejo de envió de datos mediante Kafka.
* Creación de pruebas funcionales para asegurar el correcto comportamiento de cada módulo.
* Despliegue de los servicios con Docker.
* Elaboración de documentación técnica y manual de usuario.
* Pruebas finales y validación del sistema.

Cada entrega de una nueva funcionalidad tendrá una duración estimada mínima de dos semanas, dependiendo de la complejidad y los recursos disponibles.

**Recursos**

Para el desarrollo de GreenhouseIOT se contemplan tanto recursos humanos como físicos:

Humanos

El equipo ideal del proyecto estaría compuesto por cinco perfiles clave, cada uno enfocado en áreas específicas del desarrollo:

Jefe de proyecto: encargado de planificar, supervisar y coordinar todas las fases del desarrollo.

Backend developer: encargado de la implementación de la lógica de negocio, API y conexiones con la base de datos. Se encargaría del mantenimiento y configuración de la base de datos y de las tareas relacionadas con Docker.

Frontend developer: responsable del desarrollo de la parte visual e interactiva de la plataforma. Se encargaría de conseguir una interfaz amigable para el usuario haciendo uso de librerías de componentes UI.

**Físicos**

Los recursos físicos necesarios para llevar a cabo el proyecto son:

* Hardware: equipos informáticos para cada miembro del equipo, capaces de ejecutar entornos de desarrollo y contenedores de Docker de manera eficiente.
* Software: no se prevén costes adicionales, ya que todas las tecnologías utilizadas son open source (React, FastAPI, MariaDB, Kafka y Docker).

### 2.1.3 Presupuesto

El presupuesto para el desarrollo de GreenhouseIoT se ha estimado considerando los costes asociados tanto a los recursos humanos como a los recursos físicos e infraestructurales necesarios para garantizar el correcto desarrollo del proyecto.

**Recursos Humanos**

El equipo de trabajo estará formado por los siguientes perfiles profesionales, junto con su coste mensual aproximado y la duración estimada del proyecto (3 meses):

|  |  |  |  |
| --- | --- | --- | --- |
| Puesto | Coste mensual (€) | Duración (meses) | Coste total (€) |
| Jefe de proyecto | 2.000 | 3 | 6.000 |
| Backend developer | 1.700 | 3 | 5.100 |
| Frontend developer | 1.700 | 3 | 5.100 |

Total, RRHH 16.200

Nota: Se ha estimado un equipo reducido enfocado en cubrir las áreas críticas del proyecto, prescindiendo de perfiles secundarios para optimizar los costes. Otra opción más económica sería contratar a un desarrollador fullstack y alargar ligeramente la duración del desarrollo.

**Recursos Físicos**

Hardware: Se estima una amortización de 500 € por equipo de desarrollo (portátiles o sobremesa de altas prestaciones), para un total de 3 equipos:

500 € x 3 = 1.500 €

Software: No se contemplan costes adicionales debido al uso exclusivo de herramientas y tecnologías open source como FastAPI, React, MariaDB, Kafka y Docker.

**Costes Indirectos**

Oficina: Se optará por el trabajo remoto por lo que no se aplican costes debidos a oficinas físicas.

Para el desarrollo y testeo de la aplicación será necesaria la construcción del por lo menos 1 módulo hidropónico inteligente. Los costes asociados a su construcción se estiman en 100€ x módulo.

En este caso se considerará que se construye solo 1 módulo ya que tras el desarrollo el cliente será responsable y propietario de la compra y mantenimiento del módulo físico.

**Presupuesto Total**

|  |  |
| --- | --- |
| Concepto | Coste (€) |
| Recursos Humanos | 16.200 |
| Hardware | 1.500 |
| Otros costes | 100€ |
| **Total** | **17.800€** |

Este presupuesto está sujeto a posibles ajustes según la evolución del proyecto, cambios en los plazos o incorporación de nuevos requerimientos.

## 2.2 Plan de trabajo

En esta sección se detalla cómo se han establecido las diferentes tareas del desarrollo de la aplicación GreenhouseIoT, así como el tiempo aproximado de dedicación a cada una. El plan de trabajo sigue una metodología incremental, permitiendo validar cada funcionalidad antes de avanzar a la siguiente, asegurando un desarrollo controlado y eficiente.

### 2.2.1 Identificación de tareas

**Tarea 1**. Investigación sobre el sector y tecnologías.

El primer paso fue realizar un estudio del ámbito del IoT aplicado a invernaderos, identificando necesidades comunes en la monitorización y gestión de cultivos. Se analizaron soluciones existentes y se definieron los requisitos clave para nuestra plataforma. A partir de ahí, se seleccionaron las tecnologías más adecuadas (FastAPI, React, MariaDB, Kafka y Docker).

**Tarea 2.** Recopilación de datos de ejemplo.

Para probar el funcionamiento de la aplicación, se generaron datos de ejemplo que simulan los datos generados por los sensores finales del prototipo.

**Tarea 3.** Selección del nombre y creación del logotipo.

Se definió la identidad visual del proyecto, seleccionando el nombre GreenhouseIoT y diseñando un logotipo que refleje la modernización de la agricultura mediante tecnologías digitales.

**Tarea 4.** Diseño de la interfaz de usuario y estructura.

Se diseñó la estructura de la aplicación web, priorizando una navegación clara y simple. Se definieron paletas de colores y componentes visuales para facilitar una experiencia intuitiva, especialmente orientada a usuarios sin conocimientos técnicos.

**Tarea 5.** Desarrollo de la aplicación.

Diseño de la base de datos y creación del repositorio.

Se definió el esquema de la base de datos MariaDB, estableciendo tablas y relaciones necesarias para gestionar los datos de sensores y usuarios. Además, se creó un repositorio en GitHub para gestionar el control de versiones.

**Tarea 6.** Configuración inicial de los servicios.

Se inicializaron los proyectos de backend (FastAPI) y frontend (React), verificando la correcta comunicación entre ambos mediante peticiones básicas.

**Tarea 7**. Panel de control de invernaderos.

Se creó una interfaz para que los usuarios puedan visualizar los invernaderos registrados, consultar sus datos en tiempo real y modificar configuraciones básicas.

**Tarea 8.** Visualización de datos históricos y gráficos.

Se añadieron gráficos interactivos para consultar la evolución de los parámetros ambientales registrados por los sensores.

**Tarea 9.** Gestión de alertas y notificaciones.

Se desarrolló un sistema de avisos que alerta al usuario si algún parámetro supera los valores críticos establecidos.

**Tarea 10.** Integración de Kafka para la transmisión de datos.

Se implementó Kafka como middleware para gestionar el flujo de datos desde los dispositivos IoT al backend, asegurando escalabilidad y eficiencia.

**Tarea 11.** Configuración de Docker y despliegue de servicios.

Se contenedorizó toda la aplicación usando Docker, facilitando su despliegue en cualquier entorno y la replicación del sistema.

**Tarea 12.** Documentación del proyecto.

Durante el desarrollo se ha ido generando documentación técnica que describe la arquitectura, funcionamiento y despliegue del sistema, además del manual de usuario.

**Tarea 13.** Corrección de errores.

A lo largo del desarrollo se han resuelto incidencias surgidas durante la integración de nuevas funcionalidades y tras las pruebas realizadas.

**Tarea 14.** Pruebas unitarias y validación final.

Se llevaron a cabo pruebas unitarias y de integración para asegurar que todos los módulos funcionaran correctamente de manera independiente y conjunta, validando el producto final antes de su entrega.

### 2.2.2. Asignación de tareas y estimación

### 2.2.3. Planificación de tareas2.3. Gestión de recursos

### 2.3.1. Especificación de recursos

### 2.3.2. Asignación de recursos

## 2.4. Gestión de riesgos

2.4.1. Identificación de riesgos2.5. Legislación y normativa

Capítulo 3

# Solución

## 3.1. Descripción de la solución3.2. El proceso de desarrollo

### 3.2.1. Análisis3.2.2. Diseño

## 3.3. El producto del desarrollo

Capítulo 4

# Evaluación

## 4.1. Proceso de evaluación

## 4.1.1. Forma de evaluación

## 4.1.2. Casos de prueba4.2. Análisis de resultadosConclusión

# Lista de referencias

# Anexo 1 - Seguimiento de Proyecto

# Anexo 2 - Manual de usuario