

GRADO EN INGENIERÍA INFORMÁTICA
CURSO 2023-24

**INTERFAZ INMERSIVA DE MONITORIZACIÓN DE
SENSORES REMOTOS**

AUTOR

CARLOS SAIZ BLANCO

TUTOR

ALEJANDRO REY LÓPEZ

LEGANÉS, SEPTIEMBRE DE 2024



Esta obra se encuentra sujeta a la licencia Creative Commons
Reconocimiento – No Comercial – Sin Obra Derivada

Resumen

En este proyecto se ha desarrollado un dispositivo capaz de recopilar datos de sensores del mundo real para transmitirlos al usuario final a través de realidad aumentada. En concreto, este sistema se ha orientado hacia la creación de un sistema de monitorización de plantas en realidad aumentada, que almacena telemetrías actuales y anteriores para tener un mayor control de una única planta o de varias al mismo tiempo.

En primer lugar, para la construcción del dispositivo capaz de recopilar la información, ha sido necesario realizar las conexiones y escribir el código que hace funcionar todos los sensores en una placa Arduino.

Posteriormente, ha sido necesaria la creación de un servidor capaz de recibir los datos recopilados, además de canales de comunicación seguros y confiables entre todos los elementos del sistema.

Por último, con ayuda del motor Unity3D y la librería Vuforia, se ha creado una aplicación compatible con dispositivos móviles que muestra la información en realidad aumentada, permitiendo al usuario final ver las condiciones ambientales de una o varias plantas en tiempo real de forma fácil e intuitiva.

Palabras clave: Sensores, Realidad aumentada, Telemetrías, Arduino, Unity3D, Vuforia.

Abstract

This project involved the creation of a device capable of collecting data from real-world sensors and transmitting it to users through augmented reality. Specifically, this system was developed for monitoring plants in augmented reality. This system stores both current and historical telemetry, which helps in having a better control of one or multiple plants at the same time.

Firstly, for the construction of the device capable of collecting information, it was necessary to build the connections and write the code necessary to make all the sensors work on an Arduino board.

It was also necessary to create a server capable of receiving all the data from the sensors and the construction of secure and reliable communication channels between all the elements of the system.

Finally, with tools like Unity3D and the Vuforia library, an application compatible with mobile devices was developed. This application shows the live information in augmented reality, allowing the user to see the environmental conditions of one or multiple plants in an easy and intuitive way.

Palabras clave: Sensors, Augmented reality, Telemetry, Arduino, Unity3D, Vuforia.

Índice

	1
RESUMEN	3
ABSTRACT	5
ÍNDICE	7
ÍNDICE DE FIGURAS	9
ÍNDICE DE TABLAS	10
1. INTRODUCCIÓN	12
1.1. MOTIVACIÓN	12
1.2. OBJETIVOS	14
1.3. ESTRUCTURA DEL DOCUMENTO	15
2. ESTADO DEL ARTE	16
2.1. PROYECTOS SIMILARES	16
2.2. TECNOLOGÍAS EXISTENTES	20
2.2.1. <i>Visualización de entorno virtual</i>	20
2.2.2. <i>Tecnologías de entornos 3D</i>	23
2.2.3. <i>Microcontroladores</i>	24
2.2.4. <i>Tecnologías de comunicación</i>	25
2.3. TECNOLOGÍAS UTILIZADAS	26
2.3.1. <i>Comunicaciones</i>	26
2.3.2. <i>Microcontroladores</i>	27
2.3.3. <i>Motores 3D</i>	28
2.3.4. <i>Tecnologías AR</i>	28
3. ANÁLISIS	29
3.1. DESCRIPCIÓN DEL PROYECTO	29
3.2. CASOS DE USO	29
3.3. REQUISITOS	33
3.3.1. <i>Usuario</i>	33
3.3.2. <i>Sistema</i>	35
3.4. MATRIZ DE TRAZABILIDAD	39
3.5. METODOLOGÍA USADA PARA EL DESARROLLO DEL PROYECTO	40
4. MARCO REGULADOR	41
4.1. MARCO LEGAL	41
4.2. LICENCIAS	42
5. ENTORNO SOCIOECONÓMICO	43
5.1. PLANIFICACIÓN	43
5.1.1. <i>Diagrama de Gantt</i>	43
5.1.2. <i>Diagrama de PERT</i>	45
5.2. PRESUPUESTO PARA LA REALIZACIÓN DEL DISPOSITIVO	45
5.3. PRESUPUESTO PARA EL DESARROLLO DEL TRABAJO	46
5.4. IMPACTO SOCIOECONÓMICO	49

6.	DISEÑO	50
6.1.	HARDWARE	50
6.1.1.	<i>Esquema de conexiones del microcontrolador</i>	50
6.1.2.	<i>Esquema de comunicaciones</i>	52
6.2.	SOFTWARE	53
6.2.1.	<i>Conexión con el servidor</i>	53
6.2.2.	<i>Servidor</i>	54
6.2.3.	<i>Interfaz del servidor</i>	55
6.2.4.	<i>Aplicación AR</i>	57
7.	PRUEBAS	60
7.1.	VALIDACIÓN DE REQUISITOS DE USUARIO	60
7.2.	VALIDACIÓN DE REQUISITOS DEL SISTEMA	62
8.	CONCLUSIONES	65
8.1.	RETROSPECTIVA DE OBJETIVOS	65
8.2.	FUTUROS AÑADIDOS	66
8.3.	CONCLUSIONES PERSONALES	67
9.	BIBLIOGRAFÍA	68
10.	GLOSARIO DE TÉRMINOS	72

Índice de figuras

FIGURA 1. GRÁFICO SOBRE EL VALOR DE MERCADO DE VR Y AR HASTA 2032 SEGÚN LAS PREVISIONES [4].	13
ILUSTRACIÓN 2. INTERFAZ DE "MAGIC FLOWERPOT" [5].	16
ILUSTRACIÓN 3. VERSIÓN FINAL OFFL-LINE DE LA APLICACIÓN PARA VISUALIZAR INFORMACIÓN OBTENIDA CON RPAS [6].	17
ILUSTRACIÓN 4. IMAGEN DEL DISPOSITIVO GARDUINO [7].	18
ILUSTRACIÓN 5. ESQUEMA DE CONEXIONES DE ARDUINO PLANT MONITOR [8].	19
ILUSTRACIÓN 6. USUARIO UTILIZANDO UNAS GAFAS DE VR [11].	20
ILUSTRACIÓN 7. EJEMPLO DE LAS <i>HOLOLENS 2</i> , GAFAS AR DE <i>MICROSOFT</i> [14].	20
ILUSTRACIÓN 8. IMAGEN DE <i>POKEMON GO</i> , EJEMPLO DE APLICACIÓN AR [17].	22
ILUSTRACIÓN 9. ENTORNO DE DESARROLLO DEL MOTOR UNITY3D [19].	23
ILUSTRACIÓN 10. EJEMPLO DE UN MICROCONTROLADOR [20].	24
ILUSTRACIÓN 11. PLACA NODEMCU ESP8266 V2 [25].	27
ILUSTRACIÓN 12. EJEMPLO DE RECONOCIMIENTO DE IMÁGENES CON <i>VUFORIA</i> [27].	28
ILUSTRACIÓN 13. DIAGRAMA DE CASOS DE USO.	30
ILUSTRACIÓN 14. EJEMPLO DE KANBAN [28].	40
ILUSTRACIÓN 15. DIAGRAMA DE GANTT PREVIO AL COMIENZO DEL PROYECTO.	43
ILUSTRACIÓN 16. DIAGRAMA DE GANTT FINAL.	44
ILUSTRACIÓN 17. DIAGRAMA DE PERT.	45
ILUSTRACIÓN 18. ESQUEMA DE CONEXIONES DEL MICROCONTROLADOR.	50
ILUSTRACIÓN 19. FOTOGRAFÍA DE LA CARCASA DEL MICROCONTROLADOR (SE APRECIA EL CÓDIGO QR, EL LED EN LA PARTE SUPERIOR Y EL SENSOR DE AGUA CLAVADO EN LA TIERRA).	51
ILUSTRACIÓN 20. ESQUEMA DE COMUNICACIONES.	52
ILUSTRACIÓN 21. EJEMPLO DE DICCIONARIOS DEL SERVIDOR.	54
ILUSTRACIÓN 22. INTERFAZ DEL SERVIDOR.	55
ILUSTRACIÓN 23. VISTA DEL DESPLEGABLE DE TELEMETRÍAS.	56
ILUSTRACIÓN 24. VENTANA EMERGENTE PARA EL BORRADO DE TELEMETRÍAS.	56
ILUSTRACIÓN 25. ESCENA DEL PROYECTO EN UNITY3D.	57
ILUSTRACIÓN 26. CAPTURA DE LA APLICACIÓN AR MIENTRAS SE BUSCA EL SERVIDOR.	58
ILUSTRACIÓN 27. CAPTURA DE LA APLICACIÓN AR MOSTRANDO DATOS.	59

Índice de tablas

TABLA 1. COMPARATIVA VR /AR [15].	21
TABLA 2. EJEMPLO DE FORMATO DE TABLAS DE CASOS DE USO.	30
TABLA 3. CU-U-01	31
TABLA 4. CU-U-02	31
TABLA 5. CU-A-01.	31
TABLA 6. CU-A-02.	31
TABLA 7. CU-A-03.	32
TABLA 8. CU-A-04.	32
TABLA 9. CU-A-05.	32
TABLA 10. CU-A-06.	32
TABLA 11. EJEMPLO DE FORMATO DE TABLA DE REQUISITOS.	33
TABLA 12. RU01.	33
TABLA 13. RU02.	33
TABLA 14. RU03.	34
TABLA 15. RU04.	34
TABLA 16. RU05.	34
TABLA 17. RU06.	34
TABLA 18. RU07.	34
TABLA 19. RU08.	35
TABLA 20. RU09.	35
TABLA 21. RS01.	35
TABLA 22. RS02.	35
TABLA 23. RS03.	36
TABLA 24. RS04.	36
TABLA 25. RS05.	36
TABLA 26. RS06.	36
TABLA 27. RS07.	36
TABLA 28. RS08.	37
TABLA 29. RS9.	37
TABLA 30. RS10.	37
TABLA 31. RS11.	37
TABLA 32. RS12.	37
TABLA 33. RS13.	38
TABLA 34. MATRIZ DE TRAZABILIDAD DE REQUISITOS.	39
TABLA 35. PRECIOS DE HARDWARE.	45
TABLA 36. DESGLOSE DE HORAS POR PUESTO.	46
TABLA 37. DESGLOSE DE COSTE DE PERSONAL.	46
TABLA 38. COSTES DE SOFTWARE	47
TABLA 39. COSTES DE HARDWARE.	47
TABLA 40. COSTES DE MATERIAL FUNGIBLE.	47
TABLA 41. COSTES INDIRECTOS.	48
TABLA 42. COSTES TOTALES.	48
TABLA 43. PLANTILLA PARA CASOS DE PRUEBA.	60
TABLA 44. PU-01.	60
TABLA 45. PU-02.	60
TABLA 46. PU-03.	61
TABLA 47. PU-04.	61
TABLA 48. PU-05.	61
TABLA 49. PU-06.	62
TABLA 50. PS-01.	62
TABLA 51. PS-02.	62

TABLA 52. PS-03.....	63
TABLA 53. PS-04.....	63
TABLA 54. PS-05.....	63
TABLA 55. PS-06.....	63
TABLA 56. PS-07.....	64

1. Introducción

En este apartado se introducirá el trabajo, se hablará de los pasos llevados a cabo para la elección de la temática, así como los objetivos que se pretenden alcanzar durante el desarrollo de este.

1.1. Motivación

El mundo actual vive un nuevo proceso de revolución industrial con la llamada “Industria 4.0”.

Esta nueva industria combina técnicas avanzadas de producción con el uso de tecnologías inteligentes y supondrá un nuevo cambio de paradigma que tendrá impacto en ecosistemas, organizaciones e individuos [1].

Dentro de esta nueva industria que se está abriendo camino desde hace años, encontramos dos tecnologías como son la realidad aumentada (AR) y la realidad virtual (VR). El uso de estas tecnologías permite ahorrar costes y tiempo al facilitar el acceso a la información, además de facilitar y simplificar procesos.

Todo esto se debe a que ambas tecnologías ofrecen una experiencia inmersiva que permite interactuar en tiempo real con el entorno físico y virtual de forma simultánea.

Estos avances han llegado a campos como el entretenimiento, la educación, la medicina o la arquitectura, entre otras, donde ya se aplica AR para el aprendizaje, planificación de cirugías o visualización tridimensional de edificios [2].

En un afán de integrar los beneficios que ofrece la realidad aumentada a un entorno cotidiano-profesional, surge la idea de crear un entorno en el que poder trasladar datos de sensores embebidos en el mundo real que capturan información y la trasladan a la realidad aumentada, proporcionando información contextual de los elementos monitorizados. El sistema ideado para este trabajo consiste en implementar un sistema que permita monitorizar una o múltiples plantas y ver su estado en tiempo real por medio de realidad aumentada.

Este tipo de sistemas puede ayudar también a nivel industrial en huertos o viveros, ya que se ha demostrado a lo largo de los últimos años que la introducción de la realidad aumentada en la industria favorece la productividad, llegando a aumentar en un 32% en algunos sectores [3].

Además de todas estas ventajas para la industria, el mercado de AR y VR es un mercado creciente a un ritmo exponencial, pudiendo generar 372 billones de dólares estadounidenses para el año 2032 según las estimaciones [4].

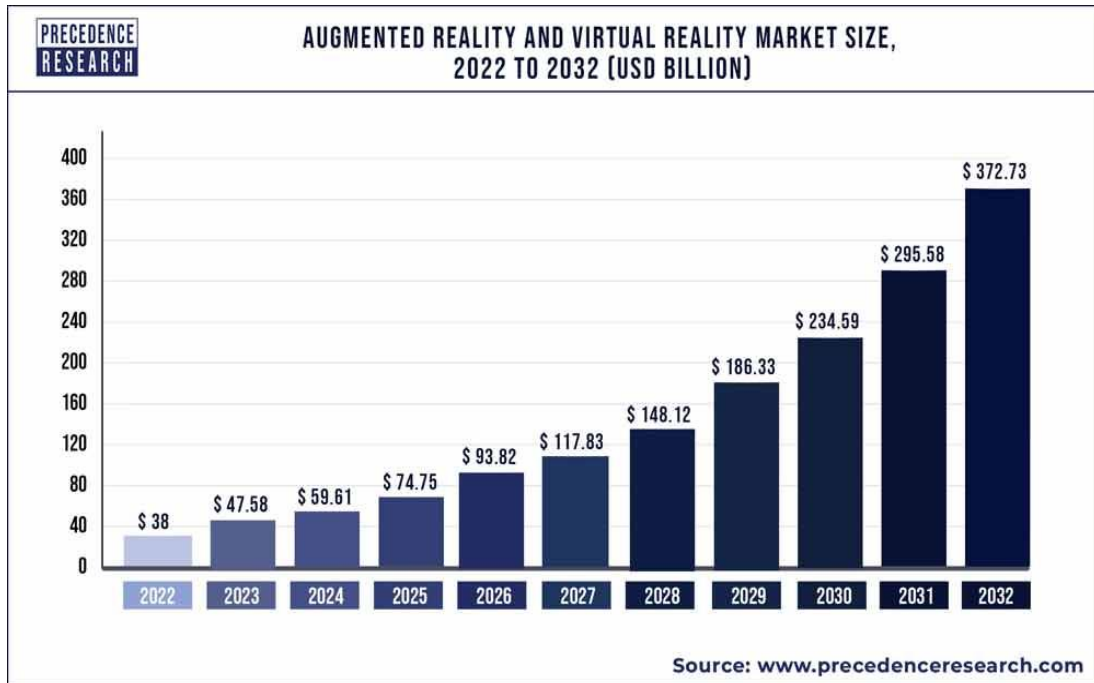


Figura 1. Gráfico sobre el valor de mercado de VR y AR hasta 2032 según las previsiones [4].

El hecho de que la Realidad Aumentada sea una tecnología en auge, con un gran mercado y capaz de aportar valor en la industria, sumado a que no es un tema que se desarrolle profundamente en los contenidos del grado de Ingeniería Informática, han hecho que, aplicar AR a un sistema de monitorización de sensores, sea un tema interesante para abordar en este trabajo de fin de grado.

La elección de este tema permite ampliar ciertos conocimientos que sí han sido estudiados, como la comunicación por *sockets* y *websockets*, con otros contenidos que no han sido tratados, como la propia AR o el uso de motores 3D como *Unity*.

1.2. Objetivos

El objetivo principal del trabajo es poder llevar datos capturados en el mundo real a un entorno virtual de AR que se pueda visualizar mediante el uso de gafas VR/AR o mediante el uso de un dispositivo móvil con cámara.

En concreto, el objetivo es poder **monitorizar una o múltiples plantas, recopilando datos sobre temperatura, humedad o nivel de agua de la tierra, para facilitar el cuidado de jardines, huertos o viveros**. Este sistema, permite ahorrar tiempo, al poder ver el estado de las plantas y tomar decisiones, simplemente mirando directamente a estas.

Para ello, los objetivos que se deben cumplir son los siguientes:

- Decidir qué datos capturar del mundo real, cuáles son los instrumentos adecuados para capturarlos e implementar toda la lógica capaz de realizar esas lecturas de datos.
- Crear un servidor capaz de tratar, almacenar y proporcionar dichos datos en tiempo real.
- Crear canales de conexión seguros entre los dispositivos que capturan los datos, el servidor y el usuario final que los recibe.
- Diseñar un entorno de AR capaz de mostrar los datos y de funcionar con distintos dispositivos como móviles, tablets o gafas de VR/AR

1.3. Estructura del documento

El documento consta de los siguientes apartados:

- **Introducción:** apartado en el que se hace una breve exposición del problema que se busca resolver, la motivación detrás del trabajo y los objetivos que se espera cumplir.
- **Estado del arte:** apartado en el que se expone toda la investigación técnica realizada a lo largo del proyecto y en la que se justifican algunas decisiones tomadas durante el desarrollo del trabajo.
- **Análisis:** apartado en el que se analiza el problema más a fondo, se introducen requisitos y casos de uso, además de la metodología utilizada para el desarrollo del proyecto
- **Marco regulador:** apartado en el que se exponen las leyes y normativas que pueden afectar a este proyecto, además de la disponibilidad de las licencias utilizadas para el desarrollo de este.
- **Entorno socioeconómico:** apartado en el que se realiza un análisis económico del proyecto, detallando los presupuestos y analizando las aportaciones de este trabajo a los usuarios.
- **Diseño:** apartado en el que se detallan los aspectos relativos al desarrollo del trabajo y la construcción del sistema.
- **Pruebas:** apartado en el que se detallan las pruebas realizadas para verificar el cumplimiento de los requisitos definidos.
- **Conclusiones:** apartado en el que se realiza una reflexión final sobre el cumplimiento de objetivos y los añadidos que se podrían haber hecho o se pueden hacer en un futuro al sistema.

2. Estado del arte

En este apartado se explicarán una serie de tecnologías existentes que son útiles para la realización de este proyecto, la justificación de cuáles se han utilizado o descartado y un análisis de productos similares en el mercado.

El objetivo de este apartado es el de justificar algunas decisiones de diseño tomadas durante el desarrollo del trabajo para poder entender mejor el resultado final del mismo.

2.1. Proyectos similares

Para este proyecto es necesario estudiar qué trabajos similares existen y qué soluciones implementaron.

Magic Flowerpot

El proyecto “***Magic Flowerpot***”, es un trabajo realizado en la Universidad Carlos III de Madrid en el que se integran dos de los conceptos principales de este proyecto, la realidad aumentada y la monitorización de plantas con sensores.

En este trabajo se recopila información de una planta real para trasladarla a una plataforma de reconocimiento de plantas. Una vez reconocida, se puede trasladar a un jardín AR en el que se crea una copia virtual de esa planta con información relevante sobre ella.

Esa misma planta puede ser monitorizada en AR con las gafas ***HoloLens 2***, a través de los sensores conectados a un microcontrolador, para observar una simulación de su evolución y crecimiento acorde a las condiciones climáticas externas.

En este caso, el objetivo es crear un juego educativo que permita aprender sobre el crecimiento de las plantas, pero muchos de los conceptos utilizados son aplicables para este proyecto [5].



Ilustración 2. Interfaz de “Magic Flowerpot” [5].

Uso de realidad aumentada para la monitorización de información obtenida con RPAS

Este trabajo ha sido desarrollado por dos estudiantes de Ingeniería de Sistemas Aeroespaciales en la Universidad Politécnica de Cataluña. El tema principal de este trabajo tiene dos aspectos relevantes, como son el uso de sensores para recopilar información ambiental en tiempo real y el uso de la Realidad Aumentada para visualizarlos.

Se utilizan los datos recopilados por los sensores de un **RPAS** (*Remotely Piloted Aircraft System*). En concreto, se utiliza un dron para recopilar datos sobre la radiación ambiental, guardando la ubicación de las mediciones y generando un objeto 3D para marcar dichas posiciones. Posteriormente el usuario puede acercarse a cada uno de los puntos generados y realizar la lectura de los datos.

Este tema en un principio puede parecer muy distinto al proyecto de monitorización de plantas mediante sensores, pero tiene muchas similitudes además del uso de sensores y AR. Se utiliza **Unity3D** con la librería **Vuforia** para generar los objetos virtuales en el entorno, se utiliza un dispositivo móvil para visualizar la Realidad Aumentada y se utiliza una arquitectura cliente-servidor para la transmisión de datos [6].

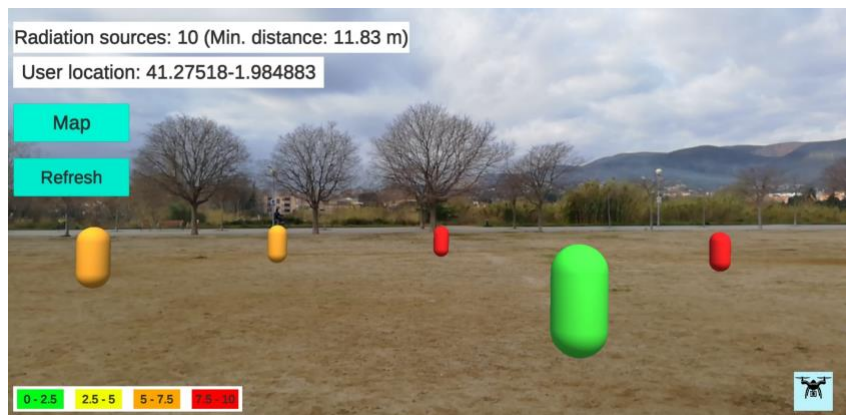


Ilustración 3. Versión final OFFL-LINE de la aplicación para visualizar información obtenida con RPAS [6].

Garduino

Este proyecto ha sido desarrollado por dos estudiantes del Interdisciplinary Center (IDC). Estos dos estudiantes han desarrollado un tutorial que permite construir un dispositivo para monitorizar plantas basado en Arduino. Este dispositivo es capaz de obtener datos de sensores y enviar notificaciones de estado a dispositivos móviles.

En la página web del proyecto facilitan la lista de componentes necesarios para replicar el proyecto, así como todo el código necesario para que cualquiera pueda construir el dispositivo.

Este dispositivo tiene muchas similitudes con el proyecto tratado en este TFG, ya que la finalidad es la de poder monitorizar plantas de forma remota. La principal diferencia es que este proyecto no integra la realidad aumentada como forma de mostrar la información, optando por una solución más simple como el envío de notificaciones en dispositivos móviles [7].



Ilustración 4. Imagen del dispositivo Garduino [7].

Arduino Plant Monitor

Este proyecto desarrollado por un usuario en un foro online muestra cómo construir un sistema capaz de monitorizar plantas y tomar acciones sobre ellas de forma automática. Este dispositivo es capaz de adoptar las condiciones de humedad, temperatura y luz necesarias para una planta de manera totalmente automática a partir de los datos recopilados por los sensores.

Este dispositivo se basa en una placa Arduino a la que se conectan sensores y relés para integrar todas las funcionalidades. Puede regar una planta, iluminarla o modificar su temperatura cuando cualquiera de las condiciones ambientales esté fuera de los límites establecidos por el usuario.

El creador de este sistema ofrece un tutorial detallado para que cualquier usuario pueda replicarlo, incluyendo también el software utilizado.

Este proyecto es muy interesante porque, pese a no utilizar realidad aumentada, ofrece un añadido muy útil para el concepto tratado en este TFG, ya que no solo permite monitorizar plantas. Este dispositivo ofrece un sistema que permite modificar las condiciones ambientales de forma automática sin requerir la interacción humana [8].

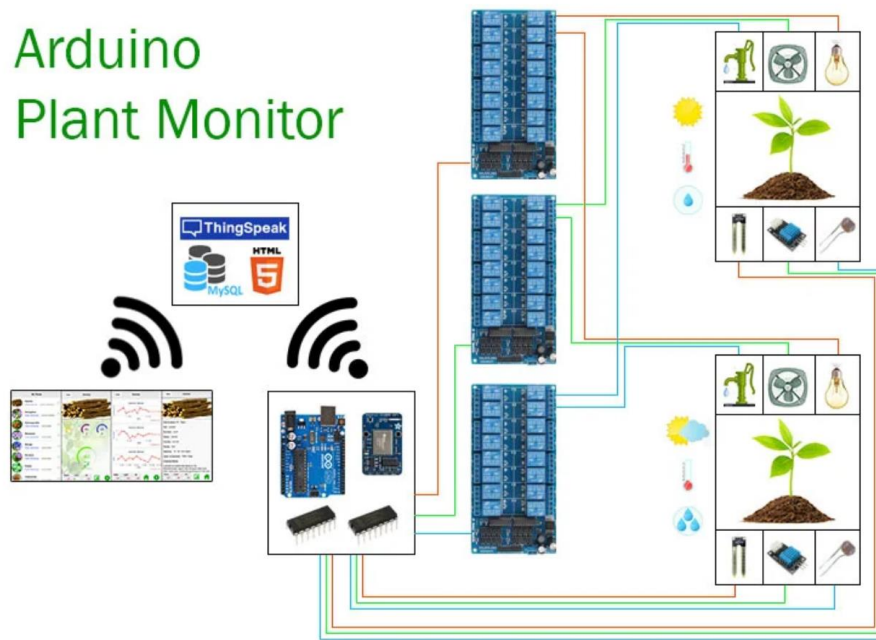


Ilustración 5. Esquema de conexiones de Arduino Plant Monitor [8].

2.2. Tecnologías existentes

2.2.1. Visualización de entorno virtual

Tecnologías inmersivas:

Cuando se habla de tecnologías inmersivas, suele estar referido a VR y AR.

La Realidad Virtual (VR) permite al usuario adentrarse dentro de un mundo inmersivo generado por ordenador, proporcionándole la sensación de que está en un entorno real. [9]. Para ello se utilizan las llamadas “Gafas de realidad virtual” que son capaces de transmitir imágenes distintas a cada ojo y crear esa sensación inmersiva. [10].



Ilustración 6. Usuario utilizando unas gafas de VR [11].

Por otro lado, mientras con VR el usuario se sumerge en un mundo simulado, la Realidad Aumentada (AR) integra elementos simulados al mundo real, permitiendo una interacción sin desconectar al usuario de su entorno [12]. Además, para el uso de esta tecnología, no es necesario el uso de las gafas de realidad virtual, se puede integrar también en dispositivos como teléfonos móviles u ordenadores, entre otros dispositivos con cámara. De esta forma, se consiguen eliminar barreras de entrada y facilitar el aprendizaje al utilizar dispositivos utilizados en la vida cotidiana [13].



Ilustración 7. Ejemplo de las *HoloLens 2*, gafas AR de *Microsoft* [14].

Comparativa VR / AR:

	VR	AR
Coste	Mayor coste de entrada al necesitar equipamiento específico.	Asequible para todos los públicos al poderse utilizar con dispositivos cotidianos.
Aplicaciones	Videojuegos, simulación de entornos o entretenimiento general.	Videojuegos, Arquitectura, Medicina o Educación, entre otras.
Dispositivos	Gafas de VR y dispositivos de rastreo.	Dispositivos móviles, dispositivos con cámara o gafas de VR.
Accesibilidad	Mayor barrera de entrada, ya que se utilizan dispositivos específicos para este fin.	Más accesible al utilizar dispositivos usados en rutinas del día a día (teléfonos móviles)

Tabla 1. Comparativa VR /AR [15].

Realidad Aumentada:

Las tecnologías AR tienen algunas aplicaciones comunes con las tecnologías VR, pero también tienen grandes ventajas cruciales que la hacen idónea para el desarrollo de este proyecto, que pretende integrar elementos virtuales al mundo real, no crear un mundo virtual.

Para ello, existen varios tipos de AR disponibles:

- **AR basada en localización:** esta tecnología utiliza los datos de sensores GPS, brújula y acelerómetro para superponer objetos 3D virtuales en espacios reales. Se utiliza para diversos fines, entre ellos el más destacado es en el videojuego móvil "*Pokémon GO*", que fue uno de los pioneros.
- **AR basada en proyección:** a diferencia de la anterior, esta tecnología no permite el movimiento. Se utiliza un proyector fijo y una cámara de seguimiento, por lo que sólo es aplicable a una ubicación. Con el uso de AR basada en proyección, se pueden crear representaciones realistas de un objeto 3D en superficies planas. Se utiliza principalmente en la industria manufacturera, siendo útil para guiar a los trabajadores y dar instrucciones que pueden variar con el tiempo.

- **AR basada en la superposición:** utiliza objetos que existen físicamente en el mundo real para superponer una imagen u objeto. Se puede utilizar para mostrar una vista más detallada sobre dicho objeto o información adicional.
- **AR basada en el contorno:** utiliza cámaras especiales para delinear objetos con líneas de contorno, que se usan como guía para posicionar objetos con mayor precisión. Se utiliza, entre otros, en sistemas de navegación de coches para mejorar la seguridad en condiciones de baja visibilidad.
- **AR basada en marcadores:** a diferencia de las cuatro tecnologías anteriores, esta tecnología utiliza marcadores para lograr posicionar imágenes u objetos. Para ello, se busca en el entorno si existe algún objeto, llamado marcador, que haya sido preprogramado y pueda ser reconocible. El ejemplo más claro de esta tecnología es el uso de códigos QR como marcadores.

Todas estas tecnologías permiten introducir imágenes, sonidos u objetos 3D al mundo real de diversas formas y son aplicadas en muchas aplicaciones utilizadas diariamente en todo el mundo [16]



Ilustración 8. Imagen de *Pokemon GO*, ejemplo de aplicación AR [17].

2.2.2. Tecnologías de entornos 3D

Los entornos 3D se suelen construir en motores de videojuegos, que son soluciones de software que aportan un entorno sobre el que trabajar, ofreciendo la capacidad de renderizar objetos tanto 3D como 2D, modificar la iluminación, las texturas o el comportamiento de objetos mediante código, gestionar las físicas o el sonido, entre otras muchas funciones.

Existen múltiples motores 3D, pero los dos más importantes en la actualidad, disponibles para uso libre, son **Unreal Engine** y **Unity3D**. Estos dos motores aportan versatilidad al poder crear entornos y proyectos para múltiples plataformas y dispositivos, además de permitir el desarrollo en **C++** y **C#** respectivamente, dos de los lenguajes de programación más populares [18].

Estos motores 3D tienen múltiples librerías de código que facilitan la integración de VR y AR con mundos virtuales. Además, disponen de una gran cantidad de documentación y foros de discusión que ayudan al aprendizaje y desarrollo de aplicaciones en estos programas.



Ilustración 9. Entorno de desarrollo del motor Unity3D [19].

2.2.3. Microcontroladores

Un microcontrolador es un dispositivo electrónico programable, que puede realizar una serie de instrucciones. Su comportamiento es muy similar al de un ordenador, pero con un tamaño mucho menor, de coste más reducido y con propósitos menos generales.

Al igual que un ordenador, tienen una CPU, memoria RAM y ROM, puertos de entrada y salida de datos, etc [20].

Existen múltiples alternativas útiles para este proyecto, desde placas basadas en **Arduino** hasta algunos miniordenadores como la **RaspberryPi**, que, pese a ser un miniordenador y no un microcontrolador, podría cumplir también con el mismo propósito [21]

Estos dispositivos se pueden utilizar para poder recopilar y transmitir información mediante sensores o conectar módulos funcionales.

Para recopilar información de los sensores y alimentarlos con energía disponen de puertos de Entrada/Salida de Propósito General (GPIO). Estos puertos tienen modos de alto o bajo voltaje y, como su nombre indica, pueden configurarse como puertos de entrada o de salida. Además, permiten conectar múltiples tipos de sensores, como el "**DHT 22**", capaz de medir la temperatura y la humedad ambiente [22].

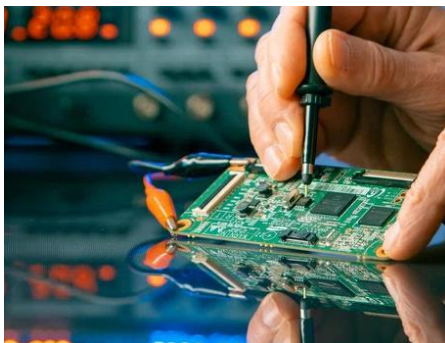


Ilustración 10. Ejemplo de un microcontrolador [20].

2.2.4. Tecnologías de comunicación

Para este proyecto es necesario comunicar el microcontrolador que recoge la información de los sensores, para que pueda transmitir dicha información al dispositivo receptor.

En este caso, existen múltiples opciones:

- Comunicación por cable: se pueden utilizar los puertos COM para transmitir la información al servidor o dispositivo receptor de la información. Este sistema presenta limitaciones, ya que complica la integración en distintos dispositivos.
- Conexión por Bluetooth: este sistema utiliza el protocolo Bluetooth para transmitir información. Presenta una gran complejidad a la hora de implementar las comunicaciones de todo el sistema, además de no permitir que el servidor sea tan fácilmente escalable.
- Conexión por sockets: los sockets permiten comunicar dos dispositivos mediante paquetes con el protocolo TCP o UDP. Presentan una implementación de bajo nivel que puede no ser útil en todos los escenarios [23].
- Conexión por websockets: permiten usar el protocolo HTTP para el enviar mensajes, lo que aporta las ventajas de UDP y la confiabilidad de TCP. Estas ventajas hacen a los websockets un método idóneo para implementar una arquitectura cliente-servidor.
Dentro de los websockets, encontramos una versión superior basada en el protocolo HTTPS que permite un nivel de seguridad superior, verificando las conexiones con certificados TLS (siglas para “Seguridad en la capa de transporte”) [24].

2.3. Tecnologías utilizadas

2.3.1. Comunicaciones

En este proyecto, las comunicaciones son cruciales. Todo el sistema depende del paso de mensajes confiables y rápidos que permitan mostrar la información en tiempo real.

Para elegir un buen método de comunicación se deben tener varios aspectos en cuenta:

- Usabilidad: si se quiere hacer el proyecto compatible con varios dispositivos distintos, se deben utilizar tecnologías disponibles en múltiples dispositivos. Por ello, se descarta el uso de puertos COM.
- Escalabilidad: el objetivo principal es poder monitorizar una o varias plantas al mismo tiempo, para ello es esencial que el sistema sea fácilmente escalable. Por ello, se descarta la comunicación por Bluetooth también, ya que complicaría la arquitectura cliente-servidor.
- Rapidez: los datos deben ser transmitidos en tiempo real. Esto deja como únicas opciones el uso de sockets o websockets.

En las primeras versiones del desarrollo, se utilizó un sistema cliente-servidor con sockets TCP. Este sistema funcionaba correctamente, pero ofrecía varias limitaciones al ser un protocolo de bajo nivel [24].

Es por ello por lo que se decidió utilizar websockets. En concreto, se utilizaron websockets basados en el protocolo HTTP. Se desestimó el uso del protocolo HTTPS por estos motivos:

- Ventajas insuficientes: en este contexto, el nivel de protección extra ofrecido sobre el protocolo HTTP, es innecesario. No se transmite ningún dato sensible, información personal o cualquier otro dato confidencial, sólo se transmite telemetría de los sensores.
- Complejidad: al ser un protocolo basado en certificados, la arquitectura es mucho más compleja. Todos los dispositivos conectados al servidor deben acceder con su certificado, lo que complica significativamente su despliegue y mantenimiento.

Es por esto por lo que se decidió el uso de websockets HTTP para gestionar las comunicaciones de todos los dispositivos y microcontroladores con el servidor.

2.3.2. Microcontroladores

El microcontrolador es la parte clave del dispositivo, ya que se encarga de recopilar la información de los sensores y transmitirla al servidor y al usuario.

Para este proyecto es esencial tener métodos de comunicación con los que poder implementar los sockets, además de tener un tamaño lo más reducido posible para que pueda quedar integrado en la planta sin obstaculizar.

Teniendo en cuenta lo anterior, la opción más adecuada es utilizar una placa **Arduino** que disponga de algún módulo WiFi ya integrado que permita reducir las dimensiones.

Es por ello por lo que se ha decidido utilizar una placa “**NodeMCU ESP8266 V2**”, la cual ofrece varias ventajas:

- Tamaño reducido, que permite a su vez reducir las dimensiones del dispositivo final.
- WiFi integrado, sin necesidad de utilizar módulos externos.
- Led que indica el estado de la placa.
- Pines accesibles.
- USB para poder programar la placa.

Esta placa es fácil de conseguir y tiene un coste reducido en el mercado [25].



Ilustración 11. Placa NodeMCU ESP8266 V2 [25].

2.3.3. Motores 3D

Para poder construir el entorno virtual donde se mostrará la información recopilada por los sensores, se ha tomado la decisión de utilizar el motor de videojuegos **Unity3D**.

Se ha elegido esta opción por varios motivos, entre ellos:

- Existencia de múltiples librerías y documentación que facilita la integración de AR en el proyecto.
- Experiencia previa en el uso de este motor gráfico.
- Compatibilidad con múltiples plataformas.
- Curva de aprendizaje más sencilla.

Además, para la lógica encargada de hacer funcionar la realidad aumentada, se ha utilizado la librería **"Vuforia"**. Esta librería ofrece compatibilidad con las principales plataformas móviles y con algunas VR/AR como **"Hololens 2"** o **"Magic Leap 2"**. [26]

2.3.4. Tecnologías AR

Para este proyecto, se ha utilizado AR basada en marcadores, debido a su simplicidad y a la facilidad que ofrece a la escalabilidad en entornos en los que se utilicen múltiples microcontroladores de forma simultánea. Para añadir un nuevo microcontrolador reconocible, basta con añadir un nuevo QR que pueda ser escaneado por la aplicación.

Como se ha explicado anteriormente, para aplicar esta tecnología se ha utilizado la librería **"Vuforia"**, configurada para utilizar marcadores. En este modo, la librería utiliza una imagen con unos patrones determinados que puede reconocer. Una vez reconocidos estos patrones, el dispositivo sabe de qué conjunto de sensores y dónde tiene que mostrar la información. Este sistema de reconocimiento permite que el sistema no necesite más que una cámara para poder funcionar, facilitando así su integración en distintos dispositivos [26].

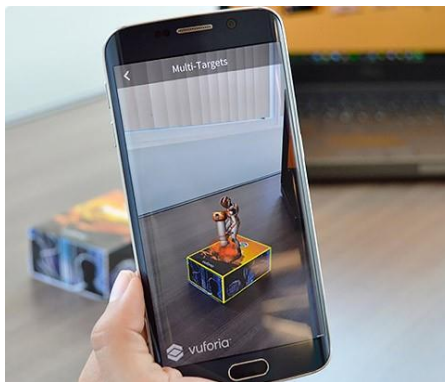


Ilustración 12. Ejemplo de reconocimiento de imágenes con **Vuforia** [27].

3. Análisis

En este apartado se realizará un análisis previo del proyecto en el que se explicarán y definirán varios aspectos claves a tener en cuenta. También se introducirán requisitos a cumplir y métricas de evaluación.

3.1. Descripción del proyecto

La idea original del proyecto era tratar de introducir datos de sensores del mundo real a un entorno virtual. Una vez estudiadas varias propuestas, se decidió que esa idea se iba a aplicar, en específico, a la monitorización de sensores ambientales que permitan controlar el estado de una planta. En consecuencia, también se optó por utilizar la realidad aumentada en vez de la realidad virtual, ya que se buscaba solucionar un problema en la realidad y no en una simulación 3D.

Para poder hacer uso de esta herramienta de monitorización, el usuario deberá ejecutar la aplicación en su dispositivo capaz de reproducir realidad virtual, posteriormente escaneará con su cámara el código QR asociado a cada planta, mostrando así sobre dicho código todos los datos que se están enviando en tiempo real al servidor y pudiendo tomar decisiones en base a esos datos.

3.2. Casos de uso

En este apartado se realizará un análisis de los posibles casos de uso que pueden aparecer durante la utilización de la aplicación.

Lo primero que hay que señalar es que, si bien este proyecto va dirigido a distintos públicos, sólo existe un tipo de usuario que utilice el sistema, además de la figura del administrador. Esto es debido a que la forma de interactuar con la aplicación es simple y no requiere de una distinción de roles específicos para cada público.

Se incluye un diagrama para representar estas acciones. De nuevo, la simplicidad de la interfaz de la aplicación hace que el diagrama sólo cuente con dos acciones posibles realizadas por el usuario, teniendo entre medias una acción realizada por el servidor y que también se ha querido representar. Por otro lado, el administrador puede realizar las funciones básicas del sistema, como la gestión de dispositivos y microcontroladores autorizados y la visualización de telemetrías anteriores. Se han añadido además los siguientes términos en las relaciones:

- “*include*”: sirve para representar que una acción sólo puede ocurrir si la acción a la que apunta sucede.
- “*expand*”: sirve para representar que es una acción opcional que se desbloquea con la ejecución de la opción anterior.

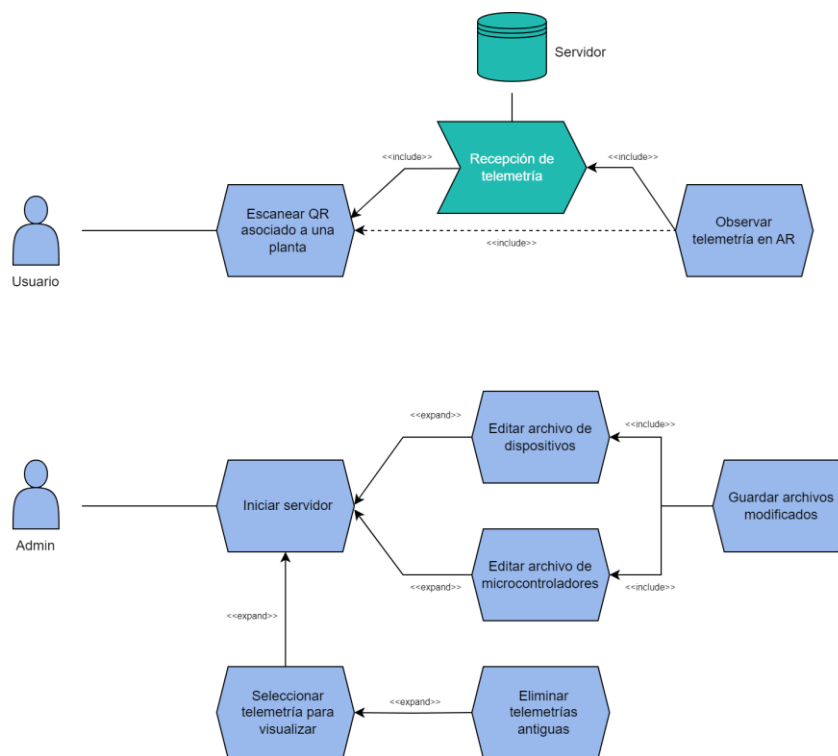


Ilustración 13. Diagrama de casos de uso.

A continuación, se va a detallar la información del diagrama en tablas.

Las tablas tendrán el siguiente formato:

ID	<id del caso de uso con el formato CU-Y-XX, donde Y representa si es un usuario (U) o un administrador (A) el que la realiza y XX es el número asignado para ese caso de uso>
Nombre	<nombre del caso de uso>
Descripción	<breve descripción del caso de uso>
Precondiciones	<acciones que se deben cumplir previamente>
Postcondiciones	<acciones que deben suceder posteriormente>

Tabla 2. Ejemplo de formato de tablas de casos de uso.

Siguiendo con este formato, se pueden representar los siguientes casos de uso:

ID	CU-U-01
Nombre	Escanear QR asociado a una planta.
Descripción	El usuario escaneará el código QR asociado a la planta de la que quiere recibir telemetría y enviará esa información al servidor.
Precondiciones	El usuario debe tener conexión con el servidor y tener la cámara activada mientras enfoca el código QR.
Postcondiciones	El usuario enviará al servidor un identificador de la planta sobre la que quiere recibir telemetría.

Tabla 3. CU-U-01

ID	CU-U-02
Nombre	Observar telemetría en AR.
Descripción	El usuario utilizará la tecnología AR para observar la telemetría asociada a cada planta en superposición con el entorno real.
Precondiciones	El usuario debe tener conexión con el servidor y haber escaneado el código QR asociado a la planta.
Postcondiciones	El usuario recibirá telemetría en tiempo real y de forma ininterrumpida para la planta que está observando.

Tabla 4. CU-U-02

ID	CU-A-01
Nombre	Iniciar servidor.
Descripción	El administrador podrá iniciar el servidor para comenzar a recibir y distribuir telemetría.
Precondiciones	El administrador debe abrir el programa para gestionar el servidor.
Postcondiciones	El administrador recibirá un mensaje y telemetrías en la consola habilitada para ello.

Tabla 5. CU-A-01.

ID	CU-A-02
Nombre	Editar archivo de dispositivos.
Descripción	El administrador podrá editar el archivo que contiene los dispositivos registrados para añadir o eliminar alguno de ellos.
Precondiciones	El administrador debe pulsar el botón “dispositivos”.
Postcondiciones	El administrador verá reflejados los cambios en el cuadro habilitado para ello.

Tabla 6. CU-A-02.

ID	CU-A-03
Nombre	Editar archivo de microcontroladores.
Descripción	El administrador podrá editar el archivo que contiene los microcontroladores registrados para añadir o eliminar alguno de ellos.
Precondiciones	El administrador debe pulsar el botón “microcontroladores”.
Postcondiciones	El administrador verá reflejados los cambios en el cuadro habilitado para ello.

Tabla 7. CU-A-03.

ID	CU-A-04
Nombre	Guardar archivos modificados.
Descripción	El administrador podrá guardar los cambios realizados en los archivos correspondientes.
Precondiciones	El administrador debe pulsar el botón “guardar” y tener abierto alguno de los archivos editables.
Postcondiciones	El administrador verá cómo se aplican los cambios realizados al archivo.

Tabla 8. CU-A-04.

ID	CU-A-05
Nombre	Visualizar telemetrías.
Descripción	El administrador podrá visualizar las telemetrías actuales y anteriores enviadas por los microcontroladores.
Precondiciones	El administrador debe seleccionar una de las telemetrías guardadas con el selector ubicado en la zona de los botones.
Postcondiciones	El administrador verá cómo se muestran las telemetrías en el cuadro habilitado para ello.

Tabla 9. CU-A-05.

ID	CU-A-06
Nombre	Eliminar telemetrías antiguas.
Descripción	El administrador podrá eliminar archivos antiguos con telemetrías.
Precondiciones	El administrador debe pulsar el botón “eliminar” y confirmar la operación en la ventana emergente.
Postcondiciones	El administrador verá como el archivo de telemetrías se elimina y deja de estar disponible para su visualización.

Tabla 10. CU-A-06.

3.3. Requisitos

En este apartado se van a definir los requisitos de usuario y del sistema. Para ello se va a utilizar la siguiente plantilla:

ID	<id del requisito con el formato RY-XX, donde Y representa si es de usuario (U) o de sistema (S) y XX es el número asignado para ese requisito>
Nombre	<nombre del requisito>
Descripción	<breve descripción del requisito>
Prioridad	<alta media baja>
Verificabilidad	<verificable no verificable>

Tabla 11. Ejemplo de formato de tabla de requisitos.

3.3.1. Usuario

En este subapartado se definirán los requisitos específicos de los dos usuarios del sistema, el usuario y el administrador.

ID	RU01
Nombre	Estado de conexión.
Descripción	El usuario podrá ver el estado de la conexión mediante un mensaje en la interfaz.
Prioridad	Media.
Verificabilidad	Verificable.

Tabla 12. RU01.

ID	RU02
Nombre	Cámara en la interfaz.
Descripción	El usuario tendrá disponible la imagen de lo que está capturando la cámara en la interfaz de la aplicación.
Prioridad	Alta.
Verificabilidad	Verificable.

Tabla 13. RU02.

ID	RU03
Nombre	Color de la interfaz.
Descripción	El usuario verá el texto de la interfaz de un color que le indicará el estado de la conexión, de forma complementaria al texto del estado de conexión.
Prioridad	Baja.
Verificabilidad	Verificable.

Tabla 14. RU03.

ID	RU04
Nombre	Escaneo de código QR.
Descripción	El usuario podrá escanear con la cámara en busca de códigos QR asociados a microcontroladores
Prioridad	Alta.
Verificabilidad	Verificable.

Tabla 15. RU04.

ID	RU05
Nombre	Visualización de datos.
Descripción	El usuario podrá ver la información en tiempo real en una ventana flotante mediante AR, situada encima del código QR.
Prioridad	Alta.
Verificabilidad	Verificable.

Tabla 16. RU05.

ID	RU06
Nombre	Múltiples plantas.
Descripción	El usuario podrá visualizar los datos de otra planta simplemente enfocando con su cámara el código QR de otro microcontrolador.
Prioridad	Alta.
Verificabilidad	Verificable.

Tabla 17. RU06.

ID	RU07
Nombre	Control de dispositivos.
Descripción	El administrador podrá eliminar o añadir dispositivos a la lista de autorizados mediante el uso de la interfaz del servidor.
Prioridad	Alta.
Verificabilidad	Verificable.

Tabla 18. RU07.

ID	RU08
Nombre	Telemetrías anteriores.
Descripción	El administrador tendrá acceso a un registro de telemetrías anteriores de cada planta, mediante el uso de la interfaz del servidor y que podrá consultar o eliminar del sistema.
Prioridad	Media.
Verificabilidad	Verificable.

Tabla 19. RU08.

ID	RU09
Nombre	Estado del sistema.
Descripción	El administrador tendrá acceso a los mensajes de estado del sistema mediante una consola integrada en la interfaz del servidor.
Prioridad	Baja.
Verificabilidad	Verificable.

Tabla 20. RU09.

3.3.2. Sistema

En este subapartado se desarrollarán los requisitos asociados a la funcionalidad del sistema.

ID	RS01
Nombre	Disponibilidad del sistema.
Descripción	El sistema debe estar siempre disponible para recibir y enviar telemetría, gestionando los posibles imprevistos en dicho proceso.
Prioridad	Alta.
Verificabilidad	Verificable.

Tabla 21. RS01.

ID	RS02
Nombre	Control de dispositivos.
Descripción	El sistema debe llevar un control de todos los dispositivos conectados que utilizan el servidor, para poder permitir o denegar su acceso.
Prioridad	Alta.
Verificabilidad	Verificable.

Tabla 22. RS02.

ID	RS03
Nombre	Protocolo seguro de transmisión.
Descripción	El sistema debe ser construido sobre un canal de transmisión seguro que proteja a los usuarios.
Prioridad	Alta.
Verificabilidad	Verificable.

Tabla 23. RS03.

ID	RS04
Nombre	Recepción de información.
Descripción	El servidor debe poder recibir información de los microcontroladores y distribuirla a los dispositivos.
Prioridad	Alta.
Verificabilidad	Verificable.

Tabla 24. RS04.

ID	RS05
Nombre	Tiempo de transmisión.
Descripción	El servidor debe ser capaz de transmitir la información de los controladores a los dispositivos receptores, al menos, con la misma cadencia con las que los recibe para poder garantizar una transmisión en tiempo real.
Prioridad	Alta.
Verificabilidad	Verificable.

Tabla 25. RS05.

ID	RS06
Nombre	Guardado de información.
Descripción	El servidor debe ser capaz de guardar y clasificar la información de los controladores en un historial de telemetrías.
Prioridad	Media.
Verificabilidad	Verificable.

Tabla 26. RS06.

ID	RS07
Nombre	Búsqueda del servidor.
Descripción	Los microcontroladores y dispositivos deben ser capaces de encontrar el servidor en su misma red local para iniciar la transmisión.
Prioridad	Media.
Verificabilidad	Verificable.

Tabla 27. RS07.

ID	RS07
Nombre	Tratamiento de errores.
Descripción	El sistema debe enviar mensajes de error a los dispositivos en caso de producirse alguno durante la transmisión.
Prioridad	Media.
Verificabilidad	Verificable.

Tabla 28. RS08.

ID	RS09
Nombre	Edición de listado de dispositivos.
Descripción	El servidor debe poder guardar la información cuando se edita el listado de dispositivos autorizados mediante la interfaz del servidor.
Prioridad	Media.
Verificabilidad	Verificable.

Tabla 29. RS9.

ID	RS10
Nombre	Visualización de telemetrías.
Descripción	El servidor debe poder recuperar todas las telemetrías cuando el usuario las solicita.
Prioridad	Media.
Verificabilidad	Verificable.

Tabla 30. RS10.

ID	RS11
Nombre	Envío de telemetrías.
Descripción	El sistema debe ser capaz de recibir la información del servidor en los dispositivos y proporcionarla a la interfaz.
Prioridad	Alta.
Verificabilidad	Verificable.

Tabla 31. RS11.

ID	RS12
Nombre	Escalabilidad.
Descripción	El sistema debe ser capaz de gestionar múltiples microcontroladores y dispositivos receptores de forma simultánea.
Prioridad	Alta.
Verificabilidad	Verificable.

Tabla 32. RS12.

ID	RS13
Nombre	Captura de vídeo.
Descripción	El sistema debe ser capaz de realizar una captura de vídeo en tiempo real para poder reconocer códigos QR.
Prioridad	Alta.
Verificabilidad	Verificable.

Tabla 33. RS13.

3.4. Matriz de trazabilidad

En este apartado se desarrollará la matriz de trazabilidad de los requisitos, que permite ver la relación entre los requisitos de usuario y los requisitos de sistema.

	RU01	RU02	RU03	RU04	RU05	RU06	RU07	RU08	RU09
RS01	X		X						
RS02					X		X		X
RS03					X				
RS04					X				
RS05					X				
RS06								X	
RS07	X		X						
RS08	X		X						X
RS09							X		
RS10								X	
RS11				X		X			
RS12				X		X			
RS13		X		X		X			

Tabla 34. Matriz de trazabilidad de requisitos

3.5. Metodología usada para el desarrollo del proyecto

Para el desarrollo de este trabajo se ha utilizado la metodología Kanban.

Kanban se basa en los principios de optimización, eficiencia y productividad. En base a estos, se quiere alcanzar el máximo rendimiento, centrándose en generar lo que se necesita, cuando se necesita y con la calidad necesaria. Para conseguir esto, Kanban utiliza la técnica *"Just in time"*.

La implementación de Kanban es sencilla, se dispone de un tablero que cuenta generalmente con 3 secciones en las que añadir y mover tarjetas con las tareas necesarias para completar el proceso:

- Pendiente
- En proceso
- Completado

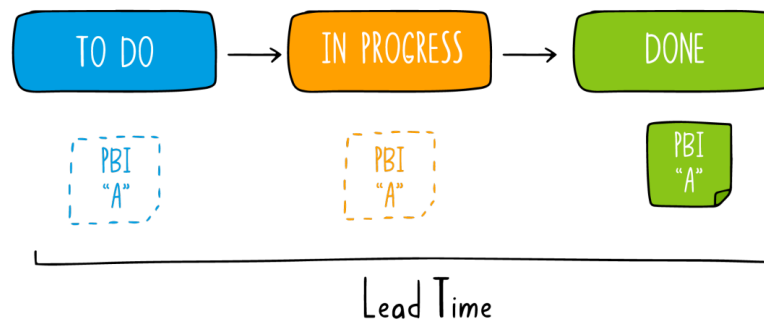


Ilustración 14. Ejemplo de Kanban [28].

Aunque permite añadir todos los pasos del flujo de tareas deseados para un proceso completo.

Se ha decidido utilizar esta metodología debido a la modularidad del proyecto, que permite dividir la mayoría de las tareas en las que trabajar en tareas independientes de las anteriores. Esto permite definir tareas con objetivos muy claros y llevar un registro de las tareas pendientes de realizar y de las que ya se han completado o han iniciado su proceso.

Además, este proyecto se puede dividir en 4 subproyectos principales:

- Funcionalidad de los microcontroladores
- Funcionalidad del servidor
- Funcionalidad de los dispositivos
- Comunicación entre ellos

Esta división hace que Kanban sea ideal para recapitular los avances en todos estos subproyectos y no olvidar o trabajar dos veces en alguna de las tareas [28].

4. Marco regulador

4.1. Marco legal

En el supuesto caso de que esta aplicación llegase a comercializarse, debe cumplir con la legislación vigente.

Pese a no manejar datos personales ni almacenar imágenes, esta aplicación utiliza datos de sensores capturadas en entornos privados y captura imágenes que procesa para comunicarse con el servidor, por lo que se debe garantizar el cumplimiento de las siguientes leyes:

[Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales:](#)

Esta ley publicada en el BOE con la referencia “BOE-A-2018-16673” es la encargada de regular el tratamiento de los datos personales de las personas físicas y limitar el uso de la informática para “garantizar el honor y la intimidad personal y familiar de los ciudadanos y el pleno ejercicio de sus derechos”.

Además, se reconoce el derecho de los ciudadanos para oponerse a que sus datos personales sean utilizados para fines distintos a los que justificaron su obtención [29].

Este texto legal obliga a informar a los usuarios de qué datos se recopilan, cómo se almacenarán y qué uso se dará a los datos, además de exigir el consentimiento expreso de los usuarios para poder recopilar dichos datos. Esta ley es aplicable sólo para la explotación de la aplicación en territorio nacional de España.

[REGLAMENTO \(UE\) 2016/679 DEL PARLAMENTO EUROPEO Y DEL CONSEJO de 27 de abril de 2016 relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos y por el que se deroga la Directiva 95/46/CE:](#)

En el caso de que la aplicación se lanzase en cualquier país de la Unión Europea, debería respetar, además de las leyes reguladoras de cada país, este reglamento europeo.

Esta ley establece normas para asegurar la protección de los datos personales de los individuos en la UE. En este texto, se exige el consentimiento explícito del usuario para el tratamiento de sus datos, además de imponer medidas de protección y exigir la notificación de posibles violaciones de estas para los datos almacenados.

Con la aprobación de este reglamento, se otorga a los ciudadanos derechos de acceso, rectificación y eliminación de sus datos personales, manteniendo en todo momento el control de estos [30].

4.2. Licencias

En este apartado se detallará todo el software utilizado para el desarrollo del proyecto, así como, sus respectivas licencias.

- **Arduino IDE y librerías:** para este proyecto se ha utilizado el IDE de Arduino. Este programa permite editar código y gestionar las librerías necesarias, además de permitir grabar el código en la memoria de la placa. Todo este software es de uso libre para cualquier fin, pero en caso de ser distribuido a terceros, se debe publicar el código fuente en caso de incluir librerías con la licencia *GNU General Public License*. Además, se debe cumplir con el artículo 4.d de la licencia LGPL [31].
- **Visual Studio:** se ha utilizado el IDE “Visual Studio” para programar tanto en C# para la parte de Unity, como en Python para el servidor. Está sujeto a la licencia MIT, se cede bajo licencia y no es objeto de venta [32].
- **Unity:** es el motor gráfico que se ha utilizado para desarrollar el apartado de realidad aumentada. Se debe elegir la licencia de uso adecuada, ya que están distribuidas por tramos dependiendo del nivel de ingresos producidos por el uso de esta herramienta [33].
- **Vuforia:** es el plugin utilizado junto con Unity para integrar la realidad aumentada en el proyecto. En el plan básico, el cual se ha utilizado, es de libre uso y permite la publicación y comercialización de aplicaciones que utilicen este software [34].
- **Word:** se ha utilizado para desarrollar esta memoria. Es de uso gratuito para estudiantes y usos educativos [35].
- **Draw.io:** es la herramienta utilizada para crear los diagramas en la memoria. Es de uso totalmente libre para cualquier fin [36].

5. Entorno socioeconómico

5.1. Planificación

En este apartado se expondrá la planificación tanto inicial como real del proyecto mediante diagramas de Gantt y de PERT que muestran el tiempo de realización de todas las tareas del proyecto.

5.1.1. Diagrama de Gantt

En este subapartado se encuentran dos diagramas de Gantt, el primero realizado con la planificación previa al desarrollo del proyecto y el segundo, una vez finalizado.

Estos diagramas se han desarrollado utilizando la aplicación web “**TeamGantt**”.

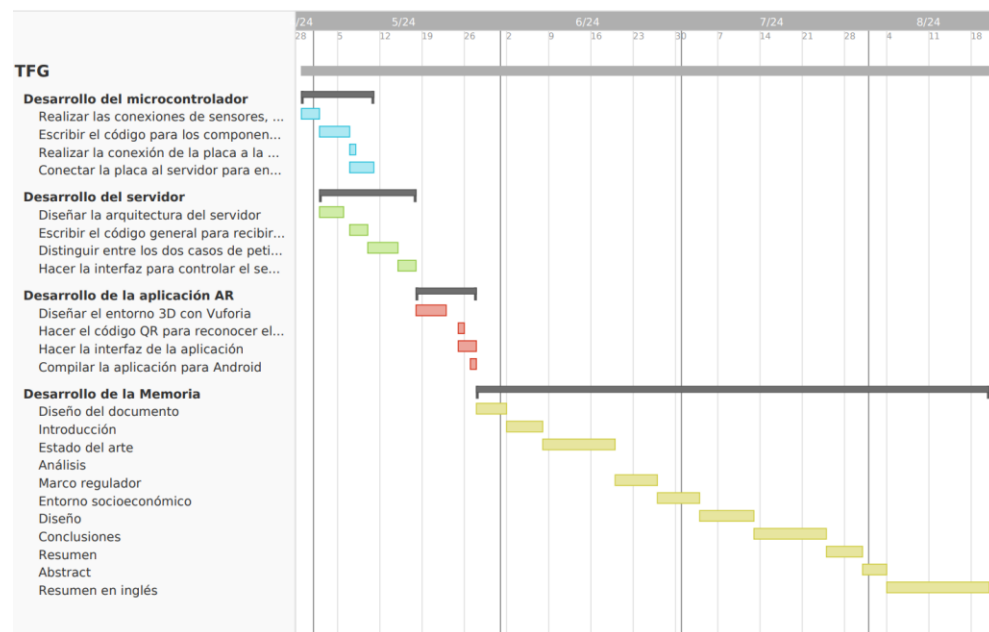


Ilustración 15. Diagrama de Gantt previo al comienzo del proyecto.

En este primer diagrama se ve la planificación inicial en la que se distinguen los cuatro módulos principales del proyecto, los tres propios del funcionamiento de la aplicación y la memoria.

En los proyectos de software muchas veces la planificación inicial no refleja la realidad al final del desarrollo, por lo que se incluye otro diagrama con la planificación final.

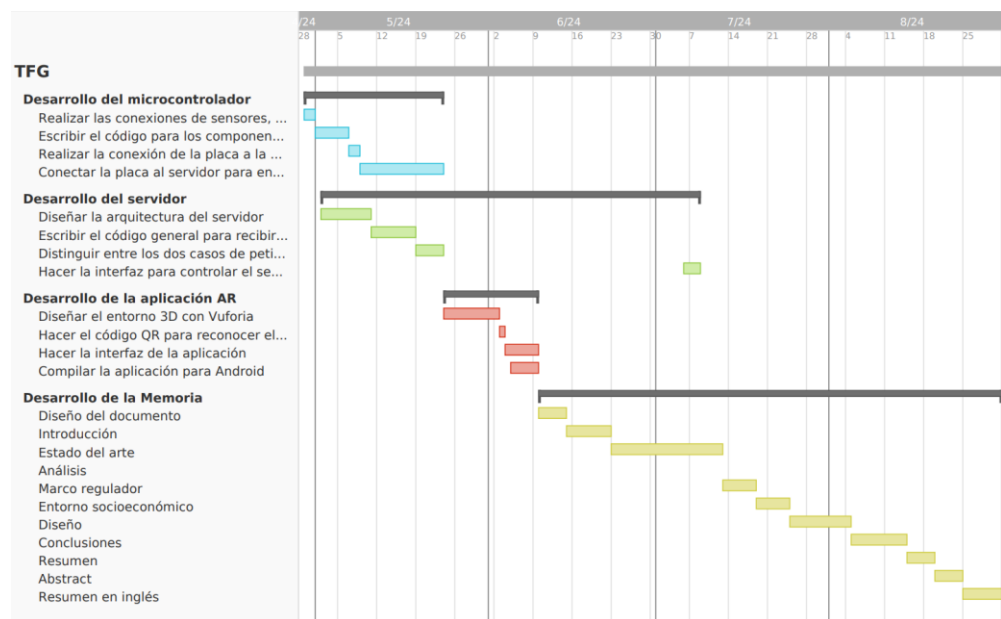


Ilustración 16. Diagrama de Gantt final.

En este diagrama se aprecia cómo surgen complicaciones durante el desarrollo que causan que el tiempo de una tarea sea subestimado, el de otras sobreestimado e incluso, que algunas tareas se pospongan a una fase más avanzada del desarrollo.

5.1.2. Diagrama de PERT

En el diagrama de PERT se pueden ver las distintas tareas, sus relaciones entre ellas y cómo se colocan en la línea temporal.

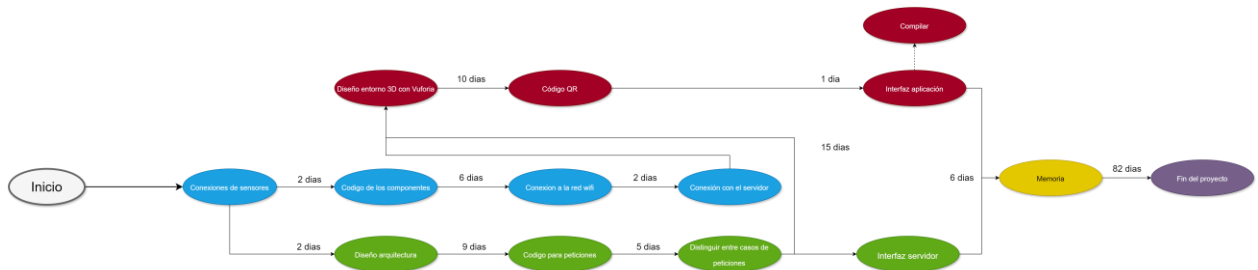


Ilustración 17. Diagrama de PERT.

5.2. Presupuesto para la realización del dispositivo

En este apartado se van a detallar los costes de llevar a un entorno de producción real este proyecto.

Costes de software:

Los costes de software para este proyecto son de 0€, debido a que todo el software utilizado es gratuito y de libre uso, excepto Unity que a partir de 200.000€ anuales de beneficio, requiere de una licencia [33].

Costes de Hardware:

Los costes de hardware incluyen principalmente el microcontrolador y sus componentes, se asume que los usuarios disponen de un ordenador en el que hacer funcionar el servidor, así como de un teléfono móvil que utilizar para la realidad aumentada.

Los precios han sido extraídos del portal web “**Amazon**”.

Componente	Precio (€)
Placa NodeMCU ESP8266 V2	8,49
Sensor DHT 22	5,8
Water sensor	4
Caja de conexiones para proteger los componentes	2,5
Total	20,79

Tabla 35. Precios de hardware

5.3. Presupuesto para el desarrollo del trabajo

Costes de personal:

Para desarrollar este trabajo se han necesitado unas 350 horas, divididas durante cuatro meses y medio, a una media de 2,5 horas al día.

Durante esas 350 se han asumido distintos roles, con el siguiente desglose de horas:

Puesto	Tiempo trabajado (horas)
Jefe de proyecto	60
Ingeniero de Hardware	50
Ingeniero de Software	80
Diseñador UI	30
Analista	70
Responsable de pruebas	60
Total	350

Tabla 36. Desglose de horas por puesto.

Por lo que, si tenemos en cuenta los sueldos para cada puesto según mercado, el coste es el siguiente:

Puesto	Coste (€/hora)	Total (€)
Jefe de proyecto	30	1.800
Ingeniero de Hardware	20	1.000
Ingeniero de Software	20	1.600
Diseñador UI	18	540
Analista	25	1.750
Responsable de pruebas	20	1.200
Total		7.890

Tabla 37. Desglose de coste de personal.

Costes de software:

En esta sección se va a detallar el coste de los productos de software utilizados para la realización del proyecto.

La agencia tributaria permite para este tipo de productos una amortización máxima del 33% anual en un periodo de hasta 6 años. Se va a optar por amortizar lo máximo posible anual. Teniendo en cuenta que la cantidad a amortizar se calcula de la siguiente forma [37]:

$$\text{Cantidad amortizable} = \text{precio_total} * \text{porcentaje_amortización} * (\text{Meses utilizado}/12)$$

Sabiendo que el desarrollo del proyecto ha durado 4,5 meses, el coste del software es el siguiente:

Producto	Coste (€)	Amortización (€)
Windows 11 Home	145.00	17,94
Microsoft Office 365 para estudiantes	0	0
Pycharm	0	0
Arduino IDE	0	0
Unity3D Personal	0	0
Visual Studio Code	0	0
Draw.io	0	0
Total		17,94

Tabla 38. Costes de software

Costes de hardware:

En la siguiente tabla se detallan los costes del hardware utilizado. La forma de calcular la amortización es igual que en el anterior apartado, sólo que en esta categoría, el porcentaje máximo anual es un 20% en un máximo de 10 años:

Producto	Coste (€)	Amortización (€)
Surface Pro 7 i7 256GB + teclado	1200	90
OnePlus Nord 3 5G	420	31,5
Impresora Canon PIXMA MG3650	59	4,4
Total		125,9

Tabla 39. Costes de hardware.

Costes de material fungible:

En esta tabla se detallan los costes fungibles que surgen durante el desarrollo:

Producto	Coste (€)
Tinta para la impresora	15,99
Total	15,99

Tabla 40. Costes de material fungible.

Costes indirectos:

Los costes indirectos son una aproximación, ya que es imposible calcular exactamente el coste de estos servicios que se ha utilizado:

Producto	Coste (€/mes)	Total (€)
Luz	15	67,5
Internet	25	112,5
Total		180

Tabla 41. Costes indirectos.

Costes totales:

Los costes indirectos son una aproximación, ya que es imposible calcular exactamente el coste de estos servicios que se ha utilizado:

Producto	Coste (€)
Personal	7.890
Software	17,94
Hardware	125,9
Material fungible	15,99
Costes indirectos	180
Total	8.229,83

Tabla 42. Costes totales.

A este coste hay que sumarle un 20% de beneficio para hacer rentable el desarrollo, por lo que el precio total sin IVA es de **9.875,16€**.

Si a este valor le sumamos el IVA del 21%, el precio final es de **11.948,94€**

5.4. Impacto socioeconómico

En este apartado se analizarán los beneficios que puede aportar este producto a la sociedad.

En un mundo en el que la tendencia es el aumento del consumo de vegetales, una herramienta que permita mejorar la producción de frutas y verduras puede ser muy beneficiosa para el sector.

Este producto puede ayudar en la producción a pequeña o gran escala de frutas, verduras y plantas en general, ya que permite monitorizar sus condiciones ambientales de forma rápida, simplemente mirando hacia ellas. Esto ayudará a aumentar la eficiencia, así como, el control sobre el estado actual y anterior de las plantas para poder tomar decisiones sobre su riego, humedad o temperatura, pudiendo llegar a automatizar el proceso utilizando los datos recopilados.

Esta tecnología, al ser relativamente barata de implementar, permite que también pueda ser utilizada por particulares en entornos privados. Desde el usuario que tiene un huerto en casa, al que tiene una pequeña planta en el salón.

También podría ser introducida en un entorno educativo, enseñando a los niños a cuidar de plantas de una forma más entretenida y dinámica, a la vez que se introduce una tecnología en auge como es la realidad aumentada.

Por último, el desarrollo de esta memoria y la documentación, han conllevado un trabajo de investigación que podrá ser utilizado en futuros trabajos como documentación. De la misma forma, también ha servido al autor de este trabajo para aprender sobre muchos aspectos como la realidad aumentada, el uso de placas Arduino, formas de comunicar dispositivos, el uso de motores 3D o el desarrollo de prototipos.

6. Diseño

6.1. Hardware

En este apartado se explicará la arquitectura del microcontrolador, así como las conexiones entre todos los elementos del sistema.

6.1.1. Esquema de conexiones del microcontrolador

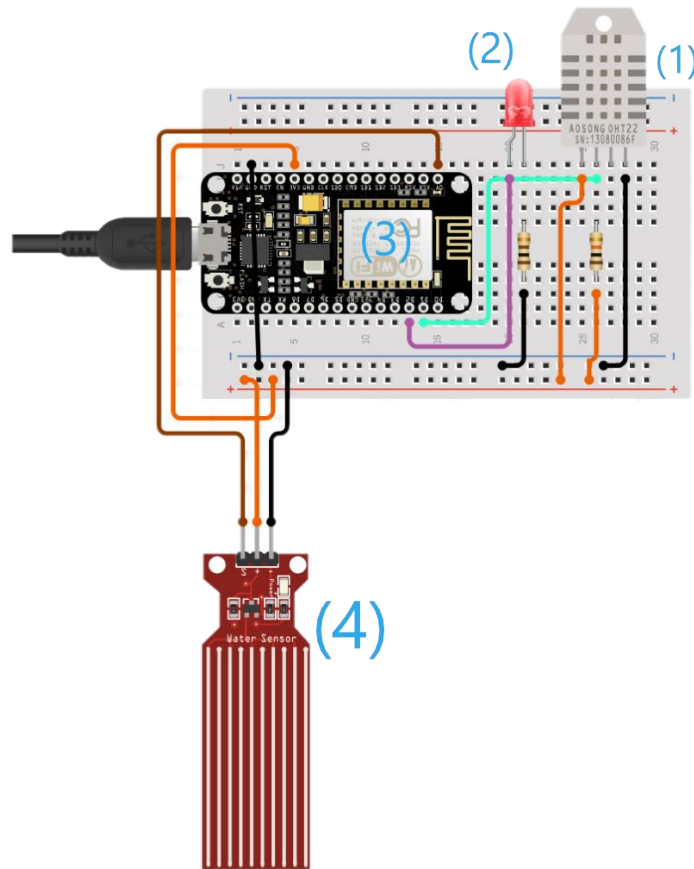


Ilustración 18. Esquema de conexiones del microcontrolador.

En este diagrama de conexiones se puede apreciar el microcontrolador **Node MCU (3)** en el centro del diseño. A este microcontrolador se le conectan los distintos componentes.

En primer lugar, es necesario conocer la temperatura ambiente, además de la humedad relativa, por lo que se ha añadido un sensor **DHT22 (1)**. Este componente es capaz de añadir ambas funcionalidades al sistema en un solo módulo. Tiene cuatro pines:

- Pin de VCC de 5V.
- Pin de toma de tierra.
- Pin de datos para transmitir la información.
- Pin sin uso/sin conectar que no ofrece funcionalidad.

En este sensor, como se puede apreciar en el diagrama es necesario añadir una resistencia para el pin de datos.

Otra necesidad crítica para el sistema es poder medir el nivel de agua, para ello se ha empleado un **Water Sensor (4)**. Este sensor es capaz de medir el nivel de agua, ofreciendo distintos valores dependiendo de la conductividad a la que se encuentre sometido. Sobre este sensor cabe destacar que tiene una vida útil bastante corta, debido a que, al estar expuesto a electricidad y agua, acaba generando corrosión en un corto periodo de tiempo. Esto se puede solucionar conectando el pin VCC a un puerto GPIO de 3.3V, permitiendo activar la corriente eléctrica sólo cuando se quiere realizar una lectura, ralentizando el proceso de corrosión, pero sacrificando precisión en los datos. Ambas configuraciones para este sensor son válidas y dependen de las necesidades del cliente que debe decidir entre priorizar precisión o durabilidad.

Este sensor dispone de los siguientes pines:

- Pin de VCC de 5V o 3.3V.
- Pin de toma de tierra.
- Pin de datos para transmitir la información.

Por último, es importante que el usuario conozca el estado de la conexión de su microcontrolador de una forma fácil e intuitiva. Por ello, se ha añadido un led naranja (2) que indica el estado de la conexión, manteniéndose encendido mientras se realiza el intercambio de telemetría con el servidor.



Ilustración 19. Fotografía de la carcasa del microcontrolador (se aprecia el código QR, el led en la parte superior y el sensor de agua clavado en la tierra).

6.1.2. Esquema de comunicaciones

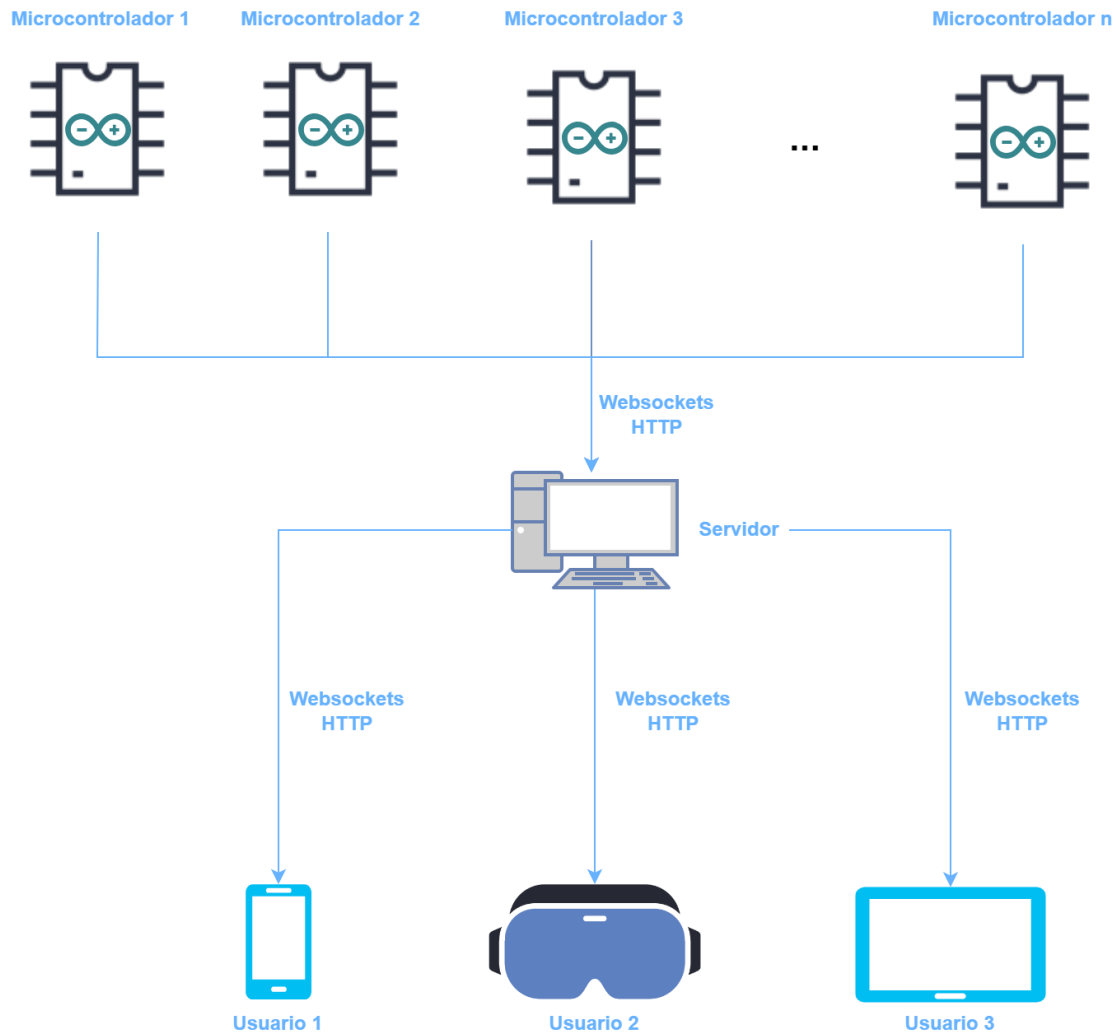


Ilustración 20. Esquema de comunicaciones.

En este esquema se aprecia como uno o varios controladores establecen una conexión con el servidor mediante **websockets HTTP**. El servidor recibe esas peticiones de conexión y sabe que vienen por parte de un microcontrolador, manteniendo el canal abierto y almacenando las telemetrías si no se supera el límite de tiempo para el intercambio de información.

Por otra parte, los dispositivos capaces de mostrar la información en realidad aumentada establecen una conexión con el servidor, también mediante **websockets HTTP**, siendo capaces de recibir telemetrías en tiempo real al escanear un código QR asociado a un microcontrolador.

6.2. Software

El código del proyecto está disponible en el siguiente repositorio:
<https://github.com/Carlossb02/TFG-CARLOS>

6.2.1. Conexión con el servidor

El primer paso para que todo el sistema pueda funcionar es establecer la conexión entre sus diferentes componentes.

Las conexiones se realizan en una red local conociendo la IP del servidor, pero surge el problema de que las IP son variables y no siempre es la misma, lo que hace que el usuario tenga que estar reconfigurando las direcciones de red para hacer funcionar el sistema. Además, en este proyecto reconfigurar el sistema cada vez que cambie la IP es inviable, ya que habría que volver a compilar y grabar el programa en el microcontrolador. Para solucionar este problema se han aplicado estas dos soluciones:

- Multicast DNS (mDNS).
- Búsqueda automática de la IP del servidor.

Puesto que, como se ha comentado previamente, todos los componentes del sistema están integrados en la misma red local, ambas opciones son aplicables y factibles para resolver el problema.

En el caso de la aplicación desarrollada en Unity, no existe soporte para mDNS sin librerías externas, por lo que la opción de la búsqueda de la IP en la red es una opción interesante. Tras implementar un algoritmo que prueba la conexión con todas las IP en el mismo rango de direcciones al que está conectado, siempre se encuentra el servidor en unos pocos segundos, solucionando el problema sin sacrificar la experiencia del usuario.

Sin embargo, este método es inviable en un microcontrolador como el que se está utilizando, ya que el mismo proceso de buscar y tratar de conectarse a las distintas IP lleva varios minutos, algo inviable en términos de usabilidad. Por ello, se ha decidido utilizar la librería *“ESP8266mDNS”*, la cual es capaz de encontrar y resolver servicios anunciados en la red, pudiendo obtener la IP del servidor directamente sin realizar una búsqueda en todas las direcciones. Para ello, el servidor debe anunciarse en la red, en este caso se ha utilizado la librería *“Zeroconf”* en Python, que permite crear un servicio en la IP y puerto actual, para posteriormente anunciarlo en la red [38].

6.2.2. Servidor

El microcontrolador se encarga de recopilar los datos de los sensores y transmitirlos, pero es necesario alguna forma de almacenar esa información y transmitirla al usuario final.

Para ello se creó un servidor en **Python**, primero utilizando la tecnología de sockets y, posteriormente, el servidor definitivo con la librería “**websockets**” de **Python**.

Al ejecutar la aplicación del servidor, se despliega una interfaz con la que el administrador puede interactuar. Además, se inicia el servidor HTTP y se anuncia en red el servicio creado con la librería “**Zeroconf**”, que permite que los microcontroladores lo encuentren.

El servidor construido en **Python** está siempre listo para recibir peticiones. A la hora de recibir una solicitud, distingue por su IP si se trata de un microcontrolador enviando telemetría, de un dispositivo esperando a recibir telemetría o si debe descartar la petición al tener un origen desconocido.

En el caso de recibir telemetría de un microcontrolador, se encarga de almacenarla en un JSON para tener un registro de todas las telemetrías, a la vez que marca el microcontrolador como conectado y guarda su última transmisión en dos diccionarios globales. Si no se transmite nueva telemetría en el tiempo de cadencia de los sensores, se procede a cerrar el canal de comunicación y a marcar el microcontrolador como desconectado.

Por otra parte, cuando un dispositivo se conecta esperando recibir telemetría, envía al servidor un mensaje indicando de qué microcontrolador quiere recibirla. En caso de existir y estar conectado, el servidor procede a reenviar al dispositivo la información almacenada en el diccionario global que se actualiza cada vez que se recibe por parte del microcontrolador, lo que garantiza una recepción de información en tiempo real. Si el microcontrolador estuviese desconectado, el dispositivo recibiría un mensaje indicando esta situación para no recibir telemetría desfasada.

```
Ejemplo_conectados = {"192.168.1.20": True, "192.168.1.30": False}
conectados = {}

Ejemplo_datos = {"192.168.1.20": [40.0, 24.0, "Bajo"], "192.168.1.30": [None, None, None]}
datos = {}
```

Ilustración 21. Ejemplo de diccionarios del servidor.

En la ilustración 18, se puede ver de una forma más clara la explicación anterior. En los ejemplos se ve cómo el primer microcontrolador está conectado (True) y tiene su última telemetría guardada en otro diccionario, sin embargo, el segundo microcontrolador ha perdido o cortado la conexión, por lo que no existe una telemetría reciente suya en el servidor. De esta forma se evita que los usuarios reciban información desfasada desde el mismo instante en el que el microcontrolador se desconecta del servidor.

6.2.3. Interfaz del servidor

Para facilitar la configuración del servidor, se ha construido una interfaz sencilla que ofrece varias funcionalidades.

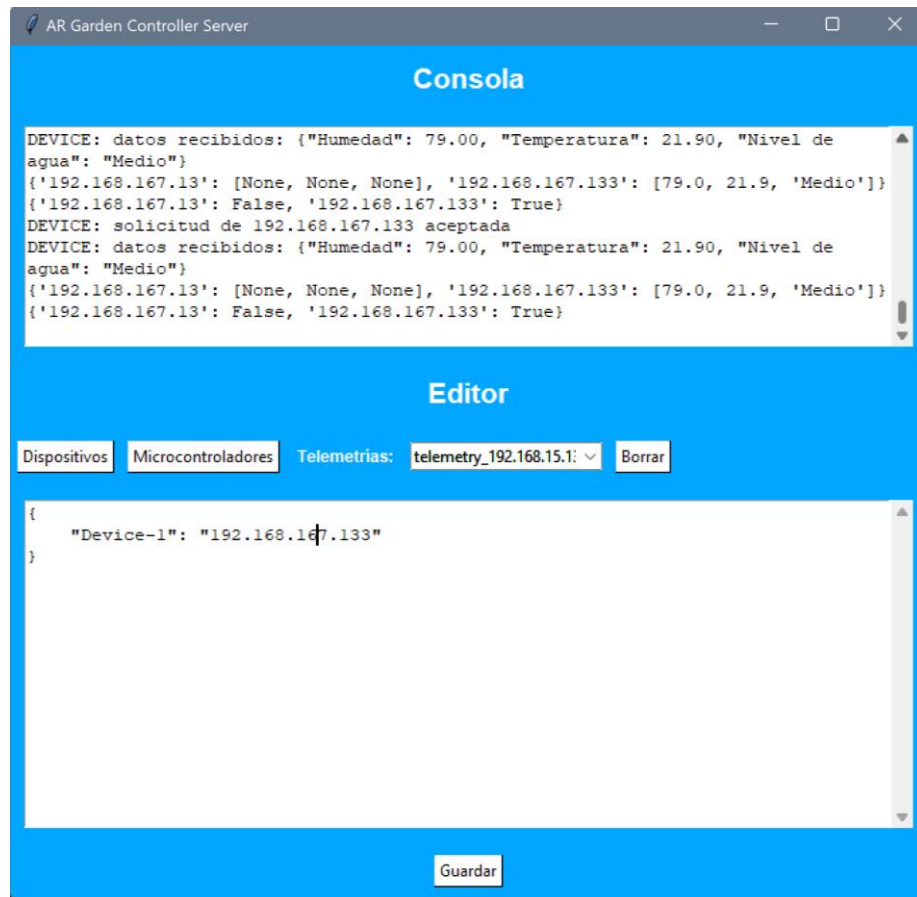


Ilustración 22. Interfaz del servidor.

Esta interfaz, como se puede apreciar, se divide en dos bloques principales:

Consola:

En este bloque el administrador del sistema puede ver todos los mensajes de control del sistema, desde información de las conexiones o transmisiones de datos, hasta errores de los componentes del sistema.

Editor:

En este bloque el administrador del sistema puede administrar los dispositivos y microcontroladores autorizados, añadiendo, eliminando o modificando la lista y aplicando los cambios en tiempo real.

Por otro lado, también se puede ver el historial de las telemetrías anteriores de los microcontroladores y borrar las que ya no tengan importancia.

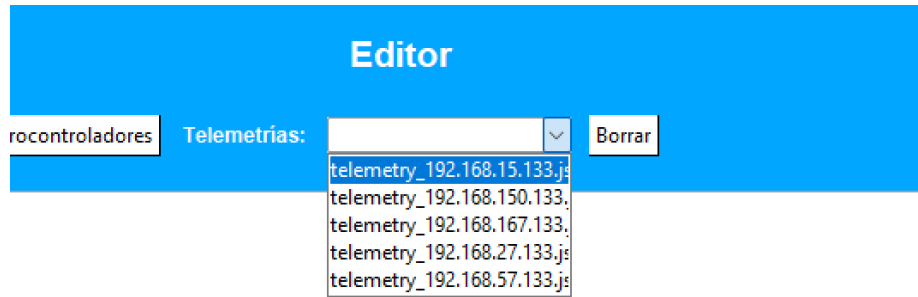


Ilustración 23. Vista del desplegable de telemetrías.

Como medida de protección para evitar la pérdida de información, antes de eliminar una telemetría, se debe confirmar la acción con una ventana emergente:

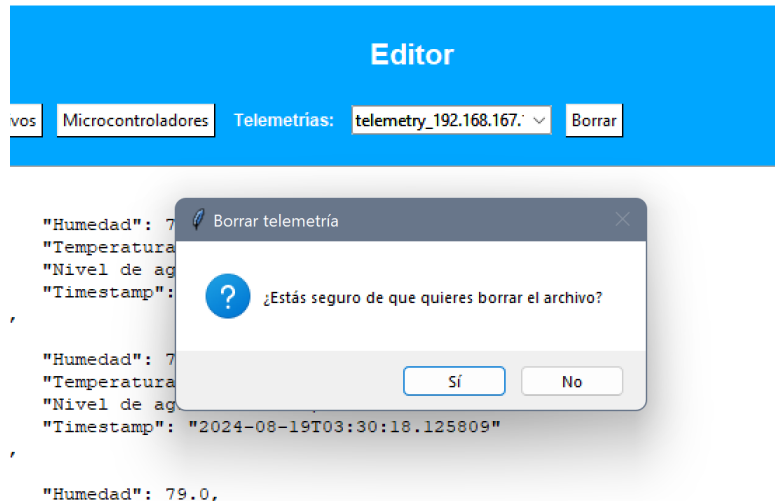


Ilustración 24. Ventana emergente para el borrado de telemetrías.

6.2.4. Aplicación AR

La aplicación AR está desarrollada con el motor de videojuegos **Unity3D**. Este motor permite compilar la aplicación para varios dispositivos **Windows**, **IOS** o **Android**, añadiendo también soporte para gafas VR o gafas AR como las **Hololens** de **Microsoft**.

Como se ha mencionado anteriormente, la aplicación está construida con **Vuforia**. Tras instalar la librería en Unity, lo primero que se debe hacer es sustituir la cámara principal por la “**ARCamera**” que proporciona **Vuforia**.

Posteriormente, hay que añadir el componente “**ImageTarget**”, al que se le debe asociar una librería de imágenes previamente validadas desde el portal para desarrolladores de **Vuforia**. Cabe destacar que la imagen debe tener patrones reconocibles, puesto que será utilizada para posicionar los elementos en el entorno.

Dentro de este “**ImageTarget**” se deben incluir todos los elementos que se quieren mostrar una vez se ha reconocido la imagen. En este caso, se han incluido el texto con la información y el cuadro que sirve para enmarcar ese texto.

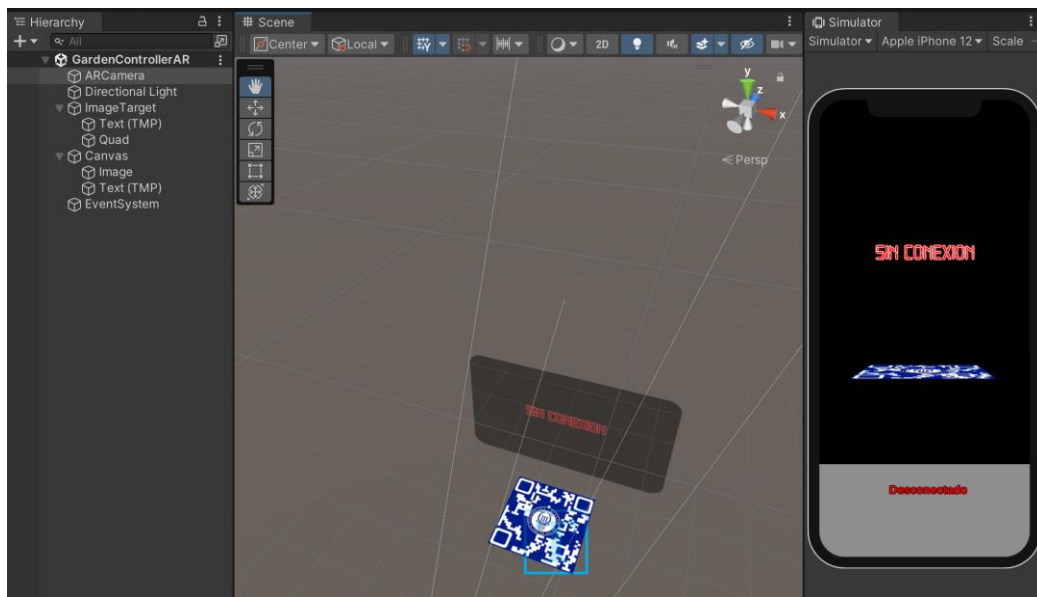


Ilustración 25. Escena del proyecto en Unity3D.

En la **Ilustración 25** se puede ver la interfaz del motor Unity. A la izquierda se observan todos los componentes de la escena, en el centro se ve cómo están contruidos y posicionados los elementos en la escena y a la derecha, se ve un ejemplo de cómo se mostraría la aplicación durante su ejecución (Aparece un fondo negro, ya que no hay ninguna cámara conectada al ordenador desde el que se está trabajando en el proyecto, este fondo será sustituido por la imagen de la cámara una vez se ejecute en un dispositivo compatible).

Por último, uno de los requisitos era informar al usuario del estado de la conexión con el servidor, para ello se ha incluido una pequeña interfaz que muestra los siguientes mensajes de estado en tiempo real.

Buscando el servidor:



Ilustración 26. Captura de la aplicación AR mientras se busca el servidor.

En la **Ilustración 26** se puede observar la interfaz de la aplicación mientras se produce la búsqueda del servidor en la red. Se aprecia cómo se muestra en amarillo que se está buscando el servidor en la red.

Conectado:



Ilustración 27. Captura de la aplicación AR mostrando datos.

En la **Ilustración 27** se observa que ya se ha producido la conexión con el servidor y se están recibiendo datos del microcontrolador llamado “Device-1” en tiempo real. Los datos se muestran sobre el código QR que la aplicación está reconociendo gracias a sus patrones y muestran los valores de temperatura, humedad y nivel de agua.

7. Pruebas

En este apartado se van a detallar las pruebas realizadas para validar el funcionamiento del sistema y el cumplimiento de los requisitos.

Para ello, se va a utilizar la siguiente plantilla:

ID	<id del caso de prueba con el formato PY-XX, donde Y representa si es de usuario (U) o de sistema (S) y XX es el número asignado para esa prueba>
Nombre	<nombre del caso de prueba>
Funcionamiento esperado	<resultado esperado del caso de prueba>
Funcionamiento validado	<resultado real del caso de prueba>
Requisitos involucrados	<lista de los requisitos involucrados en el caso de prueba>
Estado	<verificado no verificado>

Tabla 43. Plantilla para casos de prueba.

7.1. Validación de requisitos de usuario

ID	PU-01
Nombre	Estado de la conexión con el servidor.
Funcionamiento esperado	La interfaz mostrará un mensaje indicando el estado de conexión con el servidor.
Funcionamiento validado	La interfaz muestra en la parte inferior un mensaje que indica cuál es el estado actual de la conexión con el servidor.
Requisitos involucrados	RU-01, RU-03.
Estado	Verificado.

Tabla 44. PU-01.

ID	PU-02
Nombre	Escaneo de códigos QR con la cámara.
Funcionamiento esperado	La aplicación será capaz de utilizar la cámara de los dispositivos para reconocer códigos QR asociados a microcontrolador.
Funcionamiento validado	La aplicación es capaz de reconocer códigos QR asociados a microcontroladores utilizando la cámara.
Requisitos involucrados	RU-02, RU-04, RU-06.
Estado	Verificado.

Tabla 45. PU-02.

ID	PU-03
Nombre	Visualización de la información.
Funcionamiento esperado	La aplicación será capaz de mostrar la información recibida del servidor sobre el código QR escaneado en realidad aumentada.
Funcionamiento validado	La aplicación es capaz de mostrar la información recibida del servidor, en tiempo real, mostrándola sobre el código QR en realidad aumentada.
Requisitos involucrados	RU-02, RU-04, RU-05, RU-06.
Estado	Verificado.

Tabla 46. PU-03.

ID	PU-04
Nombre	Edición de dispositivos y microcontroladores autorizados.
Funcionamiento esperado	La interfaz del servidor permitirá editar y guardar en tiempo real la lista de dispositivos y microcontroladores autorizados.
Funcionamiento validado	La interfaz del servidor permite editar y guardar la lista de dispositivos y microcontroladores autorizados en tiempo real.
Requisitos involucrados	RU-07.
Estado	Verificado.

Tabla 47. PU-04.

ID	PU-05
Nombre	Visualización de telemetrías anteriores.
Funcionamiento esperado	La interfaz del servidor permitirá visualizar telemetrías anteriores.
Funcionamiento validado	La interfaz del servidor permite visualizar telemetrías anteriores mediante un desplegable situado en la parte central.
Requisitos involucrados	RU-08.
Estado	Verificado.

Tabla 48. PU-05.

ID	PU-06
Nombre	Visualización de mensajes de estado.
Funcionamiento esperado	La interfaz del servidor mostrará los mensajes de estado del sistema.
Funcionamiento validado	La interfaz del servidor muestra los mensajes de estado del sistema en una consola situada en la parte superior.
Requisitos involucrados	RU-09.
Estado	Verificado.

Tabla 49. PU-06.

7.2. Validación de requisitos del sistema

ID	PS-01
Nombre	Control de dispositivos.
Funcionamiento esperado	El servidor dispondrá de una lista de dispositivos y microcontroladores autorizados para poder autorizar o denegar el acceso.
Funcionamiento validado	El servidor dispone de una lista de dispositivos y microcontroladores, con sus direcciones IP asociadas que permite autorizar o denegar el acceso.
Requisitos involucrados	RS-02, RS-10.
Estado	Verificado.

Tabla 50. PS-01.

ID	PS-02
Nombre	Manejo de información.
Funcionamiento esperado	El servidor será capaz de recibir información de los microcontroladores y distribuirla a los dispositivos en tiempo real.
Funcionamiento validado	El servidor es capaz de recibir información de los microcontroladores y distribuirla a los dispositivos en tiempo real.
Requisitos involucrados	RS-01, RS03, RS04, RS05, RS07, RS-12.
Estado	Verificado.

Tabla 51. PS-02.

ID	PS-03
Nombre	Historial de telemetrías.
Funcionamiento esperado	El servidor será capaz de guardar y recuperar telemetrías antiguas cuando el usuario lo solicite.
Funcionamiento validado	El servidor es capaz de almacenar y recuperar telemetrías antiguas cuando el usuario lo solicita.
Requisitos involucrados	RS-06, RS-11.
Estado	Verificado.

Tabla 52. PS-03.

ID	PS-04
Nombre	Búsqueda del servidor.
Funcionamiento esperado	El servidor será visible en la red y todos los elementos del sistema serán capaces de encontrarlo automáticamente.
Funcionamiento validado	El servidor es visible en la red y todos los elementos del sistema son capaces de encontrarlo automáticamente.
Requisitos involucrados	RS-08.
Estado	Verificado.

Tabla 53. PS-04.

ID	PS-05
Nombre	Visualización de errores.
Funcionamiento esperado	El sistema mostrará mensajes de error en caso de producirse.
Funcionamiento validado	En caso de producirse errores, el sistema los muestra en la interfaz del servidor y de la aplicación.
Requisitos involucrados	RU-09.
Estado	Verificado.

Tabla 54. PS-05.

ID	PS-06
Nombre	Sistema escalable.
Funcionamiento esperado	El sistema será capaz de manejar varios microcontroladores y varios dispositivos receptores a la vez.
Funcionamiento validado	El sistema es capaz de manejar varios microcontroladores y dispositivos receptores al mismo tiempo.
Requisitos involucrados	RU-13.
Estado	Verificado.

Tabla 55. PS-06.

ID	PS-07
Nombre	Reconocimiento de códigos QR.
Funcionamiento esperado	El sistema será capaz de reconocer códigos QR y obtener la información del microcontrolador asociado.
Funcionamiento validado	El sistema es capaz de reconocer códigos QR y obtener la información del microcontrolador asociado.
Requisitos involucrados	RU-14.
Estado	Verificado.

Tabla 56. PS-07.

8. Conclusiones

En este apartado se realizará una reflexión acerca del proyecto, del cumplimiento de los objetivos marcados y de las posibles mejoras o añadidos que pueden ser realizados a este sistema en un futuro.

8.1. Retrospectiva de objetivos

En primer lugar, para realizar una retrospectiva, se deben retomar los objetivos marcados en un inicio en el punto 1.2:

Decidir qué datos capturar del mundo real, cuáles son los instrumentos adecuados para capturarlos e implementar toda la lógica capaz de realizar esas lecturas de datos:

Este objetivo se ha cumplido. Utilizando sensores de humedad, temperatura y nivel de agua se ha conseguido obtener datos relevantes para la monitorización de las plantas, consiguiendo implementar la lógica para hacerlos funcionar.

Crear un servidor capaz de tratar, almacenar y proporcionar dichos datos en tiempo real:

Para este objetivo se ha construido un servidor en Python que cumple con lo esperado desde el inicio del proyecto, habiendo sido validado durante las pruebas, por lo que también se puede considerar como un objetivo cumplido.

Crear canales de conexión seguros entre los dispositivos que capturan los datos, el servidor y el usuario final que los recibe:

Este objetivo se puede considerar cumplido pero no al nivel máximo esperado desde el inicio, ya que como se ha explicado durante el desarrollo de la memoria, no ha sido viable implementar el protocolo HTTPS y se ha tenido que utilizar el protocolo HTTP que proporciona menos seguridad.

Diseñar un entorno de AR capaz de mostrar los datos y de funcionar con distintos dispositivos como móviles, tablets o gafas de VR/AR:

Este objetivo se ha cumplido parcialmente, ya que, si bien la aplicación para móviles y tablets Android funciona correctamente, surgieron varios problemas a la hora de implementarla en gafas VR/AR. El primer problema fueron los múltiples errores a la hora de compilar y ejecutar la aplicación para las *Hololens 2*, siempre se producían errores de compatibilidad entre Unity y el dispositivo. Tras estos problemas, se decidió compilar la aplicación para las gafas *Meta Quest 2*, en estas gafas se consiguió que la ejecutar la aplicación, pero debido a la incompatibilidad de Vuforia con este dispositivo, fue imposible hacer funcionar la cámara AR, por lo que el dispositivo era inservible. De todos los dispositivos disponibles en el laboratorio

de la UC3M en Leganés, las únicas gafas compatibles son las **Hololens 2** [26]. Debido a los problemas técnicos previamente mencionados, fue imposible hacer funcionar la aplicación en este dispositivo, quedando este proyecto únicamente disponible para dispositivos móviles.

En general, se puede observar cómo se han cumplido la mayoría de los objetivos, aunque en alguno no se ha podido alcanzar el nivel máximo esperado. Además, se han obtenido conocimientos complementarios a los recibidos durante el estudio del grado de Ingeniería Informática que han permitido al autor de este trabajo profundizar en detalle en algunas áreas.

8.2. Futuros añadidos

Teniendo en cuenta los objetivos anteriores y las ideas descartadas o aplazadas durante el desarrollo del trabajo, en un futuro podría ser interesante incorporar las siguientes mejoras al proyecto:

- Añadir una bomba de agua conectada a un relé que permita regar la planta desde el dispositivo mientras se visualizan las telemetrías e incluso automatizar el proceso desde el servidor.
- Añadir un componente que permita proporcionar calor a la planta en caso de que la temperatura sea muy baja, automatizando nuevamente el proceso desde el servidor.
- Aplicar IA para reconocer el tipo de planta y obtener sus condiciones ambientales ideales. Esto se podría realizar con la librería “**TensorFlow**” de **Google**, que permite entrenar modelos para clasificar imágenes [39].
- Conseguir hacer funcionar la aplicación en gafas VR/AR, aplicando otras soluciones fuera de **Vuforia** o desarrollando una solución propia que permita hacer funcionar **Vuforia** en otras gafas como las **Meta Quest 2**

Como se puede apreciar, existen varias mejoras futuras que convertirían este proyecto en un producto mucho más interesante a nivel comercial, pero que escapan de los objetivos de este proyecto por cuestiones de tiempo y de viabilidad actual.

8.3. Conclusiones personales

En este apartado se van a exponer las conclusiones del autor de este trabajo de fin de grado una vez finalizado el proyecto.

Este trabajo ha sido un reto completamente nuevo, puesto que se han utilizado tecnologías que no entran dentro del alcance del grado en Ingeniería Informática, como la realidad aumentada, combinadas con otras tecnologías que sí se han trabajado en algunas de las asignaturas del grado.

Uno de los mayores aprendizajes ha sido en la planificación de proyecto, puesto que, al principio la lista de tareas pendientes era enorme y difícil de abarcar sin un buen plan de trabajo. También ha cambiado el modo de organizarse, ya que la mayoría de los trabajos realizados durante el desarrollo del grado, han sido desarrollados en parejas o en grupos, por lo que trabajar en todo y ser responsable del 100% del proyecto, asumiendo todos los roles al mismo tiempo, ha sido una forma completamente nueva de trabajar.

También cabe destacar las múltiples posibilidades que ofrece un trabajo basado en realidad aumentada, puesto que, durante el desarrollo, han ido surgiendo añadidos que no estaban previstos en un inicio o grandes cambios en la estructura básica del código que modifican internamente todo el funcionamiento. Muchas de esas ideas para añadidos, desgraciadamente, han tenido que ser descartadas por falta de tiempo para desarrollar y probar el dispositivo, pero pueden ser integradas en futuras versiones.

En general, ha sido un proyecto que ha aportado muchos conocimientos nuevos y ampliado otros. Embarcarse en un proyecto así, desde cero y sin nada definido, es una gran forma de aprendizaje que obliga a realizar un estudio y una investigación previa, además de solucionar por cuenta propia los múltiples problemas que van surgiendo durante el desarrollo.

9. Bibliografía

- [1] Deloitte, «¿Qué es la Industria 4.0?», [En línea]. Available: <https://www2.deloitte.com/es/es/pages/manufacturing/articles/que-es-la-industria-4.0.html>. [Último acceso: 8 Julio 2024].
- [2] SAP, «¿Qué es la realidad aumentada (AR)?», [En línea]. Available: <https://www.sap.com/spain/products/scm/industry-4-0/what-is-augmented-reality.html>. [Último acceso: 08 Julio 2024].
- [3] M. Abraham y M. Annunziata, «La realidad aumentada ya está mejorando el rendimiento de los trabajadores», Harvard Business Review, 13 Marzo 2017. [En línea]. Available: <https://hbr.org/2017/03/augmented-reality-is-already-improving-worker-performance>. [Último acceso: 08 Julio 2024].
- [4] Precedence Research, «Augmented Reality And Virtual Reality Market», [En línea]. Available: <https://www.precedenceresearch.com/augmented-reality-and-virtual-reality-market>. [Último acceso: 09 Julio 2024].
- [5] T. Zarraonandia, P. Díaz, A. Montero y I. Aedo, «Magic Flowerpot», Universidad Carlos III de Madrid, Barcelona, 2019.
- [6] H. Hinojosa Iranzo y E. Miralles Llauradó, «Uso de realidad aumentada para la monitorización de información obtenida con RPAS», Universitat Politècnica de Catalunya, Barcelona, 2020.
- [7] V. Dayan, «Garduino – the Smart Garden With Arduino», Autodesk Instructables, 10 Marzo 2020. [En línea]. Available: <https://www.instructables.com/Garduino-the-Smart-Garden-With-Arduino/>. [Último acceso: 27 Agosto 2024].
- [8] M. Korchi, «Arduino Plant Monitor», Autodesk Instructables, 16 Enero 2017. [En línea]. Available: <https://www.instructables.com/Arduino-Plant-Monitor/>. [Último acceso: 27 Agosto 2024].
- [9] H. E. Lowood, «Virtual Reality», 11 Julio 2024. [En línea]. Available: <https://www.britannica.com/technology/virtual-reality/Entertainment>. [Último acceso: 16 Julio 2024].
- [10] IFEMA, «Cómo funcionan las gafas de realidad virtual», 04 Agosto 2022. [En línea]. Available: <https://www.ifema.es/noticias/tecnologia/como-funcionan-gafas-vr>. [Último acceso: 16 Julio 2024].
- [11] J. A. P. Estapé, «Gafas de realidad virtual HTC Vive, características y configuración», Computer Hoy, 06 Abril 2016. [En línea]. Available: <https://computerhoy.com/noticias/zona-gaming/gafas-realidad-virtual-htc-vive-caracteristicas-configuracion-42957>. [Último acceso: 16 Julio 2024].

- [12] Business, «Diferencias entre realidad virtual y realidad aumentada,» INESDI, 22 Marzo 2024. [En línea]. Available: <https://www.inesdi.com/blog/realidad-aumentada-y-realidad-virtual-diferencias/>. [Último acceso: 17 Julio 2024].
- [13] J. Azurdy, «Realidad Virtual y Aumentada: Ventajas y Desventajas,» encora, 05 Junio 2020. [En línea]. Available: <https://www.encora.com/es/blog/realidad-virtual-y-aumentada-ventajas-y-desventajas>. [Último acceso: 17 Julio 2024].
- [14] J. Pocayo, «Microsoft refuerza su compromiso con Hololens 2,» Metaverse News, 20 Junio 2023. [En línea]. Available: <https://metaverse-news.es/microsoft-refuerza-su-compromiso-con-hololens-2/>. [Último acceso: 17 Julio 2024].
- [15] P. J. Saéz Martínez, «AR vs VR: Diferencias entre realidad aumentada y realidad virtual,» Onirix, 11 Abril 2023. [En línea]. Available: <https://www.onirix.com/es/ar-vs-vr/>. [Último acceso: 26 Julio 2024].
- [16] Enhance, «Marcos de realidad aumentada: tipos, elementos y tendencias,» 14 Octubre 2022. [En línea]. Available: <https://www.ienhance.co/blog/marcos-de-realidad-aumentada-tipos-elementos-y-tendencias>. [Último acceso: 03 Agosto 2024].
- [17] M. Carter y B. Egliston, «Pokémon Go Wants to Make 3D Scans of the Whole World for 'Planet-Scale Augmented Reality Experiences.' Is that Good?,» Nextgov, 18 Junio 2020. [En línea]. Available: <https://www.nextgov.com/emerging-tech/2020/06/pokemon-go-wants-make-3d-scans-whole-world-planet-scale-augmented-reality-experiences-good/166232/>. [Último acceso: 17 Julio 2024].
- [18] A. C. Carrasco, «¿Qué es un motor de videojuegos?,» Observatorio GATE, 04 Julio 2018. [En línea]. Available: <https://blogs.upm.es/observatoriogate/2018/07/04/que-es-un-motor-de-videojuegos/>. [Último acceso: 17 Julio 2024].
- [19] Unity, «Unity Use Cases: Solutions for Gaming & Industry | Unity,» [En línea]. Available: https://unity.com/_next/image?url=https%3A%2F%2Fcdn.sanity.io%2Fimages%2Ffuvbjlp%2Fproduction%2Fbdd4ab9208069e280930c85cf38e5015b3582535-1910x938.png&w=3840&q=75. [Último acceso: 17 Julio 2024].
- [20] Universidad Europea, «Microcontrolador: ¿Qué es y para qué sirve?,» 06 Mayo 2024. [En línea]. Available: <https://universidadeuropea.com/blog/que-es-microcontrolador/>. [Último acceso: 18 Julio 2024].
- [21] BBVA API Market, «Mejores alternativas a Arduino: del Do it Yourself al Internet de las Cosas,» BBVA API Market, [En línea]. Available: <https://www.bbvaapimarket.com/es/mundo-api/mejores-alternativas-arduino-del-do-it-yourself-al-internet-de-las-cosas/>. [Último acceso: 24 Julio 2024].
- [22] Oracle, «7 General Purpose Input/Output (GPIO),» Oracle, 2012. [En línea]. Available: https://docs.oracle.com/javame/config/cldc/rel/3.4/core/da/html/device_access/gpio.htm. [Último acceso: 27 Julio 2024].
- [23] M. J. Donahoo y K. L. Calvert, «TCP/UDP Sockets,» de *TCPIP Sockets in C*, Burlington, Elsevier Inc, 2009, pp. 3-4.

- [24] A. Lombardi, «WebSockets,» de *WebSockets*, Sebastopol, O'Reilly Media Inc, 2015, pp. 7-10.
- [25] L. d. V. Hernández, «NodeMCU tutorial paso a paso desde cero,» Programar fácil, [En línea]. Available: <https://programarfácil.com/podcast/nodemcu-tutorial-paso-a-paso/>. [Último acceso: 20 Julio 2024].
- [26] Vuforia, «Vuforia - Platform support - Recommended Devices,» Vuforia, [En línea]. Available: <https://developer.vuforia.com/library/platform-support/recommended-devices>. [Último acceso: 24 Julio 2024].
- [27] Vuforia, «Multi Targets,» [En línea]. Available: <https://developer.vuforia.com/sites/default/files/vuforia-library/articles/Features/Images/Multi%20Targets/feature-image-multi-targets.jpeg>. [Último acceso: 2024 Julio 20].
- [28] Consejería de Economía, Ciencia y Agenda Digital de la Junta de Extremadura, «Agile Process Management y Agile Framework: KANBAN,» Mérida, Junta de Extremadura, 2023, pp. 5-50.
- [29] Boletín Oficial del Estado, «Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales,» 07 Diciembre 2018. [En línea]. Available: <https://www.boe.es/buscar/act.php?id=BOE-A-2018-16673>. [Último acceso: 08 Agosto 2024].
- [30] Parlamento Europeo y Consejo de la Unión Europea, «REGLAMENTO (UE) 2016/679 DEL PARLAMENTO EUROPEO Y DEL CONSEJO de 27 de abril de 2016 relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos,» 27 Abril 2016. [En línea]. Available: <https://www.boe.es/doue/2016/119/L00001-00088.pdf>. [Último acceso: 08 Agosto 2024].
- [31] Arduino, «Licensing for products based on Arduino,» 08 Julio 2024. [En línea]. Available: <https://support.arduino.cc/hc/en-us/articles/4415094490770-Licensing-for-products-based-on-Arduino>. [Último acceso: 08 Agosto 2024].
- [32] Microsoft, «TÉRMINOS DE LICENCIA DEL SOFTWARE DE MICROSOFT,» Microsoft, [En línea]. Available: <https://code.visualstudio.com/license?lang=es>. [Último acceso: 12 Agosto 2024].
- [33] Unity, «Unity Editor Software Terms,» Unity, 6 Noviembre 2023. [En línea]. Available: <https://unity.com/es/legal/editor-terms-of-service/software>. [Último acceso: 12 Agosto 2024].
- [34] Vuforia, «Pricing and Licensing Options,» Vuforia, [En línea]. Available: <https://developer.vuforia.com/library/faqs/pricing-and-licensing-options>. [Último acceso: 12 Agosto 2024].

- [35] Microsoft, «Office 365 Education,» Microsoft, [En línea]. Available: <https://www.microsoft.com/es-es/education/products/office>. [Último acceso: 12 Agosto 2024].
- [36] Draw.io, «Usage terms for diagrams created in draw.io,» Draw.io, [En línea]. Available: <https://www.drawio.com/doc/faq/usage-terms>. [Último acceso: 12 Agosto 2024].
- [37] Agencia Tributaria, «1. Por coeficientes de amortización lineal,» 1 Enero 2015. [En línea]. Available: <https://sede.agenciatributaria.gob.es/Sede/ayuda/manuales-videos-folletos/manuales-practicos/irpf-2023/c07-rendimientos-actividades-economicas-estimacion-directa/fase-1-determinacion-rendimiento-neto/amortizaciones-dotaciones-ejercicio-fiscalmente-deducib>. [Último acceso: 15 Agosto 2024].
- [38] Equipo editorial de IONOS, «Multicast DNS: la resolución de nombres para redes locales,» IONOS, 10 Agosto 2020. [En línea]. Available: <https://www.ionos.es/digitalguide/servidores/know-how/multicast-dns/>. [Último acceso: 2024 Agosto 22].
- [39] TensorFlow, «Por qué TensorFlow,» [En línea]. Available: <https://www.tensorflow.org/about?hl=es-419>. [Último acceso: 23 Agosto 2024].

10. Glosario de términos

- **VR:** realidad virtual.
- **AR:** realidad aumentada.
- **Unity:** motor de videojuegos multiplataforma que permite desarrollo de entornos 3D.
- **Sockets:** canales de comunicación que permiten intercambiar datos.
- **TCP:** protocolo de transmisión de mensajes basado en conexiones. Es más fiable y más lento.
- **UDP:** protocolo de transmisión de mensajes sin conexión. Es menos fiable y más rápido.
- **Hardware:** conjunto de componentes físicos de un sistema.
- **Software:** código utilizado para controlar el comportamiento de un sistema informático.
- **USB:** Universal Serial Bus
- **HTTP:** Protocolo de Transferencia de Hipertexto. Es un protocolo de transferencia de información a través de archivos en la web.
- **HTTPS:** Protocolo Seguro de Transferencia de Hipertexto. Es una versión segura de HTTP.