

Diseño e implementación de un sistema inteligente de gestión de residuos en la UAB: The UAB Campus as a Living Lab

Víctor-Javier Rey Gómez

8 de febrero de 2024

Resumen— Actualmente, existe una urgente necesidad de gestionar la creciente cantidad de residuos producidos. En este proyecto, se pretende implementar un sistema de gestión de residuos en el campus de la UAB mediante el uso de tecnología avanzada. Este sistema inteligente se basa en la creación de contenedores inteligentes capaces de proporcionar información en tiempo real a través de la tecnología LoRaWAN, lo que permite la conexión de estos contenedores a servidores en Internet. La información recopilada se utilizará para optimizar las rutas de recolección y, con el tiempo, identificar las áreas prioritarias para reducir la generación de residuos dañinos. Además, se está desarrollando una aplicación móvil para los usuarios del campus, diseñada para fomentar el reciclaje mediante la oferta de incentivos y premios. Esta aplicación utilizará tecnología de escaneo de imágenes y detección basada en inteligencia artificial para clasificar de manera óptima los residuos.

Palabras clave— IA, Ciudades inteligentes, Aplicación web/app, React, GPT Vision, Computación, Reciclaje, LoRaWAN, Campus UAB, Gestión inteligente y Residuos.

Abstract— Currently, there is an urgent need to manage the increasing amount of waste being produced. In this project, the aim is to implement a waste management system at the UAB campus using advanced technology. This intelligent system is based on the creation of smart containers capable of providing real-time information through LoRaWAN technology, allowing these containers to connect to Internet servers. The collected information will be used to optimize collection routes and, over time, identify priority areas for reducing the generation of harmful waste. Additionally, a mobile application is being developed for campus users, designed to encourage recycling by offering incentives and rewards. This application will use image scanning and artificial intelligence-based detection to classify waste optimally.

Keywords— AI, SmartCities, Mobile Web/APP, React, GPT Vision, Computing, Recycling, LoRaWAN, UAB campus, Smart management and Waste.

1 INTRODUCCIÓN

EN la era actual, donde la sostenibilidad se ha convertido en un imperativo, las instituciones tienen el desafío y la responsabilidad de incorporarse al camino haciendo uso de prácticas ambientalmente responsables. Una de las áreas en las que la innovación ha tomado protagonismo es la gestión de residuos.

- E-mail de contacto: victorrej31879@gmail.com
- Mención realizada: Ingeniería del Software
- Trabajo autorizado por: Fernando Luis Vilariño Freire (Ciencias de la computación)
- Curso 2023/24

La gestión inteligente de residuos[27] incorpora características y metodologías diseñadas para mejorar la eficiencia en la recolección y disposición de residuos, la planificación inteligente de rutas, la minimización de la generación de desechos y la reducción del impacto ambiental, entre otros. Se destaca la tendencia clara hacia la integración de tecnologías como el Internet de las Cosas (IoT)[5], la Inteligencia Artificial (IA)[6], el Análisis de Datos, el Blockchain[15], la Computación en la Nube[14] y el BigData[4], por mencionar algunos.

Con una visión hacia el futuro, la Universidad Autónoma de Barcelona (UAB) busca unirse a esta revolución. El interés de la UAB se centra en adaptar su campus al concepto

de SmartCity[29] y en este proyecto se propone implementar un sistema avanzado de gestión de residuos.

Desde el Centro de visión por computador (CVC), la propuesta se centra en la adaptación y optimización de los contenedores de residuos existentes, equipándolos con tecnologías emergentes que permitan una recolección y disposición más efectiva y eficiente. Este proyecto no sólo busca adaptar lo que ya se realiza en otros contextos, sino también introducir enfoques novedosos que reflejen la identidad y las necesidades específicas de la UAB. Existe un programa en la UAB llamado *The UAB Campus as a Living Lab*[24]. Con el que se cooperará para obtener información ya recopilada por ellos que puede servir para implementar este proyecto.

Este documento está organizado en diferentes apartados y secciones que explican como se ha ido desarrollando este proyecto, se puede encontrar una lista de objetivos, metodologías, requisitos, también un estado del arte, análisis de viabilidad, una planificación, presupuestos, resultados y se finalizará con una conclusión. A parte podremos encontrar información extra y detallada en los anexos disponibles al final del mismo.

1.1. Objetivos

Para realizar este proyecto se proponen los siguientes objetivos sobre que se debe hacer durante la duración del mismo.

Generales:

- **Implementación de la Gestión Inteligente de Residuos:** Adoptar una metodología de gestión inteligente de residuos en la UAB para mejorar la eficiencia, sostenibilidad y eficacia del proceso de recolección y disposición de residuos en el campus.
- **Aprender a utilizar la tecnología LoRaWAN[7]:** Mediante experimentación y ensayo y error conseguir interconexiones de sensores y tener una perspectiva de esta tecnología más amplia.
- **Hacer uso de sensores IoT[33]:** Conseguir utilizar diversos sensores capaces de registrar datos físicos y poder combinar la información para implementarla en el proyecto.

Específicos:

- **Integración Tecnológica:** Implementar soluciones basadas en IoT, IA y análisis de datos para optimizar la recolección y tratamiento de residuos.
- **Rediseño de Contenedores:** Adaptar los contenedores existentes para que sean inteligentes, facilitando la segregación de residuos, optimizando la recolección y proporcionando información en tiempo real sobre su contenido y estado.
- **Optimización de Rutas:** Utilizar herramientas de análisis de datos y algoritmos para diseñar rutas inteligentes de recolección de residuos, reduciendo costes y tiempo.
- **Monitorización y Feedback:** Establecer sistemas de monitorización en tiempo real para seguir el flujo y la

disposición de los residuos, y proporcionar *feedback* constante para la mejora continua del sistema.

- **Integración con Otros Sistemas:** Asegurar que la gestión inteligente de residuos se integre de manera coherente con otros sistemas y proyectos de SmartCity en el campus.

A futuro:

- **Sostenibilidad Económica:** Buscar alianzas, patrocinadores o modelos de financiación que aseguren la sostenibilidad económica del proyecto a largo plazo y expansión a otros lugares.

1.2. Estado del arte

En este apartado, se ofrecerá una visión general del estado de otras implementaciones de gestión de residuos inteligentes a nivel global, en España y, específicamente, en universidades españolas.

A nivel mundial, la gestión inteligente de residuos se está consolidando como una realidad en la administración urbana y medioambiental. Las tecnologías emergentes como IoT, IA, *blockchain*, combinado con Análisis de Datos, computación en la nube, *Big Data* y Software para aplicaciones en *smartphone* están siendo adoptadas para:

- Optimizar las rutas de recolección: A través de sensores en contenedores y análisis de datos, las rutas de los camiones recolectores pueden ser optimizadas para ahorrar tiempo y recursos.
- Monitorear en tiempo real: La capacidad de monitorizar los niveles de llenado de contenedores en tiempo real evita desbordamientos y permite una recolección más eficiente.
- Educación y sensibilización: Aplicaciones móviles y plataformas digitales están siendo utilizadas para educar a la población sobre reciclaje y reducción de residuos.

[9][18][3]

En España, el interés y la adopción de tecnologías para una gestión de residuos ha aumentado. Empresas internacionales como SENSONEO[30] han establecido alianzas con entidades locales para implementar estas soluciones en varias ciudades.

- Madrid: En la capital, la implementación de sistemas inteligentes ha permitido optimizar la recogida de residuos en una metrópolis densamente poblada, reduciendo la huella de carbono y mejorando la eficiencia.[31]
- Mollerusa: Sirve como un ejemplo de cómo las soluciones inteligentes pueden ser adaptadas a contextos más pequeños, demostrando que la gestión inteligente de residuos no está reservada únicamente para las grandes ciudades.[32]

También Ecoembes[8] está apostando por estas nuevas tecnologías, en concreto el proyecto *CircularChain*[22], desarrollado por Ecoembes y Minsait[23], busca impulsar la

transición hacia una economía circular en España mediante la implementación de tecnología *blockchain*. Esta iniciativa permite a diversas entidades como administraciones públicas compartir, controlar datos y transacciones relacionadas con la gestión de residuos de forma segura y eficiente. Además, facilita la trazabilidad no modificable de la información y la implementación de sistemas de auditoría inteligente, garantizando que se cumplen compromisos medioambientales y proporcionando un registro transparente e inalterable de todas las operaciones realizadas.

La UAB muestra un interés proactivo en adoptar estas tecnologías con proyectos en colaboración con ECIU-UTC[34]. Este proyecto está dentro de SMART-ER[35], amplifica las iniciativas preexistentes de la *ECIU University* y propone una hoja de ruta basada en talleres colaborativos y desarrollo de métodos de aprendizaje innovadores.

También otras universidades en España han comenzado a explorar la gestión inteligente de residuos. Es relevante mencionar que: La Universidad Politécnica de Valencia ha implementado desde 1996 un sistema proactivo de gestión de residuos a través de su Área de Medio Ambiente, enfocándose en la recogida selectiva y la segregación de residuos peligrosos y no peligrosos. Este sistema ha sido un reto en términos de diseño e implementación metodológica, y una oportunidad para sensibilizar a su comunidad de más de 36,000 miembros sobre prácticas ambientales responsables.[36]

2 METODOLOGÍA

Con el conocimiento adquirido gracias a mi mención en software he optado por incluir una metodología que permita flexibilidad ante cambios, que pueda ser colaborativa y que pueda generar entregas constantes a lo largo del proyecto. He optado por escoger metodologías ágiles[17], en particular *Scrum*[2] y *Kanban*[19], estas promueven adaptabilidad, entrega de resultados de forma regular e incorporación continua del *feedback* en cada etapa del proceso. También considero esencial incluir *Git*[13] como control de versiones para poder tener un historial y poder gestionar nuevas funcionalidades sin modificar el código original.

Kanban es un sistema de gestión visual que permite controlar el flujo de trabajo. En este proyecto:

- Las tareas se visualizarán en un tablero *Kanban*, donde se moverán desde *TODO*, pasando por *IN PROGRES* hasta *DONE*.
- Este enfoque facilita la identificación de cuellos de botella y permite una distribución equilibrada del trabajo.
- La transparencia del tablero *Kanban* garantiza que los stakeholders, en este caso la UAB, estén al tanto del progreso del proyecto en tiempo real.

Aunque *Kanban* proporciona una visualización muy intuitiva y control del flujo de trabajo, *Scrum* introduce una estructura cíclica basada en *sprints*[28] que es beneficiosa para la entrega periódica de resultados:

- *Sprints* Bi-semanales: El proyecto se dividirá en *sprints* de dos semanas. Al final de cada *sprint*, se entregará un producto funcional, ya sea un documento, una demostración o un prototipo.

- Reuniones de Planificación: Al inicio de cada *sprint*, el representante de la UAB se reunirá para definir los objetivos y tareas del *sprint*.
- Revisión y Retrospectiva: Al final de cada *sprint*, se presentarán los resultados a la UAB y se recogerá su *feedback*. Posteriormente, se realizará una evaluación para ver qué ha funcionado y qué se puede mejorar en el próximo *sprint*.

3 ANÁLISIS DE VIABILIDAD

3.1. Estudio de mercado

3.1.1. Segmentación del Mercado

La segmentación del mercado en este contexto podría dividirse principalmente entre instituciones educativas, entidades públicas y empresas privadas. En el caso de la UAB, nos enfocamos en una institución educativa. El mercado objetivo sería, por lo tanto, universidades que buscan implementar o mejorar sistemas de gestión de residuos mediante el uso de soluciones basadas en SmartCities.

3.1.2. Competencia

Los competidores pueden variar desde empresas tecnológicas privadas especializadas en soluciones para la gestión de residuos y startups enfocadas en sostenibilidad. Estas compañías pueden tener un bagaje empresarial extenso y pueden contar con la experiencia y red en el sector pero pueden no estar tan avanzadas tecnológicamente. Analizar nuestras fortalezas y debilidades contra estos competidores es crucial para entender cómo posicionar la propuesta de manera única y competitiva.

3.1.3. Usuarios

Los usuarios objetivo en este contexto son las instituciones públicas. Sus necesidades principales son el encontrar soluciones para la gestión de residuos, que sean viables en términos de costes. También se deben incluir soluciones que sean escalables, adaptables y que puedan integrarse con otros sistemas existentes. Lo propuesto en este proyecto debe ofrecer un sistema que sea autónomo en su funcionamiento y que proporcione datos que puedan ser utilizados para mejorar el estado actual.

3.1.4. Propuesta de Valor

La propuesta de valor diferencial se centra en una solución que vaya más allá de la gestión inteligente de residuos, introduciendo enfoques novedosos acorde a las necesidades específicas de un campus universitario. En nuestro caso se ha optado por hacer partícipes a los propios usuarios del campus con la integración, en la APP de la UAB, una funcionalidad con tecnología de reconocimiento de imágenes para ayudar a clasificar esos residuos y que mediante su uso irán generando recompensas tales como convalidación de créditos ECTS[26] y descuentos en diferentes actividades y servicios en el campus. Además de implementar estas tecnologías, la solución también se enfoca en la adaptación de los contenedores de residuos existentes.

3.2. Análisis Técnico

3.2.1. Stack tecnológico

Para generar un ecosistema sólido se ha optado por utilizar tecnologías muy versátiles. Podemos diferenciar en Hardware y Software:

Para el prototipo Hardware: Chip TTGO LoRa ESP32 Wifi+Bluetooth: cerebro y controlador que gestionará la lógica de los sensores con una antena LoRa integrada que permitirá intercomunicaciones entre el servidor y otras antenas con una frecuencia de 868 MHz. Sensor LiDar VL53L0XV2: Sensor para medir cómo de lleno puede estar un contenedor, gracias a sus mediciones láser. Sensor Giroscópico MPU-6050: sensor para comprobar cuando se ha descargado el contenido del contenedor a un camión. Sensor humedad DHT-11: Registra mediciones de temperatura y humedad, que pueden servir para controlar el ciclo de compostaje en el orgánico. Fritzing[12]: programa libre de automatización de diseño electrónico. ArduinoIDE[16]: para programar la lógica del Chip TTGO LoRa ESP32 Wifi+Bluetooth. Para Sofware: Visual Studio Code[20]: IDE para programar el *frontend* y *backend*. Figma[10]: Página para poder crear los diseños de *frontend*. MERN[25]: Stack tecnológico que incluye; MongoDB, como base de datos; Express, entorno de trabajo para aplicaciones web; React, librería Javascript de código abierto diseñada para crear interfaces de usuario facilitando el desarrollo de aplicaciones en una sola página; Node.js, entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor basado en el lenguaje de programación JavaScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos. GraphQL[11]: GraphQL es un lenguaje de consulta y manipulación de datos para APIs. ApolloServer[1]: Apollo Server proporciona características avanzadas, como integración con Apollo Client y manejo eficiente de datos. GraphQL Yoga: Una biblioteca basada en Express que simplifica la creación de servidores GraphQL. GitHub[21]: control de versiones.

3.2.2. Ubicación

La implementación inicial se realizará en el campus de la UAB. Al utilizar el campus como una SmartCity, se puede generar un entorno controlado y dinámico antes de considerar una posible expansión.

3.3. Análisis de Riesgos

Aquí se presentan un seguido de riesgos que tiene este proyecto y con los que se deberá convivir e intentar paliar aplicando metodologías que permitan mitigarlos o hasta eliminarlos, al final del apartado se hace un pequeño plan de contingencia capaz de evitar estos riesgos.

3.3.1. Identificación de riesgos

- R-02 Fallo de Hardware: Causa Potencial: Deterioro de dispositivos, sobrecalentamiento, o daños accidentales. Impacto: Interrupción de las operaciones y posibles retrasos en la ejecución del proyecto.
- R-03 Problemas de Software Causa Potencial: Bugs, incompatibilidad entre versiones, o fallos de sistema.

Impacto: Compromiso de la calidad del proyecto y retrasos debidos a la necesidad de solucionar problemas imprevistos.

- R-06 Errores de Conectividad en la Red LoRaWAN Causa Potencial: Fallos en la comunicación entre los nodos y el servidor central. Impacto: Pérdida de datos y fallas en la operación del sistema.
- R-11 Añadir Nuevas Funcionalidades Causa Potencial: Descubrimiento de necesidades adicionales tras pruebas iniciales o *feedback* de *stakeholders*. Impacto: Incremento en los recursos y tiempo necesario para implementar nuevas funciones.
- R-13 Eliminación de Funcionalidades Planeadas Causa Potencial: Observar redundancias o inutilidades durante las fases de testeo. Impacto: Posible desequilibrio en otras funcionalidades y alteración del flujo de trabajo del sistema.

[Ver todo el listado de riesgos aquíA.3](#)

3.3.2. Mitigación de riesgos

- M-01 Control de Versiones Implementar un sistema de control de versiones, como Git, que permita gestionar los diferentes estados del código fuente y poder revertir a versiones anteriores si es necesario.
- M-02 *Backups* Regulares Establecer una rutina de copias de seguridad automáticas en múltiples ubicaciones (local, cloud, unidades externas) para garantizar que los datos del proyecto están seguros y actualizados.
- M-06 Monitoreo Continuo Implementar herramientas de monitoreo en la red LoRaWAN para identificar y corregir proactivamente problemas de conectividad o transmisión.
- M-08 Pruebas de Seguridad Implementar pruebas de seguridad para identificar y corregir posibles vulnerabilidades antes de que la aplicación se lance al público.
- M-13 Reuniones Regulares de Revisión Establecer encuentros periódicos con los clientes para revisar el progreso y discutir posibles ajustes.

[Ver todo el listado de mitigaciones de riesgos aquíA.4](#)

3.3.3. Plan de contingencia

Documentación Dinámica: Asegurarse de que la documentación del proyecto sea fácil de ajustar y modificar para acomodar cambios sin demandar una sobrecarga de trabajo.

Análisis de Impacto del Cambio: Antes de implementar cambios en los requisitos, realizar un análisis de impacto que permita entender las implicaciones en tiempo, costos y recursos.

Control de Versiones: Implementar un sistema de control de versiones para poder rastrear y revertir cambios si resulta necesario, o si los nuevos requisitos introducen nuevos problemas.

Proceso de Recuperación de Datos: Establecer un protocolo claro para recuperar datos desde las copias de seguridad en caso de pérdida de datos, asegurando que los miembros del equipo sepan cómo actuar y minimizar el tiempo de inactividad.

Mecanismos de Reporte de Incidentes: Implementar una herramienta o plataforma que permita a los miembros del equipo informar de manera eficiente sobre incidentes o fallos técnicos para una rápida intervención. Tener un equipo designado para abordar problemas críticos identificados durante la fase de testing para realizar correcciones rápidas.

Reevaluación del Calendario: En caso de incidentes que afecten los plazos, revisar el cronograma del proyecto y ajustar las fechas de entrega, redistribuir recursos si es necesario y comunicar los cambios a todas las partes interesadas.

Estrategia de Escalabilidad: Tener recursos adicionales que puedan ser asignados para abordar problemas críticos inesperados en la implementación sin afectar otras áreas del proyecto.

Comunicación con los Stakeholders: Mantener líneas de comunicación abiertas y transparentes con los stakeholders, informándoles sobre cualquier desafío y las estrategias para superarlos.

Recopilación de Información: Reunir nuevamente la información desde fuentes originales o alternativas en caso de que la recreación de documentos sea necesaria.

Validación de Datos: Asegurarse de que la información recuperada o recreada sea validada para asegurar su precisión y coherencia con el proyecto.

4 REQUISITOS

En este apartado se listan los Requisitos funcionales y no funcionales necesarios para poder generar un producto completo y satisfactorio. Para clasificar los distintos tipos de requisitos se priorizan de la siguiente forma: Alta, Media y Baja.

4.1. Requisitos funcionales

- RF-01: El sistema debe recopilar datos en tiempo real sobre el control de residuos en todo el campus. Alta
- RF-08 La aplicación debe ofrecer información personalizada sobre la gestión de residuos y consejos para el reciclaje . Baja
- RF-09 Utilizar tecnología LoRa para la comunicación entre los contenedores y el sistema central. Alta
- RF-14 Los usuarios registrados deben poder iniciar sesión utilizando su dirección de correo electrónico y contraseña. Alta
- RF-21 La aplicación debe contar con una interfaz de usuario intuitiva que sea fácil de navegar y utilizar. Alta
- RF-25 Utilizar algoritmos de aprendizaje automático para planificar rutas de recolección de residuos de manera dinámica, teniendo en cuenta la información en tiempo real sobre el nivel de llenado de los contenedores. Media

[Ver todo el listado de requisitos funcionales aquíA.1](#)

4.2. Requisitos no funcionales

- RNF-01 El sistema debe ser eficiente en la gestión de datos y recursos para minimizar el consumo de energía y recursos. Alta
- RNF-03 Garantizar la seguridad de los datos recopilados y la privacidad de los usuarios. Alta
- RNF-04 El sistema debe estar disponible las 24 horas del día para garantizar una gestión continua de residuos. Alta
- RNF-09 Garantizar tiempos de respuesta rápidos para alertas de contenedores llenos y comunicaciones con usuarios. Alta

[Ver todo el listado de requisitos no funcionales aquíA.2](#)

5 PLANIFICACIÓN

La siguiente planificación esquematiza las fases con sus actividades y correspondientes entregables y el calendario con información de las fechas, de forma que se establecen las actividades, duraciones y fechas previstas, ofreciendo una hoja de ruta detallada del proyecto. Además, se especifica cómo se llevarán a cabo estas fases mediante metodologías *agile*.

5.1. Fases

5.1.1. Fase Inicial

- Calendario: 18/09/2023 a 27/10/2023 Duración: 55,2h Actividades: Documentación inicial (7h), Análisis (9h), Diseño(24h) y Pruebas LoRaWAN(15h)
- Milestone: Reunión de seguimiento: Día: 08/10/2023 Duración: 0,2h
- Entregables: Documentación inicial, Documentación diseño.

5.1.2. Fase de Programación servidor

- Calendario: 30/10/2023 a 01/12/2023 Duración: 57,2h Actividades: Lógica(24h), Cloud(12h) y Pruebas de Conexión(21h)
- Milestone: Reunión de seguimiento: Día: 12/11/2023 Duración: 0,2h
- Entregables: Documentos de código de *backend* y Documentos de resultados de pruebas.

5.1.3. Fase de Implementación

- Calendario: 04/12/2023 a 26/01/2024 Duración: 152,4h Actividades: LoRaWAN(22h), APP Smartphone(130h): (Diseño APP(30h), Desarrollo vista usuario APP(50h), Desarrollo vista usuario profesional APP(50h))
- Milestone: Reunión de seguimiento: Día: 17/12/2023 Duración: 0,2h

- Milestone: Reunión de seguimiento: Día: 21/01/2024 Duración: 0,2h
- Entregables: Documentos de diseño red, Documentos de pruebas de los casos de uso, Documentos de diseño de APP y Documentos de código *frontend*

5.1.4. Fase de Testing

- Calendario: 29/01/2024 a 04/02/2024 Duración: 25,2h
Actividades: Pruebas de caja negra(6h), Pruebas de caja blanca(7h), Pruebas interconexión sensores y servidor(6h) y Pruebas interconexión servidor y APP(6h).
- Milestone: Reunión de seguimiento: Día: 04/02/2024 Duración: 0,2h
- Entregables: Documentos de pruebas del código.

5.1.5. Fase de Documentación

- Calendario: 05/02/2024 a 06/02/2024 Duración: 10h
Actividades: Redactar memoria final(10h).
- Entregables: Documento final

5.2. Aplicación de metodologías *agile*

Para poder llevar un buen ritmo de entregables se ha escogido hacer *sprints* usando metodología SCRUM con una duración quincenal. Estos *sprints* tendrán un ciclo que constará primeramente en asignar tareas del *backlog* para centrarse en un tema en específico, después diseñar, programar, implementar y por último testear, en cada ciclo. Para llevar a cabo la metodología *Kanban* hay un diagrama preparado con todas las tareas en el *backlog* y, para poder enlazarlo con scrum, las tareas se irán asignando en cada *sprint* al estado TODO, conforme se vayan escogiendo se pasaran a IN PROGRESS hasta finalizar en el estado DONE, el *sprint* debe acabar cuando todas las tareas escogidas al inicio del *sprint* hayan finalizado y estén en estado DONE, si queda alguna tarea por hacer se pasaría al siguiente *sprint*. Para este proyecto se ha considerado hacer 7 *sprints*, de forma que ocupen toda la longitud del proyecto, teniendo en cuenta una longitud cada uno de 2 semanas exceptuando el *sprint*(6) que conforma la creación de la APP que se ha decidido ampliarlo a 1 mes.

- *Sprint 1:* Conforma la fase inicial del proyecto incluyendo actividades de la fase inicial como la documentación, análisis y, del apartado de diseño, el diseño inicial únicamente. Constará con un entregable de documentación inicial y de diseño inicial.
- *Sprint 2:* En este sprint se realizará la parte de diseño de la BBDD y del código más un seguido de pruebas con material de Arduino tal como sensores y en específico sensores con antenas LoRa para familiarizarse con el entorno, de esta forma se finaliza con la fase inicial del proyecto. Constará con un entregable de los diseños en uml más un documento con las pruebas realizadas y sus resultados.

■ *Sprint 3:* Aquí se comienza la fase de programación del servidor donde se realizaran las actividades de la lógica del código, como la implementación de la base de datos y el desarrollo de algoritmos de clasificación. Constará con entregables del código del algoritmo de clasificación y de la base de datos.

- *Sprint 4:* Continuará después del *sprint 3*, acabando de conformar la fase de programación del servidor realizando las actividades de Cloud, que constará de configurar el cloud e implementar el servidor, y pruebas de conexión que asegurarán la correcta comunicación entre el servidor y los sensores LoRa para poder recopilar las primeras informaciones y poder hacer la simulación de uso, finalizando así la fase de programación del servidor. Constará con entregables de documentación de configuración y de resultados de las pruebas y simulación.
- *Sprint 5:* Despues de completar los primeros 4 *sprints*, en este se empezará con la fase de implementación, que abarca todo lo relacionado con LoRaWAN, diseño de la red, implementación y prueba de red, casos de uso e interconexión de sensores LoRa y además poder generar las primeras peticiones e información funcionales. Constará con entregables de documentación de diseño y prototipo del primer dispositivo funcional.
- *Sprint 6:* Le tomará el relevo al anterior sprint realizando la parte del desarrollo de la APP consistiendo en el diseño y posterior implementación del código *frontend*. Constará de un entregable que será la aplicación.
- *Sprint 7:* Este *sprint* finalizará el proyecto haciendo hincapié en todo el testing e integración de la aplicación con el servidor y del servidor con los sensores. También se finalizará la documentación Constará con dos entregables, el informe de los resultados de los test y el dossier del proyecto finalizado.

6 ARQUITECTURA

La arquitectura de este proyecto se basa en dos grandes grupos, la arquitectura de hardware y la de software. Cada una se expondrá en detalle a continuacion. Para ver el diseño completo Ver **Figura 8**

6.1. Arquitectura Hardware

En esta sección se presenta la arquitectura de hardware que se ha creado para realizar este proyecto.

- **BlueCampusDevice:** se trata del dispositivo que se encargará de recibir las mediciones de los sensores para enviarlos por LoRa y está compuesto por:

- Un chip ESP32 integrado con una antena LoRa para enviar los datos.
- Un giroscopio para detectar el angulo del contenedor para cuando lo vayan a recoger.
- Un sensor Lidar que se encargará de medir la distancia a los residuos para determinar como de lleno está el contenedor (menor distancia significa que el contenedor esta más lleno).

- Un sensor de humedad y temperatura para realizar dichas mediciones.
- Una batería con placa solar para la alimentación.

Ver Figura 1 Ver Prototipo A.12

- Un servidor con disponibilidad de 24h.
- Una Base de datos no relacional para guardar todas las mediciones.

6.2. Arquitectura Software

En esta sección se presenta la arquitectura de software que se ha creado para realizar este proyecto. Esta arquitectura se basa en el modelo **cliente-servidor**, la App y la Web harán peticiones asíncronas al servidor que funcionará como un modelo de **API** y la intercomunicación será mediante el **protocolo http**. La comunicación con la base de datos será mediante consultas GraphQL[11], el servidor está implementado con ApolloServer[1]. Para concretar, el servidor se alojará en una **máquina virtual** dentro del CVC que permitirá que este activo y accesible las **24h**, también se alojará el host de la página web. Para el uso de mapas y de autenticación se usarán APIs de Google Cloud tales como **Google Maps** y **Firebase**. Ver Figura 2



Fig. 1: Arquitectura BlueCampusDevice

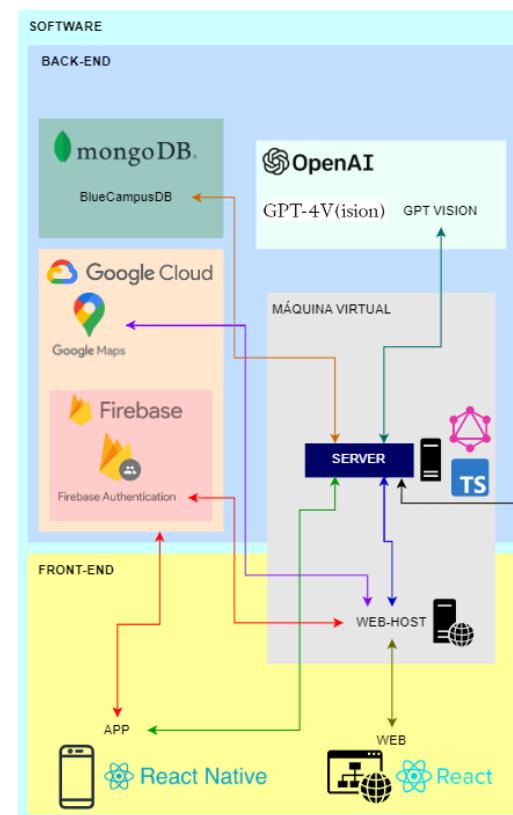


Fig. 2: Arquitectura del Software

7 DISEÑO

7.1. Diseño de BlueCampusDevice

El diseño consta de dos partes: la primera se colocará dentro del contenedor en la parte superior del mismo, con los sensores mirando hacia abajo, y con la batería. La segunda parte estará en el exterior y contará de un módulo con placa solar y la antena LoRa, para captar la luz solar y tener buena cobertura respectivamente. Ver diseño A.5

7.2. Diseño de la Red

Este apartado detalla el diseño de la red para monitorear los contenedores de reciclaje utilizando dispositivos denominados "BlueCampusDevice". Cada contenedor está equipado con un dispositivo para medir diversos parámetros ambientales y de capacidad.

7.2.1. Componentes del sistema

- BlueCampusDevice: Dispositivo integrado en cada contenedor de reciclaje. Equipado con sensores para medir temperatura, humedad, nivel de llenado y movimiento (giroscopio).
- Contenedores de Reciclaje: Cinco tipos según el material a reciclar - azul, amarillo, verde, marrón y gris.
- Zonas: Cada zona comprende un set de cinco contenedores de reciclaje, cada uno con un BlueCampusDevice.
- Comunicación LoRa: Los dispositivos envían datos a través de la red LoRa.

- Gateways LoRa: Dos gateways reciben los datos de los dispositivos y los transmiten al servidor.
- Servidor - ApolloServer: Procesa y almacena los datos recibidos.
- Base de Datos: MongoDB, que utiliza GraphQL para la gestión de datos.
- Clientes:
 - Cliente Web: Interactúa con el servidor mediante HTTP y Apollo Client.
 - Cliente App: Aplicación móvil con funcionalidades similares al cliente web.

7.2.2. Diseño Red LoRaWAN

Los dispositivos LoRa se comunican haciendo uso de un diagrama en doble estrella, cada gateway puede recibir indistintamente los datos de cada BlueCampusDevice de forma única. Ver **Figura 3**

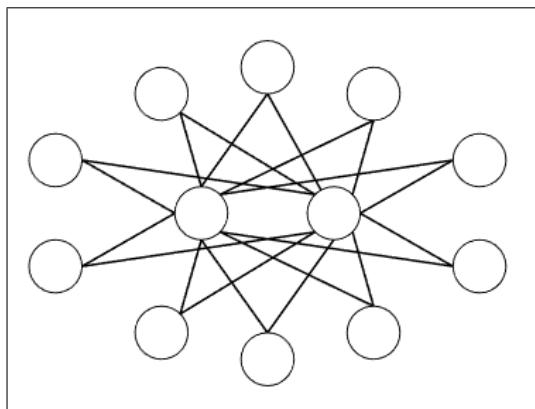


Fig. 3: Diagrama doble estrella

7.2.3. Flujo de datos

- Recolección de Datos: Cada BlueCampusDevice recopila datos de su contenedor asignado.
- Transmisión de Datos: Los dispositivos envían los datos recopilados a través de la red LoRa.
- Recepción de Datos en Gateways: Los gateways de LoRa reciben los datos y los envían al servidor ApolloServer.
- Procesamiento y Almacenamiento de Datos: ApolloServer procesa los datos y los almacena en MongoDB.
- Acceso a Datos por Clientes: Los clientes web y de app hacen peticiones HTTP al servidor para acceder a los datos.

7.2.4. Diagrama de la red

Ver **diagrama 9**

8 CASOS DE USO

8.1. Caso: Usuario Gestor

Este tipo de usuario se encargará de revisar la medición de los sensores y monitorizar que todo vaya de forma correcta. **Objetivo:** Consultar las mediciones de los contenedores en el campus. **Flujo Principal:** El usuario abre la aplicación web. Selecciona la opción de iniciar sesión con credenciales o registrarse. Se abre la página principal con los datos en un dashboard. El usuario selecciona, busca un contenedor o zona de interés. La aplicación muestra la información correspondiente con la búsqueda. Ver **diagrama 6**

8.2. Caso: Usuario Campus

Este tipo de usuario utilizará el sistema para reciclar y obtener recompensas. **Objetivo:** Escanear un residuo para determinar su tipo y recibir información sobre su manejo adecuado. **Flujo Principal:** El usuario abre la aplicación móvil. Selecciona la opción de iniciar sesión con credenciales o registrarse. Selecciona la opción de escanear residuo. Utiliza la cámara de su dispositivo para escanear el código o imagen del residuo. La aplicación utiliza la IA para identificar el tipo de residuo. Muestra al usuario información sobre cómo reciclar o desechar adecuadamente el residuo y su recompensa. Ver **diagrama 7**

9 RESULTADOS

El proyecto se centró en crear un sistema para gestionar residuos en el campus universitario, promoviendo el reciclaje entre la comunidad. Se ha combinado tecnología con prácticas de gestión para lograr un entorno más sostenible. Se implementaron contenedores inteligentes equipados con sensores y tecnología LoRaWAN, permitiendo monitorear en tiempo real el llenado y tipo de residuos, facilitando así una recolección más eficiente y gestionar desde una página web. Además, se desarrolló una aplicación móvil con tecnología de escaneo de imágenes y AI, diseñada para ayudar a los usuarios a clasificar correctamente los residuos y motivar el reciclaje mediante incentivos.

Resultados teóricos de la Implementación de Contenedores Inteligentes Existen 22 zonas, con 5 contenedores por zona, en el campus con capacidad para instalar los sensores en puntos estratégicos del campus. Los contenedores inteligentes son capaces de recopilar cada uno 1440 registros al día (1 registro cada minuto). Estos registros pueden indicar que, residuos se depositan, cuándo y dónde es necesario un mayor número de recogidas. Además, los contenedores ubicados cerca de áreas de alta actividad, como comedores y zonas de estudio, se suelen llenar más rápidamente y requieren recolecciones más frecuentes. La capacidad de recibir información en tiempo real sobre el estado de los contenedores permite que las rutas de recolección se planifiquen con datos concretos sobre el nivel de llenado de los contenedores, y se pueden adaptar las rutas de recolección, lo que llevaría a una reducción significativa en el número de recolecciones innecesarias y, por ende, en los costos operativos.

Optimización de Rutas de Recolección Las rutas de recolección se basan en horarios fijos, lo que resulta en visitas innecesarias a contenedores que pueden estar solo parcialmente llenos, mientras que otros contenedores, más llenos, pueden no ser atendidos con la frecuencia necesaria. Con los datos recopilados, se pueden ajustar las rutas de recolección, priorizando los contenedores que se llenan más rápidamente. Esta optimización no solo mejoraría la eficiencia operativa, sino que también ha tenido un impacto positivo desde el punto de vista ambiental y económico. La reducción en el número de viajes de recolección llevaría a una disminución en el consumo de combustible de los vehículos de recolección, lo que a su vez reduciría las emisiones de CO₂. **Identificación de Áreas Prioritarias para la Reducción de Residuos** A través del análisis de los datos recopilados por los contenedores inteligentes, se pueden identificar qué zonas son las de mayor generación de residuos y de qué tipo son. En respuesta a esto, se pueden implementar campañas de concienciación sobre el uso de productos.

Desarrollo y Uso de la Aplicación Móvil La aplicación móvil diseñada para el proyecto incluye funcionalidades como el escaneo de objetos para identificar su correcta clasificación, un sistema de recompensas por buenas prácticas de reciclaje, y mapas interactivos que muestran la ubicación de los contenedores inteligentes más cercanos.

Adopción y Uso La aplicación idealmente está prevista para incorporar a la aplicación de la UAB para acceder a la mayor parte de la comunidad universitaria.

Desafíos y Limitaciones Uno de los principales desafíos puede ser la no utilización de esta funcionalidad por parte de algunos usuarios del campus y mantener la conectividad de los contenedores inteligentes. El sistema puede enfrentar limitaciones en términos de cobertura completa del campus y en la precisión del escaneo de algunos tipos de residuos en la aplicación móvil. El proyecto ha cumplido y superado varias de sus metas iniciales, demostrando el potencial de la tecnología avanzada en la gestión ambiental.

Recomendaciones para Futuras Investigaciones o Implementaciones Se recomendaría expandir la cobertura de contenedores inteligentes y mejorar las funcionalidades de la aplicación móvil. También sería beneficioso explorar asociaciones con organizaciones externas para inyectar financiación.

10 PRESUPUESTO

10.1. Financiación

En este apartado se desglosará el precio y las horas por cada fase para extraer un presupuesto inicial que podrá ir variando a lo largo del proyecto.

10.1.1. Presupuesto inicial

■ Fase Inicial:

- Planificaciones y analista: 16h * 10€/h = 16 €
- Diseñador: 24h * 12€/h = 288€
- Tester: 15h * 15€/h = 225€

Subtotal: 673€

■ Fase de Programación servidor:

- Programador *Backend*: 36h * 17€/h = 612€
- Experto en redes: 21h * 15€/h = 315€

Subtotal: 927€

■ Fase de Implementación:

- Diseñador y experto en redes: 22h * 15€/h = 330€
- Diseñador: 30h * 12h = 360€
- Programador *Frontend*: 110h * 18€/h = 1.980€

Subtotal: 2.670€

■ Fase de Testing:

- Tester: 25h * 15€/h = 375€

Subtotal: 375€

■ Fase de Documentación:

- Documentación: 10h * 10€/h = 100€

Subtotal: 100€

■ Material: Número de zonas = 22, número de contenedores = 5

- TTGO LORA ESP32 con OLED: (Número de zonas) * (número de contenedores) * 19,95 € = 2.194,50€
- Sensor LiDAR VL53L0X: (Número de zonas) * (número de contenedores) * 4,75€ = 522,50€
- Sensor Giroscopio MPU-6050: (Número de zonas) * (número de contenedores) * 2,45€ = 269,50€
- Sensor humedad DHT-11: (Número de zonas) * (número de contenedores) * 2,80€ = 308,00€

Subtotal: 3.294,50€(instalación de todos los prototipos en el campus) (Unidad prototipo: 29,95€)

Total: 8.039,50€

11 CONCLUSIONES

El proyecto ha sido un éxito en cuanto a cumplimiento de objetivos tanto generales como específicos, al implementar un sistema desde cero que permita la gestión inteligente de contenedores en todo el campus de la UAB mediante los sensores BlueCampus. La adopción de la tecnología LoRaWAN, combinada con la integración de sensores IoT, ha permitido una recolección de datos precisa y en tiempo real, crucial para poder monitorizar y gestionar rutas de recolección.

El rediseño inteligente de los contenedores ha facilitado la segregación y recolección de residuos junto con la nueva funcionalidad a añadir en la aplicación de la UAB que, mediante inteligencia artificial, es capaz de ayudar a separar cada residuo en su respectivo contenedor. La optimización de rutas contribuirá a reducir costes y mejorar la eficiencia, que con ayuda de la implementación de sistemas de monitorización en tiempo real ha permitido un seguimiento detallado del flujo de residuos, proporcionando datos valiosos para la mejora continua del sistema.

Además, esta integración es coherente con otros sistemas de SmartCity porque refleja un enfoque integral y visionario, haciendo que el proyecto llegue a ser sostenible, y con el potencial para expandirse y adaptarse a futuras necesidades y ubicaciones.

AGRADECIMIENTOS

En primer lugar, quiero expresar mi más sincero agradecimiento a mi tutor, Fernando, cuya guía y apoyo me han permitido crear este proyecto. Sus consejos y contribuciones me han ayudado a esquematizar, aprender y generar un proyecto con futuro, y que también me han abierto las puertas al Centro de Visión por Computador (CVC) de la UAB. En el CVC, he podido aprender con charlas la vanguardia de las nuevas tecnologías y proyectos con profesionales y expertos. vuelvo a agradecer la oportunidad de tutorizarme y de creer en este proyecto.

Asimismo, me siento profundamente agradecido con mi familia, que ha sido mi pilar y apoyo. Un agradecimiento especial a mis padres y hermano, cuya paciencia y ánimo han sido incondicionales. A mis abuelos, gracias por creer en mí y por estar hasta las tantas esperandome para comer. Su cariño y ánimo han sido fundamentales a lo largo del proyecto.

No puedo dejar de mencionar a mis amigos y amigas cercanos, cuyo soporte ha sido un refugio y motivación por su confianza en mis capacidades y los ratos de despejarnos.

A todos, mi más profundo agradecimiento.

REFERENCIAS

- [1] Apollo. *Apollo Server is an open-source, specification-compliant GraphQL server that's compatible with any GraphQL client, including Apollo Client. It's the best way to build a production-ready, self-documenting GraphQL API that can use data from any source.* Abr. de 2020. URL: <https://www.apollographql.com/docs/apollo-server/>.
- [2] Atlassian. *¿Qué es scrum?* Mayo de 2022. URL: <https://www.atlassian.com/es/agile/scrum>.
- [3] Marcello Basani. *Tecnologías inteligentes de residuos sólidos: ¿Dónde estamos y hacia dónde vamos?* Mayo de 2023. URL: <https://blogs.iadb.org/agua/es/tecnologias-inteligentes-de-residuos-solidos-donde-estamos-y-hacia-donde-vamos/>.
- [4] Oracle Corporation. *¿Qué es big data?* Mayo de 2021. URL: <https://www.oracle.com/es/big-data/what-is-big-data/>.
- [5] Oracle Corporation. *¿Qué es el IoT?* Feb. de 2014. URL: <https://www.oracle.com/es/internet-of-things/what-is-iot/>.
- [6] Oracle Corporation. *¿Qué es la IA? Conoce la inteligencia artificial.* Abr. de 2014. URL: <https://www.oracle.com/es/artificial-intelligence/what-is-ai/>.
- [7] Becolve Digital. *Qué es LoRaWAN.* Feb. de 2020. URL: <https://becolve.com/blog/que-es-lorawan/>.
- [8] ecoembes. *Organización sin ánimo de lucro que cuida del medioambiente a través del reciclaje y el eco-diseño de los envases domésticos ligeros en España.* Sep. de 2023. URL: <https://www.ecoembes.com/es>.
- [9] eSmasrCITY. *Gestión Inteligente de Residuos – Soluciones para las Smart Cities.* Oct. de 2019. URL: <https://www.esmartercity.es/comunicaciones/comunicacion-gestion-inteligente-residuos-soluciones-smart-cities>.
- [10] Figma. *UI design tool.* Sep. de 2016. URL: <https://www.figma.com/>.
- [11] The GraphQL Foundation. *GraphQL is a query language for APIs and a runtime for fulfilling those queries with your existing data. GraphQL provides a complete and understandable description of the data in your API, gives clients the power to ask for exactly what they need and nothing more, makes it easier to evolve APIs over time, and enables powerful developer tools.* Nov. de 2017. URL: <https://graphql.org/>.
- [12] fritzing. *Fritzing is an open-source hardware initiative that makes electronics accessible as a creative material for anyone. We offer a software tool, a community website and services in the spirit of Processing and Arduino.* Feb. de 2024. URL: <https://fritzing.org/>.
- [13] Git. *Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.* Ago. de 2023. URL: <https://git-scm.com/>.
- [14] IBM. *¿Qué es la computación en la nube?* Jun. de 2020. URL: <https://www.ibm.com/es-es/topics/cloud-computing>.
- [15] IBM. *¿Qué es la tecnología blockchain?* Mayo de 2019. URL: <https://www.ibm.com/es-es/topics/blockchain>.
- [16] Karl Söderby y Jacob Hylén. *Getting Started with Arduino IDE 2.* Sep. de 2022. URL: <https://docs.arduino.cc/software/ide-v2/tutorials/getting-started-ide-v2>.
- [17] Salesforce LATAM. *Metodologías Ágiles: qué son y cómo pueden ayudarte.* Dic. de 2021. URL: <https://www.salesforce.com/mx/blog/que-son-metodologias-agiles/>.
- [18] INMA LIDÓN. *Robots de la basura, contenedores inteligentes y big data: así se gestionarán en el futuro los residuos de las ciudades.* Dic. de 2021. URL: <https://www.elmundo.es/ciencia-y-salud/medio-ambiente/2021/04/21/607d98e7fd9ff9288b45c3.html>.
- [19] Julia Martins. <https://asana.com/es/resources/what-is-kanban>. Oct. de 2022. URL: <https://asana.com/es/resources/what-is-kanban>.

- [20] Microsoft. *Getting Started Visual Studio Code*. Nov. de 2021. URL: <https://code.visualstudio.com/docs>.
- [21] Microsoft. *GitHub es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git*. Feb. de 2008. URL: <https://github.com/>.
- [22] Minsait. *Ecoembes y Minsait despliegan una red blockchain para aumentar la transparencia e incentivar la colaboración en la economía circular*. Abr. de 2021. URL: <https://www.minsait.com/es/actualidad/media-room/ecoembes-y-minsait-despliegan-una-red-blockchain-para-aumentar-la>.
- [23] Minsait. *Ecoembes y Minsait despliegan una red blockchain para aumentar la transparencia e incentivar la colaboración en la economía circular*. Abr. de 2021. URL: <https://www.minsait.com>.
- [24] UAB OpenLabs. *Qué es un Living Lab*. Dic. de 2020. URL: <https://www.uab.cat/web/conoce-labs/que-es-un-living-lab-1345823359027.html>.
- [25] openwebinars. *MERN es un conjunto de marcos/tecnologías utilizados para el desarrollo web de aplicaciones que consta de MongoDB, React JS, Express JS y Node JS*. Abr. de 2023. URL: <https://openwebinars.net/blog/mern-stack-que-es-y-que-ventajas-ofrece/>.
- [26] Ministerio de educación y formación profesional. *Créditos ECTS*. Jul. de 2010. URL: <https://www.educacionyfp.gob.es/italia/dam/jcr:b53864d2-65a3-4526-abf4-61ef02f5be34/el-sistema-universitario-espa-ol2.pdf>.
- [27] REGUSA RECYCLING & RECOVERY. *RETOS A LOS QUE SE ENFRENTAN LAS CIUDADES INTELIGENTES*. Dic. de 2021. URL: <https://regusa.es/la-gestion-inteligente-de-residuos-en-una-smart-city/>.
- [28] Max Rehkopf. *¿Qué son los sprints?* Nov. de 2021. URL: <https://www.atlassian.com/es/agile/scrum/sprints>.
- [29] Repsol. *¿Qué son las Smart Cities? ¿En qué consisten y cuáles son?* Sep. de 2023. URL: <https://www.repsol.com/es/energia-futuro/tecnologia-innovacion/smart-cities/index.cshtml>.
- [30] Sensoneo. *Empresa líder mundial en soluciones de residuos inteligentes*. Sep. de 2023. URL: <https://sensoneo.com/es/>.
- [31] Sensoneo. *Madrid contará con la mayor instalación de gestión inteligente de residuos de Europa*. Feb. de 2022. URL: <https://sensoneo.com/es/madrid-smart-waste-instalacion/>.
- [32] Sensoneo. *Reference: Ultrasonic bin sensors monitoring waste in Mollerussa, Spain*. Jul. de 2023. URL: <https://sensoneo.com/reference/monitoring-waste-mollerussa-spain/>.
- [33] Tokio. *Sensores IoT: qué son, para qué sirven, tipos, características y más*. Dic. de 2020. URL: <https://www.tokioschool.com/noticias/sensores-iot/>.
- [34] UAB. *ECIU-UTC En qué consisteix?* Dic. de 2021. URL: [https://www.uab.cat/web/projecte-smart-er-projectes/eciu-utc-1345878460411.html](https://www.uab.cat/web/projecte-smart-er/projectes/eciu-utc-1345878460411.html).
- [35] UAB. *Proyecto SMART-ER*. Mar. de 2020. URL: <https://www.uab.cat/web/proyecto-smart-er-1345836766878.html>.
- [36] Universitat Politècnica de València. *Sistema de Gestión Ambiental de la Universitat Politècnica de València*. Inf. tèc. Universitat Politècnica de València, 2017. URL: <https://riunet.upv.es/bitstream/handle/10251/97682/UPV.MA-INF.RSGA.2017-UPV.pdf?sequence=1>.

APÉNDICE

A.1. Requisitos funcionales

- RF-01: El sistema debe recopilar datos en tiempo real sobre el control de residuos en todo el campus. Alta
- RF-02 El sistema debe asignar un identificador único a cada contenedor para rastrear su ubicación y recopilar datos específicos de cada uno de ellos. Alta
- RF-03 El sistema debe ser capaz de utilizar sensores para evaluar la calidad de los residuos en los contenedores. Media
- RF-04 El sistema debe ser capaz de utilizar sensores para medir el nivel de llenado en los contenedores. Alta
- RF-05 Los datos recopilados deben ser almacenados en una base de datos histórica para permitir un análisis a largo plazo de los patrones de generación de residuos. Media
- RF-06 El sistema debe ser capaz de detectar cuando un contenedor alcanza un nivel de llenado predeterminado y generar alertas para la recolección. Alta
- RF-07 Debe existir una aplicación móvil para usuarios que permita registrar su contribución al reciclaje y recibir recompensas, incluyendo créditos ECTS. Alta
- RF-08 La aplicación debe ofrecer información personalizada sobre la gestión de residuos y consejos para el reciclaje . Baja
- RF-09 Utilizar tecnología LoRa para la comunicación entre los contenedores y el sistema central. Alta
- RF-10 Monitorizar el estado de los contenedores en tiempo real, incluyendo ubicación y nivel de llenado. Alta
- RF-11 El sistema debe permitir la diferenciación entre los profesionales encargados de la gestión de residuos y los usuarios que reciclan. Alta
- RF-12 Mejorar la gestión de la recogida de residuos mediante el uso de aprendizaje automático para optimizar las rutas de los camiones de recogida. Alta
- RF-13 Los usuarios deben tener la capacidad de registrarse en la aplicación móvil proporcionando información personal como nombre, correo electrónico y contraseña. Alta
- RF-14 Los usuarios registrados deben poder iniciar sesión utilizando su dirección de correo electrónico y contraseña. Alta
- RF-15 Permitir que los usuarios inicien y cierren sesión de manera segura y que sus sesiones se mantengan activas. Alta
- RF-16 Establecer políticas de seguridad de contraseñas que incluyan requisitos de longitud mínima y complejidad. Alta

- RF-17 El sistema debe permitir la configuración de niveles de llenado predeterminados para diferentes tipos de contenedores. Media
- RF-18 Debe ser posible configurar alertas personalizadas para notificar a los profesionales encargados de la recolección cuando un contenedor alcance un nivel de llenado específico. Media
- RF-19 La aplicación móvil debe permitir a los usuarios registrar sus actividades de reciclaje. Media
- RF-20 El sistema debe calcular recompensas de manera personalizada para cada usuario en función de su contribución al reciclaje y sus hábitos de reciclaje. Alta
- RF-21 La aplicación debe contar con una interfaz de usuario intuitiva que sea fácil de navegar y utilizar. Alta
- RF-22 El sistema debe estar diseñado para integrarse de manera efectiva con la plataforma LoRa para garantizar una comunicación confiable y segura entre los contenedores y el sistema central. Alta
- RF-23 Implementar algoritmos de inteligencia artificial que puedan analizar los datos de los contenedores y usuarios para identificar patrones y tendencias útiles para la gestión de residuos. Media
- RF-24 Asegurarse de que el *backend* sea escalable para manejar un crecimiento en la cantidad de datos a medida que se expande el sistema. Media
- RF-25 Utilizar algoritmos de aprendizaje automático para planificar rutas de recolección de residuos de manera dinámica, teniendo en cuenta la información en tiempo real sobre el nivel de llenado de los contenedores. Media

A.2. Requisitos no funcionales

- RNF-01 El sistema debe ser eficiente en la gestión de datos y recursos para minimizar el consumo de energía y recursos. Alta
- RNF-02 Debe ser escalable para adaptarse al crecimiento futuro de la universidad y la adición de más contenedores y usuarios. Alta
- RNF-03 Garantizar la seguridad de los datos recopilados y la privacidad de los usuarios. Alta
- RNF-04 El sistema debe estar disponible las 24 horas del día para garantizar una gestión continua de residuos. Alta
- RNF-05 La aplicación móvil y la interfaz de usuario deben ser intuitivas y fáciles de usar para todos los usuarios. Alta
- RNF-06 Integrar los datos y recursos disponibles a través del programa de la UAB para aprovechar la información existente. Alta

- RNF-07 El desarrollo debe seguir un enfoque *Kanban* y *Agile*, con *sprints* quincenales para entregables funcionales. Alta
- RNF-08 Establecer un plan de mantenimiento continuo para garantizar el funcionamiento óptimo del sistema. Alta
- RNF-09 Garantizar tiempos de respuesta rápidos para alertas de contenedores llenos y comunicaciones con usuarios. Alta

A.3. Riesgos

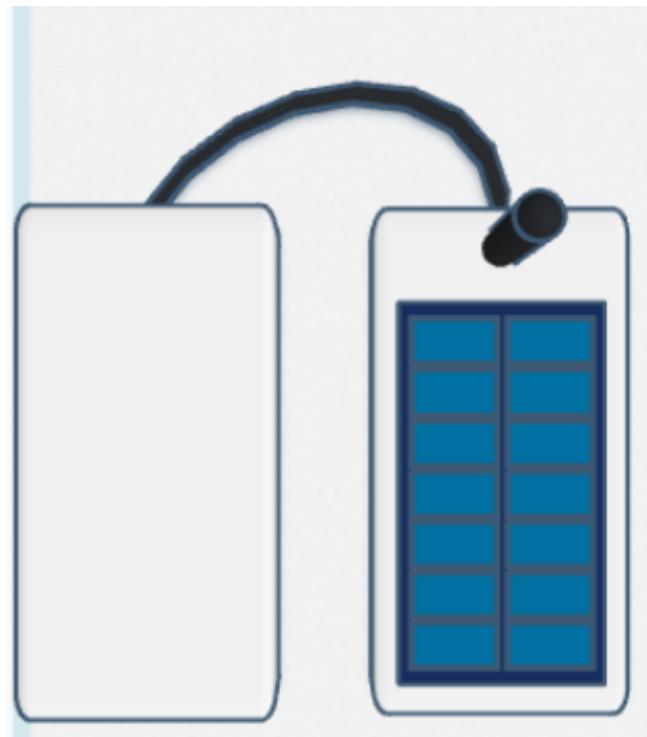
- R-01 Pérdida de Datos Causa Potencial: Fallo en los sistemas de almacenamiento o error humano. Impacto: Pérdida de información crucial y posible retraso en los plazos de entrega.
- R-02 Fallo de Hardware: Causa Potencial: Deterioro de dispositivos, sobrecalentamiento, o daños accidentales. Impacto: Interrupción de las operaciones y posibles retrasos en la ejecución del proyecto.
- R-03 Problemas de Software Causa Potencial: Bugs, incompatibilidad entre versiones, o fallos de sistema. Impacto: Compromiso de la calidad del proyecto y retrasos debidos a la necesidad de solucionar problemas imprevistos.
- R-04 Problemas de Compatibilidad en la Red LoRaWAN Causa Potencial: Dificultades en la interoperabilidad con dispositivos o fallos en la transmisión de datos. Impacto: Inconsistencias en la recopilación de datos y problemas en las comunicaciones del sistema.
- R-05 Problemas de Rendimiento de la APP Causa Potencial: Errores en el código, diseño ineficiente y/o no intuitivo, o problemas de interfaz de usuario. Impacto: Experiencia de usuario deteriorada, rechazo de la aplicación por parte de los usuarios y posibles retrasos para corregir problemas tras el lanzamiento.
- R-06 Errores de Conectividad en la Red LoRaWAN Causa Potencial: Fallos en la comunicación entre los nodos y el servidor central. Impacto: Pérdida de datos y fallas en la operación del sistema.
- R-07 Problemas de Seguridad en la APP Causa Potencial: Vulnerabilidades en el código que podrían ser explotadas para realizar ataques. Impacto: Exposición de datos sensibles y pérdida de confianza de los usuarios.
- R-08 Eliminación Accidental Causa Potencial: Errores humanos como eliminación accidental de archivos. Impacto: Interrupción en la coherencia y calidad de la documentación final.
- R-09 Corrupción de Archivos Causa Potencial: Fallo en el almacenamiento o infección por malware. Impacto: Inaccesibilidad o pérdida de datos importantes.
- R-10 Perdida de Datos por Actualización de Software Causa Potencial: Actualizaciones de software que no son compatibles con versiones anteriores de documentos. Impacto: Incompatibilidad o pérdida de información durante las conversiones de formato.

- R-11 Añadir Nuevas Funcionalidades Causa Potencial: Descubrimiento de necesidades adicionales tras pruebas iniciales o *feedback* de stakeholders. Impacto: Incremento en los recursos y tiempo necesario para implementar nuevas funciones.
- R-12 Modificación de Funcionalidades Existentes Causa Potencial: Inadecuación de las funcionalidades actuales a las necesidades prácticas de los usuarios. Impacto: Posible reescritura de código y reajuste del diseño y funcionalidades existentes.
- R-13 Eliminación de Funcionalidades Planeadas Causa Potencial: Observar redundancias o inutilidades durante las fases de testeo. Impacto: Posible desequilibrio en otras funcionalidades y alteración del flujo de trabajo del sistema.

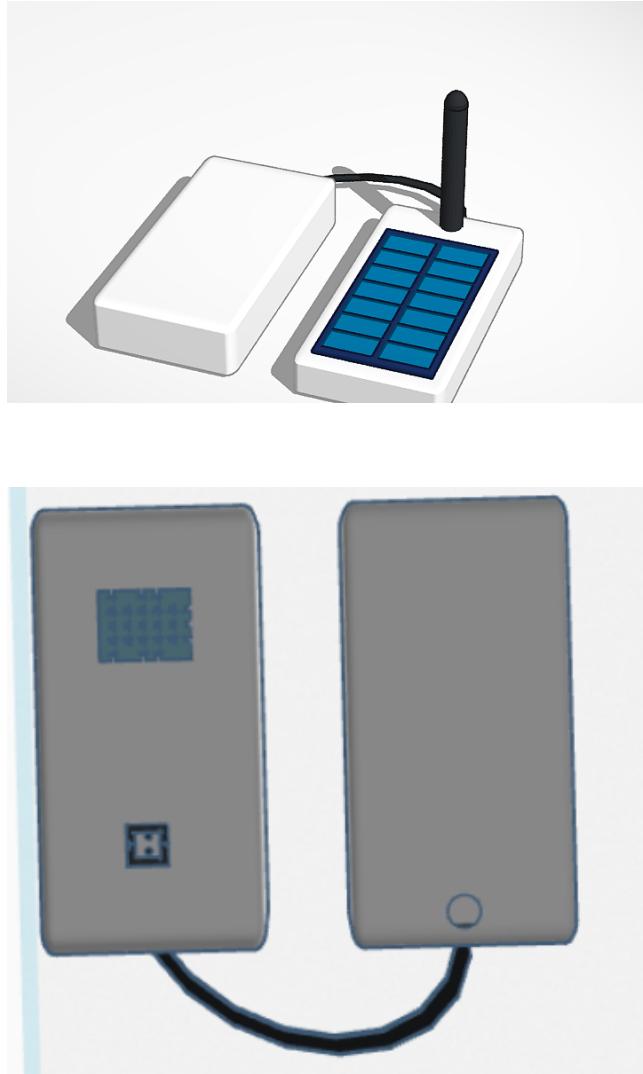
A.4. Mitigación de riesgos

- M-01 Control de Versiones Implementar un sistema de control de versiones, como Git, que permita gestionar los diferentes estados del código fuente y poder revertir a versiones anteriores si es necesario.
- M-02 *Backups* Regulares Establecer una rutina de copias de seguridad automáticas en múltiples ubicaciones (local, cloud, unidades externas) para garantizar que los datos del proyecto están seguros y actualizados.
- M-03 Validaciones y Pruebas Constantes Realizar pruebas frecuentes para validar la funcionalidad y evitar el avance sobre código defectuoso o datos corruptos.
- M-04 Pruebas Unitarias y de Integración Desarrollar y ejecutar pruebas unitarias durante cada etapa del desarrollo para garantizar que cada componente funcione correctamente antes de avanzar.
- M-05 Pruebas de Usabilidad Incluir revisiones de usabilidad para identificar y corregir problemas de interfaz y experiencia de usuario en la APP.
- M-06 Monitoreo Continuo Implementar herramientas de monitoreo en la red LoRaWAN para identificar y corregir proactivamente problemas de conectividad o transmisión.
- M-07 Pruebas Automatizadas Implementar pruebas automáticas que puedan ser ejecutadas en cada etapa del desarrollo para identificar rápidamente errores y corregirlos.
- M-08 Pruebas de Seguridad Implementar pruebas de seguridad para identificar y corregir posibles vulnerabilidades antes de que la aplicación se lance al público.
- M-09 Beta Testing[36] Considerar la implementación de una fase de pruebas beta con usuarios reales para obtener comentarios y realizar mejoras antes del lanzamiento oficial.

- M-10 Almacenamiento en la Nube Utilizar plataformas de almacenamiento en la nube para guardar la documentación en un lugar seguro y accesible.
- M-11 Copias Físicas y Digitales Mantener copias físicas de documentos cruciales y *backups* digitales en diferentes ubicaciones.
- M-12 Canales de Comunicación Efectivos Implementar una plataforma de comunicación eficiente donde los stakeholders puedan proporcionar retroalimentación y observaciones.
- M-13 Reuniones Regulares de Revisión Establecer encuentros periódicos con los clientes para revisar el progreso y discutir posibles ajustes.
- M-14 Prototipos y Pruebas Continuas Ofrecer versiones y prototipos del proyecto en diferentes fases para recibir *feedback* temprano y realizar ajustes antes de llegar a fases avanzadas.



A.5. Diseño BlueCampusDevice



A.6. Esquema Hardware sensor

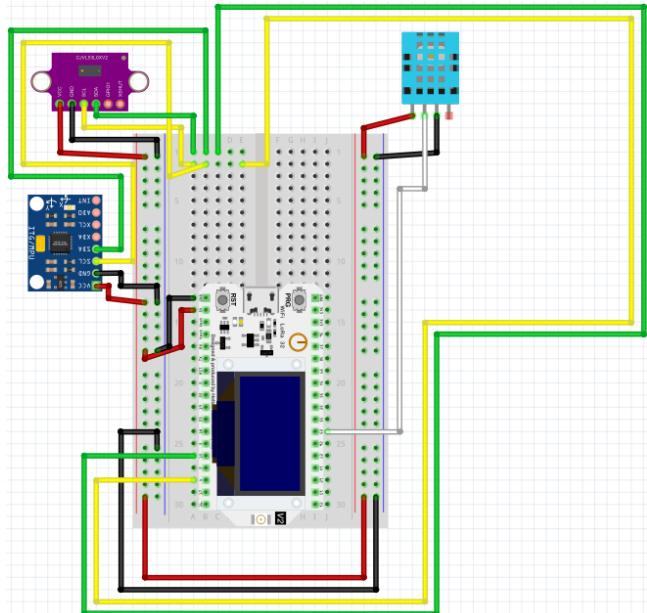


Fig. 4: Esquema del Hardware del sensor

A.7. Diagrama Gantt

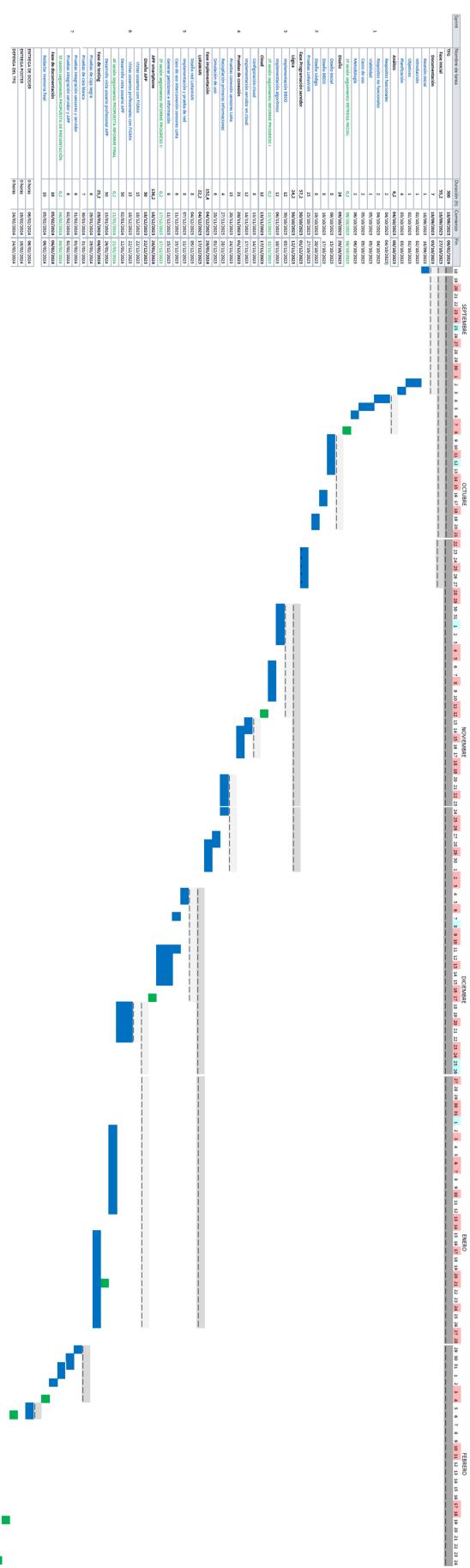


Fig. 5: Diagrama de Gantt del proyecto

A.8. Caso de uso usuario gestor

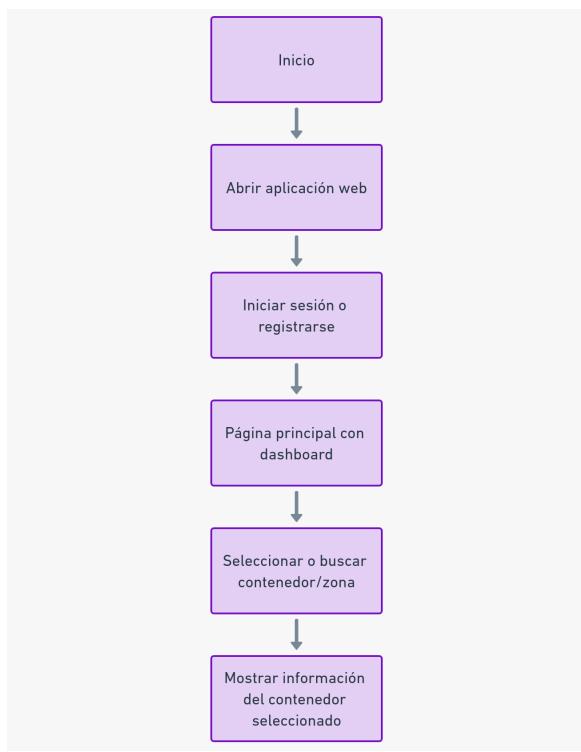


Fig. 6: Usuario gestor

A.9. Caso de uso usuario campus

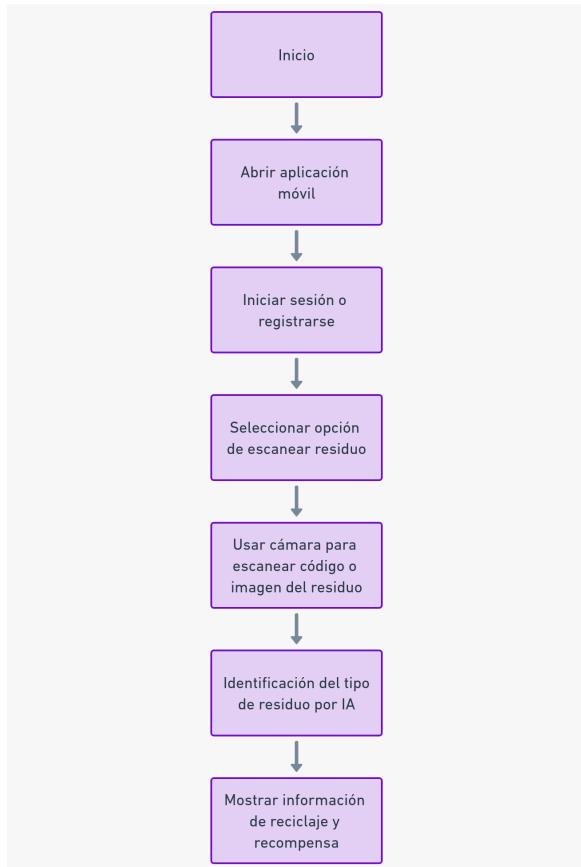


Fig. 7: Usuario campus

A.10. Arquitectura completa

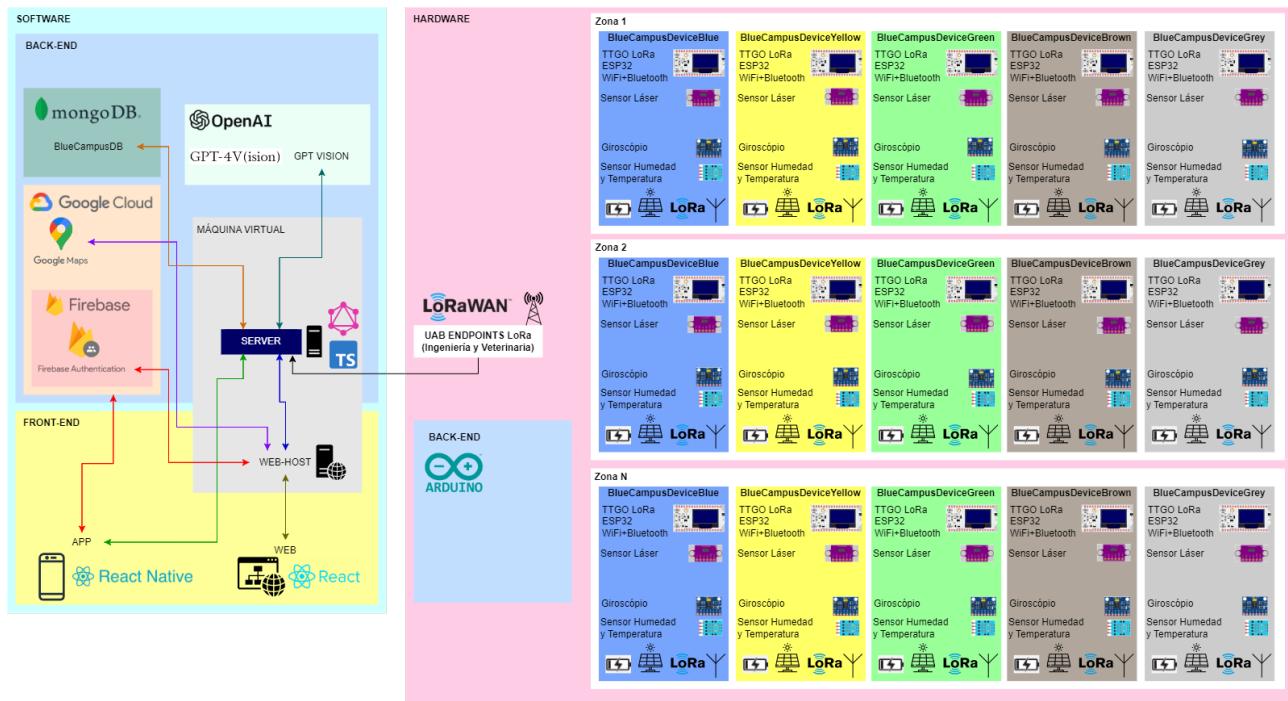


Fig. 8: Arquitectura completa HW-SW

A.11. Diseño de la red

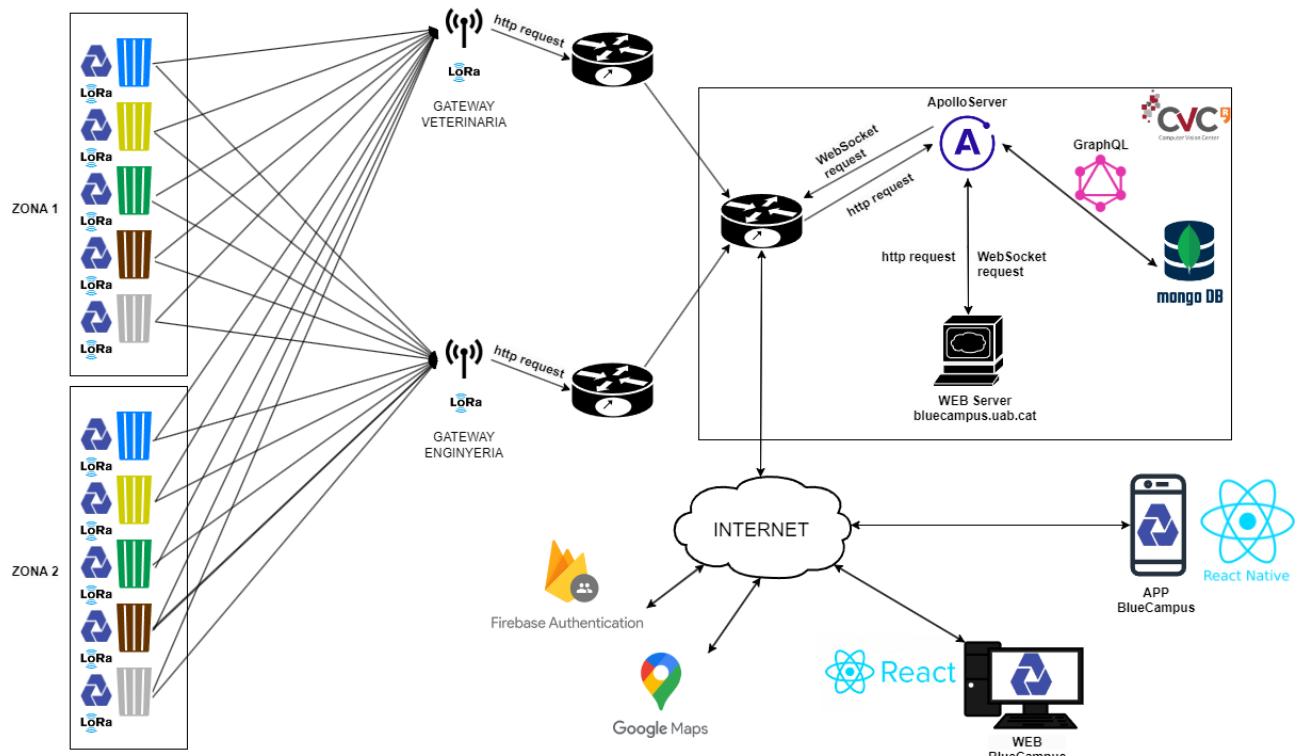


Fig. 9: Diseño de la red

A.12. Prototipo

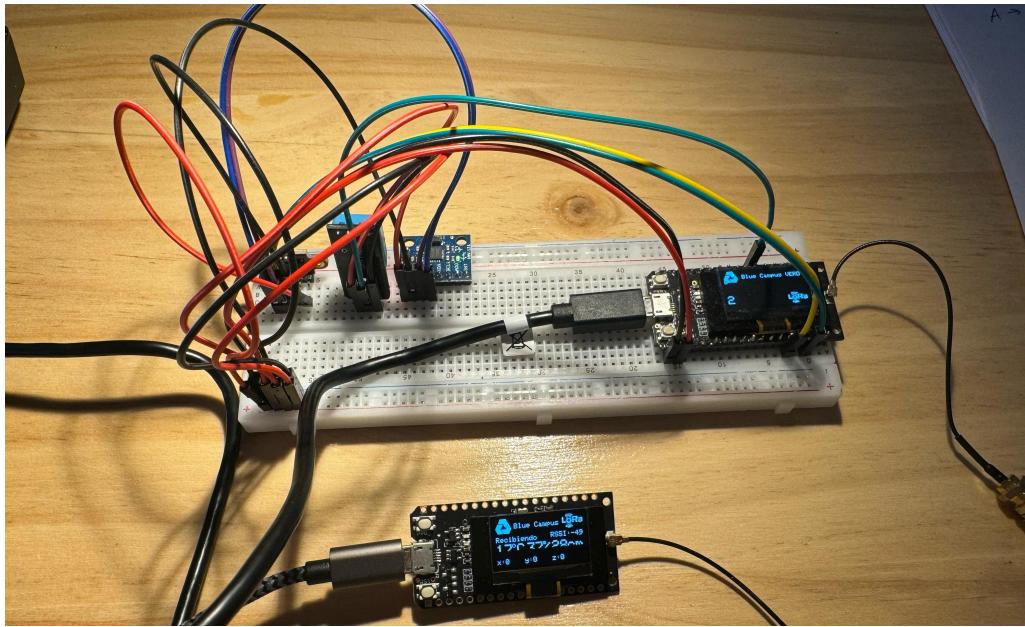


Fig. 10: Sender y receiver

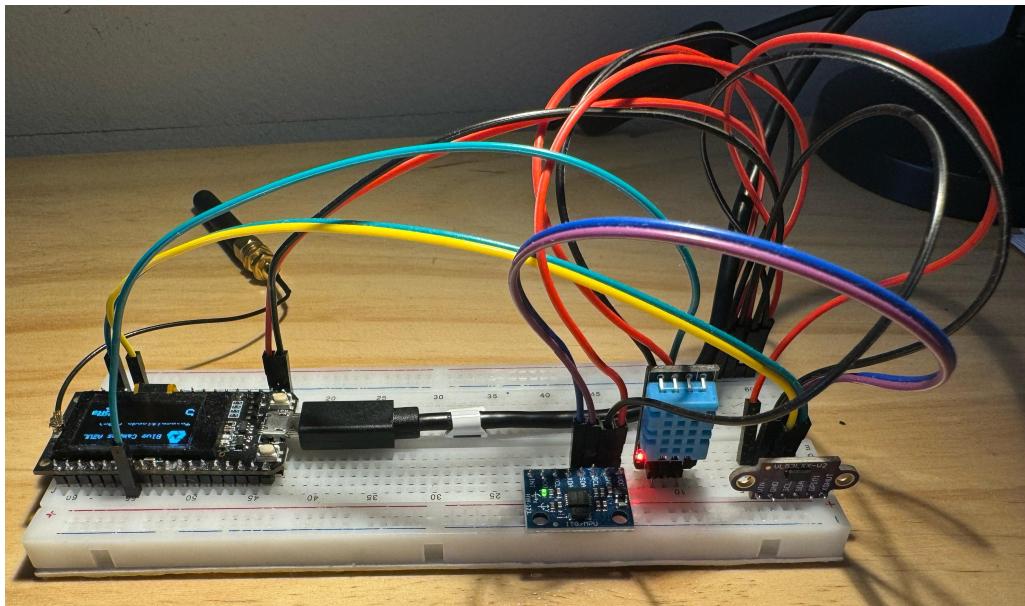


Fig. 11: Sender

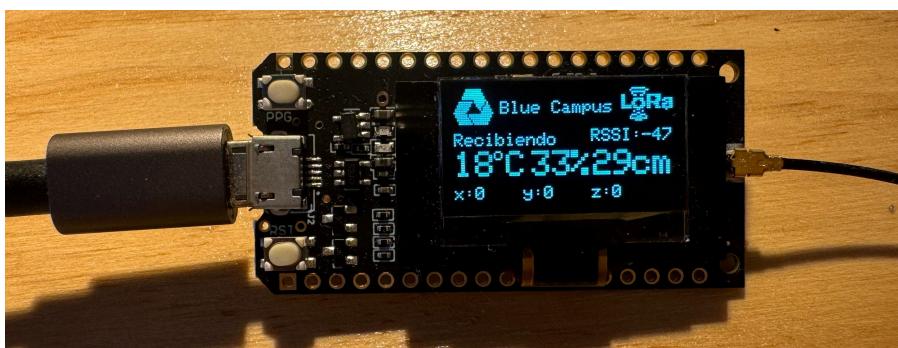


Fig. 12: Receiver