

---

Faculty Attendance Management System

---

Sistema de Gestión de Asistencia del Profesorado

---



## TRABAJO DE FIN DE GRADO

Autores:

**Ángel Gallego Muñoz: 9.4**

**Raúl Durán Catalán: 9.0**

Directores:

**Daniel Bascones García**

**Borja Morcillo Salgado**

Universidad Complutense de Madrid

26 de Mayo del 2025

*Dedicado a nuestros tutores Daniel  
y Borja que han estado tanto en las  
buenas, como en las tan buenas.*

*Dedicado a Jose Manuel Mendias,  
porque el que vale vale y el que  
no pa' software.*

*Dedicado a nuestras familias y  
amigos que nos han apoyado un poco*

# Abstract

The faculties of the Complutense University of Madrid must monitor teaching activities, ensuring that it is taught normally throughout the entire academic year. At the end of the year, all faculties are required to report any incidents that occurred during the period and the number of teaching hours that were not taught, in order to analyze how the institution is performing.

Currently, at the Faculty of Computer Science, this monitoring is carried out using sheets distributed in the classrooms that must be signed by the professors and collected manually. After verifying each sheet, notifications are sent to the professors for every class that was not taught to check if there is any justification for the absence.

In this project, we aim to eliminate the manual burden of collecting sheets in classrooms and verifying conducted classes by implementing a system that tracks teacher attendance and displays previous days' incidents on a web page. To achieve this, each classroom will be equipped with a node built around an ESP32-C6 microcontroller. At these nodes, teachers can identify themselves using an RFID card, and the nodes will transmit attendance data via LoRa communication to a server, implemented on a Raspberry Pi 5, where the database and the web page are hosted.

As a result, a system has been developed capable of managing the entire area of the Faculty of Computer Science and the classrooms in the multipurpose building, with an average autonomy of just over 3 years. The server will be responsible for managing the data and generating the final report to be submitted, which includes a summary of the incidents throughout the academic year.

**Keywords:** Faculty Attendance Management, Teaching monitoring, LoRa, RFID, IoT, web application, ESP32.

# Resumen

Las facultades de la Universidad Complutense de Madrid han de realizar un seguimiento de la docencia, procurando que ésta se imparta con normalidad durante todo el año académico. Al finalizar el año, todas las facultades tienen la obligación de presentar qué incidencias han ocurrido durante el periodo y cuántas horas de docencia no han sido impartidas para hacer un análisis de cómo está funcionando la institución.

Actualmente, en la facultad de informática este seguimiento se realiza mediante hojas repartidas en los aulas que han de ser firmadas por los profesores y recogidas a mano. Después de la comprobación de cada una de las hojas, se procede a notificar cada una de las clases no impartidas a los profesores para comprobar que exista alguna justificación para dicha falta.

En este proyecto queremos liberar toda esa carga manual de la recogida de hojas en las aulas y comprobación de las clases impartidas mediante un sistema que realice el seguimiento de la asistencia de los profesores y el visualizado de las incidencias de días previos en una página web. Para ello dispondremos de un nodo, implementado con un microcontrolador ESP32-C6, en cada aula. En dichos nodos los profesores podrán identificarse con una tarjeta RFID y estos nodos trasladaran la asistencia mediante comunicación LoRa a un servidor, implementado en una Raspberry Pi 5, donde se encuentra la base de datos y la página web.

Como resultado se ha desarrollado un sistema capaz de gestionar todo el área de la facultad de informática y las aulas del edificio multiusos, con una autonomía media de algo más de 3 años. El servidor se encargará de gestionar los datos y generar el informe final que ha de entregarse en el que se recoge el resumen de incidencias del año.

**Palabras clave:** Control de docencia, Seguimiento docente, LoRa, RFID, IoT, Aplicación web, ESP32.

# Índice general

Índice de figuras

Índice de tablas

<b>1. Introducción</b>	<b>1</b>
1.1. Antecedentes y motivación . . . . .	1
1.1.1. El problema del control de impartición de clases . . . . .	1
1.1.2. Situación actual . . . . .	2
1.2. Objetivos . . . . .	4
1.3. Plan de trabajo . . . . .	5
<b>2. Introduction</b>	<b>6</b>
2.1. Background and motivation . . . . .	6
2.1.1. The problem of monitoring faculty attendance . . . . .	6
2.1.2. Current situation . . . . .	7
2.1.3. Previous system: Manual control via signatures . . . . .	8
2.2. Objectives . . . . .	9
2.3. Work plan . . . . .	10
<b>3. Contexto Tecnológico</b>	<b>11</b>
3.1. Tecnologías de Comunicación Inalámbrica entre nodos . . . . .	12
3.1.1. LoRa . . . . .	12
3.1.2. ZigBee . . . . .	13

3.1.3.	Bluetooth . . . . .	14
3.2.	Tecnologías de registro e identificación . . . . .	15
3.2.1.	RFID . . . . .	16
3.2.2.	NFC . . . . .	17
3.2.3.	Biometría Dactilar . . . . .	18
3.3.	Alternativas de Placas de Bajo Coste con Capacidad IoT . . . . .	18
3.3.1.	Raspberry Pi . . . . .	19
3.3.2.	ODROID . . . . .	20
3.3.3.	Espressif ESP32 . . . . .	20
3.4.	Web/Server . . . . .	21
3.4.1.	Tecnologías para desarrollo web . . . . .	21
<b>4.</b>	<b>Implementación</b>	<b>24</b>
4.1.	Desarrollo Servidor . . . . .	25
4.1.1.	Base de Datos . . . . .	25
4.1.2.	Proceso Automático . . . . .	26
4.1.3.	Funcionalidades implementadas . . . . .	28
4.2.	Desarrollo de los nodos . . . . .	34
4.2.1.	Componentes hardware . . . . .	34
4.2.2.	Desarrollo software . . . . .	39
4.2.3.	Librerías externas . . . . .	39
4.2.4.	Librerías de ESP-IDF . . . . .	40
4.3.	Guía de instalación . . . . .	41
4.3.1.	Instalación servidor . . . . .	41
4.3.2.	Instalación de los nodos . . . . .	43
<b>5.</b>	<b>Resultados</b>	<b>45</b>
5.1.	Repositorios del proyecto . . . . .	45

5.2.	Estructura del sistema . . . . .	45
5.3.	Funcionamiento de la web . . . . .	46
5.4.	Funcionamiento de los nodos . . . . .	46
5.5.	Pruebas . . . . .	47
5.5.1.	Servidor . . . . .	47
5.5.2.	Consumo . . . . .	47
5.5.3.	Distancia . . . . .	49
<b>6.</b>	<b>Conclusiones y trabajo futuro</b>	<b>51</b>
6.1.	Conclusiones . . . . .	51
6.2.	Líneas de trabajo futuro . . . . .	51
<b>7.</b>	<b>Conclusion and future work</b>	<b>53</b>
7.1.	Conclusion . . . . .	53
7.2.	future work . . . . .	53
<b>8.</b>	<b>Contribuciones personales</b>	<b>55</b>
8.1.	Aportaciones de Raúl Durán Catalán . . . . .	55
8.1.1.	Declaración de requisitos . . . . .	55
8.1.2.	Investigación . . . . .	55
8.1.3.	Sistema empotrado . . . . .	56
8.1.4.	Memoria . . . . .	56
8.2.	Aportaciones de Ángel Gallego Muñoz . . . . .	57
8.2.1.	Diseño e implementación de la base de datos . . . . .	57
8.2.2.	Desarrollo del backend del servidor . . . . .	57
8.2.3.	Desarrollo de la interfaz web . . . . .	58
8.2.4.	Memoria . . . . .	58
	<b>Bibliography</b>	<b>59</b>

# Índice de figuras

1.1. Sistema de control de presencia TM-626 . . . . .	2
2.1. Time Attendance Control System TM-626 . . . . .	7
3.1. Esquema de la arquitectura LoRa[22] . . . . .	13
4.1. Esquema del sistema. . . . .	24
4.2. Estructura de la base de datos. . . . .	25
4.3. Servicios automáticos . . . . .	28
4.4. Ventana de inicio de sesión . . . . .	28
4.5. Ventana principal del sistema . . . . .	29
4.6. Listado de profesores registrado en el sistema . . . . .	30
4.7. Listado de profesores registrado en el sistema . . . . .	30
4.8. Listado de aulas registrado en el sistema . . . . .	31
4.9. Listado de asignaturas registrado en el sistema . . . . .	32
4.10. Listado de días no lectivos registrado en el sistema . . . . .	32
4.11. Listado de incidencias registrado en el sistema . . . . .	33
4.12. Estadísticas generadas por el sistema . . . . .	34
4.13. Firebeetle 2 ESP32-C6 . . . . .	35
4.14. Datasheet ESP32 Firebeetle 2 C6 . . . . .	35
4.15. Modulo LoRa DX-LR01 . . . . .	36
4.16. Modulo RFID RC522 . . . . .	36
4.17. Esquema de conexiones . . . . .	37

4.18. Conexiones de los modulos al ESP32 . . . . .	37
4.19. Transistor 2N7000 . . . . .	37
4.20. Tarjeta RFID MIFARE Classic 1k . . . . .	38
4.21. Batería 1100 mA . . . . .	38
4.22. Modelo de la caja . . . . .	39
4.23. Diagrama de Estados de los nodos . . . . .	41
4.24. Variable AULA a modificar . . . . .	44
5.1. Conexiones para la prueba de consumo . . . . .	48
5.2. Consumo en modo deep-sleep . . . . .	48
5.3. Transacción de 6 segundos . . . . .	49
5.4. Transacción fallida de 10 segundos . . . . .	49

# Índice de tablas

3.1. Bandas de frecuencia LoRaWAN por región o país. [17]. . . . .	12
3.2. Características clave de la tecnología LoRa . . . . .	13
3.3. Características principales de la tecnología Zigbee . . . . .	15
3.4. Características principales de la tecnología Bluetooth . . . . .	16
3.5. Características principales de la tecnología RFID . . . . .	17
3.6. Características principales de la tecnología NFC . . . . .	17
3.7. Características principales de la biometría dactilar . . . . .	18
3.8. Comparativa de placas de desarrollo . . . . .	21
3.9. Diferentes tecnologías Web y sus características . . . . .	23

# Capítulo 1

## Introducción

Actualmente, una de las principales prioridades de las universidades es la calidad de sus propios grados o másteres, y uno de los factores que afectan directamente a este aspecto es el seguimiento de qué clases se han impartido y cuáles no.

Este Trabajo de Fin de Grado aborda la transformación digital de un sistema tradicionalmente analógico: el control de qué clases se han impartido y cuáles no. Hasta ahora, este seguimiento se ha realizado mediante hojas de firma en papel distribuidas en las aulas día tras día, donde los profesores deben registrar manualmente su asistencia y la realización de cada sesión docente, para su posterior recogida y recuento de firmas. Este método presenta importantes limitaciones en términos de eficiencia, fiabilidad y aprovechamiento de la información recopilada.

La propuesta desarrollada en este TFG consiste en diseñar e implementar una solución tecnológica que permita digitalizar completamente este proceso, eliminando el uso de papel, automatizando la recopilación y procesamiento de datos, y proporcionando herramientas de análisis que faciliten la toma de decisiones basadas en la información recogida.

### 1.1. Antecedentes y motivación

#### 1.1.1. El problema del control de impartición de clases

El control efectivo de las clases impartidas en el entorno universitario representa un desafío administrativo significativo para las instituciones de educación superior. Este proceso de verificación es fundamental por múltiples razones:

En primer lugar, las universidades tienen la responsabilidad de garantizar que la docencia programada se realice según los planes de estudio aprobados. Esto responde tanto a compromisos con los estudiantes, los cuales se recogen en sus matrículas esperando recibir la formación acordada, como a requisitos de calidad establecidos por organismos reguladores.

Desde una perspectiva de gestión académica, resulta imprescindible verificar el cum-

plimiento de las obligaciones docentes del profesorado establecidas en su carga lectiva. Este seguimiento permite identificar incidencias recurrentes que puedan requerir ajustes organizativos o medidas correctivas.

Adicionalmente, contar con registros precisos sobre la impartición real de las clases facilita la planificación académica a medio y largo plazo, permitiendo optimizar la asignación de espacios, detectar necesidades educativas específicas y realizar ajustes en los horarios cuando sea necesario.

Existen diferentes sistemas comercializados para solventar este problema, desde aplicaciones web o multiplataforma que permitan registrar la actividad como si de fichar se tratase, la aplicación de Factorial es un buen ejemplo de ello. También existen dispositivos físicos que incluyen medidas más robustas de seguridad, como lectores dactilares para contabilizar la asistencia, como el TM-626 de la Figura 1.1. Al tratarse de sistemas de fichaje creados para empresas, estos necesitan saber cuándo entra del aula el profesor y cuándo sale, para contabilizar su turno.



Figura 1.1: Sistema de control de presencia TM-626

Este tipo de sistemas tiene una funcionalidad muy útil en el contexto en el que estamos tratando y es el aviso y justificación de faltas que la gran mayoría de estos sistemas posee. Sin embargo, también tendríamos de funcionalidades que no se utilizan en este contexto, como puede ser la gestión de vacaciones o la gestión de nóminas y finanzas, ya que estos sistemas tienen un enfoque más en el trabajo, lo cual nuestro sistema está más orientado en control de docencia, no en el trabajo de cada profesor.

### 1.1.2. Situación actual

Una opción sería implementar alguna de las opciones comercializadas, pero existiría un coste por mantenimiento y muchas funcionalidades que no son útiles en el contexto que queremos implementarlo. Por eso se propuso un proyecto de una forma automática mediante un dispositivo que recogiese el acceso de un profesor al aula y lo enviase a un servidor donde se gestionaría. Para tener en cuenta todos los requisitos necesarios para la implementación y buen uso de este dispositivo, se tuvo una reunión con Mar Santos Franco, jefa de la sección de personal.

El sistema impuesto actualmente se basado en recogida de hojas del día anterior de las aulas, junto a ellas se toman también los registros de los laboratorios. Una vez recabadas todas las actas de la facultad, se comprueban una a una que todos los horarios propuestos hayan sido cubiertos por los profesores, ya sea en el aula o en el laboratorio, puesto que hay horarios que tienen ambos asignados, o en el caso de que haya dos o más profesores asignados a un horario, que al menos uno haya asistido. Una vez comprobadas todas las actas, se procede a notificar las incidencias, que ocurren cuando ninguno de los profesores asignados a un horario no se encuentra en ninguna de las actas del aula o laboratorio asignado.

Dado lo cual, estas incidencias pueden dividirse en varios subtipos:

**Falta:** Aquellas incidencias que se deben a las faltas de docencia que se dan sin justificación. Estas incidencias pueden dar lugar a otro tipo de incidencia una vez que han sido justificadas o simplemente ser justificadas.

**Suplencia:** Se produce cuando el profesor asignado no ha impartido la docencia de ese horario, pero otro profesor ha sido el encargado de impartir la docencia. Esta incidencia puede ser justificada con anterioridad generando la incidencia y asignando a un profesor la clase de ese día en específico, o justificando con posterioridad una vez que se ha impartido la docencia por un profesor no asignado en un principio.

**Sustitución:** Ocurre cuando un profesor asignado no va a impartir una clase durante un cierto tiempo, y se asigna otro profesor que sea el encargado de impartir la docencia, al igual que en las suplencias se puede justificar con anterioridad asignando otro profesor a la clase, o con posterioridad una vez que otro profesor haya comenzado a impartir la docencia de esa asignatura.

**Recuperación:** Hace referencia a una clase no impartida, pero es trasladada a otro horario fuera del habitual con el mismo profesor encargado de impartirla para recuperar esas horas perdidas, después de lo cual quedaría justificada una vez impartida la docencia.

**Intercambio de clase:** Implica un cambio de horario entre dos profesores, siempre que haya sido notificado con antelación.

En el caso de que surja una incidencia sin justificación previa, a día de hoy normalmente faltas, el protocolo a seguir es detectar la incidencia y, acto seguido, notificar mediante correo electrónico al causante de la incidencia y al jefe del departamento del que es miembro, para poner en conciencia que existe una incidencia para un horario en específico a la cual no tiene justificación. En respuesta a esto, se espera recibir una respuesta que justifique la incidencia o cambie el tipo de la misma.

Todo este proceso de recogida de actas y análisis de las mismas se realiza para la creación de un informe obligatorio anual para inspecciones sobre todas las incidencias ocurridas durante el año académico. En el caso de la facultad de informática, realizan un informe por cada cuatrimestre y a final del año unen ambos informes para aportar también el informe anual. En dicho informe han de aportar el número de clases impartidas durante el año académico, el número de incidencias y, de ellas, cuántas han sido justificadas y cuántas no tienen justificación. Los informes generados se almacenan; sin embargo, todas las actas diarias de curso únicamente se almacenan durante ese curso. Una vez el informe ha sido enviado y aprobado, todos los registros de asistencia de las aulas y los laboratorios

se eliminan.

### **Sistema anterior: control manual mediante firmas**

El sistema tradicional implementado hasta ahora consiste en hojas de firma físicas ubicadas en cada aula del campus. Al comienzo o finalización de cada sesión, el profesor responsable debe firmar manualmente en la casilla correspondiente a su asignatura, horario y grupo, confirmando así que la clase programada ha sido efectivamente impartida.

Este método presenta numerosas deficiencias que comprometen su eficacia. En primer lugar, todo el procesamiento es manual: las hojas de firma deben ser recogidas periódicamente por personal administrativo, quien posteriormente debe revisar cada documento, verificar que todas las firmas correspondientes estén presentes e introducir manualmente esta información en algún sistema de registro. Este proceso consume una cantidad considerable de recursos humanos y tiempo.

Además, al ser en formato físico, las hojas de papel son susceptibles a daños, pérdidas o deterioro. Una hoja extraviada o dañada supone la pérdida irreversible de los datos de asistencia correspondientes a múltiples sesiones. A esto se añade que la información no está disponible en tiempo real, sino que debe esperar a ser procesada manualmente, lo que impide detectar y resolver rápidamente incidencias como clases no impartidas. Otro de los problemas es el almacenamiento físico: las hojas de firma acumuladas generan un volumen considerable de documentación que debe ser archivada físicamente, ocupando espacio y dificultando su consulta posterior.

Finalmente, este sistema supone un inconveniente para el profesorado, ya que pueden olvidar firmar al estar concentrados en la preparación o conclusión de sus clases, especialmente cuando tienen sesiones consecutivas en diferentes aulas de la facultad.

La ineficacia de este sistema manual, junto con sus limitaciones para proporcionar información oportuna y procesable, motiva el desarrollo de una alternativa digital que aproveche las tecnologías actuales para mejorar significativamente este proceso esencial en la gestión universitaria. La implementación de un sistema digital no solo busca eliminar las deficiencias del método tradicional, sino también aportar nuevas funcionalidades como la generación automática de informes.

## **1.2. Objetivos**

El objetivo principal es desarrollar un sistema digital distribuido para el control de impartición de clases universitarias que sustituya el actual método basado en firmas manuscritas en papel. Este sistema se estructurará mediante una arquitectura de nodos interconectados que permitirá un registro electrónico eficiente y una gestión centralizada de la información.

El proyecto se centrará en diseñar e implementar nodos electrónicos individuales para cada aula donde los profesores podrán registrar fácilmente la impartición de sus clases mediante un sistema de tarjetas identificativas. Estos nodos periféricos se conectarán de

forma inalámbrica a un nodo central que recopilará la información de todas las aulas del campus, actualizará una base de datos centralizada y alojará la aplicación web de gestión del sistema. La comunicación entre nodos se realizará mediante un protocolo seguro y eficiente que garantice la integridad de los datos incluso en situaciones de conectividad intermitente.

Complementariamente, se desarrollará una interfaz web intuitiva que permitirá visualizar en tiempo real el estado de impartición de las clases, generar informes personalizados sobre cumplimiento docente, administrar usuarios, aulas y asignaturas, y gestionar incidencias relacionadas con el registro de clases. Esta plataforma web facilitará tanto la operación diaria del sistema como el análisis estadístico del cumplimiento docente. La solución incorporará un sistema de autenticación mediante tarjetas que identificará unívocamente a cada profesor, simplificando el proceso de registro y aumentando su fiabilidad.

### 1.3. Plan de trabajo

Para garantizar el cumplimiento de los objetivos, el proyecto se ha estructurado en cinco fases secuenciales.

Fase de análisis y diseño (8 semanas) El proceso comenzará con un estudio detallado del sistema actual y la identificación de requisitos específicos, seguido de una evaluación comparativa de tecnologías disponibles (como LoRa, Wi-Fi o Bluetooth) para seleccionar las más adecuadas en términos de alcance, consumo energético y coste. Paralelamente, se definirá la arquitectura general del sistema, incluyendo el diseño de la base de datos relacional que almacenará los registros de asistencia y el modelo de datos para garantizar escalabilidad y seguridad.

Fase de desarrollo hardware (4 semanas) En esta etapa se construirán prototipos funcionales de los nodos de registro, utilizando microcontroladores como ESP32-C6. Se implementará el firmware para gestionar sensores biométricos o RFID, se configurará el nodo maestro (responsable de agregar datos) y se validará la comunicación mediante pruebas de alcance y estabilidad en diferentes escenarios.

Fase de desarrollo software (4 semanas) Aquí se implementará el servidor central integrándolo a una base de datos MySQL. Se desarrollará una aplicación web con funciones de administración de usuarios, visualización de registros y generación de informes.

Fase de integración y pruebas (4 semanas) Se realizará la interconexión de todos los componentes en un entorno controlado, seguida de una prueba piloto en 2-3 aulas con simulaciones de escenarios reales (picos de acceso, fallos de conexión, suplantación de identidad). Los resultados permitirán optimizar el rendimiento y corregir posibles inconsistencias.

Fase de documentación y finalización (2 semanas) Incluye la redacción de la memoria técnica con diagramas de flujo y especificaciones, y manuales de instalación, la preparación de una presentación ejecutiva para la defensa del proyecto, y la entrega formal de los prototipos funcionales.

# Capítulo 2

## Introduction

Currently, one of the main priorities for universities is the quality of their own undergraduate and master's programs, and one of the factors that directly affects this aspect is monitoring which classes have been taught and which have not.

This Final Degree Project addresses the digital transformation of a traditionally analog system: the monitoring of which classes have been taught and which have not. Until now, this monitoring has been conducted through paper sign-in sheets distributed daily in classrooms, where professors must manually record their attendance and the completion of each teaching session, for later collection and signature counting. This method presents significant limitations in terms of efficiency, reliability, and the utilization of the information collected.

The proposal developed in this Final Degree Project consists of designing and implementing a technological solution that fully digitizes this process, eliminating the use of paper, automating data collection and processing, and providing analytical tools that facilitate decision-making based on the information gathered.

### 2.1. Background and motivation

#### 2.1.1. The problem of monitoring faculty attendance

Effective monitoring of classes taught in the university environment represents a significant administrative challenge for higher education institutions. This verification process is essential for several reasons:

First, universities have the responsibility to ensure that scheduled teaching is carried out according to the approved curricula. This responds both to commitments made to students, who expect to receive the agreed-upon education as per their enrollment, and to quality requirements established by regulatory bodies.

From an academic management perspective, it is crucial to verify compliance with the teaching obligations of faculty members as established in their workload. This monitoring

helps identify recurring issues that may require organizational adjustments or corrective measures.

Additionally, having accurate records of the actual delivery of classes facilitates medium- and long-term academic planning, enabling optimization of space allocation, identification of specific educational needs, and adjustments to schedules when necessary.

There are various commercial systems available to address this issue, ranging from web or multiplatform applications that allow activity recording similar to clocking in, with the Factorial application being a good example, to physical devices that include more robust security measures such as fingerprint readers to track attendance, like the TM-626 shown in Figure 2.1. Since these clocking-in systems were designed for businesses, they need to know when the professor enters and leaves the classroom to calculate their shift.



Figura 2.1: Time Attendance Control System TM-626

This type of system has a very useful feature in the context we are addressing: the notification and justification of absences, which the vast majority of these systems include. However, they also offer functionalities that are not used in this context, such as vacation management or payroll and finance management, since these systems are more work-oriented, whereas our system is focused on attendance control, not on each professor's work management.

### **2.1.2. Current situation**

One option would be to implement one of the commercial solutions available, but these come with maintenance costs and many features that are not useful in the context we want to apply it. Therefore, a project was proposed to automate the process by using a device that records a professor's entry to the classroom and sends the data to a server for management. To consider all necessary requirements for implementing and properly using this device, a meeting was held with Mar Santos Franco, head of the personnel department.

The current system is based on collecting the previous day's sheets from the classrooms, along with the records from the laboratories. Once all the faculty reports are

gathered, each schedule is verified one by one to ensure that all classes have been covered by professors, whether in the classroom or laboratory—since some schedules include both—or in cases where two or more professors are assigned to a schedule, at least one has attended. After verifying all reports, incidents are notified, which occur when none of the professors assigned to a schedule appear in any of the classroom or laboratory reports.

Given this, these incidents can be divided into several subtypes:

**Absence:** Incidents that correspond to attendance absences without justification. These incidents may lead to another type of incident once they have been justified or may simply be justified.

**Substitution:** This occurs when the assigned professor has not taught the class for that schedule, but another professor has taken over the teaching. This incident can be justified in advance by creating the incident and assigning a professor to teach that specific class, or justified afterward once the class has been taught by a professor who was not originally assigned.

**Class recovery:** Refers to a class that was not taught as scheduled but is rescheduled to a different time outside the usual schedule with the same professor responsible for teaching it, to recover the lost hours. Once the class is delivered, the absence is then considered justified.

**Class Swap:** Involves a schedule change between two professors, provided it has been notified in advance.

In the case of an incident arising without prior justification, usually absences, the current protocol is to detect the incident and immediately notify, via email, both the person responsible for the incident and the head of their department, to make them aware that there is an unjustified incident for a specific schedule. In response, a justification for the incident is expected or a change in the type of incident.

The entire process of collecting and analyzing reports is carried out to produce a mandatory annual report for inspections on all incidents that occurred during the academic year. At the Faculty of Computer Science, a report is made for each semester, and at the end of the year, both reports are combined to produce the annual report. This report must include the number of classes taught during the academic year, the number of incidents, and of these, how many were justified and how many were not. The generated reports are stored; however, all daily attendance sheets are only kept for that academic year. Once the report has been submitted and approved, all attendance records from classrooms and laboratories are deleted.

### 2.1.3. Previous system: Manual control via signatures

The traditional system implemented until now consists of physical sign-in sheets placed in each classroom on campus. At the beginning or end of each session, the responsible professor must manually sign in the box corresponding to their subject, schedule, and group, thereby confirming that the scheduled class has indeed been taught.

This method presents numerous shortcomings that compromise its effectiveness. First, all processing is manual: the sign-in sheets must be periodically collected by administrative staff, who then have to review each document, verify that all required signatures are present, and manually enter this information into a recording system. This process consumes a significant amount of human resources and time.

Moreover, since the sheets are in physical form, they are susceptible to damage, loss, or deterioration. A lost or damaged sheet means the irreversible loss of attendance data for multiple sessions. Additionally, the information is not available in real time but must wait to be processed manually, which prevents the quick detection and resolution of incidents such as unattended classes. Another problem is physical storage: accumulated sign-in sheets create a considerable volume of documentation that must be physically archived, taking up space and making later consultation difficult.

Finally, this system poses an inconvenience for professors, as they may forget to sign when focused on preparing or concluding their classes, especially when they have consecutive sessions in different classrooms across the faculty.

The inefficiency of this manual system, along with its limitations in providing timely and actionable information, motivates the development of a digital alternative that leverages current technologies to significantly improve this essential process in university management. The implementation of a digital system not only seeks to eliminate the shortcomings of the traditional method but also to provide new functionalities such as automatic report generation.

## 2.2. Objectives

The primary objective is to develop a distributed digital system for monitoring university class delivery that replaces the current method based on handwritten paper signatures. This system will be structured through an architecture of interconnected nodes, enabling efficient electronic registration and centralized information management.

The project will focus on designing and implementing individual electronic nodes for each classroom, where professors can easily record their class delivery using an identification card system. These peripheral nodes will wirelessly connect to a central node that collects information from all classrooms on campus, updates a centralized database, and hosts the system's management web application. Communication between nodes will use a secure and efficient protocol that ensures data integrity even in situations of intermittent connectivity.

Additionally, an intuitive web interface will be developed to provide real-time visualization of class delivery status, generate customized reports on teaching compliance, manage users, classrooms, and subjects, and handle incidents related to class registration. This web platform will facilitate both daily system operation and statistical analysis of teaching compliance.

The solution will incorporate an authentication system using cards that uniquely identify each professor, simplifying the registration process and increasing its reliability.

## 2.3. Work plan

To ensure the achievement of the objectives, the project has been structured into five sequential phases.

**Analysis and Design Phase (8 weeks)** The process will begin with a detailed study of the current system and the identification of specific requirements, followed by a benchmarking of available technologies (such as LoRa, Wi-Fi, or Bluetooth) to select the most suitable in terms of range, power consumption, and cost. Simultaneously, the overall system architecture will be defined, including the design of the relational database to store attendance records and the data model to ensure scalability and security.

**Hardware Development Phase (4 weeks)** In this stage, functional prototypes of the registration nodes will be built using microcontrollers like the ESP32-C6. Firmware will be implemented to manage biometric sensors or RFID, the master node (responsible for aggregating data) will be configured, and communication will be validated through range and stability tests in different scenarios.

**Software Development Phase (4 weeks)** The central server will be implemented and integrated with a MySQL database. A web application will be developed featuring user management, record visualization, and report generation functionalities.

**Integration and Testing Phase (4 weeks)** All components will be interconnected in a controlled environment, followed by a pilot test in 2-3 classrooms simulating real scenarios (peak access, connection failures, identity spoofing). The results will enable performance optimization and correction of possible inconsistencies.

**Documentation and Completion Phase (2 weeks)** This includes writing the technical report with flowcharts and specifications, installation manuals, preparing an executive presentation for the project defense, and the formal delivery of functional prototypes.

# Capítulo 3

## Contexto Tecnológico

En el campo de la tecnología, especialmente en años recientes, hemos visto un progreso notable en áreas clave como la comunicación inalámbrica y los sistemas de identificación. Estas mejoras no solo han cambiado diferentes industrias, sino que también han creado muchas oportunidades para desarrollar soluciones tecnológicas más efectivas y accesibles. En este contexto, hay una amplia gama de tecnologías que se destacan por sus características y usos específicos, cada una ayudando de manera especial a optimizar procesos en diversos entornos.

Este capítulo tiene como objetivo proporcionar una perspectiva integral de las tecnologías fundamentales que intervienen en la comunicación, el almacenamiento y la identificación. Se examinarán tanto sus conceptos básicos como sus ventajas y desventajas de acuerdo con las exigencias particulares del proyecto. Asimismo, se revisarán las diversas opciones de hardware que facilitan la implementación de soluciones de Internet de las Cosas (IoT) a un precio más accesible, lo cual es un aspecto fundamental en este trabajo. En particular, se pondrá el foco en las alternativas económicas y con bajo consumo de energía, que constituyen una de las prioridades en el diseño y creación de los nodos de apoyo para las aulas en el contexto de este proyecto.

La implementación de estas soluciones comienza con un enfoque en la optimización del consumo energético de los nodos. Dado que estos dispositivos deben ser lo más autónomos posible, se prioriza la utilización de tecnologías que permitan su funcionamiento durante largos períodos sin necesidad de recargas frecuentes. Además, uno de los aspectos cruciales en esta fase es el coste por nodo. Por lo tanto, es fundamental seleccionar las tecnologías adecuadas para la comunicación inalámbrica entre nodos, pues muchas de las placas disponibles en el mercado ya integran módulos para diversas tecnologías de comunicación, lo que puede resultar en una reducción significativa del coste total del hardware. Por esto, se investigarán placas de reducido tamaño, bajo consumo energético y con precios asequibles.

## 3.1. Tecnologías de Comunicación Inalámbrica entre nodos

Para este proyecto barajamos distintas tecnologías de bajo consumo por su eficiencia a la hora de gestionar la red en el caso de que fuese necesario crear una red de nodos. En algunas de ellas, el alcance efectivo entre nodos para su correcta comunicación sin pérdida de datos y la eficiencia energética de cada una son factores determinantes. Uno de los puntos clave de este proyecto es contar con nodos que trabajen durante largos períodos sin necesidad de recargarlos ni que requieran mantenimiento.

### 3.1.1. LoRa

LoRa (Long Range) es una tecnología de comunicación inalámbrica adquirida por Semtech en 2012, compañía que perfeccionó los chips para dispositivos y gateways, promoviendo su expansión global, creada por Nicolas Sornin y Olivier Seller en Cycleo en 2008.[12]

Esta tecnología es utilizada sobre todo en aplicaciones que requieren comunicación de larga distancia. En estas aplicaciones, el gasto energético también es mínimo. Su implementación se facilita gracias a que LoRa opera principalmente en bandas de frecuencia no licenciadas, pero reguladas por zonas. Esta capacidad ha hecho posible su utilización exitosa en sectores tan diversos como agricultura inteligente, con sensores que sirven para monitorizar la humedad del suelo, ciudades inteligentes, con sistemas de iluminación urbana, y logística, que es aplicada al rastreo de contenedores y objetos.

Región/País	Banda(s) de Frecuencia LoRaWAN (MHz)
Europa	863–870
Estados Unidos, Canadá	902–928
China	470–510, 779–787
Japón	920–928
India	865–867
Australia, Nueva Zelanda	915–928
Corea del Sur	920–923

Tabla 3.1: Bandas de frecuencia LoRaWAN por región o país. [17].

Entre las principales ventajas que encontramos de LoRa, destacamos su alcance excepcional, que en condiciones óptimas es capaz de cubrir distancias de hasta 15 kilómetros[24]. Esto lo convirtió en una opción muy atractiva para nuestra aplicación, ya que simplificaba la necesidad de crear una red entre todos los nodos. De este modo, se evita tener los nodos operativos constantemente a la espera de recibir datos que tener que reenviar, aspecto importante ya que así podríamos escalar tantos nodos como quisiéramos sin la necesidad de ampliar ni configurar la red, puesto que cada nodo es independiente y se conecta con el servidor. Además de ello, su bajo consumo energético era uno de los puntos fuertes de esta tecnología, perfecto para dispositivos remotos alimentados por baterías, los cuales pueden funcionar durante largos periodos de tiempo. A nivel de seguridad, LoRa

también incorpora cifrado AES-128, lo que asegura que las comunicaciones sean seguras y protegidas contra posibles amenazas externas[17].

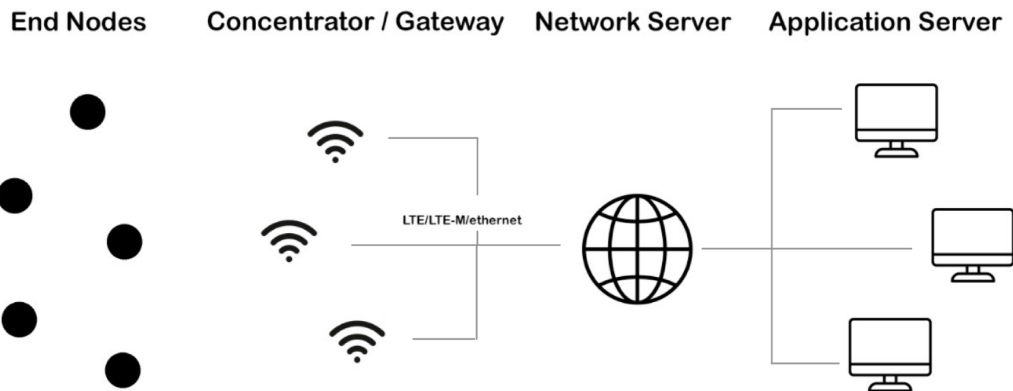


Figura 3.1: Esquema de la arquitectura LoRa[22]

No obstante, LoRa presenta algunas limitaciones que deben ser tomadas en cuenta a la hora de elegirla como solución para determinados proyectos. Una de las principales desventajas es su velocidad de transmisión, que es relativamente baja en comparación con otras tecnologías de comunicación inalámbrica. La velocidad máxima de transmisión de LoRa es de 50 kbps, lo que la hace poco adecuada para aplicaciones que requieren un gran volumen de datos o aquellas que necesitan comunicaciones en tiempo real. Esto puede ser un factor limitante en ciertos contextos donde se necesite transmitir información de manera rápida y constante, sin embargo, para nuestro nodo nos vale, ya que este transmite poca información de forma puntual para identificar el profesor y el aula donde se ubica el nodo. Además, LoRa depende de gateways para la conectividad, lo que implica que la infraestructura debe ser adecuada y suficiente para cubrir el área de implementación. En el caso de la facultad, aunque tiene muchas paredes y obstáculos, la conexión LoRa es capaz de cubrir toda la facultad.[3, 10]

Característica	Valor
Alcance	Entre 2 y 5 km en zonas urbanas
Velocidad de datos	0.3 kbps – 50 kbps
Consumo energético	Muy bajo (ideal para baterías)
Topología de red	Punto a Multipunto
Capacidad de dispositivos	Miles por gateway
Seguridad	AES de 128 bits

Tabla 3.2: Características clave de la tecnología LoRa

### 3.1.2. ZigBee

Zigbee nació a finales de la década de los años 90 como una iniciativa impulsada por varias grandes empresas tecnológicas como Philips, Motorola y Honeywell. El objetivo principal de este esfuerzo conjunto fue desarrollar un estándar inalámbrico que no solo fuera accesible en términos económicos, sino que también destacara por su eficiencia en

el consumo energético. Este enfoque resultó ser clave para el desarrollo de una tecnología útil en aplicaciones de bajo consumo y coste. En 2003 se constituyó la Zigbee Alliance, una organización que se encargó de supervisar y promover el crecimiento del estándar. Finalmente, en 2004, se lanzó la primera versión del protocolo Zigbee, marcando el inicio de su evolución dentro del mundo de las comunicaciones inalámbricas. [26]

En primer lugar, el alcance de Zigbee es relativamente largo en comparación a otras tecnologías, ya que su alcance es entre 10 y 100 metros, lo que lo hacía una gran opción a la hora de poder crear una red sin necesidad de nodos intermedios que actuaran como repetidores, sobre todo entre plantas de la facultad, pero no la suficiente como para poder usarse de punto a punto en muchos de los casos. Su velocidad de transferencia, aunque más alta que la de LoRa, llegando a los 250 kbps, sigue siendo una velocidad bastante baja entre todas las tecnologías propuestas; sin embargo, como ya comentamos antes, la baja velocidad de transferencia no desventaja en este proyecto debido a la poca información tratada y no requiere una transmisión constante. Otro aspecto desafiante era la configuración inicial de la red, que es más compleja en comparación con otras tecnologías inalámbricas.[19]

Una de las características que distingue a Zigbee y le otorga una ventaja considerable frente a otras tecnologías inalámbricas es su capacidad para formar redes en malla, también conocidas como redes mesh. Esta característica es fundamental, ya que, al permitir que los dispositivos puedan comunicarse a través de varios puntos intermedios, mejora notablemente la fiabilidad de la red. De esta forma, si algún dispositivo deja de funcionar o se desconecta, los datos pueden ser reenviados por otros dispositivos, garantizando que la comunicación no se interrumpa. Este punto fue muy atractivo a la hora de considerar una malla como método de comunicación entre los nodos que fue el plan inicial a la hora de definir el proyecto, sin embargo, eso producía que los nodos tuviesen que estar siempre activos, reduciendo la eficiencia energética del sistema.

A la hora de formar la red Zigbee, los dispositivos o nodos que la componen pueden asumir tres roles fundamentales, cada uno de los cuales tiene una función específica dentro del ecosistema Zigbee. El coordinador es el nodo central y único en la red, cuyo trabajo principal es crear y gestionar la red, enrutar los paquetes de datos y permitir nuevas conexiones de dispositivos. Generalmente, el coordinador está permanentemente encendido y, en muchos casos, conectado a Internet para poder enviar los datos recopilados a un servidor central. Por otro lado, los routers actúan como intermediarios, facilitando la transmisión de datos entre los dispositivos, aunque, a diferencia del coordinador, no tienen la capacidad de aceptar nuevos dispositivos en la red. Por último, los end devices o dispositivos finales son los nodos más simples, ya que solo pueden enviar y recibir datos, pero no tienen la capacidad de enrutar información y solo se pueden conectar a routers o al coordinador, lo que impedía implementar la idea inicial de una red sin nodos intermedios. [4]

### **3.1.3. Bluetooth**

Bluetooth fue desarrollado inicialmente por Ericsson en 1989 con el objetivo de crear auriculares inalámbricos. Este proyecto surgió como una respuesta a la necesidad de eliminar los cables de los dispositivos de audio, mejorando la experiencia de uso. Sin embargo,

Característica	Valor
Frecuencia	2.4 GHz (global), 868 MHz (Europa)
Velocidad de datos	Hasta 250 kbps
Alcance	10–100 m (según entorno)
Topología de red	Estrella, árbol, malla
Número de nodos soportados	Hasta 65.000 dispositivos
Consumo energético	Muy bajo (ideal para batería)
Seguridad	AES de 128 bits

Tabla 3.3: Características principales de la tecnología Zigbee

no fue sino hasta 1997, cuando IBM y Ericsson comenzaron a explorar la posibilidad de integrar esta tecnología en portátiles, que la visión de Bluetooth se expandió a otros dispositivos electrónicos. Decidieron convertir Bluetooth en un estándar abierto para permitir su adopción a nivel mundial. Para formalizar este proceso, en 1998 se fundó el Bluetooth Special Interest Group (Bluetooth SIG), una organización encargada de promover y desarrollar esta tecnología, con la participación de gigantes de la industria como Intel, Nokia y Toshiba.[25]

Una de las principales ventajas de Bluetooth es su bajo consumo energético, especialmente con la versión Bluetooth Low Energy (BLE), que está diseñada específicamente para aplicaciones que requieren un uso eficiente de la batería, punto por el cual fue una de las principales opciones para implementar en los nodos. Además, Bluetooth ofrece una amplia compatibilidad entre dispositivos de diferentes fabricantes, lo que facilita su integración en un ecosistema diverso como el que se estaba barajando y que el módulo ya está integrado en la placa del servidor y en las de los nodos sin requerir de módulos externos. La facilidad de configuración es otra de sus fortalezas ya que al ser una tecnología tan normalizada en todos los dispositivos existirían muchas librerías que simplificarían mucho la comunicación entre el sistema y la creación del código.

A pesar de estas ventajas, Bluetooth enfrenta ciertos desafíos que pueden limitar su rendimiento en algunas situaciones. Esta interferencia puede afectar la calidad de la conexión y la estabilidad de la transmisión de datos. Además, el alcance de Bluetooth es relativamente limitado en comparación con otras tecnologías de comunicación inalámbrica, lo que restringe su uso en aplicaciones que requieren conexiones a larga distancia. Esto sería solventable mediante su función de generar redes entre los nodos con su versión BLE Mesh, permitiendo mayor conectividad y alcance gracias a nodos intermedios, sin embargo, perderíamos cierta eficiencia energética al no poder dejar los nodos suspendidos por si se produjese una comunicación en la red.[6]

## 3.2. Tecnologías de registro e identificación

Las tecnologías de registro e identificación han transformado profundamente la manera en que interactuamos con los sistemas digitales en la actualidad. Estos avances han dado lugar a una variedad de métodos y enfoques que permiten autenticar identidades y otorgar acceso a diferentes servicios, cada uno con características particulares en cuanto a seguridad, facilidad de uso y fiabilidad. A lo largo de los años, hemos sido testigos de

Característica	Valor
Frecuencia	2.4 GHz (ISM)
Velocidad de datos	Hasta 2 Mbps
Alcance	Hasta 10 m
Número de dispositivos	Hasta 8 entre sí
Consumo energético	Muy bajo (ideal para baterías)
Seguridad	AES de 128 bits

Tabla 3.4: Características principales de la tecnología Bluetooth

una evolución continua en estas tecnologías, que han pasado de ser simples contraseñas a soluciones mucho más sofisticadas que emplean biometría, reconocimiento facial y otros métodos avanzados de verificación.

### 3.2.1. RFID

La tecnología Radio Frequency Identification (RFID) se basa en el uso de campos electromagnéticos para la identificación y seguimiento automático de tarjetas o etiquetas adheridas a objetos. A diferencia de otras tecnologías de comunicación de corto alcance, como Near Field Communication (NFC), RFID permite operar a distancias significativamente mayores, lo que amplía su ámbito de aplicación en entornos industriales y comerciales.

Existen dos categorías principales en los sistemas RFID: pasivos y activos. Los sistemas RFID pasivos no cuentan con fuente de alimentación propia, sino que dependen de la energía electromagnética emitida por el lector para activar el chip y transmitir la información almacenada, lo que sería las etiquetas de identificación. Esto limita su alcance operativo, aunque permite una mayor miniaturización y un costo de producción muy reducido. Por otro lado, los sistemas RFID activos incorporan una batería interna, lo que les permite emitir señales de forma autónoma y alcanzar distancias de lectura considerablemente mayores, a costa de un mayor tamaño y costo.

En cuanto a sus características, podemos destacar su eficiencia energética, ya que la frecuencia de operación de los módulos activos es modificable, lo cual, como ya hemos venido diciendo, es muy importante para este proyecto. Esto permite que podamos reducir su frecuencia, lo que resulta en un aumento considerable en la eficiencia, ya que las frecuencias altas requieren más energía para manejar las oscilaciones de señal, además de que se atenúan más rápidamente al atravesar obstáculos, lo que obliga a transmitir a más potencia para mantener una fiabilidad adecuada. La comunicación suele ser unidireccional, lo cual no representa un problema, ya que solo usaremos esta tecnología para leer las tarjetas. La velocidad varía en función de la frecuencia que se use, que ronda los 2 kbps. Aunque la velocidad es baja, es suficiente para nuestro caso, ya que el volumen de datos es mínimo y no requiere transmisión continua.

A pesar de sus múltiples ventajas —entre ellas, su bajo coste, alta fiabilidad y facilidad de integración en procesos industriales—, la tecnología RFID también plantea ciertos retos, especialmente en lo relativo a la privacidad y la seguridad. [11, 21]

<b>Característica</b>	<b>LF</b>	<b>HF</b>	<b>UHF</b>
Frecuencia	125–134 kHz	hasta 13.56 MHz	860–960 MHz
Velocidad de transferencia	2 kbps	hasta 424 kbps	hasta 640 kbps
Rango de lectura	hasta 10 cm	hasta 1 m	hasta 15 m
Eficiencia energética	Alta	Alta	Moderada
Materiales penetrables	Alto	Medio	Sensible
Interferencia electromagnética	Baja	Baja	Moderada

Tabla 3.5: Características principales de la tecnología RFID

### 3.2.2. NFC

La tecnología NFC, cuyas siglas corresponden a Near Field Communication (Comunicación de Campo Cercano), ha emergido como una de las innovaciones más relevantes en el ámbito de las comunicaciones inalámbricas de corto alcance. Esta tecnología permite el intercambio de datos entre dispositivos que se encuentran a una distancia extremadamente corta, generalmente de tan solo unos pocos centímetros, utilizando campos electromagnéticos para la transmisión de información. La principal ventaja de NFC radica en su simplicidad y velocidad de uso.

La tecnología NFC se caracteriza por su comunicación de corto alcance (hasta 10 cm), lo que proporciona alta seguridad al garantizar que solo los usuarios cercanos puedan ser identificados, lo cual sería útil en nuestro sistema de identificación para evitar accesos no autorizados. Además, su bajo consumo energético la hace ideal para tarjetas o dispositivos que no requieren batería, como las tarjetas inteligentes pasivas, lo que facilita la implementación sin necesidad de mantener los dispositivos cargados. La comunicación bidireccional entre el lector y el dispositivo también permitiría una interacción rápida y efectiva, mejorando la experiencia del usuario, mientras que su compatibilidad con RFID facilita la integración con sistemas existentes[2]. También, el sistema depende de que los dispositivos (smartphones o tarjetas) sean compatibles con NFC, lo que podría generar problemas si no todos los profesores tienen dispositivos habilitados para NFC. Finalmente, aunque NFC es seguro y fácil de usar, su implementación puede ser costosa frente a opciones como RFID[2, 16]

<b>Característica</b>	<b>Valor</b>
Frecuencia	13.56 MHz
Velocidad de transferencia	106 - 424 kbps
Rango de lectura	Hasta 10 cm (óptimo: 4–5 cm)
Eficiencia energética	Alta
Tipo de comunicación	Bidireccional
Seguridad	AES de 128 bits, 3DES o DES 56 bits
Interferencia electromagnética	Baja

Tabla 3.6: Características principales de la tecnología NFC

### 3.2.3. Biometría Dactilar

Los sistemas de reconocimiento de huellas dactilares constituyen una de las tecnologías biométricas más consolidadas y utilizadas en la actualidad. Su funcionamiento se basa en el análisis de características únicas de cada individuo, tales como las minucias[14] —que comprenden bifurcaciones, terminaciones de crestas y otros puntos singulares—, así como los patrones generales de las crestas papilares presentes en las yemas de los dedos.

Gracias a su precisión y relativa facilidad de implementación, esta tecnología ha sido ampliamente adoptada en diversos ámbitos. Entre sus aplicaciones más comunes se encuentra el desbloqueo de dispositivos electrónicos personales, como teléfonos móviles y ordenadores portátiles, donde ofrece un equilibrio adecuado entre seguridad y usabilidad. Asimismo, se emplea en sistemas de control de acceso físico a instalaciones sensibles o restringidas, así como en sistemas de registro de asistencia laboral, donde proporciona una forma eficiente y segura de verificar la identidad de los empleados.

No obstante, a pesar de su alta fiabilidad en condiciones ideales, los sistemas basados en huellas dactilares pueden presentar ciertas limitaciones. Factores como la humedad, la suciedad o la presencia de lesiones en los dedos pueden dificultar la captura adecuada de la huella, provocando errores de autenticación o falsos rechazos. Este tipo de inconvenientes resalta la importancia de complementar este tipo de soluciones con tecnologías biométricas adicionales o sistemas multifactor, especialmente en contextos donde se requiera un alto nivel de seguridad y disponibilidad.[7, 8, 18]

Característica	Valor
<b>Velocidad</b>	1-2 segundos por identificación
<b>Requerimientos de hardware</b>	Sensor óptico, capacitivo o ultrasónico
<b>Falsos negativos</b>	Menor al 1 %
<b>Falsos positivos</b>	Menor al 0.1 %
<b>Costo</b>	10 - 100 EUR
<b>Seguridad</b>	Alta, huellas únicas
<b>Eficiencia energética</b>	Alta

Tabla 3.7: Características principales de la biometría dactilar

### 3.3. Alternativas de Placas de Bajo Coste con Capacidad IoT

La integración de dispositivos del Internet of Things (IoT) con servidores de bajo coste constituye una solución tecnológica cada vez más extendida, especialmente en entornos donde se requiere realizar procesamiento local, gestionar grandes volúmenes de datos provenientes de sensores, y coordinar la comunicación bidireccional con múltiples actuadores. Este enfoque permite descentralizar ciertas tareas computacionales, reducir la latencia en la toma de decisiones y disminuir la dependencia de servicios en la nube, lo cual puede ser especialmente beneficioso en aplicaciones críticas o en ubicaciones con conectividad limitada.

Este tipo de arquitectura se emplea habitualmente en sistemas embebidos, redes de sensores distribuidos, domótica, automatización industrial y entornos educativos, donde los costes de implementación y mantenimiento son factores determinantes. La elección de servidores de bajo consumo y reducido coste, como placas de desarrollo tipo Raspberry Pi, BeagleBone o ESP32, posibilita la construcción de sistemas escalables y modulares que combinan eficiencia energética, flexibilidad y capacidad de adaptación a distintas necesidades.

A continuación, se analizan diversas alternativas para la implementación de este tipo de soluciones, teniendo en cuenta criterios como el rendimiento computacional, la conectividad, la compatibilidad con periféricos, la facilidad de programación y la disponibilidad de recursos en entornos de desarrollo abiertos.

### 3.3.1. Raspberry Pi

La Raspberry Pi se ha consolidado como una de las plataformas de referencia en el ámbito del Internet of Things (IoT), gracias a su notable versatilidad, su facilidad de uso y el respaldo de una comunidad global activa que genera abundante documentación, tutoriales y soporte técnico. Esta popularidad la convierte en una opción altamente accesible tanto para entornos educativos como para desarrollos profesionales e industriales de pequeña y mediana escala.

Uno de los principales puntos fuertes de la Raspberry Pi es su compatibilidad con múltiples sistemas operativos, entre los cuales destacan Raspberry Pi OS —una distribución optimizada basada en Debian— y otras alternativas como Ubuntu, Arch Linux o sistemas orientados a tareas específicas, como Home Assistant OS o Kali Linux. Esta flexibilidad permite adaptar la plataforma a un amplio rango de aplicaciones, desde servidores locales hasta nodos de procesamiento en sistemas distribuidos.

En términos de expansión funcional, la Raspberry Pi dispone de una extensa gama de módulos de ampliación denominados HATs (Hardware Attached on Top), que facilitan la incorporación de nuevas capacidades como pantallas táctiles, interfaces de comunicación industrial (CAN, RS-485), controladores de motores, módulos GPS o sistemas de almacenamiento adicionales. Esta modularidad es especialmente útil para el desarrollo de prototipos personalizados en entornos IoT heterogéneos.

Asimismo, la presencia de pines GPIO (General Purpose Input/Output) permite la conexión directa con sensores y actuadores, lo que simplifica la creación de interfaces físicas con el entorno. Esta característica, combinada con la conectividad inalámbrica integrada —Wi-Fi y Bluetooth en las versiones más recientes, como la Raspberry Pi 4 y Raspberry Pi 400—, proporciona una solución integral para el diseño de nodos inteligentes que pueden operar de forma autónoma o en red.

En definitiva, la Raspberry Pi representa una plataforma altamente competente para proyectos IoT, permitiendo a los desarrolladores implementar soluciones funcionales, económicas y escalables con un bajo umbral de entrada en cuanto a complejidad técnica.[23]

### 3.3.2. ODROID

Las placas ODROID, desarrolladas por la empresa surcoreana Hardkernel, representan una alternativa robusta y de alto rendimiento dentro del ecosistema de plataformas para aplicaciones IoT. Estas placas se caracterizan por incorporar procesadores potentes, como los de la familia Amlogic, que les confieren una capacidad de procesamiento significativamente superior a la de otras soluciones de bajo coste, como la Raspberry Pi. Esta potencia las convierte en una opción idónea para escenarios donde se requieren operaciones computacionalmente intensivas, como el procesamiento de grandes volúmenes de datos, la ejecución de algoritmos de inteligencia artificial o el control simultáneo de múltiples dispositivos y sensores.

Un ejemplo destacado es el modelo ODROID-N2+, que incorpora un procesador Amlogic S922X con arquitectura big.LITTLE de seis núcleos (cuatro Cortex-A73 y dos Cortex-A53), lo que le permite manejar tareas exigentes con eficiencia energética. Además, este modelo cuenta con una excelente gestión térmica gracias a su diseño pasivo optimizado, lo que favorece un funcionamiento estable incluso en entornos con alta carga sostenida.

Sin embargo, una de las principales limitaciones de la línea ODROID es la ausencia de conectividad inalámbrica integrada en la mayoría de sus modelos, incluyendo el ODROID-N2+. Específicamente, no disponen de Wi-Fi ni Bluetooth incorporados, lo cual obliga a los desarrolladores a adquirir e integrar módulos externos mediante interfaces USB si se requiere dicha funcionalidad. Esta carencia puede suponer un inconveniente en proyectos donde la comunicación inalámbrica es un requisito esencial o donde se pretende reducir el número de componentes adicionales.

A pesar de esta limitación, las placas ODROID ofrecen un excelente equilibrio entre rendimiento, estabilidad y soporte para sistemas operativos avanzados como Ubuntu, Debian y Android, lo que las posiciona como una opción muy competitiva para la construcción de servidores locales o gateways IoT de alto rendimiento.[9]

### 3.3.3. Espressif ESP32

La placa ESP32 es una de las placas de desarrollo más conocidas en el campo de la electrónica y el IoT y fue desarrollada por Espressif Systems. Existen numerosos modelos, que se adaptan a distintas necesidades dependiendo la conectividad, rendimiento y consumo que necesite el proyecto. Entre los modelos más comunes se encuentran los modelos ESP32-S2, que destaca por su orientación a aplicaciones seguras con conectividad Wi-Fi y capacidad de comunicación USB OTG, y ESP32-C6, una versión más reciente que introduce WiFi 6, Bluetooth 5 LE, así como protocolos de red como Zigbee, convirtiéndolo en una opción muy potente y moderna para dispositivos conectados.

La ESP32-S2 cuenta con un núcleo Xtensa LX7 de alto rendimiento y conectividad USB, características ideales para aplicaciones que requieren una comunicación directa con ordenadores u otros dispositivos USB.

Por otro lado, la ESP32-C6 se presenta como una evolución enfocada al bajo consumo energético y mejora en la eficiencia de red, destacándose especialmente en entornos IoT de

nueva generación gracias a su conectividad avanzada. Esta variante adopta una arquitectura basada en RISC-V, y está diseñada específicamente para integrarse en aplicaciones modernas que requieren comunicación inalámbrica y eficiencia energética.

También se destaca por su amplia gama de pines GPIO, que permiten la interfaz directa con una variedad de sensores, actuadores y otros periféricos. También tiene ADC, DAC, puertos de comunicación como SPI, I2C, UART, salidas PWM y sensores táctiles capacitivos, lo que lo hace extremadamente versátil para aplicaciones de control y monitoreo. La mayoría de los modelos del ESP32 tienen memoria flash incorporada para almacenamiento; en algunos modelos, esto se complementa con ranuras para tarjetas microSD.

Un gran atractivo de la ESP32 radica en su bajo consumo energético, dado que cuenta con varios modos de ahorro de energía, lo que resulta útil para proyectos que funcionan con batería o requieren un funcionamiento prolongado sin necesidad de mantenimiento. La placa también tiene un soporte de software bastante amplio que permite programarla en varios entornos como ESP-IDF (framework oficial de Espressif) y Arduino IDE, entre otros.

Placa	Procesador	RAM	Memoria Incorporada	Módulos Integrados	Rango de Precios
Raspberry Pi	ARM Cortex-A72, 1.5 GHz (Quad Core)	2GB 4GB 8GB	No	Wi-Fi, Bluetooth	35-75 €
ODROID-N2+	Amlogic S922X, 2.2 GHz (Hexa Core)	4GB 8GB	No	Ninguno (requiere módulos externos)	80-100 €
ESP32-S2	Xtensa LX7 Single Core, 240 MHz	320 kB SRAM	4MB Flash	Wi-Fi 2.4GHz, USB OTG	6-12 €
ESP32-C6	RISC-V 32-bit Single Core, 160 MHz	512 kB SRAM	4-8MB Flash	Wi-Fi 6, Bluetooth 5 (LE), Zigbee	6-14 €

Tabla 3.8: Comparativa de placas de desarrollo

## 3.4. Web/Server

### 3.4.1. Tecnologías para desarrollo web

El desarrollo web se basa en una combinación de tecnologías fundamentales que permiten la creación de sitios y aplicaciones web dinámicas, la cual necesitamos para poder mostrar al administrador sin conocimiento en BBDD todo lo recopilado por el sistema. Entre las más utilizadas se encuentran HTML, JavaScript y PHP. HTML define la estructura y el contenido de las páginas web, JavaScript añade interactividad y dinamismo en el lado del cliente, mientras que PHP se encarga de la lógica del servidor y la gestión

de bases de datos. Estas tecnologías trabajan juntas para proporcionar experiencias web completas y funcionales.

## HTML

HTML (*HyperText Markup Language*) es el lenguaje estándar usado para estructurar y definir el contenido de las páginas web. Su función principal es organizar el contenido mediante etiquetas que representan diversos elementos, como párrafos, encabezados, tablas, listas, imágenes, enlaces y formularios. Cada una de estas etiquetas proporciona un formato específico para el contenido, permitiendo que el navegador interprete y muestre el texto de manera estructurada y comprensible para el usuario.

Una de las ventajas más destacadas de HTML es su compatibilidad universal. Gracias a su naturaleza estándar, HTML funciona en todos los navegadores web y dispositivos, lo que garantiza que las páginas web sean accesibles desde diferentes plataformas sin problemas de compatibilidad.

Aunque HTML es esencial para la creación de páginas web estáticas y la estructuración del contenido, por sí solo no es suficiente para proporcionar interactividad avanzada. Para ello, se combina habitualmente con JavaScript, un lenguaje de programación que amplía las capacidades de HTML mediante la interacción con el usuario, la validación de formularios y la manipulación dinámica del contenido.

## JavaScript

JavaScript es un lenguaje de programación dinámico y versátil que juega un papel fundamental en el desarrollo web moderno, especialmente en lo que respecta a la creación de páginas interactivas y dinámicas. A diferencia de otros lenguajes de programación que se ejecutan en el servidor, JavaScript se ejecuta directamente en el navegador del cliente, lo que permite reducir la carga en el servidor y mejorar la velocidad de ejecución al realizar tareas directamente en el navegador sin necesidad de comunicarse continuamente con el servidor. Esta característica es especialmente beneficiosa en aplicaciones que requieren una rápida interacción con el usuario, como formularios interactivos, juegos en línea y aplicaciones de procesamiento en tiempo real.

Una de las capacidades más poderosas de JavaScript es su compatibilidad con otros lenguajes y tecnologías, lo que le permite integrarse perfectamente en aplicaciones web complejas. A menudo se utiliza junto con PHP, el lenguaje del lado del servidor, para crear aplicaciones web dinámicas que combinan la lógica del servidor con la interacción del lado del cliente.

Además de su uso en el navegador, JavaScript también se ha expandido al lado del servidor mediante tecnologías como Node.js. Node.js permite a los desarrolladores utilizar JavaScript para construir aplicaciones del lado del servidor, lo que mejora la eficiencia al permitir el uso de un solo lenguaje tanto para el cliente como para el servidor. Esto ha llevado a una mayor unificación en el desarrollo web, reduciendo la complejidad al eliminar la necesidad de dominar múltiples lenguajes de programación. [1]

## PHP

PHP (*Hypertext Preprocessor*) es el lenguaje de programación del lado del servidor que se ha seleccionado para el desarrollo de la parte dinámica de nuestra aplicación web. Este lenguaje se caracteriza por su capacidad para generar contenido en tiempo real según las interacciones del usuario o los datos obtenidos de bases de datos, lo que nos permitirá crear una experiencia web personalizada y dinámica.

Una de las principales ventajas de PHP es su fácil integración con HTML mediante etiquetas específicas (`<?php ?>`), lo que facilita la creación de páginas web dinámicas. Gracias a esta integración, podremos combinar la estructura estática de HTML con la capacidad de PHP para generar contenido dinámico, lo que optimiza el desarrollo y la organización del código.

PHP es una tecnología multiplataforma, lo que significa que puede funcionar en diferentes sistemas operativos como Windows, Linux y macOS, y se puede desplegar en servidores web comunes como Apache o Nginx. Esto nos asegura una gran flexibilidad al momento de implementar la solución, permitiendo su ejecución en diversas plataformas sin perder rendimiento.

La capacidad de PHP para interactuar de forma eficiente con bases de datos como MySQL, PostgreSQL o SQLite es otra de sus grandes ventajas. Dado que nuestro proyecto requiere un manejo eficaz de los datos, PHP se presenta como la herramienta idónea para gestionar la información de manera rápida y segura. Además, su integración con estas bases de datos nos permitirá crear, leer, actualizar y eliminar datos en tiempo real, lo que es fundamental para la funcionalidad dinámica de la aplicación.

En conclusión, PHP es la tecnología más adecuada para este proyecto debido a su capacidad para generar contenido dinámico, interactuar con bases de datos de manera eficiente y su compatibilidad multiplataforma. Con PHP, podremos desarrollar una aplicación web sólida, escalable y con una excelente experiencia de usuario. [15]

<b>Tecnología</b>	<b>Tipo</b>	<b>Uso Principal</b>	<b>Ejecución</b>	<b>Ejemplos Clave</b>
HTML	Lenguaje de marcado	Estructura y contenido estático	Cliente	Páginas estáticas, formularios
JavaScript	Lenguaje de programación	Interactividad y dinamismo	Cliente	Validación, animaciones
PHP	Lenguaje de lado servidor	Contenido dinámico basado en datos	Servidor	Páginas dinámicas

Tabla 3.9: Diferentes tecnologías Web y sus características

# Capítulo 4

## Implementación

Después de lo visto en el capítulo anterior, podemos identificar los elementos fundamentales necesarios para la implementación de este proyecto. En primer lugar, son necesarios varios nodos autónomos colocados en cada aula. Cada uno de estos tendrá una batería recargable, un lector RFID para la identificación, un módulo LoRa para la comunicación inalámbrica y un microcontrolador ESP32-C6 que actúa como unidad central de procesamiento. El diseño de estos nodos prioriza la eficiencia energética. Estos nodos se encargan de recoger los datos de asistencia mediante el lector RFID y transmitir estos datos de forma inalámbrica mediante el módulo LoRa.

El nodo central está constituido por una Raspberry Pi 5, la cual está conectada a la red eléctrica y recibe las señales inalámbricas mediante el módulo LoRa. Este cumple dos funciones: recibir los datos del resto de nodos y, como servidor central, alojar la base de datos y la aplicación web para la administración. Todo esto se ve ejemplificado en la Figura 4.1

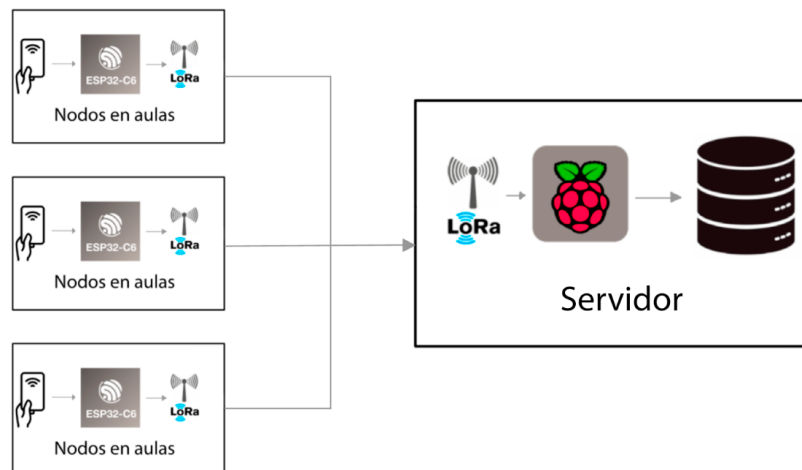


Figura 4.1: Esquema del sistema.

## 4.1. Desarrollo Servidor

Las tecnologías usadas son CSS, JavaScript y PHP, ya que son las aprendidas en el grado.

El desarrollo del servidor se ha realizado utilizando PHP, el cual es capaz de interactuar con la base de datos y generar dinámicamente los contenidos que ve el usuario. Esto permite implementar funciones necesarias como la autenticación de usuarios, la gestión de registros académicos y la automatización de procesos administrativos, como la generación diaria de asistencias y el control de incidencias. La elección de estas tecnologías se debe a que son las aprendidas durante el grado.

### 4.1.1. Base de Datos

Como se muestra en Figura 4.2, esta base de datos está diseñada para gestionar un entorno educativo, permitiendo el control y seguimiento de todos los elementos necesarios para la organización académica, los cuales serían la capacidad de guardar profesores, qué días son las clases, en qué aulas son y un control de quién ha dado la clase y quién no para ponerlo como incidencia. A continuación, se explica para qué sirve cada tabla:

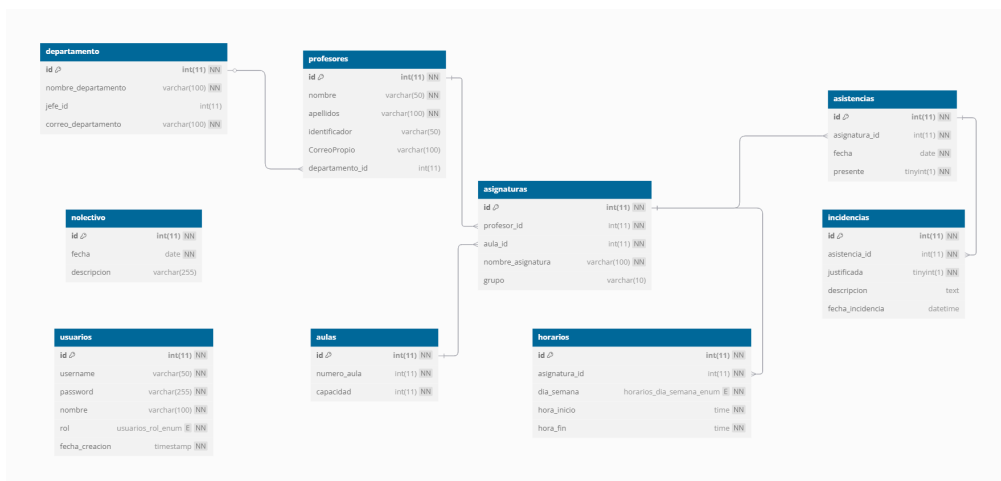


Figura 4.2: Estructura de la base de datos.

- **PROFESORES:** Almacena la información de los docentes que trabajan en la institución. Es fundamental para tener un registro centralizado del personal académico, permitiendo asignarles asignaturas y vincularlos a departamentos específicos.
- **DEPARTAMENTO:** Gestiona las unidades académicas o áreas de conocimiento en las que se organiza la institución. Cada departamento puede tener múltiples profesores asociados y un profesor designado como jefe, lo que facilita la organización jerárquica y administrativa.
- **AULAS:** Contiene la información sobre los espacios físicos donde se imparten las clases. Esto permite gestionar la capacidad y disponibilidad de los espacios docentes para la planificación de horarios y asignación de grupos.
- **ASIGNATURAS:** Representa las materias que se imparten en la institución. Sirve como nexo central, relacionando profesores con aulas específicas y organizando los contenidos académicos en grupos de estudio.
- **HORARIOS:** Gestiona la programación temporal de las asignaturas. Permite establecer cuándo se imparte cada asignatura, controlando días de la semana y horas de inicio y fin para evitar solapamientos y optimizar recursos.
- **ASISTENCIAS:** Registra la presencia de los estudiantes en las clases. Es esencial para el seguimiento académico, controlando la participación y asistencia a las diferentes asignaturas programadas.
- **INCIDENCIAS:** Documenta situaciones especiales relacionadas con la asistencia. Permite registrar y justificar ausencias o problemas que afecten al normal desarrollo de las clases.
- **NOLECTIVO:** Registra los días sin actividad académica como festivos, vacaciones o días de formación. Es crucial para la correcta planificación del calendario académico y gestión de horarios.
- **USUARIOS:** Almacena los diferentes datos para el inicio de sesión de la aplicación, así como datos de cada usuario como el nombre y el rol que se le ha asignado

En MySQL, esta estructura permite implementar un sistema completo de gestión académica con capacidad para consultas complejas que facilitan la toma de decisiones administrativas, optimización de recursos y seguimiento docente.

#### 4.1.2. Proceso Automático

Uno de los objetivos de este proyecto es reducir muchas de las acciones repetitivas y mecánicas que se realizan en este proceso. Para ello se ha ideado un sistema que genere de forma diaria las asistencias de todas las clases, antes eran las hojas de firma que se colocaban en cada clase, y que se generen las incidencias sin necesidad de verificar la asistencia uno por uno.

## Proceso diario automático

El diagrama que se muestra en la Figura 4.3 comienza con un proceso automático que realiza el sistema a diario. Este proceso tiene como objetivo gestionar los horarios y determinar si se trata de un día lectivo o no. En primer lugar, el sistema consulta a la Base de Datos para obtener los horarios del día. También solicita una lista de los días no lectivos (por ejemplo, festivos o vacaciones). La Base de datos devuelve la información solicitada al sistema.

Posteriormente, el sistema utiliza esta información para verificar si el día actual es lectivo. Esta verificación es crucial para decidir las acciones que se deben tomar a continuación.

### Decisión: Día lectivo o no lectivo

En este punto, el flujo del proceso se divide en dos posibles caminos:

- **Si es un día lectivo:** El sistema genera automáticamente registros de asistencia para las clases programadas ese día. Esto asegura que haya un control inicial sobre las asistencias y ausencias.
- **Si no es un día lectivo:** El sistema simplemente no genera registros de asistencia, ya que no hay actividades académicas programadas.

Esta lógica permite optimizar los recursos del sistema y evitar procesar datos innecesarios en días no lectivos.

### Proceso de registro de asistencia

En este apartado del diagrama entra en juego la interacción del Usuario (por ejemplo, un profesor o administrador). Durante el transcurso del día, el Usuario marca las asistencias en clase utilizando el sistema. Una vez registradas, el sistema actualiza la información correspondiente en la Base de Datos.

### Gestión de ausencias

Cuando se detectan ausencias tras registrar las asistencias, el sistema genera automáticamente incidencias relacionadas con las ausencias detectadas. Estas incidencias incluyen detalles como el nombre del profesor, la clase en cuestión y el aula.

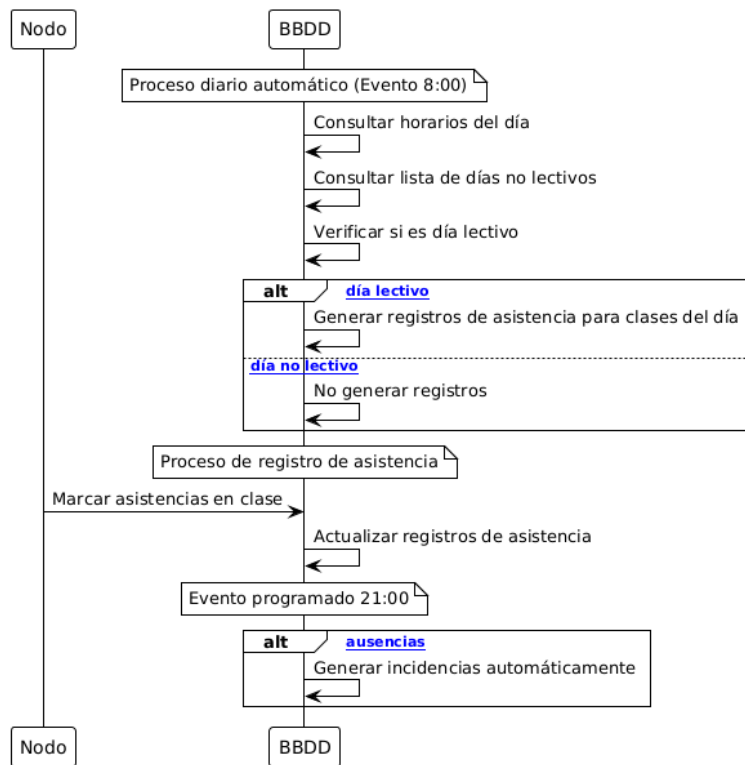


Figura 4.3: Servicios automáticos

### 4.1.3. Funcionalidades implementadas

#### Inicio de Sesión

La primera interacción del usuario es la ventana de inicio de sesión, como se puede ver en la Figura 4.4, la cual es la capa de seguridad que tiene nuestra aplicación, donde hay que poner el nombre de usuario y la contraseña que se guarda como *hash* en la base de datos. Si el usuario no existe o la contraseña está mal puesta, muestra un error, pero si todo es correcto, ésta lleva al menú principal.

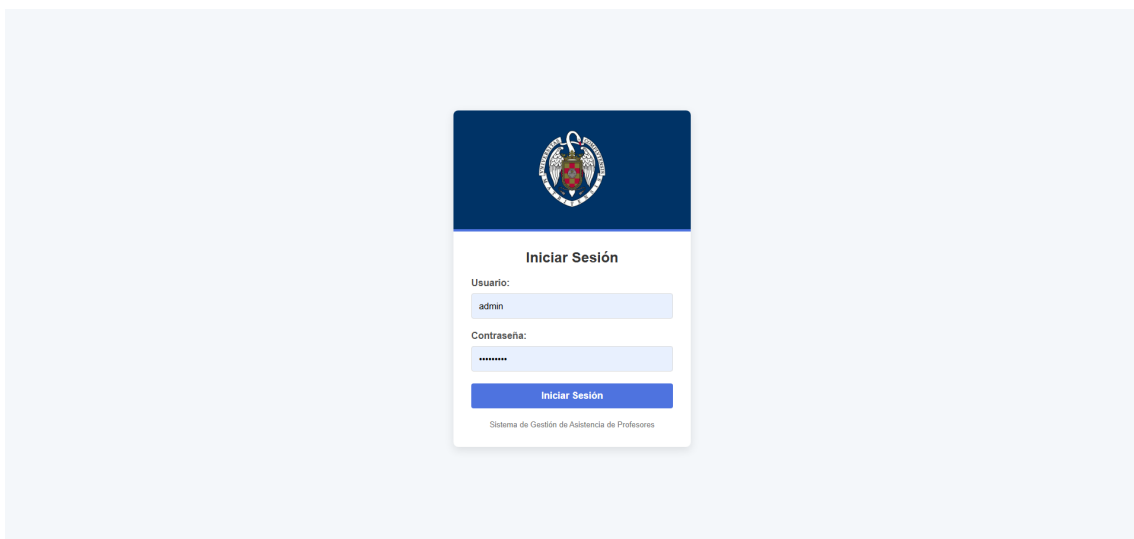


Figura 4.4: Ventana de inicio de sesión

## Menú principal

Una vez hecho el inicio de sesión, como se muestra en la Figura 4.5 se muestra esta pantalla donde podemos ver a la izquierda un menú que estará en todas las ventanas de la aplicación, la cual es un acceso rápido a todas las partes más importantes de la aplicación. Esta ventana, al ser la principal, te lleva a los accesos que más se pueden utilizar como primera vez.

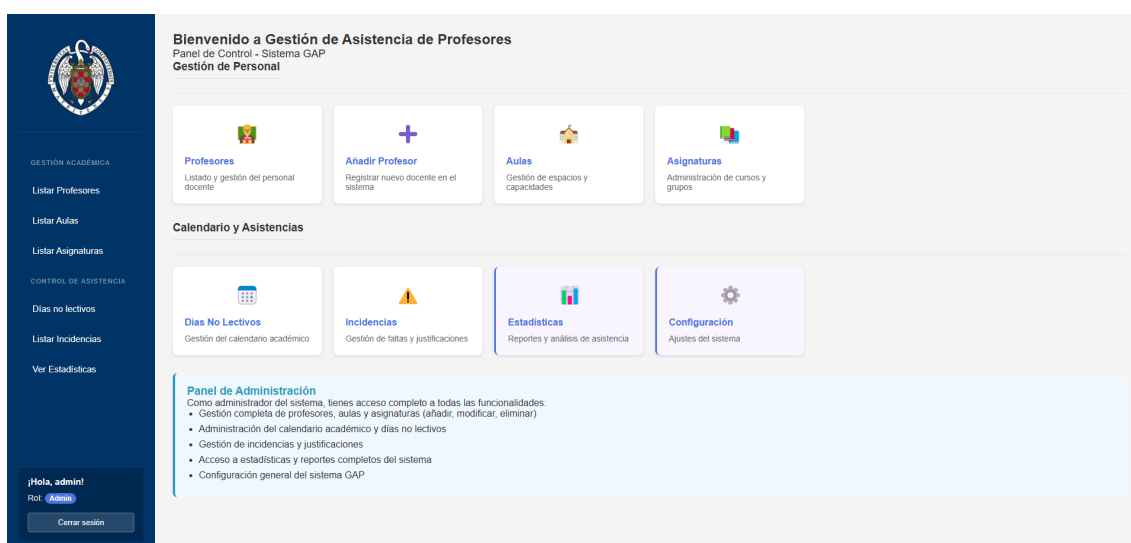


Figura 4.5: Ventana principal del sistema

## Listado de Profesores

La interfaz de Listado de Profesores, como demuestra la Figura 4.6, presenta una tabla interactiva que muestra de manera clara y ordenada la información esencial de cada docente. En cada fila se pueden consultar los datos personales del profesor, como su nombre y apellidos.

Para facilitar la gestión y la búsqueda de información, la tabla permite ordenar los registros por cualquiera de sus columnas y dispone de filtros intuitivos que agilizan la localización de profesores según su nombre o apellidos. Además, la interfaz incorpora botones de acción que permiten añadir nuevos profesores al sistema, editar la información de los existentes o eliminar registros, siempre solicitando una confirmación previa antes de realizar cualquier acción irreversible. De este modo, se garantiza una administración eficiente y segura del personal docente.

**Profesores**  
 Profesor > Listado de todos los Profesores

Buscar por nombre o apellido

Nombre	Apellidos	Acción
Raquel	Díaz Sánchez	<input type="button" value="Ver"/> <input type="button" value="Modificar"/> <input type="button" value="Eliminar"/>
Javier	Fernández González	<input type="button" value="Ver"/> <input type="button" value="Modificar"/> <input type="button" value="Eliminar"/>
Eva	Fernández Sánchez	<input type="button" value="Ver"/> <input type="button" value="Modificar"/> <input type="button" value="Eliminar"/>
Ángel	Gallego Muñoz	<input type="button" value="Ver"/> <input type="button" value="Modificar"/> <input type="button" value="Eliminar"/>
Alberto	García Martínez	<input type="button" value="Ver"/> <input type="button" value="Modificar"/> <input type="button" value="Eliminar"/>
David	García Sánchez	<input type="button" value="Ver"/> <input type="button" value="Modificar"/> <input type="button" value="Eliminar"/>
Miguel	González García	<input type="button" value="Ver"/> <input type="button" value="Modificar"/> <input type="button" value="Eliminar"/>
Marta	González Hernández	<input type="button" value="Ver"/> <input type="button" value="Modificar"/> <input type="button" value="Eliminar"/>
Ana	González Pérez	<input type="button" value="Ver"/> <input type="button" value="Modificar"/> <input type="button" value="Eliminar"/>
Daniel	Hernández González	<input type="button" value="Ver"/> <input type="button" value="Modificar"/> <input type="button" value="Eliminar"/>
Laura	Hernández López	<input type="button" value="Ver"/> <input type="button" value="Modificar"/> <input type="button" value="Eliminar"/>
Carlos	López García	<input type="button" value="Ver"/> <input type="button" value="Modificar"/> <input type="button" value="Eliminar"/>

« Primera < Anterior 1 2 > Siguiente » Última »  
 Mostrando 12 de 24 registros (Página 1 de 2)

Figura 4.6: Listado de profesores registrado en el sistema

## Datos del profesor

En esta ventana se permite gestionar la información de los profesores registrados, tal como se observa en la Figura 4.7. La interfaz muestra datos personales como nombre, apellidos, identificador, correo electrónico y otros datos de contacto. En la parte inferior hay un horario semanal (de lunes a viernes) donde se pueden asignar o visualizar las horas de clase del profesor. El horario está organizado por horas y días de la semana. La ventana también incluye botones de acción como Volver, Modificar y Eliminar para gestionar los registros.

**Profesores**  
 Datos de Profesor

Nombre:

Apellidos:

Identificador:

Correo Propio:

Departamento:

Correo Departamento:

**PRIMER CUATRIMESTRE**

	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES
9					
10					
11					
12					
13					
14					
15					FAL (4P) Aula 101
16					
17					SO (4P) Aula 101
18					
19					
20					

Figura 4.7: Listado de profesores registrado en el sistema

## Listado de Aulas

La Figura 4.8 muestra cómo proporciona información detallada sobre los espacios disponibles para la impartición de clases, mostrando para cada aula su número o identificador y la capacidad en cuanto al número de estudiantes que puede albergar.

El listado está diseñado para facilitar la gestión y búsqueda de aulas, permitiendo filtrar los resultados según su identificador.

Número de Aula	Capacidad	Acción
101	24	<a href="#">Ver datos</a>   <a href="#">Modificar</a>   <a href="#">Eliminar</a>
102	25	<a href="#">Ver datos</a>   <a href="#">Modificar</a>   <a href="#">Eliminar</a>
103	40	<a href="#">Ver datos</a>   <a href="#">Modificar</a>   <a href="#">Eliminar</a>
104	30	<a href="#">Ver datos</a>   <a href="#">Modificar</a>   <a href="#">Eliminar</a>
105	25	<a href="#">Ver datos</a>   <a href="#">Modificar</a>   <a href="#">Eliminar</a>
106	35	<a href="#">Ver datos</a>   <a href="#">Modificar</a>   <a href="#">Eliminar</a>
201	40	<a href="#">Ver datos</a>   <a href="#">Modificar</a>   <a href="#">Eliminar</a>
202	45	<a href="#">Ver datos</a>   <a href="#">Modificar</a>   <a href="#">Eliminar</a>
203	30	<a href="#">Ver datos</a>   <a href="#">Modificar</a>   <a href="#">Eliminar</a>
204	35	<a href="#">Ver datos</a>   <a href="#">Modificar</a>   <a href="#">Eliminar</a>
205	28	<a href="#">Ver datos</a>   <a href="#">Modificar</a>   <a href="#">Eliminar</a>
301	50	<a href="#">Ver datos</a>   <a href="#">Modificar</a>   <a href="#">Eliminar</a>

Figura 4.8: Listado de aulas registrado en el sistema

## Listado de Asignaturas

La ventana de Listado de Asignaturas, como se queda reflejado en la Figura 4.9 ofrece una visión clara y estructurada de todas las asignaturas incluidas en el plan de estudios. Para cada materia, se muestran datos relevantes como el nombre de la asignatura, el curso, el aula donde se imparte y el profesor que la imparte.

Esta interfaz no solo facilita la consulta de información, sino que también permite gestionar de manera eficiente la asignación y reasignación de profesores a las distintas asignaturas, contribuyendo así a una administración académica más ágil y flexible.

**Asignaturas**  
Listado de todas las asignaturas registradas.

Buscar por nombre de asign | Buscar por profesor | Buscar por aula |

Nombre Asignatura	Aula	Grupo	Profesor	Acciones
Álgebra	102	1ºB	Carlos López García	<input type="button" value="Ver"/>   <input type="button" value="Modificar"/>   <input type="button" value="Eliminar"/>
Algoritmos	105	2ºA	Laura Hernández López	<input type="button" value="Ver"/>   <input type="button" value="Modificar"/>   <input type="button" value="Eliminar"/>
Aprendizaje Automático	203	3ºA	Marina Pérez García	<input type="button" value="Ver"/>   <input type="button" value="Modificar"/>   <input type="button" value="Eliminar"/>
Bases de Datos	106	2ºB	Javier Fernández González	<input type="button" value="Ver"/>   <input type="button" value="Modificar"/>   <input type="button" value="Eliminar"/>
Big Data	101	4ºA	Marta González Hernández	<input type="button" value="Ver"/>   <input type="button" value="Modificar"/>   <input type="button" value="Eliminar"/>
Cálculo	103	1ºA	Elena Martínez Rodríguez	<input type="button" value="Ver"/>   <input type="button" value="Modificar"/>   <input type="button" value="Eliminar"/>
Compiladores	204	4ºB	Jorge Pérez Hernández	<input type="button" value="Ver"/>   <input type="button" value="Modificar"/>   <input type="button" value="Eliminar"/>
Computación Gráfica	201	4ºC	Eva Fernández Sánchez	<input type="button" value="Ver"/>   <input type="button" value="Modificar"/>   <input type="button" value="Eliminar"/>
Desarrollo Web	102	2ºA	Diego López Pérez	<input type="button" value="Ver"/>   <input type="button" value="Modificar"/>   <input type="button" value="Eliminar"/>
Estadística	101	1ºC	Sara Rodríguez Fernández	<input type="button" value="Ver"/>   <input type="button" value="Modificar"/>   <input type="button" value="Eliminar"/>
Estructura de Computadores	103	2ºC	Carmen López Rodríguez	<input type="button" value="Ver"/>   <input type="button" value="Modificar"/>   <input type="button" value="Eliminar"/>
Fal	101	4ºB	Ángel Gallego Muñoz	<input type="button" value="Ver"/>   <input type="button" value="Modificar"/>   <input type="button" value="Eliminar"/>

« Primera < Anterior 1 2 Siguiente > Última »  
Mostrando 12 de 24 registros (Página 1 de 2)

Figura 4.9: Listado de asignaturas registrado en el sistema

## Gestión Días No Lectivos

Esta ventana, como se puede ver en la Figura 4.10 permite administrar fechas especiales mediante filtros temporales, así como permite al administrador gestionar de manera eficiente los días festivos a lo largo del curso académico, facilitando la marcación de jornadas no lectivas individuales directamente sobre un calendario visual integrado en la interfaz.

**Días No Lectivos**  
Calendario > Listado de Días No Lectivos

Buscar por descripción |

Fecha	Descripción	Acción
25/12/2025	Navidad	<input type="button" value="Eliminar"/>
08/12/2025	Inmaculada Concepción	<input type="button" value="Eliminar"/>
06/12/2025	Día de la Constitución	<input type="button" value="Eliminar"/>
09/11/2025	Día de la Almudena	<input type="button" value="Eliminar"/>
01/11/2025	Día de Todos los Santos	<input type="button" value="Eliminar"/>
12/10/2025	Día de la Hispanidad	<input type="button" value="Eliminar"/>
15/08/2025	Asunción de la Virgen	<input type="button" value="Eliminar"/>
25/07/2025	Santiago Apóstol	<input type="button" value="Eliminar"/>
24/06/2025	San Juan	<input type="button" value="Eliminar"/>
15/05/2025	San Isidro	<input type="button" value="Eliminar"/>
02/05/2025	Día de la Comunidad de Madrid	<input type="button" value="Eliminar"/>
01/05/2025	Día del Trabajo	<input type="button" value="Eliminar"/>

Figura 4.10: Listado de días no lectivos registrado en el sistema

## Listado Incidencias

La ventana de Listado de Incidencias, como se puede ver en la Figura 4.11 ofrece un registro detallado de todas las incidencias relacionadas con la impartición de clases. Para cada caso, se muestran datos clave como la fecha en que ocurrió la incidencia, las

asignaturas afectadas, el profesor responsable, el estado de la incidencia (ya sea justificada o no justificada) y, en su caso, la justificación correspondiente.

Esta interfaz facilita el seguimiento visual de las incidencias mediante un sistema de códigos de colores que permite distinguir rápidamente su estado y tipo. Además, proporciona al administrador herramientas para gestionar las justificaciones, modificar el estado de las incidencias según corresponda y generar comunicaciones dirigidas a los departamentos afectados. De este modo, se agiliza la resolución de incidencias y se mejora la coordinación interna entre los distintos responsables académicos.

**Listado de Incidencias**  
Informe > Incidencias

Fecha de inicio: dd/mm/aaaa Fecha de fin: dd/mm/aaaa Estado: Todas [Filtrar](#) [Restablecer Filtro](#)

Estado	Fecha	Asignatura	Profesor	Correo del Profesor	Acciones
X No justificada	15/05/2025	Matemática Discreta	Dorzhi García España	dorzhi.garcia@ucm.es	<a href="#">Justificar</a> <a href="#">Ver Información</a> <a href="#">Eliminar</a>
X No justificada	15/05/2025	Ingeniería del Software	María González Rodríguez	maria.gonzalez@ucm.es	<a href="#">Justificar</a> <a href="#">Ver Información</a> <a href="#">Eliminar</a>
X No justificada	15/05/2025	Programación Avanzada	María González Rodríguez	maria.gonzalez@ucm.es	<a href="#">Justificar</a> <a href="#">Ver Información</a> <a href="#">Eliminar</a>
X No justificada	15/05/2025	Algoritmos	Daniel Lopez Escobar	daniel.lopez@ucm.es	<a href="#">Justificar</a> <a href="#">Ver Información</a> <a href="#">Eliminar</a>
X No justificada	13/05/2025	Inteligencia Artificial	Lucía López García	lucia.lopez@ucm.es	<a href="#">Justificar</a> <a href="#">Ver Información</a> <a href="#">Eliminar</a>
X No justificada	13/05/2025	Sistemas Operativos	Pablo Sánchez Martínez	pablo.sanchez@ucm.es	<a href="#">Justificar</a> <a href="#">Ver Información</a> <a href="#">Eliminar</a>
X No justificada	13/05/2025	Compiladores	Jorge Pérez Hernández	jorge.perez@ucm.es	<a href="#">Justificar</a> <a href="#">Ver Información</a> <a href="#">Eliminar</a>
X No justificada	13/05/2025	Matemática Discreta	Dorzhi García España	dorzhi.garcia@ucm.es	<a href="#">Justificar</a> <a href="#">Ver Información</a> <a href="#">Eliminar</a>
✓ Justificada	04/04/2025	Desarrollo Web	Diego López Pérez	diego.lopez@ucm.es	<a href="#">Ver Justificación</a> <a href="#">Eliminar</a>
X No justificada	04/04/2025	Big Data	Marta González Hernández	marta.gonzalez@ucm.es	<a href="#">Justificar</a> <a href="#">Ver Información</a> <a href="#">Eliminar</a>
✓ Justificada	04/04/2025	Minería de Datos	Adrián Rodríguez López	adrian.rodriguez@ucm.es	<a href="#">Ver Justificación</a> <a href="#">Eliminar</a>
X No justificada	04/04/2025	Aprendizaje Automático	Marina Pérez García	marina.perez@ucm.es	<a href="#">Justificar</a> <a href="#">Ver Información</a> <a href="#">Eliminar</a>

« Primera < Anterior 1 2 3 4 Siguiente > Última »  
Mostrando 12 de 60 registros (Página 2 de 5)

Figura 4.11: Listado de incidencias registrado en el sistema

## Visualización Estadísticas

La ventana de Visualización de Estadísticas, siendo la Figura 4.12 un ejemplo de la ventana, la cual está diseñada para que los administradores puedan analizar de manera eficiente los datos de asistencia y las incidencias docentes en un período específico. Utilizando filtros de fecha, el usuario puede seleccionar el rango temporal de interés y, mediante un botón dedicado, generar las estadísticas correspondientes. La información se presenta en una tabla que resume los datos más relevantes, como el total de asistencias registradas, los días no lectivos, el número de incidencias justificadas y sin justificar, así como el total de incidencias detectadas durante el intervalo seleccionado.

Esta herramienta no solo agiliza la generación de informes administrativos, sino que también facilita la detección de patrones o posibles problemas en la asistencia del profesorado. Por ejemplo, permite identificar rápidamente si existe un número elevado de ausencias sin justificar, lo que puede señalar áreas que requieren atención, o comprobar la eficacia del proceso de justificación de incidencias. En definitiva, la visualización de estadísticas contribuye a una gestión más transparente y proactiva del control de asistencia docente.

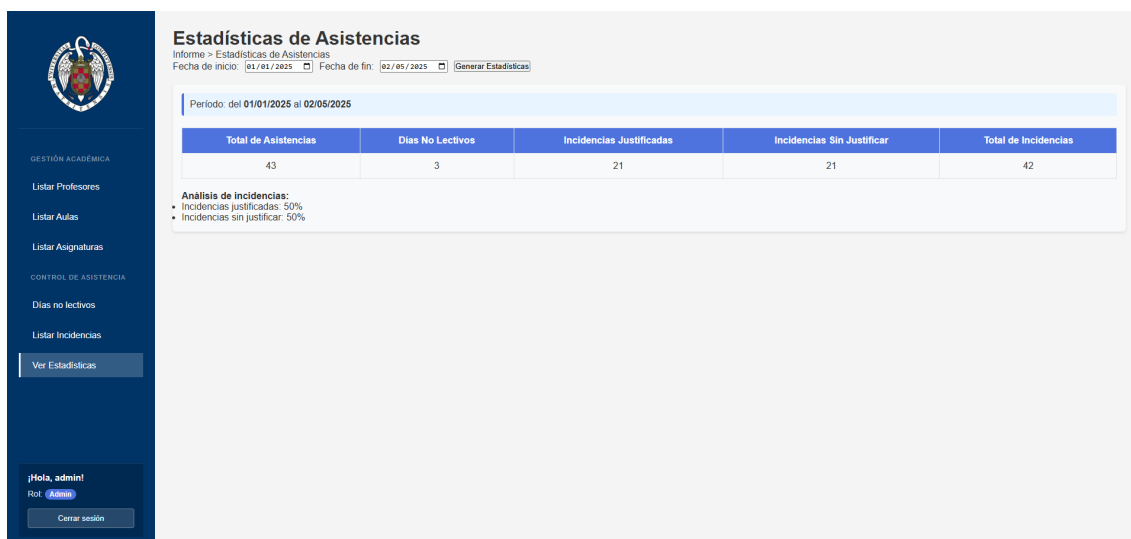


Figura 4.12: Estadísticas generadas por el sistema

## 4.2. Desarrollo de los nodos

Entre las múltiples placas basadas en el ESP32-C6, se ha elegido la FireBeetle 2 ESP32-C6 para este proyecto, principalmente por su módulo de carga de batería integrado y su consumo ultra bajo. Estas características la convierten en una plataforma ideal para proyectos autónomos alimentados por batería, como es el caso de este proyecto.

Los nodos de registro representan el punto de interacción directa entre los profesores y el sistema, permitiendo registrar su asistencia y la impartición de clases de manera sencilla y rápida mediante el uso de tarjetas NFC. Estos dispositivos, diseñados para ser instalados en cada aula, utilizan como microcontrolador el ESP32 Firebeetle 2 C6 de DFRobot, como se puede ver en la Figura 4.13, el módulo LoRa usado es el DX-LR01 de DX-Smart, como se aprecia en la Figura 4.15, el módulo RFID es el RC522 V133, como se puede ver en la Figura 4.16 y una batería de Ion Litio de 3.7V y 1100mA, como la que vemos en la Figura 4.21.

### 4.2.1. Componentes hardware

Todos los componentes hardware del proyecto venían sin los pines soldados, por eso fue necesario preparar los componentes soldando los pines. Tras la soldadura, realizaron comprobaciones con un voltímetro de que no hubiese cortocircuitos entre los pines, además de comprobar cada una de las conexiones entre pines y los pines del microprocesador de los módulos y la placa del ESP32. Una vez comprobados todos los componentes, se conectaron los módulos a la Firebeetle 2, esta placa viene con algunos pines preparados para bajo consumo en algunos de los protocolos como el UART o el I2C.

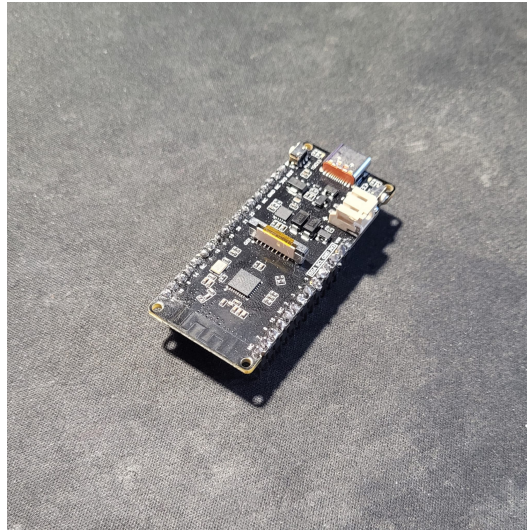


Figura 4.13: Firebeetle 2 ESP32-C6

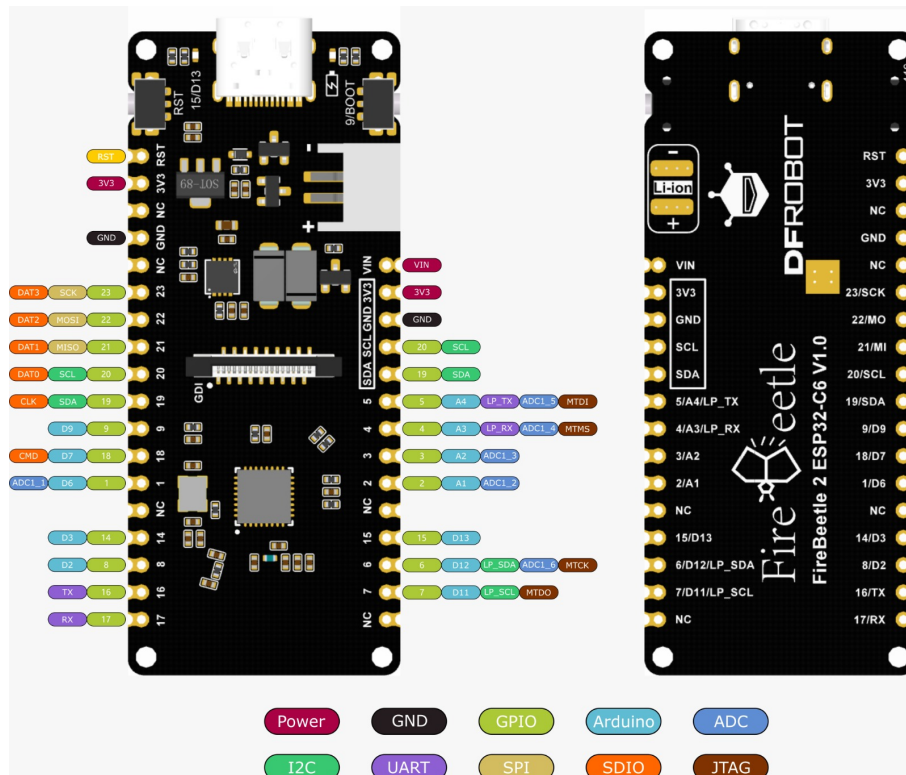


Figura 4.14: Datasheet ESP32 Firebeetle 2 C6

De los diversos módulos LoRa que utilizan distintos protocolos, decidimos usar un módulo que se conecta a través de UART, para así también aprovechar los pines de bajo consumo que posee esta placa. Estos pines de bajo consumo fueron diseñados para reducir el consumo en la transmisión de datos en tecnologías que lo permitiesen como Bluetooth Low Energy o LoRa hasta en un 90% en algunos casos. Además de estas conexiones de comunicación, los módulos LoRa, el rango de voltaje operativo está entre los 3.3V-5.5V, lo que permite que puedan ser alimentados directamente desde los pines de la placa.

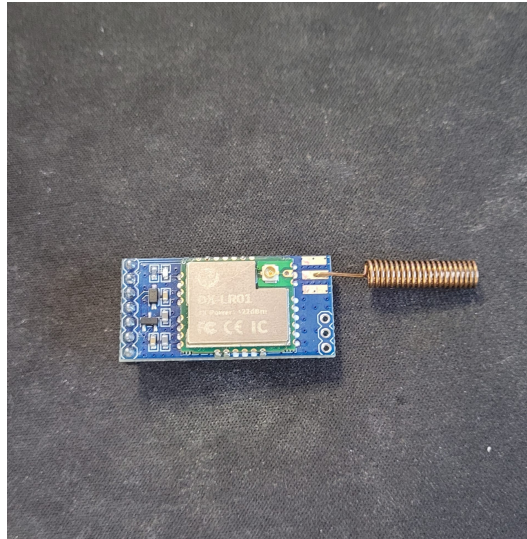


Figura 4.15: Modulo LoRa DX-LR01

En cuanto a los módulos RFID el más común que hay es el RC522, así que por eso optamos por utilizar este en lugar de otros módulos. En concreto, este módulo utiliza SPI como protocolo de interconexión con la placa. Opcionalmente este módulo también tiene un pin IRQ para interrupciones pero no es obligatorio conectarlo y puesto que la interrupción de los nodos se realiza mediante un botón, no lo hemos conectado. Como cualquier otro módulo también requiere de conexiones de energía, y puesto que este módulo opera a 3.3V también podemos alimentarlo mediante una conexión directa con la placa.

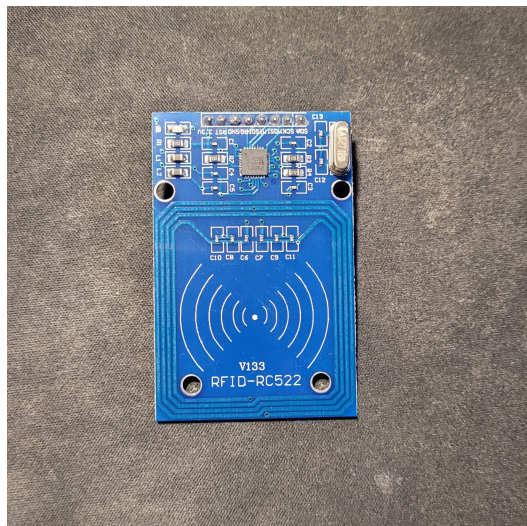


Figura 4.16: Modulo RFID RC522

El diagrama de conexiones de los módulos del nodo es como el que se muestra en la Figura 4.17, una vez todo conectado queda como se aprecia en la Figura 4.18

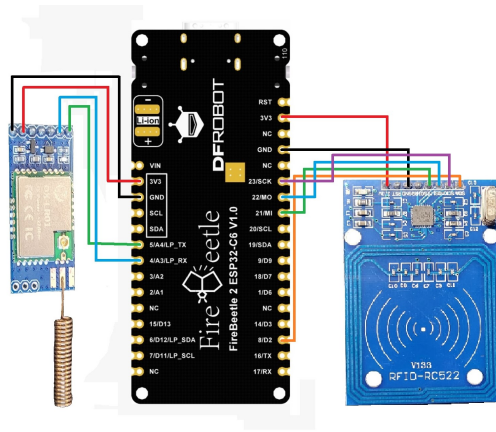


Figura 4.17: Esquema de conexiones

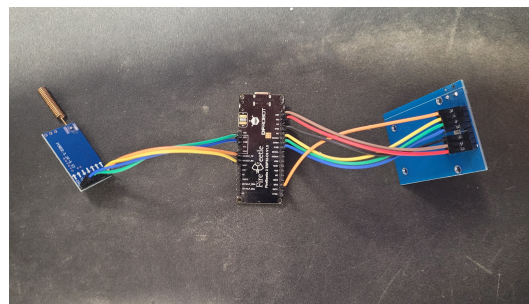


Figura 4.18: Conexiones de los módulos al ESP32

Los dos módulos anteriores los conectamos a un transistor 2N7000 como el de la Figura 4.19, capaz de manejar una corriente máxima de 200mA, suficiente para poder dar corriente a los dos módulos y un voltaje máximo de 60V[20]. Este transistor lo controlaremos con el pin 2 de la ESP32 para evitar que los módulos se mantengan encendidos durante el tiempo de deep-sleep y suministrándoles corriente durante las transacciones, así reduciremos por completo el consumo de los módulos.

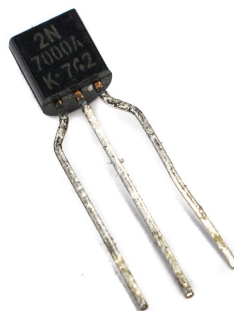


Figura 4.19: Transistor 2N7000

Otro componente fundamental muy relacionado al módulo RFID son las tarjetas que funcionarán como relación del profesor asignado a la misma con el sistema. Estas tarjetas utilizadas son las MIFARE Classic 1k con frecuencia de 13,56 MHz, que son unas tarjetas

muy estandarizadas de 1 kbyte de almacenamiento. Todas estas tarjetas disponen de un UID, Unique Identifier, de 4 bytes que no es modificable.



Figura 4.20: Tarjeta RFID MIFARE Classic 1k

Las baterías utilizadas tienen 1100 miliamperios con un voltaje de 3.7 voltios que se conectan a un puerto específico para baterías de Ion Litio en la placa. Por la arquitectura que dispone la placa, estas baterías pueden ser cargadas directamente conectando la placa mediante USB tipo C a alimentación, lo que facilita mucho la carga de los nodos.



Figura 4.21: Batería 1100 mA

Como se muestra en la Figura 4.22, es el diseño utilizado para almacenar todos los componentes, es una caja simple con dos compartimentos, en el que en uno de ellos almacenamos el ESP32 y el módulo LoRa y el otro compartimento está dirigido a la batería y el módulo RFID que está pegado a la tapa frontal para reducir la distancia de lectura a las tarjetas. Esta caja está impresa con una impresora 3D con filamento PLA, ácido poliláctico, un termoplástico muy común para este tipo de impresiones.

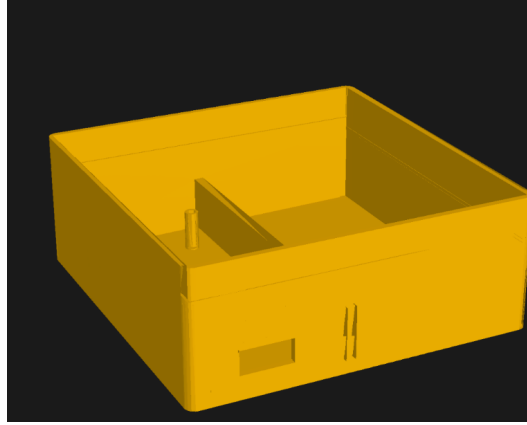


Figura 4.22: Modelo de la caja

### 4.2.2. Desarrollo software

A la hora de realizar el desarrollo para los módulos de espressif, existen como principales alternativas el IDE de Arduino, y el propio de espressif ESP-IDF. El primero ofrece soporte para muchos microcontroladores y un sistema más amigable para principiantes y el segundo ofrece compatibilidad con la gama de espressif y un sistema más complejo de gestión de proyectos a gran escala.

Al comienzo del desarrollo empezamos utilizando el Arduino IDE, ya que es un IDE muy amigable para nuevos usuarios con una gran cantidad de librerías de fácil acceso desde el mismo, con él estuvimos probando el funcionamiento de los módulos. Para el módulo RFID tenía una integración perfecta con la librería MFRC522 y muy fácil de usar. Sin embargo, cuando empezamos a probar con el módulo LoRa se encontraron varios problemas. La mayoría de las librerías para módulos LoRa utilizaban SPI o I2C como protocolo de comunicación, y nuestros módulos se comunicaban por UART. Esto hizo que comenzáramos a probar un IDE más avanzado que permitiese profundizar más con la placa. Al final, y a pesar de no encontrar librerías para módulos LoRa, la librería propia para la UART del ESP-IDF era más que suficiente para poder utilizarlos.

Finalmente, para el desarrollo del código de los nodos hemos utilizado el ESP-IDF, ya que incluye muchas de las librerías para los protocolos de comunicación de la placa, además de ser muy flexible en cuanto a la configuración de la placa. También encontramos una adaptación antigua de la librería para el módulo RFID, que utilizábamos en el Arduino IDE, en C para el ESP-IDF.

Para la implementación del proyecto nos hemos apoyado en numerosas librerías, tanto externas como del propio entorno.

### 4.2.3. Librerías externas

- 'MFRC522' hecho por anuragmandal[5] para la comunicación de manera sencilla con el módulo RFID. Esta posee funciones implementadas a través del protocolo SPI para comprobar cuando hay una tarjeta presente y lectura y escritura de varios tipos de tarjetas RFID diferentes.

#### 4.2.4. Librerías de ESP-IDF

- **'uart.h'** para la comunicación con el módulo LoRa, ya que este retransmite lo que le llega por ella.
- **'spi\_master.h'** y **'spi\_common.h'** crea el canal de spi utilizado después por la librería **'MFRC522.h'**.
- **'gpio.h'** para la configuración de los pines como el utilizado para controlar el transistor.
- **'esp\_log.h'** genera LOGS de depuración en los nodos.
- **'task.h'** para los delays dentro del código.
- **'esp\_sleep.h'** se encarga de inducir el modo deep-sleep y las interrupciones que lo despiertan.
- **'esp\_timer.h'** con ella podemos crear interrupciones que se ejecuten al x tiempo.

Una vez seleccionado el IDE y las librerías que se utilizarán en el proyecto, se diseñó el funcionamiento basado en el diagrama de estados de la Figura 4.23, de 4 estados:

Deep-Sleep, estado de suspensión, minimizando el consumo, esperando la interrupción del botón para comenzar con la operación. Al recibir la interrupción del botón, pasamos al estado Wake-Up.

Wake-Up, estado de inicialización y configuración del módulo RFID para realizar la operación, esperaremos a que se detecte alguna tarjeta o pasen 10 segundos. Si se detecta una tarjeta, se pasará al estado Read-Card. Si han pasado los 10 segundos y no se ha detectado ninguna tarjeta, se volverá al estado deep-sleep.

Read-Card, estado de lectura de los valores de la tarjeta e inicialización y configuración del módulo LoRa para realizar la comunicación con el servidor. Al terminar de inicializar el módulo, se pasará al estado Send-Data.

Send-Data, estado de envío, a través del módulo LoRa, del dato leído de la tarjeta mediante el módulo RFID y se volverá al estado Deep-Sleep hasta que haya que realizar otra operación.

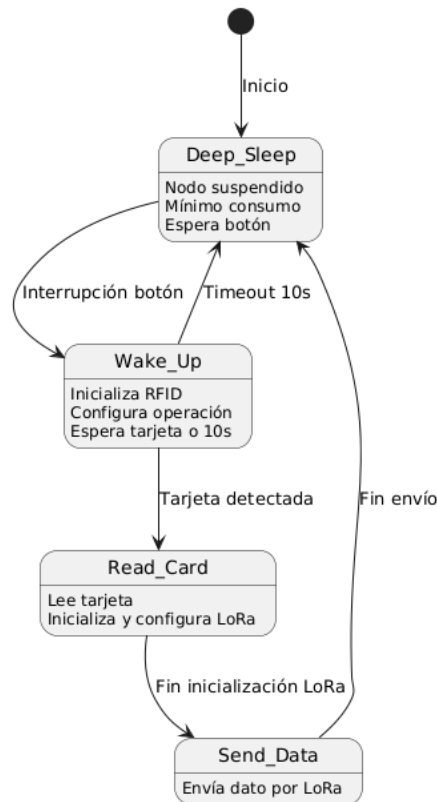


Figura 4.23: Diagrama de Estados de los nodos

## 4.3. Guía de instalación

### 4.3.1. Instalación servidor

Aquí mostramos cómo configurar el servidor para que pueda alojar la página web.

#### Instalar Docker en la Raspberry Pi 5

1. Instala las dependencias necesarias:

```
sudo apt install -y apt-transport-https ca-certificates curl
gnupg lsb-release
```

2. Añade la clave GPG oficial de Docker:

```
curl -fsSL https://download.docker.com/linux/debian/gpg |
sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-
keyring.gpg
```

3. Configura el repositorio:
-

```
echo "deb [arch=arm64 signed-by=/usr/share/keyrings/docker-  
archive-keyring.gpg] https://download.docker.com/linux/  
debian $(lsb_release -cs) stable" | sudo tee /etc/apt/  
sources.list.d/docker.list > /dev/null
```

4. Actualiza e instala Docker:

```
sudo apt update  
sudo apt install -y docker-ce docker-ce-cli containerd.io
```

5. Añade tu usuario al grupo Docker para evitar usar sudo(es necesario un reinicio):

```
sudo usermod -aG docker $USER
```

6. Instala Docker Compose:

```
sudo apt install -y docker-compose
```

## Clonar el repositorio del proyecto desde GitHub

1. Instala Git si aún no está instalado:

```
sudo apt install -y git
```

2. Clona tu repositorio de GitHub:

```
git clone https://github.com/AngelGallegoUCM/GAP.git ~/[  
nombre-de-la-carpeta]  
cd ~/[nombre-de-la-carpeta]
```

## Iniciar los contenedores

1. Asegúrate de estar en el directorio del proyecto:

```
cd ~/[nombre-de-la-carpeta]
```

2. Inicia los contenedores con Docker Compose:

```
docker-compose up -d --build
```

3. Verifica que los contenedores estén funcionando:

```
docker-compose ps
```

## En casos de error al iniciar el contenedor

- Asegúrate de que los puertos 80, 8080 y 3306 estén libre, para comprobarlo:

```
sudo lsof -i :80
```

- En caso afirmativo, usa :

```
sudo systemctl stop [proceso]
sudo systemctl disable [proceso]
```

## Acceder a tu aplicación web

1. Accede a tu página web desde cualquier navegador en tu red local:

- Web: `http://[IP-DE-LA-RASPBERRY]:80`
- phpMyAdmin: `http://[IP-DE-LA-RASPBERRY]:8080`

2. Inicia sesión en phpMyAdmin:

- Usuario: `root`
- Contraseña: `admin123!`

3. Inicia sesión en la página web:

- Usuario: `admin`
- Contraseña: `Admin123!`

### 4.3.2. Instalación de los nodos

Aquí mostramos como configurar los nodos que se ubicarán en las aulas.

#### Instalación ESP-IDF

1. En VsCode instalamos la extensión "ESP-IDF"
2. Configuramos la extensión con la versión v5.4.1
3. Configuramos el chip con el ESP32-C6 chip with LP core (via builtin USB-JTAG)
4. Seleccionamos el puerto correspondiente a nuestra placa
5. Seleccionamos el método de flasheo del chip con UART

## Configuración del nodo de las aulas

Para configurar el nodo a un aula correspondiente, basta con modificar el código de la variable AULA como muestra la Figura 4.24, al número del aula con formato 4 dígitos, por ejemplo "0013".

```
9  #include "esp_log.h"
10 #include "freertos/FreeRTOS.h"
11 #include "freertos/task.h"
12 #include "esp_sleep.h"
13 #include "esp_timer.h"
14
15 #define BUF_SIZE 128
16 #define WAKEUP_GPIO GPIO_NUM_2
17 #define LED_GPIO GPIO_NUM_15
18 #define UART_PORT UART_NUM_1
19 #define TX_PIN GPIO_NUM_4
20 #define RX_PIN GPIO_NUM_5
21 #define SPI_CLK 21
22 #define SPI_MOSI 22
23 #define SPI_MISO 23
24 #define SPI_SS 8
25
26 const char* AULA = "0009";
27
28 static const char* TAG = "Nodo";
29
30 void interrupt_handler(void* arg) {
31     gpio_config_t io_conf = {
32         .pin_bit_mask = (1ULL << LED_GPIO),
33         .mode = GPIO_MODE_OUTPUT,
```

Figura 4.24: Variable AULA a modificar

## Programar el nodo

Por último, solo queda compilar el código y flashear la placa con él. Para ello, pulsamos el botón con un icono de un fuego, 'ESP-IDF: compilar, flashear y monitorear', este se encargará de todo el proceso y cargará el programa en la placa.

# Capítulo 5

## Resultados

### 5.1. Repositorios del proyecto

**Repositorio del docker con el servidor:** [github.com/AngelGallegoUCM/GAP](https://github.com/AngelGallegoUCM/GAP)

**Repositorio de los nodos:** [github.com/AngelGallegoUCM/NodosGAP](https://github.com/AngelGallegoUCM/NodosGAP)

### 5.2. Estructura del sistema

El sistema cuenta con un servidor implementado en una Raspberry Pi y un nodo por cada aula que se comunicarán con el servidor a través de una conexión LoRa.

El servidor cuenta con una base de datos en la que se guarda toda la información de profesores, aulas, asignaturas, horarios e incidencias y posee triggers diarios para generar las incidencias producidas durante el día. También contamos con una página web para poder ver la información de la base de datos de manera más visual y poder gestionarla fácilmente como agregar profesores o asignaturas y modificarlos. Por último, tenemos un programa en Python que se encarga de recibir los datos del LoRa y comunicarse con la base de datos.

Los nodos cuentan con un módulo RFID para la lectura de las tarjetas y un módulo LoRa para el envío de los datos de la lectura de la tarjeta, además del aula a la que está asignado ese nodo. Adicionalmente, para optimizar el consumo energético, los nodos poseen un botón para comenzar una transacción de lectura de tarjeta que se encarga de activar un transistor que es el encargado de suministrar corriente a los distintos módulos conectados.

## 5.3. Funcionamiento de la web

Con la web podemos modificar la información de gestión académica importante de la base de datos para añadir, eliminar o modificar profesores, aulas, asignaturas y horarios. También existe la posibilidad de gestionar la información relacionada con el control de asistencia, esta es la relativa a los días no lectivos del curso y lo más importante, la gestión de incidencias.

La gestión de incidencias es el punto clave de la página web, en la web podemos ver el listado de incidencias ocurridas durante un periodo concreto y filtrar por cuáles queremos ver, eliminarla y ver la justificación que tienen o, en el caso de que no esté justificada, justificarla manualmente.

En la web se puede visualizar las estadísticas de las incidencias para un periodo concreto, en ellas se puede diferenciar el total de asistencias del periodo, los días no lectivos durante el periodo, las incidencias justificadas y sin justificar del mismo, y el total de incidencias. También, encontramos un análisis de las incidencias en el que se aprecia qué porcentaje del total de incidencias han sido justificadas y cuál han sido sin justificar. Todos estos datos son relevantes de cara a realizar el informe de control de docencia que hay que presentar a la universidad al final de cada año lectivo, en el que se exponen el total de incidencias ocurridas en el curso, cuántas y qué porcentaje han sido justificadas y cuántas sin justificar.

## 5.4. Funcionamiento de los nodos

El funcionamiento de los nodos es muy amigable para el usuario. Cada profesor tendrá su propia tarjeta identificativa que le representará y cada aula poseerá un nodo por el que poder identificarse. El profesor que acceda al aula debe dirigirse al nodo, presionar el botón que lo activa y dispondrá de una ventana de 10 segundos para poder pasar la tarjeta por el frontal del nodo, lo que leerá el contenido de la tarjeta y transmitirá la información referente al aula en la que se ubica el nodo y el profesor que ha pasado la tarjeta. Si en ese plazo de 10 segundos no se ha pasado ninguna tarjeta, el nodo volverá al estado Deep-Sleep.

La autonomía media de los nodos suponiendo las transacciones necesarias en el peor de los casos, todos los días clases de 1 hora, que son 990 clases al año, además de un margen de transacciones fallidas, un 10% de las clases tendrán una transacción fallida, 100 transacciones y sumando el tiempo de Deep-Sleep de un año completo, contando con las festividades. Tenemos un consumo de 360,1 mA anuales, lo que nos da una autonomía de poco más de 3 años con las baterías utilizadas.

El alcance de los nodos es más que suficiente para abarcar toda la facultad de informática, según las pruebas realizadas en varios puntos de la misma, e incluso es tal el alcance de este sistema que se podría implementar en las aulas del edificio multiusos para las clases que se imparten allí. Los nodos poseen un alcance de entre 300 y 500 metros, dependiendo de las interferencias y obstáculos que haya en el camino.

## 5.5. Pruebas

Una vez finalizada la implementación del software, comprobamos que todas las hipótesis realizadas en cuanto al funcionamiento del sistema fueran óptimas. Para ello, medimos el consumo de los nodos y las distancias de transmisión al servidor.

### 5.5.1. Servidor

Las pruebas de estrés del servidor en una Raspberry Pi 5 de 16 GB, nos dieron como resultado que la Raspberry Pi 4 que estuvimos investigando es más que suficiente para poder soportar el servidor, la base de datos y el script de Python. Ejecutando todo lo mencionado anteriormente en nuestra Raspberry Pi 5 el consumo de RAM es menor a 2 GB. La placa de menor coste que se puede adquirir para el funcionamiento del servidor es la Raspberry Pi 4 de 2 GB.

Para el almacenamiento del sistema estaríamos usando cerca de 10 GB, teniendo en cuenta el sistema operativo, Apache para levantar la página web en PHP, Mysql para almacenar la información del sistema y Phpmyadmin para poder interactuar con la base de datos con una interfaz gráfica. Para el almacenamiento de los datos de nuestro sistema estaremos usando aproximadamente 100 MB anuales. Sumando el almacenamiento tanto del sistema como los datos con una tarjeta microSD, que es la que utilizan las raspberry, de 16 GB es suficiente para su uso, si bien es cierto que recomendamos una tarjeta de 32 GB para mejorar el rendimiento del sistema, y la diferencia de coste no es muy significativa.

### 5.5.2. Consumo

Las pruebas de consumo se ajustaron bastante a lo que habíamos estado barajando durante la investigación de los componentes. Estas pruebas las estuvimos realizando con un kit de análisis de corriente de la marca Nordic como se ve en la Figura 5.1, el Nordic Power Profile Kit II[13], el cual tiene una precisión de 100nA y 1mA dependiendo el amperaje medido.

En el modo deep-sleep el nodo consume una media de  $23\mu\text{A}$  como se aprecia en la Figura 5.2. Con este consumo y las baterías que estamos utilizando manteniendo el modo deep-sleep, los nodos podrían permanecer 47.826 horas dormidos, lo que serían casi 5 años y medio.

Durante una transacción de 6 segundos como la que se puede ver en la Figura 5.3, que es el tiempo desde que se ha pulsado el botón hasta que se pasa la tarjeta en este caso. Durante esta transacción tenemos un pico máximo de 125 mA cuando se realiza envío de datos a través del módulo LoRa. Durante la transacción se consume una carga de aproximadamente 400 mC que es igual a 0,1 mAh, esto nos da un total de 11.000 transacciones con una sola carga de la batería, anualmente la Universidad Complutense tiene 18 semanas de impartición de clases, y en el peor de los casos que hubiese una clase cada hora se estarían produciendo 990 transacciones al año.

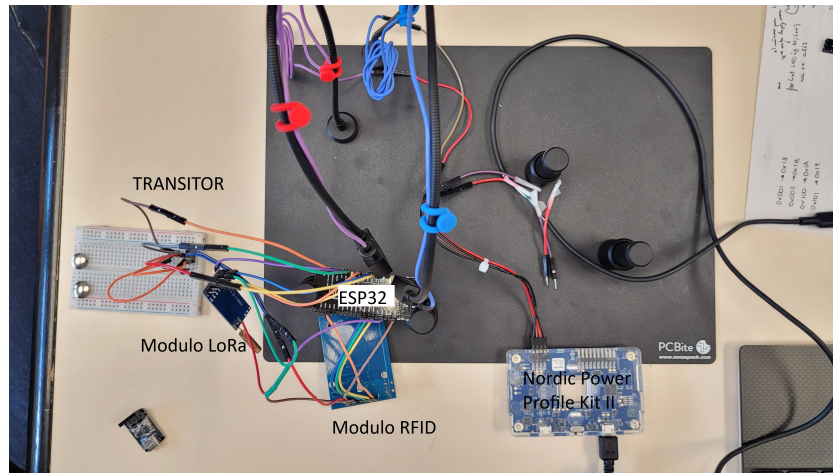


Figura 5.1: Conexiones para la prueba de consumo

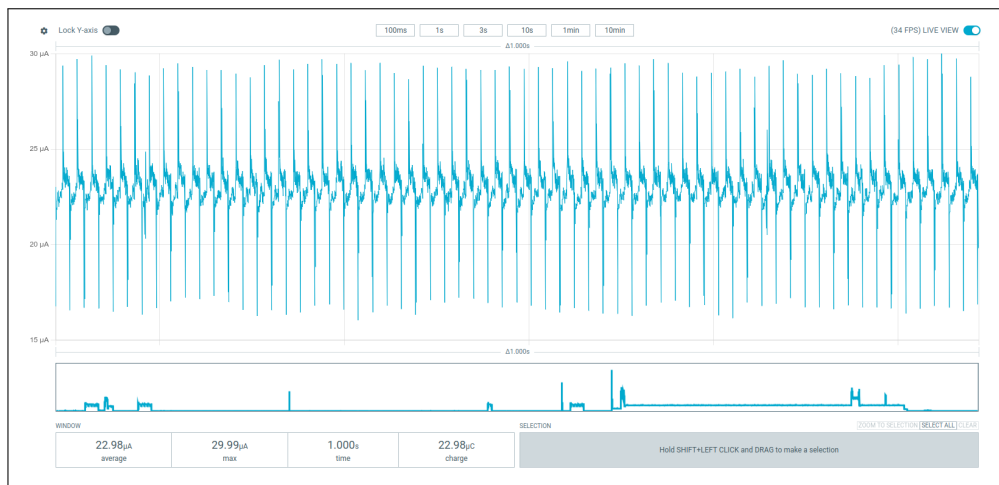


Figura 5.2: Consumo en modo deep-sleep

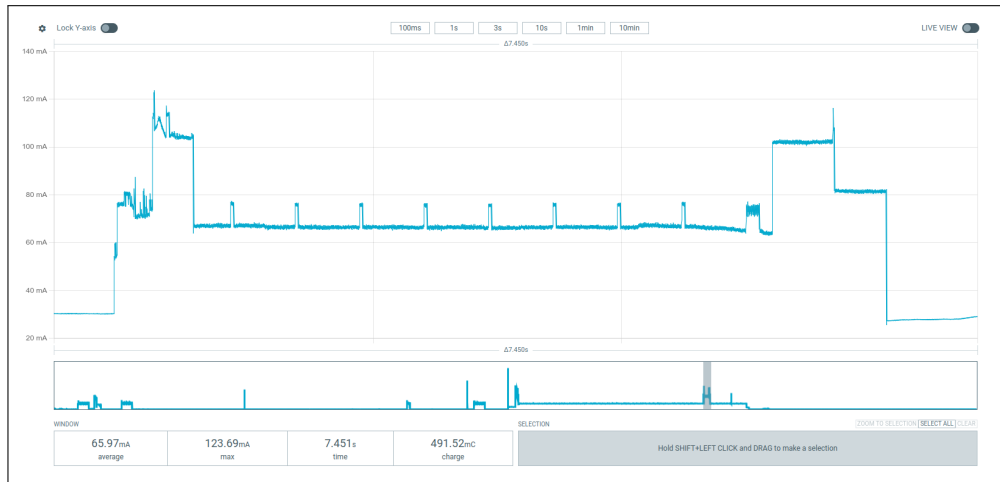


Figura 5.3: Transacción de 6 segundos

Por último, medimos cuánto consume una transacción que no se produzca, sino que se esperen los 10 segundos y vuelva automáticamente al modo deep-sleep como en la Figura 5.4. En este fallo se consume una carga de aproximadamente 600 mC que es igual a 0,166 mAh, esto nos da un total de 6.600 transacciones fallidas con una sola carga de la batería.

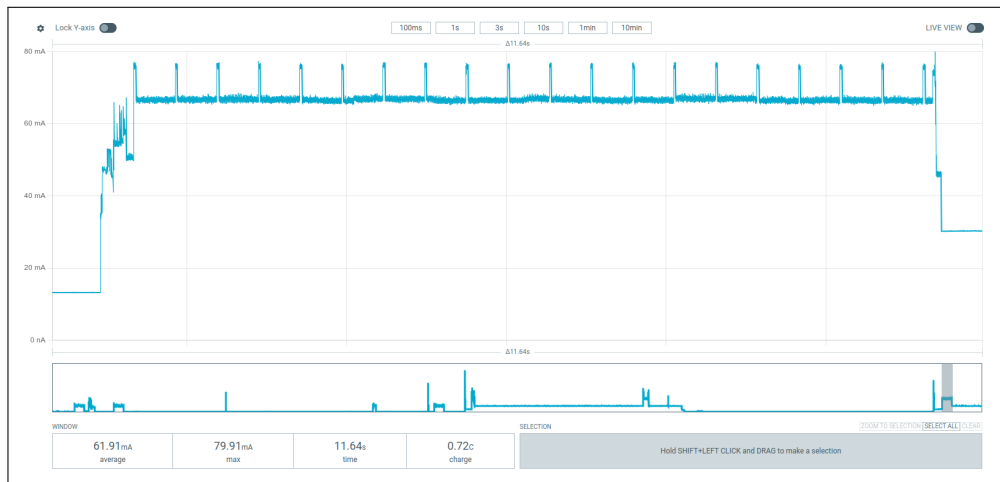


Figura 5.4: Transacción fallida de 10 segundos

Haciendo un análisis medio de los consumos suponiendo las 990 transacciones y además que un 10% de ellas se hagan también como transacciones fallidas, 100 aproximadamente, y sumando el tiempo en deep-sleep del nodo durante el año, el nodo estaría consumiendo 360,1 mA, lo que nos otorga una autonomía de poco más de 3 años útiles.

### 5.5.3. Distancia

Para las pruebas de distancia, ubicamos el servidor en la zona de despachos de la segunda planta, más específicamente en el despacho 229, y realizamos diversos envíos de

información con el módulo desde distintos puntos de la facultad en los que la señal llegaba sin ninguna interferencia.

- **Laboratorio 7**, ubicado en la misma planta en la otra punta de la planta, hay 65 metros en línea recta con las paredes de por medio.
- **Aula 13**, ubicada en la planta 1 en la otra punta de la planta, hay 66 metros en línea recta con paredes y el suelo de por medio.
- **Terraza**, ubicada en la planta baja en la otra punta de la planta, hay 66,5 metros en línea recta con paredes y el suelo de por medio.
- **Secretaria**, ubicada en la planta baja en la misma punta de la planta, hay 10 metros en línea recta con el suelo de por medio.
- **Letras FDI**, ubicada en la planta baja al suroeste de la facultad, hay 79 metros en línea recta con una pared de por medio.
- **Final del camino de tierra a la multiusos**, ubicada suroeste de la facultad, hay 285 metros en línea recta con una pared de por medio.
- **Escaleras María Zambrano (Norte)**, ubicada suroeste de la facultad, hay 356 metros en línea recta con una pared de por medio.
- **Escaleras María Zambrano (Sur)**, ubicada suroeste de la facultad, hay 376 metros en línea recta con la facultad de historia delante.
- **Hall Multiusos**, ubicada en la planta baja de la multiusos, hay 294 metros con la multiusos de por medio.
- **Laboratorios Multiusos**, ubicada en la planta baja de la multiusos, hay 303 metros con la multiusos de por medio.
- **Cafetería Multiusos (Exterior)**, ubicada en la planta baja de la multiusos, hay 309 metros con la multiusos parcialmente de por medio.

Uno de los límites que se encontraron fue el interior de la cafetería de la multiusos, ubicada en la planta baja de la multiusos, hay 308 metros con la multiusos de por medio. En este punto la señal comienza a debilitarse por las interferencias de la gente y la multiusos y en algunas ocasiones no llegaba el mensaje completo.

# Capítulo 6

## Conclusiones y trabajo futuro

### 6.1. Conclusiones

Hasta la fecha, el control de asistencia de la impartición de las clases se está realizando de forma rudimentaria, haciendo uso de hojas de papel donde firmar. Esto conlleva unos riesgos de pérdida de información, una indisponibilidad de los datos en tiempo real y una alta carga administrativa. Para atacar esos problemas, proponemos una implantación de nodos autónomos en las diferentes aulas formadas con microcontroladores ESP32-C6, lectores RFID y comunicación LoRa. Esta red de nodos registran la asistencia de los profesores y envían estos datos a un servidor central (Raspberry Pi 5), donde almacenamos en una base de datos, gestionamos y visualizamos en la página web de administración.

Las utilidades de este sistema son: la capacidad de registrar la asistencia del profesorado y poder verlo en tiempo real, generar informes estadísticos de la franja de tiempo que se desee y la generación automática de las incidencias sin depender de un departamento.

Dentro de los logros que se han alcanzado con el Sistema de Gestión de Asistencia del Profesorado son, en primer lugar, la digitalización completa del proceso de registro de asistencia, eliminando por completo el uso de papel y cumpliendo así uno de los objetivos principales del proyecto. Otro es que la automatización de tareas administrativas ha permitido que procesos que antes requerían una considerable dedicación de recursos humanos, como la detección de incidencias y la generación de informes, se realicen ahora de manera automática. Esto ha liberado al personal administrativo, que puede centrarse en tareas de mayor importancia. Y, finalmente, otro de los logros alcanzados es la calidad y disponibilidad de la información, ya que los datos de asistencia de clases están ahora disponibles en tiempo real y con un alto grado de fiabilidad.

### 6.2. Líneas de trabajo futuro

Basándonos en los resultados obtenidos y las limitaciones identificadas, proponemos diversas líneas de evolución para nuestro sistema. En primer lugar, el desarrollo de una aplicación móvil es una prioridad clara para optimizar la experiencia del usuario. Esta

aplicación complementaria permitiría a los profesores acceder a toda la funcionalidad del sistema desde sus dispositivos personales, eliminando la necesidad de acudir a un ordenador para consultar horarios o justificar incidencias. Utilizando tecnologías como React Native o Flutter, se podría crear una solución multiplataforma que funcione tanto en iOS como en Android, cubriendo de este modo todos los posibles dispositivos del profesorado. La aplicación incluiría notificaciones que alertarían inmediatamente sobre cambios de horario, incidencias detectadas o recordatorios de clases próximas. Una característica particularmente innovadora sería la capacidad de utilizar el propio smartphone como alternativa a las tarjetas NFC para el registro de asistencia, aprovechando la tecnología NFC presente en la mayoría de los dispositivos actuales o, alternativamente, implementando un sistema basado en códigos QR ubicados en las aulas.

Para la gestión de las incidencias, ahora mismo sigue siendo bastante manual aunque se ha eliminado toda la carga de detectar las incidencias, pero se podría implementar algún mecanismo con inteligencia artificial que fuera capaz de detectar el tipo de incidencia que se ha producido para clasificarla automáticamente para reducir al mínimo la interacción con el sistema, limitándolo a una revisión mensual para comprobar que no hay ninguna incidencia que el sistema no haya sido capaz de tratar.

Paralelamente, resulta esencial mejorar la integración con otros sistemas universitarios para crear un ecosistema digital. Esto implica el desarrollo de una conexión específica para cada una de las plataformas de gestión académica utilizadas por la universidad, como pueden ser sistemas de matriculación, planificación docente o campus virtuales. Esta integración permitiría la sincronización automática de horarios oficiales, asignación de aulas y cambios en la carga docente, eliminando la necesidad de introducir manualmente esta información en nuestro sistema.

Otra de las cosas que se pueden hacer es mejorar el envío con el módulo LoRa, para que este sea capaz de verificar que el mensaje ha sido recibido por el servidor. Se podría implementar un temporizador que, si no ha recibido una confirmación desde el servidor, continúe enviando el mismo código hasta obtener dicha confirmación.

# Capítulo 7

## Conclusion and future work

### 7.1. Conclusion

Until now, class attendance has been tracked in a rudimentary manner using paper-based signature sheets. This approach carries risks of information loss, lacks real-time data availability, and imposes a significant administrative burden. To address these issues, we propose the deployment of autonomous nodes in classrooms, built with ESP32-C6 microcontrollers, RFID readers, and LoRa communication. This network of nodes records teacher attendance and sends the data to a central server (Raspberry Pi 5), where it is stored in a database, managed, and visualized through an administration website.

The system offers several key functionalities: the ability to record teacher attendance and view it in real time, generate statistical reports for any desired time range, and automatically report incidents without the need for departmental intervention.

Among the achievements of the Faculty Attendance Management System are, firstly, the complete digitization of the attendance registration process, entirely eliminating the use of paper and fulfilling one of the main objectives of the project. Another significant achievement is the automation of administrative tasks, allowing processes that previously required substantial human resources—such as incident detection and report generation—to be handled automatically. This has freed administrative staff to focus on higher-value tasks. Lastly, the system has improved the quality and availability of information, with class attendance data now accessible in real time and with a high degree of reliability.

### 7.2. future work

Based on the results obtained and the limitations identified, we propose several future development directions for our system. First, the development of a mobile application is a clear priority to optimize the user experience. This companion app would allow teachers to access the full functionality of the system from their personal devices, eliminating the need to use a computer to check schedules or justify incidents. Using technologies such as

React Native or Flutter, a cross-platform solution could be created to work on both iOS and Android, thus covering the full range of devices used by faculty members. The app would include notifications to immediately alert users about schedule changes, detected incidents, or upcoming class reminders. A particularly innovative feature would be the ability to use the smartphone itself as an alternative to NFC cards for attendance registration, leveraging the NFC technology present in most modern devices or, alternatively, implementing a QR code-based system located in the classrooms.

Currently, incident management remains relatively manual—although the burden of detecting incidents has been eliminated, it would be possible to implement an AI-powered mechanism capable of identifying the type of incident that has occurred and automatically classifying it. This would further minimize user interaction with the system, reducing it to a monthly review to ensure that no unhandled incidents remain.

In parallel, enhancing integration with other university systems is essential to build a cohesive digital ecosystem. This would involve developing specific connectors for each academic management platform used by the university, such as enrollment systems, teaching planning tools, or virtual campuses. Such integration would allow for the automatic synchronization of official timetables, classroom assignments, and teaching load changes, eliminating the need to manually enter this data into our system.

Another area for improvement is the LoRa communication module. Specifically, enhancing message delivery by implementing an acknowledgment mechanism from the server. A timer could be used so that, if confirmation is not received, the node continues to send the same code until the server successfully acknowledges receipt.

# Capítulo 8

## Contribuciones personales

### 8.1. Aportaciones de Raúl Durán Catalán

Mi trabajo se centró en el desarrollo de los nodos y la comunicación con el servidor, abarcando desde la investigación de las tecnologías de comunicación, el diseño del hardware, la implementación del sistema empotrado y la declaración de los requisitos del sistema.

#### 8.1.1. Declaración de requisitos

Al comienzo del proyecto, realicé una propuesta inicial de requisitos que debíamos abarcar en el apartado hardware: conectividad, diseño, autonomía y costo.

También realicé la propuesta inicial de los requisitos del sistema que gestionábamos a través de la web que incluí, la capacidad de visualizar de manera sencilla las faltas de la docencia en los horarios. Estos requisitos se vieron ampliados tras la reunión con Mar, teniendo en cuenta los distintos tipos de incidencias, como se realizaba la comunicación y gestión de dichas incidencias.

#### 8.1.2. Investigación

En cuanto al apartado de investigación de este proyecto, mi trabajo incluye:

- Investigación de distintas tecnologías de comunicación que fuesen aplicables para un proyecto en el Internet de las Cosas.
- Comparación de las tecnologías teniendo en cuenta los requisitos del sistema para su utilización.
- Búsqueda de los módulos de las tecnologías seleccionadas para su implementación.

- Investigación de los protocolos de comunicación que utilizaban los distintos módulos para comunicarse con la placa.
- Investigación de distintos microprocesadores para la implementación de los módulos, teniendo especial énfasis en un regulador de voltaje de bajo consumo como el HM6245.
- Investigación de distintas placas o mini ordenadores capaces comunicarse mediante la tecnología seleccionada que tuviesen capacidad para gestionar la base de datos y la aplicación web.

### 8.1.3. Sistema empotrado

En cuanto al diseño y la implementación del sistema empotrado, los nodos han sido implementados en C con el IDE ESP-IDF y la aplicación de conexión con el servidor es un programa en Python, mi trabajo incluyó:

- Soldadura de los pines de las placas de los nodos y los módulos de comunicación.
- Conexión de los módulos con las placas de los nodos y la placa del servidor.
- Diseño e Implementación del programa para el funcionamiento de los nodos.
- Diseño e implementación del programa para la conexión del servidor con los nodos.
- Pruebas de consumo de los nodos.
- Pruebas de conectividad entre nodos y servidor con grandes distancias y obstáculos.

### 8.1.4. Memoria

En la memoria, me he encargado de escribir la parte correspondiente a mis investigaciones sobre las tecnologías de comunicación y las placas en el capítulo de contexto tecnológico.

En el capítulo de implementación me he encargado de escribir la implementación de los nodos.

Mayoritariamente he escrito el capítulo de resultados, ya que me he encargado de realizar las pruebas.

En el capítulo de introducción y conclusión he escrito aproximadamente la mitad, ya que los hemos ido escribiendo juntos.

## 8.2. Aportaciones de Ángel Gallego Muñoz

Mi trabajo se centró en el desarrollo integral del sistema software, abarcando desde el diseño de la base de datos hasta la implementación de la interfaz web y su integración con los componentes hardware.

### 8.2.1. Diseño e implementación de la base de datos

Al inicio del proyecto, realicé un análisis para definir el modelo entidad-relación que pudiera cumplir las necesidades de gestión de asistencia del profesorado. Esto implicó:

- Identificación de todas las entidades principales del sistema (profesores, departamentos, aulas, asignaturas, horarios, asistencias, incidencias, días no lectivos y usuarios).
- Definición de las relaciones entre estas entidades, estableciendo cardinalidades y atributos apropiados.
- Diseño de un esquema relacional normalizado para garantizar la integridad de los datos y evitar redundancias innecesarias.
- Implementación de la base de datos en MySQL, creando todas las tablas necesarias con sus respectivas claves primarias y foráneas.
- Desarrollo de la configuración inicial del servidor mediante Docker

Esta estructura de datos permite almacenar toda la información necesaria para el sistema.

### 8.2.2. Desarrollo del backend del servidor

Para el desarrollo del backend del servidor, utilicé PHP para implementar la lógica de la página web. Esto incluyó:

- Desarrollo de un sistema de autenticación y autorización basado en roles, que controla el acceso a las diferentes funcionalidades según el perfil del usuario.
- Implementación de módulos de verificación automática que analizan las asistencias y generan incidencias.
- Generación automatizada de registros de asistencia para las clases programadas en días lectivos.

### 8.2.3. Desarrollo de la interfaz web

En cuanto a la interfaz de usuario, diseñé una página basada en HTML, CSS y JavaScript, priorizando la usabilidad y la accesibilidad. Mi trabajo incluyó:

- Diseño e implementación del sistema de inicio de sesión, con validación de credenciales y gestión de sesiones.
- Desarrollo del menú principal que proporciona acceso a todas las funcionalidades del sistema.
- Creación de interfaces para la gestión de usuarios, profesores, aulas, asignaturas y horarios.
- Implementación de formularios interactivos con validación en tiempo real para la entrada de datos.
- Desarrollo de tablas dinámicas para la visualización de información, con funcionalidades de ordenación y filtrado.
- Implementación de vistas específicas para la gestión de incidencias, permitiendo su clasificación y justificación.
- Desarrollo de un módulo de estadísticas que proporciona informes visuales sobre asistencias e incidencias.
- Implementación de la interfaz para la configuración del sistema y parámetros generales.

Para mejorar la experiencia de usuario, las interfaces desarrolladas incluyen un esquema de colores coherente en todo el sistema, que facilite la lectura y la navegación por el mismo, iconos que ayudan a identificar rápidamente las diferentes funcionalidades y mensajes de confirmación y error que guían al usuario durante la interacción con el sistema.

### 8.2.4. Memoria

Me ocupé de escribir la parte de la memoria relacionada con las tecnologías que existen a la hora de crear una página web.

Con la parte de implementación escribí sobre el funcionamiento de la base de datos así como de la página web. También expliqué cómo hay que realizar la configuración para poder usar el servidor mediante Docker.

# Bibliografía

- [1] 98thPercentile contributors. *JavaScript in Modern Web Development*. n.d. URL: <https://www.98thpercentile.com/blog/javascript-in-modern-web-development/>.
- [2] Hussein Ahmad Al-Ofeishat y Mohammad A.A. Al Rababah. «Near Field Communication (NFC)». En: *IJCSNS International Journal of Computer Science and Network Security* (2012). URL: [http://paper.ijcsns.org/07\\_book/201202/20120216.pdf](http://paper.ijcsns.org/07_book/201202/20120216.pdf).
- [3] Thomas Clausen Aloÿs Augustin Jiazi Yi 1 y William Mark Townsley. «A Study of LoRa: Long Range and Low Power Networks for the Internet of Things». En: *Sensors* 16.9 (2016), pág. 1466. URL: <https://www.mdpi.com/1424-8220/16/9/1466>.
- [4] Ankur Tomar. *Introduction to Zigbee technology*. 2011. URL: <https://eclass.uoa.gr/modules/document/file.php/DI367/E/introduction-to-zigbee-technology.pdf>.
- [5] anuragmandal. *Libreria MFRC522*. 2019. URL: <https://github.com/anuragmandal/ESPIDF-INTERFACE-MFRC522?tab=readme-ov-file>.
- [6] Blikai contributors. *What is Bluetooth? Types, Advantages and Applications*. n.d. URL: <https://www.blikai.com/blog/components-parts/what-is-bluetooth-types-advantages-and-applications>.
- [7] J. A. Calderón Velasco et al. «Biometría dactilar: una nueva alternativa de controlar efectivamente la asistencia a clases». En: *Investigación e Innovación en Ingenierías* 6.1 (2018), págs. 27-39. URL: <https://doi.org/10.17081/invinno.6.1.2773>.
- [8] Cuadernos de Seguridad. *¿Por qué cada huella dactilar es única?* 2018. URL: <https://cuadernosdeseguridad.com/2018/12/por-que-cada-huella-dactilar-es-unica/>.
- [9] Hardkernel Team. *ODROID-N2+ with 4GByte RAM*. n.d. URL: <https://www.hardkernel.com/shop/odroid-n2-with-4gbyte-ram-2/>.
- [10] Ivy eMeter. *LoRa and LoRaWAN technology: Principle and advantages and disadvantages*. n.d. URL: <https://www.ivy-emeter.com/html/lora-and-lorawan-technology-principle-and-advantages-and-disadvantages.html>.
- [11] Chris Karlof y David Wagner. «Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures». En: *Ad Hoc Networks* 1.2-3 (2003), págs. 293-315. URL: <https://www.sciencedirect.com/science/article/pii/S016740480500204X>.
- [12] MOKOSmart. *Historia del desarrollo de los dispositivos LoRa*. n.d. URL: <https://www.mokosmart.com/es/development-history-of-lora-devices/>.
- [13] Nordic. *Nordic Power Profile Kit II DataSheet*. URL: [https://www.mouser.es/datasheet/2/297/Power\\_Profiler\\_Kit\\_II\\_PB\\_v1\\_0-1928294.pdf](https://www.mouser.es/datasheet/2/297/Power_Profiler_Kit_II_PB_v1_0-1928294.pdf).
- [14] NÜO Planet. *¿Cómo funciona un lector de huella?* 2022. URL: <https://blog.nuoplanet.com/lector-de-huella-como-funciona>.
- [15] PHP Documentation Team. *Manual de PHP*. 2025. URL: <https://www.php.net/manual/es/>.
- [16] RF Page contributors. *Applications and Future of Near Field Communication*. n.d. URL: <https://www.rfpage.com/applications-near-field-communication-future/>.
- [17] S. Zahoor and A. U. Rahman. *Requirements, Deployments, and Challenges of LoRa Technology: A Survey*. 2023. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9842427/>.
- [18] SECOM News. *Fingerprint Recognition Technology and Its Practical Applications in 2024*. 2024. URL: <https://www.secom.vn/en/column/fingerprint-recognition-technology/>.
- [19] Sensbi. *What are the advantages and disadvantages of Zigbee systems?* Dic. de 2022. URL: <https://sensbi.com/2022/12/19/what-are-the-advantages-and-disadvantages-of-zigbee-systems/>.
- [20] STMicroelectronics. *DataSheet 2N700*. 2006. URL: <https://www.alldatasheet.com/datasheet-pdf/view/170639/STMICROELECTRONICS/2N700.html>.
- [21] TechTarget contributors. *RFID (radio frequency identification)*. n.d. URL: <https://www.techtarget.com/iotagenda/definition/RFID-radio-frequency-identification>.
- [22] The Things Network. *LoRaWAN Architecture*. 2024. URL: <https://www.thethingsnetwork.org/docs/lorawan/architecture/>.
- [23] Tiendatec. *Raspberry Pi 4 Modelo B - 2GB*. 2025. URL: <https://www.tiendatec.es/raspberry-pi/gama-raspberry-pi/1099-raspberry-pi-4-modelo-b-2gb-5056561800332.html>.
- [24] Vecoel. *Qué es LoRa, cómo funciona y características principales*. URL: <https://www.vencoel.com/que-es-lora-como-funciona-y-caracteristicas-principales/>.
- [25] Wikipedia contributors. *Bluetooth*. n.d. URL: <https://en.wikipedia.org/wiki/Bluetooth>.
- [26] Wikipedia contributors. *Zigbee*. n.d. URL: <https://en.wikipedia.org/wiki/Zigbee>.