



Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA

MÁSTER EN INGENIERÍA INFORMÁTICA

Integración de dispositivos médicos en la plataforma Blue Care

Alumno: David Pastor Pérez

Tutor UVA: Jesús María Vegas Hernández

Tutor de empresa: Luis Platón Arias (Blue)

Resumen

La integración de dispositivos médicos en plataformas digitales desempeña un papel crucial en la atención médica y monitorización de pacientes, permitiendo un cuidado más personalizado y eficiente. En este documento se presenta un proyecto que desarrolla una aplicación Android capaz de extraer datos de diferentes dispositivos médicos, específicamente de un glucómetro y un termómetro, mediante tecnología Bluetooth. Los datos recopilados se transmiten a servidores en la nube alojados en Amazon Web Service (AWS) pertenecientes a Blue, el envío de los datos se realiza utilizando el protocolo MQTT para una monitorización continua y remota.

Esta aplicación responde a la necesidad de solucionar la monitorización a personas vulnerables, como pueden ser las personas mayores o pacientes con enfermedades crónicas, permitiendo recolectar datos de salud en tiempo real. Al facilitar la obtención de datos desde el hogar de las personas se alinea con las tendencias actuales de atención sanitaria dirigidas a reducir y retrasar la institucionalización y ofrecer soluciones asequibles.

En este documento se abordan las diferentes fases del proyecto, desde el análisis y diseño hasta la implementación y pruebas. Se comentan los objetivos principales, como la conexión y extracción de datos a través de Bluetooth, transmisión segura y validación funcional de la aplicación. Además, se muestra el diseño flexible que permitirá la integración de nuevos dispositivos médicos, como pulsioxímetros o monitores de presión arterial, mostrando la aplicación como una solución escalable y adaptable.

Este proyecto busca ser una solución tecnológica y ha sido orientado hacia el mercado comercial, con la intención de convertirse en un producto viable en el sector.

Abstract

The integration of medical devices into digital platforms plays a crucial role in medical care and patient monitoring, enabling more personalised and efficient care. This paper presents a project that develops an Android application capable of extracting data from different medical devices, specifically a glucometer and a thermometer, using Bluetooth technology. The collected data is transmitted to cloud servers hosted on Amazon Web Service (AWS) belonging to Blue, using the MQTT protocol for continuous and remote monitoring.

This application responds to the need to solve the monitoring of vulnerable people, such as the elderly or patients with chronic diseases, allowing health data to be collected in real time. By facilitating the collection of data from people's homes, it is in line with current healthcare trends aimed at reducing and delaying institutionalisation and providing affordable solutions.

This paper discusses the different stages of the project, from analysis and design to implementation and testing. It discusses the main objectives, such as connecting and extracting data via Bluetooth, secure transmission and functional validation of the application. It also shows the flexible design that will allow the integration of new medical devices, such as pulse oximeters or blood pressure monitors, presenting the application as a scalable and adaptable solution.

This project aims to be a technological solution and is aimed at the commercial market, with the intention of becoming a viable product in the sector.

Índice general

Resumen	3
Abstract	5
1. Introducción	1
1.1. Contexto y motivación	1
1.2. Objetivos	2
1.2.1. Objetivos generales	3
1.2.2. Objetivos específicos	3
1.3. Descripción del proyecto	3
1.4. Estructura de la memoria	4
2. Estado del arte	5
2.0.1. Proyectos y tecnologías existentes	5
2.0.2. Proyectos Empresariales	6
2.0.3. Proyectos Europeos Active Assisted Living	7
2.0.4. Posibles alternativas al protocolo MQTT	8
2.0.5. Privacidad en la monitorización de datos de salud	9
3. Planificación y gestión	11
3.1. Planificación temporal	11

3.2. Gestión de riesgos	13
3.3. Presupuesto	17
3.3.1. Hardware	17
3.3.2. Software	18
3.3.3. Personal	18
3.3.4. Coste total	19
4. Análisis y diseño	21
4.1. Requisitos funcionales	21
4.2. Requisitos no funcionales	22
4.3. Casos de uso	23
4.3.1. Definición de actores	23
4.3.2. Diagrama de casos de uso	24
4.3.3. Descripción de casos de uso	25
4.4. Modelo de dominio	34
4.4.1. Descripción de las clases.	34
4.5. Diagrama de despliegue	36
4.6. Diseño base de datos	37
4.7. Diseño para la integración de otros dispositivos médicos	39
4.8. Prototipo inicial	40
5. Tecnologías y servicios utilizados	43
5.1. Android Studio	43
5.2. GitHub	44
5.3. Bluetooth	44
5.4. MQTT	45
5.5. MQTT Explorer	45

6. Implementación	47
6.1. Interfaz gráfica	47
6.2. Arquitectura y Gestión de Datos	51
6.2.1. Conexión con los dispositivos médicos	51
6.2.2. Conexión y obtención de datos termómetro	52
6.2.3. Conexión y obtención de datos glucómetro	53
6.2.4. Conexión con Broker MQTT en AWS	54
7. Pruebas	55
8. Conclusiones y trabajo futuro	61
Bibliografía	63
A. Manual de instalación	65
B. Manual de usuario	69
B.1. Iniciar sesión	69
B.2. Vincular nuevo dispositivo	70
B.3. Editar dispositivo	73
B.4. Desvincular dispositivo	73
B.5. Visualizar datos	75
B.6. Enviar datos manualmente	75
B.7. Visualizar información y editar parámetros	76

Índice de figuras

1.1. Arquitectura	4
3.1. Diagrama de Gantt	13
4.1. Diagrama de casos de uso	24
4.2. Modelo de dominio	34
4.3. Diagrama de despliegue	37
4.4. Diseño base de datos	37
4.5. Prototipo Login	41
4.6. Prototipo gestor Bluetooth	41
4.7. Prototipo glucómetro tabla	42
4.8. Prototipo termómetro tabla	42
5.1. Logotipo Android Studio	43
5.2. Logotipo GitHub	44
5.3. Logotipo Bluetooth	44
5.4. Logotipo MQTT	45
5.5. Logotipo MQTT Explorer	45
6.1. Login	48
6.2. Menú lateral	49

6.3. Gestor de dispositivos Bluetooth	49
6.4. Tabla con los datos obtenidos del glucómetro	50
6.5. Tabla con los datos obtenidos del termómetro	50
6.6. Información del dispositivo	51
A.1. Ejemplo activar orígenes desconocidos Android 14 (1)	66
A.2. Ejemplo activar orígenes desconocidos Android 14 (2)	67
A.3. Ejemplo activar orígenes desconocidos Android 14 (3)	67
A.4. Ejemplo activar orígenes desconocidos Android 14 (4)	68
B.1. Iniciar sesión	70
B.2. Buscar dispositivos	71
B.3. Vincular dispositivos 1	71
B.4. Vincular dispositivos 2	72
B.5. Vincular dispositivos 3	72
B.6. Editar dispositivos	73
B.7. Desvincular dispositivos	74
B.8. Desvincular dispositivos	74
B.9. Visualizar datos	75
B.10. Enviar datos manualmente	76
B.11. Visualizar información y editar parámetros	77

Índice de tablas

3.1. Reducción de personal debido a una baja laboral.	13
3.2. Demoras en la recepción de componentes.	14
3.3. Problemas en la integración de sistemas.	14
3.4. Modificaciones en los requisitos del proyecto.	14
3.5. Fallos en la comunicación interna del equipo.	15
3.6. Incompatibilidad de software.	15
3.7. Estimaciones inexactas de tiempo y costo.	15
3.8. Problemas de rendimiento del sistema.	16
3.9. Dificultades excesivas en el desarrollo.	16
3.10. Desarrollo incorrecto de interfaces.	16
3.11. Pérdida de datos.	17
3.12. Coste hardware.	18
3.13. Coste software.	18
3.14. Coste personal.	18
3.15. Coste total.	19
4.1. Requisitos funcionales.	22
4.2. Requisitos no funcionales.	23
4.3. ACT-001 Administrador.	23
4.4. ACT-002 Usuario.	23

4.5. ACT-003 Broker MQTT.	23
4.6. UC-001 Iniciar sesión.	25
4.7. UC-002 Vincular dispositivo.	26
4.8. UC-003 Desvincular dispositivo.	26
4.9. UC-004 Editar dispositivo.	27
4.10. UC-005 Visualizar información Tablet.	27
4.11. UC-006 Buscar dispositivos.	28
4.12. UC-007 Visualizar datos glucómetro.	28
4.13. UC-008 Enviar datos manualmente glucómetro.	29
4.14. UC-009 Enviar datos manualmente termómetro.	29
4.15. UC-010 Visualizar datos termómetro.	30
4.16. UC-011 Tomar medida con glucómetro.	30
4.17. UC-012 Tomar medida con termómetro.	31
4.18. UC-013 Enviar datos Broker MQTT.	31
4.19. UC-014 Almacenar datos glucómetro.	32
4.20. UC-015 Almacenar datos termómetro.	32
4.21. UC-016 Cambiar parámetros de configuración.	33
6.1. Health Thermometer Service Characteristics.	52
6.2. Glucose service characteristics.	53
7.1. Batería de pruebas para el caso CPSYS-001 Iniciar sesión.	56
7.2. Batería de pruebas para el caso CPSYS-002 Buscar dispositivos.	56
7.3. Batería de pruebas para el caso CPSYS-003 Vincular dispositivo.	56
7.4. Batería de pruebas para el caso CPSYS-004 Desvincular dispositivo.	56
7.5. Batería de pruebas para el caso CPSYS-005 Editar dispositivo.	57
7.6. Batería de pruebas para el caso CPSYS-006 Visualizar información Tablet.	57

7.7. Batería de pruebas para el caso CPSYS-007 Tomar medida glucómetro	57
7.8. Batería de pruebas para el caso CPSYS-008 Tomar medida termómetro. . . .	58
7.9. Batería de pruebas para el caso CPSYS-009 Visualizar datos glucómetro. . .	58
7.10. Batería de pruebas para el caso CPSYS-010 Visualizar datos termómetro. . .	58
7.11. Batería de pruebas para el caso CPSYS-011 Enviar datos manualmente glucómetro.	58
7.12. Batería de pruebas para el caso CPSYS-012 Enviar datos manualmente termómetro.	59

Capítulo 1

Introducción

La integración de dispositivos médicos en plataformas digitales representa un avance en la mejora de la atención médica y la monitorización de los pacientes, lo que permite un cuidado más personalizado y eficiente. En un mundo cada vez más conectado, el desarrollo de aplicaciones que permitan la extracción y gestión de datos provenientes de sensores médicos supone un avance significativo.

El objetivo de este proyecto es desarrollar una aplicación Android capaz de extraer datos de un glucómetro y un termómetro mediante la tecnología Bluetooth, y publicar la información en los servidores que tenga Blue Care en la nube de AWS, utilizando el protocolo MQTT. Esta solución facilita la monitorización continua y remota, aportando valor tanto a pacientes como a profesionales de la salud.

En el siguiente documento, se presenta una visión del proyecto, desde el planteamiento hasta la descripción de las diferentes tareas realizadas, los objetivos y la planificación del trabajo junto a su análisis y diseño. Además, se aborda el estado del arte en cuanto a las tecnologías y proyectos relacionados. Por último, se hablará de los resultados obtenidos y la discusión de estos.

1.1. Contexto y motivación

En la sociedad actual, la atención y monitorización de las personas vulnerables, como pueden ser ancianos o personas con alguna discapacidad, es un desafío constante. Con el envejecimiento de la población y el aumento de enfermedades crónicas, es necesario desarrollar soluciones tecnológicas que nos permitan un seguimiento continuo y eficaz a estos pacientes, ayudándolos a una mejor calidad de vida. La empresa Blue, dispone de una plataforma dedicada a la monitorización de estas personas en situación de vulnerabilidad, busca integrar diferentes dispositivos médicos en su plataforma para poder mejorar la calidad de vida.

Este proyecto se centra en la monitorización de la glucosa y la temperatura corporal. Un

control regular de los niveles de glucosa en sangre puede ser crucial para los pacientes diabéticos. Ya que puede permitir a los profesionales de la salud ajustar los tratamientos y prevenir posibles complicaciones futuras. De igual manera, la medición de la temperatura corporal es un indicador importante en la monitorización de la salud general del paciente. Aunque el enfoque principal está en estos dos parámetros, el proyecto tiene el potencial de extenderse hacia la integración de otros dispositivos médicos, como puede ser un pulsioxímetro para medir la saturación de oxígeno en sangre, un monitor de presión arterial, un sensor de frecuencia cardíaca, etc. La inclusión de estos dispositivos permitiría un seguimiento más integral del estado de salud del paciente, proporcionando información que podría permitir una detección temprana de problemas. El principal objetivo es convertirse en un producto comercial sólido destinado a ofrecer soluciones innovadoras en el ámbito de la salud. Se ha diseñado pensando en un mercado real, con la intención de que sea utilizado por consumidores.

Sin embargo, muchos pacientes tienen dificultades para poder llevar a cabo un control riguroso tanto de sus niveles de glucosa en sangre como de su temperatura corporal. Por lo tanto, la integración y monitorización remota y en tiempo real puede ser crucial para esto.

El proyecto se centra en el desarrollo de una aplicación Android que conecta con un glucómetro comercial y un termómetro comercial, ambos mediante tecnología Bluetooth, permitiendo la extracción y gestión de los datos. Estos datos se enviarán mediante MQTT a un servidor en la nube de AWS.

La tendencia actual en el cuidado de personas vulnerables, como mayores, personas con discapacidad o enfermedades mentales, se enfoca en mantener su permanencia en los hogares, evitando o retrasando su posible institucionalización. Esto se refleja en la Estrategia estatal para un nuevo modelo de cuidados en la comunidad (2024-2030) [19].

Muchas de estas personas requieren un seguimiento diario de sus parámetros, como puede ser la tensión, temperatura, saturación de oxígeno, niveles de glucosa, peso, etc. Por ello, es necesario desarrollar una solución de bajo coste que permita la obtención de todos estos parámetros desde el hogar.

Esta solución debe ser asequible para lograr una alta difusión y debe de funcionar de manera desatendida, dada la limitada habilidad de los usuarios potenciales. La mayoría de las opciones en el mercado no cumplen con estos requisitos, lo que justifica el desarrollo de una solución a medida.

La motivación de este proyecto radica en la necesidad de proporcionar herramientas tecnológicas que apoyen el cuidado de personas vulnerables, mejorando su calidad de vida y la carga en el sistema de salud. La integración de diferentes dispositivos en una plataforma permite una atención mucho más personalizada y proactiva que puede detectar enfermedades en etapas tempranas.

1.2. Objetivos

A continuación, se presentan los objetivos generales y específicos que orientarán el desarrollo y la ejecución del proyecto.

1.2.1. Objetivos generales

Desarrollo de aplicación Android para la extracción mediante bluetooth de los datos obtenidos de dispositivos médicos y su publicación en los servidores en la nube de Blue Care mediante protocolo MQTT.

1.2.2. Objetivos específicos

- Diseñar la interfaz de usuario de la aplicación.
- Implementar la conectividad Bluetooth entre la aplicación Android y dispositivos médicos comerciales.
- Diseñar e implementar la extracción de datos en la aplicación Android.
- Integrar el protocolo MQTT en la aplicación para la publicación de los datos en los servidores de la nube.
- Validar la funcionalidad y seguridad de la aplicación.

1.3. Descripción del proyecto

El proyecto se centra en el desarrollo de una solución que permita extraer y gestionar de manera eficiente los datos provenientes de un glucómetro y un termómetro específicos, con el objetivo de facilitar la monitorización remota de pacientes y mejorar la atención médica.

La parte central del proyecto es el desarrollo de una aplicación Android, programada en Kotlin, diseñada para la extracción de datos mediante Bluetooth desde un glucómetro y un termómetro de la marca Beurer. Las razones principales para utilizar Kotlin en este proyecto, en lugar de otros lenguajes como Java, es principalmente que Kotlin es el lenguaje recomendado por Google para el desarrollo de aplicaciones Android, lo que garantiza un soporte sólido y compatibilidad con las últimas librerías. Además, Kotlin ofrece una sintaxis más moderna y concisa que permite escribir código más limpio y menos propenso a errores. También es totalmente compatible con Java pudiendo utilizar librerías escritas en Java sin problemas.

Esta aplicación actuará como intermediario entre los dispositivos médicos y la plataforma en la nube de Blue Care, permitiendo la recopilación de datos de los pacientes de manera automatizada y en tiempo real. La aplicación se encargará de establecer conexiones seguras con los dispositivos, obtener los datos mediante Bluetooth, transformarlos y transmitirlos a los servidores en la nube utilizando el protocolo MQTT.

Para garantizar la seguridad y privacidad de los datos sensibles, se implementarán medidas de cifrado y autenticación en todas las etapas de comunicación, desde la extracción hasta su transmisión y almacenamiento. En la Figura (1.1) se puede observar la arquitectura general de la aplicación. En ella se puede ver cómo el usuario utilizado cualquiera de los dispositivos

disponibles, los cuales envían los datos mediante Bluetooth a la tablet. La tablet almacena los datos en la Base de datos local y los lo envía mediante MQTT al Broker alojado en AWS.

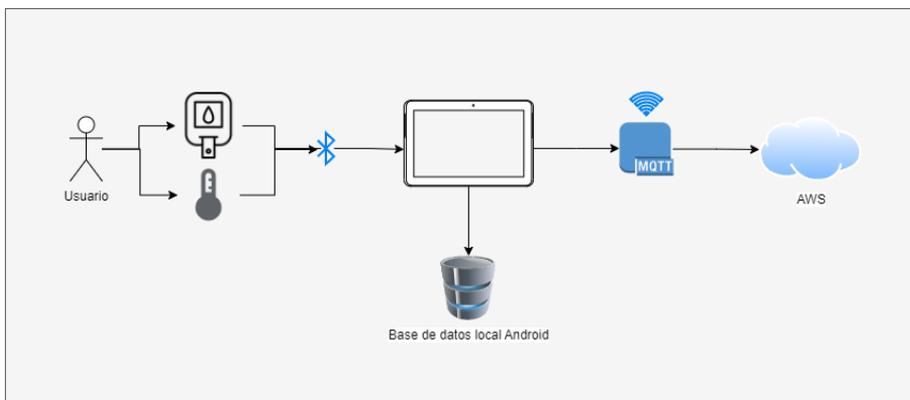


Figura 1.1: Arquitectura

1.4. Estructura de la memoria

- Capítulo 1: Introducción. En el aparecen la estructura de la memoria, contexto y los objetivos del proyecto.
- Capítulo 2: Estado del arte. En el aparecen los diferentes proyectos y tecnologías actuales.
- Capítulo 3: Planificación y gestión. Información sobre como se ha planificado y gestionado el proyecto, junto a sus riesgos incluyendo un presupuesto del coste.
- Capítulo 4: Análisis y diseño. Toda la información sobre la elicitación de requisitos, casos de uso, modelo de dominio, diagramas de secuencia, diagrama de despliegue y el diseño de la base de datos.
- Capítulo 5: Tecnologías y servicios utilizados. Descripción de las herramientas y tecnologías utilizadas para el desarrollo.
- Capítulo 6: Implementación. Explicación de la toma de decisiones.
- Capítulo 7: Pruebas. Pruebas realizadas junto a el resultado obtenido en ellas.
- Capítulo 8: Conclusiones. Recoge las conclusiones finales sobre la realización del proyecto, tanto los objetivos cumplidos como posibles líneas futuras de trabajo.
- Bibliografía. Referencias bibliográficas consultadas durante la realización del proyecto.
- Apéndices. Contiene los manuales para poder instalar y utilizar la aplicación.

Capítulo 2

Estado del arte

En la integración de dispositivos médicos con plataformas digitales, se han realizado en los últimos años diversos avances que han permitido transformar la forma en la que se recopilan, gestionan y utilizan los datos de salud. En este proyecto nos vamos a centrar en los glucómetros, dispositivos críticos para la monitorización de glucosa en sangre.

Aplicaciones de salud y glucómetros: existen diferentes aplicaciones móviles diseñadas para la monitorización de la actividad deportiva y la salud, las cuales se conectan con un “wearable”. Estas aplicaciones abarcan desde el control del ritmo cardíaco hasta el control de glucosa en sangre, monitorización de la actividad física, etc. Un ejemplo de estos dispositivos podría ser la aplicación Android para ECG Istel, esta aplicación permite la conexión de los dispositivos de electrocardiograma Istel a través de Bluetooth, facilitando la monitorización remota de la actividad cardiaca de los pacientes y gestionar estos datos en su plataforma digital.

En el ámbito específico de los glucómetros, se han desarrollado varias soluciones que integran conectividad Bluetooth para facilitar la transferencia de datos entre los dispositivos móviles. Estos sistemas permiten a los usuarios enviar lecturas de glucosa en sangre a un dispositivo móvil que almacena los datos. Algunas de estas aplicaciones también incluyen características adicionales como la integración con podómetros para ayudar a monitorizar la actividad física de la persona.

2.0.1. Proyectos y tecnologías existentes

Revisando literatura en páginas como Google docs se han encontrado diferentes proyectos en el ámbito académico, que comentaremos a continuación.

Algunos proyectos optaron por diseñar sus propios dispositivos para medir los niveles de glucosa. Por ejemplo, en este proyecto [7] optaron por utilizar un microcontrolador ESP32. Este microcontrolador se comunicaba con un módulo GSM o una aplicación móvil para almacenar los datos en una base de datos.

En este proyecto [20] se utiliza Android 4 y Bluetooth 2.1 SPP para conectar un glucómetro con una aplicación Android.

La APP se integrará con un medidor de glucosa mediante Bluetooth, permitiendo al usuario enviar muestras de sangre desde el medidor al dispositivo móvil. La aplicación registra y almacena lecturas importantes como los niveles de azúcar en la sangre (medidos en mmol/L), la fecha y la hora de la lectura. Además, cuenta con un podómetro para rastrear la actividad física del usuario y asegurarse de que cumplen con los 30 minutos de ejercicio diario recomendados para mejorar la circulación sanguínea.

Los datos se suben a firebase, también se almacenan en un SQLite, habla también de una API, pero no se indica el dispositivo que está utilizando para el proyecto.

Tecnologías de comunicación inalámbrica: el uso de tecnologías inalámbricas, como Bluetooth Low Energy (BLE) y WI-FI, han permitido la conectividad entre dispositivos médicos y dispositivos móviles de una manera eficiente y en tiempo real.

Protocolos de comunicación: el protocolo MQTT gana popularidad en los últimos años en el ámbito del IoT y la salud digital debido a su eficiencia y capacidad de soportar la comunicación entre sistemas distribuidos.

Aunque existan diferentes avances en este campo, todavía existen desafíos importantes en la integración de dispositivos médicos en las plataformas digitales, como la interoperabilidad entre dispositivos de diferentes fabricantes, estandarización de formato de datos, garantía en la seguridad de la información. Este proyecto se sitúa en este contexto, aprovechando los conocimientos y tecnologías existentes para poder desarrollar una solución innovadora y eficaz en este campo.

2.0.2. Proyectos Empresariales

Es importante destacar las diferentes empresas que están en el sector de la telemedicina y teleasistencia, cada una tiene diferentes enfoques y tecnologías que responden a diferentes necesidades en el cuidado de la salud, especialmente en el monitoreo de personas vulnerables y mayores.

Comitas e-health[6], empresa española con más de 25 años de experiencia, ha sido pionera en el sector de la telemedicina diagnosticada. Durante su trayectoria, ha centrado su actividad en el proveedor exclusivo para las fuerzas armadas, destacándose por sus maletines. Estos dispositivos ofrecen soporte médico y diagnósticos en cualquier lugar, incluso en entornos remotos o de difícil acceso. Recientemente ha ampliado su oferta al sector civil, adaptando sus soluciones a residencias de mayores, prisiones y barcos pesqueros. Además, recientemente lanzo un servicio de radiografía y ecografía a domicilio en Madrid y alrededores, utilizando unidades móviles medicalizadas.

Essence SmartCare España[23], parte del Essence Group, se especializa en soluciones avanzadas de teleasistencia y telemedicina dirigidas principalmente a proveedores de servicios para personas mayores. Esta consolidada como uno de los principales fabricantes de sistemas

de teleasistencia reactiva (PERS) y de soluciones que promueven la vida independiente de las personas mayores. Ofrece una plataforma integral que permite hacer un seguimiento completo de la salud y movilidad de los pacientes en sus hogares, es adaptable a las necesidades cambiantes de los pacientes y proveedores de atención médica.

MESI [18], ha innovado en telemedicina a través de el MESI mTABLET. Esta solución ofrece una evaluación médica predictiva con una amplia gama de módulos de hardware, extensiones software y diferentes opciones de integración. Entre las capacidades que ofrece se encuentra el ECG de 12 derivaciones, el índice tobillo-brazo, índice dedo del pie-brazo, espirometría básica, monitoreo de presión arterial y oximetría de pulso. La versatilidad de opciones la convierte en una herramienta esencial para los profesionales de la salud.

2.0.3. Proyectos Europeos Active Assisted Living

A continuación se describen algunos proyectos europeos del tipo AAL, enfocados en mejorar la calidad de vida de las personas mayores a través de tecnologías.

AHEAD

El proyecto AHEAD [1] tiene como objetivo sensores fisiológicos miniaturizados en dispositivos de gafas auditivas junto con sensores ambientales. Estos sensores recopilan datos fisiológicos y del entorno, que se procesan mediante algoritmos específicos para proporcionar información relevante. La finalidad es ofrecer servicios personalizados para mejorar la calidad de vida diaria, recalibrar dispositivos auditivos a distancia y monitorear el estado físico.

Objetivos: El objetivo principal del proyecto AHEAD es mejorar la calidad de vida de las personas mayores, ayudándolas a llevar una vida activa e independiente.

Resultados esperados e impacto: Se espera que los servicios proporcionados ayuden a los usuarios a vivir de forma más independiente, promoviendo la confianza y una vida saludable, y mejorando la comunicación social.

Ageing in Balance (AiB)

El proyecto AiB [2] busca desarrollar una nueva solución para prevenir las caídas en personas mayores. AiB se centrará en desarrollar un modelo innovador que incluya factores de riesgo intrínsecos y extrínsecos, utilizando datos de sensores ambientales y tecnologías complementarias como juegos cognitivos. Este modelo ofrecerá una evaluación clara del riesgo de caídas, permitiendo desarrollar herramientas de prevención.

Objetivos: El objetivo es prevenir las caídas de personas mayores y, con ello, reducir los costes asociados a su atención y aumentar la calidad de vida.

Resultados esperados e impacto: Los resultados esperados incluyen herramientas para la evaluación del riesgo de caídas, tanto para uso profesional como para autoevaluación.

Además, el proyecto desarrollará software de prevención para motivar a los usuarios a cambiar su comportamiento.

AMICA

El proyecto AMICA [3] busca emular la consulta médica a domicilio mediante la auscultación y la entrevista. Se obtienen diariamente señales fisiológicas a través de un sensor ad hoc, y esta información se complementa con la proporcionada por el paciente mediante un dispositivo móvil específico. Al combinar ambos tipos de información, el sistema puede activar alarmas médicas, ajustar aspectos del tratamiento o estilo de vida del paciente, o incluso sugerir una hospitalización.

Objetivos: El proyecto AMICA (Autonomía, Motivación y Autogestión Individual para pacientes con EPOC) tiene como principal objetivo la gestión de la enfermedad pulmonar obstructiva crónica (EPOC) y la atención médica de los pacientes que la padecen. AMICA pretende ofrecer soluciones tecnológicas para la gestión a largo plazo de la EPOC mediante TIC innovadoras. Estas soluciones incluyen la detección temprana de exacerbaciones.

Resultados esperados e impacto: Se espera que AMICA proporcione indicadores medibles para la detección temprana de exacerbaciones a través de un dispositivo móvil y un sensor ad hoc. Además, se prevé una mejora en la calidad de vida de los pacientes con EPOC mediante la reducción de hospitalizaciones, lo que a su vez disminuye los costes asociados a la enfermedad.

2.0.4. Posibles alternativas al protocolo MQTT

El protocolo MQTT es una opción eficiente y ligera para la transmisión de datos, pero existen otros protocolos que también podrían ser considerados. A continuación, se presentan algunas alternativas viables:

- **CoAP (Constrained Application Protocol)[16]:** es un protocolo diseñado para dispositivos con recursos limitados, muy utilizado en IoT. Está optimizado para funcionar sobre redes con baja capacidad de procesamiento y ancho de banda, al igual que MQTT puede funcionar sobre UDP.
- **AMQP (Advanced Message Queuing Protocol)[8]:** protocolo orientado a mensajes que utiliza principalmente en entornos empresariales para aplicaciones más robustas, donde se requiere fiabilidad en la entrega de mensajes. Soporta más funcionalidades que MQTT teniendo encolamiento avanzado, transacciones y enrutamiento. Sin embargo estas características pueden añadir complejidad y sobrecarga, lo que lo hace menos eficiente.
- **DDS (Data Distribution Service)[9]:** es un protocolo especialmente diseñado para aplicaciones distribuidas en tiempo real, tiene capacidades integradas de descubrimiento automático de dispositivos, es particularmente útil en sistemas donde se requieren datos

en tiempo real y alta fiabilidad, como en aplicaciones críticas de salud. Sin embargo, su complejidad y consumo de recursos son mayores en comparación a MQTT.

2.0.5. Privacidad en la monitorización de datos de salud

La privacidad de datos es una parte fundamental en cualquier sistema que maneje información sensible, y adquiere mucha más importancia en este caso, donde se monitorizan datos críticos como los niveles de glucosa y la temperatura corporal, es necesario garantizar que los datos sean gestionados con el mayor nivel de seguridad para proteger la confidencialidad de los usuarios.

Protección de datos en dispositivos médicos

Los dispositivos como glucómetros y termómetros que nos permiten una transmisión de datos a través de Bluetooth representan un avance, pero a su vez también es un riesgo potencial en términos de seguridad. Es esencial que la comunicación entre los dispositivos y la aplicación móvil estén debidamente encriptadas para evitar el acceso no autorizado. La utilización de protocolos como BLE implementan medidas adicionales para asegurar la transmisión de datos y prevenir ataques de intermediarios. Una de las principales medidas es el uso de un proceso de emparejamiento seguro, que incluye la generación de un código temporal, durante cada vinculación. Este proceso de emparejamiento y cifrado además asegura la protección de datos utilizando técnicas como el cifrado AES-CCM de 128 bits.

Regulaciones y normativas de privacidad

Otro de los aspectos clave es cumplir con las regulaciones de protección de datos, como el Reglamento General de Protección de Datos. Estas normativas [10] establecen directrices claras sobre como deben manejarse los datos estando estos en la categoría de los datos sensibles, concretamente datos de la salud, asegurando que la información solo se utilice para los fines acordados. El proyecto debe de garantizar transparencia en el tratamiento de datos, especificando como se almacenan, quien accede y durante cuanto tiempo estarán. Para asegurar la protección de estos datos, se deben seguir las siguientes consideraciones:

- Consentimiento explícito [10]: los usuarios deben dar su consentimiento explícito para el tratamiento de sus datos.
- Transparencia [11]: se debe garantizar la total transparencia en el tratamiento de datos.
- Principio de minimización [12]: solo deben recogerse los datos estrictamente necesarios.
- Cifrado y medidas de seguridad [13]: los datos deben protegerse mediante técnicas de cifrado y autenticación.
- Responsabilidad y trazabilidad [14]: se deben asegurar que los datos cuenten con trazabilidad y registro de accesos y modificaciones.

Anonimización y cifrado

Un factor relevante es la implementación de anonimización y cifrado de los datos. Con la anonimización se permite que los datos sean procesados sin vincularse directamente a la identidad de un usuario. Además, el uso de tecnologías de cifrado es fundamental para asegurar la protección de la información sensible.

Transferencia a través de MQTT

Para la utilización del protocolo MQTT en la transmisión de datos es necesario que utilice TLS, para cifrar las comunicaciones y evitar que puedan ser interceptados o manipulados durante su envío. Además, es necesaria una autenticación para verificar la identidad de los dispositivos y servidores que participan en la comunicación

Capítulo 3

Planificación y gestión

3.1. Planificación temporal

Antes de comenzar con el proyecto se tiene que planear de manera temporal las distintas fases del mismo, para esta planificación se tiene en cuenta que el desarrollador principal trabaja a jornada completa en una empresa y el mayor tiempo de desarrollo se agrupara durante festivos y fines de semana.

Para realizar la planificación se ha utilizado la metodología en cascada que es un modelo lineal de diseño de software, la decisión para utilizar esta metodología es debido a que es la que mejor encaja con los recursos disponibles, ya que solo existe una persona para el desarrollo completo del proyecto, y se tiene que ir desarrollando fase a fase.

El proyecto se divide en las siguientes fases: requisitos, análisis, diseño, programación y pruebas.

Requisitos (4 Semanas): en esta fase se recopilan y documentan los requisitos esenciales del proyecto. Esto incluye la revisión de la literatura existente para poder comprender el estado del arte, el estudio de las tecnologías relevantes, la identificación de los posibles candidatos de glucómetros que se utilizarán, y la especificación detallada de los requisitos del sistema.

Análisis (1 Semanas): durante esta fase, se lleva a cabo un análisis exhaustivo de los requisitos previamente definidos. El objetivo es asegurar que se entienden claramente y que se pueden cumplir con ellos.

Diseño (3 Semanas): en esta fase, se desarrollan las soluciones arquitectónicas y de interfaz necesarias para el sistema. Esto incluye el diseño de la interfaz del usuario, la arquitectura de comunicación Bluetooth, el sistema para el almacenamiento de los datos. Esta fase es elemental para poder establecer la estructura del sistema y como se interactúa con los diferentes componentes

Programación (7 Semanas): esta fase es la más larga del proyecto y abarca la implementación de todas las funcionalidades definidas en las fases anteriores. Se incluye la implementación del gestor Bluetooth, visualización de los datos tanto del termómetro como el glucómetro, configuración de la comunicación MQTT, y el desarrollo de las funcionalidades de comunicación Bluetooth y el procesamiento y almacenamiento de datos.

Pruebas (2 Semanas): en esta fase se realizarán las pruebas necesarias para poder asegurar que el sistema funciona correctamente y cumple con todos los requisitos especificados en las primeras fases, el objetivo es identificar y corregir cualquier defecto o error que pueda existir antes de su entrega final.

Documentación (3 Semanas): en esta última fase, se lleva a cabo la documentación, donde se recopila y estructura toda la información necesaria para que el proyecto sea comprensible para terceros. Esto incluye manuales de usuario, documentación técnica y cualquier otro tipo de información.

En la Figura (3.1) se puede observar el diagrama de Gantt creado para representar la planificación. Este diagrama está dividido en las diferentes fases mencionadas con anterioridad y, como se puede observar, todas las tareas son secuenciales, ya que no se podía paralelizar ninguna.

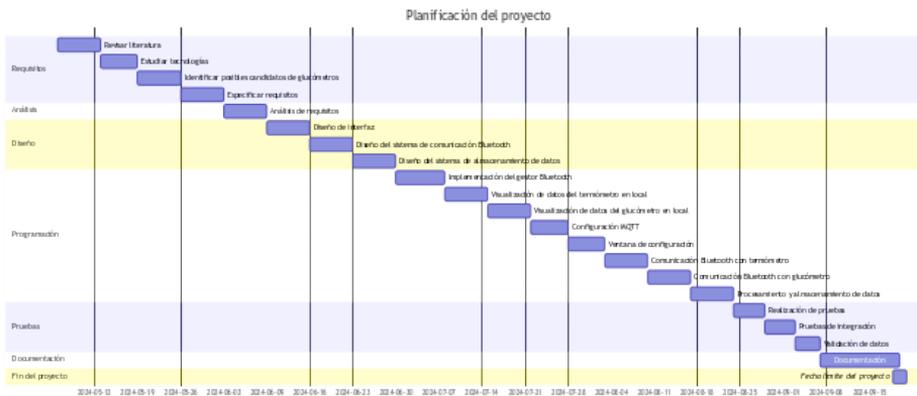


Figura 3.1: Diagrama de Gantt

3.2. Gestión de riesgos

En este apartado vamos a exponer los riesgos que puede tener la realización del proyecto. Todo proyecto tiene unos riesgos, algunos de ellos se pueden asumir, pero para otros tienes que tener un plan, el cual busque reducir el riesgo lo máximo posible, y si no es posible reducir el riesgo tienes que tener pensado cómo mitigar lo máximo posible las consecuencias del mismo. A continuación se muestra una serie de tablas en las que se recogen cada uno de los riesgos junto a su plan de contingencia.

Riesgo	Reducción de personal debido a una baja laboral.
Probabilidad	Media.
Causa	Enfermedad o problemas personales del personal.
Consecuencia	Prolongación de los plazos estimados.
Impacto	Alto.
Estrategia	Aceptación del riesgo.
Plan de contingencia	Asumir el impacto del riesgo sin implementar medidas adicionales.

Tabla 3.1: Reducción de personal debido a una baja laboral.

3.2. GESTIÓN DE RIESGOS

Riesgo	Demoras en la recepción de componentes.
Probabilidad	Alta.
Causa	Problemas de suministro con los proveedores.
Consecuencia	Atraso en el cronograma del proyecto.
Impacto	Alto.
Estrategia	Mitigación del riesgo.
Plan de contingencia	Establecer relaciones con proveedores alternativos o mantener un inventario adicional.

Tabla 3.2: Demoras en la recepción de componentes.

Riesgo	Problemas en la integración de sistemas.
Probabilidad	Media.
Causa	Incompatibilidades entre módulos o tecnologías empleadas.
Consecuencia	Aumento en el tiempo requerido para el desarrollo.
Impacto	Medio.
Estrategia	Mitigación del riesgo.
Plan de contingencia	Realizar pruebas de integración frecuentes desde las etapas iniciales del proyecto para detectar problemas a tiempo.

Tabla 3.3: Problemas en la integración de sistemas.

Riesgo	Modificaciones en los requisitos del proyecto.
Probabilidad	Media.
Causa	Cambios en las necesidades del cliente o en los objetivos del proyecto.
Consecuencia	Necesidad de re-trabajo y posibles retrasos.
Impacto	Alto.
Estrategia	Mitigación del riesgo.
Plan de contingencia	Implementar un proceso de control de cambios que evalúe el impacto antes de aceptar cualquier modificación en los requisitos.

Tabla 3.4: Modificaciones en los requisitos del proyecto.

Riesgo	Fallos en la comunicación interna del equipo.
Probabilidad	Baja.
Causa	Insuficientes reuniones o herramientas de comunicación inadecuadas.
Consecuencia	Malentendidos y errores en la ejecución del proyecto.
Impacto	Medio.
Estrategia	Mitigación del riesgo.
Plan de contingencia	Establecer reuniones periódicas y asegurar la disponibilidad de herramientas de comunicación eficaces para todo el equipo.

Tabla 3.5: Fallos en la comunicación interna del equipo.

Riesgo	Incompatibilidad de software.
Probabilidad	Media.
Causa	Diferentes versiones de software o dependencias conflictivas.
Consecuencia	Retrasos en la implementación y aumento de costos.
Impacto	Alto.
Estrategia	Mitigación del riesgo.
Plan de contingencia	Establecer un entorno de desarrollo estándar desde el inicio del proyecto para evitar problemas de compatibilidad.

Tabla 3.6: Incompatibilidad de software.

Riesgo	Estimaciones inexactas de tiempo y costo.
Probabilidad	Media.
Causa	Planificación inicial deficiente.
Consecuencia	Retrasos en el cronograma o incremento en el presupuesto.
Impacto	Alto.
Estrategia	Mitigación del riesgo.
Plan de contingencia	Incluir márgenes de tiempo y presupuesto adicionales en cada fase del proyecto.

Tabla 3.7: Estimaciones inexactas de tiempo y costo.

3.2. GESTIÓN DE RIESGOS

Riesgo	Problemas de rendimiento del sistema.
Probabilidad	Baja.
Causa	Diseño de software ineficiente.
Consecuencia	Aumento en los plazos de entrega.
Impacto	Alto.
Estrategia	Mitigación del riesgo.
Plan de contingencia	Realizar pruebas de carga y optimización en los algoritmos que demandan más recursos.

Tabla 3.8: Problemas de rendimiento del sistema.

Riesgo	Dificultades excesivas en el desarrollo.
Probabilidad	Media.
Causa	Falta de experiencia del equipo.
Consecuencia	Prolongación de los plazos previstos.
Impacto	Alto.
Estrategia	Mitigación del riesgo.
Plan de contingencia	Reservar tiempo adicional en las fases más complejas del proyecto.

Tabla 3.9: Dificultades excesivas en el desarrollo.

Riesgo	Desarrollo incorrecto de interfaces.
Probabilidad	Baja.
Causa	Falta de coordinación y comunicación.
Consecuencia	Necesidad de rehacer partes del desarrollo.
Impacto	Bajo.
Estrategia	Mitigación del riesgo.
Plan de contingencia	Crear prototipos previos para clarificar y validar las interfaces antes de su implementación final.

Tabla 3.10: Desarrollo incorrecto de interfaces.

Riesgo	Pérdida de datos.
Probabilidad	Baja.
Causa	Fallos en los dispositivos de almacenamiento.
Consecuencia	Prolongación de los plazos y posible pérdida de información crítica.
Impacto	Alto.
Estrategia	Eliminación del riesgo.
Plan de contingencia	Realizar copias de seguridad triples al finalizar cada jornada (almacenamiento en la nube, dispositivos externos e internos).

Tabla 3.11: Pérdida de datos.

Para minimizar la mayoría de estos riesgos, es fundamental mantener una comunicación constante y efectiva con el cliente. También debemos de tener una revisión periódica de los avances, para intentar ajustarnos a los tiempos de planificación. Para la mitigación de de posibles riesgos también se tiene que tener un margen en todas las etapas del proyecto, esto se le conoce como ‘float’.

Algunos de los riesgos previstos llegaron a producirse, comenzando por la demora en la obtención de componentes (3.2), debido a que el proveedor tardo en entregarnos muestras del termómetro y del glucómetro. Para mitigar este riesgo, se obtuvieron los dispositivos de otro proveedor para poder disponer de ellos a tiempo.

Otro de los riesgos que se produjo fue la dificultad excesiva en el desarrollo de algunas funcionalidades (3.9), ya que la experiencia con la comunicación Bluetooth y MQTT era limitada, lo que generó retrasos. No obstante, como estos tiempos ya habían sido estimados, el riesgo fue mitigado adecuadamente.

3.3. Presupuesto

Todo proyecto software tiene unos costes, los cuales se pueden dividir en tres partes: hardware, software y personal. A continuación vamos a realizar un desglose de precios de los tres componentes.

3.3.1. Hardware

En este apartado se recopila todo el equipo hardware (3.12) utilizado y necesario para el desarrollo del proyecto. Para calcular el coste por hora de los dispositivos, se considera su vida útil y un uso laboral de 8 horas diarias, cinco días a la semana. Un portátil, por ejemplo, tiene una vida útil estimada de 5 años. En cuanto a los dispositivos móviles suelen tener una vida útil entre 18 y 24 meses en función de la calidad del dispositivo. Los dispositivos como el glucómetro recomiendan cambiarlos más o menos de 4 a 5 años. Para los cálculos de tiempo

3.3. PRESUPUESTO

se estima un total de 150 horas que es el equivalente a los créditos del TFM, en cuanto a los elementos que se utilizan por unidades como las tiras y las lancetas se consumieron un total de 50 unidades realizando pruebas.

Descripción	Coste unitario	Coste por hora	Coste
Portátil - ASUS VivoBook 15	478 €	0,049 €	7,35 €
Tablet - Samsung Galaxy tab A9	195 €	0,050 €	7,5 €
Glucómetro Beurer GL49	40 €	0,0042 €	0,63 €
Termómetro Beurer FT95	45 €	0,0046 €	0,69 €
Tiras reactivas Beurer 50 (Unidades)	30 €	0,6 €	30 €
Lancetas Beurer 100(Unidades)	12 €	0,12 €	6 €
Solución control OneTouch	4 €	- €	4 €
Total			56,17 €

Tabla 3.12: Coste hardware.

3.3.2. Software

En este apartado se incluye todo el coste de las herramientas software (3.13) utilizadas durante el desarrollo.

Descripción	Coste unitario
Android Studio	0 €
AWS - MQTT	0,70 \$ (por millón de mensajes)
Total	0 €

Tabla 3.13: Coste software.

3.3.3. Personal

En este apartado se tiene en cuenta el coste total del personal (3.14) implicado en el desarrollo del proyecto, en este caso solo se dispone de una persona.

Descripción	Coste unitario	Horas
Salario programador sénior	45 €/H	150
Total	6750 €	

Tabla 3.14: Coste personal.

3.3.4. Coste total

A continuación se muestra la suma de todos los presupuestos (3.15), con el fin de obtener el coste total del proyecto software.

Descripción	Coste
Coste hardware	56,17 €
Coste software	0 €
Coste personal	6750 €
Total	6806,17 €

Tabla 3.15: Coste total.

Capítulo 4

Análisis y diseño

4.1. Requisitos funcionales

Los requisitos funcionales (4.1) especifican el comportamiento que el sistema debe tener para cumplir con los objetivos del proyecto. En esta sección se detallan todos los requisitos del proyecto.

Código	Requisito	Descripción
FRQ-001	Inicio de sesión	El sistema deberá proporcionar un sistema de autenticación para los usuarios, permitiendo el acceso seguro a las funcionalidades de la aplicación.
FRQ-002	Buscar dispositivos	El sistema deberá ser capaz de escanear y listar los dispositivos Bluetooth disponibles para su conexión.
FRQ-003	Vincular dispositivos	El sistema deberá permitir a los usuarios vincular dispositivos seleccionados para establecer una conexión segura y duradera.
FRQ-004	Desvincular dispositivos	El sistema deberá permitir a los usuarios desvincular dispositivos previamente conectados, eliminando la conexión existente.
FRQ-005	Editar propiedades de los dispositivos	El sistema deberá permitir a los usuarios modificar las propiedades configurables de los dispositivos vinculados, como nombres y preferencias.
FRQ-006	Almacenar datos Glucómetro	El sistema deberá ser capaz de recibir y almacenar de manera segura los datos obtenidos desde el glucómetro conectado.
FRQ-007	Visualizar datos Glucómetro	El sistema deberá proporcionar una interfaz que permita a los usuarios visualizar de forma clara y detallada los datos obtenidos desde el glucómetro.

Continúa en la siguiente página

4.2. REQUISITOS NO FUNCIONALES

FRQ-008	Enviar datos manualmente del glucómetro	El sistema deberá permitir a los usuarios enviar manualmente los datos del glucómetro a un servidor o dispositivo externo.
FRQ-009	Almacenar datos Termómetro	El sistema deberá ser capaz de recibir y almacenar de manera segura los datos obtenidos desde el termómetro conectado.
FRQ-010	Visualizar datos Termómetro	El sistema deberá proporcionar una interfaz que permita a los usuarios visualizar de forma clara y detallada los datos obtenidos desde el termómetro.
FRQ-011	Enviar datos manualmente del termómetro	El sistema deberá permitir a los usuarios enviar manualmente los datos del termómetro a un servidor o dispositivo externo.
FRQ-012	Visualizar información de la Tablet	El sistema deberá permitir a los usuarios acceder y visualizar información relevante del estado y las configuraciones de la Tablet utilizada en el proyecto.

Tabla 4.1: Requisitos funcionales.

4.2. Requisitos no funcionales

Los requisitos no funcionales (4.2) establecen las cualidades y restricciones que el sistema debe de cumplir con el fin de garantizar calidad, usabilidad, seguridad, entre otros aspectos. A diferencia de los requisitos funcionales, los requisitos no funcionales se centran en como debe funcionar el sistema.

Código	Requisito	Descripción
NFRQ-01	Compatibilidad de la aplicación	La aplicación tiene que ser compatible con versiones superiores a Android 14.
NFRQ-02	Conectividad MQTT	La aplicación deberá utilizar el protocolo MQTT para conectarse a un broker alojado en AWS, garantizando una comunicación eficiente y segura con los servicios en la nube.
NFRQ-03	Comunicación Bluetooth	La aplicación debe implementar el protocolo Bluetooth utilizando GATT (Generic Attribute Profile) para compartir información con los dispositivos conectados de manera estable y segura.
NFRQ-04	Almacenamiento de datos	La aplicación deberá utilizar una base de datos local en Android para almacenar los datos de los dispositivos, asegurando el acceso rápido y eficiente a la información necesaria para su funcionamiento.
NFRQ-05	Topic MQTT	La aplicación deberá utilizar el topic Bluecare/Bluetablet/id-Tablet/MAC.

Continúa en la siguiente página

Tabla 4.2: Requisitos no funcionales.

4.3. Casos de uso

4.3.1. Definición de actores

El sistema comprende varios actores. En primer lugar, el Administrador (ACT-001) (4.3) será el encargado de vincular los dispositivos necesarios y será el único que podrá visualizar los datos en el sistema. Mientras tanto, el Usuario (ACT-002) (4.4) se encargará de interactuar con los dispositivos vinculados para poder tomar las mediciones. Por último el actor externo Broker MQTT (EXT-001) (4.5), se encargará de recibir los datos enviados por los dispositivos y procesarlos.

ACT-001	Administrador
Autores	David Pastor
Descripción	Este actor representa al administrador de la aplicación es el único que puede acceder a ella

Tabla 4.3: ACT-001 Administrador.

ACT-002	Usuario
Autores	David Pastor
Descripción	Este actor representa al cliente que toma las mediciones de los diferentes dispositivos.

Tabla 4.4: ACT-002 Usuario.

EXT-001	Broker MQTT
Autores	David Pastor
Descripción	Este actor externo representa al Broker MQTT que envía la información.

Tabla 4.5: ACT-003 Broker MQTT.

4.3.2. Diagrama de casos de uso

En esta sección se presenta el diagrama de casos de uso (4.1), elaborado a partir de toda la información adquirida en los apartados anteriores.

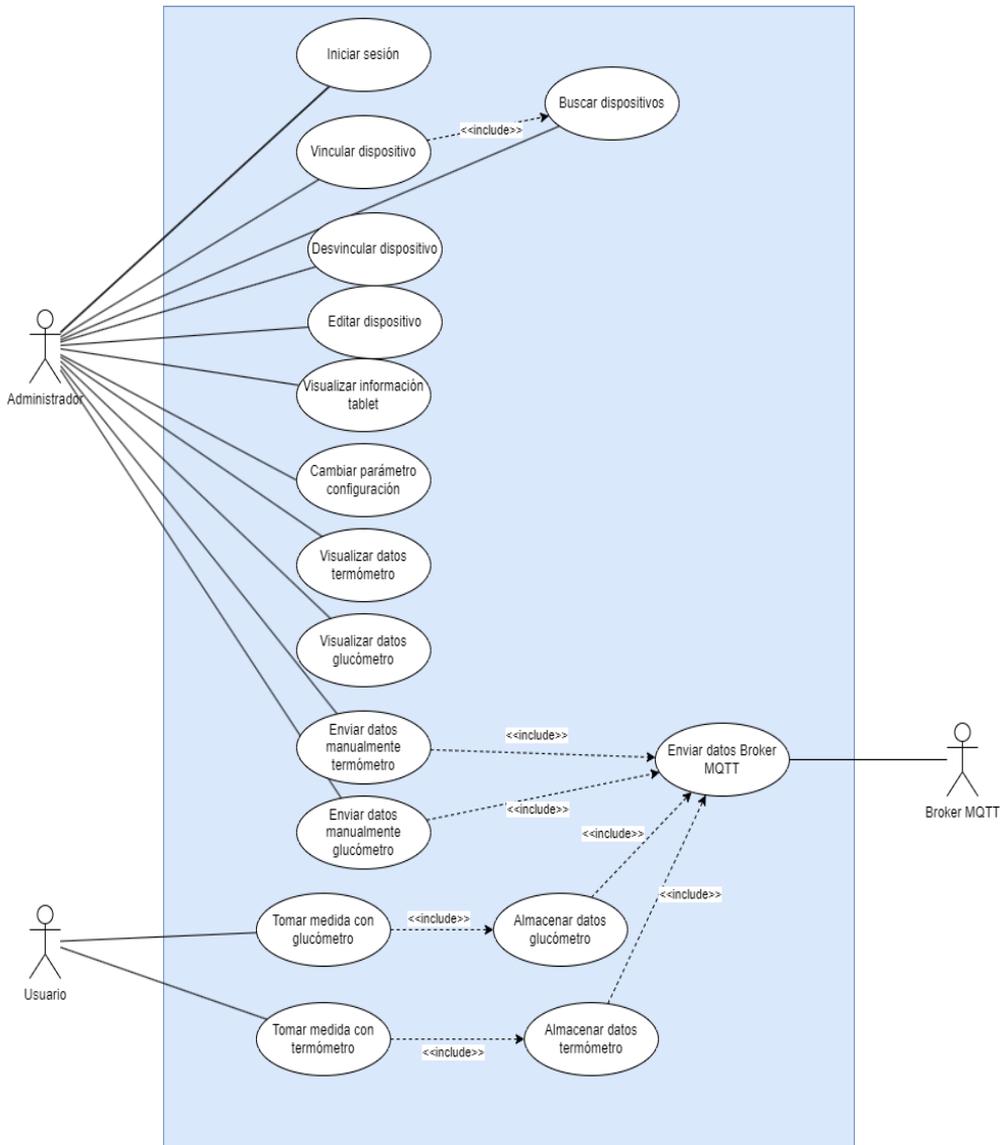


Figura 4.1: Diagrama de casos de uso

4.3.3. Descripción de casos de uso

A continuación se describe cada uno de los casos de uso definidos en el diagrama.

UC-001	Iniciar sesión
Autores	David Pastor Pérez
Fuentes	Luis Platón Arias
Actor principal	Administrador.
Dependencias	Ninguno.
Descripción	El sistema debe permitir al Administrador autenticarse mediante un nombre de usuario y una contraseña.
Precondición	El Administrador debe tener credenciales válidas.
Secuencia normal	<ol style="list-style-type: none">1. El Administrador abre la aplicación.2. El Administrador introduce su nombre de usuario y contraseña.3. El sistema valida las credenciales.4. El sistema proporciona acceso al Administrador si las credenciales son correctas.5. El Administrador accede a la pantalla de inicio.
Postcondición	El Administrador está autenticado y puede acceder al sistema.
Excepción	<ol style="list-style-type: none">2.a. El Administrador introduce credenciales incorrectas.3.a. El sistema muestra un mensaje de error indicando que las credenciales son incorrectas.

Tabla 4.6: UC-001 Iniciar sesión.

UC-002	Vincular dispositivo
Autores	David Pastor Pérez
Fuentes	Luis Platón Arias
Actor principal	Administrador.
Dependencias	Buscar dispositivos.
Descripción	El sistema debe permitir al Administrador vincular un dispositivo a la aplicación después de buscar dispositivos disponibles.
Precondición	El dispositivo debe estar en modo de emparejamiento.
Secuencia normal	1-El Administrador selecciona un dispositivo de la lista de dispositivos disponibles. 2-El sistema envía una solicitud de vinculación al dispositivo. 3-El dispositivo acepta la solicitud y se vincula con la aplicación.
Postcondición	El dispositivo queda vinculado y aparece en la lista de dispositivos vinculados.
Excepción	2.a-El dispositivo no responde a la solicitud. 3.a-El sistema muestra un mensaje de error y permite reintentar.

Tabla 4.7: UC-002 Vincular dispositivo.

UC-003	Desvincular dispositivo
Autores	David Pastor Pérez
Fuentes	Luis Platón Arias
Actor principal	Administrador.
Dependencias	Dispositivo debe estar vinculado.
Descripción	El sistema debe permitir al Administrador desvincular un dispositivo previamente vinculado.
Precondición	El dispositivo debe estar vinculado a la aplicación.
Secuencia normal	1-El Administrador selecciona el dispositivo a desvincular. 2-El sistema envía una solicitud de desvinculación al dispositivo. 3-El dispositivo acepta la solicitud y se desvincula de la aplicación.
Postcondición	El dispositivo se elimina de la lista de dispositivos vinculados.
Excepción	2.a-El dispositivo no responde a la solicitud. 3.a-El sistema muestra un mensaje de error y permite reintentar.

Tabla 4.8: UC-003 Desvincular dispositivo.

UC-004	Editar dispositivo
Autores	David Pastor Pérez
Fuentes	Luis Platón Arias
Actor principal	Administrador.
Dependencias	El dispositivo debe estar vinculado.
Descripción	El sistema debe permitir al Administrador editar las propiedades de un dispositivo vinculado.
Precondición	El dispositivo debe estar vinculado a la aplicación.
Secuencia normal	1-El Administrador selecciona el dispositivo a editar. 2-El sistema muestra la pantalla de edición de propiedades. 3-El Administrador realiza los cambios necesarios y guarda los cambios.
Postcondición	Las propiedades del dispositivo se actualizan en el sistema.
Excepción	2.a-El Administrador introduce datos inválidos. 3.a-El sistema muestra un mensaje de error y permite corregir los datos.

Tabla 4.9: UC-004 Editar dispositivo.

UC-005	Visualizar información Tablet
Autores	David Pastor Pérez
Fuentes	Luis Platón Arias
Actor principal	Administrador.
Dependencias	Ninguno.
Descripción	El sistema debe permitir al Administrador visualizar la información del dispositivo de la Tablet.
Precondición	El Administrador debe estar autenticado en el sistema.
Secuencia normal	1-El Administrador accede a la sección de información de la Tablet. 2-El sistema muestra los datos relevantes de la Tablet en pantalla.
Postcondición	El Administrador visualiza la información actualizada de la Tablet.
Excepción	2.a-El sistema no puede recuperar la información. 3.a-El sistema muestra un mensaje de error.

Tabla 4.10: UC-005 Visualizar información Tablet.

UC-006	Buscar dispositivos
Autores	David Pastor Pérez
Fuentes	Luis Platón Arias
Actor principal	Administrador.
Dependencias	Ninguno.
Descripción	El sistema debe permitir al Administrador buscar dispositivos disponibles para vincular.
Precondición	El Administrador debe estar en la pantalla de búsqueda de dispositivos.
Secuencia normal	1-El Administrador inicia la búsqueda de dispositivos. 2-El sistema muestra una lista de dispositivos disponibles en el área.
Postcondición	El Administrador visualiza una lista de dispositivos disponibles para vincular.
Excepción	2.a-No se encuentran dispositivos disponibles. 3.a-El sistema muestra un mensaje informativo y permite reintentar.

Tabla 4.11: UC-006 Buscar dispositivos.

UC-007	Visualizar datos glucómetro
Autores	David Pastor Pérez
Fuentes	Luis Platón Arias
Actor principal	Administrador.
Dependencias	Almacenar datos Glucómetro.
Descripción	El sistema debe permitir al Administrador y al Usuario visualizar los datos almacenados del glucómetro.
Precondición	Los datos del glucómetro deben estar almacenados en el sistema.
Secuencia normal	1-El Administrador selecciona la opción para visualizar datos del glucómetro. 2-El sistema muestra los datos almacenados en pantalla.
Postcondición	Los datos del glucómetro se muestran correctamente en la interfaz.
Excepción	2.a-No hay datos almacenados para mostrar. 3.a-El sistema muestra un mensaje informativo y permite reintentar.

Tabla 4.12: UC-007 Visualizar datos glucómetro.

UC-008	Enviar datos manualmente glucómetro
Autores	David Pastor Pérez
Fuentes	Luis Platón Arias
Actor principal	Administrador.
Dependencias	Visualizar datos glucómetro, Enviar datos mediante MQTT.
Descripción	El sistema debe permitir al Administrador enviar manualmente los datos del glucómetro a través de MQTT.
Precondición	Los datos del glucómetro deben estar disponibles para el envío.
Secuencia normal	1-El Administrador selecciona la opción para enviar datos manualmente. 2-El sistema muestra los datos disponibles para el envío. 3-El Administrador confirma el envío. 4-El sistema envía los datos al broker MQTT.
Postcondición	Los datos del glucómetro se envían correctamente al broker MQTT.
Excepción	2.a-El Administrador cancela el envío. 3.a-El sistema muestra un mensaje de cancelación.

Tabla 4.13: UC-008 Enviar datos manualmente glucómetro.

UC-009	Enviar datos manualmente termómetro
Autores	David Pastor Pérez
Fuentes	Luis Platón Arias
Actor principal	Administrador.
Dependencias	Visualizar datos termómetro, Enviar datos mediante MQTT.
Descripción	El sistema debe permitir al Administrador enviar manualmente los datos del termómetro a través de MQTT.
Precondición	Los datos del termómetro deben estar disponibles para el envío.
Secuencia normal	1-El Administrador selecciona la opción para enviar datos manualmente. 2-El sistema muestra los datos disponibles para el envío. 3-El Administrador confirma el envío. 4-El sistema envía los datos al broker MQTT.
Postcondición	Los datos del termómetro se envían correctamente al broker MQTT.
Excepción	2.a-El Administrador cancela el envío. 3.a-El sistema muestra un mensaje de cancelación.

Tabla 4.14: UC-009 Enviar datos manualmente termómetro.

4.3. CASOS DE USO

UC-010	Visualizar datos termómetro
Autores	David Pastor Pérez
Fuentes	Luis Platón Arias
Actor principal	Administrador.
Dependencias	Almacenar datos Termómetro.
Descripción	El sistema debe permitir al Administrador visualizar los datos almacenados del termómetro.
Precondición	Los datos del termómetro deben estar almacenados en el sistema.
Secuencia normal	1-El Administrador selecciona la opción para visualizar datos del termómetro. 2-El sistema muestra los datos almacenados en pantalla.
Postcondición	Los datos del termómetro se muestran correctamente en la interfaz.
Excepción	2.a-No hay datos almacenados para mostrar. 3.a-El sistema muestra un mensaje informativo y permite reintentar.

Tabla 4.15: UC-010 Visualizar datos termómetro.

UC-011	Tomar medida con glucómetro
Autores	David Pastor Pérez
Fuentes	Luis Platón Arias
Actor principal	Usuario.
Dependencias	Ninguno.
Descripción	El sistema debe permitir al Usuario tomar una medida con el glucómetro y almacenar los datos.
Precondición	El glucómetro debe estar correctamente vinculado y en funcionamiento.
Secuencia normal	1-El Usuario inicia el proceso de medición con el glucómetro. 2-El glucómetro toma la medida y envía los datos a la aplicación. 3-El sistema almacena los datos de la medida.
Postcondición	Los datos de la medida se almacenan correctamente en el sistema.
Excepción	2.a-El glucómetro no puede tomar la medida. 3.a-El sistema muestra un mensaje de error y permite reintentar.

Tabla 4.16: UC-011 Tomar medida con glucómetro.

UC-012	Tomar medida con termómetro
Autores	David Pastor Pérez
Fuentes	Luis Platón Arias
Actor principal	Usuario.
Dependencias	Ninguno.
Descripción	El sistema debe permitir al Usuario tomar una medida con el termómetro y almacenar los datos.
Precondición	El termómetro debe estar correctamente vinculado y en funcionamiento.
Secuencia normal	1-El Usuario inicia el proceso de medición con el termómetro. 2-El termómetro toma la medida y envía los datos a la aplicación. 3-El sistema almacena los datos de la medida.
Postcondición	Los datos de la medida se almacenan correctamente en el sistema.
Excepción	2.a-El termómetro no puede tomar la medida. 3.a-El sistema muestra un mensaje de error y permite reintentar.

Tabla 4.17: UC-012 Tomar medida con termómetro.

UC-013	Enviar datos Broker MQTT
Autores	David Pastor Pérez
Fuentes	Luis Platón Arias
Actor principal	Administrador, Usuario Broker MQTT.
Dependencias	Ninguno.
Descripción	El sistema debe enviar los datos recopilados (glucómetro y termómetro) al broker MQTT en AWS.
Precondición	El sistema debe tener los datos recopilados listos para enviar.
Secuencia normal	1-El sistema se conecta al broker MQTT en AWS. 2-El sistema envía los datos al broker MQTT. 3-El broker MQTT confirma la recepción de los datos.
Postcondición	Los datos se envían correctamente y se almacenan en AWS.
Excepción	2.a-El sistema no puede conectar con el broker MQTT. 3.a-El sistema muestra un mensaje de error y permite reintentar.

Tabla 4.18: UC-013 Enviar datos Broker MQTT.

4.3. CASOS DE USO

UC-014	Almacenar datos glucómetro
Autores	David Pastor Pérez
Fuentes	Luis Platón Arias
Actor principal	Usuario.
Dependencias	Tomar medida con glucómetro.
Descripción	El sistema debe almacenar los datos tomados del glucómetro en una base de datos local en Android.
Precondición	Los datos del glucómetro deben estar disponibles para almacenar.
Secuencia normal	1-El Usuario toma una medida con el glucómetro. 2-El sistema almacena los datos de la medida en la base de datos local.
Postcondición	Los datos del glucómetro se almacenan correctamente en la base de datos local.
Excepción	2.a-El sistema no puede almacenar los datos. 3.a-El sistema muestra un mensaje de error y permite reintentar.

Tabla 4.19: UC-014 Almacenar datos glucómetro.

UC-015	Almacenar datos termómetro
Autores	David Pastor Pérez
Fuentes	Luis Platón Arias
Actor principal	Usuario.
Dependencias	Tomar medida con termómetro.
Descripción	El sistema debe almacenar los datos tomados del termómetro en una base de datos local en Android.
Precondición	Los datos del termómetro deben estar disponibles para almacenar.
Secuencia normal	1-El Usuario toma una medida con el termómetro. 2-El sistema almacena los datos de la medida en la base de datos local.
Postcondición	Los datos del termómetro se almacenan correctamente en la base de datos local.
Excepción	2.a-El sistema no puede almacenar los datos. 3.a-El sistema muestra un mensaje de error y permite reintentar.

Tabla 4.20: UC-015 Almacenar datos termómetro.

UC-016	Cambiar parámetros de configuración
Autores	David Pastor Pérez
Fuentes	Luis Platón Arias
Actor principal	Administrador.
Dependencias	Ninguna.
Descripción	El sistema permite al Administrador modificar los parámetros de configuración, como el intervalo de envío de mensajes, que se almacenan de manera global en el sistema Android.
Precondición	El Administrador debe tener acceso a la pantalla de configuración.
Secuencia normal	1-El Administrador accede a la pantalla de información del dispositivo. 2-El Administrador introduce el nuevo intervalo de tiempo (en horas). 3-El sistema almacena el nuevo valor en las preferencias globales de Android. 4-El servicio de fondo es reiniciado para aplicar el nuevo valor de configuración.
Postcondición	El nuevo intervalo de tiempo queda correctamente almacenado y aplicado en el sistema.
Excepción	3.a-El sistema no puede almacenar el nuevo valor. 4.a-El sistema muestra un mensaje de error y permite reintentar.

Tabla 4.21: UC-016 Cambiar parámetros de configuración.

4.4. Modelo de dominio

A continuación se muestra el modelo de dominio del sistema (4.2), el cual define las principales entidades y relaciones del sistema. Este modelo se ha diseñado con una clara orientación hacia la ampliabilidad, permitiendo futuras incorporaciones de dispositivos médicos y funcionalidades sin necesidad de rediseñar el sistema. Principalmente en las clases de SubscriptionDevice y CommandResponseDevice, que permite diferenciar entre los dos tipos de obtención de información de los dispositivos.

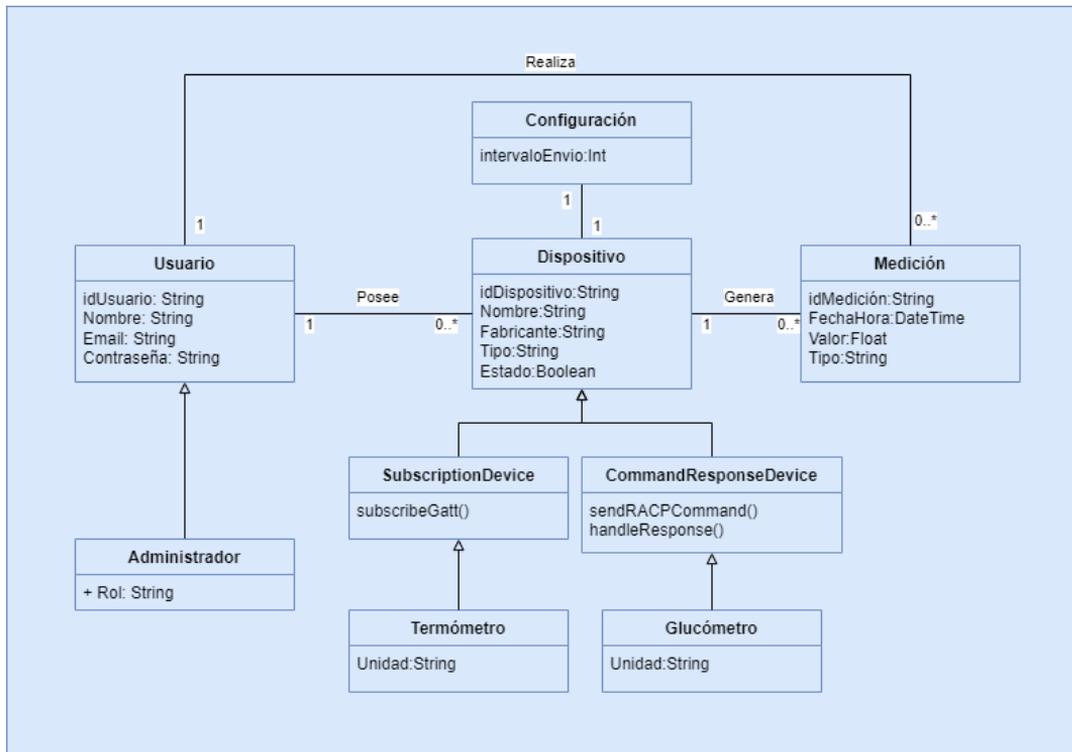


Figura 4.2: Modelo de dominio

4.4.1. Descripción de las clases.

Usuario: Clase que representa a los usuarios que utilizarán la aplicación.

- idUsuario: String - Identificador único del usuario.
- Nombre: String - Nombre del usuario.
- Email: String - Correo electrónico del usuario.

- Contraseña: String - Contraseña para autenticación.

Administrador: Subclase de **Usuario** que añade el rol de administración para gestionar el sistema.

- Rol: String - Rol específico.

Dispositivo: Clase base que representa un dispositivo conectado al sistema.

- idDispositivo: String - Identificador único del dispositivo.
- Nombre: String - Nombre del dispositivo.
- Fabricante: String - Fabricante del dispositivo.
- Tipo: String - Tipo de dispositivo (Glucómetro, Termómetro).
- Estado: Boolean - Estado de conexión (Vinculado/Desvinculado).

SubscriptionDevice: Clase intermedia que extiende **Dispositivo** y representa dispositivos que utilizan únicamente la suscripción GATT para la recepción continua de datos. Esta clase proporciona un método para gestionar las suscripciones sin necesidad de enviar comandos adicionales.

- **suscribirGatt()**: Método que inicia la suscripción al servicio GATT para recibir datos en tiempo real.

CommandResponseDevice: Clase intermedia que extiende **Dispositivo** y representa dispositivos que requieren la interacción mediante comandos (por ejemplo, RACP) para solicitar datos y luego recibir las respuestas a través de una suscripción.

- **enviarComandoRACP()**: Método para enviar un comando RACP al dispositivo.
- **suscribirRespuesta()**: Método que gestiona la suscripción para recibir las respuestas del dispositivo tras enviar el comando.

Glucómetro: Subclase de **CommandResponseDevice** que representa un glucómetro que envía comandos y luego espera respuestas para obtener los datos de medición.

- Unidad: String - Unidad de medida de la glucosa (mg/dL).

Termómetro: Subclase de **SubscriptionDevice** que representa un termómetro que utiliza la suscripción GATT para obtener datos de temperatura.

- Unidad: String - Unidad de medida de la temperatura (C, F).
- ComandoRACP: Boolean - Indica si el dispositivo utiliza el protocolo RACP para enviar comandos y recibir respuestas.

Medición: Clase que representa una medición realizada por cualquier dispositivo.

- idMedicion: Integer - Identificador único de la medición.
- FechaHora: DateTime - Fecha y hora de la medición.
- Valor: Float - Valor de la medición.
- Tipo: String - Tipo de medición (Glucosa, Temperatura).

Configuración: Clase que representa la configuración del dispositivo.

- intervaloEnvio: Integer - Valor en horas que indica la frecuencia de envío automático de datos.

4.5. Diagrama de despliegue

El diagrama de despliegue (4.3) proporciona una representación visual de la arquitectura física del sistema, detallando como los componentes se distribuyen entre los diferentes dispositivos.

1. **Dispositivos médicos:** tenemos dos dispositivos que son los encargados de capturar glucosa y temperatura. Ambos representados como bloques "device" cada uno de estos dispositivos se conectan con la Tablet Android mediante una conexión Bluetooth.
2. **Tablet Android:** se representa como un "device" que actúa como núcleo del sistema, dentro del dispositivo se despliega la aplicación, que es la encargada de gestionar la interacción con los dispositivos médicos y procesar la información, tiene una relación interna con una base de datos local.
3. **AWS:** la aplicación móvil se conecta a un servicio proporcionado por AWS, representado como un "device" dentro de este servicio se encuentra un Broker MQTT, que es el encargado de gestionar los datos que le llegan.

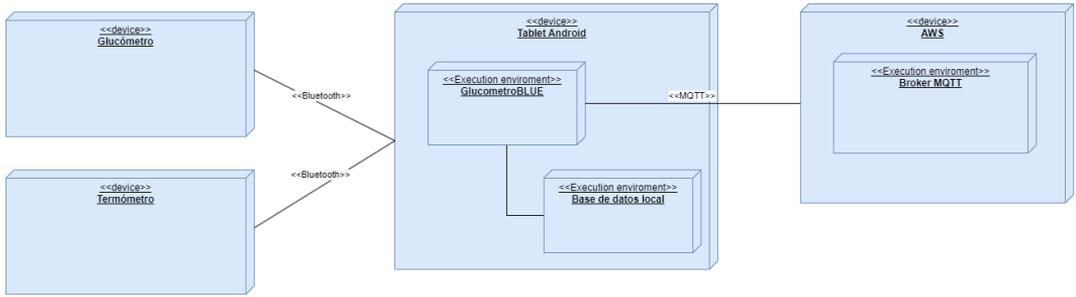


Figura 4.3: Diagrama de despliegue

4.6. Diseño base de datos

En esta sección se presentará el diseño de la base de datos local (4.4) utilizada en la aplicación, esta base de datos es fundamental para poder gestionar la información de los dispositivos conectados, así como de las diferentes mediciones obtenidas. Se describirán las tablas principales detallando la función de cada una junto con los campos que contienen.

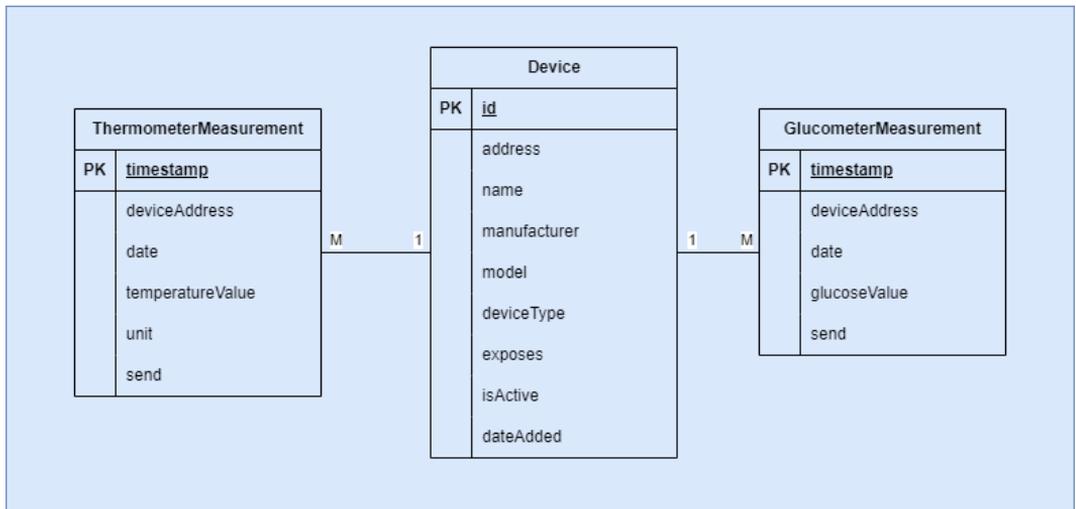


Figura 4.4: Diseño base de datos

Devices: esta tabla almacena la información de los dispositivos conectados y su tipo, esta formada por los siguientes campos:

- **ID:** este campo es la clave primaria y almacena un identificado único para cada dispositivo.

- **Address:** este campo contiene la dirección MAC del dispositivo Bluetooth. Esto permite identificar el dispositivo, este campo no es clave primaria ya que un dispositivo puede vincularse y desvincularse varias veces.
- **Name:** nombre del dispositivo, suele ser un nombre amigable al usuario para poder reconocerlo.
- **Manufacturer:** nombre del fabricante.
- **Model:** nombre del modelo específico.
- **DeviceType:** almacena el tipo de dispositivo, como puede ser glucómetro o termómetro.
- **Exposes:** se almacena un JSON que describe las funcionalidades o propiedades que el dispositivo expone.
- **IsActive:** valor booleano que indica si el dispositivo está actualmente activo.
- **DateAdded:** campo que almacena un valor representando la fecha en la que el dispositivo fue añadido.

GlucometerMeasurements: esta tabla almacena las diferentes medidas de glucosa obtenidas de un glucómetro.

- **Timestamp:** clave primaria de la tabla y representa el momento exacto en formato Unix en el que se realizó una medición.
- **DeviceAddress:** este campo contiene la dirección MAC del dispositivo Bluetooth que tomó la medida.
- **Date:** Almacena fecha y hora de la medición en un formato legible, esto es útil para mostrar al usuario directamente sin necesidad de convertir el timestamp.
- **GlucoseValue:** contiene el valor de la glucosa medido por el dispositivo con las unidades.
- **Send:** valor booleano que indica si la medición fue enviada al Broker MQTT. Esto ayuda a evitar una pérdida de datos o un reenvío de datos duplicado.

ThermometerMeasurements: esta tabla almacena las diferentes medidas de glucosa obtenidas de un termómetro.

- **Timestamp:** clave primaria de la tabla y representa el momento exacto en formato Unix en el que se realizó una medición.
- **DeviceAddress:** este campo contiene la dirección MAC del dispositivo Bluetooth que tomó la medida.

- **Date:** Almacena fecha y hora de la medición en un formato legible, esto es útil para mostrar al usuario directamente sin necesidad de convertir el timestamp.
- **TemperatureValue:** contiene el valor de la temperatura medido por el dispositivo.
- **Unit:** especifica la unidad de la temperatura medida, como "C" para Celsius o "F".
- **Send:** valor booleano que indica si la medición fue enviada al Broker MQTT. Esto ayuda a evitar una pérdida de datos o un reenvío de datos duplicado.

4.7. Diseño para la integración de otros dispositivos médicos

Uno de los objetivos es poder ofrecer una solución que no solo permita monitorizar la glucosa y temperatura corporal, sino que pueda ser una plataforma escalable para poder integrar diferentes tipos de dispositivos médicos. La arquitectura de la aplicación ha sido diseñada con una alta flexibilidad, facilitando la integración de nuevos dispositivos como pueden ser pulsioxímetros, monitores de presión arterial, medidores de frecuencia cardíaca, etc.

El primer aspecto clave es la utilización de Bluetooth como protocolo de comunicación para la extracción de los datos, esto permite un diseño fácilmente ampliable a dispositivos que utilicen este mismo protocolo, lo cual es común en los dispositivos actuales. La mayoría utilizan protocolos compatibles con BLE y la aplicación está diseñada para manejar este tipo de dispositivos, lo que permitirá a futuro añadir nuevos dispositivos sin cambiar significativamente la arquitectura.

Además se diseñaron módulos independientes que realizan la conexión, lectura, envío y transformación de los datos, para cada uno de los dispositivos, pudiendo añadir nuevos módulos a futuro sin necesidad de modificar los módulos existentes.

Otro aspecto clave es el uso del protocolo MQTT para la transmisión de datos, el servidor permite una escalabilidad alta en la recepción de datos ya que es un protocolo liviano y escalable, lo que permite poder recibir datos de muchos dispositivos diferentes.

Las base de datos local que permite tener un historial y no perder datos en caso de posibles desconexiones con internet, es fácilmente ampliable creando nuevas tablas para almacenar datos del nuevo dispositivo y poder mostrarlo en la aplicación.

La seguridad y privacidad de los datos han sido prioritarios en el diseño del sistema, empleando sistemas que cifran los datos tanto en la transmisión de datos a través de Bluetooth, como en la comunicación mediante el protocolo de datos, asegurando así la escalabilidad del sistema no comprometa la confidencialidad de los datos.

La integración de parámetros configurables, como el intervalo de envío de datos, permite ajustar el funcionamiento del sistema de acuerdo a las necesidades individuales, además esto permite a futuro añadir nuevos parámetros personalizados para los nuevos dispositivos.

4.8. Prototipo inicial

Antes de iniciar el desarrollo, se diseñaron bocetos de las diferentes pantallas de la aplicación, con el fin de obtener una idea básica del resultado final.



Figura 4.5: Prototipo Login



Figura 4.6: Prototipo gestor Bluetooth

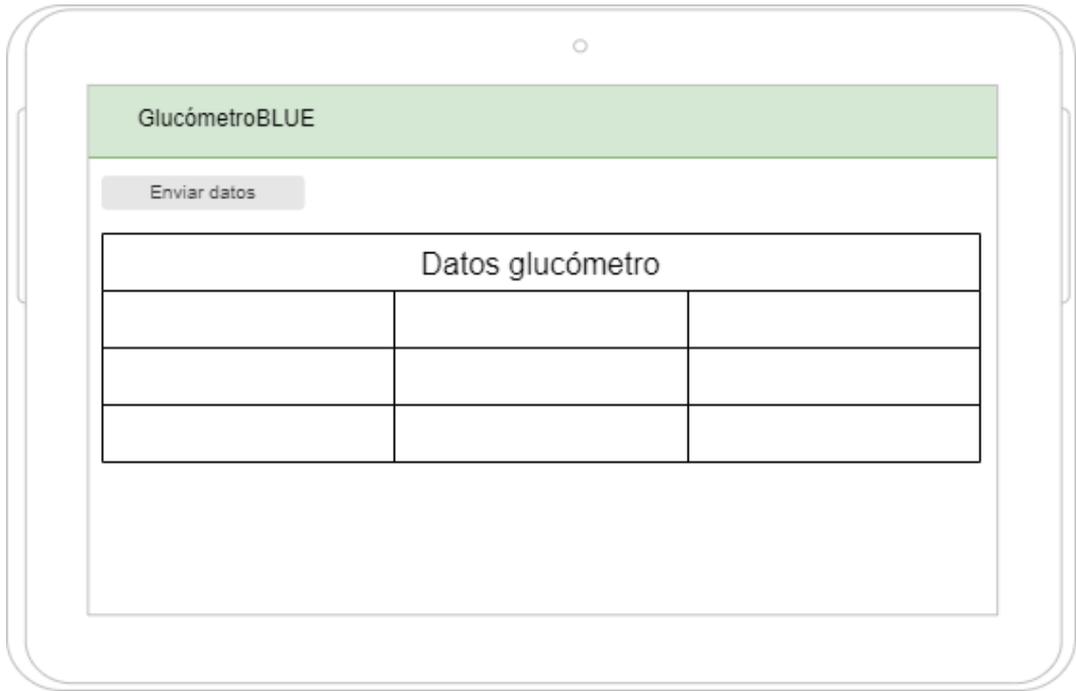


Figura 4.7: Prototipo glucómetro tabla

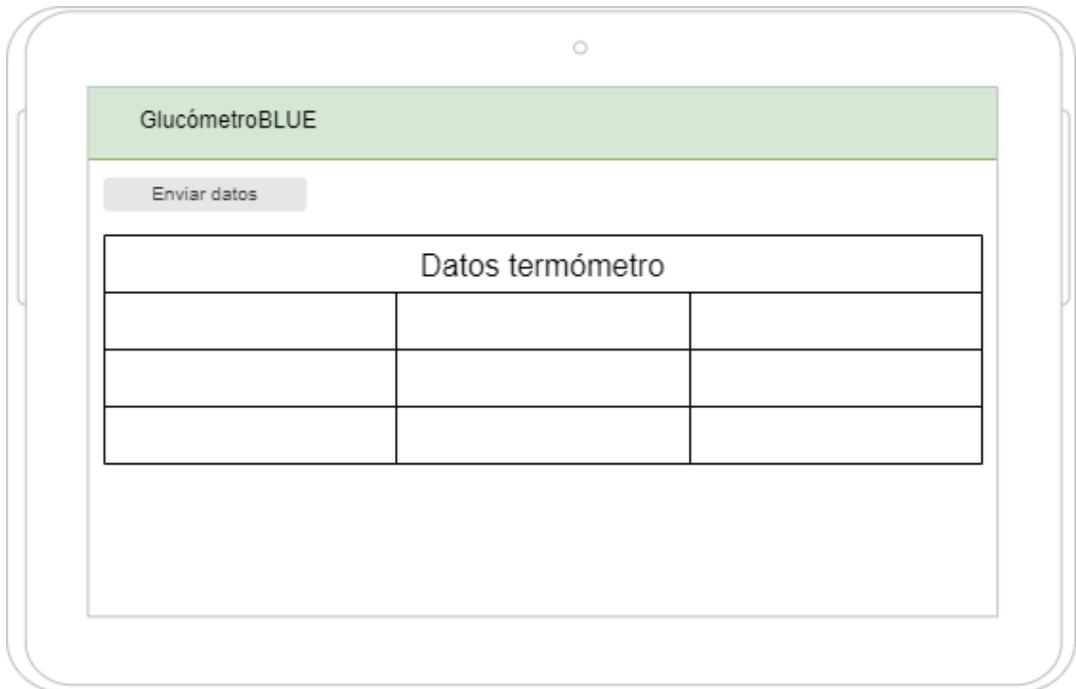


Figura 4.8: Prototipo termómetro tabla

Capítulo 5

Tecnologías y servicios utilizados

En este capítulo se van a exponer las distintas tecnologías y servicios utilizados durante el desarrollo de este trabajo de fin de grado.

5.1. Android Studio



Figura 5.1: Logotipo Android Studio

Android studio [4] es el entorno de desarrollo oficial de Google. Fue publicado en diciembre de 2014 sustituyendo al actual entorno de desarrollo para aplicaciones móviles el cual era Eclipse. Android studio esta publicado bajo la licencia de apache 2.0. Está disponible para los sistemas operativos Windows, Mac OS y GNU/Linux.

El lenguaje preferido por Google para el desarrollo de aplicaciones Android es Kotlin pero Android Studio permite el uso de otros lenguajes como Java o C++.

Características principales:

- Soporte para la construcción basada en Gradle.
- Refactorización específica de Android.

- Plantillas para crear diseños comunes de Android.
- Soporte para programar aplicaciones en Android Wear.
- Soporte integrado para Google Cloud Plataform.
- Dispositivo virtual de Android que permite ejecutar y probar aplicaciones.
- Consola de desarrollador.

5.2. GitHub



Figura 5.2: Logotipo GitHub

GitHub [15] es un portal creado para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador. El software que opera GitHub fue escrito en Ruby on Rails. Desde enero de 2010, GitHub opera bajo el nombre de GitHub, Inc. Anteriormente era conocida como Logical Awesome LLC.

5.3. Bluetooth



Figura 5.3: Logotipo Bluetooth

Bluetooth [17] es una tecnología que permite la conexión inalámbrica entre PCs, dispositivos móviles y periféricos como auriculares, teclados, ratones, etc. Utilizando transmisiones de radio de corto alcance en la frecuencia de 2,4 GHz. Los dispositivos se emparejan en un proceso en el que se intercambia información de seguridad y conexión, lo que facilita futuras conexiones automáticas.

Existen dos versiones **Bluetooth Classic** y **Bluetooth Low Energy**. ‘‘Classic’’ se utiliza para aplicaciones que requieren una transferencia de datos constante y dispone de 79 canales. BLE, por otra parte esta optimizado para conexiones esporádicas y de bajo consumo energético, con 40 canales disponibles. Ambos usan una técnica de salto de frecuencia para mejorar la calidad de la señal y reducir las interferencias.

5.4. MQTT



Figura 5.4: Logotipo MQTT

MQTT [5] es un protocolo de mensajería basado en estándares, o un conjunto de reglas, que se utiliza para la comunicación de un equipo a otro. Los sensores inteligentes, los dispositivos portátiles y otros dispositivos de Internet de las cosas (IoT) generalmente tienen que transmitir y recibir datos a través de una red con recursos restringidos y un ancho de banda limitado. Estos dispositivos IoT utilizan MQTT para la transmisión de datos, ya que resulta fácil de implementar y puede comunicar datos IoT de manera eficiente. MQTT admite la mensajería entre dispositivos a la nube y la nube al dispositivo.

5.5. MQTT Explorer

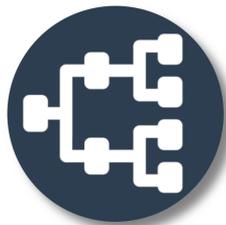


Figura 5.5: Logotipo MQTT Explorer

MQTT Explorer [21] es un cliente MQTT completo que ofrece una visión estructurada de los ‘‘Topics’’ de MQTT, facilitando la gestión de dispositivos y servicios del broker, incluye características como, la visualización de ‘‘Topics’’, eliminación, búsqueda, eliminación recursiva, visualización de mensajes, publicación, etc.

Capítulo 6

Implementación

En este capítulo se presentan las diferentes etapas de la implementación, comenzando con la interfaz gráfica. Posteriormente, se abordan los aspectos fundamentales y la explicación sobre como se realizan las diferentes conexiones.

6.1. Interfaz gráfica

En este apartado se describe el desarrollo de la interfaz gráfica basado en el boceto que se realizó en la fase de diseño.

A continuación se presentan las diferentes pantallas que muestran el resultado final de la interfaz gráfica de la aplicación.

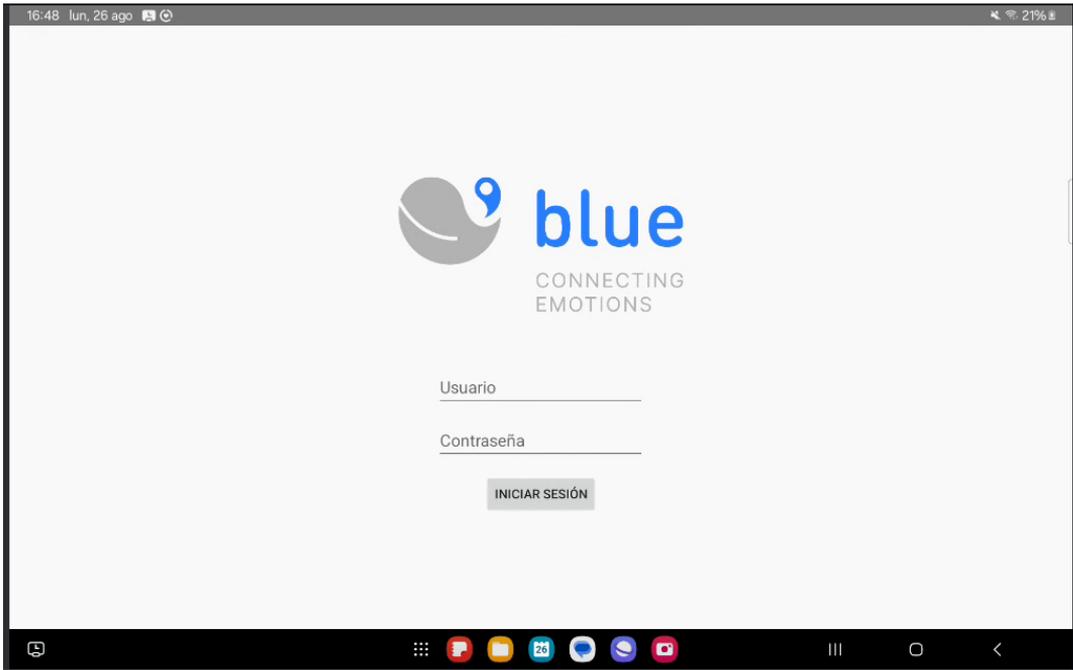


Figura 6.1: Login

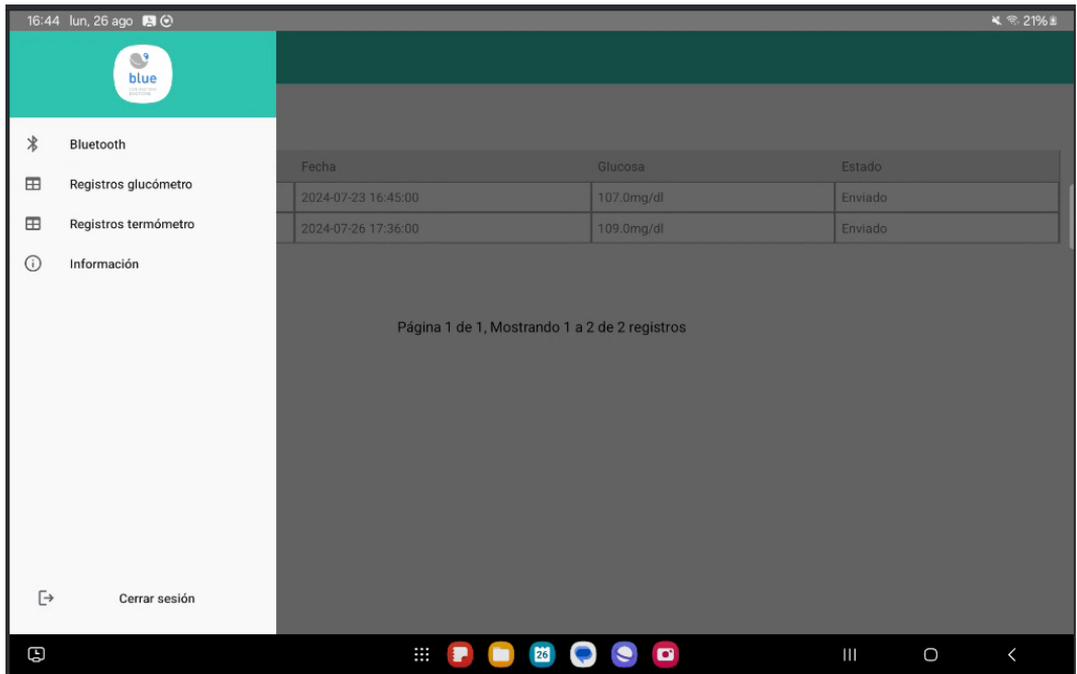


Figura 6.2: Menú lateral

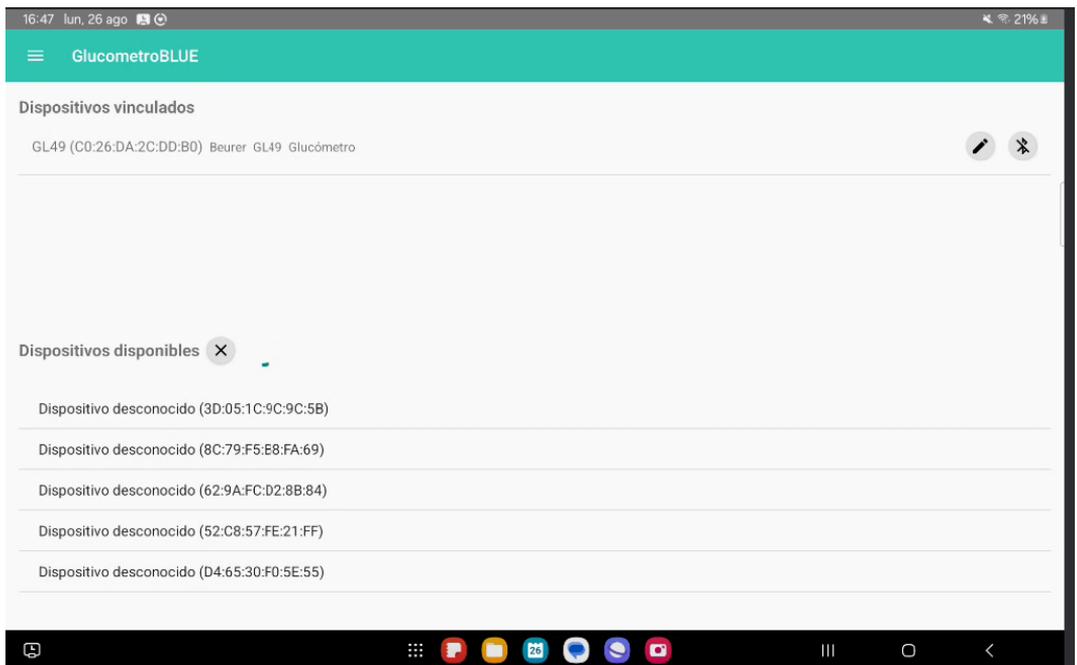
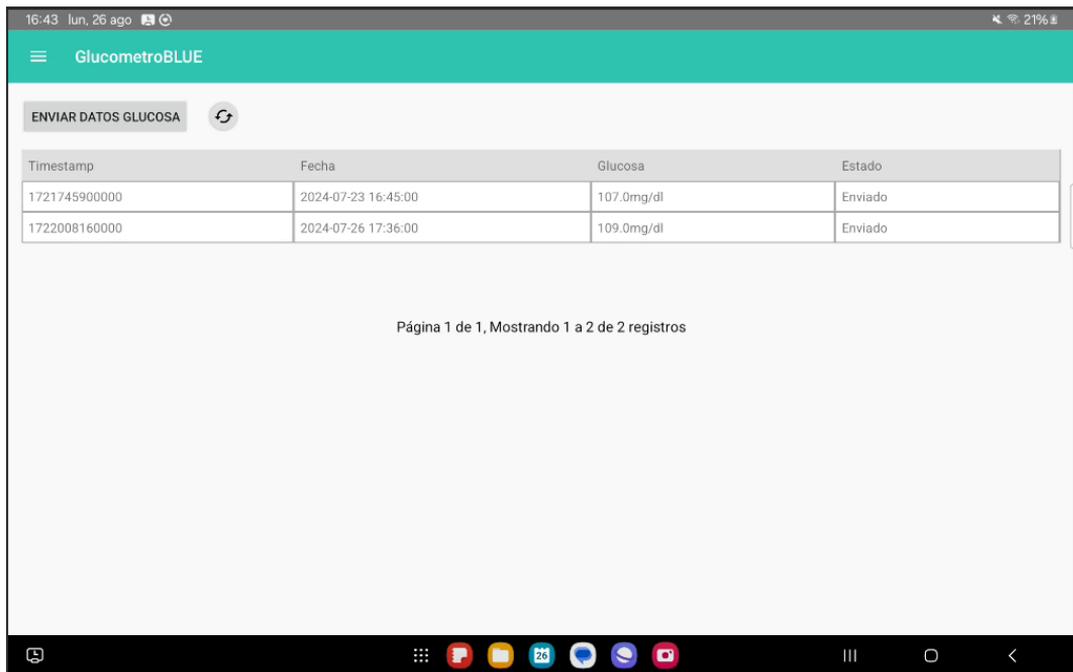


Figura 6.3: Gestor de dispositivos Bluetooth

6.1. INTERFAZ GRÁFICA

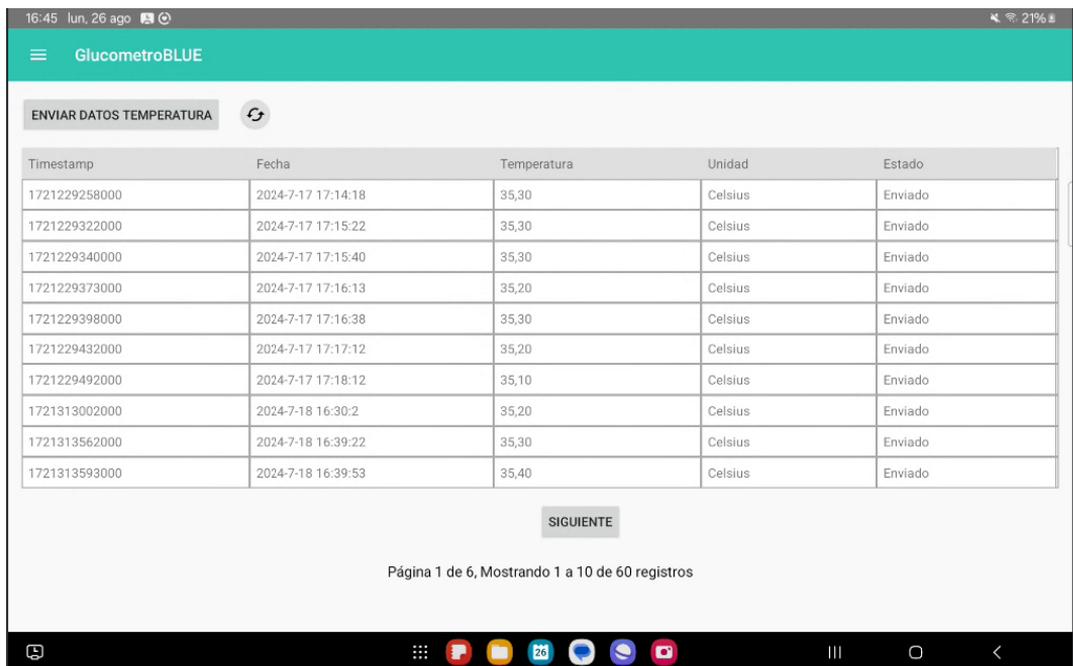


The screenshot shows the 'GlucometroBLUE' app interface. At the top, there is a teal header with a menu icon and the text 'GlucometroBLUE'. Below the header, there is a button labeled 'ENVIAR DATOS GLUCOSA' with a refresh icon. The main content area contains a table with the following data:

Timestamp	Fecha	Glucosa	Estado
1721745900000	2024-07-23 16:45:00	107.0mg/dl	Enviado
1722008160000	2024-07-26 17:36:00	109.0mg/dl	Enviado

Below the table, the text 'Página 1 de 1, Mostrando 1 a 2 de 2 registros' is displayed. The bottom of the screen shows an Android-style navigation bar with various icons.

Figura 6.4: Tabla con los datos obtenidos del glucómetro



The screenshot shows the 'GlucometroBLUE' app interface. At the top, there is a teal header with a menu icon and the text 'GlucometroBLUE'. Below the header, there is a button labeled 'ENVIAR DATOS TEMPERATURA' with a refresh icon. The main content area contains a table with the following data:

Timestamp	Fecha	Temperatura	Unidad	Estado
1721229258000	2024-7-17 17:14:18	35,30	Celsius	Enviado
1721229322000	2024-7-17 17:15:22	35,30	Celsius	Enviado
1721229340000	2024-7-17 17:15:40	35,30	Celsius	Enviado
1721229373000	2024-7-17 17:16:13	35,20	Celsius	Enviado
1721229398000	2024-7-17 17:16:38	35,30	Celsius	Enviado
1721229432000	2024-7-17 17:17:12	35,20	Celsius	Enviado
1721229492000	2024-7-17 17:18:12	35,10	Celsius	Enviado
1721313002000	2024-7-18 16:30:2	35,20	Celsius	Enviado
1721313562000	2024-7-18 16:39:22	35,30	Celsius	Enviado
1721313593000	2024-7-18 16:39:53	35,40	Celsius	Enviado

Below the table, there is a button labeled 'SIGUIENTE'. The text 'Página 1 de 6, Mostrando 1 a 10 de 60 registros' is displayed. The bottom of the screen shows an Android-style navigation bar with various icons.

Figura 6.5: Tabla con los datos obtenidos del termómetro

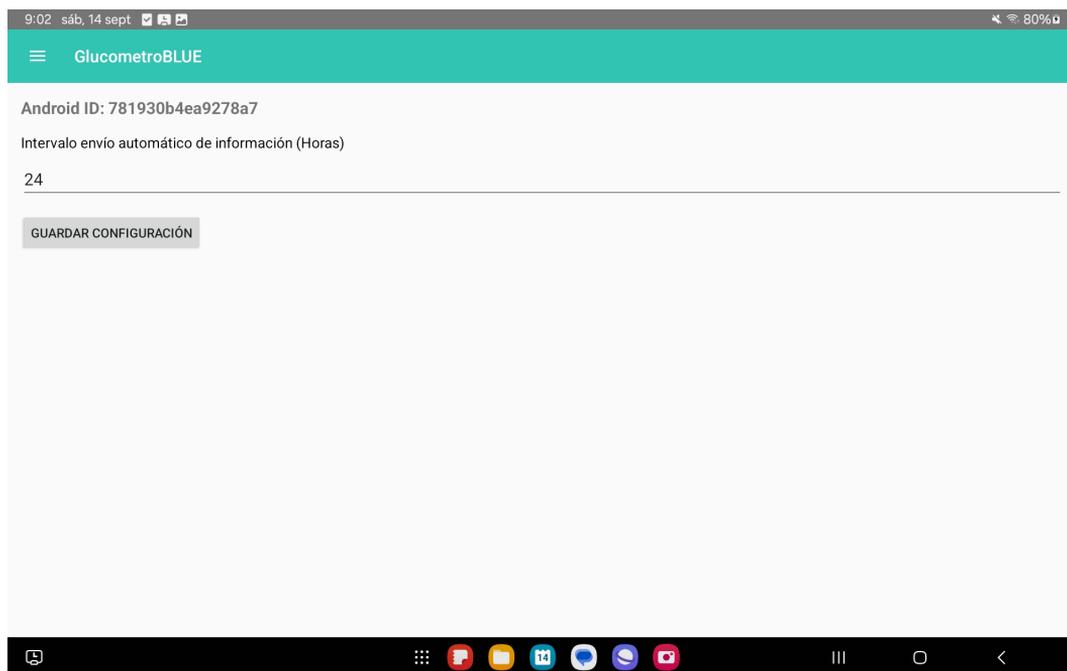


Figura 6.6: Información del dispositivo

6.2. Arquitectura y Gestión de Datos

En esta sección se muestra la arquitectura general del sistema y cómo se gestiona la conexión y el intercambio de datos con los diferentes dispositivos, su procesamiento y el envío final al Broker MQTT que se encuentra en AWS

6.2.1. Conexión con los dispositivos médicos

Para la conexión con los dispositivos se utiliza la tecnología Bluetooth, utilizando en concreto la API BLE (Bluetooth Low Energy) de Android para manejar la conectividad. Esta conectividad se realiza utilizando el perfil GATT (Generic Attribute Profile). GATT define un conjunto de reglas para enviar y recibir paquetes de datos conocidos como atributos. Los atributos son organizados en servicios y características que permiten a los dispositivos comunicarse de una manera estructurada. Las características son una entidad que contiene un valor, y es a través de esta, que los datos, como las mediciones de temperatura o glucosa, son transferidos.

La aplicación escanea los dispositivos disponibles y, una vez detectados, permite vincular el dispositivo deseado. La vinculación se gestiona mediante identificadores únicos de cada dispositivo, lo que asegura una conexión persistente y específica para cada dispositivo. Para manejar esta comunicación se implementó un servicio de Android en segundo plano, que se

encarga de mantener la conexión activa y gestionar las posibles interrupciones o reconexiones. Esto asegura que la conexión sea persistente.

6.2.2. Conexión y obtención de datos termómetro

Para el termómetro, la aplicación se conecta al dispositivo mediante BLE y descubre los servicios disponibles. El servicio que nos interesa sería 'Health Thermometer'. Este servicio contiene dos características pero la que nos interesa obtener es la de 'Temperature Measurement', la aplicación se suscribe a esta característica, permitiendo que las mediciones de temperatura sean enviadas automáticamente desde el termómetro cada vez que se toma una nueva lectura. Estos valores son recibidos a tiempo real y se procesan dentro de la aplicación, convirtiéndolos en un formato legible para posteriormente almacenarlas en la base de datos local, para después enviarlas mediante MQTT al Broker. El envío de datos se valida en la base de datos local para asegurar que solo se envía una única vez y evitar la duplicidad de datos.

A continuación se muestra un ejemplo del servicio (6.1) junto con sus dos características. El UUID es el identificador único de la característica, y con este valor se suscribe la aplicación al dispositivo para obtener las mediciones

Nombre	UUID	Propiedades
Temperature Measurement	0x2A1C	Indicate
Measurement Interval	0x2A21	Read

Tabla 6.1: Health Thermometer Service Characteristics.

A continuación se muestra un ejemplo de los datos recibidos cuando existe una nueva medición.

[0x07 0x10 0x03 0x00 0xFF 0xE1 0x07 0x04 0x0D 0x0E 0x07 0x00 0xFF]

- **0x07** – Los flags configurados.
- **0x10 0x03 0x00 0xFF** – Valor de temperatura de 78.4.
- **0xE1 0x07** – Año 2017.
- **0x04** – Mes Abril.
- **0x0D** – Día 13.
- **0x0E** – Hora 14.
- **0x07** – Minuto 07.
- **0x00** – Segundo 0.
- **0xFF** – Tipo de medición.

6.2.3. Conexión y obtención de datos glucómetro

El proceso para el glucómetro es similar en cuanto al uso de GATT, pero con diferencias en el manejo de datos. El glucómetro también expone un servicio, pero las mediciones de glucosa están codificadas en un formato específico conocido como SFLOAT, que requiere un procesamiento adicional. Este formato incluye una mantisa y un exponente que deben ser correctamente interpretados para convertir los datos en un valor de glucosa legible.

Al igual que el termómetro, la aplicación se suscribe a la característica 'Glucose Measurement' que se encuentra dentro del servicio 'Glucose', pero a diferencia del termómetro existe un paso previo para que el valor nos llegue. Esto se obtiene mediante la característica que maneja el acceso y control de los registros, conocida como 'Record Access Control Point (RACP)'. Para que el glucómetro comience a enviar las mediciones almacenadas, es necesario que la aplicación envíe un comando específico a esta característica. Este comando es una instrucción que le indica al dispositivo que inicie la transmisión de datos. Cuando esto ocurre los datos llegan al igual que el termómetro en la suscripción hablada anteriormente. Una vez decodificados, los valores de glucosa se almacenan en la base de datos local junto con la marca de tiempo y se envían al Broker mediante MQTT.

A continuación se muestra un el servicio de glucosa (6.2) junto con sus características.

Name	Assigned Number	Properties	Note
Glucose Measurement	0x2A18	Notify	Measurements are received here
Glucose Measurement Context	0x2A34	Notify	Used to transmit the measurement marker
Glucose Feature	0x2A51	Read	Low Battery Detection, Sensor Malfunction Detection, Sensor Sample Size, Sensor Strip Insertion Error Detection, Sensor Strip Type Error Detection, Sensor Result High-Low Detection, Sensor Temperature High-Low Detection, Sensor Read Interrupt Detection, General Device Fault, Time Fault, Multiple Bond
Record Access Control Point	0x2A52	Indicate, Write	Used to ask for the glucose measurement data

Tabla 6.2: Glucose service characteristics.

A continuación se muestra un ejemplo de los datos recibidos y su interpretación.

[0x12 0x07 0x00 0xE4 0x07 0x04 0x0F 0x03 0x23 0x00 0x64 0xB0 0x11]

- **0x12** – La concentración de glucosa y los campos Tipo-Ubicación de la muestra están presentes, la unidad es mg/dL y el registro incluye información de contexto.
- **0x00 0x07** – Número de secuencia 7.
- **0x07 0xE4 0x04 0x0F 0x03 0x23 0x00** – Fecha y hora: 15.04.2020 03:35:00.
- **0xB0 0x64** – 0.001 kg/l.
- **0x11** – Tipo: Sangre capilar completa y Ubicación: Dedo.

Para el cálculo de la glucosa [22] se tiene que tener en cuenta que esta en formato SFLOAT, y a continuación se va a explicar como se obtiene el valor de aquí. En primer lugar es necesario leer los flags que nos determina la unidad de la concentración de glucosa (mg/dl o mol/L). Sabiendo esto tenemos que tener en cuenta que SFLOAT ocupa 2 bytes, y se compone de dos partes, el Mantisa (12 bits), los primeros 12 bits representan la parte significativa del número y los últimos 4 bits representan el exponente en base 10, que puede ser positivo o negativo. La concentración de glucosa se obtendría con la siguiente fórmula:

$$\text{Glucosa} = \text{Mantisa} \times 10^{\text{Exponente}} \quad (6.1)$$

6.2.4. Conexión con Broker MQTT en AWS

Para establecer la conexión con AWS desde la aplicación Android, se sigue un proceso que busca asegurar la autenticidad y la seguridad de la conexión. Esto se logra mediante el uso de certificados y claves que se almacenan de forma segura en el dispositivo. Para la conexión se utiliza el patrón Singleton que gestiona una única conexión y se encarga de que se mantenga activa mientras la aplicación esta en uso. La aplicación puede enviar datos al Broker MQTT publicando mensajes en tópicos específicos. Estos mensajes pueden contener cualquier tipo de información que la aplicación necesite transmitir.

A continuación se muestra un ejemplo de mensaje para el termómetro:

```
{
  "timestamp": 1725375091000,
  "date": "2024-9-3 16:51:31",
  "temperatureValue": "34,50",
  "unit": "Celsius"
}
```

Capítulo 7

Pruebas

En este capítulo se documentan las pruebas realizadas sobre los distintos casos de uso implementados en el sistema, con el objetivo de validar el correcto funcionamiento y cumplimiento de todas las funcionalidades del sistema. Estas pruebas se basan en asegurar el correcto funcionamiento entre las diferentes interacciones entre el usuario y los dispositivos comprobando que sean eficientes, precisas y seguras.

Durante el desarrollo del sistema además se realizaron diferentes pruebas unitarias de cada parte realizada para ir validando las funcionalidades.

Para cada caso de uso, se detallan los siguientes aspectos:

- **Descripción:** Breve explicación de la funcionalidad evaluada.
- **Resultado esperado:** Comportamiento previsto del sistema.
- **Privacidad y seguridad:** Consideraciones sobre la protección de datos y la seguridad de la comunicación.
- **Resultado de la prueba:** Validación de los resultados obtenidos tras la ejecución de la prueba.

CPSYS-001	Iniciar sesión
Descripción	La prueba consiste en que un usuario (Administrador) inicie sesión en la aplicación, verificando la autenticación segura de las credenciales.
Resultado esperado	La aplicación valida correctamente las credenciales y muestra la pantalla principal.
Resultado de la prueba	Correcto.

Tabla 7.1: Batería de pruebas para el caso CPSYS-001 Iniciar sesión.

CPSYS-002	Buscar dispositivos
Descripción	El Administrador busca dispositivos disponibles para vincular.
Resultado esperado	Se muestra una lista de dispositivos disponibles.
Privacidad y seguridad	Los dispositivos encontrados no deben exponer información sensible sin la debida autenticación.
Resultado de la prueba	Correcto.

Tabla 7.2: Batería de pruebas para el caso CPSYS-002 Buscar dispositivos.

CPSYS-003	Vincular dispositivo
Descripción	El Administrador debe vincular un dispositivo a la aplicación.
Resultado esperado	El dispositivo se vincula correctamente y aparece en la lista de dispositivos vinculados.
Privacidad y seguridad	El proceso de vinculación debe asegurarse mediante claves cifradas y autenticación mutua.
Resultado de la prueba	Correcto.

Tabla 7.3: Batería de pruebas para el caso CPSYS-003 Vincular dispositivo.

CPSYS-004	Desvincular dispositivo
Descripción	El Administrador debe desvincular un dispositivo previamente vinculado.
Resultado esperado	El dispositivo se elimina de la lista de dispositivos vinculados.
Privacidad y seguridad	Deben eliminarse las claves y permisos de acceso vinculados al dispositivo para evitar accesos no autorizados futuros.
Resultado de la prueba	Correcto.

Tabla 7.4: Batería de pruebas para el caso CPSYS-004 Desvincular dispositivo.

CPSYS-005	Editar dispositivo
Descripción	El Administrador debe editar las propiedades de un dispositivo vinculado.
Resultado esperado	Los cambios en las propiedades del dispositivo se guardan correctamente.
Privacidad y seguridad	Los cambios deben verificarse para evitar la modificación de parámetros críticos sin la debida autenticación.
Resultado de la prueba	Correcto.

Tabla 7.5: Batería de pruebas para el caso CPSYS-005 Editar dispositivo.

CPSYS-006	Visualizar información de la Tablet
Descripción	El Administrador debe poder visualizar la información del dispositivo Tablet.
Resultado esperado	La información se visualiza correctamente.
Privacidad y seguridad	Los datos sensibles de la Tablet no deben mostrarse sin la debida autenticación.
Resultado de la prueba	Correcto.

Tabla 7.6: Batería de pruebas para el caso CPSYS-006 Visualizar información Tablet.

CPSYS-007	Tomar medida glucómetro
Descripción	El Usuario toma una medida con el glucómetro, y los datos son enviados automáticamente al broker MQTT sin intervención adicional.
Resultado esperado	La medida se toma correctamente y los datos se envían automáticamente al broker MQTT, sin errores.
Privacidad y seguridad	Los datos deben cifrarse antes de enviarse y la conexión al broker MQTT debe estar autenticada para evitar interceptaciones.
Resultado de la prueba	Correcto.

Tabla 7.7: Batería de pruebas para el caso CPSYS-007 Tomar medida glucómetro .

CPSYS-008	Tomar medida termómetro
Descripción	El Usuario toma una medida con el termómetro, y los datos son enviados automáticamente al broker MQTT sin intervención adicional.
Resultado esperado	La medida se toma correctamente y los datos se envían automáticamente al broker MQTT, sin errores.
Privacidad y seguridad	Los datos deben cifrarse antes de enviarse y la conexión al broker MQTT debe estar autenticada para garantizar la seguridad de los datos.
Resultado de la prueba	Correcto.

Tabla 7.8: Batería de pruebas para el caso CPSYS-008 Tomar medida termómetro.

CPSYS-009	Visualizar datos glucómetro
Descripción	El Administrador visualiza los datos almacenados del glucómetro.
Resultado esperado	Los datos se visualizan correctamente en la interfaz.
Privacidad y seguridad	Los datos deben estar protegidos en tránsito, garantizando la privacidad del usuario.
Resultado de la prueba	Correcto.

Tabla 7.9: Batería de pruebas para el caso CPSYS-009 Visualizar datos glucómetro.

CPSYS-010	Visualizar datos termómetro
Descripción	El Administrador visualiza los datos del termómetro almacenados en el sistema.
Resultado esperado	Los datos del termómetro se muestran correctamente en la interfaz.
Privacidad y seguridad	Los datos deben estar protegidos con cifrado en tránsito, respetando la privacidad del usuario.
Resultado de la prueba	Correcto.

Tabla 7.10: Batería de pruebas para el caso CPSYS-010 Visualizar datos termómetro.

CPSYS-011	Enviar datos manualmente glucómetro
Descripción	El Administrador envía manualmente los datos del glucómetro al broker MQTT.
Resultado esperado	Los datos se envían correctamente al broker.
Privacidad y seguridad	Los datos deben cifrarse antes de ser enviados y se debe garantizar la autenticidad del destinatario (broker MQTT).
Resultado de la prueba	Correcto.

Tabla 7.11: Batería de pruebas para el caso CPSYS-011 Enviar datos manualmente glucómetro.

CPSYS-012	Enviar datos manualmente termómetro
Descripción	El Administrador envía manualmente los datos del termómetro al broker MQTT.
Resultado esperado	Los datos se envían correctamente al broker.
Privacidad y seguridad	Los datos deben cifrarse antes de ser enviados y se debe verificar la autenticidad del destinatario (broker MQTT).
Resultado de la prueba	Correcto.

Tabla 7.12: Batería de pruebas para el caso CPSYS-012 Enviar datos manualmente termómetro.

Capítulo 8

Conclusiones y trabajo futuro

En este proyecto se ha logrado desarrollar una aplicación Android funcional que permite la monitorización de datos de dispositivos médicos, específicamente glucómetros y termómetros, mediante Bluetooth. La aplicación ha cumplido con los objetivos establecidos inicialmente, facilitando la extracción y gestión de los datos provenientes de estos dispositivos. Estos datos se envían a servidores en la nube utilizando el protocolo MQTT, permitiendo una monitorización continua y remota que aporta valor a los pacientes. Además se implementaron las medidas de privacidad y seguridad para proteger los datos sensibles y se validó mediante pruebas la funcionalidad de la aplicación, para asegurar un funcionamiento correcto.

Este proyecto se alinea con la tendencia actual en el cuidado de personas vulnerables, proporcionando una solución asequible y fácil de usar que apoya la monitorización en el hogar. Esto permite a las personas mayores y a aquellos con enfermedades crónicas evitar o retrasar la institucionalización.

Desde el inicio, el proyecto ha estado orientado hacia la creación de un producto comercial viable, con la intención de ofrecer soluciones innovadoras en el ámbito de la salud. Se ha diseñado pensando en los consumidores y un mercado real, con una arquitectura flexible que permite la integración de nuevos dispositivos médicos en el futuro. Esto permite que la aplicación no solo sea específica para la glucosa y temperatura, sino una plataforma escalable para la inclusión de más dispositivos.

Trabajo Futuro

A partir de la versión actual, se identifican como trabajo futuro los siguientes aspectos a mejorar:

- **Mejora de la interfaz gráfica:** uno de los aspectos a mejorar sería una interfaz gráfica que pueda proporcionar una experiencia al usuario más intuitiva y atractiva. Esto puede involucrar el rediseño de la interfaz.

-
- **Añadir nuevos dispositivos:** la ampliación del sistema añadiendo nuevos dispositivos médicos como pulsioxímetros, tensiómetros, etc. es una parte fundamental. Esto requerirá la creación de nuevas clases y la integración en el sistema.
 - **Pantalla de visualización de datos única:** la implementación de una pantalla centralizada para visualizar todos los datos, en la que se puedan seleccionar diferentes tipos de medidas y dispositivos en el mismo lugar.
 - **Nuevos parámetros de configuración:** Añadir nuevos parámetros de configuración en la correspondiente ventana para hacer al sistema más personalizable y que permita configurar todo en función del usuario objetivo.

Estas mejoras no solo optimizarán la aplicación existente, sino que además fortalecerá su posición en el mercado como una solución innovadora y adaptable. La visión a largo plazo es consolidar el producto como una herramienta comercial viable.

Bibliografía

- [1] AHEAD. *AHEAD*. en. <https://www.aal-europe.eu/projects/ahead/>. Accessed: 2024-9-13. Dic. de 2012.
- [2] AIB. *AIB - AAL programme*. en. <https://www.aal-europe.eu/projects/aib/>. Accessed: 2024-9-13. Jul. de 2012.
- [3] AMICA. *AMICA*. en. <https://www.aal-europe.eu/projects/amica/>. Accessed: 2024-9-13. Jun. de 2012.
- [4] android. *Download Android Studio and SDK tools — Android Studio*. en. URL: <https://developer.android.com/studio?hl=es> (visitado 24-08-2024).
- [5] AWS. *¿Qué es MQTT?* es. <https://aws.amazon.com/es/what-is/mqtt/>. Accessed: 2024-8-24. URL: <https://aws.amazon.com/es/what-is/mqtt/>.
- [6] Comitas. *Telemedicina*. es. <https://comitas.es/>. Accessed: 2024-8-31. Dic. de 2023.
- [7] Conforme-Mendoza-Enma-Monserrate. *PLATAFORMA DE ALARMAS PARA PERSONAS DE LA TERCERA EDAD CON DIABETES*. <https://repositorio.uisrael.edu.ec/bitstream/47000/2606/1/UISRAEL-EC-ELTD-378.242-035.pdf>. [Accessed 22-07-2024]. 2020.
- [8] Kelly Cullinane. *ISO and IEC approve OASIS AMQP Advanced Message Queuing Protocol*. en. <https://www.oasis-open.org/news/pr/iso-and-iec-approve-oasis-amqp-advanced-message-queuing-protocol/>. Accessed: 2024-9-9. Mayo de 2014.
- [9] Dds-foundation. <https://www.dds-foundation.org/what-is-dds-3/>. Accessed: 2024-9-9.
- [10] Gdpr. <https://gdpr-text.com/es/read/article-9/>. Accessed: 2024-9-18.
- [11] Gdpr. <https://gdpr-text.com/es/read/article-12/>. Accessed: 2024-9-18.
- [12] Gdpr. <https://gdpr-text.com/es/read/article-5/>. Accessed: 2024-9-18.
- [13] Gdpr. <https://gdpr-text.com/es/read/article-32/>. Accessed: 2024-9-18.
- [14] Gdpr. <https://gdpr-text.com/es/read/article-30/>. Accessed: 2024-9-18.
- [15] github. *Build software better, together*. en. URL: <https://github.com> (visitado 24-08-2024).
- [16] Huawei. *Protocolo CoAP*. en. <https://forum.huawei.com/enterprise/es/protocolo-coap/thread/667241330188894208-667212896255488000>. Accessed: 2024-9-9.

- [17] intel. *Build software better, together*. es. Accessed: 2024-8-24. URL: <https://www.intel.la/content/www/xl/es/products/docs/wireless/how-does-bluetooth-work.html#:~:text=La%20tecnolog%C3%ADa%20Bluetooth%20conecta%20las,frecuencia%20de%202%2C4%20GHz..>
- [18] MESI. *Bienvenida a MESI*. es. <https://www.mesimedical.com/es>. Accessed: 2024-8-31.
- [19] Ministerio de Derechos Sociales y Agenda 2030. *Estrategia estatal para un nuevo modelo de cuidados en la comunidad (2024-2030)*. 2024. URL: <https://www.mdsocialesa2030.gob.es/derechos-sociales/docs/Estrategia-para-nuevo-modelo-cuidados-en-la-comunidad.pdf>.
- [20] Fintan Murphy. *Glucohealth*. <https://norma.ncirl.ie/2673/1/fintanmurphy.pdf>. [Accessed 22-07-2024]. 2017.
- [21] Thomas Nordquist. *MQTT explorer*. en. <https://mqtt-explorer.com/>. Accessed: 2024-8-24.
- [22] White Paper. *Personal Health Devices Transcoding*. https://www.bluetooth.com/wp-content/uploads/2019/03/PHD_Transcoding_WP_v16.pdf. Accessed: 2024-9-3.
- [23] SmartCare. *SmartCare Solutions*. es. <https://www.essencesmartcare.com/es/solutions/smartcare-solutions/>. Accessed: 2024-8-31. Abr. de 2024.

Apéndice A

Manual de instalación

Este manual explica los pasos necesarios para instalar la APK del proyecto compilada con Android Studio en un dispositivo Android. Dado que la APK actualmente no se distribuye a través de Google Play, es necesario habilitar la instalación desde fuentes no seguras.

Requisitos previos:

- Dispositivo Android versión 14 o superior.
- La APK compilada del proyecto.
- Cable USB para transferir la APK al dispositivo.

Para transferir la APK al dispositivo Android, conecta el dispositivo al ordenador que contiene el archivo APK y transfíerelo a la carpeta de *Descargas* en el dispositivo.

Una vez con la APK se encuentra en el dispositivo, es necesario habilitar la instalación de aplicaciones desde fuentes desconocidas, ya que la APK no proviene de Google Play.

A continuación, se muestra un ejemplo de cómo habilitar esta opción en una Tablet Android con versión 14.

1. Abre la app *Ajustes*.
2. Utiliza el buscador e introduce “Instalar” (A.1).
3. Selecciona la opción *Instalar apps desconocidas*.
4. Pulsa nuevamente en *Instalar apps desconocidas*(A.2).
5. Habilita el permiso para la aplicación *Mis archivos*(A.3).

Ahora ya es posible instalar la APK. Para ello, abre la aplicación *Mis archivos* y en la carpeta de *Descargas* en la que copiamos el instalador tenemos que pulsar sobre la APK, y aceptar la instalación (A.4).

Al iniciar la aplicación por primera vez, es necesario aceptar todos los permisos solicitados, de lo contrario la aplicación no funcionara correctamente.

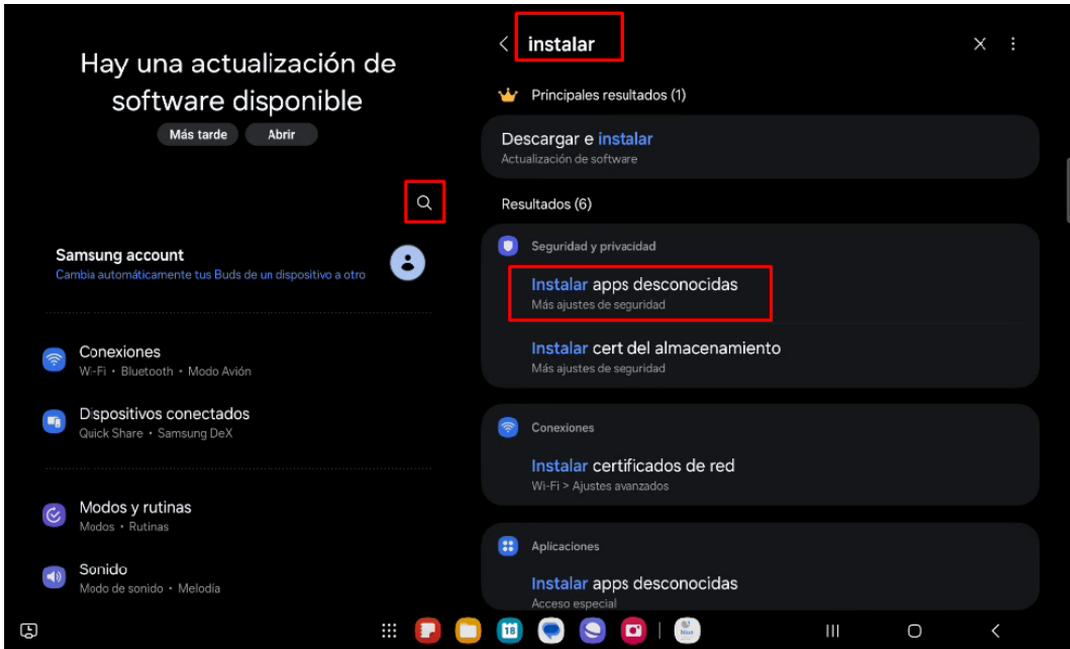


Figura A.1: Ejemplo activar orígenes desconocidos Android 14 (1)

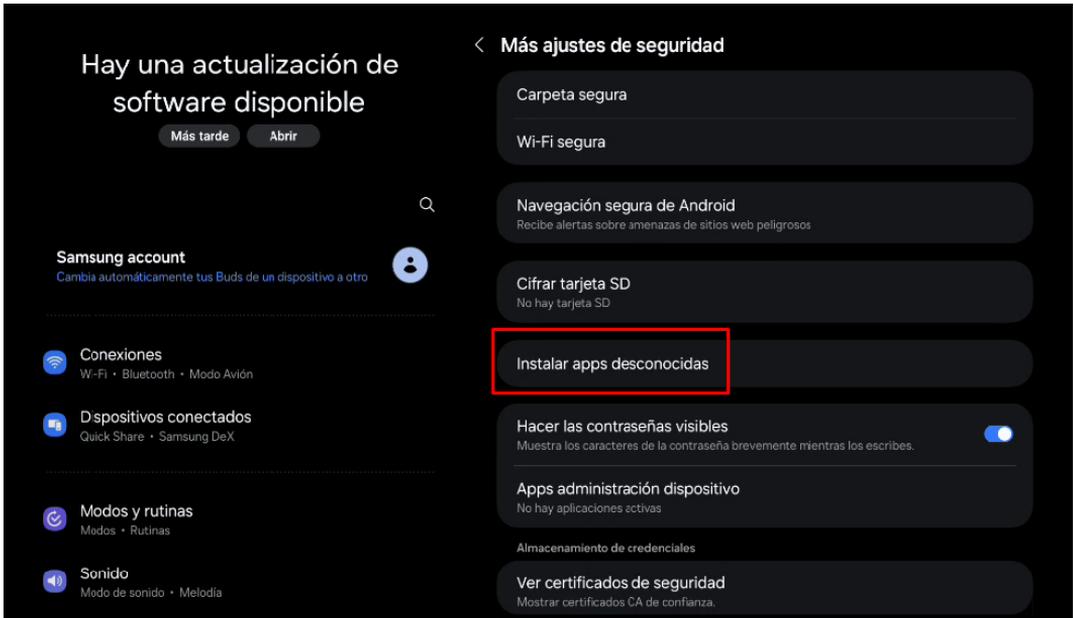


Figura A.2: Ejemplo activar orígenes desconocidos Android 14 (2)

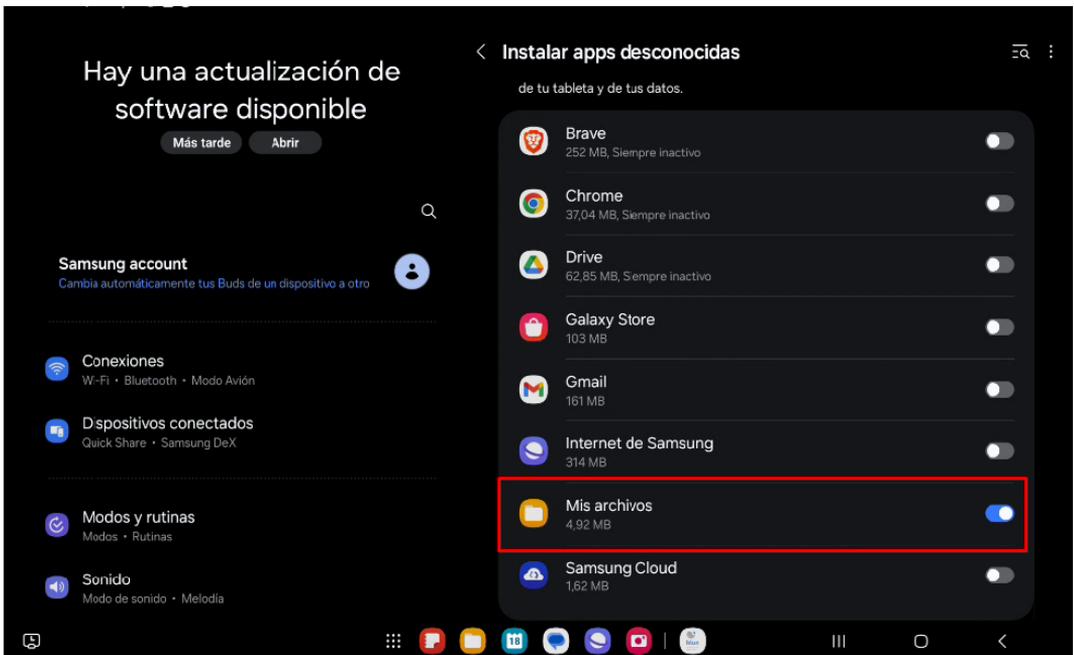


Figura A.3: Ejemplo activar orígenes desconocidos Android 14 (3)

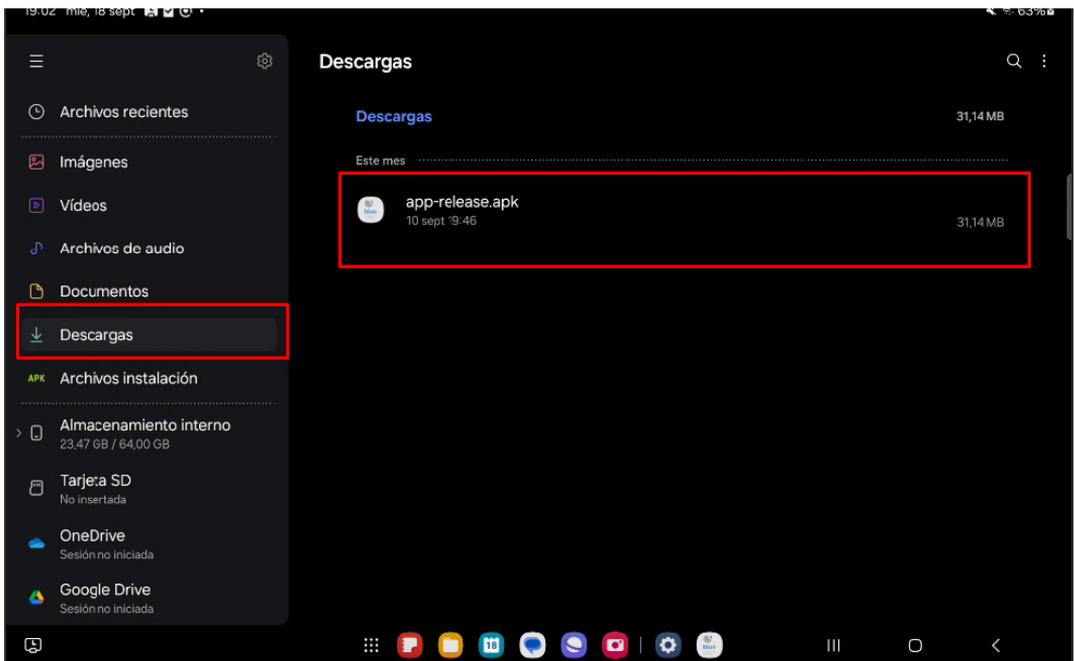


Figura A.4: Ejemplo activar orígenes desconocidos Android 14 (4)

Apéndice B

Manual de usuario

El objetivo de este manual es mostrar, mediante una serie de imágenes y comentarios las diferentes funcionalidades de la aplicación

B.1. Iniciar sesión

Para iniciar sesión se tienen que introducir los datos correctos de Usuario y Contraseña y posteriormente pulsar el botón de iniciar sesión (B.1), si los datos son correctos se abrirá la pantalla principal de la aplicación. 5

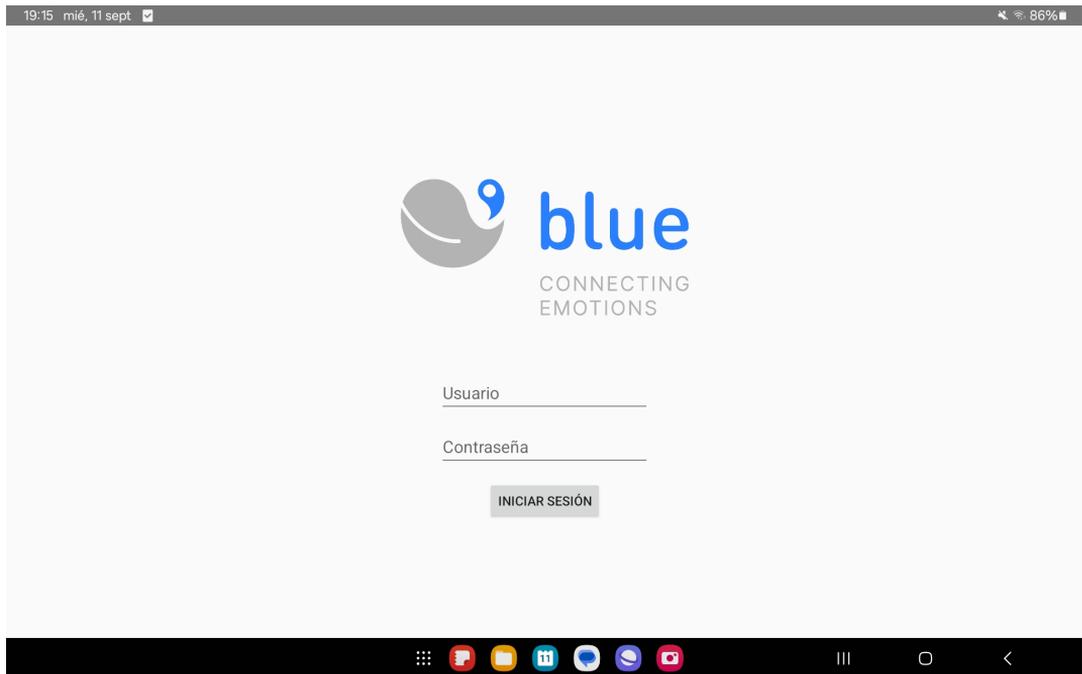


Figura B.1: Iniciar sesión

B.2. Vincular nuevo dispositivo

Para poder vincular un nuevo dispositivo, es necesario ir a la pantalla del **gestor de dispositivos**. Una vez dentro, se deben buscar los dispositivos disponibles. Para esto, es necesario pulsar el botón indicado en la imagen (B.2). Cuando se localice el dispositivo que se desea añadir, se debe de pulsar sobre el (B.3). A continuación, se mostrara una ventana para introducir el código de vinculación (B.4). Si el código introducido es correcto, el dispositivo se mostrara en la sección de dispositivos vinculados junto con sus datos (B.5).

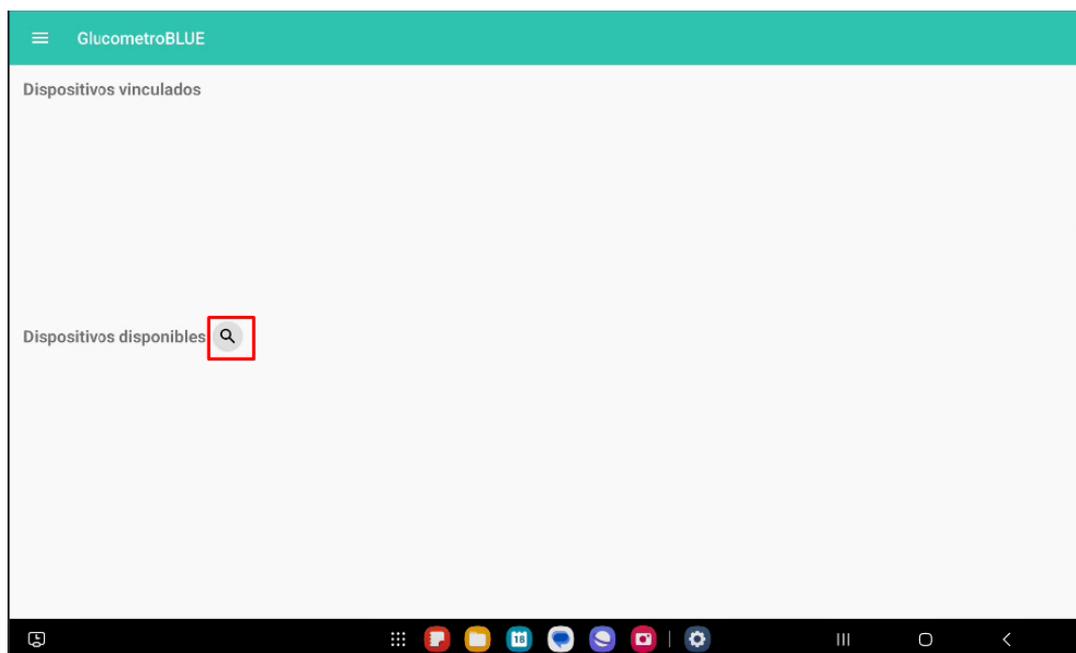


Figura B.2: Buscar dispositivos

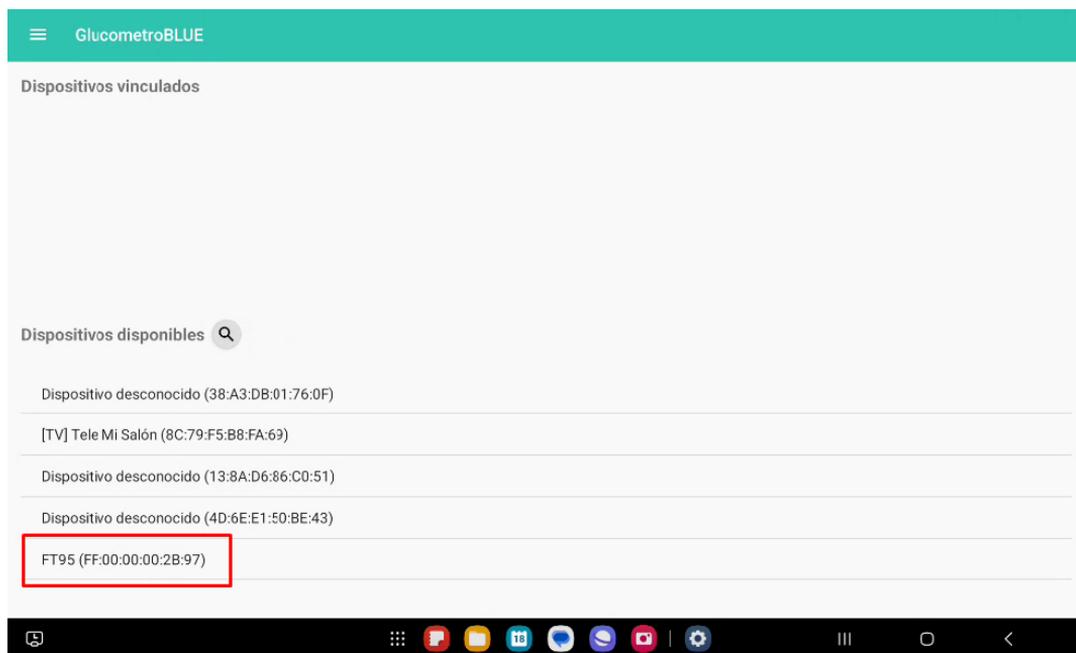


Figura B.3: Vincular dispositivos 1

B.2. VINCULAR NUEVO DISPOSITIVO

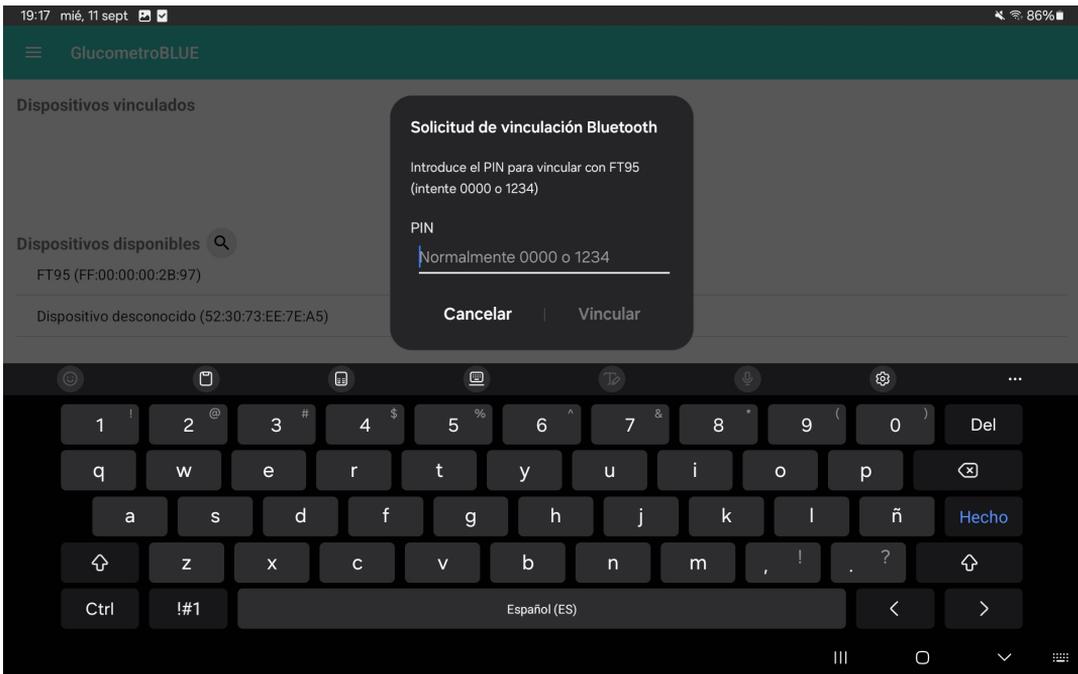


Figura B.4: Vincular dispositivos 2

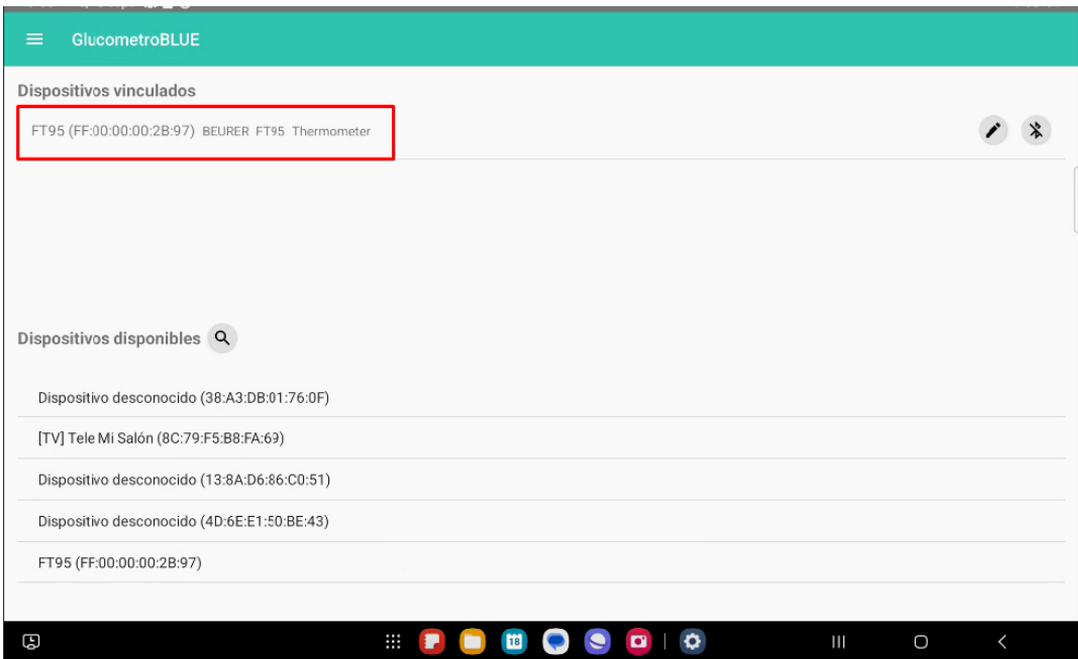


Figura B.5: Vincular dispositivos 3

B.3. Editar dispositivo

Si se desea editar alguno de los datos de un dispositivo, se debe pulsar el botón con el icono de un lápiz. A continuación, se abrirá una ventana en la que se podrán modificar el **nombre**, el **fabricante** y el **tipo de dispositivo**. Estos valores se asignan automáticamente en la vinculación, pero si en algún momento es necesario editarlos, se puede hacer desde este menú (B.6).

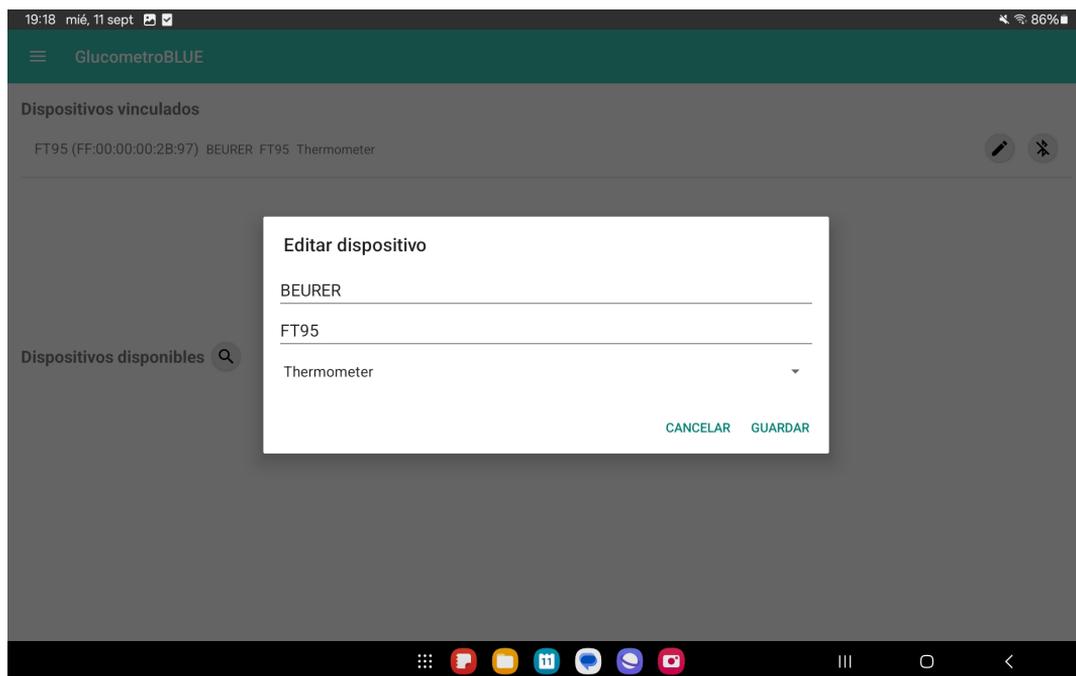


Figura B.6: Editar dispositivos

B.4. Desvincular dispositivo

Para desvincular un dispositivo, es necesario que haya al menos uno en el listado de dispositivos vinculados. Se debe pulsar el botón con el icono de Bluetooth tachado (B.7), tras el cual se mostrara un menú de confirmación (B.8). Si se pulsa en **Aceptar**, el dispositivo se desvinculara correctamente.

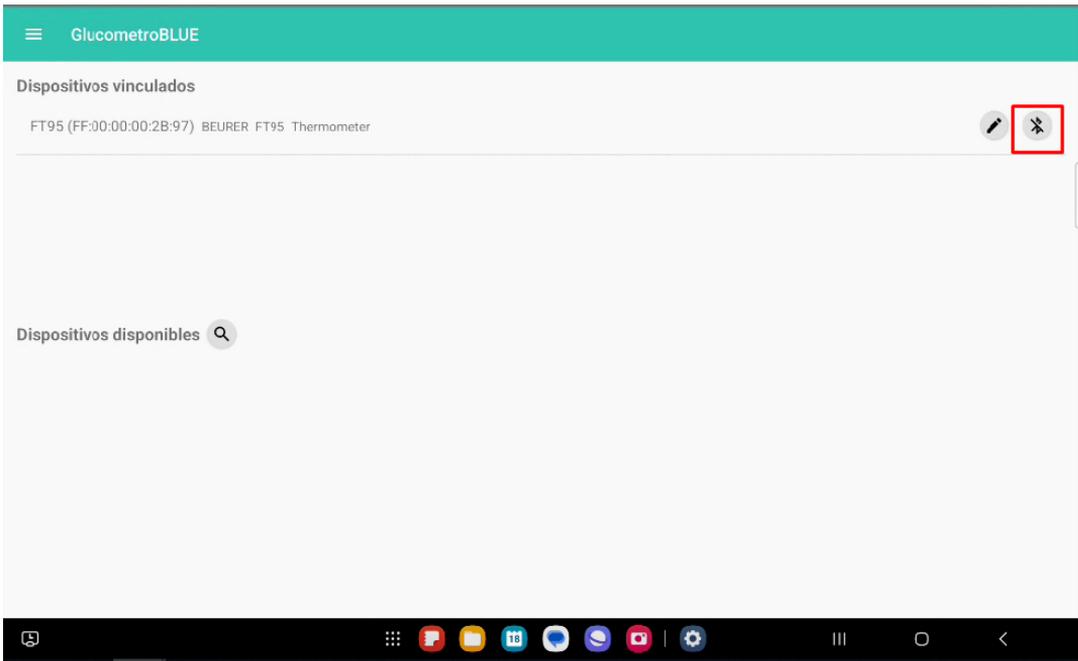


Figura B.7: Desvincular dispositivos

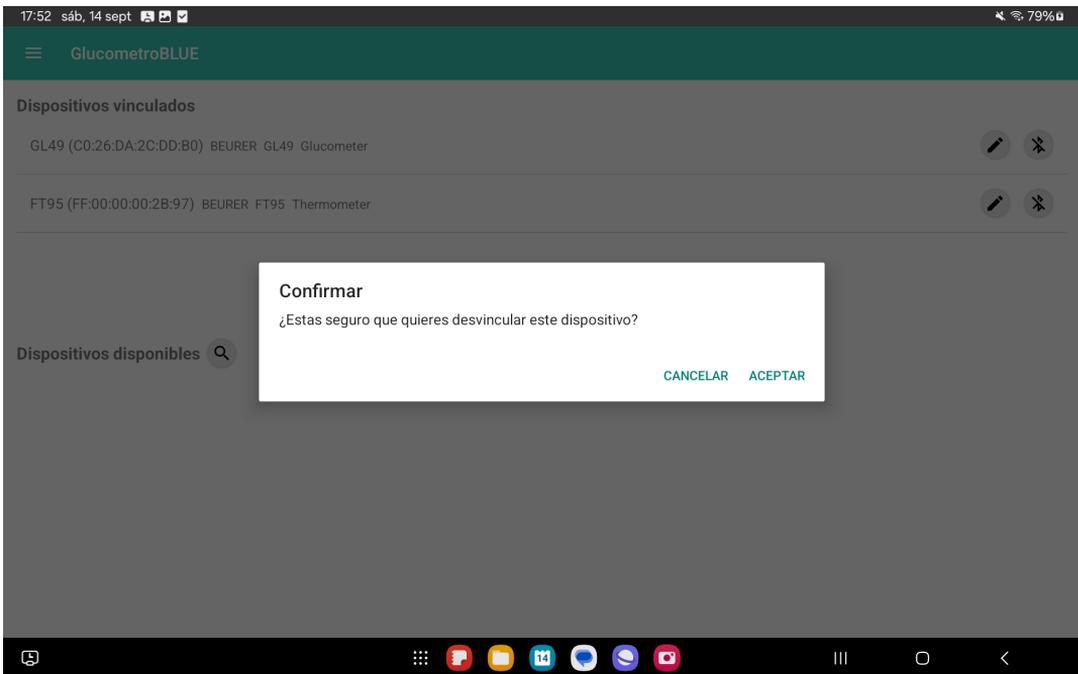


Figura B.8: Desvincular dispositivos

B.5. Visualizar datos

Para poder visualizar el historial de los datos obtenidos hasta el momento, es necesario acceder a una de las dos pantallas de visualización: ya sea la del glucómetro o la del termómetro. Una vez dentro, si existen registros, se podrán observar y navegar entre las diferentes páginas de la tabla (B.9).

Timestamp	Fecha	Temperatura	Unidad	Estado
1473237904000	2016-9-7 10:45:4	35,30	Celsius	Enviado
1722266418000	2024-7-29 17:20:18	34,90	Celsius	Enviado
1722266847000	2024-7-29 17:27:27	35,00	Celsius	Enviado
1722266879000	2024-7-29 17:27:59	35,00	Celsius	Enviado
1722267227000	2024-7-29 17:33:47	35,10	Celsius	Enviado
1722267252000	2024-7-29 17:34:12	35,00	Celsius	Enviado
1722267814000	2024-7-29 17:43:34	35,00	Celsius	Enviado
1722267934000	2024-7-29 17:45:34	35,10	Celsius	Enviado
1722267954000	2024-7-29 17:45:54	35,00	Celsius	Enviado
1722267977000	2024-7-29 17:46:17	35,00	Celsius	Enviado

SIGUIENTE

Página 1 de 7, Mostrando 1 a 10 de 61 registros

Figura B.9: Visualizar datos

B.6. Enviar datos manualmente

Si ocurre algún error en el envío de los datos durante una medición y no se desea esperar el envío automático programado, es posible realizar el envío manualmente pulsando el botón ubicado en la parte superior en ambas pantallas(B.10). Si hay datos disponibles para enviar, se enviarán y mostrará un mensaje de confirmación.

B.7. VISUALIZAR INFORMACIÓN Y EDITAR PARÁMETROS

The screenshot shows the GlucometroBLUE application interface. At the top, there is a teal header with a menu icon and the text 'GlucometroBLUE'. Below the header, there is a button labeled 'ENVIAR DATOS TEMPERATURA' (highlighted with a red box) and a circular refresh icon. The main content area displays a table with the following columns: Timestamp, Fecha, Temperatura, Unidad, and Estado. The table contains 10 rows of data. Below the table, there is a 'SIGUIENTE' button and a status message: 'Página 1 de 7, Mostrando 1 a 10 de 70 registros'. The bottom of the screen shows a standard Android navigation bar with various icons.

Timestamp	Fecha	Temperatura	Unidad	Estado
1473237904000	2016-9-7 10:45:4	35,30	Celsius	Enviado
1722267227000	2024-7-29 17:33:47	35,10	Celsius	Enviado
1722267252000	2024-7-29 17:34:12	35,00	Celsiua	Enviado
1722267814000	2024-7-29 17:43:34	35,00	Celsius	Enviado
1722267934000	2024-7-29 17:45:34	35,10	Celsius	Enviado
1722267954000	2024-7-29 17:45:54	35,00	Celsius	Enviado
1722267977000	2024-7-29 17:46:17	35,00	Celsius	Enviado
1722612362000	2024-8-2 17:26:2	35,10	Celsius	Enviado
1722612441000	2024-8-2 17:27:21	35,20	Celsius	Enviado
1722673749000	2024-8-3 10:29:9	35,30	Celsius	Enviado

Figura B.10: Enviar datos manualmente

B.7. Visualizar información y editar parámetros

Si se desea conocer el ID de la Tablet para hacer un seguimiento o modificar algún parámetro de configuración, es necesario entrar en la pantalla de información. Una vez dentro, en la parte superior se muestra el ID, y justo debajo se encuentra el intervalo de horas para el envío automático de datos. Si se desea cambiar el intervalo de envío, basta con modificar el valor y pulsar en el botón de guardar (B.11).

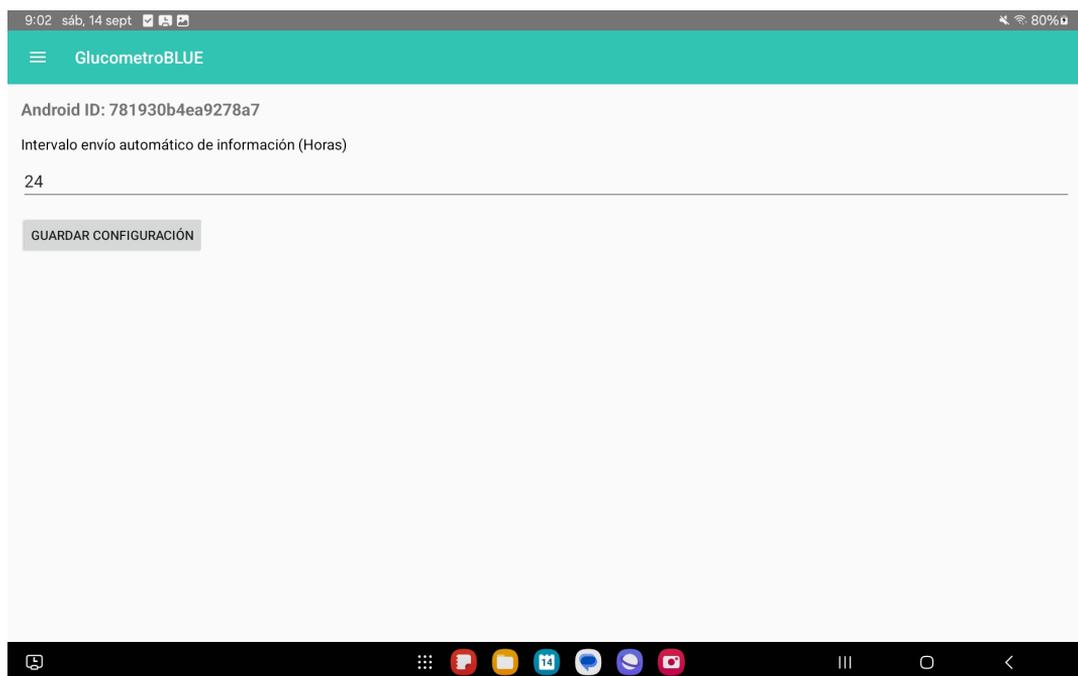


Figura B.11: Visualizar información y editar parámetros