



UNIVERSIDAD DE GRANADA

Facultad de Ciencias

GRADO EN INGENIERÍA
ELECTRÓNICA INDUSTRIAL

TRABAJO FIN DE GRADO
PLATAFORMA IOT
PARA EL
SEGUIMIENTO DE LA
ACTIVIDAD FÍSICA
EN PARQUES
BIOSALUDABLES

Presentado por:
D./D^a. Pablo Rubio Jiménez

Tutor:
Prof. D. Oresti Baños Legrán
Cotutor:
Prof. D. Miguel Damas Hermoso

Curso académico 2020/2021



UNIVERSIDAD DE GRANADA

GRADO EN INGENIERIA ELECTRÓNICA INDUSTRIAL

AUTORIZACIÓN DE LECTURA DE TRABAJO FIN DE CARRERA

D. Oresti Baños Legrán y D. Miguel Damas Hermoso profesores del Departamento de Arquitectura y Tecnología de Computadores de la Universidad de Granada, como director/es del Trabajo Fin de Grado titulado “Plataforma IoT para el seguimiento de la actividad física en parques biosaludables” y realizado por el alumno D. Pablo Rubio Jiménez.

CERTIFICA/N: que el citado Trabajo Fin de Grado, ha sido realizado y redactado por dicho alumno y autorizan su presentación.

Granada,

Fdo. Prof1_____

Fdo. Prof2_____



UNIVERSIDAD DE GRANADA

GRADO EN INGENIERIA ELECTRÓNICA INDUSTRIAL

AUTORIZACIÓN DE DEPÓSITO EN LA BIBLIOTECA

Yo, D/Dña. Pablo Rubio Jiménez con DNI 43213823C, autor del Trabajo Fin de Grado titulado “Plataforma IoT para el seguimiento de la actividad física en parques biosaludables” realizado en la Universidad de Granada

DECLARO: explícitamente que asumo la originalidad del trabajo, entendida en el sentido de que no ha utilizado fuentes sin citarlas debidamente.

AUTORIZO: al depósito de dicho Trabajo en la Biblioteca de la Universidad de Granada, y de la visualización a través de Internet.

Granada,

Fdo. D/Dña. Pablo Rubio Jiménez



UNIVERSIDAD DE GRANADA

GRADO EN INGENIERIA ELECTRÓNICA INDUSTRIAL

Plataforma IoT para el seguimiento de la actividad física en parques biosaludables

Autor: Pablo Rubio Jiménez

Directores: Oresti Baños Legrán

Miguel Damas Hermoso

Departamento: Arquitectura y Tecnología de Computadores

Palabras clave: IoT, parques biosaludables, LoRaWAN, BLE, monitorización, microcontrolador, backend, máquina fitness.

Resumen:

El uso de parques biosaludables está aumentando, sin embargo, se sabe muy poco sobre cuánto se utilizan. El objetivo de este proyecto es diseñar un dispositivo que se pueda instalar en los equipos de fitness de los parques biosaludables y que recopile datos del uso de los equipos. También se pretende diseñar toda una plataforma de Internet de las Cosas que procese los datos con el fin de visualizar y evaluar los resultados obtenidos.



UNIVERSIDAD DE GRANADA

GRADO EN INGENIERIA ELECTRÓNICA INDUSTRIAL

IoT platform for monitoring physical activity in outdoor fitness parks

Author: Pablo Rubio Jiménez

Directors: Oresti Baños Legrán

Miguel Damas Hermoso

Department: Computer Architecture and Technology.

Keywords: IoT, outdoor fitness parks, LoRaWAN, BLE, monitoring, microcontroller, backend, outdoor fitness equipment

Abstract:

The use of outdoor fitness parks is increasing, however very little is known about the how much are used these parks. The aim of this project is to design a device that can be installed in the fitness equipment of the outdoor fitness parks and that gathers data of the usage of the equipment. It is also intended to design an entire Internet of Things platform that processes the data in order to visualize and evaluate the results obtained

Índice general

Índice de figuras	7
Índice de tablas	8
Índice de listings	9
1. Introducción	10
1.1. Contexto	10
1.2. Motivación	10
1.3. Objetivos	11
1.4. Estructura del proyecto	12
2. Estado del arte	13
2.1. Estudios, patentes y aplicaciones en la actualidad	13
2.2. Tabla resumen de los artículos, patentes y aplicaciones estudiados . . .	18
3. Análisis y propuestas de diseño para parques biosaludables	25
3.1. Tipos de equipos de gimnasia	25
3.1.1. Trabajo del Tren Superior	26
3.1.2. Trabajo del Tren Inferior	29
3.1.3. Trabajo del Tren Superior e Inferior	30
3.2. Legislación	31
3.2.1. UNE 16630:2015	31
3.2.2. ASTM F3101-15	34
3.3. Análisis DAFO	35
3.4. Conclusión del análisis	37
3.5. Propuestas de diseño	38
3.5.1. Propuesta de diseño 1	38
3.5.2. Propuesta de diseño 2	39
3.5.3. Propuesta de diseño 3	41
3.5.4. Propuesta de diseño 4	41
3.5.5. Propuesta de diseño 5	42
3.5.6. Propuesta de diseño 6	42

4. Estudio de diseño propuesto	44
4.1. Tipo de Sensor	45
4.2. Tipo de conexión a Internet	45
4.3. Servidor de Red	47
4.4. Tipo de conexión entre nodos y dispositivo central	47
4.5. Microcontrolador para dispositivo maestro	50
4.6. Microcontrolador para dispositivo nodo	52
4.7. Batería y consumo de los dispositivos	53
4.7.1. Consumo del dispositivo maestro	53
4.7.2. Consumo del dispositivo nodo	56
4.8. Protección de los dispositivos	58
5. Desarrollo del diseño	60
5.1. Diseño Electrónico	60
5.1.1. Listado de componentes	60
5.1.2. Pines y conexiones	61
5.1.3. Esquema electrónico	64
5.1.4. Diseño de la placa PCB	66
5.2. Diseño Software	72
5.2.1. Componentes necesarios para el Software	72
5.2.2. Funcionamiento general de los dispositivos	73
5.2.3. Registro de los datos	74
5.2.4. Interrupción por Actividad e Inactividad	74
5.2.5. Funcionamiento del Timer	75
5.2.6. Envío y recepción de datos por BLE	75
5.2.7. Envío de datos por LoRaWAN	75
5.2.8. Protocolo	76
5.2.9. Programación de los dispositivos	86
5.3. Diseño del backend	87
5.3.1. Tratamiento de los datos	88
5.3.2. Visualización de los datos	91
5.4. Diseño de la carcasa	96
6. Planificación y presupuesto	98
6.1. Diagrama de Gantt	98
6.2. Presupuesto	104
6.2.1. Coste material	104
6.2.2. Coste de mano de obra	104
7. Conclusiones y líneas futuras	106
Bibliografía	108
ANEXOS	111

ÍNDICE GENERAL

A. Código del dispositivo maestro	112
B. Código del dispositivo nodo	131

Índice de figuras

2.1. Diagrama esquemático de la estructura de las instalaciones de interacción de energía solar. EPEE 2016. Xiyin Gao, Yue Han y Ke Su. (2016)	14
2.2. Diseño de equipo fitness al exterior. Outdoor Fitness Equipment Auxiliary Device. Hu Hong (2019). Fuente: Patente N°. CN208582936 . .	16
2.3. Diseño de caseta para parque biosaludable. Outdoor fitness equipment with intelligent action recognition function. Zhang Peifei (2014). Fuente: Patente N°. CN104147772	17
3.1. Circuito de Calistenia. Fuente: Deportes Urbanos	25
3.2. Máquina Volante. Fuente: Parques Troe (2015)	26
3.3. Máquina Autobús. Fuente: B2Sports	26
3.4. Máquina Caballo. Fuente: Parques Troe (2015)	27
3.5. Máquina Cintura. Fuente: Parques Troe (2015)	27
3.6. Máquina Timón. Fuente: Parques Troe (2015)	27
3.7. Máquina Surf. Fuente: Parques Troe (2015)	27
3.8. Máquina Ascensor. Fuente: Parques Troe (2015)	28
3.9. Máquina Masaje. Fuente: Grup Fábregas	28
3.10. Máquina Baúl. Fuente: B2Sport	28
3.11. Máquina Esqui de fondo. Fuente: Grup Fábregas	28
3.12. Máquina Barras. Fuente: Parques Troe (2015)	29
3.13. Máquina Tumbona. Fuente: Parques Troe (2015)	29
3.14. Máquina Paloma. Fuente: B2Sport	29
3.15. Máquina Patines. Fuente: Parques Troe (2015)	30
3.16. Máquina Jota. Fuente: Pavimentos Infantiles	30
3.17. Máquina Columpio. Fuente: Parques Troe (2015)	30
3.18. Máquina Pony. Fuente: Grup Fábregas	31
3.19. Máquina Helicóptero. Fuente: B2Sport	31
3.20. Esquema de propuesta de diseño 1. Fuente: elaboración propia.	39
3.21. Esquema de propuesta de diseño 2. Fuente: elaboración propia.	40
3.22. Esquema de propuesta de diseño 3. Fuente: elaboración propia.	41
3.23. Esquema de propuesta de diseño 4. Fuente: elaboración propia.	42
3.24. Esquema de propuesta de diseño 5. Fuente: elaboración propia.	43
3.25. Esquema de propuesta de diseño 6. Fuente: elaboración propia.	43

ÍNDICE DE FIGURAS

4.1. Módulo RAK4600. Fuente: RAKWireless.	52
4.2. Esquema de la estructura del diseño. Fuente: elaboración propia.	59
5.1. Diagrama de bloques del RAK4600. Fuente: RAKWireless.	61
5.2. Esquema de conexiones del RAK4600. Fuente: RAKWireless.	62
5.3. Conexiones del acelerómetro ADXL345. Fuente: Analog Devices.	63
5.4. Prototipo del dispositivo. Fuente: elaboración propia.	64
5.5. Esquema electrónico. Fuente: elaboración propia.	65
5.6. Placa PCB parte frontal. Fuente: elaboración propia.	66
5.7. Placa PCB parte trasera. Fuente: elaboración propia.	67
5.8. Placa 3D frontal. Fuente: elaboración propia.	68
5.9. Placa 3D trasera. Fuente: elaboración propia.	69
5.10. Placa 3D lateral. Fuente: elaboración propia.	70
5.11. Prototipo de placa PCB fabricada sin componentes. Fuente: elaboración propia.	71
5.12. Prototipo de placa PCB fabricada con componentes, imagen 1. Fuente: elaboración propia.	71
5.13. Prototipo de placa PCB fabricada con componentes, imagen 2. Fuente: elaboración propia.	72
5.14. RAKDAP1. Fuente: RAKWireless	73
5.15. FT232. Fuente: Amazon	73
5.16. The Things Indoor Gateway. Fuente: The Things Network	76
5.17. Diagrama del protocolo para el dispositivo central-1. Fuente: elaboración propia.	77
5.18. Diagrama del protocolo para el dispositivo central-2. Fuente: elaboración propia.	78
5.19. Diagrama del protocolo para el dispositivo central-3. Fuente: elaboración propia.	79
5.20. Diagrama del protocolo para el dispositivo central-4. Fuente: elaboración propia.	80
5.21. Diagrama del protocolo para el dispositivo central-5. Fuente: elaboración propia.	81
5.22. Diagrama del protocolo para el dispositivo nodo-1. Fuente: elaboración propia.	83
5.23. Diagrama del protocolo para el dispositivo nodo-2. Fuente: elaboración propia.	84
5.24. Diagrama del protocolo para el dispositivo nodo-3. Fuente: elaboración propia.	85
5.25. Diagrama de estados del dispositivo nodo. Fuente: elaboración propia.	86
5.26. Diagrama de estados del dispositivo maestro. Fuente: elaboración propia.	87
5.27. Plataforma de TTN. Fuente: elaboración propia.	88
5.28. Dashboard en TagoIO. Fuente: elaboración propia.	91

5.29. Captura de la aplicación de TagoIO. Imagen 1. Fuente: elaboración propia.	92
5.30. Captura de la aplicación de TagoIO. Imagen 2. Fuente: elaboración propia.	93
5.31. Captura de la aplicación de TagoIO. Imagen 3. Fuente: elaboración propia.	93
5.32. Diseño de carcasa. Imagen 1. Fuente: elaboración propia.	96
5.33. Diseño de carcasa. Imagen 2. Fuente: elaboración propia.	97
5.34. Diseño de carcasa. Imagen 3. Fuente: elaboración propia.	97
5.35. Diseño de carcasa. Imagen 4. Fuente: elaboración propia.	97
6.1. Diagrama de Gantt. Imagen 1. Fuente: elaboración propia.	99
6.2. Diagrama de Gantt. Imagen 2. Fuente: elaboracion propia.	100
6.3. Diagrama de Gantt. Imagen 3. Fuente: elaboración propia.	101
6.4. Diagrama de Gantt. Imagen 4. Fuente: elaboración propia.	102
6.5. Diagrama de Gantt. Imagen 5. Fuente: elaboración propia.	103

Índice de tablas

2.1.	Tabla resumen de los artículos estudiados	19
2.2.	Tabla resumen de los artículos estudiados (continuación de Tabla 2.1)	20
2.3.	Tabla resumen de los artículos estudiados (continuación de la Tabla 2.2)	21
2.4.	Tabla resumen de las patentes estudiadas	22
2.5.	Tabla resumen de las patentes estudiadas (continuación de la Tabla 2.4)	23
2.6.	Tabla resumen de las aplicaciones comerciales estudiadas	24
3.1.	Medidas mínimas del espacio de ejercicio	33
3.2.	Tipos de suelo en función de la altura de caída libre	33
3.3.	Ánalisis DAFO. Fortalezas y debilidades	36
3.4.	Ánalisis DAFO. Oportunidades y amenazas.	37
3.5.	Datos posibles a obtener	40
4.1.	Comparativa entre Sigfox, LoRaWAN y NB-IoT. Kais Mekki et al. (2019) [29]	48
4.2.	Comparación de Zigbee y BLE	49
4.3.	Diferentes módulos en el mercado	51
4.4.	Estimación de consumo medio de dispositivo maestro con RAK4600 .	56
4.5.	Estimación de consumo medio de dispositivo nodo con RAK4600 .	58
5.1.	Conexiones del RAK4600.	63
6.1.	Costes de los materiales y recursos	104
6.2.	Costes de la mano de obra	104
6.3.	Costes totales del proyecto.	105

Índice de listings

5.1.	Decodificador de datos en TTN. Fuente: elaboración propia.	87
5.2.	Tratamiento de los datos en TagIO. Fuente: elaboración propia . . .	89
5.3.	Widget para envío de datos Downlink en TagIO. Fuente: TagIO . .	94
A.1.	Código del dispositivo maestro. Fuente: elaboración propia.	112
A.2.	Código del dispositivo maestro (continuación del listing A.1). Fuente: elaboración propia. .	114
A.3.	Continuación del listing A.2. Fuente: elaboración propia.	116
A.4.	Continuación del listing A.3. Fuente: elaboración propia.	116
A.5.	Continuación del listing A.4. Fuente: elaboración propia.	121
A.6.	Continuación del listing A.5. Fuente: elaboración propia.	124
A.7.	Continuación del listing A.6. Fuente: elaboración propia.	125
A.8.	Continuación del listing A.7. Fuente: elaboración propia.	126
B.1.	Código del dispositivo nodo. Fuente: elaboración propia.	131
B.2.	Continuación del listing B.1. Fuente: elaboración propia	132
B.3.	Continuación del listing B.2. Fuente: elaboración propia.	133
B.4.	Continuación del listing B.3. Fuente: elaboración propia.	134
B.5.	Continuación del listing B.4. Fuente: elaboración propia.	136
B.6.	Continuación del listing B.5. Fuente: elaboración propia.	137
B.7.	Continuación del listing B.6. Fuente: elaboración propia.	138

Capítulo 1

Introducción

1.1. Contexto

Los parques biosaludables son espacios verdes o espacios al aire libre ubicados en las ciudades para la realización de la actividad deportiva. Están compuestos por equipos de gimnasia y normalmente se presentan en modo de circuito deportivo. Estos equipos o máquinas suelen ser robustas, de tal forma, que puedan soportar las inclemencias atmosféricas, el trato urbano y los problemas que puedan surgir por estar a la intemperie. También son máquinas bastante sencillas, de forma que no es muy complicado su uso. No están diseñados para un rango determinado de edad, aunque es cierto que los principales consumidores son las personas mayores. No obstante, cada vez más, estos equipos son usados por gente joven.

Son muy conocidos los beneficios y las ventajas de los parques biosaludables, sin embargo, se ignora el uso que se les da. No se sabe ni qué máquinas se usan, cuánto ni cómo. Conviene destacar que estos equipos deportivos se ubican en zonas de exterior, en ocasiones alejadas de puntos de electricidad y también alejadas de los sistemas de telecomunicación más comunes. Esto no ocurre en los lugares de interior como pueden ser los gimnasios, en los que las máquinas tienen un acceso fácil a la electricidad y a la posibilidad de obtención y transmisión de datos deportivos. De este desconocimiento sobre los parques biosaludables, surge la idea de monitorizar estos equipos deportivos de forma similar a las máquinas de gimnasio, pero teniendo en cuenta los requerimientos de éstos.

1.2. Motivación

Los avances de la tecnología en la actualidad han hecho que el Internet de las Cosas (Internet of Things, IoT) esté en constante crecimiento y cada vez más personas conozcan el término y estén familiarizados con él. IoT no usa una tecnología específica, sino que cualquier tecnología puede formar parte de un sistema IoT. La idea es interconectar dispositivos con sensores que sean capaces de recolectar datos

de cualquier ámbito sin la necesidad de una persona física y éstos se puedan compartir o enviar a una red privada o a internet. En la actualidad, estos sistemas capaces de recolectar datos y enviarlos están presentes en todo tipo de sectores, incluso en áreas cotidianas, como puede ser el hogar. Uno de los retos del IoT es poder crear una red de dispositivos interconectados capaces de obtener la mayor cantidad de datos útiles posibles en cualquier ámbito. Es por ello, que el progreso de la tecnología y el IoT van de la mano. Este avance es lo que va a permitir poder llevar el IoT a lugares no tan accesibles y obtener datos de mayor calidad.

En la actualidad existen una gran cantidad de sensores de todo tipo capaces de medir cualquier magnitud, desde la aceleración hasta la altura o la posición gps. Uno de los mayores avances que ha permitido el auge del IoT han sido los protocolos y tecnologías de telecomunicación y los SoC (System on a Chip). Estos últimos son chips de tamaño reducido con todos los módulos y elementos necesarios para poder formar un pequeño computador. Estos SoC son cada vez más pequeños con un mayor rendimiento y con un muy bajo consumo. En cuanto a los protocolos y tecnologías de telecomunicación, en la actualidad, existen sistemas capaces de enviar datos a corta, a media o incluso a muy larga distancia del orden de kilómetros, tales como Sigfox o LoRa. Este crecimiento de la tecnología y la importancia que está teniendo el IoT son la principal motivación para la realización de este proyecto.

1.3. Objetivos

El objetivo principal de este proyecto es desarrollar un sistema que permita la monitorización de los equipos de fitness ubicados en los parques biosaludables de forma remota. De este objetivo principal surgen otros mucho más específicos:

- Estudiar en profundidad los tipos de equipos deportivos, sus usos y sus beneficios.
- Investigar sistemas de telecomunicaciones capaces de enviar datos desde los parques biosaludables a Internet con un bajo consumo.
- Diseñar un dispositivo totalmente funcional, que pueda ser instalado en el equipo deportivo de los parques biosaludables y sea capaz de obtener datos de su uso.
- Implementar el diseño anteriormente mencionado.
- Reducir el consumo del dispositivo al mínimo posible.
- Diseñar una plataforma digital en la que poder tratar y visualizar los datos que se vayan recibiendo de los equipos deportivos.
- Implementar la plataforma digital mencionada en el punto anterior.

1.4. Estructura del proyecto

La elaboración de este proyecto se ha dividido en 7 capítulos en los que se detalla tanto la búsqueda e investigación sobre el tema a tratar, el desarrollo del proyecto y las conclusiones obtenidas de éste. También se han añadido dos anexos al final de éste con el código utilizado. A continuación se realiza una pequeña explicación de cada capítulo.

- Capítulo 1: Introducción. Se expone el contexto, las motivaciones, los objetivos y la estructura del presente proyecto.
- Capítulo 2: Estado del arte. En este capítulo se presentan todos los estudios y proyectos encontrados a día de hoy que tratan de solucionar el tema en cuestión.
- Capítulo 3: Análisis y propuestas de diseño para parques biosaludables. En este capítulo se estudian en profundidad los tipos de equipos de gimnasia usados en los parques biosaludables, junto con la legislación que presentan. También se analizan las oportunidades y amenazas, junto con las fortalezas y debilidades que ofrecen. Y en función de ellas, se exponen diferentes propuestas de diseño.
- Capítulo 4: Estudio de diseño propuesto. Una vez seleccionado el diseño a tratar, en este capítulo se estudia en profundidad la tecnología a usar para cumplir con dicho diseño.
- Capítulo 5: Desarrollo del diseño. Este capítulo trata de mostrar el diseño e implementación del diseño propuesto.
- Capítulo 6: Planificación y presupuesto. Este capítulo trata de mostrar la planificación llevada en la elaboración del proyecto y los costes asociados a éste.
- Capítulo 7: Conclusiones y líneas futuras. Este es el último capítulo y trata de exponer las conclusiones de todo el estudio y proponer líneas futuras o usos futuros para el proyecto diseñado.

Capítulo 2

Estado del arte

2.1. Estudios, patentes y aplicaciones en la actualidad

Los parques biosaludables, o conocidos anteriormente como "parques geriátricos", tienen el potencial de acercar a la población en general al deporte, pero se sabe muy poco de qué tipo de personas usan estos equipos de gimnasia, cómo y cuánto los usan. Por ello, la existencia de esa necesidad de monitorizar las máquinas de gimnasia con el fin de obtener datos de su uso. Esos datos podrían llevar a mejorar los parques biosaludables e incluso para que el propio deportista pudiera realizar un entrenamiento de mayor calidad.

En la actualidad, no es mucha la información que existe sobre la monitorización de los parques biosaludables. De hecho, la gran mayoría de los datos obtenidos en estudios, se han realizado mediante el método SOPARC (System for Observing Play and Recreation in Communities)[1] [2]. Este método consiste en la adquisición de datos manual mediante la propia visualización. Es decir, se usa a una persona para que observe las máquinas de gimnasia y tome datos, tales como el género o la edad del deportista, el tiempo de uso o el tipo de máquina usada. Estos estudios no suelen tener una larga duración, ya que es muy difícil que una persona observe y tome datos constantemente y durante un largo periodo de tiempo. Como consecuencia, la importancia de conseguir un sistema de monitorización o seguimiento de la actividad física automatizado para estos equipos de gimnasia exteriores.

Hoy en día, los parques biosaludables no son una ciencia muy estudiada, sin embargo hay algunos proyectos o investigaciones que han estudiado la posibilidad de monitorizar los equipos deportivos. Hsu, Chang, Chen, Lin y Chou (2020) [3] llevaron a cabo una investigación en Taiwán sobre un dispositivo de monitorización IoT (Internet of Things) para este tipo de máquinas, en el que el usuario podía visualizar sus datos deportivos en una aplicación en el móvil. El dispositivo consistía en la sensorización de las máquinas y el envío de los datos a la nube. Todo ello se

realizaba mediante un sensor de posición que monitoreaba el paso de una persona en una máquina, en concreto la máquina de fitness conocida como AirWalker, muy similar a una máquina elíptica típica de gimnasio, en España es conocida como esquí de fondo, tal y como se muestra en figura 3.11. Despues esos datos eran enviados a la nube mediante un microcontrolador y rescatados por los usuarios mediante una aplicación.

De forma similar, Cori Sweet, Soghra Jarvandi, Heather Sedges y Karen Franc (2020) [4] realizaron otro estudio sobre cómo y cuando se usan este tipo de máquinas deportivas. En este caso, no se diseñó un sistema de sensorización, sino que se usó un dispositivo comercial de monitorización de la actividad física, conocido como FitBit Zip. Este dispositivo usa un acelerómetro de 3 ejes para obtener los datos. En este caso, como este dispositivo comercial no está dedicado a máquinas de gimnasia, solamente se tomaron como datos viables los tiempos de uso. Los datos en este caso no se subieron a la nube y no estaban destinados al usuario, sino que se recogieron mediante la tecnología Bluetooth y la propia aplicación comercial del móvil. El objetivo de los datos era conocer cuándo y cómo se usan las máquinas de ejercicio al aire libre en los parques biosaludables para posibles inversiones o mejoras de éstas.

No solo se ha investigado la idea de obtener datos sobre el uso de los parques biosaludables, también se ha investigado la idea de que el propio usuario pueda interaccionar con las máquinas. Yue Han, Xiyin Gao y Ke Su (2016) [5] estudiaron esa idea, y su diseño, mostrado en la Figura 2.1, se basó en tres módulos: módulo de energía solar, módulo de interacción y módulo de fitness. El primer módulo, el de energía solar, se encargaría de la generación de electricidad mediante el uso de la luz solar, mediante placas solares. Este módulo alimentaría a los siguientes.

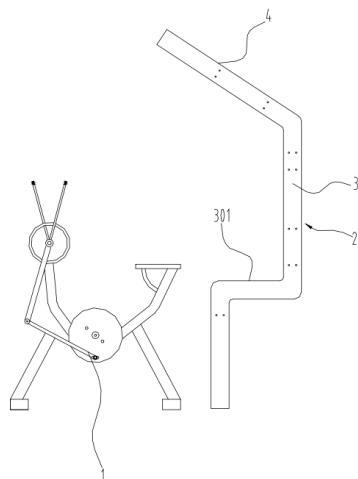


Figura 2.1: Diagrama esquemático de la estructura de las instalaciones de interacción de energía solar. EPEE 2016. Xiyin Gao, Yue Han y Ke Su. (2016)

El módulo de interacción se basaba en el concepto de que el deportista pudiera interaccionar con la máquina. Se usaría un sensor de infrarrojos para que el módulo fuera capaz de detectar cuando hubiera una persona y ésta fuera capaz de visualizar los datos deportivos que fuera generando según se ejercitara.

De igual forma, Xiyin Gao, Chunrong Zhao y Yue Han (2017) [6] propusieron tres diseños de equipos de gimnasia al aire libre basados en las demandas del modelo Kano, teoría de desarrollo de productos y de satisfacción del cliente. Los tres diseños propuestos fueron: diseño de una máquina fitness solar interactiva hombre-máquina, diseño de instalaciones de fitness de paternidad y diseño de la bicicleta de fitness para niños. Con respecto al primer diseño que se propuso, los equipos serían capaces de interactuar con el usuario en los campos de guía de voz, reproductor de música, información, retroalimentación, etc. Mediante un sensor de infrarrojos la máquina detectaría cuando una persona va a usar el equipo deportivo y enviaría una señal a una unidad de control. Esta unidad se encargaría de activar el reproductor de música o la pantalla con datos como el tiempo de uso o cualquier tipo de dispositivo con la función de interactuar con el usuario. De igual forma que Yue Han, et al. (2016) [5], para alimentar todo el sistema de interacción del equipo se propuso un módulo solar y una batería, que se cargaría mediante la generación de electricidad con un panel solar.

A parte de estos artículos e investigaciones también existen patentes y aplicaciones comerciales capaces de monitorizar la actividad física en los parques biosaludables. De hecho, en la patente Monitoring device for outdoor bodybuilding equipment (China. Patente Nº. CN104850942, 2018) [7] se usa un sensor de aceleración y un giroscopio para obtener datos y enviarlos mediante la tecnología ZigBee a una estación base donde se procesarían los datos y se subirían a un servidor en la nube mediante GPRS o 3G.

De forma similar se envían y se procesan los datos en la patente Microwave radar device for outdoor fitness equipment (China. Patente Nº. CN110119114, 2019)[8], la cual se distingue por la forma de obtener esos datos. En esta patente, el diseño es capaz de captar la ubicación geográfica en tiempo real, el tipo de equipo, la frecuencia de uso, el género y la edad del usuario. Para ello, propone usar un sensor de infrarrojos que reconozca si alguien se encuentra usando la máquina. Mediante un módulo de adquisición de datos compuesto por un sensor de aceleración y un giroscopio conectados al eje de rotación del equipo se obtienen ciertos datos. Y mediante otro módulo de reconocimiento facial compuesto por un detector de distancia, un colector de imágenes y un procesador de imágenes se podría captar el género y la edad de las personas.

Otro tipo de diseño propuesto en diferentes patentes es el de añadir una pantalla para poder interaccionar con el usuario y que éste pueda observar sus datos deportivos según se ejercita. La patente Outdoor Fitness Equipment Auxiliary De-

vice (China. Patente N°. CN208582936, 2019) [9] tal y como se ve en la Figura 2.2 incluye un poste a modo de soporte, donde se ubica una pantalla de visualización. El poste sería hueco por el interior para que pudiese circular la línea de suministro de energía solar, generada mediante una placa fotovoltaica colocada en la parte superior del poste. También se incluiría una especie de cubierta con iluminación LED para que se pudiera ejercitarse de noche y en días lluviosos. Los sensores usados en el equipo de gimnasia al exterior serían un sensor de inducción magnética o un sensor fotoeléctrico con un disco codificador según el tipo de equipo. Los sensores irían conectados a un procesador de datos y a la pantalla.

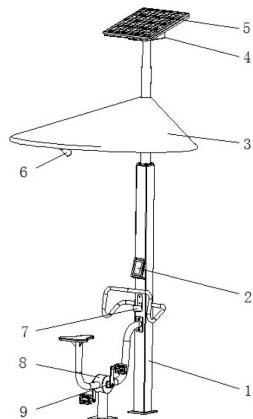


Figura 2.2: Diseño de equipo fitness al exterior. Outdoor Fitness Equipment Auxiliary Device. Hu Hong (2019). Fuente: Patente N°. CN208582936

Muy similar es la patente Outdoor intelligent body building station (China, Patente N°. CN104147772, 2014) [10] que se basa en una caseta exterior donde se colocan los equipos de fitness al aire libre. El diseño propuesto se puede visualizar en la Figura 2.3 y consta de un sensor de velocidad, un temporizador, una pantalla, un procesador de datos, una unidad de multimedia, una unidad de monitorización y una unidad de posicionamiento de GPS. La energía se generaría mediante paneles solares colocados en el tejado de la caseta, mediante un molino de viento colocado también en el tejado y mediante el propio movimiento de las máquinas de fitness. Los datos serían procesados y mostrados en las pantallas.

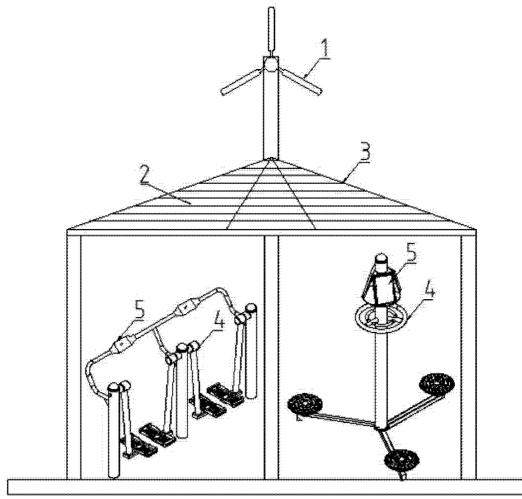


Figura 2.3: Diseño de caseta para parque biosaludable. Outdoor fitness equipment with intelligent action recognition function. Zhang Peifei (2014). Fuente: Patente Nº. CN104147772

Otro tipo de adquisición de datos en las máquinas de parques biosaludables es la propuesta en la patente Outdoor fitness equipment with intelligent action recognition function (China, Patente Nº. CN110141830, 2019) [11], en la que se propone instalar unos marcadores visuales en las partes móviles de los equipos, de tal forma que cuando se ejercite, los marcadores se muevan con la parte móvil de la máquina. Mediante una cámara se podría detectar ese movimiento de los marcadores y con un sistema de análisis de imágenes, posicionar constantemente los marcadores con coordenadas con respecto a un punto de referencia, y obtener datos tales como el tiempo de uso, la velocidad de ejercitación o los kilómetros recorridos.

Con respecto a las aplicaciones móviles (apps), en la actualidad existen varias que permiten registrar todos los entrenamientos realizados. La gran mayoría de aplicaciones son comerciales, es decir, sirven solamente para determinadas máquinas de la propia compañía de la aplicación. Como es el caso de TGO Activate [12], Norwell Outdoor Fitness v2 [13], Proludic Sport [14] o Caloo Pulse App [15] que permiten registrar datos de los ejercicios seleccionando la máquina de gimnasia concreta de su empresa y el parque en el que se encuentra. Todas requieren un registro previo en el que hay que aportar datos personales como la edad, el peso, la condición física, el género, la altura o incluso la frecuencia cardíaca en reposo. Todas las aplicaciones anteriores son gratuitas y están disponibles tanto para el sistema operativo de Android como para iOS.

A parte de estas aplicaciones en las que solamente se pueden ubicar y seleccionar las máquinas de la propia empresa, existe la aplicación LPA Entrénate [16] diseñada por el Ayuntamiento de Las Palmas de Gran Canaria con la finalidad de promover

el deporte en los parques biosaludables. La aplicación sirve para buscar y ubicar los diferentes parques biosaludables que hay en la ciudad de Las Palmas de Gran Canaria. Igual que en las otras aplicaciones, también es necesario un registro con datos personales. La aplicación registra de forma sencilla datos tales como la duración del entrenamiento, las calorías consumidas, los parques visitados o los entrenamientos realizados.

En definitiva, no se han realizado muchos estudios sobre la actividad física que proporcionan estos tipos de equipos de gimnasia al aire libre, de hecho, los pocos realizados han sido mediante encuestas o por el método mencionado anteriormente (SOPARC) [1] [2]. En cuanto a los artículos o investigaciones encontrados y mencionados, todos tienen en común la necesidad de sensorizar los equipos para poder monitorizar constantemente la actividad física. Las patentes encontradas, igual que los artículos e investigaciones, también tienen en común, que todos utilizan diferentes tipos de sensores en los equipos, para conseguir aportar la mayor cantidad de datos deportivos. No ocurre lo mismo en cuanto a las aplicaciones comerciales ya existentes en el mercado, las cuales no usan ningún tipo de sensor, solamente con la aplicación, ciertos datos personales e introduciendo el tiempo y tipo de equipo a usar, muestran datos tales como las calorías consumidas o el tiempo de uso.

2.2. Tabla resumen de los artículos, patentes y aplicaciones estudiados

A continuación, en las Tablas 2.1, 2.2, 2.3, 2.4, 2.5 y 2.6 se pueden ver en detalle las características de todos los artículos, patentes y aplicaciones comerciales encontrados y estudiados.

Tabla 2.1: Tabla resumen de los artículos estudiados

Estudios	Tecnología de sensado	Envío de datos	Tecnología IoT	Interfaz de Usuario	...
Hsu et al. (2020) [3]	Switch Reed como sensor de posición.	Se usa una conexión Wifi.	ESP8266 para enviar los datos mediante Wifi a un servidor. Mediante un software de interfaz (API) se comunica con los usuarios.	Se usa una aplicación llamada LINE.	...
Sweet et al. (2020) [4]	Dispositivo comercial, en este caso el FitBit Zip. Es un dispositivo con un acelerómetro de 3 ejes.	Se usa una conexión bluetooth entre el dispositivo FitBit Zip y el móvil.	N/A	El usuario puede ver los datos mediante la aplicación comercial de FitBit.	...
Choi et al. (2013) [17]	Acelerómetro de 3 ejes conectado a la máquina de ejercicio.	N/A	N/A	N/A	...
Chow y Ho (2018) [18]	Dispositivo de medición metabólica portatil (Cosmed K4B2) y un cinturón polar para la frecuencia cardíaca que se le coloca al usuario.	N/A	N/A	N/A	...
Chow et. al (2017) [19]	SOPARC (System for Observing Play and Recreation in Communities). Mediante un observador independiente tomando datos.	N/A	N/A	N/A	...
Gao et al. (2016) [5]	Sensor infrarrojos	N/A	N/A	Reproductor de música instalado en el dispositivo, no especifica si habría una pantalla o no.	...
Gao et al. (2017) [6]	Sensor infrarrojos	Se envían a una unidad de control mediante cable.	N/A	Reproductor de música y pantalla con el tiempo.	...
Cranney et al. (2015) [20]	Observación (SOPARC) y entrevistas a los usuarios.	N/A	N/A	N/A	...
Chow (2013) [21]	Observación (SOPARC) y entrevistas a los usuarios.	N/A	N/A	N/A	...

CAPÍTULO 2. ESTADO DEL ARTE

Tabla 2.2: Tabla resumen de los artículos estudiados (continuación de Tabla 2.1)

Estudios	Datos Obtenidos	nº Particípantes	Duración del estudio	Robustez	Seguridad	...
Hsu et al. (2020) [3]	Longitud y frecuencia de los pasos y tiempo de uso de la máquina.	3 mujeres entre 24 y 27 años. Y un entrenador personal	60 días	N/A	N/A	...
Sweet et al. (2020) [4]	Tiempos de uso de la máquina.	N/A	150 días	Pequeña caja magnética, donde se introduce el Fitbit.	Cinta adhesiva del mismo color que la máquina para disimularlo.	...
Choi et al. (2013) [17]	Número de veces usada la máquina.	N/A	N/A	El sensor de aceleración es robusto y fiable frente al paso del tiempo o posibles golpes.	N/A	...
Chow y Ho (2018) [18]	Intensidad de los ejercicios de las maquinas fitness al aire libre.	16 adultos(8 hombre y 8 mujeres) de más de 65 años	N/A	N/A	N/A	...
Chow et. al (2017) [19]	Género del usuario, etnias, edades número de usuarios y intensidad de la actividad física	Observadores (no se especifican cuantos)	N/A	N/A	N/A	...
Gao et al. (2016) [5]	Tiempo de uso y efecto del ejercicio.	N/A	N/A	Los materiales y el diseño de la estructura deben soportar daños provocados por el hombre.	Toda la parte susceptible está en el interior de un poste.	...
Gao et al. (2017) [6]	Tiempo de uso, frecuencia de uso.	N/A	N/A	N/A	N/A	...
Cranney et al. (2015) [20]	Género del usuario, categoría de edad, nivel de actividad.	Usuarios del parque de habla inglesa mayores de 18 años.	423 días	N/A	N/A	...
Chow (2013) [21]	Percepciones de las máquinas y experiencias con su uso.	55 personas mayores de 50 años (27 hombres y 28 mujeres)	N/A	N/A	N/A	...

Tabla 2.3: Tabla resumen de los artículos estudiados (continuación de la Tabla 2.2)

Estudios	Localización	Anonimidad de los datos	Conclusión
Hsu et al. (2020) [3]	Taiwán	N/A	Se obtuvieron buenos datos sobre como mejora la flexibilidad, resistencia y potencia muscular de los participantes a lo largo de todo el estudio.
Sweet et al. (2020) [4]	Desconocida (9 equipos repartidos en 6 pueblos rurales de menos de 10.000 personas)	Se obtienen datos anónimos.	Ayudó a comprender mejor cómo y cuando las personas usan el equipo.
Choi et al. (2013) [17]	En cuatro máquinas de ejercicio al aire libre. No se especifica la ubicación exacta.	N/A	Se concluye el artículo asegurando el funcionamiento del algoritmo al 95 %.
Chow y Ho (2018) [18]	Cuatro máquinas distintas de ejercicio al aire libre	N/A	Ayudó a saber que máquinas requieren más energía y esfuerzo, y cuáles menos.
Chow et. al (2017) [19]	Parque Xihu en Tainan (al sur de Taiwán)	N/A	Se proporcionaron datos importantes para poder promover estos equipos para la actividad física entre los usuarios de los parques.
Gao et al. (2016) [5]	N/A	N/A	Haciendo los equipos más interactuables, hará que se usen mucho más estos tipos de máquinas.
Gao et al. (2017) [6]	N/A	N/A	Se especificó que aunque sean solamente prototipos podrían ayudar al desarrollo de estos tipos de máquinas de gimnasia.
Cranney et al. (2015) [20]	Parque en Maroubra, Sydney, Australia	N/A	Ayudó a confirmar el hecho de que los parques biosaludables en Australia se usan y son útiles para la salud de las personas.
Chow (2013) [21]	Parque Dongning y Parque Xihu (Taiwán)	Firmaron formularios de consentimiento todos los participantes.	Concluyen el estudio aclarando que los datos obtenidos pueden ayudar a mejorar las máquinas pero que habría que estudiar más estos tipos de máquina de gimnasia.

Tabla 2.4: Tabla resumen de las patentes estudiadas

Patente	Tecnología de sensado	Envío de datos	Tecnología IoT	...
Zheng et al. (2018) [7]	Acelerómetro y giroscopio.	Los datos se envían a una estación base mediante la tecnología zigBee.	Los datos que se obtienen, se suben a la nube mediante GPRS o 3G	...
Cai y Ren (2019) [22]	Sensor de vibración, giroscopio y sensor de proximidad para el cuerpo humano	Los datos de los sensores son analizados en una terminal y enviados a un servidor en la nube o a una tarjeta SD.	Los datos se suben a un servidor en la nube.	...
Sun et al. (2019) [8]	Módulo de detección de rostros (módulo de detección de distancia, colector de imágenes y procesador de imágenes), módulo de adquisición de datos (acelerómetro y un giroscopio) y el módulo de detección de gestos (mediante la recepción del eco de señales microondas de un radar)	Los datos se envían a una estación base mediante la tecnología de ZigBee. La estación base envía los datos a un servidor en la nube mediante GPRS o 3G	Se usa ZigBee para conectar las diferentes máquinas con la estación base. La estación base se conecta a un servidor en la nube mediante GPRS o 3G. El usuario o cliente puede ver los datos mediante un dispositivo con conexión a internet como puede ser un PC o un móvil.	...
Hu (2019) [9]	Sensor de inducción magnética con un imán o sensor fotoeléctrico con un disco codificador según el tipo de máquina	Los datos se muestran directamente en una pantalla en la máquina de forma cableada	N/A	...
Zhang et al. (2017) [23]	Acelerómetro, un giroscopio, un sensor fotoeléctrico, un sensor de inducción magnética y un sensor de proximidad.	Los datos se envían remotamente a un terminal o a un servidor en la nube.	N/A	...
Jin et al. (2018) [24]	N/A	Los datos se envían a una pantalla en la misma máquina y por Bluetooth al móvil	N/A	...
Zhang (2014) [10]	Sensor de velocidad y un temporizador	N/A	N/A	...
Shi et al. (2019) [11]	Marcadores visuales en las partes móviles. Se reconoce el movimiento de los marcadores mediante una cámara y se analizan los datos.	N/A	N/A	...

Tabla 2.5: Tabla resumen de las patentes estudiadas (continuación de la Tabla 2.4)

Patente	Interfaz de Usuario	Datos Obtenidos	Lugar de la Patente	Año de publicación
Zheng et al. (2018) [7]	El usuario o cliente puede descargar o ver los datos mediante una app con conexión a Internet	Tiempo de uso y frecuencia de uso	China	2018
Cai y Ren (2019) [22]	El cliente puede observar esos datos accediendo al servidor en la nube, con cualquier dispositivo	Tiempo de uso y frecuencia de uso	China	2019
Sun et al. (2019) [8]	El usuario o cliente puede descargar o ver los datos mediante una app con conexión a Internet o cualquier otro dispositivo con conexión.	Se obtiene el tiempo de uso y la frecuencia de uso de cada máquina. También el sexo y la edad de la persona que la usa.	China	2019
Hu (2019) [9]	Pantalla en la máquina en la que el usuario puede ver los datos	Tiempo de uso, km recorridos o energía consumida	China	2019
Zhang et al. (2017) [23]	Los datos se visualizan en una pantalla en la misma máquina y con un botón táctil se puede interactuar con el usuario.	Tiempo de uso, frecuencia de uso, energía consumida y repeticiones en la máquina.	China	2017
Jin et al. (2018) [24]	Mediante una pantalla en la misma máquina se pueden visualizar los datos y tambien mediante una app en el móvil.	Tiempo de uso, repeticiones, frecuencia de uso, calorías y otros parámetros deportivos.	China	2018
Zhang (2014) [10]	El usuario puede visualizar los datos en una pantalla e incluso iniciar sesión de su cuenta.	Calorías consumidas, tiempo de uso y velocidad	China	2014
Shi et al. (2019) [11]	N/A	Tiempo de uso, velocidad, kilómetros recorridos.	China	2019

Tabla 2.6: Tabla resumen de las aplicaciones comerciales estudiadas

Aplicación	Obtención de los datos	Datos personales necesarios	Datos obtenidos	Empresa
TGO Activate [12]	Los datos son obtenidos ingresando en la aplicación el comienzo y la finalización del ejercicio, señalando el equipo y el parque concreto que se está usando.	Nombre, Email, Edad, condición física de salud (posibles enfermedades), código postal, etnia, genero e ingresos familiares mensuales.	Energía simulada (KWh), calorías (Kcal) y tiempo de ejercicio (minutos)	The Great Outdoor Gym Company (TGO)
LPA Entrénate [16]	Los datos son obtenidos ingresando en la aplicación el comienzo y la finalización del ejercicio, señalando el equipo y el parque concreto que se está usando.	Nombre, E-mail, Peso en Kg, Altura en cm, Sexo y Frecuencia cardíaca en reposo.	Duración de entrenamiento, calorías, Parques visitados y entrenamientos realizados	Ayuntamiento de Las Palmas de Gran Canaria
Norwell Outdoor Fitness v2 [13]	Se eligen las máquinas que se han usado en un parque concreto ubicado. Pero no se inicia un tiempo, sino que simplemente se eligen las máquinas que se han usado en todo el entrenamiento	Nombre, E-mail, Altura, género, fecha de nacimiento y Nacionalidad	Porcentaje de las partes del cuerpo entrenado	Norwell Outdoor Fitness
Proludic Sport [14]	Los datos son obtenidos ingresando en la aplicación el comienzo y la finalización del ejercicio, señalando el equipo concreta que se va a usar.	Altura, Fecha de nacimiento, Peso, Sexo (Conexión con la app Salud de Apple)	Ejercicios completados y calorías	Proludic
Caloo Pulse App [15]	Se selecciona el parque en donde se va a entrenar y se selecciona un tipo de entrenamiento concreto diferenciado por niveles.	Nombre, E-mail	Ejercicio realizado y tiempo de ejercicio	Caloo Pulse

Capítulo 3

Análisis y propuestas de diseño para parques biosaludables

3.1. Tipos de equipos de gimnasia

Los equipos de gimnasia de los parques biosaludables son bastante simples y robustos. Existen equipos que trabajan el tren superior del cuerpo humano, el tren inferior y ambas a la vez. Estos equipos de gimnasia pueden dividirse en varios grupos, en función de si tienen movimiento o son fijas o de las partes del cuerpo que se ejercitan. Las máquinas de estos tipos de zonas suelen ser máquinas sin ningún tipo de regulación, en cuanto a resistencias o a pesos. Aunque es cierto que existen máquinas comerciales que si permiten modificar el peso o la resistencia de la máquina, son muy pocos parques los que presentan estos tipos de equipos.

Es muy habitual encontrar en parques biosaludables máquinas con algún mecanismo de movimiento, pero también existen otros tipos de equipos de gimnasia fijos conocidos como calistenia, que permiten realizar ejercicios físicos con el propio peso corporal. En la Figura 3.1 se puede ver un ejemplo de este tipo de equipos de gimnasia fijos.



Figura 3.1: Circuito de Calistenia. Fuente: Deportes Urbanos

A continuación, se muestran las máquinas más comunes y más habituales en los parques biosaludables según E. Hernández (2009) [25]. Se dividen según su grupo de trabajo:

3.1.1. Trabajo del Tren Superior

- **Volante:** El volante ayuda a mejorar la flexibilidad y la potencia de las articulaciones del hombro, codo y muñeca. Y consiste en girar el volante en círculos en el sentido horario y antihorario. Se puede ver en la Figura 3.2 una imagen de este tipo de máquina.
- **Autobús:** Mejora la coordinación entre el cerebro y las manos, y fortalece la musculatura de los miembros superiores del cuerpo humano. Consiste en tirar coordinadamente con las dos manos de las tiras del equipo. En la Figura 3.3 se muestra una imagen de este tipo de máquina.



Figura 3.2: Máquina Volante. Fuente: Parques Troe (2015)



Figura 3.3: Máquina Autobús. Fuente: B2Sports

- **Caballo:** Musula tanto los hombros como el pecho y mejora la flexibilidad de las articulaciones, como el hombro o codo. Consiste en un asiento donde el usuario debe sentarse y agarrar las asas con las dos manos, empujar hacia adelante hasta estirar los brazos completamente y volver a la posición inicial. Se puede ver este tipo de máquina en la Figura 3.4.
- **Cintura:** Refuerza la musculatura abdominal y lumbar, y mejora la flexibilidad de la cadera y de la columna vertical. Para usarse correctamente hay que subirse a los reposapiés, sujetarse con las dos manos a la barra y girar la cadera hacia ambos lados. Se puede observar en la Figura 3.5.



Figura 3.4: Máquina Caballo. Fuente: Parques Troe (2015)



Figura 3.5: Máquina Cintura. Fuente: Parques Troe (2015)

- **Timón:** Mejora la coordinación de los hombros y los refuerza. Consiste en agarrar la manivela de la rueda y hacerla girar como si se tratase de un timón de verdad. En la Figura 3.6 se muestra una imagen de la máquina Timón.
- **Surf:** Ejercita la columna y la cadera a la vez que mejora la función cardíaca y pulmonar. Consiste en balancear la piernas de un lado al otro sujetándose con las manos a las barras. Esta máquina se muestra en la Figura 3.7.



Figura 3.6: Máquina Timón. Fuente: Parques Troe (2015)



Figura 3.7: Máquina Surf. Fuente: Parques Troe (2015)

- **Ascensor:** Desarrolla la musculatura del pecho, la espalda y los hombros a la vez que mejora su flexibilidad. Su funcionamiento consiste en tirar hacia abajo de las agarraderas mientras se está sentado manteniendo la espalda recta. En la Figura 3.8 se muestra una imagen de esta máquina.
- **Masaje:** Relaja la cadera y la espalda, a la vez que mejora la fatiga corporal. Simplemente hay que colocar la espalda sobre la columna de masaje y realizar movimientos lentos. Se puede observar esta máquina en la Figura 3.9.



Figura 3.8: Máquina Ascensor. Fuente: Parques Troe (2015)



Figura 3.9: Máquina Masaje. Fuente: Grup Fábregas

- **Baúl:** Relaja la cadera y muscula los miembros abdominales y lumbares. Hay que sentarse en el baúl, agarrarse de los arcos y subir las piernas hasta conseguir un ángulo recto con el cuerpo. La imagen de esta máquina se muestra en la Figura 3.10.
- **Esquí de fondo:** Refuerza la musculatura lumbar y abdominal y mejora la flexibilidad y la agilidad. Su uso es muy sencillo, solamente hay que subirse a los estribos, coger los palos con las manos y mover las piernas y los brazos adelante y atrás. Se puede ver esta máquina en la Figura 3.11.



Figura 3.10: Máquina Baúl. Fuente: B2Sport



Figura 3.11: Máquina Esqui de fondo. Fuente: Grup Fábregas

- **Barras:** Tal y como se muestra en la Figura 3.12 esta máquina desarrolla la musculatura de los hombros y de los pectorales. Para su uso es necesario agarrar las barras, doblar los codos y subir y bajar sin tocar el suelo con las piernas.
- **Tumbona:** Sirve para tonificar los músculos del abdomen. Para usarlo es necesario tumbarse en el equipo con los pies bajo los pedales, las manos debajo

de la cabeza y realizar abdominales. En la Figura 3.13 se muestra un ejemplo de esta máquina.

- **Paloma:** Favorece la musculatura de los hombros y de la espalda. Simplemente hay que agarrarse al manillar y realizar flexiones de brazos. Se puede ver una imagen de este tipo de máquina en la Figura 3.14.



Figura 3.12: Máquina Barras. Fuente: Parques Troe (2015)



Figura 3.13: Máquina Tumbona. Fuente: Parques Troe (2015)

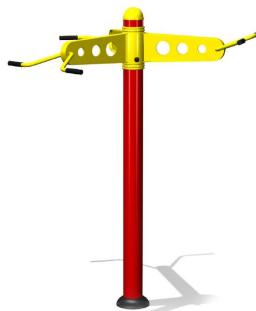


Figura 3.14: Máquina Paloma. Fuente: B2Sport

3.1.2. Trabajo del Tren Inferior

- **Patines:** Aumenta la capacidad cardiaca y pulmonar a la vez que mejora la coordinación y la movilidad de las piernas. Hay que subirse en los pedales, sujetarse a la barra y mover las piernas alternativamente. Esta máquina se muestra en la Figura 3.15.
- **Jota:** Ayuda a mejorar la flexibilidad de las partes inferiores del cuerpo. Hay que colocar la pierna en la barra e inclinarse sin doblar la rodilla. En la Figura 3.16 se muestra esta máquina.
- **Columpio:** Desarrolla la fuerza y la agilidad de las piernas. Hay que sentarse en el asiento y colocar los pies en los pedales, después solamente hay que

empujar sobre los pedales y dejarse caer posteriormente. Se puede observar esta máquina en la Figura 3.17.



Figura 3.15: Máquina Patines. Fuente: Parques Troe (2015)



Figura 3.16: Máquina Jota. Fuente: Pavimentos Infantiles



Figura 3.17: Máquina Columpio. Fuente: Parques Troe (2015)

3.1.3. Trabajo del Tren Superior e Inferior

- **Pony:** Esta máquina se puede observar en la Figura 3.18 y desarrolla la musculatura de brazos y piernas, cintura, espalda y abdomen. También ayuda a la coordinación de los cuatro miembros del cuerpo. Hay que sentarse en el asiento, colocar los pies sobre los pedales y empujar hacia delante.
- **Helicóptero:** Favorece la coordinación del tren superior y el inferior. Y ayuda a la musculación del pectoral, espalda y abdominales. Hay que agarrarse del volante, subir las piernas e intentar girar el cuerpo en ambos sentidos. Esta máquina se muestra en la Figura 3.19.



Figura 3.18: Máquina Pony. Fuente: Grup Fábregas



Figura 3.19: Máquina Helicóptero. Fuente: B2Sport

3.2. Legislación

3.2.1. UNE 16630:2015

Actualmente, todos los equipos de gimnasia exterior deben cumplir con la norma UNE 16630:2015 [26] en los siguientes países: Alemania, Antigua Republica Yugoslava de Macedonia, Austria, Bélgica, Bulgaria, Chipre, Croacia, Dinamarca, Eslovaquia, Eslovenia, España, Estonia, Finlandia, Francia, Grecia, Hungría, Irlanda, Islandia, Italia, Letonia, Lituania, Luxemburgo, Malta, Noruega, Países Bajos, Polonia, Portugal, Reino Unido, República Checa, Rumania, Suecia, Suiza y Turquía.

Esta norma establece y especifica los requisitos generales de seguridad para la fabricación, instalación, inspección y mantenimiento de estos equipos deportivos. Sin embargo, esta norma no cubre a los equipos accionados eléctricamente, las instalaciones de entrenamiento funcional o las pistas de obstáculos del estilo militar. A continuación, se resume esta norma según los aspectos más importantes a tener en cuenta.

Materiales

La gran mayoría de estos equipos deben durar años al exterior, sufriendo lluvias, bajadas de temperatura o cualquier otro fenómeno que pudiera estropear los equipos. Por ello, que se deben de usar materiales adecuados a estas características. La norma Europea establece que estos materiales deben fabricarse profesionalmente y que se debe tener especial cuidado en los revestimientos y en el desecho final del material.

Deben usarse materiales que no sean susceptibles de producir chispas con el fin de evitar incendios. Cuando se usan materiales como la madera o asociados, se debe tener en cuenta que se han de diseñar de modo que se evite la acumulación de agua y sean maderas poco susceptibles de astillamiento. Cuando se utilizan materiales metálicos se debe tener una protección frente a las condiciones atmosféricas y a la

corrosión catódica. También deben estar protegidas con revestimientos sin toxicidad y evitar que se pueda pelar o descolchar alguna capa que se produzca. Los materiales sintéticos o el caucho deben estar protegidos frente a las radiaciones ultravioletas y al oxígeno, que pueden estropear estos tipos de materiales.

No se deben utilizar materiales nocivos o peligrosos para la salud, por lo que se prohíben los siguientes materiales: amianto, plomo, formaldehido, compuestos del mercurio o aceites de alquitrán, entre otros.

Diseño

Los equipos deben estar diseñados de tal forma que el usuario pueda reconocer fácilmente su uso, que un mal gesto o uso no conlleve un daño grave para el usuario y que el ajuste de los equipos sea automático, de forma que cuando un usuario deje de usar el equipo, éste se prepare automáticamente para el inicio del siguiente usuario. Deben estar fijados a la superficie en la que se apoyan de forma permanente. No deben tener clavos o extremos de cables que sobresalgan del equipo, ni componentes con aristas o bordes cortantes. Las junturas de soldaduras deben quedar lisas y las partes salientes como roscas deben estar cubiertas permanentemente. En cuanto a las esquinas, bordes o salientes con los que pueda encontrarse el usuario al realizar el ejercicio deben estar redondeados cuando sobresalgan más de 8 mm, y el radio de curvatura mínimo debe ser de 3 mm.

En caso de que la máquina tenga una superficie donde apoyar los pies, esta superficie debe ser antideslizante y debe poder drenar líquidos. Cada apoyo debe tener una anchura mínima de 100 mm y una longitud mínima de 300 mm. Además debe tener un reborde de protección en la parte delantera y a ambos lados. Y por último, los cables metálicos que puedan presentar algún equipo debe ser de hilo galvanizado resistente a la corrosión con un diámetro comprendido entre los 25 mm y 45 mm.

Espacios y zonas

Cada equipo de gimnasia debe tener un espacio mínimo de entrenamiento. Para determinar el espacio del ejercicio se debe tener en cuenta el tipo de ejercicio que ofrece cada máquina. Las medidas mínimas del espacio de entrenamiento o de ejercitación son las mostradas en la Tabla 3.1.

Los equipos de entrenamiento deben estar alejados de las zonas de juego, como parques infantiles, o separados mediante barreras u otros obstáculos.

Tipo de uso	Radio (mm)	Altura (mm)
De pie	1000	2200
Sentado	1000	1500 (desde la altura del asiento)
Suspendido	500	300 por encima y 2200 por debajo de la posición de sujeción en suspensión.
Tumbado	1000	2200

Tabla 3.1: Medidas mínimas del espacio de ejercicio

Suelos

Los materiales que se usan para el suelo dependen de la altura de caída libre que tenga cada máquina. En la Tabla 3.2 se muestran los diferentes tipos de suelos permitidos.

Material del suelo	Profundidad de capa mínima (mm)	Altura de caída máxima (mm)
Hormigón/Piedra	-	≤ 1000
Superficies asfaltadas	-	≤ 1000
Tierra natural	-	≤ 1200
Césped	-	≤ 1500
Mantillo de corteza	200 300	≤ 2000 ≤ 3000
Viruta de Madera	200 300	≤ 2000 ≤ 3000
Arena	200 300	≤ 2000 ≤ 3000
Grava	200 300	≤ 2000 ≤ 3000

Tabla 3.2: Tipos de suelo en función de la altura de caída libre

Información en los equipos

Según la normativa, en los equipos debe mostrarse mediante un letrero informativo toda la información mínima necesaria para el correcto funcionamiento del ejercicio:

- Instrucciones sobre los ejercicios con sus imágenes correspondientes.
- Funciones principales que realiza el equipo.
- Información de seguridad.
- Peso máximo autorizado.

También deben estar marcados con la siguiente información:

- Nombre y dirección del fabricante o del vendedor.
- Referencia del equipo y año de fabricación.
- Número y fecha de esta norma europea (EN 16630)

Instalación

Para instalar las máquinas de los parques biosaludables se requiere la instalación de una cimentación previa, a la que se anclará el poste principal de la máquina con el fin de ganar estabilidad y fijación.

Cada máquina debe tener una zona de seguridad en la que no debe haber ningún otro elemento, como por ejemplo otra máquina. Estas zonas de seguridad o también llamadas zonas de movimiento varían según la caída libre que tenga cada máquina. Se establece una distancia mínima de 1,5 metros si la altura de caída libre es menor a 1,5 metros. A partir de esa altura de caída libre, la zona de movimiento se debe ampliar hasta un máximo de 2,5 metros.

3.2.2. ASTM F3101-15

La asociación ASTM (American Society for Testing and Materials) es una organización de normas internacionales que desarrolla y publica acuerdos voluntarios de normas técnicas para una amplia gama de materiales, productos, sistemas y servicios. Existe la norma ASTM F3101-15 (Standard Specification for Unsupervised Public Use Outdoor Fitness Equipment) [27], que tiene como objetivo ayudar en la fabricación, en el diseño, en la instalación e incluso al uso de los parques biosaludables para reducir posibles riesgos. Esta norma sería la equivalente a la UNE 16630:2015 pero en los territorios de Estados Unidos. A continuación, se detallan los puntos más característicos de esta norma.

Materiales y construcción

Los materiales a usar en las máquinas deben ser materiales con una durabilidad demostrada al entorno del aire libre. También se debe tener en cuenta los peligros que pueden producir los atrapamientos o enredos producidos por la degradación de los materiales por un uso prolongado o por vandalismo.

Los metales deben protegerse frente a la corrosión mediante pinturas, recubrimientos en polvo, galvanizado u otros tratamientos. Los plásticos u otros materiales que puedan degradarse con los rayos ultravioletas (UV) deben estar protegidos con algún tipo de recubrimiento u otros medios de protección. Las maderas deben ser tratadas para que no sufran la pudredumbre natural, sean resistentes a los insectos

y a la humedad, no acumulen agua y no se astillen con facilidad. Se prohíben materiales nocivos como el arseniato de cobre cromado, la creosota, el pentaclorofenol, el óxido de tributilestaño, entre otros.

Las partes móviles deberán tener cojinetes para reducir la fricción o el desgaste. En caso de que la máquina tenga cables de acero, se debe proteger los extremos de éstos para que no sean accesibles por el usuario y no deben poder aflojarse, deshilacharse o desenredarse.

Diseño

Los equipos de gimnasia de los parques biosaludables deben estar fijados al suelo y no deben poder moverse por ningún usuario. Las máquinas no deben tener ninguna parte afilada accesible al usuario y los extremos abiertos de los tubos deben tener tapones que no puedan ser quitados sin herramientas específicas.

Las máquinas que realicen movimientos pendulares deben tener topes amortiguadores para evitar el desgaste o la rotura, éstos amortiguadores no deben ser accesibles. Las empuñaduras y los reposapiés deben estar diseñados de tal forma que no puedan producir atrapamientos. En caso de las empuñaduras, deben tener un diámetro de más de 16 mm. Tanto las empuñaduras como los reposapiés deben tener algún material antideslizante.

Zonas y espacios

Se debe diferenciar la zona de entrenamiento y la zona de espacio libre. La zona de espacio libre debe ser de al menos 900 mm de ancho y no debe haber ningún objeto que pueda interferir con esta zona, aunque se puede interponer con otras zonas de espacio libre de otra máquina pero siempre sin interferir en las zonas de entrenamiento. La zona de entrenamiento la determinará el fabricante según el tipo de máquina y su movimiento.

Información en los equipos

Igual que en la norma UNE 16630:2015, esta norma establece que los equipos deben indicar o estar señalizados con toda la información necesaria para su uso, peligros u otro tipo de información que se considere importante.

3.3. Análisis DAFO

Los parques biosaludables tienen una gran cantidad de beneficios y ventajas para la salud de las personas, aunque también presenta algunos inconvenientes junto con oportunidades para estos equipos y posibles amenazas. A continuación, en las

CAPÍTULO 3. ANÁLISIS Y PROPUESTAS DE DISEÑO PARA PARQUES BIOSALUDABLES

Tablas 3.3 y 3.4, se muestra un análisis DAFO (Debilidades, Amenazas, Fortalezas y Oportunidades) sobre los equipos deportivos al aire libre y su posible monitorización.

Fortalezas	Debilidades
Favorece la socialización	Dependencia horaria (luz solar o cierre horario de las zonas)
Aumenta la movilidad de las articulaciones	Dependencia climatológica (lluvia, frío...)
Ayudan a mejorar la coordinación	Falta de monitorización (Posibilidad de mal uso)
Mejoran la flexibilidad y la resistencia	Dificultad para hacer un seguimiento del entrenamiento
Mejora la oxigenación cerebral y de los tejidos	Falta de dispositivos de entretenimiento mientras se ejercita
Mejora del proceso cardiovascular y digestivo	No hay un límite de tiempo o de personas de uso y acceso
Mejora las funciones cognitivas, sensoriales y motrices del cerebro	Posibilidad de vandalismo
El deporte al aire libre es gratuito	No tienen un mantenimiento constante
El sol mejora la producción de Vitamina D	
Existe un amplio repertorio de ejercicios	
Mejoran el equilibrio	
Ayuda a prevenir lesiones	
Para todas las edades (altura mayor de 1,4 m, según norma Europea)	

Tabla 3.3: Análisis DAFO. Fortalezas y debilidades

Oportunidades	Amenazas
Posibilidad de mayor uso por la época marcada por la pandemia del Covid, tanto sanitaria como económicaamente.	Preferencia por máquinas de gimnasios (con entretenimiento y seguimiento deportivo)
Localización geográfica (España) con buenas condiciones climatológicas	Poca investigación
Imposibilidad de seguimiento deportivo sin dispositivos externos. Oportunidad de realizar un diseño de seguimiento deportivo.	Desconocimiento de los beneficios de los parques biosaludables por parte de la población.
Desconocimiento de la cantidad de uso de las máquinas	Lluvia, frío o otras singularidades climatológicas.
Inexistencia de dispositivos para monitorizar estos equipos.	
Desconocimiento de datos deportivos consumidos por parte de los usuarios	
Monitorización de vandalismo	
Posibilidad de conocer el uso de las máquinas para posible mejor mantenimiento	
Monitorización climatológica	

Tabla 3.4: Análisis DAFO. Oportunidades y amenazas.

3.4. Conclusión del análisis

Todos los equipos de gimnasia de los parques biosaludables deben estar diseñados e instalados según las normas mencionadas anteriormente. En caso de cumplimiento de las normas, tanto la Europea como la Americana, se asegura que la máquina sea lo suficientemente robusta para que pueda soportar durante un largo periodo de tiempo las inclemencias temporales, el frecuente uso urbano e incluso el vandalismo. También, asegura un grado de protección hacia el usuario, de forma que no sea posible ningún tipo de atrapamiento, corte por objetos punzantes o daños por caídas.

Después de realizar una búsqueda exhaustiva de los equipos de gimnasia de los parques biosaludables, la gran mayoría de equipos de fitness están diseñados mediante tubos de acero galvanizado o acero inoxidable para proteger del proceso de oxidación del metal. También presentan un recubrimiento mediante pintura en polvo para protegerse de la corrosión y alargar así la vida útil de la máquina. Al ser tubos y no piezas macizas, la gran mayoría de piezas de los equipos de gimnasia exterior están huecas en su interior.

Con respecto a las piezas de plástico, se suele usar el polietileno como material principal y en cuanto a las piezas metálicas como los tornillos, suele usarse el acero inoxidable para protegerse de la corrosión.

3.5. Propuestas de diseño

Los parques biosaludables, como se vió en la tabla 3.3 y 3.4, presentan ciertas ventajas e inconvenientes. En base a ello, se proponen ciertos diseños o soluciones para aprovechar las oportunidades y cubrir las amenazas presentes.

3.5.1. Propuesta de diseño 1

Se propone la idea de cubrir la necesidad y oportunidad de poder realizar un seguimiento de la actividad deportiva de los usuarios de los parques biosaludables. Para ello, sería necesario dotar de sensores a la máquina. Los sensores pensados son los siguientes: sensores magnéticos (interruptor Reed o sensor Hall), sensor de infrarrojos o láser y sensor acelerómetro. Cada uno de estos sensores presenta alguna ventaja o inconveniente.

Los sensores magnéticos deberían estar colocados en la parte fija de la máquina y un imán en la parte móvil, de tal forma que cada vez que el imán pase por delante del sensor, éste lo detecte y se pueda obtener la velocidad de entrenamiento. Muy similar a un velocímetro usado habitualmente en las bicicletas. Se podrían obtener datos como la velocidad, la distancia o el tiempo de uso. Además, son muy resistentes frente a las vibraciones y frente a los ambientes hostiles. Las desventajas que presentan este tipo de sensores son la necesidad de un imán y que para el cálculo de la velocidad o la distancia recorrida sería necesario una configuración predeterminada en función del sitio exacto donde se coloque el sensor y del tipo de máquina.

En el caso del sensor de infrarrojos o láser debería estar colocado en una posición estratégica para que cuando la máquina se mueva, la parte móvil de la máquina corte el haz de infrarrojos y el sensor lo detecte. Usando la misma lógica que con los sensores magnéticos, se podría obtener la velocidad de entrenamiento y por tanto la distancia recorrida. Un problema, es la necesidad de que el sensor esté a la intemperie para que pueda emitir y recibir los haces de luz infrarroja.

En cuanto al acelerómetro, se debería colocar en la parte móvil de la máquina, pero no necesariamente a la intemperie, sino que podría escondérse incluso en el interior de los tubos de la máquina o en alguna carcasa, y así evitar problemas de lluvia o robo. Este sensor podría detectar la aceleración que sufre la parte móvil de la máquina, y por tanto se podría obtener la velocidad de entrenamiento y la distancia recorrida.

Estos sensores deberían estar controlados y conectados mediante un microcontrolador, a poder ser de bajo consumo, puesto que debería estar alimentado por una batería, y que exista la posibilidad de tener conexión a internet. Por ello, se propone un microcontrolador con algún tipo de conexión a Internet inalámbrica.

Una vez se obtienen los datos, éstos son subidos a la nube, para que puedan ser rescatados posteriormente. Para la empresa de la máquina o el organismo encargado de los parques biosaludables, estos datos podrían servir para saber cuánto y cómo se usan las máquinas. Para el usuario podría servir para monitorizar y seguir su entrenamiento.

Se propone usar el protocolo MQTT para que el usuario pueda visualizar su actividad deportiva en directo. La máquina funcionaría como publicador y el usuario como suscriptor. El "topic" sería un código o número característico de cada máquina. Por lo que se podrían observar los datos que van generando.

A continuación, en la Figura 3.20 se muestra un pequeño esquema básico del diseño.

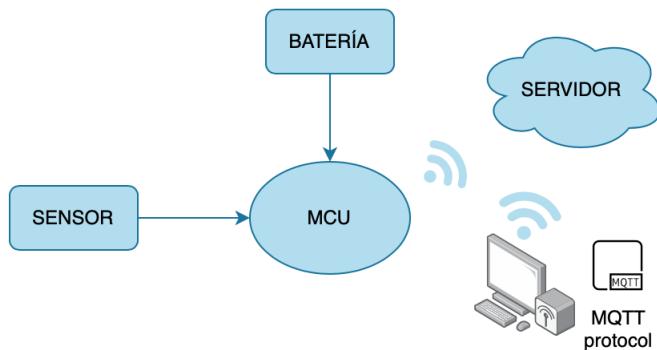


Figura 3.20: Esquema de propuesta de diseño 1. Fuente: elaboración propia.

3.5.2. Propuesta de diseño 2

Se propone algo similar al diseño anterior, en este caso, la parte de sensorización es exactamente igual en cuanto a la idea de añadir sensores para la captación de datos deportivos. En cuanto al microcontrolador a usar, también se propone uno que tenga algún tipo de conexión inalámbrica para conectarlo a Internet. Pero se añade un módulo NFC (Near Field Communications) o RFID (Radio Frequency Identification) con el que el usuario mediante una tarjeta, con el móvil o con un smartwatch se podría identificar y así hacer el ejercicio más personal. Al identificarse, el usuario

podría visualizar los datos de su actividad deportiva y realizar un seguimiento de ellos.

Para ello, el usuario debería pasar la tarjeta, móvil o smartwatch por el lector NFC antes de empezar el ejercicio. Así los datos que se generasen en el entrenamiento, se podrían asociar a una persona y ésta podría visualizarlos. Al asociarse a una persona, se podrían calcular las calorías o datos más personales, y así cada persona podría realizar un seguimiento de su actividad física. Los datos obtenidos podrían ser del estilo de la Tabla 3.5.

Hora	Tiempo	Velocidad Media	Distancia Recorrida	Máquina	Usuario
19:00	12 min	5 km/h	1 km	Elíptica	0001
19:37	7 min	4 km/h	0.46 km	Elíptica	0034
21:13	3 min	6 km/h	0.3 km	Surf	-
21:32	14 min	7 km/h	1.63 km	Elíptica	-
08:45	22 min	3 km/h	1.1 km	Elíptica	0452
10:34	35 min	5 km/h	2.91 km	Surf	0223
...

Tabla 3.5: Datos posibles a obtener

De tal forma, que cada usuario podría recoger sus datos en una aplicación y llevar un seguimiento. Asociando un peso, altura o datos personales a cada usuario, se podrían obtener datos como las calorías consumidas. Los organismos o las empresas de estas máquinas podrían también saber qué máquinas se usan más y cuales menos. A continuación, en la Figura 3.21 se muestra un pequeño esquema básico del diseño.

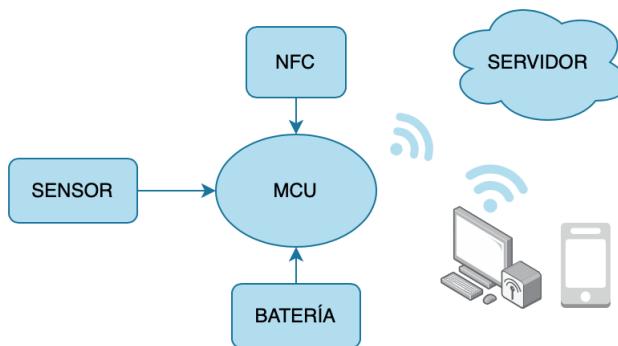


Figura 3.21: Esquema de propuesta de diseño 2. Fuente: elaboración propia.

3.5.3. Propuesta de diseño 3

Se propone un diseño con botones para que el propio usuario puede indicar cuando empieza a usarse la máquina y cuando deja de usarla. Serviría para saber cuando se inicia el entrenamiento y cuando se deja de usar. Obviamente, serviría para todo tipo de máquinas de fitness, tanto con movimiento como sin movimiento. No obtendría datos de la actividad física, sino que simplemente se sabría cuando inicia y cuando termina el usuario de entrenar.

Con respecto a los datos, se enviarían igual que en los diseños anteriores mediante Wifi o alguna otra tecnología. Los datos servirían para conocer cuánto y cuando se usa una máquina, pero no el cómo. A continuación, en la Figura 3.22 se muestra un pequeño esquema básico del diseño.

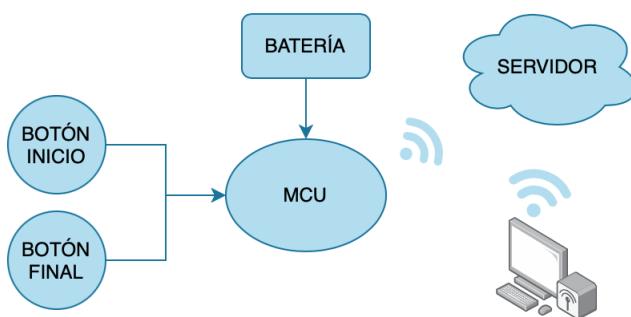


Figura 3.22: Esquema de propuesta de diseño 3. Fuente: elaboración propia.

3.5.4. Propuesta de diseño 4

Se propone sensorizar la máquina con alguno de los sensores mencionados en las propuestas anteriores, y mostrar los datos deportivos como el tiempo, la velocidad o la distancia recorrida en una pantalla. Así, el usuario podría monitorizar su entrenamiento mientras se ejercita.

Para que el usuario interactúe con la máquina se propone añadir botones con los que se pueda indicar el peso, la altura o la edad, y así poder mostrar por la pantalla datos deportivos más personales como las calorías consumidas. Aún así, datos como el tiempo de uso de cada máquina se subirían a la nube para poder saber qué máquinas son las que más se usan y las que menos.

Esta vez se propone usar una red 3G o 4G para el envío de datos, puesto que los parques biosaludables normalmente no tienen una red wifi cercana. A continuación, se muestra un pequeño esquema básico del diseño en la Figura 3.23.

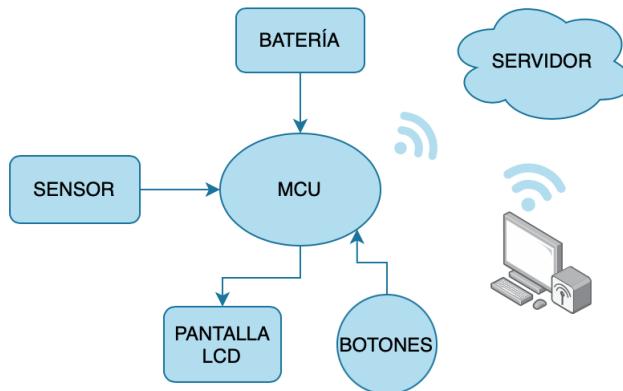


Figura 3.23: Esquema de propuesta de diseño 4. Fuente: elaboración propia.

3.5.5. Propuesta de diseño 5

Se propone un dispositivo con un sensor acelerómetro, un microcontrolador con conexión a Internet, un módulo NFC, una batería y un panel solar. Con el sensor se podrían obtener datos del movimiento de las máquinas. Podría tener una conexión Wifi o cualquier tipo de comunicación inalámbrica. El módulo NFC serviría de igual forma que en la propuesta del diseño 2, explicado anteriormente, para poder identificar a los usuarios con una tarjeta, móvil o smartwatch con tecnología NFC. Los datos se enviarían a la nube, para que posteriormente se pudieran rescatar por los usuarios.

El dispositivo estaría alimentado por un pequeño panel solar y una batería, de tal forma que mediante la luz solar se pudiera cargar la batería. Todos los elementos estarían integrados en algún tipo de carcasa o mecanismo, que se podría instalar en las máquinas en forma de abrazadera a los tubos con movimiento de las máquinas. En la Figura 3.24 se muestra un pequeño esquema básico del diseño.

3.5.6. Propuesta de diseño 6

Se propone la idea de un dispositivo central, encargado de enviar los datos a la nube, y varios dispositivos colocados en las máquinas, encargados de recolectar los datos y enviarlos al dispositivo central.

El envío de los datos desde el dispositivo central a la nube se podría realizar con cualquier tipo de tecnología inalámbrica. Para el envío de los datos recogidos de los dispositivos en las máquinas al dispositivo central, se propone el uso de Bluetooth o cualquier otra tecnología con un alcance similar. Con este diseño, se podrían monitorizar varias máquinas a la vez, sin la necesidad de instalar un módulo transmisor de datos a la nube en cada máquina. Con ello, se ahorraría tanto en coste como en consumo de batería. En la Figura 3.25 se muestra un pequeño esquema básico.

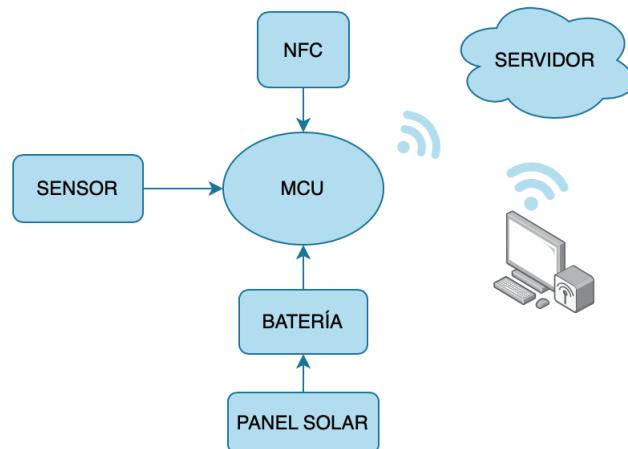


Figura 3.24: Esquema de propuesta de diseño 5. Fuente: elaboración propia.

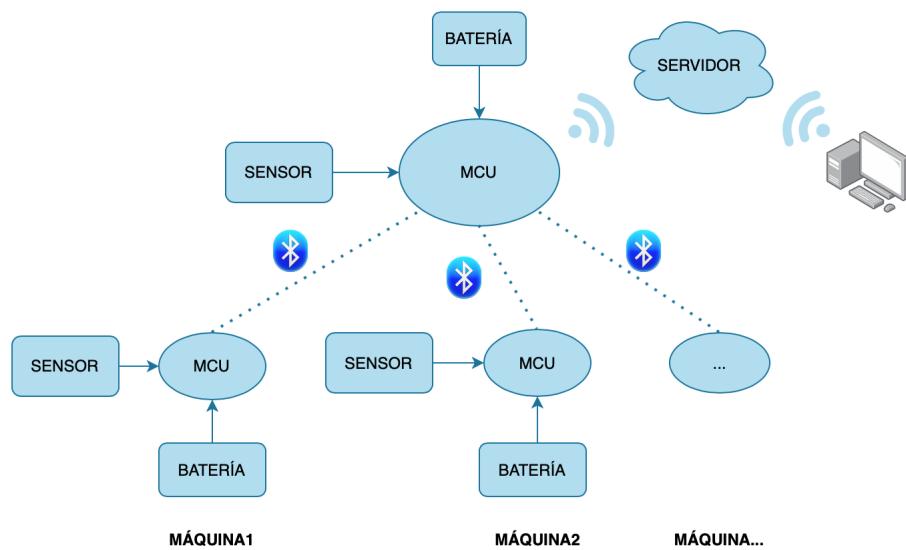


Figura 3.25: Esquema de propuesta de diseño 6. Fuente: elaboración propia.

Capítulo 4

Estudio de diseño propuesto

Según los posibles diseños pensados, se ha optado por la propuesta de diseño 6, mostrado en la Figura 3.25, con la finalidad y función de detectar cuánto se usa una máquina. Esta información sería interesante tanto para la empresa de los equipos deportivos como para el organismo que controle los parques biosaludables.

Se han descartado varios diseños, entre ellos, la opciones con la tecnología NFC, ya que ello implica a los usuarios y puede ser complejo tanto su uso como su diseño. También se ha descartado la opción de pantalla LCD en la misma máquina por los requisitos que debe cumplir este tipo de máquinas: como es la robustez y la posible existencia de vandalismo. El diseño que se escoja va a tener que instalarse en las máquinas y sufrir las consecuencias de estar a la intemperie, como es el clima y la accesibilidad libre para todas las personas. Por el vandalismo, también se descarta la opción de añadir algún tipo de placa solar capaz de alimentar al dispositivo.

Para cumplir con los requisitos de los parques biosaludables se propone la utilización de un sensor capaz de captar el movimiento y de detectar si la máquina se está usando o no. Para el control del sensor se plantea usar un microcontrolador de bajo consumo y con tecnologías de red de área personal como Bluetooth o Zigbee para poder enviar los datos a un dispositivo central, encargado de enviar los datos a la nube.

Se propone por lo tanto, dos tipos de dispositivos, el dispositivo central o maestro y los dispositivos nodos o esclavos. El dispositivo central se encargará de recoger datos de una máquina mediante un sensor, recibir los datos de los demás dispositivos y enviar todos los datos a la nube. En cambio, los dispositivos nodos se limitarán a recoger los datos mediante un sensor y enviarlos al dispositivo central, sin tener que enviar los datos a la nube. Los dispositivos deberán estar protegidos y camuflados en el entorno, ya que se expondrán a la climatología y a posibles vandalismos. Por ello, la importancia de que se protejan bien frente a polvo y líquido, y que se camuflen con la máquina de tal forma que parezca que el dispositivo y la máquina forman un conjunto y no sistemas separados.

4.1. Tipo de Sensor

Los tipos de sensores pensados para la realización de este proyecto son sensores magnéticos, sensores infrarrojos y sensores de vibración.

Los sensores magnéticos detectan campos magnéticos de imanes o electroimanes. Por ello, que en este proyecto sería necesario la instalación de un imán en las máquinas deportivas para poder detectar el movimiento de éstas. Un problema que podría existir es el hecho de que se creen interferencias magnéticas y el sensor pueda no funcionar como debería. Este fenómeno no debería ocurrir por el hecho de que los materiales comunes de este tipo de máquinas deportivas son el acero galvanizado o acero inoxidable. La necesidad de instalar un imán en la máquina puede dificultar el diseño por requerir colocar el sensor magnético en una posición determinada. Por ello, que se descarta como elección de sensor para el diseño de este proyecto.

En cuanto al sensor de infrarrojos, éste emite una señal infrarroja y en función de cómo se recepcione se puede detectar tanto la proximidad como el movimiento. Un problema lógico de este tipo de sensor es la necesidad de que la parte emisora y receptora tengan que estar a la intemperie, lo que puede ocasionar problemas. Por ello, este tipo de sensor también se descarta.

Un sensor de vibración, obviamente, detecta la vibración. Ésta es medible por dos factores: la frecuencia y la intensidad. Normalmente cuando se mide la vibración, lo que realmente se está midiendo es la aceleración, la velocidad o el desplazamiento, parámetros que pueden indicar la intensidad de la vibración. La aceleración es el parámetro más usado. Los tipos de sensores de vibración más comunes son el acelerómetro, sensor de velocidad y sensor de proximidad o desplazamiento. El acelerómetro es el sensor de vibración más usado, y por ello y por la versatilidad que tienen se ha escogido como sensor para la realización del diseño de este proyecto. Después de examinar diferentes acelerómetros comerciales se ha optado por el modelo ADXL345 de Analog Devices, que se caracteriza por ser un acelerómetro de 3 ejes con conexión I^2C y SPI, con un rango de aceleración desde los -2g hasta los 16g (siendo g la gravedad). También presenta dos pines de salida de interrupción (INT1 y INT2) que serán muy útiles para este proyecto.

4.2. Tipo de conexión a Internet

Las propuestas de usar la conexión Wifi como medio para enviar los datos a Internet no son realistas, ya que sería necesario que en todos los parques hubiera cobertura Wifi. Uno de los beneficios que tiene la conexión Wifi es la rápida tasa de datos, pero a costa de un mayor consumo. En el proyecto a diseñar, el consumo es un factor crítico a tener en cuenta, ya que en las zonas en las que se ubican este tipo de máquinas no suele haber puntos de electricidad disponibles.

Por todo ello, se propone como elección un tipo de red con cobertura bastante amplia como para poder cubrir todo un parque biosaludable y con un consumo bajo para poder realizar un diseño duradero. Este tipo de redes se conocen como LP-WAN (Low Power Wide Area Network), y se caracterizan por tener un largo alcance y una baja velocidad de transmisión de datos, que las hacen ideales para dispositivos alimentados con baterías y diseños característicos del IoT. Hay varios tipos de tecnologías fundamentadas en este tipo de redes, pero las tres más desarrolladas y accesibles son SigFox, LoRaWAN y NB-IoT.

Sigfox

Sigfox se caracteriza por ser un tipo de conectividad pensada para comunicaciones de baja velocidad y por tanto con muy bajo consumo energético. Utiliza la tecnología Ultra Narrow Band (UNB) bidireccional, transmitiendo por un canal de espectro muy estrecho y consiguiendo así, un largo alcance del orden de kilómetros y una robustez frente a interferencias.

Sigfox, según su propia información [28], tiene una velocidad de transferencia de datos de unos 100 bps con una limitación de 140 mensajes diarios de 12 bytes por dispositivo (según el tipo de suscripción que se tenga con la compañía Sigfox). Tiene también una velocidad de descarga de 600 bps con una limitación de descarga de 4 mensajes de 8 bytes al día y por dispositivo. Para poder transmitir utiliza bandas de radio ISM (Industrial, Scientific and Medical) sin licencia. En Europa se suele usar la banda de 868 MHz, con un ancho de banda de 200 kHz.

Como ya se ha mencionado, el uso de Sigfox requiere una suscripción en la compañía, y por tanto se convierte en un servicio de pago. Una de las ventajas de Sigfox es que la red de backend (servidor de aplicación, acceso a datos) viene incluido en la suscripción, facilitando la gestión de éstos.

LoRa

LoRa (Long Range) es la tecnología que se encarga de la modulación que se usa en las redes con protocolo LoRaWAN. LoRa utiliza un tipo de modulación en radiofrecuencia patentado por la empresa Semtech. Este tipo de modulación se conoce como Chirp Spread Spectrum (CSS) y tiene las ventajas de poder alcanzar largas distancias de comunicación y tener una gran robustez frente al ruido o a las interferencias. La velocidad de datos de LoRa es variable y depende del ancho de banda utilizado y del factor de dispersión (Spreading Factor, SF). Según las regiones en las que se use, este ancho de banda puede ser del orden de 125 KHz, 250 KHz o 500 KHz y el factor de dispersión lo elige el dispositivo final, facilitando así la elección de la velocidad de transmisión de los datos. Aunque la velocidad máxima será de 50 kbps. Este tipo de tecnología es bidireccional y abierta, es decir permite transmitir y recibir datos. LoRa utiliza, igual que Sigfox, bandas de radio ISM sin necesidad de licencia. En Europa, la frecuencia que se suele usar es la de 868 MHz.

NB-IoT

La tecnología NB-IoT (Narrowband-IOT) es una de las iniciativas de la organización 3GPP. Este tipo de tecnología usa una modulación OFDM (Orthogonal Frequency Division Multiplexing). Usa las bandas celulares de comunicación y por tanto usa bandas licenciadas, es necesario un suministrador u operador. Una de las características más importantes que tiene NB-IoT es la velocidad de transmisión de 200 kbps con mensajes de máximo 1600 bytes.

En la Tabla 4.1, realizada por Kais Mekki et al. (2019) [29] se puede ver una comparativa de estas tres tecnologías.

En síntesis, las tres tecnologías podrían ser útiles para la implementación de este trabajo. Aunque es cierto que el poco despliegue que tiene NB-IoT la descarta para el desarrollo de este proyecto. Una desventaja que presenta Sigfox frente a LoRaWAN, es la limitación de mensajes durante el día y el bajo tamaño del mensaje permitido, ya que es posible que para el caso estudiado sea necesario el envío de una cantidad de mensajes o datos superiores. Por ello, por la baja velocidad de datos y la necesidad de una suscripción que presenta Sigfox, la tecnología que se va a usar en adelante será LoRaWAN.

4.3. Servidor de Red

La tecnología de LoRaWAN requiere de puertas de enlace entre el dispositivo e Internet. Estas puertas de enlace se conocen como gateways. La empresa conocida como The Things Network (TTN) es una iniciativa para establecer una red global de Internet of Things. Esta iniciativa consiste en globalizar miles de gateways con el fin de poder conectar nodos con la tecnología de LoRa de forma gratuita a internet, en concreto a sus servidores, en los que se puede enlazar a diferentes servidores de aplicación con los que poder visualizar los datos. Por ello y por la amplia cobertura que presentan en todo el mundo, se propone su uso.

4.4. Tipo de conexión entre nodos y dispositivo central

Para poder conectar todos los nodos con el dispositivo encargado de enviar los datos a Internet, es necesario un tipo de conexión de red con un alcance desde los 10 a los 100 metros. Se necesita este alcance por las necesidades de ubicación de este tipo de máquinas, las cuales pueden estar separadas entre ellas por varios metros. Este tipo de red se conoce como PAN (Personal Area Network) y se caracteriza por ser un tipo de conexión entre dispositivos cercanos y de uso personal. Las dos tecnologías más usadas son ZigBee y Bluetooth de bajo consumo (Bluetooth Low Energy, BLE).

	SIGFOX	LORAWAN	NB-IOT
Modulación	BPSK	CSS	QPSK
Frecuencia	Bandas libres ISM (433 MHz en Asia, 868 MHz en Europa y 915 MHz en Norte América)	Bandas libres ISM (433 MHz en Asia, 868 MHz en Europa y 915 MHz en Norte América)	Banda de frecuencia protegida LTE
Ancho de banda	100 Hz	125 kHz, 250 kHz y 500 kHz	200 kHz
Velocidad de datos máxima	100 bps	50 kbps	200 kbps
Bidireccional	Limitado/Half-Duplex	Sí/Half-Duplex	Sí/Half-Duplex
Máximos mensajes/día	140 (UL), 4 (DL)	Ilimitado	Ilimitado
Longitud máxima de carga	12 bytes (UL), 8 bytes (DL)	243 bytes	1600 bytes
Alcance	10 km (urbano), 40 km (rural)	5 km (urbano), 20 km (rural)	1 km (urbano), 10 km (rural)
Inmunidad a Interferencias	Muy alta	Muy alta	Baja
Autenticación y Encriptación	No soportado	Sí (AES 128b)	Sí (Encriptación LTE)
Tasa de datos adaptable	No	Sí	No
Proceso de transferencia	Los dispositivos finales no se unen a una sola estación base	Los dispositivos finales no se unen a una sola estación base	Los dispositivos finales se unen a una sola estación
Localización	Sí (RSSI)	Sí (TDOA)	No (Bajo especificación)
Permite red privada	No	Sí	No
Estandarización	La empresa Sigfox en colaboración con ETSI	LoRa-Alliance	3GPP

Tabla 4.1: Comparativa entre Sigfox, LoRaWAN y NB-IoT. Kais Mekki et al. (2019) [29]

Bluetooth Low Energy, BLE

El Bluetooth Low Energy es un tipo de red inalámbrica de área personal diseñada por Bluetooth Special Interest Group (Bluetooth SIG) de muy bajo consumo. Ope-

ra en la banda de frecuencia libre de ISM de 2.4 GHz pudiendo usar 40 canales de 2MHz y con múltiples niveles de potencia que van desde 1 mW hasta los 100 mW. La velocidad de los datos puede ir desde los 125 kbps hasta los 2 Mbps, teniendo un tiempo mínimo de envío de datos de 3 ms. Utiliza una modulación GFSK (Modulación por desplazamiento de frecuencia Gaussiana). El alcance de este tipo de tecnología puede variar en función de su configuración, pudiendo alcanzar los 100 metros.

Zigbee

Zigbee es un tipo tecnología de red basado en la especificación IEEE 802.15.4 diseñado para transmitir pequeños paquetes de datos con distancias cortas y con un bajo consumo de energía. Se caracteriza por su uso de tipología en malla, lo que permite que todos los nodos funcionen también como repetidores. Utiliza la banda de frecuencia ISM de 2.4 GHz para todo el mundo, con una velocidad de transmisión de los datos de hasta 250 kbps. Puede alcanzar un máximo de unos 65536 nodos conectados en una misma red Zigbee. Presenta un rango de comunicación de hasta los 300 metros en el exterior. En la Tabla 4.2 recogida de Lin et al. (2015) [30] se compara la tecnología Zigbee y BLE.

	Bluetooth Energy	Low	Zigbee
Estandar IEEE	-		IEEE 802.15.4-2003
Banda de frecuencia	2.4 GHz		868/915 MHz; 2.4 GHz.
Máxima velocidad de datos	1 Mbps		250 kbps
Alcance	> 50 m		10-100 m
Potencia nominal TX	0 dBm		-25 a 0 dBm
Número de canales RF	79		25 (16 en 2.4 GHz)
Banda ancha del canal	2 MHz		0.3/0.6; 2 MHz
Modulación	GFSK		O-QPSK
Propagación	FHSS		DSSS
Celda básica	Piconet		Estrella
Extensiones de las celdas básicas	-		Cluster Tree, mesh
Máximo numero de nodos	> 65000		> 65000
Protección de datos	16-bit CRC		16-bit CRC
Conectividad	Soporta dispositivos con Bluetooth V4.0+		Dispositivos dedicados
Inmunidad frente a interferencias	Esquema de salto de frecuencia adaptativa		Selección de canal dinámico
Diseño MAC	Principalmente TDMA		Flexible

Tabla 4.2: Comparación de Zigbee y BLE

La tabla anterior se refiere a la versión de BLE V4.2. Es cierto que la nueva versión V5.0 tiene un aumento de velocidad de transmisión de datos que alcanza los 2 Mbps.

Uno de los factores más importantes a tener en cuenta es el coste y el consumo de estas dos tecnologías. El consumo es similar, aunque es cierto que la tecnología Bluetooth Low Energy consume por norma general un poco menos que Zigbee y por ello se alarga su vida útil. En cuanto, al coste, la tecnología BLE está mucho más implementada en el mercado, y por ello los precios son muy bajos en comparación con Zigbee.

También se ha estudiado la idea de usar Radiofrecuencia (RF) básica, sin protocolos como Bluetooth o Zigbee. Ello, implicaría usar un módulo específico de radiofrecuencia y muchas más líneas de código de programación y tiempo para asegurar la conexión entre los dispositivos. En cuanto al precio y la disponibilidad de usar esta tecnología, hay menos productos en el mercado que Bluetooth, aunque los precios son similares. Por todo ello, se propone para el diseño el uso de BLE como tecnología para conectar los nodos al dispositivo central o maestro, descartando por lo tanto a Zigbee y al uso de radiofrecuencia básica sin protocolo.

4.5. Microcontrolador para dispositivo maestro

Para intentar minimizar el tamaño del dispositivo, se ha optado por la elección de un módulo en el que ya venga implementado tanto el microcontrolador como la tecnología de transmisión BLE o la tecnología de transmisión LoRa. Así se ahorraría tanto tamaño como algunas prestaciones con respecto a añadir varios módulos externos. Uno de los factores más importantes para la elección entre una placa u otra es el consumo de cada una. Interesa un muy bajo consumo para hacer durar al máximo y alargar la vida útil de la batería. Se ha optado por alimentar el dispositivo mediante una batería, por ello que el módulo a escoger debe dar la posibilidad de conexión a este tipo de alimentación.

Estos dos puntos y la posibilidad de conexión I^2C o SPI para poder conectar el acelerómetro son los tres parámetros más importantes en la elección de un dispositivo. Realizando una búsqueda de dispositivos comerciales, se han encontrado dos opciones. La primera opción es un módulo que tenga integrado tanto el microcontrolador como el Bluetooth como la tecnología LoRa. La otra opción es un módulo integrado por un microcontrolador con Bluetooth, junto con otro módulo externo con tecnología LoRa. Los posibles módulos encontrados son los mostrados en la Tabla 4.3.

El consumo de la transmisión LoRa es la parte que más consume de la placa. Aunque para el proyecto no implicará un alto consumo ya que estará activo muy

poco tiempo, solamente cuando se tenga algún dato que enviar, estando en resto del tiempo siempre apagado o en modo ultra bajo consumo. El consumo medio de estos módulos es de unos 120 mA transmitiendo a máxima potencia, reduciéndose hasta los 30-50 mA transmitiendo a menor potencia. De igual forma ocurre con la tecnología Bluetooth Low Energy, el consumo medio de todos los módulos comerciales rondan los 5 - 15 mA transmitiendo o recibiendo datos. Pero el tiempo que estará activo este módulo también será muy corto, estando la mayoría del tiempo en modo bajo consumo.

Módulo (MCU + BLE + LoRa)					
Modelo	Marca	Consumo TX/RX BLE	Consumo TX/RX LoRa	Modo Ultra Bajo Consumo	Precio
RAK4600	RAK	9 mA / 11,5 mA	125 mA / 17 mA	11,2 uA	9,20 €
Módulo (MCU + BLE)					
Modelo	Marca	Consumo BLE TX / RX	Modo Ultra Bajo Consumo	Memoria Flash	Precio
BCM-32F7611-G01	Holtek	9 mA / 15,5 mA	Modo Sleep: 7 mA; Modo Deep-Sleep: 35 uA	64 KB	4,09 €
MAX32666	Maxim Integrated	4,3 mA / 3,2 mA	Modo Sleep: 2,1 mA; Modo Deep-Sleep: 10 uA	1 MB	5,99 €
ESP32-WROOM-32E	Espressif	130 mA / 100 mA	Modo Sleep: 30 mA; Modo Deep-Sleep: 150 uA; Modo Hibernación: 5 uA	4 MB	2,42 €
CYBT-413034-02	Cypress	5,6 mA / 5,9 mA	Modo Deep-Sleep: 1,6 uA	1 MB	9,97 €
MDBT42Q-512KV2	Raytac	16,6 mA / 11,7 mA	Modo Sleep: 1,2 uA	512 KB	8,95 €
RN4020	Microchip	16 mA / 16 mA	Modo Sleep: 5 uA; Modo Deep-Sleep: 0,9 uA	64 KB	8,87 €
Módulo LORA					
Modelo	Marca	Consumo LO-RRA TX / RX	Modo Ultra Bajo Consumo	Interfaz	Precio
RFM95W	Hoperf	120 mA / 12,1 mA	Modo Sleep: 1 uA; Modo idle: 1,5 uA; Modo Standby: 1,8 mA	SPI	19,95 €
Eric-Lora	LPRS	125 mA / 11,2 mA	-	UART	19,95 €
RFM69HCW	Hoperf	130 mA / 16 mA	Modo Sleep: 1 uA; Modo Idle: 1,2 uA; Modo Standby: 1,5 mA	SPI	9,95 €

Tabla 4.3: Diferentes módulos en el mercado

En cuanto a los microcontroladores todos los encontrados tienen modos de configuración en los que se puede hacer reposar o dormir el dispositivo consumiendo lo mínimo posible. Para ello, será importante la elección del diseño de software, el cual ayudará a que se consuma más o menos energía.

Después de analizar las características, el consumo, la disponibilidad y el precio de los módulos comerciales se ha optado para el dispositivo central o maestro por el módulo RAK4600 mostrado en la Figura 4.1, que se basa en el chip nRF52832 (MCU + BLE) de Nordic Semiconductor y en el módulo LoRa SX1276 de Semtech, y por tanto compone todas las tecnologías necesarias para realizar la función deseada a un coste menor con respecto a las demás opciones.



Figura 4.1: Módulo RAK4600. Fuente: RAKWireless.

4.6. Microcontrolador para dispositivo nodo

Para el dispositivo nodo, se necesita un microcontrolador que tenga una interfaz I^2C o SPI para poder controlar el sensor y la conectividad Bluetooth Low Energy, para poder enviar esos datos al dispositivo maestro. Obviamente este microcontrolador debe tener modos de ultra bajo consumo para que pueda durar el máximo tiempo posible con una batería. Se ha optado por usar el mismo módulo que en el dispositivo maestro, el RAK4600, lo cual facilitará la programación software y configuración de la conexión entre ellos. También se ha escogido por tener conexión Bluetooth de la versión 5.0V y por tanto una velocidad de transmisión que puede alcanzar los 2Mbps.

La única diferencia que habrá entre el dispositivo maestro y los nodos es como estará configurado cada uno. El maestro deberá recibir los datos y enviarlos a la

nube mediante LoRa. En cambio, los nodos tendrán desactivada esta última función para reducir su consumo, y solamente recogerán datos y los enviarán por BLE al maestro. Los dos dispositivos a nivel físico serán iguales, la única diferencia por tanto será el software.

4.7. Batería y consumo de los dispositivos

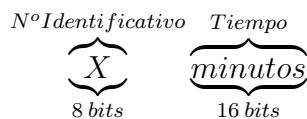
4.7.1. Consumo del dispositivo maestro

El dispositivo se colocará en una determinada posición de la máquina en la que será muy difícil el acceso a una toma de corriente. Por ello, que se deberá de alimentar mediante baterías. También por el bajo consumo de los dispositivos, las baterías son ideales para este tipo de proyectos.

La batería debe ser pequeña, duradera y debe poder conectarse fácilmente al dispositivo. Para la elección de la batería, será necesario realizar una estimación del consumo medio que tendrá nuestro dispositivo. La idea es que el microcontrolador esté todo el tiempo en un modo de bajo consumo (dormido) y cuando el acelerómetro detecte movimiento se active o despierte al microcontrolador, que tendrá que registrar los datos y volver al modo de bajo consumo. En un momento del día el dispositivo activará el módulo Bluetooth, el módulo LoRa, recibirá los datos de los demás nodos y los enviará a la nube mediante LoRa.

Para calcular el consumo medio del dispositivo son necesarios los siguientes parámetros:

- **Tamaño de los datos:** se van a registrar datos sobre el tipo de máquina que es usada (con un dígito decimal se podrían registrar hasta 10 dispositivos) y el tiempo de entrenamiento.



Por cada persona que usa la máquina se registran unos nuevos tiempos de entrenamiento. Por ello, el tamaño de datos que enviará cada nodo por persona será:

$$24\text{bits}/\text{persona} = (8\text{bits del numero identificativo} + 16\text{bits del tiempo de entrenamiento})$$

Eso será el tamaño de datos de un solo nodo por persona. El tamaño de datos real que deberá enviar el dispositivo maestro a la nube será la suma de todos

los datos de todos los nodos. O lo que es lo mismo:

$$T_{datos} = 24 \text{ bits/persona} \cdot N^o \text{Personas} \cdot N^o \text{nodos}$$

- **Consumo del acelerómetro (ADXL345):** El sensor acelerómetro estará en modo medición durante todo el tiempo, ya que tiene que estar activo para detectar movimiento en cualquier momento del día. El consumo del modelo ADXL345 es el siguiente:
 - Consumo modo medición: $23 \mu A$
 - Consumo modo espera: $0,1 \mu A$
- **Consumo del microcontrolador en modo consumo normal:** El microcontrolador estará en modo de consumo normal cuando tenga que registrar los datos. Como esto es una estimación del diseño y no se tienen datos reales, se va a suponer que el tiempo que tardará el microcontrolador en registrar los datos y volver al modo bajo consumo será de 2 segundos por persona (1 segundo para registrar el inicio del movimiento de la máquina y 1 segundo para registrar el fin del movimiento de la máquina). El consumo en modo normal del dispositivo RAK4600 es de $6 mA$.
- **Consumo del microcontrolador recibiendo datos por BLE:** El dispositivo recogerá datos durante todo el día y a una cierta hora (se han supuesto, en el cálculo, las 23:00 horas) se recibirán los datos de los nodos mediante BLE. El tiempo de envío dependerá del tamaño de los datos a enviar y de la velocidad de transmisión BLE:
 - Velocidad de transmisión BLE: $1 Mbps$.
 - Tamaño de los datos: $24 \text{ bits/persona} \cdot N^o \text{Personas} \cdot N^o \text{nodos}$. (Suponiendo que hay 8 nodos, el tamaño de los datos sería de 192 bits/persona).

$$t_{BLE} = \frac{192 \text{ bits/personas} \cdot n^o \text{ personas}}{1 Mbps}$$

El consumo del dispositivo RAK4600 recibiendo datos mediante BLE es de $11,5 mA$.

- **Consumo del microcontrolador enviando datos por LoRa:** De igual forma que en el BLE, el envío de los datos se realizará en un cierto momento del día, una vez se han recibido todos los datos de los nodos. El tiempo de envío también dependerá de la velocidad de transmisión de LoRa y del tamaño de los datos.

- Velocidad de transmisión LoRa: 50 kbps
- Tamaño de los datos: $192 \text{ bits/personas} \cdot N^{\circ} \text{Personas}$.

$$t_{LoRa} = \frac{192 \text{ bits/persona} \cdot N^{\circ} \text{Personas}}{50 \text{ kbps}}$$

- **Consumo del microcontrolador en modo bajo consumo:** El microcontrolador deberá estar en modo bajo consumo todo el tiempo excepto en los momentos en los que se registran los datos, se reciben los datos por BLE de los nodos y se envían por LoRa. El consumo en este modo para el dispositivo RAK4600 es de $11,2 \mu\text{A}$.

Para calcular el consumo medio de este dispositivo se debe seguir la siguiente fórmula:

$$I_{medio} = \frac{I_{BLE} \cdot t_{BLE} + I_{LoRa} \cdot t_{LoRa} + I_{Bajo} \cdot t_{Bajo} + I_{normal} \cdot t_{normal}}{t_{BLE} + t_{LoRa} + t_{normal} + t_{Bajo}} + I_{sensor}$$

Donde:

- I_{medio} : Consumo medio del dispositivo maestro.
- I_{BLE} : Consumo del dispositivo recibiendo datos por BLE.
- t_{BLE} : Tiempo que estará recibiendo datos el dispositivo mediante BLE (nulo excepto en cierto momento: 23:00 horas).
- I_{LoRa} : Consumo del dispositivo transmitiendo datos por LoRa.
- t_{LoRa} : Tiempo que estará el dispositivo enviando datos mediante LoRa (nulo excepto en cierto momento: 23:00 horas).
- I_{Bajo} : Consumo del dispositivo en modo bajo consumo.
- t_{Bajo} : Tiempo que estará el dispositivo en modo bajo consumo.
- I_{normal} : Consumo del dispositivo en modo consumo normal.
- t_{normal} : Tiempo que estará el dispositivo en modo consumo normal.
- I_{sensor} : Consumo del acelerómetro en modo de medición.

Para poder calcular el consumo medio es necesario saber las personas que usarán las máquinas al día. Como esto es una estimación y todavía no es el cálculo real, se han usado datos de uso de las máquinas entre semana del estudio de Chow et al.(2017) [19] tal y como se muestra en la Tabla 4.4.

Horas	Personas	Consumo medio (mA)	Horas	Personas	Consumo medio (mA)
01:00	0	0,0342	13:00	2	0,041
02:00	0	0,0342	14:00	3	0,044
03:00	0	0,0342	15:00	9	0,064
04:00	0	0,0342	16:00	14	0,08
05:00	0	0,0342	17:00	16	0,087
06:00	0	0,0342	18:00	4	0,047
07:00	20	0,100	19:00	6	0,054
08:00	25	0,117	20:00	7	0,057
09:00	16	0,087	21:00	0	0,0342
10:00	7	0,057	22:00	0	0,0342
11:00	7	0,057	23:00 ¹	0	0,121
12:00	3	0,044	24:00	0	0,0342
Consumo medio (mA)			0,057 mA		

Tabla 4.4: Estimación de consumo medio de dispositivo maestro con RAK4600

Sabiendo ya el consumo medio del dispositivo, se puede calcular de forma muy sencilla la duración que podría tener la batería. Suponiendo que se usarán pilas alcalinas no recargables del tipo AA, que otorgan 1.5 V cada una (dos pilas colocadas en serie otorgarán 3 V) y tienen una capacidad de entre 2000 y 3000 mAh, la duración de las pilas sería de:

$$T_{bateria} = \frac{C_{bat}}{I_{medio}} = \frac{2000 \text{ mAh}}{0,057 \text{ mA}} = 35087,71 \text{ horas} \approx \boxed{4 \text{ años}}$$

Usando pilas alcalinas del tipo AA con una capacidad de 2000 mAh, el dispositivo debería poder durar 4 años funcionando. Aunque es cierto que este tiempo se podría ver reducido por el ciclo de vida de las pilas y de su desgaste. Hay que aclarar que esto es una estimación previa al diseño, utilizando datos y tiempos de consumo teóricos y no reales.

4.7.2. Consumo del dispositivo nodo

El dispositivo nodo también debe ir alimentado por una batería. Ésta debe poder durar lo máximo posible. Para calcular la duración o la capacidad de la batería también se debe calcular el consumo medio que tendría el dispositivo. Para ello, se ha realizado el mismo cálculo que para el dispositivo maestro, pero obviando el parámetro de Lora y suponiendo un consumo de transmisión de datos mediante BLE para el RAK4600 de 9 mA.

¹Hora en la que se reciben los datos por BLE del nodo y se envían por LoRa.

El consumo medio se calcularía de la siguiente manera:

$$I_{medio} = \frac{t_{normal} \cdot I_{normal} + t_{bajo} \cdot I_{bajo} + t_{BLE} \cdot I_{BLE}}{t_{normal} + t_{bajo} + t_{BLE}} + I_{sensor}$$

Donde:

- I_{medio} : Consumo medio.
- t_{normal} : Tiempo que el microcontrolador está en modo consumo normal.
- I_{normal} : Consumo del microcontrolador en modo de consumo normal.
- t_{bajo} : Tiempo que el microcontrolador está en modo bajo consumo.
- I_{bajo} : Consumo del microcontrolador en modo bajo consumo.
- t_{BLE} : Tiempo que el microcontrolador estará enviando datos mediante BLE. (Será nulo en todo momento del día excepto a las 23:00 horas que es cuando se ha escogido que se envíen los datos).
- I_{BLE} : Consumo del microcontrolador transmitiendo mediante BLE.
- I_{sensor} : Consumo del acelerómetro en modo de medición.

De igual forma que el dispositivo maestro se han usado datos de uso de las máquinas entre semana del estudio de Chow et al. (2017) [19] tal y como se ve en la Tabla 4.5.

Suponiendo también que se usan pilas alcalinas del tipo AA y que el consumo medio del dispositivo nodo es de 0.053 mA, se puede calcular la duración de las pilas de igual forma que se ha hecho en el dispositivo maestro:

$$T_{Bateria} = \frac{C_{bat}}{I_{medio}} = \frac{2000 \text{ mAh}}{0,053 \text{ mA}} = 37735,85 \text{ horas} \approx \boxed{4,3 \text{ años}}$$

La duración de una pila con capacidad de 2000 mAh podría ser de un poco más de 4 años. Esta duración se podría ver reducida por el desgaste de las pilas. Como ya se indicó en el apartado de consumo del dispositivo maestro, esto es simplemente una estimación con datos teóricos y no reales.

Horas	Personas	Consumo medio (mA)	Horas	Personas	Consumo medio (mA)
01:00	0	0,0342	13:00	2	0,041
02:00	0	0,0342	14:00	3	0,044
03:00	0	0,0342	15:00	9	0,064
04:00	0	0,0342	16:00	14	0,08
05:00	0	0,0342	17:00	16	0,087
06:00	0	0,0342	18:00	4	0,047
07:00	20	0,100	19:00	6	0,054
08:00	25	0,117	20:00	7	0,057
09:00	16	0,087	21:00	0	0,034
10:00	7	0,057	22:00	0	0,034
11:00	7	0,057	23:00 ¹	0	0,0346
12:00	3	0,044	24:00	0	0,034
Consumo medio (mA)			0,053 mA		

Tabla 4.5: Estimación de consumo medio de dispositivo nodo con RAK4600

4.8. Protección de los dispositivos

Para proteger toda la circuitería de los dispositivos se introducirá toda la parte electrónica en una carcasa o envolvente con la protección adecuada al exterior. Esta carcasa deberá ser lo suficientemente robusta para sufrir posibles golpes o uso continuo. También un punto importante será la apariencia que tenga, se deberá diseñar de tal forma que el dispositivo se camuflé en la máquina, dándole colores y texturas similares a éstas para que no parezca un componente externo a la máquina y sea tan visible como para facilitar el vandalismo.

En cuanto a la protección que debe tener la carcasa, ésta debe tener un grado de protección adecuado a la posible entrada de polvo y líquido. Esta protección se mide con el grado de protección IP. Esta normativa indica dos tipos de agentes a los que ofrecer protección: el polvo y el líquido. Esta protección se suele indicar con el código IPXX, siendo la primera X el nivel de protección frente al polvo y la segunda X frente al líquido. Para el diseño estudiado se ha escogido como mínimo un grado de protección IP67, que indica que en ningún caso entrará polvo en el dispositivo y que el equipo deberá soportar la inmersión completa a un metro durante 30 minutos.

También será importante el anclaje a la máquina, éste deberá hacerse con tornillería o mecanismos poco accesibles para los usuarios, con el fin de evitar posibles robos. El anclaje deberá ser lo suficientemente robusto como para soportar el trato urbano, golpes o fuerte condiciones climatológicas como pueden ser fuertes vientos.

¹Hora en la que se envían los datos por BLE del nodo al maestro

Para el desarrollo de este diseño, se diseñará un prototipo de carcasa que aunque si tendrá la apariencia deseada no ofrecerá el grado de protección adecuado ni será lo suficientemente robusto como para poder convertirse en un producto final.

Resumiendo todos los aspectos del diseño, la Figura 4.2 muestra el esquema de la estructura final del sistema.

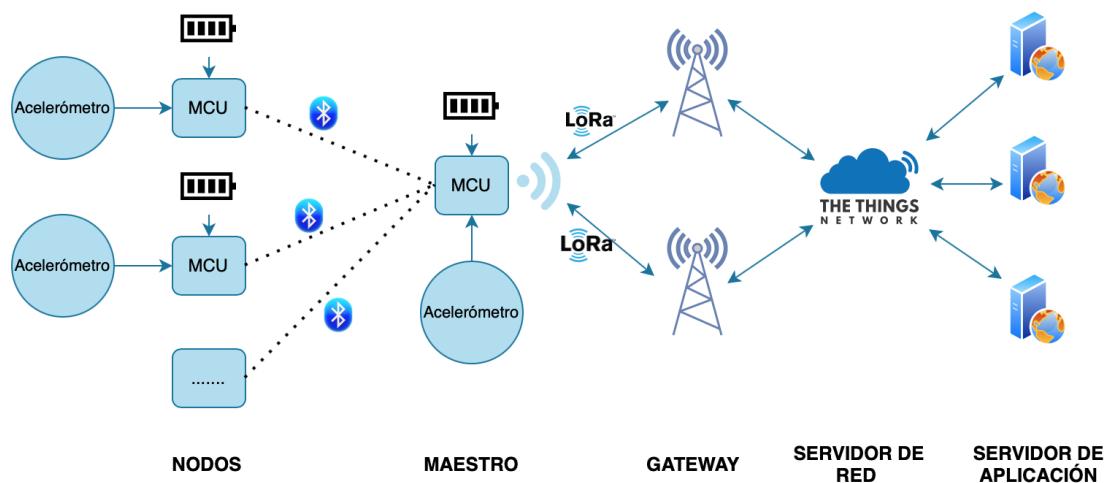


Figura 4.2: Esquema de la estructura del diseño. Fuente: elaboración propia.

Capítulo 5

Desarrollo del diseño

5.1. Diseño Electrónico

5.1.1. Listado de componentes

RAK4600

El RAK4600 es un módulo con tecnología BLE y LoRa además de contener un microcontrolador de bajo consumo. El módulo pertenece a la compañía de RAKWireless y está destinado para usarse en proyectos IoT. El microcontrolador en el que se basa el módulo es un microcontrolador de la familia de los nrf52 de la empresa Nordic Semiconductor, en concreto el modelo nrf52832. Este microcontrolador se caracteriza por permitir el protocolo BLE y tener un bajo consumo. Este microcontrolador esta diseñado en torno al modelo de procesador Cortex-M4 a 64MHz. Presenta también una memoria Flash de 512 KB de memoria junto con una RAM de 64 KB. A parte, permite las conexiones con periféricos mediante SPI, I^2C , UART, PDM o I^2S y tiene un voltaje de alimentación de entre los 1.7 a los 3.6 V. El módulo RAK4600 también contiene el chip LoRa SX1276 de Semtech. Este chip permite la comunicación mediante LoRa y el uso del protocolo LoRaWAN. El módulo, por lo tanto, se estructura de la forma mostrada en la Figura 5.1.

ADXL345

Para el desarrollo de este proyecto será imprescindible el uso de este acelerómetro. Se ha escogido el modelo ADXL345 de la empresa Analog Devices y se caracteriza por tener un ultra bajo consumo de $23 \mu A$ en modo de medición y $0,1 \mu A$ en modo standby, por tener una alta resolución de medición (13 bits) de hasta los $\pm 16g$ (siendo g la gravedad), por tener un voltaje de alimentación de 2.0V a 3.6V, por presentar conexiones mediante las interfaces digitales de SPI e I^2C y por presentar varios modos de detección de actividad o inactividad, de caída libre, de toque o doble toque.

En este caso, se usará la conexión de I^2C como medio de comunicación entre el

acelerómetro y el microcontrolador. También se usará uno de los pines (INT1), que presenta este modelo de acelerómetro, para realizar interrupciones en el microcontrolador y poder detectar la actividad e inactividad.

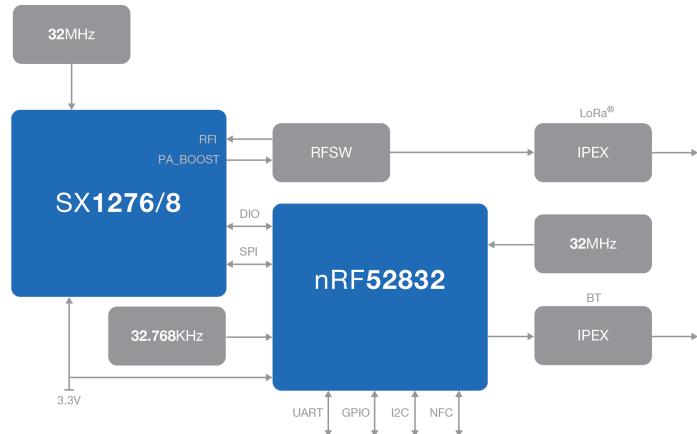


Figura 5.1: Diagrama de bloques del RAK4600. Fuente: RAKWireless.

5.1.2. Pines y conexiones

RAK4600

El modelo RAK4600 tiene 42 conexiones o pines, tal y como se puede ver en la Figura 5.2. Los pines necesarios para este proyecto serán los relacionados con la conexión I^2C , con una conexión de interrupción con el acelerómetro, que en este caso se usará un pin GPIO, las conexiones de alimentación, las conexiones de SWD para la programación y las conexiones UART para depuración.

Para la conexión I^2C son necesarios dos pines característicos de este tipo de conexión, SDA y SCL. Ambas conexiones deben tener sus respectivas resistencias de pull-up de $10\ k\Omega$, tal y como se recomienda en la hoja de datos del microcontrolador. Las conexiones de SDA y SCL corresponden a los pines 4 y 5, respectivamente del módulo RAK4600.

En cuanto a los pines del tipo GPIO, este módulo puede configurar como entrada y salida los siguientes pines: pin 4 (P0.13), pin 5 (P0.12), pin 6 (P0.09), pin 7 (P0.10), pin 9 (P0.18), pin 10 (P0.19), pin 21 (P0.14) y pin 24 (P0.17), siendo los pines entre paréntesis las conexiones con el microcontrolador nrf52832. Para este diseño se ha escogido el pin 21 para utilizarlo como entrada para la interrupción de la detección de la actividad, aunque podría haber sido cualquier otro pin mencionado anteriormente.

Para la programación del microcontrolador será necesario una conexión del tipo SWD, y para ello se usarán los pines del módulo SWDIO y SWCLK, éstos son los

pines 37 y 38, respectivamente. Para poder depurar la programación será necesaria una conexión en serie o UART, y por tanto será necesario usar los pines 22 y 23 del módulo de RAK4600, que son los equivalentes a RX y TX del modo UART.

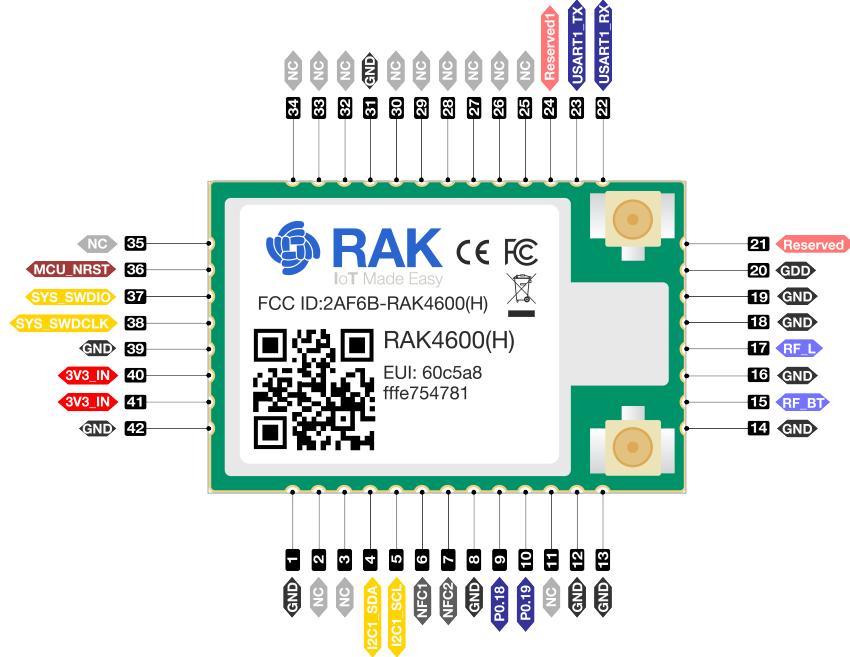


Figura 5.2: Esquema de conexiones del RAK4600. Fuente: RAKWireless.

Por último, las conexiones de alimentación del módulo se realizarán por los dos pines de entrada de 3.3 V, equivalentes a los pines 40 y 41. Estos pines irán conectados a la batería encargada de alimentar el diseño, con condensadores de desacoplamiento adecuados para filtrar la señal y suavizarla a un valor sin ruido. En la Tabla 5.1, se puede ver un resumen de las conexiones del módulo RAK4600 para este proyecto.

ADXL345

Este modelo de acelerómetro, ADXL345, presenta 14 pines tal y como se puede ver en la Figura 5.3. En este caso se usarán los pines para la conexión I^2C con el módulo RAK4600, el pin de interrupción por detección de actividad y los pines de alimentación.

Para la conexión I^2C se usarán los pines 13 y 14 del acelerómetro, que son los equivalentes a las conexiones SDA y SCL, respectivamente. Este tipo de conexión servirá para poder configurar el acelerómetro desde el microcontrolador y establecer los límites por que se considerará actividad o inactividad. Para la interrupción por detección de actividad se va a usar el pin INT1 del acelerómetro, equivalente al pin 8.

Pin	Uso o función	Pin	Uso o función
Pin 1	GND	Pin 22	RX UART
Pin 2	No se conecta	Pin 23	TX UART
Pin 3	No se conecta	Pin 24	No se conecta
Pin 4	SDA I^2C	Pin 25	No se conecta
Pin 5	SCL I^2C	Pin 26	No se conecta
Pin 6	No se conecta	Pin 27	No se conecta
Pin 7	No se conecta	Pin 28	No se conecta
Pin 8	GND	Pin 29	No se conecta
Pin 9	No se conecta	Pin 30	No se conecta
Pin 10	No se conecta	Pin 31	GND
Pin 11	No se conecta	Pin 32	No se conecta
Pin 12	GND	Pin 33	No se conecta
Pin 13	GND	Pin 34	No se conecta
Pin 14	GND	Pin 35	No se conecta
Pin 15	No se conecta	Pin 36	No se conecta
Pin 16	GND	Pin 37	SWDIO
Pin 17	No se conecta	Pin 38	SWCLK
Pin 18	GND	Pin 39	GND
Pin 19	GND	Pin 40	Entrada 3.3V
Pin 20	GND	Pin 41	Entrada 3.3V
Pin 21	GPIO	Pin 42	GND

Tabla 5.1: Conexiones del RAK4600.

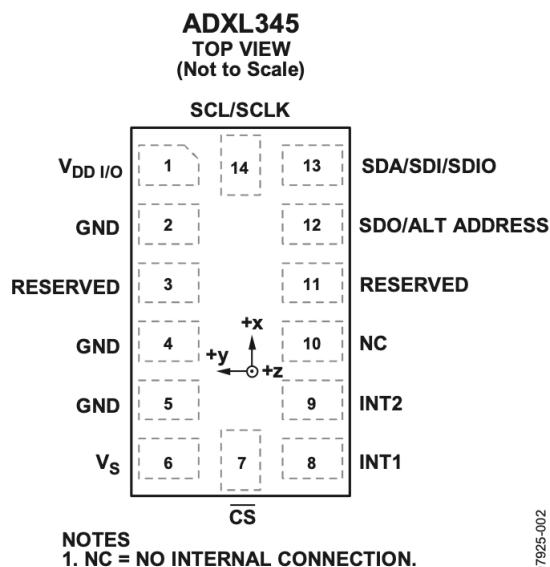


Figura 5.3: Conexiones del acelerómetro ADXL345. Fuente: Analog Devices.

5.1.3. Esquema electrónico

El esquema electrónico del diseño se muestra en la Figura 5.5 donde se pueden ver todas las conexiones entre el acelerómetro y el módulo RAK4600. El diseño es sencillo, únicamente se han agregado condensadores de desacoplo en la entrada de la alimentación tanto del módulo RAK4600 como del acelerómetro y ciertas resistencias de pull-up para la conexión I^2C .

Una vez comprobado el diseño electrónico se han montado dos prototipos con todos los componentes y funcionalidades, uno para el dispositivo maestro y otro para el dispositivo nodo. Físicamente son iguales, la única diferencia estará en el software. A continuación, en la Figura 5.4 se puede ver este prototipo.

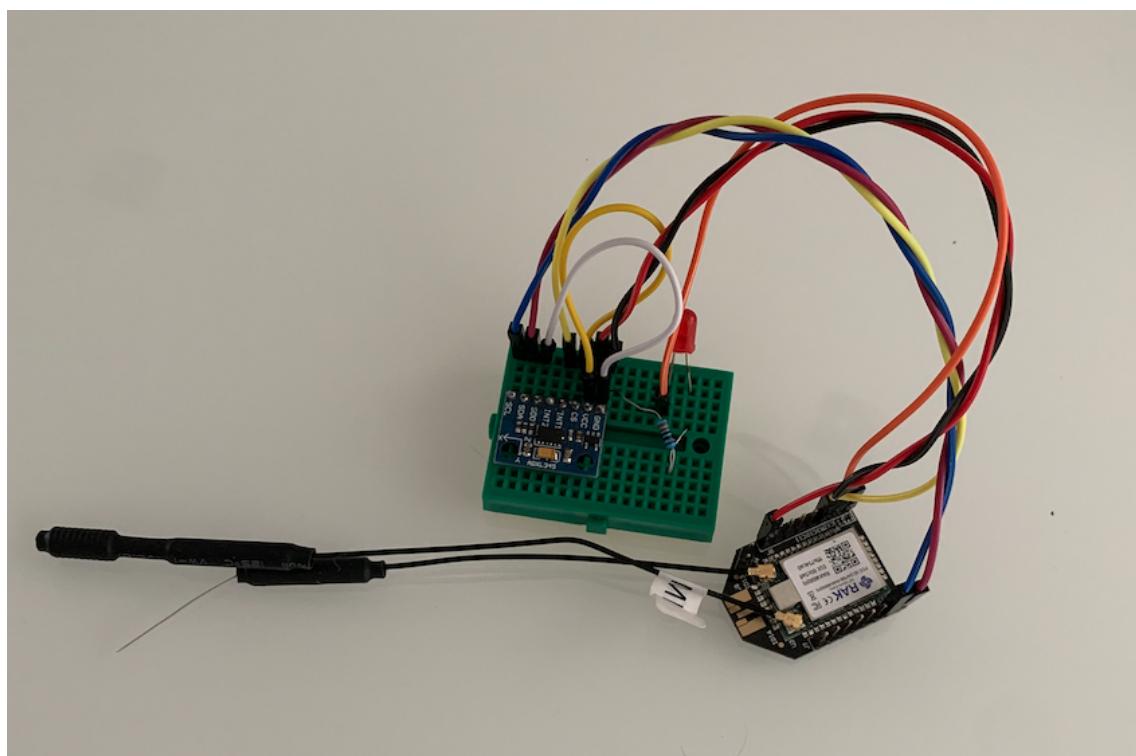


Figura 5.4: Prototipo del dispositivo. Fuente: elaboración propia.

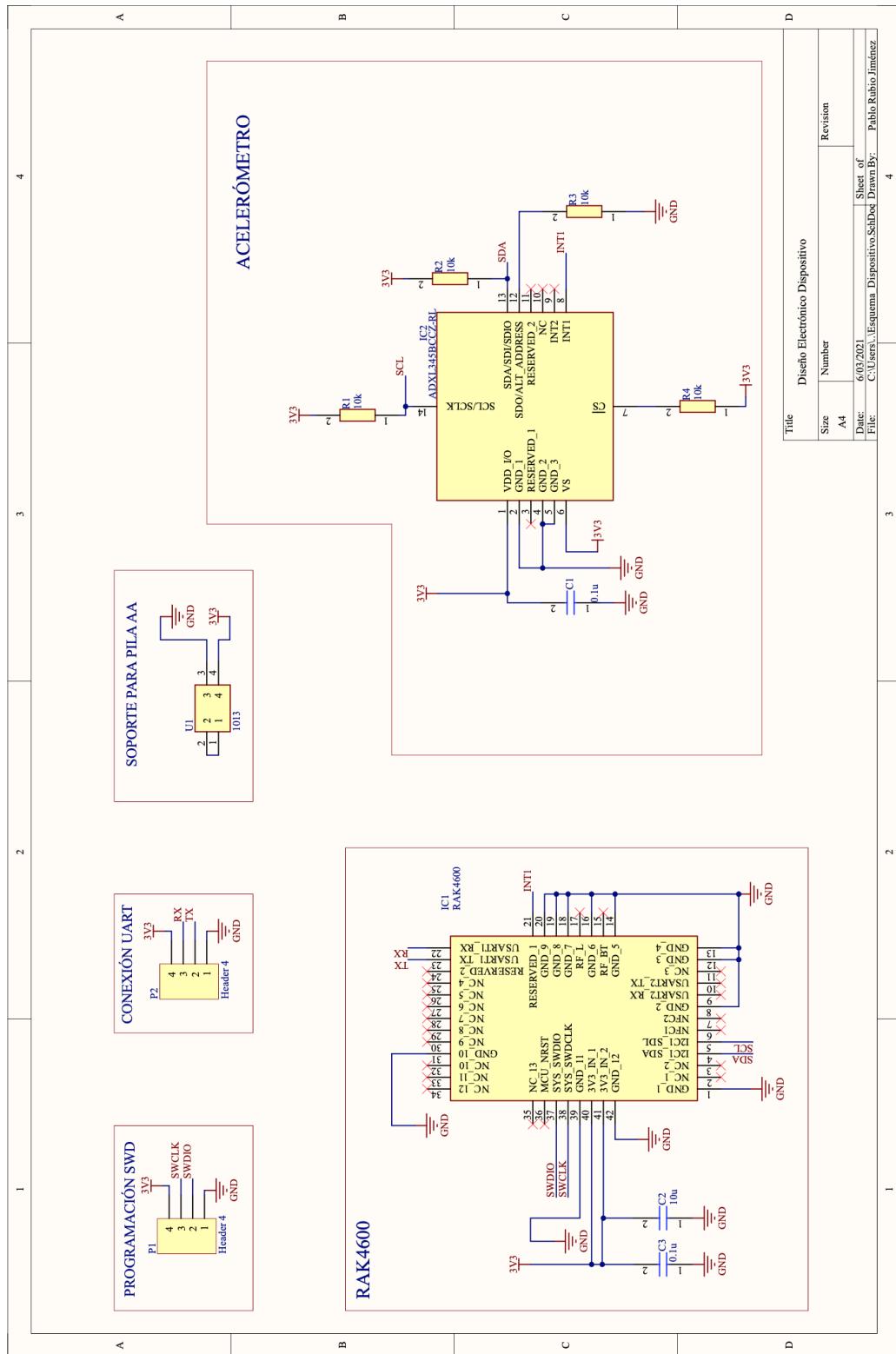


Figura 5.5: Esquema electrónico. Fuente: elaboración propia.

5.1.4. Diseño de la placa PCB

Con el fin de poder llevar el dispositivo cableado, mostrado en la Figura 5.4, a un prototipo mucho más realista se ha diseñado una placa PCB, tal y como se puede ver en las Figuras 5.6, 5.7, 5.8, 5.9 y 5.10. Esta placa compartirá diseño tanto para el dispositivo maestro como para el dispositivo nodo. La placa tiene unas dimensiones de 64 x 32,26 mm y se ha llevado a fabricar y montar físicamente, tal y como se puede ver en las Figuras 5.11, 5.12 y 5.13.

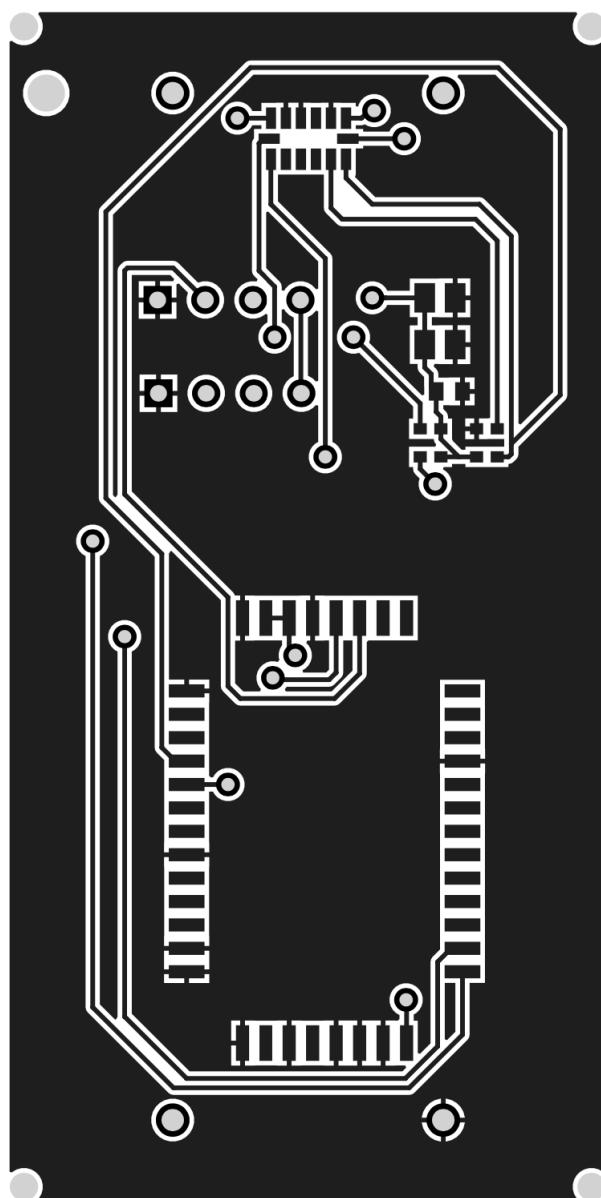


Figura 5.6: Placa PCB parte frontal. Fuente: elaboración propia.

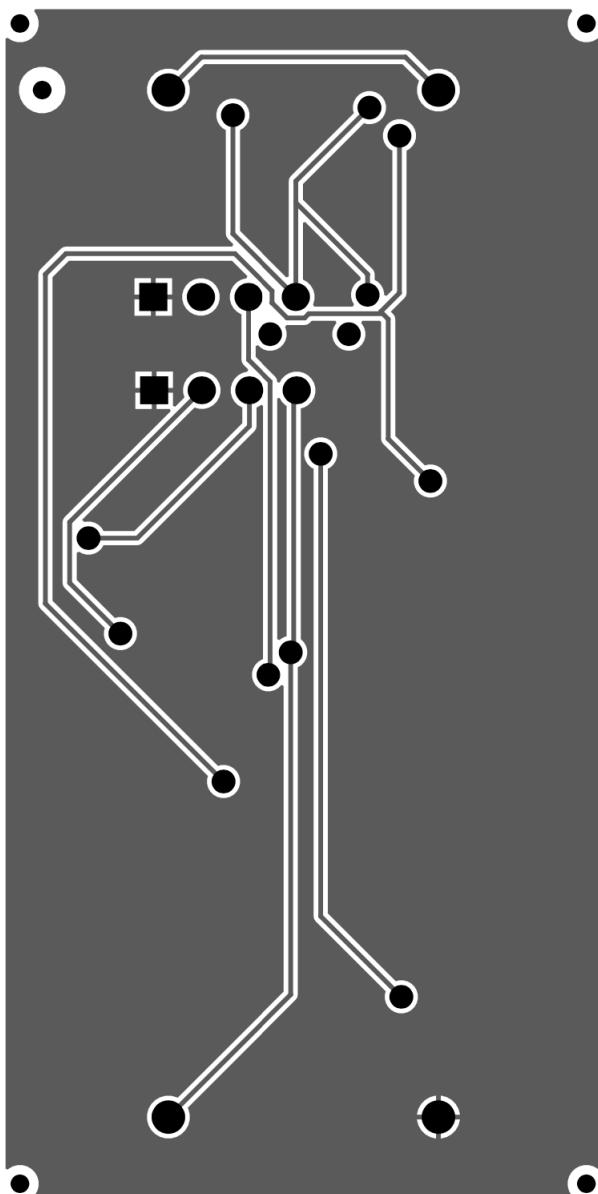


Figura 5.7: Placa PCB parte trasera. Fuente: elaboración propia.

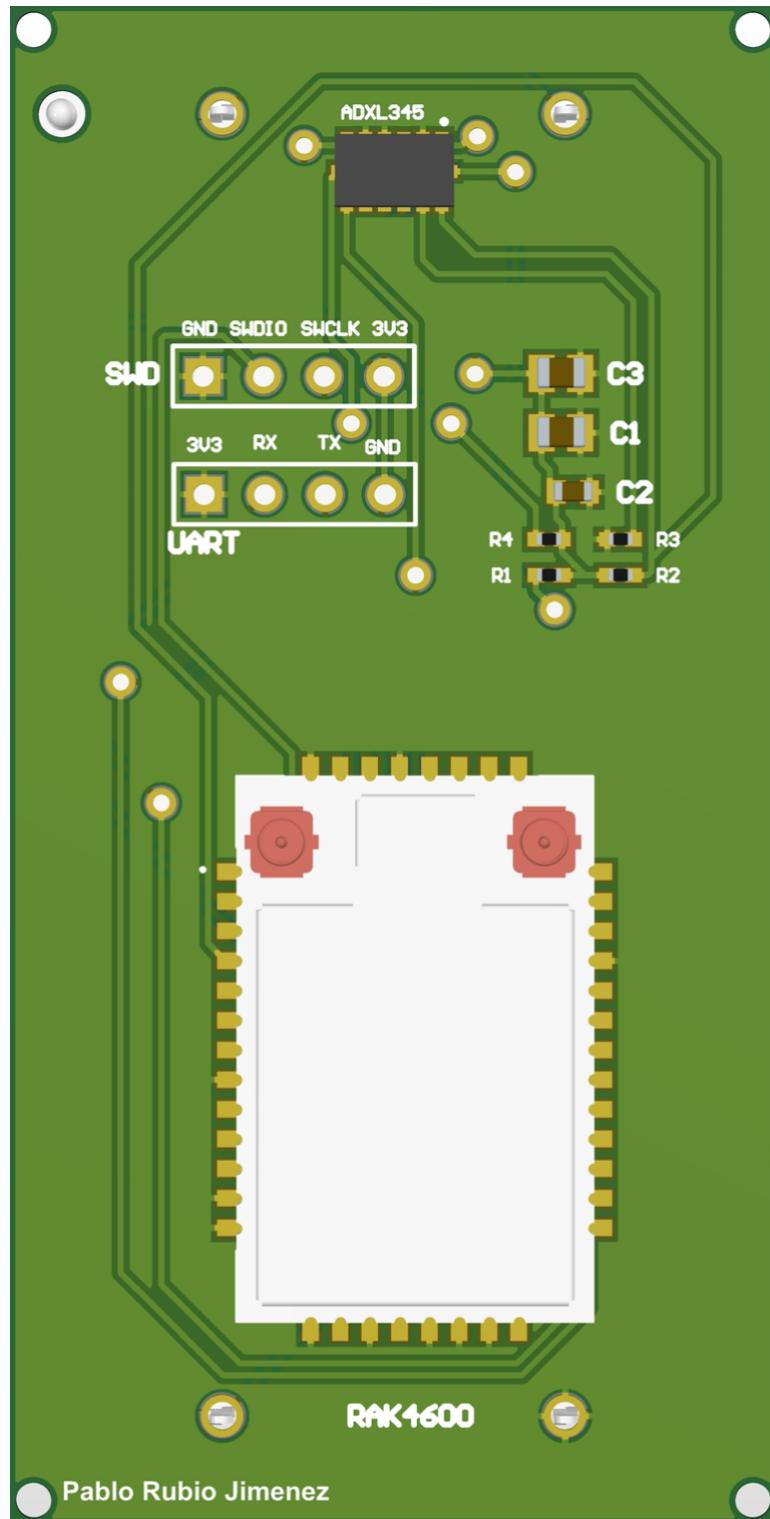


Figura 5.8: Placa 3D frontal. Fuente: elaboración propia.

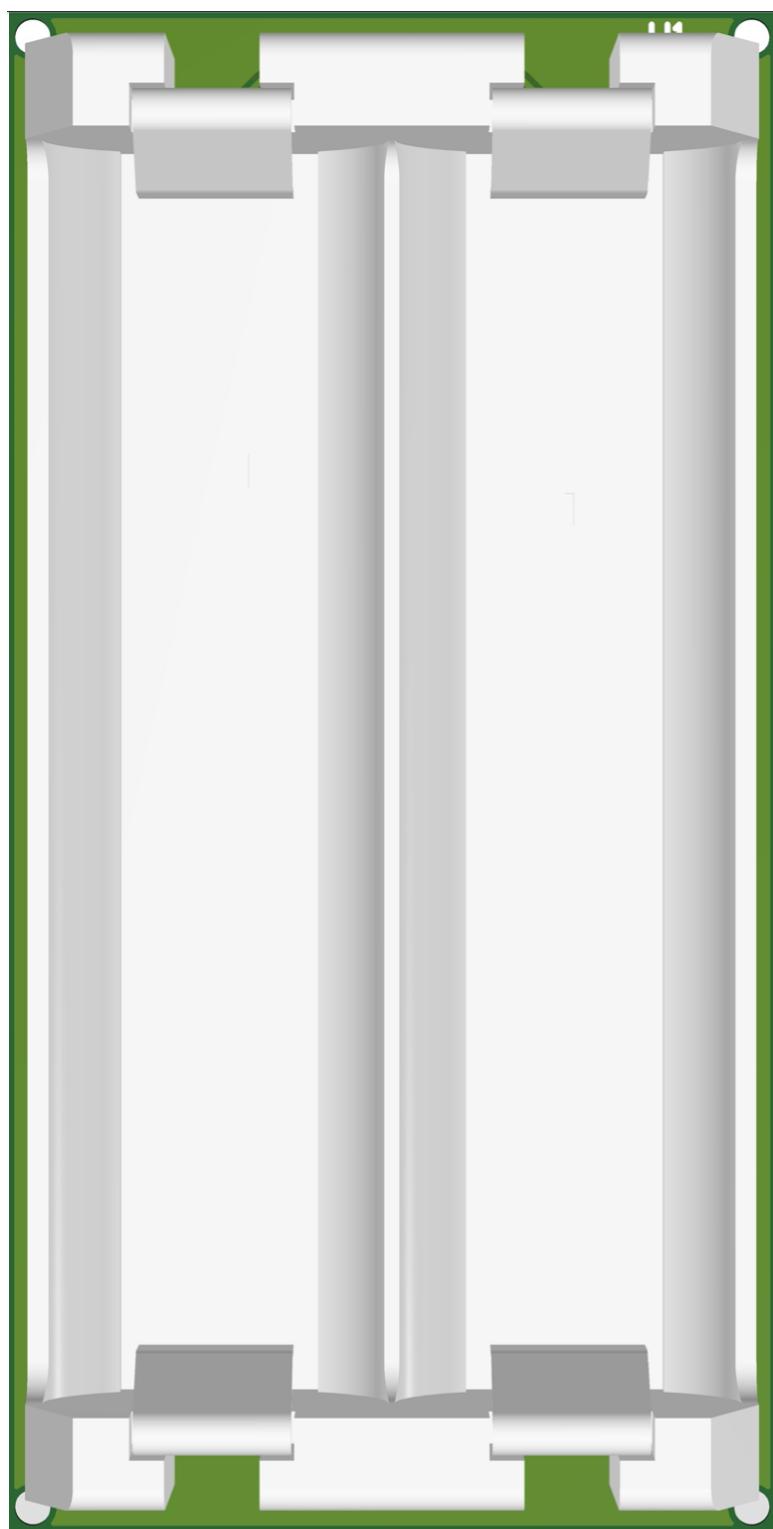


Figura 5.9: Placa 3D trasera. Fuente: elaboración propia.

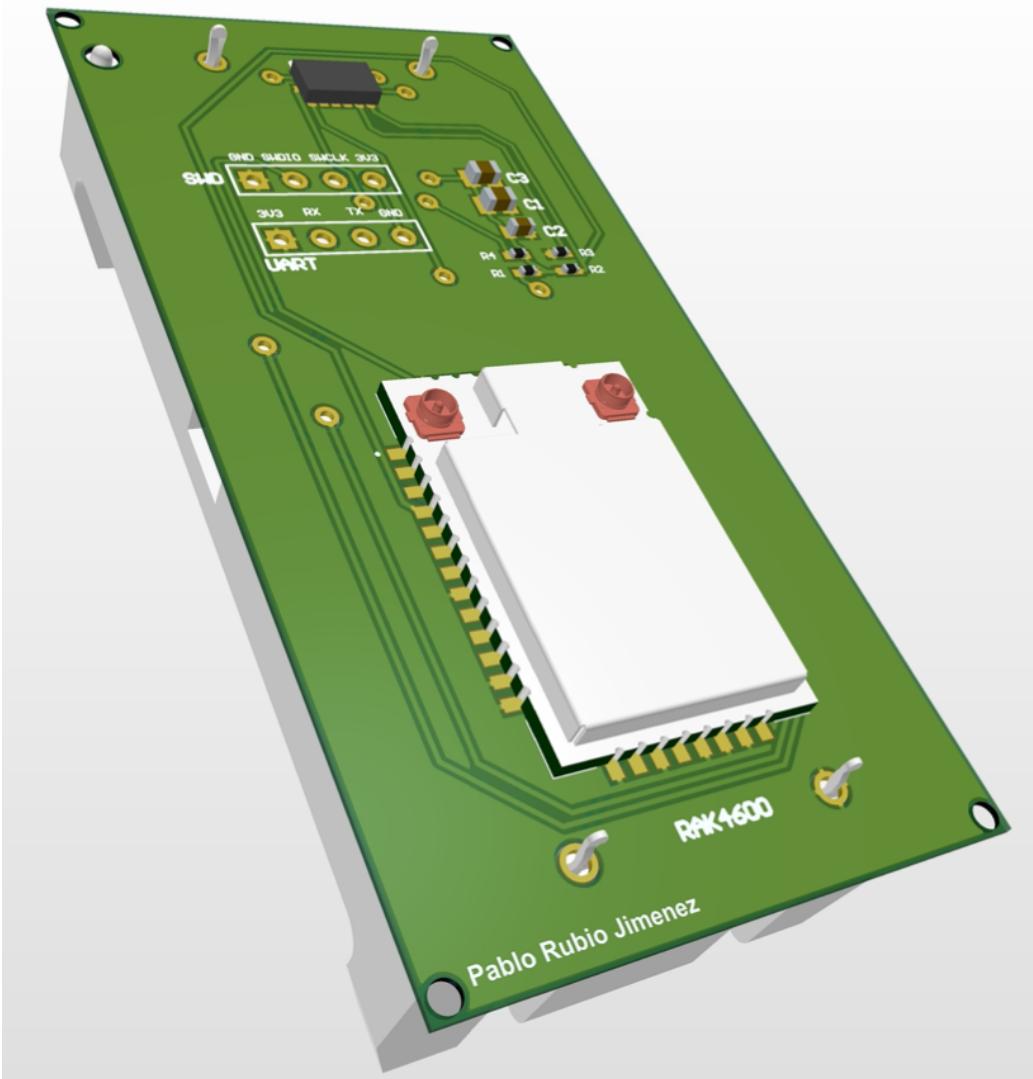


Figura 5.10: Placa 3D lateral. Fuente: elaboración propia.

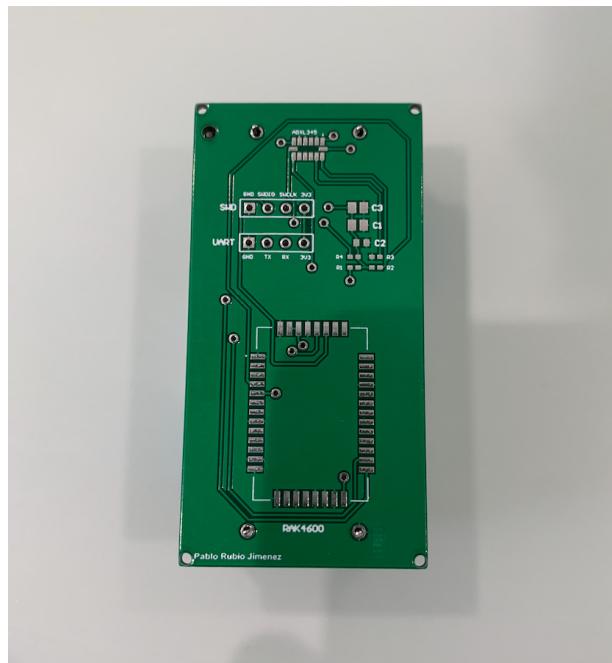


Figura 5.11: Prototipo de placa PCB fabricada sin componentes. Fuente: elaboración propia.



Figura 5.12: Prototipo de placa PCB fabricada con componentes, imagen 1. Fuente: elaboración propia.



Figura 5.13: Prototipo de placa PCB fabricada con componentes, imagen 2. Fuente: elaboración propia.

5.2. Diseño Software

5.2.1. Componentes necesarios para el Software

Entorno de programación

El módulo RAK4600 está compuesto por un módulo Lora SX1276 y un microcontrolador con Bluetooth NRF52832. La compañía RAKWireless indica que se puede programar el módulo en varios entornos o interfaces, entre ellas Arduino IDE, SEGGER Embedded Studio y RAKWireless Unified Interface (RUI). Esta última la proporciona la propia compañía de RAKWireless y hay multitud de ejemplos en los que se simplifica mucho la programación. RUI define una API con la que se facilita mucho la programación con protocolos como Bluetooth, LoRaWAN, I2C o SPI. También permite acceder a la memoria Flash y acceder al modo de bajo consumo.

En cuanto a SEGGER Embedded Studio, es el entorno de programación recomendado por la empresa del microcontrolador, Nordic Semiconductor. También presenta multitud de librerías y ejemplos con los que poder desarrollar el sistema. Esta es la opción más completa, con la que se pueden realizar más funcionalidades, pero a costa de una mayor complejidad. Arduino IDE, es un entorno muy intuitivo y sencillo, pero a cambio presenta ciertas limitaciones en cuanto a la familia de microcontroladores nrf52. El lenguaje de programación en todos los entornos mencionados anteriormente es C o C++, y todos presentan librerías específicas tanto para el bajo consumo, como para LoRaWAN, BLE e I^2C .

Para el estudio en cuestión, después de haber probado las tres opciones, se usará el IDE de Arduino por la versatilidad que presenta y por la gran cantidad de información que existe en comparación con los demás entornos. Al usar este entorno de

programación, se asumirá una limitación en cuanto al bajo consumo, pero se ganará facilidad de programación en cuanto a los protocolos de LoRaWAN y Bluetooth.

Dispositivo programador

Para programar el dispositivo es necesario un programador con conexión SWD y conexión UART para la depuración. La compañía del módulo RAK4600 propone el uso de su dispositivo programador, el RAKDAP1 mostrado en la Figura 5.14, que adapta la conexión USB a SWD.



Figura 5.14: RAKDAP1. Fuente: RAKWireless

En cuanto a la conexión UART, se usará el dispositivo comercial FT232, mostrado en la Figura 5.15, que adapta la conexión UART a USB, para así poder depurar todo el proceso.



Figura 5.15: FT232. Fuente: Amazon

5.2.2. Funcionamiento general de los dispositivos

Dispositivo nodo

La función de este dispositivo será la de detectar la actividad, registrar el dato y enviarlo al dispositivo maestro mediante Bluetooth. El dispositivo estará en modo

de bajo consumo todo el tiempo posible, y solo saldrá de éste cuando el acelerómetro detecte movimiento y se deba registrar el dato. También saldrá del modo bajo consumo cuando salte una alarma en un cierto tiempo, a modo de temporizador (timer) en el que se enviarán todos los datos recogidos al dispositivo central por Bluetooth. Pero el resto del tiempo, el dispositivo no deberá de realizar más funciones con el fin de ahorrar consumo lo máximo posible.

Dispositivo maestro

En cuanto al dispositivo central, el funcionamiento será muy parecido al nodo. La única diferencia será que cuando salte la alarma por el temporizador no se enviarán los datos por Bluetooth, sino que se recibirán todos los datos de los dispositivos nodos, y éstos se enviarán mediante el protocolo LoRaWAN.

5.2.3. Registro de los datos

El dato que se va a registrar es la duración del entrenamiento para cada persona. Y por ello, será imprescindible que el dispositivo sea capaz de detectar la actividad de la máquina, pero también la inactividad, ya que es importante saber cuando se deja de usar ésta.

Una vez el dispositivo detecte el inicio de la actividad, deberá tomar el tiempo de inicio del entrenamiento, y en cuanto se detecte la inactividad, éste tomará el tiempo de finalización de entrenamiento. Con ambos tiempos se podrá conocer la duración del mismo. Con este dato sería posible, posteriormente, saber cuantas personas han usado la máquina y cuánto tiempo ha estado entrenando cada persona. La forma en la que se registrarán los datos será mediante una cadena de texto, en la que se mostrará un identificador y posteriormente los tiempos de entrenamiento, con un separador entre ellos. (DATO: nodo;23;45;34).

5.2.4. Interrupción por Actividad e Inactividad

La interrupción por actividad o inactividad la va a generar el acelerómetro. El funcionamiento de la interrupción es sencillo. Cuando se detecta actividad, el acelerómetro genera un pulso que es el que va a interrumpir el microcontrolador. Una vez se ha detectado la actividad, no se va a detectar más actividad hasta que primero se detecte el estado de inactividad. Con ello, se consigue que el acelerómetro no esté enviando interrupciones constantemente mientras se ejercita, sino que para que detecte actividad debe venir de un estado de inactividad.

Para configurar el acelerómetro hay que escribir mediante I^2C en la dirección correspondiente y en los registros que se deseen configurar. Algunas de las configuraciones que se van a realizar son activar las interrupciones por actividad e inactividad,

seleccionar el pin INT1 como pin de salida de interrupción, elegir los límites por los que se considerará actividad o no y el modo de consumo del acelerómetro.

5.2.5. Funcionamiento del Timer

En cuanto al temporizador, éste se configurará al inicio del programa y su función será la de interrumpir la actividad para comenzar el modo de envío y recepción de los datos mediante Bluetooth. El tiempo establecido para el temporizador se determinará al inicio de la ejecución del programa. Este tiempo se podrá configurar desde cualquier dispositivo con conexión a internet y acceso a la plataforma posteriormente explicada. De tal forma, que el tiempo se enviará mediante LoRaWAN al dispositivo maestro y éste se encargará de enviar el tiempo a los nodos mediante Bluetooth.

5.2.6. Envío y recepción de datos por BLE

Los nodos, en un momento dado, deben enviar los datos recogidos al maestro mediante el protocolo de BLE, para que éste posteriormente los envíe a Internet mediante LoRaWAN. El proceso de envío se explicará más a fondo en el apartado 5.2.8, pero a modo general primero debe haber una conexión entre el nodo y el maestro y posteriormente el nodo debe enviarle los datos.

5.2.7. Envío de datos por LoRaWAN

Una vez el maestro ya tiene todos los datos debe comunicarse y enviar los datos por LoRaWAN. Para ello, será necesario un gateway que funcionará como puerta de enlace entre LoRaWAN y la red con conexión a Internet, en este caso Wifi. El Gateway usado para este trabajo ha sido el modelo comercial The Things Indoor Gateway mostrado en la Figura 5.16.

El gateway deberá ser configurado para conectarse a la página web The Things Network (TTN), que nos servirá como plataforma para gestionar la comunicación entre los dispositivos e Internet. El proceso de comunicación del dispositivo con el gateway y por tanto con la plataforma TTN de forma general es el siguiente: primero el dispositivo maestro debe enviar una petición de unión a la que la plataforma TTN, ésta deberá aceptarla y entonces se enlazarán. Una vez están enlazados, es posible enviar y recibir datos en ambos sentidos.

Para que TTN pueda reconocer el dispositivo y aceptar la unión, la petición de unión debe contener tres códigos específicos: Global Device ID, Application ID y Application Key.



Figura 5.16: The Things Indoor Gateway. Fuente: The Things Network

El Global Device ID (DevEUI) es un número específico que determina que dispositivo que se va a comunicar mediante LoRaWAN, en este caso el dispositivo maestro. El Application ID (AppEUI) es un número que identifica el servidor de aplicación que se usará, en nuestro caso es proporcionado por TTN. El Application Key (AppKey) es un código que permitirá la comunicación y el envío de datos entre los dispositivos y la plataforma TTN.

5.2.8. Protocolo

Para el diseño software se ha creado un protocolo de actuación, en el que se definen todas las posibles situaciones en el desarrollo de la actividad normal del dispositivo. A continuación, se explicará paso a paso el protocolo para cada uno de los dispositivos con diagramas de flujo, con los que se facilita mucho el entendimiento.

Protocolo dispositivo maestro

- **Recogida del tiempo de temporizador mediante LoRaWAN**

En la Figura 5.17, se puede observar que el primer paso al iniciar el dispositivo maestro es el estado de configuración. En esta parte de configuración, el dispositivo debe comunicarse mediante LoRaWAN a internet y recoger el dato del tiempo para el temporizador. Este tiempo se usará para determinar cuando se interrumpirá el proceso de recogida de datos y se recogerán los datos de los nodos para después enviarlos por LoRaWAN a Internet.

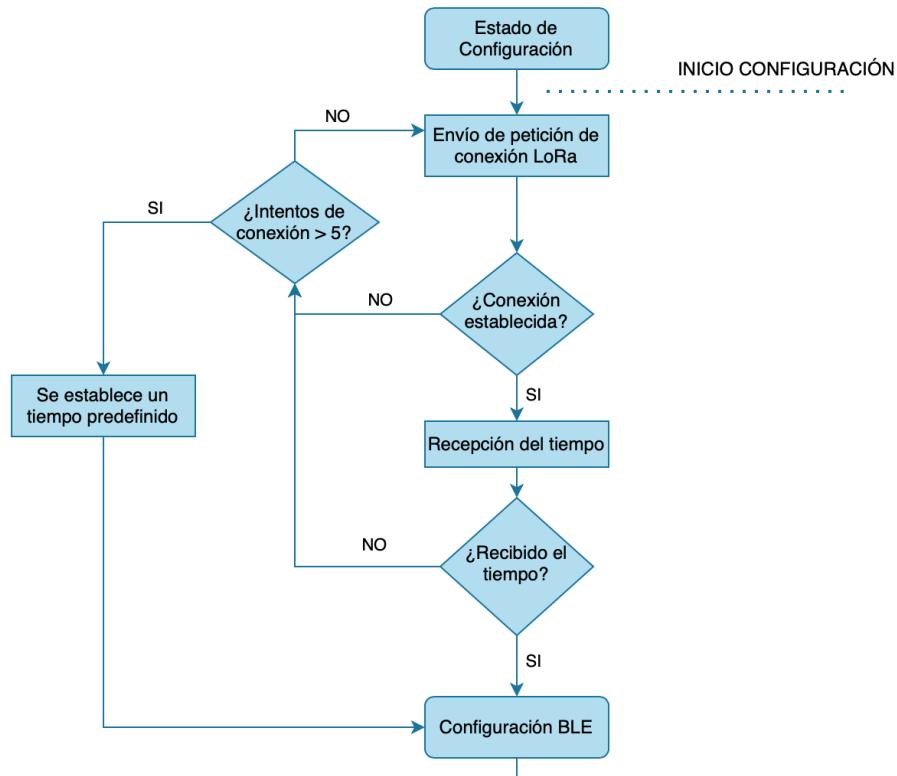


Figura 5.17: Diagrama del protocolo para el dispositivo central-1. Fuente: elaboración propia.

Una vez se inicia la configuración, el dispositivo debe enviar una petición de conexión a la plataforma TTN y ésta debe confirmar la conexión. Una vez se confirma, se establece la conexión y se permite el envío y recibo de datos. Una vez se confirma la conexión, la plataforma TTN envía el tiempo establecido al dispositivo. Éste lo recibe y procede a la configuración del Bluetooth para enviárselo a los nodos.

En caso de que por cualquier motivo no se establezca conexión con la plataforma TTN, el dispositivo maestro irá enviando peticiones cada 10 segundos, hasta un máximo de 5 intentos. Si en todos estos intentos no se ha establecido conexión, el dispositivo no vuelve a intentarlo y él mismo establece un tiempo predefinido. Si esto ocurre, una vez se establece el tiempo predefinido se procede a configurar el Bluetooth de igual forma que si se hubiera establecido conexión.

■ Envío del tiempo a los nodos mediante BLE

En esta parte, tal y como se puede ver en la Figura 5.18 se debe configurar el Bluetooth para poder enviar el tiempo para el temporizador a los nodos.

Una vez está configurado, se establece un tiempo de escaneo de 20 segundos en el que el dispositivo escaneará durante ese tiempo a todos los nodos que se muestren. En caso de que encuentre algún nodo, se conectará a él y se procederá a escanear más nodos, a los que se conectarán también.

Al pasar los 20 segundos establecidos de escaneo, el dispositivo maestro procederá a enviar el tiempo a todos los nodos encontrados y conectados. Al enviar el tiempo, el maestro se desconecta de todos los nodos y se termina el proceso de configuración para la obtención del tiempo para el temporizador y comienza el estado de funcionamiento normal en el que se registran los datos.

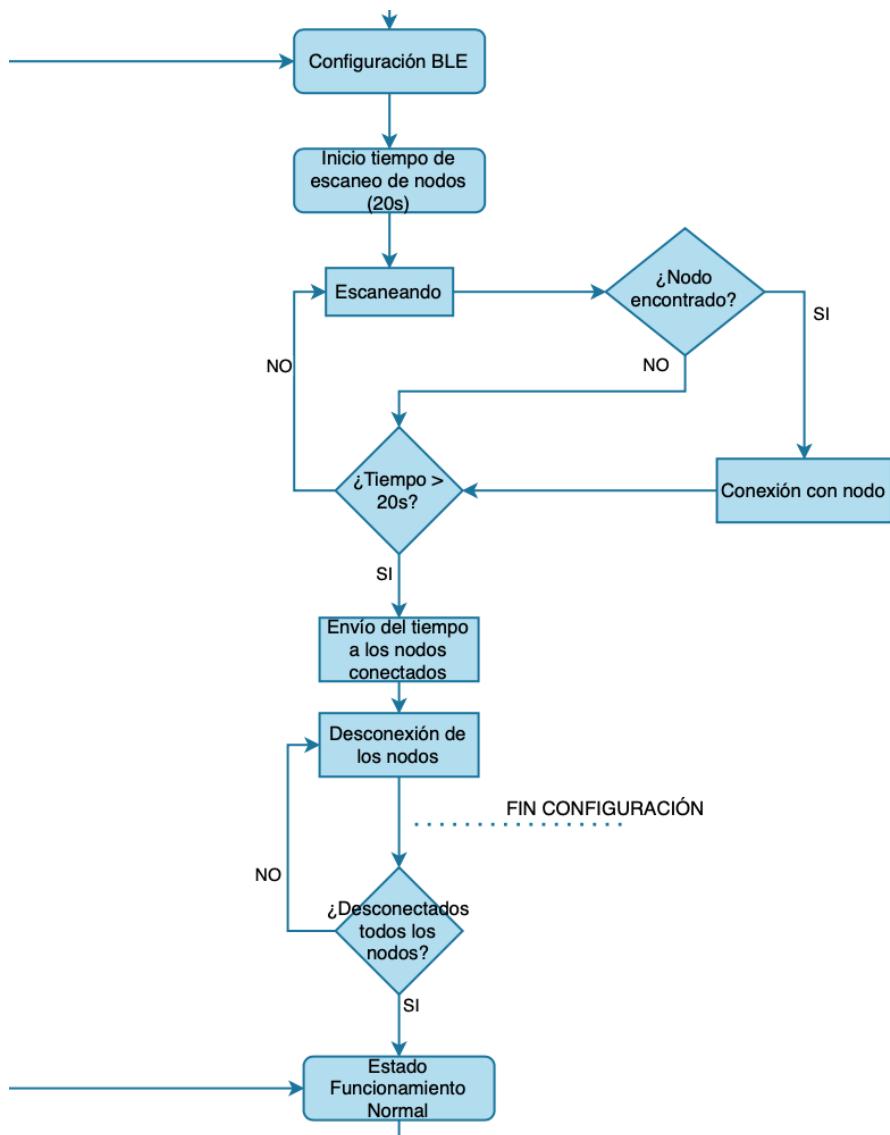


Figura 5.18: Diagrama del protocolo para el dispositivo central-2. Fuente: elaboración propia.

- Registro de los datos de la actividad deportiva

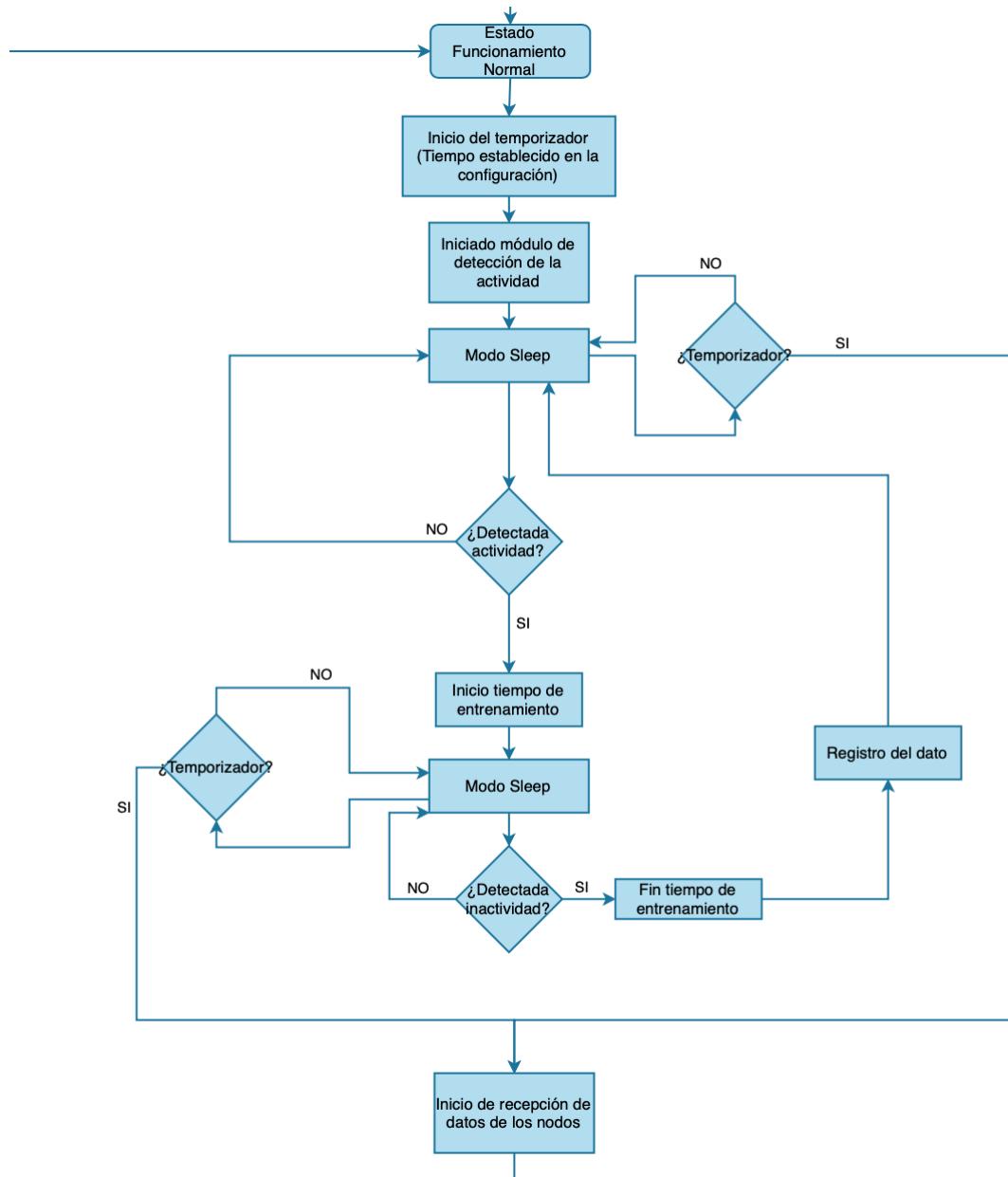


Figura 5.19: Diagrama del protocolo para el dispositivo central-3. Fuente: elaboración propia.

En este estado, mostrado en el diagrama de la Figura 5.19, el dispositivo configura e inicia el temporizador con el tiempo obtenido en la configuración anterior. Una vez es iniciado, se configura el acelerómetro con los parámetros deseados para la detección de la actividad e inactividad, y en seguida se va a un estado de bajo consumo, llamado modo sleep.

En este modo de bajo consumo, el dispositivo está en constante atención por si el acelerómetro detecta actividad. En ese caso, se toma el tiempo de inicio en el que ha comenzado la actividad y vuelve al estado de bajo consumo. Una vez ha detectado la actividad, el acelerómetro no vuelve a detectar más actividad hasta que detecte inactividad durante un cierto tiempo. En caso de que se detectara inactividad, se tomaría el tiempo de fin de entrenamiento y se registraría el dato de duración de entrenamiento, y retornaría al estado de bajo consumo. Este proceso se realiza constantemente hasta llegar al tiempo establecido en el temporizador, cuando se interrumpe la actividad del dispositivo y comienza la recepción de los datos de los nodos.

- **Recepción de los datos de los nodos por BLE**

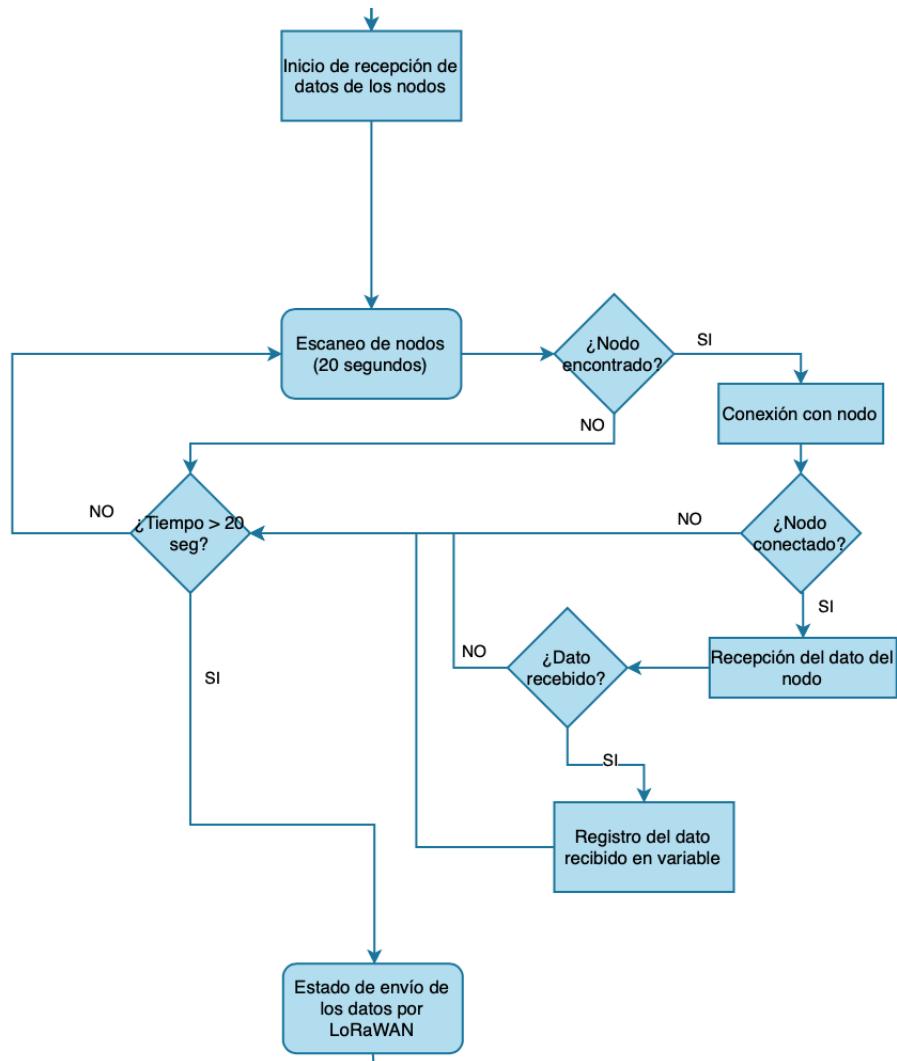


Figura 5.20: Diagrama del protocolo para el dispositivo central-4. Fuente: elaboración propia.

Una vez se ha iniciado la parte de recepción de los datos de los nodos, tal y como se observa en el diagrama de la Figura 5.20, el dispositivo maestro comienza a escanear nodos durante 20 segundos. En cuanto encuentra uno, se conecta a él, recibe el dato enviado por el nodo y lo registra en una variable. Hasta que no hayan pasado los 20 segundos el dispositivo maestro va a estar escaneando nodos, conectándose a ellos y recibiendo los datos. En el momento, en el que se cumplen los 20 segundos de escaneo, el dispositivo se dispone a enviar los datos obtenidos mediante LoRaWAN a Internet, a la plataforma TTN.

- Envío de los datos obtenidos mediante LoRaWAN a Internet.

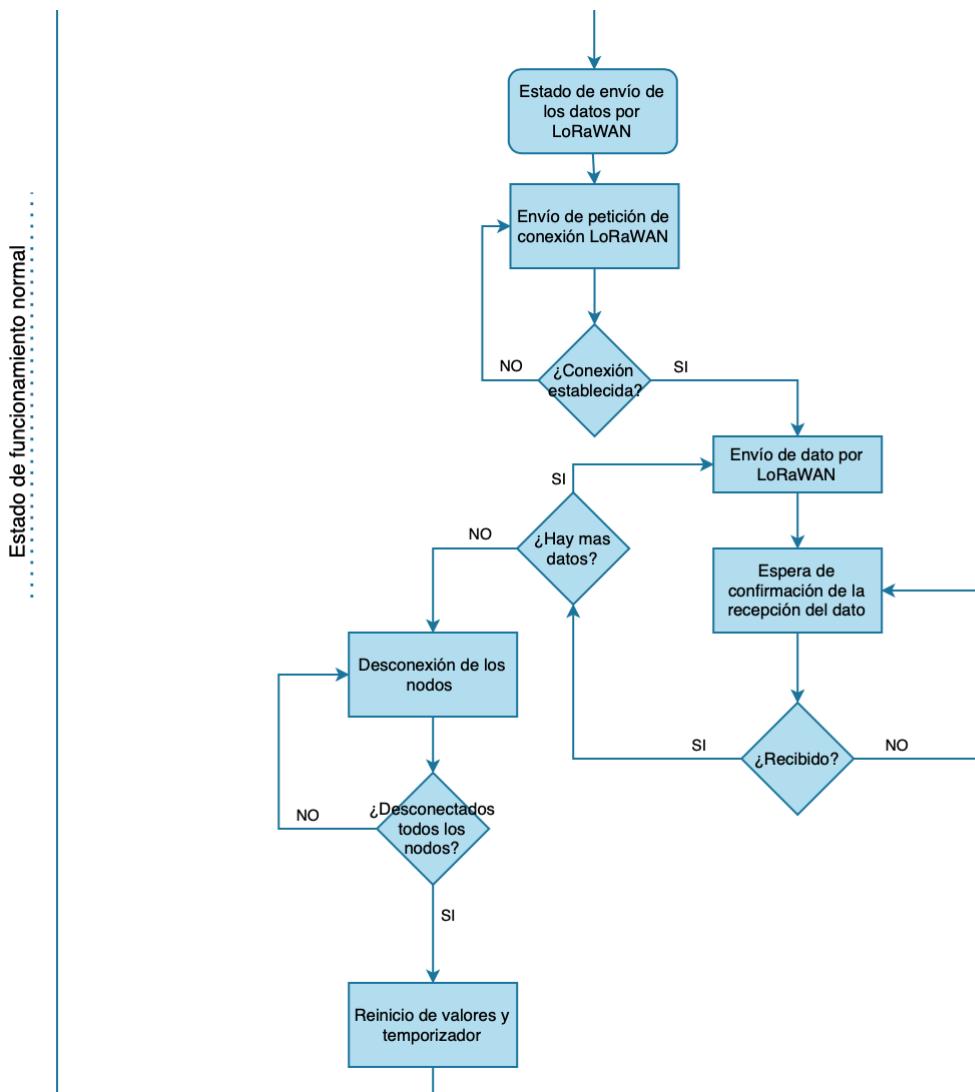


Figura 5.21: Diagrama del protocolo para el dispositivo central-5. Fuente: elaboración propia.

El envío de los datos por LoRaWAN, tal y como se puede ver en el diagrama de la Figura 5.21, se realiza de forma similar al primer estado de configuración cuando se recibía el dato del tiempo.

Primero el dispositivo envía la petición de conexión mediante LoRaWAN, en caso de conexión correcta, se envía el dato de los tiempos de entrenamiento del maestro. Y se espera a recibir la confirmación del envío. Una vez está confirmado, se comprueba si hay más datos. En el caso de que los nodos le hayan enviado sus datos en el proceso anterior, existirán más datos que enviar. En ese caso, se repite el proceso de envío y confirmación hasta que ya no haya más datos que enviar y entonces el dispositivo maestro se desconecta de todos los nodos conectados. Una vez se desconecta de todos los nodos, se reinician todas las variables y valores, y se reinicia el temporizador. En este punto, comienza un nuevo ciclo de recogida de datos y envío a Internet.

- **Reinicio del dispositivo**

Se ha establecido, que el dispositivo se reinicie de forma autónoma una vez al día para que pueda volver a configurarse el tiempo del temporizador diariamente. De tal forma, cada día podría determinarse con qué frecuencia se quieren obtener los datos de entrenamiento.

Protocolo dispositivo nodo

- **Obtención del tiempo del temporizador**

Como se ve en la Figura 5.22, lo primero que hace el dispositivo nodo, es configurar el Bluetooth para poder obtener el dato del tiempo para el temporizador desde el dispositivo maestro. Una vez se encuentra configurado, realiza el proceso llamado advertising. Este proceso consiste en mostrarse con ciertos datos como el nombre del dispositivo y el servicio que ofrece mediante Bluetooth, para que el dispositivo maestro cuando escanea, pueda detectar los dispositivos nodos.

Una vez realiza el advertising, el nodo está a la espera de que el dispositivo maestro se conecte a él. En el momento en el que se realiza la conexión, el nodo recibe el dato del tiempo, y entonces se va a un estado de espera para que el maestro se desconecte del nodo. Cuando el dispositivo maestro se desconecta del nodo, éste comienza el estado de funcionamiento normal, en el que se permite el registro de la actividad deportiva.

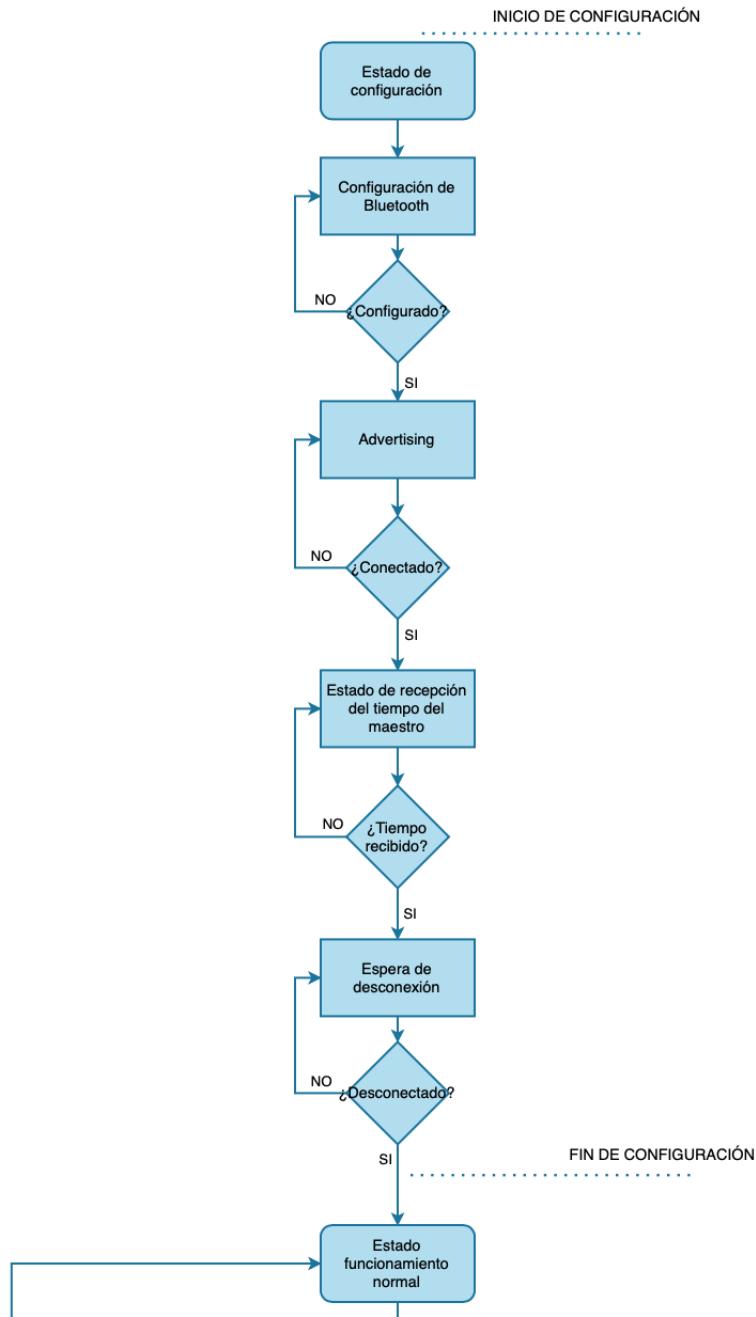


Figura 5.22: Diagrama del protocolo para el dispositivo nodo-1. Fuente: elaboración propia.

■ Registro de los datos de la actividad deportiva

A la hora de registrar los datos deportivos, el dispositivo nodo funciona exactamente de igual forma que el dispositivo maestro, tal y como se puede observar en la Figura 5.23. Con la única diferencia que cuando se llega al tiempo del temporizador en vez de recibir los datos de los nodos, el dispositivo nodo debe

enviar el dato recogido al dispositivo maestro.

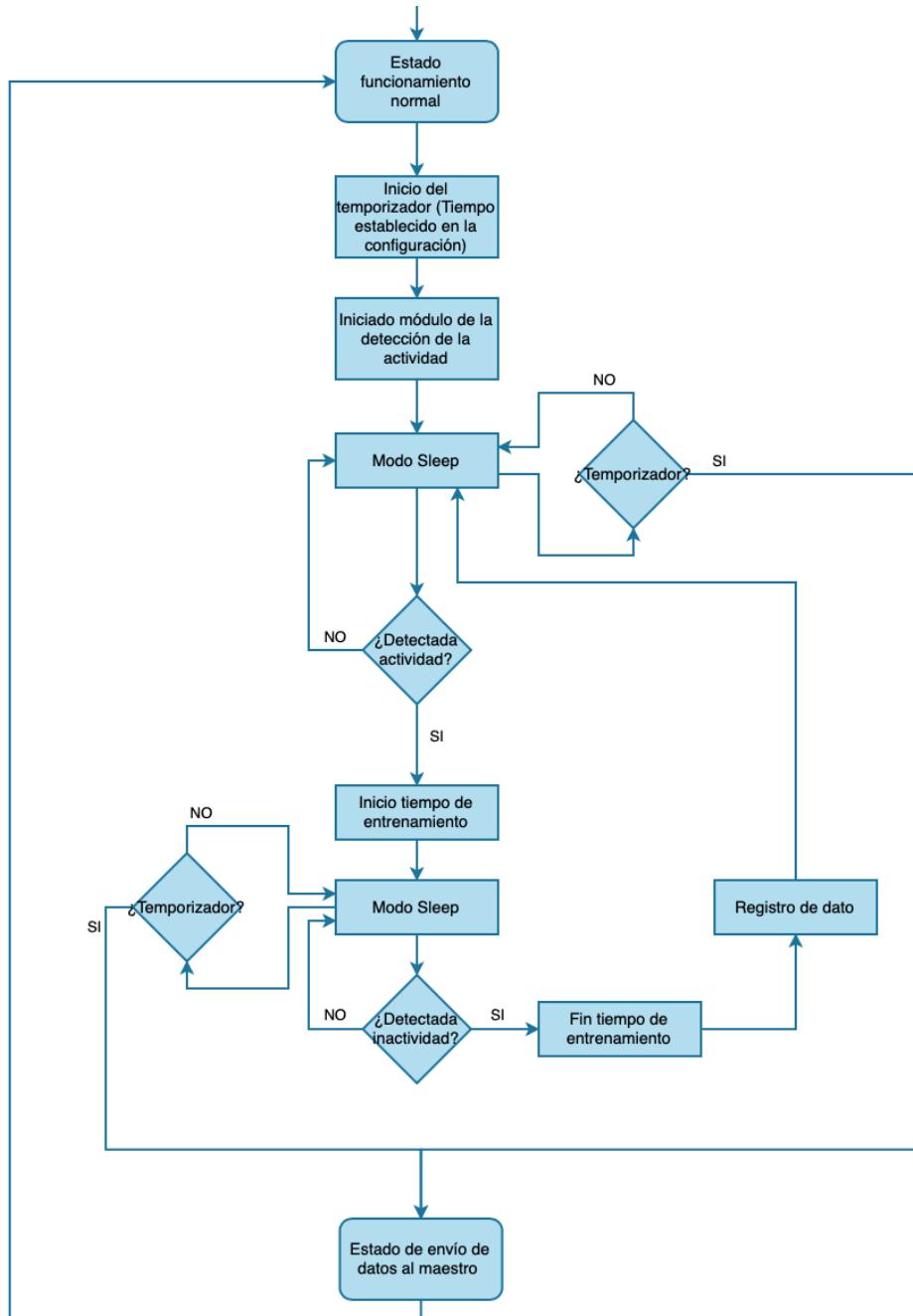


Figura 5.23: Diagrama del protocolo para el dispositivo nodo-2. Fuente: elaboración propia.

- Envío de los datos al dispositivo maestro por BLE

Esta es la última parte del protocolo para el dispositivo nodo. En ella, se envían los datos obtenidos del entrenamiento al dispositivo maestro mediante

Bluetooth. Para ello, el nodo realiza el proceso de advertising, hasta que el dispositivo maestro se conecta, y entonces el nodo le envía los datos. Una vez enviados, se espera a que el dispositivo maestro se desconecte y entonces se reinician los valores, las variables y el temporizador. Una vez reiniciado todo, comienza de nuevo el modo de recogida de datos y comienza un nuevo ciclo. Este proceso se muestra en la Figura 5.24.

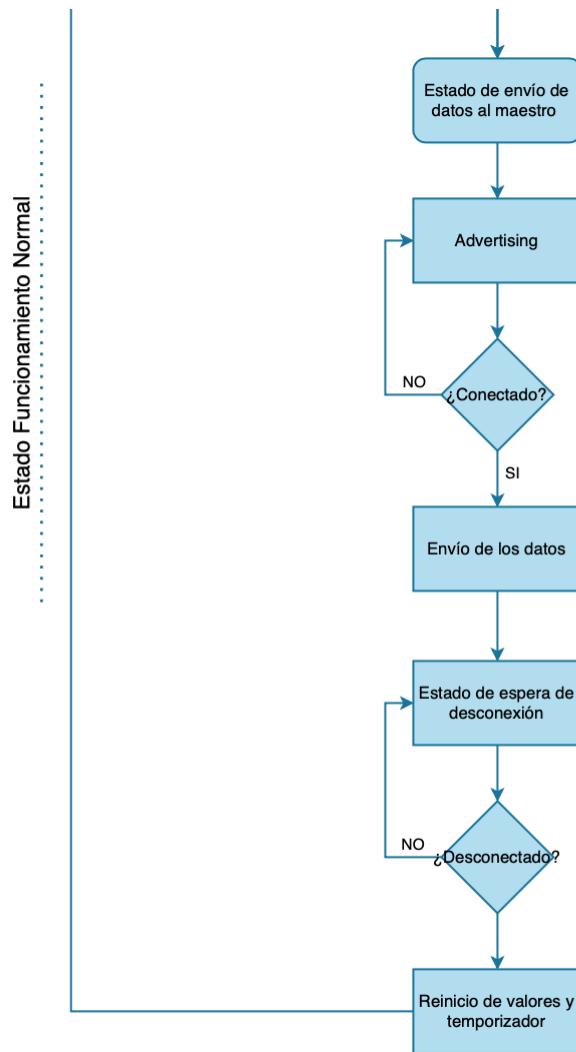


Figura 5.24: Diagrama del protocolo para el dispositivo nodo-3. Fuente: elaboración propia.

- **Reinicio del dispositivo** De igual forma que el dispositivo maestro, los nodos se reiniciarán en un cierto momento del día para volver a configurar el tiempo del temporizador.

5.2.9. Programación de los dispositivos

Para realizar la programación de los dispositivos se ha usado el entorno de programación de Arduino IDE. Por lo que se ha realizado con lenguaje C++. Se ha seguido un esquema de máquina de estados, para que el dispositivo pueda ir de estado a estado de forma sencilla. A continuación, en las Figuras 5.25 y 5.26 se muestra un diagrama para cada tipo de dispositivo, en el que se puede visualizar de forma sencilla, todos los estados por los que pasa cada dispositivo.

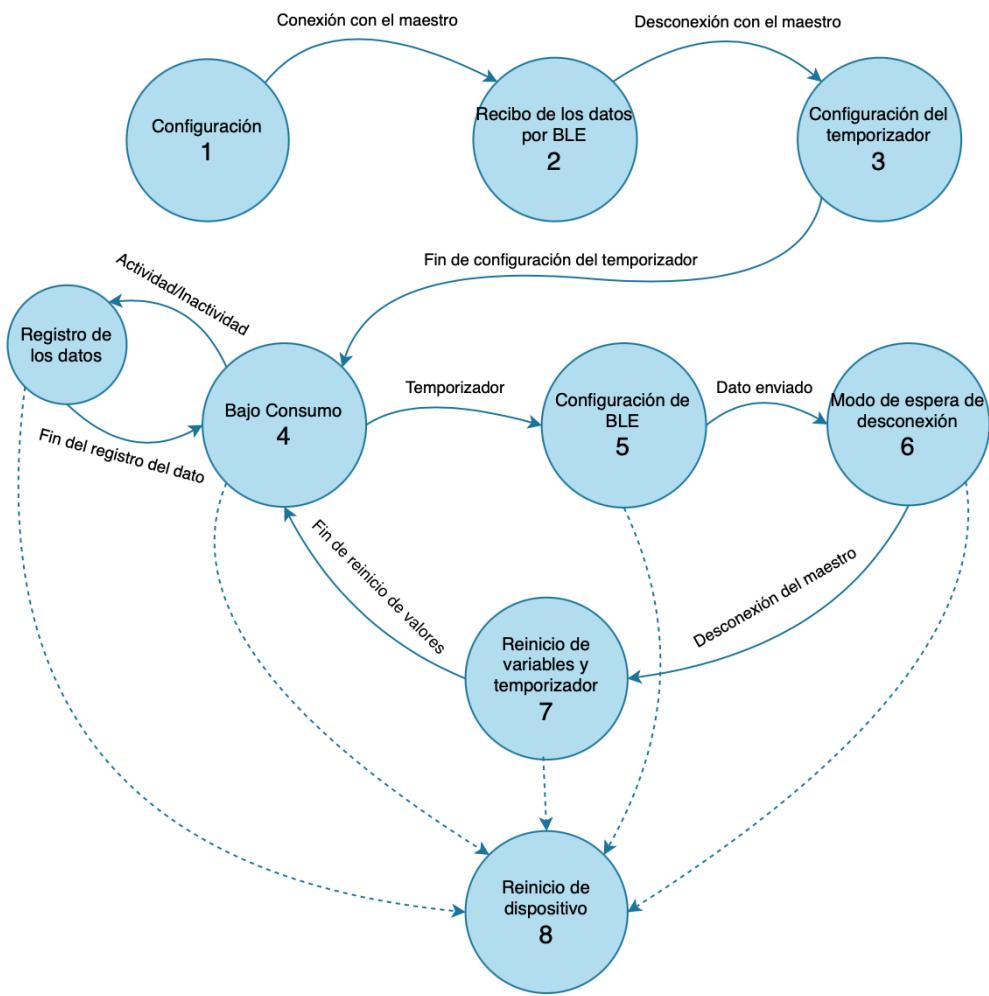


Figura 5.25: Diagrama de estados del dispositivo nodo. Fuente: elaboración propia.

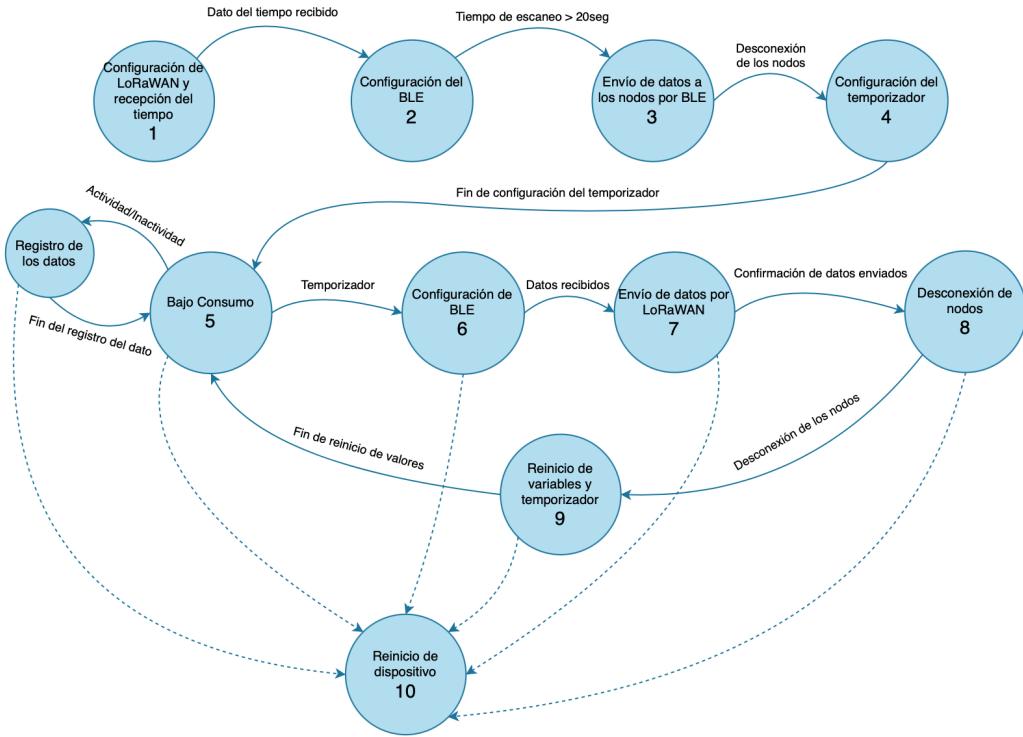


Figura 5.26: Diagrama de estados del dispositivo maestro. Fuente: elaboración propia.

5.3. Diseño del backend

Una vez los datos son enviados a la plataforma de TTN, éstos se pueden visualizar de forma instantánea en la misma, tal y como se puede ver en la Figura 5.27. Los datos son recibidos en la plataforma de TTN en hexadecimal, por lo que ha sido necesario el siguiente script en la propia plataforma para poder visualizar los datos en forma de texto. Este script se ha realizado en Javascript y se puede observar en el Listing 5.1.

```

1 function Decoder(bytes, port) {
2   return {Datos: String.fromCharCode.apply(null, bytes)};
3 }
```

Listing 5.1: Decodificador de datos en TTN. Fuente: elaboración propia.

En TTN los datos no se pueden guardar ni tratar de forma sencilla. Por ello, se ha realizado un diseño de tratamiento y visualización de los datos. La plataforma TTN permite conectarse con integraciones o servidores externos en los que es posible realizar aplicaciones para poder visualizar los datos y tratarlos a nuestro antojo. Una de estas integraciones que mejor se adapta a las necesidades del proyecto es TagoIO, la cual nos permite una gran cantidad de widgets con los que poder visualizar los

datos de forma instantánea. También permite el tratamiento de los datos mediante código Node.js o Python 3.

Figura 5.27: Plataforma de TTN. Fuente: elaboración propia.

5.3.1. Tratamiento de los datos

El dato obtenido y enviado desde los dispositivos a la red es una cadena de texto del siguiente estilo:

Identificador;tiempo1;tiempo2;tiempo3...

El primer concepto es el nombre del dispositivo que ha registrado los datos. Las siguientes partes del dato son los diferentes tiempos de duración del entrenamiento de las máquinas. Todo ello, separado mediante el separador ";" para que sea sencillo el reconocimiento entre un tiempo y otro y el tratamiento se pueda realizar de forma más visual y fácil.

La integración TagoIO, como ya se ha mencionado permite tratar los datos mediante scripts en Node.js o Python 3. Por la sencillez y por la gran información que hay en la propia página de TagooIO se han tratado los datos mediante Node.js (JavaScript) para obtener el tiempo acumulado de entrenamiento en el tiempo en el que se ha tomado los datos y las personas que han usado la máquina en ese mismo tiempo. Este último dato se calcula contando los registros que hay de duraciones de entrenamiento. Este dato no es muy realista, ya que una persona puede parar de ejercitarse cierto tiempo y volver a entrenar. Entonces, el dispositivo registrará

dos tiempos de entrenamiento cuando solamente una persona ha usado la máquina. Aún no siendo realista, el dato puede ayudar estadísticamente o puede ser una aproximación de las veces que se ha usado la máquina. El script usado para realizar el tratamiento de los datos se puede observar en el Listing 5.2.

```

2 const rawValues = payload.find(item=>item.variable === 'datos');

4 if (rawValues){

6 const valoresString = rawValues.value;
7 const valoresSplit = valoresString.split(';');
8 const nombreDisp = valoresSplit[0];
9 var sumanodo1 = 0;
10 var sumacentral = 0;
11 var numeropersonascentral;
12 var numeropersonasnodo1;

14 if (valoresString === 'nodo1')
{
16   sumanodo1 = 0;
17   numeropersonasnodo1 = 0;
18   payload.push({ "variable": "sumanodo1", "value": sumanodo1, "unit": "seg",
19     "metadata": { "x": 0.72, "y": 0.47, "color": "green" } });
20   payload.push({ "variable": "numeropersonasnodo1", "value":
21     numeropersonasnodo1, "unit": "personas", "metadata": { "x": 0.69, "y":
22       0.62, "color": "green" } });
23 }
24 else if (nombreDisp === 'nodo1')
{
25   var j = 0;
26
27   for (var i = 1; i < valoresSplit.length; i++ )
28   {
29
30     sumanodo1 += parseInt(valoresSplit[i]);
31     if (valoresSplit[i] === '0')
32     {
33       j++;
34       continue;
35     }
36
37     numeropersonasnodo1 = i-j;
38
39   }
40
41   payload.push({ "variable": "sumanodo1", "value": sumanodo1, "unit": "seg",
42     "metadata": { "x": 0.72, "y": 0.47, "color": "green" } });
43   payload.push({ "variable": "numeropersonasnodo1", "value":
44     numeropersonasnodo1, "unit": "personas", "metadata": { "x": 0.69, "y": 0.62, "color": "green" } });
45 }
```

```
:0.62, "color": "green"}});  
44}  
46 if (valoresString == 'Central')  
{  
48    sumacentral = 0;  
49    numeropersonascentral = 0;  
50    payload.push({"variable": "sumacentral", "value": sumacentral, "unit":  
51      "seg", "metadata": {"x": 0.25, "y": 0.39, "color": "blue"}});  
52    payload.push({"variable": "numeropersonascentral", "value":  
53      numeropersonascentral, "unit": "personas", "metadata": {"x": 0.24,  
54      "y": 0.53, "color": "blue"}});  
55}  
56 else if (nombreDisp == 'Central')  
{  
57    var j = 0;  
58    for (var i = 1; i < valoresSplit.length; i++)  
59    {  
60        if (valoresSplit[i] == '0')  
61        {  
62            j++;  
63            continue;  
64        }  
65        sumacentral += parseInt(valoresSplit[i]);  
66        numeropersonascentral = i-j;  
67    }  
68    payload.push({"variable": "sumacentral", "value": sumacentral, "unit":  
69      "seg", "metadata": {"x": 0.25, "y": 0.39, "color": "blue"}});  
70    payload.push({"variable": "numeropersonascentral", "value":  
71      numeropersonascentral, "unit": "personas", "metadata": {"x": 0.24,  
72      "y": 0.53, "color": "blue"}});  
73}  
74}
```

Listing 5.2: Tratamiento de los datos en TagoIO. Fuente: elaboración propia

Suponiendo que el dato obtenido es "nodo1;10;5;3;7;12". El dato indica que son los tiempos de entrenamiento obtenidos por el nodo número 1, y que la máquina se ha usado 5 veces durante 10 minutos, 5 minutos, 3 minutos, 7 minutos y 12 minutos. Mediante el tratamiento de los datos se obtiene por tanto un uso total de 5 veces y 37 minutos en total.

5.3.2. Visualización de los datos

En TagoIO existe la posibilidad de crear un dashboard (tablero) con varios widgets en los que es posible representar los datos ya tratados. En la Figura 5.28, se puede visualizar el tablero creado donde se muestran los diferentes widgets identificados mediante un número para poder explicarlos y mencionarlos a continuación.



Figura 5.28: Dashboard en TagoIO. Fuente: elaboración propia.

A continuación, se explica el dashboard creado teniendo como referencia la Figura 5.28.

- **Widget 1:** En este widget se representa el tiempo acumulado de los dispositivos a lo largo del tiempo. También mediante un botón mostrado en los puntos 1.1 y 1.2 de la Figura 5.28 es posible acceder a todos los registros de tiempos acumulados obtenidos para cada dispositivo.
- **Widget 2:** En este widget se representa el número de veces que se ha usado un equipo deportivo a lo largo del tiempo. Mediante un botón mostrado en los puntos 2.1 y 2.2 de la Figura 5.28 se puede acceder a todos los registros de las veces de uso de los equipos.
- **Widget 3:** Se representa el último dato recibido de cualquier dispositivo. De igual forma que en los widgets anteriores, mediante un botón mostrado en el punto 3.1 de la Figura 5.28 se puede acceder a todos los registros de todos los datos recibidos.
- **Widget 4:** Este widget se usa para poder enviar el dato de configuración del temporizador a los dispositivos. Para ello, es necesario insertar el tiempo en minutos en el que los dispositivos deberán enviar los datos y ciclar en el botón

“Enviar”. Para poder haber realizado este widget ha sido necesario la creación de un script mostrado en el Listing 5.3.

- **Widget 5:** En este widget se representan los últimos datos de tiempo acumulado y de las veces de uso recibidos de una forma visual e interactiva.

El dashboard creado donde se visualizan todos los datos se puede ver en el siguiente enlace: [Enlace Web](#).

TagoIO también dispone de aplicación móvil, es decir se pueden visualizar los datos y enviar el tiempo de configuración para el temporizador desde el propio móvil con la propia aplicación de TagoIO. En las Figuras 5.29, 5.30 y 5.31 se pueden ver algunas capturas de la aplicación.



Figura 5.29: Captura de la aplicación de TagoIO. Imagen 1. Fuente: elaboración propia.



Figura 5.30: Captura de la aplicación de TagoIO. Imagen 2. Fuente: elaboración propia.

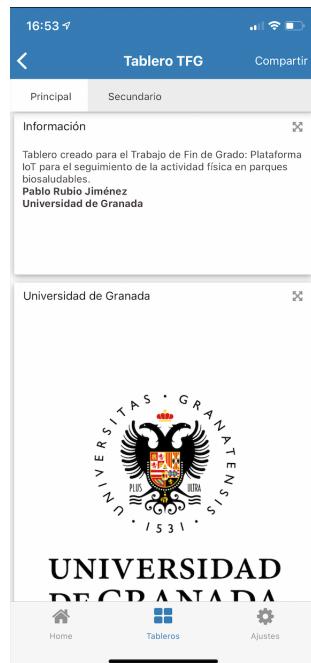


Figura 5.31: Captura de la aplicación de TagoIO. Imagen 3. Fuente: elaboración propia.

Mediante el siguiente script proporcionado por TagoIO, se permite enviar mensajes desde el dashboard de TagoIO al dispositivo con TTN como intermediario. El

script se muestra en el Listing 5.3.

```
1 const { Analysis, Account, Utils } = require('@tago-io/sdk');
2 const axios = require('axios');
3
4 async function init(context, scope) {
5   if (!scope[0]) return context.log('This analysis must be triggered by
6     a widget.');
7
8   context.log('Downlink analysis started');
9
10  const env = Utils.envToJson(context.environment);
11  if (!env.account_token) return context.log('Missing "account_token"
12    environment variable');
13  else if (env.account_token.length !== 36) return context.log('Invalid
14    "account_token" in the environment variable');
15
16  const account = new Account({ token: env.account_token });
17
18  sent by the widget/dashboard.
19  const payload = scope.find(x => x.variable === 'form_payload') || {
20    value: env.payload, origin: env.device_id };
21  const port = scope.find(x => x.variable === 'form_port') || { value:
22    env.default_PORT };
23  if (!payload || !payload.value || !payload.origin) return context.log
24    ('Missing "form_payload" in the data scope.');
25  else if (!port || !port.value) return context.log('Missing "form_port"
26    in the data scope.');
27
28  const device_id = payload.origin; // All variables that trigger the
29  analysis have the "origin" parameter, with the TagoIO Device ID.
30  if (!device_id) return context.log('Device ID <origin> not found in
31  the variables sent by the widget/dashboard.');
32
33
34  const device_tokens = await account.devices.tokenList(device_id, {
35    page: 1, fields: [ 'name', 'serie_number', 'last_authorization' ],
36    amount: 10 });
37  const token = device_tokens.find(x => x.serie_number && x.
38    last_authorization);
39  if (!token) return context.log("Couldn't find a token with serial/
40    authorization for this device");
41
42  // Get the connector ID from the device
43  const { network: network_id } = await account.devices.info(device_id)
44  ;
45  if (!network_id) return context.log('Device is not using a network.')
46  ;
47
48  const network = await account.integration.networks.info(network_id, [
49
```

```

    'id', 'middleware_endpoint', "name"]);
if (!network.middleware_endpoint) return context.log("Couldn't find a
    network middleware for this device.");
37

39 const params = await account.devices.paramList(device_id);
let downlink_param = params.find(x => x.key === 'downlink');
41 downlink_param = { id: downlink_param ? downlink_param.id : null, key
    : 'downlink', value: String(payload.value), sent: false };
await account.devices.paramSet(device_id, downlink_param);
43

45 context.log('Trying to send the downlink');
46 const data = {
    device: token.serie_number,
47     authorization: token.last_authorization,
    payload: payload.value,
48     port: port.value,
49 };
50

51 await axios.post(`https://${network.middleware_endpoint}/downlink`,
    data)
52 .then((result) => {
    context.log(`Downlink accepted with status ${result.status}`);
53 })
54 .catch((error) => {
    context.log(`Downlink failed with status ${error.response.status}`);
55     context.log(error.response.data || JSON.stringify(error));
56 });
57 }

58 module.exports = new Analysis(init);
61

```

Listing 5.3: Widget para envío de datos Downlink en TagoIO. Fuente: TagoIO

5.4. Diseño de la carcasa

El dispositivo debe ir instalado en las máquinas de ejercitación, por ello, que es imprescindible una carcasa que sea capaz de colocarse en las barras y que resista el movimiento y la intemperie. Por ello, debe ser una carcasa robusta pero a la vez debe mimetizarse con el entorno para evitar posibles vandalismos. También uno de los puntos a tener en cuenta, con respecto al diseño es el hecho de poder abrir la carcasa en momentos dados para poder cambiar las baterías o intercambiar el dispositivo en caso de avería o similar. Para ello, se ha usado una tapadera unida mediante tornillería para evitar que sea sencilla la apertura en cuanto a vandalismo.

Por todo ello, se ha diseñado un prototipo de carcasa que no cumplirá con los requisitos de protección frente a la exposición a la intemperie, pero si servirá para poder adherirse a las máquinas y poder obtener datos de ellas. En las siguientes Figuras 5.32, 5.33, 5.34 y 5.35 se puede ver la carcasa diseñada mediante el software Solidworks, la cual permitirá la adhesión a la máquina mediante un sistema de abrazadera, típica en los componentes que se suelen añadir a las barras de las bicicletas y que deben soportar el movimiento.

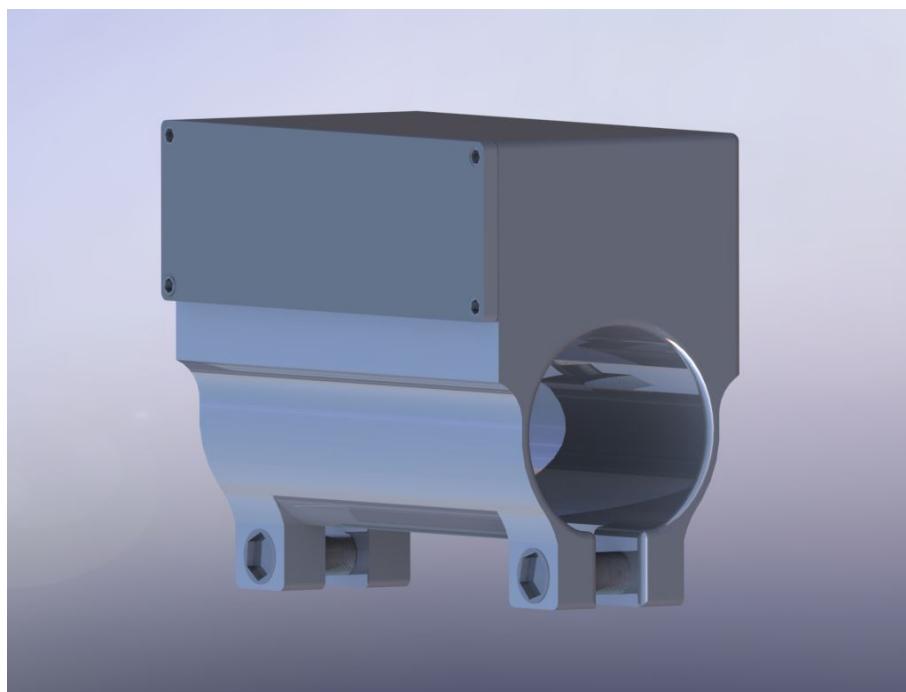


Figura 5.32: Diseño de carcasa. Imagen 1. Fuente: elaboración propia.

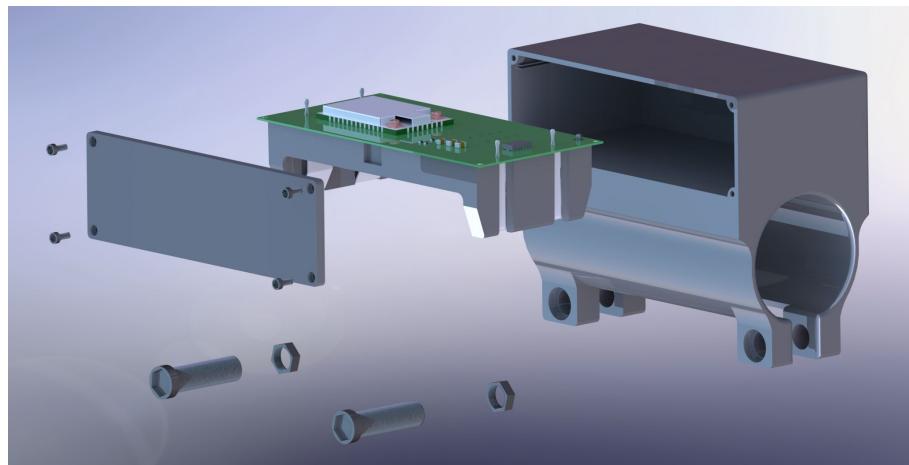


Figura 5.33: Diseño de carcasa. Imagen 2. Fuente: elaboración propia.

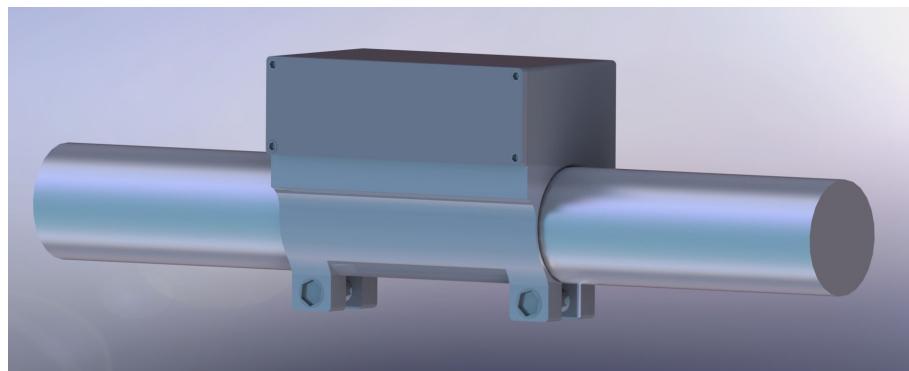


Figura 5.34: Diseño de carcasa. Imagen 3. Fuente: elaboración propia.

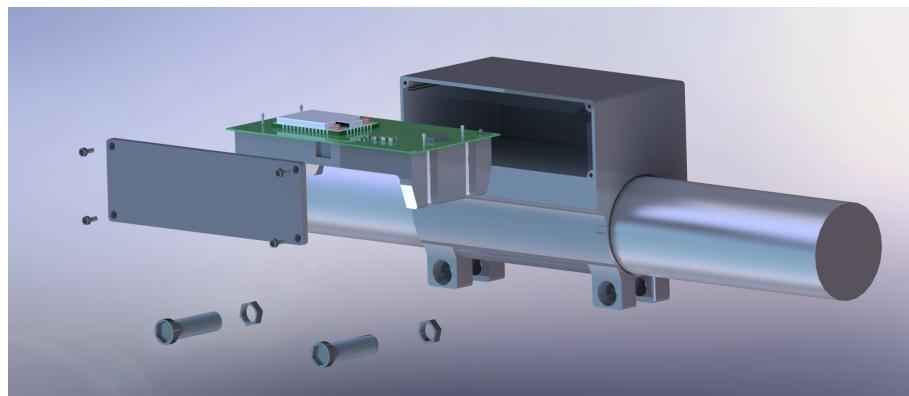


Figura 5.35: Diseño de carcasa. Imagen 4. Fuente: elaboración propia.

Capítulo 6

Planificación y presupuesto

6.1. Diagrama de Gantt

Con el fin de mostrar toda la planificación se ha realizado un Diagrama de Gantt donde se puede visualizar todo el proceso de realización del proyecto con los recursos y fechas concretas. Este diagrama de Gantt se muestra en las Figuras 6.1, 6.2, 6.3, 6.4 y 6.5. La realización del proyecto se ha dividido en 6 partes: el estado del arte, el análisis en profundidad de los parques biosaludables, las propuestas de diseño, el estudio del diseño propuesto, el desarrollo del diseño y la documentación.

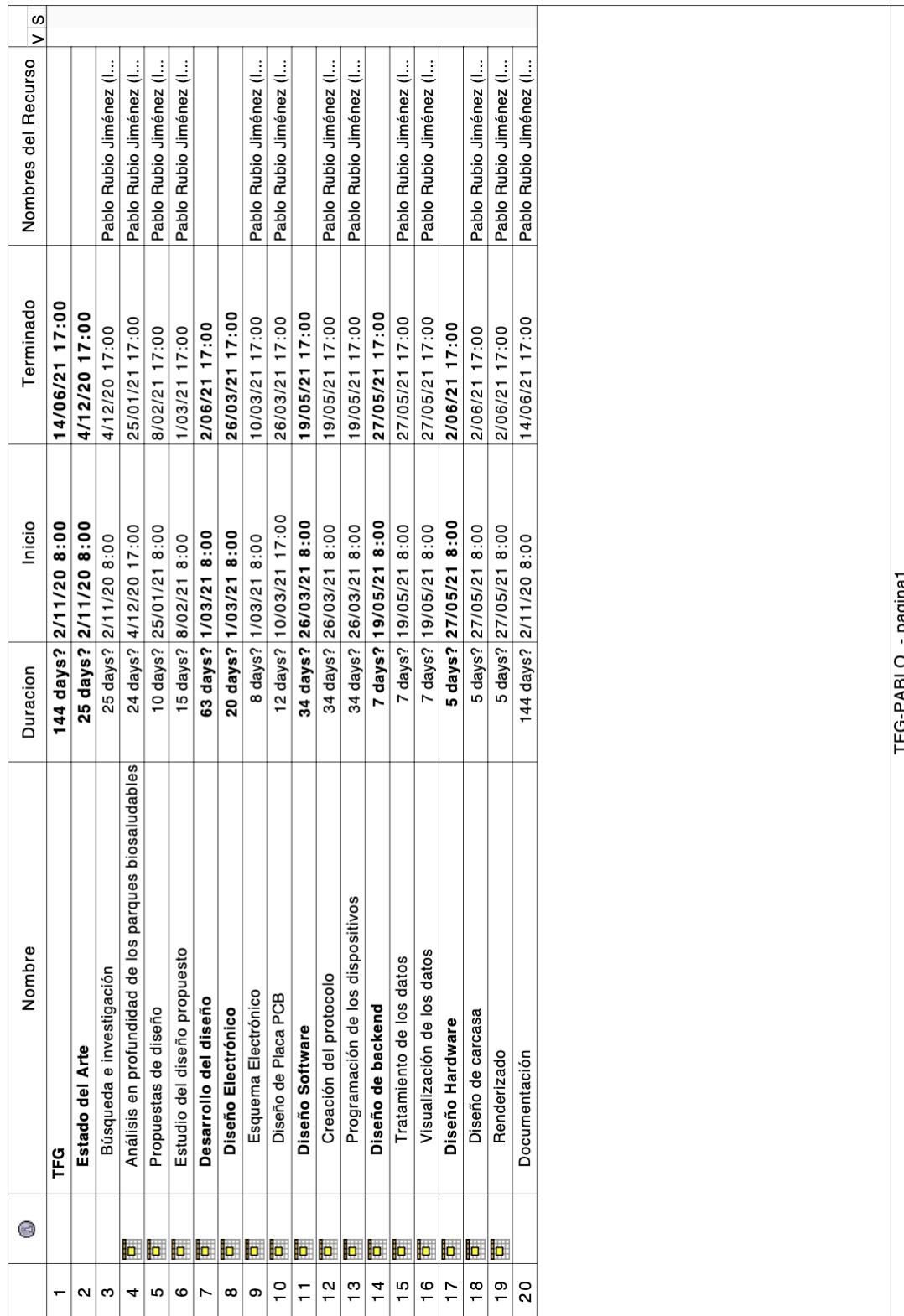


Figura 6.1: Diagrama de Gantt. Imagen 1. Fuente: elaboración propia.

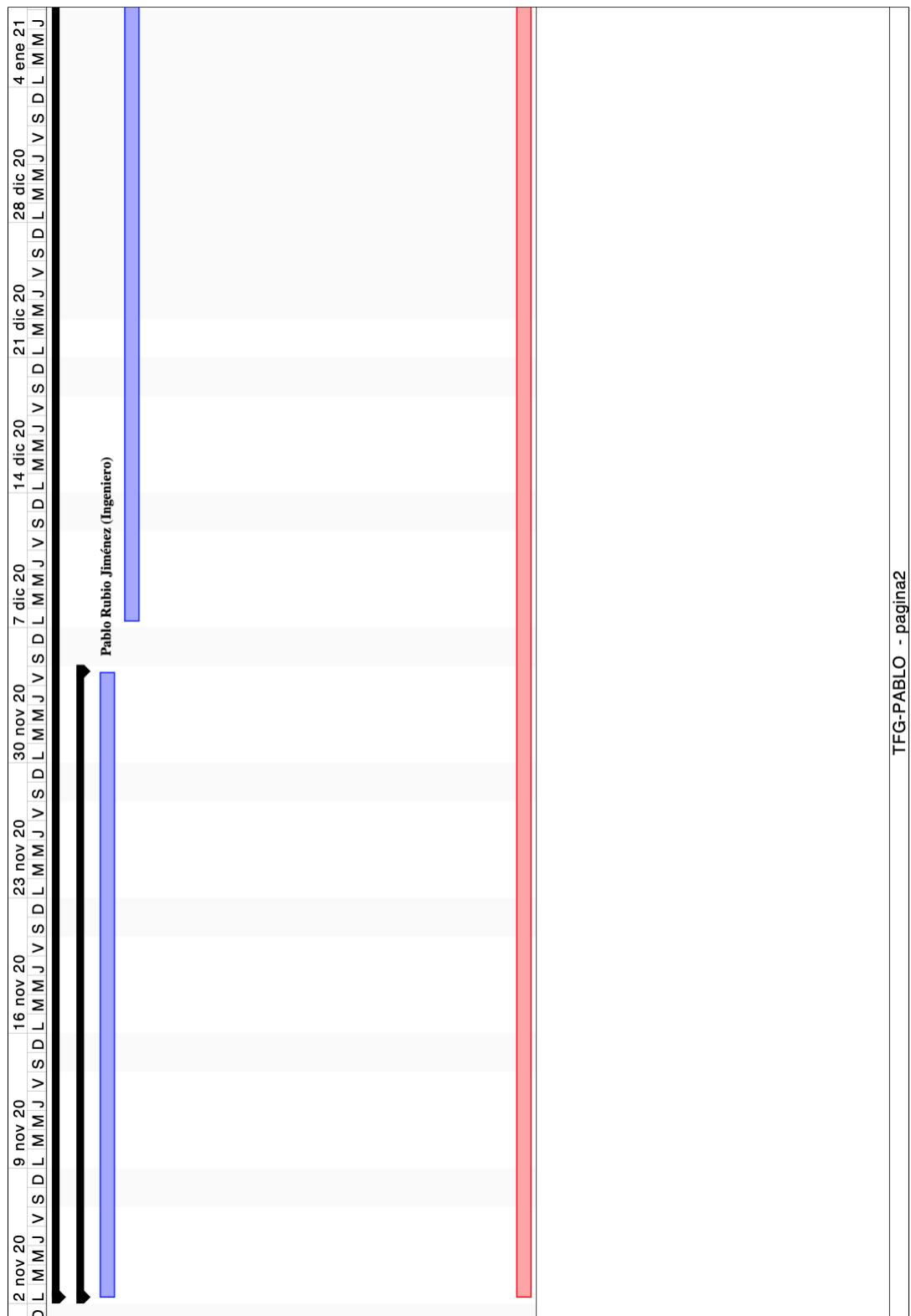


Figura 6.2: Diagrama de Gantt. Imagen 2. Fuente: elaboracion propia.

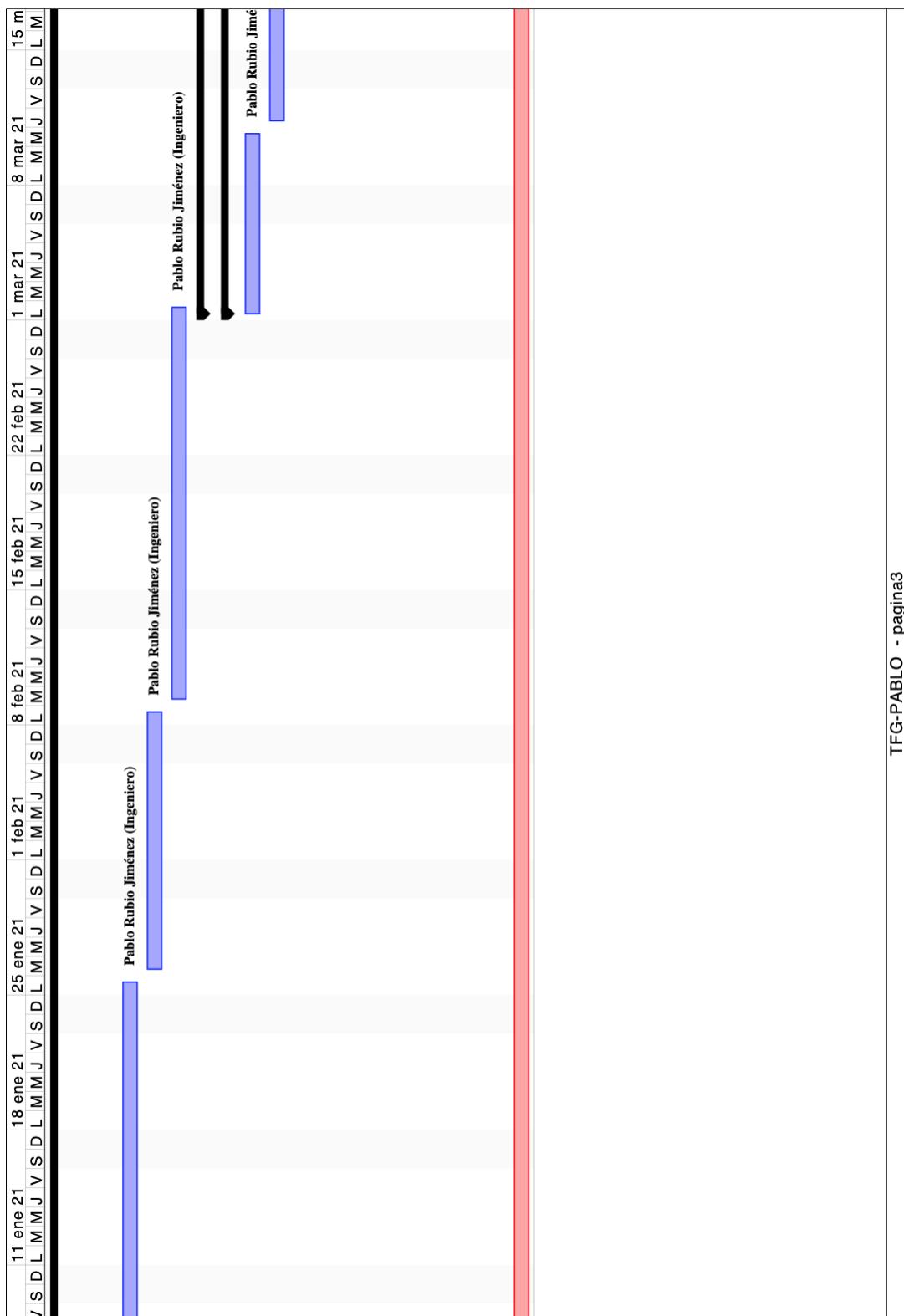


Figura 6.3: Diagrama de Gantt. Imagen 3. Fuente: elaboración propia.

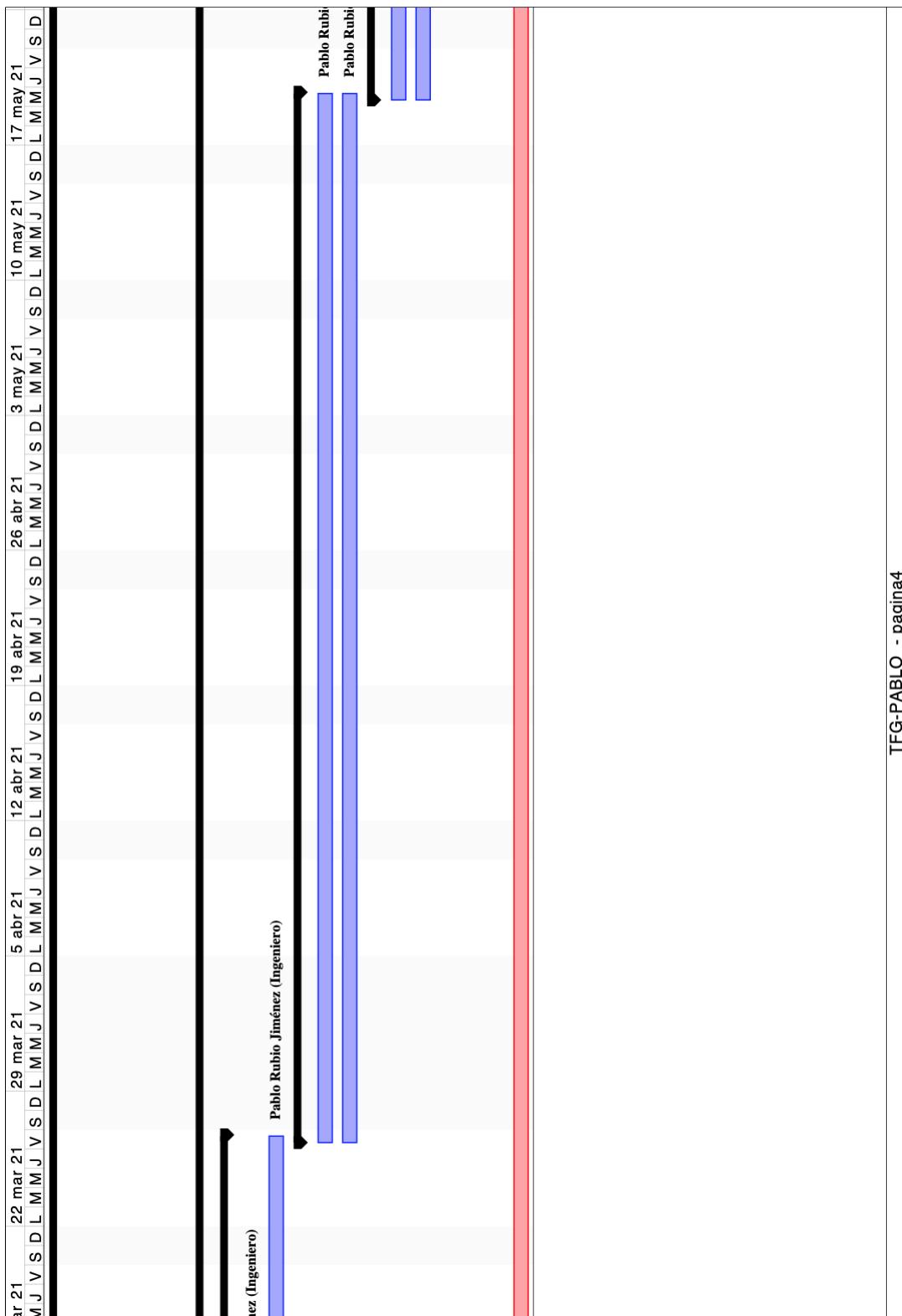


Figura 6.4: Diagrama de Gantt. Imagen 4. Fuente: elaboración propia.

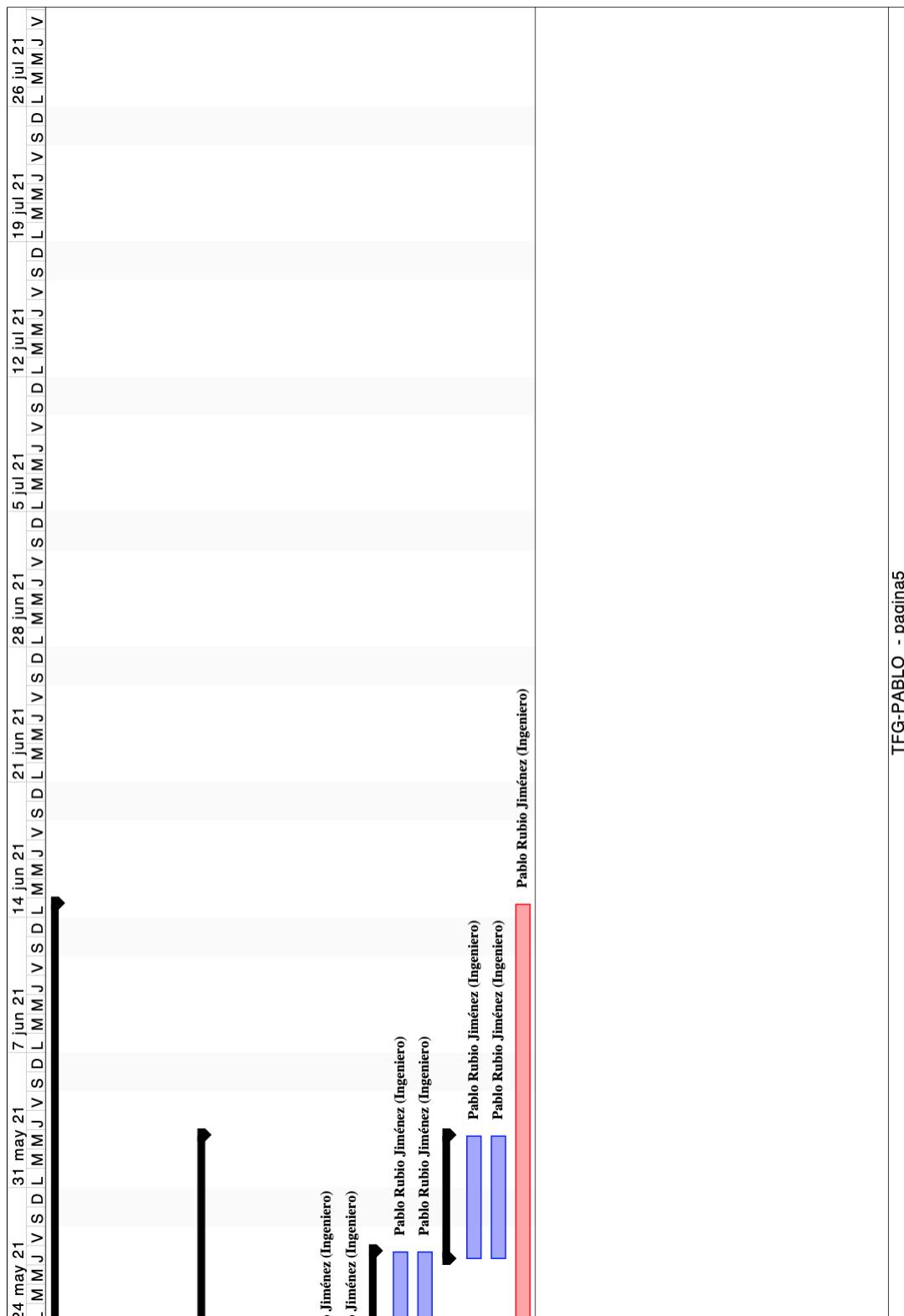


Figura 6.5: Diagrama de Gantt. Imagen 5. Fuente: elaboración propia.

6.2. Presupuesto

6.2.1. Coste material

El coste material está compuesto por todos aquellos componentes físicos o no físicos que han sido necesarios para la elaboración del proyecto. En la Tabla 6.1 se muestra un listado de todos los componentes con sus respectivos costes.

Tabla 6.1: Costes de los materiales y recursos

Recurso	Cantidad	Precio Unitario	TOTAL
Acelerómetro ADXL345	2	2,11 €	4,22 €
Led Rojo	2	1,04 €	2,08 €
Kit de resistencias	1	2,78 €	2,78 €
Kit de condensadores	1	4,59 €	4,59 €
RAK4600 WisDUO x5	1	41,20 €	41,20 €
RAK4600 Placa de Adaptación	2	16,92 €	33,84 €
RAKDAP1	1	8,41 €	8,41 €
FT232	1	11,99 €	11,99 €
The things Gateway Indoor	1	89,00 €	89,00 €
Cables x120	1	6,99 €	6,99 €
Componentes para placa PCB	1	43,75 €	43,75 €
Portapilas AA	2	4,50 €	9,00 €
Placas PCB x5	1	22,50 €	22,50 €
Licencia SolidWork 3 meses	1	1.010,00 €	1.010,00 €
Licencia Altium Designer 1 año	1	1.747,00 €	1.747,00 €

El coste material y de los recursos utilizados total es: **3.037,35 €**.

6.2.2. Coste de mano de obra

A parte del coste material, existe el coste de la mano de obra mostrado en la Tabla 6.2. En este caso, los únicos costes de mano de obra utilizados para la elaboración del proyecto han sido los del autor del proyecto y el servicio de soldadura de los componentes sobre la placa PCB. El coste de la mano de obra del autor del proyecto va en función de las horas trabajadas. En este caso, se han trabajado 349 horas y considerando un sueldo medio para un Ingeniero Electrónico Industrial Junior de 20,33 €/hora según Jobted [31], el coste total sería de 7.095,17 €. El coste del servicio de soldadura de los componentes sobre la placa PCB ha sido de 100,00 €.

Tabla 6.2: Costes de la mano de obra

Recurso	Coste
Ingeniero Electrónico Industrial	7.095,17 €
Servicio de soldadura	100,00 €

Los costes de mano de obra totales, por tanto, han sido de **7.195,17 €**. En resumen, los costes totales se pueden ver en la siguiente Tabla 6.3.

Tabla 6.3: Costes totales del proyecto.

Recurso	Coste
Coste Material	3037,35 €
Coste de mano de obra	7195,17 €
Coste Total	10.232,52 €

Capítulo 7

Conclusiones y líneas futuras

Como conclusión me gustaría resaltar el gran aprendizaje que he obtenido y lo completo e interesante que me ha resultado este proyecto. A pesar de encontrarme con varios problemas a la hora de desarrollarlo, estoy muy satisfecho con los resultados de éste. Uno de los obstáculos ha sido la programación de los dispositivos, ya que he tenido que aprender protocolos de telecomunicación que no conocía como LoRaWAN o Bluetooth Low Energy, además de programar un microcontrolador atípico o poco común en este tipo de proyectos universitarios. Quizás esta haya sido la dificultad más grande que me he encontrado en el desarrollo del proyecto. Otro de los problemas o dificultades ha sido el consumo del dispositivo, por ciertas limitaciones en cuanto a software. A pesar de haberse reducido todo lo posible no se ha podido llegar al deseado. De todas formas, para un primer prototipo, creo que el resultado es óptimo y que podría servir para una posible mejora en el futuro.

En cuanto a los resultados obtenidos, se ha conseguido cumplir con todos los objetivos propuestos al principio del proyecto. Se ha logrado diseñar e implementar un dispositivo capaz de instalarse en los equipos de fitness de los parques biosaludables y poder monitorizarlos de forma remota. También se ha cumplido el objetivo de diseñar e implementar una plataforma capaz de tratar los datos obtenidos y poder visualizarlos de forma inmediata. Con este dispositivo se puede calcular el tiempo total que una máquina ha estado en movimiento y las veces que se ha usado. Datos que considero muy interesantes para poder conocer mejor el uso de los parques biosaludables.

Considero que este proyecto es un muy buen primer punto de partida y creo que abre un gran cantidad de posibilidades con el auge que está teniendo el Internet of Things. El proyecto podría ser útil para conocer mejor el uso de los equipos fitness al aire libre y también para poder llevar un control del mantenimiento de éstos. Se podría saber con el dispositivo qué máquinas son las que más se utilizan y las que menos. En cuanto a este último aspecto, el mantenimiento, el dispositivo podría tener una amplia línea de negocio para las empresas que gestionan el mantenimiento de estos equipos deportivos, como pueden ser los ayuntamientos o las empresas

específicas de cada equipo.

En definitiva, pienso que los resultados del proyecto han sido muy positivos y se han cumplido todos los requisitos u objetivos propuestos en el plazo previsto. Realmente espero que este proyecto sirva para posibles mejoras futuras y se pueda hacer un correcto uso de éste. Con este último fin se ponen a disposición todos los archivos y documentos utilizados en la elaboración del proyecto en un repositorio Github de público acceso en el siguiente enlace: [Github](#).

Bibliografía

- [1] Hsueh-wen Chow, Andrew J. Mowen y Guan-lin Wu. «Who Is Using Outdoor Fitness Equipment and How? The Case of Xihu Park». En: *International Journal of Environmental Research and Public Health* 14.4 (2017). ISSN: 1660-4601. DOI: 10.3390/ijerph14040448. URL: <https://www.mdpi.com/1660-4601/14/4/448>.
- [2] Diana C. Parra, Thomas L. McKenzie, Isabela C. Ribeiro, Adriano A. Ferreira Hino, Mariah Dreisinger, Kathryn Coniglio, Marcia Munk, Ross C. Brownson, Michael Pratt, Christine M. Hoehner y Eduardo J. Simoes. «Assessing Physical Activity in Public Parks in Brazil Using Systematic Observation». En: *American Journal of Public Health* 100.8 (2010). PMID: 20558792, págs. 1420-1426. DOI: 10.2105/AJPH.2009.181230. eprint: <https://doi.org/10.2105/AJPH.2009.181230>. URL: <https://doi.org/10.2105/AJPH.2009.181230>.
- [3] C. -T. Hsu, Y. -H. Chang, J. -S. Chen, H. -H. Lin y J. -Y. Chou. «Implementation of IoT Device on Public Fitness Equipment for Health Physical Fitness Improvement». En: *2020 International Conference on Mathematics and Computers in Science and Engineering (MACISE)*. 2020, págs. 236-239. DOI: 10.1109/MACISE49704.2020.00050.
- [4] Cori Sweet, Soghra Jarvandi, Heather Sedges y Karen Franck. «Activity Tracker for Assessing Outdoor Fitness Equipment Use». En: *Journal of Park and Recreation Administration* 0.0 (2020). ISSN: 2160-6862. DOI: 10.18666/JPRA-2020-10433. URL: <https://js.sagamorepub.com/jpra/article/view/10433>.
- [5] Yue Han, Xiyin Gao y Ke Su. «The Interaction Design of Outdoor Fitness Facilities Based on Solar Energy». En: *Proceedings of the 2016 International Conference on Energy, Power and Electrical Engineering*. Atlantis Press, 2016, págs. 6-9. ISBN: 978-94-6252-258-9. DOI: <https://doi.org/10.2991/epee-16.2016.2>. URL: <https://doi.org/10.2991/epee-16.2016.2>.
- [6] Xiyin Gao, Chunrong Zhao y Yue Han. «The Research of Outdoor Fitness Facilities Design Based on Kano Model». En: *Proceedings of the 2017 2nd International Conference on Automation, Mechanical and Electrical Engineering (AMEE 2017)*. Atlantis Press, 2017, págs. 233-237. ISBN: 978-94-6252-393-7. DOI: <https://doi.org/10.2991/amee-17.2017.48>. URL: <https://doi.org/10.2991/amee-17.2017.48>.

- [7] Zheng Weitao, Yi Mingnong, Wang Jian, Peng Liming, Hu Pu, Hong Wei, Zhang Jian, Ma Yong, Liu Jia, Liu Binkui y et al. *Monitoring device for outdoor bodybuilding equipment*. N°Patente: CN104850942B. Jun. de 2018.
- [8] Sun Jia, Wu Tao y Yu Lei. *Microwave radar device for outdoor fitness equipment*. N°Patente: CN110119114A. Ago. de 2019.
- [9] Hu Jianchao. *Outdoor Fitness Equipment Auxiliary Device*. N°Patente: WO2019241954A1. Dic. de 2019.
- [10] Zhang Peifei. *Outdoor intelligent body building station*. N° Patente: CN104147772A. Wuhan Haokang Fitness Equipment CO LTD, nov. de 2014.
- [11] Shi Danwei, Li Chuanbing, Lu Xianyou y Huang Yongchun. *Outdoor fitness equipment with intelligent action recognition function*. N°Patente: CN110141830A. Shenzhen Taishan Sports Tech Corp LTD y Shandong Taishan Sports Tech Co LTD, ago. de 2019.
- [12] *TGO Activate*. The Great Outdoor Gym Company. URL: <https://www.tgogc.com/tgo-activate-app>.
- [13] *Norwell Outdoor Fitness v2*. Norwell Outdoor Fitness. URL: <https://www.norwelloutdoorfitness.com/norwell-app>.
- [14] *Proludic Sport*. Proludic. URL: <https://www.proludic.co.uk/news/video-gallery/proludic-sport-app/>.
- [15] *Caloo Pulse App*. Caloo. URL: <https://www.caloo.co.uk/blog/caloo-pulse-app/>.
- [16] *LPA Entrénate*. Ayuntamiento de Las Palmas de Gran Canaria. URL: <https://apps.apple.com/es/app/lpa-entrenate/id1058771649>.
- [17] K. S. Choi, Y. S. Joo y S. Kim. «Automatic exercise counter for outdoor exercise equipment». En: *2013 IEEE International Conference on Consumer Electronics (ICCE)*. 2013, págs. 436-437. DOI: 10.1109/ICCE.2013.6486965.
- [18] H-w Chow y C-H Ho. «Does the use of outdoor fitness equipment by older adults qualify as moderate to vigorous physical activity?» En: *PLoS ONE* 13(4): e0196507. 2018. DOI: 10.1371/journal.pone.0196507.
- [19] Hsueh-wen Chow, Andrew J. Mowen y Guan-lin Wu. «Who Is Using Outdoor Fitness Equipment and How? The Case of Xihu Park». En: *International Journal of Environmental Research and Public Health* 14.4 (2017). ISSN: 1660-4601. DOI: 10.3390/ijerph14040448. URL: <https://www.mdpi.com/1660-4601/14/4/448>.
- [20] Leonie Cranney, Philayrath Phongsavan, Maina Kariuki, Vicki Stride, Ashleigh Scott, Myna Hua y Adrian Bauman. «Impact of an outdoor gym on park users' physical activity: A natural experiment». En: *Health Place* 37 (2016), págs. 26-34. ISSN: 1353-8292. DOI: <https://doi.org/10.1016/j.healthplace.2015.11.002>. URL: <http://www.sciencedirect.com/science/article/pii/S1353829215001495>.

BIBLIOGRAFÍA

- [21] Hw. Chow. «Outdoor fitness equipment in parks: a qualitative study from older adults' perceptions». En: *BMC Public Health* 13 (2013). ISSN: 1216. DOI: 10.1186/1471-2458-13-1216.
- [22] Cai Xuefeng y Ren Na. *Method for detecting outdoor sports equipment and terminal equipment*. N°Patente: CN109765821A. Mayo de 2019.
- [23] Zhang Wenxiang, Zeng Xinhuang y Liu Chunchun. *Power supplying and data collecting method for outdoor fitness equipment*. N°Patente: CN107185166A. Sep. de 2017.
- [24] Jin Jinag; Fan Xiaomin; Wu Aijun. *Intelligent sunshade ceiling with outdoor body-building apparatus cooperation use*. N°Patente: CN208267432U. Dic. de 2018.
- [25] E. Hernández Aparicio. «ESTUDIO DE LOS CIRCUITOS BIOSALUDABLES PARA LA TERCERA EDAD EN ESPAÑA». Español. En: *Revista Internacional de Medicina y Ciencias de la Actividad Física y del Deporte / International Journal of Medicine and Science of Physical Activity and Sport* (2009). ISSN: 1577-0354. URL: <https://www.redalyc.org/articulo.oa?id=54222999003>.
- [26] AENOR. *Equipos fijos de entrenamiento físico instalados al aire libre. Requisitos de seguridad y métodos de ensayo*. Inf. téc. UNE-EN 16630:2015. Madrid: AENOR, 2015.
- [27] ASTM. *Standard Specification for Unsupervised Public Use Outdoor Fitness Equipment*. Inf. téc. ASTM F3101-15. Estados Unidos: ASTM, 2015.
- [28] Sigfox España. *Resumen de la tecnología*. Accedido: 01-02-2021. URL: <https://www.sigfox.es/%C2%BFque-es-iot>.
- [29] Kais Mekki, Eddy Bajic, Frederic Chaxel y Fernand Meyer. «A comparative study of LPWAN technologies for large-scale IoT deployment». En: *ICT Express* 5.1 (2019), págs. 1-7. ISSN: 2405-9595. DOI: <https://doi.org/10.1016/j.icte.2017.12.005>. URL: <http://www.sciencedirect.com/science/article/pii/S2405959517302953>.
- [30] J. Lin, T. Talty y O. K. Tonguz. «On the potential of bluetooth low energy technology for vehicular applications». En: *IEEE Communications Magazine* 53.1 (2015), págs. 267-275. DOI: 10.1109/MCOM.2015.7010544.
- [31] Jobted. *Sueldos medios en España: Jobted*. URL: <https://www.jobted.es/salario/ingeniero-electr%C3%B3nico>.

ANEXOS

Anexo A

Código del dispositivo maestro

En el siguiente código se pueden ver las librerías añadidas y todas las variables globales.

```
//LIBRERIAS
2 #include <Wire.h>
# include <ADXL345_WE.h>
4 #include <Arduino.h>
# include <bluefruit.h>
6 #include <Adafruit_LittleFS.h>
# include <InternalFileSystem.h>
8 #include <lmic.h>
# include <hal/hal.h>
10 #include <SPI.h>
# include "NRF52TimerInterrupt.h"
12 #include "NRF52_ISR_Timer.h"
# include <stdlib.h>
14
//////VARIABLES E INSTANCIAS //////
16 #define ADXL345_I2CADDR 0x53
# define TIMER_INTERRUPT_DEBUG 0
18 #define _TIMERINTERRUPT_LOG_LEVEL_ 0
# define PININT1 14
20
NRF52Timer ITimer0 (NRF_TIMER_1);
22 NRF52_ISR_Timer ISR_Timer0;

24 NRF52Timer ITimer1 (NRF_TIMER_3);
NRF52_ISR_Timer ISR_Timer1;
26
ADXL345_WE myAcc(ADXL345_I2CADDR);

28 unsigned long temporizador_predefinido = 400;
30 volatile uint32_t temporizador;
volatile uint32_t temporizador_variable;
32 unsigned long tiempo_reinicio = 600;
volatile uint8_t tiempo_espera = 20; //Tiempo en el que estara el
dispositivo buscando nodos para configurarlos.
34 unsigned long tiempo;
```

```

36 volatile uint8_t estado = 1;
37 typedef struct
38 {
39     char name[16+1];
40     uint16_t conn_handle;
41     BLEClientUart clientUart;
42 }prph_info_t;
43
44 prph_info_t prphs[3];
45
46 volatile bool power = true;
47 int cont = 0;
48 volatile int posicion = 0;
49 uint8_t datos = 0;
50 unsigned long tiempoinicio;
51 unsigned long tiempofin;
52
53 void registro_dato();
54
55
56 SoftwareTimer blinkTimer;
57
58 volatile bool parar_lorawan = false;
59 volatile bool parar_ble = false;
60 volatile bool salir_estado5 = false;
61 volatile bool salir_ble1 = false;
62 volatile bool salir_lorawan1 = false;
63 volatile bool reinicio = false;
64
65
66
67
68 String datoscentral = "Central";
69 char datos1[100];
70 char datos2[100];
71 char datos3[100];
72 char datos4[100];
73
74 volatile uint8_t contador = 0;
75
76 volatile uint8_t tiempo_inactividad = 4;
77 static const u1_t PROGMEM APPEUI[8]={0x3F,0x19,0x04,0xD0,0x7E,0xD5,0xB3
78 ,0x70};
79 void os_getArtEui (u1_t* buf) { memcpy_P(buf, APPEUI, 8);}
80
81 static const u1_t PROGMEM DEVEUI[8]={0x3E,0x4A,0x75,0xFE,0xFF,0xA8,0xC5
82 ,0x60};
83 void os_getDevEui (u1_t* buf) { memcpy_P(buf, DEVEUI, 8);}
84
85 static const u1_t PROGMEM APPKEY[16] = { 0xDC,0x20,0x25,0xE8,0xCA,0xD9
86 ,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00}
```

ANEXO A. CÓDIGO DEL DISPOSITIVO MAESTRO

```
86     ,0x0E,0x4A,0xF7,0x96,0xB0,0x9E,0xC1,0x82,0x38,0x0E };
```

```
87 void os_getDevKey ( u1_t* buf) { memcpy_P(buf, APPKEY, 16);}
```

```
88 volatile uint32_t lora_count = 0;
```

```
89 static osjob_t sendjob;
```

```
90 char Datos_conf[50] = "Configuracion";
```

```
92 uint8_t N = 10;
```

```
93 int Q;
```

```
94 volatile uint8_t cont1 = 0;
```

```
95 volatile uint8_t variable = 0;
```

```
96 typedef enum {
```

```
97     LORA_UNCONFIRMED,
```

```
98     LORA_CONFIRMED
```

```
99 };
```

```
102 const unsigned TX_INTERVAL = 10;
```

```
104 const lmic_pinmap lmic_pins = {
```

```
105     .nss = 4,
```

```
106     .rxtx = LMIC_UNUSED_PIN,
```

```
107     .rst = LMIC_UNUSED_PIN,
```

```
108     .dio = {27, 28, 29},
```

```
109 };
```

Listing A.1: Código del dispositivo maestro. Fuente: elaboración propia.

La siguiente parte del código muestra una serie de funciones para crear variables de tamaño dinámico.

```
2 ///////////////////////////////////////////////////////////////////VARIABLES DINAMICAS////////////////////////////////////////////////////////////////
```

```
4
```

```
6 int* datostiempos;
```

```
7 size_t counta;
```

```
8 size_t capacity;
```

```
10 void reinicio_datos()
```

```
{
```

```
12     Serial.println("Valores reiniciados");
```

```
13     datoscentral = "";
```

```
14     datoscentral = "Central";
```

```
15     datos = 0;
```

```
16     cont = 0;
```

```
17     posicion = 0;
```

```
18     memset(datos2, 0, sizeof(datos2));
```

```
}
```

```
20 void convertir_datos()
```

```
{
```

```
24     Serial.println("Datos convertidos");
25     int datos_len = datoscentral.length() + 1;
26     datoscentral.toCharArray(datos1, 100);
27 }
28 void CreateList(size_t _capacity)
29 {
30     datostiempo = new int[_capacity];
31     capacity = _capacity;
32     counta = 0;
33 }
34 void AddItem(int item)
35 {
36     ++counta;
37     if(counta > capacity)
38     {
39         size_t newSize = capacity*2;
40         resize(newSize);
41     }
42     datostiempo[counta - 1] = item;
43 }
44 void RemoveTail()
45 {
46     --counta;
47 }
48
49 void Trim()
50 {
51     resize(counta);
52 }
53
54 }
55 void resize(size_t newCapacity)
56 {
57     int* newList = new int[newCapacity];
58     memmove(newList, datostiempo, counta * sizeof(int));
59     delete [] datostiempo;
60     capacity = newCapacity;
61     datostiempo = newList;
62 }
63
64 void PrintList()
65 {
66     Serial.print("Datos Tiempos: ");
67     for (size_t index = 0; index < counta; index++)
68     {
69         Serial.print(datostiempo[index]);
70         Serial.print(' ');
71     }
72     Serial.println();
73 }
74 }
```

ANEXO A. CÓDIGO DEL DISPOSITIVO MAESTRO

Listing A.2: Código del dispositivo maestro (continuación del listing A.1). Fuente: elaboración propia.

Parte de código relacionada con la configuración del temporizador.

```
1 ///////////////////////////////////////////////////TEMPORIZADOR/////////////////////////////////////////////////
3 void funcion_reinicio()
{
5   Serial.println("REINICIO");
7   reinicio = true;
9   estado = 10;
11  NVIC_SystemReset();
13
15 void funcion_tempo(TimerHandle_t xTimerID)
17 {
19   (void) xTimerID;
21   Serial.println("TIEMPO");
23   salir_estado5 = true;
25   estado = 6;
27   Serial.println("Parar temporizador");
29
31 void tiempo_de_espera()
33 {
```

Listing A.3: Continuación del listing A.2. Fuente: elaboración propia.

La siguiente parte de código representa todas las funciones relacionadas con el protocolo LoRaWAN.

```
1 //////////////////////////////////LORAWAN/////////////////////////////////////////////////
3 void apagar_lorawan()
{
5   os_clearCallback(&sendjob);
7   LMIC_shutdown();
9   Serial.println("LoRaWAN apagado");
11  parar_lorawan = true;
```

```

9|     if (estado == 1){estado = 2;}
10|    if (estado == 7){estado = 8; salir_lorawan1 = true;}
11}
13
15
16 void envio_lorawan_conf(osjob_t* j)
17{
18    if (LMIC.opmode & OP_TXRXPEND)
19    {
20        Serial.println(F("OP_TXRXPEND, not sending"));
21    }
22    else
23    {
24        LMIC_setTxData2(1, (unsigned char *) Datos_conf, strlen(Datos_conf)
25        , LORA_CONFIRMED);
26        Serial.println("Dato enviado a LoRaWAN");
27        contador++;
28
29        if (contador == 5 && temporizador == 0)
30        {
31            Serial.println("No se ha establecido conexión con LoRaWAN");
32            Serial.println("Se establece el temporizador predeterminado de
33            400 segundos");
34            temporizador = temporizador_predefinido;
35            apagar_lorawan();
36        }
37    }
38}
39
40 void envio_lorawan_datos(osjob_t* j)
41{
42    switch(variable)
43    {
44        case 0:
45            Serial.println("CASO 0");
46            convertir_datos();
47            if (strlen(datos1) > 1)
48            {
49                Serial.println("Datos1 hay datos");
50                LMIC_setTxData2(1, (unsigned char *)datos1, strlen(datos1),
51                LORA_CONFIRMED);
52                Serial.println(F("Paquete enviado 1"));
53                variable++;
54            }
55        else
56        {
57            Serial.println("No hay datos1");
58            variable = 0;
59        }
60    }
61    break;

```

ANEXO A. CÓDIGO DEL DISPOSITIVO MAESTRO

```
59|     case 1:
60|         Serial.println("CASO 1");
61|         if(strlen(datos2) > 1 && *datos2 != '0')
62|         {
63|             LMIC_setTxData2(1, (unsigned char *)datos2, strlen(datos2),
64| LORA_CONFIRMED);
65|             Serial.println(F("Paquete enviado 2"));
66|             variable++;
67|         }
68|     else
69|     {
70|         Serial.println("No hay m s datos para enviar por LoRaWAN");
71|         variable = 0;
72|         apagar_lorawan();
73|         break;
74|     }
75|     break;
76|
77|     case 2:
78|
79|         Serial.println("CASO 2");
80|         if(strlen(datos3) > 1 && *datos3 != '0')
81|         {
82|             LMIC_setTxData2(1, (unsigned char *)datos3, strlen(datos3),
83| LORA_CONFIRMED);
84|             Serial.println(F("Paquete enviado 3"));
85|             variable++;
86|         }
87|     else
88|     {
89|         Serial.println("No hay m s datos para enviar por LoRaWAN");
90|         variable = 0;
91|         apagar_lorawan();
92|         break;
93|     }
94|     break;
95|
96|     case 3:
97|
98|         Serial.println("CASO 3");
99|         if(strlen(datos4) > 1 && *datos4 != '0')
100|         {
101|             LMIC_setTxData2(1, (unsigned char *)datos4, strlen(datos4),
102| LORA_CONFIRMED);
103|             Serial.println(F("Paquete enviado 4"));
104|             variable = 0;
105|         }
106|     else
107|     {
108|         Serial.println("No hay m s datos para enviar por LoRaWAN");
109|         variable = 0;
110|         apagar_lorawan();
111|         break;
112|     }
113| }
```

```
109     }
110     break;
111 }
112 }
113 }
114
115
116
117
118
119 void onEvent (ev_t ev) {
120
121     switch(ev) {
122         case EV_SCAN_TIMEOUT:
123             Serial.println(F("EV_SCAN_TIMEOUT"));
124             break;
125         case EV_BEACON_FOUND:
126             Serial.println(F("EV_BEACON_FOUND"));
127             break;
128         case EV_BEACON_MISSED:
129             Serial.println(F("EV_BEACON_MISSED"));
130             break;
131         case EV_BEACON_TRACKED:
132             Serial.println(F("EV_BEACON_TRACKED"));
133             break;
134         case EV_JOINING:
135
136             Serial.println(F("Comenzando a Unirse..."));
137             break;
138         case EV_JOINED:
139             Serial.println(F("Enlazado con LoraWAN"));
140             {
141                 u4_t netid = 0;
142                 devaddr_t devaddr = 0;
143                 u1_t nwkKey[16];
144                 u1_t artKey[16];
145                 LMIC_getSessionKeys(&netid, &devaddr, nwkKey, artKey);
146                 Serial.println("Unido a LoraWAN");
147                 Serial.print("netid: ");
148                 Serial.println(netid, DEC);
149                 Serial.print("devaddr: ");
150                 Serial.println(devaddr, HEX);
151                 Serial.print("artKey: ");
152                 for (size_t i=0; i<sizeof(artKey); ++i) {
153                     Serial.print(artKey[i], HEX);
154                 }
155                 Serial.println("");
156                 Serial.print("nwkKey: ");
157                 for (size_t i=0; i<sizeof(nwkKey); ++i) {
158                     Serial.print(nwkKey[i], HEX);
159                 }
160                 Serial.println("");
161 }
```

```

163         lora_count = 1;
165     }
166
167     LMIC_setLinkCheckMode(0);
168     break;
169
170     case EV_JOIN_FAILED:
171         Serial.println(F("EV_JOIN_FAILED"));
172         break;
173     case EV_REJOIN_FAILED:
174         Serial.println(F("EV_REJOIN_FAILED"));
175         break;
176     case EV_TXCOMPLETE:
177
178         if (estado == 1)
179         {
180             if (LMIC.dataLen == 1)
181             {
182                 temporizador = LMIC.frame[LMIC.dataBeg + 0];
183                 Serial.println("Dato de temporizador recibido");
184                 Serial.print("Tiempo de temporizador: ");
185                 Serial.println(temporizador);
186
187                 apagar_lorawan();
188                 break;
189             }
190             else
191             {
192
193                 if (LMIC.txrxFlags & TXRX_ACK)
194                 {
195                     Serial.println(F("Recibida Confirmacion \r\n"));
196                 }
197
198                 if (LMIC.dataLen)
199                 {
200                     Serial.print(F(" Received "));
201                     Serial.print(LMIC.dataLen);
202                     Serial.println(F(" bytes of payload"));
203
204                 }
205             }
206
207             break;
208         case EV_LOST_TSYNC:
209             Serial.println(F("EV_LOST_TSYNC"));
210             break;
211         case EV_RESET:
212             Serial.println(F("EV_RESET"));
213             break;
214         case EV_RXCOMPLETE:

```

```

215     Serial.println(F("EV_RXCOMPLETE"));
217     break;
218 case EV_LINK_DEAD:
219     Serial.println(F("EV_LINK_DEAD"));
220     break;
221 case EV_LINK_ALIVE:
222     Serial.println(F("EV_LINK_ALIVE"));
223     break;

225 case EV_TXSTART:

227     if (estado == 1)
228     {
229         os_setTimedCallback(&sendjob, os_getTime() + sec2osticks(
230             TX_INTERVAL), envio_lorawan_conf);
231     }
232     if (estado == 7)
233     {
234         os_setTimedCallback(&sendjob, os_getTime() + sec2osticks(
235             TX_INTERVAL), envio_lorawan_datos);
236     }
237     break;
238 default:
239     Serial.print(F("Unknown event: "));
240     Serial.println((unsigned) ev);
241     break;
242 }

243

245 void lorawan_conf()
246 {
247     os_init();
248     LMIC_reset();
249     if (estado == 1){envio_lorawan_conf(&sendjob);}
250     if (estado == 7){envio_lorawan_datos(&sendjob);}
251     while(1)
252     {
253         if (parar_lorawan == true) {break;}
254         os_runloop_once();
255     }
256 }

257

258 }
```

Listing A.4: Continuación del listing A.3. Fuente: elaboración propia.

El siguiente fragmento del código representa toda la parte de la configuración y ejecución para el Bluetooth Low Energy (BLE).

1 ////////////////BLUETOOTH/////////////

```

3   void scan_callback(ble_gap_evt_adv_report_t* report)
4   {
5     Serial.println("Detectado dispositivo");
6     Serial.println("Conectando...");
7     Bluefruit.Central.connect(report);
8   }
9
10 int findConnHandle( uint16_t conn_handle )
11 {
12   for( int id=0; id<3; id++)
13   {
14     if (conn_handle == prphs[id].conn_handle)
15     {
16       return id;
17     }
18   }
19   return -1;
20 }
21
22 void conexion_callback( uint16_t conne_handle )
23 {
24   int id = findConnHandle(BLE.CONN_HANDLE_INVALID);
25   if(id < 0) return;
26   prph_info_t* peer = &prphs[id];
27   peer->conn_handle = conne_handle;
28   Bluefruit.Connection(conne_handle)->getPeerName( peer->name, sizeof(
29     peer->name)-1 );
30   Serial.print("Conectado a ");
31   Serial.println(peer->name);
32   if(prphs[id].clientUart.discover(conne_handle))
33   {
34     prphs[id].clientUart.enableTXD();
35     Bluefruit.Scanner.resume();
36   }
37   else
38   {
39     Bluefruit.disconnect(conne_handle);
40   }
41 }
42
43 void desconexion_callback( uint16_t conn_handle , uint8_t reason )
44 {
45   (void) conn_handle;
46   (void) reason;
47   int id = findConnHandle(conn_handle);
48   if(id<0) return;
49   prphs[id].conn_handle = BLE.CONN_HANDLE_INVALID;
50   Serial.print(prphs[id].name);
51   Serial.println(" desconectado");
52 }
53

```

```

55 | void bleuart_rx_callback(BLEClientUart& uart_svc)
56 | {
57 |     if (estado == 6)
58 |     {
59 |         Serial.println("Recibiendo dato por BLE");
60 |         while (uart_svc.available())
61 |         {
62 |             Serial.println("");
63 |             Serial.print("Dato recibido: ");
64 |             if (uart_svc.read(datos2, sizeof(datos2)-1))
65 |             {
66 |                 Serial.println(datos2);
67 |             }
68 |             if (strlen(datos2) > 1)
69 |             {
70 |                 estado = 7;
71 |                 salir_ble1 = true;
72 |                 break;
73 |             }
74 |         }
75 |     }
76 |
77 |     void config_ble()
78 |     {
79 |         Bluefruit.begin(0, 3);
80 |         Bluefruit.setTxPower(4);
81 |         Bluefruit.setName("Dispositivo Central");
82 |         for (uint8_t idx=0; idx<3;idx++)
83 |         {
84 |             prphs[idx].conn_handle = BLE_CONN_HANDLE_INVALID;
85 |             prphs[idx].clientUart.begin();
86 |             prphs[idx].clientUart.setRxCallback(bleuart_rx_callback);
87 |
88 |         }
89 |         Bluefruit.Central.setConnectCallback(conexion_callback);
90 |         Bluefruit.Central.setDisconnectCallback(desconexion_callback);
91 |         Bluefruit.Scanner.setRxCallback(scan_callback);
92 |         Bluefruit.Scanner.restartOnDisconnect(false);
93 |         Bluefruit.Scanner.filterUuid(BLEUART_UUID_SERVICE);
94 |         Bluefruit.Scanner.setInterval(160, 80);
95 |         Bluefruit.Scanner.useActiveScan(false);
96 |         Serial.println("BLE Configurado");
97 |         if (estado == 2)
98 |         {
99 |             Serial.println("Comienzo de conexión con nodos");
100 |             ITimer1.attachInterruptInterval(tiempo_espera*1000,
101 |                                             temporizador_handler);
102 |
103 |         }
104 |         Bluefruit.Scanner.start(20000);
105 |         Serial.println("Escaneando . . .");

```

```

107 }
109 void sendAll( uint32_t buf)
{
111   Serial.print("Enviando a todos los nodos: ");
112   Serial.println(buf);
113   for( uint8_t id=0; id<3 ;id++)
114   {
115     prph_info_t* peer = &prphs[ id ];
116     if (peer->clientUart .discovered())
117     {
118       delay(1000);
119       peer->clientUart .write( buf );
120     }
121   }
123 }

125 void desconexion_ble()
{
126   for( uint8_t id = 0; id <3; id++)
127   {
128     uint16_t conn_handle = prphs[ id ].clientUart .connHandle();
129     Bluefruit .disconnect( conn_handle );
130   }
131 }
```

Listing A.5: Continuación del listing A.4. Fuente: elaboración propia.

En este fragmento se muestra la función setup(), en la que se configuran las conexiones I^2C y los puertos serie. También se muestra la configuración del acelerómetro.

```

1 ///////////////////////////////////////////////////////////////////SETUP/////////////////////////////////////////////////////////////////
2
4 void setup()
{
5   Wire.begin();
6   Serial.begin(115200);
7   Serial.println("Inicio Dispositivo Central");
8   Serial.println();
9   CreateList(1);
10
11 }
12
13
14
15
16
17
18 void conf_acelerometro()
{
19   if (!myAcc.init()) {
20 }
```

```

    Serial.println("ADXL345 no conectado!");
22 }
myAcc.setCorrFactors(-266.0, 285.0, -268.0, 278.0, -291.0, 214.0);
24 myAcc.setDataRate(ADXL345_DATA_RATE_25); //Se establece el DATA RATE
    del acelerometro en 25 Hz
myAcc.setRange(ADXL345_RANGE_4G); //Se establece el rango de medida
    del acelerometro en 4g.
26 attachInterrupt(digitalPinToInterrupt(PININT1), registro_dato,
    ISR_DEFERRED | RISING); //Se declara la interrupcion por actividad
    del acelerometro.
myAcc.setActivityParameters(ADXL345_AC_MODE, ADXL345_XYZ, 0.8); //Se
    establecen los parametros necesarios para la deteccion de actividad
    (Modo AC, tres ejes, limite de deteccion)
28 myAcc.setInactivityParameters(ADXL345_DC_MODE, ADXL345_XYZ, 1,
    tiempo_inactividad); //Se establecen los parametros necesarios para
    la deteccion de la inactividad (Modo AC, tres ejes, limite de
    deteccion, tiempo para deteccion)
myAcc.setLinkBit(true); //Se configura el Link Bit para que una vez
    detecte actividad no vuelva a detectarla hasta que detecte
    inactividad.
30 myAcc.setInterrupt(ADXL345_ACTIVITY, INT_PIN_1); //Se configura el
    Pin INT1 como la interrupcion de salida del acelerometro para la
    actividad.
myAcc.setInterrupt(ADXL345_INACTIVITY, INT_PIN_1); //Se configura el
    Pin INT1 como la interrupcion de salida del acelerometro para la
    inactividad.
32 Serial.println("Acelerometro configurado");
}

```

Listing A.6: Continuación del listing A.5. Fuente: elaboración propia.

Ahora se muestra en el siguiente fragmento la función en la cual se registra el dato deportivo.

```

2 ///////////////REGISTRO DEL DATO///////////
4
5 void registro_dato()
6 {
7     byte intSource = myAcc.readAndClearInterrupts();
8     if(myAcc.checkInterrupt(intSource, ADXL345_ACTIVITY))
9     {
10         tiempo = 0;
11         datos++;
12         tiempoinicio = millis();
13         Serial.print("ACTIVIDAD ");
14         Serial.println(datos);
15     }
16     if(myAcc.checkInterrupt(intSource, ADXL345_INACTIVITY))
17     {
18         tiempofin = millis();
19         if(datos == 0){Serial.println("INACTIVIDAD");}
20         else

```

ANEXO A. CÓDIGO DEL DISPOSITIVO MAESTRO

```
22  {
23      Serial.println("INACTIVIDAD");
24      tiempo = tiempofin - tiempoinicio;
25      if(tiempo < tiempo_inactividad*1000)
26      {
27          Serial.println(tiempo/1000);
28          datoscentral.concat(";" + String(tiempo/1000));
29          Serial.print("Datos: ");
30          Serial.println(datoscentral);
31      }
32  }  
33  else
34  {
35      tiempo = tiempo - tiempo_inactividad*1000;
36      Serial.println(tiempo/1000);
37      datoscentral.concat(";" + String(tiempo/1000));
38      Serial.print("Datos: ");
39      Serial.println(datoscentral);
40  }
41 }
42 myAcc.readAndClearInterrupts();
```

Listing A.7: Continuación del listing A.6. Fuente: elaboración propia.

Por último, la siguiente parte de código muestra la función principal `loop()`.

```
1
2
3 void loop()
4 {
5     switch(estado)
6     {
7
8         //Estado de configuracion en el que se conecta con lorawan y se
9         //recibe el tiempo del temporizador.
10        case 1:
11
12            Serial.println("Configurando dispositivo");
13            Serial.println("A la espera de conexión con LoRaWAN... ");
14            lorawan_conf();
15            if(reinicio == true){estado = 10;}
16            break;
17
18
19
20
21        //Estado de configuracion en el que se configura el bluetooth y se
22        //comienza a escanear nodos
23        case 2:
```

```
25|     Serial.println("Configurando los dem s nodos....");
26|
27| config_ble();
28| while(1)
29| {
30|     if(parar_ble == true){break;}
31|     sd_app_evt_wait();
32|     if(reinicio == true){estado = 10; break;}
33| }
34| parar_ble = false;
35| break;

36|
37| //Estado de configuracion en el que se env a el tiempo a todos los
38| //nodos conectados.
39| case 3:
40| ITimer1.stopTimer();
41| Serial.println("Tiempo de escaneo superado");
42| Serial.println("Enviando datos a los nodos conectados");
43| sendAll(temporizador);
44| desconexion_ble();
45| uint8_t conexiones;

46|
47| while(1)
48| {
49|     for(uint8_t id = 0; id<3; id++)
50|     {
51|         uint16_t conn_handle = prphs[id].clientUart.connHandle();
52|         if(Bluefruit.connected(conn_handle) == true)
53|         {
54|             conexiones++;
55|         }
56|
57|     }
58|
59|     if(conexiones == 0)
60|     {
61|         estado = 4;
62|         break;
63|     }
64|     else
65|     {
66|         desconexion_ble();
67|     }
68| }
69| if(reinicio == true){break; estado = 10;}
70| break;

71|
72|
73| //Estado en el que se configura el temporizador con el tiempo
74| //recibido por LoRaWAN y se configura tambi n el aceler metro.
75| case 4:
```

ANEXO A. CÓDIGO DEL DISPOSITIVO MAESTRO

```
77     temporizador_variable = temporizador;
78     Serial.print("Temporizador variable: ");
79     Serial.println(temporizador_variable);
80
81     temporizador_variable = temporizador;
82     blinkTimer.begin(temporizador_variable*1000*60, funcion_tempo, NULL,
83     , false);
84     blinkTimer.start();
85     Serial.println("Temporizador iniciado");
86     conf_acelerometro();
87     estado = 5;
88     if(reinicio == true){estado = 10;
89     break;
90
91 //Estado en de normalidad de bajo consumo en el que se est
92 registrando datos hasta que salta el temporizador y se sale de este
93 estado al estado de recepción de los datos de los nodos.
94 case 5:
95
96     while(1)
97     {
98         if(salir_estado5 == true){break;}
99         sd_app_evt_wait();
100        if(reinicio == true){estado = 10; break;}
101
102    }
103    salir_estado5 = false;
104    break;
105
106 //Estado en el que se configura el bluetooth para recibir los datos
107 de los nodos, se conecta a los nodos y se reciben los datos.
108 case 6:
109
110     Serial.println("Final de toma de datos");
111     Serial.println("Configurando BLE");
112
113     config_ble();
114     while(1)
115     {
116         if(salir_ble1 == true){break;}
117         sd_app_evt_wait();
118         if(reinicio == true){estado = 10; break;}
119     }
120     salir_ble1 = false;
121     break;
122
123 //Estado en el que se envían los datos recibidos por LoRaWAN
124 case 7:
125     Serial.println("Enviando datos a LoRaWAN... ");
126     parar_lorawan = false;
```

```
125    lorawan_conf();
126    while(1)
127    {
128        if(salir_lorawan1 == true){break;}
129        sd_app_evt_wait();
130        if(reinicio == true){estado = 10; break;}
131    }
132    salir_lorawan1 = false;
133    break;
134
135
136 //Estado en el que se desconectan todos los nodos.
137 case 8:
138
139     Serial.println("Desconectando todos los nodos....");
140     desconexion_ble();
141     conexiones = 0;
142     while(1)
143     {
144         for(uint8_t id = 0; id<3; id++)
145         {
146             uint16_t conn_handle = prphs[id].clientUart.connHandle();
147             if(Bluefruit.connected(conn_handle) == true)
148             {
149                 conexiones++;
150             }
151         }
152     }
153
154     if(conexiones == 0)
155     {
156         estado = 9;
157         break;
158     }
159     else
160     {
161         desconexion_ble();
162     }
163
164 }
165
166 if(reinicio == true){estado = 10;}
167 break;
168
169
170 //Estado en el que se reinician todos los valores de las variables
171 y el temporizador.
172 case 9:
173
174     reinicio_datos();
175
blinkTimer.begin(temporizador_variable*1000, funcion_tempo, NULL,
```

ANEXO A. CÓDIGO DEL DISPOSITIVO MAESTRO

```
177     false);
178     blinkTimer.start();

179     Serial.println("Temporizador iniciado");
180     estado = 5;
181     if(reinicio == true){estado = 10;}
182     break;

183     //Estado en el que se reinicia el dispositivo para comenzar una
184     //nueva configuración.
185     case 10:
186     Serial.println("REINICIO");
187     NVIC_SystemReset();
188     break;

189 }

190 }

191 }

192 }
```

Listing A.8: Continuación del listing A.7. Fuente: elaboración propia.

Anexo B

Código del dispositivo nodo

En el siguiente código se pueden ver las librerías añadidas y todas las variables globales para el dispositivo nodo.

```
//////////LIBRERIAS, VARIABLES E INSTANCIAS//////////
2 #include <Wire.h>
3 #include <ADXL345_WE.h>
4 #include <Arduino.h>
5 #include <bluefruit.h>
6 #include <Adafruit_LittleFS.h>
7 #include "NRF52TimerInterrupt.h"
8 #include "NRF52_ISR_Timer.h"
9 #define TIMER_INTERRUPT_DEBUG 0
10 #define _TIMERINTERRUPT_LOG_LEVEL_ 0
11 #define ADXL345_I2CADDR 0x53
12 #define PININT1 14

14
15 ADXL345_WE myAcc(ADXL345_I2CADDR);
16 BLEUart bleuart;
17 NRF52Timer ITimer0 (NRF_TIMER_1);
18 NRF52_ISR_Timer ISR_Timer1;
19 SoftwareTimer blinkTimer;
20 volatile uint8_t datos = 0;
21 void registro_dato();

22 volatile bool salir_ble = false;
23 volatile bool salir_recibirdato = false;
24 volatile bool salir_estado4 = false;
25 volatile bool salir_ble1 = false;
26 volatile bool salir_desconexion = false;
27 volatile bool reinicio = false;

28 volatile uint8_t estado = 1;
29 volatile uint8_t temporizador;
30 unsigned long tiempoinicio;
31 unsigned long tiempofin;
32 unsigned long tiempo;
33 uint8_t tiempo_inactividad = 5;
```

ANEXO B. CÓDIGO DEL DISPOSITIVO NODO

```
36| unsigned long tiempo_reinicio = 600;  
38|  
39| String datos1 = "nodo1";  
40| char datos2[100];
```

Listing B.1: Código del dispositivo nodo. Fuente: elaboración propia.

En la siguiente parte del código se muestran varias funciones con las que poder generar variables de tamaño dinámico.

```
1 ///////////////////////////////////////////////////////////////////VARIABLES DINAMICAS/////////////////////////////////////////////////////////////////  
3  
5 int * datostiempos;  
5 size_t counta;  
5 size_t capacity;  
7  
9 void reinicio_datos()  
{  
11    datos1 = "";  
11    datos1 = "nodo1";  
13    datos2[0] = '\0';  
13    datos = 0;  
15    Serial.println("Datos reiniciados");  
17}  
17  
19 void convertir_datos()  
{  
21    int datos_len = datos1.length() + 1;  
21    datos1.toCharArray(datos2, 100);  
21    Serial.println("Datos convertidos");  
23}  
25  
27 void CreateList(size_t _capacity)  
{  
29    datostiempos = new int[_capacity];  
29    capacity = _capacity;  
29    counta = 0;  
31}  
33  
33 void AddItem(int item)  
{  
35    ++counta;  
35    if(counta > capacity)  
37    {  
39        size_t newSize = capacity * 2;  
39        resize(newSize);  
39    }  
41    datostiempos[counta - 1] = item;  
43}  
43 void RemoveTail()
```

```

45     {
46         —counta;
47     }
48     void Trim()
49     {
50         resize(counta);
51     }
52     void resize(size_t newCapacity)
53     {
54         int* newList = new int[newCapacity];
55         memmove(newList, datostiempos, counta * sizeof(int));
56         delete[] datostiempos;
57         capacity = newCapacity;
58         datostiempos = newList;
59     }
60     void PrintList()
61     {
62         Serial.print("Datos Tiempos: ");
63         for (size_t index = 0; index < counta; index++)
64         {
65             Serial.print(datostiempos[index]);
66             Serial.print(' ');
67         }
68         Serial.println();
69     }
70 }
71 }
```

Listing B.2: Continuación del listing B.1. Fuente: elaboración propia

El siguiente fragmento de código muestra toda la parte relacionada con el temporizador.

```

1 ////////////////TEMPORIZADOR/////////////
3
4 void funcion_reinicio()
5 {
6     Serial.println("REINICIO");
7     reinicio = true;
8     estado = 8;
9     NVIC_SystemReset();
10 }
11
12 void funcion_tempo(TimerHandle_t xTimerID)
13 {
14     (void) xTimerID;
15     if (estado == 4)
16     {
17         salir_estado4 = true;
18         estado = 5;
```

ANEXO B. CÓDIGO DEL DISPOSITIVO NODO

```
19    }
20}
21 void temporizador_handler()
22{
23    if (estado == 3)
24    {
25        ISR_Timer1.run();
26        ISR_Timer1.setInterval(tiempo_reinicio*1000, funcion_reinicio);
27    }
28}
```

Listing B.3: Continuación del listing B.2. Fuente: elaboración propia.

Este fragmento de código muestra toda la parte relacionada con el Bluetooth Low Energy (BLE) para el dispositivo nodo.

```
2 ///////////////////////////////////////////////////////////////////BLUETOOTH/////////////////////////////////////////////////////////////////
3 void conexion_callback(uint16_t conn_handle)
4{
5    Serial.println("Conectado");
6    Bluefruit.Advertising.stop();
7    BLEConnection* connection = Bluefruit.Connection(conn_handle);
8
9    if (estado == 1)
10    {
11        salir_ble = true;
12        estado = 2;
13    }
14
15    if (estado == 5)
16    {
17        delay(3000);
18        convertir_datos();
19        Serial.print("Datos enviados: ");
20        Serial.println(datos2);
21        bleuart.write(datos2);
22
23    }
24
25}
26
27 void desconexion_callback(uint16_t conn_handle, uint8_t reason)
28{
29    (void) conn_handle;
30    (void) reason;
31    Serial.println("Desconectado del maestro");
32    if (estado == 2)
33    {
34        salir_recibirdato = true;
35        estado = 3;
36    }
37    if (estado == 5)
```

```

40     {
41         salir_ble1 = true;
42         estado = 6;
43     }
44 }
45 void bleuart_rx_callback(uint16_t conn_hdl)
46 {
47     (void) conn_hdl;
48     Serial.println("Recibiendo datos... ");
49     uint8_t* prueba;
50     while(bleuart.available())
51     {
52         *prueba = bleuart.read();
53         temporizador = *prueba;
54         Serial.print("El dato recibido es: ");
55         Serial.println(temporizador);
56         if (temporizador != 0)
57         {
58             break;
59         }
60     }
61 }
62 void startAdv()
63 {
64     Bluefruit.Advertising.addFlags(
65         BLE_GAP_ADV_FLAGS_LE_ONLY_GENERAL_DISC_MODE);
66     Bluefruit.Advertising.addTxPower(); //Se añade la potencia que
67     //tendra el paquete de Advertising
68     Bluefruit.Advertising.addService(bleuart); //Se añade el servicio al
69     //paquete de advertising
70     Bluefruit.ScanResponse.addName(); //Se añade el nombre al proceso de
71     //advertising
72     Bluefruit.Advertising.restartOnDisconnect(true); //Se activa el auto
73     //advertising en caso de desconexion
74     Bluefruit.Advertising.setInterval(32, 244); //Se configura el
75     //intervalo del advertising
76     Bluefruit.Advertising.setFastTimeout(30);
77     Bluefruit.Advertising.start(0); //0 significa que no para de realizar
78     //el advertising despues de n segundos.
79     Serial.println("Advertising... ");
80 }
81 void config_ble()
82 {
83     Bluefruit.autoConnLed(false); //Se desconecta el led de conexion
84     Bluefruit.configPrphBandwidth(BANDWIDTHMAX); //Se configura el
85     //periferico con el maximo ancho de banda
86     Bluefruit.begin(); //Comienza la configuracion de BLE
87     Bluefruit.setTxPower(4); //Se configura la potencia de transmision a

```

ANEXO B. CÓDIGO DEL DISPOSITIVO NODO

```
4
84 Bluefruit.setName("nodo1"); //Se configura el nombre con el que se
     mostrara
bleuart.setRxCallback(bleuart_rx_callback);
86 BluefruitPeriph.setConnectCallback(conexion_callback); //Se indica
     la funcion a la que se ira el programa cuando se realice la
     conexion
BluefruitPeriph.setDisconnectCallback(desconexion_callback); //Se
     indica la funcion a la que se ira el programa cuando se realice la
     desconexion.
88 bleuart.begin(); //Comienza el servicio de BleUart.
Serial.println("BLE configurado");
90 startAdv(); //Comienza el proceso de advertising.
}
92
93 void enviar_datos_ble()
94 {
95     delay(3000);
96     convertir_datos();
97     bleuart.print(datos2);
98     Serial.print("Datos enviados: ");
99     Serial.println(datos2);
100 }
```

Listing B.4: Continuación del listing B.3. Fuente: elaboración propia.

Ahora se representa la parte de la configuración del código, la función setup() y la configuración del acelerómetro.

```
2 ////////////////SETUP///////////////
4
6 void setup()
{
    Wire.begin();
    Serial.begin(115200);
    delay(3000);
10   Serial.println("Inicio Dispositivo nodo");
    Serial.println();
    CreateList(1);

14 }
16 void conf_acelerometro()
{
18   if (!myAcc.init()) {
        Serial.println("ADXL345 no conectado!");
20   }
    myAcc.setCorrFactors(-266.0, 285.0, -268.0, 278.0, -291.0, 214.0);
22   myAcc.setDataRate(ADXL345::DATA_RATE_25); //Se establece el DATA RATE
     del acelerometro en 25 Hz
    myAcc.setRange(ADXL345::RANGE_4G); //Se establece el rango de medida
     del acelerometro en 4g.
```

```

24| attachInterrupt(digitalPinToInterrupt(PININT1), registro_dato,
  | ISR_DEFERRED | RISING); //Se declara la interrupcion por actividad
  | del acelerometro.
myAcc.setActivityParameters(ADXL345_AC_MODE, ADXL345_XYZ, 0.8); //Se
  | establecen los parametros necesarios para la deteccion de actividad
  | (Modo AC, tres ejes, limite de detección)
26| myAcc.setInactivityParameters(ADXL345_DC_MODE, ADXL345_XYZ, 0.8,
  | tiempo_inactividad); //Se establecen los parametros necesarios para
  | la detección de la inactividad (Modo AC, tres ejes, limite de
  | detección, tiempo para detección)
myAcc.setLinkBit(true); //Se configura el Link Bit para que una vez
  | detecte actividad no vuelva a detectarla hasta que detecte
  | inactividad.
28| myAcc.setInterrupt(ADXL345_ACTIVITY, INT_PIN_1); //Se configura el
  | Pin INT1 como la interrupcion de salida del acelerometro para la
  | actividad.
myAcc.setInterrupt(ADXL345_INACTIVITY, INT_PIN_1); //Se configura el
  | Pin INT1 como la interrupcion de salida del acelerometro para la
  | inactividad.
30| Serial.println("Acelerometro configurado");
}

```

Listing B.5: Continuación del listing B.4. Fuente: elaboración propia.

En este fragmento se muestran las funciones que registran los datos deportivos.

```

//////////REGISTRO DEL DATO/////////
2| void registro_dato()
4{
  byte intSource = myAcc.readAndClearInterrupts();
6| if(myAcc.checkInterrupt(intSource, ADXL345_ACTIVITY))
{
 8|   tiempo = 0;
 9|   datos++;
10|   tiempoinicio = millis();

12|   Serial.println("ACTIVIDAD ");
14}
15| if(myAcc.checkInterrupt(intSource, ADXL345_INACTIVITY))
{
16|
17|   tiempofin = millis();
18|   if(datos == 0){Serial.println("INACTIVIDAD");}
19|   else
20{
21|     Serial.println("INACTIVIDAD");
22|     tiempo = tiempofin - tiempoinicio;
23|     if(tiempo < tiempo_inactividad*1000)
24{
25|       Serial.println(tiempo/1000);
26|       datos1.concat(";" + String(tiempo/1000));
27|       Serial.print("Datos: ");
28}
}

```

ANEXO B. CÓDIGO DEL DISPOSITIVO NODO

```
30         Serial.println(datos1);
31     }
32     {
33         tiempo = tiempo - tiempo_inactividad*1000;
34         Serial.println(tiempo/1000);
35         datos1.concat(";" + String(tiempo/1000));
36         Serial.print("Datos: ");
37         Serial.println(datos1);
38     }
39 }
40 }
41 myAcc.readAndClearInterrupts();
42 }
```

Listing B.6: Continuación del listing B.5. Fuente: elaboración propia.

Por último, se muestra la función principal loop() del dispositivo nodo.

```
//////////LOOP///////////
2 void loop()
4 {
5     switch(estado)
6     {
7
8         //Estado en el que se configura el Bluetooth y comienza el
9         //advertising para mostrarse hacia el maestro.
10        case 1:
11
12            Serial.println("Configurando BLE... ");
13            config_ble();
14            while(1)
15            {
16                if(salir_ble == true){break;}
17                sd_app_evt_wait();
18
19            }
20            salir_ble = false;
21            break;
22
23         //Estado en el que se reciben los datos del temporizador desde el
24         //maestro
25         case 2:
26             while(1)
27             {
28                 if(salir_recibirdato == true){break;}
29                 sd_app_evt_wait();
30             }
31             salir_recibirdato = false;
32             break;
33 }
```

```
//Estado en el que se configura el temporizador para el env o de  
los datos , para el reinicio y tambi n el aceler metro.  
34 case 3:  
  
36 ITimer0.attachInterruptInterval(tiempo_reinicio*1000,  
temporizador_handler);  
37 blinkTimer.begin(temporizador*1000, funcion_tempo);  
38 blinkTimer.start();  
39 Serial.println("Temporizador iniciado");  
40 conf_acelerometro();  
41 estado = 4;  
42 break;  
  
44 //Estado de normalidad de bajo consumo en el que se est  
registrando datos hasta que salta el temporizador y se sale de este  
estad o hacia el estad o de env o de los datos al maestro.  
45 case 4:  
46 while(1)  
47 {  
48     if(salir_estado4 == true){break;}  
49     sd_app_evt_wait();  
50 }  
51 salir_estado4 = false;  
52 break;  
  
56 //Estado en el que se configura el Bluetooth para el env o de los  
datos al maestro y comienza el advertising , se env an los datos , y  
se espera a que el maestro se desconecte.  
57 case 5:  
58 Serial.println("Final de toma de datos");  
59 Serial.println("Configurando BLE");  
60 blinkTimer.stop();  
61 startAdv();  
62 while(1)  
63 {  
64     if(salir_ble1 == true){break;}  
65     sd_app_evt_wait();  
66 }  
67 salir_ble1 = false;  
68 break;  
72 //Estado en el que se reinician los datos , las variables y el  
temporizador para iniciar un nuevo ciclo  
73 case 6:  
74 reinicio_datos();  
75 blinkTimer.reset();  
76 blinkTimer.begin(temporizador*1000, funcion_tempo);
```

ANEXO B. CÓDIGO DEL DISPOSITIVO NODO

```
blinkTimer.start();
80 Serial.println("Temporizador iniciado");
81 estado = 4;
82 break;

83 //Estado en el que se reinicia el dispositivo para una nueva
84 configuración.
85 case 7:
86     Serial.println("REINICIO");
87     NVIC_SystemReset();
88     break;
89 }

90 }

91 }
```

Listing B.7: Continuación del listing B.6. Fuente: elaboración propia.