



PROYECTO FINAL DE GRADO

Grado en Ingeniería Informática

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE MONITORIZACIÓN DE PLAZAS DE PARKING

Autora: Irene Casares Rodríguez

Director: José Manuel Palomares Muñoz

Noviembre, 2022



UNIVERSIDAD DE CÓRDOBA



ABREVIATURAS

ADR	Adaptative Data Rate
AES	Advanced Encryption Standard
API	Application Programming Interface
App	Aplicación móvil
AppEUI	Application Extended Unique Identifier
AppKey	Application Key
BPSK	Binary Phase-Shift Keying
CayenneLPP	Cayenne Low Power Payload
CRC	Cyclic Redundancy Check
CSS	Chirp Spread Spectrum
CU	Caso de Uso
DevEUI	Device Extended Unique Identifier
DL	Downlink (mensaje de bajada)
GND	Ground
GSM	Global System for Mobile
GSMA	Global System for Mobile Communications Association
HEX	Numeración hexadecimal
HTTP	Hypertext Transfer Protocol
I2C	Inter-Integrated Circuit
IDE	Integrated Development Environment
IoT	Internet of Things
IR	Infrarrojos
ISM	Industrial, Scientific and Medical bands
JoinEUI	Join Extended Unique Identifier
JSON	JavaScript Object Notation
LDR	Light Dependent Resistor
Li-Po	Polímero de Litio
LoRa	Long Range Radio
LoRaWAN	Long Range Wide Area Network
LPWAN	Low Power Wide Range Networks
LRTVD	Light Ray Tracing and Visioning Device
LTE	Long Term Evolution
MAC	Media Access Control

NB-IoT	Narrowband-IoT
OBJ	Objetivo del Sistema
PCB	Printed Circuit Board
PDS	Product Design Specification
QPSK	Quadrature Phase-Shift Keying
RAT	Radio Access Technology
RF	Requisito Funcional
RNF	Requisito No Funcional
RSSI	Received Signal Strength Indicator
SBC	Single-Board Computer
SCL	Serial Clock
SDA	Serial Data
SI	Sistema Internacional
TFG	Trabajo Fin de Grado
TMR	Triple Modular Redundancy
TTN	The Things Network
TTS	The Things Stack
UI	User Interface
UIT	Unión Internacional de Telecomunicaciones
UL	Uplink (mensaje de subida)
UML	Unified Modeling Language
UNB	Ultra Narrow Band
URL	Uniform Resource Locator
uT	microTeslas
Vcc	Voltage Common Collector

ÍNDICE GENERAL

<i>ABREVIATURAS.....</i>	<i>I</i>
<i>ÍNDICE DE ILUSTRACIONES.....</i>	<i>VII</i>
<i>ÍNDICE DE TABLAS</i>	<i>XI</i>
1. INTRODUCCIÓN.....	1
2. DEFINICIÓN DEL PROBLEMA.....	3
2.1. DEFINICIÓN DEL PROBLEMA REAL.....	3
2.2. DEFINICIÓN DEL PROBLEMA TÉCNICO.....	3
2.2.1. FUNCIONAMIENTO	3
2.2.2. ENTORNO	4
2.2.3. VIDA ESPERADA	4
2.2.4. CICLO DE MANTENIMIENTO	5
2.2.5. COMPETENCIA	5
2.2.6. ASPECTO EXTERNO	5
2.2.7. ESTANDARIZACIÓN	6
2.2.8. CALIDAD Y FIABILIDAD.....	6
2.2.9. PROGRAMA DE TAREAS	6
2.2.10. PRUEBAS	7
2.2.11. SEGURIDAD	7
3. OBJETIVOS	9
4. ANTECEDENTES.....	11
5. RESTRICCIONES.....	15
5.1. FACTORES DATO	15
5.2. FACTORES ESTRATÉGICOS.....	15
5.2.1. MÉTODOS DE DETECCIÓN	15
5.2.2. TECNOLOGÍAS DE COMUNICACIÓN INALÁMBRICA	21
6. RECURSOS.....	31

6.1. RECURSOS HUMANOS.....	31
6.2. RECURSOS HARDWARE.....	31
6.2.1. PROTOTIPO	31
6.3. RECURSOS SOFTWARE	33
6.3.1. HERRAMIENTAS PRINCIPALES.....	33
6.3.2. HERRAMIENTAS AUXILIARES	34
7. ESPECIFICACIÓN DE REQUISITOS.....	35
7.1. INTRODUCCIÓN	35
7.2. PARTICIPANTES DEL PROYECTO.....	35
7.3. OBJETIVOS DEL SISTEMA (OBJ)	36
7.4. REQUISITOS DEL SISTEMA	39
7.4.1. REQUISITOS FUNCIONALES (RF).....	39
7.4.2. REQUISITOS NO FUNCIONALES (RNF)	43
7.4.3. MATRIZ DE TRAZABILIDAD (RF-RNF/OBJ)	46
8. ANÁLISIS FUNCIONAL.....	47
8.1. IDENTIFICACIÓN DE ACTORES DEL SISTEMA.....	47
8.2. CASOS DE USO (CU)	47
8.2.1. MATRIZ DE TRAZABILIDAD (RF/CU)	58
8.3. DIAGRAMA DE ACTIVIDADES	58
9. DISEÑO DEL SISTEMA	61
9.1. DISEÑO ESTRUCTURAL	61
9.1.1. DIAGRAMA DE CLASES	61
9.2. DISEÑO DE LA ARQUITECTURA DEL SISTEMA.....	62
9.3. DISEÑO DEL ALGORITMO DE DETECCIÓN DE VEHÍCULOS	64
9.4. DISEÑO DE LA INTERFAZ.....	67
9.4.1. INTERFAZ DE LA PLATAFORMA DE MONITORIZACIÓN	67
9.4.2. INTERFAZ DE LA APLICACIÓN MÓVIL	68
10. PRUEBAS	71
10.1. PRUEBAS UNITARIAS.....	71

10.1.1. PRUEBAS DE CAJA BLANCA.....	71
10.1.2. PRUEBAS DE CAJA NEGRA.....	72
10.2. PRUEBAS DE INTEGRACIÓN.....	73
10.3. PRUEBAS DEL SISTEMA	73
11. <i>IMPLANTACIÓN</i>.....	74
11.1. PROTOTIPO DE DISPOSITIVO	75
11.1.1. HARDWARE.....	75
11.1.2. SOFTWARE	77
11.2. DESPLIEGUE EN ENTORNO	81
11.2.1. THE THINGS NETWORK	81
11.2.2. CONEXIÓN Y USO DE TTN.....	83
11.3. PLATAFORMA DE MONITORIZACIÓN.....	85
11.3.1. CAYENNE MYDEVICES	86
11.4. APLICACIÓN MÓVIL	91
12. <i>RESULTADOS EXPERIMENTALES</i>	105
13. <i>CONCLUSIONES</i>.....	111
13.1. CONCLUSIONES PERSONALES	111
13.2. CONCLUSIONES TÉCNICAS.....	112
14. <i>FUTURAS MEJORAS</i>	113
15. <i>BIBLIOGRAFÍA</i>.....	115
A. ANEXO I –MANUAL DE USUARIO	119
1. Introducción	121
2. Manual del Dispositivo	121
2.1. Integrar dispositivo en el sistema.....	121
2.1.1. Añadir dispositivo a aplicación en TTN/TTS.....	121
2.1.2. Instalar dispositivo en plaza de aparcamiento	126
2.1.3. Inicialización del dispositivo	127
2.1.4. Añadir dispositivo a plataforma de monitorización.....	129
2.1.5. Añadir dispositivo a aplicación móvil.....	130

2.2.	Eliminar dispositivo del sistema	132
2.2.1.	Eliminar dispositivo de aplicación móvil.....	132
2.2.2.	Eliminar dispositivo de Cayenne	133
2.2.3.	Eliminar dispositivo de TTN	134
2.2.4.	Desinstalar dispositivo de la plaza de aparcamiento.....	135
2.3.	Enviar mensaje <i>downlink</i> a dispositivo	136
3.	Manual de la Plataforma de Monitorización	137
3.1.	Dashboard general del proyecto.....	137
3.2.	Histórico de datos	138
3.3.	Análisis gráfico	139
4.	Manual de la App	140
4.1.	Compartir la App	140
4.2.	Instalar y acceder a la App.....	141
4.3.	Navegar por la App	143
4.3.1.	Pantalla principal: Mapa.....	143
4.3.2.	Pantalla secundaria: Plazas.....	144
4.3.3.	Obtener ruta hasta la plaza	145
4.3.4.	Filtrar resultados	146
B.	<i>ANEXO II – MANUAL DE CÓDIGO</i>	<i>147</i>
1.	Aplicación del dispositivo inalámbrico	149
2.	Aplicación web de Apps Script.....	155
C.	<i>ANEXO III - PRUEBAS.....</i>	<i>157</i>

ÍNDICE DE ILUSTRACIONES

Ilustración 1 - Comunicación dispositivos de Smart Parking. (Perković, Šolić, et al., 2020).....	12
Ilustración 2 - Bosch Parking Lot Sensor (Bosch, 2019).....	13
Ilustración 3 - Base del nodo Smart Parking de Libelium (Libelium, 2022b)	13
Ilustración 4 – Fotorresistencia (SunFounder, 2021c).....	16
Ilustración 5 – Rango de intensidad luminosa del LDR expuesto a la luz solar (Bachani et al., 2016)....	16
Ilustración 6 – Sensor ultrasónico HC-SR04 (Santos, 2021)	17
Ilustración 7 - Detección de vehículos mediante sensor ultrasónico (Perković, Šolić, et al., 2020).....	17
Ilustración 8 – Sensor de obstáculos IR (SunFounder, 2021b)	18
Ilustración 9 - Detección de vehículos mediante sensor IR (Perković, Šolić, et al., 2020).....	18
Ilustración 10 – Magnetómetro HMC5883L (SunFounder, 2021a).....	19
Ilustración 11 - Detección de vehículos mediante sensor magnético (Perković, Šolić, et al., 2020)	19
Ilustración 12 – LPWAN vs otras tecnologías inalámbricas (Alenezi et al., 2020)	21
Ilustración 13 – Gráfica comparación tecnologías inalámbricas (Campos, 2020)	21
Ilustración 14 – Alcance vs ‘data rate’ en tecnologías inalámbricas (Mekki et al., 2019)	22
Ilustración 15 – Cobertura Sigfox en España (Sigfox, n.d.).....	23
Ilustración 16 – Protocolo Sigfox (UnaBiz Spain, n.d.).....	23
Ilustración 17 – Cobertura NB-IoT de Vodafone en España (Vodafone España, 2022).....	24
Ilustración 18 – Pila de tecnología LoRa/LoRaWAN (Semtech, 2020).....	25
Ilustración 19 – Arquitectura típica de red LoRaWAN (The Things Network, n.d.)	26
Ilustración 20 – Tasa de datos de LoRa según SF (Semtech, 2019)	27
Ilustración 21 – Distribución de gateways LoRaWAN registrados en TTN en España	28
Ilustración 22 - Arduino MKR WAN 1310 (Arduino, 2022b).....	32
Ilustración 23 - Módulo GY-271 HMC5883L (AZ-Delivery, 2022).....	32
Ilustración 24 - Batería LiPo 3.7V 2000mAh (EEMB & Amazon, 2022).....	33
Ilustración 25 - Carcasa envolvente (VONVOFF & Amazon, 2022).....	33
Ilustración 26 - Diagrama de Casos de Uso	48
Ilustración 27 - Diagrama de Actividades.....	59
Ilustración 28 - Diagrama de Clases	61
Ilustración 29 - Arquitectura del Sistema.....	62
Ilustración 30 - Metodología para detectar vehículos mediante un magnetómetro (Sifuentes de la Hoya et al., 2015).....	64
Ilustración 31 - Algoritmo detección de vehículos.....	66
Ilustración 32 - Diseño UI plataforma monitorización: Tablero general	67
Ilustración 33 - Diseño UI plataforma monitorización: Información dispositivo.....	68
Ilustración 34 – Diseño UI aplicación móvil: Pantalla principal (Mapa)	68
Ilustración 35 – Diseño UI aplicación móvil: Listado de plazas.....	69
Ilustración 36 - Esquema de conexión del dispositivo	76

<i>Ilustración 37 - Prototipo de dispositivo.....</i>	77
<i>Ilustración 38 - Variaciones del campo magnético estando la plaza libre.....</i>	78
<i>Ilustración 39 - Flujo general del programa.....</i>	81
<i>Ilustración 40 - Cobertura LoRaWAN de TTN en Urbanización Las Jaras (Córdoba)</i>	82
<i>Ilustración 41 - Cobertura LoRaWAN de TTN en el Campus de Rabanales.....</i>	83
<i>Ilustración 42 - Aplicación en The Things Network.....</i>	84
<i>Ilustración 43 - Formato CayenneLPP en la consola de TTN.....</i>	85
<i>Ilustración 44 - Integración de Cayenne con TTN.....</i>	87
<i>Ilustración 45 - Representación en Cayenne de los datos enviados por un dispositivo</i>	88
<i>Ilustración 46 - Dashboard general de Cayenne.....</i>	88
<i>Ilustración 47 - Histórico de datos en Cayenne.....</i>	89
<i>Ilustración 48 - Representación gráfica de datos en Cayenne</i>	89
<i>Ilustración 49 - Dashboard de Cayenne compartido.....</i>	90
<i>Ilustración 50 - Dashboard de Cayenne compartido. Visualización desde dispositivo móvil.</i>	90
<i>Ilustración 51 - Flujo de información de TTN a la aplicación móvil.....</i>	91
<i>Ilustración 52 - Integración de Apps Script con TTN</i>	92
<i>Ilustración 53 – Aplicación web de Apps Script</i>	93
<i>Ilustración 54 - Implementación de aplicación web</i>	93
<i>Ilustración 55 - Trigger para actualizar tiempo meteorológico en Apps Script</i>	94
<i>Ilustración 56 - Hoja de cálculo con información a representar en la app.....</i>	95
<i>Ilustración 57 - Fuente de información de app Smart Parking (AppSheet).....</i>	96
<i>Ilustración 58 - Información de la app con columnas virtuales</i>	97
<i>Ilustración 59 - Configuración vistas aplicación móvil</i>	98
<i>Ilustración 60 - Configuración vista app: Mapa</i>	99
<i>Ilustración 61 - Regla de formato Mapa: pin verde cuando LIBRE</i>	99
<i>Ilustración 62 - Configuración de vista app: Plazas (1)</i>	100
<i>Ilustración 63 - Configuración de vista app: Plazas (2)</i>	100
<i>Ilustración 64 - Desplegar app y convertir en white-label.....</i>	101
<i>Ilustración 65 - Aplicación móvil AppSheet</i>	102
<i>Ilustración 66 - Smart Parking: Mapa.....</i>	103
<i>Ilustración 67 - Smart Parking: Plazas.....</i>	103
<i>Ilustración 68 - Pruebas con vehículo KIA Carnival.....</i>	105
<i>Ilustración 69 - Pruebas con vehículo Audi A1</i>	105
<i>Ilustración 70 - Posibles posiciones del dispositivo respecto al vehículo</i>	106
<i>Ilustración 71 - Representación pruebas de posición del dispositivo</i>	107
<i>Ilustración 72 - Obtener DevEUI</i>	122
<i>Ilustración 73 - Añadir dispositivo a TTN (1)</i>	123
<i>Ilustración 74 - Añadir dispositivo a TTN (2)</i>	123
<i>Ilustración 75 - Añadir dispositivo a TTN (3)</i>	124
<i>Ilustración 76 - Configurar ubicación de dispositivo en TTN (1).....</i>	125

<i>Ilustración 77 - Configurar ubicación de dispositivo en TTN (2).....</i>	125
<i>Ilustración 78 - Instalar dispositivo en superficie de la plaza.....</i>	126
<i>Ilustración 79 - Inicializar dispositivo (1).....</i>	127
<i>Ilustración 80 - Inicializar dispositivo (2).....</i>	128
<i>Ilustración 81 - Añadir dispositivo a Cayenne (1).....</i>	129
<i>Ilustración 82 - Añadir dispositivo a Cayenne (2).....</i>	130
<i>Ilustración 83 - Añadir dispositivo a app (1).....</i>	130
<i>Ilustración 84 - Añadir dispositivo a app (2).....</i>	131
<i>Ilustración 85 - Añadir dispositivo a app (3).....</i>	131
<i>Ilustración 86 - Añadir dispositivo a app (4).....</i>	132
<i>Ilustración 87 - Eliminar dispositivo de Cayenne (1)</i>	133
<i>Ilustración 88 - Eliminar dispositivo de Cayenne (2)</i>	133
<i>Ilustración 89 - Eliminar dispositivo de TTN (1).....</i>	134
<i>Ilustración 90 - Eliminar dispositivo de TTN (2).....</i>	134
<i>Ilustración 91 - Eliminar dispositivo de TTN (3).....</i>	135
<i>Ilustración 92 - Enviar mensaje downlink (1)</i>	136
<i>Ilustración 93 - Enviar mensaje downlink (2)</i>	137
<i>Ilustración 94 - Configurar dashboard general del proyecto (Cayenne).....</i>	138
<i>Ilustración 95 - Acceder al histórico de datos (Cayenne).....</i>	138
<i>Ilustración 96 - Realizar análisis gráfico en Cayenne (1).....</i>	139
<i>Ilustración 97 - Realizar análisis gráfico en Cayenne (2).....</i>	139
<i>Ilustración 98 - Compartir app (1)</i>	140
<i>Ilustración 99 - Compartir app (2)</i>	140
<i>Ilustración 100 - Compartir app (3)</i>	141
<i>Ilustración 101 - Instalar app (1)</i>	141
<i>Ilustración 102 - Instalar app (2)</i>	141
<i>Ilustración 103 - Acceso a app (1).....</i>	142
<i>Ilustración 104 - Acceso a app (2).....</i>	142
<i>Ilustración 105 - Acceso directo a app</i>	142
<i>Ilustración 106 - Pantalla principal: Mapa</i>	143
<i>Ilustración 107 - Acceso a información desde Mapa.....</i>	143
<i>Ilustración 108 - Pantalla secundaria: Plazas</i>	144
<i>Ilustración 109 - Acceso a información desde vista Plazas.....</i>	144
<i>Ilustración 110 - Obtener ruta hasta la plaza</i>	145
<i>Ilustración 111 - Filtrar en Mapa.....</i>	146
<i>Ilustración 112 - Filtrar en Plazas</i>	146

ÍNDICE DE TABLAS

Tabla 1 – Comparativa sensores.....	20
Tabla 2 – Comparativa tecnologías inalámbricas LPWAN.....	28
Tabla 3 - Participante Irene Casares Rodríguez.....	35
Tabla 4 - Participante José Manuel Palomares Muñoz.....	36
Tabla 5 – OBJ-1: Implementación del sistema	37
Tabla 6 – OBJ-2: Diseño del dispositivo de detección.....	37
Tabla 7 – OBJ-3: Implementación de red inalámbrica	38
Tabla 8 – OBJ-4: Integración de plataforma de monitorización	38
Tabla 9 – OBJ-5: Desarrollo de aplicación móvil	38
Tabla 10 – OBJ-6: Diseño del algoritmo de detección.....	39
Tabla 11 – OBJ-7: Diseño de protocolo de mensajes	39
Tabla 12 – RF-1: Detectar vehículo.....	40
Tabla 13 – RF-2: Detectar iluminación.....	40
Tabla 14 – RF-3: Transmitir por red inalámbrica.....	40
Tabla 15 – RF-4: Recibir información	41
Tabla 16 – RF-5: Proporcionar ocupación e iluminación.....	41
Tabla 17 – RF-6: Proporcionar ubicación	42
Tabla 18 – RF-7: Modificar parámetros en remoto	42
Tabla 19 – RF-8: Visualizar información.....	42
Tabla 20 – RNF-1: Información del dispositivo en tiempo real.....	43
Tabla 21 – RNF-2: Precisión del dispositivo.....	43
Tabla 22 – RNF-3: Alimentación del dispositivo	44
Tabla 23 – RNF-4: Uso de la alimentación del dispositivo	44
Tabla 24 – RNF-5: Confinamiento del dispositivo.....	44
Tabla 25 – RNF-6: Información del sistema en tiempo real.....	45
Tabla 26 – RNF-7: Hardware y Software open-source.....	45
Tabla 27 - Matriz de trazabilidad RF-RNF/OBJ.....	46
Tabla 28 - CU-1: Detectar vehículo.....	49
Tabla 29 - CU-2: Comprobar cambio de estado.....	50
Tabla 30 - CU-3: Detectar iluminación.....	50
Tabla 31 - CU-4: Enviar mensaje (dispositivo)	51
Tabla 32 - CU-5: Recibir mensaje	52
Tabla 33 - CU-6: Añadir dispositivo.....	53
Tabla 34 - CU-7: Eliminar dispositivo	54
Tabla 35 - CU-8: Enviar mensaje (administrador).....	55
Tabla 36 - CU-9: Monitorizar información	56
Tabla 37 - CU-10: Consultar aplicación.....	56

<i>Tabla 38 - CU-11: Ocupar plaza</i>	57
<i>Tabla 39 - CU-12: Liberar plaza</i>	57
<i>Tabla 40 - Matriz de trazabilidad RF/CU</i>	58
<i>Tabla 41 - Tabla de conexiones del dispositivo</i>	76
<i>Tabla 42 - Ejemplos de mensajes downlink</i>	80
<i>Tabla 43 - Comparativa para determinar la mejor posición</i>	107
<i>Tabla 44 – Validación de las pruebas experimentales</i>	109



1. INTRODUCCIÓN

Uno de los temas más importantes tratados por la Comisión Europea y la mayoría de los países del mundo es el desarrollo de modelos urbanos destinados a aumentar la calidad de vida de las personas que trabajan y viven en ellos.

En sintonía con la *Estrategia de Movilidad Segura, Sostenible y Conectada 2030* del MITMA¹ (eje estratégico 5 (MITMA, 2021)), la Movilidad Inteligente y Sostenible es uno de los conceptos centrales en la visión de la *Smart City*, donde el Internet de las Cosas (IoT) juega un papel muy importante (Nižetić et al., 2019; Perković, Damjanović, et al., 2020).

El problema fundamental que subyace al concepto de Smart City es la reducción de la congestión del tráfico en la ciudad. Los conductores dedican, en promedio, un 25% del tiempo de conducción en encontrar un lugar para aparcar, generando aproximadamente el 35% de la congestión del tráfico diario (Rueda, 2022). Todo esto, además de aumentar su frustración, incrementa drásticamente el consumo de combustible, lo que contribuye a la contaminación del aire en las ciudades.

Una de las principales vías para reducir esa congestión y sus problemas derivados sería desarrollar sistemas de aparcamiento inteligentes destinados a encontrar, de manera más eficiente, un lugar libre para estacionar.

En el presente Trabajo Fin de Grado se propone la creación de un sistema prototípico para la monitorización de plazas de aparcamiento que dé respuesta a este problema. A fin de crear una solución flexible y escalable, se ve la necesidad de utilizar tecnologías inalámbricas para el envío y recepción de datos, empleando redes de amplia distancia que permitan dar cobertura a centenares de dispositivos. Así mismo, se presenta una problemática respecto a la posible congestión de las redes de comunicación, por lo que se debe hacer un uso muy eficiente de este recurso.

¹ Ministerio de Transportes, Movilidad y Agenda Urbana



2. DEFINICIÓN DEL PROBLEMA

En este apartado se tratará de definir el problema al que se quiere dar solución, identificando las necesidades que se presentan y estableciendo los objetivos esenciales que satisfagan esas exigencias. Esta definición se realizará a través de dos vertientes: el problema real, que aporta una visión desde la perspectiva del usuario final, y el problema técnico, que presenta un enfoque desde la ingeniería.

2.1. DEFINICIÓN DEL PROBLEMA REAL

El problema real que se plantea para este proyecto es el de implementar un sistema prototipo de monitorización de plazas de aparcamiento, el cual abarcaría desde los dispositivos que detecten el estado de las plazas hasta las aplicaciones de monitorización utilizadas por el administrador y los usuarios finales para conocer su ocupación en tiempo real.

Tal y como se ha comentado previamente, este sistema utilizaría redes de comunicación inalámbrica de larga distancia que dieran cobertura a toda la infraestructura de dispositivos que se pudiera desplegar y facilitaran su posible escalabilidad.

Como valor añadido, el sistema daría información acerca de la incidencia del Sol sobre la plaza, de forma que se pudiera inferir si la plaza se encuentra bajo luz solar directa o a la sombra. Así, los usuarios podrían decidir qué plaza ocupar según sus condiciones y preferencias, algo especialmente interesante en días con altas temperaturas.

2.2. DEFINICIÓN DEL PROBLEMA TÉCNICO

Una vez definido el problema real, se identifica el problema técnico mediante la técnica *Product Design Specification* (PDS), que permite establecer todos los condicionantes del problema.

2.2.1. FUNCIONAMIENTO

El dispositivo prototipo, integrado con algún tipo de sensor, implementaría un algoritmo que le permitiera detectar la presencia de un vehículo situado en la plaza.



Además, obtendría datos de la incidencia del Sol sobre la plaza, aportando información extra a los usuarios a la hora de escoger la plaza que más les convenga.

La información obtenida y procesada por el dispositivo se enviaría a través de una red inalámbrica y se haría llegar tanto a la aplicación de usuario como a la plataforma de monitorización utilizada por el administrador.

El dispositivo también podría recibir datos enviados por el administrador del sistema que le indicaran algún cambio en su comportamiento.

Finalmente, la plataforma de monitorización permitiría al administrador realizar diferentes análisis sobre la información histórica de las plazas, mientras que la aplicación de usuario les mostraría la ubicación de las plazas e información sobre su estado en tiempo real.

2.2.2. ENTORNO

Los dispositivos se instalarían en la superficie de plazas de aparcamiento situadas en el exterior, lo que afectaría a su estructura y a los sensores utilizados, así como a la autonomía ofrecida por su fuente de alimentación. Además, estarían expuestos a diferentes condiciones meteorológicas, incluyendo lluvia, cambios de temperatura y humedad.

Por otro lado, el administrador del sistema podría enviar mensajes a los dispositivos desde cualquier navegador web, como Google Chrome, Firefox u Opera.

Así mismo, los usuarios finales accederían a la aplicación de visualización desde un dispositivo móvil. Esta aplicación estaría diseñada para el público general, sin necesidad de experiencia previa en este tipo de tecnología IoT, por lo que debería permitir un rápido y fácil uso.

2.2.3. VIDA ESPERADA

En un sistema de estas características sería conveniente que los dispositivos utilizados perduraran durante años, solamente requiriendo un cambio de batería cuando fuera necesario. Lo mismo aplicaría a la aplicación móvil, cuya vida útil, como todo producto software, dependería de su repercusión.



Sin embargo, al tratarse de un sistema prototipo, su vida esperada abarcará el tiempo de desarrollo de este proyecto, pudiendo ser extendida con posteriores actualizaciones o mejoras que se quisieran realizar.

2.2.4. CICLO DE MANTENIMIENTO

Con relación a los dispositivos, necesitarían un cambio de su fuente de alimentación cada cierto tiempo, supeditado al consumo realizado. Además, dependiendo del deterioro producido por las condiciones meteorológicas o por la interacción de usuarios, podrían requerir una sustitución de algunos componentes o del dispositivo completo.

Por otro lado, cabría la posibilidad de que precisaran algún tipo de actualización de su firmware o software al surgir formas más eficientes de implementar sus funcionalidades.

Así mismo, una vez desarrollada la aplicación, también podría necesitar actualizaciones con el fin de mejorar su rendimiento o incluso para incorporar nuevas funcionalidades.

2.2.5. COMPETENCIA

La industria IoT está en auge, sobre todo con relación a la Smart City, aumentando día a día los proyectos desarrollados, ya sean por entidades públicas o privadas. Por lo tanto, existe cierta competencia en este ámbito.

Sin embargo, son pocos los proyectos basados en hardware y software *open-source*, algo que distinguiría a este proyecto, además del bajo costo que se pretende conseguir.

2.2.6. ASPECTO EXTERNO

Respecto a los dispositivos, que se instalarían en la superficie de la plaza asignada, se confinarían en estructuras herméticas que permitieran su correcto funcionamiento independientemente de las condiciones a las que se vieran sometidos.

Por otro lado, la aplicación final debería ser amigable e intuitiva, así como facilitar la navegabilidad y la interacción de los usuarios.



2.2.7. ESTANDARIZACIÓN

Al tratarse de dispositivos prototipos, su diseño estructural no se sometería a ninguna normativa de estandarización. Sin embargo, respecto a la comunicación inalámbrica, sí podría estar sujeta al estándar de la tecnología finalmente escogida.

Por otro lado, en referencia a la estructura del código implementado en esos dispositivos, al igual que la aplicación final, se procurará seguir los estándares del lenguaje de programación utilizado.

2.2.8. CALIDAD Y FIABILIDAD

Para asegurar la calidad del sistema, se establecerán puntos de control donde se evalúe el cumplimiento de los requisitos establecidos, como la detección sin errores de un vehículo o el correcto envío de los datos. Respecto a la detección, será admisible que vehículos estrechos como las motocicletas no sean captados, ya que la intención principal es detectar vehículos de gran tamaño, como turismos o furgonetas.

La fiabilidad es lo que principalmente define a un sistema, pues mide la capacidad de proporcionar de manera continuada un servicio correcto. En este caso, un servicio correcto de los dispositivos sería detectar la presencia de un vehículo, sin aportar falsos positivos ni negativos, y enviar esa información a través de una red inalámbrica. Sin embargo, al tratarse de un prototipo que no proporciona un servicio crítico, se admitirán breves intervalos de funcionamiento erróneo. Para mejorar su fiabilidad, se podrían aplicar técnicas de redundancia como la replicación de los sensores utilizados, formando un sistema de *Triple Modular Redundancy* (TMR), aunque esto incrementaría el coste final del dispositivo.

Por otro lado, la aplicación final debería aportar calidad y fiabilidad al sistema representando correctamente la información actualizada de las plazas para cumplir con la visualización en tiempo real.

2.2.9. PROGRAMA DE TAREAS

En general, el Proyecto se dividirá en las siguientes etapas:

- **Fase de Estudio:** fase inicial basada en el estudio previo del problema, incluidos sus antecedentes, y de la teoría necesaria para resolverlo, siendo



vital el análisis de sensores de detección de obstáculos y de tecnologías inalámbricas.

- **Fase de Diseño:** una vez realizado el estudio teórico y analizados los objetivos, se procederá a diseñar un sistema que cumpla con los requisitos.
- **Fase de Implementación:** a partir del diseño concretado en la fase anterior, se realizará la implementación del sistema, donde se integrarán los dispositivos, conectados a una red inalámbrica, y se visualizará su información en la aplicación final.
- **Fase de Pruebas:** en esta fase, el sistema se someterá a diferentes pruebas para detectar y corregir los errores que pudieran producir sus componentes.
- **Fase de Documentación:** simultáneamente al resto de fases, se realizará la documentación, comprendiendo la Memoria final del Trabajo y los anexos necesarios.

2.2.10. PRUEBAS

Para certificar el correcto funcionamiento del sistema, así como de sus partes individuales, se verá sometido a diferentes pruebas, como pueden ser la detección de vehículos de distinto tipo o el envío de numerosos mensajes para examinar la congestión de la red.

2.2.11. SEGURIDAD

Con relación a los dispositivos, sería conveniente que cumplieran el *Reglamento electrotécnico para baja tensión e ITC* (Ministerio de Ciencia y Tecnología, 2021), minimizando posibles riesgos como cortocircuitos, derivaciones a tierra o falsos contactos. Por otro lado, se podría buscar alguna apariencia identificativa de su estructura envolvente para hacerlos resaltar en su lugar de instalación y que no constituyeran un peligro para los usuarios de su entorno.

Respecto a la seguridad de la aplicación, no se prevé el registro de los usuarios para su uso, por lo que no será necesario el tratamiento de sus datos.



3. OBJETIVOS

El objetivo principal del presente proyecto consiste en implementar un sistema prototípico de monitorización de plazas de aparcamiento.

En el contexto de IoT, la implementación de soluciones para la detección de estacionamientos libres se basa en el desarrollo de dispositivos que puedan detectar la presencia de vehículos y enviar estos datos a un servidor central para su posterior procesamiento.

Así, el sistema integraría un hardware y software específicos. Fundamentándose en una pequeña red de dispositivos prototípicos que se comunicarían mediante tecnología inalámbrica, la información obtenida y procesada sería visualizada en una aplicación móvil para favorecer la accesibilidad de los usuarios finales.

Partiendo del objetivo principal, surgen otros más específicos:

- Seleccionar el sensor adecuado para la detección de vehículos, centrándose en soluciones que mantengan una alta precisión y reduzcan el costo total.
- Concebir un algoritmo que permita, a partir de los datos aportados por el sensor, discernir la presencia o no de un vehículo.
- Conocer la incidencia del Sol sobre las plazas libres con relación a si reciben luz solar directa o no.
- Seleccionar el tipo de red inalámbrica más apropiado para este sistema, priorizando aquellas que permitan un mayor alcance manteniendo un bajo consumo.
- Diseñar un protocolo de comunicación desde el dispositivo hasta el servidor de red y viceversa, permitiendo la interacción remota y considerando una gestión eficiente de la comunicación.
- Integrar una plataforma de monitorización para la representación de los datos y su monitoreo por parte del administrador del sistema.



4. ANTECEDENTES

El Internet de las cosas integra el hardware, el software y la conectividad de red que permiten que los objetos se detecten y controlen de forma remota a través de la red existente. Tal integración permitiría a los usuarios monitorear los lugares de estacionamiento disponibles, lo que lleva a una mayor eficiencia, precisión y beneficio económico (Gupta et al., 2017).

Según Vîlceanu et al., (2019), son numerosas las ventajas potenciales de implementar un sistema de aparcamiento inteligente o *Smart Parking*, incluyendo:

- Predicción y detección precisa de la ocupación de plazas en tiempo real.
- Guía a los residentes y visitantes a aparcamiento disponible.
- Optimización del uso del espacio de aparcamiento.
- Asistencia al flujo de tráfico en la ciudad aprovechando la tecnología IoT.
- Decisiones inteligentes a partir de datos, incluyendo aplicaciones de estado en tiempo real y análisis históricos.
- Mejoras en el entorno urbano, reduciendo la emisión de CO₂ y otros contaminantes.
- Mejor supervisión y gestión en tiempo real del espacio de estacionamiento disponible, lo que podría traducirse en una importante generación de ingresos.

En los últimos años se han propuesto una serie de soluciones encaminadas a resolver el problema de encontrar plazas de aparcamiento, así como establecer un entorno verde y aumentar la calidad de vida en las ciudades inteligentes (Kotb et al., 2016; Lin et al., 2017). En esencia, la mayoría de estas soluciones se basan en una aplicación móvil a través de la cual los usuarios reciben información sobre las plazas de aparcamiento libres (**Ilustración 1**).

Gupta et al. (2017) y Sabbea (2017) proponen unos sistemas en los que incluyen el uso de un sensor ultrasónico, Arduino Uno, módulo Wi-Fi ESP8266-01, servidor en la nube y aplicaciones móviles de usuario final.



Por otro lado, Šarić & Mihaljević (2017) describen algunas soluciones de *Smart Parking* implementadas en la ciudad de Dubrovnik, las cuales también incluyen redes de sensores y enrutadores en el lado del hardware, y software para aplicaciones web y móviles.

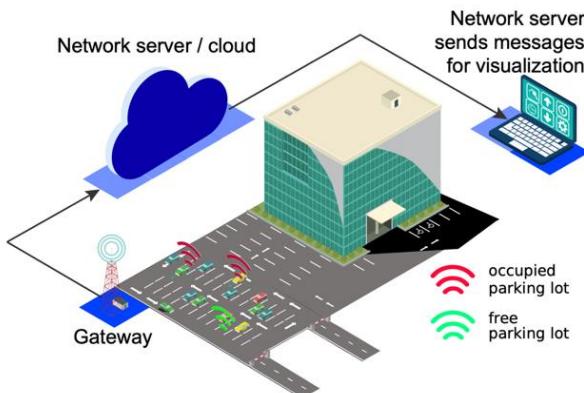


Ilustración 1 - Comunicación dispositivos de Smart Parking. (Perković, Šolić, et al., 2020)

Propuestas más actuales son las de Floris et al. (2022), que utilizan magnetómetros para detectar los vehículos y tecnología Wi-Fi para enviar la información, o la de Kadusic et al. (2022), empleando la comunicación NB-IoT.

A nivel comercial, cada vez se ofertan más soluciones completas que permiten desplegar rápidamente un sistema de aparcamiento inteligente.

Entre esas soluciones comerciales se encuentran las de Bosch (2021) y Libelium (2022c), que ofrecen sendos dispositivos de detección de vehículos (**Ilustración 2** y **Ilustración 3**) basados en dos sensores independientes: magnetómetro y radar. Alimentados por baterías de litio, pueden durar entre 5 y 10 años en funcionamiento, dependiendo de su configuración y uso, mientras soportan cambios ambientales e influencias mecánicas como vehículos pesados. Además, utilizan la tecnología LoRaWAN para comunicarse, alargando la vida de la batería y permitiendo una amplia cobertura.



Ilustración 2 - Bosch Parking Lot Sensor (Bosch, 2019)



Ilustración 3 - Base del nodo Smart Parking de Libelium (Libelium, 2022b)

El principal inconveniente que presentan estas soluciones comerciales es el precio de sus dispositivos, superando los 200 € por unidad, además de la obligación de uso de su aplicación y nube, como en el caso de Libelium.

Por eso, en este Proyecto se pretende obtener un sistema prototípico con un bajo coste, en comparación a los anteriores, y priorizando el *open-source* en su desarrollo.



5. RESTRICCIONES

En este apartado se exponen las restricciones existentes en el ámbito del diseño y que condicionan la elección de una u otra alternativa en el desarrollo del Proyecto. Estas limitaciones se dividen en dos grupos:

- **Factores dato:** restricciones inamovibles e inherentes al Proyecto o al entorno en el que se desarrolla el mismo.
- **Factores estratégicos:** opciones de diseño escogidas por la proyectista según criterios propios y sin ninguna clase de imposición externa.

5.1. FACTORES DATO

Entre estas limitaciones se encuentran:

- **Limitaciones de tiempo:** se deberá terminar antes del 10 de noviembre, fecha límite de entrega del TFG en la convocatoria de noviembre 2022.
- **Funcionamiento y uso:** el dispositivo prototipo desarrollado debe ser de fácil instalación y manejo. Además, debe tener tal autonomía que no obligue a una interacción física reiterada con el mismo tras su colocación.
- **Uso objetivo:** se espera que cualquier persona pueda conocer los datos actuales de ocupación, por lo que la visualización final de los mismos debe ser accesible para todos.

5.2. FACTORES ESTRATÉGICOS

Dentro de los factores estratégicos se consideran diferentes propuestas para la consecución de los objetivos:

5.2.1. MÉTODOS DE DETECCIÓN

Existen muchos métodos de detección de obstáculos. Entre los más utilizados para el aparcamiento inteligente se encuentran los sensores ultrasónicos, infrarrojos y, sobre todo, magnetómetros. Hay otros menos fiables, pero también utilizados, como pueden ser las fotorresistencias.



5.2.1.1. FOTORRESISTENCIA

Una fotorresistencia, *Light Dependent Resistor* (LDR) o fotocélula, como la de la **Ilustración 4**, es una resistencia variable controlada por la luz. Esta resistencia exhibe fotoconductividad, es decir, disminuye al aumentar la intensidad de la luz incidente (SunFounder, 2021c).

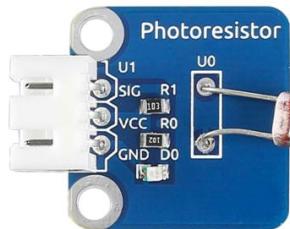


Ilustración 4 – Fotorresistencia (SunFounder, 2021c)

En el estudio de Bachani et al. (2016) se observa esa completa dependencia respecto a las condiciones lumínicas, en concreto a la luz solar (**Ilustración 5**). Esto complica enormemente su uso en exteriores para el caso que nos ocupa, donde debería detectar la disminución de la luz recibida al situarse un vehículo sobre el dispositivo.

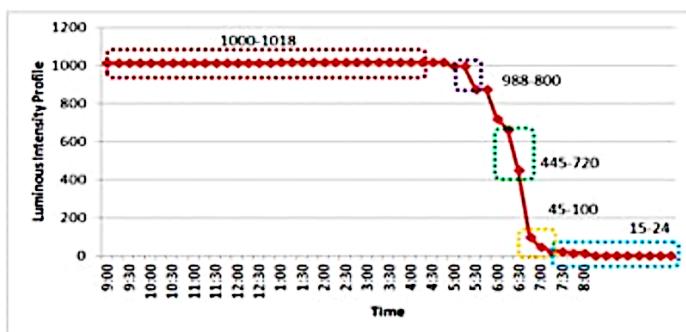


Ilustración 5 – Rango de intensidad luminosa del LDR expuesto a la luz solar (Bachani et al., 2016)

5.2.1.2. SENSOR DE ULTRASONIDO

Un sensor ultrasónico (**Ilustración 6**) consta de un transmisor ultrasónico, un receptor y un circuito de control. Este sensor utiliza sonido para detectar objetos y medir la distancia hasta ellos según el tiempo necesario para recibir el eco de una onda de sonido ultrasónico emitida por el transmisor, reflejada por el objeto y recibida por el receptor del sensor (Santos, 2021).



Ilustración 6 – Sensor ultrasónico HC-SR04 (Santos, 2021)

Este tipo de sensor es capaz de detectar objetos a una distancia de hasta 5 metros, dependiendo del modelo, aunque también puede ser impreciso (Perković, Šolić, et al., 2020).

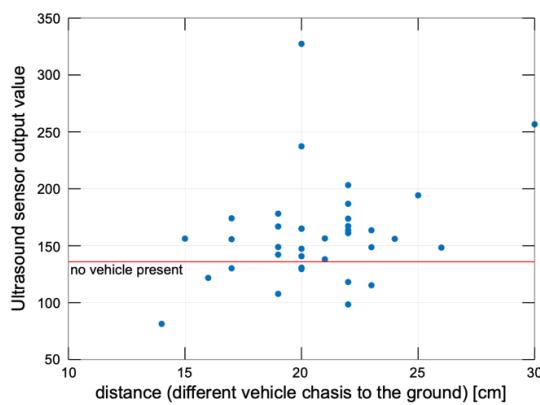


Ilustración 7 - Detección de vehículos mediante sensor ultrasónico (Perković, Šolić, et al., 2020)

Perković, Šolić, et al. (2020) muestra en la Ilustración 7 algunas mediciones con predicciones falsas (menores que el valor umbral) en la detección de vehículos con chasis de diferentes alturas, lo que indica que el sensor no es fiable para cada detección realizada. Por otro lado, Paidi et al. (2018) mencionan su sensibilidad a los cambios ambientales como la lluvia o la nieve, por lo que podría no ser apto para exteriores.

5.2.1.3. SENSOR DE OBSTÁCULOS INFRARROJO

Un sensor de obstáculos infrarrojo (IR) como el de la Ilustración 8 consta principalmente de un transmisor de infrarrojos, un receptor y un potenciómetro. Existen otros modelos en los que el transmisor y el receptor están separados en sendos dispositivos, pero eso implicaría colocar uno de ellos en una estructura suspendida sobre la plaza a monitorizar, lo que sería una instalación mucho más intrusiva.

Este dispositivo funciona según el principio de reflexión infrarroja para detectar obstáculos: si no hay ningún objeto, el rayo infrarrojo emitido se debilitará con la



distancia recorrida y finalmente desaparecerá; mientras que, si hay un objeto que bloquea y refleja la luz infrarroja, el receptor infrarrojo recibirá la señal, confirmando la presencia de un obstáculo (SunFounder, 2021b).

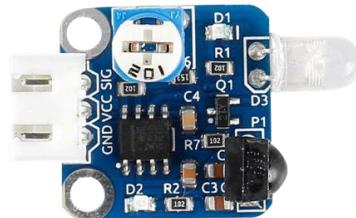


Ilustración 8 – Sensor de obstáculos IR (SunFounder, 2021b)

Bachani et al., (2016) muestra resultados en los que, a diferencia del LDR, su desempeño no se ve tan influido por las condiciones lumínicas en las que trabaja. Sin embargo, varios autores (Paidi et al., 2018; Solic et al., 2019) mencionan su sensibilidad a los cambios ambientales, lo que tampoco lo haría adecuado para estacionamientos en exterior.

Por otro lado, Perković, Šolić, et al. (2020) expone en la **Ilustración 9** algunas imprecisiones en los resultados obtenidos en la detección de vehículos, pudiendo deberse a la distancia entre el chasis y el suelo.

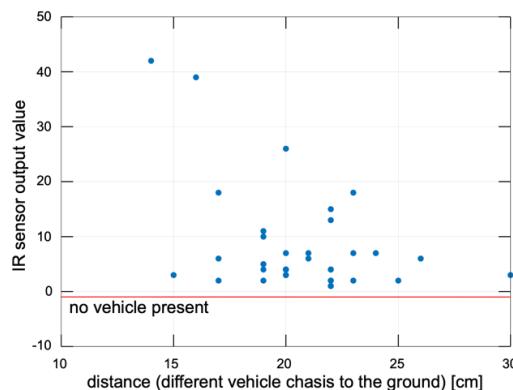


Ilustración 9 - Detección de vehículos mediante sensor IR (Perković, Šolić, et al., 2020)

5.2.1.4. SENSOR MAGNÉTICO

Un magnetómetro o sensor magnético como el de la **Ilustración 10** es un dispositivo que mide las características de un campo magnético, incluida la magnitud, dirección y sus cambios durante el tiempo. Los valores medidos se pueden suponer como un vector con componentes en los ejes X, Y y Z (Solic et al., 2019).

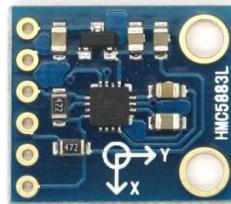


Ilustración 10 – Magnetómetro HMC5883L (SunFounder, 2021a)

Este sensor es diferente a los anteriores respecto a que no se puede utilizar para detectar objetos compuestos de cualquier material, sino que deben tener cierta cantidad de componentes magnéticos, lo que complica ligeramente el proceso de detección.

En presencia de material ferromagnético, cuya permeabilidad magnética es mayor al aire que lo rodea, las líneas de flujo del campo magnético terrestre se distorsionan. Un sensor magnético situado en las inmediaciones puede detectar esa distorsión al comparar la variación ocurrida respecto a los valores previos, señalando así la presencia de un objeto magnético (Sifuentes de la Hoya et al., 2015). Dependiendo de la cantidad de material ferromagnético, la distorsión puede ser mayor, lo que facilitaría su detección.

En el análisis de Perković et al. (2020), a diferencia de los otros sensores, el magnetómetro presenta resultados bastante precisos (**Ilustración 11**), aunque pueden depender de la estructura metálica del chasis.

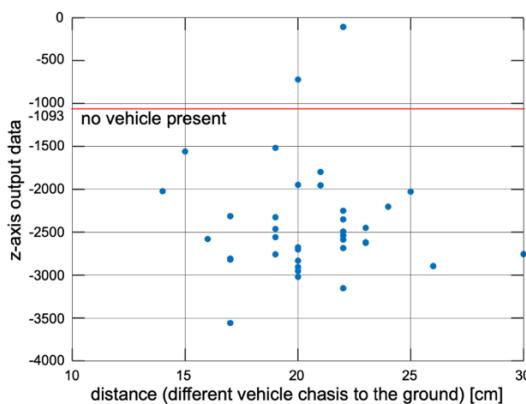


Ilustración 11 - Detección de vehículos mediante sensor magnético (Perković, Šolić, et al., 2020)

Como indica Paidi et al. (2018), este sensor no es sensible a los cambios ambientales, pero debe situarse cerca del vehículo para detectarlo, por lo que se debe colocar bajo el mismo, sobre la superficie de la plaza.



5.2.1.5. COMPARATIVA

Una vez analizados varios posibles sensores, se presenta una comparativa en la **Tabla 1** que ayude a la elección del más apropiado para este trabajo.

Para relacionar la fiabilidad de cada sensor, se han ordenado de menos fiable (4) a más (1) según los resultados de Perković, Šolić, et al. (2020), donde también se presenta el consumo medio de cada uno.

Respecto al tamaño, se han tenido en cuenta los módulos anteriormente mostrados, que incluyen la electrónica necesaria para el uso de los sensores.

Tabla 1 – Comparativa sensores

CARACTERÍSTICA	TIPO DE SENSOR			
	LDR	HC-SR04 (Ultrasónico)	IR	HMC5883L (Magnetómetro)
Máxima distancia de detección	Depende de la cantidad de luz recibida	400 cm	30 cm	Depende de la cantidad de material ferromagnético
Orden Fiabilidad (Perković, Šolić, et al., 2020)	4	3	2	1
Consumo medio (Perković, Šolić, et al., 2020)	103.11 mW	89.21 mW	106.03 mW	82.54 mW
Tamaño	2 x 2 cm	4.5 x 2 cm	2 x 2 cm	1.5 x 1.5 cm
Funcionamiento independiente de las condiciones lumínicas	-	✓	✓	✓
Desempeño apropiado en exteriores	-	-	-	✓
Desempeño apropiado en estructura cerrada (carcasa)	-	-	-	✓

Con esta comparativa se puede ver que el magnetómetro ensombrece a los demás sensores, superándolos tanto en fiabilidad como en consumo energético, además de ser el más idóneo para el prototipo a desarrollar debido a su tamaño y desempeño en una estructura cerrada.



5.2.2. TECNOLOGÍAS DE COMUNICACIÓN INALÁMBRICA

En la aplicación al IoT las tecnologías tradicionales (p.ej. WiFi o 3G) presentan ciertas carencias, sobre todo con relación al consumo y coste. Con el objetivo de cubrir esas necesidades, surgieron nuevas tecnologías de comunicación inalámbrica (Pardal, 2017).

Para posibilitar el despliegue de mayores iniciativas IoT se diseñaron las *Low Power Wide Range Networks* (LPWAN), un grupo de tecnologías de red de área amplia y baja potencia que presentan diferentes configuraciones según el uso de frecuencias con o sin licencia y de estándares abiertos o patentados (Rupareliya, 2022).

En la **Ilustración 12** se puede ver una comparación de varias tecnologías de transmisión inalámbrica según los tres ejes entorno a los que pivotan: alcance, capacidad de transmisión y consumo energético.

	Bluetooth	BLE	ZigBee	WiFi	Cellular M2M	LPWAN
long range (>10 km)	✗	✗	✗	✗	✓	✓
low data rate (<5 K bit/s or 20–256 bytes per message)	✓	✓	✓	✗	✗	✓
low power (to last 5–years on single battery)	✓	✓	✓	✗	✗	✓
low cost	✓	✓	✓	✓	✗	✓

Ilustración 12 – LPWAN vs otras tecnologías inalámbricas (Alenezi et al., 2020)

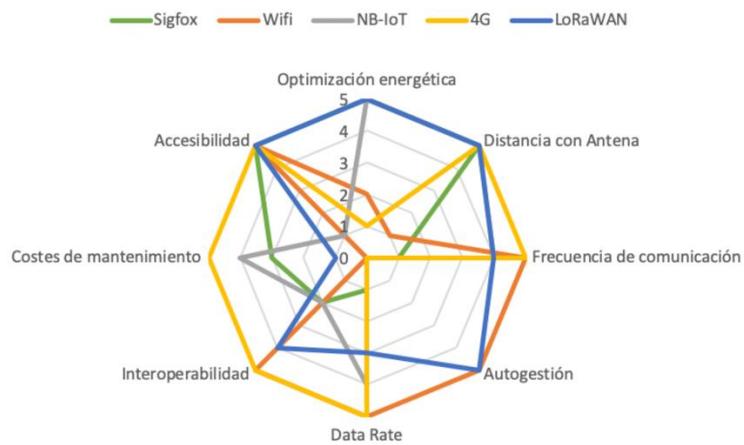


Ilustración 13 – Gráfica comparación tecnologías inalámbricas (Campos, 2020)

Actualmente no existe una tecnología que, manteniendo un bajo coste, sobresalga en los tres ejes mencionados, sino que suele sacrificarse alguno en pro de los demás



(**Ilustración 13**). Por ejemplo, el 4G tiene muy buen alcance y *data rate*, pero consume mucha energía en su transmisión. Por otro lado, el Bluetooth tiene un consumo energético muy bajo y una buena *data rate*, pero permite un alcance muy limitado. Finalmente, las LPWAN (Sigfox, NB-IoT y LoRaWAN, entre otras) son tecnologías que tienen muy buen alcance y un consumo muy pequeño, pero sacrifican la cantidad de datos que se pueden transmitir (**Ilustración 13 e Ilustración 14**). No obstante, es por esto que son las más indicadas para las aplicaciones IoT, donde se envía poca información a una larga distancia y se busca una extensa autonomía de los dispositivos.

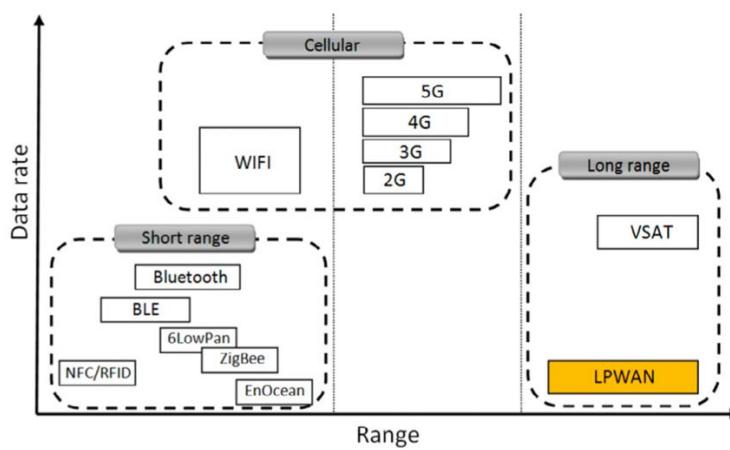


Ilustración 14 – Alcance vs ‘data rate’ en tecnologías inalámbricas (Mekki et al., 2019)

5.2.2.1. SIGFOX

Sigfox² es un operador de red LPWAN que ofrece una solución de conectividad IoT de extremo a extremo basada en sus tecnologías patentadas. Así, los dispositivos dependen de la infraestructura implementada por el operador local³ de Sigfox, por lo que necesitan una suscripción.

Actualmente, gracias a la gran infraestructura de la que dispone, Sigfox ofrece cobertura en 45 países. En la **Ilustración 15** se muestra la cobertura que proporciona en España.

Respecto a su alcance, en el estudio de Mekki et al. (2019) se indica que puede llegar a los 10 km en entorno urbano y a los 40 km en el rural.

² Sigfox ha sido adquirido por UnaBiz, firma proveedora de servicios de IoT a gran escala, pero ha mantenido su nombre como marca. (Heliot Europe, 2022)

³ Sigfox España: <https://partners.sigfox.com/companies/sigfox-spain>

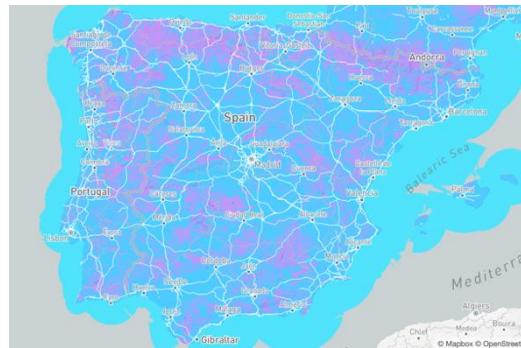


Ilustración 15 – Cobertura Sigfox en España (Sigfox, n.d.)

Sigfox implementa sus estaciones base propietarias equipadas con radios cognitivas definidas por software y las conecta a los servidores backend mediante una red basada en IP. Los dispositivos finales conectados a sus estaciones base utilizan modulación por desplazamiento de fase binaria (BPSK) en una banda ultra estrecha (UNB) de 100 Hz, lo que proporciona un bajo nivel de ruido, un bajo consumo de energía y una alta sensibilidad del receptor (Hossain & Markendahl, 2021; Mekki et al., 2019).

Por otro lado, Sigfox usa bandas Industriales, Científicas y Médicas (ISM) sin licencia, como por ejemplo 868 MHz en Europa, 915 MHz en América del Norte y 433 MHz en Asia (Mekki et al., 2019).

Debido a la regulación de las bandas sin licencia, un dispositivo puede transmitir (*uplink*) hasta 140 mensajes/día y recibir (*downlink*) hasta 4 mensajes/día (**Ilustración 16**). Estos mensajes, que se envían a una velocidad de hasta 100 bps (Mekki et al., 2019), pueden ocupar un máximo de 12 bytes para subida y 8 bytes para bajada.

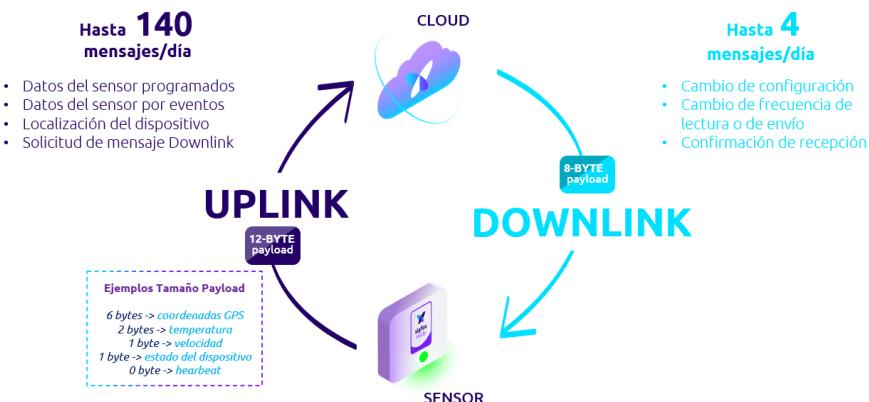


Ilustración 16 – Protocolo Sigfox (UnaBiz Spain, n.d.)



5.2.2.2. NB-IoT

Narrowband-IoT (NB-IoT) es una tecnología de acceso de radio celular (RAT) estandarizada por 3GPP (2016) en respuesta al rápido crecimiento del mercado de la conectividad de área amplia y baja potencia (GSMA, 2019).

NB-IoT trabaja en la infraestructura celular existente de *Long Term Evolution* (LTE) y *Global System for Mobile* (GSM), permitiendo a los proveedores de telecomunicaciones móviles añadir rápidamente conectividad IoT celular a sus carteras de servicios (Shea, 2017; Vos, 2022). Asimismo, opera en el espectro con licencia, teniendo bandas reservadas en el rango de los 700 MHz y 800 MHz, por ejemplo, lo que facilita la penetración en el interior de los edificios.

Según GSMA (2022), hay desplegadas 110 redes NB-IoT por todo el mundo, siendo Telefónica y Vodafone las únicas dos operadoras que proporcionan esa conectividad en España (**Ilustración 17**).



Ilustración 17 – Cobertura NB-IoT de Vodafone en España (Vodafone España, 2022)

Respecto a su alcance, en el estudio de Mekki et al. (2019) se indica que puede llegar al kilómetro en entorno urbano y a los 10 km en el rural.

NB-IoT, inicialmente conocido como Cat-NB1, ocupa un ancho de banda de frecuencia de 180 kHz, que corresponde a un bloque de recursos en la transmisión LTE (GSMA, 2019). Por otro lado, el protocolo de comunicación NB-IoT se basa en el protocolo LTE, reduciendo al mínimo sus funcionalidades y mejorándolas según sea necesario para las aplicaciones de IoT (Mekki et al., 2019). Así mismo, emplea la modulación por desplazamiento de fase cuaternaria (QPSK).



En su versión más reciente, Cat-NB2, presenta una velocidad máxima de transferencia de datos de 200 kbps. Además, tiene una cantidad ilimitada de mensajes que se pueden enviar por día con una longitud de carga útil de un máximo de 1600 bytes (Byte Lab, 2022).

5.2.2.3. LoRA/LoRaWAN

LoRa es una técnica patentada de modulación de espectro ensanchado derivada de la tecnología *Chirp Spread Spectrum* (CSS) existente. Creada por Semtech para estandarizar las redes LPWAN, LoRa ofrece una compensación entre sensibilidad y velocidad de datos mientras opera, en Europa, en un canal de ancho de banda fijo de 125 kHz o 250 kHz (canales de enlace ascendente) y 250 kHz (canales de enlace descendente) (LoRa Alliance, 2015; Semtech, 2020).

La modulación CSS propaga la señal de banda estrecha en una banda de frecuencia grande, lo que permite que la señal sea muy resistente al ruido e inmune a las interferencias. Sin embargo, en la modulación LoRa el espectro ensanchado se consigue con una señal que varía continuamente su frecuencia (Väänänen & Hämäläinen, 2022).

Construido sobre la modulación LoRa está LoRaWAN (**Ilustración 18**), un protocolo de capa de control de acceso a medios (MAC) que administra la comunicación entre los dispositivos y las puertas de enlace o *gateways*, definiendo cómo los dispositivos usan el hardware LoRa y el formato de los mensajes (The Things Network, 2021). De esta forma, LoRaWAN define el protocolo de comunicación y la arquitectura del sistema (**Ilustración 19**), mientras que LoRa define la capa física.

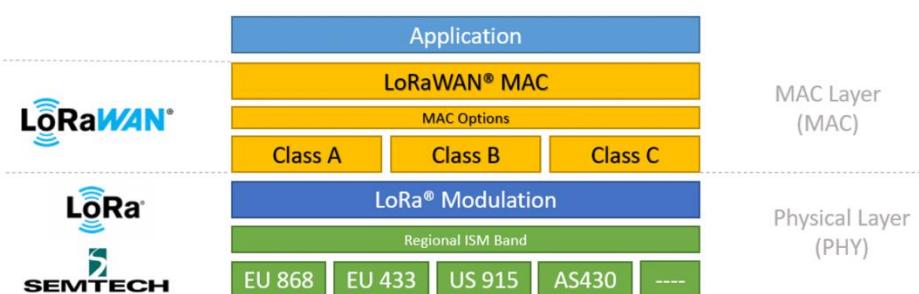


Ilustración 18 – Pila de tecnología LoRa/LoRaWAN (Semtech, 2020)

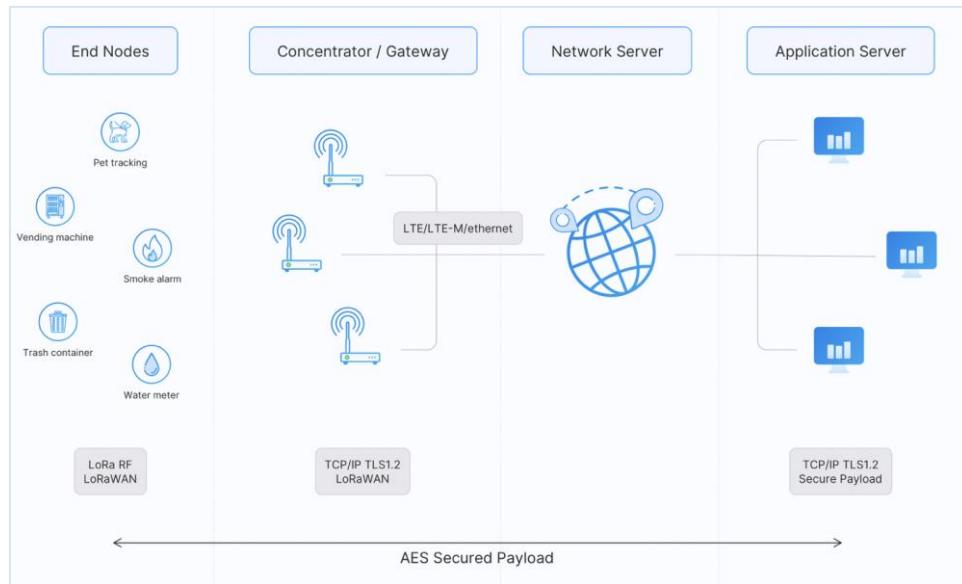


Ilustración 19 – Arquitectura típica de red LoRaWAN (The Things Network, n.d.)

Este protocolo LoRaWAN, desarrollado y mantenido por LoRa Alliance, es un estándar abierto y recientemente aprobado por la Unión Internacional de Telecomunicaciones⁴ (UIT) (LoRa Alliance, 2021).

Al igual que Sigfox, LoRa utiliza bandas ISM sin licencia, es decir, 868 MHz en Europa, 915 MHz en América del Norte y 433 MHz en Asia (Mekki et al., 2019).

Por otro lado, la modulación LoRa emplea seis factores de dispersión (SF7 a SF12) para adaptar la tasa de datos y la compensación de alcance. Un factor de dispersión más alto permite un alcance mayor a expensas de una tasa de datos más baja, y viceversa (**Ilustración 20**). Así, la velocidad de datos de LoRa está entre 300 bps y 50 kbps, según el factor de dispersión y el ancho de banda del canal (Mekki et al., 2019). Además, el protocolo LoRaWAN define un mecanismo adaptativo de tasa de datos (ADR) que optimiza la tasa utilizada, maximizando la vida útil de la batería (Väänänen & Hämäläinen, 2022).

Así mismo, la longitud máxima de carga útil para cada mensaje es de 243 bytes (Mekki et al., 2019; Semtech, 2020).

⁴ Agencia especializada de las Naciones Unidas para las tecnologías de la información y la comunicación.

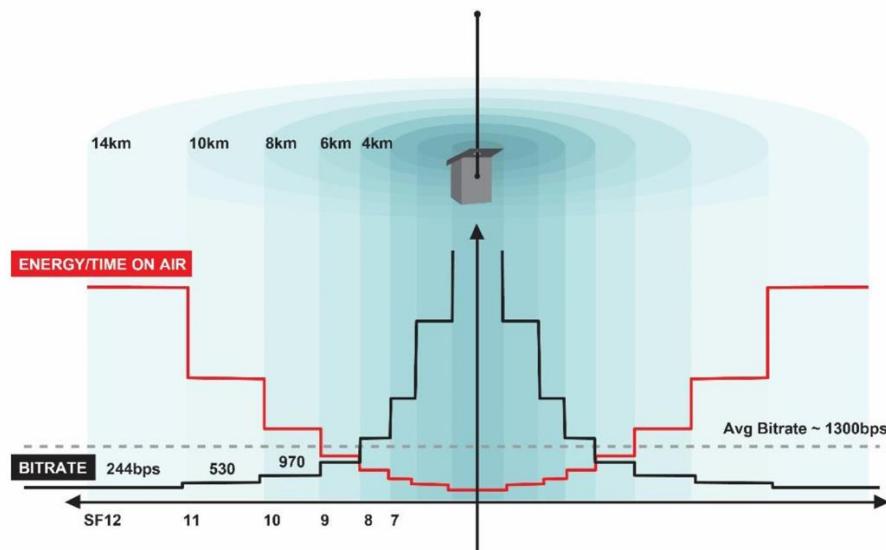


Ilustración 20 – Tasa de datos de LoRa según SF (Semtech, 2019)

A diferencia de las demás LPWAN, LoRaWAN permite el despliegue de redes propias autogestionadas y establece unos límites mucho menos restrictivos que los impuestos por un operador privado (Campos, 2020). Esos límites están marcados por el estándar ETSI EN 300 220-1 (ETSI, 2017), donde se establece un ciclo de trabajo o *duty cycle* en referencia a la fracción de tiempo que un recurso está ocupado. Por ejemplo, cuando un dispositivo transmite en un canal durante 2 unidades de tiempo cada 10 unidades de tiempo, este dispositivo tiene un *duty cycle* del 20% (The Things Network, 2020b).

El ciclo de trabajo de los dispositivos de radio suele estar regulado por los gobiernos y acostumbra a establecerse en el 1%. Cada dispositivo (nodo y *gateway*) se debe programar de tal manera que se mantenga dentro de los márgenes regulados, pero, aun así, estos límites son mucho más flexibles que en los casos anteriores.

Este *duty cycle* del 1% puede traducirse como una transmisión de 36 seg/hora por dispositivo y canal, algo más que suficiente para los objetivos del IoT.

Por otra parte, existen iniciativas cuyo principio es construir una red LoRaWAN abierta con cobertura internacional. Para ello, todos los miembros de la comunidad permiten que su infraestructura (*gateways*) pueda ser utilizada por todos. Esto significa que, si se está cerca de uno de estos *gateways* públicos, no es necesario instalar uno propio para conectarse a la red. Respecto a la distancia máxima a los *gateways*, se ofrece un alcance de hasta 5 km en entornos urbanos y 15 km en zonas rurales (Semtech, 2020).



En la **Ilustración 21** se puede ver la cobertura ofrecida en España por uno de los sistemas colaborativos más importantes: The Things Network (TTN).

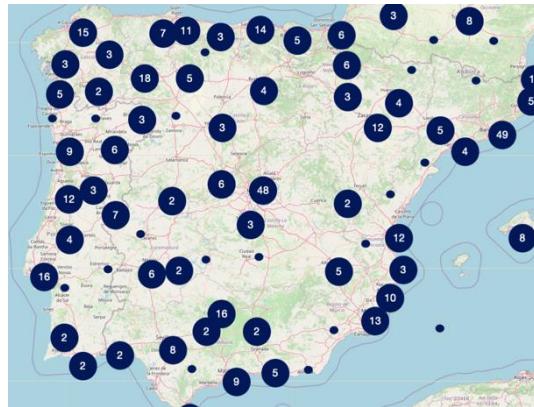


Ilustración 21 – Distribución de gateways LoRaWAN registrados en TTN en España⁵

5.2.2.4. COMPARATIVA

A partir de la información detallada, se presenta una comparativa en la **Tabla 2** que ayude a la elección de la solución más apropiada para este trabajo.

Tabla 2 – Comparativa tecnologías inalámbricas LPWAN

CARACTERÍSTICA	TECNOLOGÍA LPWAN		
	Sigfox	NB-IoT	LoRaWAN
Alcance (urbano y rural)	10 km y 40 km	1 km y 10 km	5 km y 15 km
Espectro	Sin licencia (p.ej. 868 MHz en Europa)	Con licencia (p.ej. 700 MHz)	Sin licencia (p.ej. 868 MHz en Europa)
Ancho de banda	0.1 kHz	180 kHz	125/250 kHz
Modulación	BPSK	QPSK	LoRa (CSS)
Velocidad máxima	100 bps	200 kbps	300 bps - 50 kbps
Tamaño máximo de mensaje	12 bytes (UL) y 8 bytes (DL)	1600 bytes	243 bytes
Mensajes/día	140 (UL) y 4 (DL)	Ilimitado	Ilimitado (1%)
No requiere suscripción	-	-	✓

⁵ <https://www.thethingsnetwork.org/map>



Como se puede observar, LoRaWAN es una muy buena opción ya que ofrece una amplia banda ancha y una velocidad apropiada para las soluciones IoT, además de facilitar un extenso alcance incluso en entornos urbanos.

Aun así, el factor determinante para escoger esta tecnología ha sido que los dispositivos se puedan conectar a redes LoRaWAN públicas ya establecidas, sin necesidad de desplegar una red privada ni de obtener una suscripción a un operador.



6. RECURSOS

6.1. RECURSOS HUMANOS

- **Autora:** Irene Casares Rodríguez

Alumna de 4º de Grado en Ingeniería Informática. Mención en Computadores.

- **Director:** José Manuel Palomares Muñoz

Doctor en Informática por la Universidad de Granada, actualmente profesor titular del departamento de Ingeniería Electrónica y de Computadores de la Escuela Politécnica Superior de la Universidad de Córdoba.

6.2. RECURSOS HARDWARE

Para realizar el estudio, análisis, diseño e implementación del proyecto se ha utilizado el computador personal de la autora, un MacBook Air con las siguientes características:

- SoC M1 basado en ARM (8 núcleos)
- 8GB de memoria unificada
- 256GB SSD

Con relación a la red inalámbrica, se accederá a través de la infraestructura de red LoRaWAN ofrecida por TTN (véase **Despliegue en entorno**).

6.2.1. PROTOTIPO

El prototipo de dispositivo de detección de vehículos está formado por la integración de los siguientes componentes:

- **Arduino MKR WAN 1310:**

Esta placa de código abierto (**Ilustración 22**), basada en el procesador de bajo consumo Microchip SAMD21 (Microchip & Arduino, 2018), consigue la conectividad LoRaWAN integrando el módulo Murata CMWX1ZZABZ



LoRa (Murata & Arduino, 2018). Para ampliar el rango GSM, utiliza una antena con conexión U.FL.

Además, incluye el chip criptográfico ECC508, memoria Flash SPI de 2MB y cargador de batería, teniendo un control mejorado del consumo de energía (Arduino, 2022b).

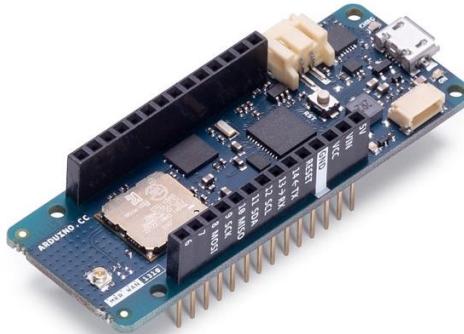


Ilustración 22 - Arduino MKR WAN 1310 (Arduino, 2022b)

- **Módulo GY-271 HMC5883L de AZ-Delivery:**

Este sensor magnético triaxial (**Ilustración 23**) integra el chip HMC5883L y utiliza el protocolo I2C para comunicarse con el Arduino, el cual lo alimenta con 3.3V.

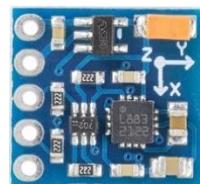


Ilustración 23 - Módulo GY-271 HMC5883L (AZ-Delivery, 2022)

- **Batería Li-Po 3.7 V 2000mAh:**

Siguiendo las especificaciones del Arduino MKR, se utilizará una batería de polímero de litio (Li-Po) de 2000mAh como la de la **Ilustración 24**. Esta batería tiene un voltaje nominal de 3.7 V, lo que da una capacidad teórica total de energía de 7.4 Wh, equivalente a 26640 Ws. Además, incluye un conector JST-PH para conectarla directamente con la placa.



Ilustración 24 - Batería LiPo 3.7V 2000mAh (EEMB & Amazon, 2022)

■ Caja de conexiones de VONVOFF:

Esta caja de conexiones (**Ilustración 25**) está fabricada con material ABS, un plástico muy resistente que no se corroa fácilmente. Se ha utilizado como carcasa envolvente del dispositivo debido a su protección IP65, que la hace hermética e impide la entrada de agua y polvo.



Ilustración 25 - Carcasa envolvente (VONVOFF & Amazon, 2022)

6.3. RECURSOS SOFTWARE

En primer lugar, el ordenador personal utilizado trabaja con macOS Monterey 12.4 (Apple Wiki, 2022) como sistema operativo.

Por otro lado, se han utilizado las siguientes herramientas software para la consecución del Proyecto:

6.3.1. HERRAMIENTAS PRINCIPALES

- **Arduino 1.8.19:** IDE que permite escribir y cargar programas en placas compatibles con Arduino (2021a).
- **Consola TTS/TTN:** aplicación web de gestión de TTS para LoRaWAN que permite registrar aplicaciones, dispositivos finales o puertas de enlace,



monitorear el tráfico de la red o configurar opciones relacionadas con la red (The Things Industries, 2022).

- **Google Apps Script:** plataforma de JavaScript basada en la nube que permite integrar y automatizar tareas en todos los productos de Google Workspace (2022), como Sheets, Docs, Gmail, etc.
- **Google Sheets:** herramienta de hojas de cálculo basada en la nube que permite analizar datos rápida y eficientemente (Google, 2022).
- **AppSheet:** plataforma de desarrollo *no code* basada en la nube que permite desarrollar aplicaciones móviles y webs a partir de hojas de cálculo y herramientas de bases de datos (Google Cloud, 2022).
- **Cayenne myDevices:** plataforma de prototipado de dispositivos IoT compatible con Arduino. (myDevices, 2022)
- **Lenguajes de programación:** Arduino (C/C++) (Arduino, 2022a) y JavaScript (Google Apps Script, 2022a; MDN Web Docs, 2022).

6.3.2. HERRAMIENTAS AUXILIARES

- **Fritzing 0.9.10:** herramienta de código abierto para la automatización de diseños electrónicos (Fritzing, 2022).
- **Diagrams.net:** herramienta online de código abierto para la creación de diagramas (diagrams.net, 2022).
- **Lucidchart:** plataforma online de diagramación (Lucidchart, 2022).
- **Microsoft Word 16.47:** software para el procesamiento de textos (Microsoft 365, 2022).



7. ESPECIFICACIÓN DE REQUISITOS

7.1. INTRODUCCIÓN

El trabajo a realizar consiste en implementar un sistema prototípico de monitorización de plazas de aparcamiento. Este sistema estaría dirigido tanto a usuarios expertos, encargados de su administración, como a usuarios no expertos, que lo utilizarían como fuente de información sobre la ocupación de las plazas.

Los problemas que se plantean son:

- Diseño de un prototípico de dispositivo de detección de vehículos.
- Implementación de una red inalámbrica para el envío y recepción de información.
- Integración de una plataforma de monitorización administrativa.
- Desarrollo de una aplicación móvil para la consulta de la información por parte de los usuarios finales.

7.2. PARTICIPANTES DEL PROYECTO

Las personas que participarán en el Proyecto son las siguientes:

Tabla 3 - Participante Irene Casares Rodríguez

Participante	Irene Casares Rodríguez
Organización	Escuela Politécnica Superior de Córdoba
Rol	Desarrollador
Comentarios	Alumna de Ingeniería Informática de la Escuela Politécnica Superior de la Universidad de Córdoba. Llevará a cabo el diseño y la implementación total del sistema prototípico, además de ejercer como usuario para las pruebas.

*Tabla 4 - Participante José Manuel Palomares Muñoz*

Participante	José Manuel Palomares Muñoz
Organización	Escuela Politécnica Superior de Córdoba
Rol	Director
Comentarios	Doctor en Informática por la Universidad de Granada, actualmente profesor titular del departamento de Ingeniería Electrónica y de Computadores de la Escuela Politécnica Superior de la Universidad de Córdoba. Soporte en el diseño del sistema.

7.3. OBJETIVOS DEL SISTEMA (OBJ)

En este apartado se tratará de indicar las particularidades del sistema, partiendo de los objetivos fundamentales para llevar a cabo su implementación:

- **OBJ-1:** Implementación del sistema de monitorización (**Tabla 5**)
- **OBJ-2:** Diseño del prototipo de dispositivo de detección (**Tabla 6**)
- **OBJ-3:** Implementación de red inalámbrica (**Tabla 7**)
- **OBJ-4:** Integración de plataforma de monitorización (**Tabla 8**)
- **OBJ-5:** Desarrollo de aplicación móvil (**Tabla 9**)
- **OBJ-6:** Diseño del algoritmo de detección (**Tabla 10**)
- **OBJ-7:** Diseño de un protocolo de comunicación de mensajes (**Tabla 11**)



Tabla 5 – OBJ-1: Implementación del sistema

OBJ-1	Implementación del sistema de monitorización
Descripción	Implementación de un sistema prototipo de monitorización de plazas de aparcamiento que ofrezca información sobre su ocupación a partir de los datos aportados por dispositivos de detección de vehículos.
Subobjetivos	OBJ-2 Diseño del prototipo de dispositivo de detección OBJ-3 Implementación de red inalámbrica OBJ-4 Integración de plataforma de monitorización OBJ-5 Desarrollo de aplicación móvil
Comentarios	Ninguno

Tabla 6 – OBJ-2: Diseño del dispositivo de detección

OBJ-2	Diseño del prototipo de dispositivo de detección
Descripción	Diseño e implementación de un prototipo de dispositivo que detecte la presencia de un vehículo, así como la iluminación de la plaza, y envíe la información a través de una red inalámbrica.
Subobjetivos	OBJ-3 Implementación de red inalámbrica OBJ-6 Diseño del algoritmo de detección OBJ-7 Diseño de un protocolo de comunicación de mensajes
Comentarios	Se priorizará el bajo consumo y coste, así como el uso de recursos <i>open-source</i> .

*Tabla 7 – OBJ-3: Implementación de red inalámbrica*

OBJ-3	Implementación de red inalámbrica
Descripción	Integrar los dispositivos de detección en una red inalámbrica a través de la cual puedan enviar y recibir información.
Subobjetivos	OBJ-7 Diseño de un protocolo de comunicación de mensajes
Comentarios	Se deberá escoger el tipo de red LPWAN más adecuada y un dispositivo para la detección que aporte esa tecnología.

Tabla 8 – OBJ-4: Integración de plataforma de monitorización

OBJ-4	Integración de plataforma de monitorización
Descripción	Integrar una plataforma que permita al administrador monitorizar la ocupación de las plazas y su histórico de datos.
Subobjetivos	Ninguno
Comentarios	Ninguno

Tabla 9 – OBJ-5: Desarrollo de aplicación móvil

OBJ-5	Desarrollo de aplicación móvil
Descripción	Desarrollar una aplicación móvil que permita a los usuarios finales conocer la ocupación de las plazas en tiempo real.
Subobjetivos	Ninguno
Comentarios	La aplicación deberá ser lo suficientemente intuitiva para que pueda utilizarla cualquier usuario, independientemente de sus conocimientos.

*Tabla 10 – OBJ-6: Diseño del algoritmo de detección*

OBJ-6	Diseño del algoritmo de detección
Descripción	Desarrollar un algoritmo que permita discernir la presencia o no de un vehículo según la información aportada por un sensor de detección de obstáculos.
Subobjetivos	Ninguno
Comentarios	Ninguno

Tabla 11 – OBJ-7: Diseño de protocolo de mensajes

OBJ-7	Diseño de un protocolo de comunicación de mensajes
Descripción	Diseñar un protocolo de mensajes que permita interpretar la información enviada y recibida por el dispositivo de detección.
Subobjetivos	Ninguno
Comentarios	Ninguno

7.4. REQUISITOS DEL SISTEMA

7.4.1. REQUISITOS FUNCIONALES (RF)

Se espera que los dispositivos prototipo cumplan los siguientes requisitos de funcionamiento:

- **RF-1:** Detectar vehículo en la plaza asignada (**Tabla 12**)
- **RF-2:** Detectar iluminación de la plaza (**Tabla 13**)
- **RF-3:** Transmitir información por red inalámbrica (**Tabla 14**)
- **RF-4:** Recibir información por red inalámbrica (**Tabla 15**)

*Tabla 12 – RF-1: Detectar vehículo*

RF-1	Detectar vehículo en la plaza asignada
Dependencias	OBJ-1 Implementación del sistema de monitorización OBJ-2 Diseño del prototipo de dispositivo de detección OBJ-6 Diseño del algoritmo de detección
Descripción	El dispositivo debe detectar la ocupación de la plaza asignada por parte de un vehículo.
Comentarios	Solo se tendrán en cuenta la ocupación por parte de vehículo de tamaño mediano como turismos o furgonetas.

Tabla 13 – RF-2: Detectar iluminación

RF-2	Detectar iluminación de la plaza
Dependencias	OBJ-2 Diseño del prototipo de dispositivo de detección
Descripción	El dispositivo debe detectar la iluminación de la plaza con relación a si está expuesto a luz solar directa o si se encuentra a la sombra.
Comentarios	Ninguno

Tabla 14 – RF-3: Transmitir por red inalámbrica

RF-3	Transmitir información por red inalámbrica
Dependencias	OBJ-1 Implementación del sistema de monitorización OBJ-2 Diseño del prototipo de dispositivo de detección OBJ-3 Implementación de red inalámbrica OBJ-7 Diseño de un protocolo de comunicación de mensajes
Descripción	El dispositivo debe transmitir y recibir información haciendo uso de una red inalámbrica.
Comentarios	Se debe utilizar el protocolo de mensajes establecido.

*Tabla 15 – RF-4: Recibir información*

RF-4	Recibir información por red inalámbrica
Dependencias	OBJ-2 Diseño del prototipo de dispositivo de detección OBJ-3 Implementación de red inalámbrica OBJ-7 Diseño de un protocolo de comunicación de mensajes
Descripción	El dispositivo debe poder recibir información de forma remota para alterar determinados parámetros.
Comentarios	Se debe utilizar el protocolo de mensajes establecido.

Respecto al sistema en general, se espera que cumpla los siguientes requisitos de funcionamiento:

- **RF-5:** Proporcionar información sobre ocupación e iluminación (**Tabla 16**)
- **RF-6:** Proporcionar ubicación de las plazas (**Tabla 17**)
- **RF-7:** Modificar parámetros del dispositivo de forma remota (**Tabla 18**)
- **RF-8:** Visualizar información (**Tabla 19**)

Tabla 16 – RF-5: Proporcionar ocupación e iluminación

RF-5	Proporcionar información sobre ocupación e iluminación
Dependencias	OBJ-1 Implementación del sistema de monitorización OBJ-2 Diseño del prototipo de dispositivo de detección OBJ-4 Integración de plataforma de monitorización OBJ-5 Desarrollo de aplicación móvil
Descripción	El sistema debe proporcionar información sobre la ocupación de las plazas y su iluminación según los datos enviados por los dispositivos.
Comentarios	Ninguno

*Tabla 17 – RF-6: Proporcionar ubicación*

RF-6	Proporcionar ubicación de las plazas
Dependencias	OBJ-1 Implementación del sistema de monitorización OBJ-4 Integración de plataforma de monitorización OBJ-5 Desarrollo de aplicación móvil
Descripción	El sistema debe proporcionar la ubicación de las plazas analizadas.
Comentarios	Ninguno

Tabla 18 – RF-7: Modificar parámetros en remoto

RF-7	Modificar parámetros del dispositivo de forma remota
Dependencias	OBJ-2 Diseño del prototipo de dispositivo de detección OBJ-3 Implementación de red inalámbrica OBJ-7 Diseño de un protocolo de comunicación de mensajes
Descripción	El sistema debe permitir la modificación remota de ciertos parámetros del dispositivo como umbrales de comparación o período de envío.
Comentarios	Se debe utilizar el protocolo de mensajes establecido.

Tabla 19 – RF-8: Visualizar información

RF-8	Visualizar información
Dependencias	OBJ-1 Implementación del sistema de monitorización OBJ-4 Integración de plataforma de monitorización OBJ-5 Desarrollo de aplicación móvil
Descripción	La visualización de la información debe realizarse a través de plataformas de monitorización y/o aplicaciones móviles.
Comentarios	Ninguno



7.4.2. REQUISITOS NO FUNCIONALES (RNF)

Se espera que el dispositivo prototipo cumpla los siguientes requisitos no funcionales:

- **RNF-1:** Información del dispositivo en tiempo real (**Tabla 20**)
- **RNF-2:** Precisión del dispositivo (**Tabla 21**)
- **RNF-3:** Alimentación del dispositivo (**Tabla 22**)
- **RNF-4:** Uso de la fuente de alimentación del dispositivo (**Tabla 23**)
- **RNF-5:** Confinamiento del dispositivo (**Tabla 24**)

Tabla 20 – RNF-1: Información del dispositivo en tiempo real

RNF-1	Información del dispositivo en tiempo real
Dependencias	OBJ-1 Implementación del sistema de monitorización OBJ-2 Diseño del prototipo de dispositivo de detección OBJ-3 Implementación de red inalámbrica
Descripción	El dispositivo debe proporcionar información en tiempo real sobre los cambios en el estado de la plaza y su iluminación.
Comentarios	Ninguno

Tabla 21 – RNF-2: Precisión del dispositivo

RNF-2	Precisión del dispositivo
Dependencias	OBJ-2 Diseño del prototipo de dispositivo de detección
Descripción	La precisión del dispositivo no debe verse afectada por las condiciones meteorológicas (p.ej. luminosidad, viento o lluvia).
Comentarios	Ninguno

*Tabla 22 – RNF-3: Alimentación del dispositivo*

RNF-3	Alimentación del dispositivo
Dependencias	OBJ-2 Diseño del prototipo de dispositivo de detección
Descripción	El dispositivo debe ser alimentado a través de una batería.
Comentarios	Ninguno

Tabla 23 – RNF-4: Uso de la alimentación del dispositivo

RNF-4	Uso de la fuente de alimentación del dispositivo
Dependencias	OBJ-2 Diseño del prototipo de dispositivo de detección
Descripción	El dispositivo debe hacer un uso eficiente de su fuente de alimentación.
Comentarios	Debe tener una autonomía que impida la interacción física reiterada con el mismo tras su instalación.

Tabla 24 – RNF-5: Confinamiento del dispositivo

RNF-5	Confinamiento del dispositivo
Dependencias	OBJ-2 Diseño del prototipo de dispositivo de detección
Descripción	El dispositivo debe estar confinado en una estructura hermética que le permita soportar diferentes condiciones meteorológicas.
Comentarios	Ninguno

Respecto al sistema en general, se espera que cumpla los siguientes requisitos no funcionales:

- **RNF-6:** Información del sistema en tiempo real (**Tabla 25**)
- **RNF-7:** Hardware y software *open-source* (**Tabla 26**)



Tabla 25 – RNF-6: Información del sistema en tiempo real

RNF-6	Información del sistema en tiempo real
Dependencias	OBJ-1 Implementación del sistema de monitorización OBJ-3 Implementación de red inalámbrica OBJ-4 Integración de plataforma de monitorización OBJ-5 Desarrollo de aplicación móvil
Descripción	El sistema debe proporcionar información sobre las plazas en tiempo real, con relación a su ocupación e iluminación.
Comentarios	Ninguno

Tabla 26 – RNF-7: Hardware y Software open-source

RNF-7	Hardware y software <i>open-source</i>
Dependencias	OBJ-1 Implementación del sistema de monitorización OBJ-2 Diseño del prototipo de dispositivo de detección OBJ-4 Integración de plataforma de monitorización OBJ-5 Desarrollo de aplicación móvil
Descripción	En el desarrollo del sistema se debe dar preferencia a hardware y software <i>open-source</i> .
Comentarios	Ninguno



7.4.3. MATRIZ DE TRAZABILIDAD (RF-RNF/OBJ)

En la **Tabla 27** se presenta la matriz de trazabilidad que permite relacionar los objetivos con los correspondientes requisitos.

Tabla 27 - Matriz de trazabilidad RF-RNF/OBJ

	OBJ-1	OBJ-2	OBJ-3	OBJ-4	OBJ-5	OBJ-6	OBJ-7
RF-1	X	X				X	
RF-2		X					
RF-3	X	X	X				X
RF-4		X	X				X
RF-5	X	X		X	X		
RF-6	X			X	X		
RF-7		X	X				X
RF-8	X			X	X		
RNF-1	X	X	X				
RNF-2		X					
RNF-3		X					
RNF-4		X					
RNF-5		X					
RNF-6	X		X	X	X		
RNF-7	X	X		X	X		



8. ANÁLISIS FUNCIONAL

En este capítulo se realiza el análisis funcional del sistema a partir de casos de uso, que describen las posibles secuencias de interacción entre el sistema y los actores externos con relación a un objetivo particular. Así, se representarán los diagramas de casos de uso para establecer un mapa general de las funcionalidades más importantes del sistema, así como los diagramas de actividades necesarios para especificar con mayor profundidad esos casos de uso.

8.1. IDENTIFICACIÓN DE ACTORES DEL SISTEMA

Los actores serán aquellas entidades externas que utilicen el sistema y demanden una funcionalidad específica del mismo, ya sean operadores humanos o materiales imprescindibles que formen parte del ámbito del sistema.

De esta forma, el sistema desarrollado contará con tres tipos de actores:

- **Dispositivo:** generalmente encargado de detectar la presencia de un vehículo y de enviar esa información.
- **Administrador:** encargado de añadir dispositivos al sistema, configurarlos y mantenerlos, así como analizar su información para realizar los cambios necesarios.
- **Usuario final:** utilizará la aplicación móvil para conocer la ocupación de las plazas y, posiblemente, tomar una de ellas.

8.2. CASOS DE USO (CU)

En este apartado se realizará una descripción de las funcionalidades que ofrecería el sistema, representadas en el diagrama de la **Ilustración 26**:

- **CU-1:** Detectar vehículo (**Tabla 28**)
- **CU-2:** Comprobar cambio de estado (**Tabla 29**)
- **CU-3:** Detectar iluminación (**Tabla 30**)
- **CU-4:** Enviar mensaje (dispositivo) (**Tabla 31**)



- **CU-5:** Recibir mensaje (**Tabla 32**)
- **CU-6:** Añadir dispositivo (**Tabla 33**)
- **CU-7:** Eliminar dispositivo (**Tabla 34**)
- **CU-8:** Enviar mensaje (administrador) (**Tabla 35**)
- **CU-9:** Monitorizar información (**Tabla 36**)
- **CU-10:** Consultar aplicación (**Tabla 37**)
- **CU-11:** Ocupar plaza (**Tabla 38**)
- **CU-12:** Liberar plaza (**Tabla 39**)

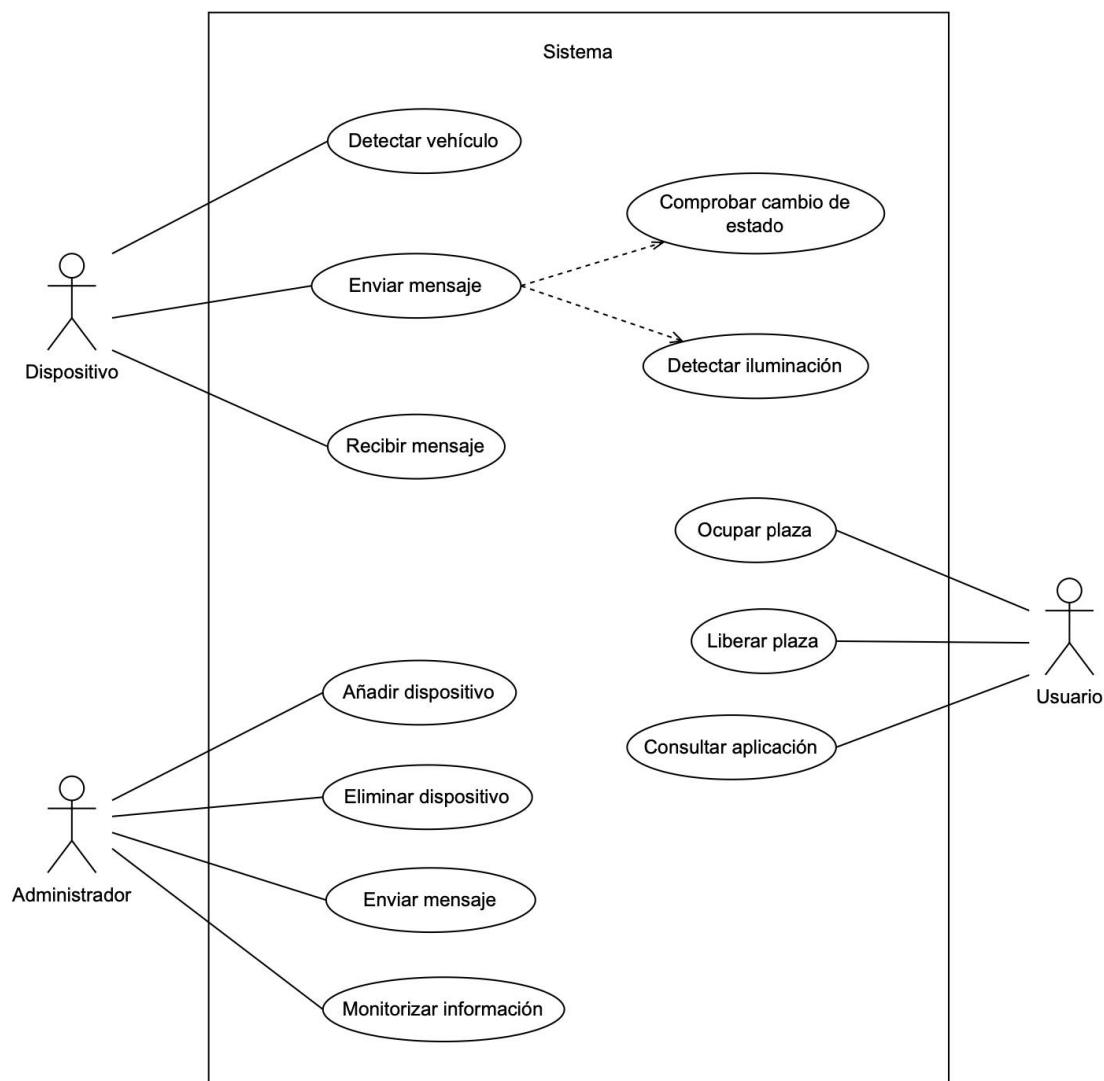


Ilustración 26 - Diagrama de Casos de Uso



Tabla 28 - CU-1: Detectar vehículo

CU-1	Detectar vehículo	
Dependencias	CU-2 Comprobar cambio de estado	
Descripción	El dispositivo detecta la presencia o no de un vehículo en la plaza.	
Actores	Dispositivo	
Pre-condición	Ninguna	
Flujo principal	Paso	Acción
	1	El dispositivo percibe el campo magnético terrestre.
	2	El dispositivo detecta una variación y determina que la plaza está OCUPADA.
	3	Se realiza el caso de uso <i>Comprobar cambio de estado</i> .
Post-condición	Estado de la plaza actualizado en el dispositivo.	
Flujo alternativo	Paso	Acción
	2	El dispositivo no capta variación en el campo magnético terrestre.
	2.1	El dispositivo determina que la plaza está LIBRE.



Tabla 29 - CU-2: Comprobar cambio de estado

CU-2	Comprobar cambio de estado	
Dependencias	CU-1 Detectar vehículo CU-3 Enviar mensaje	
Descripción	El dispositivo comprueba si ha habido un cambio en el estado de la plaza.	
Actores	Dispositivo	
Pre-condición	El dispositivo ha leído el estado de la plaza, detectando o no un vehículo.	
Flujo principal	Paso	Acción
	1	El dispositivo comprueba si el estado determinado es diferente al de la lectura anterior.
	2	Si el estado es diferente, 2.1 Se realiza el caso de uso <i>Enviar mensaje</i> .
Post-condición	Ninguna	
Flujo alternativo	Ninguno	

Tabla 30 - CU-3: Detectar iluminación

CU-3	Detectar iluminación	
Dependencias	Ninguna	
Descripción	El dispositivo detecta la iluminación de la plaza.	
Actores	Dispositivo	
Precondiciones	Han pasado 15 minutos desde la última lectura de la iluminación.	
Flujo	Paso	Acción
	1	El dispositivo detecta la luz incidente en la plaza.
	2	Se realiza el caso de uso <i>Enviar mensaje (dispositivo)</i> .
Postcondiciones	Información de iluminación actualizada.	
Flujo alternativo	Ninguno	



Tabla 31 - CU-4: Enviar mensaje (dispositivo)

CU-4	Enviar mensaje (dispositivo)	
Dependencias	CU-1 Detectar vehículo CU-2 Comprobar cambio de estado CU-3 Detectar iluminación	
Descripción	El dispositivo envía un mensaje al servidor de red.	
Actores	Dispositivo	
Precondiciones	Ha habido un cambio de estado de la plaza (ocupado/libre) o de la iluminación (sol/sombra).	
Flujo	Paso	Acción
	1	El dispositivo conforma el mensaje según el protocolo establecido.
	2	El dispositivo envía el mensaje <i>uplink</i> al servidor de red.
Postcondiciones	Información de la plaza actualizada en el sistema.	
Flujo alternativo	Ninguno	



Tabla 32 - CU-5: Recibir mensaje

CU-5	Recibir mensaje	
Dependencias	CU-8 Enviar mensaje (administrador)	
Descripción	El dispositivo recibe un mensaje <i>downlink</i> del servidor de red.	
Actores	Dispositivo	
Precondiciones	El administrador ha enviado un mensaje al dispositivo.	
Flujo	Paso	Acción
	1	El dispositivo recibe el mensaje, lo decodifica y lo procesa.
	2	El dispositivo actúa según la información recibida:
	2.a	Toma los valores actuales del campo magnético como referencia de plaza LIBRE.
	2.b	Modifica los umbrales de comparación.
	2.c	Añade el estado de la batería al próximo mensaje.
	2.d	Modifica el período de envío.
Postcondiciones	Se han realizado las modificaciones solicitadas.	
Flujo alternativo	Ninguno	



Tabla 33 - CU-6: Añadir dispositivo

CU-6	Añadir dispositivo	
Dependencias	Ninguna	
Descripción	El administrador añade un dispositivo de detección al sistema.	
Actores	Administrador	
Precondiciones	El administrador ha accedido a la consola del servidor de red.	
Flujo	Paso	Acción
	1	El administrador programa el dispositivo para que se conecte al sistema.
	2	El administrador instala el dispositivo en una plaza determinada.
	3	El administrador configura el dispositivo en el servidor de red, indicando su ubicación.
	4	El administrador configura el dispositivo en la plataforma de monitorización.
	5	El administrador configura el dispositivo en la aplicación móvil.
	6	El administrador indica al dispositivo que tome valores de referencia LIBRE.
Postcondiciones	Nuevo dispositivo instalado y configurado en el sistema.	
Flujo alternativo	Ninguno	



Tabla 34 - CU-7: Eliminar dispositivo

CU-7	Eliminar dispositivo	
Dependencias	Ninguna	
Descripción	El administrador elimina un dispositivo del sistema.	
Actores	Administrador	
Precondiciones	El dispositivo estaba integrado en el sistema.	
Flujo	Paso	Acción
	1	El administrador elimina el dispositivo de la aplicación móvil.
	2	El administrador elimina el dispositivo de la plataforma de monitorización.
	3	El administrador elimina el dispositivo de la aplicación en el servidor de red.
	4	El administrador desinstala el dispositivo de su lugar en la plaza.
Postcondiciones	Dispositivo eliminado del sistema.	
Flujo alternativo	Ninguno	



Tabla 35 - CU-8: Enviar mensaje (administrador)

CU-8	Enviar mensaje (administrador)	
Dependencias	CU-5 Recibir mensaje	
Descripción	El administrador envía un mensaje a un dispositivo del sistema.	
Actores	Administrador	
Precondiciones	El dispositivo ha enviado un mensaje <i>uplink</i> al servidor de red y se ha abierto la ventana de mensajes <i>downlink</i> .	
Flujo	Paso	Acción
	1	El administrador accede a la consola del servidor de red.
	2	El administrador conforma el mensaje según el protocolo establecido.
	3	El administrador envía el mensaje desde la consola.
Postcondiciones	Mensaje <i>downlink</i> enviado.	
Flujo alternativo	Ninguno	



Tabla 36 - CU-9: Monitorizar información

CU-9	Monitorizar información				
Dependencias	CU-5 Enviar mensaje (dispositivo)				
Descripción	El administrador visualiza la información del sistema en una plataforma de monitorización.				
Actores	Administrador				
Precondiciones	El administrador tiene acceso a la plataforma de monitorización.				
Flujo	Paso	Acción			
	1	El administrador accede a la plataforma de monitorización.			
	2	El administrador visualiza la información en la plataforma.			
	2.1	El administrador realiza análisis mediante gráficas.			
Postcondiciones	Ninguna				
Flujo alternativo	Ninguno				

Tabla 37 - CU-10: Consultar aplicación

CU-10	Consultar aplicación		
Dependencias	Ninguna		
Descripción	El usuario consulta la aplicación para conocer la ocupación de las plazas de aparcamiento.		
Actores	Usuario		
Precondiciones	El usuario se ha descargado la aplicación móvil.		
Flujo	Paso	Acción	
	1	El usuario accede a la aplicación.	
	2	El usuario consulta el mapa.	
	2.1	El usuario consulta información específica de cada plaza.	
Postcondiciones	Ninguna		
Flujo alternativo	2	El usuario consulta el listado de plazas.	



Tabla 38 - CU-11: Ocupar plaza

CU-11	Ocupar plaza	
Dependencias	Ninguna	
Descripción	El usuario ocupa una plaza de aparcamiento.	
Actores	Usuario	
Precondiciones	La plaza estaba libre.	
Flujo	Paso	Acción
	1	El usuario coloca su vehículo en la plaza, sobre el dispositivo de detección.
Postcondiciones	El dispositivo detecta el vehículo y actualiza el estado de la plaza.	
Flujo alternativo	Ninguno	

Tabla 39 - CU-12: Liberar plaza

CU-12	Liberar plaza	
Dependencias	Ninguna	
Descripción	El usuario libera una plaza de aparcamiento.	
Actores	Usuario	
Precondiciones	La plaza estaba ocupada.	
Flujo	Paso	Acción
	1	El usuario mueve su vehículo fuera de la superficie de la plaza.
Postcondiciones	El dispositivo detecta que ya no hay un vehículo sobre él y actualiza el estado de la plaza.	
Flujo alternativo	Ninguno	



8.2.1. MATRIZ DE TRAZABILIDAD (RF/CU)

En la **Tabla 40** se presenta la matriz de trazabilidad que permite relacionar los casos de uso con los correspondientes requisitos funcionales.

Tabla 40 - Matriz de trazabilidad RF/CU

	CU-1	CU-2	CU-3	CU-4	CU-5	CU-6	CU-7	CU-8	CU-9	CU-10	CU-11	CU-12
RF-1	X	X									X	X
RF-2			X									
RF-3				X								
RF-4					X				X			
RF-5	X		X	X						X	X	
RF-6						X	X					
RF-7					X				X			
RF-8										X	X	

8.3. DIAGRAMA DE ACTIVIDADES

El Diagrama de Actividades es un diagrama de flujo del proceso multi-propósito que se usa para modelar el comportamiento dinámico de un sistema o sus componentes.

En la **Ilustración 27** se presenta el diagrama de las actividades realizadas por el dispositivo, especificadas previamente mediante los Casos de Uso, debido a su mayor complejidad respecto a las demás funcionalidades.

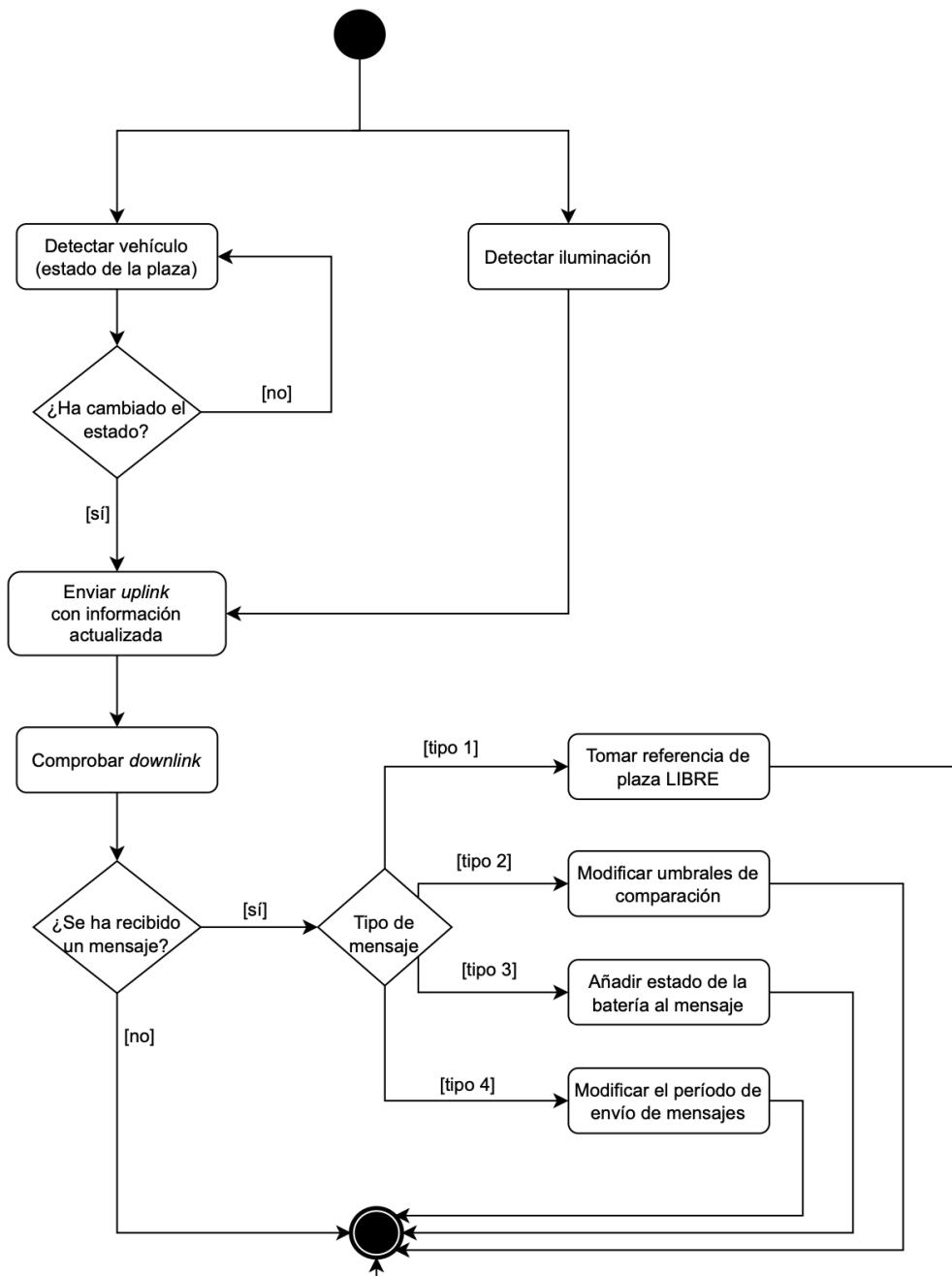


Ilustración 27 - Diagrama de Actividades



9. DISEÑO DEL SISTEMA

En este capítulo se define la arquitectura del sistema, así como especificaciones de diseño de sus componentes.

9.1. DISEÑO ESTRUCTURAL

Para especificar y documentar los aspectos estáticos del sistema se utilizarán diagramas estructurales de UML como los diagramas de clases.

9.1.1. DIAGRAMA DE CLASES

Los Diagramas de Clases describen la estructura del sistema mostrando sus clases, atributos, operaciones y las relaciones entre los objetos.

En la **Ilustración 28** es muestra el diagrama de clases del sistema a alto nivel, definiendo los objetos que intervienen en el mismo y sus relaciones.

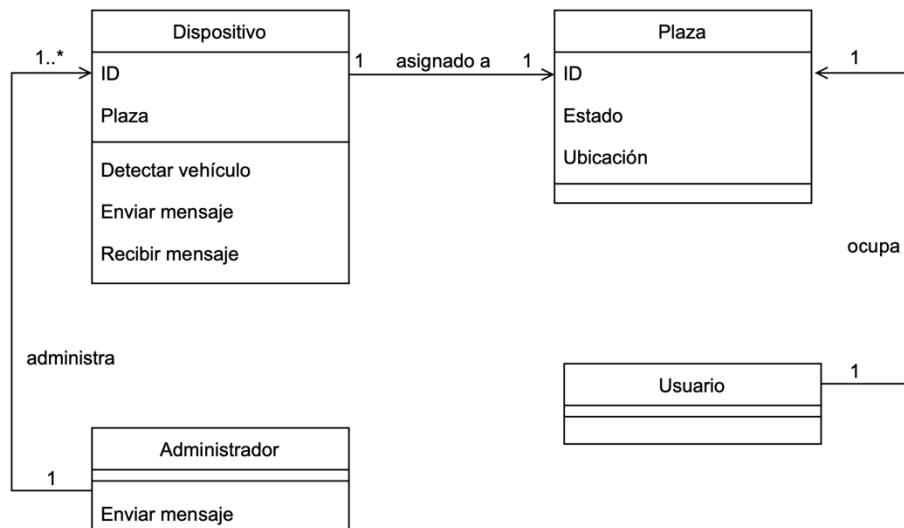


Ilustración 28 - Diagrama de Clases



9.2. DISEÑO DE LA ARQUITECTURA DEL SISTEMA

Haciendo uso de una red LoRaWAN pública (TTN), la Arquitectura del Sistema quedaría como en la **Ilustración 29**.

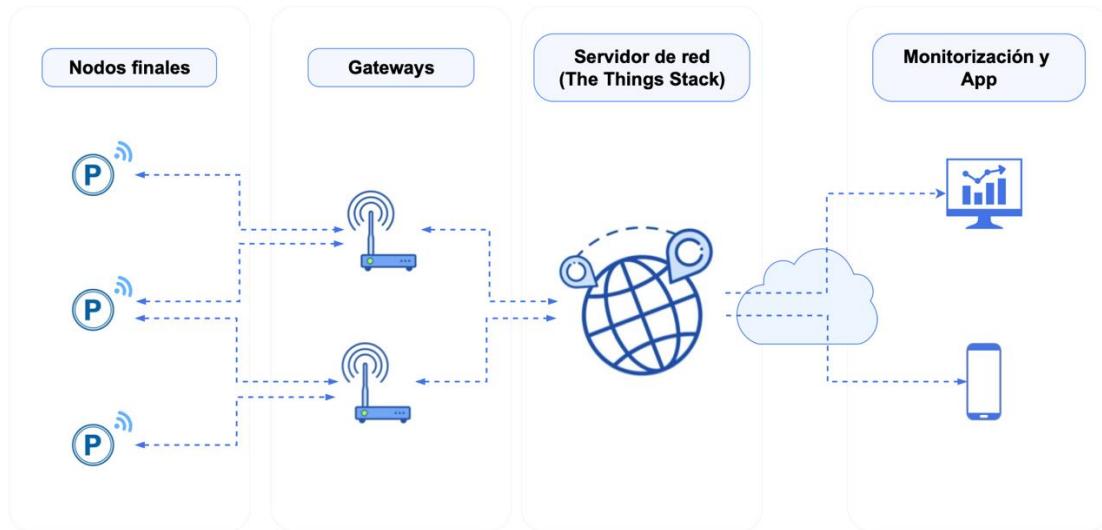


Ilustración 29 - Arquitectura del Sistema

En primer lugar, se encuentran los **dispositivos o nodos finales**, que integran el sensor magnético. Estos se encargarán de recoger la información referente a la ocupación del aparcamiento, codificarla y enviarla (mensaje *uplink*) al servidor de red LoRaWAN a través de *gateways* de radio que utilicen la modulación LoRa. Cada paquete enviado por un nodo sería recibido por todos los *gateways* a su alcance, reduciendo significativamente la tasa de errores de paquetes.

Por otro lado, estos nodos finales podrían recibir información (mensaje *downlink*) enviada desde el servidor de red, permitiendo la interacción remota con el administrador para modificar ciertos parámetros de su configuración, como las variables de comparación o la frecuencia de envío.

En segundo lugar, se encuentran los **gateways LoRaWAN**, que reenviarían los mensajes de radio LoRa recibidos. Estos elementos funcionan completamente en la capa física, enlazando la información hasta y desde el servidor de red a través de Wi-Fi, Ethernet o una conexión celular.



Al recibir un paquete desde un nodo final, los *gateways* comprobarían la integridad de los datos mediante el CRC y lo reenviarán hasta el servidor de red LoRa junto con algunos metadatos como el nivel RSSI de recepción del mensaje. Por otro lado, al recibir mensajes *downlink* desde el servidor de red, lo reenviarían directamente hasta el nodo correspondiente sin realizar ninguna interpretación de la carga útil o *payload* (Semtech, 2020).

Al participar de un sistema colaborativo como TTN, si se estuviera en el radio de alcance de alguna de estas puertas de enlace, no haría falta instalar ni configurar otras, sino que se utilizarían esos mismos *gateways* públicos.

En tercer lugar, se encuentra el **servidor de red** *The Things Stack* (TTS), que administra toda la red controlando dinámicamente sus parámetros para adaptar el sistema a las condiciones variables, además de establecer conexiones AES seguras de 128 bits para el transporte de datos de extremo a extremo. Asimismo, aunque garantiza la autenticidad de todos los dispositivos y la integridad de sus mensajes, no puede acceder a los datos transmitidos (Semtech, 2020).

Respecto a los mensajes *uplink* recibidos en el servidor, y dado que varios *gateways* pueden recibir el mismo paquete desde un único dispositivo, el servidor de red elimina las copias innecesarias. Por otro lado, al enviar mensajes *downlink* hasta un nodo final, selecciona el *gateway* con mejor nivel de RSSI de entre todos los que previamente recibieron mensajes del nodo, pues este será el que esté más cerca del dispositivo final en cuestión (Semtech, 2020).

Este elemento tiene otras funcionalidades, como reenviar *payloads* de mensajes *uplink* a los servidores de aplicaciones correspondientes o reenviar *payloads* de mensajes *downlink* provenientes de cualquier servidor de aplicaciones a cualquier dispositivo conectado a la red (Semtech, 2020).

Por último, se encuentran la **plataforma de monitorización** y la **aplicación móvil**, dirigida a los usuarios finales.



9.3. DISEÑO DEL ALGORITMO DE DETECCIÓN DE VEHÍCULOS

Para la detección de vehículos mediante los dispositivos sensores, se seguiría la metodología presentada por Sifuentes de la Hoya et al. (2015), la cual implica percibir cambios en el campo electromagnético de la Tierra.

Debido al material ferromagnético presente en el chasis inferior del vehículo, su permeabilidad magnética es mayor al aire que lo rodea. Esto distorsiona las líneas de flujo del campo magnético terrestre (**Ilustración 30**), lo que puede ser detectado por un sensor magnético situado bajo el vehículo.

Para detectar esa distorsión hay que tener en cuenta que el magnetómetro solo mide el campo magnético actual, por lo que se debería tomar un valor como referencia cuando la plaza esté libre. Este valor se compararía con los que se vayan tomando posteriormente para saber si la plaza se ha ocupado o si ha dejado de estarlo.

La dificultad está en que el campo electromagnético de la Tierra no es estable, ya que es generado por el movimiento de aleaciones de hierro fundido en el núcleo externo. Es por eso que no se pueden comparar directamente dos medidas tomadas por el magnetómetro, sino que habría que establecer un umbral que marcaría cuándo se podría considerar que son diferentes.

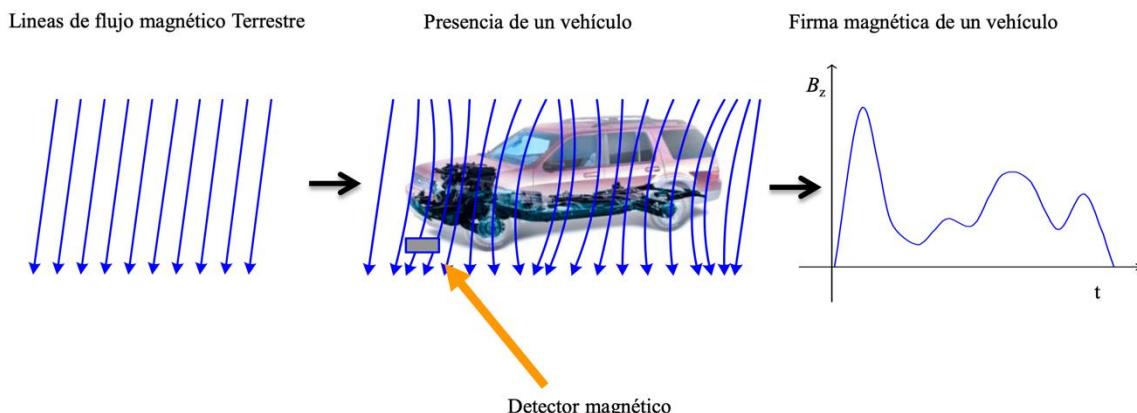


Ilustración 30 - Metodología para detectar vehículos mediante un magnetómetro (Sifuentes de la Hoya et al., 2015)



Como se mencionó previamente al analizar los **Métodos de detección**, los valores medidos por el magnetómetro se pueden suponer como un vector con componentes en los ejes X, Y y Z; cuyo módulo se puede obtener a partir de $\sqrt{X^2 + Y^2 + Z^2}$. Por otro lado, para que dos vectores sean iguales matemáticamente deben tener el mismo módulo, dirección y sentido.

Por eso, la mejor forma de comparar la referencia con los sucesivos vectores obtenidos sería a través de la diferencia entre sus módulos y del ángulo que forman. Si alguno de estos valores fuera mayor que el umbral establecido, los vectores serían lo bastante diferentes como para considerar que ha habido una gran variación del campo magnético debida a la presencia de un vehículo.

Así, el algoritmo de detección de vehículos seguiría el diagrama de flujo de la **Ilustración 31.**

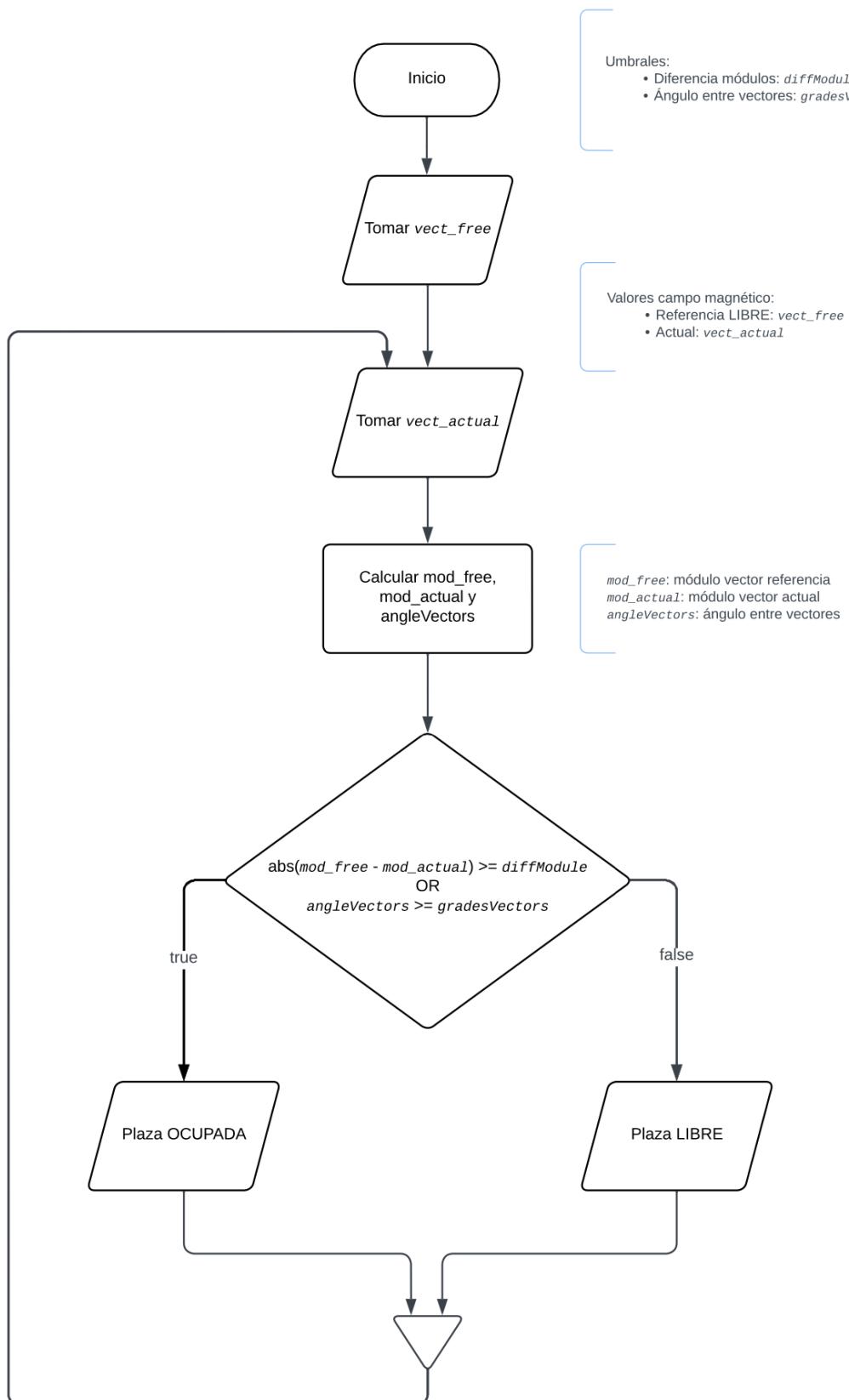


Ilustración 31 - Algoritmo detección de vehículos



9.4. DISEÑO DE LA INTERFAZ

En este apartado se presenta el diseño de las interfaces de usuario (UI) de la plataforma de monitorización y de la aplicación móvil, que deben ser simples y con usabilidad, permitiendo una interacción intuitiva, clara y concisa.

9.4.1. INTERFAZ DE LA PLATAFORMA DE MONITORIZACIÓN

En la **Ilustración 32** se exhibe el diseño del tablero general de monitorización, en el que se muestra un mapa con las ubicaciones de los dispositivos y los estados de las plazas.

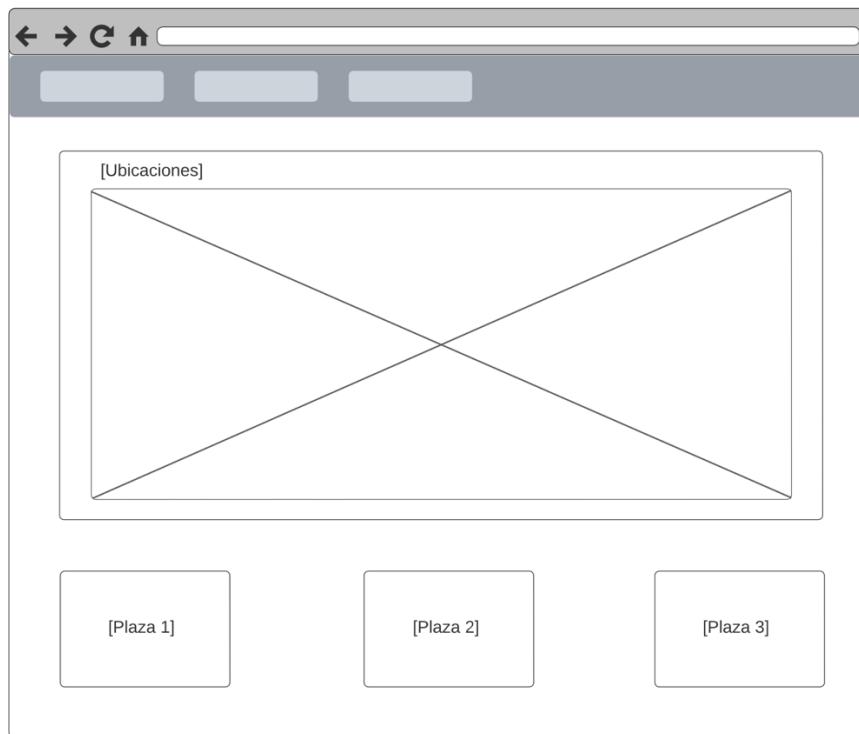


Ilustración 32 - Diseño UI plataforma monitorización: Tablero general

Por otro lado, en la **Ilustración 33** se muestra la vista individual de cada dispositivo con su información.

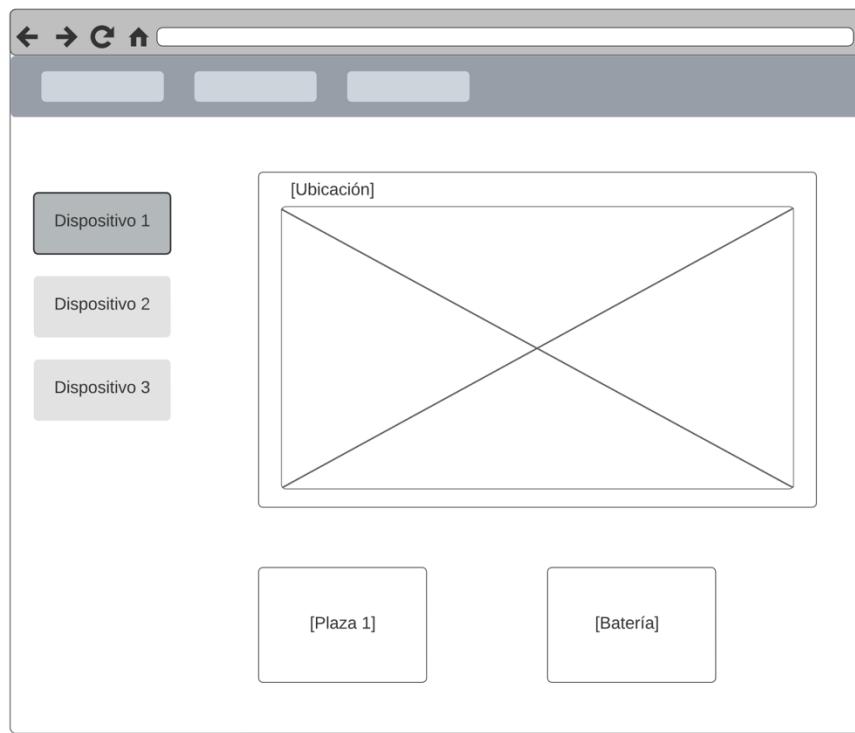


Ilustración 33 - Diseño UI plataforma monitorización: Información dispositivo

9.4.2. INTERFAZ DE LA APLICACIÓN MÓVIL

En la **Ilustración 34** se presenta el diseño de la pantalla principal, en la que se muestra directamente el mapa donde se situarían las ubicaciones de las plazas.

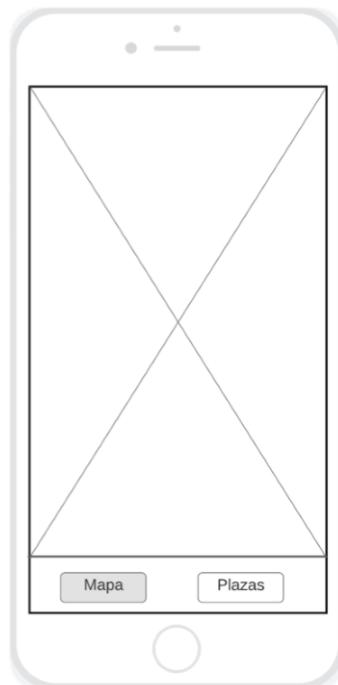


Ilustración 34 – Diseño UI aplicación móvil: Pantalla principal (Mapa)



Por otro lado, en la **Ilustración 35** se muestra la vista del listado de plazas, con su información de estado y un acceso directo a su ubicación.

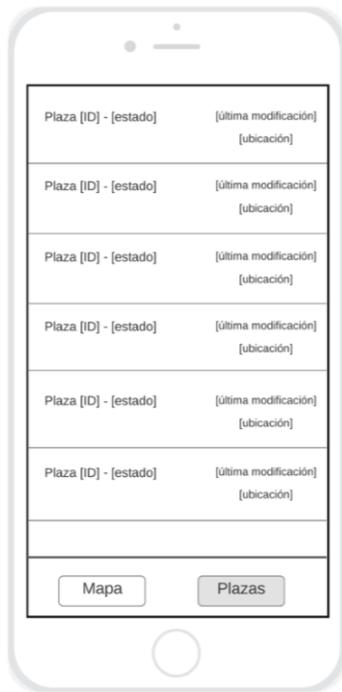


Ilustración 35 – Diseño UI aplicación móvil: Listado de plazas



10. PRUEBAS

La fase de pruebas es una parte fundamental en el desarrollo de cualquier sistema, pues determinará la calidad del mismo. En el desarrollo del sistema se realizan un conjunto de controles que permiten detectar errores cuanto antes, ya que a medida que el desarrollo avanza, cualquier corrección podría ser mucho más costosa tanto en tiempo como en esfuerzo.

10.1. PRUEBAS UNITARIAS

Considerando este sistema como un conjunto de componentes que interactúan entre sí, es necesario probar estos componentes por separado para comprobar que cada uno de ellos funciona correctamente y que produce unos resultados satisfactorios.

10.1.1. PRUEBAS DE CAJA BLANCA

Las pruebas de caja blanca presentan un enfoque estructural y buscan la validación interna de los componentes del sistema analizando sus caminos de ejecución.

Con relación al dispositivo de detección, se procurará testear cada una de las líneas de código que conformen su programa. Las combinaciones que se pueden llevar a cabo en la ejecución del programa pueden ser numerosas debido a la existencia de bucles, sentencias condicionales, etc. Sin embargo, es necesario realizar una cobertura de sentencias que demuestre que cada parte del código se ejecuta correctamente.

De esta forma, a medida que se implementen las distintas rutinas de ejecución, se comprobará su eficiencia y estructura para asegurar que no producen ningún tipo de error. Estas pruebas se realizarán de la siguiente forma:

- Se comprobará si se almacena y trata correctamente toda la información contenida en las estructuras internas de datos.
- Se comprobará que el flujo de información se realiza correctamente para todas las ramas posibles.
- Se probarán los límites de los bucles.



- Se comprobará la ejecución de cada una de las sentencias que formen parte del programa, así como la ejecución completa del mismo.

10.1.2. PRUEBAS DE CAJA NEGRA

Las pruebas de caja negra presentan un enfoque funcional y buscan la validación externa de los componentes del sistema. Se apoyan en la especificación de los requisitos y se centran en las salidas reconocibles desde el sistema y sus componentes individuales. Es decir, intentan encontrar casos en los que un componente del sistema no se atenga a su especificación.

Con relación al dispositivo de detección, para comprobar si cumple sus funcionalidades:

- Se revisará la idoneidad de la metodología de detección de vehículos, explicada en el **Diseño del Algoritmo de Detección de Vehículos**, colocando diferentes vehículos sobre el dispositivo y observando si establece la plaza como ocupada o libre cuando sea necesario.
- Se comprobará el envío y recepción de mensajes, analizando si el dispositivo modifica su comportamiento cuando se le indique.
- Se analizará si los umbrales de comparación establecidos inicialmente permiten una detección sin falsos positivos ni negativos, actuando sobre la precisión del dispositivo.
- Se comprobará si detecta la iluminación en la plaza, estimando si se encuentra directamente bajo luz solar o a la sombra.

En apoyo de sus funcionalidades, se analizará en qué posición se deberá colocar el dispositivo, ya que autores como Lou et al. (2020) insisten en la importancia de su posición bajo el vehículo, pues la diferencia en la cantidad de material ferromagnético influye en la precisión del magnetómetro.

Por ello, y siguiendo el ejemplo de varios estudios (Floris et al., 2022; Lou et al., 2020; Sifuentes de la Hoya et al., 2015), se evaluarán los valores obtenidos situando el dispositivo bajo un vehículo en 3 posiciones diferentes: eje delantero, centro del vehículo y eje trasero.



Por otro lado, respecto a la plataforma de monitorización integrada:

- Se comprobará la visualización correcta y en tiempo real de los datos de los dispositivos.
- Se comprobará la funcionalidad de análisis gráficos con los datos históricos.

Finalmente, respecto a la aplicación móvil diseñada para los usuarios finales:

- Se comprobará la visualización correcta y en tiempo real de los datos de los dispositivos.
- Se comprobará la rapidez de interacción con los usuarios, sin problemas de sincronización.

10.2. PRUEBAS DE INTEGRACIÓN

El objetivo de las pruebas de integración es la validación de los flujos de datos entre los componentes del sistema. La estrategia seguida será la integración incremental, combinando el siguiente componente a probar con aquellos ya validados.

Comenzando por el dispositivo de detección y alcanzando la aplicación móvil, estas pruebas se realizarán de la siguiente forma:

- En cada paso incremental se comprobará que la interacción entre el componente añadido y los ya probados funciona correctamente.
- Se comprobará que el flujo de información entre los componentes se realiza correctamente.

10.3. PRUEBAS DEL SISTEMA

Tras la integración de los componentes, se debe validar el sistema en su conjunto para comprobar que se cumplen todos los requisitos funcionales, que el funcionamiento y rendimiento son adecuados, y que se realiza una ejecución correcta en condiciones límite y de sobrecarga.

Los casos de prueba del sistema pueden ser casos basados en los requisitos y obtenidos gracias a técnicas de caja negra aplicadas a las especificaciones. En este proyecto se utilizarán las propias pruebas de caja negra realizadas sobre la plataforma de



monitorización y la aplicación móvil para comprobar las funcionalidades generales del sistema.



11. IMPLANTACIÓN

En este apartado se describe en detalle la implantación del sistema realizado, que se ha dividido principalmente en cinco partes:

- Diseño del prototipo de dispositivo sensor inalámbrico
- Programación del dispositivo
- Conexión a red LoRaWAN
- Integración con plataforma de monitorización
- Creación de prototipo de aplicación de usuario para muestra de datos

11.1. PROTOTIPO DE DISPOSITIVO

El dispositivo desarrollado ha sido concebido como un prototipo para probar el concepto del sistema en cuestión. Así pues, no es una versión apta para la producción final.

A continuación, se detalla el diseño del prototipo de dispositivo, centrándose en el hardware y el software desarrollado para su control.

11.1.1. HARDWARE

Como se indica en el apartado de **Recursos**, se ha utilizado el Arduino MKR WAN 1310 a modo de placa de desarrollo y el módulo GY-271 HMC5883L de AZ-Delivery como magnetómetro. Integrando ambos elementos se obtiene el prototipo final, que es alimentado por una batería Li-Po de 3.7 V y 2000 mAh.

En la **Ilustración 36** y en la **Tabla 41** se muestran, respectivamente, el esquema y la tabla de conexiones del dispositivo.

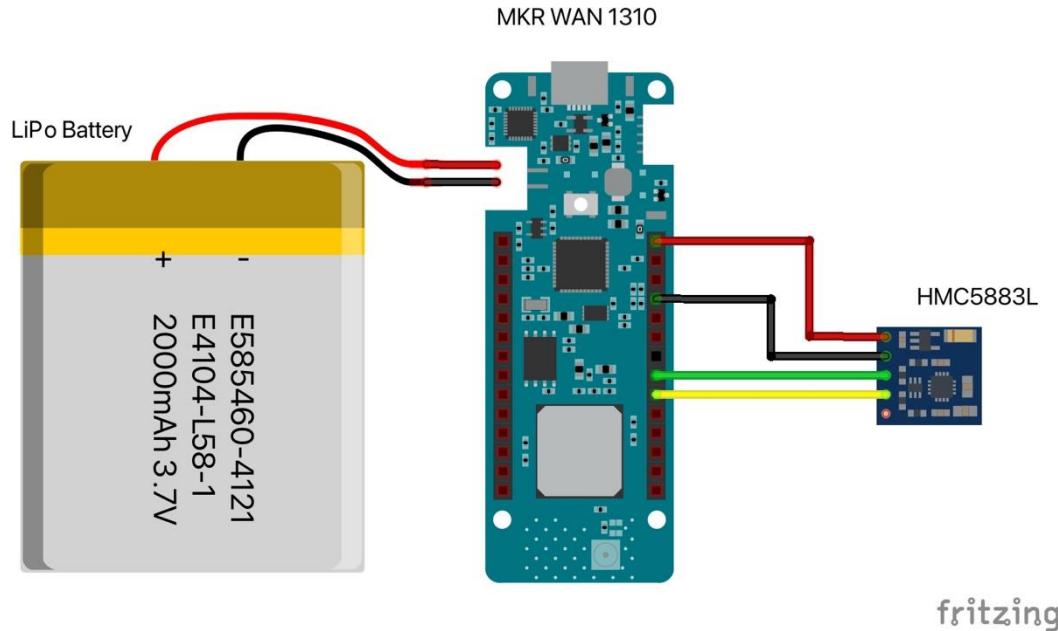


Ilustración 36 - Esquema de conexión del dispositivo⁶

Tabla 41 - Tabla de conexiones del dispositivo

Arduino	Magnetómetro
5V	Vcc
GND	GND
D12 (SCL)	SCL
D11 (SDA)	SDA

En la **Ilustración 37** se muestra una imagen real del prototipo utilizado, donde ya se incorpora la antena incluida con el Arduino MKR para potenciar su alcance.

⁶ Realizado con la herramienta software Fritzing a partir del diseño *open-source* del MKR, ofrecido por Arduino (2022), y el del HMC5883L, ofrecido por la comunidad (https://github.com/leico/Boladge_HMC5883L_Digital_Compass_Fritzing).

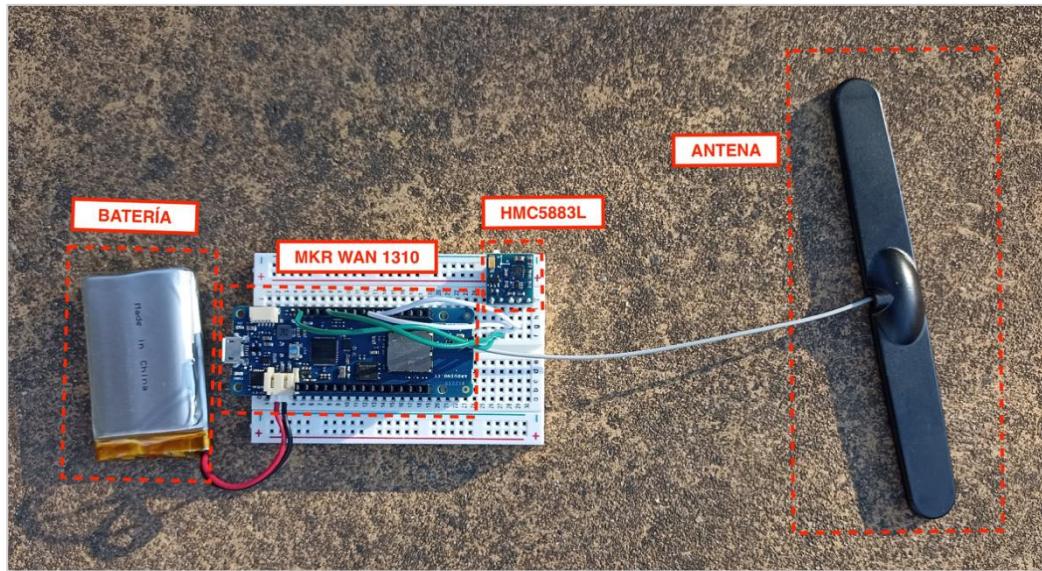


Ilustración 37 - Prototipo de dispositivo

11.1.2. SOFTWARE

Tal y como se ha comentado en el **Diseño de la Arquitectura del Sistema**, este dispositivo actúa como nodo final, por lo que debe detectar la presencia del vehículo, codificar la información y enviarla a la red LoRaWAN. Además, debe tratar los mensajes recibidos desde el servidor de red y actuar en consecuencia.

■ Lectura del campo magnético:

En primer lugar, para la obtención de datos con el magnetómetro HMC5883L se ha utilizado la librería Adafruit_HMC5883_U.h de Adafruit (2020). Esta librería, basada en la Adafruit_Sensor.h, también de (Adafruit, 2022), proporciona una capa de abstracción entre la aplicación y el sensor, permitiendo el uso inmediato de datos en unidades del SI.

■ Detección de vehículos:

Para la detección de vehículos se ha implementado el algoritmo diseñado previamente en **Diseño del Algoritmo de Detección de Vehículos**. Como ya se ha comentado, en la aplicación de este algoritmo son necesarios unos valores umbrales que permitan determinar si el campo magnético terrestre ha variado lo suficiente como para considerar que hay un vehículo ocupando la plaza.



Inicialmente, para establecer unos valores umbrales que permitieran comparar la referencia con las sucesivas lecturas, se analizaron las pequeñas variaciones producidas debido a las propias fluctuaciones del núcleo terrestre.

Estas variaciones se pueden ver en la **Ilustración 38**, donde MOD_FREE es el módulo del vector del campo magnético tomado como referencia con la plaza libre, MOD_NEW es el vector leído en las siguientes iteraciones, también estando la plaza libre, y ANGLE es el ángulo que forman ambos vectores.

TIME	DEV_EUI	TAG	MOD_FREE	MOD_NEW	ANGLE
Tue Aug 16 2022 19:13:22	A8610A33344B7115	Dispositivo 1	55,33	55,33	0
Tue Aug 16 2022 19:14:22	A8610A33344B7115	Dispositivo 1	55,33	55,12	0,2
Tue Aug 16 2022 19:15:22	A8610A33344B7115	Dispositivo 1	55,33	55,54	0,37
Tue Aug 16 2022 19:16:22	A8610A33344B7115	Dispositivo 1	55,33	55,05	0,03
Tue Aug 16 2022 19:17:22	A8610A33344B7115	Dispositivo 1	55,33	55,41	0,16
Tue Aug 16 2022 19:19:23	A8610A33344B7115	Dispositivo 1	55,33	55,28	0,34
Tue Aug 16 2022 19:20:23	A8610A33344B7115	Dispositivo 1	55,33	55,31	0,13
Tue Aug 16 2022 19:22:23	A8610A33344B7115	Dispositivo 1	55,33	55,25	0,14
Tue Aug 16 2022 19:23:23	A8610A33344B7115	Dispositivo 1	55,33	55,41	0,22
Tue Aug 16 2022 19:24:23	A8610A33344B7115	Dispositivo 1	55,33	55,44	0,22
Tue Aug 16 2022 19:27:24	A8610A33344B7115	Dispositivo 1	55,33	56,21	20,2
Tue Aug 16 2022 19:29:24	A8610A33344B7115	Dispositivo 1	55,33	55,48	0,28
Tue Aug 16 2022 19:31:24	A8610A33344B7115	Dispositivo 1	55,33	55,44	0,64

Ilustración 38 - Variaciones del campo magnético estando la plaza libre

Observando esas fluctuaciones se determinó que, de forma temporal y a falta de posteriores pruebas, el umbral se establecería en 2.0 uT para la diferencia absoluta entre los módulos y en 3.0 ° para el ángulo entre los vectores.

■ Codificación y envío de mensaje uplink:

Respecto a la composición del mensaje que envía el dispositivo, inicialmente se incluían el módulo del vector del campo magnético tomado como referencia cuando la plaza estaba libre, el módulo del vector obtenido en la última lectura, el ángulo que formaban esos vectores y el estado de la plaza. Posteriormente, se modificaría el mensaje para que solo enviara el estado.

Con relación a la codificación del mensaje *uplink*, se ha utilizado Cayenne LPP. La clase definida por la librería CayenneLPP.h de myDevices (2018) facilita el envío de datos a través de redes LoRaWAN, ya que cumple con la restricción de tamaño de la carga útil reduciéndola a 11 bytes. Además, permite el envío de datos de múltiples sensores a la vez (The Things Network, 2020a; The Things Stack, 2021b).



Este proceso de codificación, realizado mediante la función `getCayenneFormat()`, solo se lleva a cabo cuando se vaya a enviar un mensaje. Originalmente, el dispositivo está programado para leer el estado de la plaza cada minuto y solo realizar un envío si ha cambiado ese valor (de libre a ocupado, y viceversa). El período de lectura es fijo, pero el de envío se puede modificar de forma remota, como se explica más adelante.

Respecto al proceso de envío del mensaje al servidor de red TTS, realizado mediante la función `sendMsg()`, el dispositivo debe estar previamente enlazado con una aplicación de TTN. Esto se realiza al iniciar o resetear el dispositivo, en la función `setup()` del programa, utilizando las claves proporcionadas por TTN (véase apartado **Conexión y uso de TTN**) que se encuentran en ARDUINO_SECRETS.h (**Anexo II – Manual de Código**).

En lo referente a la conexión con LoRaWAN y el envío/recepción de mensajes, se ha utilizado la librería `MKRWAN.h` de (Arduino, 2021b), creada principalmente para los MKR WAN 1300/1310. Esta librería proporciona APIs para la comunicación con redes LoRaWAN utilizando el protocolo LoRaWAN y radios LoRa.

■ **Recepción de mensaje *downlink*:**

Debido a la política de mensajes de LoRaWAN, solamente tras cada envío de mensaje *uplink* se abre una ventana de tiempo para recibir un mensaje *downlink*, el cual indicaría algún tipo de modificación del comportamiento del dispositivo.

Para ello, se ha establecido un protocolo de mensajes de bajada, enviados desde el servidor de red, que pueden ser de 4 tipos:

- **Tipo 1:** Indica al dispositivo que debe tomar los valores actuales como referencia. Este mensaje se envía al colocar el dispositivo en la plaza (libre) por primera vez.
- **Tipo 2:** Modifica los umbrales de comparación: la diferencia entre módulos y el ángulo entre los vectores.
- **Tipo 3:** Indica al dispositivo que añada el estado de la batería al próximo mensaje.



- **Tipo 4:** Modifica el período de envío, pudiendo ser cada x minutos (opción 1) o solamente al detectar un cambio de estado de la plaza (opción 2).

La estructura de estos mensajes sería: `tipo[-valor1[-valor2]]`

En la **Tabla 42** se muestran algunos ejemplos de mensajes *downlink* de distinto tipo:

Tabla 42 - Ejemplos de mensajes downlink

Mensaje	Descripción
1	Tomar valores de referencia
2-2.50-4.00	Modificar los umbrales con 2.5 uT y 4 °
3	Añadir estado de la batería
4-1-5	Enviar mensaje cada 5 minutos
4-2	Enviar mensaje solamente al detectar cambio de estado de la plaza

La recepción y procesamiento del mensaje recibido se realizan en la función `receiveMsg()`, mientras que la modificación del comportamiento según el tipo de mensaje se realiza en la función `getInfoMsg()`.

Así pues, el programa desarrollado sigue, de forma general, el diagrama de la **Ilustración 39**. En este no se plasman las posibles acciones desencadenadas por la recepción de mensajes *downlink*, sino que sigue el flujo general en el que no se recibirían.

El código desarrollado en la programación del dispositivo se puede consultar en el **Anexo II – Manual de Código**, en concreto en la **Aplicación del dispositivo inalámbrico**.

Con relación a la funcionalidad de lectura de iluminación, no se ha incluido porque se ha considerado que un envío constante de mensajes por parte del dispositivo, como ocurriría en ese caso, drenaría la batería en muy poco tiempo. Además, tomar datos de iluminación mediante sensores como los LDR, por ejemplo, no aportaría suficiente información como para saber si la luz recibida es directa o simplemente luz ambiente, haciendo imposible saber si la plaza se encuentra bajo luz solar directa o a la sombra. Por eso, se ha decidido dejar esta funcionalidad para **Futuras Mejoras**.

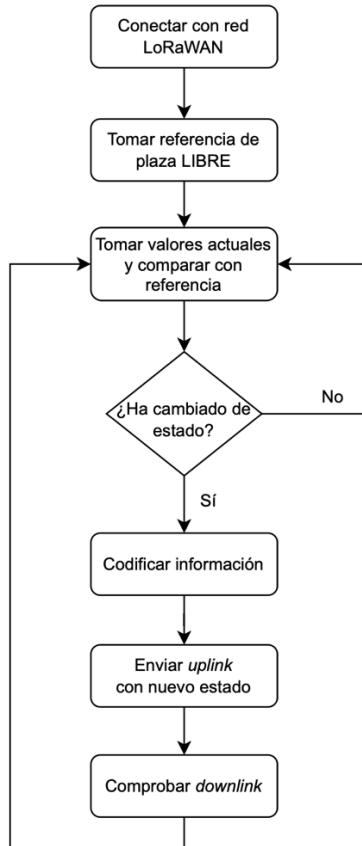


Ilustración 39 - Flujo general del programa

11.2. DESPLIEGUE EN ENTORNO

Para el despliegue de los dispositivos en zonas con cobertura LoRaWAN, se ha utilizado la infraestructura de red ofrecida por la comunidad de The Things Network.

11.2.1. THE THINGS NETWORK

Como ya se comentó en el análisis de **LoRa/LoRaWAN**, TTN es una plataforma colaborativa, libre y gratuita del Internet de las Cosas que permite crear redes, dispositivos y soluciones utilizando LoRaWAN.

Por otro lado, The Things Stack es un servidor de red LoRaWAN de nivel empresarial construido sobre un núcleo de código abierto⁷. TTS permite desplegar y administrar redes LoRaWAN en hardware propio o en la nube.

⁷ <https://github.com/TheThingsNetwork/lorawan-stack>



Para ello, ofrece diferentes implementaciones, entre las que se encuentra la *Community Edition*. Esta versión, dirigida por TTN, es su servidor de red de uso gratuito, el cual proporciona la red LoRaWAN basada en la comunidad más grande del mundo (The Things Stack, 2022b).

Es esta comunidad la que pone su infraestructura (*gateways*) a disposición de todos, permitiendo la conexión directa a la red sin necesidad de desplegar una propia. Aun así, si los dispositivos no se encontraran en las cercanías de un *gateway*, se podría instalar uno rápidamente para dar cobertura a la zona, ampliando así la infraestructura de esta red.

En el desarrollo de este proyecto no hizo falta instalar ninguna puerta de enlace ya que el entorno donde se realizó el despliegue recibía cobertura de varios *gateways*, como se puede ver en la **Ilustración 40**.

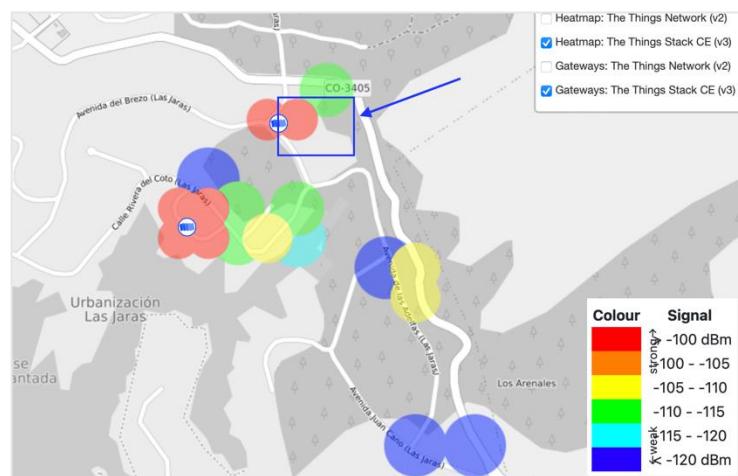


Ilustración 40 - Cobertura LoRaWAN de TTN en Urbanización Las Jaras (Córdoba)⁸

En la provincia de Córdoba hay ya una comunidad de 42 contribuidores que aporta 11 *gateways* públicos⁹, muchos de ellos instalados por el Aula de Transformación Digital FiWare de la Universidad de Córdoba en zonas como el Campus de Rabanales (**Ilustración 41**).

⁸ <https://ttnmapper.org/heatmap/>

⁹ <https://www.thethingsnetwork.org/community/cordoba-es/>

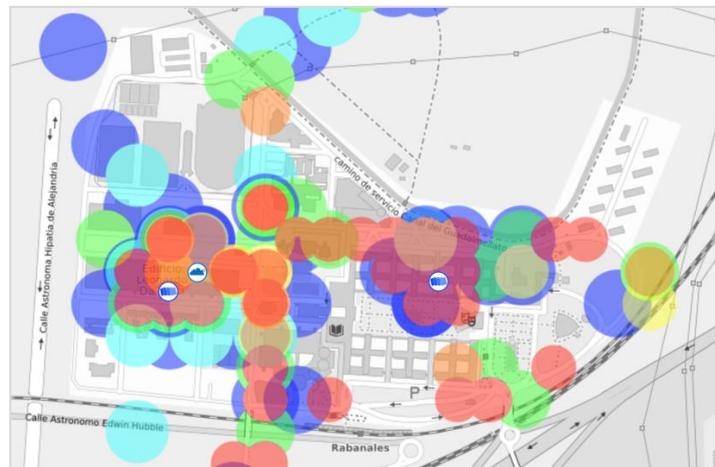


Ilustración 41 - Cobertura LoRaWAN de TTN en el Campus de Rabanales

11.2.2. CONEXIÓN Y USO DE TTN

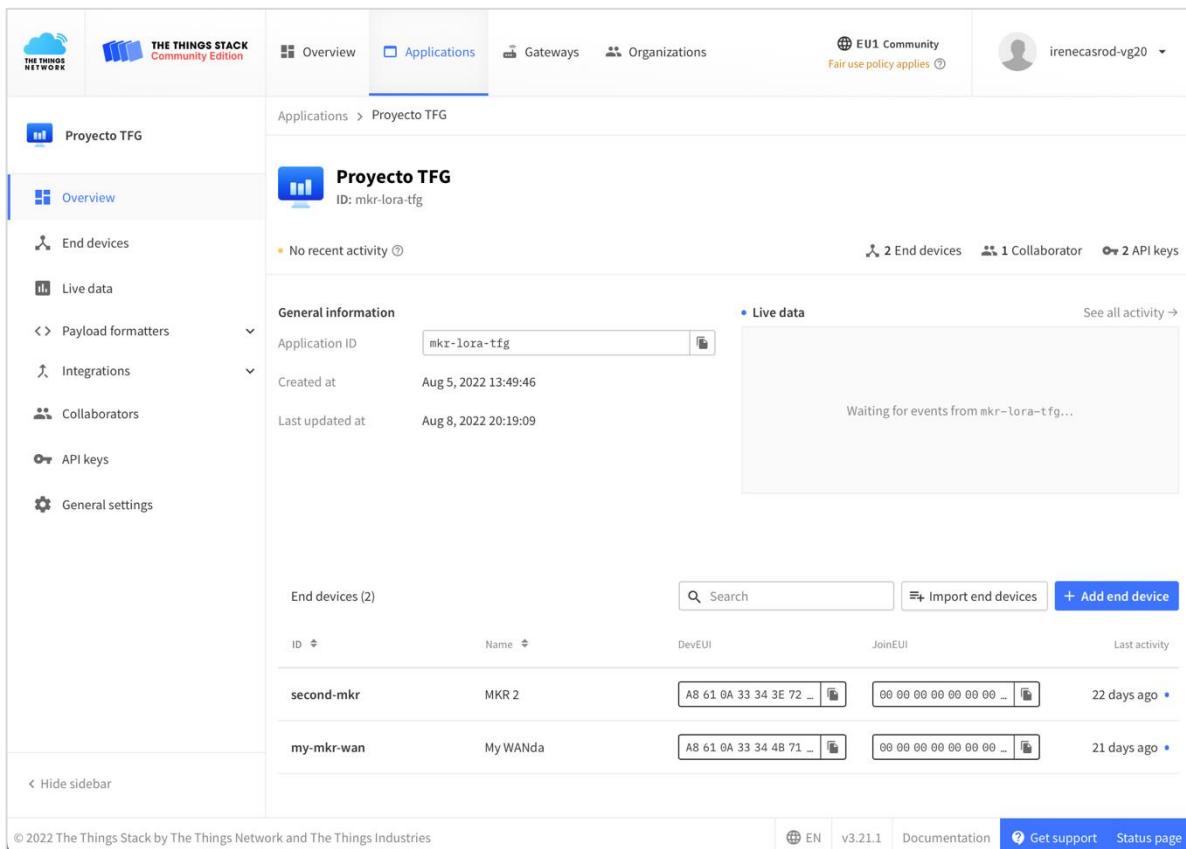
Tal y como se ha comentado en la descripción del **Software** del dispositivo, para que el nodo pueda enviar mensajes por la red LoRaWAN debe estar registrado en la plataforma TTN.

En TTN los dispositivos se administran en aplicaciones (**Ilustración 42**), las cuales pueden contener un número ilimitado de dispositivos, aunque suelen usarse para dividir los dispositivos por función o área geográfica (The Things Stack, 2022a).

Para añadir un dispositivo a la aplicación (véase **Integrar dispositivo en el sistema**), son necesarios unos valores que proporcionan seguridad al enlace:

- **DevEUI:** el identificador del dispositivo, aportado por el fabricante.
- **JoinEUI (AppEUI):** identifica al servidor en la activación y unión a la red.
- **AppKey:** clave de encriptación utilizada en el envío/recepción de mensajes.

Por otro lado, al configurar un dispositivo en TTN, el administrador indica las coordenadas de su ubicación, que será enviada a la plataforma de monitorización y a la aplicación móvil para la consulta de los usuarios.

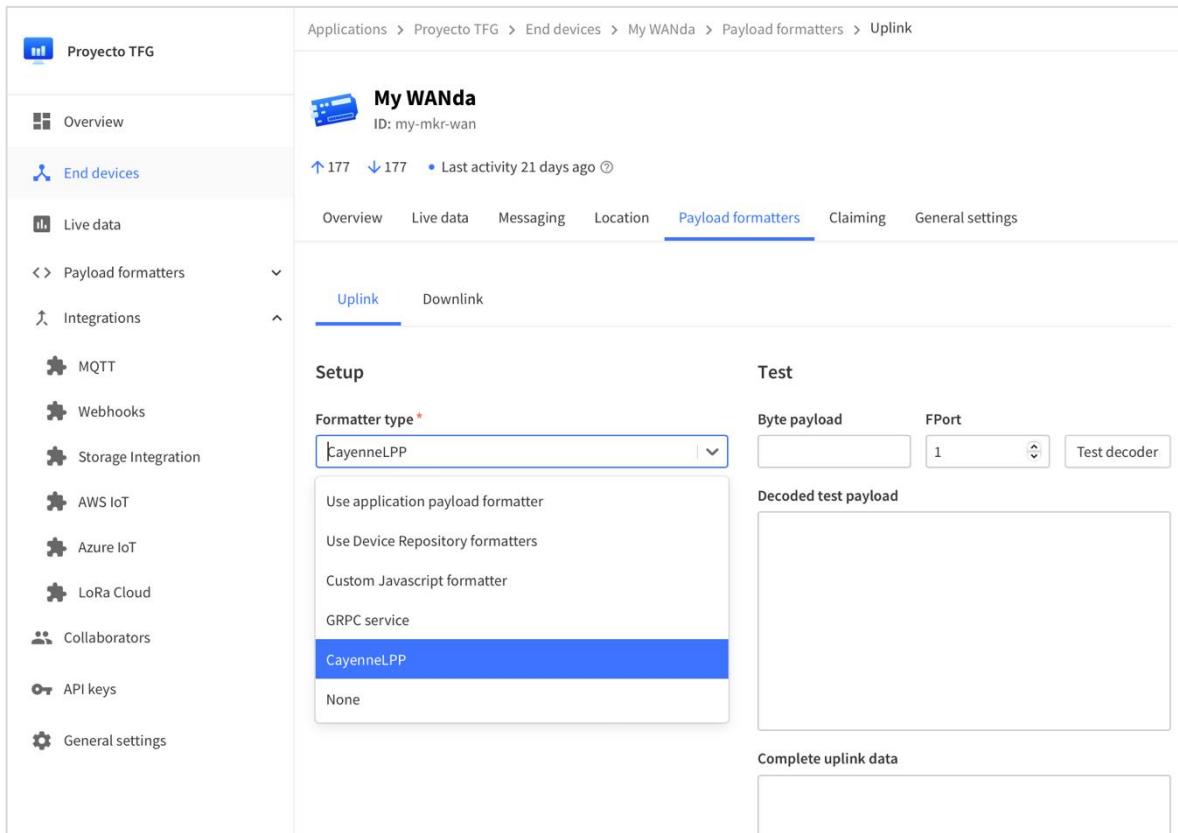



The screenshot shows the 'Applications' section of The Things Stack Community Edition. The application 'Proyecto TFG' is selected. The sidebar on the left includes options like Overview, End devices, Live data, Payload formatters, Integrations, Collaborators, API keys, and General settings. The main content area shows 'General information' with fields for Application ID (mkr-lora-tfg), Created at (Aug 5, 2022 13:49:46), and Last updated at (Aug 8, 2022 20:19:09). It also shows 'Live data' with a message 'Waiting for events from mkr-lora-tfg...'. Below this, there's a table for 'End devices (2)' with columns for ID, Name, DevEUI, JoinEUI, and Last activity. Two devices are listed: 'second-mkr' (MKR 2) and 'my-mkr-wan' (My WANDA). At the bottom, there are links for EN, v3.21.1, Documentation, Get support, and Status page.

Ilustración 42 - Aplicación en The Things Network

Desde la propia consola de TTN, el administrador puede enviar mensajes *downlink* a los dispositivos en formato JSON o directamente como una cadena de bytes en HEX. Debido al protocolo de mensajes *downlink* establecido para el **Prototipo de dispositivo**, se deberá enviar como una cadena convertida a HEX (véase **Enviar mensaje *downlink* a dispositivo**).

Por otro lado, en la ventana de *Payload formatters* se podría introducir una función en JavaScript que decodificara y mostrara en la consola el mensaje enviado por un dispositivo. Sin embargo, la ventaja de haber usado CayenneLPP para la codificación del mensaje *uplink* es que TTN lo incluye entre las opciones de formato (**Ilustración 43**). Por eso, no ha sido necesario crear una función que procesara el mensaje, sino que bastaba con seleccionar CayenneLPP entre las opciones de tipo de formato de *uplink*.



The screenshot shows the TTN web interface for managing end devices. On the left, there's a sidebar with various integration options like MQTT, Webhooks, Storage Integration, AWS IoT, Azure IoT, LoRa Cloud, Collaborators, API keys, and General settings. The main panel displays the details for an end device named "My WANda" (ID: my-mkr-wan). It shows activity statistics (177 uplinks, 177 downlinks, last activity 21 days ago) and tabs for Overview, Live data, Messaging, Location, Payload formatters (which is currently selected), Claiming, and General settings. Below these tabs, there are two tabs: Uplink and Downlink. The Uplink tab is active. In the "Payload formatters" section, there's a "Setup" area where "Formatter type*" is set to "CayenneLPP". To the right, there's a "Test" section with fields for "Byte payload" (set to 1) and a "Decoded test payload" area. At the bottom, there's a "Complete uplink data" section.

Ilustración 43 - Formato CayenneLPP en la consola de TTN

11.3. PLATAFORMA DE MONITORIZACIÓN

TTN ofrece muchas integraciones que ayudan al procesamiento de los datos y a actuar sobre ellos mediante la activación (*trigger*) de eventos. De esta manera, se puede implementar una solución para monitorear los datos en alguna plataforma IoT en la nube o incluso programar *downlinks* que se enviarán al dispositivo final (The Things Network, 2022).

Unas de las integraciones ofrecidas son los *webhooks*, que permiten enviar peticiones HTTP con mensajes relacionados con la aplicación a determinados puntos finales HTTP.

Estos *webhooks* se pueden configurar para enviar la petición HTTP en diferentes casos, como la llegada de un mensaje *uplink*, el envío de un mensaje *downlink* o incluso cuando otra integración emita un evento.

A continuación, se presenta la plataforma de monitorización, integrada mediante un *webhook*, que utilizaría el administrador del sistema.



11.3.1. CAYENNE MYDEVICES

Con el objetivo de monitorizar los datos de los dispositivos se ha utilizado Cayenne, una plataforma IoT gratuita desarrollada por myDevices. Presenta una solución visual de arrastrar y soltar que fue diseñada para ayudar a los usuarios a crear prototipos de IoT y, más tarde, llevarlos a producción (myDevices, 2022).

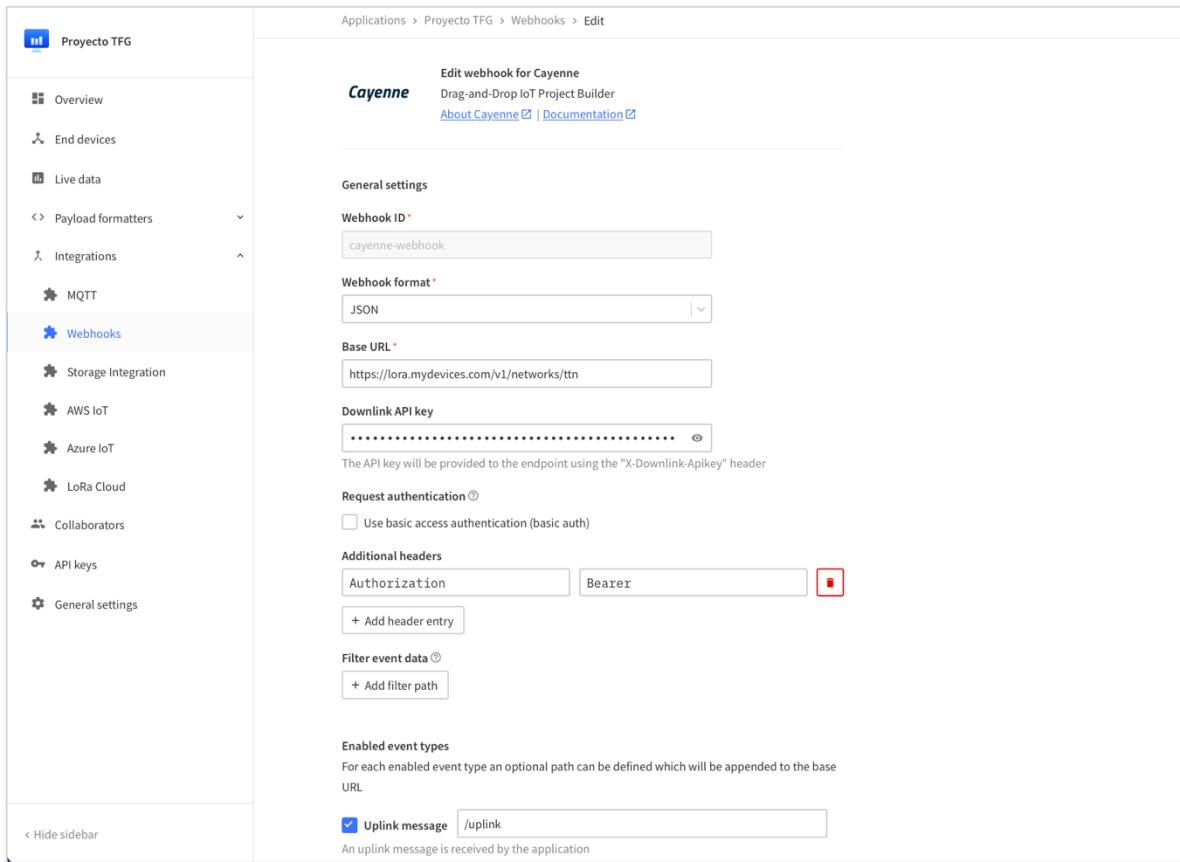
El motivo principal para escoger esta plataforma fue el uso de la codificación CayenneLPP, también desarrollada por myDevices, permitiendo una decodificación instantánea de la información recibida.

La integración de Cayenne myDevices con TTN sigue un procedimiento bastante sencillo, como demuestra The Things Stack (2021a).

En primer lugar, se ha creado una integración de *webhook* en TTN (**Ilustración 44**) a través de la cual se reenvía la información hasta Cayenne. Para hacer más sencillas estas integraciones, existen varias plantillas creadas por la comunidad, incluida una para Cayenne en la que basta con indicar un ID de *webhook* para crearla.

A través de ese *webhook*, TTN enviará una petición HTTP POST al recibir un mensaje de algún dispositivo, añadiendo a esa información algunos metadatos como el devEUI, el RSSI, la ubicación o la hora de recepción.

En segundo lugar, se ha configurado un dispositivo en Cayenne, donde se pueden añadir diferentes microcontroladores, sensores, ordenadores SBC, etc. (véase **Añadir dispositivo a plataforma de monitorización**).



The screenshot shows the Cayenne IoT Project Builder interface. On the left, a sidebar menu includes options like Overview, End devices, Live data, Payload formatters, Integrations (MQTT, Webhooks selected), Storage Integration, AWS IoT, Azure IoT, LoRa Cloud, Collaborators, API keys, and General settings. The main panel is titled "Edit webhook for Cayenne" and shows the "Cayenne" section. It contains fields for Webhook ID (cayenne-webhook), Webhook format (JSON), Base URL (https://lora.mydevices.com/v1/networks/ttn), Downlink API key (redacted), Request authentication (checkbox for basic access auth), Additional headers (Authorization set to Bearer), Filter event data (checkbox for uplink message, path /uplink), and Enabled event types (checkbox for Uplink message). A note states: "For each enabled event type an optional path can be defined which will be appended to the base URL".

Ilustración 44 - Integración de Cayenne con TTN

El componente principal de esta plataforma es su *dashboard*, que permite visualizar datos, configurar reglas, programar eventos, etc. En el momento en que se envíe un mensaje *uplink* con información en formato CayenneLPP, los *widgets* del *dashboard* de Cayenne se encargarán de representarlo (**Ilustración 45**).

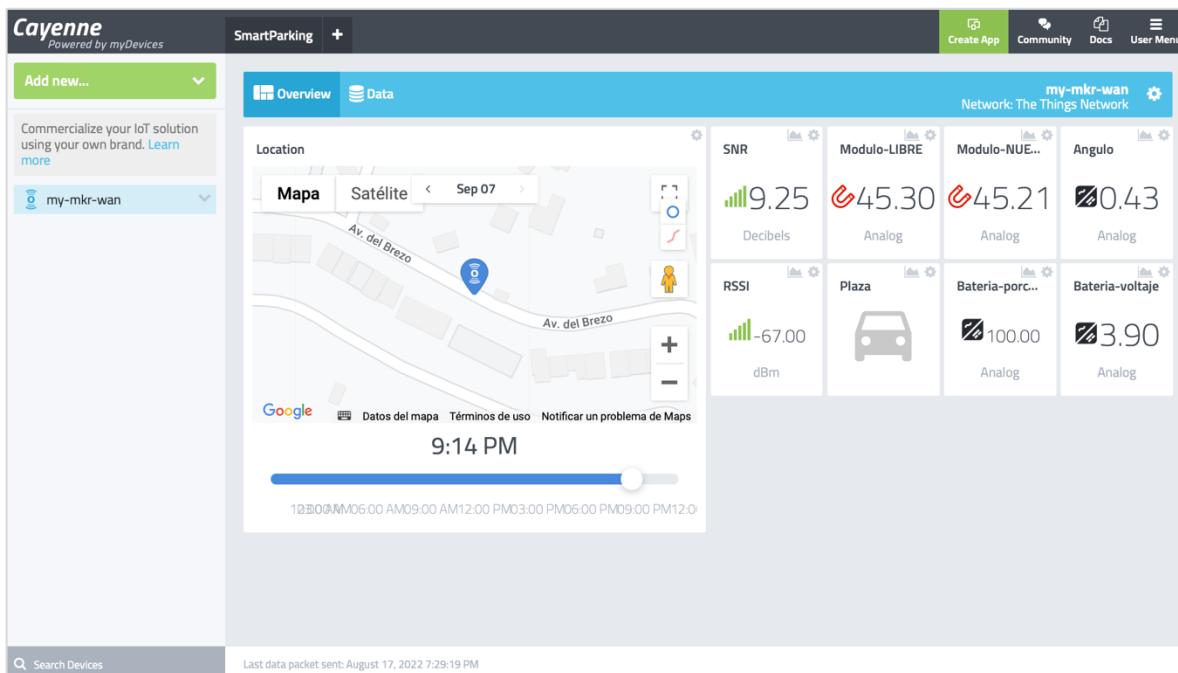


Ilustración 45 - Representación en Cayenne de los datos enviados por un dispositivo

Por otro lado, en el *dashboard* general del proyecto, que se ha llamado *SmartParking*, se puede ver información de todos los dispositivos que se hayan conectado (**Ilustración 46**).

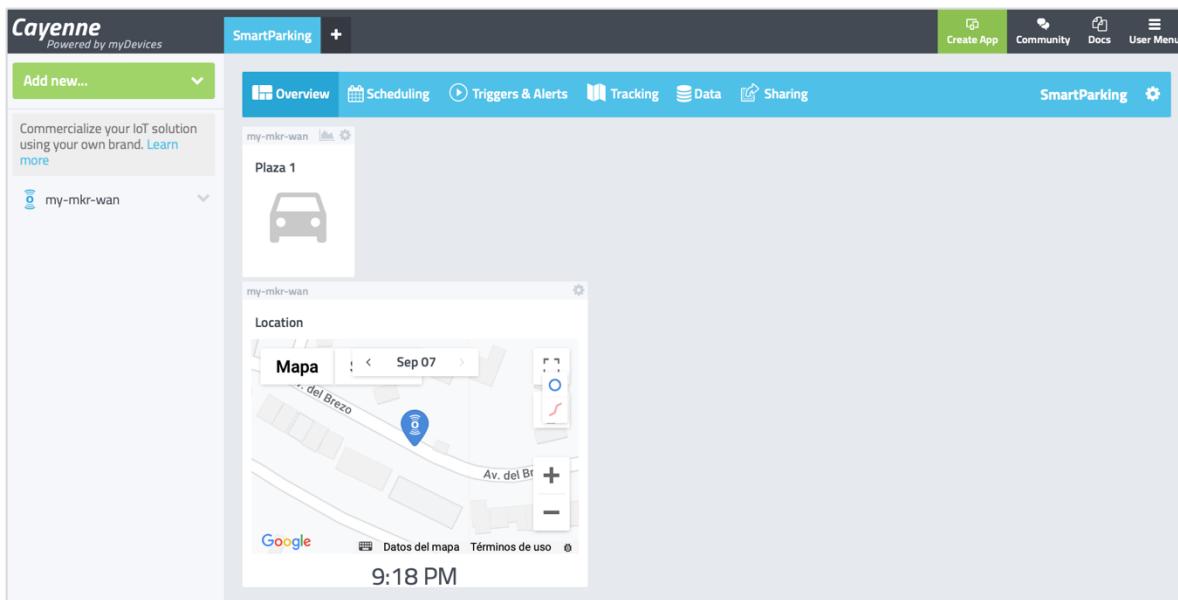


Ilustración 46 - Dashboard general de Cayenne



Otra utilidad que presenta Cayenne es el mantenimiento de un histórico de datos en su sistema (**Ilustración 47**), así como la posibilidad de realizar diferentes análisis y representaciones con ellos (**Ilustración 48**).

The screenshot shows the Cayenne Data page for a project titled "SmartParking". The main interface includes tabs for Overview, Scheduling, Triggers & Alerts, Tracking, Data, and Sharing. The "Data" tab is active, displaying a table of historical sensor data. The table has columns for Timestamp, Device, Channel, Sensor Name, Sensor ID, Data Type, Unit, and Values. The data shows multiple entries for "Plaza 1" over a period from August 16, 2022, at 9:33:40 to 9:45:11. The values are mostly 1, with some spikes at specific times.

Timestamp	Device	Channel	Sensor Name	Sensor ID	Data Type	Unit	Values
2022-08-16 9:33:40	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	1
2022-08-16 9:32:40	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	1
2022-08-16 9:31:40	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	1
2022-08-16 9:30:40	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	
2022-08-16 9:29:39	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	
2022-08-16 9:28:41	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	
2022-08-16 9:27:52	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	1
2022-08-16 9:26:52	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	1
2022-08-16 9:25:52	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	1
2022-08-16 9:24:52	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	1
2022-08-16 9:23:52	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	1
2022-08-16 9:22:51	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	1
2022-08-16 9:21:51	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	1
2022-08-16 9:20:51	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	
2022-08-16 9:19:51	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	1
2022-08-16 9:18:51	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	1
2022-08-16 9:17:51	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	1
2022-08-16 9:16:51	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	1
2022-08-16 9:15:51	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	1
2022-08-16 9:14:51	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	1

Ilustración 47 - Histórico de datos en Cayenne



Ilustración 48 - Representación gráfica de datos en Cayenne



Finalmente, Cayenne da la opción de compartir su tablero mediante un enlace¹⁰, usando la opción *Sharing* en la **Ilustración 46**. De esta forma, cualquier persona puede visualizar la información fácilmente desde cualquier lugar (**Ilustración 49**).

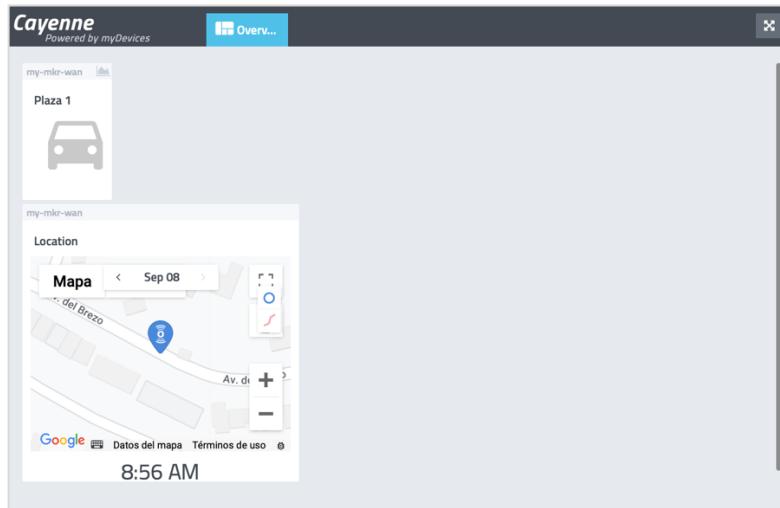


Ilustración 49 - Dashboard de Cayenne compartido

En lo que respecta a la visualización desde un dispositivo móvil (**Ilustración 50**), esta opción no consigue la mejor adaptabilidad. Por eso, se han buscado otras alternativas que acerquen la información al público general de una forma más atractiva, como la **Aplicación móvil** desarrollada.

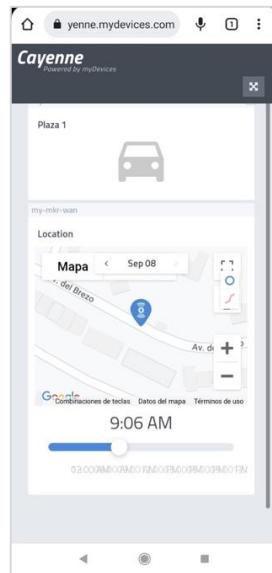


Ilustración 50 - Dashboard de Cayenne compartido. Visualización desde dispositivo móvil.

¹⁰ <https://cayenne.mydevices.com/shared/62f429424d9be65e639756e9>



11.4. APPLICACIÓN MÓVIL

Para el desarrollo de la aplicación móvil se han utilizado varios servicios de Google presentados en los **Recursos Software**: Apps Script, Google Sheets y AppSheet.

Cada mensaje enviado por un dispositivo, una vez alcance el servidor de red, seguirá el flujo de la **Ilustración 51**. TTN enviará ese mensaje a la aplicación web implementada en Apps Script, donde se procesarán los datos deseados y se actualizarán en una hoja de cálculo de Google Sheets. Finalmente, AppSheet obtendrá de esta hoja de cálculo la información lista para representarse en la aplicación móvil.



Ilustración 51 - Flujo de información de TTN a la aplicación móvil

La implementación de este proceso se ha dividido en tres partes:

- Integración de Apps Script en TTN.
- Procesamiento de información en Apps Script y actualización en Google Sheets.
- Representación de la información en la aplicación creada en AppSheet.

Al igual que la plataforma de monitorización, se ha utilizado una integración de *webhook* para conectar TTN a Apps Script. En este caso no se ha utilizado una plantilla, sino que se ha configurado un *webhook* general (**Ilustración 52**) donde se indica la URL de la aplicación web implementada en Apps Script. Este *webhook* se activa para los mismos eventos que la plataforma de monitorización: un mensaje *uplink* enviado desde algún dispositivo configurado en la aplicación de TTN.

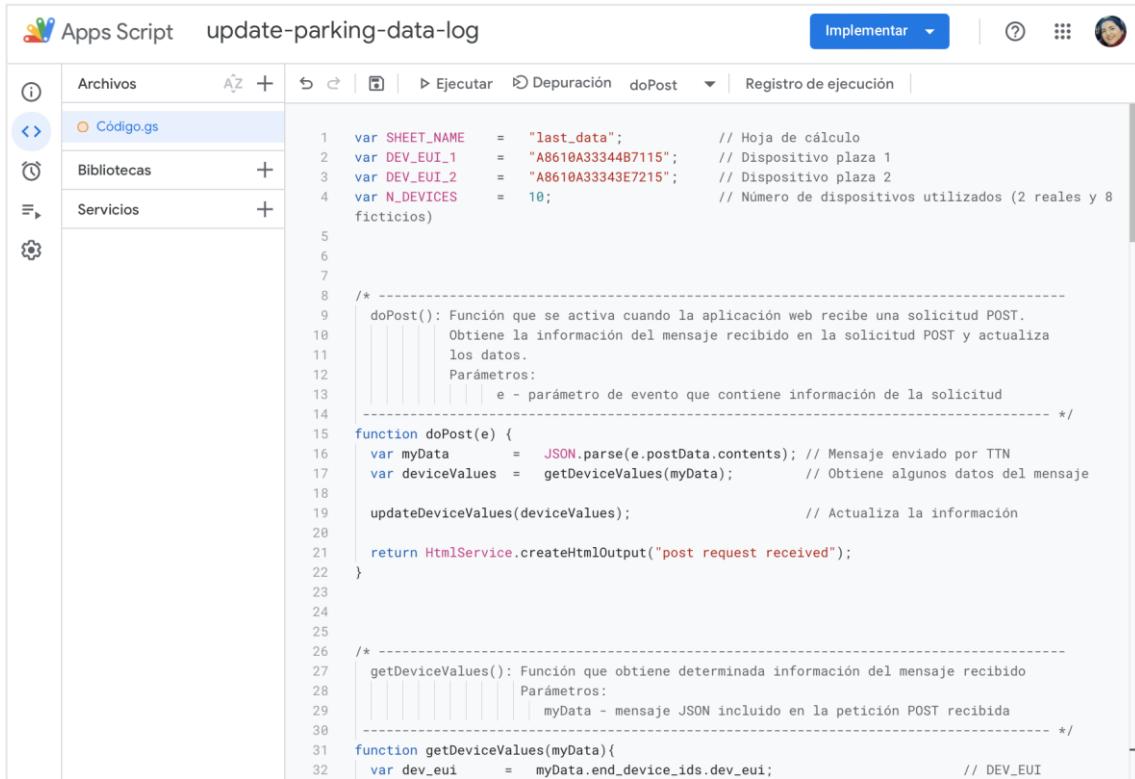


The screenshot shows the 'Edit webhook' configuration page in The Things Stack Community Edition. The left sidebar is collapsed, showing options like Overview, End devices, Live data, Payload formatters, Integrations, MQTT, Webhooks (selected), Storage Integration, AWS IoT, Azure IoT, LoRa Cloud, Collaborators, API keys, and General settings. The main panel has a breadcrumb path: Applications > Proyecto TFG > Webhooks > Edit. The 'Edit webhook' section contains fields for Webhook ID (parking-data-log), Webhook format (JSON), Base URL (https://script.google.com/macros/s/AKfycbwPibDGQdC-ZjIRScFgVw), Downlink API key (empty), Request authentication (checkbox for basic auth), Additional headers (+ Add header entry), Filter event data (+ Add filter path), and Enabled event types (checkbox for Uplink message selected, path /path/to/webhook). A note below says 'An uplink message is received by the application'.

Ilustración 52 - Integración de Apps Script con TTN

Como se ha comentado, cuando TTN recibe un mensaje *uplink* de un dispositivo, lo reenvía en una petición HTTP POST hasta Apps Script (**Ilustración 53**), donde se activa la función `doPost()` para procesar el mensaje (véase **Aplicación web de Apps Script**). Se han utilizado 2 dispositivos reales y 8 ficticios para hacer una prueba más real, ubicándolos esta vez en el Campus de Rabanales.

Con cada petición POST recibida, se obtienen determinados datos del mensaje (identificador del dispositivo, estado de la plaza, ubicación...) mediante la función `getDeviceValues()` y se actualiza la hoja de cálculo en `updateDeviceValues()`.

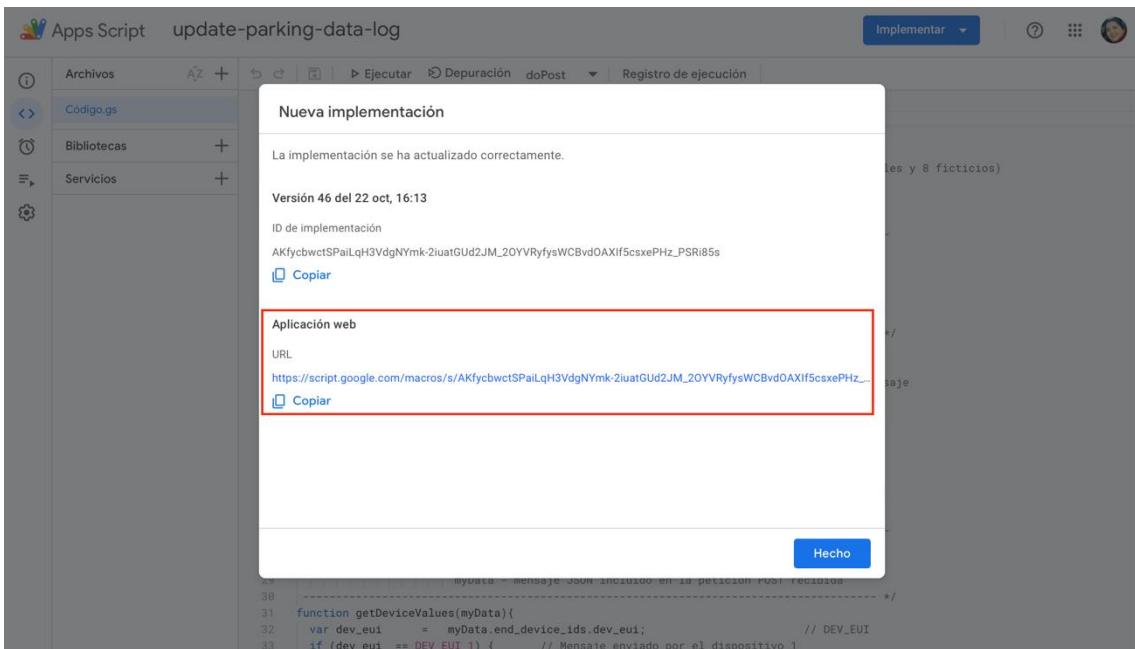


The screenshot shows the Google Apps Script interface with the script file 'update-parking-data-log'. The code implements a POST handler to update device values from a JSON message. It includes functions for handling the POST request and extracting device values from the JSON payload.

```
1 var SHEET_NAME = "last_data"; // Hoja de cálculo
2 var DEV_EUI_1 = "A8610A3334B7115"; // Dispositivo plaza 1
3 var DEV_EUI_2 = "A8610A33343E7215"; // Dispositivo plaza 2
4 var N_DEVICES = 10; // Número de dispositivos utilizados (2 reales y 8 ficticios)
5
6
7
8 /* -----
9  * doPost(): Función que se activa cuando la aplicación web recibe una solicitud POST.
10 * Obtiene la información del mensaje recibido en la solicitud POST y actualiza
11 * los datos.
12 * Parámetros:
13 *   e - parámetro de evento que contiene información de la solicitud
14 * -----
15 function doPost(e) {
16   var myData = JSON.parse(e.postData.contents); // Mensaje enviado por TTN
17   var deviceValues = getDeviceValues(myData); // Obtiene algunos datos del mensaje
18
19   updateDeviceValues(deviceValues); // Actualiza la información
20
21   return HtmlService.createHtmlOutput("post request received");
22 }
23
24
25
26 /* -----
27 * getDeviceValues(): Función que obtiene determinada información del mensaje recibido
28 * Parámetros:
29 *   myData - mensaje JSON incluido en la petición POST recibida
30 * -----
31 function getDeviceValues(myData){
32   var dev_eui = myData.end_device_ids.dev_eui; // DEV_EUI
```

Ilustración 53 – Aplicación web de Apps Script

El script desarrollado se debe publicar como una aplicación web (Google Apps Script, 2022b). La URL de esa implementación, mostrada en la Ilustración 54, es la que se introduce en la configuración del webhook de la Ilustración 52.



The screenshot shows the 'Nueva implementación' (New deployment) dialog box. It displays the deployment history, the deployment ID, and the generated application web URL. The URL is highlighted with a red box.

Nueva implementación
La implementación se ha actualizado correctamente.
Versión 46 del 22 oct, 16:13
ID de implementación
AKfycbwctSPaiLqH3VdgNYmk-2iuatGUd2JM_20YVRyfysWCbvdoAXIf5csxePHz_PSI85s
Copiar
Aplicación web
URL
https://script.google.com/macros/s/AKfycbwctSPaiLqH3VdgNYmk-2iuatGUd2JM_20YVRyfysWCbvdoAXIf5csxePHz_PSI85s/exec
Copiar
Hecho

Ilustración 54 - Implementación de aplicación web



Por otro lado, se ha incluido una nueva funcionalidad que permite ofrecer información sobre el tiempo meteorológico en la ubicación de cada dispositivo. Para ello, se ha utilizado la API de OpenWeatherMap (2022) en la función `getWeather()`. Como esta información debe ser actualizada más a menudo de lo que lo harán los dispositivos, se ha establecido un *trigger* en Apps Script (**Ilustración 55**) para que se actualice cada 15 minutos mediante la función `updateExtraInfoAllDevices()`.

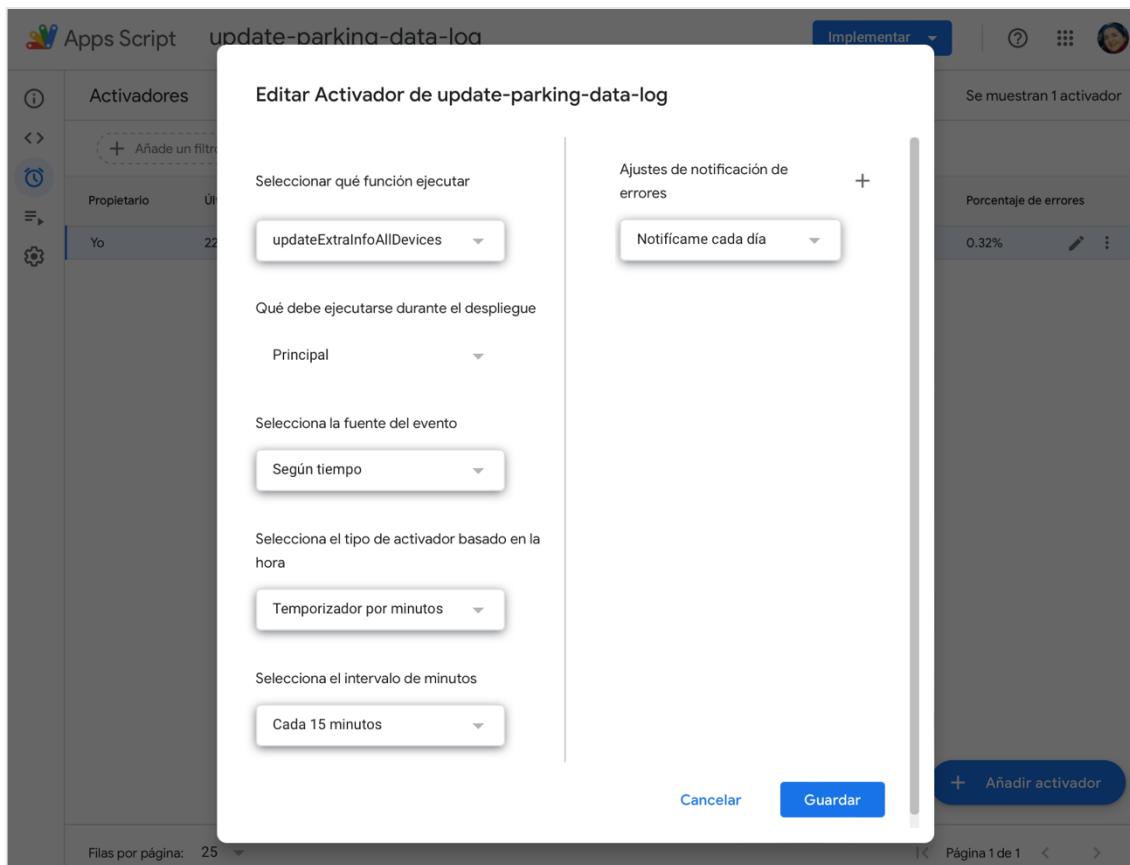


Ilustración 55 - Trigger para actualizar tiempo meteorológico en Apps Script

En la **Ilustración 56** se puede ver la hoja de cálculo donde se irá actualizando la información procesada.



The screenshot shows a Google Sheets document titled "parking_data_log". The data is organized into columns A through I, with headers: time, dev_eui, plaza, mod_free, mod_new, angle, status, address, and Temperat. The data consists of 11 rows of sensor readings for 10 different parking spots (Plaza 1 to Plaza 10). The "address" column contains coordinates, and the "Temperat" column shows temperatures in degrees Celsius. The interface includes standard spreadsheet tools like filters, sort, and search.

time	dev_eui	plaza	mod_free	mod_new	angle	status	address	Temperat
2 Sat Oct 22 2022 15:58:31 GMT+020 A8610A33344B7115		Plaza 1	45,3	45,21	0,43		0 37.914765, -4.724311	24.71°C
3 Sat Oct 22 2022 15:58:31 GMT+020 A8610A33343E7215		Plaza 2	16,94	17,08	0,38		0 37.914823, -4.725539	24.71°C
4 Sat Oct 22 2022 15:58:32 GMT+020 A8610A33344B7111		Plaza 3	17,2	57,53	3,12		1 37.914705, -4.723086	24.71°C
5 Sat Oct 22 2022 15:58:32 GMT+020 A8610A33344B7112		Plaza 4	27,5	29,19	2,11		0 37.914337, -4.722720	24.72°C
6 Sat Oct 22 2022 15:58:32 GMT+020 A8610A33344B7113		Plaza 5	23,27	48,13	6,5		1 37.913834, -4.722756	24.71°C
7 Sat Oct 22 2022 15:58:33 GMT+020 A8610A33344B7114		Plaza 6	44,5	47,54	4,12		0 37.915811, -4.721464	24.68°C
8 Sat Oct 22 2022 15:58:33 GMT+020 A8610A33344B7116		Plaza 7	27,2	69,06	3,5		1 37.915698, -4.719562	24.68°C
9 Sat Oct 22 2022 15:58:33 GMT+020 A8610A33344B7117		Plaza 8	37,45	39,53	2,1		0 37.916005, -4.720535	24.68°C
10 Sat Oct 22 2022 15:58:33 GMT+020 A8610A33344B7118		Plaza 9	23,15	47,38	5,67		1 37.916053, -4.721401	24.68°C
11 Sat Oct 22 2022 15:58:33 GMT+020 A8610A33344B7119		Plaza 10	38,09	68,18	2,87		1 37.913326, -4.719191	24.71°C
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								

Ilustración 56 - Hoja de cálculo con información a representar en la app

Por último, se ha creado la aplicación móvil **Smart Parking** mediante la plataforma AppSheet, que permite obtener datos desde diferentes fuentes, como hojas de cálculo o bases de datos. En este caso, se ha utilizado la hoja de cálculo de la **Ilustración 56** como fuente de información para la aplicación (**Ilustración 57**).

Se ha optado por utilizar Google Sheets en lugar de implementar una base de datos por los siguientes motivos:

- La información también se envía a la plataforma de Cayenne, por lo que se utiliza su propia base de datos.
- La información utilizada en la aplicación se puede representar en una sola tabla, con tantas filas como dispositivos se incluyan en el sistema, como se puede ver en la **Ilustración 56**.

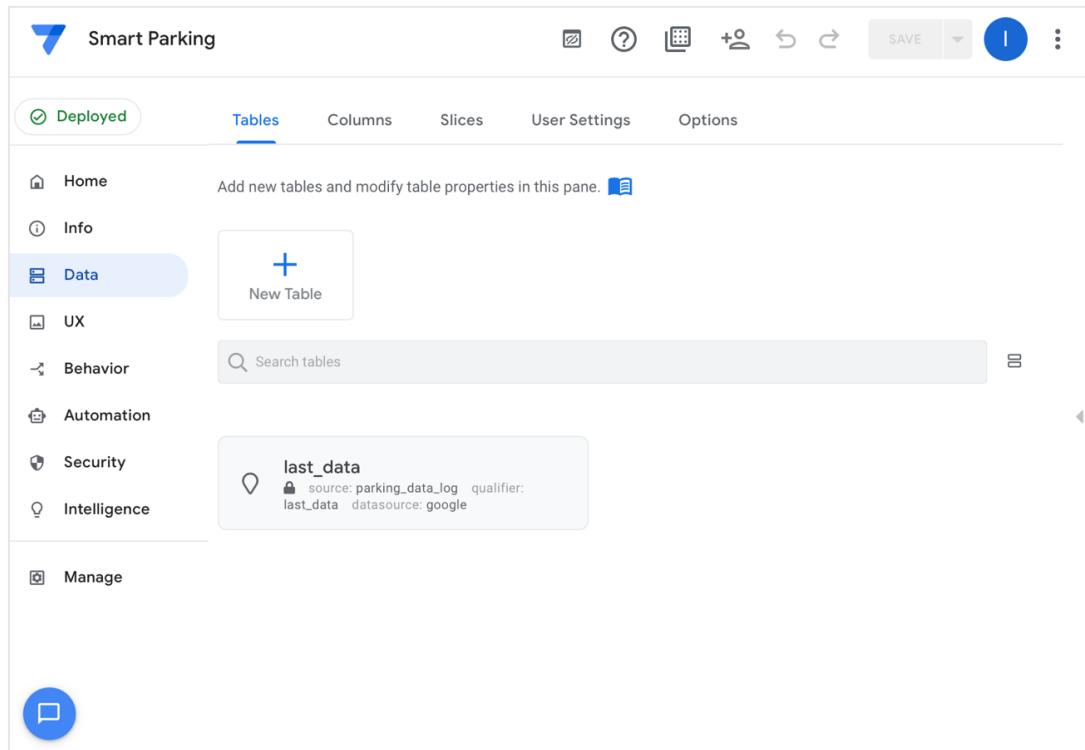
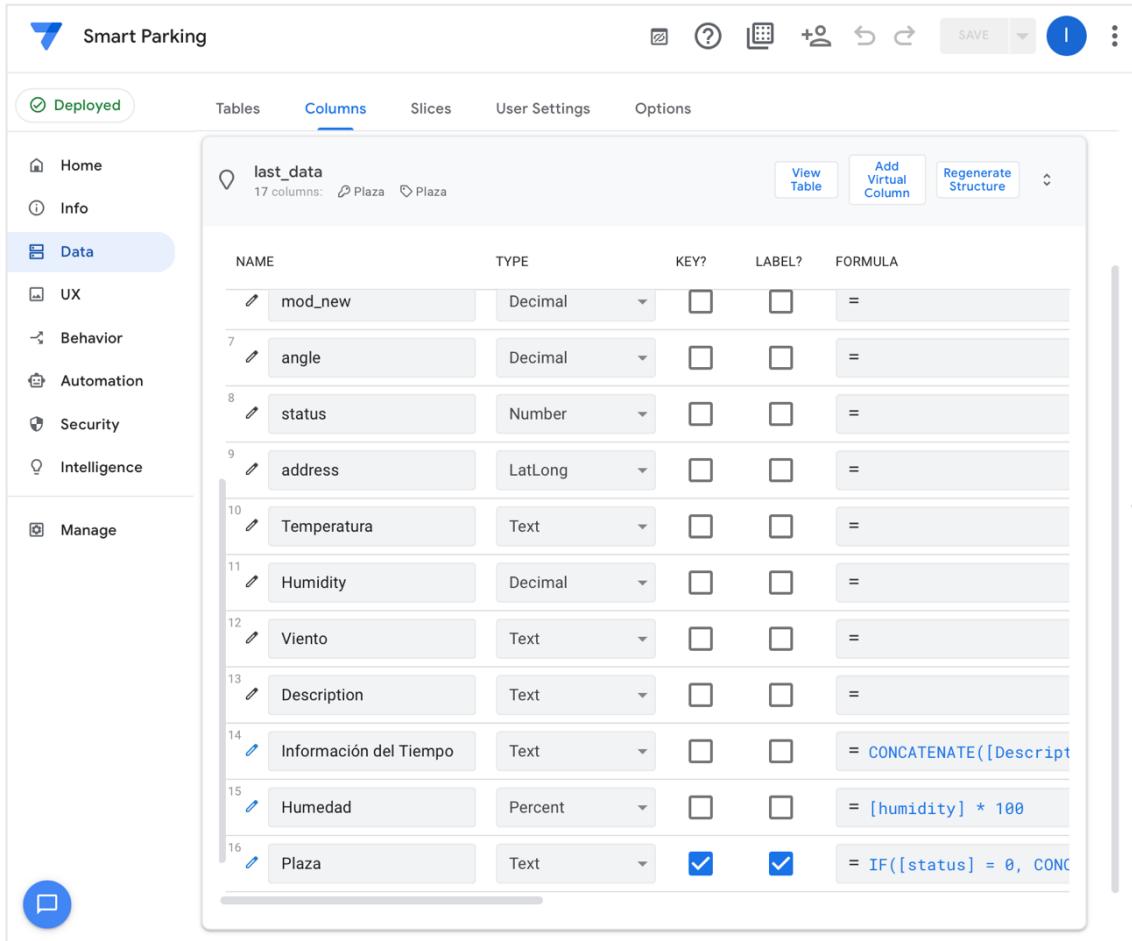


Ilustración 57 - Fuente de información de app Smart Parking (AppSheet)

A partir de la información obtenida se forma una tabla interna de la aplicación (**Ilustración 58**) donde se pueden añadir columnas “virtuales” cuyo valor dependa de los valores de otras columnas. Por ejemplo, la columna *Plaza* contiene el valor ‘LIBRE’ si *status* es ‘0’ y ‘OCUPADO’ si *status* es ‘1’. Así mismo, la columna *Información del Tiempo* concatena las columnas *Description*, *Temperatura*, *Humedad* y *Viento* para representar toda la información meteorológica de forma conjunta.

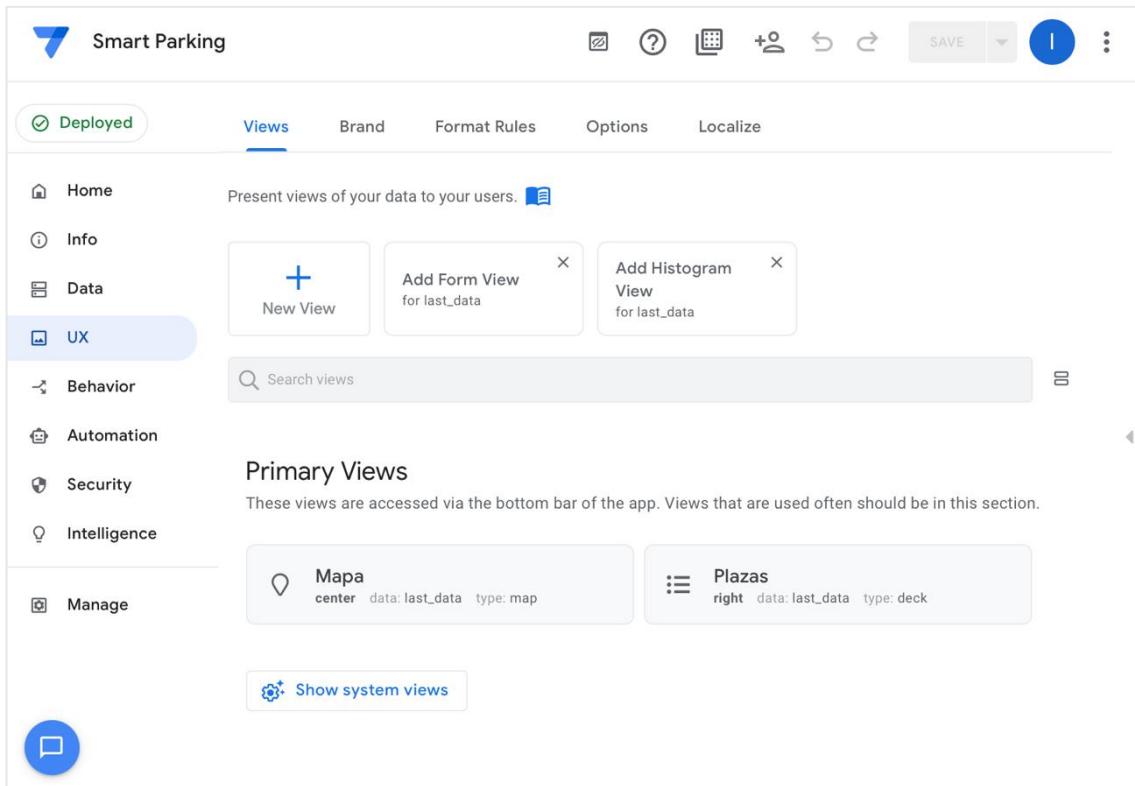


The screenshot shows the 'Smart Parking' application interface. On the left, there's a sidebar with navigation links: Home, Info, Data (which is selected), UX, Behavior, Automation, Security, Intelligence, and Manage. Below the sidebar is a blue circular button with a white square icon. The main area displays a table named 'last_data' with 17 columns. The table structure is as follows:

	NAME	TYPE	KEY?	LABEL?	FORMULA
1	mod_new	Decimal	<input type="checkbox"/>	<input type="checkbox"/>	=
2	angle	Decimal	<input type="checkbox"/>	<input type="checkbox"/>	=
3	status	Number	<input type="checkbox"/>	<input type="checkbox"/>	=
4	address	LatLong	<input type="checkbox"/>	<input type="checkbox"/>	=
5	Temperatura	Text	<input type="checkbox"/>	<input type="checkbox"/>	=
6	Humidity	Decimal	<input type="checkbox"/>	<input type="checkbox"/>	=
7	Viento	Text	<input type="checkbox"/>	<input type="checkbox"/>	=
8	Description	Text	<input type="checkbox"/>	<input type="checkbox"/>	=
9	Información del Tiempo	Text	<input type="checkbox"/>	<input type="checkbox"/>	= CONCATENATE([Descript]
10	Humedad	Percent	<input type="checkbox"/>	<input type="checkbox"/>	= [humidity] * 100
11	Plaza	Text	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	= IF([status] = 0, CONC

Ilustración 58 - Información de la app con columnas virtuales

Siguiendo el diseño de la **Interfaz de la aplicación móvil**, se han implementado las dos vistas principales que se muestran en la **Ilustración 59**.



The screenshot shows the configuration interface for a mobile application named "Smart Parking". The top navigation bar includes icons for settings, help, and saving. Below the title, there are tabs for "Deployed", "Views" (which is selected), "Brand", "Format Rules", "Options", and "Localize". On the left, a sidebar lists categories: Home, Info, Data, UX (selected), Behavior, Automation, Security, Intelligence, and Manage. A search bar at the bottom of the sidebar is labeled "Search views". The main area is titled "Primary Views" and contains two cards: "Mapa" (center, data: last_data, type: map) and "Plazas" (right, data: last_data, type: deck). There is also a button labeled "Show system views".

Ilustración 59 - Configuración vistas aplicación móvil

La primera vista configurada es la del Mapa (**Ilustración 60**), donde se muestran las ubicaciones de los dispositivos a partir de las coordenadas guardadas en la columna *address*. Además, se han establecido unas reglas de formato para que el pin de la ubicación aparezca de un color u otro (verde o rojo) según el estado de la plaza (**Ilustración 61**).

Por otro lado, se ha configurado la vista de Plazas (**Ilustración 62 e Ilustración 63**) para que muestre un listado de las plazas del sistema, indicando información de su estado, el tiempo meteorológico y la última actualización. Para esta vista también se han establecido reglas de formato como en la vista de Mapa, de forma que la información de estado de la plaza se muestre de un color u otro según su valor.



Mapa
center data: last_data type: map

[Show in preview](#) [Copy](#) [Delete](#)

View name
The unique name for this view.

Mapa

For this data
Which table or slice to display.

last_data

❖ Use slices to filter your data ⓘ

View type
What kind of view this is.

calendar deck table gallery detail map chart dashboard form onboarding card

Position
Where the button to access this view is located.

left most left center right right most menu ref

View Options

Map column
What column to use for location data.

address

Ilustración 60 - Configuración vista app: Mapa

Format Rules

Libre-address
condition: [status] = 0

Rule name
A unique name for this formatting rule

For this data
This rule applies to rows from this table

If this condition is true
Condition to check before formatting

Format these columns and actions
CTRL+click or CMD+click to select multiple columns

Visual Format

Icon

map-marker-alt

All Solid Regular Light

Highlight color

green

Text color

custom

Ilustración 61 - Regla de formato Mapa: pin verde cuando LIBRE



The screenshot shows the configuration interface for a 'Plazas' view in Tableau. At the top, there's a header with a list icon, the view name 'Plazas', a right arrow, 'data: last_data', 'type: deck', and buttons for 'Show in preview', 'Copy', 'Delete', and a refresh icon.

View name: Plazas

For this data: last_data

View type: What kind of view this is. Options include calendar, deck, table, gallery, detail, map, chart, dashboard, form, onboarding, and card. 'deck' is selected.

Position: Where the button to access this view is located. Options include left most, left, center, right (selected), right most, menu, and ref.

View Options:

- Sort by:** Sort the rows by one or more columns. An 'Add' button is available.
- Group by:** Group rows by the values in one or more of their columns. An 'Add' button is available.
- Group aggregate:** Display a numeric summary of the rows in each group. A dropdown menu is set to 'NONE'.
- Main image:** The image column to display for each row. A dropdown menu is shown.

Ilustración 62 - Configuración de vista app: Plazas (1)

This screenshot shows detailed configuration options for the 'Plaza' view:

- Primary header:** The top text for each row. Set to 'Plaza'.
- Secondary header:** The bottom text for each row. Set to 'Información del Tiempo'.
- Summary column:** The top-right text for each row. Set to 'Última actualización'.
- Nested table column:** An optional nested table (list of ref values). A dropdown menu is shown.
- Image shape:** What shape should the main image be? Options are Square Image (selected), Round Image, and Full Image.
- Show action bar:** Show action buttons at the bottom of each row? A toggle switch is turned on.
- Actions:** Which actions to include in the action bar? An 'Add' button is available.

Ilustración 63 - Configuración de vista app: Plazas (2)

Tras configurar la aplicación, se encuentra en estado “prototipo”. Si se quisiera desplegar (**Ilustración 64**), debería pasar ciertas comprobaciones. También se puede



convertir en aplicación “white-label” para que los usuarios la puedan descargar desde plataformas como Google Play Store, pero no se ha realizado porque para ello habría que obtener una licencia de pago para cada usuario que la descargue.

Aun sin ser “white-label”, la aplicación se puede compartir con hasta 10 usuarios para que la prueben, siendo suficiente para este Proyecto.

The screenshot shows the deployment interface for the 'Smart Parking' app. At the top, there's a navigation bar with icons for file operations, help, and settings, followed by buttons for 'SAVE', a dropdown menu, and a three-dot menu. Below the navigation is a tabs bar with 'Deployed' (highlighted with a green checkmark), 'Author', 'Versions', 'Integrations', 'Deploy' (underlined in blue), 'Monitor', and 'Broadcast Notifications'. On the left, a sidebar lists categories: Home, Info, Data, UX, Behavior, Automation, Security, Intelligence, and Manage (which is selected and highlighted in blue). The main content area has two sections: 'Deployment Check' (with a note about running a check before use) and 'White-Label' (with a note about converting to a custom mobile app). At the bottom of the main content are buttons for 'Pause app' and 'Switch to recovery mode'. A large blue circular button with a white square icon is located at the bottom left of the main content area.

Ilustración 64 - Desplegar app y convertir en white-label

Como no se ha publicado la aplicación, los usuarios con los que se comparta deben acceder a través de la propia aplicación móvil de AppSheet, donde se puede visualizar la aplicación desarrollada e interactuar con ella (véase **Compartir la App**).

En la **Ilustración 65** se puede ver el ícono de la aplicación de AppSheet y un acceso directo a la app creada: **Smart Parking**.

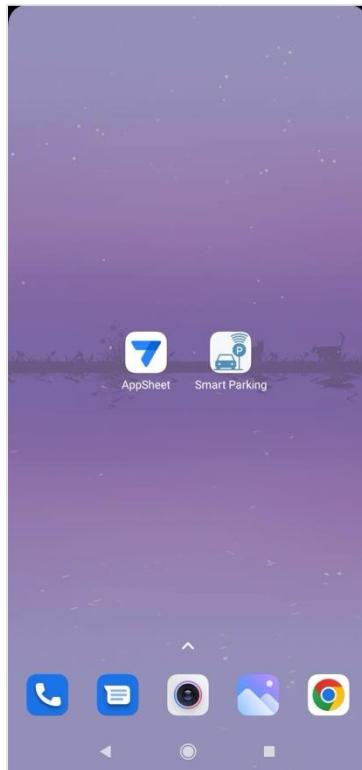


Ilustración 65 - Aplicación móvil AppSheet

Al acceder a la app **Smart Parking** (véase **Instalar y acceder a la App**) aparece de forma predeterminada la vista de “Mapa” (**Ilustración 66**), donde se puede ver la ocupación de las plazas, según el color del pin, y conocer parte de su información (véase **Pantalla principal: Mapa**).

Además, al ser AppSheet una herramienta de Google, incluye la integración con Google Maps, por lo que permite acceder a la propia aplicación de Google Maps y descubrir la ruta hasta la plaza (véase **Obtener ruta hasta la plaza**).

Por otro lado, la vista de “Plazas” se presenta en la **Ilustración 67**, donde se muestra un listado de las plazas y su ocupación, permitiendo acceder a la información completa de cada una (véase **Pantalla secundaria: Plazas**).



Ilustración 66 - Smart Parking: Mapa

Plaza	Estado	Fecha/Hora
Plaza 1 - LIBRE	Libre	22/10/2022 17:58
Plaza 2 - LIBRE	Libre	22/10/2022 17:58
Plaza 3 - OCUPADO	Ocupado	22/10/2022 17:58
Plaza 4 - LIBRE	Libre	22/10/2022 17:58
Plaza 5 - OCUPADO	Ocupado	22/10/2022 17:58
Plaza 6 - LIBRE	Libre	22/10/2022 17:58
Plaza 7 - OCUPADO	Ocupado	22/10/2022 17:58
Plaza 8 - LIBRE	Libre	22/10/2022 17:58

Ilustración 67 - Smart Parking: Plazas

Al configurar la tabla de la fuente de información (**Ilustración 58**), las columnas se pueden marcar como “*searchable*”, permitiendo así el filtrado de información en la aplicación. Por ejemplo, si se marca la columna Plaza como “*searchable*”, se podrán filtrar las plazas por LIBRE u OCUPADO (véase **Filtrar resultados**).



12. RESULTADOS EXPERIMENTALES

En este capítulo se detallan los resultados de las pruebas experimentales realizadas y los cambios que hayan podido ocasionar.

Siguiendo la estrategia de pruebas establecida en el capítulo de **Pruebas**, se comenzó por las **Pruebas Unitarias**. En la realización de las **Pruebas de caja blanca** aplicadas al dispositivo se comprobó que se realizaba correctamente el flujo de información y que el código se ejecutaba correctamente, entre otras cuestiones planteadas.

Seguidamente, se llevaron a cabo las **Pruebas de caja negra**, cuyos resultados se exponen a continuación:

Con el fin de comprobar si la metodología utilizada es apropiada, se colocaron dos vehículos con diferente cantidad de material ferromagnético sobre el dispositivo.

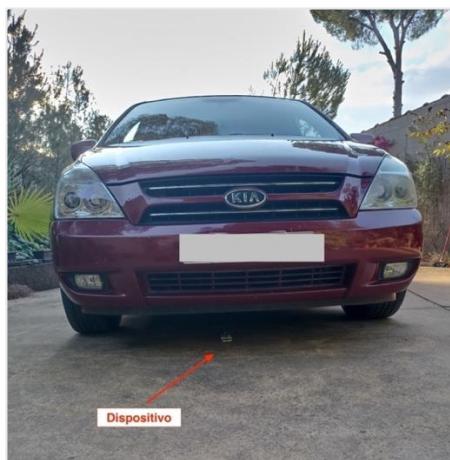


Ilustración 68 - Pruebas con vehículo KIA Carnival

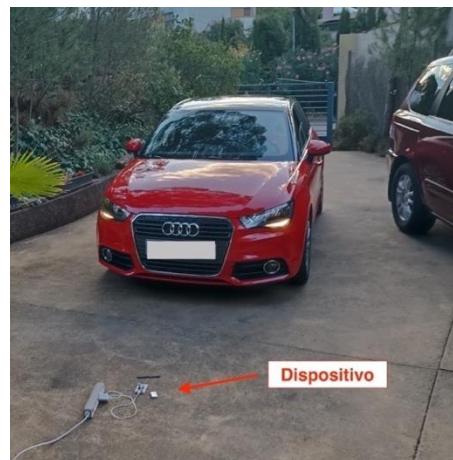


Ilustración 69 - Pruebas con vehículo Audi A1

Para los umbrales de comparación se utilizaron los establecidos inicialmente (véase la explicación del **Software del Prototipo de dispositivo**): 2.0 uT para la diferencia entre módulos y 3.0 ° para el ángulo entre los vectores.

Con esas condiciones, ambos vehículos, un KIA Carnival (**Ilustración 68**) y un Audi A1 (**Ilustración 69**), fueron detectados correctamente por el dispositivo (véase el **Anexo III - Pruebas**, en concreto **Pruebas para verificar la metodología**). De esta forma, se determinó que la metodología de detección utilizada era apropiada.



Paralelamente, se comprobó que el dispositivo enviaba y recibía correctamente los mensajes, modificando su comportamiento cuando fuera necesario: para llevar a cabo la detección de vehículos, el dispositivo debía tomar valores de referencia estando la plaza libre. Esto se le indicó con un mensaje de tipo 1 (véase el protocolo de mensajes *downlink* descrito en **Software**), el cual fue atendido correctamente.

Por otra parte, en estas pruebas se utilizó un período de envío de 1 minuto para ver claramente la detección. Con el fin de informar solamente al detectar un cambio de estado en la plaza, se envió un mensaje *downlink* de tipo 4 – opción 2, modificando así ese período.

Con relación a la precisión del dispositivo en la detección de vehículos, su posición juega un papel importante, como ya se comentó en el diseño de las **Pruebas de caja negra**. Para determinar la ubicación idónea, se evaluarían los valores obtenidos situándolo en 3 posiciones diferentes bajo un vehículo, representadas en la **Ilustración 70**: eje delantero (P1), centro del vehículo (P2) y eje trasero (P3).

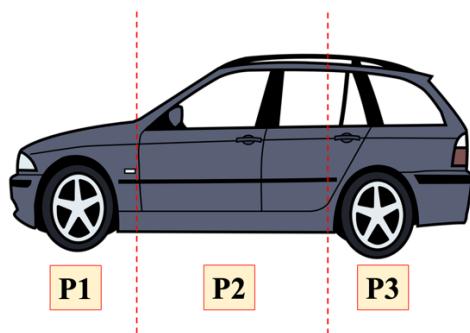


Ilustración 70 - Posibles posiciones del dispositivo respecto al vehículo

El análisis de Floris et al. (2022) concluye que la posición óptima para el sensor es bajo el eje delantero (P1) o el trasero (P3). Esto se reafirma en estudios de otros autores como Lou et al. (2020) o Sifuentes de la Hoya et al. (2015).

Sin embargo, impulsores de soluciones comerciales de *Smart Parking* como Libelium (2022a) o Bosch (2020), indican que sus dispositivos se deben colocar en el centro de la plaza (P2) para garantizar una mejor precisión del sensor.

En vista de la diferencia de opiniones, se realizó un pequeño estudio que determinara de forma definitiva la posición utilizada en la implantación de este sistema. En el anexo de las **Pruebas para determinar la posición del dispositivo y los umbrales**



de comparación se pueden ver los datos obtenidos, que se representan gráficamente en la **Ilustración 71**.

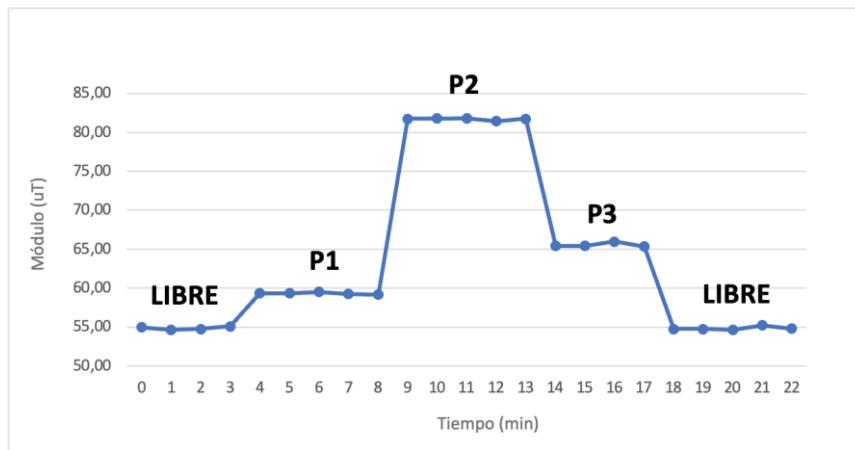


Ilustración 71 - Representación pruebas de posición del dispositivo

Siguiendo la metodología de Floris et al. (2022) para analizar estos datos, si se calcula la menor variación obtenida en cada posición y se compara con las variaciones mínimas en las otras posiciones (marcadas en color rojo en la **Tabla 43**), la mayor de ellas indicará en qué posición se detecta una distorsión superior de las líneas de flujo magnético. Esta será la posición que aporte mayor fiabilidad y precisión al dispositivo.

Tabla 43 - Comparativa para determinar la mejor posición

LIBRE (Referencia)	OCUPADO			VARIACIÓN MÓDULO		
	P1	P2	P3	P1	P2	P3
54,97	59,35	81,74	81,74	4,38	26,77	10,48
54,97	59,34	81,81	81,81	4,37	26,84	10,44
54,97	59,50	81,83	81,83	4,53	26,86	11,00
54,97	59,27	81,45	81,45	4,30	26,48	10,37
54,97	59,18	81,74	-	4,21	26,77	-

A partir de esta comparación se corrobora lo que se intuía en la **Ilustración 71**, y es que la mejor posición, según estos experimentos, sería bajo el centro del vehículo (P2).



Habiendo determinado la mejor posición, y con los valores obtenidos en estas mismas pruebas, se establecieron los umbrales de comparación definitivos que minimizarían los casos de falsos positivos o negativos.

Respecto al umbral de la diferencia entre módulos, se utiliza precisamente la mínima variación obtenida anteriormente para la posición P2. Sabiendo que el umbral de 2.0 uT establecido al inicio fue suficiente para detectar correctamente los vehículos en las **Pruebas para verificar la metodología**, y a fin de asegurar esa detección, se decidió determinar el tercio de la mínima variación en la posición P2 (26,48 uT) como valor umbral: 8,82 uT.

Por otro lado, con relación al umbral del ángulo entre vectores, también se determinó el tercio del mínimo ángulo obtenido en la posición P2 (13,95 °) como valor umbral: 4,65 °.

Se debe recordar que diferentes vehículos causan diferentes variaciones en la magnitud medida, incluso situando el dispositivo en la misma posición. Estas pruebas se han realizado con un único vehículo (Audi A1 de la **Ilustración 69**), por lo que estos resultados no se pueden generalizar.

Respecto a las pruebas referentes a la iluminación de la plaza, no se pudieron validar ya que, como se indicó en **Implantación**, no ha sido posible incluir esa funcionalidad en los dispositivos.

Tras las evaluaciones del dispositivo de detección, se llevaron a cabo aquellas relacionadas con la plataforma de monitorización y la aplicación móvil, comprobando sus funcionalidades de visualización, interacción con el usuario y análisis gráficos, en el caso del monitoreo.

Por otro lado, en la ejecución de las **Pruebas de Integración** se comprobó que el flujo de información entre los componentes se realizaba correctamente con cada incorporación, analizando en la consola de TTS los mensajes enviados por el dispositivo y, posteriormente, visualizando su información en la plataforma de monitorización y app.

Finalmente, en las **Pruebas del Sistema** se confirmó que ofrecía un funcionamiento y rendimiento adecuados, cumpliendo con las funcionalidades generales del sistema, excepto la relacionada con la información sobre la iluminación de la plaza.



A continuación, se muestra en la **Tabla 44** un resumen de la validación de las pruebas experimentales realizadas.

Tabla 44 – Validación de las pruebas experimentales

Pruebas Unitarias	
Validación interna del dispositivo	✓
Detección vehículos en dispositivo	✓
Envío información del dispositivo	✓
Recepción información en dispositivo	✓
Umbrales de comparación del dispositivo	✓
Detección iluminación en dispositivo	✗
Visualización en plataforma monitorización	✓
Análisis gráficos en plataforma monitorización	✓
Visualización en app	✓
Interacción en app	✓
Pruebas de Integración	
Flujo de información dispositivo a TTN	✓
Flujo de información TTN a plataforma monitorización	✓
Flujo de información TTN a app	✓
Pruebas del Sistema	
Proporcionar información sobre ocupación de plazas	✓
Proporcionar información sobre iluminación de plazas	✗
Proporcionar ubicación de plazas	✓
Modificar remotamente el comportamiento del dispositivo	✓
Visualizar información en plataforma y app	✓



13. CONCLUSIONES

13.1. CONCLUSIONES PERSONALES

La finalidad principal de este TFG era expandir el conocimiento en áreas del Internet de las Cosas nunca antes tratadas por la autora.

El desarrollo de este Proyecto ha permitido descubrir tecnologías inalámbricas empleadas en el IoT, reconociendo las diferencias y utilidades de cada una, así como las restricciones que conllevan. Se han estudiado diferentes sensores de obstáculos, comprendiendo cuáles son más útiles en un entorno en el exterior y cuáles mejoran su precisión en el interior.

Por otro lado, en la búsqueda de un método de detección de vehículos, se han adquirido conocimientos sobre el campo magnético terrestre, los sensores magnéticos y cómo convertir los datos que aportan estos sensores en información útil para el problema a resolver.

Así mismo, se han conocido diferentes plataformas de monitorización de la información, ganando experiencia en el uso del lenguaje JavaScript, el protocolo HTTP y en la utilización de APIs de servicios como los de Google u OpenWeatherMap.

Por otra parte, se han adquirido competencias en la búsqueda de información, analizando decenas de artículos de investigación y desarrollo, y corroborando sus resultados. Además, se han obtenido habilidades para diseñar e implementar desde cero un proyecto de ingeniería, poniendo en práctica conocimientos sobre la documentación de un sistema especialmente dirigidos a la creación del manual técnico y de usuario.

A nivel académico, se han aplicado competencias de asignaturas como *Proyectos*, en el desarrollo de la documentación, *Sistemas Tolerantes a Fallos*, en el diseño del dispositivo y sistema en general, o *Redes de Comunicaciones Específicas*, en el estudio de redes de sensores inalámbricas y en la manipulación del dispositivo Arduino.



13.2. CONCLUSIONES TÉCNICAS

En la elaboración de este TFG, enfocado a una movilidad sostenible, se ha desarrollado un sistema prototípico para la monitorización de plazas de aparcamiento.

Se considera que este sistema cumple con los objetivos más importantes, pues se ha implementado un prototípico de dispositivo capaz de detectar vehículos con una alta precisión, para lo que se ha diseñado un algoritmo de detección personalizado para el problema en cuestión. Además, se ha implementado un sistema de comunicación inalámbrica que permite enviar la información tanto a una plataforma de monitorización, dedicada al administrador del sistema, como a una aplicación móvil, dedicada al usuario final.

A nivel general, se han cumplido los requisitos establecidos con relación a la detección de vehículos, transmisión y recepción de datos, y visualización de la información.

Sin embargo, no ha sido posible cumplir uno de los requisitos del dispositivo y, por defecto, del sistema: la funcionalidad que permitiría ofrecer información sobre si la plaza se encuentra bajo luz solar directa. Esta función requeriría un envío constante de datos por parte del dispositivo para mantener la información actualizada, algo que, con el prototípico diseñado, podría drenar la batería en muy poco tiempo. En un futuro, con una investigación más profunda y, probablemente, un nuevo diseño, es posible que se pudiera ofrecer esta capacidad.

En ausencia de la funcionalidad anterior, se incluyó un servicio de consulta de información meteorológica en la ubicación de las plazas, utilizando una API que permitiera a los usuarios conocer una previsión del tiempo en la zona donde quieren aparcar.

Finalmente, para maximizar la precisión del dispositivo en la detección de vehículos, se realizó un estudio de posición para saber en qué lugar de la superficie de la plaza se debería colocar el dispositivo, determinando el centro de la plaza como mejor opción.



14. FUTURAS MEJORAS

El sistema desarrollado, al igual que el dispositivo de detección de vehículos, es un prototipo, por lo que se proponen posibles mejoras a implementar en un futuro:

- Para mejorar la fiabilidad del dispositivo, se podría aplicar la redundancia de sensores formando un sistema TMR.
- Como reto tecnológico, se podría mejorar el rendimiento y uso de la fuente de alimentación, alargando su autonomía.
- Para disminuir el tamaño del dispositivo y minimizar el riesgo de fallos en el cableado, se podrían integrar los componentes del dispositivo en una placa de circuito impreso (PCB).
- Investigar métodos de detección en tiempo real de la iluminación u otras formas estimativas, como la información aportada por el dispositivo de Ramírez Faz (2005), denominado *Light Ray Tracing and Visioning Device* (LRTVD). Se trata de una cámara hemisférica que aportaría una proyección con las trayectorias solares vistas desde una ubicación específica y permitiría estimar si la plaza de aparcamiento recibiría luz solar directa en un momento determinado.
- Perfeccionar la obtención de umbrales de comparación utilizados en el algoritmo de detección de vehículos, produciendo un mecanismo dinámico e independiente de la localización del dispositivo.
- Diseñar la envolvente del dispositivo siguiendo estándares de fabricación, protección y seguridad.
- Mejorar la interacción entre el usuario y la aplicación móvil, aportando funcionalidades como la ocupación virtual temporal de una plaza de aparcamiento por parte de un usuario, la cual aparecería como ocupada al resto de usuarios durante un corto período de tiempo y a la espera de ser realmente utilizada.



15. BIBLIOGRAFÍA

- 3GPP. (2016). *Standardization of NB-IoT completed*. https://www.3gpp.org/news-events/1785-nb_iot_complete
- Adafruit. (2020). *Repositorio librería Adafruit_HMC5883_U*. https://github.com/adafruit/Adafruit_HMC5883_Unified
- Adafruit. (2022). *Repositorio librería Adafruit_Sensor*. https://github.com/adafruit/Adafruit_Sensor
- Alenezi, M., Chai, K. K., Chen, Y., & Jimaa, S. (2020). Ultra-dense LoRaWAN: Reviews and challenges. *IET Communications*, 14(9), 1361–1371. <https://doi.org/10.1049/iet-com.2018.6128>
- Apple Wiki. (2022). *macOS 12.4*. https://apple.fandom.com/wiki/MacOS_12.4
- Arduino. (2021a). *Arduino IDE*. <https://www.arduino.cc/en/software>
- Arduino. (2021b). *Repositorio librería MKRWAN*. <https://github.com/arduino-libraries/MKRWAN>
- Arduino. (2022a). *Arduino Language*. <https://www.arduino.cc/reference/en/>
- Arduino. (2022b). *Arduino MKR WAN 1310*. <https://store.arduino.cc/products/arduino-mkr-wan-1310>
- Arduino. (2022c). *Connecting MKR WAN 1310 to The Things Network (TTN)*. <https://docs.arduino.cc/tutorials/mkr-wan-1310/the-things-network>
- Arduino. (2022d). *MKR WAN 1310*. <https://docs.arduino.cc/hardware/mkr-wan-1310>
- AZ-Delivery. (2022). *Módulo de brújula GY-271*. <https://www.az-delivery.de/es/products/gy-271-kompassmodul-kompass-magnet-sensor-fuer-arduino-und-raspberry-pi>
- Bachani, M., Qureshi, U. M., & Shaikh, F. K. (2016). Performance Analysis of Proximity and Light Sensors for Smart Parking. *Procedia Computer Science*, 83, 385–392. <https://doi.org/10.1016/j.procs.2016.04.200>
- Bosch. (2019). *Parking Sensor - Datasheet*. <https://lora-alliance.org/wp-content/uploads/2019/07/parking-lot-sensor-datasheet.pdf>
- Bosch. (2020). *Parking Lot Sensor TPS110 EU - User manual*. <https://www.alliot.co.uk/wp-content/uploads/2020/01/Bosch-parking-lot-sensor-user-manual-en.pdf>
- Bosch. (2021). *BOSCH Parking Lot Sensor*. https://lora-alliance.org/lora_products/bosch-parking-lot-sensor/
- Byte Lab. (2022). *LPWAN technologies used in IoT*. <https://www.byte-lab.com/lpwlan-technologies-used-in-iot/>
- Campos, F. (2020). *LPWAN: qué son y para qué se utilizan*. <https://www.m2mlogitek.com/lpwan-que-son-y-para-que-se-utilizan/>
- diagrams.net. (2022). *diagrams.net*. <https://www.diagrams.net/doc/>
- EEMB, & Amazon. (2022). *Batería Li-Po 3.7V 2000mAh*. <https://www.amazon.es/EEMB-103454-Recargable-pol%C3%ADmero-Conector/dp/B08214DJLJ>
- ETSI. (2017). *ETSI EN 300 220-1 European Standard*. https://www.etsi.org/deliver/etsi_en/300200_300299/30022001/03.01.01_60/en_30022001v030101p.pdf
- Floris, A., Porcu, S., Atzori, L., & Girau, R. (2022). A Social IoT-based platform for the deployment of a smart parking solution. *Computer Networks*, 205, 108756. <https://doi.org/10.1016/j.comnet.2021.108756>
- Fritzing. (2022). *Fritzing*. <https://fritzing.org/download/>
- Google. (2022). *Google Sheets*. <https://www.google.es/intl/es/sheets/about/>



- Google Apps Script. (2022a). *Spreadsheet Service*. <https://developers.google.com/apps-script/reference/spreadsheet?hl=en>
- Google Apps Script. (2022b). *Web Apps*. <https://developers.google.com/apps-script/guides/web?hl=en>
- Google Cloud. (2022). *Google AppSheet*. <https://cloud.google.com/appsheet?hl=es>
- Google Workspace. (2022). *Google Apps Script*.
<https://workspace.google.com/intl/es/products/apps-script/>
- GSMA. (2019). *NB-IoT Deployment Guide – Release 3*. <https://www.gsma.com/iot/resources/nbiot-deployment-guide-v3/>
- GSMA. (2022). *Mobile IoT Network Launches*. <https://www.gsma.com/iot/mobile-iot-commercial-launches/>
- Gupta, A., Kulkarni, S., Jathar, V., Sharma, V., & Jain, N. (2017). Smart Car Parking Management System Using IoT. *American Journal of Science, Engineering and Technology*, 2(4), 112–119.
- Heliot Europe. (2022, April 21). *UnaBiz takes over IoT provider and network operator Sigfox*.
<https://www.heliotgroup.com/en/unabiz-takes-over-iot-provider-and-network-operator-sigfox/>
- Hossain, M. I., & Markendahl, J. I. (2021). Comparison of LPWAN Technologies: Cost Structure and Scalability. *Wireless Personal Communications*, 121(1), 887–903.
<https://doi.org/10.1007/s11277-021-08664-0>
- Kadusic, E., Zivic, N., Ruland, C., & Hadzajlic, N. (2022). A Smart Parking Solution by Integrating NB-IoT Radio Communication Technology into the Core IoT Platform. *Future Internet*, 14(8), 219. <https://doi.org/10.3390/fi14080219>
- Kotb, A. O., Shen, Y.-C., Zhu, X., & Huang, Y. (2016). iParker—A New Smart Car-Parking System Based on Dynamic Resource Allocation and Pricing. *IEEE Transactions on Intelligent Transportation Systems*, 17(9), 2637–2647. <https://doi.org/10.1109/TITS.2016.2531636>
- Libelium. (2022a). *Deployment and installation*. <https://development.libelium.com/smart-parking-technical-guide/deployment>
- Libelium. (2022b). *Smart Parking node*. <https://development.libelium.com/smart-parking-technical-guide/smartparkingnode>
- Libelium. (2022c). *Smart Parking*. <https://www.libelium.com/iot-products/smart-parking/>
- Lin, T., Rivano, H., & le Mouel, F. (2017). A Survey of Smart Parking Solutions. *IEEE Transactions on Intelligent Transportation Systems*, 18(12), 3229–3253.
<https://doi.org/10.1109/TITS.2017.2685143>
- LoRa Alliance. (2015). *A technical overview of LoRa® and LoRaWAN*.
<https://www.tuv.com/content-media-files/master-content/services/products/1555-tuv-rheinland-lora-alliance-certification/tuv-rheinland-lora-alliance-certification-overview-lora-and-lorawan-en.pdf>
- LoRa Alliance. (2021). *LoRaWAN® Formally Recognized as ITU International Standard for Low Power Wide Area Networking*. <https://lora-alliance.org/lora-alliance-press-release/lorawan-formally-recognized-as-itu-international-standard-for-low-power-wide-area-networking/>
- Lou, L., Li, Q., Zhang, Z., Yang, R., & He, W. (2020). An IoT-Driven Vehicle Detection Method Based on Multisource Data Fusion Technology for Smart Parking Management System. *IEEE Internet of Things Journal*, 7(11), 11020–11029. <https://doi.org/10.1109/JIOT.2020.2992431>
- Lucidchart. (2022). *Lucidchart Overview*. <https://www.lucidchart.com/pages/product>
- MDN Web Docs. (2022). *JavaScript*. <https://developer.mozilla.org/es/docs/Web/JavaScript>
- Mekki, K., Bajic, E., Chaxel, F., & Meyer, F. (2019). A comparative study of LPWAN technologies for large-scale IoT deployment. *ICT Express*, 5(1), 1–7.
<https://doi.org/10.1016/j.icte.2017.12.005>
- Microchip, & Arduino. (2018). *Microchip SAMD21 datasheet*. https://content.arduino.cc/assets/mkr-microchip_samd21_family_full_datasheet-ds40001882d.pdf
- Microsoft 365. (2022). *Microsoft Word*. <https://www.microsoft.com/es-es/microsoft-365/word?activetab=tabs%3afaqheaderregion3>



- Ministerio de Ciencia y Tecnología. (2021). *Reglamento electrotécnico para baja tensión e ITC*. https://www.boe.es/biblioteca_juridica/codigos/codigo.php?id=326&modo=2¬a=0
- MITMA. (2021). *Eje 5 - Movilidad Inteligente / Estrategia de Movilidad Segura, Sostenible y Conectada 2030*. <https://esmovilidad.mitma.es/ejes-estrategicos/movilidad-inteligente>
- Murata, & Arduino. (2018). *Murata CMWX1ZZABZ LoRa module datasheet*. https://content.arduino.cc/assets/mkrwan1310-murata_lora_module-type_abz.pdf
- myDevices. (2018). *Repositorio librería CayenneLPP*. <https://github.com/myDevicesIoT/CayenneLPP>
- myDevices. (2022). *Cayenne Docs*. <https://developers.mydevices.com/cayenne/docs/intro/>
- Nižetić, S., Djilali, N., Papadopoulos, A., & Rodrigues, J. J. P. C. (2019). Smart technologies for promotion of energy efficiency, utilization of sustainable resources and waste management. *Journal of Cleaner Production*, 231, 565–591. <https://doi.org/10.1016/j.jclepro.2019.04.397>
- OpenWeatherMap. (2022). *One Call API 3.0*. <https://openweathermap.org/api/one-call-3>
- Paidi, V., Fleyeh, H., Håkansson, J., & Nyberg, R. G. (2018). Smart parking sensors, technologies and applications for open parking lots: a review. *IET Intelligent Transport Systems*, 12(8), 735–741. <https://doi.org/10.1049/iet-its.2017.0406>
- Pardal, P. (2017). *Redes de Área Extensa para aplicaciones de IoT: modelado de comunicaciones Sigfox*.
- Perković, T., Damjanović, S., Šolić, P., Patrono, L., & Rodrigues, J. J. P. C. (2020). Meeting Challenges in IoT: Sensing, Energy Efficiency, and the Implementation. In *Fourth International Congress on Information and Communication Technology* (pp. 419–430). https://doi.org/10.1007/978-981-15-0637-6_36
- Perković, T., Šolić, P., Zargariasl, H., Čoko, D., & Rodrigues, J. J. P. C. (2020). Smart Parking Sensors: State of the Art and Performance Evaluation. *Journal of Cleaner Production*, 262, 121181. <https://doi.org/10.1016/j.jclepro.2020.121181>
- Ramírez Faz, J. (2005). *Desarrollo de un Prototipo para la Cuantificación de la Disponibilidad de Radiación Solar en Ubicaciones Sombreadas*. E.T.S.I. Industriales UNED.
- Rueda, A. (2022). Un 35% del tráfico lo generan conductores que buscan aparcamiento. *Motor EL PAÍS*. <https://motor.elpais.com/actualidad/un-35-del-trafico-lo-generan-conductores-que-buscan-aparcamiento/>
- Rupareliya, K. (2022). *LPWAN vs. LoRaWAN®: The Better Technology For IoT Device Connectivity*. <https://www.intuz.com/blog/lpwan-vs-lorawan-the-better-technology-for-iot-device>
- Sabbea, M. O. B. (2017). Design and Development of a Smart Parking System. *Journal of Automation and Control Engineering*, 66–69. <https://doi.org/10.18178/jace.6.2.66-69>
- Santos, S. (2021). *ESP8266 NodeMCU with HC-SR04 Ultrasonic Sensor with Arduino IDE*. <https://randomnerdtutorials.com/esp8266-nodemcu-hc-sr04-ultrasonic-arduino/>
- Šarić, A., & Mihaljević, B. (2017). Smart Parking System in the City of Dubrovnik. *RITHink Multidisciplinary Online Journal*, 6. https://www.rithink.hr/brocure/pdf/vol6_2017/1509301359_4__Andrej__ari__Branko_Mihaljevi__SMART_PARKING_SYSTEM_IN_THE_CITY_OF_DUBROVNIK.pdf
- Semtech. (2019). *Understanding the LoRa® Adaptive Data Rate*. https://lora-developers.semtech.com/uploads/documents/files/Understanding_LoRa_Adaptive_Data_Rate_Downloadable.pdf
- Semtech. (2020). *LoRa® and LoRaWAN®: A Technical Overview*. https://lora-developers.semtech.com/uploads/documents/files/LoRa_and_LoRaWAN-A_Tech_Overview-Downloadable.pdf
- Shea, S. (2017). *LPWAN (low-power wide area network)*. <https://www.techtarget.com/iotagenda/definition/LPWAN-low-power-wide-area-network>
- Sifuentes de la Hoya, E. , Enríquez Aguilera, F. J., Cota Ruiz, J. de D., & Silva Aceves, J. M. (2015). MAGNETÓMETRO DE BAJO COSTO PARA DETECTAR VEHÍCULOS ESTÁTICOS. *Cultura Científica Y Tecnológica*, 2(49), 74–84.



- Sigfox. (n.d.). *Sigfox Coverage*. <https://www.sigfox.com/en/coverage>
- Solic, P., Perkovic, T., Konsa, T., Zargariasl, H., & Patrono, L. (2019). Smart Parking Sensor Performance Evaluation. *2019 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 1–6. <https://doi.org/10.23919/SOFTCOM.2019.8903827>
- SunFounder. (2021a). *GY-271 HMC5883L 3-Axis Magnetic Electronic Compass Module*. http://wiki.sunfounder.cc/index.php?title=GY-271_HMC5883L_3-Axis_Electronic_Compass_Module
- SunFounder. (2021b). *IR Obstacle Avoidance Sensor*. https://docs.sunfounder.com/projects/sensorkit-v2-arduino/en/latest/lesson_31.html#lesson-31-ir-obstacle-avoidance-sensor
- SunFounder. (2021c). *Photoresistor sensor*. https://docs.sunfounder.com/projects/electronic-kit/en/latest/lesson_5_Photoresistor_a.html
- The Things Industries. (2022). *TTS/TTN Console*. <https://www.thethingsindustries.com/docs/getting-started/console/>
- The Things Network. (n.d.). *LoRaWAN Architecture*. <https://www.thethingsnetwork.org/docs/lorawan/architecture/>
- The Things Network. (2020a). *CayenneLPP*. <https://www.thethingsnetwork.org/docs/devices/arduino/api/cayennelpp/>
- The Things Network. (2020b). *Duty Cycle*. <https://www.thethingsnetwork.org/docs/lorawan/duty-cycle/>
- The Things Network. (2021). *What are LoRa and LoRaWAN?* <https://www.thethingsnetwork.org/docs/lorawan/what-is-lorawan/>
- The Things Network. (2022). *Applications & Integrations*. <https://www.thethingsnetwork.org/docs/applications-and-integrations/>
- The Things Stack. (2021a). *Cayenne*. <https://www.thethingsindustries.com/docs/integrations/cloud-integrations/cayenne/>
- The Things Stack. (2021b). *CayenneLPP*. <https://www.thethingsindustries.com/docs/integrations/payload-formatters/cayenne/>
- The Things Stack. (2022a). *Adding Devices*. <https://www.thethingsindustries.com/docs/devices/adding-devices/>
- The Things Stack. (2022b). *What Is The Things Stack?* <https://www.thethingsindustries.com/docs/getting-started/what-is-tts/>
- UnaBiz Spain. (n.d.). *¿Qué es Sigfox?* <https://www.sigfox.es/faqs>
- Väänänen, O., & Hämäläinen, T. (2022). Efficiency of temporal sensor data compression methods to reduce LoRa-based sensor node energy consumption. *Sensor Review*, 42(5), 503–516. <https://doi.org/10.1108/SR-10-2021-0360>
- Vîlceanu, R., Ternauciu, A., & Onița, M. (2019). A Student Smart Parking Solution using Raspberry Pi. *International Journal of Innovative Technology and Exploring Engineering*, 9(1), 2727–2733. <https://doi.org/10.35940/ijitee.A4912.119119>
- Vodafone España. (2022). *Mapa de Cobertura Móvil Vodafone*. <https://www.vodafone.es/c/conocenos/es/vodafone-espana/mapa-cobertura-movil/>
- VONVOFF, & Amazon. (2022). *Caja de conexiones estanca*. <https://www.amazon.es/Conexiones-Impermeable-instrumentos-electrónica-100x68x50mm/dp/B08QV1FNV4?th=1>
- Vos, G. (2022). *What is Narrowband IoT (NB-IoT)?* <https://www.sierrawireless.com/iot-blog/what-is-nb-iot/>



A. ANEXO I – MANUAL DE USUARIO





1. Introducción

En este anexo se ilustra el uso del dispositivo de detección, la plataforma de monitorización y la aplicación móvil, para lo que se ha dividido en tres secciones:

- **Manual del Dispositivo:** muestra el proceso de instalación y desinstalación, indicando los diferentes pasos a seguir en cada caso, así como el método de envío de mensajes *downlink*. Está dirigido al administrador del sistema.
- **Manual de la Plataforma de Monitorización:** muestra cómo administrar los dispositivos, cómo acceder al histórico de datos y cómo realizar análisis gráficos. También está dirigido al administrador del sistema.
- **Manual de la App:** muestra cómo acceder a la app y cómo navegar por ella, visualizando el mapa y el listado de plazas. Este manual está especialmente dirigido al usuario final.

2. Manual del Dispositivo

2.1. Integrar dispositivo en el sistema

Para la incorporación exitosa del dispositivo en el sistema se deben seguir diferentes pasos:

- (1) Añadir a TTN
- (2) Instalar en plaza de aparcamiento
- (3) Inicializar dispositivo
- (4) Añadir a Cayenne
- (5) Añadir a App

2.1.1. Añadir dispositivo a aplicación en TTN/TTS

El primer paso para integrar el dispositivo en el sistema es añadirlo a la aplicación del servidor de red (TTS). Esto se puede hacer siguiendo varios tutoriales (Arduino, 2022c; The Things Stack, 2022a), pero a continuación se muestran los pasos más importantes.



Como se comentó en **Conexión y uso de TTN**, para la conexión con la red LoRaWAN son necesarias las siguientes claves:

- **DevEUI**: único para el módulo LoRa integrado en el dispositivo, por lo que es aportado por el fabricante. Se puede consultar desde el IDE Arduino con el siguiente programa:

```
LoRaModem modem;

void setup() {
    modem.begin(EU868);
    Serial.print("Your device EUI is: ");
    Serial.println(modem.deviceEUI());
}
```

Ilustración 72 - Obtener DevEUI

- **AppEUI** (o JoinEUI): identificador que, si no lo aporta el fabricante, como ocurre con el Arduino MKR WAN 1310, se puede obtener generando un número aleatorio acorde a los estándares IEEE 802¹¹ o utilizando una cadena de ceros. Puede utilizarse el mismo para varios dispositivos.
- **AppKey**: si no lo aporta el fabricante, puede generarse aleatoriamente. Debe ser única para cada dispositivo.

Para añadir un nuevo dispositivo, debe situarse en la consola de TTS, en la aplicación correspondiente, seleccionar “+ Add end device” y configurar los diferentes parámetros que aparecen. Para escoger el modelo de dispositivo, se puede utilizar el repositorio de dispositivos LoRaWAN que incluye TTN:

¹¹ <https://www.ieee802.org>



The screenshot shows the 'Applications' tab selected in the header. Below it, the path 'Applications > Proyecto TFG > End devices' is visible. The main area displays a table of 'End devices' with one entry: 'second-mkr' (ID: MKR 2, DevEUI: AB 61 0A 33 34 3..., JoinEUI: 00 00 00 00 00 0..., Last activity: Aug 16, 2022). The sidebar on the left includes links for 'Live data', 'Payload formatters', 'Integrations', 'Collaborators', 'API keys', and 'General settings'. The 'End devices' link is highlighted with a red box and a circled '2'.

Ilustración 73 - Añadir dispositivo a TTN (1)

The screenshot shows the 'Register end device' page. It starts with a QR code scanning section. Below it, the 'End device type' section is highlighted with a red box and a circled '4'. Under 'Input Method', the 'Select the end device in the LoRaWAN Device Repository' option is selected. The 'Profile (Region)' dropdown is also highlighted with a red box. The 'Frequency plan' dropdown is highlighted with a red box and a circled '5'. The page includes a detailed description of the 'Arduino MKR WAN 1310' board and a 'Product website' link.

Ilustración 74 - Añadir dispositivo a TTN (2)



The screenshot shows the TTN web interface for adding a new device. The left sidebar lists various project components: Projecto TFG, Overview, End devices (highlighted), Live data, Payload formatters, Integrations, Collaborators, API keys, and General settings. The main area displays the Arduino MKR WAN 1310 development board and its technical specifications under the heading '(OTAA), Class A'. Below this is a 'Product website' link. The 'Frequency plan' dropdown is set to 'Europe 863-870 MHz (SF9 for RX2 - recommended)'. The 'Provisioning information' section contains fields for 'JoinEUI' (with a red circle labeled 6), 'DevEUI' (containing the value 'A8 61 0A 33 34 4B 71 1E'), 'AppKey' (containing the value 'AB BB 1C B2 56 2A 69 A6 29 30 D0 6B C9 81 27 1'), and 'End device ID' (containing 'my-mkr-wan'). The 'After registration' section has two radio button options: 'View registered end device' (selected) and 'Register another end device of this type'. At the bottom right of the provisioning section is a large red-bordered 'Register end device' button (labeled 7).

Ilustración 75 - Añadir dispositivo a TTN (3)

Las claves AppEUI (JoinEUI) y AppKey deben indicarse en el fichero de configuración del dispositivo (véase los **Parámetros de conexión LoRaWAN – ARDUINO_SECRETS.h**) para poder establecer la conexión con la red LoRaWAN.

Finalmente, se debe configurar la ubicación exacta donde se va a instalar el dispositivo:



The screenshot shows the 'Applications' section of The Things Stack. The sidebar includes links for Overview, End devices (highlighted with a red circle 2), Live data, Payload formatters, Integrations, Collaborators, API keys, and General settings. The main area displays a table of end devices with columns for ID, Name, DevEUI, JoinEUI, and Last activity. Two devices are listed: 'second-mkr' and 'my-mkr-wan' (highlighted with a red circle 3).

Ilustración 76 - Configurar ubicación de dispositivo en TTN (1)

The screenshot shows the 'Location' tab for the device 'My WANda'. The sidebar includes links for Overview, End devices (highlighted with a red circle 4), Live data, Payload formatters, Integrations, Collaborators, API keys, and General settings. The main area shows a map of a city area with a blue marker indicating the device's location. Below the map, there is a form to 'Set end device location manually' with fields for Latitude, Longitude, and Altitude. The 'Latitude' field contains '37,914765', the 'Longitude' field contains '-4,724311', and the 'Altitude' field contains '131'. At the bottom, there are buttons for 'Save changes' (highlighted with a red circle 6) and 'Remove location data'.

Ilustración 77 - Configurar ubicación de dispositivo en TTN (2)



2.1.2. Instalar dispositivo en plaza de aparcamiento

El dispositivo debe instalarse en el centro de la plaza para que los vehículos sitúen su eje central sobre el mismo, de forma que se maximice su precisión.

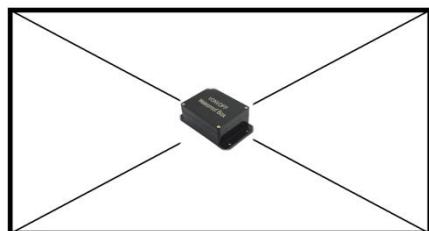


Ilustración 78 - Instalar dispositivo en superficie de la plaza

Debido a la estructura de la caja escogida como carcasa envolvente (**Ilustración 25**), hay dos opciones de instalación:

- A) **Sobre la superficie:** la caja envolvente del dispositivo se debe fijar a la superficie de la plaza mediante un componente adhesivo, por lo que la zona del pavimento no debe tener huecos que lo impidan, pues es crucial para la superficie de contacto y el efecto adhesivo. Para ello, se debe aplicar el adhesivo en el centro de la base del dispositivo, colocarlo sobre la superficie y aplicar presión. Además, para asegurar la adhesión se debe atornillar a la superficie con tornillos de 6 mm de diámetro.
- B) **Semienterrado:** se debe perforar un agujero de 135 mm de ancho y 35 mm de profundidad, introducir la carcasa y llenar el espacio sobrante con cemento, permitiendo que la envolvente sobresalga de la superficie unos 15 mm aproximadamente.

El dispositivo de detección en sí mismo se coloca dentro de la caja, que se debe sellar utilizando cuatro tornillos en su parte superior. Esto permitiría un acceso rápido y cómodo cuando hubiera que llevar a cabo alguna actividad de mantenimiento, como el cambio de batería.



2.1.3. Inicialización del dispositivo

Una vez instalado en la superficie de la plaza, se inicializa el dispositivo para que realice la conexión con la red LoRaWAN (véase **Cuerpo principal – MAIN.ino**). En el caso del dispositivo diseñado, la inicialización consistiría en conectar la batería o, si ya lo estuviera, en resetear el dispositivo.

Posteriormente, se le debe indicar que tome los valores actuales del campo magnético como referencia de que la plaza se encuentra LIBRE. Para ello, desde la consola de TTS se debe acceder al dispositivo en cuestión y enviarle un mensaje *downlink* en HEX con la cadena de caracteres correspondiente. El mensaje que indica “tomar referencia” es del tipo 1 y se codifica como “1”, por lo que esta cadena en HEX sería: 31

The screenshot shows the THE THINGS STACK Community Edition interface. On the left, there's a sidebar with options like Overview, End devices (highlighted with a red box and circled with a red number 2), Live data, Payload formatters, Integrations, Collaborators, API keys, and General settings. The main area has tabs for Overview, Applications (highlighted with a red box and circled with a red number 1), Gateways, and Organizations. Under Applications, it says EU1 Community, No SLA applicable. The main content area shows the 'Applications > Proyecto TFG > End devices' section. It lists two end devices: 'second-mkr' (DevEUI: A8 61 0A 33 34 3..., Name: MKR 2) and 'my-mkr-wan' (DevEUI: A8 61 0A 33 34 4..., Name: My WANda). Both entries have a 'Last activity' column showing dates (Aug 16, 2022 and Aug 17, 2022 respectively).

Ilustración 79 - Inicializar dispositivo (1)



The screenshot shows the Project TFG interface. On the left, there's a sidebar with options like Overview, End devices (which is selected), Live data, Payload formatters, Integrations, Collaborators, API keys, and General settings. The main area shows a device named "My WANda" with ID "my-mkr-wan". Below the device name, there are statistics: ↑ 177, ↓ 177, and Last activity Aug 17, 2022. A navigation bar at the top of the main content area includes tabs: Overview, Live data, Messaging (selected), Location, Payload formatters, Claiming, and General settings. The "Messaging" tab is highlighted with a red box and has a red circle with the number 4 above it. Below the tabs, there are two buttons: "Uplink" and "Downlink", with "Downlink" also highlighted with a red box and a red circle with the number 5 above it. The main content area is titled "Schedule downlink" and contains sections for "Insert Mode" (with "Replace downlink queue" selected), "FPort*" (set to 1), "Payload type" (with "Bytes" selected), and "Payload" (containing the value "31"). A red box surrounds the "Payload" section, and a red circle with the number 6 is placed to its left. At the bottom right of the main content area, there's a button labeled "Schedule downlink" with a red box around it and a red circle with the number 7 above it. A link "Hide sidebar" is located at the bottom left.

Ilustración 80 - Inicializar dispositivo (2)



2.1.4. Añadir dispositivo a plataforma de monitorización

En la plataforma Cayenne myDevices se pueden añadir diferentes microcontroladores, sensores, ordenadores SBC, etc. Los dispositivos con tecnología LoRa se encuentran catalogados según las redes a las que están conectados.

The screenshot shows the Cayenne web interface with the following interface elements:

- Header:** Cayenne Powered by myDevices, SmartParking +, Create App, Community, Docs, User Menu.
- Left Sidebar:** Add new..., Device/Widget (highlighted with a red box and circled 1), Event, Trigger, Project.
- Central Panel:** A tree view of device categories:
 - Single Board Computers
 - Microcontrollers (highlighted with a red box and circled 2)
 - Sensors
 - Actuators
 - Extensions
 - LoRa (highlighted with a red box and circled 3)
 - Acklio
 - Actility
 - ChirpStack
 - CityKinect
 - Everynet
 - Helium
 - Kerlink
 - Loriot
 - Objenious
 - OrbiWise
 - Pixel Networks
 - Sagemcom
 - Semtech
 - Senet
 - SenRa
 - Spark
 - Stream
 - Swisscom
 - The Things Network
 - X-Telia
- Right Panel:** A list of LoRa devices with small icons:
 - 1M2M ED160B: Generic with many sensors and connectors
 - AAEON AIOT-ILNDO1: Industrial LoRa Node platform
 - Abeway MasterTracker: Low Power Industrial GPS Tracker
 - Abeway Micro Tracker: GPS & BLE Tracker
 - AC Outlet and Switch: Tektelic AC Control and Energy Monitoring
 - AcSiP EK-S765XB: S765 EVB in X-Bee Form Factor
 - AcSiP S765: LoRa development board
 - Adeunis Analog EU: Giving IoT connectivity to wired sensors

Ilustración 8I - Añadir dispositivo a Cayenne (1)

Debido a que el Arduino MKR WAN 1310 no se encuentra entre los modelos oficiales de la red *The Things Network*, se debe utilizar un modelo genérico CayenneLPP disponible para los dispositivos que utilizan ese formato de mensaje. Este modelo también existe en otras redes como *Actility*, por lo que hay que asegurarse que se escoge el de TTN. Si no es así, la información no llegará a Cayenne ya que se habrá escogido el servidor de red erróneo.

En la configuración de ese modelo genérico solo hay que introducir el DevEUI del dispositivo, indicar que ya está registrado en la red TTN y que su ubicación es estática.



Devices & Widgets

- DEVICES
- Single Board Computers
- MicroControllers
- Sensors
- Actuators
- Extensions
- LoRa
 - Acklio
 - Actility
 - ChirpStack
 - CityKinect
 - Everynet
 - Helium
 - Kerlink
 - Loriot
 - Objenious

Seeed SenseCAP Wireless Air Temperature and Humidity Sensor
Barometric Pressure

1M2M ED1608
Generic with many sensors and connectors

AAEON AIOT-ILND01
Industrial LoRa Node platform

Abeeway MasterTracker
Low Power Industrial GPS Tracker

Abeeway Micro Tracker
GPS & BLE Tracker

AC Outlet and Switch
Tektelic AC Control and Energy Monitoring

Enter Settings

Cayenne Cayenne LPP
Cayenne Low Power Payload

This device uses Cayenne LPP

Name
my-mkr-wan

DevEUI
A8610A33344B7115

Activation Mode
Already Registered

Tracking

Location
This device doesn't move

C. Leonardo Da Vinci, 14014, Córdoba, España

Add device

Ilustración 82 - Añadir dispositivo a Cayenne (2)

Una vez añadido, al haber utilizado la codificación CayenneLPP, la información se actualizará automáticamente con cada mensaje *uplink* enviado desde el dispositivo.

2.1.5. Añadir dispositivo a aplicación móvil

Al llegar a TTN un mensaje enviado por algún dispositivo, se incluye en una petición HTTP POST y se envía mediante el *webhook* configurado hasta la aplicación web implementada en Apps Scripts, además de a Cayenne. Para que esa información se incluya en la hoja de cálculo que utiliza la app como fuente de datos, se debe procesar el mensaje recibido e insertar la información en las celdas correspondientes. Para ello, se deben añadir las siguientes líneas:

The screenshot shows the Google Apps Script interface with the following details:

- Title:** update-parking-data-log
- Toolbars:** Implementar, ? (Help), and a user profile icon.
- Sidebar:** Shows sections for Archivos, Código.gs (selected), Bibliotecas, and Servicios.
- Code Area:** The script code is displayed:

```
1 var SHEET_NAME = "last_data"; // Hoja de cálculo
2 var DEV_EUI_1 = "A8610A3334B7115"; // Dispositivo plaza 1
3 var DEV_EUI_2 = "A8610A33343E7215"; // Dispositivo plaza 2
4 var N_DEVICES = 10; // N° de dispositivos utilizados (2 reales y 8 ficticios)
5 var DEV_EUI_3 = "XXXXXXXXXXXXXX";
```
- Numbered Callouts:** A red circle highlights line 1 of the code.

Ilustración 83 - Añadir dispositivo a app (1)



Archivos A-Z + ⌂ ⌂ Ejecutar Depuración doPost ▾ Registro de ejecución |

○ Código.gs

Bibliotecas +

Servicios +

```
26 /* -----
27     getDeviceValues(): Función que obtiene determinada información del mensaje recibido
28         Parámetros:
29             myData - mensaje JSON incluido en la petición POST recibida
30 -----
31     function getDeviceValues(myData){
32         var dev_eui      = myData.end_device_ids.dev_eui;           // DEV_EUI
33         if (dev_eui == DEV_EUI_1) { // Mensaje enviado por el dispositivo 1
34             var device      = "Plaza 1";                                // ID Plaza
35         }
36         else if (dev_eui == DEV_EUI_2) { // Mensaje enviado por el dispositivo 2
37             var device      = "Plaza 2";
38         }
39         else if (dev_eui == DEV_EUI_3) { // Mensaje enviado por el dispositivo 3
40             var device      = "Plaza 3";
41         }
42         var modFree      = myData.uplink_message.decoded_payload.analog_in_1; // Módulo referencia
43         var modNew       = myData.uplink_message.decoded_payload.analog_in_2; // Módulo actual
44         var angle        = myData.uplink_message.decoded_payload.analog_in_4; // Ángulo
45         var status       = myData.uplink_message.decoded_payload.digital_in_5; // Estado plaza
46         var latitude    = myData.uplink_message.locations.user.latitude; // Latitud (ubicación)
47         var longitude   = myData.uplink_message.locations.user.longitude; // Longitud (ubicación)
48         var address     = latitude + ', ' + longitude; // Ubicación
49
50         return { dev_eui, device, modFree, modNew, angle, status, latitude, longitude, address };
51     }
52 }
```

Ilustración 84 - Añadir dispositivo a app (2)

Archivos A-Z + ⌂ ⌂ Ejecutar Depuración doPost ▾ Registro de ejecución |

○ Código.gs

Bibliotecas +

Servicios +

```
57 /**
58     updateDeviceValues(): Función que actualiza la hoja de cálculo con la información del
59         mensaje recibido.
60         Parámetros:
61             deviceValues - determinados datos extraídos del mensaje
62 -----
63     function updateDeviceValues(deviceValues){
64         // HOJAS DE CÁLCULO UTILIZADAS
65         var sheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName(SHEET_NAME); // Última actualización
66
67         // VALORES A INSERTAR LOG
68         // time | dev_eui | device | modFree | modNew | angle | status | address | temperature | humidity | wind | description
69         var time          = Date();
70         var weatherValues = getWeather(deviceValues.latitude, deviceValues.longitude);
71
72         var info = [time,deviceValues.dev_eui,deviceValues.device,deviceValues.modFree, deviceValues.modNew,
73                     deviceValues.angle,deviceValues.status,deviceValues.address, weatherValues.temperature, weatherValues.humidity,
74                     weatherValues.wind, weatherValues.description];
75
76         if (deviceValues.dev_eui == DEV_EUI_1) var row = 2;           // Fila del dispositivo 1
77         else if (deviceValues.dev_eui == DEV_EUI_2) var row = 3; // Fila del dispositivo 2
78         else if (deviceValues.dev_eui == DEV_EUI_3) var row = 4; // Fila del dispositivo 3
79
80         for (let i = 0; i < info.length; i++) {
81             sheet.getRange(row, i + 1).setValue(info[i]);           // Actualizar fila
82         }
83
84         SpreadsheetApp.flush();                                     // Aplicar cambios
85     }
86 }
```

Ilustración 85 - Añadir dispositivo a app (3)



```
Archivos A-Z + ⌛ ⌚ | Ejecutar Depuración doPost ▾ | Registro de ejecución |  
Código.gs  
Bibliotecas +  
Servicios +  
87 /* UBICACIONES DISPOSITIVOS: SON NECESARIAS PARA ACTUALIZAR EL TIEMPO CADA 15 MIN  
88 - DEV_1 : 37.914765, -4.724311 - DEV_6 : 37.915811, -4.721464  
89 - DEV_2 : 37.914823, -4.725539 - DEV_7 : 37.915698, -4.719562  
90 - DEV_3 : XXXXXXXX, YYYYYYYY - DEV_8 : 37.916005, -4.720535  
91 - DEV_4 : 37.914337, -4.722720 - DEV_9 : 37.916053, -4.721401  
92 - DEV_5 : 37.913834, -4.722756 - DEV_10 : 37.913326, -4.719191  
93  
94 ORDEN EN VECTORES: DEV_1, DEV_2, DEV_3, DEV_4, DEV_5, DEV_6, DEV_7, DEV_8, DEV_9, DEV_10  
95 */  
96 var latitudeVector = [37.914765, 37.914823, XXXXXXXX, 37.914337, 37.913834, 37.915811, 37.  
915698, 37.916005, 37.916053, 37.913326];  
97 var longitudeVector = [-4.724311, -4.725539, YYYYYYYY, -4.722720, -4.722756, -4.721464, -4.  
719562, -4.720535, -4.721401, -4.719191];  
98
```

Ilustración 86 - Añadir dispositivo a app (4)

2.2. Eliminar dispositivo del sistema

Para eliminar un dispositivo del sistema de forma definitiva se deben seguir los siguientes pasos:

- (1) Eliminar dispositivo de App
- (2) Eliminar dispositivo de Cayenne
- (3) Eliminar dispositivo de TTN
- (4) Desinstalar dispositivo de la plaza de aparcamiento

2.2.1. Eliminar dispositivo de aplicación móvil

El proceso para eliminar el dispositivo de la app consta solo de dos partes:

- 1) Eliminar las líneas de código de la aplicación web que procesaban su información (véase **Añadir dispositivo a aplicación móvil**).
- 2) Eliminar la fila correspondiente de la hoja de cálculo que actúa como fuente de información para la app (**Ilustración 56**).



2.2.2. Eliminar dispositivo de Cayenne

Para eliminar un dispositivo de la plataforma de monitorización Cayenne, hay que situarse en el dashboard individual del dispositivo correspondiente y acceder a su configuración:

The screenshot shows the Cayenne dashboard interface. On the left, there's a sidebar with a dropdown menu set to 'my-mkr-wan' (marked with a red circle '1'). The main area is the 'Overview' tab, which includes a map showing the location of the device (marked with a blue pin) and various sensor readings in a grid format. In the top right corner of the main panel, there's a 'Configure' button (marked with a red circle '2').

Ilustración 87 - Eliminar dispositivo de Cayenne (1)

The screenshot shows the Cayenne dashboard interface, specifically the 'General' settings page for the device 'my-mkr-wan'. The 'Device Name' is listed as 'my-mkr-wan'. Under 'Activation Mode', it says 'Already registered'. At the bottom, there's a red button labeled 'Remove Device' (marked with a red circle '3'), and a message below it stating 'This action cannot be undone'.

Ilustración 88 - Eliminar dispositivo de Cayenne (2)



2.2.3. Eliminar dispositivo de TTN

Al igual que en el caso de Cayenne, para eliminar un dispositivo de TTN hay que acceder a su configuración desde la consola:

The screenshot shows the 'Applications' section of the The Things Stack interface. A red box labeled '1' highlights the 'Applications' tab. A red box labeled '2' highlights the 'End devices' link in the left sidebar. A red box labeled '3' highlights the device 'my-mkr-wan' in the list.

ID	Name	DevEUI	JoinEUI	Last activity
second-mkr	MKR 2	A8 61 0A 33 34 3...	00 00 00 00 00 0...	Aug 16, 2022 •
my-mkr-wan	My WANda	A8 61 0A 33 34 4...	00 00 00 00 00 0...	Aug 17, 2022 •

Ilustración 89 - Eliminar dispositivo de TTN (1)

The screenshot shows the 'General settings' tab for the device 'My WANda'. A red box labeled '4' highlights the 'General settings' tab in the top navigation bar. The 'Basic' section is expanded, showing fields for 'End device ID*' (my-mkr-wan) and 'AppEUI' (00 00 00 00 00 00 00 00).

Ilustración 90 - Eliminar dispositivo de TTN (2)



The screenshot shows the 'End devices' configuration page in the TTN web interface. On the left, there's a sidebar with options like 'Proyecto TFG', 'Overview', 'End devices' (which is selected and highlighted in blue), 'Live data', 'Payload formatters', 'Integrations', 'Collaborators', 'API keys', and 'General settings'. The main panel has several input fields and sections: 'Optional end device description; can also be used to save notes about the end device' (empty), 'Network Server address' (eu1.cloud.thethings.network), 'Application Server address' (eu1.cloud.thethings.network), 'External Join Server' (checkbox 'Enabled' is unchecked), 'Join Server address' (eu1.cloud.thethings.network), and 'Attributes' (button '+ Add attributes'). Below these is a note: 'Attributes can be used to set arbitrary information about the entity, to be used by scripts, or simply for your own organization'. At the bottom are two buttons: 'Save changes' (blue) and 'Delete end device' (red, circled with a red border and the number 5).

Ilustración 91 - Eliminar dispositivo de TTN (3)

2.2.4. Desinstalar dispositivo de la plaza de aparcamiento

En primer lugar, para evitar desperfectos en el equipo electrónico, se debe desatornillar la parte superior de la caja, extraer el dispositivo de detección y desconectar la batería.

Posteriormente, las acciones a acometer dependerán de la opción de instalación escogida:

- A) **Sobre la superficie:** primeramente, se deben desatornillar los elementos de protección. Para quitar completamente el dispositivo, es necesario usar un martillo y un cincel para destruir el efecto adhesivo al cincelar la base de la caja paralelamente a la superficie del espacio de estacionamiento.
- B) **Semienterrado:** en este caso se debe utilizar un martillo y un cincel para romper el cemento y extraer la caja. Finalmente, se debe rellenar con cemento u hormigón el agujero resultante para devolver la zona de aparcamiento a su estado inicial.



2.3. Enviar mensaje *downlink* a dispositivo

Como se ha visto en la **Inicialización del dispositivo**, enviar un mensaje *downlink* a un dispositivo es un proceso muy sencillo. Por ejemplo, si se quisiera indicar al dispositivo un cambio en el período de envío para que lo realice cada 5 minutos, el mensaje a enviar contendría la cadena “4-1-5”, que convertida a HEX sería: 34 2D 31 2D 35

The screenshot shows the THE THINGS STACK Community Edition interface. The top navigation bar includes 'THE THINGS NETWORK' icon, 'THE THINGS STACK Community Edition' logo, 'Overview' (highlighted with a red box), 'Applications' (highlighted with a red box and circled with a red number 1), 'Gateways', 'Organizations', 'EU1 Community' (with 'No SLA applicable' note), and a user profile icon.

The left sidebar menu has items: 'Projecto TFG' (selected), 'Overview' (highlighted with a red box and circled with a red number 2), 'End devices' (highlighted with a red box and circled with a red number 2), 'Live data', 'Payload formatters', 'Integrations', 'Collaborators', 'API keys', and 'General settings'.

The main content area shows the 'Applications > Proyecto TFG > End devices' view. It displays a table of end devices:

ID	Name	DevEUI	JoinEUI	Last activity
second-mkr	MKR 2	A8 61 0A 33 34 3...	00 00 00 00 0...	Aug 16, 2022 •
my-mkr-wan	My WANda	A8 61 0A 33 34 4...	00 00 00 00 0...	Aug 17, 2022 •

Ilustración 92 - Enviar mensaje *downlink* (1)



Ilustración 93 - Enviar mensaje downlink (2)

3. Manual de la Plataforma de Monitorización

Previamente se ha mostrado cómo **Añadir dispositivo a plataforma de monitorización Cayenne**. A continuación, se muestra cómo manejar el dashboard general, acceder al histórico de datos y crear gráficas para analizar esos datos.

3.1. Dashboard general del proyecto

Para visualizar información de varios dispositivos a la vez, se deben anclar los widgets deseados en el dashboard general del proyecto. Para ello, basta con arrastrarlos desde el desplegable del dispositivo correspondiente:

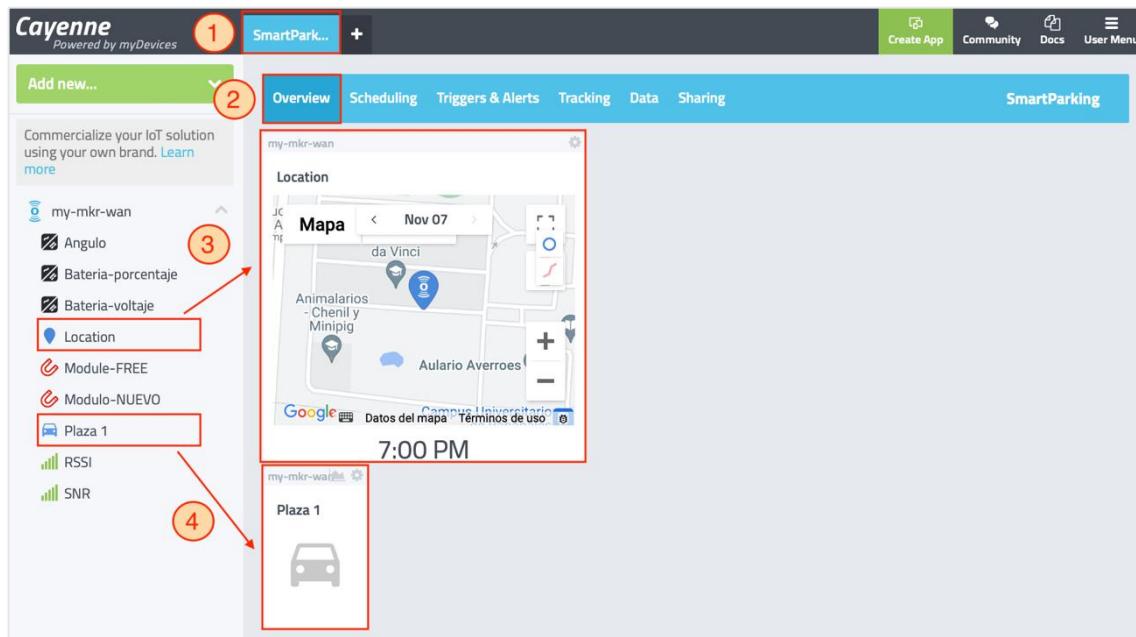


Ilustración 94 - Configurar dashboard general del proyecto (Cayenne)

3.2. Histórico de datos

Desde el mismo dashboard general se puede acceder al histórico de datos mediante el menú “Data”, pudiendo filtrar la información por variables recibidas, como el estado de la plaza. Además, es posible descargar el conjunto de datos en un fichero csv mediante la opción “Download”.

Timestamp	Device	Channel	Sensor Name	Sensor ID	Data Type	Unit	Values
2022-08-16 9:33:40	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	1
2022-08-16 9:32:40	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	1
2022-08-16 9:31:40	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	1
2022-08-16 9:30:40	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	1
2022-08-16 9:29:39	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	1
2022-08-16 9:28:41	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	1
2022-08-16 9:27:52	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	1
2022-08-16 9:26:52	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	1
2022-08-16 9:25:52	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	1
2022-08-16 9:24:52	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	1
2022-08-16 9:23:52	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	1
2022-08-16 9:22:51	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	1
2022-08-16 9:21:51	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	1
2022-08-16 9:20:51	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	1
2022-08-16 9:19:51	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	1
2022-08-16 9:18:51	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	1
2022-08-16 9:17:51	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	1
2022-08-16 9:16:51	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	1
2022-08-16 9:15:51	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	1
2022-08-16 9:14:51	my-mkr-wan	5	Plaza 1	51007570-1a6f-11ed-baf...	digital_sen...	d	1

Ilustración 95 - Acceder al histórico de datos (Cayenne)



3.3. Análisis gráfico

Para analizar gráficamente los valores históricos de una variable, por ejemplo, el estado de una plaza o la variación del campo magnético leído por un dispositivo, se puede escoger esa opción desde el dashboard individual del dispositivo:

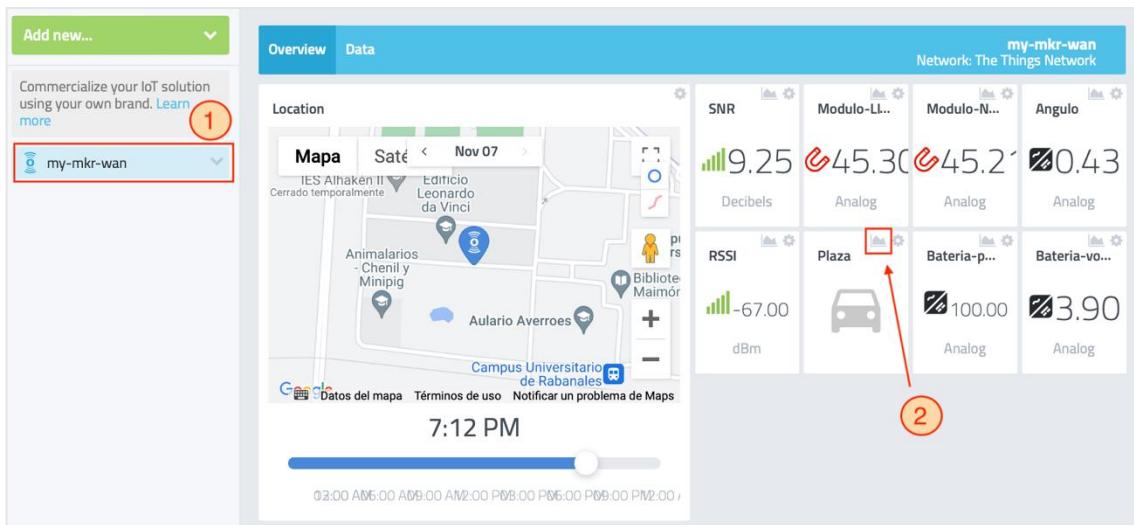


Ilustración 96 - Realizar análisis gráfico en Cayenne (1)

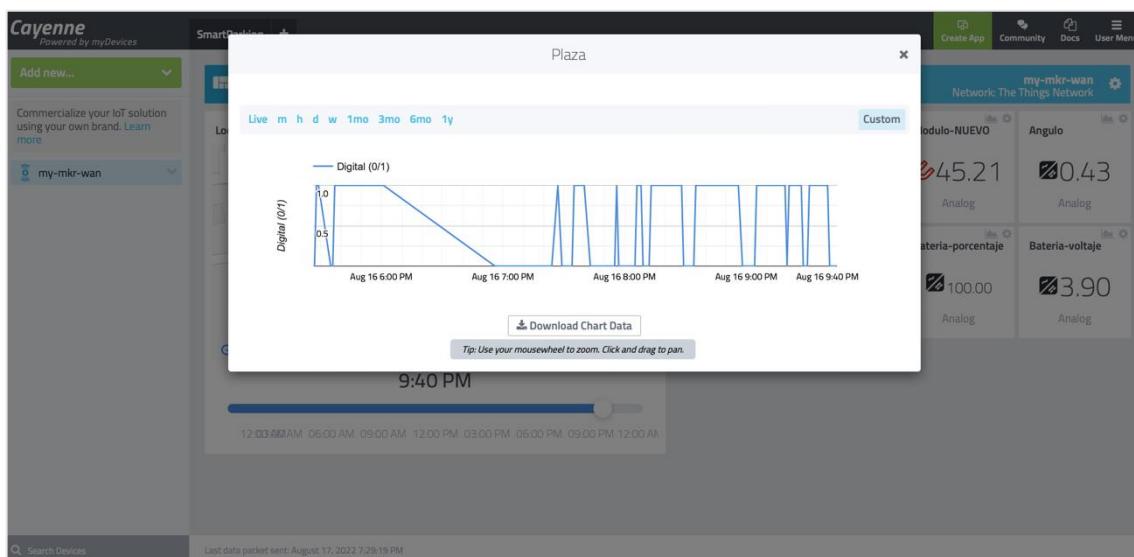


Ilustración 97 - Realizar análisis gráfico en Cayenne (2)



4. Manual de la App

4.1. Compartir la App

Como se comentó en la implementación de la **Aplicación móvil**, y al tratarse de un prototipo de app, no se ha publicado como *white-label*. Por eso, actualmente solo pueden acceder a la app aquellas personas con las que se comparta. Para ello, se debe acceder al editor de AppSheet:

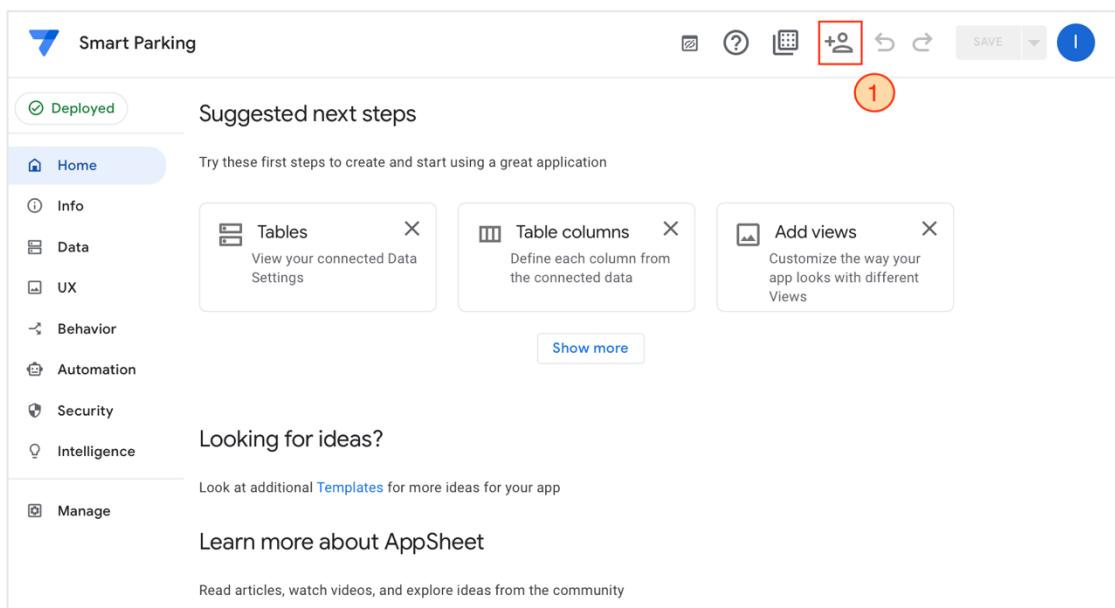


Ilustración 98 - Compartir app (1)

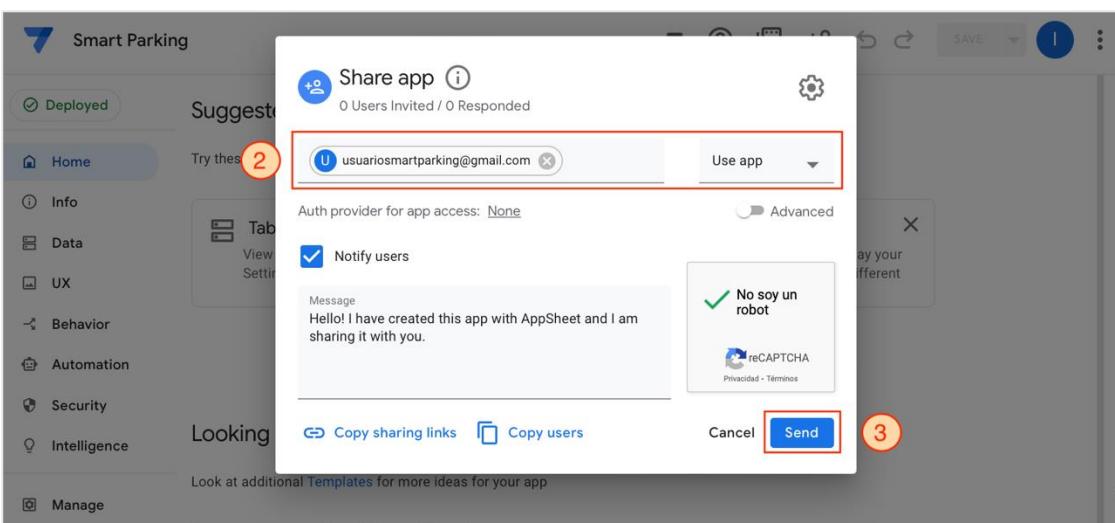


Ilustración 99 - Compartir app (2)

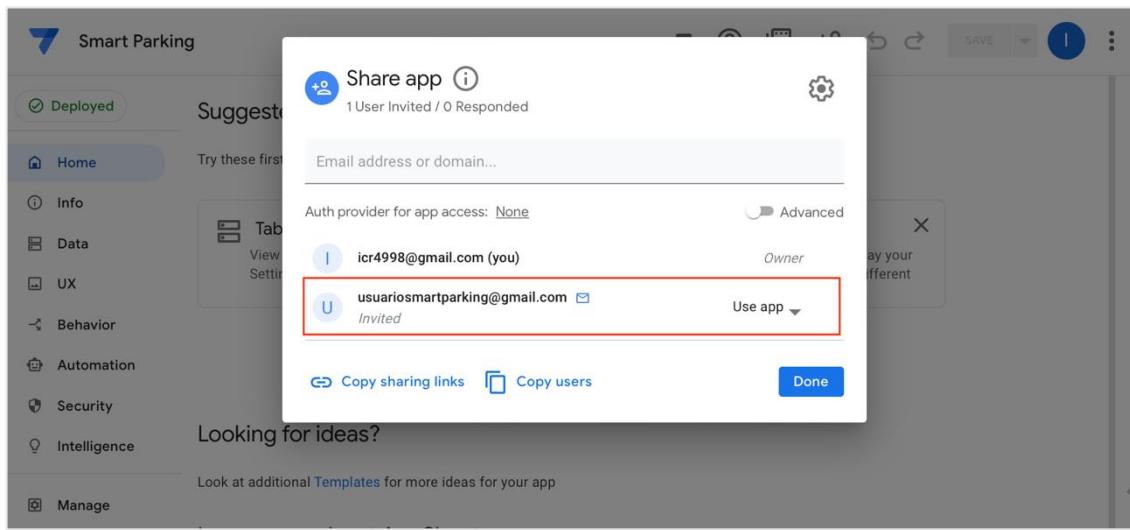


Ilustración 100 - Compartir app (3)

4.2. Instalar y acceder a la App

Los usuarios con los que se comparta la app recibirán un correo electrónico dándole dos opciones: instalar la app o abrirla en el navegador. Al seleccionar la opción de instalar, indica que es necesario instalar la aplicación de AppSheet:

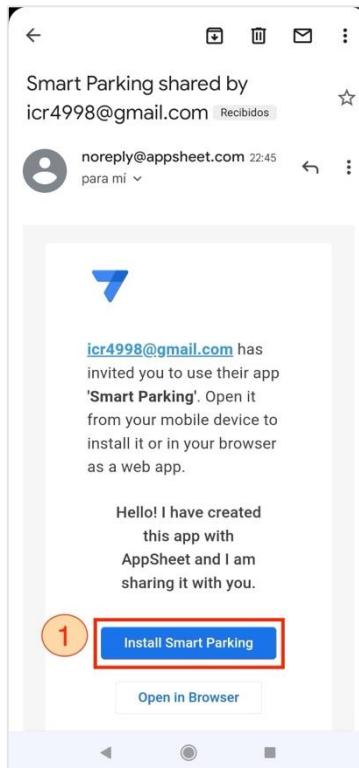


Ilustración 101 - Instalar app (1)

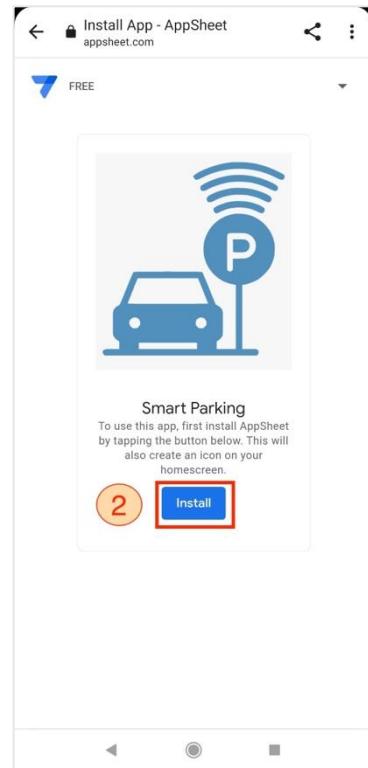


Ilustración 102 - Instalar app (2)



Una vez instalada la app de AppSheet, se debe iniciar sesión. Entonces, la app **Smart Parking** se encontrará entre las “*Compartidas conmigo*”:

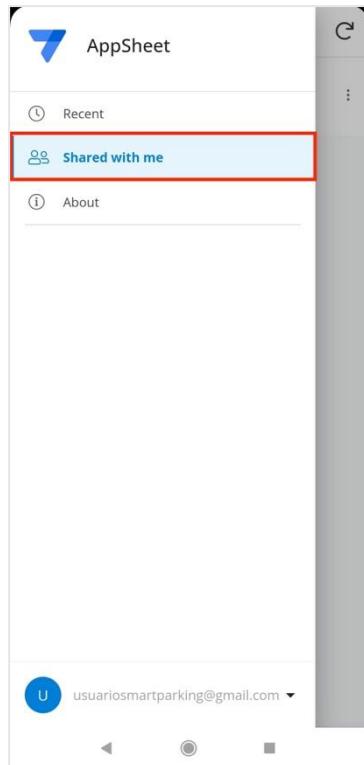


Ilustración 103 - Acceso a app (1)

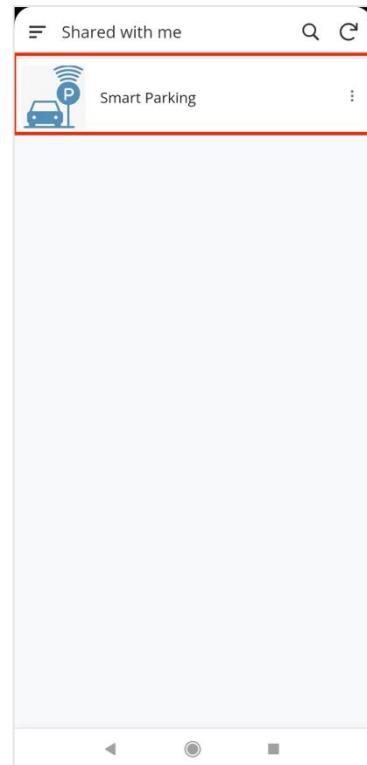


Ilustración 104 - Acceso a app (2)

Al instalar la app **Smart Parking** a través de AppSheet, se crea un acceso directo, además del propio de AppSheet. Se puede utilizar cualquiera de las dos opciones para acceder a la aplicación móvil desarrollada.

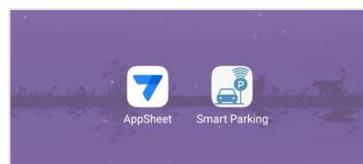


Ilustración 105 - Acceso directo a app



4.3. Navegar por la App

4.3.1. Pantalla principal: Mapa

Al acceder a la aplicación *Smart Parking* aparece de forma predeterminada la vista de “Mapa” (**Ilustración 106**), donde se pueden seleccionar las plazas para conocer parte de su información (**Ilustración 107**).

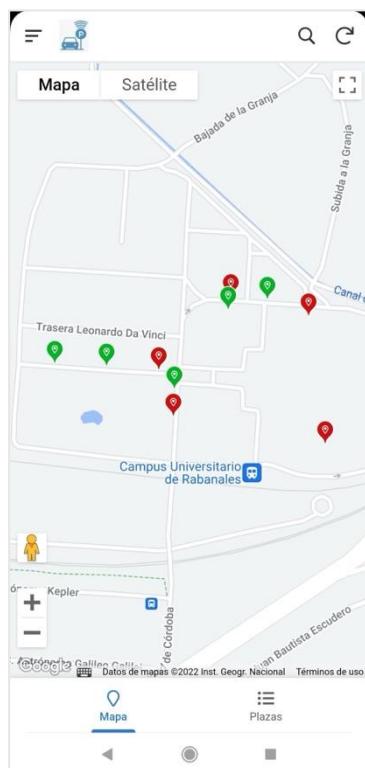


Ilustración 106 - Pantalla principal: Mapa

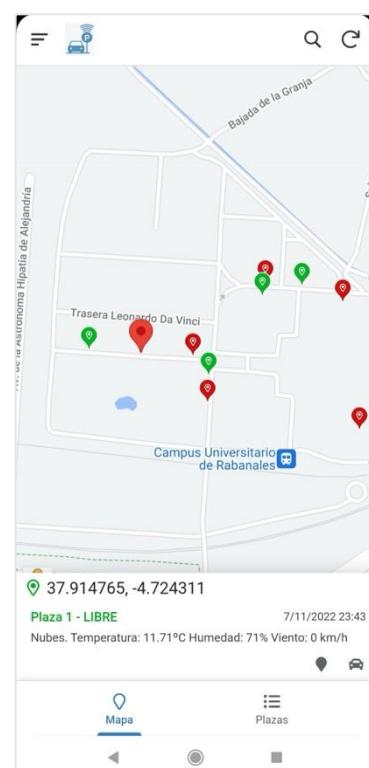


Ilustración 107 - Acceso a información desde Mapa



4.3.2. Pantalla secundaria: Plazas

En la vista de “Plazas” se muestra un listado de las plazas del sistema y su ocupación (**Ilustración 108**). Desde esta vista se puede acceder a la información completa de cada plaza para, por ejemplo, consultar la información del tiempo más reciente (**Ilustración 109**).

Plaza 1 - LIBRE	22/10/2022 17:58
Cielo claro. Temperatura: 24.71°C Humedad: 65% Viento: 11.12...	
Plaza 2 - LIBRE	22/10/2022 17:58
Cielo claro. Temperatura: 24.71°C Humedad: 65% Viento: 11.12...	
Plaza 3 - OCUPADO	22/10/2022 17:58
Cielo claro. Temperatura: 24.71°C Humedad: 65% Viento: 11.12...	
Plaza 4 - LIBRE	22/10/2022 17:58
Cielo claro. Temperatura: 24.71°C Humedad: 65% Viento: 11.12...	
Plaza 5 - OCUPADO	22/10/2022 17:58
Cielo claro. Temperatura: 24.71°C Humedad: 65% Viento: 11.12...	
Plaza 6 - LIBRE	22/10/2022 17:58
Cielo claro. Temperatura: 24.68°C Humedad: 65% Viento: 11.12...	
Plaza 7 - OCUPADO	22/10/2022 17:58
Cielo claro. Temperatura: 24.68°C Humedad: 65% Viento: 11.12...	
Plaza 8 - LIBRE	22/10/2022 17:58
Mapa	Plazas

Ilustración 108 - Pantalla secundaria: Plazas

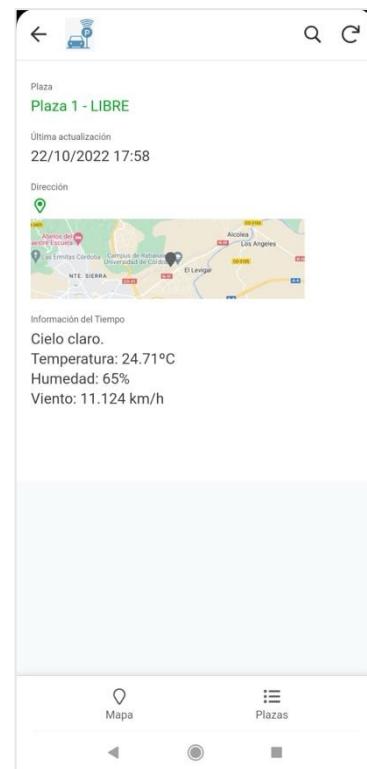


Ilustración 109 - Acceso a información desde vista
Plazas



4.3.3. Obtener ruta hasta la plaza

Como se ha comentado, AppSheet es una herramienta de Google, por lo que incluye la integración con Google Maps. Gracias a esto, se puede enlazar con la aplicación de Google Maps para descubrir la ruta hasta la plaza. Para ello, se debe seleccionar el símbolo de “coche” que aparece en la zona de información de una plaza determinada.

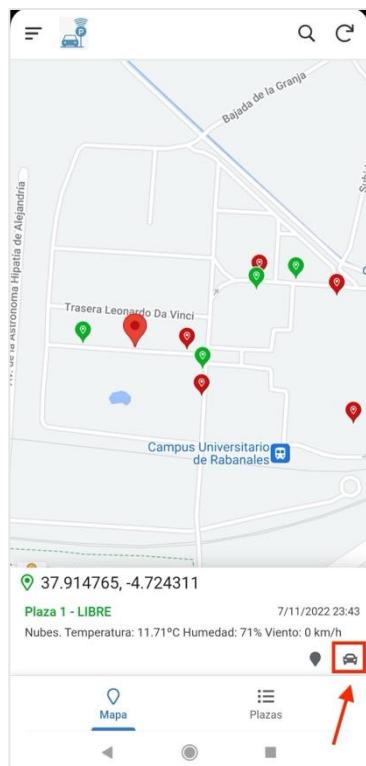


Ilustración 110 - Obtener ruta hasta la plaza



4.3.4. Filtrar resultados

Para filtrar los resultados basta con buscar la palabra clave deseada, (p. ej. “LIBRE”) desde la vista de “Mapa” (**Ilustración 111**) o la del listado de “Plazas” (**Ilustración 112**).

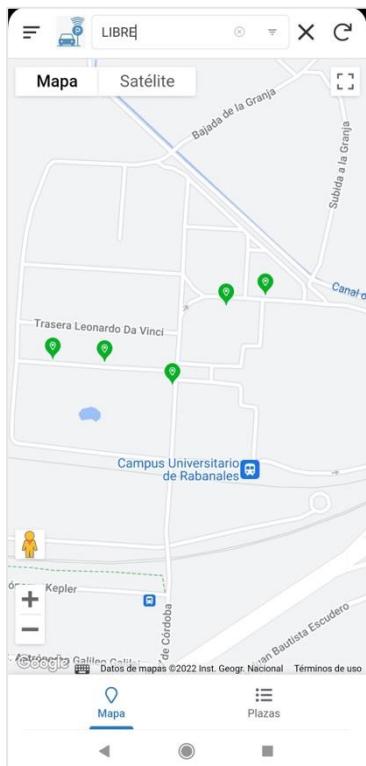


Ilustración 111 - Filtrar en Mapa

Plaza 1 - LIBRE	22/10/2022 17:58
Cielo claro. Temperatura: 24.71°C Humedad: 65% Viento: 11.12...	...
Plaza 2 - LIBRE	22/10/2022 17:58
Cielo claro. Temperatura: 24.71°C Humedad: 65% Viento: 11.12...	...
Plaza 4 - LIBRE	22/10/2022 17:58
Cielo claro. Temperatura: 24.71°C Humedad: 65% Viento: 11.12...	...
Plaza 6 - LIBRE	22/10/2022 17:58
Cielo claro. Temperatura: 24.68°C Humedad: 65% Viento: 11.12...	...
Plaza 8 - LIBRE	22/10/2022 17:58
Cielo claro. Temperatura: 24.68°C Humedad: 65% Viento: 11.12...	...

Ilustración 112 - Filtrar en Plazas



B. ANEXO II – MANUAL DE CÓDIGO





1. Aplicación del dispositivo inalámbrico

1.1. Cuerpo principal – MAIN.ino

```
1  /*----- LIBRERÍAS -----*/
2  #include <Adafruit_HMC5883_U.h>          // Magnetómetro
3  #include <Adafruit_Sensor.h>                // Tipo de sensor
4  #include <MKR WAN.h>                      // Modem LoRa y envío/recepción mensajes
5  #include <CayenneLPP.h>                    // Cayenne myDevices ==> formato mensajes
6  #include <Wire.h>                         // sleep y reducir consumo
7  #include <ArduinoLowPower.h>               // sleep y reducir consumo
8
9
10 /*----- FICHEROS AUXILIARES -----*/
11 #include "VECTOR.h"                        // Operaciones vectoriales
12 #include "FUNCTIONS.h"                     // Funciones auxiliares
13 #include "arduino_secrets.h"               // Claves aplicación en TTN
14
15 /*----- CLAVES TTN -----*/
16 String appEui = SECRET_APP_EUI;           // ID de la aplicación
17 String appKey = SECRET_APP_KEY;            // Clave de la aplicación para este dispositivo
18
19 /* OBJETOS UTILIZADOS: magnetómetro, modem LoRa y Cayenne (formato mensajes) */
20 Adafruit_HMC5883_Unified mag = Adafruit_HMC5883_Unified(12345); // Magnetometer con ID '12345'
21 LoRaModem modem;                         // Modem Lora para envío/recepción mensajes
22 CayenneLPP lpp(51);                     // Objeto CayenneLPP para codificación de mensajes
23
24 /* VARIABLES DE TIEMPO: Utilizadas para saber si es el momento de enviar el mensaje */
25 int minutes = 0;                          // Variable de tiempo para enviar cada 'minutes' minutos
26
27
28 void setup(){
29     Serial.begin(9600);
30
31     /*----- HMC5883L (MAGNETÓMETRO) -----*/
32     mag.begin();
33
34     /*----- CONEXIÓN LORAWAN -----*/
35     modem.begin(EU868);                  // Establecer frecuencia de la región europea
36     delay(1000);
37     modem.joinOTAA(appEui, appKey);    // Conexión a una aplicación de TTN mediante sus claves
38     modem.minPollInterval(60);
39 }
40
41
42
43 void loop(){
44
45     /*----- HMC5883L (MAGNETÓMETRO) -----*/
46     /*----- VALORES DE REFERENCIA -----*/
47     /*----- */
48     sensors_event_t event; // Nuevo evento del magnetómetro
49     mag.getEvent(&event); // Obtención de valores del campo magnético (uT)
50
51
52     /*----- VALORES ACTUALES -----*/
53     /*----- */
54     if(takeRef){ /* Esta flag indica si tomar valores de referencia.
55         | Puede manipularse mediante un mensaje DOWNLINK de tipo 1. */
56         vect_free = {event.magnetic.x, event.magnetic.y, event.magnetic.z, -100.00};
57         getModule(vect_free);
58         takeRef = false; /* Reestablecer flag de referencia para no volver a tomar los valores
59         | hasta que se indique */
60     }
61
62
63     vect_actual = {event.magnetic.x, event.magnetic.y, event.magnetic.z, -100.00};
64     getModule(vect_actual);
65
66     Serial.print("\n\n Intensidad de referencia: "); Serial.print(vect_free.module); Serial.print(" uT");
67     Serial.print("\n\n Intensidad actual: "); Serial.print(vect_actual.module); Serial.print(" uT");
68     Serial.println("\n");
69
70     /*----- ÁNGULO ENTRE VECTORES -----*/
71     /*----- */
72     /* Tomar ángulo formado entre el vector de referencia y el actual */
73     angleVectors = getAngleVectors(vect_free, vect_actual);
74     Serial.print("\nÁngulo entre vectores: "); Serial.println(angleVectors);
75
76
77
78
79
80
```



```
81      /*
82      *----- ACTUALIZACIÓN ESTADO PLAZA -----*/
83      /*
84      */
85      /*
86      * A partir de los valores de referencia y los umbrales establecidos,
87      * se comparan con los valores actuales. Si se encuentran dentro de
88      * los umbrales, es decir, si se cumple isOccupied(), significa que
89      * la plaza está ocupada.
90      * Estos umbrales se han establecido universalmente de tal manera
91      * que no importe la forma en que diferentes vehículos distorsionen el
92      * campo magnético.
93      *
94      * Además, solo se cambia el estado de la plaza cuando es diferente
95      * al estado anterior.
96      *
97      */
98      if(isOccupied() and (statusCar == 0)){ // Si está ocupado, pero estaba libre
99          statusCar = 1;
100         changedStatus = true; // Se ha cambiado el estado: libre --> ocupado
101     }
102     else if(!isOccupied() and (statusCar == 1)){ // Si está libre, pero estaba ocupado
103         statusCar = 0;
104         changedStatus = true; // Se ha cambiado el estado: ocupado --> libre
105     }
106     Serial.print("\nEstado plaza (0: libre    1: ocupado): "); Serial.println(statusCar);
107
108
109
110
111     /*
112     *----- ENVÍO DEL MENSAJE -----*/
113     /*
114     // Para ahorrar energía, solo se da formato al mensaje cuando se vaya a realizar el envío
115     if(atChange and changedStatus){ /* Si se quiere enviar solo al cambiar estado
116         y ha habido cambio */
117         getCayenneFormat();
118         sendMsg();
119         changedStatus = false; // Se reestablece la flag del cambio de estado
120     }
121     else if(!atChange){ /* Si no se quiere enviar mensaje al cambiar de estado,
122         sino cada 'nMin' minutos.
123             La situación del campo magnético se lee cada
124             minuto, pero se envía según indique esta flag. */
125         if(minutes != nMin) minutes++;
126         else{
127             getCayenneFormat();
128             sendMsg();
129             minutes = 0;
130         }
131     }
132
133
134     /*
135     *----- RECEPCIÓN DEL MENSAJE -----*/
136     /*
137     receiveMsg();
138
139
140     LowPower.sleep(interval); //Se obtiene vector magnético cada minuto, pero se envía cuando se indique
141
142 } // Fin función 'loop()'
```



```
147
148
149
150 /*
151 |   getCayenneFormat(): Función para codificar con CayenneLPP el mensaje a enviar.
152 -----*/
153 void getcayenneFormat(){
154     lpp.reset(); // Se resetea el objeto CayenneLPP
155     /* Se añade al mensaje como valor digital:
156      *   - Estado de la placa (0: libre 1: ocupado)
157      *   Por otro lado, si se han solicitado, se añaden como valores analógicos:
158      *   - Voltaje de la batería
159      *   - Porcentaje remanente de batería
160    */
161    /* Estos valores se enviaban en las pruebas. En la versión final solo debe enviar el estado.
162       lpp.addAnalogInput(1, vect_free.module);
163       lpp.addAnalogInput(2, vect_actual.module);
164       lpp.addAnalogInput(3, angleVectors);
165    */
166    lpp.addDigitalInput(4, statusCar);
167    if(sendBattery){ /* Si se ha pedido el estado de la batería mediante un mensaje
168                      DOWNLINK de tipo 3, se añade al mensaje UPLINK a enviar. */
169        statusBattery(voltage, batteryPercentage);
170        lpp.addAnalogInput(5, voltage);
171        lpp.addAnalogInput(6, batteryPercentage);
172        sendBattery = false; // Se reestablece la flag que pedía el estado de la batería
173    }
174 }
175
176
177
178 /*
179 |   sendMsg(): Función para enviar el mensaje previamente codificado con CayenneLPP.
180 -----*/
181 void sendMsg(){
182     int err;
183     modem.beginPacket();
184     modem.write(lpp.getBuffer(), lpp.getSize()); // Enviar mensaje UPLINK codificado en formato CayenneLPP
185     err = modem.endPacket(true);
186     if (err > 0) {
187         Serial.println("\nMessage sent correctly!");
188     } else{
189         Serial.println("\nError sending message :(\n");
190     }
191 }
192
193
194
195
196 /*
197 |   receiveMsg(): Función en la que se comprueba si se ha recibido mensaje DOWNLINK y, si es así, se
198 |   procesa su información. Posteriormente, en getInfoMsg(), se actúa acorde a los datos
199 |   recibidos.
200 -----*/
201 void receiveMsg(){
202     if (modem.available()) { //Si se ha recibido mensaje DOWNLINK
203         /* Se lee la cadena recibida */
204         char mesgRcv[64]; //Hasta 64 caracteres recibidos
205         int i = 0;
206         while (modem.available()) {
207             mesgRcv[i++] = (char)modem.read();
208         }
209         String strRcv = String(mesgRcv);
210         strRcv.trim();
211         Serial.print("Received: ");
212         Serial.print(strRcv);
213
214         //***** PROCESAMIENTO DEL MENSAJE DOWNLINK *****
215         /* Las siguientes líneas procesan el mensaje DOWNLINK recibido:
216          *   Siguiendo el protocolo establecido, este mensaje debe enviar una cadena de valores separados
217          *   por un guión '-'. El primer valor es el tipo de mensaje y los demás, si se incluyen, dependen
218          *   de ese tipo de mensaje.
219         */
220         String val[10]; // Se admiten hasta 10 valores
221         int nVal = 0, index;
222         while (strRcv.length() > 0){
223             index = strRcv.indexOf('-');
224             if (index == -1){
225                 val[nVal++] = strRcv;
226                 break;
227             }
228             else{
229                 val[nVal++] = strRcv.substring(0, index);
230                 strRcv = strRcv.substring(index+1);
231             }
232         }
233
234         // Tras procesar el mensaje DOWNLINK, el vector 'val[]' contiene los valores recibidos.
235         getInfoMsg(val); // Acciones según tipo de mensaje
236     }
237 }
```



1.2. Funciones auxiliares – FUNCTIONS.h

```

1 #ifndef FUNCTIONS_h
2 #define FUNCTIONS_h
3
4 #include "VECTOR.h" // Operaciones vectoriales
5
6 /* BATERÍA */
7 #define ADC_BATTERY (32u)
8 float battery_voltage = 3.9;
9 bool sendBattery = false; // Flag que indica la petición del estado de batería
10 float voltage;
11 int batteryPercentage;
12
13 /*
14 * FLAG PARA TOMAR VARIABLES DE REFERENCIA
15 */
16 bool takeRef = true; // TRUE ==> tomar valores de referencia (plaza libre)
17 // FALSE ==> no tomar valores de referencia
18
19 /*
20 * VARIABLES DEL ESTADO DE LA PLAZA
21 */
22 int statusCar; // 0 ==> libre
23 // 1 ==> ocupado
24
25 bool changedStatus; // TRUE ==> ha habido cambio de estado
26 // FALSE ==> no ha habido cambio de estado
27
28 /*
29 * FLAG PARA ENVIAR MENSAJE AL CAMBIAR DE ESTADO
30 */
31 bool atChange = true; // TRUE ==> enviar solo al cambiar el estado
32 // FALSE ==> enviar cada 'nMin' minutos
33
34 /*
35 * VARIABLES UMBRALES DE COMPROBACIÓN DE ESTADO
36 */
37 float diffModule = 8.82; // Umbral módulo (uT)
38 float gradesVectors = 4.65; // Umbral ángulo entre vectores (grados)
39 float angleVectors; // Ángulo que forman el vector de referencia y el actual
40
41 /*
42 * VARIABLES PARA TIPO DE MENSAJE Y PERÍODO DE ENVÍO
43 */
44 int typeMsg; // Tipo de mensaje DOWNLINK
45 int nMin = 1; // Cada cuántos minutos enviar mensaje
46
47
48
49
50
51 *****
52 ----- FUNCIONES AUXILIARES -----
53 *****
54
55 /*
56 isOccupied(): Función para conocer el estado de la plaza.
57 | Si la variación del vector magnético actual respecto al de referencia es mayor a los
58 | umbrales establecidos, la plaza está ocupada.
59 */
60 bool isOccupied(){
61     if((abs(vect_free.module - vect_actual.module) >= diffModule) or (angleVectors >= gradesVectors))
62         return true;
63 }

```



```
66
67
68 /*
69     getInfoMsg(): Función para obtener la información del mensaje DOWNLINK según el tipo.
70     | Parámetro:
71     |     val[] - Vector con el mensaje procesado en receiveMsg()
72 */
73 /*
74 * | TIPO DE MENSAJE | ESTRUCTURA | SIGNIFICADO |
75 *
76 * | 1 | "1" | Tomar valores de referencia |
77 * | 2 | "2-diffModule-gradesVectors" | Modificar umbrales de comparación |
78 * | 3 | "3" | Enviar estado de batería |
79 * | 4 (opt 1) | "4-opcion-nMin" | Modificar período envío: cada 'nMin' minutos |
80 * | 4 (opt 2) | "4-opcion" | Modificar período envío: al cambiar estado plaza |
81 *
82 */
83 void getInfoMsg(String val[]){
84     typeMsg = val[0].toInt(); // Tipo de mensaje DOWNLINK recibido
85     switch (typeMsg){
86         case 1: libre = true; break;
87         case 2: diffModule = val[1].toFloat(); gradesVectors = val[2].toFloat(); break;
88         case 3: sendBattery = true; break;
89         case 4:
90             int option = val[1].toInt();
91             switch (option){
92                 case 1: atChange = false; nMin = val[2].toInt(); break;
93                 case 2: atChange = true; break;
94             }
95     }
96 }
97
98
99
100
101
102 /*
103     statusBattery(): Función para tomar el estado de la batería
104     Parámetros:
105     | voltage | - voltaje actual de la batería
106     | batteryPercentage | - porcentaje actual de la batería
107 */
108 void statusBattery(float &voltage, int &batteryPercentage){
109     analogReadResolution(10);
110     analogReference(AR_INTERNAL1V0);
111     int batteryLevel = analogRead(ADC_BATTERY);
112     voltage = batteryLevel * (battery_voltage / 1023.0);
113     batteryPercentage = ((voltage * 100) / battery_voltage);
114     Serial.print("Voltage: "); Serial.print(voltage);
115     Serial.print(" Batery Percentage: "); Serial.print(batteryPercentage); Serial.println("%\n");
116 }
117
118
119 #endif
120
121
```



1.3. Funciones operaciones vectoriales – VECTOR.h

```
1 #ifndef VECTOR_h
2 #define VECTOR_h
3
4 #include <Arduino.h>
5
6 /*-----*/
7 /* VALORES DEL VECTOR */
8 /*-----*/
9 struct myVector{
10     float x;           // Componente x
11     float y;           // Componente y
12     float z;           // Componente z
13     float module;      // Módulo del vector
14 }vect_free, vect_actual;
15
16
17 /****** FUNCIONES OPERACIONES VECTORIALES ******/
18
19 /*-----*/
20 /*-----*/
21
22 /*
23     getModule(): Función para obtener el módulo de un vector a partir de sus
24     componentes.
25     Parámetro:
26         | vect - objeto de tipo myVector que incluye las componentes
27 */
28 void getModule(myVector &vect){
29     vect.module = sqrt(pow(vect.x, 2) + pow(vect.y, 2) + pow(vect.z, 2));
30 }
31
32
33 /*
34     dotProduct(): Función para obtener el producto escalar entre dos vectores.
35     Parámetros:
36         | vect1 - myVector que representa al primer vector
37         | vect2 - myVector que representa al segundo vector
38 */
39 float dotProduct(myVector &vect1, myVector &vect2){
40     float product = vect1.x * vect2.x + vect1.y * vect2.y + vect1.z * vect2.z;
41     return product;
42 }
43
44
45 /*
46     getAngleVectors(): Función para obtener el ángulo formado entre dos vectores.
47     Parámetros:
48         | vect1 - myVector que representa al primer vector
49         | vect2 - myVector que representa al segundo vector
50 */
51 float getAngleVectors(myVector &vect1, myVector &vect2){
52     float aux = (dotProduct(vect1, vect2))/(vect1.module * vect2.module);
53     float angle = acos(aux) * 180.0 / PI;
54     return angle;
55 }
56
57
58#endif
59
```

1.4. Parámetros de conexión LoRaWAN – ARDUINO_SECRETS.h

```
1 #define SECRET_APP_EUI "0000000000000000" // ID de la aplicación
2
3 /* Cada dispositivo tiene una clave diferente para conectarse a la aplicación,
4    por lo que es necesario modificar la siguiente línea con la clave correspondiente
5    según el dispositivo. */
6 #define SECRET_APP_KEY "390B6574B5A317101C552788D67BFBE0" //MKR 1
7 //#define SECRET_APP_KEY "3D1D27AC539E9B6212624AD9652EF62C" //MKR 2
8
9
```



2. Aplicación web de Apps Script

2.1. Procesar mensaje de TTN y actualizar hoja de cálculo

```
1 var SHEET_NAME      = "last_data";           // Hoja de cálculo
2 var DEV_EUI_1        = "A8610A33344B7115";     // Dispositivo plaza 1
3 var DEV_EUI_2        = "A8610A33343E7215";     // Dispositivo plaza 2
4 var N_DEVICES        = 10;                   // Número de dispositivos utilizados (2 reales y 8 ficticios)
5
6
7
8 /*
9  * doPost(): Función que se activa cuando la aplicación web recibe una solicitud POST.
10 *           Obtiene la información del mensaje recibido en la solicitud POST y actualiza
11 *           los datos.
12 *           Parámetros:
13 *           e - parámetro de evento que contiene información de la solicitud
14 */
15 function doPost(e) {
16   var myData       = JSON.parse(e.postData.contents); // Mensaje enviado por TTN
17   var deviceValues = getDeviceValues(myData);          // Obtiene algunos datos del mensaje
18
19   updateDeviceValues(deviceValues);                    // Actualiza la información
20
21   return HtmlService.createHtmlOutput("post request received");
22 }
23
24
25
26 /*
27  * getDeviceValues(): Función que obtiene determinada información del mensaje recibido
28 *           Parámetros:
29 *           | myData - mensaje JSON incluido en la petición POST recibida
30 */
31 function getDeviceValues(myData){
32   var dev_eui      = myData.end_device_ids.dev_eui;           // DEV_EUI
33   if (dev_eui == DEV_EUI_1) { // Mensaje enviado por el dispositivo 1
34     var device      = "Plaza 1";                                // ID Plaza
35   }
36   else if (dev_eui == DEV_EUI_2) { // Mensaje enviado por el dispositivo 2
37     var device      = "Plaza 2";
38   }
39   var modFree      = myData.uplink_message.decoded_payload.analog_in_1; // Módulo referencia
40   var modNew       = myData.uplink_message.decoded_payload.analog_in_2; // Módulo actual
41   var angle        = myData.uplink_message.decoded_payload.analog_in_4; // Ángulo
42   var status       = myData.uplink_message.decoded_payload.digital_in_5; // Estado plaza
43   var latitude     = myData.uplink_message.locations.user.latitude;    // Latitud (ubicación)
44   var longitude    = myData.uplink_message.locations.user.longitude;   // Longitud (ubicación)
45   var address      = latitude + ', ' + longitude;                // Ubicación
46
47   return { dev_eui, device, modFree, modNew, angle, status, latitude, longitude, address };
48 }
49
50
51 /*
52  * updateDeviceValues(): Función que actualiza la hoja de cálculo con la información del
53 *                      mensaje recibido.
54 *                      Parámetros:
55 *                      | deviceValues - determinados datos extraídos del mensaje
56 */
57 function updateDeviceValues(deviceValues){
58   // HOJAS DE CÁLCULO UTILIZADAS
59   var sheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName(SHEET_NAME); // Última actualización
60
61   // VALORES A INSERTAR LOG
62   // time | dev_eui | device | modFree | modNew | angle | status | address | temperature | humidity | wind | description
63   var time      = Date();
64   var weatherValues = getWeather(deviceValues.latitude, deviceValues.longitude);
65
66   var info = [time,deviceValues.dev_eui,deviceValues.device,deviceValues.modFree, deviceValues.modNew,deviceValues.angle,
67   deviceValues.status,deviceValues.address, weatherValues.temperature, weatherValues.humidity, weatherValues.wind, weatherValues.description];
68
69   if (deviceValues.dev_eui == DEV_EUI_1) var row = 2;      // Fila del dispositivo 1
70   else if (deviceValues.dev_eui == DEV_EUI_2) var row = 3; // Fila del dispositivo 2
71   for (let i = 0; i < info.length; i++) {
72     sheet.getRange(row, i + 1).setValue(info[i]);         // Actualizar fila
73   }
74
75   SpreadsheetApp.flush();                                // Aplicar cambios
76 }
77
78 }
```



2.2. Obtener el tiempo meteorológico en una ubicación

```
82
83  /* UBICACIONES DISPOSITIVOS: SON NECESARIAS PARA ACTUALIZAR EL TIEMPO CADA 15 MIN
84  | - DEV_1 : 37.914765, -4.724311      - DEV_6 : 37.915811, -4.721464
85  | - DEV_2 : 37.914823, -4.725539      - DEV_7 : 37.915698, -4.719562
86  | - DEV_3 : 37.914705, -4.723086      - DEV_8 : 37.916005, -4.720535
87  | - DEV_4 : 37.914337, -4.722720      - DEV_9 : 37.916053, -4.721481
88  | - DEV_5 : 37.913834, -4.722756      - DEV_10 : 37.913326, -4.719191
89
90  ORDEN EN VECTORES: DEV_1, DEV_2, DEV_3, DEV_4, DEV_5, DEV_6, DEV_7, DEV_8, DEV_9, DEV_10
91 */
92 var latitudeVector      = [37.914765, 37.914823, 37.914705, 37.914337, 37.913834, 37.915811, 37.915698, 37.916005, 37.916053, 37.
93 913326];
94 var longitudeVector     = [-4.724311, -4.725539, -4.723086, -4.722720, -4.722756, -4.721464, -4.719562, -4.720535, -4.721481, -4.
95 719191];
96
97 /**
98  * -----
99  * getWeather(): Función que obtiene la información del tiempo en una ubicación determinada
100 |           a partir de la API de OpenWeatherMap.
101 |           Parámetros:
102 |               latitude - latitud de la ubicación
103 |               longitude - longitud de la ubicación
104 */
104 function getWeather(latitude, longitude){
105 // INFORMACIÓN PARA OBTENER EL TIEMPO:
106 /*  lat      = latitud
107  lon      = longitud
108  API key = 774dcf442041de7dbb0de2d18e7cfb96 (https://openweathermap.org/api)
109  lang     = es (idioma de la descripción)
110  units    = metric
111 */
112
113 // Obtener el tiempo actual
114 var url = "api.openweathermap.org/data/2.5/weather?lat=" +latitude+ "&lon=" +longitude+ "&appid=774dcf442041de7dbb0de2d18e7cfb96&
115 lang=es&units=metric";
116 var response = UrlFetchApp.fetch(url, {method: 'GET',headers: {'Content-Type': 'application/json','Accept': 'application/json'} });
117
118 var weatherData      = JSON.parse(response.getContentText()); // Información obtenida
119 var temperature      = weatherData.main.temp + "°C";
120 var humidity         = weatherData.main.humidity + "%";
121 var wind              = ((parseFloat(weatherData.wind.speed)*3600)/1000).toString() + " km/h";
122 // m/s ==> ((m/s)*3600)/1000 ==> km/h
123 var description       = (weatherData.weather[0].description.charAt(0).toUpperCase() + (weatherData.weather[0].description).slice
124 (1)); // Primera letra de la descripción a mayúscula
125
126 return { temperature, humidity, wind, description };
127 }
```

2.3. Actualizar el tiempo en todas las ubicaciones

```
131
132 /**
133  * -----
134  * updateExtraInfoAllDevices(): Función que actualiza la información del tiempo de todas
135  |           las plazas. Se activa cada 15 minutos mediante un trigger.
136 */
137 function updateExtraInfoAllDevices(){
138   var sheet = SpreadsheetApp.getActiveSpreadsheet().getSheetByName(SHEET_NAME); // Última actualización
139   for (let i = 0; i < N_DEVICES; i++) {
140     var weatherValues = getWeather(latitudeVector[i], longitudeVector[i]); //Vectores globales con coordenadas
141     var time = Date();
142     var row = i + 2; // Comienza por fila 2 (device 1)
143     // COLUMNAS A MODIFICAR:  time(1), temperature(9), humidity(10), wind(11), description(12)
144     sheet.getRange(row, 1).setValue(time);
145     sheet.getRange(row, 9).setValue(weatherValues.temperature);
146     sheet.getRange(row, 10).setValue(weatherValues.humidity);
147     sheet.getRange(row, 11).setValue(weatherValues.wind);
148     sheet.getRange(row, 12).setValue(weatherValues.description);
149   }
150   SpreadsheetApp.flush();
151 }
```



C. ANEXO III - PRUEBAS





1. Pruebas para verificar la metodología

1 - PRUEBAS PARA SABER SI LA METODOLOGÍA DE DETECCIÓN ES LA CORRECTA.								
(SE USAN UMBRALES DE TANTEO QUE SE REEMPLAZARÁN POR LOS OBTENIDOS EMPIRICAMENTE POSTERIORMENTE)								
Umbrales de tanteo: diffModule ≥ 2.0 y gradesVectors ≥ 3.0								
COMENTARIO	TIME	DEV_EUI	TAG	MOD_FREE	MOD_NEW	ANGLE	STATUS	ADDRESS
LIBRE Inicialmente, el dispositivo toma los valores de referencia de plaza libre (MOD_FREE)	Tue Aug 16 2022 20:08:09 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	48,43	0,02	0	37.914765, -4.724311
	Tue Aug 16 2022 20:09:09 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	48,44	0,02	0	37.914765, -4.724311
	Tue Aug 16 2022 20:10:09 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	48,44	0,06	0	37.914765, -4.724311
	Tue Aug 16 2022 20:11:09 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	48,43	0,02	0	37.914765, -4.724311
	Tue Aug 16 2022 20:12:09 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	48,64	0,12	0	37.914765, -4.724311
KIA (eje delantero) Entra vehículo y sitúa su eje delantero sobre el dispositivo. Es detectado (STATUS).	Tue Aug 16 2022 20:13:10 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	72,05	28,71	1	37.914765, -4.724311
	Tue Aug 16 2022 20:14:10 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	72,08	28,33	1	37.914765, -4.724311
	Tue Aug 16 2022 20:15:10 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	72,20	28,73	1	37.914765, -4.724311
	Tue Aug 16 2022 20:16:10 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	72,06	28,61	1	37.914765, -4.724311
	Tue Aug 16 2022 20:17:10 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	72,05	28,73	1	37.914765, -4.724311
	Tue Aug 16 2022 20:18:11 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	71,90	28,74	1	37.914765, -4.724311
	Tue Aug 16 2022 20:19:10 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	72,02	28,69	1	37.914765, -4.724311
	Tue Aug 16 2022 20:20:10 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	72,05	28,71	1	37.914765, -4.724311
	Tue Aug 16 2022 20:21:11 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	71,65	28,83	1	37.914765, -4.724311
	Tue Aug 16 2022 20:22:11 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	71,99	28,67	1	37.914765, -4.724311
	Tue Aug 16 2022 20:23:11 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	72,08	28,76	1	37.914765, -4.724311
	Tue Aug 16 2022 20:24:11 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	71,99	28,75	1	37.914765, -4.724311
	Tue Aug 16 2022 20:25:11 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	72,02	28,69	1	37.914765, -4.724311
	Tue Aug 16 2022 20:26:11 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	72,05	28,73	1	37.914765, -4.724311
	Tue Aug 16 2022 20:27:11 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	72,03	28,67	1	37.914765, -4.724311
	Tue Aug 16 2022 20:28:11 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	72,00	28,54	1	37.914765, -4.724311
LIBRE Sale el vehículo y vuelven los valores sin perturbación. Detecta que vuelve a estar libre (STATUS).	Tue Aug 16 2022 20:29:12 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	48,69	0,55	0	37.914765, -4.724311
	Tue Aug 16 2022 20:30:12 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	48,25	1,20	0	37.914765, -4.724311
	Tue Aug 16 2022 20:31:12 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	48,48	1,31	0	37.914765, -4.724311
	Tue Aug 16 2022 20:32:12 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	48,61	1,08	0	37.914765, -4.724311
	Tue Aug 16 2022 20:33:12 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	48,45	1,07	0	37.914765, -4.724311
	Tue Aug 16 2022 20:34:12 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	48,34	1,20	0	37.914765, -4.724311
Audi (eje delantero) Entra vehículo y sitúa su eje delantero sobre el dispositivo. Es detectado (STATUS).	Tue Aug 16 2022 20:35:12 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	51,75	10,36	1	37.914765, -4.724311
	Tue Aug 16 2022 20:36:13 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	51,55	10,67	1	37.914765, -4.724311
	Tue Aug 16 2022 20:37:13 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	51,66	10,30	1	37.914765, -4.724311
	Tue Aug 16 2022 20:38:13 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	51,68	10,10	1	37.914765, -4.724311
	Tue Aug 16 2022 20:39:13 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	51,44	10,25	1	37.914765, -4.724311
	Tue Aug 16 2022 20:40:13 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	51,55	10,13	1	37.914765, -4.724311
	Tue Aug 16 2022 20:41:13 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	51,56	10,20	1	37.914765, -4.724311
Audi (centro) Se mueve el vehículo y sitúa su centro sobre el dispositivo. Aumenta la variación (MOD_NEW) respecto a la referencia (MOD_FREE). Detecta que vuelve a estar libre (STATUS).	Tue Aug 16 2022 20:42:13 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	51,46	10,33	1	37.914765, -4.724311
	Tue Aug 16 2022 20:43:13 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	38,27	5,56	1	37.914765, -4.724311
	Tue Aug 16 2022 20:44:13 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	77,86	16,27	1	37.914765, -4.724311
	Tue Aug 16 2022 20:45:14 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	77,68	16,24	1	37.914765, -4.724311
	Tue Aug 16 2022 20:46:14 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	77,68	16,21	1	37.914765, -4.724311
	Tue Aug 16 2022 20:47:14 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	78,05	16,39	1	37.914765, -4.724311
	Tue Aug 16 2022 20:48:14 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	77,68	16,22	1	37.914765, -4.724311
LIBRE Sale el vehículo y vuelven los valores sin perturbación. Detecta que vuelve a estar libre (STATUS).	Tue Aug 16 2022 20:49:14 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	77,68	16,22	1	37.914765, -4.724311
	Tue Aug 16 2022 20:50:14 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	77,67	16,20	1	37.914765, -4.724311
	Tue Aug 16 2022 20:51:14 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	48,14	1,13	0	37.914765, -4.724311
	Tue Aug 16 2022 20:52:15 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	48,25	1,20	0	37.914765, -4.724311
	Tue Aug 16 2022 20:53:15 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	48,45	1,07	0	37.914765, -4.724311
LIBRE Sale el vehículo y vuelven los valores sin perturbación. Detecta que vuelve a estar libre (STATUS).	Tue Aug 16 2022 20:54:15 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	48,69	0,55	0	37.914765, -4.724311
	Tue Aug 16 2022 20:55:15 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	48,79	0,09	0	37.914765, -4.724311
	Tue Aug 16 2022 20:56:15 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	48,58	1,10	0	37.914765, -4.724311
	Tue Aug 16 2022 20:57:15 GMT+0200 (Centro)	A8610A33344B7115	Dispositivo 1	48,43	48,09	1,10	0	37.914765, -4.724311



2. Pruebas para determinar la posición del dispositivo y los umbrales de comparación

2 - PRUEBAS PARA DETERMINAR LA POSICIÓN IDÓNEA BAJO EL VEHÍCULO (A PARTIR DE ESTAS PRUEBAS SE OBTIENEN EMPÍRICAMENTE LOS UMBRALES FINALES DE COMPARACIÓN)								
COMENTARIO	TIME	DEV_EUI	TAG	MOD_FREE	MOD_NEW	ANGLE	STATUS	ADDRESS
LIBRE Inicialmente, el dispositivo toma los valores de referencia de plaza libre (MOD_FREE)	Tue Aug 17 2022 12:01:50 GMT+0200 (Cer)	A8610A33344B7115	Dispositivo 1	54,97	54,97	0,00	0	37.914765, -4.724311
	Tue Aug 17 2022 12:02:49 GMT+0200 (Cer)	A8610A33344B7115	Dispositivo 1	54,97	54,64	0,01	0	37.914765, -4.724311
	Tue Aug 17 2022 12:03:51 GMT+0200 (Cer)	A8610A33344B7115	Dispositivo 1	54,97	54,76	0,02	0	37.914765, -4.724311
	Tue Aug 17 2022 12:04:49 GMT+0200 (Cer)	A8610A33344B7115	Dispositivo 1	54,97	55,10	0,16	0	37.914765, -4.724311
Audi (eje delantero) Entra vehículo y sitúa su eje delantero sobre el dispositivo. Es detectado (STATUS).	Tue Aug 17 2022 12:06:50 GMT+0200 (Cer)	A8610A33344B7115	Dispositivo 1	54,97	59,35	19,98	1	37.914765, -4.724311
	Tue Aug 17 2022 12:07:50 GMT+0200 (Cer)	A8610A33344B7115	Dispositivo 1	54,97	59,34	19,94	1	37.914765, -4.724311
	Tue Aug 17 2022 12:08:50 GMT+0200 (Cer)	A8610A33344B7115	Dispositivo 1	54,97	59,50	19,67	1	37.914765, -4.724311
	Tue Aug 17 2022 12:09:50 GMT+0200 (Cer)	A8610A33344B7115	Dispositivo 1	54,97	59,27	19,94	1	37.914765, -4.724311
	Tue Aug 17 2022 12:10:50 GMT+0200 (Cer)	A8610A33344B7115	Dispositivo 1	54,97	59,18	19,78	1	37.914765, -4.724311
Audi (centro) Se mueve el vehículo y sitúa su centro sobre el dispositivo. Aumenta la variación (MOD_NEW) respecto a la referencia (MOD_FREE).	Tue Aug 17 2022 12:11:50 GMT+0200 (Cer)	A8610A33344B7115	Dispositivo 1	54,97	81,74	14,08	1	37.914765, -4.724311
	Tue Aug 17 2022 12:12:50 GMT+0200 (Cer)	A8610A33344B7115	Dispositivo 1	54,97	81,81	14,04	1	37.914765, -4.724311
	Tue Aug 17 2022 12:13:50 GMT+0200 (Cer)	A8610A33344B7115	Dispositivo 1	54,97	81,83	13,95	1	37.914765, -4.724311
	Tue Aug 17 2022 12:14:51 GMT+0200 (Cer)	A8610A33344B7115	Dispositivo 1	54,97	81,45	14,02	1	37.914765, -4.724311
	Tue Aug 17 2022 12:15:51 GMT+0200 (Cer)	A8610A33344B7115	Dispositivo 1	54,97	81,74	14,10	1	37.914765, -4.724311
Audi (eje trasero) Se mueve el vehículo y sitúa su eje trasero sobre el dispositivo. Disminuye la variación (MOD_NEW) respecto a la referencia (MOD_FREE).	Tue Aug 17 2022 12:16:51 GMT+0200 (Cer)	A8610A33344B7115	Dispositivo 1	54,97	65,45	1,16	1	37.914765, -4.724311
	Tue Aug 17 2022 12:17:51 GMT+0200 (Cer)	A8610A33344B7115	Dispositivo 1	54,97	65,41	1,04	1	37.914765, -4.724311
	Tue Aug 17 2022 12:18:51 GMT+0200 (Cer)	A8610A33344B7115	Dispositivo 1	54,97	65,97	1,49	1	37.914765, -4.724311
	Tue Aug 17 2022 12:19:51 GMT+0200 (Cer)	A8610A33344B7115	Dispositivo 1	54,97	65,34	1,18	1	37.914765, -4.724311
LIBRE Sale el vehículo y vuelven los valores sin perturbación. Detecta que vuelve a estar libre (STATUS).	Tue Aug 17 2022 12:20:51 GMT+0200 (Cer)	A8610A33344B7115	Dispositivo 1	54,97	54,77	0,27	0	37.914765, -4.724311
	Tue Aug 17 2022 12:21:51 GMT+0200 (Cer)	A8610A33344B7115	Dispositivo 1	54,97	54,76	0,02	0	37.914765, -4.724311
	Tue Aug 17 2022 12:22:52 GMT+0200 (Cer)	A8610A33344B7115	Dispositivo 1	54,97	54,64	0,01	0	37.914765, -4.724311
	Tue Aug 17 2022 12:23:52 GMT+0200 (Cer)	A8610A33344B7115	Dispositivo 1	54,97	55,20	0,10	0	37.914765, -4.724311
	Tue Aug 17 2022 12:24:52 GMT+0200 (Cer)	A8610A33344B7115	Dispositivo 1	54,97	54,83	0,03	0	37.914765, -4.724311