Universidad de Alcalá Escuela Politécnica Superior

Grado en Ingeniería Electrónica y Automática Industrial

Trabajo Fin de Grado

Herramienta para la monitorización y asistencia de personas con deterioro cognitivo leve durante sus actividades en exteriores.

Autor: Rodrigo José Juberías Bailón

Tutor: Mª del Carmen Pérez Rubio Cotutor: Alejandro García Requejo

UNIVERSIDAD DE ALCALÁ

Escuela Politécnica Superior

Grado en Ingeniería en Electrónica y Automática Industrial

Trabajo Fin de Grado

Herramienta para la monitorización y asistencia de personas con deterioro cognitivo leve durante sus actividades en exteriores

Autor:	Rodrigo José Juberías Bailón
Tutor/e	es: Mª del Carmen Pérez Rubio
	Alejandro García Requejo
TRIBUNAL:	
Preside	nte: D. Ignacio Fernández Lorenzo
Vocal 1	º: D. Pablo Ramos Sainz
Vocal 2	º: Dª. M.ª del Carmen Pérez Rubio
CALIFIC	ACIÓN:
FECHA:	••••••

A mis padres y hermana, las personas que más me han ayudado en esta larga pero bonita travesía

AGRADECIMIENTOS

Personalmente me gusta hacer las cosas por orden de importancia. De modo que en primer lugar quería agradecer a mis padres y a mi hermana por haberme acompañado a lo largo de todos estos años en toda mi formación, apoyándome en todo momento y aguantándome en esos días donde todo sale al revés y hay que desahogarse con quién menos culpa tiene de que las cosas no hayan ido bien. Sin vosotros ese reto hubiese sido muchísimo más duro y quién sabe si posible.

Agradecer al resto de mi familia por su constante interés en conocer cómo iba la carrera y animarme a seguir mejorando y peleando por un objetivo tan atractivo como es el de ser ingeniero en un ámbito que siempre me ha fascinado.

Por otro lado, agradecer a todas aquellas personas que durante este camino me ha apoyado en algún que otro momento de las diversas maneras en las que he necesitado ese apoyo. Amigos de toda la vida, amigos que he ido conociendo a lo largo de mi vida escolar y a los que he conocido durante esta última etapa.

Agradecer enormemente a mi tutora Mª Carmen y a mi cotutor Alejandro por su paciencia para cuadrar horarios con mis exigencias laborales y personales, por su atención cuando he necesitado de ella y por su colaboración en todo momento, especialmente cuando alguna parte de este interesante proyecto se atascaba hasta el punto de parecer no tener solución.

A todos ellos, simplemente quiero decirles que me siento super afortunado por haberles tenido cerca durante todo este proceso, gracias.

ÍNDICE

AGRADEC	IMIENTOS5
ÍNDICE DE	E FIGURAS10
ÍNDICE DE	E TABLAS
GLOSARIO	D DE ACRÓNIMOS Y SÍMBOLOS17
RESUMEN	21
ABSTRAC'	Γ
RESUMEN	EXTENDIDO
CAPÍTULO) 1: INTRODUCCIÓN
1.1	Antecedentes y estado actual del tema
1.2	Objetivos del Trabajo de Fin de Grado29
1.3	Contexto
1.4	Estructura del documento30
CAPÍTULO	2: ANTECEDENTES
2.1	Dispositivos wearables GNSS
2.2	Conectividad Dispositivos wearables a larga distancia32
2.3	Herramientas para el desarrollo de aplicaciones multiplataforma34

CAPÍTULO	O 3: DESARROLLO	36
3.1	Introducción	36
3.2	Dispositivo DRAGINO LGT-92 LoRa GPS Tracker	37
3.3	Arquitectura del sistema IoT	43
3.4	Programación de la aplicación	46
CAPÍTULO	O 4: RESULTADOS	64
4.1	Introducción	64
4.2	Pruebas experimentales	64
4.3	Conclusiones	76
CAPÍTULO) 5: CONCLUSIONES Y LÍNEAS FUTURAS	77
5.1	Conclusiones	77
5.2	Líneas futuras	78
CAPÍTULO	O 6: PRESUPUESTO	80
CAPÍTULO	7: PLIEGO DE CONDICIONES	82
CAPÍTULO	O 8: MANUAL DE USUARIO	83
CAPÍTULO) 9: BIBLIOGRAFÍA	90
ANEXO: C	ÓDIGOS	95

ÍNDICE DE FIGURAS

•	Figura 1. Comparación de sistemas de comunicación [Bertech 2022]33
•	Figura 2. Estructura de red LoRaWAN [DeepData 2022]33
•	Figura 3 . Comparación de tipos de <i>apps</i> [Aplicaciones 2022] 35
•	Figura 4. Diagrama general del trabajo propuesto
•	Figura 5. Dragino LGT-92 [Dragino 2019]
•	Figura 6. Diagrama de bloques Dragino LGT-9241
•	Figura 7. Diagrama de bloques envíos Dragino LGT9243
•	Figura 8. Almacenamiento del <i>payload</i> del Dragino LGT9245
•	Figura 9. Procesado de datos para su almacenamiento en InfluxDB46
•	Figura 10. Esquema de la aplicación móvil desarrollada
•	Figura 11. Pantalla inicio (izq.) y pantalla registro del cuidador (dcha.)49
•	Figura 12 Pantalla registro del paciente (izq.) y pantalla acceso (dcha.)49
•	Figura 13. Pantalla menú opciones (izq.) y pantalla paciente (dcha.)53
•	Figura 14. Pantalla ajustes54
•	Figura 15. Pantalla de información55
•	Figura 16. Pantallas niveles de batería56
•	Figura 17. Pantalla selección de periodo (izq.) y pantalla de ubicaciones (dcha.) 57
•	Figura 18. Pantalla ruta, <i>markers</i> con información

•	Figura 19. Pantalla listado de pautas médicas (izq.) y pantalla registro de pauta médica (dcha.)
•	Figura 20. Pantalla selección del centro del límite (izq.), pantalla de selección de radio y nombre del límite (centro) y pantalla de confirmación del límite (dcha.)
•	Figura 21. Pantalla creación del punto prohibido
•	Figura 22. Pantalla pautas médicas
•	Figura 23. Pantalla agenda telefónica
•	Figura 24. Datos recogidos con el <i>script</i> MQQ65
•	Figura 25. Datos almacenados en InfluxDB
•	Figura 26. Cuentas registradas y pantalla de acceso denegado
•	Figura 27. Figura 27. Alarma por salir de un límite (dcha.) y alarma por entrar en un punto prohibido (izq.)
•	Figura 28. Funcionamiento de agenda telefónica
•	Figura 29. Funcionamiento de lista de medicación
•	Figura 30. Ruta realizada, vista en mapa (izq.) y en satélite (dcha.)70
•	Figura 31. Visualización del mapa con puntos muy similares70
•	Figura 32. Visualización del mapa una vez eliminados los puntos muy similares
•	Figura 33. Pantalla de notificaciones
•	Figura 34. Pantalla de alarma por botón de auxilio
•	Figura 35. Dispositivo situado en el bolsillo del paciente
•	Figura 36. Dispositivo colgado al cuello

•	Figura 37. Ruta realizada, a la izquierda el mapa y a la derecha la vista de satélite	74
•	Figura 38. Visualización de los puntos registrados durante la ruta	.75
•	Figura 39. Manual de usuario: Primera pantalla (izq.) y pantalla de registro (dcha.)	83
•	Figura 40. Manual de usuario: Pantalla de acceso incompleto (izq.) y pantalla acceso completo (dcha.)	
•	Figura 41. Manual de usuario: Pantalla principal	85
•	Figura 42. Manual de usuario: Menú lateral	86
•	Figura 43. Manual de usuario: Pantalla batería	86
•	Figura 44. Manual de usuario: Acceder a las pantallas para añadir/elim elementos	
•	Figura 45. Manual de usuario: Añadir (izq.) y eliminar (dcha.) elementos	88
•	Figura 46. Manual de usuario: Agenda	89
•	Figura 47. Recogida de datos	95
•	Figura 48. Procesado de datos	.96
•	Figura 49. Almacenamiento de datos	.96
•	Figura 50. Código para dibujar mapas	97
•	Figura 51. Código actualizar pantalla	97
•	Figura 52. Botones de redirección a la siguiente página	.98
•	Figura 53. Código para dibujar mapas	99
•	Figura 54. Actualización de pantallas	100

•	Figura 55. Código para las alertas	100
•	Figura 56. Solicitud a la base de datos	101
•	Figura 57. Escritura en la base de datos	102
•	Figura 58. Borrar elementos de la base de datos	103

ÍNDICE DE TABLAS

•	Tabla 1. Payload del dispositivo Dragino-LGT92 [Dragino 2021]	4 4
•	Tabla 2. Recursos: Costes de software	80
•	Tabla 3. Recursos: Costes de hardware	80
•	Tabla 4. Recursos: Costes de mano de obra	81
	Tabla 5. Recursos: Costes totales	

GLOSARIO DE ACRÓNIMOS Y SÍMBOLOS

- ADC: Analog Digital Converter (Convertidor Analógico Digital).
- API: Application Programming Interfaces (Interfaz de Programación de Aplicaciones).
- AppKey: Application Key (Clave para el cifrado entre el nodo y la aplicación).
- **BLE:** *Bluetooth Low Energy* (Bluetooth de Baja Energía).
- **DeviceEUI:** Device identifier (Identificador Único del Dispositivo).
- **EEPROM:** *Electrically Erasable Programmable Read-Only Memory,* (ROM programable y borrable eléctricamente).
- **INE:** Instituto Nacional de Estadística.
- **GSM:** Global System for Mobile Communications (Sistema Global para Comunicaciones Móviles).
- GNSS: Global Navigation Satellite System (Sistemas de Satélites de Navegación Global).
- **GPS:** *Global Positioning System* (Sistema de Posicionamiento Global).
- I2C: Inter Integrated Circuits (Circuitos Inter-Integrados)
- IMU: Inertial Measurement Unit (Unidad de Medición Inercial).

- **IoT:** *Internet Of Things* (Internet de las Cosas).
- **LED:** *Light Emitting Diode* (Diodo Emisor de Luz)
- LPWAN: Low Power Wide Area Networks (Redes de Área Ancha de Baja Potencia).
- LTE-M: Long Term Evolution cat M1 (Evolución a Largo Plazo Categoría M1).
- **MQTT:** Message Queing Telemetry Transport.
- **NB-IoT:** Narrow Band Internet Of Things (Banda estrecha Internet de las cosas).
- RAM: Random Access Memory (Memoria de Acceso Aleatorio).
- **ROM:** *Read Only Memory* (Memoria de Solo Lectura)
- SCL: System Clock (Sistema de Reloj)
- **SDA:** *System Data* (Sistema de Datos)
- **SIM:** Subscriber Identity Module (Módulo de Identidad del Suscriptor).
- SPI: Serial Peripheral Interface (Interfaz Periférica de Serie)
- **TFG:** Trabajo Fin de Grado.
- **TTN:** *The Things Network* (El Internet de las Cosas)

- UART: Universal Asynchronous Receiver Transmitter (Transmisor Receptor AsíncronoUniversal).
- UMTS: *Universal Mobile Telecommunications System* (Sistema Universal de Telecomunicaciones Móviles).

RESUMEN

El envejecimiento de la población implica una serie de consecuencias, con nuevos desafíos para los sistemas médicos y asistenciales para atender al colectivo de personas mayores y favorecer su autonomía y bienestar. Así, el deterioro cognitivo o la fragilidad son enfermedades que suelen relacionarse con la edad.

Las nuevas tecnologías permiten lograr un mayor bienestar y autonomía en las personas en etapas incipientes de la enfermedad, mejorando su calidad de vida y proporcionando tranquilidad a los cuidadores o familiares.

En este contexto, el presente trabajo se centra en el diseño de una aplicación multiplataforma basada en el dispositivo Dragino LoRaWAN GPS-Tracker LGT92 que permite monitorizar a personas con deterioro cognitivo leve en sus salidas al exterior de la vivienda. Se permitirá la geolocalización en tiempo real, así como el almacenamiento de las rutas y la generación de avisos con la ubicación del paciente en caso de que solicite asistencia, o exceda los límites de la zona de paseo segura. Este sistema fomenta la autonomía de los pacientes y permite extraer información de la progresión de la enfermedad. Además, utiliza la red de comunicación inalámbrica de bajo consumo LoRaWAN para ofrecer cobertura en lugares remotos.

Palabras claves: LoRaWAN, geolocalización, aplicación multiplataforma, comunicación inalámbrica, monitorización.

ABSTRACT

The population ageing implies some consequences with new challenges for the medical and assistance systems to attend elderly people and collaborate to improve their self-care and wellness. Cognitive impairment and frailty are illness related to the age.

The new technologies allow to give a higher wellness and care to the people in the illness incipient stages, increasing their life's quality and providing their family or caregivers some peace of mind.

In this context, the present project deals with the design of a multiplatform application based on the Dragino LoRaWAN GPS-Tracker LGT92, which allows monitoring people with low cognitive impairment when they are outdoors.

The application will provide the geolocation at any time, its routes storage, number of steps and it will generate alerts when the patient requires assistance or enters in an unsafe place. This system helps to give the patient some autonomy and allow the caregivers to receive information about the illness progress. Moreover, it uses the low consumption network LoRaWAN to have coverage in remote places.

Keywords: LoRaWAN, Geolocation, Multiplatform Application, Wireless Communication, Monitoring.

RESUMEN EXTENDIDO

Durante los últimos 20 años la población española ha sufrido un gran envejecimiento. Según datos del Instituto Nacional de Estadística (INE) [Sanitas, 2017] hay 2 millones más de personas en la tercera edad y todavía se espera que este sector de la población siga creciendo. Según el INE, "en España los mayores representan el 18,7% del total de la población y se estima que alcancen el 35% para 2066". Asimismo, en el mismo artículo se recoge que "uno de cada cuatro hogares españoles se ve afectado por un familiar con alzhéimer".

Esto supone un importante desafío a nivel sanitario, pues se deberán atender más pacientes con problemas relacionados con la edad (deterioro cognitivo, demencia, fragilidad, etc.) y para ello habrá que mejorar la cobertura sanitaria actual.

Con esto, el presente Trabajo Fin de Grado (TFG) se propone para tratar de mejorar la calidad de vida de las personas que sufren trastornos cognitivos de carácter leve, así como de las personas que les rodean (familiares o cuidadores) que deben hacerse cargo de dicha situación. Para ello, se va a aprovechar la tecnología que ofrece el dispositivo *Dragino LoRaWAN GPS Tracker LGT92* [Dragino, 2021], con objeto de desarrollar una herramienta que permita monitorizar personas con deterioro cognitivo leve en sus salidas al exterior, generando alarmas en caso de emergencia y ayudando al seguimiento objetivo de la evolución de la enfermedad.

El *Dragino Tracker LGT92* lleva integrada la tecnología GPS (*Global Positioning Sytem*) que permite conocer la localización del paciente en exteriores. Además, posee un acelerómetro de 9 ejes con el que se puede conocer la cantidad de movimiento del paciente, así como detectar una posible caída del mismo. Este dispositivo también dispone de un botón de auxilio que el paciente puede presionar cuando lo necesite con objeto de recibir asistencia.

Para comunicar todos estos datos que se recogen en el dispositivo, se utilizará la tecnología *LoRaWAN* [LoRaWAN,2022]. Dicha tecnología es la ideal en estos casos ya que se trata de comunicación por radiofrecuencia, 868 MHz en Europa, lo cual permite realizar la comunicación incluso en entornos rurales sin acceso a la red de telefonía siempre que se disponga de algún punto donde conectar a internet el *Gateway* (antena) que recogerá la información del dispositivo. De este modo se podrá localizar al paciente y dar servicio en entornos rurales, sin cobertura de telefonía móvil.

Se ha tenido en cuenta que la sociedad de la tercera edad tiene un importante arraigo al entorno rural. La España del siglo XX fue una España muy rural hasta que se dio el éxodo a la ciudad entre 1950 y 1975. Esto ofrece un importante abanico de posibles pacientes que podrían rehuir de la posibilidad de trasladarse a la ciudad cerca de sus cuidadores; además de aquellos que habiendo emigrado vuelven a sus orígenes para pasar una vejez más tranquila.

La tecnología LoRa ofrece la opción de cobertura inalámbrica de bajo consumo en entornos rurales en los que la red móvil puede no estar accesible y con una atención sanitaria que puede tener carencias de servicio respecto a los ofrecidos en entornos urbanos [INE, 2021].

Una vez recibida la información por el *Gateway* de *LoRa*, este la sube a la red. Se usará *The Things Network* para el almacenamiento de datos de las comunicaciones llevadas a cabo mediante *LoRaWAN*. Una vez montada la red de transmisión entre el dispositivo y la plataforma de almacenamiento de datos, se debe lanzar un servidor que recoja la información recibida en TTN (*The Things Network*), la procese y la almacene en una base de datos que se podrá consultar y de donde reciba toda la información el cuidador a cargo del paciente que porta el dispositivo *Dragino LoRaWAN GPS Tracker LGT92*.

La información recibida en el servidor TTN ha sido trasladada a un broker MQTT, del que se extrae la información. Para ello, se ha creado un servicio web programado en Python encargado de diseccionar la información recibida, escoger la que se necesite, procesarla y almacenarla en InfluxDB. InfluxDB es la plataforma que permite crear una base datos, visualizar las tablas de los datos que se almacenan, sus respectivos gráficos y recoger dichos datos mediante la aplicación multiplataforma que se ha creado para que el cuidador esté en todo momento informado acerca de la situación del paciente.

El cuidador recibirá toda la información acerca del paciente en su dispositivo móvil. Para ello se va a usar el *framework* multiplataforma IONIC, el cual permite programar una aplicación web mediante *Angular* y posteriormente traducirla tanto para Android como para iOS. Esto ofrece una mayor flexibilidad, al no tener que exigir al cuidador un teléfono móvil inteligente específico, pues Android e iOS son los principales sistemas operativos usados en telefonía móvil.

La aplicación multiplataforma diseñada permitirá el registro del cuidador y la monitorización del paciente. Para ello, se recogerá la información del paciente, almacenada en la base de datos, y se mostrará en un menú de opciones donde el cuidador podrá escoger los distintos tipos de datos que desea consultar (localización, pasos realizados, estado del botón de alarma, batería del dispositivo, ...).

La aplicación cuenta con un sistema de alerta para avisar en tiempo real de situaciones que así lo puedan requerir. Cuando el paciente pulsa el botón de socorro se debe avisar al cuidador de manera instantánea y dar una prioridad máxima, que no se le otorga al nivel de batería del dispositivo, por ejemplo.

La aplicación debe tener una interfaz amistosa con el cuidador, para facilitarle en todo momento la monitorización del paciente, aunque el cuidador no esté acostumbrado al uso de las *apps* de los teléfonos inteligentes. Para ello, se usará un menú de botones que redirigirá a las pantallas con los distintos datos almacenados. Se proporcionará también una pantalla con un manual de usuario que facilite el manejo de la aplicación.

Por otro lado, también se permitirá configurar ciertos ajustes sobre la aplicación al cuidador, para ofrecerle una comodidad óptima durante el uso de esta. Además, se le permitirá configurar alertas que pueda considerar importantes en la monitorización del paciente usando los datos que obtiene el dispositivo *Dragino LoRaWAN GPS Tracker LGT92*, como por ejemplo un aviso en caso de que el paciente se aleje de una zona permitida o se acerque a otra prohibida.

La recogida de datos, su procesamiento, almacenamiento, análisis y visualización permitirá mejorar la calidad de vida tanto del cuidador como del paciente que sufre un trastorno cognitivo leve de una forma económica y eficaz en entornos de exterior.

CAPÍTULO 1: INTRODUCCIÓN

1.1 Antecedentes y estado actual del tema

El aumento de la esperanza de vida implica asimismo una mayor prevalencia de enfermedades asociadas a la edad, como puede ser el deterioro cognitivo o demencia. Así, se han llevado a cabo numerosos estudios tanto sanitarios como tecnológicos para poder detectar cambios de conducta que indiquen posibles deterioros cognitivos, dada la importancia de una atención médica temprana. Asimismo, se trata de retrasar al máximo la institucionalización, permitiendo una mayor permanencia en el hogar, con objeto de mejorar la calidad de vida de los pacientes y reducir gastos del sistema sanitario [Elsevier 2019].

Este proyecto aborda la parte tecnológica que ofrece diversas alternativas para mejorar la calidad de vida de las personas que padecen enfermedades cognitivas leves o un inicio de demencia. Para ello se han estudiado wearables que lleven incorporados GNSS (Global Navigation Satellite Sytems), con objeto de tener al paciente localizado en exteriores en todo momento, permitiendo una autonomía supervisada y generando alarmas en caso de emergencias. Los wearables son dispositivos que el paciente lleva consigo en su vestimenta. Existen pulseras, relojes, colgantes, calcetines, ropa, etc. En [Stavropoulos 2020] y [Cote 2021] se presenta una revisión actualizada de trabajos que emplean wearables en pacientes con demencia, indicando el enorme potencial de estos dispositivos para ayudar a la independencia y mejorar la sostenibilidad a largo plazo de los sistemas sanitarios. No obstante, la penetración de estos dispositivos en los sistemas de salud es todavía muy baja y muchas de las propuestas se quedan en fase piloto de pruebas en el laboratorio, con lo que es importante dedicar esfuerzos para contribuir a su penetración en el mundo real y evaluar el grado de aceptación en personas mayores [Neubauer 2018].

Además, para desarrollar servicios y aplicaciones es necesario conectar usuarios, sensores, interfaces, almacenamiento de datos y procesamiento. En general, se requieren comunicaciones inalámbricas para cumplir con los requisitos de no intrusión y escalabilidad.

Hay múltiples opciones en función de su rango, velocidad de datos y consumo [AIOTI 2015]. Tecnologías como Bluetooth Low Energy (BLE) o ZigBee permiten comunicaciones de corto alcance con bajas velocidades de transmisión de datos y bajo consumo de energía. WiFi y Bluetooth ofrecen velocidades de datos más altas, pero implican mayores consumos. Las tecnologías celulares, como GSM (2G), UMTS (*Universal Mobile Telecommunications System*) (3G) o LTE (*Long Term Evolution*) (4G), ofrecen áreas de cobertura de kilómetros a altas tasas de envío de datos, pero implican consumos elevados y pueden presentar problemas de cobertura en entornos rurales aislados. Por otro lado, recientemente, las redes LPWAN (*Low Power Wide Area Networks*, Redes de Área Extensa y Baja Potencia) permiten comunicación de datos a larga distancia con un bajo consumo, pero a costa de una menor tasa de transmisión [Oliveira 2017] [Tayeh 2020]. NB-IoT (*Narrow Band - Internet Of Things*) y LTE-M pueden ser una buena solución para entornos rurales por su bajo consumo eléctrico y compromiso de cobertura en lugares de difícil acceso [Lauridsen 2016].

En este trabajo, se ha llevado a cabo un análisis de distintos productos que ofrece el mercado para ver cuál es el que mejor encaja en las exigencias del proyecto, considerando que debe permitir la monitorización de personas con deterioro cognitivo leve en sus salidas al exterior de la vivienda. El dispositivo debe ser económico, versátil, abierto a su programación, además de ser sencillo y cómodo para el paciente. Existen varios dispositivos atractivos, pero de precios desorbitados que podrían suponer una importante barrera para el paciente y aumentar el riesgo de discriminación según el nivel económico. Otras soluciones no permiten ajustar su programación, lo que impide una mejor adecuación del dispositivo al paciente y sus necesidades. Así, se ha optado por usar el dispositivo Dragino LoRaWAN LGT-92 [Dragino 2019]. Dicho dispositivo permite modificar su programación, es económico, de uso sencillo y se comunica por LoRaWAN, ofreciendo la posibilidad de comunicaciones en entornos rurales que pueden no tener cobertura de la red de telefonía móvil. Se trata de un dispositivo con un botón de alarma que debe ser presionado si el paciente requiere asistencia inmediata y un botón de encendido y apagado en el lateral. Tiene un diseño muy simple que se espera que goce de una buena aceptación.

1.2 Objetivos del Trabajo de Fin de Grado

Este proyecto tiene como objetivo principal el desarrollo de una herramienta para monitorización en tiempo real y asistencia a personas con deterioro cognitivo leve durante sus actividades en entornos de exteriores. La monitorización se llevará a cabo a partir de la información que recoja un dispositivo Dragino LoRaWAN GPS Tracker LGT92 [Dragino 2019]. Este dispositivo cuenta con GPS para la localización en exteriores del paciente, con sensores inerciales como un acelerómetro de 9 ejes que permite medir la cantidad de movimiento del paciente; y con comunicación mediante LoRa para la transmisión de dichos datos al cuidador mediante la aplicación desarrollada. Se va a diseñar una aplicación multiplataforma que pueda ser ejecutada en prácticamente cualquier *smartphone* para que el cuidador pueda conocer el estado del paciente en todo momento.

Esta aplicación permitirá:

- Conocer la posición del paciente en cada instante.
- Almacenar y visualizar las rutas que ha seguido el paciente, así como conocer los puntos en los que ha estado, de cara a poder inferir rutinas o detectar cambios.
- Recibir notificaciones instantáneas cuando el paciente pulse el botón de emergencia solicitando asistencia.
- Ser avisados si el paciente se aleja de la zona segura o si por el contrario se acerca a algún lugar que el cuidador considere peligroso.
- Almacenar pautas médicas y recibir avisos cuando el paciente deba medicarse.

Para ello se usará el *framework* de IONIC que permitirá desarrollar una aplicación híbrida o multiplataforma y luego traducirla tanto para Android como para iOS.

Para tener localizado al paciente y poder mostrar sus coordenadas en un mapa se usará la API (Interfaz de Programación de Aplicaciones) de Google Maps.

1.3 Contexto

El presente trabajo se ha realizado en el Departamento de Electrónica de la Universidad de Alcalá, en el grupo de investigación GEINTRA-US&RF, dentro del área de diseño de espacios inteligentes para la vida independiente. Específicamente, el trabajo se ha desarrollado en el marco de dos proyectos: "Improving and fostering accessibility, active living and wellness of PWD by means of localization techniques (PeaceOfMind-UAH)", ref. PID2019-105470RA-C33), financiado por el Ministerio de Ciencia, Innovación y Universidades de España; y el proyecto "Sistema de evaluación de cambios de comportamiento para la detección temprana de trastornos cognitivos en entornos rurales mediante técnicas NILM y wearables (RACC)", ref. CM/JIN/2021-016, financiado por la Comunidad de Madrid y la Universidad de Alcalá.

1.4 Estructura del documento

Este documento abarca 9 capítulos en los que se desarrollan todos los aspectos tratados en este Trabajo de Fin de Grado (TFG).

En el primer capítulo se introduce brevemente la temática del TFG, se expone el marco del mismo y se aclaran los objetivos finales del proyecto. En el capítulo siguiente se resumen las herramientas que se utilizan a lo largo del proyecto para lograr los objetivos previamente establecidos. Se detallan las distintas herramientas disponibles y la elección de las que se usan y sus motivos.

El capítulo 3 es donde se desarrolla la parte principal del proyecto. A lo largo del capítulo se explica todo el proceso de comunicación entre el paciente y el cuidador mediante el uso del Dragino LGT92, la comunicación por LoRaWAN, el almacenamiento de los datos en InfluxDB, el diseño de la aplicación multiplataforma por medio de IONIC y finalmente la visualización de los datos al ejecutar dicha aplicación en el dispositivo móvil del cuidador. Se desarrollan todos estos procesos de forma individual y, finalmente, en el capítulo 4 se visualizan los resultados producidos tras la unión de todos ellos.

Debido a que se trata de un proyecto que se podría extender aún más, en el capítulo 5 se plantean posibles líneas futuras que permitan avanzar con la aplicación y poder mejorar el proyecto inicial llevado a cabo en este TFG. Además, se incluyen las conclusiones derivadas de la realización del proyecto.

En el capítulo 6 se recogen los costes del proyecto y se redacta un presupuesto. A lo largo del capítulo 7 se informa acerca de qué requisitos se deben cumplir para hacer un uso adecuado de la aplicación desarrollada que se completa en el capítulo siguiente con la redacción de un manual de usuario.

Finalmente, se incluye la bibliografía y un anexo con los extractos más relevantes del código desarrollado.

CAPÍTULO 2: ANTECEDENTES

2.1 Dispositivos Wearables GNSS (GPS)

Los *wearables* son dispositivos electrónicos que porta el usuario y que incluso permiten cierta interacción. Estos dispositivos pueden incorporar microprocesadores que realizarán ciertas funciones como por ejemplo la geolocalización por GPS, control del ritmo cardíaco, control de la temperatura corporal, etc.

Los *wearables* permiten nuevas posibilidades tecnológicas de una manera discreta y poco intrusiva pues se trata de pulseras o relojes, por ejemplo, que incluyen todo un sistema electrónico para distintas aplicaciones. Generalmente, estos dispositivos se usan para controlar la salud del usuario, su actividad física, su seguridad o simplemente para comunicarse con el teléfono móvil y tener acceso a ciertas funciones del mismo de una manera más cómoda.

Este tipo de dispositivos abre un nuevo abanico de posibilidades en sectores como el sanitario ya que permite la monitorización de pacientes de una forma efectiva y apenas intrusiva. Estos dispositivos integran un microprocesador que permite incluir sensores y diversas funcionalidades en el dispositivo.

Además, en muchos casos llevan incluida una memoria que permite almacenar datos para posteriormente analizarlos y consultarlos si así se desea. Por último, destacar que se ha avanzado mucho recientemente en la comunicación de los *wearables* mediante diversas maneras como Bluetooth, WiFi, LoRaWAN, etc. [Wearables 2021].

En este proyecto se han estudiado los *wearables* GNSS, que son aquellos que llevan incorporada la tecnología GNSS, permitiendo localizar al usuario en exteriores trasmitiendo sus coordenadas. De este modo se satisface la necesidad de saber dónde está el paciente en sus excursiones al exterior de la vivienda.

El GNSS es el "sistema de navegación global por satélite". Toda una red de satélites envía señales de posicionamiento a los receptores GNSS que, a partir de dichas señales, determinan la ubicación. GPS, del gobierno de Estados Unidos, fue el primer GNSS y es el más usado a día de hoy, sin embargo, existen otros tipos de GNSS para posicionamiento por satélite [Polar, 2022].

El cálculo de una posición sobre la Tierra requiere de 3 satélites a los que hay que añadir un cuarto para determinar la altitud, así ya se puede determinar la posición tridimensional. Para hallar dicha posición se usa el tiempo recorrido por las señales GNSS en su recorrido entre el satélite y el receptor. Por ello los satélites están equipados con relojes atómicos de gran precisión pues un mínimo error en el tiempo puede suponer una diferencia de kilómetros [GNSS, 2022].

2.2 Conectividad dispositivos wearables a larga distancia

Los dispositivos *wearables* más interesantes en este proyecto son aquellos que permiten la comunicación a larga distancia sin dependencia de la red de telefonía. Por ello se descartan modos de comunicación de corto alcance como Bluetooth.

Entre las opciones de comunicación a larga distancia existen 2 opciones: comunicación por GSM, o bien mediante redes LPWAN, según puede observarse en la figura 1.

Las siglas GSM corresponden al nombre en inglés del Sistema Global de Comunicaciones Móviles. Sin embargo, este tipo de comunicación es más comúnmente conocida como 2G. GSM opera en Europa en el espectro radioeléctrico de 900 y 1800 MHz. La comunicación se realiza a través de ondas de radio. Las estaciones móviles, dispositivos con la tarjeta SIM que los identifica, se comunican con una estación base que se unen a la red mediante un controlador que gestiona los recursos de comunicación. Los controladores están conectados por cable con un centro conmutador que recopila los datos [GSM, 2022]. GSM requiere de cobertura para poder llevar a cabo la comunicación. Esto puede suponer un problema en entornos rurales donde la cobertura móvil es reducida o incluso inexistente. Por otro lado, este tipo de conexión requiere de una tarjeta SIM (Subscriber Identity Modul) y un servicio de telefonía móvil constante lo cual puede suponer un pequeño inconveniente económico.

Por su parte, las redes LPWAN (*Low Power Wide Area Networks*, Redes de Área Extensa y Baja Potencia) permiten la comunicación mediante un consumo reducido. Se trata de una alternativa al sistema GSM, aunque solo sirve para volúmenes reducidos de datos, no para vídeo ni audio.

Son redes que permiten la instalación de numerosos nodos, lo cual posibilita dar respuesta en territorios extensos. Es un sistema no muy costoso. Existen varias redes como Sigfox, una compañía que ha implantado su propia red en el territorio y se puede acceder a ella contratando sus servicios; LoRaWAN, es una red estándar desarrollada sobre la modulación radio LoRa; NB-IoT, similar a Sigfox pero que requiere de una tarjeta SIM [LPWAN, 2022].

La tecnología LoRaWAN usa una modulación de radio-frecuencia patentado por Semtech. Hace uso de las bandas de radio ISM (Industriales, científicas y médicas) por debajo de 1 GHz, trabajando en Europa en la banda de 863 MHz a 870 MHz.. Se trata de una tecnología con alta tolerancia a interferencias, bajo consumo, alcance de hasta 10 km en condiciones óptimas y baja trasferencia de datos de hasta 255 bytes.

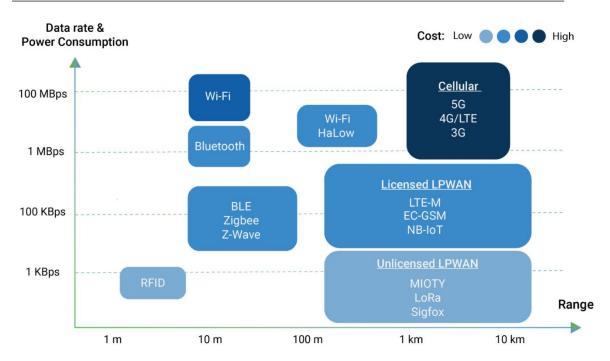


Figura 1. Comparación de sistemas de comunicación [Bertech, 2022].

El protocolo de comunicación de LoRaWAN se compone de *gateways* (antenas) que reciben y envían información hacia los nodos, y de nodos que son los dispositivos finales que envían y reciben información hacia el *gateway* [Semtech, 2021], según representa la Figura 2.

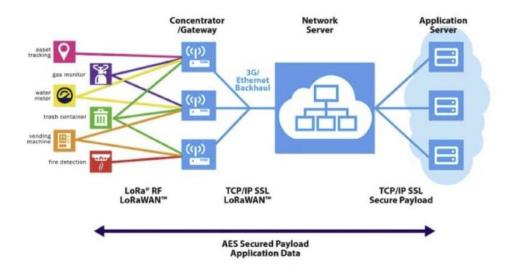


Figura 2. Estructura de red LoRaWAN [DeepData 2022].

2.3 Herramientas para desarrollo de aplicaciones

Según se muestra en la Figura 3, a la hora de desarrollar aplicaciones existen varias posibilidades: herramientas o *frameworks* nativos; herramientas o *frameworks web;* o bien herramientas o *frameworks* híbridos.

Las aplicaciones nativas son aquellas que se desarrollan para un único equipo o plataforma. Se desarrollan en el lenguaje de programación específico de dicho equipo o plataforma. Este tipo de aplicaciones se adapta al 100% al dispositivo y puede usar todas sus funcionalidades (GNSS, cámara, acelerómetro, etc.).

Debido a que se trata de una aplicación específica para un sistema operativo determinado, exige de un nuevo desarrollo totalmente nuevo para los dispositivos con otro sistema operativo como puede suceder con teléfonos móviles Android e iOS. Esto eleva el tiempo de desarrollo y encarece la producción.

Las principales herramientas para desarrollar este tipo de *apps* son "Android Studio" para aplicaciones para el sistema operativo Android y Xcode para el sistema operativo iOS.

Las aplicaciones web se desarrollan a través de tecnologías web que se pueden usar accediendo a un navegador a través de internet. Esto conlleva un coste reducido y un desarrollo sencillo, pues se distribuyen a los usuarios mediante el uso del navegador y sólo hay que implementar un diseño responsive que permita adaptar la web al dispositivo desde donde se ejecuten. Un diseño web responsive es el que es capaz de adaptarse a pantallas de diferentes tamaños con un solo sitio web. El sistema detecta automáticamente el ancho de la pantalla y a partir de ahí adapta todos los elementos de la página, desde el tamaño de letra hasta las imágenes y los menús, para ofrecer al usuario la mejor experiencia posible [Responsive 2022].

Sin embargo, esto conlleva un uso mucho más limitado ya que no se puede acceder a los recursos específicos del dispositivo. Además, la seguridad depende del navegador, lo cual no es lo más adecuado. Estas aplicaciones se desarrollan habitualmente con Angular.

Finalmente, las aplicaciones híbridas, también conocidas como aplicaciones multiplataforma, poseen características de estas dos ya mencionadas. Se aprovecha la versatilidad del desarrollo *web* con el uso de algunas funciones *hardware* de las aplicaciones nativas. La aplicación se visualiza en un *WebView* y finalmente se integra como una aplicación nativa tras una traducción según el sistema operativo donde se desee implementar la *app*. No se requieren de versiones diferentes para cada sistema operativo, el propio *framework* se encargará de la traducción al sistema operativo deseado.

El principal inconveniente de este tipo de *apps* es el rendimiento, pues no tienen tanta calidad como las *apps* nativas y su velocidad de carga es mucho más lenta.

Existen infinidad de *frameworks* que permiten desarrollar este tipo de aplicaciones como IONIC, Flutter o Rect Native [Aplicaciones, 2022].

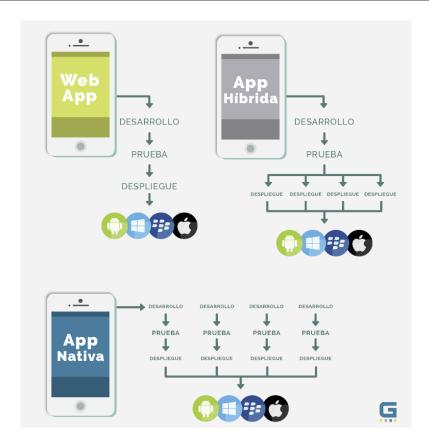


Figura 3. Comparación de tipos de *apps* [Aplicaciones 2022].

En este proyecto se ha decidido usar el *framework* IONIC para llevar a cabo el desarrollo de una aplicación híbrida. Se ha escogido este tipo de aplicación ya que es más rápida que una aplicación nativa y con un mismo código se podrá dar servicio a cualquier dispositivo independientemente de su sistema operativo. Al tratarse de una aplicación de recopilación y muestreo de datos, no requiere de recursos específicos del dispositivo por lo que la principal ventaja de las aplicaciones nativas no sería relevante como para optar por un desarrollo más complejo. Por su parte se ha descartado una aplicación *web* ya que, aunque son sencillas de desarrollar, podrían tener una interfaz mucho menos intuitiva para el usuario. Además, es preferible tener cierta seguridad a la hora de representar datos como la localización de un paciente de estas características.

Por otro lado, se escoge el *framework* IONIC por su excelente similitud a una aplicación nativa, dado que proporciona una interfaz mucho más atractiva e intuitiva para el cuidador. También cabe destacar que posee una importante comunidad de desarrolladores y usuarios lo que constituye una ventaja a la hora de consultar acerca de cómo desarrollar ciertos detalles de la aplicación [Ionic 2022].

CAPÍTULO 3: DESARROLLO

3.1 Introducción

Este proyecto aborda el desarrollo de una aplicación multiplataforma que permite monitorizar a personas con leves deterioros cognitivos a larga distancia, durante sus paseos en el exterior de la vivienda. Para ello se usan diversas herramientas y tecnologías.

En primer lugar, se dota al paciente con un dispositivo Dragino LGT 92 [Dragino 2019]. Este dispositivo es el encargado de recoger todos los datos del paciente y trasmitirlos mediante LoRaWAN. Estos datos son recibidos por un *gateway* o antena que se encarga de recopilarlos y almacenarlos en la *web*. Para almacenarlos en la *web* se hace uso de The Things Network (TTN), el servidor de red en la nube para la especificación LoRaWAN.

Una vez se ha conseguido enviar los datos desde el dispositivo y almacenarlos en la web se deben extraer para poder guardarlos en una base de datos para su posterior visualización y procesamiento. Para ello se ejecuta un cliente MQTT (Message Queing Telemetry Transport) mediante el lenguaje de programación Python. Esto permitirá recoger los datos de TTN, procesarlos y almacenarlos ordenadamente en InfluxDB, la base de datos donde se recopilará toda la información.

The Things Network (TTN) es un servidor de red LoRaWAN en código abierto. Es una red pública con miembros en todo el mundo. Toda la infraestructura de la puerta de enlace es administrada por los usuarios de la red. Además, The Things Network es una red pública altamente segura que admite un verdadero cifrado de extremo a extremo, mitigaciones contra varios ataques de intermediarios y soporte para diferentes claves de cifrado de 128 bits para cada dispositivo final [TTN 2022].

Por su parte, InfluxDB es una base de datos de series temporales diseñada para el almacenamiento rápido y de alta disponibilidad y la recuperación de datos de series temporales. Puede funcionar como una solución independiente, o se puede utilizar para procesar datos. Es una base de datos de código abierto y cuenta con su propio lenguaje de consulta inspirado en SQL. Posee etiquetas que permiten indexar series para consultas rápidas y eficaces [InfluxDB 2022].

Finalmente, se desarrolla la aplicación móvil que permitirá al cuidador leer los datos que ha ido recogiendo el Dragino del paciente. Estos datos se muestran a través de la aplicación mediante una solicitud a la base de datos que proporcionará la información requerida al usuario.

La Fig. 4 muestra un diagrama general del sistema propuesto.

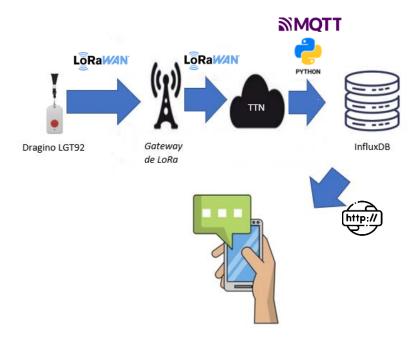


Figura 4. Diagrama general del trabajo propuesto.

3.2 Dispositivo Dragino LGT-92 LoRa GPS Tracker

I. Visión general.

El dispositivo Dragino LGT-92 (ver figura 5) [Dragino 2019] es un rastreador de bajo consumo y de código abierto. Incluye un acelerómetro de 9 ejes que permitirá controlar el movimiento y posibles caídas. Como bien indica su nombre completo, dispone de la tecnología GPS para poder ser ubicado en el mapa siempre que haya visibilidad satelital. Todos estos datos son trasmitidos mediante comunicación LoRaWAN, la tecnología inalámbrica que permite enviar datos y alcanzar rangos largos a velocidades de transmisión de datos bajas.

El Dragino LGT-92 LI está alimentado por una batería recargable de 1000 mA, lo cual permite un largo periodo de uso debido a su escaso consumo. Además, dispone de un botón de alarma que al activarse envía un mensaje de alerta [Dragino 2019].

Estas características permiten tener localizado el dispositivo en todo momento y, por tanto, a su portador. También se puede supervisar el movimiento del portador y emitir avisos mediante la pulsación del botón de alarma. Esto abre un abanico de posibles usos para el dispositivo. Se puede usar como *tracker* para localizar al portador, como medio de comunicación para avisar de un problema a distancia mediante el botón, para controlar la actividad física gracias a su acelerómetro, etc.



Figura 5. Dragino LGT-92 [Dragino 2019].

II. Especificaciones generales

El dispositivo Dragino LGT-92 se basa en el microcontrolador STM32L072CZT6 [Dragino 2019]. El microcontrolador es el que controla todo el dispositivo, sus sensores, comunicaciones, almacenamiento en memoria, etc. Dispone de una RAM de 20 kB y una EEPROM de 6 kB [Dragino, 2019].

La memoria de acceso aleatorio (RAM) es la memoria de la computadora que almacena la información que un programa necesita mientras se ejecuta [RAM 2022].

Mientras que la EEPROM (*Electrically Erasable Programmable Read-Only Memory*, ROM programable y borrable eléctricamente) es un tipo de memoria ROM (Memoria de Solo Lectura) que puede ser programada, borrada y reprogramada eléctricamente. Puede ser leída un número ilimitado de veces, pero sólo puede ser borrada y reprogramada unas determinadas veces [EEPROM, 2022].

El STM32L072CZT6 trabaja a una frecuencia de 32 MHz y con una tensión de alimentación entre 1.65 V y 3.6 V. Su corriente nominal es de 290 nA. Esto demuestra que efectivamente se trata de un microcontrolador de un consumo mínimo ideal para mantener un dispositivo de estas características en funcionamiento durante un largo periodo de tiempo sin necesidad de una batería excesivamente elevada.

Trabaja a temperaturas entre -40°C y 85°C, luego son unas condiciones más que aceptables para la aplicación propuesta. Posee una memoria *Flash* de 192 kB que permite un almacenamiento no volátil suficiente.

Este microcontrolador posee 2 conversores Digital-Analógico (D/A), 3 canales *Inter Integrated Circuits* (I2C) para la comunicación interna, 2 canales *Serial Peripheral Interface* (SPI), *System Data* (SDA) y *System Clock* (SCL), para comunicación interna también, 6 *timers* o contadores para controlar el tiempo internamente y 4 canales *Universal Asynchronous Receiver-Transmitter* (UART) para comunicación con el exterior en este caso.

Se trata de unas características bastante completas y adecuadas para poder controlar todo el sistema que conforma el dispositivo Dragino LGT-92 [STM 2022].

Por otro lado, la Unidad de Medición Inercial (IMU) se encarga de controlar el movimiento del dispositivo mediante un acelerómetro de 9 ejes que incorpora un giroscopio, un magnetómetro y un acelerómetro todos ellos de triple eje [Dragino, 2019].

El acelerómetro es un MPU9250 de 9 ejes. Posee comunicación SPI e I2C, luego se puede comunicar con el STM del dispositivo. También concuerda en sus especificaciones de funcionamiento, pues se alimenta con 2.5 V y opera en temperaturas entre -40 °C y 85 °C. Posee una corriente de operación de 3.5 mA pero que se ve drásticamente reducida a 8 uA en modo reposo.

Este sensor permite conocer en todo momento la posición del Dragino LGT-92 y así detectar posibles caídas y momentos de reposo del usuario, pues al colocar el dispositivo en una mesa, por ejemplo, el dispositivo no envía datos con la misma frecuencia ya que entra en modo reposo al no estar siendo portado por el usuario. Además, este tipo de sensores permite programar contadores de pasos a partir de los datos que proporciona el dispositivo al moverse [MPU, 2022].

Para completar el dispositivo se dispone de un módulo GPS L76-L [GPS L76 2022]. Este módulo permite tener localizado el dispositivo en exteriores, obtiene sus coordenadas x, y, z. Posee un tiempo de inicio menor a 30 s y uno de reinicio menor a 1 s. El tiempo de inicio es el que tarda desde que se enciende el dispositivo hasta que el GPS está operativo mientras que el de reinicio es el que trascurre cuando el microcontrolador requiere las coordenadas GPS y debe despertar al módulo GPS para que este actúe.

El módulo GPS posee una precisión de 2.5 m. Esta precisión es habitualmente suficiente para localizar personas en espacios exteriores.

El consumo de este módulo depende de su modo de operación. Cuando está en modo reposo el consumo es de apenas 8 uA, mientras que en estado operativo es de 13 mA en modo *tracking* y de 16 mA cuando está adquiriendo las coordenadas y trasmitiéndoselas al microprocesador [Dragino, 2019].

Por último, se incluye un botón de alarma para que el dispositivo se pueda usar de forma manual. En este caso, dicho botón sirve para enviar un mensaje de alerta instantáneo avisando de una emergencia al receptor. Gracias a que se trata de un dispositivo de código abierto, se puede programar la funcionalidad del botón al gusto del desarrollador. Para este proyecto se usa la funcionalidad por defecto que consiste en el envío de un mensaje de alerta inmediato y el paso a un "modo de emergencia" que posee unas características diferentes al modo normal como el aumento de la frecuencia de los envíos de datos al receptor. Esto hará que el receptor reciba el aviso de forma inmediata y que, en caso de perderse el mensaje, se siga informando de la emergencia con una frecuencia mayor que cuando no se está en dicho estado.

El modo emergencia está programado para ejecutarse al producirse una pulsación de 3 s de duración. Dicha duración se puede programar, pero no se ha modificado pues se considera adecuada para evitar avisos erróneos por pulsaciones involuntarias.

Finalmente, una vez que el dispositivo ha recopilado todos los datos, estos deben ser trasmitidos a un receptor. Para ello se usa la comunicación LoRaWAN mediante el módulo LoRa SX1276/1278. Este módulo posee un alcance de trasmisión de aproximadamente 8 km en espacios abiertos. Funciona para umbrales de tensión de alimentación entre 3.6 V y 5.5 V. Trabaja en frecuencia de 868 MHz.

Posee una potencia de trasmisión de alrededor a 1 W, en torno a 610 mA de corriente. Durante la recepción, la corriente baja hasta unos 20 mA.

La comunicación se lleva a cabo a través de la interfaz UART con una velocidad de 9600 baudios por defecto, pero acepta velocidades entre 1200 y 115200 baudios. Dispone de un conector para incorporar una antena para mejorar el alcance de este. [LoRA 2022].

Este módulo permite recoger los datos del dispositivo a distancia gracias a su trasmisión hacia el *gateway* encargado de recibirlos. Se trata de un módulo pequeño de modo que se puede incorporar la antena en el interior de encapsulado del dispositivo sin necesidad de habilitar ninguna salida para la antena, lo cual es ideal para evitar que esta se pueda romper durante el uso del dispositivo por parte del paciente.

La figura 6 representa un diagrama de bloques que recoge las funcionalidades principales del dispositivo Dragino LGT-92.

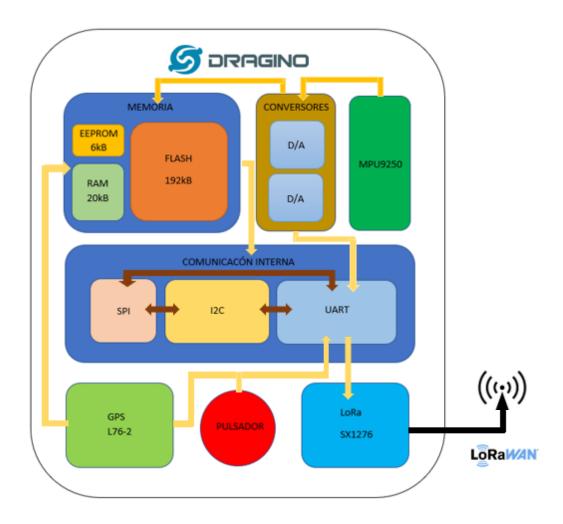


Figura 6. Diagrama de bloques Dragino LGT-92.

III. Software

Dragino proporciona *software* libre de su dispositivo para que este pueda ser manipulado por el usuario de modo que incluya una configuración personalizada según necesidades particulares. Esto permite conocer el funcionamiento interno del dispositivo y modificar lo que se desee.

El Dragino LGT-92 es un dispositivo que a través de sus sensores GPS indica la latitud y longitud donde se encuentra el dispositivo. Para ello se usa una función que pone en funcionamiento el dispositivo GPS y este recoge los datos necesarios para determinar la posición del paciente. Además, manipulando el código fuente que maneja el microcontrolador se podría incorporar la lectura de la altitud del dispositivo.

En este caso no se ha incorporado esta funcionalidad ya que para posicionar al paciente en exteriores no se requiere de la altitud y por tanto se ha descartado incorporarla para evitar una menor duración de la batería y mayor complejidad en la trasmisión de datos aportando información inútil o irrelevante.

Mediante la lectura de la tensión a la salida de la batería se puede conocer el nivel de ésta. El programa incluye dicha lectura entre los datos recogidos para que al trasmitir y recibir los datos se pueda visualizar su nivel. Al hacerse una lectura en mV se plantea la posibilidad de traducir el valor a porcentajes, sin embargo, se ha decidido hacer dicha conversión fuera del programa que se ejecuta en el microcontrolador para evitar carga en este y que dicho cálculo no afecte en el rendimiento del dispositivo, pues es un cálculo sencillo que se puede llevar a cabo fuera del dispositivo.

El *software* también incorpora una interrupción que se activa cuando la entrada correspondiente al botón de alarma se encuentra en nivel alto durante un tiempo de 3 segundos. Al activarse dicha interrupción el dispositivo pasa al modo emergencia que lo primero que hace es cambiar su variable *booleana* correspondiente a *TRUE*. Al producirse este cambio se envían los datos actuales instantáneamente a la vez que se modifica la frecuencia de envío de datos. Esta variable permanece en *TRUE* durante un tiempo determinado. Todos estos tiempos se pueden modificar modificando el valor de las variables correspondientes. Se ha decidido continuar adelante con los valores por defecto. Mantener el estado de emergencia durante el tiempo suficiente como para haber enviado los datos 10 veces también se ha considerado aceptable ya que, aun perdiéndose algún envío por fallos en la trasmisión o recepción, se dan las opciones suficientes como para recibir el mensaje.

Los envíos en modo emergencia se hacen cada 2 minutos, de modo que si algún envío falla brevemente se volverá a informar de la situación. Los envíos fuera del modo de emergencia se hacen cada 5 minutos en modo activo y cada hora en modo reposo. Estos 2 modos los determina el acelerómetro. Para ello, este controla la posición del dispositivo a lo largo del tiempo de forma que cuando esta no varía y lo detecta, entra en modo reposo. Por ejemplo, cuando el paciente deposita el dispositivo en su mesilla de noche y se acuesta, el acelerómetro detecta que el dispositivo se encuentra en reposo, ya no está siendo portado por el paciente de modo que reduce la frecuencia de envío. Esto permite evitar la trasmisión de datos irrelevantes a la vez de ahorrar batería. Los envíos no se anulan para usar estos como señales de vida, pues si este dejase de enviar el cuidador no podría saber si es porque el paciente se encuentra en reposo o no lo lleva consigo, o si el dispositivo no envía porque no tiene batería y debe ser recargado. Así, el cuidador puede avisar al paciente en caso de que no quede batería o bien avisar de que se le ha olvidado el dispositivo ya que éste manda señales de vida, pero está en modo reposo.

El programa también permite la opción de recibir la posición del dispositivo según sus 3 ejes, pero se ha descartado implementarlo ya que se ha considerado innecesaria dicha información. Es irrelevante saber si el dispositivo se encuentra más o menos inclinado, más teniendo en cuenta que este se puede llevar en un bolsillo o colgado en el cuello.

Por otro lado, se ha analizado la función que modifica el estado y color del LED que hay en la parte delantera del dispositivo, pero no se ha añadido ningún cambio ya que la funcionalidad es simple y carece de un papel importante en el dispositivo, más si el cuidador va a poder conocer el estado del dispositivo en todo momento.

La figura 7 resume el flujo de envío de datos del dispositivo.

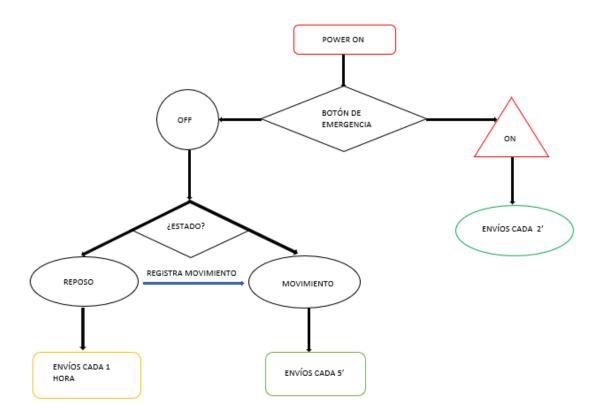


Figura 7. Diagrama de bloques envíos Dragino LGT92.

3.3 Arquitectura del sistema IoT

I. Visión general.

Se conoce como sistema internet de las cosas (IoT) a la red de objetos físicos implementada con el fin de llevar a cabo la conexión con otros dispositivos y sistemas para intercambiar datos a través de internet [Oracle 2022].

El sistema IoT del proyecto engloba desde el dispositivo Dragino que recoge los datos hasta la aplicación que los muestra. Todo ello pasando por las distintas partes encargadas de almacenar, procesar o trasmitir los datos, según se mostró en la Fig. 7.

II. Comunicación LoRaWAN

El Dragino LGT92- LoRaWAN lleva a cabo la trasmisión de datos a través de la red LoRaWAN, tal y como se indica en su nombre. El dispositivo emite por radiofrecuencia los datos a una distancia de 8 km aproximadamente en espacios abiertos, pues al tratarse de radiofrecuencias, el alcance se ve reducido al tener que traspasar muros o espacios cerrados. Sin embargo, no basta única y exclusivamente con la emisión de los datos. Estos deberán ser recopilados por un colector. Para ello se debe implementar un *gateway* de LoRaWAN que trabaje en la frecuencia del Dragino-LGT92, 868MHz, y recopile la información que este último trasmite. Al igual que sucede con el Dragino-LGT92, el *gateway* posee un alcance determinado dependiendo de la antena de escucha de la que disponga. En este proyecto se han probado distintas antenas para realizar las pruebas necesarias; no obstante, dependiendo de la situación, localización y posibilidades del paciente se deberá escoger un tipo de antena u otra, esto es, se trata de un objeto que se debe adaptar al medio donde se desarrollará la monitorización y las exigencias de seguimiento del paciente.

El *gateway* es el encargado de llevar a cabo el paso a internet. Recopila la información por la comunicación LoRaWAN y cambia el medio de trasmisión hacia la red de internet. Para ello se debe conectar a internet el *gateway* mediante las instrucciones que aporta el fabricante. Una vez realizada dicha conexión, el *gateway* debe ser registrado en la *web* TTN para almacenar correctamente los envíos que recopila el *gateway*. Se debe registrar una cuenta de TTN, registrar el *gateway* y así almacenar los datos de forma individual y segura evitando interferencias con otras antenas o dispositivos.

El Dragino-LGT92 envía los datos mediante una cadena de 18 bits que contiene toda la información del dispositivo y de los sensores de este. En la tabla 1 se muestra cómo envía la información el Dragino y en la figura 8 como se almacena ésta en la *web* de TTN.

Tamaño (bytes)	4	4	2	1	2 (opcional)	2 (opcional)	1 (opcional)	2 (opcional)
Valor	Latitud	Longitud	Alarma & batería	FLAG	Roll	Pitch	HDOP	Altitud

Tabla 1. Payload del dispositivo Dragino-LGT92 [Dragino 2021].

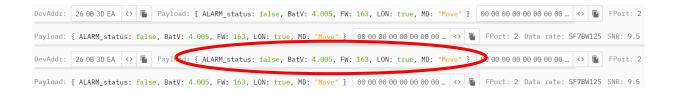


Figura 8. Almacenamiento del payload del Dragino LGT92.

Dicha información debe ser procesada ya que varios de los datos que se envían desde el dispositivo carecen de interés o son repetitivos. Por ello el *broker* MQTT almacena la información en un *string* de caracteres, selecciona la parte importante, la procesa y la almacena ordenadamente en la base de datos InfluxDB.

III. Tratamiento de los datos.

A la hora de almacenar la información es mucho más eficaz almacenar los datos de forma separada, agrupándolos según características similares, frente a ordenarlos en *strings* de infinitos de caracteres donde se encuentra toda la información sin diseccionar. Para ello se ha usado una base de datos donde se guardará la información de una forma mucho más ordenada y accesible.

Para trasmitir la información desde la *web* de TTN hasta la base de datos se desarrolla el cliente MQTT mediante el lenguaje de programación Python. El *software* desarrollado lleva a cabo la conexión a la cuenta de TTN mediante las claves e identificaciones que esta genera y pone a disposición del usuario. Una vez realizada la conexión se mantiene a la espera escuchando lo que recibe la *web* de TTN de forma que cuando llega información desde el *gateway* la procesa dividiéndola y almacenándola en distintas variables con sus respectivos valores. Dichas variables almacenan ahora la información numérica o lógica de los datos obtenidos por el dispositivo Dragino-LGT92.

Finalmente, el *software* lleva a cabo la suscripción a la cuenta de la base de datos de InfluxDB y almacena las variables con sus respectivos valores en los campos de memoria de la base de datos.

Resumiendo, con todo este proceso se comunica la información por LoRaWAN desde el dispositivo Dragino-LGT92 hasta una base de comunicaciones donde está el *gateway* receptor, se permuta la comunicación a la red de internet, se almacena la información en internet, se elimina la información desechable, se ordena la información deseada y finalmente se almacena de una forma muy accesible en una base de datos para su posterior consulta cuando el cuidador lo requiera.

En la figura 9 se puede observar cómo se procesa la información para adquirir los datos necesarios desde el *string* de caracteres para su correcto almacenamiento.

```
b'{"end_device_ids":{"device_id":"eui-70b3d57ed004e774","application_ids":{"appl
ication_id":"rodrigo-lgt-uah"},"dev_eui":"70B3D57ED004E774","join_eui":"00000000
00000000","dev_addr":"260BB0A0"},"correlation_ids":["as:up:01GD3QGP5PQ5MWPE1S1AY
6G729", "gs:conn:01GD3QFDKS9HZSTVRT9DC2VV3Y", "gs:up:host:01GD3QFDKXFD3V7DORR3YPSGQ9", "gs:uplink:01GD3QGNY2VSKN1B30HAYEZAZ0
  ,"rpc:/ttn.lorawan.v3.GsNs/HandleUplink:01GD3QGNY28BR666X236MM9MWW","rpc:/ttn.l
"","fp://tth.1orawan.v3.Gsms/aandleUplink:01GD3QGF5NHF2SVZ2E3NJ2AB8Z"],"received_at":"2022-09-
16T18:03:32.917655Z","uplink message":("session_key_id":"AXNHeCYsMVUKxRYuU3vGbg=
=","f_port":2,"frm_payload":"AAAAAAAAAAABPPWM=","decoded_payload":("ALARM_status"
:true,"BatV":4.005,"FW":163,"LON":true,"MD":"Move"},"decoded_payload_warnings":[
                                                                                                                                                                                  Alarm: True
"GPS failed to obtain location"], "rx_metadata":[{"gateway_ids":{"gateway_ids":\"codri-gateway-uah", "eui":"58AOCBFFFE801FD4"}, "time":"2022-09-16T18:03:32.614283084
                                                                                                                                                                                  BatV: 3.975
 Z","timestamp":40024460,"rssi":-37,"channel_rssi":-37,"snr":8.75,"uplink_token"
                                                                                                                                                                                  %Bat: 70
"Ch8KHQoRcm9kcmktZ2F0ZXdheS11YWgSCFigy//+gB/UEIzzihMaDaj09JKZBhDt5a/BAiDg1ZON1QE =","received at":"2022-09-16T18:03:32.640264226Z"}],"settings":{"data rate":{"lo
                                                                                                                                                                                  Latitud: 40.639412
 ra":{"bandwidth":125000,"spreading_factor":12}},"coding_rate":"4/5","frequence
                                                                                                                                                                                  Longitud: -3.226019
"867700000", "timestamp":40024460, "time":"2022-09-16T18:03:32.6142830842"), "recei ved_at":"2022-09-16T18:03:32.674863912Z", "consumed_airtime":"1.482752s", "locatio ns":{"frm-payload":{"latitude":40.513932, "longitude":-3.350349, "source":"SOURCE_
GPS"); "version_ids":{"brand_id":"dragino", "model_id":"lgtsy; "hardware_version"
:"_unknown_hw_version_", "firmware_version":"l.6.6", "band_id":"EU_863_870"); "new
ork_ids":{"net_id":"000013", "tenant_id":"ttn", "cluster_id":"eul", "cluster_addres
s":"eul.cloud.thethings.network"}}}
```

Figura 9. Procesado de datos para su almacenamiento en InfluxDB.

3.4 Programación de la aplicación

I. Visión general

La aplicación multiplataforma es la herramienta final del proyecto mediante la cual se trasmite toda la información sobre el paciente, al cuidador. Se desarrolla una aplicación multiplataforma que se ejecutará en cualquier dispositivo móvil para lograr una monitorización a tiempo completo y real gracias al uso de una de las herramientas más cotidianas en la actualidad: el teléfono móvil. De este modo, el cuidador solo deberá hacer uso de su teléfono para ejecutar la aplicación y conocer el estado del paciente de forma instantánea y en tiempo real, así como recopilar el histórico que puede permitir detectar cambios en rutinas diarias.

La aplicación será la encargada de hacer las solicitudes a la base de datos de InfluxDB donde se almacena la información para inmediatamente mostrársela al cuidador de la manera más sencilla y accesible posible aprovechando la infinidad de posibilidades que brinda cualquier *smartphone*.

Disponer de una aplicación multiplataforma brinda al usuario importantes mejoras en la visualización de los datos. Se pueden representar sobre un mapa las coordenadas del dispositivo, el estado de la batería, tener un sistema de notificaciones instantáneo y eficaz, etc.

la aplicación. INICIO **REGISTRO** CREAR **ACCESO**

A continuación, en la figura 10, se muestra un flujograma básico del funcionamiento de

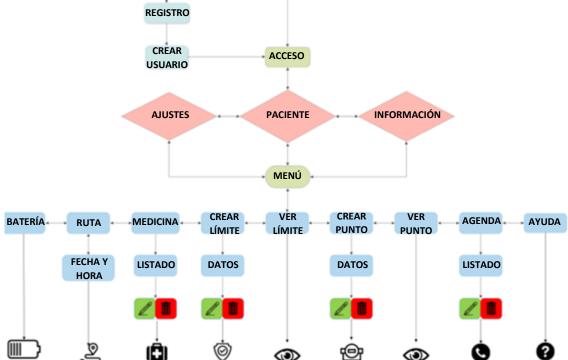


Figura 10. Esquema de la aplicación móvil desarrollada.

II. Gestión de datos

Durante la monitorización del paciente se recogen múltiples datos que deberán ser seleccionados para un muestreo eficaz, pues no todos ellos serán útiles en un momento determinado o su información carecerá de valor transcurrido un tiempo. Por eso, el proceso del muestreo de la información requiere de un estudio previo de la información de la que se dispone, identificando cual puede ser interesante para el usuario, cual imprescindible y cual carece de interés.

Al ejecutarse la aplicación se van haciendo solicitudes a la base de datos o queries para que la base de datos trasmita la información deseada a la aplicación, ésta la gestione internamente y la muestre por pantalla de la manera más rápida e intuitiva posible. Mediante las *queries* se selecciona la cantidad de información que se solicita la base de datos para que se trasmita únicamente la información necesaria y acelerar el proceso de análisis y presentación de dicha información.

Por ejemplo, uno de los datos que se recopila es el nivel de batería del dispositivo para alertar al cuidador cuando la batería se esté agotando y que éste avise al paciente o bien cargue él mismo la batería. A la hora de solicitar esta información y representarla no resulta necesario leer el registro entero de los valores que se han recogido sobre el nivel de la batería. Por ello, al desarrollar la *query* sobre el estado de la batería se implementa una solicitud del último valor registrado en la base de datos y cuando el servidor lo comunica, se representa gráficamente o incluso se emite una alerta si es necesario.

De este modo se agiliza la visualización de los datos optimizando el proceso de solicitud y envío de datos. Evidentemente, cada tipo de dato posee unas características particulares que requieren unas *queries* u otras. Para ello se analiza cada tipo de dato y se desarrolla la solicitud a la base de datos más adecuada para cada situación. En algunos casos, también se permite al usuario la posibilidad de configurar dichas *queries* indirectamente ofreciéndole la posibilidad de escoger el rango de datos que desea utilizar. Esto puede resultar útil, por ejemplo, si el cuidador quiere conocer dónde ha estado el paciente en un rango de tiempo determinado y no sólo la última posición del mismo o un rango predefinido que quizá sea de interés.

III. Registro de cuidadores

En primer lugar, se lleva a cabo el registro de los cuidadores asociados al dispositivo que se le otorga al paciente. De este modo solo el propio cuidador puede acceder a la información de su paciente. Esta función se desarrolla con la finalidad de otorgar privacidad y seguridad al paciente. Además, esto permite evitar interferencias entre la información de un paciente u otro. Una vez registrado el cuidador, éste deberá acceder a la aplicación con un nombre identificador y una contraseña determinados por él mismo durante su registro.

Esta funcionalidad se desarrolla de una manera simple debido a que en la actualidad se dispone de un único dispositivo de monitorización. Más adelante, se deberá restringir cada dispositivo y todos sus datos con su correspondiente clave en una base de datos mucho mayor que almacene la información de todos los dispositivos y ésta esté asociada a un cuidador determinado.

A continuación, en la figura 11 se muestran las pantallas diseñadas para este proceso de registro y en la figura 12 el acceso.



Figura 11. Pantalla inicio (izq.) y pantalla registro del cuidador (dcha.).



Figura 12. Pantalla registro del paciente (izq.) y pantalla acceso (dcha.).

IV. Monitorización del recorrido

El dispositivo Dragino LGT-92 dispone de la tecnología GPS, lo cual permite tener localizado el dispositivo en espacios exteriores con visibilidad de los satélites. Gracias a la API de Google Maps se puede representar en un mapa las coordenadas emitidas por el dispositivo. Esta API es de pago, pero posee un cierto periodo de prueba que permite usar dicha API antes de contratarla si se está satisfecho con sus servicios. Se ha escogido esta API ya que es la más familiar para cualquier usuario en la parte de representación en mapas.

En el menú inicial de la aplicación se carga la última posición del dispositivo ya que es la información más relevante del paciente y así se le puede tener ubicado nada más arrancar la aplicación.

Por otro lado, se ofrece la posibilidad de conocer los puntos en los que ha estado el paciente en un periodo de tiempo determinado por el cuidador. Para cada ubicación se marcará y mostrará la hora a la que se registró la misma. Para poder hacer más intuitiva esta información se ofrece al usuario la opción de trazar una estimación de cómo ha podido ser la ruta realizada entre dichos puntos y conocer así por donde se ha ido moviendo el paciente en el periodo de tiempo solicitado por el cuidador.

Esta ruta es una estimación ya que la API de Google Maps unirá los puntos sucesivos mediante las rutas óptimas, de modo que si el paciente ha ido de un punto a otro por el camino más largo no se podrá conocer exactamente. Este error es un problema menor debido a la frecuencia de envío de las coordenadas y ya que se estima que el paciente se moverá a pie, de forma que el fallo en la ruta puede ser entre calles próximas y el interés real es el de conocer por qué áreas se ha movido o si hay cambios significativos en los trayectos de unos días a otros, más allá de conocer con gran exactitud la ruta realizada.

La API de Google Maps, también permite ampliar o disminuir el *zoom* del mapa, así como moverse por él. Se ofrece la posibilidad de manejar los distintos tipos de mapa: estándar, satélite y relieve. También se ofrece la posibilidad de situarnos sobre el mapa y tener una visualización real del entorno, lo cual es muy útil a la hora de conocer dónde está o ha estado el paciente y si puede suponer un riesgo para su salud.

De la misma manera, Google Maps recoge los distintos negocios o lugares de interés que se encuentran sobre el mapa. Esto es muy interesante en ciertos casos de monitorización en los que se desee que el paciente no se acerque a lugares inadecuados para sí mismo como pueden ser los bares para personas con problemas de alcoholemia.

V. Definición de alarmas

En este proyecto se ha optado por la comunicación de la información al cuidador mediante la tecnología que facilitan los *smartphones*. Esta decisión ha sido motivada por la simbiosis actual entre el ser humano y la telefonía móvil. Actualmente España es el país líder en Europa en cuanto a la penetración de los *smartphones* por habitante. A día de hoy, un 55,2% de los españoles posee un teléfono inteligente [Esendex, 2022]. Este porcentaje se ve incrementado a medida que se descartan las personas de la tercera edad, pues para este sector de la población el manejo de los teléfonos inteligentes es una tarea complicada. No obstante, se trata de la herramienta ideal para la recopilación de la información por parte del usuario o cuidador.

Como ya se ha ido comentando, es precisamente el sector de la población de la tercera edad, el sector qué más va a requerir de la monitorización y, por tanto, es el sector que menos actuará como cuidador. Esto hace que el porcentaje de potenciales cuidadores con teléfono inteligente propio sea bastante elevado. Además, cabe destacar, que el mercado de la telefonía móvil es un mercado extendido que ofrece distintas opciones para el usuario a precios muy asequibles eliminando así las posibles barreras económicas a este proyecto.

Sin duda, otro de los factores clave en la elección del *smartphone* es la simultaneidad de sus notificaciones o alertas. Cualquier aplicación puede informar de manera instantánea acerca de cualquier suceso. Esto permite avisar al cuidador rápidamente de cualquier tipo de alerta.

Cada alerta dispondrá de su nivel de importancia dependiendo del suceso que sobre el que informe. La importancia de las alertas se verá reflejada en su modo de actuación, de modo que las alertas más importantes utilizarán más recursos para atraer la atención del usuario o cuidador.

1) Alerta severa: si el paciente mantiene presionado el botón de emergencia, está indicando una alerta severa que debe ser transmitida inmediatamente al cuidador. Se trata de una alerta que no puede cesar hasta que el cuidador haya confirmado la recepción de la información. Por ello, se configurará una alerta sonora que no cesará hasta que el cuidador no presione un botón indicando haber sido notificado acerca de la situación. Además, se encenderá la pantalla con un símbolo de emergencia a la vez que el teléfono suena y vibra.

Por otro lado, existen otro tipo de alertas menores que podrán ser configuradas por el usuario indicando el grado con el que se transmitirá dicha alerta.

- 2) Alerta normal: el usuario podrá definir cómo desea ser informado si el paciente abandona unos límites seguros dentro de su localización. Por defecto se notificará con una alerta sonora de un tiempo reducido.
- 3) Notificación: cuando la batería del dispositivo sea menor al 20% se avisará al cuidador con una notificación sencilla

También se debe tener en cuenta cuándo enviar dichas notificaciones o alertas. Una vez que el cuidador ha indicado que ha recibido la alerta causada por el botón de emergencia, esta alerta cesa, aunque el Dragino siga enviando el aviso durante un tiempo por seguridad. Algo similar sucede con el estado de la batería, lógicamente esta seguirá siendo menor al 20% hasta que sea cargada de nuevo. Sin embargo, en este caso seguramente se deba recordar el estado de la batería cada cierto periodo de tiempo antes de que éstas terminen de agotar definitivamente si el cuidador olvida cargarla o que el propio cuidador pueda indicarle al paciente la necesidad de cargar la batería.

VI. Interfaz gráfica

La aplicación multiplataforma desarrollada en este proyecto para monitorizar personas con déficits cognitivos leves se implementa para mostrar la información recopilada al cuidador. Por ello, la aplicación debe ser los más sencilla y práctica posible para una trasmisión de la información precisa.

Se desarrolla una interfaz gráfica lo más amigable con el usuario posible. Se trata de representar toda la información gráficamente para lograr descifrar la información rápidamente y que esta parte no sea monótona y aburrida como podría ser si solo se trasmitiese mediante texto. Se asocian distintos elementos gráficos con cada tipo de dato recopilado y se hace uso de estos elementos gráficos para mostrar la información deseada por el usuario.

En primer lugar, se accede a un menú de registro donde se indican los campos que se deben rellenar y estos adquieren un color verde al ser rellenados correctamente (véase la Figura 11). Tras rellenar todos ellos, el color del botón de registro cambia a un color más sólido indicando que el registro se puede completar. Así se procede también con la pantalla de inicio de sesión o ingreso.

Una vez registrado, se accede a la pantalla del menú principal (véase la figura 12) donde se muestra en un mapa la última ubicación recibida desde del dispositivo y se indica la fecha y hora para que el cuidador conozca cuando ha sido el último envío de dichos datos por parte del dispositivo. Esto permite determinar si la información recibida es relevante o si por el contrario es demasiado antigua.

De este modo, además de conocer inmediatamente la ubicación del paciente, el cuidador puede conocer cuánto lleva el dispositivo sin enviar datos pudiendo conocer así si está apagado o fuera de conexión.

Esta pantalla se puede refrescar deslizando hacia abajo. Al refrescar la pantalla se vuelve a hacer una llamada a la base de datos para revisar si existen nuevos datos y actualizar la geolocalización del dispositivo Dragino LGT92. Además, esta pantalla se actualiza automáticamente. El usuario puede definir en ajustes cada cuantos minutos desea dicha actualización automática.

En la figura 13 en la pantalla paciente (dcha.) puede verse como en la parte superior izquierda se muestra un botón que despliega un menú donde aparecen una serie de botones que dan acceso al resto de la información. Cada botón lleva consigo un emoticono que describe el tipo de información a la que conducirá dicho botón. Esto permite disponer de un menú inicial muy sencillo y visualizar rápidamente la información más relevante. Dicho menú también se podrá desplegar deslizando hacia la derecha.

En la esquina superior derecha se muestra un botón con el símbolo (?) que al pulsarlo muestra una ventana con la información acerca de la pantalla actual para ayudar al cuidador con el uso de la aplicación.

Deslizando hacia abajo se puede actualizar el mapa y por tanto la ubicación del paciente.

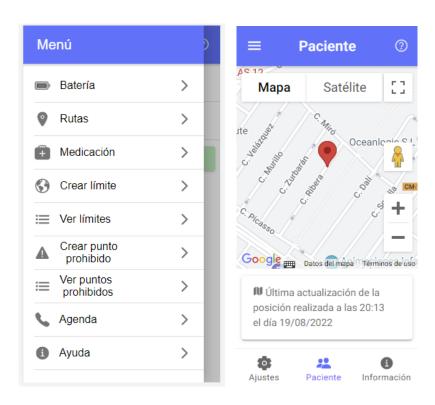


Figura 13. Pantalla menú opciones (izq.) y pantalla paciente (dcha.).

A la izquierda de la pantalla de inicio se encuentra la pantalla de ajustes con las distintas opciones que se ofrecen para que el usuario pueda configurar determinadas características de la aplicación a su gusto. Se pueden escoger el tipo de notificaciones: vibración y sonido, solo vibración o nada. También se puede escoger cada cuantos minutos se actualizará automáticamente la pantalla principal donde se puede ver la ubicación actual del paciente. En la figura 14 se muestra la pantalla de ajustes con algunos de los ajustes opcionales.

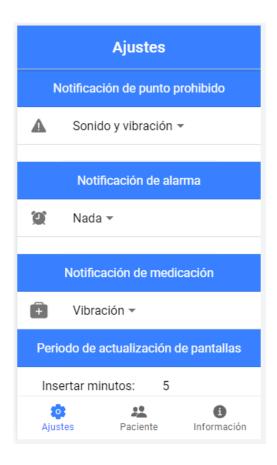


Figura 14. Pantalla ajustes.

A la derecha se encuentra la pantalla de información donde se detallan las distintas utilidades de la aplicación, su manejo y los desarrolladores (véase la figura 15). Esto permite al usuario conocer todo acerca del manejo de la aplicación de una manera rápida y sencilla.



Figura 15. Pantalla de información.

Para facilitar la transmisión de la información, se emplean diversos elementos gráficos. En primer lugar, se hace uso del mapa para mostrar la ubicación del paciente o incluso se traza una línea en el mapa para conocer la ruta que ha seguido. Este método de representación es mucho más eficaz y útil que mostrar una tabla con la latitud y longitud del paciente, que puede ser más ardua de interpretar para el cuidador.

Algo similar sucede con la batería. Es mucho más rápido de entender una imagen donde se pueda ver el nivel de la batería del dispositivo que usar texto para conocer el estado de esta. Sin embargo, en este caso un simple emoticono puede ser insuficiente.

Así, se ha dotado a la imagen de color: el color verde indica un estado de carga adecuado o completo. Por el contrario, se emplea un color rojo cuando se disponga de un 10% o menos de batería. Se emplea el amarillo cuando la batería es un 30% o menos. De este modo el cuidador recibe rápidamente cuando debe empezar a estar pendiente de la batería o incluso cargarla cuanto antes. La figura 16 muestra varios ejemplos de funcionamiento para distintos niveles de batería.

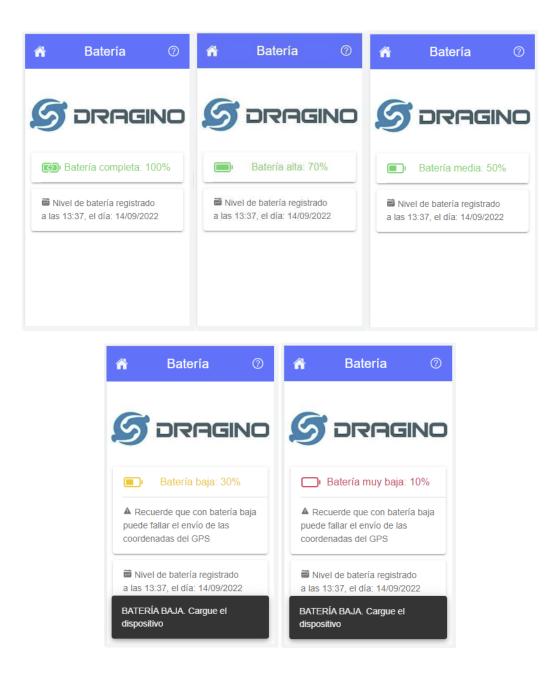


Figura 16. Pantallas con distintos niveles de batería.

Este tipo ayudas a la hora de expresar información también se añade a las notificaciones como la del aviso del botón de emergencia. La aparición de un símbolo de emergencia en rojo dota de un mayor poder visual a la notificación que si solo se expresase por texto la alerta.

Además, también existe una opción que permite dibujar en el mapa todas las posiciones del paciente en un periodo especificado por el cuidador. De este modo el cuidador puede controlar donde ha estado el paciente sin tener que estar revisando la última posición del paciente de forma continua. A la hora de mostrar todas las ubicaciones, estas se filtran para evitar saturar el mapa de marcadores innecesarios, es decir, si el paciente no se mueve más allá de unos 300 m, aproximadamente, se descarta la ubicación ya que con el marcador anterior ya se muestra que el paciente ha estado en dicho punto.

La figura 17 muestra un ejemplo donde se han solicitado las posiciones del paciente el día 19 de agosto entre las 03:12 y las 20:12. Se puede observar que no hay puntos repetidos ya que estos son eliminados al no aportar información nueva.

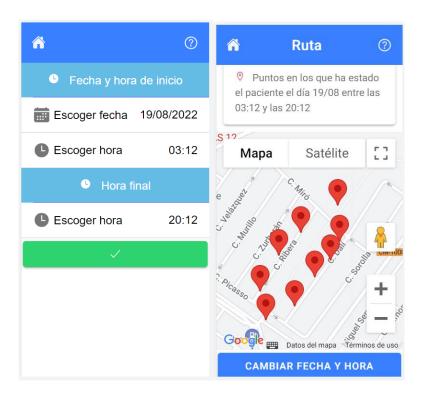


Figura 17. Pantalla selección de periodo (izq.) y pantalla de ubicaciones (dcha.).

Para evitar confusiones con los puntos marcados sobre el mapa, si el usuario pulsa sobre el marcador se muestra la hora en la que el paciente estuvo en ese punto. En la figura 18 se muestra este funcionamiento en un ejemplo con una ruta con 3 puntos marcados. Se pulsa sobre dos de los tres marcadores y se obtiene la hora a la que el paciente estuvo en ambos puntos.

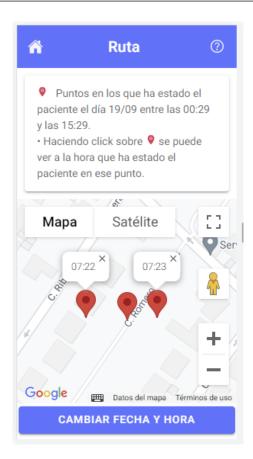


Figura 18. Pantalla de la ruta seguida, marcadores con información.

Además de la información que recoge el dispositivo Dragino-LGT92 se ha decidido dotar al usuario de la posibilidad de almacenar en la *app* distintos datos que pueden ser útiles a la hora de cuidar de un paciente con deterioros cognitivos.

En primer lugar, se crea un apartado con nombre "Medicación" que permitirá al usuario registrar los distintos medicamentos que debe tomar el paciente, la fecha de inicio y fin del tratamiento, cada cuántas horas se debe tomar el medicamento y se permite registrar una breve descripción del medicamento para facilitar el reconocimiento del medicamento al cuidador. Esto facilitará al cuidador llevar un seguimiento más preciso y sencillo de la medicación del paciente, pudiendo ponerse en contacto con este si fuera necesario asegurarse de que la pauta médica está siendo seguida por el paciente. Además, al registrar las pautas médicas, el usuario podrá luego configurar unas alertas por si desea recibir notificaciones cada vez que el paciente tenga que tomar la siguiente dosis.

En la figura 19 se muestra cómo funciona la pantalla donde se puede ver la lista de medicamentos a tomar y como se deben tomar y la pantalla mediante la cual el cuidador puede añadir una pauta de medicación a dicha lista.

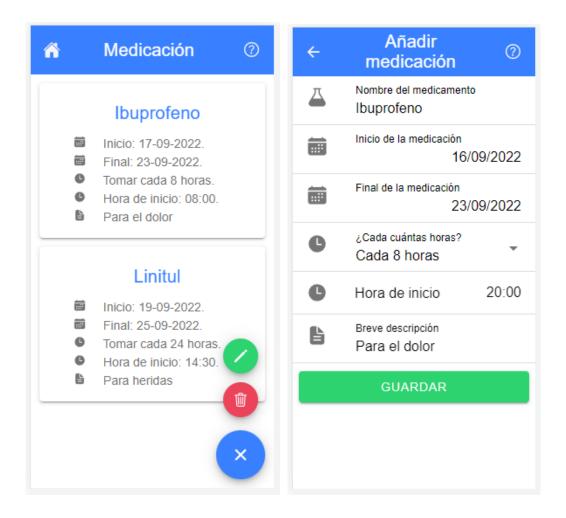


Figura 19. Pantalla listado de pautas médicas (izq.) y pantalla registro de pauta médica (dcha.).

Por otro lado, se permite al cuidador registrar unos límites seguros. Estos límites seguros son áreas donde el paciente puede estar sin peligro. Si el paciente sale de estos límites, se genera una alerta al cuidador. Los límites seguros se establecen desde un punto central y un radio determinados por el cuidador. El punto central se podrá establecer introduciendo las coordenadas deseadas por el usuario o bien aprovechando la geolocalización del dispositivo Dragino-LGT92.

Estos límites se almacenan en una base de datos y se pueden eliminar si se desea desecharlos en un momento determinado.

Al generar el límite se muestra al usuario el centro seleccionado y posteriormente, como queda el límite establecido antes de confirmar su almacenamiento para que el cuidador pueda visualizar sobre un mapa el límite creado.

A continuación, en la figura 20 se puede apreciar el proceso de creación de un límite seguro para el paciente.

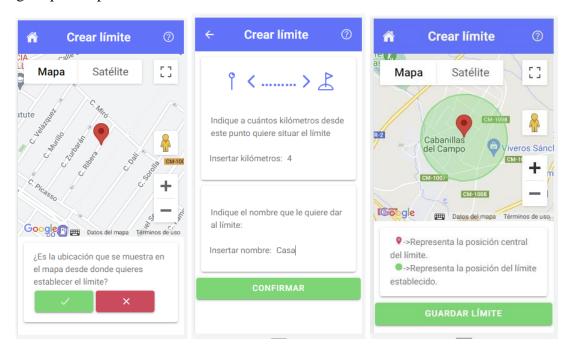


Figura 20. Pantalla selección del centro del límite (izq.), pantalla de selección de radio y nombre del límite (centro) y pantalla de confirmación del límite (dcha.).

Del mismo modo, se ofrece al usuario la posibilidad de establecer puntos prohibidos en el mapa. Estos puntos prohibidos son lugares donde el cuidador desea que no se acerque el paciente. Puede ser útil para personas con dependencias que se desea mantener alejados de los lugares que les generan dichas dependencias, véase casino en ludópatas o bien bares en personas con problemas de alcoholemia.

Los puntos prohibidos son generados de la misma forma que los límites. Se pueden registrar bien por localización o bien por introducción de las coordenadas.

Una vez registrado el punto prohibido se establece una notificación con el nivel de alerta que desee el usuario cada vez que el paciente se encuentre a menos de 200 m de dicho punto.

Esta funcionalidad, al igual que la de los límites, presenta un problema y es la periodicidad de envío del dispositivo Dragino-LGT92. En el peor de los casos, la alerta no será generada hasta 5 minutos pasado el límite debido a que ese es el periodo de envío. Mientras no se recibe un envío no se actualiza la posición del paciente y, por tanto, no se analiza la misma. Estos 5 minutos de periodo de envío son configurables, sin embargo, se considera un tiempo que puede ser suficiente para permitir al cuidador actuar acorde a la información recibida. Aumentar la frecuencia de envío podría reducir mucho la duración de la batería del dispositivo de modo que se ha desechado dicha opción y se ha considerado el periodo de 5 minutos adecuado.

En la figura 21 se muestra la pantalla elaborada para crear un punto prohibido. Si el paciente se acerca a dicho punto, el cuidador será notificado.



Figura 21. Pantalla creación del punto prohibido.

Según se adelantaba anteriormente, se ha creado una pantalla donde el cuidador puede almacenar las pautas de medicación del paciente. De este modo, es más sencillo llevar un orden con la medicación del paciente o incluso recordarle al mismo las dosis que debe injerir cada día. El cuidador debe recoger el nombre del medicamento, la fecha de inicio de la medicación, la fecha final de la misma y cada cuantas horas de debe recibir dicha medicación. Además, se ofrece al usuario la opción de escribir una breve descripción del medicamento. Posteriormente se muestran las medicaciones que debe tomar el paciente cada día y se anulan automáticamente las pautas finalizadas una vez ha pasado el tiempo de medicación indicado.

En todo momento se pueden añadir o eliminar rutinas de medicación por parte del usuario. La figura 22 es un ejemplo de la pantalla de recordatorio de la pauta de medicación configurada.



Figura 22. Pantalla pautas médicas.

Por último, se pone a disposición del usuario una agenda telefónica. Dicha agenda telefónica viene con el teléfono de emergencias (112) por defecto y permite realizar llamadas instantáneas en caso de emergencia. Al disponer de esta agenda telefónica, el cuidador puede añadir un listado de teléfonos a los que puede acceder rápidamente y realizar la llamada desde la misma agenda de la forma más rápida posible, lo cual puede ser importante en ciertas situaciones de emergencia.

Se dispone de un buscador para lograr una mayor rapidez, de la posibilidad de agregar y eliminar contactos. Sin embargo, el uso ideal de esta funcionalidad es agregar unos pocos números relacionados con la situación médica en la que se centra este proyecto para lograr una rapidez óptima en situaciones de emergencia.

En la figura 23 se muestra la pantalla de dicha agenda telefónica creada para situaciones relacionadas con el cuidado del paciente.

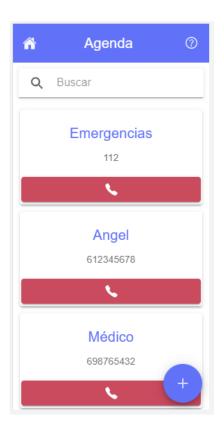


Figura 23. Pantalla agenda telefónica.

CAPÍTULO 4: RESULTADOS

4.1 Introducción

Una vez que se han llevado a cabo todos los procesos de este proyecto (estudio del problema, selección de herramientas para su solución, reconocimiento del funcionamiento del Dragino-LGT92, implementación y uso del servidor MQTT ya integrado en TTN, implementación de una base de datos y programación de la aplicación), se procede a realizar las pruebas necesarias para validar los resultados del mismo.

En primer lugar, las pruebas se realizan con una pequeña antena LoRaWAN pensada para espacios interiores y que únicamente permite desplazar el dispositivo Dragino-LGT92 unos 300 m de donde se establece la antena. Se trata de la antena *The Things Indoor Gateway*. Es una antena diseñada para ser un *gateway* LoRaWAN de bajo coste, configuración sencilla, con frecuencia central de 868 MHz y entre 300 y 500 m de alcance dependiendo del entorno [Antena TTN 2022]. Esto permite registrar datos en las calles adyacentes de donde se encuentra la antena. En entornos rurales sin tantos edificios, se puede esperar un rango algo mayor.

Posteriormente, se hará uso de una antena (LORIX One [LORIX 2022]) de mayor alcance que permite una mayor lejanía y hacer pruebas con los límites que se establecen sobre el paciente. LORIX One es una antena LoRaWAN que lleva el sistema operativo LORIX OS incluido, el cual actualiza periódicamente los propios parches de seguridad de la antena. Se trata de un sistema operativo totalmente abierto y, por tanto, personalizable. Posee una memoria RAM de 128 MB y un almacenamiento interno de 512 MB. Dispone de un ARM Cortex-A5 600 MHz en su hardware. Estas características ofrecen un buen rendimiento para poder almacenar datos a una distancia mayor, 2 km aproximadamente; y se podría trabajar con varios dispositivos si en un futuro así se desea [LORIX 2022]. Esta última prueba permite simular el funcionamiento real de la red de comunicación y el Dragino para lograr el objetivo propuesto al inicio del proyecto.

4.2 Pruebas experimentales

Inicialmente se realiza una sencilla prueba cerca de la antena. Se enciende el Dragino-LGT92 y se comprueba el funcionamiento del servidor MQTT, el almacenamiento en la base de datos de InfluxDB y finalmente la recepción y muestreo de los datos en la aplicación. Se comprueba que los datos recibidos (nivel de batería, localización por GPS y estado del botón de alarma) son correctos. Para ello se pone el dispositivo Dragino LGT-92 en funcionamiento, se espera a que se vayan recibiendo los datos enviados desde el mismo, se leen en la base de datos y se comprueba que la información emitida finalmente en la *app* sea correcta. Esto se realiza mostrando por consola los datos recibidos y comprobándolos. Posteriormente, se pulsa el botón de emergencia, se espera que la alarma se muestre con el valor "*True*"; se realiza la prueba

con el dispositivo totalmente cargado para obtener una batería del 100% y se comprueba que las coordenadas de latitud y longitud se correspondan con la localización del dispositivo desde donde se realiza la prueba.

En la figura 24 se muestran los valores enviados por el dispositivo. Dichos valores son a su vez almacenados en InfluxDB (véase figura 25).

```
Alarm: True
BatV: 4.0
%Bat: 100
Latitud: 40.51467
Longitud: -3.34949
```

Figura 24. Datos recogidos con el *script* MQTT cuando se ha pulsado el botón de emergencia.

```
_value _field __measurement

100 nivel_bateria% datos

40.51 latitud_gps datos

1 boton_emergencia datos

-3.349 longitud_gps datos
```

Figura 25. Datos almacenados en InfluxDB.

Se puede apreciar que los datos almacenados en InfluxDB son correctos. Es cierto que la latitud y longitud han sido mostradas con menos decimales, sin embargo, en el archivo .csv que almacena InfluxDB los datos están completos tal y como se han recogido mediante el *script* de Python para arrancar el servidor MQTT.

Una vez que se comprueba que todo este proceso funciona adecuadamente, se continúa haciendo pruebas con las distintas funcionalidades que se han ido añadiendo a la aplicación para mejorar la experiencia del usuario y explotar al máximo la información que se recibe a través del Dragino.

En primer lugar, se trabaja con el apartado de registro y acceso a la aplicación. En esta primera versión, se muestra un sencillo sistema de registro donde se requieren datos propios del usuario, del dispositivo y del paciente. Inicialmente, se almacena el *email* y contraseña del usuario para requerirlos y comprobarlos durante el acceso. Además, se recogen los identificadores del dispositivo para más adelante asociar cada cuenta con su dispositivo o dispositivos si se desea controlar más de un paciente. De este modo, cada usuario tendría acceso únicamente a los dispositivos que haya adquirido para evitar problemas de privacidad.

En la figura 26 se muestran las cuentas almacenadas y sus respectivas contraseñas en un mismo campo con el formato "email**contraseña", las cuales coinciden con los registros realizados. Además, también se puede observar que, al introducir las credenciales de forma errónea, no se permite el acceso y aparece un mensaje avisando al usuario de que el correo electrónico o la contraseña son erróneos.



Cuentas creadas:

rodri@uah.es**123456

prueba@gmail.com**a1b2c3d4

Figura 26. Cuentas registradas y pantalla de acceso denegado.

Por otra parte, se crean los límites y puntos prohibidos asegurando que la información confirmada se almacena de forma adecuada en la base de datos que la aplicación irá leyendo para mostrar las alertas o notificaciones necesarias cuando la situación así lo requiera. Se realiza la prueba de salir del límite y acudir al punto prohibido para visualizar la correspondiente alarma. (ver figura 27).



Figura 27. Alarma por salir de un límite (dcha.) y alarma por entrar en un punto prohibido (izq.)

Se comprueba el funcionamiento de la agenda agregando algún número de teléfono personal y pulsando el botón de llamar para comprobar que funciona correctamente. Se demuestra que, efectivamente, se trata de un servicio mucho más eficaz que una agenda donde se pueden tener guardados muchos teléfonos de ámbitos muy diferentes, y a los que puede costar acceder en situaciones de alto estrés como puede darse al detectar una alarma de este tipo.

En la figura 28 se muestra como el contacto "Ángel" recibe una llamada al pulsar el botón rojo de llamar:

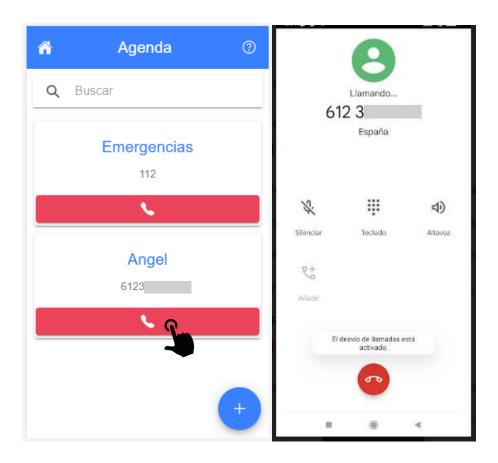


Figura 28. Funcionamiento de agenda telefónica.

Del mismo modo, se verifica que se almacenen correctamente en la base de datos de InfluxDB con el formato:

"nombre**Fecha_inicioaaFecha_finalbbPeriodoccDescipcionddHora_Inicio" las distintas pautas médicas que se incluyan o eliminen en la pantalla habilitada para ello.

Este tipo de formatos se usa para almacenar en un solo *string* toda la información y poder trocearla cuando sea necesario.

En la figura 29 se muestran las pautas médicas registradas y su almacenamiento en la base de datos.



Ibuprofeno**17-09-2022aa23-09-2022bb8ccPara el dolordd08:00 Linitul**19-09-2022aa25-09-2022bb24ccPara heridasdd14:30

Figura 29. Funcionamiento de la lista de medicación.

Finalmente, al haber comprobado todos los procesos de almacenamiento y lectura de la información, presentación de los datos y funcionamiento correcto de la aplicación, se procede a realizar las pruebas de distancia con el sistema de comunicación LoRaWAN.

Para ello se conecta el receptor a la red de internet y se va alejando el Dragino-LGT92 de él. Esto permite además obtener distintas posiciones de la localización GPS y comprobar que se pueden registrar varias posiciones en un periodo de tiempo a escoger por el usuario.

Esta prueba se realiza en un entorno urbano, lo cual no es lo más adecuado para la comunicación LoRaWAN ya que esta funciona mejor en espacios abiertos.

Se consigue mandar información dando una vuelta por las calles adyacentes (ver figura 30) y mostrando las distintas posiciones en el mapa. Al realizar esta prueba se detecta que en ocasiones se envían posiciones muy similares en un breve espacio de tiempo (ver figura 31). Así, se decide reducir la representación de dichas posiciones exigiendo una mínima distancia entre la posición a representar y la anterior para evitar muestrear muchas veces una misma posición en los momentos en los que el paciente no se mueva (ver figura 32).

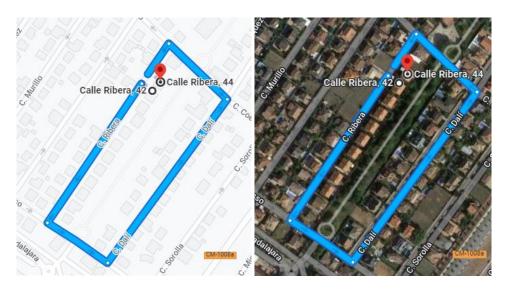


Figura 30. Ruta realizada, vista en mapa (izq.) y en satélite (dcha.).

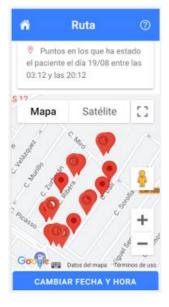


Figura 31. Visualización del mapa con puntos muy similares.

(Puntos a ser eliminados, con hueco libre en su interior)

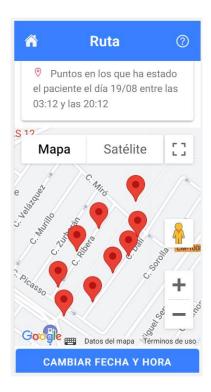


Figura 32. Visualización del mapa una vez eliminados los puntos muy similares.

Debido al periodo de envío que tiene programado el Dragino-LGT92, se decide que la opción de revisar las posiciones en un periodo de tiempo sea restringida a unas horas determinadas, es decir, el cuidador deberá escoger el día y las horas entre las que quiera conocer las posiciones del paciente. Esto le permitirá no tener que estar constantemente chequeando la posición del paciente. Además, si el cuidador sabe que a una determinada hora el paciente debe encontrarse en un punto concreto (médico, tienda, etc.) lo puede comprobar mediante esta opción o revisando la ubicación en tiempo real en la pantalla principal, funcionalidad que también se revisa en este ensayo ya que ahora el dispositivo se va desplazando y se puede comprobar que, deslizando en la pantalla hacia abajo, la posición se actualiza adecuadamente cada vez que se recibe la información desde el Dragino-LGT92.

Durante este experimento se descarta dibujar un trazado que vaya marcando la ruta del paciente, pues Google Maps crea el trazado óptimo entre puntos, pero no tiene por qué ser el trazado real. También se ha tenido en cuenta que representar el trazado entre muchos puntos ralentiza excesivamente el muestreo de la información para presentar datos que no pueden ser muy precisos por el motivo mencionado.

Por todo ello, se considera que mostrando los puntos donde ha estado el paciente, el cuidador puede conocer la situación del paciente e intuir las posibles rutas de este configurando el periodo de muestreo adecuadamente.

En todo momento, ya sea con la aplicación abierta o no, se ejecutan lecturas constantes a la base de datos para conocer si se debe dar información al usuario acerca del estado del paciente (alarma y posición GPS) o de la batería del dispositivo. Si se detecta la necesidad de notificar al usuario acerca del paciente, se interrumpe toda funcionalidad de la aplicación para comunicar ese aviso. Para notificar batería baja o pauta médica tan solo se emite una notificación en la barra superior (ver figura 33), pues no se considera una notificación severa. Sin embargo, si el paciente pulsa el botón de auxilio, sale del límite establecido o llega a un punto prohibido, se enciende el teléfono del cuidador y se emite una alarma sonora con un funcionamiento similar a un despertador hasta que esta alarma sea desactivada por el cuidador. La alarma se desactiva pulsando el botón "Ver posición" el cual redirecciona a la pantalla principal donde se puede ver la posición actual del paciente. En la figura 34 se muestra la pantalla en el caso de alerta severa, y se muestra cómo se indica si dicha alerta es a causa de la posición del paciente o bien si este ha pulsado el botón de auxilio.



Figura 33. Pantalla de notificaciones



Figura 34. Pantalla de alarma por botón de auxilio.

Tras concluir los ensayos iniciales con la antena pequeña. Se realiza una prueba con mayor distancia en la Escuela Politécnica Superior de Alcalá de Henares. Para ello se sitúa el dispositivo en el bolsillo del paciente (ver figura 35) y se realiza una ruta por los alrededores de la facultad. (El dispositivo también se puede llevar colgado al cuello con la cuerda que ofrece el fabricante, ver figura 36).



Figura 35. Dispositivo situado en el bolsillo del paciente.



Figura 36. Dispositivo colgado al cuello.

Se inicia la ruta indicada en la figura 37 y se va comprobando como se actualiza la posición que aparece en la pantalla principal de la aplicación con los distintos puntos recogidos. Tras finalizar la ruta, se visualiza que los puntos indicados en la pantalla de visualización de rutas son coherentes con el trayecto recorrido (ver figura 38).

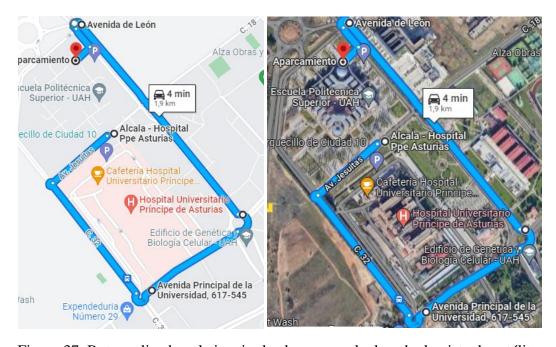


Figura 37. Ruta realizada, a la izquierda el mapa y a la derecha la vista de satélite.



Figura 38. Visualización de los puntos registrados durante la ruta.

Por último, una vez registrados los datos de una pequeña ruta y con distintas posiciones almacenadas en la base de datos de InfluxDB, se procede a permitir el uso de la aplicación a una persona ajena al desarrollo de la misma para comprobar que, efectivamente, se trata de una aplicación de uso sencillo y con plenitud de pantallas de ayuda si el usuario lo necesita en algún momento. Tras esta última prueba, se realizan unas pequeñas modificaciones para optimizar la sencillez del uso de la aplicación a partir de la experiencia aportada por dicho usuario ajeno al proyecto y a la programación de la aplicación.

4.3 Conclusiones

Tras la realización de todas las pruebas experimentales se procede a hacer una valoración de la primera versión de la aplicación desarrollada y proponer posibles mejoras o implementación de nuevas funcionalidades en versiones futuras.

Se ha realizado un ensayo a menor escala (antena de poco alcance) que ha permitido hacer ensayos básicos donde no se requería de gran movilización del dispositivo y se ha podido trabajar de forma más cómoda para llevar a cabo las primeras pruebas y modificaciones.

Una vez que se han superado estas pruebas básicas e iniciales, se ha realizado una prueba con una antena de mayor alcance, más en línea con una situación real sobre la que se enfoca este proyecto. De este modo se han sacado las últimas conclusiones y se han aportado las soluciones definitivas al desarrollo de la aplicación móvil.

Los resultados obtenidos han sido bastante satisfactorios, pues se considera el objetivo inicial cumplido, se ha desarrollado una aplicación de uso sencillo que permite al cuidador estar informado acerca de la situación del paciente tal y como se ha podido comprobar en los ensayos mencionados anteriormente.

CAPÍTULO 5: CONCLUSIONES Y LÍNEAS FUTURAS

5.1 Conclusiones

Los resultados de esta primera versión muestran que se han logrado los objetivos establecidos al inicio del TFG.

En primer lugar, se ha desarrollado un servidor MQTT que ha permitido recoger toda la información que trasmitía el dispositivo por LoRaWAN. De esta manera se ha logrado hacer efectivo el sistema de envío y recepción de la información para posteriormente almacenarla en TTN. Todo este sistema de comunicación mantendrá informado en todo momento al cuidador de la situación del paciente para lograr su monitorización y cuidado a distancia.

Además, gracias al servidor MQTT también se puede procesar y almacenar ordenada y correctamente toda la información trasmitida. Esto va a permitir al usuario de la aplicación, el cuidador, poder consultar la información en tiempo real o incluso la información de un momento pasado. Para ello se ha hecho uso de la base de datos *online* InfluxDB, la cual ha resultado sumamente útil para almacenar los datos permitiendo el acceso a los mismos en todo momento.

InfluxDB ha sido una base de datos óptima para este proyecto ya que dispone de guía en todo momento para poder disfrutar y explotar todas las posibilidades que ofrece, todo ello de forma gratuita lo cual facilita el desarrollo del proyecto.

Por otro lado, se considera un acierto la decisión de optar por el desarrollo de una aplicación híbrida frente a las demás opciones. Desarrollar una aplicación híbrida ha permitido un desarrollo más rápido, sencillo y eficaz. Las posibles desventajas que ofrecen este tipo de aplicaciones no han afectado al proyecto, pues se ha podido hacer uso de las funcionalidades del hardware del teléfono móvil gracias a las distintas librerías ya creadas para hacer uso de los recursos del teléfono deseados. Además, al ser una aplicación que muestra datos, no se trata de una aplicación con una gran gestión de recursos lo cual podría suponer un problema por la eficiencia de las aplicaciones nativas frente a las híbridas.

El *framework* Ionic dispone además de una comunidad muy grande lo que permite en cualquier momento solicitar y recibir ayuda en la red. Esto ratifica la decisión inicial de usar este *framework* para desarrollar una aplicación híbrida para poder monitorizar a personas con deterioro cognitivo leve en sus actividades en exteriores.

En la aplicación híbrida se ha desarrollado una pantalla de inicio de sesión bastante funcional y comercial. Se recogen todos los datos necesarios para que se pueda ofrecer una experiencia óptima al usuario y posteriormente se ha desarrollado una pantalla de acceso sencilla pero segura. Además, se ha dejado preparado un sistema para

recuperación de contraseñas mediante correo electrónico o SMS para implementar el servicio en un futuro si así se desea.

Se considera un gran acierto el uso del *framework* de Google Maps ya que este permite representar la posición del paciente en un entorno muy común, pues prácticamente todos los teléfonos móviles disponen de dicho *framework* lo cual asegura que la mayoría de los usuarios ya sabrán hacer uso de dicha funcionalidad. Además, Google Maps ofrece posibilidades muy interesantes como la visualización del entorno de una posición lo que permitirá al cuidador conocer en todo momento el área alrededor al punto donde se encuentra el paciente.

La pantalla que muestra el estado de la batería es una pantalla bastante sencilla, pero con gran poder gráfico para lograr una comprensión rápida y sencilla de la información que se trasmite.

Las pruebas realizadas con las funcionalidades añadidas de forma externa a la información recibida por el Dragino-LGT92 (agenda telefónica y registro de pautas médicas) indican que la aplicación desarrollada facilita el cuidado al encargado de la supervisión del paciente y le permite actuar rápidamente en caso de necesitar ponerse en contacto con el paciente o su médico por vía telefónica.

Finalmente, se valoran muy positivamente las notificaciones al cuidador acerca de posibles alertas como puede ser batería baja, el paciente saliendo de los límites o llegando a un punto libre, la hora de medicarse o incluso una situación de emergencia donde el paciente haya requerido ayuda pulsando el botón de emergencia.

De la misma manera, se considera de utilidad el sistema de ajustes de notificaciones que permite al cuidador escoger el nivel de alerta en cada una de ellas, pudiendo solicitar que sean notificaciones sonoras e intensas o simplemente una notificación sencilla en la barra superior del dispositivo móvil del cuidador. Se ofrece incluso la posibilidad de no notificar alguno de estos casos si en un momento determinado así lo desea el propio cuidador.

Por último, cabe destacar la satisfacción del colaborador externo al proyecto con la manejabilidad de la aplicación logrando así el objetivo de que esta sea lo más sencilla posible para el cuidador, de modo que pueda majear y entender todas y cada una de las funcionalidades de la aplicación desarrollada a lo largo de este proyecto.

5.2 Líneas futuras

Se presenta la primera versión de la aplicación que permitirá al cuidador conocer la situación del paciente que se encuentra bajo su cuidado en sus paseos al exterior. Se trata de una versión bastante completa como ya se ha ido explicando a lo largo de este documento acerca del proyecto que se ha llevado a cabo, sin embargo, existen posibles líneas futuras o mejoras que se proponen para implementar en futuras versiones.

En primer lugar, durante el registro se solicitan al usuario los códigos "Device EUI" y "AppKEY", ambos códigos son identificadores únicos y propios que posee cada

dispositivo Dragino. Estos códigos permiten asignar a cada usuario su dispositivo o dispositivos Dragino, de forma que cada usuario puede ver los datos de su paciente únicamente. De este modo, se podría incluso implementar la opción de registrar varios pacientes para un mismo cuidador.

Se ha dejado desarrollada la pantalla que permitirá recuperar la contraseña por email o SMS. Se debería implementar ambos servicios mediante un generador de claves que sustituya la clave olvidada por la generada por dicho servicio y que envíe la nueva clave bien por SMS o email según lo desee el usuario.

De la misma manera, se podría añadir la opción de que el usuario cambie su contraseña si así lo desea en un momento determinado.

Llegado el momento, sería muy interesante programar el Dragino-LGT92 para que cuente los pasos del paciente a lo largo del día aprovechando el acelerómetro que este tiene. Una vez programado esto y el envío de esta información, se podría crear un nuevo apartado en el menú que represente los pasos que da cada día el paciente. Se podría configurar los días que desea visualizar el paciente y comparar los datos mediante un diagrama de barras e incluso general alertas si la actividad física del paciente disminuye repentinamente.

Por su parte, en el Dragino LGT92 también permite ser programado para detectar caídas, de modo que en un futuro se podría crear un aviso cuando suceda la mencionada situación para que el cuidador pueda ser alertado inmediatamente y pueda actuar en consecuencia.

Llegado el momento, se puede incluir una opción que permita al usuario reordenar el listín de su agenda. Actualmente, los contactos añadidos se almacenan en orden cronológico según cuando se añaden, sin embargo, en futuras versiones podría ser interesante ofrecer al cuidador la posibilidad de reordenar la agenda a su gusto.

CAPÍTULO 6: PRESUPUESTO

En este apartado se recogen los costes que suponen llevar a cabo este proyecto:

• Costes en recursos de software

Concepto	Cantidad	Coste	Amortización	Uso	Subtotal
		unitario	(años)	(meses)	(€)
		(€)			
Windows 10	1	2.200	1	12	2.200
Keil µVision	1	3.300	1	4	1.100
The Things	1	0	-	-	-
Network					
Microsoft	1	90	1	6	45
Office					
Ionic	1	0	-	-	-
framework					
InfluxDB	1	0	-	-	-
IDLE	1	0	-	-	
Python					
TOTAL					3.345

Tabla 2. Recursos: Costes de software.

• Costes en recursos de *hardware*

Concepto	Cantidad	Coste unitario (€)	Amortización (años)	Uso (meses)	Subtotal (€)
PC 64 Bits,		549	6	12	91.5
8Gb de	1				
RAM, 256					
GB SSD					
DRAGINO					
LGT92					
LoRaWAN	1	43.5	8	12	5.44
GPS					
Tracker					
The Things					
Indoor	1	102.29	10	12	10.23
Gateway					
Smartphone	1	179.99	3	8	39.78
4Gb de					
RAM,					
Android 11					
TOTAL					146.95

Tabla 3. Recursos: Costes de hardware.

• Costes en mano de obra

Concepto	Cantidad (horas)	Coste Unitario (€)	Subtotal (€)
Documentación	50	15	750
Desarrollo de la aplicación	350	60	21.000
Desarrollo del sistema <i>hardware</i>	30	60	1.800
Coste total			23.550

Tabla 4. Recursos: Costes de mano de obra.

• Costes totales

Concepto	Subtotal	
	(€)	
Recursos software	3.345	
Recursos hardware	146,95	
Mano de obra	23.550	
TOTAL	27.041,95	

Tabla 5. Recursos: Costes totales

CAPÍTULO 7: PLIEGO DE CONDICIONES

Para poder disfrutar de las posibilidades que ofrece este proyecto se deben cumplir una serie de requisitos.

El paciente debe disponer del dispositivo Dragino LGT-92 y se debe encontrar dentro del alcance de la antena LoRaWAN para que esta pueda recoger la información y subirla a la red de internet, por tanto, la antena debe tener conexión a internet también.

Por su parte, el cuidador debe tener acceso a un teléfono móvil con sistema operativo Android o iOS para poder instalar la aplicación y ejecutar esta misma en su dispositivo, que desde donde recogerá toda la información que muestra la aplicación creada.

Para la recepción de la información, se necesita conexión a internet para conectarse con la base de datos de InfluxDB que es donde se almacena la información. Bien es cierto, que ciertas funcionalidades (agenda o listado de pautas médicas) no requieren conexión a internet.

Por otro lado, como ya se ha explicado, se pueden solicitar las posiciones del paciente en un tiempo determinado con independencia de si en dicho periodo se tenía o no acceso a internet. En este caso, solo es necesario tener acceso a internet durante la solicitud de la información.

CAPÍTULO 8: MANUAL DE USUARIO

En primer lugar, al arrancar la aplicación se solicita al usuario que escoja si desea registrase o acceder. En caso de no estar registrado, el usuario debe registrarse y posteriormente ya podrá acceder introduciendo las credenciales creadas durante el registro. (véanse las figuras 39 y 40).

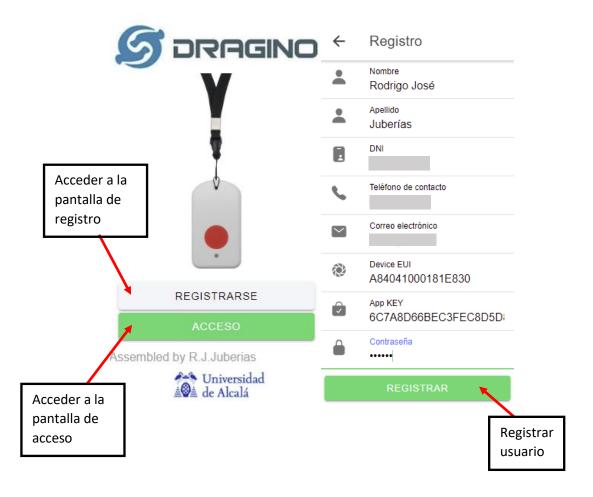


Figura 39. Manual de usuario: Primera pantalla (izq.) y pantalla de registro (dcha.).



Figura 40. Manual de usuario: Pantalla de acceso incompleto (izq.) y pantalla de acceso completo (dcha.).

Una vez llevado a cabo el acceso, se muestra la pantalla principal. La pantalla principal indica la última posición del paciente. Gracias al *framework* de Google Maps, si se arrastra el muñeco naranja y se suelta en un determinado punto del mapa se puede ver el entorno real de dicho punto. Abajo se tienen 3 botones: paciente (pantalla principal), información y ajustes. En la pantalla de información se indica cómo obtener ayuda, información acerca del Dragino-LGT92 y del proyecto desarrollado. Por su parte, en la pantalla de ajustes es donde el usuario podrá definir las distintas alertas y cómo se notificarán estas. Cada pantalla dispone de un botón (?) arriba a la derecha, desde el cual se abre un panel de ayuda respecto a la pantalla en la que se encuentra el usuario para facilitar su uso (véase la figura 41).

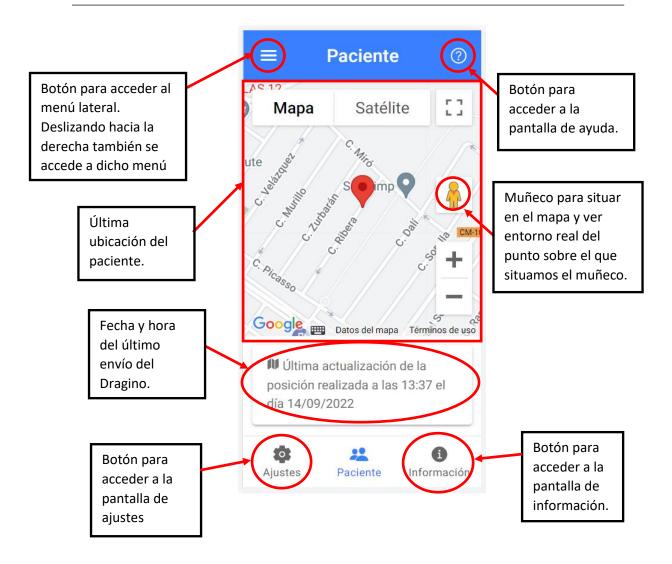


Figura 41. Manual de usuario: Pantalla principal.

Deslizando hacia la derecha o pulsando arriba a la izquierda sobre las 3 líneas horizontales, se abre el menú lateral. El menú lateral ofrece la opción de redirigir a las distintas pantallas de la aplicación.

Desde este menú se puede acceder a la información acerca de la batería, las posiciones donde ha estado el usuario, las pautas de medicación, crear o ver límites y puntos prohibidos, la agenda o incluso la pantalla de ayuda (véase la figura 42).



Figura 42. Manual de usuario: Menú lateral.

Existen pantallas con un botón con el símbolo de una casa arriba a la derecha, es un botón que devuelve al usuario a la pantalla inicial. Si en lugar de una casa hay una flecha hacia la izquierda, regresa a la pantalla anterior (véase la figura 43).



Figura 43. Manual de usuario: Pantalla batería.

Existen pantallas, como el caso de la agenda, medicación, lista de límites o de puntos prohibidos, en las que abajo a la derecha aparece un botón azul con el símbolo + en su interior. Al pulsar sobre dicho botón aparecen otros dos, uno verde para añadir medicamento, contactos, límites o puntos según en la pantalla que esté el usuario; y otro rojo para eliminarlos (ver figura 44). Si se pulsa en añadir se redirecciona al formulario para añadir lo que se desee. Si por el contrario se pulsa sobre eliminar, redirecciona a una pantalla donde escoger qué eliminar (ver figura 45).

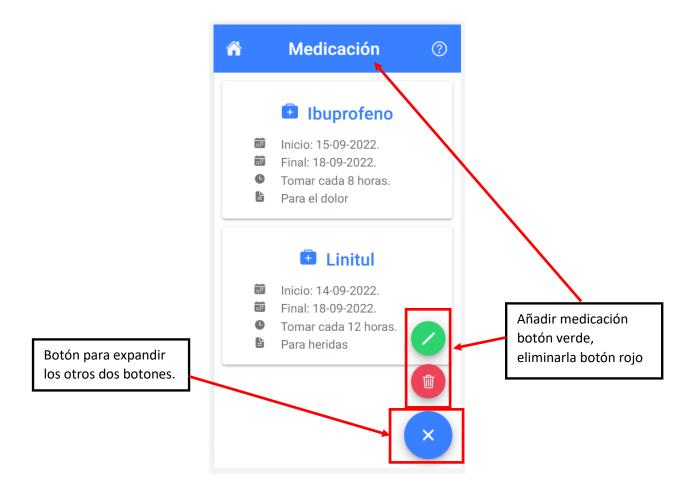


Figura 44. Manual de usuario: acceder a las pantallas para añadir/eliminar elementos.

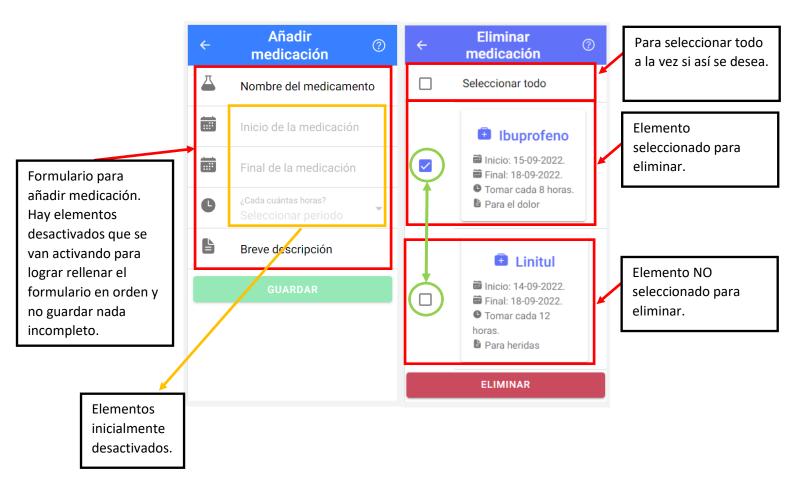


Figura 45. Manual de usuario: añadir (izq.) y eliminar (dcha.) elementos.

Por último, hay que destacar que se dispone de una agenda telefónica. Para acceder a ella, se debe pulsar el botón "Agenda" en el menú lateral. En dicha agenda se pueden eliminar o agregar contactos de la misma manera que se agregan o eliminan medicamentos. Además, esta agenda permite llamar de forma inmediata pulsando el botón rojo de llamada que ofrece cada contacto. Existe un buscador al inicio de la agenda para hallar el contacto rápidamente en caso de tener una agenda muy extensa (ver figura 46).

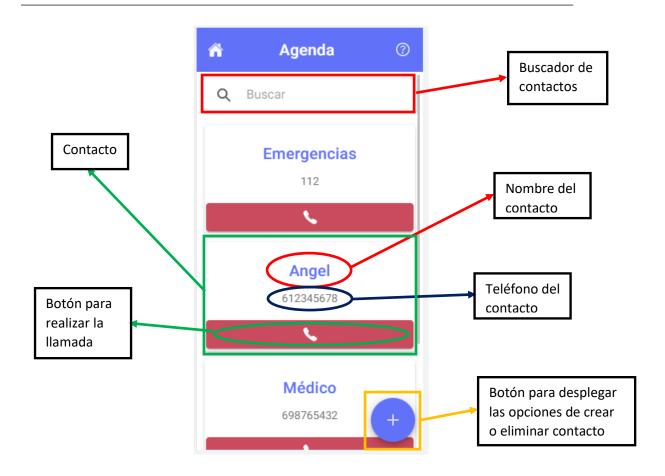


Figura 46. Manual de usuario: agenda.

CAPÍTULO 9: BIBLIOGRAFÍA

• [AIOTI, 2015] https://aioti.eu/wp-content/uploads/2017/03/AIOTIWG01Report2015-Applications.pdf si quieres meter algo de 5G, puedes usar esta otra: Microsoft Word - AIOTI IoT relation and impact on 5G 190308 R2 published.docx.

• [Antena TTN 2022]

https://www.thethingsnetwork.org/docs/gateways/thethingsindoor/ último acceso septiembre 2022.

- [Aplicaciones 2022] https://www.gsoft.es/articulos/que-necesito-web-apps-app-nativa-o-app-hibrida/#:~:text=Las%20Apps%20nativas%20son%20aquellas,programaci%C3%B3n%20espec%C3%ADfico%20de%20cada%20equipo, último acceso junio 2022.
- [Bertech, 2022] https://behrtech.com/blog/6-leading-types-of-iot-wireless-tech-and-their-best-use-cases/ ultimo acceso septiembre 2022.
- [Cote 2021] A. Cote, R. J. Phelps, N. S. Kabiri, J. S. Bhangu, K. Thomas, "Evaluation of wearable technology in dementia: a systematic review and meta-analysis", Front. Med. 7:501104. doi: 10.3389/fmed.2020.501104, 2021.
- [**Dragino 2021**] https://www.dragino.com/products/lora-lorawan-end-node/item/142-lgt-92.html, último acceso marzo 2022.
- [DeepData 2022] https://deepdata.es/red-lora/, último acceso julio 2022.

- **[EEPROM 2022]** https://es.wikipedia.org/wiki/EEPROM, último acceso junio 2022.
- [Elsevier 2019] T. Vega Alonso, M. Miralles Espí, J.M. Mangas Reina, D. Castrillejo Pérez, A.I. Rivas Pérez, M. Gil Costa, A. López Maside, E. Arrieta Antón, J.E. Lozano Alonso, M. Fragua Gil. https://www.elsevier.es/es-revistaneurologia-295-avance-resumen-prevalencia-deterioro-cognitivo-espanaestudio-S0213485316302171, último acceso junio 2022. Prevalencia del deterioro cognitivo en España. Estudio Gómez de Caso en redes centinelas sanitarias.
- [Esendex 2022] https://www.esendex.es/blog/post/espana-es-el-pais-europeo-con-mas-smartphones-por-habitante/#:~:text=Tanto%20es%20as%C3%AD%20que%20un,inteligente%20a%20d%C3%ADa%20de%20hoy, último acceso julio 2022.
- [GNSS 2022] https://gpstotal.org/es/gps/gnss, último acceso junio 2022.
- [GPS L76 2022]:

https://www.didacticaselectronicas.com/index.php/comunicaciones/gps/m%C3 %B3dulo-gnss-176-l-modulos-chip-de-con-gps-gnss-inal%C3%A1mbricos-de-comunicaci%C3%B3n-176-l-quectel-superficiales-smd-quectel-detail, último acceso junio 2022.

- **[GSM 2022]** https://www.universidadviu.com/es/actualidad/nuestros-expertos/que-es-gsm-y-como-funciona, último acceso junio 2022.
- [INE 2021] https://www.ine.es/expo_anuarios/1945-1975.html, último acceso febrero 2022.
- [InfluxDB 2022]: https://openwebinars.net/blog/que-es-influxdb-y-primeros-pasos/, último acceso junio 2022.
- [Ionic 2022]: https://ionicframework.com/, último acceso julio 2022.
- [Lauridsen 2016] M. Lauridsen, I. Z. Kovacs, P. Mogensen, M. Sorensen and S. Holst, "Coverage and Capacity Analysis of LTE-M and NB-IoT in a Rural Area," 2016 IEEE 84th Vehicular Technology Conference (VTC-Fall), 2016, pp. 1-5, doi: 10.1109/VTCFall.2016.7880946.

• [LoRa 2022]

https://www.didacticaselectronicas.com/index.php/comunicaciones/lora/modulo-lora-433mhz-modulos-tarjetas-de-comunicaci%C3%B3n-transmisores-lora-lora-de-largo-alcance-rf-inal%C3%A1mbricos-wireless-para-iot-sx1276-sx1278-de-433mhz-detail, último acceso junio 2022.

- [LoRaWAN,2022] https://lorawan.es/, último acceso marzo 2022.
- **[LORIX 2022]** https://iot.wifx.net/en/products/lorix-one/ último acceso septiembre 2022.
- [LPWAN 2022] https://gruposinelec.com/que-son-las-redes-lpwan/#:~:text=Las%20redes%20LPWAN%20(Low%20Power,de%20datos%20a%20larga%20distancia, último acceso junio 2022.
- [Marker Store 2022] https://maker-store.es/home-automation/industrial-automation/lora-iot-sensoren/4208/dragino-tracker-gps-lgt-92-version-pilas-2x-aa, último acceso septiembre 2022.
- [MPU 2022] Datasheet MPU9250, Mouser, https://www.mouser.es/ProductDetail/TDK-InvenSense/MPU-9250?qs=u4fy%2FsgLU9OhGjFkQSZssA%3D%3D
- [Neubauer 2018] Neubauer, N., et al. "Reflections of the use of locating technologies with persons with dementia: proceedings of a key stakeholder forum." Neurodegener Dis Manag 8(3): 195-205.
- [Oliveira 2017] R. Oliveira, L. Guardalben, and S. Sargento, "Long range communications in urban and rural environments", in 2017 IEEE Symposium on Computers and Communications (ISCC), Heraklion, Greece, 2017, pp. 810–817.
- [Oracle 2022] https://www.oracle.com/es/internet-of-things/what-is-iot/ último acceso julio 2022.

- [Polar2022]https://support.polar.com/es/what-is-gnss?product_id=99780&category=gps último acceso junio 2022.
- **[RAM 2022]** https://www.dell.com/support/kbdoc/es-es/000148441/what-is-memory-ram, último acceso junio 2022.
- [Responsive 2022]: https://www.cyberclick.es/que-es/diseno-web-responsive último acceso julio 2022.
- [Sanitas 2017] https://sanitasdatasalud.es/mas-de-12-millones-de-personas-enespana-sufrenalzheimer/#:~:text=En%20el%20caso%20de%20Espa%C3%B1a,de%20enfermedades%20como%20el%20alzh%C3%A9imer, último acceso enero 2022.
- [Semtech 2021] https://www.catsensors.com/es/lorawan/tecnologia-lora-y-lorawan, último acceso junio 2022.
- **[Stavropoulos 2020]** T. Stavropoulos et al. "IoT Wearable sensors and devices in elderly care: a literature review", Sensors, Vol. 20, 2826, pp. 1-22, 2020.
- **[STM 2022]** Datasheet STM32L072CZT6, STMicroelectronics, https://octopart.com/stm32l072czt6-stmicroelectronics-74568732?gclid=EAIaIQobChMIuJDDkOW--AIVx5TVCh3_VQFdEAAYASAAEgKt2PD_BwE.
- [Tayeh 2020]: https://hal.archives-ouvertes.fr/hal-02993852/document Gaby Bou Tayeh, Joseph Azar, Abdallah Makhoul, Christophe Guyeux, Jacques Demerjian. A Wearable LoRa-Based Emergency System for Remote Safety Monitoring. International Wireless Commu[1]nications and Mobile Computing Conference, Jun 2020, Limassol, Cyprus. ffhal-02993852f.

• [TTN 2022]:

https://www.thethingsnetwork.org/community/barranquilla/post/the-thingsnetwork, último acceso junio 2022.

- [UG65 2022]: https://aithings.store/lorawan-gateways/milesight-ug65-868m-ealorawan-gateway?gclid=CjwKCAjw1ICZBhAzEiwAFfvFhP1R7cYVK0JCFVmBQVeM KpImXvgXqq5th0UEWvYIKLfIL5R2_2JYbhoC9pcQAvD_BwE, último acceso septiembre 2022.
- [Wearables 2021] https://www.dispositivoswearables.net/, último acceso junio 2022

ANEXO: CÓDIGOS

Se muestran aquí las partes de código más relevantes del *script* programado en Python para lanzar el servidor MQTT. En la figura 47 se muestra cómo se recogen los datos desde TTN, a continuación, en la figura 48 se indica cómo se procesan los datos y finalmente en la figura 49 cómo los datos ya procesados son almacenados en la base de datos de InfluxDB.

```
import paho.mqtt.client as mqttClient
import time
from datetime import datetime
from influxdb client import InfluxDBClient, Point, WritePrecision
from influxdb client.client.write api import SYNCHRONOUS
# You can generate an API token from the "API Tokens Tab" in the UI
token =
org =
bucket = "DraginoLGT92"
Longitud prev=0.0
Latitud prev=0.0
def on connect(client, userdata, flags, rc):
    if rc == 0:
        print ("Connected to broker")
                                        #Use global variable
        global Connected
        Connected = True
                                        #Signal connection
    else:
       print("Connection failed")
def on message(client, userdata, message):
    info sin procesar=str(message.payload)
```

Figura 47. Recogida de datos.

```
#-----#
Str Alarm PosIni=int(info sin procesar.find("ALARM")+14)
Str Alarm PosFin=Str Alarm PosIni+1
Str Alarm=info sin procesar[Str Alarm PosIni:Str Alarm PosFin]
print(Str Alarm)
if (Str Alarm == "f"):
   Alarm= False
else:
   Alarm=True
print("Alarm: "+str(Alarm))
#-----#
Str BatV PosIni=int(info sin procesar.find("BatV")+6)
Str BatV Max=info sin procesar[Str BatV PosIni:Str BatV PosIni+15]
long BatV=int(Str BatV Max.find(","))
Str BatV PosFin=Str BatV PosIni+long BatV
Str_BatV=info_sin_procesar[Str_BatV_PosIni:Str_BatV_PosFin]
BatV=float(Str BatV)
if BatV>=4.0:
   Bat_pct=100
if BatV>=3.85 and BatV<4.0:
   Bat pct=70
if BatV>=3.70 and BatV<3.85:
   Bat pct=50
if BatV>=3.40 and BatV<3.70:
   Bat_pct=30
if BatV<3.40:
   Bat_pct=10
print("BatV: "+str(BatV))
print("%Bat: "+str(Bat pct))
```

Figura 48. Procesado de datos.

Figura 49. Almacenamiento de datos.

En los archivos .html se programa el aspecto de cada pantalla y en los archivos .ts el funcionamiento de estas.

En primer lugar, se muestran las partes de código más importantes de los archivos .html.

En la figura 50 aparece el código usado para programar todas las *header* o barras superiores de cada pantalla.

Figura 50. Código para dibujar mapas.

Al deslizar hacia abajo, se actualizan las pantallas. Para ello se añade al código de cada pantalla un fragmento que muestra que deslizando para abajo se actualiza y muestra una pantalla informando de que se está cargando hasta finalizar la actualización de la pantalla (ver figura 51).

Figura 51. Código actualizar pantalla.

Para finalizar con la parte del código .html, en la figura 52 se muestra un fragmento de código donde se incluyen botones que según la opción escogida redirige a una página u otra.

Figura 52. Botones de redirección a la siguiente página.

En la figura 53 se muestra el código empleado a la hora de dibujar los mapas con la API de Google Maps. Se incluye la función mostrarMapa() que dibuja el mapa y la función addMarker(marker:Marker) que es la que dibuja el marcador sobre el mapa.

```
mostrarMapa(){
    // create a new map by passing HTMLElement
   const mapEle: HTMLElement = document.getElementById('map');
   // create LatLng object
   var myLatLng = {lat: this.latitude, lng: this.longitude};
   // console.log(myLatLng)
   // create map
   this.map = new google.maps.Map(mapEle, {
      center: myLatLng,
     zoom: 16
   });
   // this.directionsDisplay.setMap(this.map);
    google.maps.event.addListenerOnce(this.map, 'idle', () => {
      mapEle.classList.add('show-map');
      const marker={
      position: {
        lat:this.latitude,
       lng: this.longitude
      title: 'Paciente'
      this.addMarker(marker);
    });
  addMarker(marker: Marker) {
   return new google.maps.Marker({
      position: marker.position,
     title: marker.title,
      map: this.map
```

Figura 53. Código para dibujar mapas.

A continuación, se muestra en la figura 54 cómo se actualiza cada pantalla. Al deslizar hacia abajo se ejecuta la función doRefresh(event) que llama a la función necesaria para actualizar y espera hasta el final de la ejecución de dicha función.

```
doRefresh(event) {
   //console.log('Actualizando...');
   this.loadMap();
   event.target.complete();
}
```

Figura 54. Actualización de pantallas.

En la figura 55 se muestra el código empleado para presentar alertas. Las alertas son las pantallas de ayuda que se solicitan pulsando el botón (?) que aparece arriba a la derecha en cada pantalla. Aportan información al usuario.

```
async presentAlert() {
  const alert = await this.alertCtrl.create({
    backdropDismiss:false,
    header: 'Información',
    message: '• Deslizar para abajo para actualizar. <br>
    buttons: [{
        text: 'Entendido',
        }
    ]
  });
  await alert.present();
}
```

Figura 55. Código para las alertas.

También se presenta en la figura 56 la estructura del código empleado para realizar las peticiones a la base de datos y recoger los datos deseados:

```
bateria(){
const token =
const org=
 // const bucket = 'DraginoLGT92'
 const {InfluxDB} = require('@influxdata/influxdb-client')
 const url='
 const client = new InfluxDB({url, token})
 let queryClient = client.getQueryApi(org)
 let fluxQuery = 'from(bucket: "DraginoLGT92")\
 |> range(start: -30d)\
 |> filter(fn:(r) => r._measurement == "datos")\
 |> filter(fn:(r) => r._field == "nivel_bateria%")\
 queryClient.queryRows(fluxQuery, {
 next: (row, tableMeta) => {
  const o = tableMeta.toObject(row)
   this.bat = o._value
  this.time=o._time
 complete: () => {
 console.log('Finished SUCCESS')
})
```

Figura 56. Solicitud a la base de datos.

De la misma manera, se adjunta en la figura 57 el código para almacenar datos desde el teléfono móvil.

```
saveData(){
 const token =
 // const bucket = 'DraginoLGT92'
 const {InfluxDB} = require('@influxdata/influxdb-client')
 const url='
 const client = new InfluxDB({url, token})
 let org =
 let bucket = `DraginoLGT92`
 let writeClient = client.getWriteApi(org, bucket, 'ns')
 let point = new Point('telefonos')
 .tag('usuario', 'host1' )
 .stringField('value', this.nombre+"**"+this.telefono)
 void setTimeout(() => {
  writeClient.writePoint(point)
 }, 0 * 1000) // separate points by 1 second
 void setTimeout(() => {
  writeClient.flush()
 }, 1000)
```

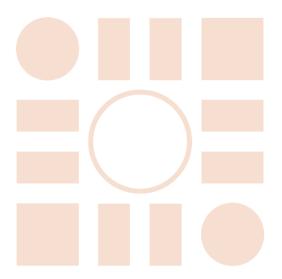
Figura 57. Escritura en la base de datos.

Por último, en la figura 58 se demuestra el código empleado para eliminar datos de la base de datos.

```
delete(){
  const url='
 const token =
 const org=
 const bucket = 'DraginoLGT92'
 const {InfluxDB} = require('@influxdata/influxdb-client')
 const influxDB = new InfluxDB({url,token})
 const deleteAPI = new DeleteAPI(influxDB);
 console.log("delete")
 console.log(this.checks.length)
  for(let i=0;i<this.checks.length;i++)</pre>
   if(this.checks[i]!=1){
     this.checks[i]=0;
   console.log(this.checks)
  for(let i=0;i<this.checks.length;i++){</pre>
   if(this.checks[i]==1){
     let segundo_start;
     segundo_start=("00")
     let segundo_stop;
      segundo\_stop = ("0" + (parseInt(this.times[i].slice(17,19)) + 1)).slice(-2)
     deleteAPI.postDelete({
       org,
       bucket,
         start:this.times[i].slice(0,17)+segundo_start+".000Z",
         stop:this.times[i].slice(0,17)+segundo_stop+".000Z",
         predicate: '_measurement=telefonos'
     })
```

Figura 58. Borrar elementos de la base de datos.

Universidad de Alcalá Escuela Politécnica Superior



ESCUELA POLITECNICA SUPERIOR

