



GRADO EN INGENIERÍA INFORMÁTICA

TRABAJO DE FINAL DE GRADO

Localización basada en Bluetooth

Autor:
Cindy Matilde AVILA ISLA

Supervisor:
Luby SHANDRA
Tutor académico:
Mercedes FERNÁNDEZ REDONDO

Fecha de lectura: 29 de abril de 2021
Curso académico 2020/2021

Resumen

Este documento recoge el Proyecto Final de Grado de la estudiante Cindy Avila Isla, realizado durante las 300 horas de estancia en prácticas en la empresa 720tec.

El proyecto realizado se basa en el desarrollo de una aplicación de configuración de balizas electrónicas, aunque en un principio el objetivo del proyecto fue la investigación de la tecnología y desarrollo de una plataforma de posicionamiento basada en la tecnología *Bluetooth* utilizando balizas y antenas.

A lo largo de este documento se verá el desarrollo de una aplicación móvil para el aprovisionamiento y mantenimiento de las balizas. Desde la metodología a utilizar para el desarrollo de este proyecto hasta la implementación del mismo.

Por otro lado, en plano secundario se abordará la investigación de sistemas de localización de interiores con dispositivos *Bluetooth* 5.1 y la búsqueda de proveedores de dicho sistema, quedó claro que los proveedores de este sistema no lo venden sin antes hacer una valoración del caso de uso real.

Palabras clave

Balizas, eddystone, Bluetooth, Android

Keywords

Beacon, eddystone, Bluetooth, Android

Agradecimientos

A mi madre por enseñarme a no darme por vencida y por ser mi gran apoyo durante todo el proceso de la quimioterapia, gracias por ser mi motor y motivo, sin ti este sueño de poder terminar el grado no hubiese sido posible.

A la profesora Asunción Castaño por interceder ante la empresa 720tec para poder realizar nuestras prácticas en tiempos de pandemia que no era nada fácil. Y de corazón agradecer a la empresa 720tec por darme esta grandiosa oportunidad, gracias por vuestra paciencia, por acogerme como una más de vuestro equipo. Gracias Jose, Luby, Borja y gracias a todo el equipo 720tec.

Gracias a mi tutora del proyecto Mercedes Fernández porque sin ella no hubiese sido posible presentar el TFG. Y también quiero agradecer a mis ex compañeras de la UJI y que en la actualidad son más que amigas, son mis hermanas de la vida que me incentivaron a continuar.

Y por último y no menos importante, agradecer a toda la familia Evangelio Pleno por acogerme como una más de la familia, fueron muy importantes a la hora de que se me diera la oportunidad de volver a la UJI a terminar el grado.

Quiero terminar este agradecimiento con un versículo que me acompañó en esta etapa de estudios en la UJI que coincidió con una de las etapas más duras de mi vida, es la siguiente:

«Aunque ande en valle de sombra de muerte, no temeré mal alguno, porque Tú estarás conmigo; tu vara y tu cayado me infundirán aliento.» Salmos 23:4

Índice general

1. Introducción	11
1.1. Contexto y motivación del proyecto	11
1.2. Objetivos del proyecto	12
2. Descripción del proyecto	13
2.1. Fases	13
2.1.1. Desarrollo de la App BeaconManager	13
2.1.2. Investigación de los sistemas de localización en interiores con dispositivo <i>Bluetooth</i>	14
2.1.3. Búsqueda de receptores con tecnología Bluetooth 5.1	14
2.2. Tecnologías utilizadas	15
2.2.1. Bluetooth	15
2.2.2. Baliza	18
2.3. Estructura de la memoria	19
3. Planificación del proyecto	21
3.1. Metodología	21
3.2. Planificación	24
3.3. Estimación de recursos y costes del proyecto	24
3.4. Seguimiento del proyecto	26

4. Análisis y diseño del sistema	29
4.1. Contexto	29
4.2. Estudio previo de la App del fabricante	31
4.3. Análisis del sistema	33
4.3.1. Requisitos del proyecto	33
4.4. Casos de uso	33
4.5. Análisis de las tecnologías	36
4.5.1. Baliza iBKS Plus	36
4.5.2. Protocolo Eddystone	37
4.6. Diseño de la arquitectura del sistema	40
5. Implementación y pruebas	41
5.1. Detalles de implementación	41
5.1.1. Aplicación BeaconManager	41
6. Conclusiones	47
A. Manual App BeaconManager v1.0	51
B. Interfaz final	69
B.1. Listar dispositivos BLE	69
B.2. Ver información detallada de la baliza	70
B.3. Configurar baliza	71
B.4. Modo conectable	72

Índice de figuras

3.1. Sprint	22
3.2. tareas	22
4.1. Portainer	30
4.2. Transtainer	30
4.3. Puerta por horas	30
4.4. Carretilla tipo CV	31
4.5. Carretilla tipo RS	31
4.6. Baliza iBKS-Plus	31
4.7. Listar dispositivos cercanos	32
4.8. Información detallada de una baliza	32
4.9. Configurar slots	32
4.10. Diagrama de casos de uso	34
4.11. Chip nRF51822	37
4.12. Pilas iBKS Plus	37
4.13. Acciones para listar dispositivos	38
4.14. Acciones para leer información de la baliza	39
4.15. Estructura global del sistema	40

5.1. Estructura del proyecto	42
5.2. Listar	43
5.3. Leer slots	43
5.4. Listar dispositivos ceranos	44
5.5. Información detallada de una baliza	44
5.6. Configurar slots	44
5.7. Modificar nombre de la baliza	45
5.8. Escribiendo en los slots correspondientes	45
5.9. Modificación efectuada	45
B.1. Listar dispositivos BLE v.2	69
B.2. Leer información de la baliza	70
B.3. Mostrar la información leída de la baliza	70
B.4. Modificar nombre de la baliza v.2	71
B.5. Escribiendo en los slots correspondientes v.2	71
B.6. Modificación efectuada v.2	71
B.7. Pulsar en conectable	72
B.8. Escribir las credenciales del administrador	72
B.9. Modo conectable activado	72

Índice de cuadros

3.1. Estimación de la planificación inicial.	25
3.2. Costes del proyecto	26
3.3. Estimación en horas de la planificación final del proyecto.	28

Capítulo 1

Introducción

Índice

1.1. Contexto y motivación del proyecto	11
1.2. Objetivos del proyecto	12

1.1. Contexto y motivación del proyecto

El proyecto de *Localización basada en Bluetooth* se va a desarrollar en la empresa 720tec S.L. Esta empresa ha sido fundada en el año 2014 y en sus seis años de actividad ya ha logrado un premio en la categoría Joven Empresa otorgada en 21^o edición de los Premios Empresa del Año otorgado por el periódico Mediterráneo. 720tec es una empresa tecnológica, respetada por la calidad de sus productos y servicios, actualmente se encuentra situada en el box B del edificio CEEI de Castellón de la Plana.

La misión de 720tec es articular un conjunto de productos y servicios que permiten a sus diferentes clientes el desarrollo de actividades innovadoras, que apunten a la creación y uso del conocimiento a través de la tecnología, que aumenten su capacidad de resolución de problemas y contribuyan a la generación de ventajas competitivas.

Uno de los grandes bloques de especialización de 720tec es el diseño, la provisión y el mantenimiento de infraestructuras de comunicación, computación y colaboraciones avanzadas, on promise y en la nube. El principal objetivo es acercar a la pyme infraestructuras y modelos de gestión reservados hasta ahora a grandes corporaciones.

El otro bloque y no menos importante es la acción empresarial, 720tec desarrolla proyectos especiales, en los que la integración de productos tecnológicos diferenciadores constituye la clave de su éxito.

La motivación para el desarrollo de este proyecto surge de la necesidad de complementar un proyecto que ha sido desarrollado por 720tec para un cliente que necesita evitar que los contenedores que se descargan de los barcos se pierdan por el puerto, es decir realizar un gestión eficiente de las

cargas. Por un lado 720tec empezó creando una aplicación que llevará cada camionero instalado en su móvil denominado *Transport*. Por otro lado, están las balizas que estarán instaladas en determinadas zonas y que emitirán una UID específica que indicará a la aplicación móvil por donde está pasando el camión. Según los datos que se van obteniendo, será el portainer o el trastainer quien elegirá el siguiente camión que descargará del barco que está ubicado cerca del portainer o trastainer.

Este proyecto necesita de una aplicación propia que permita al personal de mantenimiento/instalación del puerto poder configurar las balizas con slots UID y TLM indicadas para que la app *Transport* pueda realizar su función. Además se necesitaba investigar cual sería el receptor adecuado para montar un sistema de localización con dispositivos *Bluetooth 5.1*. Esta última parte se verá modificada por la falta de proveedores de receptores con tecnología *Bluetooth 5.1 AoA* que quieran venderlo para caso de estudio.

Esta última parte del proyecto se verá modificada a lo largo del desarrollo de este. A falta de proveedor vamos cambiar los roles, las balizas actuarán de receptores y la app *Transport* instalada en los móviles funcionarán como emisores. Se investigará la forma de añadir inteligencia a la app *Transport* para que elija el camión de forma más eficiente.

1.2. Objetivos del proyecto

El principal objetivo del proyecto es investigar sobre la localización de interiores con dispositivos *Bluetooth 5.1*. En un principio se espera poder trasladar los resultados de la investigación a un proyecto real. Para la realización del proyecto es necesario llevar a cabo una serie de objetivos, estos se logran siguiendo una serie de tareas que se mencionan a continuación:

- Investigación de los sistemas de localización en interiores utilizando *Bluetooth 5.1* .
- Familiarizarse con la tecnología a utilizar.
- Búsqueda y comparación de dispositivos candidatos.
- Puesta en marcha de balizas y antenas.
- Desarrollo de la aplicación de mantenimiento de balizas.
- Pruebas de localización basadas en la tecnología.
- Visualización de los datos.
- Escenario teórico de posicionamiento.
- Prueba en un escenario real.
- Conclusiones y cierres.

Desde el punto de vista estratégico, el objetivo del proyecto es trasladar este sistema de localización con dispositivos *Bluetooth* a otros clientes. Por otro lado, el objetivo operacional y táctico, es que el sistema facilite la posición de los contenedores descargados de los barcos.

Capítulo 2

Descripción del proyecto

Índice

2.1. Fases	13
2.1.1. Desarrollo de la App BeaconManager	13
2.1.2. Investigación de los sistemas de localización en interiores con dispositivo <i>Bluetooth</i>	14
2.1.3. Búsqueda de receptores con tecnología Bluetooth 5.1	14
2.2. Tecnologías utilizadas	15
2.2.1. Bluetooth	15
2.2.2. Baliza	18
2.3. Estructura de la memoria	19

El proyecto a desarrollar se va a dividir en cuatro fases. La primera es el desarrollo de la *App BeaconManager*, la segunda es la investigación de los sistemas de localización en interiores con dispositivos *Bluetooth*, la tercera es la búsqueda de proveedores de receptores con tecnología *Bluetooth 5.1* y la cuarta son pruebas en escenario teórico y real del sistema de localización en interiores.

Aquí podría decir que la ultima fase se modificó, al desarrollar una aplicación de configuración de balizas de un proyecto real, se ha decidido estudiar una aplicación desarrollada por 720tec denominada "Transporter", donde cada camionero la lleva instalada en su móvil. El puerto por donde los camioneros se desplazan tienen instaladas y configuradas las balizas en portainers, trastainers y carretillas.

2.1. Fases

2.1.1. Desarrollo de la App BeaconManager

Esta fase del proyecto se ha dividido en dos partes, en la primera parte es imprescindible investigar la documentación existente sobre balizas *iBKS Plus* y el protocolo *Eddystone* además de familiarizarse en la práctica con las balizas y con sus librerías correspondientes. En la segunda parte, se ha de

desarrollar una App para configurar las balizas de forma automática. La aplicación la denominamos *BeaconManager*. Lo que se desea con esta aplicación es configurar los slots de las balizas de forma más fácil que con la propia aplicación del fabricante.

En la Figura 4.7, Figura 4.8 y Figura 4.9 se puede observar el flujo de trabajo para realizar una modificación en cualquiera de los slots o la configuración global de la baliza. La empresa 720tec, tiene pensando comprar sobre 100 dispositivos *beacon* para un proyecto que tiene en una empresa de Bilbao, donde se instalarán y asignarán uno o varias balizas por maquinaria. Ellos necesitan que el proceso de configurar la baliza sea lo menos complicado posible, de hecho solo configurar dos slots edystone; UID y TLM, con una características que ya están establecidas en un archivo JSON.

2.1.2. Investigación de los sistemas de localización en interiores con dispositivo *Bluetooth*

Esta segunda fase del proyecto no tiene nada que ver con la primera, en esta vamos directo a investigar sistemas de localización en interiores con dispositivos *Bluetooth Low Energy (BLE)*. Para ello, se ha investigado las versiones que se están utilizando a lo largo de estos años y la versión emergente 5.1 que da mejor precisión que las anteriores y que puede competir con los sistemas de localización de interiores *UWB* que son más costosos.

Bluetooth 5.1 en sus nuevas funciones introduce la búsqueda de dirección, tiene la posibilidad de determinar la ubicación del dispositivo BLE con una precisión de hasta un centímetro. Esto se debe a la nueva capacidad de calcular la dirección de una señal Bluetooth mediante el ángulo de llegada (AoA), que se utilizará junto con la intensidad de la señal detectada (RSSI) para determinar y rastrear la ubicación de dispositivos y ofrecer una mayor precisión.

2.1.3. Búsqueda de receptores con tecnología Bluetooth 5.1

Después de la fase de investigación, se presentó un hizo una presentación de las versiones de tecnologías bluetooth, la empresa decidió que se buscara proveedores con tecnología *BLE 5.1*. Durante la investigación de posibles proveedores de receptores, hemos encontrado a la empresa Insoft, Quppa, Sewio, Localino, BluEpyc,etc. Entre ellos descartamos muchos porque su hardware solo está disponible para RTLS UWB, otros solo localizan con tecnología Bluetooth 4.0 midiendo la intensidad de la señal (RSSI). De todos ellos nos quedamos con Insoft y Quppa.

Al comunicarnos con Quppa, me encontré con una gran problema, ellos no quieren vender sus receptores sin saber toda la información del caso de uso real, y fue imposible lograr que nos vendan sus receptores. Por otro lado, de Insoft nos interesa el receptor "Locator Node AoA 1200", esta empresa alemana nos dijo que por el momento esos receptores solo están a la venta en Alemania, no pueden venderlo fuera de dicho país.

2.2. Tecnologías utilizadas

2.2.1. Bluetooth

La tecnología Bluetooth [1], opera en la banda de frecuencia sin licencia de 2,4GHz, admite múltiples opciones de radio que permiten a los desarrolladores crear productos que cumplan los requisitos de conectividad exclusivos de su mercado. Esta tecnología será la base del proyecto. En la primera fase del proyecto se establece una comunicación entre dispositivos móviles y balizas utilizando esta tecnología y en la segunda fase se investiga sistemas de localización en tiempo real (RTLS) usando la tecnología *Bluetooth* 5.1.

Los dispositivos *Bluetooth* se gestionan mediante una topología de radio frecuencia denominada "topología en estrella". Un grupo de dispositivos sincronizados de esta manera forman un *piconet*, este puede contener un maestro y hasta siete esclavos activos, con otros esclavos que no participan activamente en la red. En un *piconet*, el canal de radio físico es compartido por un grupo de dispositivos que están sincronizados con un reloj común y un patrón de salto de frecuencia, siendo el dispositivo maestro el que proporciona las referencias de sincronización.

El canal físico o enlace inalámbrico se subdivide en unidades de tiempo conocidas como ranuras. Los datos se transmiten entre dispositivos con *Bluetooth* en paquetes que se colocan en estas franjas. El salto de frecuencia tiene lugar entre la transmisión o la recepción de paquetes, por lo que los paquetes que componen una transmisión pueden enviarse por diferentes frecuencias dentro de la banda ISM.

La tecnología Bluetooth utiliza los principios de *inquiry* y *inquiry scan* de los dispositivos. Los dispositivos de escaneo escuchan en frecuencias conocidas los dispositivos que están preguntando activamente. Cuando se recibe una consulta, el dispositivo de exploración envía una respuesta con la información necesaria para que el dispositivo de consulta determine y muestre la naturaleza del dispositivo que ha reconocido su señal. Las versiones de *Bluetooth* [5] que podemos encontrar hasta la fecha son las siguientes:

- Bluetooth 1.0: Fue la primera versión lanzada en 1999, y precisamente por ser el primer paso para la tecnología, tuvo muchos problemas de conectividad y seguridad. Actualmente se encuentra en desuso.
- Bluetooth 1.1: Fue la primera actualización del Bluetooth 1.0, y llegó en 2002. Es ya una versión madura y comercial con una tasa de transmisión de alrededor de 721 kbps.
- Bluetooth 1.2: La segunda actualización del Bluetooth 1.0, que llegó en 2003 y redujo las interferencias que existían todavía en la versión 1.1 para terminar de afinar la tecnología.
- Bluetooth 2.0: La segunda gran versión de Bluetooth llegó en 2004, aunque compatible con la 1.2. Su principal característica fue la introducción de la Enhanced Data Rate, (EDR o tasa mejorada de datos), que le permitía ofrecer una tasa de transmisión de algo más de 2 Mb/s.
- Bluetooth 2.1: La siguiente versión lanzada en 2007, que mantuvo la misma tasa de transmisión que su antecesora. Sin embargo, introdujo una característica que cambió el Bluetooth para siempre, la posibilidad de que un terminal pudiera agregar a otro, y conectarse automáticamente sin necesidad de un PIN ni nada por el estilo.

- Bluetooth 3.0. Esta versión llegó en 2009, y trajo consigo el término o apellido HS de High Speed (gran velocidad). Supuso una gran mejora en la tasa de transmisión, que podía llegar a alcanzar los 24 Mbps.
- Bluetooth 4.0: Esta es posiblemente una de las versiones más importantes del estándar, ya que con ella se empezó a combatir el consumo excesivo de batería de la tecnología mediante Bluetooth Low Energy. La versión llegó en 2010, mantuvo la tasa de transferencia de 24 Mb/s y gracias a su bajo consumo se empezó a utilizar en dispositivos más pequeños y menos potentes.
- Bluetooth 4.1: Llegó en 2013, y se caracterizó por dirigirse sobre todo al mundo del IoT (Internet de las cosas) al permitir la conexión entre dispositivos pequeños sin intermediarios.
- Bluetooth 4.2: Esta siguiente actualización llegó en 2014, e implementó el protocolo IPv6 para permitir la conexión directa a través de Internet.
- Bluetooth 5.0: A mediados de 2016 llegó la gran revolución, un nuevo paso adelante que doblaba la tasa de transferencia, cuatruplicaba el alcance, seguía con un bajo consumo pensado en el IoT y permitía una mayor cantidad de datos en cada mensaje transferido. En cuanto a números, la tasa de transferencia era de hasta hasta 50 Mb/s, y el alcance hasta 240 metros.
- Bluetooth 5.1: La nueva versión llegada en 2019 giraba en torno a la localización, permitiendo que los dispositivos puedan saber la ubicación de otros dispositivos a los que estén conectados con un margen de centímetros. También podrá identificar la dirección de donde proviene una señal que está buscando.
- Bluetooth LE Audio (5.2): Es un nuevo estándar publicado principios del 2020, y que mejora la calidad de sonido y la eficiencia energética en todos los dispositivos que se conectan para funciones de audio. Utiliza un nuevo códec, el Low Complexity Communication Codec (LC3), que puede comprimir y descomprimir de forma más eficiente los datos que se transmiten y permite la transferencia de audio a varios dispositivos a la vez.

Bluetooth 5.1

Bluetooth 5.1 es uno de los últimos estándares de la tecnología *Bluetooth* que aporta nuevas mejoras. Construida sobre el actual estándar *Bluetooth 5.0*, la nueva tecnología Bluetooth 5.1 no solo mejorará la estabilidad de la conexión, sino que también proporcionará datos en tiempo real, incluyendo la localización y la detección de proximidad. La tecnología *Bluetooth 5.1* permitirá encontrar los dispositivos *Bluetooth 5.1* y localizarlos a unos centímetros de su ubicación exacta, entre sus mejoras se podrá encontrar:

- Búsqueda de dirección: Mejora la localización del dispositivo, permite a los dispositivos *Bluetooth* determinar la dirección de la transmisión de la señal *Bluetooth*. Los sistemas de localización con dispositivos *Bluetooth* antes de la llegada del nuevo estándar 5.1, se basaban en estimar la distancia a partir del indicador de fuerza de la señal recibida (RSSI). Este nuevo estándar 5.1, utilizará una combinación de de dos métodos diferentes para determinar el ángulo desde el que se transmite una señal *Bluetooth*, esto dará una posición con un alto grado de precisión. El cálculo de la posición basado en método del ángulo de llegada (AoA) también se denomina triangulación. A diferencia de la lateralidad, la triangulación utiliza no sólo la distancia al generador de señales, sino también los ángulos para determinar la posición de un

objeto. Para determinar la posición en dos dimensiones se necesitan dos ángulos y una longitud. En el método del ángulo de llegada (AoA), el dispositivo hacia el que se determina la dirección, como una etiqueta en una solución RTLS (Sistema de localización en tiempo real), transmite una señal especial de radiogoniometría utilizando una sola antena. El dispositivo receptor, como un localizador en esa misma solución RTLS, tiene múltiples antenas dispuestas en un conjunto. Cuando la señal transmitida atraviesa el conjunto, el dispositivo receptor ve una diferencia de fase de la señal debido a la diferencia de distancia entre cada una de las antenas de su conjunto y la antena transmisora. El dispositivo receptor toma muestras IQ de la señal mientras cambia entre la antena activa del conjunto. Basándose en los datos de las muestras IQ, el dispositivo receptor puede calcular la dirección relativa de la señal. El método AoA de radiogoniometría está pensado para su uso con RTLS y con soluciones de proximidad, como los servicios de información de puntos de interés (PoI).

Por otro lado, el método del ángulo de salida (AoD), el dispositivo hacia el que se está determinando la dirección, como una baliza de localización en una solución IPS, transmite una señal especial utilizando múltiples antenas dispuestas en un conjunto. El dispositivo receptor, como un teléfono móvil en esa misma solución IPS (sistema de posicionamiento en interiores), tiene una sola antena. Cuando las múltiples señales del dispositivo transmisor atraviesan la antena del dispositivo receptor, éste toma muestras IQ. Basándose en los datos de las muestras IQ, el dispositivo receptor puede calcular la dirección relativa de la señal. El método AoD de radiogoniometría está pensado para su uso en soluciones IPS, como las utilizadas para la búsqueda de caminos.

Bluetooth low energy (BLE)

Bluetooth low energy es la versión inteligente y de bajo consumo de la tecnología inalámbrica Bluetooth. Bluetooth Low Energy, también comercializado como Bluetooth Smart, comenzó como parte de la especificación básica de Bluetooth 4.0. Diseñado inicialmente por Nokia como Wibree antes de ser adoptado por el Grupo de Interés Especial (SIG) de Bluetooth, su objetivo inicial era proporcionar un estándar de radio con el menor consumo de energía posible, optimizado específicamente para un bajo coste, un bajo ancho de banda, un bajo consumo y una baja complejidad. BLE es una tecnología que pretende funcionar durante un largo periodo de tiempo con una o un par de simples pilas, como es el caso de las balizas que pueden estar en funcionamiento durante más de un año sin cambiar la pila según el tipo de configuración. Sus características son las siguientes:

- Menor consumo de energía: El consumo de energía máxima es inferior a 15mA y el consumo medio es aproximadamente 1 A. El consumo de energía activa se reduce a una décima parte del consumo de energía del *Bluetooth* clásico. En aplicaciones de bajo ciclo de trabajo, una pila de botón podría proporcionar entre 5 y 10 años de funcionamiento fiable.
- Rentabilidad y compatibilidad: Para conseguir compatibilidad con la tecnología *Bluetooth* clásica y eficiente en costes hay dos tipos de chipset, tecnología de modo dual con funcionalidad BLE y Bluetooth clásica y la tecnología BLE autónoma optimizada para dispositivos pequeños que funcionan con baterías y cuyo objetivo es el bajo coste y el bajo consumo de energía
- Robustez, seguridad y fiabilidad: La tecnología BLE utiliza la misma tecnología de salto de frecuencia adaptable (AFH) que la tecnología Bluetooth clásica. Esto permite a BLE lograr

una transmisión robusta en los entornos de radiofrecuencia ruidosos que se encuentran en las aplicaciones domésticas, industriales y médicas. Para minimizar el coste y el consumo de energía del uso de AFH, la tecnología BLE ha reducido el número de canales a 40 canales de 2 MHz de ancho en lugar de los 79 canales de 1 MHz de ancho utilizados con la tecnología Bluetooth clásica.

- **Facilidad de uso e integración:** Una piconet BLE se basa normalmente en un maestro conectado a varios esclavos. Un dispositivo puede ser maestro o esclavo, pero nunca ambos. El maestro controla la frecuencia con la que los esclavos pueden comunicarse y el esclavo sólo se comunica a petición del maestro. Una nueva característica que añade BLE con respecto a la tecnología Bluetooth clásica es la función de publicidad. Con esta función, un dispositivo que actúa como esclavo puede anunciar que tiene algo que transmitir al maestro. Un mensaje publicitario también puede incluir un evento o un valor de medición.
- **Alcance de la conexión:** La tecnología BLE tiene una modulación ligeramente diferente a la de la tecnología *Bluetooth* clásica. Esta diferenciación de modulación ofrece un alcance de hasta 300 metros con un chipset de radio de 10 dBm (máximo de BLE).

2.2.2. Baliza

Un beacon es una baliza electrónica, este es un pequeño transmisor inalámbrico que utiliza tecnología Bluetooth de bajo consumo para enviar señales a otros dispositivos inteligentes cercanos. Son uno de los últimos avances en tecnología de localización y marketing de proximidad. En pocas palabras, conectan y transmiten información a los dispositivos inteligentes haciendo que la búsqueda e interacción basadas en la localización sean más fáciles y precisas.

El hardware consiste en un microcontrolador con un chip de radio bluetooth LE y una batería, generalmente de botón. Los nuevos chips están optimizados para trabajar con BLE, versiones anteriores fueron diseñados con bluetooth clásico, lo que requería un mayor consumo de energía. Los beacons transmiten una señal con una potencia fija, conocida como Tx Power. A medida que la señal viaja en el aire la intensidad de la señal va disminuyendo con la distancia. Con un Tx Power superior la señal puede viajar distancias más largas, lo que significa mayor consumo y un Tx Power menor se traduce a menor rango de alcance pero menos consumo de batería.

La periodicidad con la que un beacon emite una señal se conoce como *advertising interval*. Un intervalo de 100 ms significa que la señal se emite cada 100 milisegundos, es decir, 10 veces en un solo segundo. Un intervalo de 500 ms significa que la señal se emite solo dos veces por segundo, lo que se traduce en menos consumo de energía. Cuando *advertising interval* aumenta la duración de la batería aumenta pero la capacidad de respuesta del dispositivo receptor disminuye. No hay una elección óptima del *advertising interval*. Las aplicaciones que necesiten baja latencia deben elegir intervalos más bajos. En cambio, los que necesiten mayor duración de la batería necesitan un intervalo mayor

2.3. Estructura de la memoria

La estructura de este documento es de 6 capítulos más el anexo. Cada capítulo está dividido en secciones exceptuando el último capítulo dedicado a la conclusión, la estructura es la siguiente:

- Introducción, donde se detalla el contexto y la motivación por la cual la empresa quiere desarrollar este proyecto, además de definir los objetivos del mismo.
- Descripción del proyecto, se detalla las fases por las que pasa este proyecto, además de la tecnología a usar.
- Planificación del proyecto, se indica y se describe la metodología a utilizar en el desarrollo de este proyecto, además de realizar una planificación y estimación de los recursos que se van a utilizar para su desarrollo.
- Análisis y diseño del sistema, se indica el contexto en el que se va usar la aplicación a desarrollar, se define los requisitos del sistema, se hace una análisis de la tecnología que se usará en esta aplicación y por último se indica el diseño de la arquitectura del sistema.
- Implementación y pruebas, se detalla como ha sido el proceso de desarrollo del sistema y se muestra la interfaz final de cada uno de los casos de uso.

Aunque no lo he puesto como item en la lista anterior, en el anexo se detalla la investigación sobre sistemas de localización en interiores con dispositivos *Bluetooth*, los proveedores encontrados, el manual que se desarrollo para la primera versión de la aplicación *Beacon Manager*.

Capítulo 3

Planificación del proyecto

Índice

3.1. Metodología	21
3.2. Planificación	24
3.3. Estimación de recursos y costes del proyecto	24
3.4. Seguimiento del proyecto	26

En este capítulo se describe la metodología que se ha utilizado para el desarrollo del proyecto, así como la planificación inicial y las modificaciones que se han realizado a lo largo del desarrollo del mismo, además de los recursos que se han necesitado para llevarlo a cabo.

3.1. Metodología

En este apartado se define la metodología a utilizar para el desarrollo del proyecto y las herramientas que se utilizan a lo largo de la misma.

La empresa 720Tec utiliza en sus equipos de desarrollo el software *Microsoft Teams*¹ como herramienta para la gestión del desarrollo de sus proyectos. Esta herramienta sigue una metodología ágil *Scrum Manager*², las bases de esta metodología radican en su transparencia a la hora de poner en común ideas, en lo adaptable que es a los cambios y en su capacidad de llevar un registro del trabajo que muestra si el equipo en sí está funcionando.

Esta metodología nos permite dividir el proyecto en una serie de *sprints* y cada *sprint* en una serie de tareas. El *sprint* es el centro de esta metodología ágil, es como un contenedor de tareas del proceso. Todo lo que ocurre en una iteración para entrega de valor está dentro de un *sprint*. En este proyecto la duración máxima de un sprint es de una semana, aunque en algún momento por causas de fuerza mayor se ha añadido una semana más.

¹*Microsoft Teams*: <https://docs.microsoft.com/es-es/microsoftteams/teams-overview>

²*Scrum Manager*: <https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html>

Las reuniones o *daily meeting* son un punto clave de esta metodología, sobre todo la primera, la que se denomina *sprint planning*, es donde se indican los diferentes objetivos a los que tiene que llegar el proyecto y las tareas que se van a abordar en cada *sprint*. Es importante indicar que estas reuniones nos permiten poner en práctica el trabajo en equipo, los *sprints* y las tareas que contendrán estos mismos es el resultado de intercambiar ideas, dar cada miembro del equipo su punto de vista, ser críticos constructivos con cada uno de los objetivos que se van consiguiendo y en el caso de no llegar a terminar las tareas de un *sprint*, ver entre todos el motivo de no sacar adelante la tarea e indicar alternativas para llegar al objetivo.

A continuación se muestra la Figura 3.1, que ilustra uno de los *sprints* del proyecto con sus respectivas tareas. Se puede observar que el nombre del *sprint* se denomina Cambios en la App. Los items de la lista de control son todas las tareas que se tienen que desarrollar en ese *sprint*, esa lista es lo que se denomina el *sprint backlog*

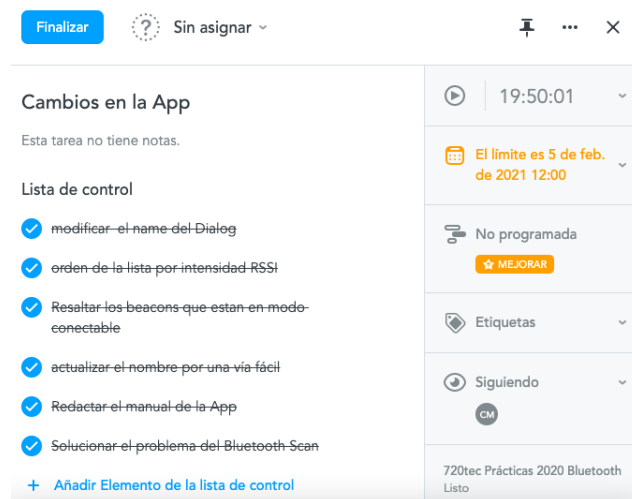


Figura 3.1: Sprint

En la Figura 3.2, muestra la forma en que el equipo lleva un control de todos los sprints, mostrando las diferentes etapas por las que pasa. En el software *Microsoft Teams*, en equipo, sección tareas se puede encontrar esta herramienta. El *sprint* puede pasar por las etapas: abiertas, en progreso, revisión y listo.

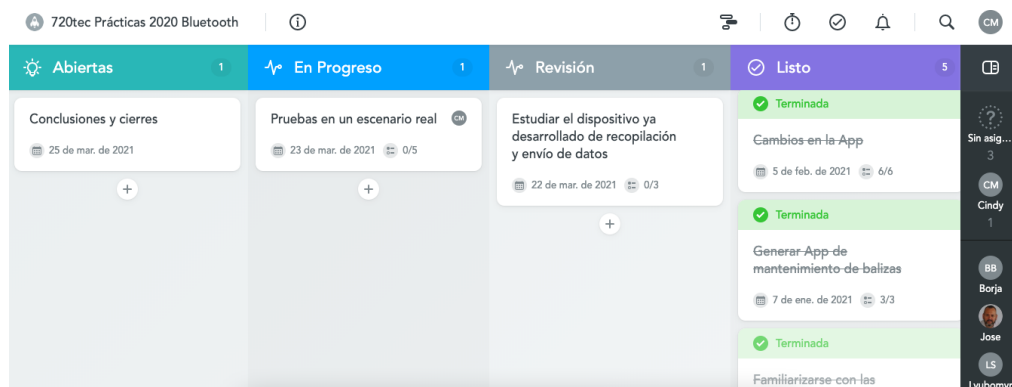


Figura 3.2: tareas

En la primera reunión o *sprint planning*, definimos una serie de sprints con sus respectivas tareas a abordar para llegar al objetivo que en un principio es desarrollar un sistema de localización en interiores con dispositivos *Bluetooth*. A continuación, se lista los *sprints* con sus respectivas tareas:

- Investigación sobre sistemas de localización en interiores utilizando Bluetooth.
 - Tecnologías que utilizan .
 - *Smoothing* localización.
 - Algoritmos de localización.
 - Protocolos de comunicación.
- Familiarizarse con las tecnologías.
 - Familiarizarse con el uso de balizas.
- Generar aplicación de mantenimiento de balizas.
 - Exploración SDK.
 - Desarrollo de la aplicación.
 - Testeo.
- Escenario teórico de posicionamiento.
 - Investigar o seleccionar los algoritmos de localización.
 - Investigar o seleccionar los dispositivos.
 - Montar escenario de prueba.
 - Testear soluciones desarrolladas.
 - Conclusiones.
- Prueba en un escenario real.
 - Revisión del dispositivo localizador.
 - Revisión del algoritmo de localización.
 - Revisión de las balizas.
 - Implementación de las mejoras encontradas.
 - Análisis de los resultados.
- Conclusiones y cierres.

La ventaja de esta metodología es que es fácil de aprender, permite agilizar los procesos, en algún punto puede parecer como ventaja o desventaja la posibilidad de realizar el camino más corto para conseguir el objetivo de un sprint.

3.2. Planificación

En este apartado se detallará tanto la planificación inicial, en el Cuadro 3.1 se describe la planificación inicial del proyecto, el número de horas que se va a distribuir en el tiempo. En el apartado anterior se indicó las tareas que se llevarían a cabo en cada *sprint*. En la siguiente tabla se muestra el número de horas por *sprint*, de tal forma que al final se hayan cumplido las trescientas horas de prácticas en la empresa, sin contar las horas no presenciales en el desarrollo de los documentos quincenales y la redacción del TFG.

Toda esta planificación se va a dividir en principio en cuatro fases:

- Desarrollo de la aplicación de mantenimiento de balizas.
- Investigación de los sistemas de localización en interiores con dispositivos *Bluetooth*.
- Búsqueda de receptores con tecnología *Bluetooth 5.1*.
- Implementación del sistema de localización en interiores con dispositivos *Bluetooth* tanto para un caso teórico y real.

En este apartado es necesario indicar que cuando se hizo la planificación inicial de este proyecto no se tenía pensado un caso de uso real en que se podría poner en marcha este sistema, en lo que si había un caso de uso real era en la aplicación de la configuración de balizas. Y cuando se habla de un sistema de localización en interiores con dispositivos *Bluetooth*, se tenía pensado desde un principio buscar un proveedor de sistemas de localización en interiores en tiempo real con dispositivos *Bluetooth* que nos proporcionará los receptores, emisores y el SDK, en ningún momento se planteo desarrollar desde cero.

3.3. Estimación de recursos y costes del proyecto

En esta sección se presenta la estimación del coste que supone la realización de este proyecto. Por una parte se contemplan los recursos humanos (programador/investigador en las 300 horas de trabajo) y por otro lado los recursos materiales y de servicios contratados.

Según el convenio del TIC 2021, en las tablas salariales³ en el Área 3 Grupo E Nivel 1 (programador junior) recibe un sueldo de 1058,42 €, a esto la empresa tiene que añadir un 31,05% en concepto de contingencias comunes, lo que supone añadir 328,64 € a los 1058,42 €, quedando un total de 1387,06 €/mes, unos 8,67 €/hora.

El software utilizado en el proyecto es gratuito. En un principio si se hubiera encontrado proveedor de RTLS con dispositivos Bluetooth que quieran vendernos sus productos a un caso teórico, se hubiese añadido a los costes entre 2000 € - 3000 €. Aunque como dijimos anteriormente este apartado no se llevó a cabo y no se realizó la compra.

³Nómina Convenio TIC: <https://bankintercomite.es/pag/bk/general/nomina/tic/index.php?conc=tablas>

Planificación inicial		
Sprint		Horas
Investigación sobre sistemas de localización en interiores utilizando Bluetooth		15h
<i>Tareas</i>	Tecnologías que utilizan	2.5h
	<i>Smoothing</i> localización	2.5h
	Algoritmos de localización	5h
	Protocolos de comunicación	5h
Familiarizarse con las tecnologías		15h
<i>Tarea</i>	Familiarizarse con el uso de balizas	15h
Generar aplicación de mantenimiento de balizas		21h
<i>Tareas</i>	Exploración SDK	2h
	Desarrollo de la aplicación	18h
	Testeo	1h
Escenario teórico de posicionamiento		51h
<i>Tareas</i>	Investigar o seleccionar los algoritmos de localización	10h
	Investigar o seleccionar los dispositivos	10h
	Montar escenario Prueba	20h
	Testear soluciones desarrolladas	6h
	Conclusiones	5h
Pruebas en escenario real		120h
<i>Tareas</i>	Revisión del dispositivo localizador	20h
	Revisión del algoritmo de localización	20h
	Revisión de las balizas	20h
	Implementación de las mejoras encontradas	40h
	Análisis de los resultados	20h
Conclusiones y cierres		78h
Total		300h

Cuadro 3.1: Estimación de la planificación inicial.

- Recursos humanos
 - 300 horas x 8,67 €/hora=2600,74 €
- Recursos materiales
 - Portátil Acer TravelMate P215-53 = 759 €, suponiendo que su vida útil sea de cinco años, supondría 12,65 €/mes
 - Balizas iBks-Plus 150 €

Concepto	Unidades	Precio unitario	Total
Programador	300 horas	8,67 €	2600,74 €
Portátil	3 meses	12,65 €/mes	37,95 €
Balizas iBks-Plus	1	150 €	150 €
Total			2788,69 €

Cuadro 3.2: Costes del proyecto

En el Cuadro 3.2, se observa el desglose de los recursos humanos y materiales utilizados en el desarrollo de este proyecto.

3.4. Seguimiento del proyecto

En este apartado se detalla como se ha realizado el seguimiento y el control del proyecto, así como se han ido resolviendo las desviaciones para reducir su impacto de la planificación inicial, explicada en el apartado 3.2

El seguimiento del proyecto se ha realizado por parte de la empresa con el equipo de desarrollo todos los jueves y por parte de la Universidad enviando informes quincenales a mi tutora de proyecto.

Durante la primera semana se empezó con una pequeña investigación de los sistemas de localización Bluetooth que existen actualmente en el mercado, a la siguiente semana después de una breve presentación del resultado de la investigación se pasó a familiarizarse con las balizas iBks-Plus y su respectivo SDK.

Todo iba según lo planeado, hasta que se empezó a desarrollar la aplicación de mantenimiento de balizas. A la hora de desarrollar la aplicación en *Android Studio* surgió una incompatibilidad entre las librerías del fabricante y la versión del SDK de *Android Studio*, no había forma que listar los dispositivos BLE (*Bluetooth Low Energy*). Hasta que se encontró que el problema era una incompatibilidad pasó más tiempo de lo planificado. Una vez solucionado este problema, se pasó a desarrollar la versión inicial de la aplicación que permitía leer y configurar los slots UID y TLM de la baliza con las especificaciones indicadas en un fichero *JSON*.

Se tenían que configurar más de cien balizas, con las características indicadas por el supervisor del proyecto en la empresa. Nos dimos cuenta que era necesario realizar otra versión de la aplicación que nos permitiera listar las balizas en orden según el *RSSI*, la baliza más cercana se indicaría con

un menor *RSSI* y sería la que iría primera en la lista. También hubo otros cambios que se indicarán en los apartados siguientes y otros problemas que se han tenido que solucionar.

Por otro lado, en la fase de investigación, el supervisor del proyecto de la empresa se mostró interesado en los RTLS de Qupa y otros RTLS con BLE 5.1. Todo parecía que podíamos llegar a tiempo a los objetivos propuestos, pero se vino un gran problema, los proveedores que se encontraron querían un caso de uso real. En este punto es necesario indicar que esta parte es un submundo, todo lo contrario a la vida real que cualquier empresa estuviera deseosa de vender su producto, en ese submundo es todo lo contrario, el proveedor estudia si el cliente está dentro del tipo de clientes que quiere para su empresa. En esta fase el tiempo se fue entre búsqueda de proveedores, envíos de correos y entrevistas. Se puede decir, que esta fase fue de las más estresante porque se veía que el tiempo se nos venía encima.

Al final, hubiese sido mejor plantearse desde un principio el desarrollar el hardware y software necesario para implementar un RTLS con dispositivos BLE 5.1. Finalmente el supervisor de la empresa decidió que con la investigación, la aplicación de configuración de balizas y el manual de la aplicación ya se tenía material suficiente para redactar el TFG. La planificación ha quedado como se muestra en el Cuadro 3.3

Planificación final		
Sprint		Horas
Investigación sobre sistemas de localización en interiores utilizando Bluetooth		15h
<i>Tareas</i>	Tecnologías que utilizan	2.5h
	<i>Smoothing</i> localización	2.5h
	Algoritmos de localización	5h
	Protocolos de comunicación	5h
Familiarizarse con las tecnologías		15h
<i>Tarea</i>	Familiarizarse con el uso de balizas	15h
Generar aplicación de mantenimiento de balizas		109h
<i>Tareas</i>	Exploración SDK	19h
	Desarrollo de la aplicación	80h
	Testeo	10h
Escenario teórico de posicionamiento de las balizas		26h
<i>Tarea</i>	Buscar y/o estudiar el o los receptores	26h
Cambios en la aplicación BeaconManager		57h
<i>Tareas</i>	Modificar el name Dialog	0.5h
	Orden de la lista por intensidad RSSI	0.5h
	Resaltar las balizas conectables y sombrear las balizas no conectables	1h
	Solucionar el problema del Bluetooth scan	19h
	Redacción y revisiones del manual de la aplicación	36h
Conclusiones y cierres		78h
Total		300h

Cuadro 3.3: Estimación en horas de la planificación final del proyecto.

Capítulo 4

Análisis y diseño del sistema

Índice

4.1. Contexto	29
4.2. Estudio previo de la App del fabricante	31
4.3. Análisis del sistema	33
4.3.1. Requisitos del proyecto	33
4.4. Casos de uso	33
4.5. Análisis de las tecnologías	36
4.5.1. Baliza iBKS Plus	36
4.5.2. Protocolo Eddystone	37
4.6. Diseño de la arquitectura del sistema	40

En este capítulo antes de analizar y diseñar el sistema, se considera necesario explicar el propósito de diseñar una aplicación que configure balizas. Después, se va a hacer un estudio previo de la aplicación de configuración de balizas del proveedor que la empresa ya había elegido. Luego se explicará el desarrollo de una aplicación más sencilla para los usuarios a la hora de configurar o simplemente leer la información de estas. Esta nueva aplicación se denominará *BeaconManager*, se definirán los requisitos y los casos de uso, además de la arquitectura del sistema y cómo se comunican los dispositivos que lo integran.

4.1. Contexto

La aplicación que se ha desarrollado se aplicará a un caso real y es el complemento de un proyecto que fue desarrollado por la empresa **720tec**. Este proyecto se titula *Identificación de camiones por balizas Bluetooth*. Desde el punto de vista de su funcionalidad, se pretende que mediante la instalación de balizas *Bluetooth* en los vehículos, sea capaz de identificar por proximidad esta grúa desde un camión a través de una aplicación instalada en un teléfono móvil.

La empresa ha tenido que balizar gran parte del entorno (portainer, transtainer, puertas por horas, carretillas) con balizas que deben estar previamente configuradas. Utilizar la aplicación del propio

fabricante es algo complejo. La empresa cree necesario desarrollar una aplicación más sencilla, sobre todo por el hecho de que el usuario principal es un personal de mantenimiento que solo tendrá que instalar la baliza en los puntos indicados y configurar el nombre de la misma según el tipo de vehículo donde ha sido instalada. Las balizas se instalarán en los siguientes vehículos:

- Portainers P03, P04, P116 con doble baliza, vease Figura 4.1.
- Transtainers RT02, RT11, RT30 con doble baliza, vease Figura 4.2.
- Carretillas RS71, CV56 baliza simple, vease Figura 4.4 y Figura 4.5.
- Puerta por horas con doble baliza, vease Figura 4.3.

El nombre de las balizas se forma a partir del nombre del vehículo guión y un número (0,1,2) que identificará su posición. Si el vehículo solo se equipa con una baliza, el número será el 0 (RS71-0, CV56-0). Aquellos vehículos o puertas que se equipan con dos balizas se formarán con el nombre del vehículo guión y el número 1 y 2 (P03-1, P03-2. Véase Figura 4.1). Siendo la baliza 1 la situada a la derecha de la grúa, mirando la grúa siempre desde la posición del camión.



Figura 4.1: Portainer



Figura 4.2: Transtainer

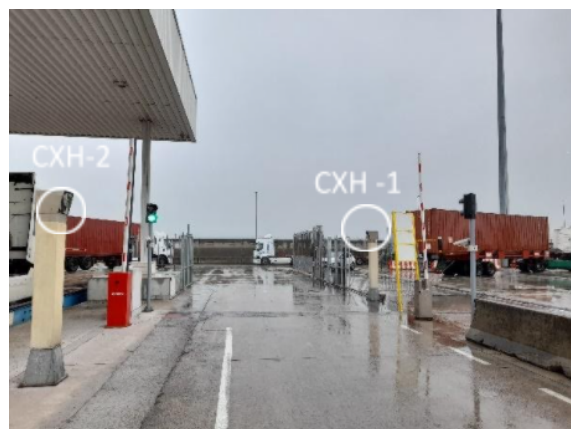


Figura 4.3: Puerta por horas



Figura 4.4: Carretilla tipo CV



Figura 4.5: Carretilla tipo RS

La nomenclatura es importante, esta información será emitida por el slot UID. La idea es que cuando el camión pase (con la aplicación instalada en el móvil del camionero) sabrá en que posición está; es decir si está cerca de cualquiera de los vehículos antes mencionados.

4.2. Estudio previo de la App del fabricante

Antes de pasar a analizar el sistema que se va a desarrollar es necesario indicar que la baliza ya había sido elegida por la empresa **720tec**, ellos decidieron elegir las balizas iBKS-Plus Figura 4.6 de la empresa Accent Systems.



Figura 4.6: Baliza iBKS-Plus

La baliza iBKS-Plus cuenta con una aplicación del fabricante tanto para *iOS* como para *Android*. Esta *App* permite configurar las balizas, se denomina *iBKS Config Tool*. Entre las funciones de esta aplicación podemos encontrar:

- Listar todas las balizas disponibles alrededor y ver la dirección MAC del dispositivo, los paquetes de publicidad en tiempo real y si la baliza está en modo conectable o no, vease Figura 4.7.
- Conectarse a los dispositivos de baliza iBKS que están en modo conectable y leer las siguientes características: slots de iBeacon, slots de Eddystone, nombre del dispositivo, contraseña, modos conectable y restablecimiento de fábrica, vease Figura 4.8.
- Configurar los slots de iBeacon o de Eddystone, vease Figura 4.9

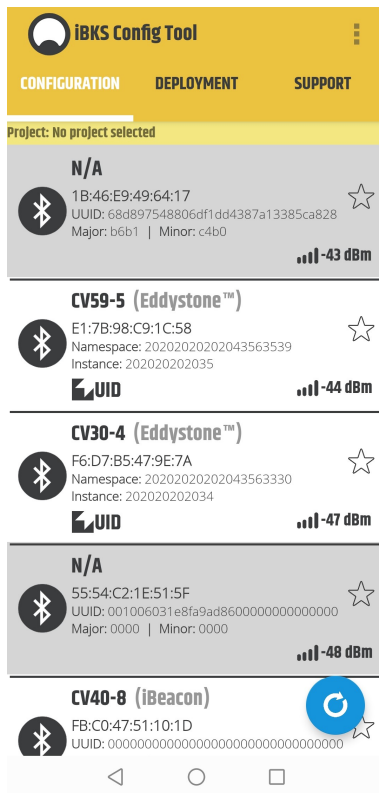


Figura 4.7: Listar dispositivos cercanos

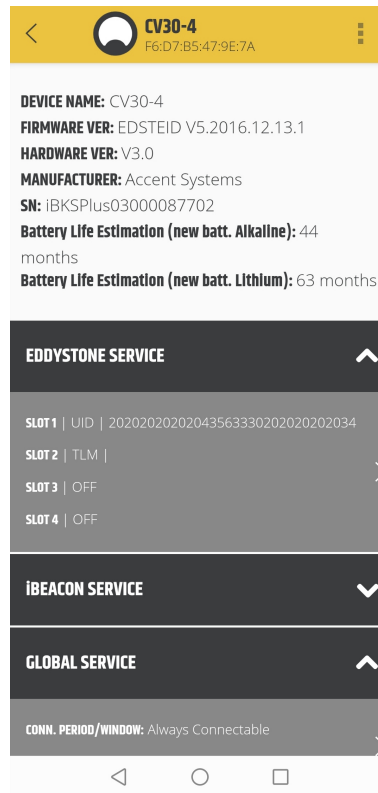


Figura 4.8: Información detallada de una baliza

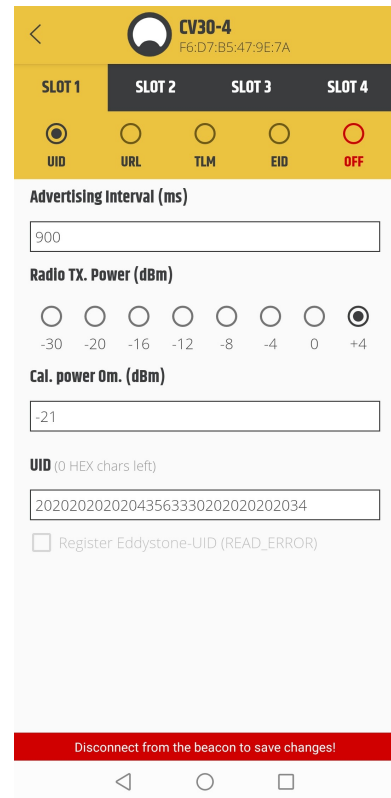


Figura 4.9: Configurar slots

Los motivos por los que la empresa no va a utilizar la aplicación del fabricante son primero a que se necesita tener conocimientos para saber si hay que configurar los slots de un servicio u otro, y segundo que se debe saber que datos se van a modificar y que significa cada parámetro que se está modificando. Los jefes de la empresa *720tec* decidieron que era necesario desarrollar una aplicación más sencilla, ya que va destinada a usuarios de mantenimiento, que simplemente instalarán las balizas en los puntos determinados, con la configuración de la nomenclatura indicada según el lugar de donde será instalada. Además solo trabajarían con los slots *UID* y *TLM* de Eddystone.

4.3. Análisis del sistema

En este apartado se realizará el modelado del sistema. Se detallaran los requisitos del sistema, los casos de uso y los usuarios del sistema.

4.3.1. Requisitos del proyecto

Los requisitos del proyecto son las características que debe cumplir este sistema para cumplir los objetivos esperados. Teniendo conocimiento del tipo de usuario que va a hacer uso de esta aplicación y de las necesidades que tienen que cubrir, los requisitos que tiene que cumplir este sistema son los siguientes:

- Listar todos las balizas disponibles en los alrededores.
- Mostrar la lista de balizas en orden de intensidad de señal RSSI.
- Indicar cuando una baliza está en modo no conectable o conectable.
- Conectarse a una baliza.
- Leer información de una baliza.
- Mostrar información de una baliza.
- Configurar una baliza con la nomenclatura indicada en la sección de propósito y con todos los parámetros indicados en un archivo JSON.
- Modificar una baliza que está en modo no conectable a conectable, y que esto solo pueda hacerlo el administrador.

4.4. Casos de uso

Una vez definido los requisitos, se puede identificar los casos de uso [2] y los actores que intervienen en el sistema. A primera vista, se puede identificar dos actores, el primero el trabajador de mantenimiento y el segundo es el administrador. Este puede realizar todos los casos de uso del trabajador de mantenimiento, solo que este puede identificarse y cambiar una baliza de modo no conectable a conectable.

Este sistema consta de cinco casos de uso, son los siguientes:

- Listar dispositivos BLE.
- Ver información detallada de la baliza.
- Configurar baliza.

- Modificar el estado de la baliza de no conectable a conectable.

En la Figura 4.10, se muestran los actores y su correspondiente relación con los casos de uso. Después, se detalla cada caso de uso.

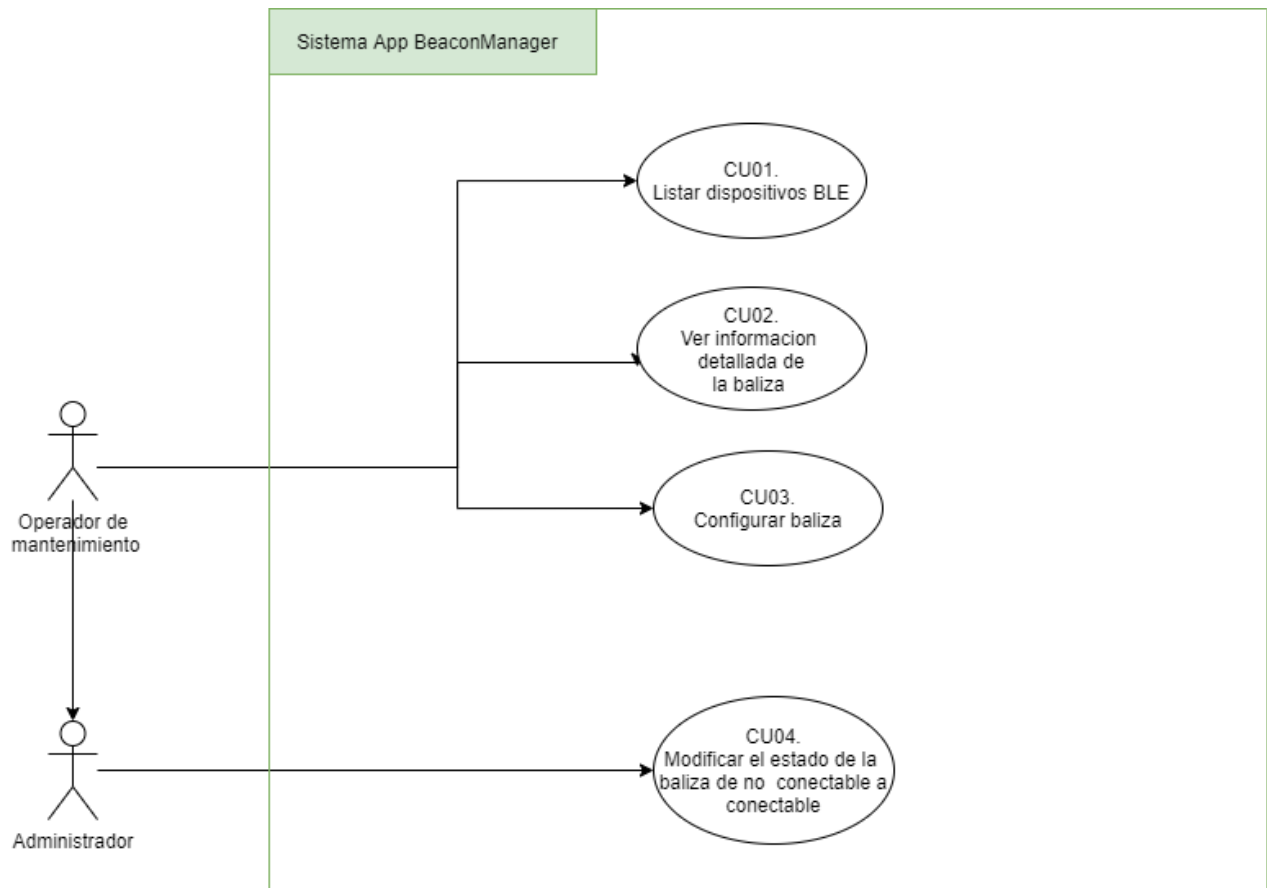


Figura 4.10: Diagrama de casos de uso

Listar dispositivos BLE	
Identificador	CU01
Nombre	Listar dispositivos BLE.
Descripción	El sistema muestra todos los dispositivos BLE que se encuentran a su alcance.
Actor principal	Operador de mantenimiento.
Secuencia de pasos	<ol style="list-style-type: none"> 1 Escanear dispositivos Bluetooth. 2 Filtrar dispositivos BLE y los que emiten tramas Eddystone. 3 Mostrar en cada ítem de la lista información simple de las balizas, como MAC, nombre RSSI. 4 Reordenar por intensidad de la señal RSSI. 5 Pintar la lista; resaltar el primer elemento de la lista de color verde si es conectable y rojo si es no conectable. Además los dispositivos no conectables se verán en gris.

Ver información detallada de la baliza	
Identificador	CU02
Nombre	Ver información detallada de la baliza.
Descripción	El sistema debe permitir seleccionar una baliza, conectarse a esta y mostrar toda su información.
Actor principal	Operador de mantenimiento.
Secuencia de pasos	
	1 Seleccionar un baliza de la lista.
	2 Establecer conexión con la baliza seleccionada.
	3 Leer toda la información a mostrar de la baliza.
	4 Mostrar toda la información leída.
Condición previa	El usuario ha tenido que completar el caso de uso CU01.

Configurar baliza	
Identificador	CU03
Nombre	Configurar baliza.
Descripción	El sistema debe permitir modificar el nombre de la baliza.
Actor principal	Operador de mantenimiento.
Secuencia de pasos	
	1 Seleccionar la opción configurar baliza.
	2 Introducir el nuevo nombre de la baliza según la nomenclatura indicada.
	3 Seleccionar <i>ok</i> .
	4 Se escribe el nuevo nombre de la baliza.
	5 Se modifica el estado de conexión de la baliza, de conectable a no conectable.
Condición Previa	El usuario ha tenido que completar el caso de uso CU02.

Modificar el estado de la baliza de no conectable a conectable	
Identificador	CU04
Nombre	Modificar el estado de la baliza de no conectable a conectable.
Descripción	El sistema permite modificar el estado de conexión de no conectable a conectable, solo al usuario que tenga las credenciales de administrador (el usuario y la contraseña).
Actor principal	Administrador.
Secuencia de pasos	
	1 Selecciona en el menú la opción conectable.
	2 El usuario escribe sus credenciales y se identifica.
	3 El sistema comprueba las credenciales.
	4 Si las credenciales son correctas modifica el estado de conexión de la baliza de no conectable a conectable, sino se queda en el mismo estado.
Condición previa	El usuario ha tenido que completar el caso de uso CU02.

4.5. Análisis de las tecnologías

En esta sección lo que se abordará es analizar el funcionamiento tanto de la baliza como el del protocolo *Eddystone*, lo que es cada uno ya se ha explicado en el Capítulo 2.

4.5.1. Baliza iBKS Plus

En el Capítulo 2 definimos lo que es una baliza, en este apartado estudiaremos más a fondo la baliza iBKS Plus¹, sus características son las siguientes:

Dimensiones	84 x 84 x 24 mm
Chip	Nordic nRF51822
Protocolo de radio	Bluetooth Low Energy (BLE)
Rango de distancia	Hasta 100m
Batería	4 AA
Consumo de corriente en reposo	3,5 a 3,8 μA
Protocolos de baliza	iBeacon, Eddystone: UID, URL, TLM y EID

El chip *Nordic nRF51822*² es un sistema en chip (SoC) inalámbrico de 2,4 GHz de muy bajo consumo que integra el transceptor de 2,4 GHz de la serie nRF51, usa una CPU ARM Cortex M0 de 32 bits, junto con una amplia disponibilidad de flash, 256kB/128kB en total con 40kB-180kB disponibles para el desarrollo de aplicaciones. La densidad de código y la velocidad de ejecución son considerablemente mayores que en las plataformas de 8/16 bits. El sistema de interconexión de periféricos programable (PPI) proporciona un bus de 16 canales para la comunicación directa y autónoma de los periféricos del sistema sin intervención de la CPU. Esto aporta tiempos de latencia predecibles para la interacción entre periféricos y las ventajas de ahorro de energía asociadas a dejar la CPU inactiva. En la Figura 4.11 podemos ver el chip nRF51822 y en la Figura 4.12 la alimentación para el iBKS Plus.

El dispositivo dispone de 2 modos de alimentación global ON/OFF, pero todos los bloques del sistema y los periféricos tienen un control de gestión de la energía individual que permite una conmutación automática RUN/IDLE para los bloques del sistema basada sólo en los que se necesitan/no se necesitan para realizar tareas concretas. La nueva radio constituye la base del rendimiento del nRF51822. La radio soporta Bluetooth de baja energía y es compatible en el aire con los productos de la serie nRF24L de Nordic Semiconductor. La potencia de salida es ahora escalable desde un máximo de +4dBm hasta -20dBm en pasos de 4dB. La sensibilidad se incrementa en cada nivel y ofrece rangos de sensibilidad (dependiendo de la velocidad de datos) de -96 a -85dBm, con -93dBm para Bluetooth de baja energía.

¹ *Datasheet iBKS Plus*: https://accent-systems.com/wp-content/uploads/iBKS_Plus_datasheet_rev2.pdf

² *Datasheet Nordic nRF51822*: https://infocenter.nordicsemi.com/pdf/nRF51822_PS_v3.1.pdf

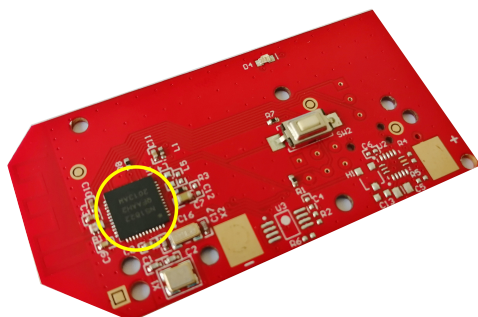


Figura 4.11: Chip nRF51822



Figura 4.12: Pilas iBKS Plus

La estimación de la duración de la batería dependerá de:

- Si trabajará con los dos protocolo iBeacon y/o Eddystone.
- El número de slots que estén activos.
- La potencia y el periodo de publicidad definido en cada slot.
- si la baliza se quedará en estado conectable o no conectable.

En el caso de la aplicación a desarrollar se trabajará solo con el protocolo Eddystone y cuando se configure una baliza, esta se quedará en estado no conectable, esto quiere decir que en el caso de querer volver a configurar la baliza, simplemente se tendrá que apagar la baliza y volver a encenderla. Cuando se vuelve a encender, la baliza se encontrará 30" en modo conectable, tiempo suficiente para conectarse a esa baliza y volver a configurar o cambiar su estado de no conectable a conectable, todo esto está explicado en el Anexo, en la sección Manual App BeaconManager. Además es necesario indicar que cuando se modifica el nombre de un dispositivo, en algunos móviles cuando se vuelve a listar, se lista con el nombre nuevo, pero en otros es necesario encender y apagar la baliza.

4.5.2. Protocolo Eddystone

Eddystone es un protocolo de balizas de Google, que permite a los dispositivos *Bluetooth Low Energy* transmitir datos a través de distancias cortas a un receptor. Las principales ventajas de este protocolo sobre el resto de los existentes es que permite establecer y enviar cuatro tipos de trama:

- *Eddystone-UID*: Un ID único y estático con un componente de espacio de nombres de 10 bytes y un componente de instancia de 6 bytes.

- *Eddystone-URL*: Una URL comprimida que, una vez analizada y descomprimida, es directamente utilizable por el cliente.
- *Eddystone-TLM*: Datos de estado de la baliza que son útiles para el mantenimiento de la flota de balizas y alimentan el punto final de diagnóstico de la API de balizas de proximidad de Google.
- *Eddystone-EID*: Una trama de baliza variable en el tiempo que puede ser resuelta a un identificador estable por un resolvidor vinculado, como Proximity Beacon API.

Teniendo claro la definición de los requisitos funcionales y la tecnología a utilizar para llevarlas a cabo, es necesario tener claro las acciones que se tienen que realizar para la consecución de cada uno de ellos. Véase Figura 4.13, Figura 4.14

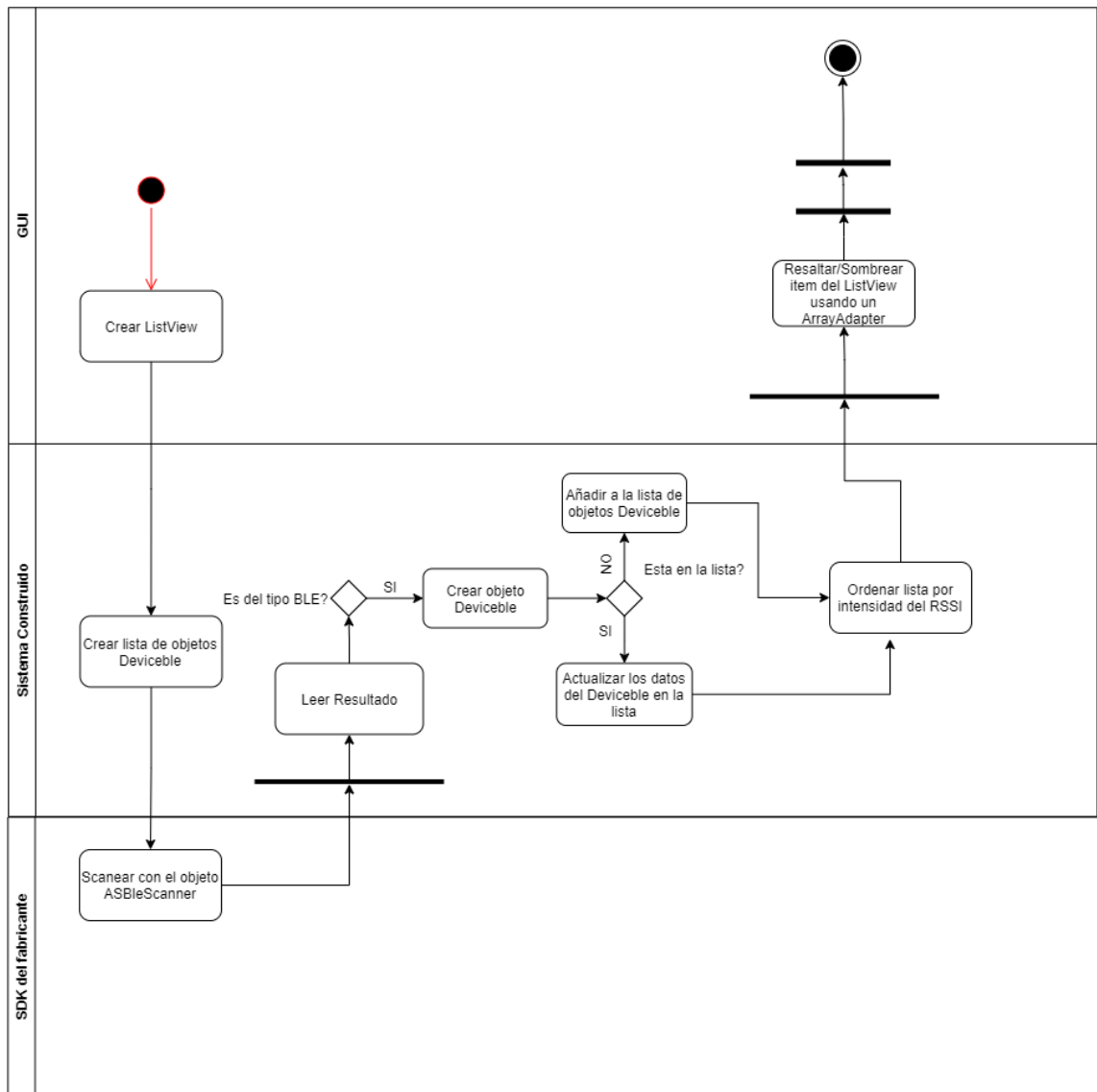


Figura 4.13: Acciones para listar dispositivos

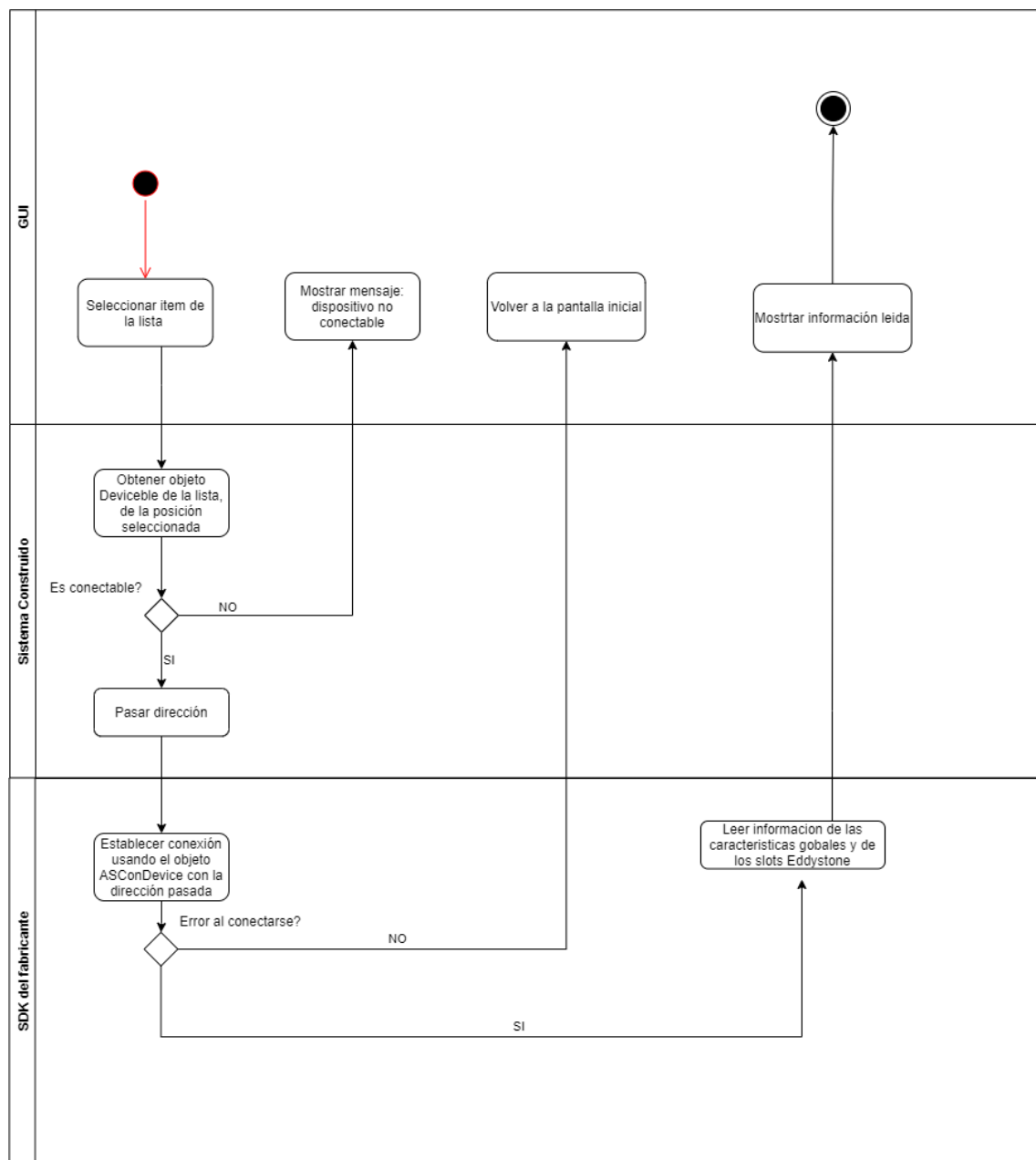


Figura 4.14: Acciones para leer información de la baliza

4.6. Diseño de la arquitectura del sistema

Para comprender mejor la interacción entre los componentes especificados anteriormente, se muestra en la Figura 4.15 un esquema general que detalla la estructura del sistema. Por un lado tenemos las n balizas en el entorno y por otro la aplicación. La aplicación puede listar todas las balizas que están a su alcance, pero solo puede establecer conexión con una a la vez.

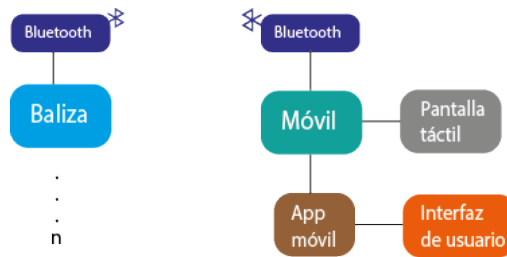


Figura 4.15: Estructura global del sistema

Capítulo 5

Implementación y pruebas

Índice

5.1. Detalles de implementación	41
5.1.1. Aplicación BeaconManager	41

En este capítulo se detallan las decisiones que se han tomado a lo largo de la implementación y desarrollo de la aplicación *BeaconManager*. También se indicará los problemas encontrados al testear esta aplicación en móviles de diferentes marcas.

5.1. Detalles de implementación

Como se ha comentado en el capítulo 3, este apartado corresponderá con los siguientes *sprints*: familiarizarse con la tecnologías, generar la aplicación de mantenimiento de balizas y cambios en la aplicación, este último es un sprint resultado de la modificación de la planificación.

5.1.1. Aplicación BeaconManager

Para la implementación de esta aplicación se va a utilizar el IDE *Anroid Studio*, véase la estructura del proyecto Figura 5.1 . Este usa el patrón Modelo - Vista - Controlador (MVC):

- Modelo: Aquí van las diferentes representaciones creadas de los datos, como por ejemplo, en este caso irían las clases *DeviceBle*, *SlotsEddy*, *EddystoneSlots*, *Confindialog* y *LoginDialog*.
- Vista: Aquí se refiere a las interfaces con las que interactuará el usuario, en este caso sería todos los *layouts*.
- Controlador: Aquí va la lógica de la aplicación, se encarga de responder a eventos, peticiones de la vista por parte del usuario, como por ejemplo, en este caso irían *MainActivity*, *BeaconActivity* y *DevicebleAdapter*.

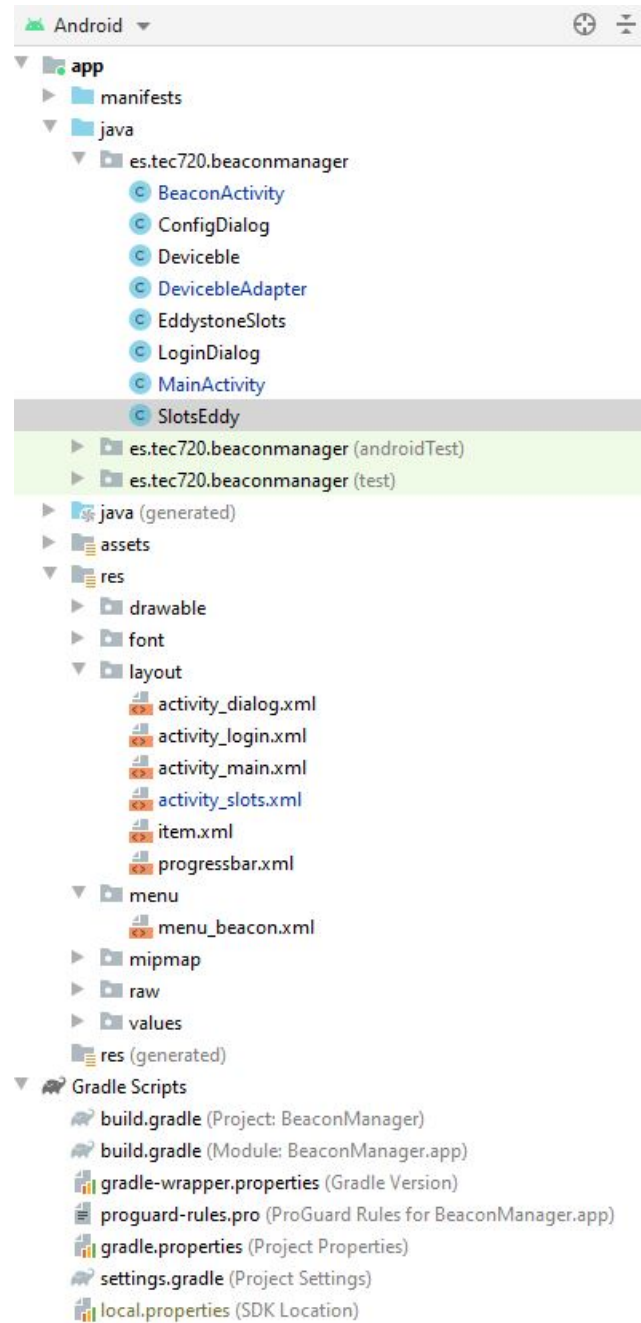


Figura 5.1: Estructura del proyecto

Al desarrollar la aplicación, me encontré con un problema que me hizo perder muchísimo tiempo, no había forma de listar las balizas cercanas. Resultó ser un problema de dependencias en el gradle [4] y fallo en el adaptador *Bluetooth*, además de tener muy poca experiencia trabajando con interfaces de *Android* [3], por lo que ha sido necesario consultar manuales.

Una vez superado el problema anterior, por fin se logró listar los distintos dispositivos *BLE* cercanos, no se tenía una interfaz, simplemente se mostraba por consola. En si se estaba teniendo una toma de contacto con el *SDK*¹ del fabricante *Accents Systems*

En un principio se quería listar todos los dispositivos cercanos (Figura 5.2), seleccionar una baliza, conectarse, leer todos los slots (Figura 5.3) y configurar todos los slots.

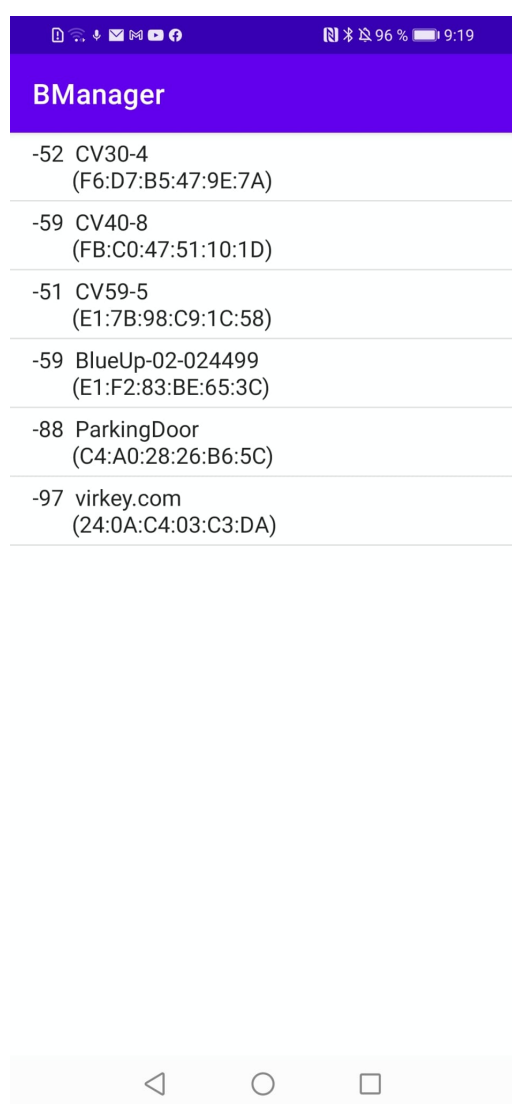


Figura 5.2: Listar

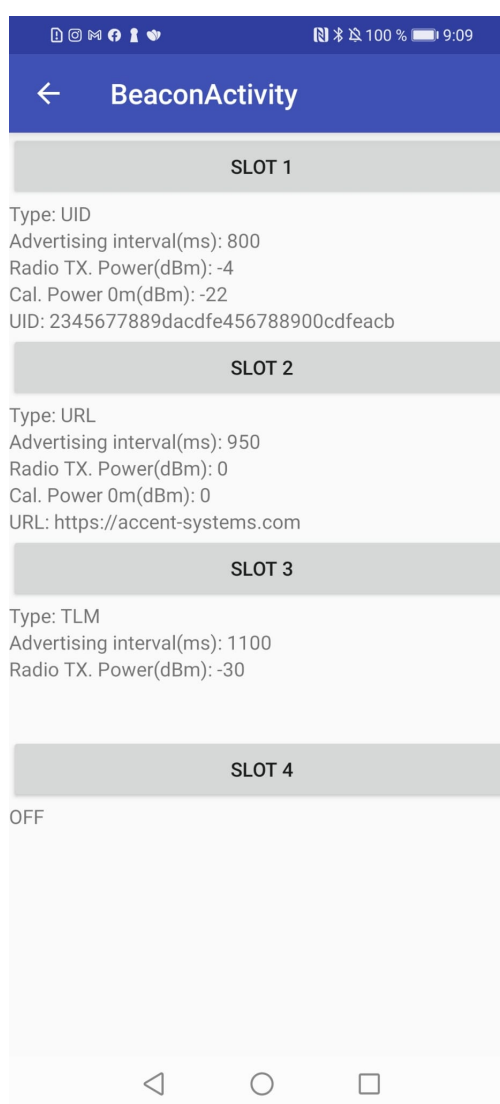


Figura 5.3: Leer slots

¹SDK *iBKS Plus*: <https://github.com/accentystems/iBKS-SDK-Android/blob/master/doc>

Después de varias reuniones, se llegó al punto de que no se quería una aplicación donde se pudiera configurar uno a uno los parámetros, eso ya nos lo ofrecía la aplicación del fabricante. Se decidió crear un fichero *JSON*, donde estaban ya todos los parámetros establecidos. El usuario solo tenía que configurar el nombre del dispositivo de acuerdo a una nomenclatura que ya va explicada en el manual de la aplicación.

Se logró llegar a una versión inicial del programa, de hecho sirvió para configurar 100 balizas, luego fueron enviadas a una empresa de Bilbao

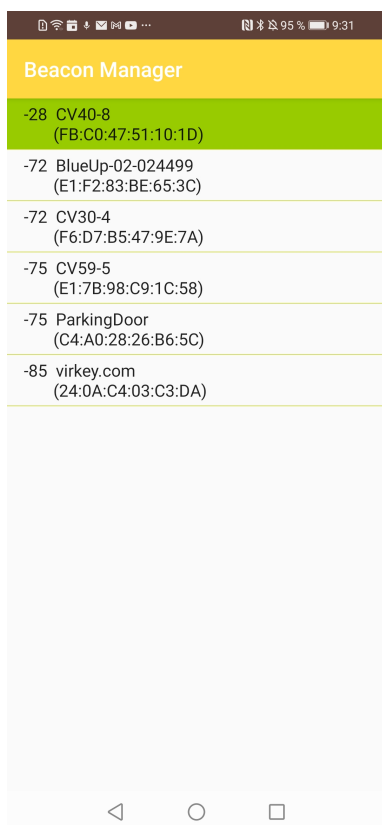


Figura 5.4: Listar dispositivos cercanos

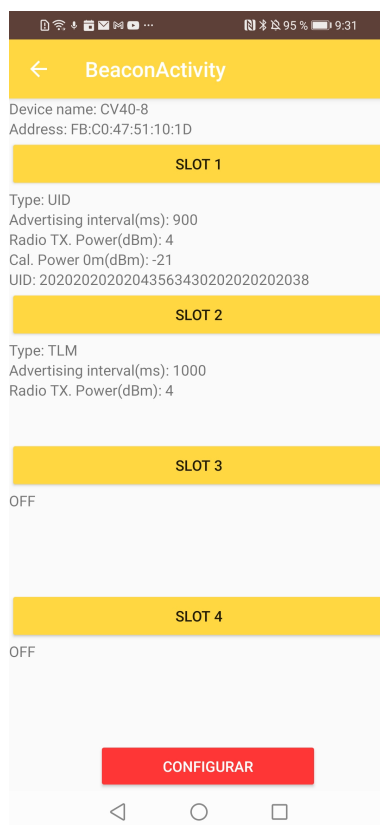


Figura 5.5: Información detallada de una baliza

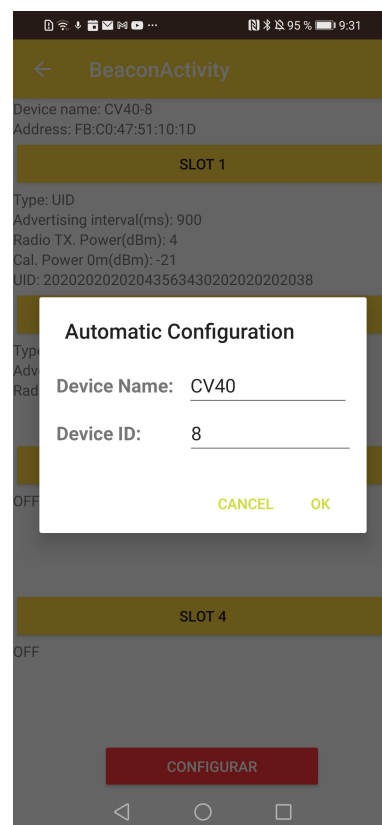


Figura 5.6: Configurar slots

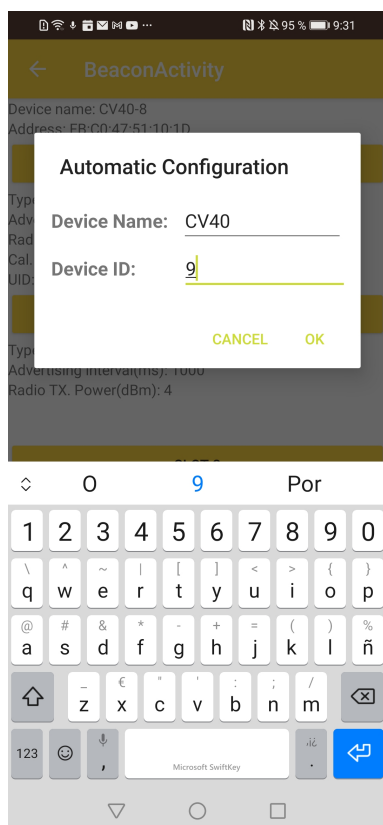


Figura 5.7: Modificar nombre de la baliza

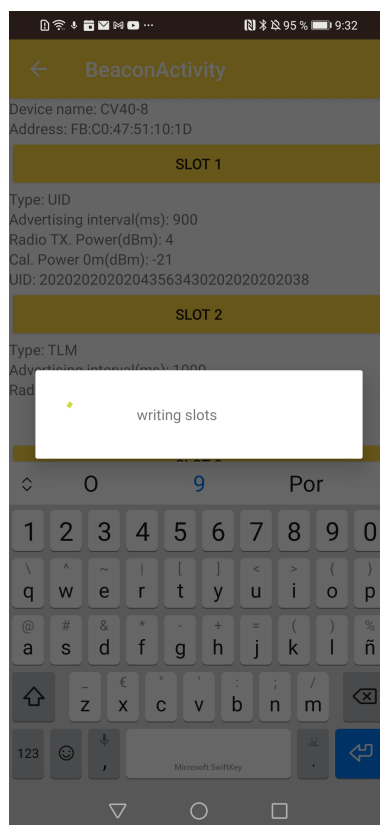


Figura 5.8: Escribiendo en los slots correspondientes

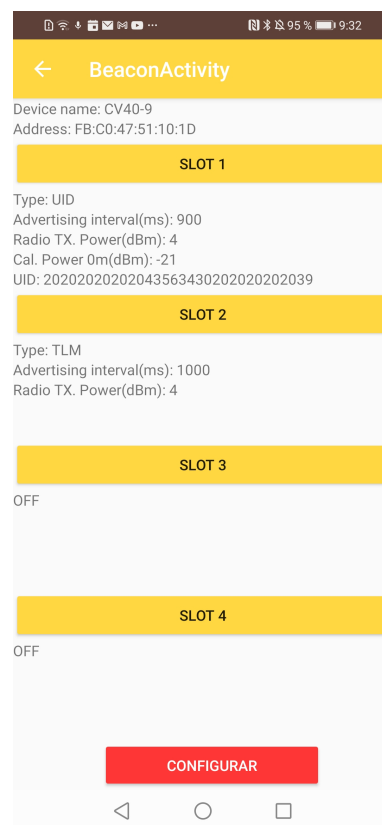


Figura 5.9: Modificación efectuada

Se hicieron pruebas en varias reuniones, la aplicación funcionó, pero estaba la duda de que si realmente estamos configurando la baliza más cercana, aunque los leds se encienden al desconectarlos después de haber realizado alguna modificación en la baliza.

Por otro lado, lo que se decidió en una de las reuniones fue que después de configurar, modifiquemos la característica de conectable a no conectable, además esta opción será añadida en el menú, donde solo el usuario con las credenciales de administrador será el que pueda modificar esta especificación global; de no conectable a conectable, de esta forma nos aseguramos que no cualquiera pueda realizar este cambio.

Cuando hablamos de modo connectable, quiere decir que la aplicación está desarrollada de tal manera que cuando alguien configura, es decir cambia el nombre de la baliza, este se queda en modo *no connectable*, para ello sigue los siguientes pasos:

1. Se borran todos los slots anteriores.
2. Del nuevo nombre y el id introducido se obtiene el dato *UID*.
3. Se configura el slot *UID* y el slot *TLM* con los datos definidos en un fichero. *JSON* más el dato *UID* obtenido en el paso anterior.
4. Se deja el *beacon* en modo *no connectable*.

Dejar en modo *no connectable* implica que no puedes establecer conexión con dicha baliza, sería necesario apagar y volver a encender, teniendo solo treinta segundos desde que lo enciendes para poder establecer conexión con la baliza. La idea de añadir una opción que permita modificar el modo a *connectable*, pero como es algo que no lo debe hacer cualquiera se añadió esta opción en el menú, pulsando en los tres puntos se tiene la opción de modificar el modo, pero para ello tendrías que introducir las credenciales de administrador.

Se ha realizado una serie de pruebas, entre ellas se ha encontrado que la App ***BeaconManager*** se comporta de forma muy distinta entre un Huawei, un Samsung Galaxy A21s y un Xiaomi MI MIX 2. En principio cuando el escaneo era en baja latencia, en los móviles Huawei no es necesario hacer un stop para luego seleccionar la baliza, basta con seleccionar la baliza de la lista y se muestra su configuración, en cambio en el Samsung Galaxy A21s y un Xiaomi MI MIX 2 es necesario tener un botón para pausar el escaneo y a continuación seleccionar la baliza de la lista. De hecho la aplicación del fabricante te obliga a pausar el escaneo para poder seleccionar un *item* de la lista. Este problema lo hablé con el supervisor y posiblemente uno de los factores que influye es la velocidad a la hora de escanear y mostrar en UI, cuando pulsas sobre un elemento de la lista es como si no detectará nada. Hablamos con el fabricante y cambiamos el tipo de escaneo a balanceado y añadimos una funcionalidad más al botón de actualizar lista; es decir, la primera vez que pulsas el botón, se pausa el escaneo y la segunda vez actualiza la lista.

Capítulo 6

Conclusiones

En estos tres meses de estancia en prácticas, se ha conseguido el objetivo secundario de desarrollar una aplicación de aprovisionamiento y mantenimiento de balizas. Del objetivo principal, de desarrollar un sistema de localización con dispositivos *Bluetooth*, se llevó a cabo una investigación y búsqueda de proveedores, pero por temas de ajenos a mí, no logramos obtener un proveedor de esta tecnología en tiempo de pandemia. La búsqueda de proveedores para este tipo de tecnología es un submundo con todas las complicaciones que eso conlleva. Considero que hubiera sido mejor, desarrollar también el hardware porque es más fácil comprar un componente porque no tienes que pasar por filtros.

En el ámbito formativo, este proyecto se ha realizado en el IDE de *Android*, utilizando el lenguaje de programación *Java*. Considero que lo más complicado a lo que me he enfrentado en este proyecto ha sido el *gradle* de *Android*, me hizo perder mucho tiempo porque por culpa de una dependencia no me listaba las balizas. Y por otro lado programar las interfaces en *Android*, en un principio costó, pero a medida que vas cogiendo práctica, todo fluye.

En el ámbito personal, siento que si hubiese tenido ayuda para encontrar un proveedor hubiera llegado a cumplir todos los objetivos del proyecto, pero es algo a lo que una/o en la vida laboral se va a enfrentar tarde o temprano. Por otro lado, el ambiente en la empresa es muy agradable, el trato es cercano, los compañeros son amables.

Bibliografía

- [1] Web Bluetooth. Tecnología bluetooth. <https://www.bluetooth.com/>. [Consulta: 30 de Noviembre de 2020].
- [2] Web Casos de uso. Especificación de casos de uso. <https://sites.google.com/site/sistemadeacademiaelohim/especificacion-de-casos-de-uso>. [Consulta: 10 de Diciembre de 2020].
- [3] Web jtech. Diseño de interfaces gráficas en android. <http://www.jtech.ua.es/dadm/restringido/android/sesion03-apuntes.pdf>. [Consulta: 15 de Diciembre de 2020].
- [4] Web Stackoverflow. Fallo en gradle sync. <https://stackoverflow.com/questions/64409047/gradle-sync-failed-unsupported-method-syncissue-getmultilinemessage-androi>. [Consulta: 20 de Diciembre de 2020].
- [5] Web Xacata. Versiones de bluetooth. <https://www.xataka.com/basics/bluetooth-diferencias-caracteristicas-sus-clases-versiones>. [Consulta: 30 de Noviembre de 2020].

Anexo A

Manual App BeaconManager v1.0



MANUAL
BEACON MANAGER

00
Índice

Instalación	4
Encender/Apagar	6
Conectar con una baliza	9
Configurar	13
Anexo	17

01

Instalación

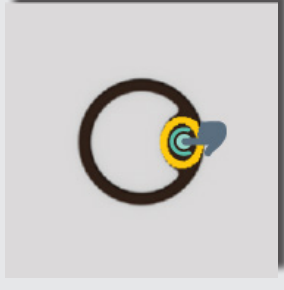
1. Acceder a play store
2. Buscar BeaconManager

Nota: La aplicación BeaconManager requiere como mínimo la versión de android 9 y tener habilitados el bluetooth y location.

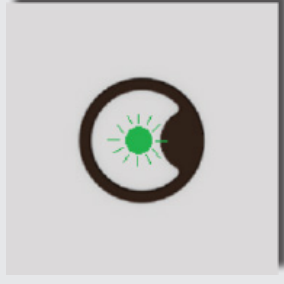
02 Encender/Apagar

1. Encender

1. Pulsar el boton una vez

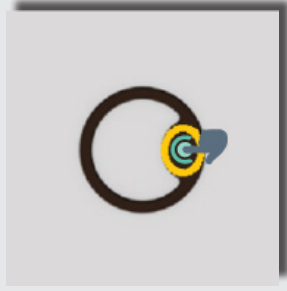


2. El led empezará a parpadear durante 30".
El Tiempo suficiente para poder conectarse a la baliza

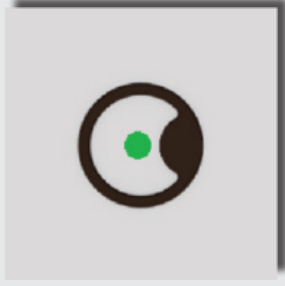


2. Apagar

1. Pulsar el boton una vez



2. El led estará encendido 3" y se apagará



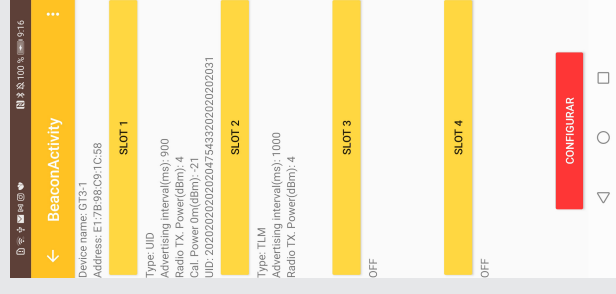
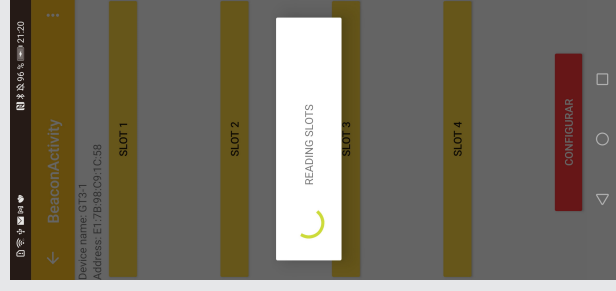
03 Conectar con una baliza

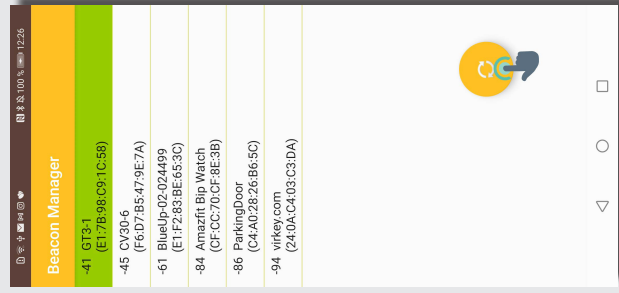
03. Conectar con una baliza



1. Pulsar en la baliza a conectar, conectado!

2. Esperar a que los slots sean leídos



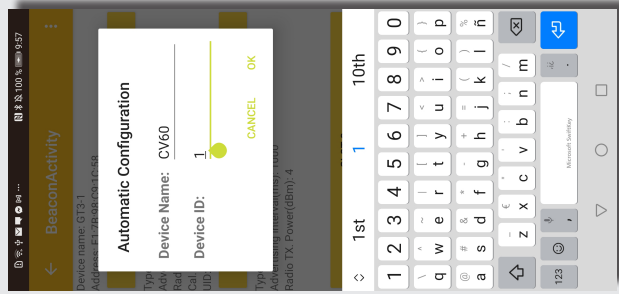
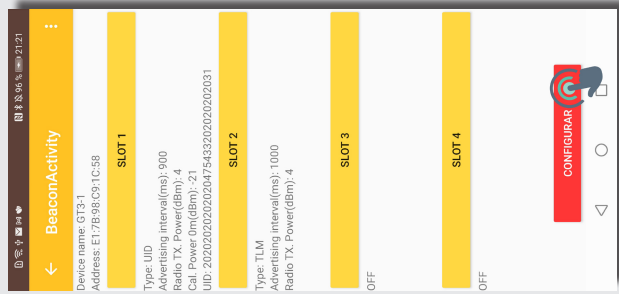


Nota: Si necesitas actualizar el listado de dispositivos, pulsar en el botón circular amarillo.



04 Configurar

04. Configurar



4. Introducir el nombre del dispositivo y la id, tal como se ha explicado en el anexo, en el apartado de nomenclatura.

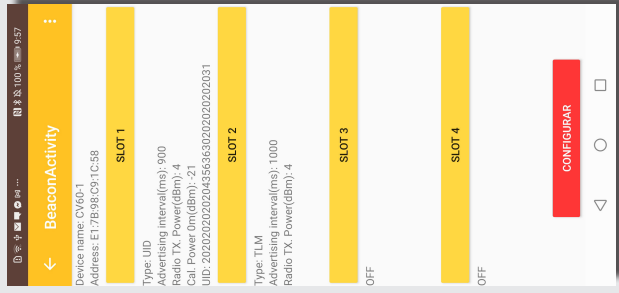
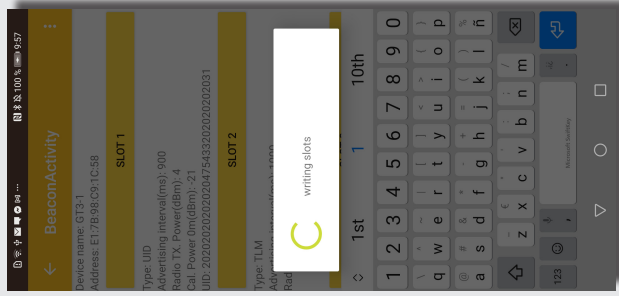
5. Pulsar en "ok"

6. Esperar a que se actualicen los datos introducidos.

1. Abrir la app BeaconManager

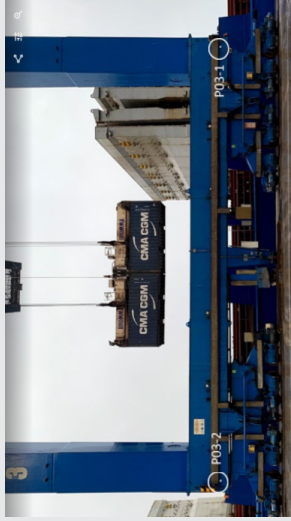
2. Conectar con la baliza a configurar, explicada en el apartado anterior

3. Una vez se ha establecido conexión con la baliza, pulsar en configurar



05 Anexo

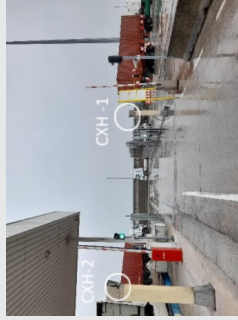
1. Ubicación de balizas



Portainer

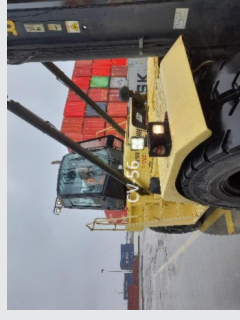


Trastainer



Puerta por horas

Carretillas



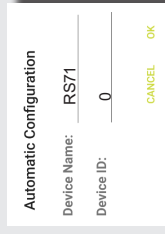
2. Nomenclatura

Antes de pasar a configurar una baliza, es necesario tener en cuenta dónde estará ubicada. En el apartado anterior se ha visto las posibles ubicaciones de las balizas según el tipo de vehículo o puertas.

- Portainers: P03-1, P03-2, con doble baliza.
- Transtainers: RT02-1, RT02-2, con doble baliza.
- Carretillas: RS71-0, CV56-0, con una baliza.
- Puerta por horas CXH-1 CXH-2, con doble baliza.

Las carretillas solo tendrán instalada una baliza, por ejemplo configuramos la carretilla RS71-0

Device Name - Device ID



Automatic Configuration

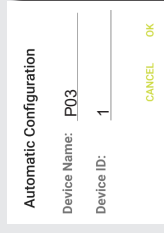
Device Name: RS71

Device ID: 0

CANCEL OK

Aquellos vehículos o puertas que tengan instaladas dos balizas, tendrán que diferenciar entre la baliza de la derecha y la izquierda, mirando la grua desde la posición del camión. El ID de la baliza derecha será 1 y el de la izquierda será 2

Baliza derecha del portainer P03



Automatic Configuration

Device Name: P03

Device ID: 1

CANCEL OK

Baliza izquierda del portainer P03



Automatic Configuration

Device Name: P03

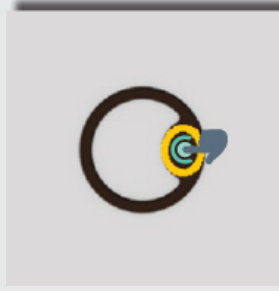
Device ID: 2

CANCEL OK

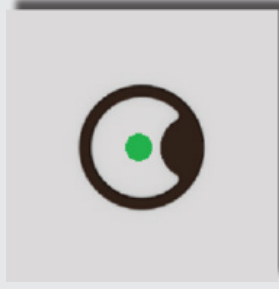
3. Modo Connectable

Cuando una baliza ya ha sido configurada se queda en modo "No Connectable". Si se requiere que la baliza vuelva a estar en modo "Connectable" se debe seguir los siguientes pasos:

1. Apagar



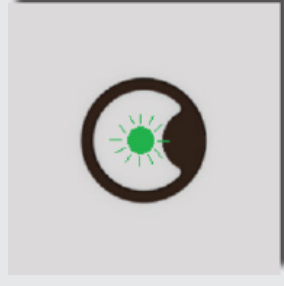
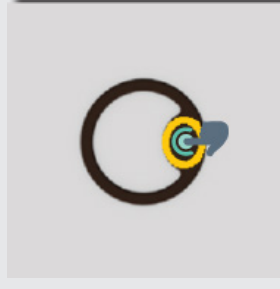
1.1 Pulsar el boton una vez



1.2 El led permanecerá encendido 3"

2. Encender

2.1 Pulsar el boton una vez

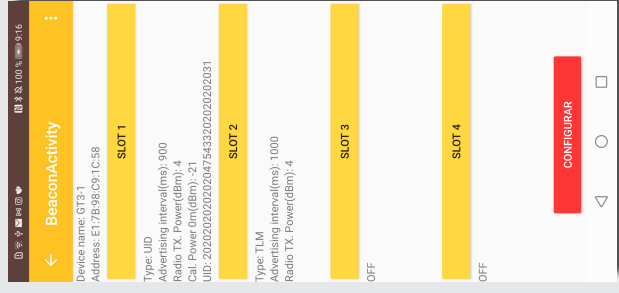
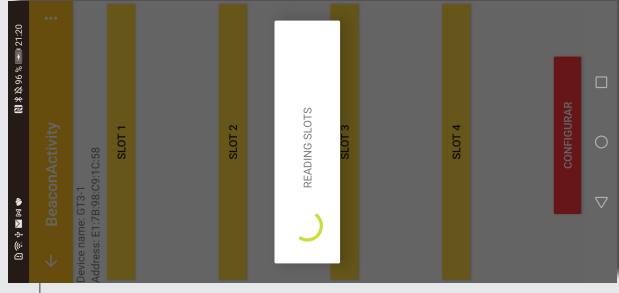


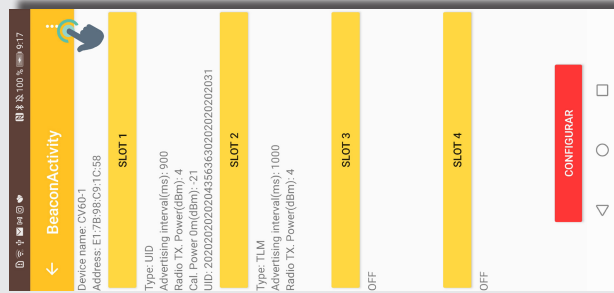
2.2 El led empezará a parpadear durante 30" y en ese tiempo el beacon está en modo "Connectable"

3. BeaconManager App

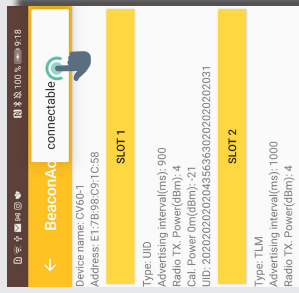


1. Abrir app BeaconManager.
2. Pulsar en la baliza a conectar, conectado!
3. Esperar a que los slots sean leídos

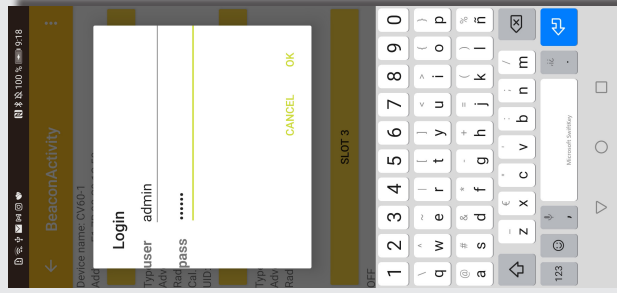




4. Pulsar una vez en los tres puntos



5. Pulsar en connectable



6. Autenticar
user= admin
pass= qwerty



7. Si los datos que se han introducido son correctos, saldrá un mensaje en la parte inferior de la pantalla: "Connectable mode -> ON"



Anexo B

Interfaz final

B.1. Listar dispositivos BLE

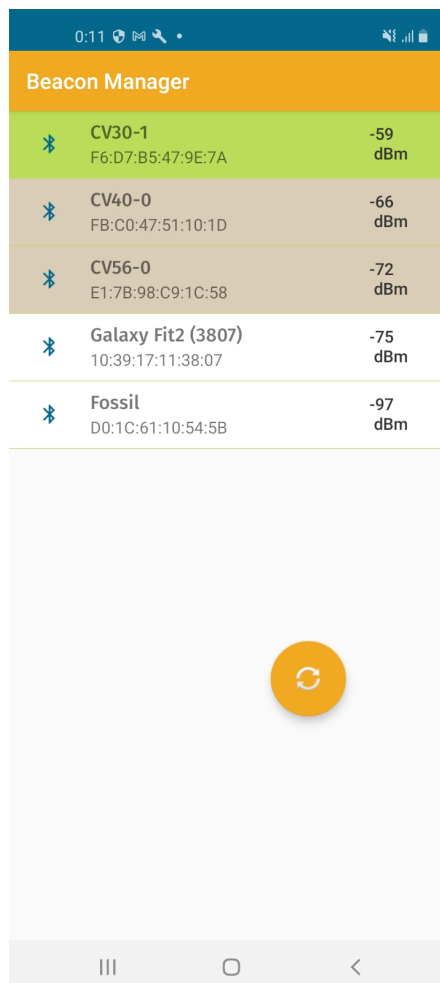


Figura B.1: Listar dispositivos BLE v.2

B.2. Ver información detallada de la baliza

Después de seleccionar una baliza de la lista anterior Figura B.1

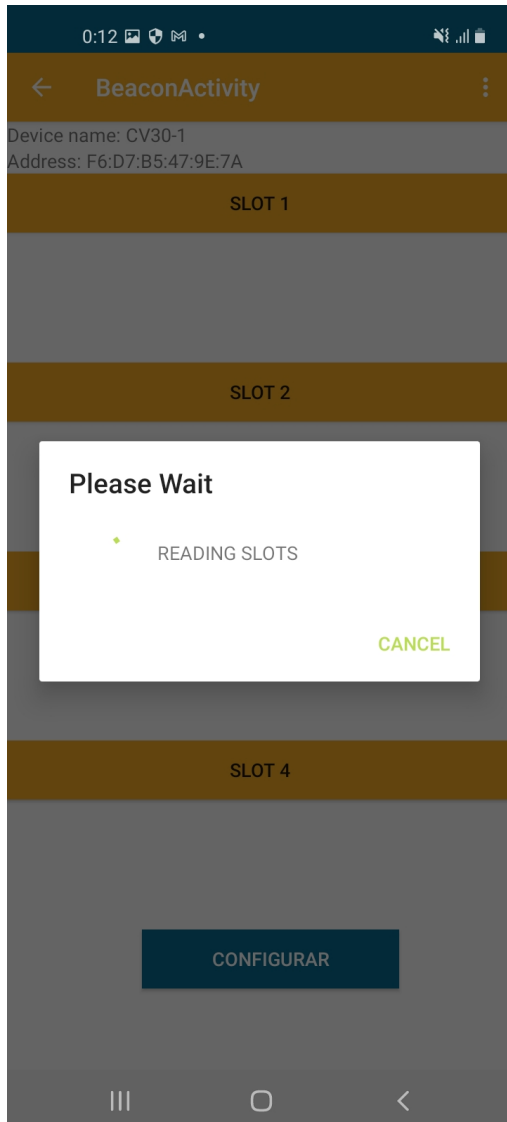


Figura B.2: Leer información de la baliza

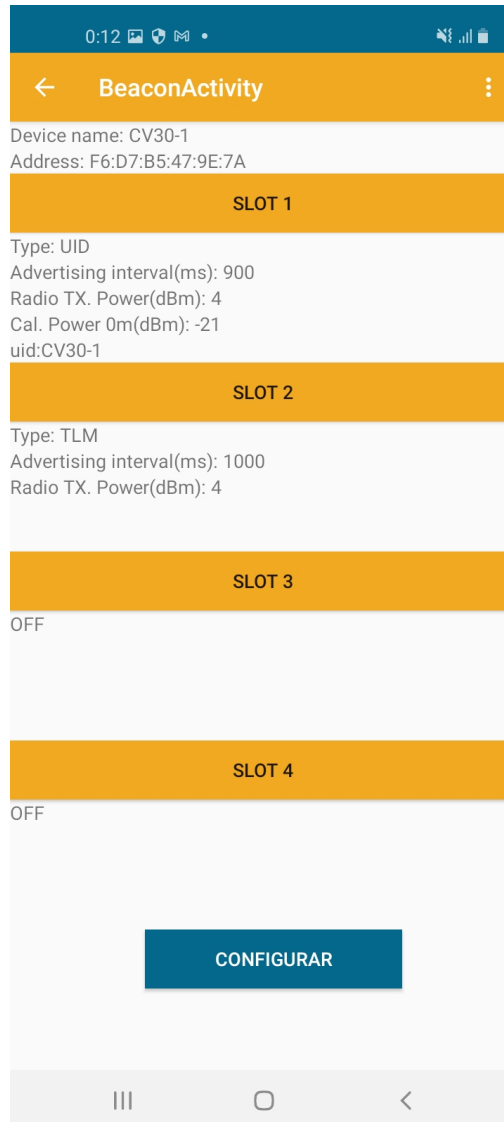


Figura B.3: Mostrar la información leída de la baliza

B.3. Configurar baliza

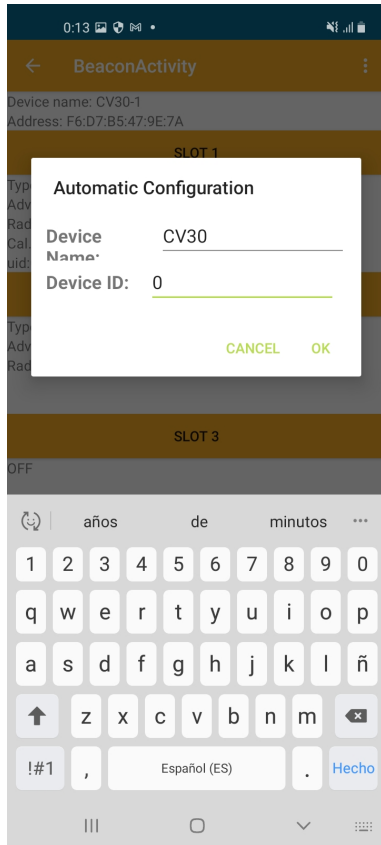


Figura B.4: Modificar nombre de la baliza v.2

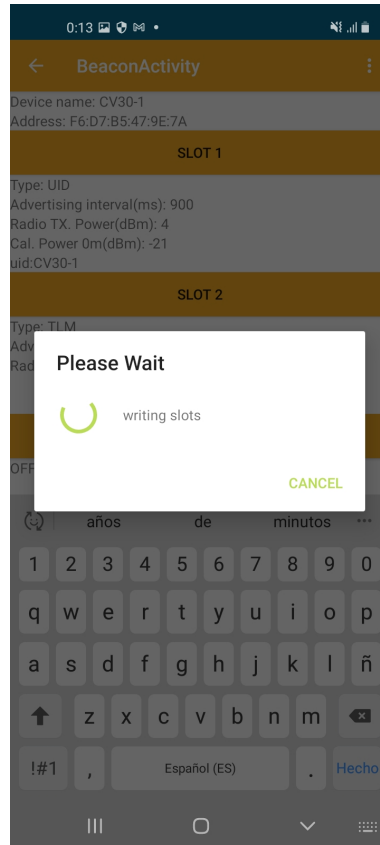


Figura B.5: Escribiendo en los slots correspondientes v.2

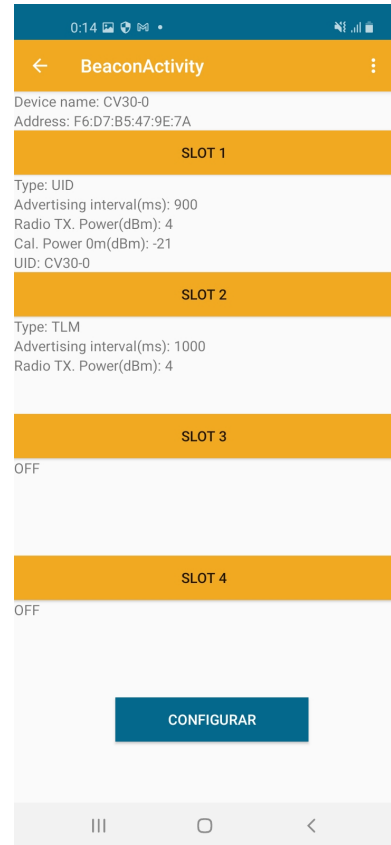


Figura B.6: Modificación efectuada v.2

B.4. Modo conectable

Después de configurar una baliza, esta queda en modo no conectable. Si se quiere que una baliza vuelva a estar en modo conectable, se apaga y se enciende dicha baliza, a partir de ese momento se tiene 30" para volver a conectar a dicha baliza. Una vez conectado se pulsas en los tres puntos, esto está situado en el menú, en el lado derecho. En las siguientes imágenes se puede observar todo el proceso.

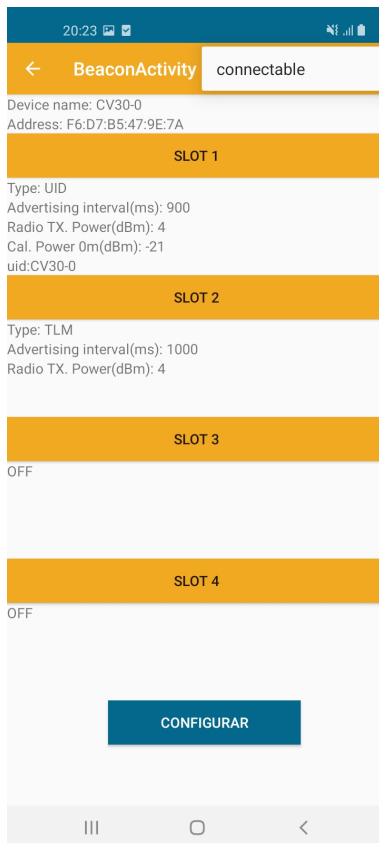


Figura B.7: Pulsar en conectable

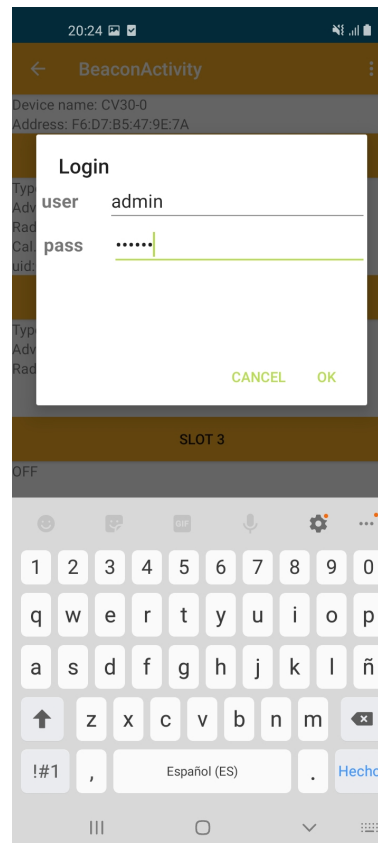


Figura B.8: Escribir las credenciales del administrador

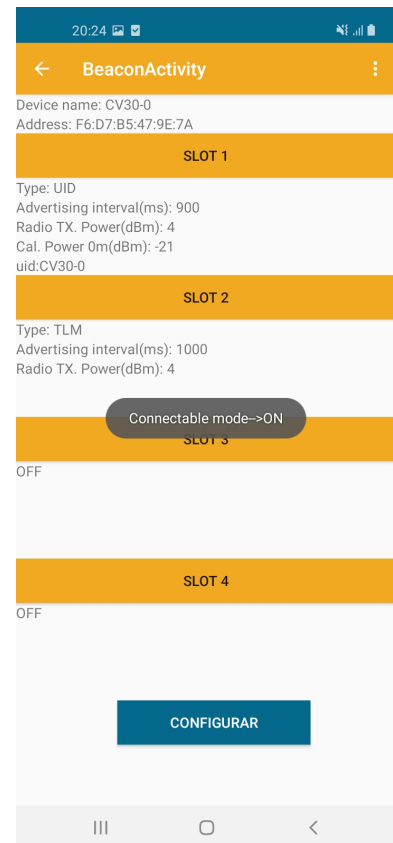


Figura B.9: Modo conectable activado