



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

Universitat Politècnica de València

Trabajo Fin de Grado

Ingeniería en Electrónica Industrial y Automática

# **Diseño y desarrollo de un sistema de monitorización y limitación en tiempo real del aforo en espacios cerrados**

## **Documentos:**

- 1 Memoria
- 2 Planos
- 3 Pliego de condiciones
- 4 Presupuesto
- 5 Anexo I

## **Autor:**

D. Ferran Ballester Mainar

## **Tutor:**

D. Ángel Perles Ivars

Curso académico: **2019-2020**

**Alboraia, noviembre de 2020**



*A mi familia y amigos que me quieren, escuchan,  
aconsejan y apoyan en todo lo que pueden.*

*A mi hermana Alba fuente de inspiración y  
apoyo.*

*A mis grandes amigos Jorge y Josep por  
ayudarme con todo lo que les he pedido y más.*

*Y a Alicia por este año tan raro pero tan bonito.*



## Resumen

---

La situación sanitaria en la que nos encontramos ha supuesto un reto para muchos sectores de nuestra sociedad. Uno de los más afectados ha resultado ser nuestra forma de relacionarnos, tan cercana y afectiva. Ahora es momento de cuidar bien quién se nos acerca a una distancia demasiado corta.

El presente proyecto nace de la idea de despreocuparnos de ese miedo, sobretodo en espacios cerrados donde se encuentre mucha gente de diferentes círculos interpersonales, como puede ser una fábrica, un museo, una gran catedral, etc.

A lo largo de estos meses de pandemia, han ido apareciendo en el mercado soluciones a esta problemática, pero la gran mayoría de ellas se basan únicamente en recopilar información de la ubicación donde ha existido un contacto cercano y enviarla a unos servidores.

Lo que se propone en este proyecto es hacer uso de la tecnología IoT para que el usuario final lleve un dispositivo capaz de detectar que se esté respetando la distancia de seguridad y en ese caso indique al usuario lo que está ocurriendo, haciendo partícipe activo al usuario final y no solo un mero productor de datos y estadísticas.

**Palabras clave:** IoT, distancia social, Beacon, Bluetooth Low Energy, LoRaWAN, Pycom, Python, Tag.



## Resum

---

La situació sanitària en la qual ens trobem ha suposat un repte per a molts sectors de la nostra societat. Un dels més afectats ha resultat ser la nostra manera de relacionar-nos, tan pròxima i afectiva. Ara és moment de cuidar bé qui se'ns acosta a una distància massa curta.

El present projecte naix de la idea de despreocupar-nos d'ixe por, sobretot en espais tancats on es trobe molta gent de diferents cercles interpersonals, com pot ser una fàbrica, un museu, una gran catedral, etc.

Al llarg d'aquests mesos de pandèmia, han anat apareixent en el mercat solucions a aquesta problemàtica, però la gran majoria d'elles es basen únicament a recopilar informació de la ubicació on ha existit un contacte pròxim i enviar-la a uns servidors.

El que es proposa en aquest projecte és fer ús de la tecnologia IoT perquè l'usuari final porte un dispositiu capaç de detectar que s'estiga respectant la distància de seguretat i en ix cas indique a l'usuari el que està ocorrent, fent partícip actiu a l'usuari final i no sols un mer productor de dades i estadístiques.

**Paraules clau:** IoT, distància social, Beacon, Bluetooth Low Energy, LoRaWAN, Pycom, Python, Tag.



## Abstract

---

The health situation in which we find ourselves has been a challenge for many sectors of our society. One of the most affected has turned out to be our way of relating, so close and affectionate. Now is the time to take good care of who approaches us at a too short distance.

This project is born from the idea of not worrying about that fear, especially in closed spaces where many people from different interpersonal circles are gathered, such as a factory, a museum, a great cathedral, etc.

Throughout these pandemic months, solutions to this problem have been appearing on the market, but the vast majority of them are based solely on collecting information from the location where there has been close contact and sending it to servers.

What is proposed in this project is to make use of IoT technology so that the end user carries a device capable of detecting that the safety distance is being respected and in that case it indicates to the user what is happening, making the user an active participant final and not just a mere producer of data and statistics.

**Keywords:** IoT, social distance, Beacon, Bluetooth Low Energy, LoRaWAN, Pycom, Python.





UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

*Diseño y desarrollo de un sistema de  
monitorización y limitación en tiempo real del  
aforo en espacios cerrados*

# 1. Memoria

**Autor:**

D. Ferran Ballester Mainar

**Tutor:**

D. Ángel Perles Ivars

Alboraia, noviembre de 2020

# Índice

---

<b>1. Objeto del proyecto .....</b>	<b>4</b>
<b>2. Antecedentes .....</b>	<b>5</b>
<b>3. Factores a considerar .....</b>	<b>6</b>
3.1. Limitaciones .....	6
3.1.1. Limitaciones externas .....	6
3.1.2. Limitaciones internas .....	7
3.2. Condicionantes .....	7
3.2.1. Facilidad de instalación .....	7
3.2.2. Facilidad de uso .....	7
3.2.3. Mantenimiento .....	8
3.3. Especificaciones iniciales .....	8
<b>4. Planteamiento de soluciones alternativas y justificación de la solución adoptada .....</b>	<b>9</b>
4.1. Sistema de posicionamiento .....	9
4.1.1. Algoritmo de geolocalización .....	9
4.1.2. Protocolo de comunicación .....	12
4.1.2.1. LoRa .....	12
4.1.2.2. Bluetooth Low Energy .....	13
4.2. Señalización .....	14
4.3. Tecnología inalámbrica .....	15
4.3.1. WiFi .....	15
4.3.2. SigFox .....	16
4.3.3. LoRa .....	16
4.3.4. Bluetooth Low Energy .....	17
4.4. Hardware .....	17
4.4.1. Arduino MKR WAN 1310 .....	17
4.4.2. ESP32 BLE on Arduino IDE .....	18
4.4.3. LoPy4 .....	18
<b>5. Descripción detallada de la solución adoptada.....</b>	<b>19</b>
5.1. Arquitectura del sistema .....	19
5.2. Elección y diseño del hardware .....	21

*Diseño y desarrollo de un sistema de monitorización y limitación en tiempo real del aforo en espacios cerrados*

5.2.1.	Beacons .....	21
5.2.2.	Tags .....	26
5.2.2.1.	Batería .....	27
5.2.2.2.	Comunicación LoRa.....	27
5.2.2.3.	Sistemas de señalización .....	28
5.3.	Diseño de los Tags .....	32
5.4.	Diseño del software.....	34
5.4.1.	Posicionamiento del Tag.....	37
5.4.2.	Comunicación BLE.....	40
5.4.3.	Comunicación LoRa.....	41
5.4.4.	Configuración de salidas GPIO.....	43
<b>6.</b>	<b>Pruebas realizadas.....</b>	<b>44</b>
6.1.	Montaje del prototipo .....	44
6.2.	Test de funcionamiento .....	46
6.2.1.	Prueba 1.....	47
6.2.2.	Prueba 2.....	47
<b>7.</b>	<b>Conclusiones .....</b>	<b>48</b>
7.1.	Sobre el trabajo realizado.....	48
7.2.	Mejoras.....	48
<b>8.</b>	<b>Bibliografía .....</b>	<b>50</b>

## GLOSARIO DE SIGLAS Y ABREVIATURAS

IoT	Internet de las cosas
Tag	Dispositivo portátil que portan los usuarios
RSSI	Received Signal Strength Indicator
LoRa	Protocolo de comunicación inalámbrica diseñado para largas distancias
LoRaWAN	Long Range Wide Area Network
BLE	Bluetooth Low Energy
Beacon	Dispositivo emisor de señales BLE
LED	Light Diode Emisor
LCD	Liquid Crystal Display
Buzzer	Zumbador electrónico
LPWAN	Low Power Wide Area Network
CPU	Central Processing Unit
GPIO	General Purpose Input/Output
PCB	Printed Circuit Board
CAD	Computer assisted design
RGB	Siglas en inglés de Rojo, Verde y Azul
ABP	Authentication By Personalisation

## **1. Objeto del proyecto**

---

El objeto del presente proyecto es desarrollar un prototipo de un sistema de monitorización y control del aforo en espacios cerrados que sean propensos a masificarse, como puede ser un museo o cualquier lugar turístico cerrado, también siendo extrapolable al sector de la industria.

El sistema ofrecerá el seguimiento del aforo en tiempo real, así como el control y limitación de éste interactuando directamente con los usuarios asistentes, con el objetivo de que sean conscientes del estado actual del aforo y puedan decidir cambiar su ubicación o ruta en base a esto.

Con esta finalidad, se pretende desarrollar un dispositivo portátil (Tag) que portarán los asistentes, que se comunicarán entre ellos y con un servidor central. A partir de las señales recibidas al servidor central se creará un mapa de calor que clarificará en qué determinados espacios se sobrepasa el aforo recomendado. Cuando dos dispositivos se encuentren demasiado cerca, se accionará una respuesta en estos mismos, que los asistentes interpretarán según las instrucciones recibidas por el personal trabajador del recinto a la entrada.

## 2. Antecedentes

---

La situación sanitaria actual nos ha llevado a extremar las medidas de seguridad y distanciamiento social. Esto ha hecho que algunos sectores industriales y sobretodo de servicios, como es el sector turístico y cultural, se vean gravemente afectados ante esta nueva situación. Aquí nace la necesidad de este proyecto, buscar una solución al sector cultural y turístico que nos ayude a reactivar la economía de estos sectores, asegurándose del cumplimiento de esta nueva conducta social como es el distanciamiento.

Actualmente existen distintas opiniones en cuanto a la distancia ideal en la que dos personas (equipadas con medidas de seguridad higiénica, es decir, mascarilla) no se encontrarán en una situación de riesgo frente al contagio del SARS-CoV-2, pero esta distancia en ningún caso ha de ser inferior a 1.5 metros, por lo que se usará esta medida referencia como distancia de seguridad.

Se trata de una situación excepcional, en la que tenemos que responder de una forma rápida y eficaz. Los intentos previos hasta el momento han sido limitar los espacios y zonas por las que los asistentes y los trabajadores pueden circular libremente, aparte, obviamente, de las medidas higiénicas pertinentes. También se ha usado la tecnología para adquirir datos del estado real del aforo, pero en ningún caso se ha hecho un uso activo de la tecnología para que el aforo se autocontrole. Lo cual aumenta los recursos tanto materiales como humanos que se usan con este fin.

Actualmente existen algunos sistemas similares, como pueden ser los beacons o algún otro dispositivo localizable, pero su uso más extendido es únicamente la localización. Lo que se propone en este proyecto es utilizar esa tecnología y reinventarla para que se le añada una nueva funcionalidad como es el autocontrol del posicionamiento de personas en espacios masificados.

### **3. Factores a considerar**

---

El primer paso en la realización del proyecto es realizar un estudio de las necesidades que pretendemos cubrir y de los factores que van a limitarnos y a condicionar el desarrollo de este proyecto.

#### **3.1. Limitaciones**

Se diferencia entre limitaciones externas, que son ajenas a nuestro trabajo, y limitaciones internas.

##### **3.1.1. Limitaciones externas**

Este tipo de limitaciones dependen de los factores externos según el hardware a utilizar en el prototipo.

Las 2 principales limitaciones son el consumo energético y el precio total del hardware. Éstas van a afectar de forma indirecta a las demás limitaciones externas que se exponen a continuación.

- **Limitaciones en el sistema de posicionamiento**

Se debe escoger un sistema de posicionamiento del que se obtenga una buena precisión, que no llegue a ser muy pobre como para tener un margen de error demasiado grande. Tampoco se puede obviar que un sistema de posicionamiento demasiado preciso nos haría utilizar una capacidad de procesamiento muy elevada, lo cual se pretende evitar en lo posible.

- **Limitaciones en la comunicación con el servidor**

La comunicación con el servidor tiene que permitir que un número elevado de dispositivos envíen paquetes de información prácticamente al mismo tiempo, y que ello no implique una saturación del servidor. También se debe tener en cuenta que los paquetes de información que se envíen no sean de un tamaño muy elevado, puesto que afectaría al consumo de energía. Tiene que ser un protocolo de comunicación que consuma la mínima cantidad de energía

---

posible, puesto que los dispositivos son portátiles y tienen una batería limitada.

- Limitaciones en la señalización

En relación con la señalización al usuario portador del Tag en el caso de que no se cumplan correctamente las condiciones de aforo, tiene que ser una señalización clara y directa, que aunque éste se encuentre distraído sea capaz de detectarla a pesar de la situación del entorno (mucho ruido o visibilidad nula) y poder actuar en consecuencia.

### **3.1.2. Limitaciones internas**

La principal limitación interna a la que se hace frente es la capacidad de integrar todas las tecnologías necesarias para desarrollar el prototipo en un mismo sistema embebido, que sea capaz de usar diferentes protocolos de comunicación al mismo tiempo y con un consumo energético bajo.

## **3.2. Condicionantes**

Los condicionantes existentes en el proyecto son los siguientes.

### **3.2.1. Facilidad de instalación**

Todo el sistema debe ser lo más sencillo de instalar posible, debido a que no se pretende realizar unas obras de gran magnitud en el espacio en el que va a ser instalado.

La instalación se debe poder hacer en un periodo de tiempo máximo de 1 día para que la empresa cliente no vea afectada en exceso su producción o la calidad de sus servicios.

### **3.2.2. Facilidad de uso**

Se trata de un sistema que se va a tener que acoplar a las diferentes necesidades de todos los usuarios. Se ha de tener en cuenta que no todas las personas tenemos las mismas capacidades tanto sensoriales como motrices, por lo que la señalización tendrá que realizarse mediante más de una vía sensorial al ser humano,

para que aquellas personas con diversidad funcional no sean excluidas dentro del sistema.

### **3.2.3. Mantenimiento**

El mantenimiento que este sistema necesita ha de ser mínimo. Los dispositivos portátiles tienen que ser lo más robustos y duraderos posibles para así evitar roturas o un funcionamiento no adecuado. En cuanto a la comunicación con el servidor tiene que ser completamente autónoma, sin que ningún operario tenga que intervenir activamente en ésta.

## **3.3. Especificaciones iniciales**

Ante estas limitaciones y estos condicionantes se procede a realizar una lista de especificaciones que el prototipo tendrá que abarcar y cubrir de una forma efectiva.

- Sistema de posicionamiento con una buena precisión de menos de 2,5 metros de margen de error.
- Cálculo de posición sencillo, mínima capacidad de procesamiento a usar.
- Conexión Tag-Servidor simple, no es necesaria una gran capacidad de envío de datos.
- Señalización clara, sin espacio a confusiones.
- Señalización sonora.
- Señalización mediante vibración.
- Sistema embebido capaz de integrar diferentes tipos de protocolos de comunicación.
- Consumo mínimo de energía, tanto por el sistema embebido como por los periféricos que pueda llevar integrados.
- Duración de la batería mínima de 12 meses.
- Sencillez en el diseño: peso ligero
- Tamaño reducido del dispositivo portátil.

## 4. Planteamiento de soluciones alternativas y justificación de la solución adoptada

---

Para el planteamiento de las distintas alternativas se diferencian cuatro subapartados a estudiar: el sistema de posicionamiento, la señalización de las alertas por incumplimiento de la distancia de seguridad, la conexión y el tipo de red al que se enviarán los datos y el tipo de hardware a utilizar en el prototipo.

### 4.1. Sistema de posicionamiento

En el sistema de posicionamiento se debe priorizar un sistema que tenga una buena precisión, puesto que es primordial que los Tags no superen los límites definidos por la distancia de seguridad, sin que ello conlleve un gasto energético demasiado grande por la necesidad de procesamiento al calcular la posición.

En este apartado se estudiarán dos materias: el algoritmo mediante el cual se calculará la posición y el protocolo de comunicación usado para calcular los parámetros necesarios que usará el algoritmo.

#### 4.1.1. Algoritmo de geolocalización

Al hablar de algoritmos de geolocalización, antes se debe especificar qué método de localización se seguirá para obtener un resultado fiable y seguro. De estos métodos no siempre se obtiene el parámetro de la ubicación directamente, es entonces donde entra en juego el algoritmo a utilizar.

En el artículo *Algoritmos y métodos para la localización en interiores* [1] se diferencian los siguientes métodos de localización:

- Identificación por celda:

Este método, utilizado por redes WLAN y WPAN, se obtiene la posición directamente en función de la identidad del dispositivo conectado a esta red. Se trata del método más sencillo, puesto que no necesita un algoritmo posterior para calcular la localización, pero es un método con una precisión demasiado pobre para el proyecto.

- Ángulo de llegada:

En este método se juega con el ángulo de recepción de las señales a una antena receptora. Dependiendo de este ángulo, el algoritmo posteriormente diseñado debe ser capaz de determinar la ubicación exacta del dispositivo emisor. Éste método funciona bien en entornos rurales, donde las interferencias físicas a las señales son nulas o muy pequeñas.

- Potencia de la señal recibida:

Este método se basa en la pérdida energética que sufre una señal cuando se desplaza a través de un medio, en el caso del proyecto el aire. Podemos medir esto como el RSSI – Recieved Signal Strength Indicator. A partir de este valor es posible calcular la distancia desde la que ha sido enviada la señal. Para que esta técnica funcione se necesitarían un mínimo de 3 dispositivos receptores o emisores, dependiendo de la arquitectura de la red. La precisión obtenida con este método suele ser bastante buena. Pero a medida que la señal pierda energía la precisión será mucho menor y aumentará el margen de error.

- Métodos basados en tiempo:

Se trata de un método parecido al anterior, con la diferencia que la magnitud medida es el tiempo que tarda la señal en llegar a los diferentes receptores. De nuevo se necesitarían un mínimo de 3 receptores o emisores para que este método funcione.

Una vez analizados los diferentes métodos vemos que la mejor alternativa para el proyecto resulta el **método de la potencia de la señal recibida**, puesto que la magnitud con la que se trabaja en este método es de fácil acceso a través de un sistema embebido capaz de recibir señales y cuantificar la potencia de estas.

A partir del método de localización escogido para el proyecto se procede a comparar los diferentes algoritmos de geolocalización que serían útiles. Éstos podemos encontrarlos en el artículo mencionado anteriormente [1]:

- Algoritmos matemáticos:

Se trata de algoritmos que emplean el cálculo para determinar la ubicación de un dispositivo. Se emplea fundamentalmente la trigonometría. En nuestro caso deberíamos contar con un mínimo de 3 dispositivos emisores o receptores para poder realizar la técnica de la triangulación circular. Ésta consiste en la intersección de circunferencias de radio igual a la distancia única a cada dispositivo emisor/receptor. El punto de intersección de estas circunferencias corresponde a la posición en un plano 2D del dispositivo a ubicar. Este algoritmo también se puede usar empleando otras figuras geométricas, pero el cálculo se complica demasiado comparado con el aumento de precisión que obtenemos.

- Algoritmos probabilísticos:

Estos algoritmos se basan en determinar la probabilidad de que un dispositivo se encuentre en una localización de acuerdo con una historia de estimaciones o posiciones previamente conocidas. Esto significa que se deberá hacer uso de una memoria en la que se vayan almacenando las ubicaciones ya calculadas, lo que supone un gran inconveniente para el proyecto, puesto que además de la necesidad de tener en cuenta esto en la fase de diseño del hardware, el tiempo de cómputo y la capacidad de procesamiento del sistema tendrá que ser mucho mayor.

- Algoritmos inteligentes:

Estos algoritmos hacen uso de técnicas de inteligencia artificial y sus principales características como el aprendizaje, el entrenamiento, la solución basada en casos y el manejo de incertidumbre para resolver estos problemas surgidos del cálculo de la localización. Se pueden descartar estos algoritmos, puesto que nuestro proyecto no pretende estar basado en estas técnicas de inteligencia artificial.

Teniendo en cuenta todos los algoritmos anteriormente mencionados, la mejor alternativa es el **algoritmo matemático de triangulación circular**, puesto que las distancias medidas a cada localizador se pueden obtenerlas mediante la RSSI de las

señales y la capacidad de cálculo que ha de ser usada es muy aceptable teniendo en cuenta que el prototipo tiene que ser lo más eficiente posible energéticamente hablando.

#### **4.1.2. Protocolo de comunicación**

A partir de las soluciones adoptadas en el apartado anterior se procede a elegir un protocolo de comunicación que dé acceso a la variable RSSI de las señales transmitidas. En vista de esto, se plantea el uso de LoRa o de Bluetooth Low Energy.

##### **4.1.2.1. LoRa**

LoRa [2] es una modulación inalámbrica creada para el uso en aplicaciones long-range low power low-data rate desarrollada por la empresa Semtech.

Las siglas de LoRaWAN vienen de Long Range Wide Area Network. Se refiere a una especificación para redes de baja potencia y rango de área amplio, LPWAN (Low Power Wide Area Network). LoRaWAN es un protocolo optimizado para dispositivos alimentados por baterías y que requieren conexión al IoT.



*Figura 1. Logo de LoRa.*

Esta tecnología permite el envío y recepción de información con un largo alcance, alrededor de 20 km en condiciones favorables y entre 2 y 3 km en condiciones normales.

La LoRa Alliance [3] se encarga del desarrollo del estándar y su evolución. Se trata de un protocolo de red actualizado y libre. Su implementación es muy flexible, por lo que el usuario puede crear de forma sencilla una red privada.

En cuanto a su implementación como sistema de posicionamiento es muy útil en espacios interiores. Se puede ver en el estudio *Indoor Vehicles Geolocation Using LoRaWAN* [4], que su implementación es dependiente del uso de gateways de

LoRaWAN que recibirán las señales enviadas por los Tags. Posteriormente esta información se enviará a un servidor en el que se computará el cálculo necesario para obtener la posición. Estos gateways tienen un precio al consumidor demasiado alto como para poder plantearlo como posible en el proyecto.

En cuanto a la precisión, según el estudio anteriormente nombrado [4], los resultados y las conclusiones hablan de un margen de error de entre 20 y 30 metros, lo que excede los límites planteados para el proyecto.

#### 4.1.2.2. Bluetooth Low Energy

El *Bluetooth Low Energy* (BLE) [5], también llamado *Bluetooth Smart* es una versión ligera del protocolo Bluetooth clásico. Se introdujo con la actualización 4.0. Fue un proyecto que empezó Nokia llamado *Wibree* antes de que fuese adoptado por la compañía Bluetooth SIG.

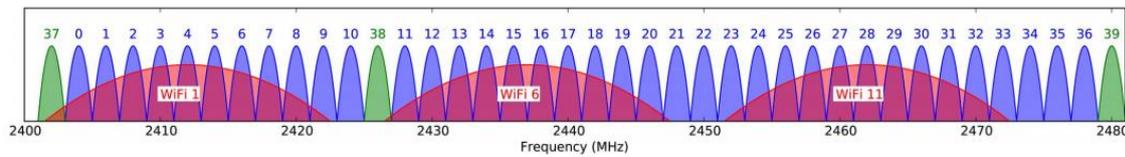


Figura 2. Logo de Bluetooth Smart.

Se trata de un protocolo de comunicación entre dispositivos que, como explica el estudio *Location Fingerprint With Bluetooth Low Energy Beacons* [6], utiliza 40 canales de 2 MHz cada uno en la banda de los 2.4 GHz, también utilizada por el protocolo WiFi. El BLE se basa en señales llamadas “advertisements” que los dispositivos emiten cada cierto tiempo programado anteriormente. Estas señales se emiten en los canales número 37, 38 y 39 como se observa en la figura 3.

---

#### 4. Planteamiento de soluciones alternativas y justificación de la solución adoptada



*Figura 3. Los 40 canales BLE y los 3 canales WiFi más usados. Los “advertisement” solo ocurren en los canales 37, 38 y 39.*

Si se atiende a los resultados y a las conclusiones de este estudio [6], con el uso de Beacons, emisores de señal BLE, se llega a conseguir una precisión muy buena, teniendo un margen de error de aproximadamente 2,6 metros, cuando se usan una cantidad de 1 Beacon por cada 30 metros cuadrados. Además la arquitectura de este protocolo es de bajo coste, puesto que los Beacons son una solución hardware muy barata.

Atendiendo a la buena precisión y al bajo coste del posible sistema, el uso del **BLE** como protocolo de comunicación entre los dispositivos es la mejor alternativa para implantarlo en el proyecto.

## **4.2. Señalización**

En cuanto a la señalización, es necesaria una señalización que no dé pie a confusiones, que sea clara y concisa. Para esto se pueden considerar 3 medios de señalización a los usuarios portadores de los Tags: la visual, la auditiva y la señalización por vibración.

La visual requeriría de una serie de LEDs o algún tipo de pantalla LCD que emitiesen luces llamativas. La auditiva, por su parte, la emisión de una serie pitidos o sonidos estridentes producidos por un altavoz o un Buzzer. Y por último se quiere señalar mediante una vibración, por lo que será necesaria la implementación de un motor vibrador en nuestro prototipo.

Una vez planteado esto, hay que tener en cuenta la finalidad real del prototipo, y es que diferentes personas, con diferentes capacidades y necesidades, sean conscientes de que no se está respetando la distancia de seguridad, por lo que se van a implementar más de una de las soluciones mencionadas anteriormente, para que en el caso de que

una persona no tenga capacidad visual o auditiva, pueda de todas formas notar la vibración. Pero al querer consumir la mínima energía posible, se va a descartar la opción de la señalización visual. En resumen, se implementará un sistema de vibración y un sistema de señalización sonora.

### 4.3. Tecnología inalámbrica

La red de tecnología inalámbrica a usar en el proyecto es de vital importancia, puesto que a partir de ésta se verá muy influenciada la elección del hardware a usar y los diferentes escenarios en los que esta tecnología podrá utilizarse.

A continuación se exponen una serie de alternativas que han sido estudiadas con la finalidad de elegir la que mejor encaje con el proyecto.

#### 4.3.1. WiFi

La tecnología WiFi [7] es un protocolo que permite la comunicación entre varios dispositivos electrónicos, como ordenadores, smartphones, tablets, televisores, videoconsolas, etc. Esta tecnología permite que los dispositivos puedan conectarse entre sí o la red de Internet a través de un punto de acceso.



Figura 4. Logo de la marca WiFi.

El WiFi nace en 1999 ante el problema de establecer una vía de comunicación inalámbrica entre distintos dispositivo y a red de Internet. Las empresas encargadas del desarrollo de esta tecnología fueron 3Com, Nokia, Symbol, Airones, Intersil y Lucent Technologies. De la unión de estas empresas surgió la *Wireless Ethernet Compatibility Alliance*, que en abril de 2000 certifica la interoperabilidad de equipos según la norma IEEE 802.11b, bajo la marca WiFi.

La tecnología WiFi clásica trabaja en una frecuencia de 2.4 GHz, aunque también existe una versión más actual que trabaja en la banda de los 5 GHz. Ésta

---

## 4. Planteamiento de soluciones alternativas y justificación de la solución adoptada

permite una mayor transmisión de datos y un mayor ancho de banda en detrimento de la distancia a la que puede operar.

Esta tecnología tiene un alcance estimado de entre 100 y 200 metros, y permite una cantidad de transmisión de datos muy elevada, lo que hace que su consumo energético sea muy elevado. En cuanto a la distancia de transmisión es de un valor aceptable, pero el proyecto es muy dependiente del consumo energético, lo que hace que se rechace esta alternativa.

### **4.3.2. SigFox**

SigFox [8] es una tecnología de la familia de las LPWAN. Se trata de un tipo de tecnología muy similar a LoRa. Esta tecnología trabaja en el mismo espectro frecuencial de las ondas electromagnéticas para enviar los paquetes de datos, 868 MHz en Europa y 902 MHz para EE.UU. También se basa en una filosofía de mínimo consumo energético. El nivel máximo de potencia de sus señales en la zona Europea no ha de superar los 14 dB.

El espectro frecuencial en el que trabaja está dividido en 400 canales de 100 Hz cada uno. Capaz de enviar unos 140 paquetes de mensajes al día, siendo su consumo muy bajo.

Hasta aquí se trata de una tecnología extremadamente similar a LoRa, el problema con esta tecnología reside en que no es de uso gratuito, hay que darse de alta en sus servidores y realizar el pago de su suscripción anualmente. Esta es la razón por la que esta alternativa va a ser rechazada, se pretende que el sistema funcione sin tener que renovar ninguna licencia y con el mínimo mantenimiento posible.

### **4.3.3. LoRa**

Como se ha explicado en el apartado 4.1.2.1. LoRa es una modulación inalámbrica creada para el uso en aplicaciones long-range low power low-data rate. Esta tecnología ha sido rechazada para su uso como sistema de posicionamiento, pero teniendo en cuenta sus características de transmisión de datos como el alcance típico de entre 2 a 3 kilómetros, posibilidad de modificar la latencia entre envío de paquetes y

cantidad de información enviada muy reducida lo hace ideal para su uso en el proyecto como protocolo de comunicación entre los Tags y un servidor central.

#### **4.3.4. Bluetooth Low Energy**

El BLE, ya explicado en el apartado 4.1.2.2, se trata de una versión más ligera del Bluetooth clásico. Este protocolo también trabaja en el rango de frecuencia de los 2.4 GHz, pero en 40 canales de 2 MHz cada uno, utilizando los canales 37, 38 y 39 para el envío de señales “advertising”.

Se ha de tener en cuenta que el uso de esta tecnología ha sido escogido para el sistema de posicionamiento, por lo que se podrá aprovechar su funcionalidad para el caso de detectar otro Tag emisor de señales “advertising” que infrinja la norma de distanciamiento de seguridad.

Como conclusión, se hará uso tanto de la tecnología LoRa para la conexión Tag-Servidor, como de la tecnología BLE para el caso de conexión Tag-Tag que activará las señales pertinentes descritas en el apartado 4.2.

### **4.4. Hardware**

El hardware elegido será clave a la hora de determinar el precio final del prototipo. Según lo estudiado y decidido en los apartados anteriores se muestran 2 opciones, diseñar el PCB del sistema embebido de cero, buscando todos los componentes necesarios o utilizar un microcontrolador ya a la venta en el mercado, al que se le añadirá el hardware necesario.

Debido al coste en términos de tiempo, se ha decidido optar por la segunda alternativa. A continuación se describen los posibles microcontroladores a elegir.

#### **4.4.1. Arduino MKR WAN 1310**

El módulo Arduino MKR WAN 1310 [9] se trata de un sistema embebido con la CPU ATSAMD21, capaz de proveer conectividad a la red LoRa usando el servidor Arduino IoT Cloud. También es capaz de almacenar una cantidad de información gracias a su memoria Flash de 2 MBytes.

---

## **4. Planteamiento de soluciones alternativas y justificación de la solución adoptada**

Esta alternativa nos ofrece un lenguaje de programación sencillo, al mismo tiempo que compatibilidad con una gran cantidad de módulos de Arduino. Pero el inconveniente principal es que se necesita otro módulo con capacidad para comunicarse a través del protocolo BLE, por lo que esta alternativa aumentaría demasiado el precio del prototipo.

#### **4.4.2. ESP32 BLE on Arduino IDE**

Este módulo se trata de un sistema embebido capaz de comunicarse utilizando tanto el protocolo BLE como el WiFi. Con éste módulo surge el mismo problema que con el módulo del apartado anterior, se necesita añadir un módulo capaz de usar la tecnología LoRa, lo que añade más complejidad a la hora de programar, puesto que se contaría con más de un micro controlador.

#### **4.4.3. LoPy4**

Por último, el módulo LoPy4 [11] consiste en un sistema embebido que se caracteriza por 2 cosas. La primera es el lenguaje de programación que se utiliza, Python. Concretamente una versión optimizada para microcontroladores llamada MicroPython. La segunda es que el módulo incluye protocolos de comunicación de 4 tecnologías diferentes: LoRa, Sigfox, WiFi y BLE.

Debido a sus características parece el módulo ideal a utilizar en el proyecto, puesto que no se necesitará ningún módulo adicional y esto hará que el precio final no se vea demasiado incrementado debido a la elección del hardware.

## 5. Descripción detallada de la solución adoptada

---

En este apartado se procede a describir de forma más precisa cada una de las decisiones que se han ido tomando a la hora de desarrollar el prototipo. Hay que tener siempre en mente que no se trata de un producto con una finalidad de introducción en el mercado, por lo que en el caso de que el proyecto avanzase y se decidiera sacar al mercado, habría que modificar algunas de las decisiones tomadas para que se ajusten a las certificaciones vigentes del mercado europeo.

La solución adoptada la podemos dividir en cuatro elementos a diseñar.

### 5.1. Arquitectura del sistema

La arquitectura a diseñar va a constar de 2 tipos de tecnologías, LoRaWAN y BLE. Respecto a la primera, en la figura 5 se observa el esquema típico de una red LoRaWAN, en el que cada dispositivo LoRa envía datos al Gateway más cercano y éste es el encargado de enviar esta información al servidor a través de conexión 3G o Ethernet. A continuación, el servidor es el encargado de procesar los datos en las distintas aplicaciones creadas según las necesidades de cada proyecto.

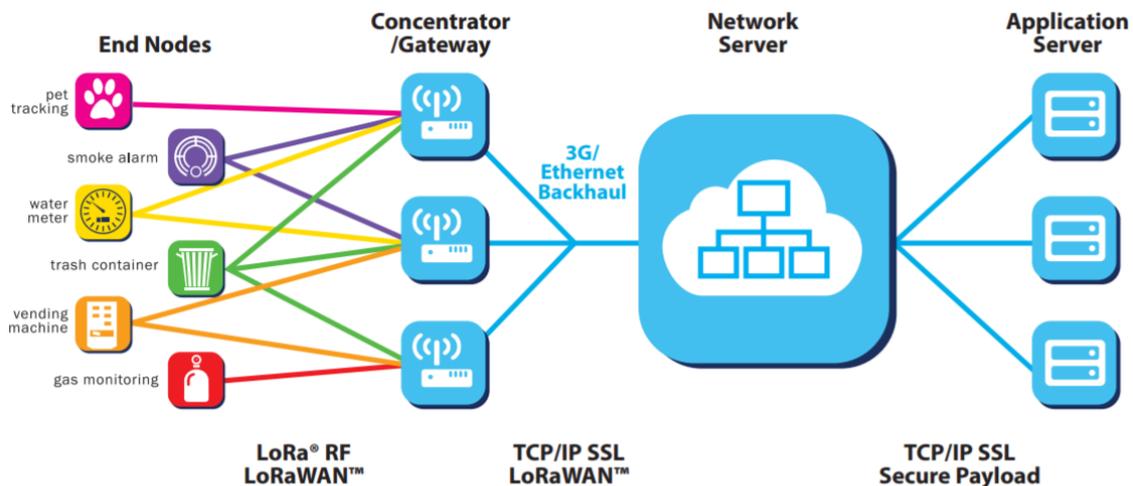


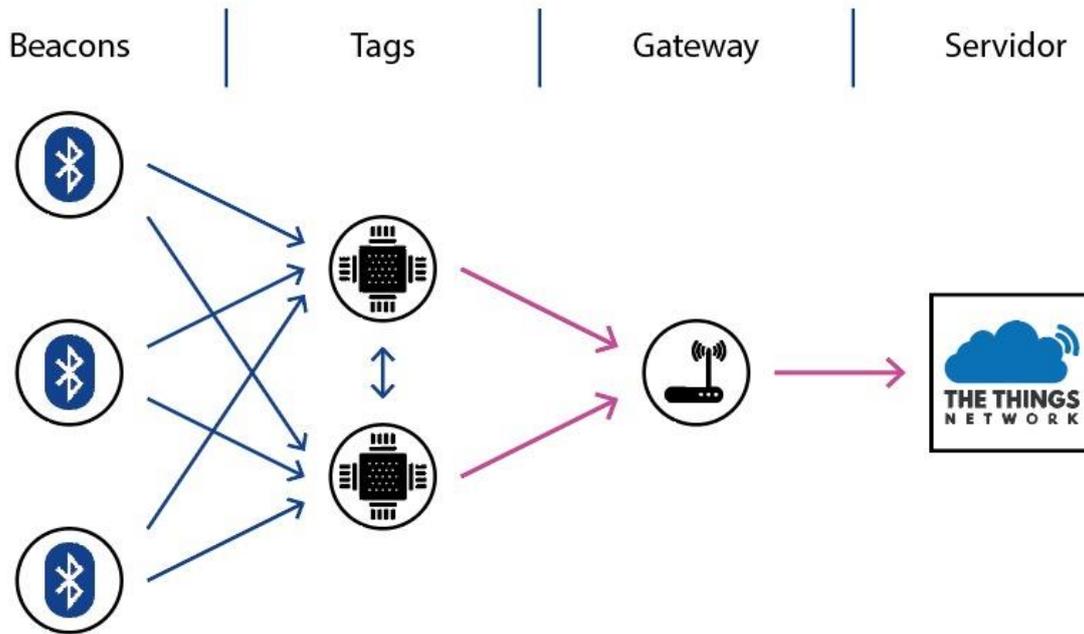
Figura 5. Arquitectura de un sistema de comunicación LoRaWAN

---

## 5. Descripción detallada de la solución adoptada

Por otra parte, el esquema de comunicación BLE no está tan definido. Esta comunicación puede ser entre una gran variedad de dispositivos sin que estos tengan que estar conectados a algún servidor central.

Una vez analizadas las dos tecnologías de comunicación la arquitectura propuesta para el prototipo se puede ver en el esquema de la figura 6.



*Figura 6. Esquema de la arquitectura de conexión del prototipo, en azul las conexiones mediante BLE y en rosa las conexiones mediante LoRaWAN.*

El dispositivo principal, los Tags, están comunicándose por 2 vías al mismo tiempo. La primera es con los Beacons BLE, los cuales enviarán las señales “advertising” por los canales correspondientes, como ha sido explicado en el apartado 4.1.2.2., al mismo tiempo los Tags van a estar comunicándose entre ellos mediante BLE para comprobar que las señales recibidas no superan el límite de potencia, lo que significaría que no se está respetando la distancia de seguridad.

Por otra parte, en el momento que esto ocurra, los Tags enviarán las señales correspondientes mediante una conexión LoRaWAN al Gateway más cercano. A partir

de aquí el Gateway será el encargado de conectar con el servidor y enviarle la información.

En este prototipo vamos a obviar esa última parte, puesto que será el potencial cliente el que decidirá qué servidor en la nube quiere utilizar y mediante qué Gateway lo hará. En este caso, en el apartado 5.4. se explicará cómo se debería realizar esa conexión y la solución que ha sido adoptada.

## **5.2. Elección y diseño del hardware**

A partir de la arquitectura propuesta en el apartado anterior, a continuación se va a diseñar todo el hardware necesario para poner en funcionamiento el prototipo. Por una parte debemos seleccionar los beacons y por otra todo el hardware que se incluirá en los Tags.

### **5.2.1. Beacons**

Los beacons o balizas electrónicas [12] son dispositivos transmisores de señales BLE a otros dispositivos que se encuentren dentro de un rango de actuación. Estas balizas tienen un identificador único universal, lo que usará el prototipo para reconocer que la señal BLE que ha recibido viene de uno de los beacons y no de otro Tag.

Los beacons a utilizar se requiere que sean lo más duraderos posible, y de fácil instalación y puesta en marcha. Es por eso que se usarán los beacons de la marca Accent Systems [13], en concreto el model iBKS 105 (figura 7). Se trata de un beacon de un tamaño muy reducido y de un diseño muy discreto, perfecto para el uso que se le pretende dar. Éstos funcionan con una batería de botón tipo CR2477 y su duración se estima entre 30 y 46 meses.

La empresa Accent Systems pone a disposición una aplicación móvil, tanto para Android como IOS, llamada iBKS Config Tool. Ésta permite modificar los parámetros característicos de los beacons. En las figuras 8, 9, 10, 11 y 12 se muestran los pasos, marcados por los rectángulos rojos, para configurar la ventana de tiempo en la que el beacon emitirá la señal “advertising”.



Figura 7. Beacon iBKS 105 de Accent Systems.

Paso 1: Se escanean las señales Bluetooth y elegimos el beacon deseado.

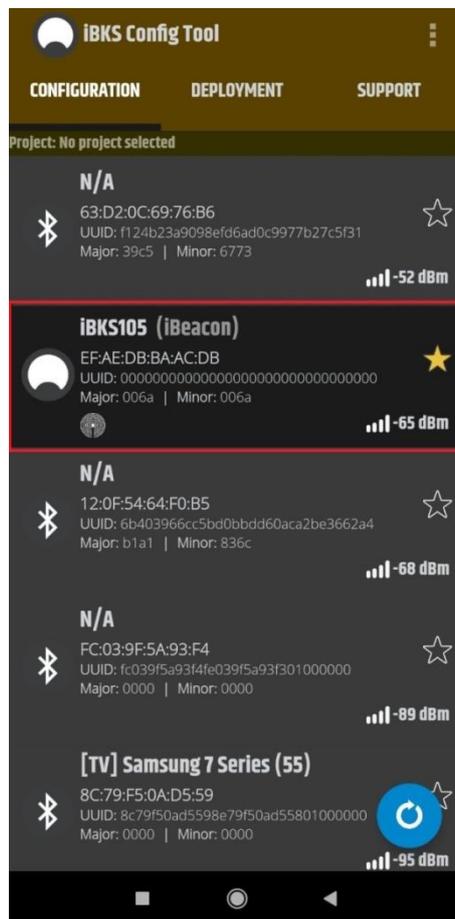
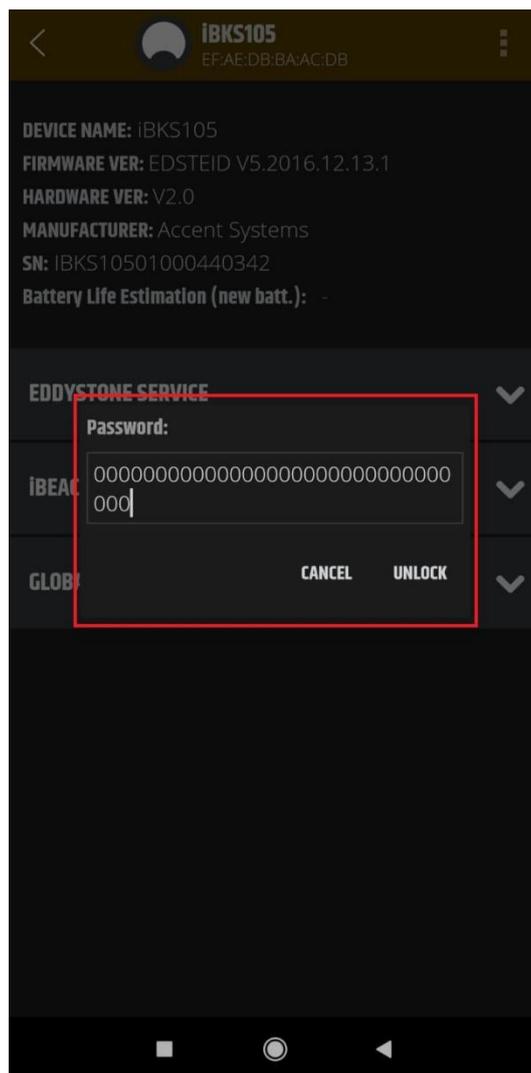


Figura 8. Paso 1.

Paso 2: Se introduce la contraseña anteriormente establecida. Se trata de una combinación de 32 números hexadecimales. En el caso de no haber cambiado nunca esta contraseña estos 32 números serán todo ceros.



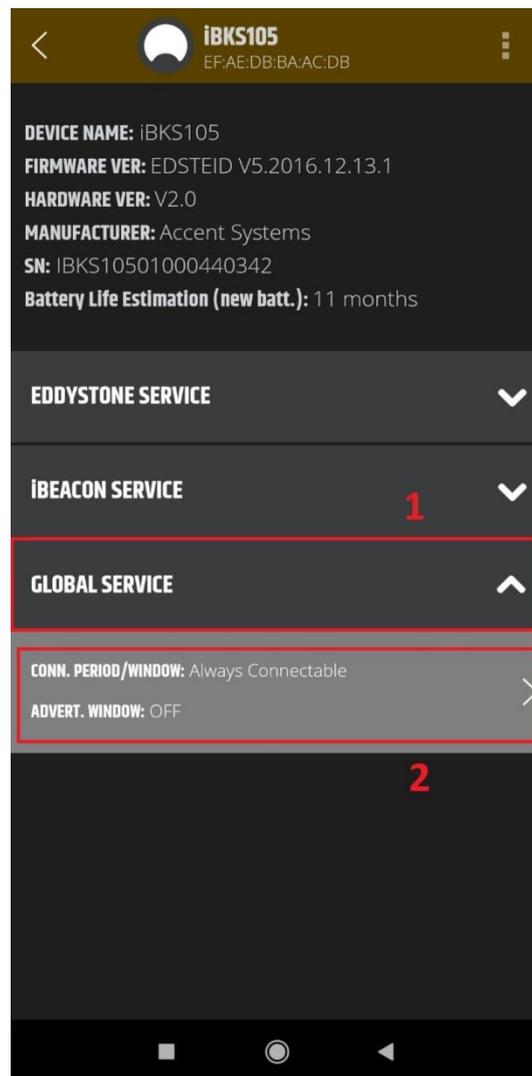
*Figura 9. Paso 2.*

---

## 5. Descripción detallada de la solución adoptada

*Diseño y desarrollo de un sistema de monitorización y limitación en tiempo real del aforo en espacios cerrados*

Paso 3: Se accede a los ajustes globales siguiendo el orden de la figura 10.



*Figura 10. Paso 3.*

Paso 4: Se activa la ventana temporal de emisión de la señal “advertising”. A continuación se selecciona la hora en la que el beacon comenzará a emitir y la hora en la que dejará de emitir. La diferencia entre estas dos no se permite que sea mayor de 17 horas. En este caso el beacon comenzará a emitir a las 8:00 AM y dejará de emitir a la 1:00 AM, completando así un ciclo completo de 17 horas emitiendo señales “advertising”.

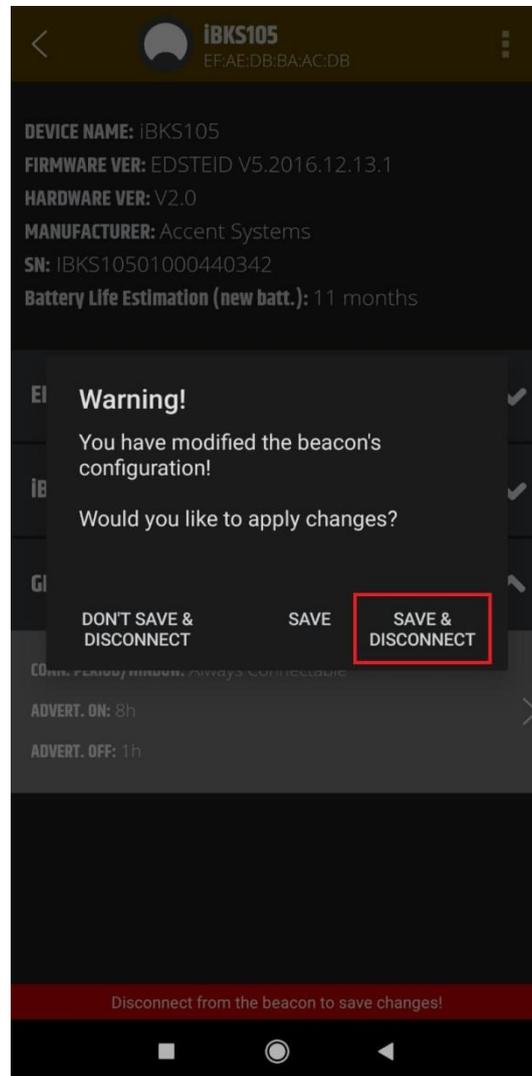


Figura 11. Paso 4.

---

## 5. Descripción detallada de la solución adoptada

Paso 5: Por último, se sale de la configuración del beacon a través de la flecha de la esquina superior izquierda y se selecciona la opción “guardar y desconectar”.



*Figura 12. Paso 5.*

### **5.2.2. Tags**

En cuanto al hardware necesario para el prototipo de los Tags, éste va a contar con 4 subsistemas. El principal subsistema es el microcontrolador escogido en el apartado 4.4., el LoPy4 [14].

Todo el conjunto de subsistemas se integrarán en un PCB de prototipado de 70 x 90 mm<sup>2</sup> de la marca LAOMAO modelo A0052, en el que se integrarán todos los componentes que habrán de soldarse posteriormente.

### **5.2.2.1. Batería**

El prototipo debe tener una batería que cumpla 2 funciones, aportar la suficiente energía para hacer funcionar todo el sistema y que tenga la suficiente capacidad energética para que su duración sea de al menos 12 meses.

Al tratarse de un prototipo, no importa que su tamaño sea extremadamente eficiente, por lo que se ha decidido utilizar una batería tipo 18650 [15]. Éstas son baterías de iones de Litio, con un voltaje nominal de 3.7 V lo que las hace perfectas para la alimentación del microcontrolador y con una capacidad de almacenamiento energético de 2400 mAh. Esto es más que suficiente para el prototipo, puesto que los dos protocolos de comunicación son de un mínimo consumo y los sistemas de señalización, estudiados en el apartado 5.2.2.3. están diseñados para consumir la mínima cantidad de energía.

La batería escogida deberá introducirse en un portapilas [16], también llamado battery holder, adecuado al tipo de pila 18650.

Adicionalmente vamos a instalar en el Tag un interruptor de tres pines [17] para desconectar la batería de todo el sistema en el momento que no se quiera utilizar, alargando así su duración por mucho más tiempo.

### **5.2.2.2. Comunicación LoRa**

Como ya se ha estudiado en el apartado 4.1.2.1., el protocolo de comunicación LoRa funciona en Europa en la frecuencia 868, por lo que el sistema va a necesitar el uso de una antena de ese rango de frecuencias. Por suerte, la propia empresa distribuidora del módulo LoPy4, Pycom, ofrece una antena [18] especialmente diseñada para este uso. El kit que vamos a implementar cuenta con la antena en sí y con el conector necesario para conectarlo al módulo.

En cuanto a la antena necesaria para establecer comunicación mediante el protocolo BLE, no es necesario que se instale una adicional, puesto que el módulo LoPy4 ya lleva una incorporada.

### **5.2.2.3. Sistemas de señalización**

En este apartado va a estudiarse el diseño eléctrico necesario para que los sistemas de señalización funcionen correctamente. De la hoja de datos del módulo LoPy4 [11] comprobamos que la corriente máxima de los pines GPIO es de 12 mA, siendo la recomendada no superior a 6 mA. Por lo que este último valor será tomado como máximo a la hora de diseñar el sistema.

Se han elegido un Buzzer magnético modelo CEM-1203(42) [19] y un motor vibrador modelo 310-101 [20]. Ambas hojas de datos muestran que las corrientes de funcionamiento (35 mA y 75 mA respectivamente) son ampliamente superiores de la corriente máxima de los pines GPIO (6 mA), por lo que será necesario diseñar un circuito amplificador de corriente mediante el uso de un transistor. Para ello hemos escogido el transistor NPN modelo BC547C [21]. Este transistor funcionará como un interruptor, dejando pasar la corriente eléctrica entre el colector y el emisor, en el momento en el que se le aplique una diferencia de potencial entre la base y el emisor superior a 0.7 V (voltaje de saturación base-emisor). A partir de los componentes escogidos se diseña dos variantes del esquema eléctrico: una para el buzzer y otra para el motor vibrador.

- Esquema eléctrico buzzer magnético:

Partimos de las ecuaciones características del transistor y de los datos relevantes tanto del transistor como del buzzer.

En la figura 13 podemos apreciar el esquema eléctrico típico de un transistor como amplificador de corriente.

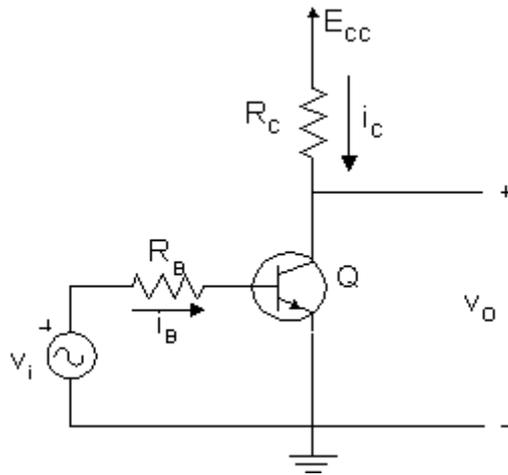


Figura 13. Esquema eléctrico transistor-amplificador

En este esquema la resistencia  $R_C$  corresponde con la impedancia del buzzer ( $42 \Omega$ ) y otra resistencia en serie en el caso de que sea necesaria,  $E_{cc}$  el voltaje de la batería ( $3.7 \text{ V}$ ) y  $V_i$  el voltaje de los pines GPIO del módulo LoPy4 ( $2.65 \text{ V}$ ). A partir de aquí se han de calcular los valores de las resistencias para que la corriente que pase por el buzzer sea la deseada.

$$E_{cc} = 3.7 \text{ V}; \quad V_i = 2.65 \text{ V}; \quad I_C = 35 \text{ mA}; \quad R_{buzzer} = 42 \Omega; \quad I_{B_{max}} = 6 \text{ mA};$$

$$V_{BE} = 0.7 \text{ V}; \quad V_{CE} = 0.1 \text{ V};$$

Si asumimos una  $R_B$  de  $1 \text{ k}\Omega$ :

$$I_B = \frac{V_i - V_{BE}}{R_B} = 1.95 \text{ mA} < I_{B_{max}}$$

$$R_C = \frac{E_{cc} - V_{CE}}{I_C} = 102.85 \Omega$$

$$R_C = R_{buzzer} + R_{aux}$$

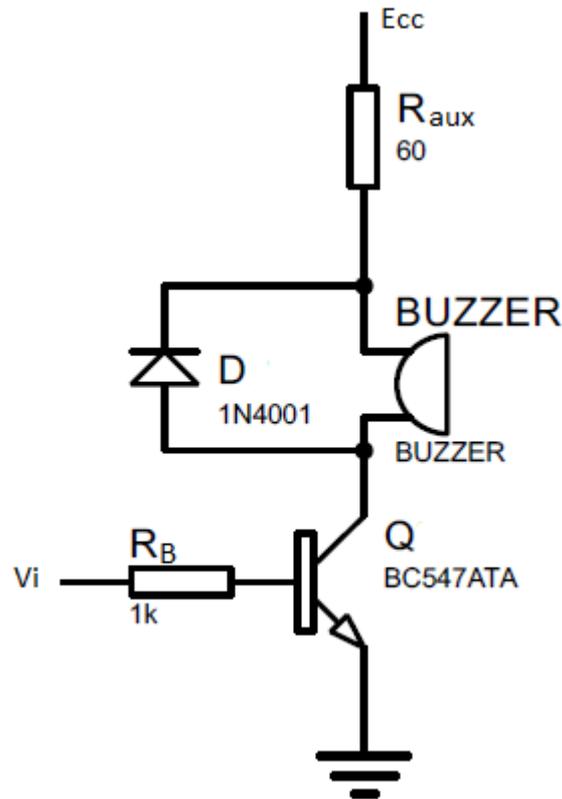
$$R_{aux} = R_C - R_{buzzer} = 60.85 \Omega$$

$$\text{Normalizando: } R_{aux} = 60 \Omega$$

---

## 5. Descripción detallada de la solución adoptada

A partir de estos resultados, podemos diseñar el esquema eléctrico de la figura 14:



*Figura 14. Esquema eléctrico del amplificador de corriente del buzzer magnético.*

Como se puede observar, se ha añadido un diodo modelo 1N4001 [22] para evitar sobrecargas en el buzzer.

- Esquema eléctrico motor vibrador:

En este caso, se procede del mismo modo que en el caso del buzzer, pero dado que el motor necesita una mayor corriente de funcionamiento, se va a modificar un poco el esquema de la figura 13. El motor se ubicará a continuación del emisor del transistor para que la corriente que pase sea la suma de la corriente del colector y la corriente de la base.

Se ha de tener en cuenta de nuevo que la impedancia del motor es de  $75 \Omega$ ,  $E_{cc}$  es el voltaje de la batería (3.7 V),  $V_i$  es el voltaje de los pines GPIO

del módulo LoPy4 (2.65 V) y que la corriente de arranque del motor es de 85 mA, por lo que se supondrá esa corriente como corriente del emisor.

$$E_{cc} = 3.7 V; \quad V_i = 2.65 V; \quad I_E = 85 mA; \quad R_{motor} = 75 \Omega; \quad I_{B\_max} = 6 mA;$$

$$V_{BE} = 0.7 V; \quad V_{CE} = 0.1 V;$$

De nuevo asumimos una  $R_B$  de 1 k $\Omega$ :

$$I_B = \frac{V_i - V_{BE}}{R_B} = 1.95 mA < I_{B\_max}$$

$$I_E = I_B + I_C; \quad I_C = I_E - I_B = 83.05 mA$$

$$R_C = \frac{E_{cc} - V_{CE}}{I_C} = 40.88 \Omega$$

$$\text{Normalizando: } R_C = 39 \Omega$$

A partir de estos resultados, podemos diseñar el esquema eléctrico de la figura 15:

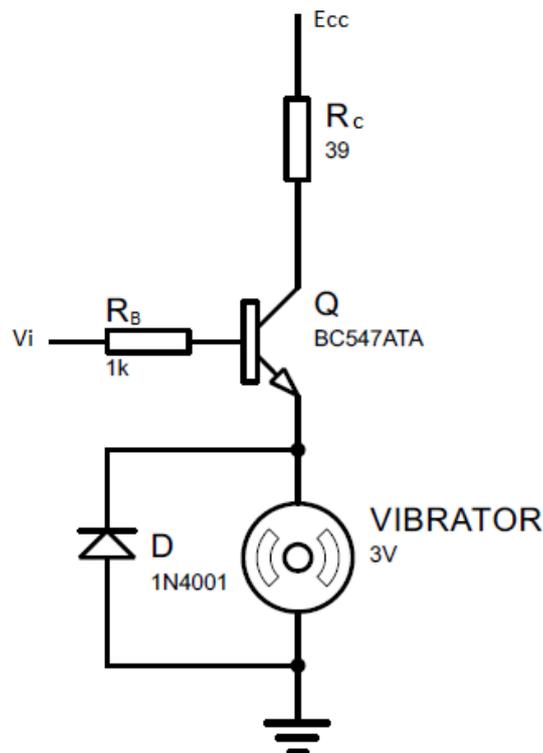


Figura 15. Esquema eléctrico del amplificador de corriente del motor vibrador.

---

## 5. Descripción detallada de la solución adoptada

Como ya se ha hecho en el esquema eléctrico del amplificador de corriente del buzzer magnético, se ha añadido el mismo diodo de protección al motor para evitar sobrecargas.

### **5.3. Diseño de los Tags**

Los Tags son el producto fundamental del sistema. Son los dispositivos que los usuarios portarán consigo, de ahí que el diseño de estos resulte tan importante.

Se asume que los Tags irán colgados mediante una cinta al cuello de los usuarios, por lo que se debe añadir una forma de enganchar la cinta con el Tag.

Como se ha especificado en las condiciones iniciales del proyecto (apartado 3.3), el diseño del Tag ha de ser de un tamaño reducido y peso ligero. Las decisiones tomadas en los apartados anteriores hacen que haya poco margen de maniobra en cuanto al peso, puesto que no se puede variar el peso de los componentes hardware. Pero el tamaño de este viene muy condicionado por la colocación de los componentes.

El material escogido es el plástico, puesto que es un material económico, ligero y suficientemente robusto para el uso que se va a hacer de este prototipo. El método de fabricación será de impresión en 3D.

El diseño del Tag se simplifica en una caja en la que los distintos componentes hardware serán introducidos y fijados. De esta idea, y teniendo en cuenta todas las medidas de los diferentes componentes, se obtienen unas medidas de 136 x 84 x 27 mm<sup>3</sup>, teniendo en cuenta unas paredes de 2 mm de grosor. Para hacer el tacto de esta caja más agradable se han redondeado las esquinas con un radio de 4 mm.

A continuación se describe la posición de cada componente dentro de esta caja de forma descendente, teniendo en mente la vista de planta de la caja con la parte superior hacia arriba:

- Enganche:

Al tratarse del elemento mediante el cual podrá colgarse el Tag, éste se situará en el exterior de la cara superior.

Se trata de un arco de radio interior de 10 mm y de radio exterior de 15 mm. A través de este arco se pasará una cinta de 60 cm de longitud que se doblará por la

mitad y se coserán sus extremos, quedando colgado el Tag a una altura razonable para una persona adulta.

- **Batería:**

Al ser el elemento más pesado, se situará, introducida en el soporte correspondiente, en la parte superior de la caja, para que los movimientos provocados por los usuarios no produzcan unas inercias muy grandes. Este soporte [16] se fijará mediante un muro situado a 22 mm del borde superior y de 10 mm de altura que recorra todo el ancho de la caja.

- **Interruptor:**

A una distancia de 4 mm del muro que separa el soporte de la batería se creará un corte en la cara izquierda de la caja con las medidas exactas del interruptor [17]. Para fijarlo únicamente hará falta sellarlo con silicona por el borde para prevenir holguras y sellarlo.

- **PCB:**

Para integrar y sujetar el PCB de prototipado se han creado unas columnas de 5 mm de altura y de 2,5 mm de radio, y encima de estas, otras columnas concéntricas de 5 mm de altura y 1 mm de radio. Éstas se sitúan centradas en la parte inferior de la caja y con la distancia entre ellas necesaria para que su posición cuadre a la perfección con la posición de los agujeros de taladro del PCB.

Estas columnas más finas se introducirán en los agujeros de taladro del PCB, y se fijarán con el uso de silicona para fijarlo y que no queden holguras.

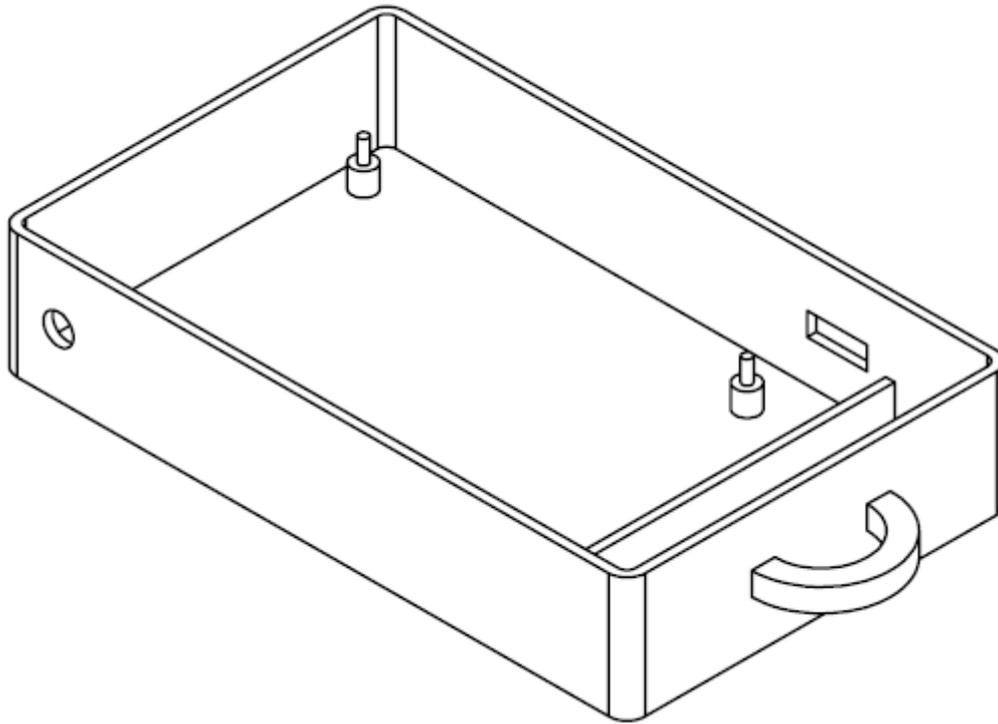
- **Antena:**

Del mismo modo que se ha procedido con el interruptor, se creará un corte en la cara derecha de la caja. Este corte será una circunferencia de radio 3,25 mm por el que pasará el conector de la antena incluido el en kit de Pycom [18].

A continuación se muestra en la figura 16 una vista isométrica del modelo creado en el software CAD SolidWorks siguiendo la descripción dada en este apartado.

---

## **5. Descripción detallada de la solución adoptada**



*Figura 16. Vista isométrica del diseño del Tag.*

En este prototipo se ha decidido dejar descubierta la tapa superior para que los componentes que lo forman sean de fácil montaje. Y, para sellarla, se usará un metacrilato unido con 4 puntos de silicona, cada uno en una de las esquinas.

#### **5.4. Diseño del software**

El software es el encargado de hacer funcionar al Tag. Éste tiene que ser lo más simple posible, para evitar un malgasto de energía en la ejecución del mismo.

Para implementarlo se usará el módulo LoPy4. En su página web [23] explica detalladamente cómo configurar el módulo para poder empezar a programarlo. En este caso se ha decidido utilizar la aplicación Visual Studio Code junto con la extensión instalada en ella Pymakr.

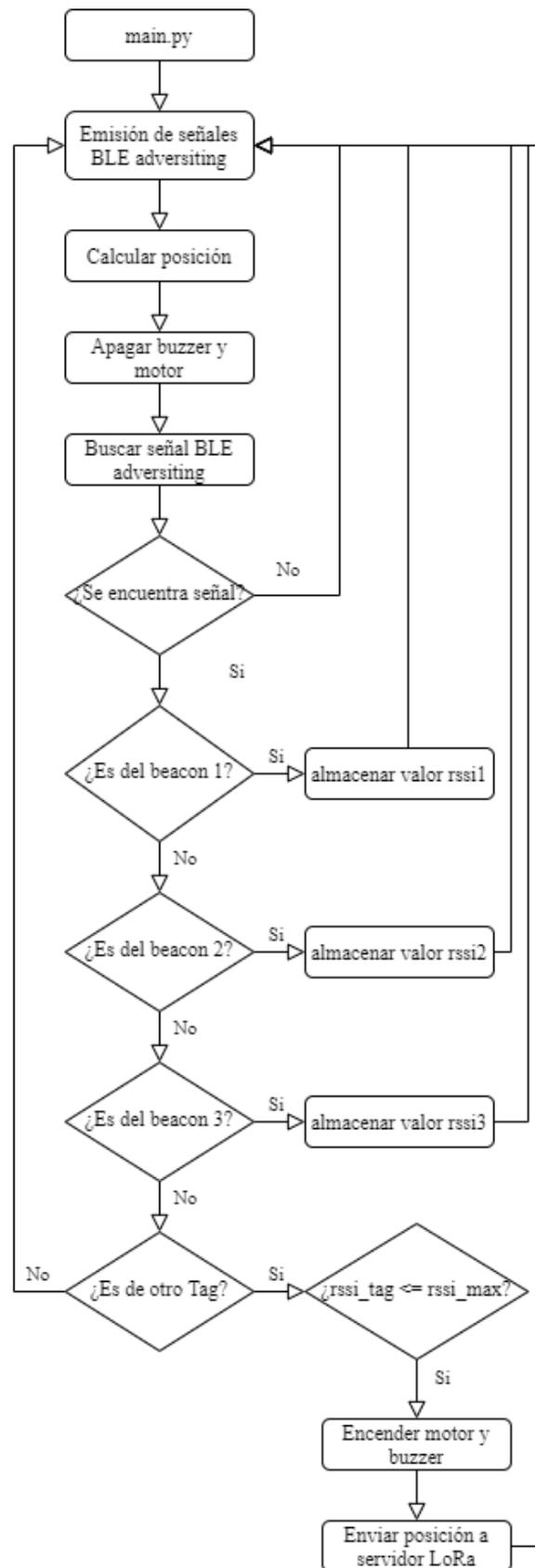
Cabe destacar que el módulo LoPy4 incorpora un LED RGB, del cual se hará uso con el único objetivo de recibir información adicional a la hora de realizar la programación. En el prototipo final no será necesaria su implementación.

El módulo tiene que programarse en lenguaje Python. Y, para su correcto funcionamiento, necesita dos archivos: *boot.py* y *main.py*. El primero se asemeja a la función *setup()* de Arduino. Se trata de un archivo que se ejecuta al inicializarse el módulo, éste puede servir para hacer comprobaciones sobre el correcto funcionamiento de los periféricos o similares. En este proyecto vamos a usarlo para apagar un parpadeo en el LED RGB que viene predefinido de fábrica, comprobar el uso de ese LED RGB enviando diferentes señales correspondientes a diferentes colores y como una comprobación del funcionamiento del motor y el buzzer.

En cambio el archivo *main.py* se trata del archivo principal del módulo. En este se establecerán los parámetros iniciales y se entrará en un bucle infinito siguiendo el diagrama de flujo que se puede observar en la figura 17. De este archivo se pueden analizar los siguientes subapartados: posicionamiento del Tag, la comunicación BLE entre beacons-Tags y Tags-Tags, la comunicación con el servidor LoRa y la configuración de las señales de salida hacia el motor y el buzzer. Se puede encontrar todo el código en el Anexo I.

---

## 5. Descripción detallada de la solución adoptada



*Figura 17. Diagrama de flujo del funcionamiento del código.*

### 5.4.1. Posicionamiento del Tag

Para calcular la posición del Tag en cada instante vamos a utilizar la metodología empleada en el estudio *Indoor Vehicles Geolocalization Using LoRaWAN* [4]. En este estudio se emplea el protocolo LoRa para realizar el cálculo, pero la metodología es idéntica con cualquier tipo de señal electromagnética, por lo que se puede aplicar a nuestro caso.

Se comienza definiendo el espacio a trabajar. Para este sistema se ha definir un espacio diferente según las necesidades del cliente. En este caso el código está pensado para una habitación cerrada de 5 metros de ancho por 8 de largo. Está claro que este sistema está pensado para superficies mucho más grandes, pero las condiciones en las que lo vamos a simular no precisan de ser demasiado extensas.

El posicionamiento de los beacons se realizará según se muestra en la figura 18.

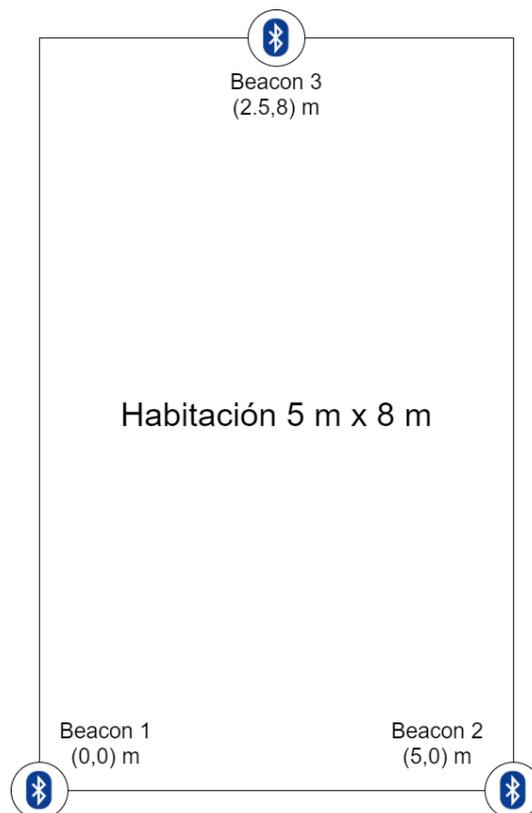


Figura 18. Posicionamiento de los Beacons en la habitación a simular.

---

## 5. Descripción detallada de la solución adoptada

A continuación se hace uso del algoritmo matemático de triangulación circular, explicado en el apartado 4.1.1. Para ello será necesario conocer la distancia a la que se encuentra el Tag de cada beacon. Esto se puede hacer, como se explica en el artículo previamente mencionado [4], con la siguiente fórmula:

$$d = 10^{\frac{\text{Potencia medida} - \text{RSSI}}{10 \cdot n}}$$

Siendo  $d$  la distancia a calcular, *Potencia Medida* el rssi que medimos empíricamente desde una distancia de 1 metro a la fuente de señales, en este caso el beacon. Esta medida se ha hecho experimentalmente y ha dado un resultado de -60. *RSSI* es el valor medido para el que se quiere conocer la distancia y por último  $n$  es un factor llamado gradiente distancia-potencia, que en condiciones ideales es  $n = 2$ , pero en situaciones realistas este factor adquiere un valor entre 2 y 6, en este caso se usará  $n = 4$ .

Por último llega el momento de calcular la posición relativa del Tag con respecto a los 3 beacons. El procedimiento matemático está bien explicado en el artículo *Indoor Vehicles Geolocalization Using LoRaWAN* [4], por lo que a continuación se muestran las ecuaciones finales que se implementarán en la función que sirva para calcular la posición del Tag.

$$\begin{cases} x = \frac{CE - FB}{EA - BD} \\ y = \frac{CD - AF}{BD - EA} \end{cases}$$

$$A = -2x_{b1} + 2x_{b2}$$

$$B = -2y_{b1} + 2y_{b2}$$

$$C = d_{b1}^2 - d_{b2}^2 - x_{b1}^2 + x_{b2}^2 - y_{b1}^2 + y_{b2}^2$$

$$D = -2x_{b2} + 2x_{b3}$$

$$E = -2y_{b2} + 2y_{b3}$$

$$F = d_{b2}^2 - d_{b3}^2 - x_{b2}^2 + x_{b3}^2 - y_{b2}^2 + y_{b3}^2$$

Siendo  $x_{bn}$  e  $y_{bn}$  la posición del beacon  $n$ ,  $d_{bn}$  la distancia medida al beacon  $n$ , y  $X$  e  $Y$  las coordenadas en las que se posiciona el Tag.

Este algoritmo se implementará mediante dos funciones. La primera, llamada *calc\_dist()* será la encargada de pasado un valor rssi calcular a cuánta distancia se encuentra la fuente emisora. A continuación podemos ver el código propuesto:

```
def calc_dist(rssi):
    n = 4 #valor del parámetro n, idealmente 2, en el mundo real entre 2 y 6
    return 10**((-60 -rssi)/(10*n))
```

La segunda función se llamará *position()*, se le pasarán 3 parámetros que corresponden con la distancia a cada uno de los beacons, calculada con la función *calc\_dist()* y será la encargada de implementar las fórmulas previamente descritas para conseguir los valores  $x$  e  $y$  en los que se ubica el Tag.

```
def position(dist_b1, dist_b2, dist_b3):
    #inicializamos las posiciones de los beacons
    #beacon 1
    x1 = 0
    y1 = 0
    #beacon 2
    x2 = 5
    y2 = 0
    #beacon3
    x3 = 2.5
    y3 = 8

    A = -2 * x1 + 2 * x2
    B = -2 * y1 + 2 * y2
    C = (dist_b1)**2 - (dist_b2)**2 - (x1)**2 + (x2)**2 - (y1)**2 + (y2)**2
    D = -2 * x2 + 2 * x3
    E = -2 * y2 + 2 * y3
    F = (dist_b2)**2 - (dist_b3)**2 - (x2)**2 + (x3)**2 - (y2)**2 + (y3)**2

    calculated_x = ((C*E)-(F*B))/((E*A)-(B*D))
    calculated_y = ((C*D)-(A*F))/((B*D)-(A*E))

    return calculated_x, calculated_y
```

---

## 5. Descripción detallada de la solución adoptada

## 5.4.2. Comunicación BLE

La comunicación BLE es la más vital del sistema, puesto que en ella se basa tanto el sistema de posicionamiento como la señal de alarma que avisa que no se cumple con la distancia de seguridad.

En la web de Pycom, se pueden encontrar un extenso y detallado manual [24] en el que explica cómo utilizar la clase Bluetooth dentro de sus módulos.

Dentro del archivo *main.py* se tiene que crear un objeto de clase Bluetooth.

```
#Creamos el objeto Bluetooth
bt = Bluetooth()
```

A continuación, y basándonos en el diagrama de flujo de la figura 17, habrá que emitir las señales advertising, para ello, primero se usará la función *bluetooth.set\_advertisement()*, en la que la única información útil que se enviará será el nombre del dispositivo, en este caso 'Tag'. Y después usaremos la función *bluetooth.advertise()* para comenzar a emitir.

```
#configuramos la señal advertisement que será emitida
bt.set_advertisement(name = 'Tag')
#comenzamos a emitir advertisements
bt.advertise(True)
```

Según el diagrama de flujo de la figura 17, esta es la función por la que debería comenzar el bucle infinito, pero este módulo es capaz de estar continuamente emitiendo a pesar de estar ejecutando otra parte del programa principal, por lo que no será necesario volver a llamar a esta función.

Una vez entramos en el bucle infinito, se ha de estar buscando señales advertising que recibe el módulo y de las que lleguen deberemos analizar el nombre del dispositivo emisor, para diferenciar si provienen de uno de los beacons o de otro Tag.

En el caso de que las señales procedan de la primera fuente, encenderemos el LED RGB de un color característico por cada beacon y almacenaremos el valor de su rssi. A continuación podemos ver el ejemplo del beacon 1, el cual tiene el nombre de iBKS105\_1.

```

#buscamos señales advertising
adv = bt.get_adv()

if adv:

    #señal advertisement beacon 1
    if bt.resolve_adv_data(adv.data, Bluetooth.ADV_NAME_CMPL) == 'iBKS105_1':

        pycom.rgbled(0x000011) #encendemos led azul

        rssi1 = adv.rssi #almacenamos el valor rssi del beacon 1

```

En el caso de recibir una señal advertising de un Tag se procede con el mismo código pero comprobando si el nombre del dispositivo es 'Tag'.

### 5.4.3. Comunicación LoRa

La comunicación LoRa se iniciará en el momento en el que un valor rssi proveniente de un Tag sea mayor que el establecido como distancia de seguridad (1,5 metros). Este valor se puede sacar despejando la variable *RSSI* de la fórmula utilizada en el apartado 5.4.1 para calcular la distancia. Teniendo en cuenta que la *Potencia Medida* en este caso ha cambiado a -50, este valor del rssi máximo será de -57 y se definirá como una variable global.

Antes de intentar comunicarse, se debe de haber registrado al dispositivo en un servidor LoRa para tener acceso a este. En este caso la balanza se ha decantado por el servidor *The Things Network* [25]. Éste es un servidor miembro de la LoRa Alliance [3], abierto y gratuito para todo el mundo. Únicamente es necesario tener cobertura de alguno de los gateways de la amplia red que tiene conectados en todo el mundo. Aquí en Valencia hay 4 en la Universitat Politècnica de València.

Una vez esto decidido hay que registrar el dispositivo siguiendo las instrucciones que se dan tanto en la web de Pycom [23] como en la de *The Things Network*. Teniendo en cuenta que para este caso se va a establecer una conexión LoRaWAN ABP debido a que no requiere permiso desde el Gateway receptor de la señal, The Things Network creará las siguientes cadenas hexadecimales:

- Device Address

---

## 5. Descripción detallada de la solución adoptada

## *Diseño y desarrollo de un sistema de monitorización y limitación en tiempo real del aforo en espacios cerrados*

- Network Session Key
- App Session Key

Las tres serán necesarias como parámetros en nuestro código, para que el módulo LoPy4 pueda establecer conexión.

Como hemos visto anteriormente, Pycom posee un extenso y detallado manual [24] en el que explica cómo utilizar la clase Bluetooth dentro de sus módulos, pero también posee otro manual [26] en el que explica cómo utilizar la clase LoRa.

En este caso crearemos un objeto LoRaWAN y lo inicializaremos con los parámetros característicos. Y se crearán las variables globales con los códigos hexadecimales obtenidos del servidor.

```
#configuramos el objeto LoRa
lora = LoRa(mode = LoRa.LORAWAN, region = LoRa.EU868)

#Configuramos los parámetros LoRa
lora.init(mode = LoRa.LORAWAN, region = LoRa.EU868, power_mode = LoRa.TX_ONLY
, adr = False, public = True, tx_retries = 2, device_class = LoRa.CLASS_A)

#definimos los parámetros, dados por el servidor de LoRa, necesarios para
conectarnos
dev_addr = ubinascii.unhexlify('26013430')

nwk_swkey = ubinascii.unhexlify('7349D8E7DEF523503CF2A02D95770E03')

app_swkey = ubinascii.unhexlify('0FD71B550AE9E67C236D0E406BB1D2A4')
```

Volviendo al caso con el que se ha comenzado el subapartado, el código necesario para establecer la conexión con el servidor mediante LoRaWAN ABP será el siguiente.

```
#nos conectamos con el servidor LoRa
lora.join(activation = LoRa.ABP, auth = (dev_addr, nwk_swkey, app_swkey))

if lora.has_joined():

    s = socket.socket(socket.AF_LORA, socket.SOCK_RAW)
    s.setsockopt(socket.SOL_LORA, socket.SO_DR, 5)
    s.setblocking(True)
    s.send("ubicación: x:" + str(x) + " y:" + str(y)) #enviamos la ubicac
ión en coordenadas x e y
```

```
s.setblocking(False)
```

#### 5.4.4. Configuración de salidas GPIO

Por último, otra parte fundamental del código es la configuración de las salidas GPIO a las que irán conectadas el motor vibrador y el buzzer. Se tratan de Pines de Entrada/Salida de propósito general, por lo que su implementación en el código es muy sencilla. Simplemente hay que definir a qué pin conectaremos cada periférico y en qué modo trabajará ese pin. En el código que podemos ver a continuación se muestra su configuración.

```
#definimos el pin 22 para el motor vibrador como pin de salida
motor = Pin('P22', mode = Pin.OUT)

#definimos el pin 23 para el buzzer como pin de salida
buzzer = Pin('P23', mode = Pin.OUT)
```

Para dar valores a estos pines habrá que llamar a la función *value()* implementada en el objeto *Pin* de la librería *machine*.

## **6. Pruebas realizadas**

---

En el siguiente apartado se va a mostrar el montaje y construcción del prototipo realizado por el autor del proyecto. También se explicarán una serie de pruebas de test que se realizarán y se analizarán los resultados.

### **6.1. Montaje del prototipo**

En primer lugar se procede a la impresión en 3D de la caja del Tag, en las figuras 19 y 20, se puede observar el proceso de impresión y la caja una vez finalizada respectivamente.

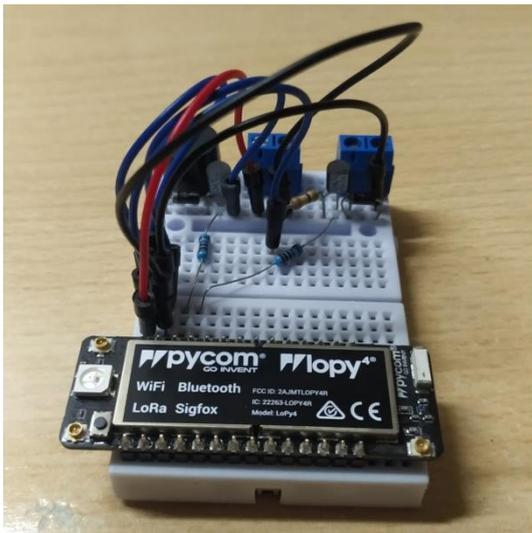


*Figura 19. Proceso de impresión 3D.*

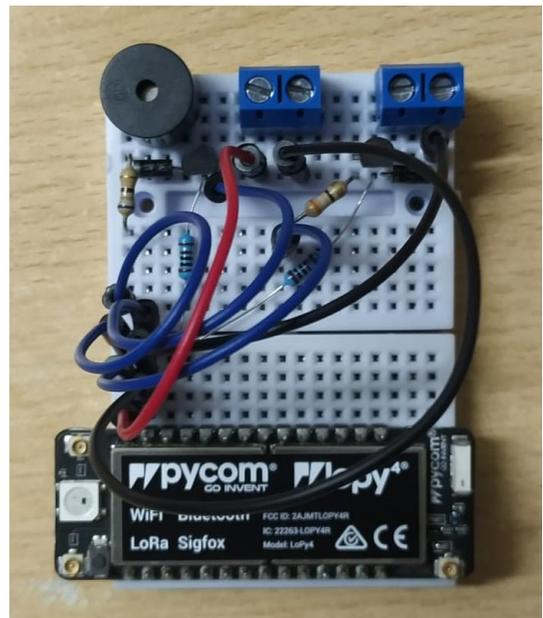


*Figura 20. Caja prototipo impresa en 3D.*

A continuación se sueldan los componentes al PCB de prototipado. En este punto, nos encontramos con el problema de la pésima calidad de soldadura del material usado para ello, en su defecto se procedió a usar las conocidas como breadboards para el montaje del circuito electrónico. Aquí conectamos según el esquema eléctrico diseñado todos los componentes excepto la batería, el interruptor, el motor vibrador y la antena. El resultado de este paso se puede ver en las figuras 21 y 22.



*Figura 20. Circuito montado en breadboard. Vista 1.*



*Figura 22. Circuito montado en breadboard. Vista 2.*

Por último, fijaremos con silicona los demás componentes a la caja y los conectaremos con el circuito ya montado en la breadboard. Como remate final coseremos una cinta al enganche de la parte superior para poder colgarlo al cuello. Podemos ver una imagen del resultado final en la figura 23.

---

## 6. Pruebas realizadas



*Figura 23. Montaje final del Tag.*

Opcionalmente podemos ponerle una lámina de metacrilato y fijarla a las cuatro esquinas de la caja con silicona, pero por comodidad a la hora de reprogramar el microcontrolador, se ha decidido dejar al aire.

## **6.2. Test de funcionamiento**

A continuación, se someterá el prototipo a 2 pruebas mediante las cuales comprobaremos el correcto funcionamiento del sistema. Éstas se realizarán al aire libre

colocando los beacons como se muestra en la figura 18, simulando una habitación de 5 x 8 metros.

### **6.2.1. Prueba 1**

La primera prueba consistirá en simular un uso normal del sistema por un usuario final. Para ello 2 personas portarán los Tags colgados y se acercarán, comprobando si se activan las alarmas cuando se incumpla la distancia de seguridad.

Tras realizar la prueba, se han obtenido resultados satisfactorios, puesto que los Tags pitaban y vibraban cuando se acercaban demasiado y dejaban de hacerlo cuando se alejaban más allá de la distancia de seguridad, aunque a veces tardaban más debido a que el microcontrolador estaba esperando a la confirmación de la conexión con el servidor LoRa.

### **6.2.2. Prueba 2**

En la segunda prueba se comprobará el buen funcionamiento del sistema de posicionamiento, ya que esta será la información que se le enviará al servidor. Para ello se conectará el microcontrolador a un PC portátil mediante la placa de expansión de Pycom [27] y se modificará ligeramente el programa para que muestre por el puerto serie el mensaje que enviaría al servidor. A continuación se colocarán muy cerca primero del beacon 1 y después del beacon 2, y se acercará una persona con un Tag encendido hasta que salte la alarma indicando que se ha incumplido la distancia de seguridad. En ese momento se mostrará por el puerto serie la coordenadas calculadas por el microcontrolador con respecto a los beacons de la sala.

Tras realizar la prueba se obtuvieron los siguientes resultados:

- En el caso de colocar el portátil cerca del beacon 1 (0,0):  
 $x = 0,720432$ ,  $y = 1.120546$ .
- En el caso de colocar el portátil cerca del beacon 2 (5,0):  
 $x = 4,480601$ ,  $y = 0,352019$ .

Estos datos se encuentran dentro del margen de error de 2,5 metros establecido en las especificaciones iniciales del apartado 3.3.

---

## **6. Pruebas realizadas**

## **7. Conclusiones**

---

### **7.1. Sobre el trabajo realizado**

Tras realizar este proyecto podemos sacar varias conclusiones. La primera y la más evidente es la puesta en práctica de una gran cantidad de conocimientos adquiridos durante el curso del Grado en Ingeniería Electrónica Industrial y Automática. Conocimientos relacionados tanto con circuitería electrónica básica, diseño CAD, programación en diferentes lenguajes, etc.

En relación con el contexto actual originado por la pandemia mundial del Covid-19, este proyecto se ha mostrado eficaz a la hora de cumplir con una necesidad básica como es el distanciamiento social, por lo que podemos estar satisfechos en ese aspecto.

Otra conclusión que se saca es la necesidad de seguir aprendiendo, si hay algo de lo que he sido consciente en el momento en el que hacía el trabajo, es de lo mucho que me queda por aprender tanto de IoT, como de una gran cantidad de áreas de trabajo relacionadas tanto con el mercado actual, como con intereses propios o mera curiosidad.

Este trabajo ha supuesto un reto para mí en muchos aspectos, puesto que me he visto en la situación de aprender un lenguaje de programación totalmente nuevo, buscar información específica de las diferentes ofertas en el mercado y tratar de dar siempre con la mejor solución aunque eso signifique invertir mucho más tiempo en ello.

Para concluir, puedo exponer que el resultado ha sido satisfactorio desde un punto de vista profesional, puesto que se ha conseguido cumplir con los objetivos marcados para el desarrollo del proyecto. Pero también personal, debido a que ha afianzado mis conocimientos teóricos y prácticos.

### **7.2. Mejoras**

Una vez acabado y visto con perspectiva, el proyecto goza de un gran margen de mejora en gran parte de sus características.

Una de las más evidentes es el tamaño del dispositivo portátil, al no diseñar nosotros directamente el PCB, si no que usamos uno que se suele utilizar para prototipado, el tamaño que obtenemos que dejan los componentes dentro del dispositivo es exageradamente grande, así como el tamaño completamente desproporcionado de la antena LoRaWAN.

Esto último podría corregirse si se utilizara una antena embebida, pero esta solución no es posible implementarla en nuestro prototipo por lo expuesto en el apartado anterior. En el caso de que lo hubiésemos hecho, el tamaño total se vería considerablemente reducido.

Por último otra mejora posible sería la capacidad del sistema para que varios Tags estuviesen vinculados entre ellos, dando la posibilidad a que, aunque no se respetara la distancia de seguridad, éstos no emitiesen ningún tipo de alarma.

## 8. Bibliografía

---

- [1] Erik Ortiz Guerra, Samuel Montejo Sánchez. Algoritmos y métodos para la localización en interiores. Universidad Central “Marta Abreu” de Las Villas, Santa Clara.
  - [2] Página oficial LoRaWAN. <https://lorawan.es/>
  - [3] Página oficial LoRa Alliance. <https://lora-alliance.org>
  - [4] Pietro Manzoni, Carlos T.Calafate, Juan-Carlos Cano and Enrique Hernández-Orallo. Indoor Vehicles Geolocalization Using LoRaWAN. Universitat Politècnica de València, 31 May 2019.
  - [5] Kevin Townsend. Introduction to Bluetooth Low Energy, 2020
  - [6] Ramsey Faragher and Robert Harle. Location Fingerprinting With Bluetooth Low Energy Beacons. IEEE Journal on selected areas in communications, vol. 33, november 2015.
  - [7] Página Wikipedia de WiFi. <https://es.wikipedia.org/wiki/Wifi>
  - [8] Página oficial SigFox. <https://www.sigfox.com>
  - [9] Link de venta Arduino MKR WAN. <https://store.arduino.cc/mkr-wan-1310>
  - [10] Servidor IoT de Arduino. <https://www.arduino.cc/en/IoT/HomePage>
  - [11] Módulo LoPy4 con 4 tecnologías de comunicación LoRa, Sigfox, BLE y WiFi.  
[https://pycom.io/product/lopy4/?gclid=CjwKCAjw1ej5BRBhEiwAfHyh1C3NQTFEovkd\\_yFmpjyt0hhEOfejwD\\_1lR1IHH0dDpxi3i34RRo0DRoCTXMQAvD\\_BwE](https://pycom.io/product/lopy4/?gclid=CjwKCAjw1ej5BRBhEiwAfHyh1C3NQTFEovkd_yFmpjyt0hhEOfejwD_1lR1IHH0dDpxi3i34RRo0DRoCTXMQAvD_BwE)
  - [12] Página Wikipedia de Baliza Electrónica (Beacon). [https://es.wikipedia.org/wiki/Baliza\\_electr%C3%B3nica](https://es.wikipedia.org/wiki/Baliza_electr%C3%B3nica)
  - [13] Página web oficial de accent systems, beacon iBKS 105. <https://accent-systems.com/es/producto/ibks-105/>
  - [14] LoPy4 specsheets v1.1, March 2020.
  - [15] Baterías 18650 Li-ion 3.7V. <https://tienda.bricogeek.com/baterias-lipo/1438-pack-de-2-bater%C3%ADas-li-ion-18650-2400mah-37v.html>
  - [16] 18650 Battery Holder datasheet. May 2019.
-

- [17] E-Switch EG1218 datasheet, October 2007.
- [18] Sigfox/LoRa Antenna kit datasheet, by Pycom.
- [19] Magnetic Buzzer CEM-1203(42) datasheet. March, 2006.
- [20] Shaftless Vibration Motor 310-101 datasheet. 2008.
- [21] BC547C Amplifier NPN Transistor datasheet, 1996.
- [22] Diode 1N4001 datasheet. 2006.
- [23] Página web oficial de Pycom. <https://pycom.io/>
- [24] Enlace web a la descripción del firmware de la comunicación BLE del módulo LoPy4. <https://docs.pycom.io/firmwareapi/pycom/network/bluetooth/>
- [25] Páginal oficial del servidor LoRaWAN The Things Network <https://www.thethingsnetwork.org/>
- [26] Enlace web a la descripción del firmware de la comunicación LoRa del módulo LoPy4. <https://docs.pycom.io/firmwareapi/pycom/network/lora/>
- [27] Placa de expansión Pycom. <https://pycom.io/product/expansion-board-3-0/>



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

*Diseño y desarrollo de un sistema de  
monitorización y limitación en tiempo real del  
aforo en espacios cerrados*

## 2. Planos

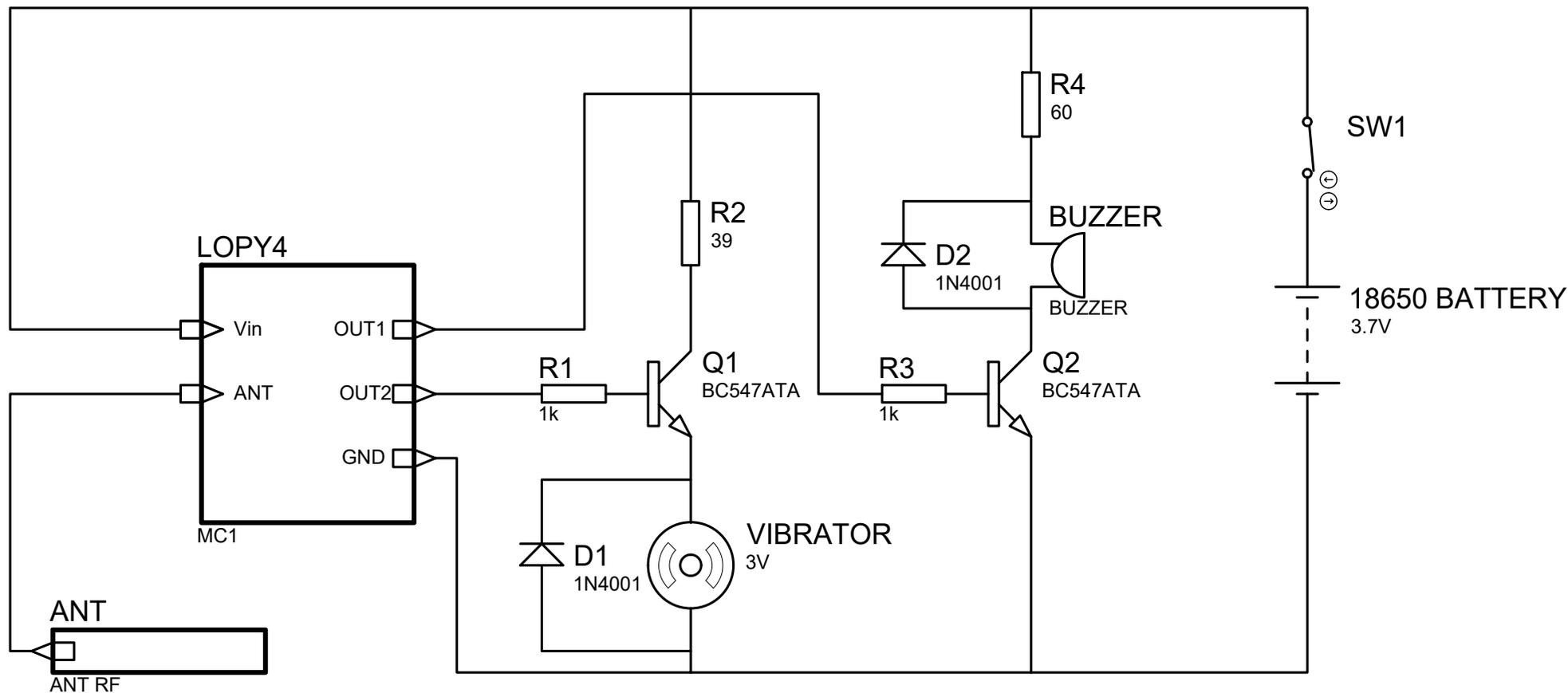
**Autor:**

D. Ferran Ballester Mainar

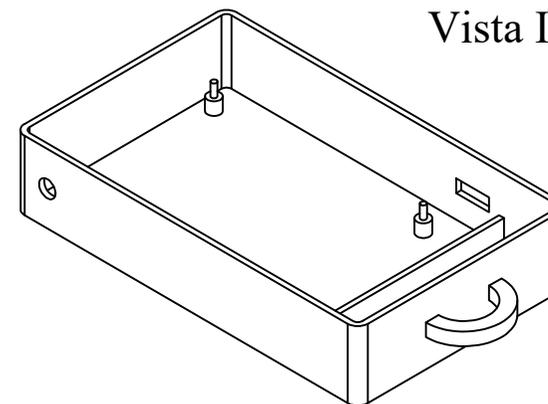
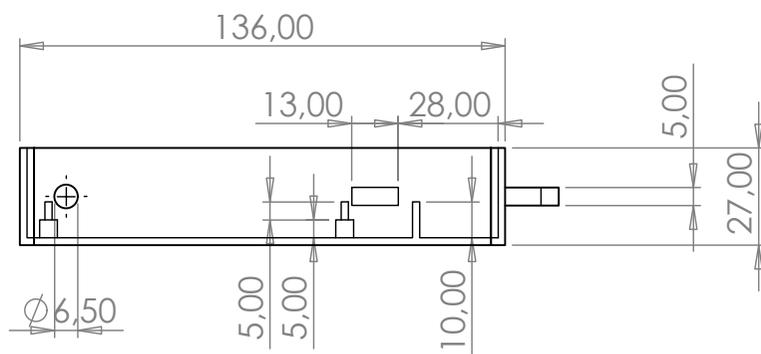
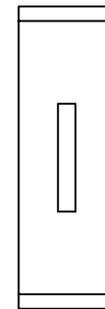
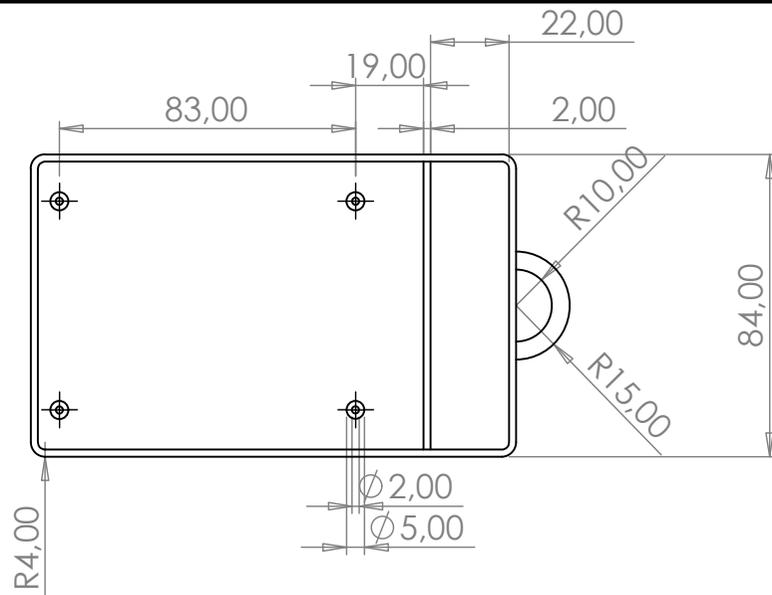
**Tutor:**

D. Ángel Perles Ivars

Alboraia, noviembre de 2020



Nº Plano: <b>1</b>	Nombre Plano: <b>Esquema Eléctrico Tag</b>		 <b>UNIVERSITAT POLITÈCNICA DE VALÈNCIA</b>
Título Proyecto: Diseño y desarrollo de un sistema de monitorización y limitación en tiempo real del aforo en espacios cerrados	Autor: <b>Ferran Ballester Mainar</b>		
	Escala: <b>1:1</b>	Referencia: <b>20-001</b>	Fecha: <b>01/11/2020</b>



Vista Isométrica

Nº Plano: <b>2</b>	Nombre Plano: <b>Carcasa Tag</b>		 <b>UNIVERSITAT POLITÈCNICA DE VALÈNCIA</b>
Título Proyecto: Diseño y desarrollo de un sistema de monitorización y limitación en tiempo real del aforo en espacios cerrados	Autor: <b>Ferran Ballester Mainar</b>		
	Escala: <b>1:2</b>	Referencia: <b>20-002</b>	Fecha: <b>01/11/2020</b>



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

*Diseño y desarrollo de un sistema de  
monitorización y limitación en tiempo real del  
aforo en espacios cerrados*

### **3. Pliego de condiciones**

**Autor:**

D. Ferran Ballester Mainar

**Tutor:**

D. Ángel Perles Ivars

Alboraia, noviembre de 2020

# Índice

---

<b>1. Definición y alcance del pliego .....</b>	<b>2</b>
<b>2. Condiciones y normas de carácter general .....</b>	<b>3</b>
2.1. Instalación .....	3
2.2. Seguridad.....	3
2.3. Utilización .....	4
2.4. Mantenimiento .....	4
<b>3. Condiciones particulares y fabricación .....</b>	<b>5</b>
3.1. Materiales.....	5
3.2. Fabricación de la caja del Tag.....	5
3.3. Montaje de los componentes del Tag .....	6
3.4. Componentes electrónicos.....	6

## 1. Definición y alcance del pliego

---

El presente documento corresponde a la especificación técnica del diseño e implementación de un prototipo de un sistema de monitorización y control del aforo en espacios cerrados haciendo uso de tecnologías inalámbricas IoT.

El presente pliego de condiciones pretende describir las condiciones bajo las cuales se debe fabricar, reproducir y hacer uso del sistema cuyo diseño se explica en la memoria, así como los planos y toda la documentación técnica aportada.

El alcance de este documento abarca la descripción tanto del proceso de fabricación de los componentes propios, como del proceso de montaje de estos para integrarlos en el dispositivo principal, llamado Tag. Así mismo se pretende describir el proceso de instalación y de testeo, y las condiciones de mantenimiento a las que deberá estar sometido el sistema una vez instalado.

## 2. Condiciones y normas de carácter general

---

El prototipo del sistema se ha diseñado bajo unas condiciones de funcionamiento específicas, por lo que tanto su uso y fabricación deben adecuarse con ellas. A continuación se exponen una serie de condiciones y normas de carácter general.

### 2.1. Instalación

Como se trata de un prototipo, la instalación tendrá que efectuarse por personal cualificado que conozca en profundidad el funcionamiento del sistema.

La comunicación con el cliente ha de ser fundamental en el periodo de desarrollo, puesto que el software tiene que integrar las coordenadas en las que se ubicarán los beacons.

Por otra parte se necesitará una suscripción a un servidor LoRaWAN, del cual será responsable el cliente.

Una vez completados estos pasos, se tendrá que realizar un pequeño test en el que comprobaremos el correcto funcionamiento de 2 Tags. Si las medidas enviadas al servidor de LoRaWAN son correctas, se dará por favorable la instalación del sistema y se podrá proceder a la instalación del software en los demás dispositivos. En caso contrario se deberá recalibrar el software y volver a realizar la prueba.

### 2.2. Seguridad

En cuanto a la seguridad del sistema, los Tags no han de ser manipulados internamente por personal no autorizado para no dañar sus componentes. También habrá que tener especial cuidado con la manipulación de la antena, ya que al estar colocada en el exterior del dispositivo pueden sufrir daños.

El dispositivo portátil está clasificado como seguro, ya que su fuente de energía son baterías de bajo voltaje. También se han tenido en cuenta posibles sobretensiones por cargas electroestáticas y se ha equipado al sistema de 2 diodos de protección para cada actuador.

---

## 2. Condiciones y normas de carácter general

Se ha tenido en cuenta en el desarrollo del dispositivo portátil la siguiente normativa, que deberá ser revisada para cada versión o variación que se le efectúe al sistema:

- 206/95/CE de baja tensión y seguridad general.
- Cumplimiento de los estándares de emisión electromagnética UNE-EN 61000-6-3 y 61000-6-4, y del estándar de producto EN 55032.

El módulo LoPy4 está certificado tanto por el mercado CE como por la LoRa Alliance.

### **2.3. Utilización**

El primer paso será colocar el Tag colgado del cuello del usuario que vaya a portarlo. Los Tags han de encenderse cambiando de posición el interruptor de forma ordenada, y una vez encendido uno, alejarlo al menos 2 metros de los que aún no lo estén. Esto prevendrá que en el momento en el que se conecten los demás salte la alarma de proximidad y sea molesto para usuarios presentes.

Al hacer uso de la red LoRaWan *The Things Network* no es responsabilidad del autor del proyecto que esté disponible y funcionando continuamente, ya que es una red de uso abierto y comunitario. En el caso en el que el cliente final desee conectarse a un servidor diferente, la garantía de que se establezca comunicación recae en los responsables del servidor.

Se debe tener en cuenta que se trata de un prototipo, por lo que no está asegurado su funcionamiento libre de fallos.

### **2.4. Mantenimiento**

Este sistema requiere de un mantenimiento mínimo.

Periódicamente se deberá comprobar el correcto funcionamiento tanto de los beacons como de los Tags, con el objetivo de que únicamente sea necesario cambiar las baterías de éstos cuando alguna falle.

Al tratarse de un prototipo, muy posiblemente el sistema sufrirá actualizaciones y mejoras, las cuales corren a cargo del autor del proyecto.

## 3. Condiciones particulares y fabricación

---

### 3.1. Materiales

A continuación se muestra una lista de todos los materiales que se usarán en la fabricación del sistema.

- Beacon iBKS105, Accent Systems.
- Ácido Poliláctico (PLA).
- Módulo LoPy4, Pycom.
- Batería 18650 3.7 V.
- Soporte de batería 18650.
- Interruptor de 3 pines.
- Cable de cobre de 0.49 mm<sup>2</sup>.
- Conector Screw Terminal.
- Motor vibrador 310-101.
- Zumbador magnético CEM-1203(42).
- Transistor BC547C.
- Resistencias de 39  $\Omega$ , 60  $\Omega$  y 1 k  $\Omega$ .
- Diodo 1N4001.
- Antena Sigfox/LoRaWAN, Pycom.
- Conector coaxial RF, Pycom.
- Aleación de estaño y plomo 60% - 40% respectivamente.
- Cola termofusible (silicona).
- Cinta de raso de anchura 1,5 cm.
- Lámina de metacrilato transparente de 2 mm de grosor.

### 3.2. Fabricación de la caja del Tag

La fabricación de la caja en la que se montarán todos los componentes se producirá de forma exclusiva mediante el uso de impresoras 3D. Se utilizará el PLA (Ácido Poliláctico) como material base. Se trata del polímero más común usado en la impresión en 3D.

---

## 3. Condiciones particulares y fabricación

La impresión 3D podrá realizarse tanto en impresores profesionales como caseras, puesto que únicamente se necesitan el archivo de extensión .STL que nos viene dado desde el programa de diseño CAD SolidWorks.

### **3.3. Montaje de los componentes del Tag**

A continuación se describe el proceso a seguir para realizar un correcto montaje de los componentes dentro del Tag. Este montaje se llevará a cabo por personal cualificado en un entorno seguro y evitando descargas electrostáticas sobre los circuitos integrados que puedan dañar su integridad.

Para comenzar se tendrán que soldar los componentes electrónicos mediante el uso de la aleación de estaño y plomo al 60% - 40% respectivamente, así como las conexiones entre ellos con cables de cobre. Los componentes que se soldarán aquí son: el módulo LoPy4, los conectores, los diodos, los transistores, las resistencias y el buzzer.

A continuación se fijará el PCB a los cilindros que sobresalen del suelo de la caja con ayuda de silicona. También se usará esta silicona para fijar el soporte de la batería, el interruptor, y el cable de conexión de la antena en sus lugares correspondientes. También se fijará con silicona el motor vibrador al suelo de la caja para que transmita las vibraciones por todo el Tag.

Por último se usará silicona para sellar la caja con una lámina de metacrilato que permita ver a través de ella todos los componentes que componen el Tag. Se pasará la cinta de raso de 90 cm de longitud y se coserán los extremos de ella, quedando el tag colgado a una altura razonable para el cuello.

### **3.4. Componentes electrónicos**

Los componentes electrónicos a montar en el PCB de prototipado serán únicamente los especificados por el autor del proyecto, y descritos en las hojas de especificaciones correspondientes a cada uno de ellos, con el fin de evitar fallos en el funcionamiento.

Con el fin de ceder un poco de flexibilidad al fabricante, se podrán sustituir, con la aprobación del autor del proyecto o el ingeniero responsable, por componentes de características idénticas o muy similares.

Todos los componentes empleados deberán cumplir con la normativa de Restricción de Sustancias Peligrosas en aparatos eléctricos y electrónicos correspondiente a cada uno de ellos.



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

*Diseño y desarrollo de un sistema de  
monitorización y limitación en tiempo real del  
aforo en espacios cerrados*

## 4. Presupuesto

**Autor:**

D. Ferran Ballester Mainar

**Tutor:**

D. Ángel Perles Ivars

Alboraia, noviembre de 2020

# Índice

---

<b>1. Sobre el presupuesto .....</b>	<b>2</b>
<b>2. Materiales.....</b>	<b>2</b>
2.1. Componentes electrónicos.....	2
2.2. Impresión 3D de la carcasa .....	2
2.3. Materiales de fabricación y ensamblado .....	3
<b>3. Mano de obra.....</b>	<b>4</b>
<b>4. Resumen .....</b>	<b>5</b>

## 1. Sobre el presupuesto

---

El presupuesto del proyecto se ha realizado con la premisa de simular con un prototipo real el sistema final mediante el uso de dos Tags y tres beacons. Esto supone que todo el material presupuestado sea únicamente aplicable a este caso particular. Para cualquier otro caso, podría tener costes adicionales no contemplados aquí.

Los precios de los diferentes componentes electrónicos se han obtenido mediante distribuidores autorizados nacionales e internacionales.

## 2. Materiales

---

### 2.1. Componentes electrónicos

Ud	Descripción	Cantidad	Fabricante	Precio (€)	Total (€)
-	Batería 18650 Li-ion 2400 mAh 3,7 V	2	Sparkfun	2,98	5,96
-	Carcasa batería 18650	2	MDP	3,54	7,08
-	Interruptor de 3 pines	2	E-Switch	1,35	2,70
-	Zumbador magnético CEM-1203(42)	2	CUI INC	0,80	1,60
-	Motor vibrador 310-101	2	Precision Microdrivers	3,90	7,80
-	Módulo de desarrollo LoPy4	2	Pycom	33,06	66,12
-	Conector de cables con 2 pines	4	Sparkfun	0,95	3,80
-	Transistor BC547C	4	On Semiconductor	0,15	0,61
-	Resistencia 1 k $\Omega$	4	Multicomp Pro	0,11	0,42
-	Resistencia 39 $\Omega$	2	Multicomp Pro	0,06	0,13
-	Resistencia 60 $\Omega$	2	Multicomp Pro	0,11	0,22
-	Diodo 1N4001	4	American Power Devices	0,29	1,17
-	Antena Sigfox/LoRaWAN + Conector coaxial RF	2	Pycom	16,21	32,42
-	Beacon iBKS105	3	Accent Systems	16,66	50,00
m	Cable de cobre	0,03	Alpha Wire	0,47	0,01
Total					180,05

*Tabla 1. Gasto en componentes electrónicos.*

El precio final del conjunto de componentes electrónicos es de **180,05 €**.

### 2.2. Impresión 3D de la carcasa

Se ha utilizado la impresora TEVO Tornado, se trata de una impresora 3D FDM de deposición de material fundido. Para estimar el coste se ha tenido en cuenta la cantidad de material PLA en kg necesaria para crear las dos carcasas (62 gramos cada

una) y el gasto energético total consumido teniendo en cuenta la potencia consumida y las horas totales que ha estado en funcionamiento para cada una de las carcasas (7 horas por carcasa).

Ud	Descripción	Cantidad	Precio (€)	Total (€)
kg	PLA filamento negro 1,75mm 1,25g/cm3	0,124	14,44	1,79
h	Coste de las horas de impresión 3D	14	0,25	3,50
-	Coste de preparación y calibración	2	3,00	6,00
Total				11,29

*Tabla 2. Gasto en impresión 3D.*

El precio final del proceso de impresión 3D asciende a **11,29 €**.

### **2.3. Materiales de fabricación y ensamblado**

En este apartado se detalla el gasto en material adicional relacionado con la fabricación y ensamblaje de los dos Tags

Ud	Descripción	Cantidad	Precio (€)	Total (€)
-	Lámina de metacrilato transparente 2mm	1	8,05	8,05
g	Cola termofusible	3,5	0,04	0,14
g	Aleación estaño - plomo 60% - 40%	2	0,10	0,20
Total				8,39

*Tabla 3. Gasto en material adicional.*

El precio final de los materiales de fabricación y ensamblado asciende a la cantidad de **8,39 €**.

### 3. Mano de obra

La mano de obra empleada en el diseño, testeo y fabricación de este prototipo ha supuesto el mayor coste al proyecto. Esto se debe a que los materiales son muy económicos en comparación con las horas de trabajo invertidas.

El precio de la mano de obra empleada se va a estimar con el precio facturable por hora de un ingeniero técnico recién graduado en una ingeniería de la rama industrial. Se ha estimado un valor de 30€/h. Para las horas empleadas en tareas que no se necesita un grado de cualificación tan alto como puede ser el montaje y ensamblaje de las diferentes piezas, por lo que se supondrá un valor de 20€/h

Se ha de tener en cuenta todos los aspectos en los que se ha invertido tiempo durante la realización de este proyecto. Este tiempo total ha supuesto alrededor de 300 horas, correspondiente a los 12 ECTS que representan el trabajo de fin de grado en la Escuela Superior Técnica de Ingeniería del Diseño.

Ud	Descripción	Parcial	Cantidad	Precio (€)	Total (€)
h	Diseño del esquema electrónico en Proteus		30	30,00	900,00
h	Diseño mecánico de la carcasa		15	30,00	450,00
h	Diseño del software		110	30,00	3300,00
	Estudio previo	50			
	Desarrollo del código fuente	40			
	Testeo y depuración código final	20			
h	Ensamblaje del prototipo		6,75	20,00	135,00
	Montaje y soldadura manual PCB	4			
	Ensamblaje componentes en la carcasa	2			
	Inspección visual	0,25			
	Test de funcionamiento	0,5			
h	Redacción y generación de documentos		80	30,00	2400,00
	Redacción memoria de proyecto	50			
	Generación de planos	5			
	Generación de pliego de condiciones	10			
	Generación de presupuestos	10			
	Maquetación y presentación	5			
Total					7185,00

Tabla 4. Gasto en mano de obra y desarrollo del proyecto.

El precio final del conjunto de la mano de obra es de **7.185 €**.

## 4. Resumen

---

A continuación, en la tabla 5 se observa el resumen del presupuesto total que incluye el presupuesto de cada una de las secciones anteriores.

Apartado	Descripción	Parcial (€)	Total (€)
2.	Materiales		199,73
2.1.	Componentes electrónicos	180,05	
2.2.	Impresión 3D de la carcasa	11,29	
2.3.	Materiales de fabricación y ensamblado	8,39	
3.	Mano de obra		7185,00
Total			7384,73

I.V.A.	21%	1550,79
Porcentaje de beneficio	6%	443,08
<b>Coste total del proyecto</b>		<b>9378,60</b>

*Tabla 5. Resumen coste total del proyecto.*

La elaboración del proyecto tiene un coste total asociado de NUEVE MIL TRESCIENTOS SETENTA Y OCHO EUROS CON SESENTA CÉNTIMOS DE EURO (**9.378,60 €**).



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

*Diseño y desarrollo de un sistema de  
monitorización y limitación en tiempo real del  
aforo en espacios cerrados*

## **5. Anexo I**

**Autor:**

D. Ferran Ballester Mainar

**Tutor:**

D. Ángel Perles Ivars

**Alboraia, noviembre de 2020**

## Índice

---

1. Archivo boot.py .....	2
2. Archivo main.py – Tag 1.....	4
3. Archivo main.py – Tag 2.....	8

## 1. Archivo boot.py

---

```
from machine import Pin
import time
import pycom

motor = Pin('P22', mode = Pin.OUT)
motor.value(0)

buzzer = Pin('P23', mode = Pin.OUT)
buzzer.value(0)

pycom.heartbeat(False)

print("Iniciando Sistema")
motor.value(1)
buzzer.value(1)

pycom.rgbled(0xFFFFFF)
time.sleep(0.1)

pycom.rgbled(0xEEEEEE)
time.sleep(0.1)
buzzer.value(0)

pycom.rgbled(0xDDDDDD)
time.sleep(0.1)

pycom.rgbled(0xCCCCCC)
time.sleep(0.1)
buzzer.value(1)

pycom.rgbled(0xBBBBBB)
time.sleep(0.1)

pycom.rgbled(0xAAAAAA)
time.sleep(0.1)
buzzer.value(0)

pycom.rgbled(0x999999)
time.sleep(0.1)

pycom.rgbled(0x888888)
time.sleep(0.1)
buzzer.value(1)
```

```
pycom.rgbled(0x777777)
time.sleep(0.1)

pycom.rgbled(0x666666)
time.sleep(0.1)
buzzer.value(0)

pycom.rgbled(0x555555)
time.sleep(0.1)

pycom.rgbled(0x444444)
time.sleep(0.1)
buzzer.value(1)

pycom.rgbled(0x333333)
time.sleep(0.1)

pycom.rgbled(0x222222)
time.sleep(0.1)
buzzer.value(0)

pycom.rgbled(0x111111)
time.sleep(0.1)

pycom.rgbled(0x000000)
time.sleep(0.5)

pycom.rgbled(0x110000)
time.sleep(0.2)

pycom.rgbled(0x111100)
time.sleep(0.2)

pycom.rgbled(0x001100)
time.sleep(0.2)

pycom.rgbled(0x001111)
time.sleep(0.2)

pycom.rgbled(0x000011)
time.sleep(0.2)

pycom.rgbled(0x110011)
time.sleep(0.2)

pycom.rgbled(0x111111)
```

---

0.

```
buzzer.value(1)
time.sleep(0.8)

motor.value(0)
buzzer.value(0)
pycom.rgbled(0x000000)
time.sleep(0.5)
```

## 2. Archivo main.py – Tag 1

---

```
#importamos las librerías necesarias

from network import Bluetooth
from network import LoRa

from machine import Pin

import socket

import time

import pycom

import ubinascii

#definimos la función que calcula la distancia a la que se encuentra el emisor
#de una señal recibida
def calc_dist(rssi):
    n = 4 #valor del parámetro n, idealmente 2, en el mundo real entre 2 y 6
    return 10**((-60 -rssi)/(10*n))

#definimos la función que calcula la posición del Tag
def position(dist_b1, dist_b2, dist_b3):
    #inicializamos las posiciones de los beacons
    #beacon 1
    x1 = 0
    y1 = 0
    #beacon 2
    x2 = 5
    y2 = 0
    #beacon3
    x3 = 2.5
    y3 = 8
```

```

A = -2 * x1 + 2 * x2
B = -2 * y1 + 2 * y2
C = (dist_b1)**2 - (dist_b2)**2 - (x1)**2 + (x2)**2 - (y1)**2 + (y2)**2
D = -2 * x2 + 2 * x3
E = -2 * y2 + 2 * y3
F = (dist_b2)**2 - (dist_b3)**2 - (x2)**2 + (x3)**2 - (y2)**2 + (y3)**2

calculated_x = ((C*E)-(F*B))/((E*A)-(B*D))
calculated_y = ((C*D)-(A*F))/((B*D)-(A*E))

return calculated_x, calculated_y

#definimos el pin 22 para el motor vibrador como pin de salida
motor = Pin('P22', mode = Pin.OUT)
#inicializamos el motor apagado
motor.value(0)

#definimos el pin 23 para el buzzer como pin de salida
buzzer = Pin('P23', mode = Pin.OUT)
#inicializamos el buzzer apagado
buzzer.value(0)

#configuramos el objeto LoRa
lora = LoRa(mode = LoRa.LORAWAN, region = LoRa.EU868)

#Configuramos los parámetros LoRa
lora.init(mode = LoRa.LORAWAN, region = LoRa.EU868, power_mode = LoRa.TX_ONLY
, adr = False, public = True, tx_retries = 2, device_class = LoRa.CLASS_A)

#definimos los parámetros, dados por el servidor de LoRa, necesarios para con
ectarnos
dev_addr = ubinascii.unhexlify('26013A06')

nwk_swkey = ubinascii.unhexlify('DD0C1B1760D0527D02CC227CA32992FA')

app_swkey = ubinascii.unhexlify('5CF472BA6AF892422E88476B499F99EE')

#Creamos el objeto Bluetooth
bt = Bluetooth()

#configuramos la señal advertisement que será emitida
bt.set_advertisement(name = 'Tag')

#comenzamos a emitir advertisements
bt.advertise(True)

```

---

0.

```

#definimos la máxima señal rssi para cumplir con la distancia de seguridad
rssi_max = -57

#inicializamos las variables received signal strength indicator de los beacons a -10 puesto que es un valor no realista
#posteriormente se actualizarán
rssi1 = -10
rssi2 = -10
rssi3 = -10

#escaneamos señales Bluetooth indefinidamente
bt.start_scan(-1)

while True:

    #cálculo de la posición
    if rssi1 != -10 and rssi2 != -10 and rssi3 != -10: #si ya se han actualizado las variables rssi de los 3 beacons

        dist1 = calc_dist(rssi1)
        dist2 = calc_dist(rssi2)
        dist3 = calc_dist(rssi3)

        x, y = position(dist1, dist2, dist3)

    #apagamos motor y buzzer
    motor.value(0)
    buzzer.value(0)

    #buscamos señales advertising
    adv = bt.get_adv()

    if adv:

        #señal advertisement beacon 1
        if bt.resolve_adv_data(adv.data, Bluetooth.ADV_NAME_CMPL) == 'iBKS105_1':

            pycom.rgbled(0x0000FF) #encendemos led azul

            rssi1 = adv.rssi #almacenamos el valor rssi del beacon 1

            #señal advertisement beacon 2
        elif bt.resolve_adv_data(adv.data, Bluetooth.ADV_NAME_CMPL) == 'iBKS105_2':

            pycom.rgbled(0x00FF00) #encendemos led verde

            rssi2 = adv.rssi #almacenamos el valor rssi del beacon 2

```

```

#señal advertisement beacon 3
elif bt.resolve_adv_data(adv.data, Bluetooth.ADV_NAME_CMPL) == 'iBKS105_3
':

    pycom.rgbled(0xFFFF00) #encendemos led amarillo

    rssi3 = adv.rssi        #almacenamos el valor rssi del beacon 3

#señal advertisement de otro Tag
elif bt.resolve_adv_data(adv.data, Bluetooth.ADV_NAME_CMPL) == 'Tag':

    rssi_tag = adv.rssi    #almacenamos el valor rssi del Tag

#comprobamos si el valor de la señal rssi supera a la máxima permitida
if rssi_tag >= rssi_max:

    pycom.rgbled(0xFF0000) #encendemos led rojo

    #encendemos motor y buzzer
    motor.value(1)
    buzzer.value(1)

    #nos conectamos con el servidor LoRa
    lora.join(activation = LoRa.ABP, auth = (dev_addr, nwk_swkey, app_swk
ey))

    if lora.has_joined():

        s = socket.socket(socket.AF_LORA, socket.SOCK_RAW)
        s.setsockopt(socket.SOL_LORA, socket.SO_DR, 5)
        s.setblocking(True)
        s.send("ubicación: x:" + str(x) + " y:" + str(y)) #enviamos la ub
icación en coordenadas x e y
        s.setblocking(False)

        #print('Connected')
        #print('x: ' + str(x) + ', y: ' + str(y)) #enviamos por
el puerto serie para la prueba 2.

```

### 3. Archivo main.py – Tag 2

---

```
#importamos las librerías necesarias

from network import Bluetooth
from network import LoRa

from machine import Pin

import socket

import time

import pycom

import ubinascii

#definimos la función que calcula la distancia a la que se encuentra el emisor de una señal recibida
def calc_dist(rssi):
    n = 4 #valor del parámetro n, idealmente 2, en el mundo real entre 2 y 6
    return 10**((-60 -rssi)/(10*n))

#definimos la función que calcula la posición del Tag
def position(dist_b1, dist_b2, dist_b3):
    #inicializamos las posiciones de los beacons
    #beacon 1
    x1 = 0
    y1 = 0
    #beacon 2
    x2 = 5
    y2 = 0
    #beacon3
    x3 = 2.5
    y3 = 8

    A = -2 * x1 + 2 * x2
    B = -2 * y1 + 2 * y2
    C = (dist_b1)**2 - (dist_b2)**2 - (x1)**2 + (x2)**2 - (y1)**2 + (y2)**2
    D = -2 * x2 + 2 * x3
    E = -2 * y2 + 2 * y3
    F = (dist_b2)**2 - (dist_b3)**2 - (x2)**2 + (x3)**2 - (y2)**2 + (y3)**2

    calculated_x = ((C*E)-(F*B))/((E*A)-(B*D))
    calculated_y = ((C*D)-(A*F))/((B*D)-(A*E))

    return calculated_x, calculated_y
```

```

#definimos el pin 22 para el motor vibrador como pin de salida
motor = Pin('P22', mode = Pin.OUT)
#inicializamos el motor apagado
motor.value(0)

#definimos el pin 23 para el buzzer como pin de salida
buzzer = Pin('P23', mode = Pin.OUT)
#inicializamos el buzzer apagado
buzzer.value(0)

#configuramos el objeto LoRa
lora = LoRa(mode = LoRa.LORAWAN, region = LoRa.EU868)

#Configuramos los parámetros LoRa
lora.init(mode = LoRa.LORAWAN, region = LoRa.EU868, power_mode = LoRa.TX_ONLY
, adr = False, public = True, tx_retries = 2, device_class = LoRa.CLASS_A)

#definimos los parámetros, dados por el servidor de LoRa, necesarios para con
ectarnos
dev_addr = ubinascii.unhexlify('2601372C')

nwk_swkey = ubinascii.unhexlify('20335FCD163F815E9C5C81852EC7C17A')

app_swkey = ubinascii.unhexlify('63606FDC3D19C99AB44D6F08486FC698')

#Creamos el objeto Bluetooth
bt = Bluetooth()

#configuramos la señal advertisement que será emitida
bt.set_advertisement(name = 'Tag')

#comenzamos a emitir advertisements
bt.advertise(True)

#definimos la máxima señal rssi para cumplir con la distancia de seguridad
rssi_max = -57

#inicializamos las variables received signal strength indicator de los beacons
a -10 puesto que es un valor no realista
#posteriormente se actualizarán
rssi1 = -10
rssi2 = -10
rssi3 = -10

#escaneamos señales Bluetooth indefinidamente

```

---

0.

```

bt.start_scan(-1)

while True:

    #cálculo de la posición
    if rssi1 != -10 and rssi2 != -10 and rssi3 != - 10: #si ya se han
actualizado las variables rssi de los 3 beacons

        dist1 = calc_dist(rssi1)
        dist2 = calc_dist(rssi2)
        dist3 = calc_dist(rssi3)

        x, y = position(dist1, dist2, dist3)

    #apagamos motor y buzzer
    motor.value(0)
    buzzer.value(0)

    #buscamos señales advertising
    adv = bt.get_adv()

    if adv:

        #señal advertisement beacon 1
        if bt.resolve_adv_data(adv.data, Bluetooth.ADV_NAME_CMPL) == 'iBKS105_1':

            pycom.rgbled(0x0000FF) #encendemos led azul

            rssi1 = adv.rssi          #almacenamos el valor rssi del beacon 1

            #señal advertisement beacon 2
        elif bt.resolve_adv_data(adv.data, Bluetooth.ADV_NAME_CMPL) == 'iBKS105_2
':

            pycom.rgbled(0x00FF00) #encendemos led verde

            rssi2 = adv.rssi          #almacenamos el valor rssi del beacon 2

            #señal advertisement beacon 3
        elif bt.resolve_adv_data(adv.data, Bluetooth.ADV_NAME_CMPL) == 'iBKS105_3
':

            pycom.rgbled(0xFFFF00) #encendemos led amarillo

            rssi3 = adv.rssi          #almacenamos el valor rssi del beacon 3

            #señal advertisement de otro Tag
        elif bt.resolve_adv_data(adv.data, Bluetooth.ADV_NAME_CMPL) == 'Tag':

```

```

rssi_tag = adv.rssi      #almacenamos el valor rssi del Tag

#comprobamos si el valor de la señal rssi supera a la máxima permitida
if rssi_tag >= rssi_max:

    pycom.rgbled(0xFF0000) #encendemos led rojo

    #encendemos motor y buzzer
    motor.value(1)
    buzzer.value(1)

    #nos conectamos con el servidor LoRa
    lora.join(activation = LoRa.ABP, auth = (dev_addr, nwk_swkey, app_swk
ey))

    if lora.has_joined():

        s = socket.socket(socket.AF_LORA, socket.SOCK_RAW)
        s.setsockopt(socket.SOL_LORA, socket.SO_DR, 5)
        s.setblocking(True)
        s.send("ubicación: x:" + str(x) + " y:" + str(y)) #enviamos la ub
icación en coordenadas x e y
        s.setblocking(False)

        #print('Connected')
        #print('x: ' + str(x) + ', y: ' + str(y))           #enviamos por
el puerto serie para la prueba 2.

```