

Diseño, análisis y construcción de la carga útil y mejora de un robot de inventario RFID basado en estigmergia.

Hernández Melià, Lluís

Curs 2019-2020

Director: VÍCTOR CASAMAYOR, RAFAEL POUS

GRAU EN ENGINYERIA INFORMÀTICA



**Universitat
Pompeu Fabra
Barcelona**

**Escola
Superior Politècnica**

Treball de Fi de Grau

DISEÑO, ANÁLISIS Y CONSTRUCCIÓN DE LA CARGA ÚTIL Y MEJORA DE UN ROBOT DE INVENTARIO RFID BASADO EN ESTIGMERGIA.

TREBALL FI DE GRAU DE
Lluís Hernández Melià

Director: Victor Casamayor, Rafael Pous

Grau en Enginyeria en Informàtica

Curs 2019-2020



Universitat
Pompeu Fabra
Barcelona

Escola
d'Enginyeria

Agradecimientos

En primer lugar, quisiera agradecer a mi tutor, Víctor Casamayor, la ayuda que me ha prestado en la realización del trabajo, así como su incondicional paciencia y sus enseñamientos durante este año.

A Rafael Pous, tutor también del trabajo, por brindarme la oportunidad de trabajar en un proyecto como este y aprender de él. También a José Luis Sanz, diseñador industrial de Keonn, por su trabajo y horas de desarrollo en la creación de los diseños en 3D, así como su experiencia y ayuda en los aspectos de diseño del robot. Al equipo de investigación de la Universidad, por su ayuda, horas de compañía, y apoyo durante el proyecto. A todo el equipo de producción de Keonn, por la ayuda en el montaje del prototipo final.

Finalmente, a todos mis amigos y a mi familia, por su apoyo absoluto durante toda la carrera. En especial a mis padres, sin los cuales nada de esto podría haber sido posible. Gracias.

Resumen

El objetivo del trabajo es el análisis y mejora de diseño de un robot RFID de inventario para *retail* basado en estigmergia. Se evalúan cuatro configuraciones para la carga útil del robot, tratando de encontrar la mejor distribución posible, minimizando la altura y optimizando la capacidad de lectura. Se llega a la conclusión de mantener una distribución cuadrada de las antenas y disminuir la altura del robot a 60 cm. Asimismo, se lleva a cabo un estudio sobre la batería del robot, suprimiendo una fuente de alimentación externa y caracterizando la batería integrada de la base robótica. También se optimiza la distribución de los componentes internos, analizando la mejor disposición y teniendo en cuenta la modularidad del sistema. Una vez desarrollados los aspectos mencionados, se construye un nuevo prototipo acorde a estos. Además, se lleva a cabo un estudio sobre el diseño exterior del robot y la posibilidad de crear una carcasa que proteja los componentes internos. Finalmente, se contempla la inclusión de un sistema de notificaciones a través de leds, que se integrarían en la carcasa, y se lleva a cabo el desarrollo parcial de éste, implementando unos leds para la monitorización de la batería.

Resum

L'objectiu del treball és l'anàlisi i millora de disseny d'un robot RFID d'inventari per a *retail* basat en estigmergia. S'avaluen quatre configuracions per a la càrrega útil del robot, tractant de trobar la millor distribució possible, minimitzant l'altura i optimitzant la capacitat de lectura. S'arriba a la conclusió de mantenir una distribució quadrada de les antenes i disminuir l'altura del robot a 60 cm. Així mateix, es duu a terme un estudi sobre la bateria del robot, suprimint una font d'alimentació externa i caracteritzant la bateria integrada de la base robòtica. També s'optimitza la distribució dels components interns, analitzant la millor disposició tenint en compte la modularitat del sistema. Una vegada desenvolupats tots els aspectes mencionats, es construeix un nou prototip acord a aquests. A més, es duu a terme un estudi sobre el disseny exterior del robot i la possibilitat de crear una carcassa que protegeixi els components interns. Finalment, es contempla la inclusió d'un sistema de notificacions a través de leds, que s'integrarien en la carcassa, i es duu a terme el desenvolupament parcial d'aquests, implementant uns leds per la monitorització de la bateria.

Abstract

The objective of this project is to analyze and improve the design of an RFID robot based on stigmergy for retail inventory. Four configurations are evaluated for the robot's payload, trying to find the best possible distribution, minimizing the height and optimizing the reading capacity. The conclusion is to maintain a square distribution of the antennas and decrease the height of the robot to 60 cm. Likewise, a study is carried out on the robot's battery, removing an external power supply and characterizing the integrated battery of the robotic base. The distribution of the internal components is also optimized, analyzing the best arrangement and taking into account the modularity of the system. Once the mentioned aspects have been developed, a new prototype is built according to these. In addition, a study is carried out for the exterior design of the robot and for the possibility of creating a case that protects the internal components. Finally, the inclusion of a notification system through LEDs is contemplated, which would be integrated into the casing, and its partial development is carried out, implementing some LEDs for battery monitoring.

Índice

Lista de Figuras	9
1. INTRODUCCIÓN.....	11
1.1 Motivación	11
1.2 Estado de la técnica.....	12
a) Metodología de diseño	12
b) Tecnología RFID	12
c) ROS Software.....	16
d) Robots RFID de inventario	17
e) Prototipo Preliminar	19
1.3 Hipótesis y propuestas de mejora	24
2. DISEÑO Y PROTOTIPADO DE SUBSISTEMAS .. ¡Error! Marcador no definido.	
2.1 Disposición y altura de las antenas	25
a) Hipótesis y propuestas.....	25
b) Desarrollo de la solución	29
c) Testeado y análisis	30
2.2 Adaptación de la fuente de alimentación	50
a) Hipótesis y propuesta	50
b) Desarrollo de la solución	50
c) Testeado y análisis	54
2.3 Distribución interna de componentes	57
a) Requerimientos	57
b) Modularidad.....	58
2.4 Construcción de un nuevo prototipo	59
a) Características	59
b) Construcción del Prototipo	60
2.5 Presupuesto	63
3. DISEÑO EXTERIOR E INTERFAZ	67
3.1 Diseño exterior (<i>look and feel</i>)	67
a) Propuesta de carcasa y requerimientos	67
b) Estudio de estilo.....	68
3.2 Human-Robot Interface	70
a) Sistema de notificaciones	71
4. CONCLUSIONES.....	79
4.2 Trabajo futuro	80

Bibliografia.....	81
Anexos	85

Lista de Figuras

Fig. 1: reader [6]	13
Fig. 2: tags RFID [7]	13
Fig. 3: antena [6].....	13
Fig. 4: ejemplo de sistema RFID [8].	13
Fig. 5: ejemplo de estructura del archivo .csv.	15
Fig. 6: vista frontal 2D de la antena SP11 [6], [10].....	15
Fig. 7: radiación en 3D del rango de la antena [6], [10].....	15
Fig. 8: radiación en 2D de la antena [6], [10].....	15
Fig. 9: ejemplificación de un sistema de ROS.....	16
Fig. 10: Tally [14].....	17
Fig. 11: Stockbot [15].....	17
Fig. 12: Tory [16], [17].....	17
Fig. 13: Advanrobot-100 [18].....	17
Fig. 14: esquema del diseño preliminar con componentes numerados [19], [20], [21] .	20
Fig. 15: robot preliminar.....	21
Fig. 16: esquema de conexiones del robot preliminar.	22
Fig. 17: rango aproximado del robot actual.....	23
Fig. 18: ejemplificación del algoritmo que sigue el robot para la navegación.....	23
Fig. 19: representación de la lectura del robot en un pasillo.	26
Fig. 20: modelo 3D de la configuración Inclínada.	26
Fig. 21: modelo 3D de la configuración Cuadrada.....	27
Fig. 22: modelo 3D de la configuración Romboidal.	27
Fig. 23: modelo 3D de 2 niveles. Antenas frontales inclinadas y laterales verticales....	28
Fig. 24: modelo 3D de 2 niveles. Antenas frontales y laterales inclinadas.	28
Fig. 25: estructura con medidas para la configuración Cuadrada y Romboidal.....	29
Fig. 26: estructura con medidas para la configuración Inclínada.	29
Fig. 27: prototipo construido de la configuración Inclínada.	30
Fig. 28: prototipo construido de la configuración Cuadrada/Romboidal.	30
Fig. 29: renderizado 3D del entorno con el recorrido del robot. Vista superior.....	31
Fig. 30: renderizado 3D del entorno, vista isométrica.....	31
Fig. 31: recorrido del robot señalado en una imagen real.	31
Fig. 32: mapa del análisis 3.1. Leyenda en el recuadro inferior derecho.	39
Fig. 33: comparativa entre la configuración Cuadrada y la Inclínada.	40
Fig. 34: esquema de la caja de tags.	42
Fig. 35: configuración Cuadrada corriendo la prueba de altura a nivel de suelo.	42
Fig. 36: output del script.....	43
Fig. 37: esquema de configuraciones destacadas en cada prueba.	48
Fig. 38: esquema altura - calidad de lecturas.....	49
Fig. 39: conexiones disponibles que proporciona el Kobuki Base.....	51
Fig. 40: extracto del Datasheet del láser [26].	51
Fig. 41: propuesta del nuevo esquema de conexiones.....	52
Fig. 42: conversor DCDC 12 V a 5 V 3 A [27].....	53
Fig. 43: conversor DCDC 12 V a 24 V [28].....	53
Fig. 44: cable USB a micro USB [29].	53

Fig. 45: adaptador Molex [30].	53
Fig. 46: adaptador Molex [31].	53
Fig. 47: adaptación de un cable Molex. Pasos 1), 2) y 3).	53
Fig. 48: adaptación de los componentes con conexiones Molex. Pasos 4), 5) y 6).	54
Fig. 49: output de la llamada de status a la Kobuki Base.	55
Fig. 50: estructura de módulos, con las salidas de cada uno.	59
Fig. 51: propuesta de prototipo.	60
Fig. 52: perspectiva lateral en 2D del diseño definitivo, con detalles de medidas.	61
Fig. 53: renderizado 3D del diseño final. Vista isométrica.	61
Fig. 54: explosionado 3D del prototipo final.	62
Fig. 55: especificación de piezas del explosionado.	62
Fig. 56: vista lateral del prototipo.	63
Fig. 57: vista frontal del prototipo.	63
Fig. 58. propuesta de prototipo con carcasa	68
Fig. 59: lámpara minimalista con asa, diseño de Miguel Milá [36]	69
Fig. 60: Mac Pro 2013 [37].	69
Fig. 61: estufa Cecotec [38].	69
Fig. 62: carrito de compra [39].	69
Fig. 63: auriculares inalámbricos GT1 [40].	70
Fig. 64: Amazon echo [41].	70
Fig. 65: tipología para las notificaciones traseras.	73
Fig. 66: disposición de notificaciones usuales.	73
Fig. 67: esquema de pines de la Raspberry [45].	74
Fig. 68: conexiones de tierra, datos y 5V respectivamente de la tira de leds [44].	74
Fig. 69: esquema de relaciones entre nodos del gestor de Batería.	75
Fig. 70: esquema del script “node_BatteryManager”.	77
Fig. 71: esquema del script “node_LedsController”.	78

1. INTRODUCCIÓN

Este primer capítulo trata la motivación del proyecto, así como los aspectos generales del estado de la técnica. Finalmente se introducen las hipótesis y propuestas desarrolladas en el trabajo.

1.1 Motivación

En este proyecto se debe elaborar la mejora de diseño y prototipado de un robot de inventario RFID basado en estigmergia, cuyo objetivo es diferenciarse del resto de robots de RFID del mercado.

El diseño es la planificación y creación de un nuevo producto o sistema para solucionar un problema específico, con la intención de mejorar la experiencia humana [1]. Como dijo Steve Jobs:

“design is not just what it looks like and feels like. Design is how it works.”¹

Elaborar un buen diseño es importante, tanto a nivel funcional como a nivel visual:

A nivel funcional, lo principal es que el sistema cumpla su cometido. Un buen diseño es necesario para asegurar la robustez y estabilidad del prototipo, hacerlo manejable, transportable y facilitar un futuro mantenimiento. También puede evitar costes innecesarios, optimizando los componentes, la distancia entre ellos, y prescindiendo de algunos no necesarios. Al mismo tiempo, también es importante contemplar la modularidad del sistema.

Por otro lado, a nivel visual, es necesario obtener un producto que se diferencie de la competencia. Un producto que destaque frente a otros y atraiga la curiosidad, siempre será más dado a ser vendido [2]. Se debe obtener un producto visualmente lógico, bien construido y de calidad. Asimismo, y combinado también con el aspecto funcional, cabe destacar la importancia de la interacción entre el usuario y el robot, que debe ser intuitiva y sencilla.

Se elaborará un análisis extenso del robot y sus necesidades, y se propondrán una serie de mejoras. Se redefinirá el diseño del robot, llegando a un nuevo prototipo que trate de mejorar aspectos como su estabilidad, su distribución de componentes o su interfaz, que son claves para el óptimo funcionamiento en un prototipo definitivo.

¹ Del artículo “The Guts of a New Machine”, 2003, New York Times

1.2 Estado de la técnica

a) Metodología de diseño

La metodología que se lleva a cabo en los diferentes análisis de este proyecto se basa en los pasos estándar para el proceso de diseño de ingeniería [3], [4]. Éstos son:

1. **Definición del problema o necesidad:** Se define concretamente el objetivo del proceso.
2. **Investigación de fondo (*background research*):** se investiga el entorno del proyecto (e.g. el mercado actual de robots RFID) y se analiza si hay alguno con los mismos objetivos.
3. **Especificar los requerimientos:** se analizan los pasos que deben llevarse a cabo para solucionar la necesidad.
4. **Brainstorm, evaluar y elegir una solución:** Se lanzan una serie de ideas y propuestas, con sus pros y contras, de las cuales se elegirán una o unas pocas a llevar a cabo.
5. **Prototipar y desarrollar la Solución:** Se lleva a cabo el desarrollo de la idea o ideas que se hayan decidido en la fase anterior.
6. **Testear la solución:** Se testea la idea llevada a cabo. Si los test cumplen las expectativas, se finaliza con la iteración. En caso contrario, se retorna al punto 4 para testear otras soluciones o modificaciones de la solución que no ha dado el resultado esperado.

Esta metodología es iterativa, por tanto, el ciclo se repite hasta alcanzar un resultado final. Asimismo, la metodología se lleva a cabo tanto en el hilo principal del proyecto como en los hilos secundarios, como se ve más adelante en el proceso de elegir una disposición de las antenas (pág. 25), o de suprimir la batería externa del robot (pág. 50).

b) Tecnología RFID

La tecnología RFID (identificación por Radio Frecuencia), es un sistema de identificación sin contacto. Este sistema es una alternativa a otros medios de identificación como los códigos de barras [5]. En concreto la variante de RFID que se usa en el proyecto es la UHF (*ultra high frequency*), que tiene las siguientes características:

- Usa tecnología de radio.
- Su ancho de banda es de unos 900MHz (860-960MHz).
- Permite la comunicación a larga distancia (5-6 metros) sin necesidad de tener visión entre el punto de emisión y recepción de la onda.

La tecnología RFID se usa en varios ámbitos. Uno de los sectores donde la RFID ha arraigado más es en el sector del *retail* (venta al por menor), ya que permite llevar a cabo diversos procesos (e.g. inventario de una tienda) de manera más rápida y precisa que otras tecnologías tradicionales.

El sistema RFID se compone de cuatro componentes básicos: un *reader*², una antena, un tag y un sistema de información. El *reader* (Fig. 1), emite una radiofrecuencia a través de una antena (Fig. 3). Se dice que el *reader* interroga a los tags. Esta radiofrecuencia es recibida por los tags RFID (Fig. 2), unas etiquetas con un chip pasivo integrado, que al recibir la radiofrecuencia (se dice que el tag es iluminado por el *reader*) responden al *reader* enviando el código único que identifica ese tag. Los tags utilizan la potencia de la onda emisora para retornar la señal. Cuando el *reader* recibe la información de los tags que ha irradiado, la envía a un sistema de información. Todo este proceso se puede observar en la Fig. 4 de manera esquematizada.

El robot usa la tecnología RFID como medio para llevar a cabo un inventario rápido y preciso. Aunque no sea necesaria la visión directa entre emisor y receptor para localizar los objetos, es necesario que cada producto tenga un tag asociado adherido para poder llevar a cabo el inventario de manera autónoma.



Fig. 1: reader [6]

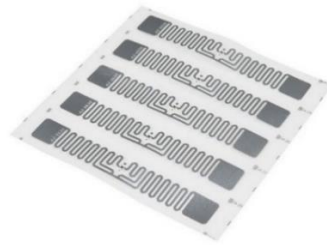


Fig. 2: tags RFID [7]



Fig. 3: antena [6]

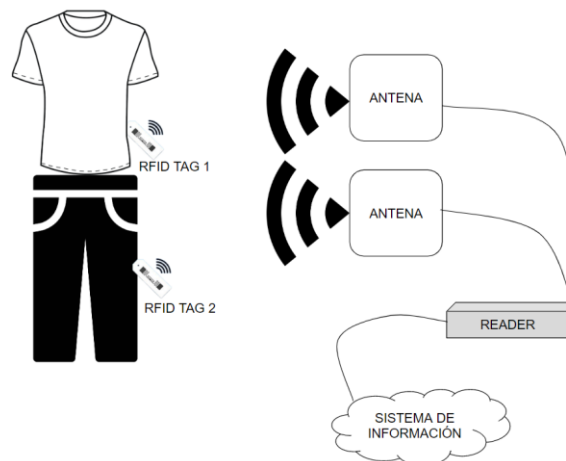


Fig. 4: ejemplo de sistema RFID [8].

En la Fig. 1 se observa un *reader*; en la Fig. 2, cuatro tags RFID; en la Fig. 3, una antena RFID. En la última figura (Fig. 4) se percibe la siguiente situación: el *reader* interroga con las 2 antenas. Al irradiarse, los dos tags responden con su ID propio, que recibe el *reader* y manda la información al sistema de información.

² Llamado *reader* o lector.

Interferencias en la RFID

Debe tenerse en cuenta que la RFID es una tecnología que se ve afectada por su entorno en un dominio real [9]. Hay diversos elementos que pueden provocar interferencias y evitar que se lean los tags, por ejemplo:

- Interferencias debidas a ondas de radiofrecuencia. La fusión de dos ondas en contrafase produce que se cancelan ambas, provocando que la onda dirigida a leer el tag se anule. A este tipo de interferencia se le llama interferencia destructiva. Por tanto, otros dispositivos RFID emitiendo radiofrecuencias o incluso las propias ondas emitidas por el dispositivo pueden provocar interferencias.
- Rebotes que tenga la onda en el escenario. Las ondas rebotan completamente en el metal, que tiene un coeficiente de reflexión de 1, y se amortiguan más en otros materiales, como fluidos o cajas de cartón, que causan atenuación en la señal.

Este tipo de interferencias no pueden ser evitadas, pero se suelen tomar otras medidas para asegurar el mejor funcionamiento posible del sistema. Algunas de estas son:

- El aislamiento físico del *reader*: se aísla el *reader* en una jaula de Faraday, que hace que el campo electromagnético en el interior de ésta se mantenga estable, y de esta manera evitar corrientes internas inducidas.
- Apantallamiento o protección de las antenas, dejando un espacio de 1cm aproximadamente entre la antena y una placa de metal que apantalla, que permiten hacer más efectivo el rango de la antena.

Sistema utilizado

El sistema RFID que se usa en el proyecto es el compuesto por el AdvanReader 150.03 (Fig. 1) y cuatro Advantennas SP11 (Fig. 3), ambos productos de Keonn. También se usa una Raspberry Pi como sistema de información y tags RFID.

El AdvanReader y las Advantennas se conectan y se llevan a cabo las lecturas de tags (proceso de inventario). A continuación, el AdvanReader se comunica con la Raspberry a través de una API REST (*Representational State Transfer*), controlada por un software de bajo nivel, llamado Advannet. Éste almacena toda la información sobre las lecturas en un archivo .csv (*comma-separated values*), que posteriormente se envía a la Raspberry Pi. En la Fig. 5 se puede observar un ejemplo de un archivo .csv generado por el *reader* y enviado a la Raspberry.

	A	B	C	D	E	F	G	H	I	J
1	EVENT_TYPE	EVENT_DETAILS	HEX_EPC	DECODED_EPC	DEVICE_ID	DEVICE_IP	ANTENNA_PORT	OPERATION_MODE	TIME_STAMP	RSSI
2	TAG_READ	e28011700000020910000023	,reader-01	192.168.15.101	1	2016/02/11 17:32:07.030	-75			
3	TAG_READ	e28011700000020910000016	,reader-01	192.168.15.101	2	2016/02/11 17:32:07.284	-70			
4	TAG_READ	e28011700000020910000006	,reader-01	192.168.15.101	1	2016/02/11 17:32:07.107	-77			
5	TAG_READ	e28011700000020910000066	,reader-01	192.168.15.101	2	2016/02/11 17:32:07.277	-67			
6	TAG_READ	e28011700000020910000011	,reader-01	192.168.15.101	2	2016/02/11 17:32:07.292	-63			
7	TAG_READ	e28011700000020910000001	,reader-01	192.168.15.101	1	2016/02/11 17:32:07.163	-74			
8	TAG_READ	e28011700000020910000068	,reader-01	192.168.15.101	1	2016/02/11 17:32:07.082	-73			
9	TAG_READ	e28011700000020910000046	,reader-01	192.168.15.101	1	2016/02/11 17:32:07.090	-71			
10	TAG_READ	e28011700000020910000097	,reader-01	192.168.15.101	1	2016/02/11 17:32:07.024	-68			
11	TAG_READ	e28011700000020910000013	,reader-01	192.168.15.101	2	2016/02/11 17:32:07.269	-66			
12	TAG_READ	e28011700000020910000003	,reader-01	192.168.15.101	2	2016/02/11 17:32:07.243	-63			
13	TAG_READ	e28011700000020910000058	,reader-01	192.168.15.101	2	2016/02/11 17:32:07.273	-62			
14	TAG_READ	e28011700000020910000014	,reader-01	192.168.15.101	2	2016/02/11 17:32:07.287	-66			
15	TAG_READ	e28011700000020910000073	,reader-01	192.168.15.101	1	2016/02/11 17:32:07.069	-75			
16	TAG_READ	e28011700000020910000051	,reader-01	192.168.15.101	2	2016/02/11 17:32:07.295	-62			
17	TAG_READ	e28011700000020910000041	,reader-01	192.168.15.101	2	2016/02/11 17:32:07.235	-57			

Fig. 5: ejemplo de estructura del archivo .csv.

El .csv recoge en cada línea una lectura de un tag. Se guardan varios parámetros. Aquellos más útiles en este proyecto son:

- *TIME_STAMP*: indica en que instante de tiempo se lee el tag.
- *HEX_EPC*: código único (ID) de cada tag, llamado “epc” (*electronic product code*).
- *ANTENNA_PORT*: indica por qué puerto del *reader* ha sido leído el tag.

En cuanto a las antenas SP11 de Keonn, constan de una amplitud de rayo de 70°, y una máxima distancia efectiva de 10m [10]. Su eje de radiación es el que se puede ver en las siguientes figuras:

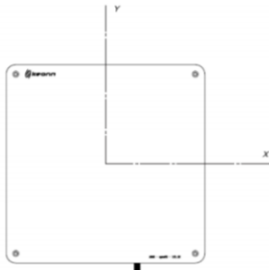


Fig. 6: vista frontal 2D de la antena SP11 [6], [10].

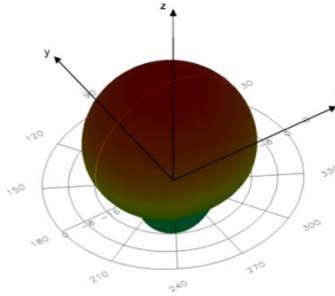


Fig. 7: radiación en 3D del rango de la antena [6], [10].

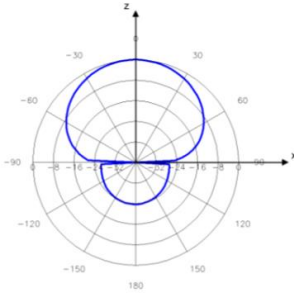


Fig. 8: radiación en 2D de la antena [6], [10].

En la Fig. 6, se observa la estructura de la antena en el eje x-y. En la Fig. 7, se observa el rango de radiofrecuencias que ofrece la antena, visualizado en 3D en el plano x-y-z. Finalmente, en la Fig. 8 se observa la vista superior en 2D en el eje x-z de la radiación de la antena.

c) ROS Software

ROS [11] es un *framework* open-source para programar robots. Es compatible con plataformas Unix y varios lenguajes de programación; para más detalles sobre ROS y sus componentes, ver [12]. Se compone de una estructura específica, en la cual se pueden encontrar varios componentes [13]. Algunos de ellos son:

- **Nodos:** son programas que ejecutan procesos. Cada nodo es independiente de otro, aunque son gestionados por un nodo principal, llamado ‘ROS Master’, que los registra y habilita. Los nodos se comunican entre ellos a través de *topics*, por los que circulan mensajes.
- **Topics:** son los canales por los que se comunican los nodos. Se puede considerar un puente entre dos nodos por el que circulan mensajes.
- **Publishers:** se encuentran dentro de los nodos, y son los encargados de publicar mensajes en los *topics*.
- **Subscribers:** de igual manera a los Publishers, se encuentran dentro de los nodos, y son los encargados de suscribirse a los *topics* para recibir los mensajes enviados por los *Publishers*.
- **Mensaje:** Los mensajes son valores (pueden ser varios tipos de datos, ya sean enteros, *Strings*, ...) que los nodos publican en los tópicos a través de los *Publishers*.

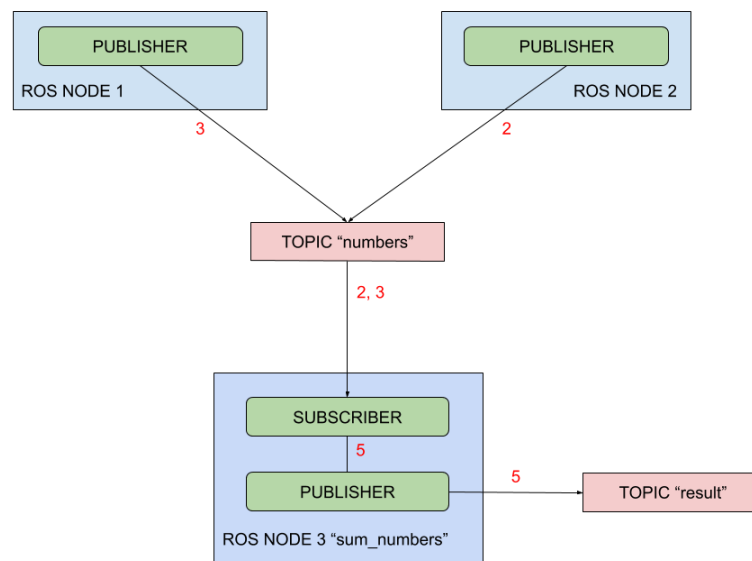


Fig. 9: ejemplificación de un sistema de ROS.

En la Fig. 9 se observa un ejemplo de una estructura de nodos de ROS. Los nodos 1 y 2 tienen un *Publisher* cada uno, y se encargan de publicar números en el tópico “numbers”. El nodo 3, llamado “sum-numbers”, contiene un *Publisher* y un *Subscriber* y se encarga de sumar los números recibidos por el *topic* con el *Subscriber* y publicarlos a otro *topic*, llamado “result”, con el *Publisher*.

En este proyecto se programa en Python. ROS se ejecuta dentro de la Raspberry para controlar todo el sistema del robot.

d) Robots RFID de inventario

Actualmente, en el mercado existen varios robots de inventario que usan la tecnología RFID. Algunos usan otras tecnologías útiles en el sector del *retail*, por ejemplo, cámaras 3D para detectar estanterías. Estos robots tienen grandes dimensiones, y están diseñados para leer la máxima magnitud de tags posible. Algunos de los modelos que se analizan son el robot Tally [14], Stockbot [15], Tory [16], [17] , o el AdvanRobot-100 [18], visibles en las figuras inferiores:



Fig. 10: Tally [14].



Fig. 11: Stockbot [15].



Fig. 12: Tory [16],
[17].



Fig. 13:
Advanrobot-100
[18].

- TALLY (Fig. 10), de Simbe Robotics [14], tiene un diámetro de base de 47cm, una altura de 162.5cm y un peso de 30 Kg. Se observa como en su base tiene un láser para detectar obstáculos. También tiene una pantalla frontal, 40 sensores para navegar, y cerca de una docena de cámaras para detectar estanterías. En cuanto a tecnología RFID, consta de una antena con un rango de 4m para leer a corta distancia. Tally requiere de un mapeado inicial del local para poder funcionar.
- El STOCKBOT (Fig. 11) de Pal Robotics [15] tiene un diámetro de base de 50cm y una altura de 190cm. Tiene 8 antenas RFID (4 en cada lateral) y un láser situado en la parte inferior. Tiene también algunas cámaras, aunque no se especifica su número ni otro tipo de sensores.
- TORY (Fig. 12), de MetraLabs [16], [17], tiene un diámetro de base de 50cm y una altura de 190cm. Funciona también con un sistema de mapeado del local, y está equipado con cámaras y antenas RFID, aunque no se especifica la cantidad. Tiene un peso de 60Kg.

- El ADVANROBOT-100 (Fig. 13) es la apuesta de Keonn [18] para competir con los robots anteriormente citados. Sus dimensiones son de 50cm de diámetro de base por 195cm de altura. Consta de 4 antenas RFID en cada lado, así como de una cámara para detectar obstáculos y un sensor láser para mapear.

	Altura	Laser	Cámaras	Mapa Entorno	Peso	N.º antenas
Tally	162.5cm	Si	Si	Si	30Kg	1
Stockbot	190cm	Si	Si	Si	n.e. ³	8
Tory	190cm	Si	Si	Si	60Kg	n.e.
AdvanRobot-100	195cm	Si	Si	Si	47Kg	8

Tabla 1: comparativa entre los robots estudiados.

En la Tabla 1 se muestra una breve comparativa entre los 4 robots, mostrando en cada caso su altura, si consta de láser y cámaras, si es capaz de crear un mapa del entorno, su peso y su número de antenas.

Tras el análisis individual de cada producto, junto con la comparativa en la Tabla 1, se puede concluir que la tendencia actual de los robots de inventario para *retail* radica en un solo robot, de grandes dimensiones y peso (lo que implica que necesita muchos sensores y mucha potencia), cuyo objetivo es escanear el máximo porcentaje de tags RFID posible. En todos los robots, la campaña de marketing principal que se hace en sus páginas web, es la atribución de altos porcentajes de lectura [14]–[18]. Estos dispositivos son caros (de 30.000 a 60.000 €) y funcionan de manera individual, es decir, no son capaces de conectarse con otros robots para colaborar y compartir información. Además, se basan en el mapeado para la navegación, y por tanto necesitan elaborar un mapa de su entorno antes de poder operar en él. Todos tienen las antenas RFID en los laterales, menos Tory, cuyas antenas se sitúan en forma romboidal. Asimismo, constan de diversos sensores, característica usual en los robots grandes que requieren de mapeado, ya que están preparados para no colisionar nunca y capturar todas las características de su alrededor. Entre todos los sensores, destacan:

- Múltiples cámaras 3D, cuyo objetivo es captar obstáculos y ayudar a crear el mapa del local. Algunos robots, como Tally, constan también de cámaras 2D para detectar agujeros en las estanterías para reponer inventario. Las cámaras suelen ir situadas en la parte alta del robot.
- Sensor láser que ayuda a resolver el SLAM (*Simultaneous Localization and Mapping*) para la creación del mapa y el evitado de obstáculos. Este tipo de sensor es muy común en robótica, ya que (como su nombre indica) permite localizar objetos y mapear. Tiene un rango de visión de 360 grados, y consta de una alta tasa de lectura, con elevada exactitud y precisión. Se puede observar que todos los robots lo integran en la parte inferior de su estructura.

³ No especificado

e) Prototipo Preliminar

El objetivo del equipo de investigación es crear un robot que se pueda convertir en una alternativa al Advanrobot-100 y a su competencia actual en el mercado. Siendo los modelos actuales grandes y caros, la propuesta consiste en elaborar un robot con las siguientes características, que lo hacen más sencillo:

- Menores dimensiones.
- Menos sensores.
- Más autónomo operativamente.
- Menor coste de desarrollo.

El prototipo existente consta de las siguientes características:

Físicamente, tiene unas dimensiones menores en comparación a los robots de inventario del mercado actual. Al estar la mayoría de productos de tiendas a una altura normalmente no superior a 160cm, no se necesita un robot con una altura similar para leer a tal altura.

En cuanto a sus sensores, el robot solo consta de los *bumpers* del Turtlebot, un sensor láser, un sensor de ultrasonidos y 4 antenas, que se usan tanto para guiar la navegación como para realizar el inventario. Estos componentes se explican más detalladamente en el siguiente apartado “Diseño y componentes”.

En cuanto a la navegación y el inventario, el robot es autónomo, y eso hace que la interacción persona robot sea más sencilla. Se explica con más detalle la navegación del robot en el apartado “Navegación”.

Por inferencia de lo explicado, se obtiene un robot más sencillo, y por tanto éste tendrá un precio más reducido que la competencia.

Diseño y componentes

Actualmente el robot tiene un diseño preliminar, llevado a cabo solo para hacer pruebas, sin contemplarse aspectos de diseño que cabrá tener en cuenta, como su estabilidad o la distribución de sus componentes. Se puede considerar una prueba de concepto. Consta de dos partes diferenciadas: una base robótica móvil y una carga útil que se sitúa encima y que contiene todos los componentes para alcanzar la misión de inventario. En concreto, el robot consta de los elementos que podemos ver en la siguiente imagen:

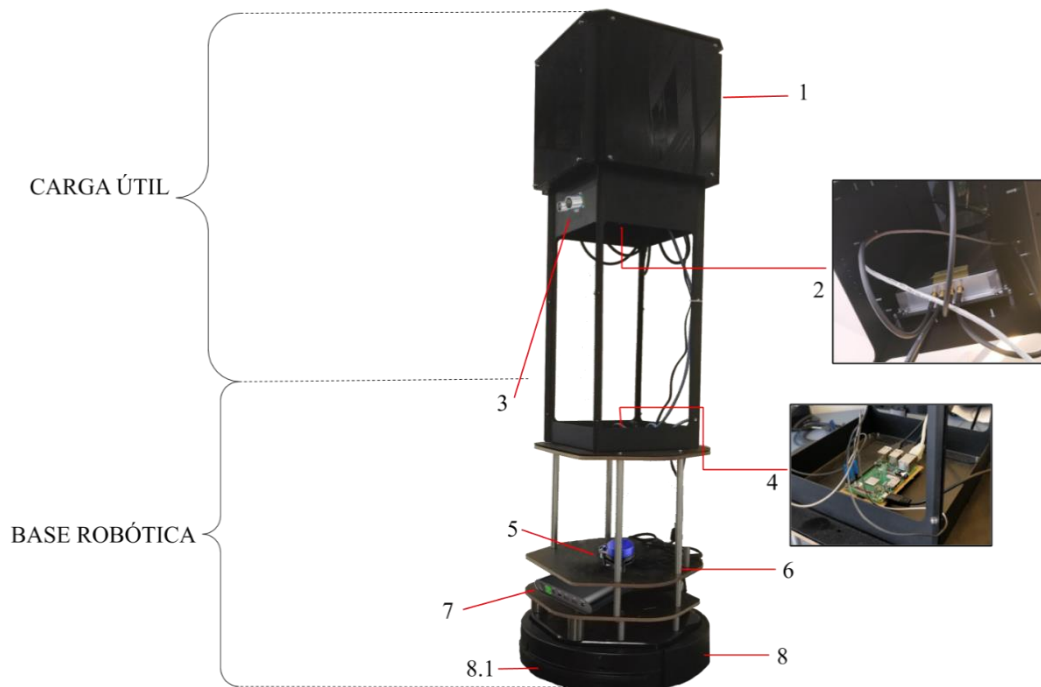


Fig. 14: esquema del diseño preliminar con componentes numerados [19], [20], [21]

En la Fig. 14 se puede observar una imagen del robot con sus partes numeradas y algunas perspectivas detalladas. Los componentes referentes a la carga útil del robot son:

1. Advantennas SP11 [6].
 - Están apantalladas.
 - Se usan 4 antenas en disposición cuadrada.
 - Se encuentran bajo una capa de metacrilato que las protege y ayuda al apantallamiento.
2. Keonn AdvanReader 153.03 [6].
 - Consta de 4 puertos.
 - Se encuentra dentro de una caja de Faraday.
 - Se sitúa escondido dentro del cubo formado por las 4 antenas.

Por otro lado, los componentes que componen la base robótica móvil del robot son:

3. Parallax 28015 [20].
 - Es un sensor de ultrasonidos⁴.
 - Apertura angular de unos 70°.

⁴ Llamado sensor de ultrasonidos o sonar.

- Detecta la distancia midiendo el tiempo que tarda un sonido desde que es emitido hasta que regresa.
 - Se complementa con el láser para la detección de obstáculos.
 - Detecta obstáculos a mayor altura que el láser.
4. Raspberry Pi 3B+ [21].
 - Es el “cerebro” del robot, el procesador que da las órdenes.
 5. Láser YDLIDAR X4 [19]
 - Detecta obstáculos a baja altura.
 - Calcula la distancia midiendo el tiempo que tarda un rayo de luz desde que es emitido hasta que rebota y es reflejado de vuelta.
 - Tiene un rango de detección limitado al eje horizontal.
 - Lee en 360°.
 6. Estructura para Kobuki Base.
 - Compuesta por bases de madera y soportes de aluminio.
 7. Batería externa Litium Tanker [22].
 - Se usa para la alimentación de algunos componentes del sistema.
 8. Kobuki Base [23].
 - Es una base de investigación móvil diseñada para investigación robótica.
 - Consta de bumpers (8.1), que son sensores mecánicos que detectan cuando el robot choca contra algún obstáculo.

En la Fig. 15 se puede observar una imagen del robot situado en un entorno llevando a cabo un inventario:



Fig. 15: robot preliminar.

Las medidas del robot son 35cm × 35cm × 110cm. Se puede observar que el robot es bastante alto considerando el espacio que se deja sin componentes. La única ventaja que ofrece ese diseño es la altura que se aporta a las antenas para leer a más altura.

El láser se encuentra en la parte inferior, y va girando para detectar obstáculos a su alrededor. El sensor ultrasónico se sitúa en una altura de 1m, y se complementa con el láser para detectar obstáculos y de esta manera evitar que el robot colisione contra objetos de mayor altura, como podrían ser mesas.

En cuanto a la estabilidad del robot, su altura hace que éste se balancee considerablemente al navegar en las pruebas que se llevan a cabo. Es muy sensible a pequeñas protuberancias de suelo (<5mm), y eso provoca un balanceo en la estructura. Al frenar también se percibe un balanceo muy evidente. Uno de los motivos de la inestabilidad del prototipo es que tiene un peso considerable en la parte superior, donde se encuentran las 4 antenas y el *reader*.

Conexiones y Alimentación

A continuación, en la Fig. 16 se puede observar el esquema de conexiones del robot preliminar:

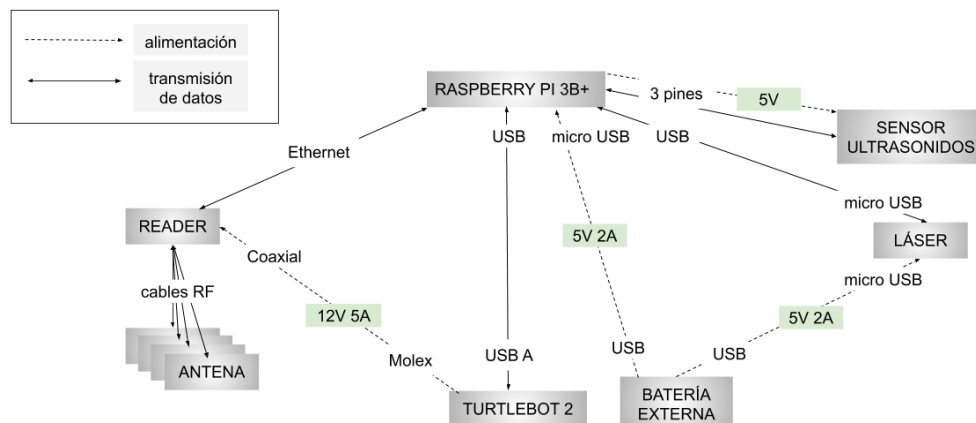


Fig. 16: esquema de conexiones del robot preliminar.

En la Fig. 16 se observa cómo, en cuanto a la distribución de potencia:

- La batería, de 50.000 mAh, alimenta a la Raspberry y al láser con 5 V y 2 A, a través de dos cables USB a micro USB.
- El Turtlebot, con su propia batería de 4.400 mAh alimenta el *reader* a través de una conexión de 12 V y 5 A.
- La Raspberry alimenta el sensor de ultrasonidos directamente a través de los pines (uno para tierra, otro para la alimentación, y otro para datos que contiene el *trigger* y el *echo*).

En cuanto a conectividad:

- La Raspberry se conecta a través de un cable Ethernet al *reader*, y a éste se le conectan 4 antenas SP11 con un cable especial para RFID, que es un cable coaxial con conectores RF.
- Al láser se conectan 2 cables micro USB, uno para los datos y otro para la alimentación. El cable de datos se conecta a la Raspberry.
- El Turtlebot tiene conectores Molex, a excepción del que se conecta con la Raspberry Pi para datos, que se trata de un cable de USB A (Turtlebot) a USB (Raspberry).

Navegación

El sistema de navegación del robot sigue una metodología parecida a la de las hormigas cuando realizan tareas como buscar comida, llamada estigmergia. Un sistema robótico basado en estigmergia tiene la característica de cambiar el entorno para añadir información, y crear una comunicación entre diferentes robots, que se hace de manera indirecta a través del entorno compartido. En este caso, los robots comparten una lista con los tags leídos y el número de veces que han sido observados. De esta manera, cuando un robot lee un tag, ya sabe si él u otro robot lo ha leído anteriormente, y ese es uno de los factores que tiene en cuenta para moverse por el entorno.

Un robot que crea un mapa del entorno para la navegación es capaz de situarse en unas coordenadas dentro del espacio. En cambio, el robot lleva a cabo una navegación en tiempo real, evitando los obstáculos cercanos y navegando sin situarse de manera precisa en el entorno, aunque siguiendo tags por él. Este tipo de navegación disminuye el número de sensores necesarios y la complejidad de uso, aunque hace el circuito del robot impreciso y no controlable.

Para detectar los obstáculos el robot utiliza el láser, el sensor de ultrasonidos y los *bumpers*. Cuando el láser o el sónar detectan un obstáculo cercano, el robot se para, retrocede, gira unos grados y continúa su marcha. Si el láser y el sónar fallan y el robot colisiona, los *bumpers* detectan el choque, y el robot actúa como en el caso anterior, de manera que retrocede, gira y sigue su navegación.

El algoritmo de navegación del robot se basa en leer los tags con las 4 antenas, agruparlos respecto a cada una de las 4 direcciones, y calcular un valor para cada dirección en función de la cantidad de tags nuevos detectados, cantidad de tags que ya habían sido leídos anteriormente y obstáculos cercanos. A continuación, el robot se mueve en la dirección que tenga un valor superior [24].

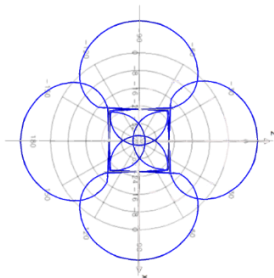


Fig. 17: rango aproximado del robot actual.

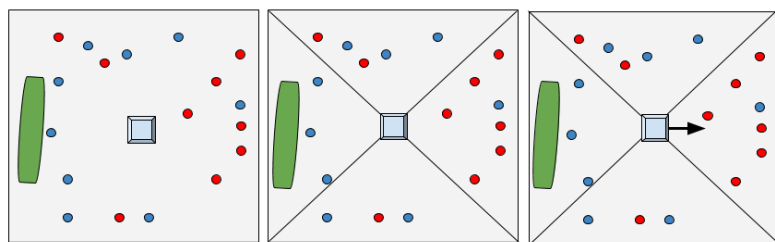


Fig. 18: ejemplificación del algoritmo que sigue el robot para la navegación.

En la Fig. 17 se puede observar un esquema del rango aproximado de las antenas, con el glóbulo de detección de las 4 antenas visto desde arriba. En la Fig. 18 se puede observar un esquema ejemplificado de un caso de uso del movimiento del robot. La figura verde hace referencia a un obstáculo. Los puntos azules, a los tags que han sido descubiertos y explorados por el mismo robot u otros robots, y los rojos a los tags que han sido leídos por primera vez. Además, se descartan direcciones donde hayan sido detectados obstáculos cercanos. Por ese motivo, el robot decide ir hacia la derecha, que es donde hay más tags que no se han leído con anterioridad y menos obstáculos visibles.

1.3 Hipótesis y propuestas de mejora

Se analiza el robot preliminar y sus necesidades. Se proponen una serie de hipótesis, con sus respectivas propuestas de mejora, que se someterán a los pasos estándar para el proceso de diseño de ingeniería. Las hipótesis son las siguientes:

1. **Hipótesis:** se pueden reducir considerablemente las dimensiones del robot para seguir satisfaciendo sus necesidades y su objetivo principal, a la vez que se consigue un diseño más ergonómico, más estable, sencillo y mejor para el movimiento del robot y su transporte. Se estima que se reducirá la calidad de lectura de los tags, pero si este porcentaje de pérdidas no es extremado se puede asumir para cumplir con los objetivos del robot.
 - **Propuesta:** reducir las dimensiones en altura, disminuyendo la elevación de las antenas.
2. **Hipótesis:** se puede mejorar la lectura de los tags cambiando la configuración de las antenas.
 - **Propuesta:** cambiar la configuración de las antenas y testear. Nada indica a priori que las 4 antenas en posición cuadrada y paralelas deban ser la mejor opción.
3. **Hipótesis:** se puede reducir la cantidad de componentes del sistema suprimiendo la batería. Esto proporcionaría diversas ventajas, como menos puntos de error o reducción del coste del dispositivo.
 - **Propuesta:** Suprimir la batería externa y alimentar el robot entero con la batería integrada del Kobuki base. Se deberá hacer un análisis de la batería y conexiones del robot para evaluar si la solución es viable.
4. **Hipótesis:** se puede contemplar un diseño modular mejorando la distribución de los componentes.
 - **Propuesta:** cambio de diseño interno, analizando el mejor lugar para situar los componentes, contemplando la modularidad del sistema.

Asimismo, se observan necesidades que deben ser satisfechas en el robot definitivo. Éstas son:

1. El usuario final necesita una manera de comunicarse con el robot.
 - **Propuesta:** Elaborar una interfaz para con el usuario final. ¿Cómo indicará el robot su estado, o la batería?
2. Se necesita un prototipo definitivo para lanzar al mercado.
 - **Propuesta:** elaborar un prototipo definitivo, contemplando aspectos como la protección completa del sistema interno del robot.

Se ha elaborado un repositorio en GitHub, que recoge los archivos, datos e información de todas las pruebas que se llevan a cabo en el proyecto relacionadas con las hipótesis y necesidades planteadas, y puede ser utilizado como complemento de lectura de la memoria. Se puede acceder a través del siguiente enlace: github.com/calilluis/TFG-RFID_Robot

2. DISEÑO Y PROTOTIPADO DE SUBSISTEMAS

Este capítulo contempla el diseño del sistema interno del robot. Se tratan las hipótesis 1, 2, 3 y 4.

Primero (punto 2.1) se tratan las hipótesis 1 y 2. Se hace un análisis y testeo de diferentes configuraciones de antenas, para elegir la mejor disposición de éstas en el prototipo definitivo. Junto a este análisis se contempla la elevación que tendrá el prototipo final, ya que ésta depende de la altura a la que se sitúen las antenas.

En segundo lugar (punto 2.2) se trata la hipótesis 3, llevándose a cabo el estudio sobre la supresión de la batería externa.

En tercer lugar (punto 2.3) se trata la hipótesis 4. Se contempla la distribución de componentes dentro del sistema, haciendo hincapié en la modularidad del robot, y se crea un nuevo prototipo del robot (punto 2.4), que cumple los requisitos que se observan en los primeros tres apartados.

Finalmente (punto 2.5), se contempla el desarrollo de un presupuesto de ejecución material, contemplando el precio de los componentes que se han necesitado para desarrollar el nuevo prototipo.

2.1 Disposición y altura de las antenas

En este primer apartado se lleva a cabo un análisis de lecturas con diversas configuraciones de antenas para encontrar la configuración y altura idóneas, tratando de optimizar el diseño del robot maximizando las lecturas de tags.

a) Hipótesis y propuestas

A partir de las hipótesis 1 y 2, referentes a la reducción de altura y cambio en la distribución de las antenas, se llevan a cabo una serie de propuestas de diseño para probar distintas maneras de situar las antenas, buscando obtener la mejor configuración posible para maximizar la lectura de tags reduciendo la altura del robot. De estos diseños, se elegirán algunos para testear.

Factores que se consideran para la creación de los nuevos diseños:

- Reducir considerablemente la altura del robot.
- Priorizar las antenas laterales para la lectura.

Se hipotetiza que para la mayoría de lecturas del robot, tienen más importancia las antenas laterales, ya que el robot pasa por los pasillos siempre con la misma orientación, y los tags que se deben leer están situados en los laterales de los pasillos. Una representación de este hecho puede ser observado en la Fig. 19, donde se observa que las antenas laterales están más centradas en la lectura.

- Restringir la colocación de las antenas al círculo circunscrito del robot (35cm), ya que si no el robot podría colisionar con las antenas y no con los bumpers.

- Priorizar la disposición y altura de las antenas por delante de otros aspectos de diseño (como por ejemplo la distribución de los demás componentes).

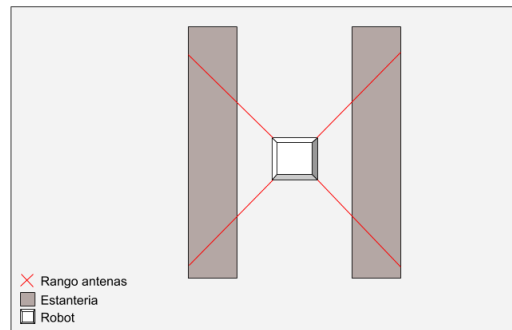


Fig. 19: representación de la lectura del robot en un pasillo.

Para todos los posibles diseños se hacen replicas 3D a escala con Tinkercad [25], una herramienta online para diseños 3D. Los diseños propuestos son los siguientes:

- **Inclinada:** dos antenas inclinadas y dos verticales. Las inclinadas se sitúan en los laterales del robot, para así conseguir el máximo rango de altura en los pasillos. Se plantea la hipótesis de que al tener el robot una altura tan reducida, inclinar las antenas puede hacer que se pueda leer a alturas más elevadas. En la Fig. 20 se percibe su diseño 3D junto con la flecha que señala su dirección de avance. En la parte inferior derecha se observa el rango de las antenas, percibiendo que las inclinadas tienen un rango menor en cuanto a distancia (pero lo tienen mayor en cuanto a altura).

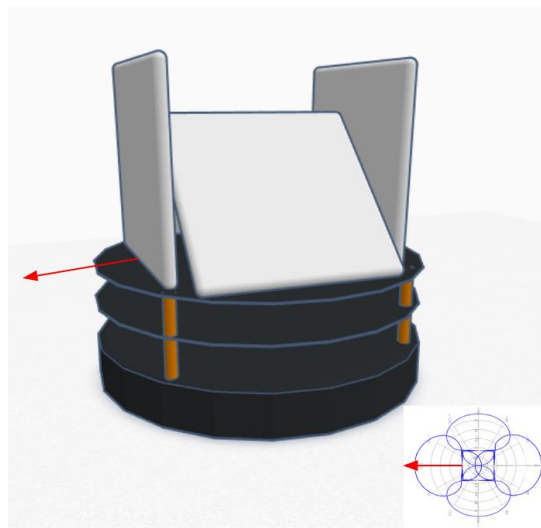


Fig. 20: modelo 3D de la configuración Inclinada.

- **Cuadrada:** Antenas verticales en disposición cuadrada ligeramente inclinadas hacia el interior, sin que sobresalgan del perímetro de la base. En la Fig. 21 se puede observar el modelo de Tinkercad correspondiente, junto con el rango de las antenas en la parte inferior derecha. La flecha roja señala la dirección de avance del robot.

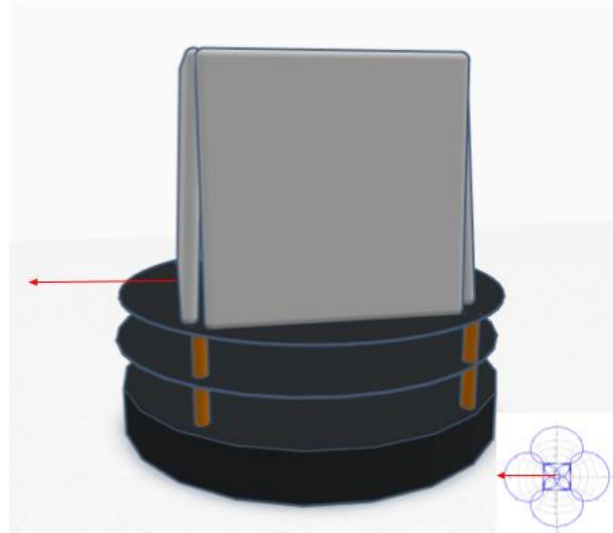


Fig. 21: modelo 3D de la configuración Cuadrada.

- **Romboidal:** Misma combinación que la Cuadrada, pero girando el bloque de antenas 45°. Se parte de la hipótesis de que tal vez puede dar un mejor resultado situar 2 antenas apuntando a los laterales, que solo 1 (caso de la configuración Cuadrada). Además, este diseño se basa en la configuración de antenas que lleva el robot Tory, analizado anteriormente en “d) Robots RFID de inventario”, donde se observa que las antenas de Tory están situadas romboidalmente.

En la Fig. 22 se observa el renderizado 3D y su respectivo rango por antena. La flecha roja señala la dirección de avance del robot.

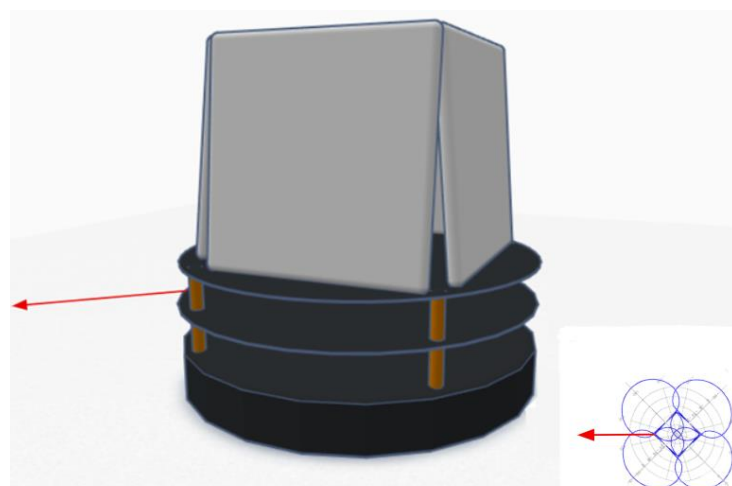


Fig. 22: modelo 3D de la configuración Romboidal.

- **Dos niveles:** se proponen 2 diseños con 2 niveles de altura para las antenas, observables en las Fig. 23 y Fig. 24. El motivo de estas configuraciones es que las antenas laterales están más elevadas, para leer a más altura, mientras que las delanteras se inclinan levemente para conseguir también más rango de altura. La diferencia entre las dos propuestas es que, en la segunda, las antenas laterales se inclinan también, mientras que en la primera propuesta se mantienen rectas.

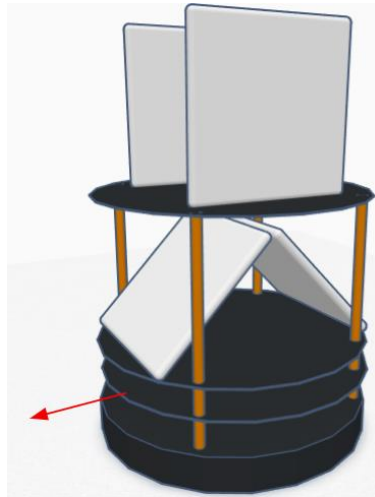


Fig. 23: modelo 3D de 2 niveles. Antenas frontales inclinadas y laterales verticales.

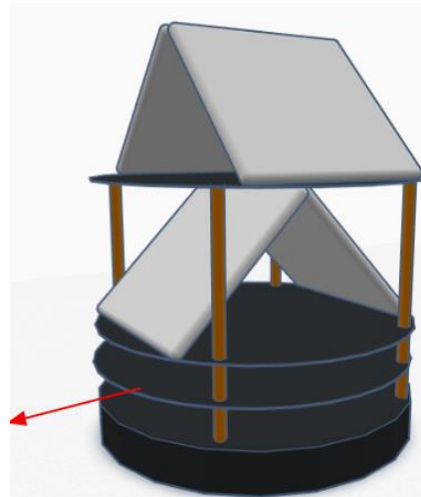


Fig. 24: modelo 3D de 2 niveles. Antenas frontales y laterales inclinadas.

Conclusión de Propuestas

Finalmente, se deciden probar los diseños de un nivel, ya que reducen la complejidad del robot y de las posibles pruebas, aparte de conservar mejor la simetría. Las configuraciones de dos niveles distorsionan la proporción del diseño y se podría crear una diferencia de lectura demasiado grande entre las 2 antenas laterales y las 2 delanteras, lo cual puede provocar un problema en la navegación.

Se prueban los diseños Inclinada, Cuadrada, y Romboidal, junto a la configuración original, para poder comparar los resultados con las demás configuraciones.

b) Desarrollo de la solución

Para poder probar los diferentes diseños, se construyen dos estructuras con cartón. Una para la configuración Cuadrada y Romboidal, y otra para la configuración Inclinada. Las dimensiones se calculan partiendo de las siguientes dimensiones:

- Una antena mide 20.7 cm \times 20.7 cm.
- El diámetro del Turtlebot es de 35 cm.

Las estructuras con sus respectivas medidas se muestran en las figuras Fig. 25 y Fig. 26 inferiores. La Fig. 25 se usa para las configuraciones Cuadrada y Romboidal, y la Fig. 26 para la Inclinada.

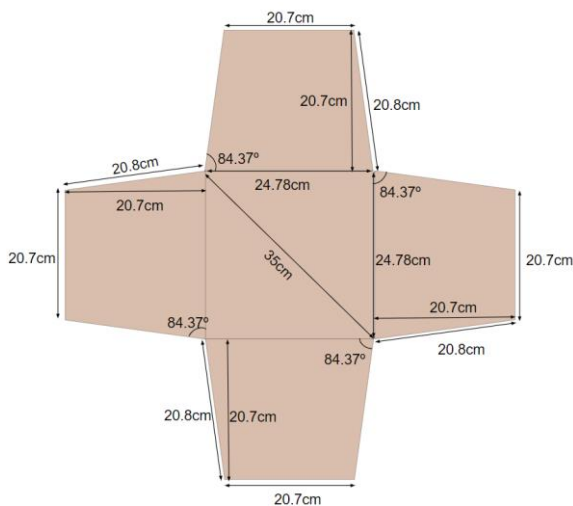


Fig. 25: estructura con medidas para la configuración Cuadrada y Romboidal.

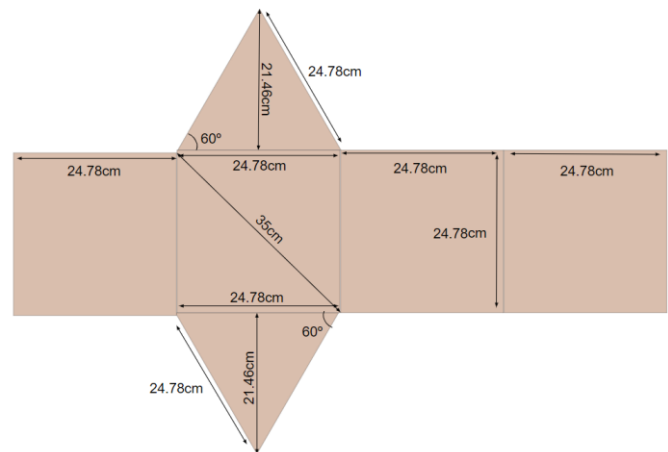


Fig. 26: estructura con medidas para la configuración Inclinada.

Para el montaje del sistema, se atornillan las antenas a las estructuras y se retiran el láser y el sensor de ultrasonidos. Éstos no son necesarios, ya que se manejará el robot manualmente. En la primera estructura, el *reader* se sitúa en el interior de ésta. En la segunda el *reader* no cabe debido a la inclinación de las antenas laterales, y por ello se sitúa en un nivel intermedio entre el Turtlebot y la estructura.

En las siguientes figuras se pueden observar las 2 estructuras montadas en el robot.



Fig. 27: prototipo construido de la configuración Inclínada.



Fig. 28: prototipo construido de la configuración Cuadrada/Romboidal.

En la Fig. 27, se muestra el prototipo Inclínada llevando a cabo una prueba de inventario, mientras que en la Fig. 28 se observa una imagen del prototipo con configuración Cuadrada/Romboidal.

c) Testeado y análisis

Se plantean 2 pruebas distintas para el testeado de los prototipos:

1. **Prueba de entorno:** Se simula un espacio real, en el que el robot circula por un circuito. Útil para evaluar el rendimiento general del robot en un entorno realista.
2. **Prueba en altura:** se hace circular el robot por delante de una caja de tags a diferentes alturas y distancias. Al reducirse las dimensiones del robot, se debe evaluar si en alturas elevadas se pierde capacidad de lectura, y hasta que altura se pueden obtener lecturas fiables.

1. Prueba de entorno

Se hace circular el robot en un espacio controlado (interior de almacén, Keonn) simulando los pasadizos de un local. El robot se controla publicando mensajes manualmente a un topic (e.g. `rostopic pub -r 2 /robot1/cmd_vel_mux/input/teleop geometry_msgs/Twist (1,0,0) (0,0,0)`). Hay 1375 tags repartidos en estanterías (603 tags) y en cajas (772 tags). Éstos a la vez se dividen en bloques de diferente tamaño.

El entorno usado es el siguiente:

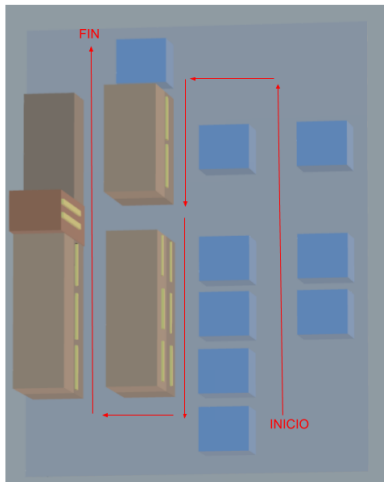


Fig. 29: renderizado 3D del entorno con el recorrido del robot. Vista superior.

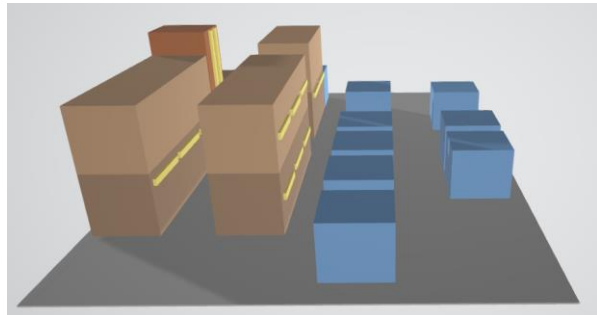


Fig. 30: renderizado 3D del entorno, vista isométrica.



Fig. 31: recorrido del robot señalado en una imagen real.

En las Fig. 29 y Fig. 30, se muestra un renderizado en 3D del entorno. Las cajas se visualizan en azul. En marrón, las estanterías y la columna. Por último, los tags se representan en color amarillo, aunque en el interior de las cajas también hay. En la Fig. 31 se puede observar el circuito en una fotografía real del entorno. El robot primero pasa entre las cajas, da la vuelta, vuelve a pasar entre las estanterías y las cajas, y vuelve a rotar para hacer la última pasada con estanterías en ambos lados.

Cada una de las 4 configuraciones se hace circular por el recorrido dos veces, consiguiendo así un total de 8 resultados ($4 \text{ configuraciones} \times 2 \text{ repeticiones}$), que a la hora de procesar se tratan todos como resultados independientes.

Se lleva a cabo un análisis sobre los resultados obtenidos. Se explica en el siguiente apartado.

Metodología

De los datos obtenidos en los circuitos se hacen una serie de análisis, de los que se extraerán unas conclusiones. Los diferentes análisis que se llevan a cabo son los siguientes:

1 - Porcentaje de lecturas en bruto:

Se calcula la cantidad de tags (únicos⁵) leídos por la configuración testada divididos entre la suma de todos los tags (únicos) leídos por todas las configuraciones. Se evalúa por tanto en este análisis la capacidad de lectura general de los prototipos, contemplando tags dentro y fuera del circuito.

2 - Porcentaje de lecturas filtradas:

Se calcula la cantidad de tags (únicos) leídos por la configuración testada divididos entre la suma de los tags que se encuentren en el circuito. Los tags de dentro del circuito se recogen en un fichero, y se encuentran divididos por bloques. En el análisis anterior se consideraban todos los tags leídos por al menos una configuración, y en este se consideran solo aquellos que se encuentren dentro de los bloques de tags del circuito.

2.1 - Filtro doble:

Se calcula la cantidad de tags (únicos) leídos por la configuración testada divididos entre la suma de los tags que se encuentren en el circuito y que hayan sido leídos al menos por una configuración. Este segundo filtro proporciona una capa más para asegurar que los tags que se tienen en cuenta son leíbles, y se intentan observar unos resultados más ajustados a las capacidades de la tecnología RFID, ya que se descartan:

- Tags que consten en la base de datos del circuito pero que realmente se hayan suprimido en el entorno real.
- Tags dañados o defectuosos que no puedan ser leídos.
- Tags situados en puntos demasiado remotos como para ser leídos.

2.2 - Porcentajes por antena:

Idéntico al resultado 2.1, pero mostrando el porcentaje que lee cada antena. Este análisis es útil para observar si los tags se leen de manera repartida entre las distintas direcciones, y para analizar las distintas hipótesis referentes a una mejor lectura en las antenas laterales.

3 - Mapa de doble filtrado:

Mapa del circuito con el porcentaje de lecturas conseguido por la configuración en cada bloque de tags. Se observa por cada bloque el promedio del porcentaje leído en las 2 iteraciones de cada configuración, usando los valores obtenidos en el análisis 2.1. Se juntan las 4 configuraciones en el mismo mapa.

⁵ Un mismo tag puede ser leído varias veces, pero solo se contempla una vez para el cálculo.

Para cada uno de los análisis se crea un script específico. Todos ellos se comportan de la misma manera: cargan un .csv con los datos recogidos por el *reader* y lo procesan para extraer los diferentes resultados que se mencionan en el apartado anterior. Asimismo, todos procesan los 8 mismos ficheros, que representan las lecturas leídas en las dos repeticiones de las 4 diferentes configuraciones, y se ejecutan todos de la misma manera: `python <nombre_script.py>` (e.g. `python script_entorno_1-lecturasEnBruto.py`).

Se pueden encontrar los scripts en el enlace de GitHub siguiente: https://github.com/calilluis/TFG-RFID_Robot/entorno, donde se explican brevemente. En este no solo se incluyen los scripts de cada análisis, sino que también se encuentran disponibles las tablas de resumen de resultados, en el archivo “tablas_analisis_entorno.xlsx” para los análisis 1, 2, 2.1 y 2.2, y “mapa_pruebaDobleFiltrado.xlsx” para el mapa del análisis 3.

Análisis de Resultados

Los resultados de los diferentes análisis se resumen en tablas, y se analizan en este apartado.

1 - Porcentaje de lecturas en bruto:

REPET. /CONFIG.	ORIGINAL	INCLINADA	CUADRADA	ROMBOIDAL
1	75,22%	65,14%	79,24%	68,11%
2	76,69%	60,34%	71,88%	68,32%
Media	75,96%	62,74%	75,56%	68,22%
Desviación Estándar (σ)	0.010	0.034	0.052	0.001

Tabla 2: porcentaje de lecturas en bruto por configuración y repetición. Se muestra la media y la desviación estándar.

En la Tabla 2 se muestran los porcentajes de lecturas en bruto por cada configuración en las 2 repeticiones, así como la media de cada configuración y su desviación estándar.

Se observa que, en este primer caso:

- De las configuraciones propuestas, la que tiene mejores porcentajes es la Cuadrada, aunque hay bastante diferencia de porcentajes entre las 2 iteraciones (es la configuración con más desviación estándar). La Romboidal en este caso es la configuración con menor desviación estándar, aunque con lecturas generales por debajo de la Cuadrada. La configuración Inclínada ofrece unos resultados por debajo de las otras, obteniendo unos porcentajes 5 puntos por debajo de la Romboidal y 13 por debajo de la Cuadrada.
- En cuanto a la diferencia entre la configuración Original y las propuestas, el promedio de lecturas de las nuevas configuraciones es de 68.84%, en contraposición del 75.96% de promedio de la configuración Original (7.125% menos).

2 - Porcentaje de lecturas filtradas:

REPET. /CONFIG.	ORIGINAL	INCLINADA	CUADRADA	ROMBOIDAL
1	85,39% (+10,17%)	82,89% (+17,75%)	78,35% (-0,89%)	82,59% (+14,48%)
2	85,47% (+8,78%)	81,91% (+21,57%)	76,00% (+4,12%)	71,84% (+3,52%)
Media	85,43%	82,40%	77,18%	77,22%
Desviación Estándar (σ)	0.001	0.007	0.017	0.076

Tabla 3: porcentaje de lecturas filtradas por configuración y repetición. Se muestra la media y la desviación estándar.

En la Tabla 3 se muestra el resumen de resultados de este segundo análisis donde, como ya se ha comentado, se tienen en cuenta solamente los tags del circuito. La tabla sigue la misma estructura que la anterior, pero mostrando entre paréntesis la diferencia de mejora en contraposición al análisis anterior.

- En este análisis, de las 3 propuestas, la configuración Inclínada es la que obtiene un mejor resultado. Es la más estable (menor σ) y con mayores porcentajes. Se observa como la configuración Cuadrada queda esta vez por debajo de la Inclínada. La Romboidal, por otro lado, de promedio obtiene unos resultados un 0.1 % mejores que la Cuadrada, aunque su desviación entre las dos repeticiones es elevada, por lo que se puede considerar la peor de las 3 propuestas.
- En promedio, la configuración Original obtiene un 6.5% más de tags leídos en el entorno respecto al promedio de porcentajes de las otras configuraciones. Se observa un decrecimiento de esta diferencia en comparación con el primer análisis. A través de esta observación, se puede concluir que la configuración Original había leído tags fuera del circuito que las otras no habían leído.
- En cuanto a la comparativa de este análisis con el anterior, se observa que, al limitar los resultados a los tags del circuito, hay un incremento en las lecturas. De promedio, se obtiene un 78.93% vs el 68.84% del análisis del punto 1 (sin contar la configuración Original). Se observa como la configuración que más aumenta es la Inclínada; una posible explicación es que se trata de una configuración que lee muy bien a corto rango (tags dentro del circuito) pero que no tiene un alcance muy elevado (tags fuera del circuito).

2.1 - Porcentaje con filtro doble:

REPET. /CONFIG.	ORIGINAL	INCLINADA	CUADRADA	ROMBOIDAL
1	92,57% (+7,18%)	88,70% (+5,81%)	83,56% (+5,21%)	88,46% (+5,87%)
2	92,65% (+7,18%)	87,91% (+6,00%)	81,26% (+5,26%)	76,36% (+4,52%)
Media	92,61%	88,31%	82,41%	82,41%
Desviación Estándar (σ)	0.001	0.016	0.086	0.006

Tabla 4: porcentaje de lecturas con filtro doble, por configuración y repetición. Se muestra la media y la desviación estándar.

En este caso, se muestran en la Tabla 4 los resultados del análisis 2.1. Esta tabla sigue la misma estructura que la del análisis anterior.

- Observando las 3 configuraciones propuestas, la Inclínada es la que sigue ofreciendo un mejor resultado, tanto a nivel de lecturas como en una menor σ . La Romboidal obtiene unos resultados dispares entre las dos iteraciones, de la misma manera que en el análisis anterior. La Cuadrada ofrece unos resultados un 6% por debajo de la Inclínada.
- En cuanto a la comparativa de las propuestas con la configuración Original, en este caso el promedio de las 2 iteraciones de la configuración Original obtiene un 8.23% más que el promedio de las otras configuraciones. La Original obtiene unas lecturas del 92%, con lo que se puede observar que hay tags (< 8%) que han sido leídos por otra configuración, pero no por la Original.
- Comparando este análisis con el anterior, se observa que, de promedio en las 2 repeticiones de todas las configuraciones, se mejora un 5.84%. La configuración Original es la que más aumenta su porcentaje, seguida de la Inclínada. Esto se debe a que, como ya se ha percibido, la configuración Original ha leído tags que las demás configuraciones no han leído.

2.2 - Porcentajes por antena:

Este análisis se lleva a cabo con los resultados del análisis 2.1, con la única diferencia de separar los porcentajes obtenidos por antena. La configuración Romboidal se trata por separado, ya que sus direcciones están giradas 45°. Por tanto, se consideran las direcciones “frontal derecha”, “frontal izquierda”, “trasera derecha” y “trasera izquierda” para la configuración Romboidal, y “frontal”, “trasera”, “derecha” e “izquierda” para las demás configuraciones.

Los resultados se dividen en la Tabla 11 y Tabla 13 para la configuración Romboidal, y en las tablas Tabla 10 y la Tabla 12 para las demás configuraciones. Éstas se encuentran disponibles en el apartado de Anexos. Para cada tabla, se muestran a continuación el Gráfico 1 , Gráfico 2, Gráfico 3 y Gráfico 4, que permiten observar de manera visual la repartición de lecturas por antena en cada configuración para detectar aquellas más estables.

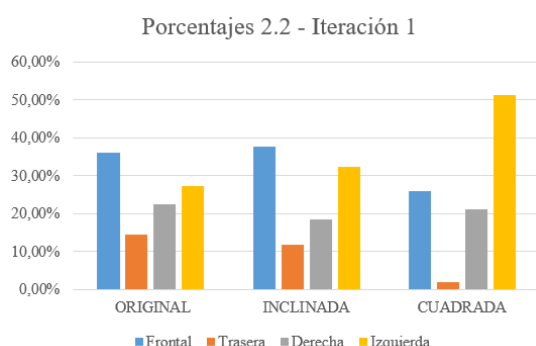


Gráfico 1: Porcentajes de lectura de cada configuración (menos la Romboidal) en cada dirección. Iteración 1. Referente a la Tabla 10.

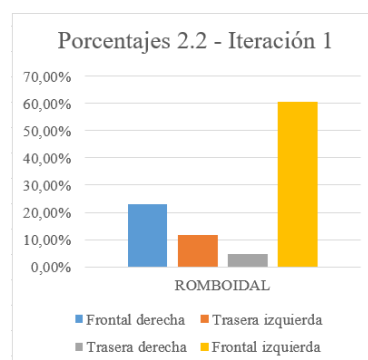


Gráfico 2: Porcentajes de lectura de la configuración Romboidal en cada dirección. Iteración 1. Referente a la Tabla 11.

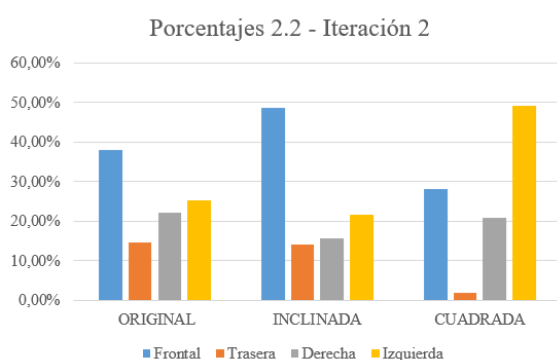


Gráfico 3: Porcentajes de lectura de cada configuración (menos la Romboidal) en cada dirección. Iteración 2. Referente a la Tabla 12.

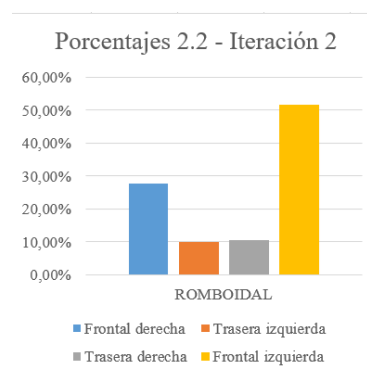


Gráfico 4: Porcentajes de lectura de la configuración Romboidal en cada dirección. Iteración 2. Referente a la Tabla 13.

Las conclusiones que se obtienen del análisis son las siguientes:

- En los gráficos Gráfico 1 y Gráfico 3 se observa la obtención de un porcentaje casi nulo en la antena trasera (puerto 3) de la configuración Cuadrada. Se podría considerar como un outlier, ya que la antena podría no estar bien conectada. Cabe

tener en cuenta este hecho, ya que tal vez la configuración Cuadrada podría haber obtenido unos resultados mejores.

- Al observar los 4 gráficos superiores se distingue que:
 - No hay mucha diferencia entre las 2 iteraciones, todas las configuraciones se comportan de una manera similar, lo cual puede indicar que los resultados obtenidos son coherentes con la realidad.
 - La distribución más equilibrada de lecturas por antena se consigue en la configuración Original, ya que es la configuración con menos diferencia de lectura en sus antenas ($\sigma = 0.087$). La sigue la configuración Inclínada ($\sigma = 0.132$). La Cuadrada obtiene un resultado menos equilibrado que las anteriores ($\sigma = 0.184$), que en el caso de que las antenas estuvieran mal conectadas, podría haber obtenido un resultado más equilibrado. La configuración Romboidal obtiene una diferencia elevada entre sus antenas traseras y sus delanteras, en este caso obteniendo la diferencia más elevada de lecturas por antena de las 4 configuraciones testeadas ($\sigma = 0.208$).
- En los porcentajes obtenidos por dirección se observa que, sin considerar la configuración Romboidal:
 - En las antenas frontales, se obtiene un 35.73% de lecturas.
 - En las traseras, un 9.75%.
 - En las antenas derechas, un 20.09%.
 - Finalmente, en las izquierdas se obtiene un 34.43%.

La antena frontal y la izquierda son las que más leen, seguidos de la antena derecha. La trasera es la que lee menos siempre. Esto se podría justificar por el hecho de que los tags por defecto están configurados en un modo en el que se desactivan un tiempo al ser leídos por primera vez. Por tanto, dado el tipo de movimiento del robot (siempre hacia delante) es normal que la antena frontal lea más tags nuevos que la trasera. Por otro lado, al principio del circuito el robot tiene la mayoría de tags situados a su izquierda; ese podría ser un motivo de justificación de su mayor cantidad de lecturas por la antena izquierda.

En la configuración Romboidal se percibe:

- En la antena frontal derecha, un 25.42% de lecturas.
- En la trasera izquierda, un 10.82% de lecturas.
- En la antena trasera derecha, un 7.61% de lecturas.
- En la frontal izquierda, un 56.16% de lecturas.

En las antenas traseras se obtiene un porcentaje bajo comparado con las frontales, de igual manera que las demás configuraciones obtienen un menor porcentaje en su antena posterior.

3 - Mapa de doble filtrado:

El análisis 3 se sitúa en un mapa, para así poder visualizar el porcentaje de cada bloque. Como se ha mencionado anteriormente, en este enlace se puede encontrar el documento “mapa_pruebaDobleFiltrado.xlsx”, que contiene el mapa del análisis 3 (en la hoja 1), junto con un mapa para cada configuración e iteración (en la hoja 2). Se analiza el mapa de la hoja 1, disponible en la Fig. 32 de la siguiente página, ya que en éste se visualizan los resultados de las 4 configuraciones a la vez, lo que facilita la comparativa.

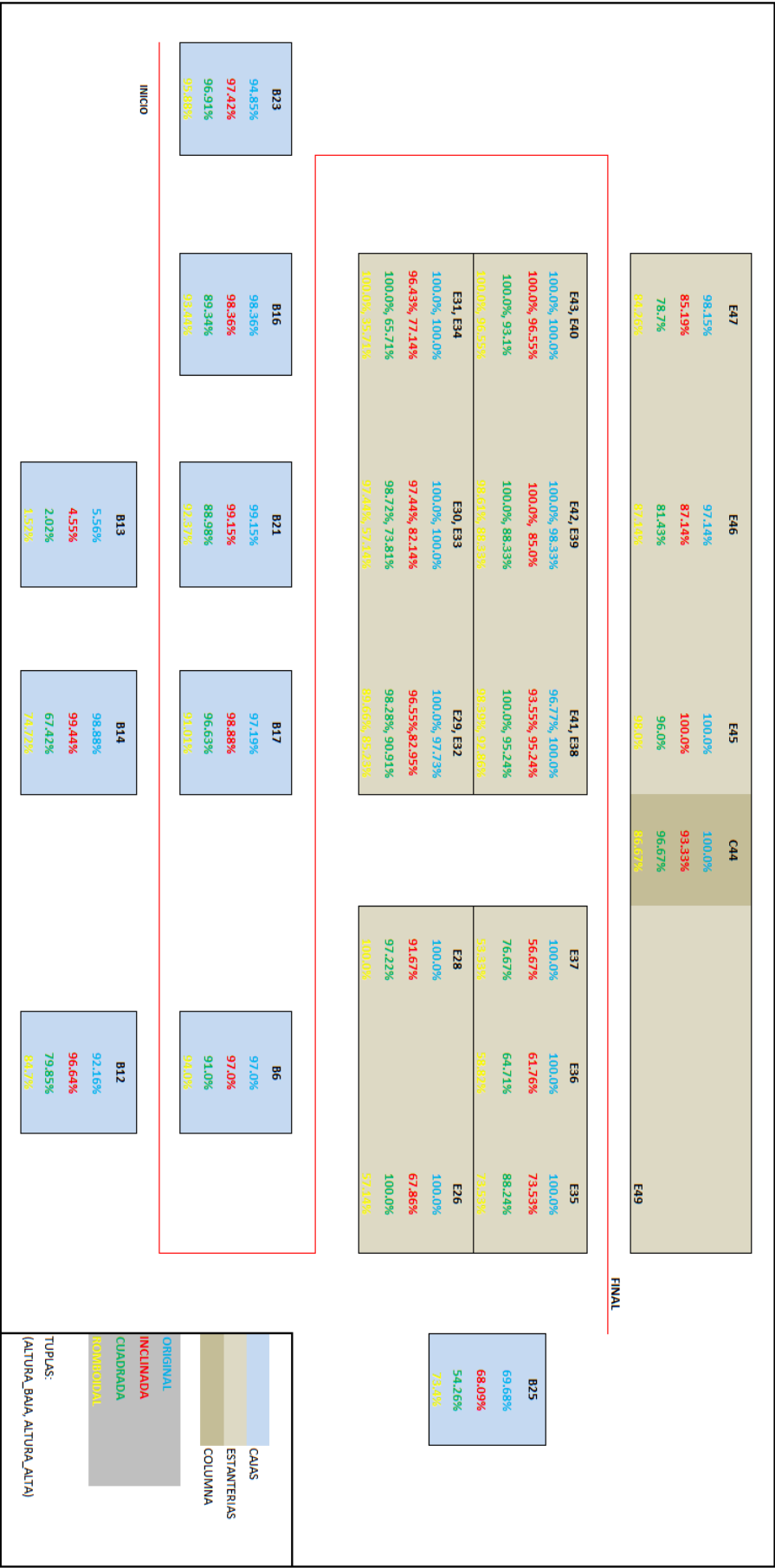


Fig. 32: mapa del análisis 3.1. Leyenda en el recuadro inferior derecho.

Como se indica en la leyenda de la Fig. 32 , en azul claro se sitúan las cajas del recorrido; en marrón flojo, las estanterías. Algunas estanterías tienen dos pisos. En tal caso, el orden es primero la altura baja (estante B), y después la altura alta (estante A); en color marrón fuerte se sitúan las columnas.

Los porcentajes azules, rojos, verdes y amarillos indican el valor del promedio de las 2 repeticiones para cada una de las configuraciones Original, Inclinada, Cuadrada y Romboidal, respectivamente.

Observaciones:

- De las 3 configuraciones propuestas, la que obtiene unos mejores porcentajes en la zona de las cajas es la Inclinada, mientras que en las estanterías la configuración Cuadrada es la que muestra un mejor rendimiento. Por ello, se decide compararlas entre ellas. Se puede observar la comparativa entre éstas dos configuraciones en la figura siguiente.



Fig. 33: comparativa entre la configuración Cuadrada y la Inclinada.

En la Fig. 33 se observan de color rojo los bloques de tags donde se obtienen unos mejores resultados en la configuración Inclinada, y de color verde en el caso de unos mejores resultados en la Cuadrada. Los bloques de color gris flojo indican que ambas configuraciones obtienen el mismo porcentaje de lectura.

- La caja B13 obtiene unos resultados muy malos, podría ser un outlier, aunque no se considera como tal.
- En las estanterías centrales, que constan de dos pisos de estanterías, se observan siempre mejores resultados en las alturas bajas (estante B) que en las alturas altas (estante A). En concreto, para cada configuración, la diferencia es la siguiente:
 - Inclinada → Estante B: 97.32%. Estante A: 86.5%.
 - Diferencia: 10.82%
 - Original → Estante B: 99.46%. Estante A: 99.34%.
 - Diferencia: 0.12%
 - Cuadrada → Estante B: 99.5%. Estante A: 84.51%.
 - Diferencia: 14.99%
 - Romboidal → Estante B: 97.35%. Estante A: 75.97%.
 - Diferencia: 21.38%

Se observa que la configuración con menor diferencia es la Original, debido a su mayor altura. La Inclinada muestra el resultado con menos diferencia entre los

estantes A y B de las tres configuraciones propuestas. Por otro lado, se observa que la configuración Cuadrada ha leído mejor en el estante B, y la Inclínada en el estante A.

La configuración Romboidal obtiene un resultado similar a las configuraciones Cuadrada e Inclínada en el estante B, pero un porcentaje bajo en el estante A. Sus resultados en el estante A pueden ser el motivo de su peor rendimiento en comparación a las demás configuraciones.

Conclusiones prueba de entorno

A continuación, se presentan las conclusiones obtenidas en esta prueba:

- Los porcentajes obtenidos en los análisis 2, 2.1 y 2.2 son superiores en la configuración Inclínada en comparación a las demás configuraciones. Por otro lado, se puede considerar que la mejor configuración sin aplicar ningún filtro es la Cuadrada, por sus mejores resultados en el análisis 1, que solo mide la capacidad de lectura en general. No se puede concluir en este caso qué configuración de las dos sería mejor en un entorno real, aunque se considera que la configuración Inclínada da un mejor resultado en esta prueba de entorno por su superioridad en la mayoría de análisis.
- La configuración Romboidal es la menos estable en cuanto a resultados, como se puede observar en su elevada desviación estándar entre sus dos iteraciones. En la segunda iteración se obtiene un porcentaje de lecturas bajo.
- Las antenas traseras siempre leen menos.
- Se refuta la hipótesis de que las antenas laterales leen más. El análisis 2.2 se demuestra que no es así.
- La configuración Inclínada lee mejor que la Cuadrada en alturas altas, como se demuestra en el análisis 3.
- El espacio en el que se hace esta prueba puede provocar interferencias, ya que está lleno de otros dispositivos RFID activos, y por ese motivo se observan en algunos casos resultados raros y menores lecturas (e.g.: caja B13).
- En ninguno de los modelos (quitando la configuración Original) se apantallan las antenas. Se podrá observar en la prueba de altura cómo el hecho de proteger las antenas hace que se obtengan unos mejores resultados. Se deduce que se hubieran obtenido unos resultados mejores al proteger las antenas en la prueba de entorno, pero no se repiten los resultados debido a que en este caso se intenta comparar las configuraciones entre ellas, y no su valor absoluto de lectura.

2. Prueba de altura

En esta prueba se lleva a cabo un análisis de lecturas del robot, pero centrándose en la altura máxima a la que pueden llegar a leer las configuraciones.

Para llevar a cabo esta segunda prueba, se utiliza una caja de cartón que contiene una matriz de 100 tags, con 20 filas por 5 columnas. Cada columna se separa de otra unos 10 cm. Por otro lado, las filas están equiespaciadas la medida del tag. Cada fila está a 5.5 cm de la siguiente. En total la caja mide 50×118.5 cm.

Se hace circular cada configuración con la caja a 3 alturas distintas (evaluando su capacidad de lectura variable en el eje z), y para cada altura se hace circular el robot a 3 distancias diferentes (evaluando su capacidad de lectura variable en el eje y):

-1. Altura baja- 0 cm - 118.5 cm	-2. Altura Media - 74 cm – 192.5 cm	-3. Altura elevada- 132 cm – 250,5 cm
A 40 cm.	A 40 cm.	A 40 cm.
A 70 cm.	A 70 cm.	A 70 cm.
A 100 cm.	A 100 cm.	A 100 cm.

Tabla 5: tabla resumen de la prueba de altura. Se sitúa la caja a 3 alturas distintas, y se hace circular el robot a 3 distancias distintas.

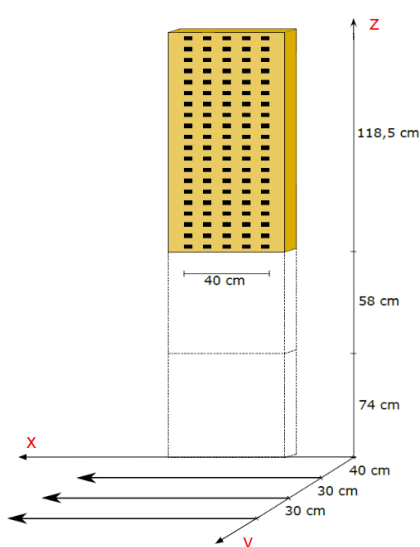


Fig. 34: esquema de la caja de tags.



Fig. 35: configuración Cuadrada corriendo la prueba de altura a nivel de suelo.

En la Fig. 34 se puede ver un esquema de las medidas de la caja, con las 3 alturas en el eje vertical “z”, y las tres distancias en el eje horizontal “y”. Asimismo, en la Fig. 35 se puede observar una fotografía de una prueba real, en concreto de la configuración Cuadrada a 40cm de la caja a baja altura.

En esta prueba, a diferencia de la prueba de entorno, se contemplan las configuraciones con las antenas protegidas, ya que en este caso no solo se requiere que la prueba sea comparativa, sino que también se necesita saber hasta qué punto lee bien cada

configuración. Eso lleva a intentar generar pruebas lo más cercanas posible a la realidad, donde el robot definitivo llevará las antenas apantalladas. El apantallamiento debe dar, en teoría, un mejor resultado. Se crean 3 nuevos modelos idénticos a los anteriores, pero con las antenas protegidas, y se testean junto a los modelos sin apantallar. Las configuraciones Cuadrada y Romboidal se protegen con el mismo sistema que integra la configuración Original, mientras que la configuración Inclínada se protege manualmente. Se forra la estructura de cartón con papel de aluminio para crear una pantalla trasera, y se incluye un espacio de 1cm entre la antena y la pantalla de aluminio.

Metodología

Se genera, para cada prueba en una altura y distancia distintas, un archivo .csv diferente, que contiene las lecturas del robot en esa prueba. Por tanto, cada configuración obtiene 3 alturas \times 3 distancias = 9 archivos.

Para el procesamiento de las pruebas, se usa un solo *script*. Éste procesa el archivo .csv, dando como output aquellos tags que se encuentran en la caja, pero no han sido leídos por la configuración. El script se encuentra disponible en https://github.com/calilluis/TFG-RFID_Robot/altura, junto con una breve explicación del código. En la figura siguiente se observa un ejemplo de la salida del script:

```
El tag: e28011700000020910000025 no ha sido leído por la configuracion.  
El tag: e28011700000020910000078 no ha sido leído por la configuracion.  
Se han leído 98/100 tags.
```

Fig. 36: output del script.

En la Fig. 36 se observa el output del script, en este caso ejecutando el análisis de la configuración Romboidal a 100cm de la caja en altura baja. En esta prueba se percibe que los tags 25 y 78 no han sido leídos.

Análisis de Resultados

En el último enlace mencionado (https://github.com/calilluis/TFG-RFID_Robot/altura) se pueden ver los resultados de la prueba de altura en el archivo adjunto “tablas_analisis_altura.xlsx”, donde también se muestran gráficamente los tags no leídos en cada prueba y el resumen de resultados en tablas.

Se divide el estudio de esta prueba en función de las alturas, obteniendo 3 análisis de todas las configuraciones con sus respectivas distancias. En estos análisis se muestran los resultados sin apantallar y apantallados, ya que interesa conocer el efecto del apantallamiento en los prototipos. De todos modos, se contemplarán solamente los resultados apantallados para la comparativa entre configuraciones, ya que en el prototipo final las antenas irán apantalladas.

Los resultados de las pruebas a 3 alturas diferentes se pueden ver recogidos en las tablas Tabla 6, Tabla 7 y Tabla 8. Cada tabla consta de los resultados obtenidos para cada configuración, en sus dos versiones (apantallada y sin apantallar). Dentro de cada casilla de resultados se observan en una tupla los errores obtenidos en las 3 distancias posibles

(40cm, 70cm, 100cm). También se observa una suma de errores por configuración y por apantallamiento, en las que se engloban las 3 distancias.

Análisis 1 – Altura baja:

Los resultados de esta primera prueba, de 0 cm a 118,5cm, se observan en la Tabla 6 siguiente.

	ORIGINAL	INCLINADA	CUADRADA	ROMBOIDAL	TOTAL
NO-PROT.	-	(0,0,1) ERR	(0,0,3) ERR	(1, 2,2) ERR	9 ERR
PROTECTED	(0,1, -) ERR	(0,1,2) ERR	(0,2,0) ERR	(0,0,1) ERR	7 ERR
TOTAL	-, 1 ERR	1, 3 ERR	3, 2 ERR	5, 1 ERR	

Tabla 6: resumen de resultados de la primera altura.

- En cuanto a la comparativa entre las propuestas apantalladas, tal y como se ve en la Tabla 6, hay resultados bastante parejos. Hay buenos resultados generales en todas las configuraciones, aunque la configuración que obtiene un mejor rendimiento es la Romboidal, ya que consta de solamente 1 error en su versión apantallada. La siguen la configuración Cuadrada, con 2 errores, y finalmente la Inclínada, con 3 errores.
- Comparando las configuraciones propuestas con la Original, se observa que la Original y la Romboidal obtienen el mismo número de errores, uno en este caso; son las configuraciones que tienen menos errores. Se debe tener en cuenta que en la prueba a 100cm, la configuración Original obtiene unos resultados incongruentes, que se consideran como un outlier y por tanto no se incluyen.
- Se observa que a medida que el robot se aleja de los tags en el eje horizontal “y” se obtienen peores resultados. En concreto, a 40 cm se obtienen 0 errores, a 70cm se obtienen 4, y a 100 cm se obtienen 3. Por tanto, se percibe en esta prueba que el mejor resultado se obtiene en la menor distancia.
- Se puede observar que proteger las antenas reduce el número de errores, en este caso un 33% respecto a las pruebas sin proteger⁶; en concreto, se obtienen 7 errores apantallando, y 9 sin apantallar. Cabe destacar que este calculo es general, pero que en la configuración Inclínada se obtienen más errores en su versión apantallada (3 errores) que en su versión sin apantallar (1 error).

⁶ En el cálculo se omite la configuración Original, que solo tiene la versión con las antenas apantalladas.

Análisis 2 - Altura Media:

Para esta nueva prueba, a una altura de 74cm a 192.5cm, se muestran los resultados resumidos en la Tabla 7.

	ORIGINAL	INCLINADA	CUADRADA	ROMBOIDAL	TOTAL
NO-PROT.	-	(0,0,6) ERR	(5,5,11) ERR	(5, 0,2) ERR	34 ERR
PROTECTED	(0,0,0) ERR	(2,0,4) ERR	(3,1,0) ERR	(1,1,3) ERR	15 ERR
TOTAL	0 ERR	6, 6 ERR	21, 4 ERR	7, 5 ERR	

Tabla 7: resumen de resultados de la segunda altura.

- En cuanto a los errores entre las propuestas apantalladas, se dan más fallos que en la primera altura, pero buenos resultados generales. La configuración con menos errores es la Cuadrada, con 4 errores, seguida de la Romboidal con 5 y la Inclínada con 6.
- La configuración Original no obtiene errores en esta segunda altura, dando un resultado superior a las demás.
- Se obtienen unos resultados peores que en la primera prueba, observando que el número de errores en las versiones apantalladas se dobla (de 7 errores a 15). Se empieza a observar que a mayor altura, peores lecturas se obtienen, ya que el robot se aleja tanto vertical como horizontalmente de su objetivo de lectura.
- En cuanto a las observaciones de distancia, los mejores resultados se obtienen a 70 cm (2 errores), seguido de los resultados a 40 cm (6 errores), y finalmente la 3a distancia a 100 cm es la que da peores resultados (7 errores). Esto puede ser debido a las siguientes razones:
 - Los rebotes de los rayos eran más favorables en la 2ª prueba que en las otras dos.
 - La corta distancia entre el robot y los tags cuando éstos se encuentran a mayor altura no es favorable.
Esta hipótesis tiene como base el rango de la antena, que obtiene su mayor rango de detección al llegar a la mitad de su globo de detección. Por tanto, al estar los tags a mayor altura, se debe dejar un espacio entre el robot y los tags.
- Se sigue observando una mejora muy substancial al proteger las antenas (un 54,5% menos de errores respecto a las protegidas). En concreto, se obtienen 34 errores sin apantallar, y 15 apantallando.
- Cabe destacar la posibilidad de que el resultado de la configuración Cuadrada sin apantallar sea un *outlier*, ya que se obtiene un número de errores sesgado en comparación con las demás configuraciones sin apantallar. Sin embargo, no se tienen pruebas suficientes como para afirmar que sea un *outlier*, y por tanto no se considera como tal.

Análisis 3 – Altura elevada:

Los resultados de esta última prueba, de 132 cm a 250,5 cm, se observan en la Tabla 8 siguiente.

	ORIGINAL	INCLINADA	CUADRADA	ROMBOIDAL	TOTAL
NO-PROT.	-	(3,4,3) ERR	(0,7,5) ERR	(2,5,0) ERR	29 ERR
PROTECTED	(0,0,0) ERR	(4,3,6) ERR	(2,3,1) ERR	(1,2,3) ERR	25 ERR
TOTAL	0 ERR	10, 13 ERR	12, 6 ERR	7, 6 ERR	

Tabla 8: resumen de resultados de la tercera altura.

- En esta prueba, las propuestas apantalladas que mejores resultados ofrecen son la Cuadrada y la Romboidal, obteniendo el mismo número de errores (6 errores). En la Inclínada se obtiene un resultado malo, ya que su número de errores es más de el doble que en las otras 2 configuraciones (13 errores).
- La configuración Original sigue sin obtener ningún error de lectura en las pruebas, por lo que se ve que su elevación es buena para leer a grandes alturas.
- Los resultados siguen empeorando en comparación a las dos alturas anteriores. Se obtiene un total de 10 errores más en las configuraciones apantalladas de esta prueba que en la anterior. En cuanto a distancia, a 40 cm se obtienen 7 errores; a 70cm, 6 errores; a 100cm, 10 errores. Al igual que en la prueba 2, se observa que la mejor distancia es la de 70cm.
- Se sigue viendo una disminución de los errores al proteger las antenas (-13.8% respecto a las pruebas sin proteger). No obstante, se sigue observando como en la configuración Inclínada se siguen obteniendo más errores en su versión apantallada.

Conclusiones prueba de altura

Se pueden concluir los siguientes puntos:

- Se comprueba que siempre se deben proteger las antenas, ya que en casi la totalidad de las pruebas se obtienen mejores resultados en las configuraciones protegidas.
- Como era de esperar, como más alejados están los tags en altura del robot, peor es la calidad de lectura de las configuraciones propuestas.
- La distancia corta (40cm) parece ser la mas adecuada para alturas bajas, y la distancia intermedia (70cm) para alturas más elevadas, ya que en la primera altura se obtienen menos errores a 40cm, y en las otras dos hay menos fallos a 70cm. Esto es debido al ángulo de incidencia de la onda en el tag, que varía en función de la altura y distancia. Generalmente un ángulo de incidencia en perpendicular al tag facilita la lectura del mismo.
- Ninguna configuración es superior a las demás en esta prueba.
 - La configuración Romboidal obtiene unos mejores resultados en la primera altura. En la segunda la Cuadrada, y en la tercera hay un empate entre la Romboidal y la Cuadrada. Se puede considerar un empate entre estas dos configuraciones en esta prueba.
 - La configuración Inclínada obtiene los peores resultados.
 - Por otro lado, la configuración Original obtiene unos resultados mejores en todos los casos que las propuestas, lo que ya era de esperar por su elevada altura, que parece ser la clave de sus buenos resultados.
- La configuración Inclínada apantallada queda siempre por detrás de las configuraciones Cuadrada y Romboidal, y es la única configuración que demuestra peores resultados en su versión apantallada que en su versión sin apantallar.

Esto puede ser debido a que el apantallamiento de las configuraciones cúbicas se hace de la misma manera que la configuración Original, con una mejor calidad de construcción y protección. En la Inclínada, al hacerse el apantallamiento con papel de aluminio, es muy probable que no se hiciera una protección efectiva, y por ese motivo ha quedado por detrás. Se probó un apantallamiento profesional, pero fue descartado debido a la obtención de unos resultados todavía peores. Cabe contemplar pues la posibilidad de que el apantallamiento, debido a la distribución de las antenas, restrinja el ángulo de lectura de las mismas, al quedar las inclinadas por debajo de las verticales.

Conclusión final de resultados:

En general, se pueden resumir las pruebas con las siguientes conclusiones:

No se obtiene una mejor configuración en cualquier caso, si no que cada una destaca en situaciones distintas. En este caso, en la prueba de entorno, muestran un mejor rendimiento las configuraciones Inclínada y Cuadrada, mientras que en las pruebas de altura se dan mejores resultados en la Cuadrada y la Romboidal. En el esquema de la Fig. 37 siguiente se puede observar visualmente. Las configuraciones que destacan en una prueba u otra son la Inclínada y la Romboidal, mientras que la Cuadrada y la Original destacan en ambas.

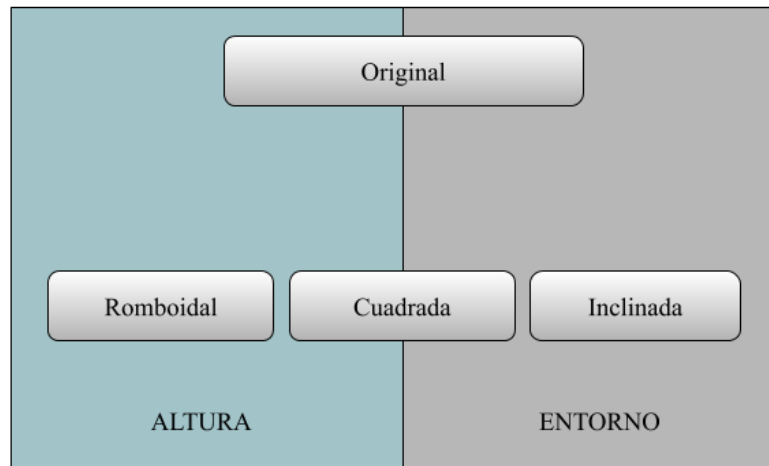


Fig. 37: esquema de configuraciones destacadas en cada prueba.

En concreto, para cada configuración se puede extraer que:

- En la Inclínada se obtienen muchos errores en la prueba de altura, debido a sus múltiples errores en la versión apantallada, que empeoran los resultados en vez de mejorarlos. Al haber la posibilidad de que las antenas verticales tapen las inclinadas, puede ser arriesgado elegir esta configuración sin estudiarlo con más detenimiento. Por otro lado, como se ve en los resultados 2.2 de entorno, la inclinación de las antenas laterales no ha producido que éstas lean más que las otras, y la configuración obtiene unos buenos resultados en la prueba de entorno, destacando por encima de las demás.
- La configuración Romboidal es inestable en cuanto a resultados, como muestra sobre todo en la prueba de entorno, percibiéndose mucha diferencia entre las dos iteraciones. Por otro lado, en la prueba de altura obtiene un buen resultado. Su inestabilidad en la prueba de entorno hace que no se pueda escoger actualmente; se deberían hacer más pruebas para comprobar si la inestabilidad de la prueba de entorno es puntual o no.
- La Cuadrada parece ser la configuración más polivalente, ya que destaca en ambas pruebas, obteniendo resultados buenos, y que quizás podrían haber sido mejores en caso de que la hipótesis de que había una antena mal conectada en la prueba de entorno fuese cierta.

Se decide que finalmente se elegirá una configuración Cuadrada, que es la que obtiene unos mejores resultados (tanto la Original como la Cuadrada propuesta, que es la más

polivalente de las propuestas). Para simplificar el modelo, se omitirá la leve inclinación a la que se sometían las antenas de la configuración Cuadrada propuesta, ya que no se obtiene evidencia de que influya en la calidad de las lecturas y la inclinación máxima de 84° que se puede conseguir es ínfima, aparte de que aumenta la complejidad del prototipo.

En cuanto a al apantallamiento y a la altura a la que situar las antenas, que definirá la altura del prototipo definitivo, se puede concluir que ambos son factores clave en la calidad de las lecturas. Los resultados de las configuraciones apantalladas son mejores que las otras, y la altura ha demostrado ser el factor diferencial para obtener unos resultados óptimos. Por otro lado, también cabe recordar que, a mayor altura, peor es la estabilidad del robot. En conclusión, en el prototipo definitivo se mantendrá el apantallamiento de las antenas, lo cual ya era un factor que se iba a tener en cuenta, y por otro lado se situará la altura del robot en un punto intermedio entre el modelo Original y los 30 cm de los modelos testeados. En concreto, la configuración elegida se situará a una altura de unos 60 cm, para tratar de obtener un punto intermedio entre capacidad de lectura y altura del prototipo, concepto que puede ser observado en la Fig. 38 de la página siguiente.

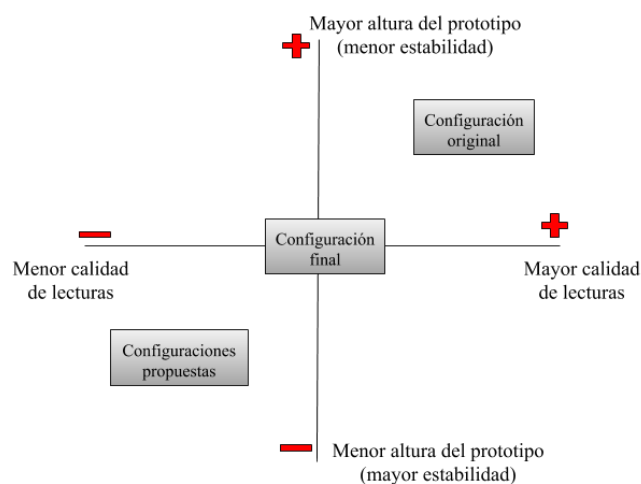


Fig. 38: esquema altura - calidad de lecturas

En la Fig. 38 se observa como la configuración Original está desviada hacia una mejor calidad de lecturas, pero con una altura demasiado elevada, mientras que las configuraciones propuestas, al tener una altura demasiado baja, obtienen también una peor lectura. En la configuración final se buscará un punto intermedio entre estas dos para obtener una calidad de lecturas equilibrada.

2.2 Adaptación de la fuente de alimentación

En este segundo apartado se desarrolla la supresión de la batería externa del robot, así como un análisis y caracterización de la batería de la Kobuki base.

a) Hipótesis y propuesta

En la sección “1.3 Hipótesis y propuestas de mejora”, se plantea la idea de suprimir la batería externa y que los componentes del robot se alimenten únicamente con la batería del Kobuki Base (hipótesis 3).

El cambio soluciona los problemas que causa el uso de la batería externa, como por ejemplo el hecho de que se enciende y carga independientemente de la base, lo que dificulta su inclusión en un diseño final. Asimismo, su supresión proporciona grandes ventajas en el diseño final, como:

- Evitar el punto de error del sistema creado por la batería.
- Reducir el peso del producto.
- Ahorrar el coste de 150€ de la batería.

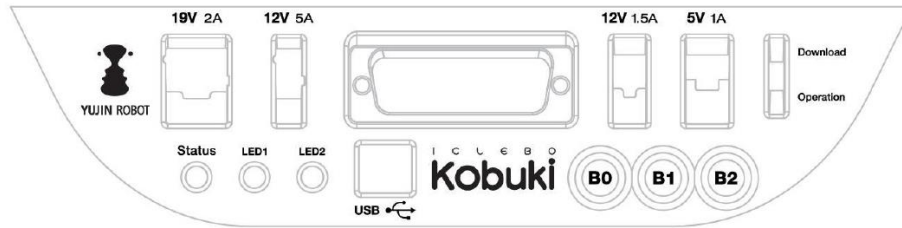
Para poder suprimir la batería, se debe hacer un análisis de ésta y considerar los aspectos clave que pueden posibilitar el cambio. Éstos son:

- Garantizar suficientes conexiones de alimentación disponibles en la Kobuki Base, las cuales se pueden observar en la Fig. 39.
- Adaptar el esquema de conexiones actual a uno nuevo con las fuentes de alimentación disponibles de la Kobuki Base.
- Compatibilizar los valores de voltaje y amperaje de las nuevas conexiones con los requerimientos de potencia de los componentes que van conectados a la batería externa (láser y Raspberry). Este punto va estrechamente relacionado con el anterior.
- Asegurar que la autonomía no se disminuya a menos de 2 horas.

b) Desarrollo de la solución

Adaptación del esquema de conexiones

Para adaptar el sistema de conexiones con los conectores de la base, primero se evalúan las conexiones de alimentación disponibles de la Kobuki base. En la Fig. 39 se muestran todas las conexiones. La conexión de 19 V 2 A solo está disponible con la base cargando, ya que es una conexión especial para conectar un portátil, y por tanto no se puede utilizar. La conexión USB-A se usa para el envío de datos de la Raspberry a la base, y la conexión de 12 V 5 A para la alimentación del *reader*. Por tanto, las otras dos conexiones (12 V 1.5 A, 5V 1 A) están libres para su posible uso.



- 19V/2A: Laptop power supply

Fig. 39: conexiones disponibles que proporciona el Kobuki Base.

Se decide alimentar la Raspberry con los 12 V 1.5 A, ya que para ésta se recomiendan 5 V de voltaje y 2-3 A para conectar periféricos, y la conexión de 5 V 1 A no cumple esta restricción. Con la primera de 12 V 1.5 A, usando un conversor a 5 V 3 A, se puede llevar a cabo la adaptación.

El láser, anteriormente alimentado con una entrada de 5 V y 2 A, ahora se debe alimentar con la fuente de 5 V 1 A del Turtlebot. Por tanto, se debe asegurar que éste pueda funcionar a 1 A de corriente. Tras la búsqueda de sus especificaciones, se observa que éste puede consumir un máximo de 0.48 A y, por tanto, se puede alimentar con 5 V 1 A. En la Fig. 40 se observa una captura del Data Sheet del láser, donde se observan sus parámetros eléctricos de consumo.

CHART2 YDLIDAR X4 ELECTRICAL PARAMETER					
Item	Min	Typical	Max	Unit	Remarks
Supply voltage	4.8	5	5.2	V	Excessive voltage might damage the Lidar while low affect normal performance
Voltage ripple	0	50	100	mV	Excessive ripple affect normal performance
Starting current	400	450	480	mA	Higher current required at start-up
Standby current	280	300	340	mA	System idle, motor rotation
Working current	330	350	380	mA	System work, motor rotation

Fig. 40: extracto del Datasheet del láser [26].

Por último, se introduce también la inclusión de otro conversor DCDC, ya que se decide cambiar la versión del *reader* por una más reciente. En concreto se cambia el AdvanReader 150.03 por el AdvanReader 160. El nuevo *reader* funciona a 24 V en comparación de los 12 V del primer modelo y, por tanto, se necesita un conversor de 12 V a 24 V para posibilitar el cambio.

A continuación, se observa en la Fig. 41 la adaptación del esquema de conexiones original (disponible en la Fig. 16) suprimiendo la batería externa, para poder observar de manera esquemática como debe llevarse a cabo la adaptación.

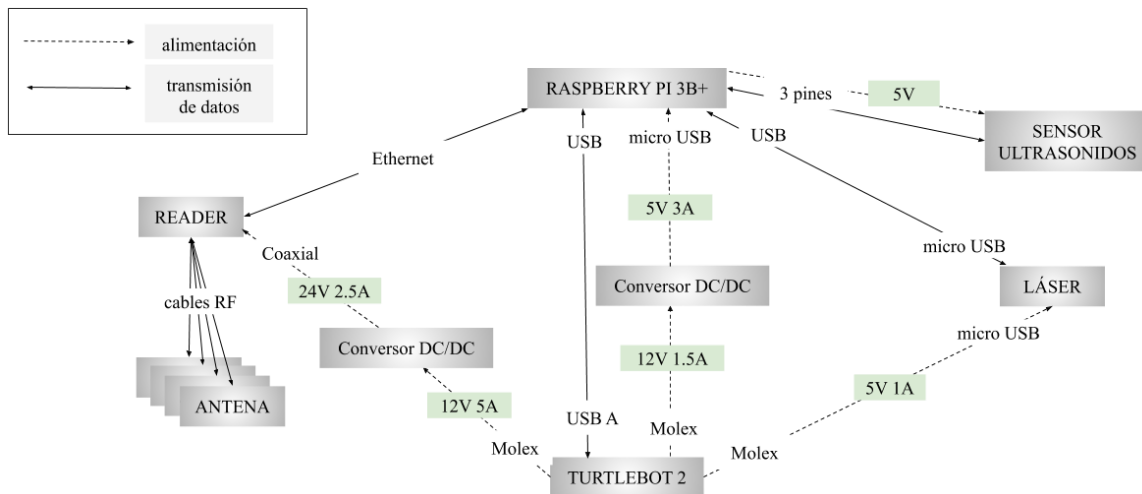


Fig. 41: propuesta del nuevo esquema de conexiones.

En esta figura se observa la supresión de la batería, anteriormente situada en la parte inferior derecha de la imagen. Se incluyen 2 nuevos componentes (los dos convertidores) y se crean 3 nuevos enlaces:

- Enlace de alimentación entre el Turtlebot y la Raspberry, de un conector Molex a un micro USB, pasando por un nuevo componente (un convertor DCDC) para pasar de los 12 V 1.5 A que proporciona la toma del Turtlebot a 5 V 3 A.
- Enlace de alimentación entre el Turtlebot (Molex) y el láser (micro USB) a través de la toma de 5 V 1 A de la base.
- Adaptación de la relación entre el Turtlebot y el *reader*, incorporando un convertor de 12 V a 24 V.

Adaptación de las conexiones

Tras adaptar el esquema de conexiones y asegurar que el cambio es posible a nivel teórico y técnico, se procede a la compra de los componentes necesarios para la adaptación del hardware. Se adquieren:

- Un convertor DCDC de 12 V 1.5 A a 5 V 3 A. Se busca la facilidad de un micro USB en la salida del convertor para facilitar el montaje en la Raspberry Pi, como puede observarse en la Fig. 42.
- Un cable de USB a micro USB (Fig. 44).
- Adaptadores para las conexiones Molex (Fig. 45, Fig. 46) de la entrada del convertor y de la entrada del cable que se conectará al láser, ya que las conexiones de salida del Turtlebot son todas Molex.



Fig. 42: conversor DCDC 12 V a 5 V 3 A [27].



Fig. 43: conversor DCDC 12 V a 24 V [28].



Fig. 44: cable USB a micro USB [29].



Fig. 45: adaptador Molex [30].



Fig. 46: adaptador Molex [31].

Para la adaptación del sistema, se siguen una serie de pasos. En la Fig. 47 se observan los puntos a seguir para la adaptación de un conector Molex, y en la Fig. 48 los pasos para la adaptación del sistema, usando lo explicado en la Fig. 47.

Adaptación de un cable a conector Molex:

- 1) El cable de la Fig. 44 se pela por el extremo del USB, dejando solamente los cables positivo y negativo para adaptarlos al conector Molex.
- 2) El metal de la Fig. 45 se gripa con el lado positivo y negativo del cable crudo.
- 3) Se introducen los dos extremos del cable dentro de la pieza de plástico de la Fig. 46. De esta manera, queda un cable con un micro USB en un extremo y un conector Molex en el otro.

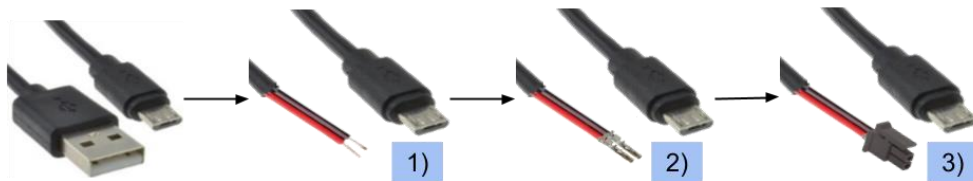


Fig. 47: adaptación de un cable Molex. Pasos 1), 2) y 3).

En la Fig. 47 se observa en detalle la adaptación del cable USB a un conector Molex, que se utiliza para el láser.

Adaptación de los componentes:

- 1) El convertor DCDC del *reader* se adapta con un conector Molex en un extremo (como se ha visto en la lista de pasos anterior) y el coaxial del *reader* en el otro. Después se conecta el cable coaxial al *reader* y el Molex a la Kobuki base.
- 2) Para el convertor DCDC de la Raspberry, se adapta un conector Molex en el extremo de cable pelado del convertor y se conecta con el Kobuki base. El otro extremo (con el conector micro USB) se conecta a la Raspberry.
- 3) Para el láser, se adapta un cable de micro USB a Molex, y se conecta el extremo del micro USB a la alimentación del láser y el Molex al Kobuki base.

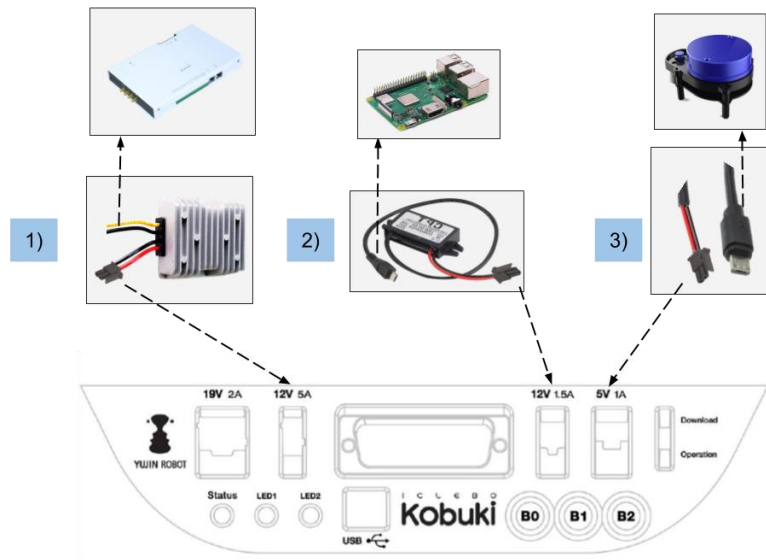


Fig. 48: adaptación de los componentes con conexiones Molex. Pasos 4), 5) y 6).

En la Fig. 48 se observa el ajuste de las conexiones de los componentes para la adaptación del sistema sin la batería externa.

c) Testeado y análisis

Se testea la batería del Kobuki Base alimentando todo el sistema. Se lleva a cabo una prueba con dos repeticiones, que después se analizan.

La prueba consiste en hacer girar al robot sobre sí mismo leyendo hasta que se apague. La primera repetición se lleva a cabo a una velocidad de rotación de 0.3 rad/s, que se puede considerar una velocidad media, y la segunda a 0.5 rad/s, que se puede considerar una velocidad máxima a la que podría llegar el robot. Se cambia la velocidad para comprobar que no hay mucha diferencia entre resultados con velocidad variante.

Se llevan a cabo dos análisis distintos. En primera instancia, se mide la duración de la batería del robot, para asegurar que su autonomía es superior a 2 horas después del cambio de batería. En segunda instancia se caracteriza el comportamiento de la batería con más precisión, llevando a cabo una división de ésta en 4 segmentos de igual duración. Estos dos análisis son condicionantes, es decir, si en el primero no se obtiene una duración

mayor a dos horas, no se lleva a cabo el segundo, ya que se consideraría que la autonomía no es suficiente y se debería buscar otro método de alimentación.

Análisis 1: duración de la batería

En la primera repetición se consigue una duración de 200 minutos, mientras que en la segunda se obtienen 193 minutos, un poco menos debido a la mayor velocidad de rotación. Se debe destacar que en una prueba real el robot probablemente obtendría mayor autonomía, ya que éste normalmente no se mueve continuamente sin parar, si no que va haciendo pausas.

Se puede considerar que los resultados obtenidos están dentro del margen que se considera válido para llevar a cabo el cambio de batería, ya que se obtiene una autonomía aproximada de 3 horas.

Por tanto, se decide suprimir finalmente la batería externa y alimentar todos los componentes con la batería integrada del Kobuki base, ya que los beneficios que proporciona su supresión son mucho mayores que la reducción de batería del robot. Al validarse los resultados de este primer análisis, se procede al segundo.

Análisis 2: división en 4 segmentos

Se caracteriza con más precisión la batería de la Kobuki base, dividiéndola en 4 segmentos de igual duración.

Durante las dos repeticiones, se hace una suscripción al *topic* de ROS encargado del estado de la base, obteniendo de esta manera un mensaje cada 40Hz. Estos mensajes tienen el nombre *status*, y su estructura se puede observar en la figura inferior:

```
header:
  seq: 1448
  stamp:
    → secs: 1580220819
      nsecs: 765472463
    frame_id: ''
time_stamp: 60804
bumper: 0
wheel_drop: 0
cliff: 0
left_encoder: 0
right_encoder: 0
left_pwm: 0
right_pwm: 0
buttons: 0
charger: 0
→ battery: 163
bottom: [1476, 1718, 1538]
current: [0, 0]
over_current: 0
digital_input: 0
analog_input: [4095, 4095, 4095, 4095]
```

Fig. 49: output de la llamada de status a la Kobuki Base.

Se recibe el status constantemente durante ambas repeticiones y se guarda el mensaje en un documento de texto. A continuación, una vez el robot agota su autonomía, se pasa un filtro al archivo para retener solamente los datos referentes al intervalo de tiempo, y los referentes a la batería (señalados en la Fig. 49 superior). El intervalo de valores de batería va de 163 como máximo, a 97 como mínimo antes de que el robot se apague. Estos valores no referencian ninguna unidad de medida.

Se muestran los resultados extraídos en el Gráfico 5 y Gráfico 6, donde se puede ver el progreso de la batería desde su inicio hasta su agotamiento en ambas pruebas:

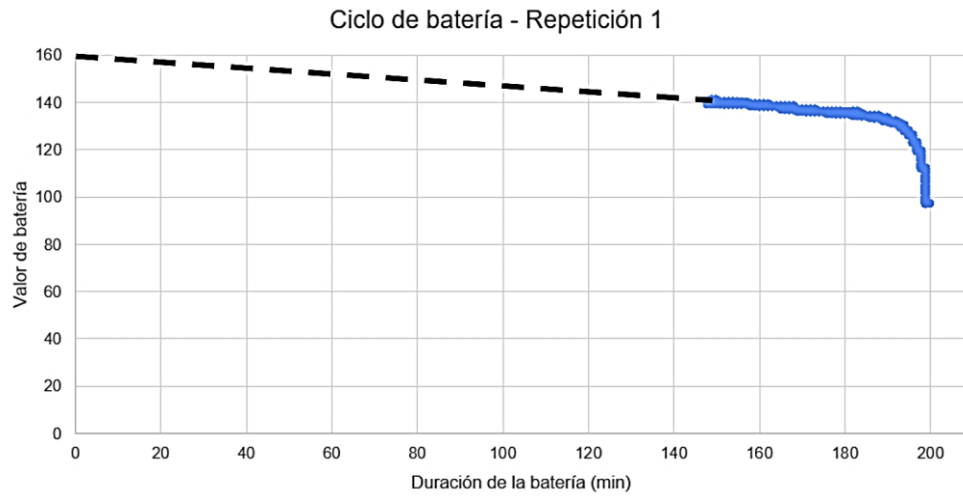


Gráfico 5: primera prueba de batería con menor velocidad (0.3 rad/s).

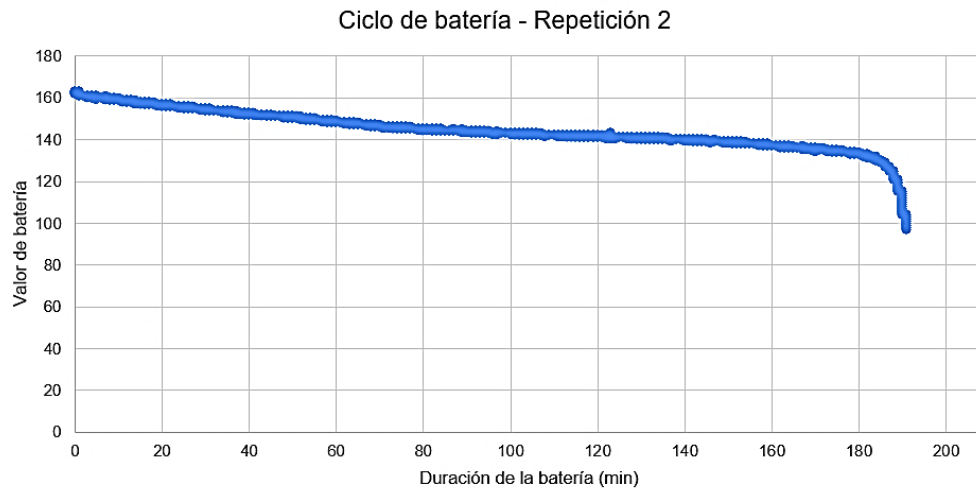


Gráfico 6: segunda prueba de batería con mayor velocidad (0.5 rad/s)

En ambos gráficos se percibe como la batería no decrece linealmente, si no que sigue una función exponencial decreciente ($f(x) = -e^x$).

En el caso de la repetición 1 (Gráfico 5), se observa una línea negra discontinua hasta que el valor de batería es 140, en el minuto 148. Esto es debido a que se empiezan a recoger datos más tarde debido a un error, pero la prueba se considera válida de igual manera, ya que permite recoger la duración total. La línea discontinua marca la estimación de los valores de batería, a partir de la tendencia observada en la segunda repetición.

En la repetición 2 (Gráfico 6), se percibe como la batería decrece con más rapidez, sin llegar a los 200 minutos de la repetición 1. En este segundo caso se empiezan a grabar los valores de batería desde el principio, y se obtiene el espectro total de valores, que va de 163 a 97.

La duración de la batería se divide en 4 segmentos, repartiendo los valores [163-97] en 4 partes de duración aproximada, observando el segmento de tiempo en el que el valor de

la batería es muestreado en la segunda prueba. En concreto se lleva a cabo una segmentación en los siguientes puntos:

- 163-151 → correspondiente al intervalo de batería alta (100% - 75%).
- 150-143 → correspondiente al intervalo de batería media (75% - 50%).
- 142-139 → correspondiente al intervalo de batería baja (50% - 25%).
- 138-97 → correspondiente al intervalo de batería crítica (25% - 0%).

De esta manera, quedan 4 intervalos con una duración aproximada de 48 minutos cada uno.

2.3 Distribución interna de componentes

En la sección “1.3 Hipótesis y propuestas de mejora”, se plantea la idea de mejorar la distribución de componentes del sistema contemplando un diseño modular (hipótesis 4).

En este punto se desarrolla el estudio de esa hipótesis, elaborando un análisis de cómo deben disponerse los componentes en el interior del robot.

a) Requerimientos

Se consideran los siguientes aspectos y restricciones:

- El robot debe tener una altura final de unos 60 - 65cm, un punto intermedio entre los 30cm de las pruebas llevadas a cabo y los 110cm del prototipo original⁷. De esta manera se consigue una altura intermedia, pero el robot será capaz de pasar por debajo las mesas sin chocar, ya que éstas suelen tener una elevación de 70-75cm.
- El láser se debe situar en un punto donde los otros elementos del sistema obstaculicen lo menos posible su visión.
- El peso debe estar distribuido de manera que se baje el centro de gravedad lo máximo posible.
- El *reader* y las antenas se deben situar lo más próximos posible, ya que no es bueno para la tecnología RFID que los cables que conectan el *reader* con las antenas sean muy largos.
- Los componentes deben ser accesibles para un posible futuro mantenimiento del robot.
- El sensor de ultrasonidos se debe situar en una altura intermedia, asegurando la detección de la parte superior del robot. Por otro lado, su visión no puede ser obstruida por ningún material.

⁷ Según las conclusiones obtenidas en “2.1 Disposición y altura de las antenas”

- Los componentes internos deben protegerse de alguna manera. Más adelante (pág. 67) se desarrolla este apartado.
- La modularidad del sistema se debe contemplar. Se explica en el siguiente apartado.

b) Modularidad

Es importante que el sistema sea modular, ya sea parcial o totalmente, ya que la modularidad ofrece muchos beneficios, algunos de ellos mencionados en [32], [33]:

- Reduce la complejidad del sistema, permitiendo abordar problemas por separado y mejorar de manera independiente diferentes aspectos de diseño. Los módulos ocultan sus detalles, y solo muestran públicamente sus capacidades y conexiones.
- Reduce los tiempos de producción y ensamblaje, haciéndolos más cortos que los de una construcción típica.
- Reduce los costos de producción.
- Posibilita la combinación de distintos componentes preexistentes.
- Facilita la personalización para cada cliente y adaptación al entorno requerido.
- Facilita la reparación del robot, pudiendo atajar la avería por módulo.

Por tanto, contemplando los requerimientos expuestos, se propone una configuración del sistema en dos módulos: un módulo RFID y uno de Navegación.

- Módulo RFID: compuesto por la carga útil del robot, los componentes RFID. Específicamente, las antenas y el *reader*.
 - o Encargado de la lectura de los tags.
 - o Extraíble de manera sencilla.
 - o Posibilidad de intercambiar diversos módulos RFID, lo que permite cambiar la configuración de las antenas sin afectar al bloque de navegación.
- Módulo de Navegación: compuesto por la base robótica móvil. En concreto la Raspberry, el sensor de ultrasonidos, el láser y el Turtlebot.
 - o Encargado de la correcta navegación del robot.
 - o El sensor de ultrasonidos se sitúa a media altura, entre los dos módulos, pero anclado al bloque de navegación.

Las conexiones entre módulos son sencillas. Se obtienen dos relaciones, una ethernet de la Raspberry al *reader* para datos, y otra del Turtlebot hacia el *reader* para alimentación. Este concepto se observa en la Fig. 50 inferior, donde se observan los dos módulos, con sus respectivos componentes, y las dos relaciones que existen entre ellos.

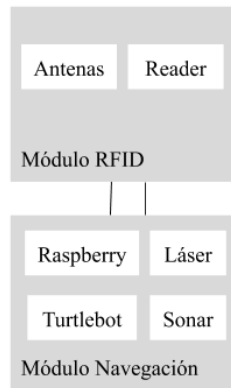


Fig. 50: estructura de módulos, con las salidas de cada uno.

2.4 Construcción de un nuevo prototipo

Se propone un diseño para la construcción de un nuevo prototipo (su estructura interna) y la distribución de componentes dentro de éste, y se lleva a cabo la construcción y montaje del mismo.

a) Características

Se elabora una propuesta de diseño preliminar para el nuevo robot, cuyas características son extraídas de:

- Las conclusiones del punto 2.1, que determinan la configuración de las antenas y la altura del prototipo.
- Las conclusiones del punto 2.2, que fijan la supresión de la batería externa y el esquema de conexiones óptimo.
- Los requerimientos del punto 2.3, que fijan la distribución interna de componentes.

El resumen de características de la propuesta es el siguiente:

- Altura de 60cm.
- Configuración cuadrada para las antenas.
- Alimentación del sistema con la batería integrada de la base.
- Optimización de la distribución de los componentes, bajando el centro de gravedad.

- Separación del sistema en dos módulos, facilitando el desacoplamiento de éstos a través de un anclaje de rápido desmontaje.

Se crea un diseño rápido a escala con Tinkercad que cumpla con las características, disponible en la figura inferior:

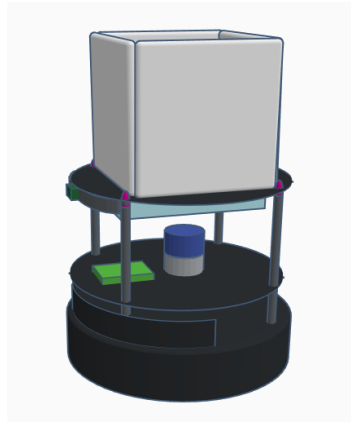


Fig. 51: propuesta de prototipo.

En la Fig. 51 se observa en azul oscuro el láser; en verde claro, la Raspberry; en verde oscuro, el sensor de ultrasonidos; por último, el *reader* se sitúa de color azul claro, anclado a la parte superior, y las antenas de color gris. Se observan 4 tornillos de color lila. Éstos están pensados para ser un anclaje rápido para poder separar de manera rápida el módulo RFID.

b) Construcción del Prototipo

Se lleva a cabo el diseño definitivo con AutoCAD a escala con todos los componentes para la impresión 3D.

Se llega a esta versión en base a las sugerencias hechas al diseñador industrial. Hay una serie de cambios respecto a la propuesta. Éstos son:

- Ahorro en materiales, creando agujeros donde el material no es necesario.
- El *reader* finalmente se sitúa de manera vertical, ya que horizontalmente no cabe correctamente en el interior de la estructura debido a los conectores. De esta manera, además, se facilita la conexión de las antenas con el *reader*, dejando las conexiones de éste más cercanas a las antenas. En contraposición, el peso se reparte un poco peor.
- Se crea un asa en la parte superior, formada por un puente de metal, para mejorar el agarre y manejo del prototipo.

El diseño se puede observar en las figuras siguientes:

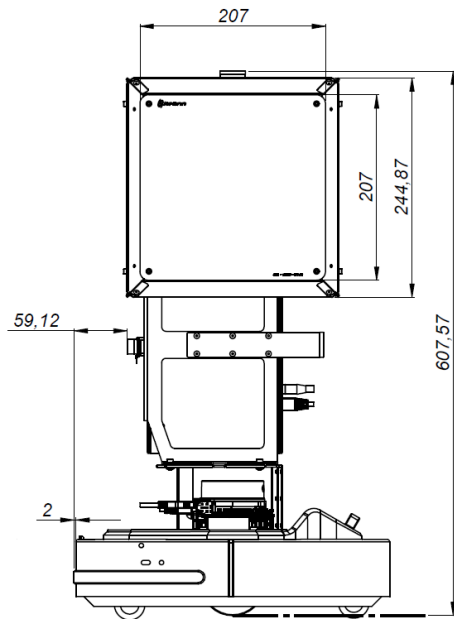


Fig. 52: perspectiva lateral en 2D del diseño definitivo, con detalles de medidas.

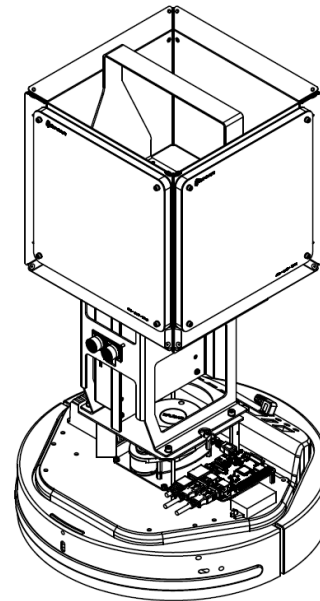


Fig. 53: renderizado 3D del diseño final. Vista isométrica.

En la Fig. 52 se observa la perspectiva lateral del robot, con algunas medidas. Las dimensiones de la antena son $20.7\text{cm} \times 20.7\text{cm}$, aunque al apantallarlas ocupan un total de $23.6\text{cm} \times 24.5\text{cm}$. El sensor de ultrasonidos se sitúa a una altura de 30cm desde la base. El robot consta de una altura total de 60.76cm de altura.

En la segunda Fig. 53 se observa el renderizado 3D del robot, donde se puede percibir el asa en la parte superior, así como la Raspberry y el conversor pequeño en la parte izquierda del robot. El conversor grande para el *reader* se sitúa a la derecha.

En la siguiente figura se observa un explosionado 3D del diseño:

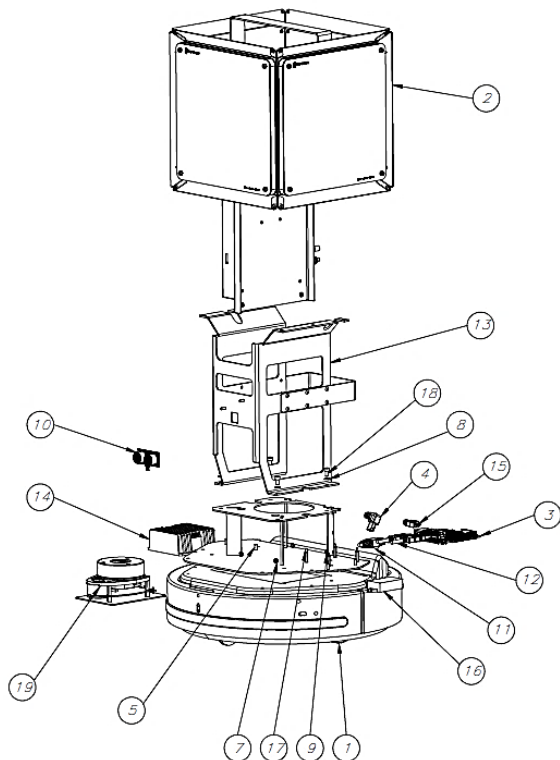


Fig. 54: explosionado 3D del prototipo final

N.º DE ELEMENTO	N.º DE PIEZA	CANTIDAD
1	Turtlebot-base.02	1
2	ASM-RFID.03	1
3	raspberry_pi_3_b_plus	1
4	USBFIREWIRE_COM-BR	1
5	PENC-90400-AL239005.01	1
6	ASM-Laser-X4.01	1
7	PNBS-WSH-D127B-M3-A2	4
8	PNBS-WSH-D127B-M4-A2	4
9	PNBS-NUT-D934-M03-A2	4
10	ping-ultrasonic-range-finder	1
11	RJ45-male-Kabel	1
12	USB-male	2
13	PENC-90403-AL239005.01	1
14	Converter-12V-to-24V-240W	1
15	USBFIREWIRE_COM-MCBS	1
16	Converter-12V-to-5V-3A-15W	1
17	PNBS-STF-MF-M2x4x15-MS	4
18	PNBS-SCR-D912-M4x8-A2	4
19	ASM-Laser-RPLIDAR.01	1

Fig. 55: especificación de piezas del explosionado.

En la Fig. 54 se puede observar el detalle de los componentes del prototipo final. Se observan componentes como el láser (19), el convertor para el *reader* (14), el sensor de ultrasonidos (10), la Raspberry (3) o el convertor pequeño (16). La estructura señalada con el número 13 sostiene y separa el módulo RFID del módulo de navegación.

Una vez elaborado el diseño, se manda imprimir el 3D del modelo, y se procede al montaje del prototipo.

A continuación, se pueden observar algunas imágenes del prototipo construido:



Fig. 56: vista lateral del prototipo.



Fig. 57: vista frontal del prototipo.

En la Fig. 56 se observa una vista lateral del robot, donde se aprecian las conexiones del robot a la base. En la Fig. 57, se enseña la vista frontal, percibiéndose el sensor de ultrasonidos en una altura intermedia, con la Raspberry y el conversor pequeño en la parte izquierda y el conversor del *reader* en la derecha. El peso total del prototipo construido es de 6,6 Kg, un peso muy reducido en comparación con los robots RFID del mercado actual, que se recuerda tienen un peso de unos 45 Kg de media.

2.5 Presupuesto

Se recoge el presupuesto de todas las piezas necesarias para la construcción del robot, junto con la mano de obra, para contrastar su económico coste de construcción en comparación con los robots RFID actuales, que tienen un precio de venta de 30.000 a 60.000€. Se entiende que, aunque el precio de venta se encuentra hinchado según los márgenes en los que trabaja la empresa, el coste de construcción sigue siendo mayor.

En la tabla siguiente se contempla el presupuesto de ejecución material (PEM), considerando solamente los aspectos desarrollados en el proyecto. No se incluye, por tanto, el precio de construcción de una carcasa, cuya propuesta se explica en el siguiente capítulo, y que se estima incrementaría el precio del robot unos 3.000€. Por otra parte, sí se tiene en cuenta el precio del material para notificaciones, cuyo desarrollo se explica también en el siguiente capítulo.

Cantidad	Componentes	Coste
1	Kobuki Base	499 €
1	Raspberry Pi 3B+	35 €
1	Láser YDLIDAR X4	90 €
1	Parallax 28015	27 €
1	AdvanReader 160	785 €
4	Advantenna SP11	280 €
1	Conversor DCDC 12V a 24V	21 €
1	Conversor DCDC 12V a 5V 3A	2 €
Cantidad	Cableado	Coste
4	Cable RF	8,50 €
2	Cable USB a Micro USB 0,6m	2,60 €
1	Cable Ethernet RJ45	1 €
1	Cable USB B a USB A	8 €
3	Adaptador Molex 0430250200	0,90 €
6	Adaptador Molex 0462355002	1,80 €
Cantidad	Estructura	Coste
1	ASM-RFID.03	139,25 €
1	PENC-90400-AL239005.01	60 €
1	PENC-90403-AL239005.01	47,5€
1	ASM-Laser-RPLIDAR.01	17,5 €
Cantidad	Tornillería	Coste
4	PNBS-STF-MF-M2x4x15-MS	2 €
4	PNBS-WSH-D127B-M3-A2	0,004 €
4	PNBS-NUT-D934-M03-A2	0,012 €
4	PNBS-SCR-D912-M4x8-A2	0,08 €
4	PNBS-WSH-D127B-M4-A2	0,004 €
Cantidad	Notificaciones	Coste
1	Tira leds WS2812B	25 €
Cantidad	Mano de obra	Coste
1	Coste de montaje	200 €
Total:		2.253 €

Tabla 9: presupuesto de ejecución material del robot.

En la Tabla 9, se observa el presupuesto de ejecución material, mostrando cada componente, la cantidad usada y el precio total (precio por unidad \times número de piezas). Se contemplan los materiales usados hasta el momento, así como el coste de mano de obra recurrente para la construcción de un robot. En el coste de montaje, se contempla que el robot podría ser montado y construido en 8h (una jornada laboral); por tanto, se calcula el trabajo de un ingeniero durante 8 horas a 25€/h brutos.

No se contempla en la tabla el coste de desarrollo de ingeniería, que se refiere a las horas dedicadas para el diseño del robot, y que se estimaría como el trabajo de un ingeniero 8 horas diarias durante 6 semanas a 25 €/hora brutos, que resulta en un coste de 6.000 €.

El resultado total, sumando el coste de construcción del robot mejorado (2.253 €) y el coste de ingeniería de la mejora de diseño (6.000 €) es de 8.253 € para el robot actual. Cabe indicar que este precio se incrementaría en la versión final del robot, al tener que incluir una carcasa. Por otro lado, industrializar el proceso vería reflejado un decremento considerable en el precio, y el coste de ingeniería, al tratarse de un coste puntual, sería diluido según el número de unidades.

3. DISEÑO EXTERIOR E INTERFAZ

En este capítulo se contemplan los aspectos interactivos del robot, como el diseño visual o la elaboración de un sistema de notificaciones para la comunicación robot-humano.

3.1 Diseño exterior (*look and feel*)

Con el fin de crear un diseño definitivo del robot para su comercialización, es necesario tener en cuenta su aspecto visual. Para ello, en este apartado se desarrolla la propuesta de una carcasa para el robot, así como una metodología para el estudio de estilo, llevada a cabo para crear un diseño exterior atractivo considerando aspectos funcionales. Este apartado se empieza a elaborar a nivel teórico, pero no se llega a realizar el prototipado de la carcasa.

a) Propuesta de carcasa y requerimientos

Se propone la integración de una carcasa que cubra el robot entero, para suplir algunas necesidades no solucionadas con el prototipado de la parte interna del robot. Algunas ventajas de tener una carcasa son las siguientes:

- Posibilitar la producción un producto definitivo a nivel comercial.
- Proteger los componentes internos del polvo o posibles golpes involuntarios.
- Proporcionar una capa de abstracción sobre el diseño interior para simplificar el robot a nivel visual a la vez que se obtiene un resultado más atractivo.
- Facilitar el mantenimiento del robot. Al quitar la carcasa se dejan al descubierto los componentes internos, por lo que acceder a éstos es más sencillo que si se protegiera cada uno por separado.

Se llega a los siguientes requerimientos de los que debe disponer la carcasa:

- El material de construcción que se elige es el policarbonato.
 - Es más fácil de moldear y es más resistente a impactos que otros materiales (e.g. metacrilato).
- La carcasa debe constar de dos partes.
 - Una primera parte opaca que lo cubra hasta la altura del láser.
 - Otra parte transparente que cubra el robot desde la base hasta el láser, uniéndose con la primera parte. Esta parte es necesaria para la correcta lectura del láser, ya la visión de éste no puede ser obstruida por obstáculos.
- La carcasa debe tener un pequeño agujero por el que sobresalga el sensor de ultrasonidos.
 - Éste, a diferencia del láser, no puede ser obstruido por ningún material, aunque sea transparente.

- La carcasa debe integrar unas hendiduras.
 - Éstas facilitan el manejo y transporte del robot, así como la extracción de la carcasa.
- El diseño debe ser minimalista, basado en las curvas.
 - Como señalan algunos artículos [34], un diseño curvo siempre es más fácil de entender para el ojo humano. Asimismo, los diseños con formas afiladas se asocian más con el peligro y la agresividad que los diseños curvos, que se relacionan con lo inofensivo y amigable, como se explica en [35]. Se desea un robot que pueda circular entre usuarios sin provocar incomodidad en ellos.
- En la carcasa se debe integrar la capa de comunicación del robot con el usuario.
 - Toda la información sobre este aspecto se encuentra disponible en el punto “ 3.2 Human-Robot Interface”.

Tras la fase de ideación, se lleva a cabo un diseño preliminar de la carcasa sobre el diseño interno propuesto en la Fig. 51. Éste puede observarse en la Fig. 58 inferior:

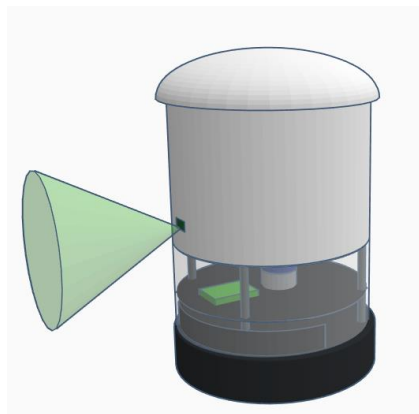


Fig. 58. propuesta de prototipo con carcasa

En ésta se puede observar el sensor de ultrasonidos sobresaliendo de la carcasa, con su respectivo rango de detección. También se percibe la propuesta para crear una hendidura, rodeando la parte superior de la carcasa para facilitar el agarre desde cualquier punto y evitar que el usuario final deba agacharse para coger el robot.

b) Estudio de estilo

Se lleva a cabo un ejercicio de diseño de estilo propuesto por José Luis Sanz, diseñador industrial de Keonn, para el estudio de los posibles diseños exteriores del robot. Este ejercicio, llamado “Si fuera”, consiste en buscar una serie de objetos diferentes a lo que se quiere diseñar, como por ejemplo mesas, cafeteras, ... y exponer aquellos que a nivel visual atraigan más la atención para el robot. Debe hacerse la pregunta siguiente: “Si el robot fuera un/una [objeto], ¿qué [objeto] sería?”. El documento relacionado con este ejercicio se encuentra en https://github.com/calilluis/TFG-RFID_Robot/estudio_estilo.

Este ejercicio ha ayudado en la elección de varias decisiones de diseño, como:

- La elección de un diseño minimalista y curvado para la carcasa.
- La inclusión de un asa para el robot.
- El uso de leds para las notificaciones.

Para la elección de un diseño para la carcasa, se han elegido objetos que cumplieran con los requisitos de un diseño curvado y liso, sin ángulos muy marcados. Se observa que la mayoría de los diseños escogidos son minimalistas y elegantes. Se pueden observar un par de ejemplos en las figuras Fig. 59 y Fig. 60 inferiores. Para la carcasa se busca un diseño similar.



Fig. 59: lámpara minimalista con asa, diseño de Miguel Milá [36].



Fig. 60: Mac Pro 2013 [37].

En la inclusión de un asa para el robot, se encuentran diversos objetos, como los que se pueden ver en las figuras siguientes, que constan de asas para mejorar su transporte o agarre:



Fig. 61: estufa Cecotec [38].



Fig. 62: carrito de compra [39].

En la decisión del uso de leds para notificación, el ejercicio demuestra que existen diversos dispositivos con objetivo minimalista que utilizan los leds para notificar; por ejemplo, algunos auriculares inalámbricos, o el asistente de voz Amazon Echo, disponibles en las figuras siguientes:



Fig. 63: auriculares inalámbricos GT1 [40].



Fig. 64: Amazon echo [41].

En la Fig. 63 se observan los auriculares inalámbricos Haylou GT1, de una submarca de Xiaomi. En estos se pueden observar 4 leds frontales para indicar la batería restante. En la Fig. 64 se muestra el Amazon Echo, que consta de una tira de leds que lo recorre circularmente, permitiendo la visualización de notificaciones en 360 grados. Este último concepto es el que se propone para las notificaciones principales del robot, mientras que se usa el concepto de los 4 leds de los auriculares para las notificaciones de batería del robot. El desarrollo detallado del uso de leds para notificaciones se encuentra en el apartado “Metodología de notificación” en la página 72.

3.2 Human-Robot Interface

Dentro del aspecto interactivo del robot, cabe destacar la interfaz entre el robot y el usuario final, conocido como *Human-Robot Interface* (HRI). Una buena interfaz es primordial para la correcta comunicación entre el usuario y el robot; el usuario debe saber que problema o mensaje está notificando el robot de manera sencilla. Por otro lado, un buen diseño de interfaz robot-humano es importante, ya que como se menciona en [42], puede afectar directamente al deseo del usuario final de utilizar el robot. Si este presenta una interacción difícil, considerada pesada para el usuario final, el robot puede llegar a no usarse.

El robot destaca por su autonomía en la navegación, como se menciona en “e) Prototipo Preliminar”. Este factor simplifica la interfaz entre el usuario y el robot respecto a los robots RFID actuales. De la misma manera, es importante que se gestione correctamente la atención que le da el usuario al robot. Eso implica que el robot debe informar de manera clara y precisa sus necesidades, señalando con precisión como proceder ante un problema, o comunicando de manera efectiva qué está fallando.

Por tanto, al necesitarse una interfaz entre el usuario y el robot, se contemplan los aspectos comentados y se desarrolla un sistema de notificaciones para la comunicación del robot con el usuario final.

a) Sistema de notificaciones

Requerimientos

Se analizan las necesidades de notificación del robot. Se elaboran una serie de requisitos, y se agrupan en 4 grupos para una mejor asociación. En concreto, se dividen en notificaciones generales, notificaciones de navegación, notificaciones de batería del robot, y notificaciones de error.

- Notificaciones generales del robot:
 - Debe notificar un inventario inminente, para que el usuario pueda ver que el robot está cumpliendo el programa/calendario marcado, y, por ejemplo, para hacer que algunas operaciones de tienda se paren y que así el robot pueda hacer inventario.
 - Debe notificar los procesos de cargado de datos o encendido del dispositivo, para proveer *feedback* sobre su estado en todo momento.
 - Debe comunicar el proceso de inventario, para evitar que se interfiera el proceso o se obstaculice su navegación.
 - Debe notificar la cercana finalización del inventario, para mantener informado al usuario sobre su estado y, por ejemplo, evitar la obstaculización de su base de carga.
- Notificaciones de incidencia en la navegación del robot:
 - Debe notificar los posibles atascos con el entorno en la navegación, por ejemplo, una caja en un lugar inesperado que lo bloquea. De esta manera el robot puede alertar que necesita ayuda externa para seguir navegando.
 - Debe notificar que no es capaz de encontrar la base de carga por sí mismo, para avisar que necesita ayuda externa para llevarlo de vuelta a la base.
- Notificaciones de batería del robot:
 - Debe notificar sus diferentes estados de batería para asegurar el correcto funcionamiento del robot y el cumplimiento de su misión. Se dividen los estados de batería en “batería cargando”, “alta” (100 % - 75 %), “media” (75 % - 50%), “baja” (50 % - 25 %) y “crítica” (25 % - 0 %).
- Notificaciones de error:
 - El robot debe notificar si sufre un error en el módulo RFID.
 - El robot debe notificar si sufre un error en el módulo de navegación.
 - El robot debe notificar si sufre un error interno desconocido.

Las notificaciones de error son importantes y necesarias para notificar al usuario si algo no está funcionando correctamente en el sistema, a la par que mejora el mantenimiento del robot y el atajo rápido de problemas

Metodología de notificación

Se valoran distintas maneras de mostrar las notificaciones, y finalmente se decide que el uso de leds puede ser la mejor opción. Se plantean otros métodos, como la inclusión de una pantalla, pero los principios de simplicidad en los que se basa el robot hacen decantar la balanza hacia un sistema de notificaciones más sencillo, aunque no por ello menos preciso.

Para las notificaciones usuales (generales y navegación), que deben ser más visibles, se decide usar una tira de leds grande para cada una, y para las notificaciones de uso inusual o de consulta (batería y errores), se deciden usar leds individuales.

Se lleva a cabo una división de leds en 4 colores según su categoría, para facilitar una mejor relación para el usuario final. Los colores consideran los siguientes aspectos: los leds de batería se disponen de color verde, usualmente usado para simbolizar la autonomía. Por otro lado, la diferencia de leds rojos y leds verdes es debida a que en Europa se relaciona el color rojo con el peligro, y el verde con la seguridad. Por otro lado, en las tiras de leds (azul y naranja), el color naranja se asocia con una notificación más inminente que una azul. El naranja es un color usual que no indica peligro ni seguridad, pero que requiere atención (e.g. en un semáforo). El color azul se asocia mejor con una notificación usual informativa que no requiere atención inmediata.

Específicamente, para cada requerimiento de notificación, se define un comportamiento diferente de la iluminación:

- **Notificaciones generales:** Tira de leds RGB azul claro.
 - Pálpito suave: Inventario inminente.
 - Oscilación vertical: Cargando datos.
 - Encendido siempre: inventario en curso.
 - Parpadeos largos (2s): inventario finalizando.
- **Notificaciones de navegación:** Tira de leds RGB naranja.
 - Encendido siempre: sistema de navegación atascado.
 - Pálpito suave: imposibilidad de encontrar la base de carga.
- **Notificaciones de Batería:** 4 leds RGB.
 - 4 palpitando en verde: cargando.
 - 4 leds verdes encendidos: batería al 100 % - 75 %.
 - 3 leds verdes encendidos: batería al 75 % - 50 %.
 - 2 leds verdes encendidos: batería al 50 % - 25 %.
 - 1 led palpitando en rojo: batería al 25 % - 0 %.
- **Notificaciones de Error:** 2 leds rojos verticales.
 - Led superior encendido: error en el módulo RFID.
 - Led inferior encendido: error en el módulo de navegación.
 - Los dos parpadeando: error desconocido.

Disposición y tipografía

Se lleva a cabo una propuesta de diseño para la disposición y tipografía de las notificaciones. En esta, se integran las notificaciones en la carcasa que cubriría el robot. Los leds se dispondrían detrás del policarbonato, consiguiendo una retroiluminación led:

se percibiría una iluminación detrás del policarbonato, aunque a simple vista los leds no serían visibles si estuvieran apagados.

Los cuatro leds de batería y los dos de errores se disponen en la parte inferior trasera del robot, ya que son notificaciones que no necesitan una visualización usual a larga distancia. Por ese motivo, se propone la inclusión de un icono representativo para que el usuario final recuerde con facilidad este tipo de notificaciones. Para la batería, se propone un símbolo de una pila, y para los errores, un símbolo de una herramienta, referenciando la necesidad de mantenimiento. Asimismo, los leds de error tendrían la forma del robot para una mejor simbología. El led superior haría referencia al módulo RFID y el inferior al módulo de navegación.

Por otro lado, las tiras de leds de notificaciones generales y de navegación se situarían en círculo alrededor del robot, ya que de esta manera las notificaciones pueden ser vistas en 360 grados. Estas notificaciones no constarían de simbología. Son notificaciones que el usuario final tiene que recordar, ya que se deben reconocer desde una cierta distancia.

En las siguientes figuras se muestra el diseño gráficamente de la propuesta tanto de los leds traseros inferiores como de los anillos superiores. En la Fig. 65 se muestra la tipografía de las notificaciones de batería y error. Por otro lado, en la Fig. 66, se observan las tiras de leds principales rodeando la parte superior del robot.

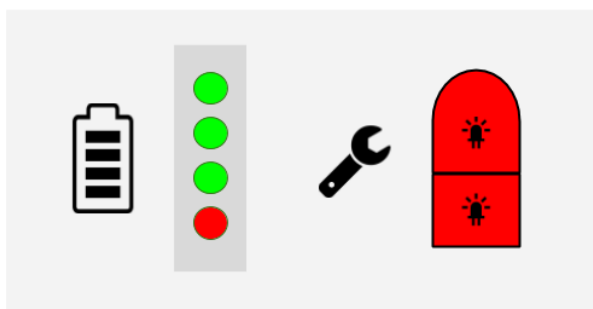


Fig. 65: tipología para las notificaciones traseras.

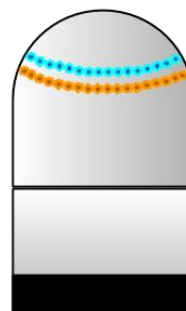


Fig. 66: disposición de notificaciones usuales.

Leds de batería

Se compran e implementan los leds referentes a la batería.

Compra y primer contacto de leds

Se analiza el mercado de leds para decidir cuál es la mejor opción a escoger para el robot. Finalmente se llega a las siguientes conclusiones:

- Se decide comprar una tira de leds y se recortará después para utilizar solamente 4 para las notificaciones de batería, pudiendo usar el sobrante de la bovina para el resto de notificaciones.
- La tira no debe necesitar una fuente de alimentación externa a la Raspberry, ya que no hay suficientes conexiones de alimentación disponibles. Por ello, se necesita de una tira de leds de 5V que se pueda alimentar directamente a través de los pines de la Raspberry.

- Para poder cortar los leds, se decide usar el tipo de leds WS2812B, que tiene la característica de poder controlar cada led por separado. Todos son independientes entre sí. Por tanto, se puede cortar la tira de leds de la largaria deseada sin provocar problemas en la tira.

Se compran los leds de [44], y se usarán también para la tira de leds de notificaciones principales y para los dos leds de error. La tira se recorta y conecta a los pines de la Raspberry, que se pueden ver en la Fig. 67. La tira de leds tiene tres conexiones, que se pueden observar en la Fig. 68:

- La de alimentación, que se conecta al pin 4 (5V).
- La conexión de datos, que se conecta al pin 12 (GPIO18).
- La conexión de tierra, que se conecta al pin 6 (GND).

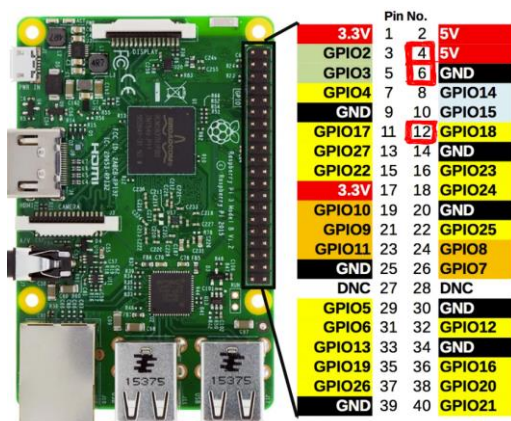


Fig. 67: esquema de pines de la Raspberry [45].

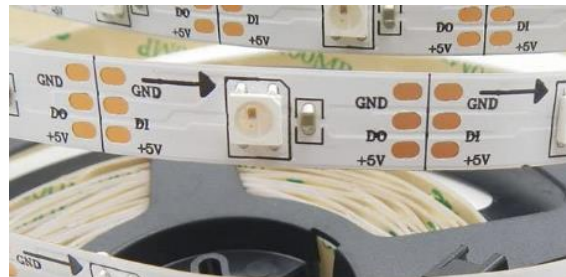


Fig. 68: conexiones de tierra, datos y 5V respectivamente de la tira de leds [44].

Tras haber conectado los leds a la Raspberry de manera física, se sigue el tutorial de [46] para el control de leds a través de una Raspberry. Éste usa la librería `rpi_ws281x` [47]. Se programa un script para comprobar que los leds se conectan bien y que puedan ser controlados sin problemas. En el siguiente apartado se integra la librería de control de leds en un nodo de ROS para su integración con el sistema del robot.

Scripts

Se integra el trabajo sobre los leds de batería para incorporarlos en el sistema ROS que corre en la Raspberry Pi.

Para programar en ROS se debe tener en cuenta que existen, como se ha comentado anteriormente, una serie de componentes principales que forman una estructura concreta: se crean nodos, que en su interior tienen *Publishers* y *Subscribers*. Los *Publishers* publican mensajes en *topics*, y los *Subscribers* se suscriben a los *topics* para recibir los mensajes de los *Publishers*. De esta manera los nodos se comunican entre ellos.

Para integrar los leds de batería, se crean dos nodos; cada uno se programa en un script diferente. En concreto, se crean un nodo de gestión de batería y otro de control de leds. Los scripts se encuentran disponibles en https://github.com/calilluis/TFG-RFID-Robot/leds_bateria con los nombres “script_node_BatteryManager.py” y “script_node_LedsController.py”.

El nodo de gestión de batería tiene como objetivo transformar los valores de batería de la base en un estado de batería controlado. El nodo de control de leds se encarga de encender y apagar los leds físicos en función del estado de batería que haya calculado el nodo anterior.

En primer lugar se implementa el nodo de gestión de batería, *Battery Manager*, visible en la parte inferior izquierda del esquema de la Fig. 69. El nodo de Kobuki (visible en la parte superior izquierda de la Fig. 69) ya existe y solo se usa para acceder al valor de batería. El nodo de control de leds, *Node Leds Controller*, se implementa en el segundo script y se observa en la parte inferior derecha de la Fig. 69.

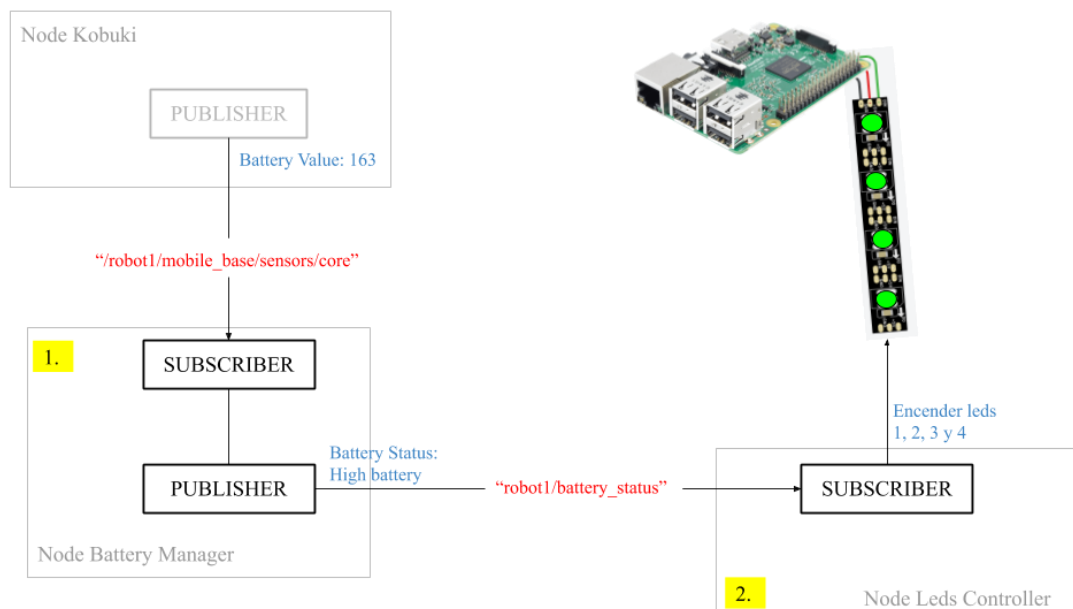


Fig. 69: esquema de relaciones entre nodos del gestor de Batería.

En la Fig. 69 se puede observar un fragmento de la estructura de nodos de ROS del robot, en este caso haciendo referencia a la parte de batería que se implementa. Se observan las relaciones entre nodos que actúan en los 2 scripts. De color rojo se encuentran los topics; en azul, un ejemplo de mensaje a través de la estructura de nodos hasta su último nivel (encender las luces).

1. Script “node_BatteryManager”

En este script, cuyo esquema se observa en la Fig. 70, se observa cómo:

- En el *main* se crea un nuevo nodo llamado “BatteryManager”, así como una clase de Python, llamada “BatCommunication”, donde se integran el *Publisher* y el *Subscriber* del nodo. Se observa en las dos primeras cajas de la Fig. 70.
- En la clase se crea un *Subscriber* que se suscribe al *topic* “/robot1/mobile_base/sensors/core” para poder obtener el valor de batería actual, cuyo rango se sabe que va de 163 a 97.
 - Al recibir los mensajes del *topic*, ejecuta la función “f_bat_status” para calcular el estado (*status*) de la batería actual en función del valor de batería recibido.
 - Los posibles estados que puede reportar la función son: ["high battery", "medium battery", "low battery", "critical battery"]. También se hace una comprobación para observar si la batería se está cargando. Se observa el esquema de la función “f_bat_status” en la caja roja superior de la Fig. 70.
 - Se explica cómo se lleva a cabo el cálculo del estado de la batería en el siguiente subapartado “Cálculo del estado de batería”.
- También se crea un *Publisher*, que se encarga de publicar el estado calculado por el *Subscriber* en el *topic* “robot1/battery_status” con la función “publish_data”. Esta función se observa esquemáticamente en la caja roja inferior del esquema de la Fig. 70.
 - Los mensajes que envía el *Publisher* pueden ser los siguientes:
 - “Status: charging”
 - “Status: high battery”
 - “Status: medium battery”
 - “Status: low battery”
 - “Status: critical battery”
 - El *Publisher* publica el mensaje en un *topic*, que se comunica con otro nodo, encargado de controlar los leds y encenderlos en función del status recibido. Este otro nodo se implementa en el siguiente script.

En la Fig. 70 se observa el esquema del código explicado:

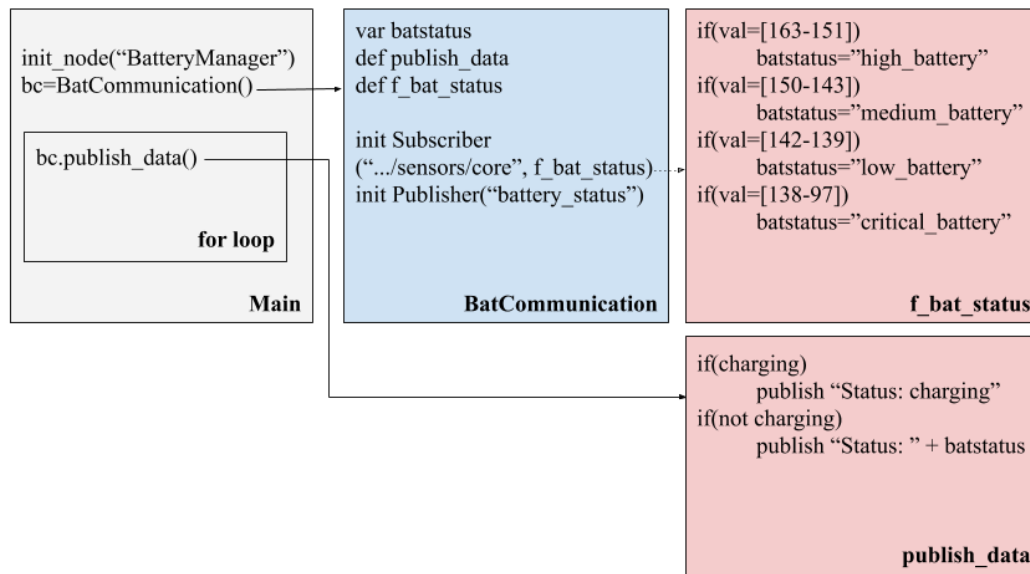


Fig. 70: esquema del script “node_BatteryManager”.

El código principal se sitúa de color gris. En azul, la clase “BatCommunication”, y en rojo las funciones. La función “publish_data” se lleva a cabo secuencialmente cada poco tiempo, y la “f_bat_status” solamente es llamada cuando el *Subscriber* recibe un valor.

Cálculo del estado de batería: los posibles estados se calculan implementando una cola (*queue*): Los datos de la batería van fluctuando, ya que los valores se publican a mucha frecuencia, de manera que una vez que se obtiene un valor de batería (e.g. 103→102), nada impide que en la siguiente comprobación se vuelva a pasar a un valor de batería superior (e.g. 103→102→103). Por ese motivo, se hace un promedio de los últimos 200 valores que se tienen de batería, que no son más de 5 segundos de datos recibidos. Solamente cuando la media de 200 valores ha pasado el umbral (*threshold*) de la siguiente fase de batería, se cambia el estado de la batería. Por ejemplo: se pasa de “high battery” a “medium battery” solo cuando el promedio de las últimas 200 muestras es menor o igual a 150, y superior a 142.

Además, para asegurar que el promedio no retrocede (algo improbable pero que podría pasar), se programa el código de manera que una vez se pasa de un estado de batería a uno menor, ya no se pueda volver atrás.

Si no se hicieran estas comprobaciones, podrían ocurrir problemas que empeorarían la experiencia de usuario, como por ejemplo que se estarían mostrando 2 leds encendidos, luego 3, y luego 2 otra vez, y así sucesivamente por un cierto tiempo hasta que los valores de batería bajasen lo suficiente.

2. Script “node_LedsController”

Se incorpora el otro nodo (inferior derecho de la Fig. 69). En él, se crea un *Subscriber* que se conecta con el *Publisher* del script anterior a través del *topic* “battery_status”, y cambia el color de los leds conectados a la Raspberry en función del estado recibido. En la Fig. 71 se observa un esquema de este script.

En concreto, en el script se hace lo siguiente:

- En el *main* se crea el nodo “LedsController” y se llama a la clase “ActivateLeds”. Se observa en la primera caja del esquema de la Fig. 71.
- En la clase “ActivateLeds” (caja azul del esquema):
 - Se crea un objeto *NeoPixel* (una tira de leds virtual) con la configuración apropiada, llamado *strip*.
 - Se crea un *Subscriber*, que se suscribe al *topic* “battery_status”, y que tiene como *callback* la función “f_act_leds”, referenciada en la primera caja roja del esquema inferior. Ésta, a partir del estado recibido en el *Subscriber* y enviado desde el *Publisher* del script anterior, decide qué leds debe encender.
 - Si recibe “critical_battery”, enciende un led en rojo.
 - Si recibe “low_battery”, enciende dos leds verdes.
 - Si recibe “medium_battery” enciende tres leds en verde.
 - Si recibe “high_battery”, enciende los 4 leds en verde.

Para encender los leds, desde “f_act_leds” se llama a la función “colorWipe”, disponible en la segunda caja roja del esquema, que ejecuta una función (“setPixelColor”) que enciende los leds físicos a través de la *strip* de leds virtual.

En la Fig. 71 se puede observar el esquema del código explicado:

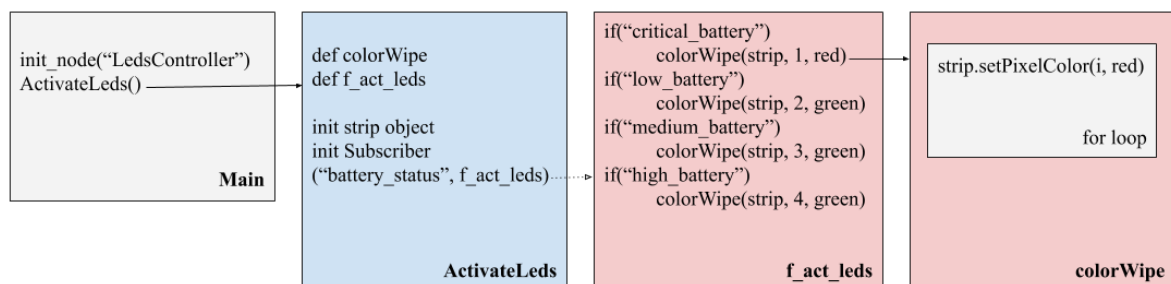


Fig. 71: esquema del script “node_LedsController”.

El código principal se sitúa de color gris. En azul, la clase “ActivateLeds”. En rojo se observan la función “f_act_leds”, que solo se llama cuando el *Subscriber* recibe un dato, y que entonces llama la función “colorWipe”.

4. CONCLUSIONES

El objetivo principal del proyecto, la mejora del diseño del robot, se cumple, ya que se consigue un prototipo con todas las consideraciones clave para obtener un producto acabado. Éstas son el estudio e implementación de la mejor configuración posible de las antenas, la optimización de los componentes internos y de la estructura del sistema, y la contemplación de una interfaz usuario - robot integrada en una carcasa que protege los componentes internos.

Se analizan los requerimientos del robot y se plantean una serie de hipótesis en función a ellos. Se hipotetiza que se puede mejorar la capacidad de lectura del robot cambiando la configuración de las antenas a la vez que se reduce la altura de éstas, y por ende la altura del robot. Se testean cuatro prototipos (llamados: Original, Inclínada, Cuadrada y Romboídal) en dos pruebas, una para evaluar el comportamiento general en un entorno realista, y otra para evaluar la capacidad de lectura en altura. Se llega a la conclusión de que se puede llevar a cabo una reducción en altura del robot, ya que mejora la estabilidad y el manejo, pero que ésta no puede ser excesiva debido a que provoca que la calidad de las lecturas disminuya notablemente. Por otro lado, no se puede observar una mejora significativa al cambiar la configuración; la Original destaca en ambas pruebas debido a su elevada altura, la Inclínada destaca en la prueba de entorno, la Romboídal en la de altura, y la Cuadrada tiene puntos favorables en ambas pruebas. Por tanto, se decide mantener la configuración cuadrada de las antenas, aunque se lleva a cabo una reducción en altura, y el prototipo final se construye a 60 cm, que es un punto intermedio entre los 110cm de la configuración Original y los 30 cm de las otras configuraciones.

Por otra parte, se plantea la posibilidad de reducir la cantidad de componentes del robot suprimiendo la batería externa. Esta hipótesis se ratifica, ya que se realiza la adaptación del sistema de conexiones y en las posteriores pruebas se consigue una autonomía de unas 3 h, un valor que se considera adecuado para que el robot pueda conseguir su objetivo. Asimismo, se lleva a cabo una caracterización más detallada de la batería del robot, analizándola y dividiendo su duración en cuatro intervalos de tiempo similares, con el objetivo de usar esa división en la implementación posterior de las notificaciones de batería.

También se plantea la posibilidad de crear un diseño modular, a la vez que se mejora la distribución de los componentes, y en consecuencia se propone un cambio de diseño interno que contemple la modularidad. Se desarrolla la propuesta cumpliendo con los requisitos planteados, y se consigue un diseño interior optimizado y modular. Algunos beneficios de este diseño es que se consigue separar la carga útil de la base robótica, de manera que solamente se pasan dos cables de conexión entre un módulo y otro.

Una vez contempladas y estudiadas las hipótesis, resta cubrir una serie de necesidades que se deben integrar para obtener un robot completo. En primera instancia, se necesita una manera de comunicar el robot y el usuario final, y por ello se propone la creación de una interfaz. Se plantean todas las necesidades de comunicación y se elabora una propuesta de sistema de notificaciones a través de leds, que consta de cuatro tipos de notificaciones distintas: generales, de navegación, de batería y de error. Se llega hasta la integración final en ROS de los leds para la monitorización de la batería, faltando por

desarrollar las demás. Cabe destacar que el uso de leds añade simplicidad al sistema y permite seguir con la estética minimalista que se busca para el robot.

En segunda instancia, se define la necesidad de crear un prototipo que pueda estar preparado para lanzar al mercado. Para ello, se plantea la necesidad de crear una carcasa protectora que integre los leds de notificación, con el objetivo de proteger los componentes internos y mejorar la estética, entre otros beneficios. El sistema de notificaciones y la carcasa protectora no se terminan de desarrollar y, por tanto, el producto no se puede dar por acabado y no está preparado para salir al mercado. De todas formas, el trabajo recoge los estudios necesarios para desarrollar estos apartados cuando llegue el momento.

Para finalizar, se puede concluir que el robot ha mejorado mucho en comparación con su versión anterior. Se han trabajado todas las hipótesis planteadas y se han podido ratificar o refutar, obteniendo en cualquier caso datos útiles e interesantes a tener en cuenta en el desarrollo del robot. Éste es perfectamente funcional sin la carcasa, y puede ser usado para elaborar un inventario RFID de manera robusta, sin percibir balanceo en su movimiento, teniendo control sobre su batería y con una buena optimización de sus componentes internos.

4.2 Trabajo futuro

Quedan pendientes algunos aspectos en los que es interesante llevar a cabo un análisis más extenso, así como algunos pasos pendientes en el desarrollo del robot.

En primer lugar, es interesante contemplar un análisis más extenso de las lecturas del robot con las diferentes configuraciones, esta vez con las antenas a una altura de 60cm, para seguir con la búsqueda de una mejor disposición de las antenas. Se podrían testear otras configuraciones, como las de dos niveles. También se debería estudiar el apantallamiento en la configuración Inclínada, y la posibilidad de que éste restrinja el rango de detección de las antenas. Para estas nuevas pruebas, sería interesante llevar a cabo el estudio tanto en una cámara anecoica (espacio libre de interferencias y rebotes de onda), para observar de manera precisa cómo se comportan las configuraciones, como en un entorno real, donde los resultados pueden ser muy diferentes.

En segundo lugar, y debido a la situación actual con el COVID-19, se pausa el desarrollo del robot. Se llega al prototipado de un diseño sin carcasa, y a la implementación de los leds de batería. Por tanto, queda como trabajo futuro la implementación de los leds de notificación usual (notificaciones generales y de navegación), así como la implementación de los leds de error traseros. Asimismo, queda pendiente el prototipado de la carcasa. Ésta incluiría el sistema de notificaciones, un asa para mejorar el manejo del robot, y un anclaje sencillo para que la carcasa se pueda retirar con rapidez.

Finalmente, queda pendiente también una prueba de inventario del nuevo prototipo en un dominio real, en el que se pruebe en un entorno como una tienda (o en la biblioteca de la universidad) para ver cómo se comportan las lecturas en un local con tags bien estructurados.

Bibliografía

- [1] Angela Elizabeth Metri, “The Importance of Design - Medium,” Dec. 2018. <https://medium.com/@aemetri/the-importance-of-design-187c3ce45961> (accessed Apr. 18, 2020).
- [2] J. Harrison, “Why Design is Critically Important for Your Business,” Jan. 22, 2018. <https://www.weidert.com/blog/importance-of-design> (accessed Apr. 18, 2020).
- [3] Wikipedia, “Engineering design process .” https://en.wikipedia.org/wiki/Engineering_design_process (accessed Apr. 15, 2020).
- [4] Science Buddies, “The Engineering Design Process.” <https://www.sciencebuddies.org/science-fair-projects/engineering-design-process/engineering-design-process-steps> (accessed Apr. 15, 2020).
- [5] Wikipedia, “RFID .” <https://es.wikipedia.org/wiki/RFID> (accessed Apr. 16, 2020).
- [6] “Keonn Wiki.” <http://wiki.keonn.com/> (accessed Apr. 15, 2020).
- [7] SparkFun Electronics, “UHF RFID Tag - WRL-14147 .” <https://www.sparkfun.com/products/14147> (accessed Apr. 16, 2020).
- [8] M. Morenza-Cinos, V. Casamayor-Pujol, J. Soler-Busquets, J. L. Sanz, R. Guzmán, and R. Pous, “Development of an RFID inventory robot (AdvanRobot),” *Stud. Comput. Intell.*, vol. 707, pp. 387–417, 2017, doi: 10.1007/978-3-319-54927-9_12.
- [9] and H. P. Botero, Julián, David González, “La interferencia como un factor que afecta el desempeño de un sistema RFID,” *Tecnura*, vol. 13, no. 25, pp. 145–153, 2009, doi: 10.14483/22487638.6677.
- [10] Keonn Technologies, “Advantenna-SP11™ High gain compact RFID UHF antenna.” 2019, [Online]. Available: <https://www.keonn.com/rfid-components/antennas/advantenna-sp11.html>.
- [11] ROS, “ROS.org ,” *Powering the world’s robots*. <https://www.ros.org/> (accessed Apr. 28, 2020).
- [12] “ROS/Introduction - ROS Wiki.” <http://wiki.ros.org/ROS/Introduction> (accessed Apr. 15, 2020).
- [13] Juan J. Romero Marra, “Conceptos básicos de ROS | Romerobots Blog,” *Conceptos básicos de ROS*, 2014. <https://jjromeromarras.wordpress.com/2014/08/27/conceptos-basicos-de-ros/> (accessed Apr. 15, 2020).
- [14] “Simbe | Say hello to Tally.” *Say hello to Tally*. <https://www.simberobotics.com/platform/tally/> (accessed Apr. 15, 2020).
- [15] “Stockbot Technical Specifications.” Accessed: Apr. 15, 2020. [Online]. Available: www.pal-robotics.com.

- [16] "TORY. - MetraLabs." <https://www.metalabs.com/en/rfid-robot-tory/> (accessed Apr. 15, 2020).
- [17] MetraLabs, "About Tory." Accessed: Apr. 15, 2020. [Online]. Available: https://www.metalabs.com/wp-content/uploads/2019/06/2019_about-tory_en.pdf.
- [18] Keonn Technologies, "RFID robot for automatic inventory: AdvanRobot," *Keonn*, [Online]. Available: <http://keonn.com/systems/view-all-2/inventory-robots.html>.
- [19] Generation Robots, "YDLIDAR X4 laser range finder." <https://www.generationrobots.com/en/403022-360-degree-laser-scanner-yd-lidar-x4.html> (accessed Apr. 15, 2020).
- [20] Parallax Inc, "PING Ultrasonic Distance Sensor 28015 ." <https://www.parallax.com/product/28015> (accessed Apr. 17, 2020).
- [21] Raspberry Pi, "Raspberry Pi 3 Model B+ ." <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/> (accessed May 09, 2020).
- [22] Litionite, "Tanker 90W / 50.000mAh." <https://www.litionite.com/product/tanker/> (accessed May 09, 2020).
- [23] ROS Components, "Kobuki," 2016. <https://www.roscomponents.com/es/robots-moviles/97-kobuki.html> (accessed Apr. 15, 2020).
- [24] R. Pous, "Conceptualizing the Internet of Things." 2019.
- [25] "Tinkercad." <https://www.tinkercad.com> (accessed Apr. 25, 2020).
- [26] lubyn, "YDLIDAR X4 DATASHEET," 2015. Accessed: Apr. 15, 2020. [Online]. Available: www.ydlidar.com.
- [27] Amazon, "MXECO DC Converter 12V To 5V 3A 15W with Micro USB." https://www.amazon.es/MXECO-Reliable-Technology-Converter-CPT-UL-6/dp/B07QFDRDZC/ref=sr_1_fkmr2_1? (accessed Apr. 29, 2020).
- [28] Amazon.es, "CC 12V a 24V ." https://www.amazon.es/gp/product/B07Q2CHC2Q/ref=ppx_yo_dt_b_asin_title_o04_s00? (accessed May 26, 2020).
- [29] Amazon, "kenable 008976 - Cable USB (0,6 m, USB A, Micro-USB B)." https://www.amazon.es/Micro-Blindado-Rápido-Charge-Cable/dp/B07167BGG3/ref=sr_1_55? (accessed Apr. 29, 2020).
- [30] DigiKey, "0462355002 Molex | Conectores, interconectores | ." <https://www.digikey.com/product-detail/es/molex/0462355002/WM2259CT-ND/2354691> (accessed Apr. 29, 2020).
- [31] DigiKey, "0430250200 Molex | Connectors, Interconnects | ." <https://www.digikey.es/product-detail/en/molex/0430250200/WM1783-ND/252496> (accessed Apr. 29, 2020).
- [32] Liferay, "The Benefits of Modularity." <https://help.liferay.com/hc/en-us/articles/360035828171-The-Benefits-of-Modularity> (accessed Apr. 30, 2020).

- [33] Smashing Robotics, “All About Self-Reconfiguring Modular Robots .” <https://www.smashingrobotics.com/all-about-self-reconfiguring-modular-robots-advantages-and-disadvantages/> (accessed Apr. 30, 2020).
- [34] A. T., “Why Rounded Corners Are Easier on the Eyes.” <https://uxmovement.com/thinking/why-rounded-corners-are-easier-on-the-eyes/> (accessed May 07, 2020).
- [35] T. Koe, “Rounded or Sharp-Corner Buttons?,” Jul. 2018. <https://uxplanet.org/rounded-or-sharp-corner-buttons-def3977ed7c4> (accessed May 07, 2020).
- [36] Santa & Cole, “Asa, de Miguel Milá .” <https://www.santacole.com/es/catalogo/lamparas-de-sobremesa/asa> (accessed May 28, 2020).
- [37] Apple, “Mac Pro 2013.” <https://www.apple.com/la/mac-pro-2013/specs/> (accessed May 28, 2020).
- [38] Amazon, “Cecotec Calefactor.” https://www.amazon.es/Cecotec-Calefactor-oscilación-termostato-sobrecalentamiento/dp/B07JJC7S5R/ref=sr_1_15 (accessed May 19, 2020).
- [39] Ebay, “Carro Compra Rolser LN Dos.” <https://www.ebay.es/i/153441234827> (accessed May 19, 2020).
- [40] Amazon, “HAYLOU GT1 Pro .” <https://www.amazon.es/HAYLOU-Auriculares-inalámbricos-Reducción-Auricular/dp/B07ZT1WPLV> (accessed May 19, 2020).
- [41] Amazon, “Echo Plus.” https://www.amazon.es/Echo-Plus-2-ª-generación-controlador/dp/B07CT3VM3H/ref=asc_df_B07CT3VM3H/ (accessed May 19, 2020).
- [42] J. A. Adams, “Critical Considerations for Human-Robot Interface Development,” 2002. Accessed: Apr. 30, 2020. [Online]. Available: www.aaai.org.
- [43] M. A. Goodrich and D. R. Olsen, “Seven principles of efficient human robot interaction,” *Proc. IEEE Int. Conf. Syst. Man Cybern.*, vol. 4, no. May, pp. 3943–3948, 2003, doi: 10.1109/icsmc.2003.1244504.
- [44] Amazon, “WS2812B Dream Color DC 5V (5M-WS2812B-30-NP).” https://www.amazon.es/píxeles-individualmente-direccionable-iluminación-impermeable/dp/B07439RXD3/ref=sr_1_4_sspa? (accessed May 01, 2020).
- [45] MBRobotics, “Raspberry PI - GPIOs .” <http://mbrobotics.es/blog/raspberry-pi-gpios/> (accessed May 01, 2020).
- [46] Raspberry Pi Tutorials, “Connect and Control WS2812 LED Strips via Raspberry.” <https://tutorials-raspberrypi.com/connect-control-raspberry-pi-ws2812-rgb-led-strips/> (accessed May 01, 2020).
- [47] GitHub (jgarff), “jgarff/rpi_ws281x: Userspace Raspberry Pi PWM library for WS281X LEDs.” https://github.com/jgarff/rpi_ws281x (accessed May 20, 2020).

Anexos

Prueba de entorno- “2.2 - Porcentajes por antena”:

Iteración 1:

PUERTO\CONFIGURACIÓN	ORIGINAL	CUADRADA	INCLINADA
FRONTAL	36,03% (2)	25,89% (4)	37,53% (2)
TRASERA	14,41% (3)	1,79% (3)	11,66% (3)
DERECHA	22,32% (4)	21,15% (1)	18,49% (4)
IZQUIERDA	27,24% (1)	51,17% (2)	32,32% (1)

Tabla 10: Porcentajes leídos por cada configuración (menos la Romboidal) en cada una de las direcciones cardinales. Repetición 1. Entre paréntesis se muestra el puerto del reader al que la antena se conecta en cada dirección.

PUERTO\CONFIGURACIÓN	ROMBOIDAL
FRONTAL DERECHA	23,14% (4)
TRASERA IZQUIERDA	11,54% (3)
TRASERA DERECHA	4,76% (1)
FRONTAL IZQUIERDA	60,56% (2)

Tabla 11: Porcentajes leídos por la configuración Romboidal en cada una de las direcciones cardinales. Iteración 1. Entre paréntesis se muestra el puerto del reader al que la antena se conecta en cada dirección.

Iteración 2:

PUERTO\CONFIGURACIÓN	ORIGINAL	CUADRADA	INCLINADA
FRONTAL	38,03% (2)	28,16% (4)	48,73% (2)
TRASERA	14,67% (3)	1,97% (3)	14% (3)
DERECHA	22,08% (4)	20,82% (1)	15,7% (4)
IZQUIERDA	25,22% (1)	49,05% (2)	21,57% (1)

Tabla 12: Porcentajes leídos por cada configuración (menos la Romboidal) en cada una de las direcciones cardinales. Iteración 2. Entre paréntesis se muestra el puerto del reader al que la antena se conecta en cada dirección.

PUERTO\CONFIGURACIÓN	ROMBOIDAL
FRONTAL DERECHA	27,7% (4)
TRASERA IZQUIERDA	10,1% (3)
TRASERA DERECHA	10,45% (1)
FRONTAL IZQUIERDA	51,75% (2)

Tabla 13: Porcentajes leídos por la configuración Romboidal en cada una de las direcciones cardinales. Iteración 2. Entre paréntesis se muestra el puerto del reader al que la antena se conecta en cada dirección.