

---

# Diseño e Implementación de una Plataforma Móvil Para Aplicaciones de Robótica de Enjambre - Fase III

---

Julio Aq'ab'al Rodriguez Xicará



UNIVERSIDAD DEL VALLE DE GUATEMALA  
Facultad de Ingeniería



**Diseño e Implementación de una Plataforma Móvil Para  
Aplicaciones de Robótica de Enjambre - Fase III**

Trabajo de graduación presentado por Julio Aq'ab'al Rodriguez Xicará para optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

2022

Vo.Bo.:

(f) \_\_\_\_\_  
Dr. Luis Alberto Rivera Estrada

Tribunal Examinador:

(f) \_\_\_\_\_  
Dr. Luis Alberto Rivera Estrada

(f) \_\_\_\_\_  
M. Sc. Miguel Zea

(f) \_\_\_\_\_  
M. Sc. Carlos Esquit

Fecha de aprobación: Guatemala, 22 de junio de 2022.

---

## Prefacio

---

Dedico este trabajo de graduación a mi familia, ellos son los pilares más importantes en mi vida.

A mi madre: por haberme enseñado los valores del estudio y brindarme todo su apoyo.

A mi hermano: por apoyarme en los momentos de dudas, desafíos y motivarme a lograr mis metas.

A mi cuñada: por su apoyo y por ser un hermano para ella. A mis sobrinos: por su cariño.

Un especial agradecimiento a mi padre, mi primer maestro, mi consejero, que el Creador lo tenga en su descanso eterno y que desde acá sepa que su inspiración y los valores que me transmitió hoy se materializan en el presente trabajo y en la culminación de mi formación académica.

---

## Índice

---

<b>Prefacio</b>	<b>III</b>
<b>Lista de figuras</b>	<b>VII</b>
<b>Lista de cuadros</b>	<b>VIII</b>
<b>Resumen</b>	<b>IX</b>
<b>Abstract</b>	<b>X</b>
<b>1. Introducción</b>	<b>1</b>
<b>2. Antecedentes</b>	<b>2</b>
<b>3. Justificación</b>	<b>7</b>
<b>4. Objetivos</b>	<b>8</b>
<b>5. Alcance</b>	<b>9</b>
<b>6. Marco teórico</b>	<b>10</b>
6.1. Robótica de Enjambre . . . . .	10
6.1.1. Particle Swarm Optimization (PSO) . . . . .	10
6.2. Comunicación Inalámbrica . . . . .	11
6.2.1. Protocolos TCP/IP, UDP . . . . .	11
6.3. Sensores de Proximidad . . . . .	11
6.3.1. Sensor Ultrasónico . . . . .	11
6.4. Raspberry Pi . . . . .	12
6.4.1. Raspberry Pi 3 B + . . . . .	13
6.4.2. Raspberry Pi Zero W . . . . .	14
6.4.3. Raspberry Pi 4 . . . . .	14
6.5. Servomotor . . . . .	15
6.6. Motor DC . . . . .	15
6.7. Motores con caja reductora . . . . .	16

6.8. Drivers Para Motores . . . . .	16
6.8.1. Driver TB6612FNG . . . . .	17
6.9. Baterías de Li-Ion . . . . .	17
6.10. Robots Móviles . . . . .	19
<b>7. Diseño electrónico de la nueva plataforma</b>	<b>20</b>
7.1. Sistema de Control . . . . .	20
7.2. Módulo de detección . . . . .	21
7.3. Sistema de movimiento . . . . .	21
7.4. Módulo de visión por computadora . . . . .	22
7.5. Módulo de detección de objetos . . . . .	23
7.6. Sistema de potencia . . . . .	24
7.7. Esquemático final . . . . .	25
<b>8. Diseño mecánico de la nueva plataforma</b>	<b>28</b>
8.1. Dimensiones físicas de la fase II . . . . .	28
8.2. Base de la nueva plataforma . . . . .	28
8.3. Estructura final . . . . .	30
8.4. Diseño PCB . . . . .	31
8.5. Comparación plataforma fase II y fase III . . . . .	34
<b>9. Programación de la plataforma</b>	<b>37</b>
9.1. Programación Raspberry Pi Zero . . . . .	37
9.1.1. Sistema de movimiento . . . . .	38
9.1.2. Sistema de visión por computadora . . . . .	39
9.1.3. Comunicación Inalámbrica . . . . .	41
9.1.4. Sistema de detección . . . . .	41
<b>10. Pruebas de la plataforma</b>	<b>42</b>
10.1. Comunicación por medio TCP y UDP . . . . .	42
10.2. Sistema de visión por computadora . . . . .	47
10.3. Detección de objetos . . . . .	52
10.4. Movimientos de los motores . . . . .	55
<b>11. Conclusiones</b>	<b>58</b>
<b>12. Recomendaciones</b>	<b>59</b>
<b>13. Bibliografía</b>	<b>60</b>

---

## Lista de figuras

---

1.	Imagen de un <i>E-Puck</i> en físico [1]. . . . .	2
2.	Imagen de un Pi-Puck en físico [3]. . . . .	3
3.	Diagrama del Kilobot [4]. . . . .	4
4.	Vista final del robot para aplicaciones en robótica de enjambre de la fase 1 de la Universidad del Valle de Guatemala. [5] . . . . .	5
5.	Bytebot fase 2 [6]. . . . .	6
6.	Sensor de proximidad inductivo [12]. . . . .	12
7.	LV-MaxSonar-EZ1 [13]. . . . .	12
8.	<i>Raspberry Pi</i> GPIO pines [16]. . . . .	13
9.	Modelo físico Modelo <i>Raspberry Pi</i> 3B+ [16]. . . . .	14
10.	Modelo físico <i>Raspberry Pi Zero W</i> [17]. . . . .	14
11.	Modelo físico <i>Raspberry Pi 4</i> [16]. . . . .	15
12.	Servomotor [18]. . . . .	15
13.	Motor DC [19]. . . . .	16
14.	Motor con caja reductora [19]. . . . .	16
15.	Driver de Motor L298N [19]. . . . .	17
16.	Driver para Motor TB6612FNG [21] . . . . .	17
17.	Batería 18650 Marca Samsung de 2500mAh [22] . . . . .	18
18.	Robot móvil [19]. . . . .	19
19.	Esquema generado en <i>Altium Designer</i> [24]. . . . .	21
20.	<i>Raspberry Pi</i> Camera Module 2 NoIR[24]. . . . .	22
21.	<i>Raspberry Pi</i> Camera Module V2 conectada a una Raspberry Pi Zero[24]. . . . .	23
22.	Conexión por medio de PWM[13] . . . . .	23
23.	Esquema de detección generado en Altium Designer . . . . .	24
24.	Esquemático final hecho en Altium Designer. . . . .	25
25.	Esquemático final hecho en <i>Fritzing</i> . . . . .	26
26.	Modelo 3D del proyecto Zerobot [26]. . . . .	29
27.	Primera iteración. . . . .	29
28.	Segunda iteración. . . . .	30
29.	Tercera iteración modelo 3D. . . . .	31

30.	Modelo 3D Vista inferior. . . . .	31
31.	Modelo 3D Vista superior. . . . .	32
32.	Modelo impreso Vista superior. . . . .	32
33.	Modelo impreso Vista inferior. . . . .	33
34.	Modelo impreso Vista frontal. . . . .	33
35.	Planos de la base generados en Inventor. . . . .	34
36.	PCB realizada en Altium Designer. . . . .	34
37.	PCB con el modelo de <i>Raspberry Pi Zero W</i> . . . . .	35
38.	Pantalla inicial <i>Raspberry Pi Zero W</i> utilizando VNC . . . . .	38
39.	Configuración de pines . . . . .	38
40.	Fotografía tomada con resolución 640×480. . . . .	40
41.	Transmisión en vivo desde un servidor local. . . . .	40
42.	Comunicación vía TCP: Servidor operacional . . . . .	43
43.	Comunicación vía TCP: Cliente conectado al servidor . . . . .	43
44.	Interfaz del usuario desde la Raspberry Pi 3 dirección: 192.168.0.2 (dirección IP especificada únicamente para los ejemplos presentados en estas pruebas). . . . .	45
45.	Pantalla de la plataforma móvil para aplicaciones de robótica de enjambre dirección: 192.168.0.8 (dirección IP especificada únicamente para los ejemplos presentados en estas pruebas). . . . .	46
46.	Pantalla del cliente Raspberry Pi 3 dirección: 192.168.0.20 (dirección IP especificada únicamente para los ejemplos presentados en estas pruebas). . . . .	46
47.	Fotografía desde la Raspberry Pi NoIR camera V2 con el balance de blancos: auto . . . . .	48
48.	Fotografía desde la Raspberry Pi NoIR camera V2 con el balance de blancos: off . . . . .	49
49.	Fotografía desde la Raspberry Pi NoIR camera V2 con el balance de blancos: sunlight . . . . .	49
50.	Vídeo desde la Raspberry Pi NoIR camera V2 con el balance de blancos: auto . . . . .	50
51.	Transmisión en vivo realizada desde la plataforma móvil y visualizada en un dispositivo Android . . . . .	51
52.	Transmisión en vivo realizada desde la plataforma móvil y visualizada en una computadora . . . . .	51
53.	Resultado de la medición del sensor LV-MaxSonar-EZ1 . . . . .	52
54.	Resultado de la medición del sensor LV-MaxSonar-EZ1 . . . . .	53
55.	Resultado de la medición del sensor LV-MaxSonar-EZ1 . . . . .	53
56.	Resultado de la medición del sensor LV-MaxSonar-EZ1 . . . . .	53
57.	Resultado de la medición del sensor LV-MaxSonar-EZ1 . . . . .	54
58.	Resultado de la medición del sensor LV-MaxSonar-EZ1 . . . . .	54
59.	Movimiento hacia adelante y en reversa programado para la plataforma . . . . .	56
60.	Movimiento para el giro hacia la izquierda y giro hacia la derecha programado para la plataforma . . . . .	56
61.	Movimiento hacia adelante para la izquierda y hacia adelante para la derecha programado para la plataforma . . . . .	57
62.	Movimiento en reversa para la izquierda y reversa para la derecha programado para la plataforma . . . . .	57

---

## Lista de cuadros

---

1.	Consumo de corriente de la nueva plataforma . . . . .	25
2.	Presupuesto detallado de la nueva plataforma . . . . .	36
3.	Configuración de giro para las llantas. . . . .	39
4.	Comandos para los distintos movimientos en las llantas. . . . .	39
5.	Porcentajes de error en los sensores LV-MaxSonar-EZ1 . . . . .	52
6.	Porcentaje de funcionamiento de los motores y RPM alcanzadas . . . . .	55

---

## Resumen

---

Se realizó una plataforma móvil siguiendo la línea de investigación que tiene la Universidad del Valle de Guatemala para aplicaciones en robótica de enjambre. La plataforma es la tercera fase en la línea de investigación que empezó con creación de plataformas en el 2017. Esta nueva plataforma es un cambio completo en programación y diseño con respecto a la fase II realizada en el 2018. En el hardware se mantienen algunos de los componentes seleccionados en la fase II. Las contribuciones a esta nueva plataforma son la implementación de un nuevo controlador que es la *Raspberry Pi Zero W*, un nuevo módulo de visión de computadora por medio de una cámara que brinda resultados para la obtención de fotos, videos y transmisiones en vivo, creación de nuevas funciones para el desplazamiento y nuevas formas de comunicarse de manera inalámbrica utilizando protocolos TCP o UDP.

El controlador *Raspberry Pi Zero W*, fue programado en el lenguaje de *Python* siendo otra diferencia significativa con respecto a la fase II que fue totalmente hecha en C. La interfaz para el usuario se hizo para que pueda controlado desde otro dispositivo que únicamente cumpla con el requisito de poder ejecutar códigos de *Python* y que se encuentren conectados a la misma red. La *Raspberry Pi Zero W* recibe energía de una batería recargable 18650 3.7 V 9800 mAh que permite un tiempo de funcionamiento de 3 horas continuas. Esta batería es suficiente para poder encender los componentes que se encuentren conectados como el servomotor, los sensores y la cámara.

---

## Abstract

---

A mobile platform was developed following the research line of the Universidad del Valle de Guatemala for swarm robotics applications. The platform is the third phase in the line of research that began with the creation of platforms in 2017. This new platform is a complete change in programming and design with respect to phase II carried out in 2018. In the hardware some of the components selected in phase II are maintained. The contributions to this new platform are the implementation of a new controller which is the Raspberry Pi Zero W, a new computer vision module by means of a camera that provides results for obtaining photos, videos and live streams, creation of new functions for scrolling and new ways to communicate wirelessly using TCP or UDP protocols.

The Raspberry Pi Zero W controller was programmed in the Python language being another significant difference with respect to Phase II which was entirely made in C. The user interface was made so that it can be controlled from another device that only meets the requirement of being able to run Python code and that are connected to the same network. The Raspberry Pi is powered by a rechargeable battery 18650 3. 7 V 9800 mAh battery that allows an operating time of 3 continuous hours. This battery is enough to power the components that are connected as the servomotor, sensors and camera.

# CAPÍTULO 1

---

## Introducción

---

El presente trabajo de tesis tomó como base el modelo anterior de la fase II, el cual ya contaba con las siguientes capacidades: detección de objetos por medio de sensores ultrasónicos *LV-MaxSonar-EZ1* con una zona de detección de 292°, movilización de la plataforma por motores de corriente directa controladas individualmente y monitoreo de parámetros del motor como lo son el desplazamiento, velocidad y sentido de orientación utilizando un *encoder* para cada llanta, comunicación inalámbrica vía Wi-Fi utilizando el módulo *ESP-286* para comunicarse con el ordenador.

Para esta nueva fase se toma en cuenta las recomendaciones de la fase anterior, además de la selección de los componentes para la detección de objetos y la movilización. Para optimización de la nueva plataforma se realizan cambios en las dimensiones físicas de las llantas y se cambia la batería por una con menores dimensiones físicas y con más potencia que la anterior, para facilidad de utilidad en trabajos de robótica de enjambre realizados en la Universidad del Valle de Guatemala. Se decidió cambiar el microcontrolador por el modelo *Raspberry Pi Zero W* con programación en *Python*, que facilita la adaptación para posibles algoritmos de robótica de enjambre desarrollados dentro de la Universidad. La plataforma anterior trabajó en lenguaje C por lo que se tuvo que realizar una migración de funcionalidades. El nuevo sistema de detección de imágenes se implementa por medio de una cámara *Noir* para diferentes tipos de funciones según sea la necesidad del usuario. Estas pueden ser grabación de vídeo, toma de fotografías o realización de transmisiones en directo, la cámara utilizada ha sido probada en otro tipo de proyectos que involucran reconocimiento de símbolos, los cuales en tecnología de robótica de enjambre pueden llegar a ser de utilidad. El nuevo modelo toma como referencia la forma circular, con los motores al borde, los sensores y el servo en la parte central del robo. Sin embargo, este tiene forma más estirada no correspondiendo con una forma geométrica simple.

# CAPÍTULO 2

---

## Antecedentes

---

### E-puck

El *E-Puck* es un sistema embebido como se observa en la Figura 1, en el cual se tiene una serie de diversos campos que lo vuelven una plataforma de aprendizaje que es atractiva para los estudiantes, entre estas se encuentran: mecanismos, programación y electrónica. Todo el software, diseño y documentos de producción están disponibles bajo una licencia de hardware y software de código abierto [1].



Figura 1: Imagen de un *E-Puck* en físico [1].

Para poder utilizar el *E-Puck* existen distintas herramientas de software compatibles para esta plataforma Educativa, entre ellos *Enki* y *Webots* [2].

*Webots* implementa un sistema muy eficaz para poder realizar los experimentos, siendo

un proceso relativamente simple de cuatro pasos.

- Seleccionar el modelo
- Programar
- Simular
- Probar en el modelo físico

## Pi Puck

Una *Raspberry Pi* con interfaz con un robot *E-Puck* Desarrollada por *York Robotics Laboratory (YRL)* en la Universidad de York y GCtronic

El Pi-puck es una extensión para *E-Puck* y *E-Puck2*, lo que permite montar una computadora de placa única *Raspberry Pi Zero W* en el robot, agregando soporte para Linux, periféricos adicionales y más posibilidades de expansión.



Figura 2: Imagen de un Pi-Puck en físico [3].

## Kilobots

El Kilobot es un robot de enjambre de bajo costo de 3,3 cm de altura desarrollado por Radhika Nagpal y Michael Rubenstein en la Universidad de Harvard. Pueden actuar en grupos, hasta mil, para ejecutar comandos programados por usuarios que no podrían ser ejecutados por robots individuales. Un problema con la investigación sobre colectivos de robots es que el costo de las unidades individuales es alto. El costo total de las piezas del Kilobot es de menos de 15 dólares. Además de su bajo costo, tiene aplicaciones como transporte colectivo, interacción humano-enjambre y auto ensamblaje de formas.

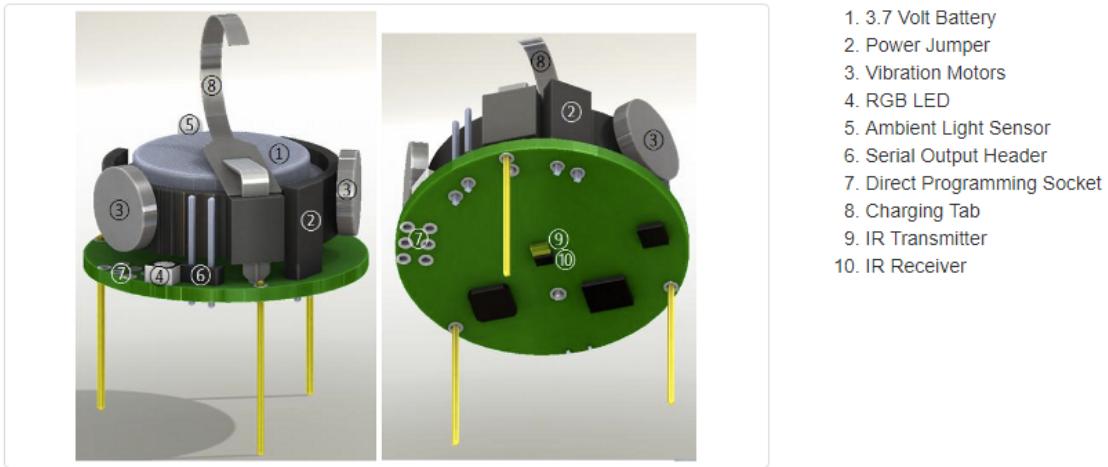


Figura 3: Diagrama del Kilobot [4].

## Plataforma Robótica para aplicaciones en Robótica de Enjambre Fase 1

Esta plataforma fue desarrollada en la Universidad del Valle de Guatemala en el año 2017 para aplicaciones en robótica de enjambre, como proyecto de graduación en su primera fase [5].

Se diseñó con una falda de 6 sensores ultrasónicos HC-SR04 para detectar objetos alrededor de la plataforma posicionados a 60° entre cada uno de ellos, estos 6 sensores se controlan en una placa PCB por separado de la principal, que es un decodificador de 3 a 8 líneas que permite controlar el *Trigger* de todos los sensores, utilizando únicamente 3 pines de salida digital de un microcontrolador y utiliza lógica combinacional para que el *Echo* de los sensores llegue a un solo pin de entrada digital [5].

La unidad de control es un microcontrolador Teensy 3.2 basado en ARM Cortex con 256 KB de memoria flash, 256 KB de memoria caché, 64 KB de RAM que se programa en lenguaje de arduino. Tiene una IMU, que se compone de un giroscopio y acelerómetro, ambos de tres ejes. Su movimiento es diferencial, es decir, con dos llantas y un motor anclado en cada una de ellas y controladas de forma independiente, utilizando un *ball caster* como apoyo. Se utiliza un controlador de motores DC doble HDRV8833 para el manejo de las llantas con señales PWM. Tiene un *encoder* instalado en cada llanta para obtener datos de la velocidad, distancia recorrida y sentido en el que gira cada una. Utiliza un módulo ESP-8266 para la comunicación Wi-Fi con un ordenador [5].

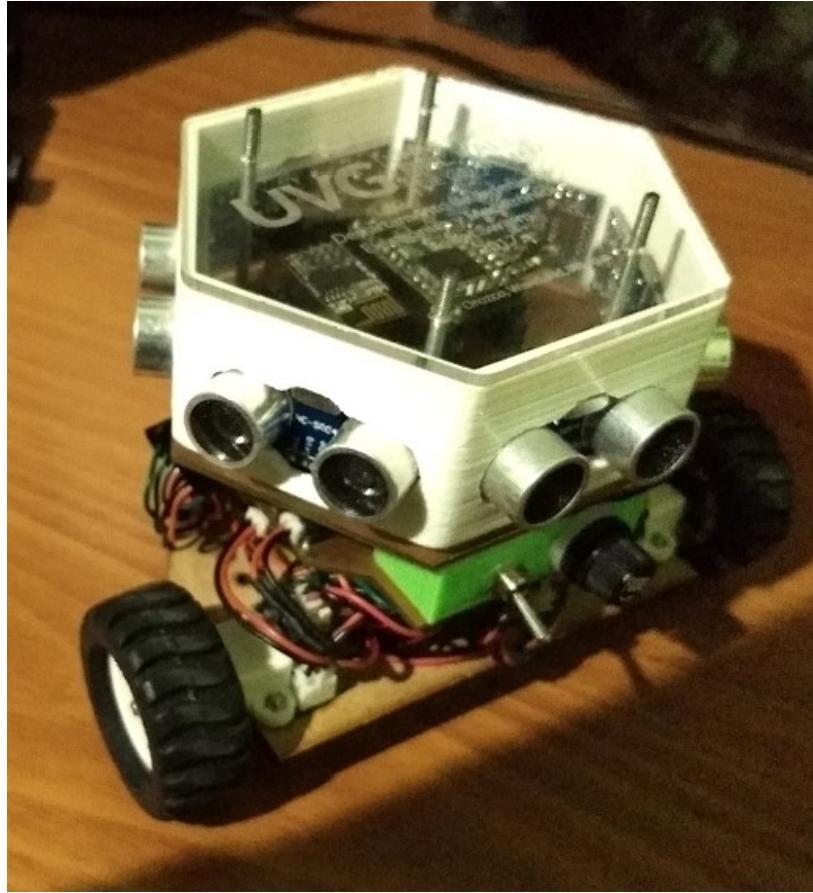


Figura 4: Vista final del robot para aplicaciones en robótica de enjambre de la fase 1 de la Universidad del Valle de Guatemala. [5]

## Plataforma Robótica para aplicaciones en Robótica de Enjambre Fase 2

Dentro de la Universidad se han realizado otros trabajos relacionados con la creación de una plataforma robótica para implementar algoritmos de enjambre, siendo el más reciente es el del Ing. Carlos Javier Lima Cordón [6] realizado en 2018.

Para controlar la plataforma se utilizó comunicación Wi-Fi, además de que utiliza el microcontrolador PIC32MX250F128B. El trabajo de Lima constituyó la segunda fase en el desarrollo de una nueva plataforma para aplicaciones en robótica de enjambre.

En esta fase se logró un funcionamiento por módulos para la plataforma, no se tiene registro de un funcionamiento implementando programas de robótica de enjambre, las limitaciones de esta plataforma se ven en la velocidad del controlador, las dimensiones físicas de este llegan a ser demasiado grandes para poder utilizarse de manera eficaz en una mesa de pruebas y por ultimo la detección de objetos con su entorno puede llegar a ser ineficiente.

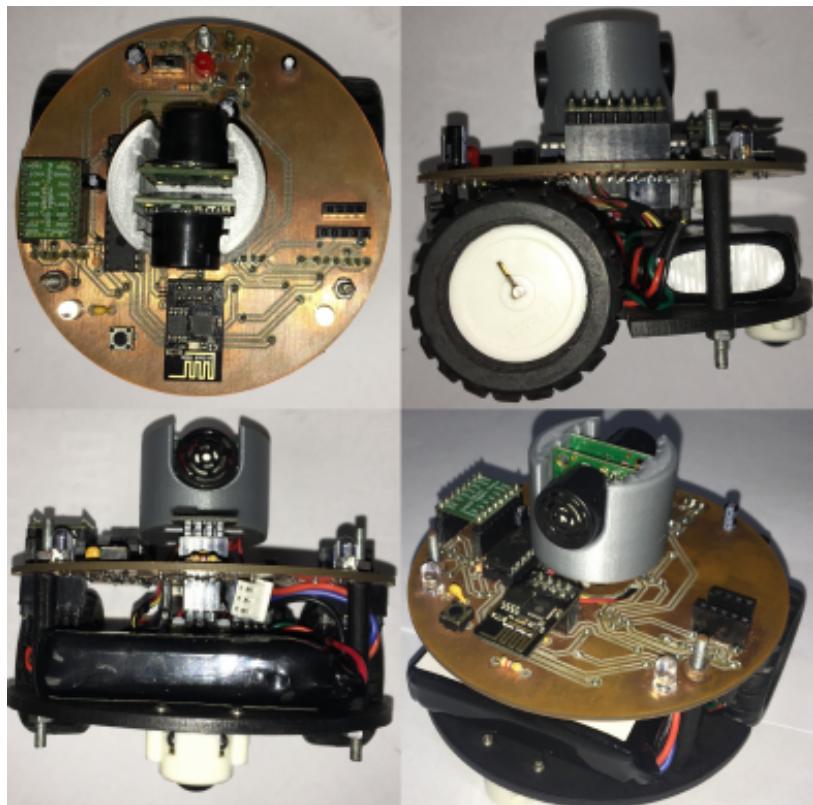


Figura 5: Bytebot fase 2 [6].

## CAPÍTULO 3

---

### Justificación

---

Una de las formas de colaboración entre seres como hormigas, abejas, etc. han sido los comportamientos por enjambre. El poder replicar este tipo de comportamiento ahora es posible gracias a la robótica y los diferentes tipos de algoritmos.

La Universidad del Valle de Guatemala ha estado trabajando en la línea de investigación de robótica de enjambre (Swarm) desde hace varios años, teniendo avances significativos de diversos algoritmos que se han desarrollado a nivel de simulación. Se busca migrar estos algoritmos a sistemas físicos. Por lo cual ahora surge la necesidad de tener plataformas robóticas más adecuadas.

Se busca realizar una nueva plataforma que pueda llegar a aprovechar los diferentes módulos implementados en fases anteriores, además de los nuevos que se estarán implementando y que estos puedan ser de ayuda a la hora del estudio de algoritmos para robótica de enjambre en diferentes aplicaciones a la vida cotidiana. El proyecto busca sentar las bases para que las personas puedan tener a su disposición un robot con capacidades básicas, pero necesarias para poder trabajar y analizar en tiempo real los comportamientos de algoritmos de robótica de enjambre.

# CAPÍTULO 4

---

## Objetivos

---

### Objetivo General

Diseñar e implementar una nueva plataforma móvil para aplicaciones de robótica de enjambre, de bajo costo y dimensiones reducidas.

### Objetivos Específicos

- Utilizar el dispositivo *Raspberry Pi* para el desarrollo de la nueva plataforma.
- Expandir la funcionalidad del módulo de control inalámbrico desarrollado en la fase anterior.
- Implementar un sistema de visión por computadora básico por medio de un módulo de cámara.
- Realizar un análisis de potencia eléctrica y costo de la plataforma.

## CAPÍTULO 5

---

### Alcance

---

Este trabajo de graduación representa la tercera fase de la plataforma desarrollada para aplicaciones en robótica de enjambre del año 2018, en esta se tienen recomendaciones para disminuir el tamaño y precio de la plataforma, además de un cambio en los sensores para una mejor optimización en las mediciones y en el costo de estos. Sin embargo, después de una investigación entre diferentes tipos de sensores los seleccionados para la fase II siguen siendo la opción más óptima.

El nuevo modelo de la plataforma se centra en una migración para poder implementarse con códigos en desarrollo dentro de la Universidad y que sean compatibles con la plataforma *Raspberry Pi Zero W*. También se centra en una mejora de las dimensiones físicas del modelo anterior, convirtiéndolo en un modelo más ordenado y en mejorar la facilidad de acceso para mantenimiento, reducción de peso en el modelo para aliviar la carga de los motores y la implementación de una fuente de poder de menores dimensiones pero que sea más duradera que la anterior.

La nueva plataforma mantiene el sistema de movimiento y detección de objetos de la fase anterior, ahora trabajado en el lenguaje *Python*. Un nuevo sistema de visión por computadora, en el cual se implementó una cámara para tomar fotografías, vídeos o realizar transmisiones en directo.

La nueva plataforma deja las herramientas de hardware listas y optimizadas para ser usadas en algún programa en específico que se acople a los módulos de esta tercera fase.

# CAPÍTULO 6

---

## Marco teórico

---

### 6.1. Robótica de Enjambre

La robótica de enjambres es un área de investigación relativamente nueva inspirada en sistemas biológicos como las colonias de hormigas o abejas. Se compone de un sistema que consta de muchos robots pequeños con mecanismos de control simples capaces de lograr comportamientos colectivos complejos a nivel de enjambre, como agregación, formación de patrones y transporte colectivo, por nombrar algunos [7].

#### 6.1.1. Particle Swarm Optimization (PSO)

Los métodos PSO se atribuyen originalmente a los investigadores Kennedy, Eberhart y Shi. En un principio fueron concebidos para elaborar modelos de conductas sociales, como el movimiento descrito por los organismos vivos en una bandada de aves o un banco de peces. Posteriormente el algoritmo se simplificó y se comprobó que era adecuado para problemas de optimización. El libro de Kennedy y Eberhart describe numerosos aspectos teóricos de la PSO y la inteligencia de enjambre. Un amplio estudio de las aplicaciones de PSO se puede encontrar en Poli. PSO permite optimizar un problema a partir de una población de soluciones candidatas, denotadas como “partículas”, moviendo éstas por todo el espacio de búsqueda según reglas matemáticas que tienen en cuenta la posición y la velocidad de las partículas. El movimiento de cada partícula se ve influido por su mejor posición local hallada hasta el momento, así como por las mejores posiciones globales encontradas por otras partículas a medida que recorren el espacio de búsqueda. El fundamento teórico de esto es hacer que la nube de partículas converja rápidamente hacia las mejores soluciones [8].

Un algoritmo PSO trabaja con una población (llamada nube o enjambre) de soluciones candidatas (llamadas partículas). Dichas partículas se desplazan a lo largo del espacio de búsqueda conforme unas simples reglas matemáticas. El movimiento de cada partícula de-

pende de su mejor posición obtenida, así como de la mejor posición global hallada en todo el espacio de búsqueda. A medida que se descubren nuevas y mejores posiciones, éstas pasan a orientar los movimientos de las partículas. El proceso se repite con el objetivo, no garantizado, de hallaren algún momento una solución lo suficientemente satisfactoria.

## 6.2. Comunicación Inalámbrica

La comunicación inalámbrica o sin cables es aquella en la que la comunicación (emisor/receptor) no se encuentra unida por un medio de propagación físico, sino que se utiliza la modulación de ondas electromagnéticas a través del espacio [9].

### 6.2.1. Protocolos TCP/IP, UDP

En el nivel de transporte de datos de protocolos TCP/IP, TCP proporciona un servicio confiable orientado a la conexión, es decir, los paquetes llegan sin error y en el orden en el que se envían.

Por otro lado, UDP es un protocolo que proporciona un servicio orientado a datagramas, no asegurando que los paquetes lleguen a su destino, y si llegaran, no garantizando su orden. UDP es un protocolo más simple que TCP, y mucho menos fiable, aunque más rápido. Es útil para aplicaciones que sean simples, que no necesiten de una transmisión fiable de datos, o incluso que necesiten que sus datos sean transmitidos lo más rápidamente posible [10].

## 6.3. Sensores de Proximidad

Un sensor de proximidad es un sensor electrónico que puede detectar la presencia de objetos en su alrededor sin contacto físico real. Para detectar objetos, el sensor de proximidad irradia o emite un haz de radiación electromagnética, generalmente en forma de luz infrarroja, y detecta el reflejo para determinar la proximidad o distancia del objeto del sensor [11].

- Sensor Inductivo
- Sensor Magnético
- Sensor Capacitivo
- Sensor infrarrojo

### 6.3.1. Sensor Ultrasónico

Los sensores ultrasónicos miden la distancia mediante el uso de ondas ultrasónicas. El cabezal emite una onda ultrasónica y recibe la onda reflejada que retorna desde el objeto.

Los sensores ultrasónicos miden la distancia al objeto contando el tiempo entre la emisión y la recepción [12].



Figura 6: Sensor de proximidad inductivo [12].

### LV-MaxSonar-EZ1

Se alimenta con un voltaje entre 2.5V - 5.5V y consume una corriente nominal de 2mA. Posee un rango de detección entre 15cm - 6.45m con una resolución de 2.54cm. Puede comunicar la respuesta por PWM, señal analógica o por comunicación Serial UART. Muestrea a una frecuencia de 20Hz. Es uno de los ultrasónicos con dimensiones físicas más pequeñas, son de 22x20x15 mm [13].

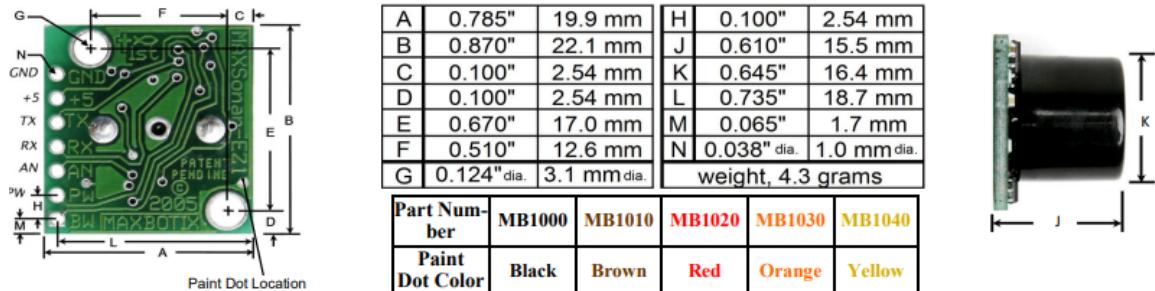


Figura 7: LV-MaxSonar-EZ1 [13].

## 6.4. Raspberry Pi

*Raspberry Pi*, una computadora del tamaño de una tarjeta de crédito eficiente y rentable, recibe la luz del sol de la fundación United Kingdom-*Raspberry Pi* con el objetivo de iluminar y potenciar la enseñanza de la informática en escuelas y otros países en desarrollo. Desde sus inicios, varias comunidades de código abierto han contribuido mucho a las aplicaciones de código abierto, sistemas operativos y otras computadoras de factor de forma pequeño similares a *Raspberry Pi*. Hasta la fecha, investigadores, aficionados y otros entusiastas de los sistemas integrados de todo el planeta están realizando proyectos asombrosos utilizando Pi, que parece increíble y tiene una implementación lista para usar. Desde su lanzamiento,

*Raspberry Pi* se encuentra en constante desarrollo y mejora tanto en términos de hardware como de software, lo que a su vez convierte a Pi en una “Computadora de pleno derecho” con la posibilidad de ser considerada para casi todas las tareas informáticas intensivas. El objetivo de este trabajo de investigación es esclarecer qué es *Raspberry Pi*, por qué se requiere *Raspberry Pi*, generaciones de *Raspberry Pi*, sistemas operativos disponibles hasta la fecha en Pi y otro hardware disponible para el desarrollo de proyectos [14].

*Raspberry Pi* admite hardware adicional como cámara, kit de módulo de componentes, placa Gertboard y placa HAT, lo que facilita la conexión de miles de hardware de terceros como botones, LED y el usuario puede realizar varias operaciones en Pi [15].

*Raspberry Pi* se considera una computadora pequeña, potente y eficiente con factor de forma compacto y es muy barata de adquirir.

Además de los dispositivos de entrada y salida simples, los pines GPIO se pueden usar con una variedad de funciones alternativas, algunas están disponibles en todos los pines, otras en pines específicos. Como por ejemplo:

- PWM(Pulse with Modulation).
- SPI(Serial Peripheral Interface).
- I2C(Inter-Integrated Circuit).
- Serial

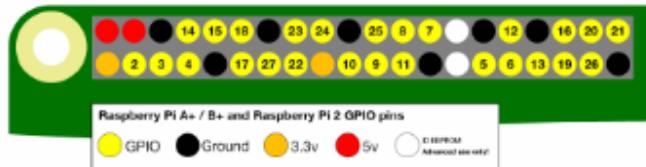


Fig. 3: Raspberry Pi GPIO board.

Figura 8: *Raspberry Pi* GPIO pines [16].

#### 6.4.1. Raspberry Pi 3 B +

*Raspberry Pi 3 B +* proporciona un conjunto de pines GPIO que permiten controlar los componentes electrónicos para la computación física. Esta versión específica monta un procesador de cuatro núcleos de 64 bits a 1.4 GHz, SDRAM LPDDR2 de 1 GB, disco SSD de 64 GB, Wi-Fi de doble banda, conexión Bluetooth y Ethernet, y un encabezado GPIO extendido de 40 pines [16].



Figura 9: Modelo fisico Modelo *Raspberry Pi* 3B+ [16].

#### 6.4.2. Raspberry Pi Zero W

Un modelo mucho más pequeño, menor potente y más barato pensado especialmente para el Internet de las Cosas (IOT). A pesar de su limitado hardware, cuenta con Wi-Fi y Bluetooth ademas, este sigue siendo un 40 por ciento más potente que el primer Rapsberry Pi [17].



Figura 10: Modelo fisico *Raspberry Pi Zero W* [17].

#### 6.4.3. Raspberry Pi 4

Este apenas tiene modificaciones en el diseño a comparación de su predecesor (*Raspberry Pi* 3 B +), conserva muchos de los principios de sus antecesoras: la posición y disposición de casi todos los conectores y chips es la misma, y en uno de los extremos tenemos el conector de red y los cuatro puertos USB, aunque en este caso están invertidos con respecto a la *Raspberry Pi* 3 Model B/+ . Muchos son los cambios a nivel interno, con su procesador de mayores núcleos, mejora en su conectividad y el aumento en puertos [16].

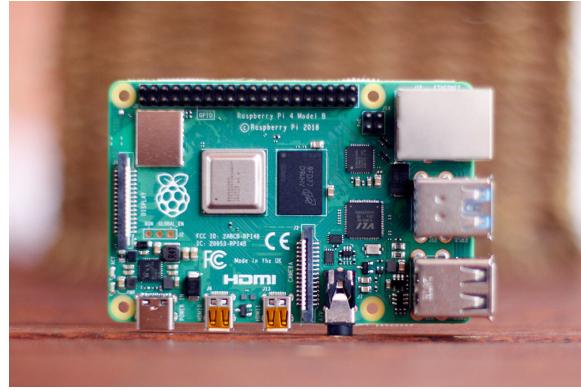


Figura 11: Modelo físico *Raspberry Pi* 4 [16].

## 6.5. Servomotor

Un servomotor es un actuador rotativo o motor que permite un control preciso en términos de posición angular, aceleración y velocidad, capacidades que un motor normal no tiene. Utiliza un motor normal y lo combina con un sensor para la retroalimentación de posición. El controlador es la parte más sofisticada del servomotor, ya que está diseñado específicamente para este fin [18].



Figura 12: Servomotor [18].

## 6.6. Motor DC

El motor de corriente continua, denominado también motor de corriente directa, motor CC o motor DC, es una máquina que convierte energía eléctrica en mecánica, provocando un movimiento rotatorio, gracias a la acción de un campo magnético [19].

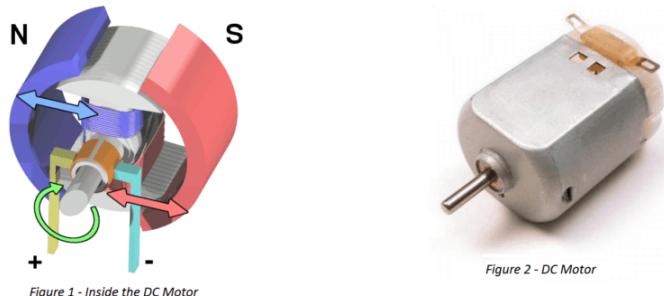


Figure 1 - Inside the DC Motor

Figure 2 - DC Motor

Figura 13: Motor DC [19].

## 6.7. Motores con caja reductora

Los motores con cajas reductoras son la combinación de un motor DC y una caja de engranajes. La adición de un cabezal de engranajes a un motor reduce la velocidad al tiempo que aumenta la salida de par. Los parámetros más importantes en lo que respecta a los motores de engranajes son la velocidad (rpm), el par (lb-in) y la eficiencia [19].



Figure 3 - DC Gear Motor

Figura 14: Motor con caja reductora [19].

## 6.8. Drivers Para Motores

Un controlador (Driver) de motor es un pequeño amplificador de corriente; la función de los Drivers de motor es tomar una señal de control de baja corriente y luego convertirla en una señal de corriente más alta que pueda conducir un motor.[20]

Los motores de corriente continua (Motor DC) se utilizan ampliamente en sistemas de velocidad variable y sistemas de control de posición en los que se requiere una buena respuesta transitoria y un buen rendimiento en estado estable. Cuando un sistema Driver controla la corriente de armadura a través de un convertidor electrónico de potencia, lo que realmente controla es el nivel de la relación velocidad-torque del motor.[20]

Según sean de los requisitos de la aplicación, el convertidor de potencia para un motor de corriente continua puede elegirse entre una serie de topologías. Por ejemplo, un convertidor de tiristor semicontrolado o un convertidor de conmutación PWM. Se usa un rectificador controlado junto con un inversor de fuente de voltaje modulado por onda cuadrada o ancho de pulso (PWM) para crear el sistema controlador de velocidad-par.[20]

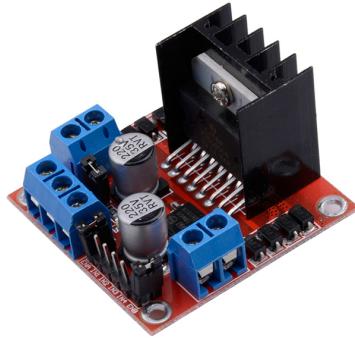


Figura 15: Driver de Motor L298N [19].

### 6.8.1. Driver TB6612FNG

El TB6612FNG es un controlador (driver) de motores que nos permite manejar dos motores de corriente continua desde cualquier microcontrolador que posea una salida PWM, variando tanto la velocidad como el sentido de giro. Internamente está formado por dos puentes-H, elimina posibles cortocircuito por errores de uso. En el caso del TB6612FNG los puentes-H están formados por transistores MOSFET.[21]

El TB6612FNG permite controlar intensidades de corriente superiores, siendo capaz de suministrar 1.2A por canal de forma continua, y 3.2A de pico.[21]

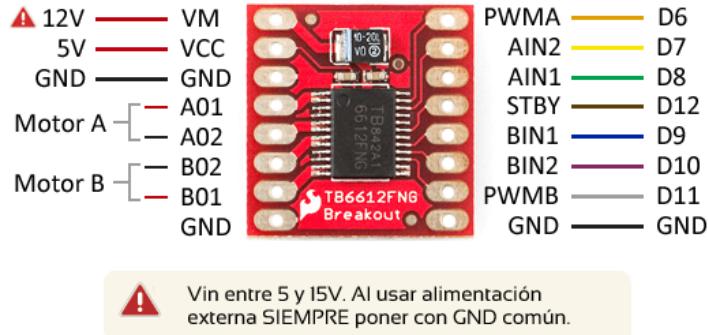


Figura 16: Driver para Motor TB6612FNG [21]

## 6.9. Baterías de Li-Ion

La batería de iones de litio, también denominada batería Li-Ion, es un dispositivo diseñado para almacenamiento de energía eléctrica que emplea como electrolito, una sal de litio que procura los iones necesarios para la reacción electroquímica reversible que tiene lugar entre el cátodo y el ánodo [22].

Las propiedades de las baterías de Li-ion, como la ligereza de sus componentes, su elevada capacidad energética y resistencia a la descarga, la ausencia de efecto memoria o su capacidad para operar con un elevado número de ciclos de regeneración, han permitido el diseño de

acumuladores livianos, de pequeño tamaño y variadas formas, con un alto rendimiento, especialmente adaptados para las aplicaciones de la industria electrónica de gran consumo [22].



Figura 17: Batería 18650 Marca Samsung de 2500mAh [22]

## 6.10. Robots Móviles

“Un robot móvil es un vehículo de propulsión autónoma y movimiento (re) programado por medio de control automático para realizar una tarea específica.” [19] Los robots móviles se clasifican en guiados y no guiados. El vehículo guiado está restringido a un conjunto de trayectorias predefinidas en su área de trabajo. [19]

Estas trayectorias están indicadas por líneas ópticas ó magnéticas o una secuencia de movimientos guardados en la memoria. El robot en ningún momento puede abandonar la trayectoria [19].



Figura 18: Robot móvil [19].

# CAPÍTULO 7

---

## Diseño electrónico de la nueva plataforma

---

En el diseño electrónico se tuvo como principal objetivo la selección de los diferentes componentes para que su funcionamiento tanto individual como en conjunto fueran los aceptados para cumplir los objetivos de cada módulo. Se tomaron en cuenta el consumo eléctrico y el espacio físico que ocupa cada uno. Se realizó una placa PCB en el programa *Altium Designer* en la cual se encuentran los componentes electrónicos. La PCB tiene como objetivo ser el sistema de conexiones eléctricas para todos los módulos, además de poderse adaptarse en futuros proyectos, únicamente rediseñando la estructura física de la plataforma, dependiendo del modelo de *Raspberry Pi* que se deseé utilizar (el modelo utilizado en este trabajo es la *Raspberry Pi Zero W*).

### 7.1. Sistema de Control

En la fase II se utilizó el microcontrolador PIC32MX250F128B. Si bien este tiene la capacidad de realizar el control de todos los módulos, se ve limitado en el no traer incluido un sistema de comunicación inalámbrica y la velocidad de respuesta que este tiene. Por este motivo y por que en la Universidad del Valle de Guatemala se encuentra trabajando en una linea de investigación con respecto a robótica de enjambre para software en los lenguajes de programación que son *Python* y C. Se decidió utilizar el controlador *Raspberry Pi*, ya que este implementa varios agregados extra que no incluye el microcontrolador de la fase pasada, como la opción de un módulo de comunicación integrado. Además de poder manejar la opción de hilos dentro del controlador y controlar diferentes tareas al mismo tiempo dentro de la plataforma.

## 7.2. Módulo de detección

Siguiendo la fase II se decidió seguir optando por el sistema generado en la fase anterior, el cual consiste en 2 sensores de gran potencia para poder recorrer gran distancia y un servomotor que logra realizar un giro de hasta  $180^{\circ}$  para lograr un barrido dentro del plano XY.

El servomotor a utilizar fue considerado en base a el tamaño de este, ya que se desea que el tamaño de la plataforma sea el más compacto y transportable para ser utilizado en ambientes controlados dentro de la Universidad.

## 7.3. Sistema de movimiento

Para que la plataforma pueda desplazarse y analizar diferentes datos desde distintos puntos de una superficie. Es necesario implementar un sistema de movimiento el cual incluye una *ball caster* [23] que sirve para ser un punto de apoyo para el direccionamiento por medio de los motores, 2 motores de 12 V 600 RPM los cuales cuentan con una caja reductora y que vienen implementados con *encoders* magnéticos. Cualquiera de los modelos de *Raspberry Pi* no pueden llegar a poner en funcionamiento el motor DC por si solo, por lo que se utiliza el controlador de motores TB6612FNG[21] para manejar el sentido de los motores y la velocidad esta ultima por medio de señales PWM. Para garantizar el funcionamiento de los motores estos son alimentados por una batería de Li-ion 18650 de 3.7 V 9800 mAh. En la Figura 19 se presenta el diagrama esquemático del sistema de movimiento.

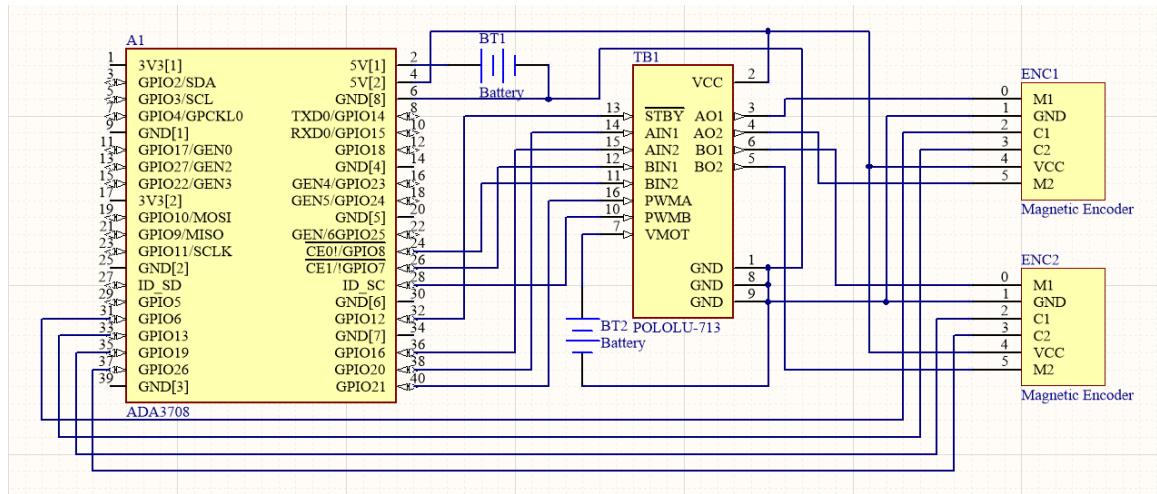


Figura 19: Esquema generado en *Altium Designer* [24].

Los pines a utilizar dentro de la *Raspberry Pi Zero W* son los siguientes:

- GPIO18: Movimiento hacia adelante Motor A (AIN1)
- GPIO23: Movimiento hacia atrás Motor A (AIN2)
- GPIO24: Control de velocidad para Motor A (PWMA)

- GPIO25: Movimiento hacia adelante Motor B (BIN1)
- GPIO8: Movimiento hacia atrás Motor B (BIN2)
- GPIO7: Control de velocidad para Motor B (PWMB)
- GPIO1: Pin de requisito para funcionamiento del driver (STBY)
- GPIO12: Señal A1 del *encoder* para el motor A
- GPIO16: Señal A2 del *encoder* para el motor A
- GPIO20: Señal B1 del *encoder* para el motor B
- GPIO21: Señal B2 del *encoder* para el motor B

## 7.4. Módulo de visión por computadora

Para la creación de un sistema de visión por computadora existen diferentes opciones para implementar en una *Raspberry Pi*, como el poder adaptarle una *webcam* y crear alguna librería para poder manejar datos de fotos y vídeo. Sin embargo, para mayor adaptación y aprovechamiento del espacio se escogió la *Raspberry Pi: Pi Noir Camera V2* desarrollada por RPi Foundation [24]. Esta es una cámara adaptable a la mayor cantidad de ambientes de luz, con total compatibilidad con el sistema de *Raspberry Pi* por lo que es ideal para poder implementarla en los proyectos a desarrollar. En la Figura 20 se observa el modelo físico utilizado para la plataforma.



Figura 20: *Raspberry Pi* Camera Module 2 NoIR[24].

Para la conexión de esta con la *Raspberry Pi Zero* no se utiliza ningún puerto de los GPIO, sino que la conexión se realiza por medio de un cable flex con los pines necesarios para conectarlos al puerto de cámara que ya viene instalado de fabrica en la *Raspberry Pi*.



Figura 21: *Raspberry Pi* Camera Module V2 conectada a una *Raspberry Pi* Zero[24].

## 7.5. Módulo de detección de objetos

Siguiendo la metodología implementada en la fase II [6] se utilizan los sensores ultrasónicos LV-MaxSonar-EZ1 junto con el servomotor Sub micro Servo 3.7g este con la intención de poder aprovechar en su totalidad la función de giro de 180° que dispone. Los sensores cuentan con la capacidad de poder obtener los datos de 3 maneras, por medio de una señal analógica, por medio de PWM o UART. La forma compatible sin utilizar algún otro dispositivo de intermediario entre la *Raspberry Pi* fue por medio de la lectura en PWM en la Figura 22 se presenta la forma de conexión para utilizar la forma por PWM. El Servomotor también funciona por medio de señales PWM desde la *Raspberry Pi* con librerías por defecto. En la Figura 23 se presenta el esquema de detección de objetos producido en el programa *Altium Designer*

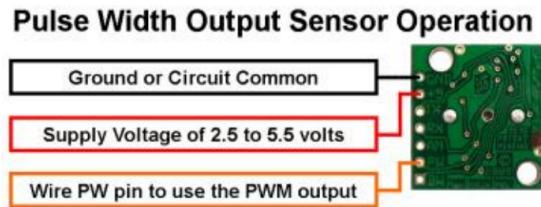


Figura 22: Conexión por medio de PWM[13]

A medida que el sonido avanza 340 metros por segundo, se toma en cuenta la distancia y se mide la distancia hacia el objeto. La distancia es obtenida multiplicando la velocidad y el tiempo ocurrido entre cada medición.

Los pines de conexión a la *Raspberry Pi* Zero son lo siguientes:

- GPIO14: Señal PWM para el Sensor ultrasónico 1.
- GPIO15: Señal PWM para el Sensor ultrasónico 2.
- GPIO2: Pin de datos PWM para el servomotor.

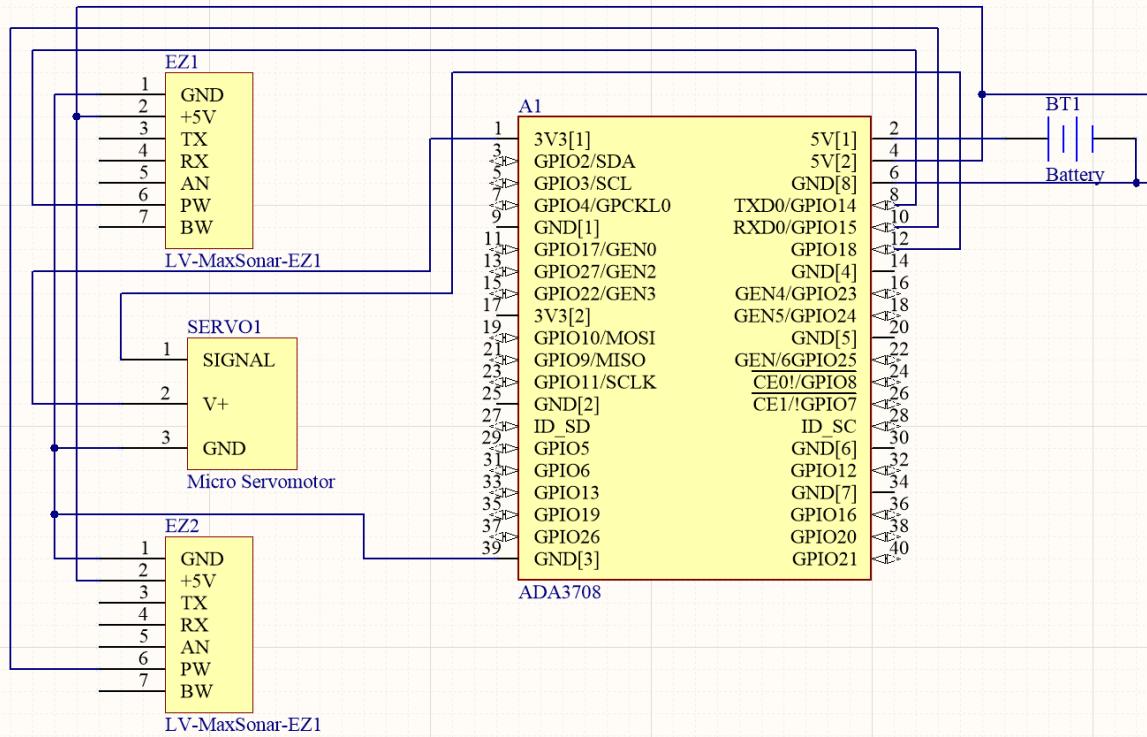


Figura 23: Esquema de detección generado en Altium Designer

## 7.6. Sistema de potencia

Con todos los sistemas implementados, se buscó que la plataforma sea controlada de manera autónoma sin necesidad de tener que conectar la plataforma con un cargador a la *Raspberry Pi*. La fuente de alimentación para la plataforma es una batería de Li-ion 18650 de 3.7 V 9800 mAh, debido a que la *Raspberry Pi* únicamente puede funcionar con un voltaje de 5 V se utiliza un regulador de voltaje a 5 V. Para los motores DC se alimentan con su propia fuente, que también es una batería Li-ion 18650 de 3.7 V 9800 mAh pero sin que haya ningún regulador. Debido a que no solo se estará conectado la *Raspberry Pi* sino que también los módulos se hace una sumatoria de la exigencia promedio de cada posible componente que consuma corriente.

Sabiendo el consumo total que será necesario para el funcionamiento de la plataforma, se calculó el tiempo que ofrece una batería 18650 de 3.7 V 9800 mAh. Se asume que la capacidad práctica de la batería se presenta entre un 50-75 por ciento con respecto de su capacidad nominal presentada por el fabricante por lo que se pone en el peor escenario del 50 por ciento [25], además se le multiplica un factor de 70 por ciento que representa la eficiencia del regulador de voltaje con una batería de 3.7 V.

$$\frac{\text{CapacidadDeLaBateria}}{\text{ConsumoBateriaB}} \times 0.5 \times 0.75 \rightarrow \frac{9800\text{mAh}}{600\text{mA}} \times 0.5 \times 0.75 = 5\text{horas} \quad (1)$$

$$\frac{\text{CapacidadDeLaBateria}}{\text{ConsumoBateriaA}} \times 0.5 \rightarrow \frac{9800\text{mAh}}{100\text{mA}} \times 0.5 = 40\text{horas} \quad (2)$$

Cuadro 1: Consumo de corriente de la nueva plataforma

Fuente de poder	Bateria A	Bateria B
<i>Controlador</i>		
Raspberry Pi Zero W		200mA
<i>Motores</i>		
Sub-Micro Servo 3.7g		150mA
Motor Reductor Mini 12VDC 600RPM	100mA	
<i>Componentes electronicos</i>		
TB6612FNG Dual Motor Driver Carrier		20mA
Raspberry Pi NoIR - Módulo de cámara V2		100mA
Gravity: MiniQ Robot Chassis <i>encoder</i>		100mA
LV-MaxSonar-EZ1		2mA
<b>Total</b>	<b>100mA</b>	<b>600mA</b>

El regulador de voltaje a utilizar se conectó directamente a los pines de la *Raspberry Pi* para poder utilizarlo, este tiene un porcentaje del 90 de eficiencia, siendo lo demás disipado en calor.

- PIN 2(5V): Pin OUT+
- PIN 6(GND): Pin OUT-

## 7.7. Esquemático final

El esquemático final se muestra en la Figura 24. Se aprecia la conexión entre los distintos módulos utilizados. En la figura 25 se muestra la conexión esperada con todos los componentes físicos (La batería AA es únicamente tomada como referencia dentro del programa para generar el diagrama, la batería utilizada es la 18650 3.7 V 9800 mAh).

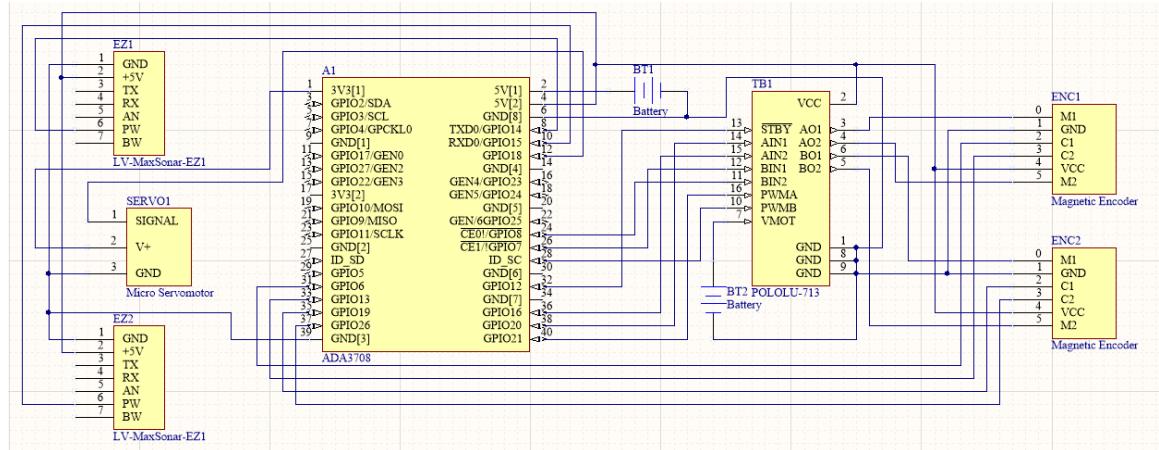


Figura 24: Esquemático final hecho en Altium Designer.

Se detallan el uso de todos los pines conectados a la *Raspberry Pi Zero W*:

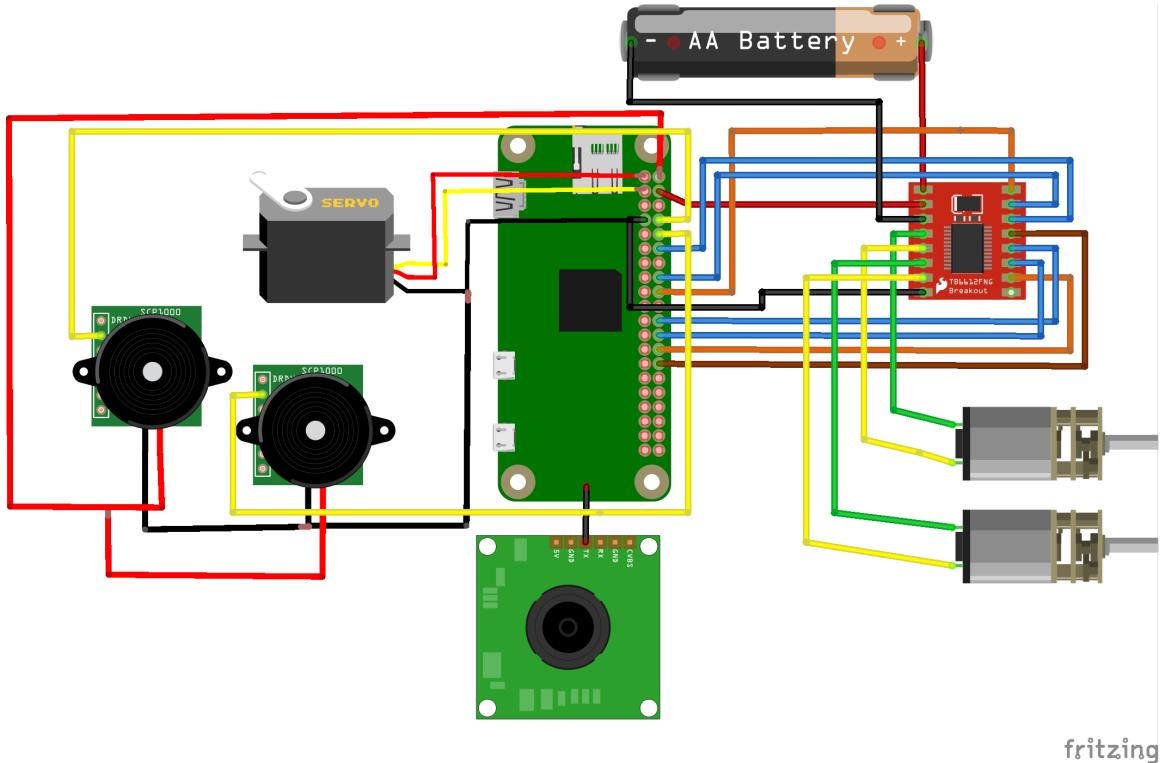


Figura 25: Esquemático final hecho en *Fritzing*.

- PIN 2(5V): Pin OUT+ recibe la señal de 5V
- PIN 6(GND): Pin OUT- recibe la señal de GND
- GPIO14: Señal PWM para el Sensor ultrasónico 1.
- GPIO15: Señal PWM para el Sensor ultrasónico 2.
- GPIO2: Pin de datos PWM para el servomotor.
- GPIO18: Movimiento hacia adelante Motor A (AIN1)
- GPIO23: Movimiento hacia atrás Motor A (AIN2)
- GPIO24: Control de velocidad para Motor A (PWMA)
- GPIO25: Movimiento hacia adelante Motor B (BIN1)
- GPIO8: Movimiento hacia atrás Motor B (BIN2)
- GPIO7: Control de velocidad para Motor B (PWMB)
- GPIO1: Pin de requisito para funcionamiento del driver (STBY)
- GPIO12: Señal A1 del *encoder* para el motor A
- GPIO16: Señal A2 del *encoder* para el motor A
- GPIO20: Señal B1 del *encoder* para el motor B

- GPIO21: Señal B2 del *encoder*para el motor B

# CAPÍTULO 8

---

## Diseño mecánico de la nueva plataforma

---

Esta parte se realizó con ayuda del software Autodesk Inventor Professional 2022. Se llega a la creación de distintas iteraciones para el modelo final. Se diseño por medio de bocetos la geometría deseada, para luego por medio de extrucciones dentro del mismo programa se puedan llevar los bocetos a 3D. Para agilizar el proceso de fabricación de la plataforma únicamente se utilizo la herramienta de Inventor para obtener los planos de la base, para poder hacer uso de una cortadora láser y poder imprimir en material MDF la base, y así ensamblar todos los componentes.

### 8.1. Dimensiones físicas de la fase II

El cambio que se hizo desde la fase I a la fase II fue con el objetivo de mantener la forma de un círculo, esto manteniendo una forma muy similar a las referencias como el E-puck[1], las dimensiones del robot son de 10 cm de diámetro y 8.36 cm de alto. El área (incluyendo las llantas) que ocupa en el plano XY, es de  $84.2 \text{ cm}^2$ .

### 8.2. Base de la nueva plataforma

Para la nueva base se tomo como inspiración inicial el diseño realizado por el proyecto *Zerobot* [26] mostrado en la Figura 26, ya que este contaba con dimensiones reducidas que era de los primeros objetivos del proyecto, por lo que en una primera iteración se llegó a una imagen inicial como es en el Figura 27.

Se tuvieron deficiencias en este tipo de diseño. Si bien las funcionalidades y los componentes que eran la *Raspberry Pi Zero W*, una batería de litio 18650, motores DC, el agregar

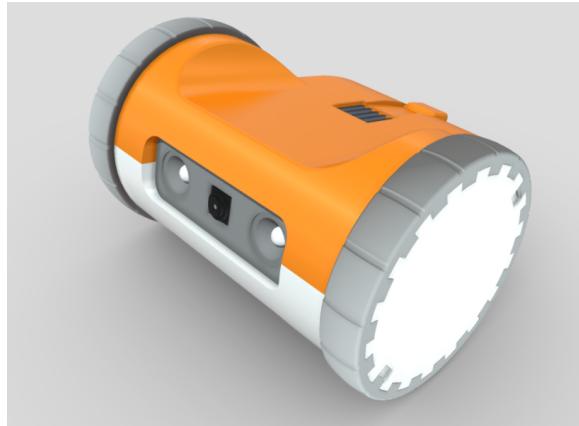


Figura 26: Modelo 3D del proyecto *Zerobot* [26].

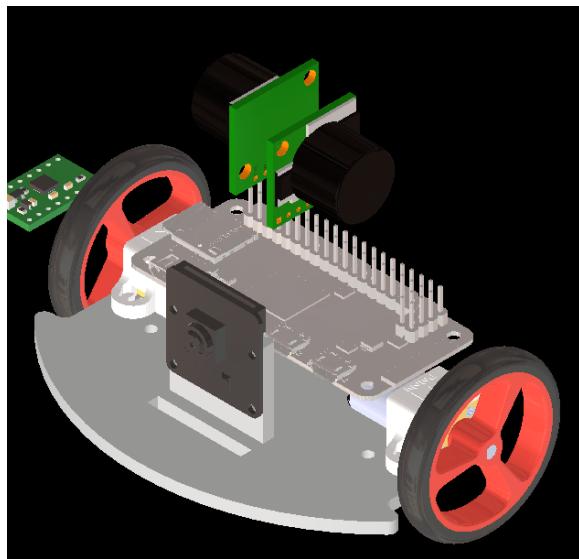


Figura 27: Primera iteración.

los sensores LV-EZ1, un servomotor y los encoders magnéticos, requería de un espacio mucho mayor, por lo que este diseño fue descartado.

Como segunda iteración se llegó a la conclusión de la necesidad de un diseño similar aplicado a la I y II fase en el desarrollo de una nueva plataforma para algoritmos de robótica de enjambre. Con una base hecha de MDF de 3.00 mm únicamente pensada para colocar los componentes se llegó a la siguiente iteración, ilustrada en la Figura 28.

En la Figura 28 ya se puede observar cómo se aprovecha mejor el espacio pensando en las necesidades de potencia y compatibilidad para la plataforma, siendo el problema principal el posicionamiento de los sensores LV-EZ1 y el servomotor. Este diseño quedó descartado por no poder resolver de manera efectiva la necesidad de posicionamiento de todos los componentes necesarios.

En la tercera iteración para el diseño de la nueva plataforma presentada en la Figura 29, este toma como base el diseño de la segunda iteración, se logró hacer un reacomodo de los

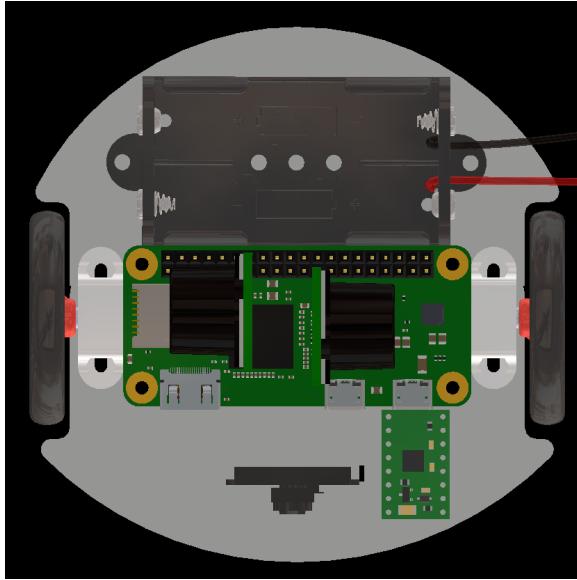


Figura 28: Segunda iteración.

componentes además de reducción de dimensiones. La *Raspberry Pi* ahora se ve por la parte de abajo, está sujetada por tornillos M2 × 10 mm de cabeza de queso, en esta iteración se le agrega una *casterball* para lograr que la plataforma tenga estabilidad. La cámara se acomoda en la parte de enfrente y se mantienen 2 baterías 18650 de 9800 mAh de 3.7 V para que la plataforma sea operada por estas baterías se tiene un módulo regulador de voltaje a 5 V siendo esto suficiente para poder alimentar a toda la plataforma. En la Figura 30 se presenta el modelo 3D generado en *Inventor* desde una vista inferior y en la Figura 31 se presenta el modelo 3D generado en *Inventor* desde una vista superior.

Este diseño, a diferencia de la fase II incluye una protección trasera para las llantas, evitando posibles choques traseros que puedan afectar a las llantas de manera directa. Se hace un cambio en el grosor de las llantas, ya que esta plataforma esta planeada para usarse en ambientes controlados para pruebas de robótica de enjambre, por lo que se cambia las llantas robustas de la fase anterior por llantas más delgadas y más ligeras. Otro cambio es en los *encoders*, estos se cambian por *encoders* magnéticos los cuales vienen integrados directamente en los motores. Se hace el ajuste para la implementación por separado de las baterías 18650, y se deja el espacio necesario para que la PCB se ajuste a la medida junto a la base, así aprovechando al máximo el espacio y dimensiones de la plataforma.

### 8.3. Estructura final

Teniendo la placa base finalmente impresa, junto a la PCB estos se sujetaron usando tornillos de 2mm con cabezas cónicas y cabeza de queso, la conexión en conjunto a los componentes se realizaron utilizando cable de 1 hilo medida No.32. El sujetador utilizado para los sensores ultrasónicos fue el diseño implementado en la fase anterior. El diseño final completo se ve en una vista superior en la Figura 32 , en una vista inferior para la Figura 33 y una vista frontal presentada en la Figura 34. Los planos presentados con las medidas

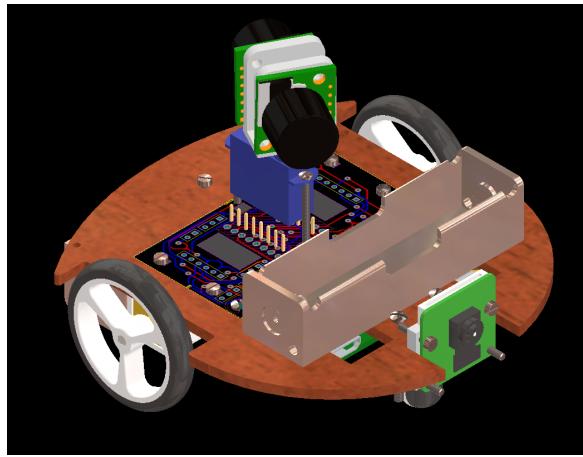


Figura 29: Tercera iteración modelo 3D.

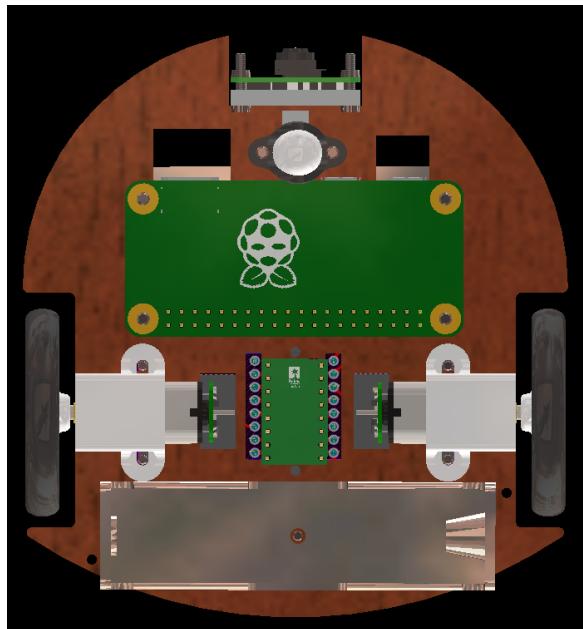


Figura 30: Modelo 3D Vista inferior.

exactas generadas en *Inventor* se presentan en la Figura 35

#### 8.4. Diseño PCB

Se realizó el diseño en el programa *Altium Designer*. Debido a que ninguno de los componentes tuvo que manejar una corriente mayor a los 150mA el ancho mínimo para las pista fue de 20 mils. Los ambientes considerados fueron una temperatura entre 20° y 25° grados, y voltajes entre 3.3 V y 5 V. La placa fue realizada en doble cara, esto para facilitar la implementación en la base hecha en MDF. En la parte inferior del PCB se encuentran la mayoría de soldaduras de todos los *headers*, además de las entradas de *headers* para la *Raspberry Pi* y los *headers* para el *driver* TB662FNG, ninguno de los componentes está directamente

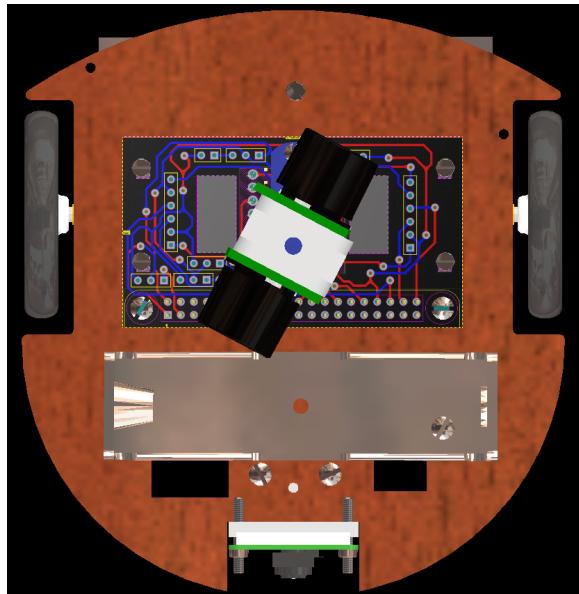


Figura 31: Modelo 3D Vista superior.

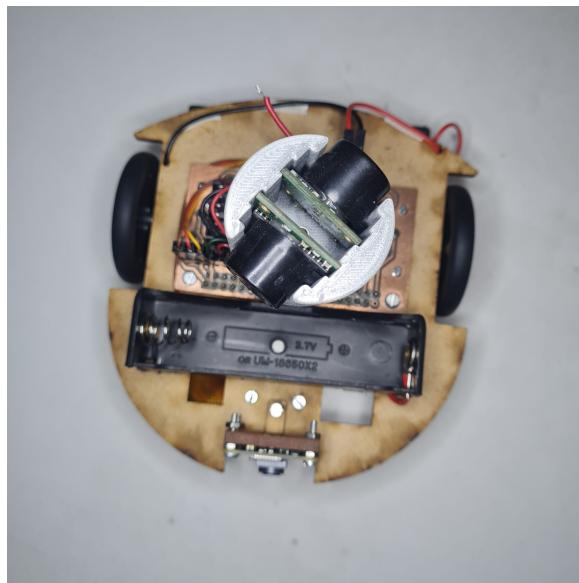


Figura 32: Modelo impreso Vista superior.

soldado al PCB para facilitar a futuros usuarios el poder cambiar componentes arruinados o actualizar los módulos sin tener que modificar ninguna parte de la PCB. En la parte inferior de la Figura 36 se encuentran las entradas para los *headers* que conectan con los *encoders* de los motores, la conexión con la batería, los sensores y el servomotor, el cual está posicionado con 2 tornillos de M2 × 25 para sostenerlo.

La PCB tiene dimensiones de 6.5 cm de largo y 3.5 cm de altura. Este diseño contempla la posibilidad de utilizar diferentes *Raspberry Pi*, ya que los pines de conexión son compatibles con otros modelos. La única modificación se debe de realizar en la base de MDF. En la Figura 37 se presenta la PCB con tomado en cuenta el espacio con una *Raspberry Pi Zero*

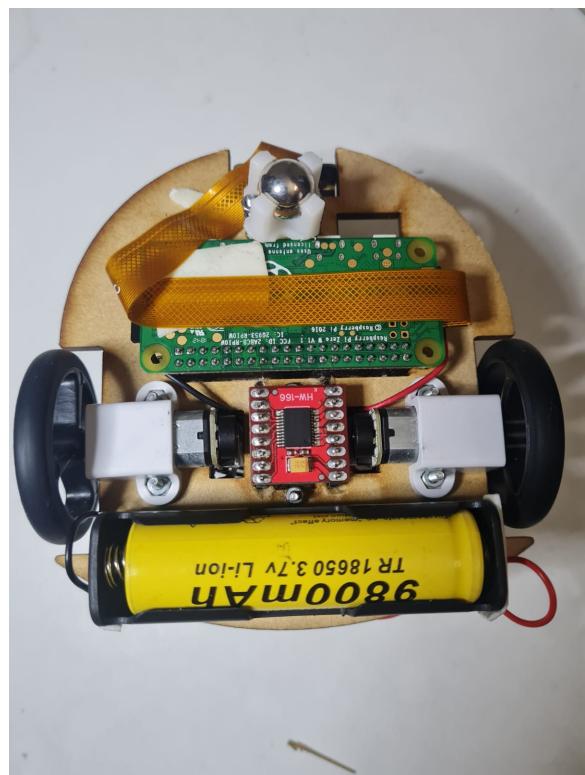


Figura 33: Modelo impreso Vista inferior.

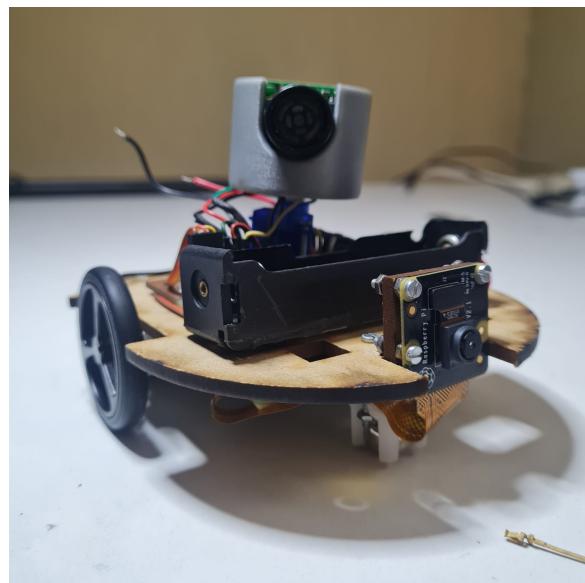


Figura 34: Modelo impreso Vista frontal.

*W.*

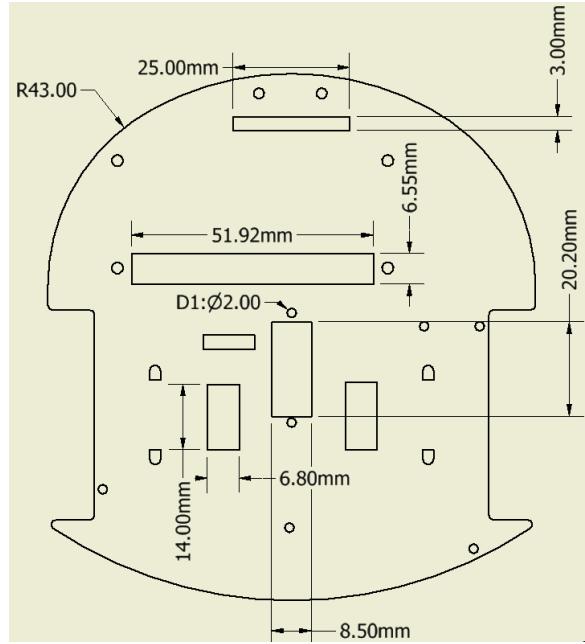


Figura 35: Planos de la base generados en Inventor.

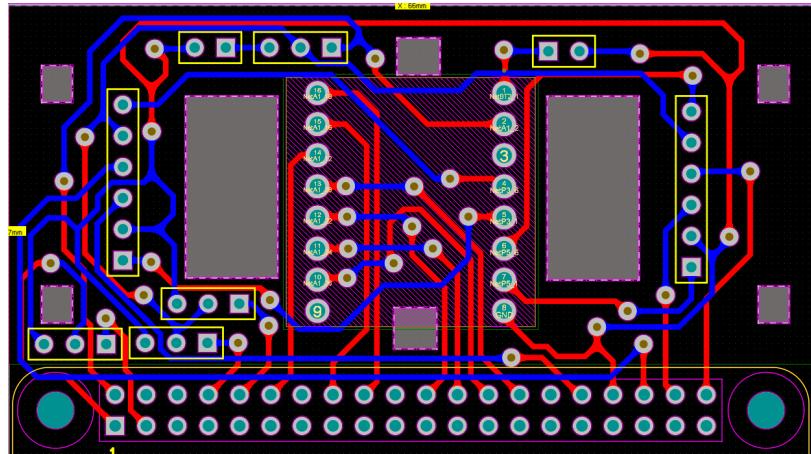


Figura 36: PCB realizada en Altium Designer.

## 8.5. Comparación plataforma fase II y fase III

Para la nueva plataforma se toman cambios bastante significativos como el cambio de microcontrolador, *encoders*, motores, servomotor y sección de potencia. Estos nuevos componentes implementados siendo la *Raspberry Pi Zero W*, el uso de una batería más potente de 9800mAh, motores integrados con *encoders* y la implementación de un sistema de detección de imágenes. En la parte estructural los cambios realizados son el cambio de una forma geométrica por una figura que asemeja a una elipse, la PCB no ocupa el mismo espacio que la figura realizada en MDF y la nueva plataforma únicamente esta conformada por un nivel. La nueva plataforma únicamente eleva su precio en comparación a la plataforma pasada un 16 %. Se logran reducir aspectos como el tamaño y peso de la plataforma, la pasada siendo

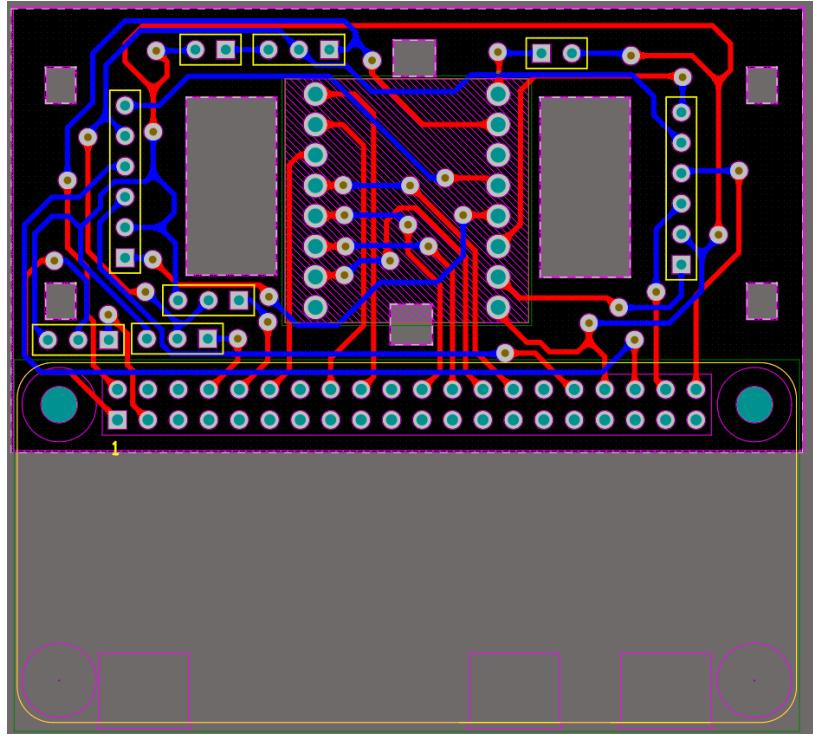


Figura 37: PCB con el modelo de *Raspberry Pi Zero W*.

de 90 mm de alto y 1 kg de peso, mientras que la nueva plataforma alcanza únicamente los 70 mm de alto y 0.5 kg de peso. Esto fue importante para reducir la carga que reciben los motores y que estos puedan operar por mucho más tiempo al no tener que lograr consumir su corriente tope. La altura se redujo, ya que para las diferentes pruebas para los algoritmos de robótica de enjambre son probados en ambientes controlados y que no suelen llegar a ser muy espaciosos.

Cuadro 2: Presupuesto detallado de la nueva plataforma

No.	Cantidad	Componente	Precio
1	2	Porta batería 18650	\$1.69
2	1	Batería 18650 9800mAh	\$13.02
3	1	DC-DC 0.9-5V to 5VDC USB	\$1.56
4	1	Raspberry Pi Zero W	\$13.02
5	1	Sub-Micro Servo 3.7g	\$4.56
6	2	Motor reductor mini 12VDC 600RPM	\$11.72
7	1	TB6612FNG Dual Motor Driver Carrier	\$4.56
8	1	Raspberry Pi NoIR - Módulo de cámara V2	\$26.04
9	1	Magnetic Encoder Pair	\$3.91
10	2	LV-MaxSonar-EZ1	\$58.59
11	1	Pololu Wheel 40×7mm Pair - White	\$3.26
12	1	Pololu Ball Caster with 3/8 Plastic Ball	\$1.30
13	1	Otros(Switch,Headers,tornillos)	\$9.18
Total			\$152.41

# CAPÍTULO 9

---

## Programación de la plataforma

---

Para la programación de la plataforma se realizó completamente en *Python*. El código realizado en la plataforma esta optimizado para ser utilizado en la versión 3.7.0, en este lenguaje se crearon nuevas librerías para poder mejorar la compatibilidad y facilidad de los usuarios que deseen hacer uso de esta.

### 9.1. Programación Raspberry Pi Zero

Para la programación en *Raspberry Pi Zero W* se tienen diferentes formas y lenguajes para poder programarla. La escogida fue en el sistema operativo Raspbian contenido en una tarjeta SD de uso exclusivo solo para el sistema operativo. El sistema operativo posee la característica de poder arrancar sin necesidad de introducir ninguna contraseña. Para poder programar la placa únicamente se necesita un monitor con entrada HDMI, un mouse con salida USB y un teclado con salida USB. También se pueden implementar estos dos últimos vía *Bluethooth*. Si no se cuenta con ninguno de estos componentes también se pueden acceder a la *Raspberry Pi Zero W* por medio de conexión inalámbrica, utilizando el programa VNC.

La programación se realizó en el lenguaje *Python*. Cada módulo fue desarrollado por separado para poder comprobar el funcionamiento individual de cada uno de los módulos de la plataforma, siendo estos sistema de movimiento, sistema de detección de imágenes, sistema de detección de objetos y sistema de comunicación.

Las distintas conexiones realizadas se presentan en la Figura 39, en esta se ven los pines GPIO que se realizan para las conexiones físicas.

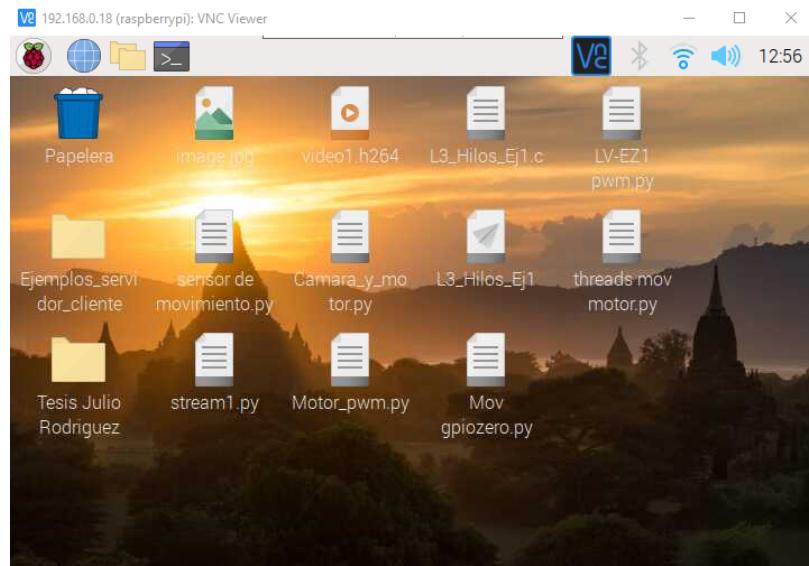


Figura 38: Pantalla inicial *Raspberry Pi Zero W* utilizando VNC

	<b>3V3</b>	(1)	(2)	<b>5V</b>	5V
Servomotor	<b>GPIO2</b>	(3)	(4)	<b>5V</b>	GND
	<b>GPIO3</b>	(5)	(6)	<b>GND</b>	GND
	<b>GPIO4</b>	(7)	(8)	<b>GPIO14</b>	Radar 1
	<b>GND</b>	(9)	(10)	<b>GPIO15</b>	Radar 2
	<b>GPIO17</b>	(11)	(12)	<b>GPIO18</b>	Adelante A
	<b>GPIO27</b>	(13)	(14)	<b>GND</b>	
	<b>GPIO22</b>	(15)	(16)	<b>GPIO23</b>	Retroceso A
	<b>3V3</b>	(17)	(18)	<b>GPIO24</b>	PWM A
	<b>GPIO10</b>	(19)	(20)	<b>GND</b>	
	<b>GPIO9</b>	(21)	(22)	<b>GPIO25</b>	Adelante B
	<b>GPIO11</b>	(23)	(24)	<b>GPIO08</b>	Retroceso B
	<b>GND</b>	(25)	(26)	<b>GPIO07</b>	PWM B
	<b>GPIO00</b>	(27)	(28)	<b>GPIO01</b>	STBY
	<b>GPIO05</b>	(29)	(30)	<b>GND</b>	
	<b>GPIO06</b>	(31)	(32)	<b>GPIO12</b>	Encoder A1
	<b>GPIO13</b>	(33)	(34)	<b>GND</b>	
	<b>GPIO19</b>	(35)	(36)	<b>GPIO16</b>	Encoder A2
	<b>GPIO26</b>	(37)	(38)	<b>GPIO20</b>	Encoder B1
	<b>GND</b>	(39)	(40)	<b>GPIO21</b>	Encoder B2

Figura 39: Configuración de pines

### 9.1.1. Sistema de movimiento

Para la programación de este sistema se utilizaron 3 pines para cada motor, 1 para la señal PWM, conectado directamente al *driver* del motor, para la configuración del PWM únicamente es necesario agregar la librería PWMOutputDevice que ya viene integrada en las librerías instaladas por defecto en el sistema Raspbian y los otros 2 pines controlan el movimiento del motor hacia adelante o hacia atrás. Estas son señales digitales, por lo que

también se debe agregar la librería DigitalOutputDevice.

Para el control de la velocidad únicamente es necesario especificar el porcentaje necesario para el ciclo de trabajo en el PWM, y para el sentido de giro de cada llanta lo que se debe de hacer es enviarle una señal positiva a alguno de las entradas, mientras la otra esta apagada como se establece en el Cuadro 3.

Cuadro 3: Configuración de giro para las llantas.

<b>GPIO20</b>	<b>GPIO16</b>	<b>GPIO01</b>	<b>GPIO07</b>	<b>Left wheel</b>	<b>Right Wheel</b>
L	H	x	x	Forward	x
H	L	x	x	Reverse	x
x	x	L	H	x	Forward
x	x	H	L	x	Reverse

Como se observa en el cuadro 4, las diferentes funciones para el sistema de movimiento cumplen con la mayor parte de movimientos esperados para una plataforma de este tipo.

Cuadro 4: Comandos para los distintos movimientos en las llantas.

<b>Comando</b>	<b>Función</b>
<i>AllStop()</i>	Detiene el movimiento de ambos motores
<i>forwardDrive()</i>	Ambos motores van hacia adelante a maxima velocidad
<i>reverseDrive()</i>	Ambos motores van hacia atras a maxima velocidad
<i>spinLeft()</i>	Se cambia de posición hacia la izquierda
<i>spinRight()</i>	Se cambia de posición hacia la derecha
<i>forwardTurnRight()</i>	Se avanza hacia la izquierda
<i>forwardTurnLeft()</i>	Se avanza hacia la derecha

### 9.1.2. Sistema de visión por computadora

Para este sistema se utilizo la cámara V2.1 NoIR (No Infrared filter). Esto se refiere a las fotos que se tomen a la luz del día tendrán un filtro, pero que si se aplica con luz nocturna se tendrán buenos resultados. Se le aplica un filtro para tomar las imágenes con el espectro de colores naturales para el día.

Se debe habilitar la opción de usar la cámara dentro de la *Raspberry Pi*. Una vez habilitado se tienen diferentes comandos que se pueden meter dentro de la consola de programación para la toma de imágenes o toma de videos. La cámara se configuro a una resolución de 640x480 y cuenta con opción para poder voltear la imagen para posicionar físicamente la cámara en donde se le pida.

Además de las funciones para tomar fotos o videos se incluyó la parte de realizar grabaciones en directo para servidores locales, esto utilizando las librerías de *sockets* y la de servidor. Al igual que en video las transmisiones en directo suceden hasta 30 cuadros por segundo por lo que se da una imagen bastante fluida.

Para hacer posible la transmisión en directo se necesita que la *Raspberry Pi* este conec-

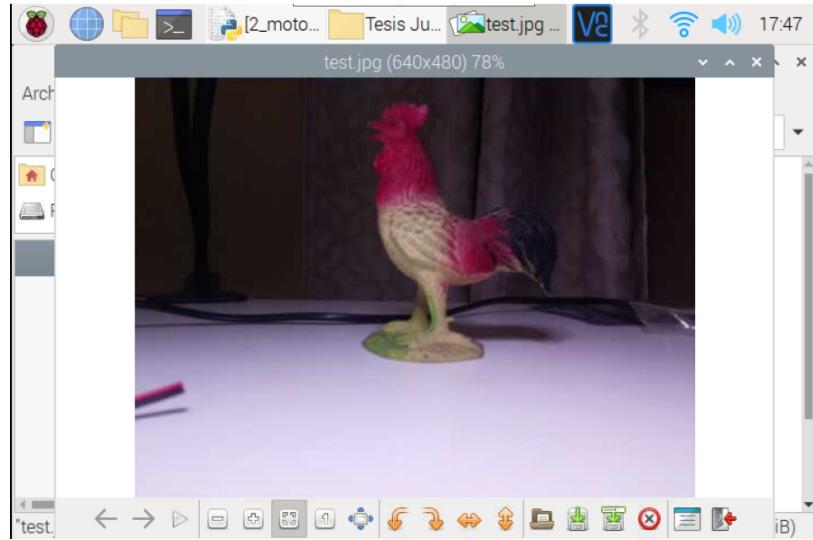


Figura 40: Fotografía tomada con resolución  $640 \times 480$ .

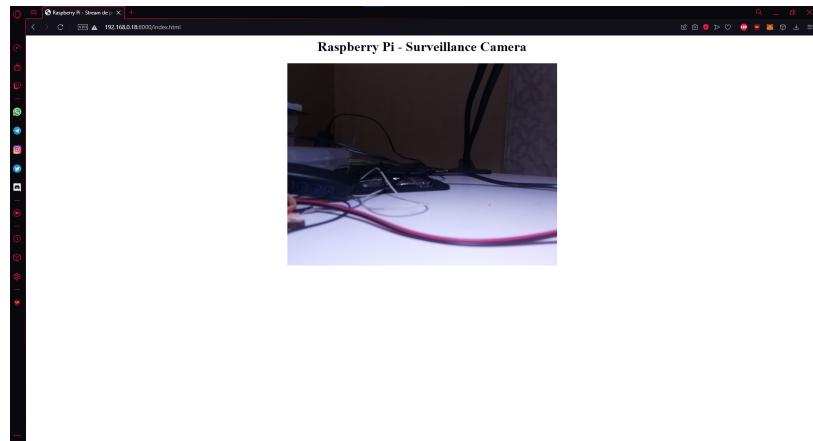


Figura 41: Transmisión en vivo desde un servidor local.

tada vía Wifi a la misma red en la que esta el otro dispositivo en donde se desea visualizar la transmisión. Además, se le debe de asignar una dirección de *broadcast* local para inicializar el envío o escucha de datos que empieza desde la *Raspberry Pi*.

Comando	Función
<i>Foto()</i>	Esta función permite que la cámara tome una fotografía en resolución $640 \times 480$
<i>Video(X)</i>	Esta función permite que la cámara tome un vídeo, en donde X es la cantidad de tiempo que durará el vídeo en formato H.264
<i>Stream()</i>	Esta función permite que la cámara realice una transmisión en vivo hacia la dirección IP de la plataforma en el puerto asignado

### 9.1.3. Comunicación Inalámbrica

En esta parte se realizó una comunicación cliente y un servidor dentro de una red local, utilizando una configuración de sockets por medio de una conexión UDP. El cliente es la *Raspberry Pi*, en el cual este procede a recibir y enviarle datos al servidor. El cliente recibe órdenes desde el servidor como los son comandos para control de los motores, extracción de distancia por medio de los sensores de movimiento, ángulo de giro para el servomotor, toma de fotos, vídeo y realización de transmisiones en directo. Debido a que la comunicación UDP es un protocolo de enviar datos sin esperar que alguien los reciba se optó por utilizar el servicio de broadcast, el cual nos permite utilizar especificar una IP global a la cual todos los dispositivos que estén escuchando llegarán a recibir los datos.

Comando	Función
<code>receive_file_size(sock)</code>	Función para cuantificar la cantidad de datos a recibir
<code>receive_file(sock, filename)</code>	Recibe datos: fotos, vídeos, archivos de texto.

### 9.1.4. Sistema de detección

En este sistema se toman los sensores utilizados anteriormente para la fase II. Sin embargo, la manera para leerlos es distinta, ya que ahí se mapean las señales PWM para poder hacer una comparación desde el microcontrolador y poder decir la distancia desde cada toma. Para esta nueva medición se decidió utilizar tiempos de inicio y final. La fórmula a utilizar es el tiempo de inicio cuando el sensor detecta un cambio y el tiempo final de este.

$$Distancia = \frac{(T_{final} - T_{inicio}) * 3400}{2} \quad (3)$$

El programa utilizado es usando el sensor ultrasónico, a medida que el sonido avanza a 3400 cm/s se tiene esta consideración para calcular la distancia, 3400cm/s entonces la distancia puede ser obtenida multiplicando la velocidad por la diferencia entre el tiempo final y el tiempo inicial de mediciones.

Comando	Función
<code>measureX(Y)</code>	Esta función pone en funcionamiento la detección de objetos, 'X' corresponde al sensor que se está midiendo y 'Y' a la cantidad de tiempo entre cada medición y cuyo resultado es en centímetros

# CAPÍTULO 10

---

## Pruebas de la plataforma

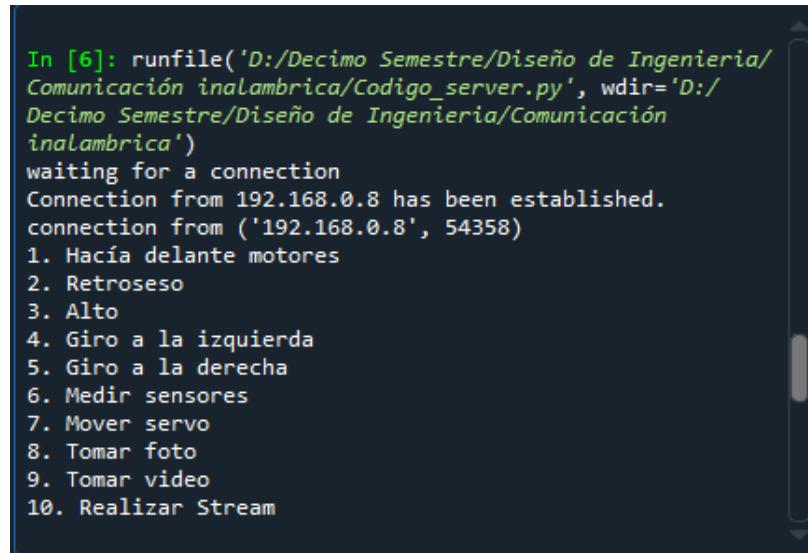
---

Para la realización de pruebas el objetivo fue el funcionamiento de todos los módulos de la plataforma. Que estos brindaran los datos esperados como respuesta de los motores, la medición de distancia, obtención de fotos, vídeo, transmisión en directo y comunicación a diferentes receptores de la información.

### 10.1. Comunicación por medio TCP y UDP

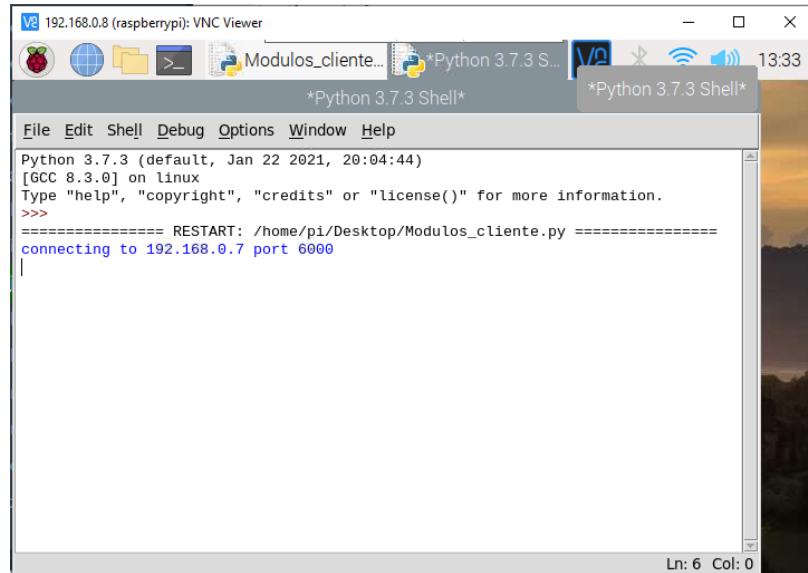
Debido a que esta plataforma únicamente se presenta como medio para futuros trabajos se deja la opción de realizar una comunicación inalámbrica para el envío de datos de diferentes formas. La primera forma de comunicación es por medio de conexión TCP, en la cual consiste en una comunicación directa entre un cliente y servidor. En esta se tiene la conexión directa del cliente con el servidor por medio de la IP del servidor. Es una conexión en la que obligadamente el servidor debe estar operando antes de que alguno de los clientes pueda conectarse, cuando el servidor funciona con normalidad este detecta la dirección IP del cliente que se está conectando, y permite que exista una comunicación directa entre ambos. Si el cliente deja de intentar conectarse, el servidor sólo queda a la espera si existe alguna otra plataforma o cliente que desee recibir órdenes directas. Si el servidor deja de funcionar automáticamente corta la funcionalidad de comunicación del cliente, ya que este estaba a la espera de alguna instrucción. En la Figura 42 se muestra cómo en principio, al ejecutar el programa, este espera a realizar la conexión con el cliente. De lo contrario no se puede llegar a realizar ninguna otra acción, al conectarse con el cliente que corresponde a la IP: 192.168.0.8 (dirección únicamente para este ejemplo específico), en este se despliega un menú de opciones que se mandan desde la consola del servidor para que sean ejecutadas en la plataforma que contiene todos los módulos (cliente). Para la Figura 43 se nos muestra la vista desde el servidor, el programa ejecutado en el IDE de *Python*, en principio este busca

conectarse al servidor en la IP de este, al poder realizar de manera existosa la conexión se informa que estará conectado a la dirección IP: 192.168.0.7 en el puerto 6000 (dirección IP y puerto especificados únicamente para los ejemplos presentados en estas pruebas).



```
In [6]: runfile('D:/Decimo Semestre/Diseño de Ingeniería/Comunicación inalámbrica/Código_server.py', wdir='D:/Decimo Semestre/Diseño de Ingeniería/Comunicación inalámbrica')
waiting for a connection
Connection from 192.168.0.8 has been established.
connection from ('192.168.0.8', 54358)
1. Hacia delante motores
2. Retroceso
3. Alto
4. Giro a la izquierda
5. Giro a la derecha
6. Medir sensores
7. Mover servo
8. Tomar foto
9. Tomar video
10. Realizar Stream
```

Figura 42: Comunicación vía TCP: Servidor operacional



```
V 192.168.0.8 (raspberrypi): VNC Viewer
File Edit Shell Debug Options Window Help
Python 3.7.3 (default, Jan 22 2021, 20:04:44)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=====
RESTART: /home/pi/Desktop/Modulos_cliente.py
=====
connecting to 192.168.0.7 port 6000
```

Figura 43: Comunicación vía TCP: Cliente conectado al servidor

El segundo método para la comunicación para posibles trabajos de robótica de enjambre es la conexión por medio de UDP que es más simple. Para UDP no es necesario que exista esa diferenciación entre cliente-servidor, no se depende de una confirmación entre un dispositivo con el otro para realizar envío de datos, tampoco se debe aceptar nuevas conexiones para mantener el programa en funcionamiento. Únicamente trata de asociarse a una IP que no esté ocupada dentro la conexión de red inalámbrica. Los dispositivos únicamente se unen a la dirección IP asignada anteriormente para poder fluir sus datos por ese canal, ya solo depende de si alguien está escuchando para recibir esos datos o no. En UDP se tiene la característica de poder realizar transmisión por medio de *broadcast* la cual es una dirección especial (no asociada a un dispositivo específico). Típicamente es de la forma X.X.X.255, su principal característica es el poder enviar un mensaje a esa dirección especial y que todos los dispositivos que se encuentren en la red y estén esperando a recibir un mensaje lo llegan a recibir. Para la configuración de todos los clientes se realiza una asociación a la IP de *broadcast* junto al puerto a utilizar. Ese es el único requisito para poder empezar a recibir y enviar datos, las funciones para enviar mensajes por UDP son diferentes con respecto a las de TCP. Para TCP solo hay un receptor, en UDP se debe de especificar por si se desea enviar a alguna otra dirección, la función de recibir datos es igual para UDP como para TCP, en este no existe ningún cambio para el uso que se le pueda dar estos ejemplos.

Para verificar el funcionamiento correcto de la transmisión por vía UDP se utilizaron 3 dispositivos. El de la Figura 44 se utiliza como medio de conexión el cual le manda datos de órdenes para cualquier dispositivo que esté escuchando. Este garantiza que un dispositivo controlado por un usuario pueda pedirle a la plataforma los datos de los sensores, movimiento del servomotor, la toma de fotos, videos o realización del *stream* o que logre controlar los movimientos del robot.

El segundo dispositivo de la Figura 45 es la plataforma que ha sido creada. En esta se encuentran integrados todos los módulos descritos anteriormente. Su función principal es la de recibir órdenes para completar un objetivo, el cual puede ser la detección de objetos, el uso del módulo de visión por computadora o realizar una tarea de movimiento de un punto A hacia un punto B. En la Figura 45 se observa cuando recibe la orden de tomar una fotografía y mandarla.

El último dispositivo es una *Raspberry Pi* 3 con la exclusiva función de verificar que los datos transferidos en la dirección de *broadcast* estén siendo transmitidos. Como se observa en la Figura 46 esta muestra los mensajes que se han estado enviando estos mensajes corresponden a los enviados por el primer dispositivo y los últimos son la respuesta por parte de la plataforma móvil.

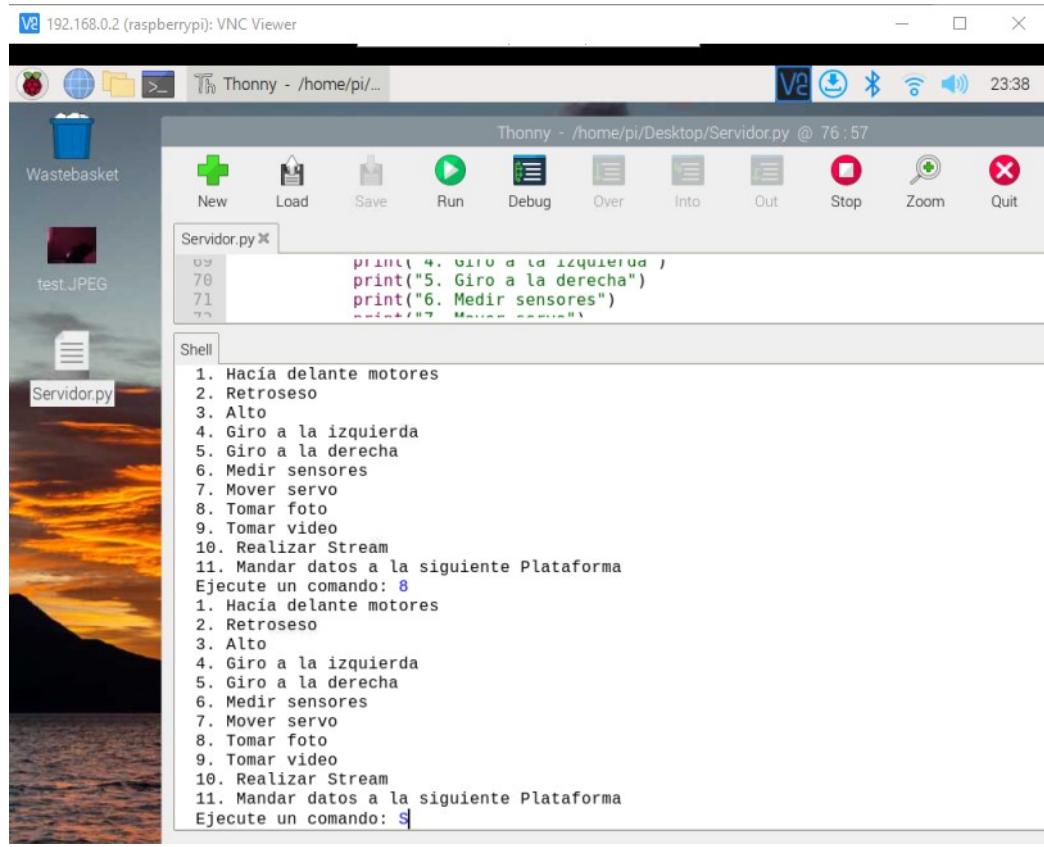


Figura 44: Interfaz del usurario desde la Raspberry Pi 3 dirección: 192.168.0.2 (dirección IP especificada únicamente para los ejemplos presentados en estas pruebas).

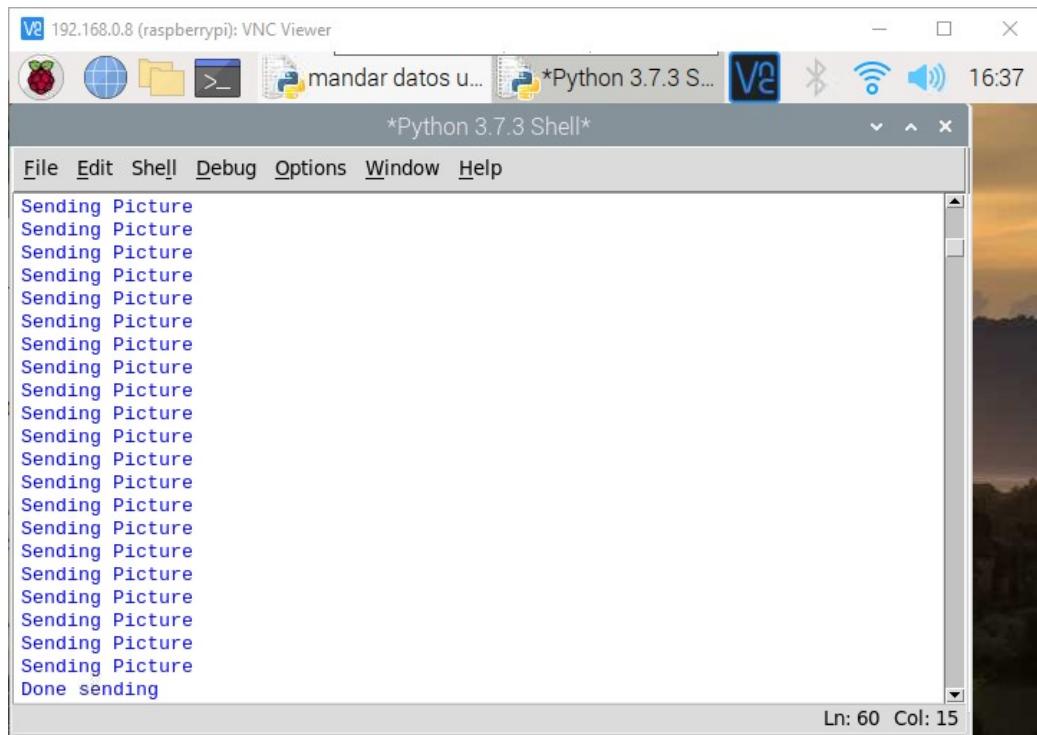


Figura 45: Pantalla de la plataforma móvil para aplicaciones de robótica de enjambre dirección: 192.168.0.8 (dirección IP especificada únicamente para los ejemplos presentados en estas pruebas).

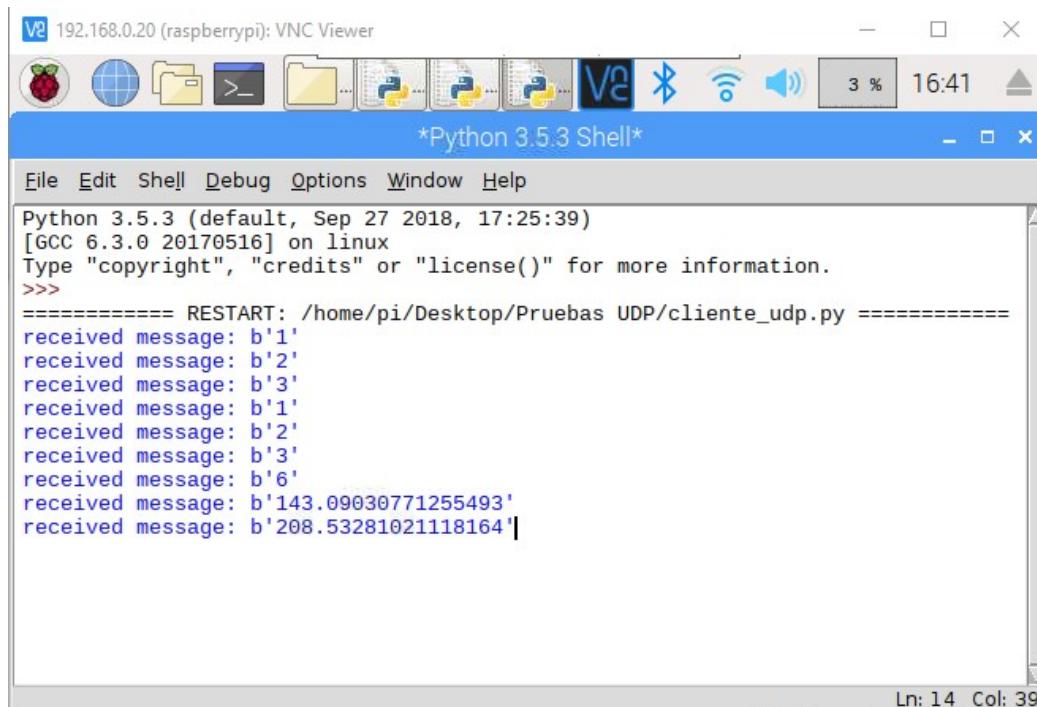


Figura 46: Pantalla del cliente Raspberry Pi 3 dirección: 192.168.0.20 (dirección IP especificada únicamente para los ejemplos presentados en estas pruebas).

## 10.2. Sistema de visión por computadora

En este sistema se demuestran las distintas opciones que ofrece la Pi Camera V2 NoiR. Aparte de la toma de fotos, se tiene la configuración de para la toma de videos por una cantidad específica de tiempo y finalmente se realiza el *stream* desde un servidor local generado por la *Raspberry Pi*.

Para la resolución se tienen las opciones de 1080 a 30 cuadros por segundo, 720 a 60 cuadros por segundo y  $640 \times 480$  a 60 o 90 cuadros por segundo [24]. Para la toma de imágenes, video y realización de la transmisión en directo se optó por utilizar la resolución de  $640 \times 480$  a 30 cuadros por segundo para realizar con mayor rapidez las mediciones en la *Raspberry Pi Zero W*. El *stream* se genera por medio de la librería *socket* con la extensión de *http*, para el funcionamiento de la dirección web, esta se une a la dirección IP que tome la *Raspberry Pi* cuando esta se conecte a una red inalámbrica, pero se debe de seleccionar un puerto disponible. Los puertos abarcan una gran cantidad de posibles números pero es preferible escoger entre el rango de 500 - 50000 este puerto ya no podrá ser utilizado más que exclusivamente para la función de la transmisión en vivo. A esta dirección se puede ingresar desde cualquier dispositivo que este conectado a la red inalámbrica.

Los archivos soportados para la toma de fotos fueron:

- Archivos JPEG
- Archivos JPG
- Archivos PNG

Los archivos soportados para la toma de videos fueron:

- Archivos h264

Para poder acceder a la transmisión en directo desde cualquier dispositivo los requerimientos son:

- Conexión a la misma red
- Introducir en el Navegador: Dirección.IP.del.Stream : Puerto.Especificado Ejemplo: 192.168.0.33 : 9876

Para la captura de cualquier tipo de imágenes se hicieron pruebas con el balance de blancos que la librería PiCamera [27] nos ofrece los cuales tiene las siguientes opciones:

- off
- auto
- sunlight
- cloudy

- shade
- tungsten
- fluorescent
- incandescent
- flash
- horizon

El modo preferido fue el de “auto” ya que como se puede observar en la Figura 47 este se adapta a las condiciones de luz que existen en el ambiente, otros modos como “off” presentado en la Figura 48 o “sunlight” presentado en la Figura 49 se adaptan a situaciones muy específicas dentro del espectro de la luz. Durante la toma de vídeo y *stream*, se observa el cambio en el balance de blancos a medida que la cámara percibe un cambio de luz en el ambiente.

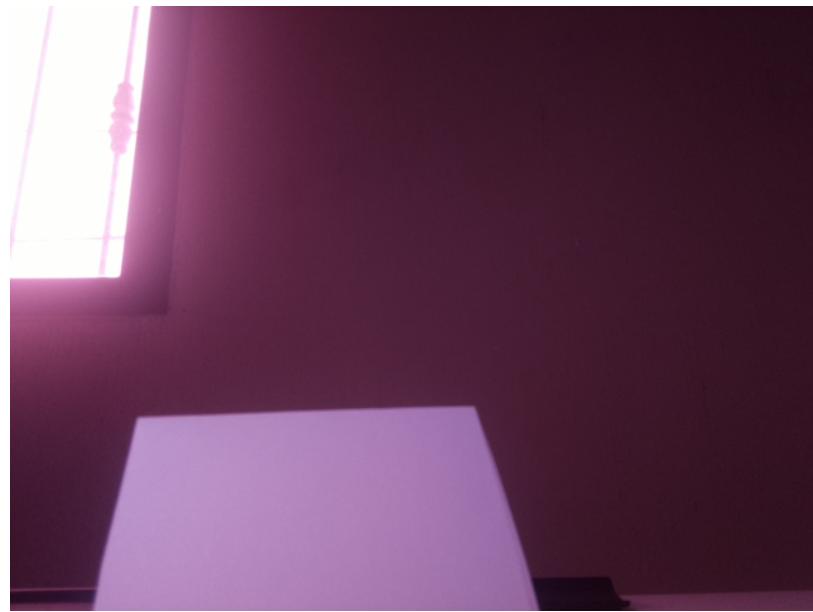


Figura 47: Fotografía desde la Raspberry Pi NoIR camera V2 con el balance de blancos: auto

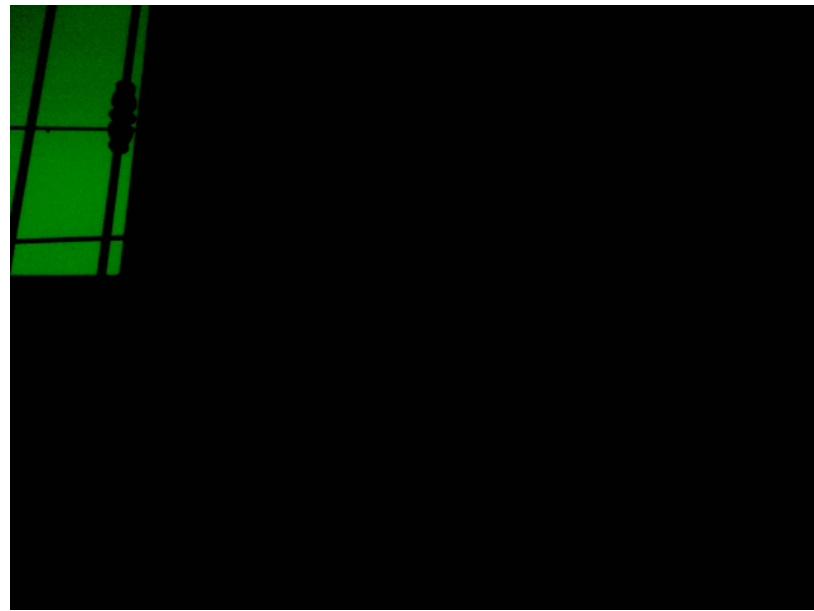


Figura 48: Fotografía desde la Raspberry Pi NoIR camera V2 con el balance de blancos: off

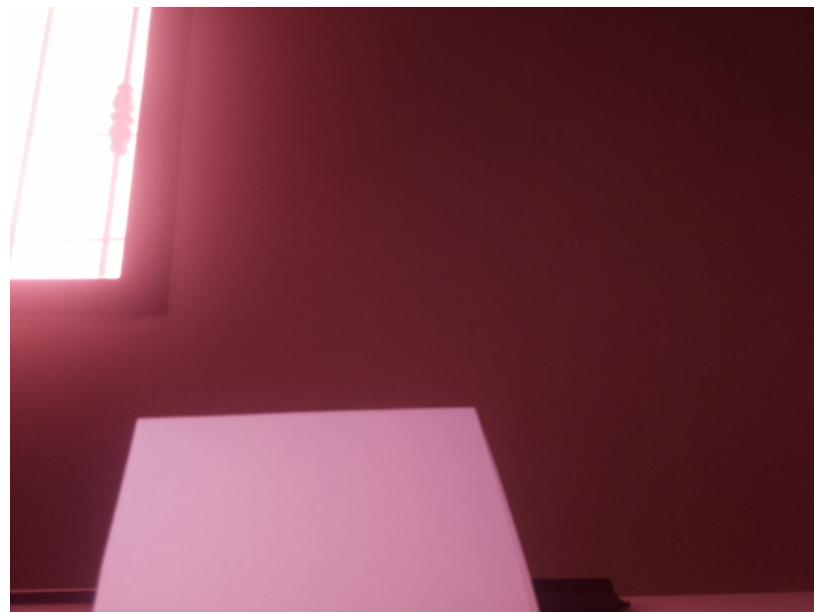


Figura 49: Fotografía desde la Raspberry Pi NoIR camera V2 con el balance de blancos: sunlight

Para la toma de vídeos se puede llegar a especificar el momento en que se desea tomar y la cantidad de tiempo en que estará grabando en la Figura 50 se realizó una prueba de únicamente 10 segundos. Por cada minuto, con una resolución de  $680 \times 420$  a 30 cuadros por segundo el espacio de almacenamiento es alrededor de 60 MB. Se realizaron pruebas de grabación de vídeo hasta para 20 min.

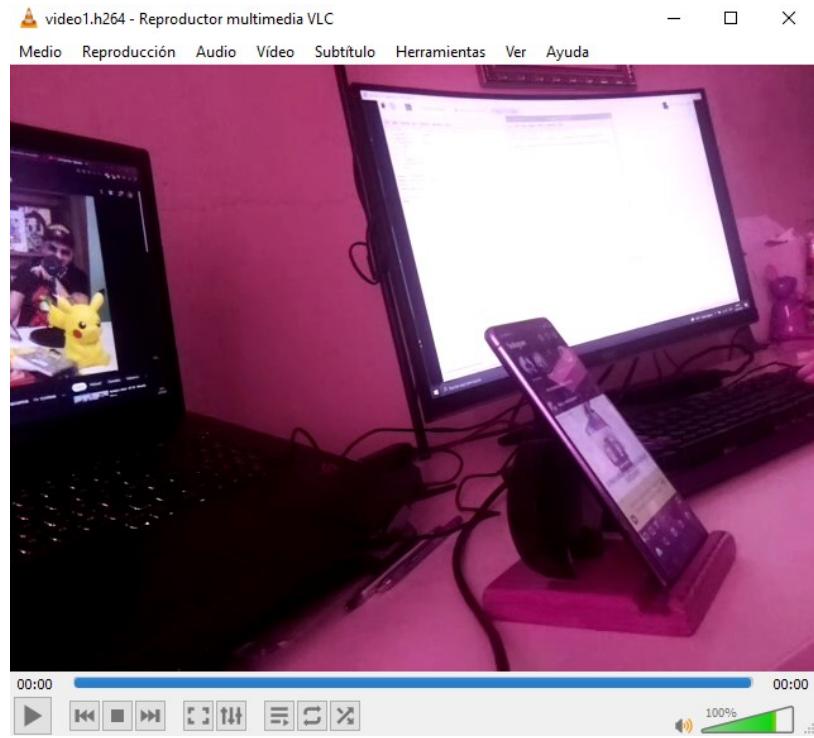


Figura 50: Vídeo desde la Raspberry Pi NoIR camera V2 con el balance de blancos: auto

En la Figura 51 y la Figura 52 se muestra la visualización de la transmisión realizada por la plataforma móvil y en la que se puede acceder en cualquier dispositivo que se encuentre conectado a la red.



Figura 51: Transmisión en vivo realizada desde la plataforma móvil y visualizada en un dispositivo Android

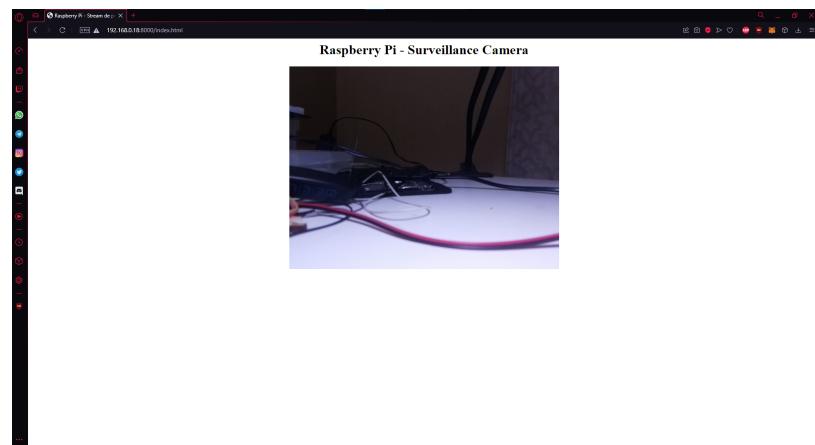


Figura 52: Transmisión en vivo realizada desde la plataforma móvil y visualizada en una computadora

### 10.3. Detección de objetos

El objetivo de esta prueba era demostrar la confiabilidad en el uso de los sensores ultrasónicos LV-MaxSonar-EZ1. Estos emiten ondas ultrasónicas a 1.75 cm del centro de la plataforma. Dentro de la plataforma de *Raspberry Pi* se hace un arreglo para poder manejar de manera eficaz los datos dados por el sensor, sin embargo, por utilizar factores como asumir la velocidad del sonido, y las mediciones en el tiempo para la parte inicial y final, estos pueden llegar a dar datos erróneos limitados por la capacidad del controlador. +

Para este sistema se realizaron pruebas con distintos números de iteraciones 5, 20 y 50 para diferentes distancias. Se puso a medir un objeto de papel totalmente blanco con dimensiones de 9.5 cm de alto por 11 cm de largo y las distancias exactas a medir a partir de la salida del sensor fueron de 20, 25 y 50cm. En el Cuadro 5 se muestran los porcentajes de error con respecto a los mediciones realizadas y los valores teóricos medidos. En las Figuras 53 - 58 muestran el comportamiento de cada prueba realizada para las distintas distancias e iteraciones realizadas, todas las gráficas muestran la regresión lineal que describe su comportamiento.

Cuadro 5: Porcentajes de error en los sensores LV-MaxSonar-EZ1

Iteraciones	Medida Exacta	Promedio del sensor	Porcentaje de error
20	20	20.65	3.29 %
20	50	49.66	0.68 %
50	25	23.43	6.26 %
50	50	51.54	3.07 %
5	20	21.51	7.54 %
5	50	39.22	21.55 %

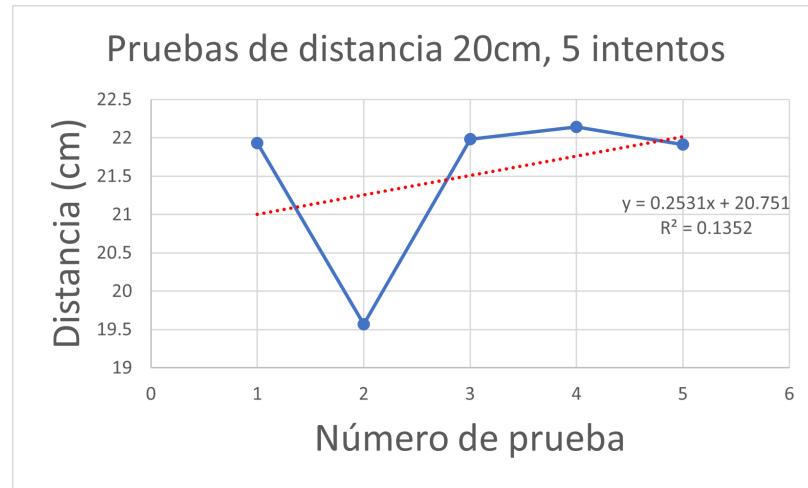


Figura 53: Resultado de la medición del sensor LV-MaxSonar-EZ1

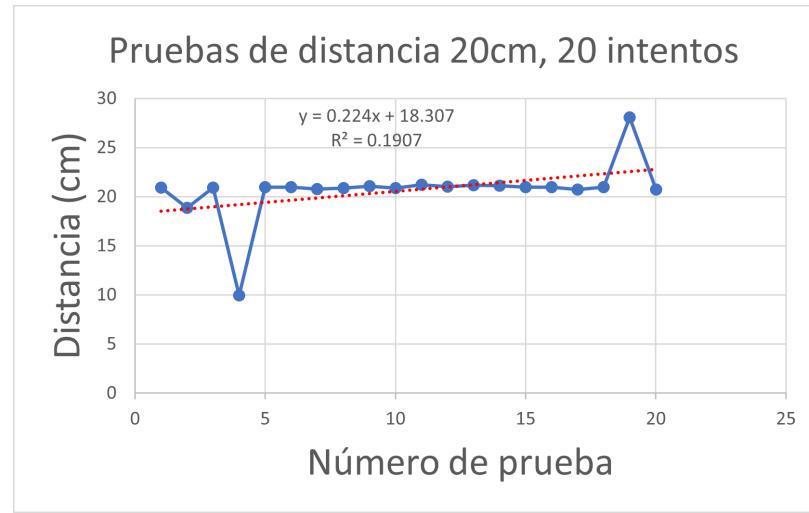


Figura 54: Resultado de la medición del sensor LV-MaxSonar-EZ1

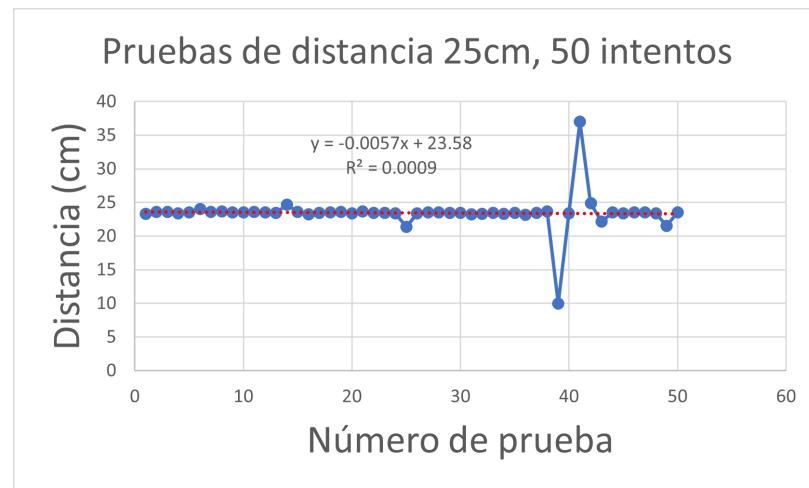


Figura 55: Resultado de la medición del sensor LV-MaxSonar-EZ1

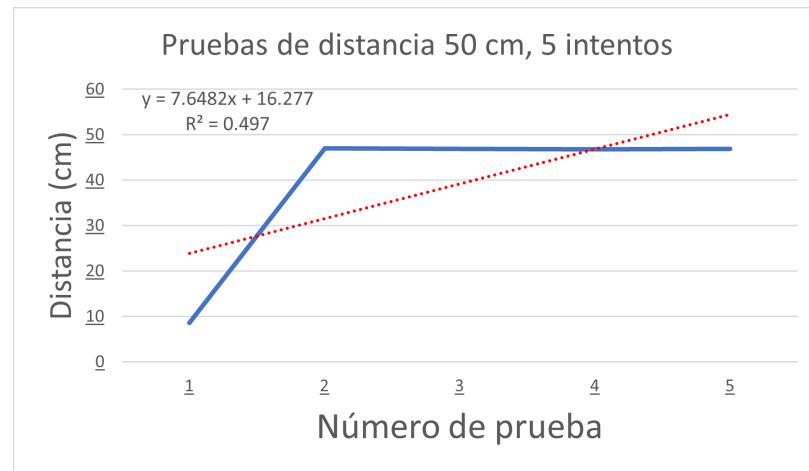


Figura 56: Resultado de la medición del sensor LV-MaxSonar-EZ1

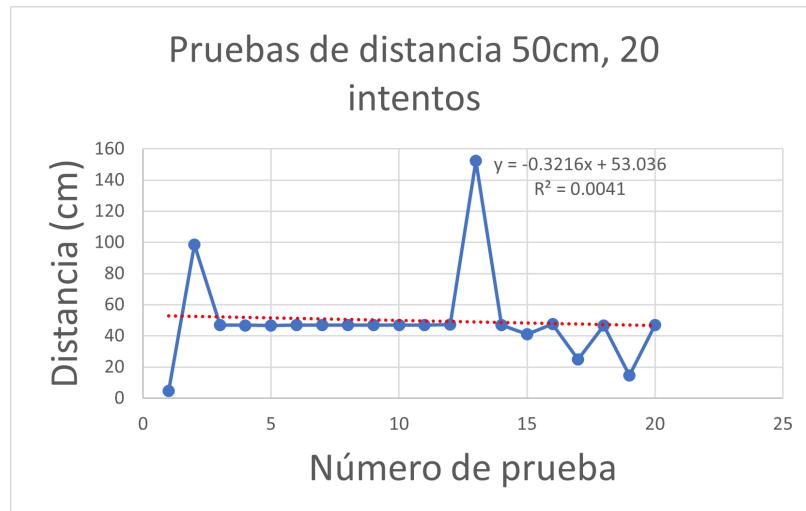


Figura 57: Resultado de la medición del sensor LV-MaxSonar-EZ1

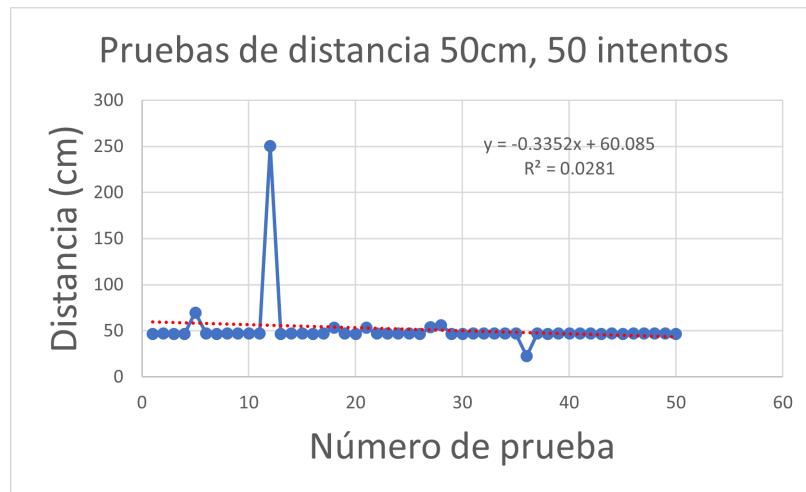


Figura 58: Resultado de la medición del sensor LV-MaxSonar-EZ1

## 10.4. Movimientos de los motores

Para la realización de este módulo se generaron distintas funciones las cuales completan todas las posibles direcciones que pueden a llegar a tomar los motores para avanzar de un punto A a un punto B.

En la Figura 59 se presenta la simulación de la ejecución de los movimientos para avance en línea recto y retroceso en línea recta.

En la Figura 60 se presenta la simulación de la ejecución de los movimientos para el giro hacia la izquierda y giro hacia la derecha.

En la Figura 61 se presenta la simulación de la ejecución de los movimientos para un avance en diagonal hacia adelante para la izquierda y la derecha.

En la Figura 61 se presenta la simulación de la ejecución de los movimientos para un avance en diagonal hacia atrás para la izquierda y la derecha.

La *Raspberry Pi Zero W* manda una señal de 1000 Hz a dos pines GPIO para que estos controlen el ciclo de trabajo de los motores. Los motores se mueven de manera funcional con la batería 18650 3.7 V 9800 mAh, pero esta fuente de poder no cumple con las condiciones necesarias para obtener los valores máximos de revoluciones por minuto que especifica el fabricante (ser ejecutados a 12 V para alcanzar las 600 RPM). Se realizan las mediciones para poder determinar los valores máximos alcanzados para la velocidad con respecto al ciclo de trabajo deseado, en el Cuadro 6 se especifican las RPM alcanzadas y el porcentaje de ciclo de trabajo para poder obtenerlos.

Ciclo de trabajo	RPM
100 %	185
90 %	167
80 %	148
70 %	130
60 %	111
50 %	93
40 %	74
30 %	56
20 %	37
10 %	19

Cuadro 6: Porcentaje de funcionamiento de los motores y RPM alcanzadas

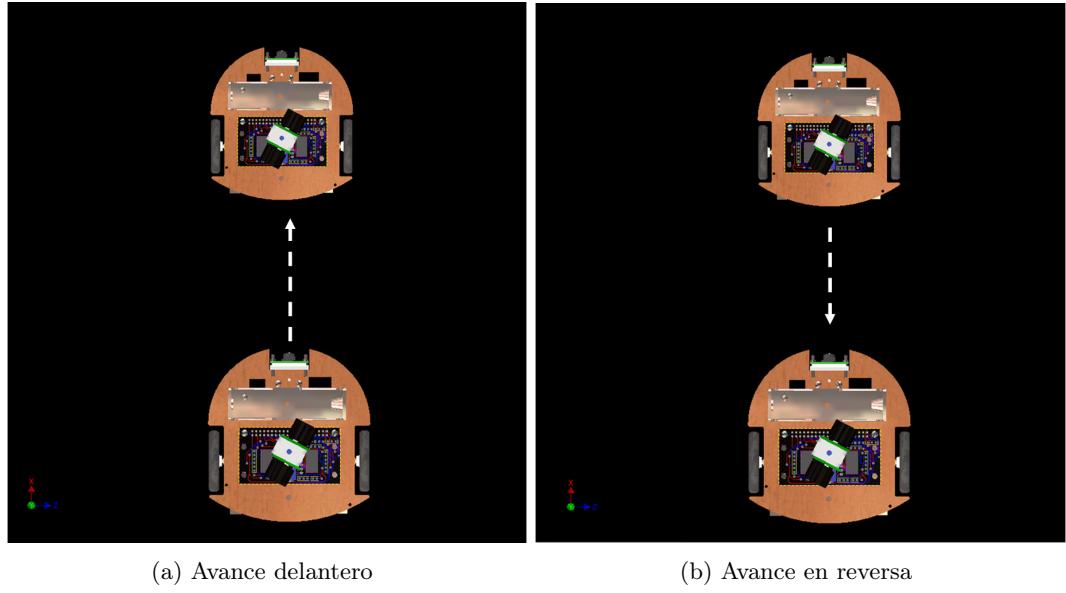


Figura 59: Movimiento hacia adelante y en reversa programado para la plataforma

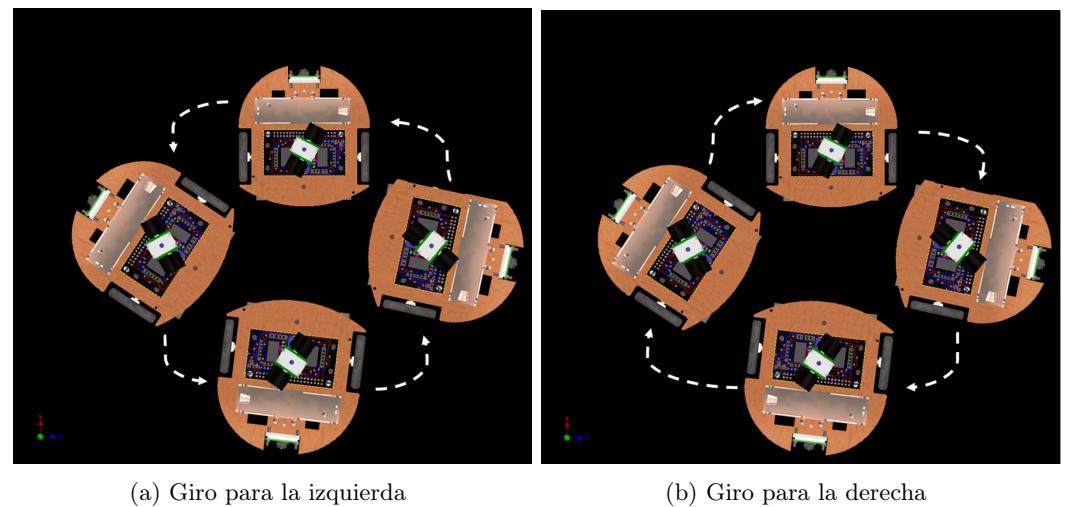
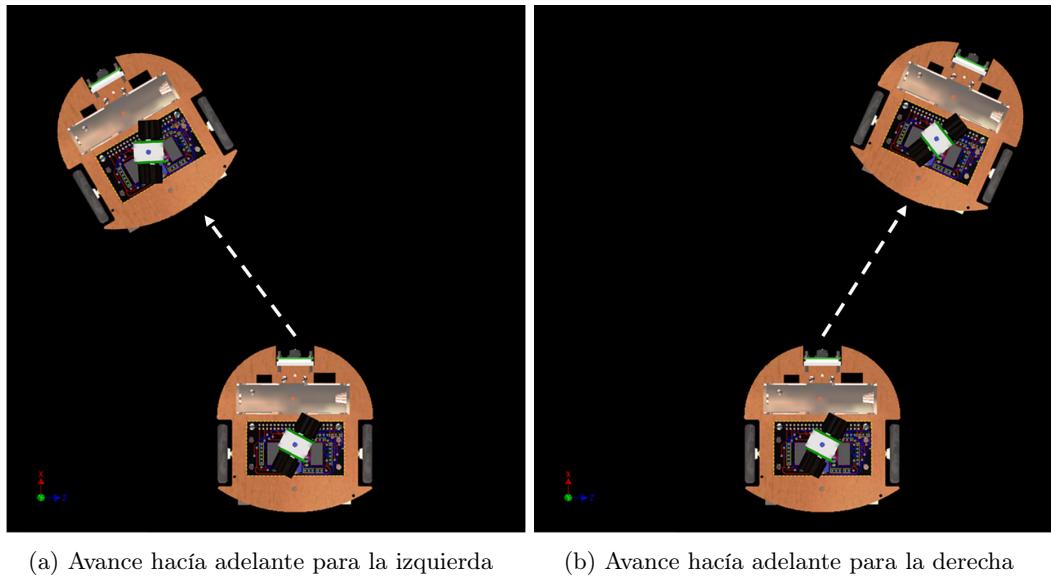


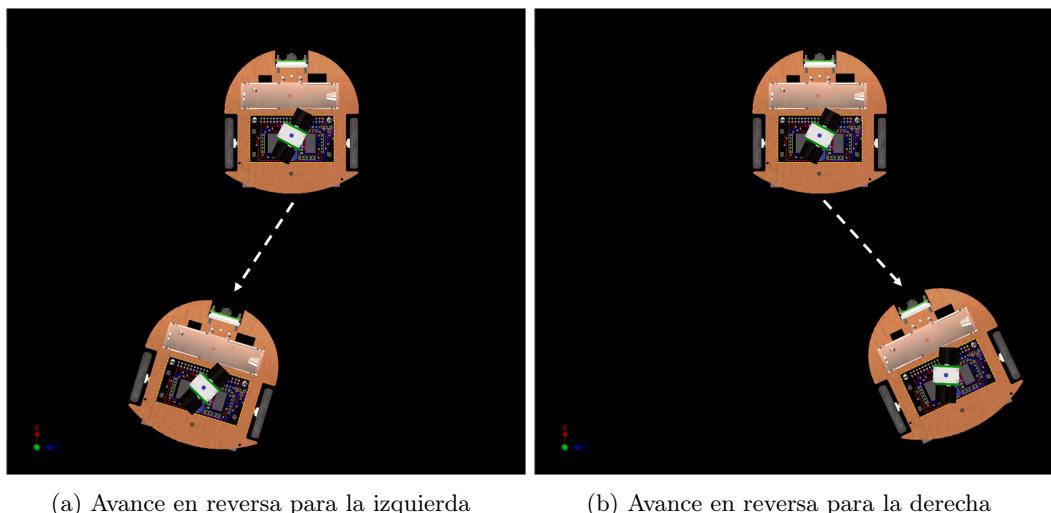
Figura 60: Movimiento para el giro hacia la izquierda y giro hacia la derecha programado para la plataforma



(a) Avance hacia adelante para la izquierda

(b) Avance hacia adelante para la derecha

Figura 61: Movimiento hacia adelante para la izquierda y hacia adelante para la derecha programado para la plataforma



(a) Avance en reversa para la izquierda

(b) Avance en reversa para la derecha

Figura 62: Movimiento en reversa para la izquierda y reversa para la derecha programado para la plataforma

# CAPÍTULO 11

---

## Conclusiones

---

- El implementar una cámara y un microcontrolador con mayor compatibilidad con algoritmos de robótica de enjambre aumentó en un 26 % el costo de la plataforma en comparación a la fase II.
- La plataforma *Raspberry Pi* permitió una adaptación exitosa de los módulos de movimiento, detección de movimiento y de transmisión de datos.
- El sistema de visión por computadora permite realizar fotografías, captura de vídeo y realizar transmisiones en directo desde una red local.
- El prototipo final tuvo un costo de \$152.41 con la mayoría de productos encontrados localmente.
- La plataforma tiene un consumo de corriente máximo de 600 mA y su tiempo de funcionamiento con una batería 18650 3.7 V 9800 mAh es de aproximadamente 5 horas.
- La plataforma puede comunicarse con otros dispositivos por medio del protocolo TCP/IP. Es posible usar tanto el modo basado en conexión (TCP) como el modo sin conexión (UDP).

# CAPÍTULO 12

---

## Recomendaciones

---

- Los sensores ultrasónicos son los de mayor valor en la plataforma, por lo que se debe analizar si la detección de objetos debe de ser con obstáculos tan alejados, o si se pueden implementar sensores de corto alcance como los sensores infrarrojos para programación de detección de obstáculos.
- La funcionalidad de la transmisión en directo únicamente funciona para red local. El implementar una nueva funcionalidad para transmisión de datos desde diferentes redes ayudaría para poder trabajar con la plataforma desde distintos lugares y que esta permanezca en un espacio designado permanentemente.
- Se recomienda validar la plataforma con algoritmos de robótica de enjambre que se han desarrollado a nivel simulación y que se han probado anteriormente con plataformas no móviles.

## CAPÍTULO 13

---

### Bibliografía

---

- [1] P. Gonçalves, P. Torres, C. Alves, F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptoč, S. Magnenat, J.-C. Zufferey, D. Floreano y A. Martinoli, “The e-puck, a Robot Designed for Education in Engineering,” *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, vol. 1, ene. de 2009.
- [2] O. Michel, “WebotsTM: Professional Mobile Robot Simulation,” *International Journal of Advanced Robotic Systems*, vol. 1, mar. de 2004. DOI: 10.5772/5618.
- [3] A. Millard, R. Joyce, J. Hilder, C. Fleseriu, L. Newbrook, W. Li, L. McDaid y D. Halliday, “The Pi-puck extension board: a Raspberry Pi interface for the e-puck robot platform,” jun. de 2017. DOI: 10.1109/IROS.2017.8202233.
- [4] M. Rubenstein, C. Ahler, N. Hoff, A. Cabrera y R. Nagpal, “Kilobot: A low cost robot with scalable operations designed for collective behaviors,” *Robotics and Autonomous Systems*, vol. 62, págs. 966-975, jul. de 2014. DOI: 10.1016/j.robot.2013.08.006.
- [5] E. I. J. O. D. C. G. J. J. M. R. O. G. W. C. V. R. V. Z. W. A. O. C. y. R. A. A. B. Hernández Woc, “Reingeniería de Megaproyectos Fase 1,” Tesis de licenciatura, Universidad Del Valle de Guatemala, 2017.
- [6] C. J. Lima, “Diseño e implementación de una nueva plataforma móvil para aplicaciones en robótica de enjambre,” Tesis de licenciatura, Universidad Del Valle de Guatemala, 2018.
- [7] P. Kalla, R. K. Ramj y P. Ravindranath, “Swarm home robots,” en *2017 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, 2017, págs. 139-144. DOI: 10.1109/ICCE-ASIA.2017.8307836.
- [8] J. Kennedy y R. Eberhart, “Particle swarm optimization,” en *Proceedings of ICNN’95 - International Conference on Neural Networks*, vol. 4, 1995, 1942-1948 vol.4. DOI: 10.1109/ICNN.1995.488968.
- [9] Real Academia de Ingeniería de España, *Comunicación inalámbrica*, <https://diccionario.raing.es/es/lema/comunicación-inalámbrica>, Visitado: 3-04-2022.
- [10] “Introducción de Aplicaciones UDP en Redes Privadas Virtuales,” en *III Jornadas de Ingeniería Telemática*, Barcelona (Spain), sep. de 2001, págs. 397-404.

- [11] Definition from Techopedia, *What is a Proximity Sensor?* <https://www.techopedia.com/definition/15003/proximity-sensor>, Visitado: 30-03-2021.
- [12] A. Al-Mahturi y R. Rahim, “ULTRASONIC SENSOR FOR DISTANCE MEASUREMENT,” en. mar. de 2016, págs. 9-14, ISBN: 978-967-448-052-3.
- [13] MaxBotix Inc., *MB1010 LV-MaxSonar-EZ1*, <https://hardzone.es/reviews/perifericos/analisis-raspberry-pi-3-modelo-b/>, Visitado: 03-04-2022.
- [14] A. Nayyar y V. Puri, “Raspberry Pi-A Small, Powerful, Cost Effective and Efficient Form Factor Computer: A Review,” *International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE)*, vol. 5, págs. 720-737, dic. de 2015.
- [15] ——, “Raspberry Pi-A Small, Powerful, Cost Effective and Efficient Form Factor Computer: A Review,” *International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE)*, vol. 5, págs. 720-737, dic. de 2015.
- [16] R. Berra, F. Setti y M. Cristani, “Berrick: a low-cost robotic head platform for human-robot interaction,” oct. de 2019, págs. 559-566. DOI: 10.1109/SMC.2019.8913932.
- [17] Rubpen Velasco, *Análisis: Raspberry Pi 3 Modelo B+*, <https://hardzone.es/reviews/perifericos/analisis-raspberry-pi-3-modelo-b/>, Visitado: 10-09-2021.
- [18] H. Hogan, “Servo motors,” vol. 52, mayo de 2005. dirección: [https://www.researchgate.net/publication/294776126\\_Servo\\_motors](https://www.researchgate.net/publication/294776126_Servo_motors).
- [19] ISL Products International Ltd., *DC Motor and DC Gear motor*, <https://islproducts.com/design-note/dc-motor-dc-gear-motor-basics/>, Visitado: 10-08-2021.
- [20] Universidad Simón Bolívar., *Driver de motor DC – Electrónica de potencia*, <https://dademuch.com/2018/04/26/driver-de-motor-dc-electronica-de-potencia/>, Visitado: 19-03-2022.
- [21] LUIS LLAMAS., *CONTROLAR DOS MOTORES DC CON ARDUINO Y DRIVER TB6612FNG*, <https://www.luisllamas.es/arduino-motor-dc-tb6612fng/>, Visitado: 19-03-2022.
- [22] Securame., *Baterias de Li-Ion: Ventajas, desventajas, y mantenimiento*, <https://www.securame.com/blog/baterias-de-li-ion-ventajas-desventajas-y-mantenimiento/>, Visitado: 19-03-2022.
- [23] Pololu., *Pololu Ball Casters*, <https://www.pololu.com/category/45/pololu-ball-casters>, Visitado: 3-04-2022.
- [24] Raspberry Pi Foundation., *Raspberry Pi Camera Module 2 NoIR*, <https://www.raspberrypi.com/products/pi-noir-camera-v2/>, Visitado: 3-04-2022.
- [25] M. Á. B. Saucedo, “MEDICIÓN DE LA CAPACIDAD PRÁCTICA DE BATERÍAS RECARGABLES DE LI-ION TIPO 18650,” vol. 1, pág. 18, sep. de 2019.
- [26] Max.K, *ZeroBot - Raspberry Pi Zero FPV Robot*, <https://hackaday.io/project/25092-zerobot-raspberry-pi-zero-fpv-robot>, Visitado: 20-08-2021.
- [27] Dave Jones Revision, *Source code for picamera.camera*, [https://picamera.readthedocs.io/en/release-1.13/\\_modules/picamera/camera.html](https://picamera.readthedocs.io/en/release-1.13/_modules/picamera/camera.html), Visitado: 4-04-2022.

