

# **“FITOBOT” ROBOT MÓVIL AUTÓNOMO PARA MONITOREO FITOSANITARIO EN INVERNADERO FORESTAL**

*“FITOBOT” AUTONOMOUS MOBILE ROBOT FOR PHYTOSANITARY  
MONITORING IN FOREST GREENHOUSES*

**Ángel Vergara Betancourt**

Tecnológico Nacional de México / ITS de Zacapoaxtla, México  
*angel.vb@zacapoaxtla.tecnm.mx*

**Valentín Contreras Hernández**

Tecnológico Nacional de México / ITS de Zacapoaxtla, México  
*valentincontrerash14@gmail.com*

**José Ángel Reyes Guzmán**

Tecnológico Nacional de México / ITS de Zacapoaxtla, México  
*ing.angel.reyes00@gmail.com*

**Recepción:** 12/noviembre/2024

**Aceptación:** 14/marzo/2025

## **Resumen**

El uso de dispositivos robóticos para realizar estudios fitosanitarios en ambientes agroforestales confinados mejora la precisión, eficiencia y rapidez en la detección de enfermedades, plagas y estrés que afectan el crecimiento y desarrollo de las plantas. El proyecto FitoBot se centra en el diseño e implementación de un sistema robótico móvil orientado al análisis fitosanitario en espacios confinados. Este proyecto se describe en 3 etapas: diseño y construcción de una plataforma robótica móvil, integración de sistema de adquisición de datos ambientales y percepción visual y programación de algoritmos de navegación autónoma. El hardware utilizado incluye Raspberry Pi, sensores ambientales, cámara, baterías y comunicación remota. Como lenguaje de programación se ha empleado Python y la librería de visión artificial OpenCV. Como resultado se logró la construcción y programación de un robot móvil con capacidad de navegación autónoma, detección de variables climáticas y visualización del entorno.

**Palabras Clave:** Fitosanidad, Monitoreo ambiental, Navegación autónoma, Robot móvil, Visión artificial.

## **Abstract**

*The use of robotic devices to carry out phytosanitary studies in confined agroforestry environments improves the accuracy, efficiency and speed in the detection of diseases, pests and stresses that affect plant growth and development. The FitoBot project focuses on the design and implementation of a mobile robotic system aimed at phytosanitary analysis in confined spaces. This project is described in 3 stages: design and construction of a mobile robotic platform, integration of environmental data acquisition and visual perception system and programming of autonomous navigation algorithms. The hardware used includes Raspberry Pi, environmental sensors, camera, batteries, and remote communication. Python and the OpenCV artificial vision library have been used as programming languages. As a result, the construction and programming of a mobile robot with autonomous navigation capacity, detection of climatic variables and visualization of the environment was achieved.*

**Keywords:** *Autonomous navigation, Artificial vision, Environmental monitoring, Mobile robot, Phytosanitary.*

## **1. Introducción**

Ante el crecimiento exponencial de la población y en consecuencia la alta demanda de alimentos, se plantean desafíos significativos en el sector agropecuario. Uno de ellos, asegurar el abastecimiento de productos agrícolas que satisfagan las necesidades básicas del ser humano. Ante este escenario y como parte de una nueva política agrícola del país, el gobierno de México ha propuesto una agenda estratégica para la autosuficiencia alimentaria y rescate del campo mexicano, y que forma parte de las agendas del Tecnológico Nacional de México [TecNM, 2024]. Esta estrategia tiene como intención aumentar de manera sostenible y sustentable la producción de insumos, y así asegurar el abastecimiento de alimentos para satisfacer las necesidades de las poblaciones locales. Sin embargo, una de las problemáticas que enfrentan los pequeños y medianos productores para lograr una autosuficiencia alimentaria, son las condiciones fitosanitarias del sector primario y la falta de transformación y agregación de valor a

la producción. Es decir, la gestión integral de plagas, enfermedades y condiciones de estrés en los cultivos. No obstante, los métodos tradicionales de monitoreo fitosanitario a menudo son laboriosos y propensos a la falta de precisión, lo que puede resultar en pérdidas significativas de cosechas y la necesidad de la aplicación excesiva de productos químicos. Ante ello, la agricultura moderna se enfrenta a la necesidad de adoptar tecnologías avanzadas que permitan mejorar la eficiencia, la productividad y garantizar la calidad de los cultivos.

En la actualidad, por ejemplo, la tecnología se ha hecho presente en el sector agrícola, con contribuciones como la agricultura de precisión, la agricultura 4.0, la automatización y la robótica agrícola [Santos-Valle, 2021]. En esa dirección, se han reportado trabajos que se enfocan en la protección fitosanitaria de los cultivos a través del consumo óptimo de insumos (agua, fertilizantes y fitosanitarios) y la disminución de los costes operativos [Planas, 2018]. También se han reportado trabajos que discuten la propuesta de uso de plataformas robóticas móviles para realizar tareas repetitivas, peligrosas y tediosas en el interior de invernaderos [Rodríguez, 2016] y el uso de algoritmos de navegación autónoma para la planificación de trayectorias [Álvarez, 2020].

De igual manera, con el auge de tecnologías emergentes como la industria 4.0, la inteligencia artificial (IA) y el internet de las cosas (IoT), la visión artificial se ha consolidado y ha resultado fundamental en el sector agrícola [Kakani, 2020], [Sharma, 2021], ofreciendo soluciones destinadas a mejorar la productividad y la calidad del sector agropecuario hacia un futuro medio ambiental sostenible con un enfoque agroecológico [Quy, 2022]. Similarmente se han presentado técnicas basadas en visión artificial y aprendizaje automático para clasificar frutas, predecir enfermedades y medir la calidad del suelo, con la finalidad de combatir la escasez de alimentos y aumentar la calidad de estos [Sood, 2021].

Finalmente, también se tienen trabajos donde se utiliza el algoritmo de detección de objetos en tiempo real YOLOv5 para detectar e identificar plagas de insectos en la agricultura [Ahmad, 2022]. Otro ejemplo, es el uso de YOLOv7 en procesos de detección, por ejemplo, en [Wu, 2022] se explora el uso de este algoritmo en combinación con el aumento de datos para detectar frutos de *Camellia oleifera* en

escenas complejas, mientras que en [Zhou, 2022] se propone un método de fusión de aprendizaje profundo basado en la percepción visual y el procesamiento de imágenes para localizar de forma adaptativa y activa los puntos de reconocimiento y recolección de frutos de *Camellia oleifera*.

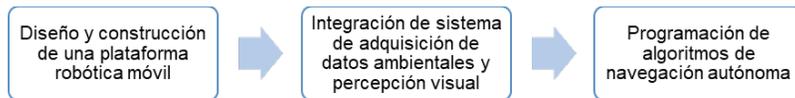
Por lo tanto, el enfoque propuesto en este proyecto pretende abordar esta problemática mediante la implementación de un sistema autónomo que combina visión artificial y robótica móvil. Para ello se ha desarrollado un sistema robótico móvil denominado "FitoBot", el cual tiene como propósito mejorar la productividad agrícola, reducir los impactos ambientales y promover prácticas agrícolas sostenibles y eficientes, contribuyendo significativamente a cumplir con algunos Objetivos de Desarrollo Sostenible (ODS) establecidos por las Naciones Unidas en la Agenda 2030, por ejemplo ODS 2: Hambre Cero, ODS 9: Industria, Innovación e Infraestructura, ODS 12: Producción y Consumo Responsables, ODS 13: Acción por el Clima y ODS 15: Vida de Ecosistemas Terrestres [United Nations Department of Economic and Social Affairs, 2023].

El objetivo de este proyecto consiste en la construcción y programación de un robot móvil con capacidad de navegación autónoma, la implementación de un sistema de monitoreo remoto de variables ambientales y la implementación de un sistema de visión artificial. Se pretende que, con el desarrollo del sistema autónomo de monitoreo fitosanitario "FitoBot", se facilite la ejecución de tareas de detección temprana de enfermedades, plagas y estrés en las plantas en entornos agroforestales confinados, lo que facilitará respuestas rápidas y específicas para mantener la salud de los cultivos y así contribuir con la estrategia de autosuficiencia alimentaria en el país.

## **2. Métodos**

El desarrollo del robot móvil se estructuró en 3 etapas: diseño y construcción de una plataforma robótica móvil, integración de sistema de adquisición de datos ambientales y percepción visual y programación de algoritmos de navegación autónoma, Figura 1. Para iniciar con el diseño del robot, bajo la condición de que debe ser capaz de navegar en entornos confinados, se propone desarrollar un

sistema robótico dotado de sensores, que sea capaz de desplazarse de forma autónoma dentro de un invernadero. Para el caso de estudio particular, el entorno experimental donde se desea operar el robot se describe en la Figura 2.



*Fuente: elaboración propia*

Figura 1 Desarrollo metodológico del robot móvil FitoBot.



*Fuente: elaboración propia*

Figura 2 Entorno experimental para estudios fitosanitarios.

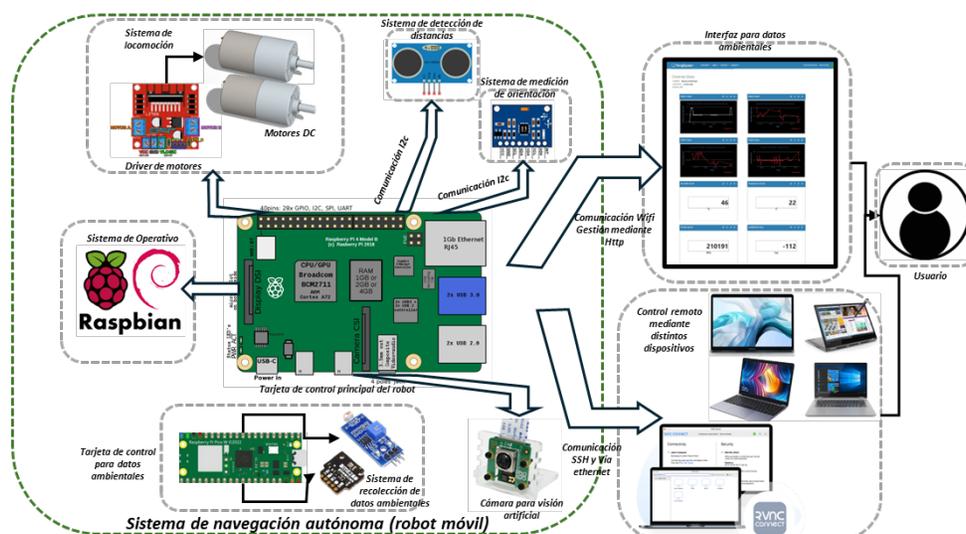
Dicho espacio, consiste en una superficie el terreno que presenta cierta irregularidad y un área de aproximadamente  $4 \times 6 \text{ m}$ . La ruta que el robot debe transitar se describe por una trayectoria en forma de “U”, y las plantas que se desean analizar, se encuentran a una altura de aproximadamente  $30 \text{ cm}$ . El interior del invernadero alcanza una temperatura superior a los  $50 \text{ }^\circ\text{C}$ , y está acondicionado para modificar las condiciones de humedad y temperatura, así como adaptarse a cultivos como el jitomate o el café. De acuerdo con el entorno experimental, se propuso que el diseño del robot fuera de tamaño reducido y con la capacidad de percibir su entorno.

A partir de las condiciones del espacio de navegación del robot descritas previamente, se requirieron los componentes siguientes:

- Hardware:
  - ✓ Kit de chasis del tanque de robot de orejas inteligentes de metal TP 101.
  - ✓ Microcomputadora de bajo costo Raspberry Pi 4.
  - ✓ Placa de desarrollo de bajo consumo Raspberry Pico W.

- ✓ Sensor ultrasónico para medición de distancia HC-SR04.
- ✓ Unidad de medición inercial (IMU) MPU6050, de 6 grados de libertad (DoF) (acelerómetro de 3 ejes y giroscopio de 3 ejes).
- ✓ Módulo para el control de motores Puente H L298N.
- ✓ Módulo de medición de variables ambientales BME280, que incluye sensores de presión, humedad y temperatura.
- ✓ Módulo de Cámara para Raspberry Pi Cam 2 de 8 MP.
- Software:
  - ✓ Sistema operativo Raspbian.
  - ✓ Python 3.x.
  - ✓ MicroPython.
  - ✓ Librería Open CV.
  - ✓ Servidor IoT ThinkSpeak mediante comunicación HTTP.
  - ✓ VNC Viewer.

Los componentes se han integrado en la plataforma robótica acorde con el esquema de la Figura 3, incluye tres subsistemas: el subsistema de control y navegación del robot, un subsistema de percepción del entorno a través de la visión de una cámara, y un subsistema de monitoreo de variables ambientales.



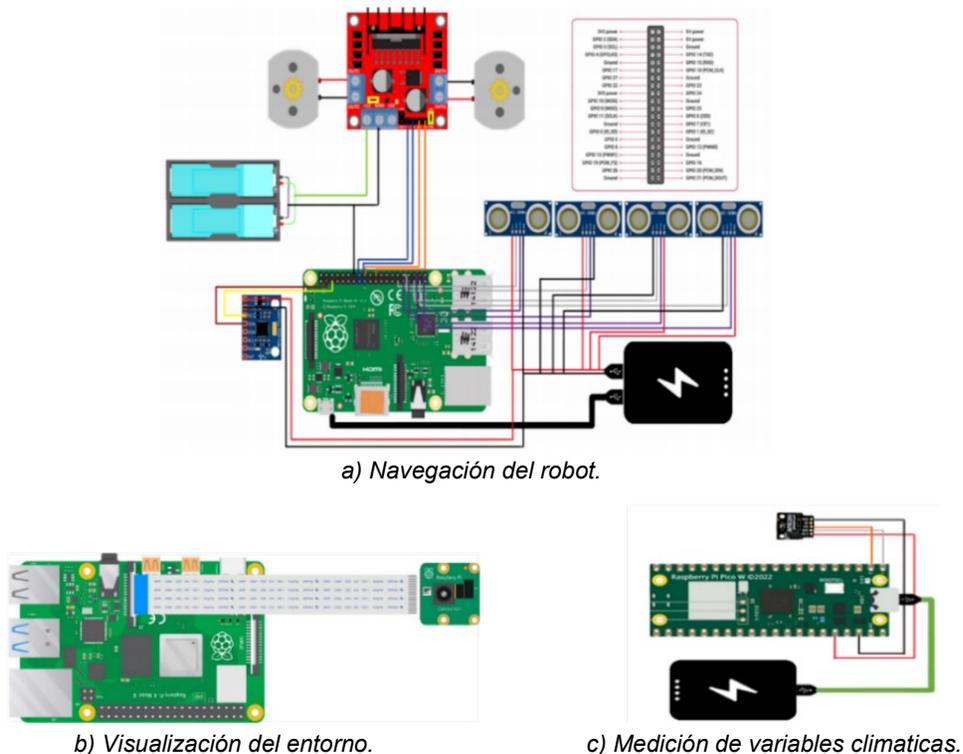
Fuente: elaboración propia

Figura 3 Arquitectura del sistema robótico FitoBot.

El control y el monitoreo remoto de este prototipo requirió de protocolos de comunicación I2C, protocolo HTTP mediante WiFi y comunicación a dispositivos móviles y computadora mediante VNC Viewer a través de ethernet y comunicación SSH.

### Diseño y construcción de una plataforma robótica móvil

Considerando que se desea desarrollar un robot con capacidad de navegación autónoma, resulta indispensable que este sea capaz de percibir su entorno, por lo cual, se ha propuesto el uso de 4 de sensores ultrasónicos HC-SR04 colocados al frente, izquierda, derecha y atrás del robot. Esto le permitió al robot tener una visión sobre los 360° con un rango de alcance de hasta 2 m. Así mismo, se integró un sensor inercial MPU6050, que le proporciona información al robot, sobre su propio estado inercial, en particular su orientación dentro del terreno. Estos sensores, al igual que un puente H L298N se conectaron a la tarjeta Raspberry Pi 4 a través de puertos GPIO como se muestra en el diagrama de la Figura 4a.



Fuente: elaboración propia

Figura 4 Esquema de conexiones del sistema FitoBot.

## Sistema de adquisición de datos ambientales y percepción visual

Para el caso del módulo de la cámara, se decidió una segunda tarjeta Raspberry Pi 4, con la intención de no saturar la memoria y capacidad de procesamiento de la tarjeta principal, conectando dicha cámara al puerto CSI de esta Raspberry Pi, como se observa en la Figura 4b. Cabe mencionar que las imágenes colectadas con este dispositivo se almacenaron de manera local para su posterior análisis en una computadora de mayor rendimiento computacional.

Para la integración de sensores ambientales se utilizó una tarjeta de desarrollo Raspberry Pico W y el módulo BME280. El procesamiento de estos se realizó utilizando principios de IoT, enviando la información recolectada a la nube utilizando el servidor ThinkSpeak mediante protocolo de comunicación HTTP, como se observa en la Figura 4c.

La percepción visual se basó en algoritmos de procesamiento digital de señales, los cuales utilizaron principios de segmentación de imágenes para el reconocimiento de color. Para ellos se optó por analizar las imágenes en el espacio de color HSV (Hue, Saturation, Value) y se utilizó la técnica de umbralización [Solomon, 2011].

La umbralización consiste en separar los píxeles de una imagen en diferentes clases o regiones según su color, tomando como referencia un valor de umbral, para las tres componentes del espacio de color,  $H$ : Matiz (color),  $S$ : Saturación (intensidad del color) y  $V$ : Valor (luminosidad o brillo). Con estos parámetros, la umbralización se define por Ecuación 1.

$$\begin{aligned} H_{min} \leq H(x, y) \leq H_{max} \\ \text{Si } S_{min} \leq S(x, y) \leq S_{max} \Rightarrow \text{pixel es parte del objeto} \quad (1) \\ V_{min} \leq V(x, y) \leq V_{max} \end{aligned}$$

Donde,  $H(x, y)$ ,  $S(x, y)$  y  $V(x, y)$  son los valores de los componentes HSV para el píxel en las coordenadas  $(x, y)$ , y  $H_{min}$ ,  $H_{max}$ ,  $S_{min}$ ,  $S_{max}$ ,  $V_{min}$  y  $V_{max}$  son los umbrales establecidos para los componentes de color.

## Programación de algoritmos de navegación autónoma

La navegación autónoma de un robot consiste en la capacidad del sistema robótico para moverse de forma independiente en distintos entornos, sin intervención

humana, considerando además tareas como la planificación, la percepción y el control de este. Para lograr este aspecto, se han desarrollado y reportado diversas técnicas, estrategias y algoritmos de navegación autónoma, como por ejemplo las descritas en [Alamri, 2021]. Entre los algoritmos para navegación de robots en entornos confinados, se tienen aquellos que conocen a priori el entorno de navegación y los que desconocen su entorno, Tabla 1.

Tabla 1 Algoritmos de navegación autónoma para robots móviles.

Conocen el entorno	Desconocen el entorno
• Algoritmo de Lee	• Algoritmo Wall-Follower
• Algoritmo de Dead End	• Algoritmos Tremaux
• Algoritmos Soukup	• Algoritmo Random mouse
• Algoritmo Maze Router	• Algoritmo Pledge
• Algoritmos Hadlock	
• Algoritmo Flood Fill	

Fuente: elaboración propia

El primer grupo se caracteriza por tener conocimiento de su entorno, esto se logra por medio de métodos de mapeo, además del registro de posiciones y trayectorias. Mientras que el segundo grupo de algoritmos, desconocen el área a navegar, por lo tanto, el desplazamiento que vaya a tener el robot móvil será regido por la lógica del algoritmo a emplear.

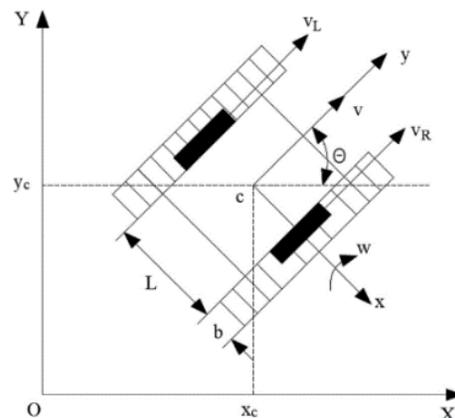
De los algoritmos descritos en la Tabla 1, se optó por analizar 4 de ellos: Random-Mouse, Wall-Follower, Pledge y Flood-Fill. La lógica de operación de estos algoritmos es:

- *Random mouse*. Con este algoritmo el robot puede recorrer el entorno para encontrar un punto final designado, el robot avanza en línea recta hasta encontrar un obstáculo en su camino, en ese caso, el robot tomará un camino distinto al girar a la izquierda o derecha y continua con su recorrido. Este proceso se realiza hasta que el robot encuentre el punto de meta.
- *Wall-Follower*. Este algoritmo conocido en español como seguidor de paredes, utiliza sensores de presencia y es uno de los algoritmos más utilizados para solución de laberintos, por su simplicidad y bajo costo de

implementación. La lógica de este algoritmo se basa en el seguimiento de paredes o del área a evaluar. Si se encuentra con un obstáculo el robot lo evitará siguiendo la regla de la mano derecha o la mano izquierda.

- *Pledge*. Este algoritmo también conocido como algoritmo de promesa, consiste en una versión optimizada de wall-follower. Al momento de pasar por una intersección el robot toma una “promesa” al decidir girar en sentido antihorario y continua su recorrido, sin embargo, si vuelve a cruzar por esa intersección no gira a la izquierda, sino que sigue su recorrido hacia adelante.
- *Flood-Fill*. Este algoritmo traducido al español como llenado por inundación, es un método que permite la exploración de todo el entorno y define el camino más eficiente para llegar a la meta. Sin embargo, uno de los inconvenientes de este método es que requiere conocer el entorno de navegación antes de explorarlo para así realizar la navegación.

La simulación de la navegación del robot se sustenta en el modelo matemático de un robot tipo oruga, como el que se muestra en la Figura 5.



Fuente: [Xu, 2019]

Figura 5 Esquema de conexiones del sistema FitoBot.

En este caso, se considera el robot sobre un plano  $x, y$ , y un ángulo de rotación  $\theta$  [Xu, 2019]. Por lo tanto, la Ecuación de posición se describe por el vector de la Ecuación 2.

$$p = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad (2)$$

El cuerpo del robot y el ancho de la oruga son  $L, b$ , mientras que las velocidades de las ruedas izquierda y derecha son  $v_L$  y  $v_R$  respectivamente, por lo que se puede obtener la fórmula del modelo cinemático de los robots, siendo que la expresión de velocidades se describe por la Ecuación 3.

$$\dot{p} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \cos\theta & \frac{1}{2} \cos\theta \\ \frac{1}{2} \sin\theta & \frac{1}{2} \sin\theta \\ -\frac{1}{L+2b} & \frac{1}{L+2b} \end{pmatrix} \begin{pmatrix} v_L \\ v_R \end{pmatrix} \quad (3)$$

Considerando que la velocidad lineal y angular del robot son  $v$  y  $\omega$ , la relación con las velocidades de las ruedas del robot se tiene en la Ecuación 4.

$$\begin{pmatrix} v \\ \omega \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{L+2b} & \frac{1}{L+2b} \end{pmatrix} \begin{pmatrix} v_L \\ v_R \end{pmatrix} \quad (4)$$

Por lo tanto, se obtiene la Ecuación 5.

$$\dot{p} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ \omega \end{pmatrix} \quad (5)$$

Y el radio de giro del robot está dado por la Ecuación 6.

$$R = \frac{v}{\omega} = \frac{L+2b}{2} \cdot \frac{v_L + v_R}{-v_L + v_R} \quad (6)$$

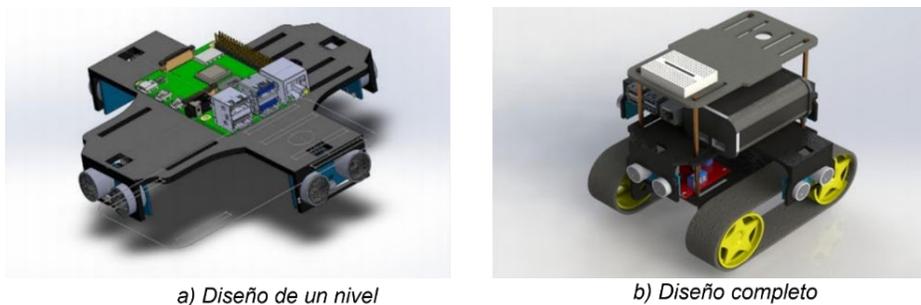
A partir de la Ecuación 6 se puede observar que el estado de movimiento del robot es diferente cuando la velocidad de la rueda izquierda y la rueda derecha son diferentes. De esta forma se tiene tres casos, que el robot seguirá según lo que detecte mediante sensores y en función del algoritmo seleccionado para la navegación. Por ejemplo:

- Si  $v_L = -v_R$ ,  $R = 0$ , el robot se mueve sobre sí mismo.
- Si  $v_L = v_R$ ,  $R \rightarrow \infty$ , muestra que el robot se mueve en línea recta.

- Si  $v_L \neq v_R$ , indica que el robot se mueve circunferencialmente con radio  $R$ .

### 3. Resultados

Con respecto al diseño y construcción de una plataforma robótica móvil, se tuvo como resultado una plataforma robótica móvil que presenta un sistema de locomoción tipo oruga, que le permite desplazarse en superficies irregulares. Esta plataforma está basada en un chasis metálico comercial tipo tanque TP 101, pero con modificaciones a su estructura original, ya que para la integración física de los tres subsistemas descritos en la Figura 4, fue necesario desarrollar una estructura modular de 3 niveles, como la que se observa en la Figura 6. Dicha estructura se diseñó en SolidWorks y se manufacturó con impresión 3D. En el primer nivel se colocaron los componentes encargados del movimiento del robot, los motores, el puente H y el giroscopio. El segundo nivel se utilizó para alojar la Raspberry principal, los sensores ultrasónicos y las baterías, mientras que en el nivel superior se colocaron, el microcontrolador Raspberry Pi W, el sensor BME280, la Raspberry secundaria y el módulo de la cámara.



Fuente: elaboración propia

Figura 6 Diseño de las estructuras adicionales del robot y de la versión final.

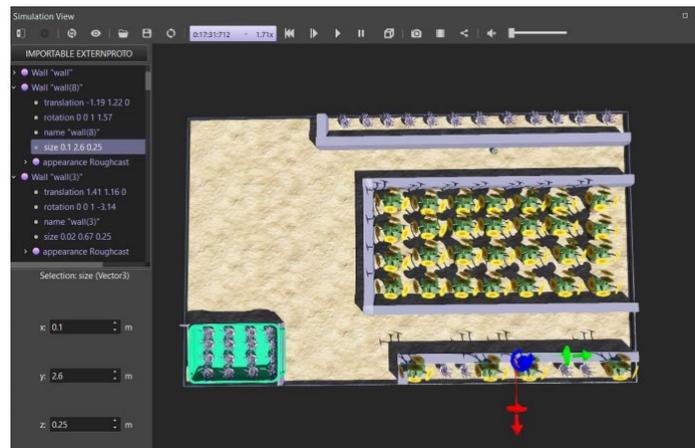
La integración final del robot, con los componentes eléctricos y mecánicos se muestra en la Figura 7. Las dimensiones del prototipo fueron de 19.4 cm de ancho, 21 cm de largo y 17.5 cm de altura, lo que le permite navegar en espacios de 25 cm de ancho y con paredes de altura aproximada a 20 cm. Con respecto a la implementación de algoritmos de navegación autónoma, primero se realizaron pruebas mediante simulación configurando un escenario similar al entorno

experimental, tal y como se observa en la Figura 8. Para ello se utilizó el software de código abierto Webots, desarrollado por Cyberbotics Ltd, Webots [Cyberbotics: Robotics simulation with Webots, s. f.] y se realizaron pruebas con los 4 algoritmos seleccionados, Random-Mouse, Wall-Follower, Pledge y Flood-Fill.



*Fuente: elaboración propia*

Figura 7 Prototipo robótico móvil FitoBot.



*Fuente: elaboración propia*

Figura 8 Escenario de simulación en Webots.

Los resultados del desempeño en tareas de navegación en ambientes de simulación considerando métricas de tiempo de recorrido, tipo de recorrido, porcentaje de superficie explorada y evaluación cualitativa del desempeño durante su recorrido se sintetizan en la Tabla 2. A partir de estos resultados, se procedió a programar el robot. Para ello se utilizó el sistema operativo Raspbian utilizando el lenguaje de programación Python 3.x. Se codificaron los códigos analizados anteriormente utilizando comunicación I2C para la lectura de los sensores ultrasónicos y el sensor

inercial, además de que se crearon archivos .csv para guardar los datos recabados. En la Figura 9, se muestran evidencias de desarrollo experimental dentro de un invernadero.

Tabla 2 Resultados de la simulación de algoritmos de navegación autónoma.

Algoritmo	Random mouse	Wall-Follower	Pledge	Flood-Fill
Tiempo de recorrido [min]	18.45	17.45	17.3	19.3
Tipo de recorrido	Aleatorio	Perimetral	Semiperimetral /aleatorio	Semiperimetral /aleatorio
Porcentaje de superficie explorada	92%	78%	81%	83%
Evaluación cualitativa del desempeño durante su recorrido	Aceptable	Regular	Regular	Regular

*Fuente: elaboración propia*



*Fuente: elaboración propia*

Figura 9 Navegación autónoma de FitoBot en un entorno experimental confinado.

En la Tabla 3 se resume el desempeño de los algoritmos utilizando métricas como tiempo de recorrido, porcentaje de superficie recorrida, número de colisiones presentadas y eficiencia del robot y el consumo de energía.

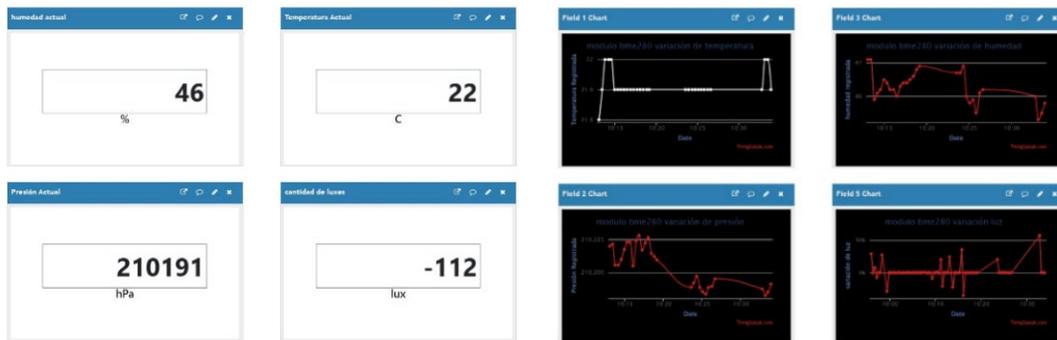
En cuanto a los resultados de los subsistemas de monitoreo remoto de variables ambientales, se desarrolló un sistema IoT que era capaz de monitorear la humedad relativa, la temperatura, la presión barométrica y la luminosidad dentro del espacio experimental. La información recolectada, fue enviada a la nube utilizando el servidor ThinkSpeak mediante protocolo de comunicación HTTP. Un ejemplo de los resultados de estas mediciones se muestra en la Figura 10. En lo que respecta al sistema de visión artificial, se logró configurar al FitoBot para que mediante Python y librerías de OpenCV, fuera capaz de detectar y procesar imágenes de plantas y códigos QR con datos de estas. El propósito de este sistema es obtener información como el nombre o especie, ubicación y determinar mediante reconocimiento visual,

si una planta presenta alguna situación de enfermedad, plaga, estrés o alguna otra condición fitosanitaria que afecte su salud. En la Figura 11, se muestra un diagrama a bloques, del proceso que se ha seguido para el procesamiento y reconocimiento digital de imágenes. En la Figura 12, se muestra evidencia de este proceso. Esta información se almacenó de forma interna para su posterior procesamiento.

Tabla 3 Desempeño de navegación autónoma de FitoBot en entorno físico.

Algoritmo	Random mouse	Wall-Follower	Pledge	Flood-Fill
Tiempo de recorrido [min]	23.8	30.2	25.2	38.3
Eficiencia de desplazamiento	Alta	Alta	Media	Media
Tipo de recorrido	Aleatorio	Perimetral	Semi-perimetral / aleatorio	Semi-perimetral / aleatorio
Porcentaje de superficie explorada	90%	83%	80%	85%
Colisiones en el dispositivo	26	24	22	28
Consumo de energía	Bajo	Bajo	Medio	Alto

Fuente: elaboración propia

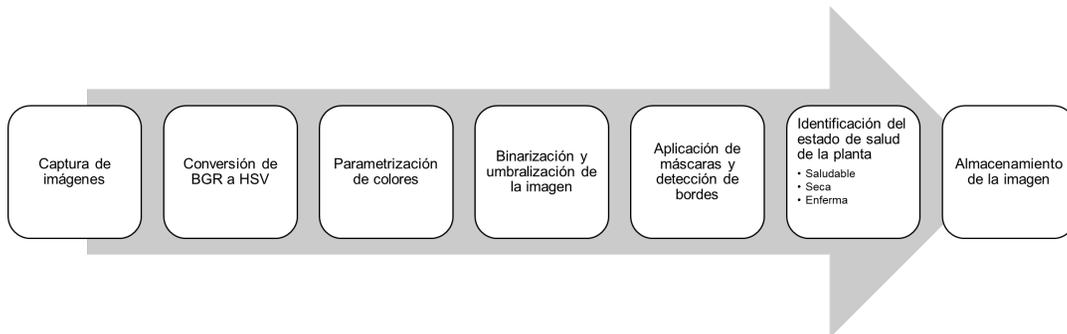


a) Medición de variables.

b) Gráfica de las variables.

Fuente: elaboración propia

Figura 10 Datos ambientales recolectados por FitoBot y visualizados en la nube.



Fuente: elaboración propia

Figura 11 Etapas del procesamiento digital de imágenes.

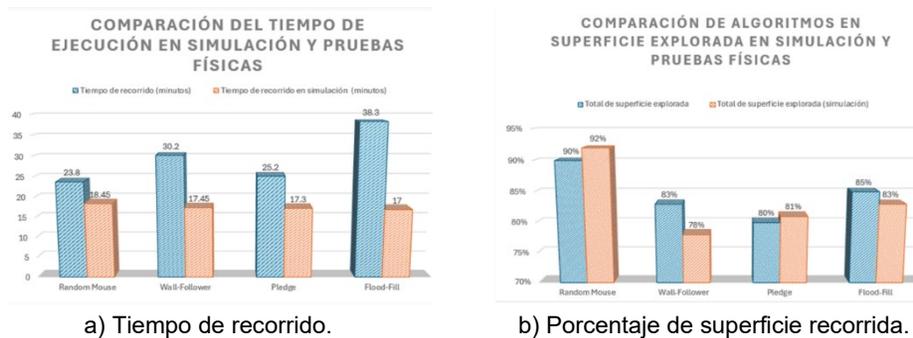


Fuente: elaboración propia

Figura 12 Reconocimiento visual y lectura de códigos QR con FitoBot.

#### 4. Discusión

Con los resultados obtenidos se procedió a realizar un análisis de estos. Para el caso de navegación autónoma, se compararon los tiempos de recorrido entre los cuatro algoritmos analizados, tanto en simulación como en el entorno real, tal y como se observa en la Figura 13a. De forma similar, se comparó el porcentaje de superficie recorrida bajo las mismas condiciones. Los resultados se presentan en la Figura 13b.



Fuente: elaboración propia

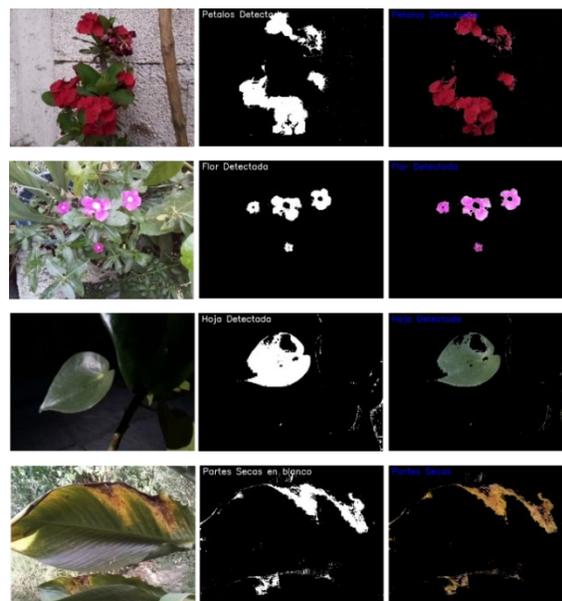
Figura 13 Comparación de métricas de desempeño de navegación autónoma.

De acuerdo con las gráficas de la Figura 13, se puede observar que, en etapa de simulación los cuatro algoritmos analizados requirieron de un tiempo similar para completar el recorrido, siendo ligeramente mejor el algoritmo flood fill. Sin embargo, para el caso de la experimentación en campo, el algoritmo que requirió menor tiempo de navegación fue random mouse. Por lo tanto, bajo este criterio, podría establecerse que, para condiciones de navegación en terrenos de características

similares, no sería necesario implementar algoritmos complejos, por lo que disminuiría la exigencia de recursos computacionales, favoreciendo la simplicidad y bajo costo del prototipo.

Realizando un análisis similar para comparar entre el porcentaje de terreno explorado con cada uno de los algoritmos propuesto, se observó que presentan distinto desempeño. De las gráficas, el algoritmo que permitió un recorrido mayor fue también el algoritmo random mouse. Por lo que se refuerza la conclusión, de que se puede implementar este tipo de sistemas en entornos confinados, sin la necesidad de utilizar algoritmos complejos y de alta demanda computacional.

Por otra parte, con respecto al análisis de las imágenes y con base en los resultados descritos en la Figura 14, se demuestra que FitoBot es capaz de capturar imágenes con la calidad suficiente para procesarlas.



*Fuente: elaboración propia*

Figura 14 Visión artificial para diferentes condiciones fitosanitarias.

Con la ayuda de OpenCV, se realizaron operaciones de procesamiento de imágenes como binarización, umbralización y detección de color, para identificar y reconocer características fitosanitarias específicas de las plantas. Por ejemplo, a través del color rojo y rosa, se lograron detectar los pétalos de las plantas, mientras que, a través del color verde, se identificaron hojas sanas y finalmente mediante el

color amarillo, se observó la condición de una planta con necesidad de agua. Estos resultados de reconocimientos de imágenes son algunos de los resultados iniciales, sin embargo, se planea integrar algoritmos de reconocimiento destinados a estas tareas, para lograr resultados precisos y eficientes.

## **5. Conclusiones**

Como conclusiones de este trabajo se lograron la comparación de algoritmos de navegación tanto en entornos de simulación como en entornos físicos, así como la detección y reconocimiento de imágenes mediante procesamiento computacional utilizando la librería OpenCV.

Habiendo explorado distintos algoritmos de navegación autónoma, se determinó que el algoritmo Random mouse fue el que presentó mejor desempeño de navegación con respecto al tiempo de requerido para explorar el entorno y con respecto al porcentaje de superficie explorada. Por lo que, para investigaciones futuras con este prototipo, será considerado utilizar este algoritmo, dado su simplicidad en programación y requerimientos mínimos de hardware.

Finalmente, en lo referente a monitoreo remoto de variables ambientales, los datos recabados serán útiles para realizar predicciones respecto a dinámica de los microclimas dentro de los entornos explorados y estudiar así, los factores que influyen en las condiciones fitosanitarias del entorno. En lo que respecta a la visión artificial, ha sido posible visualizar distintas plantas e identificar a través del reconocimiento de color condiciones específicas de cada una de ellas.

Como futuras líneas de investigación se plantea clasificar aspectos fitosanitarios utilizando herramientas de aprendizaje automático, explorar otros algoritmos de navegación y analizar datos ambientales para una gestión de recursos más eficiente.

## **6. Bibliografía y Referencias**

- [1] Ahmad, I., Yang, Y., Yue, Y., Ye, C., Hassan, M., Cheng, X., Wu, Y., & Zhang, Y. (2022). Deep learning based detector yolov5 for identifying insect pests. *Applied Sciences*, 12(19), 10167. <https://doi.org/10.3390/app121910167>.

- [2] Alvarez, G., & Flor, O. (2020). Desempeño en métodos de navegación autónoma para robots móviles. *Minerva*, 1(2), 19-29. <https://doi.org/10.47460/minerva.v1i2.8>.
- [3] Cyberbotics: Robotics simulation with Webots. (s. f.). Recuperado 16 de julio de 2024, de <https://cyberbotics.com/>.
- [4] Kakani, V., Nguyen, V. H., Kumar, B. P., Kim, H., & Pasupuleti, V. R. (2020). A critical review on computer vision and artificial intelligence in food industry. *Journal of Agriculture and Food Research*, 2(1).
- [5] Planas de Martí, S. (2018). Agricultura de precisión y protección de cultivos. *Revista de Ingeniería*, 47.
- [6] Quy, V. K., Hau, N. V., Anh, D. V., Quy, N. M., Ban, N. T., Lanza, S., Randazzo, G., & Muzirafuti, A. (2022). IoT-enabled smart agriculture: Architecture, applications, and challenges. *Applied Sciences*, 12(7), 3396. <https://doi.org/10.3390/app12073396>.
- [7] Rodríguez, F., Sánchez-Hermosilla, J., Rivera, L., & Giménez, A. (2016). Análisis de viabilidad de la utilización de robots en invernaderos. Propuesta para el modelo Almería.
- [8] Santos Valle, S. y Kienzle, J. (2021). Agricultura 4.0: Robótica agrícola y equipos automatizados para la producción agrícola sostenible. *Gestión integrada de cultivos*, N. 24. Roma, FAO.
- [9] Sharma, A., Jain, A., Gupta, P., & Chowdary, V. (2021). Machine Learning Applications for Precision Agriculture: A Comprehensive Review. *IEEE Access*, 9, 4843–4873.
- [10] Solomon, C., & Breckon, T. (2011). *Fundamentals of Digital Image Processing: A practical approach with examples in Matlab*. John Wiley & Sons.
- [11] Sood, S., & Singh, H. (2021). Computer vision and machine learning based approaches for food security: A review. *Multimedia Tools and Applications*, 80(18), 27973-27999.
- [12] TecNM. (2024). *Agenda estratégica del TecNM para la autosuficiencia alimentaria y rescate del campo mexicano*. Tecnológico Nacional de México.

[https://www.tecnm.mx/pdf/agendas/Autosuficiencia\\_alimentaria\\_y\\_rescate.pdf](https://www.tecnm.mx/pdf/agendas/Autosuficiencia_alimentaria_y_rescate.pdf).

- [13] United Nations Department of Economic and Social Affairs. (2023). Informe de los Objetivos de Desarrollo Sostenible 2023: Edición especial. [Erscheinungsort nicht ermittelbar] United Nations.
- [14] Wu, D., Jiang, S., Zhao, E., Liu, Y., Zhu, H., Wang, W., & Wang, R. (2022). Detection of camellia oleifera fruit in complex scenes by using yolov7 and data augmentation. *Applied Sciences*, 12(22), 11318. <https://doi.org/10.3390/app122211318>.
- [15] Xu, R. (2019). Path planning of mobile robot based on multi-sensor information fusion. *EURASIP Journal on Wireless Communications and Networking*, 2019 (1), 44.
- [16] Zhou, Y., Tang, Y., Zou, X., Wu, M., Tang, W., Meng, F., Zhang, Y., & Kang, H. (2022). Adaptive active positioning of camellia oleifera fruit picking points: Classical image processing and yolov7 fusion algorithm. *Applied Sciences*, 12(24), 12959. <https://doi.org/10.3390/app122412959>.