

# UNIVERSIDAD MILITAR NUEVA GRANADA

FACULTAD DE INGENIERIA MECATRONICA



## MONOGRAFIA: DISPOSITIVO ESCALADOR

ALUMNOS: JUAN ESTEBAN VIVAS HERNANDEZ  
DUVAN STIVELROMERO PINILLA  
DILAN EDUARDO CUELLAR CORREDOR  
DAVID FERNANDO SALAZAR MUÑOZ  
NICOLAS LAMILLA VEGA  
OSCAR ANDRES PAEZ CASAS

PROFESOR: JORGE ALEXANDER APONTE RODRIGUEZ

MATERIA: DISEÑO MECATRONICO

AÑO: 2024-1

# **I. Índice**

1. Glosario
2. Resumen
3. Introducción
4. Delimitación

## 4.1. Objetivos Generales

## 4.2. Objetivos Específicos

## 5. Desarrollo del Tema

### 5.1. Contexto del proyecto

### 5.2. Investigación de diseño

### 5.3. Diseño del Dispositivo Robot Oruga Escalador

#### 5.3.1. Requisitos y Especificaciones

#### 5.3.2. Cinemática

#### 5.3.3. Mecánica y Estructura

#### 5.3.4. Sistema de Propulsión

#### 5.3.5. sistema de alimentación

#### 5.3.6. Sistema de Navegación y Control

#### 5.3.7. implementación total

### 5.4. Implementación y Pruebas

#### 5.4.1. Prototipo y Fabricación

#### 5.4.2. Pruebas de Funcionamiento

5.5. Resultados

5.6. Conclusiones

6. Anexos

Anexo A: Diagramas y planos

Anexo B: Gastos y cotizaciones

Anexo C: Códigos

# **1. Glosario**

- Dispositivo robot oruga escalador: Robot capaz de desplazarse verticalmente por escaleras mediante un mecanismo de orugas.
- Autonomía: Capacidad de funcionar de manera independiente sin intervención humana.
- Propulsión: Sistema que proporciona la fuerza necesaria para el movimiento del robot.
- Navegación: Proceso de determinar la ruta óptima y evitar obstáculos durante el desplazamiento.
- Control: Conjunto de algoritmos y sistemas que regulan el movimiento y comportamiento del robot.
- Sensores: Dispositivos utilizados para detectar y medir variables del entorno, como proximidad, temperatura, presión, entre otros.
- Sistema de alimentación: Conjunto de componentes que suministran energía eléctrica al robot, como baterías y sistemas de gestión de energía
- Motor brushless: Tipo de motor eléctrico que no requiere escobillas para la conmutación de energía en el rotor, utilizando en su lugar un control electrónico. Se caracterizan por su eficiencia y durabilidad, siendo comúnmente utilizados en aplicaciones que requieren funcionamiento suave y eficiente, como en robots y vehículos eléctricos.
- BMS (Sistema de Gestión de Batería): Controla y protege la batería, equilibrando el voltaje y evitando problemas como sobrecargas y cortocircuitos.

- Estructura B2P2: Es un modelo inicial de un vehículo terrestre no tripulado que utiliza un sistema de orugas para su movilidad. El nombre "B2P2" identifica este prototipo en particular, haciendo referencia a características específicas del diseño, componentes, o alguna otra característica distintiva del vehículo.
- Denavit-Hartenberg: Es una convención utilizada en robótica para describir la cinemática de un robot manipulador. Proporciona un método sistemático para asignar marcos de referencia a cada articulación del robot, lo que simplifica la descripción matemática de su geometría y cinemática.

<b>Lista de figuras</b>	<b>pg</b>
1. Planos robot con rueda tipo estrella	10
2. Plano robot con palanca trasera	11
3. Plano robot tipo oruga	12
4. Descomposición funcional	13
5. geometría de un robot diferencial	16
6. Plano estructura B2P2	17
7. Asignación de marcos	17
8. Parámetros de Denavit-Hartenberg	18
9. Plano del chasis	19
10. Plano de bandeja	19
11. Ensamble del chasis	20
12. Implementación eléctrica	21
13. Conexión de motores	22
14. Motor brushless	24
15. SERVO MG995	25
16. Bateria	26
17. Drivers Riorand	27
18. regulador DC	27

19.Arduino MEGA	28
20.Sharp	28
21.interfaz	29
22. Modelo red neuronal	30
23.prueba trayectorias	31
24. Prototipo final	32
25. Barra de hierro	33
26. Cola de pato	34

## **Lista de Tablas**

1.Casa de la calidad	13
2. Matriz de decisión	14

## **Lista de Anexos**

1.Plano de partes del diseño	36
2.Gastos indirectos	38
3.codigos	39

## **2. Resumen:**

En esta monografía se ha presentado el diseño y desarrollo de un robot oruga escalador. El robot está diseñado para transportar objetos desde el sótano hasta el tercer piso de la universidad militar. El robot utiliza un mecanismo de oruga para moverse por superficies planas y un sistema de escalar escaleras para subir y bajar peldaños. El robot también está equipado con sensores y un sistema de navegación para evitar obstáculos y llegar a su destino de forma segura

### **3. Introducción:**

La movilidad en entornos complejos y en estructuras con múltiples niveles representa un desafío significativo para la robótica contemporánea. En este contexto, el diseño y desarrollo de dispositivos robóticos capaces de navegar de manera eficiente y autónoma por escaleras se ha convertido en un área de investigación y desarrollo de gran interés. Estos dispositivos, como el robot oruga escalador de escaleras propuesto en este proyecto, tienen el potencial de ofrecer soluciones innovadoras para una amplia gama de aplicaciones.

El trabajo se enfoca en el diseño y desarrollo de un dispositivo robot oruga escalador de escaleras para abordar los desafíos de la movilidad vertical en entornos urbanos. Este dispositivo tiene la capacidad única de desplazarse de manera segura y eficiente desde el sótano hasta el tercer piso de un edificio, superando obstáculos y cambios de inclinación presentes en las escaleras convencionales.

El diseño de este robot oruga se basa en una cuidadosa consideración de los principios de la mecánica, la electrónica y la informática. Se prestó especial atención al desarrollo de un sistema de propulsión robusto y eficiente, capaz de dar la tracción necesaria para superar los obstáculos de las escaleras. Además, se ha integrado un sistema de navegación y control avanzado que permite al robot detectar y evitar obstáculos, planificar rutas óptimas y mantener el equilibrio durante el desplazamiento.

El dispositivo propuesto representa una contribución significativa al campo de la robótica móvil, ya que combina tecnologías de vanguardia con un enfoque práctico y orientado a la aplicación. Se espera que este robot oruga escalador de escaleras tenga un amplio impacto en una variedad de industrias y aplicaciones, desde la logística y el transporte hasta la seguridad y el rescate en situaciones de emergencia.

En las secciones siguientes, se presentará en detalle el proceso de diseño y desarrollo del dispositivo, incluyendo el contexto y la motivación del proyecto, el estado del arte en tecnologías relacionadas, el diseño mecánico y electrónico del

robot, las pruebas y validaciones realizadas, y las conclusiones y perspectivas futuras. Este trabajo representa un paso importante hacia la creación de soluciones robóticas innovadoras y eficientes para la movilidad vertical en entornos urbanos complejos.

## **4. Delimitación**

### **4.1. Objetivos Generales**

Desarrollar e implementar un dispositivo autónomo de transporte para portátiles de características estándar, capaz de seguir una ruta específica, superando obstáculos como escaleras. El diseño y construcción del dispositivo incluirá componentes estructurales propios, permitiendo el uso de partes estandarizadas para mejorar su eficiencia y seguridad. El proyecto se llevará a cabo durante el presente semestre, enfocado en garantizar la seguridad de quienes interactúen con el dispositivo, asegurando un transporte eficiente y confiable del portátil, implementando tecnologías innovadoras.

### **4.2 Objetivos específicos**

- Realizar pruebas de validación del diseño y la construcción de la estructura del dispositivo, asegurando su resistencia y durabilidad bajo diversas condiciones de uso, específicamente el ascenso de escaleras.
- Realizar pruebas exhaustivas del dispositivo para asegurar su capacidad autónoma y eficiente para completar satisfactoriamente el recorrido.
- Evaluar la eficacia del dispositivo para completar la ruta propuesta en un tiempo no mayor de 15 minutos, todo esto garantizando la integridad de las personas que interactúen con el dispositivo.

## **5. Desarrollo del tema**

### **5.1. contexto del proyecto**

Este proyecto no solo aborda la necesidad de desarrollar tecnologías robóticas para la movilidad vertical, sino que también responde a la demanda interna de gestionar proyectos científicos dentro de las instalaciones de nuestra universidad. La creación de un dispositivo robot oruga escalador de escaleras no solo representa un avance en ingeniería robótica, sino también una oportunidad para fortalecer la investigación y la colaboración dentro de nuestra comunidad académica.

La gestión de este proyecto científico dentro de nuestras instalaciones universitarias ofrece una serie de ventajas, incluyendo el acceso a recursos y expertos en diversas disciplinas, la oportunidad de involucrar a estudiantes en actividades de investigación y desarrollo, y la posibilidad de establecer colaboraciones con otras instituciones y empresas. Además, la realización de este proyecto en un entorno académico fomenta el intercambio de conocimientos y la formación de futuros profesionales en el campo de la robótica y la ingeniería.

Aunque el proyecto se centra inicialmente en el diseño y desarrollo de un dispositivo robot oruga escalador de escaleras para usarlo en las instalaciones de la universidad, puede que esta tecnología se extienda más allá de los límites de nuestra institución en el futuro. La colaboración con empresas locales y la participación en proyectos de investigación a nivel nacional e internacional podrían abrir nuevas oportunidades para la aplicación y comercialización de esta tecnología en una variedad de contextos fuera de la universidad.

## **5.2. Investigación del diseño**

Durante la fase inicial de investigación y evaluación de tecnologías existentes, se consideraron varios tipos de robots para abordar el desafío de subir escaleras. Entre las opciones evaluadas se encontraban el robot con ruedas tipo estrella hexápodo y el robot tipo carro con una palanca trasera. Sin embargo, tras un análisis exhaustivo de las ventajas y limitaciones de cada tipo de robot, se decidió optar por el diseño de un robot tipo oruga para este proyecto.

El principal factor que influyó en esta decisión fue la capacidad del robot tipo oruga para ofrecer una combinación única de tracción, estabilidad y versatilidad en una variedad de superficies y entornos. A diferencia del robot con ruedas tipo estrella hexápodo, que podría enfrentar dificultades para adaptarse a escaleras y desniveles, y del robot tipo carro con una palanca trasera, cuyo diseño podría ser menos eficiente en términos de maniobrabilidad y capacidad de escalada, el robot

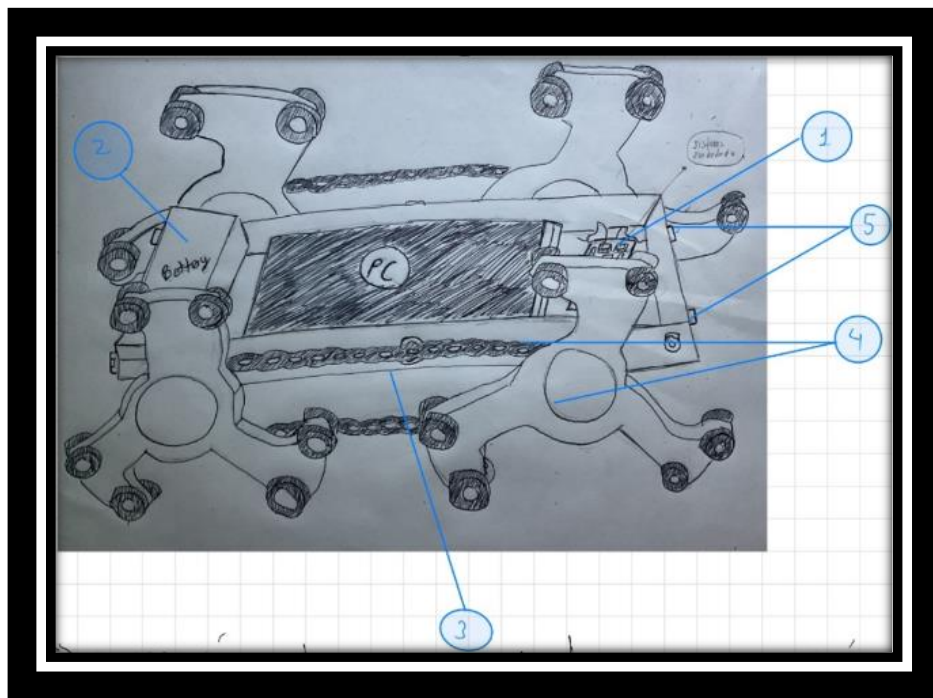


tipo oruga ofrece una solución robusta y probada para la navegación en entornos verticales complejos.

Además, el diseño modular y la configuración flexible de un robot tipo oruga permiten una fácil adaptación a diferentes tamaños y geometrías de escaleras, así como la integración de sensores y sistemas de control avanzados para la navegación autónoma. Estas características hacen del robot tipo oruga la elección ideal para abordar los desafíos específicos de movilidad vertical que presenta este proyecto.

- **Robot con Ruedas Tipo Estrella Hexápodo**

Al principio se consideró al robot con ruedas tipo estrella hexápodo por su capacidad para desplazarse sobre terrenos irregulares, lo que sugiere una posible adaptabilidad a escaleras y desniveles. Estudios recientes han demostrado que este tipo de configuración puede presentar limitaciones en estabilidad y capacidad de escalada en superficies inclinadas o con obstáculos [1]. Además, la complejidad mecánica y cinemática de un robot hexápodo puede aumentar la dificultad de implementar sistemas de navegación y control efectivos para la navegación autónoma en entornos urbanos [2].



*Figura 1 Planos robot con rueda tipo estrella*

Donde:

1. Navegación autónoma, control y comunicación
2. Alimentación de energía
3. Integración del sistema
4. Seguridad y fiabilidad
5. Sensores y percepción

### • Robot Tipo Carro con Palanca Trasera

El robot tipo carro con una palanca trasera se consideró por su potencial para superar obstáculos y cambios de inclinación utilizando un mecanismo de impulso. Este diseño puede presentar limitaciones en términos de estabilidad y eficiencia energética durante la navegación en escaleras y superficies irregulares [3],

investigaciones previas han señalado que. Además, la dependencia de un mecanismo de palanca para la tracción puede aumentar la complejidad mecánica del robot y dificultar la implementación de sistemas de control efectivos para la navegación autónoma [4].

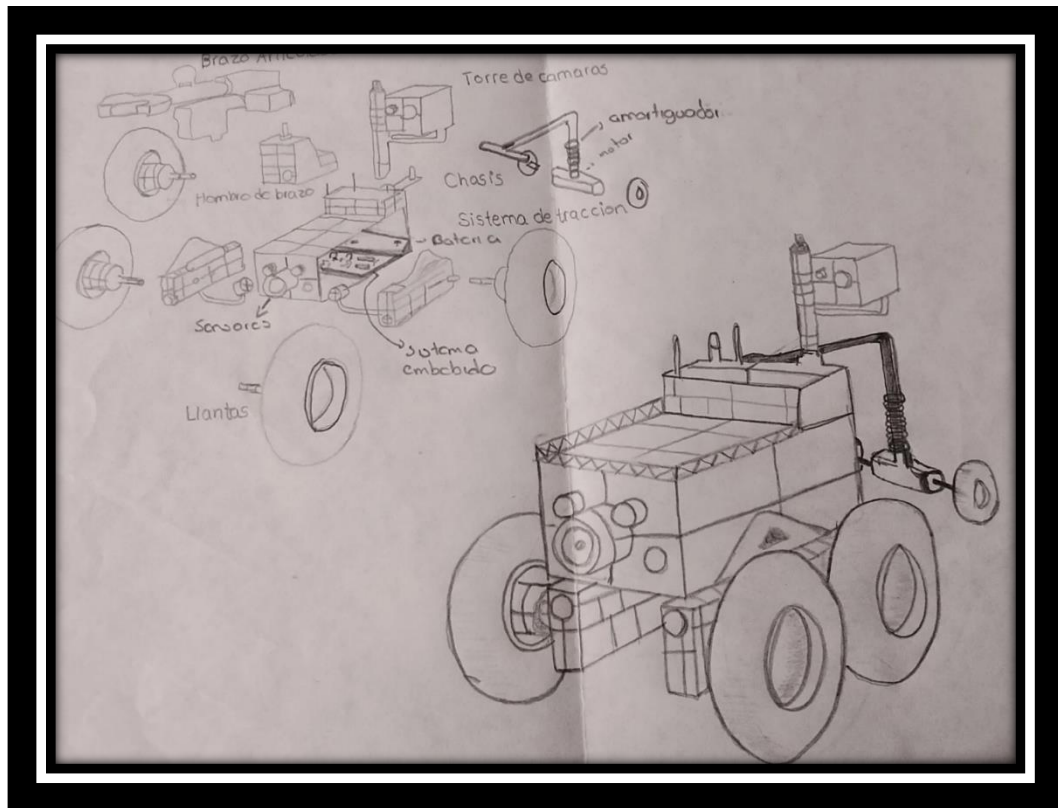
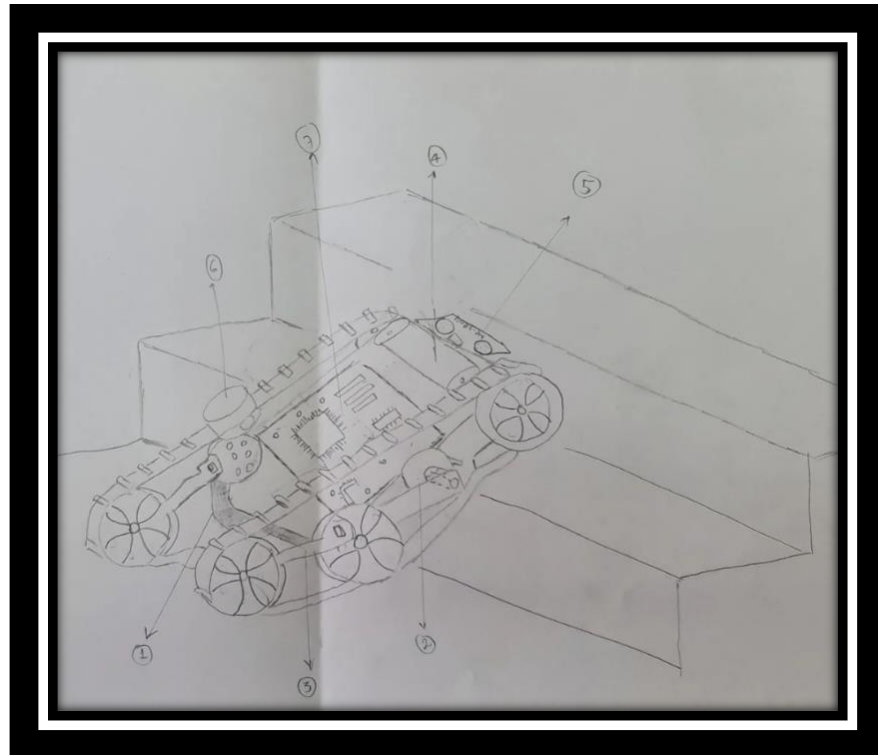


Figura 2. Plano robot con palanca trasera

- **El robot escalador tipo oruga**

Ha sido seleccionado como la opción preferida para abordar los desafíos de movilidad vertical en entornos urbanos y estructuras con múltiples niveles. Este tipo de robot se caracteriza por su capacidad para proporcionar una tracción excepcional y una estabilidad superior en una variedad de superficies, incluyendo escaleras y desniveles. Investigaciones previas han demostrado que los robots escaladores tipo oruga son capaces de superar obstáculos y cambios de inclinación de manera eficiente y segura, lo que los convierte en una opción ideal para aplicaciones de transporte y exploración en entornos urbanos [5]. Además, la configuración modular y la flexibilidad de diseño de los robots tipo oruga permiten una fácil adaptación a diferentes entornos y aplicaciones específicas, así como la integración de sistemas avanzados de navegación y control para la autonomía [6].



*Figura 3. Plano robot tipo oruga*

Donde:

1. Cajón para el pc
2. Control de los motores
3. Control del modulo
4. Sistema de alimentación (batería)
5. Sensor de proximidad
6. Regulador de batería
7. Integración de sistema

### **5.3. Diseño del dispositivo**

#### **5.3.1. Requisitos y Especificaciones**

Para el diseño lo primero fue realizar una serie de estudios para un proyecto y así saber primero si era viable y también como empezar a realizarlo. Estos son:

- **Descomposición funcional**

Al desglosar el sistema en funciones definidas, como locomoción y navegación, sensores y percepción, control y actuadores, seguridad y fiabilidad, alimentación de energía, comunicación y control autónomo, y la integración del sistema, podemos abordar eficazmente los aspectos clave del proyecto.

La descomposición funcional nos permite una visión detallada de cada aspecto del proyecto, lo que va a facilitar la asignación de recursos, la planificación de actividades y la toma de decisiones estratégicas. Además, se puede observar en el diseño la relación de cada parte de la descomposición con las partes del dispositivo como se muestra en la figura 3.

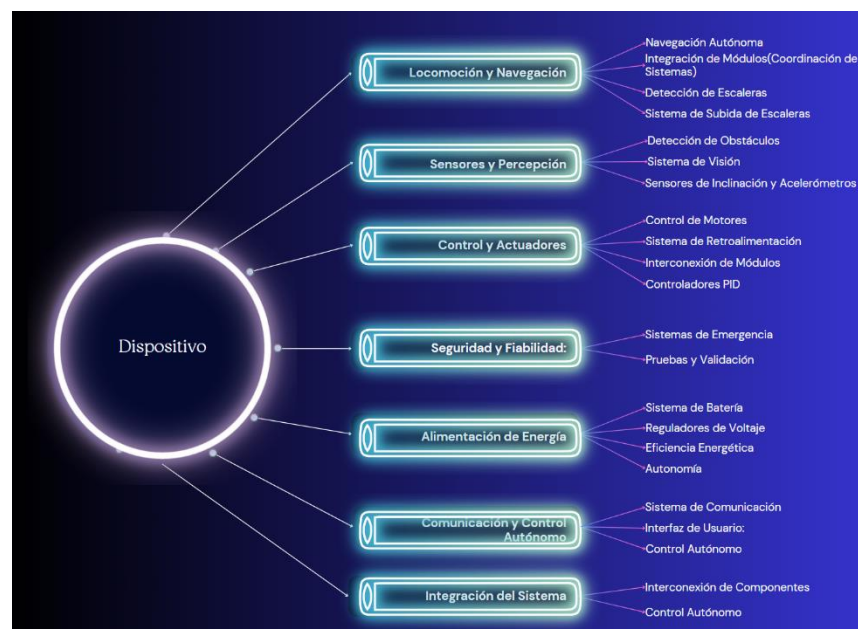


Figura 4. Descomposición funcional

- **Casa de la calidad**

Luego se utilizó otro método el cual fue la casa de la calidad con base en el proyecto la cual se utilizó para la gestión de la calidad para identificar y priorizar los requisitos del cliente y traducirlos en características del producto o servicio que se deben cumplir.

El chasis es la principal característica, pero no más importante si se considera, aunque tenga más peso que las demás, todas las características son importantes

para que el proyecto tenga un buen desempeño. Pero el chasis es importante sobre todo por el peso y la distribución de las demás implementaciones dentro el dispositivo. Todo esto se puede evidenciar en la tabla 1.

		características de ingeniería						
		↑	na	↓	na	na	↓	↓
	FUNCION DE PESO	sistema de control	forma del chasis	potencia actuadores	sistema de comunic	interfaz	dimensiones	# modulos
seguridad	5	5	3	5	5	3	1	1
costo	4	3	5	5	3	3	3	5
resistente a golpes	3	0	5	1	0	0	3	1
facil de manufacturar	4	0	5	0	0	0	5	5
materiales reciclables	3	0	5	0	0	0	3	3
facil ensamblaje	4	0	3	0	0	0	3	5
bajo consumo de energia	4	5	5	5	3	1	3	3
Interfaz Amigable con el usuario	4	1	0	1	1	5	0	0
i	Importancia absoluta	61	117	72	53	51	79	89
	Importancia relativa	11,68582375	22,4137931	13,7931	10,1533	9,77	15,1341	17,049808
	Decisión de ingeniería	5	1	4	6	7	3	2

*Tabla 1 Casa de la calidad*

- **Matriz de decisión**

Por último, se utilizó una matriz de selección en donde se evidencia porque se tomó la decisión de utilizar el modelo de robot oruga que fue antes mencionado en el apartado del estado del arte, con respecto a ciertas características, esto nos ayuda a evaluar y comparar varias alternativas u opciones en función de un conjunto de

criterios predefinidos. En el contexto de tu proyecto de diseño y desarrollo de un robot escalador tipo oruga. Como se muestra en la tabla 2.

Especiales de diseño	Funcion de peso	↑	↓	→
		diseño 1 (oruga)	diseño 2 (diferencial con aspas)	diseño 3 (diferencial con amortiguador)
sistema de control	11,668	9	8	9
forma del chasis	22,41	10	7	9
potencia actuadores	13,79	8	9	8
sistema de comunicació n	10,15	4	4	4
interfaz	9,77	5	5	5
dimensiones	15,13	10	7	10
# modulos	17,04	10	3	8

Tabla 2 *matriz de decisión*

### 5.3.2. Cinemática

El estudio cinemático que se utilizó para el dispositivo fue el de un robot móvil ya que los movimientos que se realizarán serán similares, en donde se fijó un sistema de referencia  $\Sigma_o(X_o, Y_o, Z_o)$  de una forma conveniente. Luego se ubicó el sistema de referencia  $\Sigma_r(X_r, Y_r, Z_r)$

Lo cual nos permite hallar los componentes de translación y la orientación de un robot móvil respecto al sistema de referencia  $\Sigma_o$ , todo esto se muestra en la figura 5.

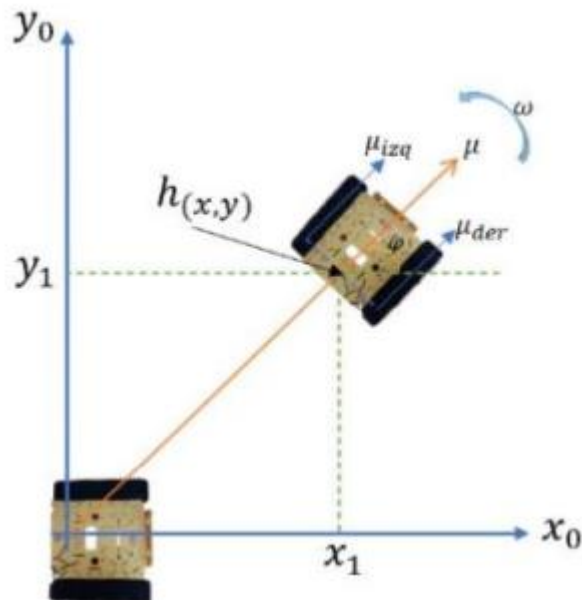


Figura 5. geometría de un robot diferencial

Luego como resultado del análisis geométrico la cinemática diferencial con el punto de control ubicado en el centro del eje que une las ruedas se da por las siguientes ecuaciones:

$$\begin{aligned} h_x &= x_1 & \dot{h}_x &= \mu \cos(\varphi) \\ h_y &= y_1 & \dot{h}_y &= \mu \sin(\varphi) \end{aligned}$$

$$\dot{\varphi} = \omega$$

$$\begin{bmatrix} \dot{h}_x \\ \dot{h}_y \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \cos\varphi & 0 \\ \sin\varphi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mu \\ \omega \end{bmatrix}$$

$$\begin{matrix} \uparrow & & \uparrow & & \uparrow \\ \dot{h} & & J & & \dot{q} \end{matrix}$$

$\mu$  = Velocidad lineal  
 $\omega$  = Velocidad angular  
 $J$  = Matriz Jacobiana



Luego para describir la estructura cinemática nos enfocamos en la estructura B2P2 que es la se utilizara para nuestro dispositivo, y así poder cumplir el objetivo de subir las escaleras.

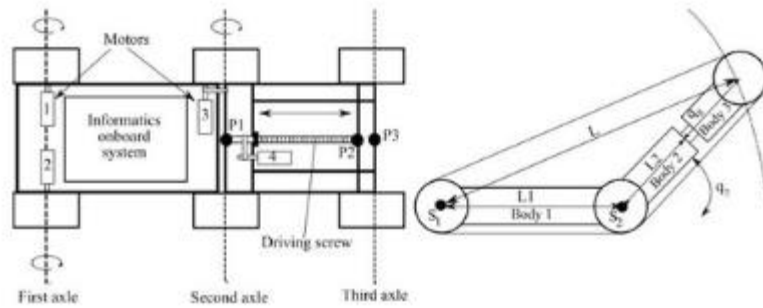


Figura 6. Plano estructura B2P2

Luego para esta descripción se utilizó el método en robótica de Denavit Hartenberg en dónde asignamos marcos de referencia como se muestra en la figura 7.

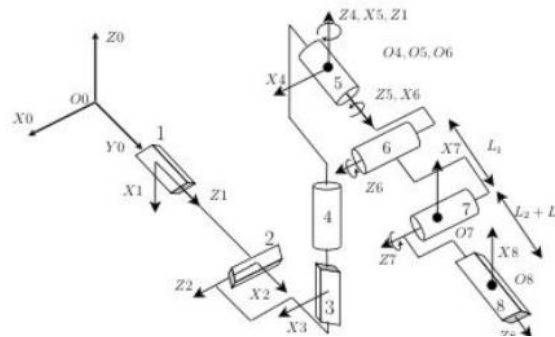


Figura 7Asignación de marcos

Luego de la asignación de marcos se pueden hallar los parámetros respectivos para esta estructura, en la figura 8 se pueden observar

$j$	$\sigma_j$	$\alpha_j$	$d_j$	$\theta_j$	$r_j$
1	1	$-\frac{\pi}{2}$	0	$\frac{\pi}{2}$	$q_1$
2	1	$\frac{\pi}{2}$	0	$\frac{\pi}{2}$	$q_2$
3	1	$-\frac{\pi}{2}$	0	$-\frac{\pi}{2}$	$q_3$
4	0	0	0	$q_4$	0
5	0	$-\frac{\pi}{2}$	0	$q_5 - \frac{\pi}{2}$	0
6	0	$-\frac{\pi}{2}$	0	$q_6 - \frac{\pi}{2}$	0
7	0	0	$L_1$	$q_7 + \frac{\pi}{2}$	0
8	1	$\frac{\pi}{2}$	0	0	$L_2 + q_8$

Figura 8. parámetros de Denavit-Hartenberg

### 5.3.3. Mecánica y estructura

En esta sección se analizarán los planos y desde esto quedará una de las partes más importantes en la estructura del chasis, pues vendrán todas las partes y donde estará casi el soporte para que el dispositivo tenga una estructura funcional.

Se puede observar la forma del chasis que será rectangular y sus medidas será de 37.7X30 cm.

El chasis será a base de aleación de aluminio 6061 el cual es popular debido a su alta resistencia y excelente mecanizabilidad y se usa en aplicaciones que requieren resistencia estructural, y es comúnmente utilizado para proyectos de ingeniería.

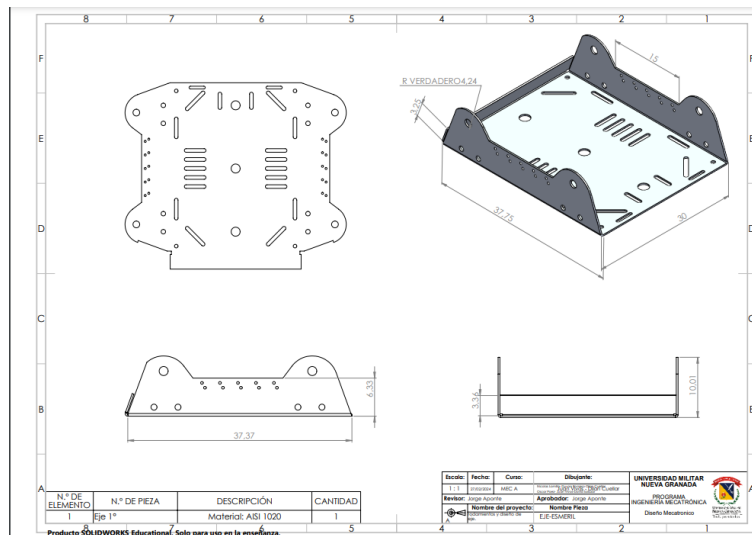


Figura 9. Plano del chasis

Se diseñó la bandeja del robot para transportar al pc, uno de los objetivos principales del proyecto, para esta pieza se necesitó un tamaño adecuado para un computador comercial de 14 pulgadas aproximadamente.

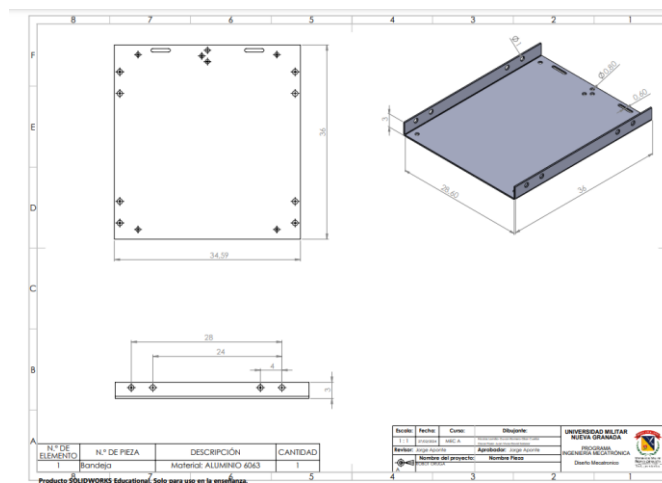
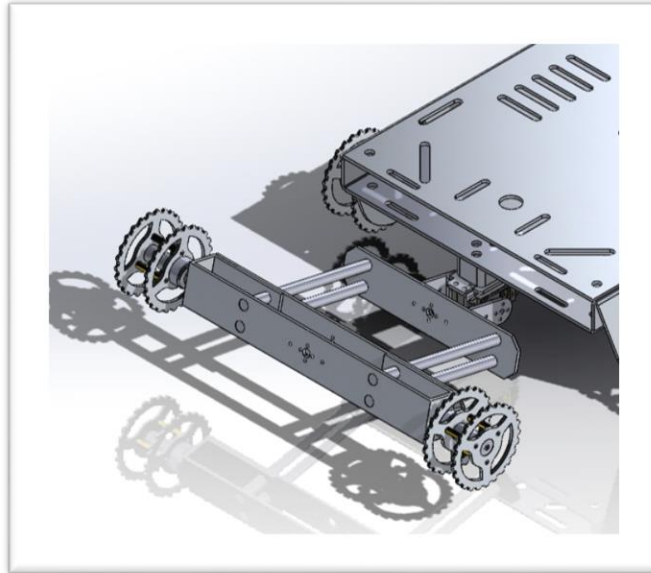


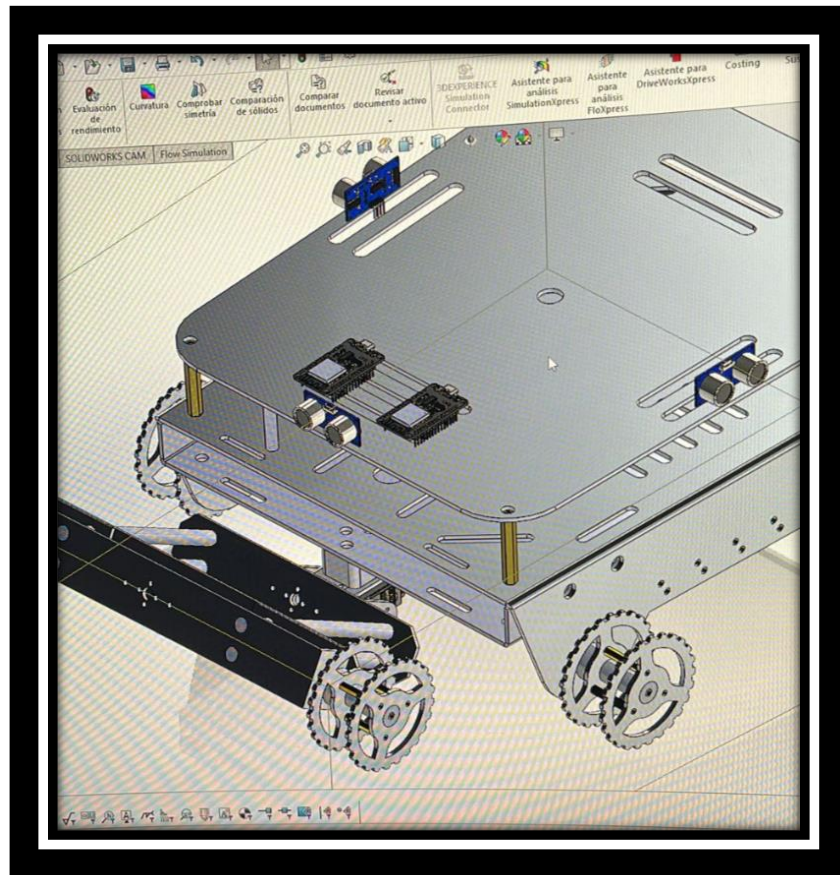
Figura 10. plano de bandeja

Para que haya un tipo de soporte entre el chasis y la bandeja se diseñó una pieza separadora, la cual servirá para amortiguar el contacto entre estos dos elementos antes mencionados, el material de la pieza será aluminio 6061. Para la base separadora, se diseñó una pieza específica destinada a separar un componente del robot. Esta base se concibió para proporcionar un soporte adecuado y seguro, cumpliendo así con los requerimientos del proyecto según el anexo 1.



*Figura 11. Ensamble del chasis*

Para el ensamble del proyecto, se diseñó una configuración integral que integra todos los componentes de manera funcional. Este ensamble fue meticulosamente planificado para garantizar la cohesión y la compatibilidad entre las diversas partes del robot, asegurando así su óptimo rendimiento y funcionamiento coordinado.



*Figura 12. Implementación eléctrica en chasis*

Para la implementación eléctrica en el chasis, se diseñó un sistema cuidadosamente integrado de cables, conectores y componentes electrónicos. Este diseño se centró en garantizar una distribución eficiente de la energía eléctrica dentro del chasis, así como en proporcionar conexiones seguras y confiables para todos los dispositivos y circuitos necesarios. El objetivo principal fue optimizar el rendimiento eléctrico del robot y garantizar su funcionamiento sin fallos.

### 5.3.4. Sistema de propulsión

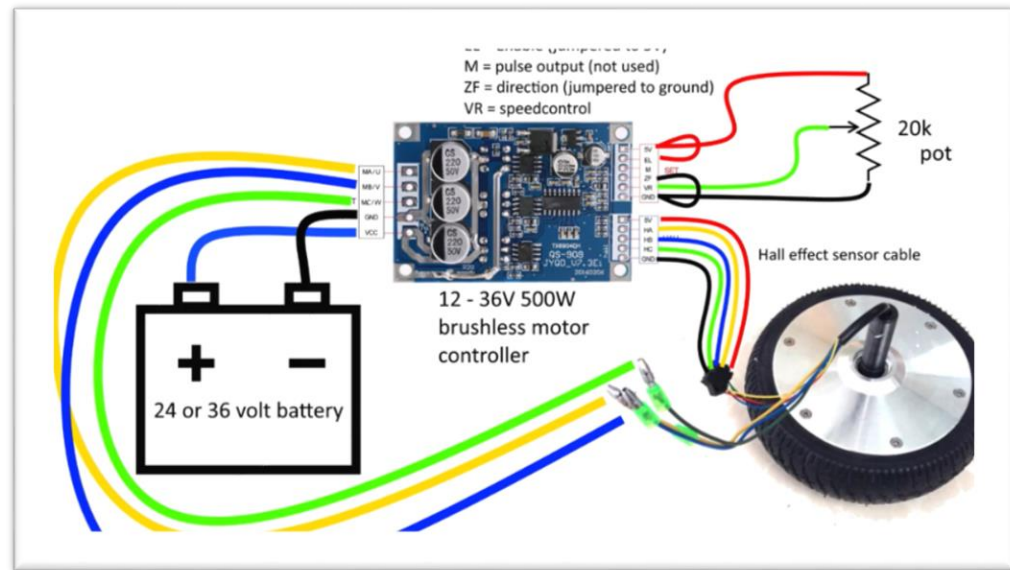


Figura 13. Conexión de Motores

Para hallar el torque necesario para subir las escaleras se realizaron los siguientes cálculos, y luego al torque mínimo se dividió en 2, para saber cuál sería el torque para cada uno de los motores que utilizaremos

$$w = 15kg \cdot 9.8 \frac{m}{s^2} = 147N$$

(1)

con un angulo de  $45^\circ$

(2)

$$Wx = w \cdot \text{sen}(45) = 147 \cdot \text{sen}(45) = 103.94$$

(3)

$$W_y = w \cdot \cos(45) = 147 \cdot \cos(45) = 103.94$$

(4)

$$\mathcal{M}_d = 0.1$$

(5)

$$\theta = 45^\circ$$

(6)

$$fd = \mathcal{M}_d \cdot N$$

(7)

$$\Sigma F_x = 0$$

(8)

$$F - fd - W_x = 0$$

(9)

$$F = \mathcal{M}_d \cdot N + W_x$$

(10)

$$F = (0.1)(103.94) + 103.94 = 114.33N$$

(11)

$$\Sigma F_y = 0$$

(12)

$$N - W_y = 0$$

(13)

$$Torque = F \cdot \text{Brazo palanca}$$

(14)

$$\text{Brazo palanca} = 0.05$$

(15)

$$T = 114.33 \cdot 0.05 = 5.7165 \text{ N} \cdot \text{m}$$

(16)

*Conversion a Kg · cm para motor comercial*

(17)

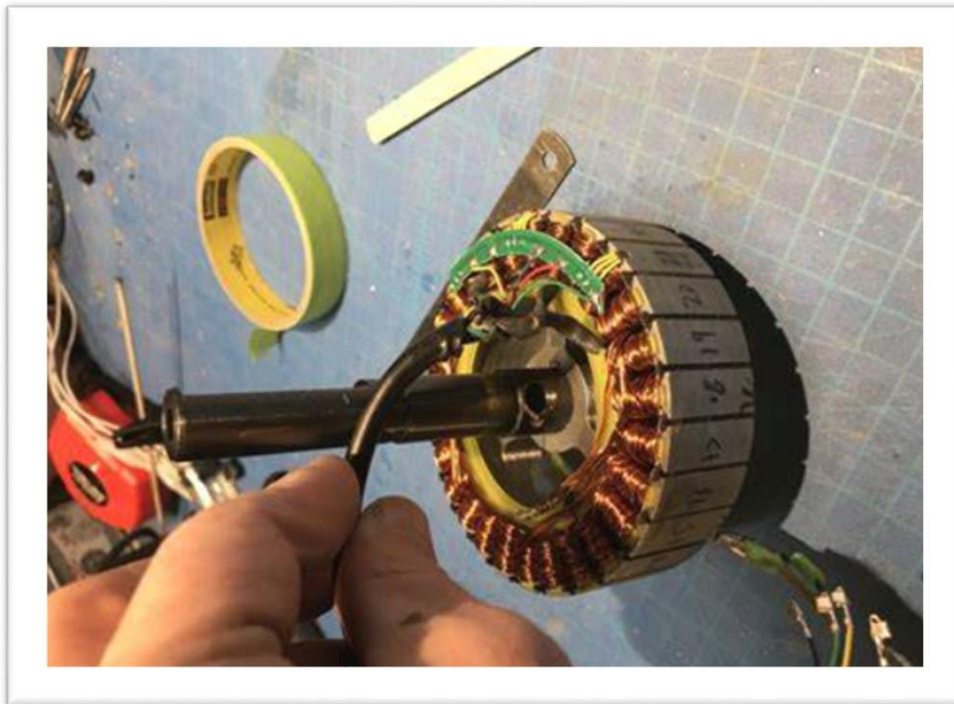
$$5.7165 \text{ N} \cdot \text{m} = 58.33163 \text{ Kg} \cdot \text{cm}$$

(18)

$$\frac{58.33163}{2} = 29.15 \text{ kg} \cdot \text{cm}$$

(19)

En la figura 10 se observa el motor a utilizar por cada lado del dispositivo.



*Figura 14. Motor brushless*



### 5.3.5. Sistema de alimentación

La batería de litio de 36V 4.4AH 158WH es ideal para alimentar el dispositivo por sus características y capacidades específicas. Equipada con un sistema de gestión de batería (BMS), esta batería tiene una capacidad de carga de 500 ciclos, esta batería ofrece una durabilidad excepcional.

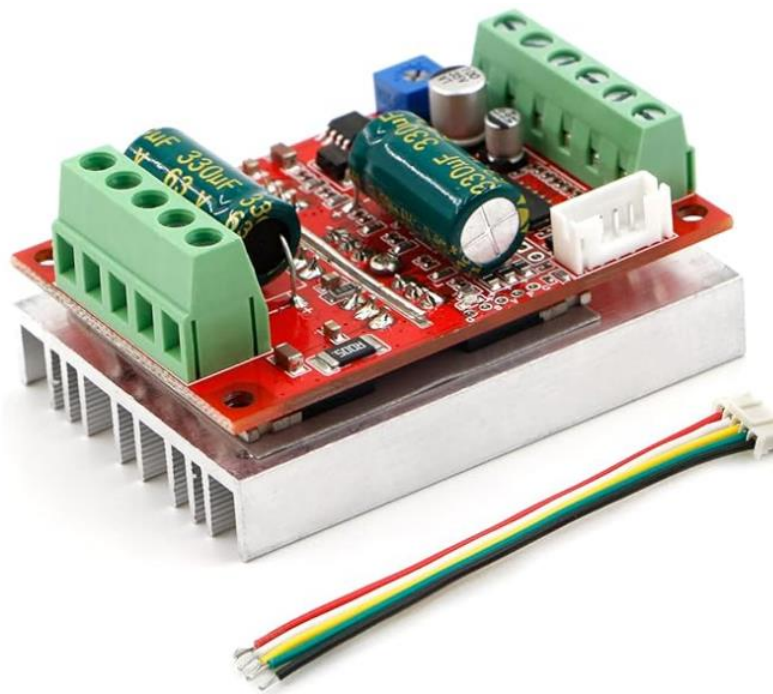


*Figura 16. Batería*

### 5.3.5. Sistema de Navegación y control

Para la parte de navegación y control se va a utilizar unos drivers especiales para el control de velocidad y posición para un motor brushless “RioRand 350W 6-60V PWM DC”, los cuales tienen un voltaje máximo de 60 Voltios y una potencia de 350 Vatios cada uno.

Estos drivers tienen como entrada a un potenciómetro externo que variara la velocidad en nuestro caso se utilizara un microcontrolador Arduino Mega, reemplazando el potenciómetro y variando este parámetro.



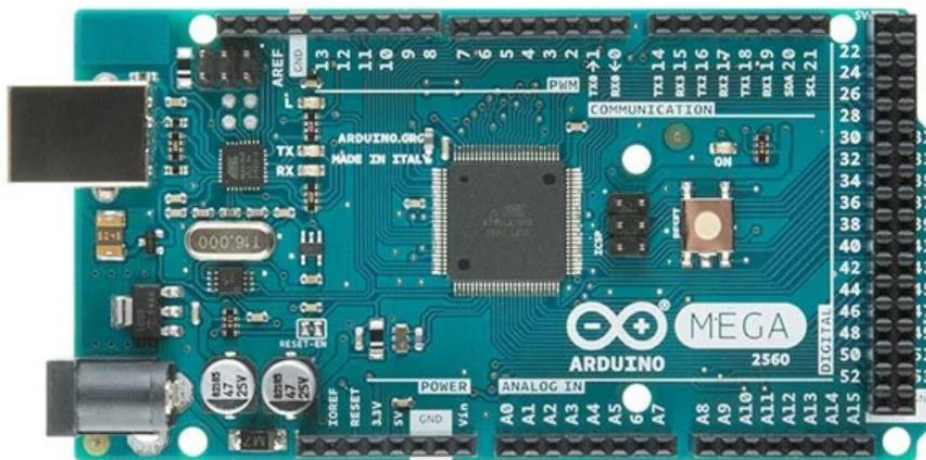
*Figura 17.*

Para que nos quemaran los drivers por el sobre picó de corriente se utilizó un regulador de voltaje DC Este nos permite tener un voltaje regulado a partir de una fuente de alimentación con un voltaje mayor.



*Figura 18. Regulador*

Se utilizó un microcontrolador, en este caso fue el Arduino mega, Tiene un total de 54 pines digitales para entradas y salidas. Para nuestro caso usamos un PWM para el control de velocidad mediante pulsos.



*Figura 19. Arduino MEGA*

Para la parte de navegación se realizó un mapeo por medio de la sensorica , se usaron los sensores sharp 2y0a21

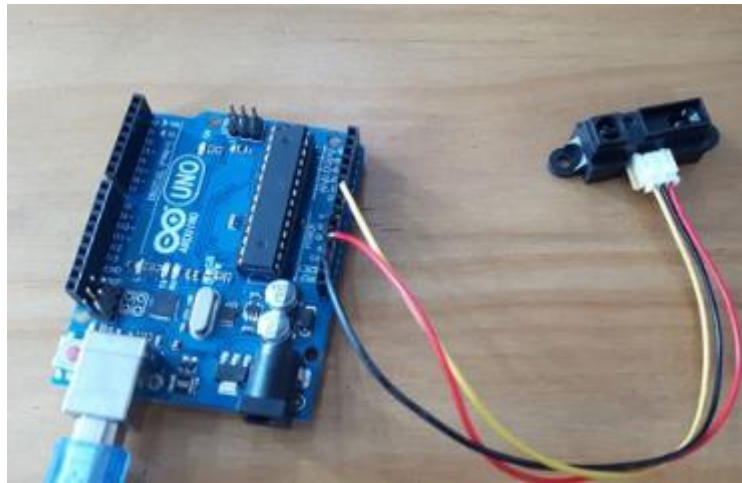


Figura 20. Sharp

es un sensor de distancia por infrarrojos (IR) que mide la distancia entre el sensor y un objeto mediante la reflexión de un haz de luz infrarroja. Implementamos este sensor para medir la distancia de las paredes para que, a cierta distancia, los drivers hagan una comunicación para parar el motor derecho o izquierdo para que haga los giros y siga su trayectoria. Por otro lado, para la navegación del robot se utilizó la esp32-cam para la detección de personas y obstáculos.

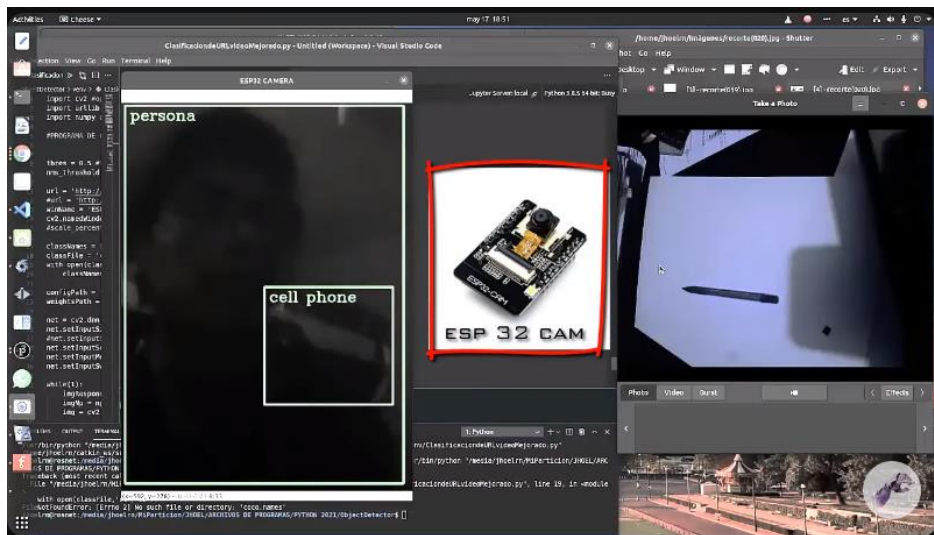


Figura 21. interfaz

El fin del proceso es para mapear el lugar donde el robot oruga estará ubicado, para ello lo primero que se realizó fue capturar imágenes de su entorno y crear un mapa, evitar obstáculos y planificar trayectorias.

Luego con Inteligencia artificial se realizó una red convolucional para el procesamiento de imágenes, se tomó una base de datos de todas las capturas realizadas de la esp32-cam para así crear un mapa, en donde lo más importante fue detectar características clave en las imágenes, como esquinas, líneas y formas. La red convolucional que se realizó se utilizó para este propósito.

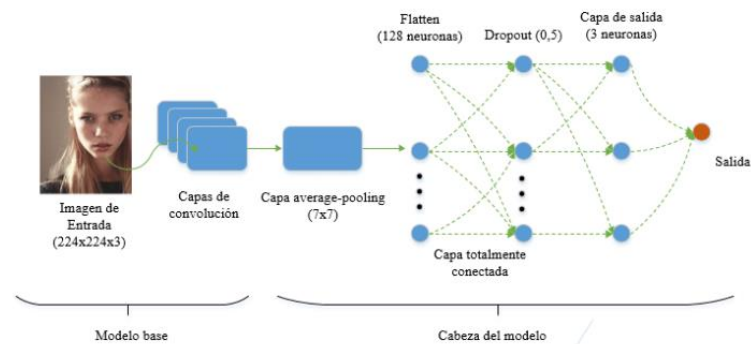
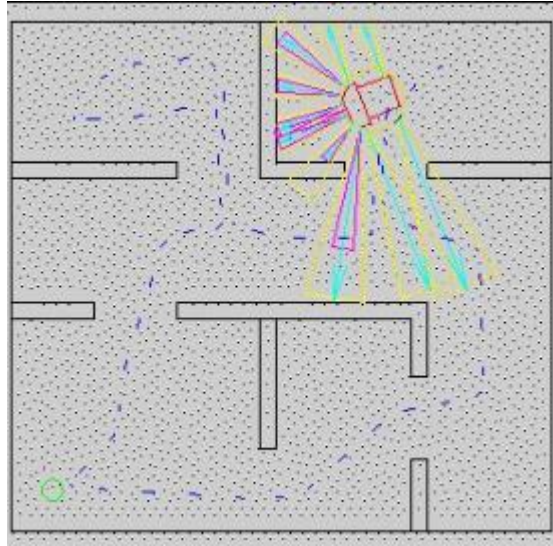


Figura 22. Modelo red neuronal

Las CNNs se componen de múltiples capas que incluyen convoluciones, pooling (submuestreo) y capas totalmente conectadas. Las capas de convolución aplican filtros para extraer características como bordes, texturas y formas de las imágenes, conservando las relaciones espaciales entre píxeles. Las capas de pooling reducen la dimensionalidad de las características, lo que disminuye la carga computacional y ayuda a prevenir el sobreajuste. Finalmente, las capas totalmente conectadas utilizan las características extraídas para clasificar la imagen o detectar objetos específicos. Gracias a su capacidad para aprender y reconocer patrones visuales de manera jerárquica, las CNNs son ampliamente utilizadas en tareas como la clasificación de imágenes, el reconocimiento facial y la detección de objetos.

Luego de obtener esta detección de imágenes gracias a la Inteligencia Artificial, se realiza la construcción del mapeo con Python, ya que, con las características detectadas, el robot puede comenzar a construir un mapa del entorno. Esto puede ser

representado como una cuadrícula (mapa de ocupación) donde las celdas se marcan como ocupadas o libres en función de la detección de obstáculos.



*Figura 23. Pruebas de trayectorias*

Luego con el control de los motores, se procede a hacer las trayectorias, implementando algoritmos de búsqueda como A\*, y también se hizo como tiempos del pwm, es decir que para los giros y la velocidad en la que sube el robot, se determinan ciertos timers de pwm, para así hacer el control de giros y velocidad del robot, y llegar a su destino.

## **5.4. Implementación y Pruebas**

### **5.4.1. Prototipo y Fabricación**

Para el prototipo final se adecuaron varias partes más al diseño inicial, para lograr un trabajo eficiente tras las pruebas que se mencionaran en el inciso 5.4.2, el



robot al final con la implementación tiene un peso de 17.4 Kg en promedio. Con un peso de 1.8kg de prototipado rápido y con partes tanto en aluminio como en hierro.



*Figura 24 . Prototipo final*

#### **5.4.2. Pruebas de Funcionamiento**

Tras implementarlo todo lo anterior, se procedió a realizar las pruebas, donde se debió considerar dónde quedaría el centro de gravedad del robot, pues al intentar subir las escaleras, no había compensación y se iba para atrás, hacia donde estaba el peso de los motores. Así que se le instalaron unas piezas en hierro para nivelar el peso de adelante y que no sucediera lo que había pasado anteriormente.



*Figura 25. Barra de hierro*

Para este inconveniente también se re agregaron a la parte trasera unas ruedas, para que hicieran el trabajo de una “Cola de pato”, así asegurar que al subir las escaleras y con ayuda del peso de las barras de hierro el robot no se fuera para atrás y simplemente con la ayuda de la fuerza de los motores lograra su objetivo.



*Figura 26. Cola de pato*

Luego se hicieron las pruebas pertinentes y el robot subió un pequeño peldaño con 5 escaleras, pero al probarlo en las escaleras más largas de la universidad, vimos que había que aplicarle más voltaje para que los motores dieran más fuerza.

Luego para dar el giro cuando fuera necesario por medio del microcontrolador según a qué lado se quisiera girar se detenía un motor y se hacía trabajar el otro, así, si quería que girara a la izquierda se detenía el motor izquierdo y se trabaja



con el derecho y si se quería girar a la derecha se hacía la acción opuesta. Para esto se utilizó un código en el Arduino mega que está en anexos 4.

## **5.5. Resultados y discusiones**

Durante el desarrollo y las pruebas del robot oruga diseñado, se realizaron experimentos para evaluar su rendimiento y capacidad de superar estos obstáculos. A continuación, se presentan los resultados obtenidos.

1.

- Objetivo: Evaluar la capacidad del robot para subir escaleras con peldaños de 20 cm de altura y 20 cm de profundidad.
- Resultado: El robot logró subir las escaleras con éxito en el 80% de los intentos. En el 20% restante, el robot experimentó una falla en su centro de gravedad.

2.

- Objetivo: Medir el tiempo que tarda el robot en subir un tramo de escaleras
- Resultado: El tiempo promedio fue de 45 segundos por escalera, (se realizaron varias pruebas)

3.

- Objetivo: Determinar la capacidad del robot para transportar el portátil
- Resultado: El robot pudo transportar hasta 5 kg adicionales sin pérdida significativa de rendimiento, es decir puedo cargar el pc y hasta 3 kg mas.

El robot puede subir escaleras y transportar cargas adicionales, con un rendimiento consistente y robusto. Sin embargo, los problemas relacionados con la estabilidad y la necesidad de optimizar el tiempo de ascenso indican áreas claras para futuras mejoras. En particular, trabajar en la estabilización del centro de gravedad y mejorar la eficiencia del sistema de control podrían aumentar significativamente la fiabilidad y velocidad del robot. Con estos ajustes, el robot tendría un potencial aún mayor para aplicaciones prácticas en entornos desafiantes y exigentes.

## Conclusiones

- Fue necesario realizar pruebas de validación del diseño y construcción del dispositivo, enfocándose en su capacidad para soportar diversas condiciones de uso. Puesto que estas pruebas nos ayudaron a asegurar que el dispositivo y su estructura era adecuada para cumplir su objetivo que es básicamente ascender de un piso a otro por medio de escaleras, para asegurarnos fue necesario someter al dispositivo a escenarios que simulaban el desgaste y la tensión que podría experimentar en su uso cotidiano.
- Las pruebas en distintos escenarios nos ayudaron a confirmar que el dispositivo podía operar de manera autónoma y eficiente. Esto incluyó evaluar su capacidad para realizar el recorrido de manera satisfactoria sin intervención humana, para esto se enfocó el uso de la parte sensorica y la implemetacion de inteligencia artificial para las trayectorias.
- Para asegurar que el dispositivo cumpliría la ruta propuesta en el tiempo especificado, fue necesario medir matemáticamente las revoluciones por segundo que alcanzaban los motores sin llegar a un sobre pico de corriente para que los drivers soportaran. También se consideró todos los factores que podrían influir en el tiempo de recorrido como la gente y obstáculos que se pudieran presentar a la hora de realizar el recorrido.
- Garantizar la integridad y seguridad de las personas que interactúen con el dispositivo es fundamental, para ello también se requirió un sistema de alerta y también se utilizó material aislante en su implantación total.

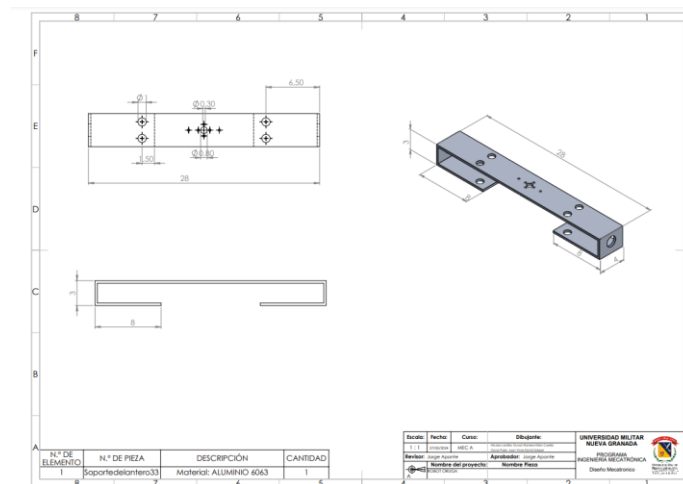
## Bibliografía

- (1) Kim, H., & Gat, E. (2008). 3-D Locomotion Algorithms for a Hexapod Walking Robot. *IEEE Transactions on Robotics*, 24(4), 855-864.

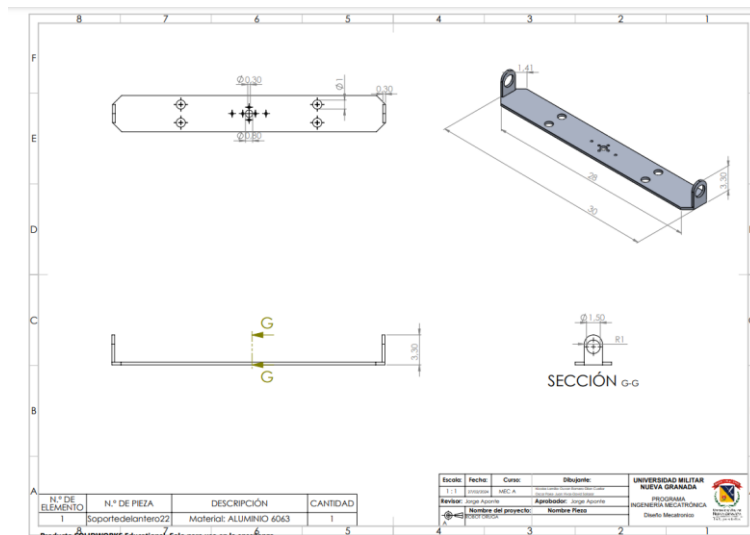
- (2) Raibert, M. H., & Hodgins, J. K. (1991). Dynamic control of a bipedal and quadrupedal robot. *ASME Journal of Dynamic Systems, Measurement, and Control*, 113(3), 378-385.
- (3) Hockman, B. R., & Hodges, S. R. (2010). Performance of a mobility system for a stair climbing wheelchair. *Journal of Rehabilitation Research & Development*, 47(7), 669-682.
- (4) Sun, W., Liu, Z., & Cheng, X. (2016). Design and Control of a Climbing Robot with a Compliant Mechanism. *Journal of Bionic Engineering*, 13(2), 229-240.
- (5) Hirose, S. (2006). Biologically Inspired Robots: Snake-Like Locomotors and Manipulators. *Oxford University Press*.
- (6) Hong, S., Lee, D., & Lee, J. (2018). Design and Implementation of a Quadruped Robot with Reconfigurable Body for Adaptability to Various Working Environments. *Sensors*, 18(11), 3679.

## Anexo 1

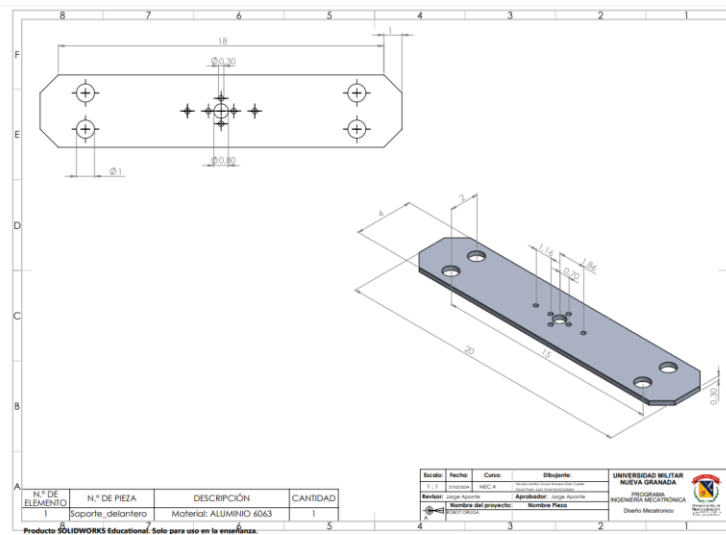
### Planos del diseño y partes en físico



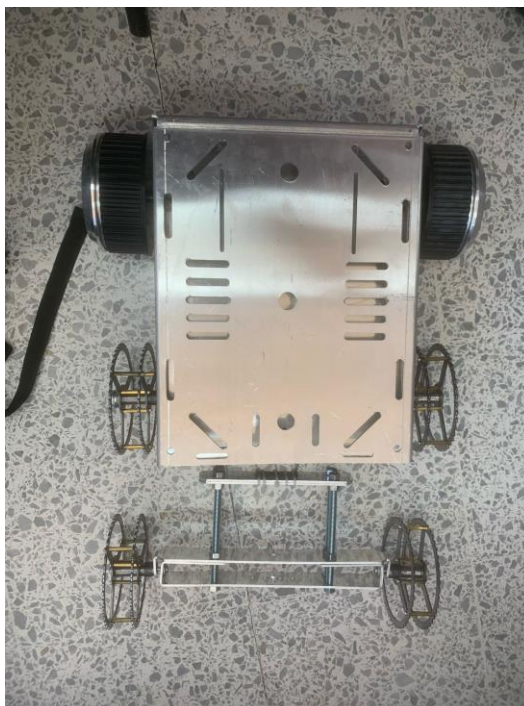
Plano Soporte Delantero33



Plano Soporte Delantero22



Plano Soporte Delantero



## Anexo 2

## Gastos indirectos

Concepto de Costo Indirecto	Costo (en pesos colombianos) por semana	Total (para 15 semanas para 6 personas)
Alquiler de Espacio	0 (utilizando instalaciones universitarias)	0
Servicios Públicos	10.000 (electricidad en casa)	150.000
Equipos de estudio	0 (proporcionado por la universidad)	0
Gastos de Comunicación	5.000(teléfono, internet)	75.000
Capacitación de Personal	0 (puede ser proporcionada por la universidad)	0
Transporte	30.000(equipos de la universidad)	450000
Otros Gastos Generales	10000(gastos diversos)	150.000
Total de Costos Indirectos	-	825000

## Gastos directos

Gastos directos							
Concepto	duvan	David	Nicolas	Juan	Dilan	Oscar	Total
chasis	\$ 130.000,00	\$ 100.000,00	\$ 50.000,00	\$ 200.000,00	\$ 100.000,00	\$ 100.000,00	\$ 700.000,00
platineta	\$ 50.000,00	\$ 100.000,00	\$ 50.000,00	\$ 50.000,00		\$ 50.000,00	\$ 300.000,00
soportes servo			\$ 30.000,00				\$ 30.000,00
servo		\$ 9.000,00		\$ 20.000,00			\$ 29.000,00
tapa rueda					\$ 110.000,00		\$ 110.000,00
varilla HR 3/4" x 2 tramos 5cm acero 1020				\$ 59.000,00			\$ 59.000,00
controladores			\$ 277.000,00				\$ 277.000,00
correas	\$ 35.500,00	\$ 35.500,00	\$ 35.500,00	\$ 35.500,00	\$ 59.000,00	\$ 35.500,00	\$ 236.500,00
tornillos torniquetes		\$ 51.000,00					\$ 51.000,00
drivers, soporte servo y cables	\$ 100.000,00				\$ 154.000,00	\$ 50.000,00	\$ 304.000,00
sensores			\$ 100.000,00	\$ 54.000,00		\$ 60.000,00	\$ 214.000,00
rodamientos e impresiones 3D guías		\$ 170.000,00					\$ 170.000,00
correa	\$ 15.000,00						\$ 15.000,00
regulador y arduino			\$ 30.000,00	\$ 80.000,00			\$ 110.000,00
REDUCTOR VOLTAJE MEDIDOR VOLTAJE	\$ 70.000,00			\$ 7.000,00		\$ 35.000,00	\$ 97.000,00
biarra y disco			\$ 30.000,00			\$ 29.000,00	\$ 59.000,00
sensorica y control			\$ 35.000,00		\$ 70.000,00	\$ 100.000,00	\$ 205.000,00
código de correa		\$ 25.000,00					\$ 25.000,00
biarra	\$ 5.000,00				\$ 12.000,00		\$ 17.000,00
ruedas 3d		\$ 70.000,00				\$ 70.000,00	\$ 140.000,00
guías Aluminio		\$ 35.000,00					\$ 35.000,00
Subtotal	\$ 425.500,00	\$ 595.500,00	\$ 607.500,00	\$ 535.500,00	\$ 505.000,00	\$ 514.500,00	\$ 3.183.500,00

## Anexo 3

### Codigo para control de velocidad

// Definir el pin PWM al que está conectado el ESC

const int escPin = 9; // Cambia este número al pin que hayas utilizado

```
void setup() {
```

```
    // Configurar el pin como salida
```

```
    pinMode(escPin, OUTPUT);
```

```

// Inicializar la señal PWM a un valor neutro
analogWrite(escPin, 0);
}

void loop() {
    // Simulación de incremento de velocidad
    for (int speed = 0; speed <= 255; speed += 5) {
        analogWrite(escPin, speed);
        delay(100); // Esperar un tiempo para ver el cambio de velocidad
    }

    // Simulación de decremento de velocidad
    for (int speed = 255; speed >= 0; speed -= 5) {
        analogWrite(escPin, speed);
        delay(100); // Esperar un tiempo para ver el cambio de velocidad
    }
}

```

### **Código para giro del robot**

```
// Pines del motor izquierdo
const int motorIzqAdelante = 5;
const int motorIzqAtras = 6;
const int motorIzqDir1 = 2;
const int motorIzqDir2 = 3;

// Pines del motor derecho
const int motorDerAdelante = 9;
const int motorDerAtras = 10;
const int motorDerDir1 = 7;
const int motorDerDir2 = 8;

// Velocidad de los motores (0-255)
const int velocidad = 150;

void setup() {
    // Configurar los pines de los motores como salidas
    pinMode(motorIzqAdelante, OUTPUT);
    pinMode(motorIzqAtras, OUTPUT);
    pinMode(motorIzqDir1, OUTPUT);
    pinMode(motorIzqDir2, OUTPUT);
    pinMode(motorDerAdelante, OUTPUT);
    pinMode(motorDerAtras, OUTPUT);
    pinMode(motorDerDir1, OUTPUT);
    pinMode(motorDerDir2, OUTPUT);
}
```



```

// Detener ambos motores al inicio
detenerMotores();
}

void loop() {
    // Ejemplo de movimientos
    avanzar();
    delay(2000);
    girarDerecha();
    delay(1000);
    avanzar();
    delay(2000);
    girarIzquierda();
    delay(1000);
    detenerMotores();
    delay(2000);
}

// Funciones de control del robot

void avanzar() {
    // Motor izquierdo hacia adelante
    digitalWrite(motorIzqDir1, HIGH);
    digitalWrite(motorIzqDir2, LOW);
    analogWrite(motorIzqAdelante, velocidad);
}

```

```

    analogWrite(motorIzqAtras, 0);

    // Motor derecho hacia adelante
    digitalWrite(motorDerDir1, HIGH);
    digitalWrite(motorDerDir2, LOW);
    analogWrite(motorDerAdelante, velocidad);
    analogWrite(motorDerAtras, 0);
}

void girarDerecha() {
    // Detener motor derecho
    analogWrite(motorDerAdelante, 0);
    analogWrite(motorDerAtras, 0);

    // Motor izquierdo hacia adelante
    digitalWrite(motorIzqDir1, HIGH);
    digitalWrite(motorIzqDir2, LOW);
    analogWrite(motorIzqAdelante, velocidad);
    analogWrite(motorIzqAtras, 0);
}

void girarIzquierda() {
    // Detener motor izquierdo
    analogWrite(motorIzqAdelante, 0);
    analogWrite(motorIzqAtras, 0);
}

```

```

// Motor derecho hacia adelante
digitalWrite(motorDerDir1, HIGH);
digitalWrite(motorDerDir2, LOW);
analogWrite(motorDerAdelante, velocidad);
analogWrite(motorDerAtras, 0);
}

```

```

void detenerMotores() {
    analogWrite(motorIzqAdelante, 0);
    analogWrite(motorIzqAtras, 0);
    analogWrite(motorDerAdelante, 0);
    analogWrite(motorDerAtras, 0);
}

```

### **Código para detección de objetos con esp32 cam**

```

#include "esp_camera.h"
#include <WiFi.h>
#include "esp_http_server.h"
#include "dl_lib.h"
#include "fr_forward.h"
#include "fr_flash.h"
#include "fd_forward.h"

#define CAMERA_MODEL_AI_THINKER

#include "camera_pins.h"

```

```

const char* ssid = "YOUR_SSID";
const char* password = "YOUR_PASSWORD";

// Crear un servidor HTTP
httpd_handle_t camera_httpd = NULL;

// Configuración de la cámara
void startCameraServer() {
    httpd_config_t config = HTTPD_DEFAULT_CONFIG();

    httpd_uri_t index_uri = {
        .uri      = "/",
        .method    = HTTP_GET,
        .handler    = [](httpd_req_t *req) -> esp_err_t {
            httpd_resp_set_type(req, "text/html");
            httpd_resp_send(req, (const char*) "Hola, ESP32-CAM está funcionando",
                HTTPD_RESP_USE_STRLEN);
            return ESP_OK;
        },
        .user_ctx  = NULL
    };

    if (httpd_start(&camera_httpd, &config) == ESP_OK) {
        httpd_register_uri_handler(camera_httpd, &index_uri);
    }
}

```

```

void setup() {
    Serial.begin(115200);
    Serial.setDebugOutput(true);
    Serial.println();

    camera_config_t config;
    config.ledc_channel = LEDC_CHANNEL_0;
    config.ledc_timer = LEDC_TIMER_0;
    config.pin_d0 = Y2_GPIO_NUM;
    config.pin_d1 = Y3_GPIO_NUM;
    config.pin_d2 = Y4_GPIO_NUM;
    config.pin_d3 = Y5_GPIO_NUM;
    config.pin_d4 = Y6_GPIO_NUM;
    config.pin_d5 = Y7_GPIO_NUM;
    config.pin_d6 = Y8_GPIO_NUM;
    config.pin_d7 = Y9_GPIO_NUM;
    config.pin_xclk = XCLK_GPIO_NUM;
    config.pin_pclk = PCLK_GPIO_NUM;
    config.pin_vsync = VSYNC_GPIO_NUM;
    config.pin_href = HREF_GPIO_NUM;
    config.pin_sscb_sda = SIOD_GPIO_NUM;
    config.pin_sscb_scl = SIOC_GPIO_NUM;
    config.pin_pwdn = PWDN_GPIO_NUM;
    config.pin_reset = RESET_GPIO_NUM;
    config.pin_xclk = XCLK_GPIO_NUM;

```

```

config.xclk_freq_hz = 200000000;
config.pixel_format = PIXFORMAT_JPEG;

if(psramFound()){
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10;
    config.fb_count = 2;
} else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12;
    config.fb_count = 1;
}

// Inicializar la cámara
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
    Serial.printf("Error inicializando la cámara: 0x%x", err);
    return;
}

// Conectar a la red WiFi
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}

```

```
Serial.println("");  
Serial.println("Conectado a WiFi");  
Serial.print("Dirección IP: ");  
Serial.println(WiFi.localIP());  
  
startCameraServer();  
}  
  
void loop() {  
    // Captura de imágenes y procesamiento aquí  
}
```