

**FACULTAD DE INGENIERÍA**

Escuela Académico Profesional de Ingeniería Mecatrónica

Trabajo de Investigación

**Diseño de un robot móvil teleoperado a distancia  
para rastreo de personas en un derrumbe  
de una edificación**

Harvith Brayan Echevarría Zenayuca  
Lenin Kepler Guerra Galarza  
Ricky Bryan Quiñones Fabián

Para optar el Grado Académico de  
Bachiller en Ingeniería Mecatrónica

Huancayo, 2020

Repositorio Institucional Continental  
Trabajo de investigación



Esta obra está bajo una Licencia "Creative Commons Atribución 4.0 Internacional" .

## **AGRADECIMIENTO**

En primer lugar, agradecemos a Dios y a nuestros padres que siempre estuvieron ahí para darnos palabras de apoyo desde un inicio y también por brindarnos la oportunidad de obtener una vida profesional.

De igual manera agradecemos a nuestros docentes de la Universidad Continental por brindarnos una enseñanza integral que ayudaron a mejorar nuestros conocimientos y por fomentar en nosotros valores que contribuyeron en nuestra formación personal y profesional.

Por último, agradecemos a todos nuestros compañeros y familiares que gracias a ellos se hizo posible concluir nuestra investigación.

## **DEDICATORIA**

Nuestro presente trabajo va dedicado a todas las personas que nos brindaron su apoyo incondicional, gracias a ellos se hizo posible culminar de manera satisfactoria esta investigación.

## ÍNDICE GENERAL

AGRADECIMIENTO .....	I
DEDICATORIA .....	II
ÍNDICE GENERAL .....	III
ÍNDICE DE FIGURAS.....	VI
ÍNDICE DE TABLAS .....	VIII
RESUMEN .....	IX
ABSTRACT.....	X
INTRODUCCIÓN .....	XI
CAPÍTULO I PLANTEAMIENTO DEL PROBLEMA .....	1
1.1 Planteamiento y formulación del problema .....	1
1.2 Formulación del problema .....	4
1.2.1 Problema general .....	4
1.2.2 Problemas específicos.....	4
1.3 Objetivos de la investigación .....	4
1.3.1 Objetivos general .....	4
1.3.2 Objetivos específicos .....	5
1.4 Justificación.....	5
1.4.1 Justificación teórica .....	5
1.4.2 Justificación practica .....	5
1.4.3 Justificación metodológica .....	5
CAPITULO II MARCO TEORICO .....	6
2.1 Antecedentes de la investigación .....	6
2.1.1 Internacionales.....	6
2.1.2 Nacionales .....	7
2.2 Bases teóricas .....	8
2.2.1 Robot .....	8
2.2.2 Robots móviles .....	8
2.2.3 Tipos de robots móviles según su sistema de movimiento.....	9
2.2.4 Arquitectura de un robot móvil .....	11

2.2.5 Teleoperación .....	11
2.2.6 Componentes en un sistema de teleoperación .....	11
2.2.7 Sistemas de navegación .....	12
2.2.8 Detección de obstáculos .....	18
2.2.9 Tecnologías de reconocimiento y supervisión.....	19
2.2.10 Cámara termográfica AMG8833 .....	20
2.2.11 Fuentes de energía .....	24
2.2.12 Motores.....	25
2.2.13 Drivers de motores.....	26
2.2.14 Programación.....	27
2.2.15 Sistema de comunicación inalámbrica .....	28
2.2.16 Módulo NRF24L01 .....	28
2.2.17 Origen de los derrumbes.....	28
2.3 Definición de términos básicos .....	30
<b>CAPITULO III METODOLOGIA .....</b>	<b>31</b>
3.1 Diseño metodológico.....	31
3.1.1 Tipo de investigación.....	31
3.1.2 Alcance de investigación .....	31
3.1.3 Metodología aplicada para el desarrollo de solución .....	32
<b>CAPITULO IV ANALISIS Y DISEÑO DE LA SOLUCIÓN.....</b>	<b>36</b>
4.1 Estado del arte .....	36
4.1.1 Carro tractor RX-130.....	36
4.1.2 Ixnamiki Olinki.....	37
4.2 Lista de exigencias .....	37
4.3 Black-Box.....	39
4.4 Caja blanca .....	41
4.5 Matriz morfológica.....	42
4.6 Evaluación técnica – económica del concepto de solución.....	46
4.6.1 Valoración técnica .....	46
4.6.2 Valoración económica .....	46
4.6.3 Selección de sistema optimo.....	47
4.7 Diseño mecánico .....	48

4.7.1 Sistema de movimiento .....	48
4.7.2 Estructura.....	49
4.7.3 Motores.....	50
4.8 Diseño electrónico.....	51
4.8.1 Controlador.....	51
4.8.2 Control de motores .....	53
4.8.3 Selección de sistema de comunicación inalámbrica.....	54
CAPITULO V SIMULACIÓN.....	55
5.1 Simulación de procesos.....	55
5.2 Resultados .....	63
5.2.1 Placa de control en PCB .....	63
5.2.2 Placa de mando en PCB.....	64
5.2.3 Modelamiento de prototipo final .....	65
CONCLUSIONES .....	66
TRABAJOS FUTUROS .....	67
REFERENCIAS BIBLIOGRÁFICAS.....	68
ANEXOS .....	70

## ÍNDICE DE FIGURAS

Figura 1. Región Loreto 23 de diciembre del 2019 7:02 pm .....	1
Figura 2. Ixnamiki Olinki .....	3
Figura 3. Interacción robot-entorno. ....	9
Figura 4. Robot móvil con ruedas.....	10
Figura 5. Robot móvil con patas .....	10
Figura 6. Robot TRI-TRACK S300167.....	11
Figura 7. Arquitectura de robot móvil .....	11
Figura 8. Componentes en un sistema de teleoperación .....	12
Figura 9. Diagrama de conexión modulo GY-GPS6MV2.....	13
Figura 10. Configuración GPS.....	14
Figura 11. Configuración monitor serial.....	15
Figura 12. Programación GPS-1 .....	16
Figura 13. Programación GPS-2 .....	16
Figura 14. Configuración GPS-3 .....	17
Figura 15. Datos codificados del módulo GY – GPS6MV2.....	17
Figura 16. Diagrama de conexión sensor ultrasónico HC–SR04 .....	18
Figura 17. Configuración HC–SR04.....	19
Figura 18. Datos del sensor ultrasónico HC–SR04 .....	19
Figura 19. Cámara termográfica AMG8833.....	21
Figura 20. Pines de cámara AMG8833.....	21
Figura 21. Configuración AMG8833.....	22
Figura 22. Test de prueba Cámara termográfica AMG8833 .....	22
Figura 23. Programación AMG8833 .....	23
Figura 24. Programación AMG8833 .....	23
Figura 25. Datos en array de cámara termográfica AMG8833.....	24
Figura 26. Motor DC.....	25
Figura 27. Driver L298N .....	26
Figura 28. Abastecimiento del L298N.....	27
Figura 29. NRF24L01 .....	28
Figura 30. Modelo de VDI 2206.....	32

Figura 31. Caja negra.....	33
Figura 32. Caja blanca .....	34
Figura 33. Matriz morfológica.....	34
Figura 34. Carro tractor RX-130.....	36
Figura 35. Ixnamiki Olinki .....	37
Figura 36. Caja blanca .....	41
Figura 37. Grafica de evaluación técnica – económica .....	47
Figura 38. Concepto de solución .....	48
Figura 39. Controlador en placa de control .....	51
Figura 40. Controlador en placa de mando.....	52
Figura 41. Circuito de potencia de motor .....	53
Figura 42. Inicialización del robot.....	55
Figura 43. Verificación de conexión.....	56
Figura 44. Conexión establecida entre el robot oruga y la estación de mando.....	56
Figura 45. Error de conexión .....	57
Figura 46. Verificando sensor ultrasónico.....	57
Figura 47. Correcto estado del sensor ultrasónico .....	58
Figura 48. Revisar sensor ultrasónico.....	58
Figura 49. Verificación de GPS.....	59
Figura 50. GPS en línea .....	59
Figura 51. GPS desconectado .....	60
Figura 52. Longitud obtenida de GPS.....	60
Figura 53. Latitud obtenida de GPS.....	61
Figura 54. Detección de obstáculo mediante sensor ultrasónico. ....	61
Figura 55. Mensaje de alerta objetivo a 30cm. ....	62
Figura 56. Mensaje de alerta objetivo a 15cm. ....	62
Figura 57. Placa de control PCB.....	63
Figura 58. Placa de mando PCB .....	64
Figura 59. Renderizado final del prototipo .....	65

## ÍNDICE DE TABLAS

Tabla 1. Población afectada en el terremoto del 2007 .....	3
Tabla 2. Diferencias entre baterías NiMH y Li-Po .....	25
Tabla 3. Población y viviendas afectadas ante la ocurrencia de eventos hidrológicos.....	29
Tabla 4. Formato de lista de exigencias.....	33
Tabla 5. Lista de exigencias.....	37
Tabla 6. Matriz de funciones .....	42
Tabla 7. Valoración técnica .....	46
Tabla 8. Valoración económica .....	46
Tabla 9. Sistema de movimiento por ruedas .....	49
Tabla 10. Sistema de movimiento oruga.....	49
Tabla 11. Sistema de movimiento por patas .....	49
Tabla 12. Tipos de materiales .....	50
Tabla 13. Motor paso a paso .....	50
Tabla 14. Motor DC con caja reductora.....	51
Tabla 15. Servomotor.....	51
Tabla 16. Conexiones ATMEGA328P .....	52
Tabla 17. Funciones de circuito de potencia.....	53
Tabla 18. Características de comunicación inalámbrica.....	54

## RESUMEN

Se desarrollo un robot móvil teleoperado a distancia para rastreo de personas de un derrumbe de una edificación. Utilizando la metodología VDI 2206, se desarrolló los dominios mecánico, electrónico y control del robot. En el dominio mecánico se determinó el uso del sistema de movimiento de tipo oruga por su eficiencia de movimiento en terrenos accidentados y alta tracción, además una estructura de aluminio que pueda resistir los posibles impactos del derrumbe al momento de uso del robot dentro de la edificación.

En el dominio electrónico se determinó el uso de dos placas: una de control donde se instalan el sensor ultrasónico, modulo GPS, antena RFD24L01 y los motores y la segunda placa de mando que cuenta con dos joysticks una pantalla de video, un LCD para la recepción de datos y la antena RFD24L01 y por último en el dominio de control se desarrolló el sistema de comunicación y de recepción y envió de datos desde el robot móvil hasta el operador.

**Palabras clave:** Robot móvil, teleoperado a distancia, VDI2206, rastreo de personas.

## **ABSTRACT**

A remote operated mobile robot was developed to track people from a building collapse. Using the VDI 2206 methodology, the mechanical, electronic and control domains of the robot were developed. In the mechanical domain, the use of the caterpillar-type movement system was determined due to its movement efficiency on uneven terrain and high traction, in addition to an aluminum structure that can resist the possible impacts of the collapse when the robot is used inside the building.

In the electronic domain, the use of two boards was determined: one control board where the ultrasonic sensor, GPS module, RFD24L01 antenna and motors are installed and the second control board that has two joysticks, a video screen, an LCD for data reception and the RFD24L01 antenna. Finally, in the control domain, the communication and data reception and sending system from the mobile robot to the operator was developed.

**Keywords:** Mobile robot, remote operation, VDI2206, people tracking.

## INTRODUCCIÓN

En los últimos años ocurrieron desastres naturales en las distintas zonas del territorio peruano, en el cual según el Instituto Geofísico del Perú reportó un total de 550 eventos sísmicos ocurridos en el año 2019 esto evidencia que en poco tiempo puede aparecer un terremoto de alta categoría dañando todos los edificios en Perú porque en su gran mayoría no tienen una construcción antisísmica.

Y por lo tanto ocasionarían más pérdidas humanas y desesperación, como es en el caso de la cantidad de personas fallecidas en el terremoto del 2007 que ascendió a un total de 596 y más de 434 mil personas damnificadas.

Basándonos en los índices que nos dejan los desastres naturales aparece la necesidad de investigar el uso de los robots teleoperados aplicados en el rescate de personas que puedan apoyar a los rescatistas a la localización de personas en un derrumbe de una edificación.

Como objetivos nos planteamos el diseño de un robot móvil teleoperado a distancia para rastreo de personas en un derrumbe de una edificación, encontrar el diseño adecuado para el sistema de movimiento del robot móvil, identificar la tecnología de comunicación inalámbrica que permita el enlace entre el robot y el operador y, por último, encontrar el método para identificar a las personas que se encuentren en un derrumbe de una edificación.

Esta investigación utilizara sensores ultrasónicos para esquivar obstáculos y mediante una cámara con visión térmica incluida que estarán conectados a un robot móvil guiado remotamente por radiofrecuencia de baja señal. Podrá ayudar a los bomberos en la localización de las personas y así tomar las decisiones correctas para salvaguardar sus vidas y la de las personas que están siendo rescatadas.

# CAPÍTULO I

## PLANTEAMIENTO DEL PROBLEMA

### 1.1 Planteamiento y formulación del problema

En los últimos años ocurrieron desastres naturales en las distintas zonas del territorio peruano, en el cual según el Instituto Geofísico del Perú reportó un total de 550 eventos sísmicos ocurridos en el año 2019, el más resaltante fue en la región Loreto el 23 de diciembre que ocasiono derrumbes en varias edificaciones como se observa en la figura 1, esto evidencia que en poco tiempo puede aparecer un terremoto de alta categoría dañando todos los edificios en Perú porque en su gran mayoría no tienen una construcción antisísmica.



Figura 1. Región Loreto 23 de diciembre del 2019 7:02 pm disponible en Informe de Sismo del Gobierno del Perú

Perú pertenece al cinturón de fuego del Pacífico y por eso tenemos un potencial sísmico elevado porque en esta zona la Tierra usualmente libera más del 85% de la energía que se acumula en su interior (1) . A consecuencia de estos sismos se produce derrumbes las cuales llegan a un desenlace fatal en el que la población no se encuentra debidamente preparada con un plan de evacuación posteriormente al sismo, peor aún el escaso apoyo del Estado Peruano hacia los organismos encargados del rescate de las personas (Bomberos, Rescatistas, etc.) y a los mismos afectados. De ahí que, por un desastre de intensidad mayor, se reporta personas atrapadas en el lugar ocurrido con un limitado tiempo de sobrevivencia y con recursos limitados, siendo muchos de ellos personas gravemente heridas

La población peruana no se encuentra preparada si ocurre un desastre de gran magnitud porque no hay cultura de prevención en los ciudadanos, tampoco hay planes de evacuación y principalmente por la gran cantidad de edificaciones informales que no cumplen con las normas de construcción según CAPECO el 80% de viviendas son construcciones informales y la mitad son vulnerables al colapso en un terremoto de alta intensidad. (2)

Estos motivos ocasionarían más pérdidas humanas y desesperación, como es en el caso de la cantidad de personas fallecidas en el terremoto del 2007 que se puede visualizar en la tabla 1, el miércoles 15 de agosto en la zona sur de la costa central de Perú ocurrió el mayor sismo registrado en estos 100 últimos años y el más catastrófico en la costa central en los últimos 290 años donde la cantidad de pérdidas humana fue inmensa (3).

Tabla 1. Población afectada en el terremoto del 2007

Zonas Afectadas	Población Damnificada a/	Población Afectada EDAN b/	Población Afectada a/+b/	Heridos	Muertos c/
<b>ICA</b>	<b>363 841</b>	<b>157 369</b>	<b>521 210</b>	<b>1 109</b>	<b>586</b>
Chincha	147 520	44 916	192 436	256	
Ica	155 660	60 501	216 161	149	
Pisco	59 971	50 522	110 493	701	
Palpa	690	1 430	2 120	3	
<b>LIMA</b>	<b>59 483</b>	<b>40 371</b>	<b>99 854</b>	<b>155</b>	<b>10</b>
Cañete	47 527	27 801	75 328	20	
Huachipaico	70	2 005	2 075	0	
Lima	636	175	811	128	
Yauyos	11 075	9 985	21 060	2	
Callao	175	405	580	5	
<b>HUANCAVELICA</b>	<b>10 810</b>	<b>20 870</b>	<b>31 680</b>	<b>4</b>	<b>0</b>
Castrovilleja	7 060	10 320	17 380	0	
Huancavelica	470	365	835	4	
Huaytará	3 280	10 185	13 465	0	
<b>AYACUCHO</b>	<b>460</b>	<b>2 450</b>	<b>2 910</b>	<b>0</b>	<b>0</b>
Cangallo	330	450	780	0	
Huamanga	100	250	350	0	
La Mar	0	5	5		
Lucanas	0	1 120	1 120	0	
Parinacochas	0	525	525	0	
Páucar del Sara Sara	30	50	80	0	
<b>JUNÍN</b>	<b>20</b>	<b>0</b>	<b>20</b>		
Huancayo	20	0	20	0	
<b>Total</b>	<b>434 614</b>	<b>221 060</b>	<b>655 674</b>	<b>1 268</b>	<b>596</b>

Disponible en Instituto Nacional de Defensa Civil

Basándonos en los índices que nos dejan los desastres naturales aparece la necesidad de investigar el uso de los robots tele operados aplicados en el rescate de personas encontrando un ejemplo claro de la aplicación en el robot llamado Ixnamiki Olinki que se observa en la figura 2, un dato importante es que ayudo a rescatar a 11 personas el año 2017 en la colonia del Valle que se encuentra en la ciudad de México. Este prototipo fue creado en la Universidad Panamericana que desde sus inicios el prototipo tenía el objetivo de apoyar en las labores de búsqueda (4).



Figura 2. Ixnamiki Olinki  
Disponible en el diario La Jornada

El uso de estos vehículos no tripulados en los últimos años se incrementó alrededor del mundo. Usualmente se usaba en un ámbito militar pero ahora tienen más aplicaciones incluso para un propósito recreativo. Según Peter Diamandis esto se debe por el desarrollo tecnológico y uso de módulos como el GPS, cámaras digitales, computadores y todo mediante una comunicación inalámbrica. Y por ellos estos vehículos se han estado aplicado en muchos campos como en minería, agricultura, construcción, entretenimiento, etc. (5).

Sin embargo, estos sistemas tienen una limitante y es que solo sirve para sistemas externos, pero al ingresar en un lugar los obstáculos pueden dañarlo si no se tiene una correcta manipulación. En cambio, un robot que se movilice por vía terrestre puede adaptarse para que pueda moverse en superficies de cualquier tipo utilizando un sistema de movimiento tipo oruga.

Por estas razones nos planteamos diseñar un robot móvil, el cual gracias a los sensores integrados permitirá el reconocimiento y rastreo de personas perdidas entre los escombros ya sean ocasionada por desastres naturales o derrumbes de edificaciones.

## **1.2 Formulación del problema**

### **1.2.1 Problema general**

¿Cómo realizar el diseño de un robot móvil teleoperado a distancia para rastreo de personas en un derrumbe de una edificación?

### **1.2.2 Problemas específicos**

- ✓ ¿Qué diseño es el adecuado para el sistema de movimiento del robot móvil?
- ✓ ¿Qué tecnología de comunicación inalámbrica permite el enlace entre el robot móvil y el operador?
- ✓ ¿Cómo se identificará a las personas que se encuentren en un derrumbe de una edificación?

## **1.3 Objetivos de la investigación**

### **1.3.1 Objetivos general**

- ✓ Realizar el diseño de un robot móvil teleoperado a distancia para rastreo de personas en un derrumbe de una edificación.

### **1.3.2 Objetivos específicos**

- ✓ Encontrar el diseño adecuado para el sistema de movimiento del robot móvil.
- ✓ Identificar la tecnología de comunicación inalámbrica que permita el enlace entre el robot móvil y el operador.
- ✓ Encontrar el método para identificar a las personas que se encuentren en un derrumbe de una edificación.

## **1.4 Justificación**

### **1.4.1 Justificación teórica**

Esta investigación aportara conocimiento en la localización de personas utilizando un sistema de reconocimiento mediante cámaras termografías cuyos resultados podrán ayudar a futuras investigaciones y proyectos en la búsqueda de personas en un derrumbe de una edificación.

### **1.4.2 Justificación practica**

Esta investigación utilizara sensores ultrasónicos para esquivar obstáculos y mediante una cámara con visión térmica incluida que estarán conectados a un robot móvil guiado remotamente por radiofrecuencia de baja señal. Podrá ayudar a los bomberos en la localización de las personas y así tomar las decisiones correctas para salvaguardar sus vidas y la de las personas que están siendo rescatadas.

### **1.4.3 Justificación metodológica**

Esta investigación utilizara una metodología de diseño electrónico mediante el software de simulación Proteus para la verificación y simulación del sistema de electrónico para un futuro análisis de mejoras mediante el análisis de resultados.

## **CAPITULO II**

### **MARCO TEORICO**

#### **2.1 Antecedentes de la investigación**

##### **2.1.1 Internacionales**

Desarrollo de una plataforma robótica móvil para la búsqueda posterior a un sismo, y mitigación de incendios de baja escala (6). La investigación consistió el desarrollo de un robot para buscar personas después de un sismo, además con un sistema de riego para combatir incendios de baja escala. Este modelo utiliza un sistema mecánico-electrónico y además de un sistema de control inalámbrico. Los puntos clave en el diseño del sistema mecánico fueron el diseño de tamaño reducido para poder ingresar en espacios pequeños y el sistema oruga para poder movilizarse en cualquier relieve. El monitoreo y búsqueda se realizó a través de cámaras con sensores ultrasónicos todo esto se encuentra implementado en el sistema electrónico.

Diseño e implementación del prototipo de un robot para exploración de edificaciones y localización de personas en casos de derrumbes (7). La investigación consistió en el diseño de un prototipo que pueda ayudar a los rescatistas en la búsqueda de personas en estructuras que han colapsado por derrumbes o sismos, en el diseño se utilizó un sistema de movimiento tipo oruga y el material del chasis fue aluminio por ser un material de bajo costo, resistente

y ligero. Para el sistema de control se utilizó la tarjeta Raspberry donde configuraron los servicios DHCP y además de la integración de un módulo WI-FI para que pueda utilizarse como un punto y así visualizar los datos desde un Smartphone. El robot cuando ingresa al lugar de rescate proporciona los datos de la situación de la edificación mediante los sensores y la distancia máxima para transmisión de datos fue hasta de 15 metros desde el lugar de control.

Diseño e implementación de un prototipo robótico móvil teleoperado inalámbricamente para inspección en lugares de alto riesgo (8). La investigación consiste en el diseño y prototipado de un robot móvil que pueda inspeccionar lugares peligrosos a la seguridad ciudadana. El robot tiene la capacidad de ser maniobrado inalámbricamente por medio de un control remoto y transmitir audio y video al mismo tiempo. La finalidad de esta investigación es la inspección y reconocimiento del entorno en sitios de posibles reuniones delictivas por medio del robot, mientras que los agentes del orden monitorizan y teledirigen al robot desde un lugar seguro; salvaguardando sus vidas. El alcance máximo de línea de vista es de 260 metros y en interiores es de 90 metros con una autonomía máxima de 50 minutos, lo que lo convierte en un robot espía a corta distancia.

### **2.1.2 Nacionales**

Robot móvil para detección de personas en lugares inaccesibles frente a desastres naturales en la ciudad Arequipa: caso sismos (9). La investigación consistió en el diseño de un robot que se pueda movilizar por vía terrestre y pueda revelar la ubicación de personas en lugares de difícil acceso en la capital de Arequipa, en el apartado electrónico cuenta con un sensor ultrasónico que le permite evadir obstáculos mientras es maniobrado por radiofrecuencia en adición a ello, cuenta con un módulo GPS que utiliza el microcontrolador para el envío de las coordenadas en tiempo real y para la detección usa una cámara con un sistema de reconocimiento.

Vehículo aéreo no tripulado para vigilancia en ambientes cerrados con detección de personas y obstáculos a su alrededor (10). La investigación consistió en el diseño de un vehículo aéreo que se utiliza en vigilancia y seguridad en ambientes.

La metodología de diseño usada fue la VDI221. Este diseño cuenta con una estructura que evita los daños por colisión en el vuelo. Para la vigilancia utiliza una cámara equipada con visión nocturna y para la transmisión de video utiliza una frecuencia baja para que la señal pueda traspasar paredes y pisos en un radio de 50 m<sup>2</sup> a la redonda.

Robot explorador con realidad virtual para la seguridad en colegios estatales (11). La investigación consistió en la implementación de un robot explorador con realidad virtual para el incremento de la seguridad. EL implementar un robot explorador, para la video vigilancia, con realidad virtual genera un sistema inmersivo entre hombre y máquina. Contar con un diseño mecánico tipo oruga facilito el desplazamiento del prototipo, además mediante el diseño electrónico se pudo hallar la demanda eléctrica para poder dimensionar las baterías para una autonomía mínima de 2 horas. A pesar del incremento del tiempo de retardo el manejo del desplazamiento del robot es aceptable, aunque puede ser reducido si se conecta a una red local inalámbrica que le permitiría al operario encontrarse a salvo y sin exponerse en el uso del robot.

## **2.2 Bases teóricas**

### **2.2.1 Robot**

Un robot es una maquina electrónica reprogramable con capacidad de ejecutar operaciones de forma autónoma con supervisión humana o siguiendo parámetros previamente definidos. (12)

### **2.2.2 Robots móviles**

Un robot móvil tiene la característica de desplazarse en distintos entornos gracias al sistema de locomoción que posee y completar tareas como: mapeo, exploración, búsqueda, etc.

Un robot móvil este compuesto por un sistema mecánico, sensores, actuadores y un sistema de control, que trabajan en un bucle cerrado de lectura y ejecución de comandos. (12)

El sistema de control procesa la información recolectada de su entorno mediante los sensores como se puede observar en la figura 3, con el objetivo de enviar ordenes predefinidas a los actuadores y volver a recolectar información.

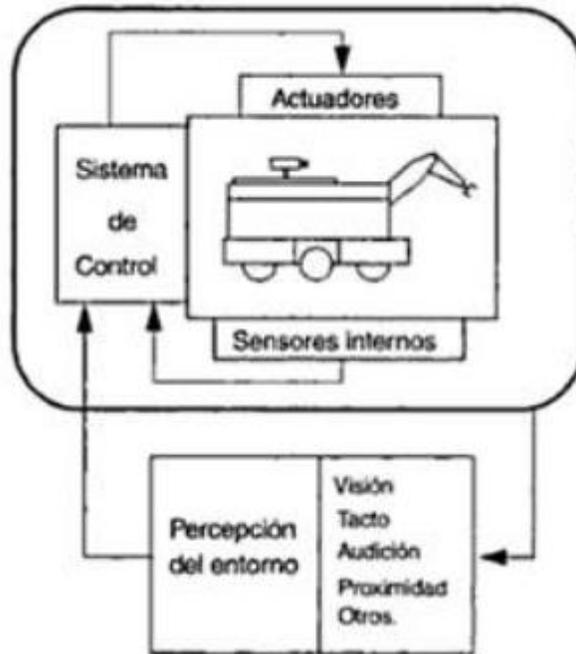


Figura 3. Interacción robot-entorno.  
disponible en “Robótica, manipuladores y robots móviles” por Ollero A. 2001, p.4

Los sensores internos se encargan de inspeccionar el estado de los actuadores y sistema electrónico y los sensores externos registran datos del entorno del robot para posteriormente interpretarlos y procesarlos en el controlador. (13)

### 2.2.3 Tipos de robots móviles según su sistema de movimiento

#### 2.2.3.1 Sistema de movimiento por ruedas

El sistema de movimiento por ruedas es el más utilizado por la facilidad de implementación y control. En terrenos regulares y libres de obstáculos, pueden alcanzar altas velocidades, pero carece de fuerza de tracción.



Figura 4. Robot móvil con ruedas  
disponible en: <https://bit.ly/3qoqTiG>

### 2.2.3.2 Sistema de movimiento por patas

El sistema de movimiento por patas está conformado por varios mecanismos que se usan también para el punto de anclaje con el territorio. Pueden desplazarse eficientemente en terrenos accidentados. De acuerdo al número de extremidades aumenta o se reduce el control del sistema de equilibrio



Figura 5. Robot móvil con patas  
disponible en: <https://bit.ly/2Vkw1G0>

### 2.2.3.3 Sistema de movimiento oruga

Sistema equipado con pistas de deslizamientos para el movimiento, tiene alta adaptabilidad sobre cualquier tipo de relieve y puede subir pendientes de determinado ángulo.

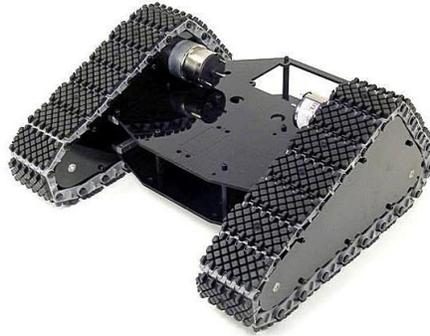


Figura 6. Robot TRI-TRACK S300167 disponible en Amazon Store

### 2.2.4 Arquitectura de un robot móvil

Es la descripción de las características de la estructura del sistema del robot móvil, sirven para detallar y representar el funcionamiento de todo el sistema.

(14)

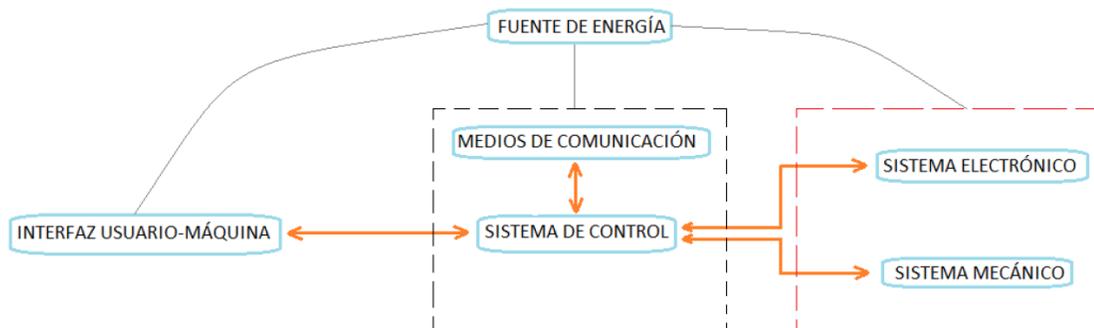


Figura 7. Arquitectura de robot móvil disponible en “Ingeniería de Control Moderna” 5ta edición de Katsuhiko O. 2010; p.124

### 2.2.5 Teleoperación

La teleoperación significa controlar un dispositivo a distancia desde una estación mediante un operario. En quipos avanzados se incluye el uso de transmisión de imágenes para que el operario pueda operar con mayor exactitud. Este sistema es altamente utilizado para exploración de lugares con dificultad de acceso o que representen un peligro para el ingreso de personas. (13)

### 2.2.6 Componentes en un sistema de teleoperación

Un sistema de teleoperación fue planteado para ayudar a los humanos a realizar ciertas tareas en entornos insalubres u hostiles como por ejemplo lugares radioactivos, lugares con gases tóxicos, zonas nucleares, etc.

Para poder operar en estas circunstancias los seres humanos han desarrollado robots que logren hacer operaciones específicas como: recolección de datos, desactivación de bombas, etc. En esos entornos protegiendo el bienestar del operador reduciendo los factores de riesgo.

Un sistema de teleoperación está conformado por 2 partes principales: El sistema maestro que es controlado por el operador desde un dispositivo de mando y el sistema esclavo que es el robot a controlar como se observa en la figura 8. (13)

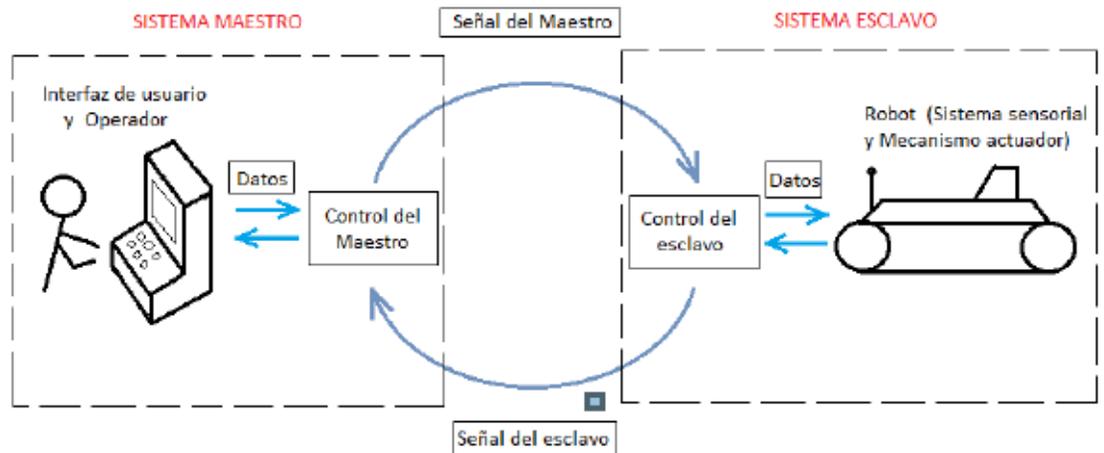


Figura 8. Componentes en un sistema de teleoperación disponible en “Robótica, manipuladores y robots móviles” por Ollero A. 2001, p.25

## 2.2.7 Sistemas de navegación

Un sistema de navegación permite conocer la posición exacta del robot desde el dispositivo de mando.

### 2.2.7.1 Sistema de posicionamiento global (GPS)

El GPS fue desarrollado por el Departamento de Defensa de los Estados Unidos de América para determinar la posición se utilizan de 4 a un número mayor de satélites mediante la trilateración. La precisión de los datos varía en metros del punto real, pero si se usa un GPS diferencial la precisión puede ser de centímetros.

### 2.2.7.2 Modulo GPS GY- GPS6MV2

Este módulo cuenta con una memoria EEPROM con una pila de botón para mantener los datos de la misma, además de un indicador LED y una antena. Para la conexión y configuración cuenta con pines de VCC, Tx, Rx y Gnd y todo esto se hace mediante un interfaz serial en contacto con el controlador principal.

### 2.2.7.3 Conexión modulo GY- GPS6MV2

Las conexiones van de la siguiente manera: el pin 3.3V se conecta a una fuente VCC del controlador, el pin GND se conecta al pin GND del controlador el pin Tx y Rx se conecta a cualquier pin digital de controlador, una mejor explicación se puede visualizar en la figura 9.

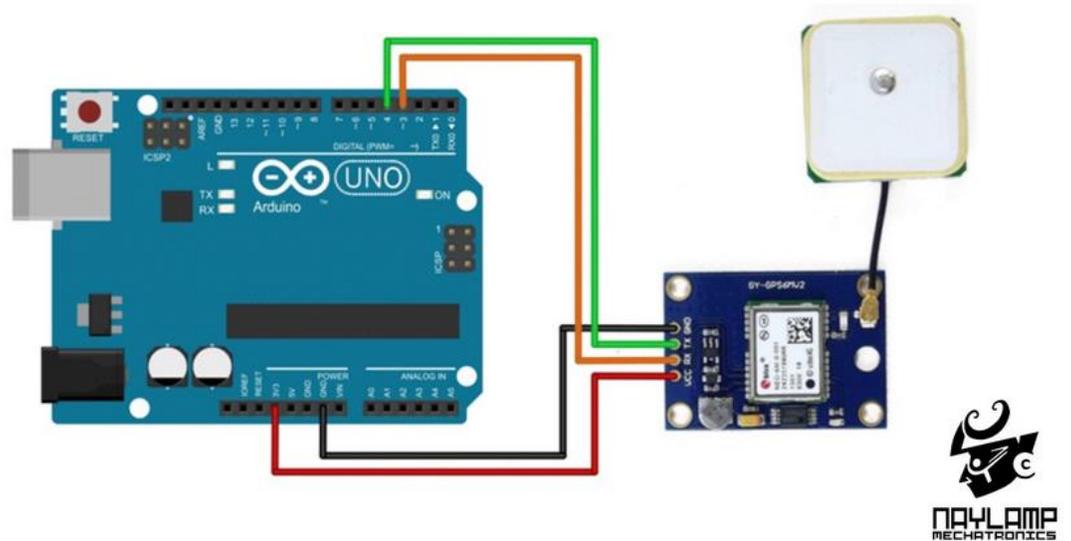


Figura 9. Diagrama de conexión modulo GY-GPS6MV2 disponible en: <https://bit.ly/39tt9iA>

### 2.2.7.4 Configuración y programación del módulo GY – GPS6MV2

Para este proceso se hace uso del software de Arduino y se coloca los siguientes comandos como se muestra en la figura 10.

The image shows a screenshot of the Arduino IDE interface. The window title is "arduino\_lcd Arduino 1.8.13". The menu bar contains "Archivo", "Editar", "Programa", "Herramientas", and "Ayuda". The toolbar includes icons for a checkmark, a back arrow, a copy icon, a paste icon, and a refresh icon. The code editor shows the following C++ code:

```
arduino_lcd $
#include <SoftwareSerial.h>

SoftwareSerial gps(4,3);
char dato=' ';
void setup()
{
  Serial.begin(115200);
  gps.begin(9600);
}

void loop()
{
  if(gps.available())
  {
    dato=gps.read();
    Serial.print(dato);
  }
}
```

Figura 10. Configuración GPS  
elaboración propia

Este programa tiene la función de leer los datos que obtiene el GPS a una velocidad y para poder visualizarlos en el monitor serial se tiene que configurar a 115200 baudios como se muestra en la figura 11.

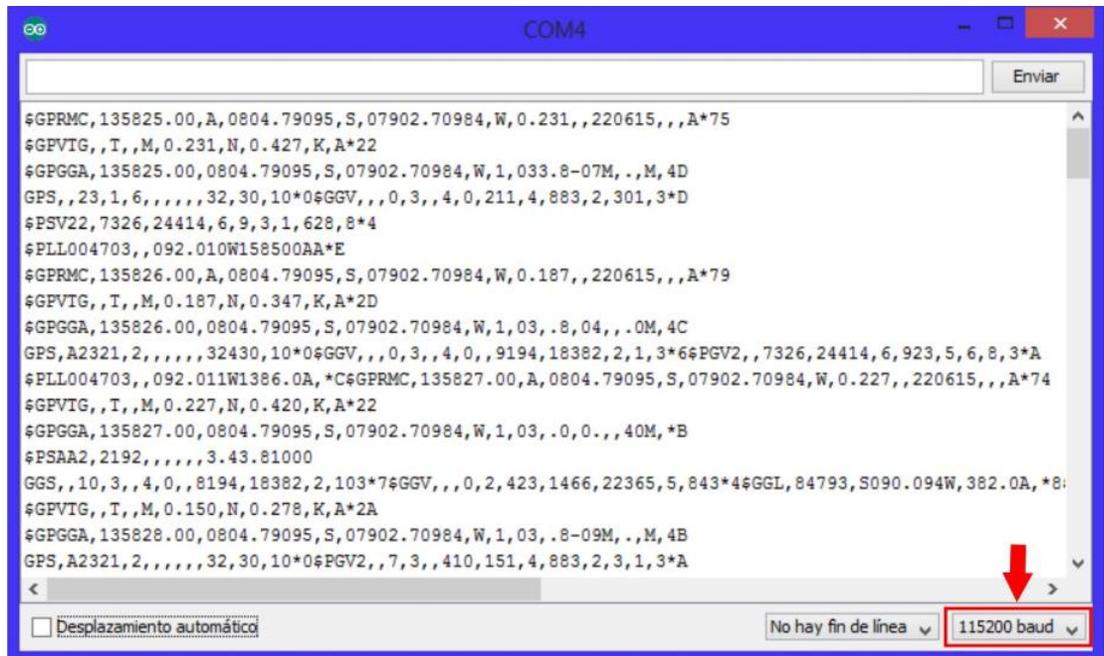


Figura 11. Configuración monitor serial disponible en: <https://bit.ly/3o9RAWq>

Los datos obtenidos en el monitor serial siguen el protocolo NMEA (National Marine Electronics Association) que tienen la siguiente estructura:

**\$GPRMC,044755.000,A,4322.0439,N,00854.5210,W,0.42,65.45,020620**

de acuerdo a esa estructura podemos extraer los datos de la siguiente manera:

- **044755.000:** Interpreta la hora GMT (04:47:55)
- **A:** Verifica que el dato sea valido
- **4322.0439:** Interpreta la longitud (43° 22.0439´)
- **N:** Interpreta al norte
- **00854.5210:** Interpreta la latitud (8° 54.5210´)
- **W:** Interpreta al oeste
- **0.42:** Interpreta la velocidad en nudos
- **65.45:** Interpreta la orientación en grados
- **020620:** Interpreta la fecha (2 de junio del 2020)

Como se puede observar en la figura 10 los datos obtenidos necesitan una codificación para su mayor comprensión y para esto nos ayuda la librería

TinyGPS que es incluye al IDE de Arduino. Luego se compila el siguiente código que se muestra en las figuras 12, 13 y 14 que nos sirve para ver los datos completos y codificados que nos proporciona el GPS.

```

arduino_1d Arduino 1.8.13
Archo Editor Programa Herramientas Ayuda

arduino_1d.h
#include "SoftwareSerial.h"
#include "TinyGPS.h"

/* This sample code demonstrates the normal use of a TinyGPS object.
 * It requires the use of SoftwareSerial, and assumes that you have a
 * 4800 baud serial GPS device hooked up on pins 4(rx) and 5(tx).
 */

TinyGPS gpr;
SoftwareSerial ss(10, 9);
static void smartDelay(unsigned long ms)
{
  static void print_float(float val, float invalid, int len, int prec);
  static void print_int(unsigned long val, unsigned long invalid, int len);
  static void print_date(TinyGPS gpr);
  static void print_status(char *str, int len);
}

void setup()
{
  Serial.begin(115200);
  Serial.println("Testing TinyGPS library v. "); Serial.println(TinyGPS::library_version());
  Serial.println("by Nicky Baxter");
  Serial.println();
  Serial.println("Date MDD Latitude Longitude Fix Date Time Date Alt Course Speed Card Distance Course Card Chars Distance (Checksum)");
  Serial.println(" (deg) (deg) Age (m) --- from GPS --- --- to London --- SK SK Fail");
  Serial.println("-----");
  ss.begin(4800);
}

void loop()
{
  float flat, fion;
  unsigned long age, date, time, chare = 0;
  unsigned short sentences = 0, failed = 0;
  static const double LONDON_LAT = 51.508131, LONDON_LON = -0.128002;

  print_int(gpr.available(), TinyGPS::GPS_INVALID_SATELLITES, 8);
  print_int(gpr.hdop(), TinyGPS::GPS_INVALID_HDOP, 8);
  gpr_fix(gpr.location(), flat, fion, age);
  print_float(flat, TinyGPS::GPS_INVALID_F_ANGLE, 10, 4);
  print_float(fion, TinyGPS::GPS_INVALID_F_ANGLE, 11, 4);
  print_int(age, TinyGPS::GPS_INVALID_AGE, 8);
  print_date(gpr);
  print_float(gpr.altitude(), TinyGPS::GPS_INVALID_ALTITUDE, 7, 2);
  print_float(gpr.course(), TinyGPS::GPS_INVALID_F_ANGLE, 7, 2);
  print_float(gpr.speed(), TinyGPS::GPS_INVALID_F_ANGLE, 6, 2);
  print_int(gpr.course(), TinyGPS::GPS_INVALID_F_ANGLE ? "****" : TinyGPS::cardinal(gpr.course()), 4);
  print_int(flat == TinyGPS::GPS_INVALID_F_ANGLE ? 0xFFFFFFFF : (unsigned long)TinyGPS::distance_between(flat, fion, LONDON_LAT, LONDON_LON) / 1000, 0xFFFFFFFF, 9);
  print_int(fion == TinyGPS::GPS_INVALID_F_ANGLE ? 0xFFFFFFFF : (unsigned long)TinyGPS::distance_between(flat, fion, LONDON_LAT, LONDON_LON) / 1000, 0xFFFFFFFF, 9);
}

```

Figura 12. Programación GPS-1 elaboración propia

```

arduino_1d Arduino 1.8.13
Archo Editor Programa Herramientas Ayuda

arduino_1d.h
print_int(flat == TinyGPS::GPS_INVALID_F_ANGLE ? 0xFFFFFFFF : (unsigned long)TinyGPS::distance_between(flat, fion, LONDON_LAT, LONDON_LON) / 1000, 0xFFFFFFFF, 9);
print_float(flat == TinyGPS::GPS_INVALID_F_ANGLE ? TinyGPS::GPS_INVALID_F_ANGLE : TinyGPS::course_to(flat, fion, LONDON_LAT, LONDON_LON), TinyGPS::GPS_INVALID_F_ANGLE, 7, 2);
print_int(flat == TinyGPS::GPS_INVALID_F_ANGLE ? "****" : TinyGPS::cardinal(TinyGPS::course_to(flat, fion, LONDON_LAT, LONDON_LON)), 4);

gpr_start(chare, sentences, ifailed);
print_int(chare, 0xFFFFFFFF, 4);
print_int(sentences, 0xFFFFFFFF, 10);
print_int(ifailed, 0xFFFFFFFF, 3);
Serial.println();
}

smartDelay(1000);
}

static void smartDelay(unsigned long ms)
{
  unsigned long start = millis();
  do
  {
    while (ss.available())
      gpr.encode(ss.read());
  } while (millis() - start < ms);
}

static void print_float(float val, float invalid, int len, int prec)
{
  if (val == invalid)
  {
    while (len-- > 2)
      Serial.print(" ");
    Serial.print(" ");
  }
  else
  {
    Serial.print(val, prec);
    int vl = abs((int)val);
    int flen = prec + (val < 0.0 ? 2 : 1) // . and -
    flen += vl >= 100 ? 4 : vl >= 10 ? 3 : 1;
    for (int i=flen; i--len; ++i)
      Serial.print(" ");
  }
  smartDelay(0);
}

static void print_int(unsigned long val, unsigned long invalid, int len)
{
  char sz[32];
  if (val == invalid)
    strcpy(sz, "*****");
  else
    sz[0]

```

Figura 13. Programación GPS-2 elaboración propia



## 2.2.8 Detección de obstáculos

Un sistema de detección de obstáculos es importante es un robot teleoperado ya que evita la colisión con algún objeto o superficie al momento del control del robot.

### 2.2.8.1 Sensor ultrasónico HC-SR04

Los sensores ultrasónicos detectan objetos con distancia entre centímetros hasta metros. La operación de detección la realiza mediante la emisión de un sonido y el tiempo en que regresa al sensor y analiza los datos para determinar la distancia en la que se encuentra el objeto

### 2.2.8.2 Conexión del sensor ultrasónico HC-SR04

Las conexiones van de la siguiente manera: el pin 5V se conecta a una fuente VCC del controlador, el pin GND se conecta al pin GND del controlador el pin Trig y el pin Echo se conectan a cualquier pin digital de controlador, una mejor explicación se puede visualizar en la figura 16.

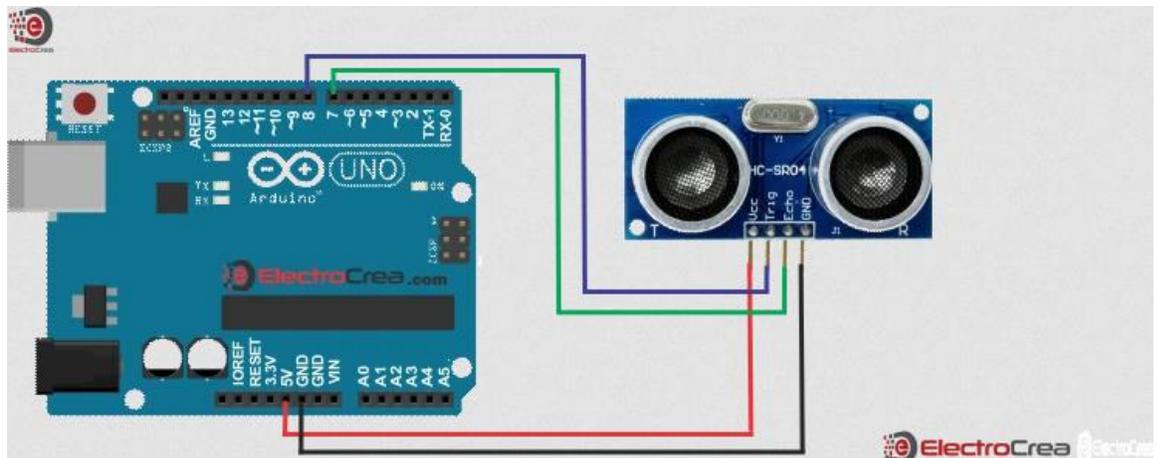


Figura 16. Diagrama de conexión sensor ultrasónico HC-SR04 disponible en: <https://bit.ly/36nHYkT>

### 2.2.8.3 Configuración y programación sensor ultrasónico HC-SR04

A continuación, en la figura 17 se muestra el código que debe programarse en el software Arduino IDE para la programación y detección del sensor ultrasónico HC-SR04

```
arduino_ide Arduino 1.8.13
Archivo Editar Programa Herramientas Ayuda

arduino_ide$

#define echoPin 7 // Echo Pin
#define trigPin 8 // Trigger Pin
#define LEDPin 13

int maximumRange = 200; // Maximo rango a detectar en cm
int minimumRange = 0; // Minimo rango a detectar en cm
long duration, distance; // Duración usada para calcular la distancia

void setup() {
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT); // Trig es la salida del ultrasonido
  pinMode(echoPin, INPUT); // Echo es la entrada de el sonido regresando al sensor.
  pinMode(LEDPin, OUTPUT);
}

void loop() {
  //El siguiente ciclo de Trig y Echo es usado para determinar
  //la distancia del objeto mas cercano emitiendo sonidos
  //y calculando el tiempo que tardan en regresar*/
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration/59.2;

  if (distance >= maximumRange || distance <= minimumRange){
    //Si la distancia medida es mayor que 200cm y menor que 0cm
    //el monitor serial indicara que esta Fuera de rango y encendera el led*/
    Serial.println("Fuera de Rango");
    digitalWrite(LEDPin, HIGH);
  }
  else {
    //Por otro lado si el objeto se encuentra en el rango
    //de medicion muestra la distancia en el monitor serial*/
    Serial.print(distance);
    Serial.println(" CM");
    digitalWrite(LEDPin, LOW);
  }
  delay(50); //Restrano de 50ms entre lectura y lectura
}
```

Figura 17. Configuración HC–SR04 elaboración propia

Al terminar de compilar y ejecutar el código las mediciones del sensor nos aparecerán en el monitor serial del Arduino IDE como se observa en la figura 18.

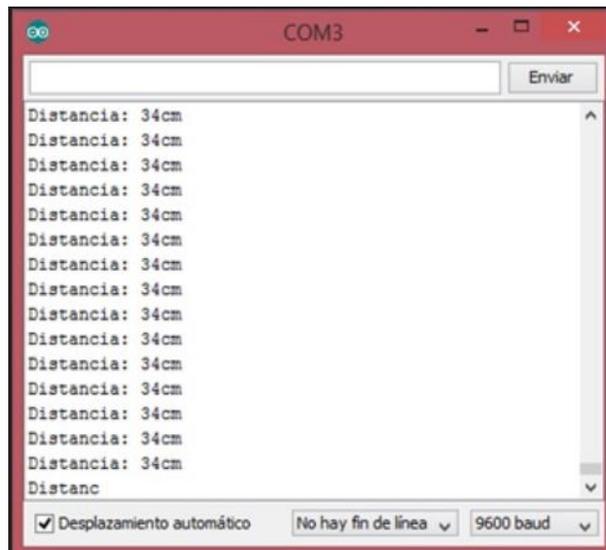


Figura 18. Datos del sensor ultrasónico HC–SR04 disponible en: <https://bit.ly/3mpcb8C>

## 2.2.9 Tecnologías de reconocimiento y supervisión

Las más nombradas e innovadoras tecnologías especializadas en supervisar y reconocer son las siguientes:

### **2.2.9.1 Fotografías aéreas**

Con el uso de cámaras poco precisas instaladas en los aeroplanos, se diseñaron los primeros prototipos de los actuales dispositivos de detección remota. Revisando las capturas fotográficas se podía apreciar el posicionamiento y el número de las fuerzas atacantes. Esto les permitía tener una ubicación exacta y podían prevenir futuros ataques. También se usó una película infrarroja para precisar la vegetación y diferenciarlas con los accesorios de camuflaje. (15)

### **2.2.9.2 Escáneres multiespectrales**

En los años recientes, este tipo de escáneres están siendo muy necesarios e indispensables para la detección remota. Su aplicación es variada ya que va desde la tierra hasta los satélites.

Estos detectores multiespectrales de cada escáner están diseñados para percibir radiaciones de lugares determinados. La cantidad de conductos, su tamaño y posición en la sombra electromagnética es distinta para cada sensor multiespectral, los cuales son importantes para obtener las distintas características de valor espectral y espacial. Dichas mezclas de factores ayudan a especificar el empleo de imágenes que van a ser más útiles para el sensor. (15)

### **2.2.9.3 Cámara termográfica**

Es un dispositivo que percibe imágenes mediante la energía infrarroja emitida del objetivo del disparo señalando la variación de temperatura en la imagen. La calidad de la imagen y resolución de la misma varía del modelo de cámara adquirida. (15)

### **2.2.10 Cámara termográfica AMG8833**

Esta cámara termográfica utiliza el sensor de Panasonic AMG8833 que está compuesto por un array de sensores térmicos de 8x8, que le permite detectar temperaturas entre 0°C y 80°C

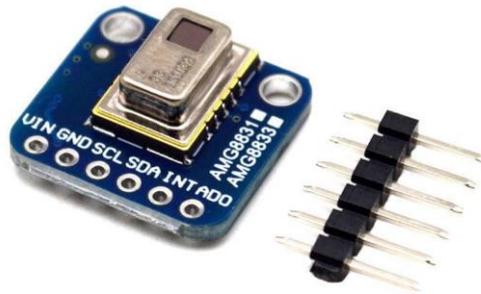


Figura 19. Cámara termográfica AMG8833 disponible en: <https://bit.ly/37qxGjf>

#### 2.2.10.1 Conexión cámara termográfica AMG8833

Las conexiones van de la siguiente manera: el pin Vin se conecta a una fuente VCC del controlador, el pin GND se conecta al pin GND del controlador, el pin SCL, SDA, INT y ADO se conectan a cualquier pin digital previamente configurado de controlador, los pines de conexión se encuentran en la figura 20 para una mejor visualización.



Figura 20. Pines de cámara AMG8833 disponible en: <https://bit.ly/3loPKzd>

#### 2.2.10.2 Configuración y programación AMG8833

Para configurar correctamente la cámara se necesita descargar la librería Adafruit\_AMG88xx e instalar en el software Arduino IDE y luego compilar el siguiente código para verificar la integridad de la cámara.

```
arduino_1 Arduino 1.8.13
Archivo Editor Programa Herramientas Ayuda
arduino_1
#include <Wire.h>
#include <Adafruit_AMG88xx.h>

Adafruit_AMG88xx amg;

void setup() {
  Serial.begin(9600);
  Serial.println(F("AMG88xx test"));

  bool status;

  // default settings
  status = amg.begin();
  if (!status) {
    Serial.println("Could not find a valid AMG88xx sensor, check wiring!");
    while (1);
  }

  Serial.println("-- Thermistor Test --");
  Serial.println();
  delay(100); // let sensor boot up
}

void loop() {
  Serial.print("Thermistor Temperature = ");
  Serial.print(amg.readThermistor());
  Serial.println(" °C");

  Serial.println();

  //delay a second
  delay(1000);
}
```

Figura 21. Configuración AMG8833  
elaboración propia

Luego al abrir el monitor serial se visualizará la temperatura que está captando la cámara como se ve en la imagen 22.

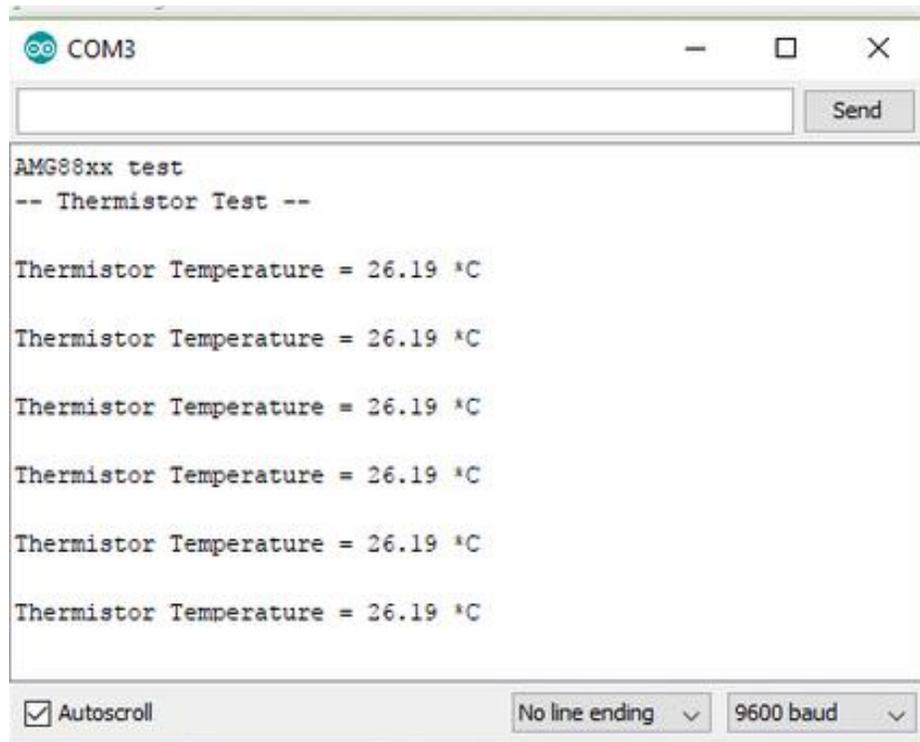


Figura 22. Test de prueba Cámara termográfica AMG8833  
disponible en: <https://bit.ly/33u7Dqb>

Una vez comprobado la lectura de los sensores térmicos se procede a compilar y ejecutar el siguiente código que se muestra en las figuras 23 y 24 para la visualización del array de valores en el monitor serial y en la pantalla LCD.

```

Arduino IDE - Programa: AM8833
Arduino IDE

#include <Adafruit_GFX.h> // Core graphics library
#include <Adafruit_ST7735.h> // Hardware-specific library
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_AM8833.h>
#define TFT_CS 10 //chip select pin for the TFT screen
#define TFT_RST 9 // you can also connect this to the Arduino reset
// in which case, set chip #define pin to 0!
#define TFT_DC 8

//low range of the sensor (this will be blue on the screen)
#define MINTEMP 22
//high range of the sensor (this will be red on the screen)
#define MAXTEMP 34

//the colors we will be using
const uint16_t PROGMEM colors[] = {
  0x0000, 0x000F, 0x001F, 0x0030, 0x0040, 0x0050, 0x0060, 0x0070, 0x0080, 0x0090, 0x00A0, 0x00B0, 0x00C0, 0x00D0, 0x00E0, 0x00F0,
  0x0100, 0x0110, 0x0120, 0x0130, 0x0140, 0x0150, 0x0160, 0x0170, 0x0180, 0x0190, 0x01A0, 0x01B0, 0x01C0, 0x01D0, 0x01E0, 0x01F0,
  0x0200, 0x0210, 0x0220, 0x0230, 0x0240, 0x0250, 0x0260, 0x0270, 0x0280, 0x0290, 0x02A0, 0x02B0, 0x02C0, 0x02D0, 0x02E0, 0x02F0,
  0x0300, 0x0310, 0x0320, 0x0330, 0x0340, 0x0350, 0x0360, 0x0370, 0x0380, 0x0390, 0x03A0, 0x03B0, 0x03C0, 0x03D0, 0x03E0, 0x03F0,
  0x0400, 0x0410, 0x0420, 0x0430, 0x0440, 0x0450, 0x0460, 0x0470, 0x0480, 0x0490, 0x04A0, 0x04B0, 0x04C0, 0x04D0, 0x04E0, 0x04F0,
  0x0500, 0x0510, 0x0520, 0x0530, 0x0540, 0x0550, 0x0560, 0x0570, 0x0580, 0x0590, 0x05A0, 0x05B0, 0x05C0, 0x05D0, 0x05E0, 0x05F0,
  0x0600, 0x0610, 0x0620, 0x0630, 0x0640, 0x0650, 0x0660, 0x0670, 0x0680, 0x0690, 0x06A0, 0x06B0, 0x06C0, 0x06D0, 0x06E0, 0x06F0,
  0x0700, 0x0710, 0x0720, 0x0730, 0x0740, 0x0750, 0x0760, 0x0770, 0x0780, 0x0790, 0x07A0, 0x07B0, 0x07C0, 0x07D0, 0x07E0, 0x07F0,
  0x0800, 0x0810, 0x0820, 0x0830, 0x0840, 0x0850, 0x0860, 0x0870, 0x0880, 0x0890, 0x08A0, 0x08B0, 0x08C0, 0x08D0, 0x08E0, 0x08F0,
  0x0900, 0x0910, 0x0920, 0x0930, 0x0940, 0x0950, 0x0960, 0x0970, 0x0980, 0x0990, 0x09A0, 0x09B0, 0x09C0, 0x09D0, 0x09E0, 0x09F0,
  0x0A00, 0x0A10, 0x0A20, 0x0A30, 0x0A40, 0x0A50, 0x0A60, 0x0A70, 0x0A80, 0x0A90, 0x0AA0, 0x0AB0, 0x0AC0, 0x0AD0, 0x0AE0, 0x0AF0,
  0x0B00, 0x0B10, 0x0B20, 0x0B30, 0x0B40, 0x0B50, 0x0B60, 0x0B70, 0x0B80, 0x0B90, 0x0BA0, 0x0BB0, 0x0BC0, 0x0BD0, 0x0BE0, 0x0BF0,
  0x0C00, 0x0C10, 0x0C20, 0x0C30, 0x0C40, 0x0C50, 0x0C60, 0x0C70, 0x0C80, 0x0C90, 0x0CA0, 0x0CB0, 0x0CC0, 0x0CD0, 0x0CE0, 0x0CF0,
  0x0D00, 0x0D10, 0x0D20, 0x0D30, 0x0D40, 0x0D50, 0x0D60, 0x0D70, 0x0D80, 0x0D90, 0x0DA0, 0x0DB0, 0x0DC0, 0x0DD0, 0x0DE0, 0x0DF0,
  0x0E00, 0x0E10, 0x0E20, 0x0E30, 0x0E40, 0x0E50, 0x0E60, 0x0E70, 0x0E80, 0x0E90, 0x0EA0, 0x0EB0, 0x0EC0, 0x0ED0, 0x0EE0, 0x0EF0,
  0x0F00, 0x0F10, 0x0F20, 0x0F30, 0x0F40, 0x0F50, 0x0F60, 0x0F70, 0x0F80, 0x0F90, 0x0FA0, 0x0FB0, 0x0FC0, 0x0FD0, 0x0FE0, 0x0FF0
};

Adafruit_ST7735 tft = Adafruit_ST7735(TFT_CS, TFT_DC, TFT_RST);

```

Figura 23. Programación AMG8833 elaboración propia

```

Arduino IDE - Programa: AM8833
Arduino IDE

Adafruit_ST7735 tft = Adafruit_ST7735(TFT_CS, TFT_DC, TFT_RST);

Adafruit_AM8833 amg;
unsigned long delayTime;
float pixels[AM8833_PIXEL_ARRAY_SIZE];
uint16_t displayPixelWidth, displayPixelHeight;

void setup() {
  Serial.begin(9600);
  Serial.println(F("AM8833 thermal camera"));
  tft.begin(INITR_144GREENTAB); // initialize a ST7735 chip, black tab
  tft.fillScreen(ST7735_BLACK);

  displayPixelWidth = tft.width() / 8;
  displayPixelHeight = tft.height() / 8;

  //tft.setRotation(3);

  bool status;
  // default settings
  status = amg.begin();
  if (!status) {
    Serial.println("Could not find a valid AMG8833 sensor, check wiring!");
    while (1);
  }
  Serial.println("Thermal Camera Test --");
  delay(100); // let sensor boot up
}

void loop() {
  //read all the pixels
  amg.readPixels(pixels);

  for(int i=0; i<AM8833_PIXEL_ARRAY_SIZE; i++) {
    uint16_t colorIndex = map(pixels[i], MINTEMP, MAXTEMP, 0, 255);
    colorIndex = constrain(colorIndex, 0, 255);

    //draw the pixel!
    tft.fillRect(displayPixelWidth * floor(i / 8), displayPixelHeight * (i % 8),
      displayPixelHeight, displayPixelWidth, amgColors[colorIndex]);
  }
}

```

Figura 24. Programación AMG8833 elaboración propia

Terminada la ejecución del código se puede visualizar los valores termográficos en el monitor serial como se visualiza en la figura 25.

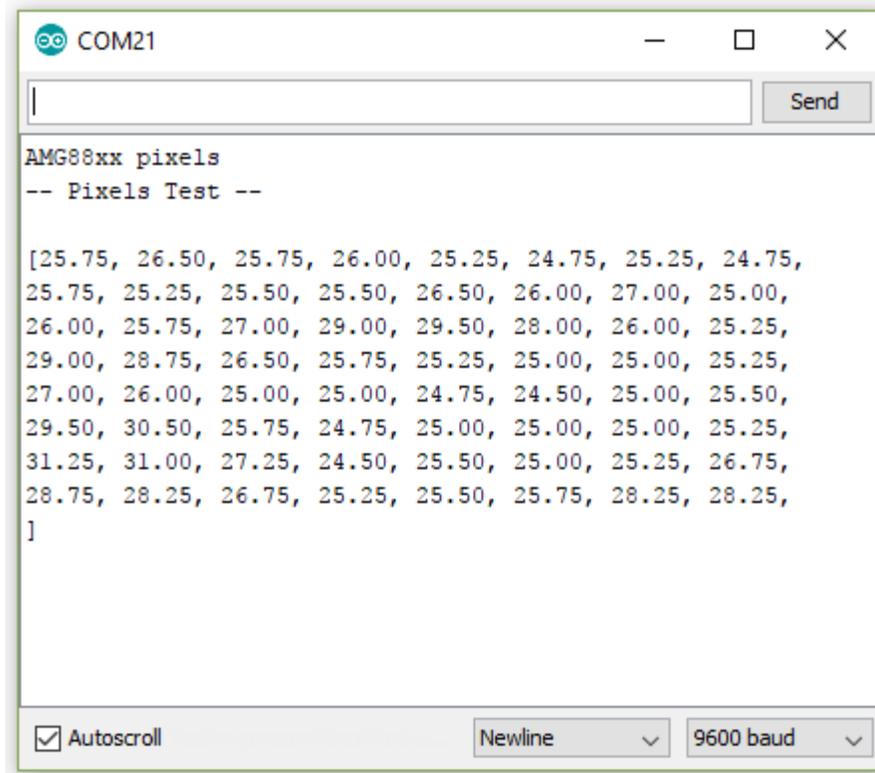


Figura 25. Datos en array de cámara termográfica AMG8833 disponible en: <https://bit.ly/36lMcJt>

## 2.2.11 Fuentes de energía

Se ubicaron dos medios de alimentación de energía que son muy empleados en los distintos modelos de sistemas tele operada y son los siguientes:

### 2.2.11.1 Baterías Eléctricas

En la tabla 2 se puede observar las baterías más usadas que son las baterías níquel-metal hidruro (NiMH) y las baterías de litio (Li-Po) con sus respectivas características. Se puede apreciar que el Li-Po es de lejos mejor que la otra batería, pero cuenta con un inconveniente y es que son muy costosas.

Tabla 2. Diferencias entre baterías NiMH y Li-Po

Variables	NiMH	Li-Po
Volumen	Mayor	Menor
Precio	A mitad de precio que Li-Po	Alto
Peso	Mayor	Menor
Recargable	Si	Si
Velocidad de carga (Amps)	1	8

Nota: NiMH: Niquel-metal Hidruro; Li-Po: Polimero de iones de litio  
Elaboración propia

### 2.2.11.2 Combustible

Este tipo de generador de energía no es muy usado en drones o vehículos operados a distancia, pero se ha visto algunos drones que si los emplean como fuente de energía hidrogeno gaseoso, propano e hidrogeno coronizado. La empresa (Intelligent Energy) ha implementado un VANT multi rotor el cual reacciona con un tipo de hidrógeno que este comprimido. Aquí se puede ver que este tipo de combustible le da una duración como máximo de dos horas en comparación con las baterías eléctricas se distingue que tiene la capacidad de quintuplicar la capacidad de estos almacenadores eléctricos.

### 2.2.12 Motores

El motor es un aparato que transmuta la energía eléctrica a una energía mecánica también conocida como motor de corriente continua entre sus siglas DC, dicho motor produce un movimiento rotatorio. En algunos otros tipos, ejecutan tracción sobre un carril. Dichos motores también se perciben como motores lineales (16).



Figura 26. Motor DC disponible en tienda electrónica Conmaquel

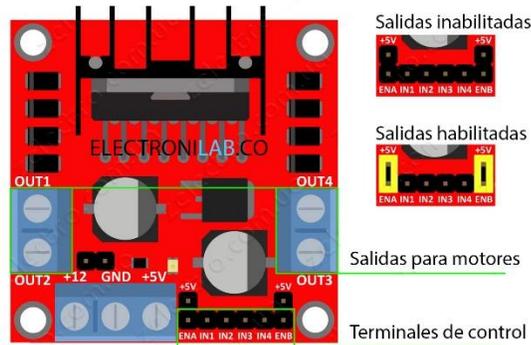
### 2.2.13 Drivers de motores

Es indispensable para lograr controlar la velocidad del motor como también el sentido de giro, si el usuario habilito o deshabilita las entradas que posee el driver lograra conseguir lo mencionado, de tal manera se mantiene estable nuestra Unidad de Control. Hoy en día no es posible encontrar los controladores apropiados para la aplicación de los motores, por lo tanto, se tomó en cuenta el controlador:

#### 2.2.13.1 Driver L298N

El módulo L298N permite al usuario manipular un motor DC con caja reductora por cada driver (17).

Figura 27. Driver L298N



disponible en tienda electrónica Conmaquel

Asimismo, cuenta con los elementos imprescindibles para ejecutar sin obligación de adquirir componentes adicionales, ya sea diodos de protección o un LM7805 que se un regulador de voltaje y sirve para abastecer con 5 voltios al segmento lógico del circuito integrado L298N. Tiene a su disposición con jumper de elección para disponer cada una de las salidas del L298N (A y B). (17).

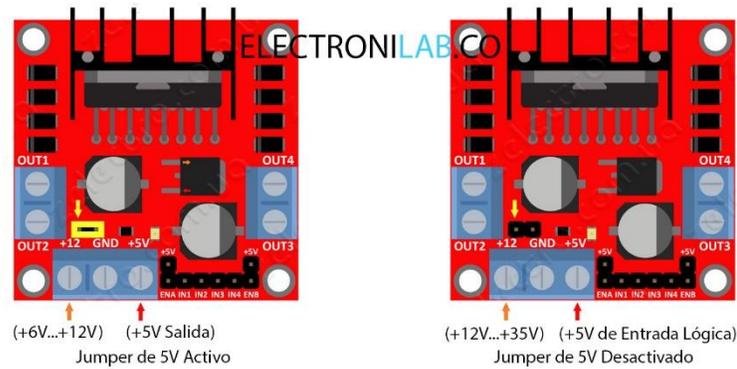


Figura 28. Abastecimiento del L298N disponible en tienda electrónica Conmaquel

El abastecimiento puede ser de dos modos debido al LM7805. Si se selecciona los 5 voltios, el módulo adquiere un abastecimiento de entre 6 voltios a 12 voltios, y si los 5 voltios está inactivo, permite un abastecimiento de entre 12 voltios a 35 voltios (17).

#### 2.2.14 Programación

La Programación es el desarrollo en el que se codifica, diseña, limpia y preservar el código fuente de un programa. Además, los diferentes softwares de programas se encargan de verificar la integridad del código fuente (18).

La finalidad es de establecer un sistema mediante software, que más adelante será empleado de una forma autónoma por medio de un ordenador. Para llegar asemejarse al lenguaje natural, la programación sigue un patrón de reglas y diferentes instrucciones, normas y expresiones. Para brindar un resultado a un problema definido se requiere seguir normas o reglas las cuales se emplean para la creación de un programa y con el llegar a ejecutar. (18).

Para que la computadora siga una secuencia de pasos de órdenes tiene un responsable el cual es el lenguaje de programación. De esta manera se concluye que el intermediario entre el usuario y el ordenador es el lenguaje de programación, en el cual se puede llegar a dar respuestas a diferentes problemas mediante el ordenador además haciendo uso de diversas funciones, que el ordenador lo interpreta y lo ejecuta para un trabajo específico (18).

### 2.2.15 Sistema de comunicación inalámbrica

Es el proceso por el cual se puede intercambiar datos entre dos o más dispositivos que pueden estar en movimiento o estar a una cierta distancia excluyendo el uso de cables u otro medio físico (19).

### 2.2.16 Módulo NRF24L01

Es un dispositivo electrónico que opera en la frecuencia de 2.4 GHz de muy bajo consumo energético y rango de hasta 1000 metros si tiene integrado un amplificador de ruido bajo (LNA), de fácil uso e integración en un sistema Arduino



Figura 29. NRF24L01  
disponible en tienda electrónica Conmaquel

### 2.2.17 Origen de los derrumbes

En el Perú, los fenómenos geológicos como desprendimiento de rocas, derrumbes y deslizamientos, se originan en gran parte de sus 106 cuencas hidrográficas. En la tabla 3 podemos observar la población afectada por deslizamiento o inundaciones en todas las regiones del Perú por lo que se determina que la mayor parte de las casas son afectadas por deslizamientos o denominados huaycos.

Tabla 3. Población y viviendas afectadas ante la ocurrencia de eventos hidrológicos

Región	Número de centros poblados probablemente		Población probablemente		Número de viviendas probablemente	
	afectados por deslizamientos	afectados por inundaciones	afectada por deslizamientos	afectada por inundaciones	afectada por deslizamientos	afectada por inundaciones
Amazonas	239	26	42 244	6 951	13 598	2 049
Áncash	6 990	165	583 066	73 223	193 596	20 018
Apurímac	3 827	0	374 764	0	137 575	0
Arequipa	4 781	12	1 016 142	858	298 310	844
Ayacucho	7 029	4	530 308	434	196 751	137
Cajamarca	5 068	31	909 708	13 867	288 739	4 544
Callao	1	0	277 895	0	80 434	0
Cusco	7 627	60	593 973	108 562	194 340	28 914
Huancavelica	6 436	6	424 478	47	146 685	30
Huánuco	5 408	47	331 380	164 989	106 182	42 421
Ica	477	294	69 873	254 507	24 601	67 486
Junín	2 672	8	794 632	504	227 923	257
La Libertad	2 913	397	449 146	818 892	128 851	203 354
Lambayeque	371	714	53 737	935 983	14 658	220 678
Lima	4 041	149	770 179	2 208 957	226 085	510 999
Moquegua	1 232	5	82 084	59 694	32 635	17 920
Pasco	2 049	17	177 764	237	50 287	124
Piura	2 025	373	531 712	750 411	134 184	180 134
Puno	7 552	924	769 149	368 114	312 467	130 538
San Martín	7	0	75	0	40	0
Tacna	634	17	100 620	74	32 931	78
Tumbes	93	27	19 768	128 018	6 101	34 971
<b>Total</b>	<b>71 472</b>	<b>3 276</b>	<b>8 902 697</b>	<b>5 894 322</b>	<b>2 846 973</b>	<b>1 465 496</b>

Disponibile en DGOT- MNAM. 2018

### 2.3 Definición de términos básicos

- 1 **Sensor:** Es un dispositivo electrónico con capacidad de detectar magnitudes físicas y lo convierte en una señal eléctrica.
- 2 **Actuador:** Es un dispositivo que proporciona fuerza o movimiento a partir de una señal eléctrica de ingreso.
- 3 **Microcontrolador:** Es un circuito integrado que puede programarse y ejecutar ordenes previamente grabadas en la memoria.
- 4 **Arduino:** Es una plataforma diseñada para la programación de una placa que contienen un microcontrolador y sirve para el almacenamiento de procesos simples y complejos
- 5 **Robot:** Es una maquina programable y automática que puede realizar determinadas ordenes de manera autónoma y sustituir la mano humana en algunas tareas, por ejemplo, algo peligroso o repetitivo.
- 6 **Teleoperado:** Dispositivos que puede ser controlado a distancia por una persona de manera directa o mediante un ordenador.
- 7 **Sismo:** Vibración generada por el movimiento repentino en las capas tectónicas de la tierra.
- 8 **Derrumbe:** Caída, hundimiento, hecatombe o desplome de una cosa que está firme, usualmente ocurre en una edificación.
- 9 **EEPROM:** Es un tipo de memoria que puede mantener los datos incluso sin tener suministro de energía
- 10 **VCC:** Siglas que significan corriente continua
- 11 **Baudio:** Se refiere la rapidez de transmisión de símbolos por segundo.
- 12 **GND:** (Ground) significa tierra y es un punto que significa 0 voltios.
- 13 **GMT:** (Greenwich Mean Time) Hora del meridiano de Greenwich

## **CAPITULO III**

### **METODOLOGIA**

#### **3.1 Diseño metodológico**

##### **3.1.1 Tipo de investigación**

La investigación tecnológica utiliza de conocimientos adquiridos y a la vez se adquieren unos nuevos para dar solución a problemas que beneficien la sociedad. (20)

Para solucionar la problemática planteada previamente, utilizaremos los conocimientos adquiridos de diseño mecánico, electrónico y control para diseñar el robot móvil teleoperado a distancia y se determinaran los resultados mediante simulaciones.

##### **3.1.2 Alcance de investigación**

El alcance de la investigación es aplicado porque se adapta los resultados de la investigación experimental para diseñar nuevas tecnologías que se usaran para la solución de los problemas de la sociedad. (21)

El presente trabajo tiene un alcance de investigación aplicado porque utilizaremos conocimientos adquiridos en el área de diseño 3D para el modelamiento del prototipo de nuestro robot y la búsqueda de resultados mediante simulaciones.

### 3.1.3 Metodología aplicada para el desarrollo de solución

La metodología utilizada será VDI 2206.

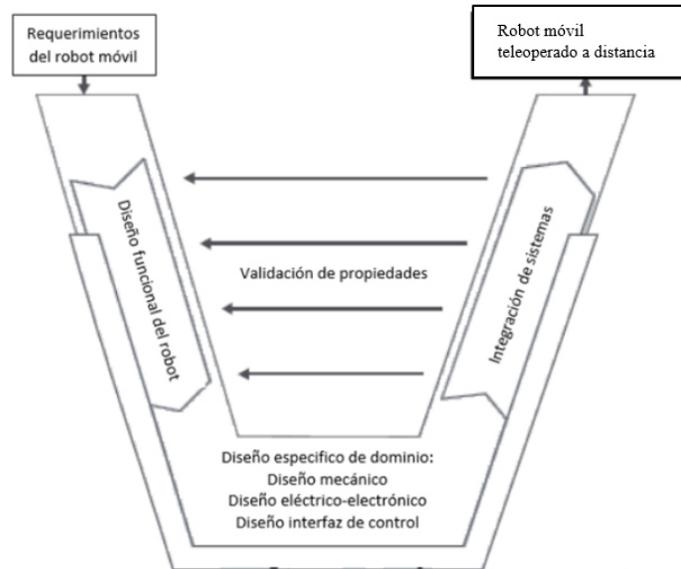


Figura 30. Modelo de VDI 2206

disponible en ScienceDirect VDI 2206- A New Guideline for the Design of Mechatronic Systems

En la figura N°10 se puede observar que usaremos una metodología en base a validación de parámetros para cumplir nuestros requerimientos

#### **Etapa 1**

**Fase de especificaciones:** Definir y verificar los requerimientos

El inicio y fin del proyecto forman los extremos del ciclo. Esta fase se compone del análisis de requisitos y especificaciones.

Se deben establecer los requerimientos para el diseño del robot móvil como medidas físicas y mecánicas que le permitan desplazarse sobre un terreno accidentado.

➤ **Estado del arte**

Robots ya desarrollados en el ámbito de inspección o exploración de determinados espacios.

➤ **Lista de exigencias**

Tabla 4. Formato de lista de exigencias

<b>LISTA DE EXIGENCIAS</b>		<b>Universidad Continental</b>	<b>Pág:</b>	
<b>PROYECTO</b>		<b>Ingeniería Mecatrónica</b>	<b>Fecha:</b>	
<b>Características</b>	<b>Deseos o Exigencias</b>	<b>Condiciones</b>	<b>Responsable</b>	

Elaboración propia

**Etapa 2**

**Fase funcional:** Definir funciones

En esta etapa se debe formular una estructura de funciones que puede considerarse como una caja negra. Por esa razón se debe plantear la secuencia de operaciones que se realizara en el desarrollo del diseño general.

➤ **Black Box**

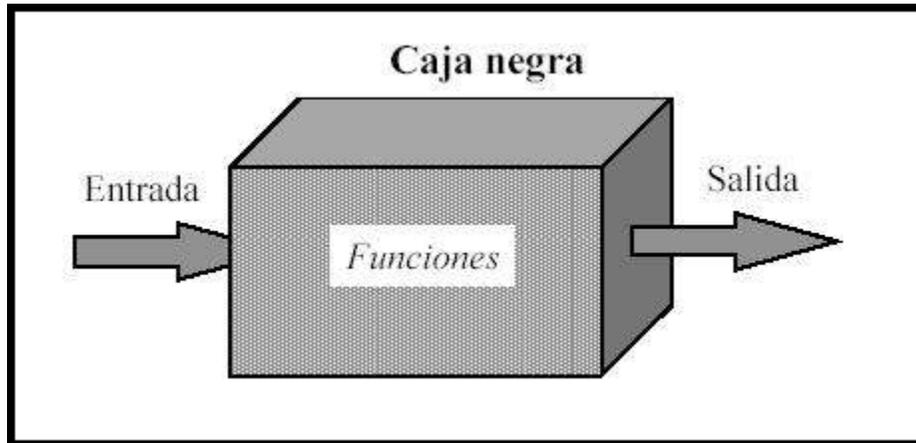


Figura 31. Caja negra  
Disponibile en EcuRed

➤ **Caja blanca**

**WHITE BOX TESTING APPROACH**

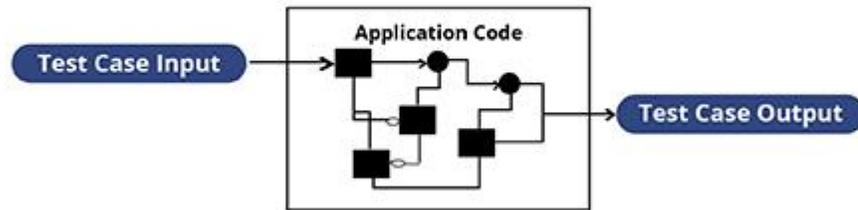


Figura 32. Caja blanca  
Disponible en Invensis

➤ **Matriz morfológica**

Subfunciones \ Soluciones		Soluciones					
		1	2	...	j	...	m
1	$F_1$	$S_{11}$	$S_{12}$		$S_{1j}$		$S_{1m}$
2	$F_2$	$S_{21}$	$S_{22}$		$S_{2j}$		$S_{2m}$
⋮		⋮	⋮		⋮		⋮
i	$F_i$	$S_{i1}$	$S_{i2}$		$S_{ij}$		$S_{im}$
⋮		⋮	⋮		⋮		⋮
n	$F_n$	$S_{n1}$	$S_{n2}$		$S_{nj}$		$S_{nm}$

2
1

Figura 33. Matriz morfológica  
Disponible en desafío latinoamericano de innovadores

**Etapas 3**

**Fase de diseño:** Definir componentes del diseño

- ✓ Diseño mecánico del robot
  - Selección del sistema de movimiento y estructura del robot
- ✓ Diseño electrónico del robot

Selección de la fuente de alimentación del robot, control de los motores y diseño de placa electrónica para la etapa de control

- ✓ Diseño del sistema de reconocimiento del robot

Selección del sistema de reconocimiento del robot

- ✓ Diseño de control del robot

Selección de medio de comunicación del robot al operario y diseño de aplicación de control

#### **Etapa 4**

**Fase de codificación:** Implementación del diseño y simulación

En esta etapa se debe implementar los componentes del diseño cumpliendo la estructura de funciones para la simulación y verificación de los requerimientos

## CAPITULO IV

### ANALISIS Y DISEÑO DE LA SOLUCIÓN

#### 4.1 Estado del arte

##### 4.1.1 Carro tractor RX-130

Es un robot tractor de la empresa PANATEC Agua & medio ambiente, que sirve para la inspección de tuberías desde 140 mm hasta 1600 mm. Dispone de una cámara a color con obturador automático. Un sensor interno de presión digital. Iluminación delantera con 3 leds de alta potencia 12000 lux. Las medidas son de 310 x 110 x 90mm y un peso de 6 Kg. La transmisión de datos es mediante un cable de la misma empresa.



Figura 34. Carro tractor RX-130  
Disponible en PANATEC Agua & Medio ambiente

#### 4.1.2 Ixnamiki Olinki

El robot fue prototipado en la facultad de ingeniería de la Universidad Panamericana de Aguascalientes por sus estudiantes, entre sus funciones cuenta con una cámara térmica para la búsqueda de supervivientes en las zonas más afectadas, un sensor para detectar los niveles de CO2, además de contar con un brazo mecánico con capacidad de hacer trabajo duro o portar aparatos de comunicación, comunicación wifi o ethernet, con un sistema oruga que le permite desplazarse y tener una gran fuerza de arranque.

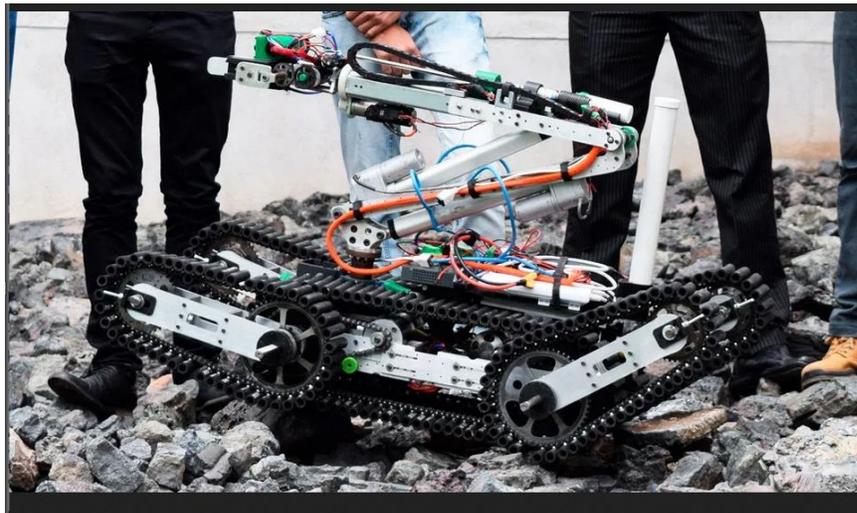


Figura 35. Ixnamiki Olinki  
Disponible en el diario “El Financiero”

#### 4.2 Lista de exigencias

Tabla 5. Lista de exigencias

<b>LISTA DE EXIGENCIAS</b>		<b>Universidad Continental</b>	<b>Pág.:</b>	1 de 2
<b>PROYECTO</b>		<b>Ingeniería Mecatrónica</b>	<b>Fecha:</b>	24/08/2020
<b>Características</b>	<b>Deseos o Exigencias</b>	<b>Condiciones</b>	<b>Responsable</b>	
Función	E	Diseñar un robot móvil	Grupo de trabajo	
Función	E	La máquina deberá poder movilizarse en terreno accidentado	Grupo de trabajo	

<b>LISTA DE EXIGENCIAS</b>		<b>Universidad Continental</b>	<b>Pág.:</b>	2 de 2
Función	D	En todo momento se buscará que el robot sea funcional y económico	Grupo de trabajo	
Función	E	El alcance de control no debe ser menor a 100 metros a la redonda	Grupo de trabajo	
Función	E	Las cámaras deberán transmitir las imágenes en tiempo real y con un mínimo de delay	Grupo de trabajo	
Función	E	El sistema electrónico contará con reguladores de voltaje, actuadores y sensores que envíen datos hacia un controlador	Grupo de trabajo	
Función	E	El sistema de control podrá controlar los motores de movimiento y giro	Grupo de trabajo	
Función	D	El robot tendrá un sistema de alerta de seguridad para verificar el funcionamiento correcto de todos los componentes	Grupo de trabajo	
Ergonomía	E	El sistema electrónico debe ser conectado con un orden para realizar reparaciones y/o mantenimientos	Grupo de trabajo	
Geométrica	D	Las dimensiones del robot no serán mayores a 60 x 50 x 50 cm	Grupo de trabajo	
Energía	E	El consumo de energía deberá estar correctamente distribuido	Grupo de trabajo	
Energía	D	El robot debe contar con baterías que suministren la energía requerida para su correcto funcionamiento	Grupo de trabajo	

Elaboración propia

### 4.3 Black-Box



#### ENTRADAS:

- ✓ **Señal visual y auditiva de la cámara normal.**  
Imagen y sonido enviados en tiempo real de acuerdo a la posición de la cámara.
- ✓ **Señal visual de la cámara termográfica.**  
Imagen con patrones de calor indispensable para la identificación de personas atrapadas por la diferencia de temperatura captada en las imágenes
- ✓ **Datos del sensor de proximidad**  
Sirve para evitar algún tipo de colisión en el movimiento del robot deteniendo los motores hasta cambiar la trayectoria y no tener obstáculos
- ✓ **Ordenes de movimiento del operador**  
Establecidos a través de dos joysticks uno del movimiento del robot y el segundo para la rotación de la base de la cámara para tener una imagen de 360 grados
- ✓ **Verificación del estado de los componentes electrónicos**  
Al encender el robot, el controlador verifica que todos los componentes emitan o reciban datos antes de la ejecución del operador, es indispensable para verificar la integridad y correcto funcionamiento del sistema
- ✓ **Datos del GPS**  
Los datos que el GPS nos proporciona son la hora GMT, longitud, latitud, velocidad en nudos, orientación en grados y la fecha. Estos datos

nos proporcionan la ubicación exacta del robot y es indispensable para establecer la posición exacta del robot cuando encuentra una persona atrapada entre los escombros.

✓ **Señal de encendido**

Orden de encendido emitida del operador hacia el robot.

**SALIDAS:**

✓ **Señal visual captada de las cámaras en el panel de control**

Imagen recibida de las cámaras en el panel de control indispensable para establecer una correcta trayectoria para el desplazamiento del robot y la verificación de personas de acuerdo a su temperatura corporal

✓ **Movimiento del robot**

Acción de movimiento del robot de acuerdo a la orden recibida previamente del joystick del mando de control

✓ **Posicionamiento del GPS**

Datos finales obtenidos del GPS visualizados en la pantalla LCD.

✓ **Rotación de la cámara**

Posicionamiento final de la cámara de acuerdo al joystick del mando de control

#### 4.4 Caja blanca

En la siguiente estructura de funciones descrita de manera gráfica en la figura 36 se representa las funciones principales del robot a diseñar, cumpliendo así con los objetivos del proceso

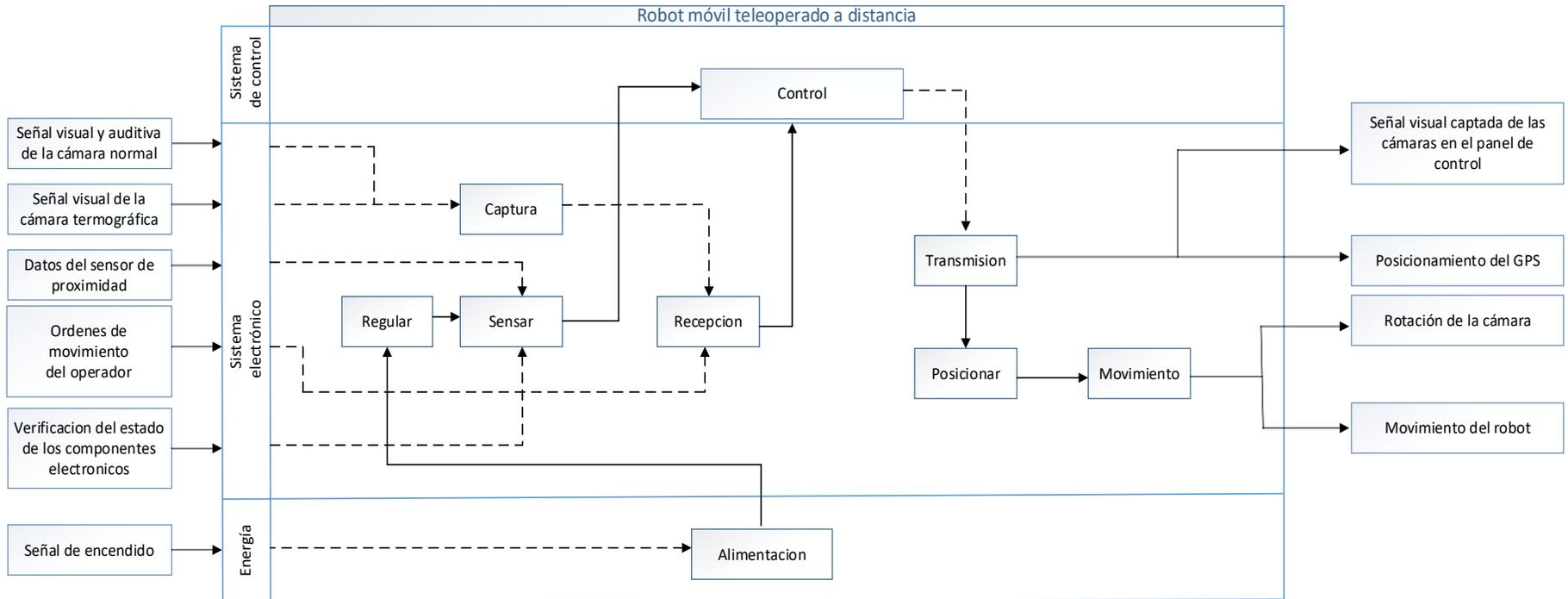
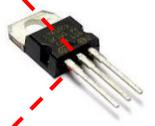
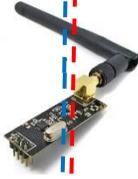


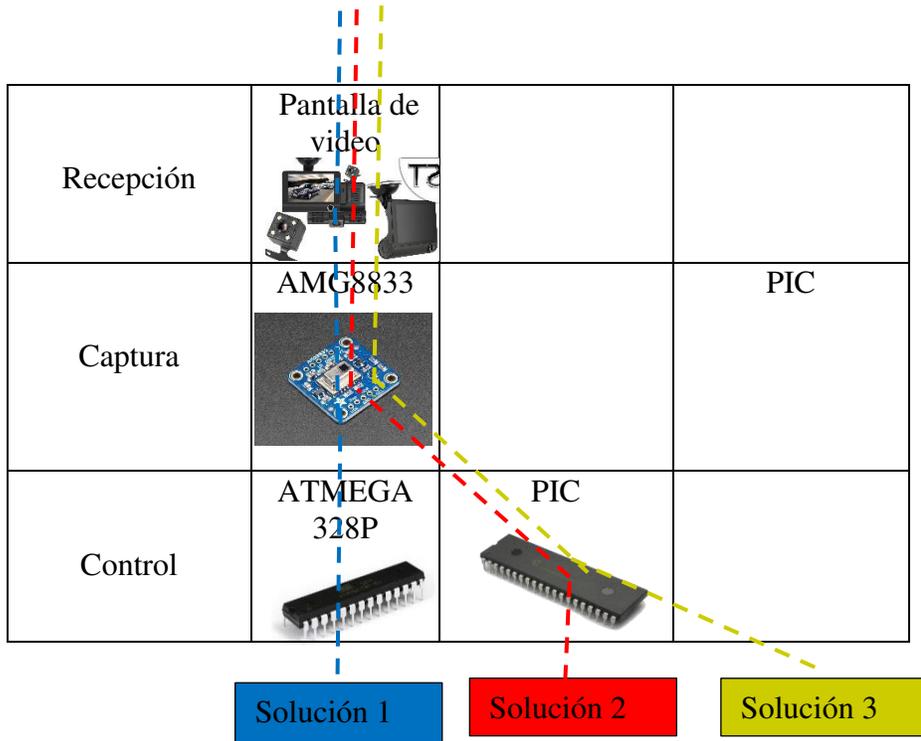
Figura 36. Caja blanca  
Elaboración propia

#### 4.5 Matriz morfológica

La matriz que se visualiza en la tabla 6 conforma algunas alternativas que deberán cumplir las funciones propuestas anteriormente, de esta forma se plantea algunas soluciones para luego analizarlo y encontrar un diseño óptimo del robot oruga de alta tracción.

Tabla 6. Matriz de funciones

FUNCIONES PARCIALES PRINCIPALES	PORTADORES DE FUNCIONES		
Alimentación	Batería 	Solar 	Cable de poder 
Regular	LM2596F 	LM7805 	AMS1117 
Sensor	Ultrasonico 	Fotoelectrico 	
Movimiento	Oruga 	Ruedas 	Patas 
Posicionar	GPS 		
Transmisión	NRF24L01 	Bluetooth HC-06 	

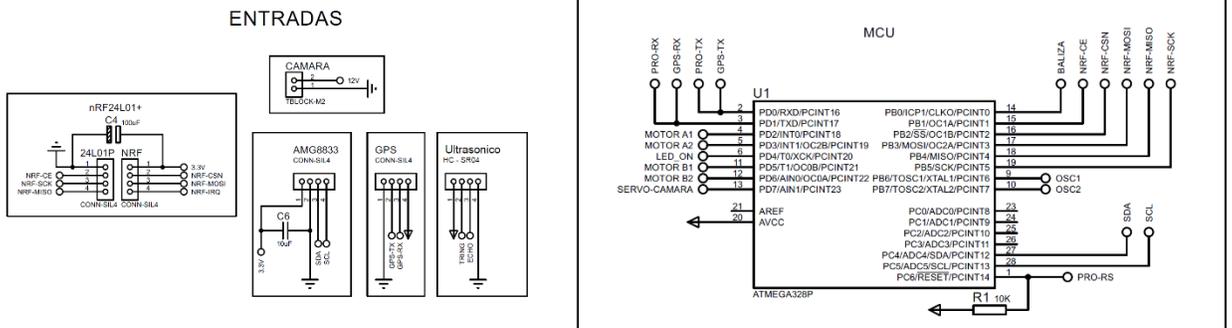


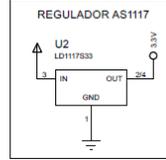
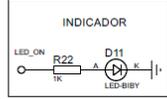
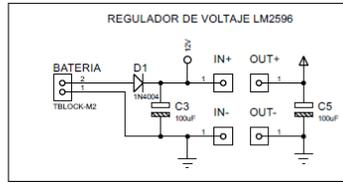
### Concepto de solución 1

Diseño mecánico

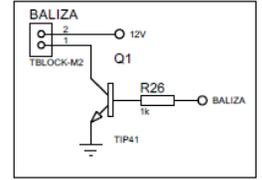
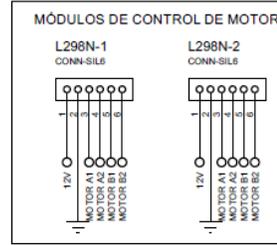
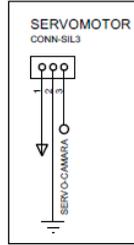


Diseño electrónico





## SALIDAS

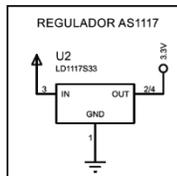
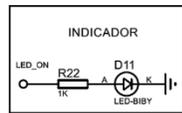
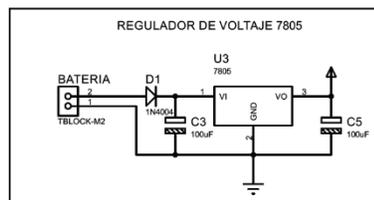
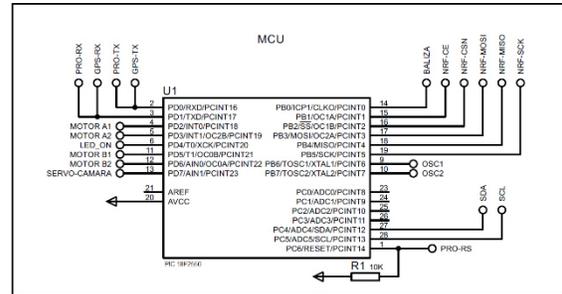
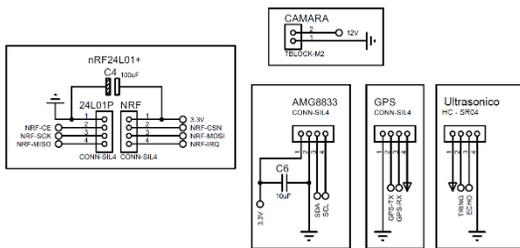


## Concepto de solución 2 Diseño mecánico

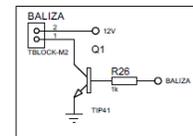
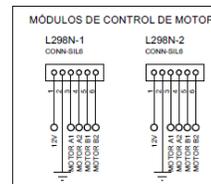
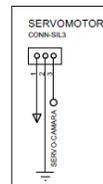


## Diseño electrónico

### ENTRADAS



### SALIDAS



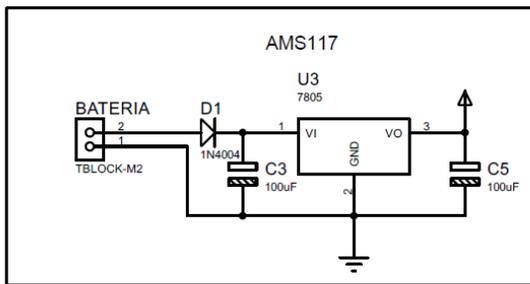
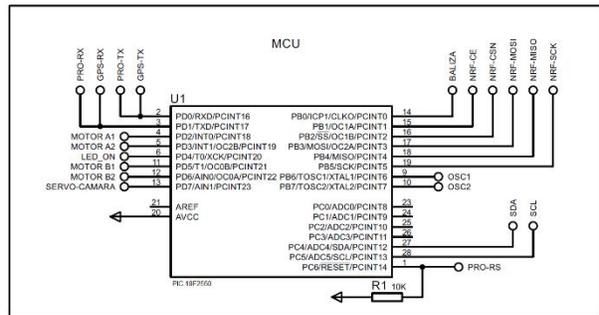
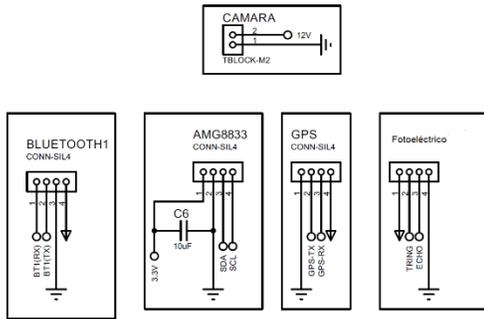
# Concepto de solución 3

## Diseño mecánico

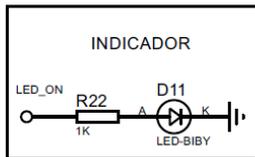
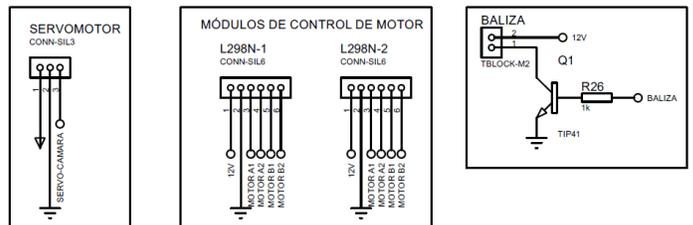


## Diseño electrónico

### ENTRADAS



### SALIDAS



## 4.6 Evaluación técnica – económica del concepto de solución

### 4.6.1 Valoración técnica

Tabla 7. Valoración técnica

Diseño Mecatrónico <b>Valoración técnica (xi)</b>									Área de Diseño		
<b>PROYECTO:</b> Robot móvil teleoperado a distancia											
p: Puntuación de 0 a 4 (Escala de valores según VDI 2206) 0 = Muy bajo, 1 = Bajo, 2 = Medio, 3 = Regular, 4 = Perfecto (ideal)											
g: Peso ponderado de acuerdo a los criterios de evaluación											
Criterios de evaluación para diseños en fase de proyectos o conceptos											
Variantes de Concepto			Solución 1			Solución 2		Solución 3		Sol. Ideal	
			S1			S2		S3		S ideal	
Nº	Criterios de calificación	g	p	gp	p	gp	p	gp	P	gp	
1	Función	8	4	32	2	16	3	24	4	32	
2	Movilidad	6	3	18	3	18	3	18	4	24	
3	Geometría	10	3	30	2	20	3	30	4	40	
4	Mantenimiento	7	3	21	2	14	3	21	4	28	
5	Rapidez	8	3	24	2	16	3	24	4	32	
6	Fabricación	8	3	24	2	16	2	16	4	32	
7	Montaje	7	3	21	2	14	2	14	4	28	
8	Ergonomía	8	3	24	2	16	2	16	4	32	
9	Autonomía	6	3	18	3	18	3	18	4	24	
10	Facilidad de manejo	8	3	24	3	24	2	16	4	32	
11	Estabilidad	7	3	21	2	14	3	21	4	28	
12	Seguridad	6	3	18	3	18	3	18	4	24	
13	Complejidad	6	3	18	2	12	2	12	4	24	
Puntaje $\sum gp$			293			216		248		380	
Valor Técnico xi			0.77			0.57		0.65		1.00	

Elaboración propia

### 4.6.2 Valoración económica

Tabla 8. Valoración económica

Diseño Mecatrónico <b>Valoración económica (yi)</b>									Área de Diseño		
<b>PROYECTO:</b> Robot móvil teleoperado a distancia											
p: Puntuación de 0 a 4 (Escala de valores según VDI 2206) 0 = Muy bajo, 1 = Bajo, 2 = Medio, 3 = Regular, 4 = Perfecto (ideal)											
g: Peso ponderado de acuerdo a los criterios de evaluación											
Criterios de evaluación para diseños en fase de proyectos o conceptos											
Variantes de Concepto			Solución 1			Solución 2		Solución 3		Sol. Ideal	
			S1			S2		S3		S ideal	
Nº	Criterios de calificación	g	p	gp	p	gp	p	gp	P	gp	
1	Costo de mantenimiento	6	3	18	3	18	2	12	4	24	
2	Costo de material	9	3	27	2	18	2	18	4	36	
3	Costo de fabricación	8	3	24	2	16	2	16	4	32	
4	Costo de montaje	6	3	18	3	18	2	12	4	24	
5	Costo de diseño	6	3	18	1	6	3	18	4	24	
Puntaje $\sum gp$			105			76		76		140	
Valor económico yi			0.75			0.54		0.54		1.00	

Elaboración propia

### 4.6.3 Selección de sistema optimo

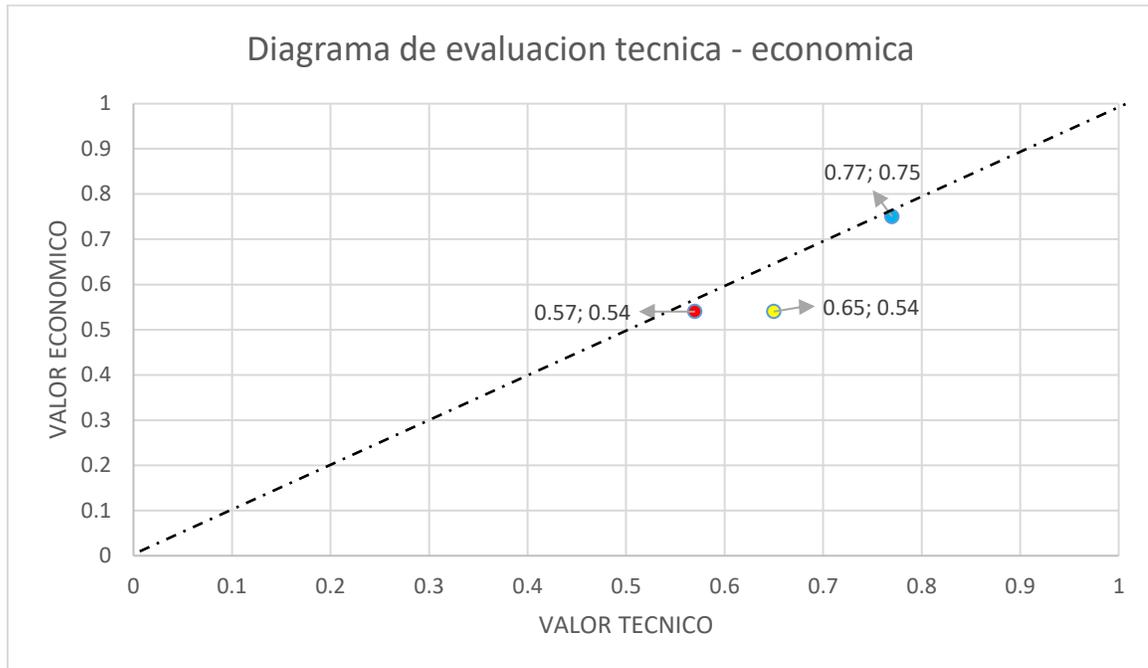


Figura 37. Grafica de evaluación técnica – económica  
Elaboración propia

#### Leyenda:

Solución 1: Color azul \*

Solución 2: Color rojo \*

Solución 3: Color amarillo \*

A partir de la evaluación técnica – económica realizada se concluye que la solución optimo es el concepto de solución numero 1 (0.77, 0.75), porque es más próximo a la recta y a la solución ideal (1,1), como se muestra en la figura 37.

#### La solución 1 de la matriz morfológica se hará con el siguiente procedimiento:

En la figura 38 se puede observar el concepto de solución, para el control del robot se hará uso del microcontrolador ATMEGA 328P por la facilidad de programación y compatibilidad con distintos dispositivos el suministro de energía partirá de una batería de 12V de alto amperaje, para el correcto suministro de voltaje hacia controlador, sensores y otros dispositivos se usará el regulador de voltaje LM2596F que cuenta con un módulo display para la correcta calibración luego se hará uso del sensor ultrasónico para detectar obstáculos mientras se manipula el robot. El sistema de movimiento elegido será el de tipo oruga por su facilidad de movimiento

en terrenos irregulares, alta tracción y costo medio para localizar la posición del robot se hará uso de un módulo GPS. Para la transmisión de datos captados imágenes y video se hará uso del módulo NRF24L01 que puede alcanzar una amplitud de envío y recepción de datos en 1000m a la redonda, para la visualización de imágenes se usará un panel de video que estará incorporado en el mando de control y para la vista termográfica se hará uso del módulo AMG8833 que puede detectar a un humano o cuerpo calorífico desde una distancia de 7 metros.

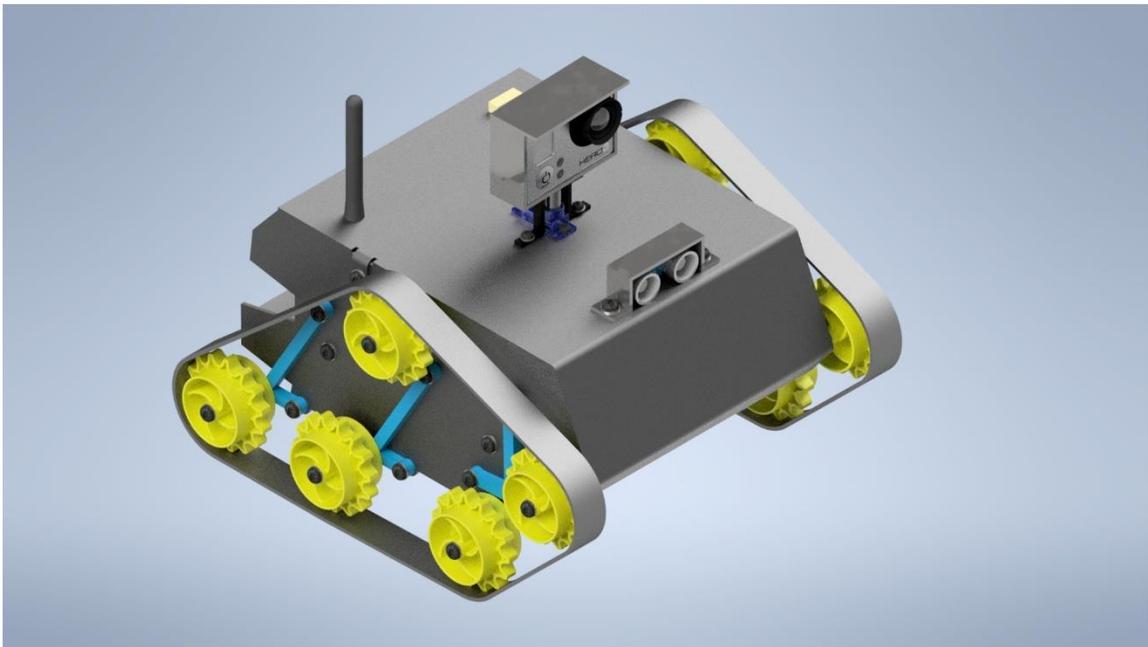


Figura 38. Concepto de solución  
Elaboración propia

## 4.7 Diseño mecánico

### 4.7.1 Sistema de movimiento

Para seleccionar adecuadamente el sistema de movimiento se compara las ventajas y desventajas de 3 tipos de sistemas de movimiento

Tabla 9. Sistema de movimiento por ruedas

<b>Ventajas</b>	<b>Desventajas</b>
Fácil implementación	Son resbaladizos en terrenos blandos
Velocidad en terrenos regulares	Dificultad de desplazamiento en terrenos irregulares
Facilidad de mantenimiento	
Facilidad de control	

Elaboración propia

Tabla 10. Sistema de movimiento oruga

<b>Ventajas</b>	<b>Desventajas</b>
Mejor adaptabilidad a terrenos irregulares	Baja velocidad
Costo medio	Dificultad moderada de implementación
Alta tracción	
Fácil de controlar	

Elaboración propia

Tabla 11. Sistema de movimiento por patas

<b>Ventajas</b>	<b>Desventajas</b>
Alta capacidad de moverse en terrenos irregulares	Implementación compleja
Puede subir escalones o superficies pequeñas	Alto costos
	Alto consumo energético
	Necesita un buen control para mantener la estabilidad

Elaboración propia

De acuerdo a las tablas 9, 10 y 11 se puede determinar que el sistema que se adecua a nuestro propósito es el sistema de movimiento oruga porque las condiciones del terreno de operación son accidentados y necesitamos una alta tracción.

#### 4.7.2 Estructura

Los materiales deben ser resistentes para soportar los impactos a la estructura además de proteger los dispositivos electrónicos.

Tabla 12. Tipos de materiales

Material	Ventajas	Desventajas
Acero	<ul style="list-style-type: none"> <li>➤ Alta resistencia</li> <li>➤ Capacidad de deformación</li> <li>➤ Fácil adquisición</li> <li>➤ Corrosión al medio ambiente</li> <li>➤ Variedad de aplicaciones de acuerdo a la aleación</li> </ul>	<ul style="list-style-type: none"> <li>➤ Alta densidad</li> <li>➤ Alto costo en elaciones especiales</li> <li>➤ Deformaciones irreversibles</li> </ul>
Acrílico	<ul style="list-style-type: none"> <li>➤ No es corrosivo al medio ambiente</li> <li>➤ Resistencia química</li> <li>➤ Densidad baja</li> <li>➤ Diseños específicos con corte</li> <li>➤ Dureza</li> </ul>	<ul style="list-style-type: none"> <li>➤ Alta fragilidad</li> <li>➤ Riesgo de abrasión</li> </ul>
Aluminio	<ul style="list-style-type: none"> <li>➤ Ligero</li> <li>➤ Resistencia mecánica media</li> <li>➤ Facilidad de trabajos en forja</li> <li>➤ Baja corrosión al medio ambiente</li> <li>➤ Bajo costo</li> </ul>	<ul style="list-style-type: none"> <li>➤ Susceptible a rayones</li> <li>➤ Corrosión galvánica al entrar en contacto con otros materiales</li> <li>➤ Deformaciones no reversibles</li> </ul>
ABS	<ul style="list-style-type: none"> <li>➤ Creación de piezas de alta complejidad geométrica</li> <li>➤ Baja densidad</li> <li>➤ Resistente a la corrosión</li> </ul>	<ul style="list-style-type: none"> <li>➤ Fácil desgaste a la abrasión</li> <li>➤ Costo elevado</li> <li>➤ Soporta temperaturas de -20 °C a 80 °C</li> </ul>

Elaboración propia

De acuerdo al análisis de la tabla 12 se puede determinar que el material a usar para la estructura será aluminio por la mediana resistencia mecánica, ligereza y bajo costo.

#### 4.7.3 Motores

Los motores proporcionaran el desplazamiento al robot y posicionamiento de las cámaras. Solo se tomarán en cuenta los motores DC ya que fueron de gran utilización en la robótica.

Tabla 13. Motor paso a paso

Ventajas	Desventajas
Exactitud y precisión en giros	Sufre calentamiento en el bobinado del motor
Facilidad de control	Implementación complicada
Control de velocidad	Baja velocidad

Elaboración propia

Tabla 14. Motor DC con caja reductora

Ventajas	Desventajas
Fácil control	No es exacto
Control de velocidad	No tiene precisión
Fácil implementación	Difícil de mantener determinada posición
Se puede determinar el estado y la posición	
Alta tracción	

Elaboración propia

Tabla 15. Servomotor

Ventajas	Desventajas
Fácil control	Es preciso
Fácil implementación	Costo medio
Variedad de tamaños	Dificultad de adquisición
Puede mantener una posición específica	
Capacidad de mantener una posición específica	

Elaboración propia

De acuerdo a los datos de las tablas 13, 14 y 15 se puede determinar que los motores para usar en el sistema oruga serán del tipo Motor DC con caja reductora por la fuerza de tracción, fácil control y la capacidad de terminar el estado y para el movimiento de la estructura de la cámara se usara un servomotor por la capacidad de precisión en giro y la capacidad de mantener una posición específica.

## 4.8 Diseño electrónico

### 4.8.1 Controlador

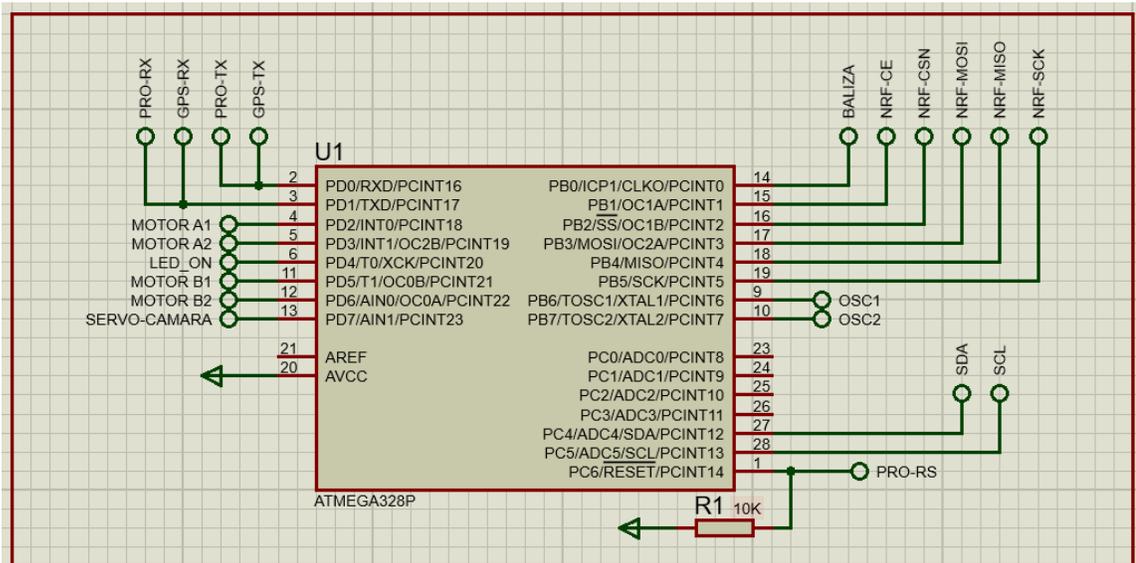


Figura 39. Controlador en placa de control

Elaboración propia

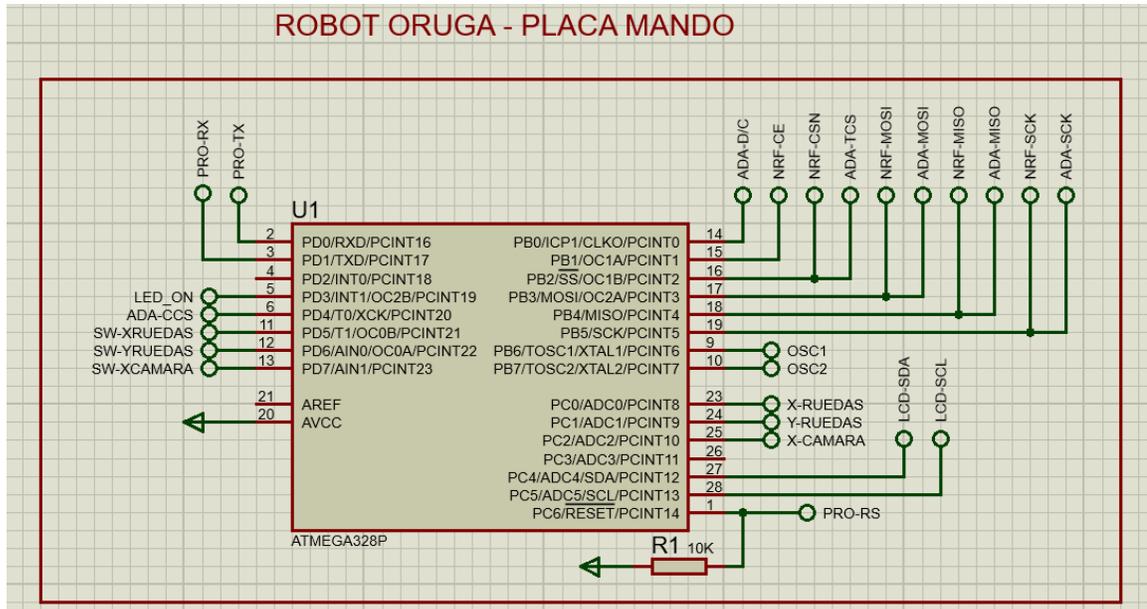


Figura 40. Controlador en placa de mando  
Elaboración propia

El controlador seleccionado para nuestro robot es el microcontrolador ATMEGA328P que tiene una plataforma de Hardware libre que permite la facilidad de programación e interacción de diversos dispositivos electrónicos siendo la herramienta preferida para usuarios inexpertos o expertos en conocimiento de electrónica. El microcontrolador se encargará de analizar e interpretar los datos de entrada y salida de los dispositivos electrónicos implementados como el sensor de proximidad, la cámara, los joysticks, los motores etc. En la tabla 16 se muestra las conexiones que se encuentran en el microcontrolador.

Tabla 16. Conexiones ATMEGA328P

Localizacion	Dispositivo	Numero de pines
Placa de control	NRF24L01	5
	GPS	2
	Servomotor	1
	Motor DC con caja reductora	8
	Led de verificación	1
Placa de control	NRF24L01	5
	Joystick movimiento	4
	Joystick cámara	2
	Led de verificación	1
	LCD	2

Elaboración propia

De acuerdo al número de pines I/O de la figura 39 y 40 en cada placa el microcontrolador ATMEGA328P puede abastecer sin ningún problema los dispositivos electrónicos enlazados

#### 4.8.2 Control de motores

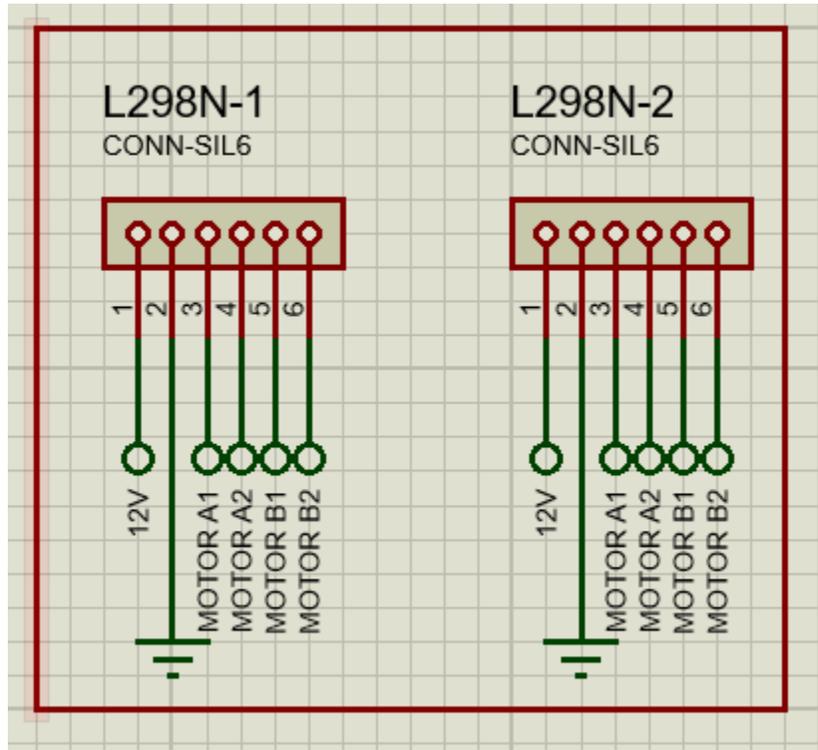


Figura 41. Circuito de potencia de motor  
Elaboración propia

En la figura 41 se muestra que nuestro circuito de potencia está conformado por dos módulos L298N que controlan la velocidad y torque del motor DC con caja reductora. Las funciones del circuito de potencia se pueden observar en la tabla 17.

Tabla 17. Funciones de circuito de potencia

Pines	Funciones
12v	Alimentación
MOTORA1-1	Entrada digital para control de movimiento
MOTORA2-1	Entrada digital para control de velocidad
MOTORB1-1	Salida eléctrica para movimiento horario con PWM
MOTORB2-2	Salida eléctrica de movimiento antihorario con PWM

Elaboración propia

### 4.8.3 Selección de sistema de comunicación inalámbrica

El sistema de comunicación del robot permite recibir órdenes de movimiento o tareas que solicite el operador. Además, de enviar datos como localización del robot, imágenes de cámaras o lecturas de los sensores. En la tabla 18 se muestra los datos del ancho de banda y alcance de algunos sistemas de redes inalámbricas más usados para un robot.

Tabla 18. Características de comunicación inalámbrica

<b>Tipo</b>	<b>Alcance</b>	<b>Ancho de banda</b>
<b>Bluetooth</b>	10m	0.5 – 600 Mbps
<b>Wifi</b>	100m	2 – 54 Mbps
<b>Radiofrecuencia</b>	1000m	250 kbps – 10 Mbps

Elaboración propia

## CAPITULO V

### SIMULACIÓN

Todo el proceso de simulación se realizará en el software Proteus.

#### 5.1 Simulación de procesos

Al encender el robot lo primero que observaremos será un mensaje de inicialización en el LCD como se observa en la figura 42.

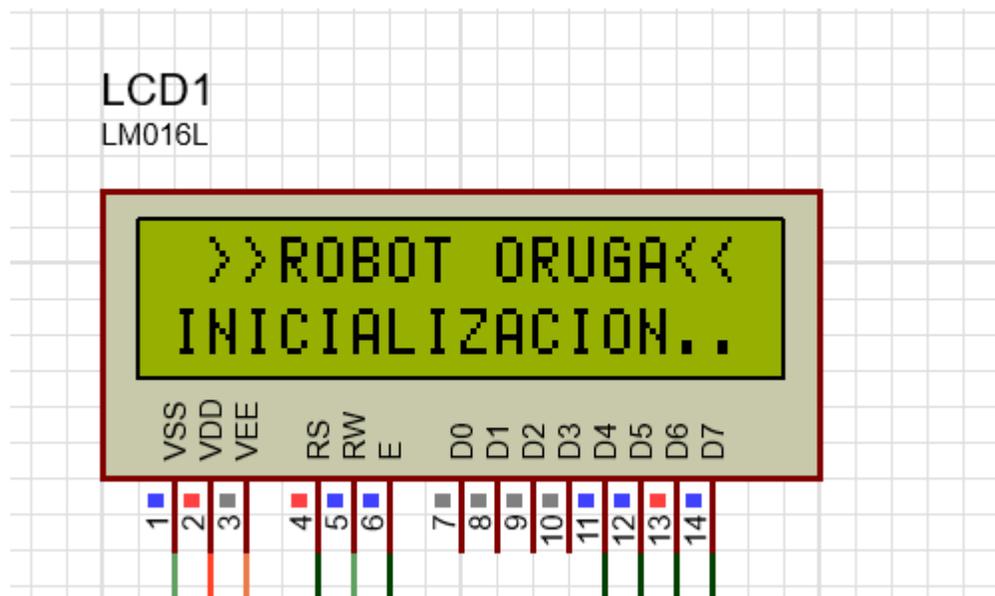


Figura 42. Inicialización del robot  
elaboración propia

Luego verificara la conexión entre el mando de control y el robot oruga como se observa en la figura 43.

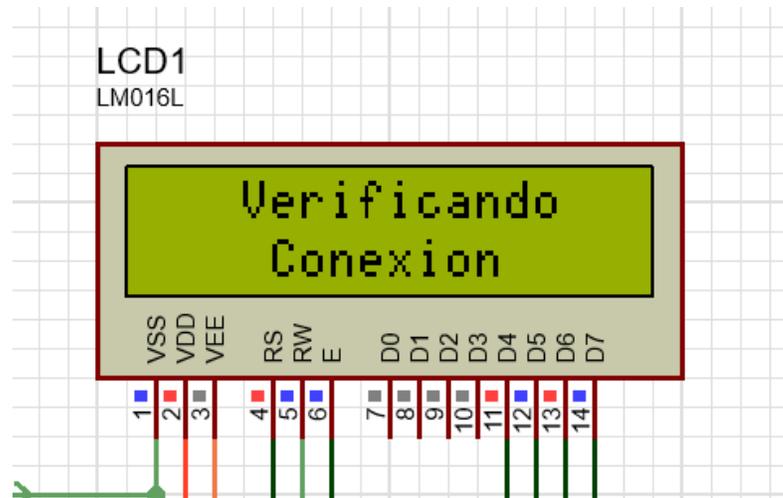


Figura 43. Verificación de conexión  
elaboración propia

Si la conexión es satisfactoria aparecerá el siguiente mensaje de verificación como se observa en la figura 44.

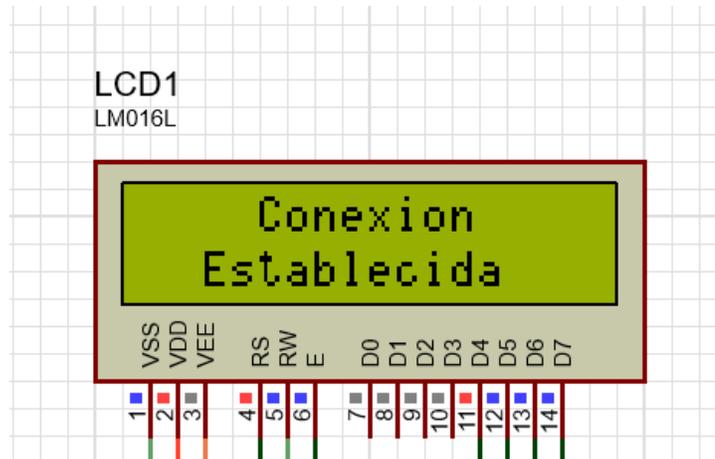


Figura 44. Conexión establecida entre el robot oruga y la estación de mando  
elaboración propia

En caso contrario aparecerá el siguiente mensaje como se observa en la figura 45. Que significa que no hay conexión entre el robot oruga y el mando de control.

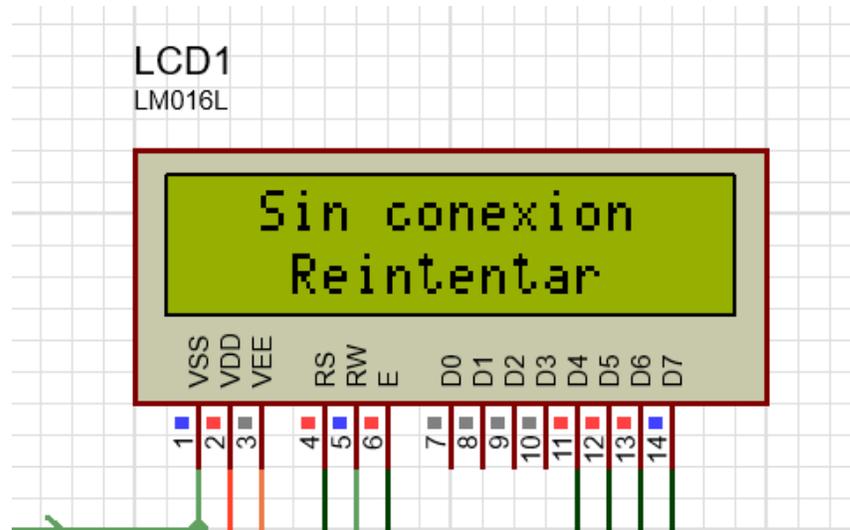


Figura 45. Error de conexión  
elaboración propia

Una vez verificado el enlace entre el robot oruga y el mando de control se verificará el estado del sensor ultrasónico como se observa en la figura 46.

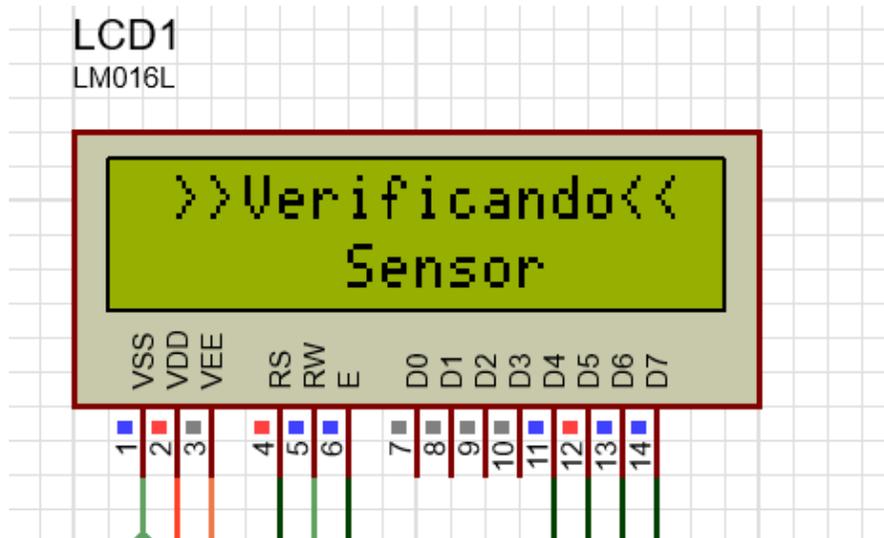


Figura 46. Verificando sensor ultrasónico  
elaboración propia

Si el sensor ultrasónico recibe datos y esta correctamente instalado, en el LCD aparecerá el siguiente mensaje estado: correcto como se observa en la figura 47.



Figura 47. Correcto estado del sensor ultrasónico  
elaboración propia

Por otro lado, si el sensor no recibe datos o esta desconectado aparecerá en la pantalla LCD el siguiente mensaje estado: corregir como se observa en la figura 48.

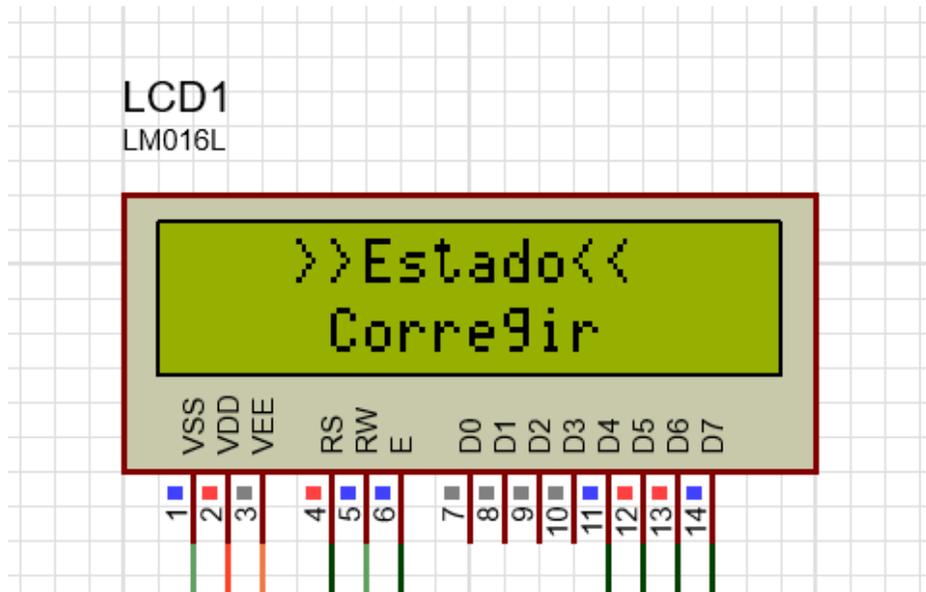


Figura 48. Revisar sensor ultrasónico  
elaboración propia

Una vez verificado el sensor ultrasónico, el controlador pasara a verificar la señal del GPS como como se observa en la figura 49.

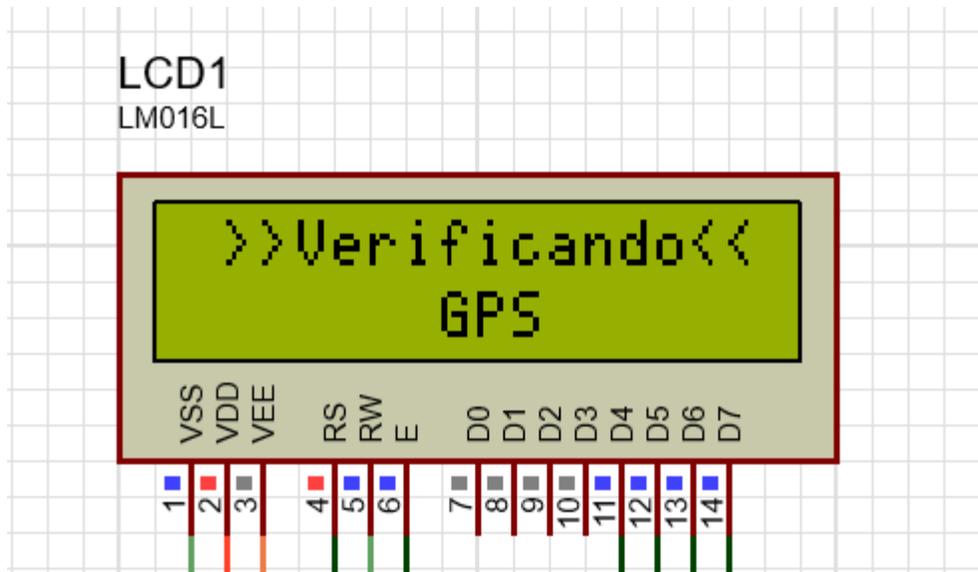


Figura 49. Verificación de GPS  
elaboración propia

Si el GPS está recibiendo y transfiriendo datos aparecerá el siguiente mensaje en el LCD como se observa en la figura 50.

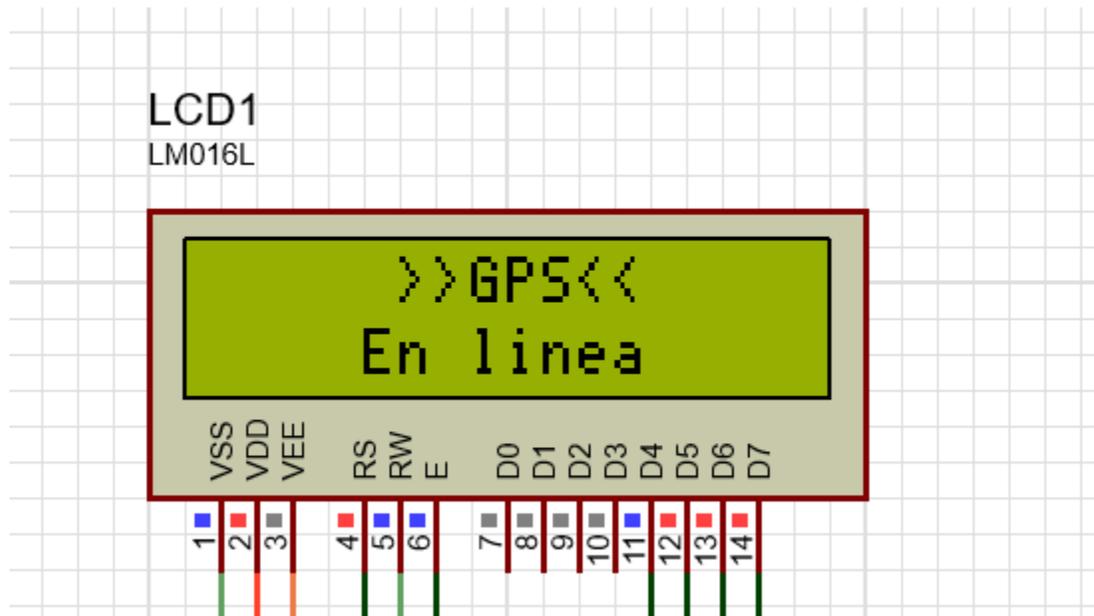


Figura 50. GPS en línea  
elaboración propia

Si el GPS no recibe ningún dato o no transfiere al mando de control aparecerá el siguiente mensaje en el LCD como se observa en la figura 51.

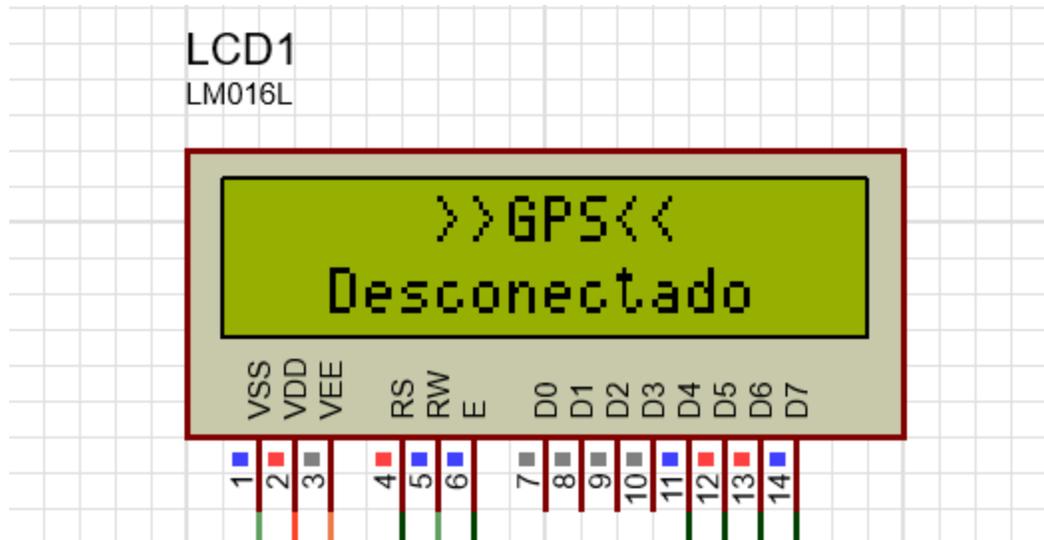


Figura 51. GPS desconectado  
elaboración propia

Una vez el GPS este en línea empezara a enviar los datos en tiempo real como se observa en las figuras 52 y 53.

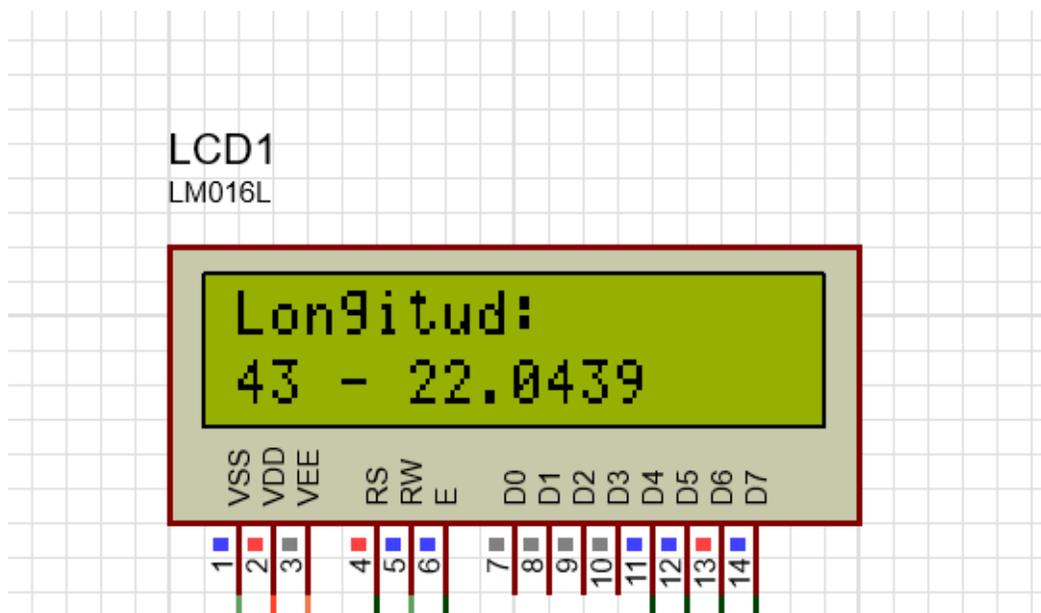


Figura 52. Longitud obtenida de GPS  
elaboración propia

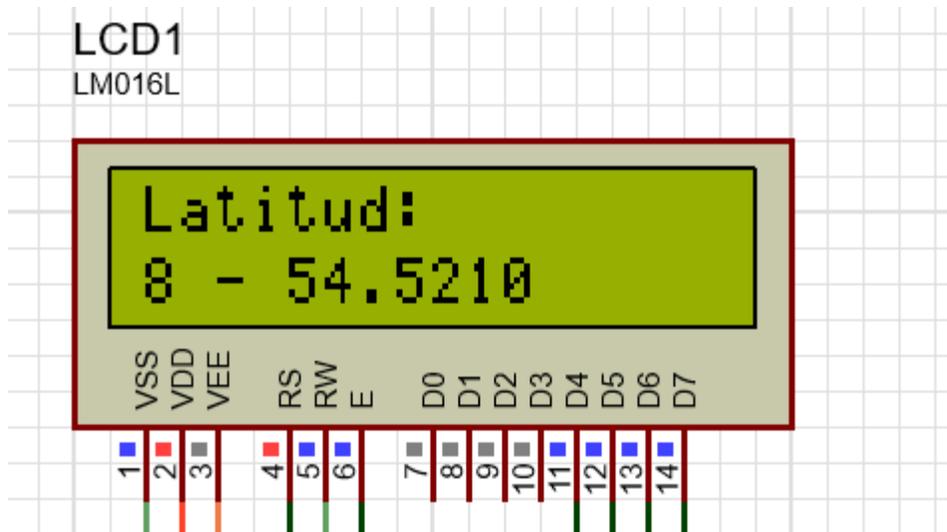


Figura 53. Latitud obtenida de GPS  
elaboración propia

Todos estos datos son el sistema de seguridad para utilizar el robot oruga de alta tracción. Cuando ya se está operando el robot y se encuentra con un objetivo que tiene una pendiente muy elevada y no puede escalar aparecerá los siguientes mensajes como se observa en la figura 54.

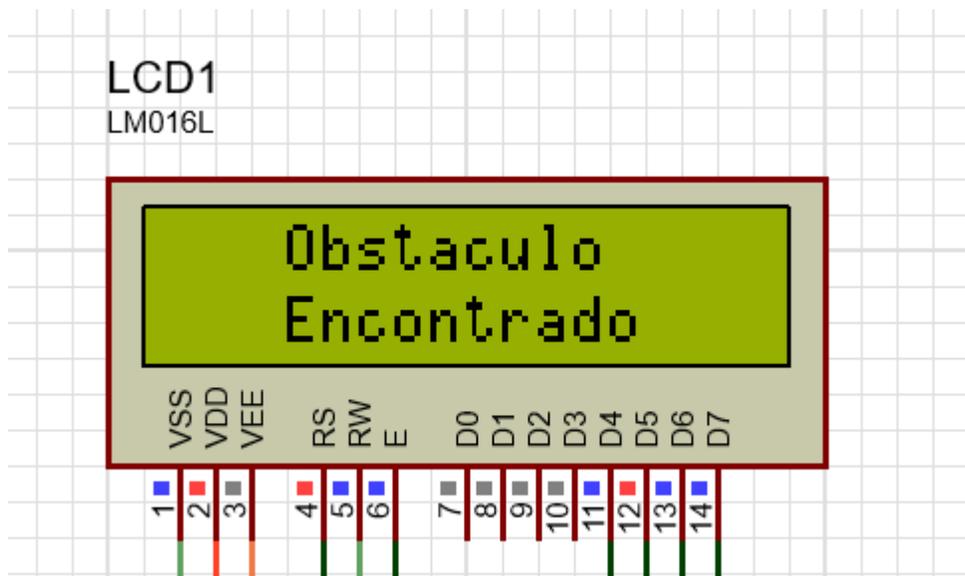


Figura 54. Detección de obstáculo mediante sensor ultrasónico.  
elaboración propia

Si te acercas aún más al obstáculo aparecerán mensajes de alerta como se observa en las figuras 55 y 56.

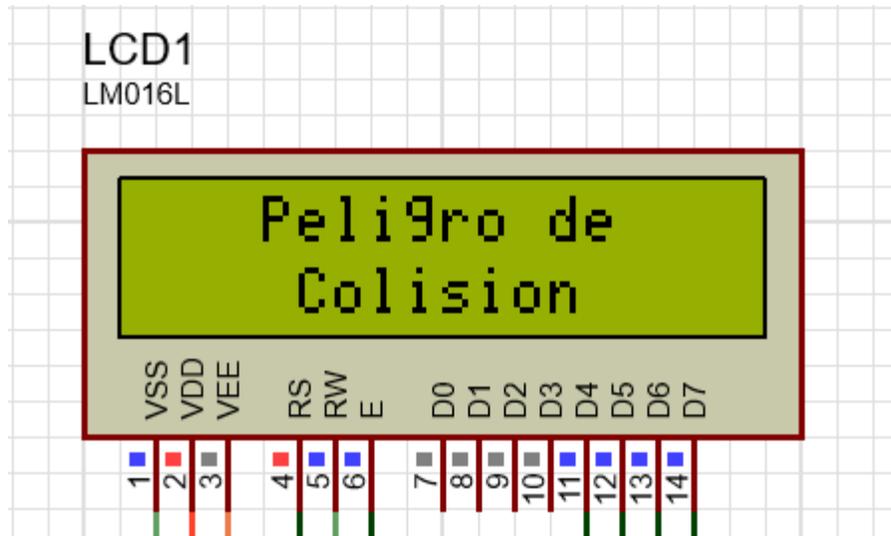


Figura 55. Mensaje de alerta objetivo a 30cm.  
Elaboración propia

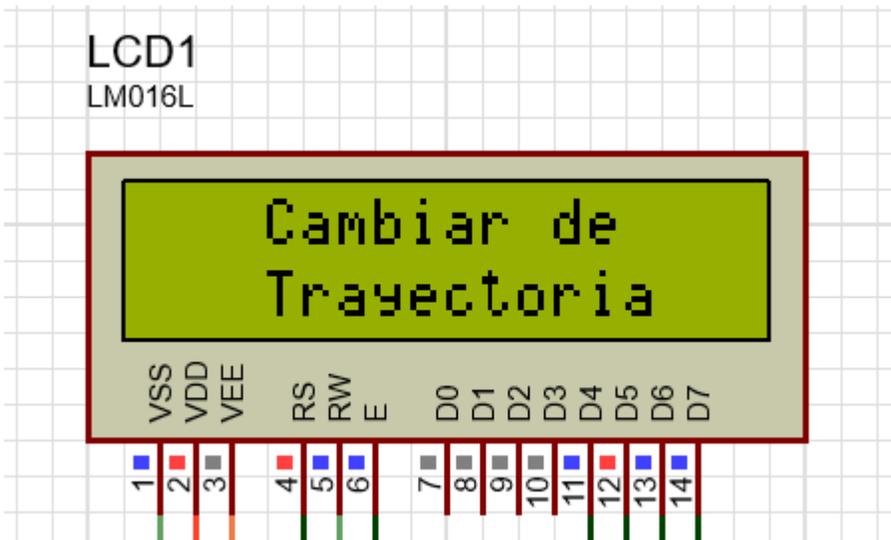


Figura 56. Mensaje de alerta objetivo a 15cm.  
Elaboración propia



### 5.2.2 Placa de mando en PCB

Diseño de placa de mando (PCB) en software Proteus

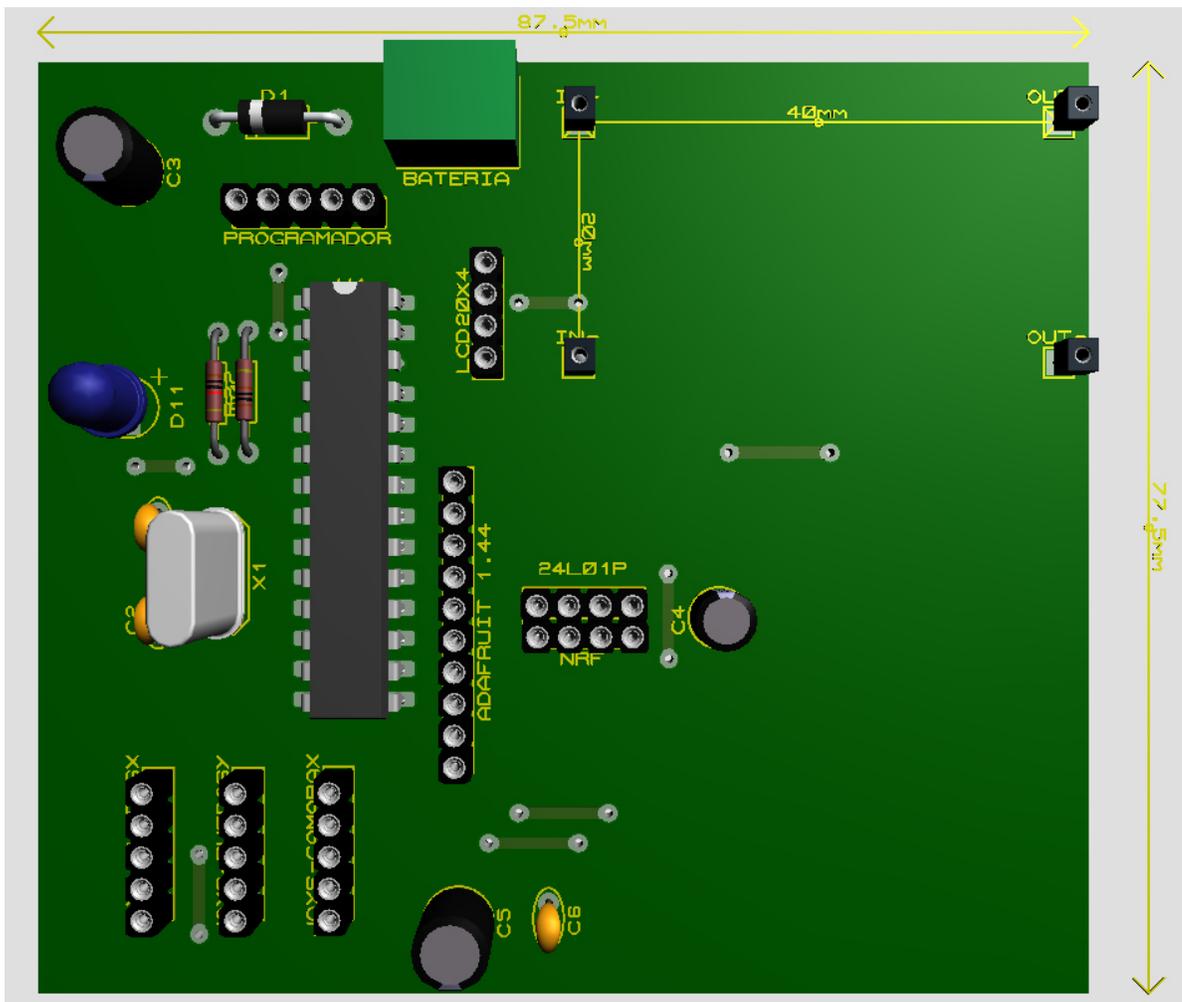


Figura 58. Placa de mando PCB  
Elaboración propia

En la figura 58 se muestra el diseño final de la placa de control donde se ubican todos los componentes electrónicos distribuidos correctamente para el correcto funcionamiento del robot oruga de alta tracción.

### 5.2.3 Modelamiento de prototipo final

Diseño de prototipo final en el software Inventor

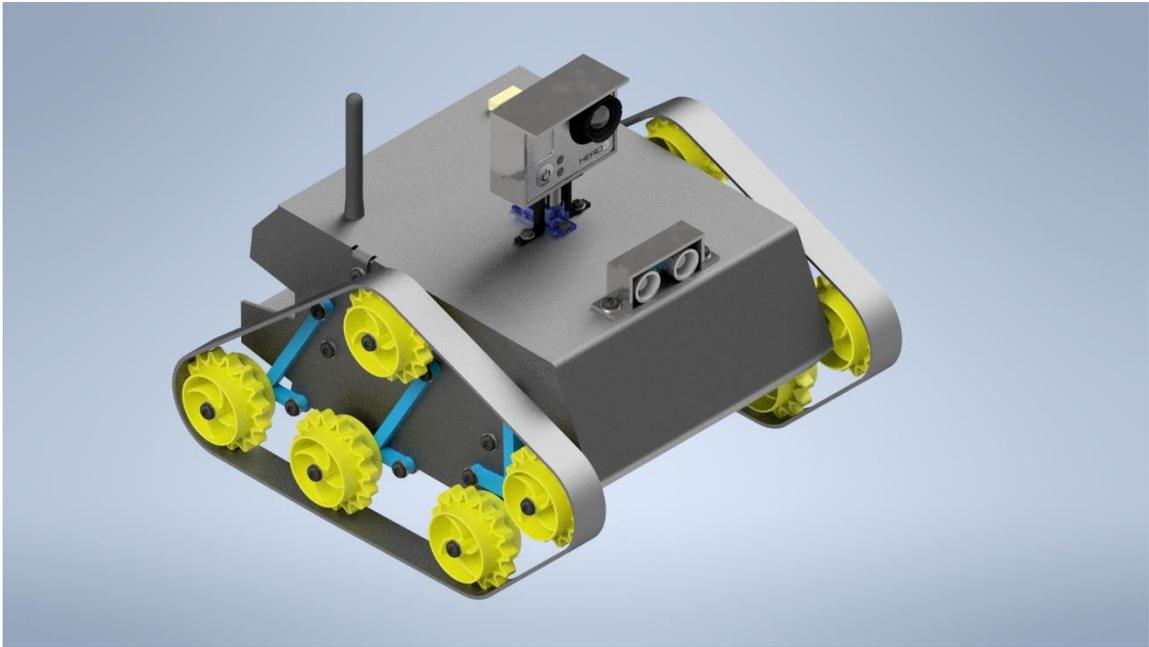


Figura 59. Renderizado final del prototipo  
Elaboración propia

En la figura 59 se puede visualizar el modelado final de robot oruga con todos los parámetros previamente establecidos.

## CONCLUSIONES

- El diseño del robot móvil teleoperado a distancia para rastreo de personas en un derrumbe de una edificación ayudara a los bomberos y rescatistas en sus labores de riesgo localizar personas desde un lugar seguro controlando el robot y evitando exponer su integridad física y la de las personas que están siendo rescatadas.
- El diseño adecuado para el sistema de movimiento del robot móvil será el de tipo oruga por su fuerza de tracción y capacidad de movimiento en terrenos accidentados.
- La tecnología de comunicación inalámbrica que permite el enlace entre el robot móvil y el operador será mediante radio frecuencia utilizando un módulo RFI24L01 que permite un alcance de transmisión de datos de 1000 metros a la redonda que aumenta el rango de trabajo del robot móvil.
- El método para encontrar identificar a las personas en un derrumbe de una edificación será mediante una cámara montada en el robot móvil y una cámara termográfica AMG8833 que puede detectar a una persona hasta en una distancia de 7 metros.

## **TRABAJOS FUTUROS**

- Se recomienda que el robot tenga un espacio para kits de primeros auxilios para que si una víctima necesita alguna intervención rápida puedan los rescatistas auxiliarla en el momento.
- Se recomienda que el diseño del chasis del robot oruga tenga una forma ovalada para distribuir la fuerza de los posibles impactos.
- Se recomienda instalar una cámara termográfica de mayor calidad para ampliar el alcance de inspección del robot.
- Se recomienda tener baterías de repuesto para incrementar la autonomía del robot.
- Se recomienda conseguir un panel de control avanzado para aumentar las funciones del robot.

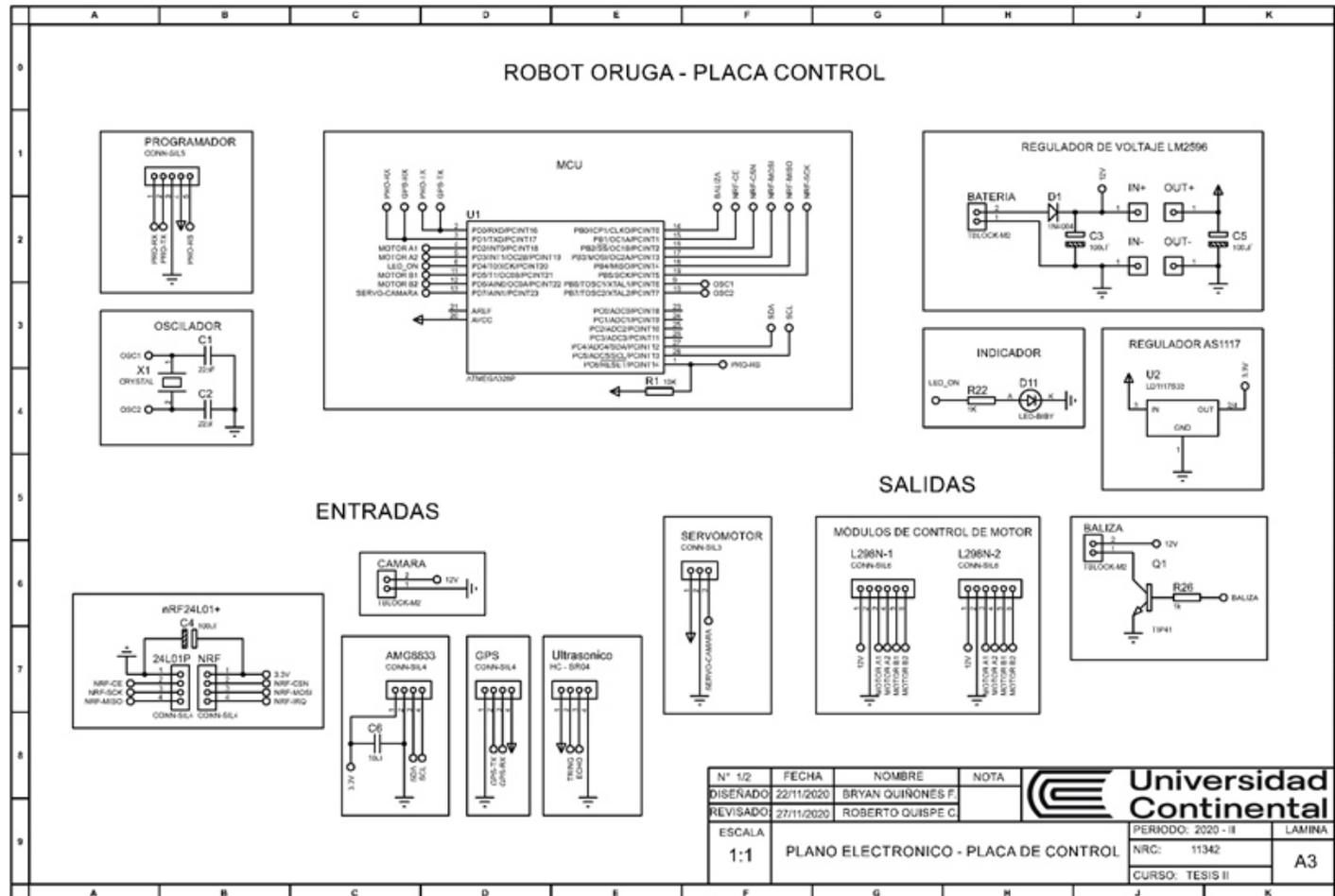
## REFERENCIAS BIBLIOGRÁFICAS

1. **Tavera, Hernando.** Perú, un país altamente sísmico. *Sociedad Geológica del Perú.* [En línea] [Citado el: 2019 de Julio de 2020.] <https://www.sgp.org.pe/alerta-peru-un-pais-altamente-sismico/>.
2. **J., Perez.** AC ARQUITECTOS. [En línea] CEA, 2019 de Mayo de 18. [Citado el: 2020 de Setiembre de 19.] <https://acarquitectos.com.pe/wonderful-serenity-has-taken/>.
3. **Tavera, Hernando, Bernal, Isabel y Salas, Henry.** *El Sismo de Pisco del 15 de Agosto, 2007 (7.9 Mw).* Pisco : s.n., 2007.
4. **Méndez, Karla.** Este robot ayudó a sacar vivas de entre escombros a 11 personas. *EXCELSIOR.* [En línea] 23 de Septiembre de 2017. [Citado el: 2020 de Mayo de 26.] <https://www.excelsior.com.mx/nacional/2017/09/27/1191152#imagen-3>.
5. **Carlos del Valle, Jose.** Antamina. [En línea] 25 de Febrero de 2019. [Citado el: 28 de Abril de 2020.] <https://www.antamina.com/noticias/innovacion-drones-antamina-caza-eficiencia/>.
6. **Sanchez, Luis.** *Desarrollo de una plataforma robotica movil para la busqueda posterior a un sismo, y mitigacion de incendios de baja escala.* Escuela Superior Politecnica de Chimborazo, Chimborazo : 2017.
7. **Mite Labre, Jorge Rainier y Uquillas Mendoza, Luis Mendoza.** *Diseño e implementación del prototipo de un robot para exploración de edificaciones y localización de personas en casos de derrumbes.* Escuela Superior Politecnica del Litoral, Guayaquil : 2016.
8. **Gallegos Correa, Alejandro José.** *Diseño e implementación de un prototipo robótico móvil teleoperado inalámbricamente para inspección en lugares de alto riesgo.* Universidad de las Fuerzas Armadas ESPE, Sangolquí : 2018.
9. **Choquehuanca Huaracallo, Ronald y Llaiqui Lobon, Luis Miguel.** *Robot móvil para detección de personas en lugares inaccesibles frente a desastres naturales en la ciudad Arequipa: caso sismos.* Universidad Católica de Santa María, Arequipa : 2020.
10. **Postigo Huanqui, Sergio Renato.** *Vehículo aéreo no tripulado para vigilancia en ambientes cerrados con detección de personas y obstáculos a su alrededor.* Pontifica Universidad Católica del Perú, Lima : 2018.
11. **Sotelo Gomez, Anthony.** *Robot explorador con realidad virtual para la seguridad en colegios estatales.* Universidad Privada del Norte, Trujillo : 2019.
12. **Sosa Reyna, Claudia M.** La robótica presente en la Uacsyt. *CienciaUAT.* [En línea] 2 de enero de 2006. [Citado el: 2 de octubre de 2020.] <https://www.redalyc.org/articulo.oa?id=441942906014>. ISSN: 2007-7521.
13. **Ollero Baturone, Anibal.** *Robótica, manipuladores y robots móviles.* Sevilla : marcotombo, 2001. ISBN:84-267-1313-0.
14. **Ogata, Katsuhiko.** *Ingeniería de control moderna.* [ed.] Miguel Martin Romo. 5ta edicion. Madrid : Prentice-hall, 2010. ISBN:978-84-8322-660-5.
15. **Kienitz, Ulrich.** Optris infrared measurements. [En línea] Optris, 12 de Enero de 2017. [Citado el: 1 de octubre de 2020.] <https://www.optris.es/imagenes-termicas>.
16. **GEEKBOT ELECTRONICS.** Motores DC. [En línea] 21 de Abril de 2015.

17. **Electronilab.** *Tutorial: Uso de Driver L298N para motores DC y paso a paso con Arduino.* [En línea] 17 de Mayo de 2014.
18. **ConceptoDefinición.** Programación Informática. [En línea] 4 de enero de 2016. [Citado el: 5 de octubre de 2020.] <https://conceptodefinicion.de/programacion-automatizada/>.
19. **tecnología, Area.** Area tecnologia. [En línea] 12 de Diciembre de 2015. [Citado el: 1 de Octubre de 2020.] <https://www.areatecnologia.com/informatica/tecnologia-inalambrica.html>.
20. **Vargas Cordero, Zoila Rosa.** LA INVESTIGACIÓN APLICADA: UNA FORMA DE CONOCER LAS REALIDADES CON EVIDENCIA. [En línea] 15 de Diciembre de 2009. [Citado el: 1 de Octubre de 2020.] [https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiopO3h\\_qrsAhVuD7kGHcppCq0QFjACegQIAhAC&url=https%3A%2F%2Fwww.redalyc.org%2Fpdf%2F440%2F44015082010.pdf&usg=AOvVaw3aGcta2H5YmS-6PEAka3k0](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiopO3h_qrsAhVuD7kGHcppCq0QFjACegQIAhAC&url=https%3A%2F%2Fwww.redalyc.org%2Fpdf%2F440%2F44015082010.pdf&usg=AOvVaw3aGcta2H5YmS-6PEAka3k0).
21. **Investigadores.** Investigacion cientifica. [En línea] 15 de Abril de 2015. [Citado el: 1 de Octubre de 2020.] <https://investigacioncientifica.org/alcance-la-investigacion-cientifica/>.

# ANEXOS

## Plano electrónico de placa de control





# Plano General

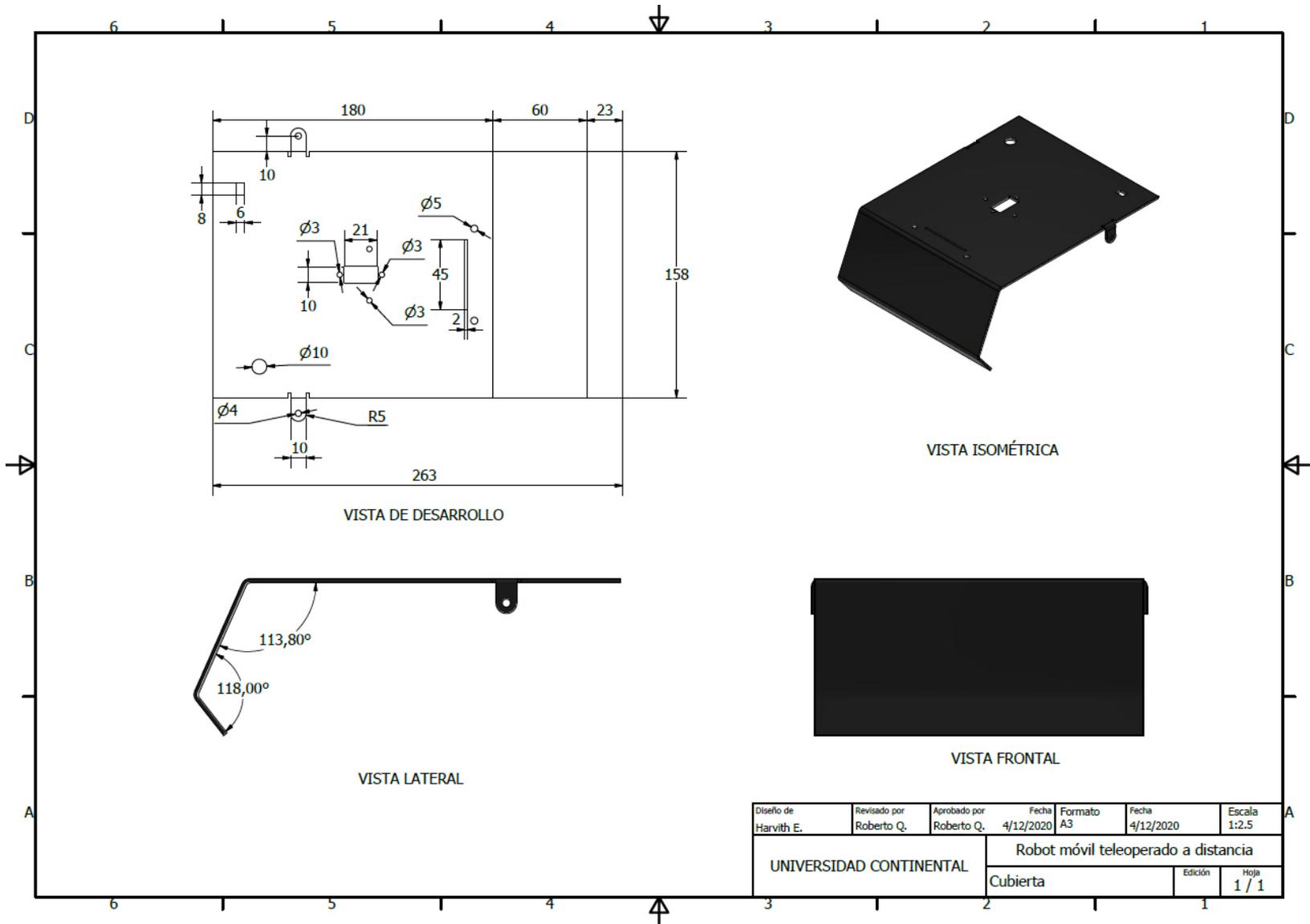
**VISTA FRONTAL**

**VISTA ISOMÉTRICA**

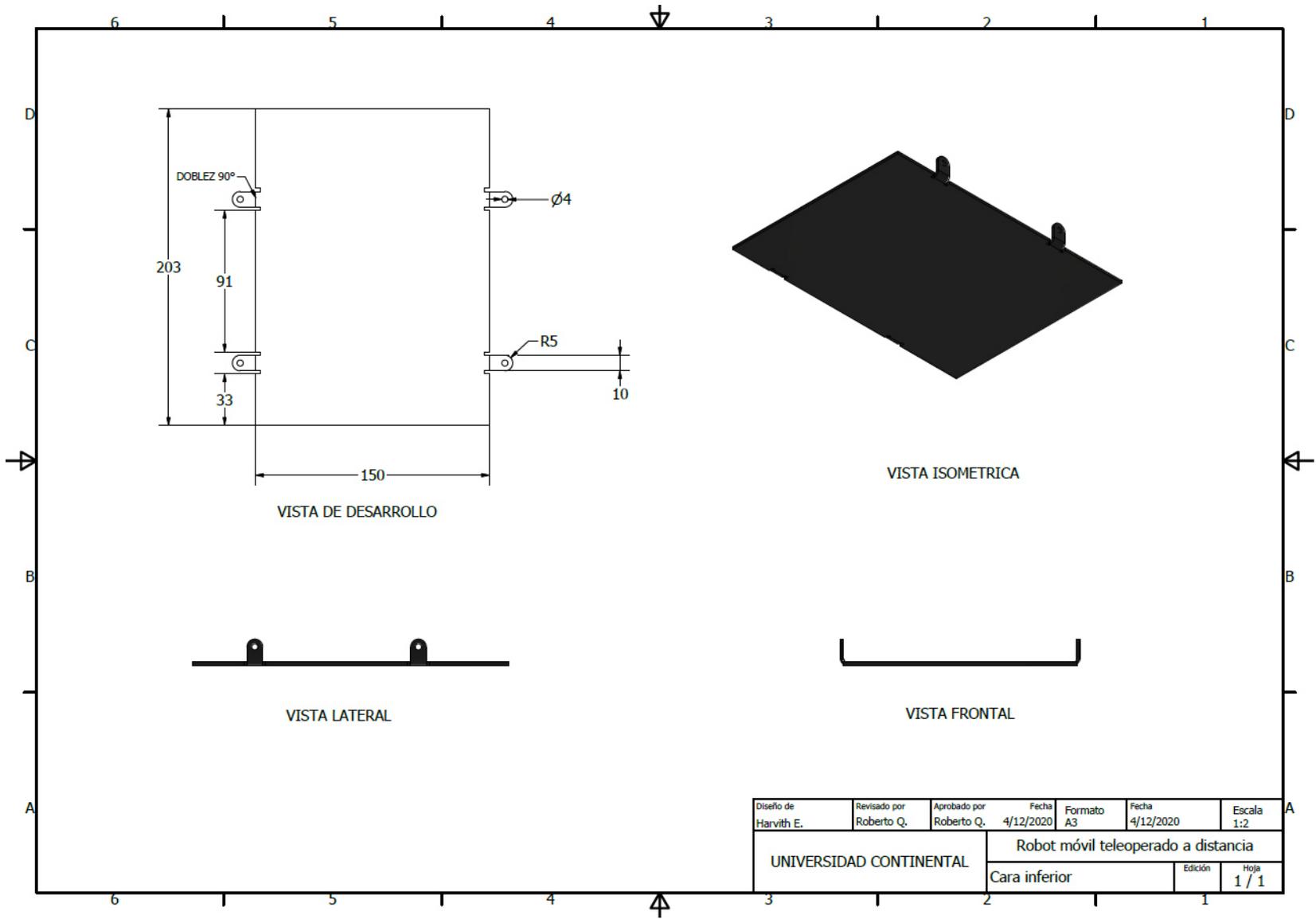
**EXPLOSION**

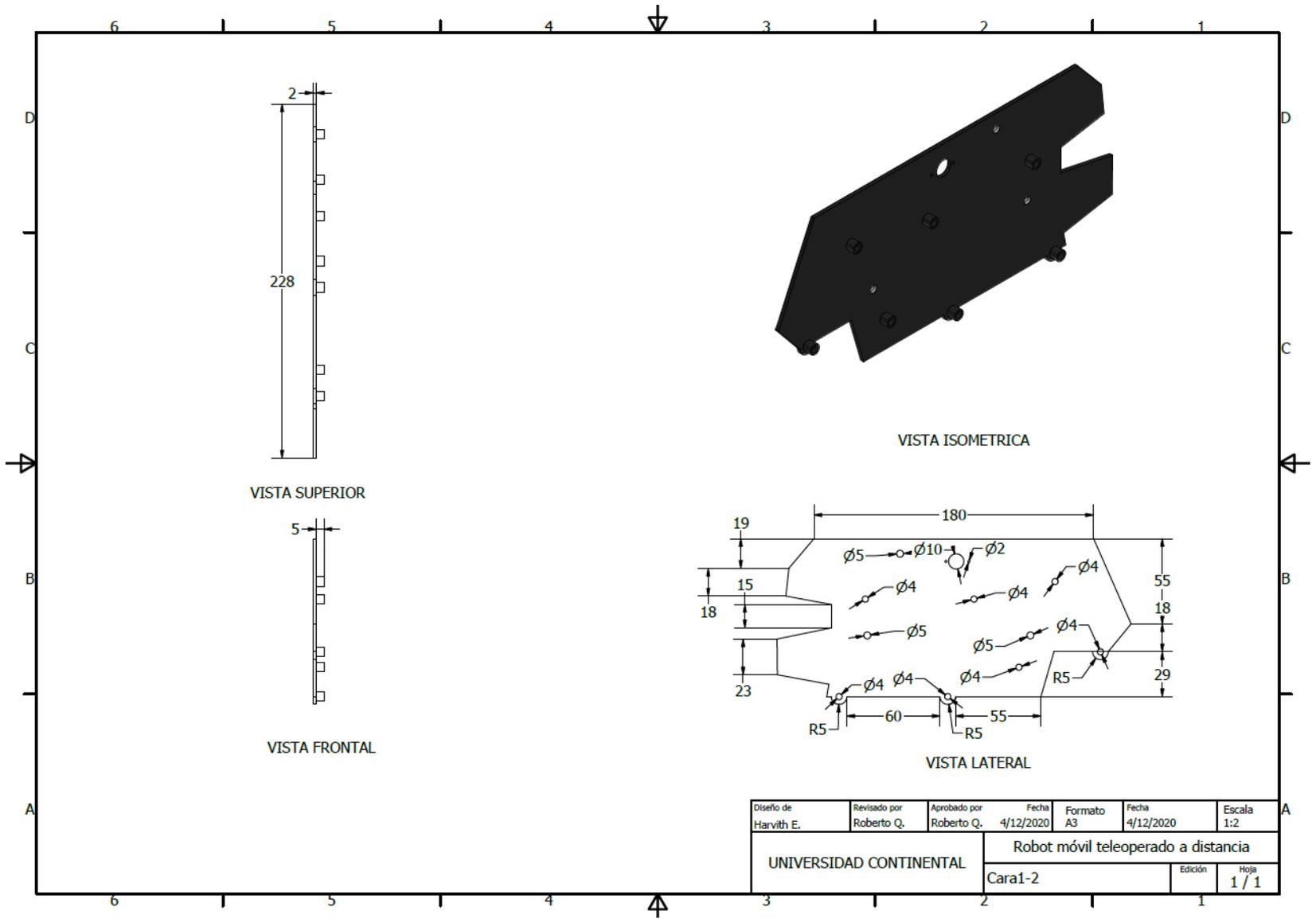
LISTA DE PIEZAS			
ELEMENTO	CTDAD	Nº DE PIEZA	DESCRIPCIÓN
1	1	Cara1	Aluminio
2	2	Union-A	ABS
3	5	Piñon lado A	Acero
4	1	Union-B	ABS
5	1	Motor DC-A	
6	16	ISO 7045 - M4 x 12 - 4.8 - Z	Tornillo de cabeza cilíndrica redondeada
7	54	ISO 7089 - 4	Arandelas planas - Serie normal
8	14	ISO 4035 - M4	Tuercas hexagonales estrechas (achafianadas)
9	10	ISO 7045 - M4 x 16 - 4.8 - Z	Tornillo de cabeza cilíndrica redondeada
10	4	KS B 1021 - M 1.7 x 4	Tornillos de cabeza ranurada
11	1	Pasador	ABS
12	1	Cara2	Aluminio
13	2	Union2-A	ABS
14	5	Piñon lado B	Acero
15	1	Union2-B	ABS
16	1	Motor DC-B	
17	1	Pasador	
18	1	Cubierta	Aluminio
19	6	ISO 7045 - M4 x 10 - 4.8 - Z	Tornillo de cabeza cilíndrica redondeada
20	8	ISO 4032 - M4	Tuercas hexagonales, tipo 1
21	1	Cara inferior	Aluminio
22	1	RouterGL300	
23	1	Servo Motor SF90	
24	6	ISO 7089 - 3	Arandelas planas - Serie normal
25	4	ISO 7045 - M3 x 8 - 4.8 - Z	Tornillo de cabeza cilíndrica redondeada
26	4	ISO 4032 - M3	Tuercas hexagonales, tipo 1
27	1	Ese soporte camara	Aluminio
28	1	Base eje camara	Aluminio
29	1	Soporte camara	Aluminio
30	1	GoPro Hero 3 HD Camera	
31	1	HC-SR04	
32	1	Antena	STEP AP214
33	1	ublox NEO-6M	STEP AP214
34	1	soporteultrasonico	
35	2	Correa sincrona	

Diseño de Harvith F.	Revisado por Roberto Q.	Aprobado por Roberto Q.	Fecha 4/12/2020	Formato A2	Fecha 4/12/2020	Escala 0,4 : 1
UNIVERSIDAD CONTINENTAL			Robot móvil teleoperado a distancia			
Plano de ensamble			Edición		Hoja 1 / 1	

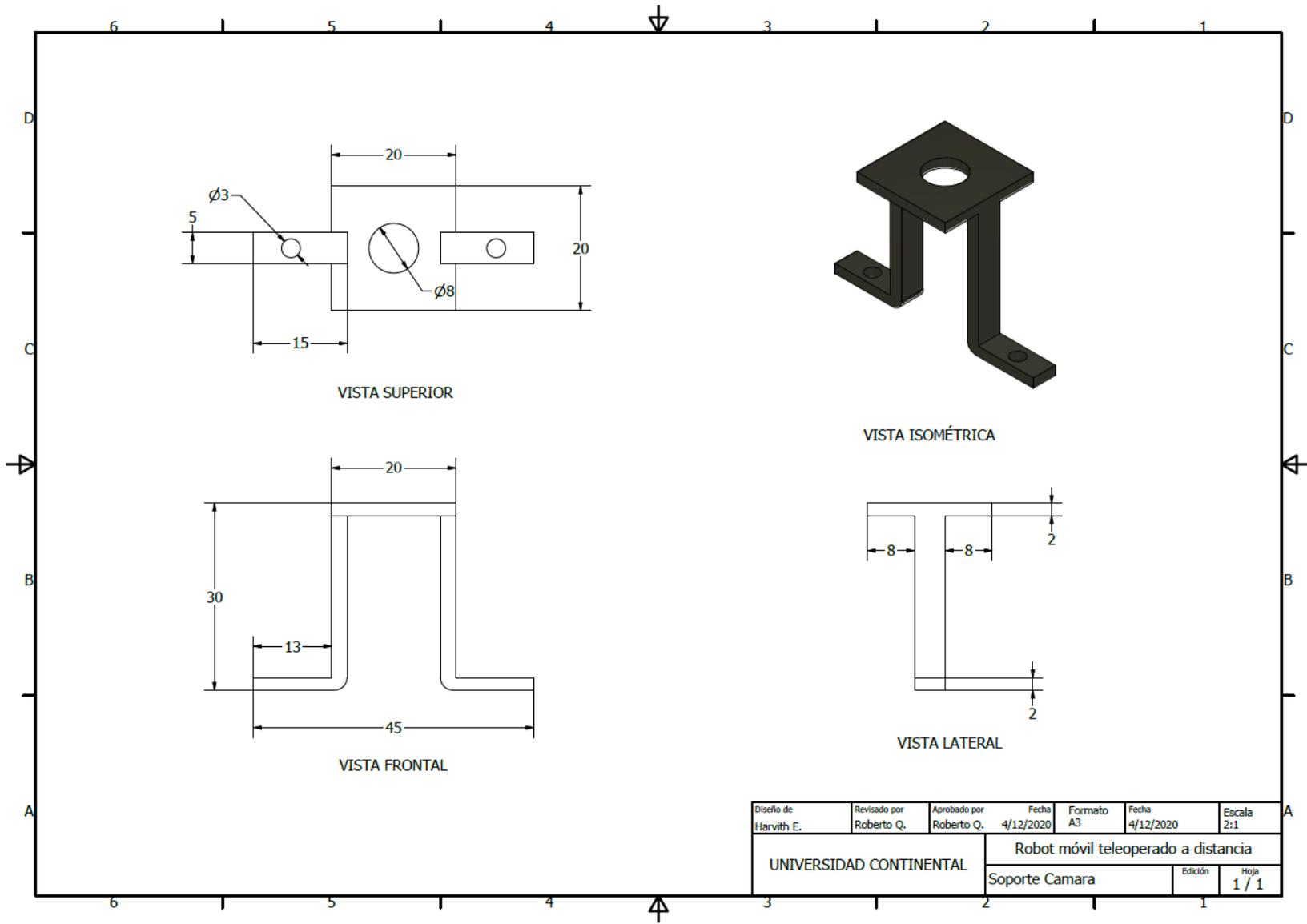


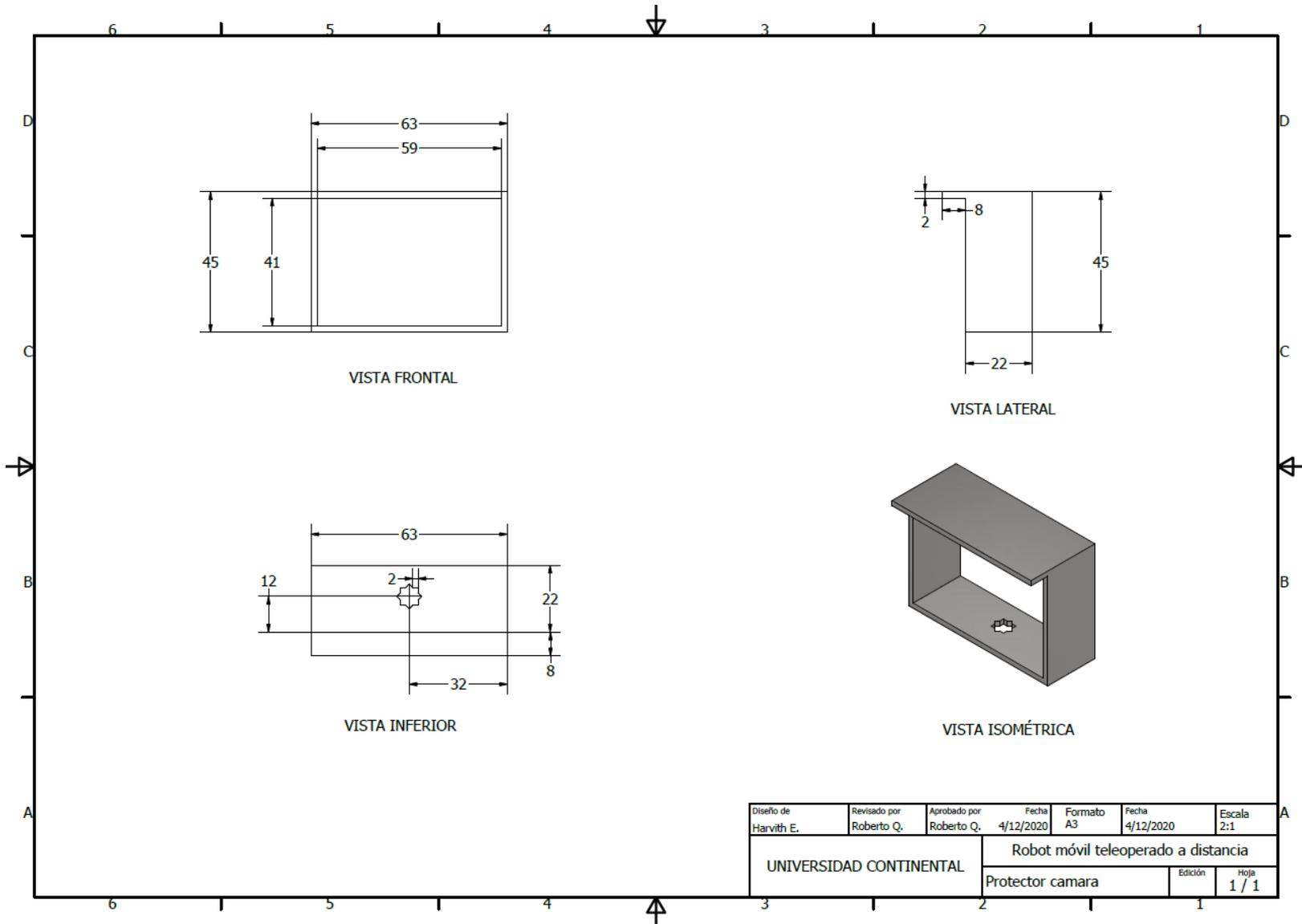
Diseno de Harvith E.	Revisado por Roberto Q.	Aprobado por Roberto Q.	Fecha 4/12/2020	Formato A3	Fecha 4/12/2020	Escala 1:2.5
UNIVERSIDAD CONTINENTAL			Robot móvil teleoperado a distancia			
			Cubierta		Edición	Hoja 1 / 1

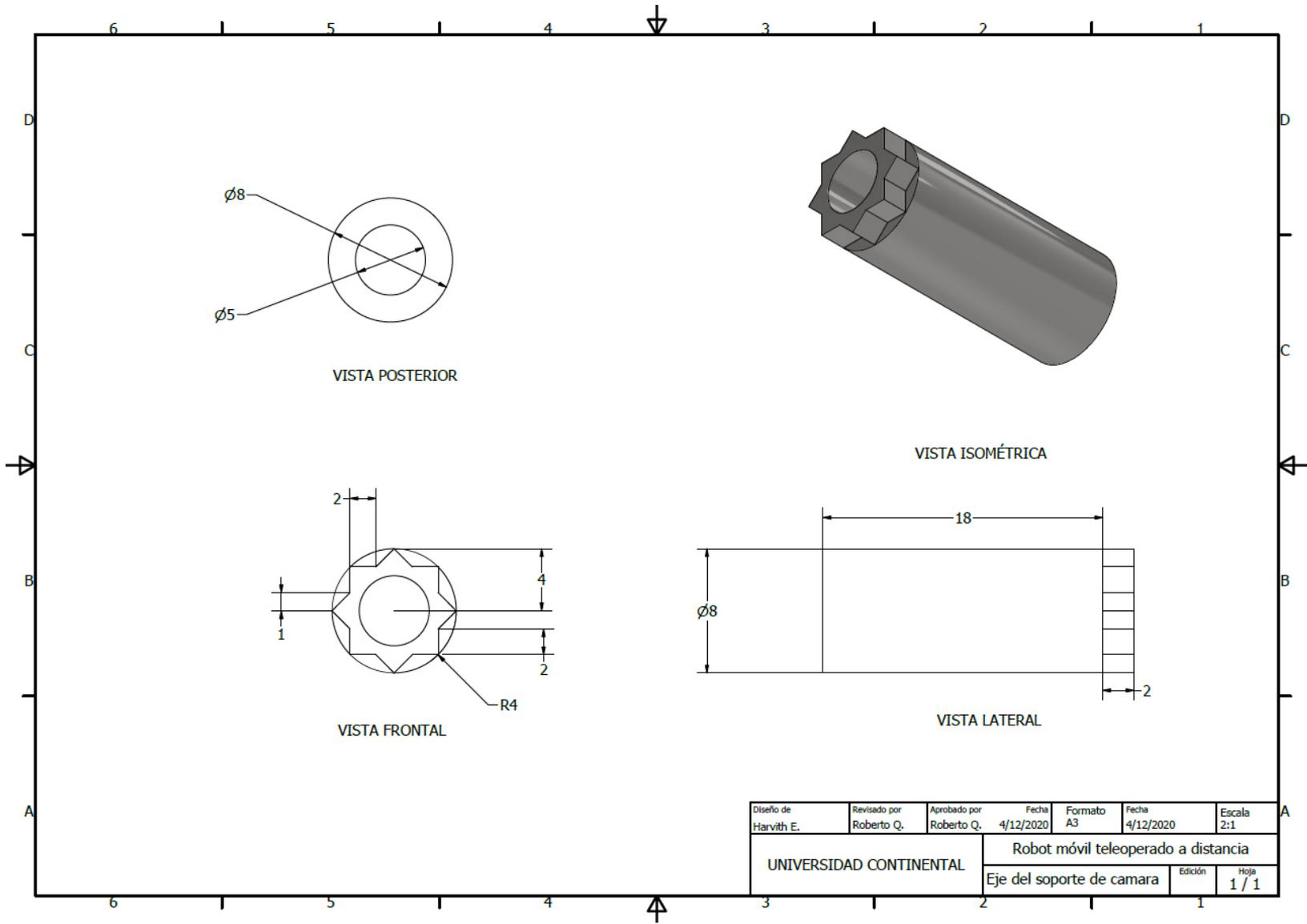




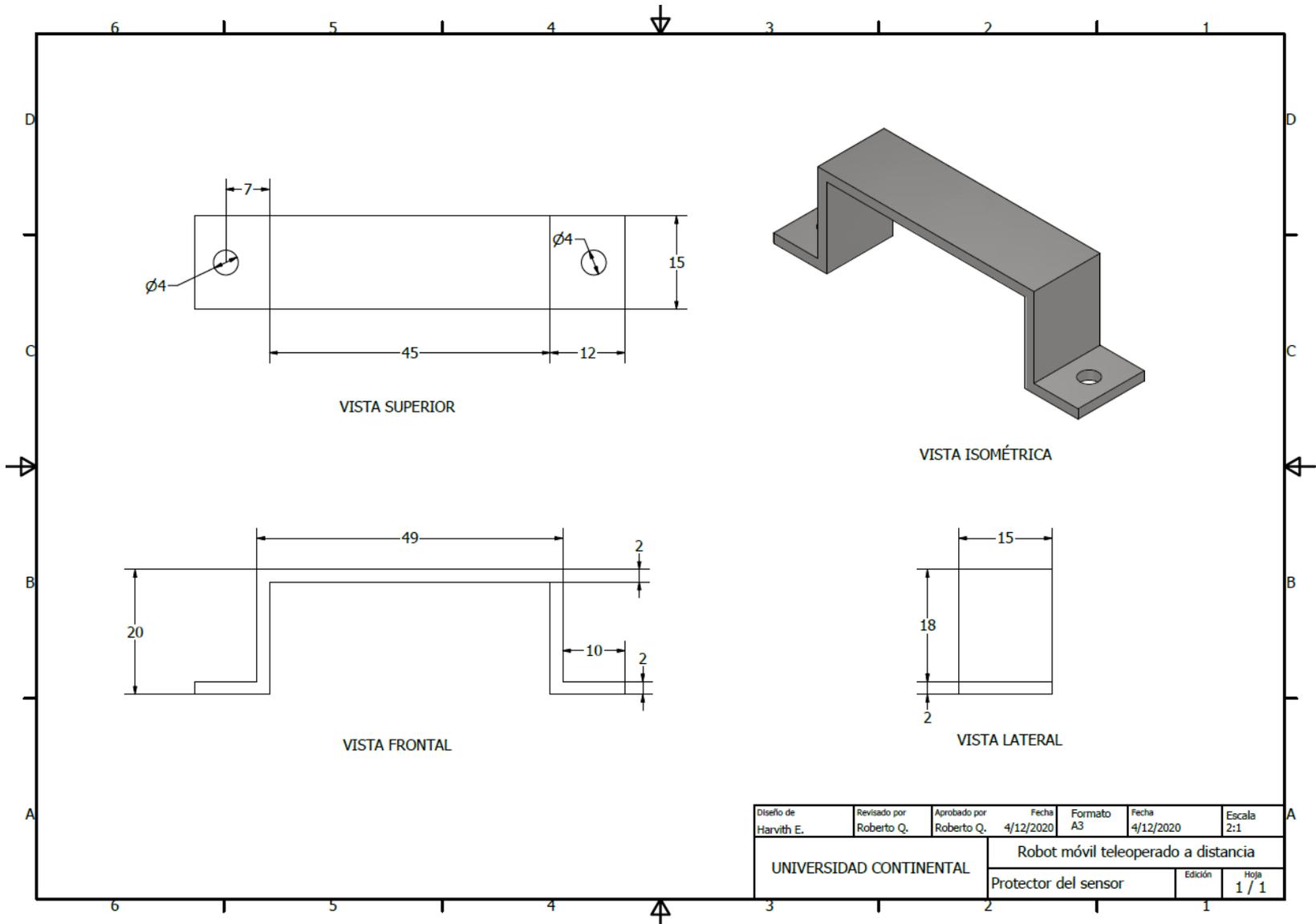
Diseño de Harvith E.	Revisado por Roberto Q.	Aprobado por Roberto Q.	Fecha 4/12/2020	Formato A3	Fecha 4/12/2020	Escala 1:2
UNIVERSIDAD CONTINENTAL			Robot móvil teleoperado a distancia			
Cara1-2					Edición	Hoja 1 / 1







Diseño de Harvith E.	Revisado por Roberto Q.	Aprobado por Roberto Q.	Fecha 4/12/2020	Formato A3	Fecha 4/12/2020	Escala 2:1
UNIVERSIDAD CONTINENTAL			Robot móvil teleoperado a distancia			
Eje del soporte de camara			Edición	Hoja 1 / 1		



## Código de programación

```
robot_movil_teleoperado_a_distancia$
//Robot movil teleoperado a distancia para rastreo de personas en un derrumbe de una edificacion
#include <LiquidCrystal.h>
#include <SoftwareSerial.h>
#include <TinyGPS.h>
SoftwareSerial gps(4,3); char dato= ' ';
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
int M1A=10;
int M1B=11;
int M2A=8;
int M2B=9;
int M3A=12;
int M3B=13;
int M4=5;
int M5=3;
int M4ON=6;
int M4OFF=7;
int M5ON=2;
int M5OFF=4;
#define echoPin 7 // Echo Pin
#define trigPin 8 // Trigger Pin
#define LEDPin 13
int maximumRange = 200; // Maximo rango a detectar en cm
int minimumRange = 0; // Minimo rango a detectar en cm long duration,
distance; // Duración usada para calcular la distancia
SoftwareSerial ss(10, 9);
static void smartdelay(unsigned long ms);
static void print_float(float val, float invalid, int len, int prec);
static void print_int(unsigned long val, unsigned long invalid, int len);
static void print_date(TinyGPS *gps);
static void print_str(const char *str, int len);
#include <Wire.h>
#include <Adafruit_AMG88xx.h>
Adafruit_AMG88xx amg;
#include <Adafruit_GFX.h>
#include <Adafruit_GFX.h>
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_AMG88xx.h>
#define TFT_CS 10
#define TFT_RST 9
#define TFT_DC 8
const uint16_t camColors[] = {0x480F,0x400F,0x400F,0x400F,0x4010,0x3810,0x3810,0x3810,0x3010,0x3010, 0x3010,0x2810,0x2810,0x2810,0x2810,0x2010,0x2010,0x2010,0x1810,0x1810, 0x1811,0x1811,0x1011,0x1011,0x1011,0x0811,0x0811,0x0811,0x0011,0x0011, 0x0011,0x0011,0x0011,
0xE5A0,0xE580,0xE560,0xE540,0xE520,0xE500,0xE4E0,0xE4C0,0xE4A0,0xE480, 0xE460,0xE440,0xE420,0xE400,0xE3E0,0xE3C0,0xE3A0,0xE380,0xE360,0xE340, 0xE320,0xE300,0xE2E0,0xE2C0,0xE2A0,0xE280,0xE260,0xE240,0xE220,0xE200, 0xF1E0,0xF1C0,0xF1A0,0xF180,0xF160,0xF140,0xF100,0xFOE0,0x
Adafruit_SI7735 tft = Adafruit_SI7735(TFT_CS, TFT_DC, TFT_RST);
Adafruit_AMG88xx amg; unsigned long delayTime;
float pixels[AMG88xx_PIXEL_ARRAY_SIZE];
```

```
robot_movil_teleoperado_a_distancia $
float pixels[RMG88xx_PIXEL_ARRAY_SIZE];
uint16_t displayPixelWidth, displayPixelHeight;
void setup() {
  lcd.begin(16, 2);
  lcd.print(" Peligro de ");
  gps.begin(9600);
  Serial.print("Testing TinyGPS library v. ");
  Serial.println(TinyGPS::library_version());
  Serial.println("by Mikal Hart"); Serial.println();
  Serial.println("Sats HDOP Latitude Longitude Fix Date Time Date Alt Course Speed Card Distance Course Card Chars Sentences Checksum");
  Serial.println(" (deg) (deg) Age Age (m) --- from GPS --- --- to London --- RX RX Fail"); Serial.println("-----");
  pinMode(M1A, OUTPUT);
  pinMode(M1B, OUTPUT);
  pinMode(M2A, OUTPUT);
  pinMode(M2B, OUTPUT);
  pinMode(M3A, OUTPUT);
  pinMode(M3B, OUTPUT);
  pinMode(M4, OUTPUT);
  pinMode(M5, OUTPUT);
  pinMode(M4GN, OUTPUT);
  pinMode(M4OFF, OUTPUT);
  pinMode(M5ON, OUTPUT);
  pinMode(M5OFF, OUTPUT);
  Serial.begin(9600);
  analogWrite(M4, 0);
  analogWrite(M5, 0);
  digitalWrite(M4ON, HIGH);
  digitalWrite(M4OFF, LOW);
  digitalWrite(M5ON, HIGH);
  digitalWrite(M5OFF, LOW);
  pinMode(trigPin, OUTPUT); //Trig es la salida del ultrasonido
  pinMode(echoPin, INPUT); //Echo es la entrada de el sonido regresando
  pinMode(LEDpin, OUTPUT);
}

void loop() {

  lcd.setCursor(0,1);
  lcd.print(" Collision");
  if(gps.available())
  {
    dato=gps.read();
    Serial.print(dato); }
  float lat, lon;
  unsigned long age, date, time, chars = 0;
```

```

robot_movil_teleoperado_a_distancia$
float flat, flon;
unsigned long age, date, time, chars = 0;
unsigned short sentences = 0, failed = 0; static const double LONDON_LAT = 51.508131, LONDON_LON = -0.128002;
print_int(gps.satellites(), TinyGPS::GPS_INVALID_SATELLITES, 5);
print_int(gps.hdop(), TinyGPS::GPS_INVALID_HDOOP, 5);
gps.f_get_position(&flat, &flon, &age);
print_float(flat, TinyGPS::GPS_INVALID_F_ANGLE, 10, 6);
print_float(flon, TinyGPS::GPS_INVALID_F_ANGLE, 11, 6);
print_int(age, TinyGPS::GPS_INVALID_AGE, 5);
print_date(gps); print_float(gps.f_altitude(),
TinyGPS::GPS_INVALID_F_ALTITUDE, 7, 2);
print_float(gps.f_course(), TinyGPS::GPS_INVALID_F_ANGLE, 7, 2);
print_float(gps.f_speed_kmph(), TinyGPS::GPS_INVALID_F_SPEED, 6, 2);
print_str(gps.f_course() == TinyGPS::GPS_INVALID_F_ANGLE ? "****" : TinyGPS::cardinal(gps.f_course()), 6);
print_int(flat == TinyGPS::GPS_INVALID_F_ANGLE ? 0xFFFFFFFF : (unsigned long)TinyGPS::distance_between(flat, flon, LONDON_LAT, LONDON_LON) / 1000, 0xFFFFFFFF, 9);
print_float(flat == TinyGPS::GPS_INVALID_F_ANGLE ? TinyGPS::GPS_INVALID_F_ANGLE : TinyGPS::course_to(flat, flon, LONDON_LAT, LONDON_LON), TinyGPS::GPS_INVALID_F_ANGLE, 7, 2);
print_str(flat == TinyGPS::GPS_INVALID_F_ANGLE ? "****" : TinyGPS::cardinal(TinyGPS::course_to(flat, flon, LONDON_LAT, LONDON_LON)), 6);
gps.stats(&chars, &sentences, &failed);
print_int(chars, 0xFFFFFFFF, 6);
print_int(sentences, 0xFFFFFFFF, 10);
print_int(failed, 0xFFFFFFFF, 9);
Serial.println();
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = duration/58.2;
if (distance >= maximumRange || distance <= minimumRange){
Serial.println("Fuera de Rango");
digitalWrite(LEDpin, HIGH);
} else {
Serial.print(distance);
Serial.println(" CM");
digitalWrite(LEDpin, LOW);
}
delay(50);

Serial.print("Thermistor Temperature = ");
Serial.print(amtg.readThermistor());
Serial.println(" °C");
Serial.println;
delay(1000);
amtg.readPixels(pixels);
for(int i=0; i<AMG88xx_PIXEL_ARRAY_SIZE; i++){

```

```

robot_movil_teleoperado_a_distancia$
for(int i=0; i<AMG883xx_PIXEL_ARRAY_SIZE; i++){
uint8_t colorIndex = map(pixels[i], MINTEMP, MAXTEMP, 0, 255);
colorIndex = constrain(colorIndex, 0, 255);
tft.fillRect(displayPixelHeight * floor(i / 8), displayPixelWidth * (i % 8), displayPixelHeight, displayPixelWidth, camColors[colorIndex]); }

if (Serial.available()) {
  char dato= Serial.read();
  Serial.print("Recivido:" );
  Serial.println(dato);

  if(dato=='a')
  {
    M12ON();

  }
  else if(dato=='b')
  {
    M12OFF();

  }
  else if(dato=='c')
  {
    M3ON();

  }
  else if(dato=='d')
  {
    M3OFF();

  }
  else if(dato=='x')
  {
    x();
  }
  else if(dato=='e')
  {
    Serial.println(dato);
    analogWrite (M4,25);
    analogWrite (M5,25);
  }
  else if(dato=='f')
  {
    Serial.println(dato);
    analogWrite (M4,50);
    analogWrite (M5,50);
  }
}

```

robot\_movil\_teleoperado\_a\_distancia \$

```
    analogWrite(M5,50);
}
else if(dato=='g')
{
    Serial.println(dato);
    analogWrite(M4,75);
    analogWrite(M5,75);
}
else if(dato=='h')
{
    Serial.println(dato);
    analogWrite(M4,100);
    analogWrite(M5,100);
}
else if(dato=='i')
{
    Serial.println(dato);
    analogWrite(M4,125);
    analogWrite(M5,125);
}
else if(dato=='j')
{
    Serial.println(dato);
    analogWrite(M4,150);
    analogWrite(M5,150);
}
else if(dato=='k')
{
    Serial.println(dato);
    analogWrite(M4,175);
    analogWrite(M5,175);
}
else if(dato=='l')
{
    Serial.println(dato);
    analogWrite(M4,200);
    analogWrite(M5,200);
}
else if(dato=='m')
{
    Serial.println(dato);
    analogWrite(M4,225);
    analogWrite(M5,225);
}
else if(dato=='n')
{
```

```
Serial.println(dato);
analogWrite(M4,250);
analogWrite(M5,250);
}
else if(dato=='o')
{
derecha();
}
else if(dato=='p')
{
izquierda();
}
else if(dato=='q')
{
parar();
}
else if(dato=='r')
{
iniciar();
}
else if(dato=='s')
{
M1ALON();
}
else if(dato=='t')
{
M1ALOFF();
}
else if(dato=='u')
{
M2ALON();
}
else if(dato=='v')
{
M2ALOFF();
}
}
}
```

void M1ALON()

robot\_movil\_teleoperado\_a\_distancia\$

```
void M12ON()
{
  digitalWrite (M1A, HIGH);
  digitalWrite (M1B, LOW);
  digitalWrite (M2A, HIGH);
  digitalWrite (M2B, LOW);
}
void M12OFF()
{
  digitalWrite (M1A, LOW);
  digitalWrite (M1B, HIGH);
  digitalWrite (M2A, LOW);
  digitalWrite (M2B, HIGH);
}
void M3ON()
{
  digitalWrite (M3A, HIGH);
  digitalWrite (M3B, LOW);
}
void M3OFF()
{
  digitalWrite (M3A, LOW);
  digitalWrite (M3B, HIGH);
}
void x()
{
  digitalWrite (M1A, LOW);
  digitalWrite (M1B, LOW);
  digitalWrite (M2A, LOW);
  digitalWrite (M2B, LOW);
  digitalWrite (M3A, LOW);
  digitalWrite (M3B, LOW);
}
void derecha()
{
  digitalWrite (M4ON, LOW);
  digitalWrite (M4OFF, LOW);
  digitalWrite (M5ON, LOW);
  digitalWrite (M5OFF, LOW);
  analogWrite (M4, 0);
  analogWrite (M5, 0);

  delay(1000);

  digitalWrite (M4ON, HIGH);
  digitalWrite (M4OFF, LOW);
```

robot\_movil\_teleoperado\_a\_distancia \$

```
digitalWrite (M5ON, HIGH);
digitalWrite (M5OFF, LOW);
analogWrite (M4, 150);
analogWrite (M5, 150);
}
void izquierda()
{
  digitalWrite (M4ON, LOW);
  digitalWrite (M4OFF, LOW);
  digitalWrite (M5ON, LOW);
  digitalWrite (M5OFF, LOW);
  analogWrite (M4, 0);
  analogWrite (M5, 0);

  delay(1000);

  digitalWrite (M4ON, LOW);
  digitalWrite (M4OFF, HIGH);
  digitalWrite (M5ON, LOW);
  digitalWrite (M5OFF, HIGH);
  analogWrite (M4, 150);
  analogWrite (M5, 150);

}
void parar ()
{
  digitalWrite (M4ON, LOW);
  digitalWrite (M4OFF, LOW);
  digitalWrite (M5ON, LOW);
  digitalWrite (M5OFF, LOW);
  analogWrite (M4, 0);
  analogWrite (M5, 0);
}
void iniciar() {
  digitalWrite (M4ON, HIGH);
  digitalWrite (M4OFF, LOW);
  digitalWrite (M5ON, HIGH);
  digitalWrite (M5OFF, LOW);
  analogWrite (M4, 150);
  analogWrite (M5, 150);
}
void M1ALON() {
  digitalWrite (M1A, HIGH);
  digitalWrite (M1B, LOW);
}
```

robot\_movil\_teleoperado\_a\_distancia \$

```
}  
void parar ()  
{  
  digitalWrite (M4ON, LOW);  
  digitalWrite (M4OFF, LOW);  
  digitalWrite (M5ON, LOW);  
  digitalWrite (M5OFF, LOW);  
  analogWrite (M4, 0);  
  analogWrite (M5, 0);  
}  
void iniciar() {  
  digitalWrite (M4ON, HIGH);  
  digitalWrite (M4OFF, LOW);  
  digitalWrite (M5ON, HIGH);  
  digitalWrite (M5OFF, LOW);  
  analogWrite (M4, 150);  
  analogWrite (M5, 150);  
}  
void M1ALON() {  
  digitalWrite (M1A, HIGH);  
  digitalWrite (M1B, LOW);  
}  
void M1ALOFF() {  
  digitalWrite (M1A, LOW);  
  digitalWrite (M1B, HIGH);  
}  
void M2ALON() {  
  digitalWrite (M2A, HIGH);  
  digitalWrite (M2B, LOW);  
}  
void M2ALOFF() {  
  digitalWrite (M2A, LOW);  
  digitalWrite (M2B, HIGH);  
}  
}  
<
```