

**UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA  
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES  
ESCUELA DE ELECTRÓNICA Y TELECOMUNICACIONES**



**“IMPLEMENTACIÓN DE UN PROTOTIPO DE ROBOT MÓVIL  
CON VISIÓN NOCTURNA PARA VIGILANCIA Y SEGURIDAD.”**

**TESIS DE GRADO**

**Previa a la obtención del título de  
INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES**

**AUTORA:**

**KATTY ZENAIDA NARANJO SUÁREZ**

**TUTOR:**

**Ing. Sendey Vera González.**

**LA LIBERTAD – ECUADOR**

**2013**

**UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA  
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES  
ESCUELA DE ELECTRÓNICA Y TELECOMUNICACIONES**

**“IMPLEMENTACIÓN DE UN PROTOTIPO DE ROBOT MÓVIL  
CON VISIÓN NOCTURNA PARA VIGILANCIA Y SEGURIDAD.”**

**TESIS DE GRADO**

**Previa a la obtención del título de  
INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES**

**AUTORA:**

**KATTY ZENAIDA NARANJO SUÁREZ**

**TUTOR:**

**Ing. Sendey Vera González.**

**LA LIBERTAD – ECUADOR**

**2013**

**La Libertad, 10 de Noviembre del 2013**

### **APROBACIÓN DEL TUTOR**

En mi calidad de Tutor del trabajo de investigación, “**IMPLEMENTACIÓN DE UN PROTOTIPO DE ROBOT MÓVIL CON VISIÓN NOCTURNA PARA VIGILANCIA Y SEGURIDAD**” elaborado por la Srta. Katty Naranjo Suárez, egresado de la Carrera de Electrónica y Telecomunicaciones, Escuela de Electrónica y Telecomunicaciones, Facultad de Sistemas y Telecomunicaciones de la Universidad Estatal Península de Santa Elena, previo a la obtención del Título de Ingeniero en Electrónica y Telecomunicaciones, me permito declarar que luego de haber orientado, estudiado y revisado, la apruebo en todas sus partes.

**Atentamente**

---

**Ing. Sendey Vera González.**

**TUTOR**

## **DEDICATORIA**

Dedico éste trabajo a la memoria de mi hermana Anabelle; su espíritu, su corazón y su alegría siempre inspirarán mi vida.

*Katty*

## **AGRADECIMIENTO**

A Dios y a la Virgen María, por ser el aliento y el apoyo incondicional en mi existencia, por darme determinación para seguir adelante en esta etapa de mi vida.

A mi madre Lorenza, por su sacrificio, amor y comprensión, por ser la persona que me ha acompañado durante toda mi vida. A mi padre Francisco, por su cariño sin limitaciones y por apoyarme. A mis hermanas, por compartir conmigo alegrías y tristezas, travesuras y fracasos; Anabelle, Raquel y Patricia. A todos mis sobrinos, que son una parte fundamental; Alfredo, Julissa, Belén, Nayely, Nicole, Raquelita y Celso. A mi fiel compañero, uno de los seres importantes en mi vida, siempre dispuesto a dar y recibir cariño, esperando atento por jugar y compartir; Chopi.

A mis compañeros, juntos hemos pasado momentos inolvidables. A mis profesores, por la sabiduría que me transmitieron en el desarrollo de mi formación profesional.

Al Ing. Sendey Vera, tutor de tesis, por su tiempo, por su valiosa guía y asesoramiento a la realización de este proyecto.

Por todo su apoyo, que sin ustedes no estaría en el lugar en el que me encuentro ahora.

Gracias.

## TRIBUNAL DE GRADO

---

Ing. Freddy Villao Santos, MsC.  
Decano de la Facultad de  
Sistemas y Telecomunicaciones

---

Ing. Washington Torres Guin, Msc  
Director de la Escuela de  
Electrónica y Telecomunicaciones

---

Ing. Sendey Vera González  
Profesor Tutor

---

Ing. Samuel Bustos Gaibor  
Profesor Área

---

Ab. Milton Zambrano Coronado, MsC.  
Secretario General - Procurador

**UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA  
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES  
ESCUELA DE ELECTRÓNICA Y TELECOMUNICACIONES**

**“IMPLEMENTACIÓN DE UN PROTOTIPO DE ROBOT MÓVIL CON  
VISIÓN NOCTURNA PARA VIGILANCIA Y SEGURIDAD.”**

Autora: Katty Naranjo.

Tutor: Ing. Sendey Vera.

**RESUMEN**

El desarrollo del prototipo de robot móvil de seguridad con visión nocturna permitirá ayudar al cuerpo de guardia a mejorar el control de seguridad y monitoreo. Este prototipo de robot móvil tiene como objetivo monitorear el área del laboratorio de electrónica de la Escuela de Electrónica y Telecomunicaciones de la Universidad Estatal Península de Santa Elena; ejecutando rutas programadas mediante sensores e incorpora una cámara inalámbrica para transmisión de video; en la parte de control emplearemos la placa Arduino. Tiene dos modos de operación, se puede seleccionar en qué modo desea que trabaje el prototipo sea este como Robot Autónomo, en esta forma de trabajo hace uso del sensor de ultrasonido en la detección de obstáculo y sensores PIR para detección de personas en movimiento; o como Robot Tele-operado en esta forma de trabajo el robot obedece a órdenes de navegación enviadas desde la PC mediante un enlace de radiofrecuencia. Programamos con el software Arduino, simulamos con Proteus ISIS, la parte de control y monitorización con Visual Basic Express 2010; y la base de datos del informe en SQL Server Express.

## Índice General

Portada	I
Aprobación del Tutor	III
Dedicatoria	IV
Agradecimiento	V
Tribunal de Grado	VI
Resumen	VII
Índice General	VIII
Índice de Tablas	XI
Índice de Figuras	XII
Índice de Anexos	XIV
<b>INTRODUCCIÓN</b>	<b>1</b>
<b>CAPÍTULO I</b>	
<b>1. MARCO REFERENCIAL</b>	<b>3</b>
1.1. Identificación del problema	3
1.2. Situación actual del problema	4
1.3. Justificación del tema	5
1.4. Objetivos	6
1.4.1. Objetivo general	6
1.4.2. Objetivos específicos	6
1.5. Hipótesis	6
1.6. Resultados esperados	7
<b>CAPÍTULO II</b>	
<b>2. MARCO TEÓRICO</b>	<b>8</b>
2.1. Antecedentes	8
2.1.1. Históricos	9
2.2. Bases teóricas	10
2.2.1. Robótica	11



2.2.2. Sistema de locomoción	12
2.2.3. Sensores de proximidad	16
2.2.4. Tecnología Zigbee	19
2.2.5. Arduino	22
2.2.6. Herramientas para el desarrollo del prototipo	23
2.3. Variables	25
2.4. Términos básicos	26
<b>CAPÍTULO III</b>	
<b>3. ANÁLISIS</b>	27
3.1. Diagrama de procesos	27
3.1.1. Descripción funcional de los procesos	28
3.2. Identificación de requerimientos	30
3.3. Análisis del sistema	31
3.3.1. Análisis Técnico	32
3.3.2. Análisis Económico	34
3.3.3. Análisis Operativo	36
3.4. Métodos e Instrumentos de investigación	37
3.4.1. Metodología de la investigación	37
3.4.2. Instrumentos de recolección de información	39
3.5. Análisis e interpretación de los resultados de encuesta	40
<b>CAPÍTULO IV</b>	
<b>4. DISEÑO</b>	49
4.1. Arquitectura de la solución	50
4.1.1. Diagrama de actividades del prototipo	58
4.2. Diseño de la interfaz	61
<b>CAPÍTULO V</b>	
<b>5. IMPLEMENTACIÓN</b>	64
5.1. Construcción	65

5.1.1. Sistema de control	67
5.1.2. Sistema sensorial	68
5.1.3. Sistema de comunicación	69
5.1.4. Sistema actuador	70
5.1.5. Sistema de potencia y alimentación	71
5.2. Pruebas	75
5.2.1. Pruebas y resultados del sensor ultrasónico	75
5.2.2. Pruebas y resultados de los sensores PIR	77
5.2.3. Pruebas y resultados del recorrido autónomo libre de Obstáculo	78
5.2.4. Pruebas y resultados del recorrido autónomo con un intruso	81
5.3. Documentación	82
5.3.1. Comprobación de hipótesis	82
5.3.2. Manual de Usuario	83
CONCLUSIONES	84
RECOMENDACIONES	85
BIBLIOGRAFÍA	86
ANEXOS	87
GLOSARIO	

## Índice de Tablas

Tabla 2.1. Variable independiente	25
Tabla 2.2. Variable dependiente	25
Tabla 3.1. Requerimientos de suministros para el prototipo	34
Tabla 3.2. Requerimientos de software para el prototipo	32
Tabla 3.3. Requerimientos de hardware para programación	32
Tabla 3.4. Requerimientos de hardware para el prototipo	33
Tabla 3.5. Costo requerimientos de suministros para el prototipo	36
Tabla 3.6. Costo requerimientos de software para el prototipo	34
Tabla 3.7. Costo requerimientos de hardware para programación	34
Tabla 3.8. Costo requerimientos de hardware para el prototipo	35
Tabla 3.9. Costo total para el desarrollo del prototipo	36
Tabla 4.1. Funcionamiento del sistema de locomoción	57
Tabla 4.2. Consumo de energía de los elementos en una ronda	58
Tabla 5.1. Pines de Arduino para el prototipo	68
Tabla 5.2. Resultado de pruebas del ultrasónico	77
Tabla 5.3. Resultado de pruebas de los sensores PIR	79
Tabla 5.4. Resultado de pruebas del recorrido autónomo libre de obstáculo	81
Tabla 5.5. Resultado de pruebas del recorrido autónomo con un intruso	82

## Índice de Figuras

Figura 2.1. Diagrama del robot móvil	13
Figura 2.2. Esquema básico de un puente H	15
Figura 2.3. PWM	16
Figura 2.4. Sensor Ultrasonido	17
Figura 2.5. Funcionamiento de los sensores de ultrasonido	18
Figura 2.6. Sensor PIR	18
Figura 2.7. Conexiones mínimas requeridas para el Xbee	20
Figura 2.8. Arduino Arduclema	22
Figura 3.1. Diagrama de procesos del prototipo de Robot Móvil	28
Figura 3.2. Gráfico N°1	42
Figura 3.3. Gráfico N°2	42
Figura 3.4. Gráfico N°3	43
Figura 3.5. Gráfico N°4	43
Figura 3.6. Gráfico N°5	44
Figura 3.7. Gráfico N°6	44
Figura 3.8. Gráfico N°7	45
Figura 3.9. Gráfico N°8	45
Figura 3.10. Gráfico N°9	46
Figura 3.11. Gráfico N°10	47
Figura 3.12. Gráfico N°11	47
Figura 3.13. Gráfico N°12	48
Figura 3.14. Gráfico N°13	48
Figura 4.1. Diagrama del prototipo	51
Figura 4.2. Diseño de la estructura	52
Figura 4.3. Diseño de ubicación de elementos del prototipo.	53
Figura 4.4. Diseño del sistema de locomoción	56
Figura 4.5. Diagrama de actividades Modo Tele-Operado	59
Figura 4.6. Diagrama de actividades Modo Automático	60
Figura 4.7. Pantalla Inicial del programa de monitorización de	62

vigilancia y seguridad	
Figura 4.8. Pantalla principal del programa de monitorización de vigilancia y seguridad, Modo Manual	63
Figura 4.9. Pantalla principal del programa de monitorización de vigilancia y seguridad, Modo Automático	64
Figura 5.1. Construcción prototipo base	66
Figura 5.2. Construcción prototipo motores	67
Figura 5.3. Construcción prototipo base-estructura	67
Figura 5.4. Construcción prototipo sensores	68
Figura 5.5. Conexión Arduino y sensor HC-SR04	69
Figura 5.6. Conexión Arduino y sensor DYP-ME003	70
Figura 5.7. Conexión Arduino y Xbee S1	71
Figura 5.8. Conexión puente H	71
Figura 5.9. Conexión Arduino y motores	72
Figura 5.10. Esquema de la tarjeta reguladora de voltaje	73
Figura 5.11. Esquema de la tarjeta control de estado de la batería	74
Figura 5.12. Conexión interna del prototipo	75
Figura 5.13. Prototipo terminado	75
Figura 5.14. Configuración del sensor de ultrasonido	76
Figura 5.15. Prueba sensor ultrasonido	77
Figura 5.16. Configuración de los sensores PIR	78
Figura 5.17. Prueba sensores PIR	79
Figura 5.18. Prueba recorrido autónomo	80
Figura 5.19. Prueba recorrido autónomo libre de obstáculo	81
Figura 5.20. Prueba recorrido autónomo con un obstáculo	83

## **Índice de Anexos**

Anexo 1. Encuesta

Anexo 2. Simulación en Matlab del movimiento del robot con tracción diferencial.

Anexo 3. Diagrama del circuito electrónico general.

Anexo 4. Código Fuente Algoritmo de programación en Arduino.

Anexo 5. Código Fuente de la interfaz en Visual Basic 2010.

Anexo 6. Diagrama de los circuitos impresos.

Anexo 7. Manual de Usuario.

## INTRODUCCIÓN

La Robótica es fundamental para el desarrollo científico y tecnológico a nivel mundial; sobre todo para mejorar la seguridad en los laboratorios de Electrónica es necesario fortalecer la vigilancia de la Escuela de Electrónica y Telecomunicaciones de la Universidad Estatal Península de Santa Elena, por ello es imprescindible incorporar nueva tecnología robótica de seguridad.

Gracias a toda la información y referencias de investigaciones sobre robótica, la hipótesis que rige esta tesis asevera que la implementación del robot móvil con visión nocturna para vigilancia y seguridad ayudará al cuerpo de guardia a mejorar el control de seguridad y monitoreo en el área del laboratorio de la Escuela de Electrónica y Telecomunicaciones de la Universidad Estatal Península de Santa Elena.

El proyecto incluye cinco capítulos: En el Capítulo 1 se detalla la identificación del problema, situación actual, la justificación, los objetivos generales, objetivos específicos, hipótesis y resultados esperados del proyecto.

En el Capítulo 2 se encuentra los antecedentes históricos de la UPSE, base teórica, términos básicos y operacionalización de variables.

En el Capítulo 3 se detalla los requerimientos del proyecto, el cual indica las especificaciones técnicas, presentando una lista de necesidades y diagramas donde se evidencia el análisis realizado. Para lograrlo se siguió una metodología de investigación científica; hipotético-deductivo, cuya colaboración nos otorga obtención rápida de datos, experimentación a través de ensayo y error, y los resultados obtenidos se basan en las variables encontradas en el campo de pruebas.

En el Capítulo 4 se muestra el modelo que se construirá para la solución que más se adapte a los requerimientos y necesidades del proyecto.

En el Capítulo 5 se presenta las evidencias gráficas, registro fotográfico y explicación descriptiva de la realización del proyecto.

Se resolvieron problemas de alimentación, se depuró el software y se unificó todas las etapas para que integren un solo sistema, procesando y ejecutando órdenes que dirijan al móvil desde un punto inicial a uno final pre-definido en un entorno de prueba diseñado. El prototipo de robot móvil monitoreará una determinada área de interés por parte del usuario, es capaz de guiarse de forma autónoma a través de un entorno establecido con anterioridad dirigiéndose a un destino definido.

Finalmente se exponen las conclusiones y recomendaciones para obtener un óptimo funcionamiento del sistema.



# **CAPÍTULO I**

## **MARCO REFERENCIAL**

### **1. Marco Referencial**

En este capítulo se detalla la identificación del problema, situación actual, la justificación, los objetivos generales, objetivos específicos, hipótesis y resultados esperados del proyecto.

#### **1.1. Identificación del problema**

Se ha notado que en la guardia nocturna que es realizada por vigilantes, estas personas están expuestas a posibles peligros al darse algún evento de robo, esto puede inclusive desencadenar un enfrentamiento con el intruso.

Adicionalmente el tiempo de las rondas de los guardias se ve limitado a la cantidad de personal existente en la institución, dejando espacios de tiempo sin resguardar.

Específicamente en el laboratorio de la Escuela de Electrónica y Telecomunicaciones de la Universidad Estatal Península de Santa Elena se realiza las rondas típicas, las cuales conllevan a la situación antes mencionada. Además la falta de tecnología robótica utilizada para mejorar el control de vigilancia en estas rondas nocturnas es inexistente en la institución.

## **1.2. Situación actual del problema**

La seguridad y vigilancia es uno de los campos en el cual ha evolucionado la robótica, es por ello que se han creado diferentes tecnologías que ayudan a cumplir con las tareas para el cual fue diseñado un determinado prototipo. En estos robots se ha instalado sistemas de vigilancia que permiten observar y controlar un establecimiento de forma autónoma. Además tienen sistemas propios que le permiten reaccionar ante eventos externos, como pueden ser la presencia de un individuo u objeto.

Se debe señalar la elevada aplicación de robótica de seguridad en países como Estados Unidos o Japón, aunque su uso se ha extendido a nivel mundial proporcionando ventajas de un significativo aumento en la eficacia y la reducción de operativos humanos para las funciones habituales de seguridad.

En la actualidad en Ecuador debido a la inseguridad se ha desarrollado soluciones tecnologías como cámaras IP, control de acceso, cercas eléctricas y alarmas contra robo.

La problemática se detecta al conocer que en el control de la vigilancia en las rondas nocturnas del área del laboratorio de la Escuela de Electrónica y Telecomunicaciones de la Universidad Estatal Península de Santa Elena no se utiliza tecnología robótica, esto implica que el control sobre la vigilancia debe ser solo mediante personas, estas rondas típicas que se realizan en la institución, son las rondas planificadas para comprobar el perímetro de la propiedad; al utilizar un modelo de robot de vigilancia ayudará a disminuir esa dependencia de utilizar personas para determinada acción.

En la Universidad Estatal Península de Santa Elena no existen prototipos de robot móvil de seguridad por lo tanto optamos por implementar este proyecto que aportará una solución a una problemática social.

### **1.3. Justificación del tema**

Este proyecto plantea implementar un prototipo de robot móvil para vigilancia y seguridad, que permitirá monitorear el área del laboratorio de la Escuela de Electrónica y Telecomunicaciones de la Universidad Estatal Península de Santa Elena; así como responder ante posibles situaciones de riesgo; como preventiva de seguridad y de cierta forma conservar la integridad del vigilante.

La tecnología robótica ayudará al vigilante a realizar sus rondas, por lo que se tiene la ventaja de contar con una máquina inspeccionando el área durante las jornadas de trabajo alertando inclusive ante situaciones adversas.

## **1.4. Objetivos**

Los objetivos son esenciales para la culminación del proyecto porque establecen un curso, revelan prioridades y sientan las bases para planificar, organizar y controlar con eficiencia.

### **1.4.1. Objetivo General**

Diseñar y construir un prototipo de robot móvil con visión nocturna mediante el uso de tecnología robótica para el control de vigilancia y monitoreo en el área del laboratorio de la Escuela de Electrónica y Telecomunicaciones de la Universidad Estatal Península de Santa Elena.

### **1.4.2. Objetivo Específicos**

- Investigar y establecer la estructura del diseño físico y de control del robot para dar estabilidad al prototipo.
- Diseñar las tarjetas electrónicas para acoplar el sistema de potencia con el sistema de locomoción y control
- Diseñar el software para el control y monitoreo del robot móvil con visión nocturna.
- Implantar el prototipo de robot móvil con visión nocturna mediante el uso de sensores y actuadores para el control de vigilancia y monitoreo.

## **1.5. Hipótesis**

La implementación del prototipo de robot móvil con visión nocturna para vigilancia y seguridad ayudará al cuerpo de

guardia a mejorar el control de vigilancia y monitoreo en el área del laboratorio de la Escuela de Electrónica y Telecomunicaciones de la Universidad Estatal Península de Santa Elena.

#### **1.6. Resultados esperados**

Se espera que la estructura sea la adecuada para adaptar los elementos utilizados y proporcione soporte y estabilidad al robot.

Con la elaboración de las tarjetas electrónicas se podrá acoplar el sistema de potencia con el sistema de locomoción y control para ejecutar efectivamente la ruta en el área determinada.

Con este software se espera poder controlar eficientemente el robot móvil con visión nocturna y poder brindarle la visión del entorno al guardia.

Se pretende que con la implantación del prototipo se pueda mejorar el control de seguridad y monitoreo en el área del laboratorio de la Escuela de Electrónica y Telecomunicaciones.

# **CAPÍTULO II**

## **MARCO TEÓRICO**

### **2. Marco Teórico**

En este capítulo se da a conocer los antecedentes históricos y legales de la Universidad Estatal Península de Santa Elena. Además de la teoría relacionada a los componentes utilizados en este proyecto.

#### **2.1. Antecedentes**

Se presenta los antecedentes históricos de la Universidad Estatal Península de Santa Elena.

### **2.1.1. Históricos**

En diferentes épocas y a través de varias instituciones, ciudadanos peninsulares identificados con el quehacer socio-educativo de la comunidad, realizaron en su turno, una serie de acciones y actividades para conseguir el funcionamiento de un centro de educación superior en nuestra península.

En la década de los 80 se establecen varios colegios estatales y particulares en la zona peninsular y se siente con mayor fuerza la necesidad de contar con un establecimiento de educación superior en la Península de Santa Elena, que permita a los bachilleres, que por cientos, egresaban de las aulas de los colegios secundarios, continuar sus estudios superiores.

Desde 1984 a 1994 las Municipalidades de Salinas y Santa Elena y diversas instituciones cívicas realizan gestiones en procura de institucionalizar la Educación Superior en la Península de Santa Elena, consiguiéndose el funcionamiento de la Extensión Universitaria de la Universidad de Guayaquil en las áreas de Ingeniería Industrial con el Programa de Tecnología Industrial; Filosofía y Letras, Ingeniería Comercial e Idiomas.

A pesar de esto, el clamor de los peninsulares por contar con un centro de educación autónomo e independiente, ante la serie de problemas surgidos en las diversas extensiones, crece cada vez más y es por eso que en 1992 se constituye el Comité de gestión pro-universidad en la Península de Santa Elena con la participación del Abg. Xavier Tomalá Montenegro Director Ejecutivo, Carmen León de Lyle Presidenta, Abg. Pedro Reyes Laínez Vicepresidente, Ing. Juan Montenegro Muñoz, Ángel Rubio Ortega, Dr. Miguel Pazmiño, Carmen Lyle León Vocales, actúa

como Secretario de la Dirección Ejecutiva el profesor Milton González Santos.

El 30 de agosto de 1995, el Abg. Xavier Tomalá Montenegro, en su calidad de Director Ejecutivo del comité de gestión, presenta en el seno de dicho comité la exposición de motivos y el Proyecto de Ley para crear la Universidad a nivel estatal que se denomina Universidad del Pacífico en la Península de Santa Elena.

El referido proyecto fue aprobado por todos los miembros del comité de gestión y fue presentado en el Congreso Nacional en septiembre de 1995, el mismo que es acogido y auspiciado por el Diputado de ese entonces, profesor Juan José Castelló y aprobado por el Congreso Nacional el 9 de junio de 1996. La comunidad peninsular recibió la noticia con gran entusiasmo y por iniciativa de la Radio La Voz de la Península se convoca a diferentes autoridades de la península para ampliar el comité de gestión y establecer acciones y estrategias para conseguir la aprobación final por parte del Presidente de la República, Arq., Sixto Durán Ballén, quien pese a todo, vetó totalmente el Proyecto privando a nuestra región de un derecho inalienable como es el derecho a la Educación Superior.<sup>1</sup>

## **2.2. Bases Teóricas**

Es importante mencionar los conceptos principales sobre tecnología robótica, así como también los componentes del proyecto; esto implica el hardware, dentro del cual se hará énfasis en los módulos de locomoción, sensores, control, comunicación y la aplicación del software que se utiliza para controlar el prototipo.

---

<sup>1</sup>Página Oficial de la Universidad Estatal Península de Santa Elena, Históricas, (<http://www.upse.edu.ec/resena-historica.html#sthash.lkZUrbkw.dpuf>)



### 2.2.1. La Robótica

La robótica es la rama de la tecnología que se dedica al diseño, construcción, operación, disposición estructural, manufactura y aplicación de los robots; combina diversas disciplinas como son: la mecánica, la electrónica, la informática, la inteligencia artificial, la ingeniería de control y la física.

El término robot se popularizó con el éxito de la obra RUR (Robots Universales Rossum) escrita por Karel Capek en 1920. En la traducción al inglés de dicha obra, la palabra checa robota, que significa trabajos forzados, fue traducida como robot. El término robótica es acuñado por Isaac Asimov, definiendo a la ciencia que estudia a los robots.<sup>2</sup>

**Definición de Robot.-** Un robot, es un agente artificial mecánico o virtual. Es una máquina usada para realizar un trabajo automáticamente y que es controlada por una computadora.<sup>3</sup>

La Organización Internacional para la Estandarización provee su definición de robot en ISO 8373: “un manipulador automáticamente controlado, reprogramable, multiuso, programable en tres o más ejes, que pueden estar fijos en un lugar o movilizarse para ser usado en aplicaciones de automatización industrial”.

En general, un robot, para ser considerado como tal, debería presentar algunas de estas propiedades:

- No es natural, sino que ha sido creado artificialmente.
- Puede sentir su entorno.
- Puede manipular cosas de su entorno.

---

<sup>2</sup> Wikipedia, Robótica, [en línea], (<http://es.wikipedia.org/wiki/Robótica>)

<sup>3</sup> Alegs, Diccionario de Informática: Definición de robot, [en línea], Argentina, 2012, Disponible (<http://www.alegsa.com.ar/Dic/robot.php>)

- Tiene cierta inteligencia o habilidad para tomar decisiones basadas en el ambiente o en una secuencia pre-programada automática.
- Es programable.
- Puede moverse en uno o más ejes de rotación o traslación.
- Puede realizar movimientos coordinados.<sup>4</sup>

### **Clasificación de los robots.-** Según su arquitectura.

La arquitectura, es definida por el tipo de configuración general del robot. La subdivisión de los robots, con base a su arquitectura son: Androides, Móviles, Poliarticulados, Híbridos y Zoomórficos.

**Robots móviles.-** Son robots con grandes capacidades de desplazamiento, basadas en carros o plataformas y dotadas de un sistema locomotor de tipo rodante. Siguen su camino por telemando o guiándose por la información recibida de su entorno a través de sus sensores, pueden incluso a llegar a sortear obstáculos y están dotados de un nivel relativamente elevado de inteligencia. Existe una gran variedad de estructuras mecánicas para robots junto con los sistemas de locomoción más comunes para robots móviles.<sup>5</sup>

#### **2.2.2. Sistema de locomoción**

Los robots con ruedas son la solución más simple para conseguir la movilidad en terrenos firmes y libres de obstáculos. Son sencillos de controlar. Los robots móviles emplean diferentes tipos de locomoción mediante ruedas que les confieren características y propiedades diferentes respecto a la eficiencia energética, dimensiones, cargas útiles y maniobrabilidad.

---

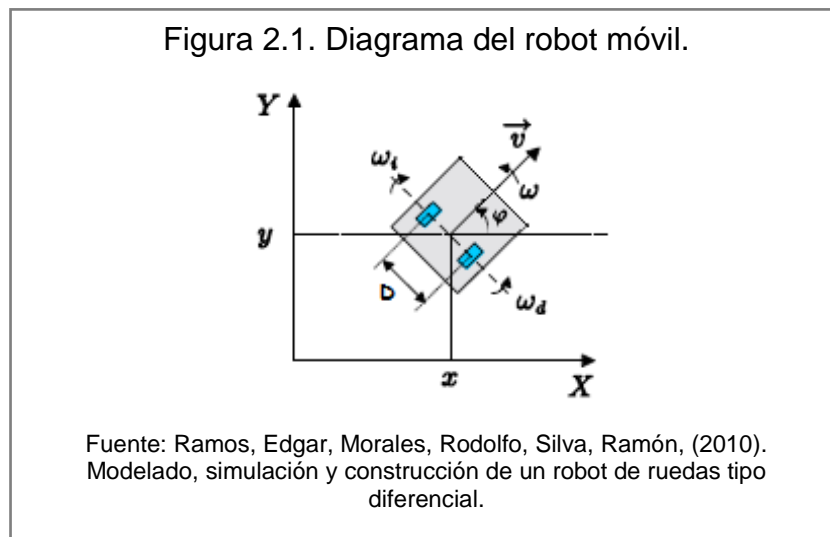
<sup>4</sup> Wikipedia, Tres leyes de la Robótica, ([http://es.wikipedia.org/wiki/Tres\\_leyes\\_de\\_la\\_Robótica](http://es.wikipedia.org/wiki/Tres_leyes_de_la_Robótica))

<sup>5</sup> Wikipedia, Robot, (<http://es.wikipedia.org/wiki/Robot>)

**Robot móvil con tracción diferencial.-** La configuración por tracción diferencial se caracteriza porque el movimiento se consigue con dos ruedas, acopladas cada una a su propio motor, teniendo normalmente uno o dos ruedas de apoyo. De esta forma se logra que el robot avance en línea recta fijando ambos motores a la misma velocidad, también se le puede hacer girar en una u otra dirección cuando se aplican velocidades diferentes.

Asimismo, para hacer más tratable el problema de modelado, en la obtención de los modelos cinemático de RMR, es común introducir suposiciones de diseño y operación como:

- a) El robot se mueve sobre una superficie plana.
- b) El deslizamiento en las ruedas izquierda y derecha es despreciable.
- c) El robot es rígido y no cuenta con partes flexibles.
- d) El robot tiene restricciones no holonómicas.



**Modelo cinemático del robot móvil.-** Para el modelado del robot móvil se considera que el movimiento del robot se realiza en el plano XY, con una configuración de tracción diferencial. En la

Figura 2.1 se muestra el diagrama del robot móvil para desarrollar el modelo cinemático de un robot diferencial. De acuerdo a la figura  $(x, y)$  denota la posición del punto medio del eje que une las dos llantas,  $\varphi$  describe el ángulo que forma el eje de simetría del móvil respecto al eje  $X$  positivo,  $\omega d$  y  $\omega i$  son las velocidades lineales de las llantas derecha e izquierda, respectivamente y  $D$  es la separación entre los ejes de las llantas.<sup>6</sup>

De las ecuaciones siguientes es claro que para  $D$  dado, la única forma de imponer que las variables del sistema  $(x, y, \theta)$ , generen una trayectoria deseada queda determinada por la elección apropiada de  $\omega d$  y  $\omega i$ , las cuales son conocidas como variables de control del móvil.<sup>7</sup>

$$x = \frac{\omega_i - \omega_d}{2} \cos(\theta) \quad (1.1)$$

$$y = \frac{\omega_i - \omega_d}{2} \sin(\theta) \quad (1.2)$$

$$\theta = \frac{\omega_d - \omega_i}{D} \quad (1.3)$$

**Motor De Corriente Continua.-** Es una máquina que transforma energía eléctrica en energía mecánica por medio de interacciones electromagnéticas. Un motor de corriente continua está compuesto de un estator y un rotor. En muchos motores C.C., generalmente los más pequeños, el estator está compuesto de imanes para crear un campo magnético. El funcionamiento se basa en la interacción entre el campo magnético del imán permanente y el generado por las bobinas, ya sea una atracción o una repulsión, hacen que el eje del motor comience su

---

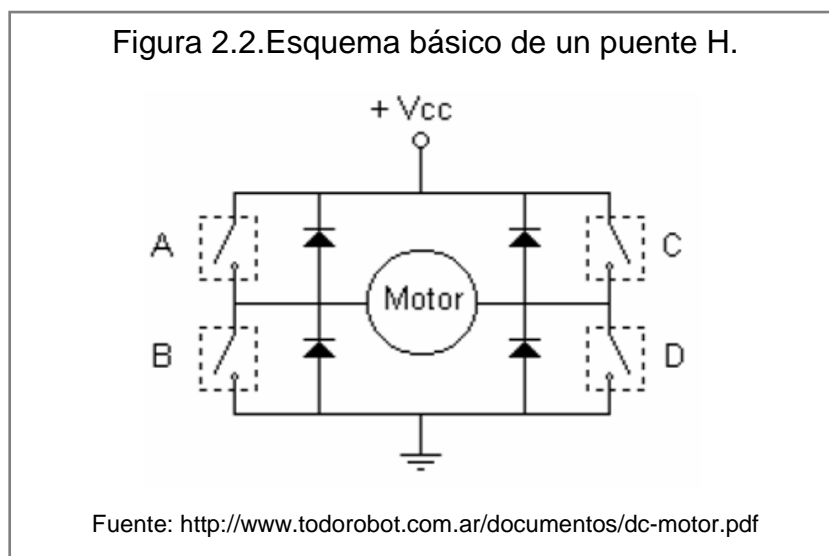
<sup>6</sup> Ramos, Edgar, Morales, Rodolfo, Silva, Ramón, (2010). Modelado, simulación y construcción de un robot de ruedas tipo diferencial, Mexico.

<sup>7</sup> Negrete, Marco, Simulación del modelo cinemático de un robot de par diferencial.

movimiento. El fácil control de posición y velocidad de este tipo de motor, lo han convertido en una de las mejores opciones en aplicaciones de control y automatización de procesos.<sup>8</sup>

**Driver de potencia para control de motores.-** La etapa de potencia es el elemento que hace de intermediario entre las partes mecánicas y electrónicas. La necesidad de usar esta etapa intermedia radica en la escasa capacidad para suministrar corriente, ya que por sí sola no sería capaz de mover los motores. El método más sencillo para lograr la intensidad requerida es mediante el uso de transistores funcionando como interruptor; en este proyecto se utilizará circuitos integrados Puente H L298.

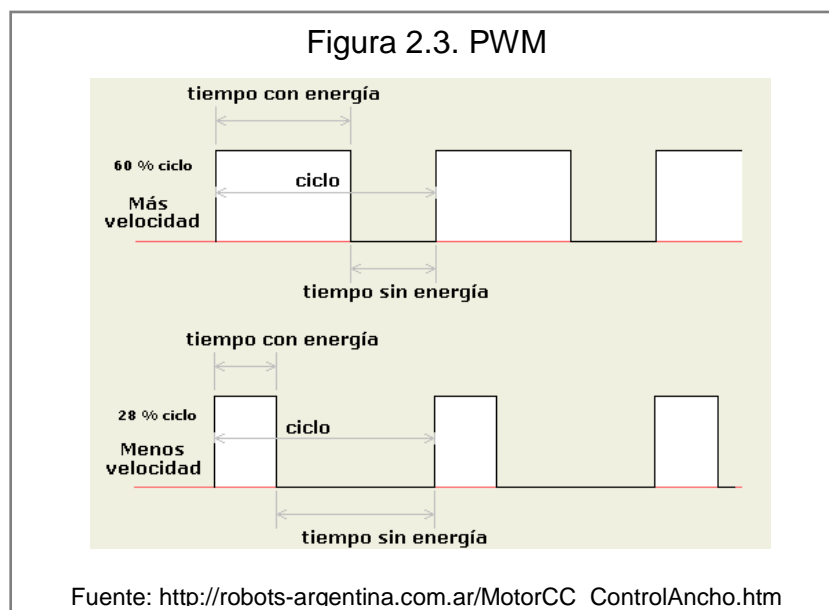
El uso de estos circuitos permitirá invertir la corriente en los motores, permitiendo cambiar su sentido de giro. Su funcionamiento es muy sencillo, ya que como puede observarse, permaneciendo A y D cerrados y B y C abiertos, la corriente circula en un sentido, mientras que si mantenemos abiertos los primeros y cerrados los segundos, el sentido será opuesto.<sup>9</sup>



<sup>8</sup>Todorobot, Motores de Corriente Continua (DC), (<http://www.todorobot.com.ar/documentos/dc-motor.pdf>)

<sup>9</sup> Unicrom, Principio básico de funcionamiento de un motor CC, [en línea], ([http://www.unicrom.com/Tut\\_MotorCC.asp](http://www.unicrom.com/Tut_MotorCC.asp))

**Control de posición por PWM.-** La modulación por anchura de pulso, PWM, es una de los sistemas más empleados para el control de motores. Este sistema consiste en generar una onda cuadrada en la que se varía el tiempo que el pulso está a nivel alto, manteniendo el mismo período, con el objetivo de modificar la posición del motor según se desee.



### 2.2.3. Sensores de Proximidad

En robótica un sensor utiliza un transductor que capta información del entorno para ser utilizada por el robot. Los sensores son de diferentes tipos dependiendo de su funcionamiento y de la forma de adquirir los estímulos externos.

Los sensores de proximidad son elementos que detectan objetos o señales que se encuentran cerca del radio de acción del sensor.<sup>10</sup> En el proyecto se utilizará sensores de ultrasonido y PIR.

<sup>10</sup>Sensores externos, Universidad de Alicante, España, 2009, (<http://www.dccia.ua.es/dccia/inf/asignaturas/ROB/optativos/Sensores/externos.html>)

**Sensores Ultrasónicos.-** Los sensores ultrasónicos tienen como función principal la detección de objetos a través de la emisión y reflexión de ondas acústicas. Los ultrasonidos son vibraciones del aire de la misma naturaleza que el sonido audible pero de una frecuencia más elevada, superior a 20000Hz, no audibles por el oído humano.

Figura 2.4. Sensor Ultrasonido.



Fuente: <http://articulo.mercadolibre.com.ec/>

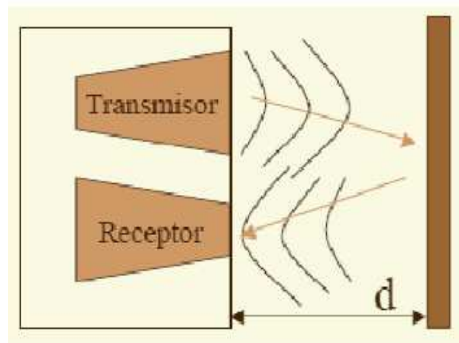
La mayoría de los sensores de ultrasonido se basan en la emisión de un pulso de ultrasonido cuyo lóbulo, o campo de acción, es de forma cónica. Midiendo el tiempo que transcurre entre la emisión del sonido y la percepción del eco se puede establecer la distancia a la que se encuentra el obstáculo que ha producido la reflexión de la onda sonora, utilizando la ecuación 1.4.

Donde  $V$  es la velocidad del sonido en el aire y  $t$  es el tiempo transcurrido entre la emisión y recepción del pulso.<sup>11</sup>

$$d = \frac{1}{2} V * t \quad (\text{Ec. 1.4})$$

<sup>11</sup> Galeon, Sensores de proximidad, España, 2010, (<http://sensoresdeproximidad.galeon.com/#ultrasonico>)

Figura 2.7. Funcionamiento de los sensores de ultrasonido.



Fuente: <http://www.eis.uva.es>.

**Sensor PIR.-** El sensor PIR corresponde a las siglas Pasive Infra Red. Es un dispositivo piro eléctrico (detector de calor). Lo que mide es el cambio de calor, no la intensidad de calor. El calor medido es el calor irradiante cercano al infrarrojo que no es visible. Este sensor detecta movimiento mediante un promedio del calor irradiado en el tiempo.

En el proyecto se empleará el sensor Dyp Me003 por sus características y fácil configuración.

Figura 2.6. Sensor PIR



Fuente: <http://articulo.mercadolibre.com.ec/>



**Sensor PIR Dyp Me003.-** El PIR está fabricado de un material cristalino que genera carga eléctrica cuando se expone a la radiación infrarroja. Los cambios en la cantidad de radiación producen cambios de voltaje que son medidos por un amplificador. Este sensor contiene unos filtros especiales llamados lentes Fresnel que enfocan las señales infrarrojas sobre el elemento sensor.

Cuando las señales infrarrojas del ambiente donde está el sensor cambian, el amplificador activa las salidas, para indicar movimiento esta salida permanece activa durante unos segundos lo que permite que el microcontrolador sepa si es que hubo movimiento. El espectro electromagnético de la radiación infrarroja, tiene una longitud de onda más larga que la luz visible no puede ser vista pero si puede ser detectada y los objetos que generan calor también generan radiación infrarroja. El PIR viene configurado para la detección del cuerpo humano. Es por eso que si uno se queda quieto frente al sensor, este no te detecta más. En teoría si un objeto que no emite calor se mueve el sensor no lo detectaría, por ejemplo un vaso rodando.<sup>12</sup>

#### **2.2.4. Tecnología Zigbee**

Zigbee es un protocolo de comunicaciones inalámbrico basado en el estándar de comunicaciones para redes inalámbricas IEEE 802.15.4. Creado por Zigbee Alliance, una organización, teóricamente sin ánimo de lucro, de más de 200 grandes empresas. Zigbee permite que dispositivos electrónicos de bajo consumo puedan realizar sus comunicaciones inalámbricas. Es especialmente útil para redes de sensores en entornos

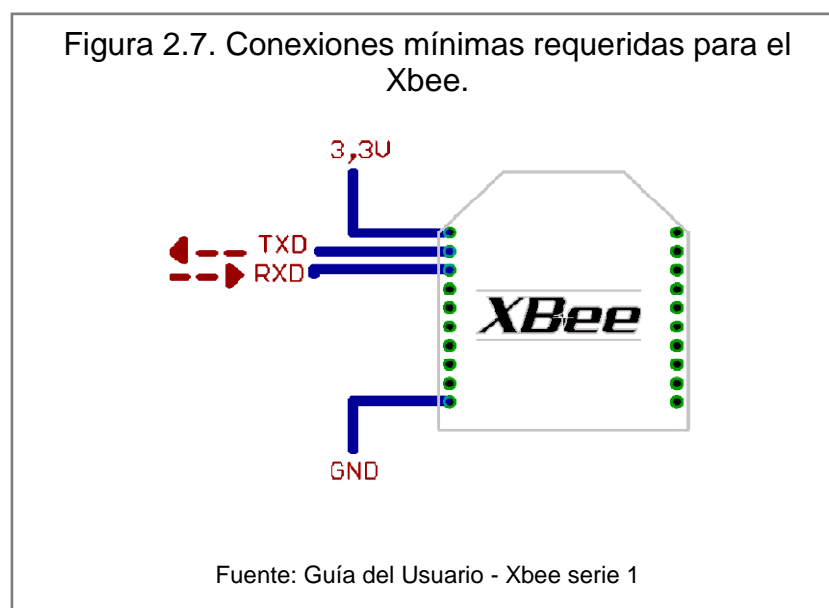
---

<sup>12</sup> Preller, Benjamín, Sensor PIR, (<http://www.slideshare.net/benjapreller/sensor-de-movimiento-pir>)

industriales, médicos y, sobre todo, domóticos. Las comunicaciones Zigbee se realizan en la banda libre de 2.4GHz. El alcance depende de la potencia de transmisión del dispositivo así como también del tipo de antenas utilizadas. La velocidad de transmisión de datos de una red Zigbee es de hasta 256kbps. Una red Zigbee la pueden formar, teóricamente, hasta 65535 equipos, es decir, el protocolo está preparado para poder controlar en la misma red esta cantidad enorme de dispositivos. Entre las necesidades que satisface el módulo se encuentran:

- Bajo costo.
- Ultra-bajo consumo de potencia.
- Uso de bandas de radio libres y sin necesidad de licencias.
- Instalación barata y simple.
- Redes flexibles y extensibles.

**Circuito básico para el Xbee.-** La figura 2.7 muestra las conexiones mínimas que necesita el módulo Xbee para poder ser utilizado. Luego de esto, se debe configurar según el modo de operación adecuado para la aplicación requerida por el usuario.



El módulo requiere una alimentación desde 2.8 a 3.4 V, la conexión a tierra y las líneas de transmisión de datos por medio del UART (TXD y RXD) para comunicarse con un microcontrolador, o directo a un puerto serial utilizando algún conversor adecuado para los niveles de voltaje.

**Modos de Operación.-** Los Xbee pueden operar en los siguientes modos:

- Modo Recibir / Transmitir
- Modo de Bajo Consumo (Sleep Mode)
- Modo de Comando.
- Modo Transparente

Para este proyecto se ha seleccionado el Modo de Operación Transparente, conexión Punto a Punto.

**Modo Transparente: Punto a Punto.-** El modo Transparente viene por defecto en los módulos Xbee. Este modo está destinado principalmente a la comunicación punto a punto, donde no es necesario ningún tipo de control. También se usa para reemplazar alguna conexión serial por cable, ya que es la configuración más sencilla posible y no requiere una mayor configuración. Sólo se debe configurar la dirección. Para ello se utilizan los comandos MY y el DL. En este modo, el módulo receptor del mensaje envía un paquete al módulo de origen llamado ACK que indica que el mensaje se recibió correctamente. Una vez configurado, el módem se encuentra listo para funcionar. Así todo lo que se transmite por el pin DIN de un módulo, es recibido por el pin DOUT del otro. Para que el modo Punto a Punto funcione, los módulos deben pertenecer a la misma PAN ID y al mismo canal.<sup>13</sup>

---

<sup>13</sup> Oyarceç, Andrés, Guía del Usuario -Xbee serie 1, (E-book)

#### 2.2.4. Arduino

Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar. Arduino puede tomar información del entorno a través de sus pines de entrada de toda una gama de sensores y puede afectar aquello que le rodea controlando luces, motores y otros actuadores. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino. Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a un ordenador, si bien tienen la posibilidad de hacerlo y comunicarse con diferentes tipos de software.

Figura 2.8. Arduino Arduclema.



Fuente: <http://arduino.cc/en/Main/arduinoBoard>

El Arduino Arduclema es una placa electrónica basada en Arduino Diecimila. Emplea el Atmega328P, contiene todo lo necesario para apoyar el microcontrolador, basta con conectarlo a un ordenador con un cable USB o con un adaptador AC-DC o batería para empezar. Incorpora unas clemas de tornillo para conexiones eléctricas más robustas y fiables. Otra característica del Arduclema es que se puede activar las resistencias de pull-up, para las conexiones I2C, mediante unos jumpers.

**Potencia.-** La tarjeta puede funcionar con un suministro externo de 6 a 20 voltios. Con menos de 7V, sin embargo, el pin de 5V puede suministrar menos de 5V. Si se utiliza más de 12V, el regulador de voltaje se puede sobrecalentar y dañar la placa. El rango recomendado es de 7 a 12 voltios.<sup>14</sup>

### 2.2.5. Herramientas para el desarrollo del prototipo

Para las diferentes funciones que realiza el prototipo existen programas que facilitan el desarrollo de las mismas. Las principales herramientas de software para el desarrollo que hemos utilizado son:

**Software de Arduino.-** El entorno de código abierto Arduino hace fácil escribir código y cargarlo a la placa E/S. Funciona en Windows, Mac OS X y Linux. El entorno está escrito en Java y basado en Processing, avr-gcc y otros programas también de código abierto.<sup>15</sup>

**Proteus.-** Es un entorno integrado diseñado para la realización completa de proyectos de equipos electrónicos en todas sus etapas: diseño, simulación y construcción; desarrollado por Labcenter Electronics. Está formada por diversas herramientas.<sup>16</sup>

**Visual Basic 2010 Express.-** Es un entorno de desarrollo integrado (IDE) para sistemas operativos Windows. Soporta varios lenguajes de programación; permite a los desarrolladores crear sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET. Así se pueden crear

---

<sup>14</sup> Página Oficial Arduino, (<http://www.arduino.cc/es/>)

<sup>15</sup> Página Oficial Arduino, (<http://www.arduino.cc/es/>)

<sup>16</sup> Ingeniería Eléctrica Electrónica, Proteus, España, (<http://www.ieespain.com/ieeproteus/>)

aplicaciones que se intercomunican entre estaciones de trabajo, páginas web y dispositivos móviles. Es de carácter gratuito y es proporcionado por la compañía Microsoft Corporation orientándose a principiantes, estudiantes y aficionados de la programación web y de aplicaciones.<sup>17</sup>

**XCT-U.-** Este es el programa con el que se configuran los módulos Xbee, no solo eso, también contiene un terminal con el cual poder mandar y recibir datos mediante el puerto que está conectado el xbee.<sup>18</sup>

**SQL Server 2008 Express.-** Es una edición gratuita de SQL Server y es una plataforma de datos ideal para aprender y ejecutar pequeñas aplicaciones de servidor y de escritorio, y para su redistribución a través de ISV. Microsoft SQL Server 2008 Express es un sistema de administración de datos eficaz y confiable que ofrece un variado conjunto de características, protección de datos y rendimiento para clientes de aplicaciones incrustadas, aplicaciones web ligeras y almacenes de datos locales, está diseñado para una implementación sencilla, disponible de forma gratuita y su redistribución con aplicaciones también es gratuita. Está diseñado para integrarse a la perfección con otras inversiones de infraestructura de servidor.<sup>19</sup>

**MATLAB.-** (Matrix Laboratory, "laboratorio de matrices") es una herramienta de software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M).<sup>20</sup>

---

<sup>17</sup> Microsoft Visual Studio, <http://www.microsoft.com/visualstudio/esn/downloads#d-2010-express>

<sup>18</sup> XCT-U, <http://www.tutoelectro.com/tutoriales/otros/x-ctu/>

<sup>19</sup> Microsoft SQL Server Express (<http://www.microsoft.com/es-es/download/details.aspx?id=1695>)

<sup>20</sup> Wikipedia, *Matlab*, (<http://es.wikipedia.org/wiki/MATLAB>)

### 2.3. Variables

Se detalla la operacionalización de las variables.

Tabla 2.1. Variable independiente

Variable independiente: Prototipo de robot móvil con visión nocturna para vigilancia y seguridad.				
Variable	Definición	Dimensiones	Indicadores	Instrumento
Prototipo de robot móvil con visión nocturna para vigilancia y seguridad	Son robots con capacidad de desplazamiento, basada en carros o plataformas y dotada de un sistema locomotor de tipo rodante. Siguen su camino por telemando o desplazamiento basado en respuesta a la programación del microcontrolador y la detección de objetos por medio de sensores con un nivel relativo de inteligencia.	Sistema de Locomoción	Eficiencia del Modelo Cinemático del robot con tracción diferencial.	Técnicas: - Observación directa. - Observación Indirecta. - Recolección de datos.  Instrumentos: -Planeamiento en Matlab de movimiento del robot.  -Simulación de circuito en Proteus.
		Sensores de Proximidad	La efectividad de detección de personas.	
		Nivel de control	Nivel de Inteligencia con que reacciona el prototipo.	
		Gestión remota	Nivel de transmisión del módulo de comunicación	

Tabla 2.2. Variable dependiente

Variable dependiente: Mejorar el control de seguridad y monitoreo.				
Variable	Definición	Dimensiones	Indicadores	Instrumento
Mejorar el control de seguridad y monitoreo	Es una tecnología diseñada para monitorear personas, objetos o bienes garantizando resguardo, cumpliendo con una inspección del lugar asignado.	Niveles de vigilancia y monitoreo de las rondas de guardia de seguridad.	Cantidad de rondas de guardia realizadas por el robot autónomo.	Técnicas: - Observación directa. - Observación Indirecta. - Encuestas
			Cantidad de veces que se ha utilizado el robot para el monitoreo remoto.	

## 2.4. Términos Básicos

RMR.- Robots móviles con ruedas.

Sensar.- Medir una condición.

MY.- La dirección de 16-bit del módulo Xbee.

SH y SL.- El número serie del módulo Xbee.

DL.- La dirección de destino para las comunicaciones inalámbricas.

ACK.- (Acknowledgment) Acuse de recibo o asentimiento, en comunicaciones entre computadores, es un mensaje que el destino de la comunicación envía al origen de ésta para confirmar la recepción de un mensaje

PAN ID.- Identificación Red de Área (Personal Area Network)

Processing.- Es un lenguaje de programación y entorno de desarrollo integrado de código abierto basado en Java, de fácil utilización, y que sirve como medio para la enseñanza y producción de proyectos multimedia e interactivos de diseño digital. Fue iniciado por Ben Fry y Casey Reas a partir de reflexiones en el Aesthetics and Computation Group del MIT Media Lab dirigido por John Maeda. Se distribuye bajo la licencia GNU GPL.



## **CAPÍTULO III**

### **ANÁLISIS**

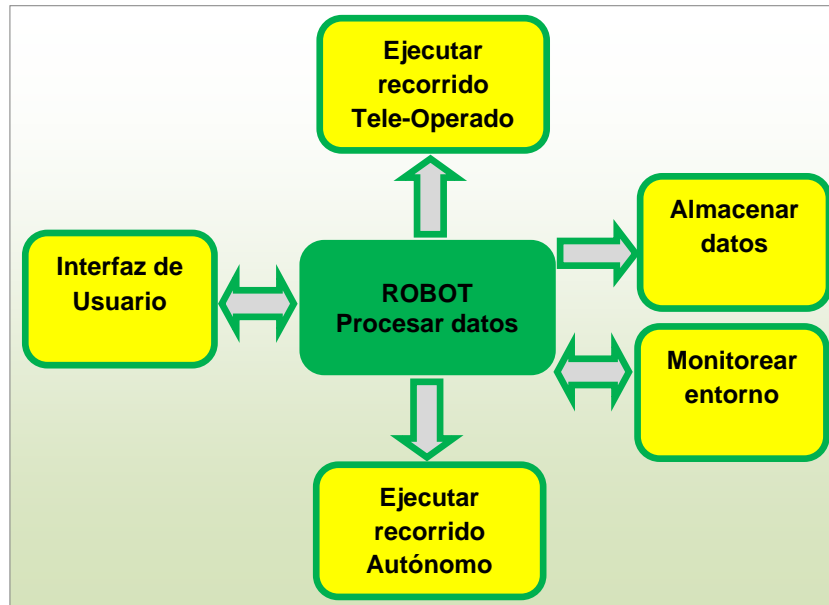
#### **3. ANÁLISIS**

Este capítulo se refiere a la etapa de análisis del proyecto, se detalla los requerimientos, especificaciones técnicas, procedimientos y viabilidad del prototipo.

##### **3.1. Diagrama del proceso**

Como todo dispositivo funcional, los robots están formados por diferentes procesos que se complementan para su correcto funcionamiento.

Figura 3.1. Diagrama de procesos del prototipo de Robot Móvil.



### 3.1.1. Descripción funcional de los procesos

En esta sección se describe cada uno de los procesos que forman parte del proyecto.

**La interfaz de usuario.-** Para fácil comprensión, la interfaz es intuitiva; por lo que cuenta con botones y flechas, además de detallar con textos cerca de su ubicación describiendo su tarea. Los botones de funciones tienen asociados tareas específicas; enviando comandos al prototipo a través del módulo de comunicación.

Estas órdenes enviadas desde la interfaz permiten seleccionar el modo de operación del prototipo de robot móvil, como también manipular los movimientos del robot, encender y visualizar la imagen captada por la cámara, y activar los sensores que detentan a intrusos en movimiento.

**Procesar datos.-** El usuario define el modo de operación del robot móvil vía software, este recibe la información de forma remota, el microcontrolador de la tarjeta Arduino procesa el comando y actúa en uno de los dos modos de uso; el modo teleoperado cumpliendo de forma remota las ordenes del usuario, o el modo de ejecución de recorrido, monitoreo y vigilancia autónomo.

**Ejecutar recorrido Tele-Operado.-** En este proceso el usuario se encargará de manipular el prototipo de robot para vigilancia de forma remota. En esta opción podrá hacer avanzar, regresar, girar al robot, encender la cámara, y encender los sensores que detectan intrusos en movimiento, en este modo está disponible la opción autónomo haciendo que el prototipo se ubique en un lugar deseado por el operario y dejándolo monitorear el área de forma estática.

**Ejecutar recorrido Autónomo.-** Cuando el usuario selecciona este modo de operación; el prototipo de robot de vigilancia realiza el recorrido de la trayectoria programada para el área determinada de forma autónoma, cumplirá un recorrido de guardia, para la detección y alarma de intruso se utiliza el sensor de ultrasonido colocado estratégicamente al frente y de los sensores PIR; estos sensores permiten informar las condiciones del entorno al sistema para permitir una navegación sin inconvenientes.

**Monitorear entorno.-** Este proceso se realizará cuando el prototipo se encuentre ejecutando el recorrido autónomo en la trayectoria programada, el robot cada cierto espacio recorrido se detendrá para monitorear el área determinada empleando los sensores PIR para detectar la presencia de personas en movimiento; y también se lo podrá seleccionar en el modo tele-

operado cuando el usuario active la función de los sensores, el prototipo deberá estar detenido para su correcto funcionamiento.

**Almacenar datos.-** Este procedimiento se realizará cuando el usuario acceda a la interfaz gráfica de control del prototipo en Visual; se almacenará los datos de horarios de uso del robot como también las alarmas generadas por detección de intruso u objeto; en una base de datos en SQL Server Express 2008, donde posteriormente se podrá generar un informe con los datos del uso y las notificaciones de las detecciones.

### **3.2. Identificación de requerimientos**

El desarrollo del proyecto está fundamentado en los objetivos generales y específicos comentados al inicio de este proyecto y debe ajustarse a los siguientes requisitos para su óptimo funcionamiento:

- El desarrollo del sistema de control del robot estará basado en el Software Arduino, un lenguaje de alto nivel bajo los entornos de programación Java.
- El programa deberá ser capaz de controlar los motores del sistema robótico, generando movimientos como avanzar, retroceder, girar y parar, además de controlar los niveles de potencia suministrados a los motores basándose en técnicas PWM por software.
- El prototipo podrá ser controlado por una interfaz de usuario en Visual Basic 2010 que permita el modo de operación del prototipo; autónomo o tele-operado.

- El prototipo de robot deberá recorrer la trayectoria programada para el área del laboratorio de electrónica de la Escuela de Electrónica y Telecomunicaciones de la Universidad Estatal Península de Santa Elena, censando su entorno para cumplir el recorrido eficientemente, en una superficie lisa, sin obstáculos y sin personas; excepto los casos de detección.
- Debe contener un algoritmo que basado en las lecturas del sensor ultrasónico permita reconocer un obstáculo en forma parcial o total dependiendo de las limitaciones propias del hardware, deberá encender la cámara para que el usuario encargado adopte precauciones oportunas.
- Debe reaccionar al detectar el movimiento de un intruso en el área monitoreada; encendiendo la cámara para que el usuario encargado pueda tomar las medidas necesarias al respecto.
- Generará informes del uso del prototipo de control de vigilancia y monitoreo y las notificaciones de detección de intrusos.

De los requisitos encontrados se pudo desarrollar el análisis de los posibles procesos y parámetros que se debía seguir para la culminación del proyecto.

### **3.3. Análisis del sistema**

El propósito de realizar este análisis es para dar a conocer la viabilidad del proyecto, los costos reales y precisos de la construcción e implementación del prototipo. Así como analizar los factores tecnológicos y de operatividad del presente prototipo de robot móvil.

### 3.3.1. Análisis técnico

Permite conocer los requerimientos técnicos que se necesita para el desarrollo del prototipo, siendo la Universidad Estatal Península de Santa Elena, la que proporciona las herramientas tecnológicas de software pues se utilizarán los equipos computacionales del laboratorio.

La elaboración del prototipo de robot móvil contará con personal inmerso en la electrónica y con capacidad técnica para diseñar e implementar el proyecto para vigilancia y seguridad. Además de todo el apoyo técnico que será proporcionado por parte del tutor. A continuación se detalla el hardware, software, personal y otros recursos que se necesita para el desarrollo del proyecto.

Tabla 3.1. Requerimientos de Suministros para el prototipo.

Recurso tecnológico	Descripción	Cant
<b>Requerimiento de suministros</b>	Paquete resma formato A4	4
	Tinta para impresora	4
	Anillados	3
	Impresora MP280	1
	Gastos varios	1

Tabla 3.2. Requerimientos de software para el prototipo.

Recurso tecnológico	Descripción	Cant
<b>Software</b>	Sistema operativo Windows 7 Ultimate	1
	Proteus ISIS Profesional (N1)	1
	Matlab versión estudiante	1
	Microsoft Visual Basic Express 2010	1
	SQL Server Express 2008	1
	Arduino V 1.0.5	1
	Software de configuración XCTU v.5.1.4	1

Tabla 3.3. Requerimiento de hardware para programación.

Recurso tecnológico	Descripción	Cant
<b>Hardware</b>	Para realizar la programación: Computadora con 500 GB de disco duro, procesador Dual Core, 2GB Ram mínimo, acelerador gráfico.	1

Tabla 3.4. Requerimientos de hardware para el prototipo.

Recursos tecnológicos	Descripción	Cant
<b>Estructura Base</b>	Estructura mecánica.	1
	Kit Llantas Sports Tamiya 56mmx25mm	1
	Tornillos y pernos	1
	Silicón transparente y cinta	1
	Tacho plástico metalizado	1
<b>Componentes Electrónicos</b>	Motores DC 12V	4
	Batería 12V / 2A	1
	Batería 9V 200mA	1
	Sensor ultrasonido SFR04	1
	Sensores PIR DYP-ME003	2
	Tarjeta capturadora de video EasyCam	1
	Cámara inalámbrica.	1
	Módulos de comunicación XBee S1.	2
	Adaptador USB para Xbee	1
	Adaptador TTL para Xbee	1
	Arduino Arduclema	1
	Tarjeta puente H L298N	1
	Disipador metálico TO220	3
	Relés 12V y 5V	2
	Regulador 7805, 7808	2
	Potenciómetro 10K	3
	Resistencias 33k, 1k, 470	8
	Transistores 2N3904	5
	Capacitores electrolítico 10uF, 1uF	4
	Capacitores cerámico 100nF	2
	Diodos IN4007	4
	Diodos Zener 7,5V	3
	Led's verde, rojo	2
	Baquelita	1
	Conectores macho-hembra	10
	Botón On/Off	2
	Borneras plásticas	1
	Fusible de vidrio 2A	1
	Porta fusible	1
	Cable eléctrico	1
	Cable conector batería	1
	Caimanes	2
Adhesivo logos	2	

### 3.3.2. Análisis económico

En este análisis se detallarán los costos que se necesitan en cada etapa del proyecto tanto en la investigación como en el diseño e implementación incluyendo los costos de hardware, software, suministros y todos los gastos previstos para elaborar el prototipo del robot móvil. La Universidad Estatal Península de Santa Elena facilitó sus laboratorios con el hardware y software necesario para el desarrollo de la programación y control del prototipo.

Tabla 3.5. Costo requerimientos de suministros para el prototipo.

Descripción	Cant	V.Uni	Valor
Paquete resma formato A4	4	\$ 4,50	\$ 18,00
Tinta para impresora	4	\$ 10,00	\$ 40,00
Anillados	3	\$ 1,00	\$ 3,00
Impresora MP280	1		\$ 120,00
Gastos varios	1		\$ 100,00
<b>Total</b>			<b>\$ 281,00</b>

Tabla 3.6. Costo requerimientos de software para el prototipo.

Descripción	Cant	Valor
Sistema operativo Windows 7 Ultimate	1	\$ 150
Proteus ISIS Profesional (N1)	1	\$ 450
Matlab versión estudiante	1	\$ 100
Microsoft Visual Basic Express 2010 *	1	\$ 0
SQL Server Express 2008 *	1	\$ 0
Arduino V1.0.5**	1	\$ 0
Software de configuración XCTU v.5.1.4 **	1	\$ 0
<b>Total</b>		<b>\$ 700</b>

Tabla 3.7. Costo requerimiento de hardware para programación.

Descripción	Cant.	Valor
Para realizar la programación: Computadora con 500 GB de disco duro, procesador Dual Core, 2GB Ram mínimo, acelerador gráfico.	1	\$ 500
<b>Total</b>		<b>\$ 500</b>



Tabla 3.8. Costo requerimientos de hardware para el prototipo.

Descripción	Cant	V.Uni	Valor
Estructura mecánica.	1		\$ 10,00
Kit Llantas Sports Tamiya 56mmx25mm	1		\$ 20,00
Tornillos y pernos	1		\$ 3,00
Silicón transparente y cinta	1		\$ 5,00
Tacho plástico metalizado	1		\$ 6,00
Motores DC 12V	4	\$50,00	\$ 200,00
Batería 12V / 2A	1		\$ 15,00
Batería 9V 200mA	1		\$ 10,00
Sensor ultrasonido SFR04	1		\$ 10,00
Sensores PIR DYP-ME003	2	\$10,00	\$ 20,00
Tarjeta capturadora de video EasyCam	1		\$ 25,00
Cámara inalámbrica.	1		\$ 80,00
Módulos de comunicación XBee S1.	2	\$40,00	\$ 80,00
Adaptador USB para Xbee	1		\$ 30,00
Adaptador TTL para Xbee	1		\$ 25,00
Arduino Arduclema	1		\$ 30,00
Tarjeta puente H L298N	1		\$ 10,00
Disipador metálico TO220	2	\$ 1,00	\$ 2,00
Relés 12V y 5V	2	\$ 1,15	\$ 2,30
Regulador 7805, 7808	2	\$ 1,15	\$ 2,30
Potenciómetro 10K	3	\$ 1,00	\$ 3,00
Resistencias 33k, 1k, 470	8	\$ 0,05	\$ 0,40
Transistores 2N3904	5	\$ 0,30	\$ 1,50
Capacitores electrolítico 10uF, 1uF	4	\$ 0,50	\$ 2,00
Capacitores cerámico 100nF	2	\$ 0,30	\$ 0,60
Diodos IN4007	4	\$ 0,25	\$ 1,00
Diodo Zener 7,5V	3	\$ 0,25	\$ 0,75
Led's verde, rojo	3	\$ 0,20	\$ 0,60
Baquelita	1		\$ 3,00
Conectores macho-hembra	10	\$ 0,50	\$ 5,00
Botón On/Off	2	\$ 1,25	\$ 2,50
Borneras plásticas	1		\$ 0,50
Fusible de vidrio 2A	1		\$ 0,15
Porta fusible	1		\$ 0,50
Cable eléctrico	1		\$ 2,00
Cable conector batería	1		\$ 2,00
Caimanes	2	\$ 0,50	\$ 1,00
Adhesivo logos	2		\$ 3,00
<b>Total</b>			<b>\$ 605</b>

Tabla 3.9. Costo total para el desarrollo del prototipo.

Descripción	Valor
Software	\$ 700
Hardware programación	\$ 500
Hardware prototipo	\$ 605
Suministros	\$ 281
<b>Total</b>	<b>\$ 2086</b>

El Costo Total de este proyecto es de \$ 2086.

\*Herramientas distribuidas bajo licencia ISV.

\*\*Herramientas distribuidas en la página oficial del producto

### 3.3.3. Análisis operativo

Luego de haber presentado la viabilidad técnica y económica se realizará a continuación un análisis operativo el cual deberá reflejar de qué manera contribuye y el grado de aceptación que tendrá su implementación.

El proyecto es factible operativamente debido a que para poder utilizarlo no necesita un alto nivel de conocimientos en robótica, según los estudios obtenidos en la investigación de campo mediante los resultados que proyecta la encuesta realizada al personal de seguridad que se muestra en el apartado de análisis de resultado de la encuesta; se comprueba que la mayoría de los guardias aceptarían el uso de este prototipo de robot móvil con visión nocturna para vigilancia; y les serviría de ayuda para mejorar el control de seguridad y monitoreo en el área del laboratorio de la Escuela de Electrónica y Telecomunicaciones de la Universidad Estatal Península de Santa Elena.

### **3.4. Métodos e Instrumentos de investigación**

En la estrategia metodológica se busca por medio de técnicas y procedimientos realizar el proyecto de investigación de la forma más eficaz y eficiente, teniendo en claro los procesos a realizarse en cada etapa, y así poder cumplir los objetivos propuestos.

#### **3.4.1. Metodología de la investigación**

Para la implementación del prototipo de robot para vigilancia y seguridad, con el objetivo de obtener resultados confiables, fidedignos, y extraídos de la realidad, se toma en cuenta el siguiente método.

#### **Método Hipotético-Deductivo**

Esta metodología consiste en hacer observaciones manipulativas y análisis, a partir de las cuales se formula la hipótesis que será comprobada mediante experimentos controlados. Las fases 1, 4, y 5 requirieron de la experiencia, es decir, es un proceso empírico; mientras que las fases 2 y 3 son racionales. Este método tuvo incidencia en el desarrollo de este proyecto.

#### **Fases del método hipotético-deductivo**

1. Detectar el problema, mediante la observación se determinó el problema y se tomó los datos significativos sobre el mismo; la tecnología robótica utilizada para mejorar el control de seguridad en las rondas nocturnas de vigilancia es inexistente en el laboratorio de la Escuela de Electrónica y Telecomunicaciones de la Universidad Estatal Península de Santa Elena.
2. Proponer hipótesis, se dedujo una hipótesis general que regirá esta tesis; la implementación del prototipo de robot móvil con visión nocturna para vigilancia y seguridad ayudará al cuerpo

de guardia a mejorar el control de vigilancia y monitoreo en el área del laboratorio de la Escuela de Electrónica y Telecomunicaciones de la Universidad Estatal Península de Santa Elena.

3. Conformar un marco teórico preliminar; los planteamientos teóricos presentados en este documento fueron las bases para realizar la investigación, proporcionando información sobre el problema estudiado.
4. Sistematizar solución del problema con ayuda de los medios identificados; se investigó y analizó la estructura del diseño físico y de control del robot para dar estabilidad al prototipo. Se estableció el sistema de locomoción y visión del prototipo para ejecutar las rondas en el área determinada. Se diseñó las tarjetas electrónicas para acoplar el sistema de potencia con el sistema de locomoción y control. Se determinó el software para el control y monitoreo del robot móvil con visión nocturna. Se implementó el prototipo de robot móvil con visión nocturna para el control de vigilancia y monitoreo. Por medio de la sistematización de los objetivos se logró solucionar el problema con la construcción del prototipo de robot móvil con visión nocturna para vigilancia y seguridad mediante el uso de tecnología robótica para el control de vigilancia y monitoreo en el área del laboratorio de la Escuela de Electrónica y Telecomunicaciones de la Universidad Estatal Península de Santa Elena.
5. Verificación de la verdad de los enunciados deducidos comparándolos con la experiencia; mediante el uso de las encuestas y de la realización de pruebas se comprobó la solución, confrontando ésta con la totalidad de los objetivos

planteados y de la información empírica pertinente. El resultado es satisfactorio, la investigación se da por concluida; la hipótesis resultó corroborada por la experiencia.

#### **3.4.2. Instrumentos de recolección de información**

Existen varios instrumentos de recolección de información, sin embargo los que se aplicarán en el proyecto son observaciones, consultas bibliográficas y encuestas referentes al tema de estudio.

La observación directa; esta técnica permite conseguir información que no se puede obtener de alguna otra forma pues permite ver el escenario real. Consiste en analizar la situación actual del problema, se la realizó en la universidad, como también en el laboratorio de electrónica universitario, con el fin de obtener los parámetros de operación del robot móvil, tales como: trayectoria de la ruta, distancia de recorrido y obstáculos previstos, datos que son necesarios para el sistema a implementar. De esta manera se pudo determinar los parámetros para el correcto funcionamiento del prototipo.

También se utiliza fuentes de investigación primaria y secundaria. Al referirse a fuentes primarias, se trata de trabajar con información de primera mano; y fuentes secundarias para la obtención de información a través de publicaciones, resúmenes, biblioteca virtual, páginas de internet, libros y proyectos ya realizados acerca del prototipo a implementar, lo que permitió tener una visión más amplia del tema.

De igual forma se utilizan las encuestas para conocer la situación actual del control de seguridad en la universidad en el área del

laboratorio; como también conocer si los guardias y personas encargadas de la seguridad de la universidad se adaptarían al uso del prototipo.

### 3.5. Análisis e interpretación de los resultados de encuesta

Una vez conocido el planteamiento del problema, realizamos un diagnóstico de la situación actual en la universidad a través de una encuesta para conocer si los guardias y personas encargadas de la seguridad de la universidad se adaptarían y le serviría un robot móvil con visión nocturna para vigilancia y seguridad mediante el uso de tecnología en robótica para control de vigilancia y monitoreo en el área del laboratorio de la Escuela de Electrónica y Telecomunicaciones de la Universidad Estatal Península de Santa Elena. Para obtener la muestra de la población a determinar para la encuesta recogimos los siguientes datos:

- Tamaño total de la población número total de trabajadores encargados de la seguridad en la Universidad  $N= 29$ .
- Error muestral deseado ( $e$ ) = 9%

Con estos datos y con la siguiente ecuación se calculó el número de trabajadores de la muestra ( $n$ ) para la encuesta.

$$n = \frac{N}{e^2(N - 1) + 1}$$

Dónde:

$N$ =Tamaño de la población, 29

$e$ = Error de estimación, 5%

$n$ = Tamaño de la muestra.

$$n = \frac{29}{0.05^2(29 - 1) + 1}$$

$$n = \frac{29}{1,07}$$

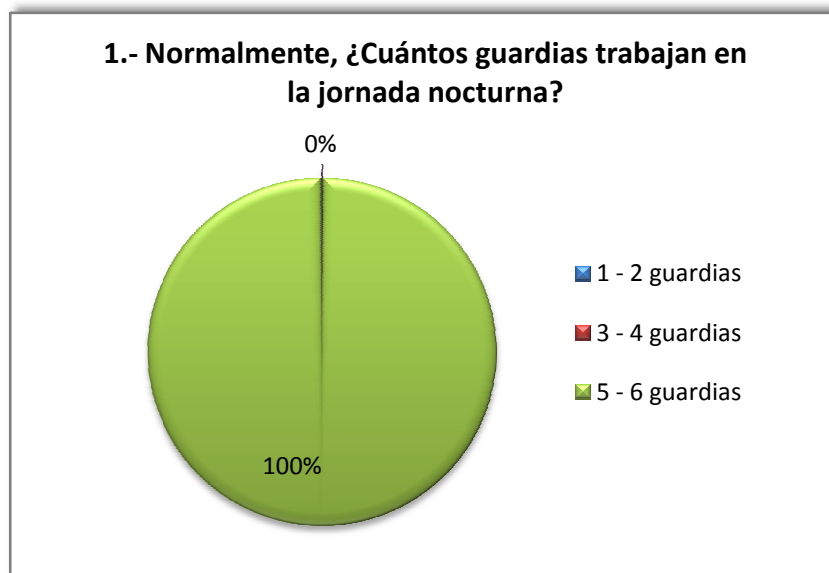
$$n = 27$$

Cuando la población es muy pequeña y el error tolerado muy pequeño, prácticamente hay que tomar a toda o casi toda la población. Por lo que para la encuesta realizada se tomo una muestra de 27 guardias.

### Resultados de la encuesta

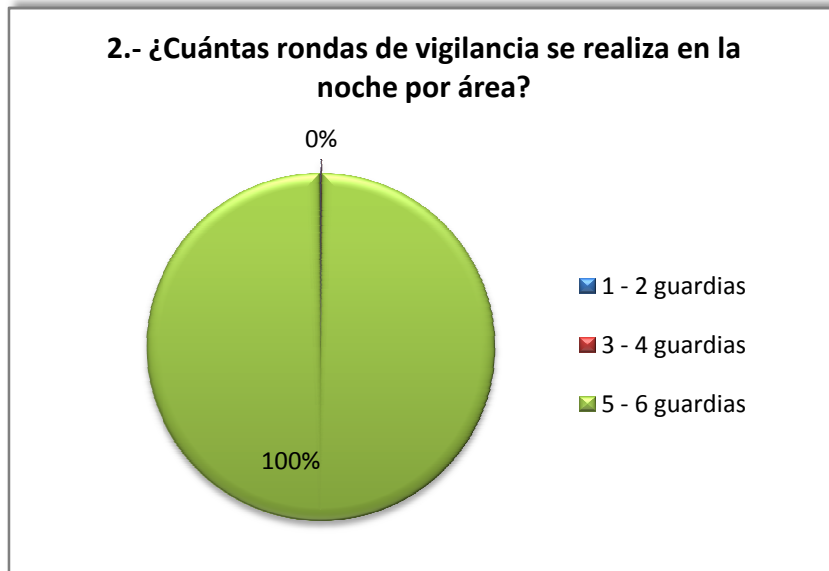
Diagnóstico de la situación actual en la Universidad para la implementación del prototipo de robot móvil con visión nocturna para vigilancia y seguridad. Formato de Encuesta Anexo 1.

Figura 3.2. Gráfico N° 1



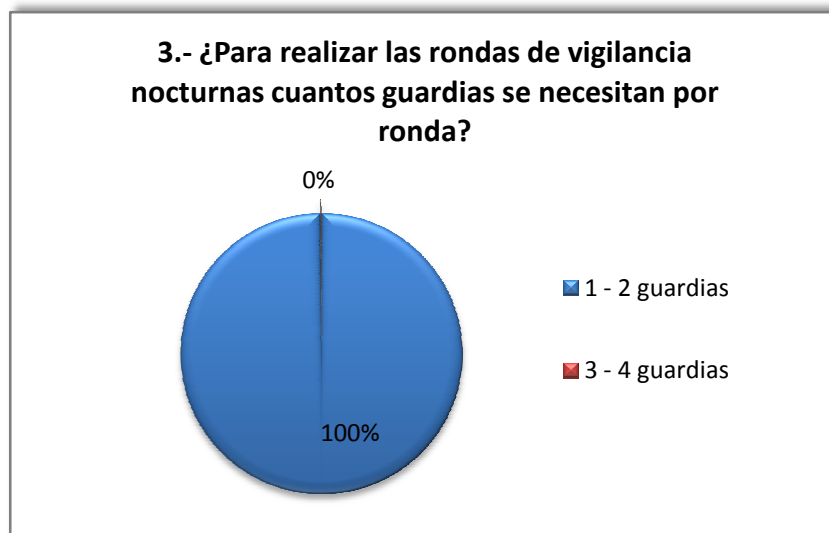
Tabulando los datos de las encuestas se determinó con esta pregunta que normalmente en la jornada nocturna se encuentran laborando cinco o seis guardias en la Universidad Estatal Península de Santa Elena.

Figura 3.3. Gráfico N° 2



Con esta pregunta se conoció que en la Universidad Península de Santa Elena se realizan entre cinco y seis rondas de vigilancia por área.

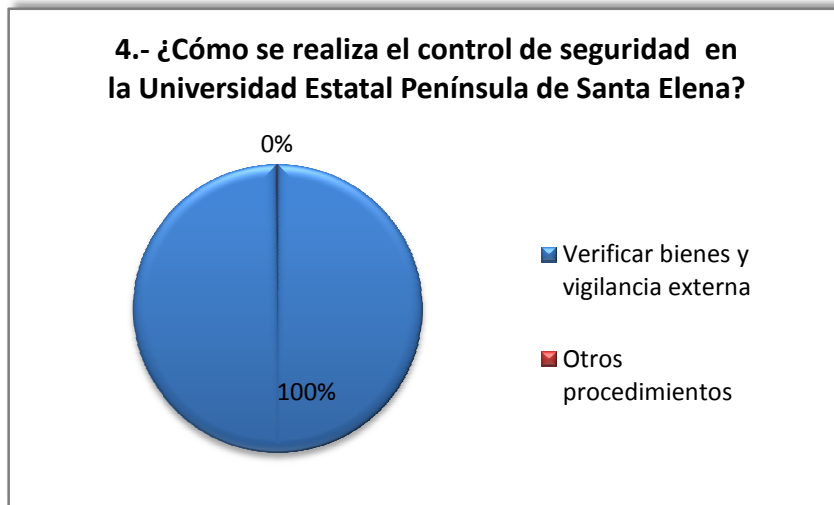
Figura 3.4. Gráfico N° 3



Se determinó que para realizar una ronda de vigilancia en un área se emplea entre 1 a 2 guardias.

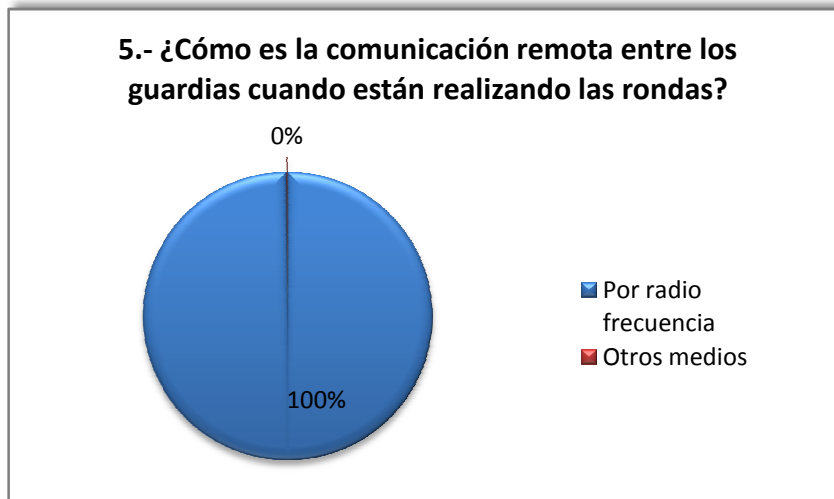


Figura 3.5. Gráfico N° 4



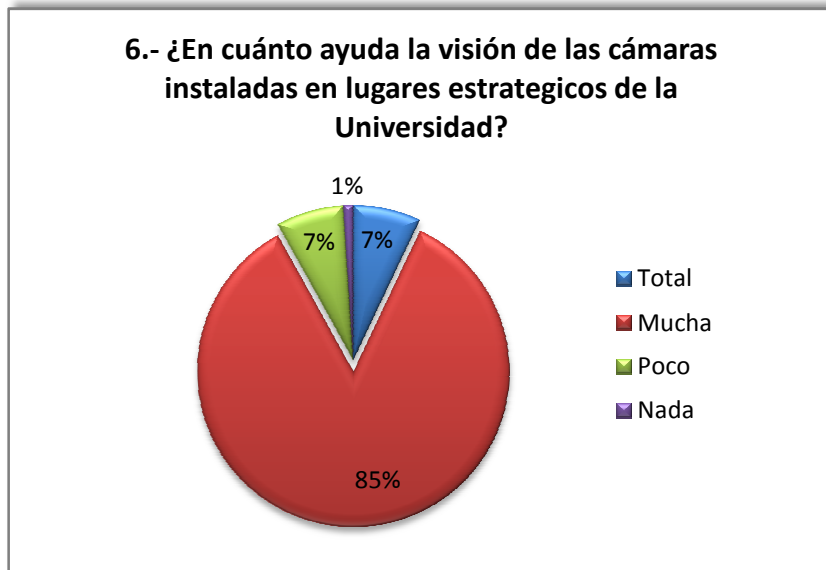
Se interpreta que para el control de seguridad en la Universidad Estatal Península de Santa Elena una vez que hayan culminado las clases; se verifica los bienes de las aulas, se detecta que los bienes de la institución estén completos, se procede a cerrar las aulas; se reporta las novedades y se inicia con la vigilancia externa de los edificios que conforman la universidad.

Figura 3.6. Gráfico N° 5



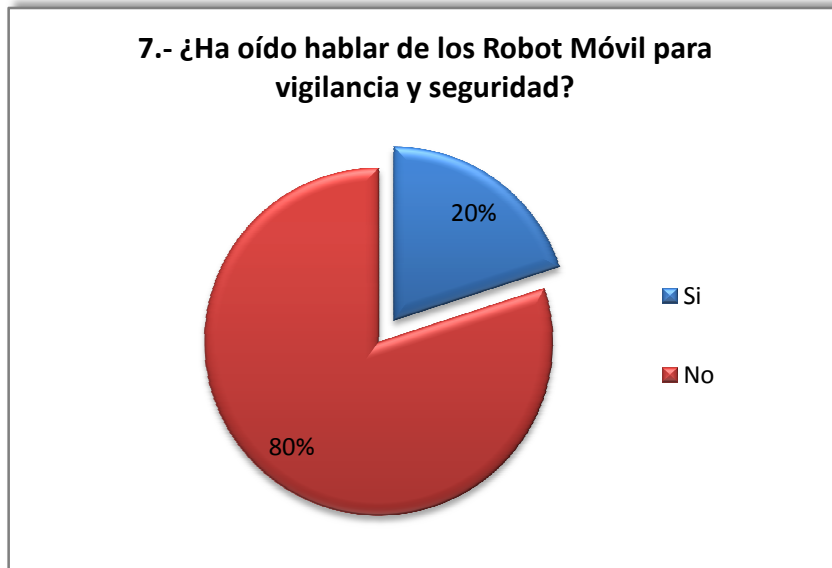
Con esta pregunta se conoció que en la Universidad Península de Santa Elena se realiza la comunicación entre los guardias por radio frecuencia cuando están realizando las rondas.

Figura 3.7. Gráfico N° 6



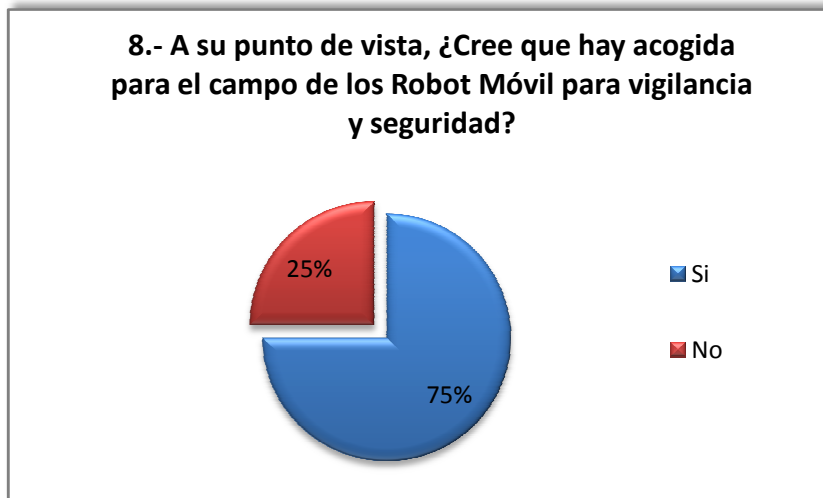
Se determinó que la visión de las cámaras instaladas ayuda en un 85% a realizar las guardias en la Universidad Península de Santa Elena.

Figura 3.8. Gráfico N° 7



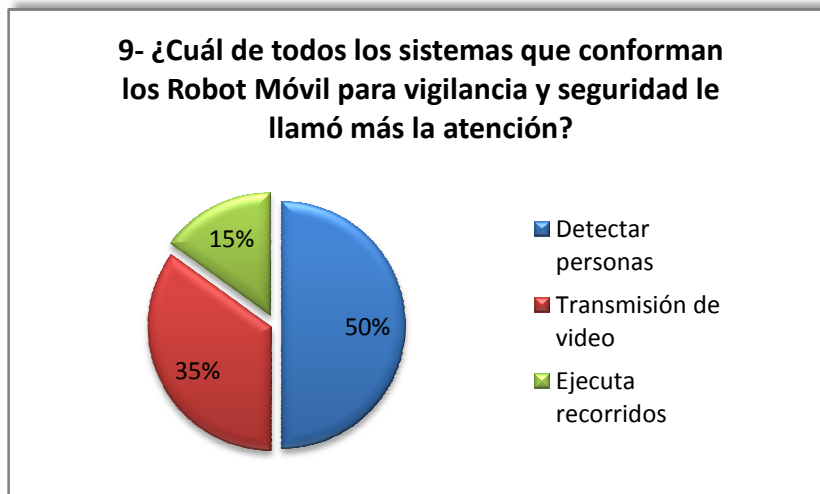
En esta gráfica se observa que el 80% de los encuestados desconocen o no han oído hablar sobre los Robots móvil para vigilancia y seguridad, mientras que el 20% si han escuchado hablar acerca de dichos robots, esto refleja que hay un grado bajo de información sobre el tema.

Figura 3.9. Gráfico N° 8



Aquí los encuestados reconocen en un 75% que sí hay aceptación para el campo de la Robótica de seguridad, sin embargo el 25% dice que no lo hay, esto puede suceder por el desconocimiento que tienen acerca de esta tendencia tecnológica.

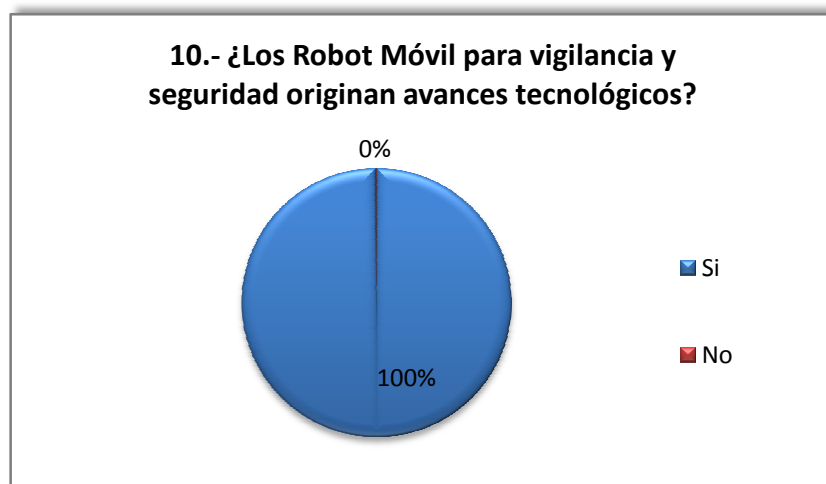
Figura 3.10. Gráfico N° 9



Los datos de la encuesta en esta pregunta reveló que el 50% de los encuestados le llama la atención los sistemas que detectan personas, el 35% opinó que les llama la atención los sistemas que transmiten video, el 15% de todos los encuestados señaló que les llama la atención los robots que realizan un recorrido.

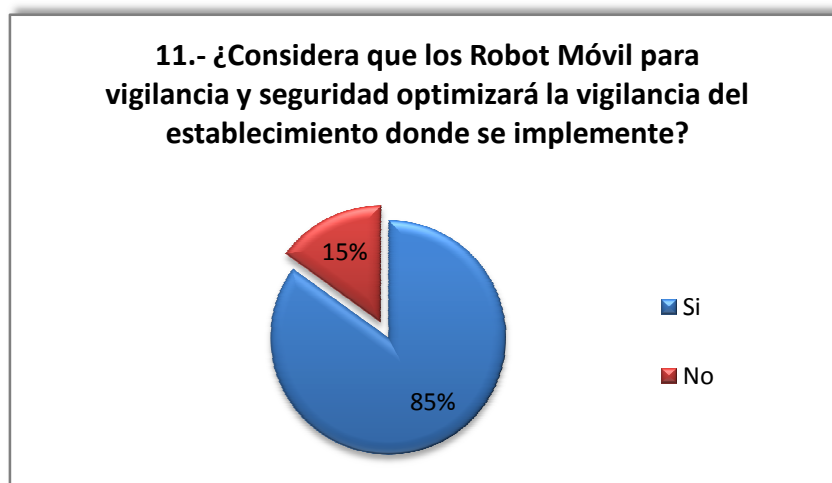
Esta información se toma en consideración, ya que esta tesis está enfocada en la implementación de un prototipo de robot móvil con los sistemas mencionados como parte complementaria para la tarea de vigilancia y seguridad, además de fomentar el interés en los demás sobre estos sistemas que conforman los robots de seguridad móvil.

Figura 3.11. Gráfico N° 10



En esta pregunta el 100% opinó casi sin dudar que la robótica de seguridad sí origina avances tecnológicos.

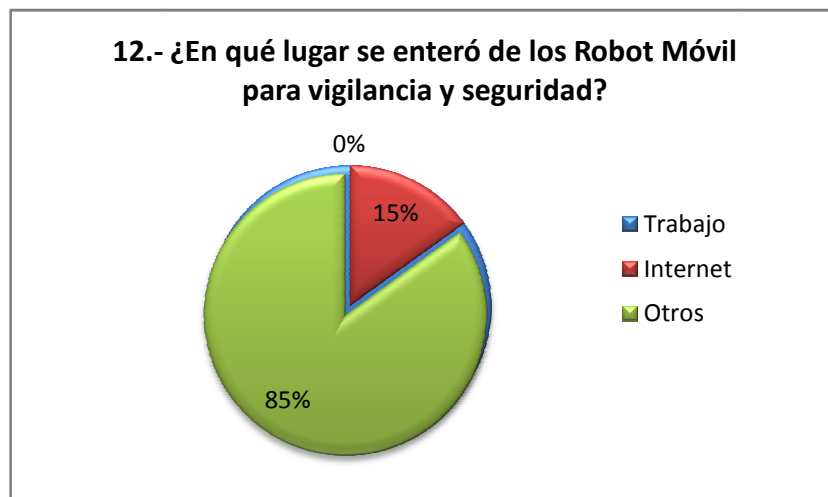
Figura 3.12. Gráfico N° 11



El 85% de los encuestados opinan en esta pregunta que la robótica de seguridad optimizará la vigilancia del establecimiento

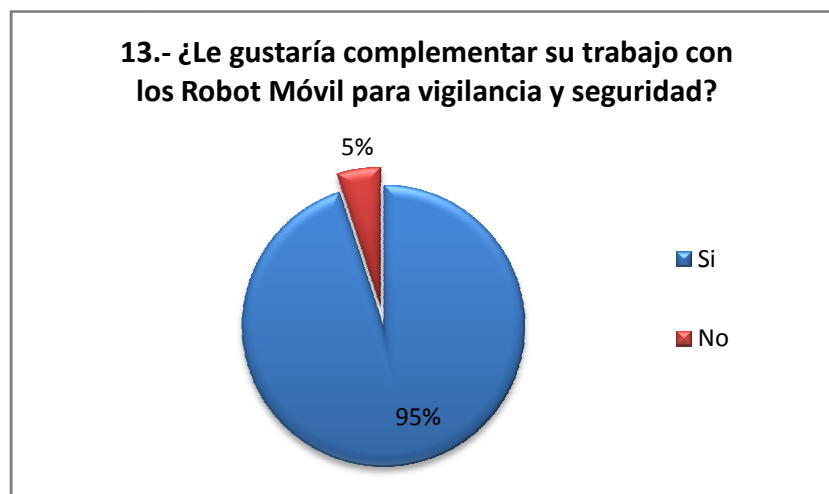
donde se implemente, esto abarca también los sistemas electrónicos de seguridad, esto demuestra que los encuestados están plenamente conscientes de que se debe invertir en nueva tecnología para obtener una protección total.

Figura 3.13. Gráfico N° 12



En esta gráfica observamos claramente que el 85% se enteró de los robots de vigilancia y seguridad, a través de otros medios donde se incluye: revistas, periódicos, por amistades y la presente encuesta, el 15% de los encuestados se enteró de los robots de seguridad y sus sistemas a través del internet.

Figura 3.14. Gráfico N° 13



Aquí se nota claramente el entusiasmo de disponer de una nueva tendencia tecnológica; así respondieron, un 95% de los encuestados, sobre el gusto de beneficiarse de un robot de seguridad, solo el 5% dijo no querer emplear esta tecnología.

La solución que se plantea en el proyecto tiene por finalidad, como ya se ha mencionado, contribuir a la gestión de vigilancia y seguridad, complementando la tarea del guardia ya que el robot móvil ayudaría a monitorear el área establecida; con los datos obtenidos se dio a conocer la acogida que tendrá este proyecto; además este prototipo se implementará en los laboratorios de la universidad y será accesible a los estudiantes por lo tanto permitirá promover la investigación para desarrollar sistemas de vigilancia con capacidad de desplazarse.

# **CAPÍTULO IV**

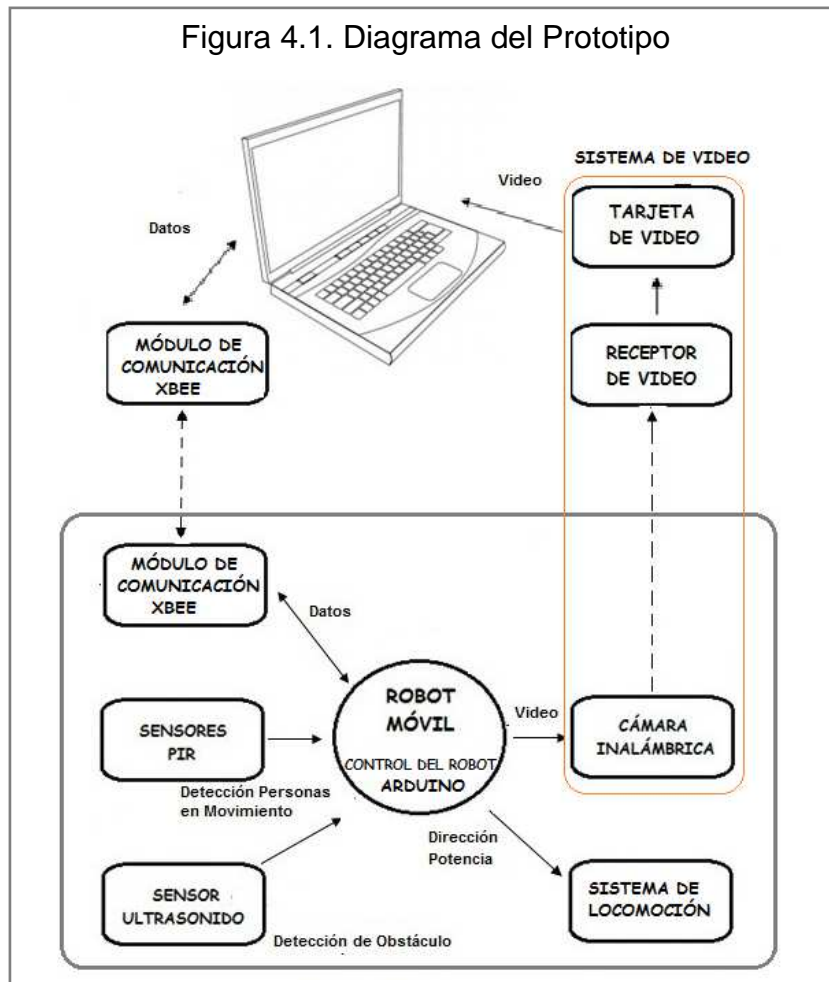
## **DISEÑO**

### **4. DISEÑO**

En el presente capítulo se explica todas las consideraciones tomadas para el desarrollo del diseño del prototipo de robot móvil.

Como muestra la figura 4.1, en forma global el prototipo puede ser manipulado remotamente por medio de la computadora; esto se realiza en modo tele-operado interactuando con el usuario que es la persona encargada de la tarea de vigilancia del establecimiento; el prototipo también puede trabajar de forma autónoma censando el entorno para

seguir la trayectoria programada y monitorear el área determinada, que es la razón principal del presente proyecto.



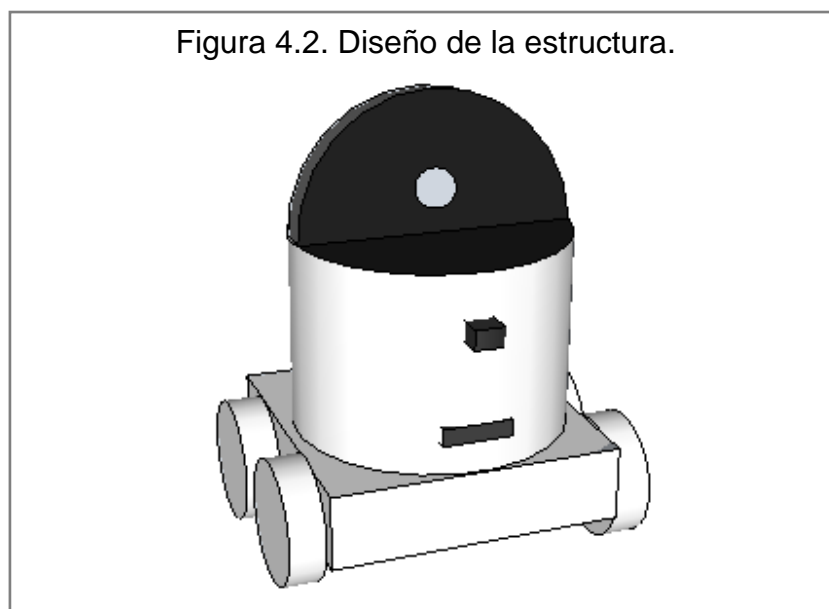
#### 4.1. Arquitectura de la solución

Se decidió por este diseño de estructura para el prototipo de robot móvil con visión nocturna para vigilancia y seguridad, porque facilita la ubicación de los elementos electrónicos; ya que el prototipo estará formado por diferentes componentes, que permitirán alcanzar los objetivos planteados, a continuación se detallará la estructura, el diseño seleccionado y los componentes del prototipo.



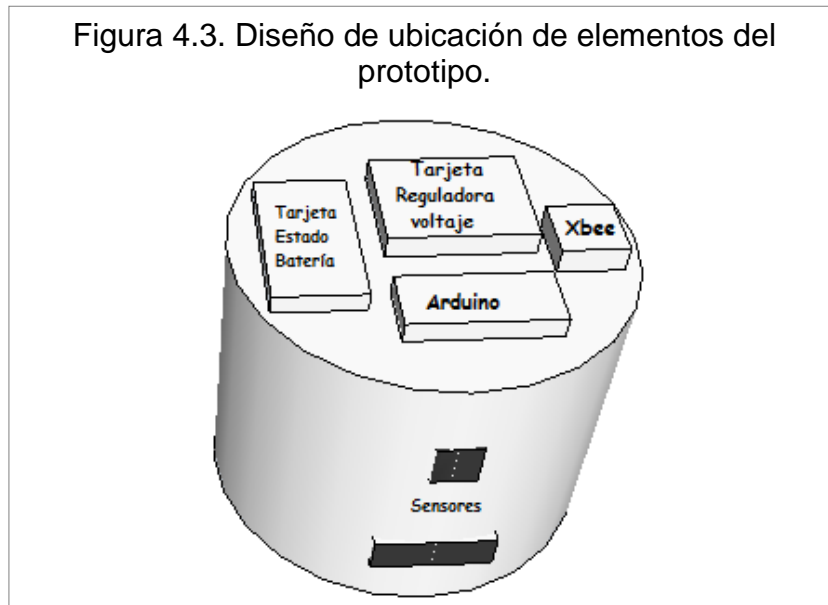
**Estructura del robot.-** Compone la forma física del robot y es la encargada de sostener sus elementos. Se puede fabricar de diferentes materiales metales, plásticos, entre otros; y de formas variadas. Tanto la forma y el material depende del propósito con el que se diseñe el robot.

Para el proyecto se seleccionó una estructura donde poder acoplar las cuatro ruedas con sus respectivos motores y tener espacio suficiente para colocar la batería por lo que se opto por una base cuadrada de latón con dimensiones de 25cm x 25cm x 10cm, en la parte de arriba se colocó un tacho de plástico metalizado de 20cm de diámetro y una altura de 20cm, esto es para tener un ángulo de vista con la cámara y los sensores. Se seleccionó este tipo de material para aligerar el peso, y poder cubrir los elementos.



En la figura 4.3 se muestra la estructura con la ubicación de los elementos del prototipo; a continuación se detalla los elementos electrónicos contenidos en esta parte de la estructura.

Figura 4.3. Diseño de ubicación de elementos del prototipo.



**Sistema de Video.-** La cámara se ubicó en la parte superior del prototipo, esto se realizó para facilitar la visibilidad del entorno al usuario. Se empleó una cámara inalámbrica con visión infrarroja para poder cumplir el monitoreo nocturno, cuenta con un receptor de 1,2 GHz que se conecta a la computadora por medio de una tarjeta capturadora de video; tiene una cobertura de 30m.

**Control del robot.-** Controla todo lo que llega por las entradas de sus sensores y provoca las reacciones necesarias, a través de sus actuadores que le imprimen movimiento al robot. En este proyecto se empleó la tarjeta Arduino Arduclerna, por la necesidad básica del proyecto, ya que cuenta con los puertos necesarios para los elementos empleados para el desarrollo del prototipo, La facilidad del código de alto nivel del Arduino, facilita el desarrollo y control del prototipo.

**Módulo de comunicación.-** Para la comunicación entre la computadora y el prototipo; se seleccionó un módulo ZigBee que trabaje sin ningún inconveniente. Una vez vistas las diferentes

alternativas con sus respectivas aplicaciones, se decidió seleccionar el módulo Xbee Serie 1; tipo de antena chip, con una potencia de transmisión de 1mW, tiene un alcance de 30m para interiores y 100m en espacios abiertos; este módulo es más óptimo para realizar la comunicación de este prototipo. Los módulos Xbee s1 tienen una cobertura de 30m para interiores que se ajusta a la cobertura de la cámara que también puede trabajar a 30m, con estos rangos de operación se logrará cumplir con la propuesta de monitorear el área establecida.

Para realizar la comunicación se utilizó la Transmisión serial transparente (modo AT), permite la comunicación punto a punto. Configuramos los módulos Xbee desde el PC utilizando el programa X-CTU. El funcionamiento del módulo Xbee Serie 1, es como si de un cable se tratase, mediante una transmisión serial, para iniciar el intercambio de datos simplemente debemos seleccionar el puerto que ocupa el módulo, configurar la velocidad y podremos comenzar la comunicación. Importante recordar conectar los RX y TX cruzados con el Arduino.

El módulo Xbee para el Arduino es usado con el adaptador Xbee Explorer Serial, ya que facilita la conexión con la tarjeta de Arduino, mientras que el módulo Xbee para la computadora emplea la tarjeta Xbee Explorer USB. Esta tarjeta aparte de conectar la computadora con el Xbee, también permite configurar fácilmente los módulos Xbee.

Los Xbee deberán tener el mismo firmware para poder comunicarse, además de la velocidad, canal y red. Configuramos los módulos con el programa XCT-U que lo podemos descargar en la página oficial de los módulos Xbee. Primero actualizamos la versión de firmware de los módulos desde el XCT-U.

Configuramos los siguientes parámetros de los módulos Xbee.

- Firmware: 10ED
- Velocidad: 9600 baudios.
- Canal : 0 (El mismo canal)
- PAN ID: 888 (La misma red)
- Para el emisor: DH: 0, DL: 1, MY: 0
- Para el receptor: DH: 0, DL: 0, MY:1

**Sensores.-** Los sensores suministran la información necesaria para que el robot interprete el entorno en el cual se encuentra. Se seleccionó para el proyecto los sensores de ultrasonido SFR04, que permiten detectar obstáculos casi con precisión con un ángulo de detección eficiente de 15°, y los sensores PIR Dyp Me003, que permiten detectar personas en movimiento en un área de cobertura de 7m; con estos sensores se podrá resolver el monitoreo del área establecida.

- **Sensor de ultrasonido HC-SR04**

El HC-SR04 es un sensor de proximidad que usa un sonar para la detección de obstáculos. Por uno de sus sonares envía la onda sonora en un cono de 15° hacia delante, y por la otra recibe de nuevo la onda sonora, que vuelve tras haber chocado con el obstáculo. Sus pines son Trig y Echo. Tiene un alcance que va desde los 2 cm hasta los 400cm. No se ve afectado por la luz su funcionamiento, pero puede tener algún problema con algunos materiales. Se alimenta a 5V y consume 15mA en este proyecto detecta obstáculos a 20 centímetros.

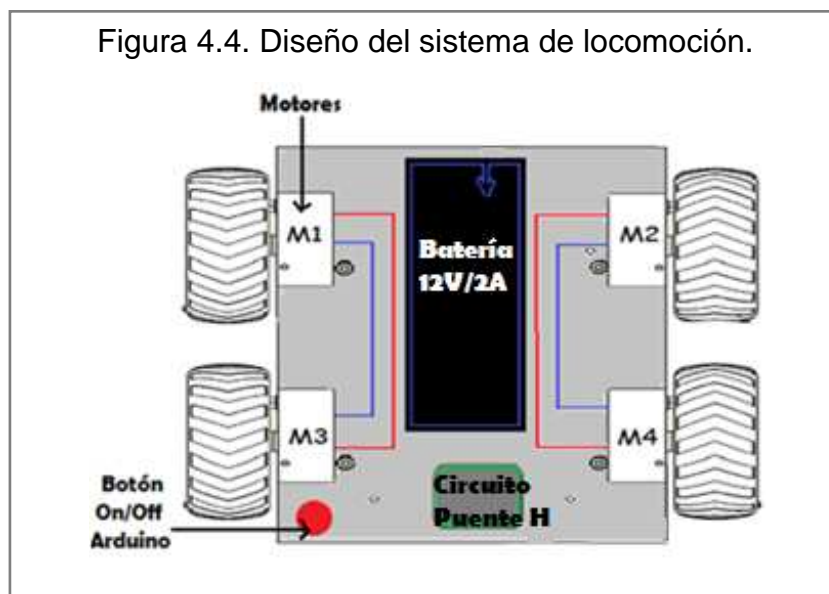
- **Sensor PIR DYP-ME003**

El sensor PIR posee 3 terminales, el pin positivo al cual se le suministró 5V, el pin negativo (GND), y el pin out se conecta al

pin in/out del Arduino (detección de movimiento). En el proyecto se empleó dos sensores PIR y se los configuró para que detecten personas en movimiento en un rango de 3 metros.

**Sistema de locomoción.-** El robot tiene un sistema que le permite moverse, según la información que se ha recibido, permitiendo que el mismo llegue a cumplir la misión para el cuál fue diseñado. Se prefirió el sistema de locomoción diferencial por su fácil implementación y control.

Las principales características a la hora de elegir un motor son el par (torque), velocidad, consumo, peso, tamaño y precio. La elección de los motores es importante porque de ellos depende la electrónica necesaria, la autonomía y las baterías a usar.



El prototipo ha utilizado para su sistema de locomoción motores DC, con caja reductora para que proporcione el torque necesario para movilizar el robot sin mucho esfuerzo por parte de los motores, se colocó 4 motores con sus respectivas ruedas; ya que

todo el peso del robot está repartido sobre las 4 ruedas; dichos motores tienen un torque de 6Kg/cm, debido al tamaño y al peso del prototipo que es aproximadamente de 2.5 Kg., el tipo de rueda que se empleo es un Kit Sports Tamiya de 4 llantas de 56mm de diámetro x 25mm de ancho, que permite mayor facilidad en la movilidad y giros que debe ejecutar para cumplir con su desplazamiento. En la figura 4.4, se muestra como están distribuidos dichos motores con sus respectivas llantas.

Para controlar el sistema de locomoción con precisión, hemos utilizado cuatro motores conectados en paralelos; por lo que los motores M1 y M3 se controlan como si fuera un solo motor con las entradas del puente (1 y 2) y los motores M2 y M4 con las entradas (3 y 4). Este sistema de locomoción básicamente es el de tracción diferencial, como su funcionamiento indica para hacer avanzar el robot alimentamos los motores M1-M3 y M2-M4, para retroceder invertimos el giro en los motores M1-M3 y M2-M4, y para los respectivos giros hacemos funcionar los motores M1-M3 avanzar hacia adelante y M2-M4 avanzar al sentido contrario. En la tabla 4.1 se puede observar el funcionamiento del sistema de locomoción.

En el Anexo 2 se podrá ver el modelado y simulación del movimiento del robot con tracción diferencial en Matlab.

Tabla 4.1. Funcionamiento del sistema de locomoción.

Movimiento	Motor M1-M3	Motor M2-M4
Avanzar	1	1
Detener	0	0
Giro Izquierda	1	0
Giro Derecha	0	1

**Sistema de alimentación.-** La fuente de poder que provee de energía al robot debe ser adecuada a la potencia requerida para el correcto funcionamiento de sus elementos. Para la elección de la batería se analizó la operación de todo el sistema. Se considera como ciclo normal de operación del robot a la secuencia de actividades formada por las siguientes tareas: recorrer ruta, sensar entorno y visualizar cámara.

Cada tarea tiene que cumplirse en un intervalo de tiempo, y dependiendo de la operación a realizar actúan ciertos actuadores con su respectivo consumo de corriente. Podemos guiarnos en esta tabla de los valores de energía requerido por el prototipo.

Tabla 4.2. Consumo de energía de los elementos en una ronda.

Elementos	Cant.	Consumo	Recorrer ruta	Sensar entorno	Visualizar entorno
Motores	4	500 mA	2000 mA	-	-
Puente H	1	50 mA	50 mA	-	-
Cámara	1	200 mA	-	-	200 mA
Xbee	1	50 mA	50 mA	50 mA	50 mA
Arduino	1	50 mA	50 mA	50 mA	50 mA
Sensor PIR	2	20 mA	-	20 mA	-
Ultrasonido	1	15 mA	15 mA	-	-
<b>Corriente Total</b>			<b>2165 mA</b>	<b>120 mA</b>	<b>300 mA</b>

Se opto por emplear una batería de 12V /2A y una de 9V /200mA, ya que el requerimiento del sistema así lo amerita. El circuito de control del prototipo de robot funciona con voltaje de alimentación de 9V, que lo proporciona la batería de 9V. Los demás componentes son alimentados por la batería de 12V, mediante el regulador de voltaje 7805 se consigue obtener 5V utilizado para los sensores, y con el 7808 los 8V para la cámara.

### 5.1.1 Diagramas de actividades del prototipo

Figura 4.5. Diagrama de actividades del Modo Tele-Operado.

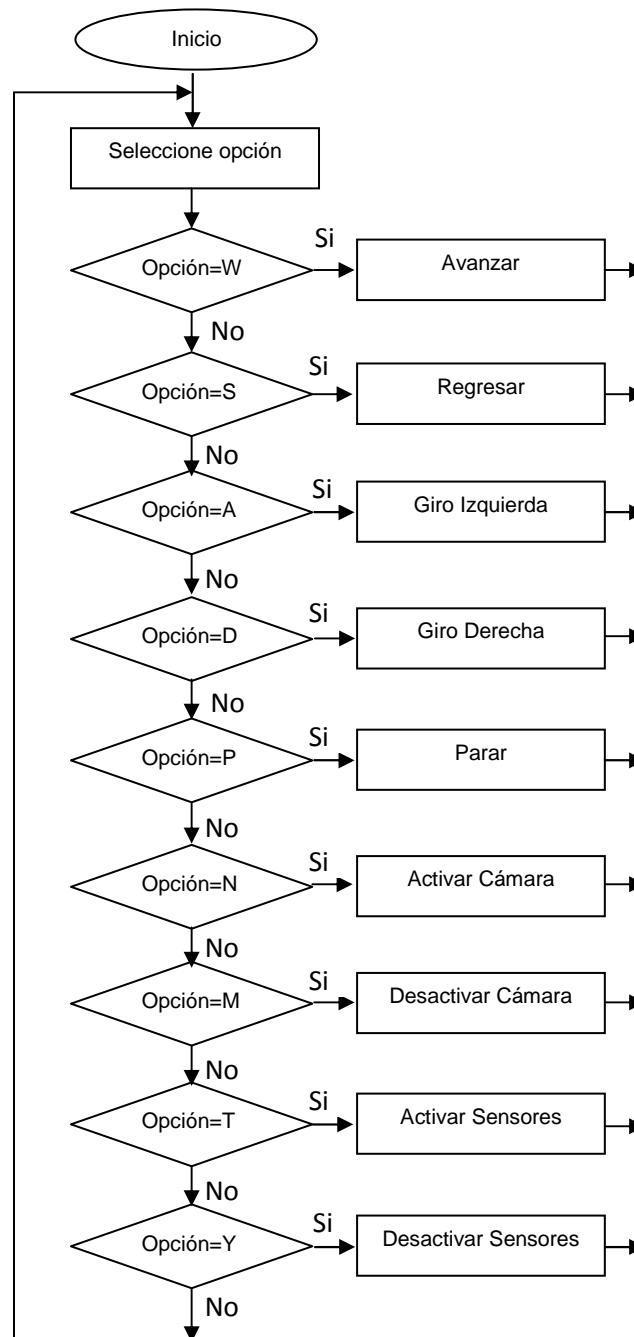
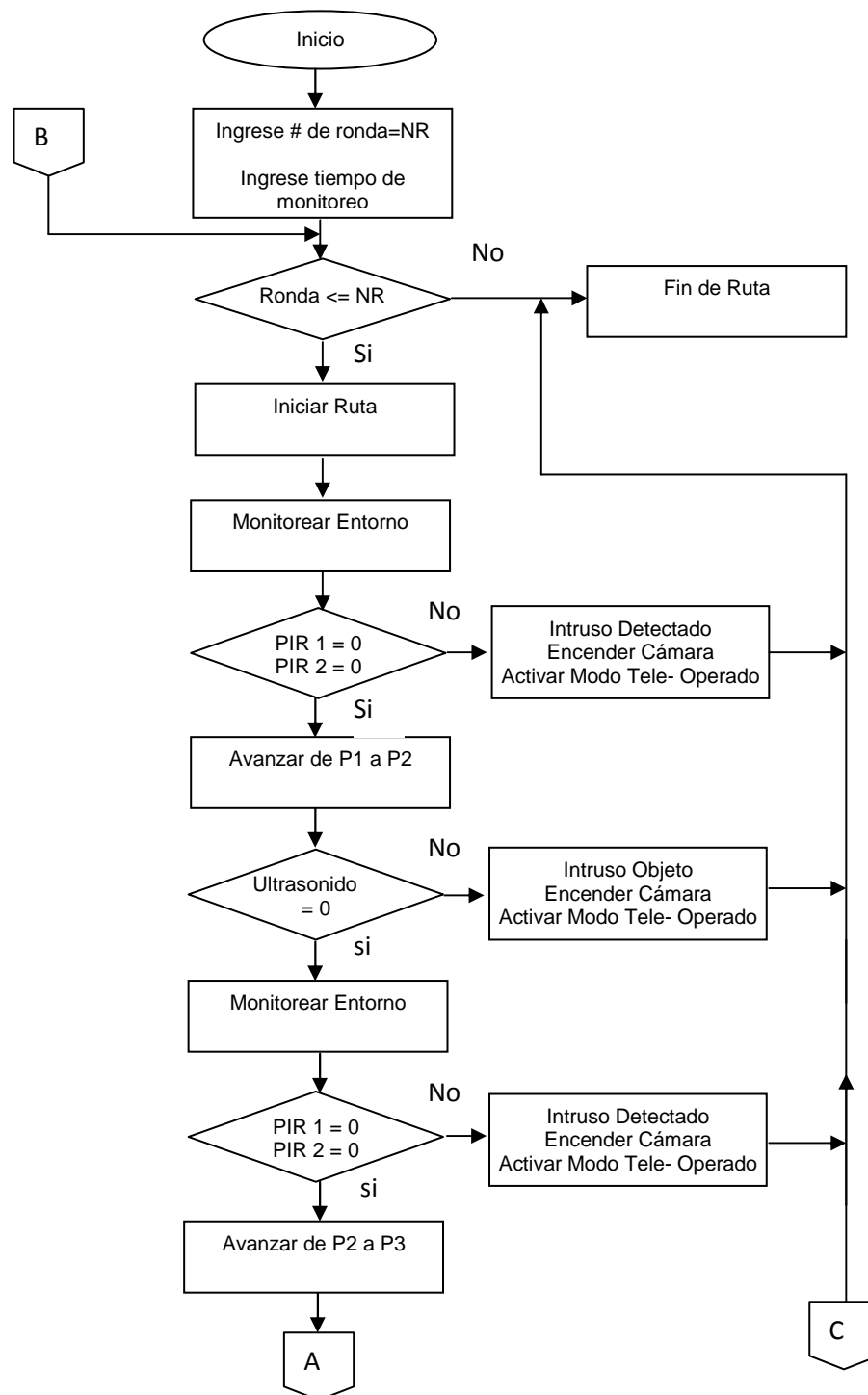
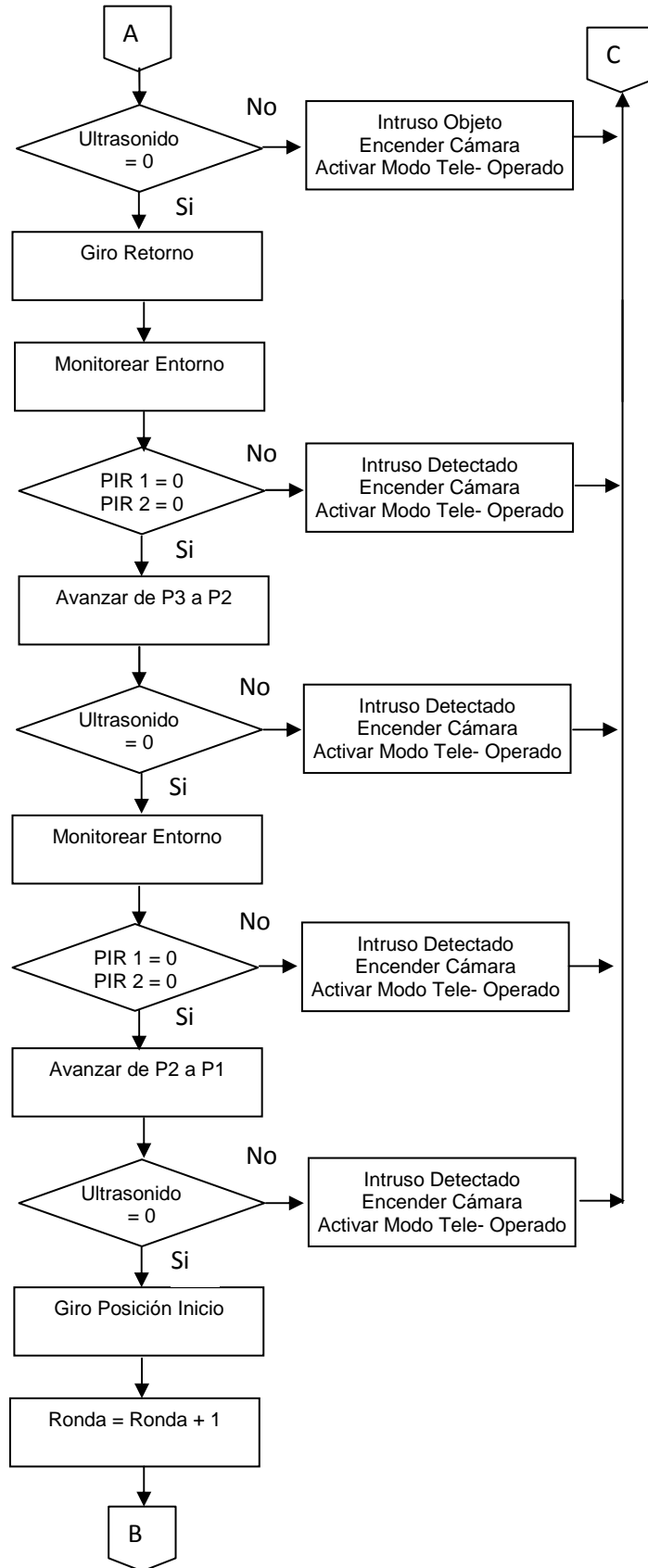




Figura 4.6. Diagrama de actividades del Modo Automático.





## 4.2. Diseño de Interfaz

El diseño de la interfaz de control y monitorización, se la realizo en la herramienta de Visual Basic 2010 Express; para el fácil uso del usuario.

Se inicia el programa seleccionando el puerto a utilizar por la comunicación inalámbrica del módulo Xbee y el tipo de cámara.

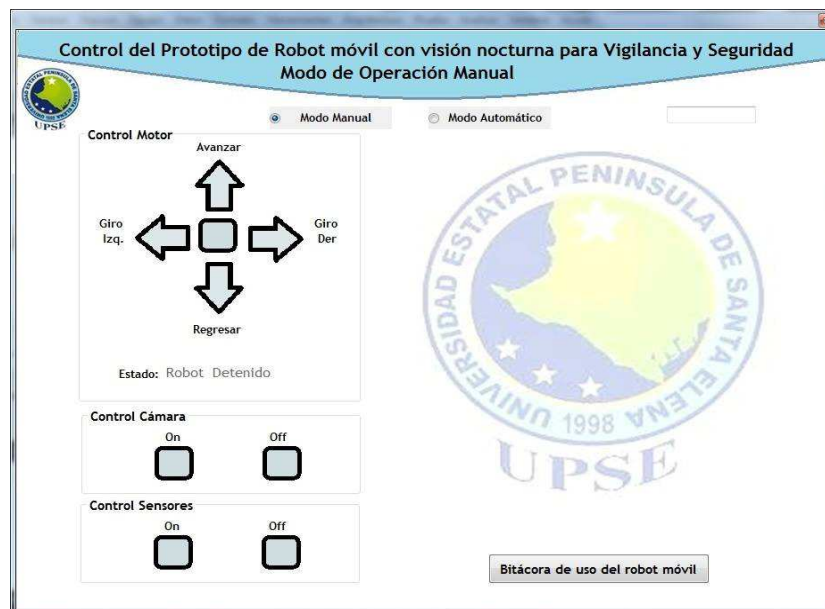
Figura 4.7. Pantalla inicial del programa de monitorización de vigilancia y seguridad.



La pantalla principal del programa de Monitorización de Vigilancia y Seguridad inicia en modo Tele-Operado para poder manejar al robot o cambiar al Modo Automático.

En el Modo Tele-Operado se encuentran todos los botones de los comandos para manipular el prototipo, encender la cámara, y activar los sensores; las órdenes son enviadas a través de los Xbee y procesadas por el Arduino.

Figura 4.8. Pantalla principal del programa de monitorización de vigilancia y seguridad, Modo Manual.



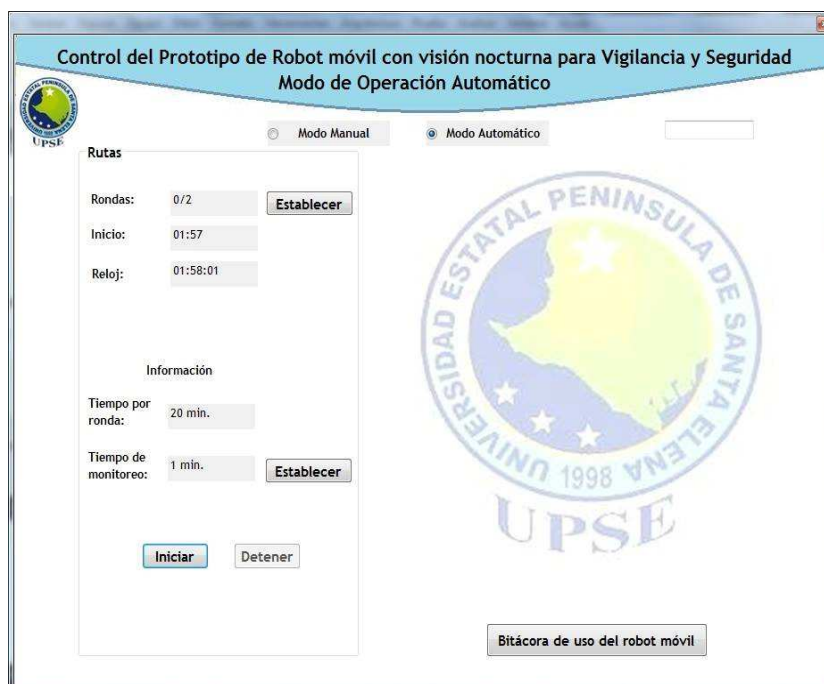
Al seleccionar el Modo Automático se debe configurar el número de rondas que debe dar el robot, como también el tiempo de monitoreo; para que comience a ejecutar el recorrido se debe seleccionar Iniciar, se envía la orden para ejecutar la ruta programada en el Arduino, censando su entorno y monitoreando por el tiempo definido por el usuario en las respectivas paradas, en esta modo se activa la cámara cuando un objeto o un intruso es detectado; se puede detener el modo automático seleccionando Detener.

También se podrá generar reportes del uso del robot móvil con visión nocturna para vigilancia y seguridad, y las notificaciones de detección de intruso. La base de datos para el reporte se realizará en SQL Server Express 2008, está compuesta con una tabla con los siguientes campos:

- Id\_registro
- Fecha
- Hora inicio
- Hora final
- Modo de operación
- Observación

En el reporte se presenta la fecha, la hora de inicio y final de uso, el modo de operación usado, si hubo o no detección, y en observación se detalla si hubo detección de intruso con la hora de la detección.

Figura 4.9. Pantalla principal del programa de monitorización de vigilancia y seguridad, Modo Automático.



Se utilizó la versión Express de los softwares Visual Basic y SQL Serve, ya que su licenciamiento es gratuito y también pueden ser distribuidos sin costos adicionales.

# **CAPÍTULO V**

## **IMPLEMENTACIÓN**

### **5. IMPLEMENTACIÓN**

La estructura del prototipo está compuesta por los siguientes componentes:

- Sistema de control.
- Sistema sensorial.
- Sistema de comunicación.
- Sistema actuador.
- Sistema de potencia/alimentación.
- Circuito electrónico

En el presente capítulo se explica todas las consideraciones tomadas en cuenta para la implementación del robot móvil con visión nocturna para vigilancia y seguridad. En el Anexo 3 se podrá observar como se ha conectado todos los circuitos y elementos del prototipo.

### 5.1. Construcción

Para la construcción de la estructura del prototipo del robot móvil para vigilancia y seguridad, se empleó una base cuadrada de 25 cm x 25 cm y 10 cm de alto, el material es de latón, fácil de moldear y ligero. Las llantas instaladas es un Kit Sports Tamiya de 4 llantas de 56mm de diámetro x 25mm de ancho, que ofrecen la tracción necesaria para ejecutar el recorrido sin deslizamiento. Se ubicaron con precisión para obtener un buen rendimiento en el sistema de locomoción.



Se instalaron los cuatro motores con sus respectivas ruedas y se colocó la batería en la posición central, asegurándola en su lugar

para no tener inconvenientes. Para que la cámara pueda tener un ángulo de captura adecuado se adaptó al prototipo base, una parte del tacho plástico metalizado de 20 cm de diámetro.

Figura 5.2. Construcción prototipo motores



Se tomó las medidas necesarias al colocar los sensores en el tacho y se midió el nivel del corte para adaptarlo a la base.

Figura 5.3. Construcción prototipo base-estructura



La cámara se colocó en la tapa del tacho, se perforó la tapa e instaló la cámara. Se ubicó el sensor de ultrasonido al frente a



una altura de 20 cm, para que pueda detectar objetos de una altura mínima de 25 cm de alto; se instalarán los sensores PIR, al frente y en la parte trasera para tener mejor cobertura en el monitoreo del entorno; se colocó el botón On/Off en un área de fácil acceso al usuario.

Figura 5.4. Construcción prototipo sensores.



La estructura del prototipo terminada, cuenta con las dimensiones proporcionadas de acuerdo con los elementos que se instalarán en su interior.

#### **5.1.1. Sistema de Control**

Se uso la tarjeta Arduino Arduclema, la cual se encarga de comandar todas las acciones del robot, tales como procesar la información proveniente de los sensores, enviar órdenes a los motores, en respuesta a la ejecución del algoritmo de control, o por órdenes enviadas desde la PC.

Utilizamos los siguientes pines del Arduino para el funcionamiento del robot:

Tabla 5.1. Pines de Arduino para el prototipo

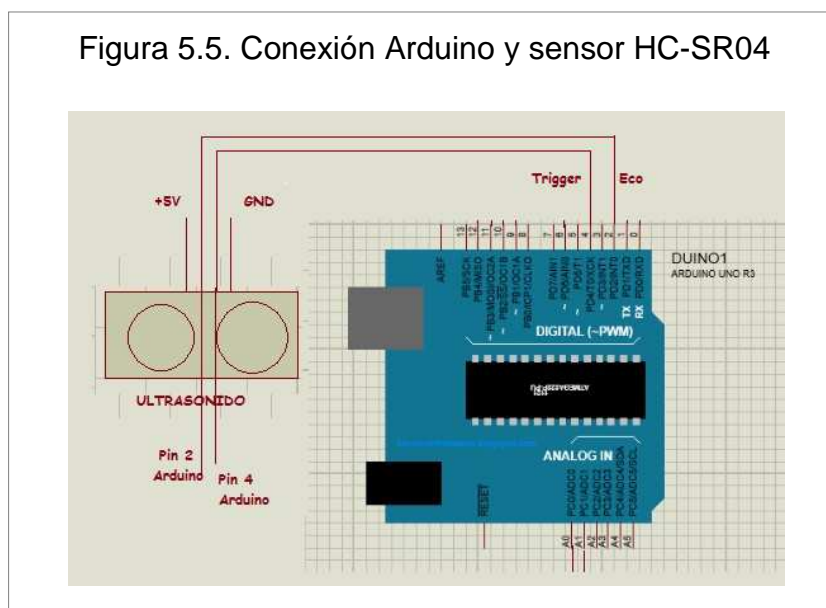
Pines de Arduino	Función
Pin 0	Tx Xbee
Pin 1	Rx Xbee
Pin 2	Eco (sensor ultrasonido)
Pin 3	Trigger (sensor ultrasonido)
Pin 6	Motor 1A (Motor 2-4)
Pin 9	Motor 2A (Motor 2-4)
Pin 10	Motor 1B (Motor 1-3)
Pin 11	Motor 2B (Motor 1-3)
Pin 13	Activar/Desactivar cámara
Pin 14	Sensor PIR delantero
Pin 15	Sensor PIR trasero
Pin GND	Tierra Común

En el Anexo 4 se podrá encontrar el algoritmo de programación del Arduino; y la programación en Visual Basic del interfaz de usuario en el Anexo 5.

### 5.1.2 Sistema sensorial

Está compuesto por sensores de ultrasonido HC-SR04 y PIR DYP-ME003.

Figura 5.5. Conexión Arduino y sensor HC-SR04



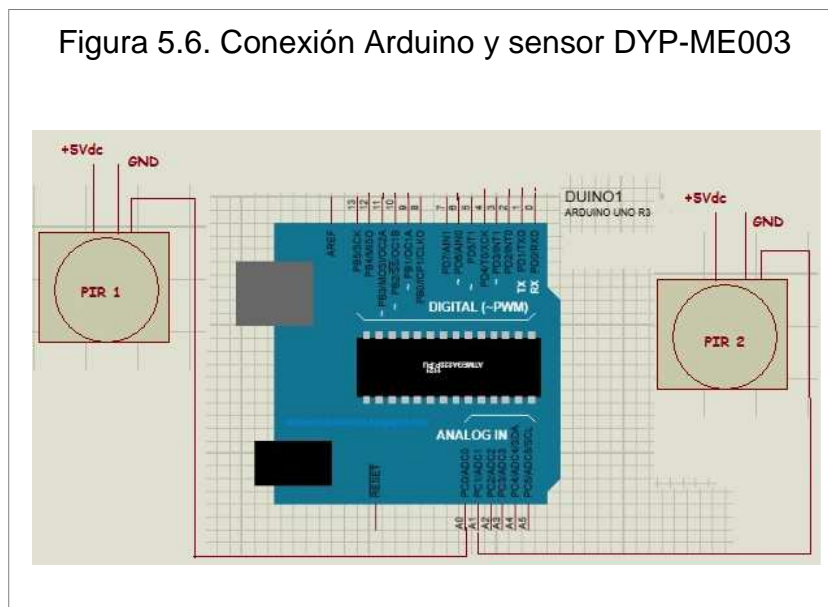
El sensor HC-SR04 tiene un alcance que va desde los 2 cm hasta los 400cm. Su funcionamiento no se ve afectado por la luz. El

sensor de ultrasonido tiene cuatro terminales, dos para la alimentación y dos para envío y recepción del pulso ultrasónico (trigger y eco); en la figura 5.5 se muestra como conectamos el sensor ultrasónico con la tarjeta Arduino, recibe la alimentación de la tarjeta reguladora de voltaje. Detecta dentro de un rango de 15°.

- **Sensor PIR DYP-ME003**

El sensor PIR posee 3 terminales, en la figura 5.6 se muestra como se ha conectado los sensores PIR con la tarjeta Arduino. Los PIR se alimentan desde la tarjeta reguladora de voltaje. Su funcionamiento es simple cuando detecta movimiento manda un voltaje de 5V al Arduino.

Figura 5.6. Conexión Arduino y sensor DYP-ME003

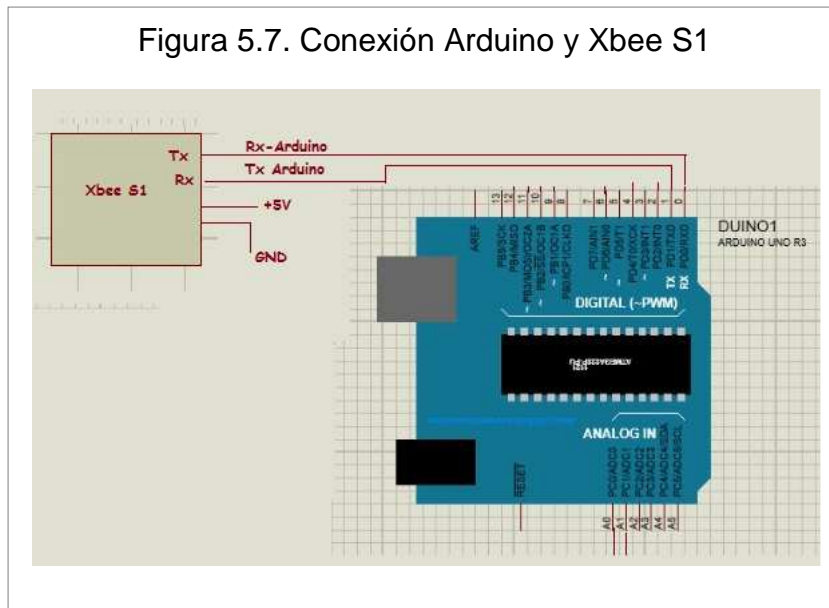


### 5.1.3. Sistema de comunicación

Para la comunicación se utilizó el módulo Xbee Serie 1; tipo de antena chip, con una potencia de transmisión de 1mW; tiene un alcance de 30m para interiores y 100m en espacios abiertos, que es suficiente para abarcar el área monitoreada. Se conectó el RX-

Arduino con el Tx-Xbee y TX-Arduino con el Rx-Xbee. Se alimenta el Xbee desde la tarjeta reguladora con 5V, ya que el Xbee utiliza el adaptador TTL que se encarga de suministrar los 3,3V que necesita el Xbee.

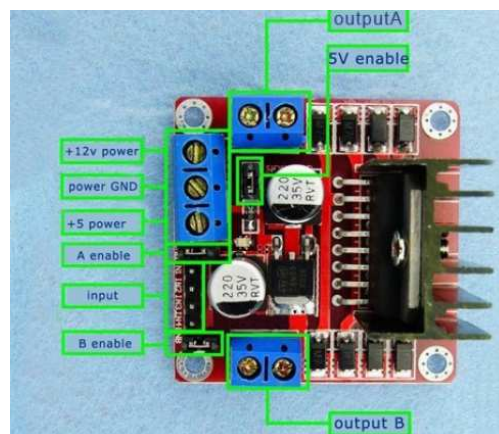
Figura 5.7. Conexión Arduino y Xbee S1



#### 5.1.4. Sistema actuador

Las ruedas del robot son accionadas, cada una, por un motor DC; los motores empleados son de 12V / 500mA con una caja reductora para aumentar su torque.

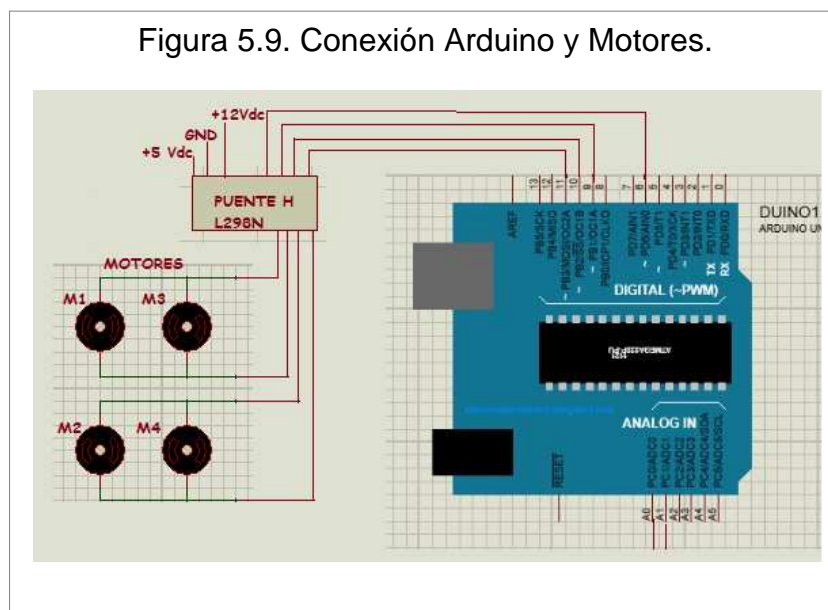
Figura 5.8. Conexión Puente H.



Fuente: <http://www.abcelectronica.net/productos/circuitos-integrados/drivers-motores/>

Para el control de los motores se empleó un integrado, puente H L298N que permite controlar el sentido de giro de los motores. El integrado L298N hace funcionar los 2 motores DC. Se conecta los motores en output A y output B, y las entradas se controlan directamente desde los pines de la tarjeta Arduino.

En la figura 5.9 se observa cómo se conectó los motores a los pines de entrada del puente H, y los pines de salida del puente H a los respectivos pines del Arduino. El integrado del puente H funciona a 5V, y para alimentar los motores conectamos a 12V, el puente también es alimentado desde la tarjeta reguladora.

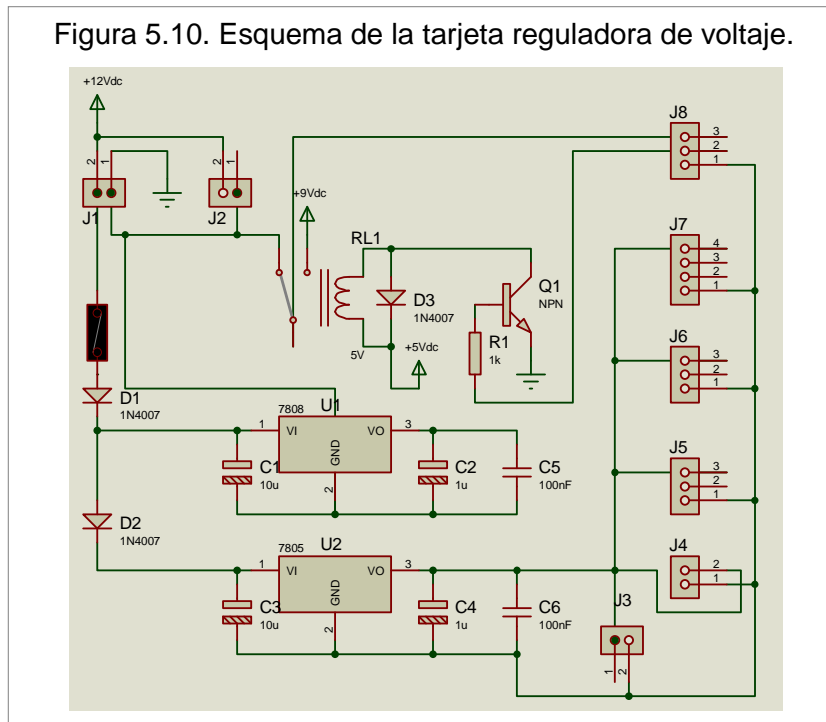


### 5.1.5. Sistema de potencia y alimentación

Para que el robot opere con autonomía de aproximadamente una hora se conectó una fuente de 12V y 2A; y una batería de 9V y 200mA para alimentar el Arduino. Se implementó una tarjeta reguladora de voltaje; utiliza los integrados de regulación de voltaje 7808, para alimentar la cámara y 7805, para alimentar los sensores y el xbee. Esta tarjeta permite regular una salida de

voltaje DC constante respectivamente de 8 y 5 Voltios. Se colocó en las entradas y salidas del regulador; los capacitores electrolítico (10uF y 1uF) que permiten estabilizar el voltaje de entrada para convertirla en un voltaje más constante, y los capacitores cerámicos (100nF) sirven para eliminar el ruido generado por cualquier elemento. Incluye también un relé para activar y desactivar la cámara. En esta tarjeta se ha colocado conectores para los sensores, xbee, cámara y motores. Se debe unir la tierra de este circuito con la de la tarjeta Arduino para su funcionamiento.

Figura 5.10. Esquema de la tarjeta reguladora de voltaje.



Se detalla los conectores de la tarjeta reguladora para los elementos del prototipo:

Conector J1: Se conectó el voltaje de entrada de 12V.

Conector J2: Se conectó la alimentación del puente H.

Conector J3: Se conectó la tarjeta TTL para alimentar el Xbee.

Conector J4: Se conectó la tierra del Arduino.

Conector J5: Se conectó el sensor PIR delantero.

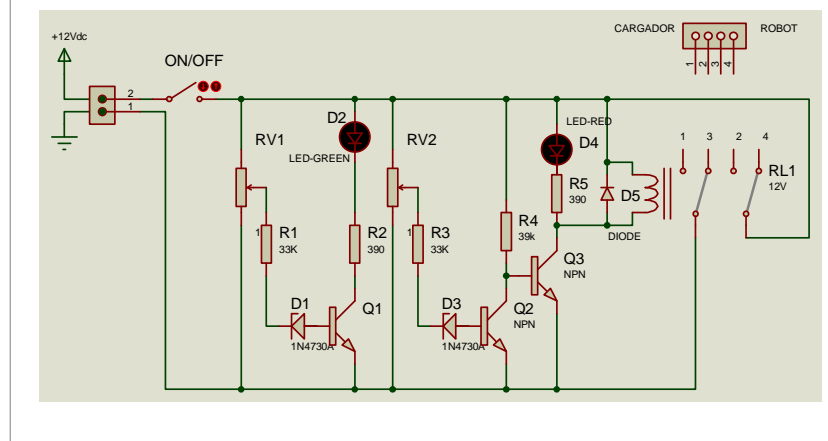
Conector J6: Se conectó el sensor PIR trasero.

Conector J7: Se conectó el sensor de ultrasonido.

Conector J8: Se conectó la cámara.

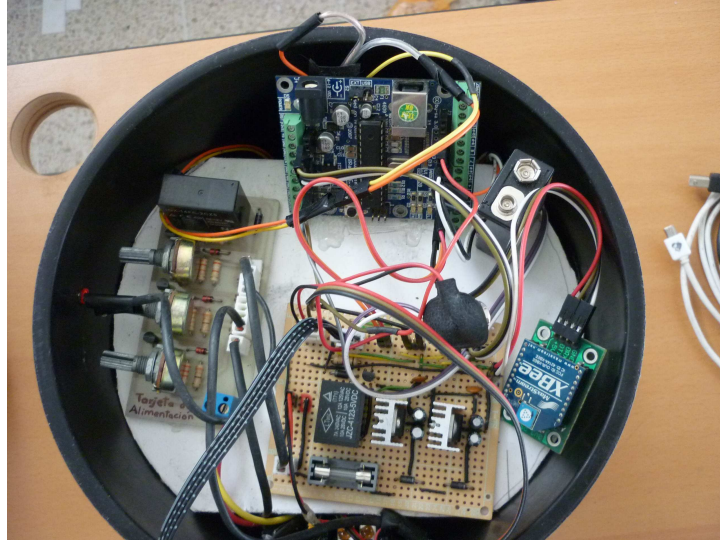
También se implementó una tarjeta para controlar el estado de la batería; medirá los voltajes de la batería e indicará en qué estado se encuentra, dando a conocer por medio de un led verde que la batería se encuentra cargada, y led rojo batería descargada. Por medio de un relé alimentado a 12V se podrá proveer energía al circuito del Robot cuando la batería este en un valor de hasta 11V, cuando este a un valor de voltaje inferior el circuito se desconectará y pasará a modo cargar.

Figura 5.11. Esquema de la tarjeta control de estado de la batería.



Como se puede observar en la figura 5.11 en el circuito se ha instalado conectores para el botón ON/OFF, los led's verde y rojo, y la alimentación del circuito del robot y el cargador; esto es para facilitar su uso ya que se puede desconectar y conectar los componentes mencionados. Se conecta la batería de 12V, directamente a la tarjeta reguladora. En el Anexo 6 se podrá ver los esquemas electrónicos de los circuitos empleados.

Figura 5.12. Conexión interna del prototipo.



Con el prototipo terminado se procederá a realizar pruebas para verificar su correcto funcionamiento.

Figura 5.13. Prototipo terminado.





## 5.2. Pruebas

Las pruebas se realizaron a los diferentes módulos que conforman el robot móvil, como también al prototipo y los resultados obtenidos son para comprobar su correcto funcionamiento y determinar las conclusiones sobre el proyecto.

### 5.2.1 Pruebas y resultados del sensor ultrasónico

Las pruebas fueron realizadas para comprobar el correcto funcionamiento del sensor ultrasónico y determinar si detecta obstáculos. Las pruebas se ejecutaron en un área despejada al frente del sensor. Se realizaron 10 intentos de cada una de las pruebas presentadas en la tabla 5.2. Se colocaron cajas con diferentes dimensiones ubicadas a distancias de entre 10 y 20 cm, que es la distancia de trabajo para el proyecto. El sensor ultrasónico tiene un ángulo de emisión de 15 grados, la probabilidad más alta de detección se localiza en el centro del cono de emisión. El sensor está al frente del prototipo, a una altura de 20cm por lo que el objeto debe tener una altura mínima de 25cm.

Figura 5.14. Configuración del sensor de ultrasonido.

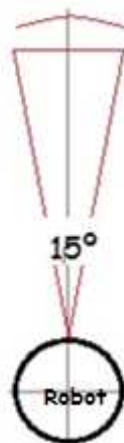


Tabla 5.2. Resultado de pruebas del sensor ultrasónico.

Prueba	Dimensiones de Objeto	Distancia objeto respecto al prototipo	Sensor Ultrasonido
1	10 cm* 10cm	10 cm	Fallido
2	10 cm* 10cm	20 cm	Fallido
3	15 cm* 15cm	10 cm	Fallido
4	15 cm* 15cm	20 cm	Fallido
5	20 cm* 20cm	10 cm	Fallido
6	20 cm* 20cm	20 cm	Fallido
7	25 cm* 25cm	10 cm	Fallido
8	25 cm* 25cm	20 cm	Correcto
9	30 cm* 30cm	10 cm	Fallido
10	30 cm* 30cm	20 cm	Correcto

El análisis de los resultados de las pruebas revela que los obstáculos deben tener un ancho mínimo de 30 cm, de preferencia objetos rectangulares, el ancho del objeto es debido a que el prototipo presenta pequeños desvíos por lo que a mayor ancho de objeto mayor probabilidad de ser detectado. Así mismo la distancia mínima a la que el robot puede estar del obstáculo se fija a 20cm para garantizar que no exista una colisión; el robot será incapaz de detectar obstáculos que se encuentren fuera del ángulo de cobertura, o de dimensiones más pequeña que la especificada, o cuando el obstáculo aparece súbitamente al frente del robot.

Figura 5.15. Prueba sensor ultrasonido.



### 5.2.2. Pruebas y resultados de los sensores PIR

Las pruebas fueron realizadas para comprobar el funcionamiento de los dos sensores PIR y determinar si son capaces de detectar personas. Las pruebas se llevaron a cabo en el laboratorio electrónico de la escuela de Electrónica y Telecomunicaciones de la UPSE. Se realizaron 10 intentos de cada una de las pruebas presentadas en la tabla 5.3. Los sensores PIR tienen un ángulo de cobertura de  $140^\circ$  y para el proyecto estos sensores están configurados a un rango que abarca 3m al frente de cada sensor; esto es para obtener una captura con la cámara sin mucha distorsión. Cuando se prende hay que esperar 5 segundos para que el sensor haga una captura del lugar, no debe haber personas en movimiento dentro del rango del sensor en el momento de calibración. Cuando detecta a la persona en movimiento envía una señal de 5V al Arduino para que encienda la cámara y pase a modo tele-operado. Los sensores están; uno al frente, y otro en la parte trasera del prototipo a una altura de 40cm por lo que detecta el movimiento del intruso.

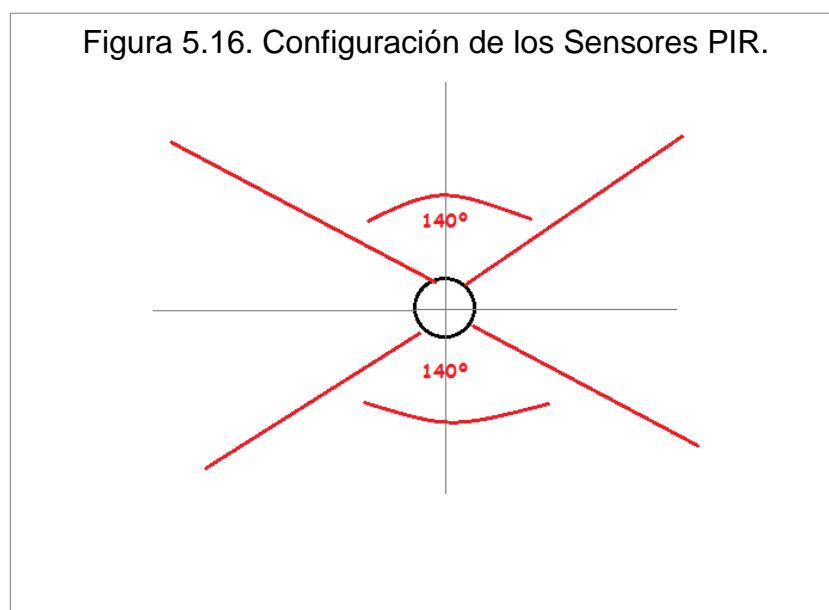
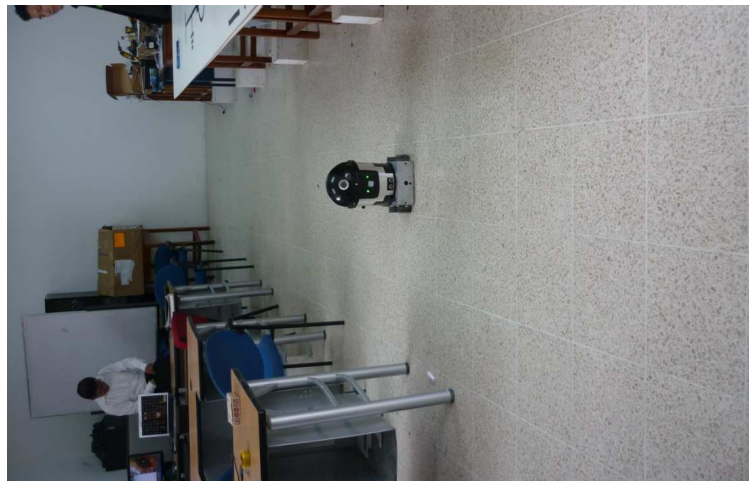


Tabla 5.3. Resultado de pruebas de los sensores PIR.

Prueba	Estado de Intruso	Sensores PIR
1	Estático	Fallido
2	Caminando a paso lento	Correcto
3	Caminando a paso normal	Correcto
4	Caminando a paso acelerado	Fallido
5	Corriendo	Fallido

Las pruebas nos dieron a conocer que el robot es capaz de detectar un intruso en un área de cobertura de tres metros; determinando que las personas tienen que estar en movimiento para poder ser detectadas; dicho movimiento puede ser caminando a paso normal o paso lento. Estas pruebas dejan en claro que la detección de personas es posible con este sensor.

Figura 5.17. Prueba sensores PIR.

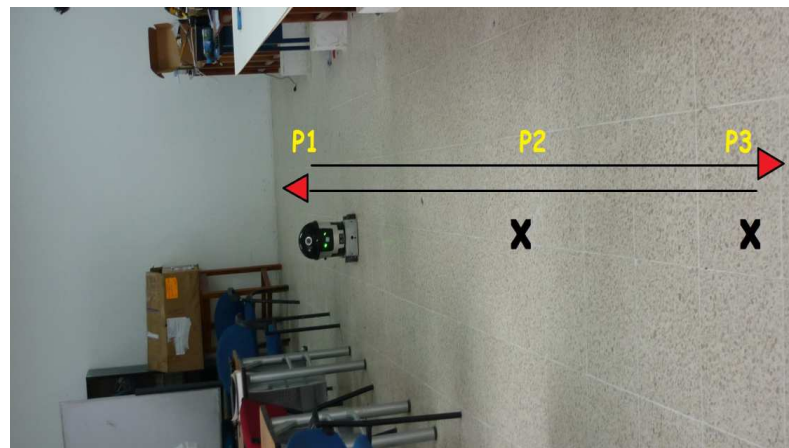


### 5.2.3. Pruebas y resultados del recorrido autónomo libre de obstáculo.

Esta prueba se realizó para conocer el comportamiento del robot móvil en un entorno libre de obstáculos con el algoritmo de

recorrido autónomo. La prueba se llevó a cabo en el laboratorio de Electrónica de la Universidad, sin presencia de obstáculos, con el robot móvil ubicado en un punto P1 con un recorrido-destino ubicada en un punto P3, luego retorno al punto P1. Desde el punto P1 a P2=200cm, y P2 a P3=200cm; recorrido total ida y vuelta= 800cm. Se considero 10 intentos de la prueba.

Figura 5.18. Prueba recorrido autónomo.



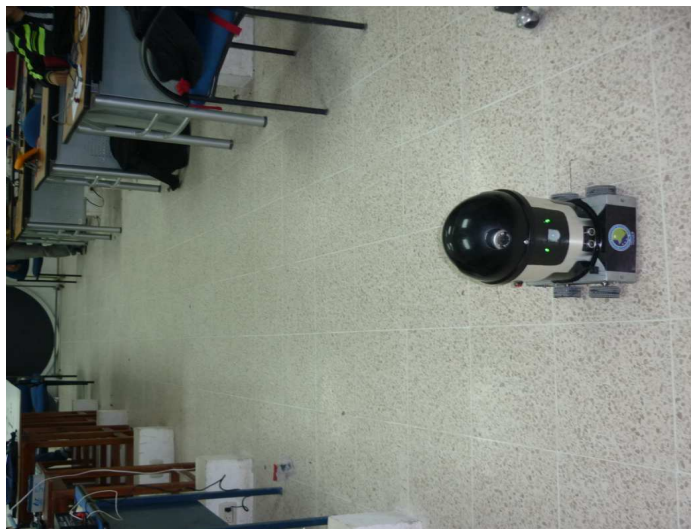
Ejecuta Recorrido Autónomo.

- 1.- Monitorear el área por 1 minuto en el punto P1.
- 2.- Avanzar del punto P1 al punto P2.
- 3.-Monitorear el área por 1 minuto en el punto P2.
- 4.- Avanzar del punto P2 al punto P3.
- 5.-Dar la vuelta el prototipo para retornar el trayecto recorrido.
- 6.-Monitorear el área por 1 minuto en el punto P3.
- 7.-Avanzar del punto P3 al punto P2.
- 8.-Monitorear el área por 1 minuto en el punto P2.
- 9.- Avanzar del punto P2 al punto P1.
- 10.-Dar la vuelta el prototipo para quedar en posición inicial.

Tabla 5.4. Resultado de pruebas del recorrido autónomo libre de obstáculo.

Prueba	Recorrido autónomo libre de obstáculo
1	Correcto
2	Correcto
3	Correcto
4	Correcto
5	Correcto
6	Correcto
7	Correcto
8	Correcto
9	Fallido
10	Fallido
<b>Resultado</b>	80%

Figura 5.19. Prueba recorrido autónomo libre de obstáculo.



Las pruebas del recorrido autónomo libre de obstáculo nos manifestaron que el robot está 80% capacitado para ejecutar el recorrido de la ruta programada; mostrando pequeñas variaciones en el tiempo que le tomó para terminar las rondas, debido al desgaste de la batería que influye en la velocidad del robot. También presentó algunas desviaciones que fueron corregidas mediante la programación.

#### 5.2.4. Pruebas y resultados del recorrido autónomo con intruso

Esta prueba se realizó con el objetivo de conocer el comportamiento del robot móvil en un entorno con un intruso entre el punto de origen y destino. La prueba se llevó a cabo en el laboratorio de Electrónica de la Universidad, con un intruso en el área a monitorear, el robot móvil ubicado en un punto P1 ejecutará el recorrido programado. Se considero 10 ensayos de la prueba, tomando en cuenta los resultados de las pruebas de los sensores PIR realizadas anteriormente, el sensor detecta cuando el intruso está en movimiento.

Tabla 5.5. Resultado de pruebas del recorrido autónomo con intruso.

Prueba	Recorrido autónomo con un intruso
1	Correcto
2	Correcto
3	Correcto
4	Correcto
5	Fallido
6	Correcto
7	Correcto
8	Correcto
9	Correcto
10	Fallido
Resultado	80%

Las pruebas del recorrido autónomo con un intruso nos manifestaron que el robot está 80% capacitado para cumplir con el recorrido de la ruta programada en el área del laboratorio de electrónica de la universidad; cumpliendo con el objetivo de monitorear el entorno y detectar intrusos. En cada uno de los ensayos el robot alcanzo su objetivo, declarando esta prueba como exitosa cuando el robot comienza avanzar, si detecta un intruso se deberá detener y encender la cámara.

Figura 5.20. Prueba recorrido autónomo con un obstáculo.



Después de terminar la etapa de pruebas se concluyó que el robot alcanzó a cumplir las especificaciones tomadas al inicio del proyecto y superó los entornos de pruebas diseñados. El prototipo es capaz de cumplir el recorrido en el laboratorio de Electrónica de la Universidad.

### **5.3. Documentación**

Se presenta la demostración de la hipótesis y el manual de usuario para la operación y mantenimiento.

#### **5.3.1. Comprobación de Hipótesis**

Una vez terminadas las pruebas al prototipo de robot móvil y con la tabulación de cada una de las preguntas de las encuestas a los encargados de la seguridad en la universidad, se comprobó la hipótesis que se propuso en el proyecto.

La mayor parte de los encuestados del personal de seguridad están totalmente consientes sobre el beneficio de usar un robot



móvil de seguridad; por tanto utilizarían este prototipo de robot móvil para vigilancia; que ayudará al cuerpo de guardia a mejorar el control de vigilancia y monitoreo en el área del laboratorio de la Escuela de Electrónica y Telecomunicaciones de la Universidad Estatal Península de Santa Elena; con las pruebas realizadas se determino que el prototipo realiza su objetivo sirviendo de apoyo a los encargados de inspeccionar el área establecida.

### **5.3.2 Manual de usuario**

La elaboración de este manual ayudará al usuario a manipular el prototipo; además de contribuir a su correcto funcionamiento y mantenimiento. Para visualizar el manual de usuario dirigirse al Anexo 7.

## Conclusiones

- El prototipo de robot móvil es una herramienta que ayudará al cuerpo de guardia a mejorar el control de vigilancia y monitoreo en el área del laboratorio de la Escuela de Electrónica y Telecomunicaciones de la Universidad Estatal Península de Santa Elena, el prototipo no pretende reemplazar al guardia sino servir de refuerzo en el proceso de seguridad, con esta herramienta los guardias tendrán un apoyo para la vigilancia y con su interfaz amigable incentivará su uso.
- Considerando el análisis de la etapa electrónica, se construyó el sistema electrónico de forma sencilla seccionándolo en módulos, haciendo que cada módulo funcione de manera independiente y tomando en cuenta que cualquier componente utilizado puede ser reemplazado convenientemente en caso de cualquier daño; además estos módulos permitieron ubicar fácilmente los circuitos dentro de la estructura del robot.
- El sistema de locomoción con tracción diferencial, genera algunas desviaciones en el trayecto, pero son controladas por software y corregidas en un 95%, cumpliendo la trayectoria programada.
- El sensor ultrasónico tiene un ángulo de emisión de 15 grados; por lo tanto objetos que se encuentren fuera del ángulo de cobertura, o de dimensiones más pequeñas que la especificada 25\*30\*30 cm no son detectados.

## Recomendaciones

- Ya que la tecnología robótica avanza cada vez más rápido, se sugiere actualizarse constantemente, usando las herramientas que estén a nuestra disposición, además de crear nuevas formas de mejorar la seguridad.
- Verificar que los elementos estén conectados y ubicados de la manera correcta, antes de poner en funcionamiento el robot.
- Evitar terrenos accidentados ya que la tracción no se diseñó para tales entornos; para un mejor desempeño en terrenos más generales, recomendamos modificar el sistema de tracción.
- Se puede mejorar la autonomía del robot móvil agregándole un sistema de navegación inercial con sensores capaz de estimar la posición, orientación y velocidad.
- Se aconseja adaptar un mejor sistema de alimentación basado en baterías de polímero de litio que reducirán el peso del robot y aumentará su tiempo de trabajo.

## Bibliografía

1. J. Craig, John, (2006). *Robótica*. (3ra ed.), México: Pearson Educacion
2. Domínguez, Silvia, Sánchez, Enrique y Sánchez, Gabriel, (2009). *Guía para elaborar una tesis*. (1ra ed.), México: McGraw Hill.
3. Adamatzky, Andrew; Komosinski, Maciej (2009). *Artificial Life Models in Hardware*. Disponible en: <http://www.ebib.com>
4. Enríquez, Rafael, (2009). *Guía de usuario de Arduino*, Disponible en internet: *Página Oficial Arduino*, (<http://www.arduino.cc/es/>)
5. Warren, John-David; Adams, Josh; Molle, Harald (2011). *Arduino Robotics*. Disponible en: <http://www.ebib.com>
6. De León, Joel, Ordóñez, Elbin y Castellanos Omar, (2010). *Tutorial Visual Basic 2010*, Guatemala.
7. Gutiérrez Suarez, Bryan, (2012). *Diseño e implementación de un algoritmo de generación de trayectorias para la evasión de un obstáculo para un robot móvil*, (Tesis para optar el Título de Ingeniero Electrónico), Pontificia Universidad Católica del Perú, Perú, Lima.
8. Zabala Barragán, Leticia, (2011). *Diseño e Implementación de prototipo robot publicitario para la escuela de Ingeniería Electrónica*, (Proyecto de grado previo a la obtención del Título de Ingeniería en Electrónica y Computación), Escuela Superior Politécnica de Chimborazo, Ecuador, Quito.

**Anexos**

## ANEXO 1

### "Formato de encuesta para personal de seguridad de la UPSE"



**Universidad Estatal Península de Santa Elena**  
**Facultad de Sistemas y Telecomunicaciones**  
**Carrera de Electrónica y Telecomunicaciones**



**Objetivo:** Obtener la información necesaria para medir la factibilidad de desarrollar e implantar un prototipo de robot móvil con visión nocturna para vigilancia y seguridad mediante el uso de tecnología en robótica para control de vigilancia y monitoreo en el área del laboratorio de la Escuela de Electrónica y Telecomunicaciones de la Universidad Estatal Península de Santa Elena.

1.- Normalmente, ¿Cuántos guardias trabajan en la jornada nocturna?

- 1 – 2 guardias
- 3 – 4 guardias
- 5 – 6 guardias
- 7 guardias en adelante

2.- ¿Cuántas rondas de vigilancia se realiza en la noche por área?

- 1 – 2 rondas
- 3 – 4 rondas
- 5 – 6 rondas
- 7 rondas en adelante

3.- ¿Para realizar las rondas de vigilancia nocturnas cuantos guardias se necesitan por ronda?

- 1 – 2 guardias
- 3 – 4 guardias

- 5 – 6 guardias
- 7 guardias en adelante

4.- ¿Cómo se realiza el control de seguridad en la Universidad Estatal Península de Santa Elena?

- Verificar bienes y vigilancia externa.
- Otros procedimientos.

5.- ¿Cómo es la comunicación remota entre los guardias cuando están realizando las rondas?

- Por radio frecuencia
- Otro medio

6.- ¿En cuánto ayuda la visión de las cámaras instaladas en lugares estratégicos de la Universidad?

- Total
- Mucha
- Poco
- Nada

7.- ¿Ha oído hablar de los Robot Móvil para vigilancia y seguridad?

- SI
- NO

9.- A su punto de vista, ¿Cree que hay acogida para el campo de los Robot Móvil para vigilancia y seguridad?

- SI
- NO

8.- ¿Cuál de todos los sistemas que conforman los Robot Móvil para vigilancia y seguridad le llamó más la atención?

- Detectar personas
- Transmisión de video
- Ejecutar recorridos

10.- ¿Los Robot Móvil para vigilancia y seguridad originan avances tecnológicos?

- SI
- NO

11.- ¿Considera que los Robot Móvil para vigilancia y seguridad optimizará la vigilancia del establecimiento donde se implemente?

- SI
- NO

12.- ¿En qué lugar se enteró de los Robot Móvil para vigilancia y seguridad?

- Trabajo
- Internet
- Otro; indique\_\_\_\_\_

13.- ¿Le gustaría complementar su trabajo con los Robot Móvil para vigilancia y seguridad?

- SI
- NO



## ANEXO 2.

### Simulación en Matlab del movimiento del robot con tracción diferencial.

Realizamos la simulación del modelo cinemático del robot móvil en el software Matlab, para observar su comportamiento respecto a su posición y orientación, ajustando las velocidades a las ruedas izquierda y derecha. Empleamos las ecuaciones del apartado de modelo cinemático en el marco teórico. Para generar una trayectoria deseada se aplican diferentes combinaciones a las velocidades  $w_d$  y  $w_i$ , reemplazando en las ecuaciones los valores correspondientes. Estos valores  $w_d$  y  $w_i$  se toma de la configuración del pwm de la programación. Para todas las simulaciones se considerarán los parámetros siguientes: una separación entre las ruedas de  $D=150\text{mm}$ . El tiempo de simulación que se considera es  $t=10\text{ s}$ .

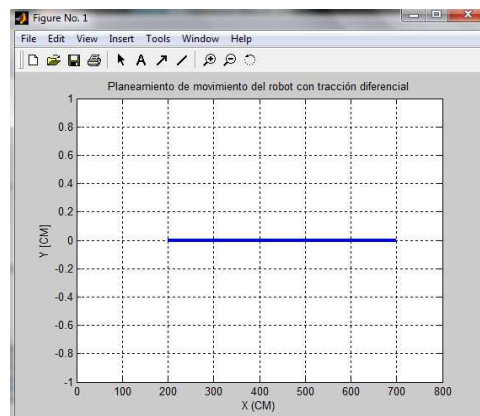
$$x = \frac{w_i - w_d}{2} \cos(\theta) \quad (1.1)$$

$$y = \frac{w_i - w_d}{2} \sin(\theta) \quad (1.2)$$

$$\theta = \frac{w_d - w_i}{D} \quad (1.3)$$

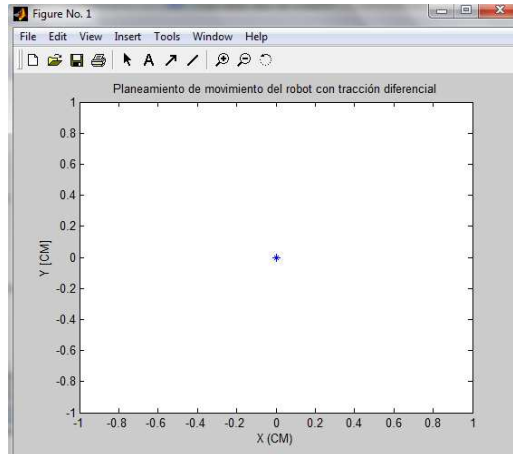
- a) *Simulación 1: Velocidades iguales en las ruedas,  $w_d = 200$  y  $w_i = 200$ .* Se genera una recta orientada en el plano XY, obteniendo la simulación mostrada.

```
v1=[0:1:100];
disp(v1);
wi=200+v1;
wd=200+v1;
D=150;
k=(wd-wi)/D;
x=((wi+wd)/2).*cos(k)
y=((wi+wd)/2).*sin(k)
plot(x,y,'linewidth',3)
grid on
axis([0 800 -1 1])
pause(0.01)
title('Planeamiento de movimiento del robot con tracción diferencial')
xlabel('X (CM)')
ylabel('Y [CM]')
```



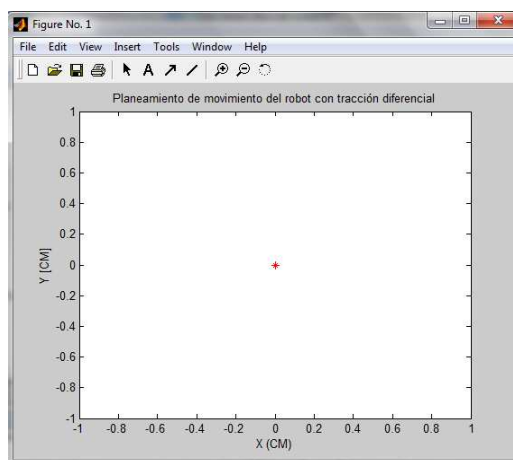
- b) *Simulación 2*: En esta simulación las velocidades de las ruedas son:  $w_d = 200$  y  $w_i = -200$ , por lo que el robot móvil gira continuamente en sentido anti horario. Trayectoria realizada por el robot móvil, para una velocidad de la rueda derecha mayor a la de la izquierda.

```
v1=[0:1:100];
disp(v1);
wi=-200-v1;
wd=200+v1;
D=150;
k=(wd-wi)/D;
x=((wi+wd)/2).*cos(k)
y=((wi+wd)/2).*sin(k)
plot(x,y,'*')
%grid on
%axis([0 800 -1 1])
pause (0.01)
title('Planeamiento de
movimiento del robot con
tracción diferencial')
xlabel('X (CM)')
ylabel('Y [CM]')
```



- c) *Simulación 3*: En esta simulación las velocidades de las ruedas son:  $w_d = -200$  y  $w_i = 200$ , por lo que el robot móvil gira continuamente en sentido horario. Trayectoria realizada por el robot móvil, para una velocidad de la rueda derecha menor a la de la izquierda.

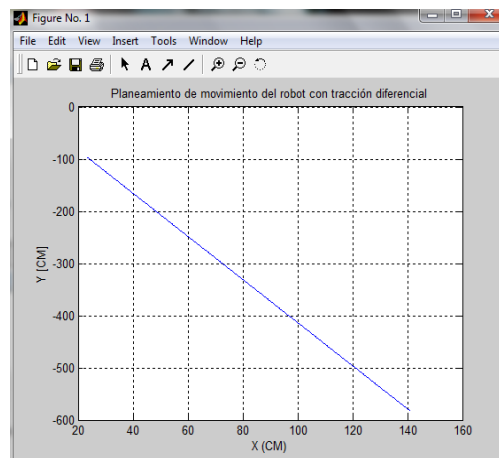
```
v1=[0:1:100];
disp(v1);
wi=200+v1;
wd=-200-v1;
D=150;
k=(wd-wi)/D;
x=((wi+wd)/2).*cos(k)
y=((wi+wd)/2).*sin(k)
plot(x,y,'r*')
%grid on
%axis([0 800 -1 1])
pause (0.01)
title('Planeamiento de
movimiento del robot con
tracción diferencial')
xlabel('X (CM)')
ylabel('Y [CM]')
```



- d) *Simulación 3:* En esta simulación las velocidades de las ruedas son: (A)  $\omega_d = 100$  y  $\omega_i = 200$ , (B)  $\omega_d = 200$  y  $\omega_i = 100$ ; por lo que el robot móvil realiza una trayectoria con un desvío, como se observa en las siguientes figuras.

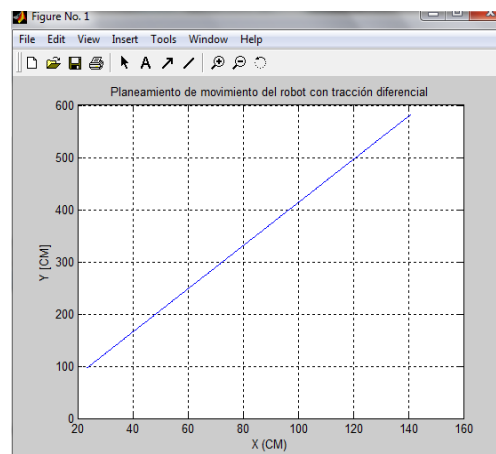
**A.** Trayectoria realizada por el robot móvil, para una velocidad de la rueda derecha menor a la de la izquierda.

```
v1=[0:1:500];
disp(v1);
wi=200+v1;
wd=100+v1;
D=150;
k=(wd-wi)/D;
x=((wi+wd)/2).*cos(k)
y=((wi+wd)/2).*sin(k)
plot(x,y)
grid on
pause (0.01)
```



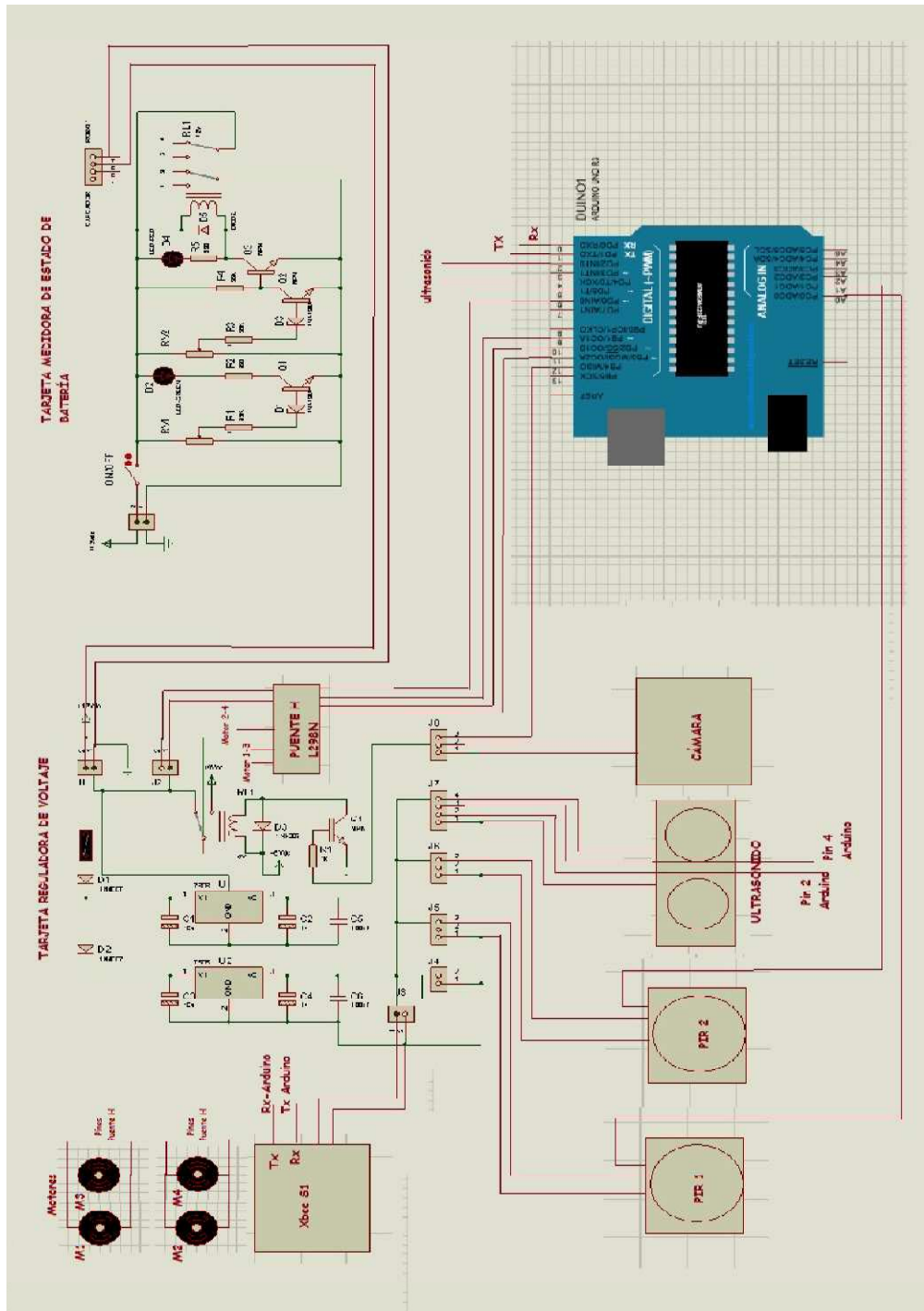
**B.** Trayectoria realizada por el robot móvil, para una velocidad de la rueda derecha mayor a la de la izquierda.

```
v1=[0:1:500];
disp(v1);
wi=100+v1;
wd=200+v1;
D=150;
k=(wd-wi)/D;
x=((wi+wd)/2).*cos(k)
y=((wi+wd)/2).*sin(k)
plot(x,y)
grid on
pause (0.01)
```



### ANEXO 3.

## “DIAGRAMA DEL CIRCUITO ELECTRÓNICO GENERAL”



## ANEXO 4

### "CÓDIGO FUENTE ALGORITMO DE PROGRAMACIÓN EN ARDUINO"

```
#include<Ultrasonic.h>
int motor1A = 6; //-----
-----MOTORES LADO IZQUIERDO
int motor2A = 9;
int motor1B = 10; //-----
-----MOTORES LADO DERECHO
int motor2B = 11;
intcamara=13; //-----
-----CÁMARA
int pir1=14; //ADELANTE---//-----
-----PIR
int pir2=15; //ATRAS
Ultrasonic ultrasonic(4,2); // (Trig PIN,Echo
PIN) //-----ULTRASONIDO
int ECHO=2;
int TRIGGER=4;

String trama="";
boolean tramacompleta=false;
booleanmodomanual=true;
intcontador_rondas=0;

intestado_adelante=0; //-----
-----ESTADOS
int estado_adelante_2=0;
int estado_adelante_3=0;

intestado_detenido=0;
int estado_detenido_2=0;
int estado_detenido_3=0;
int estado_detenido_4=0;

intestado_general=0;
int estado_pir2=0;
int contador_giro=0;
constinttiempo_detenido=10000;
constinttiempo_adelante=2250;
constintnumero_rondas=2;

intcontador_detenido=0;
int contador_detenido2=0;
int contador_detenido3=0;
int contador_detenido4=0;

intcontador_adelante=0;
int contador_adelante2=0;
int contador_adelante3=0;
int contador_adelante4=0;
void setup()
{
    Serial.begin(9600);
    pinMode(camara,OUTPUT);
    pinMode(ECHO,INPUT);
    pinMode(TRIGGER,OUTPUT);
    pinMode(pir1,INPUT);
    pinMode(pir2,INPUT);
    digitalWrite(pir1,LOW);
    digitalWrite(pir2,LOW);

    digitalWrite(camara,LOW);
}
voidloop()
{}
voidinicializa_estados()
{
    estado_adelante=0;
    estado_adelante_2=0;
    estado_adelante_3=0;

    estado_detenido=0;
    estado_detenido_2=0;
    estado_detenido_3=0;
    estado_detenido_4=0;

    contador_detenido=0;
    contador_detenido2=0;
    contador_detenido3=0;
    contador_detenido4=0;

    contador_adelante=0;
    contador_adelante2=0;
    contador_adelante3=0;
    contador_adelante4=0;
}
voidinicializa_automatiko()
{
    contador_rondas=0;
    estado_general=0;

    estado_adelante=0;
    estado_adelante_2=0;
    estado_adelante_3=0;

    estado_detenido=0;
    estado_detenido_2=0;
    estado_detenido_3=0;
    estado_detenido_4=0;

    contador_detenido=0;
    contador_detenido2=0;
    contador_detenido3=0;
    contador_detenido4=0;

    contador_adelante=0;
    contador_adelante2=0;
    contador_adelante3=0;
    contador_adelante4=0;
    if (digitalRead(camara)==HIGH)
    {digitalWrite(camara,LOW);}
}
voidserialEvent()
{
    while(Serial.available())
    {
        charinChar=(char)Serial.read();
        switch(inChar)
        {
            case 'k': vuelta_derecha(); parar();break;
            case 'p':parar();break;
            case 'w':adelante();break;
        }
    }
}
```

```

        case 's':atras();break;
        case 'a':izquierda();break;
        case 'd':derecha();break;
        case
'm':digitalWrite(camara,HIGH);break;
        case
'n':digitalWrite(camara,LOW);break;
        case 'u': pir_manual();break;
        case 'i':
digitalWrite(pir1,LOW);digitalWrite(pir2,LOW
);
        estado_pir2=0; break;
        case 'c': modomanual=true; break;
        case 'v':
modomanual=false;
inicializa_automatico();
        while
(contador_rondas<numero_rondas)
        {
        inicializa_estados();
        recorrido2();
        if (estado_general>0 ||
modomanual==true)
        {
        parar();
        break;
        }}
        break;
        default:
        if (numingresos==0)
        {
        int ingresorondas=(int)x;
        numingresos++;
        }
        if (numingresos==1)
        {
        int ingresominutos=(int)x;
        numingresos++;
        } } } }
int sensor()
{
    unsignedint tiempo, distancia;
    digitalWrite(TRIGGER, LOW);
    delayMicroseconds(5);
    digitalWrite(TRIGGER, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIGGER, LOW);
    // Calcula la distancia midiendo el
tiempo del estado alto del pin ECHO
    tiempo = pulseIn(ECHO, HIGH);
    // La velocidad del sonido=340m/s o
29 microsegundos por centimetro
    distancia= tiempo/58;
    if (distancia>=15 &&distancia<=20)
    {return 1;}
    else{return 0;}
    delay(10);}
int sensor2()
    {intdistancia=ultrasonic.Ranging(CM);
    if (distancia>=15 &&distancia<=20)
    {return 1;}
    Else{return 0;}
    delay(100);}

```

```

intpirs()
    {int pr1,pr2;
    pr1=digitalRead(pir1);
    pr2=digitalRead(pir2);
    if (pr1!=0 || pr2!=0)
    {return 1;}
    Else{ return 0;} }

void pir_manual()
{int estado_p;
    Serial.println("PIRS MANUAL");
    while (estado_p==0)
    {
    estado_p=pirs();
    if (estado_p==1)
    {
    Serial.println("ALERTA PIRS
MANUAL");
    digitalWrite(camara,HIGH);
    Serial.println("99");
    if (estado_pir2==1)
    {
    vuelta_derecha2();
    parar();estado_pir2=0;}
    break;}} }

void recorrido2()
{
    Serial.println("AUTOMATICO");
    parar();
    delay(5000);
    while(contador_detenido<tiempo_dete
nido)
    {
    intestado_p=pirs();
    if (estado_p==1)
    {
    //Serial.println("ALERTA
PIRS");
    digitalWrite(camara,HIGH);
    Serial.println("99");
    estado_detenido=1;
    contador_detenido=0;
    estado_general++;
    if (estado_pir2==1)
    {
    vuelta_derecha2();parar();estado_pir2=0;}
    break;
    }
    contador_detenido++;
    delay(1);
    }
    if (estado_detenido==0)
    {
    adelante();
    while
(contador_adelante<tiempo_adelante)
    {
    intestado_s=sensor2();
    if (estado_s==1)
    {
    //Serial.println("ALERTA
SENSOR TRAMO 1");
    digitalWrite(camara,HIGH);
    Serial.println("99");
    estado_adelante=1;

```

```

contador_adelante=0;
estado_general++;
parar();
break;
}
//Serial.println(contador_adelante);
contador_adelante++;
delay(1);
}
if (estado_adelante==0)
{
Serial.println("BASE2");
correccion();
parar();
delay(5000);
while
(contador_detenido2<tiempo_detenido)
{
intestado_p=pirs();
if (estado_p==1)
{
//Serial.println("ALERTA
PIRS BASE 2");
digitalWrite(camara,HIGH);
Serial.println("99");
estado_detenido_2=1;
contador_detenido2=0;
estado_general++;
if (estado_pir2==1)
{ vuelta_derecha2();
parar();estado_pir2=0;}
break;
}
contador_detenido2++;
delay(1);
}
if (estado_detenido_2==0)
{
adelante();
while
(contador_adelante2<tiempo_adelante)
{
intestado_s=sensor2();
if (estado_s==1)
{
//Serial.println("ALERTA
SENSOR TRAMO 2");
digitalWrite(camara,HIGH);
Serial.println("99");
estado_adelante_2=1;
contador_adelante2=0;
estado_general++;
parar();
break;
}
//Serial.println(contador_adelante2);
contador_adelante2++;
delay(1);
}
if (estado_adelante_2==0)
{
Serial.println("BASE3");
parar();
vuelta_derecha2();
parar();
delay(5000);

```

```

adelante();
while
(contador_adelante3<tiempo_adelante)
{
int estado_p=pirs();
if (estado_s==1)
{
//Serial.println("ALERTA
SENSOR TRAMO 3");
digitalWrite(camara,HIGH);
Serial.println("99");
estado_adelante_3=1;
contador_adelante3=0;
estado_general++;
if (estado_pir2==1)
{vuelta_derecha2();parar();estado_pir2
=0;}
break;
}
//Serial.println(contador_adelante3);
contador_adelante3++;
delay(1);
}
if (estado_adelante_3==0)
{
adelante();
while
(contador_detenido3<tiempo_detenido)
{
int estado_s=sensor2();
if (estado_p==1)
{
//Serial.println("ALERTA
PIRS BASE 2 VUELTA");
digitalWrite(camara,HIGH);
Serial.println("99");
estado_detenido_3=1;
contador_detenido3=0;
estado_general++;
parar();
break;
}
contador_detenido3++;
delay(1);
}
if (estado_detenido_3==0)
{
Serial.println("BASE 2 VUELTA");
correccion2();
parar();
delay(5000);
while
(contador_adelante4<tiempo_adelante)
{
int estado_p=pirs();
if (estado_s==1)
{
//Serial.println("ALERTA
PIRS BASE 2 VUELTA");
digitalWrite(camara,HIGH);
Serial.println("99");
estado_detenido_4=1;
contador_adelante4=0;
estado_general++;
if (estado_pir2==1)
{vuelta_derecha2();parar();estado_pir2
=0;}

```

```

        break;
    }
    contador_adelante4++;
    delay(1);
    }
    if (estado_detenido_4==0)
    {
        adelante();
        while
(contador_adelante4<tiempo_adelante)
    {
        int estado_s=sensor2();
        if (estado_s==1)
        {
            //Serial.println("ALERTA SENSOR
TRAMO 4");
            digitalWrite(camara,HIGH);
            Serial.println("99");
            contador_adelante4=0;
            estado_general++;
            parar(); break;
        }
        //Serial.println(contador_adelante3);
        contador_adelante4++;
        delay(1);
    }
    parar();
    vuelta_derecha();
    parar();
    contador_rondas++;
    Serial.println(String(contador_rondas))
;
}}}}}}}}
void adelante()
{
    analogWrite(motor1B,220);
    digitalWrite(motor2B,LOW);
    analogWrite(motor1A,225);
    digitalWrite(motor2A,LOW);}

Void atras()
{
    analogWrite(motor2B,220);
    digitalWrite(motor1B,LOW);
    analogWrite(motor2A,220);
    digitalWrite(motor1A,LOW); }

void izquierda()
{
    analogWrite(motor1B,255);
    digitalWrite(motor2B,LOW);
    digitalWrite(motor1A,LOW);

        analogWrite(motor2A,255);
        delay(700);
        parar();}

void derecha()
{
    digitalWrite(motor1B,LOW);
    analogWrite(motor2B,255);
    analogWrite(motor1A,255);
    digitalWrite(motor2A,LOW);
    delay(700);
    parar();}

Void vuelta_derecha()
{
    digitalWrite(motor1B,LOW);
    analogWrite(motor2B,255);
    analogWrite(motor1A,255);
    digitalWrite(motor2A,LOW);
    delay(1150);}

void vuelta_derecha()
{
    digitalWrite(motor1B,LOW);
    analogWrite(motor2B,243);
    analogWrite(motor1A,243);
    digitalWrite(motor2A,LOW);
    delay(1165);}

void vuelta_derecha2()
{
    digitalWrite(motor1B,LOW);
    analogWrite(motor2B,243);
    analogWrite(motor1A,243);
    digitalWrite(motor2A,LOW);
    delay(1185);
}

void correccion() //-----izquierda
{
    analogWrite(motor1B,200);
    digitalWrite(motor2B,LOW);
    analogWrite(motor1A,170);
    digitalWrite(motor2A,LOW);
    delay(600);}

void correccion2()
{
    analogWrite(motor1B,200);
    digitalWrite(motor2B,LOW);
    analogWrite(motor1A,170);
    digitalWrite(motor2A,LOW);
    delay(500);}

Void parar()
{
    digitalWrite(motor1A,LOW);
    digitalWrite(motor2A,LOW);
    digitalWrite(motor1B,LOW);
    digitalWrite(motor2B,LOW);}

```



## ANEXO 5

### "CÓDIGO FUENTE DE LA INTERFAZ EN VISUAL BASIC 2010"

#### Módulo

```
Imports AForge.Video.DirectShow
Imports AForge.Video
Module Module1
Public ExistenDispositivos As Boolean = False
Public DispositivosDeVideo As FilterInfoCollection
Public WithEvents FuenteDeVideo As VideoCaptureDevice = Nothing
Public Imagen As Bitmap
Public indice As Integer
Public write As MJPEGStream
EndModule
```

#### Form2

```
Imports AForge.Video.DirectShow
Public Class Form2
Private Sub Form2_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
Try
Buscar_Dispositivos()
GetSerialPortNames()
Catch ex As Exception
ex.Message.ToUpper()
EndTry
EndSub
Sub GetSerialPortNames()
For Each sp As String In My.Computer.Ports.SerialPortNames
ComboPuertos.Items.Add(sp)
Next
ComboPuertos.Text = ComboPuertos.Items(0).ToString()
EndSub
Public Sub Buscar_Dispositivos() 'Buscamos dispositivos instalados
Try
DispositivosDeVideo = New FilterInfoCollection(FilterCategory.VideoInputDevice)
If DispositivosDeVideo.Count = 0 Then
ExistenDispositivos = False
Else
ExistenDispositivos = True
Cargar_Dispositivos()
Label2.Enabled = True
EndIf
Catch ex As Exception
ex.Message.ToUpper()
EndTry
EndSub
Public Sub Cargar_Dispositivos() 'cargamos los dispositivos encontrados
Try
For i As Integer = 0 To DispositivosDeVideo.Count - 1
ComboCamaras.Items.Add(DispositivosDeVideo(i).Name.ToString())
Next
ComboCamaras.Text = ComboCamaras.Items(0).ToString()
Catch ex As Exception
ex.Message.ToUpper()
EndTry
EndSub
Private Sub Label2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Label2.Click
Try
indice = ComboCamaras.SelectedIndex
Form1.Show()
Me.Close()
Catch ex As Exception
```

```

        MsgBox(ex.ToString)
    End Try
End Sub
End Class

```

## Form3

```

Imports System.IO
Imports iTextSharp.text
Imports iTextSharp.text.pdf
Public Class Form3
Private Sub Form3_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    'TODO: carga datos en la tabla 'BaserobotDataSet.Tabla_Registro'
    Me.Tabla_RegistroTableAdapter.Fill(Me.BaserobotDataSet.Tabla_Registro)
End Sub
Public Function GetColumnasSize() As Single()
    Dim values As Single() = New Single(DataGridView1.ColumnCount - 1) {}
    For i As Integer = 0 To DataGridView1.ColumnCount - 1
        values(i) = CSng(DataGridView1.Columns(i).Width + 20)
    Next
    Return values
End Function
Public Sub ExportarDatosPDF(ByVal document As Document)
    'Se crea un objeto PdfPTable con el numero de columnas del DataGridView.
    Dim datatable As New PdfPTable(DataGridView1.ColumnCount)
    datatable.DefaultCell.Padding = 3 'Se asignan alguns propiedades pra l diseño dl PDF.
    Dim headerwidths As Single() = GetColumnasSize()
    datatable.SetWidths(headerwidths)
    datatable.WidthPercentage = 70
    datatable.DefaultCell.BorderWidth = 2
    datatable.DefaultCell.HorizontalAlignment = Element.ALIGN_CENTER
    'Se crea el encabezado en el PDF.
    Dim encabezado As New Paragraph("REPORTE DEL DIA", New Font(Font.Name = "Tahoma",
20, Font.Bold))
    Dim separa As New Paragraph(" ") 'Se crea el texto abajo del encabezado. Se
capturan los nombres de las columnas del DataGridView.
    For i As Integer = 0 To DataGridView1.ColumnCount - 1
        datatable.AddCell(DataGridView1.Columns(i).HeaderText)
    Next
    datatable.HeaderRows = 1
    datatable.DefaultCell.BorderWidth = 1
    For i As Integer = 0 To DataGridView1.Rows.Count - 2
        For j As Integer = 0 To DataGridView1.Columns.Count - 1
            datatable.AddCell((DataGridView1(j, i).Value).ToString)
        Next
        datatable.CompleteRow()
    Next
    datatable.AddCell("") 'da 2 tab entre columnas
    datatable.CompleteRow() 'imprime resultados
    document.Add(encabezado) 'Se agrega etiquetas
    document.Add(separa)
    document.Add(datatable) 'document.Add(texto)
End Sub
Private Sub BtnExportar_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnExportar.Click
    Try
        ' Intentar generar el documento.
        Dim doc As New Document(PageSize.A4.Rotate(), 10, 10, 10)
        ' Path que guarda el reporte en el escritorio de windows (Desktop).
        Dim filename As String =
Environment.GetFolderPath(Environment.SpecialFolder.DesktopDirectory) + "\Reporte.pdf"
        Dim file As New FileStream(filename, FileMode.Create, FileAccess.Write,
FileShare.ReadWrite)
        PdfWriter.GetInstance(doc, file)
        doc.Open()
        ExportarDatosPDF(doc)
        doc.Close()
        Process.Start(filename)
    Catch ex As Exception

```

```

        MessageBox.Show(ex.ToString) 'Si el intento es fallido, mostrar MsgBox.
        MessageBox.Show("No se puede generar el documento PDF.", "Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error)
    End Try
End Sub
End Class

```

## Principal

```

Imports AForge.Video.DirectShow
Imports AForge.Video
Imports System.IO
Public Class Form1
    Dim conexion As New SqlConnection, Dim comando As New SqlCommand
    Public trama A sString, Public Delegate Subleer(), Public leer1 Asleer
    Dim numrondas. Dim monitoreo, Dim inputrondas, Dim inputmonitoreo
    Sub flechas_motor()
        flecha_avanzar_on.Visible = True
        flecha_regresar_on.Visible = True
        flecha_derecha_on.Visible = True
        flecha_izquierda_on.Visible = True
        flecha_detener_on.Visible = True
        flecha_avanzar_off.Visible = False
        flecha_regresar_off.Visible = False
        flecha_derecha_off.Visible = False
        flecha_izquierda_off.Visible = False
        flecha_detener_off.Visible = False
    End Sub
    Sub flechas_camara()
        camara_on.Visible = True
        camara_off.Visible = True
        camara_on_1.Visible = False
        camara_off_1.Visible = False
    EndSub
    Sub flechas_sensor()
        flecha_sensor_on.Visible = True
        flecha_sensor_off.Visible = True
        flecha_sensor_off1.Visible = False
        flecha_sensor_on1.Visible = False
    End Sub
    Private Sub flecha_avanzar_on_Click(ByVal sender AsSystem.Object, ByVal e
    AsSystem.EventArgs) Handlesflecha_avanzar_on.Click
        Call flechas_motor()
        flecha_avanzar_off.Visible = True
        flecha_avanzar_on.Visible = False
        estado_robot.Text = "Avanzar hacia adelante"
        SerialPort1.Write("w")
    EndSub
    PrivateSubflecha_regresar_on_Click(ByVal sender AsSystem.Object, ByVal e
    AsSystem.EventArgs) Handlesflecha_regresar_on.Click
        Callflechas_motor()
        flecha_regresar_off.Visible = True
        flecha_regresar_on.Visible = False
        estado_robot.Text = "Regresar"
        SerialPort1.Write("s")
    EndSub
    PrivateSubflecha_derecha_on_Click(ByVal sender AsSystem.Object, ByVal e
    AsSystem.EventArgs) Handlesflecha_derecha_on.Click
        Callflechas_motor()
        flecha_derecha_off.Visible = True
        flecha_derecha_on.Visible = False
        estado_robot.Text = "Giro Derecho"
        SerialPort1.Write("d")
    EndSub
    PrivateSubflecha_izquierda_on_Click(ByVal sender AsSystem.Object, ByVal e
    AsSystem.EventArgs) Handlesflecha_izquierda_on.Click
        Callflechas_motor()
        flecha_izquierda_off.Visible = True
        flecha_izquierda_on.Visible = False
    EndSub

```

```

estado_robot.Text = "Giro Izquierdo"
SerialPort1.Write("a")
EndSub
PrivateSub flecha_detener_on_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles flecha_detener_on.Click
    Call flechas_motor()
    flecha_detener_off.Visible = True
    flecha_detener_on.Visible = False
    estado_robot.Text = "Robot Detenido"
    SerialPort1.Write("p")
EndSub
PrivateSub camara_on_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles camara_on.Click
    Call flechas_camara()
    camara_on.Visible = False
    camara_on_1.Visible = True
    video.Visible = True
    SerialPort1.Write("m")
    Try
        encender_camara()
    Catch ex As Exception
        ex.Message.ToUpper()
    EndTry
EndSub
PrivateSub camara_off_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles camara_off.Click
    Call flechas_camara()
    camara_off.Visible = False
    camara_off_1.Visible = True
    video.Visible = False
    SerialPort1.Write("n")
    Try
        Terminar()
    Catch ex As Exception
        ex.Message.ToUpper()
    EndTry
EndSub
PrivateSub modo_manual_CheckedChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles modo_manual.CheckedChanged
    Call flechas_motor()
    Call flechas_camara()
    control_motor.Visible = True
    control_camara.Visible = True
    control_automatico.Visible = False
    intruso_detectado.Visible = False
    If (Val(TextBox1.Text)) <> 99 Then
        video.Visible = False
    End If
    estado_robot.Text = "Robot Detenido"
    Label9.Text = "Modo de Operación Manual"
EndSub
PrivateSub modo_automatico_CheckedChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles modo_automatico.CheckedChanged
    Call flechas_motor()
    Call flechas_camara()
    control_motor.Visible = False
    control_camara.Visible = False
    control_automatico.Visible = True
    Label9.Text = "Modo de Operación Automático"
    Timer1.Start()
EndSub
PrivateSub Form1_FormClosing(ByVal sender As Object, ByVal e
As System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing
    SerialPort1.Close()
EndSub
PrivateSub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
    SerialPort1.PortName = Form2.ComboPuertos.SelectedItem

```

```

Try
SerialPort1.Open()
conexion.ConnectionString = "Data Source=(local)\SQLSERVER08;Initial
Catalog=Baserobot;Integrated Security=True"
        conexion.Open()
        comando.Connection = conexion
Catch ex AsException
        MsgBox(ex.ToString)
        Me.Close()

Finally
If SerialPort1 IsNothingThen
SerialPort1.Close()
EndIf
EndTry
If (modo_manual.Checked = TrueAnd SerialPort1.IsOpen) Then
SerialPort1.Write("p")
EndIf
EndSub
Private Sub SerialPort1_DataReceived(ByVal sender AsObject, ByVal e
AsSystem.IO.Ports.SerialDataReceivedEventArgs) Handles SerialPort1.DataReceived
        leer1 = Newleer(AddressOfproceso)
        trama = SerialPort1.ReadLine 'actua cuando le envia un salto de linea
Me.Invoke(leer1) 'llamo la delegacion
End Sub
Public Sub proceso()
        TextBox1.Text = trama
If Val(TextBox1.Text) = 99 Then
intruso_detectado.Visible = True
video.Visible = True
Try
encender_camara()
Catch ex AsException
ex.Message.ToUpper()
EndTry
EndIf
        If (Val(TextBox1.Text) = 1) Then
                Label13.Text = "1/" + Str(numrondas)
        End If
        If (Val(TextBox1.Text) = 2) Then
                Label13.Text = "2/" + Str(numrondas)
        End If
EndSub
Private Sub video1(ByVal sender AsObject, ByValeventArgsAsNewFrameEventArgs)
Try
Imagen = DirectCast(eventArgs.Frame.Clone(), Bitmap)
video.Image = Imagen
Catch ex AsException
ex.Message.ToUpper()
EndTry
EndSub
PublicSubTerminar()
Try
IfNot (FuenteDeVideoIsNothing) Then
IfFuenteDeVideo.IsRunningThen
FuenteDeVideo.SignalToStop()
FuenteDeVideo = Nothing
EndIf
EndIf
Catch ex AsException
ex.Message.ToUpper()
EndTry
EndSub
PublicSubencender_camara()
IfExistenDispositivosThen
FuenteDeVideo = NewVideoCaptureDevice(DispositivosDeVideo(indice).MonikerString)
AddHandlerFuenteDeVideo.NewFrame, NewNewFrameEventHandler(AddressOf video1)
FuenteDeVideo.Start()
EndIf

```

```

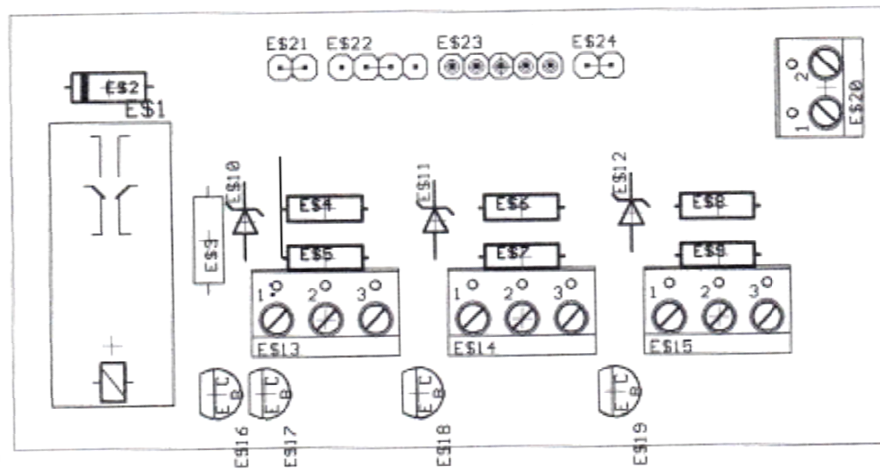
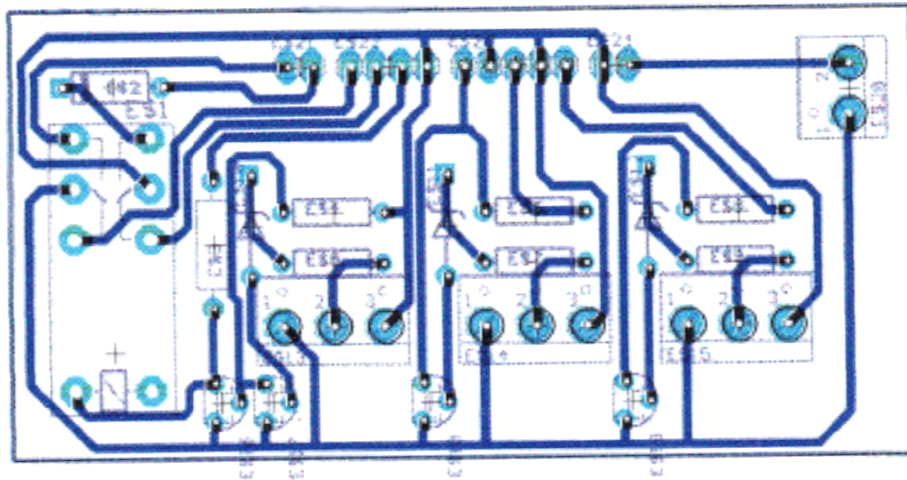
EndSub
PrivateSubmodo_manual_Click(ByVal sender AsObject, ByVal e AsSystem.EventArgs)
Handlesmodo_manual.Click
SerialPort1.Write("1")
Timer1.Enabled = False
EndSub
Private Sub PictureBox4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles flecha_sensor_on.Click
Call flechas_sensor()
flecha_sensor_on.Visible = False
flecha_sensor_off1.Visible = True
flecha_avanzar_on.Enabled = False
flecha_derecha_on.Enabled = False
flecha_detener_on.Enabled = False
flecha_izquierda_on.Enabled = False
flecha_regresar_on.Enabled = False
SerialPort1.Write("u")
End Sub
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Timer1.Tick
Label15.Text = Now.ToString("hh:mm:ss")
End Sub
Private Sub flecha_sensor_off_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles flecha_sensor_off.Click
Call flechas_sensor()
flecha_sensor_off.Visible = False
flecha_sensor_off1.Visible = True
flecha_avanzar_on.Enabled = True
flecha_derecha_on.Enabled = True
flecha_detener_on.Enabled = True
flecha_izquierda_on.Enabled = True
flecha_regresar_on.Enabled = True
SerialPort1.Write("i")
End Sub
Private Sub flecha_izquierda_off_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles flecha_izquierda_off.Click
flecha_izquierda_on.Visible = True
flecha_izquierda_off.Visible = False
End Sub
Private Sub flecha_derecha_off_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles flecha_derecha_off.Click
flecha_derecha_on.Visible = True
flecha_derecha_off.Visible = False
End Sub
Private Sub btn_iniciaauto_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btn_iniciaauto.Click, btn_paraauto.Click
SerialPort1.Write("v")
Label14.Text = Now.ToString("hh:mm")
End Sub
Private Sub btnrondas_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnrondas.Click
inputrondas = InputBox("INGRESE CANTIDAD DE RONDAS <=10", "RONDAS")
If (Val(inputrondas) > 0 And Val(inputrondas) <= 10) Then
Label13.Text = "0/" + inputrondas
End If
End Sub
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
inputmonitoreo = InputBox("INGRESE TIEMPO DE MONITOREO <=5", "MONITOREO")
If (Val(inputmonitoreo) > 0 And Val(inputmonitoreo) <= 5) Then
Label20.Text = inputmonitoreo + " min."
End If
End Sub
Private Sub Btnbitacora_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Btnbitacora.Click
Form3.ShowDialog()
End Sub
End Class

```

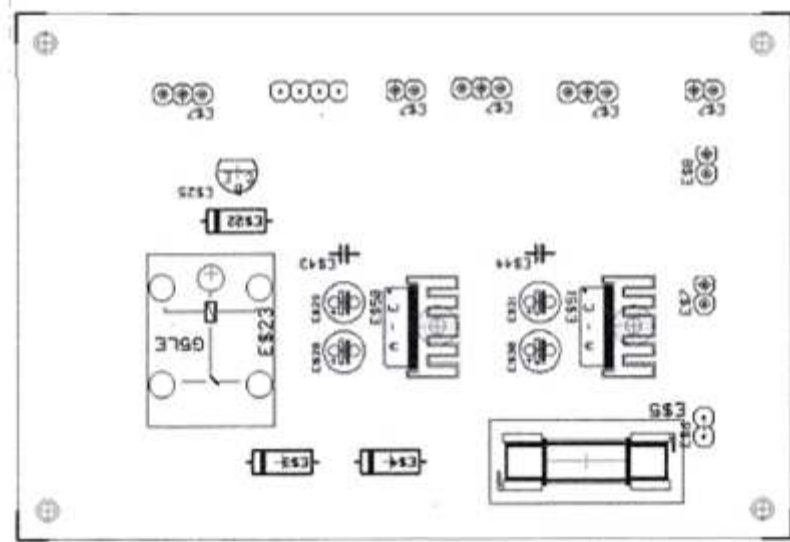
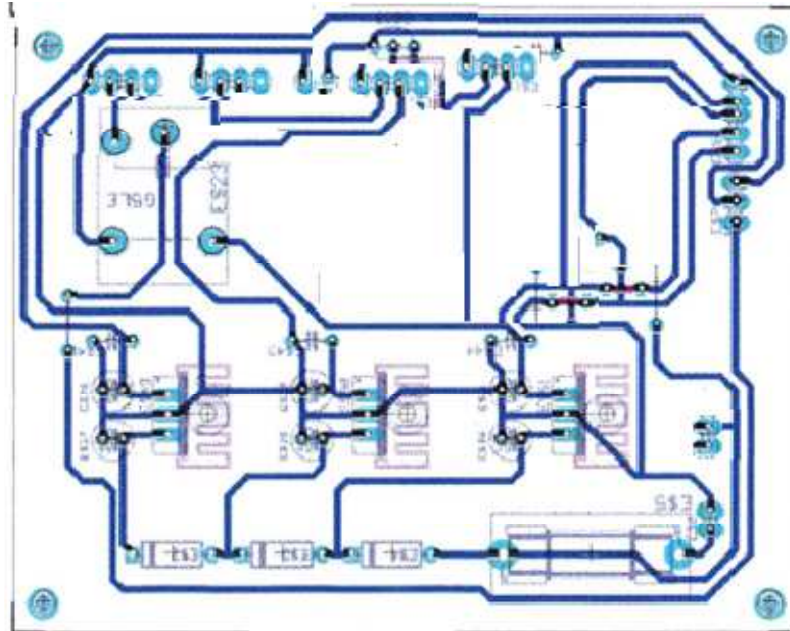
## ANEXO 6

### “DIAGRAMAS DE LOS CIRCUITOS IMPRESOS”

Circuito Impreso Tarjeta medidor de estado de batería.

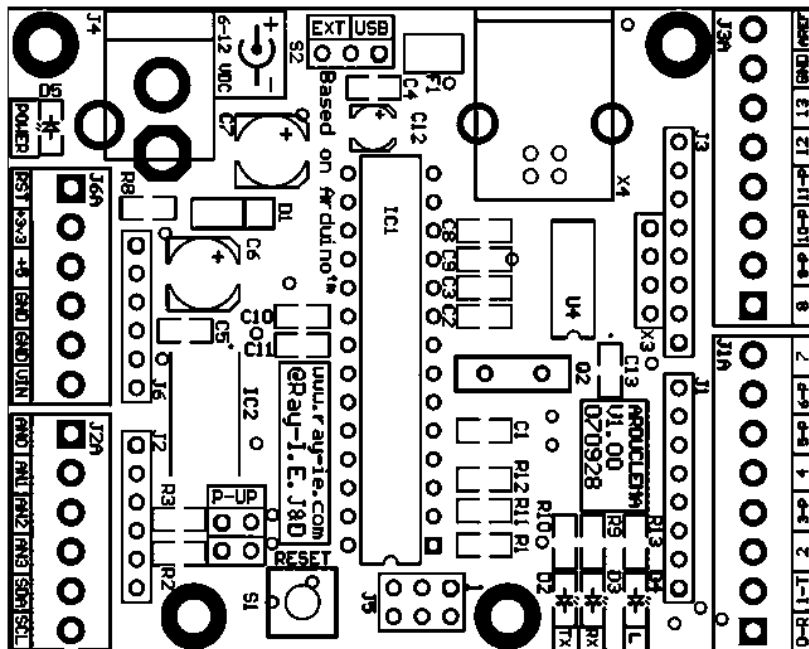
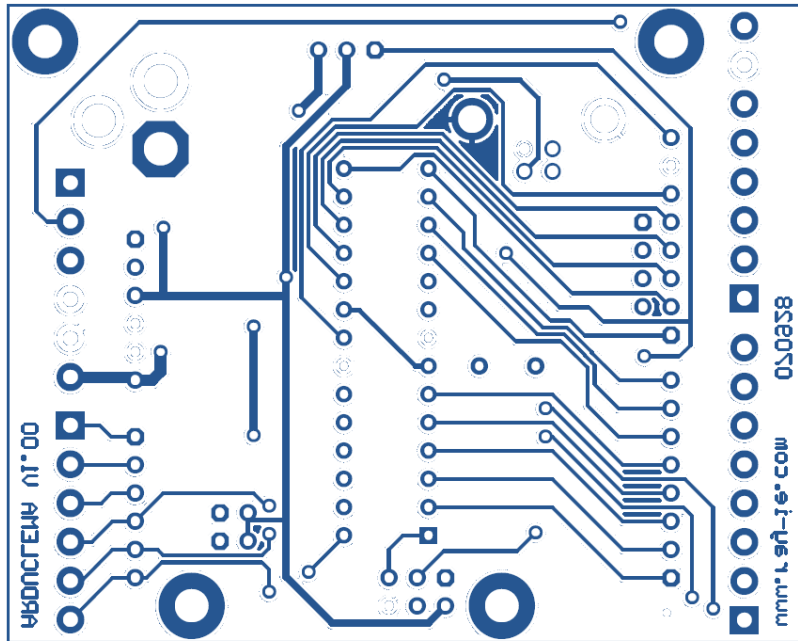


Circuito Impreso Tarjeta reguladora de Voltaje.



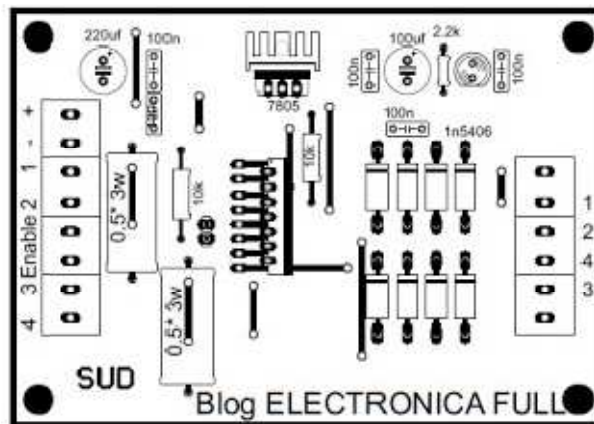
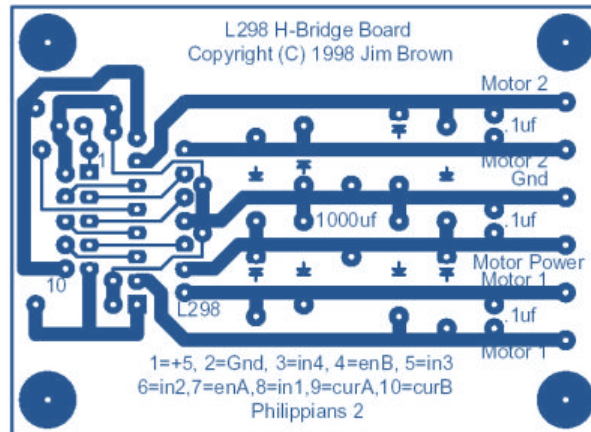


# Circuito Impreso Tarjeta Arduino.<sup>21</sup>



<sup>21</sup> [http://www.ray-ie.com/webray\\_015.htm](http://www.ray-ie.com/webray_015.htm)

## Circuito impreso tarjeta Puente H L298N<sup>22</sup>



<sup>22</sup> <http://electronicabolivia.blogspot.com/2010/05/puente-h.html>  
<http://blog.rolandopalermo.com/2013/04/control-motor-puente-h-l298.html>

## ANEXO 7

### “MANUAL DE USUARIO”

Es un sistema robótico desarrollado para complementar la tarea de vigilancia y seguridad; detecta obstáculos en su camino cuando se encuentra dirigiéndose hacia el recorrido que ha sido programado. El sensor ultrasónico reconocerá objetos en su frente siempre que cuenten con características de superficie adecuadas y de ubicación y altura, los sensores PIR detectarán personas en movimiento que irrumpen en el espacio monitoreado.

La interfaz del usuario inicia con una ventana donde se selecciona los puertos que usará el Xbee y la cámara. Seleccionamos el puerto COM que este ocupando el Xbee, esto se puede verificar ingresando a Administrador de dispositivos. Al seleccionar la cámara se debe conocer el nombre de la marca que está usando el usuario para el control de vigilancia y seguridad. Damos en Aplicar y nos aparecerá la pantalla principal.

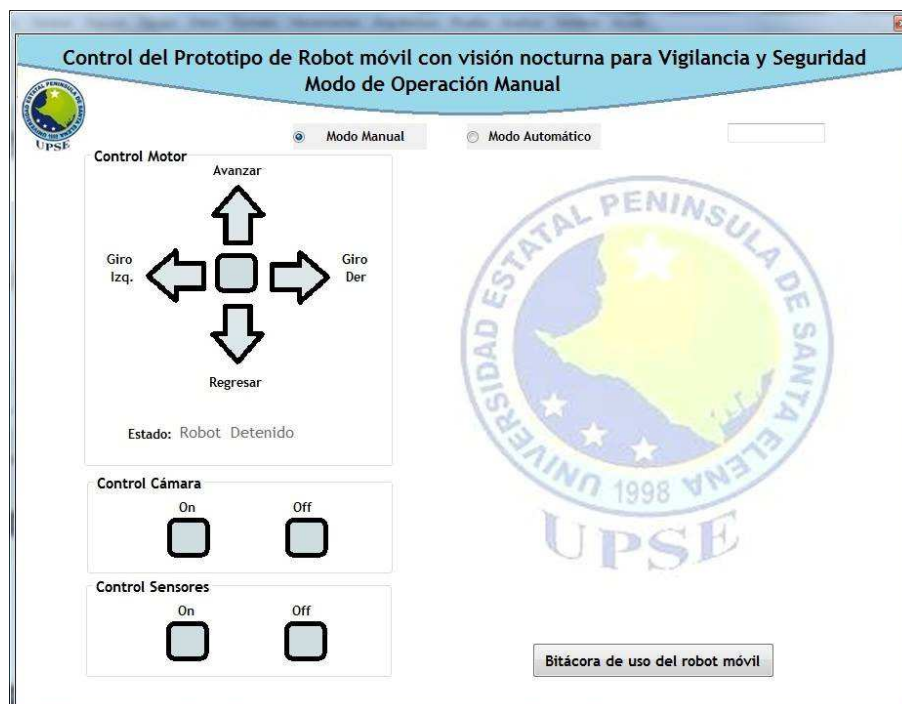
Pantalla inicial del programa de monitorización de vigilancia  
y seguridad



## Modo Tele-Operado

En la pantalla principal, se tiene dos botones principales para la activación del modo de operación del prototipo. Se inicia en modo Tele-Operado.

Pantalla principal del programa de monitorización de vigilancia y seguridad, Modo Tele-Operado.



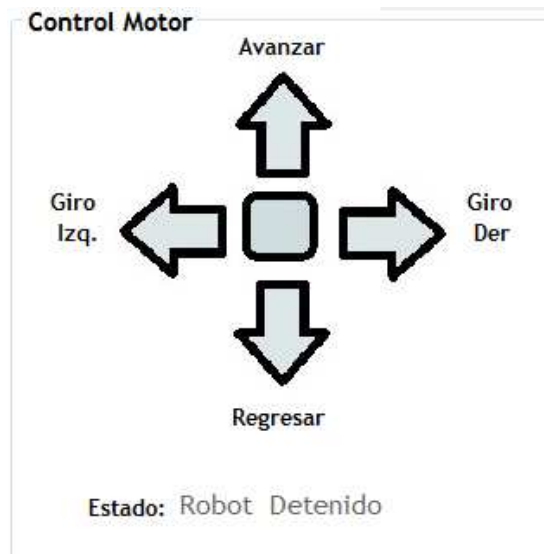
Para la navegación del robot móvil, nos guiamos en las flechas para seleccionarlas e indicar el sentido de dirección del robot. Se puede Avanzar, Retroceder, Giro Izquierda, Giro Derecha y Detener. Estas funciones se habilitan cuando el robot se encuentra en modo de Operación Tele-Operado:

### Funciones de flechas y botones

Controla el movimiento del robot. En la parte inferior de las flechas encontramos un texto que nos indica el estado en que se encuentra el robot.

- **Flecha Arriba.-** Avanza
- **Flecha Abajo.-** Retrocede
- **Flecha Derecha.-** Giro hacia la derecha
- **Flecha Izquierda.-** Giro hacia la izquierda
- **Botón Centro.-** Detener

Modo Tele-Operado para navegación del prototipo.



### Control Cámara

Modo Tele-Operado para activación de cámara del prototipo.



### Control Sensores

Modo Tele-Operado para activación de monitoreo detección de intrusos.



Estando en Modo Tele-Operado podemos activar y desactivar la cámara; como también los sensores según nuestra conveniencia para monitorizar el entorno.

Figura. Modo Tele-Operado. Cámara del prototipo activada.



### Modo Automático

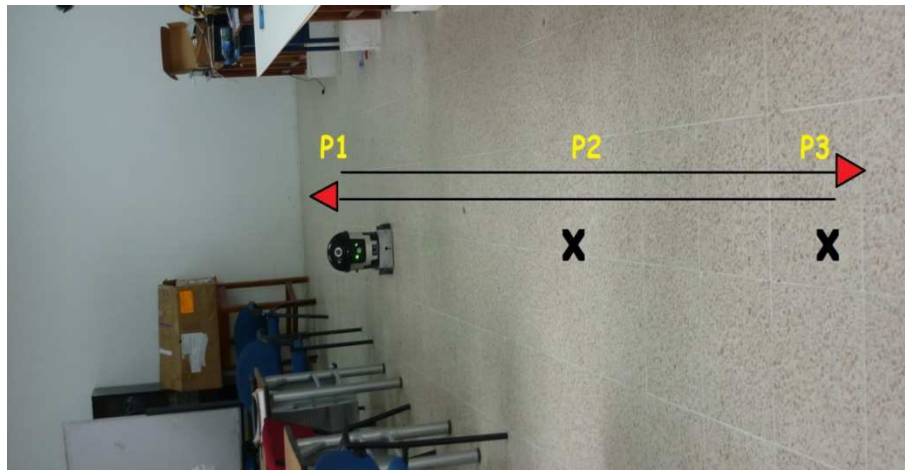
En este modo el prototipo de robot móvil, ejecuta una rutina programada, que consiste en:

- 1.- Monitorear el área por 1 minuto en el punto P1.
- 2.- Avanzar del punto P1 al punto P2.
- 3.- Monitorear el área por 1 minuto en el punto P2.
- 4.- Avanzar del punto P2 al punto P3.
- 5.- Dar la vuelta el prototipo para retornar el trayecto recorrido.
- 6.- Monitorear el área por 1 minuto en el punto P3.
- 7.- Avanzar del punto P3 al punto P2.

- 8.-Monitorear el área por 1 minuto en el punto P2.
- 9.- Avanzar del punto P2 al punto P1.
- 10.-Dar la vuelta el prototipo para quedar en posición inicial.

Esta es la ruta programada en Arduino; la ejecuta según el área prediseñada, mientras avanza de un punto a otro, va sensando con el ultrasonido, si detecta un objeto a 20cm se detiene enciende la cámara y pasa a Modo Tele-Operado, si no detecta ningún objeto mientras va avanzando llega al punto previsto y comienza el estado de monitoreo por un minuto, sensando con el PIR; si detecta el movimiento de una persona se detiene, enciende la cámara y pasa a Modo Tele-Operado, si no detecta ningún objeto mientras este en el estado de monitoreo sigue ejecutando la ruta hasta terminar el recorrido programado.

Modo Automático. Ejecución Ronda.



La ventana de Modo Automático, nos muestra información acerca de las rondas a realizar. Para ejecutar una ruta se debe seleccionar el número de rondas y el tiempo de monitoreo.

Se debe ingresar numero de rondas para el recorrido de la ruta de vigilancia y seguridad, Modo Automático.

RONDAS

INGRESE CANTIDAD DE RONDAS <=10

Aceptar

Cancelar

2

Ingresar el tiempo de monitoreo; es el tiempo de funcionamiento de los sensores PIR para detectar intrusos; al seleccionar INICIAR se envía la orden para ejecutar la ruta programada.

MONITOREO

INGRESE TIEMPO DE MONITOREO <=5

Aceptar

Cancelar

1

Pantalla principal del programa de monitorización de vigilancia y seguridad, Modo Automático.

Control del Prototipo de Robot móvil con visión nocturna para Vigilancia y Seguridad  
Modo de Operación Automático

UPSE

Modo Manual  Modo Automático

Rutas

Rondas: 0/2 Establecer

Inicio: 01:57

Reloj: 01:58:01

Información

Tiempo por ronda: 20 min.

Tiempo de monitoreo: 1 min. Establecer

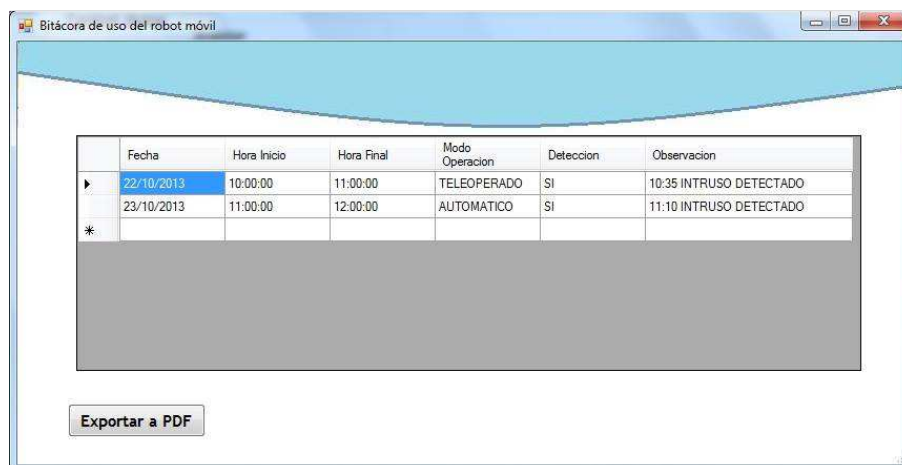
Iniciar Detener

Bitácora de uso del robot móvil



En el Modo Automático muestra la información de la ruta ejecutada, como el número de rondas dadas, el Tiempo y Fecha de inicio del recorrido; la hora actual. Y dependiendo de lo que ha sentido muestra la cámara. Se podrá generar reportes del uso del robot móvil con visión nocturna para vigilancia y seguridad.

### Bitácora el uso del Robot Móvil para vigilancia y seguridad



Fecha	Hora Inicio	Hora Final	Modo Operación	Detección	Observación
22/10/2013	10:00:00	11:00:00	TELEOPERADO	SI	10:35 INTRUSO DETECTADO
23/10/2013	11:00:00	12:00:00	AUTOMATICO	SI	11:10 INTRUSO DETECTADO
*					

Los informes son presentados en formato PDF; se puede generar un informe general con todos los registros desde su implementación.

## Prototipo

### Requisitos Técnicos.-

- 1.- Instalar la aplicación de Control de Monitoreo y Vigilancia.
- 2.- Instalar SQL Server Express 2008.
- 3.- Instalar los driver de la cámara del prototipo.
- 4.- Instalar los driver del módulo de comunicación Xbee.

**Encendido del equipo.-** Un botón del tipo pulsador en la estructura del robot; enciende el prototipo, y un interruptor en la base enciende el Arduino. La ejecución se inicia en modo tele-operado. El robot se encuentra a la espera de alguna orden que llegue por el modulo de

comunicación a través de la interfaz de usuario; quien decide el modo de operación.



**Mantenimiento.-** El robot móvil no reviste de mayores atenciones, salvo las baterías de 12V y 9V que necesita ser recargada después de cada sesión de navegación. Y verificar que no se haya desconectado nada de los cables del interior.

## GLOSARIO

**API.-** (Application Programming Interface). Interfaz de programación de aplicación es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

**ASCII.-** (American Standard Code for Information Interchange - Código Estándar Estadounidense para el Intercambio de Información), es un código de caracteres basado en el alfabeto latino, tal como se usa en inglés moderno y en otras lenguas occidentales.

**DC.-** La corriente continua se refiere al flujo continuo de carga eléctrica a través de un conductor entre dos puntos de distinto potencial, que no cambia de sentido con el tiempo.

**GPU GPL.-** (GNU General Public License) La Licencia Pública General de GNU es la licencia más ampliamente usada en el mundo del software y garantiza a los usuarios finales (personas, organizaciones, compañías) la libertad de usar, estudiar, compartir y modificar el software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

**ISV.-** El Programa de Licencias ISV de Microsoft es un programa de licencias de software a nivel mundial que ofrece a los Vendedores independientes de software una forma cómoda de licenciar productos Microsoft e integrarlos en una aplicación de software de negocio.

**LENSES FRESNEL.-** Llamada así por su inventor Augustin Fresnel, es un diseño de lentes que permite la construcción de lentes de gran apertura y una corta distancia focal sin el peso y volumen de material que debería usar en una lente de diseño convencional.

**TTL.-** (Transistor-Transistor Logic) es decir, «lógica transistor a transistor». Es una familia lógica o lo que es lo mismo, una tecnología de construcción de circuitos electrónicos digitales. En los componentes fabricados con tecnología TTL los elementos de entrada y salida del dispositivo son transistores bipolares.