



**UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO**  
**CENTRO UNIVERSITARIO UAEM ATLACOMULCO**



“Robot móvil con percepción auditiva y sistema de control a distancia”

# **T E S I S**

Que para obtener el Título de:

**Ingeniera en Computación**

Presenta:

**Heivilina Pérez Arias**

Director de Tesis:

**Dr. Everardo Efrén Granda Gutiérrez**

**Atlacomulco, México; octubre de 2019**

## RESUMEN

En este documento se presenta el diseño e implementación de un robot móvil con percepción auditiva y sistema de control a distancia, a partir de un mini robot con locomoción por ruedas, manipulado mediante comandos de voz. El objetivo principal del proyecto es la inclusión de dos métodos de comunicación del usuario con el robot: a) local, con la integración de un módulo de reconocimiento de voz instalado en la estructura del prototipo, y b) remoto, a través de una aplicación denominada “*Mi robot móvil*”, desarrollada en la plataforma de programación *MIT App Inventor 2*, diseñada para el entorno Android. El propósito de esta aplicación es establecer comunicación vía Bluetooth entre un teléfono celular y el robot, además del envío de instrucciones emitidas por medio de comandos de voz, aprovechando el reconocimiento de voz de Google.

Para el desarrollo del proyecto se hizo uso de la metodología de desarrollo de software en espiral, que cuenta con cuatro fases: planeación, modelado, construcción y despliegue. Este método fue seleccionado al efectuar un análisis para conocer la secuencia de pasos o etapas a cumplir con la realización de un sistema. Se identificó que dicha metodología corresponde con un modelo evolutivo para el desarrollo de software, que permite la integración de elementos de hardware. Al finalizar cada iteración, se obtienen actividades de retroalimentación para la mejora del diseño y de la programación.

Una característica destacable del robot es la inclusión de un sensor ultrasónico que permite la detección de objetos próximos mientras ejecuta el recorrido de una trayectoria, lo que le confiere la habilidad de evadir los obstáculos presentes, evitando choques o incidentes que causen daños a su misma estructura.

Para la etapa de pruebas se empleó un espacio confinado con paredes de cartón y obstáculos para validar su funcionamiento. Como resultado, se obtiene el desplazamiento satisfactorio del móvil dentro de un área regular, ejecutando los movimientos ordenados por el usuario mediante comandos de voz, haciendo uso de cualquiera de los dos métodos de comunicación establecidos. Independiente a la forma de manejo, el robot es capaz de realizar la evasión de obstáculos durante su recorrido.

**Palabras clave:** Robot móvil, metodología en espiral, reconocimiento de voz.

## **ABSTRACT**

The design process and implementation of a mobile robot with auditive perception and remote control is presented in this document. The basis of the project is a wheel-locomotion mini robot which is manipulated through voice commands. The main objective of the project is the inclusion of two methods for user-robot interaction: a) local, by means of the integration of a voice recognition module located in the robot structure, and b) remote, by using a specifically designed mobile application named "My mobile robot", which has been developed in the *MIT App Inventor 2* programming platform, and designed for Android environment. The purpose of this application is to provide communication via Bluetooth between a cell phone and the robot, in addition to send voice instructions by means of the Google voice recognizer.

The spiral software development methodology was employed for the development of the project, considering its four phases: planning, modeling, construction and deployment. This method was selected by performing an analysis to know the sequence of steps or stages to comply with the realization of a system. It was identified that this is an evolutionary model for software development that allows the integration of hardware elements. At the end of each iteration, feedback activities are obtained to improve the design and programming.

A noticeable feature of the robot is the use of an ultrasonic sensor that allows the detection of nearby objects while the robot is moving along a specified trajectory, thus allowing obstacle evasion, avoiding collisions or incidents that could damage the robot structure.

During the test stage, a confined space with cardboard walls and obstacles was used to validate the robot operation. The movement of the mobile robot within a regular area, was satisfactorily achieved, executing the movements ordered by the user by means of voice commands through both: local and remote communication methods. Regardless of the handling method, the robot can avoid obstacles during its route.

**Keywords:** Mobile robot, spiral methodology, voice recognition.

# ÍNDICE

DEDICATORIAS .....	i
AGRADECIMIENTOS .....	ii
RESUMEN.....	iii
ABSTRACT.....	iv
ÍNDICE .....	v
ÍNDICE DE TABLAS .....	viii
ÍNDICE DE FIGURAS.....	ix
1 INTRODUCCIÓN.....	1
2 PLANTEAMIENTO DEL PROBLEMA .....	3
2.1 Definición del problema.....	3
2.2 Objetivos de investigación .....	4
2.2.1 Objetivo general .....	4
2.2.2 Objetivos específicos .....	5
2.3 Preguntas de investigación .....	5
2.4 Justificación.....	6
2.5 Impactos .....	7
3 META DE INGENIERÍA.....	8
4 ESTADO DEL ARTE .....	9
4.1 Robótica.....	9
4.1.1 Leyes de la robótica .....	10
4.1.2 Robótica móvil.....	10
4.1.3 Clasificación de robots móviles .....	12
4.2 Comunicación inalámbrica para control de robots móviles .....	15
4.3 Microcontrolador Arduino .....	16

4.4	Reconocimiento de voz .....	18
4.4.1	Reconocimiento de voz de la aplicación “Mi robot móvil” .....	18
4.4.2	Reconocimiento de voz del módulo Elechouse v3. ....	19
4.5	Trabajos reportados en la literatura .....	19
5	MÉTODO .....	35
5.1	Estrategia para el desarrollo de software.....	35
5.2	Requerimientos o especificaciones.....	37
5.3	Modelado.....	38
5.3.1	Hardware.....	38
5.3.2	Software .....	43
5.4	Construcción.....	51
5.4.1	Ensamble del robot móvil .....	51
5.4.2	Instalación y configuración del software de programación Arduino .....	52
5.4.3	Desarrollo del algoritmo para la evasión de obstáculos.....	54
5.4.4	Comunicación y configuración del módulo de reconocimiento de voz.....	57
5.4.5	Comunicación y configuración del módulo Bluetooth. ....	59
5.4.6	Desarrollo de Aplicación: Mi robot móvil.....	61
6	RESULTADOS Y DISCUSIÓN .....	68
6.1	Despliegue .....	68
6.1.1	Funcionamiento de la aplicación: Mi robot móvil .....	68
6.1.2	Reconocimiento de los comandos de voz por el módulo.....	70
6.1.3	Detección de obstáculos .....	71
6.1.4	Comunicación Arduino - Módulo Bluetooth y aplicación móvil.....	72
6.1.5	Funcionamiento inicial del robot móvil .....	76
6.1.6	Funcionamiento final del robot móvil.....	78

CONCLUSIONES .....	84
REFERENCIAS.....	86
ANEXOS .....	91

## ÍNDICE DE TABLAS

Tabla 4.1. Matriz de referencias para determinar el estado del arte. ....	32
Tabla 5.1 Componentes del robot móvil.....	51
Tabla 5.2. Acciones de los motores para realizar la evasión de obstáculos.....	56
Tabla 5.3 Comandos básicos para la configuración del módulo Bluetooth.....	60
Tabla 6.1. Relación: Variable/ instrucción.....	74
Tabla 6.2. Datos sobre aciertos del reconocimiento de voz.....	80
Tabla 6.3. Datos de reconocimiento de comandos a distancia.....	82

# ÍNDICE DE FIGURAS

Figura 4.1. Sistemas necesarios para el funcionamiento de un robot móvil. ....	11
Figura 4.2. Locomoción por ruedas (Barrientos Sotelo, et al., 2007). ....	13
Figura 4.3. Robot chita de DARPA (Ramírez, 2017). ....	14
Figura 4.4. Analogía de seres biológicos y robot con patas. ....	14
Figura 4.5. Locomoción por orugas (Barrientos Sotelo, et al., 2007). ....	15
Figura 4.6. Diagrama a bloques del algoritmo de reconocimiento de voz (Simancas & Meléndez Pertuz, 2017) ....	22
Figura 4.7. Interfaz gráfica (Baquero Romero, et al., 2012) ....	23
Figura 4.8. Diagrama a bloques de los pasos realizados para el desarrollo de la aplicación mencionada. ....	24
Figura 4.9. Diagrama de flujo del método para el desarrollo de la aplicación de reconocimiento de voz. ....	25
Figura 4.10. Diagrama a bloques del método para la simulación del sistema de reconocimiento de voz. ....	28
Figura 5.1. Fases de la metodología de desarrollo de software en espiral (Pressman, 2010). ....	36
Figura 5.2. Representación de la configuración de tres ruedas o triciclo. ....	38
Figura 5.3. Diagrama a bloques de la estructura del robot móvil. ....	42
Figura 5.4. Diagrama de conexión del sistema electrónico. ....	42
Figura 5.5. Diagrama de casos de uso del sistema para manipulación del robot móvil...	43
Figura 5.6. Diagrama de flujo del funcionamiento del robot móvil. ....	45
Figura 5.7. Diagrama de flujo del método de comunicación remota (continuación del diagrama de la Figura 5.6. ....	46
Figura 5.8. Diagrama de flujo correspondiente a la evasión de obstáculos. ....	47
Figura 5.9. Diagrama de flujo de la comunicación y configuración del módulo Bluetooth HC-05 con el Arduino MEGA. ....	49
Figura 5.10. Diagrama de flujo del funcionamiento de la aplicación Mi robot móvil. ....	50
Figura 5.11. Ensamble del robot móvil. ....	52
Figura 5.12. Instalación del software Arduino IDE ....	53
Figura 5.13. Ejemplo de librerías respectivas a los dispositivos. ....	53



Figura 5.14. Ejemplo de la inclusión de librerías previamente instaladas. ....	54
Figura 5.15. Código fuente para la obtención de la distancia. ....	55
Figura 5.16. Representación de acciones para la evasión de obstáculos. ....	56
Figura 5.17. Comandos de configuración del módulo de Reconocimiento de voz. ....	57
Figura 5.18. Valores de fabrica del módulo/ Valores asignados. ....	61
Figura 5.19 Vista de diseño de la aplicación móvil. ....	62
Figura 5.20. Vista por bloques. ....	62
Figura 5.21. a) Vista inicial de la aplicación.    b) Representación de bloques. ....	63
Figura 5.22. Instrucciones del botón conexión. ....	64
Figura 5.23. Instrucciones para el botón hablar ....	64
Figura 5.24. Condiciones del funcionamiento del botón Enviar instrucciones.....	65
Figura 5.25. Cambiar método de comunicación ....	66
Figura 5.26. Vista previa de la aplicación. ....	67
Figura 5.27. Construir aplicación y opciones de guardado. ....	67
Figura 6.1. Instalación de la aplicación Mi robot móvil en Android. ....	69
Figura 6.2. Fin de la instalación y ejecución de la aplicación.....	69
Figura 6.3. Funcionamiento total de la aplicación: Mi robot móvil.....	70
Figura 6.4. Prueba de reconocimiento de palabras. ....	71
Figura 6.5. Alerta de detección de obstáculos.....	71
Figura 6.6. Búsqueda de dispositivos Bluetooth disponibles.....	73
Figura 6.7. Acción de botón Enviar instrucción. ....	73
Figura 6.8. Reconocimiento de voz fallido. ....	74
Figura 6.9. Recepción de caracteres enviados desde la aplicación. ....	75
Figura 6.10. Cambio de comunicación remota a local y viceversa.....	75
Figura 6.11. Cambio al método local, recepción de datos y regreso a la comunicación remota.....	76
Figura 6.12. Funcionamiento inicial del robot móvil: Desplazamiento y evasión de obstáculos.....	77
Figura 6.13. Recorrido de trayectoria sin / con obstáculos por el método remoto. ....	79
Figura 6.14. Recorrido de trayectoria sin / con obstáculos por el método local. ....	79
Figura 6.15. Recorrido de trayectoria sin / con obstáculos por el método dual.....	80

Figura 6.16. Porcentaje de aciertos en pruebas de reconocimiento de comandos de voz.  
.....81

Figura 6.17. Número de aciertos en reconocimiento de comandos a distancia para cada  
uno de los métodos de comunicación. ....83

# 1 INTRODUCCIÓN

Hasta la fecha, los avances relacionados con el área de la robótica han despertado un creciente interés, debido a que cada vez se vuelve más común el desarrollo de máquinas o dispositivos que cuentan con un sistema de movimiento autónomo, logrado a partir de la implementación de técnicas de control apoyadas por hardware y software.

Es importante mencionar que el concepto de autonomía no solo se relaciona con cuestiones energéticas, sino que también se refiere a la capacidad de percibir, modelar, planificar y actuar, para realizar tareas previamente establecidas, sin intervención del usuario o con poca participación del mismo, ya que el robot se puede desenvolver en entornos total o parcialmente conocidos (Aguilera Hernández, et al., 2017). Líneas de investigación como la visión artificial permiten a los robots distinguir objetos y planificar rutas mediante el uso de diferentes técnicas; una de ellas puede ser el uso de redes neuronales artificiales.

Para el desarrollo de este proyecto, en el apartado específico de la comunicación inalámbrica se hace uso de la tecnología Bluetooth, que de acuerdo con su cobertura se posiciona dentro de la clasificación WPAN (*Wireless Personal Area Network*, o Redes inalámbricas de área personal). Cabe considerar que, este tipo de redes no están diseñadas para sustituir a un equipo de cableado, más bien para proveer una comunicación en el espacio operativo personal.

Las redes de comunicaciones inalámbricas se encuentran en un periodo de mayor crecimiento, en comparación con las redes con cableado, sobre todo en el sector de la automatización y control. Hasta ahora existen diversos factores que favorecen este hecho, como puede ser reducción de costos en cableado, reducción de consumo energético, y sencillez de implementación (Dignani, 2011).

Con el desarrollo de este proyecto se busca hacer de la interacción entre usuario – robot, una comunicación más natural y efectiva a la que hasta ahora se presenta; por tal motivo, se planea la implementación de percepción auditiva en un robot móvil, haciendo uso del habla como un método de emisión de instrucciones.

En este trabajo de tesis se vuelve necesario llevar a cabo el diseño e integración de un robot móvil con ruedas, con la incorporación de percepción auditiva con el fin de hacer al móvil capaz de realizar acciones con respecto a movimiento, emitidas de un usuario mediante comandos de voz. Estas tareas se llevan a cabo mediante métodos de comunicación local y remoto; el primero con un módulo de reconocimiento de voz, y el segundo mediante una aplicación desarrollada para dispositivos móviles del entorno Android.

Para que un robot móvil pueda realizar satisfactoriamente las tareas para las que fue especializado, existe una condición muy importante: que conozca su entorno de trabajo, para adaptar su funcionamiento de acuerdo con el estado actual del mismo, implicando la recolección y procesamiento de información para utilizarla en el sistema de control. Esta información es generada por los sensores integrados, permitiendo controlar al vehículo con trayectorias estables; con esto se puede asegurar que el robot alcance su punto de destino con mínimos errores o sin sufrir choques que traigan como consecuencias daños en su estructura o en su entorno (Baliñas, et al., 2006).

Derivado del análisis de la literatura, se identificaron proyectos o sistemas con similitud en cuanto al método de desarrollo, pero al examinar cada uno de ellos, se obtiene como resultado que ya existen robots móviles con reconocimiento de voz pero, en estos la emisión de instrucciones es llevada a cabo mediante un módulo de reconocimiento de voz instalado en el prototipo; por lo que, se considera viable para este proyecto hacer uso de este método más la implementación de una conexión remota desde un teléfono móvil con el robot para mejorar la calidad de control de estos dispositivos, obteniendo dos alternativas de manejo. De este modo, se concluye que hasta ahora no se presenta algún caso que cuente con comunicación local y remota, haciendo uso de reconocimiento de voz para el control de sus dispositivos.

Adicionalmente, en el diseño de trabajos consultados, no se ha identificado el empleo de hardware para la evasión de obstáculos, considerándose un método factible para la integración en el presente proyecto, por lo que esta característica ayuda al recorrido del robot en su trayectoria, mejorando su autonomía.

## **2 PLANTEAMIENTO DEL PROBLEMA**

La interacción entre humanos, dispositivos y el entorno que los rodea, requiere de diferentes paradigmas de comunicación, como el reconocimiento de voz, especialmente si se trata de personas que sufren de algún problema de movilidad, pero que pueden ver, oír y hablar. Las personas con estas características, la mayoría de las veces, no pueden llevar a cabo el control completo de algunos aparatos tecnológicos modernos, como en el caso de un robot móvil, si es indispensable moverse de un lugar a otro para maniobrar el dispositivo.

### **2.1 Definición del problema**

Los avances en la robótica permiten realizar variedad de aplicaciones. Inicialmente los robots fueron construidos con fines industriales para realizar tareas repetitivas, peligrosas o para el reemplazo de actividades personales, pero hoy en día se han convertido en una nueva técnica de ayuda para personas discapacitadas; por ejemplo: (Pérez Pimentel, et al., 2013) desarrollaron una simulación robótica, donde se implementa un algoritmo de reconocimiento de voz para ayudar a personas que sufren algún problema de movilidad, a desplazarse de un lugar a otro con solo instruirle a la silla las direcciones a donde se quieren dirigir.

Se registran casos en lo que se ha llevado a cabo la implementación del reconocimiento de voz en robots móviles, pero en su mayoría el control no es a distancia, sino con el uso de módulos de reconocimiento de voz integrados en el mismo dispositivo. El problema que aún se conserva de ellos es la evasión de obstáculos.

Se requiere desarrollar un dispositivo móvil, el cual sea manipulado a través del lenguaje verbal, con esto se pretende conseguir que el móvil ejecute los movimientos que el usuario establezca mediante las siguientes palabras: Inicia, enfrente, izquierda, derecha, giro (media vuelta), reversa, detenerse. En este proyecto se planea agregar un método dual para la manipulación de los dispositivos, buscando con ello una forma de mejorar la interacción entre usuario- robot, implementando comunicación local y a distancia haciendo uso de reconocimiento de voz en ambas para maniobrar el dispositivo.

Lo que se pretende realizar con la implementación del reconocimiento de voz en un robot móvil, es mejorar la calidad de interacción existente entre personas, dispositivos y el entorno en el que se encuentran, buscando con ello una técnica más práctica de comunicación entre el usuario y los móviles, siendo un método ideal para las personas que sufren de algún tipo de problema de movilidad y no tengan que perseguir al dispositivo para su interacción.

Con la implementación de robots móviles se hace necesario la búsqueda de diferentes paradigmas de comunicación entre estos y las personas, considerando útil el reconocimiento de voz para aquellos casos en los que las personas no pueden dirigirse hasta la dirección del robot, permitiéndoles el manejo desde el lugar en que se encuentran.

Para el desarrollo de este proyecto se pretende que el reconocimiento de voz se convierta en una alternativa de manipulación para este dispositivo, haciendo referencia que, al percibir una palabra o bien una instrucción, este deberá realizar la acción indicada; en caso contrario, que no exista reconocimiento por el móvil que no se lleve a cabo ninguna acción, o continúe con una trayectoria estable.

Según la literatura, se registran casos en los cuales se ha llevado a cabo la implementación de proyectos similares, en los cuales la manipulación del robot se realiza únicamente de manera local (en el robot), por lo cual se considera importante contemplar este método de manipulación y también agregar comunicación remota por medio de un teléfono móvil; adicionalmente, la incorporación de sensores ultrasónicos instalados estratégicamente en la estructura del prototipo permitirá contar con una herramienta para la evasión de obstáculos.

## **2.2 Objetivos de investigación**

### **2.2.1 Objetivo general**

Implementar reconocimiento de voz a un robot móvil, otorgando dos métodos de comunicación con el usuario: Local y Remota; el primero; mediante la conexión de un módulo de reconocimiento de voz instalado en el prototipo y el segundo; a través de una aplicación desarrollada para comunicar el entorno Android con Arduino, a través de un

módulo Bluetooth. Además, del empleo de sensores ultrasónicos como detectores de proximidad para otorgar al móvil la capacidad de evadir obstáculos al momento de la navegación de su trayectoria, evitando así causar daños en su propia estructura.

### **2.2.2 Objetivos específicos**

- Ensamblar un mini robot móvil con ruedas: dos motorizadas estándar, una omnidireccional, incorporando un módulo Bluetooth y uno de reconocimiento de voz, así como sensores de proximidad para la evasión de obstáculos.
- Desarrollar un algoritmo para Arduino, basado en el lenguaje de programación C; que contenga una serie de instrucciones para el comportamiento y desplazamiento del dispositivo.
- Crear una aplicación del entorno Android para la emisión de instrucciones desde un teléfono móvil a Arduino, mediante el módulo Bluetooth.
- Desarrollar un sistema integrando los dos métodos de comunicación: uno con el módulo de reconocimiento de voz y el otro únicamente con la aplicación móvil para funcionamiento dual del robot.
- Validar el funcionamiento del robot móvil; que sea capaz de realizar un recorrido dentro de un área regular, de acuerdo con las instrucciones que el usuario le ordene.

## **2.3 Preguntas de investigación**

¿Cómo hacer que el móvil reconozca el lenguaje verbal?

¿Qué técnicas de Inteligencia Artificial se pueden aplicar para el desarrollo de un algoritmo para reconocimiento de voz?

¿Cuál será el margen de error tolerado existente entre ambientes con y sin ruido como, por ejemplo, en espacios en donde se presente sonido de fondo no deseado, afectando la recepción de instrucciones?

¿Cuál es el método y cómo se va a implementar para realizar la interconexión de los entornos Android -Arduino?

## **2.4 Justificación**

La robótica es un área multidisciplinaria, dentro de la cual se realiza el empleo y combinación de varias disciplinas al mismo tiempo, como lo son: electrónica, programación e inteligencia artificial, convirtiéndose en una técnica ideal para la implementación, diseño y construcción de robots o dispositivos para realizar operaciones previamente programadas.

La robótica móvil se ha transformado en un tema constante en empresas y diferentes instituciones debido a la variedad de aplicaciones y tareas que se pueden llevar a cabo por estos dispositivos. El motivo de este proyecto de investigación es con el fin de implementar un paradigma de comunicación entre humano y robot.

Con el empleo de técnicas de percepción auditiva, se busca mejorar el ambiente de comunicación entre el operador y el robot. Como beneficios resultantes se pretende que el usuario establezca el manejo de su dispositivo por medio del lenguaje hablado, además de ser ideal para uso por personas que cuentan con algún problema de movilidad y no pueden dirigirse físicamente hasta donde se encuentre el robot.

La implementación de reconocimiento de voz se considera un beneficio al momento de la comunicación entre un usuario y los robots, buscando como resultado hacer del ambiente una interacción más natural entre ambos, logrando con ello la emisión y recepción de instrucciones por medio de palabras habladas.

La aplicación principal del robot móvil que se propone está enfocada al ocio y entretenimiento debido a que, realizará movimientos autónomos a partir de las indicaciones recibidas de un individuo, contando con una técnica natural para la manipulación de este, a través del reconocimiento de voz. Además, incluye un sistema sensorial, permitiendo la percepción de objetos del entorno que puedan obstruir su paso. Todo ello programado desde la plataforma de desarrollo de software Arduino. El ocio es una de las categorías de la aplicación de dispositivos modernos y robots móviles, de suma importancia, debido a que, la mayoría de las personas de distintas edades buscan una opción de diversión y entretenimiento para sus ratos libres, siendo esta, una razón por la que el proyecto se inclina hacia este tipo de tareas.



También se contempla emplear tecnologías de comunicación inalámbrica en auge, como Bluetooth, que es ideal para una comunicación efectiva a distancia desde un dispositivo remoto para lograr la tarea que le es programada. En este caso, la navegación será controlada a través de comandos de voz y un algoritmo implementado sobre una tarjeta de desarrollo de software, comunicado de manera remota a un dispositivo móvil. El desarrollo de aplicaciones en plataformas móviles como lo son IOS y Android, cuentan con la ventaja de que el usuario realice ciertas actividades de manera rápida y sencilla, desde la comodidad de su teléfono celular. En este caso, se planea como método para manipular al robot, indicando la trayectoria para la navegación del robot.

## **2.5 Impactos**

**Científico:** La implementación del lenguaje verbal como medio de comunicación entre los robots y los humanos, pretendiendo que el robot reconozca cuando el usuario le emita una instrucción.

**Tecnológico:** Desarrollo e implementación de un mini robot móvil que responda a instrucciones otorgadas a través de comandos de voz, emitidos desde un teléfono móvil mediante una conexión a distancia.

**Social:** Potencialmente, este dispositivo puede ser de beneficio a la sociedad, ya que, aunque no se contempla que esté dirigido para uso de un grupo social en específico; podría ser ideal para ser manipulado por personas que cuentan con algún problema limitante de su movilidad corporal y por ende no cuenta con la disponibilidad de dirigirse a cualquier lugar a donde irá el robot móvil.

### **3 META DE INGENIERÍA**

Con el diseño e implementación de un robot móvil con percepción auditiva y sistema de control a distancia se busca mejorar el ambiente de interacción existente entre el humano – robot – entorno, logrando una comunicación más natural entre estos.

La meta a la que se pretende llegar es, obtener comunicación Local y Remota del usuario con el robot para lograr su manipulación mediante estos dos métodos, considerando que la primera se llevará a cabo con el uso de un módulo de reconocimiento de voz, instalado en el prototipo y la segunda, mediante una aplicación para dispositivos móviles.

El robot deberá lograr un recorrido óptimo en una trayectoria regular, indicada a través de comandos de voz por un usuario, mismo que establece el método para controlar al dispositivo; ya sea local o remoto, haciendo uso específicamente de estas palabras: Inicia, enfrente, reversa, izquierda, derecha, giro y detente.

Independientemente del tipo de comunicación a utilizar, se contará con un sensor ultrasónico instalado en la estructura del prototipo, para que al momento de presentarse con un obstáculo frente al mismo sea capaz de detenerse o girar si es necesario.

## 4 ESTADO DEL ARTE

Para la implementación de un robot móvil, manipulado a través del lenguaje verbal, se considera necesario contar con conocimiento previo acerca de conceptos básicos de la robótica, así como las características y aplicaciones más importantes en las que destacan los sistemas robotizados. A continuación, se muestra una serie de conceptos a considerar dentro del ámbito de aplicación de dicha área de la Ingeniería; posteriormente se incluye el análisis de la literatura que contempla la ejecución de proyectos similares, considerando la propuesta de algunas mejoras que se incorporan como parte del presente proyecto de investigación.

### 4.1 Robótica

La robótica es una rama de la ciencia encargada del estudio de las aplicaciones y diseño de los robots; es una rama multidisciplinaria en la cual se involucran distintos tipos de conocimientos, como lo son: electrónica, programación, informática, mecánica, entre otras, además de hacer uso de otras disciplinas derivadas de la informática, como la inteligencia y visión artificial (Cánovas, 2011).

A partir de la unión y el desarrollo de los conocimientos antes mencionados, se obtienen como resultado, dispositivos considerados como robots, diseñados para realizar: actividades peligrosas para los humanos, trabajos repetitivos, o tareas que requieren incrementar las capacidades físicas de los humanos, por mencionar algunos ejemplos.

El término robot fue empleado por primera ocasión por el escritor checo Karel Capek en 1920, siendo utilizado para referirse principalmente a dispositivos mecánicos, como: dispositivos tele operados, vehículos submarinos, exploradores terrestres, entre otros. (Suntaxi Llumiquinga, 2015). En un inicio, el autor empezaba a utilizar la palabra checa *robota*, que significa sirviente, servidumbre o trabajo forzado; posterior a ello se llevó a cabo la traducción al inglés, en la que resulto ser robot. De acuerdo con Capek esta palabra puede ser aplicada a cualquier dispositivo que cuente con cierto grado de autonomía.

El Instituto de Robótica RIA (*Robot Institute Of America*), establece en 1979 una definición oficial para el término Robot: un robot es un manipulador multifuncional

reprogramable, diseñado para mover materiales, partes, herramientas o dispositivos especializados a través de movimientos programables variables para el desarrollo de una variedad de tareas (Boada, et al., 2005) .

#### **4.1.1 Leyes de la robótica**

En el año 1950, el escritor Isaac Asimov desarrolló una obra denominada “*I Robot*” (Yo, Robot), en la cual, menciona las reglas o leyes de la robótica a las que, según el autor, debe estar comandada la inteligencia de estos dispositivos (Asimov, 1950):

1. Un robot no debe dañar a un ser humano o, por inacción, dejar que un ser humano sufra daño.
2. Un robot debe obedecer las órdenes que le son dadas por un ser humano, excepto cuando estas órdenes estén en oposición con la Primera ley.
3. Un robot debe proteger su propia existencia hasta donde esta protección no esté en conflicto con la Primera o la Segunda Ley.

En las obras de ciencia ficción del escritor Asimov, aparecían robots con lo que hacía referencia a una máquina electromecánica con capacidad de poder obrar por sí mismos, actuando de acuerdo con estas tres leyes de la robótica.

En la actualidad se vuelve indispensable retomar la importancia de estas normas, debido al desarrollo de dispositivos móviles autónomos, en casos, en los que, si se daña a una persona, no se le asume responsabilidad a la máquina, pero es más importante la seguridad y protección de la gente; por tal motivo se hace necesario que la autonomía de los robots esté ligada a ciertas medidas que evitarán que pueda causar daño a las personas o a sí mismo.

#### **4.1.2 Robótica móvil**

La robótica móvil, es una subrama de la robótica que cubre muchos campos de aplicación, como traslado de materiales, limpieza, guiado de personas, y algunas otras. Sin embargo, dentro de esta área hay problemáticas a las cuales aún no se ha encontrado alguna solución,

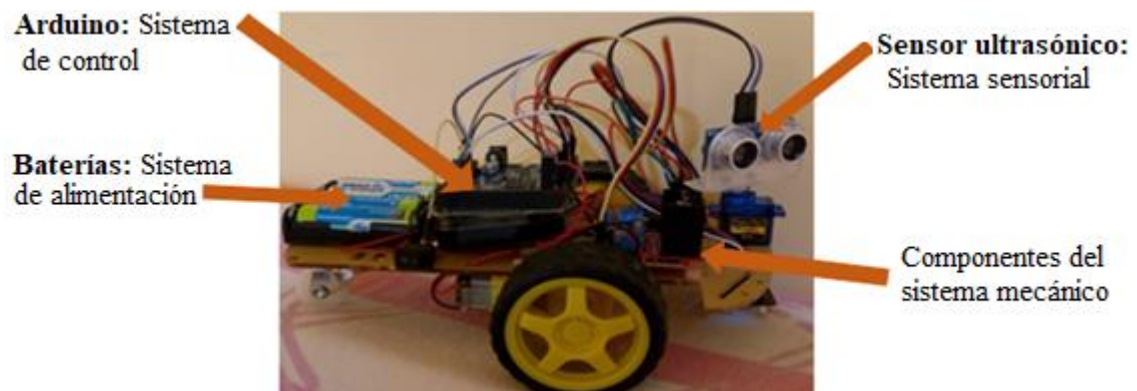
como autonomía en la toma de decisiones, interacción con su entorno o con otros robots, seguridad al usuario y eficiencia en la comunicación (Suntaxi Llumiquinga, 2015).

Se considera un robot móvil a un sistema electromecánico capaz de desplazarse, ya sea guiado o de manera autónoma, sin necesidad de estar sujeto físicamente a un solo punto. Una de sus principales características es la posesión de sensores, los cuales le permiten realizar el monitoreo a cada momento de su posición, desde un inicio o punto de origen hasta su destino (Barrientos Sotelo, et al., 2007).

Un robot móvil autónomo cuenta con capacidad de percepción del entorno (sensores); con ayuda de estas herramientas, se vuelve capaz de percibir objetos del entorno o incluso construir una representación de este. También tiene características que los identifican, como lo son: maniobrabilidad, estabilidad, capacidad de tracción, control y eficiencia.

Por consiguiente, es importante considerar que un robot operacional está constituido por cuatro sistemas relacionados entre sí, tal como se aprecia en la Figura 4.1 (Reyes, 2011):

1. Sistema mecánico: Permite el movimiento, utilizando diferentes elementos de acuerdo con el propósito del robot y ubicados estratégicamente.
2. Sistema sensorial: Proporciona información importante acerca de la ubicación de los elementos u obstáculos.
3. Sistema de control: Es el cerebro del robot, encargado de llevar a cabo la interpretación, procesamiento y tratado de la información.
4. Sistema de alimentación: Encargado de brindar la energía a cada uno de los componentes del robot.



*Figura 4.1. Sistemas necesarios para el funcionamiento de un robot móvil.*

La robótica móvil es un área de la tecnología reconocida principalmente por su empleo para la solución de problemas complejos. Los dispositivos que resultan como producto de esta área se constituyen en aplicaciones de programación, áreas de control, inteligencia artificial, percepción e instrumentación. La robótica se encarga del estudio de sistemas controlados, capaces de interactuar con su ambiente para llevar a cabo una tarea específica. De acuerdo con su diseño, cada robot es desarrollado para ejecutar ciertas funciones predeterminadas y son considerados en algunas aplicaciones, como (Bermudez, 2015):

- Operación en ambientes hostiles, por ejemplo, en refinerías de petróleo, plantas químicas o nucleares.
- Inspección de tuberías terrestres y submarinas.
- Exploración submarina.
- Reconocimiento de ambientes a donde el humano no puede llegar.
- Labores de construcción que impliquen mayor esfuerzo físico.
- Transporte de materiales en actividades de producción industrial.
- Sistemas dotados con sensores para evitar obstáculos y realizar desplazamientos seguros para personas discapacitadas.

#### **4.1.3 Clasificación de robots móviles**

Los robots móviles, realizan su desplazamiento por control de mandos o al guiarse con información percibida del entorno mediante los sensores, además de presentan un mayor grado de libertad, utilidad y agilidad que un dispositivo fijo, esta característica es considerada por la facilidad de movimiento en un espacio físico. Los sensores y actuadores son módulos de comunicación con el hardware del robot; si se necesita acceder a datos de los sensores o enviar comandos a los actuadores lo hacen a través de ellos (Ramírez, 2017).

Los robots móviles terrestres se clasifican según su estructura y la configuración de sus elementos, considerando que estos dispositivos cuentan con capacidad para desplazarse debido a su sistema locomotor, el cual puede tener multitud de sistemas de locomoción de acuerdo con la superficie en la cual se vaya a mover, ya que en función de esta se pueden adaptar ruedas, orugas, patas articuladas, o mecanismos híbridos (Torres, 2009).

- **Robot móvil con locomoción por ruedas**

La locomoción por ruedas es una de las más comunes y estables, que tiene características como estabilidad, debido a que siempre se encuentran en el piso, la geometría de las ruedas depende del tipo de superficie por donde se van a desplazar. En el caso del presente trabajo de tesis, el sistema de locomoción empleado es por ruedas y una configuración de 3 o triciclo.

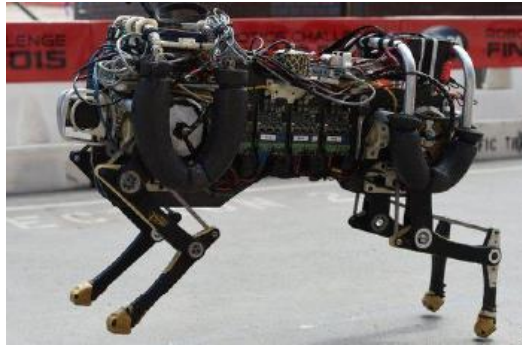
Dentro de esta clasificación se pueden encontrar varias configuraciones, dependiendo de la posición y el número de ruedas del móvil. Para hacer mención en primer lugar se encuentra la configuración Ackerman, que consta de dos ruedas traseras con tracción y dos ruedas delanteras de dirección; el otro tipo de configuración es de tres ruedas o también conocida como triciclo, que consta de dos ruedas motorizadas más una omnidireccional, o rueda loca, la cual no depende de un motor, debido a que gira libremente de acuerdo con la velocidad del robot (Andés, 2011).



*Figura 4.2. Locomoción por ruedas (Barrientos Sotelo, et al., 2007).*

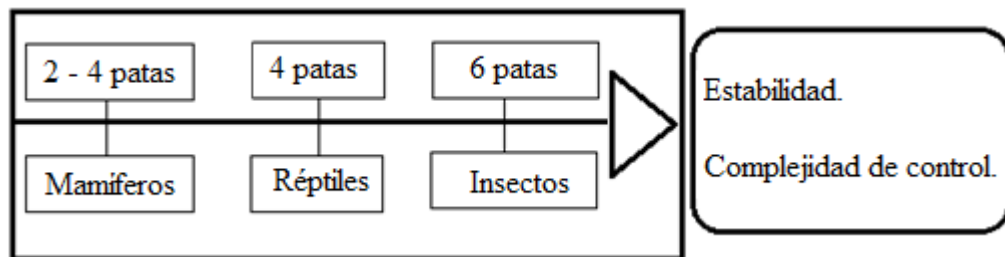
- **Robot móvil con locomoción por patas**

Dentro de la clasificación de los robots móviles que emplean locomoción por patas, hay diversos tipos, algunos de ellos, denominados bípedos que tratan de asemejar los movimientos realizados durante el caminar de un humano y otros denominados zoomórficos que en su lugar tratan de imitar a la biología de algunos animales como los robots chita de DARPA (*Defense Advanced Research Projects Agency*, Agencia de Proyectos de Investigación Avanzada de la Defensa) que se aprecia en la Figura 4.3 (Ramírez, 2017).



*Figura 4.3. Robot chita de DARPA (Ramírez, 2017).*

La locomoción por patas, en su mayoría, está basada en la imitación de movimientos realizados por seres biológicos, ya sean humanos o animales. A continuación, se muestra la Figura 4.4, donde se hace una analogía de los seres biológicos con los sistemas empleados para la locomoción con patas.



*Figura 4.4. Analogía de seres biológicos y robot con patas*

- **Robot móvil con locomoción por orugas**

El sistema de locomoción por de orugas, se considera como una alternativa a las ruedas y patas, al tratarse de lugares compactos o superficies irregulares, haciendo uso de pistas de deslizamiento y otorgando un área mayor de contacto con el suelo. Estas características hacen un sistema con mejor tracción que las ruedas y más estabilidad que las patas (González, et al., 2015).





*Figura 4.5. Locomoción por orugas (Barrientos Sotelo, et al., 2007).*

## **4.2 Comunicación inalámbrica para control de robots móviles**

Los avances tecnológicos han resultado un beneficio para la sociedad, al integrar nuevas herramientas para la solución de problemáticas y facilitar la realización de labores del día a día de las personas; sin embargo, aún existen necesidades de comunicación entre estas y la tecnología.

La comunicación inalámbrica consiste en que el emisor y receptor de información no estén unidos mediante cables o algún medio físico, haciendo uso de ondas electromagnéticas a través del espacio. A continuación, se detallan los principales protocolos de comunicación inalámbrica empleados en el ámbito de la robótica (Andés, 2011).

El avance tecnológico, junto con el crecimiento de estas necesidades ha dado lugar a que surjan tecnologías de la información con mayor eficiencia, desarrollando protocolos de comunicación inalámbrica como Bluetooth, Wi-fi y ZigBee.

- **Bluetooth**

Bluetooth, tecnología de red inalámbrica de área personal, que permite transmisión y recepción de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia. Los principales objetivos de este sistema son: eliminación de cableado, comunicación entre dispositivos móviles y fijos, creación de redes pequeñas, facilitar sincronización de datos entre diferentes equipos personales. La cobertura de alcance de Bluetooth es a una distancia máxima de 10 metros y transmite en la banda de frecuencia de los 2.4 GHz (Obregón, 2013).

- **Wi-fi**

La tecnología Wi-fi (*Wireless Fidelity*, Fidelidad Inalámbrica), en la actualidad es una de las más empleadas al momento de la creación de redes de área local. Es una tecnología que cumple con un rango de alcance dentro de un radio de 20 a 50 metros (Fields, 2015). Wi-fi está dentro de los estándares IEEE (*Institute of Electrical and Electronics Engineers*, Instituto de Ingeniería Eléctrica y Electrónica), 802.11 relacionados a las redes inalámbricas de área local (IEEE, 2019) .

Para establecer una red inalámbrica de este tipo es necesario contar con una serie de dispositivos: para iniciar, un punto de acceso, que es el encargado de ofrecer cobertura a la red, este dispositivo también funciona como un controlador central de la comunicación, a partir de este se necesita un adaptador de red inalámbrica para conectar otro dispositivo al punto de acceso.

- **ZigBee**

ZigBee, es un estándar que define un conjunto de protocolos para el armado de redes inalámbricas de corta distancia y de baja velocidad en cuestión de datos. Apoyándose en un estándar internacional IEEE 802.15.4 en las primeras capas del modelo OSI (*Open Systems Interconnection*, Interconexión de Sistemas Abiertos): física y enlace de datos. Transmite en las bandas de frecuencia, entre 868 MHz y 2.4 GHz (López, 2015).

### **4.3 Microcontrolador Arduino**

Arduino, es una plataforma para el desarrollo de software y hardware libre, compuesta a partir de una tarjeta electrónica, un microcontrolador reprogramable y un conjunto de terminales de entradas/salidas analógicas y digitales que permiten la conexión y comunicación del microcontrolador con distintos elementos como, sensores, actuadores u otros que conforman un sistema electrónico. El Arduino ejecuta una serie de instrucciones escritas por medio de código basado en el lenguaje de programación C, y grabadas desde el software IDE o Entorno de Desarrollo Integrado de Arduino (Carretero, 2016).

Existe una gran variedad de entornos para la aplicación de este microcontrolador como, (Mangado, 2017): robótica, domótica, automatización, electrónica, adquisición de datos, entre algunos otros. Arduino, es integrado en el desarrollo de prototipos autónomos interactivos de distintas áreas, por ejemplo, para adquirir información del entorno a través sensores y establecer acciones a los actuadores, como en el caso del robot desarrollado en este proyecto de tesis; el movimiento de los motores se realiza de acuerdo con la información adquirida por medio del sensor ultrasónico.

Los dispositivos Arduino, así como diversos dispositivos industriales y comerciales, incluyen un puerto serial denominado UART (*Universal Asynchronous Receiver-Transmitter*, Receptor Asíncrono-Transmisor Universal), que permite la comunicación de datos utilizando únicamente dos terminales. En la plataforma Arduino, el puerto UART se encuentra configurado en las terminales 0 y 1, identificados como Rx y Tx, haciendo referencia a la transmisión y recepción de datos en serie. De igual manera, estos dos terminales son empleados para la comunicación del Arduino con la computadora mediante un puerto USB (*Universal Serial Bus*, Bus Serie Universal); por ello, es importante mencionar que mientras se encuentre conectada la placa a través del cable la computadora, estos terminales no pueden ser utilizados como entrada o salida digital (Gonzalez Vidal, 2013).

La tarjeta Arduino MEGA, a diferencia de otros modelos, cuenta con cuatro puertos seriales; el que incluye en las terminales 0 y 1, denominado Serial0, más tres adicionales que son, Serial1, Serial2 y Serial3, que pueden ser ocupados cuando la placa se encuentre conectada con una computadora. Para disponer de más puertos seriales, es indispensable el empleo de la librería “*SoftwareSerial.h*”, que permite establecer otras terminales digitales como transmisión y/o recepción de datos. Cuando se conectan dos dispositivos mediante comunicación serial, la terminal Tx correspondiente al primer dispositivo debe ser conectado con la terminal Rx del segundo, y viceversa; esto porque mientras uno transmite, el otro recibe (ARDUINO.cl, 2015).

## **4.4 Reconocimiento de voz**

Un sistema de reconocimiento de voz consiste en el empleo de inteligencia artificial para establecer comunicación entre un usuario y un dispositivo, mediante palabras habladas. Para ello es necesario la implementación de algoritmos para convertir una señal analógica de voz a una secuencia de palabras representada en forma de texto. El objetivo principal de dichos algoritmos es, la identificación de sonidos y la extracción de características, para que el sistema reconozca cuando un individuo le dice una palabra y sea capaz de responder con una acción correspondiente (Hernández, 2016).

Hoy en día el uso del reconocimiento de voz es común; con la integración del lenguaje humano en las máquinas, se ha logrado facilitar la interacción entre el hombre y estas mismas. Resulta útil mencionar que, en el reconocimiento de voz existe la siguiente categoría (Cardenas, 2018):

1. Reconocimiento de propósito específico: Las palabras a reconocer pertenecen a un vocabulario restringido; solo una serie de comandos son identificados.
2. Reconocimiento de propósito general: Las palabras pertenecen a un vocabulario general; los comandos no son clasificados para ser reconocidos.

Al conocer esta categoría, es importante analizar el método para realizar el reconocimiento de voz, tanto en la aplicación móvil como, en el módulo empleado.

### **4.4.1 Reconocimiento de voz de la aplicación “Mi robot móvil”**

Para el desarrollo de la aplicación se integró el reconocimiento de voz de Google categorizado como, reconocedor de propósito general. El método para realizar el proceso de reconocimiento consiste en el almacenamiento y procesamiento de datos, con uso de técnicas de inteligencia artificial como, redes neuronales profundas más el uso del modelo oculto de Márkov, que se describen en seguida, (Shmyrev, 2016).

Una red neuronal es una técnica de inteligencia artificial, utilizada en la solución de problemas complejos, a través sistemas computacionales inspirados en el procesamiento de información del cerebro humano. Compuesta a partir de un conjunto de neuronas artificiales que emulan la forma y función del sistema nervioso de los seres humanos.

Algunas de sus aplicaciones son: Toma de decisiones, reconocimiento y clasificación de patrones (Tepán, 2013).

Las redes neuronales profundas (*Deep Neural Networks*), son una alternativa del aprendizaje y procesamiento automático, se usan en el reconocimiento de voz, para la extracción de características, usando una red neuronal con una gran cantidad de capas ocultas entrenadas cada vez más para obtener mejores resultados (Costumero, 2016).

El modelo oculto de Márkov (*Hidden Markov model*), hace referencia a un modelo de probabilidad estadística para representar una serie de datos temporales desconocidos, como, por ejemplo, la voz. Se utilizan para comparar patrones, clasificando datos previamente pronunciados. Funcionan como base en el reconocimiento de voz (Lucas, 2017).

#### **4.4.2 Reconocimiento de voz del módulo Elechouse v3.**

El método de reconocimiento del módulo empleado para el desarrollo de este proyecto pertenece a la categoría de propósito específico, debido a que, únicamente reconoce un grupo restringido de palabras emitidas y grabadas previamente, por un usuario. Para identificar los comandos de voz se realiza un proceso que consiste en la captura e interpretación de estos, tomando y almacenando una muestra de voz para después compararla dato a dato con cada una de las palabras guardadas en su memoria de almacenamiento (Cardenas, 2018).

### **4.5 Trabajos reportados en la literatura**

Durante la etapa de desarrollo del presente trabajo de Tesis, se considera indispensable realizar la indagación acerca de trabajos realizados previamente reportados en la literatura, con cierta similitud al proyecto aquí establecido, ya sean, en cuestión de la función que desempeñan o el método empleado para llevar a cabo dicho trabajo.

Con la búsqueda y análisis realizado sobre la literatura de trabajos relacionados al tema de investigación, se han encontrado los siguientes acontecimientos acerca del problema a

tratar, mencionando que el algunos de estos se reporta únicamente el diseño de prototipos de software o aplicaciones para el control de dispositivos.

(Moumtadi, et al., 2014) realizan el diseño de una aplicación desarrollada en Java MicroEdition, para un teléfono móvil su principal objetivo es la activación de funciones en algunos dispositivos de un hogar; que tiene como ventaja la conexión a una red inalámbrica Konnex-KNX (estándar abierto mundial para control de casas y edificios) para manipular los dispositivos conectados a la misma. Para este sistema, se empleó un método cliente-servidor. El dispositivo cliente es un teléfono móvil, al que se le instaló la aplicación; mientras que el servidor es una computadora personal, brindando el servicio Web. En dicha computadora se cuenta con el reconocimiento de palabras y también con una interfaz, que le permite integrarse a la red del edificio.

En la etapa de reconocimiento de voz, desarrollada en Matlab, existe una división de funciones en los siguientes dos apartados.

1. Etapa de entrenamiento: Obtención de las características acústicas de diferentes señales de audio, que serán usadas posteriormente para el reconocimiento por el sistema.
2. Etapa de reconocimiento: Procesamiento de señales para obtener la caracterización de comandos de voz. Este apartado, es realizado mediante la extracción de características acústicas, por lo que resulta necesario el uso del algoritmo de clasificación de patrones, denominado: Modelos ocultos de Márkov; empleado para llevar a cabo el procesamiento de señales de voz. Además de la implementación de una función, con la finalidad de que, si se pronuncian palabras que no estén dentro del vocabulario establecido, no sean reconocidas como comandos de instrucción para el sistema.

La etapa de experimentación se llevó a cabo haciendo el uso de una maqueta KNX (representación física y a escala de una red inalámbrica, que permite el manejo de sistemas domóticos, en este caso mediante comandos de voz), en la que se incluyen un ventilador y dos luces, teniendo tres opciones que controlar. Posteriormente, las pruebas consisten en repetir 10 veces el mismo comando por 10 personas, para así obtener el porcentaje de error en el sistema. De acuerdo con los resultados obtenidos durante las pruebas, se

concluye un porcentaje de confiabilidad de 87% en el reconocimiento de voz por el servidor. Cabe mencionar que estos resultados se obtienen en espacios donde el ruido exterior es muy bajo, los usuarios tienen que hablar fuerte y claro.

(Simancas & Meléndez Pertuz, 2017), reportan el desarrollo de un algoritmo en la plataforma *Neural Network Toolbox* de Matlab para el reconocimiento de voz, con la finalidad de obtener el control de equipos de asistencia motora. Las señales de información de voz fueron tomadas por medio de la aplicación *Smart Voice Recorder* del entorno de programación Android. El *toolbox* de Matlab permite realizar el diseño, entrenamiento, visualización y simulación de redes neuronales.

El filtrado realizado mediante análisis rápido de Fourier y coeficientes cepstrales de Mel; que, indican las características de la señal de la voz, y son empleados para representar el habla mediante percepción auditiva. Dichos coeficientes son derivados de la transformada de Fourier o también de la transformada del coseno discreta. Una característica representativa de los coeficientes de Mel es que sus bandas de frecuencia son situadas logarítmicamente de acuerdo con la escala de Mel (Rincón Llorent, 2007).

A partir de la toma de muestras, para el procesamiento se entiende que en las señales se incluye ruido del entorno, convirtiéndose en un factor que se tiene que eliminar en la etapa de filtrado. Los ruidos por filtrar son los que se encuentran entre un rango de frecuencia de 300-3750 Hz, al obtener estas características se aplica un filtro pasa-banda sobre la señal inicial.

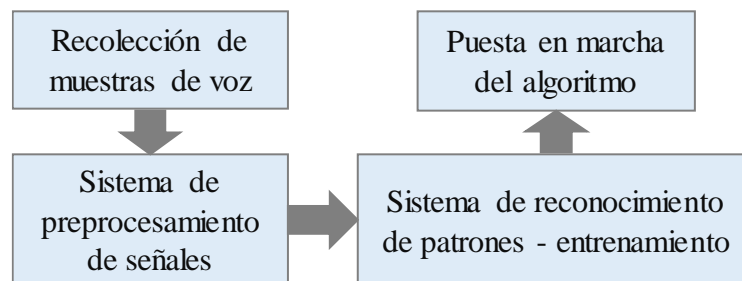
A continuación, se presentan los pasos realizados para el desarrollo del algoritmo de reconocimiento de patrones mediante comandos de voz que proponen (Simancas & Meléndez Pertuz, 2017), y en la Figura 4.6, se representa, a partir del diseño personal de un diagrama a bloques de la interpretación de su método:

1.- Recolección de muestras de voz: Tomar muestras de voz en lugares con o sin ruido, una para cada palabra del sistema para posterior a ello, puedan ser interpretados por MATLAB.

2.-Preprocesamiento de Señales: Implementación de un filtro para eliminar el ruido de las señales obtenidas en el paso número 1, mediante la transformada rápida de Fourier y análisis de los coeficientes de Mel.

3.- Sistema de reconocimiento de patrones: Desarrollo de Redes Neuronales, para entrenar el algoritmo cuando ya cuente con las muestras filtradas, para reconocer los comandos de voz.

4.-Puesta en marcha del algoritmo: Ejecución de pruebas en espacios con o sin ruido, para detectar el porcentaje de error en el algoritmo implementado y hacer comparación con otras técnicas.



*Figura 4.6. Diagrama a bloques del algoritmo de reconocimiento de voz (Simancas & Meléndez Pertuz, 2017)*

El control de los equipos se realiza mediante conexión a distancia, convirtiéndose en un punto a tomar para la implementación del presente proyecto, debido a que, la plataforma de desarrollo Arduino más la tecnología Android no se han identificado como herramientas de uso común para la obtención de señales, y de esta manera identificar la voz.

(Baquero Romero, et al., 2012), diseñan e implementan una aplicación basada en el lenguaje de programación Java para el desarrollo de funciones haciendo uso de algoritmos genéticos para realizar la caracterización y reconocimiento de comandos de voz en un mini robot tipo insecto. Emplean un algoritmo de programación dinámica DTW (*Dinamyc Time Warping*, Alineamiento Temporal Dinámico). En este proyecto se utiliza la



detección de palabras de una sola persona. Se reporta la culminación del prototipo, pero con variaciones en pruebas y resultados por falta de filtros reductores de ruido.

Para el desarrollo de esta aplicación, se consideran cuatro módulos descritos posteriormente:

1.- Obtención de la señal hablada haciendo uso de un micrófono o adjuntando una grabación de la misma computadora.

2.- Extracción de características mediante un método de LPC (*Linear Prediction Coding*, Codificación por Predicción Lineal), el cual otorga un vector descriptor de la señal de voz, equivalente a un vector de bits.

3.-Comparación de características y procesamiento de los comandos caracterizados, donde el software trabaja con el vector de números binarios equivalente a la palabra antes mencionada y las muestras guardadas para obtener el vector que tenga menor distancia o la distancia euclidiana debido a que, es la palabra para reconocer.

4.- Comunicación de las acciones de control y los actuadores del robot insecto.

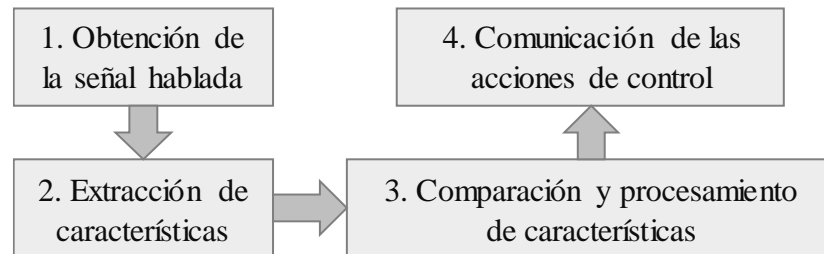
Se adjuntan la Figura 4.7 de la interfaz desarrollada en Java del proyecto que se resume, en los párrafos precedentes, donde se observan los 4 módulos explicados.



Figura 4.7. Interfaz gráfica (Baquero Romero, et al., 2012)

A partir del estudio realizado en el proyecto, se obtienen dos diagramas, uno a bloques; en la Figura 4.8, y posterior uno de flujo donde se muestra la interpretación propia del

método general planteado por los autores, para llevar a cabo la aplicación para el manejo y manipulación de un robot móvil tipo insecto por medio de reconocimiento de voz, que se muestra en la Figura 4.9.



*Figura 4.8. Diagrama a bloques de los pasos realizados para el desarrollo de la aplicación mencionada.*

La diferencia que existe entre este proyecto y el que se presenta en este documento, radica en el empleo de locomoción con patas y con ruedas, la diferencia de la plataforma de desarrollo siendo Java, este implementado sobre una FPGA (*Field Programmable Gate Array*, Arreglos de compuertas programables), en contraste con Arduino.

Además, para la comparación de las palabras también se hace uso de algoritmos genéticos, el cual no se tiene considerado. La adquisición de señales de voz, procesamiento y reconocimiento de voz se realizan a partir de la interfaz de Java mostrada en una computadora personal, en contraste a ello para las siguientes acciones será necesario hacer empleo de un módulo de reconocimiento de voz instalado en la estructura del prototipo y configurado para su funcionamiento con la plataforma Arduino.

(Cui & Xue, 2009) diseñaron un método de procesamiento de voz basado en el uso de un chip SPCE061A, en conjunto de un micrófono, bocina y circuitos programados para el procesamiento de señales con el propósito de hacer reconocimiento y permitir el acceso de personas a lugares específicos, en este sistema se trata de buscar mediante patrones de voz el control de una cerradura que permite que la puerta se abra o se bloquee el acceso.

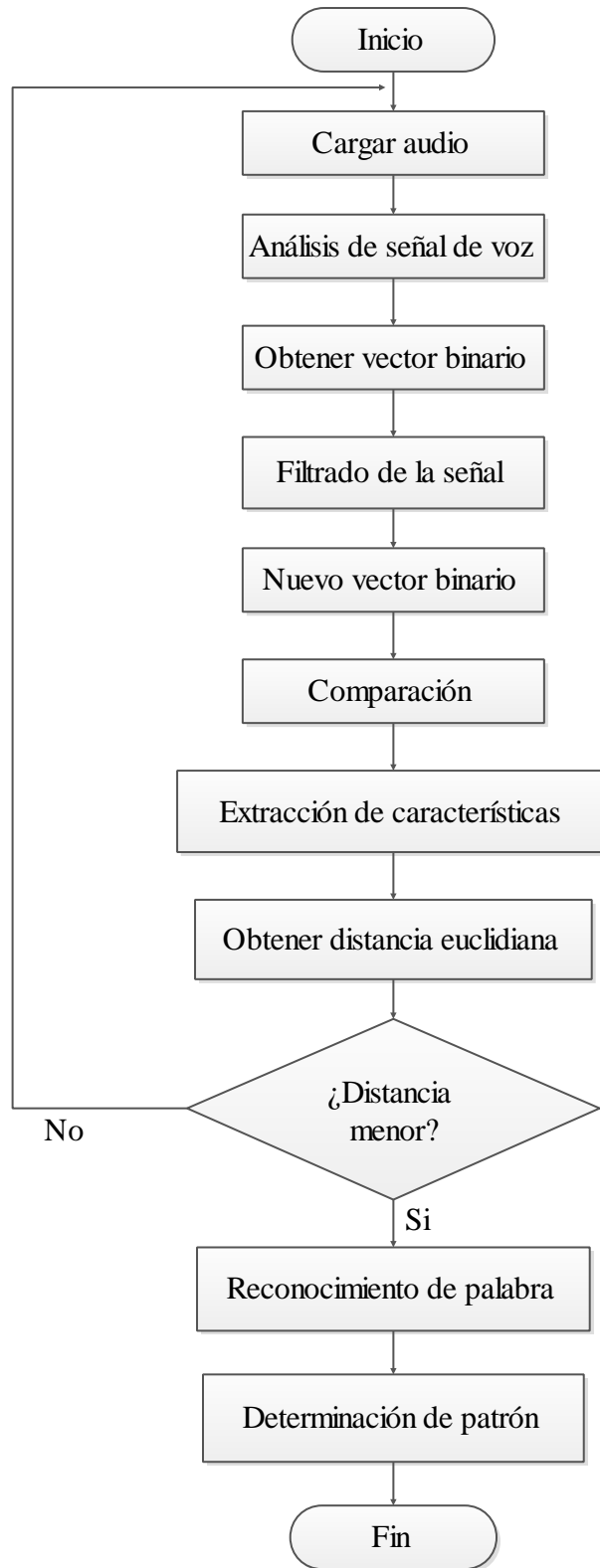


Figura 4.9. Diagrama de flujo del método para el desarrollo de la aplicación de reconocimiento de voz.

La manipulación de circuitería programada para dar acceso a un sistema mediante comandos de voz a base de un chip es la una de las diferencias que se encuentra entre este trabajo y el proyecto de investigación explicado en este documento, debido a que, se pretende llevar a cabo el control del dispositivo de dos maneras: local; con el uso de un módulo de reconocimiento de voz y remota; con una aplicación desde un teléfono móvil.

(Linares Ruiz, et al., 2004) implementaron una conexión mediante Bluetooth entre una computadora y un dispositivo electrónico programable mediante el protocolo de comunicación RS232, dentro del laboratorio de electrónica de una Universidad; incluyendo el generador de funciones y osciloscopios. Se construyó un dispositivo que, permite probar cada una de las rutinas del protocolo Bluetooth, estableciendo un enlace de comunicación del dispositivo con una computadora.

El prototipo mencionado, fue desarrollado en el entorno Matlab, en conjunto con LabVIEW. En contraste con el trabajo que se pretende desarrollar como proyecto de tesis, será efectuado en el entorno de programación de Arduino y para establecer el método de comunicación remota, se hará uso de la tecnología Bluetooth haciendo uso de sus protocolos para mantener comunicación inalámbrica entre el dispositivo de control, que será un teléfono móvil del entorno Android y el robot.

(Zhen, et al., 2010) desarrollaron un sistema de reconocimiento de voz basado en la tecnología LAN (*Local Area Network*, Red de área local) para teléfonos móviles, este sistema fue diseñado especialmente para el campo agrícola, planeado para fungir como base de búsqueda de información. Se trata de un modelo cliente-servidor donde el agricultor solicita ayuda accediendo a su aplicación y realizando alguna pregunta referente al tema del cultivo, posterior a ello recibe una respuesta, una vez que el mensaje ha sido captado por el dispositivo servidor y este realiza la consulta mediante una base de datos para establecer que va a devolver.

La conexión de este sistema se realiza a través de una línea telefónica para hacer un enlace entre el ordenador y una red pública, siendo un beneficio para la comunidad agrícola debido a que, se especifica el desarrollo de esta aplicación únicamente con el fin de realizar una base de datos para consulta de dudas con respecto a su campo y cultivo.

El desarrollo de este sistema resalta la ejemplificación del uso del reconocimiento de voz como método de control, sin embargo; el principal funcionamiento de este es como base de información, debido a que, el dispositivo que funge como servidor, ya cuenta con una base de datos, con las posibles preguntas que puedan ser realizadas con respecto al campo agrícola, y por ende también con una serie de soluciones a brindar o mandar como respuesta. A diferencia con el proyecto a desarrollar que no presenta una base de datos, sino que, incluye las instrucciones y operaciones a realizar dentro del mismo programa.

(Sanz Leon & Vera de Payer, 2015) desarrollaron una aplicación en Matlab y Simulink basada en reconocimiento de voz para un robot, para uso médico exclusivo. Creando también una GUI para la grabación de señales en tiempo real en el *Works pace* de Matlab. Empleando las funciones DTW (*Dynamic Time Warping*, Alineamiento Temporal Dinámico) y DP (*Dynamic Programming*, Programación Dinámica) respectivamente, como un método de comparación de patrones.

En este proyecto se emplea el paradigma de reconocimiento de voz, pero el prototipo se reporta como una simulación, en la que se presenta un porcentaje de error debido al ruido existencial, por lo que se considera importante para el proyecto a realizar la etapa de pruebas del algoritmo en ambientes con el menor ruido posible para un procesamiento óptimo de las señales de voz.

(Suntaxi Llumiquinga, 2015) diseñó un sistema de navegación bajo la plataforma *Microsoft Robotics Developer Studio*, en conjunto del Lenguaje VPL (*Visual Programming Language*) haciendo uso del lenguaje verbal para manipular un robot móvil. La comunicación es basada en un enlace inalámbrico entre el robot y una computadora, utilizando el protocolo de comunicación IEEE 802.15.1, al cual pertenece la tecnología Bluetooth, para la recepción y transmisión de datos.

A continuación, se presenta la secuencia de pasos del método empleado para el desarrollo de la aplicación en la plataforma VPL:

- 1.- Recibir señales por medio del servicio *Speech Recognizer*: configuración para definir palabras y frases a reconocer.

- 2.- Mando de control verbal: configuración de entrada de actividades, establecer valores de ángulo de rotación de 0° a 360°, de 30° y 45° para izquierda y derecha y 180° para media vuelta.
- 3.- Procesamiento de comandos de voz a través del servicio *GenericDifferentialDrive*: establecer valores de configuración donde se especifican las acciones que se van a realizar.
- 4.- Diseño de aplicación en *iRobot.Create.Simulation.Manifest.xml*: Permite realizar una simulación robótica establecida previamente a través de la herramienta *Visual Simulation de Microsoft Robotics Studio*.
- 5.- Implementar algoritmos de control.
- 6.- Ejecutar pruebas del control verbal sobre el modelo simulado de la plataforma móvil *iRobotCreate*.
- 7.- Inicio del programa: la primera vez se requiere el ingreso de los comandos a interpretar cuando la aplicación sea ejecutada, abrir un navegador web con un URL (*Uniform Resource Locator*, Localizador de Recursos Uniforme) en específico para establecer la configuración inicial del reconocimiento de voz, agregar las palabras y frases a reconocer, visualizar el diccionario para saber si se incluyeron y verificar la identificación de los eventos de reconocimiento de voz y las acciones a realizar al momento de la ejecución.

En la Figura 4.10, se visualiza un diagrama a bloques de la secuencia de pasos del método empleado para el desarrollo de una aplicación con reconocimiento de voz aplicada para emitir información al sistema de navegación de un robot móvil.

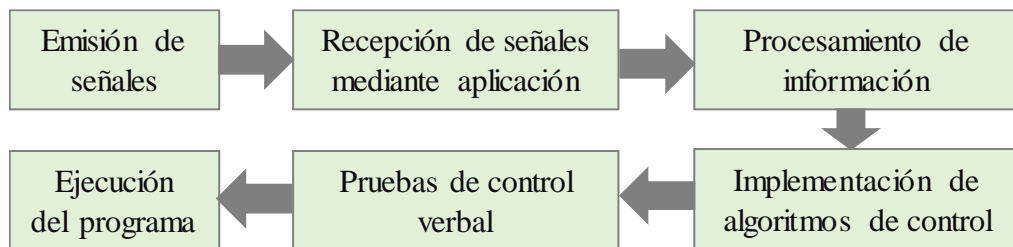


Figura 4.10. Diagrama a bloques del método para la simulación del sistema de reconocimiento de voz.

La plataforma de desarrollo de este proyecto destaca la utilización de la tecnología Bluetooth, perteneciente al protocolo IEEE 802.15.1, para realizar la comunicación inalámbrica, seguido del intercambio de datos de información entre una computadora portátil y el robot como algo efectivo; convirtiéndose en un punto a considerar como método de comunicación remota para el desarrollo del robot móvil que se pretende, estableciendo conexión mediante la tecnología inalámbrica Bluetooth entre un teléfono celular y un módulo instalado en la estructura del prototipo.

(Sung Won, et al., 2013), en el Departamento de Ingeniería Avanzada de la Universidad Kyungnam, Changwon, Corea, reportan un trabajo con similitud al desarrollado en este proyecto de tesis, en el cual se desarrolla una aplicación para la manipulación de un robot móvil mediante comandos de voz utilizados para realizar una comparación y evitar ruido con el objetivo de lograr que el robot no tenga error de navegación, el problema de la captación de voz se resuelve con la ayuda de filtros para reducir ruidos; sin embargo, tiene como ventaja que el sistema capta ordenes de diferentes personas.

El sistema consta de dos bloques principales, primero; la implementación de un micrófono inalámbrico, del cual se obtienen las palabras y segundo; un decodificador que se basa en modelos acústicos para obtener como resultado la voz sin ruido a partir del primer bloque.

El método de desarrollo de este proyecto cumple con la siguiente secuencia de pasos:

- 1.- Fuzzificación de variables de entrada y salida.
- 2.- Construcción de base a través de una regla de aprendizaje de refuerzo.
- 3.- Proceso de razonamiento del dispositivo.
- 4.- Defuzzificación de variables.

El algoritmo de navegación se realiza a partir de la obtención de un mapa local mediante una matriz de sensores ultrasónicos ensamblada como parte del prototipo, calculado los resultados por medio de una matriz de bits para hallar un punto considerable para partir, obteniendo el ángulo de rumbo del móvil.

El mapa local es de gran ayuda antes de iniciar el recorrido de la trayectoria, debido a que, permite al robot tener conocimiento del lugar por donde se desplaza, realizando también

la evasión de obstáculos, aunque persiste el error de la obtención del ángulo de rumbo, siendo un problema considerable durante el recorrido de la ruta de este vehículo.

Con el análisis correspondiente en este trabajo, se obtiene similitud en cuestión del uso de sensores de proximidad para evadir obstáculos; punto que en ninguna de las referencias anteriores se ve reflejado. El robot realiza el recorrido con la información que le es otorgada a través de las palabras, además de la que obtiene mediante el mapa local, por lo tanto, se considera la implementación de estos para mejorar la navegación del dispositivo y disminuir la presencia de posibles inconvenientes durante su recorrido. Datos importantes de la navegación del robot móvil estudiado: la velocidad máxima es de 0.52 m/s durante el desplazamiento y 1.854 rad/s de giro.

(Lima, 2011) diseña un algoritmo para la integración en un sistema automático de un robot móvil para la exploración de diferentes superficies de terrenos en lugares no aptos para personas, debido a que es especializado para llevar a cabo tareas de desminado humanitario. El principal objetivo de la implementación del sistema al robot móvil es, lograr la localización de alarmas de minas y posterior a ello obtener algunas características de la superficie del terreno explorado, como podrían ser la detección y posición de obstáculos.

Parte importante del proyecto mencionado, es que, la información recolectada a partir de la detección de los obstáculos sirve como ayuda para el mismo sistema del robot a determinar una mejor trayectoria de su recorrido. Sin embargo, en esta cuestión se muestra similitud con el presente trabajo de tesis, donde el móvil cuenta con sensores de posición o ultrasónicos para la evasión de obstáculos y con ello lograr un mejor desplazamiento de su trayectoria.

Otro módulo del sistema consiste en la identificación de alarmas de las minas, esta tarea se logra a partir de la información otorgada por un detector de metales integrado dentro del sistema de exploración del robot. Posteriormente, los datos recabados son utilizados para realizar la construcción de dos mapas, el primero incluye la representación del contorno de los obstáculos y el segundo con la evaluación del terreno.



De acuerdo a la literatura estudiada en relación con los proyectos similares en cuanto a aplicaciones y estrategias de desarrollo integradas para el reconocimiento de voz y manipulación de dispositivos, se ha determinado que es viable la implementación de un proyecto a base de Arduino, que es una plataforma de prototipos de código abierto basada en hardware y software, más un módulo Bluetooth para llevar a cabo la conexión inalámbrica del robot con un teléfono móvil y así poder realizar la recolección de las instrucciones mediante una aplicación del entorno Android que permita realizar el reconocimiento de voz.

Durante la etapa de investigación, se obtiene que existe el desarrollo de robots móviles con locomoción por ruedas y por patas, en los cuales se han incluido métodos de control mediante reconocimiento de voz de distintas formas, de manera local o de manera remota, pero no se presenta ningún caso donde se lleve a cabo la implementación de ambas para lograr el control del vehículo de una u otra forma.

Después del análisis realizado, se concluye que para mejorar este trabajo de tesis de acuerdo con los métodos empleados en trabajos antes desarrollados, reportados previamente en la literatura, es conveniente el uso de sensores ultrasónicos o de proximidad para llevar a cabo la evasión de obstáculos mientras que el móvil se encuentra en constante movimiento, permitiendo este proceso que, el robot sea capaz de realizar el recorrido de su trayectoria, desde el punto de inicio hasta el punto de destino, evitando algún choque que pueda causar daños a su propia estructura.

*Tabla 4.1. Matriz de referencias para determinar el estado del arte.*

<b>Fuente</b>	<b>Resumen</b>	<b>Metodología</b>	<b>Áreas de oportunidad</b>
Activación de funciones en edificios inteligentes utilizando comandos de voz desde dispositivos móviles. (Moumtadi, et al., 2014)	Aplicación diseñada para el uso específico de personas con problemas de movilidad. Manipulación mediante un dispositivo móvil y reconocimiento de voz. Aplicación para la activación de funciones de dispositivos electrónicos conectados a una red de un hogar.	Desarrollada en Java. En la implementación de este sistema se llevó a cabo una interfaz cliente servidor. El cliente será la persona con el teléfono móvil y el servidor el encargado de distribuir los servicios a la red. Instalación de red previa en el hogar.	Ventajas: Manipulación de dispositivos a distancia, el reconocimiento de voz no se hace mediante módulos. Desventajas: Se requiere de una red específica dentro del espacio de trabajo, por lo tanto, si la red falla, no habrá funcionamiento de ningún dispositivo.
Algoritmo de reconocimiento de comandos de voz basado en técnicas no-lineales. (Simancas & Meléndez Pertuz, 2017)	Algoritmo de reconocimiento de voz especializado para la manipulación de equipos de asistencia motora. Creación de bases de datos en las que se incluyen las palabras necesarias. Empleo de una red neuronal.	El reconocimiento de voz se hace con un módulo integrado en los dispositivos. Se ocupa una red neuronal para el entrenamiento de este sistema.	Ventaja: Apoyo de redes neuronales para el reconocimiento de las palabras. Desventaja: Dependencia de una base de datos de palabras usadas.
Reconocimiento de palabras aisladas para control de navegación de robot móvil. (Baquero Romero, et al., 2012).	Software de reconocimiento de voz para un mini robot. Desarrollado en Java, consiste en el reconocimiento de algunas palabras que le son emitidas en castellano, todas por la misma persona. No usa filtros por ello la variación en las pruebas y resultados.	Desarrollo del sistema en Java para el reconocimiento de voz se requiere pasar por el proceso de obtención de la señal hablada, y se realiza mediante un método decodificación por predicción lineal LPC más la comparación de características con el método de alineación temporal dinámica DTW.	Ventaja: No empleo de filtros para la corrección de los sonidos, puede ser considerado un buen punto para la realización del proyecto deseado al momento de las pruebas. Únicamente se reporta el prototipo del sistema.

Design and realization of an intelligent access control system based on voice recognition. (Cui & Xue, 2009)	Implementación de un sistema para restringir el acceso a personas no autorizadas a lugares específicos. Sistema de control de acceso como método de seguridad en algunos sitios.	El método principal de funcionamiento de este sistema consiste en la instalación de circuitos en una puerta más una cerradura electrónica; esta última es la que se tiene que manipular mediante palabras.	Ventajas: El sistema es capaz de dar respuestas en caso de presentar bloqueo de la entrada. Desventajas: Si se presentan fallos en la cerradura, no habrá acceso a un hogar de la persona por seguridad.
A Study on precise control of autonomous driving robot by voice recognition. (Sung Won, et al., 2013)	Sistema de control mediante comandos de voz para un robot. El sistema acepta instrucciones de diferentes personas en ambientes con ruido existencial, emitidas a través de un micrófono inalámbrico.	Los métodos utilizados para este proyecto constan en la obtención de un mapa local para el recorrido de robot. Para el reconocimiento de voz se requiere un transmisor de micrófono inalámbrico para eliminar el ruido e interferencias.	Ventajas: Empleo de filtros controladores de ruido. Manipulación del sistema aceptando instrucciones por voz de diferentes personas.
Implementación del protocolo Bluetooth para la conexión inalámbrica de dispositivos electrónicos programables. (Linares Ruiz, et al., 2004)	Conexión mediante Bluetooth entre una computadora personal y un dispositivo electrónico programable dentro del laboratorio de electrónica de una Universidad; como el generador de funciones y osciloscopios. Prototipo desarrollado en el Matlab y LabVIEW.	El método consiste prácticamente en hacer un análisis de equipos programables más utilizados en un laboratorio electrónico. Posterior a ello realizar una conexión mediante Bluetooth y comunicar estos equipos con una computadora para desde ahí poder ser manipulados.	Ventaja: La conexión mediante Bluetooth representa una manera práctica para la manipulación de los demás dispositivos a distancia. Desventaja: Solo se reporta el prototipo del sistema.
Design of voice service system for agricultural keywords recognition. (Zhen, et al., 2010)	Sistema de reconocimiento de voz para teléfonos móviles, diseñado para cuestiones agrícolas. Cuenta con búsqueda de información, generando una serie de preguntas que este será capaz de responder.	Desarrollo de una aplicación para teléfonos celulares en la cual se ha incluido una base de datos con la información necesaria para después ser entrenada para contestar una serie de preguntas agrícolas.	Ventaja: Base de información. Desventaja: Aplicación de reconocimiento de voz, enfocada únicamente para teléfonos celulares y en el proyecto que se desea se requiere hacer envío al robot.

<p>Reconocimiento de comandos de voz aplicado a sistema robótico médico. (Sanz Leon &amp; Vera de Payer, 2015)</p>	<p>Aplicación diseñada para un robot médico, haciendo uso de reconocimiento de voz para mejorar la comunicación que existe entre enfermo y el paciente. Herramienta es para uso médico exclusivo.</p>	<p>El proyecto inicio con la adquisición de señales (palabras de una sola persona) mediante un editor de audio para después incluirlo en una base de datos de entrenamiento. La grabación de señales es realizada en Matlab.</p>	<p>Ventaja: Uso exclusivo para un robot médico. Desventaja: Implementación del reconocimiento de voz en espacios sin ruidos.</p>
<p>Reconocimiento de voz usando Redes Neuronales para el control de una silla de ruedas. (Perez Pimentel, et al., 2013)</p>	<p>Aplicación de reconocimiento de patrones aplicados en el control de una silla de ruedas que permita su desplazamiento al recibir indicaciones del usuario. Sistema para el reconocimiento de voz de tiempo real para procesar la información y producir una respuesta en un determinado plazo, usando Redes Neuronales.</p>	<p>El método aplicado en el prototipo se refiere a la implementación de reconocimiento de voz para los sistemas operativos de computadoras u otros dispositivos. Se realizó empleo de prototipos en tiempo real haciendo uso de PLC y FPGA. Específicamente para el desarrollo de una red neuronal en un FPGA para el reconocimiento de voz.</p>	<p>Ventaja: Empleo de módulos de reconocimiento de voz, para la manipulación del sistema. Uso de redes neuronales para la toma de decisiones. Desventaja: Se presentan solo prototipos del proyecto mencionado.</p>
<p>Diseño e Implementación de un Sistema de navegación por voz para robots móviles con ruedas (RMR) utilizando la plataforma de desarrollo "Microsoft Robotics Developer Studio". (Suntaxi Llumiquinga, 2015)</p>	<p>Sistema de navegación para robots por comandos de voz. Desarrollo de un prototipo para llevar el control de un robot móvil mediante reconocimiento de voz. Desarrollo de un paradigma de comunicación a través del lenguaje oral considerado como un vínculo para la interacción humano - robot.</p>	<p>Para la implementación de este proyecto se realizó una simulación en el software Visual Simulation Environment de <i>Microsoft Robotics Developer Studio</i>. Para introducir el sonido a la PC se hace uso de un micrófono con auriculares conectados a la tarjeta de sonido de esta. Para la variación de ruidos se empleó la aplicación Sound Meter de Google Play.</p>	<p>Ventaja: Uso de módulos de reconocimiento de voz instalados en el dispositivo móvil, después del análisis realizado se pretende llevar a cabo el proyecto de investigación actual con módulos bluetooth. Desventaja: No se cuenta con conexión del robot a distancia.</p>

## 5 MÉTODO

Para el desarrollo del sistema de manipulación del robot móvil, es necesario tomar en cuenta factores importantes como lo son: a) la metodología de desarrollo de software empleada, en consideración de la problemática a resolver, b) los requerimientos o especificaciones de diseño con las que deberá cumplir el sistema al ser concluido y c) el diseño de ingeniería e implementación de lo ya mencionado. Dichos factores son detallados en la presente sección.

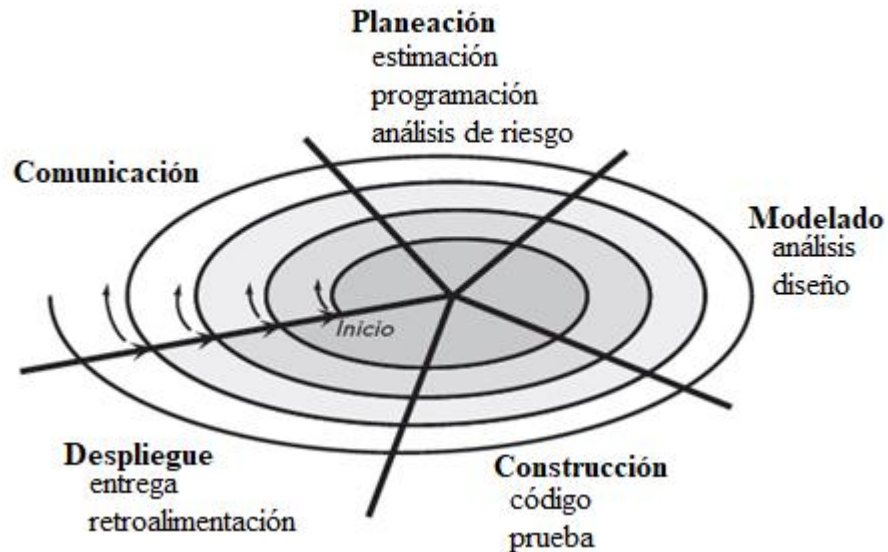
### 5.1 Estrategia para el desarrollo de software

El sistema de funcionamiento del robot móvil con percepción auditiva ha sido realizado mediante la metodología de desarrollo de software en espiral, estableciendo este método debido a que, para el desarrollo del algoritmo, es necesario llevar a cabo una secuencia de instrucciones, planificación y actividades con retroalimentación, tanto en el diseño como en la programación. Esto permite resolver algún riesgo encontrado durante la misma etapa del proceso. Por consiguiente, una vez, obtenidos el análisis y diseño, se pasó, a la construcción del algoritmo, un periodo de pruebas, retroalimentación necesaria y para finalizar con las fases de dicho modelado, la documentación del sistema.

El método de desarrollo en espiral es un modelo evolutivo del proceso del software, con el potencial para hacer un desarrollo de versiones cada vez más completas. Tiene como característica principal: el enfoque cíclico para el crecimiento incremental del grado de definición de un sistema y su implementación, mientras que disminuye su riesgo. Con el empleo de este modelo, el software se desarrolla en una serie de entregas evolutivas. Durante las primeras iteraciones, lo que se entrega puede ser un modelo o un prototipo, mientras que en las siguientes se producen versiones cada vez más completas del sistema (Pressman, 2010).

En la Figura 5.1 se muestra el diagrama común de las fases necesarias para llevar a cabo el proceso de desarrollo mediante la metodología propuesta. (Rivas, et al., 2015), señalan que el resultado del primer circuito alrededor de la espiral es la especificación del

producto; las vueltas siguientes se usan para desarrollar un prototipo y, luego, versiones cada vez más avanzadas del software. Cada paso por la región de planeación da como resultado ajustes en el plan del proyecto. El costo y la programación de actividades se ajustan con base en la retroalimentación obtenida del cliente o usuario después de la entrega. Además, el encargado del proyecto establece cuantas iteraciones se requieren para terminar el sistema.



*Figura 5.1. Fases de la metodología de desarrollo de software en espiral (Pressman, 2010).*

La etapa de desarrollo de este trabajo de tesis comenzó al definir qué tipo de sistema y prototipo se pretendían lograr, seguido de realizar un periodo de investigación y análisis acerca de los requerimientos indispensables para llevarlo a cabo; como parte fundamental, fue detallada la documentación acerca de los antecedentes y estado del arte referentes al sistema propuesto. El análisis específico sobre el problema a resolver, en este caso; la implementación de un software manipulado a través del lenguaje verbal para el control de un robot móvil, siendo capaz de reconocer la voz, procesar la información, hacer una comparación de instrucciones y posteriormente realizar las operaciones otorgadas por el usuario a través de sus palabras.

## 5.2 Requerimientos o especificaciones

En el diseño del algoritmo desarrollado para manipular el robot móvil mediante comandos de voz, principalmente se hizo uso de la etapa de planificación y análisis de la metodología en espiral explicada en el apartado anterior. Con el análisis detallado en la primera fase de este método, se obtienen como resultado las especificaciones necesarias para el sistema a desarrollar (enunciadas a continuación) y durante la planeación los resultados obtenidos son ajustes o mejoras en el plan de desarrollo del proyecto.

- Implementar la manipulación del robot móvil con ruedas, mediante comandos de voz, únicamente.
- Establecer dos métodos de comunicación del usuario con el robot: local y remota.
- El usuario debe seleccionar el método que desee utilizar.
- Las palabras por utilizar como instrucciones serán 7: inicia, enfrente, reversa, izquierda, derecha, giro y detente.
- Incluir las mismas palabras y cantidad de estas, en cualquiera de los dos métodos de comunicación.
- El móvil debe ser capaz de realizar una acción, seguido de escuchar una instrucción.
- El vehículo actuará únicamente si reconoce la palabra mencionada como una instrucción.
- Incluir en la estructura del prototipo un sensor ultrasónico para realizar la evasión de obstáculos.
- La autonomía del robot móvil dependerá de la recepción de instrucciones y la percepción de obstáculos durante la trayectoria.
- El robot debe llevar a cabo acciones de movimiento de acuerdo con una distancia considerable (15-20 cm) para la evasión de obstáculos.
- Considerar la velocidad del robot móvil para evitar algún incidente durante el recorrido de su trayectoria.
- La aplicación móvil cuenta únicamente con cuatro botones: Conectar Bluetooth, enviar instrucción, local y remoto.

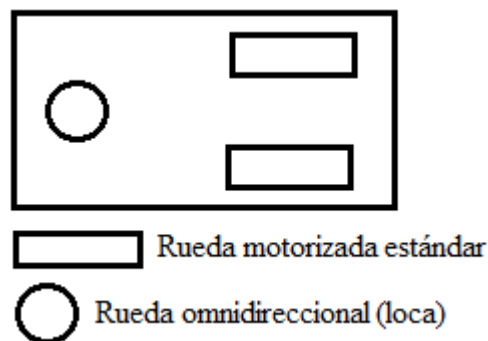
- La interfaz principal es dividida en dos secciones: Comunicación remota y cambiar método de comunicación.
- Para el control del robot desde la aplicación, es indispensable realizar en primer lugar la conexión mediante Bluetooth con el módulo que recibe las instrucciones.
- El cambio del método de comunicación a emplear se realiza desde la aplicación móvil.

### 5.3 Modelado

Para el desarrollo del algoritmo implementado en el robot móvil, fue necesario hacer un análisis, acerca de los factores o requerimientos mencionados anteriormente, para lograr el funcionamiento de dicho sistema. El primer paso fue realizar la búsqueda e investigación acerca de que software cumplía con las características necesarias para el desarrollo del algoritmo. Posterior a ello fue realizada la elección del entorno Arduino. En esta sección se presentan los componentes empleados y la función que desempeña cada uno de ellos en el conjunto del robot móvil.

#### 5.3.1 Hardware

Para dar inicio con el desarrollo de este proyecto se optó por un robot móvil con ruedas, debido a que, estos cuentan con mayor estabilidad a comparación de los que emplean locomoción por patas. En la Figura 5.2, se observa la configuración triciclo que cuenta con dos ruedas para tracción y una para la dirección.



*Figura 5.2. Representación de la configuración de tres ruedas o triciclo.*



La maniobrabilidad mide la capacidad de posicionamiento de un robot, y se define de acuerdo con la siguiente ecuación:

$$M = m + s$$

Donde:

- M = Maniobrabilidad.
- m = Grados de movilidad.
- s = Grados de direccionalidad.

En el caso del robot que se propone, cuenta dos grados de movilidad (puesto que posee dos ruedas motorizadas) y cero grados de direccionalidad (ninguna rueda permite virar por sí sola), por lo que tiene dos grados de maniobrabilidad. Esto implica que tanto el movimiento como la dirección se modifican exclusivamente mediante las dos ruedas motorizadas, modificando la velocidad angular de cada una de ellas.

A continuación, se dan a conocer los componentes que conforman el conjunto del robot móvil y la función que cada uno de estos desempeña:

Para el desarrollo del sistema de este trabajo de tesis se optó por hacer uso del microcontrolador Arduino; en específico, el *Arduino MEGA 2560*. Este es el núcleo de conexión y comunicación de todo el sistema de hardware y software que complementa al robot móvil, debido a que es el encargado del procesamiento de la información, emisión y recepción de instrucciones de los distintos dispositivos que dependen del mismo, ya sean los sensores, motores, módulos u otros.

El *Arduino MEGA 2560*, es una tarjeta de desarrollo de código abierto, que posee terminales de entradas/salidas analógicas y digitales. A continuación, se muestran las especificaciones técnicas con las que cuenta este dispositivo (ARDUINO.cl, 2015) :

- Microcontrolador: ATmega2560
- Voltaje Operativo: 5V
- Tensión de entrada: 7-12V
- Terminales digitales de Entrada/Salida: 54, (14 proveen salida PWM)
- Terminales analógicos de entrada: 16

- Velocidad del reloj: 16 MHz
- Memoria Flash: 256 KB
- EEPROM: 4 KB
- SRAM: 8 KB

Para llevar a cabo el apartado de comunicación local, el dispositivo empleado es el módulo de reconocimiento de voz *Elechouse Voice Recognition V3*, para ser instalado en el prototipo. Como su nombre lo indica, este artefacto es el encargado de realizar el reconocimiento de voz, enviando la información captada de las palabras del usuario al Arduino; en este último se realiza el trabajo de procesamiento de información y comparación de instrucciones: las palabras almacenadas previamente con los nuevos datos recibidos. Además, de ser este dispositivo el medio para grabar las palabras cuenta con una memoria para almacenarlas, independientemente sean utilizadas o no en el programa a ejecutar.

Este módulo de reconocimiento de voz cuenta con la capacidad de almacenar hasta 255 comandos de voz diferentes, pero solo se pueden utilizar 7 de ellos a la vez al ejecutar un programa implementando los comandos a utilizar, de acuerdo con las necesidades del usuario. Cabe destacar que los comandos pueden tener una duración de dos a tres segundos. El módulo tiene integrado un micrófono mono canal de 3.5 mm (PROMETEC, 2014).

En cuanto al apartado de la comunicación remota, se llevó a cabo a partir de la implementación de una aplicación móvil para entorno de Android, la cual es encargada de recibir instrucciones del usuario mediante sus palabras y posteriormente hacer envío de información correspondiente a un módulo de *Bluetooth HC-05*, encargado de la conexión y comunicación con el Arduino para procesar los datos, comparación de instrucciones y efectuar las acciones correspondientes siendo este el caso.

Otro de los componentes de hardware seleccionado para el desarrollo de este proyecto es el sensor ultrasónico con matrícula *HC-SR04*, empleado para que el móvil sea capaz de realizar la evasión de obstáculos durante el recorrido de su trayectoria. Este sensor, mide la distancia a la que se encuentre un objeto determinado, contando la demora de una señal

ultrasónica de 40 kHz del transmisor en rebotar sobre el objeto sobre el objeto y retornar al receptor; tiene dos transductores: un altavoz y un micrófono (ELECTRONILAB: Ingeniería y Diseño Electrónico, 2019).

Para el movimiento de los motores de corriente directa, el dispositivo de hardware empleado en este caso es el Driver o Puente H con matrícula *L298N*. Es un módulo controlador de corriente directa, permitiendo el control de la velocidad y dirección de estos. Las salidas A y B, del Driver se conectan a cada motor para lograr su habilitación. Además, este componente incluye un regulador de voltaje *LM7805*, que es el encargado de suministrar 5 Volts a la parte lógica del módulo (M.Castillo, 2016).

Así mismo, para realizar la programación y pruebas correspondientes del software desarrollado, se hizo uso de un equipo de cómputo, con las siguientes especificaciones:

- Sistema operativo Windows 10, 64 bits
- Procesador AMD A9, 2.9 GHz
- Memoria RAM 8 GB
- Disco duro 1 TB

En la Figura 5.3, se muestra un esquema general, representado mediante un diagrama a bloques, de la integración de componentes en el conjunto del robot móvil, donde se observa la interacción entre estos. Así mismo, en la Figura 5.4, se aprecia el diagrama del sistema electrónico, creado durante el desarrollo de la parte física del presente proyecto de tesis, donde también se observa cada uno de los componentes que ejecutan diferentes acciones, pero, en conjunto posibilitan el correcto funcionamiento de robot móvil. Cabe mencionar que entre los componentes del robot se incluyen dos fuentes de energía independientes: las baterías; que alimentan el sistema mecánico y el banco; encargado del sistema lógico del móvil. De igual manera se agregan dos bloques de comandos de voz, haciendo referencia, uno a la comunicación remota y el otro a la local.

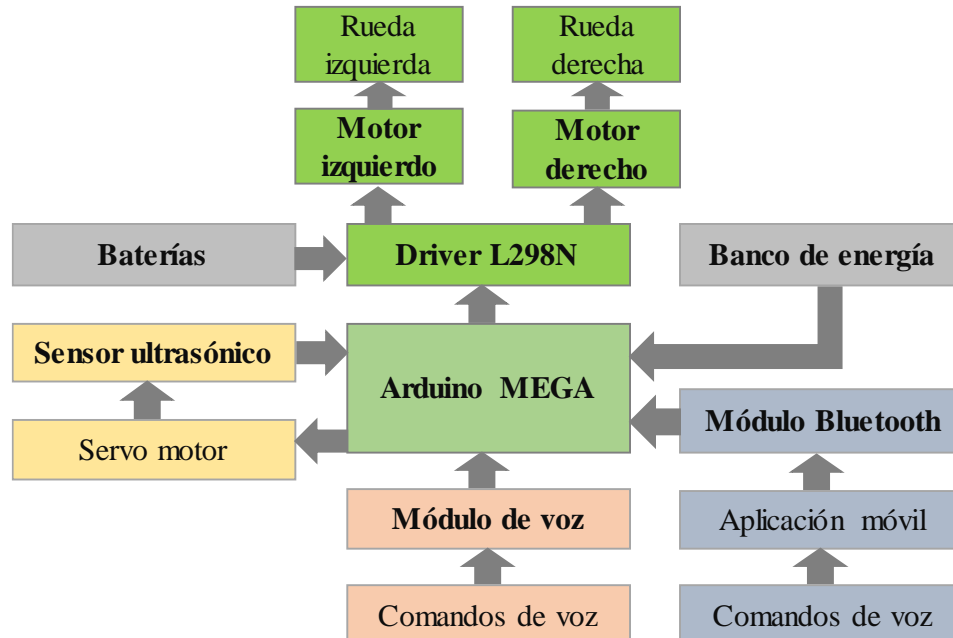


Figura 5.3. Diagrama a bloques de la estructura del robot móvil.

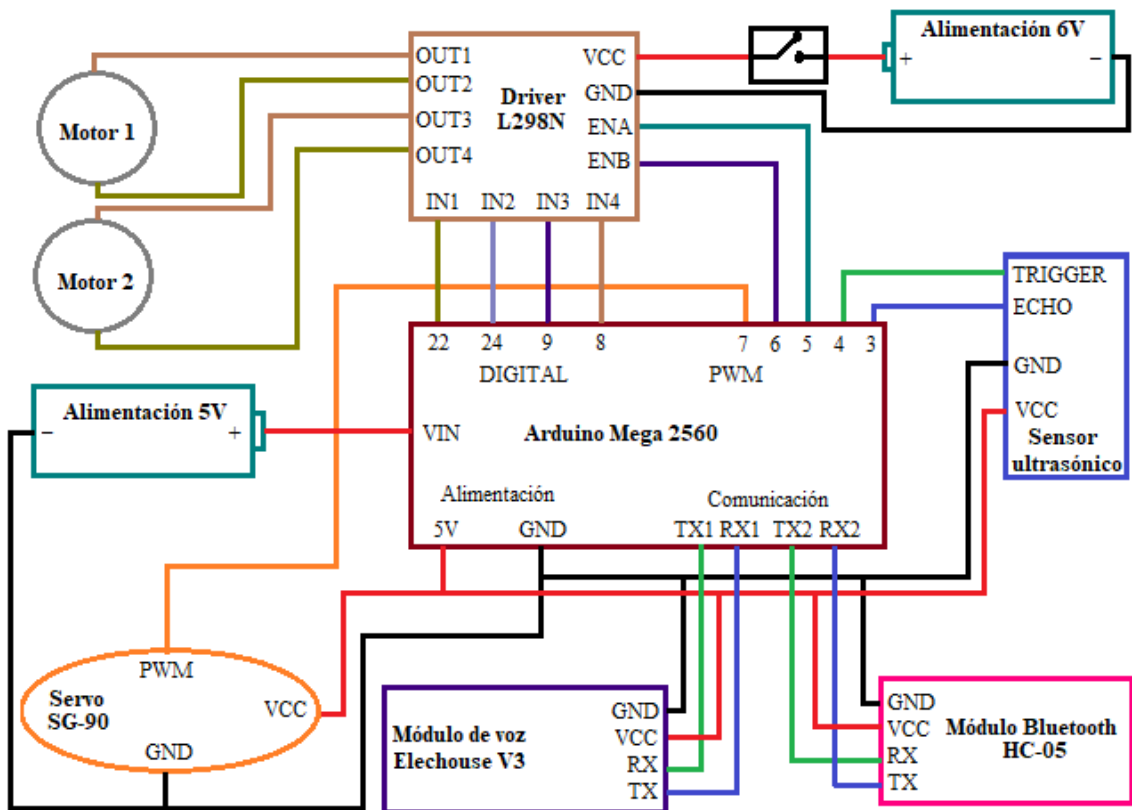


Figura 5.4. Diagrama de conexión del sistema electrónico.

### 5.3.2 Software

- Arduino

Durante la etapa de diseño y programación del algoritmo para realizar la comunicación entre los dispositivos empleados, procesamiento de información recibida y comparación de instrucciones, se optó por el entorno de programación Arduino, tomando en cuenta los requerimientos necesarios para el desarrollo del sistema, y también porque es un ambiente en el cual estos dispositivos pueden interactuar de la mejor manera debido a que, Arduino, es una plataforma de código abierto; que cuenta con librerías respectivas de los dispositivos asociados e instrucciones correspondientes, todo de forma gratuita.

Convirtiéndose en una opción conveniente para el desarrollo de este proyecto, donde se hace empleo de las librerías necesarias para que los dispositivos como el módulo Bluetooth, el de reconocimiento de voz, y el sensor ultrasónico puedan ser utilizados. Además, Arduino puede utilizarse en el desarrollo de proyectos interactivos autónomos o puede comunicarse a una computadora a través del puerto serial como lo fue en el caso de los módulos durante su etapa de configuración. En la Figura 5.5 se presenta como punto de partida en diagrama de casos de uso que representa la operación de los diferentes componentes del sistema que conforman al robot.

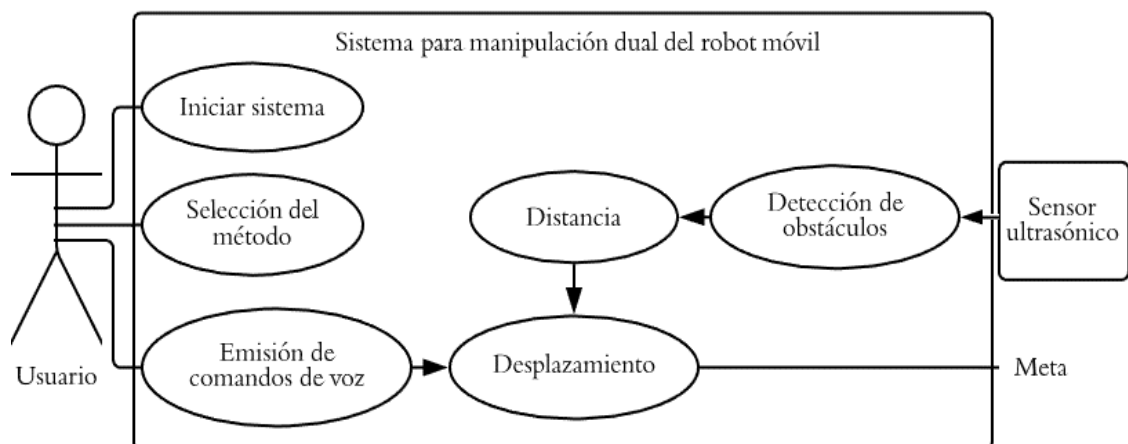


Figura 5.5. Diagrama de casos de uso del sistema para manipulación del robot móvil.

Al conocer las especificaciones de hardware y software del microcontrolador Arduino, resulta útil mencionar que éste es el encargado de procesar la información correspondiente al algoritmo desarrollado en este proyecto. Además, funge como el punto de conexión y comunicación entre cada uno de los elementos que conforman el ensamble del prototipo.

A partir de los requerimientos abordados con anterioridad, en la Figura 5.6 y Figura 5.7 se presentan los diagramas de flujo del funcionamiento del robot móvil; detallado en la sección de resultados, considerando los dos métodos de comunicación como posibles opciones, debido a que, es el usuario quien define el tipo de comunicación que desea para manejar el vehículo durante un cierto tiempo. Así mismo, en la Figura 5.8 se detalla el diagrama de flujo representativo del algoritmo desarrollado para que el robot móvil realice el desplazamiento en una trayectoria, siendo capaz de llevar a cabo la evasión de obstáculos en caso de que estos pudiesen presentarse. Para ello, es necesario tomar en cuenta una distancia considerable, en este caso de 15-20 cm, que es la que indica en qué momento el sensor detecta un objeto para posteriormente accionar el giro del móvil de acuerdo con la posición de este, ya sea adelante, izquierda o derecha.

La selección de la forma de comunicación es realizada por el usuario desde los botones pertenecientes a la interfaz de la aplicación móvil, ya que cuenta con un apartado específico para dicha función. Sin embargo; cabe mencionar que la cantidad y palabras establecidas como instrucciones es la misma para ambos métodos de comunicación, al igual que la evasión de obstáculos se lleva a cabo de la misma manera.

Además, se muestra un diagrama de casos de uso, representativo de la interacción entre el usuario y el sistema que conforma el núcleo del móvil. Donde; el usuario es el responsable de indicar la trayectoria, inicializar el sistema, seleccionar el método de comunicación y realizar la emisión de instrucciones por medio de comandos de voz hacia el robot. Independiente a la forma de manipulación, el móvil, tras recibir una instrucción realiza el desplazamiento indicado por el usuario, pero, el sensor ultrasónico es el encargado de llevar a cabo la detección de objetos que se encuentran de frente al vehículo y a partir de ello, obtener las distancias correspondientes, de la izquierda y de la derecha para establecer si se ejecuta la acción o si es necesario interrumpirla girando hacia la dirección que presenta mayor distancia con el fin de evadir los obstáculos presentes.

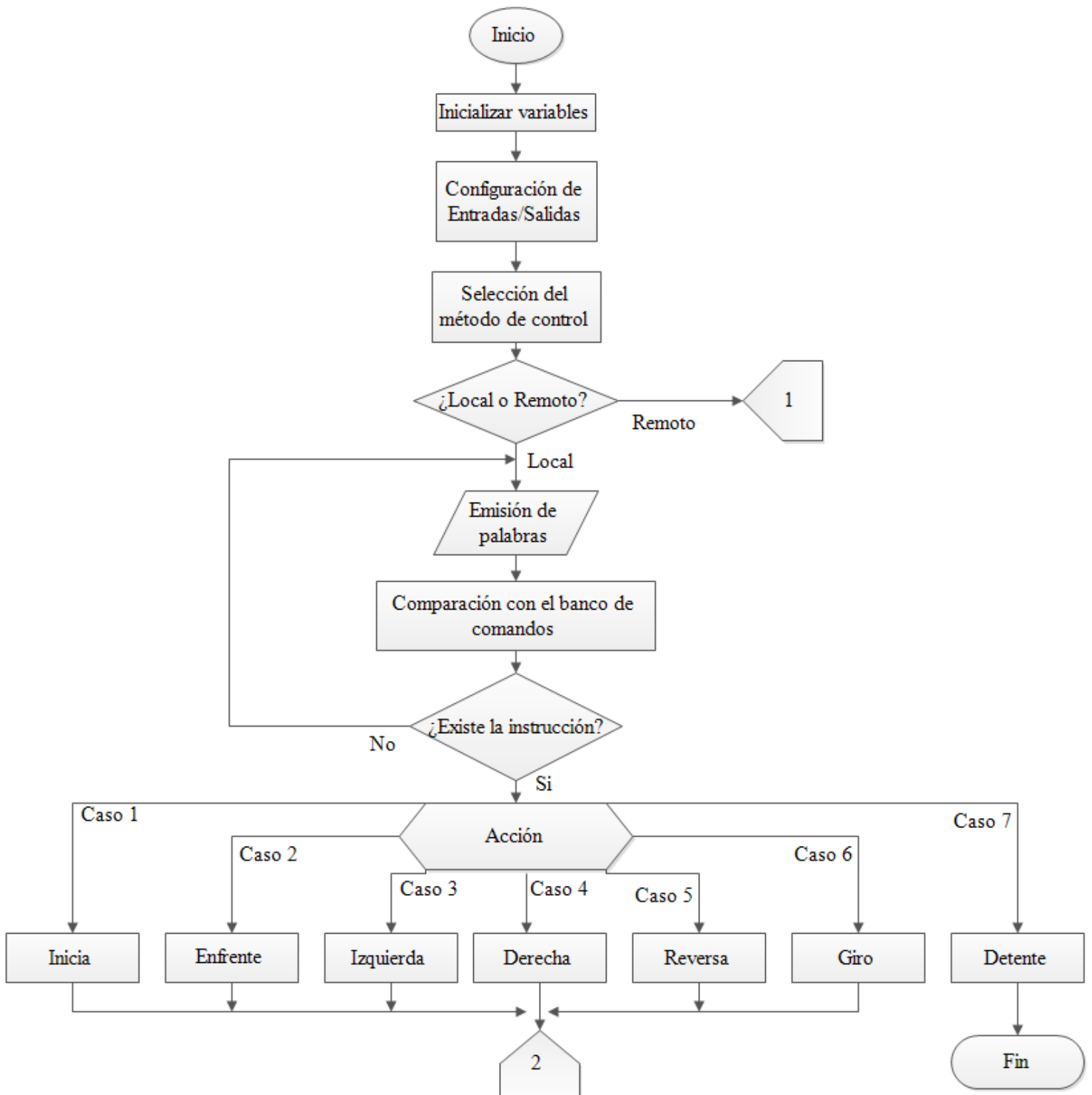


Figura 5.6. Diagrama de flujo del funcionamiento del robot móvil.

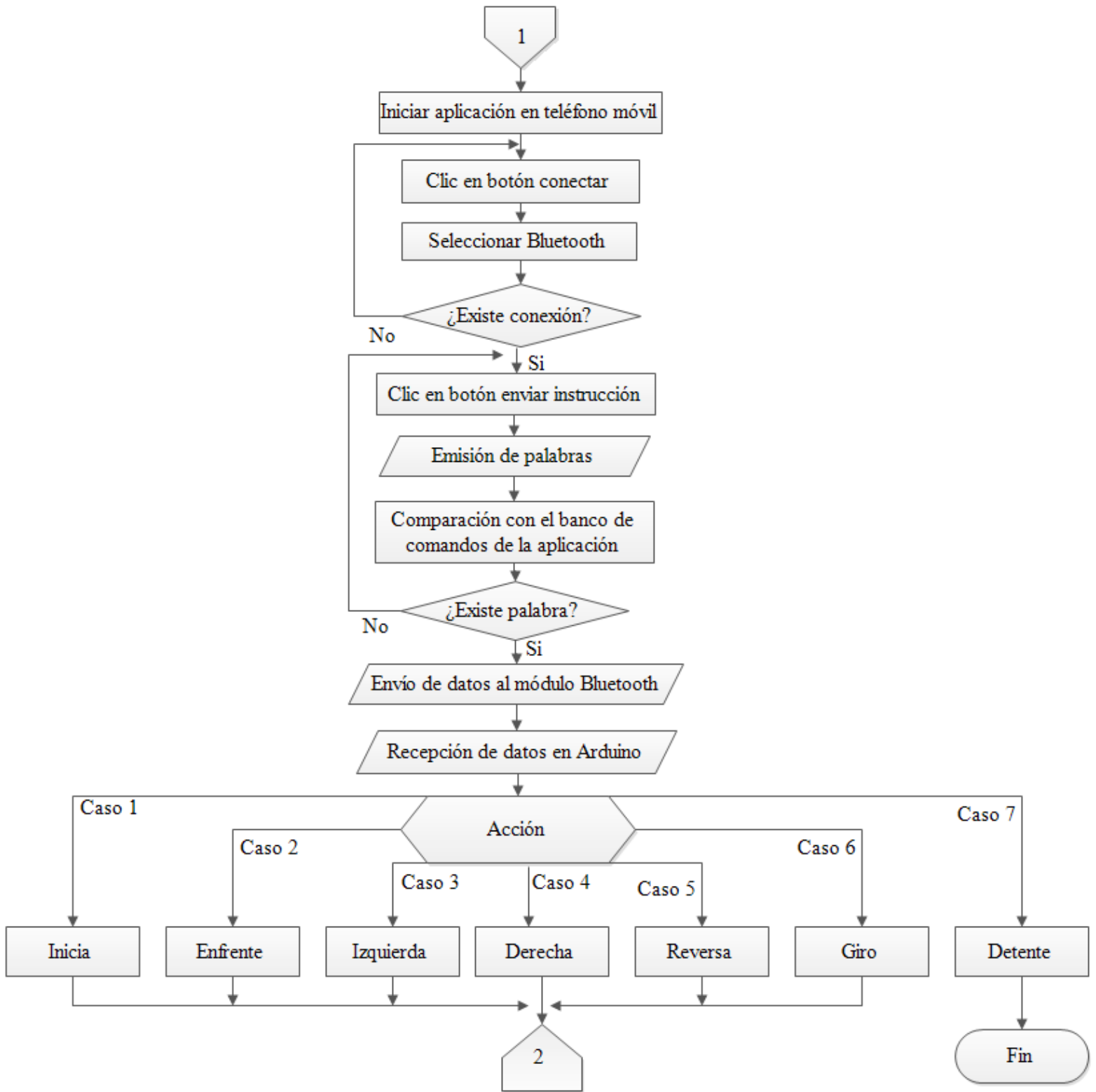


Figura 5.7. Diagrama de flujo del método de comunicación remota (continuación del diagrama de la Figura 5.6).



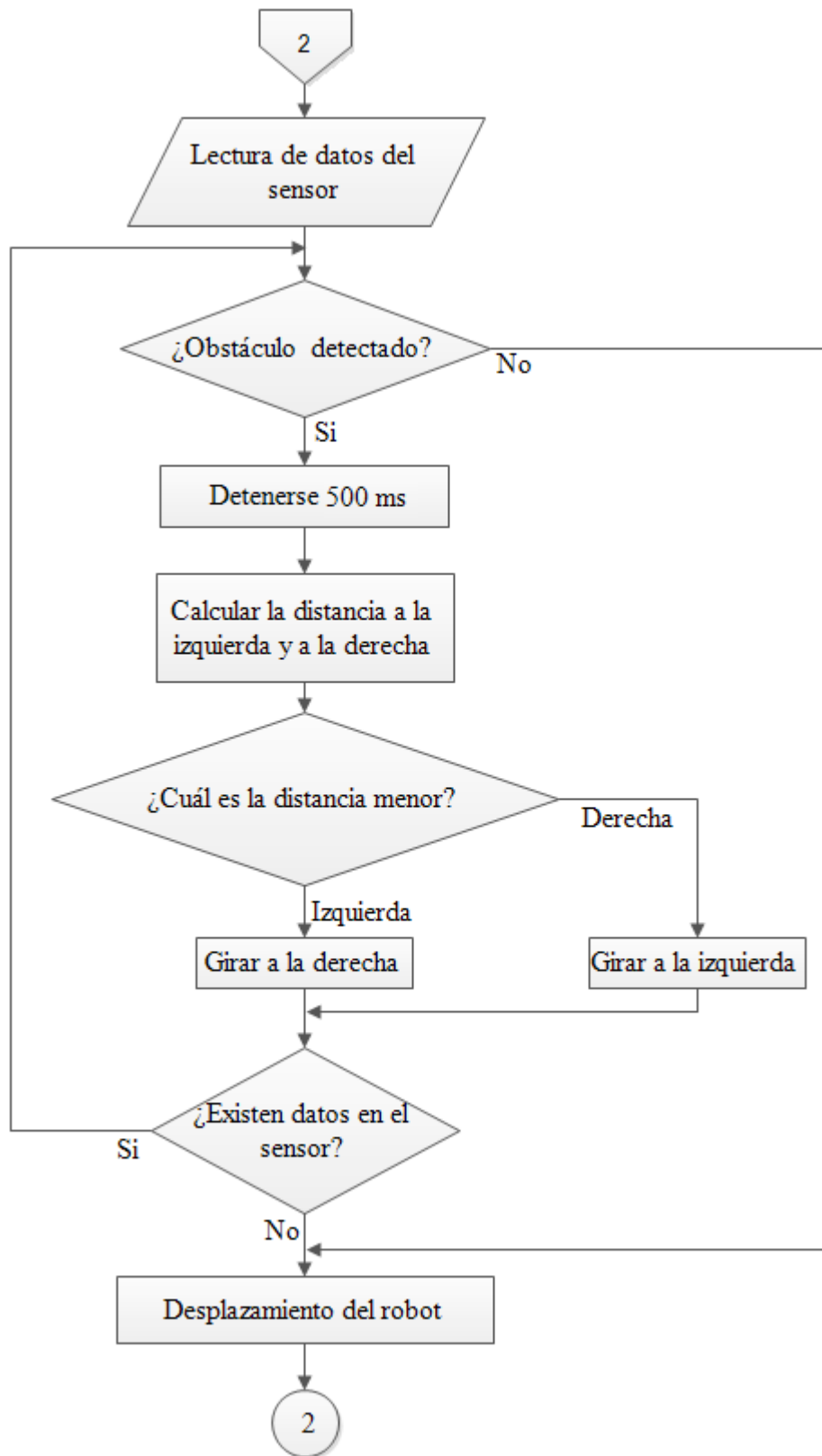


Figura 5.8. Diagrama de flujo correspondiente a la evasión de obstáculos.

- **Aplicación**

Para el desarrollo de la aplicación móvil encargada de recibir las instrucciones del usuario, realizar la conexión Bluetooth y hacer el envío de información correspondiente al módulo, para después recibirla, procesarla y compararla con las respectivas instrucciones del sistema, se optó por la herramienta de programación *MIT App Inventor 2*; que es una plataforma de Google que brinda una opción en la creación de aplicaciones para dispositivos del entorno Android. *App Inventor* es un lenguaje de programación basado en bloques y orientado a eventos. Además, es una herramienta en línea, accesible a través de un navegador web, obviamente el usuario deberá disponer de una cuenta de correo en Google (Google, 2015).

Así mismo, resulta importante, considerar los requisitos mínimos del sistema para trabajar dentro de este entorno:

- Navegador:
  - Mozilla Firefox 3.6 o superior.
  - Safari 5.0 o superior.
  - Google Chrome 4.0 o superior.
- Sistema operativo
  - Windows: Vista, Windows 7 o superior.
  - GNU/Linux: Ubuntu 8, Debian 5.

Para llevar a cabo la comunicación entre el Arduino y el módulo Bluetooth fue necesario realizar un análisis previo, acerca de la configuración para que estos dos dispositivos interactúen, logrando la transmisión y recepción de datos mediante el puerto serie. El proceso de esta configuración se muestra en la sección 5.4.4, además, se incluyen los comandos básicos para establecer las especificaciones requeridas por el módulo como, nombre, contraseña, rol (si recibe o envía datos) y velocidad de transmisión. A continuación, en la Figura 5.9 se representa el flujo correspondiente de la información entre estos dos dispositivos al momento de ser configurados.

Al finalizar el proceso de configuración entre el módulo Bluetooth y el Arduino, se da paso al desarrollo de la Aplicación móvil, descrito en la siguiente sección, de igual manera se establecen los pasos a seguir para que la aplicación sea ejecutada correctamente. Resulta útil representar el funcionamiento de esta haciendo uso de un diagrama de flujo,

para llevar el seguimiento de información desde el inicio hasta el final de su ejecución, como se observa en la Figura 5.10.

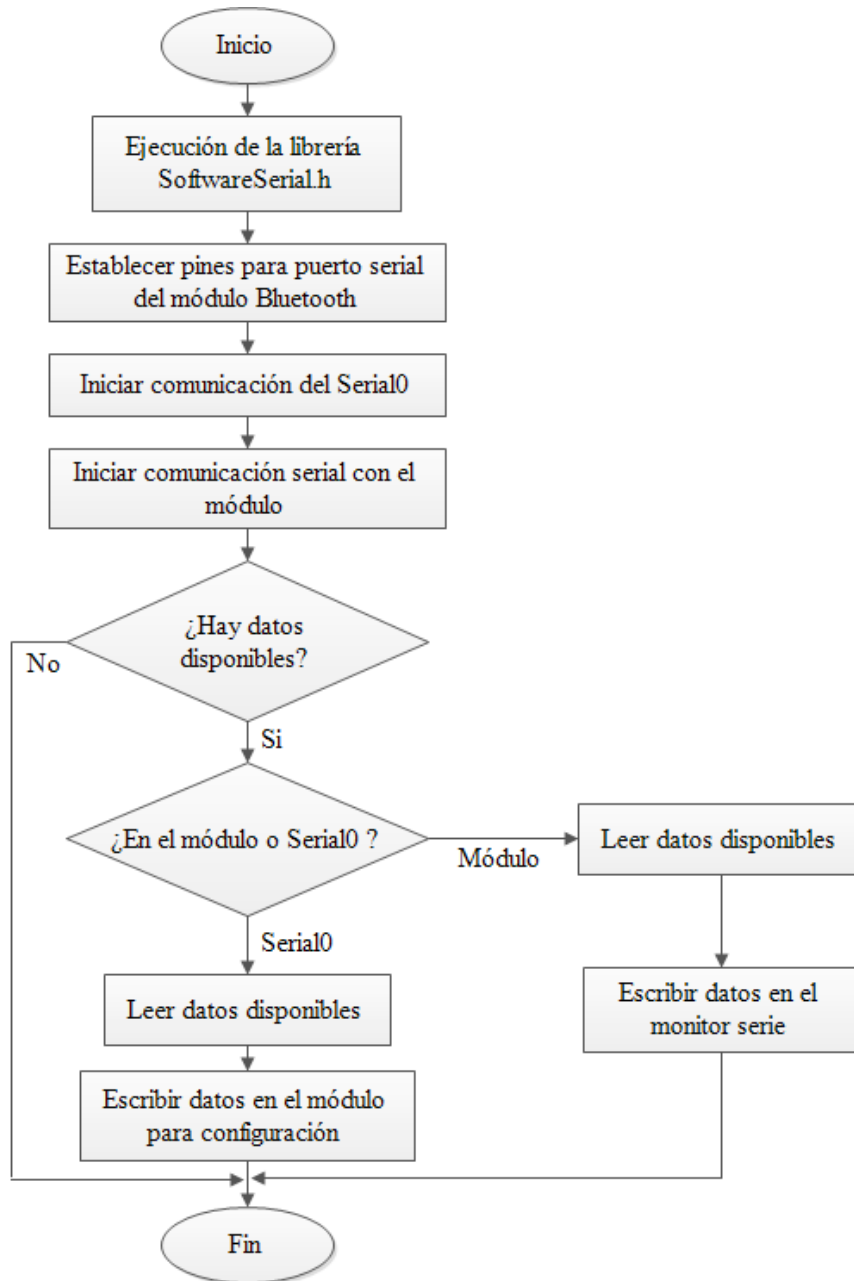


Figura 5.9. Diagrama de flujo de la comunicación y configuración del módulo Bluetooth HC-05 con el Arduino MEGA.

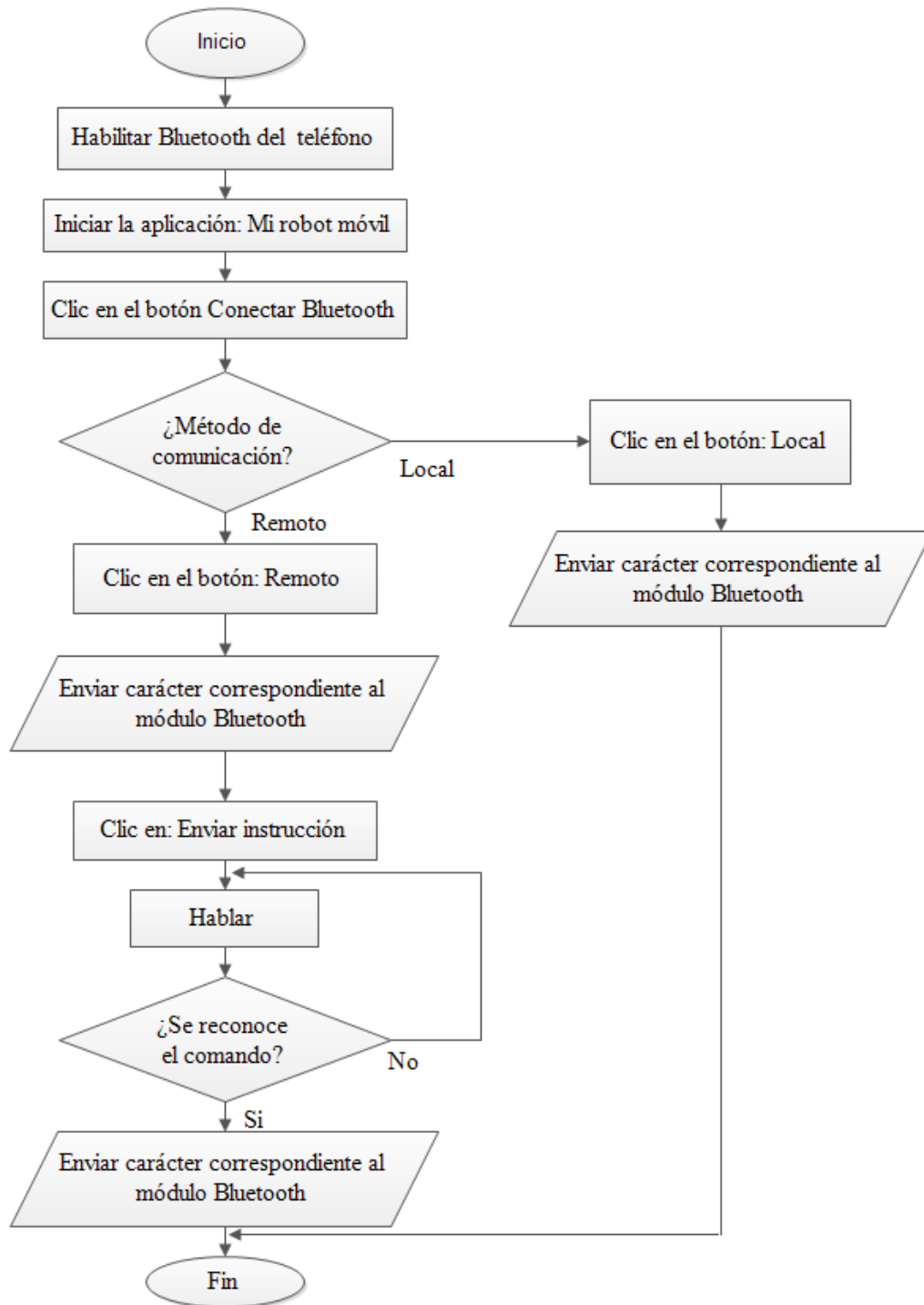


Figura 5.10. Diagrama de flujo del funcionamiento de la aplicación Mi robot móvil.

## 5.4 Construcción

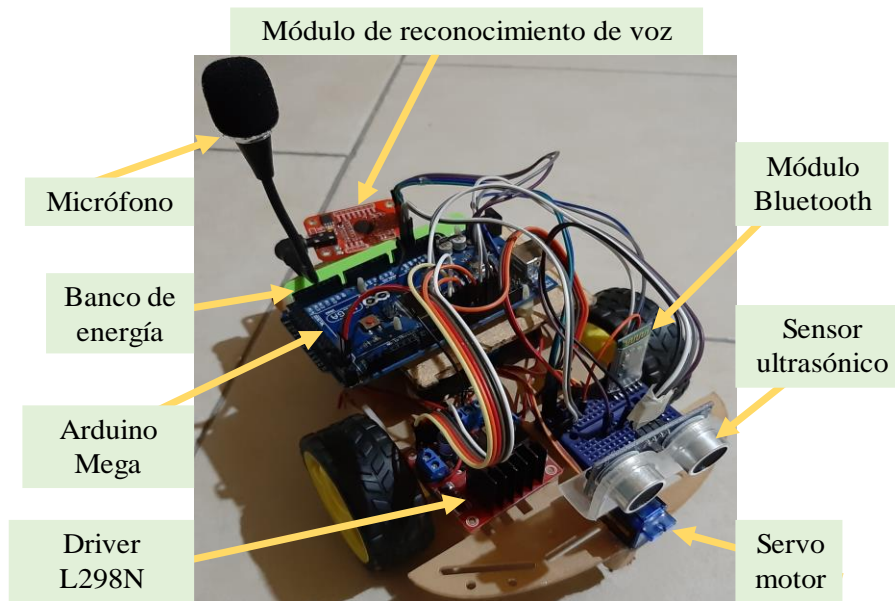
### 5.4.1 Ensamble del robot móvil

Durante la etapa de desarrollo de este trabajo de tesis, el ensamblaje del robot móvil es una de las principales acciones para dar seguimiento a la configuración de los diferentes dispositivos integrados, junto con la realización de pruebas oportunas con cada uno de ellos. El funcionamiento del robot móvil está regulado dentro del entorno de programación Arduino, donde se especifican instrucciones para ejecutar variedad de operaciones, algunas de ellas pueden ser, la detección de obstáculos, que distancia hay que considerar entre los objetos para lograr evadirlos, y la velocidad de las ruedas.

Para el ensamble del móvil, es necesario contar con cada uno de los componentes que resultan indispensables para su correcto funcionamiento, en la Tabla 5.1, se presenta un listado de cada uno de ellos, seguido de la Figura 5.11, en la cual se observa el ensamblaje completo del robot móvil.

*Tabla 5.1 Componentes del robot móvil*

<b>Cantidad</b>	<b>Componentes</b>
1	Arduino MEGA 2560
1	Driver L298N
1	Módulo de reconocimiento de voz: Elechouse Voice Recognition v3.
1	Sensor ultrasónico HC - SR04
4	Baterías a 1.5 V cada una
1	Batería a 5 V
2	Motores DC (corriente directa)
1	Módulo Bluetooth HC – 05
1	Chasis de acrílico
3	2 ruedas motorizadas, 1 omnidireccional
1	Servo motor SG90

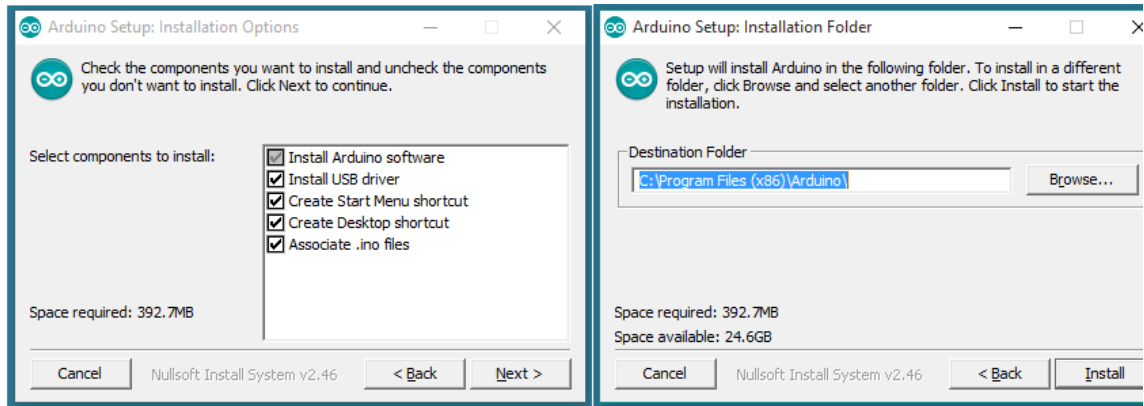


*Figura 5.11. Ensamble del robot móvil.*

Al finalizar la etapa de ensamblaje del robot móvil, lo que continua es la configuración de cada uno de los componentes integrados con el Arduino para su correcto funcionamiento, como se observa en las siguientes secciones.

#### **5.4.2 Instalación y configuración del software de programación Arduino**

Para la instalación de Arduino primeramente se descarga el software Arduino IDE (*Integrated Development Environment*, Entorno de Desarrollo Integrado) de forma gratuita, de la página oficial del mismo; es un archivo ejecutable que incluye sus controladores, para evitar la instalación manual de estos. Al tener la descarga completa, se continua con la instalación del software, permitiendo la instalación también del controlador. Es importante mencionar que el usuario es quien elige que componentes instalar, pero, es recomendable seleccionar todos los disponibles en ese momento para tener un debido funcionamiento. Al seleccionar Instalar, se da la opción de elegir un directorio, pero, es sugerible mantener el recomendado, así mismo el proceso inicia con la extracción de todos los archivos necesarios para ejecutar correctamente el software Arduino.



*Figura 5.12. Instalación del software Arduino IDE*

Como se ha mencionado la programación del sistema para el control del robot móvil con percepción auditiva, fue desarrollado mediante el uso del software Arduino, tomando en cuenta los requerimientos necesarios para el desarrollo del sistema, además de que, Arduino cuenta con librerías respectivas de los dispositivos a utilizar como, el módulo Bluetooth, el de reconocimiento de voz, y el sensor ultrasónico permitiendo el uso de ellos, para lograr su debida interacción. Al ser agregadas dichas librerías, Arduino, permite la detección del servo, por ejemplo. Además, de contener los métodos de control, clases y propiedades de este.

```
#include <Servo.h>
#include <SoftwareSerial.h>
#include "VoiceRecognitionV3.h"
```

*Figura 5.13. Ejemplo de librerías respectivas a los dispositivos*

Posterior a la investigación correspondiente sobre el software Arduino, se obtiene como resultado la necesidad de inclusión de las librerías observadas en la Figura 5.13, para el correcto funcionamiento de los dispositivos a emplear. Es importante mencionar que para el uso de librerías de Arduino es necesario incluirlas al momento de desarrollar un proyecto en este entorno, para ello se realiza lo siguiente, Figura 5.14: Ir al IDE de Arduino, en la opción menú > Programa > Incluir librería.

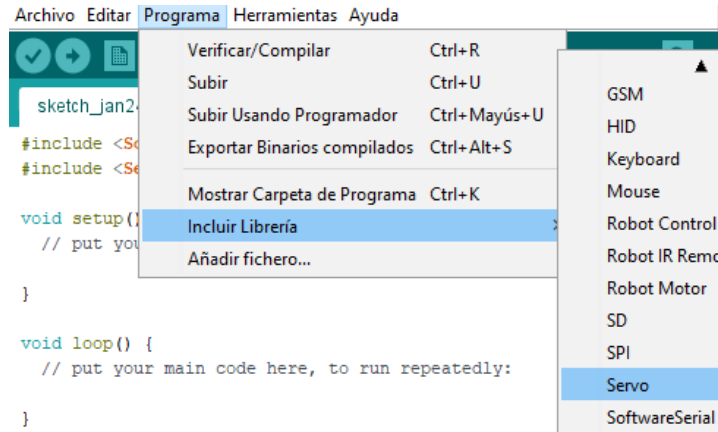


Figura 5.14. Ejemplo de la inclusión de librerías previamente instaladas.

### 5.4.3 Desarrollo del algoritmo para la evasión de obstáculos

Para iniciar la etapa de desarrollo del algoritmo de evasión de obstáculos es importante mencionar que en este apartado se hace uso del sensor ultrasónico, que es el encargado principalmente de la detección de objetos posicionados únicamente de frente al mismo, por tal motivo se integra como base el servomotor para que de acuerdo con el movimiento de su eje se tenga una posición diferente para el sensor, y de esta manera obtener la detección de objetos en tres diferentes posiciones, específicamente en los ángulos de 30, 100, 170 grados.

Antes de realizar este apartado, es útil conocer el funcionamiento de un sensor ultrasónico (Llamas, 2015): Este se encarga de medir el tiempo existente entre el envío y la recepción de un pulso sonoro, basándose en la velocidad del sonido que es 343 m/s. Para saber cuál es el tiempo que se tarda el sonido en recorrer un centímetro, es necesario convertir a unidades los datos anteriores, como se observa en la ecuación:

$$343 \frac{m}{s} \cdot 100 \frac{cm}{m} \cdot \frac{1}{1000000} \frac{s}{\mu s} = \frac{1}{29.2} \frac{cm}{\mu s}$$

Al obtener dicho resultado, la distancia se calcula a partir del tiempo entre el envío y recepción de un pulso, haciendo uso de la ecuación:

$$Distancia(cm) = \frac{Tiempo(\mu s)}{29.2 * 2}$$



El dos que se coloca en la división es referente al tiempo que un pulso se tarda en ir y regresar hasta donde se encuentra en objeto, indicando que la distancia que recorre el pulso equivale al doble de la que se necesita medir. Para calcular la distancia se obtienen las siguientes ecuaciones:

$$Tiempo = 2 * \left( \frac{Distancia}{Velocidad} \right)$$

Finalmente, se simplifica:

$$Distancia = Tiempo * \frac{Velocidad}{2}$$

Posteriormente se realiza un segmento de código, presentado a continuación, donde se implementa una instrucción para obtener la distancia en centímetros al detectar un objeto.

```
int Centimetros(){ //Función que regresa un valor de distancia en cm
    digitalWrite(Pin_trig, LOW); //2 Ms en bajo para un pulso limpio
    delayMicroseconds(2);
    digitalWrite(Pin_trig, HIGH); //pulso del trigger durante 10 Ms
    delayMicroseconds(10);
    digitalWrite(Pin_trig, LOW);
    tiempo = pulseIn(Pin_echo, HIGH); //medición del tiempo entre pulsos
    distancia = (tiempo/2) / 29; //conversión de distancia a cm
    delay(5);
    return distancia; //retornar la distancia
}
```

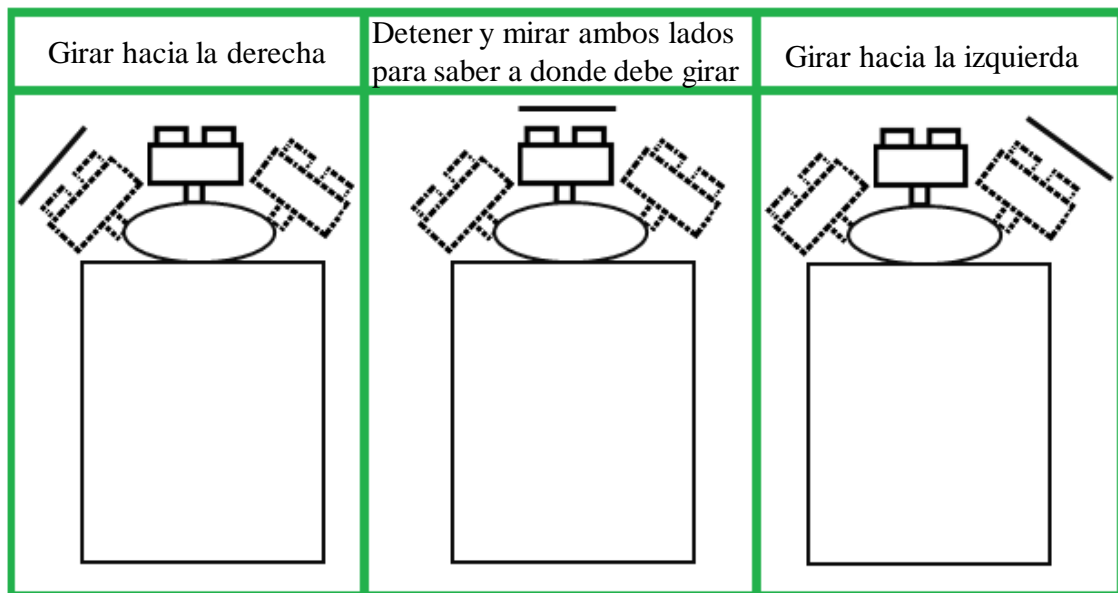
Figura 5.15. Código fuente para la obtención de la distancia.

A partir de esta instrucción, se establece el valor de 20 cm como límite para indicar que se encuentra un objeto presente y posterior a ello se definen las acciones a realizar por el móvil para evitar que este obstruya su paso. En la Tabla 5.2, se despliegan los valores correspondientes al PWM (*Pulse Width Modulation*, Modulación por Ancho de Pulsos), de los motores, que es, el que otorga la velocidad y dirección del desplazamiento del robot. Estos valores cambian cuando se presenta un objeto en una distancia igual o menor a la establecida, permitiendo la toma de decisiones de acuerdo con los datos provenientes del sensor ultrasónico. Además, en la Figura 5.16, se representan las acciones para la evasión de obstáculos en caso de que el robot detecte algún objeto, donde se indica lo siguiente:

- Si el objeto está en la izquierda entonces; gira hacia la derecha.
- Si el objeto está en la derecha entonces; gira a la izquierda.
- Si el objeto está enfrente entonces; se detiene, mira hacia ambos lados y de acuerdo con la distancia establece a donde debe girar evitando los obstáculos presentes.

*Tabla 5.2. Acciones de los motores para realizar la evasión de obstáculos.*

Posición del objeto	Acción del robot	PWM del motor derecho	PWM del motor izquierdo
Enfrente	Calcula distancia menor. Si distancia menor = izquierda. Gira a la derecha.	60	180
	Si distancia menor = derecha. Gira a la izquierda.	180	60
Izquierda	Gira a la derecha.	60	180
Derecha	Gira a la izquierda.	180	60



*Figura 5.16. Representación de acciones para la evasión de obstáculos.*

#### 5.4.4 Comunicación y configuración del módulo de reconocimiento de voz

Para dar inicio al proceso de configuración, es primordial la conexión correspondiente para la comunicación entre el módulo de reconocimiento de voz *Elechouse Voice Recognition V3* y el Arduino MEGA, seguido de ello, realizar la investigación para saber cuál es el método por seguir para la grabación de palabras o frases en dicho módulo. Como se menciona con anterioridad, al incluir la librería "*VoiceRecognitionV3.h*", Arduino, permite hacer uso de los métodos y propiedades de este dispositivo, por tal motivo, al realizar esta acción, se da inicio a grabar las siete palabras establecidas para llevar el control del robot móvil: Inicia, adelante, atrás, izquierda, derecha, vuelta y alto.

Una vez que se tiene la conexión del módulo con el Arduino se hace empleo de uno de los métodos de la librería, que es el encargado de permitir la grabación de las palabras. En el apartado Monitor serie del Arduino, se observa lo que contiene el módulo, y ahí mismo se realizan las configuraciones correspondientes. Antes de iniciar la configuración, en la Figura 5.17 se muestra una serie de ejemplos acerca de los comandos que se pueden emplear y cuál es su función.

COMMAND	FORMAT	EXAMPLE	Comment
train	train (r0) (r1)...	train 0 2 45	Train records
load	load (r0) (r1) ...	load 0 51 2 3	Load records
clear	clear	clear	remove all records in Recognizer
record	record / record (r0) (r1)...	record / record 0 79	Check record train status
vr	vr	vr	Check recognizer status
getsig	getsig (r)	getsig 0	Get signature of record (r)
sigtrain	sigtrain (r) (sig)	sigtrain 0 ZERO	Train one record(r) with signature(sig)
settings	settings	settings	Check current system settings
help	help	help	print this message

Figura 5.17. Comandos de configuración del módulo de Reconocimiento de voz.

Al realizar un análisis acerca de los comandos a emplear para la configuración del módulo, se conoce la instrucción *sigtrain* que sirve para grabar una palabra o frase. Para indicar que se va a grabar una palabra se debe colocar tanto el comando, como un número identificador de la posición que ocupa la variable, que se debe encontrar en el rango de 0-254 debido, a la cantidad de palabras que el módulo permite almacenar, además, se coloca el nombre que la variable tendrá, finalizando así, la instrucción completa: *sigtrain* + número de posición + variable, como se observa en la Figura 5.18.

```

COM4 (Arduino/Genuino Mega or Mega 2560)
sigtrain 0 Inicia

vr          vr          vr
getsig     getsig (r)    getsig 0
sigtrain   sigtrain (x) (sig) sigtrain 0 ZERO
settings   settings    settings
help       help        help
-----
sigtrain 0 Inicia
-----
Record: 0   Speak now
Record: 0   Speak again
Record: 0   Success
Success: 1
Record 0    Trained
SIG: Inicia

```

Figura 5.18. Comando de grabación de palabras en el módulo de reconocimiento de voz.

Seguido de indicar la instrucción para grabar, el usuario debe pronunciar y repetir la palabra que indicó, si el proceso es satisfactorio, se muestra un mensaje de cuál es la palabra que está grabando. Este proceso se repite las veces que sea necesario con las palabras o instrucciones correspondientes a las variables que fueron establecidas anteriormente. Lo que continua es realizar la carga de palabras al módulo y esta acción se ejecuta con el comando load (carga), más los números de posición, en este caso, como son siete palabras las que fueron grabadas, la instrucción se establece de la siguiente manera: load 0 1 2 3 4 5 6, como se muestra en la Figura 5.19.

```

load 0 1 2 3 4 5 6

Load success: 7
Record 0   Loaded
Record 1   Loaded
Record 2   Loaded
Record 3   Loaded
Record 4   Loaded
Record 5   Loaded
Record 6   Loaded

```

Figura 5.19. Carga satisfactoria de las 7 palabras a emplear.

Para finalizar la etapa correspondiente a la configuración del módulo y la grabación de las palabras, es útil realizar una verificación para comprobar cuales son las palabras guardadas y la posición que corresponde, esta acción se lleva a cabo mediante el comando getsig, más el número de la posición que se quiere observar, quedando el comando completo, por ejemplo, getsig 0. En este caso se realizó del 0 al 6 para consultar las siete palabras integradas. Como resultado se obtiene en la Figura 5.20, una lista de las palabras previamente cargadas con su respectivo número de identificación.

getsig 0	getsig 4
Signature:Inicia	Signature:Derecha
getsig 1	getsig 5
Signature:Enfrente	Signature:Giro
getsig 2	getsig 6
Signature:Reversa	Signature:Detente
getsig 3	
Signature:Izquierda	

*Figura 5.20. Verificación de las palabras grabadas y cargadas en el módulo de reconocimiento de voz.*

#### **5.4.5 Comunicación y configuración del módulo Bluetooth.**

Para la etapa de configuración, es indispensable iniciar con la conexión necesaria para llevar a cabo la comunicación inalámbrica entre el módulo Bluetooth HC-05 y el Arduino MEGA, mediante uno de los cuatro puertos seriales de los que dispone esta tarjeta de desarrollo; no debe ser el mismo puerto que se empleó para la comunicación con el módulo de reconocimiento de voz debido que, serán empleados dos puertos distintos para la comunicación con ciertos módulos.

Se indagó acerca de la configuración que se tiene que realizar en dicho módulo para poder recibir datos desde algún otro dispositivo; en este caso se le establece el rol de esclavo, para que reciba información desde un teléfono móvil, a diferencia del rol de maestro que sería empleado cuando el módulo se usa para enviar datos. Para configurar el módulo es necesario pulsar el botón que viene integrado en el mismo para acceder al modo de configuración donde por medio de comandos AT se cambian o establecen características de este como, nombre, contraseña, el rol que cumple y aspectos de la comunicación. Los comandos AT, sirven para obtener información acerca de la configuración del módulo y especificar valores necesarios; a continuación, en la Tabla 5.3.

*Tabla 5.3 Comandos básicos para la configuración del módulo Bluetooth.*

<b>Comando</b>	<b>Función</b>
AT+NAME	Nombre del módulo
AT+PSWD	Contraseña
AT+ROLE	Rol del dispositivo: 0, Esclavo y 1, Maestro
AT+UART	Parámetros de comunicación
AT+ORGL	Volver a valores de fabrica
AT+RESET	Volver a modo usuario
<b>Uso del comando</b>	Comando ?: Consulta o lectura, sin modificar. Comando = Asignación de valores.

Una vez que se tiene la conexión correspondiente y se conocen los comandos básicos, lo siguiente es, llevar a cabo la configuración necesaria. Es conveniente mencionar que el módulo Bluetooth que se emplea cuenta con dos modos de uso: configuración y usuario, el primero; necesario para otorgar el nombre, rol y establecer parámetros de comunicación, y el segundo para llevarla a cabo con algún otro dispositivo. Para entrar a estos modos, el módulo cuenta con un pulsador, que es presionado cuando se quiere cambiar de un modo a otro.

Antes de llevar a cabo la configuración, se consultan los valores con los que cuenta el módulo o valores de fábrica, para así saber cuáles son los parámetros por cambiar, en la Figura 5.18 se muestran los parámetros iniciales, por ejemplo, se consulta el nombre con el comando `AT+NAME?`, y responde `H-C-2010-06-01`. A partir de ello se procede con establecer los parámetros necesarios para llevar a cabo la comunicación con algún otro dispositivos, y con esta acción se finaliza el proceso de configuración del módulo Bluetooth donde fueron empleados los comandos mostrados en la Tabla 5.3, para poder recibir los datos enviados desde el teléfono móvil.

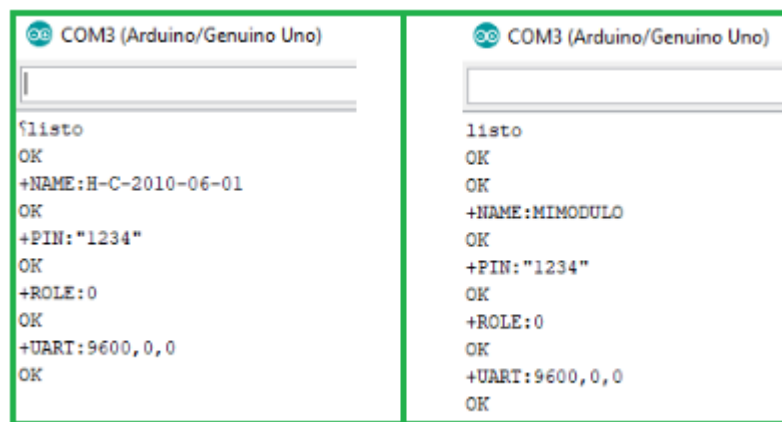


Figura 5.18. Valores de fabrica del módulo/ Valores asignados.

#### 5.4.6 Desarrollo de Aplicación: Mi robot móvil

Para llevar a cabo el diseño y desarrollo de la aplicación móvil, se hace uso de la plataforma en línea, denominada *MIT App Inventor 2*, que cuenta con dos vistas para el usuario: Diseño y bloques, donde; en la primera; se visualiza el contenido de la aplicación de acuerdo a una vista preliminar y en la segunda; se realiza el desarrollo de las instrucciones necesarias para complementar la parte lógica, de control y funcionamiento de la aplicación, como se aprecia en la Figura 5.19 y Figura 5.20.

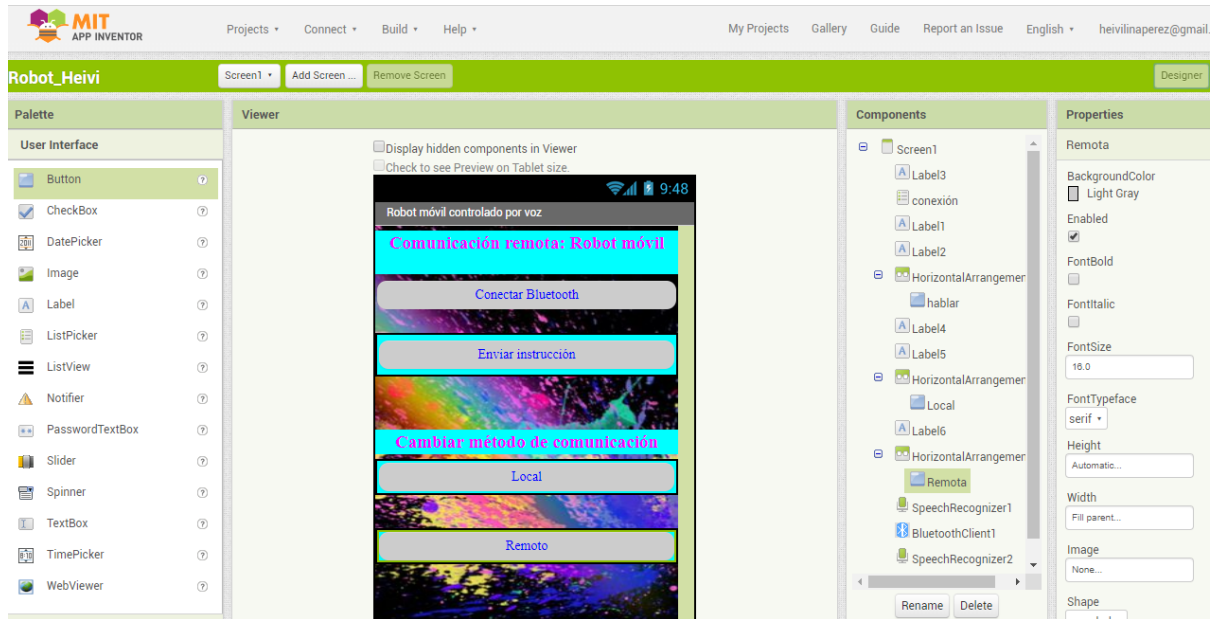


Figura 5.19 Vista de diseño de la aplicación móvil.

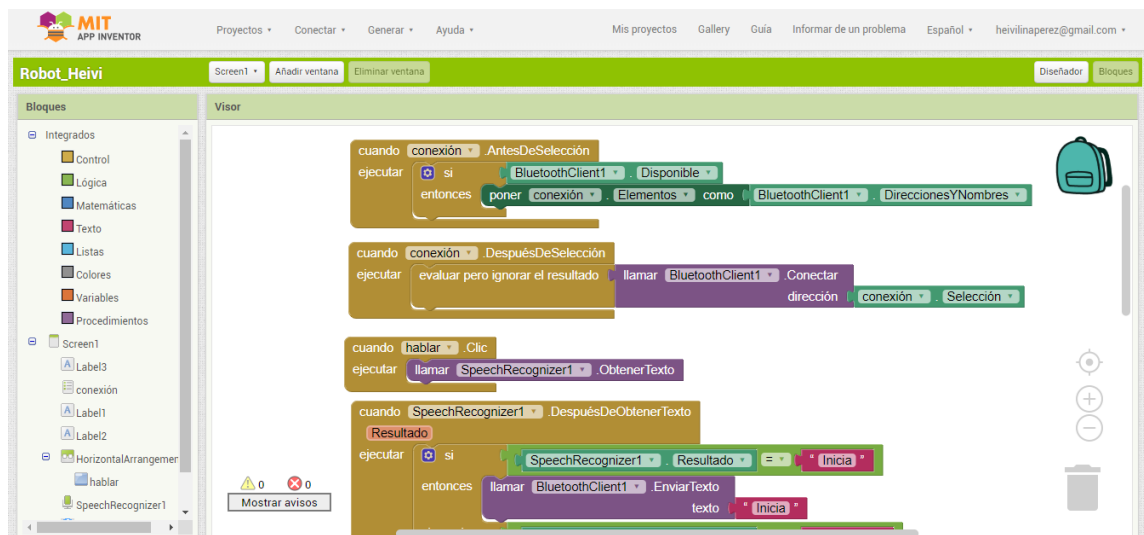


Figura 5.20. Vista por bloques.

En la vista de diseño, es donde el usuario desarrolla la interfaz principal de la aplicación. Aquí, se añaden todos y cada uno de los elementos a utilizar como lo son; etiquetas, botones, imágenes, colores, iconos, nombre de la aplicación, reconocimiento de voz, el cliente Bluetooth, entre algunos otros. Para el diseño de la interfaz, el usuario define las



especificaciones de los elementos, como, tipo, color y tamaño de fuente a emplear. Cada vez que se realiza una modificación, los cambios son guardados para así obtener una vista preliminar de la aplicación, como se muestra en la Figura 5.21.a.

Al finalizar el diseño de la interfaz, lo que corresponde es, llevar a cabo la parte lógica y de control para el funcionamiento de la aplicación, cambiando a la vista por bloques. Aquí, se especifica la función de cada uno de los botones integrados en la interfaz. A continuación, se presenta una serie de imágenes correspondientes a este apartado, iniciando con lo que representa cada bloque, identificados principalmente por su función y color, como se observa en la Figura 5.21.b. Seguido de la especificación de lo que sucede cuando se acciona cada uno de los botones.

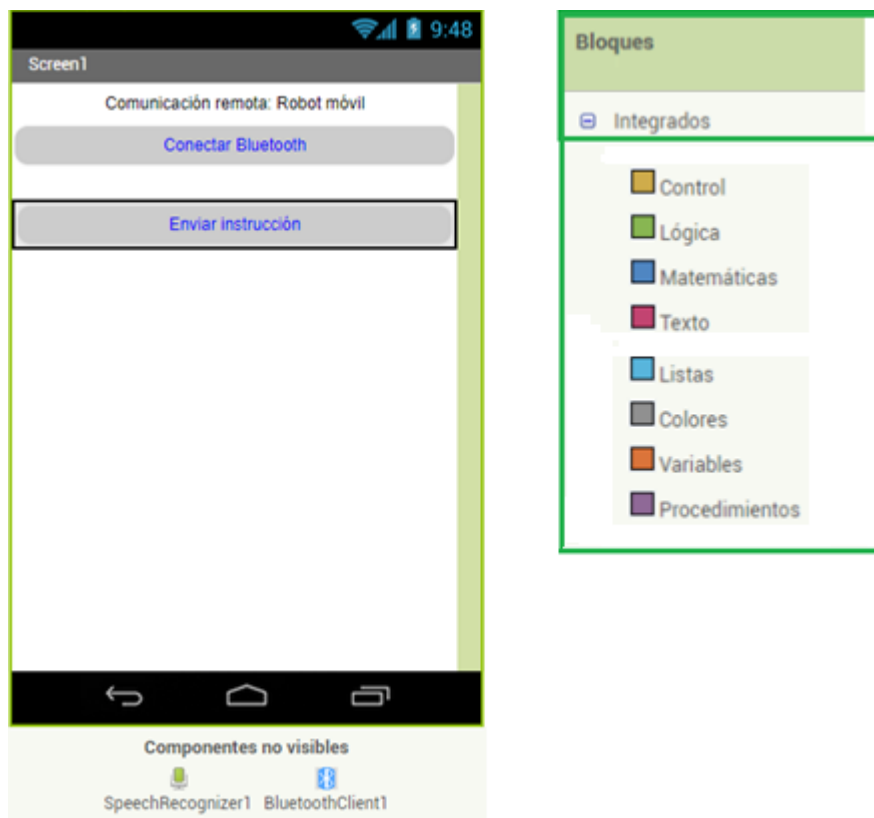


Figura 5.21. a) Vista inicial de la aplicación. b) Representación de bloques.

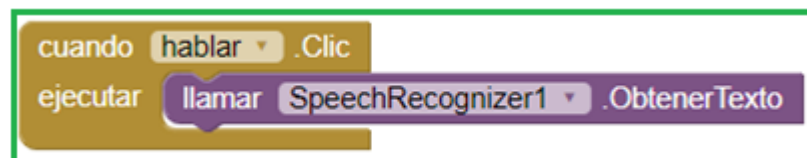
Botón Conectar Bluetooth o conexión: Corresponde a la comunicación mediante el Bluetooth del teléfono móvil donde será instalada la aplicación con el módulo a emplear, como se muestra la estructura en la Figura 5.22.

- Antes de la selección: Se establece que, si el Bluetooth del dispositivo se encuentra disponible, el botón conectar muestra los elementos de los clientes Bluetooth, en este caso direcciones y nombres de dispositivos cercanos activos.
- Después de la selección: Evalúa la dirección del Bluetooth que fue seleccionado, para iniciar la conexión remota con el dispositivo cliente.



*Figura 5.22. Instrucciones del botón conexión.*

Botón Enviar Instrucción o hablar: Al hacer clic sobre este, hace un llamado al reconocimiento de voz; para, posteriormente guarda lo que reconoce de forma temporal como texto, como se presenta en la Figura 5.23.



*Figura 5.23. Instrucciones para el botón hablar*

Después de obtener el texto de la palabra mencionada, lo que corresponde es llevar a cabo una serie de condiciones para especificar que, si el resultado del reconocimiento de voz es igual a cierto texto como, por ejemplo, inicia. Se hace el envío de un dato al dispositivo cliente, en este caso la letra A. Es importante mencionar que, dicho proceso se repite para

cada una de las palabras que se van a identificar tanto, para la obtención del texto como, para el envío de un dato correspondiente a cada una de estas, la única diferencia es el comando y el carácter, como puede ser visualizado a continuación en la Figura 5.24.

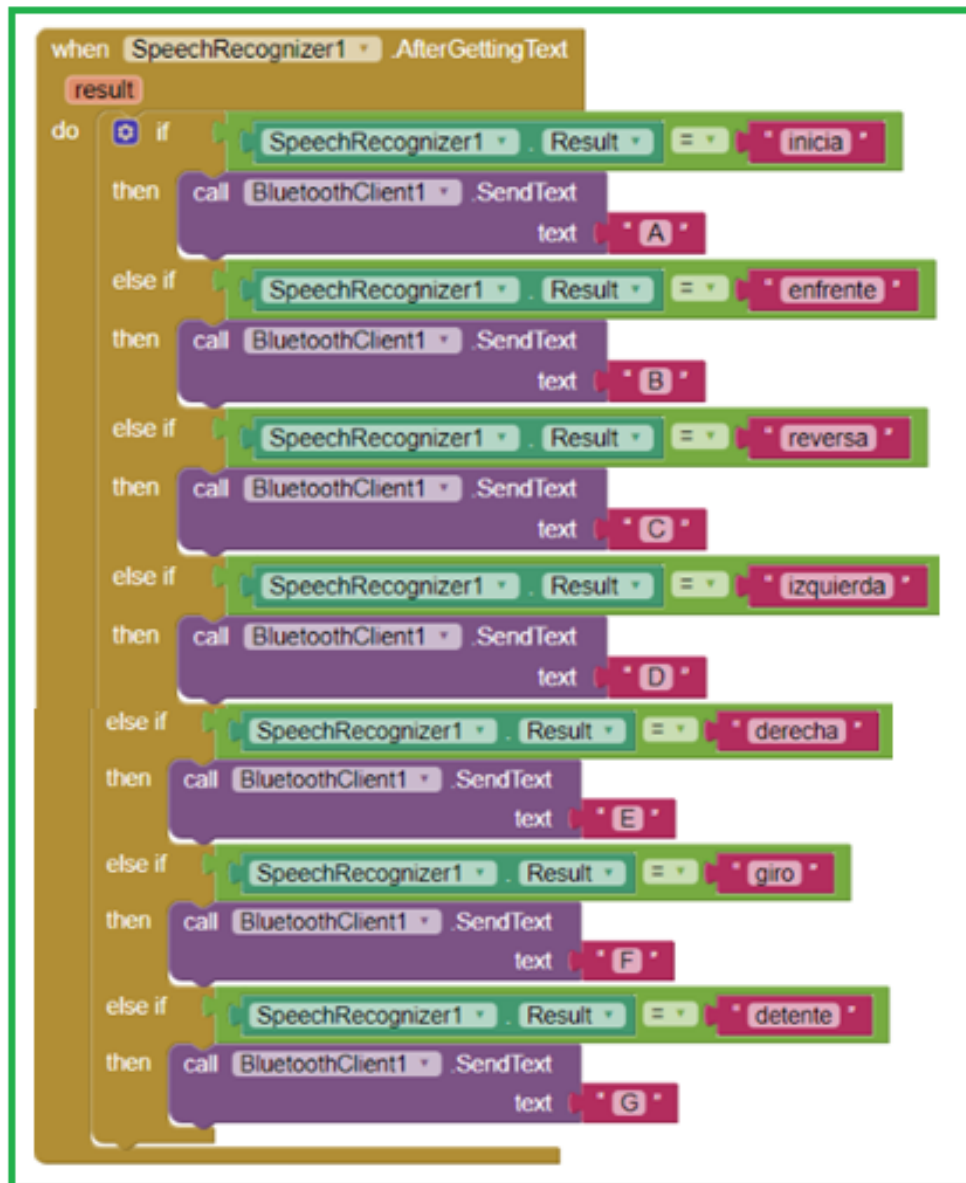


Figura 5.24. Condiciones del funcionamiento del botón Enviar instrucciones.

Los botones correspondientes a la sección para el cambio con respecto al método de comunicación a utilizar son denominados como, Local y Remota, que consiste en el empleo de un método similar a excepción, de que no hace uso del reconocimiento de voz.

Cabe mencionar que el proceso para dichos botones es el mismo; al dar clic sobre alguno, hace un llamado al Bluetooth y envía un dato correspondiente para cada uno de estos, como puede ser visualizado en la Figura 5.25.

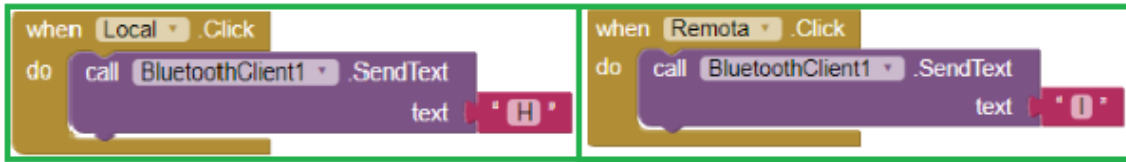
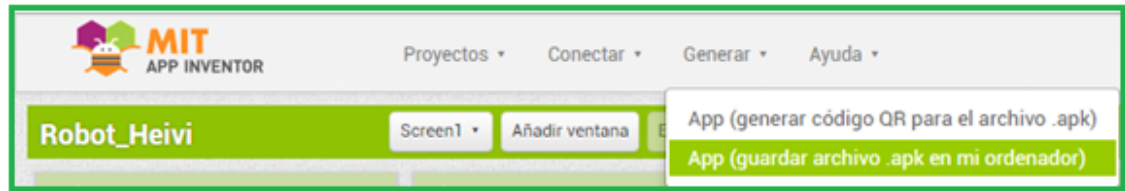


Figura 5.25. Cambiar método de comunicación

Para finalizar el desarrollo de la parte lógica en la sección de los bloques y el diseño de la interfaz, en la vista de diseñador; se obtiene una vista previa de la Aplicación. En este caso se observa, la cantidad de botones, el color, forma y tamaño de estos, entre algunos otros, como lo indica la Figura 5.26. Vista previa de la aplicación. A continuación, se guarda el proyecto perteneciente a la aplicación desarrollada, obteniendo dos posibles opciones para hacerlo; como se representa en la Figura 5.27. Construir aplicación y opciones de guardado.



*Figura 5.26. Vista previa de la aplicación.*



*Figura 5.27. Construir aplicación y opciones de guardado.*

De acuerdo con el modelo de desarrollo de software en espiral, utilizado para la elaboración de este proyecto, para terminar la etapa correspondiente a la construcción de este, es útil verificar el funcionamiento de cada uno de los apartados descritos con anterioridad, por tal motivo esto se realizó de manera individual y finalmente en el conjunto del robot móvil. Los resultados obtenidos son reportados en la siguiente sección.

## **6 RESULTADOS Y DISCUSIÓN**

Al concluir las etapas correspondientes al ensamble del robot, instalación de software, configuración de los diferentes dispositivos, más el desarrollo del algoritmo y de la aplicación móvil para el funcionamiento del prototipo se obtienen los resultados observados en la presente sección.

### **6.1 Despliegue**

En la metodología de desarrollo de software en espiral; modelo seleccionado para la elaboración del presente trabajo de tesis se establece que, la fase final de esta es el despliegue del sistema, por lo que, al finalizar la etapa de construcción es indispensable realizar una serie de pruebas acerca del funcionamiento de cada uno de los apartados desarrollados en lapso del proyecto, para finalmente reunirlos en el conjunto del robot móvil. Cada una de estas pruebas se presenta a continuación.

#### **6.1.1 Funcionamiento de la aplicación: Mi robot móvil**

Una finalizada la construcción de la aplicación Mi robot móvil, fue necesario descargarla en una computadora personal, seguido de realizar el envío de esta al almacenamiento de memoria de un teléfono celular perteneciente al entorno de Android, donde, fue instalada y posterior a ello ejecutada.

En la Figura 6.1 se da a conocer el proceso anterior; iniciado con la solicitud de permisos correspondientes para la instalación de la aplicación en el dispositivo móvil y finalizado con un mensaje de comprobación que informa que esta ha sido instalada satisfactoriamente. Después de visualizar dicha notificación se procede con su ejecución y con esta acción se hizo visible el contenido de la interfaz en el teléfono celular, como se observa en la Figura 6.2.

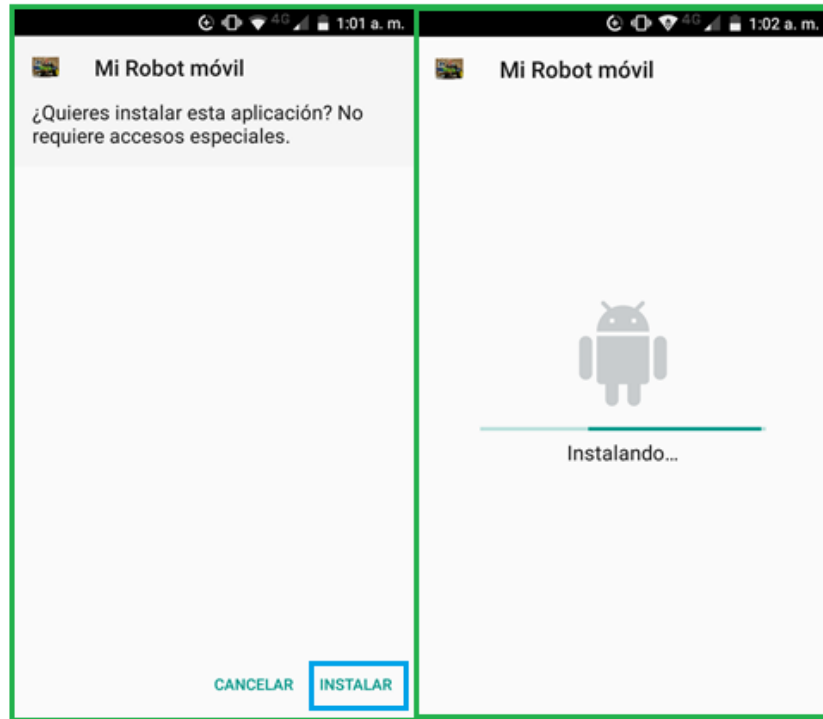


Figura 6.1. Instalación de la aplicación Mi robot móvil en Android.

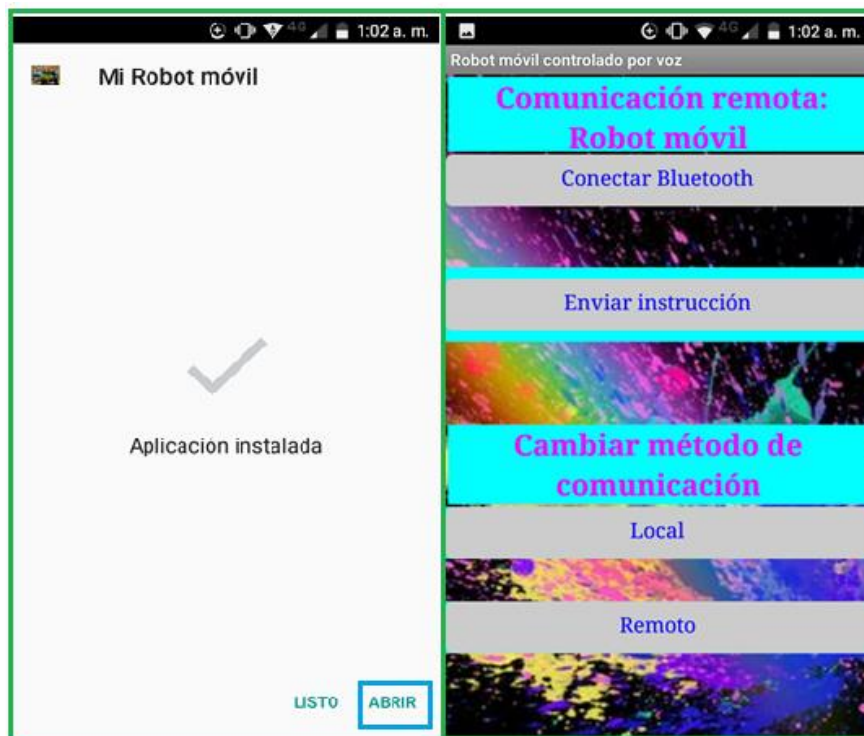


Figura 6.2. Fin de la instalación y ejecución de la aplicación.

Una vez terminada la sección de pruebas correspondientes a la aplicación móvil, resulta útil incluir una imagen representativa del funcionamiento total de la interfaz de la Aplicación móvil, donde se observan las acciones realizadas a partir de pulsar algún botón. Cabe mencionar que al hacer clic en el botón “Conectar Bluetooth”, se muestra una lista de los nombres de dispositivos bluetooth cercanos al teléfono móvil, mientras que al presionar “Enviar instrucción”, se manda a llamar el reconocimiento de voz de Google.

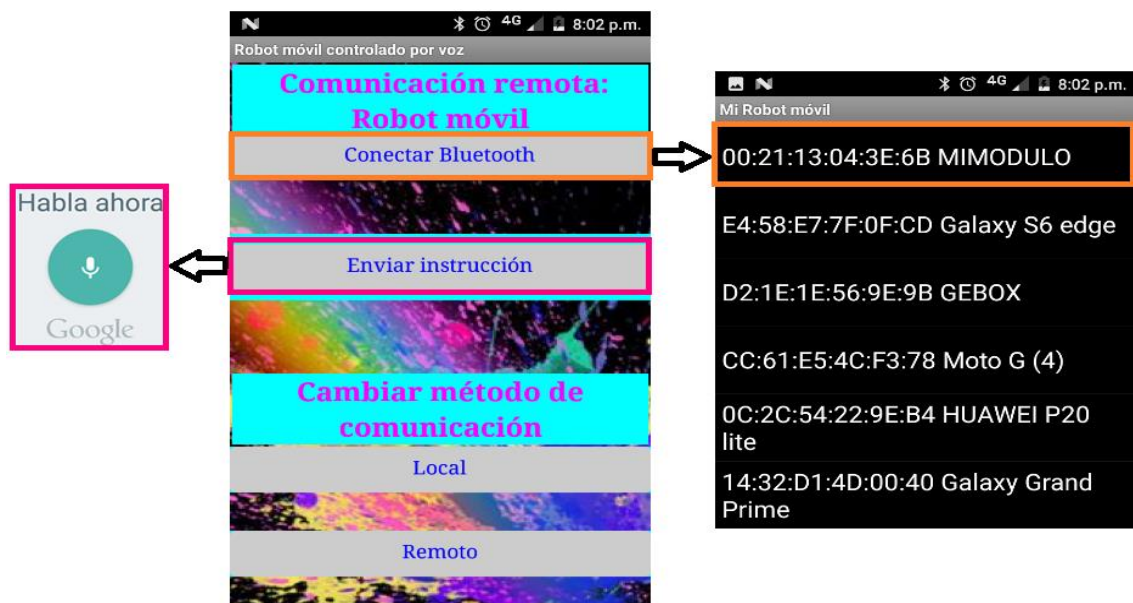


Figura 6.3. Funcionamiento total de la aplicación: Mi robot móvil.

### 6.1.2 Reconocimiento de los comandos de voz por el módulo

Cuando ya se tiene la grabación y verificación de las palabras cargadas en el módulo de reconocimiento de voz, se realizan las pruebas necesarias para saber si el prototipo cuando se encuentra trabajando en conjunto es capaz de reconocer las palabras o comandos que el usuario menciona, cuando el programa está en ejecución y ya con las condiciones específicas, para ello es indispensable que el robot este en acción, además, que el Arduino se encuentre conectado a la computadora para mediante el monitor serie visualizar los datos que están recibiendo. En la Figura 6.4, se observa que, en efecto, el módulo ha reconocido las 7 palabras que le fueron grabadas previamente.



VR Index	Group	RecordNum	Signature
0	NONE	0	Inicia
VR Index	Group	RecordNum	Signature
1	NONE	1	Enfrente
VR Index	Group	RecordNum	Signature
2	NONE	2	Reversa
VR Index	Group	RecordNum	Signature
3	NONE	3	Izquierda
VR Index	Group	RecordNum	Signature
4	NONE	4	Derecha
VR Index	Group	RecordNum	Signature
5	NONE	5	Giro
VR Index	Group	RecordNum	Signature
6	NONE	6	Detente

Figura 6.4. Prueba de reconocimiento de palabras.

### 6.1.3 Detección de obstáculos

Al finalizar el apartado correspondiente a la evasión de obstáculos una de las pruebas realizadas consiste en iniciar el desplazamiento del robot móvil, pero, sin desconectar el Arduino de la computadora para por medio del monitor serie visualiza mensajes de alerta que fueron integrados en el código para indicar cuando se ha detectado un objeto y en qué dirección está posicionado. Además de estos datos, también se manda a imprimir la distancia en centímetros que hay desde el sensor hasta el objeto más próximo.

The screenshot shows a serial monitor window titled "COM4 (Arduino/Genuino Mega or Mega 2560)". The window contains the following text output:

```

131
continuar enfrente
0
alto objeto a la izquierda
0
alto objeto a la derecha
19
continuar enfrente
177
continuar izquierda
145
continuar derecha
0
alto objeto enfrente
178
continuar izquierda

```

At the bottom of the window, there are control buttons:  Autoscroll,  Mostrar marca temporal, Nueva línea (dropdown), 9600 baudio (dropdown), and Limpiar salida.

Figura 6.5. Alerta de detección de obstáculos.

#### **6.1.4 Comunicación Arduino - Módulo Bluetooth y aplicación móvil**

Al finalizar la etapa correspondiente a la configuración del módulo, presentada en la sección de construcción. El módulo se cambia al modo usuario, para realizar las pruebas de comunicación correspondientes entre este y el teléfono celular en el que es ejecutada la aplicación Mi robot móvil.

Es importante mencionar que, para que esta aplicación funcione de manera correcta, los pasos a seguir son:

- Activar el Bluetooth en el teléfono móvil.
- Iniciar la aplicación.
- Clic en el botón Conectar Bluetooth, al realizar esta acción aparece una lista de dispositivos Bluetooth cercanos al teléfono y disponibles, en este caso seleccionar MIMODULO, que es el nombre que le fue asignado al módulo HC-05, que es el que permite la conexión con el Arduino, como se observa en la Figura 6.6.
- Clic en el botón Enviar instrucción; se hace un llamado al reconocimiento de voz; como se muestra en la Figura 6.7, si se reconoce una palabra perteneciente al grupo de instrucciones preestablecidas, se hace envío de un carácter específico al cliente Bluetooth, que es el módulo, y este es encargado de enviarlo al Arduino, para ser procesado y finalmente indicar al robot para que ejecute la acción correspondiente a este dato. En el caso contrario, si al pulsar este botón no se presenta el reconocimiento de una palabra mencionada por el usuario, la aplicación brinda la oportunidad de volver a intentarlo, pronunciando nuevamente un comando, sin tener que regresar a la interfaz principal, como se observa en la Figura 6.8.
- Si se desea realizar un cambio en cuanto al método de comunicación del usuario con el robot, este se efectúa a través de los botones denominados: Local y/o Remoto; haciendo clic sobre alguno de los dos, y con ello, nuevamente se envía al Arduino un dato respectivo a la instrucción, para indicar la decisión tomada por el usuario con respecto a la manera en que se quiere manipular el robot móvil.



Figura 6.6. Búsqueda de dispositivos Bluetooth disponibles.

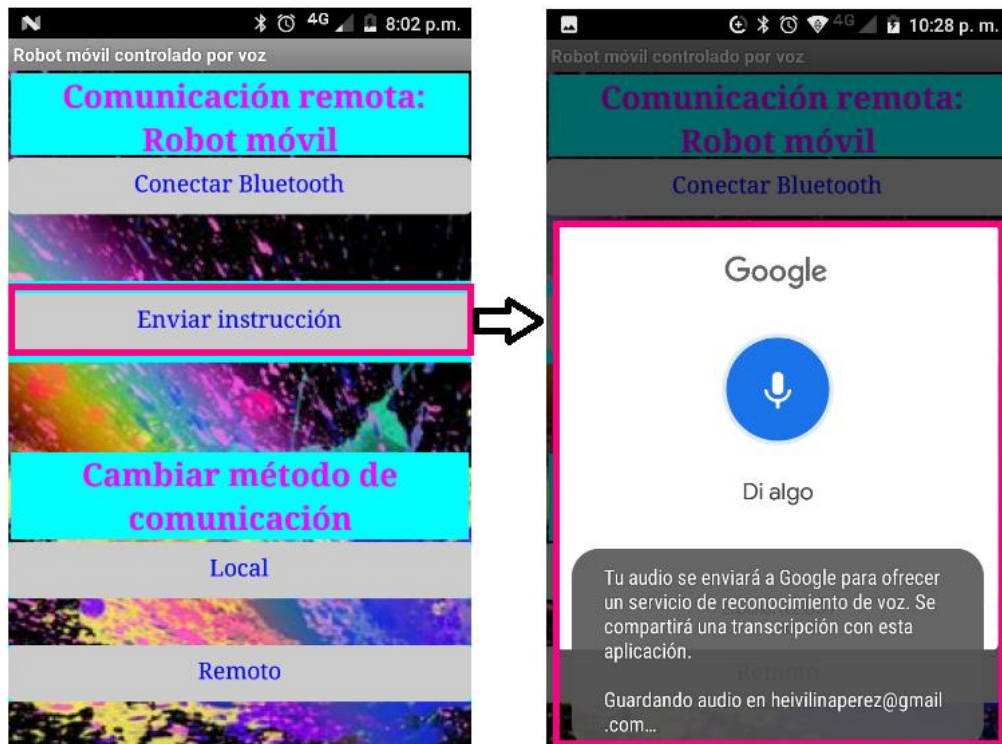


Figura 6.7. Acción de botón Enviar instrucción.

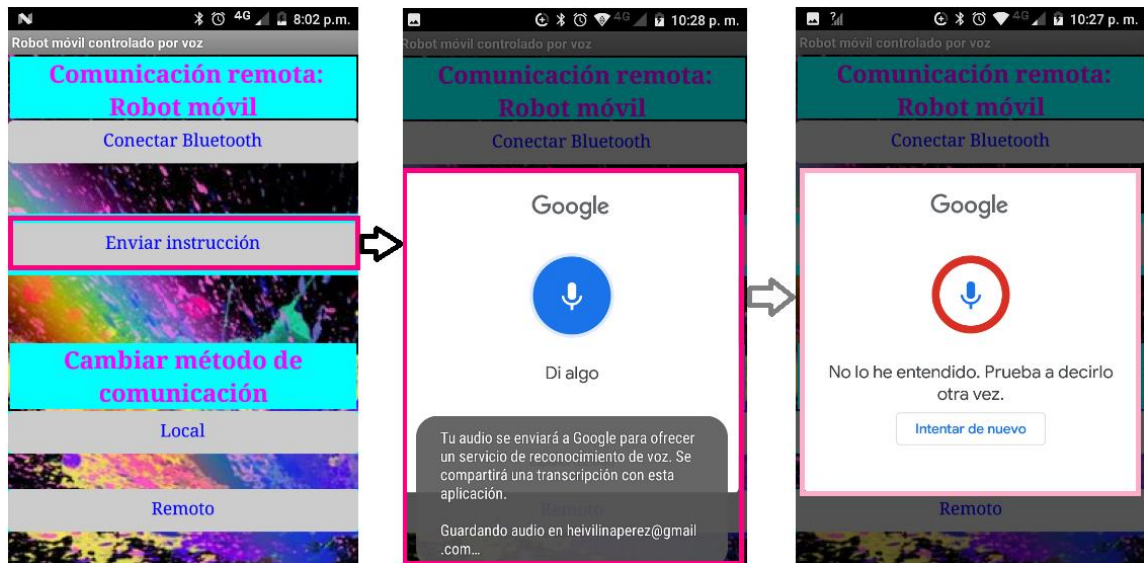
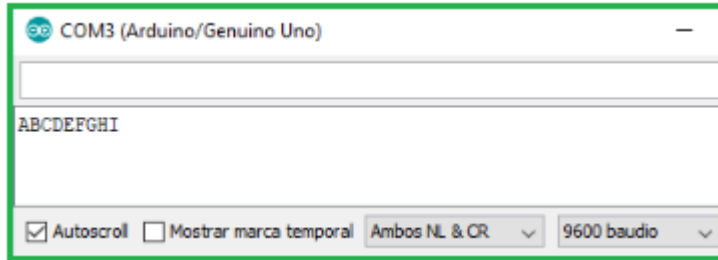


Figura 6.8. Reconocimiento de voz fallido.

Al iniciar la aplicación, es fundamental establecer la comunicación inalámbrica entre los dos dispositivos (teléfono móvil y módulo Bluetooth), posterior a ello pulsar el botón Enviar instrucción para activar el reconocimiento de voz; es el momento de hablar, al reconocer una palabra establecida como instrucción, se envía un carácter específico desde el teléfono al módulo a través del Bluetooth, por motivos de pruebas se visualizan los datos recibidos a través del módulo en el monitor serie del IDE de Arduino, en la Figura 6.9 se puede apreciar lo antes mencionado. Los caracteres fueron asignados de acuerdo con la Tabla 6.1.

Tabla 6.1. Relación: Variable/ instrucción

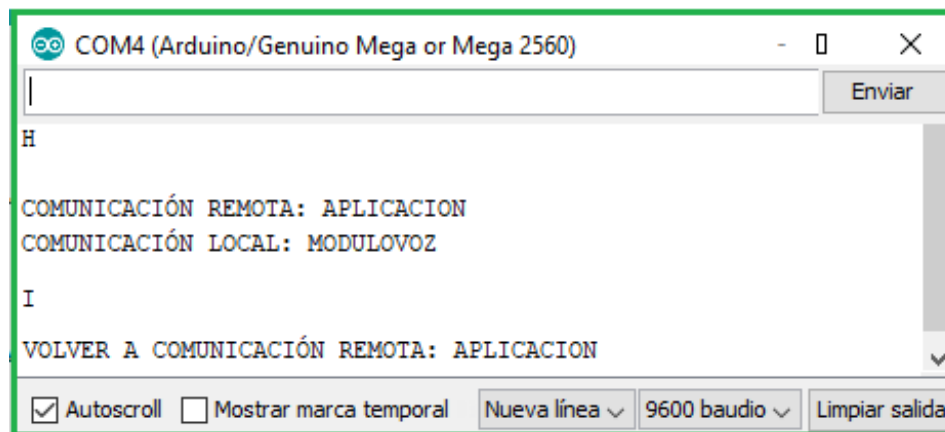
Palabra	Carácter	Palabra	Carácter
Inicia	A	Derecha	E
Enfrente	B	Giro	F
Reversa	C	Detente	G
Izquierda	D	Local	H
		Remota	I



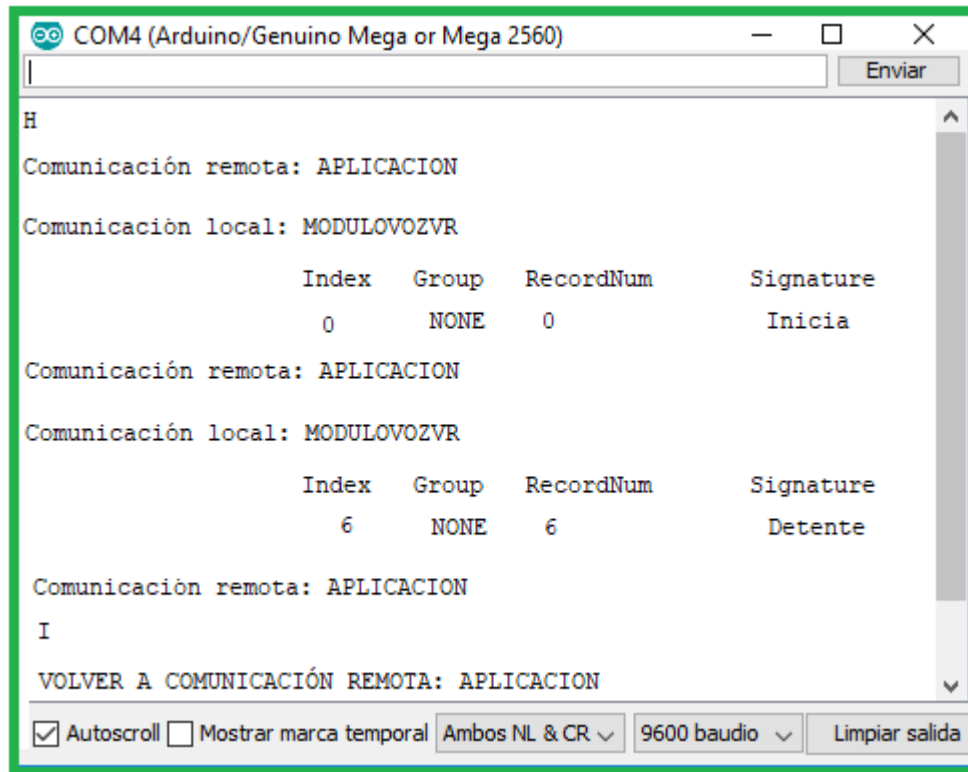
*Figura 6.9. Recepción de caracteres enviados desde la aplicación.*

El cambio del método para la manipulación del robot es una prueba más de la comunicación entre el Arduino y la Aplicación, realizada a partir del funcionamiento en conjunto del robot y es uno de los puntos principales de este proyecto. Cabe mencionar, que esta acción es ejecutada por el usuario haciendo envío de la instrucción desde los botones pertenecientes a la interfaz de la aplicación móvil.

La interacción entre Android y Arduino está presente desde que se inicia el funcionamiento del sistema del móvil. En la Figura 6.10, se muestran los mensajes impresos en el monitor serie capturados en el momento que se ejecuta el cambio en el método de manipulación para conducir el robot. Así mismo en la Figura 6.11, se observa, la especificación de cambio al método local, recepción de información de este modo y el regreso a la comunicación inicial; de manera remota. Las variables utilizadas para realizar este proceso son las mismas que se indican en la tabla anterior H; para iniciar la comunicación local, la letra I; para cancelarla y regresar al método remoto.



*Figura 6.10. Cambio de comunicación remota a local y viceversa.*



*Figura 6.11. Cambio al método local, recepción de datos y regreso a la comunicación remota.*

### 6.1.5 Funcionamiento inicial del robot móvil

Al tener conectados y configurados respectivamente los componentes del robot como, el sensor ultrasónico, el puente H y el servomotor con el Arduino, se obtiene como resultado el funcionamiento inicial del robot móvil. Es decir, el robot realiza el recorrido de una cierta trayectoria, y si se presenta un objeto, el robot es capaz de evadirlo, de manera satisfactoria, pero sin contar aun con las condiciones finales para su manipulación, como lo son, la dirección debe tomar y en qué momento ejecutarlo.

Para efectuar las pruebas correspondientes a esta fase, en la Figura 6.12 se muestra que, fue útil realizar el boceto de un pequeño rectángulo diseñado a partir de paredes de cartón, de tal manera que estas sirven para imitar un obstáculo a evadir, además de fungir como el contorno del espacio en el que se desplazó el móvil durante su recorrido.



*Figura 6.12. Funcionamiento inicial del robot móvil: Desplazamiento y evasión de obstáculos.*

Resulta útil resaltar el empleo de cada uno de los siguientes elementos: el puente H para el control de los motores de corriente directa, controlando la velocidad y dirección de estos, para el respectivo movimiento de las ruedas motorizadas. Para el módulo correspondiente a la evasión de obstáculos se hace uso del sensor ultrasónico y para que este tenga un mayor grado de alcance, se emplea el servomotor como base, permitiendo controlar la posición del eje en un momento dado.

En este caso, los componentes mencionados necesitan de una señal PWM para su funcionamiento, cabe mencionar que, entre estos difiere el ciclo de trabajo de la señal (cantidad de tiempo que la señal está en un estado lógico alto), porque la acción es distinta y trabajan en tiempos diferentes. En el código de la programación del Arduino, se define el ciclo que va a usar cierta terminal de salida del PWM. Esta señal fue utilizada para lograr el control de los motores de corriente directa mediante el puente H, encargándose de controlar la velocidad y dirección respectiva de movimiento. Además, para el funcionamiento del servomotor es indispensable hacer uso de dicha señal, con un ciclo de trabajo de un 50%, para indicar en qué posición debe ubicarse su eje, la duración del pulso establece el ángulo de giro.

### **6.1.6 Funcionamiento final del robot móvil**

En esta sección se presenta una serie de valoraciones realizadas para evaluar el desempeño del sistema de manipulación dual del móvil, haciendo uso de comandos de voz para su funcionamiento. El sistema se diseñó a partir de la integración de los dos métodos de comunicación abordados con anterioridad, más un algoritmo desarrollado con la finalidad de realizar la detección de objetos próximos para esquivarlos y cumplir el objetivo correspondiente a la evasión de obstáculos.

El proceso correspondiente a la evasión de obstáculos es ejecutado durante el recorrido de una trayectoria desde su punto de partida hasta su final, iniciando en el momento en que el móvil recibe una instrucción, sin tomar en cuenta cual es la opción seleccionada para el manejo del vehículo y cumpliendo una condición que indica que los comandos para ejecutar este proceso sean diferentes de: reversa y detente.

Para evaluar el desplazamiento del robot, durante las pruebas se optó por hacer uso de un contorno de 3 metros cuadrados, elaborado con paredes de cartón para delimitar el espacio en el que se diseñó un camino marcando las áreas de inicio y final de este. Dicho entorno se representa a partir de diagramas realizados como una herramienta para interpretar el desplazamiento del móvil, donde; con color azul se representa el camino o trayectoria optima y de un color distinto el recorrido realizado, cada color indica el método de comunicación utilizado para conducir el vehículo durante la realización de ciertas pruebas, siendo primero el remoto, seguido del local y finalmente cambio ente ambos.

A continuación, se observa una secuencia de imágenes representativas de las diferentes situaciones en las cuales el usuario dirige al móvil por medio de sus palabras, con el propósito de cumplir el trayecto establecido. Cada entorno cuenta con características distintas, como: la ruta a seguir, el método de comunicación empleado para la manipulación del robot y si durante el camino se presentan o no obstáculos. Se obtuvo un total de seis pruebas, mitad de estas con obstáculos y la otra mitad sin ninguno, finalmente se cumplió el recorrido de manera satisfactoria, las condiciones de cada una se detallan enseguida:



- a) Trayectoria deseada sin obstáculos en contraste con el recorrido del robot, mediante el método de manipulación remota.
- b) Trayectoria deseada con obstáculos en contraste con el recorrido del robot, mediante el método de manipulación remota.

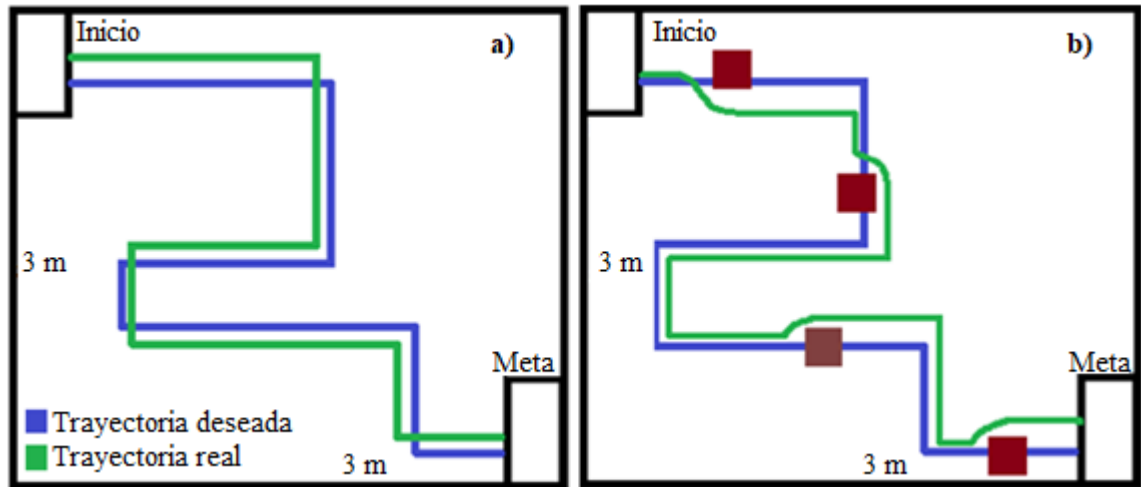


Figura 6.13. Recorrido de trayectoria sin / con obstáculos por el método remoto.

- a) Trayectoria deseada sin obstáculos en contraste con el recorrido del robot, mediante el método de manipulación local.
- b) Trayectoria deseada con obstáculos en contraste con el recorrido del robot, mediante el método de manipulación local.

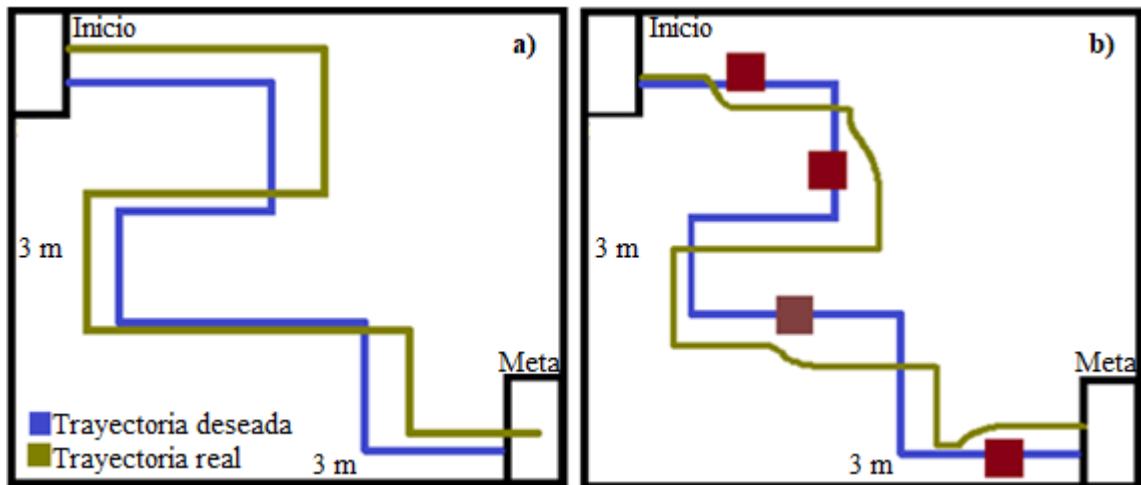


Figura 6.14. Recorrido de trayectoria sin / con obstáculos por el método local.

- a) Trayectoria deseada sin obstáculos en contraste con el recorrido del robot, haciendo los siguientes cambios en el método de manipulación: remoto / local /remoto.
- b) Trayectoria deseada con obstáculos en contraste con el recorrido del robot, haciendo los siguientes cambios en el método de manipulación: remoto / local /remoto.

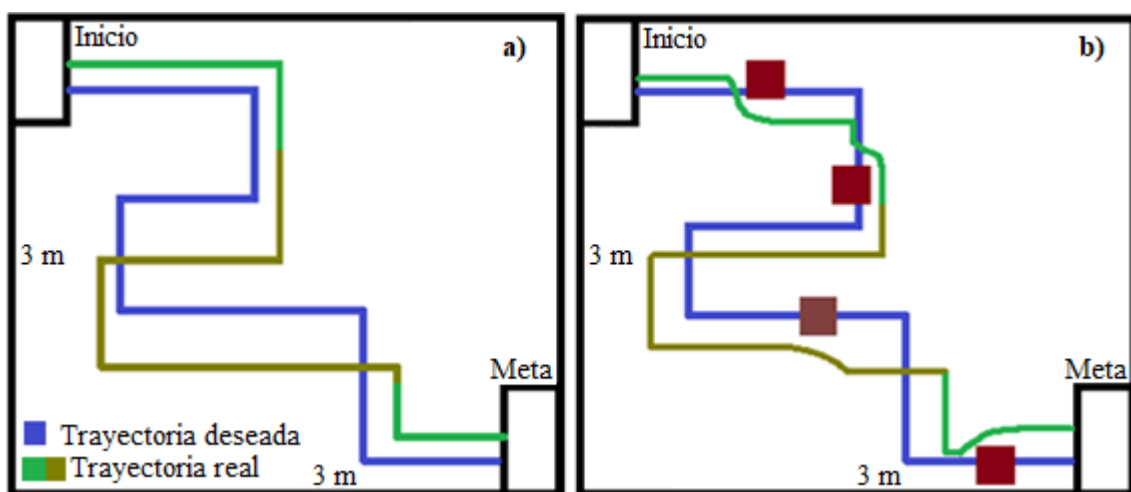


Figura 6.15. Recorrido de trayectoria sin / con obstáculos por el método dual.

Otra prueba realizada para evaluar el desempeño del sistema consiste en la comprobación del reconocimiento de comandos de voz durante la emisión de instrucciones. Se realizaron las pruebas que consisten en una serie de 40 repeticiones con cada una de las palabras que representan una instrucción para el robot. En la Tabla 6.2 se visualizan datos que indican el número de aciertos obtenidos y el porcentaje de eficacia del reconocimiento, estableciendo una relación entre los siete comandos y los dos métodos de comunicación.

Tabla 6.2. Datos sobre aciertos del reconocimiento de voz.

Método de comunicación	Parámetros	Comandos de voz						
		Inicia	Enfrente	Izquierda	Derecha	Reversa	Giro	Detente
Remoto	Aciertos	38	36	36	36	36	32	37
	Porcentaje	95%	90%	90%	90%	90%	80%	93%
Local	Aciertos	34	28	32	28	30	24	31
	Porcentaje	85%	70%	80%	70%	75%	60%	78%

Como se observa estas pruebas se realizaron con las siguientes siete palabras: inicia, enfrente, izquierda, derecha, reversa, giro y detente, que corresponden a las mismas instrucciones para los dos métodos de comunicación que integran el sistema de funcionamiento del robot móvil. Al obtener la información mencionada en la Figura 6.16 se presenta de manera gráfica para una mejor interpretación de los resultados.

Finalmente, se muestran los resultados obtenidos durante la última prueba que también consiste en la comprobación del reconocimiento de patrones de voz durante la emisión de instrucciones, pero, en esta se realizó únicamente con los comandos: Inicia y detente con la condición de poner distancia de por medio entre el robot y el usuario, haciendo uso de los dos métodos de comunicación establecidos. Para obtener los datos observados en la Tabla 6.3 se pasó por una serie de 10 repeticiones con cada una de las palabras y la distancia fue variando entre uno y siete metros como se muestra en la misma.

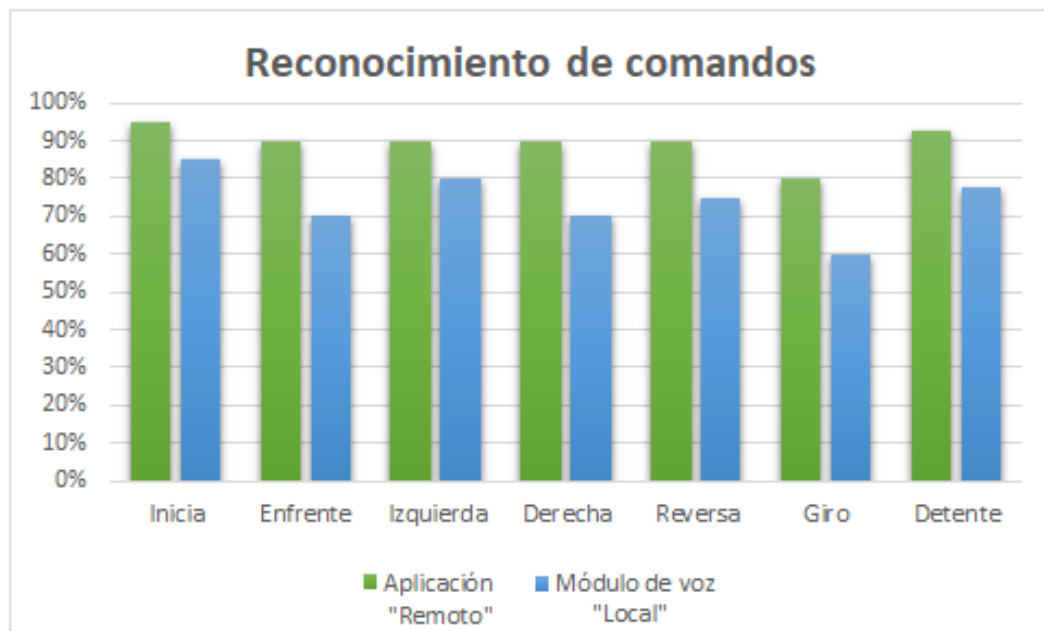


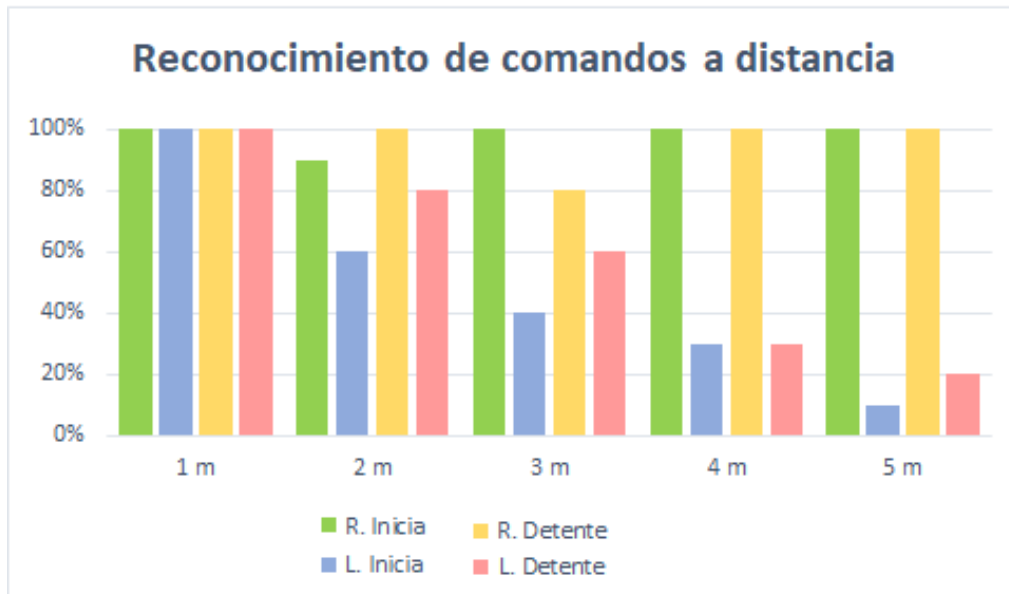
Figura 6.16. Porcentaje de aciertos en pruebas de reconocimiento de comandos de voz.

Tabla 6.3. Datos de reconocimiento de comandos a distancia.

Método	Comando	Parámetros	Valores				
		Distancia	1 m	2 m	3 m	4 m	5 m
Remoto	Inicia	Aciertos	10	9	10	10	10
		Porcentaje	100%	90%	100%	100%	100%
Local	Inicia	Aciertos	9	8	7	3	1
		Porcentaje	100%	60%	40%	30%	10%
Remoto	Detente	Aciertos	10	10	9	10	10
		Porcentaje	100%	100%	80%	100%	100%
Local	Detente	Aciertos	10	8	6	3	2
		Porcentaje	100%	80%	60%	30%	20%

Teniendo en cuenta la información anterior se pudo observar que los resultados correspondientes a la comunicación local fueron variando de acorde a la distancia presentada, donde; a los cinco metros fue mínimo el porcentaje de reconocimiento que se logró. Por otro lado, al hacer uso de la manipulación remota no se presentó ningún cambio drástico en el número de aciertos obtenidos desde el primer hasta el tercer metro se conserva una comunicación estable debido a que, en este método las instrucciones son emitidas al robot desde un teléfono móvil por medio de la tecnología Bluetooth que permite un alcance aproximado de diez metros. En la Figura 6.17 se visualiza la representación de forma gráfica de los datos mencionados donde; se indican los valores que corresponden al porcentaje de aciertos obtenidos con relación a la distancia en metros, para cada comando verificado con cada uno de los dos métodos de comunicación utilizados.

Con los datos representados estadísticamente se observó que el número de aciertos y porcentaje alcanzado depende del comando pronunciado, además del método de comunicación empleado durante la etapa de las pruebas correspondientes. Por tal motivo, como se reportó anteriormente; se optó por realizar una comparación entre los comandos “inicia” y “detente”, haciendo uso en ambos casos de las dos formas de manipulación con las que cuenta el móvil, además incrementar la distancia entre el usuario y el robot.



*Figura 6.17. Número de aciertos en reconocimiento de comandos a distancia para cada uno de los métodos de comunicación.*

## CONCLUSIONES

El principal objetivo de este proyecto de tesis consistió en la implementación de medios para el reconocimiento de voz durante la manipulación de un robot móvil con locomoción por ruedas. Para ello, se desarrolló un sistema que permite el manejo del dispositivo de dos distintas maneras: a) local, con la integración de un módulo de reconocimiento de voz instalado en el robot, y b) remota, mediante una aplicación móvil específicamente desarrollada para la comunicación vía Bluetooth entre Android y Arduino. De acuerdo con los resultados obtenidos, este objetivo se cumplió satisfactoriamente, permitiendo la manipulación dual del robot móvil, e incluso permitiendo alternar entre el uso de uno u otro modo de manipulación durante el recorrido del robot en una trayectoria regular delimitada. Independiente de la forma de manejo seleccionada, el vehículo fue capaz de terminar un recorrido establecido a través del uso de comandos de voz.

Además, mediante la inclusión de un sensor ultrasónico de distancia en la parte frontal del robot móvil, se proveyó de un medio para la evasión de obstáculos durante su trayectoria, lo que mejora su autonomía. El sensor fue colocado sobre el eje de un servomotor, que se posiciona automáticamente en tres diferentes ángulos, con la finalidad de abarcar de manera amplia el entorno y, al momento en que se detecta un objeto que impide el paso del robot, este es capaz de evadirlo y continuar con su recorrido.

Durante la etapa de pruebas, se observó al utilizar el módulo de reconocimiento de voz como medio local de comunicación, algunas palabras como “inicia” e “izquierda”, presentan un mayor porcentaje de reconocimiento que el resto de los comandos. Sin embargo, la palabra “giro” reportó el peor índice de reconocimiento. Esto se atribuye a que la entonación y complejidad sonora de la palabra afectan el entrenamiento del módulo de reconocimiento de voz, además de que la distancia del usuario al robot también afecta la detección adecuada del comando durante el recorrido. La manipulación remota, mediante el uso del reconocimiento de voz de Google, representa un mejor método para la manipulación del robot, en tanto que este se encuentre dentro del alcance de comunicación de este medio, estimado en aproximadamente 10 metros.

Cabe mencionar que, para el uso del método local, el entrenamiento de los comandos consiste en la captura y almacenamiento del sonido con ayuda de un módulo de

reconocimiento de voz, que deberá ser comparado dato a dato con cada instrucción previamente grabada, lo que dificulta relativamente el reconocimiento de las palabras. En contraste, en la comunicación en modo remoto, donde se hace uso del reconocimiento de voz de Google, el entrenamiento consiste en la captura, almacenamiento y procesamiento de datos con técnicas de inteligencia artificial como son las redes neuronales profundas más el uso del modelo oculto de Márkov, lo que le confiere un mejor desempeño.

Cabe destacar que, para el desarrollo del prototipo descrito en el presente documento, se siguió la metodología de desarrollo en espiral, que resultó muy útil para este fin al integrar elementos de hardware y software en el mismo proyecto.

## REFERENCIAS

Aguilera Hernández, M., Bautista, M. A. & Joaquín, I., 2017. *Diseño y control de robots móviles*, Reforma, p. 7.

Andés, A. C., 2011. *Coordinación y control de robots móviles basados en agentes*. Valencia: Tesis de Licenciatura, Universidad Politécnica de Valencia.

ARDUINO.cl, 2015. *ARDUINO MEGA 2560*. [En línea] Disponible en: <https://arduino.cl/arduino-mega-2560/> [Último acceso: 22 Enero 2019].

Asimov, I., 1950. *I Robot*. Barcelona: Edhasa.

Baliñas, J., Bronte, S., García, J. & Arranz, A., 2006. *Aplicaciones de robots móviles*. Reforma, p. 8.

Baquero Romero, Y., Borrero-Guerrero, H. & Alezones Campos, Z., 2012. *Reconocimiento de palabras aisladas para control de navegación de robot móvil*. [En línea] Disponible en: <https://dialnet.unirioja.es/servlet/articulo?codigo=4027705> [Último acceso: 10 Septiembre 2018].

Barrientos Sotelo, V. R., García Sánchez, J. R. & Silva Ortigoza, R., 2007. *Robots Móviles: Evolución y Estado del Arte*. Polibits, Issue 35, p. 17.

Bermudez, G., 2015. *Robots móviles: Teoría y aplicaciones*. Tecnura, Issue 10, p. 13.

Boada, M., Barber, R., Egido, V. & Salichs, M., 2005. *Sistema de navegación por voz para robots móviles autónomos*. Iberoamericana de Ingeniería Mecánica, 10(1), p. 18.

Cánovas, J. A. G., 2011. *Robótica móvil. Estudio y caracterización del robot móvil KJUNIOR, Desarrollo de una aplicación de robot laberinto*. Cartagena: Tesis de Licenciatura.

Cardenas, E. O. G., 2018. *Control de mando por voz para personas con discapacidad motriz*. Bogotá: Tesis de licenciatura: Universidad Católica de Colombia, Facultad de Ingeniería: Ingeniería en Electrónica y telecomunicaciones.

Carretero, O. S., 2016. *Casa domótica con Arduino*. Valencia: Tesis de licenciatura: Ingeniería Electrónica Industrial y Automática, Universidad Politécnica de Valencia.



Costumero, C. C., 2016. *Desarrollo de un sistema de reconocimiento de habla natural basado en redes neuronales profundas*. Madrid: Tesis de licenciatura: Ingeniería en Telecomunicación, Universidad Autónoma de Madrd.

Cui, B. & Xue, T., 2009. Design and realization of an intelligent access control system based on voice recognition. *ISECS Coloquio Internacional de Computación*, Volumen 1, p. 4.

Dignani, J. P., 2011. *Análisis del protocolo ZigBee*. La Plata: Tesis de Licenciatura: Facultad de Informática.

ELECTRONILAB: Ingeniería y Diseño Electrónico, 2019. *Sensor de distancia de ultrasonido HC-SR04*. [En línea] Disponible en: <https://electronilab.co/tienda/sensor-de-distancia-de-ultrasonido-hc-sr04/> [Último acceso: 23 Enero 2019].

Fields, L. E., 2015. Comparación de sistemas de comunicación inalámbrica para robots móviles. *Actualidad tecnológica*, p. 22.

Geekfactory.mx, 2014. *Bluetooth HC-05 y HC-06*. [En línea] Disponible en: <https://www.geekfactory.mx/tutoriales/bluetooth-hc-05-y-hc-06> [Último acceso: 18 Abril 2019].

Gonzalez Vidal, J. C., 2013. *Desarrollo de un servidor web con Arduino para monitorización y control de sensores y actuadores*. Cartagena: Proyecto Fin de Carrera. Universidad Politécnica de Cartagena.

González, R., Rodríguez, F. & Guzmán, J. L., 2015. *Robots móviles con Orugas: Historia, modelado, localización y control*. Iberoamericana de Automática e Informática Industrial, Issue 12, p. 12.

Google, 2015. *Diseño de apps con MIT App Inventor 2*. [En línea] Disponible en: <http://repositorio.puce.edu.ec/bitstream/handle/22000/13113/Anexo%203%20MIT%20App%20Inventor%202.pdf?sequence=4&isAllowed=y> [Último acceso: 23 Enero 2019].

Hernández, R., 2016. *Sistema de control activado por voz para uso en domótica*. Xalapa Veracruz: Tesis de Maestría en Ingeniería Electrónica y Computacón: Universidad Veracruzana.

IEEE, 2019. *Advancing Technology for Humanity*. [En línea] Disponible en: <https://www.ieee.org/> [Último acceso: 20 Febrero 2019].

Lima, R. C. P., 2011. *Sistema de exploración de terrenos con robots móviles: Aplicación en tareas de detección y localización de minas antipersonas..* Madrid: Tesis Doctoral: Universidad Complutense de Madrid.

Linares Ruiz, R., Quijano Vásquez, J. A. & Holguin Londoño, G. A., 2004. *Implementación del protocolo Bluetooth para la conexión inalámbrica de dispositivos electrónicos programables*. Scientia Et Technica, 10(24), p. 7.

Llamas, L., 2015. *Ingeniería, informática y diseño*. [En línea] Disponible en: <https://www.luisllamas.es/medir-distancia-con-arduino-y-sensor-de-ultrasonidos-hc-sr04/> [Último acceso: 02 Septiembre 2019].

López, J. R., 2015. *Sistema de comunicación inalámbrico basado en el protocolo ZigBee*. Madrid: Tesis de licenciatura: Ingeniería de Sistemas de Comunicaciones.

Lucas, J. T., 2017. *Machine Learning: Modelos Ocultos de Markov (HMM) y Redes Neuronales Artificiales (ANN)*. Barcelona: Tesis de licenciatura: Facultad de Matemáticas e Informática, Universidad de Barcelona.

M.Castillo, 2016. *El módulo controlador de motores l298n*. [En línea] Disponible en: <http://www.electrontools.com/Home/WP/2016/05/06/puente-h-con-driver-l298/>. [Último acceso: 25 Enero 2019].

Mangado, A. P. d. L., 2017. *Robot Arduino controlado mediante sensores y con respuestas sobre actuadores*. Tesis de licenciatura: Tecnologías de Telecomunicación.

Moumtadi, F., Granados, F. & Delgado, J. C., 2014. *Activación de funciones en edificios inteligentes utilizando comandos de voz desde dispositivos móviles*. Ingeniería, Investigación y Tecnología, XV(2), pp. 3-13.

Obregón, G. M. R., 2013. *Control inalámbrico de un robot móvil bajo la plataforma Android*. México: Tesis de Licenciatura: Instituto Politécnico Nacional.

Perez Pimentel, Y., Osuna Galán, I. & Ibarra Orozco, R., 2013. *Reconocimiento de voz usando redes neuronales para el control de una silla de ruedas*. ResearchGate, p. 17.

Pressman, R. S., 2010. *Ingeniería del software: Un enfoque práctico*. Séptima ed. México: Mc Graw Hill.

PROMETEC, 2014. *Reconocimiento de voz V3*. [En línea] Disponible en: <https://www.prometec.net/producto/modulo-reconocimiento-voz/#> [Último acceso: 22 Enero 2019].

Ramírez, J. E. R., 2017. *Robot móvil tipo (2,1) con rueda articulada RRRW*. México: Tesis de Licenciatura, Universidad Nacional Autónoma de México.

Reyes, F., 2011. *Robótica control de robots Manipuladores*. México: AlfaOmega.

Rincón Llorent, C., 2007. *Diseño, implementación y evaluación de técnicas de identificación de emociones a través de la voz*. Madrid: Tesis de licenciatura: Universidad Politécnica de Madrid.

Rivas, C. I., Corona, V. P. & Hernández, L., 2015. *Metodologías actuales de desarrollo de software*. Revista Tecnología e Innovación, 2(5), p. 7.

Sanz Leon, P. & Vera de Payer, E., 2015. *Reconocimiento de comandos de voz aplicado a sistema robótico médico*. ResearchGate, p. 7.

Shmyrev, N., 2016. *Quora: ¿Qué algoritmo de reconocimiento de voz utiliza Google?*. [En línea] Disponible en: <https://www.quora.com/What-speech-recognition-algorithms-are-used-by-Google> [Último acceso: 11 Julio 2019].

Simancas, J. & Meléndez Pertuz, F., 2017. *Algoritmo de reconocimiento de comandos de voz basado en técnicas no-lineales*. Espacios, 38(17), p. 19.

Sung Won, J., Ki Won, S., Moon Youl, P. & Eon Uck, K., 2013. *A Study on precise control of autonomous driving robot by voice recognition*. Departamento de Ingeniería Avanzada, Volumen 1, p. 3.

Suntaxi Llumiyinga, M., 2015. *Diseño e implementación de un Sistema de navegación por voz para un robot móvil con ruedas (RMR) utilizando la plataforma de desarrollo*

*Microsoft Robotics Developer Studio*. Tesis de licenciatura: Universidad de las Fuerzas Armadas ESPE, Ingeniería en Electrónica Automatización y Control.

Tepán, E. C. A., 2013. *Estudio de los principales tipos de Redes Neuronales y las herramientas para su aplicación*. Cuenca: Tesis de licenciatura: Ingeniería de Sistemas, Universidad Politécnica Salesiana Sede Cuenca.

Torres, M. A. G., 2009. *Inteligencia Artificial en robots móviles de competencia para su aplicación en el diseño de un agente inteligente en un robot seguidor de trayectorias*. Instituto Politécnico Nacional ed. México: Tesis de Licenciatura.

Zhen, L., Gao, W., Lina, Y. & Ying, Y., 2010. *Desing of voice service system for agricultural keywords recognition*. Universidad Agrícola de China, Volumen 1, p. 6.

## ANEXOS

### Código Fuente: Aplicación.

- Representación de la programación por bloques, correspondiente a la Aplicación: “Mi robot móvil”.

```
when conexión .BeforePicking
do
  if BluetoothClient1 . Available
  then set conexión . Elements to BluetoothClient1 . AddressesAndNames

when conexión .AfterPicking
do
  evaluate but ignore result call BluetoothClient1 .Connect
  address conexión . Selection

when Local .Click
do
  call BluetoothClient1 .SendText
  text " H "

when Remota .Click
do
  call BluetoothClient1 .SendText
  text " I "
```

The image displays four distinct blocks of code in a block-based programming language. Each block begins with a 'when' event trigger and a 'do' action block. The first block triggers on 'conexión .BeforePicking' and contains an 'if' statement: if 'BluetoothClient1 . Available' is true, then 'set conexión . Elements to BluetoothClient1 . AddressesAndNames'. The second block triggers on 'conexión .AfterPicking' and contains an 'evaluate but ignore result' block followed by a 'call BluetoothClient1 .Connect' block, where the 'address' parameter is 'conexión . Selection'. The third block triggers on 'Local .Click' and contains a 'call BluetoothClient1 .SendText' block with the 'text' parameter set to '" H "'. The fourth block triggers on 'Remota .Click' and contains a 'call BluetoothClient1 .SendText' block with the 'text' parameter set to '" I "'.

```
when hablar .Click
do call SpeechRecognizer1 .GetText

when SpeechRecognizer1 .AfterGettingText
result partial
do if
    SpeechRecognizer1 . Result = "inicia "
then call BluetoothClient1 .SendText
    text " A "
else if
    SpeechRecognizer1 . Result = "enfrente "
then call BluetoothClient1 .SendText
    text " B "
else if
    SpeechRecognizer1 . Result = "reversa "
then call BluetoothClient1 .SendText
    text " C "
else if
    SpeechRecognizer1 . Result = "izquierda "
then call BluetoothClient1 .SendText
    text " D "
else if
    SpeechRecognizer1 . Result = "derecha "
then call BluetoothClient1 .SendText
    text " E "
else if
    SpeechRecognizer1 . Result = "giro "
then call BluetoothClient1 .SendText
    text " F "
else if
    SpeechRecognizer1 . Result = "detente "
then call BluetoothClient1 .SendText
    text " G "
```

## Código Fuente: Software Arduino

- Algoritmo de funcionamiento y manipulación dual del robot móvil.

```
//librería: puertos seriales en terminales digitales
#include <SoftwareSerial.h>

//librería: funciones del módulo de voz
#include "VoiceRecognitionV3.h"

//librería: funciones del del servomotor
#include <Servo.h>

//puertos de comunicacion serial (Rx, Tx)
SoftwareSerial APLICACION(10, 11); //com serial puerto 1
VR MODULOVOZ(12, 13);           //com serial puerto 2

//guarda registro de 8 bits o 1 byte sin signo
uint8_t records[7];

//definicion de campo un datos, máximo 64 bytes
uint8_t buf[64];

//definir palabras a utilizar en el módulo de voz
#define Inicia (0)
#define Enfrente (1)
#define Reversa (2)
#define Izquierda (3)
#define Derecha (4)
#define Giro (5)
#define Detente (6)

//definir pines y variables para el sensor ultrasónico
#define Pin_trig 4
#define Pin_echo 3
long tiempo;
```

```

long distancia;
int cm;
int distancia_derecha;
int distancia_izquierda;
int distancia_maxima = 30;
int distancia_minima = 15;

//crear objeto servo
Servo servo;
const int pwm = 7; //pwm del servo

//definir pines a usar para los motores
int motorDer1 = 8; //El pin 8 a In1 del puente H
int motorDer2 = 24; //El pin 24 a In2 del puente H
int motorIzq1 = 22; //El pin 22 a In3 del puente H
int motorIzq2 = 9; //El pin 9 a In4 del puente H
int ruedaIzquierda = 5; //El pin 5 a EnA del puente H
int ruedaDerecha = 6; //El pin 6 aEnB del puente H

void setup() {
//establecer comunicación serial
  Serial.begin(9600);
  MODULOVOZ.begin(9600); //Módulo de reconocimiento de voz
  APLICACION.begin(38400); //Módulo Bluetooth

//conectar el servomotor al pin 7 = pwm
  servo.attach(pwm);
  servo.write(90); //ángulo de inicio
  delay(3000);

```



```

//declarar pines del Arduino como entradas o salidas
pinMode(Pin_echo, INPUT);
pinMode(Pin_trig, OUTPUT);
pinMode(motorDer1, OUTPUT);
pinMode(motorDer2, OUTPUT);
pinMode(motorIzq1, OUTPUT);
pinMode(motorIzq2, OUTPUT);
pinMode(ruedaIzquierda, OUTPUT);
pinMode(ruedaDerecha, OUTPUT);
}

//Método principal
void loop() {
//inicia comunicacion serial con bluetooth
if (APLICACION.available() > 0) {
    //lee caracter y lo almacena en variable
    char variable = APLICACION.read();
    Serial.println(variable);

    //Comparación de instrucciones con caracteres para accionar el robot
    //De acuerdo a la variable se manda llamar un método correspondiente
    if (variable == 'A') { //Si el dato que lee es A: inicia
        inicia();
        obstaculos1();
    }
    else
    if (variable == 'B') { //Si el dato que lee es B: enfrente
        enfrente();
        obstaculos1();
    }
}
}

```

```

else
if (variable == 'C') { //Si el dato que lee es C: atrás
    atras();
}
else
if (variable == 'D') { //Si el dato que lee es D: izquierda
    izquierda();
    obstaculos1();
}
else
if (variable == 'E') { //Si el dato que lee es E: derecha
    derecha();
    obstaculos1();
}
else
if (variable == 'F') { //Si el dato que lee es F:giro
    girar();
    obstaculos1();
}
else
if (variable == 'G') { //Si el dato que lee es G:detente
    detener();
}
else

while (variable == 'H') { //mientras el dato que lee es H: local
    Serial.println();
    //escucha el serial del módulo Bluetooth
    APLICACION.listen();
    Serial.write("Comunicación remota: APLICACION");
    delay(3000);
}

```

```
if (APLICACION.available() > 0) { //si la aplicacion esta disponible y tiene datos
    char variable = APLICACION.read(); //lee caracter y almacena en variable
    Serial.println(variable);
```

```
    if (variable == 'I') { //Si el dato que lee es I: remota
        Serial.write("Cambio: volver a comunicación remota: APLICACION");
        break; // rompe el ciclo y continuar con el programa
    }
}
```

```
Serial.println();
//escucha el serial del módulo de reconocimiento de voz
MODULOVOZ.listen();
Serial.write("Comunicación local: MODULOVOZ");
delay(5000);
```

```
if (MODULOVOZ.available() > 0) { //si el módulo esta disponible y tiene datos
    //condicionar la palabra que se ha reconocido para accionar el robot
    //de acuerdo con cada caso se manda llamar el método correspondiente
    int palabra = MODULOVOZ.recognize(buf, 50);
```

```
    if (palabra > 0) {
        switch (buf[1]) {
            case Inicia:
                inicia();
                obstaculos2();
                break;

            case Enfrente:
```

```
enfrente();
obstaculos2();
break;

case Reversa:
    atras();
    break;

case Izquierda:
    izquierda();
    obstaculos2();
    break;

case Derecha:
    derecha();
    obstaculos2();
    break;

case Giro:
    girar();
    obstaculos2();
    break;

case Detente:
    detener();
    break;

default:
    Serial.println("Funcion indefinida");
    break;
```

```

    }
    printVR(buf);

} //fin del if del reconocimiento de palabras
Serial.println();

} //fin de la disponibilidad del módulo de voz
Serial.println();

} //fin del ciclo while de H == LOCAL

} //fin de la disponibilidad de la aplicacion

} //fin del metodo prinipal

//Función para calcular distancia en centímetros
int Centimetros() {
    digitalWrite(Pin_trig, LOW);
    delayMicroseconds(2);
    digitalWrite(Pin_trig, HIGH); //pulso del trigger durante 10 Ms
    delayMicroseconds(10);
    digitalWrite(Pin_trig, LOW);
    tiempo = pulseIn(Pin_echo, HIGH);
    distancia = (float(tiempo)) / 53; //calcular la distancia
    delay(5);
    return distancia; //retorna la distancia
}

```

```
//Métodos para los movimientos del robot
```

```
void inicia() {  
    digitalWrite(motorDer1, LOW);  
    digitalWrite(motorDer2, HIGH);  
    digitalWrite(motorIzq1, HIGH);  
    digitalWrite(motorIzq2, LOW);  
    analogWrite(ruedaIzquierda, 100);  
    analogWrite(ruedaDerecha, 100);  
    delay(300);  
}
```

```
void enfrente() {  
    digitalWrite(motorDer1, LOW);  
    digitalWrite(motorDer2, HIGH);  
    digitalWrite(motorIzq1, HIGH);  
    digitalWrite(motorIzq2, LOW);  
    analogWrite(ruedaIzquierda, 100);  
    analogWrite(ruedaDerecha, 100);  
    delay(300);  
}
```

```
void atras() {  
    digitalWrite(motorDer1, HIGH);  
    digitalWrite(motorDer2, LOW);  
    digitalWrite(motorIzq1, LOW);  
    digitalWrite(motorIzq2, HIGH);  
    analogWrite(ruedaIzquierda, 100);  
    analogWrite(ruedaDerecha, 100);  
    delay(300);  
}
```

```
void izquierda() {  
    digitalWrite(motorDer1, LOW);  
    digitalWrite(motorDer2, HIGH);  
    digitalWrite(motorIzq1, HIGH);  
    digitalWrite(motorIzq2, LOW);  
    analogWrite(ruedaIzquierda, 80);  
    analogWrite(ruedaDerecha, 160);  
    delay(1200);  
    digitalWrite(motorDer1, LOW);  
    digitalWrite(motorDer2, HIGH);  
    digitalWrite(motorIzq1, HIGH);  
    digitalWrite(motorIzq2, LOW);  
    analogWrite(ruedaIzquierda, 100);  
    analogWrite(ruedaDerecha, 100);  
    delay(300);  
}
```

```
void derecha() {  
    digitalWrite(motorDer1, LOW);  
    digitalWrite(motorDer2, HIGH);  
    digitalWrite(motorIzq1, HIGH);  
    digitalWrite(motorIzq2, LOW);  
    analogWrite(ruedaIzquierda, 160);  
    analogWrite(ruedaDerecha, 80);  
    delay(1000);  
    digitalWrite(motorDer1, LOW);  
    digitalWrite(motorDer2, HIGH);  
    digitalWrite(motorIzq1, HIGH);  
    digitalWrite(motorIzq2, LOW);  
    analogWrite(ruedaIzquierda, 100);  
    analogWrite(ruedaDerecha, 100);
```

```

    delay(300);
}

void girar() {
    digitalWrite(motorDer1, LOW);
    digitalWrite(motorDer2, HIGH);
    digitalWrite(motorIzq1, HIGH);
    digitalWrite(motorIzq2, LOW);
    analogWrite(ruedaIzquierda, 60);
    analogWrite(ruedaDerecha, 220);
    delay(1000);
    digitalWrite(motorDer1, LOW);
    digitalWrite(motorDer2, HIGH);
    digitalWrite(motorIzq1, HIGH);
    digitalWrite(motorIzq2, LOW);
    analogWrite(ruedaIzquierda, 100);
    analogWrite(ruedaDerecha, 100);
    delay(300);
}

void detener() {
    digitalWrite(motorDer1, LOW);
    digitalWrite(motorDer2, HIGH);
    digitalWrite(motorIzq1, HIGH);
    digitalWrite(motorIzq2, LOW);
    analogWrite(ruedaIzquierda, 0);
    analogWrite(ruedaDerecha, 0);
}

```

//Método printSignature para procesar la voz



```

void printSignature(uint8_t *buf, int len) {
    int i;
    for (i = 0; i < len; i++) {
        if (buf[i] > 0x19 && buf[i] < 0x7F) {
            Serial.write(buf[i]);
        }
        else
        {
            Serial.print("[");
            Serial.print(buf[i], HEX);
            Serial.print("]");
        }
    }
}

//Método printVR para procesar la voz
void printVR(uint8_t *buf) {
    Serial.println("VR Index\tGroup\tRecordNum\tSignature");
    Serial.print(buf[2], DEC);
    Serial.print("\t\t");

    if (buf[0] == 0xFF) {
        Serial.print("NONE");
    }

    else if (buf[0] & 0x80) {
        Serial.print("UG ");
        Serial.print(buf[0] & (~0x80), DEC);
    }
    else

```

```

{
  Serial.print("SG ");
  Serial.print(buf[0], DEC);
}
Serial.print("\t");

Serial.print(buf[1], DEC);
Serial.print("\t\t");

if (buf[3] > 0) {
  printSignature(buf + 4, buf[3]);
}
else
{
  Serial.print("NONE");
}
Serial2.println("\r\n");
}

//Métodos para la evasión de obstáculos
//comunicación remota
void obstaculos1() {
  while (APLICACION.read() != 'T') {
    cm = Centimetros(); //lee la funcion de la distancia en centimetros

    //verifica si hay obstaculos en el intervalo de distancia 15-30 cm
    if (cm <= distancia_maxima && cm >= distancia_minima) {
      digitalWrite(motorDer1, LOW);
      digitalWrite(motorDer2, HIGH);
      digitalWrite(motorIzq1, HIGH);
    }
  }
}

```

```

digitalWrite(motorIzq2, LOW);
analogWrite(ruedaIzquierda, 0);
analogWrite(ruedaDerecha, 0);
Serial.println(Centimetros());

servo.write(25); //posición del servomotor a la derecha
delay(800);
distancia_derecha = Centimetros(); //guarda la distancia de la derecha

digitalWrite(motorDer1, LOW);
digitalWrite(motorDer2, HIGH);
digitalWrite(motorIzq1, HIGH);
digitalWrite(motorIzq2, LOW);
analogWrite(ruedaIzquierda, 0);
analogWrite(ruedaDerecha, 0);

servo.write(175); //posición del servomotor a la izquierda
delay(800);
distancia_izquierda = Centimetros(); //guarda la distancia de la izquierda

//si la distancia de la izquierda es mayor a la de la derecha
//indica que hay obstáculo en la derecha
if (distancia_izquierda > distancia_derecha) {
    servo.write(90); //posición del servomotor enfrente
    delay(800);
    digitalWrite(motorDer1, LOW);
    digitalWrite(motorDer2, HIGH);
    digitalWrite(motorIzq1, HIGH);
    digitalWrite(motorIzq2, LOW);
    analogWrite(ruedaIzquierda, 60);
    analogWrite(ruedaDerecha, 180);
}

```

```

delay(300);
Serial.write("obstáculo a la derecha"); //girar a la izquierda
}

//si la distancia de la derecha es mayor a la de la izquierda
//indica que hay obstáculo en la izquierda
if (distancia_izquierda < distancia_derecha) {
    servo.write(90);//posición del servomotor enfrente
    delay(800);
    Serial.write("obstáculo a la izquierda"); //girar a la derecha
    digitalWrite(motorDer1, LOW);
    digitalWrite(motorDer2, HIGH);
    digitalWrite(motorIzq1, HIGH);
    digitalWrite(motorIzq2, LOW);
    analogWrite(ruedaIzquierda, 180);
    analogWrite(ruedaDerecha, 60);
    delay(300);
}
}

else
{ //si no se detecta obstaculo, continuar la trayectoria
    Serial.write("continua, sin obstáculos");
    digitalWrite(motorDer1, LOW);
    digitalWrite(motorDer2, HIGH);
    digitalWrite(motorIzq1, HIGH);
    digitalWrite(motorIzq2, LOW);
    analogWrite(ruedaIzquierda, 100);
    analogWrite(ruedaDerecha, 100);
    delay(300);
} //fin del else de los obstaculos

```

```

    } //fin del while

} //fin de metodo evasión de obstáculos1

//Método para la evasión de obstáculos
//comunicación local
void obstaculos2() {
    int palabra = MODULOVOZ.recognize(buf, 50);
    while (palabra != Detente || palabra != Reversa) {
        Serial.println("inicia met 2");
        cm = Centimetros(); //lee la funcion de la distancia en centimetros

//verifica si hay obstaculos en el intervalo de distancia 15-30 cm
        if (cm <= distancia_maxima && cm >= distancia_minima) {
            digitalWrite(motorDer1, LOW);
            digitalWrite(motorDer2, HIGH);
            digitalWrite(motorIzq1, HIGH);
            digitalWrite(motorIzq2, LOW);
            analogWrite(ruedaIzquierda, 0);
            analogWrite(ruedaDerecha, 0);
            Serial.println(Centimetros());

            servo.write(25); //posición del servomotor a la derecha
            delay(800);
            distancia_derecha = Centimetros(); //guarda la distancia de la derecha

            digitalWrite(motorDer1, LOW);
            digitalWrite(motorDer2, HIGH);
            digitalWrite(motorIzq1, HIGH);

```

```

digitalWrite(motorIzq2, LOW);
analogWrite(ruedaIzquierda, 0);
analogWrite(ruedaDerecha, 0);

servo.write(175); //posición del servomotor a la izquierda
delay(800);
distancia_izquierda = Centimetros(); //guarda la distancia de la izquierda

//si la distancia de la izquierda es mayor a la de la derecha
//indica que hay obstáculo en la derecha
if (distancia_izquierda > distancia_derecha) {
    servo.write(90); //posición del servomotor enfrente
    delay(800);
    digitalWrite(motorDer1, LOW);
    digitalWrite(motorDer2, HIGH);
    digitalWrite(motorIzq1, HIGH);
    digitalWrite(motorIzq2, LOW);
    analogWrite(ruedaIzquierda, 60);
    analogWrite(ruedaDerecha, 180);
    delay(300);
    Serial.write("obstáculo a la derecha"); //girar a la izquierda
}

//si la distancia de la derecha es mayor a la de la izquierda
//indica que hay obstáculo en la izquierda
if (distancia_izquierda < distancia_derecha) {
    servo.write(90); //posición del servomotor enfrente
    delay(800);
    Serial.write("obstáculo a la izquierda"); //girar a la derecha
    digitalWrite(motorDer1, LOW);
    digitalWrite(motorDer2, HIGH);
}

```

```

    digitalWrite(motorIzq1, HIGH);
    digitalWrite(motorIzq2, LOW);
    analogWrite(ruedaIzquierda, 180);
    analogWrite(ruedaDerecha, 60);
    delay(300);
  }
}

else
  { //si no se detecta obstaculo, continuar la trayectoria
    Serial.write("continua, sin obstáculos");
    digitalWrite(motorDer1, LOW);
    digitalWrite(motorDer2, HIGH);
    digitalWrite(motorIzq1, HIGH);
    digitalWrite(motorIzq2, LOW);
    analogWrite(ruedaIzquierda, 100);
    analogWrite(ruedaDerecha, 100);
    delay(300);
  } //fin del else de los obstáculos

} //fin del ciclo while

} //fin de método evasión de obstáculos

```