

**UNIVERSIDAD POLITÉCNICA SALESIANA**

**SEDE CUENCA**

**CARRERA: INGENIERÍA ELECTRÓNICA**

“DESARROLLO DE UNA INTERFAZ PARA EL RECONOCIMIENTO DE COMANDOS DE VOZ APLICADO AL CONTROL DE SISTEMAS DOMÓTICOS”

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO**

**DE:**

**INGENIERO ELECTRÓNICO**

**Autores:**

Christian Gerardo Alvarez Puluchi

Miguel Armando Cajas Fuela

**Tutor:**

Ing. Diego Paúl Chacón Troya, Mdhd.

**CUENCA, JUNIO DE 2016**

## CERTIFICACIÓN

En calidad de DIRECTOR DEL PROYECTO DE TITULACIÓN “**DESARROLLO DE UNA INTERFAZ PARA EL RECONOCIMIENTO DE COMANDOS DE VOZ APLICADO AL CONTROL DE SISTEMAS DOMÓTICOS**”, elaborado por Christian Gerardo Alvarez Puluchi y Miguel Armando Cajas Fuela, declaro y certifico la aprobación del presente trabajo de titulación basándose en la supervisión y revisión de su contenido.

Cuenca, Junio de 2016



a b

Ing. Diego Paúl Chacón Troya, Mdhd.

**DIRECTOR DEL TRABAJO DE TITULACIÓN**

## DECLARATORIA DE RESPONSABILIDAD

Los autores son los únicos responsables por los contenidos, conceptos, ideas, análisis, resultados investigativos y manifestados en el presente trabajo de titulación ***“DESARROLLO DE UNA INTERFAZ PARA EL RECONOCIMIENTO DE COMANDOS DE VOZ APLICADO AL CONTROL DE SISTEMAS DOMÓTICOS”***.

Cuenca, Junio de 2016



**Christian Gerardo Alvarez Puluchi**

**CI: 140071870-4**



**Miguel Armando Cajas Fuela**

**CI: 010454143-8**

**© DERECHOS DE AUTOR**

Nosotros Christian Gerardo Alvarez Puluchi y Miguel Armando Cajas Fuela, por medio del presente documento certifico que hemos leído la Política de Propiedad Intelectual de la Universidad Politécnica Salesiana y estamos de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo de investigación quedan sujetos a lo dispuesto en la Política.

Asimismo, autorizamos a la Universidad Politécnica Salesiana para que realice la digitalización y publicación de este trabajo de investigación en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, Junio de 2016



Christian Alvarez P.

C.I: 140071870-4



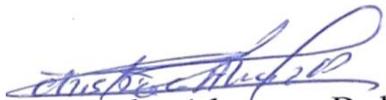
Miguel Cajas F.

C.I. 010454143-8

## CESIÓN DE DERECHOS DE AUTOR

Nosotros Christian Gerardo Alvarez Puluchi con documento de identificación N° 140071870-4 y Miguel Armando Cajas Fuela con documento de identificación N° 010454143-8, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de grado intitulado: **“DESARROLLO DE UNA INTERFAZ PARA EL RECONOCIMIENTO DE COMANDOS DE VOZ APLICADO AL CONTROL DE SISTEMAS DOMÓTICOS”**, mismo que ha sido desarrollado para optar por el título de: Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.



Christian Gerardo Alvarez Puluchi

C.I. 140071870- 4



Miguel Armando Cajas Fuela

C.I. 010454143- 8

Cuenca, 07 de Julio de 2016



---

# ÍNDICE GENERAL

---

<b>Índice General .....</b>	<b>I</b>
<b>Índice de figuras .....</b>	<b>III</b>
<b>Índice de tablas.....</b>	<b>V</b>
<b>Índice de ecuaciones.....</b>	<b>VI</b>
<b>Agradecimientos.....</b>	<b>VII</b>
<b>Dedicatoria.....</b>	<b>VIII</b>
<b>ANTECEDENTES .....</b>	<b>9</b>
<b>INTRODUCCIÓN .....</b>	<b>11</b>
<b>1. FUNDAMENTACION TEORICA .....</b>	<b>12</b>
1.1. Domótica .....	12
1.1.1. El sistema domótico .....	12
1.1.2. Aplicaciones domóticas .....	14
1.1.3. Protocolos de comunicación para domótica.....	14
1.2. Reconocimiento del habla .....	16
1.2.1. Métodos para el reconocimiento del habla .....	16
1.2.2. Diseño de un sistema de reconocimiento de voz .....	18
1.3. Reconocimiento del habla aplicado a la domótica .....	20
1.4. Microcontroladores .....	22
1.4.1. Arduino .....	22
1.5. SmartVR.....	23
<b>2. ANÁLISIS DEL SISTEMA .....</b>	<b>24</b>
2.1. Análisis técnico .....	24
2.1.1. Selección de la tarjeta de desarrollo para el reconocimiento de voz... 24	
2.1.2. Selección del microcontrolador .....	25
2.1.3. Selección del protocolo de comunicación.....	26
2.1.4. Selección de los elementos de alimentación .....	27
2.1.5. Selección de los componentes para el diseño e impresión de la Plaqueta de Circuito Impreso (PCB) de control del sistema domótico. ....	28
<b>3. DISEÑO E IMPLEMENTACIÓN DEL SISTEMA .....</b>	<b>30</b>
3.1. Diseño y solución del sistema .....	30
3.2. Diseño de hardware.....	32
3.2.1. Diseño de las carcasas .....	32
3.2.2. Diseño de la Plaqueta de Circuito Impreso (PCB).....	34

3.2.3. Impresión de la Plaqueta de Circuito Impreso (PCB) .....	36
3.3. Diseño de Software .....	38
3.3.1. Configuración de la tarjeta de reconocimiento de voz EasyVR 2.0....	38
3.3.2. Configuración de los módulos XBee .....	49
3.3.3. Programación .....	51
<b>4. FUNCIONAMIENTO .....</b>	<b>59</b>
4.1. Funcionamiento del sistema de reconocimiento de voz .....	59
4.2. Funcionamiento de la Plaqueta de Circuito Impreso (PCB) de control del sistema domótico.....	61
<b>5. ANÁLISIS DE RESULTADOS .....</b>	<b>63</b>
5.1. Análisis del sistema “Domotic Voice” .....	63
5.1.1. Prueba para determinar el nivel de ruido que soporta el sistema.....	65
5.1.2. Prueba para determinar el tiempo de respuesta del sistema. ....	68
5.1.3. Prueba para determinar la capacidad de reconocimiento que posee el sistema para diferentes voces. ....	72
5.2. Análisis económico .....	77
<b>6. CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>90</b>
6.1. Conclusiones .....	90
6.2. Recomendaciones .....	92
<b>ABREVIATURAS.....</b>	<b>94</b>
<b>ANEXOS .....</b>	<b>95</b>
<b>REFERENCIAS BIBLIOGRAFICAS.....</b>	<b>110</b>

---

## ÍNDICE DE FIGURAS

---

<i>Figura 1. 1. Esquema de actuación de un sistema domótico [2].</i>	13
<i>Figura 1. 2. Control de dispositivos mediante KNX [3].</i>	15
<i>Figura 1. 3. Red LonWorks [3].</i>	15
<i>Figura 1. 4. Módulos de Arduino [15].</i>	23
<i>Figura 1. 5. Módulo SmartVR [16].</i>	23
<i>Figura 2. 1. Tarjeta EasyVR 2.0 [17].</i>	24
<i>Figura 2. 2. Módulo de Arduino Uno [18].</i>	25
<i>Figura 2. 3. Módulo de Arduino Nano [19].</i>	26
<i>Figura 2. 4. Módulo XBee serie 2 [20].</i>	27
<i>Figura 2. 5. Batería de Lipo.</i>	27
<i>Figura 3. 1. Diagrama de flujo para el reconocimiento de voz.</i>	30
<i>Figura 3. 2. Carcasa plástica diseñada en Inventor.</i>	33
<i>Figura 3. 3. Carcasa plástica implementada.</i>	33
<i>Figura 3. 4. Cobertor de cuero.</i>	33
<i>Figura 3. 5. Carcasa de aluminio diseñada en Inventor.</i>	34
<i>Figura 3. 6. Carcasa de aluminio implementada.</i>	34
<i>Figura 3. 7. Esquemático del PCB de control para el servidor domótico.</i>	35
<i>Figura 3. 8. Layout del PCB de control para el servidor domótico.</i>	35
<i>Figura 3. 9. Diseño de las huellas del PCB.</i>	36
<i>Figura 3. 10. PCB terminado.</i>	37
<i>Figura 3. 11. Ventana para crear un nuevo proyecto.</i>	39
<i>Figura 3. 12. Ventana para dar un nombre al proyecto qxp.</i>	39
<i>Figura 3. 13. Opción para agregar archivos de audio.</i>	40
<i>Figura 3. 14. Ventana para comprimir los archivos de audio.</i>	40
<i>Figura 3. 15. Ventana de solicitud de etiqueta.</i>	41
<i>Figura 3. 16. Ventana del software QuickSynthesis5 con los archivos de audio.</i>	41
<i>Figura 3. 17. Opción para comprimir los archivos de audio.</i>	42
<i>Figura 3. 18. Ventana para el proceso de comprimir los archivos de audio.</i>	42
<i>Figura 3. 19. Ventana de Build Options.</i>	43
<i>Figura 3. 20. Ventana de Build Project.</i>	43
<i>Figura 3. 21. Ventana para seleccionar el puerto de comunicación.</i>	44
<i>Figura 3. 22. Opción para subir el archivo qxp con los comandos de audio.</i>	44
<i>Figura 3. 23. Ventana Update Custom Data.</i>	45
<i>Figura 3. 24. Ventana Update Custom Data para descarga del archivo de audio.</i>	45
<i>Figura 3. 25. Ventana para conectar la tarjeta EasyVR 2.0 con el software EasyVR-Commander.</i>	46
<i>Figura 3. 26. Opción para probar los archivos de audio.</i>	46
<i>Figura 3. 27. Opción para crear nuevos comandos de voz.</i>	47
<i>Figura 3. 28. Ventana de Train Command.</i>	47
<i>Figura 3. 29. Opción para generar código para Arduino.</i>	48
<i>Figura 3. 30. Software X-CTU.</i>	49
<i>Figura 3. 31. Opción para leer la configuración inicial del XBee.</i>	50
<i>Figura 3. 32. Opción para cargar el firmware.</i>	50
<i>Figura 3. 33. Prueba de comunicación.</i>	55

<b>Figura 3. 34.</b> Tipos de errores que detecta el programa del sistema “Domotic Voice”.	56
<b>Figura 3. 35.</b> Código para capturar los errores.	57
<b>Figura 4. 1.</b> Sistema de reconocimiento de voz “Domotic Voice”.	61
<b>Figura 4. 2.</b> Gateway de control del sistema domótico.	62
<b>Figura 5. 1.</b> Nivel de ruido en condiciones normales.	66
<b>Figura 5. 2.</b> Señal de ruido en condiciones normales proporcionado por Sound Meter.	66
<b>Figura 5. 3.</b> Nivel de ruido que genera problemas al sistema.	67
<b>Figura 5. 4.</b> Señal de ruido que genera problemas proporcionado por Sound Meter.	67
<b>Figura 5. 5.</b> Respuesta en tiempo del espectro de voz para la orden “Abre cortinas del sistema “Domotic Voice”.	68
<b>Figura 5. 6.</b> Respuesta en tiempo del espectro de voz para la orden “Menú Inicio” del software de “Reconocimiento de voz de Windows”.	70
<b>Figura 5. 7.</b> Reconocimiento de voz de Windows.	70
<b>Figura 5. 8.</b> Respuesta en tiempo del espectro de voz para la orden “Facebook” de la aplicación “Control por voz gratis”.	71
<b>Figura 5. 9.</b> Control por voz gratis.	71
<b>Figura 5. 10.</b> Respuesta en tiempo y frecuencia de Christian para la orden “Casa”.	75
<b>Figura 5. 11.</b> Respuesta en tiempo y frecuencia de Fernando para la orden “Casa”.	75
<b>Figura 5. 12.</b> Respuesta en tiempo y frecuencia de Katherine para la orden “Casa”.	76
<b>Figura 5. 13.</b> Respuesta en tiempo y frecuencia de Carlos para la orden “Casa”.	76
<b>Figura 5. 14.</b> Indicadores de rentabilidad del sistema “Domotic Voice”.	86
<b>Figura 5. 15.</b> Presupuestos de los sistemas de reconocimiento de voz de la tabla 17 con respecto al sistema “Domotic Voice”.	88
<b>Figura 5. 16.</b> Diferencia de presupuestos de los sistemas de reconocimiento de voz de la tabla 17 con respecto al sistema “Domotic Voice”.	89

---

## ÍNDICE DE TABLAS

---

<b>Tabla 1.</b> <i>Tiempos de respuestas del sistema “Domotic Voice”</i> .....	69
<b>Tabla 2.</b> <i>Tiempos de respuestas del software “Reconocimiento de voz de Windows”</i> . .....	69
<b>Tabla 3.</b> <i>Tiempos de respuestas de la aplicación “Control por voz gratis”</i> . ....	70
<b>Tabla 4.</b> <i>Promedio en segundos de los resultados obtenidos de los tres sistemas puestos a prueba.</i> .....	72
<b>Tabla 5.</b> <i>Prueba de Christian para cada orden.</i> .....	72
<b>Tabla 6.</b> <i>Prueba de Fernando para cada orden.</i> .....	73
<b>Tabla 7.</b> <i>Prueba de Katherine para cada orden.</i> .....	73
<b>Tabla 8.</b> <i>Prueba de Carlos para cada orden.</i> .....	74
<b>Tabla 9.</b> <i>Promedios obtenidos para cada orden que el sistema reconoce.</i> .....	74
<b>Tabla 10.</b> <i>Costos de dispositivos digitales del sistema.</i> .....	77
<b>Tabla 11.</b> <i>Costos de la etapa de alimentación.</i> .....	78
<b>Tabla 12.</b> <i>Costos del PCB de control del servidor domótico.</i> .....	78
<b>Tabla 13.</b> <i>Costos adicionales para el desarrollo del sistema.</i> .....	78
<b>Tabla 14.</b> <i>Valor total del sistema de reconocimiento de voz.</i> .....	79
<b>Tabla 15.</b> <i>Requisitos iniciales para la producción del sistema de reconocimiento de voz.</i> .....	80
<b>Tabla 16.</b> <i>Flujo de caja.</i> .....	85
<b>Tabla 17.</b> <i>Presupuestos de sistemas desarrollados por otras instituciones para el reconocimiento de voz.</i> .....	86

---

## ÍNDICE DE ECUACIONES

---

<b>Ecuación 1.</b> Ecuación para determinar la predicción lineal .....	17
<b>Ecuación 2.</b> Ecuación para determinar la estimación de $s(n)$ mediante las muestras anteriores ( $p$ ).....	17
<b>Ecuación 3.</b> Ecuación para determinar el error .....	17
<b>Ecuación 4.</b> Ecuación para determinar los componentes de la predicción lineal....	17
<b>Ecuación 5.</b> Ecuación para determinar los coeficientes cepstrales de frecuencia Mel (MFCC).....	18
<b>Ecuación 6.</b> Ecuación para determinar la distancia.....	20
<b>Ecuación 7.</b> Ecuación para determinar la potencia por hora que entrega la batería .....	28
<b>Ecuación 8.</b> Ecuación para determinar el tiempo de descarga que posee la batería .....	28
<b>Ecuación 9.</b> Ecuación para determinar el valor de las resistencias del regulador de voltaje.....	29
<b>Ecuación 10.</b> Ecuación para determinar la Utilidad Bruta (UB).....	82

---

## **AGRADECIMIENTOS**

---

En primer lugar damos gracias a Dios por permitirnos terminar con éxito nuestra formación universitaria, así como de manera especial al Ing. Diego Chacón tutor del proyecto técnico de titulación, por brindarnos su ayuda con sus aportes para poder terminar con éxito el desarrollo del proyecto.

Además queremos brindar nuestro sincero agradecimiento al Ing. Carlos Peralta y a los laboratoristas por permitirnos hacer uso del laboratorio de Instalaciones Civiles para llevar a cabo las pruebas del sistema desarrollado. A más de ello a cada una de las personas que de buena voluntad ayudaron a probar el sistema para poder corregir cualquier error en nuestro sistema.

Finalmente nuestro más profundo y sincero agradecimiento a nuestros padres por brindarnos su apoyo, ya que sin ellos no hubiésemos podido empezar y culminar nuestra carrera universitaria. A cada uno de nuestros familiares y amigos por la fortaleza que nos supieron dar para no perder los ánimos y alcanzar la meta final de terminar la formación universitaria.

**CHRISTIAN GERARDO – MIGUEL ARMANDO**

---

## **DEDICATORIA**

---

Gracias a Dios por permitirme tener la oportunidad de poder estudiar, gracias a mi familia por ser los principales promotores de este sueño que quizá sea el más importante en la vida de un joven. Gracias a ellos por cada día confiar y creer en las locas ideas de su hijo.

**CHRISTIAN ALVAREZ PULUCHI**

Dedicado de manera especial a Dios por darme la fortaleza y la sabiduría para poder enfrentar y culminar con éxito mi formación universitaria. A mis padres por su apoyo y ayuda incondicional que me brindan en cada momento, así como los valores y principios inculcados para ser una mejor persona. A mis hermanos, cuñada y sobrinos por su apoyo incondicional que me brindan en cada momento para no dejarme abatir bajo ninguna situación y seguir en la lucha de conseguir cada una de mis metas trazadas.

**MIGUEL CAJAS FUELA**

## ANTECEDENTES

La domótica tuvo sus inicios en Estados Unidos con el desarrollo de equipos que permitieron la automatización de viviendas y edificios, los cuales con el tiempo han ido mejorando el diseño y las prestaciones que pueden brindar a un hogar. Es a partir de estos inicios que empresas e instituciones de algunos países se dedican a la investigación y desarrollo de sistemas que puedan interactuar con el usuario para controlar su hogar.

Estos dispositivos conforme han pasado los años se han hecho más sofisticados permitiendo un mejor control de los sensores y actuadores distribuidos en un hogar. Un sistema que ha ido mejorando es el de reconocimiento de voz, el cual permite que un usuario por medio de comandos de voz pueda controlar su hogar más fácilmente sin tener que manipular los dispositivos de mando de ciertos elementos. Creando una relación entre hombre-máquina, por medio de esta herramienta rápida y directa de comunicación. A pesar de esto los sistemas de reconocimiento de voz todavía poseen ciertos limitantes debido al procesamiento de las señales de voz, ya que dependen de factores como el ambiente en el cual se va utilizar el sistema.

A pesar de estas limitantes, actualmente el reconocimiento de voz se encuentra aplicado en diversos campos, con la finalidad de conseguir un hogar más confortable, seguro e inteligente; con base en esto se viene trabajando en la introducción de las técnicas de reconocimiento de voz aplicadas a la domótica y a la robótica.

## JUSTIFICACIÓN

Un aporte adicional que se viene realizando al área de la domótica son los sistemas de reconocimiento de voz, los cuales no han sido explotados en su totalidad debido a que tienen cierto grado de complejidad y sobre todo por sus costos elevados, razón por lo que la mayoría de personas no pueden acceder a esta herramienta. Otro factor influyente, es que no existen en nuestro país empresas que se dediquen a la producción y venta de estos sistemas, por lo cual resulta necesario depender de otros países para disponer de estas tecnologías.

Sin embargo los sistemas de reconocimiento de voz comerciales son compatibles únicamente con aquellos de la misma marca, lo cual resulta ser una desventaja para aquellas personas que tienen un sistema diferente al del reconocimiento de voz. Esto se ve reflejado en el costo adicional que deberán pagar los usuarios para poder disfrutar de esta herramienta.

Por estas razones, se desarrollará un sistema de reconocimiento de voz que tenga la capacidad de adaptarse a sistemas domóticos en los cuales se pueda manipular las entradas de su servidor. Cubriendo así la necesidad de obtener un mecanismo que conjuntamente con una instalación domótica permitan un mayor confort a cualquier usuario que necesite un ambiente inteligente en el cual pueda interactuar con su hogar. Así como también puede servir de ayuda para las personas que padecen una discapacidad motriz o visual, ya que únicamente por medio de comandos de voz podrán realizar las tareas de control de los componentes de su vivienda. De acuerdo a datos estadísticos proporcionados por el Concejo Nacional de Igualdad de Discapacidades (CONADIS), existe un gran número de personas que padecen una discapacidad visual o de movilidad, lo cual les dificulta operar dispositivos de mando comunes como pantallas táctiles, control remoto, etc.

Con este sistema se pretende facilitar las actividades que hoy en día tenemos, integrando así un sistema capaz de reconocer una orden de forma verbal y transmitirla a una acción por medio de una interfaz que gobierne los sensores y actuadores distribuidos en el hogar.

## INTRODUCCIÓN

Una de las tecnologías que todavía se sigue investigando es la del reconocimiento de voz, la cual sirve como aporte para generar sistemas más inteligentes; los mismos que hoy en día no solo están presentes en el campo de la robótica, sino también son parte de los sistemas domóticos diseñados para mejorar las prestaciones de una vivienda, debido a que ayuda a desempeñar tareas dentro y fuera del hogar únicamente por medio de comandos de voz. Por tal razón nuestro proyecto va enfocado hacia esta área con la finalidad de desarrollar una interfaz capaz de controlar los sensores y actuadores distribuidos en el hogar [1].

En el presente documento se establece el desarrollo de un sistema para controlar un hogar mediante comandos de voz con una interface de conexión a sistemas domóticos en los cuales se pueda manipular y controlar las entradas de su servidor. Concretamente este documento detalla el proceso de análisis, diseño, implementación y pruebas de un prototipo de interfaz por voz que se puede conectar a un sistema domótico.

El sistema que se menciona es un sistema domótico que pueda conectarse mediante los protocolos de comunicación estándar y al mismo tiempo estaría compuesto de un servidor domótico y partes móviles distribuidas dentro de un hogar. La implementación de estos elementos no forma parte de las tareas de este proyecto, ya que se pretende realizar las respectivas pruebas en el sistema domótico LOXONE ubicado en el laboratorio de Instalaciones Civiles de la Universidad Politécnica Salesiana Sede Cuenca.

---

# CAPÍTULO 1

---

## 1. FUNDAMENTACION TEORICA

El presente capitulo dará a conocer la información más relevante sobre cada uno de los temas que intervienen en el desarrollo del proyecto, así como es el tema de la parte domótica, el reconocimiento del habla, el reconocimiento del habla aplicado en la domótica, los microcontroladores, y SmartVR, de manera que se tenga una idea clara de cada uno de estos temas.

### 1.1. Domótica

“El termino domótica proviene de la unión de la palabra en latín domus que significa casa o domicilio y de la palabra griega autouatoc que significa automática”. Por lo tanto se entiende como automatización de una casa o domicilio a partir del uso de dispositivos con cierto grado de inteligencia. La domótica tuvo sus inicios en los años 60 con la llegada de dispositivos para la automatización de edificios en Estados Unidos. Con esto se dio inicio para que la domótica puede extenderse a nuevos campos y el desarrollo de nuevos dispositivos que permitan mejorar las prestaciones que ofrece un hogar; con la finalidad de proporcionar al usuario confort, seguridad, mejorar las comunicaciones y la gestión de la energía [2].

#### 1.1.1. El sistema domótico

El sistema domótico consta de un grupo de elementos que detectan los cambios de estado de una variable y los cuales al mismo tiempo se encargan de transmitir la información a otros elementos del sistema, para que se encarguen de actuar en función de reglas establecidas por parte del usuario [2]. Un sistema domótico está compuesto por:

**Sensores:** estos dispositivos permiten detectar los cambios que puede generar una variable, para lo cual toman información que luego es transmitida a una unidad que se encarga del control de estado de las variables del sistema. Los sensores poseen características tanto estáticas como dinámicas, cuya elección dependerá de la aplicación que desee realizar el usuario; por otro lado los sensores pueden ser de tipo analógico y digital, y su elección dependerá del parámetro a medir [2]. Los sensores se clasifican en:

- Sensores de luminosidad.
- Sensores de temperatura.
- Sensores volumétricos de presencia.
- Detectores de incendios.
- Detectores de inundación.
- Detectores de gas.
- Detectores de puertas y ventanas abiertas.
- Anemómetros.
- Interruptores de lluvia.
- Contadores de energía [2].

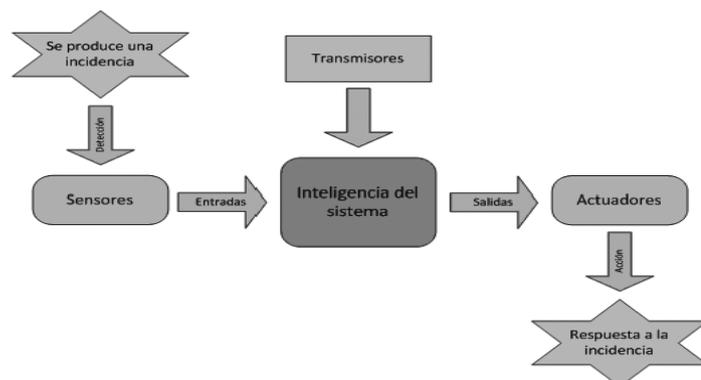
**Actuadores:** estos elementos se encargan de modificar las magnitudes del entorno y se los suele utilizar como interfaces de potencia. Los actuadores se clasifican de acuerdo al tipo de señal de entrada y a la magnitud controlada [2].

**Unidades de control:** es el elemento encargado de recibir la información de los sensores y procesarla para luego poder transmitir una orden al actuador. Se lo considera como el elemento que posee cierto grado de inteligencia dentro del sistema [2].

**Interfaces de usuario:** hace referencia a la interacción entre el sistema domótico y el usuario, es decir el medio en como el usuario se adapta al sistema y a su vez a los servicios que se automatizan del hogar [2].

**Pasarelas de comunicación:** es un dispositivo que permite controlar los elementos del hogar desde cualquier parte de la vivienda, por medio de una conexión entre las infraestructuras de comunicaciones de la vivienda y una red pública de datos [2].

En la figura 1.1 se puede observar el esquema de actuación de un sistema domótico.



**Figura 1. 1.** Esquema de actuación de un sistema domótico [2].

### **1.1.2. Aplicaciones domóticas**

Las aplicaciones domóticas son innumerables dadas las posibilidades que ofrece, por tal razón se darán a conocer las aplicaciones más importantes y que aportan para el desarrollo del proyecto planteado. Estas aplicaciones son las siguientes:

- Reducción del consumo energético.
- Control automático del encendido y apagado de las luces de la vivienda.
- Motorización de persianas.
- Selección de temperatura.
- Apertura de puertas y ventanas.
- Encendido y apagado de equipos dentro de la vivienda [2].

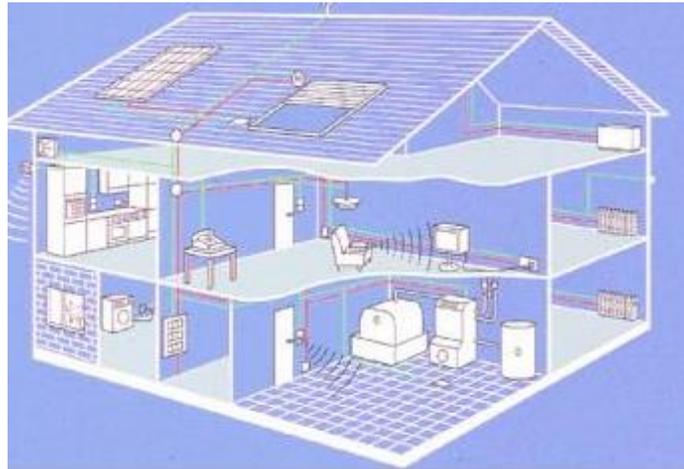
### **1.1.3. Protocolos de comunicación para domótica**

A continuación se darán a conocer los 3 protocolos de comunicación más importantes para el área de la domótica:

**KNX:** es un estándar que se lo utiliza para el área de las comunicaciones, cuyo diseño está basado para el control de todo tipo de servicios y aplicaciones en viviendas y edificios. Este estándar es independiente de cualquier plataforma de hardware y software; posee distintos modos de configuración; permite direcciones físicas y de grupos, y sobre todo aporta soluciones a la vivienda [3]. Se lo puede utilizar en las siguientes aplicaciones:

- Control de iluminación.
- Control de persianas.
- Sistemas de seguridad.
- Gestión energética.
- Calefacción.
- Sistemas de ventilación.
- Sistemas de control de edificios.
- Sistemas de control remoto.

En la figura 1.2 se puede observar el control de dispositivos mediante KNX.

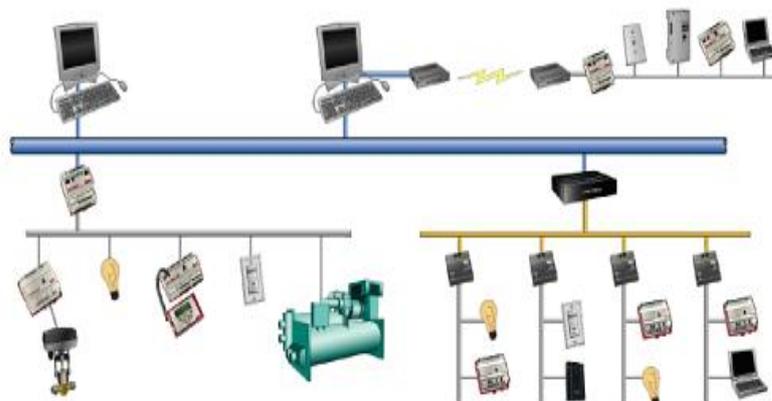


*Figura 1. 2. Control de dispositivos mediante KNX [3].*

**LonWorks:** es un protocolo abierto diseñado por Echelon en 1990 para la automatización de todo tipo de infraestructura, ofrece fiabilidad y robustez para el desarrollo de una arquitectura totalmente descentralizada. Este protocolo abarca los siete niveles OSI; permite un intercambio de información del tipo peer to peer; proporciona un cierto grado de inteligencia a sensores y actuadores, de tal manera que puedan reaccionar frente a eventos y gestionar las comunicaciones; permite emplear diferentes topologías y medios de comunicación [4]. Se lo puede utilizar en las siguientes aplicaciones:

- Automatización de edificios.
- Control industrial.
- Transporte.
- Automatización de viviendas.

En la figura 1.3 se puede observar una red LonWorks.



*Figura 1. 3. Red LonWorks [3].*

**ZigBee:** es un estándar basado en el protocolo IEEE 802.15.4, el cual permite armar redes inalámbricas de corta distancia y baja velocidad de datos. ZigBee opera en las bandas de 868MHz, 915MHz y 2.4GHz, y además permite transferir datos hasta 250Kbps. Ofrece un bajo costo en dispositivos, instalación y mantenimiento; posee un alcance de 10 metros a 1200 metros, y puede estar formado por hasta 255 nodos [5]. Se lo puede utilizar en las siguientes aplicaciones:

- Seguridad
- Lectura de instrumentos de servicios
- Sistema de riego automático
- Control de iluminación
- Control de temperatura
- Controles remotos

## **1.2. Reconocimiento del habla**

El habla es una herramienta que utilizan las personas a diario para poder comunicarse entre sí, siendo su principal elemento la voz que permite expresar la información que desea compartir una persona. Es importante tomar en cuenta que la voz no es igual para todas las personas, ya que esta varía en función del tono de voz que posean. Tomando en cuenta los aspectos del habla, se han desarrollado sistemas que reconocen este parámetro, con la finalidad de ser utilizados en otros sistemas que permitan utilizar a la voz como elemento de control.

Entre los aspectos más importantes que se debe considerar para el diseño de sistemas de reconocimiento del habla, es la parte de procesamiento de la señal de voz, las técnicas de reconocimientos de patrones, y las diferentes formas de hablar que posee cada persona, de tal manera que se pueda realizar un reconocimiento óptimo de la señal de voz emitida. En lo que respecta al procesamiento de la señal de voz, esta debe ser procesada a la entrada del sistema de manera que se extraiga la información de audio más relevante de la señal emitida; para ello es necesario reducir el ruido y cancelar el eco que se puede generar por el ambiente o por personas cercanas a donde se emita la señal de voz [6] [7].

### **1.2.1. Métodos para el reconocimiento del habla**

Existen varios métodos para el reconocimiento del habla, los cuales permiten extraer las características más importantes de la voz, por tal razón para el proyecto solo se tomarán en cuenta dos métodos que son los más utilizados para el reconocimiento del

habla. Estos métodos son: Análisis de la predicción lineal y Análisis Cepstral. A continuación se dan a conocer cada uno de ellos:

### **Análisis de la predicción lineal**

Este método representa la envolvente de una señal de voz en una forma comprimida, por medio de un modelo lineal; con la predicción lineal se pretende estimar el valor de la siguiente muestra a partir de muestras pasadas [6]. Esto se representa mediante la ecuación 1.

*Ecuación 1. Ecuación para determinar la predicción lineal [6].*

$$s(n) = s(n-1), s(n-2), \dots, s(n-k) \quad (1)$$

La estimación de  $s(n)$  mediante las muestras anteriores ( $p$ ), puede calcularse mediante la ecuación 2.

*Ecuación 2. Ecuación para determinar la estimación de  $s(n)$  mediante las muestras anteriores ( $p$ ) [6].*

$$s(n) \approx \sum_{k=1}^p a_k * s(n-k) \quad (2)$$

Cuyo error se obtiene mediante la ecuación 3.

*Ecuación 3. Ecuación para determinar el error [6].*

$$\epsilon(n) = s(n) - \sum_{k=1}^p a_k * s(n-k) \quad (3)$$

Los componentes de la predicción lineal se obtienen minimizando un criterio cuadrático en los errores de predicción; el número de coeficientes está en función de la frecuencia de muestreo, tal como se indica en la ecuación 4.

*Ecuación 4. Ecuación para determinar los componentes de la predicción lineal [6].*

$$LPC = \left( \frac{\text{Frecuencia de Muestreo}}{1000} \right) + 3 \quad (4)$$

### **Análisis Cepstral**

“Este método se basa en el análisis del espectro, debido a que proporciona información de los parámetros de la voz, obteniendo la parte de excitación, así como del filtro que representa el tracto vocal” [7].

Dentro del análisis Cepstral, los coeficientes cepstrales de frecuencia Mel (MFCC) toman un papel importante, ya que permiten extraer los parámetros de la señal de voz que están basados en la variación de los anchos de banda de las frecuencias críticas del oído [7].

*Ecuación 5. Ecuación para determinar los coeficientes cepstrales de frecuencia Mel (MFCC) [7].*

$$\text{Mel}(f) = 2595 * \log \left( 1 + \frac{f}{700} \right) \quad (5)$$

Una herramienta que se puede utilizar para el análisis Cepstral, es el software de Matlab, el cual permite obtener los coeficientes cepstrales utilizando la transformada rápida de Fourier (FFT). La función que se debe utilizar en este software es la rceps, la cual proporciona la cepstrum real de la función que se ingrese. La ventaja con este análisis, es que se pueden derivar una serie de parámetros, sin importar las distorsiones que se puedan introducir por el micrófono [7].

### **1.2.2. Diseño de un sistema de reconocimiento de voz**

En el diseño de un sistema de reconocimiento de voz es necesario tomar en cuenta tres aspectos importantes, tal como: el procesamiento de la señal de audio, la extracción de las características de la señal de voz y la identificación de los parámetros de la señal de voz; donde cada uno de ellos permiten que el sistema pueda reconocer una señal, de acuerdo a las características de audio del usuario. A continuación se dan a conocer cada uno de ellos:

#### **Procesamiento de la señal de audio**

El procesamiento de una señal de audio permite extraer la información más relevante que se requiera en un determinado proceso, para ello es indispensable aplicar un método de parametrización. En este proceso es necesario detectar el tipo de voz, eliminar el ruido y tratar de cancelar el eco que se pueda producir al momento de grabar la señal de audio [6]. Dentro del procesamiento de la señal es necesario tomar en cuenta los siguientes parámetros:

- **Digitalización:** este proceso consiste en convertir la señal de entrada analógica en una señal digital, de manera que puede ser manipulable por un software. Es decir convertir una señal con muestras continuas en una sucesión de valores discretos. En este proceso se debe considerar la tasa de muestreo, los bits por segundo, así como como el número de canales [7].
- **Filtrado:** permite reducir los ruidos que pueden generarse al momento de emitir la señal de voz en un micrófono, de manera que solo se obtenga los parámetros de voz deseada. En el diseño de un filtro se debe considerar la frecuencia de corte y paso a utilizar según las especificaciones que se requieran [7]. Un tipo de filtro que se puede utilizar, es el filtro de pre-énfasis, un filtro pasa altas de primer orden que considera las frecuencias altas para no perder la información al momento de realizar la segmentación [6].
- **Segmentación:** consiste en dividir en segmentos la señal de voz, de manera que se pueda definir el punto de inicio y final de cada palabra. Este proceso se realiza con la intención de tomar en cuenta únicamente la señal que contiene información y así evitar el uso innecesario de memoria y tiempo de cálculo que pueda ocasionar un análisis erróneo de la señal. Para ello los segmentos son guardados cada uno en una columna de una matriz, para poder realizar el procesamiento de la señal; en la etapa de procesamiento se puede utilizar una ventana de Hamming de 30 milisegundos con un desplazamiento de 10 milisegundos entre cada ventaneo [6].

### **Extracción de las características de la señal de voz**

Para la extracción de las características de la señal de voz existen varios métodos, de los cuales el más utilizado para el reconocimiento de voz, es el análisis Cepstral, el mismo que ya se dio a conocer en el punto 1.2.1

### **Identificación de los patrones de voz**

En la identificación de los patrones de voz es necesario considerar lo siguiente:

- **Cálculo de la distancia:** este cálculo permite determinar la distancia de los vectores característicos con la finalidad de ser combinados y comparados con los patrones de referencia [6]. Para poder determinar la distancia se aplica la ecuación 6.

*Ecuación 6. Ecuación para determinar la distancia [7].*

$$d = \sqrt{\sum_{i=1}^D |f_i - f_i'|^2} \quad (6)$$

Donde D son las componentes de los dos vectores característicos  $f_i$  y  $f_i'$ .

Una herramienta práctica para el cálculo de la distancia es el software de Matlab, el cual aplica la distancia Euclidiana para diferenciar la palabra emitida por la persona con la base de datos que se utiliza para el entrenamiento del sistema. Este cálculo nos proporciona un valor numérico de la distancia entre dos matrices de igual dimensiones [7].

### **1.3. Reconocimiento del habla aplicado a la domótica**

Actualmente el reconocimiento de voz se encuentra aplicado en diversos campos, y uno de estos es la domótica, con la finalidad de conseguir un hogar más confortable, seguro e inteligente; con base en esto diversas instituciones e investigadores vienen trabajando en la introducción de las técnicas de reconocimiento de voz aplicadas a la domótica, por lo cual se han ejecutado trabajos como:

Human Voice Recognition Applied to Domotics, el cual consiste en un sistema que se encarga de procesar la palabra emitida por una persona y la codifica de manera predictiva, permitiendo de esta manera controlar los elementos domóticos que posee el hogar a través del protocolo de comunicación IEEE 802.15.4 con radios ZigBeePro [8].

De igual manera se han creado sistemas de simulación domótica con interfaz verbal, uno de ellos fue desarrollado en la Universidad Autónoma de Barcelona, el cual consiste en un simulador con entorno en 3D, el cual puede recrear una vivienda y los movimientos que realiza una persona, con el afán de que por medio de comandos de voz se pueda controlar una reacción dentro del hogar. Este tipo de simuladores representan una gran ayuda al momento de realizar pruebas, ya que un sistema domótico completo representa un costo alto [9].

Por otra parte, en el mercado existen productos como el free@home desarrollado por la corporación Automation and Power Technologies (ABB), el cual permite al usuario controlar aplicaciones dentro del hogar por medio de comandos de voz, ya sea control de persianas, iluminación, calefacción, etc. El problema con estos sistemas, es que el dispositivo de reconocimiento de voz se enlaza solo con el servidor de la misma marca [10].

Otro ejemplo es el robot domótico móvil desarrollado en la Universidad Politécnica de Madrid, quien unió las funciones de la interfaz Servivox con el robot Roomba, de manera que se pueda establecer una comunicación que controle los movimientos del robot por medio de comandos de voz [11].

Un sistema que también se tomó en consideración, consiste en un control por voz cuya central de control se encuentra en una PC, este sistema permite cubrir las necesidades básicas del hogar proporcionando comodidad al usuario al tener que activar mediante micrófonos distribuidos en toda la vivienda, ya sea un sensor o un actuador. Lo interesante de este sistema es que es de tipo abierto [13].

Un ejemplo parecido al anterior, es el sistema desarrollado en el Instituto Politécnico Nacional de México, el cual consiste en un sistema que reconoce una palabra por medio de un micrófono, cuya palabra es codificada y comparada mediante una base de datos en una PC; este sistema posee un ambiente gráfico en Matlab que se encarga de realizar una predicción lineal, con el afán de obtener una comparación de coeficientes, con lo cual al ser reconocida la palabra se active o no un determinado elemento del hogar [6].

Finalmente se consideró dos sistemas desarrollados por la Universidad Politécnica Salesiana relacionados al reconocimiento de voz. El primer sistema consiste en un reconocimiento de voz basado en el algoritmo MFCC, SBC y espectrogramas [29]. El segundo sistema desarrollado por la Universidad Politécnica Salesiana consiste en una silla de ruedas controlada por medio de comandos de voz, la cual tiene un enfoque de ayuda social, ya que la silla permite a la persona que padece una discapacidad motriz moverse únicamente por medio de comandos de voz [31].

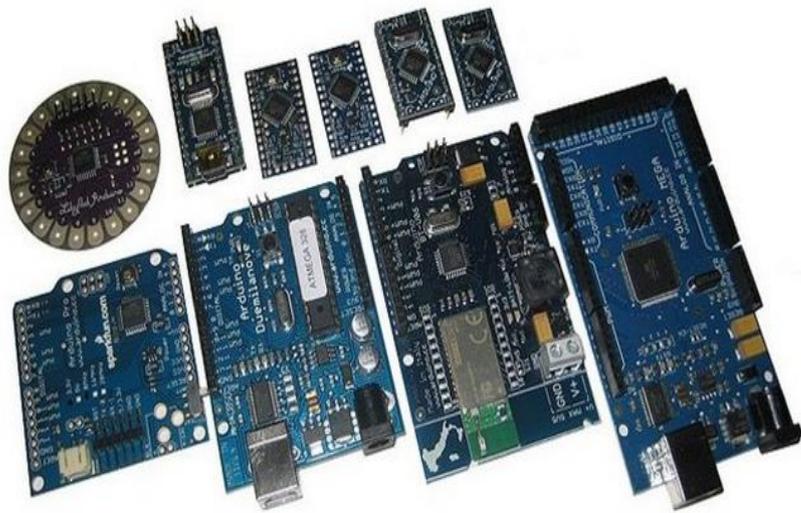
Todos estos proyectos desarrollados establecen su propia plataforma domótica para poder comunicar el sistema de reconocimiento de voz con la parte domótica; por ejemplo se utilizan plataformas de código libre como Raspberry Pi o Arduino para el desarrollo de servidores domóticos, ya que estos dispositivos permiten el uso de bibliotecas como LightwaveRF para enlazarse mediante protocolos de comunicación inalámbrica como WiFi, que permite que se establezca la comunicación con la placa de mando para el control de los dispositivos distribuidos en la vivienda [12].

#### **1.4. Microcontroladores**

Los microcontroladores son dispositivos electrónicos que permiten al usuario programar una serie de instrucciones, de tal manera que estas se ejecuten de forma autónoma. Un microcontrolador está compuesto principalmente por un CPU, diferentes tipos de memorias, y pines de entradas y salidas. Por lo cual se lo puede considerar como un computador montado en un solo chip, lo cual resulta ventajoso debido a que se puede ahorrar en tamaño y costos en un diseño de control que se desea realizar [14].

##### **1.4.1. Arduino**

Es una plataforma cuyo hardware y software son de tipo abierto, es decir que no requieren de licencias o permisos adicionales para poder utilizarlo. El microcontrolador que utiliza Arduino es del tipo AVR, el cual varía en función del modelo de la placa de Arduino. Por ejemplo el Arduino Uno tiene como microcontrolador al ATMEGA328P. Arduino posee la ventaja de poder sustituir su microcontrolador en caso de que se llegara a quemar por algún mal funcionamiento. Otra de las ventajas de Arduino son las librerías que posee para cada uno de los shield desarrollados, con los cuales se pueden controlar de manera más fácil ciertos elementos dentro de un determinado proyecto [14]. En la figura 1.4 se puede observar los diferentes módulos de Arduino.



*Figura 1. 4. Módulos de Arduino [15].*

### **1.5. SmartVR**

Es una plataforma multipropósito diseñada por la compañía Robotech, este dispositivo está basado en un procesador multipropósito de alto rendimiento CMOS, el cual puede ejecutar 100.000 operaciones por ciclo con una RAM de 512k x 8bit. Esta plataforma es idónea para el desarrollo de aplicaciones basadas en el reconocimiento de voz. El módulo está programado de fábrica con un firmware actualizable y una máquina virtual que permite el desarrollo de las aplicaciones de reconocimiento de voz [16]. En la figura 1.5 se puede observar el módulo SmartVR.



*Figura 1. 5. Módulo SmartVR [16].*

---

## CAPÍTULO 2

---

### 2. ANÁLISIS DEL SISTEMA

En el presente capítulo se da a conocer un análisis técnico sobre el sistema a implementar, en el cual se da a conocer las características del que porque fueron seleccionados ciertos dispositivos para el desarrollo del sistema de reconocimiento de voz. A continuación se da a conocer el análisis técnico:

#### 2.1. Análisis técnico

En este punto se da a conocer cada uno de los dispositivos seleccionados para llevar a cabo el sistema de reconocimiento de voz, de acuerdo a las características requeridas para el diseño. A continuación se dan a conocer cada uno de los dispositivos seleccionados:

##### 2.1.1. Selección de la tarjeta de desarrollo para el reconocimiento de voz

La tarjeta de reconocimiento de voz que se seleccionó para llevar a cabo el sistema, es una tarjeta EasyVR 2.0, la cual ofrece la posibilidad de reconocer seis idiomas diferentes; soporta hasta 32 voces definidas por el usuario; puede ser utilizado en cualquier terminal con interfaz Transmisor Receptor Asíncrono Universal (UART); se la puede alimentar a 3.3 voltios o a 5 voltios, y se la puede controlar ya sea mediante un PIC o un módulo Arduino [17]. En la figura 2.1 se puede observar la tarjeta EasyVR 2.0



*Figura 2. 1. Tarjeta EasyVR 2.0 [17].*

Otras características relevantes de la tarjeta EasyVR 2.0 son:

- Interfaz gráfica de usuario.
- Soporta altavoces de 8 ohmios.

- Reproduce sonidos de hasta 9 minutos.
- Posee 3 líneas GPIO que pueden ser controladas por los nuevos comandos de protocolo [17].

### 2.1.2. Selección del microcontrolador

Como sistema de control se seleccionó a un módulo de Arduino Uno y un módulo de Arduino Nano, los cuales disponen de un microcontrolador ATmega328P y ATmega328 respectivamente. La importancia de estos módulos, es que tanto su software como hardware son libres, lo cual resulta ventajoso al momento de llevar a cabo el proyecto, ya que no se van a requerir de la compra de una licencia para poder programar el microcontrolador. Otra ventaja con respecto a estos módulos son las librerías que poseen para el manejo de shields desarrollados para esta plataforma.

A continuación se da a conocer las especificaciones más relevantes del módulo de Arduino Uno:

- Requiere de un voltaje de alimentación de 7-12 voltios.
- Posee 14 pines digitales, de los cuales 6 pueden ser manejados como PWM.
- Posee 6 pines analógicos.
- Tiene una memoria flash de 32KB.
- Posee una SRAM de 2KB.
- Posee una EEPROM de 1KB.
- Su velocidad de reloj es de 16MHz.
- Tiene una dimensión de 68.6 milímetros de largo por 53.4 milímetros de ancho [18].

En la figura 2.2 se puede observar el módulo de Arduino Uno.



*Figura 2. 2. Módulo de Arduino Uno [18].*

A continuación se da a conocer las especificaciones más relevantes del módulo de Arduino Nano:

- Requiere de un voltaje de entrada de 7-12 voltios.
- Posee 14 pines digitales, de los cuales 6 pueden ser manejados como PWM.
- Posee 8 pines analógicos.
- Tiene una memoria flash de 16KB.
- Posee una SRAM de 1KB.
- Posee una EEPROM de 512 bytes.
- Su velocidad de reloj es de 16MHz.
- Tiene una dimensión de 45mm de largo por 18mm de ancho.

En la figura 2.3 se puede observar el módulo de Arduino Nano [19].

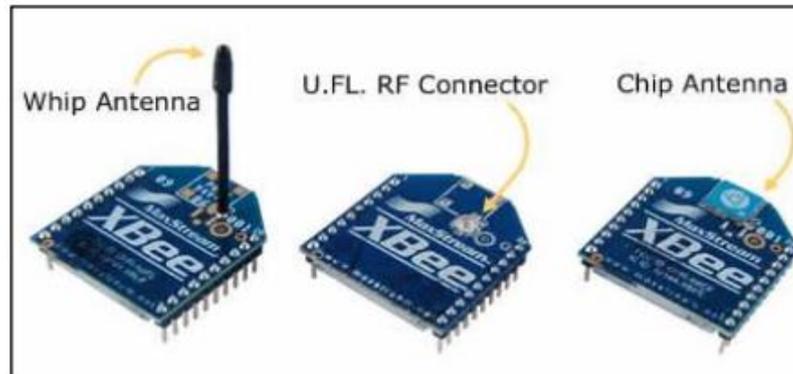


*Figura 2. 3. Módulo de Arduino Nano [19].*

### **2.1.3. Selección del protocolo de comunicación**

Se seleccionó como protocolo de comunicación entre el sistema de reconocimiento de voz y el sistema domótico al estándar ZigBee, ya que permite realizar una red inalámbrica de corta distancia, ideal para el propósito de automatizar una vivienda, y sobre todo por su bajo costo en la implementación. Para ello se hará uso de dos módulos XBee configurados como maestro y esclavo respectivamente para poder realizar la comunicación. Estos módulos poseen un alcance de 30 metros en interiores y de hasta 100 metros con antena dipolo; tienen un bajo consumo de corriente menor a los 50mA; poseen un interfaz serial y permiten 65.000 direcciones para cada uno de los 16 canales disponibles. Por otra parte resulta fácil de utilizar debido a que puede ser manipulable por medio de la plataforma de Arduino. Estos módulos XBee se clasifican de acuerdo a la serie, ya que cada uno de ellos se basan en un chipset diferente, para nuestro caso se seleccionó el XBee de la serie 2 basado en el chipset Ember que se utiliza en aplicaciones en las que se requiera de repetidores o una red

mesh; además puede ser utilizado ya sea en modo AT o API [20] [30]. En la figura 2.4 se puede observar el módulo XBee serie 2.



*Figura 2. 4. Módulo XBee serie 2 [20].*

#### **2.1.4. Selección de los elementos de alimentación**

Para la parte de alimentación se seleccionó una batería Turnigy de Lipo recargable de dos celdas, con un voltaje de 7.4 voltios y una corriente de 800mA; esta batería posee un tamaño reducido ideal para proyectos robóticos y portátiles. Con esta batería se va alimentar el módulo de Arduino Uno, el cual a su vez se encargará de alimentar a los demás componentes del sistema de reconocimiento de voz. Se debe considerar que estas baterías son más ligeras y se pueden encontrar con mayor facilidad en el mercado; además que al utilizar este tipo de baterías se puede mejorar la vida útil de un sistema electrónico y en caso de mantenimiento resulta fácil su sustitución [21]. En la figura 2.5 se puede observar la batería de Lipo.



*Figura 2. 5. Batería de Lipo.*

*Fuente: Los autores.*

Para determinar el tiempo de duración y la potencia que posee la batería de Lipo Turnigy se realizó los siguientes cálculos de acuerdo a los datos que posee la batería. A continuación se dan a conocer los resultados obtenidos:

**Datos:**

**Corriente por hora (Ih) = 0.8Ah**

**Voltaje (V) = 7.4V**

**Corriente máxima de descarga (Imax) = 50mA**

Para calcular la Potencia por hora (Ph) que posee la batería es necesario aplicar la ecuación 7 que se muestra a continuación.

*Ecuación 7. Ecuación para determinar la potencia por hora que entrega la batería [22].*

$$\text{Potencia por hora} = \text{Corriente por hora} * \text{Voltaje} \quad (7)$$

$$Ph = (0.8Ah) * (7.4V) = 5.92Wh$$

Por otro lado para determinar el tiempo de descarga (t) que posee la batería, es necesario tomar en cuenta la corriente por hora que posee la batería con respecto a la corriente máxima de descarga, es decir la carga que va a tener que abastecer la batería. Para este caso la carga vendría a representar el módulo de Arduino Uno. Para ello es necesario aplicar al ecuación 8 que se muestra a continuación.

*Ecuación 8. Ecuación para determinar el tiempo de descarga que posee la batería [22].*

$$\text{tiempo de descarga} = \frac{\text{Corriente por hora}}{\text{Corriente de descarga}} \quad (8)$$

$$t = \frac{0.8Ah}{50mA} = \frac{0.8Ah}{0.05A} = 16 \text{ horas}$$

### **2.1.5. Selección de los componentes para el diseño e impresión de la Plaqueta de Circuito Impreso (PCB) de control del sistema domótico.**

En esta Plaqueta de Circuito Impreso (PCB) se utilizará como elemento de control a un Arduino Nano, ya que permite ahorrar espacio y costos en el diseño. Por otro lado, para poder activar los elementos del sistema domótico fue necesario disponer de transistores NPN 2N3904 configurados como elevadores de voltaje, ya que el servidor

domótico funciona de forma similar a un PLC, el cual maneja voltajes de entrada y salida de 24VCC.

Las resistencias utilizadas en el diseño son de ¼ de vatio, por la razón de que el sistema en general no maneja un voltaje y corriente que pueda ocasionar que estas se calienten, además que su tamaño favorece para que el diseño final del PCB no sea muy grande.

Otra etapa implementada en el PCB, es un regulador de voltaje para la alimentación del módulo de Arduino Nano. Este regulador debe manejar un voltaje de entrada de 24VCC del sistema domótico y proporcionar un voltaje de salida de 7VCC para el módulo de Arduino. Tomando en cuenta estas necesidades se seleccionó como regulador de voltaje al regulador LM317, debido a que maneja un voltaje de hasta 37VCC y una corriente de hasta 1.5A. Para obtener el valor del voltaje deseado en la salida, es necesario calcular el valor de las resistencias que se deben colocar en el regulador. A continuación se da a conocer el cálculo realizado para obtener el valor de las resistencias:

***Ecuación 9.** Ecuación para determinar el valor de las resistencias del regulador de voltaje [23].*

$$R2 = \left( \frac{R1}{\text{Voltaje de referencia}} \right) * (\text{Voltaje de salida} - \text{Voltaje de referencia}) \quad (9)$$

Tomando en cuenta esta ecuación y los siguientes valores se pudo obtener el valor de la R2.

R1=240ohmios

Voltaje de referencia=1.2VCC

Voltaje de salida= 7VCC

$$R2 = \left( \frac{240\text{ohmios}}{1.2\text{VCC}} \right) * (7\text{VCC} - 1.2\text{VCC}) = 1160\text{ohmios} = 1\text{K}\Omega$$

Una vez realizado el diseño del PCB se debe proceder a imprimirlo, para esta etapa se seleccionó una baquelita de fibra de vidrio, cuya elección se debe a su característica de manejar temperaturas altas. Para ello el diseño e impresión del PCB se da a conocer en el Capítulo 3.

---

## CAPÍTULO 3

---

### 3. DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

El presente capítulo tiene como finalidad informar sobre el diseño e implementación del sistema de reconocimiento de comandos de voz denominado “Domotic Voice” aplicado al control de sistemas domóticos. Por tal motivo este capítulo se lo dividido en tres partes.

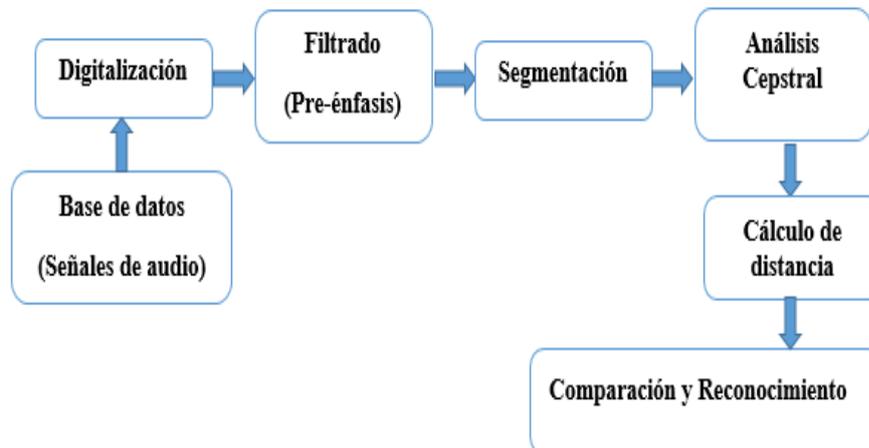
En la primera parte se hablará sobre el diseño considerado para la implementación del sistema de reconocimiento de voz; la forma en cómo se llevará a cabo la comunicación entre el sistema de reconocimiento de voz y el sistema domótico. Y por último la solución que se pretende dar con el sistema a los usuarios.

En la segunda parte se hablará sobre el diseño del hardware, en donde se dará a conocer el diseño de las carcasas, el diseño de la Plaqueta de Circuito Impreso (PCB) para el control del servidor domótico, y se explicará el método que se llevó a cabo para la impresión de este PCB.

Por último se hablará sobre el diseño del software, punto en el cual se dará a conocer la configuración de la tarjeta de reconocimiento de voz, la configuración de los módulos XBee, y se hablará sobre la programación que se llevó a cabo para obtener el sistema de reconocimiento de voz.

#### 3.1. Diseño y solución del sistema

Para diseñar la parte de reconocimiento de voz, es necesario tomar en cuenta la figura 3.1 que se muestra a continuación.



*Figura 3. 1. Diagrama de flujo para el reconocimiento de voz.*

*Fuente: Los autores.*

De acuerdo a la figura 3.1, en primer lugar se debe partir desde la creación de una base de datos, la misma que contiene todas las órdenes que el sistema deberá reconocer. Para crear la base de datos es necesario tomar en cuenta las acciones que llevará a cabo el sistema domótico, es decir los elementos del hogar que se van automatizar con los comandos de voz. Una vez considerado los elementos que se van a controlar, se debe grabar y digitalizar los audios para poder ser manipulados en el software donde se vaya a realizar el sistema de reconocimiento de voz.

Una vez digitalizado los archivos de audio, se debe proceder a la parte de filtrado, la cual es importante en el sistema, ya que permite eliminar ciertos ruidos que pueden generarse al momento de emitir la señal de audio a través del micrófono, con lo cual se garantiza que únicamente se conserve la señal de voz deseada para el procesamiento de la señal.

Posterior a la etapa de filtrado, se debe realizar la segmentación de la señal de audio, con la finalidad de extraer únicamente la información de la palabra emitida, es decir tomar un punto de inicio y un punto final.

A continuación se debe aplicar un método para el reconocimiento de voz, para el sistema se tomó como referencia el método de Análisis Cepstral, el cual consiste en realizar un análisis del espectro de la palabra emitida; para ello es importante obtener los coeficientes cepstrales que permitan extraer los parámetros de la señal de acuerdo a las variaciones del ancho de banda.

Finalmente se debe realizar la identificación de los patrones de voz, para lo cual es necesario calcular la distancia de los vectores característicos, con la intención de combinarlos y compararlos con la base de datos que se tiene como referencia.

El siguiente paso a considerar en el diseño, es la forma en cómo comunicar el sistema de reconocimiento de voz denominado “Domotic Voice” con el sistema domótico. Para dar solución a esta problemática, se tomó como referencia al protocolo de comunicación ZigBee, para ello se partió con la utilización de dos módulos XBee, los cuales se encargan de realizar la comunicación entre los dos sistemas, por medio de una configuración maestro-esclavo. Funcionando como un gateway de control que sirve como intermediario para la comunicación entre el sistema de reconocimiento de voz y el sistema domótico.

Con este diseño se pretende dar solución a todos los usuarios que deseen mejorar el confort dentro de su hogar, ya que únicamente por medio de comandos de voz podrán activar los sensores y actuadores que disponga su vivienda. Por otro lado, el sistema será de gran ayuda para las personas que sufran alguna discapacidad visual o motriz, debido a que podrán realizar el control autónomo de su hogar.

### **3.2. Diseño de hardware**

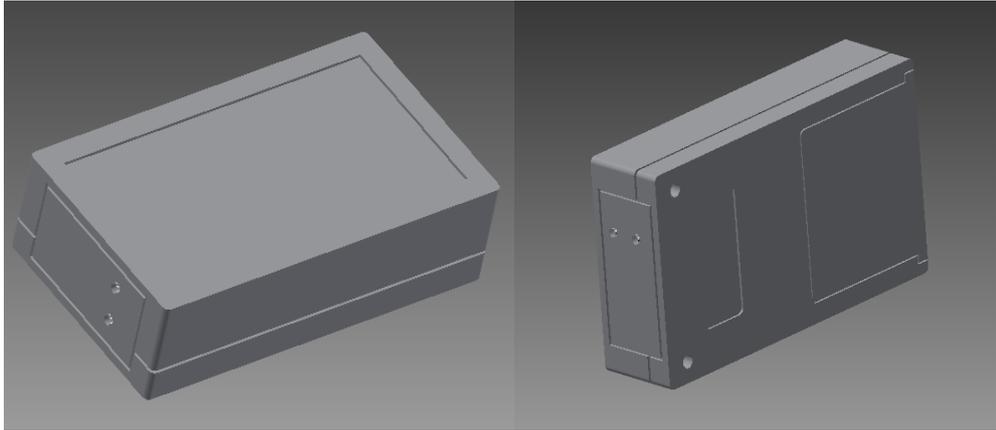
En el diseño de hardware únicamente se considera la parte de las carcasas y de la Plaqueta de Circuito Impreso (PCB) para el control del servidor domótico, debido a que los demás componentes del sistema de reconocimiento de voz tienen su propio hardware desarrollado, tal como es el caso de los módulos de Arduino, los módulos XBee y la tarjeta de reconocimiento de voz; lo cual resulta ser una ventaja en lo que respecta al tiempo que representaría el volver armar cada uno de estos módulos. Por tal motivo a continuación se dan a conocer el diseño de las carcasas y del PCB para el control del servidor domótico.

#### **3.2.1. Diseño de las carcasas**

El sistema de reconocimiento de voz está compuesto por dos carcasas, las mismas que tienen como finalidad dar mayor realce y presentación al diseño del sistema de reconocimiento de voz. Además que proporcionan permeabilidad a los componentes electrónicos, de manera que no puedan sufrir daños al entrar en contacto con la intemperie.

La primera carcasa es de plástico debido a que este material es fácil de moldear, posee un buen aislamiento eléctrico y sobre todo es ligero permitiendo que el usuario no tenga que soportar un peso exagerado que le pueda ocasionar un dolor muscular. Esta carcasa contiene a todos los componentes del sistema de reconocimiento de voz; estos componentes son el módulo de Arduino Uno, el módulo XBee maestro, la tarjeta de reconocimiento de voz y la batería de Lipo. Esta carcasa además cuenta con un cobertor de cuero, el cual le da un mayor realce y permite que el sistema de reconocimiento de voz pueda adaptarse al usuario, ya sea en su brazo o su cintura.

En la figura 3.2 se puede observar la carcasa plástica desde dos posiciones que va a servir de protección al sistema de reconocimiento de voz, la misma que fue diseñada en el software de Inventor.



**Figura 3. 2.** Carcasa plástica diseñada en Inventor.

*Fuente: Los autores.*

En la figura 3.3 se puede observar la carcasa plástica implementada que contiene el sistema de reconocimiento de voz.



**Figura 3. 3.** Carcasa plástica implementada.

*Fuente: Los autores.*

En la figura 3.4 se puede observar el cobertor de cuero que permite adaptar el sistema de reconocimiento de voz al usuario.

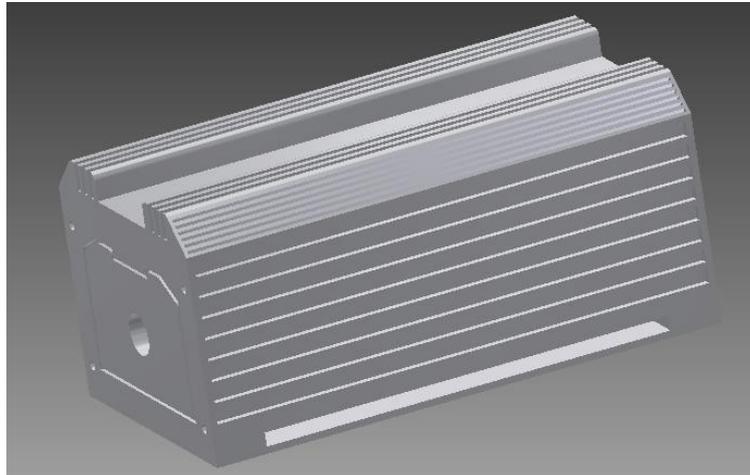


**Figura 3. 4.** Cobertor de cuero.

*Fuente: Los autores.*

El material seleccionado para la segunda carcasa, es el aluminio debido a que este material soporta mayor exposición a la intemperie y además que no tiende a oxidarse por la humedad que pueda existir en una vivienda. Con lo cual se garantiza una mejor protección al PCB que va ir ubicado a un lado del servidor doméstico.

En la figura 3.5 se puede observar la carcasa de aluminio que contiene al PCB de control de las entradas del servidor, la misma que fue diseñada en el software de Inventor.



**Figura 3. 5.** Carcasa de aluminio diseñada en Inventor.

*Fuente: Los autores.*

En la figura 3.6 se puede observar la carcasa de aluminio implementada que contiene el PCB de control de las entradas del servidor.



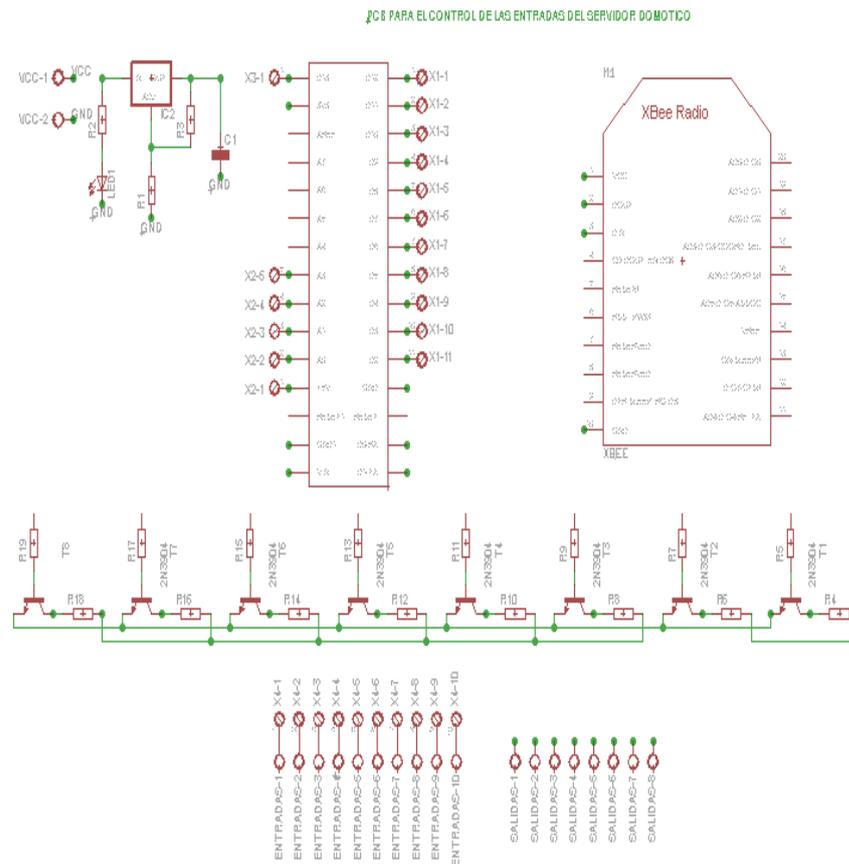
**Figura 3. 6.** Carcasa de aluminio implementada.

*Fuente: Los autores.*

### **3.2.2. Diseño de la Plaqueta de Circuito Impreso (PCB)**

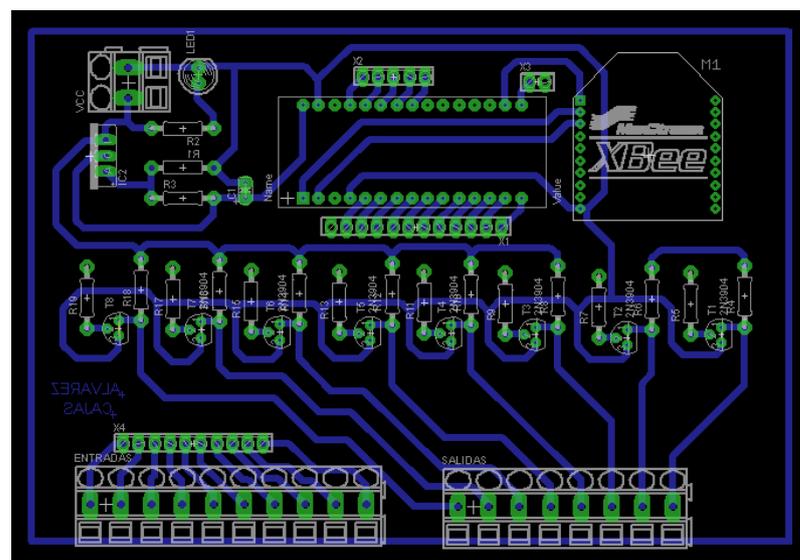
En lo que respecta al diseño de la Plaqueta de Circuito Impreso (PCB), esta fue realizada en el software Eagle con la finalidad de proporcionar un sistema que se pueda adaptar al servidor domótico. Este PCB está compuesto por un regulador de tensión que alimenta al módulo de Arduino Nano, el cual a su vez se encarga de alimentar al módulo XBee esclavo. Este módulo XBee esclavo es el encargado de recibir la señal del sistema de reconocimiento de voz y activar las entradas del servidor domótico por medio de transistores que están dispuestos en el PCB.

En la figura 3.7 se puede observar el diseño del esquemático que se utilizó para realizar el PCB de control de las entradas del servidor domótico.



**Figura 3. 7.** Esquemático del PCB de control para el servidor domótico.  
Fuente: Los autores.

En la figura 3.8 se puede observar el Layout del diseño del PCB de control para el servidor domótico.

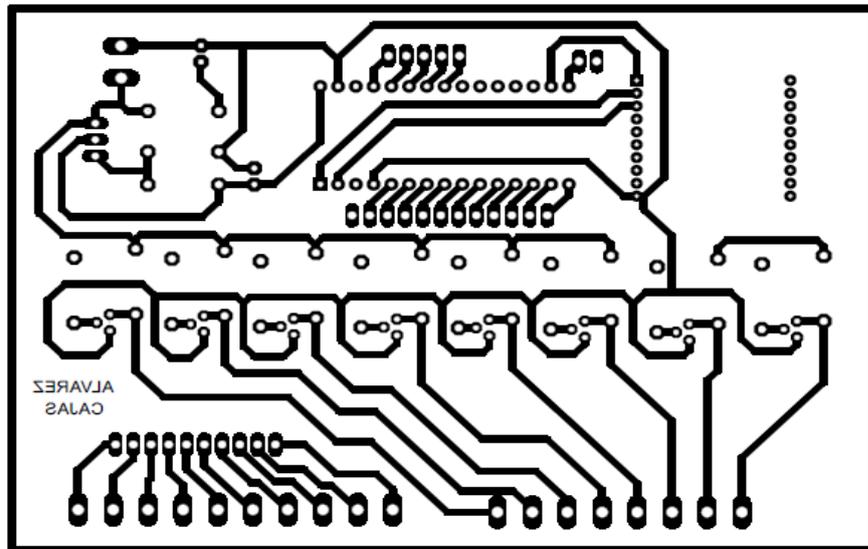


**Figura 3. 8.** Layout del PCB de control para el servidor domótico.  
Fuente: Los autores.

### 3.2.3. Impresión de la Plaqueta de Circuito Impreso (PCB)

Una vez obtenido el diseño final de la Plaqueta de Circuito Impreso (PCB), se procedió a imprimirlo de manera artesanal, para ello se debe tomar en cuenta lo siguiente:

- Imprimir el diseño del PCB en un papel fotográfico A4. En este punto se debe tomar en cuenta que la impresora debe ser del tipo láser. En la figura 3.9 se puede observar el diseño de las huellas del PCB que se imprimió sobre el papel fotográfico.



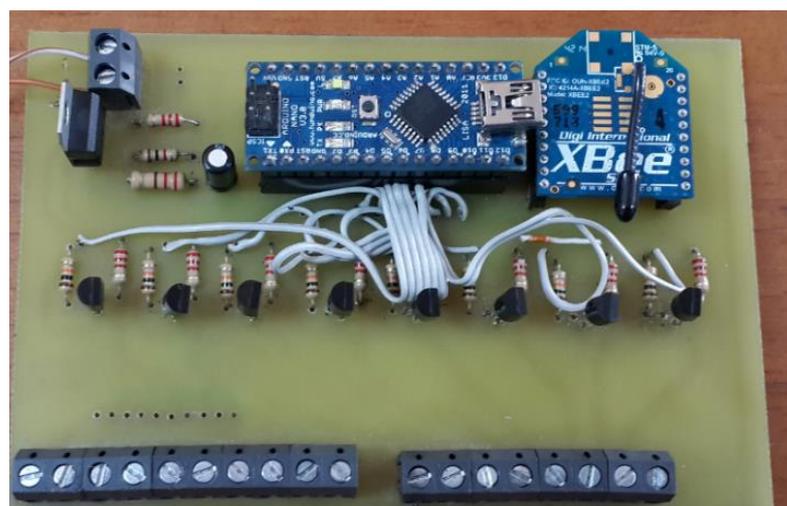
*Figura 3. 9. Diseño de las huellas del PCB.*

*Fuente: Los autores.*

- Dar un proceso de lijado por medio de una esponjilla a la baquelita de fibra de vidrio, esto ayudará a que el exceso de cobre se desprenda, así como se evitará que la grasa de los dedos quede impregnada al momento de quemar con el percloruro.
- Transferir la imagen del papel fotográfico a la baquelita de fibra de vidrio. Para ello es necesario pasar una plancha sobre el papel para que la imagen quede pegada en la baquelita. Posterior a esto se debe colocar la baquelita conjuntamente con el papel pegado en un recipiente con agua, lo cual provocará que la imagen del papel quede transferida en la baquelita.

- Después de haber transcurrido un tiempo prudente se debe sacar la baquelita del recipiente de agua y se debe proceder a desprender el papel lentamente para evitar que se borre el diseño.
- Proceder a quemar la baquelita en percloruro ferrito; en este punto es necesario colocar en un recipiente limpio el percloruro con una cantidad de agua adecuada, lo cual provocará que el diseño trasferido se mantenga en la baquelita y el resto de cobre se desprenda.
- Comprobar por medio de un multímetro que todas las pistas del circuito impreso tengan continuidad entre sí de acuerdo al diseño.
- Finalmente taladrar cada una de los pads del circuito impreso de acuerdo al diámetro que requiera cada componente electrónico.

Una vez impreso el PCB, se debe proceder a soldar cada uno de los componentes electrónicos en sus respectivos pads; en este punto es necesario tomar en cuenta la disposición de los pines de los componentes electrónicos, así como la temperatura de soldado con la finalidad de evitar que algún componente electrónico pueda averiarse debido a una mala conexión o por exceder la temperatura límite del componente. En la figura 3.10 se puede observar como quedo el PCB después de haber soldado cada uno de sus componentes.



*Figura 3. 10. PCB terminado.*

*Fuente: Los autores.*

### **3.3. Diseño de Software**

En este punto se da a conocer la forma en cómo se llevó a cabo la configuración de la tarjeta de reconocimiento de voz EasyVR 2.0, la configuración de los módulos XBee y la programación tanto del sistema de reconocimiento de voz, así como de la Plaqueta de Circuito Impreso (PCB) de control para el sistema domótico.

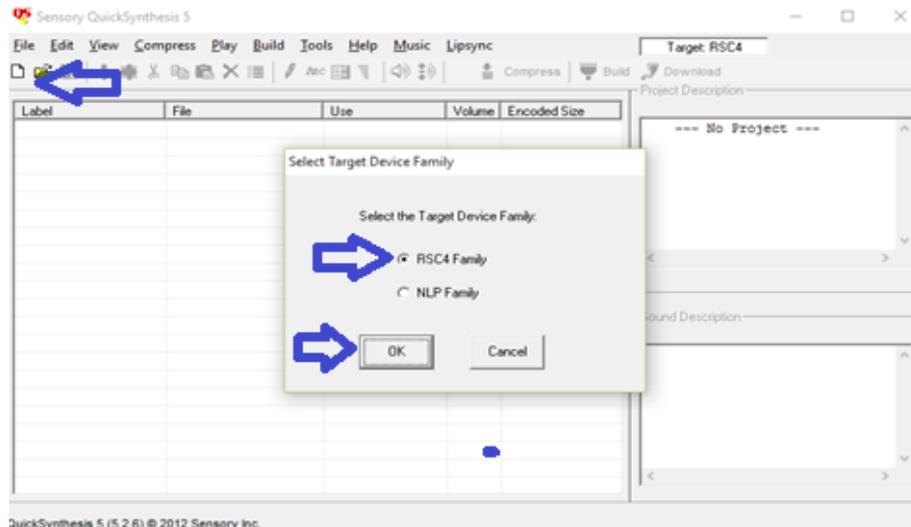
#### **3.3.1. Configuración de la tarjeta de reconocimiento de voz EasyVR 2.0**

Para llevar a cabo la configuración de la tarjeta EasyVR 2.0, es necesario obtener dos softwares que están disponible de manera gratuita en la Web. El primer software denominado EasyVR-Commander, es el que permite grabar los archivos de audio que deberá reconocer el sistema, al igual que permite realizar otras configuraciones que dispone la tarjeta de reconocimiento de voz. El segundo software denominado QuickSynthesis5, es el que permite crear un archivo compatible con el EasyVR-Commander. Este archivo es el que contiene cada uno de los comandos de audio grabados en formato WAV que van a ser reproducidos por la tarjeta EasyVR 2.0 al momento de reconocer una orden del usuario.

Los pasos a seguir para el uso de este software en la configuración de la tarjeta de reconocimiento de voz son los siguientes:

En primer lugar se debe partir desde la creación de los archivos de audio que va a reproducir la tarjeta EasyVR 2.0 al momento de reconocer una orden del usuario. Estos archivos de audio pueden ser digitalizados en cualquier software de edición de archivos de audio. Para nuestro caso estos archivos de audio fueron digitalizados en el software Audacity en formato WAV tipo mono a 8000Hz. La elección de este formato se debe a que el software QuickSynthesis5 reconoce este tipo de formato y además por que conserva la calidad de la voz emitida. Para poder crear el archivo de comandos de audio qxp que deberá reproducir la tarjeta EasyVR 2.0 se debe considerar lo siguiente:

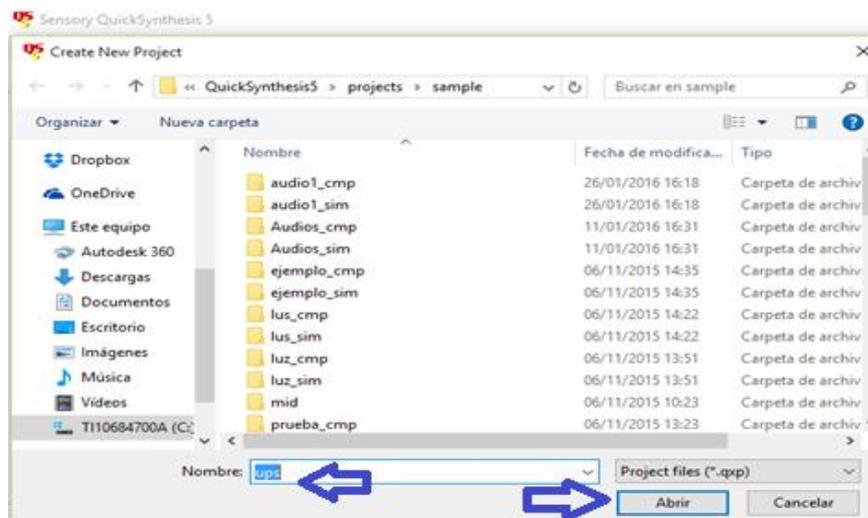
- 1.** Para que el programa funcione de manera correcta es necesario ejecutarlo como administrador. Una vez realizado este proceso se debe crear un nuevo proyecto, para ello es necesario dar un clic sobre la opción de crear nuevo proyecto ubicado en el menú superior del software. En la ventana que se despliega seleccionar la opción “RSC4 Family” y dar clic en OK. Esto se puede observar en la figura 3.11.



**Figura 3. 11.** Ventana para crear un nuevo proyecto.

*Fuente: Los autores.*

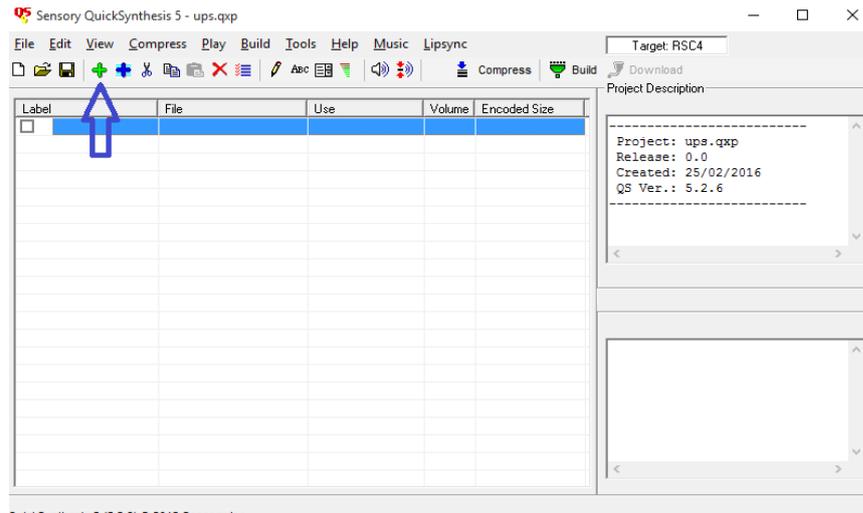
2. Colocar un nombre al proyecto, por ejemplo ups y dar un clic en la opción “Abrir”. En este punto es necesario darse en cuenta que el archivo a guardar debe ser del tipo “qxp”. Esto se puede observar en la figura 3.12.



**Figura 3. 12.** Ventana para dar un nombre al proyecto qxp.

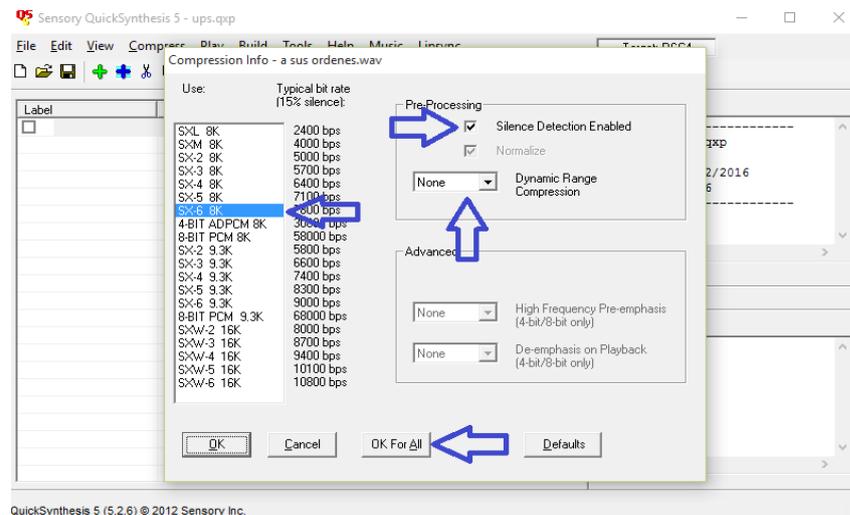
*Fuente: Los autores.*

3. Agregar los archivos de audio que deberá reconocer la tarjeta EasyVR 2.0. Esto se lo puede realizar dando un clic sobre opción “Add wav file(s) to project” disponible en el menú superior de la ventana del software. Esto se lo puede observar en la figura 3.13.



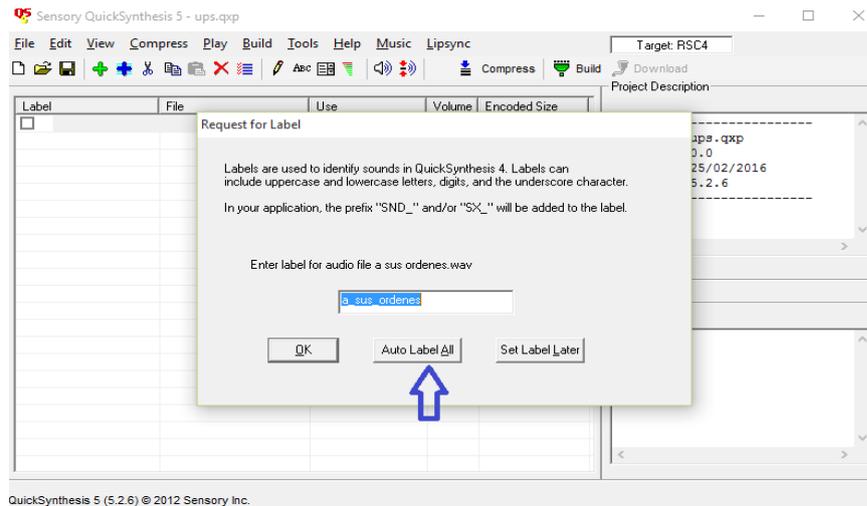
**Figura 3. 13.** Opción para agregar archivos de audio.  
Fuente: Los autores.

4. Seleccionar los archivos de audio que se van a utilizar para reproducir en la tarjeta EasyVR 2.0. Una vez realizada la selección aparecerá una ventana que permite comprimir los archivos de audio; en esta ventana se deben seleccionar cada una de las opciones que se muestran en la figura 3.14 y dar un clic en “OK For All”



**Figura 3. 14.** Ventana para comprimir los archivos de audio.  
Fuente: Los autores.

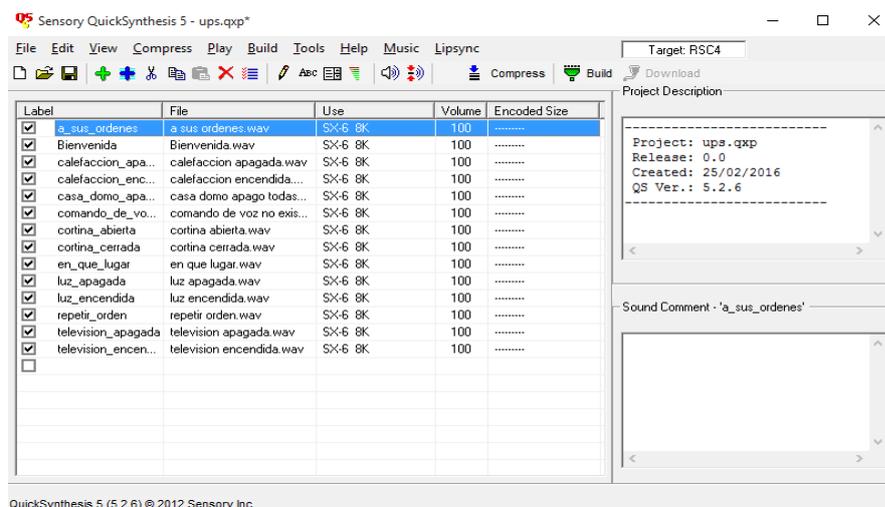
5. Una vez realizado el paso anterior aparecerá una ventana de solicitud de etiqueta, en la cual se debe seleccionar la opción de “Auto Label All” para cargar los archivos de audio. Esto se lo puede observar en la figura 3.15.



**Figura 3. 15.** Ventana de solicitud de etiqueta.

*Fuente: Los autores.*

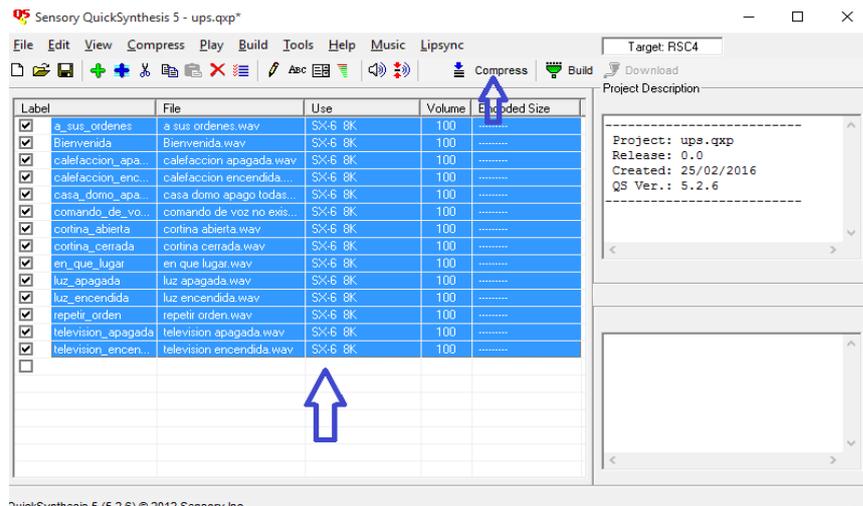
6. A continuación se podrá observar en la ventana del QuickSynthesis5 los archivos de audio que fueron cargados. Esto se lo puede observar en la figura 3.16.



**Figura 3. 16.** Ventana del software QuickSynthesis5 con los archivos de audio.

*Fuente: Los autores.*

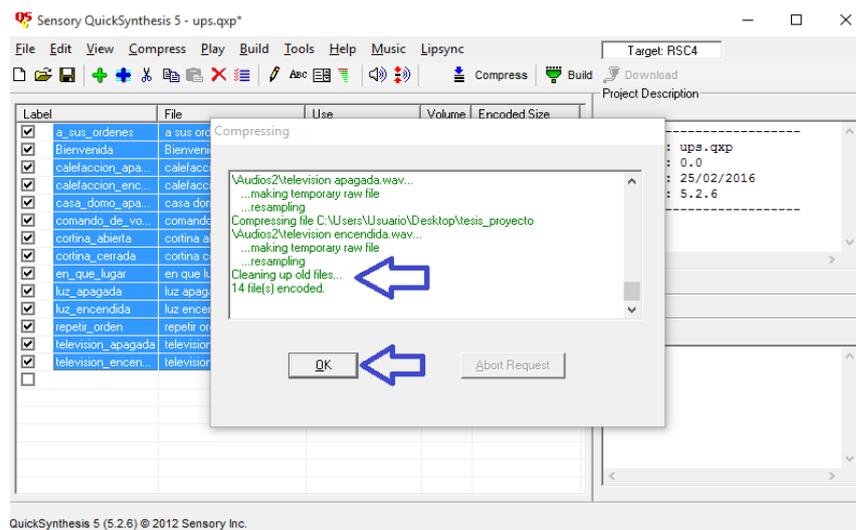
7. Después de haber cargado los archivos de audio se debe proceder a comprimirlos, para ello es necesario seleccionar cada uno de los archivos de audio y dar un clic en la opción “Compress” que se encuentra en el menú superior del QuickSynthesis5. Esto se puede observar en la figura 3.17.



**Figura 3. 17.** Opción para comprimir los archivos de audio.

Fuente: Los autores.

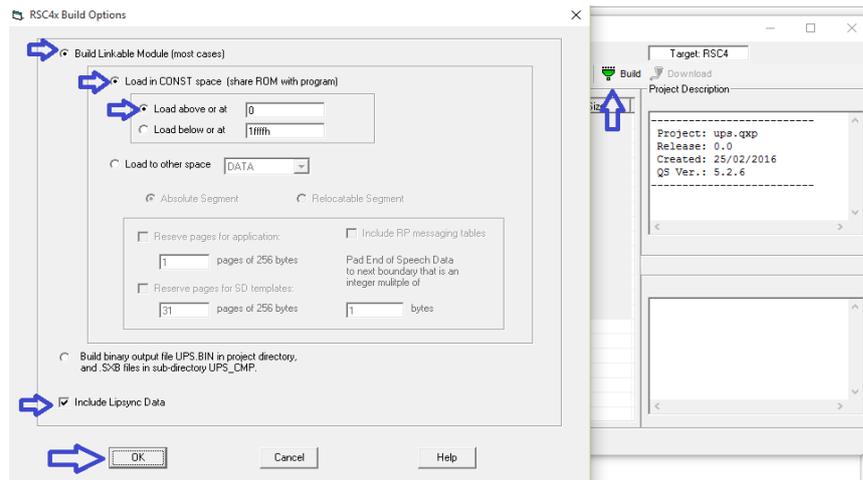
- Una vez dado clic en “Compress” se despliega una ventana que muestra este proceso, al finalizar aparecerá un mensaje en color verde que indicará que el proceso se realizó de manera exitosa. Dar un clic en “OK” para el finalizar el proceso. Esto se puede observar en la figura 3.18.



**Figura 3. 18.** Ventana para el proceso de comprimir los archivos de audio.

Fuente: Los autores.

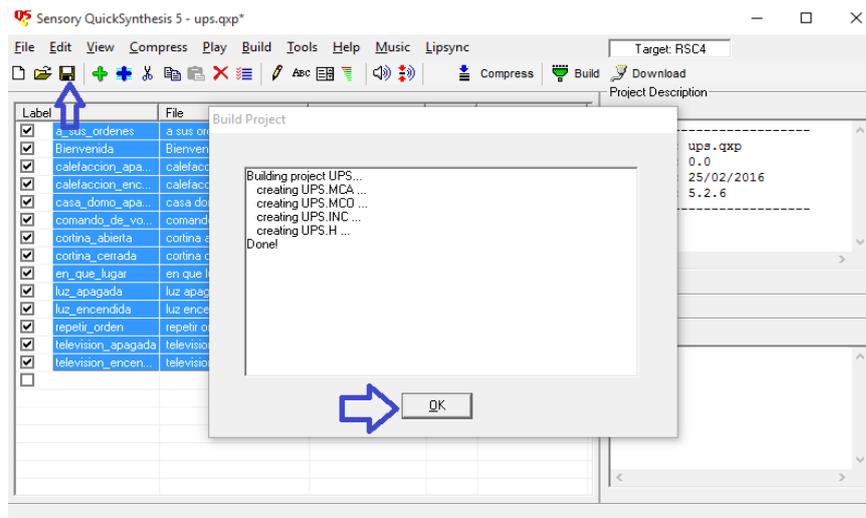
- Dar clic en la opción “Build” ubicado en el menú superior del software, esto permitirá construir el archivo con los comandos de audio que podrán ser utilizados en el software EasyVR-Commander para ser transferido a la tarjeta de reconocimiento de voz. En este punto es necesario seleccionar cada una de las opciones que se pueden observar en la figura 3.19 y dar un clic en la opción “OK”.



**Figura 3. 19.** Ventana de Build Options.

*Fuente: Los autores.*

10. Por último dar un clic en la opción “OK” de la ventana que se despliega del proceso de construcción del archivo. Además dar un clic en la opción guardar ubicado en el menú superior del software para evitar que se pierdan cada una los pasos realizados. Esto se puede observar en la figura 3.20.



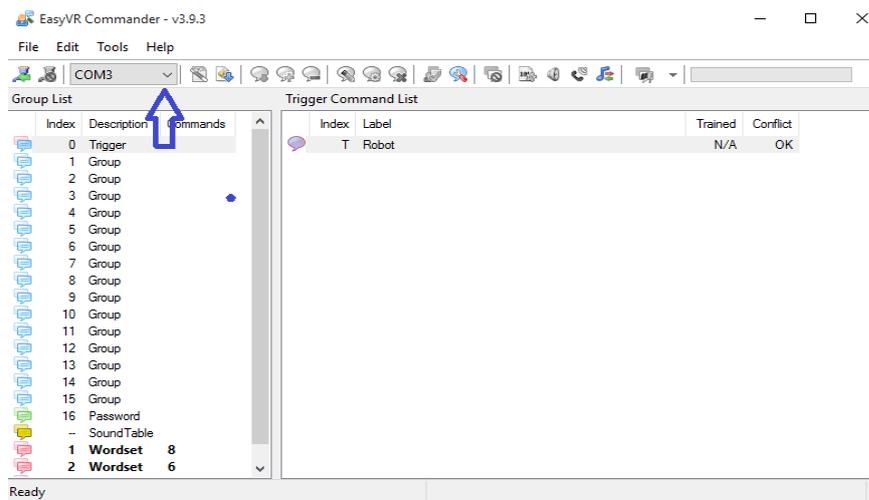
**Figura 3. 20.** Ventana de Build Project.

*Fuente: Los autores.*

Por otro lado para poder grabar el archivo de audio qxp creado en del software QuickSynthesis5 al software EasyVR-Commander, es necesario tomar en cuenta lo siguiente:

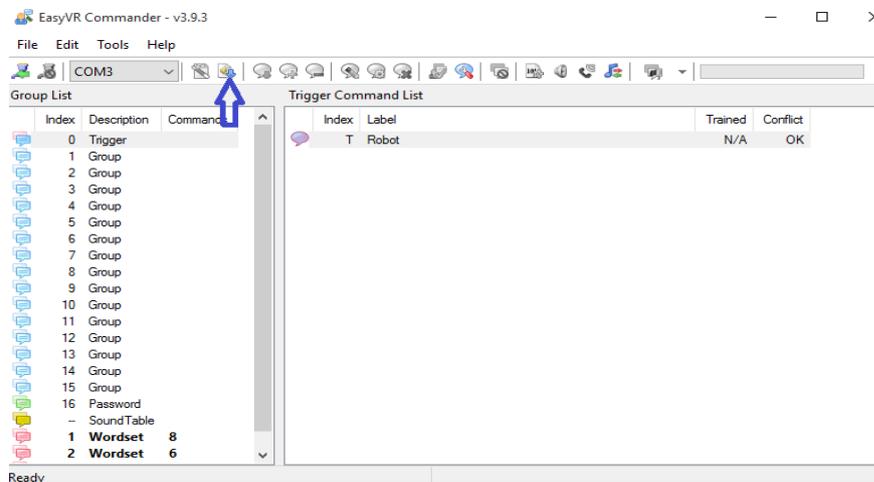
1. En primer lugar se debe conectar la tarjeta de reconocimiento de voz al módulo de Arduino Uno.
2. Colocar el jumper de la tarjeta EasyVR 2.0 en modo UP, lo cual permite grabar el archivo de comandos de audio qxp en la memoria interna que posee la tarjeta de reconocimiento de voz.

3. Conectar el módulo de Arduino Uno al computador.
4. Ejecutar como administrador el software EasyVR-Commander para que funcione de manera correcta y no de problemas al momento de configurar la tarjeta EasyVR 2.0.
5. Seleccionar el puerto en el cual se conecta el módulo de Arduino, esto es importante dado que permite establecer la comunicación entre el software y la tarjeta EasyVR 2.0. En la figura 3.21 se puede observar la ventana donde se debe seleccionar el puerto para establecer la comunicación.



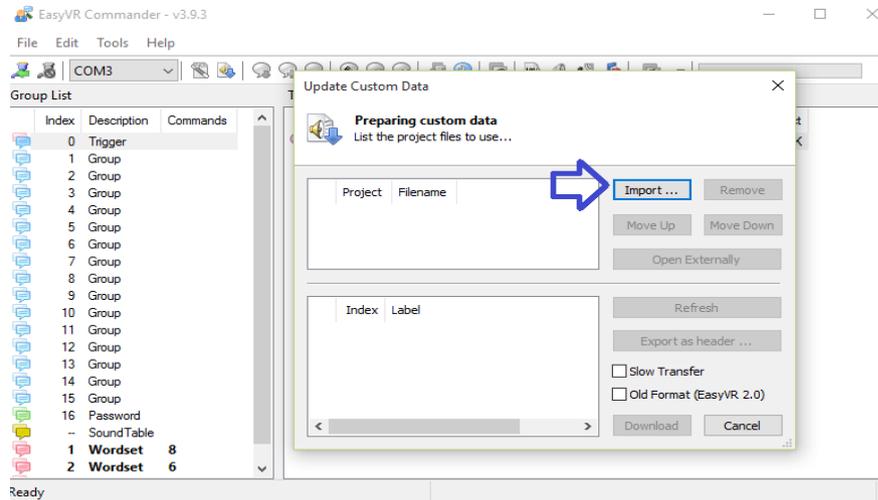
**Figura 3. 21.** Ventana para seleccionar el puerto de comunicación.  
Fuente: Los autores.

6. Lo siguiente que se debe hacer es subir el archivo de comandos de audio xqp a la memoria interna que posee la tarjeta EasyVR 2.0. Para ello es necesario dar un clic en la opción “Update Custom Data” ubicado en el menú superior del software EasyVR-Commander. En la figura 3.22 se puede observar la opción para subir el archivo xqp con los comandos de audio.



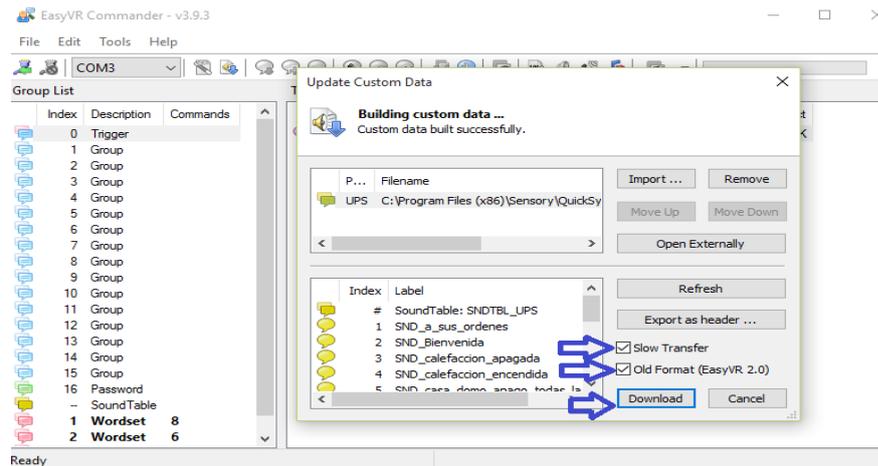
**Figura 3. 22.** Opción para subir el archivo xqp con los comandos de audio.  
Fuente: Los autores.

7. En la ventana que se despliega seleccionar la opción “Import”, la cual permite importar el archivo xqp que contiene los comandos de audio. Para nuestro ejemplo seleccionar el archivo “ups.qxp” que fue grabado del programa QuickSynthesis5. En la figura 3.23 se puede observar la ventana “Update Custom Data” para importar el archivo de audio xqp.



**Figura 3. 23.** Ventana Update Custom Data.  
Fuente: Los autores.

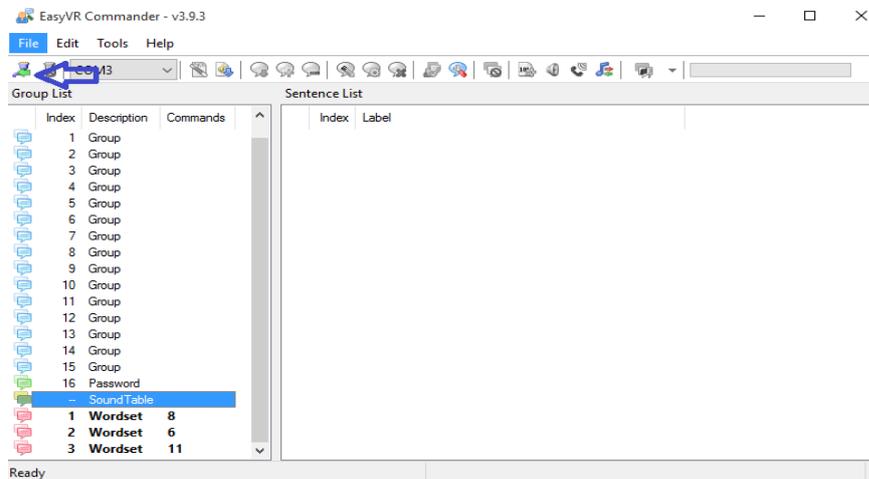
8. Una vez importado el archivo “ups.qxp” se debe proceder a descargarlo en la tarjeta EasyVR 2.0. En este punto se debe seleccionar las opciones de “Slow Transfer” y “Old Format (EasyVR 2.0)”. Luego dar un clic en la opción “Download” para descargar el archivo de audio. Esto se lo puede observar en la figura 3.24.



**Figura 3. 24.** Ventana Update Custom Data para descarga del archivo de audio.  
Fuente: Los autores.

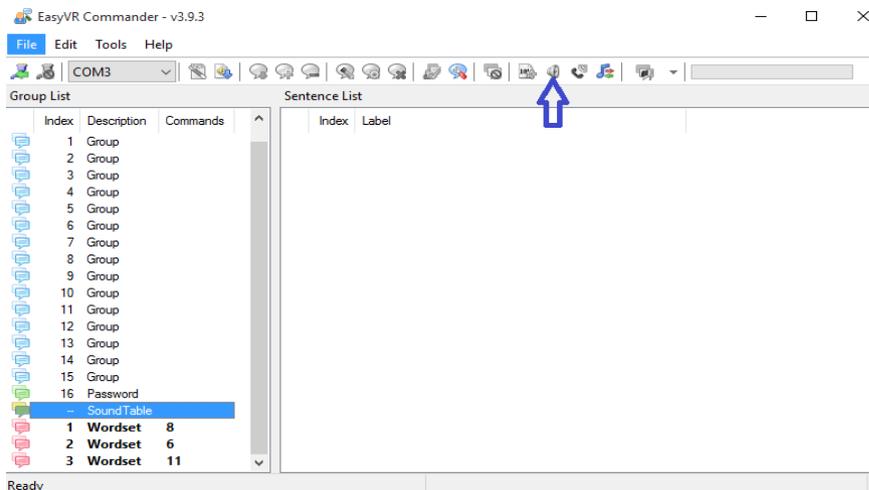
9. A continuación se puede proceder a probar los archivos de audio grabados. En este punto es necesario cambiar la posición del jumper de la tarjeta EasyVR 2.0 al modo PC.

10. Dar un clic en la opción “Connect” del menú superior del software EasyVR-Commander para comunicar la tarjeta con el software. En la figura 3.25 se puede observar la ventana donde seleccionar la opción “Connect”.



**Figura 3. 25.** Ventana para conectar la tarjeta EasyVR 2.0 con el software EasyVR-Commander.  
*Fuente: Los autores.*

11. Seleccionar el comando de audio que se desea reproducir y dar un clic en la opción “Play Sound (SPACE)”. Esto se lo puede observar en la figura 3.26.



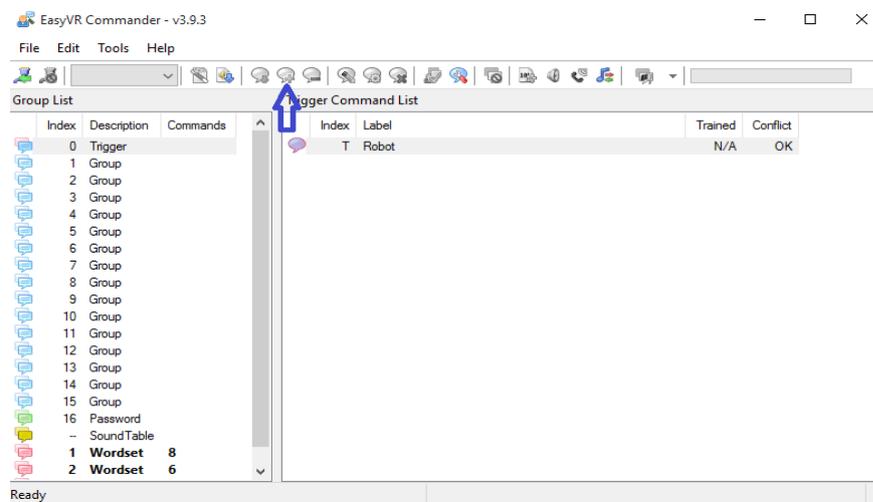
**Figura 3. 26.** Opción para probar los archivos de audio.  
*Fuente: Los autores.*

Una vez grabado los archivos de audio en la tarjeta EasyVR 2.0 para dar respuesta a las órdenes emitidas por el usuario, es necesario proceder a grabar en la tarjeta las órdenes que deberá reconocer el sistema “Domotic Voice”.

Los órdenes que se grabaron en el sistema de reconocimiento de voz son las siguientes: Casa, Sala, Comedor, Cocina, Dormitorio 1, Abre cortinas, Cierra cortinas, Apaga todo.

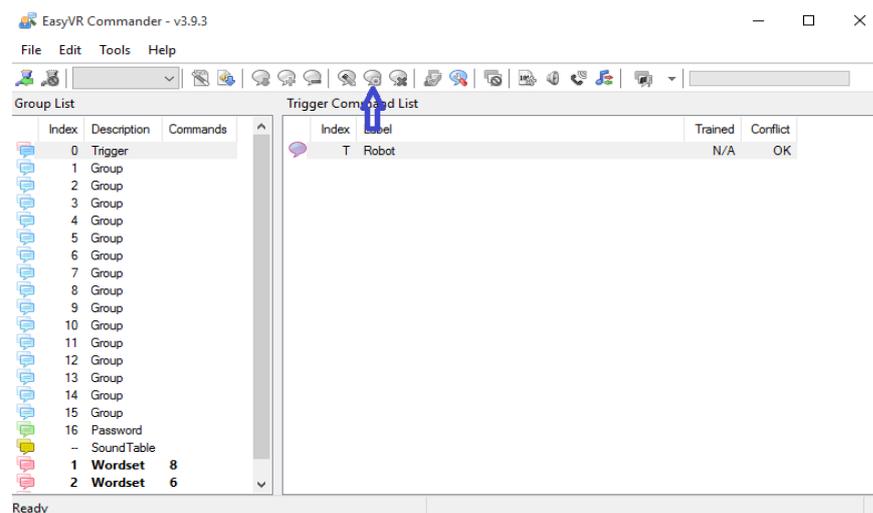
Para grabar cada uno de estos comandos de voz, es necesario tomar en cuenta el ambiente en el cual estos van a ser grabados, debido a que un ambiente ruidoso provocara que ruidos externos afecten al sistema al momento de reconocer la palabra que emita el usuario. Para grabar los comandos de voz se debe considerar lo siguiente.

1. Dar un clic en la opción “Add Command” ubicado en la parte superior de la ventana del software EasyVR-Commander para agregar un comando de voz. En este punto es necesario dar un nombre al comando que se va a crear. Esto se lo puede observar en la figura 3.27.



**Figura 3. 27.** Opción para crear nuevos comandos de voz.  
Fuente: Los autores.

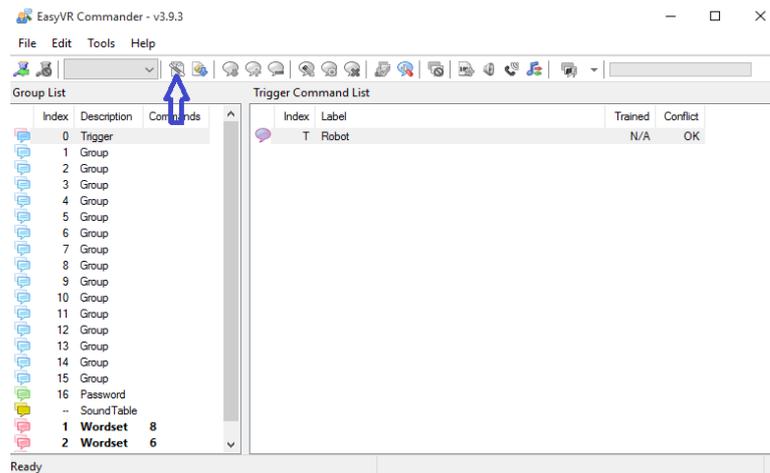
2. Entrenar cada uno de los comandos creados dando un clic en la opción “Train Command (ENTER)”, en la cual se debe repetir dos veces el mismo comando de voz para obtener mejores resultados al momento que el sistema reconozca la palabra emitida por el usuario. Esto se lo puede observar en la figura 3.28.



**Figura 3. 28.** Ventana de Train Command.  
Fuente: Los autores.

Por último para poder utilizar tanto los comandos de audio que va a reproducir la tarjeta EasyVR 2.0 y los comandos de voz que deberá reconocer el sistema, es necesario generar un código que permita utilizar estos archivos en la plataforma de Arduino para desarrollar el sistema de reconocimiento de voz. En este punto se debe tomar en cuenta lo siguiente:

1. Dar un clic en la opción “Generate Code” ubicado en la parte superior del software EasyVR Commander. Esto se lo puede observar en la figura 3.29.



**Figura 3.29.** Opción para generar código para Arduino.

*Fuente: Los autores.*

2. Guardar el archivo con un nombre y en la ubicación que se desee, en este punto es importante que tanto el archivo como la carpeta donde se va a guardar tengan el mismo nombre.
3. Para poder abrir el archivo generado por el software EasyVR-Commander en la plataforma de Arduino, es necesario que la plataforma cuente con la librería propia de la tarjeta EasyVR 2.0; con lo cual se podrá abrir el archivo y proceder a desarrollar el sistema de reconocimiento de voz.

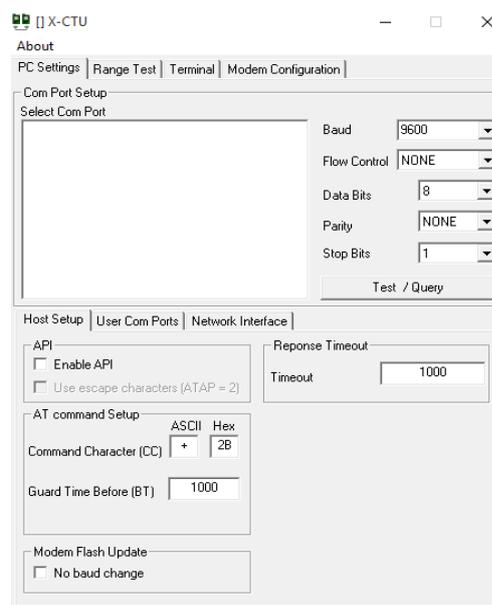
**Nota:** En cada una de las imágenes se han colocado flechas de color azul para orientar de mejor manera al lector sobre las opciones que se deben seleccionar para crear el archivo de audio en el software QuickSynthesis5 y a su vez para grabarlo en la tarjeta de desarrollo de reconocimiento de voz por medio del software EasyVR-Commander.

### 3.3.2. Configuración de los módulos XBee

En este punto es necesario configurar los dos módulos XBee que se van a utilizar en el desarrollo del sistema de reconocimiento de voz. El primer Módulo XBee debe ser configurado como maestro debido a que va a enviar las órdenes que reconozca el sistema “Domotic Voice” para activar el servidor domótico. Por otro lado, el segundo módulo XBee debe configurarse como esclavo para poder recibir las órdenes provenientes del XBee maestro y de esta manera actuar como un gateway que sirva de intermediario entre el sistema de reconocimiento de voz y el servidor domótico, con la finalidad de activar cada una de las entradas del servidor que activan los componentes del sistema.

A continuación se da a conocer cada una de los pasos que se deben seguir para configurar los módulos XBee por medio del Software X-CTU:

1. Ejecutar el software X-CTU como administrador. En la figura 3.30 se puede observar la ventana del software X-CTU.

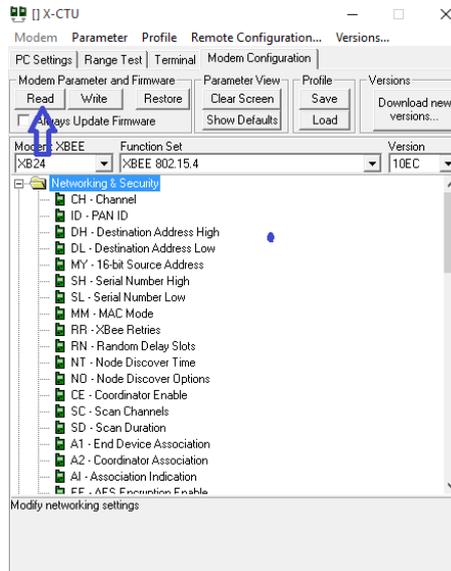


**Figura 3. 30.** Software X-CTU.

*Fuente: Los autores.*

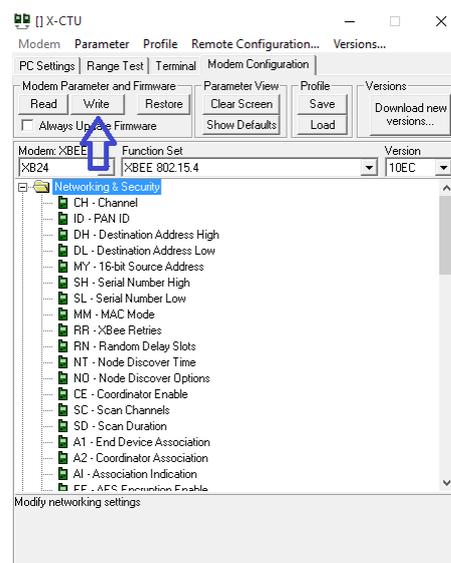
2. Conectar el módulo XBee al computador.
3. Seleccionar la velocidad de comunicación.
4. Seleccionar el puerto de comunicación y dar un clic en “Test/Query” para ver si X-CTU se puede comunicar con el módulo XBee. Al realizar esta operación aparecerá una ventana que mostrara los resultados de la comunicación. Finalmente dar un clic en “OK”.

5. En la opción “Modem Configuration” localizada en la parte superior del software X-CTU seleccionar “Read” para ver la configuración que contiene el módulo XBee. Esto se lo puede observar en la figura 3.31.



**Figura 3. 31.** Opción para leer la configuración inicial del XBee.  
Fuente: Los autores.

6. En “Modem XBee” seleccionar el modelo del módulo XBee, para nuestro caso seleccionar la opción XB24.
7. En “Function Set” seleccionar el papel que desempeñara el módulo XBee, seleccionar entre la opción de maestro o esclavo.
8. Dar un clic en la opción “Write” de la parte superior del software para cargar el firmware. Esto se lo puede observar en la figura 3.32.



**Figura 3. 32.** Opción para cargar el firmware.  
Fuente: Los autores.

9. Por último después de configurar cada módulo como maestro y esclavo, proceder a probar la comunicación por medio de la plataforma de Arduino, para lo cual es necesario disponer de las librerías de Arduino para la programación del módulo XBee.

### 3.3.3. Programación

En este punto se da conocer la manera de cómo se llevó a cabo la programación tanto del sistema de reconocimiento de voz, así como de la Plaqueta de Circuito Impreso (PCB) que permite realizar el control del sistema domótico. El código desarrollado para cada una de estos sistemas se lo realizó en la plataforma de Arduino. A continuación se da a conocer la manera de cómo se llevó a cabo la programación del sistema “Domotic Voice”:

En primer lugar se realizó un programa en Matlab con el propósito de analizar parámetros y determinar coeficientes de los filtros que necesita un sistema de reconocimiento de voz; los cuales tienen la secuencia de participación al momento del reconocimiento de la voz como se muestra en la figura 3.1. Este programa sirve para contrastar el funcionamiento del sistema desarrollado, ya que tendrá activados las mismas herramientas dentro del procesador. Para ello se ha dividido en partes la explicación de cada proceso que se debe considerar para realizar un sistema de reconocimiento de voz. A continuación se dan a conocer cada uno de ellos:

- **Pre-procesamiento de la señal de audio:** esto permite la adquisición y adecuación de las señales de voz para poder extraer la información más relevante; como su frecuencia, duración y eliminación de ruido, las cuales son necesarias para determinados procesos de la identificación de la voz. Para ello se aplica métodos de parametrización y normalización de la señal mediante el software Matlab, el cual facilita herramientas matemáticas para analizar el proceso de reconocimiento de cada uno de los comando de voz que se usó en el proyecto desarrollado.
- **Digitalización:** este proceso permite convertir la señal de entrada analógica en una señal digital, para posteriormente ser manipulada. Esta acción se la realizó en el software de Matlab mediante el comando `wavrecord(n,Fs,CH)`, el cual permite grabar con la ayuda de un micrófono señales de audio y las guarda en

un archivo de formato WAV; cabe destacar que este tipo de formato es uno de los más usados en el procesamiento de audio, debido a que conserva características importantes de la señal al ser comprimida y grabada en distintas calidades que van desde frecuencias de 11025Hz hasta 44100Hz, lo que da una mayor flexibilidad para determinar un tiempo de muestreo apropiado. En este caso se estableció una frecuencia de muestreo ( $F_s$ ) de 11025Hz y mono canal.

- **Filtrado:** esta parte está dividida en varios tramos, ya que el filtrado no solo sirve para eliminar el ruido que se adhiere a la señal al momento de adquirirla, sino también a la eliminación del silencio generado al momento de la grabación. También se aplica un filtrado para extraer datos específicos de la señal para su identificación. La eliminación del silencio consiste en eliminar las zonas de la grabación que no contengan algún tipo de información que sea útil; para ello se utiliza la ecuación de la energía en tiempo corto, lo cual provee una representación que refleje las variaciones en amplitud. El cálculo se lo realizó simultáneamente utilizando una estructura *For*, tomando muestras cada 10ms, para luego sacar un promedio de las energías y contrastarla con un porcentaje de la energía total que se toma como valor umbral para decidir si es una señal válida o se trata de silencio, en este caso se definió que la señal sea válida si es mayor o igual al 20% de la señal total.

Otro tipo de filtro utilizado es el filtro para resaltar formantes, el cual se encarga de enfatizar los formantes que no son más que los picos de intensidad en el espectro de un sonido; estos formantes se encuentran en las frecuencias altas de la señal de voz por lo cual es necesario un filtro pasa altas de orden uno para resaltar las frecuencias altas. Específicamente este filtro se implementa debido a la atenuación de la voz en las frecuencias altas, a consecuencia de la actuación del micrófono como un filtro pasa bajas, debido a su estructura de bobina que genera que se pierda este tipo de características de la voz. La ecuación que se aplica es la de un filtro lineal de primer orden, el cual es uno de los filtros de pre-énfasis más usados cuya constante típica en su función de transferencia es 0.95.

En Matlab se usa el comando filter ( $b$ ,  $a$ ,  $x$ ); en el cual el coeficiente  $b$  es el numerador de la función de transferencia,  $a$  es el denominador de la ecuación de la función de transferencia de orden uno y por último tenemos a  $x$  que es la señal que va ser filtrada.

- **Segmentación:** esto permite dividir en segmentos la señal de voz, de tal manera que se recorte en pedazos la longitud en un tiempo de 30 milisegundos; en esta etapa se puede utilizar una ventana de Hamming con un desplazamiento de 10 milisegundos entre tramo y tramo.
- **Extracción de las características de la señal de voz:** es la parte estructural de un sistema de reconocimiento de voz. Existen varias técnicas al respecto, tanto en el procesamiento digital de las señales, así como en el campo de las redes neuronales de las cuales el más utilizado dentro del procesamiento digital de señales es el análisis Cepstral, el mismo que ya se dio a conocer en el capítulo 1. Se ha tomado como referencia este método ya que no presenta una alta demanda en recursos de procesamiento y sus parámetros de obtención son estadísticamente fiables. De este método se extrae los coeficientes cepstrales; para ello una de las formas más comunes para el análisis de los patrones de voz, es mediante espectrogramas en donde por medio de herramientas matemáticas como la transformada de Fourier se puede obtener un espectrograma de la señal. La cual indica las componentes frecuenciales de dicha señal. De esta manera se extrae dos parámetros importantes que son el modelo que representa el tracto vocal y su excitación.

Por lo tanto el proceso dentro del programa consiste en tomar las señales previamente procesadas, es decir luego de la digitalización y enfatización de formantes; se procede a vectorizar cada señal y almacenarla dentro de una matriz cada uno de los vectores almacenados, para luego aplicar la transformada rápida de Fourier. Posterior a esto se toman los valores absolutos de la resultante de la transformada de Fourier y este resultado se lo representa en una escala logarítmica; para finalmente retornar a la señal en tiempo mediante la transformada rápida de Fourier.

- **Distancia:** Después de parametrizar a la señal y obtener puntos específicos, el paso siguiente es determinar la distancia entre estos puntos, para esto se usa el método de distancias euclidianas; lo cual nos permite determinar la palabra que fue emitida por el usuario. Para esto se hace una simple comparación de los vectores de distancia de la palabra emitida, con los vectores de las distancias de las palabras almacenadas en la base de datos.

Después de considerar cada uno de los resultados obtenidos en Matlab, se procedió a desarrollar el código en Arduino para el sistema de reconocimiento de voz, cuyas consideraciones se dan a conocer a continuación:

La plataforma SmartVR está desarrollada para aplicaciones de reconocimiento del habla, su modo de configuración depende de las aplicaciones que se deseen para lo cual tiene un entorno de desarrollo basado en código ensamblador. Esta plataforma usa una tarjeta que sirve tanto como entrenador y como grabador. En esta placa entrenadora se puede reajustar el programa base de la tarjeta de reconocimiento; activando o desactivando herramientas de filtrado que luego se incluirán en el microprocesador de la tarjeta de reconocimiento EasyVR, para luego poder darle una aplicación según los requerimientos.

Una vez que se ha configurado y grabado el código base con el cual el sistema “Domotic Voice” va a realizar la identificación de cada una de las palabras, se procedió a realizar un algoritmo en la plataforma de Arduino para poder interactuar con la tarjeta EasyVR. Este algoritmo realiza una interfaz entre estos dos dispositivos, que permite que la información sea bidireccional. Esto permite tomar los datos que genera la tarjeta de reconocimiento mediante una comunicación serial, los cuales son procesados por el módulo de Arduino y son convertidos posteriormente en comandos que ejecutan una acción determinada que está ligada a cada palabra. Los datos que envía la tarjeta de reconocimiento están codificados mediante direcciones de registro, las cuales también permiten dar eventos de error en el reconocimiento de la palabra. Dentro de estos eventos de error están los más comunes que son 65h que indica que el sistema no logró reconocer la palabra debido a un alto índice de ruido. También se suele generar el error 76h que indica que el tiempo establecido para reconocer la palabra fue excedido; esto sucede cuando el microprocesador no encuentra una relación entre lo que el sistema detecta en ese momento y su base de datos.

Los dos casos se los han agrupado para que el algoritmo desarrollado en Arduino identifique a cualquiera de ellos, envíe a la tarjeta de reconocimiento para que vuelva a leer el registro, y conjuntamente envíe al usuario un comando de voz que le pide repetir la orden principal.

Con la finalidad de probar la comunicación entre el módulo de Arduino y la tarjeta de reconocimiento se desarrolló un programa que proporciona el nombre del comando de voz que se ha pronunciado, mediante el monitor serial de la plataforma de Arduino.

En la figura 3.33 se puede observar las pruebas de comunicación realizadas con el programa antes mencionado, en donde se enseña brevemente las partes más importantes de este programa que permite comprobar la comunicación con la tarjeta de reconocimiento.

```
COM11
EasyVR detected, version 2
Sound table: TESIS8
Sound entries: 14
Custom Grammars: n/a
Trigger: 1 trigger
0 = CASA, Trained 2 times, OK
Group 1 has 11 command(s)
0 = SALA, Trained 2 times, OK
1 = COSINA, Trained 2 times, OK
2 = COMEDOR, Trained 2 times, OK
3 = DORMITORIO1, Trained 2 times, OK
4 = DORMITORIO2, Trained 2 times, OK
5 = ABRECOTINA, Trained 2 times, OK
6 = CIERRACORTIN, Trained 2 times, OK
7 = PRENDE_CALEF, Trained 2 times, OK
8 = APAGA_CALEF, Trained 2 times, OK
9 = LUCES, Trained 2 times, OK
10 = APAGA_TODO, Trained 2 times, OK
Say a command in Group 0
Timed out, try again...
Say a command in Group 0
Timed out, try again...
Say a command in Group 0
```

*Figura 3. 33. Prueba de comunicación.  
Fuente: Los autores*

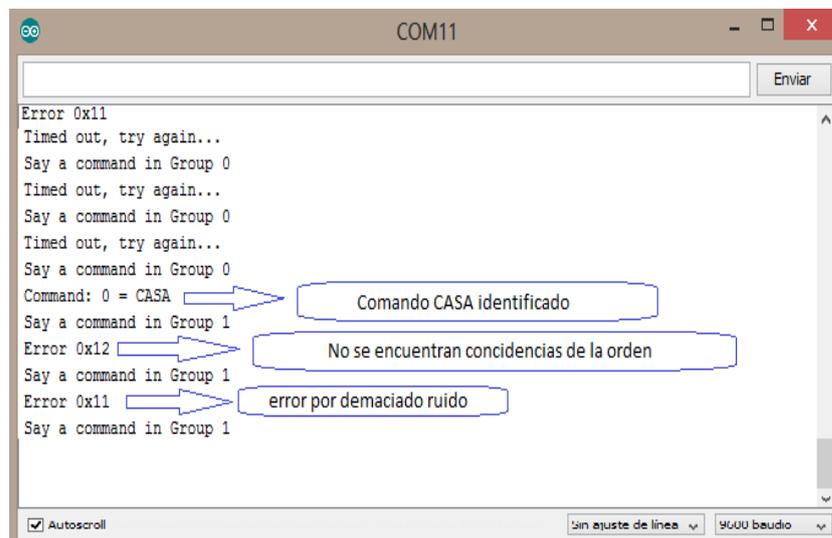
A continuación de acuerdo a la figura 3.33 se explica cada una de las partes que conforman la estructura del programa:

- **El encabezado:** esto permite verificar si existe o no un enlace con la placa de desarrollo, ya que indica si la placa ha sido detectada o no.

- **Lista de comandos:** la lista de comandos no es más que cada uno de los comandos que están registrados en la memoria del dispositivo, pero además indica si el reconocimiento ha sido correcto en el auto test.
- **Tiempo de desborde:** este parámetro es un tiempo que se considera para que el dispositivo intente hallar coincidencias con su base de datos antes de dar por válida o no una orden y pedir al usuario que la repita.

Dentro del algoritmo de testeo se ha incluido la identificación de dos tipos de errores que se imprimen en el monitor serial según ocurran en una representación hexadecimal.

En figura 3.34 se muestra la forma en cómo se presenta estos errores y que significan.



**Figura 3. 34.** Tipos de errores que detecta el programa del sistema “Domotic Voice”.  
Fuente: Los autores.

La manera como se realiza la captura del error y se lo presenta en el monitor serial de Arduino se muestra en la figura 3.35. En donde se presenta un extracto del código.



El sistema desarrollado permite apagar las luces con cada comando correspondiente al lugar donde se requiera apagar la luz. Además que el algoritmo incluye un comando adicional que permite apagar todas las luminarias que se encuentren encendidas de la casa; esto se lo realizó pensando en casos en los cuales se requiera apagar todas las luminarias y evitar el tener que apagar individualmente cada una de ellas.

Por último el dispositivo que permite realizar la interface con el servidor domótico, contiene un algoritmo que interpreta los datos recibidos por el receptor XBee y los convierte en un pulso que se envía por los respectivos puertos digitales del módulo de Arduino Nano. Estas salidas son flexibles al nivel de voltaje que se desee manejar en un rango de 7-24 voltios, y su comunicación para recibir los datos del XBee, es la de tipo serial a 9600 baudios.

---

## CAPÍTULO 4

---

### 4. FUNCIONAMIENTO

En este capítulo se dará a conocer el funcionamiento total del sistema de reconocimiento de voz “Domotic Voice”, cuya explicación se lo hará en dos partes. En la primera parte se hablará sobre el funcionamiento del sistema de reconocimiento de voz que permite enviar la señal de audio para activar los componentes del sistema domótico. En cambio, en la segunda parte se hablará sobre el funcionamiento de la Plaqueta de Circuito Impreso (PCB) de control del sistema domótico, el cual sirve de intermediario entre el sistema de reconocimiento de voz y el servidor domótico. A continuación se da a conocer el funcionamiento de cada uno de ellos:

#### 4.1. Funcionamiento del sistema de reconocimiento de voz

El sistema de reconocimiento de voz está desarrollado de tal manera que pueda ser utilizado desde cualquier parte de la vivienda siempre y cuando no se exceda el alcance que poseen los módulos XBee para realizar la comunicación entre el sistema de reconocimiento de voz y la Plaqueta de Circuito Impreso (PCB) de control del sistema domótico. Además que es adaptable y cómodo para el usuario, ya que únicamente deberá colocar el sistema en su brazo o su cintura para proceder a utilizarlo.

El sistema “Domotic Voice” está compuesto por un módulo de Arduino Uno que funciona como un controlador del sistema de reconocimiento de voz, ya que es el encargado de realizar la comunicación interna entre la tarjeta EasyVR 2.0 y el módulo XBee maestro, con lo cual se pueda transferir las órdenes hacia el PCB de control del sistema domótico. Estas órdenes son emitidas por medio del micrófono incorporado en el sistema, las mismas que deben ser reconocidas por la tarjeta EasyVR 2.0 y enviadas por medio de la comunicación inalámbrica ZigBee. Las órdenes deben emitirse siempre con una señal de activación denominada “Casa” para que el sistema entre en funcionamiento y así evitar una activación no deseada de los componentes del hogar por parte del usuario. Por ejemplo, si el usuario desea encender o apagar las luces de la cocina, este deberá decir al sistema de reconocimiento de voz la palabra “Casa” antes de dar la orden, en este caso para encender o apagar la luz de la cocina es “Cocina”.

De igual manera se debe considerar para el resto de órdenes que el sistema deberá reconocer, es así que para el caso de encender o apagar las luces del comedor, del dormitorio 1 y de la sala se debe dar antes la orden de “Casa” y por consiguiente las órdenes de “Comedor”, “Dormitorio 1” y “Sala” respectivamente para encender o apagar las luminarias. En lo que respecta a las cortinas las órdenes que se deben emitir después de dar la orden de “Casa” son “Abre cortinas” y “Cierra cortinas” para cada proceso. Además de la orden de “Apagar todo” para apagar todo lo que se encuentre activado en ese momento por el sistema.

La orden que emite el usuario es procesada por el sistema de reconocimiento de voz y transmitida de forma serial por medio de una letra que representa dicha orden, de manera que puedan ser interpretadas por el PCB de control del sistema domótico. Con esto se garantiza que cada proceso se lleve a cabo de manera segura y así se evite conflictos con el control del sistema domótico. Este proceso se lo lleva cabo por medio del módulo XBee maestro que envía estas señales para que puedan ser interpretadas por el módulo XBee esclavo, y así llevar a cabo la activación y desactivación de los componentes del sistema domótico.

Dentro del funcionamiento del sistema de reconocimiento de voz, es necesario tomar en cuenta la parte de respuestas que proporciona el sistema al momento que el usuario emite una orden, por ejemplo al momento de encender las luminarias, apagar las luminarias, abrir las cortinas, cerrar las cortinas, la tarjeta EasyVR 2.0 informa al usuario que estas órdenes fueron realizadas con éxito en el servidor domótico, lo cual es ideal para que exista una buena relación de comunicación entre el sistema y el usuario. Las respuestas que proporciona el sistema son: “Bienvenido a casa”, “A sus órdenes”, “Repetir orden”, “Luz encendida”, “Luz apagada”, “Apagado todo”, “Cortina abierta” y “Cortina Cerrada”.

Las respuestas del sistema de reconocimiento de voz fueron grabadas dentro de la memoria interna que posee la tarjeta EasyVR 2.0 de acuerdo a un proceso de configuración que se dio a conocer en el capítulo 3. Estas órdenes fueron grabadas y digitalizadas previamente en el software de Audacity de acuerdo al formato que requiere la tarjeta. En la figura 4.1 se puede observar el sistema de reconocimiento de voz “Domotic Voice” desarrollado.



*Figura 4. 1. Sistema de reconocimiento de voz “Domotic Voice”.  
Fuente: Los autores.*

#### **4.2. Funcionamiento de la Plaqueta de Circuito Impreso (PCB) de control del sistema domótico**

La Plaqueta de Circuito Impreso (PCB) de control funciona como un gateway intermediario que permite realizar la comunicación entre el sistema de reconocimiento de voz y el sistema domótico, de tal manera que las órdenes enviadas por el XBee maestro puedan ser ejecutadas en el servidor domótico. Este PCB tiene como elemento de control a un Arduino Nano que permite la comunicación interna entre el módulo XBee esclavo y el servidor domótico.

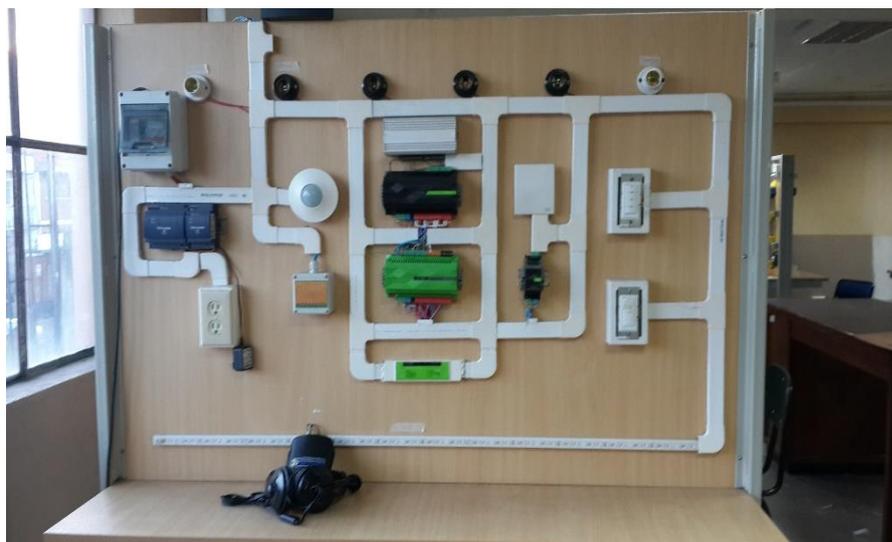
Las órdenes enviadas por el sistema de reconocimiento de voz son recibidas y enviadas de forma serial al Arduino, el cual interpreta una orden como una letra que activa y desactiva una entrada del servidor domótico. Las letras que interpreta el módulo XBee esclavo son las mismas que fueron definidas en el sistema de reconocimiento de voz para cada una de las órdenes que emita el usuario.

Una vez recibida una orden enviada por el sistema de reconocimiento de voz, procederá a activar o desactivar una entrada del módulo de Arduino Nano, el cual a su vez permite activar o desactivar una entrada del servidor domótico por medio del uso de transistores NPN 2N3904 configurados como emisor común que funcionan como elevadores de voltaje, ya que el servidor domótico funciona de forma similar a un PLC, el cual maneja voltajes de entrada y salida de 24VCC.

Al utilizar este PCB se consigue que las entradas del servidor domótico funcionen de acuerdo a las órdenes de voz que emita el usuario en el sistema “Domotic Voice”, sin provocar cambios en lo que respecta al funcionamiento de los componentes del sistema domótico de acuerdo a su programación de fábrica. Convirtiéndose únicamente en un gateway intermediario para que el sistema de reconocimiento de voz se adapte al sistema domótico y lo pueda controlar por medio de comandos de voz.

Con lo que respecta al funcionamiento del sistema domótico LOXONE, se debe considerar los tiempos de retardo que poseen los elementos al momento de ser controlados por el sistema de reconocimiento de voz “Domotic Voice”, tal es el caso que para activar las luminarias y las persianas se debe esperar un tiempo de 5 segundos después de haber dado la orden. Sin embargo se debe considerar que es necesario esperar un tiempo de 20 segundos antes de dar una segunda orden a las persianas, ya sea para subirlas o bajarlas, debido a que el sistema domótico LOXONE tiene por defecto en su programación un tiempo de retardo de 20 segundos antes de poder interpretar una nueva orden como protección del motor que realiza el accionamiento de las persianas. Por ejemplo, si se activa de forma manual o por medio de comandos de voz las persianas antes del tiempo establecido, el sistema domótico LOXONE no permite volver a utilizar las persianas, para lo cual es necesario tener que reiniciar todo el sistema.

En la figura 4.2 se puede observar el PCB de control implementado en el banco del sistema domótico LOXONE.



**Figura 4. 2.** Gateway de control del sistema domótico.  
*Fuente: los autores.*

---

## CAPÍTULO 5

---

### 5. ANÁLISIS DE RESULTADOS

En este capítulo se da a conocer los resultados obtenidos del sistema de reconocimiento de voz, a partir de las consideraciones iniciales que se tomaron en cuenta para su desarrollo y los cambios que se fueron realizando hasta obtener el sistema final implementado que se utilizó para realizar las pruebas y obtener los resultados que demuestran su funcionalidad. Así como el análisis económico del sistema desarrollado el cual permite ver la viabilidad del mismo. A continuación se dan a conocer los resultados obtenidos del sistema de reconocimiento de voz y su análisis económico:

#### 5.1. Análisis del sistema “Domotic Voice”

Para poder comunicar el sistema de reconocimiento de voz a un sistema domótico en el cual se pueda manipular y controlar las entradas del servidor, se desarrolló un PCB que funciona como un gateway intermediario que permite realizar la comunicación entre los dos sistemas. El dispositivo desarrollado interviene las señales digitales que controlan el servidor domótico y las combina con las señales del sistema de reconocimiento de voz; resultando ser una ventaja frente a una de las propuestas que se consideró inicialmente, la cual consistía en manipular las variables de los sistemas domóticos mediante una conexión TCP/IP, lo cual no fue viable debido a que ciertos sistemas como KNX usan códigos encriptados. Por tal motivo era necesario obtener una licencia del software para su manipulación, lo cual agrega un valor adicional al dispositivo que puede provocar que el sistema desarrollado se vuelva inaccesible en nuestro medio. Tomando en cuenta cada una de estas consideraciones se optó por el método presentado en este documento, el cual mediante la intervención de estradas y salidas nos permite tener un resultado satisfactorio en cuanto a proyectos no muy extensos. También se ha considerado la ventaja de no tener que manipular el código que tiene por defecto cada sistema domótico para llevar a cabo la configuración del sistema desarrollado al momento de la instalación, ya que de esta manera la conexión se vuelve más sencilla, y por lo tanto no se requiere que la persona que lo instale tenga conocimientos extensos en la materia, sino simplemente sea capaz de seguir un diagrama de conexión.

Dentro de las pruebas que se realizaron para establecer un dispositivo que nos pueda ofrecer un buen alcance y nos garantice robustez en la transmisión de datos de forma inalámbrica se puso a consideración dos módulos Wireless ESP8266 para establecer la comunicación entre el sistema de reconocimiento de voz y el PCB de control del servidor domótico. Con lo cual no se obtuvo un buen resultado, ya que se perdía la comunicación cuando existían objetos sólidos que interferían la transmisión de la señal. Además que solo se tenía una cobertura de 10 metros como máximo para emitir una orden hacia el servidor domótico. Esto resulta ser una desventaja para el usuario al tener que utilizar el sistema únicamente cerca del servidor domótico, eliminando la parte del confort que debería tener un usuario dentro de su vivienda. Para descartar estos módulos se realizaron pruebas dentro del laboratorio de Instalaciones Civiles con el sistema domótico LOXONE que posee la Universidad Politécnica Salesiana sede Cuenca; donde se emitieron órdenes que el sistema de voz reconocía, pero que sin embargo no podían ser transmitidas debido a que se superaba el alcance que poseen los módulos. Conforme se realizaban las pruebas estos módulos tendían a perder efectividad en la transmisión de la señal, debido a que no atravesaban objetos sólidos contundentes, limitando al sistema a tener que estar dentro de un rango directo de comunicación con el PCB de control del servidor domótico. Es decir se llegó a un punto que para poder comunicar los dos sistemas fue necesario tener que apuntar el sistema de reconocimiento de voz al PCB de control del servidor domótico para poder establecer la comunicación. Tomando en cuenta estas consideraciones se descartaron estos módulos como elementos de comunicación entre los dos sistemas.

Posterior a los módulos Wireless se consideró para el desarrollo del sistema de reconocimiento de voz dos módulos XBee que permiten realizar una comunicación inalámbrica ZigBee. Para ello estos módulos fueron puesto a prueba de igual manera dentro del laboratorio de Instalaciones Civiles con el sistema domótico LOXONE; donde demostraron una respuesta favorable en lo que respecta al rango de cobertura que poseen estos módulos, ya que los comandos de voz pudieron ser transmitidos fuera del laboratorio a una distancia aproximada de 30 metros, logrando reconocer el PCB de interfaz del servidor domótico fácilmente estos comandos. Además que se pudo constatar que no es necesario que el sistema de reconocimiento de voz apunte directamente al PCB de control del servidor domótico para poder comunicarse, lo cual es una ventaja, ya que el usuario puede utilizar el sistema “Domotic Voice” sin

importar los objetos que se encuentren obstaculizando su comunicación. Con estas pruebas realizadas se optó por utilizar estos módulos XBee como elementos de comunicación para los dos sistemas desarrollados.

En lo que respecta al sistema de reconocimiento de comandos de voz, este fue puesto a prueba con diferentes personas para poder obtener un sistema capaz de reconocer diferentes voces. Estas pruebas fueron realizadas a personas de sexo masculino y femenino con diferentes tonalidades de voz con la finalidad de probar la robustez del sistema cuando se lo somete a diferentes usuarios. Para ello fue necesario reajustar los valores en el proceso de filtrado del sistema “Domotic Voice” para las distintas tonalidades que poseen estas personas, logrando de esta manera que el sistema generalice la voz de la mayoría de usuarios. Sin embargo a pesar de mejorar el proceso de filtrado, el sistema “Domotic Voice” posee una desventaja en lo que respecta a los tonos de voz muy finos y débiles que poseen algunas personas como por ejemplo mujeres, ya que el sistema tiende a no reconocer el cien por ciento de las órdenes que estas personas emiten, para lo cual sería necesario tener que configurar el sistema exclusivamente para estas personas.

Para poder validar el sistema “Domotic Voice, este fue puesto a prueba para determinar su funcionalidad; para ello se realizó una prueba que permite determinar el nivel de ruido que soporta el sistema para funcionar de manera correcta; otra prueba que se llevó a cabo fue para determinar el tiempo de respuesta que posee el sistema con respecto a otros sistemas de reconocimiento de voz; y por último se realizó una prueba para determinar la capacidad que posee el sistema para reconocer una orden con diferentes personas. A continuación se dan a conocer los resultados obtenidos de cada una de las pruebas:

#### **5.1.1. Prueba para determinar el nivel de ruido que soporta el sistema.**

El sistema fue puesto a prueba mediante un sonómetro AMPROBE SM-20 que pertenece a la Universidad Politécnica Salesiana sede Cuenca y con el sonómetro Sound Meter para Android, con la finalidad de determinar el nivel de ruido que puede soportar el sistema para funcionar de forma fiable y correcta.

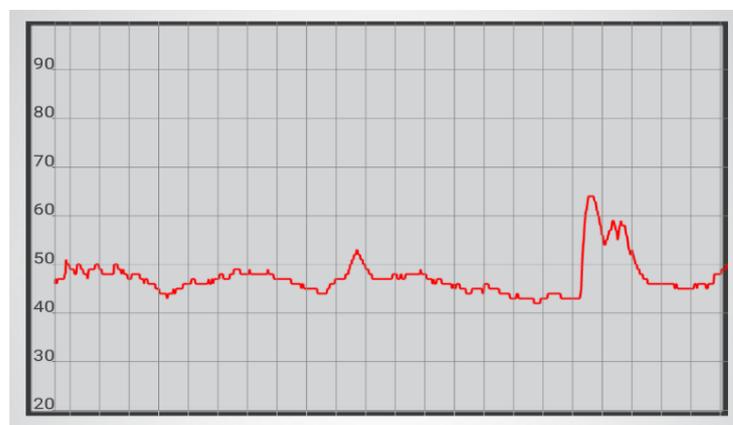
Para ello, en primer lugar se determinó el nivel mínimo y máximo de ruido que posee el laboratorio de Instalaciones Civiles en condiciones normales, es decir sin personas que se encuentren dentro del laboratorio y que puedan generar ruido al momento de utilizar el sistema de reconocimiento de voz, pero sin desestimar el ruido del ambiente. Además que en estas condiciones se grabó la base de datos de las órdenes que debe reconocer el sistema, así como la base de datos de respuestas que proporciona el sistema.

En la figura 5.1 se puede observar el valor mínimo y máximo de ruido que posee el laboratorio de Instalaciones Civiles en condiciones normales.



**Figura 5. 1.** Nivel de ruido en condiciones normales.  
*Fuente: Los autores.*

En la figura 5.2 se puede observar la gráfica de los niveles de ruido proporcionada por la aplicación Sound Meter en el laboratorio de Instalaciones Civiles en condiciones normales.



**Figura 5. 2.** Señal de ruido en condiciones normales proporcionado por Sound Meter.  
*Fuente: Los autores*

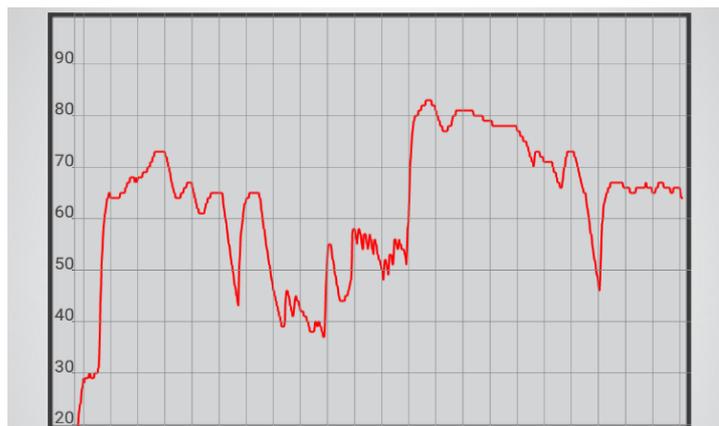
Posterior a esto se procedió a medir el nivel mínimo y máximo de ruido con los cuales el sistema de reconocimiento de voz tiende a presentar problemas al momento que el usuario emite una orden y esta no es reconocida por el sistema. Para ello fue necesario generar cierto ruido dentro del laboratorio de Instalaciones Civiles para poder realizar esta prueba.

En la figura 5.3 se puede observar el valor mínimo y máximo de ruido con los cuales el sistema tiende a presentar problemas.



**Figura 5. 3.** Nivel de ruido que genera problemas al sistema.  
*Fuente: Los autores*

En la figura 5.4 se puede observar la gráfica de los niveles de ruido con lo cual el sistema tiende a presentar problemas en su funcionamiento.

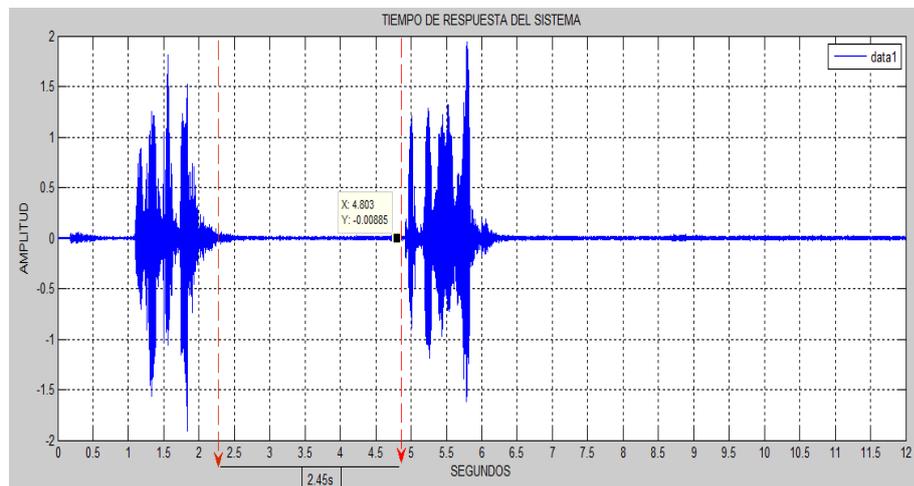


**Figura 5. 4.** Señal de ruido que genera problemas proporcionado por Sound Meter.  
*Fuente: Los autores*

### 5.1.2. Prueba para determinar el tiempo de respuesta del sistema.

Esta prueba se llevó a cabo con la finalidad de realizar una comparación de tiempos de respuesta que tarda en reconocer una orden el sistema “Domotic Voice” con respecto a otros sistemas de reconocimiento de voz, tal como es el caso del software de “Reconocimiento de voz para Windows” y la aplicación “Control por voz gratis” para Android. Con el objetivo de comparar la capacidad de procesamiento que posee cada sistema.

Para obtener el tiempo de respuesta del sistema “Domotic Voice” con respecto a los otros sistemas de reconocimiento de voz, se realizó un programa en Matlab en donde se captura la señal de audio representada en el tiempo. Este programa graba tanto la señal que da la orden al sistema de realizar algún tipo de acción y al mismo tiempo graba una señal audible de respuesta del sistema que indica que la orden ha sido tomada como válida. El objetivo de registrar las dos señales en un solo espectro de voz en el tiempo, es poder determinar la distancia en segundos que tiene una con respecto a la otra. Para una mejor demostración del procedimiento de este método utilizado, se puede observar en la figura 5.5 cómo se obtuvo los tiempos de respuesta a partir de la gráfica de la señal de audio; en este caso se tomó como referencia a la orden de “Abre Cortinas” del sistema “Domotic Voice”.



**Figura 5. 5.** Respuesta en tiempo del espectro de voz para la orden “Abre cortinas” del sistema “Domotic Voice”.

*Fuente: Los autores*

De igual manera se fue realizando el mismo procedimiento para cada una de las órdenes de los sistemas de reconocimiento de voz, cuyos resultados se dan a conocer a continuación en las siguientes tablas:

En la tabla 1 se da conocer los tiempos de respuestas proporcionados por el sistema “Domotic Voice” para cada una de las órdenes que debe reconocer.

**Tabla 1.** *Tiempos de respuestas del sistema “Domotic Voice”.*

<b>“DOMOTIC VOICE”</b>	
<b>Comando de voz</b>	<b>Tiempo en segundos [s]</b>
<b>Casa</b>	1.73s
<b>Cocina</b>	1.67s
<b>Comedor</b>	1.81s
<b>Sala</b>	1.83s
<b>Abre cortinas</b>	2.45s
<b>Cierra cortinas</b>	2.41s
<b>Apaga todo</b>	2.48s

Elaborado por: Christian Alvarez, Miguel Cajas.

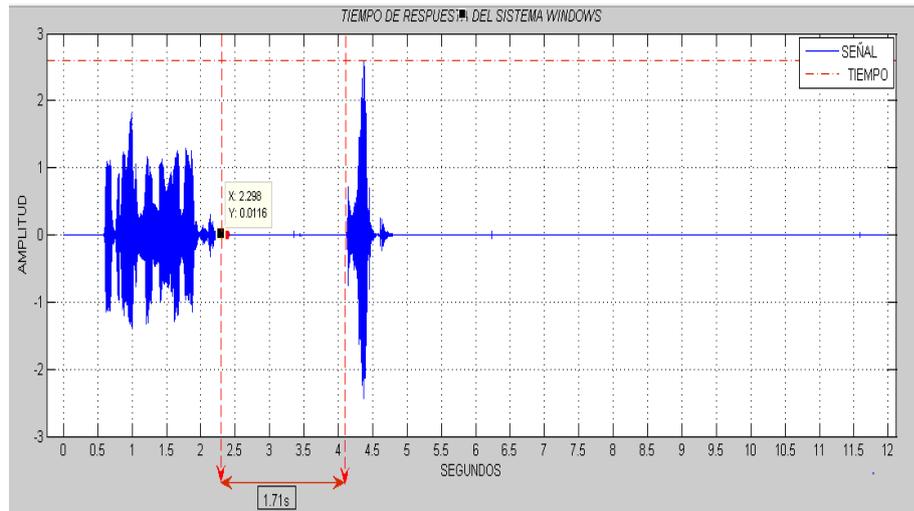
En la tabla 2 se da conocer los tiempos de respuestas proporcionados por el software de “Reconocimiento de voz de Windows” para cada una de las órdenes con las que se realizaron las pruebas.

**Tabla 2.** *Tiempos de respuestas del software “Reconocimiento de voz de Windows”.*

<b>“RECONOCIMIENTO DE VOZ DE WINDOWS”</b>	
<b>Comando de voz</b>	<b>Tiempo en segundos [s]</b>
<b>Menú inicio</b>	1.71s
<b>Abrir Word 2013</b>	1.26s
<b>Abrir Excel 2013</b>	1.45s
<b>Abrir Power Point 2013</b>	1.40s
<b>Abrir Calendario</b>	1.69s
<b>Abrir Tienda</b>	1.52s
<b>Desactivar micrófono</b>	1.02s

Elaborado por: Christian Alvarez, Miguel Cajas.

En la figura 5.6 se da a conocer el tiempo de respuesta de la orden “Menú inicio” a partir de la gráfica de la señal de audio para el software de “Reconocimiento de voz de Windows”.



**Figura 5. 6.** Respuesta en tiempo del espectro de voz para la orden “Menú Inicio” del software de “Reconocimiento de voz de Windows”.

Fuente: Los autores

En la figura 5.7 se puede observar el software de “Reconocimiento de voz de Windows” que se utilizó para realizar las pruebas.



**Figura 5. 7.** Reconocimiento de voz de Windows.

Fuente: Los autores

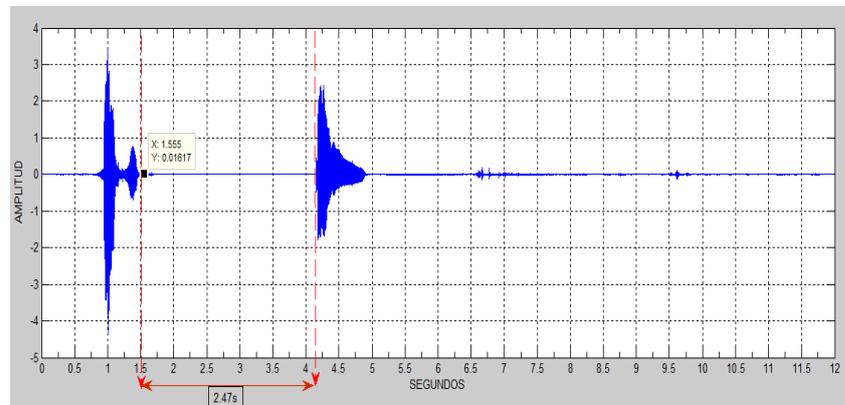
En la tabla 3 se da conocer los tiempos de respuestas proporcionados por la aplicación “Control por voz gratis” de Android para las órdenes con las que se efectuó la prueba.

**Tabla 3.** Tiempos de respuestas de la aplicación “Control por voz gratis”.

<b>“CONTROL POR VOZ GRATIS”</b>	
<b>Comando de voz</b>	<b>Tiempo en segundos [s]</b>
<b>Ayuda</b>	2.85s
<b>Cámara</b>	2.45s
<b>Facebook</b>	2.47s
<b>Descargas</b>	2.62s
<b>Galería</b>	2.83s
<b>Messenger</b>	2.71s
<b>WhatsApp</b>	2.64s

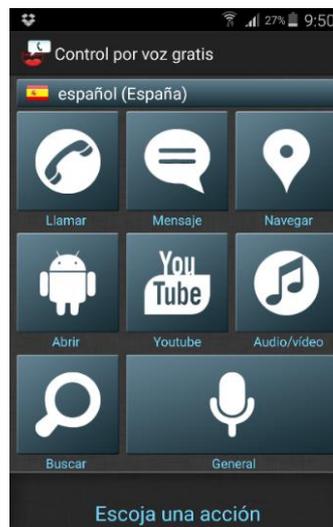
Elaborado por: Christian Alvarez, Miguel Cajas.

En la figura 5.8 se da a conocer el tiempo de respuesta de la orden “Facebook” a partir de la gráfica de la señal de audio para la aplicación “Control por voz gratis”.



**Figura 5. 8.** Respuesta en tiempo del espectro de voz para la orden “Facebook” de la aplicación “Control por voz gratis”.  
Fuente: Los autores

En la figura 5.9 se puede observar la aplicación “Control por voz gratis” de Android que se utilizó para realizar las pruebas.



**Figura 5. 9.** Control por voz gratis.  
Fuente: Los autores

De los sistemas de reconocimiento de voz se obtuvo un promedio total en segundos con respecto a los resultados obtenidos de cada una de las órdenes que se dieron a conocer en las tablas 1, 2 y 3 del tiempo en que tarda cada sistema en procesarlas. Los resultados obtenidos se dan a conocer en la tabla 4 que se muestra a continuación:

**Tabla 4.** Promedio en segundos de los resultados obtenidos de los tres sistemas puestos a prueba.

<b>Resultados obtenidos</b>	
<b>Sistema</b>	<b>Promedio (segundos [s])</b>
“Domotic Voice”	2.05s
“Reconocimiento por voz de Windows”	1.39s
“Control por voz gratis”	2.65s

Elaborado por: Christian Alvarez, Miguel Cajas.

### 5.1.3. Prueba para determinar la capacidad de reconocimiento que posee el sistema para diferentes voces.

Esta prueba se llevó a cabo para determinar la capacidad que posee el sistema “Domotic Voice” para reconocer una orden con respecto a diferentes tonalidades de voz. Para ello se realizó pruebas a cuatro personas de diferente género y edad. La prueba consiste en que cada persona realice un máximo de 10 repeticiones para cada orden que debe reconocer el sistema, sin considerar las veces que se tenga que repetir la palabra “Casa”, ya que esta palabra es la que permite que el sistema pueda entrar en funcionamiento y por ende la persona pueda dar una orden para que sea ejecutada en el sistema domótico. Para ello se tomó en cuenta un promedio para cada persona de las veces que fue necesario repetir la orden de “Casa” para llevar cabo la prueba con el resto de órdenes que reconoce el sistema. Los resultados obtenidos de cada persona se dan a conocer a continuación en las siguientes tablas:

En la tabla 5 se dan a conocer los resultados obtenidos de Christian, el cual requiere de un promedio de 2 repeticiones de la orden “Casa” antes de proceder a dar cualquiera de las ordenes que reconoce el sistema.

**Tabla 5.** Prueba de Christian para cada orden.

<b>“DOMOTIC VOICE”</b>	
<b>Comando de voz</b>	<b>Número de veces que el sistema reconoce la orden</b>
<b>Cocina</b>	<b>8 de 10</b>
<b>Comedor</b>	<b>7 de 10</b>
<b>Sala</b>	<b>9 de 10</b>
<b>Abre cortinas</b>	<b>9 de 10</b>
<b>Cierra cortinas</b>	<b>9 de 10</b>
<b>Dormitorio 1</b>	<b>8 de 10</b>
<b>Apaga todo</b>	<b>8 de 10</b>

Elaborado por: Christian Alvarez, Miguel Cajas.

En la tabla 6 se dan a conocer los resultados obtenidos de Fernando, el cual requiere de un promedio de 4 repeticiones de la orden “Casa” antes de proceder a dar cualquiera de las ordenes que reconoce el sistema.

*Tabla 6. Prueba de Fernando para cada orden.*

<b>“DOMOTIC VOICE”</b>	
<b>Comando de voz</b>	<b>Número de veces que el sistema reconoce la orden</b>
<b>Cocina</b>	<b>3 de 10</b>
<b>Comedor</b>	<b>4 de 10</b>
<b>Sala</b>	<b>4 de 10</b>
<b>Abre cortinas</b>	<b>3 de 10</b>
<b>Cierra cortinas</b>	<b>4 de 10</b>
<b>Dormitorio 1</b>	<b>3 de 10</b>
<b>Apaga todo</b>	<b>5 de 10</b>

Elaborado por: Christian Alvarez, Miguel Cajas.

En la tabla 7 se dan a conocer los resultados obtenidos de Katherine, para lo cual es necesario tomar en cuenta que esta persona requiere de un promedio de 6 repeticiones de la orden “Casa” antes de proceder a dar cualquiera de las ordenes que reconoce el sistema.

*Tabla 7. Prueba de Katherine para cada orden.*

<b>“DOMOTIC VOICE”</b>	
<b>Comando de voz</b>	<b>Número de veces que el sistema reconoce la orden</b>
<b>Cocina</b>	<b>2 de 10</b>
<b>Comedor</b>	<b>2 de 10</b>
<b>Sala</b>	<b>1 de 10</b>
<b>Abre cortinas</b>	<b>0 de 10</b>
<b>Cierra cortinas</b>	<b>1 de 10</b>
<b>Dormitorio 1</b>	<b>0 de 10</b>
<b>Apaga todo</b>	<b>3 de 10</b>

Elaborado por: Christian Alvarez, Miguel Cajas.

En la tabla 8 se dan a conocer los resultados obtenidos de Carlos, el cual requiere de un promedio de 3 repeticiones de la orden “Casa” antes de proceder a dar cualquiera de las ordenes que reconoce el sistema.

*Tabla 8. Prueba de Carlos para cada orden.*

<b>“DOMOTIC VOICE”</b>	
<b>Comando de voz</b>	<b>Número de veces que el sistema reconoce la orden</b>
<b>Cocina</b>	<b>5 de 10</b>
<b>Comedor</b>	<b>6 de 10</b>
<b>Sala</b>	<b>6 de 10</b>
<b>Abre cortinas</b>	<b>6 de 10</b>
<b>Ciera cortinas</b>	<b>6 de 10</b>
<b>Dormitorio 1</b>	<b>6 de 10</b>
<b>Apaga todo</b>	<b>5 de 10</b>

Elaborado por: Christian Alvarez, Miguel Cajas.

En la tabla 9 se da a conocer un promedio total de la capacidad de reconocimiento que posee el sistema “Domotic Voice” para cada una de las personas puestas a prueba con respecto al total de veces que el sistema reconoce cada orden.

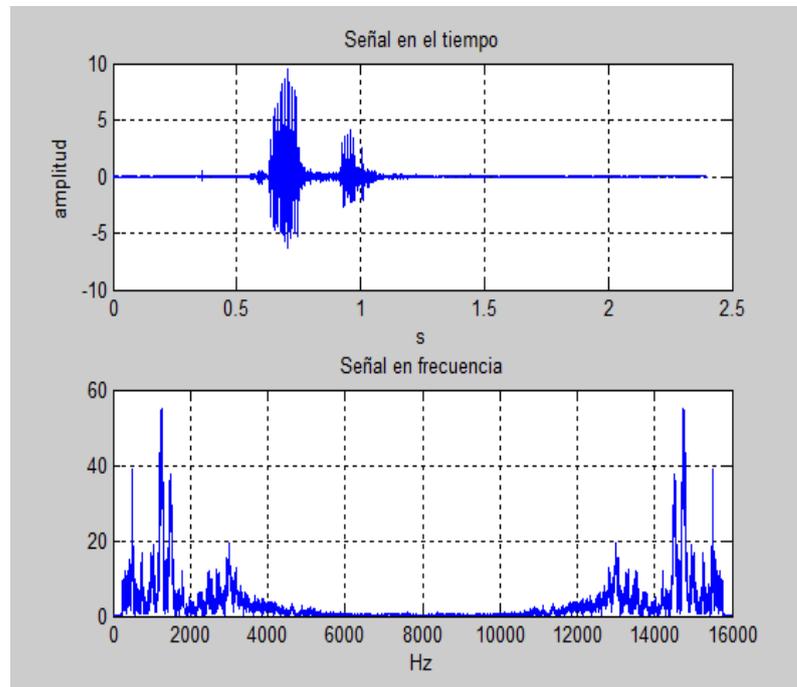
*Tabla 9. Promedios obtenidos para cada orden que el sistema reconoce.*

<b>“DOMOTIC VOICE”</b>					
<b>Comando de voz</b>	<b>Número de veces que el sistema reconoce la orden</b>				<b>Promedio</b>
	<b>Christian</b>	<b>Fernando</b>	<b>Katherine</b>	<b>Carlos</b>	
<b>Cocina</b>	8	3	2	5	<b>4</b>
<b>Comedor</b>	7	4	2	6	<b>5</b>
<b>Sala</b>	9	4	1	6	<b>5</b>
<b>Abre cortinas</b>	9	3	0	6	<b>4</b>
<b>Ciera cortinas</b>	9	4	1	6	<b>5</b>
<b>Dornitorio 1</b>	8	3	0	6	<b>4</b>
<b>Apaga todo</b>	8	5	3	5	<b>5</b>

Elaborado por: Christian Alvarez, Miguel Cajas.

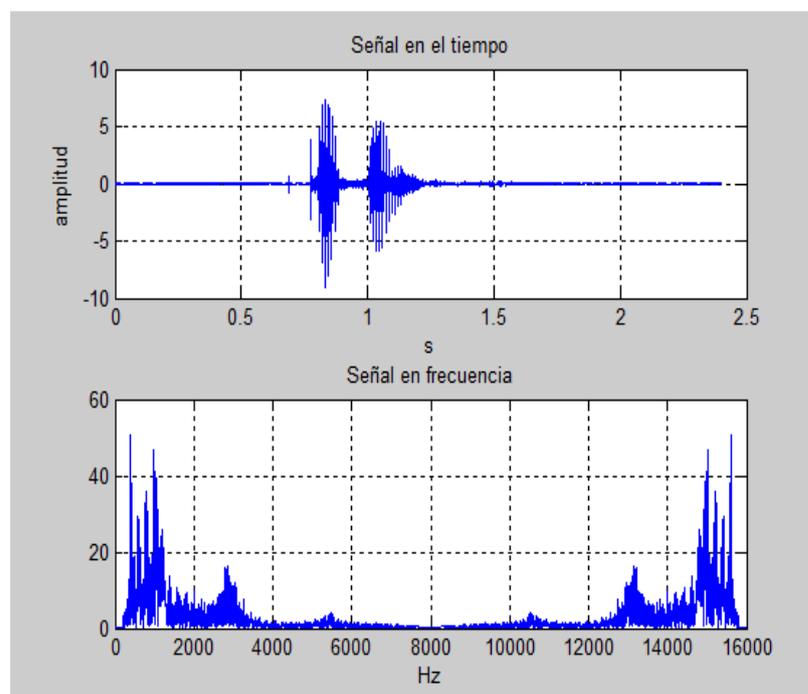
Con la finalidad de poder analizar las distintas tonalidades de voz y determinar el por qué el sistema puede ser menos eficiente con una u otra persona, se realizaron pruebas en Matlab en donde se obtuvo de cada persona la respuesta en tiempo y frecuencia de la orden “Casa”; se tomó esta orden como muestra debido a que es la orden principal que permite al usuario interactuar con el sistema. Los resultados obtenidos se dan a conocer en las siguientes figuras que se muestran a continuación:

En la figura 5.10 se puede observar la respuesta en tiempo y frecuencia de Christian para la orden “Casa”.



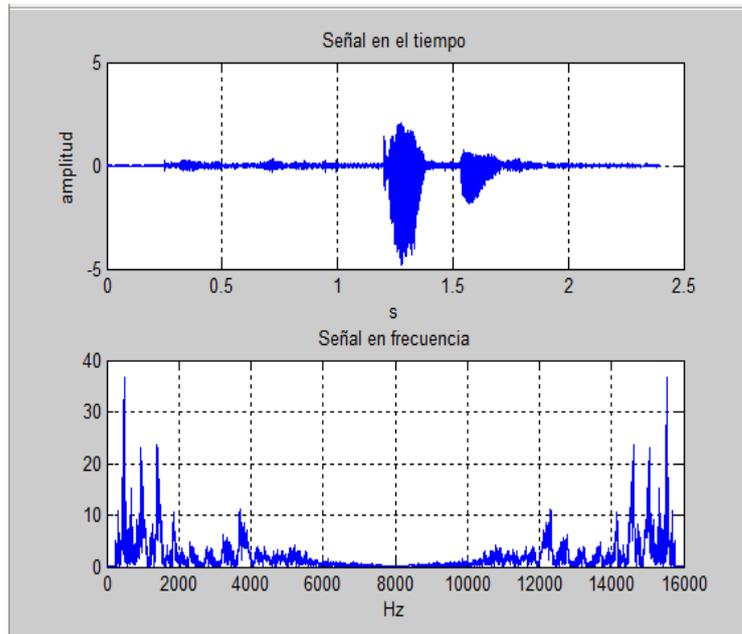
**Figura 5. 10.** Respuesta en tiempo y frecuencia de Christian para la orden “Casa”.  
Fuente: Los autores.

En la figura 5.11 se puede observar la respuesta en tiempo y frecuencia de Fernando para la orden “Casa”.



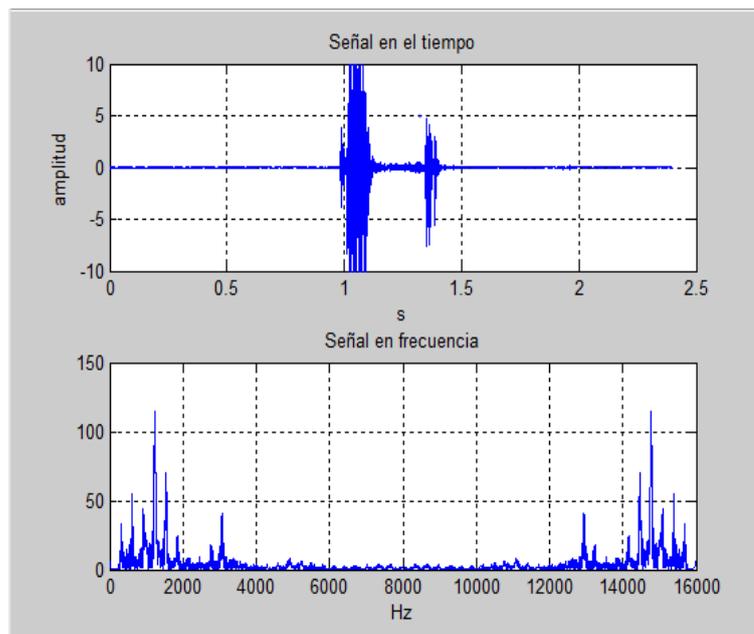
**Figura 5. 11.** Respuesta en tiempo y frecuencia de Fernando para la orden “Casa”.  
Fuente: Los autores.

En la figura 5.12 se puede observar la respuesta en tiempo y frecuencia de Katherine para la orden “Casa”.



*Figura 5. 12. Respuesta en tiempo y frecuencia de Katherine para la orden “Casa”.  
Fuente: Los autores.*

En la figura 55 se puede observar la respuesta en tiempo y frecuencia de Carlos para la orden “Casa”.



*Figura 5. 13. Respuesta en tiempo y frecuencia de Carlos para la orden “Casa”.  
Fuente: Los autores.*

De acuerdo a los resultados obtenidos de las respuestas en tiempo y frecuencia, se puede observar que existe una diferencia notoria entre cada una de las personas puestas a prueba; tal es el caso de Christian quien es la persona que el sistema “Domotic Voice” reconoce con más frecuencia y mayor número de veces. Esto se debe a que fue el encargado de grabar la base de datos que el sistema de reconocimiento de voz utiliza para comparar una orden con respecto a la que otro usuario emita. Es por esta razón que existe una diferencia entre el total de veces que el sistema reconoce a la persona que llevó a cabo la grabación de la base de datos con respecto a las otras personas puesta a prueba, las cuales no tienen una respuesta favorable, debido a que su tonalidad de voz para ciertas palabras difiere de la voz patrón del sistema de reconocimiento de voz; es por tal motivo que en el promedio obtenido en la tabla 9, se tiene como resultado un rango entre el 40% y el 50% en un balance general en el reconocimiento de cada orden.

## 5.2. Análisis económico

Este punto es necesario al momento de llevar a cabo el proyecto, ya que permite conocer la viabilidad económica del sistema a implementar. Para ello es necesario comparar costos del sistema implementado con aquellos que ya han sido desarrollados dentro y fuera del país. A continuación se hará un análisis económico del proyecto.

En la tabla 10 se dan a conocer los costos de la etapa digital del sistema.

*Tabla 10. Costos de dispositivos digitales del sistema.*

<b>DISPOSITIVOS DIGITALES</b>			
<b>Cantidad</b>	<b>Detalle</b>	<b>Precio unitario</b>	<b>Valor total</b>
1	Tarjeta de desarrollo EasyVR 2.0	\$93,00	\$93,00
1	Módulo de Arduino Uno	\$30.00	\$30.00
1	Módulo de Arduino Nano	\$18.00	\$18.00
2	Módulo XBee	\$45.00	\$90.00
		<b>TOTAL</b>	<b>\$231,00</b>

Nota: Los precios de cada uno de los dispositivos se encuentran en dólares americanos.  
Elaborado por: Christian Alvarez, Miguel Cajas.

En la tabla 11 se dan a conocer los precios de los elementos de la etapa de alimentación.

*Tabla 11. Costos de la etapa de alimentación.*

<b>ETAPA DE ALIMENTACIÓN</b>			
<b>Cantidad</b>	<b>Detalle</b>	<b>Precio unitario</b>	<b>Valor total</b>
1	Batería de Lipo	\$23,00	\$23,00
		<b>TOTAL</b>	<b>\$23,00</b>

Nota: El precio del dispositivo se encuentra en dólares americanos.

Elaborado por: Christian Alvarez, Miguel Cajas.

En la tabla 12 se dan a conocer los precios de los elementos utilizados para desarrollar la Plaqueta de Circuito Impreso (PCB) de control del servidor doméstico.

*Tabla 12. Costos del PCB de control del servidor doméstico.*

<b>PCB de control del servidor doméstico</b>			
<b>Cantidad</b>	<b>Detalle</b>	<b>Precio unitario</b>	<b>Valor total</b>
10	Transistores NPN3904	\$0,20	\$2,00
10	Borneras	\$0,25	\$2,50
1	Baquelita de fibra de vidrio	\$2,50	\$2,50
2	Sobres de percloruro férrico	\$1,00	\$2,00
1	Peineta hembra	\$0,80	\$0,80
1	Peineta macho	\$0,80	\$0,80
19	Resistencias	\$0,05	\$0,95
1	Regulador LM317	\$1,25	\$1,25
1	Diodo led	\$0,15	\$0,15
		<b>TOTAL</b>	<b>\$12,95</b>

Nota: Los precios de cada uno de los elementos se encuentran en dólares americanos.

Elaborado por: Christian Alvarez, Miguel Cajas.

En la tabla 13 se dan a conocer los precios de los elementos adicionales requeridos para el desarrollo del sistema de reconocimiento de voz.

*Tabla 13. Costos adicionales para el desarrollo del sistema.*

<b>ELEMENTOS ADICIONALES</b>			
<b>Cantidad</b>	<b>Detalle</b>	<b>Precio unitario</b>	<b>Valor total</b>
1	Carcasa plástica	\$8,00	\$8,00
1	Carcasa de aluminio	\$12,00	\$12,00
1	Auriculares OMega	\$9,00	\$9,00
1	Cobertor de cuero	\$15,00	\$15,00
		<b>TOTAL</b>	<b>\$44,00</b>

Nota: Los precios de cada uno de los elementos se encuentran en dólares americanos.

Elaborado por: Christian Alvarez, Miguel Cajas.

En la tabla 14 se da a conocer el valor total de cada una de las etapas que se han utilizado para el desarrollo del sistema de reconocimiento de voz.

**Tabla 14.** Valor total del sistema de reconocimiento de voz.

Detalle	Valor total
DISPOSITIVOS DIGITALES	\$231,00
ETAPA DE ALIMENTACIÓN	\$23,00
PCB de control del servidor domótico	\$12,95
ELEMENTOS ADICIONALES	\$44,00
<b>TOTAL</b>	<b>\$310,95</b>

Nota: Los precios de cada uno de los componentes del sistema se encuentran en dólares americanos.

Elaborado por: Christian Alvarez, Miguel Cajas.

Adicional a esto se debe considerar un costo por el programa realizado para el sistema de reconocimiento de voz, para ello se ha establecido el número total de horas (80 horas) que requirió realizar el programa con respecto al valor por hora, en este caso el valor por hora es de \$3,00. Por lo tanto el valor por programación es de **\$240,00**.

Para determinar la rentabilidad del sistema desarrollado, es necesario calcular los indicadores de rentabilidad, tales como la Tasa Mínima Aceptable de Retorno (TMAR), la Tasa Interna de Retorno (TIR) y el Valor Actual Neto (VAN), para lo cual se ha realizado una proyección de ventas para un periodo de 3 años. Se ha establecido una producción inicial de 6 sistemas de reconocimiento de voz, los cuales serán construidos en un periodo de un mes para luego poder venderlos durante el resto del año, debido a que el producto va a ser nuevo en el mercado y su aceptación dependerá de los usuarios que lo lleguen a adquirir. Posterior a ello se pretende construir 12 sistemas en el siguiente año, de igual manera se los construirá en un periodo de dos meses para el resto del año poder venderlos, y finalmente llegar a producir 18 sistemas en un periodo de 3 meses para el tercer año. En la tabla 15 se dan a conocer los requisitos iniciales que se requieren para llevar a cabo la producción del sistema de reconocimiento de voz desarrollado.

**Tabla 15.** Requisitos iniciales para la producción del sistema de reconocimiento de voz.

<b>REQUISITOS INICIALES</b>	
<b>ARRIENDO</b>	
<b>Detalle</b>	<b>Valor</b>
Costo por arriendo mensual	\$120,00
<b>Total</b>	<b>\$120,00</b>
<b>MATERIA PRIMA</b>	
<b>Detalle</b>	<b>Valor</b>
Dispositivos digitales	\$231,00
Etapas de alimentación	\$23,00
PCB de control del servidor domótico	\$12,95
Elementos adicionales	\$44,00
<b>Total</b>	<b>\$310,95</b>
<b>EQUIPOS Y HERRAMIENTAS</b>	
<b>Detalle</b>	<b>Valor</b>
Un multímetro	\$60,00
Un caudín	\$32,00
Dos chupa solda	\$20,00
Kit de herramientas	\$36,00
Dos manillas antiestáticas	\$24,00
<b>Total</b>	<b>\$172,00</b>
<b>PUBLICIDAD</b>	
Tarjetas de Publicidad	\$50,00
<b>Total</b>	<b>\$50,00</b>
<b>SERVICIOS BASICOS</b>	
<b>Detalle</b>	<b>Valor</b>
Todos los servicios básicos por mes	\$35,61
<b>Total</b>	<b>\$35,61</b>
<b>MANO DE OBRA</b>	
Salario por socio mensual	<b>\$366,00</b>
<b>Total</b>	<b>\$732,00</b>

Nota: Los precios de cada uno de los requisitos se encuentran en dólares americanos. Elaborado por: Christian Alvarez, Miguel Cajas.

Es necesario considerar que el arriendo, los servicios básicos y la mano de obra son por un solo mes, debido a que en este tiempo se construirá los 6 sistemas de reconocimiento de voz iniciales. Para el siguiente año se considera dos meses, debido a que se van a construir el doble de productos con respecto a la producción inicial. Finalmente se considera 3 meses para el último año.

Para poder obtener un costo aproximado para el producto, es necesario tomar en cuenta los requisitos iniciales que se requieren para producirlo. Con la finalidad de obtener

un valor que permita cubrir tanto lo que se requiere para producirlo, así como también obtener una ganancia. A continuación se da a conocer la manera de cómo obtener el valor por cada producto:

**Requisitos:**

- Arriendo = \$120,00
- Servicios Básicos = \$35,61
- Materia prima = \$310,95 por cada producto
- Mano de Obra = \$732,00
- Programación = \$240,00
- Herramientas = \$172,00

Para el valor de renta, servicios básicos, mano de obra, programación y herramientas es necesario dividirlo para seis, debido a que estos costos son para producir los seis sistemas tomados en cuenta para el primer año y no por un solo producto. Por tal motivo, los valores para producir un solo producto son los siguientes:

- Arriendo =  $\$120,00 / 6 = \$20,00$
- Servicios Básicos =  $\$35,61 / 6 = \$5,93$
- Materia prima = \$310,95
- Mano de Obra =  $\$732,00 / 6 = \$122,00$
- Programación =  $\$240,00 / 6 = \$40,00$
- Herramientas =  $\$172,00 / 6 = \$28,67$

**Valor por cada producto = \$527,55**

Al valor obtenido para cada producto es necesario sumarle la utilidad que se desea obtener por el mismo, para nuestro caso se plantea una utilidad del 10%. Además que se debe sumar el impuesto que se cobra actualmente por la venta de un producto, en este caso el valor por IVA es del 14%. De acuerdo a estas consideraciones se obtiene el costo final que se deberá cobrar por cada producto.

$$\text{Valor de venta por producto} = \text{Valor por cada producto} + \text{Utilidad por producto} + \text{IVA}$$

$$\text{Valor de venta por producto} = \$527,55 + \$52,75 + \$73,86 = \$654,16$$

Para determinar la ganancia que se tendrá por producto es necesario aplicar la ecuación 10 que se muestra a continuación:

*Ecuación 10. Ecuación para determinar la utilidad bruta (UB) [24].*

$$\text{Utilidad Bruta} = \text{Precio de venta} - \text{Costo del producto} \quad (10)$$

$$UB = \$654,16 - \$527,55 = \$126,61$$

Tomando en cuenta los requisitos iniciales y el costo total por cada producto, es necesario calcular los egresos que se tendrá en los 3 años. A continuación se dan a conocer los egresos de cada año:

### **EGRESOS AÑO 1**

En primer lugar se debe considerar los impuestos que se deben pagar. Para obtener este valor se debe realizar una multiplicación del costo de venta del producto por el total de productos producidos y por el IVA.

$$\text{Impuestos por productos} = (\text{Precio de venta}) \times (\text{Número de productos}) \times (\text{IVA})$$

$$\text{Impuestos por productos} = (\$654,16) \times (6) \times (0.14) = \$549,49$$

Una vez obtenido el valor de impuestos a pagar, es necesario calcular el total de egresos que se tendrán para ese año. A continuación se dan conocer el egreso total que se tendrá para el año 1.

- **Arriendo** = \$120,00
- **Servicios Básicos** = \$35,61
- **Equipos y herramientas** = \$172,00
- **Mano de obra** = \$732,00
- **Publicidad** = \$50,00

- **Materia prima para los 6 productos=** \$1865,70
- **Impuestos =** \$ 549,49

**Total de egresos para el año 1 = \$3524,80**

### **EGRESOS AÑO 2**

Los impuestos que se deben pagar en el segundo año se calculan a continuación:

**Impuestos por productos = (Precio de venta) x (Número de productos) x (IVA)**

$$\text{Impuestos por productos} = (\$654,16) \times (12) \times (0.14) = \$1098,99$$

Una vez obtenido el valor de impuestos a pagar, es necesario calcular el total de egresos que se tendrán para ese año. A continuación se dan conocer el egreso total que se tendrá para el año 2.

- **Arriendo =** \$240,00
- **Servicios Básicos =** \$71,22
- **Mano de obra =** \$1464,00
- **Publicidad =** \$50,00
- **Materia prima =** \$3731,40
- **Impuestos =** \$ 1098,99

**Total de egresos para el año 2 = \$6655,61**

A este valor se le debe considerar la tasa de inflación del 1.78% que se tiene hasta la presente fecha del 2016 en Ecuador, dando un total de **\$6774,08**. Donde su valor verdadero dependerá de la inflación que se tenga para ese año en el Ecuador.

### **EGRESOS AÑO 3**

Los impuestos que se deben pagar en el segundo año se calculan a continuación:

**Impuestos por productos = (Precio de venta) x (Número de productos) x (IVA)**

$$\text{Impuestos por productos} = (\$654,16) \times (18) \times (0.14) = \$1648,48$$

Una vez obtenido el valor de impuestos a pagar, es necesario calcular el total de egresos que se tendrán para ese año. A continuación se dan conocer el egreso total que se tendrá para el año 3.

- **Arriendo** = \$360,00
- **Servicios Básicos** = \$106,83
- **Mano de obra** = \$2196,00
- **Publicidad** = \$50,00
- **Materia prima** = \$5597,10
- **Impuestos** = \$ 1648,48

**Total de egresos para el año 3 = \$9958,41**

A este valor se le debe considerar la tasa de inflación del 1.78% que se tiene hasta la presente fecha del 2016 en Ecuador, dando un total de **\$10135,67**. Donde su valor verdadero dependerá de la inflación que se tenga para ese año en el Ecuador.

Obtenido los valores de egresos para cada año, es necesario de igual manera determinar los ingresos. Para ello se ha considerado el valor de ventas, el costo por instalación y el costo de programación por cada producto. A continuación se dan a conocer los ingresos que se tendrá para cada año:

### **INGRESOS AÑO 1**

Dado que se van a producir un total de 6 productos al año, es necesario multiplicar este valor para cada uno de los ingresos considerados.

**Venta de productos** =  $(\$654,16) \times (6) = \$3924,96$

**Costo por instalación por producto** =  $(\$50) \times (6) = \$300,00$

**Total de ingresos para el año 1 = \$4224,96**

### **INGRESOS AÑO 2**

Dado que se van a producir un total de 12 productos al año, es necesario multiplicar este valor para cada uno de los ingresos considerados.

**Venta de productos** =  $(\$654,16) \times (12) = \$7849,92$

**Costo por instalación por producto** =  $(\$50) \times (12) = \$600,00$

**Total de ingresos para el año 2 = \$8449,92**

### **INGRESOS AÑO 3**

Dado que se van a producir un total de 18 productos al año, es necesario multiplicar este valor para cada uno de los ingresos considerados.

**Venta de productos** =  $(\$654,16) \times (18) = \$11674,88$

**Costo por instalación por producto** =  $(\$50) \times (18) = \$900,00$

**Total de ingresos para el año 3 = \$12674,88**

Tomando en cuenta los egresos e ingresos que se tendrán para los tres años que se consideraron para determinar la rentabilidad del producto, se determinó el flujo de caja que se muestra en la tabla 16.

*Tabla 16. Flujo de caja.*

<b>FLUJO DE CAJA</b>			
<b>DESCRIPCION</b>	<b>AÑO 1</b>	<b>AÑO 2</b>	<b>AÑO 3</b>
INGRESOS	\$4224,96	\$8449,92	\$12674,88
EGRESOS	\$3524,80	\$6774,08	\$10135,67
<b>Total de efectivo Neto</b>	<b>\$700,16</b>	<b>\$1675,84</b>	<b>\$2539,21</b>

Nota: Los precios de los egresos e ingresos se encuentran en dólares americanos.  
Elaborado por: Christian Alvarez, Miguel Cajas.

A continuación en la figura 5.14 se da conocer los indicadores de rentabilidad obtenidos en Excel que demuestran la rentabilidad del sistema de reconocimiento de voz “Domotic Voice”. Esto se puede observar, ya que el valor de la Tasa Interna de Retorno (TIR) es mayor a la Tasa de Rentabilidad Mínima Aceptable (TMAR) que se debe considerar siempre para que un negocio pueda ser considerado rentable dentro del mercado y que la producción del mismo de ganancias a las personas involucradas en su desarrollo.

<b>Tasa de inflación</b>		1,78%	
<b>Premio al riesgo</b>		10,00%	
<b>Inversion Inicial</b>		\$ (3.075,26)	
	<b>Ingresos</b>	<b>Egresos</b>	
<b>Año</b>	<b>Valor</b>		<b>Valor</b>
1	\$ 4.224,96		\$ 3.524,80
2	\$ 8.449,92		\$ 6.774,08
3	\$ 12.674,88		\$ 10.135,67
<b>Tasa de Interes</b>		15,00%	
<b>Número de años=</b>		3 años	
<b>VAN=</b>		<b>\$ 3.234,63</b>	
<b>TIR=</b>		<b>22%</b>	
<b>TMAR=</b>		<b>12%</b>	

*Figura 5. 14. Indicadores de rentabilidad del sistema "Domotic Voice".*

*Fuente: Los autores.*

Un análisis adicional que se tomó en cuenta para la parte económica, es una comparación de presupuestos que se requieren para desarrollar un sistema de reconocimiento de voz. Los sistemas considerados corresponden a otras instituciones tanto nacionales como extranjeras que han desarrollado un sistema de reconocimiento de voz. Para ello en la tabla 17 se dan a conocer los presupuestos de materia prima de los sistemas tomados en cuenta para el análisis.

*Tabla 17. Presupuestos de sistemas desarrollados por otras instituciones para el reconocimiento de voz.*

<b>NOMBRE</b>	<b>Detalles</b>	<b>Presupuesto</b>
Sistema de control Automático de una ducha eléctrica mediante reconocimiento de la voz para personas con discapacidad en las extremidades superiores	Desarrollado por Alex Darío Acosta López de la Universidad Técnica de Ambato [25].	\$672,00
Diseño e implementación de un control remoto controlado por órdenes de voz para aplicaciones de control de una vivienda	Desarrollado por Luis Daniel Robalino Puente de la Universidad Politécnica Nacional [26].	\$227,62
Control domótico con dispositivos móviles	Desarrollado por José Alfonso Pérez Giménez de la Universidad Oberta de Catalunya [27].	\$648,00
Desarrollo de una interfaz multimodal para un robot domótico	Desarrollado por Jorge Cancela González de la	\$896,00

móvil. Control por voz y mando a distancia	Universidad Politécnica de Madrid [11].	
Sistema de control remoto para aplicaciones domóticas a través de internet	Desarrollado por Mario Rodríguez Cerezo de la Universidad Autónoma de Madrid [28].	\$1716,00
Sistema de seguridad por reconocimiento de voz	Desarrollado por Pérez Badillo Eyra Oxana, Poceros Martínez Fernando y Villalobos Ponce José Alexis en el Instituto Politécnico Nacional de México [6].	\$1.781,43

Elaborado por: Christian Alvarez, Miguel Cajas.

De acuerdo a los presupuestos de los sistemas desarrollados dentro y fuera del país que se pueden observar en la tabla 17, el sistema “Domotic Voice” tiene un costo menor con respecto a ciertos sistemas y un costo mayor a uno de ellos. Por lo cual se debe considerar el método y el alcance que poseen estos sistemas en comparación al sistema desarrollado.

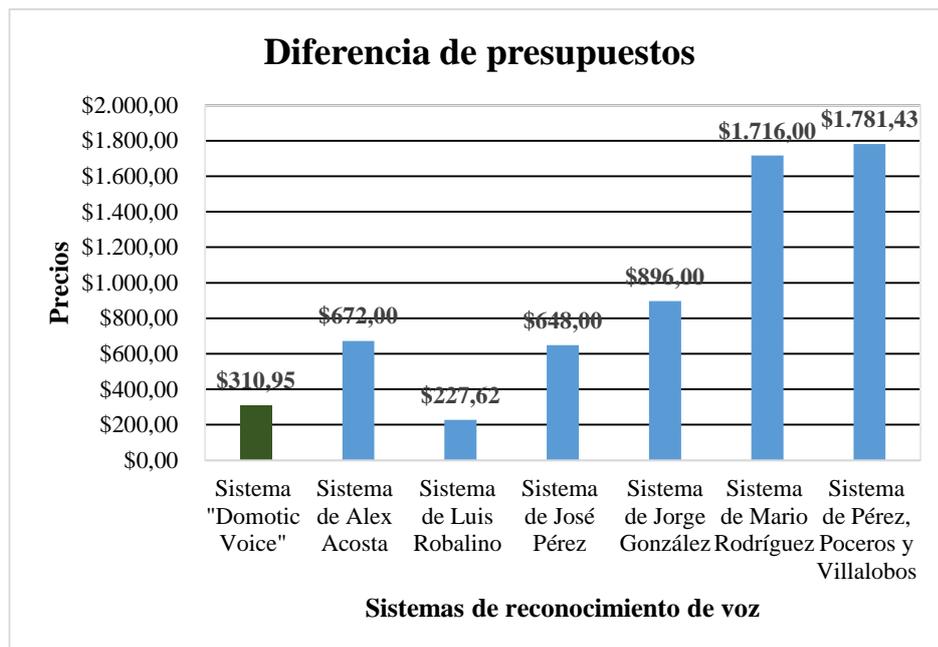
Por ejemplo el sistema desarrollado por Luis Robalino utiliza una tarjeta de desarrollo de reconocimiento de voz y como método de comunicación el protocolo ZigBee, de manera similar a lo que utiliza el sistema “Domotic Voice”. Sin embargo el sistema “Domotic Voice” posee la ventaja de poder ser utilizado en el sistema domótico LOXONE. Además que dispone de un segundo sistema que funciona como un gateway que permite comunicar el sistema con un servidor domótico en el cual se pueda manipular las entradas del servidor o sistemas en los cuales se puedan controlar los elementos del hogar por medio de relés, lo cual hace que su costo sea mayor al del sistema comparado [26].

Por otro lado, el sistema desarrollado por Alex Acosta tiene un costo mayor al sistema “Domotic Voice”, a pesar que los dos sistemas utilizan de igual manera como elemento de control un módulo de Arduino y una tarjeta EasyVR para la parte de reconocimiento de voz. Se debe considerar que el sistema “Domotic Voice” permite controlar más elementos del hogar con respecto al sistema comparado, el cual únicamente permite realizar el control de una ducha eléctrica [25].

Con respecto a los otros sistemas que se dan a conocer en la tabla 17, es necesario tomar en cuenta que estos poseen aplicaciones o elementos adicionales que difieren de los que se utilizan en el sistema “Domotic Voice”, pero sin embargo están dentro del

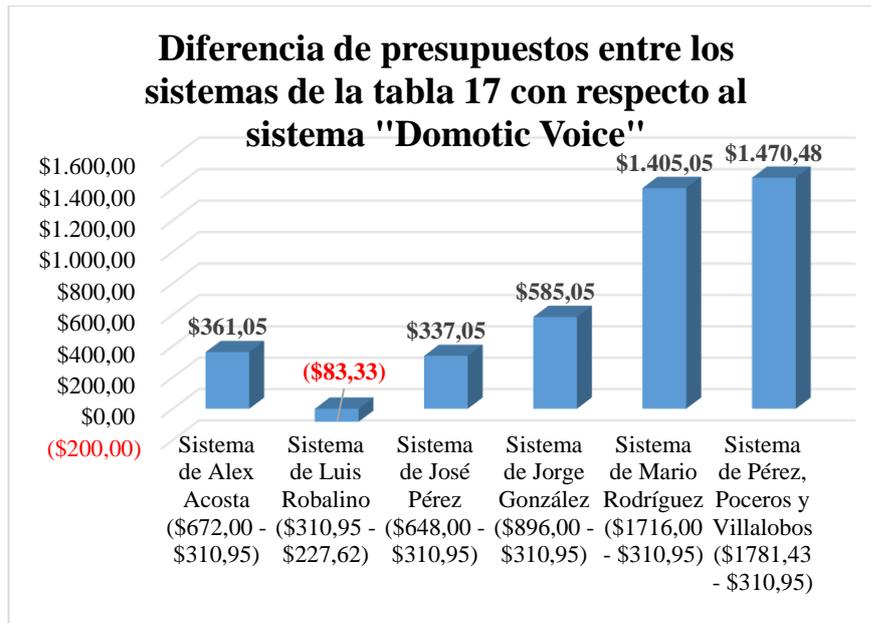
área de reconocimiento de voz para aplicaciones de control domótico; es por tal motivo que fueron consideradas dentro del análisis para comparar los costos que tienen los sistemas de reconocimiento de voz para distintas aplicaciones de control de viviendas, edificios e incluso para el control de un robot autónomo.

En la figura 5.15 se puede observar el presupuesto de cada sistema de reconocimiento de voz de la tabla 17 con respecto al sistema “Domotic Voice”.



**Figura 5. 15.** Presupuestos de los sistemas de reconocimiento de voz de la tabla 17 con respecto al sistema “Domotic Voice”.  
Fuente: Los autores.

En la figura 5.16 se puede observar la diferencia de presupuestos de los sistemas de reconocimiento de voz de la tabla 17 con respecto al sistema “Domotic Voice”.



**Figura 5. 16.** Diferencia de presupuestos de los sistemas de reconocimiento de voz de la tabla 17 con respecto al sistema "Domotic Voice".

Fuente: Los autores.

---

## CAPÍTULO 6

---

### 6. CONCLUSIONES Y RECOMENDACIONES

En este capítulo se dan a conocer las conclusiones y recomendaciones de acuerdo a los resultados obtenidos de cada una de las pruebas realizadas para determinar la funcionalidad del sistema desarrollado. A continuación se dan a conocer las conclusiones y recomendaciones:

#### 6.1. Conclusiones

De acuerdo a los resultados obtenidos en la prueba para determinar el nivel de ruido que tolera el sistema de reconocimiento de voz desarrollado, se obtuvo un rango de 28.4 dB entre el valor mínimo de 41dB y el valor máximo de 69.4 dB medidos en condiciones normales, es decir sin personas que se encuentren dentro del laboratorio y que puedan generar ruido al momento de utilizar el sistema de reconocimiento de voz, pero sin desestimar el ruido del ambiente; con lo cual se tiene un rango considerable para que el sistema “Domotic Voice” puede funcionar de manera correcta, sin presentar errores en cuanto al reconocimiento de las órdenes. Por otra parte, se tiene un rango entre el valor mínimo de 74.6 dB y el valor máximo 86.3 dB medidos para determinar la tolerancia de ruido que posee el sistema desarrollado; lo cual permite constatar que el sistema posee un limitante con respecto a los niveles altos de ruido, ya que al superar los 86 dB el sistema tiende a presentar problemas en el procesamiento de las órdenes. En este punto se debe considerar que el ruido que más afecta al sistema es el provocado por las voces de otras personas que se encuentran cerca del micrófono del dispositivo. Esto se debe a que el sistema interpreta estas voces como órdenes mal pronunciadas e intenta buscar coincidencias en su base de datos, generando un error que provoca que el sistema entre en una etapa de suspensión, es decir que no permite interpretar una orden hasta que el sistema sea reiniciado.

Con respecto al tiempo de respuesta que posee el sistema “Domotic Voice” para reconocer cada orden, se obtuvo una respuesta favorable con respecto a los sistemas que se tomaron como referencia para compararlo. Tal es el caso del software de reconocimiento de voz de Windows, el cual posee una diferencia de 0.62 milisegundos con respecto al sistema “Domotic Voice”, lo cual hace que este sistema desarrollado por Windows ejecute las órdenes en un tiempo menor a nuestro sistema desarrollado;

se debe tomar en cuenta que la diferencia de tiempos no es muy considerable a la vista de un usuario, pero sin embargo resulta ser superior en cuanto al procesamiento que posee el sistema de Windows. Por otro lado, el sistema “Domotic Voice” posee una diferencia de 0.6 milisegundos con respecto a la aplicación “Control por voz gratis” de Android, lo cual hace que nuestro sistema ejecute las órdenes en un tiempo menor a esta aplicación desarrollada para dispositivos que disponen de tecnología Android. De igual manera este tiempo no es muy notorio a la vista del usuario, pero en cuanto al procesamiento de las órdenes se tiene una ventaja con respecto al sistema comparado; las ventajas y desventajas mostradas en este análisis están ligadas directamente al procesador que posee cada dispositivo y a la metodología usada para el reconocimiento del habla de cada uno de ellos.

Por lo tanto cada una de las órdenes que el sistema “Domotic Voice” debe reconocer está dentro de un rango de procesamiento considerable con respecto a los sistemas comparados. Lo cual permite tener una buena respuesta en cuanto al tiempo que demora el sistema desarrollado en reconocer una orden para que sea ejecutada en el sistema domótico. Cabe señalar que el tiempo de procesamiento será cada vez más alto a medida que se aumente el número de órdenes en el sistema, esto como consecuencia del método de reconocimiento utilizado, ya que tiene que comparar los valores de distancia con cada palabra que ha sido almacenada como una orden.

Por otro lado el sistema fue puesto a prueba con 4 personas con características de voz distintas con la finalidad de determinar la capacidad de reconocimiento que posee el sistema para diferentes tonalidades de voz. La base de datos para el sistema se tomó de una persona de 28 años quien grabó las órdenes que deben ser reconocidas, los sujetos de prueba fueron una mujer de 23 años, una persona adulta de 40 años, y otro sujeto de 16 años. Es así que para cada una de ellas se realizó un máximo de 10 repeticiones para cada orden que debe reconocer el sistema, sin considerar las veces que se tenga que repetir la palabra “Casa”, ya que esta palabra es la que permite que el sistema pueda entrar en funcionamiento y por ende la persona pueda dar una orden para que sea ejecutada en el sistema domótico.

De acuerdo a los resultados obtenidos de las cuatro personas puestas a prueba se obtuvo una diferencia con respecto al número de veces que el sistema reconoce cada orden que emite cada una de estas personas. Esto se debe a que cada una de ellas tiene un tono de voz diferente y sobre todo tiene diferente pronunciación, lo cual provoca

cierta dificultad al sistema para el reconocimiento de las órdenes debido a que no coinciden con los patrones de voz contenidos en la base de datos. Esto se pudo corroborar con los resultados obtenidos de las respuestas en tiempo y frecuencia de la palabra “Casa” que se tomó de cada una de las personas, las cuales difieren en las magnitudes y en los tiempos lo que altera el cálculo de las distancias euclidianas que posee la voz patrón; tal es el caso del espectro de voz femenina, la cual posee una diferencia notoria al resto de espectros obtenidos de las otras tres personas, lo cual se debe a un tono de voz más agudo que provoca que el cálculo de la distancia euclidiana este alejada a los coeficientes de la voz patrón, lo cual provoca que el sistema sea incapaz de reconocer órdenes que contengan especialmente vocales cerradas, ya que estas son más distorsionadas en una voz aguda.

Del total de veces que el sistema reconoce una orden para cada una de las cuatro personas, se obtuvo un promedio que se muestra en la tabla 9, la cual da un resultado entre el 40% y el 50% de reconocimiento para cada orden. En las pruebas de repetición también se pudo observar que las órdenes de mayor longitud, así como las compuestas por dos palabras tendían a ser reconocidas más rápido que las demás órdenes; esto se debe a que este tipo de órdenes tienen más puntos de referencia para ser comparadas con respecto a las demás. Sin embargo esto también causa un tiempo mayor en la respuesta del sistema.

## **6.2. Recomendaciones**

El funcionamiento correcto del sistema de reconocimiento de voz siempre estará sujeto a algunos parámetros de uso. Debido a ciertas limitantes provocados por agentes externos, ya que este tipo de sistemas son propensos a variables como el ruido ambiental, el cual podría cambiar la diafonía de las palabras impidiendo que el sistema correlacione de manera correcta con su base de datos. Por lo tanto es necesario que cada una de las órdenes que emita el usuario sea claras, es decir se mantenga una tonalidad normal de la voz para garantizar que el sistema funcione correctamente; no es necesario emitir una voz muy fuerte o demasiado débil, ya que el sistema no reconocerá las órdenes. Además que es necesario considerar la distancia con la cual debe colocarse el usuario el micrófono, para evitar que a distancias muy cortas la respiración de la persona puede generar un ruido que distorsione la orden que emita o la distorsione por la saturación de las señales sonoras. En el caso de que el sistema vaya a ser utilizado por una persona a la cual no se le adapte de manera satisfactoria el

sistema debido a que posee una tonalidad de voz no muy común, es necesario entrenar el sistema valiéndose de una nueva base de datos con la voz de esta persona, para así garantizar que el sistema funcione correctamente. Por lo tanto podemos notar que una de las deficiencias que posee el sistema es poder englobar una alta variedad de tonalidades de voces diferente, por lo cual una mejora que se puede realizar a nuestro sistema de reconocimiento de voz para que se adapte a una gama mucho mayor de voces, es desarrollar el sistema basándonos en un método mucho más preciso como lo es una red neuronal que permita realizar un entrenamiento que generalice las voces de todas las personas, permitiendo que el sistema sea utilizado sin importar el tono de voz. Para lo cual sería necesario conseguir una tarjeta de desarrollo que posea un procesador que pueda almacenar esta red neuronal, de tal manera que se pueda obtener mejores resultados en cuanto a la generalización de la voz a una mayor velocidad. Sin embargo se debe tomar en cuenta que en el mercado nacional no existe una tarjeta de desarrollo de reconocimiento de voz que posea una buena capacidad de memoria de procesamiento, lo cual sería una desventaja para poder mejorar el sistema, ya que esto implicaría el tener que conseguir una tarjeta de desarrollo en otro país, siempre y cuando se consiga una tarjeta que cumpla con buenas características para poder montar la red neuronal. Afectando el costo final del sistema de reconocimiento de voz, ya que aumentaría el costo de la tarjeta, así como el costo adicional por el envío del producto

## ABREVIATURAS

CONADIS	Concejo Nacional de Igualdad de Discapacidades
IEEE	Institute of Electrical and Electronics Engineers
MHz	Mega Hertz
Kbps	kilobytes por segundo
$s(n)$	Estimación
$p$	Muestras anteriores
$\epsilon(n)$	Error
LPC	Componentes de Predicción Lineal
MFCC	Coefficientes Cepstrales de Frecuencia Mel
$d$	Distancia
3D	Tres Dimensiones
ABB	Automation and Power Technologies
PC	Procesador Central
CPU	Unidad Central de Proceso
CMOS	Semiconductor Complementario de Óxido Metálico
RAM	Random Access Memory
UART	Transmisor Receptor Asíncrono Universal
GPIO	General Purpose Input/Output
PWM	Modulación por Ancho de Pulso
SRAM	Static Random Access Memory
EEPROM	Electrically Erasable Programmable Read-Only Memory
mA	Miliamperios
PCB	Plaqueta de Circuito Impreso
VCC	Voltaje de Corriente Continua
A	Amperios
V	Voltios
WAV	Waveform Audio Format
$K\Omega$	Kiloohmios

## ANEXOS

### Anexo 1. Código en Arduino del sistema de reconocimiento de voz.

```
#if defined(ARDUINO) && ARDUINO >= 100
#include "Arduino.h"
#include "SoftwareSerial.h"
SoftwareSerial port(12,13);
#else // Arduino 0022 - use modified NewSoftSerial
#include "WProgram.h"
#include "NewSoftSerial.h"
NewSoftSerial port(12,13);
#endif
#include "EasyVR.h"
EasyVR easyvr(port);
//Groups and Commands
enum Groups
{
    GROUP_0 = 0,
    GROUP_1 = 1,
};
enum Group0
{
    G0_CASA = 0,
};
enum Group1
{
    G1_SALA = 0,
    G1_COSINA = 1,
    G1_COMEDOR = 2,
    G1_DORMITORIO1 = 3,
    G1_DORMITORIO2 = 4,
    G1_ABRECOTINA = 5,
    G1_CIERRACORTIN = 6,
    G1_PRENDE_CALEF = 7,
```

```

    G1_APAGA_CALEF = 8,
    G1_LUCES=9,
    G1_APAGA_TODO = 10,
};
EasyVRBridge bridge;
int8_t group, idx;
void setup()
{
    // bridge mode?
    if (bridge.check())
    {
        cli();
        bridge.loop(0, 1, 12, 13);
    }
    // run normally
    Serial.begin(9600);
    port.begin(9600);
    if (!easyvr.detect())
    {
        Serial.println("EasyVR not detected!");
        for (;;)
        {
            easyvr.setPinOutput(EasyVR::IO1, LOW);
            Serial.println("EasyVR");
            easyvr.setTimeout(5);
            easyvr.setLanguage(0);
            easyvr.playSound(14, EasyVR::VOL_FULL);
            group = EasyVR::TRIGGER; //<-- start group (customize)
            for(int i=2; i<=6; i++){// ***** NUEVO
            pinMode(i, OUTPUT); // ***** NUEVO
            } // ***** NUEVO
        }
    }
    void action();
    void loop()

```

```

{
easyvr.setPinOutput(EasyVR::IO1, HIGH); // LED on (listening)
//Serial.print("Say a command in Group ");
//Serial.println(group);
easyvr.recognizeCommand(group);
do
{
// can do some processing while waiting for a spoken command
}
while (!easyvr.hasFinished());
easyvr.setPinOutput(EasyVR::IO1, LOW); // LED off
idx = easyvr.getWord();
if (idx >= 0)
{
// built-in trigger (ROBOT)
// group = GROUP_X; <-- jump to another group X
return;
}
idx = easyvr.getCommand();
if (idx >= 0)
{
// print debug message
uint8_t train = 0;
char name[32];
//Serial.print("Command: ");
//Serial.print(idx);
if (easyvr.dumpCommand(group, idx, name, train))
{
//Serial.print(" = ");
//Serial.println(name);
}
}
else
Serial.println();
easyvr.playSound(0, EasyVR::VOL_FULL);

```

```

// perform some action
action();
}
else // errors or timeout
{
if (easyvr.isTimeout())
Serial.println("T");//Timed out, try again...
int16_t err = easyvr.getError();
if (err >= 0)
{
//Serial.print("Error ");
//Serial.println(err, HEX);
easyvr.playSound(10, EasyVR::VOL_FULL);
}
}
}
void action()
{
switch (group)
{
case GROUP_0:
switch (idx)
{
case G0_CASA:
easyvr.playSound(1, EasyVR::VOL_FULL);
// write your action code here
group = GROUP_1;// <-- or jump to another group X for composite commands
break;
}
break;
case GROUP_1:
switch (idx)
{
case G1_SALA:

```

```

// Serial .print("sala encendida");
if(digitalRead(2) == LOW){ // ***** NUEVO
digitalWrite(2, HIGH); // ***** NUEVO
Serial.write('G');
easyvr.playSound(9, EasyVR::VOL_FULL);
} // ***** NUEVO
else{ // ***** NUEVO
digitalWrite(2, LOW); // ***** NUEVO
Serial.write('H');
easyvr.playSound(8, EasyVR::VOL_FULL);
} // ***** NUEVO
group = GROUP_0; //<-- or jump to another group X for composite commands *****
MODIFICADO
break;
case G1_COSINA:
//Serial .print("cosina encendida");
if(digitalRead(3) == LOW){ // ***** NUEVO
digitalWrite(3, HIGH); // ***** NUEVO
Serial.write('A');
easyvr.playSound(9, EasyVR::VOL_FULL);
} // ***** NUEVO
else{ // ***** NUEVO
digitalWrite(3, LOW); // ***** NUEVO
Serial.write('B');
easyvr.playSound(8, EasyVR::VOL_FULL);
} // ***** NUEVO
group = GROUP_0; // <-- or jump to another group X for composite commands
;***** MODIFICADO
break;
case G1_COMEDOR:
//Serial .print("cosina encendida");
if(digitalRead(4) == LOW){ // ***** NUEVO
digitalWrite(4, HIGH); // ***** NUEVO
Serial.write('C');

```

```

easyvr.playSound(9, EasyVR::VOL_FULL);
} // ***** NUEVO
else{ // ***** NUEVO
digitalWrite(4, LOW); // ***** NUEVO
Serial.write('D');
easyvr.playSound(8, EasyVR::VOL_FULL);
} // ***** NUEVO
group = GROUP_0;// <-- or jump to another group X for composite commands
;***** MODIFICADO
break;
case G1_DORMITORIO1:
//Serial .print("cosina encendida");
if(digitalRead(5) == LOW){ // ***** NUEVO
digitalWrite(5, HIGH); // ***** NUEVO
Serial.write('E');
easyvr.playSound(9, EasyVR::VOL_FULL);
} // ***** NUEVO
else{ // ***** NUEVO
digitalWrite(5, LOW); // ***** NUEVO
Serial.write('F');
easyvr.playSound(8, EasyVR::VOL_FULL);
} // ***** NUEVO
group = GROUP_0;// <-- or jump to another group X for composite commands
;***** MODIFICADO
break;
case G1_DORMITORIO2:
// if(digitalRead(6) == LOW){ // ***** NUEVO
// digitalWrite(6, HIGH); // ***** NUEVO
//
easyvr.playSound(9, EasyVR::VOL_FULL);
// } // ***** NUEVO
// else{ // ***** NUEVO
// digitalWrite(6, LOW); // ***** NUEVO
//

```

```

//      easyvr.playSound(8, EasyVR::VOL_FULL);
//  } // ***** NUEVO
group = GROUP_0;// <-- or jump to another group X for composite commands
;***** MODIFICADO
break;
case G1_ABRECOTINA:
// write your action code here
digitalWrite(6, HIGH);
Serial.write('T');
easyvr.playSound(5, EasyVR::VOL_FULL);
group = GROUP_0;// <-- or jump to another group X for composite commands
break;
case G1_CIERRACORTIN:
// write your action code here
digitalWrite(6, LOW);
Serial.write('J');
easyvr.playSound(6, EasyVR::VOL_FULL);
group = GROUP_0;// <-- or jump to another group X for composite commands
break;
case G1_PRENDE_CALEF:
// write your action code here
easyvr.playSound(3, EasyVR::VOL_FULL);
group = GROUP_0;// <-- or jump to another group X for composite commands
break;
case G1_APAGA_CALEF:
// write your action code here
easyvr.playSound(2, EasyVR::VOL_FULL);
group = GROUP_0;//<-- or jump to another group X for composite commands
break;
case G1_APAGA_TODO:
// write your action code here
Serial.write('Z');
easyvr.playSound(13, EasyVR::VOL_FULL);
digitalWrite(2, LOW); // ***** NUEVO

```

```

digitalWrite(3, LOW); // ***** NUEVO
digitalWrite(4, LOW); // ***** NUEVO
digitalWrite(5, LOW); // ***** NUEVO
digitalWrite(6, LOW); // ***** NUEVO
digitalWrite(7, LOW); // ***** NUEVO
group = GROUP_0;// <-- or jump to another group X for composite commands
break;
case G1_LUCES:
// write your action code here
easyvr.playSound(7, EasyVR::VOL_FULL);
group = GROUP_0;// <-- or jump to another group X for composite commands
break;
}
break;
}
}

```

## **Anexo 2. Código en Arduino de la interface para la comunicación entre el sistema de reconocimiento de voz y el sistema domótico.**

```

int dato=0;
int botonSube = 0;
int botnBaja=0;
//sensor presencia
const int cosina = 2;
int statecosina=0;
int lecturacosina;
int contadorcosina=0;
//detector puerta
const int comedor = 3;
int Statecomedor =0;
int lecturacomedor;
int contadorcomedor=0;
//cuarto1
const int dormitorio1=4;

```

```

int Statedormitorio1 =0;
int lecturadormitorio1;
int contadordormitorio1=0;
//cuarto2
const int sala=A4;
int Statesala =0;
int lecturasala;
int contadorsala=0;
const int subeCortina=A5;
const int bajaCortina=A6;
const int opcional1=5;
const int opcional2=6;
const int focoCosina=7;
const int focoComedor=8;
const int focoDormitorio=9;
const int focoSala=10;
const int focoSube=11;
const int focobaja=12;
void setup()
{
Serial.begin(9600);
pinMode(cosina, INPUT);
pinMode(comedor, INPUT);
pinMode(dormitorio1,INPUT);
pinMode(sala,INPUT);
pinMode(subeCortina,INPUT);
pinMode(bajaCortina, INPUT);
pinMode(opcional1, OUTPUT);
pinMode(opcional2,OUTPUT);
pinMode(focoCosina, OUTPUT);
pinMode(focoComedor,OUTPUT);
pinMode(focoDormitorio,OUTPUT);
pinMode(focoSala,OUTPUT);
pinMode(focoSube,OUTPUT);

```

```

pinMode(focobaja,OUTPUT);
digitalWrite(focoCosina,HIGH);
digitalWrite(focoComedor,HIGH);
digitalWrite(focoDormitorio,HIGH);
digitalWrite(focoSala,HIGH);
digitalWrite(focobaja,HIGH);
digitalWrite(focoSube,HIGH);
}
void loop()
{
byte dato;
/////////////////////////////////DATO SERIAL ///////////////////////////////////
SENSOR DE MOVIMIENTO DORMITORIO 1 ///////////////////////////////////
lecturacosina=digitalRead(cosina);
if (lecturacosina!=statecosina){
if (lecturacosina == LOW) {
contadorcosina++;
}
}
statecosina=lecturacosina;
//*****
*
/////////////////////////////////DETECTOR DE LA PUERTA ///////////////////////////////////
lecturacomedor= digitalRead(comedor);
if(lecturacomedor!=Statecomedor){
if (lecturacomedor == LOW) {
contadorcomedor++;
}
}
Statecomedor=lecturacomedor;
//*****
*
/////////////////////////////////DORMITORIO 1 ///////////////////////////////////
lecturdormitorio1= digitalRead(dormitorio1);

```

```

if(lecturadormitorio1 !=Statedormitorio1){
if(lecturadormitorio1==LOW){
contadordormitorio1++;
}
}
Statedormitorio1=lecturadormitorio1;
//*****
*
//////////////////////////////////DORMITORIO 2 //////////////////////////////////
lecturasala=analogRead(sala);
if(lecturasala != Statesala){
if(lecturasala<=200){
contadorsala++;
}
}
delay(200);
}
Statesala= lecturasala;
//#####33
if (Serial.available() > 0)
{
dato = Serial.read();
}
//////////////////////////////////COSINA //////////////////////////////////
// botonCosina=digitalRead(cosina);
if(dato=='A' || contadorcosina==1 ){
digitalWrite(focoCosina,LOW);
delay(500);
digitalWrite(focoCosina,HIGH);
contadorcosina=0;
}
if(dato=='B' || contadorcosina==1){
digitalWrite(focoCosina,LOW);
delay(500);
digitalWrite(focoCosina,HIGH);
}

```

```

contadorcosina=0;
}
//*****
*
////////////////////////////////////COMEDOR //////////////////////////////////////
//botonComedor = digitalRead(comedor);
if(dato=='C' || contadorcomedor==1 ){
digitalWrite(focoComedor,LOW);
delay(500);
digitalWrite(focoComedor,HIGH);
contadorcomedor=0;
}
if(dato=='D' || contadorcomedor==1 ){
digitalWrite(focoComedor,LOW);
delay(500);
digitalWrite(focoComedor,HIGH);
contadorcomedor=0;
}
////////////////////////////////////DORMITORIO //////////////////////////////////////
//botonDormitorio = digitalRead(dormitorio1);
if(dato=='E' || contadordormitorio1==1 ){
digitalWrite(focoDormitorio,LOW);
delay(500);
digitalWrite(focoDormitorio,HIGH);
contadordormitorio1=0;
}
if(dato=='F' || contadordormitorio1==1 ){
digitalWrite(focoDormitorio,LOW);
delay(500);
digitalWrite(focoDormitorio,HIGH);
contadordormitorio1=0;
}
//*****
*

```

```

////////////////////////////////SALA //////////////////////////////////
if(dato=='G' || contadorsala==1 ){
digitalWrite(focoSala,LOW);
delay(500);
digitalWrite(focoSala,HIGH);
contadorsala=0;
}
if(dato=='H' || contadorsala==1 ){
digitalWrite(focoSala,LOW);
delay(500);
digitalWrite(focoSala,HIGH);
contadorsala=0;
}
//*****
*

////////////////////////////////SUBE //////////////////////////////////
botonSube = analogRead(subeCortina);
if(dato=='I' || botonSube<=512 ){
Serial.println("subecortina");
digitalWrite(focoSube,LOW);
delay(500);
digitalWrite(focoSube,HIGH);
}
//*****
*

////////////////////////////////BAJA////////////////////////////////
botnBaja= analogRead(bajaCortina);
//*****
*

if(dato=='J' || botnBaja<=512 ){
digitalWrite(focobaja,LOW);
delay(500);
digitalWrite(focobaja,HIGH);
}

```

```

if(dato=='Z'){
//COSINA////////////////////////////////////
digitalWrite(focoCosina,LOW);
delay(500);
digitalWrite(focoCosina,HIGH);
///COMEDOR////////////////////////////////////
digitalWrite(focoComedor,LOW);
delay(500);
digitalWrite(focoComedor,HIGH);
///DORMITORIO////////////////////////////////7////////////////////////////////77
digitalWrite(focoDormitorio,LOW);
delay(500);
digitalWrite(focoDormitorio,HIGH);
//////////FOCO SALA////////77
digitalWrite(focoSala,LOW);
delay(500);
digitalWrite(focoSala,HIGH);
///BAJA CORTIN////////7
digitalWrite(bajaCortina,LOW);
delay(500);
digitalWrite(bajaCortina,HIGH);
}
}

```

### **Anexo 3. Código en Matlab para determinar el tiempo de respuesta del sistema “Domotic Voice”**

```

clc
Fs=8000;
y1=wavrecord(10*Fs,Fs,1)';
soundsc(y1);
wavwrite(y1,Fs,'tiempo');
word1=['tiempo'];
y1=chop_silencio(y1);
z1=enfasis(y1);

```

```

seg1=segmentos(z1);
c1=cepstrum(seg1);
f1=features(c1);
save tiempo.mat word1 f1 y1;
figure(3);
plot(z1);
figure(4);
plot(abs(fft(z1)));
[wav fs] = wavread('tiempo');
soundsc(wav);
wav=transpose(wav);
x1=wav(1,1:length(wav));
[X1 f1]=freqz(x1,1,65536,fs);
Ts=1.2/fs;
N=length(x1);
n=0:N-1;
msHz=length(f1)/2000;
rangoHz=round(msHz*1000):round(msHz*1319);
figure(1);
subplot(211);
plot(n*Ts,x1/.1);
grid on;
title('Señal en el tiempo');xlabel('s');ylabel('amplitud');
subplot(212);
plot(abs(fft(wav)));
grid on;
title('Señal en frecuencia');xlabel('Hz');

```

## REFERENCIAS BIBLIOGRAFICAS

- [1] J. Martínez, “Control Domótico Por Voz”. Universidad Politécnica de Valencia, 2009. [Online]. Disponible en: [http://lorien.die.upm.es/juancho/pfcs/JC/PFC\\_jcancela.pdf](http://lorien.die.upm.es/juancho/pfcs/JC/PFC_jcancela.pdf)
- [2] F. Falcone, I. Matías, J. Militino, A. Gabilondo, I. del Villar, C. Fernández, I. Castells, J. Nazabal, C. Ruiz, “Instalaciones de telecomunicaciones para edificios”. MARCOMBO S.A, 2013, pp73-103.
- [3] J. Huidobro, R. Millán. “Manual de domótica”. Creaciones Copyright, 2010, pp.126-132. [Online]. Disponible en: [https://books.google.com.ec/books?id=V6IzqqDcfF8C&dq=estandar+KNX&source=gbs\\_navlinks\\_s](https://books.google.com.ec/books?id=V6IzqqDcfF8C&dq=estandar+KNX&source=gbs_navlinks_s)
- [4] S. Gallardo Vázquez. “Técnicas y procesos en instalaciones domóticas y automáticas”. Paraninfo, 2013, pp. 2-10. [Online]. Disponible en: [https://books.google.com.ec/books?id=PahQAqAAQBAJ&dq=protocolo+lonworks&source=gbs\\_navlinks\\_s](https://books.google.com.ec/books?id=PahQAqAAQBAJ&dq=protocolo+lonworks&source=gbs_navlinks_s)
- [5] A. Hidalgo. “Diseño y construcción de un nodo concentrador en una red de alarmas utilizando ZigBee”, 2011. [Online], Escuela Politécnica Nacional. Disponible en: <http://bibdigital.epn.edu.ec/bitstream/15000/3925/1/CD-3658.pdf>
- [6] E. Pérez, F. Poceros, J. Villalobos, “Sistema de Seguridad por reconocimiento de voz”. Instituto Politécnico Nacional, México, 2013. [Online]. Disponible en: [http://itzamna.bnct.ipn.mx/dspace/bitstream/123456789/12309/1/Sistema%20de%20Seguridad%20por%20Reconocimiento%20de%20Voz%20\(Tesis%20de%20Ingenieria%20ESIME\).pdf](http://itzamna.bnct.ipn.mx/dspace/bitstream/123456789/12309/1/Sistema%20de%20Seguridad%20por%20Reconocimiento%20de%20Voz%20(Tesis%20de%20Ingenieria%20ESIME).pdf)
- [7] F. Bellesi, F. Ortiz, “Reconocimiento de voz para aplicación en domótica”. Universidad Tecnológica Nacional, 2008. [Online]. Disponible en: <http://www.frsn.utn.edu.ar/tecnicas3/problemas/Reconocimiento%20de%20voz.pdf>
- [8] J. Camargo, L. García, E. Gaona. ” Human Voice Recognition Applied to Domotics”, 2012. [Online]. Disponible en: [dialnet.unirioja.es/descarga/articulo/5038438.pdf](http://dialnet.unirioja.es/descarga/articulo/5038438.pdf)
- [9] A. Rodríguez, ” Simulador de domótica con interfaz vocal”. Universidad Autónoma de Barcelona, 2009. [Online]. Disponible en: <http://www.recercat.cat/bitstream/handle/2072/171363/WeltonRodriguezAnderR-ETISa2008-09.pdf?sequence=1>

- [10] AUTOMÁTICA E INSTRUMENTACIÓN, “Sistema para edificios inteligentes activado por la voz”. Tecnomarket, 2015. [Online]. Disponible en: <http://www.automaticaeinstrumentacion.com/es/notices/2015/09/un-sistema-para-edificios-inteligentes-activado-por-la-voz-43189.php>
- [11] J. González, “Desarrollo de una interface multimodal para un robot doméstico móvil. Control por voz y mando a distancia”. Universidad Politécnica de Madrid, 2009. [Online]. Disponible en: [http://lorien.die.upm.es/juancho/pfcs/JC/PFC\\_jcancela.pdf](http://lorien.die.upm.es/juancho/pfcs/JC/PFC_jcancela.pdf)
- [12] Cultura Cuántica-Actualidad en informática y tecnología, “Domótica controlada por voz con Raspberry Pi”, 2013. [Online]. Disponible en: <http://www.culturacuantica.com.ar/domotica-controlada-por-voz-con-raspberry-pi/>
- [13] I. González, A. Rodríguez, “Desarrollo de sistema doméstico abierto mediante control por voz”. [Online]. Disponible en: <https://tecnoconstruccion.files.wordpress.com/2012/02/desarrollo-de-sistema-domc3b3tico-abierto-controlado-por-voz.pdf>
- [14] O. Torrente, “ARDUINO, Curso práctico de formación”. RC Libros, 2013, pp. 71-83 [Online]. Disponible en: [https://books.google.com.ec/books?id=6cZhDmf7suQC&dq=arduino&hl=es&source=gbs\\_navlinks\\_s](https://books.google.com.ec/books?id=6cZhDmf7suQC&dq=arduino&hl=es&source=gbs_navlinks_s)
- [15] Isaac. P, “Análisis comparativo de las placas de Arduino”, 2014 [Online]. Disponible en: <http://comohacer.eu/analisis-comparativo-placas-arduino-oficiales-compatibles/>
- [16] Veear, “SmartVR Quick Start Guide”. [Online]. Disponible en: [http://www.diltronic.com/wordpress/wp-content/uploads/SmartVR\\_QuickStart.pdf](http://www.diltronic.com/wordpress/wp-content/uploads/SmartVR_QuickStart.pdf)
- [17] Veear, “Manual de usuario EasyVR 2.0”. [Online]. Disponible en: [http://www.veear.eu/files/easyvr\\_user\\_manual\\_3.6.7.pdf](http://www.veear.eu/files/easyvr_user_manual_3.6.7.pdf)
- [18] Arduino, “Arduino Uno”, 2016. [Online]. Disponible en: <https://www.arduino.cc/en/Main/ArduinoBoardUno>
- [19] Arduino, “Arduino Nano”. 2016. [Online]. Disponible en: <https://www.arduino.cc/en/Main/ArduinoBoardNano>
- [20] J. Ruiz Gutiérrez, “Implementación de sistemas de transmisión de datos y señores en redes inalámbricas con XBee integrado en la Plataforma Open Hardware de Arduino”. Creative Commons Attribution, 2012. [Online]. Disponible en: [http://unicarlos.com/\\_ARDUINO/Arduino%20+%20XBee.pdf](http://unicarlos.com/_ARDUINO/Arduino%20+%20XBee.pdf)

- [21] AGelectrónica, “Batería de tecnología LiPo”. [Online]. Disponible en: <http://www.agspecinfo.com/pdfs/9/91270002.PDF>
- [22] Hamtennas, “Las batería para portable”. 2014. [Online]. Disponible en: [http://hamtennas.com/docs/Baterias\\_portable.pdf](http://hamtennas.com/docs/Baterias_portable.pdf)
- [23] Electrónica, “Cálculo de un regulador de tensión con LM317”, 2012. [Online]. Disponible en: <http://electronica-electronics.com/Tecnica/LM317-calculo.html>
- [24] M. Angulo, J. Sarmiento, “El concepto de rentabilidad en marketing”. Universidad Javeriana, 2000. [Online]. Disponible en: <http://www.javeriana.edu.co/decisiones/Julio/documentos/marketing.pdf>
- [25] A. Acosta, “Sistema de Control Automático de una ducha eléctrica mediante reconocimiento de la voz para personas con discapacidad en las extremidades superiores”. Universidad Técnica de Ambato, 2015. [Online]. Disponible en: [http://repo.uta.edu.ec/bitstream/123456789/13062/1/Tesis\\_t1034ec.pdf](http://repo.uta.edu.ec/bitstream/123456789/13062/1/Tesis_t1034ec.pdf)
- [26] L. Robalino, “Diseño e implementación de un control remoto controlado por órdenes de voz para aplicaciones de control de una vivienda”. Escuela Politécnica Nacional, 2007. [Online]. Disponible en: <http://bibdigital.epn.edu.ec/bitstream/15000/4167/1/CD-0886.pdf>
- [27] J. Pérez, “Control domótico con dispositivos móviles”. Universidad Oberta de Catalunya, 2015. [Online]. Disponible en: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/42597/3/jperezgiTFM0615memoria.pdf>
- [28] M. Rodríguez, “Sistema de control remoto para aplicaciones domóticas a través de internet”. Universidad Autónoma de Madrid, 2014. [Online] Disponible en: <http://arantxa.ii.uam.es/~jms/pfcsteleco/lecturas/20141107MarioRodriguezCerezo.pdf>
- [29] G. Martínez, G. Aguilar, “Reconocimiento de voz basado en MFCC, SBC y Espectrogramas”. Universidad Politécnica Salesiana, 2013. [Online]. Disponible en: <http://dspace.ups.edu.ec/handle/123456789/8465>
- [30] M. Loyola, P. Becerra, “Manual para la aplicación de la tecnología ZigBee para edificios inteligentes en la ciudad de Cuenca”. Universidad Politécnica Salesiana, 2015. [Online]. Disponible en: <http://dspace.ups.edu.ec/handle/123456789/7986>
- [31] V. Méndez, D. Ramírez de la Cruz, “Diseño e implementación del sistema de movimiento direccional de una silla de ruedas para ser controlada por reconocimiento de un patrón de voz mediante electrónica potencia y motores DC como actuadores”.

Universidad Politécnica Salesiana, 2015. [Online]. Disponible en:  
<http://dspace.ups.edu.ec/handle/123456789/9148>