



**UNIVERSIDAD NACIONAL DE CHIMBORAZO**

**FACULTAD DE INGENIERÍA**

**ESCUELA DE INGENIERÍA ELECTRÓNICA Y  
TELECOMUNICACIONES.**

**Trabajo de grado previo a la obtención del Título de Ingeniero en Electrónica  
y Telecomunicaciones**

**Título del proyecto:**

**DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE GAFAS  
ELECTRÓNICAS PARA PERSONAS NO VIDENTES**

**AUTOR:**

Alex Vicente Molina Espinoza

**TUTOR:**

Ing. Aníbal Llanga

**Riobamba – Ecuador**

**AÑO 2015**

Los miembros del Tribunal de Graduación del proyecto de investigación de título **DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE GAFAS ELECTRÓNICAS PARA PERSONAS NO VIDENTES**, presentado por: Alex Vicente Molina Espinoza y dirigida por: Ing. Aníbal Llanga.

Una vez escuchada la defensa oral y revisado el informe final del proyecto de investigación con fines de graduación escrito en la cual se ha constatado el cumplimiento de las observaciones realizadas, remite la presente para uso y custodia en la biblioteca de la Facultad de Ingeniería de la UNACH.

Para constancia de lo expuesto firman:

-----  
**Ing. Carlos Peñafiel**  
**Presidente del Tribunal**

  
-----  
**Firma**

-----  
**Ing. Deisy Inca**  
**Miembro del Tribunal**

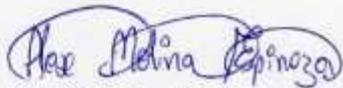
  
-----  
**Firma**

-----  
**Ing. Anibal Llanga**  
**Miembro del Tribunal**

  
-----  
**Firma**

## AUTORÍA DE LA INVESTIGACIÓN

La responsabilidad del contenido de este Proyecto de Graduación, nos corresponde exclusivamente a: Alex Vicente Molina Espinoza e Ingeniero Aníbal Llanga, Director del Proyecto. Algunas ideas, expresiones, pensamientos, conceptos se han tomado de varios autores como también del internet, ubicando la respectiva autoría como es norma de la investigación científica para enriquecer el marco teórico. En tal virtud los resultados tenidos son de exclusividad del autor, del director del proyecto de investigación y del patrimonio intelectual de la Universidad Nacional de Chimborazo.



Alex Vicente Molina Espinoza



Ing. Aníbal Llanga  
Director del Proyecto

## **AGRADECIMIENTO**

Mi más sincero agradecimiento a la Universidad Nacional de Chimborazo en especial la Facultad de Ingeniería y a la Escuela de Ingeniería Electrónica y Telecomunicaciones y a todos quienes me apoyaron en todo en el transcurso de este proceso en el cual me matriculé y hoy, gracias a ese apoyo recibido, estoy culminando mis estudios de Ingeniería, mención Telecomunicaciones.

**Alex Vicente**

## **DEDICATORIA**

A Dios, por darme la vida a través de mis padres quienes con mucho cariño, amor y ejemplo han hecho de mí una persona con valores para poder desenvolverme como profesional.

**Alex Vicente**

## ÍNDICE GENERAL

PÁGINA DE REVISIÓN.....	i
AUTORÍA DE LA INVESTIGACIÓN .....	iii
AGRADECIMIENTO .....	iv
DEDICATORIA .....	v
Índice de Tablas .....	ix
Índice de Gráficos .....	x
RESUMEN.....	xii
ABSTRACT.....	xiii
INTRODUCCIÓN .....	1

### CAPITULO I

1. <b>FUNDAMENTACIÓN TEÓRICA.....</b>	<b>3</b>
1.1.     Diseño .....	3
1.2.     Implementación .....	4
1.3.     Prototipo.....	4
1.4.     Personas no videntes .....	4
1.4.1.   Bastón blanco para personas no videntes.....	4
1.5.     Dispositivo .....	5
1.6.     Componentes electrónicos .....	6
1.6.1.   Clasificación.....	7
1.6.2.   Componentes Semiconductores .....	8
1.6.3.   Componentes activos .....	8
1.7.     Sensores .....	9
1.7.1.   Características de un sensor .....	10
1.7.2.   Clasificación de los sensores.....	11
1.7.3.   Estructura básica de un sensor (García Rodríguez) .....	13
1.7.4.   Sensores de Distancia.....	14
1.8.     Motor de corriente continua.....	22
1.8.1.   Variaciones en el diseño del motor .....	23
1.8.2.   Motor vibrador DC.....	25

1.9.	Microcontroladores .....	27
1.9.1.	Tarjeta Arduino .....	28
1.10.	Alertas de voz .....	33
1.10.1.	Sintetizador de voz estándar “Loquendo” .....	33

## **CAPÍTULO II**

2.	<b>METODOLOGÍA</b> .....	34
2.1.	Tipo de estudio.....	34
2.2.	Métodos, técnicas e instrumentos .....	34
2.2.1.	Métodos.....	34
2.2.1.1.	Analítico.....	34
2.2.2.	Técnica .....	35
2.3.	Población y muestra.....	35
2.4.	Operacionalización de variables.....	36
2.5.	Procedimientos.....	36
2.6.	Procesamiento y análisis .....	39
2.6.1.	Identificación y estudio del prototipo .....	39
2.6.2.	Selección y estudio de los equipos y sensores .....	40
2.6.3.	Test de comunicación entre el Arduino y los sensores .....	50
2.6.4.	Diseño de las etapas del prototipo.....	58
2.6.5.	Programación en el Entorno Arduino .....	59
2.6.6.	Ensamblaje de las etapas del dispositivo .....	66
2.6.7.	Recolección de datos mediante la consola del puerto serial .....	68
2.6.8.	Comprobación de la hipótesis .....	69

## **CAPÍTULO III**

3.	<b>RESULTADOS</b> .....	74
----	-------------------------	----

## **CAPÍTULO IV**

5.	<b>CONCLUSIONES Y RECOMENDACIONES</b> .....	78
5.1.	Conclusiones .....	78
5.2.	Recomendaciones .....	79

## **CAPÍTULO VI**

6.	<b>PROPUESTA</b> .....	80
6.1.	Título de la propuesta .....	80
6.2.	Introducción .....	80
6.3.	Objetivos:.....	80
6.3.1.	Objetivo general .....	80
6.3.2.	Objetivos específicos .....	80
6.4.	Fundamentación científico – técnica .....	81
6.5.	Descripción de la propuesta.....	81
6.6.	Diseño organizacional.....	82
6.7.	Monitoreo y evaluación de la propuesta .....	83
7.	<b>BIBLIOGRAFÍA</b> .....	84
8.	Apéndices y anexos.....	86

## ÍNDICE DE TABLAS

Tabla 1: Operacionalización de Variables .....	36
Tabla 2: Equipos y materiales a utilizarse.....	40
Tabla 3: Características de la tarjeta Arduino MEGA .....	42
Tabla 4: Características del sensor ultrasónico PING))).....	44
Tabla 5: Pines de conexión del sensor ultrasónico PING))) .....	45
Tabla 6: Características del módulo WTV020SD.....	46
Tabla 7: Pines de conexión del módulo WTV020SD .....	47
Tabla 8: Características del motor No. 280.....	48
Tabla 9: Pines de conexión del controlador de volumen PAM8403.....	50
Tabla 10: Valores de los tiempos de las personas que usaron las gafas .....	71

## ÍNDICE DE FIGURAS

Figura 1: Un bastón blanco desdoblado .....	5
Figura 2: Tipos de componentes electrónicos puestos en conjunto .....	6
Figura 3: Ejemplos de Sensores .....	9
Figura 4: Anemómetro digital .....	11
Figura 5: Sensor Fotoeléctrico .....	13
Figura 6: Sensor de tránsito infrarrojo .....	13
Figura 7: Estructura de un sensor .....	14
Figura 8: Sección transversal de un sensor ultrasónico .....	15
Figura 9: Esquema de detección de un sensor ultrasónico de una pieza.....	17
Figura 10: Esquema de detección de un sensor ultrasónico de dos piezas .....	18
Figura 11: Detección directa .....	19
Figura 12: Medición mediante barrera .....	20
Figura 13: Medición por reflexión .....	20
Figura 14: Medición por cabezal doble.....	21
Figura 15: Ángulo de incertidumbre .....	21
Figura 16: Zona ciega del sensor ultrasónico.....	22
Figura 17: Rotor y estator de un motor DC .....	23
Figura 18: Motor de imán permanente.....	24
Figura 19: Motor sin escobillas.....	25
Figura 20: Motor vibrador de semicilindro .....	26
Figura 21: Motor vibrador en semidisco .....	26
Figura 22: Partes que integran un microcontrolador.....	27
Figura 23: Entorno Arduino para Microsoft Windows .....	29
Figura 24: Boceto con un ejemplo de lectura análoga serial .....	30
Figura 25: Menú para acceder al sketchbook.....	30
Figura 26: Menú de librerías disponibles en el software Arduino .....	31
Figura 27: Botón de acceso al monitor serial.....	32
Figura 28: Shields para Arduino apilados.....	33
Figura 29: Diagrama de Procedimientos.....	37
Figura 30: Placa Arduino MEGA con procesador ATmega2560 .....	41
Figura 31: Diagrama de conexión de la tarjeta Arduino.....	43

Figura 32: Sensor de distancia ultrasónico PING)))	43
Figura 33: Diagrama eléctrico del sensor PING)))	44
Figura 34: Reproductor de sonido WTV020-SD	45
Figura 35: Diagrama eléctrico del reproductor WTV020-SD	46
Figura 36: Tarjeta shield para colocar sobre el Arduino	47
Figura 37: Micro motor vibrador No. 280	48
Figura 38: Especificaciones técnicas del motor vibrador No. 280	49
Figura 39: Controlador de volumen PAM8403	49
Figura 40: Diagrama eléctrico del controlador de voltaje PAM8403	50
Figura 41: Ubicación de la librería del sensor PING)))	51
Figura 42: Lista de librerías en el entorno Arduino	51
Figura 43: Conexión entre la tarjeta Arduino y los sensores PING)))	53
Figura 44: Programa de instalación del sintetizador Loquendo	54
Figura 45: Secciones del entorno del programa Loquendo	54
Figura 46: Grabación del texto sintetizado	55
Figura 47: Secciones del entorno del programa codificador	56
Figura 48: Archivo original y archivo convertido a formato .ad4	56
Figura 49: Librería del reproductor WTV020 resaltada en la lista	57
Figura 50: Tarjeta Arduino MEGA con la placa shield	58
Figura 51: Ubicación de los sensores ultrasónicos sobre las gafas	59
Figura 52: Placa Arduino y placa shield ubicadas dentro de la caja plástica	67
Figura 53: Etapas del dispositivo ensambladas	68
Figura 54: Consola serial mostrando datos recolectados del sensor ultrasónico	68
Figura 55: Tabla de t student	73
Figura 56: Campana de Gauss para desición	73
Figura 57: Plano del pasillo y la oficina usada para las pruebas	74

## RESUMEN

Actualmente en nuestro país, la inclusión de las personas con capacidades diferentes, es un tema que ha tomado fuerza en los últimos años, pero, a pesar de los esfuerzos realizados, existen barreras que todavía no se pueden superar. En base a las necesidades de este sector, el proyecto de investigación “Diseño e Implementación de un prototipo de gafas electrónicas para personas no videntes” fue desarrollado con la finalidad de facilitar a las personas no videntes, la movilidad a través de cualquier espacio, ya que por lo general, esta tarea la realiza con la ayuda de otra persona o valiéndose del bastón blanco, ocupando, en el segundo caso, sus manos para cumplir esta acción. Con la finalidad de liberar las manos del usuario cuando este camine y mediante el uso de sensores ultrasónicos de proximidad, conjuntamente con la integración a la plataforma Arduino, específicamente usando la tarjeta Arduino MEGA 2560, se diseñó e implementó un prototipo adherido a unas gafas que lleva puesto el no vidente, siendo capaz de detectar obstáculos en varios rangos seleccionables con ayuda de comandos de voz. De esta manera alertará al usuario con una leve vibración, cuando esté próximo a un objeto, facilitándole así el desplazamiento en un entorno de trabajo. Se utilizó el método Descriptivo – Explicativo, mediante la técnicas de la observación y experimentación, se aplicó una prueba con una muestra de 12 personas, con las cuales se utilizó el bastón blanco y el prototipo para la comprobación de la hipótesis, para luego llegar a las conclusiones y recomendaciones.



UNIVERSIDAD NACIONAL DE CHIMBORAZO  
FACULTAD DE INGENIERÍA  
CENTRO DE IDIOMAS



Lic. Daniela Castillo

11 de mayo de 2015

**ABSTRACT**

Currently in our country, the inclusion of people with disabilities is an issue that has gained strength in recent years, but, despite all efforts, there are still barriers that cannot be overcome. Based on the needs of this sector, the research project "Design and Implementation of an electronic glasses prototype for the blind people" was developed in order to facilitate blind people, mobility through any space. usually, this task is performed with the help of another person or making use of the white cane, in the second case, using the hands to complete this action. In order to free the hands of the user when walking and using ultrasonic proximity sensors, together with integration into the Arduino platform, specifically using the Arduino MEGA 2560 card, a prototype was designed and implemented attached to a pair of glasses worn by the blind, it is able to detect obstacles in several selectable ranges using voice commands. In this way it will alert the user with a slight vibration when an object approaches, making it easy for displacement in a working environment. The descriptive - explanatory method was used, through the technique of observation and experimentation, a test was applied to a sample of 12 people, with which the white cane and the prototype were used for hypothesis testing, then reaching conclusions and recommendations.

CENTRO DE IDIOMAS



COORDINADOR

## INTRODUCCIÓN

En nuestro país, las limitaciones y restricciones en los espacios, tanto públicos como privados, no ha permitido que las personas con discapacidades puedan integrarse de una manera más segura a las labores cotidianas que un ciudadano realiza dentro de la sociedad.

En la ciudad de Riobamba no existe ningún tipo de ayuda en lo referente a las personas con discapacidad visual ya que ellas tienen que trasladarse de un lugar a otro por sus propios medios, en ocasiones usando como bastón o soporte a un pariente o personas que le ayude, en otras ocasiones algunas personas utilizan el bastón para trasladarse de un lugar a otro, no siendo este recurso tan útil ya que al momento de caminar se van tropezando con los transeúntes o con algún obstáculo que se encuentre en la calle.

La movilización de un lugar a otro, así como también el desplazarse dentro de un espacio público (edificios, estadios, parques, etc.) o privado (casas, departamentos) es, tal vez, el mayor problema que enfrentan las personas con deficiencia visual, porque dichos lugares no cuentan con acoplamientos o adaptaciones recomendadas para que ellos puedan moverse de un punto a otro, evitando la mayor cantidad de obstáculos.

Por los antecedentes expuestos anteriormente, la presente investigación centra su objeto de estudio en el diseño e implementación de un prototipo de gafas electrónicas capaz de ayudar a las personas no videntes a captar de mejor manera su entorno, dado que al poder apreciar mejor el espacio por donde se está trasladando, dependerá menos de otras personas para llegar a su destino.

La hipótesis que se planteó en la presente investigación fue verificar si el diseño e implementación de prototipo de gafas electrónicas permitirá que las personas no videntes puedan movilizarse evadiendo obstáculos de mejor manera que al usar un bastón blanco de seguridad.

De ahí que la presente investigación tiene los siguientes capítulos: Capítulo I, se describe el enfoque teórico y la definición de los términos básicos necesarios para el desarrollo de la idea principal del funcionamiento del prototipo, en el Capítulo II se describe el Marco Metodológico que se utilizó en la investigación, así como las técnicas e instrumentos de la investigación empleadas y los pasos que se siguieron para lograr implementar de manera real el dispositivo. En el Capítulo III se muestra los resultados obtenidos, en el Capítulo IV se plantea la discusión de la investigación. Luego en el Capítulo V se expone las conclusiones y las recomendaciones, en el Capítulo VI se explica la propuesta y en la parte final de la investigación se presenta la Bibliografía y Anexos.

## **CAPITULO I**

### **1. FUNDAMENTACIÓN TEÓRICA.**

A lo largo de este trabajo de investigación se pretende mostrar el desarrollo de un dispositivo de sustitución visual-táctil de fácil uso, que brinde solo la información necesaria, y que sea de bajo costo, de modo que esté disponible a toda la sociedad por igual.

Asumiendo que en la actualidad existen desarrollos que brindan demasiada información al usuario, el objetivo de este trabajo fue desarrollar un dispositivo que minimice esta información a solo la necesaria para brindarle al usuario un área de seguridad cercana, sin verse abrumado por los datos recibidos y además que resulte simple de utilizar.

#### **1.1. DISEÑO**

Es el proceso que precede a una configuración, “pre-configuración”, en la investigación de solución en cualquier campo. Utilizándolo regularmente en el contexto de la industria, ingeniería, arquitectura, comunicación y otras disciplinas creativas siendo el paso previo a la búsqueda de una solución o un conjunto de soluciones. Diseñar requiere principalmente consideraciones funcionales, estéticas y simbólicas (Diccionario, 2014).

El proceso de diseñar, se suele aclarar con las siguientes fases: Observar y analizar el medio en el cual se desenvuelve el ser humano, descubriendo alguna necesidad. Evaluar, mediante la organización y prioridad de las necesidades identificadas. Planear y proyectar proponiendo un modo de solucionar esta necesidad, por medio de planos y maquetas, tratando de descubrir la posibilidad y viabilidad de la solución o soluciones. Ver, Construir y ejecutar llevando a la vida real la idea inicial, por medio de la investigación y la experimentación, para poder satisfacer la necesidad anteriormente establecida, de manera sencilla y en lo posible, proponiendo mejoras para un futuro desarrollo de la idea (ibídem).

## **1.2. IMPLEMENTACIÓN**

Es la construcción de una aplicación, instalación o la ejecución de un plan, idea, modelo científico, diseño, especificación, estándar, algoritmo o política (ibídem).

## **1.3. PROTOTIPO**

Es la primera versión o modelo de un producto o idea, en que se han incorporado algunas características del producto final. Se crean con rapidez y a un costo bajo para explorar la factibilidad del concepto preliminar. Se pueden fabricar a mano y de materias simples, pero también se puede contratar la pericia de un ingeniero, diseñador o desarrollador profesional. El objetivo del prototipo es ayudar al emprendedor a visualizar y refinar un producto para encontrar fallas en su aplicación (ibídem).

## **1.4. PERSONAS NO VIDENTES**

Una persona no vidente es aquella que padece de ceguera. La **ceguera** es una discapacidad física que consiste en la pérdida total o parcial del sentido de la vista. Existen varios tipos de ceguera dependiendo del grado y tipo de pérdida de visión, como la visión reducida, el escotoma, la ceguera parcial (de un ojo), total (ambos ojos) o el daltonismo (ibídem).

### **1.4.1. Bastón Blanco para personas no videntes**

El bastón blanco es una vara alargada, que sirve a las personas no videntes de guía para desplazarse de manera autónoma por la vía pública. El uso correcto del bastón blanco implica emplear un conjunto de técnicas para que una persona pueda desplazarse con cierta seguridad. Cuando se emplea la técnica rítmica de contacto de dos puntos, la puntera debe describir un arco levemente superior al ancho del cuerpo de la persona.



**Figura 1:** Un bastón blanco desdoblado.

Fuente: Universidad de Valladolid (Instituto Universitario)

Su uso data desde el año 1921, pero es en febrero de 1931, el francés Guilly d'Herbemont lanzó una campaña sobre el bastón blanco en Reino Unido, hasta que en mayo la BBC decidió expandirla en sus emisiones radiofónicas, siendo así reconocido y convertido en un símbolo para la sociedad.

La presente investigación se enfoca en proveer una alternativa al uso del bastón para las personas no videntes, mejorando la movilidad tanto en exteriores como interiores, y liberando el uso de las manos para otras actividades.

## **1.5. DISPOSITIVO**

El término dispositivo, en computación, puede hacer referencia a:

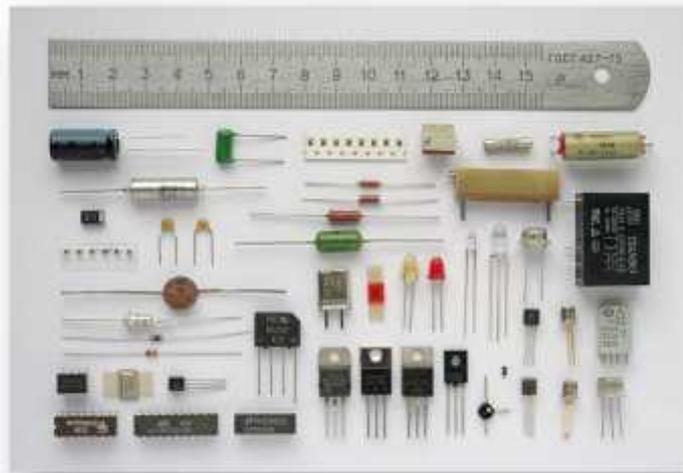
- a) Un dispositivo de almacenamiento de datos, cualquier componente que lee y escribe datos en un medio de almacenamiento.
- b) Un dispositivo de almacenamiento de acceso directo, cualquier tipo de dispositivo de almacenamiento secundario que tiene un tiempo de acceso bajo en proporción a su capacidad.
- c) Un dispositivo apuntador, componente que permite al usuario introducir datos de espacio a una computadora.
- d) El dispositivo de caracteres, componente de la computadora que se comunica con la unidad central de proceso por medio de bytes individuales.

- e) El dispositivo de bloques, componente de la computadora en el que los datos se transmiten en conjuntos indivisibles en la comunicación con la unidad central de proceso.

Además, puede referirse a:

Un dispositivo Braille, cualquier aparato electrónico que sirve para la interpretación o generación de lenguaje braille, tanto de forma física como virtual (Rodríguez, 2014).

## 1.6. COMPONENTES ELECTRÓNICOS



**Figura 2:** Tipos de componentes electrónicos puestos en conjunto

**Fuente:** <https://i.warosu.org/>

Se denomina componentes electrónicos a un dispositivo que forma parte de un circuito electrónico. Se suele encapsular, generalmente en un material cerámico, metálico o plástico, y terminar en dos o más terminales o patillas metálicas. Se diseñan para ser conectados entre ellos, normalmente mediante soldadura, a un circuito impreso.

Hay que diferenciar entre componentes y elementos. Los componentes son dispositivos físicos, mientras que los elementos son modelos o abstracciones idealizadas que constituyen la base para el estudio teórico de los mencionados

componentes. Así, los componentes aparecen en un listado de dispositivos que forman un circuito, mientras que los elementos aparecen en los desarrollos matemáticos de la teoría de circuitos (<http://commons.wikimedia.org/>).

### 1.6.1. CLASIFICACIÓN

De acuerdo con el criterio que se elija podemos obtener distintas clasificaciones. Seguidamente se detallan las comúnmente más aceptadas.

#### a) Según su estructura física

**Discretos:** son aquellos que están encapsulados uno a uno, como es el caso de los resistores, condensadores, diodos, transistores, etc.

**Integrados:** forman conjuntos más complejos, como por ejemplo un amplificador operacional o una puerta lógica, que pueden contener desde unos pocos componentes discretos hasta millones de ellos. Son los denominados circuitos integrados.

#### b) Según el material base de fabricación.

Semiconductores y no semiconductores.

#### c) Según su funcionamiento.

**Activos:** proporcionan excitación eléctrica, ganancia o control

**Pasivos:** son los encargados de la conexión entre los diferentes componentes activos, asegurando la transmisión de las señales eléctricas o modificando su nivel.

**d) Según el tipo de energía.**

**Electromagnéticos:** aquellos que aprovechan las propiedades electromagnéticas de los materiales (fundamentalmente transformadores e inductores).

**Electroacústicos:** transforman la energía acústica en eléctrica y viceversa (micrófonos, altavoces, bocinas, auriculares, etc.).

**Optoelectrónicos:** transforman la energía luminosa en eléctrica y viceversa (diodos LED, células fotoeléctricas, etc.). (ibídem).

### **1.6.2. Componentes Semiconductores**

Un semiconductor es una sustancia que se comporta como conductor o como aislante dependiendo varios factores a los que sea expuesto, como pueden ser: la temperatura del ambiente, el campo eléctrico o magnético al que sea sometido, la presión, entre otros.

El elemento semiconductor que más se usa es el silicio (Si), el segundo el germanio (Ge).

### **1.6.3. Componentes activos**

Los componentes activos son aquellos que son capaces de controlar los circuitos o de realizar ganancias. Fundamentalmente son los generadores eléctricos y ciertos componentes semiconductores. Estos últimos, en general, tienen un comportamiento no lineal, esto es, la relación entre la tensión aplicada y la corriente demandada no es lineal.

Los componentes activos semiconductores derivan del diodo de Fleming y del triodo de Lee de Forest. En una primera generación aparecieron las válvulas que permitieron el desarrollo de aparatos electrónicos como la radio o la televisión.

Posteriormente, en una segunda generación, aparecerían los semiconductores que más tarde darían paso a los circuitos integrados (tercera generación) cuya máxima expresión se encuentra en los circuitos programables (microprocesador y micro controlador) que pueden ser considerados como componentes, aunque en realidad sean circuitos que llevan integrados millones de componentes.

En la actualidad existe un número elevado de componentes activos, siendo usual, que un sistema electrónico se diseñe a partir de uno o varios componentes activos cuyas características lo condicionará. Esto no sucede con los componentes pasivos. En la siguiente tabla se muestran los principales componentes activos junto a su función más común dentro de un circuito.

## 1.7. SENSORES

Un sensor es un dispositivo que puede convertir magnitudes físicas o químicas, llamadas variables de instrumentación, en variables eléctricas. Las variables de instrumentación pueden ser, por ejemplo: temperatura, intensidad lumínica, distancia, aceleración, inclinación, desplazamiento, presión, fuerza, torsión, humedad, movimiento, pH, etc. Una magnitud eléctrica puede ser una resistencia eléctrica (como en una RTD), una capacidad eléctrica (como en un sensor de humedad), una Tensión eléctrica (como en un termopar), una corriente eléctrica (como en un fototransistor), etc. (es.wikipedia.org/wiki/Sensor, 2014)



**Figura 3:** Ejemplos de Sensores

**Fuente:** home.roboticlab.eu

Al estar conectado siempre a la variable de instrumentación, podemos decir que aprovecha una de sus propiedades, con el fin de adaptar la señal que mide para que la pueda interpretar otro dispositivo. Como ejemplo podemos citar un fototransistor, que determina la activación o desactivación de una bombilla dependiendo de la luz ambiental presente en ese momento. En otras palabras, un sensor puede convertir un tipo de energía a un impulso eléctrico.

Áreas de aplicación de los sensores: Industria automotriz, robótica, industria aeroespacial, medicina, industria de manufactura, etc. (ibídem).

### 1.7.1. Características de un sensor

Dentro de las características de los sensores, podemos citar las siguientes:

- **Rango de medida:** dominio en la magnitud medida en el que puede aplicarse el sensor.
- **Precisión:** es el error de medida esperado.
- **Offset o desviación de cero:** valor de la variable de salida cuando la variable de entrada es nula.
- **Linealidad o correlación lineal.**
- **Sensibilidad:** es la relación que existe entre la variación de la magnitud de entrada y la variación de la magnitud de salida
- **Resolución:** mínima variación de la magnitud de entrada que puede apreciarse a la salida.
- **Rapidez de respuesta:** Es la rapidez con la que el sensor entrega los datos al dispositivo al cual está conectado. Depende de la magnitud que se desea medir y también de la capacidad del sistema al que está conectado para seguir las variaciones de dicha magnitud
- **Repetitividad:** Es el error esperado al repetir varias veces la misma medida. Depende del número de repeticiones y de la precisión del sensor

También un sensor es un tipo de transductor que transforma la magnitud que se quiere medir o controlar, en otra, que facilita su medida. Pueden ser de indicación directa o pueden estar conectados a un indicador (puede ser a través de un display, aprovechando una conversión analógica a digital) de modo que los valores detectados puedan ser leídos por una persona. (ibídem).



Figura 4: Anemómetro digital

Fuente: Gill Instruments

Por lo general, la señal de salida de estos sensores no es apta para su lectura directa y a veces tampoco para su procesado, por lo que se usa un circuito de acondicionamiento, amplificadores y filtros electrónicos que adaptan la señal a los niveles apropiados para que el resto de los circuitos puedan interpretar dicha señal.

### 1.7.2. Clasificación de los sensores

Los sensores se clasifican dependiendo de la señal o magnitud de entrada al dispositivo, la señal de salida, y la naturaleza de la señal generada. (Ortega, 2014)

#### 1.7.2.1. Según el tipo de señal de entrada:

- **Mecánica:** Son aquellos sensores que miden una propiedad extrínseca de la materia, o la interacción entre dos fuerzas de distinto tipo. Como ejemplo

tenemos longitud, masa, velocidad, aceleración, fuerza, torque, presión, intensidad acústica, longitud de onda.

- **Química:** Son los sensores que pueden identificar las propiedades intrínsecas de la materia, como por ejemplo concentración, composición, potencial de oxidación/reducción, pH.
- **Eléctrica:** Son los sensores que son capaces de captar la interacción de los electrones en una corriente, como por ejemplo tenemos de voltaje, corriente eléctrica, constante dieléctrica, polarización, campo eléctrico, frecuencia dipolar.
- **Magnética:** Son los sensores que miden cambios en la intensidad de un campo magnético, como también su densidad y permeabilidad.
- **Térmica:** En este grupo tenemos los sensores que detectan la variación de la temperatura en el ambiente o en un determinado material, así como también el flujo de calor que se puede transmitir de un cuerpo a otro.

#### 1.7.2.2. Según el tipo de señal de salida

- **Sensores Analógicos:** Componen la mayor parte de sensores disponibles en el mercado, y son aquellos que entregan su señal continua en el tiempo, como por ejemplo los sensores generadores de señal.
- **Sensores Digitales:** Son aquellos sensores que poseen una salida de carácter discreto. Como ejemplo tenemos los sensores de posición, sensores codificadores incrementales, sensores auto resonantes, entre otros.

#### 1.7.2.3. Según la naturaleza de la señal generada

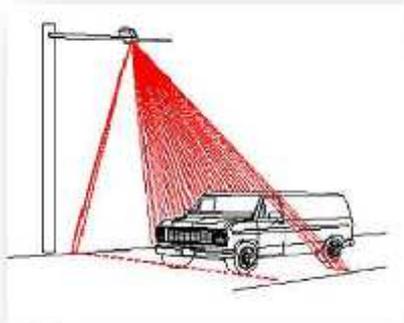
- **Sensores pasivos:** Son aquellos que utilizan una señal externa o auxiliar para poder realizar la medición de la magnitud. Ejemplos de estos sensores son los sensores de luz, de temperatura, de parámetros variables.



**Figura 5:** Sensor Fotoeléctrico

Fuente: <http://www.electan.com/images/C-7288.jpg>

- **Sensores activos:** Se los conoce también como sensores generadores de señal y son aquellos que no requieren de una señal externa para poder realizar la medición de una magnitud, ya que son capaces de emitir una señal propia. Como ejemplo de este tipo de sensores tenemos los sensores ultrasónicos, infrarojos, electroquímicos, fotovoltaicos, piezoeléctricos.



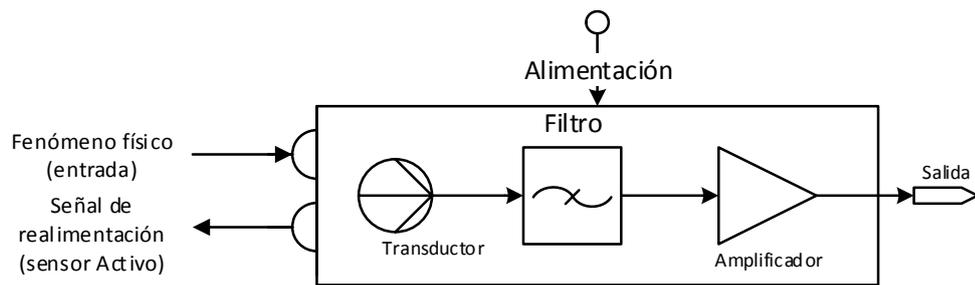
**Figura 6:** Sensor de tránsito infrarrojo

Fuente: Ingeniería de tránsito TYSSA

Algunas magnitudes pueden calcularse mediante la medición y cálculo de otras, por ejemplo, la velocidad de un móvil puede calcularse a partir de la integración numérica de su aceleración. La masa de un objeto puede conocerse mediante la fuerza gravitatoria que se ejerce sobre él en comparación con la fuerza gravitatoria ejercida sobre un objeto de masa conocida (patrón).

### 1.7.3. Estructura básica de un sensor (García Rodríguez)

Generalmente, los sensores se componen de los siguientes elementos:



**Figura 7:** Estructura de un sensor

**Fuente:** Alex Molina Espinoza

- **Transductor:** Es el que se encarga de convertir la variación de la magnitud recibidas del fenómeno físico en forma de energía, en variaciones de una magnitud eléctrica, es decir las transforma en una señal eléctrica.
- **Filtro:** Luego de recibida la señal eléctrica, el filtro se encarga de diferenciar entre la frecuencia recibida, y que será usada en el proceso de sensado, del ruido o cualquier otra señal que no sea la que se está midiendo.
- **Amplificador:** Dado que la señal convertida a magnitud eléctrica es de muy baja energía, la mayoría de las veces se requiere alterar esta señal para que pueda obtenerse una adecuada lectura de los datos, ya sea necesaria una amplificación, una modulación, etc.

#### 1.7.4. Sensores de Distancia

Los sensores de distancia están diseñados para realizar la medida de distancia lineal o desplazamiento lineal de una forma automatizada, al proporcionan una señal eléctrica según la variación física, que en este caso es la longitud. (ibídem).

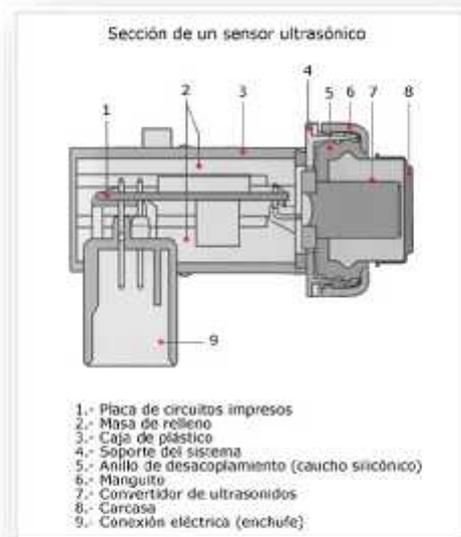
Los rangos de medida disponibles dependen de la distancia a medir; así pues hay modelos que tienen rangos de unas pocas micras y otros modelos que pueden llegar a medir cientos de metros. El formato del sensor varía conforme con tipo de tecnología usado en su construcción, siendo los principales tipos los siguientes:

- Sensores de distancia por cable

- Sensores de distancia por cinta
- Encoder lineal magnético
- Sensores de distancia magnetostrictivos
- Potenciómetros lineales
- Palpadores inductivos
- Sensores de distancia LVDT
- Sensores de distancia por Láser
- Láser de larga distancia
- Laser de perfil 2D
- Sensores ATEX
- Sensor de distancia por ultrasonidos

### 1.7.5. Sensor de distancia por ultrasonido

Los sensores ultrasónicos son detectores de proximidad que trabajan sin la interferencia de roces mecánicos y que pueden detectar un objeto a una distancia de entre 6 a 8 metros. (Farcy, 2006)



**Figura 8:** Sección transversal de un sensor ultrasónico

**Fuente:** [piemania.garcia-cuervo.net/rcdpictutorials/sensores](http://piemania.garcia-cuervo.net/rcdpictutorials/sensores)

Son sensores de tipo activo ya que tienen incorporado un emisor de ultrasonidos para realizar la medición de la distancia. Generalmente este tipo de sensores está formado por un emisor de ultrasonidos, un receptor (micrófono) y un circuito que se encarga de procesar los datos recibidos para calcular la distancia.

#### **1.7.5.1. Funcionamiento básico**

Los ultrasonidos utilizados por el sensor están en el rango de 40 khz – 400 kHz, siendo estos inaudibles por el oído humano. Este sonido es emitido por el sensor en ráfagas cortas de intervalos de 5 ms, el cual rebota en los objetos que estén frente al sensor. Luego el micrófono recibe el eco producido por el rebote, y mide el tiempo entre la emisión del sonido y el eco escuchado.

Dado que la velocidad del sonido en el aire a una temperatura ambiente de 20 °C es constante (300 m/s), el sensor calcula la distancia que ha recorrido la onda de sonido desde que salió del sensor hasta que regresó en forma de eco, solo conociendo el tiempo, mediante la fórmula del movimiento rectilíneo uniforme.

$$x = \frac{1}{2} v_s t$$

Donde  $v_s$  es la velocidad del sonido en el aire,  $t$  es el tiempo transcurrido y  $x$  es la distancia. Nótese que se debe dividir esta distancia para dos, puesto que la onda de sonido realiza el recorrido desde el sensor hasta el objeto, y desde el objeto de regreso al sensor. (ibídem).

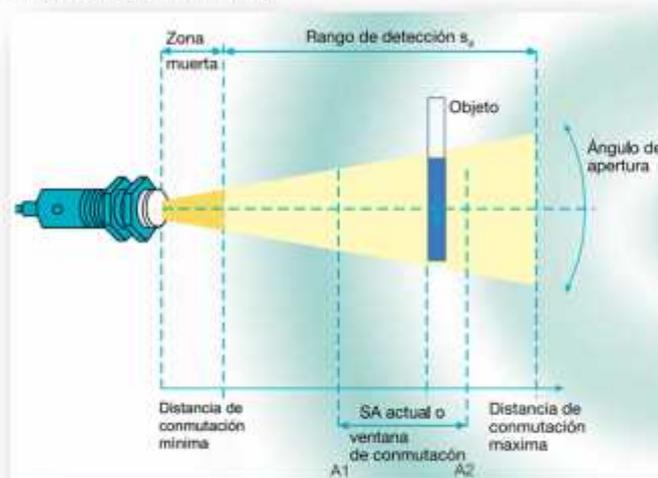
#### **1.7.5.2. Tipos de sensores ultrasónicos**

Existen principalmente dos tipos de sensores ultrasónicos, que se clasifican dependiendo de la ubicación del transmisor y receptor de señales.

### ➤ Sensores ultrasónicos de una sola pieza

Primero tenemos los sensores ultrasónicos que operan con una piezocerámica única que actúa como transmisor y receptor. Este transmisor está empotrado impermeablemente dentro de la carcasa del sensor, con espuma de poliuretano. Una de las principales desventajas de los sensores ultrasónicos de este tipo, es que, como el transmisor y receptor están en la misma pieza, el sensor, tras la emisión del ultrasonido, debe esperar un determinado tiempo para que las vibraciones desaparezcan y esté listo para recibir el eco producido por el obstáculo.

Esto implica que existe una distancia mínima de detección, a partir de la cual el sensor puede mostrar datos confiables. Por lo general, los objetos que se encuentren por debajo de esta distancia mínima, serán interpretados como si estuvieran a una distancia igual a la distancia mínima.



**Figura 9:** Esquema de detección de un sensor ultrasónico de una pieza

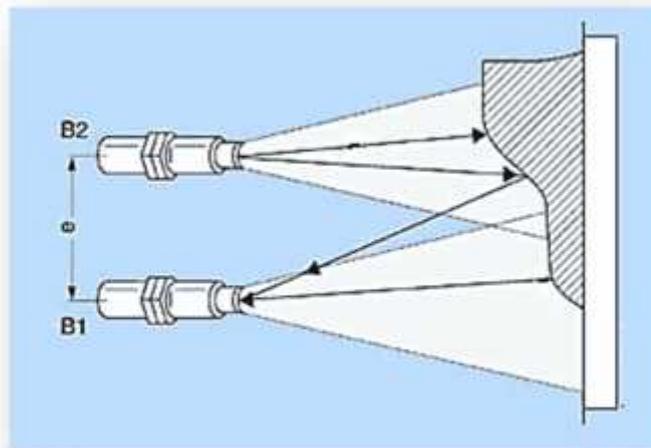
**Fuente:** pepperl-fuchs.com/ultrasonicos

Estos sensores utilizan una frecuencia ultrasónica entre 64 y 400 kHz, y la frecuencia utilizada para la repetición del impulso está entre 14 y 140 Hz. Este rango está ligado a la distancia mínima y máxima de sensibilidad, cuyos valores dependen del tipo de transductor usado en el armado del sensor. El cono sónico producido por el transmisor tiene un ángulo comprendido entre 30° y 40° y detecta los objetos que se muevan en una línea a través del eje del sensor o lateralmente en el interior del cono.

Por lo general, estos sensores están disponibles con salidas de conmutación, analógicas, o de interface RS-232. (ibídem).

➤ **Sensores ultrasónicos de dos piezas.**

En los sensores ultrasónicos de dos piezas, el dispositivo transmisor y el receptor están en partes separadas sobre el mismo sensor, haciendo de este más preciso, dado que la distancia de detección mínima (zona muerta) es menor. Las vibraciones emitidas por el emisor afectan en menor proporción a la señal recibida, lo que significa que la frecuencia de repetición es mayor, ubicándose entre 25 y 250 Hz.



**Figura 10:** Esquema de detección de un sensor ultrasónico de dos piezas

**Fuente:** [siemens.control-technics.com](http://siemens.control-technics.com)

La distancia  $e$  de separación entre el emisor y el receptor no debe superar los 15 mm, debido a que una mayor distancia influiría en el tiempo que tarda la onda de sonido al regresar al dispositivo, dando como resultado una lectura de distancia errónea.

Estos sensores también están dispuestos empotrados sobre la plaqueta del dispositivo, con recubrimiento de cerámica y rellenos de espuma de poliuretano. La frecuencia ultrasónica que utilizan está entre los 40 y 50 kHz y el cono sónico que

proyecta el emisor bordea los 50° de amplitud, haciendo que pueda detectar obstáculos dentro de una mayor área. (ibídem).

### 1.7.5.3. Configuraciones de la medición

Dependiendo del requerimiento, estos sensores pueden operar en varios modos de medición.

**Medición por detección directa:** Para realizar una detección directa, el transmisor y receptor están montados en la misma carcasa, y el sensor detecta el objeto por reflexión directa de las ondas de sonido. Es uno de los métodos más usados porque tiene como ventajas que el dispositivo será compacto y simple de operar. (García, 2013)

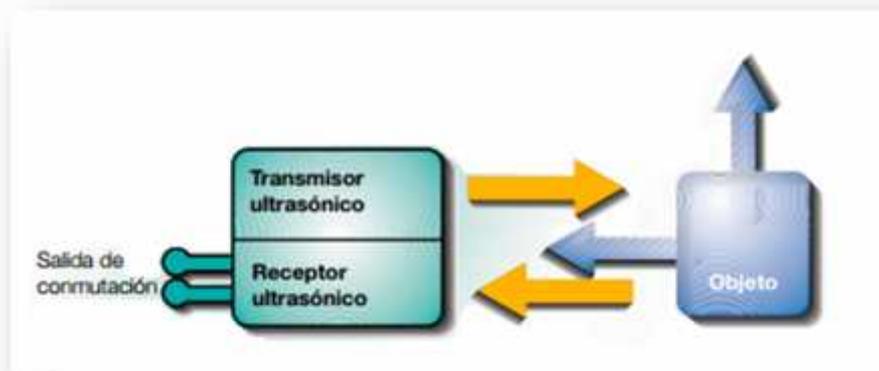
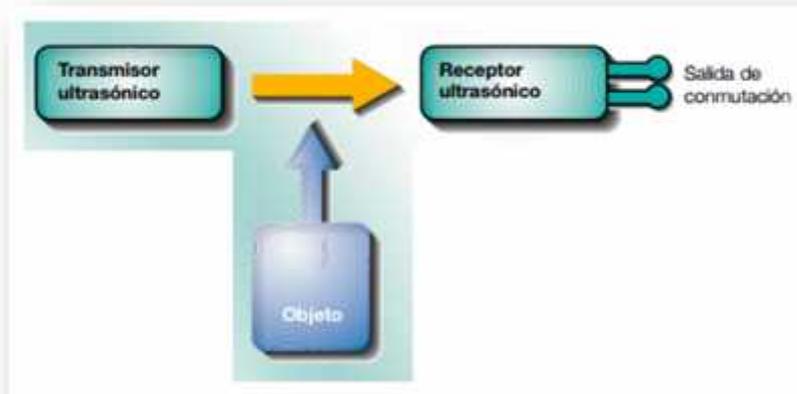


Figura 11: Detección directa

Fuente: [pepperl-fuchs.com/ultrasonicos](http://pepperl-fuchs.com/ultrasonicos)

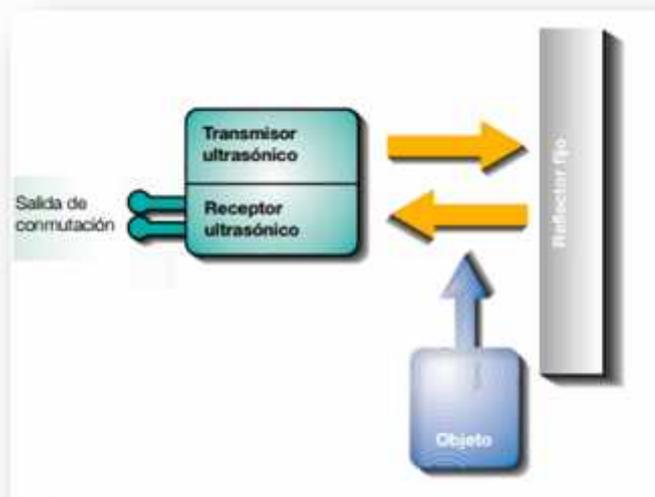
**Medición mediante barrera:** Es cuando el transmisor y el receptor están montados el uno al frente del otro, transmitiendo una señal de sonido constante. Si se interrumpe el haz ultrasónico, se activa la salida de conmutación. Este método tiene como ventaja que puede operar en rangos mayores a los 8 m. (ibídem).



**Figura 12:** Medición mediante barrera

**Fuente:** pepperl-fuchs.com/ultrasonicos

**Medición por reflexión:** Este método de medición requiere que el transmisor y el receptor estén montados en la misma carcasa, pero a diferencia del método de detección directa, se usa una placa reflectora fija para poder dirigir de mejor manera el haz ultrasónico. Este método es mayormente usado cuando se desea detectar de manera segura objetos cuyas superficies presentan poca o nula capacidad de reflexión. (ibídem).



**Figura 13:** Medición por reflexión

**Fuente:** pepperl-fuchs.com/ultrasonicos

**Medición por cabezal doble:** En un sensor con cabezal doble, el transmisor y el receptor están separados una cierta distancia, pero los ejes del transductor de cada

uno deben cruzarse entre sí para formar una zona de detección por medio de reflexión directa. Esta configuración es muy útil cuando se desean sensar objetos cuyas dimensiones son muy pequeñas, o cuando se tiene objetos que circulan a una velocidad determinada. (ibídem).

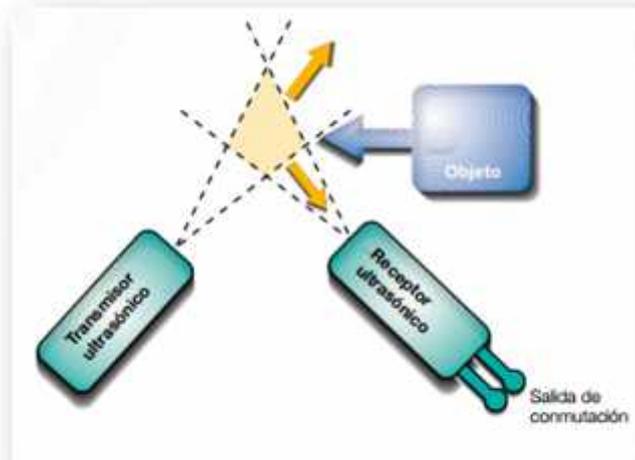


Figura 14: Medición por cabezal doble

Fuente: pepperl-fuchs.com/ultrasonicos

#### 1.7.5.4. Limitaciones de los sensores ultrasónicos

A pesar de su sencillo funcionamiento, existen varios factores que influyen de una forma u otra en las mediciones realizadas. Entre los diversos elementos que pueden alterar las lecturas adquiridas con el sensor, destacaremos las siguientes:

- El pulso ultrasónico emitido por el sensor tiene una forma cónica. El eco que se recibe como resultado de la reflexión del sonido, indica la presencia de un objeto dentro del cono acústico, mas no especifica la localización angular del mismo. Se debe tener en cuenta la probabilidad de que el objeto detectado se encuentre en la periferia del cono acústico y no en el eje central.

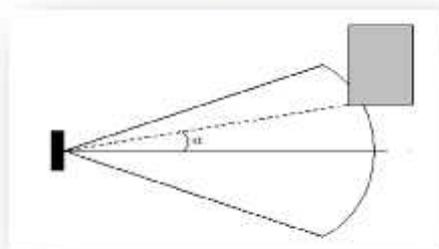
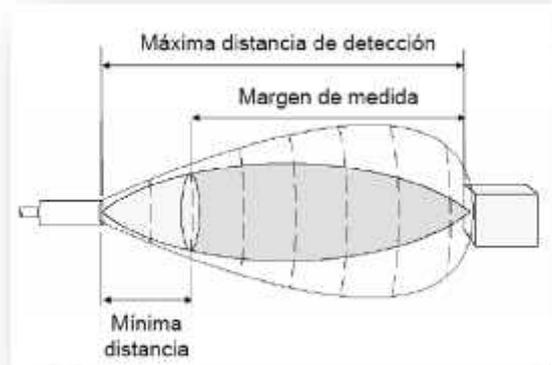


Figura 15: Ángulo de incertidumbre

Fuente: Alex Molina Espinoza

- Se debe tomar en cuenta la distancia mínima o zona ciega de un sensor ultrasónico, que es la distancia mínima a la cual el sensor nos indicará una detección de un objeto, aunque no haya ninguno presente. Esto puede ser reducido al utilizar sensores con transmisor/receptor separados.



**Figura 16:** Zona ciega del sensor ultrasónico

Fuente: Alex Molina Espinoza

- La temperatura ambiental puede repercutir en la toma de mediciones, puesto que las ondas de sonido se mueven a través de un material físico, en nuestro caso el aire, al aumentar o disminuir la temperatura, también varía la densidad del aire, haciendo que el sensor funcione fuera de los rangos establecidos y entregando medidas que pueden tener mayor rango de error que lo aceptado. (Nato, 2005)

## 1.8. MOTOR DE CORRIENTE CONTINUA

Un motor de corriente continua (conocido también como motor de corriente directa, motor CC o motor DC) es una máquina que transforma la energía eléctrica en mecánica, provocando un movimiento rotatorio, debido a la acción del campo magnético.

Una máquina de corriente continua (generador o motor) se compone principalmente de dos partes.

- **Estator:** es la parte que da soporte mecánico al dispositivo y contiene los devanados principales de la máquina, conocidos también con el nombre de

polos, que pueden ser de imanes permanentes o devanados con hilo de cobre sobre núcleo de hierro.

- **Rotor:** es la parte móvil de la máquina, generalmente de forma cilíndrica, también presenta devanado y con núcleo, alimentado con corriente directa mediante carbones o comúnmente llamados escobillas.



Figura 17: Rotor y estator de un motor DC

Fuente: [electricidad-viatger.blogspot.com/motores](http://electricidad-viatger.blogspot.com/motores)

Uno de los principales inconvenientes de estos dispositivos, es el mantenimiento que puede resultar caro y laborioso, debido principalmente al desgaste que sufren las escobillas al entrar en contacto con los devanados.

Algunas aplicaciones especiales de estos motores son los motores lineales, cuando ejercen tracción sobre un riel, o bien los motores de imanes permanentes. Los motores de corriente continua (CC) también se utilizan en la construcción de servomotores y motores paso a paso.

### 1.8.1. Variaciones en el diseño del motor

Los motores de corriente continua son elaborados con rotores bobinados, y con estatores bobinados o de imanes permanentes. Por otro lado existen muchos tipos de motores especiales, como por ejemplo los motores sin escobillas, los servomotores y los motores paso a paso, que se fabrican utilizando un motor de corriente continua como base.

### 1.8.1.1. Motores con estator bobinado

Si el estator es bobinado, existen varias configuraciones para conectar los dos bobinados de la máquina:

- Motor de DC en serie: el devanado del estator y el devanado del rotor se conectan en serie.
- Motor de DC en paralelo: el devanado del estator y el devanado del rotor se conectan en paralelo.
- Motor de DC compuesto: se utiliza una combinación de ambas configuraciones.

### 1.8.1.2. Motores de imán permanente

Los motores de imán permanente poseen varias ventajas de rendimiento frente a los motores síncronos de corriente continua de tipo excitado, por lo que han llegado a ser uno de los más usados en las aplicaciones de potencia fraccionaria. Son más pequeños, más ligeros, más eficaces y fiables que otras máquinas eléctricas alimentadas individualmente.



Figura 18: Motor de imán permanente

Fuente: <http://www.interempresas.net/Alimentaria/FeriaVirtual/Producto-Motores-de-CC-de-iman-permanentes-30278.html>

### 1.8.1.3. Motores sin escobillas

Los motores de corriente directa sin escobillas están diseñados para conmutar la tensión en sus devanados, con la ventaja de evitar el desgaste mecánico. Para este efecto utilizan controladores digitales y sensores de posición. Estos motores son frecuentemente utilizados en aplicaciones de baja potencia, por ejemplo en los ventiladores de computadoras.



**Figura 19:** Motor sin escobillas.

**Fuente:** [quadrino.com/guia-2/motores-brushless](http://quadrino.com/guia-2/motores-brushless)

### 1.8.2. Motor vibrador DC

Un motor vibrador de corriente continua es una variación del motor de estator bobinado, en el que se ha añadido un contrapeso al extremo superior del eje del rotor. La función del contrapeso es producir un movimiento armónico cuando el rotor gira a elevadas revoluciones (entre 5000 y 8000 RPM).

Dependiendo de la forma en la que el contrapeso esté ubicado sobre el eje del rotor, se podrán tener varias configuraciones, aprovechando la ley de la conservación del movimiento angular.

#### 1.8.2.1. Contrapeso en Semicilindro

Cuando la masa del contrapeso se dispone de manera que esté más cerca del eje del rotor (forma de semicilindro), el movimiento producido se manifestará en vibraciones rápidas, pero suaves. Generalmente esta disposición del contrapeso va

en conjunto con motores pequeños (0,5 cm de diámetro) para ser utilizados en dispositivos personales, como pueden ser celulares, beepers, relojes, GPS's, robots miniatura, entre otros.



**Figura 20:** Motor vibrador de semicilindro

**Fuente:** [quadrimo.com/guia-2/motores-vibradores](http://quadrimo.com/guia-2/motores-vibradores)

### 1.8.2.2. Contrapeso en Semidisco

Si la masa del contrapeso se dispone para que esté distribuida perpendicularmente al eje del rotor (forma de semidisco), el movimiento que se obtiene produce vibraciones en un intervalo más corto, pero son de mayor intensidad. Estos contrapesos son usados conjuntamente con motores medianos y grandes (cm de diámetro) y se colocan en dispositivos como controles de videojuegos, alarmas para personas con discapacidad auditiva, entre otros.



**Figura 21:** Motor vibrador en semidisco

**Fuente:** Alex Molina Espinoza

## 1.9. MICROCONTROLADORES

Un microcontrolador es un circuito integrado programable, que puede ejecutar las órdenes cargadas en su memoria. Está compuesto de varios bloques funcionales que cumplen una tarea específica. Un microcontrolador incluye en su interior las tres principales unidades funcionales de una computadora: unidad central de procesamiento (CPU), memoria y periféricos de entrada/salida. (Reyes, 2008)

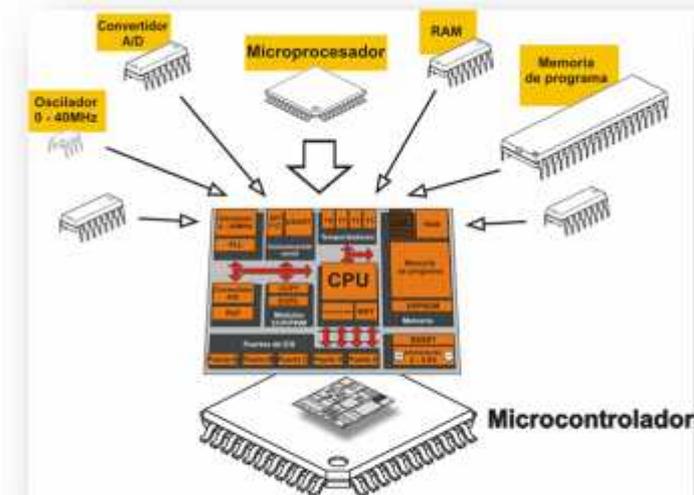


Figura 22: Partes que integran un microcontrolador

Fuente. mikroe.com/

Al momento de ser fabricado, el microcontrolador no contiene datos en la memoria ROM. Si se desea realizar o controlar algún proceso es necesario generar o crear y luego cargar en la EEPROM o equivalente, algún programa, el cual puede ser escrito en lenguaje ensamblador; sin embargo para que el programa pueda ser grabado en la memoria, debe ser codificado en sistema numérico hexadecimal que es finalmente el sistema que hace trabajar al microcontrolador cuando éste es alimentado con el voltaje adecuado y asociado a dispositivos analógicos y discretos para su funcionamiento.

### 1.9.1. Tarjeta Arduino

Arduino es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, que ha sido diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios. (Banzi, 2012)

El hardware consta de una placa con un microcontrolador Atmel AVR y puertos de entrada/salida. Los microcontroladores más usados son el Atmega168, Atmega328, Atmega1280, ATmega8 por su sencillez y bajo coste que permiten el desarrollo de múltiples y variados diseños. El software consiste en un entorno de desarrollo que implementa el lenguaje de programación Processing/Wiring y el cargador de arranque que es ejecutado en la placa. (ibídem).

#### 1.9.1.1. Lenguaje de programación

La plataforma Arduino se programa mediante el uso de un lenguaje propio basado en el lenguaje de programación de alto nivel *Processing*. Sin embargo, es posible utilizar otros lenguajes de programación y aplicaciones populares en Arduino, debido a que usa la transmisión serial de datos soportada por la mayoría de los lenguajes. Para los que no soportan el formato serie de forma nativa, es posible utilizar software intermediario que traduzca los mensajes enviados por ambas partes para permitir una comunicación fluida. Los principales lenguajes de programación compatibles con Arduino son:

- C
- C++ (mediante libSerial o en Windows)
- C#
- Java
- Matlab
- Python
- Visual Basic .NET

### 1.9.1.2. Entorno

El entorno de desarrollo de Arduino contiene un editor de texto para escribir los códigos, un área de mensajes, una consola de texto para el puerto serial, una barra de tareas con botones para las funciones más comunes y una serie de menús para interactuar con el usuario. Este se conecta al hardware del Arduino para comunicarse y cargar programas. (ibídem).



**Figura 23:** Entorno Arduino para Microsoft Windows

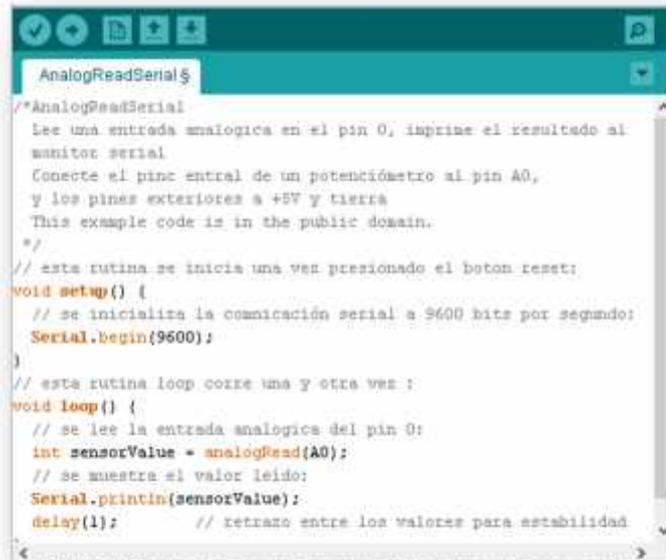
**Fuente:** Alex Molina Espinoza

### 1.9.1.3. Bocetos

El software escrito para usar en Arduino se denomina *Bocetos* (*Sketches en inglés*). Estos se escriben en el editor del texto del entorno Arduino. Los sketches son guardados con la extensión de archivo *.ino*. Esta extensión ofrece funciones como cortado/pegado y también para buscar y reemplazar texto.

El área de mensajes ofrece ayuda mientras guarda, exporta y también muestra errores. La consola muestra la salida de texto por medio del entorno del Arduino,

incluyendo mensajes de error completos e información adicional. La esquina inferior derecha de la pantalla muestra el puerto usado y la tarjeta conectada. Los botones de la barra de tarea permiten verificar y cargar programas, crear, abrir y guardar sketches, y abrir el monitor serial.



```

AnalogReadSerial5
/*AnalogReadSerial
Lee una entrada analogica en el pin 0, imprime el resultado al
monitor serial
Conecte el pino entral de un potenciómetro al pin A0,
y los pines exteriores a +5V y tierra
This example code is in the public domain.
*/
// esta rutina se inicia una vez presionado el boton reset;
void setup() {
  // se inicializa la comunicación serial a 9600 bits por segundo:
  Serial.begin(9600);
}
// esta rutina loop correz una y otra vez :
void loop() {
  // se lee la entrada analogica del pin 0:
  int sensorValue = analogRead(A0);
  // se muestra el valor leído:
  Serial.println(sensorValue);
  delay(1);      // retraso entre los valores para estabilidad
}

```

Figura 24: Boceto con un ejemplo de lectura analoga serial

Fuente: Alex Molina Espinoza

#### 1.9.1.4. Libro de bocetos

El entorno Arduino usa el concepto de *Libro de Bocetos (Sketchbook en inglés)*. Es un lugar estándar en donde se guardan los programas o bocetos. Los bocetos dentro de un sketchbook pueden ser abiertos mediante el menú *Archivo-Sketchbook* o por medio del botón *Abrir* en la barra de tareas. La primera vez que se inicie el software del entorno Arduino, este creará automáticamente un directorio para el sketchbook. Se puede ver o cambiar la localización del sketchbook desde el cuadro de diálogo de *Preferencias* en el menú *Archivo*.



Figura 25: Menú para acceder al sketchbook

Fuente: Alex Molina Espinoza

### 1.9.1.5. Librerías

Las librerías dentro de entorno Arduino proveen de una funcionalidad extra para usar en los bocetos, como por ejemplo, trabajar con hardware adicional o manipular datos externos. Para usar una librería en un boceto, se debe seleccionarla desde el menú *Sketch>Importar Librería*. Esto insertará una o más declaraciones del tipo *#include* al inicio del boceto y compilará la librería con el boceto. A causa de que las librerías son cargadas a la tarjeta conjuntamente con el boceto, estas incrementan la cantidad de espacio necesario que requiere el programa. Si un boceto no necesita de una librería, simplemente se debe borrar la declaración *#include* en el inicio del código.

La mayoría de las librerías se incluye con el software Arduino. Otras pueden ser descargadas desde una variedad de fuentes. Para añadir una librería Third party????, se debe crear un directorio llamado *Libraries* dentro del directorio del sketchbook, y luego se descomprime la librería.

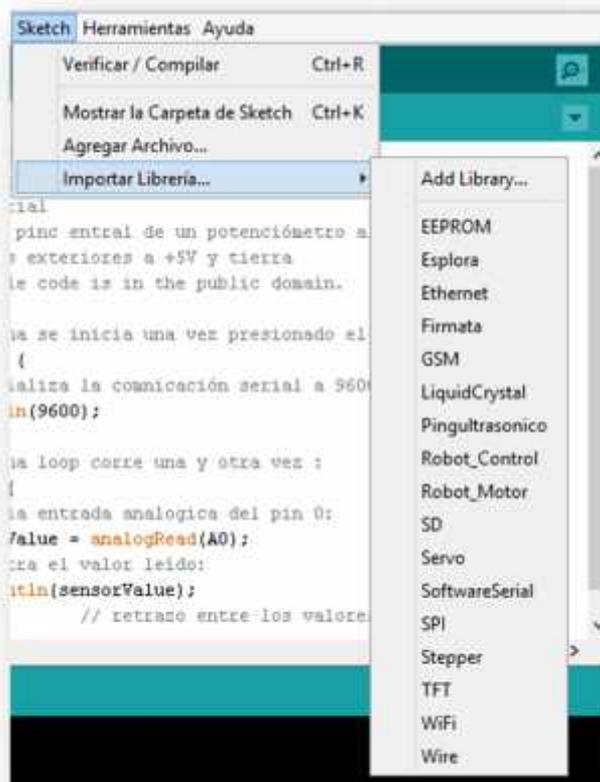


Figura 26: Menú de librerías disponibles en el software Arduino

Fuente: Alex Molina Espinoza

### 1.9.1.6. Monitor serial

El monitor serial es el encargado de mostrar los datos que están siendo enviados desde la tarjeta Arduino (sea mediante conexión USB o Serial). Se puede acceder al monitor serial mediante el botón en forma de lupa que se encuentra en la parte superior derecha de la ventana del programa. Para enviar datos a la tarjeta, se debe ingresar el texto deseado y luego presionar el botón *Send* o simplemente dar *Enter*. Se debe escoger la tasa de baudios desde el menú inferior, y esta debe coincidir con la tasa designada en la declaración *Serial.begin* del boceto.



Figura 27: Botón de acceso al monitor serial

Fuente: Alex Molina Espinoza

### 1.9.1.7. Hardware externo de terceros

El soporte para hardware adicional de terceros, puede ser añadido al directorio *hardware* del directorio del sketchbook. Las plataformas que allí se instalen pueden incluir definiciones de tarjetas (las cuales aparecerán en el directorio de tarjetas), librerías integrales, iniciadores de carga y definiciones de programación.

Para instalar una de estas opciones, se debe crear e directorio *hardware*, luego descomprimir la plataforma de terceros dentro de su propio sub-directorio. Debemos tener la precaución de no usar “arduino” como nombre de subdirectorio, puesto que se sobrescribirá el directorio integrado de la plataforma Arduino. Como ejemplo de hardware de terceros, tenemos los sensores.

### 1.9.1.8. Tarjetas tipo “Shield”

Las "Shields" son placas que pueden ser conectadas encima de la placa Arduino extendiendo sus capacidades. Las diferentes "shields" siguen la misma filosofía que el conjunto original: son fáciles de montar y su costo de producción es bajo.

Las shields suelen ser diseños bastante simples y en general de código abierto, el diseño suelen ser publicados libremente.



Figura 28: Shields para Arduino apilados

Fuente: arduino.cc

## 1.10. ALERTAS DE VOZ

### 1.10.1. Sintetizador de voz estándar “Loquendo”

El sintetizador de voz Loquendo (palabra que significa locución), es un software que provee reconocimiento, síntesis y verificación de voz y aplicaciones de verificaciones e identificaciones. Es usado en conjunto con equipos GPS para vehículos o portátiles, dispositivos para discapacitados, teléfonos inteligentes, libros electrónicos, cajeros automáticos con voz, videojuegos, artefactos domésticos controlados por voz, entre otros. Su producto comercial es uno de los más populares en la creación de videos, y es ampliamente usado gracias a su interfaz sencilla, y a la capacidad de poder reconocer y grabar en varios idiomas, además de contar con una vasta gama de voces, tanto masculinas como femeninas.

## **CAPÍTULO II**

### **2. METODOLOGÍA.**

#### **2.1. TIPO DE ESTUDIO.**

Descriptivo-Explicativo.- Con este trabajo de investigación se describió el problema desde el punto de vista de los principales beneficiarios, y a la vez se intentó explicar y comprobar los resultados una vez que el prototipo fue implementado.

#### **2.2. MÉTODOS, TÉCNICAS E INSTRUMENTOS**

##### **2.2.1. Métodos**

##### **2.2.1.1. Analítico.**

Se ocupó este tipo de metodología ya que es necesaria para entender en forma particular el funcionamiento de cada uno de los componentes necesarios para el desarrollo del prototipo, además del software que manejará dicho sistema, y la forma de interactuar de cada uno de los componentes entre sí para que el dispositivo desempeñe su función.

##### **2.2.1.2. Comparativo.**

Con este método se puede conseguir información muy apreciable para poder encontrar discrepancias u oposiciones entre los tiempos en que se demora una persona que usa el bastón de no vidente con otra que usó el prototipo objeto de estudio de la presente investigación.

## **2.2.2. Técnica**

### **2.2.2.1. Observación**

Esta técnica nos permite recolectar la mayor cantidad de información que nos ayude a resolver nuestro problema, y nos dé la pauta para desarrollar e implementar el sistema propuesto

### **2.2.2.2. Instrumentos**

Se utilizó libros, folletos, archivos, páginas web, datasheets, blogs, entre otros para la recopilación de la información que se refiere al diseño e implementación del prototipo.

## **2.3. POBLACIÓN Y MUESTRA**

La población estará dada con personas no videntes escogidas al azar.

### **2.3.1. Muestra**

<b>ESTRATOS</b>	<b>FRECUENCIAS</b>	<b>%</b>
<b>PERSONAS NO VIDENTES</b>	12	100
<b>TOTAL</b>	<b>12</b>	<b>100</b>

No se aplicó la fórmula de la muestra porque la población es manejable.

## 2.4. OPERACIONALIZACIÓN DE VARIABLES.

**Tabla 1:** Operacionalización de Variables

<b>Variable</b>	<b>Concepto</b>	<b>Indicadores</b>	<b>Técnicas e Instrumentos</b>
<b>Independiente</b> El Diseño e implementación de prototipo de gafas electrónicas	Un prototipo es una representación limitada de un producto, permite a las partes probarlo en situaciones reales o explorar su uso, creando así un proceso de diseño de iteración que genera calidad	Informática Conexión Sistemas Dispositivos Niveles	Programación en Arduino
<b>Dependiente:</b> Las personas no videntes puedan movilizarse surfeando obstáculos	Personas que perdieron su visión de una u otra forma. Movilidad a través de obstáculos sin tener contacto con estos	Rapidez Precisión Sensibilidad	Sensores Vibradores

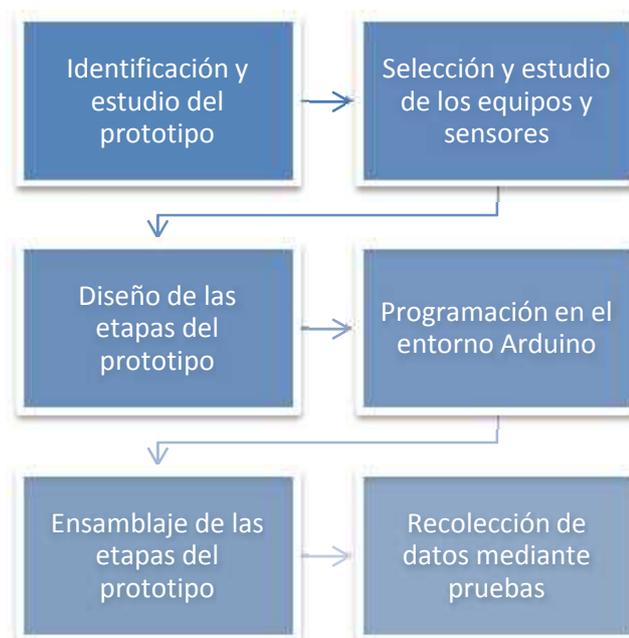
## 2.5. PROCEDIMIENTOS

Para el desarrollo de este dispositivo, se requirió seguir una serie de pasos y etapas, que en conjunto, cumplirían con el objetivo planteado. Cada etapa desempeña una función específica dentro del esquema global, y también sirve de apoyo para la siguiente etapa, haciendo del dispositivo implementado un sistema funcional.

La primera etapa del diseño se basó en identificar la mejor manera para que un dispositivo que se desee montar sobre unas gafas de uso común, pueda detectar obstáculos que se encuentren en una ruta y que provea de resultados dentro de márgenes de error y tiempos de respuesta aceptables.

Se eligió utilizar un sensor de distancia ultrasónico, puesto que son dispositivos que nos permitieron un fácil ensamblaje de las piezas que se incluyeron en el prototipo, además de tener un costo accesible para la adquisición de varios en caso de accidentes propios de las pruebas. Esto en contraste con los sensores infrarrojos que también se estudiaron como una opción plausible. Aunque su principal ventaja sobre un sensor ultrasónico era su precisión, al final quedaron descartados, puesto que en la actualidad no existen dispositivos de este tipo que puedan ser integrados a nuestros requerimientos, ya que por su tamaño, su implementación en un aparato portable no sería adecuado, más bien resultaría incómodo para el usuario. Por otro lado, su elevado costo y su muy limitado ángulo horizontal de detección, hicieron inviable que un sensor de este tipo pueda ser usado.

Después de conocer las características del sensor, se optó por utilizar la plataforma Arduino para la integración de todos los elementos del prototipo, por ser un software de distribución libre y de fácil acceso, además de que el entorno es amigable con el usuario y su programación se basa en lenguajes de fácil entendimiento.



**Figura 29:** Diagrama de Procedimientos

**Fuente:** Alex Molina Espinoza

En lo que respecta a la interacción del prototipo con el usuario, se decidió usar alertas de voz, conjuntamente con un sistema vibrador, con los cuales el dispositivo pueda anunciar la distancia a un objeto con la interacción del sentido del tacto al variar la intensidad de las vibraciones dependiendo de la proximidad entre el usuario y el obstáculo, previniendo al portador del dispositivo de un inminente choque

El montaje de estos componentes se lo dividió en dos partes:

La primera consistió en construir un *Shield* con el módulo reproductor de voz WTV020-SD para conectarla directamente a la tarjeta Arduino, resultando en un acoplamiento más compacto. Estos dos dispositivos se los colocó en una caja de plástico, que tiene unas dimensiones de 115 x 135 x 54 cm, en la cual están integrados el control de volumen para las alertas de voz, un jack de 3.5 mm que fue el adaptador para la salida de audio del sistema, un switch para el encendido y apagado de la tarjeta Arduino y los botones para la selección de rangos y el reset del sistema.

Para alimentar de energía a la tarjeta, que será la encargada de entregar el voltaje necesario a las otras partes, se decidió usar 4 baterías AA de 1.5 V de 2000 mAh conectadas en serie, que proporcionan un total de 6 V al adaptador de voltaje del Arduino. Los motores vibradores se dispusieron fuera de la caja, sobre un cinturón que se colocó el usuario a nivel de la cintura.

La segunda parte fue el acoplamiento de los 3 sensores ultrasónicos (PING))) sobre las gafas, para que la detección de los obstáculos pueda ser dirigida desde la altura de la cabeza de cada usuario. También se integraron 3 pulsadores para el control del sistema, selección del rango de distancia deseado y confirmación del rango.

Simultáneamente con la etapa de montaje de los dispositivos, se siguió con la programación en el entorno Arduino. Fue necesaria la integración de las librerías

del sensor de distancia ultrasónico PING))) y del reproductor de sonido WTV020-SD para poder manejar de manera sencilla el código, ya que las librerías contienen todas las líneas de comando necesarias que evitan una redundancia en el sketch.

Junto con la inclusión de las librerías, se diseñó el código para que el usuario pueda elegir entre tres rangos de distancia: el primer rango se denominaría Rango 1, y se ubicaría entre 4 cm a 50 cm, el segundo rango de distancia sería el Rango 2, y detectaría objetos entre 50 cm a 150 cm, y el Rango 3 detectaría obstáculos en distancias entre 151 cm a 250 cm.

Una vez diseñados los rangos de detección de objetos, se procedió a añadir las instrucciones para que la persona sea alertada de dicho objeto mediante la vibración producida por los motores DC. Dado que la eficiencia del trabajo de un motor de corriente alterna varía dependiendo del voltaje de alimentación, se usaron comandos para aprovechar las instrucciones PWM, las cuales hacen que la velocidad de rotación cambie en proporción con la distancia al obstáculo, por ende el usuario percibirá una mayor vibración cuanto más cerca se encuentre de un objeto detectado.

Compilado el código y cargado al Arduino, se integró las dos partes del sistema mediante cable multifilar, añadiendo también una conexión USB de la tarjeta a un computador para la recolección y visualización de datos mediante el puerto Serial del entorno de programación, estando listo el dispositivo para las pruebas de detección de varios tipos de obstáculos y la comprobación de la hipótesis del presente proyecto de investigación.

## **2.6. PROCESAMIENTO Y ANÁLISIS**

### **2.6.1. Identificación y estudio del prototipo**

Para diseñar el prototipo a implementarse, se realizó observaciones a la manera de utilización del bastón blanco en las personas no videntes, y se identificó que la principal desventaja al usar este implemento, es que la persona que lo usa mantiene

ocupada las manos, para mantener presente la ubicación en el entorno, además de manejar el bastón. Debido a esto, el diseño que se decidió implementar se enfocó en atender este vital inconveniente, ya que resulta beneficioso para el usuario disponer del uso de sus dos manos en todo momento.

De esta manera, el mejor diseño que se encontró para ubicar el dispositivo y sus sensores sobre el cuerpo, fue que los sensores se ubiquen sobre unas gafas que el usuario se pone en los ojos, y que el resto de elementos fueran armados en conjunto, y colocados sobre un cinturón ajustable que la persona pueda colocarse en la cintura.

Los sensores que se ubicaron sobre las gafas tuvieron mucha más facilidad para detectar un obstáculo en frente de una persona, ya que aumenta el lóbulo de sensado y se puede evitar el solapamiento de la señal con los otros sensores. Los otros elementos que se situaron sobre la cintura, permiten un simple manejo de las funciones por parte del usuario, ya que puede acceder fácilmente a los pulsadores que manejan las funciones de encendido, cambio del rango de los sensores, y el reset del sistema.

### 2.6.2. Selección y estudio de los equipos y sensores

Los sensores y dispositivos a usarse en la construcción del prototipo de gafas electrónicas se detallan en la Tabla No. 2.

**Tabla 2:** Equipos y materiales que se utilizaron

Cantidad	Dispositivo/sensor
1	Tarjeta Arduino MEGA ATmega 2560
3	Sensor ultrasónico de distancia PING))) 28015
1	Módulo reproductor de sonido WTV020-SD
1	Motor vibrador con contrapeso en semidisco
2	Motor vibrador con contrapeso en semicilindro

1	Regulador de volumen
1	Adaptador a jack de 3.5 mm
4	Baterías AA de 1950 mAH

Fuente: Alex Molina Espinoza

Mediante los sensores ultrasónicos se recibió la distancia del dispositivo al obstáculo; estos datos fueron recolectados e interpretados mediante la programación del Arduino dependiendo del rango de distancia seleccionado y como respuesta el sistema entregó una alerta vibratoria.

### 2.6.2.1. Placa Arduino MEGA 2560

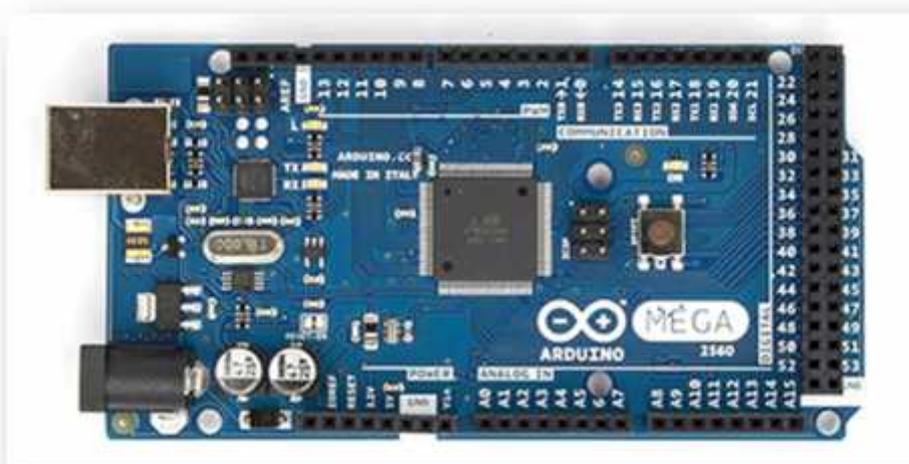


Figura 30: Placa Arduino MEGA con procesador ATmega2560

Fuente: arduino.cc

Arduino MEGA es una tarjeta basada en el micro controlador ATmega2560. Contiene todo lo necesario para dar soporte al micro controlador, simplemente necesita de una conexión a un computador mediante el cable USB o a una fuente de poder mediante un adaptador AC a DC.

#### Características:

A continuación se detallan algunas características de la placa.

**Tabla 3:** Características de la tarjeta Arduino MEGA

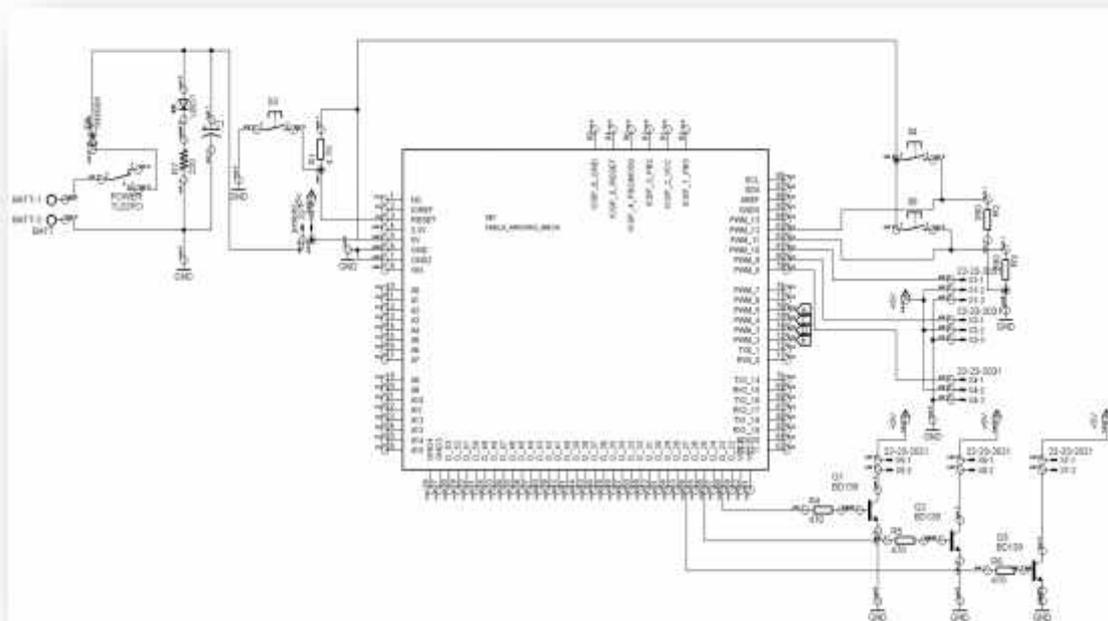
<b>Microcontrolador</b>	ATmega2560
<b>Voltaje de operación</b>	5V
<b>Voltaje de entrada ( recomendado)</b>	7-12V
<b>Voltaje de entrada (límites)</b>	6-20V
<b>Pins de E/S Digital</b>	54 (de los cuales 15 son salidas PWM)
<b>Pins de entrada analógica</b>	16
<b>Corriente DC por cada pin de E/S</b>	40 mA
<b>Corriente DC en el pin de 3.3 V</b>	50 mA
<b>Memoria Flash</b>	256 kB de los cuales 8 kB son usadas por el bootloader
<b>SRAM</b>	8 kB (ATmega2560)
<b>EEPROM</b>	4 kB (ATmega2560)
<b>Velocidad del reloj</b>	16 MHz

Fuente: Alex Molina Espinoza

El software de programación consiste en un entorno de desarrollo que implementa el lenguaje de programación Processing /Wiring y el cargador de arranque que es ejecutado en la placa. El entorno de programación no es distinto de los lenguajes ensambladores utilizados habitualmente, por lo que comparten la mayoría de comandos, cambiando solamente la sintaxis de las instrucciones

### **Características eléctricas**

En la siguiente figura se detallan los pines de conexión que se usó para conectar a la tarjeta Arduino con los otros dispositivos



**Figura 31:** Diagrama de conexión de la tarjeta Arduino

**Fuente:** Alex Molina Espinoza

En nuestro prototipo, la tarjeta Arduino constituye el cerebro del sistema, dado que es la encargada de recibir y analizar los datos que provienen del sensor, así como también decidir la alerta que será indicada al usuario.

### 2.6.2.2. Sensor ultrasónico de distancia PING))) 28015

El sensor ultrasónico de distancia que se usó es el PING))) 28015 de Parallax. Este sensor provee medición de distancias precisas sin necesidad de contacto, desde 2 cm hasta 4 metros. La conexión al microcontrolador, en nuestro caso la tarjeta Arduino, solo requirió un pin que actuó como Entrada/Salida, haciendo fácil su instalación en un circuito de control.



**Figura 32:** Sensor de distancia ultrasónico PING)))

**Fuente:** Parallax datasheet

El sensor PING))) trabaja transmitiendo una ráfaga ultrasónica y proveyendo un pulso de salida que corresponde al tiempo que requiere para que el eco de la ráfaga regrese al sensor. Utiliza el método de detección directa, mediante la configuración de doble sensor empotrado. No requiere un circuito para calibración externa.

A continuación se detalla algunas de las características del sensor:

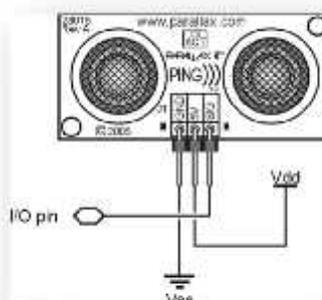
**Tabla 4:** Características del sensor ultrasónico PING)))

<b>Rango de detección</b>	de 2 cm a 3,3 m
<b>Indicador de actividad</b>	Mediante un LED
<b>Pulso TTL</b>	Bidireccional en un mismo pin; TTL de 5V o Cmos de 3,3V
<b>Pulso de entrada del disparador</b>	Pulso TTL positivo, mínimo de 2 $\mu$ s
<b>Pulso de Eco</b>	Pulso TTL positivo, entre 115 $\mu$ s a 18,5 ms
<b>Voltaje de funcionamiento</b>	+ 5 VDC
<b>Corriente de alimentación</b>	30 – 35 mA
<b>Temperatura de operación</b>	0 – 70 °C

Fuente: Alex Molina Espinoza

### Características eléctricas:

Para la conexión del sensor al microcontrolador, se debe tomar en cuenta sus pines de conexión.



**Figura 33:** Diagrama eléctrico del sensor PING)))

Fuente: Parallax datasheet

Los pines de la figura corresponden a la siguiente configuración:

**Tabla 5:** Pines de conexión del sensor ultrasónico PING)))

PIN	SEÑAL
5V/Vdd	Señal de alimentación de 5 V
GND	Señal común
SIG	Entrada/Salida de la señal del sensor

**Fuente:** Alex Molina Espinoza

En el prototipo se utilizó 3 sensores ultrasónicos de distancia, que fueron colocados sobre las gafas, para que exista un mayor ángulo de detección de los objetos.

### 2.6.2.3. Módulo reproductor de sonido WTV020-SD



**Figura 34:** Reproductor de sonido WTV020-SD

**Fuente:** Datasheet

El módulo WTV020SD es un hardware de tercero que puede ser acoplado al Arduino. Se trata de un pequeño chip que permite reproducir sonidos almacenados en una tarjeta micro SD. Solo es necesario cargar sonidos en la tarjeta de memoria, conectarla al Arduino y leer el pin de reproducción; o a su vez conectando un pulsador que funcione como botón “PLAY”.

Este dispositivo tiene las siguientes características:

**Tabla 6:** Características del módulo WTV020SD

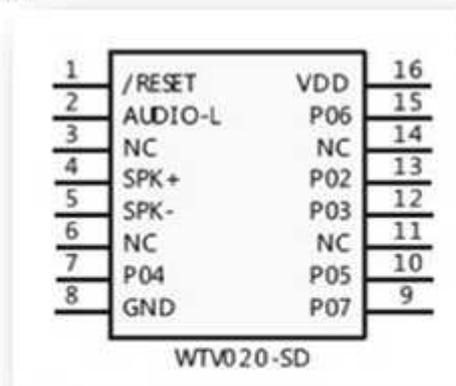
<b>Formato de archivo</b>	4-bit ADPCM (.ad4) de 6Khz a 36Khz
<b>Modos de operación</b>	Serial Mode: Interfaz (Data y Clock) para cualquier tipo de micro controlador  Key Mode: Funcionamiento autónomo sin necesidad de controlador externo.
<b>Zócalo de tarjeta</b>	Ranura para tarjetas microSD hasta 1 GB
<b>Salidas PWM</b>	2 para conectar directamente un altavoz (8 Ohm/0.5W).
<b>Voltaje de funcionamiento</b>	2,5 a 3,6 V
<b>Corriente de alimentación</b>	80 mA
<b>Corriente en reposo</b>	8 $\mu$ A

Fuente: Alex Molina Espinoza

En modo "serial" conectado al Arduino, se necesitan solo dos líneas de datos (DATA y CLOCK). Todas las operaciones como PLAY; PAUSE, STOP y VOLUME están disponibles desde el host mediante un sencillo protocolo serie. No soporta directamente archivos WAV o MP3 sino que utiliza otro formato necesario para su funcionamiento (4-bit ADPCM @ 6-32kHz). Por lo tanto, es necesario convertir los archivos antes de reproducirlos.

### Características eléctricas:

En la siguiente figura se puede apreciar el modo de conexión para la comunicación del sensor



**Figura 35:** Diagrama eléctrico del reproductor WTV020-SD

Fuente: Datasheet

Como se indicó en la figura No. 30, la conexión de los pines es de la siguiente manera:

**Tabla 7:** Pines de conexión del módulo WTV020SD

<b>PIN</b>	<b>SEÑAL</b>
Vdd	Señal de alimentación
P06	Señal de ocupado
P05	Señal de datos
P04	Señal de reloj
AUDIO-L	Señal de salida de audio
RESET	Señal para reset del sistema
GND	Señal común

**Fuente:** Alex Molina Espinoza

El reproductor de voz se colocó sobre la tarjeta Arduino mediante una tarjeta shield previamente fabricado, para reducir las dimensiones del dispositivo. Usando el diagrama de conexión, la nueva tarjeta “shield” se diseñó como se muestra en la siguiente figura:



**Figura 36:** Tarjeta shield para colocar sobre el Arduino

**Fuente:** Alex Molina Espinoza

#### 2.6.2.4. Micro motor vibrador DC de Semidisco No. 280



**Figura 37:** Micro motor vibrador No. 280

**Fuente:** Alex Molina Espinoza

El micro motor vibrador No. 280 es un vibrador diseñado para dispositivos portátiles, de juegos y de alertas por tacto, entrega un movimiento armónico relativo gracias al torque de 104g.cm, y puede ser fácilmente acoplado a microcontroladores.

Algunas de las características de este motor son:

**Tabla 8:** Características del motor No. 280

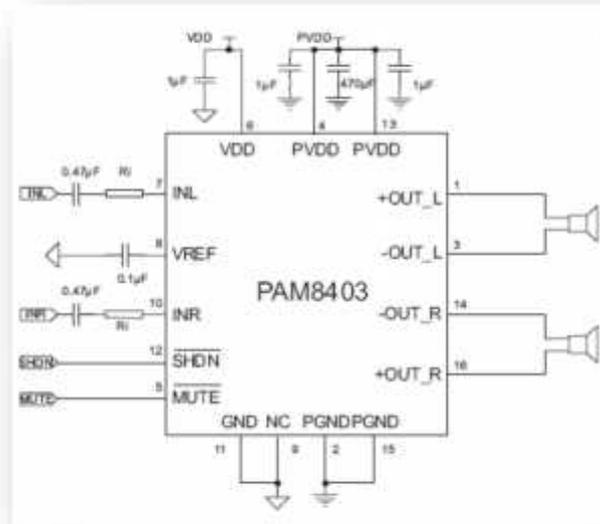
<b>Velocidad (RPM)</b>	3800 rev/min
<b>Voltaje de alimentación</b>	5V
<b>Corriente de uso</b>	0,22 A
<b>Torque</b>	104 g.cm
<b>Radio del disco</b>	6,5 mm

**Fuente:** Alex Molina Espinoza

#### **Características eléctricas**

En la siguiente figura se detalla el diagrama y conexión del micro motor.





**Figura 40:** Diagrama eléctrico del controlador de voltaje PAM8403

**Fuente:** Datasheet PAM8403

Los pines de la figura corresponden a la siguiente configuración:

**Tabla 9:** Pines de conexión del controlador de volumen PAM8403

PIN	SEÑAL
5V/Vdd	Señal de alimentación de 5 V
GND	Señal común
SIG	Entrada/Salida de la señal del sensor

**Fuente:** Alex Molina Espinoza

### 2.6.3. Test de comunicación entre el Arduino y los sensores

#### 2.6.3.1. Requisitos de software para las pruebas

Para la comunicación entre la tarjeta Arduino y los dispositivos a usarse, se necesitó la instalación del entorno y las librerías que deben estar en la computadora para las pruebas de comunicación y el desarrollo del programa a cargarse

Los programas y archivos necesarios son:

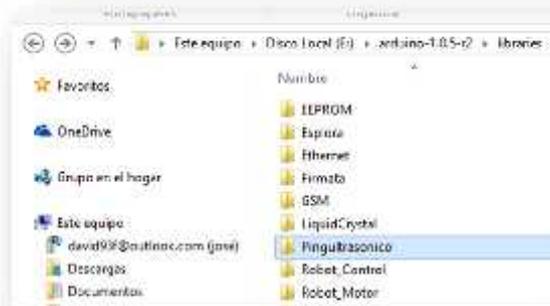
- Entorno Arduino 1.6.3
- Librería PING para los sensores de distancia ultrasónicos

- Librería WTV020sd16p para el reproductor de sonido
- Librería PMW para el control de los micro motores
- Conversor de formato de sonido UsbRecorder Versión 1.3
- Sintetizador de voz Loquendo

Hay que copiar correctamente en el directorio del entorno las librerías necesarias, para que al armar el código, el programa pueda ser compilado y no presente ningún aviso de error de librerías no declaradas.

### 2.6.3.2. Conexión del Arduino con el sensor ultrasónico PING)))

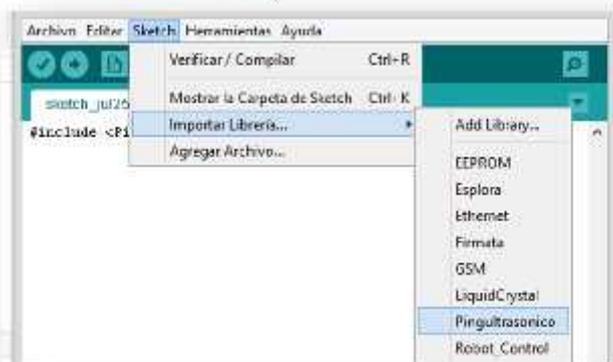
1. Se cargó la librería correspondiente al sensor PING))) en el directorio raíz del entorno Arduino



**Figura 41:** Ubicación de la librería del sensor PING)))

**Fuente:** Alex Molina Espinoza

2. Se inició el entorno Arduino y se verificó que reconozca la librería, mediante el menú *Sketch > Importar librería*



**Figura 42:** Lista de librerías en el entorno Arduino

**Fuente:** Alex Molina Espinoza

3. Se implementó el código de ejemplo de uso del sensor, para poder obtener una prueba de comunicación. El código tuvo la siguiente estructura:

```
// Pin donde se encuentra la salida del sensor:
const int pingPin = 7;
void setup() {
  Serial.begin(9600); // inicializar la comunicación serial
}

void loop()
{
  // establece las variables para la duración del impulso,
  // y los resultados de la distancia en pulgadas y centímetros
  long duration, pulg, cm;

  pinMode(pingPin, OUTPUT);
  digitalWrite(pingPin, LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin, HIGH); // el sensor se activa por un pulso
en ALTO
  delayMicroseconds(5); // se necesita un pulso BAJO para
asegurar
  digitalWrite(pingPin, LOW); // un pulso ALTO limpio
  // el mismo ping se usa para leer la señal del sensor PING))
  // un pulso en ALTO cuya duración es el tiempo (en ms) desde el
envío del
  // ping hasta la recepción del eco de un objeto

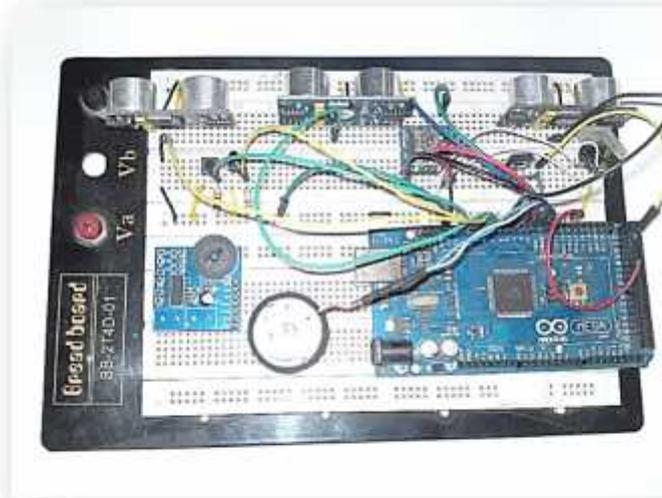
  pinMode(pingPin, INPUT);
  duration = pulseIn(pingPin, HIGH);

  // conversión de tiempo a distancia
  inches = microsecondsToInches(duration);
  cm = microsecondsToCentimeters(duration);
  // impresión de la distancia en el puerto Serial
  Serial.print(pulg);
  Serial.print("pulg, ");
  Serial.print(cm);
  Serial.print("cm");
  Serial.println();

  delay(100);
}
```

4. Se compiló el código para comprobar que no existan errores en la sintaxis del programa.

5. Se realizó la conexión entre el sensor PING))) y la tarjeta Arduino como se detalla en la siguiente figura.



**Figura 43:** Conexión entre la tarjeta Arduino y los sensores PING)))

**Fuente:** Alex Molina Espinoza

6. Se conectó el Arduino al PC mediante el cable USB y esperamos que el entorno reconozca el puerto COM, reseteamos el Arduino y cargamos el programa a la tarjeta.
7. Se verificó la comunicación entre el Arduino y el sensor PING))) mediante la consola serial, la cual debe mostrar la distancia en centímetros que existe entre el obstáculo y el sensor.

### **2.6.3.3. Grabación de las alertas de voz con el sintetizador de voz “Loquendo”**

Para la grabación de las alertas de voz, fue necesario instalar el programa Loquendo con sus diferentes motores de distribución, complementos y librerías SDK.

1. Se ejecutó el programa de instalación y se aseguró que se instalen correctamente los siguientes elementos: *Loquendo TTS 7 Engine Full Distribution*, *Loquendo TTS 7 SDK Distribution*, *Loquendo TTS 7 Spanish*

*Distribution* y *Compliments*. Además se pudo seleccionar una voz de la parte derecha del menú de instalación.

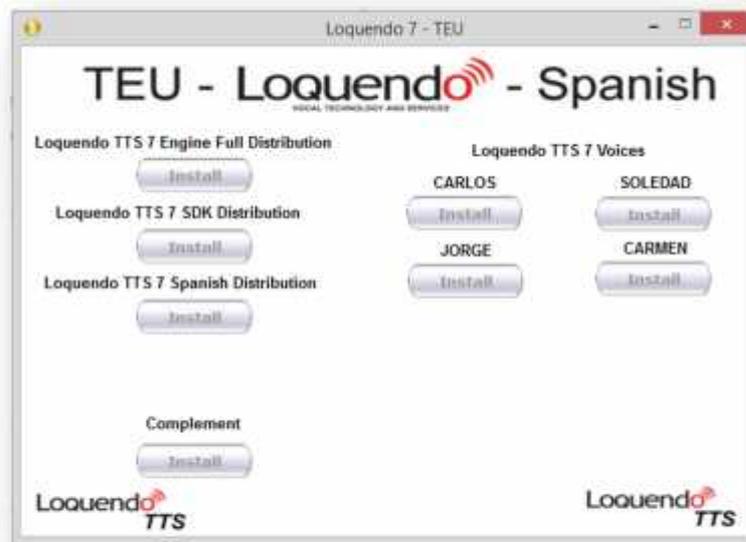


Figura 44: Programa de instalación del sintetizador Loquendo

Fuente: Alex Molina Espinoza

2. Se abrió el entorno del programa donde había tres secciones: en la primera la barra de comandos, donde se tuvo las opciones de reproducción y grabación del sonido; en la mitad de la ventana hay el recuadro para poder escribir el texto que se sintetizó, en la parte inferior estuvieron las opciones de modificación de voz en la parte izquierda, con las cuales puede cambiar la velocidad, tono, volumen, balance, reverberación y frecuencia; y en la parte derecha las opciones para seleccionar la voz, que puede ser masculina o femenina, además del idioma que se va a reproducir.

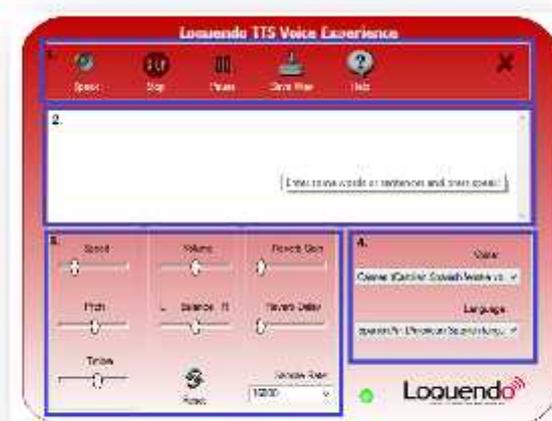


Figura 45: Secciones del entorno del programa Loquendo

Fuente: Alex Molina Espinoza

3. Se escribió el texto deseado en el recuadro y modificando la voz para que se reproduzca de una forma pausada, además se seleccionó la frecuencia en 16kHz que fue usada por el reproductor.
4. Se guardó el archivo en formato *.wav*, cambiando la salida a *Stereo* y seleccionando el directorio de salida mediante el botón *Browse*

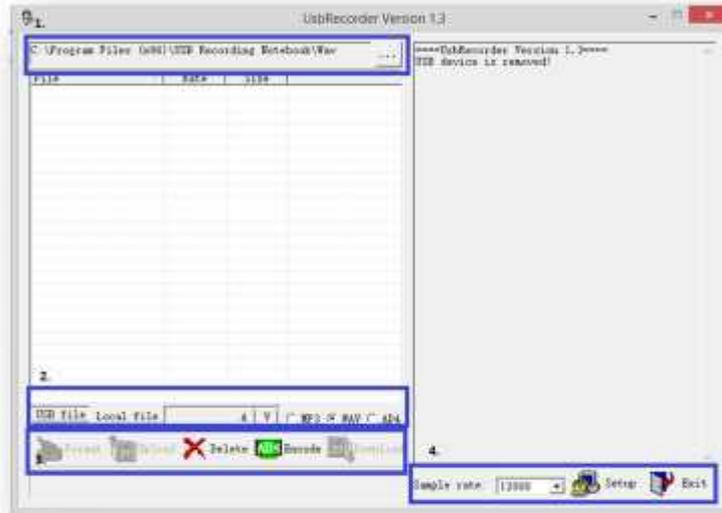


**Figura 46:** Grabación del texto sintetizado

**Fuente:** Alex Molina Espinoza

#### **2.6.3.4. Conversión del formato de los archivos de sonido**

1. Se instaló el programa *UsbRecorder* y se lo ejecutó.
2. En el entorno del programa existen 5 secciones: la primera para buscar el archivo que se va a convertir; en la segunda se hallan las pestañas para seleccionar si es un archivo que se están en un puerto *Usb*, o si es un archivo del disco duro local, además de los botones de selección para el tipo de archivo al que se desea convertir; en la tercera sección la barra de opciones y en la cuarta sección están los botones para seleccionar la frecuencia de muestreo y de configuración del programa.



**Figura 47:** Secciones del entorno del programa codificador

**Fuente:** Alex Molina Espinoza

3. Se seleccionó la pestaña *Local file* y la extensión del archivo que se busca, en este caso el archivo que se sintetizó usando *Loquendo* y se guarda con la extensión *.wav*
4. Clic en el botón *Encode*.
5. Se abrió la carpeta en donde se encontraba el archivo original, y se obtuvo otro archivo convertido a formato *.ad4*, se procedió a guardarlo en una tarjeta microSD con una capacidad no mayor a 2GB

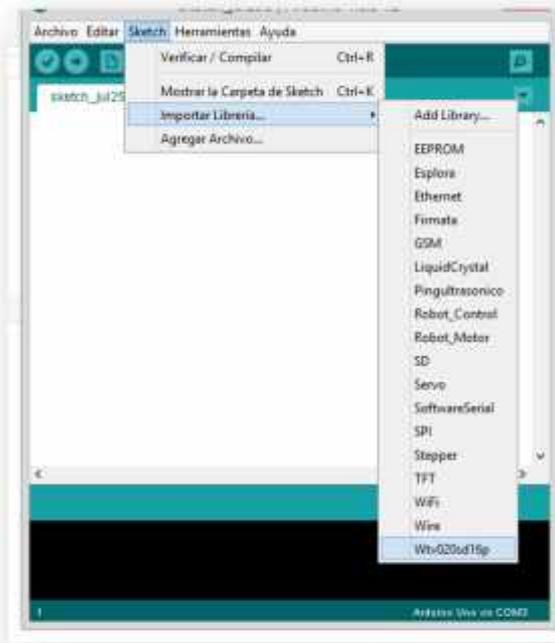


**Figura 48:** Archivo original y archivo convertido a formato *.ad4*

**Fuente:** Alex Molina Espinoza

### 2.6.3.5. Conexión del Arduino con el sensor reproductor de sonido WTV020

1. Se copió la librería del reproductor al directorio raíz del entorno Arduino, y se verificó que reconozca la librería mediante el menú *Sketch>Importar librería*.



**Figura 49:** Librería del reproductor WTV020 resaltada en la lista

**Fuente:** Alex Molina Espinoza

2. Se realizó la implementación de un código de ejemplo para probar la comunicación entre el reproductor y el Arduino. El archivo fue grabado en la tarjeta micro SD, porque el programa llamará al archivo por el nombre grabado. El programa tuvo la siguiente estructura:

```
#include <Wtv020sd16p.h> // incluye la librería del reproductor

int resetPin = 2; // Pin para reset.
int clockPin = 3; // Pin para señal de reloj.
int dataPin = 4; // Pin para señal de datos.
int busyPin = 5; // Pin para la señal ocupado.

// Variables para uso del reproductor
Wtv020sd16p wtv020sd16p(resetPin,clockPin,dataPin,busyPin);

void setup() { //Inicializa el módulo.
  wtv020sd16p.reset();
}

void loop() {
  //Reproduce un archivo de audio. Se usa el pin de ocupado.
  wtv020sd16p.playVoice(0);
  wtv020sd16p.asyncPlayVoice(1);
  //Reproduce el archivo de audio 1 por 2 segundos.
  delay(5000);
}
```

```
//Pausa el archivo de audio 1 por 2 segundos.  
wtv020sd16p.pauseVoice();  
delay(5000);  
//Resume la reproducción del archivo de audio por 2 segundos.  
wtv020sd16p.pauseVoice();  
delay(5000);  
//Detiene el archivo de audio reproduciéndose.  
wtv020sd16p.stopVoice();
```

3. Se conectó el Arduino y el reproductor de sonido, añadiendo un parlante para la salida de sonido.
4. Se compiló el sketch y se cargó al Arduino el programa mediante el cable USB. Se comprobó que la comunicación sea efectiva mediante la pulsación del botón “play”.

#### 2.6.4. Diseño de las etapas del prototipo

El dispositivo se diseñó para armarse en dos etapas: la primera consta de la tarjeta Arduino MEGA, el Shield con el reproductor de sonido WTV020, el regulador del sonido, los botones de selección de rango, el adaptador del jack de 3,5mm para la salida de audio, las 4 baterías AAA, y un motor vibrador, todos estos elementos se integraron en una caja de plástico fabricada a la medida.



**Figura 50:** Tarjeta Arduino MEGA con la placa shield, reproductor de sonido y reglador de volumen

**Fuente:** Alex Molina Espinoza

Fuera de esta caja se colocó 2 micro-motores vibradores, uno al frente y otro al lado izquierdo. En conjunto con la caja, estos elementos fueron situados sobre un cinturón ajustable, que el usuario se coloca a nivel de la cintura.

La segunda etapa consta de los sensores ultrasónicos, que se ubicaron en la parte superior de las gafas, uno al frente, uno a la izquierda y otro a la derecha. Para conectarlas a la tarjeta Arduino usaremos cable UTP multifilar, que fue ubicado dentro de una banda elástica.



**Figura 51:** Ubicación de los sensores ultrasónicos sobre las gafas

**Fuente:** Alex Molina Espinoza

### 2.6.5. Programación en el Entorno Arduino

Una vez realizadas las pruebas de comunicación entre el Arduino y los sensores, se procedió a escribir el código que tiene como objetivo controlar las dos etapas del dispositivo. Primero se incluyeron las librerías de los sensores en el *sketch*

```
#include <Wtv020sd16p.h> // Librería reproductor de sonido  
#include <Ping.h> // Librería sensor ultrasónico
```

Eso permitió que el programa cargue a la memoria de la tarjeta Arduino los comandos predeterminados, tanto para el sensor ultrasónico, como para el reproductor de sonido y el regulador de volumen.

Se declaró las variables para la entrada de datos de los sensores, usando los pines bidireccionales.

```

Ping ping = Ping(8);    // sensor del lado derecho.
Ping ping = Ping(9);    // Sensor del frente.
Ping ping = Ping(10);   // Sensor del lado izquierdo.

```

Se especificaron los pines de entradas y salidas que se utilizó en la conexión entre el Arduino y el módulo de sonido, mediante la declaración de variables de entrada y salida, usando la instrucción `int`, así como también las salidas para los 3 motores vibradores que corresponden con cada sensor.

```

int ResetPin = 5;    // El número de pin del de reset.
int clockPin = 4;    // El número de pin del reloj
int dataPin = 3;     // El número de pin de datos
int busyPin = 2;     // El número de pin del pasador ocupado
int m1=23,m2=25,m3=27; //Pines para los 3 motores

```

Se incluyó además con las variables del reproductor de sonido en la librería, y también la declaración de las variables para los botones de selección de rango con las siguientes líneas de comando:

```

Wtv020sd16p wtv020sd16p ( ResetPin, clockPin, dataPin, busyPin);
int pinselect=11,pinnext=12,r=0; //variables de los pulsadores

```

En forma de comentario se adicionó una lista en orden de los sonidos que fueron grabados en la microSD, para que puedan reproducirse al usar la instrucción `playVoice` que requiere la posición del archivo en la tarjeta de memoria.

```

// sonido 0: seleccione rango
// sonido 1: rango 1
// sonido 2: rango 2
// sonido 3: rango 3
// sonido 4: confirma 1
// sonido 5: confirma 2
// sonido 6: confirma 3

```

Luego de declaradas todas las variables e incluirlas en las librerías, se continuó con la escritura del código, que estuvo dividida en dos subrutinas; la primera es la que obliga al sistema a iniciar; en esta se escribieron las instrucciones para que el dispositivo pida seleccionar el rango al que se desea realizar la medición de la distancia, mientras que en la segunda subrutina, una vez confirmado el rango, se configuraron las distancias a las cuales los sensores envían las alertas mediante los motores vibradores.

Para inicializar la primera subrutina se incluyó la instrucción `void setup` que permite correr las líneas de comando una sola vez, en este caso para poder confirmar el rango deseado, mientras que la segunda subrutina se utilizó la instrucción `void loop`, debido a que la detección que realizan los sensores se da en una forma continua, y necesita una rutina que se repita hasta que se resetee el sistema. De manera adicional se agregó la instrucción `Serial.println` en forma de comentario para poder visualizar los datos si se desea conectar la tarjeta Arduino a la consola Serial mediante un puerto USB.

En la primera subrutina se utilizaron las siguientes líneas de comandos:

```
void setup()
{
  Serial.begin(9600);           //Velocidad del puerto serial
  wtv020sd16p.reset ();        //Instrucción para resetear de
reproductor
  pinMode(pinselect, INPUT); //Ingreso del valor hacia la variable
  pinMode(pinnext, INPUT); // Ingreso del valor hacia la variable
  configurar();                //Ir a la subrutina
}
void setup()
{
  Serial.begin(9600);           //Velocidad del puerto serial
  wtv020sd16p.reset ();
  pinMode(pinselect, INPUT);
  pinMode(pinnext, INPUT);
```

```

pinMode(m1, OUTPUT);
pinMode(m2, OUTPUT);
pinMode(m3, OUTPUT);
configurar();           //Ir a la subrutina
}

```

En esta subrutina, se ingresó la instrucción `Serial.begin`, que indica la velocidad de datos en la que la tarjeta trabaja, se incluyó también el comando de reset del reproductor de voz, para que la reproducción inicie siempre desde el primer sonido grabado en la tarjeta microSD, luego de eso se determinó que tipo de datos estarán recibiendo las variables iniciales; `pinselect` es la variable que se utilizó para guardar el rango seleccionado, y se le indicó como Input en la línea de comandos, `pinnext` es la variable que guarda el incremento para el siguiente rango, y de igual manera se inició como Input, mientras que `m1`, `m2` y `m3` son las variables de los 3 motores vibradores, y se inicializaron como Outputs (salidas). En la línea final de la subrutina se indicó que una vez inicializado el sistema, este se redirige hacia la subrutina `configurar`, la misma que contiene las siguientes líneas de comando:

```

void configurar()           //Subrutina Configurar
{
  Serial.println("SELECCIONE RANGO");
  delay(2000);
  wtv020sd16p.playVoice(0);
  delay(2000);
}

```

Una vez inicializado el sistema, se reproduce el sonido guardado en la posición 0 de la tarjeta microSD, el cual pide al usuario seleccionar el rango en el que quiere usar el dispositivo.

```

while (digitalRead(pinselect) == LOW)
{
  delay(100);
  if (digitalRead(pinnext) == HIGH)
  {
    delay(100);
    r=r+1;
    if (r > 3)

```

```

    {
      r=1;
    }
    if (r==1)
    {
      wtv020sd16p.playVoice (1);
      delay(2000);
      //Serial.println("RANGO 1");
    }
    if (r==2)
    {
      wtv020sd16p.playVoice (2);
      delay(2000);
      //Serial.println("RANGO 2");
    }
    if (r==3)
    {
      wtv020sd16p.playVoice (3);
      delay(2000);
      //Serial.println("RANGO 3");
    }
  }
}

```

Para realizar el cambio de rango, cada vez que el usuario presione el pulsador 1 (lectura del pin en HIGH), la instrucción de repetición **while** incrementa en 1 el rango a seleccionarse, y se reproduce el sonido correspondiente al rango mediante la instrucción **playVoice**.

```

    if (r==1)
    {
      wtv020sd16p.playVoice (4);
      //Serial.println("SELECCIONO EL RANGO 1");
      delay(2000);
    }
    if (r==2)
    {
      wtv020sd16p.playVoice (5);
      //Serial.println("SELECCIONO EL RANGO 2");
      delay(2000);
    }
    if (r==3)
    {
      wtv020sd16p.playVoice (6);
      //Serial.println("SELECCIONO EL RANGO 3");
      delay(2000);
    }
  }
}

```

Para realizar la selección del rango, se presiona el pulsador 2 luego de escuchar el rango, y se reproduce una confirmación de la selección del rango.

En la segunda subrutina se escribió el código encargado de interpretar los datos provenientes de los 3 sensores, dependiendo del rango seleccionado, además de enviar la señal a los vibradores una vez que se detecte un obstáculo. La línea de comandos usada es la siguiente:

```
void loop()
{
  ping1.fire();
  ping2.fire();
  ping3.fire();
  float d1 = ping1.centimeters();
  float d2 = ping2.centimeters();
  float d3 = ping3.centimeters();
  //Serial.print(d1);
  //Serial.println(" cm");
  //Serial.print(d2);
  //Serial.println(" cm");
  //Serial.print(d3);
  //Serial.println(" cm");
  delay(100);
}
```

En esta subrutina se inicializaron los sensores mediante la instrucción `ping1.fire()` y los datos obtenidos de los sensores se guardaron en las variables `d1`, `d2` y `d3`, cada uno por separado, añadiendo la instrucción `ping1.centimeters()` que guarda directamente el dato medido en centímetros. Además se añadió en forma de comentario la instrucción `Serial.print` para poder visualizar los datos obtenidos al conectar la tarjeta a un puerto USB.

```
if (d1<=50&&(r==1)) //rango 1
{
  digitalWrite(m1, HIGH);
  delay(2000);
  digitalWrite(m1, LOW);
  delay(500);
  //Serial.println("OBSTACULO AL FRENTE");
}
```

Mediante la instrucción se indicó al sensor la distancia del rango seleccionado, para el caso del **rango 1** se estableció la distancia entre 5 y 50 cm, para el **rango 2**

entre 5 y 150 cm, y para el **rango 3** entre 5 y 250 cm. Esta instrucción se repitió para cada uno de los rangos para cada sensor, y las alertas para los motores vibradores se asignaron mediante el comando `digitalWrite`, siendo `m1` la variable para el motor de la derecha, `m2` para el motor del frente y `m3` para el motor de la izquierda.

```

if ((d1<=150)&&(r==2))           //rango 2
{
  digitalWrite(m1, HIGH);
  delay(1000);
  digitalWrite(m1, LOW);
  delay(1000);
  //Serial.println("OBSTACULO AL FRENTE");
}
if ((d1<=250)&&(r==3))           //rango 3
{
  digitalWrite(m1, HIGH);
  delay(2000);
  digitalWrite(m1, LOW);
  delay(500);
  //Serial.println("OBSTACULO AL FRENTE");
}
if (d2<=50&&(r==1))             //rango 1
{
  digitalWrite(m2, HIGH);
  delay(500);
  digitalWrite(m2, LOW);
  delay(2000);
  //Serial.println("OBSTACULO AL FRENTE");
}
if ((d2<=150)&&(r==2))           //rango 2
{
  digitalWrite(m2, HIGH);
  delay(1000);
  digitalWrite(m2, LOW);
  delay(1000);
  //Serial.println("OBSTACULO AL FRENTE");
}
if ((d2<=250)&&(r==3))           //rango 3
{
  digitalWrite(m2, HIGH);
  delay(2000);
  digitalWrite(m2, LOW);
  delay(500);
  //Serial.println("OBSTACULO AL FRENTE");
}
if (d3<=50&&(r==1))             //rango 1

```

```

{
  digitalWrite(m3, HIGH);
  delay(500);
  digitalWrite(m3, LOW);
  delay(2000);
  //Serial.println("OBSTACULO AL FRENTE");
}
if ((d3<=150)&&(r==2)) //rango 2
{
  digitalWrite(m3, HIGH);
  delay(1000);
  digitalWrite(m3, LOW);
  delay(1000);
  //Serial.println("OBSTACULO AL FRENTE");
}
if ((d3<=250)&&(r==3)) //rango 3
{
  digitalWrite(m3, HIGH);
  delay(2000);
  digitalWrite(m3, LOW);
  delay(500);
  //Serial.println("OBSTACULO AL FRENTE");
}
}
}

```

### 2.6.6. ENSAMBLAJE DE LAS ETAPAS DEL DISPOSITIVO

Una vez que la programación fue cargada al Arduino y además probada con todos sus componentes individuales, se procedió a armar las dos etapas.

Para la primera etapa se necesitó la caja plástica, en la que se colocó la tarjeta Arduino, el shield con el reproductor de sonido, el regulador de volumen, las baterías, el motor vibrador y los botones pulsadores. Además se incluyó un interruptor para encender y apagar el sistema de manera sencilla.

Se hicieron orificios en la caja para las siguientes conexiones: entrada del cable USB, conector de alimentación de 5V, rodela del control del volumen, conector de 3,5 mm para audífonos y los cables para la conexión con los sensores ubicados en las gafas.

Se colocó la placa Arduino dentro de la caja, se realizó las conexiones que sirvieron para la alimentación de voltaje que recibe de la batería, luego se aseguró a la base y sobre esta se colocó el shield que contiene el reproductor de sonido y la tarjeta microSD con los comandos de voz previamente cargados en orden.



**Figura 52:** Placa Arduino y placa shield ubicadas dentro de la caja plástica

**Fuente:** Alex Molina Espinoza

Para la segunda etapa de armado, se puso el cinturón ajustable por la parte posterior de la caja, en donde se encuentran los orificios para la salida de los cables que van conectados a los sensores de las gafas, y a los mirco motores vibradores. Los motores fueron colocados en el interior del cinturón.

Una vez conectados todos los cables al interior de la caja, se colocaron los pulsadores en la tapa.



## 2.6.8. COMPROBACIÓN DE LA HIPÓTESIS

### 2.6.8.1. Planteamiento de la hipótesis estadística

En vista de que no se pudo especificar la dirección de la diferencia que existió entre las dos medias poblacionales (en este caso, hablamos de los tiempos promedios), se plantearon las siguientes hipótesis alternas unilaterales.

$H_0$  = El tiempo promedio que se demora una persona no vidente usando las gafas electrónicas en recorrer un pasillo es mayor o igual al tiempo promedio que se demora una persona no vidente en recorrer el mismo pasillo utilizando un bastón blanco.

$H_1$  = El tiempo promedio que se demora una persona no vidente con las gafas electrónicas en recorrer un pasillo es menor al tiempo promedio que se demora una persona no vidente en recorrer el mismo pasillo utilizando un bastón blanco.

$$H_0: \mu_s \geq \mu_d$$

$$H_1: \mu_s < \mu_d$$

Para medir el tiempo que se demora una persona no vidente en recorrer un pasillo, se dividió a las personas en dos grupos. Los primeros utilizaron un bastón, y se les indicó que debían llegar hasta la habitación que contenía un objeto que emitía un determinado sonido, para lo cual debían únicamente ayudarse con el bastón y no usar las manos para detectar obstáculos en la ruta. El segundo grupo se colocaría las gafas electrónicas, y de igual manera, debían llegar a la habitación donde se encontraba la fuente del sonido que escuchaban, sin utilizar las manos o brazos para detectar los obstáculos en la ruta. Los resultados fueron los siguientes:

En caso de los tiempos, se inició el cronometraje cuando pasaban de la puerta, y se detenía una vez que alcanzaban el objeto que producía el sonido. Los tiempos resultantes se redondearon a los segundos, obteniendo los datos de la Tabla 10.

### 2.6.8.2. Establecimiento del nivel de significancia

La prueba se realizó con un 95 % de confiabilidad, es decir con un nivel de significancia de  $\alpha = 0.05$ .

### 2.6.8.3. Descripción de las poblaciones y suposiciones

Se asume que las poblaciones de las que provienen las muestras están normalmente distribuidas y que sus varianzas no son conocidas. Las muestras son independientes.

### 2.6.8.4. Determinación del estadístico pertinente

El estadístico más adecuado es  $\bar{x}_s - \bar{x}_d$  que, en virtud de que se supone que las poblaciones están distribuidas en forma aproximadamente normal, podemos considerar como normalmente distribuido. Si  $H_0$  es verdadera, la media de la distribución muestral es  $\mu_s - \mu_d \geq 0$  y su varianza es  $\frac{\sigma_s^2}{n_1} - \frac{\sigma_d^2}{n_2}$ . Como  $\sigma_s^2$  y  $\sigma_d^2$  son desconocidas, no podemos calcular la varianza verdadera de  $\bar{x}_s - \bar{x}_d$  y, en consecuencia, excluimos a  $z$  como estadístico de prueba.

### 2.6.8.5. Determinación del estadístico de prueba.

Como se observó en el paso anterior,  $z$  no es el estadístico de prueba apropiado. Como se supone que las dos poblaciones están distribuidas en forma aproximadamente normal, con varianzas desconocidas pero iguales, el estadístico de prueba más adecuado es el estadístico  $t$  de Student, con  $n_1 + n_2 - 2$  grados de libertad.

$$t_0 = \frac{\bar{x}_s - \bar{x}_d}{\frac{S_s^2}{n_1} - \frac{S_d^2}{n_2}}$$

Siendo  $\bar{x}_s$  el tiempo promedio que se demoraron las personas no videntes usando las gafas electrónicas;  $\bar{x}_d$  el tiempo promedio que se demoraron las personas no

videntes usando el bastón blanco;  $S_s^2$  la varianza de los tiempos de las personas no videntes que usaron las gafas electrónicas;  $S_d^2$  la varianza de los tiempos de las personas no videntes que usaron el bastón blanco;  $n_1$  es el tamaño de la muestra 1;  $n_2$  es el tamaño de la muestra 2; además,  $n_1 = n_2$ . Consecuencia, excluimos a  $z$  como estadístico de prueba.

**Tabla 10:** Valores de los tiempos de las personas que usaron las gafas y de las personas que usaron el bastón

NÚMERO DE MUESTRAS	TIEMPO EN SEGUNDOS	
	USANDO LAS GAFAS ELECTRÓNICAS	USANDO EL BASTÓN BLANCO
1	169	49
2	92	41
3	86	43
4	60	66
5	51	54
6	130	41
7	55	37
8	121	37
9	86	42
10	68	35
11	64	49
12	64	39

**Fuente:** Alex Molina Espinoza

**Tabla 11:** Tabla de valores estadísticos.

<i>DETALLE</i>	<i>TIEMPOS</i>	
	<i>USANDO LAS GAFAS ELECTRÓNICAS</i>	<i>USANDO EL BASTÓN BLANCO</i>
Media muestral	87,17	44,42
Varianza muestral	1293,03	78,08
Número de observaciones	12	12
Diferencia hipotética de las medias	$\bar{x}_s - \bar{x}_d = 87,17 - 44,42 = 42,75$	
Grados de libertad	$n_1 + n_2 - 2 = 12 + 12 - 2 = 22$	
Estadístico tabulado t	-1,7171	
Valor crítico de t (una cola)	$t_0 = \frac{\bar{x}_s - \bar{x}_d}{\sqrt{\frac{S_s^2}{n_1} - \frac{S_d^2}{n_2}}}$ $t_0 = \frac{42,75}{\sqrt{\frac{1293,03}{12} - \frac{78,08}{12}}} = 4,25$	

**Fuente:** Observación de resultados

**Elaborado por:** Alex Molina Espinoza

### 2.6.8.6. Establecimiento de las regiones de rechazo y aceptación

Como los grados de libertad son  $12 + 12 - 2 = 22$  y como  $\alpha = 0.05$ , encontramos mediante la tabla A-3 que el valor crítico para el estadístico  $t$  es -1,7171. La prueba es unilateral, de una cola ubicada a la izquierda en el gráfico.

Grados de libertad	Distribución t: Valores críticos t				
	Área en una cola		Área en dos colas		
	0.005	0.01	0.025	0.05	0.10
1	61.657	31.821	12.706	6.314	1.978
2	9.925	6.965	4.303	2.920	1.886
3	5.841	4.541	3.182	2.353	1.886
4	4.604	3.747	2.776	2.133	1.821
5	4.032	3.365	2.571	2.015	1.753
6	3.707	3.143	2.447	1.943	1.706
7	3.499	2.968	2.365	1.895	1.677
8	3.355	2.846	2.306	1.860	1.654
9	3.250	2.761	2.262	1.833	1.635
10	3.169	2.704	2.228	1.812	1.620
11	3.106	2.661	2.201	1.796	1.609
12	3.055	2.621	2.179	1.782	1.599
13	3.012	2.583	2.160	1.771	1.590
14	2.977	2.547	2.145	1.761	1.582
15	2.947	2.514	2.131	1.752	1.575
16	2.921	2.483	2.120	1.744	1.568
17	2.898	2.457	2.110	1.738	1.562
18	2.878	2.432	2.101	1.734	1.557
19	2.861	2.410	2.093	1.729	1.553
20	2.845	2.390	2.086	1.725	1.549
21	2.831	2.371	2.080	1.721	1.545
22	2.819	2.354	2.074	1.717	1.542
23	2.807	2.338	2.069	1.714	1.539
24	2.797	2.323	2.064	1.711	1.536
25	2.787	2.309	2.060	1.708	1.533
26	2.779	2.296	2.056	1.706	1.531
27	2.771	2.283	2.052	1.703	1.528
28	2.763	2.271	2.048	1.701	1.526
29	2.756	2.260	2.045	1.699	1.524
30	2.750	2.250	2.042	1.697	1.522
31	2.744	2.241	2.040	1.696	1.520
32	2.738	2.232	2.037	1.694	1.519
34	2.730	2.219	2.032	1.688	1.516
36	2.722	2.207	2.028	1.683	1.513
38	2.714	2.195	2.024	1.678	1.510
40	2.708	2.184	2.021	1.674	1.508
45	2.699	2.167	2.014	1.670	1.505
50	2.692	2.152	2.009	1.667	1.503
55	2.686	2.139	2.004	1.665	1.501
60	2.680	2.127	2.000	1.663	1.500
65	2.674	2.115	1.997	1.661	1.499
70	2.669	2.104	1.994	1.659	1.498
75	2.664	2.094	1.992	1.658	1.497
80	2.659	2.084	1.990	1.657	1.496
90	2.652	2.068	1.987	1.656	1.495
100	2.646	2.054	1.984	1.655	1.494
200	2.641	2.045	1.982	1.654	1.493
300	2.638	2.039	1.981	1.653	1.492
400	2.636	2.035	1.980	1.653	1.492
500	2.635	2.033	1.980	1.653	1.492
750	2.634	2.031	1.980	1.653	1.492
1000	2.633	2.030	1.980	1.653	1.492
2000	2.632	2.029	1.980	1.653	1.492
Infinito	2.632	2.029	1.980	1.653	1.492

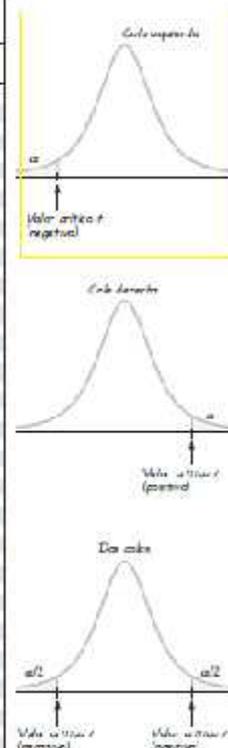


Figura 55: Tabla de t student

Fuente: Tabla de contingencia

### 2.6.8.7. Establecer la regla de decisión

Como  $4,25 > -1,7171$ , podemos aceptar la hipótesis nula. Como se puede observar en la siguiente gráfica, el valor crítico de t cae en la zona de aceptación de la hipótesis nula  $H_0$ .

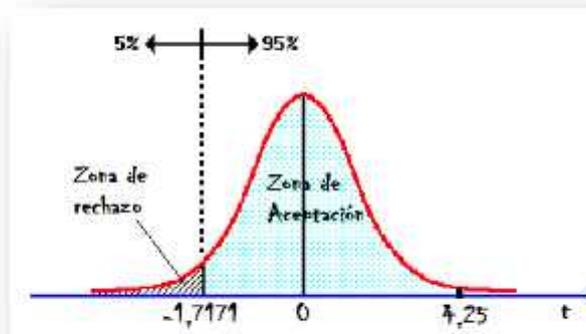


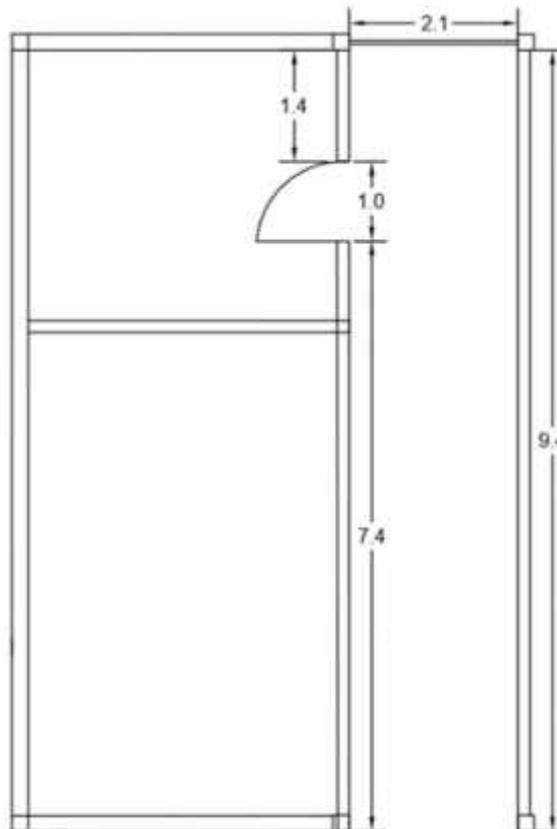
Figura 56: Campana de Gauss para decisión

Fuente: Alex Molina Espinoza

## CAPÍTULO III

### 3. RESULTADOS

Para la realización de la prueba, se requirió de la colaboración de 12 personas, que recorrieron un pasillo de una longitud de 9 metros, hasta ubicar una habitación en donde se colocó una fuente de sonido como ayuda para la localización. En un primer momento las personas realizaron la prueba usando el bastón blanco hasta llegar al interior de la habitación, mientras que luego se les colocó el prototipo de las gafas para recorrer el mismo corredor hasta ubicar la fuente de sonido.



**Figura 57:** Plano del pasillo y la oficina usada para las pruebas

**Fuente:** Alex Molina Espinoza

El entorno Arduino fue diseñado para tener una interfaz amigable y personalizable, a más de proporcionar un rápido acceso a la programación de la

tarjeta, pudiendo guardar cualquier cambio necesario de una forma simple desde cualquier computadora disponible. Debido a que se realizó la implementación de la interfaz de control del dispositivo con el entorno Arduino, se pudieron realizar cambios sencillos en cuanto a la intensidad de vibración de los motores, para que las personas puedan percibir de mejor manera los obstáculos, y también nos permitió manejar los rangos para ajustar las distancias al espacio del que se dispuso para las pruebas.

El diseño en etapas del prototipo, permitió controlar de mejor manera la forma en la que se desarrolló las pruebas del dispositivo, puesto que se pudo ajustar a cada una de las personas con facilidad, evitando que resultara incómodo el llevar puesto el cinturón o las gafas.

Uno de los inconvenientes que se presentaron al momento de recolectar los datos en la realización de las pruebas, fue que al no conocer el ambiente por donde realizarían el recorrido, algunas personas se sintieron inseguras, y esto influyó en el tiempo en que se demoraron en recorrer el pasillo hasta la habitación. Este contratiempo no impidió la realización de la prueba, pero influyó en la recolección de algunos resultados

## CAPÍTULO IV

### 4. DISCUSIÓN

El mundo de hoy se encuentra rodeado de medios técnicos y tecnológicos, aparatos que ayudan y facilitan la vida de las personas y la ocupación del tiempo libre, se descarga de pesadas tareas, tecnología al alcance de todos que recrea y ayuda a las personas.

Cuando estos medios tecnológicos además de servir a la familia de ayuda en la vida de una persona ¿por qué no utilizarlos para aquellas personas con capacidades diferentes y en este caso los no videntes? La presente propuesta tiene como finalidad ayudar a las personas no videntes y darles mayor accesibilidad a las tareas diarias y principalmente a no depender de alguien que los guíe o del bastón que usualmente utilizan y que por lo general les mantiene las manos ocupadas.

La historia del bastón blanco que data desde 1931, ha permitido a muchas personas movilizarse dentro de los lugares a que ellas tienen acceso. Entre los promotores de su uso figura el médico hepatólogo Mexicano Alfonso Ortiz Tirado, quien pidió la obligatoriedad del uso del bastón blanco. Su uso se popularizó tanto que en Estados Unidos se celebra el 15 de octubre de cada año el “Día internacional del Bastón Blanco”, pero en realidad aún se desconoce el nombre de su inventor ya que hubo muchas propuestas (Freiwald, 2014).

Los resultados de la presente investigación comprueban las hipótesis propuestas. Las personas que usan el bastón posiblemente por la costumbre de usarlo, se sienten más seguras, pero al momento de emplear otro tipo de objeto que facilite su movilidad y que para ellos resulta desconocido, naturalmente se sintieron inseguras. A pesar de los inconvenientes citados en los resultados, el prototipo de Gafas es totalmente útil para personas no videntes. Con el tiempo, este prototipo se lo puede mejorar de tal manera que sea un equipo de menos tamaño y por sobre todo estético para el uso de la persona que lo requiera.

Lo más importante de la propuesta es que las personas conozcan la idea de las Gafas para no videntes y conozcan los beneficios que les puede traer para su movilidad y que se lo lleve a las diferentes instancias para lograr un financiamiento para la fabricación de ellas. De esta forma se estará ayudando a las personas no videntes de la ciudad de Riobamba y en, lo posterior, se lo podrá llevar a todas las provincias de nuestro país.

## **CAPITULO V**

### **5. CONCLUSIONES Y RECOMENDACIONES**

#### **5.1. CONCLUSIONES**

Al terminar la presente investigación se concluye lo siguiente:

- Se diseñó e implementó una interfaz auxiliar para personas no videntes utilizando gafas con sensores de proximidad, usando el entorno Arduino, dado que es de fácil comprensión y maneja un lenguaje de programación ampliamente usado.
- Se desarrolló la interfaz que fue capaz de poder procesar e interpretar señales para determinar distancias entre una persona y un objeto próximo a esta, facilitando la posibilidad de cambios en el programa para posteriores ajustes que requiera el dispositivo, dependiendo del entorno en el que se requiera su uso.
- Se implementó el modelo de la interfaz sobre unas gafas que puedan ser llevadas fácilmente, teniendo como enfoque la comodidad de la persona que pruebe el dispositivo.
- Se realizó las pruebas en personas no videntes, tanto con el bastón blanco, como con el dispositivo, para comparar la efectividad de cada uno en un mismo ambiente, y así se pudo determinar las ventajas y desventajas del sistema implementado.

## 5.2. RECOMENDACIONES

Luego de haber realizado las conclusiones de la investigación se proponen las siguientes recomendaciones:

- El diseño e implementación de la interfaz auxiliar para personas no videntes utilizada con gafas y sensores de proximidad debe ser mejorado para facilidad de manejo de la persona que lo va a utilizar.
- La interfaz que fue capaz de procesar e interpretar señales para determinar distancias entre una persona y un objeto próximo a esta es recomendable por su facilidad de manejo, su amplio uso y desarrollo, y la infinidad de sensores que existen y configuración de distintas tarjetas.
- La sensibilidad de los motores vibradores al momento de entregar al usuario las alertas de proximidad a un objeto, puede ser mejorada con el uso de señales PWM, ya que permitirían que no sean necesarios rangos de distancia, puesto que estos pueden ser asignados a una señal de 8 bits
- El modelo que se implementó de la interfaz sobre unas gafas puede ser llevada fácilmente y es muy útil. Con otras investigaciones que se puedan realizar el modelo podría ser mejorado reduciendo el tamaño del dispositivo y la estética de los sensores sobre las gafas, para que este sea portable con mayor facilidad y comodidad para la persona que lo use.
- Las pruebas con el dispositivo en personas no videntes determinaron que este tipo de dispositivo sí se lo puede proponer y recomendar para su elaboración para la ayuda las personas no videntes.

## **CAPÍTULO VI**

### **6. PROPUESTA**

#### **6.1. TÍTULO DE LA PROPUESTA**

ELABORACIÓN DE GAFAS ELECTRÓNICAS PARA PERSONAS NO VIDENTES

#### **6.2. INTRODUCCIÓN**

En la sociedad actual, las personas no videntes o que posean algún tipo de impedimento visual, no pueden movilizarse con facilidad por un espacio público, ya que en su diseño no se han considerado las mejoras necesarias para disminuir los obstáculos físicos que afectan su capacidad para trasladarse de un lugar a otro.

Es por este motivo que diseñar un dispositivo fácil de usar y llevar, capaz de alertar de manera sencilla y efectiva sobre algún obstáculo que se encuentren cerca del usuario, se constituirá en un beneficio para este grupo de personas, dado que al mejorar su movilidad podrán integrarse de mejor manera para realizar actividades comunes

#### **6.3. OBJETIVOS:**

##### **6.3.1. OBJETIVO GENERAL**

- Elaborar unas gafas electrónicas para personas no videntes.

##### **6.3.2. OBJETIVOS ESPECÍFICOS**

- Desarrollar la interfaz capaz de poder procesar e interpretar señales para determinar la relación de un objeto con respecto a una persona.
- Usar comandos PMW para el control de las alertas para el usuario de acuerdo a las distancias de los objetos.

- Buscar mejores alternativas de fabricación de la tarjeta que permite la comunicación entre las diferentes partes del dispositivo.

#### **6.4. FUNDAMENTACIÓN CIENTÍFICO – TÉCNICA**

El dispositivo está conformado por un sensor ultrasónico denominado PING))) del fabricante Parallax; este sensor puede detectar obstáculos hasta 4 metros de distancia en condiciones normales, utilizando una onda de sonido inaudible que rebota en cualquier objeto sólido con un retraso de 5 $\mu$ s.

Como parte principal del prototipo se tiene la tarjeta Arduino MEGA, que posee un microcontrolador ATmega, esta será la encargada de analizar los datos recibidos del sensor para luego poder entregar una alerta al usuario. Las alertas del dispositivo serán de forma auditiva utilizando el reproductor de sonido WTV020 conjuntamente con un sintetizador de voz, y sensoriales mediante un motor vibrador integrado a la parte externa del dispositivo.

#### **6.5. DESCRIPCIÓN DE LA PROPUESTA**

El proyecto parte del uso de sensores de distancia, para poder detectar un objeto que se encuentre frente al usuario. Entre los sensores de distancia más eficientes y confiables de usar en condiciones ambientales normales se tiene los sensores ultrasónicos de distancia, los cuales mediante la emisión de un sonido, pueden detectar obstáculos fácilmente a una distancia aceptable de entre 2cm y 4m. El tiempo que se demora el sonido en ir desde el sensor y producirse el eco al encontrar un obstáculo, será la información que el sensor nos entregue.

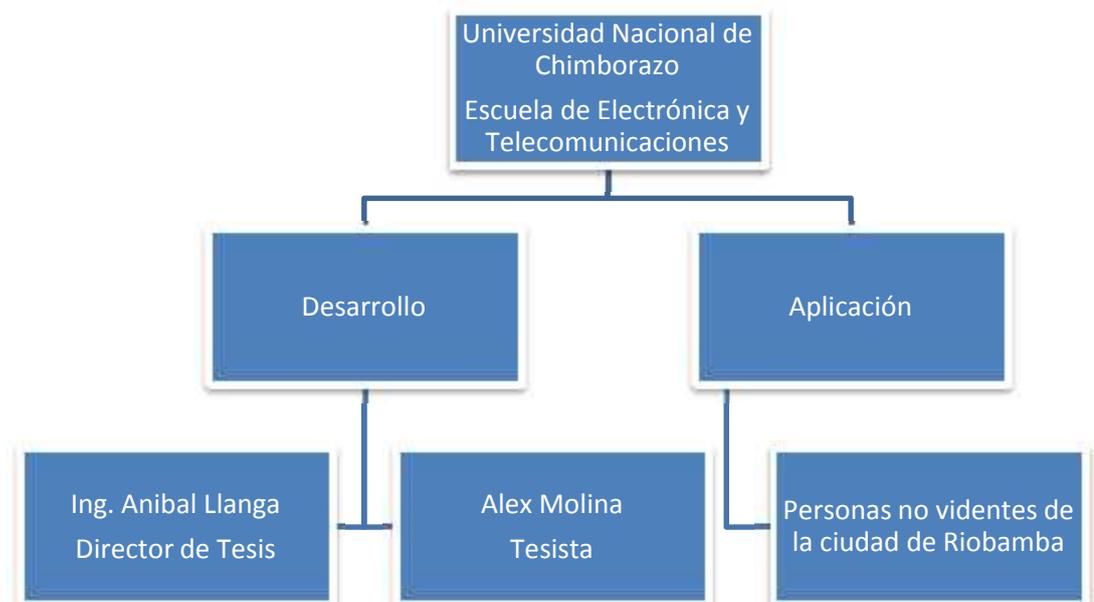
Una vez obtenidos los datos del sensor, se necesitó un microcontrolador para poder interpretarlos y obtener como resultado una medición de una distancia. Se usó la tarjeta Arduino Uno para la conversión del tiempo en distancia, además de que sirvió para poder integrar una interfaz sencilla para que el usuario interactúe con el prototipo.

Las alertas auditivas se las grabaron utilizando el sintetizador de voz Loquendo, para luego convertirlas al formato *.ad4* y se las grabó en un micro SD de 1 Gb, la cual debió ser colocada en el reproductor de sonido y conectada a la tarjeta Arduino.

El usuario pudo interactuar con el dispositivo mediante el uso de tres pulsadores, que al igual que el sensor ultrasónico, fueron montados sobre unas gafas para que sean de fácil acceso. El primer botón ayuda al usuario a seleccionar uno de los tres rangos de medición de distancia que se usa para hallar los obstáculos, ya sea que se encuentre en un entorno abierto o en un corredor estrecho. La orden de confirmación de selección de rango es ingresada mediante el segundo botón, y el tercer botón sirve para un reset general del sistema. Todos los comandos ingresados son confirmados por una alerta de voz previamente grabada.

Una vez montadas todas las partes del dispositivo, el usuario fue alertado si existe un objeto al frente mediante una alerta de voz concisa, además de la alerta sensorial proporcionada por el motor vibrador, que va variando su intensidad dependiendo de la distancia que se encuentre el objeto.

## 6.6. DISEÑO ORGANIZACIONAL



## **6.7. MONITOREO Y EVALUACIÓN DE LA PROPUESTA**

El monitoreo y la evaluación de la propuesta se la realizará a través de más pruebas con otras personas para determinar si el uso de las gafas es o no relevante. Esto se lo hará con la utilización del prototipo en tiempos más prolongados para poder observar si su aplicación es útil o no para las personas no videntes.

Además, dentro del monitoreo que se realice, se propondrá el aumento de las personas que lo usen para realizar una evaluación de resultados, efectos e impacto una vez finalizada la etapa de prueba.

## 7. BIBLIOGRAFÍA

1. Aguirre Solvez Lorena, Estudio de una red de sensores sin hilos basados en la tecnología Arduino bajo protocolos de comunicaciones Zigbee, Editorial Politécnica, Catalunya – España.
2. Arlanzón David, Introducción a la plataforma Open Source Arduino para aplicaciones domóticas y automatización, Catalunya – España, 2012
3. Banzi, M. (2012). *Arduino*. Obtenido de What is arduino: <http://www.arduino.cc/>
4. Diccionario. (2014). *Diccionario de la Real Academia de la Lengua*.
5. [es.wikipedia.org/wiki/Sensor](http://es.wikipedia.org/wiki/Sensor). (2014).
6. Farcy, R. (2006). *Electronic travel aids and orientation aids for Blind people*.
7. Freiwald, D. (2014). *Bastón Guía para personas no videntes*.
8. García, M. (2013). *Desarrollo de un kit educativo de robótica basado en Arduino*.
9. Nato, F. (2005). *The Cognitive Foundations of Movility and electronic spatial sensing for the blind*.
10. Ortega, E. (junio de 2014). *Slideshare*. Obtenido de [es.slideshare.net/clasificacion-de-sensores-36167798](http://es.slideshare.net/clasificacion-de-sensores-36167798)
11. Reyes, C. A. (2008). *Microcontroladores PIC y programación en Basic*. Quito.
12. Rodríguez, W. (2014). *Software libre para educación y educación en Ingeniería*. Bogotá - Colombia.
13. Triola, M. F. (2007). *PROBABILIDAD Y ESTADISTICA*. PEARSON EDUCACION.
14. Torrente Artero Oscar, Arduino: Curso Práctico de formación. Madrid – España.-Grupo RC. 2013
15. Yuan, D., Manduchi, R.: A Tool for Range Sensing and Environment Discovery for the Blind. Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. (2004).

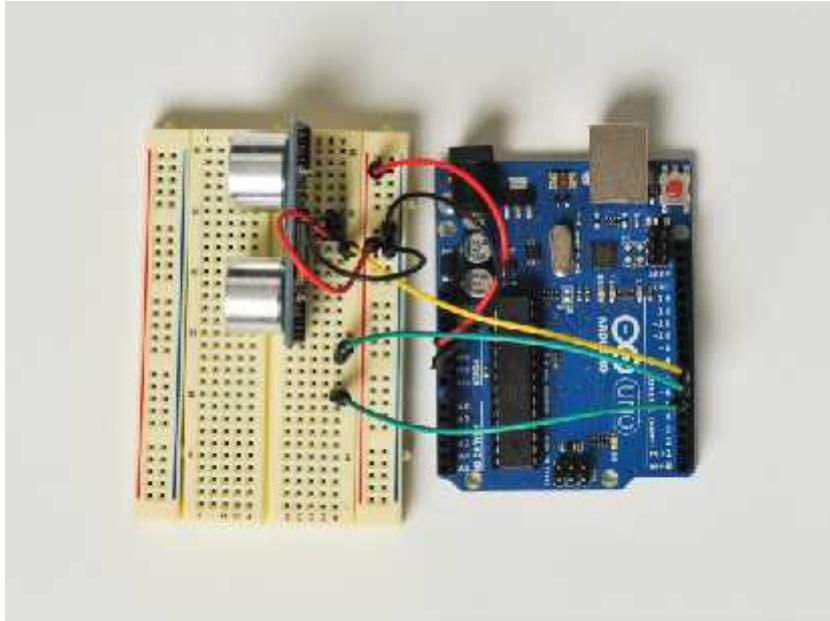
### Webgrafía

- [http://usuarios.discapnet.es/ojo\\_oido/orientacion\\_y\\_movilidad.htm](http://usuarios.discapnet.es/ojo_oido/orientacion_y_movilidad.htm)
- <http://www.tsbvi.edu>

- <http://www.x-robotics.com/sensores.htm#Reflexivo>
- <http://www.superrobotica.com/sensores.htm>
- <http://es.arduino.cc>
- <http://www.parallax.com>
- <http://www.datasheet.com>

## 8. APÉNDICES Y ANEXOS

### Anexo 1. Etapas iniciales del desarrollo del prototipo



Adquisición de datos con la tarjeta Arduino Uno y el sensor ultrasónico



Primer diseño del dispositivo con la tarjeta Arduino Uno

**Anexo 2. Evidencias de las pruebas realizadas con el bastón blanco**



Indicaciones generales del desarrollo de la prueba



Recorriendo el pasillo con el bastón blanco



Persona ingresando a la habitación luego de cruzar el pasillo



Alcanzando la fuente de sonido en la habitación

**Anexo 3. Evidencias de las pruebas realizadas con el dispositivo**



Indicaciones generales dadas por el Ing. Aníbal Llanga



Usuario recorriendo el pasillo con el dispositivo

## Anexo 4. Datasheets

[www.elechouse.com](http://www.elechouse.com)

### 1.Features

- \*OTP(one time programmable) VOICE-CHIP
- \*Operating voltage:2.5V-3.6V
- \* WTV010 with 10seconds , WTV020 with 20seconds duration at 6KHz .
- \*Sampling rate supported form 6KHz ,8KHz,10KHZ ..... 32KHz max.
- \*Higher sampling rate better sound quality and shorter duration.
- \*With DAC and PWM audio output .
- \*DAC output connect to amplifier . PWM output can direct a 0.5W/8ohms speaker
- \*Equipped with 16 bits DAC
- \* Mute will not take up any memory space.
- \* The same voices can be re-use and will not take up memory space.
- \*Multiple control modes: key mode(keyboard), parallel (COM+SBT), one line serial , two lines serial , three lines serial
- \* Quiescent current 2uA
- \*Support WAV ,16Bit, mono format.
- \*Equip with PC software for project editing.

### 2. Chip models

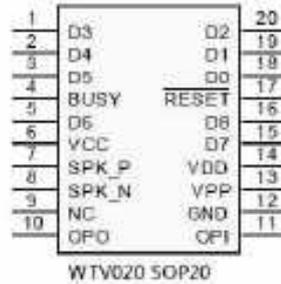
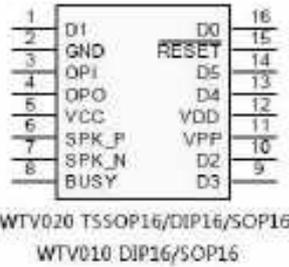
WTV010/WTV020

Model	WTV010-DIP16/SOP16/TSSOP	WTV020-DIP16/SOP16/TSSOP	WTV020-SOP20
Duration	10 s	20 s	20 s
Track	Mono	Mono	Mono
Audio output	DAC/PWM	DAC/PWM	DAC/PWM
Key board trigger I/O	4	4	7
Parallel address	4	4	7
Control modes	Keyboard / parallel(COM+SBT)/Serial	Keyboard / parallel(COM+SBT)/Serial	Keyboard / parallel(COM+SBT)/Serial

Note:The duration caculate at 6KHz sampling rate ,Control modes can be chosen in PC software .

### 3. Package sketch map

WTV010 with DIP 16 ans SOP 16 packages. And WTV020 with DIP 16 , SOP 16,TSSOP16 and SOP 20 packages.



#### 4.Pin's descriptions

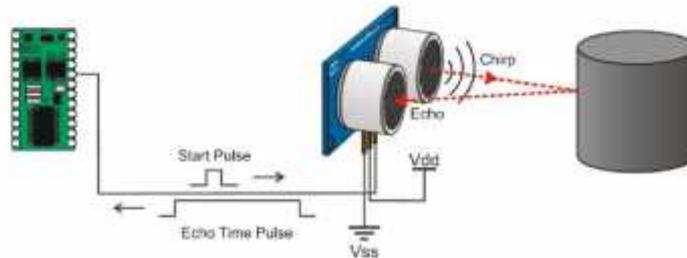
##### 4.1 For WTV010/WTV020 DIP16/SOP16/TSSOP16 package

Pin No.	Pin name	Brief	Descriptions
1	D1	K2/A0/CS	Keyboard trigger I/O / Parallel address / Three lines serial CS(Chip select)
2	GND	GND	GND
3	OPI	OPI	Internal amplifier reactive input
4	OPO	OPO	Internal amplifier reactive output
5	VCC	VCC	Chip analog power
6	SPK-P	SPK-P	DAC /PWM audio output
7	SPK-N	SPK-N	DAC /PWM audio output
8	BUSY	BUSY	Busy
9	D3	K4/A2/DI	Keyboard trigger I/O , Parallel address/ one line data /two lines serial data / three lines serial data
10	D2	K3/A1/SCK	Keyboard trigger I/O / Parallel address / Two lines serial Clock/Three lines serial Clock
11	VPP	VPP	Chip program power
12	VDD	VDD	Chip digital power
13	D4	A3	Parallel address
14	D5	D5	Data
15	RESET	RESET	Chip reset (keep low level 2ms to reset)
16	D0	K1 / TEST/SBT	Keyboard trigger I/O / Parallel address trigger SBT

## PING)))™ Ultrasonic Distance Sensor (#28015)

The Parallax PING))) ultrasonic distance sensor provides precise, non-contact distance measurements from about 2 cm (0.8 inches) to 3 meters (3.3 yards). It is very easy to connect to microcontrollers such as the BASIC Stamp®, SX or Propeller chip, requiring only one I/O pin.

The PING))) sensor works by transmitting an ultrasonic (well above human hearing range) burst and providing an output pulse that corresponds to the time required for the burst echo to return to the sensor. By measuring the echo pulse width, the distance to target can easily be calculated.



### Features

- Range: 2 cm to 3 m (0.8 in to 3.3 yd)
- Burst indicator LED shows sensor activity
- Bidirectional TTL pulse interface on a single I/O pin can communicate with 5 V TTL or 3.3 V CMOS microcontrollers
- Input trigger: positive TTL pulse, 2  $\mu$ s min, 5  $\mu$ s typ.
- Echo pulse: positive TTL pulse, 115  $\mu$ s minimum to 18.5 ms maximum.
- RoHS Compliant

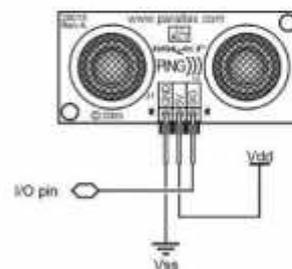
### Key Specifications

- Supply voltage: +5 VDC
- Supply current: 30 mA typ; 35 mA max
- Communication: Positive TTL pulse
- Package: 3-pin SIP, 0.1" spacing (ground, power, signal)
- Operating temperature: 0 – 70° C.
- Size: 22 mm H x 46 mm W x 16 mm D (0.84 in x 1.8 in x 0.6 in)
- Weight: 9 g (0.32 oz)

### Pin Definitions

GND	Ground (Vss)
5 V	5 VDC (Vdd)
SIG	Signal (I/O pin)

The PING))) sensor has a male 3-pin header used to supply ground, power (+5 VDC) and signal. The header may be plugged directly into solderless breadboard, or into a standard 3-wire extension cable (Parallax part #805-000012).

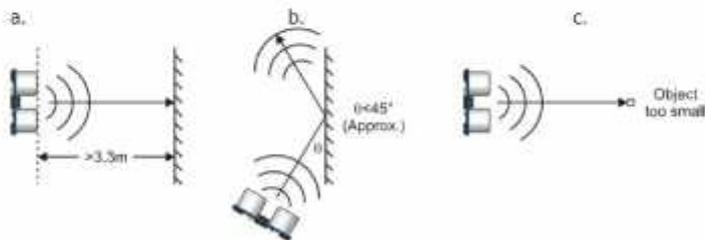




## Practical Considerations for Use

### Object Positioning

The PING))) sensor cannot accurately measure the distance to an object that: a) is more than 3 meters away, b) that has its reflective surface at a shallow angle so that sound will not be reflected back towards the sensor, or c) is too small to reflect enough sound back to the sensor. In addition, if your PING))) sensor is mounted low on your device, you may detect sound reflecting off of the floor.



### Target Object Material

In addition, objects that absorb sound or have a soft or irregular surface, such as a stuffed animal, may not reflect enough sound to be detected accurately. The PING))) sensor will detect the surface of water, however it is not rated for outdoor use or continual use in a wet environment. Condensation on its transducers may affect performance and lifespan of the device.

### Air Temperature

Temperature has an effect on the speed of sound in air that is measurable by the PING))) sensor. If the temperature ( $^{\circ}\text{C}$ ) is known, the formula is:

$$C_{sp} = 331.5 + (0.6 \times T_c) \text{ m/s}$$

The percent error over the sensor's operating range of 0 to 70  $^{\circ}\text{C}$  is significant, in the magnitude of 11 to 12 percent. The use of conversion constants to account for air temperature may be incorporated into your program (as is the case in the example B52 program given in the Example Programs section below). Percent error and conversion constant calculations are introduced in Chapter 2 of *Smart Sensors and Applications*, a Stamps in Class text available for download from the 28029 product page at [www.parallax.com](http://www.parallax.com).

**FILTERLESS 3W CLASS-D STEREO AUDIO AMPLIFIER**

**Description**

The PAM8403 is a 3W, class-D audio amplifier. It offers low THD+N, allowing it to achieve high-quality sound reproduction. The new filterless architecture allows the device to drive the speaker directly, requiring no low-pass output filters, thus saving system cost and PCB area.

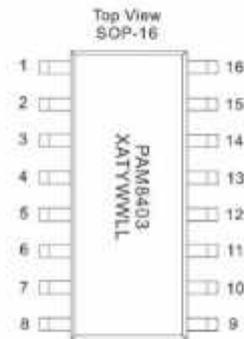
With the same numbers of external components, the efficiency of the PAM8403 is much better than that of Class-AB cousins. It can extend the battery life, which makes it well-suited for portable applications.

The PAM8403 is available in SOP-16 package.

**Features**

- 3W Output at 10% THD with a 4Ω Load and 5V Power Supply
- Filterless, Low Quiescent Current and Low EMI
- Low THD+N
- Superior Low Noise
- Efficiency up to 90%
- Short Circuit Protection
- Thermal Shutdown
- Few External Components to Save the Space and Cost
- Pb-Free Package

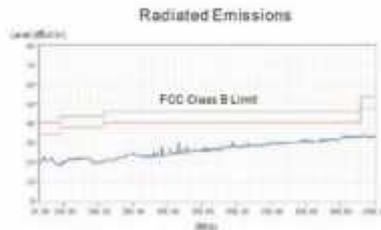
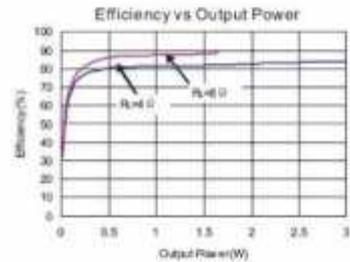
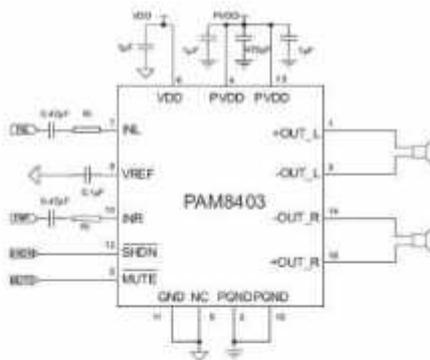
**Pin Assignments**



**Applications**

- LCD Monitors / TV Projectors
- Notebook Computers
- Portable Speakers
- Portable DVD Players, Game Machines
- Cellular Phones/Speaker Phones

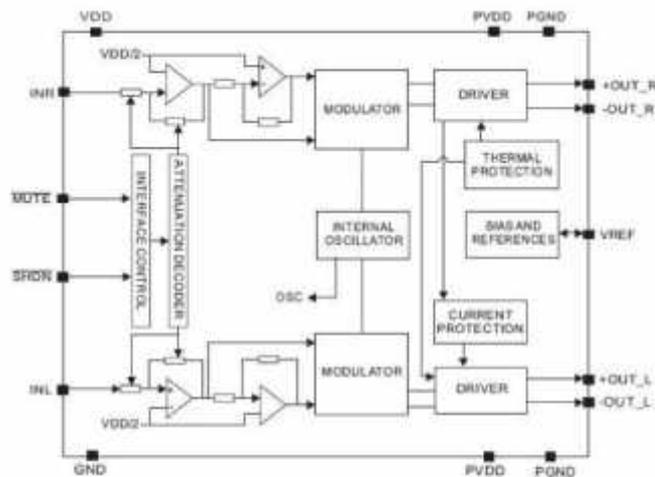
**Typical Applications Circuit**



### Pin Descriptions

Pin Number	Pin Name	Function
1	+OUT_L	Left Channel Positive Output
2	PGND	Power GND
3	-OUT_L	Left Channel Negative Output
4	PVDD	Power VDD
5	MUTE	Mute Control Input (active low)
6	VDD	Analog VDD
7	INL	Left Channel Input
8	VREF	Internal analog reference, connect a bypass capacitor from VREF to GND.
9	NC	No Connect
10	INR	Right Channel Input
11	GND	Analog GND
12	SHDN	Shutdown Control Input (active low)
13	PVDD	Power VDD
14	-OUT_R	Right Channel Negative Output
15	PGND	Power GND
16	+OUT_R	Right Channel Positive Output

### Functional Block Diagram



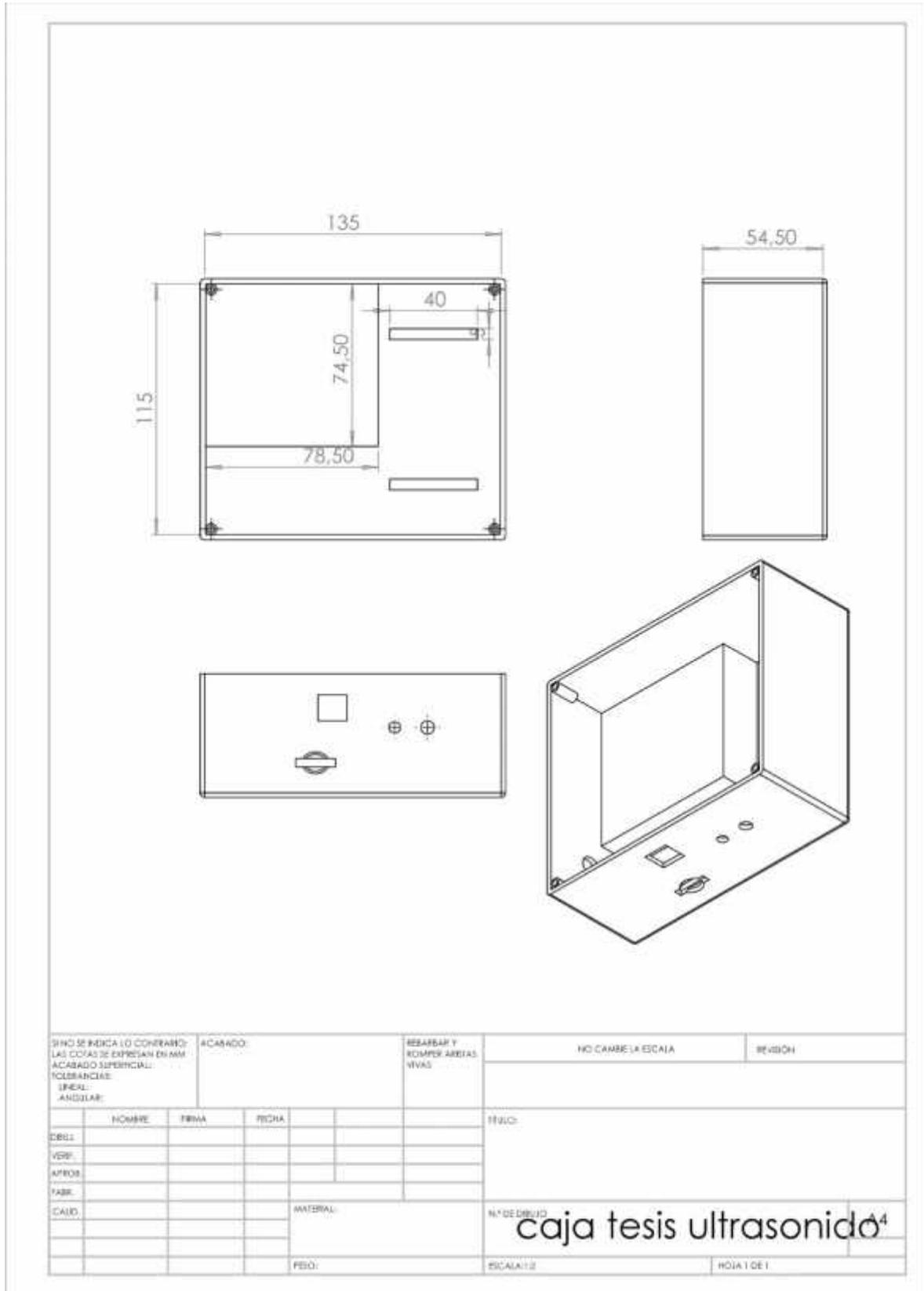


Diagrama de diseño de la caja para usar en impresora 3D