Neiva, __01-12-2017_____

Señores

CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN

UNIVERSIDAD SURCOLOMBIANA

Ciudad

El (Los) suscrito(s):

__Sebastián Ramos Díaz_____, con C.C. No. ___1.075.260.511_____,

__Javier Madden Molina Tique _____, con C.C. No. _____1082127026_____,

_____, con C.C. No. _____,

_____, con C.C. No. _____,

autor(es) de la tesis y/o trabajo de grado o _____

titulado__Diseño e implementación de prototipo de traductor móvil de lenguaje oral a lengua de señas mediante procesamiento de voz._____
presentado y aprobado en el año __2017_____ como requisito para optar al título de

__Ingeniero Electrónico_____;

Autorizo (amos) al CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN de la Universidad Surcolombiana para que con fines académicos, muestre al país y el exterior la producción intelectual de la Universidad Surcolombiana, a través de la visibilidad de su contenido de la siguiente manera:

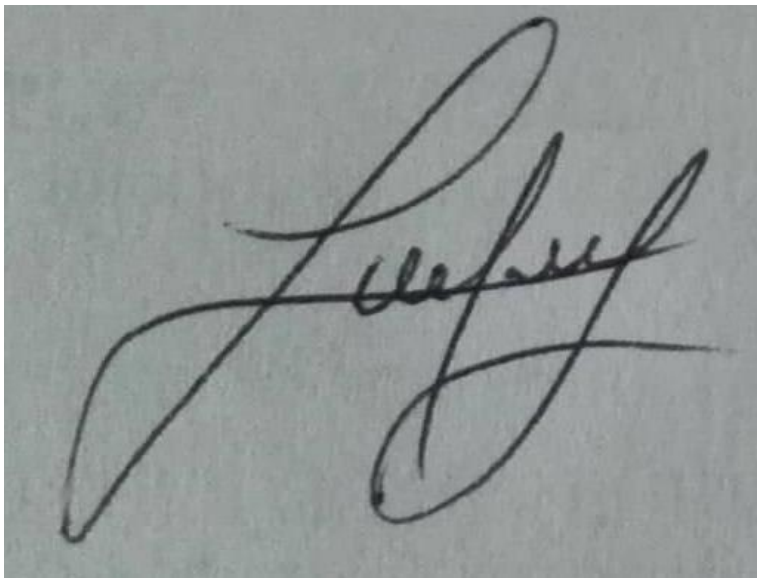
- Los usuarios puedan consultar el contenido de este trabajo de grado en los sitios web que administra la Universidad, en bases de datos, repositorio digital, catálogos y en otros sitios web, redes y sistemas de información nacionales e internacionales “open access” y en las redes de información con las cuales tenga convenio la Institución.

- Permita la consulta, la reproducción y préstamo a los usuarios interesados en el contenido de este trabajo, para todos los usos que tengan finalidad académica, ya sea en formato Cd-Rom o digital desde internet, intranet, etc., y en general para cualquier formato conocido o por conocer, dentro de los términos establecidos en la Ley 23 de 1982, Ley 44 de 1993, Decisión Andina 351 de 1993, Decreto 460 de 1995 y demás normas generales sobre la materia.

- Continúo conservando los correspondientes derechos sin modificación o restricción alguna; puesto que de acuerdo con la legislación colombiana aplicable, el presente es un acuerdo jurídico que en ningún caso conlleva la enajenación del derecho de autor y sus conexos.

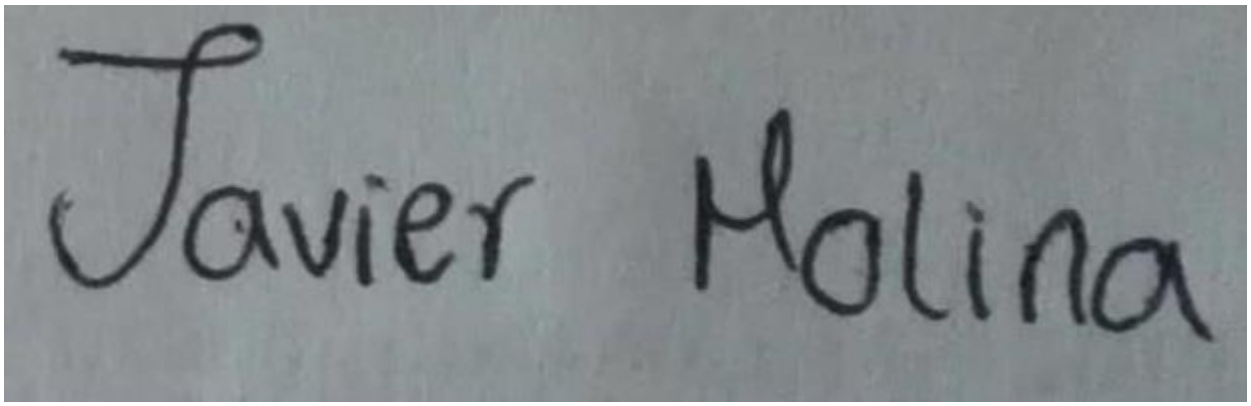
De conformidad con lo establecido en el artículo 30 de la Ley 23 de 1982 y el artículo 11 de la Decisión Andina 351 de 1993, “Los derechos morales sobre el trabajo son propiedad de los autores” , los cuales son irrenunciables, imprescriptibles, inembargables e inalienables.

EL AUTOR/ESTUDIANTE: Sebastián Ramos Díaz



Firma:_____

EL AUTOR/ESTUDIANTE: Javier Madden Molina Tique




Firma:_____

**TÍTULO COMPLETO DEL TRABAJO:** Diseño e implementación de prototipo de traductor móvil de lenguaje oral a lengua de señas mediante procesamiento de voz.

**AUTOR O AUTORES:**

| Primero y Segundo Apellido | Primero y Segundo Nombre |
|---|---|
| Ramos Díaz<br>Molina Tique | Sebastián<br>Javier Madden |

**DIRECTOR Y CODIRECTOR TESIS:**

| Primero y Segundo Apellido | Primero y Segundo Nombre |
|---|---|
| Bravo Obando | Martín Diomedes |

**ASESOR (ES):**

| Primero y Segundo Apellido | Primero y Segundo Nombre |
|---|---|
| | |

**PARA OPTAR AL TÍTULO DE:** Ingeniero Electrónico

**FACULTAD**: Ingeniería

**PROGRAMA O POSGRADO**: Ingeniería Electrónica

**CIUDAD:** Neiva **AÑO DE PRESENTACIÓN:** 2017 **NÚMERO DE PÁGINAS:** 83

**TIPO DE ILUSTRACIONES** (Marcar con una **X**)**:**

Diagramas_X__ Fotografías_X__ Grabaciones en discos___ Ilustraciones en general___ Grabados___
Láminas___ Litografías___ Mapas___ Música impresa___ Planos___ Retratos___ Sin ilustraciones___
Tablas o Cuadros_X_

**SOFTWARE** requerido y/o especializado para la lectura del documento: PDF

**MATERIAL ANEXO:**

**PREMIO O DISTINCIÓN** *(En caso de ser LAUREADAS o Meritoria)*:

**PALABRAS CLAVES EN ESPAÑOL E INGLÉS:**

| Español | Inglés | Español | Inglés |
|---|---|---|---|
| **1.** Traductor de voz a señas | - Speech to sign translator | **6.** _____ | _____ |
| **2.** Coeficientes de predicción lineal | - Linear Prediction Coefficients | **7.** _____ | _____ |
| **3.** Comunicación oral y visual | Oral and visual communication | **8.** _____ | _____ |
| **4.** _____ | _____ | **9.** _____ | _____ |
| **5.** _____ | _____ | **10.** _____ | _____ |

**RESUMEN DEL CONTENIDO:** (Máximo 250 palabras)


El sistema traductor de lenguaje oral a lengua de señas, permite a una persona mencionar un conjunto de palabras de tal modo que su simbolización en señas sea mostrada en una pantalla. Esto ayudará a cualquier hablante a comunicarse con una persona que se encuentre en condiciones de sordera. Para el desarrollo de este tipo de dispositivos se realizan las investigaciones previas concernientes al tema de coeficientes de predicción lineal aplicado al reconocimiento de voz, utilizando como lenguaje de programación Python para implementar los algoritmos que realicen la adquisición, procesamiento, extracción de características y posteriormente el reconocimiento y visualización. Finalmente se muestra la implementación del sistema y los resultados obtenidos.

**ABSTRACT:** (Máximo 250 palabras)

The translating system of oral language to sign language, allows a person to mention a set of words in such a way that their symbolization in signs is shown on a screen. This will help any speaker to communicate with a person who is deaf. For the development of this type of devices, previous research is carried out concerning the topic of linear prediction coefficients applied to speech recognition using Python programming language to implement the algorithms that perform acquisition, processing, extraction of features and later recognition and visualization. Finally the implementation of the system and the results obtained are shown.

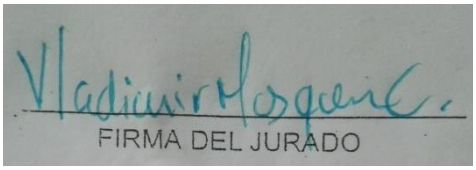

**APROBACION DE LA TESIS**

Nombre Presidente Jurado:

Firma:

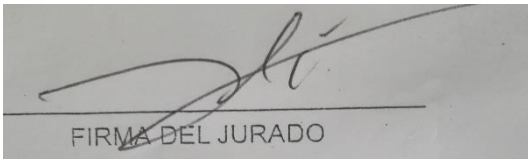
Nombre Jurado: Vladimir Mosquera Cerquera

Firma:




Nombre Jurado: Julián Adolfo Ramírez Gutiérrez

Firma:



# DISEÑO E IMPLEMENTACIÓN DE PROTOTIPO DE TRADUCTOR MÓVIL DE LENGUAJE ORAL A LENGUA DE SEÑAS MEDIANTE PROCESAMIENTO DE VOZ

SEBASTIÁN RAMOS DÍAZ
JAVIER MADDEN MOLINA TIQUE

UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
NEIVA
2017

# DISEÑO E IMPLEMENTACIÓN DE PROTOTIPO DE TRADUCTOR MÓVIL DE LENGUAJE ORAL A LENGUA DE SEÑAS MEDIANTE PROCESAMIENTO DE VOZ

SEBASTIÁN RAMOS DÍAZ
JAVIER MADDEN MOLINA TIQUE

**Trabajo de grado para optar al título de Ingeniero Electrónico**

**Director**
MARTIN DIOMEDES BRAVO OBANDO
**Ingeniero Electrónico y de Telecomunicaciones**

UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
NEIVA
2017

Nota de aceptación:

_____

_____

_____

_____

_____

_____

_____

Firma del presidente del Jurado

_____

Firma del Jurado

_____

Firma del Jurado

Neiva, Diciembre 1 de 2017

Nuestra recompensa se encuentra en el esfuerzo y no en el resultado. Un esfuerzo total es una victoria completa.

*Mahatma Gandhi*

## AGRADECIMIENTOS

Agradezco en primera instancia a Dios por regalarme la vida, por darme la oportunidad de culminar mis estudios como profesional, por concederme unos padres y hermanos ejemplares que me han guiado a través de mi camino como persona y por regalarme una nueva familia por la cual salir siempre adelante.

En segunda instancia, a mi compañero de tésis quien siempre puso todo el empeño por salir adelante con este proyecto y a nuestro director que tuvo la disposición de mostrarnos el camino para finalizar de la mejor manera este trabajo.

*Sebastián Ramos Díaz*

Mis agradecimientos van dirigidos a todas aquellas personas que con sus acciones, consejos, palabras de aliento, dedicación y devoción han esperado y hecho esto posible. Gracias por la paciencia y gran amor de parte de mi madre, familia y personas que han estado ahí, pero que no alcanzaria la hoja completa para nombrarlas; a todas les doy infinitas gracias.

*Javier Madden Molina Tique*

**CONTENIDO**

## LISTA DE TABLAS

## LISTA DE FIGURAS

## LISTA DE ANEXOS

## ACRÓNIMOS

**ASL:** American Sign Language.

**CAD:** Convertidor Análogo Digital.

**CDA:** Convertidor Digital Análogo.

**DC:** Direct Current.

**DFT:** Discrete Fourier Transform.

**DGS:** Deutsche Gebardensprache.

**DTW:** Dinamic Time Warping.

**FFT:** Fast Fourier Transform.

**HDMI:** High Definition Multimedia Interface.

**HIFI:** High Fidelity.

**IIR:** Infinite Impulse Response.

**INALSA:** Instituto Nacional de Lengua de Señas Argentina.

**INSOR:** Instituto Nacional de Sordos.

**IIR:** Infinite impulse Response.

**LCD:** Liquid Crystal Display.

**LPC:** Linear Predictive Coding.

**LSB:** Lengua de señas de Bolivia.

**LSC:** Lengua de señas Colombiana.

**LSCh:** Lengua de señas de Chile.

**LSD:** Lengua de señas Dominicana.

**LSE:** Lengua de señas de Ecuador ó España.

**LSF:** Lengua de señas de Francia.

**LSM:** Lengua de señas de México.

**LSN:** Lengua de señas de Nicaragua.

**LSP:** Lengua de señas de Panamá.

**LSP:** Lengua de señas de Perú.

**LSQ:** Lengua de señas de Québec.

**LLSS:** Lengua de señas.

**LSU:** Lengua de señas de Uruguay.

**LSV:** Lengua de señas de Venezuela.

**MB:** Mega Bytes.

**RAE:** Real Academia Española.

**RAM:** Random Access Memory.

**RNA:** Red Neuronal Artificial.

**SD:** Secure Digital.

**SSH:** Secure Shell.

**TFT:** Thin Film Transistor.

**USB:** Universal Serial Bus.

**WIFI:** Wireless Fidelity.

## RESUMEN

El sistema traductor de lenguaje oral a lengua de señas, permite a una persona mencionar un conjunto de palabras de tal modo que su simbolización en señas sea mostrada en una pantalla. Esto ayudará a cualquier hablante a comunicarse con una persona que se encuentre en condiciones de sordera. Para el desarrollo de este tipo de dispositivos se realizan las investigaciones previas concernientes al tema de coeficientes de predicción lineal aplicado al reconocimiento de voz, utilizando como lenguaje de programación python para implementar los algoritmos que realicen la adquisición, procesamiento, extracción de características y posteriormente el reconocimiento y visualización. Finalmente se muestra la implementación del sistema y los resultados obtenidos.



**PALABRAS CLAVE:**

Traductor de voz a señales, Coeficientes de predicción lineal, Comunicación oral y visual.

## ABSTRACT

The translating system of oral language to sign language, allows a person to mention a set of words in such a way that their symbolization in signs is shown on a screen. This will help any speaker to communicate with a person who is deaf. For the development of this type of devices, previous research is carried out concerning the topic of linear prediction coefficients applied to speech recognition using Python programming language to implement the algorithms that perform acquisition, processing, extraction of features and later recognition and visualization. Finally the implementation of the system and the results obtained are shown.



**KEYWORDS:**

Speech to signal translator, Linear Prediction Coefficients, Oral and visual communication.

# INTRODUCCIÓN

Son muchos los sistemas de procesamiento de voz que se utilizan hoy en día, ya que la eficiencia y efectividad de estos, han despertado gran interés en los investigadores que tratan de innovar en aplicaciones o etapas de otros sistemas que estén implementados, tales como seguridad, entre los más destacados, búsqueda de información, control por voz, etc.

La mayoría de estos sistemas son implementados en base a métodos convencionales como redes neuronales artificiales que viene de la sigla RNA, relacionado con el entrenamiento de datos para conseguir la identificación de las palabras.

El método utilizado en el presente proyecto, se basa en el uso del análisis por LPC (Linear Prediction Code) o en español codificación de predicción lineal, el cual se caracteriza por modelar el comportamiento del tracto vocal del ser humano por medio de un filtro digital todo polos. Para hallar los coeficientes de dicho filtro es necesario realizar procesos que se encarguen de extraer información o características de la señal de voz que permitan un mejor análisis de esta. Obteniendo los coeficientes deseados, es posible ejecutar algoritmos de comparación de vectores que permitan reconocer la señal. Teniendo en cuenta que el objetivo primordial del actual trabajo es traducir palabras a signos, es muy relevante contar con un sistema de visualización que despliegue la palabra que se ha mencionado en una forma simbólica. Desarrollando todo lo expresado de manera general anteriormente, es lo que hace posible implementar un dispositivo traductor de lenguaje oral a lengua de señas.

# 1. OBJETIVOS

## 1.1 OBJETIVO GENERAL

Diseñar un prototipo de traductor móvil capaz de adquirir y procesar la señal de voz con el fin de implementar un sistema traductor de lenguaje oral a lengua de señas, que permita una interacción básica entre una persona con discapacidad auditiva y un oyente común.

## 1.2 OBJETIVOS ESPECÍFICOS

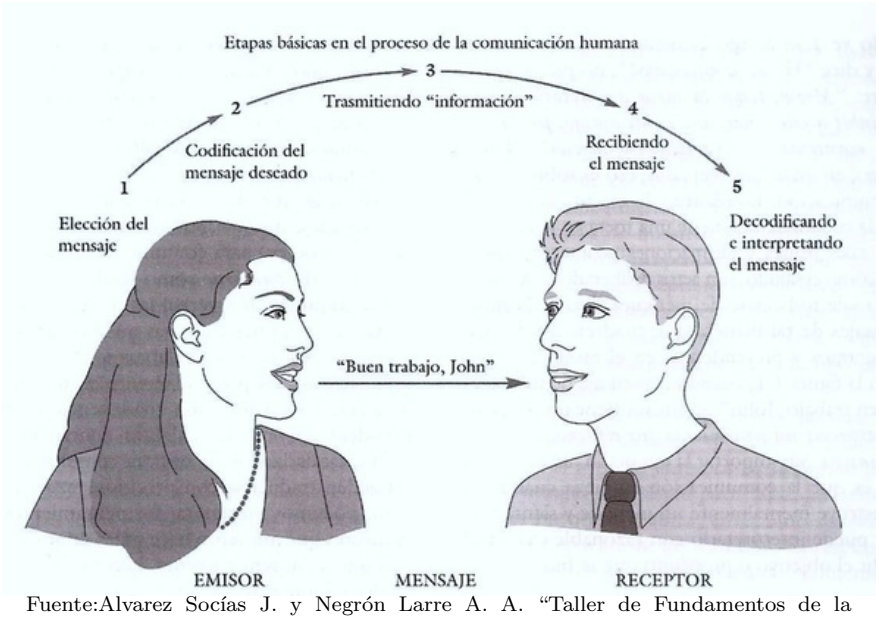
- Examinar a profundidad el método de predicción lineal enfocado al reconocimiento de los patrones de voz del sistema a implementar.
- Determinar la tecnología a utilizar en el desarrollo del proyecto, basada en las diferentes opciones de lenguaje de programación tales como C, C++ o Python.
- Construir un sistema de prueba que permita procesar y analizar señales de voz, que más adelante pueda ser acondicionado al diseño final de la herramienta.
- Almacenar un total de quince (15) comandos o palabras seleccionadas con su respectiva representación en lengua de señas al interior del sistema, de modo que se permita la consulta y respuesta del módulo de almacenamiento de datos en el momento que se requiera.
- Realizar pruebas de visualización de imágenes en la pantalla del sistema portable.
- Desarrollar un algoritmo que realice la adquisición de la señal de voz, procesamiento y visualización de la representación gráfica de las palabras almacenadas.
- Diseñar el modelo estructural del sistema portable.
- Validar la funcionalidad del sistema a través de pruebas con las personas directamente implicadas, analizando el impacto y beneficios de la implementación de este sistema.

# 2. MARCO TEÓRICO

## 2.1 COMUNICACIÓN Y LENGUAJE

Históricamente, desde la Antigua Grecia se han realizado intentos por generar voces artificiales. En muchos casos eran simplemente juegos de tuberías conectadas a un locutor humano, en otros auténticos ingenios acústicos capaces de producir sonoridades vocálicas, hasta el dìa de hoy en donde existe una estrecha relación en la comunicación hombre-máquina.[1]

Figura 1. Elementos de la comunicación




Fuente:Alvarez Socías J. y Negrón Larre A. A. "Taller de Fundamentos de la Comunicación". (2010). Instituto de Estudios Bancarios "Guillermo Subercaseaux"

En todo sistema de comunicación hay varios componentes: emisor, receptor, mensaje, código, canal y contexto. Es necesario conocer algunos aspectos de cada uno de ellos para poder integrar sistemas que funcionen de manera eficaz y eficiente. En el caso del ser humano, el emisor es el conjunto integrado por el cerebro que "piensa" el mensaje y el aparato fonatorio que lo "traduce" a una emisión acústica. El receptor es el aparato auditivo que recibe la onda sonora y la transforma en impulsos nerviosos que luego son interpretados por el cerebro; en nuestro caso hay dos receptores, un intermedio (sistema que hace parte del canal de comunicación) que es el que codifica la señal en imagen para que el sistema ocular se convierta en el receptor final. El mensaje es la idea a comunicar. El código es el lenguaje hablado. La combinación del mensaje y el código constituyen

[1] Miyara, Alberto. COMUNICACIÓN PERSONAL.(1999).

la señal. El canal puede ser el medio en el cual se propaga la onda sonora (en general el aire) o un medio de transmisión electrónico que constituye en sí mismo otro subsistema de comunicación cuyas propiedades son bien conocidas y que se aproxima en muchos casos (aunque no siempre) a la idealidad. [2] En la figura 1 se muestran los elementos principales que hacen parte del proceso de comunicación humana.

### 2.1.1 Comunicación Verbal

Para transmitir información utilizamos principalmente la palabra formando frases de estructura y longitud muy variada. Sobre el uso de las palabras podemos puntualizar que:

•Tienen diversas acepciones, que vienen dadas por el contenido de la frase de que forman parte.

•Evocan otras palabras, es decir llevan asociadas otras palabras.

•Tienen distinto valor según el contexto, una frase sacada de su contexto cambia el significado.

•Llevan un componente afectivo de grado y signo diferente según las circunstancias.

### 2.1.2 Comunicación No Verbal

Cuando emitimos un mensaje, la idea pensada se convierte en sonidos y en imágenes. Además de hablar, producimos una multitud de gestos faciales y corporales que podemos clasificar de la siguiente manera:

•La mirada humana puede adoptar formas muy diversas y expresar una amplia gama de sentimientos. (fugaces, sostenidas, luminosas, tristes, alegres etc.) El contacto visual es el primer paso para establecer una relación.

•Gestos faciales. El gesto facial por excelencia es la sonrisa, y todos somos capaces de diferenciar una sonrisa franca y espontánea de una mecánica y estudiada.

•Brazos y manos expresan diferentes mensajes. Brazos abiertos expresan acogida, cruzados, estar en espera etc.

•Postura corporal. También transmite un mensaje. La postura erguida expresa seguridad, el tronco hacia atrás transmite altivez, etc. Sin embargo hay que puntualizar que el significado de estas posturas depende en gran medida de las circunstancias, de la personalidad y de la cultura.

•La voz y el tono completan la comunicación no verbal porque tiene el poder de crear confianza, influir, sugestionar, persuadir,etc. [3] En la figura 1 se muestran los elementos

[2]Miyara, Alberto. COMUNICACIÓN PERSONAL.(1999).

[3]Alvarez Socías J. y Negrón Larre A. A. "Taller de Fundamentos de la Comunicación". (2010). Instituto de Estudios

principales que hacen parte del proceso de comunicación humana.

El lenguaje de señas se considera como comunicación no verbal teniendo en cuenta en que se utilizan señas o imágenes para interactuar con otra persona. Este tipo de lengua nace por la necesidad de comunicación entre personas que poseen implicaciones médicas en el aparato fonador, el cual es el encargado de producir los sonidos en el ser humano. Por este motivo vale la pena profundizar en el estudio del aparato fonador, para entender la forma en que se producen los sonidos que nos permiten hablar.

## 2.2 LENGUA DE SEÑAS

Esta comunicación se dio primero por gestos (manualmente) y luego con palabras. Se ha encontrado en la anatomía de humanoides de épocas prehistóricas específicas, que el espacio vocal de los primeros humanos no podría acomodar en su estructura el complejo aparato vocal con el que contamos ahora, sin embargo parece que la estructura esquelética de sus manos sugiere que éstas eran muy ágiles. En las pinturas rupestres, se ve a personas mostrando gestos, es decir intentando comunicarse a través de los mismos, en vez de hacerlo utilizando sonidos vocales.

A medida que el ser humano fue evolucionando, su necesidad de comunicarse también aumentaba. Lo que se empezó a buscar fue la comunicación sin necesidad de mantener contacto visual entre los hablantes. Probablemente, fue así como nació el uso de la voz.

En realidad no existe razón para creer que el uso de gestos excluya el habla o viceversa. Es probable que la gente haya usado ambas modalidades. Sin embargo, el uso de los gestos no alcanzó el nivel de lenguaje en términos formales, en cambio, quienes sí continuaron usando las señas, hasta la actualidad como lengua natural, son las personas sordas. Ellos son los usuarios de las señas como sistema lingüístico.[4]

### 2.2.1 Ámbito Mundial

Hay acerca de 70 millones de personas sordas en el mundo que usan la lengua de señas como su primera lengua o como lengua materna. De hecho, hay diferentes lenguas de señas por todo el mundo. Ethonogue, una enciclopedia que cataloga las 6909 lenguas vivas existentes en el mundo, ha listado en 130 las lenguas de señas usadas por las comunidades sordas; en tanto que la Universidad de Gallaudet constata la existencia de 271 lenguas de señas dispersas alrededor del mundo. [5]

A nivel de suramérica, actualmente la totalidad de paises cuentan con su reconocimiento político del uso de la lengua de señas como método natural de comunicación. En la figura 2, se puede observar un mapa en donde se describen las fechas en que cada uno

---

Bancarios “Guillermo Subercaseaux”

[4]Gracia Benavides I. “La lingüística en el lenguaje de señas”. (2004). www.cultura-sorda.org

[5]http://inalsa.cas.org.ar “Status lingüístico de la LLSS en el Mundo”. (2014).

de los países obtuvieron su reconocimiento. Argentina, hasta el 2015 no tenia este distintivo, ya que hasta en junio 2014 se iniciaron diligencias con proyectos de ley para obtener esta mención.

Figura 2. Mapa de reconocimiento de lengua de señas.


Fuente: http://inalsa.cas.org.ar/nuestra-lsa/la-ls-en-el-mundo.

### 2.2.2 Ámbito Colombiano

El Instituto Nacional para Sordos (INSOR) perteneciente al Ministerio de Educación Nacional de la República de Colombia, define a lengua de señas como la lengua natural de las personas sordas. Se basa en movimientos y expresiones a través de las manos, los ojos, el rostro, la boca y el cuerpo. Muchos sordos se comunican con esta lengua y requieren de un intérprete o persona que la maneje para relacionarse con oyentes que no la conocen. En Colombia se le llama Lengua de Señas Colombiana.[6]

[6] http://insor.gov.co/ninos/que-es-la-lengua-de-senas.

Poco se sabe sobre el origen de la lengua de señas en Colombia, ya que no existen suficientes testimonios escritos para precisar sus comienzos. Según Paulina Ramírez, sus orígenes se remontan a 1920, en un internado católico bogotano.

En 1957 aparece la primera asociación de sordos en Bogotá y un año después otra en Cali. Parece que estos sistemas de señas recibieron influencia de la lengua de señas española, a través de inmigrantes o de sordos colombianos educados en España, en los años 50. El español hablado y escrito también influyó, en esa misma época, en el auge de la educación oficial oralista.Posteriormente, en los 70, la presencia de misioneros protestantes de Estados Unidos y la formación de especialistas oyentes colombianos en ese país marcaron una fuerte influencia de la lengua de señas norteamericana en la colombiana. Diversas variedades fueron desarrollándose en el país, que se han regularizado con la formación de comunidades y asociaciones. Sin embargo, hay una base común y entendimiento entre todas estas.

El hecho de que el Gobierno Nacional hubiera decretado y reconocido, según la Ley número 324 de 1996, en su artículo 2°, la lengua de señas colombiana como propia de la comunidad sorda del país, no es más que una corroboración legal de lo que esta comunidad ya sabía y los investigadores que han trabajado el tema afirmaban hace muchos años: la lengua de señas es una lengua natural, con su propia gramática, sintaxis, vocabulario, usada por una comunidad específica.

Estos planteamientos tienen también grandes implicaciones educativas y sociales, porque conllevan el reconocimiento social de los usuarios de dicha lengua, ante los hablantes de otras lenguas. Por lo tanto, es una lengua que puede ser estudiada como cualquier otra y, además, todo esfuerzo, estudio e investigación que se hagan sobre esta forma de comunicación enriquecerá, reafirmará y realzará la importancia de la lengua de señas usada en Colombia. Su estudio contribuye también a su divulgación, y, lo más importante, al tener mayor comprensión lingüística sobre esta, se podrán mejorar los métodos de enseñanza, no solo para las personas con limitación auditiva, sino también para los familiares, amigos de estos, intérpretes y, en general, toda persona interesada en aprenderla. [7]

## 2.3 PROCESAMIENTO DIGITAL DE VOZ

Las vibraciones sonoras pueden ser representadas como señales electrónicas a través de algunos dispositivos (por ejemplo, un micrófono ), que convierte estas vibraciones en una señal de voltaje o tensión dependiente del tiempo. El resultado de la conversión se denomina señal analógica. Las señales analógicas son continuas en el sentido en que consisten en un continuo de valores.

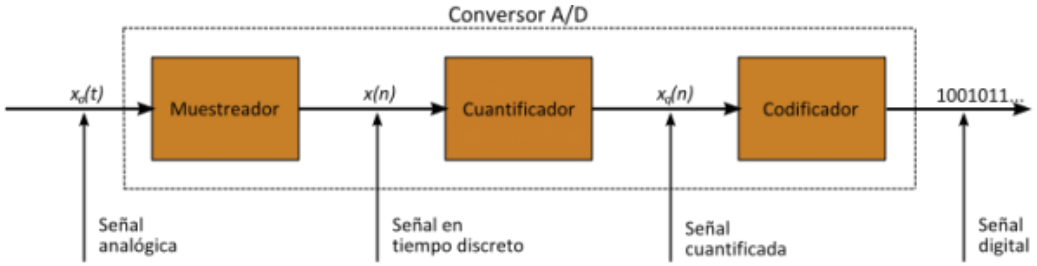
[7] Diccionario Básico de la Lengua de Señas Colombiana.

### 2.3.1 Conversión Analógica - Digital

Teniendo en cuenta que en la actualidad, la mayoria de los sistemas de comunicación o de procesamiento de señales se encuentran en el espacio digital, la señal de voz debe ser incluida en un proceso de conversión analógica a digital. Esto es posible gracias a dispositivos ya implementados conocidos como ADC (Analog Digital Converter) y DAC (Digital Analog Converter).

Al interior de los ADC, se tienen procesos importantes que merecen ser explicados con mayor profundidad, tales como se observa en el diagrama de bloques de la figura 3.

Figura 3. Pasos de conversión Analógica - Digital.




Fuente:https://www.ecured.cu/Conversión-analògico-digital

•**Muestreo:** El bloque de muestreo funciona midiendo la amplitud de la señal continua a intervalos de igual duración. Cada valor que se mide se denomina muestra (o sample) de la señal.

La distancia temporal o el intervalo de tiempo que hay entre dos muestras consecutivas se denomina período de muestreo, y se mide en segundos. Su inversa fm=1/Tm se denomina frecuencia de muestreo o sampling rate, y se mide en ciclos por segundo o Hz.

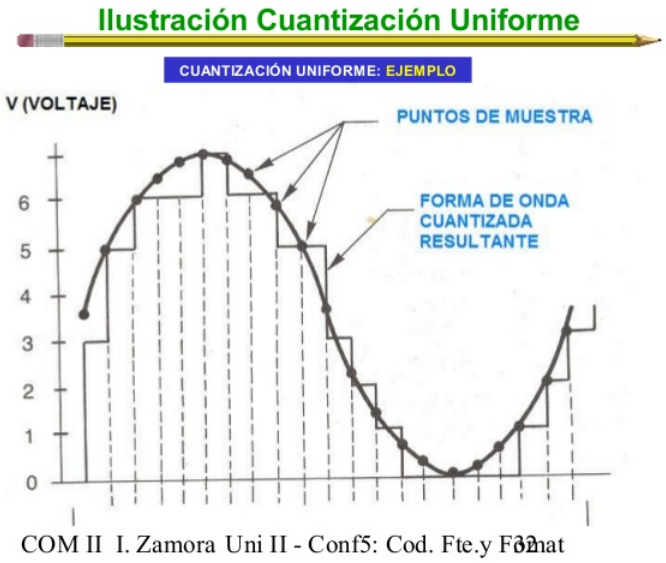
Por lo tanto, en el proceso de muestreo pasamos de una señal continua a un conjunto de muestras (es decir, puntos discretos en el tiempo).Es importante muestrear la señal lo suficientemente rápido como para capturar toda la información. El teorema de muestreo, o teorema de Nyquist, demuestra que para representar adecuadamente una sinusoide es necesario tener almenos dos muestras por cada ciclo de la sinusoide. Por tanto,para representar adecuadamente un sonido, la frecuencia de muestreo fm tiene que ser mayor, como mínimo, del doble de la frecuencia más alta contenida en la señal: fm >= 2*f(max). [8]

•**Cuantización:** Una vez la señal muestreada nos encontramos con un conjunto de muestras o de valores continuos de la amplitud de la señal. La cuantización se realiza al limitar los posibles valores de amplitud de una señal, definiendo una serie discreta

[8]Gomez Gutierrez E. , Digitalización del Sonido. (2009)

(no continua) de valores posibles. El número de posibles valores de amplitud viene determinado por la resolución del convertidor (CAD o CDA). La resolución de los convertidores depende del tamaño de la palabra que se utiliza para representar cada una de las muestras de la señal. La resolución de un convertidor se mide en número de bits de la palabra que utiliza, y un convertidor de nbits de resolución cuantizará a $2^n$ valores de la señal. [9]

Figura 4. Proceso de Cuantización.




Fuente:https://es.slideshare.net/nica2009/lecture-5-formateo-de-seales-analgicas

•**Codificación:** El objetivo fundamental de la codificación de voz es la conversión de la señal de voz a una secuencia binaria o representación digital. Dado el carácter analógico (señal continua en tiempo y amplitud) de la señal de voz, la codificación de voz conlleva un proceso básico de muestreo y cuantificación para conseguir una representación digital. Mediante el muestreo discretizamos la señal en tiempo y mediante la cuantificación discretizamos la señal en amplitud. Para que en este proceso de digitalización no exista pérdida de información, debemos muestrear la señal a una velocidad (fm) que como mínimo sea el doble de la frecuencia más alta presente en la señal que estamos discretizando.

En el proceso de discretización en amplitud debemos utilizar un número de bits por muestra (N) que resulte adecuado para la calidad deseada. Así, por ejemplo, una señal de voz con calidad telefónica tiene una frecuencia máxima de 4 kHz lo que supone una frecuencia de muestreo mínima de 8 kHz y se suele utilizar una representación con 8 bits por muestra (256 niveles de cuantificación) con una distribución logarítmica de los niveles de cuantificación a lo largo del margen dinámico de la señal, lo que supone una

[9] Gomez Gutierrez E. , Digitalización del Sonido. (2009)

velocidad de transmisión o necesidades de almacenamiento, en caso de grabación, de 64 kb/s.[10]

### 2.3.2 Preénfasis

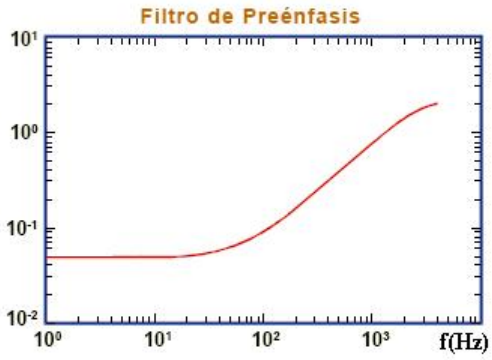
Para reducir el rango dinámico de las señales espectrales, se suele alisar el espectro para compensar los valores de las altas y de las bajas frecuencias mediante un filtro denominado Filtro de Preénfasis. Este proceso se realiza mediante una simple diferenciación:

$$s'(n) = s(n) - a * s(n-1) \tag{1}$$

donde $a$ refleja el grado de preénfasis, y suele estar en el rango de 0.9 a 1.0. Cuanto más cercano es al valor de 1.0, mayor es el efecto de preénfasis. En la siguiente figura se muestra la respuesta en frecuencia típica de un filtro de preénfasis.

Idealmente, el preénfasis sólo se debe aplicar a señales sonoras. Sin embargo, por la pequeña distorsión que se introduce en las señales aperiódicas, y por simplificar el sistema de análisis, la práctica totalidad de los sistemas actuales aplican el preénfasis a todo tipo de señales. [11] Por este motivo, el filtro preénfasis se aplica antes de realizar la segmetación, ya que no causa problemas en el procesamiento como tal.

Figura 5. Respuesta en frecuencia de filtro preénfasis.




Fuente:Galindo Riaño P., Introducciòn al reconocimiento de voz, (1996)

[10] http://physionet.cps.unizar.es/ eduardo/investigacion/voz/coder.html

[11] Galindo Riaño P., Introducciòn al reconocimiento de voz. (1996)

### 2.3.3 Segmetación en tramas

Para poder realizar el procesamiento de la voz, se realiza la suposición de que las propiedades de esta señal cambian relativamente despacio en el tiempo. Esta suposición lleva a multitud de métodos de procesamiento en la que breves segmentos de señal se aislan y procesan de forma independiente, como si fueran fragmentos de un sonido contínuo. A estos segmentos de longitud fija en los que se suele dividir la señal los denominamos tramas.

Los resultados de procesar cada trama generan un número (por ejemplo, podría ser el índice del fonema detectado) o bien varios números (por ejemplo, si el sonido es sonoro/sordo, la amplitud máxima de la señal, características de la misma, etc.). A este proceso se le denomina *Extracción de Características*. Este proceso de aislar una trama y calcular una serie de características se realiza cada cierto tiempo, denominado Longitud del Desplazamiento (o Solapamiento).[12]

La razón fundamental para realizar el estudio de la señal de voz por medio de segmentación, es que a pequeños trozos de señal, esta se comporta como cuasi-estacionaria, del modo en que se puedan extraer sus carácteristicas y analizar libremente.

Un aspecto importante a tener en cuenta es si se va a incluir o no un solapamiento entre las tramas, debido a que en muchos estudios no hay concordancia definida entre un metodo y el otro. Para esto hay que pensar en los casos en donde se relacionan la longitud de la trama (N) y el desplazamiento de la misma (m).

**N<M:** En este caso no hay solapamiento entre tramas sucesivas, perdiéndose parte de la señal. Por ello, no se suele utilizar.

**N=M**:Si bien no hay pérdida de señal, la inexistencia de correlación en los valores espectrales obtenidos de tramas consecutivas suele ser desaconsejable.

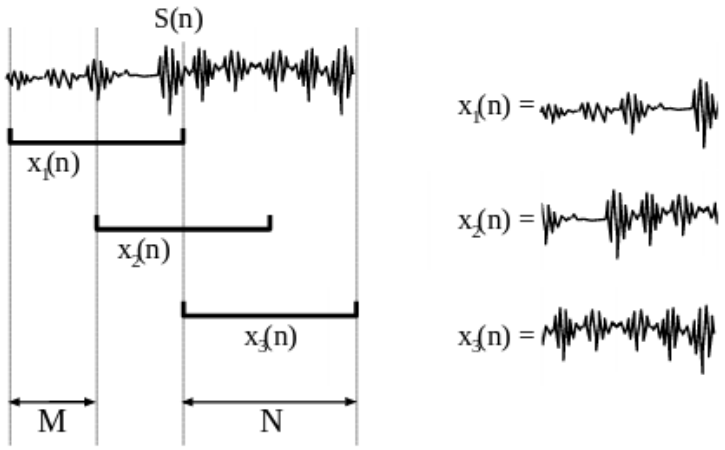
**N>M:** Es el caso más habitual. Las tramas adyacentes solapan, por lo que el análisis espectral posterior tendrá una cierta correlación entre tramas consecutivas. De hecho, si **N>>M**, las variaciones entre los valores obtenidos de sucesivos análisis espectrales son muy suaves.[13]

---

[12] Galindo Riaño P., Introducciòn al reconocimiento de voz. (1996)

[13] Galindo Riaño P., Introducción al reconocimiento de voz. (1996)

Figura 6. Solapamientro entre tramas.




Fuente:Galindo Riaño P., Introducción al reconocimiento de voz, (1996)

Una vez establecida la relación entre M y N, será preciso determinar los correspondientes valores absolutos. La selección del valor de N resulta de un compromiso que es preciso establecer entre resolución en tiempo y en frecuencia. Normalmente, los espectrogramas de banda ancha usan ventanas de una duración de unos 3 msec., lo que permite una buena resolución temporal, útil para analizar cambios rápidos de la señal. Por otro lado, los espectrogramas de banda estrecha usan ventanas del orden de 30 msec., muy útiles para determinar el período de la señal, ya la ventana suele comprender varios períodos.

Un valor óptimo de N sería el correspondiente al período de la señal. Debido a que dicho período no solo es variable de un locutor a otro, sino que es variable en el tiempo para un cierto locutor, no es posible utilizar tal medida como valor para la longitud de la trama. Un valor de N menor que el período de la señal produciría un análisis distorsionado, ya que la porción de señal analizada está truncada de forma artificial. Por ello, se suelen utilizar valores para N que comprendan varios períodos de señal (los menos posibles). Dado que los posibles valores del período de la señal varían aproximadamente entre los 40Hz. (tono grave de un hombre) y los 100 Hz.(tono agudo de mujer), un compromiso para la selección de N es[14]:

$$10 \text{ msec} < \text{Longitud de Trama} < 25 \text{ msec}$$

Cabe aclarar que el valor del desplazamiento debe ser menor al valor de la longitud de la trama, ya que entre menor sea el desplazamiento mas suaves seràn las fluctuaciones en los valores que se obtengan luego de la extracciòn de características.

[14]Galindo Riaño P., Introducciòn al reconocimiento de voz. (1996)

### 2.3.4 Ventaneo

Cuando una secuencia de muestras a procesar digitalmente, es demasiado grande, tal procesamiento puede sobrepasar las capacidades del equipo de cómputo disponible. Así entonces se toman pequeños espacios muestrales y a cada uno se le aplica el mismo procesamiento. Las ventanas se usan entonces para evitar las discontinuidades al principio y al final de los bloques analizados. Sin el uso de las ventanas, el resultado de unir las secuencias procesadas implica nuevas componentes espectrales: artefactos en imágenes y ruido de alta frecuencia o gis (sonido no nítido y con ruido constante) en audio. Las ventanas también son usadas para la decimación y la interpolación. La utilización de una ventana cambia el espectro en frecuencia de la señal.[15]

**Ventana Hamming:** Esta ventana tiene un comportamiento temporal de medio ciclo de una señal cosenoidal y normalizado en amplitud a la unidad. Esta ventana de caracteriza por el argumento N impar. Esta ventana es utilizada ya que ayuda a minimizar o atenuar la distorsion causada por las discontinuidades. Su ecuación es la siguiente y genera N muestras: a0=0.53836 ; a1=0.46164

$$w_{Hamming} =; N(n) = \sum_{l=0}^{N-1} [a_0 - a_1 cos((2\pi l)/N - 1)]\delta(n-l) \tag{2}$$

### 2.3.5 Segmentos Sonoros y Sordos

La extracción de características se basa en el cálculo de factores como energía, cruces por cero, pitch u otros valores que permitan el análisis mas detallado de la señal de voz, como por ejemplo en la identificación de sonidos sonoros y sordos. Este proceso se realiza con el fin de evitar sobreprocesamiento de los datos, ya que es completamente innecesario tener en cuenta sonidos sordos, ya que la información relevante de la señal se encuentra en los sonidos sonoros.

**Energía y Magnitud:**

Estos dos parámetros son realmente útiles para distinguir segmentos sonoros y sordos en una señal de voz, ya que los valores de estos aumentan en las partes sonoras con respecto a las sordas. Esto quiere decir que observando las funciones de energía de las palabras, en ellas se puede observar claramente que los valores de mayor energía se representan en los segmentos vocálicos de la señal, mientras que en las consonantes ocurre lo contrario. [16]

---

[15] Galindo Riaño P., Introducciòn al reconocimiento de voz. (1996)

[16] Bernal Bermudez J. , Bobadilla Sancho J. , Gòmez Vilda P. , Reconocimiento de voz y Fonética Acústica. (2000)

$$\text{Formulaciòn de la magnitud} : M(n) = 1/N \sum_{m=0}^{N-1} |x(m)| * w(n-m)$$

$$\text{Formulaciòn de la energìa} : E(n) = 1/N \sum_{m=0}^{N-1} x(m)^2 w(n-m)$$

La razón principal por la cual existe mayor energía en los segmentos sonoros, es porque el aire encuentra menos obstáculos para salir de los organos articulatorios.

### 2.3.6 Cruces por cero y máximos

Como su nombre lo indica, el índice de cruces por cero proveen la cantidad de veces en que una señal continua toma los valores de cero. Cuando se tienen señales discretas, un cruce por cero se observa cuando dos muestras seguidas o consecutivas cambian de signo o toman un valor nulo.

Generalmente, las señales con mayor frecuencia son las que tienen mayor índice de cruces por cero; por ejemplo la señal de ruido. Es por este motivo, que este parámentro se tiene en cuenta para el análisis de segmentos sonoros, ya que en ellos, existe una pequeña cantidad de cruces por cero.

### 2.3.7 Análisis LPC

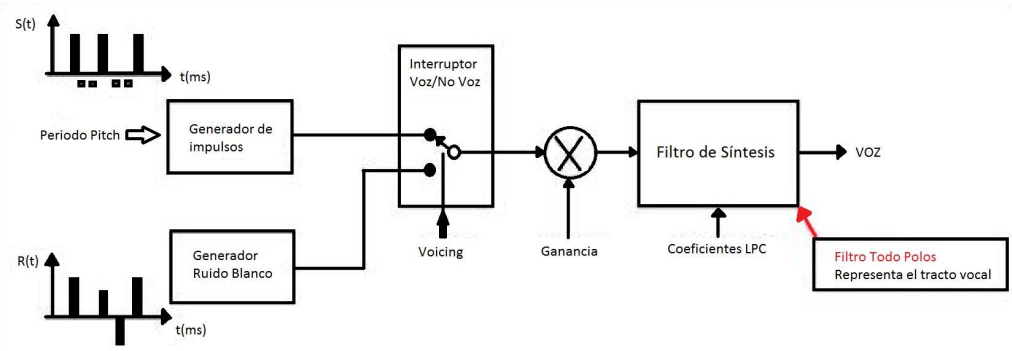
Hablando de sistemas de reconocimiento de voz basados en análisis LPC (Linear Predictive Coding), luego de que se ha realizado la identificación de segmentos sonoros, se pasa a encontrar los coeficientes de predicción lineal los cuales son útiles en la etapa de identificación de palabras.

El método LPC es considerado uno de los mas eficientes y usados en sistemas de codificación, ya que este representa de una manera muy acertada el comportamiento del tracto vocal al momento de habla. Esta representación viene dada en un filtro digital todo polos, en donde su entrada proviene de la señal luego de realizado el anterior procesamiento y su salida es predecida teniendo en cuenta los valores de muestras pasadas. La función de transferencia del filtro es la siguiente:

$$H(z) = \frac{G}{1 + a_1 * z^{-1} + a_2 * z^{-2} + ... + a_p * z^{-p}} \tag{3}$$

La figura 7 muestra el diagrama de bloques de un sistema basado en LPC.

Figura 7. Diagrama de bloques modelo LPC.




Fuente:García A., Gallego E., Domínguez E., Correa E., Rodríguez J., Universidad de Pinar del Río Cuba, Codificación de voz mediante coeficientes de predicción lineal (LPC ) sobre Microblaze, (2011)

De igual manera, el modelo del tracto vocal también se puede representar matemáticamente por medio de una ecuación en diferencias:

$$S(n) = \sum_{k=1}^{p} s(n-k)a_k + Gu(n) \tag{4}$$

El término de la sumatoria en la ecuación (4), puede interpretarse como un predictor lineal en el cual la muestra de salida s(n) se puede generar mediante una combinación lineal de p muestras anteriores. También se puede calcular el error de predicción e(n) como:

$$e(n) = s(n) - s'(n) = Gu(n) \tag{5}$$

El problema fundamental de la predicción lineal de la voz es la determinación del conjunto de coeficientes para cada trama que haga mínimo el error de predicción e(n) medio cuadrático. Una vez determinados los coeficientes para esa trama, si se aplica la excitación adecuada al sistema determinado por la ecuación (3), se obtendrá a la salida una secuencia que reproducirá la trama en cuestión no exactamente en el dominio del tiempo pero si de sus propiedades espectrales.[17]

Entre las ventajas que presenta el método de predicción lineal se encuentran:

•LPC proporciona un modelo adecuado de la señal de voz y sus parámetros se ajustan a las características del tracto vocal, especialmente en los sonidos sonoros del habla cuyas

[17]García A., Gallego E., Domínguez E., Correa E., Rodríguez J., Universidad de Pinar del Río Cuba, Codificación de voz mediante coeficientes de predicción lineal (LPC ) sobre Microblaze, (2011)

propiedades se aproximan más a la señal estacionaria que en los sonidos sordos.

- Los parámetros obtenidos mediante predicción lineal muestran un espectro suavizado que proporciona la información mas representativa de la voz. Esto evita perderse en los detalles ofrecidos por la transformada de Fourier.

- LPC es un método preciso, muy adecuado para computación, tanto por su sencillez como por la rapidez de ejecución que presentan algunos de los algoritmos hallados.

- Las pruebas realizadas con este método muestran buenos resultados en diversos campos del tratamiento automático de la voz.

**Cálculo de coeficientes óptimos LPC**

Existen dos métodos clásicos para la resolución: el método de la autocorrelación y el método de la covarianza. El primero de ellos permite una resolución recursiva que exige poca carga computacional, por lo que la gran mayoria de los sistemas de reconocimiento lo utilizan.

Partiendo de la suposición de que la señal de voz $S_n(m)$ se anula en el intervalo $0 \leq m \leq N-1$, esto equivale a asumir que la señal $S_n(m)$ ha sido multiplicada en el tiempo por una ventana w(m) que vale cero fuera del intervalo $0 \leq m \leq N-1$. De este modo, se puede expresar la señal de voz como:

$$S_n(m) = \begin{cases} s(n+m) \cdot w(m) & 0 \leq m \leq N-1 \\ 0 & resto \end{cases} \tag{6}$$

Así, el error cuadrático medio será distinto de cero en el intervalo $0 \leq m \leq N-1+p$, por lo que podremos expresarlo como:

$$E(n) = \sum_{m=0}^{N-1+p} e_n^2(m) \tag{7}$$

La expresión de la covarianza localizada quedará:

$$\phi_n(i,k) = \sum_{m=0}^{N-1+p} s_n(m-i) \cdot s_n(m-k) \begin{cases} 1 \leq i \leq p \\ 0 \leq k \leq p \end{cases} \tag{8}$$

o también,

$$\phi_n(i,k) = \sum_{m=0}^{N-1-(i-k)} s_n(m) \cdot (m+i-k) \begin{cases} 1 \leq i \leq p \\ 0 \leq k \leq p \end{cases} \tag{9}$$

La ecuación (9) es solo en función de i-k, en lugar de serlo de las variables independientes i y k, por lo que $\phi_n(i,k)$ se reduce sencillamente a la expresión de la función de autocorrelación de la ecuación (10).

$$\phi_n(i,k) = r_n(i-k) = \sum_{m=0}^{N-1-(i-k)} s_n(m) \cdot (m+i-k) \begin{cases} 1 \leq i \leq p \\ 0 \leq k \leq p \end{cases} \tag{10}$$

Debido a que la función de autocorrelación es simétrica por $r_n(-k) = r_n(k)$, las ecuaciones se pueden expresar tal y como se observa en la ecuaciòn (11).

$$\sum_{k=1}^{p} r_n(|i-k|) \cdot \hat{a}_k = r_n(i) 1 \leq i \leq p \tag{11}$$

Esta ecuación puede ser representada en forma matricial de la siguiente manera:

$$\begin{vmatrix} r_n(0) & r_n(1) & r_n(2) & \cdots & r_n(p-1) \\ r_n(1) & r_n(0) & r_n(1) & \cdots & r_n(p-2) \\ r_n(2) & r_n(1) & r_n(0) & \cdots & r_n(p-3) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ r_n(p-1) & r_n(p-2) & r_n(p-3) & \cdots & r_n(0) \end{vmatrix} \cdot \begin{vmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \hat{a}_3 \\ \cdots \\ \hat{a}_p \end{vmatrix} = \begin{vmatrix} r_n(1) \\ r_n(2) \\ r_n(3) \\ \cdots \\ r_n(p) \end{vmatrix}$$

La matriz de autocorrelaciones, de dimensión pxp, es una matriz tipo Toeplitz (simétrica, con las diagonales principales y secundarias de elementos iguales). Para la resolución de este tipo de matriz, se hace necesario el uso de algoritmos computacionales como lo es, LEVINSON DURBIN.

**Algoritmo Levinson Durbin:**

Debido al tipo de matriz que resulta luego del análisis, las ecuaciones se pueden resolver de forma recursiva mediante el método de Durbin.

Se inicializa teniendo en cuenta que: $E_0 = r_n(0); a_0(i) = 0$ para todo $i$

Resolviendo de forma iterativa las ecuaciones para m=1,2,...p. se tiene:

$$K_m = \frac{r(m) - \sum_{i=1}^{m-1} a_{m-1}(i) \cdot r_n(m-i)}{E_{m-1}} \tag{12}$$

$a_m(m) = k_m$

$a_m(j) = a_{m-1}(j) - k_m \cdot a_{m-1}(m-j) 1 \leq j \leq m-1$

Las soluciones parciales para m<p, permitiran calcular los coeficientes óptimos del filtro H(z) de orden m. La solución final que se busca para m=p, dará como resultado los

coeficientes del filtro H(z) de orden p. [18]

Este proceso se debe realizar a cada uno de los segmentos que se han enventanado en la señal de voz, de modo en que estos sean comparados con los nuevos coeficientes para obtener un reconocimiento de la señal de voz actual.

### 2.3.8 Almacenamiento de datos

Debido a que se trata de un sistema en el cual se debe reconocer la señal de voz actual, se deben tener patrones de reconocimiento bien establecidos por medio de almacenamiento de coeficientes LPC de distintas señales previamente analizadas, con el objetivo de comparar los coeficientes de la señal de voz actual, con cada uno de los coeficientes de las señales de voz almacenadas con anterioridad.

Del número de señales de voz almacenadas por cada comando, depende la calidad del patrón a establecer para un mejor reconocmiento de la señal de voz. Para esto es necesario contar con varias personas con diferentes tonalidades de voz con el objetivo de ampliar el rango de estudio de cada comando.

Para efectuar el reconocimiento de la palabra, es necesario la utilización de herramientas de comparación o identificación eficaces que ayuden a mejorar el rendimiento del sistema. Una de estas herramientas se trata de el uso de algoritmo DTW (Dinamic Time Warping).

### 2.3.9 Algoritmo DTW

Suponiendo que se tienen dos series o secuencias temporales Q y C, de longitud $m$ y $n$ respectivamente:

$$Q = q_1, q_2, ..., q_i, ..., q_n \qquad C = c_1, c_2, ..., c_j, ..., c_m \tag{13}$$

Para alinear dos secuencias usando DTW, se construye una matriz $nxm$ donde el elemento (i,j) de la matriz, contiene la distancia $\mathrm{d}(q_i, c_j)$ entre los dos puntos $q_i$ y $c_j$.(Normalmente se utiliza la distancia euclídea, por lo que $\mathrm{d}(q_i, c_j) = \mathrm{d}(q_i - c_j)^2$ ). Cada elemento de matriz (i,j) corresponde a la alineación entre los puntos $q_i$ y $c_j$. Esto se ilustra en la Figura 8.

Una trayectoria de deformación W, es un conjunto contiguo de elementos de matriz que define una correlación entre Q y C. El elemento k de W se define como $w_k = (\mathrm{i,j})_K$ así tenemos:

$$W = w_1, w_2, ..., w_k, ..., w_K \qquad max(m, n) \leq K < m + n - 1 \tag{14}$$

[18] http://arantxa.ii.uam.es/ jortega/Tema3_TAPS_resto.pdf Universidad Politécnica de Madrid

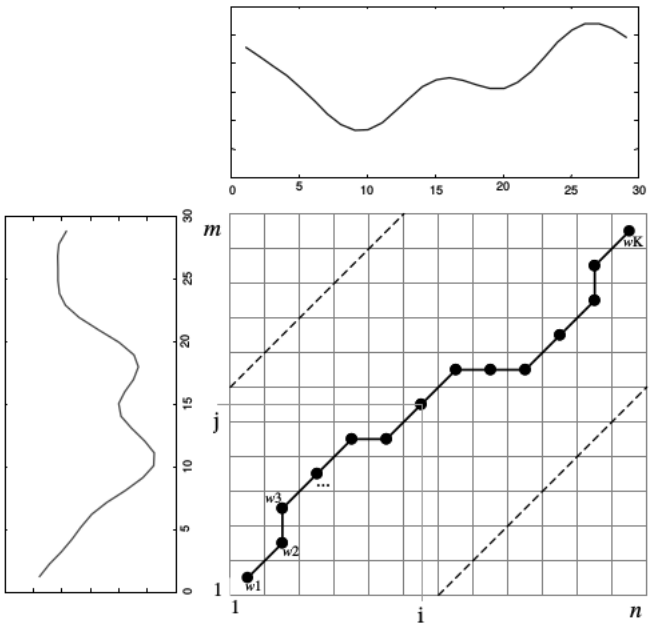
Para encontrar el menor camino de deformaciòn se tiene la siguiente ecuación:

$$DTW(Q,C) = min\left\{\frac{\sqrt{\sum_{k=1}^{k} w_k}}{K}\right\} \tag{15}$$

El K en el denominador se utiliza para compensar el hecho de que los caminos de deformación pueden tener diferentes longitudes. Este camino se puede encontrar muy eficientemente utilizando la programación dinámica para evaluar la siguiente recurrencia que define la distancia acumulada $\gamma(i,j)$ como la distancia d(i, j) que se encuentra en la celda actual y el mínimo de las distancias acumuladas de los elementos: [19]

$$\gamma(i,j) = d(q_i,c_j) + min\gamma(i-1,j-1),\gamma(i-1,j),\gamma(i,j-1) \tag{16}$$

Figura 8. Camino de deformación.



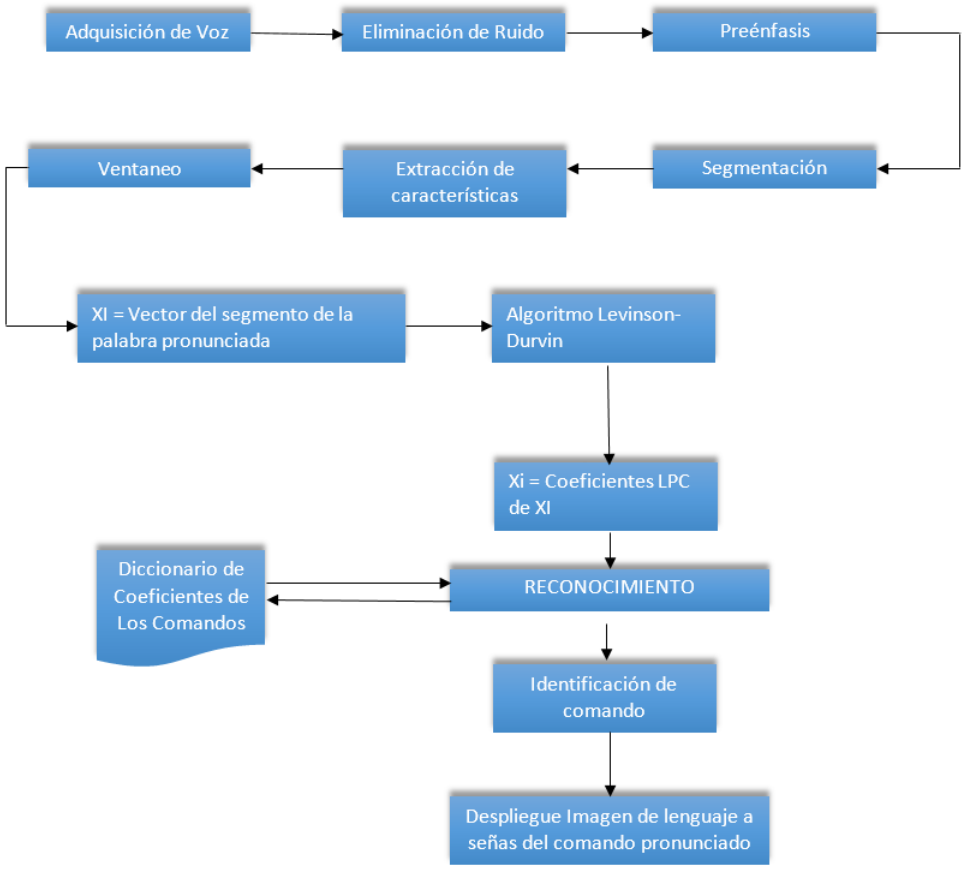

Fuente:Eamonn J. Keogh and Michael J. Pazzani, Derivative Dynamic Time Warping, (2001)

[19] Eamonn J. Keogh and Michael J. Pazzani, Derivative Dynamic Time Warping, (2001)

# 3. DISEÑO DEL SISTEMA

La gran mayoria de los proyectos que hacen parte del estado del arte trabajaron en aplicaciones en teléfonos celulares o tablets, por este motivo se pensó en la idea de fabricar un prototipo similar de modo portatil, cómodo al momento de cargarlo y con un método de análisis por predicción lineal el cual no es muy convencional en este tipo de trabajos, ya que la mayoria de estos se inclinan por utilizar tratamiento por redes neuronales.

Figura 9. Diagrama de bloques del sistema.




Fuente:Autores

Con anterioridad se decide utilizar lenguaje de programación PYTHON, ya que este es flexible, ordenado y con una comunidad de programadores ámplia. Ademas de esto, es un leguaje rápido y que simplifica mucho la parte de programación ya que no es necesario declarar tantos detalles.

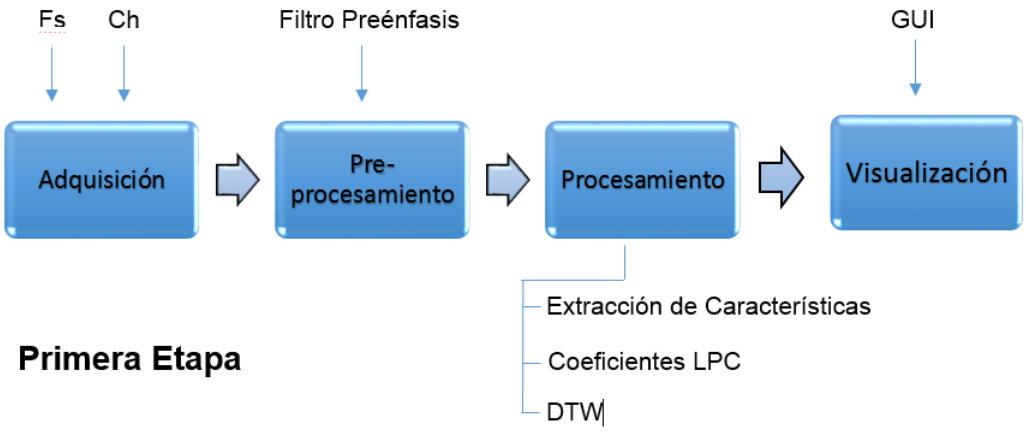
Para el diseño del sistema se desarolla un diagrama de bloques completo, en donde se pueda observar cada uno de los procesos que se ejecutan al interior del sistema. Este va desde la adquisición de la señal, hasta la visualización del signo en la pantalla. La figura 9 muestra el diagrama de bloques del sistema.

Teniendo en cuenta este diagrama de bloques, se piensa en la manera de ejecutar dichos procesos en dos etapas; una inicial en software en la que se desarrollan todas las tareas y una segunda parte que consta de análisis de hardware necesario para la implementación del dispositivo.

## 3.1 PRIMERA ETAPA

Esta primera etapa consta de un software de prueba en el cual se realiza todo el procesamiento necesario para que la palabra que se mencione, sea reconocida entre los 15 comandos establecidos. Cabe mencionar que en esta etapa van a estar relacionados todos los procesos que se observan en el diagrama de bloques anterior; sin embargo en la figura 10 se muestra un diagrama de bloques resumido para esta primera etapa.

Figura 10. Cuadro de diseño etapa software.




Fuente:Autores

### 3.1.1 Palabras y Adquisición de señal

Como punto de partida se deben fijar las palabras a reconocer para su posterior grabación y entrenamiento. Aqui se deben tener en cuenta parámetros importantes como muestrear a una frecuencia (fs) como mínimo al doble de la frecuencia de señal, la duración, los canales (Ch), entre otras. Además en analizar palabras que por sus fonemas no representen comportamiento similar al ruido blanco, tal y como lo son las palabras que llevan varias veces la letra S.

Para escoger la frecuencia de muestreo se tuvo que analizar el funcionamiento del hardware, por lo tanto en esta instancia se tuvo que ir realizando un diseño de la segunda etapa. Esto se menciona debido a que dispositivos como la tarjeta de sonido maneja una frecuencia de muestreo fija a unos 44100 Hz, por este motivo se decide definir este valor para la adquisición de la señal.

Para la grabación de las señales de voz, se opto por manejar dos canales (stereo) debido a que este mejoraba la calidad de la señal, teniendo en cuenta el micrófono que se estaba usando en dicho instante. Sin embargo, al momento de utilizar la tarjeta raspberry se tuvo que pasar el modelo de adquisición a un canal.

En cuanto a la duración de las grabaciónes se dictaminó que lo mejor sería grabar en 2 segundos debido a que un solo segundo resultaba muy limitado en palabras de varias sílabas, ya que podría cortarse la palabra al momento de grabar.

En esta etapa se realiza la instauración de la base de datos del sistema, el cual contiene alrededor de 30 grabaciones de señales de voz por cada uno de los 15 comandos. Esto se realiza por persona, por tal motivo la base de datos es de un tamaño considerable.

### 3.1.2 Filtrado y preprocesamiento

Es un hecho de que las señales de voz antes de procesar deben tener un tratamiento de filtrado de ruido y de compensación, por lo tanto se debe diseñar un filtro conocido como preénfasis que realza o potencializa las frecuencias altas de la señal. El valor del coeficiente **a1** del filtro debe estar entre 0.9 y 1, siendo este el valor ideal en el diseño del filtro. El valor del coeficiente utilizado es a1=0.95.

Luego de la limpieza de la señal, esta se segmenta con el fin de poder analizar y extraer mas adelante sus características sin problemas. Este proceso es necesario debido a que el comportamiento de la señal completa es no estacionaria, por lo tanto es bastante complicado su análsis. Segmentando la señal en tramas entre 20 y 30 ms con solapamiento de 10 ms, es posible reconocer la señal como cuasi-estacionaria. El solapamiento se realiza con el objetivo de no tener perdidas de información entre los extremos de las tramas.

Luego de que se ha realizado el entramado, se debe pensar en ejecutar una función de ventana en cada una de estas, por el motivo de atenuar el rizado en las zonas de transición y de atenuar distorsiones causadas por discontinuidades. Para ello se decide usar la ventana hamming, muy reconocida y usada según el estado del arte en este tipo de sistemas. Cabe mencionar que de acuerdo al tamaño de las tramas, es que se debe fijar el tamaño de la ventana.

### 3.1.3 Procesamiento

Siguiendo la teoría, para poder reconocer una palabra por medio del método LPC, se deben extraer algunas características que permitan minimizar procesos y analizar de mejor manera la señal de voz. Entre las características más importantes se encuentran la energía y tasa de cruces por cero que sirven para la identificación de segmentos sonoros y sordos. Los segmentos sordos se caracterizan principalmente por no contener

información relevante para el reconocimiento de voz, ya que su comportamiento se asemeja al ruido, por este motivo, es que se desechan para que estos no sean procesados al momento de hallar los coeficientes LPC.

**Extracción de características**

Para hallar los valores de energía y de cruces por cero, es necesario aplicar algoritmos matemáticos que se ejecuten en cada uno de los segmentos en que se ha dividido la señal de voz. Estos valores clasifican a una trama si es sonora o sorda, ya que la energía depende en gran parte de la magnitud de la amplitud y es de saber que en la señales de voz, la amplitud del ruido es muy baja en comparación a la amplitud de la señal donde hay actividad vocal. En cuanto a la tasa de cruces por eso, esta hace referencia a la cantidad de veces que la señal pasa por el eje 0; es decir que en las señales donde hay mayor cantidad de cruces por cero es en señales de muy alta frecuencia como lo es el ruido.

**Coeficientes LPC**

Para saber el numero adecuado de coeficientes que se deben calcular por cada segmento, se debe tener en cuenta el espectro de la señal de voz y la envolvente que generan dichos coeficientes, ya que esta envolvente, debe seguir el comportamiento del espectro de la señal de voz para procesar los datos necesarios para reconocer la señal. Para este caso, el numero ideal, fue el 16, ya que esta cantidad representaba de muy buena manera el espectro de la señal de voz. La figura 11 muestra como con 16 coeficientes (linea rosada) es posible relacionar la envolvente LPC con la envolvente espectral del segmento. Para valores menores a 16, como lo son 1, 2, 5 y 10, la linea no sigue el sentido del espectro. Sin embargo para los valores de 16 y 20 se ve una trayectoria en la gráfica muy similar al espectro.

El motivo por el cual no se escoge como valor definitivo el de 20 coeficientes, teniendo en cuenta que se ve una mejor relación, es que no es conveniente para el software debido a la carga computacional que se genera de más.

**Algoritmo DTW**

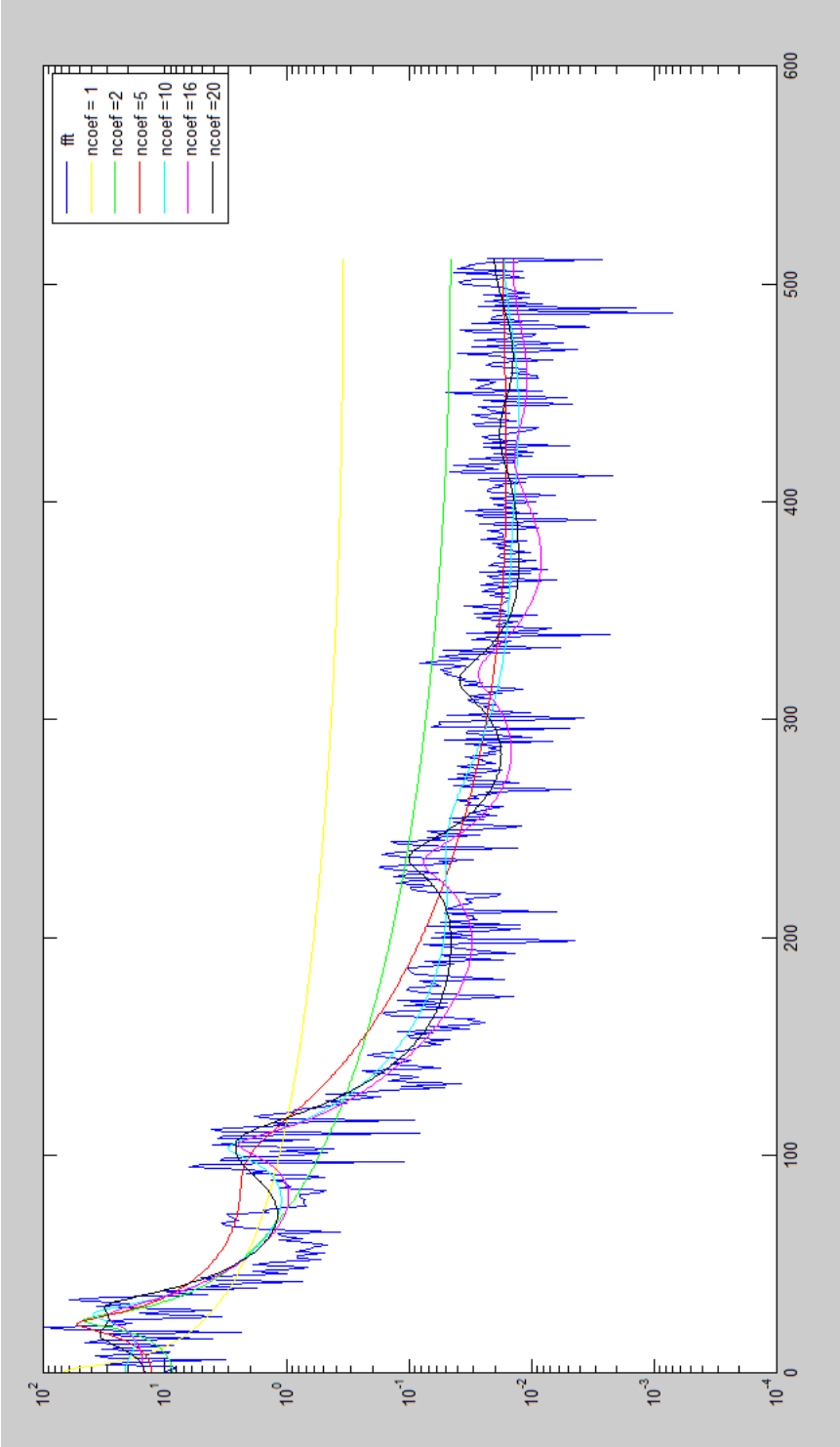
Generalmente, los proyectos relacionados con este tema, utiliza como método de identificación, la distancia euclídea; sin embargo esta no se pudo ejecutar en este caso, debido al no alineamiento temporal en las señales de la misma palabra; es decir, que al momento de segmentar las dos señales, estas no coincidían en el numero de tramas, debido a factores externos como duración en la pronunciación, ruido, etc.

Por este motivo, es que se investiga y se decide ejecutar el algoritmo DTW, ya que este provee un alineamiento entre extremos de la señal y agrupa unas tramas con otras, facilitando la comparación vectorial entre las señales. Adicionalmente, calcula la distancia mínima entre los puntos, ayudando a encontrar y reconocer la palabras que se ha mencionado.

**Visualización**

Para el diseño de la interfaz gráfica de usuario (GUI), se tuvieron en cuenta factores como la sencillez, el manejo rápido y que cumpliera con los requerimientos pactados.

Figura 11. Variación de coeficientes LPC.



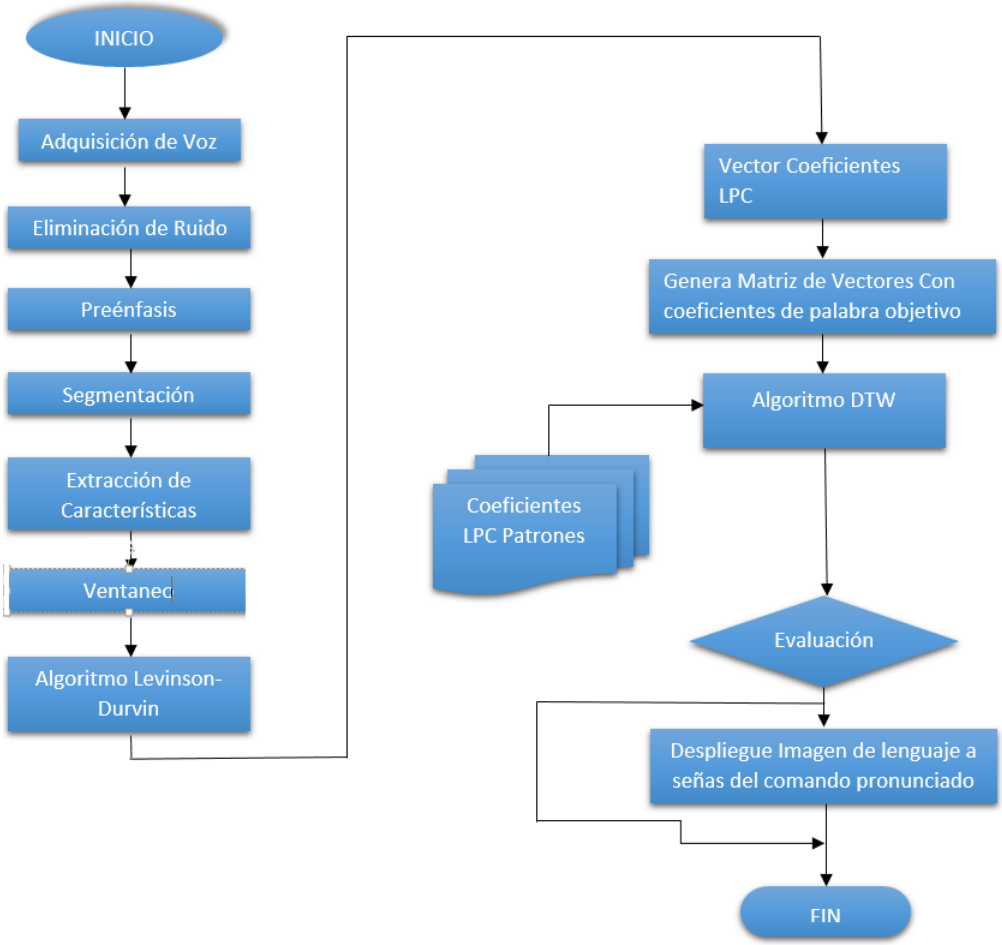

Fuente: Algoritmo de prueba.

Debido a esto se diseña un panel de presentación de la interfaz con el logo de la Institución Educativa y un despliegue completo de las 15 imágenes durante el funcionamiento. Para esta primera etapa se usaron imágenes de prueba que ayudaran a validar el funcionamiento del software de prueba. En la etapa de implementación con hardware, se utilizan imágenes más definidas y las que finalmente quedan.

Ya que se trata inicialmente de un software de prueba, se decide utilizar imágenes no definidas pero que sirvan para entender y saber cual fue la palabras que verdaderamente se va a mencionar. Para ello es necesario incluir algunas lineas de código que lea la base de datos y genere una visualización de la misma.

El diagrama de flujo que corresponde con el diseño del software es presentado en la figura 12, en donde se pueden observar todos los procesos mencionados anteriormente.

Figura 12. Diagrama de flujo del sistema.



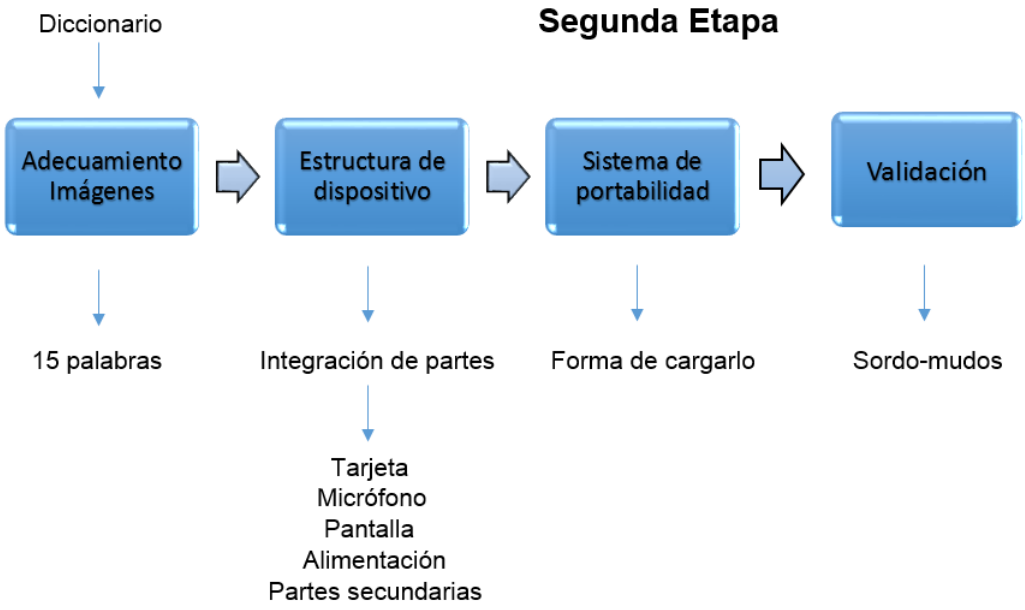

Fuente:Autores

Culminando esta etapa ya se podría pensar en una etapa de ensamblaje en hardware para tener un dispositivo físico para mostrar.

## 3.2 SEGUNDA ETAPA

Gran parte de esta etapa esta basada en la escogencia de los materiales adecuados para la implementación del dispositivo y otra pequeña parte de adecuamiento. La figura 13 muestra el diagrama de bloques para esta etapa.

Figura 13. Cuadro de diseño etapa Hardware.




Fuente:Autores

Ya que el software de prueba fue diseñado de tal manera en que sea funcional pero no definitivo, en este etapa se debe iniciar adecuando definitivamente las imágenes que se van a tener en cuenta para la visualización. Estas deben ser claras y de calidad para que se puedan observar apropiadamente.

### 3.2.1 Definición de materiales Hardware

Para la implementación del dispositivo inicialmente se tuvieron dos opciones de trabajo, las cuales consideraban la utilización de un microcontrolador o una tarjeta o placa programable. Analizando las ventajas y desventajas de cada una, se opta por utilizar una placa programable que sea pequeña, adatable al lenguaje de programación

escogido, potente y de un costo moderado. Todas estas características aplicaban a dos placas como lo son la MICROPYTHON y la RASPBERRY PI ZERO, sin embargo se decide escoger la RASPBERRY debido a que hay mayor información sobre esta, la cual nos podría aportar en el desarrollo.

Conociendo que la tarjeta no tiene entradas análogas, se debía adaptar un sistema que fuera capaz de realizar la conversión a digital, ya que el procesamiento esta en software. Las dos opciones más representativas eran incluir un circuito integrado ADC o un adaptador de audio el cual se encarga internamente de realizar dicho procesamiento. Nuevamente analizando las ventajas y desventajas se opta por el adaptador de audio debido a las mejoras que realiza sobre la señal de voz y su facilidad de manejo.

Con esto, se hace necesario tener un dispositivo transductor de señal de voz como lo es el micrófono, el cual debe ser de buena calidad ya que este debe obviar en gran medida las señales de ruido del exterior.

Hasta este momento ya se tienen los materiales necesarios para la adquisición, procesamiento y almacenamiento de las imágenes; sin embargo falta un modulo que permita visualizar el despliegue de la imágen que representa en lengua de señas la palabra que se menciona; teniendo en cuenta que debe ser pequeño, que tenga buena resolución y facil de adaptar a la tarjeta.

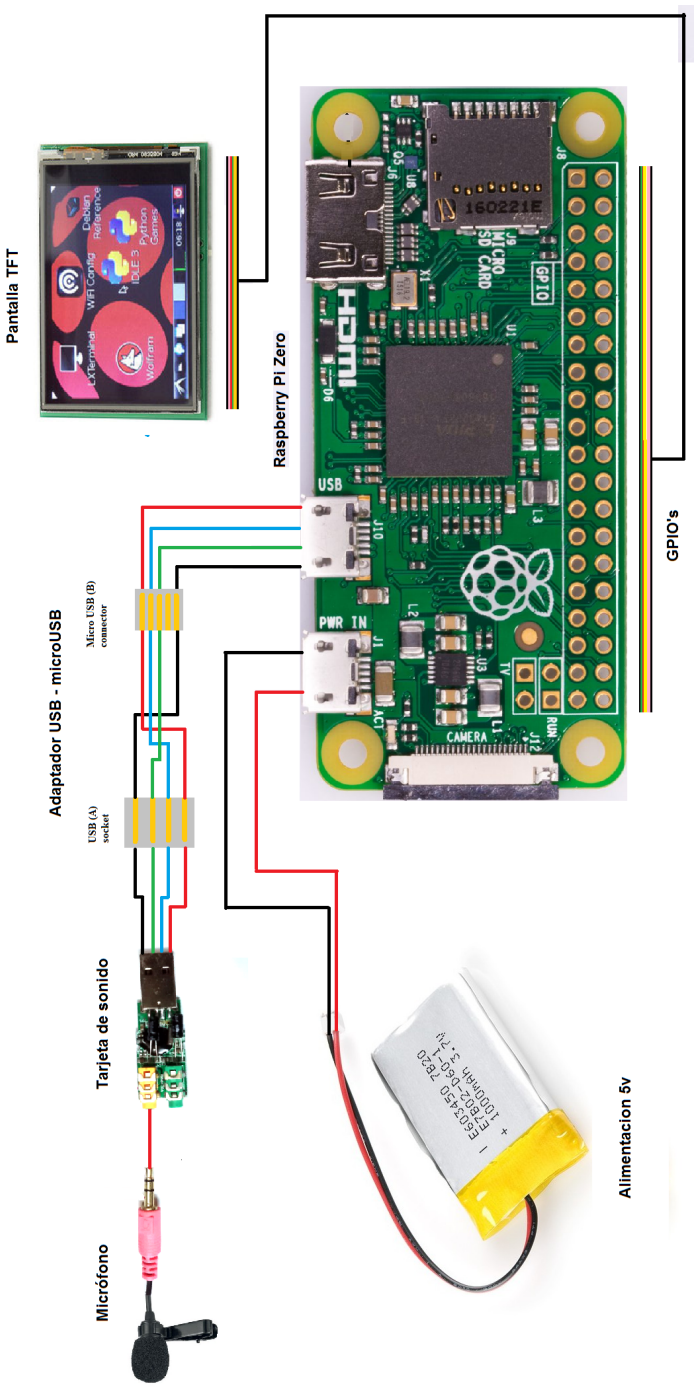
Otro aspecto importante que se tiene en cuenta, es que si el dispositivo va a ser portable, es estrictamente necesario un sistema de alimentación que permita mantener encendido el dispositivo sin la necesidad de estar conectado a la red electrica pública. Para esto se piensa en una alimentación eléctrica por medio de bateria DC (ya que la tarjeta tiene entrada de alimentación de este tipo), que satisfaga los requerimientos de voltaje y corriente de todos los dispositivos inmersos en el sistema completo.

En cuanto a estructura del dispositivo, los factores a tener en cuenta son los mencionados anteriormente; no obstante el dispositivo en esta forma no es apto para cargarlos facilmente, por este motivo, se diseña un brazalete tipo reloj pero mas ancho con sistema de sujeción por velcro para que sea fácil y rápido de poner; y que permita añadir al dispositivo sin problema alguno.

Mas adelante, en la sección de descripción de materiales utilizados para la implementación e integración del sistema, se mostrarán las imágenes correspondientes a cada uno de los materiales incluidos en el diseño.

En la siguiente figura se muestra el esquematico de como debe quedar el dispositivo luego de ser implementado.

Figura 14. Esquematico de dispositivo.




Fuente:Autores

# 4. IMPLEMENTACIÓN E INTEGRACIÓN

Para el proceso de implementación del dispositivo traductor, se tiene que tener en cuenta el diseño previamente mencionado y la manera en que se integran la etapa de software con la etapa de hardware. Dicho esto, a continuanción se describen las características importantes de cada etapa.

## 4.1 HARDWARE

Inicialmente se describen todos los materiales mas relevantes utilizados en el desarrollo del sistema, con el fin de saber las características más importantes de cada uno y conocer su función dentro del dispositivo.

### 4.1.1 Micrófono

El IK-M1 Lavalier Micrófono de Ikan es un electret condensador de solapa clip-on. Se utiliza para capturar audio en aplicaciones de diversos dispositivos como cámaras de video, grabadoras de audio, teléfonos inteligentes y mucho más. Cuenta con un patrón polar omnidireccional y tiene una respuesta de frecuencia de 65 a 18 kHz y está construido con un plug gold de 1/8 de 4 polos y cubierta de espuma.[20]

Figura 15. Micrófono.

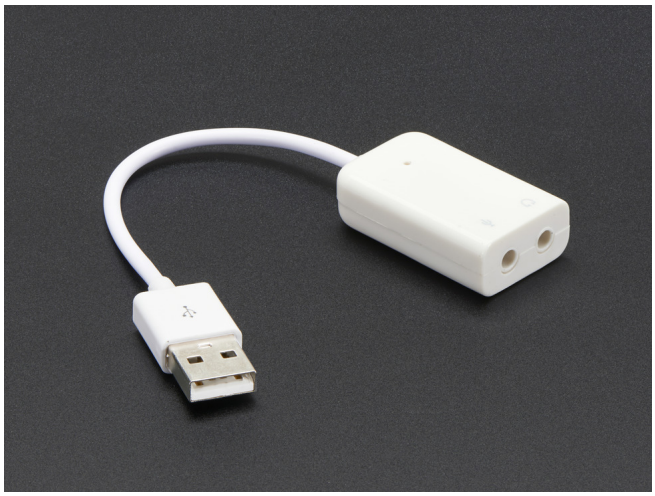
Fuente:https://www.bhphotovideo.com/c/product/1038804-REG/ikan_ik_m1_lavalier_microphone.html

[20] https://www.bhphotovideo.com/c/product/1038804-REG/ikan_ik_m1_lavalier_microphone.html

### 4.1.2 Adaptador de Audio USB

Debido a que la tarjeta de procesamiento por la cual se optó, no posee entradas analógicas, es imposible conectar directamente el micrófono a esta; por lo tanto es necesario de un dispositivo que sirva como puente entre ellos. Las soluciones más acertadas son la utilizaciòn de un ADC o un adaptador de audio de USB, en donde esta última opción se considera como la mejor debido a que causa una mejora en la calidad del sonido y mucho más práctica de utilizar. Este adaptador posee dos jack para plug stereo, por lo cual fue netamente necesario utilizar un plug de las mismas características. En la figura 16 se puede observar una imágen del adaptador.

Figura 16. Adaptador de Audio C-Media.



Fuente:https://www.adafruit.com/product/1475

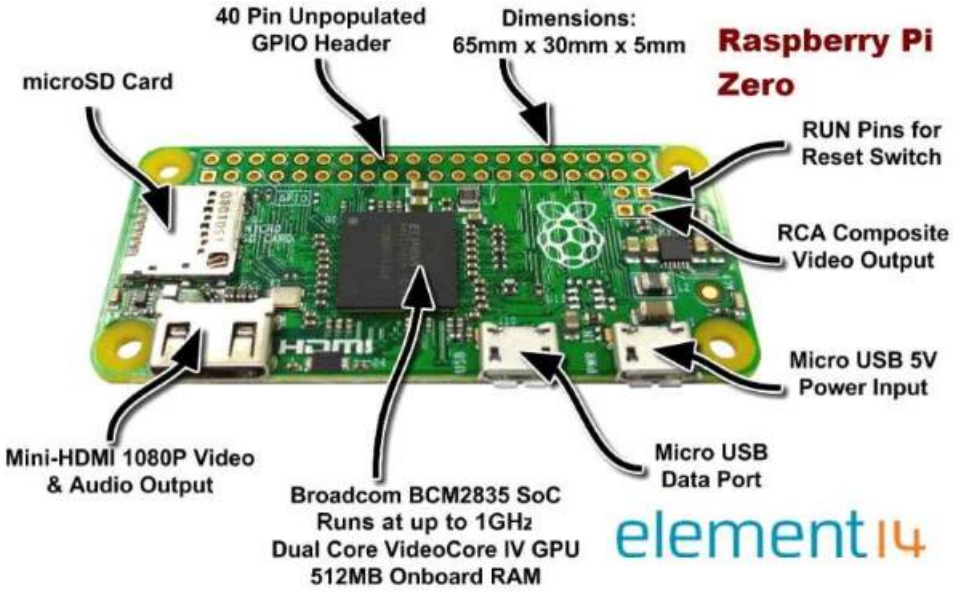
### 4.1.3 Raspberry Pi Zero

La Raspberry Pi Zero es la Raspberry Pi más económica del mercado y está orientada a aficionados de la computación educativa y emprendedores informáticos que desean tener equipos económicos para crear desarrollos y proyectos ambiciosos dentro del campo de la robótica y la automatización.

La principal característica de la Raspberry Pi Zero es que cuenta con las funcionalidades de sus predecesoras Raspberry Pi, e incluso puede llegar a ser hasta 50 por ciento más rápida que la Raspberry Pi original. La potencia de este fabuloso equipo informático viene dada por un procesador Broadcom BCM2835 con un núcleo ARM11 de 1GHz, el cual está empotrado en una placa de circuitos de 65x30 milímetros. Por supuesto, necesita memoria RAM para trabajar, pero tan sólo viene con 512MB de memoria. El sistema operativo se ejecuta desde una tarjeta de memoria del tipo microSD, lo que le aporta practicidad, velocidad, sigilo y eficiencia.

Para la salida de vídeo, la Raspberry Pi Zero viene equipada con un puerto Mini-HDMI de 1080p. En cuanto a la alimentación y transferencia de datos, están a cargo de 2 puertos Micro USB. Cabe destacar que su peso es de tan sólo 9 gramos.[21] En la figura 17 se puede observar la tarjeta Raspberry Pi Zero con sus componentes mas relevantes señalados.

Figura 17. Raspberry Pi Zero.




Fuente:https://www.raspberryshop.es/raspberry-pi-zero.php

### 4.1.4 Pantalla TFT

Este pequeño display esta diseñado específicamente para el Raspberry Pi. Cuenta con una pantalla de 2.4 pulgadas con 320x240 píxeles de 16 bits de color y una capa resistente al tacto. La placa utiliza la interfaz SPI de alta velocidad en el Raspberry Pi. Algunas características adicionales son de tipo LCD TFT, alimentación de 3.3v y 5v, 65536 colores en total, entre otras.

[21] https://www.raspberryshop.es/raspberry-pi-zero.php

Figura 18. Pantalla LCD TFT.


Fuente:https://electronilab.co/tienda/rpi-lcd-display-resistivo-2-4-320x480-tft-touchscreen-raspberry-pi2-pi3/

### 4.1.5 Módulo de Alimentación

Para la alimentación de todo el sistema se tienen en cuenta factores como consumo de corriente de la totalidad de los dispositivos conectados al módulo y el voltaje de entrada de estos, con el fin de evitar daños en los equipos electrónicos. La suma las corrientes que consumen los dispositivos no suman mas de 500 mA, por lo tanto una bateria con suministro de 600 mA es lo suficientemente adecuada para el funcionamiento del dispositivo.

Figura 19. Batería para alimentación.


Fuente:http://www.foto24.com/bateria-np-900-para-benq-dc-e720

### 4.1.6 Dispositivos Adicionales

Aunque los dispositivos mencionados anteriormente son los estrictamente necesarios para el funcionamiento del sistema traductor, se utilizaron algunos dispositivos adicionales al momento de la implementación con el objetivo de facilitar y realizar de mejor manera la fabricación del sitema; por ende se considera necesario relacionarlos en el presente documento.

**Tarjeta inalámbrica USB:** Es una tarjeta inalámbrica USB de alta tecnología, lo cual le permite tener un tamaño muy reducido con un desempeño excelente. Ofrece una transmisión estable hasta de 150Mbps, soporta USB 2.0 y es muy fácil de instalar. Diseñada para obtener una señal aceptable a pesar de su tamaño, tanto en portátiles como en equipos de escritorio. Cuenta con la tecnología High Power de 3bumen que le permite soportar ambientes hostiles de conexión y desconexión en caliente. No requiere desmontar para retirar. Nota: No es un equipo de alta potencia, la tecnología High Power es implementada para soportar ambientes hostiles.[22]

Figura 20. USB WIFI.

Fuente:http://www.3bumen.com/es/iproductos/ver/37/re_mini_2015/

La razón por la cual se utiliza este dispositivo es para minimizar los tiempos de trabajo mientras se realiza la configuración de la raspberry e instalaciones de drivers y demás. Esta tarjeta permite una conexión wifi con la tarjeta permitiendo realizar una comunicación por escritorio remoto y SSH. De esta manera no es necesario conectar teclados y pantallas a la raspberry evitando que se sobrecargue y se apague.

**Hub:** Es un dispositivo que permite concentrar varios puertos USB por medio de un solo cable. Es utilizado ante la escazes de puertos USB como la Raspberry Pi Zero, la cual solo tiene un puerto USB disponible para transmision de datos, ya que el otro es para alimentación.

[22] http://www.3bumen.com/es/iproductos/ver/37/re_mini_2015/

Figura 21. Hub no alimentado.


Fuente:https://actecom.es/cables-y-adaptadores/5119-adaptador-hub-usb-multiplicador-4-puertos-negro.html

### 4.1.7 Adaptación de materiales

Uno de los objetivos es conseguir implementar el dispositivo en un tamaño lo mas pequeño posible, por lo tanto es necesario adaptar algunos de los componentes mencionados anteriormente, con el fin de minimizar los espacios que consumen instalandolos así tal cual están.

- Para la comunicación de la Raspberry con la pantalla es necesario la utilización de una regleta de pines con el fin de no utilizar demasiados cables que puedan aumentar la probabilidad de ruido en el dispositivo. Esto es posible gracias a que la tarjeta trae consigo GPIO's (General Purpose Input Output) que facilitan soldar la regleta de los pines en estos. La tarjeta queda tal y como se observa en la figura 22, que comparandola con la tarjeta de la figura 17, se puede notar claramente la adaptación de la pantalla a esta.

Figura 22. Adaptación Raspberry con pantalla.

Fuente:Fotografía tomada de dispositivo.

•Otra forma de minimizar el uso de cable es cortando y adaptando de forma interna el plus stereo del micrófono al jack del adaptador de audio C-media, tal y como se muestra en la siguiente figura.

Figura 23. Adaptación de micrófono a tajeta de audio.

Fuente:Fotografía tomada de dispositivo.

•Mientras se desarrollaba el proyecto, se utilizo un cable OTG para la conexión del adaptador de audio son salida USB a la entrada de datos de la Raspberry tipo microUSB. Esto al momento de ensamblaje consume mucho espacio, debido a esto se modifica la salida USB del adaptador de audio a una salida microUSB, con el fin de no utilizar el cable OTG.

•Para la entrada de alimentación tipo microUSB de la tarjeta, se hace la conexión directa de un pin de carga microUSB a los pines de alimentación previamente dispuestos.

## 4.2 SOFTWARE

### 4.2.1 Software de Prueba

El desarrollo del software de prueba es una base muy importante para la culminación del proyecto, ya que allí es donde se realiza todo el procesamiento para encontrar cual fue la palabra que se ha mencionado. Al momento de desarrollar el código, se tuvo en cuenta un estructura en funciones con el fin de optimizar el algoritmo y evitar enredos en el momento de la modificación. Este fue realizado en un ordenador común en un entorno de desarrollo integrado llamado PYCHARM version Community.

### 4.2.2 Adecuamiento de Imágenes

Tal y como se ha mencionado con anterioridad, las pruebas iniciales se realizaron con imágenes que no eran aptas para la presentación; por ende se buscan imágenes mas definidas y que la persona que la vea entienda rápidamente la palabra que se mencionó. La figura 24 muestra un mosaico del total de las 15 imágenes correspondientes al listado de palabras mencionado en la sección de pruebas. Estas imágenes son fáciles de interpretar y de una claridad buena para su lectura.

En algunas de estas imágenes se hace necesario tener en cuenta los gestos que realiza la persona y en otras la trayectoria o dirección en la que apuntan las flechas.

### 4.2.3 Sistema Operativo

La Raspberry Pi Zero se caracteriza por ser reconocido como un mini ordenador, ya que cumple funciones muy similares a la de un computador común. Segun esto, la tarjeta necesita un sistema operativo en el cual funcionar que se conoce como RASPBIAN y unas librerias complementarias.

**RASPBIAN:** Raspbian es un sistema operativo libre basado en Debian y optimizado para el hardware de Raspberry Pi. Ofrece más que un sistema operativo pur, ya que viene con más de 35.000 paquetes, software precompilado incluido en un formato agradable para la instalación fácil en Raspberry Pi. [23] Un dato importante que se debe tener en cuenta, es que la Raspberry Pi Zero no cuenta con un sistema de almacenamiento interno, por lo cual es necesario instalar una tarjeta miniSD en el cual se va a instalar el sistema operativo. Por recomendaciones de tutores especialistas en el manejo de este tipo de tarjetas programables, se adquiere una miniSD de 8GB clase 10, la cual se caracteriza por manejar facilmente grandes cantidades de datos por segundo, garantizando el flujo sin problemas de los datos.

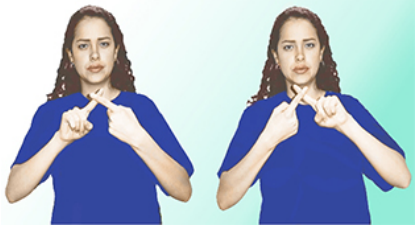
Adicionalmente a la instalación del sistema operativo Raspbian sobre la miniSD, tambien se requiere para el desarrollo del software, la instalación del algunas librerias de manejo de audio, la cuales no disponen por defecto en el interprete de python manejado.

[23] www.raspbian.org

Figura 24. Imágenes establecidas a reconocer

(a) Aceptar

(b) Amigo



(c) Bus

(d) Buscar

(e) Casa

(f) Cerrar

(g) Conocer

(h) Descubrir

(i) Domingo

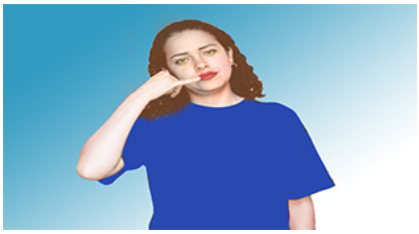
(j) Futuro

(k) Hola

(l) Ingeniero

(m) Salir

(n) Telefono



(o) Todos

Fuente: Mejoramiento de imágenes de diccionario de lengua de señas colombiana

# 5. PRUEBAS

A continuación se presentan resultados obtenidos en el desarrollo y luego de la implementación del dispositivo traductor, los cuales se analizan con el objetivo de dar conclusiones claras y concisas.

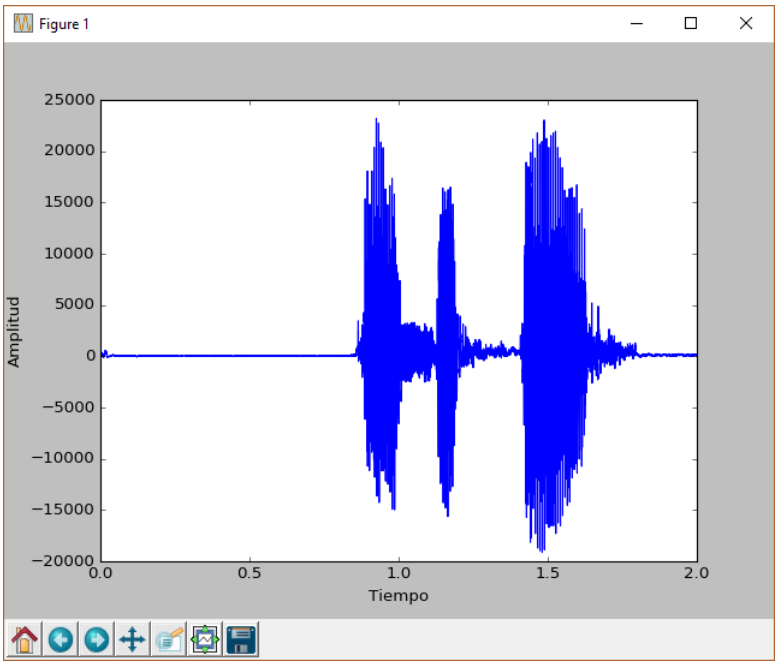
## 5.1 LISTADO DE PALABRAS

Aceptar Amigo Bus Buscar Casa Cerrar Conocer Descubrir Domingo Futuro Hola Ingeniero Salir Teléfono Todos

## 5.2 CAPTACIÓN DE SEÑAL DE VOZ

Para capturar la señal de voz se utilizan patrones caràcteristicos importantes como la cantidad de canales y la frecuencia de muestreo (44100 Hz) evitando ampliamente el aliasing. El código de programación genera el vector de muestras y la imágen de la señal de voz; por ejemplo de la palabra ACEPTAR tal y como se muestra a continuaciòn. Este código con el cual se realiza este proceso se encuentra adjunto en el anexo a. con tìtulo CAPTACIÓN DE SEÑAL DE VOZ.

Figura 25. Señal de palabra ACEPTAR




Fuente: Algoritmo de captaciòn de señal de voz.

Cabe destacar que inicialmente no se observa amplitud notable en la imágen, debido a que en esos instantes aún no se encuentra actividad vocal sobre el micrófono. Las partes representativas de la señal corresponden generalmente a segmentos vocálicos debido a su naturaleza sonora. Los parámetros tenidos en cuenta para la adquisión de la señal se muestran en la siguiente tabla.

Tabla 1. Parámetros de adquisición de señal.

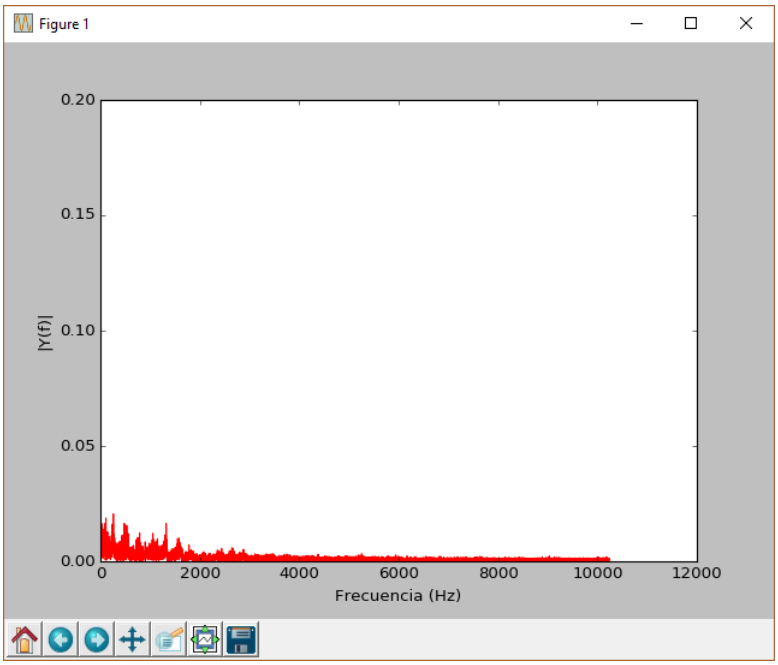
| Parámetros | |
|---|---|
| Formato | WAV |
| Frecuencia de muestreo | 44100 Hz |
| Tipo de Canal | Mono |
| Resolución | 16 bits |
| Duración | 2 segundos |

La cantidad de canales que se optan para la adquisición de la señal es debido a que el adaptador de audio C-media trabaja mal con grabaciones a dos canales. Inicialmente se pensó en una falla del adaptador, sin embargo en la insvestigación se encuentra de que este tipo de dispositivo tiene como desventaja dicho parámetro.

## 5.3 PREÉNFASIS

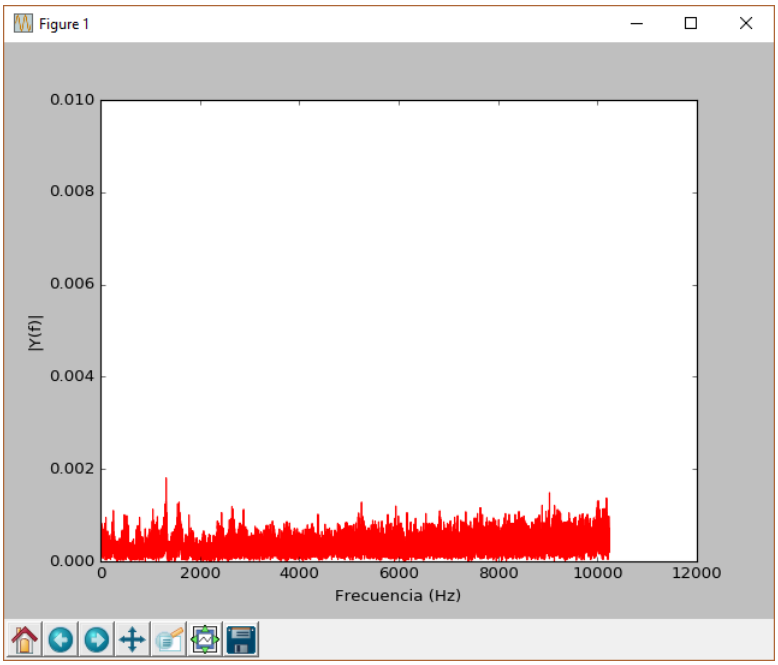
La etapa de preénfasis se realiza con el propósito de suavizar el espectro y reducir las inestabilidades de cálculo asociadas con las operaciones aritméticas de precisión finita. Además se usa para compensar la caída de -6 dB que experimenta la señal al pasar a través del tracto vocal. El efecto que trae en la señal de voz es potenciar el nivel de las frecuencias altas en proporción al aumento de la amplitud de ruido en dichas frecuencias. En otras palabras, el nivel de las frecuencias altas se aumenta para contrarestar el aumento del nivel del ruido. Esta afirmación se puede demostrar teniendo en cuenta el espectro de la señal de voz antes y luego de pasar por el filtro preénfasis. La figura 26 muestra el espectro de la señal de voz y en la figura 27 muestra como se realzan las frecuencias altas para contrarrestar el ruido.

Figura 26. Espectro de la señal aceptar




Fuente: Algoritmo.

Figura 27. Preénfasis de la señal aceptar

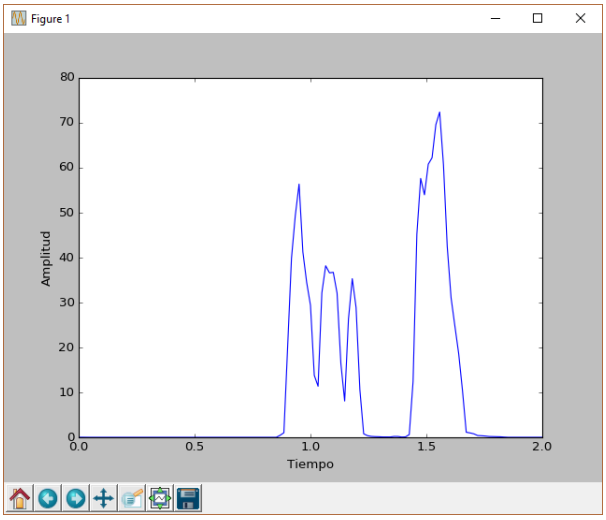



Fuente: Algoritmo.

## 5.4 ENERGÍA Y CRUCES POR CERO

Teniendo en cuenta que la señal de voz se caracteriza por su naturaleza cambiante, es bastante difícil estudiar sus características; por lo tanto se hace necesario segmentar la señal en tramas entre 20 y 30 ms (teniendo en cuenta el solapamiento entre tramas), ya que en estos fragmentos se considera que el comportamiento de la señal es cuasi-estacionaria, permitiendo la extracción de dichas características. La figura 28 muestra la energìa de la señal de voz aceptar. Si se compara esta figura con la figura 25, se puede observar que los grandes valores de energía se encuentran en las partes de mayor amplitud de la señal de voz, por ende se conoce que la energía esta directamente relacionada con la amplitud de la señal.

Figura 28. Cálculo de energía en señal de voz



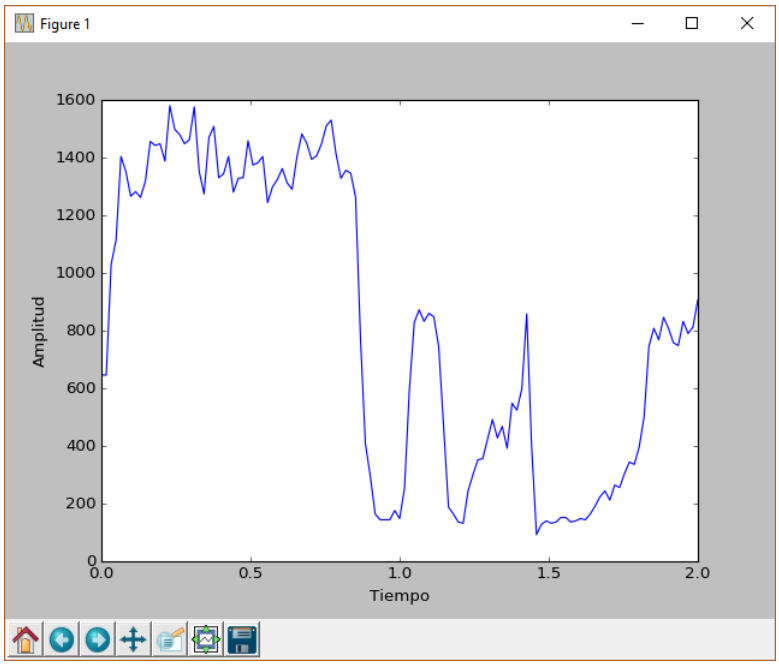

Fuente:Algoritmo de cálculo de energía

Otra característica utilizada es la tasa de cruces por cero, la cual se identifica esencialmente por un contador de veces en que la señal cruza por el eje tiempo o cuantas veces ocurren cambios de signo en las muestras de la trama estudiada (en tiempo discreto sería cuantas veces la muestra cambie de 0 a 1 o viceversa).

Alta energía + Bajo tasa de cruces por cero = Trama con Voz
Baja energía + Alta tasa de cruces por cero = Trama sin Voz

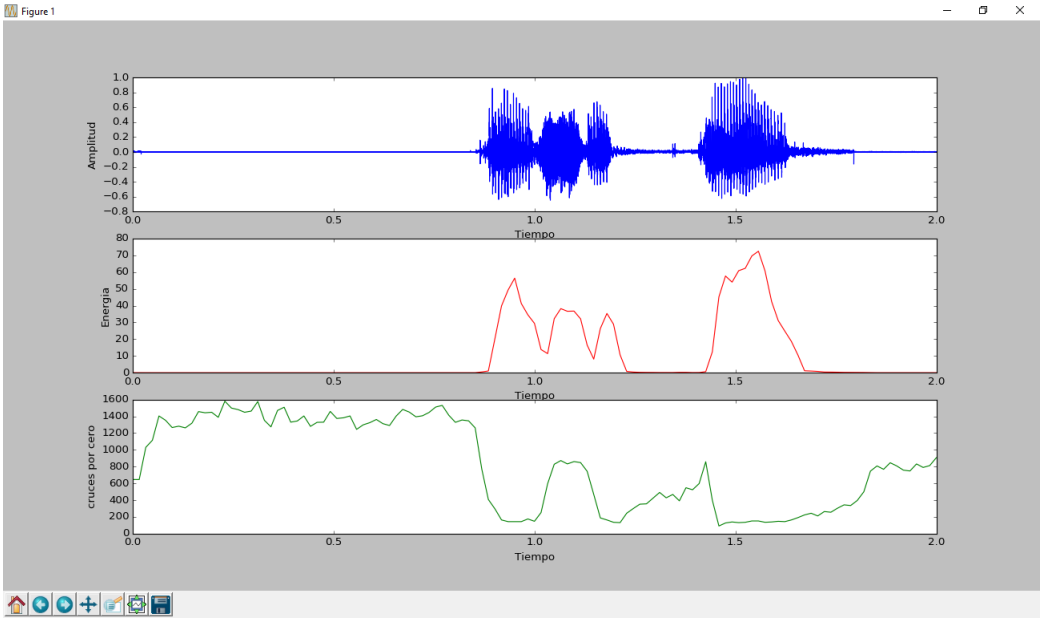
La figura 29 muestra la gráfica de tasa de cruces por cero de la señal aceptar. Cabe notar que estos valores son mas altos o notorios en los tramos de la señal en que no hay amplitud o segmentos vocálicos.

Figura 29. Cálculo de cruces por cero de palabra ACEPTAR




Fuente: Algoritmo de cálculo de tasa de cruces por cero.

Figura 30. Características de palabra ACEPTAR



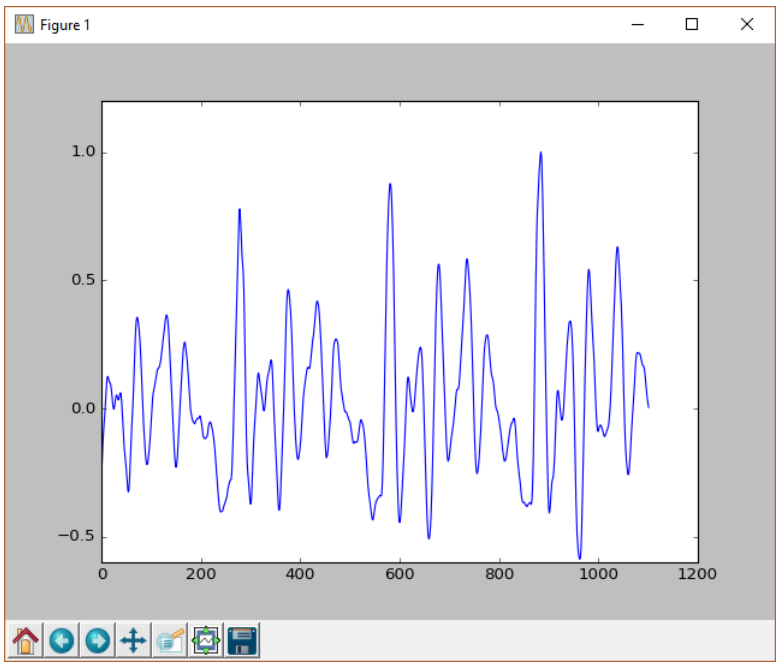

Fuente: Algoritmo.

Estas características son útiles puesto que permiten el procesamiento de solo las tramas que contienen información. Las tramas sordas simplemente no se procesan. Para examinar y entender de mejor manera lo que se ha dicho, se presenta la figura 30 en donde se mantienen juntas las graficas de la señal de voz, la energía y los cruces por cero.

## 5.5 VENTANEO

Tal y como se ha mencionado anteriormente, segmentar la señal de voz sirve como manera eficaz en el análisis de esta, por este motivo es que se realiza dicho paso durante el procesamiento. La figura 31 muestra una trama de la señal aceptar, en donde se nota claramente la periodicidad que existe en ella, por lo que se reconoce como una trama sonora útil para el análisis.

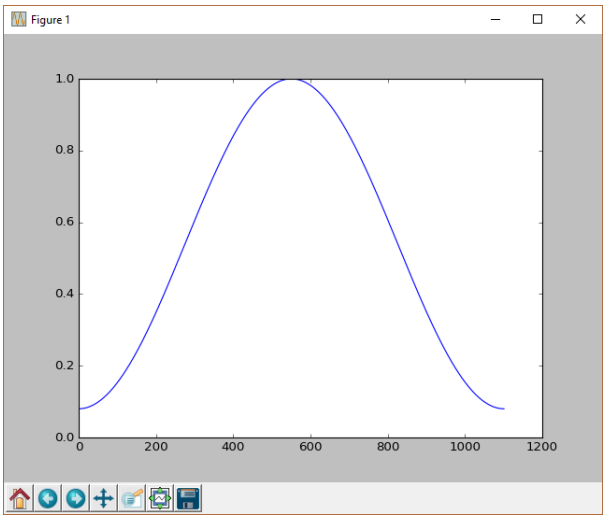
Figura 31. Trama de señal ACEPTAR




Fuente: Algoritmo de segmentación.

Luego de que se ha realizado la segmentación y la identificación de los sonidos sonoros, se procede a realizar el enventanado de la señal. Para esto es necesario escoger una señal de ventana adecuada para el estudio. La figura 32 muestra la gráfica de la ventana hamming.

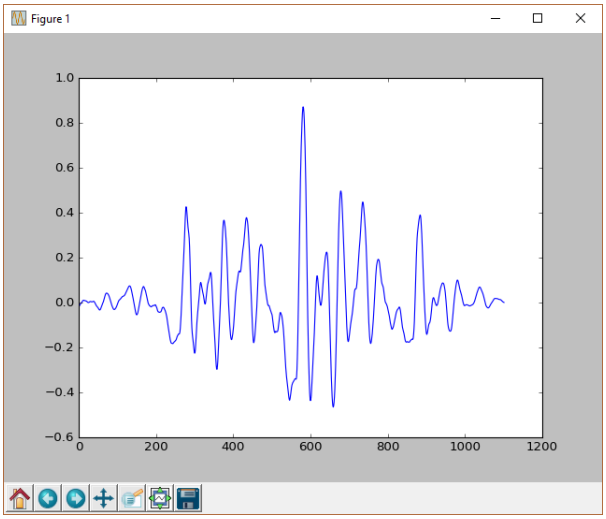
Figura 32. Ventana Hamming.




Fuente: Algoritmo de captaciòn de señal de voz.

Como se sabe que la ventana hamming suaviza el espectro en los extremos de la señal, esto tambien se puede notar en el tiempo, ya que la trama de la señal toma la forma de la envolvente de la ventana luego de que dicha trama ha sido multiplicada por esta. En la figura 33 se puede observar el efecto de la ventana hamming en la trama de la señal aceptar.

Figura 33. Efecto de ventana hamming en la trama.




Fuente: Algoritmo.

## 5.6 COEFICIENTES LPC

La cantidad de coeficientes del filtro LPC determinan en gran medida el rendimiento del sistema de reconocimiento de palabras. Por este motivo es que se debe escoger el valor de coeficientes que mejor asemeje la envolvente del espectro LPC con el espectro de la señal de voz. En la figura 11 se presenta en forma práctica dicha situación en donde se demuestra gráficamente el porque se escogen 16 coeficientes LPC por trama para el respectivo análisis.

Tabla 2. Coeficientes en la misma palabra.

| TODOS0 | TODOS1 | TODOS2 | TODOS3 | TODOS4 |
|---|---|---|---|---|
| 12,704 | 13,582 | 13,582 | 13,673 | 13,831 |
| -5,307 | -4,045 | -4,045 | -4,253 | -3,362 |
| 7,75 | 8,063 | 8,063 | 9,386 | 10,149 |
| 5,072 | 4,091 | 4,091 | 7,446 | 7,552 |
| 6,269 | 4,598 | 4,598 | 8,699 | 8,726 |
| 4,033 | 3,051 | 3,051 | 5,615 | 5,599 |
| 1,331 | 0,662 | 0,662 | 2,418 | 1,196 |
| -3,455 | -4,983 | -4,983 | -2,493 | -5,401 |
| -6,808 | -9,257 | -9,257 | -6,492 | -8,675 |
| -8,071 | -9,817 | -9,817 | -10,003 | -8,841 |
| -8,797 | -8,149 | -8,149 | -12,224 | -8,409 |
| -9,254 | -7,158 | -7,158 | -12,862 | -9,59 |
| -9,962 | -8,455 | -8,455 | -12,564 | -11,545 |
| -8,446 | -8,425 | -8,425 | -9,853 | -10,293 |
| -6,579 | -6,639 | -6,639 | -6,453 | -6,75 |
| -4,672 | -3,755 | -3,755 | -3,746 | -3,164 |

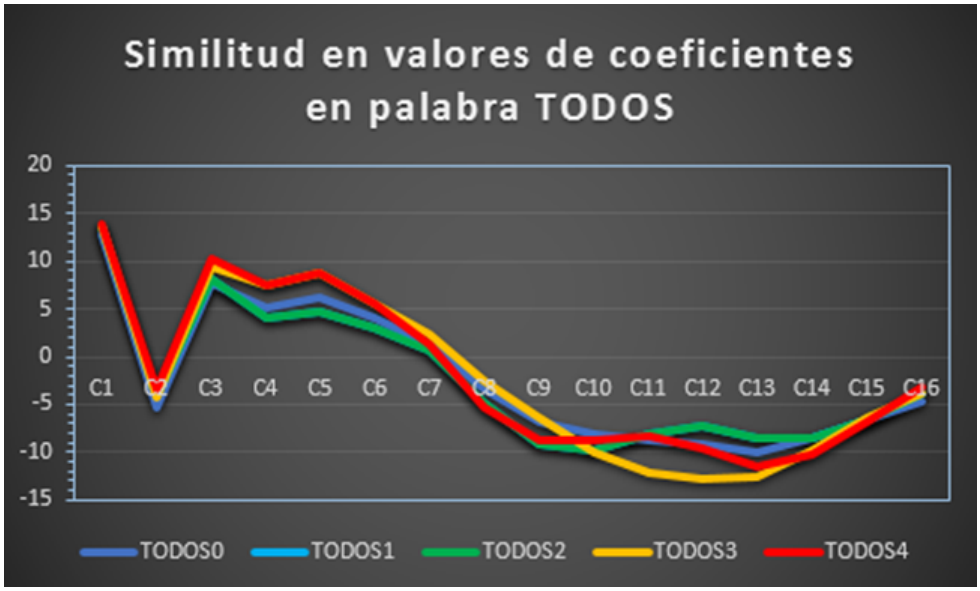
Fuente: Algoritmo LPC

Hay algunas investigaciones que se han realizado con 10 coeficientes LPC, sin embargo este valor no ofrecia grandes garantías para reconocer las palabras, ya que el filtro no es lo suficientemente definido como para rendir de una mejor manera.

Es de saber que para que las palabras sean reconocidas, los coeficientes LPC de las mismas palabras deben tener una gran similitud entre sus valores. Por este motivo es que se presenta la tabla 2, ya que allí se puede observar que entre los coeficientes LPC correspondientes a diferentes grabaciones de la misma palabra, hay grandes similitudes.

Para analizar de una mejor manera lo mencionado anteriormente, se presenta la gràfica de similitud de los coeficientes de una misma palabra en un segmento (ver figura 34). Allí se puede observar claramente como todos los coeficientes toman valores muy similares siguiendo la trayectoria que toman los demas coeficientes de las otras grabaciones.

Figura 34. Similitud en coeficientes LPC de una misma palabra.




Fuente: Autores.

En la tabla 3 se presentan los coeficientes obtenidos del primer segmento de cada una de las señales de voz estudiadas, con el fin de ver las diferencias de las magnitudes de cada uno de estos, en comparaciòn con los valores de las otras palabras.

Tabla 3. Coeficientes LPC en primer segmento.

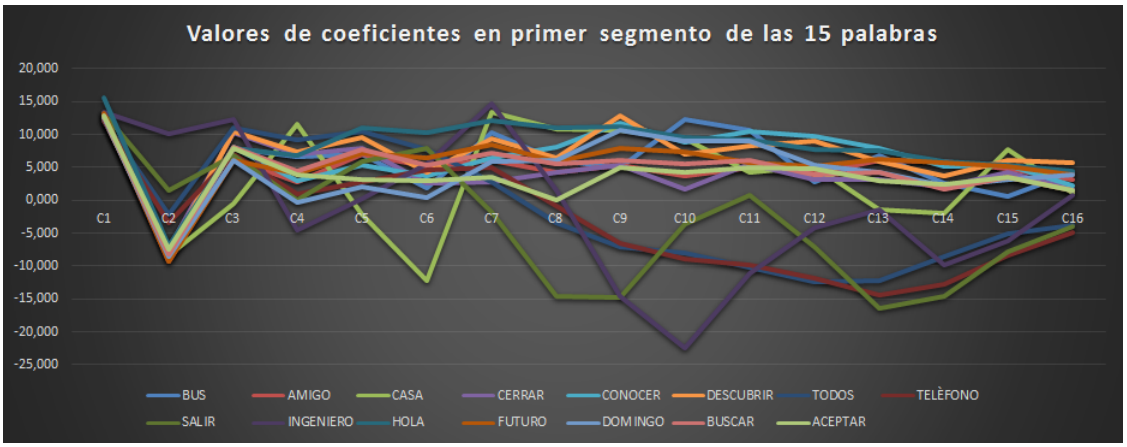
| COEFICIENTES LPC EN PRIMER SEGMENTO | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BUS | AMIGO | CASA | CERRAR | CONOCER | DESCUBRIR | TODOS | TELÉFONO | SALIR | INGENIERO | HOLA | FUTURO | DOMINGO | BUSCAR | ACEPTAR |
| 13,034 | 13,282 | 13,282 | 12,282 | 12,798 | 12,532 | 12,186 | 12,148 | 13,144 | 13,455 | 15,667 | 13,212 | 12,668 | 12,773 | 12,882 |
| -8,344 | -9,292 | -8,435 | -7,249 | -7,354 | -8,269 | -2,179 | -3,423 | 1,569 | 10,113 | -6,854 | -9,367 | -8,546 | -8,119 | -7,547 |
| 7,827 | 6,171 | -0,610 | 10,210 | 8,130 | 10,373 | 10,994 | 7,013 | 6,849 | 12,223 | 7,926 | 6,033 | 5,977 | 8,131 | 7,956 |
| 6,613 | 2,738 | 11,464 | 6,988 | 2,905 | 7,426 | 9,191 | 1,028 | 0,094 | -4,559 | 6,863 | 3,965 | -0,370 | 4,402 | 3,851 |
| 7,066 | 6,921 | -2,244 | 7,940 | 5,258 | 9,509 | 10,538 | 2,685 | 5,949 | 0,261 | 10,918 | 7,133 | 2,113 | 7,763 | 3,187 |
| 1,781 | 5,233 | -12,156 | 2,531 | 3,716 | 4,268 | 7,697 | 4,579 | 7,973 | 5,463 | 10,209 | 6,431 | 0,479 | 5,243 | 2,986 |
| 10,198 | 6,080 | 13,347 | 2,776 | 6,411 | 9,422 | 2,687 | 4,929 | -1,720 | 14,649 | 12,150 | 8,394 | 5,881 | 7,114 | 3,572 |
| 6,101 | 4,302 | 10,912 | 4,296 | 8,031 | 6,356 | -3,501 | -0,808 | -14,581 | 1,416 | 11,014 | 5,905 | 6,111 | 5,538 | -0,059 |
| 5,055 | 5,277 | 10,668 | 5,324 | 11,750 | 12,745 | -7,053 | -6,630 | -14,715 | -14,672 | 11,251 | 7,867 | 10,641 | 6,090 | 4,900 |
| 12,194 | 3,607 | 9,365 | 1,602 | 8,861 | 7,003 | -8,100 | -8,892 | -3,598 | -22,519 | 9,615 | 7,284 | 8,903 | 5,542 | 4,201 |
| 10,577 | 5,001 | 4,157 | 5,584 | 10,472 | 8,321 | -10,235 | -9,842 | 0,701 | -11,140 | 9,154 | 5,735 | 9,040 | 6,013 | 4,944 |
| 2,860 | 4,076 | 5,118 | 3,137 | 9,727 | 8,970 | -12,362 | -11,782 | -7,179 | -4,120 | 7,672 | 5,112 | 5,313 | 3,816 | 4,796 |
| 6,859 | 6,153 | -1,364 | 3,202 | 7,848 | 5,865 | -12,285 | -14,391 | -16,413 | -1,442 | 7,546 | 6,307 | 4,278 | 4,194 | 2,925 |
| 2,509 | 5,864 | -2,054 | 1,982 | 5,100 | 3,621 | -8,532 | -12,716 | -14,540 | -9,891 | 5,812 | 5,649 | 2,215 | 1,700 | 2,451 |
| 0,644 | 4,824 | 7,796 | 4,223 | 5,383 | 6,113 | -5,103 | -8,374 | -7,835 | -6,118 | 5,361 | 5,205 | 3,195 | 3,515 | 3,586 |
| 4,155 | 3,150 | 1,089 | 1,702 | 2,235 | 5,718 | -3,811 | -4,888 | -4,086 | 0,792 | 4,445 | 3,896 | 3,866 | 1,568 | 1,421 |

Fuente: Algoritmo LPC

De la tabla 3 es posible graficar el comportamiento de estos coeficientes (ver figura 35), observando en esta, que hay ciertas diferencias en los valores de algunas palabras, sin embargo entre otras palabras hay similitudes las cuales dificultaron aún más el proceso de reconocimiento de la palabra. Es importante mencionar que estos valores son del

primer segmentos, ya que en los segmentos siguientes se observan comportamientos diferentes.

Figura 35. Comportamiento de coeficientes LPC en diferentes palabras.




Fuente: Autores.

## 5.7 DTW

La gran mayoria de algoritmos de síntesis utilizan la distancia LPC o distancia euclídea como método para la reconstrucción de la señal, pero al aplicar dicha forma a la comparación de distintas palabras, resultaba imposible encontrar resultados exitosos, debido al deslineamiento temporal entre estas.

Inicialmente se penso en ajustar dicho desfase por medio de un factor de multiplicación pero esto no se podia por dos razones. Primero porque el valor de todas las tramas sonoras de una palabra no coincidian con el valor de las tramas sonoras de otra palabra. Segundo por que al multiplicar el eje de tiempo, alterabamos la posición de las muestras, haciendo imposible reconocer las palabras.

Por esta grande razón es que tan necesario el uso del algoritmo DTW, quien es el encargado de alinear la señales y comparar todas las muestras equitativamente para encontrar la menor distancia entre ellas.

En la tabla 4 se pueden observar los valores correspondientes a la distancia de los coeficientes LPC de la palabra mencionada (HOLA) frente a las palabras almacenadas previamente. Cabe aclarar que las distancias mostradas en esta tabla, corresponden a los valores mínimos de cada vector de distancia que se encuentra, ya que en la base de datos son muchas las grabaciones que se tienen por cada comando; es decir que por cada palabra se genera un vector de muchas distancias. Se notan los valores mínimos para visualizar las pequeñas diferencias entre algunas, lo que hace complejo el reconocimiento de las palabras.

También se puede observar que el valor mínimo de distancia se encuentra en la palabra hola, lo cual induce a un acierto en dicha prueba, ya que se mencionó la palabra hola y el software entrega dicho resultado.

De igual manera, se observa que la mayor distancia se encuentra en la palabra ingeniero, la cual da a entender que por sus características como cantidad de letras y formantes distintos, es que el software la da como su última opción.

Tabla 4. Tabla de distancias frente a palabra HOLA

| **VALOR DE DISTANCIAS POR DTW VS PALABRA "HOLA"** | | | | |
|---|---|---|---|---|
| **ACEPTAR** | **AMIGO** | **BUS** | **BUSCAR** | **CASA** |
| 6,6809 | 6,5586 | 7,0659 | 5,7419 | 5,5342 |
| **CERRAR** | **CONOCER** | **DESCUBRIR** | **DOMINGO** | **FUTURO** |
| 7,243 | 8,8556 | 7,4812 | 9,6915 | 7,5686 |
| **HOLA** | **INGENIERO** | **SALIR** | **TELÉFONO** | **TODOS** |
| 0,6721 | 17,0213 | 8,3119 | 7,7815 | 5,9091 |

## 5.8 PRUEBAS DE FUNCIONAMIENTO

Esta prueba de funcionamiento consiste en una fase de verificación de la eficiencia del dispositivo en el momento de reconocer la palabra que se ha mencionado. La prueba consiste en obtener las muestras de 15 personas distintas mencionando cada una de las palabras, con el fin de tener el porcentaje de acierto y desacierto de cada una de ellas.

Por ejemplo que de 15 veces que se menciona la palabra aceptar 13 son resultados favorales correspondientes a un 86,66 porciento, mientras que el 13,33 porciento restante se reparte entre las palabras amigo y cerrar. Cabe aclarar que dicho resultado fue solo relacionando la palabra ACEPTAR, ya que para otras palabras el resultado de acierto es variable, tal y como se muestra en la tabla 5.

Tabla 5. Verificación de funcionamiento.

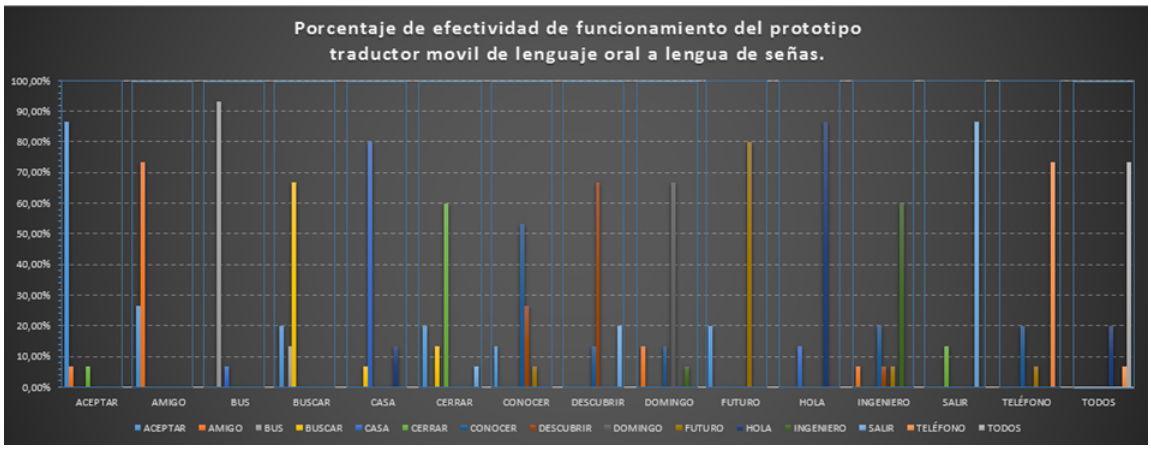
| VERIFICACIÓN DE FUNCIONAMIENTO DE DISPOSITIVO | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ACEPTAR | AMIGO | BUS | BUSCAR | CASA | CERRAR | CONOCER | DESCUBRIR | DOMINGO | FUTURO | HOLA | INGENIERO | SAUR | TELÉFONO | TODOS |
| ACEPTAR | 86,66% | 6,66% | 0,00% | 0,00% | 0,00% | 6,66% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% |
| AMIGO | 26,66% | 73,33% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% |
| BUS | 0,00% | 0,00% | 93,33% | 0,00% | 6,66% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% |
| BUSCAR | 20,00% | 0,00% | 13,33% | 66,66% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% |
| CASA | 0,00% | 0,00% | 0,00% | 6,66% | 80,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 13,33% | 0,00% | 0,00% | 0,00% | 0,00% |
| CERRAR | 20,00% | 0,00% | 0,00% | 13,33% | 0,00% | 60,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 6,66% | 0,00% | 0,00% |
| CONOCER | 13,33% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 53,33% | 26,66% | 0,00% | 6,66% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% |
| DESCUBRIR | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 13,33% | 66,66% | 0,00% | 0,00% | 0,00% | 0,00% | 20,00% | 0,00% | 0,00% |
| DOMINGO | 0,00% | 13,33% | 0,00% | 0,00% | 0,00% | 0,00% | 13,33% | 0,00% | 66,66% | 0,00% | 0,00% | 6,66% | 0,00% | 0,00% | 0,00% |
| FUTURO | 20,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 80,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% |
| HOLA | 0,00% | 0,00% | 0,00% | 0,00% | 13,33% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 86,66% | 0,00% | 0,00% | 0,00% | 0,00% |
| INGENIERO | 0,00% | 6,66% | 0,00% | 0,00% | 0,00% | 0,00% | 20,00% | 6,66% | 0,00% | 6,66% | 0,00% | 60,00% | 0,00% | 0,00% | 0,00% |
| SAUR | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 13,33% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 86,66% | 0,00% | 0,00% |
| TELÉFONO | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 20,00% | 0,00% | 0,00% | 6,66% | 0,00% | 0,00% | 0,00% | 73,33% | 0,00% |
| TODOS | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 0,00% | 20,00% | 0,00% | 0,00% | 6,66% | 73,33% |

Fuente: Autores

Los resultados enmarcados de color rojo corresponden a los porcentajes cuando el software muestra la imágen que corresponde a la palabra que se menciona. Es de notar que los mayores aciertos se encuentran en las palabras ACEPTAR, BUS, HOLA y SALIR. Cuando en la tabla se muestra un porcentaje de cero, es porque de las 15 palabras que se mencionaron, el dispositivo no mostro imágenes correspondientes a estas.

En la figura 36 se muestra gráficamente la información obtenida en la tabla 4. Allí se emarca cada palabra definida y sus respectivos porcentajes de efectividad.

Figura 36. Porcentaje de efectividad de dispositivo.



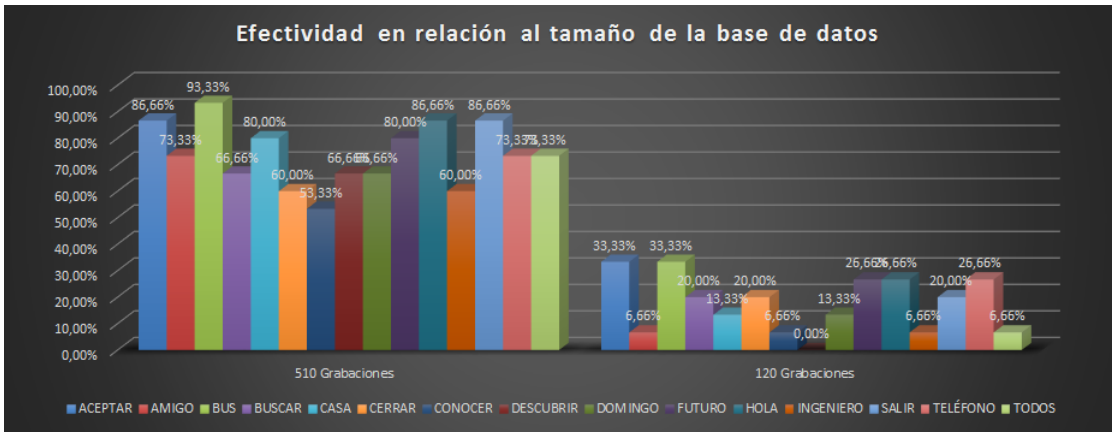

Fuente: Autores.

Cabe resaltar que para conseguir un buen reconocimiento de las palabras tal cual se muestra en la anterior tabla, se tiene que realizar un buen entrenamiento de los patrones de señales; entendiendo por entrenamiento a la instauración de una base de datos grande que amplie el rango de acierto en las palabras.

La base de datos definitiva quedo con un tamaño de 510 grabaciones de todos los 15 comandos, es decir un total de 34 grabaciones por cada palabra que se quiere procesar. En la siguiente figura se muestra la efectividad mostrada con este tamaño de base de datos y como disminuye dicho porcentaje cuando se disminuye la embergadura de la base de datos. Es claro que el porcentaje de efectividad con 120 grabaciones baja considerablemente a tal punto de obtener un reconocimiento pésimo.

Este proceso se realizó debido a los tiempos de repuesta del dispositivo, ya que con 510 grabaciones el dispositivo se toma alrededor de 35 segundos para finalizar su trabajo, sin embargo cuando se disminuyo la cantidad de grabaciones en la base de datos, este tiempo bajó en gran medida, sacrificando la efectividad del dispositivo, tal y como se muestra en la figura 37.

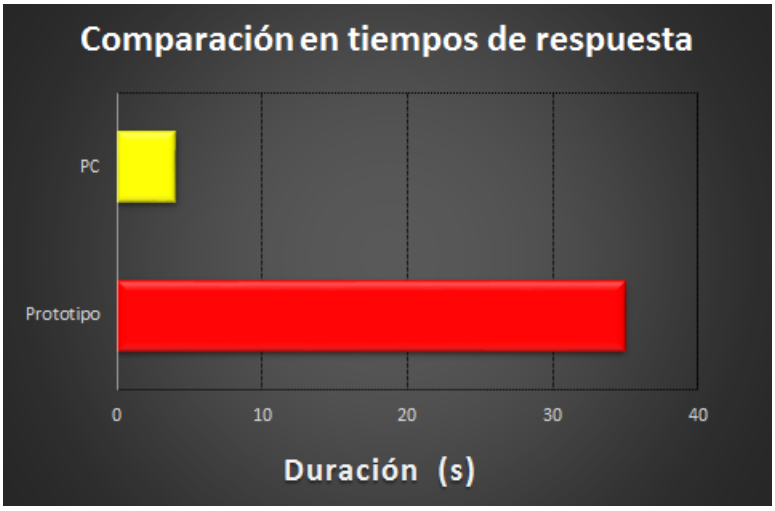
Figura 37. Efectividad en relación al tamaño de la base de datos.




Fuente: Autores.

En relación a los tiempos de respuesta del dispositivo, se presenta en la figura 38 una comparación del tiempo que se toma realizar todo el proceso en un computador de 4GB de RAM con procesador Intel de tercera generación y en el dispositivo traductor con 512MB de RAM.

Figura 38. Tiempo de respuesta en diferentes dispositivos.




Fuente: Autores.

# 6. CONCLUSIONES

Es de esperar que los valores de los coeficientes LPC que se extraen de cada una de las palabras, tengan magnitudes similares en las palabras iguales y diferencias notables en las palabras distintas. Esto debido a las similitudes y diferencias de los formantes entre palabras iguales o distintas respectivamente.

Teniendo en cuenta de que el algoritmo DTW encuentra el camino óptimo para saber la menor distancia entre puntos en comparación, cuando se evalúan las distancias de todas las palabras frente a una en especial, se obtiene que el valor de menor distancia se encuentre en la posición del vector que corresponde a la misma palabra que se ha mencionado. Aún así, la efectividad de este algoritmo se ve afectada debido a que las palabras que se mencionan no son idénticas en cuanto a duraciones en los fonemas, afectando de la misma manera los valores de los coeficientes, tal y como se aprecia en la siguiente conclusión.

La comparación de los vectores LPC de dos cuadros simultáneos correspondientes a sonidos perceptivamente iguales (por ejemplo dos instancias del fonema /a/) puede realizarse a través de la distancia LPC; es decir, la distancia euclídea entre los vectores LPC. Los resultados se corresponden bastante bien con la apreciación subjetiva, en el sentido de que dos sonidos similares exhiben una distancia LPC pequeña, y en cambio dos sonidos perceptivamente diferentes están separados por una distancia LPC considerable. El enfoque anterior tropieza con una importante dificultad, consistente en que las duraciones de los fonemas en dos emisiones diferentes de una misma palabra no son iguales. Ello implicará una considerable probabilidad de que si la comparación se efectúa entre cuadros simultáneos sus vectores LPC no coincidan por corresponder a fonemas diferentes. Esta situación podrá darse a pesar de percibirse las palabras como idénticas.

El número de coeficientes LPC que se procesan es un parámetro relevante, ya que un valor menor a 10 significa una baja precisión de representación del espectro de la señal, mientras que se demuestra que un total de 16 coeficientes es lo suficientemente bueno como para lograr reconocer las palabras. Valores mayores a estos pueden generar lecturas de formantes no deseados, los cuales no corresponden a la información real de la señal y un incremento en el procesamiento de las palabras.

Teniendo en cuenta que son 15 comandos a reconocer, y que en la base de datos son necesarias como mínimo alrededor de 30 muestras de grabación por cada uno de los 15 comandos, el procesamiento del dispositivo puede no llegar a ser rápido debido a que adicional a la cantidad de palabras que se tengan almacenadas, estas deben ser repartidas en segmentos y analizar un total de 16 coeficientes por cada uno de ellos.Esto quiere decir que son bastantes los datos que se tienen que procesar para poder traducir a lengua de señas la palabra que se ha mencionado.

En base a que el hardware no es dedicado esencialmente al trabajo de procesar las muestras de voz, hace que este trabajo se torne algo lento en el dispositivo, ya que el lenguaje de programación de python esta montado sobre un sistema operativo que consume una gran cantidad de memoria RAM. Cabe mencionar que la tarjeta posee tan solo 512 MB de RAM, que son 100 por ciento utilizados según los monitores de la tarjeta mientras esta ejecuta las tareas de reconocimiento.

Tal y como se menciona anteriormente, subjetivamente el método de coeficientes de predicción lineal es realmente efectivo para cálculos de distancia con vectores; sin embargo, objetivamente no es tan bueno para el reconocimiento de palabras aisladas utilizando comparación de patrones o plantillas. Para lo que sí esta demostrado es que son realmente útiles en la síntesis de voz o en la codificación, por tal motivo se recomienda trabajar con otros métodos que resulten mas efectivos en el reconocimiento.

Aunque el dispositivo es pequeño, puede generar molestias al momento de portarlo debido a que las personas están acostumbradas a cargar dispositivos mas pequeños y de peso ligero como los reloj. Esto debido a que la implementación del dispositivo esta realizada con hardware ya desarrollado con ciertas dimensiones a las cuales se tuvo que acomodar para poder realizar el dispositivo. Cabe mencionar que se diseño de tal manera en que se optimizara el espacio dentro de la caja que lo contiene.

La falta de conocimiento en la implementación de hardware dedicado como microprocesadores de tareas específicas y de dispositivos de montaje superficial (característicos por ser pequeños), causa que se opte por compra de dispositivos ya implementados con ciertas limitaciones en memoria tal y como la tarjeta raspberry pi zero. Cabe mencionar que hay tipos de estas tarjetas con prestaciones mucho mas altas, pero no se optaron por estas debido al tamaõ y costos de estas.

Las limitaciones en los componentes pueden generar inconvenientes en el funcionamiento del sistema, por ejemplo el caso de la tarjeta de sonido C-Media la cual viene definida para una frecuencia de muestreo de 44100Hz. Este valor genera gran cantidad de muestras que procesar, teniendo en cuenta la cantidad de datos que deben ser analizados por la tarjeta raspberry.

Una frecuencia de muestreo elevada se traduce en un mayor número de muestras por segmento, causando un incremento en el número de iteraciones en el pre-procesamiento. De igual forma, Un número elevado de coeficientes LPC implica un número elevado de iteraciones en el proceso de extracción de características, lo cual conlleva a obtener valores de coeficientes LPC altos. Estas características causan que se use más espacio en memoria y recursos en las operaciones matemáticas correspondientes, produciendo limitaciones en la implementación de hardware.

El ruido es un factor importante al momento de grabar y realizar pruebas, ya que el sistema aunque posee filtrado en hardware y software, el dispositivo es susceptible a este al momento de reconocer las palabras.

El cambio de pluj del micrófono para la entrada de sonido de la tarjeta C-Media, es un proceso obligado, ya que este tipo de tarjeta de sonido no es compatible con plugs de 4 polos. Por esto motivo tuvo que cambiarse a uno de 3 polos.

Para las configuraciones que se realizaron en la tarjeta raspberry, fue estrictamente necesario la compra de un hub alimentado USB, ya que la tarjeta tan solo posee un puerto de intercambio de datos por microUSB. El Hub tuvo que ser alimentado debido a que la tarjeta solo reconocía un dispositivo conectado cuando este no estaba enchufado a la corriente y en ocasiones se tuvo que conectar a este hub, teclados, tarjetas wifi y otros.

Para la implementación de la base de datos se vio útil el uso de herramientas de software para el manejo de audio tal y como lo es audacity, en la cual se realizó mejoramiento de las señales de voz almacenadas. Allí era posible minimizar efectos de distorsión y ruido.

Para la visualización de las imágenes se usó una pantalla con tecnologÃa TFT tipo resistiva, la cual se caracteriza por su buena definición y bajo costo, sin embargo la sensibilidad de esta es baja debido a su tipo resistivo. Por lo tanto se recomienda trabajar con pantallas capacitivas o de una tecnología mejor.

El consumo de corriente de todos los dispositivos no supera los 500 mA, por lo tanto no se vio necesario utilizar fuentes de alimentación costosas. Por este motivo se opta por una batería comúnmente utilizada en celulares la cual cumple con todos los requerimientos exigidos por los elementos y adicional a esto es de bajo costo.

La validación del funcionamiento del sistema con la comunidad sordomuda no fue posible realizarla debido a que el dispositivo no es 100 porciento efectivo, lo cual no es apto para estudios de beneficios e impacto en la sociedad. Sin embargo si se considera que desarrollos como este, puede a futuro generar grandes cambios en la forma de comunicación entre un oyente común y una persona sorda.

# 7. RECOMENDACIONES Y TRABAJO A FUTURO

- Se propone trabajar con métodos diferentes al LPC para evaluar el rendimiento del sistema, así como métodos de identificación distintos al DTW, de manera en que se pueda medir la eficacia a de uno respecto al otro a la hora de comparar comandos.

- Aumentar el número de palabras a identificar por medio de procesamiento de voz, de modo en que el prototipo sea más robusto y completo.

- Realizar un diseño de estructura mucho más portable en donde la persona que lo porta se sienta más comoda de llevarlo consigo.

- Implementar el prototipo sobre un hardware mucho más potente y robusto que sea capaz de procesar los datos de una manera mucho más rápida, mermando el tiempo de reconocimiento de la palabra.

- En caso de no trabajar con tarjetas programables, se recomienda realizar la implementación de un módulo de reconocimiento con hardware totalmente dedicado a dicha tarea.

- Realizar diseõ de estructura por medio de equipos de diseõ e impresión en 3D, las cuales pueden generar dispositivos mas pequeños y vistosos.

- Usar elementos de visualización mucho más definidos y con una sensibilidad mayor a las pantallas resistivas, por ejemplo las pantallas de tipo capacitivas u OLED.

- Implementar otro método de portabilidad del dispositivo más funcional y de aspecto elegante.

- Adicionar movimiento a las imágenes, de modo que la persona que observa la imágen, entienda mucho mejor.

# ANEXO A: CÓDIGOS

## • Captación de señal de voz

```
import wave
import pyaudio

def grabar():
    CHUNK = 8192
    FORMAT = pyaudio.paInt16
    CHANNELS = 1
    RATE = 44100
    RECORD_SECONDS = 2
    WAVE_OUTPUT_FILENAME = "Palabra.wav"
    p = pyaudio.PyAudio()
    stream = p.open(format=FORMAT,
                    channels=CHANNELS,
                    rate=RATE,
                    input=True,
                    input_device_index=1,
                    frames_per_buffer=CHUNK)

    print("* GRABANDO....")
    print("*** GRABACION FINALIZADA...")
    return
```

## • Preénfasis y Extracción de Características:

```
import numpy as np
from scipy.io.wavfile import read
from scipy.signal import lfilter
import matplotlib.pyplot as plt



def pre_enfasis(path):
    (fs, s) = read(path)
    maxp = abs(max(s))
    sn = s/(float(maxp))
    nSamples = np.int32(0.025 * fs)
    overlap = np.int32(0.01 * fs)
    nFrames = np.int32(np.ceil(len(sn) / (nSamples - overlap)))
    # zero padding to make signal length long enough to have nFrames
    padding = ((nSamples - overlap) * nFrames) - len(sn)
```

```
if padding > 0:
    signal = np.append(sn, np.zeros(padding))
else:
    signal = sn
segment = np.empty((nSamples, nFrames))
start = 0
energia_trama = np.zeros(nFrames)
zero_cross = np.zeros(nFrames)

for i in range(nFrames):
    segment[:, i] = signal[start:start + nSamples]
    E = 0
    Zcc = 0
    for j in range(nSamples):
        E = E + (segment[j, i]) ** 2
        Zcc = Zcc + abs(np.sign(segment[j, i]) - np.sign(segment[j - 1, i]))
    zero_cross[i] = Zcc
    energia_trama[i] = E
    start = (nSamples - overlap) * i
trama = []
for st in range(nFrames):
    if energia_trama[st] >= 10 and zero_cross[st] <= 300:
        trama.append(st)

iv = np.empty(len(trama)*nSamples)

sennal = np.empty((nSamples, len(trama)))
for add in range(len(trama)):
    a0 = 1.0
    b0 = 1.0
    b1 = -0.95
    a = np.array([a0])
    b = np.array([b0, b1])
    # salida del fitro pre-enfasis
    out = lfilter(b, a, segment[:, trama[add]])
    sennal[:, add] = out
    '''
    init = nSamples * add
    iv[init:init+nSamples] = segment[:, trama[add]]
    print init


print len(trama)
time = np.linspace(0, 2, len(energia_trama))
```

```
    plt.figure(1)
    time1 = np.linspace(0, 2, len(signal))
    plt.subplot(3, 1, 1)
    plt.plot(time1, signal, 'blue')
    plt.xlabel('Tiempo')
    plt.ylabel('Amplitud')
    plt.subplot(3, 1, 2)
    plt.plot(time, energia_trama, 'r')
    plt.xlabel('Tiempo')
    plt.ylabel('Energia')
    plt.subplot(3, 1, 3)
    plt.plot(iv, 'g')
    plt.xlabel('Tiempo')
    plt.ylabel('cruces por cero')
    plt.show()
    '''
    return fs, sennal
```

- **Análisis LPC:**

```
# -*- coding: utf-8 -*-
"""
Created on Sat Feb 27 22:01:37 2016

@author: ORCHISAMA
"""

#calculate LPC coefficients from sound file
from __future__ import division
import numpy as np
import matplotlib.pyplot as plt


def autocorr(x):
    n = len(x)
    variance = np.var(x)
    x = x - np.mean(x)
    r = np.correlate(x, x, mode = 'full')[-n:] #n numbers from last index (l-n to l)
    result = r/(variance*(np.arange(n, 0, -1)))
    return result


def createSymmetricMatrix(acf,p):
    R = np.empty((p,p))
```

```
    for i in range(p):
        for j in range(p):
            R[i,j] = acf[np.abs(i-j)]
    return R


def lpc(s,fs,p):

    #divide into segments of 25 ms with overlap of 10ms
    nSamples = np.int32(0.025*fs)
    overlap = np.int32(0.01*fs)
    nFrames = np.int32(np.ceil(len(s)/(nSamples-overlap)))

    #zero padding to make signal length long enough to have nFrames
    padding = ((nSamples-overlap)*nFrames) - len(s)
    if padding > 0:
        signal = np.append(s, np.zeros(padding))
    else:
        signal = s
    segment = np.empty((nSamples, nFrames))
    start = 0
    for i in range(nFrames):
        segment[:,i] = signal[start:start+nSamples]
        start = (nSamples-overlap)*i

    #calculate LPC with Yule-Walker
    lpc_coeffs = np.empty((p, nFrames))
    for i in range(nFrames):
        acf = autocorr(segment[:,i])
#         plt.figure(1)
#         plt.plot(acf)
#         plt.xlabel('lags')
#         plt.ylabel('Autocorrelation coefficients')
#         plt.axis([0, np.size(acf), -1, 1])
#         plt.title('Autocorrelation function')
#         break
        r = -acf[1:p+1].T
        R = createSymmetricMatrix(acf,p)
        lpc_coeffs[:,i] = np.dot(np.linalg.inv(R),r)
        lpc_coeffs[:,i] = lpc_coeffs[:,i]/np.max(np.abs(lpc_coeffs[:,i]))

    return lpc_coeffs
```

- **Algoritmo DTW:**

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

from __future__ import absolute_import, division
import numbers
import numpy as np
from collections import defaultdict

try:
    range = xrange
except NameError:
    pass


def fastdtw(x, y, radius=1, dist=None):
    ''' return the approximate distance between 2 time series with O(N)
        time and memory complexity

        Parameters
        ----------
        x : array_like
            input array 1
        y : array_like
            input array 2
        radius : int
            size of neighborhood when expanding the path. A higher value will
            increase the accuracy of the calculation but also increase time
            and memory consumption. A radius equal to the size of x and y will
            yield an exact dynamic time warping calculation.
        dist : function or int
            The method for calculating the distance between x[i] and y[j]. If
            dist is an int of value p > 0, then the p-norm will be used. If
            dist is a function then dist(x[i], y[j]) will be used. If dist is
            None then abs(x[i] - y[j]) will be used.

        Returns
        -------
        float
            the approximate distance between the 2 time series

        Examples
        --------
        >>> import numpy as np
        >>> import fastdtw
```

```
        >>> x = np.array([1, 2, 3, 4, 5], dtype='float')
        >>> y = np.array([2, 3, 4], dtype='float')
        >>> fastdtw.fastdtw(x, y)
        (2.0, [(0, 0), (1, 0), (2, 1), (3, 2), (4, 2)])
    '''

    if not isinstance(x, np.ndarray):
        x = np.array(x)
    if not isinstance(y, np.ndarray):
        y = np.array(y)
    if x.ndim == y.ndim > 1 and x.shape[1] != y.shape[1]:
        raise ValueError('second dimension of x and y must be the same')
    if isinstance(dist, numbers.Number) and dist <= 0:
        raise ValueError('dist cannot be a negative integer')

    if dist is None:
        dist = __difference
    elif isinstance(dist, numbers.Number):
        dist = __norm(p=dist)

    return __fastdtw(x, y, radius, dist)


def __difference(a, b):
    return abs(a - b)


def __norm(p):
    return lambda a, b: np.linalg.norm(a - b, p)


def __fastdtw(x, y, radius, dist):
    min_time_size = radius + 2

    if len(x) < min_time_size or len(y) < min_time_size:
        return dtw(x, y, dist=dist)

    x_shrinked = __reduce_by_half(x)
    y_shrinked = __reduce_by_half(y)
    distance, path = \
        __fastdtw(x_shrinked, y_shrinked, radius=radius, dist=dist)
    window = __expand_window(path, len(x), len(y), radius)
    return dtw(x, y, window, dist=dist)
```

```
def dtw(x, y, window=None, dist=lambda a, b: abs(a - b)):
    ''' return the distance between 2 time series without approximation

        Parameters
        ----------
        x : array_like
            input array 1
        y : array_like
            input array 2
        dist : function or int
            The method for calculating the distance between x[i] and y[j]. If
            dist is an int of value p > 0, then the p-norm will be used. If
            dist is a function then dist(x[i], y[j]) will be used. If dist is
            None then abs(x[i] - y[j]) will be used.

        Returns
        -------
        float
            the approximate distance between the 2 time series

        Examples
        --------
        >>> import numpy as np
        >>> import fastdtw
        >>> x = np.array([1, 2, 3, 4, 5], dtype='float')
        >>> y = np.array([2, 3, 4], dtype='float')
        >>> fastdtw.dtw(x, y)
        (2.0, [(0, 0), (1, 0), (2, 1), (3, 2), (4, 2)])
    '''
    len_x, len_y = len(x), len(y)
    if window is None:
        window = [(i, j) for i in range(len_x) for j in range(len_y)]
    window = ((i + 1, j + 1) for i, j in window)
    D = defaultdict(lambda: (float('inf'),))
    D[0, 0] = (0, 0, 0)
    for i, j in window:
        dt = dist(x[i-1], y[j-1])
        D[i, j] = min((D[i-1, j][0]+dt, i-1, j), (D[i, j-1][0]+dt, i, j-1),
                      (D[i-1, j-1][0]+dt, i-1, j-1), key=lambda a: a[0])
    path = []
    i, j = len_x, len_y
    while not (i == j == 0):
        path.append((i-1, j-1))
```

```
        i, j = D[i, j][1], D[i, j][2]
    path.reverse()
    return (D[len_x, len_y][0], path)


def __reduce_by_half(x):
    return [(x[i] + x[1+i]) / 2 for i in range(0, len(x) - len(x) % 2, 2)]


def __expand_window(path, len_x, len_y, radius):
    path_ = set(path)
    for i, j in path:
        for a, b in ((i + a, j + b)
                     for a in range(-radius, radius+1)
                     for b in range(-radius, radius+1)):
            path_.add((a, b))

    window_ = set()
    for i, j in path_:
        for a, b in ((i * 2, j * 2), (i * 2, j * 2 + 1),
                     (i * 2 + 1, j * 2), (i * 2 + 1, j * 2 + 1)):
            window_.add((a, b))

    window = []
    start_j = 0
    for i in range(0, len_x):
        new_start_j = None
        for j in range(start_j, len_y):
            if (i, j) in window_:
                window.append((i, j))
                if new_start_j is None:
                    new_start_j = j
            elif new_start_j is not None:
                break
        start_j = new_start_j

    return window
```

- **Interfaz de Usuario:**

```
# -*- coding: utf-8 -*-

# Form implementation generated from reading ui file 'Interfaz_usuario.ui'
#
```

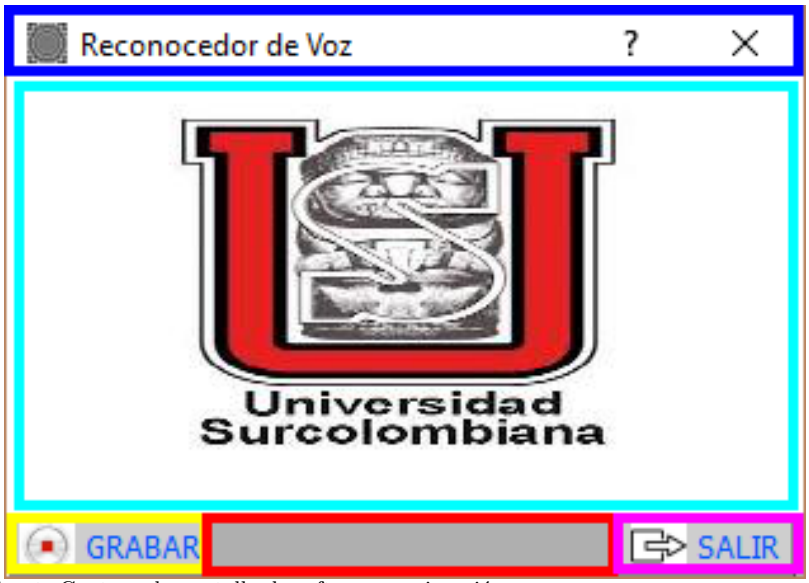
```
# Created by: PyQt4 UI code generator 4.11.4
#
# WARNING! All changes made in this file will be lost!

from PyQt4 import QtCore, QtGui

try:
    _fromUtf8 = QtCore.QString.fromUtf8
except AttributeError:
    def _fromUtf8(s):
        return s

try:
    _encoding = QtGui.QApplication.UnicodeUTF8
    def _translate(context, text, disambig):
        return QtGui.QApplication.translate(context, text, disambig, _encoding)
except AttributeError:
    def _translate(context, text, disambig):
        return QtGui.QApplication.translate(context, text, disambig)

class Ui_DialogGUI(object):
    def setupUi(self, DialogGUI):
        DialogGUI.setObjectName(_fromUtf8("DialogGUI"))
        DialogGUI.resize(314, 194)
        icon = QtGui.QIcon()
        icon.addPixmap(QtGui.QPixmap(_fromUtf8("Imagenes/fractal.jpg")), QtGui.QIcon
        DialogGUI.setWindowIcon(icon)
        DialogGUI.setStyleSheet(_fromUtf8("background-color : rgb(175, 175, 175)"))
        self.pushButton_Grabar = QtGui.QPushButton(DialogGUI)
        self.pushButton_Grabar.setGeometry(QtCore.QRect(0, 170, 81, 25))
        self.pushButton_Grabar.setAutoFillBackground(True)
        self.pushButton_Grabar.setStyleSheet(_fromUtf8("background-color : rgb(206,
"font: 75 9pt \"Tahoma\";\n"
"color:rgb(0, 85, 255);"))
        icon1 = QtGui.QIcon()
        icon1.addPixmap(QtGui.QPixmap(_fromUtf8("Imagenes/Grabar.jpg")), QtGui.QIcon
        self.pushButton_Grabar.setIcon(icon1)
        self.pushButton_Grabar.setIconSize(QtCore.QSize(22, 38))
        self.pushButton_Grabar.setAutoRepeatDelay(3)
        self.pushButton_Grabar.setObjectName(_fromUtf8("pushButton_Grabar"))
        self.label_imagen = QtGui.QLabel(DialogGUI)
        self.label_imagen.setGeometry(QtCore.QRect(0, 0, 314, 171))
        self.label_imagen.setText(_fromUtf8(""))
        self.label_imagen.setObjectName(_fromUtf8("label_imagen"))
```

```
        self.label_texto = QtGui.QLabel(DialogGUI)
        self.label_texto.setGeometry(QtCore.QRect(0, 0, 71, 25))
        self.label_texto.setStyleSheet(_fromUtf8("font: 75 10pt \"Tahoma\";\n"
"background-color: rgba(255, 255, 255, 0);"))
        self.label_texto.setText(_fromUtf8(""))
        self.label_texto.setObjectName(_fromUtf8("label_texto"))
        self.pushButtonSalir = QtGui.QPushButton(DialogGUI)
        self.pushButtonSalir.setGeometry(QtCore.QRect(240, 170, 71, 25))
        self.pushButtonSalir.setStyleSheet(_fromUtf8("background-color : rgb(206, 20
"font: 75 9pt \"Tahoma\";\n"
"color:rgb(0, 85, 255);"))
        icon2 = QtGui.QIcon()
        icon2.addPixmap(QtGui.QPixmap(_fromUtf8("Imagenes/icono_Salir.jpg")), QtGui.
        self.pushButtonSalir.setIcon(icon2)
        self.pushButtonSalir.setIconSize(QtCore.QSize(22, 50))
        self.pushButtonSalir.setObjectName(_fromUtf8("pushButtonSalir"))
        self.label_Grabar = QtGui.QLabel(DialogGUI)
        self.label_Grabar.setGeometry(QtCore.QRect(80, 170, 151, 25))
        self.label_Grabar.setStyleSheet(_fromUtf8("font: 75 9pt \"Tahoma\";\n"
"color:rgb(0, 0, 255)"))
        self.label_Grabar.setText(_fromUtf8(""))
        self.label_Grabar.setObjectName(_fromUtf8("label_Grabar"))

        self.retranslateUi(DialogGUI)
        QtCore.QMetaObject.connectSlotsByName(DialogGUI)

    def retranslateUi(self, DialogGUI):
        DialogGUI.setWindowTitle(_translate("DialogGUI", "Reconocedor de Voz", None)
        self.pushButton_Grabar.setText(_translate("DialogGUI", "GRABAR", None))
        self.pushButtonSalir.setText(_translate("DialogGUI", "SALIR", None))
```

## ANEXO B: MANUAL DE FUNCIONAMIENTO

El funcionamiento del dispositivo traductor móvil es bastante sencillo, ya que se diseño de tal manera en que cualquier usuario sea capaz de manejarla sin problema. El panel principal que se puede observar desde la pantalla es el que se muestra en la figura 28.

Figura 39. Panel principal de dispositivo traductor.




Fuente:Captura de pantalla de software en ejecución.

Reconocedor de voz ? X: Es la barra de título del software en la cual se encuentran los botones de control de ayuda y cerrar.

Panel de software: Panel en donde se despliega la imágen en lengua de señas de la palabra mencionada.

Grabar: Es el boton que se utiliza para iniciar la grabación en el dispositivo.

Espacio de información: Espacio reservado para dar información del proceso del sistema.

Salir: Boton para salir del sofware.

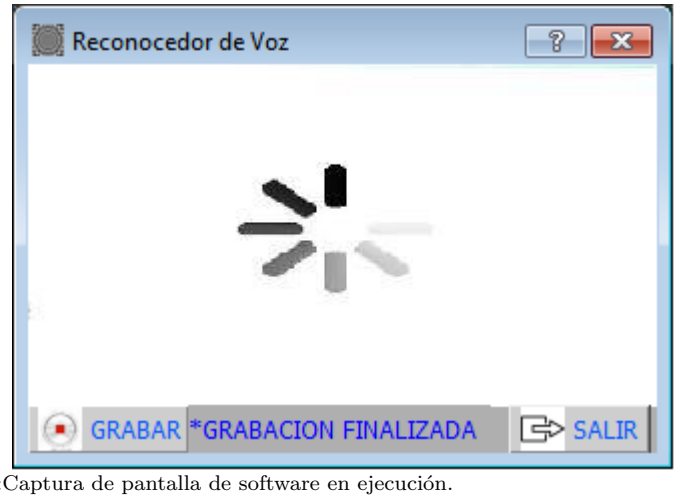
Para poner en funcionamiento el traductor simplemente se debe dar clic en el botón Grabar y este empezará a capturar la señal que el micrófono este dando, en este caso la señal de voz. Cuando se da clic en dicho botón, tanto en el espacio de información como en el panel se muestra la palabra GRABANDO, la cual indica que ya se puede empezar a hablar. Lo mencionado anteriormente se puede observar en la figura 47.

Figura 40. Capturando señal de voz.


Fuente:Captura de pantalla de software en ejecución.

Luego de que el software captura la voz, este la procesa sacando los coeficientes de dicha palabra, para compararlos y calcular distancia mínima entre toda las palabras de la base de datos y así mostrar efectivamente el comando que se menciona. Mientras el dispositivo esta procesando los datos, en el panel se despliega una secuencia de imagenes de espera, para luego dar paso a la imagen de la palabra que se ha mencionado. Esto se puede ver en las figuras 48 y 49.

Figura 41. En espera del procesamiento.




Fuente:Captura de pantalla de software en ejecución.

Figura 42. Comando identificado.


Fuente:Captura de pantalla de software en ejecución.

Tal y como se puede observar en la anterior figura, el espacio de información muestra de que la grabación ha sido finalizada, así como todo el procesamiento de la señal de voz. Pasado todo esto, la persona sordomuda puede facilmente ver la imagen que se despliega en la pantalla y entender lo que el hablante le esta tratando de comunicar.