



**UNIVERSIDAD NACIONAL DE CHIMBORAZO
FACULTAD DE INGENIERÍA**

ESCUELA DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES

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

TRABAJO DE GRADUACIÓN

Título del proyecto:

**DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE DETECCIÓN DE
RETARDO SIMPLE DE LENGUAJE POR MEDIO DE VOZ PARA NIÑOS DE
4 AÑOS.**

Autores:

Nayla Soledad Maldonado Bonifáz
Geovanny Javier Uchuari Montoya

Director:

Ing. Giovanni Cuzco

Riobamba – Ecuador

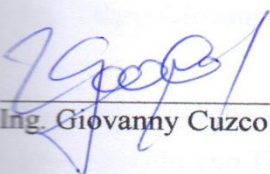
Año 2017

INFORME DEL TUTOR

Yo, Ing. Edgar Giovanni Cuzco Silva, en mi calidad de Tutor del trabajo de graduación titulado: **“DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE DETECCIÓN DE RETARDO SIMPLE DE LENGUAJE POR MEDIO DE VOZ PARA NIÑOS DE 4 AÑOS”**, tengo a bien informar que el mencionado trabajo cumple con los requisitos exigidos para ser expuestos al público, luego de ser evaluado por el tribunal designado.

Riobamba, agosto del 2017

Atentamente


Ing. Giovanni Cuzco


Firma


Firma


Firma

CERTIFICACIÓN INVESTIGACIÓN

Los miembros del Tribunal de Graduación del proyecto de investigación de título:
**“DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE DETECCIÓN DE
RETARDO SIMPLE DE LENGUAJE POR MEDIO DE VOZ PARA NIÑOS DE 4
AÑOS.”**

Presente por:

Nayla Soledad Maldonado Bonifáz

Geovanny Javier Uchuari Montoya

Y dirigida por:

Ing. Edgar Giovanni Cuzco Silva

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

Para constancia de lo expuesto firman:

Ing. Juan Carlos Cepeda
Presidente del Tribunal


Firma

Ing. Giovanni Cuzco
Director del Proyecto


Firma

Ing. Cristian Márquez
Miembro del Tribunal


Firma

AUTORÍA DE INVESTIGACIÓN

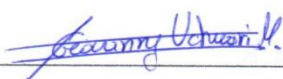
La responsabilidad del contenido de este Proyecto de Graduación, nos corresponde exclusivamente a: **Nayla Maldonado, Geovanny Uchuari e Ingeniero Giovanni Cuzco**; y el patrimonio intelectual de la misma a la Universidad Nacional de Chimborazo.

Autores:



Nayla Maldonado

060415208-2



Geovanny Uchuari

190043526-2

AGRADECIMIENTO

A Dios por forjar mi camino, bendecirme y permitir llegar a este punto de mi carrera y hacer un sueño realidad.

A mi madre Carla Bonifáz por ser un pilar importante y por la devoción que tiene a sus hijos.

A mi hermano Alexis Maldonado que con su ejemplo y apoyo me alienta cada día.

A mis abuelitos Martha Orna y Carlos Bonifáz quienes junto a mi madre me han demostrado su apoyo incondicional en momentos de debilidad.

A mis tíos Roberto y Anita por todas sus manifestaciones de afecto

A mis primos Sebastián y Galo por compartir momentos significativos conmigo y por estar siempre presentes.

Nayla Maldonado

DEDICATORIA

*El presente trabajo de Investigación
va dedicado con todo mi cariño a:*

*Mi madre Carla Bonifáz por todo el
sacrificio y fortaleza para salir
adelante sin importar los obstáculos.*

*Mi hermano por su compañía,
constancia y empeño de ser mejor.*

*Mis abuelitos por ser un ejemplo de
fortaleza, integridad y
responsabilidad*

*Tíos y primos por sus palabras de
aliento y buenos deseos.*

Nayla Maldonado

AGRADECIMIENTO

Agradezco a Dios por haberme guiado y acompañado a lo largo de mis estudios, por brindarme una vida llena de aprendizajes y experiencias.

A mi madre María Montoya, quien con su confianza y ejemplo de superación me apoyó para hoy ver mi meta alcanzada.

A mi padre Jorge Uchuari por su apoyo y orientación para iluminar mi camino y poder realizar mis estudios.

Por haberme forjado como la persona que soy actualmente, mi logro se los debo a ustedes.

Geovanny Uchuari

DEDICATORIA

El presente proyecto dedico a mis padres Jorge Uchuari y María Montoya, por los ejemplos de perseverancia y constancia que los caracteriza, por todo el cariño y confianza al educarme para llegar a ser un hombre de bien.

A mis hermanos y sobrinos quienes creyeron en mí y me apoyaron en cada momento de mis estudios para lograr mi objetivo, de formarme y ser un profesional.

Geovanny Uchuari

ÍNDICE GENERAL

ÍNDICE GENERAL	viii
ÍNDICE DE FIGURAS	xiv
ÍNDICE DE TABLAS	xv
ÍNDICE DE ANEXOS	xvi
RESUMEN.....	xvii
INTRODUCCIÓN.....	1
OBJETIVOS	3
GENERAL	3
ESPECÍFICOS	3
1 FUNDAMENTACIÓN TEÓRICA.....	4
1.1 Antecedentes.....	4
1.2. La Voz Humana	6
1.2.1. Aparato Fonador	6
1.2.2. Articulación	6
1.2.3. Cualidades de la voz	7
1.2.4. Retardo Simple de Lenguaje	7
1.2.5. Causas que provocan el Retraso Simple de Lenguaje	7
1.3. Reconocimiento de Voz.....	8
1.3.1 Etapas	8
1.4. Componentes	8
1.4.1 Micrófono	8
1.4.1 .1 Características de los Micrófonos	8
1.4.2 Raspberry pi3.....	9
1.4.3 LCD	9
1.4.4 Software.....	10
1.4.4.1Matlab 2016.....	10
1.4.4.2 Python 2.7	10
1.4.4.3 Librerías de Python 2.7	10
1.4.5 Métodos y Etapas para el Reconocimiento de Voz	11
1.4.5.1 Ingreso de la Voz	11
1.4.5.2 Pre-procesamiento.....	11
1.4.5.3 Segmentación.....	11
1.4.5.4 Ventaneo	11
1.4.5.5 Filtrado de la señal	12
1.4.5.6 LPC (Linear Prediction Coefficients).....	12
1.4.5.7 Distancia Euclidiana	13
2. METODOLOGÍA.....	13

2.2.1.	Métodos	14
2.2.1.1.	Analítico/Deductivo	14
2.2.2.	Técnicas	14
2.2.2.1.	Observación	14
2.3.	Población y muestra	14
2.4.	Hipótesis	15
2.5.	Operacionalización de variables	15
2.6.	Procedimiento	15
2.6.1	Elaboración del Sistema	15
3.	RESULTADOS Y DISCUSIÓN	21
3.1.1	Interfaz gráfica para base de datos	22
3.1.2	Interfaz para la detección de la palabra mediante la voz	22
3.1.2.1	Evaluación	22
3.3.2	Discusión	33
4.	CONCLUSIONES Y RECOMENDACIONES	34
4.1.	Conclusiones	34
4.2.	Recomendaciones	35
5.	BIBLIOGRAFÍA	36
6.	APÉNDICES Y ANEXOS	38

ÍNDICE DE FIGURAS

Figura 1. Aparato Fonador.....	6
Figura 2. Etapas de Reconocimiento de Voz	8
Figura 3. Partes de la tarjeta Raspberry Pi3	9
Figura 4. Kit LCD.....	10
Figura 5. Proceso para el desarrollo del proyecto	16
Figura 6. Curva de Chi Cuadrado	21
Figura 7. Interfaz para generar la Base de Datos	22
Figura 8. Ventana Principal de la Interfaz	22
Figura 9. Ventana de Evaluación	23
Figura 10. Resultados de Evaluación	23
Figura 11. Implementación del sistema	23
Figura 12. Muestras de las señales de la palabra chocolate.....	27
Figura 13. Producto de señales con ventana de Hamming	28
Figura 14. Normalización de señales	28
Figura 15. Filtrado de las muestras de señales de voz.....	28
Figura 16. Comparación de coeficientes LPC.....	29
Figura 17. Foto de Usuario 1	29
Figura 18. Foto de Usuario 2	30
Figura 19. Foto de Usuario 3	30
Figura 20. Foto de Usuario 4	31

ÍNDICE DE TABLAS

Tabla 1. Operacionalización de variables.	15
Tabla 2. Aciertos y Errores por cada palabra	20
Tabla 3. Resumen de resultados de muestras.	20
Tabla 4. Resultado de Chi Cuadrado.	21
Tabla 5. Coeficientes de la palabra chocolate de la base de datos.....	24
Tabla 6. Patrón de las 5 primeras palabras almacenadas.....	25
Tabla 7. Patrón de las 5 últimas palabras.....	26
Tabla 8. Lista de palabras a comparar.....	26
Tabla 9. Comparación de coeficientes de predicción lineal (LPC)	27
Tabla 10. Resultados del Usuario 1	29
Tabla 11. Resultados de Usuario 2	30
Tabla 12. Resultados de Usuario 3	31
Tabla 13. Resultados de Usuario 4	31
Tabla 14. Porcentaje de efectividad del sistema de cada palabra de detección	32
Tabla 15. Resultados de los usuarios que utilizaron el sistema.	32
Tabla 16. Resultados de los usuarios que utilizaron el sistema.	33

ÍNDICE DE ANEXOS

ANEXO 1 Listado de niños que forman la base de datos	38
ANEXO 2 Listado de niños que realizan las pruebas	39
ANEXO 3 Glosario	39
ANEXO 4 Test	40
ANEXO 5 Código	41

RESUMEN

El presente proyecto describe el diseño y construcción de un sistema de detección de retardo simple de lenguaje por medio de voz para niños de 4 años que permite al usuario grabar diez palabras para que sean comparadas con la base de datos generada previamente y determinar si la pronunciación es correcta.

El sistema consta de dos entornos gráficos amigables, el primero que permite adquirir los audios de 11 niños, y la segunda interfaz presenta un menú con las palabras que deben ser grabadas, cada una de ellas es ingresada a la tarjeta Raspberry Pi3. A los audios almacenados en la base de datos se les aplica el preprocesamiento, se calcula los coeficientes LPC para luego obtener un promedio que representarán los patrones de cada palabra, que se utilizan para realizar la comparación con el audio ingresado por el usuario utilizando el cálculo de distancia Euclidiana.

Los resultados determinan el número de palabras correctas e incorrectas, se utiliza la prueba de José Fco. Cervera y Amparo Ygual, “Evaluación e intervención en niños con trastornos fonológicos y dificultad de aprendizaje de la lectura y escritura” (Fco, e Ygual, 2001), específicamente la parte de trastorno fonológico, mostrando dichos resultados en un gráfico de barras.

Las pruebas se realizaron en la Unidad Educativa Miguel Ángel León a los estudiantes de inicial 2 del paralelo D en ambientes controlados, para determinar el funcionamiento del sistema al trabajar con audios en tiempo real.


Palabras Claves: Procesamiento del habla, voz humana, filtros FIR, grabación de audio, cancelación de ruido.

INTRODUCCIÓN

ABSTRACT

The present project describes the design and construction of a simple speech language delay detection system for children of 4 years that allows the user to record ten words to be compared with the previously generated database and to determine if the pronunciation it's correct. The system consists of two friendly graphical environments, the first one that allows to acquire the audios of 11 children, and the second interface presents a menu with the words that must be recorded, each one of them is entered to the card Raspberry Pi3. The audio stored in the database is preprocessed, the LPC coefficients are calculated and then an average that will represent the patterns of each word, which are used to make the comparison with the audio entered by the user using the calculation Away Euclidian. The results determine the number of correct and incorrect words, using the test of José Fco. Cervera and Amparo Ygual, "Evaluation and intervention in children with phonological disorders and difficulty in learning to read and write" (Fco, and Ygual, 2002), specifically the part of phonological disorder, showing such results in a bar graph. The tests were carried out in the Miguel Ángel León Educational Unit to the students of initial 2 of class D in controlled environments, to determine the operation of the system when working with audios in real time.

Keywords: speech processing, human voice, FIR filters, audio recording, noise cancellation



Reviewed by: Abarca, Sandra

ENGLISH LANGUAGE TEACHER



INTRODUCCIÓN

El habla es un proceso de comunicación de gran importancia entre los seres humanos, que va desarrollándose a partir del primer año de nacimiento, por lo que es importante brindar una atención temprana para determinar si existe la presencia o no de un trastorno de lenguaje en los niños. Se requiere del apoyo, atención e interés de los padres para una adecuada estimulación del lenguaje. (Tonato, Y 2015)

El retardo simple de lenguaje se trata de un desfase cronológico en la fonética, vocabulario y sintaxis. Se caracteriza porque los niños presentan ausencia de lenguaje, aparición tardía o permanencia de patrones lingüísticos inferiores al de su edad cronológica que puede provocar problemas en su desarrollo y en su aprendizaje. Se trata de hablantes tardíos que pueden ocasionar alteraciones fonológicas al momento de expresarse o de repetir palabras, sin embargo, en el retardo simple de lenguaje no se ve afectada la comprensión en los niños al escuchar diversas palabras. (Artigas, Rigau y García-Nonell 2008)

En la actualidad se ha conseguido por medio de procesamiento digital de señales desarrollar sistemas de reconocimiento de voz en niños, para el tratamiento de trastornos de lenguaje, específicamente para utilizarlo con escolares que presentan problemas del habla dislalia (Miranda-palma, Camal-Uc, Cen-Magaña, Gonzalez-Segura, Gonzalez-Segura, García y Narvaez-Díaz, 2007). Otras aplicaciones que utilizan la voz de los niños por medio de palabras establecidas permiten el control de un robot móvil. (Jacqueline, Apolo 2017)

Sin embargo, no se cuenta con la presencia de un sistema que ayude a la detección de problemas en el habla de los niños, por lo que en el presente proyecto se desarrolla un sistema de detección de retardo simple de lenguaje para niños de 4 años que otorga la posibilidad de reconocer la correcta pronunciación de palabras acordes a su edad.

Problema

El retardo simple del lenguaje se origina por desconocimiento e indiferencia por parte del entorno social y familiar, siendo este último el principal causante debido a que los padres de familia no permiten el desenvolvimiento y desarrollo en la producción del lenguaje en los niños, se puede producir por factores hereditarios, socioculturales o conflictos familiares, la falta de experiencia no permite estimular a los niños, por esta razón se muestran tímidos, retraídos y presentan problemas en su desarrollo y en su aprendizaje.

Justificación

Los padres de familia y educadores al encontrarse con niños que no poseen una correcta pronunciación de las palabras y que no han sido diagnosticados con anterioridad, desconocen cómo manejar la situación.

Por lo que es indispensable someter a una evaluación del habla a los niños de preescolar, por lo que se crea un sistema para detectar problemas fonológicos en los niños, que se trata de una herramienta necesaria para que los menores puedan de manera recreativa realizar pruebas de diagnóstico sin ninguna presión, lo que permite la detección de retardo simple en niños para que posteriormente realicen el tratamiento respectivo y evitar así problemas futuros en su desarrollo lingüístico.

OBJETIVOS

GENERAL

- Diseñar y construir un sistema para determinar si un niño de 4 años de edad presenta retraso simple de lenguaje.

ESPECÍFICOS

- Desarrollar un sistema que sea amigable para los niños de 4 años.
- Determinar el algoritmo más adecuado de procesamiento de voz para comparar la similitud con palabras establecidas en el test.
- Determinar la presencia de retardo simple de lenguaje en niños de 4 años.
- Realizar las pruebas en ambientes reales para medir la efectividad del sistema.

1 FUNDAMENTACIÓN TEÓRICA

1.1 Antecedentes

Un sistema de reconocimiento de voz tiene la capacidad de decodificar la señal acústica producida por el aparato fonador del usuario. En el año de 1920 se construye un juguete llamado Radio Rex, el perro que al reconocer una sola palabra (su nombre), se movía. (Padilla y Macas, 2012).

En 1952 Davis, Bidulph y Balashek de los laboratorios Bell, desarrollaron un dispositivo totalmente electrónico para reconocimiento de voz que era capaz de identificar diez dígitos considerando la pronunciación de un único usuario. En los años de 1959/1960 Deves y Mathews desarrollan la comparación de palabras iguales pero pronunciadas a diferente velocidad (Arellano, Elena, y Benito, 2003). A partir de los años 60 y 70 se introduce diferentes procedimientos para desarrollar sistemas de reconocimiento del habla continua, entre los que se tiene:

- El desarrollo de algoritmos para poder extraer las características de la señal acústica obtenida.
- Debido a la aparición de la transformada rápida de Fourier se procede a utilizar el modelado de rasgos espectrales para el reconocimiento de palabras.
- Otros sistemas contaban con el reconocimiento mediante la utilización de procesamiento cepstral.
- Para obtener las diferentes velocidades o longitud d segmentos se procede a desarrollar sistemas por medio de técnicas de warping expansión o compresión temporal. (Padilla y Macas, 2012).

A mediados de los años 80, la Agencia de Proyectos de Investigación Avanzada del Departamento de Defensa de los Estados Unidos lanza dos tareas, la primera tarea es de Resource Management que tiene un léxico de 1000 palabras y podía formar oraciones con ellas. La segunda tarea lanzada fue Wall Street Journal que contaba con 60000 palabras. (Padilla, W., Macas, D (2012).

Después de esto se introduce el reconocimiento de voz por medio de modelos probabilísticos como son los Modelos Ocultos de Márkov, además se comienza con el desarrollo de las Redes Neuronales para el reconocimiento de voz.

El presente proyecto de investigación toma como punto de partida el diseño de un sistema de reconocimiento de voz para detección de retardo simple de lenguaje, para lo cual se ha investigado la existencia de proyectos similares o relacionados al tema; por tal motivo se detallan a continuación los siguientes temas:

“UN JUEGO DE GRAVEDAD CON RECONOCIMIENTO DE VOZ PARA NIÑOS CON PROBLEMAS DE LEGUAJE”. - El proyecto elaborado por Carlos Miranda-Palma, Reyna Camal-Uc, José Cen-Magaña, Cinthia Gonzalez-Segura, Universidad Autónoma de Yucatán, utilizando lenguaje C, librerías gráficas SDL y la herramienta HTK, desarrollan un juego electrónico para estimular conocimientos de forma didáctica mediante comandos de voz a niños que tienen problemas de dislalia. (Miranda-palma, Camal-Uc, Cen-Magaña, Gonzalez-Segura, Gonzalez-Segura, García, y Narvaez-Díaz, 2007)

“ESTUDIO DE LOS MODELOS OCULTOS DE MARKOV Y DESARROLLO DE UN PROTOTIPO PARA EL RECONOCIMIENTO AUTOMÁTICO DEL HABLA.”. - El proyecto elaborado por Diana Xismenia Macas Macas y William Alfonso Padilla Pineda, Universidad Politécnica Salesiana, consiste en el estudio de los modelos de Márkov para el diseño de un prototipo de reconocimiento de comandos mediante el habla. Para lo cual se utiliza las herramientas: Hidden Márkov Model Toolkit (HTK) que el software MATLAB ofrece, Julius y clases de Java. (Padilla y Macas, 2012).

“DESARROLLO DE UN VIDEOJUEGO CON RECONOCIMIENTO DE VOZ PARA LA SELECCIÓN DE COLORES Y FIGURAS GEOMÉTRICAS”. - El proyecto elaborado por Luisa Fernanda Parra Espinel, Universidad de San Buenaventura, utilizando la herramienta Unity 3D desarrolla un videojuego educativo para repasar conocimientos de forma didáctica mediante comandos de voz. El video juego se utiliza en sistemas operativos iOS y Windows. (Parra, 2015)

“IMPLEMENTACIÓN DE UN MÓDULO DE RECONOCIMIENTO DE VOZ PARA NIÑOS MEDIANTE EL PROCESAMIENTO DE SEÑALES APLICADO EN UN CASO PRÁCTICO”. - El proyecto elaborado por Katty Beatriz Apolo Díaz y Nelly Jacqueline Coba Castillo, Escuela Superior Politécnica de Chimborazo, se basa en el reconocimiento de voz para el control de un carro robot mediante comandos de voz. Se muestra una interfaz gráfica para la base de

datos y otra para el reconocimiento de voz para niños, este proyecto de desarrolla en el software Matlab. (Jacqueline y Apolo, 2017)

1.2.La Voz Humana

La voz es un instrumento que permite la comunicación entre personas, los expertos la definen como: un sonido complejo formado por una frecuencia fundamental (determinada por la frecuencia de vibraciones de los ligamentos vocales) y un gran número de armónicos o sobre tonos.

La voz humana presenta variaciones en tres etapas de la vida, en la niñez, en la adolescencia y en la vejez. (Farfán, F, 2013)

1.2.1. Aparato Fonador

Para que la voz sea emitida se requiere la acción coordinada de varios órganos como se muestra en la Figura 1.

Hay tres elementos indispensables para la producción del sonido:

Cuerdas Vocales: Se encuentran ubicadas en la laringe.

Los Pulmones: El aire que proviene de ellos es el medio de propagación.

Caja de Resonancia: Está formada por la cavidad torácica, la faringe, cavidades oral y nasal, al igual que de elementos articulatorios (labios, dientes, alveolo, paladar, velo del paladar y la lengua).

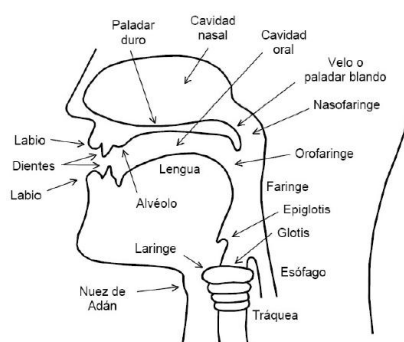


Figura 1. Aparato Fonador

Fuente: (Miyara, F)

1.2.2. Articulación

Por lo tanto, el paso de aire desde los pulmones hacia la laringe provoca un choque con las dos cuerdas vocales inferiores produciendo cambios de longitud y tensión, cuando estas cuerdas están separadas la glotis permite el paso del aire para que se produzca la respiración y no el sonido, mientras que cuando están las cuerdas

juntas la glotis de cierra, esta variación se requiere para la generación de sonidos.
(Miyara, F)

1.2.3. Cualidades de la voz

La voz presenta las cualidades que se muestran a continuación:

Intensidad: Es el volumen al cual se emite un sonido, depende de la cantidad de aire que es expulsado al momento de hablar.

Duración: Determina si se trata de un sonido largo o corto.

Timbre: Es la característica distintiva entre una voz y otra, aquí se amplifican o atenúan los armónicos de la voz.

Tono: Se trata de la frecuencia fundamental a la que vibran las cuerdas vocales, es la intensidad con la que se emite la voz.

1.2.4. Retardo Simple de Lenguaje

Es un desfase cronológico que produce dificultades más o menos graves del lenguaje. “Se considera que existe un retardo simple del lenguaje (RSL) cuando hay una buena comprensión y la evolución del lenguaje es similar a la mayoría de los niños normales, aunque con una cronología moderadamente retrasada”.
(Tonato, 2015)

Los niños tienen la capacidad de pronunciar sonidos aislados y grupos de sonidos, sin embargo, al momento de realizar frases completas o utilizar plurales presenta dificultades.

1.2.5. Causas que provocan el Retraso Simple de Lenguaje

Las causas pueden ser provocadas por motivos de orígenes como:

- Hereditarios
- Socioculturales
- Factores afectivos (sobrepotección, indiferencia, rechazo)
- Conflictos en la familia.

1.3. Reconocimiento de Voz

Se trata de procedimientos que permiten establecer la comunicación entre el ser humano y un computador.

Considerando un conjunto de información que procede de diferentes fuentes para ejecutar acciones determinadas.

1.3.1 Etapas

En la Figura 2 se muestra cual es el procedimiento que se ejecuta para la elaboración de sistemas de reconocimiento de voz.

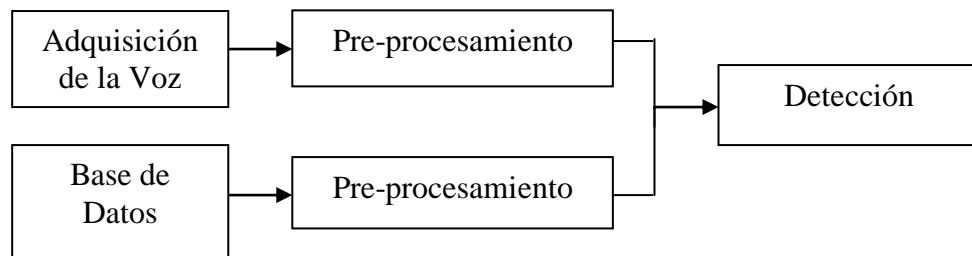


Figura 2. Etapas de Reconocimiento de Voz
Fuente: (Autores)

1.4. Componentes

1.4.1 Micrófono

El micrófono es un transductor cuya función consiste en convertir la energía de las variaciones de ondas sonoras en ondas eléctricas, es decir que convierte las señales acústicas en señales eléctricas.

1.4.1.1 Características de los Micrófonos

Sensibilidad: Relación entre la tensión que sale del micrófono y el nivel de intensidad acústica que ingresa al mismo.

Es una de las características más importantes que se considera para seleccionar el micrófono más adecuado, a mayor sensibilidad no se requieren equipos preamplificadores y por ende se disminuye el ingreso de ruido.

Relación Señal/Ruido (S/N): Se trata de la diferencia entre el nivel SPL y el ruido de fondo del micrófono. Al tener mayor relación señal a ruido mejor es el micrófono.

Directividad: Capacidad que poseen los micrófonos de captar el sonido de unas direcciones y no de otras.

Impedancia Interna: Limita el paso de la corriente, los micrófonos habituales poseen una impedancia de 200 a 600 Ohmios, aunque existen otros de mayor impedancia de 3000 Ohmios.

Los micrófonos de baja impedancia permiten utilizar cables más grandes que los micrófonos de alta impedancia.

Ruido de Fondo: Se considera al ruido propio del micrófono, 20dBA SPL es un excelente nivel de ruido, 30 dBA es un valor bueno y 40 dBA un nivel malo.

1.4.2 Raspberry pi3

Es una placa de computadora, que posee partes como: un micro procesador de 1.2 GHz de 64 bits con cuatro núcleos ARMv8, integrado en un chip Broadcom BCM2835; posee una memoria RAM de 512MB, Bluetooth 4.1, como se indica en la Figura 3. Tiene los siguientes sistemas operativos disponibles: Raspbian, Arch Linux, RaspBMC, Pidora, Windows 10. (Raspberry Pi)

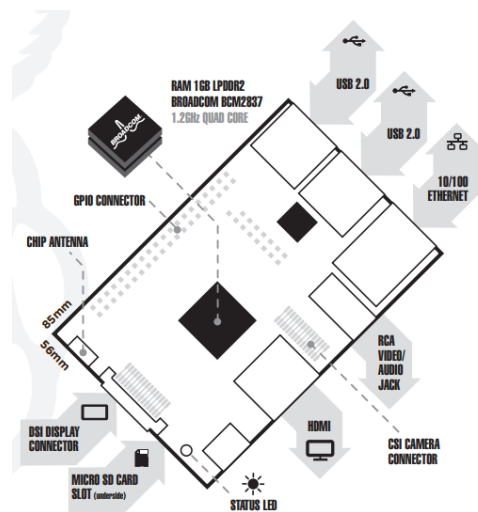


Figura 3. Partes de la tarjeta Raspberry Pi3
Fuente: raspberrypi.org

1.4.3 LCD

Para el proyecto se utiliza el kit de LCD SainSmart 7'' Display Touch Screen, posee HDMI/VGA/2AV y una placa de control de inversión, cuya conexión se muestra en la Figura 4. (SainSmart)



Figura 4. Kit LCD
Fuente: sainsmart

1.4.4 Software

Para cumplir con el objetivo que consiste en determinar cuál es el algoritmo más adecuado para el desarrollo del proyecto, se procede como primera etapa a realizar la comparación del funcionamiento de los métodos de: HTK, análisis de frecuencia y LPC en el software Matlab 2016; y posterior al resultado de este análisis se procede a generar el código para el sistema completo en el software Python 2.7.

1.4.4.1 Matlab 2016

Desarrollado por Mathworks, es un software licenciado para el análisis y desarrollo de diversos sistemas con la ayuda de herramientas que trabajan con un lenguaje propio de alto nivel. (MathWorks)

1.4.4.2 Python 2.7

Es un lenguaje de programación orientado a objetos, posee una amplia biblioteca y estructuras de alto nivel que se encuentran disponibles libremente. (Rossum, Drake, 2009).

1.4.4.3 Librerías de Python 2.7

Numpy: Paquete que ofrece herramientas sofisticadas para trabajar con matrices N de dimensiones, integración de algoritmos de C/C++, permite la transformación de Fourier.

Tkinter: Proporciona herramientas para la generación de interfaz a través del software Python.

Image: Permite cargar imágenes en la interfaz de Tkinter.

Wave: Es un módulo que permite trabajar con archivos con formato .wav

Math: Permite realizar operaciones matemáticas definidas por el estándar C.

Scipy: Permite trabajar con matrices Numpy, con retinas numéricas fáciles de usar. Dentro del paquete se encuentran las herramientas Lfilter y Hamming.

LFilter: Filtra los datos utilizando un filtro digital IIR o FIR.

Hamming: La ventana de Hamming es una conicidad formada por el uso de un coseno elevado con puntos extremos distintos de cero, optimizado para minimizar el lóbulo lateral más cercano.

Scikits: Está formado por un conjunto de módulos que ofrece Python para el procesamiento de voz y señales, dentro de este se encuentra la herramienta LPC.

Time: Proporciona varias funciones relacionadas con el tiempo.

Soundevice: Permite reproducir y grabar matrices numpy que contienen señales de audio.

Sounfile: Permite leer y escribir archivos de sonido.

Matplotlib: Se encarga de la generación de gráficos en 2D y 3D (Python.org, 2017)

1.4.5 Métodos y Etapas para el Reconocimiento de Voz

1.4.5.1 Ingreso de la Voz

Para el desarrollo del sistema de reconocimiento de voz generalmente se utiliza un micrófono para adquirir las señales de la voz generadas por los usuarios.

1.4.5.2 Pre-procesamiento

Se utiliza para eliminar señales espurias que ingresan junto con las señales de voz, es decir tiene como función eliminar el ruido y periodos de silencio que se presentan al momento de las grabaciones para obtener una información más adecuada.

1.4.5.3 Segmentación

Divide la señal en varios segmentos de análisis en los que se asume que la señal es estacionaria. Durante este proceso los segmentos se guardan en una matriz.

1.4.5.4 Ventaneo

A los segmentos de la voz se aplica una ventana de Hamming para posterior ser

aplicada un filtro.

1.4.5.5 Filtrado de la señal

Se utiliza un filtro pasa altas de primer orden para acentuar las frecuencias altas de la señal de voz, para evitar pérdidas de información.

El que se representa con la siguiente ecuación:

$$H(z) = 1 - a * z^{-1} \rightarrow \check{S}[n] = S[n] - a * S[n - 1] \quad (1)$$

1.4.5.6 LPC (Linear Prediction Coefficients)

Es un método muy utilizado para reconocimiento de voz debido a que se aproxima al sistema vocal de los seres humanos. Tiene como función representar a la señal de voz como una excitación constituida por un tren de pulsos para sonidos vocalizados. (Teplitsky, Frigon, J 2000).

Es un algoritmo que conoce lo que ocurrirá en el presente, considerando la información del pasado de la señal. LPC matemáticamente es preciso y con la ayuda de software y hardware ya existentes es de fácil desarrollo en comparación a otros métodos digitales.

El algoritmo LPC es de los más utilizados, es la combinación lineal de las muestras pasadas, generando una combinación espectral basa en segmentos cortos de la señal. La función de transferencia del filtro se describe como (Alejandro, Ortega, González, y Martínez-romo, 2012)

$$H(z) = \frac{S(z)}{U(z)} = \frac{G}{1 - \sum_{k=1}^p a_k z^{-k}} = \frac{G}{A(z)} \quad (2)$$

Donde G, a_k , y p son el parámetro de ganancia, de coeficientes del filtro, la señal filtrada $s[n]$ se relaciona mediante la excitación $u[n]$, esto es:

$$s(n) = \sum a_k s[n - k] + Gu[n] \quad (3)$$

La estimación de coeficientes de predicción se obtiene al minimizar el error de predicción, dado por:

$$e[z] = s[z] \left(1 - \sum_{k=1}^p a_k z^{-k} \right) \quad (4)$$

El error de predicción en un corto tiempo se define como:

$$E = \sum e[m]^2 = \sum (s[m] - \sum_{k=1}^p a_k s[m-k])^2 \quad (5)$$

Parte fundamental para la caracterización de la señal de coeficientes de predicción, es establecer un número adecuado de coeficientes de acuerdo a la frecuencia de muestreo, esto es:

$$p = 4 + \frac{fs}{1000} \quad (6)$$

Con lo que se obtiene un resultado óptimo de número de coeficientes LPC que puede representar el menor error cuadrático medio posible.

1.4.5.7 Distancia Euclidiana

Se encarga de calcular la menor distancia que existe entre el patrón generado en la base de datos y la palabra ingresada por el usuario.

Teniendo como resultado un valor que representa la distancia entre dos matrices que poseen iguales dimensiones.

La ecuación matemática que representa a la distancia euclidiana de dos o más dimensiones se muestra a continuación:

$$D = \left[\sum [X_{ij} - X_{ik}]^2 \right]^{1/2} \quad (7)$$

Donde X_{ij} y X_{ik} son valores de los atributos de la especie i en las estaciones j y k que se comparan. (Montaigne, 1979)

2. METODOLOGÍA

2.1. Tipo de estudio

2.1.1 Descriptivo: Se ejecuta este tipo de estudio para describir características y generalidades del problema a investigar, mediante la exploración y descripción de situaciones y lugares de la vida real.

- 2.1.2 Explicativo:** Estudia las causas y efectos que se generan del problema de estudio.
- 2.1.3 Cuantitativo:** Proceso formal, sistemático en el que se usan datos numéricos para obtener información sobre el objeto de estudio.
- 2.1.4 Experimental:** Evalúa los efectos de la intervención de un sistema, ya sea preventiva, curativa o rehabilitadora.
- 2.1.5 Experimental Puro:** Permite la distribución de usuarios en un grupo de control y otro de experimentación.

2.2. Métodos y Técnicas

2.2.1. Métodos

2.2.1.1. Analítico/Deductivo

Se utiliza el método analítico-deductivo en el proyecto ya que se empieza con un análisis particular con el algoritmo más adecuado para reconocimiento de voz y componentes necesarios para el desarrollo del sistema que permite la interacción del niño con el dispositivo de evaluación para su desempeño y un funcionamiento más óptimo.

2.2.2. Técnicas

2.2.2.1. Observación

Esta técnica consiste en la recolección de información que sea de apoyo para el desarrollo del proyecto, dando las pautas necesarias para el diseño y construcción de un sistema para detección de retardo simple en los niños de 4 años mediante el reconocimiento de voz.

2.3. Población y muestra

La población corresponde a un conjunto de elementos de los que se quiere conocer o investigar algunas características. Se trabajo con un total de 20 estudiantes de la Unidad Educativa Miguel Ángel León de inicial 2 del paralelo “D”.

El estudio se centró en dos grupos:

- **De Control:** Se trabajó con 11 niños para la generación de la base de datos
- **Experimental:** 4 niños con dos intentos cada uno para la comprobación del funcionamiento del sistema total.

2.4. Hipótesis

El diseño y construcción de un sistema de detección de retardo simple de lenguaje por medio de voz para niños de 4 años permitirá evaluar la pronunciación del niño.

2.5. Operacionalización de variables

Los indicadores e instrumentos utilizados para el desarrollo del proyecto se muestran en la Tabla 1.

Variables	Dimensione	Indicadores	Instrumentos
El diseño y construcción de un sistema de detección de retardo simple de lenguaje por medio de voz para niños de 4 años	Algoritmo para reconocimiento de voz	Ingreso de sonidos Interfaz hombre-	Micrófono Raspberry Pi3 LCD touch screen 7
	Software	Herramienta de software	Python 2.7
Evaluar la pronunciación del niño	Detección de retardo simple en niños de 4 años	Test de evaluación	Base de Datos
		Presencia	Ingreso por micrófono

Tabla 1. Operacionalización de variables.

Fuente: Autores.

2.6. Procedimiento

2.6.1 Elaboración del Sistema

Para el desarrollo de este sistema de detección del retardo simple de lenguaje, se requirió seguir una serie de procesos, que en conjunto cumpliría con el objetivo propuesto.

A continuación en la Figura 5, se presentan los siguientes pasos a seguir en la investigación propuesta:

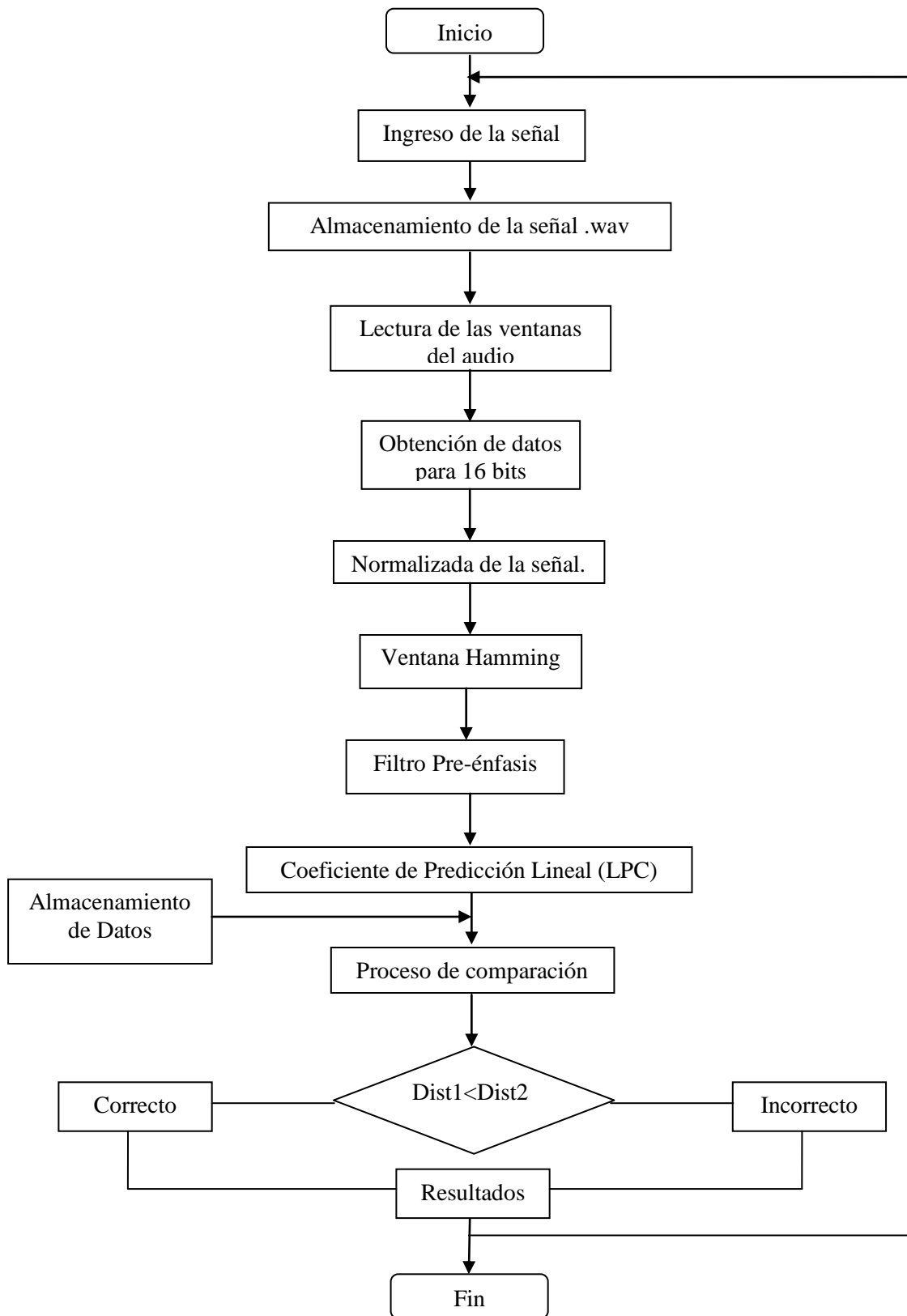


Figura 5. Proceso para el desarrollo del proyecto
Fuente: (Autores)

2.6.2 Ejecución del Sistema

Para comenzar se debe seleccionar el algoritmo más adecuado para el desarrollo del proyecto, para lo cual se procede a desarrollar en el software Matlab 2016 la comparación del funcionamiento de los algoritmos que se utilizan para el reconocimiento de voz como son: análisis de frecuencia, análisis HTK y LPC.

Una vez determinado el funcionamiento de los algoritmos antes mencionados se constató que el análisis por frecuencia entre dos señales de voz otorga datos erróneos debido a que no detecta la mala pronunciación de la palabra.

Debido a lo antes mencionado se seleccionó al algoritmo LPC como el más adecuado ya que por medio de sus coeficientes logra detectar la variación en la pronunciación de las palabras y que provee una separación entre la excitación y el modelo del tracto vocal, que permite una representación simple, además de que LPC es matemáticamente preciso y no necesita mucho tiempo de procesamiento.

Al obtener el algoritmo más adecuado se procede a la transcripción del código del software Matlab 2016 al software Python 2.7. Se efectuó la creación de la base de datos con 10 palabras (chocolate, elefante, chupete, vaca, cepillo, conejo, delfín, plato, plancha y tortuga), para ello se realizó un programa de entrenamiento que permite almacenar audios de 11 niños que pronuncien correctamente cada una de las palabras, para la generación del programa se utilizan las librerías de Python que se detallan a continuación:

Wave: Lectura de archivos wav.

Sounddevice: Sirve para empezar a guardar el audio que ingresa por el micrófono.

Soundfile: Permite guardar el archivo wav lo que se realizó la grabación.

Después de la obtención de los audios para la base de datos se realiza un segundo programa, el cual consta de diferentes etapas. En primer lugar, se diseñó la interfaz que brinda un ambiente amigable para el usuario, el cual consta de imágenes y dos botones, uno para iniciar el proceso y otro para mostrar los resultados.

Para la configuración en cada uno de los botones se hace el uso de funciones para realizar el llamado de acuerdo a lo que requiera el programa.

Para comenzar se debe pulsar el botón de inicio se abre una nueva ventana en la que constan 10 botones, cada uno con la palabra respectiva que debe el niño grabar. El niño escuchará un audio con las instrucciones que se requiere para utilizar el programa, el audio indicará que debe pulsar el primer botón para comenzar, es

decir el correspondiente a la palabra chocolate.

Luego de ello se abre una ventana donde se observa una imagen representativa y se escuchará otro audio indicando que debe pulsar en el botón de grabación para que proceda a decir la palabra.

Al regresar a la ventana de evaluación se deberá realizar el mismo procedimiento para los 9 botones restantes, es decir para las palabras: elefante, chupete, vaca, cepillo, conejo, delfín, plato, plancha y tortuga.

Durante la ejecución del programa se recolecta la información ingresada por el usuario, la cual se va a llamar más adelante para ser comparada con la base de datos.

En segundo lugar, para realizar el procesamiento de los audios tanto de ingreso del usuario como de la base de datos se realiza el procedimiento que a continuación se muestra:

- a) Se hace la lectura del archivo wav, tanto del audio ingresado por el usuario como los audios generados por la base de datos.
- b) Se calcula el valor de la longitud de audio.
- c) El archivo de audio se inicializa a partir de datos binarios o de texto en una cadena que se trabaja con 16 bits.
- d) Teniendo el audio se realiza el proceso de normalización, para que el audio tenga una ganancia que lleve la amplitud del pico promedio a un mismo nivel.
- e) Después al audio normalizado se le efectúa la segmentación.
- f) Se realiza un ventaneo con el método Hamming para minimizar las discontinuidades de la señal al principio y al final de cada segmento.
- g) Luego se filtra la señal con un filtro FIR de primer orden de manera de nivelar su espectro.
- h) Y por último al audio se aplica el algoritmo de LPC, que mostrará los coeficientes que serán comparados con el patrón de cada palabra y determinará la similitud o no de acuerdo a la pronunciación de cada niño.

Para obtener el patrón de la palabra que se mencionó anteriormente, se emplea un entrenamiento por clustering, debido a que se tiene en la base de datos almacena audios de varios usuarios, lo que no restringe el uso del prototipo a niños que se encuentren en la base de datos, sino a cualquier niño, lo que hace que el sistema sea más confiable.

El patrón de cada palabra estar formado por 11 grabaciones, de cada una se calcula los coeficientes LPC y se calcula el promedio para obtener una señal patrón como referencia para el reconocimiento.

Después de obtener el patrón se realiza la comparación del audio ingresado por el usuario con el patrón ya establecido, lo cual se puede ejecutar mediante la utilización de la distancia euclidiana o con la distancia mahalanobis.

Para esta ocasión se eligió la distancia euclidiana, debido a que este método realiza una diferencia de las matrices que son homogéneas, mientras que la distancia mahalanobis se puede utilizar cuando una de las matrices no sea homogénea.

Se obtiene la diferencia entre el patrón de cada palabra y la que fue ingresada por el usuario.

Finalmente se compara la mínima distancia, en el caso de que la palabra sea correcta la mínima distancia debe ser mayor de lo establecido, en caso contrario la pronunciación es incorrecta.

Los resultados una vez finalizado el proceso muestran mediante un diagrama de barras la cantidad de palabras bien pronunciadas, y la cantidad de palabras mal pronunciadas. Si el número de palabras mal pronunciadas es mayor o igual que 4 se muestra un mensaje de que el niño necesita tratamiento por retraso simple de lenguaje.

2.7 Comprobación de la Hipótesis

Se comprueba la hipótesis a través del método de Chi Cuadrado de Pearson, que trabaja con la fórmula que se muestra a continuación:

$$x^2 = \sum \frac{(fo - ft)^2}{ft} \quad (8)$$

Al utilizar el software IBM SPSS, se trabajó con un nivel de significancia de $\alpha=0.05$, en la Tabla 2 se muestra los resultados de la detección del sistema.

		Pronunciación		Total	
		Correcto	Incorrecto		
Palabras	Chocolate	Recuento	16	0	16
		Recuento esperado	14,3	1,7	16,0

Elefante	Recuento	14	2	16
	Recuento esperado	14,3	1,7	16,0
Vaca	Recuento	11	5	16
	Recuento esperado	14,3	1,7	16,0
Chupete	Recuento	15	1	16
	Recuento esperado	14,3	1,7	16,0
Cepillo	Recuento	15	1	16
	Recuento esperado	14,3	1,7	16,0
Conejo	Recuento	12	4	16
	Recuento esperado	14,3	1,7	16,0
Plato	Recuento	16	0	16
	Recuento esperado	14,3	1,7	16,0
Delfín	Recuento	14	2	16
	Recuento esperado	14,3	1,7	16,0
Plancha	Recuento	14	2	16
	Recuento esperado	14,3	1,7	16,0
Tortuga	Recuento	16	0	16
	Recuento esperado	14,3	1,7	16,0
Total	Recuento	143	17	160
	Recuento esperado	143,0	17,0	160,0

Tabla 2. Aciertos y Errores por cada palabra
Fuente: (Autores)

Con lo que se obtuvo un valor de 160 muestras que representan el 100% como se indica en la Tabla 3.

	Casos					
	Válido		Perdidos		Total	
	N	Porcentaje	N	Porcentaje	N	Porcentaje
Palabras * Pronunciación	160	95,8%	0	0%	160	100,0%

Tabla 3. Resumen de resultados de muestras.
Fuente: (Autores)

Hipótesis Nula

La elaboración del sistema no determinará la presencia de retardo de lenguaje en niños de 4 años

Hipótesis Alternativa

La elaboración del sistema determinará la presencia de retardo de lenguaje en niños de 4 años.

El valor calculado de χ^2 es de 17.178 con 9 grados de libertad como se muestra en la Tabla 4, y al observar en las tablas de Chi Cuadrado se considera que se tiene un χ^2 de valor de 16.9190.

Pruebas de chi-cuadrado			
	Valor	gl	Sig. asintótica (2 caras)
Chi-cuadrado de Pearson	17,178 ^a	9	,046
Razón de verosimilitud	19,351	9	,022
Asociación lineal por lineal	,446	1	,504
N de casos válidos	160		

Tabla 4. Resultado de Chi Cuadrado.

Fuente: (Autores)

Por lo que se determina que al tener un valor de 17.178 mayor que 16.9190 se rechaza la hipótesis nula como se representa en la Figura 6.

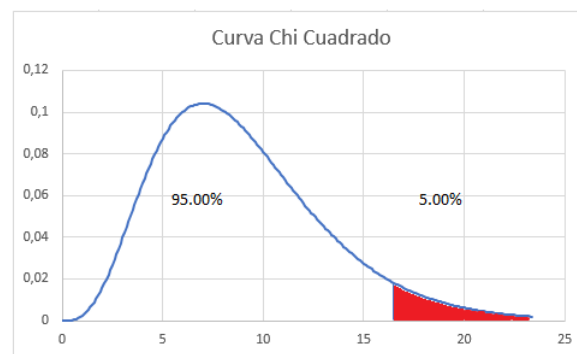


Figura 6. Curva de Chi Cuadrado

Fuente: (Autores)

3. RESULTADOS Y DISCUSIÓN

En esta parte se indica el resultado final del diseño y funcionalidad de cada una de las interfaces gráficas y diseñadas con el software Python 2.7 para la detección de la palabra que se realizó por medio de reconocimiento de voz.

Para el programa se consideró varios puntos de cálculo como eliminación de silencios, filtrado FIR de la señal y el algoritmo LPC, cuyo valor depende de la pronunciación del niño.

Para la realización del proyecto se considera dos interfaces gráficas creando archivos con extensión Py, cuyas funcionalidades serán explicadas a continuación:

3.1 Interfaz para detección de palabra para niños de 4 años

El programa de detección de la palabra consta de dos interfaces gráficas denominadas:

1. Interfaz gráfica de base de datos (entrenamiento)
2. Interfaz para la detección de la palabra de cada niño con reconocimiento de voz.

3.1.1 Interfaz gráfica para base de datos

Para realizar una base de datos se requirió a niños de 4 años de edad de la Unidad Educativa Miguel Ángel León del paralelo “D” de inicial 2, el curso consta de 20 estudiantes y que se requirió 11 niños para realizar el entrenamiento de las palabras para llevar a cabo el almacenamiento de cada audio de los niños.

La interfaz consta de 10 botones para la grabación de los audios de cada palabra como se observa en la Figura 7.

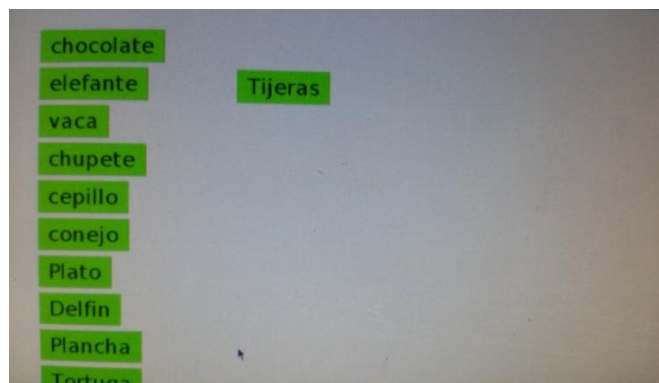


Figura 7. Interfaz para generar la Base de Datos
Fuente: (Autores)

3.1.2 Interfaz para la detección de la palabra mediante la voz

3.1.2.1 Evaluación

En la Figura 8 se muestra la pantalla de inicio de la interfaz y en la Figura 9 muestra las 10 palabras a ser grabadas.



Figura 8. Ventana Principal de la Interfaz
Fuente: (Autores)



Figura 9. Ventana de Evaluación
Fuente: (Autores)

3.1.2.2 Resultados

A continuación, se muestra los resultados obtenidos luego de las pruebas realizadas al sistema, mediante tablas se muestra si las palabras dichas por los niños fueron pronunciadas de manera correcta o incorrecta, y mediante un botón se observa en un gráfico de barras y dependiendo del número de errores, se calificará al niño con leve, moderado o severo retardo simple del lenguaje, en la Figura 10 se muestra la interfaz de los resultados.

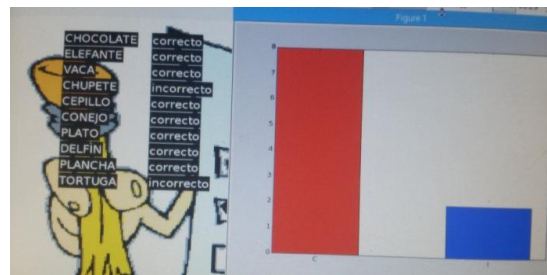


Figura 10. Resultados de Evaluación
Fuente: (Autores)

3.2 Resultados de la implementación

Para construir un sistema que sea amigable y llamativo para los niños de 4 años, se realiza un diseño para la colocación de la tarjeta Raspberry Pi3 y de la pantalla LCD en acrílico como se muestra en la Figura 11.



Figura 11. Implementación del sistema
Fuente: (Autores)

3.3 Pruebas y resultados

Para la generación de la base de datos fue necesario almacenar audios en tiempo real de 11 niños que pronunciaran correctamente cada una de las palabras, luego de tratamiento respectivo de la señal, se obtuvo los coeficientes LPC de cada uno de ellos, en la Tabla 5 que se presenta a continuación se muestra los valores obtenidos de la primera palabra, es decir chocolate.

Coeficiente de predicción lineal (LPC)

Palabra	Niño 1	Niño 2	Niño 3	Niño 4	Niño 5	Niño 6	Niño 7	Niño 8	Niño 9
	1	1	1	1	1	1	1	1	1
Chocolate	-0.1956	-0.2379	-0.2547	-0.0544	-0.2251	-0.0662	-0.4579	-0.6359	-0.3447
	0.0616	0.0268	0.0862	-0.0187	-0.0042	0.1199	0.1301	0.0427	-0.0018
	-0.2609	-0.2007	-0.1661	-0.2780	-0.2962	-0.1897	-0.1388	-0.1462	-0.2565
	-0.0687	-0.0362	-0.1340	0.0457	-0.0329	-0.1808	-0.0478	0.1443	-0.1814
	0.2915	0.3059	0.0762	0.3598	0.3371	0.1562	0.1925	0.3496	0.3676
	0.0585	0.1154	0.1388	0.0756	0.2155	0.1840	0.0244	-0.0656	-0.0384
	-0.1280	-0.0857	0.0643	-0.0771	0.0824	0.0279	0.0045	-0.1601	-0.0601
	-0.1036	-0.0569	-0.0857	0.0281	0.0515	0.0071	0.1302	0.0329	0.0366
	-0.2108	-0.1233	0.0575	-0.1697	-0.0671	-0.0543	-0.1284	-0.0771	-0.1410
	-0.0544	0.424	0.0633	0.0178	-0.0931	-0.0081	-0.0505	0.1214	0.0778
	0.0218	0.1175	-0.0561	0.0620	0.0170	-0.0322	0.0018	0.0840	0.0136
	0.1368	0.2288	0.1333	0.1669	0.1416	0.1535	0.1845	0.1283	0.0827
	0.0153	0.0050	0.1148	0.0162	0.1111	0.0918	0.0122	-0.0737	0.0156
	-0.0057	-0.0820	-0.0161	-0.0228	0.0558	0.0066	-0.0325	-0.0532	-0.0263
	0.0022	-0.0065	-0.0651	-0.0696	0.0078	-0.0479	-0.0305	0.0066	-0.0645
	-0.070	-0.0525	-0.1313	-0.1114	-0.1780	-0.1171	-0.1376	-0.0692	-0.0513
	-0.0598	-0.1086	-0.1204	-0.0467	-0.2143	-0.1685	-0.1360	-0.0531	-0.0392
	0.1490	0.1127	0.0480	0.1357	0.0256	-0.0252	0.1570	0.1869	0.0930
	0.0669	0.0550	-0.0036	0.0230	0.0765	-0.0765	0.0528	-0.0311	0.0102
	-0.0252	-0.0432	0.0585	-0.0490	-0.0077	-0.1076	-0.0144	-0.1168	-0.0899
-0.0243	-0.0363	0.0583	-0.0489	-0.0050	-0.0285	0.0479	-0.0103	-0.1328	
-0.1495	-0.1754	-0.0814	-0.1051	-0.1468	-0.1585	-0.1627	-0.1056	-0.1387	
0.1343	0.0325	0.0823	0.1388	0.0275	0.0389	0.0815	0.1961	0.1113	

Tabla 5. Coeficientes de la palabra chocolate de la base de datos.

Fuente: (Autores)

Posterior de la obtención de los coeficientes LPC de todas las palabras se procede a realizar el cálculo de promedio para generar los patrones de cada una de las diez palabras, dichos patrones se muestran a continuación en las Tablas 6 y 7.

Coeficientes LPC					
Coeficientes	Patrón Chocolate	Patrón Elefante	Patrón Vaca	Patrón Plato	Patrón Chupete
1	1	1	1	1	1
2	-0.3398	-0.5697	-0.1328	-0.2032	-0.3183
3	0.0687	0.1594	0.0205	0.0396	0.1101
4	-0.2200	-0.2994	-0.3267	-0.3380	-0.2466
5	0.0386	0.1647	0.0416	0.0633	0.1349
6	0.2741	0.3047	0.2629	0.2995	0.2698
7	0.0595	-0.0002	0.0340	0.0647	0.0241
8	-0.0196	0.0348	0.0134	0.0338	0.0810
9	-0.0070	-0.0232	0.0182	0.0025	0.0677
10	-0.0953	-0.1601	-0.1635	-0.1565	-0.1376
11	-0.0229	0.0619	0.0556	0.0168	0.0100
12	0.0345	0.0737	0.0820	0.0324	0.0015
13	0.1434	0.1173	0.1223	0.1248	0.0563
14	0.0368	0.0034	0.0825	0.0987	0.0501
15	-0.0773	-0.0122	0.0225	0.0342	0.0091
16	-0.0262	-0.0713	-0.0781	-0.0828	-0.1002
17	-0.1115	-0.1063	-0.0977	-0.1270	-0.1499
18	-0.0957	-0.0164	-0.0311	-0.0751	-0.0924
19	0.1905	0.1181	0.0854	0.0504	0.0213
20	0.0110	0.0030	0.0027	-0.0163	-0.0243
21	-0.0426	-0.0295	-0.0358	-0.0469	-0.0320
22	-0.0431	-0.0906	-0.0516	-0.0640	-0.0688
23	-0.1475	-0.1330	-0.1199	-0.1065	-0.0255
24	0.0666	0.1285	0.0161	0.0622	0.0325

Tabla 6. Patrón de las 5 primeras palabras almacenadas
Fuente: (Autores)

Coeficientes LPC					
Coeficientes	Patrón Cepillo	Patrón Conejo	Patrón Delfín	Patrón Plancha	Patrón Tortuga
1	1	1	1	1	1
2	-0.5423	-0.3252	-0.4725	-0.5665	-0.2880
3	0.4062	0.0584	0.2301	0.2844	0.0167
4	-0.2117	-0.3153	-0.2432	-0.1763	-0.3641
5	0.1613	0.0978	0.2440	0.2048	0.0446
6	0.2921	0.2803	0.3445	0.1759	0.2684
7	0.0321	0.0183	0.0471	-0.0846	0.0006
8	0.1224	0.0356	0.1187	0.1107	-0.0011
9	0.0459	-0.0007	0.0036	0.0462	-0.0267
10	-0.1761	-0.1691	-0.1831	-0.1439	-0.1742
11	0.0140	0.0201	0.0809	0.0651	0.0352
12	-0.0094	0.0168	0.0330	0.0816	0.0561
13	0.0290	0.0757	0.0591	0.0762	0.1131
14	0.0291	0.0451	0.0119	0.0763	0.0891
15	0.0167	-0.0189	-0.0462	-0.0171	0.0196

16	-0.1065	-0.1025	-0.1470	-0.0351	-0.0849
17	-0.1832	-0.1190	-0.1476	-0.0471	-0.1100
18	-0.1127	-0.0451	-0.0842	-0.0007	-0.0474
19	0.0288	0.0640	0.0239	0.0511	0.0821
20	0.0024	-0.0278	-0.0163	-0.0122	0.0421
21	-0.0188	-0.0368	-0.0252	-0.0148	0.0017
22	-0.1063	-0.0390	-0.1019	-0.0577	-0.0446
23	-0.1082	-0.0748	-0.1145	-0.1303	-0.0925
24	-0.0243	0.0835	0.0075	0.0565	0.0379

Tabla 7. Patrón de las 5 últimas palabras.
Fuente: (Autores)

Para realizar la comparación se procede a determinar la similitud articulatoria y fonológica al momento de pronunciar las palabras, lo que quiere decir que se considera palabras que al ser mal pronunciadas por los niños el movimiento del paladar, dientes, lengua, labios y la glotis serán similares a otras palabras de la base de datos.

Por tal motivo se procede a realizar la comparación entre las palabras que se muestran en la Tabla 8.

Comparación	
Palabra Pronunciada	Palabra Comparada
Chocolate	Elefante
Elefante	Chocolate
Vaca	Plato
Plato	Vaca
Chupete	Conejo
Conejo	Chupete
Cepillo	Conejo
Delfín	Plancha
Plancha	Delfín
Tortuga	Chupete

Tabla 8. Lista de palabras a comparar.
Fuente: (Autores)

Para verificar el funcionamiento de sistema se realizaron varias pruebas con niños que no se encuentran registrados en la base de datos, en la Tabla 9 muestra un caso entre dos niñas, una de ellas tiene una buena pronunciación mientras que la otra no.

Palabra	Coeficiente de predicción lineal (LPC)	
	Niña con buena pronunciación	Niña con mala pronunciación
Chocolate	1	1
	0.7741	-1.5543
	1.0693	1.5348
	1.0877	-0.8548
	0.8768	-0.1422
	0.7075	0.7333
	0.6097	-0.2187
	0.4563	-0.1537
	0.3441	0.3960
	0.2623	-0.1981
	0.3037	-0.0983
	0.1432	0.1477
	0.1213	0.3765
	0.1096	-0.3600
	0.0732	0.1402
	0.0340	0.2086
	0.0056	-0.2764
	-0.0655	0.0109
	-0.0248	0.3996
	-0.0662	-0.3179
-0.0806	-0.0169	
0.0025	0.2714	
-0.0717	-0.3412	
-0.0082	0.1882	

Tabla 9. Comparación de coeficientes de predicción lineal (LPC)
Fuente: (Autores)

Para poder realizar un análisis a partir del modelo de los coeficientes en el reconocimiento de palabras de un locutor aislado, se obtuvieron algunas gráficas que se muestran a continuación. En la Figura 12 se muestra dos señales de audio, la de color azul representa la ingresada por el usuario y la de color verde la señal obtenida de la base de datos, cada una de ellas con una duración de 3 segundos y con una frecuencia de muestreo de 44100Hz.

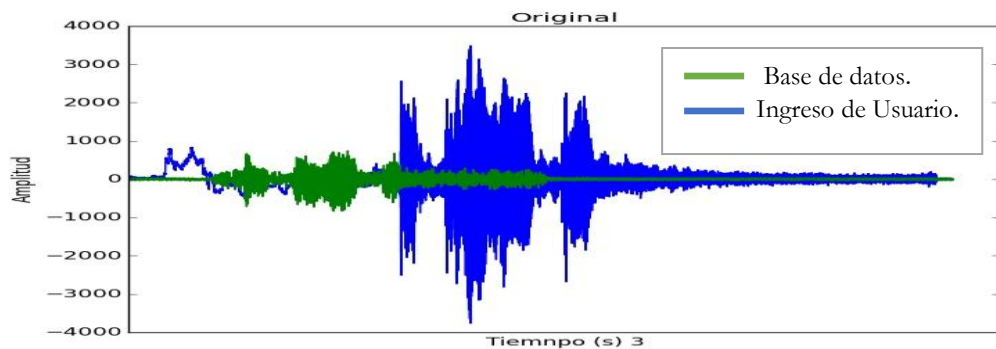


Figura 12. Muestras de las señales de la palabra chocolate.
Fuente: (Autores)

Se procede a realizar el producto entre la señal original y la ventana de Hamming, considerando que el número de muestras depende de la longitud de la señal, la de color azul es del usuario y la verde de la base de datos como se muestra en la Figura 13.

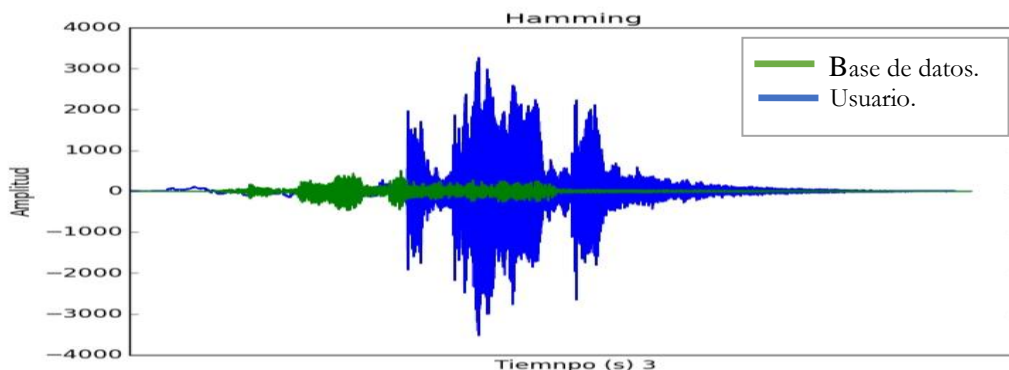


Figura 13. Producto de señales con ventana de Hamming
Fuente: (Autores)

Las gráficas de las señales normalizadas muestran que las señales con una ganancia que lleva la amplitud del pico promedio a un mismo nivel, señal azul del usuario y verde de la base de datos como se observa en la Figura 14.

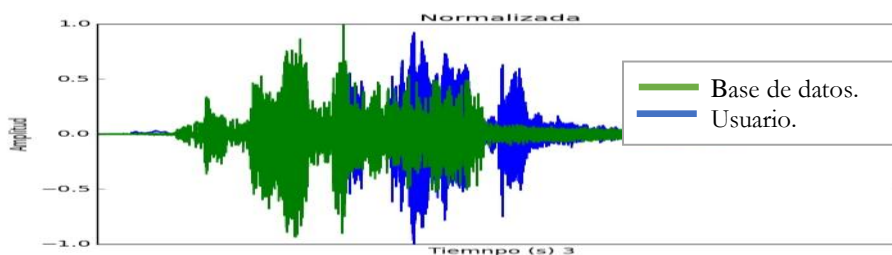


Figura 14. Normalización de señales
Fuente: (Autores)

Para obtener una señal más pura se aplica un filtrado FIR de primer orden que permite el paso de las frecuencias altas como se muestra en la Figura 15.

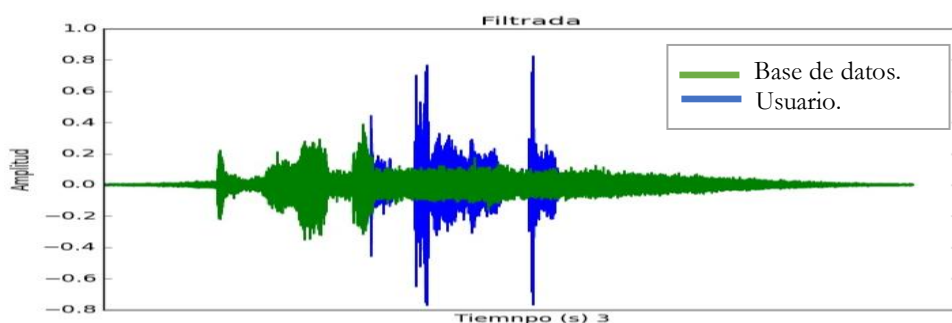


Figura 15. Filtrado de las muestras de señales de voz
Fuente: (Autores)

Se muestran los resultados de los coeficientes LPC mediante gráficas que se observan en la Figura 16, considerando que existe variación entre los coeficientes de la base de datos y de la señal ingresada por el usuario, por lo tanto, se determina que la pronunciación es incorrecta, la azul es del usuario y la roja es de la base de datos.

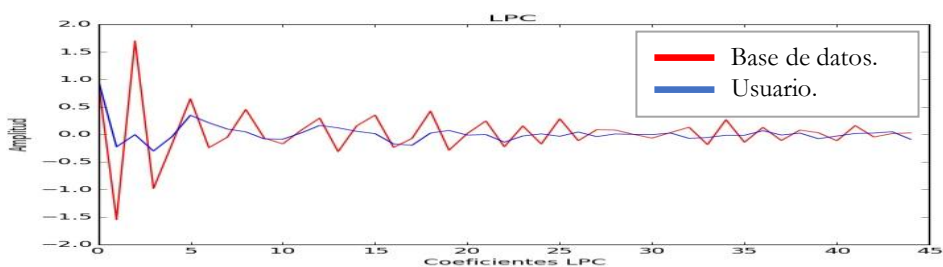


Figura 16. Comparación de coeficientes LPC

Fuente: (Autores)

3.3.1 Pruebas Realizadas

Para comprobar el funcionamiento del sistema se procede a realizar las pruebas con niños de 4 años, dando los siguientes resultados.

Usuario 1

Nombre: Romina Vinuesa (Figura 17)



Figura 17. Foto de Usuario 1

Fuente: (Autores)

Palabra	Distancia 1	Distancia 2	Pronunciación	Detección
Chocolate	1.7182	1.4495	X	✓
Elefante	1.1726	0.4593	X	X
Vaca	1.1579	1.0834	✓	X
Chupete	2.2375	1.8639	X	✓
Cepillo	2.1462	1.4360	X	✓
Conejo	1.5035	1.2997	X	✓
Plato	0.5928	0.5681	X	✓
Delfín	1.1830	0.9497	X	✓
Plancha	1.0242	0.8083	X	✓
Tortuga	0.9493	0.8065	X	✓
TOTAL			1	8
ACIERTOS			10%	80%

Tabla 10. Resultados del Usuario 1

Fuente: (Autores)

Nota: Se puede observar mediante el número de palabras pronunciadas erróneamente que la niña Romina presenta Retardo Simple de Lenguaje Severo.

Usuario 2

Nombre: Jhon Oñate (Figura 18)



Figura 18. Foto de Usuario 2

Fuente: (Autores)

Palabra	Distancia 1	Distancia 2	Pronunciación	Sistema
Chocolate	0.6632	1.0318	✓	✓
Elefante	0.0718	0.3704	✓	✓
Vaca	0.6290	0.5604	X	✓
Chupete	0.4437	0.4483	✓	✓
Cepillo	0.5741	0.7135	✓	✓
Conejo	0.9661	1.0331	✓	✓
Plato	0.4969	0.4622	X	✓
Delfín	0.5548	0.6079	✓	✓
Plancha	1.3476	1.2827	X	✓
Tortuga	1.0944	1.0842	X	✓
TOTAL			6	10
ACIERTOS			60%	100%

Tabla 11. Resultados de Usuario 2

Fuente: (Autores)

Nota: Se puede observar mediante el número de palabras pronunciadas erróneamente que la niña Jhon Oñate presenta Retardo Simple de Lenguaje Moderado.

Usuario 3

Nombre: Marcela Paredes (Figura 19)



Figura 19. Foto de Usuario 3

Fuente: (Autores)

Palabra	Distancia 1	Distancia 2	Pronunciación	Detección
Chocolate	0.4100	0.7917	✓	✓
Elefante	1.1300	1.4112	✓	✓

Vaca	1.4120	1.5785	✓	✓
Chupete	0.7865	0.8011	✓	✓
Cepillo	1.1597	1.4655	✓	✓
Conejo	1.2212	1.2421	✓	✓
Plato	1.2961	1.3987	✓	✓
Delfin	1.2053	1.5552	✓	✓
Plancha	1.1958	1.2745	✓	✓
Tortuga	1.3447	1.4287	✓	✓
TOTAL			10	10
ACIERTOS			100%	100%

Tabla 12. Resultados de Usuario 3

Fuente: (Autores)

Nota: Se puede observar que la niña Marcela Paredes no presenta Retardo Simple de Lenguaje.

Usuario 4

Nombre: Ismael Benítez (Figura 20)



Figura 20. Foto de Usuario 4

Fuente: (Autores)

Palabra	Distancia 1	Distancia 2	Pronunciación	Detección
Chocolate	0.8751	1.3066	✓	✓
Elefante	0.2225	0.7619	✓	✓
Vaca	0.6753	0.7231	✓	✓
Chupete	1.5148	1.4675	✓	X
Cepillo	1.3579	1.7156	✓	✓
Conejo	0.8841	0.8987	✓	✓
Plato	1.4805	1.5611	✓	✓
Delfin	0.6052	0.7058	X	X
Plancha	0.9865	0.9875	✓	✓
Tortuga	0.8078	0.7961	X	✓
TOTAL			8	8
ACIERTOS			80%	80%

Tabla 13. Resultados de Usuario 4

Fuente: (Autores)

Nota: Se puede observar mediante el número de palabras pronunciadas erróneamente que el niño Ismael Benítez presenta Retardo Simple de Lenguaje Leve.

El porcentaje de reconocimiento por palabra se presenta en la Tabla 14, se puede indicar que se alcanzó un 89.375% de detección de la palabra por el sistema a nivel de la pronunciación.

Palabra	Reconocimiento por intentos	Porcentaje
Chocolate	16	100%
Elefante	14	87.5%
Vaca	11	68.75%
Chupete	15	93.75%
Cepillo	15	93.75%
Conejo	12	75%
Plato	16	100%
Delfín	14	87.5%
Plancha	14	87.5%
Tortuga	16	100%
TOTAL	143	
	Efectividad	89.375%

Tabla 14. Porcentaje de efectividad del sistema de cada palabra de detección
Fuente: (Autores)

Para la evaluación y verificación del sistema se ejecuta las pruebas con dos intentos por cada niño, en la Tabla 15 Y Tabla 16 se puede observar la funcionalidad del sistema determinando en donde se obtuve una falsa respuesta en el método, obteniendo los siguientes resultados.

Palabra	Usuario 1		Usuario 2		Usuario3		Usuario4	
	Intento1	Intento2	Intento1	Intento2	Intento1	Intento2	Intento1	Int2
Chocolate	✓	✓	✓	✓	✓	✓	✓	✓
Elefante	X	✓	✓	✓	✓	✓	✓	✓
Vaca	X	X	✓	✓	✓	X	X	✓
Chupete	✓	✓	✓	X	✓	✓	✓	✓
Cepillo	✓	✓	✓	X	✓	✓	✓	✓
Conejo	✓	✓	✓	✓	✓	✓	✓	✓
Plato	✓	✓	✓	✓	✓	✓	✓	✓
Delfín	✓	✓	✓	✓	✓	✓	X	X
Plancha	✓	✓	✓	✓	✓	✓	✓	✓
Tortuga	✓	✓	✓	✓	✓	✓	✓	✓
TOTAL	8	9	10	8	10	9	8	9
ACIERTOS	80%	90%	100%	80%	100%	90%	80%	90%

Tabla 15. Resultados de los usuarios que utilizaron el sistema.
Fuente: (Autores)

Palabra	Usuario 5		Usuario 6		Usuario 7		Usuario 8	
	Intento1	Intento2	Intento1	Intento2	Intento1	Intento2	Intento1	Int2
Chocolate	✓	✓	✓	✓	✓	✓	✓	✓
Elefante	✓	✓	✓	✓	X	✓	✓	✓
Vaca	✓	✓	✓	✓	X	✓	✓	✓
Chupete	✓	✓	✓	✓	✓	✓	✓	✓
Cepillo	✓	✓	✓	✓	✓	✓	✓	✓
Conejo	✓	X	X	X	X	✓	✓	✓
Plato	✓	✓	✓	✓	✓	✓	✓	✓
Delfín	✓	✓	✓	✓	✓	✓	✓	✓
Plancha	✓	✓	X	✓	X	✓	✓	✓
Tortuga	✓	✓	✓	✓	✓	✓	✓	✓
TOTAL	10	9	8	9	6	10	10	10
ACIERTOS	100%	90%	80%	90%	60%	100%	100%	100%

Tabla 16. Resultados de los usuarios que utilizaron el sistema.

Fuente: (Autores)

3.3.2 Discusión

Mediante la implementación del sistema en el software Python 2.7 de Raspberry Pi3, y con la ayuda de las librerías que ofrece, se pudo desarrollar la detección de la pronunciación correcta o incorrecta en niños de 4 años.

El sistema permite la interacción entre humano-computadora, para la detección de palabras dichas por cualquier niño debido a que no solo permite la utilización por parte de las personas registradas en la base de datos.

El nivel de retardo simple de lenguaje se clasifica es: Leve cuando el niño presenta un desarrollo fonológico dentro de un nivel normal y no se advierte distorsiones y dificultades especiales. Moderado cuando se presenta la reducción evidente de patrones fonológicos. Severo cuando tiene reducidos los patrones fonológicos casi al mínimo.

Luego de la realización de las pruebas se obtuvieron los resultados del nivel de retardo simple de lenguaje presente en algunos de los niños que utilizaron el sistema.

4. CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

- Mediante el uso del software Python con sus librerías, se logró elaborar un sistema amigable para los niños de 4 años, donde se realiza una prueba de diagnóstico que permita determinar si presentan o no retardo simple de lenguaje.
- La selección del algoritmo más adecuado se requirió del software Matlab para evaluar el funcionamiento de cada método para el reconocimiento de voz y determinación de la pronunciación correcta o incorrecta en los niños.
- Los usuarios utilizan un sistema que otorga el 89.375% de eficiencia, lo que permite tener un método confiable para la determinación de la presencia de retardo simple de lenguaje en niños de 4 años.
- Al emplear la base de datos con diferentes usuarios otorga la mayor cantidad de resultados efectivos, gracias al diseño del sistema se puede modificar para ampliar más palabras de acuerdo al test de evaluación, sin embargo, este es un sistema independiente del estado de ánimo del usuario.
- Se realiza la comparación entre palabras que tengan similar articulación y fonema, debido a que en nuestro caso se requiere detectar la correcta o incorrecta pronunciación del usuario.
- Debido a que las pruebas se realizaron en ambientes reales, se obtuvieron algunos resultados erróneos, ya que por la presencia de ruido en el ambiente se genera problemas en el desarrollo de la comparación de palabras, porque el cálculo de los coeficientes LPC varía produciendo un desfase en los valores a comparar en la distancia Euclidiana.

4.2. Recomendaciones

Luego de haber realizado las conclusiones del proyecto se presenta las siguientes recomendaciones:

- Para obtener señales de voz más puras es necesario seleccionar un micrófono unidireccional, lo que permitirá captar de mejor manera la señal de la voz.
- El número de coeficientes que se va a calcular al utilizar LPC se determina con la relación entre la frecuencia para un tiempo de 10ms para disminuir el error al comparar la señal del patrón de la base de datos y la señal ingresada por el usuario.
- La generación de la base de datos mientras mayor número de audios tenga, mejor reconocimiento se obtiene.
- Para realizar las pruebas y obtener mejores resultados se recomienda ejecutar el sistema en un ambiente que no genere mucho ruido.





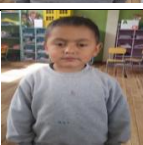





5. BIBLIOGRAFÍA

1. P, A. F. S. (2012). Algoritmo para el Reconocimiento de Comandos de Voz, *24*(34), 131–141.
2. Artigas, J., Rigau, E., & García-Nonell, K. (2008). Trastornos del lenguaje.
3. Arellano, E., Elena, M., Briseño, A., & Benito, J. (2003). Disponible en: <http://www.redalyc.org/articulo.oa?id=94402206>.
4. Teplitsky, V., & Frigon, J. (2000). Implementation of Linear Predictive Coding (LPC) of Speech, 1–15.
5. Miyara, F., & Preliminares, C. La voz humana, 1–10.
6. Farfán, F. (2013). The human voice and its classification, 50–53.
7. Fern, V. (1997). Antecedentes y desarrollo de los sistemas actuales de reconocimiento, *1*, 321–346.
8. Artigas, J., Rigau, E., & García-Nonell, K. (2008). Trastornos del lenguaje, (Bayley III).
9. Miranda-palma, C., Camal-Uc, R., Cen-magaña, J., Gonzalez-Segura, C., Gonzalez-Segura, S., García, M., & Narvaez-Díaz, L. (2007). Un Juego de Gravedad con Reconocimiento de Voz para Niños con Problemas de Lenguaje.
10. Jacqueline, N., Apolo, K. (2017). IMPLEMENTACIÓN DE UN MÓDULO DE RECONOCIMIENTO DE VOZ PARA NIÑOS MEDIANTE EL PROCESAMIENTO DE SEÑALES APLICADO EN UN CASO PRÁCTICO. ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO.
11. Padilla, W., Macas, D (2012). ESTUDIO DE LOS MODELOS OCULTOS DE MARKOV Y DESARROLLO DE UN PROTOTIPO PARA EL RECONOCIMIENTO AUTOMÁTICO DEL HABLA. Estudio de la Universidad politécnica salesiana.
12. Russell, M., Brown, C., Skilling, A., Series, R., Wallace, J., Bonham, B., Sn, W. APPLICATIONS OF AUTOMATIC SPEECH RECOGNITION TO SPEECH AND LANGUAGE DEVELOPMENT IN YOUNG CHILDREN.
13. Chávez, I, Camarena-Ibarrola, A. (2011). Wavelets en el Reconocimiento de voz, 133–137.

14. Parra, L., (2015) DESARROLLO DE UN VIDEO JUEGO CON RECONOCIMIENTO DE VOZ PARA LA SELECCIÓN DE COLORES Y FIGURAS GEOMÉTRICAS. 1–99.
15. Ramírez, M., Sotaquirá, M., De la Cruz, Esther, M., Avellaneda, G., & Ochoa, A. (2016). An Automatic Speech Recognition System for Helping Visually Impaired Children to Learn Braille, 3–6.
16. Rossum, G., Drake, Fred. Septiembre, T. (2009). El tutorial de Python.
17. Tonato, Y., (2015). “EL ENTORNO FAMILIAR Y SU RELACIÓN EN EL RETRASO SIMPLE DEL LENGUAJE EN NIÑOS DE 1 A 3 AÑOS EN EL CENTRO DE DESARROLLO INTEGRAL “MELITAS GARDEN” DE LA CIUDAD DE AMBATO”. Universidad técnica de Ambato facultad de ciencias de la salud carrera de estimulación temprana.
18. Práctica, T. Y. U. Descripción equipos de audio micrófonos, tipos y utilización práctica.
19. Alejandro, C., Ortega, D. L., González, M. M., & Martínez-romo, J. C. (2012). Reconocedor de Palabras con el uso de Regresión Lineal y Coeficiente Muestral, (44), 5–9.
20. Pérez, J., Borrero, H (2010) Identificación de Comandos de Voz utilizando LPC y Algoritmos Genéticos en Matlab. 38-50
21. Fco, J., & Ygual, A. (2001). Evaluación e intervención en niños con trastornos fonológicos y riesgo de dificultad de aprendizaje de la lectura y escritura, 1-42.
22. Raspberry Pi
23. SainSmart
24. MathWorks, MATLAB
25. Numpy.org, 2017
26. Python.org

2. APÉNDICES Y ANEXOS

ANEXO 1 Listado de niños que forman la base de datos

1	Leonel Solano	
2	Daniel Vizquete	
3	Abigail Martinez	
4	Shirley Asqui	
5	Luigui Chávez	
6	Marcelo Lamiña	
7	Lisbeth Medina	
8	Sebastian Colcha	
9	Joe Allauca	
10	Mishell Garcés	

ANEXO 2 Listado de niños que realizan las pruebas

1	Romina Vinueza
2	Jhon Oñate
3	Marcela Paredes
4	Ismael Benítez
5	Ismael Murillo
6	Daniel Vizuite
7	Joe Martínez
8	Lisbeth Medina

ANEXO 3 Glosario

Análisis cepstral: Se trata de calcular la transformada inversa de Fourier del espectro de la señal en escala logarítmica (dB)

Técnicas de warping expansión: Es la modificación de escalado de tiempo especificando un mapeado entre el tiempo en la señal original y el tiempo en la señal modificada.

LPC: Es un algoritmo que conoce lo que ocurrirá en el presente, considerando la información del pasado de la señal

Distancia Euclidiana: Se encarga de calcular la menor distancia que existe entre el patrón generado en la base de datos y la palabra ingresada por el usuario.

Python: Software libre, es un lenguaje de programación orientado a objetos, posee una amplia biblioteca y estructuras de alto nivel que se encuentran disponibles libremente.

ANEXO 4 Test

DEFINICIÓN

Se trata de una prueba de denominación de imágenes que pretende determinar el grado de la pronunciación. Está basada en la propuesta de Dodd (1995).

La pronunciación cuando la palabra es pronunciada de manera correcta o incorrecta.

Se usa un conjunto de 25 imágenes que se presenta al niño. Entre cada presentación se deja transcurrir unos minutos en los que el niño realiza alguna actividad.

Se le pide el niño que diga el nombre de la figura. Debe obtenerse una respuesta por denominación no por repetición. Se pueden dar ayudas semánticas (Es una fruta..., es un animal ...) o sintácticas (Por la carretera van los...), pero no se le debe decir la palabra para que la repita. Si el niño desconoce el nombre de la figura no queda más remedio que decirla, pero no se anota su intento de pronunciarla hasta que vuelva hacer cuando haya pasado algunos minutos.

Definimos como “intento” cualquier forma de pronunciar la palabra, sea correcta o incorrecta. No contabilizamos el intento si el niño se equivoca de palabra (ejemplo: dice “coche” ante la imagen de una moto).

Se anota la pronunciación espontánea del niño. Si realiza varios intentos cada vez se anotan esos intentos.

No se admiten los intentos obtenidos en repetición.

La valoración que se da cuando el niño tiene de 1 a 2 palabras mal pronunciadas, leve de 3 a 4 moderado de 5 para adelante grave, de la dificultad de retardo simple de lenguaje dependiendo del evaluador y de otras pruebas recomienda el tratamiento adecuado.

ANEXO 5 Código

```
from Tkinter import *
import Image
import sys
import numpy
import wave
import math
from scipy.signal import lfilter, hamming
from scikits.talkbox import lpc
import numpy as np
import time
import sounddevice as sd
import soundfile as sf
global a1
global patronchocolate, patronelefante
global patronplato, patronconejo
##dist1, dist2
#####
## Funcion de grabacion la palabra chocolate
#####
def grabacionchocolate():
    global a1
    global b1, patronchocolate, patronelefante
    fs=44100
    a1=0
    b1=0
#entrada del usuario

    duracion=3
    grabacion23=sd.rec(int(duracion*fs), samplerate=fs, channels=1, dtype='float32')
    sd.wait()
    sd.play(grabacion23,fs)
    sf.write('grab1.wav', grabacion23, fs)

    g= wave.open('grab1.wav')
    x=g.readframes(-1)
    x=numpy.fromstring(x, 'Int16')
    N=len(x)
    w=numpy.hamming(N)
    x1=x*w
    x1=lfilter( [1., -0.95], [1], x1)
    A, e, k=lpc(x1, order=23)
#print (A)

#base de datos
#chocolate
cho=wave.open('limchocolate.wav')
xcho=cho.readframes(-1)
xcho=numpy.fromstring(xcho, 'Int16')
```



```

Ncho=len(xcho)
wcho=numpy.hamming(Ncho)
x1cho=xcho*wcho
x1cho=lfilter([1., -0.95], [1], x1cho)
ent_cho, e, k=lpc(x1cho, order=23)
#ent_cho= np.array((ent_cho))
#print(ent_cho)

#chocolate1
cho1=wave.open('limchocolate1.wav')
xcho1=cho1.readframes(-1)
xcho1=numpy.fromstring(xcho1, 'Int16')
Ncho1=len(xcho1)
wcho1=numpy.hamming(Ncho1)
x1cho1=xcho1*wcho1
x1cho1=lfilter([1., -0.95], [1], x1cho1)
ent_cho1,e,k=lpc(x1cho1, order=23)
#ent_cho1= np.array((ent_cho1))
#print(ent_cho1)

#chocolate2
cho2=wave.open('limchocolate2.wav')
xcho2=cho2.readframes(-1)
xcho2=numpy.fromstring(xcho2, 'Int16')
Ncho2=len(xcho2)
wcho2=numpy.hamming(Ncho2)
x1cho2=xcho2*wcho2
x1cho2=lfilter([1., -0.95], [1], x1cho2)
ent_cho2, e, k=lpc(x1cho2, order=23)
#print(ent_cho2)

#chocolate3
cho3=wave.open('limchocolate3.wav')
xcho3=cho3.readframes(-1)
xcho3=numpy.fromstring(xcho3, 'Int16')
Ncho3=len(xcho3)
wcho3=numpy.hamming(Ncho3)
x1cho3=xcho3*wcho3
x1cho3=lfilter([1., -0.95], [1], x1cho3)
ent_cho3, e, k=lpc(x1cho3, order=23)
#print(ent_cho3)

#chocolate4
cho4=wave.open('limchocolate4.wav')
xcho4=cho4.readframes(-1)
xcho4=numpy.fromstring(xcho4, 'Int16')
Ncho4=len(xcho4)
wcho4=numpy.hamming(Ncho4)
x1cho4=xcho4*wcho4

```

```

x1cho4=lfilter([1., -0.95], [1], x1cho4)
ent_cho4, e, k=lpc(x1cho4, order=23)
#print(ent_cho4)

#chocolate5
cho5=wave.open('limchocolate5.wav')
xcho5=cho5.readframes(-1)
xcho5=numpy.fromstring(xcho5, 'Int16')
Ncho5=len(xcho5)
wcho5=numpy.hamming(Ncho5)
x1cho5=xcho5*wcho5
x1cho5=lfilter([1., -0.95], [1], x1cho5)
ent_cho5, e, k=lpc(x1cho5, order=23)
#print(ent_cho5)

#chocolate6
cho6=wave.open('limchocolate6.wav')
xcho6=cho6.readframes(-1)
xcho6=numpy.fromstring(xcho6, 'Int16')
Ncho6=len(xcho6)
wcho6=numpy.hamming(Ncho6)
x1cho6=xcho6*wcho6
x1cho6=lfilter([1., -0.95], [1], x1cho6)
ent_cho6, e, k=lpc(x1cho6, order=23)
#print(ent_cho6)

#chocolate7
cho7=wave.open('limchocolate7.wav')
xcho7=cho7.readframes(-1)
xcho7=numpy.fromstring(xcho7, 'Int16')
Ncho7=len(xcho7)
wcho7=numpy.hamming(Ncho7)
x1cho7=xcho7*wcho7
x1cho7=lfilter([1., -0.95], [1], x1cho7)
ent_cho7, e, k=lpc(x1cho7, order=23)
#print(ent_cho7)

#chocolate8
cho8=wave.open('limchocolate8.wav')
xcho8=cho8.readframes(-1)
xcho8=numpy.fromstring(xcho8, 'Int16')
Ncho8=len(xcho8)
wcho8=numpy.hamming(Ncho8)
x1cho8=xcho8*wcho8
x1cho8=lfilter([1., -0.95], [1], x1cho8)
ent_cho8, e, k=lpc(x1cho8, order=23)
#print(ent_cho8)

#chocolate9
cho9=wave.open('limchocolate9.wav')

```

```

xcho9=cho9.readframes(-1)
xcho9=numpy.fromstring(xcho9, 'Int16')
Ncho9=len(xcho9)
wcho9=numpy.hamming(Ncho9)
x1cho9=xcho9*wcho9
x1cho9=lfilter([1., -0.95], [1], x1cho9)
ent_cho9, e, k=lpc(x1cho9, order=23)
#print(ent_cho9)

#chocolate10
cho10=wave.open('limchocolate.wav')
xcho10=cho10.readframes(-1)
xcho10=numpy.fromstring(xcho10, 'Int16')
Ncho10=len(xcho10)
wcho10=numpy.hamming(Ncho10)
x1cho10=xcho10*wcho10
x1cho10=lfilter([1., -0.95], [1], x1cho10)
ent_cho10, e, k=lpc(x1cho10, order=23)
#print(ent_cho10)

#elefante
ele=wave.open('limelefante.wav')
xele=ele.readframes(-1)
xele=numpy.fromstring(xele, 'Int16')
Nele=len(xele)
wele=numpy.hamming(Nele)
x1ele=xele*wele
x1ele=lfilter([1., -0.95], [1], x1ele)
ent_ele, e, k=lpc(x1ele, order=23)
#print(ent_ele)

#elefante1
ele1=wave.open('limelefante1.wav')
xele1=ele1.readframes(-1)
xele1=numpy.fromstring(xele1, 'Int16')
Nele1=len(xele1)
wele1=numpy.hamming(Nele1)
x1ele1=xele1*wele1
x1ele1=lfilter([1., -0.95], [1], x1ele1)
ent_ele1, e, k=lpc(x1ele1, order=23)
#print(ent_ele1)

#elefante2
ele2=wave.open('limelefante2.wav')
xele2=ele2.readframes(-1)
xele2=numpy.fromstring(xele2, 'Int16')
Nele2=len(xele2)
wele2=numpy.hamming(Nele2)

```

```

x1ele2=xele2*wele2
x1ele2=lfilter([1., -0.95], [1], x1ele2)
ent_ele2, e, k=lpc(x1ele2, order=23)
#print(ent_ele2)

#elefante3
ele3=wave.open('limelefante3.wav')
xele3=ele3.readframes(-1)
xele3=numpy.fromstring(xele3, 'Int16')
Nele3=len(xele3)
wele3=numpy.hamming(Nele3)
x1ele3=xele3*wele3
x1ele3=lfilter([1., -0.95], [1], x1ele3)
ent_ele3, e, k=lpc(x1ele3, order=23)
#print(ent_ele3)

#elefante4
ele4=wave.open('limelefante4.wav')
xele4=ele4.readframes(-1)
xele4=numpy.fromstring(xele4, 'Int16')
Nele4=len(xele4)
wele4=numpy.hamming(Nele4)
x1ele4=xele4*wele4
x1ele4=lfilter([1., -0.95], [1], x1ele4)
ent_ele4, e, k=lpc(x1ele4, order=23)
#print(ent_ele4)

#elefante5
ele5=wave.open('limelefante5.wav')
xele5=ele5.readframes(-1)
xele5=numpy.fromstring(xele5, 'Int16')
Nele5=len(xele5)
wele5=numpy.hamming(Nele5)
x1ele5=xele5*wele5
x1ele5=lfilter([1., -0.95], [1], x1ele5)
ent_ele5, e, k=lpc(x1ele5, order=23)
#print(ent_ele5)

#elefante6
ele6=wave.open('limelefante6.wav')
xele6=ele6.readframes(-1)
xele6=numpy.fromstring(xele6, 'Int16')
Nele6=len(xele6)
wele6=numpy.hamming(Nele6)
x1ele6=xele6*wele6
x1ele6=lfilter([1., -0.95], [1], x1ele6)
ent_ele6, e, k=lpc(x1ele6, order=23)
#print(ent_ele6)

```

```

#elefante7
ele7=wave.open('limelefante7.wav')
xele7=ele7.readframes(-1)
xele7=numpy.fromstring(xele7, 'Int16')
Nele7=len(xele7)
wele7=numpy.hamming(Nele7)
x1ele7=xele7*wele7
x1ele7=lfilter([1., -0.95], [1], x1ele7)
ent_ele7, e, k=lpc(x1ele7, order=23)
#print(ent_ele7)

#elefante8
ele8=wave.open('limelefante8.wav')
xele8=ele8.readframes(-1)
xele8=numpy.fromstring(xele8, 'Int16')
Nele8=len(xele8)
wele8=numpy.hamming(Nele8)
x1ele8=xele8*wele8
x1ele8=lfilter([1., -0.95], [1], x1ele8)
ent_ele8, e, k=lpc(x1ele8, order=23)
#print(ent_ele8)

#elfante9
ele9=wave.open('limelefante9.wav')
xele9=ele9.readframes(-1)
xele9=numpy.fromstring(xele9, 'Int16')
Nele9=len(xele9)
wele9=numpy.hamming(Nele9)
x1ele9=xele9*wele9
x1ele9=lfilter([1., -0.95], [1], x1ele9)
ent_ele9, e, k=lpc(x1ele9, order=23)
#print(ent_ele9)

#elefante10
ent_ele10=[]
ele10=wave.open('limelefante.wav')
xele10=ele10.readframes(-1)
xele10=numpy.fromstring(xele10, 'Int16')
Nele10=len(xele10)
wele10=numpy.hamming(Nele10)
x1ele10=xele10*wele10
x1ele10=lfilter([1., -0.95], [1], x1ele10)
##print (x1ele10)
ent_ele10, e, k=lpc(x1ele10, order=23)

pch=[]
#patron chocolate

#patronchocolate=(np.sum([ent_cho, ent_cho1, ent_cho2, ent_cho3, ent_cho4, ent_cho5,

```

```

ent_cho6, ent_cho7, ent_cho8, ent_cho9, ent_cho10]))
#print (ent_cho)
#print (ent_cho1)
    patronchocolate= ent_cho + ent_cho1 + ent_cho2 + ent_cho3 + ent_cho4 + ent_cho5 +
ent_cho6 + ent_cho7 + ent_cho8 + ent_cho9 + ent_cho10
    patronchocolate1= patronchocolate /11

#print (patronchocolate)
#print (patronchocolate1)

#patron elefante
    patronelefante=ent_ele + ent_ele1 + ent_ele2 + ent_ele3 + ent_ele4 + ent_ele5 +
ent_ele6 + ent_ele7 + ent_ele8 + ent_ele9 + ent_ele10
    patronelefante1=patronelefante /11
#distancia euclideana
#distancia chocolate
    dis1=((patronchocolate1-A)**2
##dis1=np.dot(dis1, dis1)

##print (dis1)
    dist1=np.sum(dis1)
#distancia elefante
    dis2=(patronelefante1-A)**2
#dis2=np.dot(dis2, dis2)
    dist2=np.sum(dis2)
    # print (dist1)
    # print (dist2)
#comparacion
    if dist1<dist2:
        print('correcto')
        a1=a1+1
    else:
        print('incorrecto')

    # print (a1)
    # print (b1)

def grabacionelefante():
    global a1
    global b1, patronchocolate, patronelefante
    fs=44100
    a1=a1
#entrada del usuario

    duracion=3
    grabacion23=sd.rec(int(duracion*fs), samplerate=fs, channels=1, dtype='float32')
    sd.wait()
    sd.play(grabacion23,fs)
    sf.write('grab2.wav', grabacion23, fs)

```

```

g= wave.open('grab2.wav')
x=g.readframes(-1)
x=numpy.fromstring(x, 'Int16')
N=len(x)
w=numpy.hamming(N)
x1=x*w
x1=lfilter( [1., -0.95], [1], x1)
A, e, k=lpc(x1, order=23)
#print (A)

patronchocolate1= patronchocolate /11

patronelefante1=patronelefante /11

dis1=((patronchocolate1-A)**2
dist1=np.sum(dis1)
dis2=(patronelefante1-A)
dist2=np.sum(dis2)

if dist2<dist1:
    print('correcto')
    a1=a1+1
else:
    print('incorrecto')

print (a1)

def grabacionvaca():
    global a1
    global b1, patronplato
    fs=44100
    a1=a1
#entrada del usuario

    duracion=3
    grabacion23=sd.rec(int(duracion*fs), samplerate=fs, channels=1, dtype='float32')
    sd.wait()
    sd.play(grabacion23,fs)
    sf.write('grab3.wav', grabacion23, fs)

    g= wave.open('grab3.wav')
    x=g.readframes(-1)
    x=numpy.fromstring(x, 'Int16')
    N=len(x)
    w=numpy.hamming(N)
    x1=x*w
    x1=lfilter( [1., -0.95], [1], x1)
    A, e, k=lpc(x1, order=23)

```

```

#base de datos
vaca=wave.open('limvaca.wav')
xvaca=vaca.readframes(-1)
xvaca=numpy.fromstring(xvaca, 'Int16')
Nvaca=len(xvaca)
wvaca=numpy.hamming(Nvaca)
x1vaca=xvaca*wvaca
x1vaca=lfilter([1., -0.95], [1], x1vaca)
ent_vaca, e, k=lpc(x1vaca, order=23)

#vaca1
vaca1=wave.open('limvaca1.wav')
xvaca1=vaca1.readframes(-1)
xvaca1=numpy.fromstring(xvaca1, 'Int16')
Nvaca1=len(xvaca1)
wvaca1=numpy.hamming(Nvaca1)
x1vaca1=xvaca1*wvaca1
x1vaca1=lfilter([1., -0.95], [1], x1vaca1)
ent_vaca1, e, k=lpc(x1vaca1, order=23)

#vaca2
vaca2=wave.open('limvaca2.wav')
xvaca2=vaca2.readframes(-1)
xvaca2=numpy.fromstring(xvaca2, 'Int16')
Nvaca2=len(xvaca2)
wvaca2=numpy.hamming(Nvaca2)
x1vaca2=xvaca2*wvaca2
x1vaca2=lfilter([1., -0.95], [1], x1vaca2)
ent_vaca2, e, k=lpc(x1vaca2, order=23)

#vaca3
vaca3=wave.open('limvaca3.wav')
xvaca3=vaca3.readframes(-1)
xvaca3=numpy.fromstring(xvaca3, 'Int16')
Nvaca3=len(xvaca3)
wvaca3=numpy.hamming(Nvaca3)
x1vaca3=xvaca3*wvaca3
x1vaca3=lfilter([1., -0.95], [1], x1vaca3)
ent_vaca3, e, k=lpc(x1vaca3, order=23)

#vaca4
vaca4=wave.open('limvaca4.wav')
xvaca4=vaca4.readframes(-1)
xvaca4=numpy.fromstring(xvaca4, 'Int16')
Nvaca4=len(xvaca4)
wvaca4=numpy.hamming(Nvaca4)
x1vaca4=xvaca4*wvaca4
x1vaca4=lfilter([1., -0.95], [1], x1vaca4)
ent_vaca4, e, k=lpc(x1vaca4, order=23)

```



```

#vaca5
vaca5=wave.open('limvaca5.wav')
xvaca5=vaca5.readframes(-1)
xvaca5=numpy.fromstring(xvaca5, 'Int16')
Nvaca5=len(xvaca5)
wvaca5=numpy.hamming(Nvaca5)
x1vaca5=xvaca5*wvaca5
x1vaca5=lfilter([1., -0.95], [1], x1vaca5)
ent_vaca5, e, k=lpc(x1vaca5, order=23)

```

```

#vaca6
vaca6=wave.open('limvaca6.wav')
xvaca6=vaca6.readframes(-1)
xvaca6=numpy.fromstring(xvaca6, 'Int16')
Nvaca6=len(xvaca6)
wvaca6=numpy.hamming(Nvaca6)
x1vaca6=xvaca6*wvaca6
x1vaca6=lfilter([1., -0.95], [1], x1vaca6)
ent_vaca6, e, k=lpc(x1vaca6, order=23)

```

```

#vaca7
vaca7=wave.open('limvaca7.wav')
xvaca7=vaca7.readframes(-1)
xvaca7=numpy.fromstring(xvaca7, 'Int16')
Nvaca7=len(xvaca7)
wvaca7=numpy.hamming(Nvaca7)
x1vaca7=xvaca7*wvaca7
x1vaca7=lfilter([1., -0.95], [1], x1vaca7)
ent_vaca7, e, k=lpc(x1vaca7, order=23)

```

```

#vaca8
vaca8=wave.open('limvaca8.wav')
xvaca8=vaca8.readframes(-1)
xvaca8=numpy.fromstring(xvaca8, 'Int16')
Nvaca8=len(xvaca8)
wvaca8=numpy.hamming(Nvaca8)
x1vaca8=xvaca8*wvaca8
x1vaca8=lfilter([1., -0.95], [1], x1vaca8)
ent_vaca8, e, k=lpc(x1vaca8, order=23)

```

```

#vaca9
vaca9=wave.open('limvaca9.wav')
xvaca9=vaca9.readframes(-1)
xvaca9=numpy.fromstring(xvaca9, 'Int16')
Nvaca9=len(xvaca9)
wvaca9=numpy.hamming(Nvaca9)
x1vaca9=xvaca9*wvaca9
x1vaca9=lfilter([1., -0.95], [1], x1vaca9)
ent_vaca9, e, k=lpc(x1vaca9, order=23)

```

```

#vaca10
vaca10=wave.open('limvaca.wav')
xvaca10=vaca10.readframes(-1)
xvaca10=numpy.fromstring(xvaca10, 'Int16')
Nvaca10=len(xvaca10)
wvaca10=numpy.hamming(Nvaca10)
x1vaca10=xvaca10*wvaca10
x1vaca10=lfilter([1., -0.95], [1], x1vaca10)
ent_vaca10, e, k=lpc(x1vaca10, order=23)

```

```

#plato
plato=wave.open('limplato.wav')
xplato=plato.readframes(-1)
xplato=numpy.fromstring(xplato, 'Int16')
Nplato=len(xplato)
wplato=numpy.hamming(Nplato)
x1plato=xplato*wplato
x1plato=lfilter([1., -0.95], [1], x1plato)
ent_plato, e, k=lpc(x1plato, order=23)

```

```

#plato1
plato1=wave.open('limplato1.wav')
xplato1=plato1.readframes(-1)
xplato1=numpy.fromstring(xplato1, 'Int16')
Nplato1=len(xplato1)
wplato1=numpy.hamming(Nplato1)
x1plato1=xplato1*wplato1
x1plato1=lfilter([1., -0.95], [1], x1plato1)
ent_plato1, e, k=lpc(x1plato1, order=23)

```

```

#plato2
plato2=wave.open('limplato2.wav')
xplato2=plato2.readframes(-1)
xplato2=numpy.fromstring(xplato2, 'Int16')
Nplato2=len(xplato2)
wplato2=numpy.hamming(Nplato2)
x1plato2=xplato2*wplato2
x1plato2=lfilter([1., -0.95], [1], x1plato2)
ent_plato2, e, k=lpc(x1plato2, order=23)

```

```

#plato3
plato3=wave.open('limplato3.wav')
xplato3=plato3.readframes(-1)
xplato3=numpy.fromstring(xplato3, 'Int16')
Nplato3=len(xplato3)
wplato3=numpy.hamming(Nplato3)

```

```

x1plato3=xplato3*wplato3
x1plato3=lfilter([1., -0.95], [1], x1plato3)
ent_plato3, e, k=lpc(x1plato3, order=23)

#plato4
plato4=wave.open('limplato4.wav')
xplato4=plato4.readframes(-1)
xplato4=numpy.fromstring(xplato4, 'Int16')
Nplato4=len(xplato4)
wplato4=numpy.hamming(Nplato4)
x1plato4=xplato4*wplato4
x1plato4=lfilter([1., -0.95], [1], x1plato4)
ent_plato4, e, k=lpc(x1plato4, order=23)

#plato5
plato5=wave.open('limplato5.wav')
xplato5=plato5.readframes(-1)
xplato5=numpy.fromstring(xplato5, 'Int16')
Nplato5=len(xplato5)
wplato5=numpy.hamming(Nplato5)
x1plato5=xplato5*wplato5
x1plato5=lfilter([1., -0.95], [1], x1plato5)
ent_plato5, e, k=lpc(x1plato5, order=23)

#plato6
plato6=wave.open('limplato6.wav')
xplato6=plato6.readframes(-1)
xplato6=numpy.fromstring(xplato6, 'Int16')
Nplato6=len(xplato6)
wplato6=numpy.hamming(Nplato6)
x1plato6=xplato6*wplato6
x1plato6=lfilter([1., -0.95], [1], x1plato6)
ent_plato6, e, k=lpc(x1plato6, order=23)

#plato7
plato7=wave.open('limplato7.wav')
xplato7=plato7.readframes(-1)
xplato7=numpy.fromstring(xplato7, 'Int16')
Nplato7=len(xplato7)
wplato7=numpy.hamming(Nplato7)
x1plato7=xplato7*wplato7
x1plato7=lfilter([1., -0.95], [1], x1plato7)
ent_plato7, e, k=lpc(x1plato7, order=23)

#plato8
plato8=wave.open('limplato8.wav')
xplato8=plato8.readframes(-1)
xplato8=numpy.fromstring(xplato8, 'Int16')
Nplato8=len(xplato8)

```

```

wplato8=npumpy.hamming(Nplato8)
x1plato8=xplato8*wplato8
x1plato8=lfilter([1., -0.95], [1], x1plato8)
ent_plato8, e, k=lpc(x1plato8, order=23)

#plato9
plato9=wave.open('limplato9.wav')
xplato9=plato9.readframes(-1)
xplato9=npumpy.fromstring(xplato9, 'Int16')
Nplato9=len(xplato9)
wplato9=npumpy.hamming(Nplato9)
x1plato9=xplato9*wplato9
x1plato9=lfilter([1., -0.95], [1], x1plato9)
ent_plato9, e, k=lpc(x1plato9, order=23)

#plato0
plato10=wave.open('limplato.wav')
xplato10=plato10.readframes(-1)
xplato10=npumpy.fromstring(xplato10, 'Int16')
Nplato10=len(xplato10)
wplato10=npumpy.hamming(Nplato10)
x1plato10=xplato10*wplato10
x1plato10=lfilter([1., -0.95], [1], x1plato10)
ent_plato10, e, k=lpc(x1plato10, order=23)

#patron vaca

patronvaca= ent_vaca + ent_vaca1 + ent_vaca2 + ent_vaca3 + ent_vaca4 + ent_vaca5
+ ent_vaca6 + ent_vaca7 + ent_vaca8 + ent_vaca9 + ent_vaca10
patronvaca1= patronvaca /11

#patron plato
patronplato=ent_plato + ent_plato1 + ent_plato2 + ent_plato3 + ent_plato4 +
ent_plato5 + ent_plato6 + ent_plato7 + ent_plato8 + ent_plato9 + ent_plato10
patronplato1=patronplato /11

#distancia euclidean
#distancia vaca
dis1=((patronvaca1-A)**2
dist1=np.sum(dis1)

dis2=(patronplato1-A)**2
dist2=np.sum(dis2)
#comparacion
if dist1<dist2:
print('correcto')
a1=a1+1

```

```
else:  
    print('incorrecto')
```

```
print (a1)
```

```
#####  
#####Fin de la palabra vaca grabacion  
#####
```

