UNIVERSIDAD SEÑOR DE SIPÁN

FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO

Escuela Académico Profesional de Ingeniería de Sistemas

Tesis

EVALUACIÓN DE MÉTODOS PARA EL RECONOCIMIENTO DE COMANDOS DE VOZ PARA PERSONAS CON DISLEXIA

PARA OPTAR EL TÍTULO EN INGENIERÍA DE SISTEMAS

Autor:

Bach. Andy Víctor Hugo Ruiz Vargas

Asesor Metodológico:
Víctor Alexci Tuesta Monteza

Asesor Especialista
Dr. Ing. Jorge Gutiérrez Gutiérrez

Pimentel, Diciembre de 2018

Título de Tesis

“EVALUACIÓN DE MÉTODOS PARA EL RECONOCIMIENTO DE COMANDOS DE VOZ PARA PERSONAS CON DISLEXIA”.

Aprobación de tesis

Ing.Víctor Alexci Tuesta Monteza	Dr. Ing. Gutiérrez Gutiérrez Jorge
Asesor metodólogo	**Asesor especialista**

Ing. Mejía Cabrera Heber Iván

Presidente del jurado de tesis

Ing. José Manuel Bruno Sarmiento	Ing. Víctor Alexci Tuesta Monteza
Secretario del jurado de tesis	**Vocal del jurado de tesis**

DEDICATORIA

A mis padres, porque todo lo que soy se lo debo a ellos y por ellos me inspiro para finalizar este proyecto.

Agradezco a mis compañeros por el apoyo y la compañía, que me dieron palabras de aliento y me brindaron información. Cuento con ustedes siempre.

A esas personas que han estado conmigo apoyándome a lo largo de mi carrera y que significan mucho para mí. Quiero dedicarles esta tesis.

A mi asesor especialista y metodólogo por darse un tiempo para que me brinde de sus conocimientos a lo largo de esta etapa de finalizar mi proyecto.

AGRADECIMIENTO

A Dios, por permitirme estar en este momento tan especial de mi vida. Por los triunfos y los momentos difíciles, que me han enseñado a valorar lo que tengo cada día más.

A mis padres, porque creyeron en mí y porque confiaron en mí, y hoy puedo ver alcanzado mi meta, ya que siempre estuvieron impulsándome en los momentos más difíciles de mi carrera. Va por ustedes, por lo que valen, porque admiro su fortaleza y por todo lo que han hecho por mí.

A mi familia por el apoyo incondicional. Gracias por haber fomentado en mí, el deseo de superación y el anhelo de triunfo en la vida.

A todos, espero no defraudarlos y contar siempre con su valioso apoyo, sincero e incondicional.

INDICE DE CONTENIDO

“EVALUACIÓN DE MÉTODOS PARA EL RECONOCIMIENTO DE COMANDOS DE VOZ PARA PERSONAS CON DISLEXIA”. II

DEDICATORIA *3*

AGRADECIMIENTO *4*

INDICE DE CONTENIDO *5*

INDICE DE GRAFICAS *8*

RESUMEN *11*

ABSTRACT *12*

INTRODUCCION *13*

CAPITULO I: PROBLEMA DE INVESTIGACIÓN *15*

1.1. Situación problemática. *15*

1.2. Formulación del problema. *18*

1.3. Delimitación de la investigación. *18*

1.4. Justificación e importancia de la investigación. *18*

1.5. Limitaciones de la investigación. *19*

1.6. Objetivos de la investigación. *19*

CAPITULO II: MARCO TEÓRICO *20*

2.1. Antecedentes de estudios *20*

2.1.1. A nivel internacional: *20*

2.2. Estado del arte *21*

2.3. Sistemas teórico conceptuales. *25*

2.3.1. Sistemas de reconocimiento automático del habla (RAH). *25*

2.3.2. Procesamiento de Señales. *25*

2.3.3. Algoritmos de Procesamiento de señales. *26*

2.3.3.1. Predicción Lineal. *26*

2.3.3.2. Cepstrum. *26*

2.3.3.3. Predicción Lineal de los Coeficientes Cepstrales (LPCC). *29*

2.3.3.4. Análisis de Fourier *32*

2.3.3.5. Coeficientes Cepstrales de Frecuencia Mel (MFCC). *32*

2.3.3.6. Coeficientes Cepstrales de Frecuencia Lineal (LFCC). *35*

2.3.3.7. Transformada Wavelet.....37
2.3.4. Reconocimiento de voz. 38
2.3.5. Algoritmos de Reconocimiento de voz.....38
2.3.5.1. Modelos ocultos de Márkov. 38
2.3.5.2. Alineamiento temporal dinámico (DTW).....38
2.3.5.3. Redes Neuronales (NN).....38
2.3.5.4. Cuantificación Vectorial aplicada (VQ).....39
2.3.5.5. K-Vecinos más cercanos (KNN).....39
2.3.5.6. Modelos de Mezclas de Gaussianas (GMM).....39
2.4. Definición de la terminología. 40
CAPITULO III: MARCO METODOLÓGICO 42
3.1. Tipo y diseño de la investigación.....42
3.1.1. Tipo de investigación. 42
3.1.2. Diseño de investigación 42
3.2. Población y Muestra. 42
3.2.1. Población.....42
3.2.2. Muestra: 43
3.3. Hipótesis.....43
3.4. Variables 43
3.4.1. Variable de estudio. 43
3.5. Operacionalización.....44
3.6. Abordaje metodológico, técnicas e instrumentos de recolección de datos 44
3.6.1. Abordaje metodológico. 45
3.6.2. Técnicas de recolección de datos. 45
3.6.3. Instrumentos de recolección de datos. 46
3.7. Procedimiento para la recolección de datos 46
3.8. Análisis estadístico e interpretación de los datos 47
3.9. Principios éticos 48
3.10. Criterios de rigor científico. 48
4. CAPITULO IV: ANÁLISIS E INTERPRETACIÓN DE LOS RESULTADOS 49

5. CAPITULO V: PROPUESTA DE INVESTIGACIÓN 52
5.1. Base de datos con audios de comando de voz de personas con dislexia. 54
5.1.1. Creación de Base de datos..... 54
5.1.2. Seleccionar Métodos de Procesamiento Y Reconocimiento de Voz 56
5.1.2.1. Codificación Predictiva Lineal 57
5.1.2.2. Coeficientes Cepstrales de la escala de Mel 59
5.1.3. Extracción de características 64
CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES 80
6.1 Conclusiones 80
6.2 Recomendaciones 80
BIBLIOGRAFÍA..... 81

INDICE DE GRAFICAS

Gráfica 1: Volumen y evolución del mercado biométrico global..... 15

Gráfica 2: Segmentación de mercado por tecnología biométrica (2013).16

Gráfica 3: Evolución del volumen de mercado por tecnologías biométricas 16

Gráfica 4: Proceso de reconocimiento de voz. 25

Gráfica 5: Espectrograma: Señal de voz original..... 28

Gráfica 6: Espectrograma: Señal de voz con Cepstrum 28

Gráfica 7: Modelo simplificado de producción de la voz 30

Gráfica 8: Correspondencia entre la frecuencia en Hz y la frecuencia Mel 32

Gráfica 9: Diagrama de bloques para el cálculo de los MFCC's 33

Gráfica 10: Banco de filtros espaciados linealmente en la escala de frecuencia Mel 34

Gráfica 11: Vector acústico generado mediante el cálculo de los MFCC's 35

Gráfica 12: Banco de filtros generado para el cálculo de los LFCC's .. 36

Gráfica 13: Vector acústico generado por el cálculo de los LFCC's 36

Gráfica 14: Vectores acústicos generados mediante el cálculo de los MFCC's, LPCC's y LFCC's 37

Gráfica 16: Creación de la base de datos 54

Gráfica 17: Grabación de personas con y sin dislexia 55

Gráfica 18: Señal de audio LPC 59

Gráfica 19: Segmentación de la Señal .. **¡Error! Marcador no definido.**

Gráfica 20: Banco de filtros triangulares implementado. 64

Gráfica 21: Rango de la Energia de la voz 65

Gráfica 22: Rango de Root Half Square de la voz 66

Gráfica 23: Rango de Zero Crossing Rate de la voz 67

Gráfica 24: Rango de Entropy de la voz 67

Gráfica 26: Rango de la Dispersion Espectral de la voz 69

Gráfica 27: Rango de la Medida de Planitud Espectral de la voz 70

Gráfica 28: Rango de Roll-Off de la Voz: 71

Gráfica 29: Rango de la cresta de la voz 71

Gráfica 30: Rango del Ancho de banda de la voz 72

Gráfica 31: Estructura de la Red Neuronal 74

Gráfica 32: Cantidad de Entrenamientos por Error con 1 Capa 76

Gráfica 33: Cantidad de Entrenamientos por Error 77

INDICE DE TABLAS

Tabla 1. Población (Algoritmos investigados) 42

Tabla 2: Variable de estudio. 44

Tabla 3: Criterios de rigor científico 48

Tabla 4: Características de cada archivo de voz.. 54

Tabla 5 : Base de audios de Personas 55

Tabla 6: Selección Métodos de Procesamiento 56

Tabla 7: Selección Métodos de Reconocimiento 72

Tabla 8 : Salidas Deseadas del Reconocimiento de Voz 74

Tabla 9: Cantidad de Entrenamientos por Error de 1 Capa 76

Tabla 10: Cantidad de Entrenamientos por Error de 2 Capa 77

RESUMEN

Esta investigación presenta el reconocimiento de la voz humana usando visión computacional; para la implementación se utilizó el lenguaje de programación Visual Studio, para las características y algoritmos de procesamiento y reconocimiento, para la precisión el porcentaje fue alcanzado es de 71%. La problemática de que existen pocas investigaciones que presentan un método para el reconocimiento de voz humana, por lo que se presenta un mayor reconocimiento bajo nuestra realidad.

Palabras clave: Inteligencia Artificial, Visión Computacional, Reconocimiento de voz, Redes Neuronales

ABSTRACT

This research presents the recognition of the human voice with computational vision; for the implementation the Visual Studio programming language was used, for the characteristics and algorithms of processing and recognition, for the precision the percentage was reached in 71%. The problem that there is some research that presents a method for the recognition of the human voice, so it is presented as a greater recognition under our reality.

Key Words: Artificial Intelligence, Computational Vision, Speech Recognition, Neural Networks

INTRODUCCION

El presente trabajo titulado “EVALUACIÓN DE MÉTODOS PARA RESOLVER EL RECONOCIMIENTO POR COMANDO DE VOZ” es producto de una ardua investigación, que busca analizar métodos para el reconocimiento por comando de voz en el Campo de la Inteligencia Artificial.

En la actualidad, la biometría se encuentra presente en múltiples aplicaciones tales como el reconocimiento de voz, este informe muestra algoritmos para reconocer la voz según los métodos utilizados.

Principalmente se plantea la problemática de la investigación, se realiza un análisis de asertividad actual del reconocimiento de la voz, la cual es la razón de ser de esta tesis.

También se describe el objetivo general y específico, antecedentes, se muestra el marco metodológico, estado del arte, Sistemas teórico conceptuales, el cual está constituido por el tipo y diseño de la investigación, la población y muestra, la hipótesis, variables, Operacionalización, Métodos, técnicas e instrumentos de recolección de datos, procedimiento para la recolección de datos, principios éticos y criterios de rigor científico.

El proyecto dará a conocer el reconocimiento de voz, respuesta a la dificultad humana existente en la actualidad. El reconocedor de voz debe registrar datos, como ausencia de ruidos, reverberaciones, etc.

Con estas características obtenidas, se construye un conjunto de vectores que constituyen la entrada al siguiente módulo. Una de las representaciones más usadas son los coeficientes Linear Predictive Coding (LPC) y los coeficientes Mel-Frecuency Cepstrum Coefficients (MFCC). En la etapa de reconocimiento se traduce la señal de entrada. Este proceso se puede llevar a cabo de diversas formas utilizando enfoques como Redes Neuronales Artificiales (RNA).

Se muestra la propuesta de investigación, se describe detalladamente las características, algoritmos, herramientas, etc., de la propuesta elaborada.

Luego se muestra los resultados obtenidos, conclusiones y recomendaciones, en base al reconocimiento realizado.

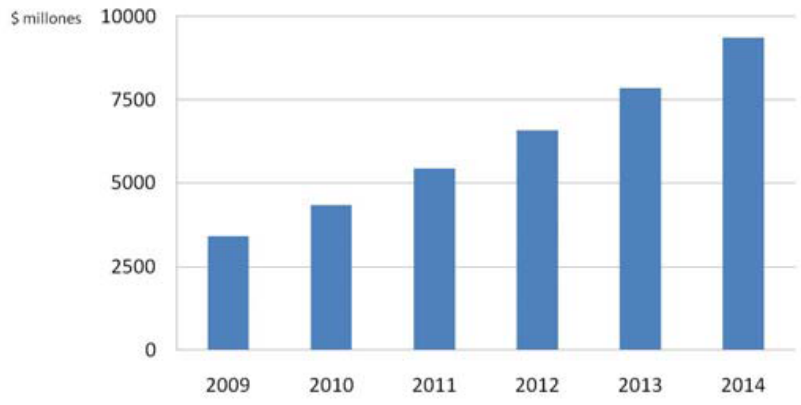
CAPITULO I: PROBLEMA DE INVESTIGACIÓN

1.1. Situación problemática.

En la actualidad, la tecnología biométrica ha permitido automatizar y mejorar los procesos de reconocimiento de voz. Se trata de un proceso similar al que habitualmente realiza el ser humano reconociendo e identificando a sus congéneres por su voz. (San-José, 2011)

UPM (2013), Según datos de Ahora Novetta Solutions (antes International Biometrics Group), empresa consultora y de investigación de mercado específica de este sector, se prevé una evolución del volumen del mercado biométrico global con tasas de crecimiento lineales durante los próximos años, como se muestra en la Gráfica 1 y 2.

Gráfica 1: Volumen y evolución del mercado biométrico global



Fuente: (Pérez, 2011)

Gráfica 2: Segmentación de mercado por tecnología biométrica (2013).

Fuente: (Pérez, 2011)

La demanda de soluciones biométricas ha cambiado en función de nuevas necesidades de seguridad y los avances tecnológicos. En la Gráfica 3 se muestra una estimación de la segmentación del mercado actual de la biometría según el tipo de tecnología utilizada.

Gráfica 3: Evolución del volumen de mercado por tecnologías biométricas

Mills $	2009	2010	2011	2012	2013	2014
Dactilar	2.280	2.869	3.556	4.218	4.947	5.792
Facial	390	510	675	848	1.097	1.417
Iris	174	288	361	480	578	730
Voz	104	109	113	136	167	189

Fuente: (Pérez, 2011)

Aguilar (2013), afirma que *"En los Smart TV el reconocimiento de voz funciona bien siempre que estuviéramos en un ambiente silencioso."*. Esto podría deberse a factores externos como el ruido, el tono de la voz, o a los métodos de reconocimiento de voz, ya que a la hora de utilizar los comandos de voz hay que repetir una y otra vez para que se reconozca el comando deseado.

MIT Technology Review (2012), afirma que “El reconocimiento de voz que ya incorporan los smartphones es quizá la parte más visible de esta tecnología. Cada vez se le está dando más uso en los móviles, sobre todo debido a que la interfaz táctil es muy útil para navegar en conjunto con los mandos de voz, pero probablemente la técnica para el reconocimiento de voz no sea efectiva al 100%”. Y precisamente para esto resulta bastante adecuado una evaluación de métodos para el reconocimiento de comando de voz.

En la investigación de Bejerano (2014), afirma que “El reconocimiento de voz tiene limitaciones en cuanto a la señal, debe estar lo mejor formada posible para evitar generar incertidumbre en la decisión tomada al final del proceso.”. Otro aspecto fundamental en la confiabilidad del sistema son los métodos y/o técnicas implementados para el procesamiento de la voz.

En cuanto a equipos sofisticados, esta investigación. Fajardo (2009), afirma que “A pesar de que existen equipos muy sofisticados que pueden reconocer la voz de un individuo, pueden ser falsificados con una simple grabación, cuando sea el caso de que una palabra fija es la que permita acceder a un lugar deseado.”

Pérez (2013) en su investigación *“Sistema de Seguridad Por Reconocimiento de Voz”* diseñó un sistema de seguridad para el hogar que funcione en base al reconocimiento de voz, utilizando algoritmo de Predicción Lineal y Análisis Cepstral. La solución para este problema fue colocar contadores que indicaran el número de coeficientes LPC que se parecieron a los coeficientes LPC grabados en la Base de datos de cada usuario y el número más cercano al máximo de Coeficientes LPC será el usuario identificado, para el software es más fácil comparar un número contra otro, que un vector de miles de números contra otro vector de miles de números, no se comprometió la integridad o seguridad del reconocimiento sólo se agilizó el procesamiento del software para que al usuario le pareciera más agradable y fluido.

La información de Bellesi (2009), nos dice en su investigación “*Reconocimiento de voz para aplicación en domótica*” consistió en el reconocimiento de una serie de palabras aisladas y de un número reducido de locutores utilizando Coeficientes de predicción lineal (LPC), Coeficientes cepstrales de frecuencia Mel, cuya conclusión descubrieron que no existe un método que identifique o reconozca en

el 100% de los casos o en cualquier situación. Finalmente, el problema nos deja la puerta abierta a futuras revisiones y mejoras de cara a proyectos un poco más ambiciosos.

Soto (2012) en su investigación “Algoritmo para el Reconocimiento de Comandos de Voz”, desarrolló un algoritmo computacional para el reconocimiento de comandos de voz, basado en la transformada Haar Wavelet que cada comando de voz sea una palabra corta y utilizado en un dispositivo personal, para un conjunto finito de comandos de voz, que opere en una plataforma de bajo costo y de prestaciones limitadas en tiempo real, obteniendo un resultado eficaz del 75% en reconocimiento por comando de voz.

El principal objetivo de la investigación es lograr extraer de una señal de entrada algunos parámetros fundamentales que caracterizan la voz humana| (como el tono, frecuencia, cadencia, duración del patrón vocal, entre otros) y así realizar una evaluación de métodos para lograr la mejor estimación, sin embargo, aún hay deficiencias es por ello que se han realizado investigaciones para superar esta limitación.

1.2. Formulación del problema.

¿Cómo evaluar métodos para el reconocimiento de comandos de voz para personas con dislexia?

1.3. Delimitación de la investigación.

El presente proyecto clasificará solamente con un formato de audio .wav mas no con otros formatos.

1.4. Justificación e importancia de la investigación.

Esta investigación está alineada: Ciencias de la Computación de Universidad Señor de Sipán.

Su importancia de esta investigación es a portar conocimiento científico en el campo de la Inteligencia Artificial, se evaluarán los algoritmos que sean más eficientes para el reconocimiento de la voz.

Los resultados obtenidos ayudaran a otros investigadores en soluciones a

problemas más específicos.

También tiene importancia a nivel social ya que obtuvimos micrófono de una laptop en zonas casi ruidosas para la grabación de voz de manera más rápida y fácil a persona que sufren con dislexia.

Es técnicamente viable ya que existen diversos algoritmos en el mundo las cuales se puede realizar investigación.

1.5. Limitaciones de la investigación.

La presente investigación de evaluación de métodos para el reconocimiento por comando de voz, estará limitado por los siguientes factores:

- Esta investigación se ve limitada al uso de voces a la base de datos dado la limitación de acceso a voces de dislexia humano de nuestra comunidad, de dicho número de voces, han sido retiradas aquellas que no poseen las condiciones adecuadas para su análisis, fueron desarrolladas en Visual Studio.

1.6. Objetivos de la investigación.

Objetivo general

Evaluar métodos para el reconocimiento de comandos de voz para personas con dislexia

Objetivos específicos

1. Generar una base de datos con audios de comando de voz de personas.
2. Seleccionar el método de procesamiento y reconocimiento de voz.
3. Implementar los métodos de reconocimiento de voz
4. Evaluar el resultado de reconocimiento de voz

CAPITULO II: MARCO TEÓRICO

2.1. Antecedentes de estudios

2.1.1. A nivel internacional:

Soto P. A. , Álvarez G., Olavarrieta S., y Cañete A. (2012) en su investigación titulada **"Algoritmo para el Reconocimiento de Comandos de Voz"**, el fin de este trabajo fué el reconocimiento de un conjunto finito de comandos de voz.

Esta trabajo de investigación tuvo como objetivo principal desarrollar un algoritmo computacional para el reconocimiento de comandos de voz, basado en la transformada Haar Wavelet que cada comando de voz sea una palabra corta y utilizado en un dispositivo personal, para un conjunto finito de comandos de voz, que opere en una plataforma de bajo costo y de prestaciones limitadas, en tiempo real y que permita un adecuado porcentaje de éxito en el reconocimiento del comando, obteniendo un resultado eficaz del 75% en reconocimiento por comando de voz.

En el trabajo de Oropeza (2010), en su investigación titulada **"Algoritmos y Métodos para el reconocimiento de voz en español mediante silabas"**, El objetivo de estudio de esta investigación fue realizar un análisis desde fonemas, hasta la palabra misma. Esto ha dado como origen una gran cantidad de resultados e implementación de algunas técnicas relacionadas, este trabajo fue enfocado al área de la sílaba y se analiza su alta sensibilidad al contexto, basado en la técnica de los Modelos Ocultos de Markov. Para finalizar, se procedió a intentar verificar el efecto que tiene considerar la acentuación de las palabras. El resultado generó un porcentaje promedio de reconocimiento 80.5%.

Camargo, García, y Gaona (2012) en su investigación titulada **"Reconocimiento de voz humana aplicado a la domótica"**. El objeto de estudio fue Controlar diversos elementos cotidianos ubicados en el hogar a través comandos de reconocimiento de voz para lo cual se Implementó un dispositivo de reconocimiento de voz humana mediante el uso de DSP (Procesador Digital de Señales) con Codificación por Predicción Lineal (LCP). Obteniendo como resultado que el algoritmo LPC utilizado tiene un

alto porcentaje de efectividad de un 90%.

2.2. Estado del arte

Reconocimiento de voz:

El reconocimiento de habla natural ha experimentado un intenso desarrollo gracias a los avances que han tenido lugar en el procesamiento de señal, algoritmos, arquitecturas y plataformas de computación. (Lumen Vox, s.f.)

Desde 1940, los laboratorios de AT&T y Bell se encargaron de desarrollar un dispositivo rudimentario para reconocer voz, fundamentándose en los principios de la fonética acústica, teniendo presente que el éxito de esta tecnología, dependería de su habilidad para percibir la información verbal compleja con alta precisión.

En la década de los 50, el sistema anterior conseguido, permitía identificación de dígitos mono-locutor, basada en medidas de resonancias espectrales del tracto vocal para cada dígito. Siguiendo esta línea, RCA Labs trabajó en el reconocimiento de 10 sílabas. Y es a finales de la década, cuando tanto la University College de Londres como el MIT Lincoln Lab, trataron de desarrollar un sistema de reconocimiento limitado de vocales y consonantes. Esta tarea parecía novedosa por el uso de información estadística y cuyo objetivo era una mejora del rendimiento en palabras de dos o más fonemas.

Fue por la década de los 60, cuando los sistemas electrónicos utilizados hasta el momento, sirvieron de pasarela a los sistemas con hardware específico, en los NEC Labs de Japón. En esta etapa, cabe destacar tres proyectos notables en la investigación de esta disciplina:

- RCA Labs tenían como objetivo un desarrollo de soluciones realistas para los problemas en la falta de uniformidad de las escalas de tiempo en el habla. Para ello, diseñaron un conjunto de métodos de normalización en el dominio temporal, detectando fiablemente el inicio y fin de discurso.
- En la Unión Soviética, T. K. Vintsyuk, propone el empleo de métodos de programación dinámica para conseguir el alineamiento temporal

de parejas de realizaciones. Surge de aquí la técnica *DTW (Dynamic Time Warping).*

- Por último, en el campo del reconocimiento de habla continua, D. R. Reddy de la Universidad de Stanford, desarrolla el seguimiento dinámico de fonemas, concluyendo su trabajo en un reconocedor de oraciones de amplio vocabulario.

Allá por los años 70, se originan críticas acerca de la viabilidad y utilidad del reconocimiento automático de habla. A pesar de esto, dicha disciplina se adentra en el mundo probabilístico, donde los principales campos de estudio son los siguientes: el reconocimiento de palabras aisladas estuvo fundamentado en el procedimiento de ajuste de patrones, programación dinámica, y más adelante, técnicas LPC (Linear Predictive Coding). Ésta última se empleó exitosamente en la codificación y compresión de la voz, a través del uso de medidas de distancias sobre el conjunto de parámetros LPC.

Los primeros intentos de reconocedores de habla continua y grandes vocabularios los llevaron a cabo IBM, con el dictado automático de voz, ARPA Speech Understanding Research, y la Universidad de Carnegie Mellon, con el exitoso sistema Hearsay I. Finalmente, en los AT&T Labs, se investigó en la dirección de los reconocedores independientes del locutor para aplicaciones telefónicas, finalizando este periodo con la realización de sistemas ASR (Automatic Speech Recognition), favorecida por tarjetas microprocesador.

La década de los 80 se inicia con una base muy asentada en la construcción de sistemas de reconocimiento, a diferencia de los anteriores que sólo reconocía vocablos aislados, ahora tienen la capacidad de tratar con palabras encadenadas fluidamente. Uno de los avances más importante es el paso de métodos basados en comparación de plantillas a otros basados en modelos estadísticos, extendiéndose el uso de los Modelos Ocultos de Markov o HMMs. Éstos experimentaron numerosas mejoras y se situaron como los mejores modelos que capturaban y modelaban la variabilidad del habla.

Las redes neuronales empezaron a tomar peso en este ámbito, y gracias al desarrollo de algoritmos de aprendizaje más eficaces, aparecieron modelos como el perceptrón.

Además, se llevan a cabo una serie de avances:

- El diseño de unidades de decodificación fonética a partir de la experiencia de fonetistas en tareas de interpretación de espectrogramas.
- La grabación de grandes bases de datos como TIMIT, que permite la comparación de resultados entre diferentes grupos de trabajo.
- El programa DARPA (Defence Advance Research Agency) contribuyó en Estados Unidos, al impulso del desarrollo de sistemas de reconocimiento para habla continua y vocabularios de gran tamaño con independencia del locutor.
- El desarrollo por parte de la CMU de su sistema SPHINX.

En los años 90, continuando con los objetivos ya propuestos anteriormente, se amplían los tamaños de vocabularios y se diversifican los campos de aplicación. Teniendo gran importancia su aplicación sobre línea telefónica, así como los resultados de este reconocimiento en entornos con condiciones adversas y ruido.

A partir del año 2000, La integración del Reconocimiento de voz en diferentes Sistemas Operativos es una realidad, se da la Integración de aplicaciones por teléfono y sitios de Internet dedicados a la gestión de reconocimiento de voz (Voice Web Browsers). También aparece el estándar VoiceXML.

En el 2011 con el Lanzamiento de Siri es cuando de verdad se le ha empezado a prestar atención al reconocimiento de voz. Y es que esta tecnología lleva funcionando desde hace varias décadas.

Cada vez se le está dando más uso en los móviles, sobre todo debido a que la interfaz táctil es muy útil para navegar, pero no lo es tanto para introducir texto. Y precisamente para esta tarea resulta bastante adecuado el reconocimiento de voz.

En el 2012 Microsoft crea un traductor de voz al estilo de Star Trek, el sistema

de Microsoft es una demostración de la más reciente tecnología de reconocimiento de voz de la empresa, basada en un software de aprendizaje inspirado en el modo de funcionamiento de las redes neuronales.

Los teléfonos inteligentes también tienen un gran ancho de banda para las conexiones de datos con la nube, donde los servidores pueden hacer todo el intenso trabajo que precisa el reconocimiento de voz y la comprensión de las consultas orales.

En el 2013 AT&T intenta mantener con vida su servicio de voz. Durante los últimos años, Skype y otros servicios de telefonía han comenzado a desplazar al servicio de voz de AT&T y la compañía tiene la esperanza de poder revertir esta tendencia. La empresa se está dirigiendo a programadores externos para que creen aplicaciones de voz y de mensajería que utilicen los números de teléfono móvil tradicionales de sus clientes.

En el 2014 aparece Cortana, un ayudante virtual por voz de Microsoft, Cortana empieza a poder controlar productos domésticos inteligentes como las luces y los termostatos con el uso de controles de voz.

La empresa de domótica Insteon, con sede en Irvine, California (EEUU), está trabajando en una aplicación para Windows Phone 8.1 que se lanzará este año y cuyo objetivo es hacer más fáciles cosas como encender las luces o modificar la temperatura de la casa dando órdenes a través de Cortana del estilo de "Insteon, apaga todas las luces", o "Insteon, baja la temperatura del termostato del salón".

Los avances producidos en el ámbito de las tecnologías del habla cada día son más significativos. En el campo del reconocimiento automático de voz, los reconocedores actuales manejan cada vez vocabularios más grandes y reducen las tasas de error, gracias al uso de algoritmos más eficientes, al uso de equipos más potentes y al aumento de complejidad de estos sistemas, con modelados más sofisticados. El amplio grado de aplicación en función de los usuarios y los distintos entornos, hacen que no haya un sistema de reconocimiento de voz universal y sea necesaria su adaptación a las condiciones de funcionamiento y al tipo de aplicación que se requiera.

2.3. Sistemas teórico conceptuales.

2.3.1. Sistemas de reconocimiento automático del habla (RAH).

Son sistemas que implementan en su estructura los algoritmos necesarios para el procesamiento y análisis de diferentes características de la voz humana con el fin de obtener un estimado de las palabras que se han dicho previamente. El reconocimiento de voz por computadora es una tarea compleja de reconocimiento de patrones y de los sistemas biométricos.

Gráfica 4: Proceso de reconocimiento de voz.

Fuente: Elaboración Propia.

2.3.2. Procesamiento de Señales.

El principal objetivo es comprimir los datos correspondientes a la señal de voz, eliminando información no pertinente al análisis fonético de la información y extraer esas características de la señal de voz que contribuyen significativamente con la detección de las diferencias fonéticas, y así realizar métodos comparativos para lograr la mejor estimación.

En esta fase se utilizan algoritmos están encargados de extraer aquellas características biométricas presentes en la señal original y que, debido a sus propiedades invariantes al tiempo, a la forma de presentación, al tipo de sensor, al método de compresión y al sistema de transmisión, se consideran como las más relevantes a efectos de comparación entre diferentes muestras.

Esta información extraída se utiliza para generar un "vector de características" que será la unidad de información utilizada para posteriores comparaciones y para el subsistema de toma de decisión.

2.3.3. Algoritmos de Procesamiento de señales.

Cada algoritmo cumple la función de extraer los parámetros fundamentales de la voz. Naturalmente algunos tienen más eficiencia que otros, pero de igual manera consumen más recurso computacional que otro.

2.3.3.1. Predicción Lineal.

La Predicción Lineal (LP) en el reconocimiento de voz, consiste en modelar el tracto vocal como un filtro digital constituido únicamente por polos (respuesta infinita al impulso o IIR), permitiendo así calcular la próxima muestra como una suma ponderada de las muestras pasadas. Este filtro de predicción se traduce en la función de transferencia de la ecuación:

$$H(z) = \frac{G}{1 - \sum_{i=1}^{P} a_i \ z^{-i}}$$

Donde G es la ganancia del filtro que depende de la naturaleza de la señal (sonora o no sonora). Entonces, dada la señal s(n), el problema consistirá en determinar los coeficientes de predicción y la ganancia.

Entonces, serán los coeficientes de predicción los que se usarán como parámetros de reconocimiento de palabras. Se han realizado varias modificaciones a este algoritmo para hacer más eficiente el reconocimiento de voz.

2.3.3.2. Cepstrum.

El Cepstrum es una herramienta muy utilizada para la representación paramétrica de las señales de voz y se define como la Transformada de Fourier del espectro logarítmico de la señal, por lo que existe un Cepstrum complejo y un Cepstrum real dependiendo de si la función logarítmica está definida para valores reales o complejos. La diferencia entre uno y otro radica en el hecho de que el Cepstrum complejo permite reconstruir la señal y el real no, ya que se pierde la información correspondiente a la fase.

Para entender las bases matemáticas del cálculo del Cepstrum es necesario considerar una secuencia estable x[n], cuya Transformada Z se puede expresar en coordenadas polares, según la ecuación:

$$X(z) = |X(z)| \cdot e^{j \measuredangle X(z)}$$

Donde |X(z)| y $\measuredangle$ X(z) representan la magnitud y el ángulo respectivamente de la Transformada Z de x[n].

Como la señal x[n] es estable, la región de convergencia para X (z) incluye el círculo unitario y la Transformada de Fourier de x[n] existe y es igual a X (ejω). El Cepstrum complejo correspondiente a x[n] se define como una secuencia estable

$\hat{X}$ [n], cuya Transformada Z se puede define:

$$\hat{X}(z) = \log\left[X(z)\right]$$

Como se requiere que $\hat{X}$[n] sea estable, la región de convergencia incluye el círculo unitario, por lo que el Cepstrum complejo se puede representar usando la Transformada Inversa de Fourier, tal y como se observa:

$$\hat{x}[n] = \frac{1}{2\pi} \cdot \int_{-\pi}^{\pi} \log\left[X\left(e^{j\omega}\right)\right] \cdot e^{j\omega n} \cdot d\omega$$

En contraste con el Cepstrum complejo, el Cx [n] o Cepstrum real de una señal, es definido como la Transformada Inversa de Fourier del logaritmo de la magnitud de la Transformada de Fourier, tal y como se muestra:

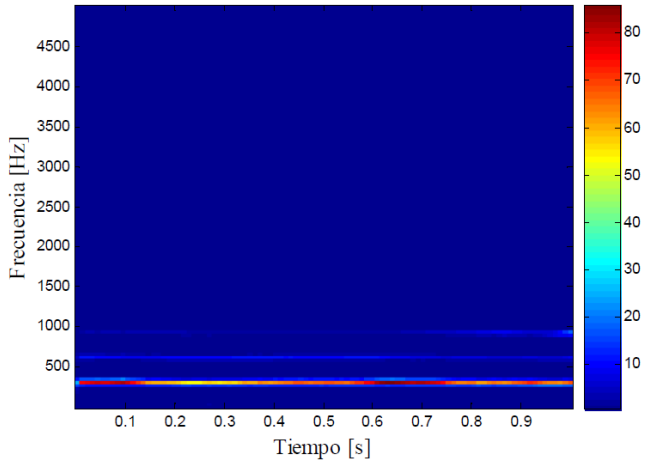
$$c_x[n] = \frac{1}{2\pi} \cdot \int_{-\pi}^{\pi} \log\left|X\left(e^{j\omega}\right)\right| \cdot e^{j\omega n} \cdot d\omega$$

El Cepstrum real se utiliza en muchas aplicaciones y como no depende de la fase de X (ejω) es mucho más fácil de calcular que el Cepstrum complejo, aunque x[n] no puede ser recuperada a partir de cx [n] (Oppenheim y Schafer (como se citó en Salcedo y Teixeira, 2009)).

El análisis cepstral es comúnmente utilizado para obtener información de la señal de voz que permita parametrizarla para luego ser usada en la fase de reconocimiento. Mediante el Cepstrum se puede separar la señal de excitación del sistema que modela la producción de la voz y la función de transferencia que modela el tracto vocal. Por esta razón es que al analizar el

espectrograma mostrado en la Gráfica 6 se pueden observar componentes frecuenciales que no se aprecian Gráfica 5.

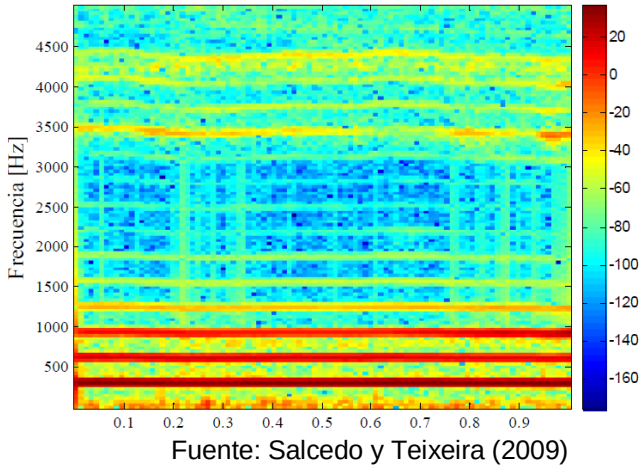
Gráfica 5: Espectrograma: Señal de voz original



Fuente: Salcedo y Teixeira (2009)

Se puede decir que esta transformada detecta las periodicidades que se observan en un espectro logarítmico como el de la Gráfica 6.

Gráfica 6: Espectrograma: Señal de voz con Cepstrum



Fuente: Salcedo y Teixeira (2009)

Dichas periodicidades, para las señales sonoras, son de dos tipos: unas rápidas, debidas a la estructura armónica del espectro, que se repiten en los múltiplos de la frecuencia fundamental Fo, y unas fluctuaciones mucho más lentas, no periódicas, que proporcionan la envolvente espectral. Estas fluctuaciones lentas se manifiestan en la parte baja del Cepstrum y

caracterizan la forma del tracto vocal. De hecho, también se reducen a unos pocos datos numéricos (del orden de 15 o 20), que parametrizan la información articulatoria. La información correspondiente a la fuente excitadora se sitúa, por el contrario, en la parte alta del Cepstrum, y corresponde básicamente a la periodicidad de Fo (detección de periodicidad y período correspondiente) y a la energía global de la señal. Esta separación de las dos informaciones permite un proceso de desconvolución de la señal de voz, para recuperar separadamente la excitación y la respuesta impulsiva del tracto vocal, en caso de que la aplicación así lo requiera (Martí, 1998).

Las representaciones paramétricas más utilizadas para trabajar con señales de voz y que están basadas en el análisis cepstral de la misma, pueden ser divididas en dos grupos: aquellas basadas en la predicción lineal del espectro y aquellas basadas en el espectro de Fourier (Davis & Mermelstein (como se citó en Salcedo y Teixeira, 2009)).

2.3.3.3. Predicción Lineal de los Coeficientes Cepstrales (LPCC).

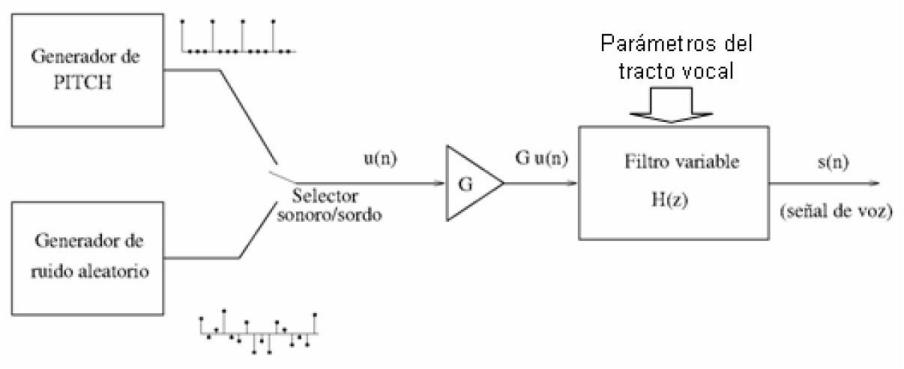
La técnica LPCC permite estimar los coeficientes Cepstrales mediante el uso del algoritmo LPC, el cual establece un modelo que permite calcular la próxima muestra de la señal mediante la función de transferencia que se define mediante la siguiente ecuación:

$$H(z) = \frac{G}{1 - \sum_{k=1}^{p} a_k \cdot z^{-k}}$$

Donde G es la ganancia del filtro, que depende de la naturaleza de la señal y ak son los coeficientes del filtro que modela el tracto vocal (San Martín (como se citó en Salcedo y Teixeira, 2009)).

El análisis anterior se basa en el modelo de producción de la voz presentado en la Gráfica 7.

Gráfica 7: Modelo simplificado de producción de la voz



Fuente: Salcedo y Teixeira (2009)

Su idea fundamental es que la voz puede modelarse a través de una combinación lineal de p muestras anteriores más una señal de excitación (periódica o ruido blanco dependiendo de la naturaleza de la señal), tal y como lo demuestra la siguiente ecuación (GTAS, n.d):

$$s[n] = \sum_{k=1}^{p} a_k \cdot s[n-k] + G \cdot u[n]$$

Donde u[n] es la entrada del filtro que modela el tracto vocal, por lo que, dada la señal s[n], el problema consiste en determinar los coeficientes de predicción ak y la ganancia G del filtro.

Los coeficientes de predicción se usan en el proceso de parametrización para calcular los coeficientes cepstrales. Por lo tanto, dada una señal s[n] (considerada estacionaria en el período de evaluación) un predictor de orden p se define según la ecuación:

$$\tilde{s}[n] = -\sum_{k=1}^{p} a_k \cdot s[n-k]$$

La determinación de los coeficientes de predicción k a se realizará minimizando el error de predicción de orden p que se comete cuando se intenta realizar la aproximación de la señal y cuya representación se puede obtener mediante la ecuación:

$$e[n] = s[n] - \tilde{s}[n] = s[n] + \sum_{k=1}^{p} a_k \cdot s[n-k]$$

Donde e[n] es la señal de error y s[n] la señal de voz.

Los coeficientes de predicción se calculan minimizando la media del error cuadrático medio con respecto a cada uno de los coeficientes. El error cuadrático $\tilde{e}$ total se define en la ecuación:

$$\tilde{e} = E\left\{e^2[n]\right\}$$

Donde E{.} es el operador valor esperado

Para obtener el mínimo error de predicción se calcula la derivada de e[n] con respecto a los coeficientes ak y se obtiene la ecuación:

$$\sum_{k=1}^{p} a_k \cdot E\left\{s[n-k]\cdot s[n-i]\right\} = -E\left\{s[n]\cdot s[n-i]\right\}, \quad 1 < i < p$$

El error de predicción mínimo viene definido por la ecuación:

$$\tilde{e}_{min} = E\left\{y^2[n]\right\} + \sum_{k=1}^{p} a_k \cdot E\left\{y[n]\cdot y[n-k]\right\}$$

Según lo anterior, el error medio cuadrático mínimo se puede escribir en función de la autocorrelación según la ecuación:

$$\tilde{e}_{min} = R(0) + \sum_{k=1}^{p} a_k \cdot R(k)$$

Estas ecuaciones son llamadas ecuaciones normales o de Yule-Walker y los coeficientes ak del predictor óptimo se obtienen resolviendo las p ecuaciones con p incógnitas (Stefanelli (como se citó en Salcedo y Teixeira, 2009)).

Como resultado de todo este análisis se puede observar que mediante el uso del LPC como técnica de predicción lineal, de todos los posibles juegos de parámetros que se pueden obtener a partir del modelado de la voz como un proceso auto-regresivo, los coeficientes ak que modelan el tracto vocal, pueden ser utilizados para el cálculo de los coeficientes cepstrales o LPCC's mediante la ecuación:

$$c_i = a_i + \sum_{k=1}^{i-1} \left(\frac{k-i}{i}\right) \cdot c_{i-k} \cdot a_k$$

De esta manera se puede observar cómo la predicción lineal de los coeficientes del filtro que modela el tracto vocal puede ser utilizada para

estimar por ejemplo la densidad espectral de potencia de la señal de voz o como en este caso, para estimar los coeficientes cepstrales utilizados para el reconocimiento del habla.

2.3.3.4. Análisis de Fourier

Para esta técnica de parametrización, las características espectrales de la señal de voz se derivan del análisis de Fourier de tiempo corto, definido por la ecuación:

$$S_x(t,f) = \int_{-\infty}^{\infty} x(\tau + t) \cdot w(\tau) \cdot e^{-j2\pi f\tau} \cdot d\tau$$

Donde x (τ) es la señal de voz y w (τ) representa la función de la ventana de análisis (por ejemplo, la ventana Hamming). De esta forma se realiza un análisis localizado de la señal mediante la aplicación de una ventana w (τ) a la señal, alrededor del instante de tiempo "t", analizada a todas las frecuencias consideradas "f". (Mahanad (como se citó en Salcedo y Teixeira, 2009)).

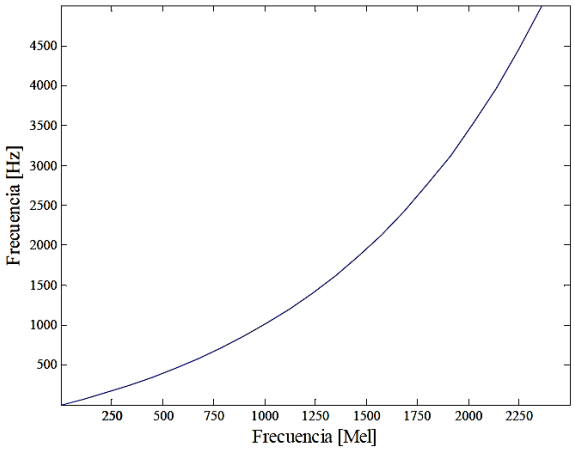
2.3.3.5. Coeficientes Cepstrales de Frecuencia Mel (MFCC).

Una familia de coeficientes directamente relacionada con los LPCC son los llamados mel-cepstrum o MFCC (Mel-Frequency Cepstrum Coefficients), los cuales son de gran utilidad en la extracción de los parámetros de la señal de voz, ya que están basados en la variación conocida de los anchos de banda de las frecuencias críticas del oído. Los filtros que se le aplican a la señal en la técnica MFCC están espaciados linealmente para frecuencias menores a 1000 Hz y logarítmicamente para frecuencias mayores de 1000 Hz, con el fin de capturar las características fonéticamente importantes del habla (Do, M). A esta escala se le denomina "Escala Mel" y su fórmula matemática se describe en la ecuación:

$$Mel(f) = 2595 \cdot \log\left(1 + \frac{f}{700}\right)$$

En la Gráfica 8 se muestra la correspondencia que existe entre la frecuencia en Hz y la escala de frecuencia mel.

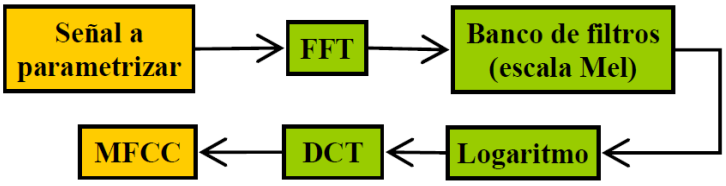
Gráfica 8: Correspondencia entre la frecuencia en Hz y la frecuencia Mel



Fuente: Salcedo y Teixeira (2009)

Los pasos necesarios para el cálculo de los MFCC's se en la Gráfica 9.

Gráfica 9: Diagrama de bloques para el cálculo de los MFCC's



Fuente: Salcedo y Teixeira (2009)

Cada una de las fases mostradas en el diagrama de la Gráfica 5, se describen a continuación:

1) Se calcula la Transformada de Fourier de Tiempo Corto X(n,ωk) a cada una de las tramas obtenidas de la etapa de pre-procesamiento mediante la ecuación:

$$X(n,\omega_k) = \sum_{m=-\infty}^{\infty} x(m)\cdot w(n-m)\cdot e^{-j\omega_k m}$$

Donde:

$$\omega_k = \frac{2\pi}{N}\cdot k$$

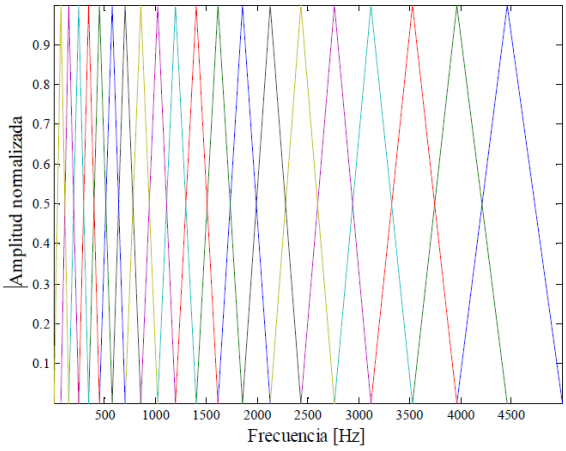
2) El cuadrado de la magnitud de X(n,ωk) es ponderado por una serie de filtros distribuidos sobre la escala Mel para luego calcular la llamada "log-

energía” del filtro l-ésimo mediante la ecuación:

$$E_{Mel}(n,l) = \frac{1}{A_l} \cdot \sum_{k=L_l}^{U_l} \left| V_L(\omega_K) \cdot X(n,\omega_K) \right|^2$$

Donde Ll y Ul son las frecuencias de corte inferior y superior del filtro l-ésimo. El banco de filtros linealmente espaciado en la escala Mel tiene la forma que se muestra en la Gráfica 10 y los filtros que lo conforman pueden ser triangulares o tener otras formas, tales como Hamming, Hanning o Kaizer, pero el triangular es el más utilizado.

Gráfica 10: Banco de filtros espaciados linealmente en la escala de frecuencia Mel



Fuente: Salcedo y Teixeira (2009)

3) Finalmente se convierte el espectro logarítmico Mel nuevamente al dominio del tiempo usando la Transformada Discreta de Coseno, dado que los coeficientes cepstrales son números reales. El cálculo de estos coeficientes se realiza mediante la ecuación:

$$C_{Mel}[n,m] = \frac{1}{R} \cdot \sum_{l=1}^{R} \log\left\{E_{Mel}(n,l)\right\} \cdot \cos\left[n\left(l-\frac{1}{2}\right)\cdot\frac{\pi}{l}\right]$$

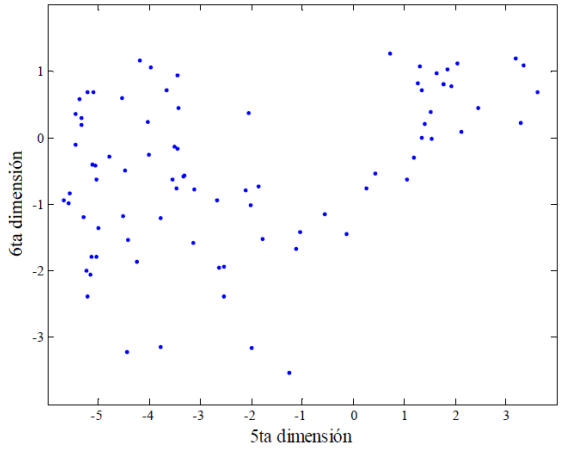
$$n = 1,2,3....K$$

Donde K es el número de coeficientes cepstrales, que por lo general se escoge entre 10 y 20.

Mediante el proceso descrito anteriormente, para cada trama de voz de duración aproximada igual a 30 ms con solapamiento, se calcula un conjunto de coeficientes cepstrales. Este es el resultado de la Transformada Discreta de Coseno de la Densidad Espectral de Potencia expresada en la escala mel. A este conjunto de coeficientes se le denomina Vector Acústico, por lo que cada entrada es transformada mediante este proceso en una secuencia de Vectores acústicos que representan las características más importantes de la voz, necesarias para el proceso de reconocimiento del habla.

Un ejemplo del vector acústico generado por el algoritmo que calcula los MFCC se muestra en la Gráfica 11, para el cual se utilizó un banco de 20 filtros

Gráfica 11: Vector acústico generado mediante el cálculo de los MFCC's

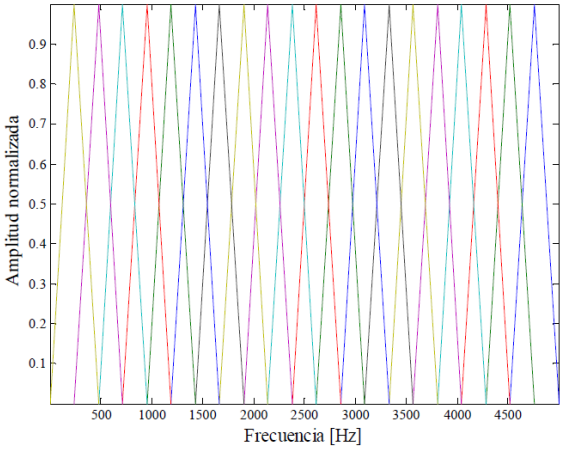


Fuente: Salcedo y Teixeira (2009)

2.3.3.6. Coeficientes Cepstrales de Frecuencia Lineal (LFCC).

El esquema bajo el cual funciona esta técnica es similar al mostrado en la Gráfica 5, la única diferencia es que los filtros que conforman el banco de filtros que se le aplica a la señal, se encuentran sobre una escala lineal, es decir: f = fMel. El banco de filtros que se genera con esta técnica se muestra en la Gráfica 12.

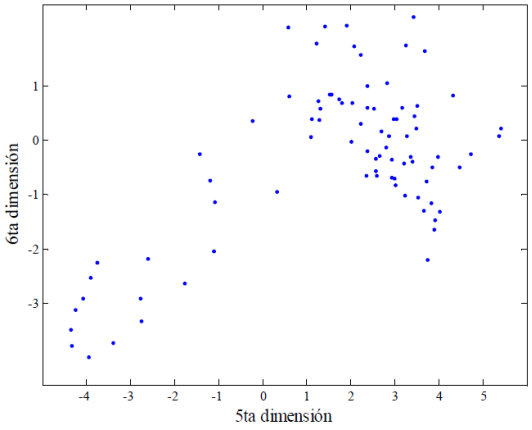
Gráfica 12: Banco de filtros generado para el cálculo de los LFCC's



Fuente: Salcedo y Teixeira (2009)

Por su parte, el Vector acústico generado por el algoritmo que calcula los coeficientes ceptrales mediante este método muestra similitudes al derivado del cálculo de los MFCC's y se observa en la Gráfica 13.

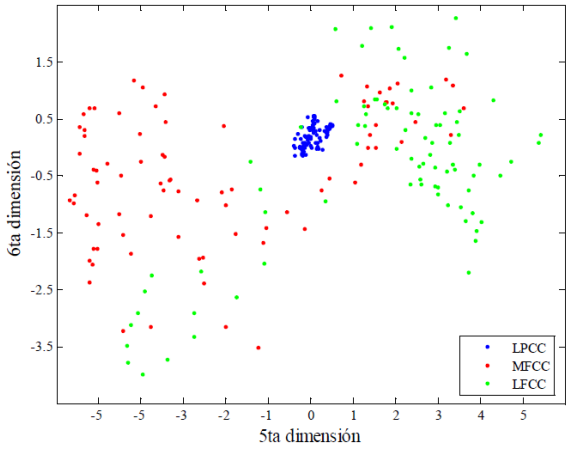
Gráfica 13: Vector acústico generado por el cálculo de los LFCC's



Fuente: Salcedo y Teixeira (2009)

A continuación, se muestra en la Gráfica 14 los vectores acústicos generados mediante el cálculo de los MFCC's, LPCC's y LFCC's.

Gráfica 14: Vectores acústicos generados mediante el cálculo de los MFCC's, LPCC's y LFCC's



Fuente: Salcedo y Teixeira (2009)

Se puede observar en la Gráfica anterior que los vectores acústicos correspondientes a los MFCC's y los LFCC's presentan diferencias en cuanto a su ubicación, pero el vector acústico de los LPCC's se destaca, ya que se encuentra restringido a los valores entre 0 y 1 por características propias del algoritmo. Cabe destacar que la diferencia entre ellos se debe principalmente a que cada uno se utiliza para la extracción de distintas características de la voz

2.3.3.7. Transformada Wavelet.

El propósito de la Transformada Wavelet (TW) es la descomposición de una señal x(t) en una combinación lineal de versiones dilatadas y desplazadas de la función madre Ψ(t), lo cual se denota a través de:

$$X(\tau,a)=\frac{1}{\sqrt{a}}\int x(t)\Psi^{\bullet}{}_{\tau,a}(t)dt$$

$$\Psi^{\bullet}{}_{\tau,a}(t)=\Psi^{\bullet}\left(\frac{t-\tau}{a}\right)$$

Donde corresponde al desplazamiento de la Wavelet madre y a es la respectiva escala. Entre los conjuntos de Wavelets más usados están la

Haar, Morlet, Daubechies y Coifman.

2.3.4. Reconocimiento de voz.

Esta parte de reconocimiento de voz, compara una señal de entrada con el conocimiento que tiene de otras señales previamente analizadas, teniendo así un clasificador o identificador de señales, el cual es capaz de mostrar la similitud que existe entre dicha entrada y cada una de las señales con las que cuenta el sistema.

2.3.5. Algoritmos de Reconocimiento de voz.

2.3.5.1. Modelos ocultos de Márkov.

Un modelo oculto de Márkov o HMM es un modelo estadístico en el que se asume que el sistema a modelar es un proceso de Márkov de parámetros desconocidos. El objetivo es determinar los parámetros desconocidos (u *ocultos*, de ahí el nombre) de dicha cadena a partir de los parámetros observables. Los parámetros extraídos se pueden emplear para llevar a cabo sucesivos análisis, por ejemplo, en aplicaciones de reconocimiento de patrones.

2.3.5.2. Alineamiento temporal dinámico (DTW).

Algoritmo para medir la similitud entre dos secuencias que pueden variar en el tiempo o la velocidad.

Las semejanzas en los patrones de voz serían detectadas cuando una persona hable lento, rápido, o incluso si hay aceleraciones y deceleraciones en el transcurso. Este algoritmo nos permite encontrar la coincidencia óptima entre dos secuencias dadas.

2.3.5.3. Redes Neuronales (NN).

Las redes neuronales son modelos de cálculo que intentan emular el comportamiento del cerebro humano mediante una topología que se asemeja a la interconexión de las células nerviosas. Se construyen a partir de unidades sencillas interconectadas que cooperan para obtener la función de transferencia global de la red, cuya forma funcional estará determinada por su arquitectura y su dinámica. Las redes neuronales pueden ser consideradas como modelos no paramétricos para aproximación de

funciones generales. Son capaces de modelar sistemas no-lineales y pueden ser usadas en multitud de tareas, tales como clasificación, memoria asociativa o agrupamiento de datos, ya que se han para generalizar de forma adecuada a partir de pocos datos de entrenamiento. Esta visión artificial, el control de procesos o la diagnosis médica. De este modo, y posibilidades de desarrollo futuras, están han sido empleadas de forma significativa en el reconocimiento automático de locutores.

Existen tres arquitecturas o modelos básicos sobre los cuales se han centrado la mayor parte de los trabajos, se ha obtenido con los tres esquemas buenos resultados en tareas de relativa complejidad. Estos son el perceptron multicapa (MLP, Multi-Layer Perceptron), las funciones de base radial (RBF, Radial Basis Functions) y LVQ (Learning Vector Quantization).

2.3.5.4. Cuantificación Vectorial aplicada (VQ).

La idea básica de la Cuantificación Vectorial (VQ) es la de sustituir un cierto vector de parámetros, obtenido del análisis de un cierto segmento de señal, por un vector similar, llamado vector código perteneciente a un diccionario finito y prefijado de vectores. Cada vector código tiene asociado un cierto índice que se convierte en la salida del cuantificador.

2.3.5.5. K-Vecinos más cercanos (KNN).

Es un método de clasificación supervisada (Aprendizaje, estimación basada en un conjunto de entrenamiento y prototipos) que sirve para estimar la función de densidad.

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^{p} (x_{ri} - x_{rj})^2}$$

La fase de entrenamiento del algoritmo consiste en almacenar los vectores característicos y las etiquetas de las clases de los ejemplos de entrenamiento.

2.3.5.6. Modelos de Mezclas de Gaussianas (GMM).

Un GMM está compuesto, básicamente, de una superposición de M

funciones de densidad de probabilidad (fdp) gaussianas, donde cada fdp está ponderada por un coeficiente de peso.

Por cada clase se estiman los parámetros de los GMM que incluyen los coeficientes de ponderación y las medias y matrices de covarianza de cada fdp gaussiana.

2.4. Definición de la terminología.

Algoritmos: Son métodos usados para la extracción de características (procesamiento) y su comparación.

Alineamiento temporal dinámico; Es una técnica surgida de la problemática inherente a diferentes realizaciones de una misma locución, en las que se observa una variabilidad interna en la duración de los grupos fónicos que la forman, de modo que no existe una sincronización temporal (alineamiento temporal).

La dislexia: Es una condición cerebral que dificulta la lectura, la ortografía, la escritura y, algunas veces, el habla. Al cerebro de las personas que tienen dislexia le cuesta reconocer o procesar ciertos tipos de información. Esto puede incluir hacer coincidir el sonido de una letra con su símbolo (tal como la letra b haciendo el sonido be), y luego combinarlos para formar una palabra. Algunas personas con dislexia no tienen problemas pronunciando o "decodificando" las palabras, pero podrían batallar para entender lo que leen. Podría llegar a ser muy difícil para las personas con dislexia leer de manera automática o, aparentemente, sin esfuerzo.

Modelos ocultos de Markov: Es un modelo estadístico en el que se asume que el sistema a modelar es un proceso de Markov de parámetros desconocidos.

Procesador Digital de Señales: Es un sistema basado en un procesador o microprocesador que posee un conjunto de instrucciones, un hardware y un software optimizados para aplicaciones que requieran operaciones numéricas a muy alta velocidad. Debido a esto es especialmente útil para el procesado y representación de señales analógicas en tiempo real: en un

sistema que trabaje de esta forma (tiempo real) se reciben muestras, normalmente provenientes de un conversor analógico/digital (ADC).

Con estas aplicaciones se puede eliminar el eco en las líneas de comunicaciones, el reconocimiento de voz, los reproductores digitales de audio, y otros.

Procesamiento: Aplicación sistemática de una serie de operaciones sobre un conjunto de datos.

Procesamiento de señales: Es el procesamiento, amplificación e interpretación de señales.

Procesamiento de Señales de Voz: Para analizar señales de voz humana

Reconocimiento de Voz: Es una aplicación derivada del reconocimiento, la cual es un área de estudio cubierta por la Inteligencia Artificial.

Redes neuronales: Son un paradigma de aprendizaje y procesamiento automático inspirado en la forma en que funciona el sistema nervioso de los animales.

Transformada de Fourier: Es una transformación matemática empleada para transformar señales entre el dominio del tiempo (o espacial) y el dominio de la frecuencia, que tiene muchas aplicaciones en la física y la ingeniería.

Transformada de Wavelet: Es un tipo especial de transformada matemática que representa una señal en términos de versiones trasladadas y dilatadas de una onda finita (denominada ondula madre).

Voz: Sonido que el aire expelido de los pulmones produce al salir de la laringe, haciendo que vibren las cuerdas vocales.

CAPITULO III: MARCO METODOLÓGICO

3.1. Tipo y diseño de la investigación.

3.1.1. Tipo de investigación.

La presente investigación es de tipo cuantitativo, porque se establecieron indicadores que brindaron información para medir los resultados de tipo numéricos, ya que de esta forma se pudo emitir estadísticas de porcentaje.

3.1.2. Diseño de investigación

El Diseño de investigación es Cuasi-experimental: Porque dentro del desarrollo del proyecto existe variables que no se puede tener un control absoluto, pero se pretende tener un mayor control posible.

3.2. Población y Muestra.

3.2.1. Población

La población está determinada por 7 métodos de procesamiento de señales y 6 algoritmos para el reconocimiento de voz tal como se puede visualizar la siguiente tabla.

Tabla 11. Población (Algoritmos investigados)

Procesamiento de Señales	Reconocimiento de Voz
• Predicción Lineal • Cepstrum • Predicción Lineal de Coeficientes Cepstrales (LPCC). • Análisis de Fourier. • Coeficientes Cepstrales de Frecuencia de Mel (MFCC). • Coeficientes Cepstrales de Frecuencia Lineal (LFCC) • Transformada de Wavelet.	• Modelos ocultos de Markov (HMM) • Alineamiento temporal dinámico (DTW) • Redes Neuronales (NN) • Cuantificación Vectorial aplicada (VQ). • Modelos de Mezclas de Gaussianas (GMM). • K-vecinos más cercanos (KNN)

Fuente: Elaboración Propia.

3.2.2. Muestra:

Por conveniencia en la presente investigación se ha utilizado para el procesamiento de señales 2 métodos: Coeficientes Predictiva Lineal y Coeficientes Ceptrales de Frecuencia de Mel (LPC, MFCC) y para el reconocimiento de voz se utilizó el algoritmo de Redes Neuronales.

3.3. Hipótesis

Utilizando el algoritmo de Redes Neuronales, se logrará una mejor precisión en el reconocimiento de voz de las personas con dislexia

3.4. Variables

3.4.1. Variable independiente

Método de procesamiento de audio.

3.4.2. Variable dependiente

Reconocimiento de comandos de voz

3.5. Operacionalización

Tabla 12: Variable de estudio.

Variable de Estudio	Dimensiones	Indicadores	**Técnicas e instrumentos de recolección de datos/Unid de medida**
Método de reconocimiento de voz	Procesamiento de señales	Tiempo	Análisis documental
		Memoria	
		Procesador	
Reconocimiento de comandos de voz	Reconocimiento de voz	Exactitud (E) $E = \frac{RP + RN}{PT}$ RP=reales positivos RN=reales negativos PT=predicciones totales	Observación
		Precisión (P) $P = \frac{RP}{RP + FP}$ RP=reales positivos FP=falsos positivos	
		Sensibilidad $S = \frac{RP}{RP + FN}$ RP=reales positivos FN=Falsos negativos	

Fuente: Elaboración Propia.

3.6. Abordaje metodológico, técnicas e instrumentos de recolección de datos

Método Analítico Deductivo: Analizar la eficacia de diferentes algoritmos.

Los métodos y procedimientos para la recolección de datos o información,

consistirán en el Análisis Deductivo.

3.6.1. Abordaje metodológico.

El método para la recolección de datos que se utiliza en el estudio, es el Análisis-Deductivo.

Del Método Análisis: este método se inicia por la identificación de cada uno de los algoritmos para el procesamiento y reconocimiento de voz. De esa manera se tiene que descomponer el objeto de estudio de la investigación para conocer sus características. En nuestro caso se tiene que conocer las fortalezas y debilidades resultantes del análisis comparativo de algoritmos para el reconocimiento de voz.

Del Método Deductivo: este método se inicia con la observación del comportamiento de los algoritmos con el propósito de señalar las características particulares contenidas explícitamente en cada algoritmo. En nuestro caso se pretende señalar las características particulares de los algoritmos tales como las fortalezas y debilidades más saltantes.

3.6.2. Técnicas de recolección de datos.

La técnica de recolección de datos que se utiliza en el estudio, es la Observación y la Evaluación.

Observación: Son los análisis que realizan tanto el tesista y el asesor especialista en el presente proyecto, por el cual se perciben deliberadamente ciertos rasgos existentes en el objeto de conocimiento. El tipo de Observación es No Participante, ya que

utilizaremos una matriz de análisis para recoger y registrar las características de los algoritmos.

Heurística: es una técnica de indagación y descubrimiento. Para nuestro caso es la manera de buscar la solución al reconocimiento de voz mediante los algoritmos.

3.6.3. Instrumentos de recolección de datos.

Matriz de Análisis: se realiza una matriz de análisis para recoger y registrar los datos, obtenidos a través de los métodos y técnicas.

3.7. Procedimiento para la recolección de datos

El procedimiento para la recolección de datos es el siguiente:

1. Recolectar el código fuente de los algoritmos.
2. Implementar los algoritmos en un lenguaje estándar.
3. Ejecutar todos los algoritmos de ciertas características.
4. Establecer los indicadores para el procesamiento de audio, determinando el tiempo y características.
5. Evaluar 2 técnicas de procesamiento de audio y 1 técnica de reconocimiento de comandos.
6. Finalmente, tomar lectura del software de reconocimiento usando 20 casos comando de voz de entrada para determinar el número de aciertos respectivos los cuales se tabularán en el cuadro comparativo y después se hará una inferencia estadística.

3.8. Análisis estadístico e interpretación de los datos

Para el análisis estadísticos e interpretación de los datos se realizó pruebas de validación cruzada para así poder observar el grado de efectividad de la técnica, tomando así la cantidad de iteraciones.

Precisión: Es denotada por p es la cantidad de resultados acertados sobre la suma de dichos resultados más el número de falsos positivos. Medida de rendimiento que calcula la tasa de aciertos de la aplicación.

Como se ve en la ecuación:

$$P = \frac{tp}{tp + fp}$$

Exactitud: Evalúa la exactitud entre el resultado global de la exactitud y la clasificación exacta.

$$Exactidud = \frac{Reales\ Positivo + Reales\ negativos}{Predicciones\ totales}$$

Donde:

Reales positivos: representan los candidatos que han sido correctamente clasificados.

Reales Negativos: Representan a los candidatos que han sido correctamente clasificado diferente a la clase de estudio. **Sensibilidad:** Es denotada por S es la cantidad de resultados acertados sobre la suma de dichos resultados más el número de falsos negativos.

Evalúa la sensibilidad entre el resultado global de la exactitud y la clasificación exacta.

$$S = \frac{tp}{tp + fp}$$

3.9. Principios éticos

En carácter general, el presente proyecto de investigación es netamente académico y no atenta contra la seguridad e intimidad de las personas.

3.10. Criterios de rigor científico.

Explicar qué criterios de rigor científico se tomarán en cuenta y que acciones o estrategias se realizarán para garantizarlos. Se sugiere considerar los que se mencionan a continuación:

Tabla 13: Criterios de rigor científico

Criterios	Características éticas de los criterios
Validez	En la presente investigación se ha definido como variable de estudio a los métodos , los cuales abarcan como dimensiones al Procesamiento de Señales y el Reconocimiento de voz.
Generalizabilidad	Para efectos de la presente investigación, la muestra se ha tomado de forma conveniente, ya que se tiene un número limitado de posibles Métodos (Algoritmos).
Fiabilidad	Para la presente investigación el tamaño de muestra seleccionada será de 3 métodos (Algoritmo): 2 métodos para el procesamiento de señales, y 1 método para el reconocimiento de voz.
Replicabilidad	Existe la posibilidad de que la presente investigación pueda repetirse y los resultados no se contradigan.

Fuente: Elaboración Propia.

CAPITULO IV: ANÁLISIS E INTERPRETACIÓN DE LOS RESULTADOS

En este capítulo se procederá a explicar los métodos (propuestos en investigaciones anteriormente desarrolladas), así como la interpretación de los resultados:

4.1- Materiales

Para la creación de la base de datos se utilizó el programa Nero WaveEditor y se realizaron 4 grabaciones de dichas palabras la cuales son (Abrir, Apagar, Cerrar, Encender) por cada persona, teniendo un total de 20 personas, las cuales se hicieron con un micrófono High Definition Audio y en formato .WAV a 22 KHz de muestreo, 16 bits de resolución y en mono canal.

4.2- Resultados

Tiempo de Procesamiento de los Filtro

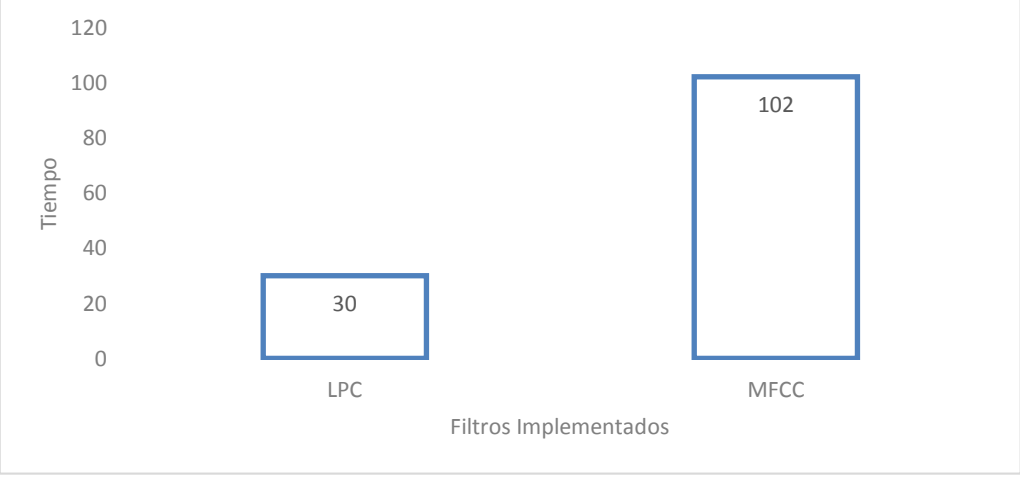
Para el Pre Procesamiento de los audios se utilizó todo el conjunto de audios y se muestra el tiempo promedio que demora cada uno de los filtros implementados. Estos resultados se pueden observar en la Tabla 14 y Grafico 15.

Tabla 14: Tabla de Tiempo de Procesamiento de los Filtros de Ruido Aplicado a las Muestras de Audio

Contenido de Audio		Abrir /Apagar/Cerrar/ Encender
Filtro	Muestras	Tiempo Promedio
LPC	20	30”
MFCC	20	102”

Fuente: Anexo 1

Gráfica 15: Ilustración de la diferencia de Tiempo de procesamiento de Filtros de Ruido Aplicado a las Muestras de Audio



Fuente: Elaboración Propia

Métricas de Evaluación:

Para evaluar el desempeño del reconocimiento de voz utilizando las redes neuronales se ha construido la matriz de confusión tal como se puede evidenciar en la tabla 14 y 15.

Tabla 14: Resumen de desempeño del método de reconocimiento de voz

BACKPROPAGATION	
80 audios	
Reconoció	57
No reconoció	23

Fuente: Elaboración Propia

Tabla 15: Desempeño del método de reconocimiento de voz mediante matriz de confusión.

	Abrir	Apagar	Cerrar	Encender
Abrir	14	2	3	1
Apagar	2	13	4	1
Cerrar	4	2	12	2
Encender	0	0	2	18

Fuente: Elaboración Propia

A partir de la matriz construida se procede a calcular las siguientes métricas:

Precisión: Mide la calidad de respuestas positivas del clasificador:
Como se ve en la ecuación:

$$P = \frac{Real\ Positivo}{Real\ Positivo + Falso\ Positivo} = \frac{14}{14+6} = 0.7$$

$$\therefore \quad P_t = 0.7135$$

Se obtuvo una Precisión de 70% y una precisión promedio de 71% de reconocimiento para los cuatro comandos de voz según la matriz de confusión establecida anteriormente.

Para calcular la **exactitud** se utiliza la siguiente formulación matemática teniendo en cuenta la matriz de confusión.

$$Exactidud = \frac{Reales\ Positivo + Reales\ negativos}{Predicciones\ totales}$$

$$Exactidud = \frac{57}{80} = 0.7125$$

Se obtuvo una Exactitud de 71% de reconocimiento para los cuatro comandos de voz según la matriz de confusión establecida anteriormente.

De igual manera se calcula la Sensibilidad que evalúa la eficiencia en la clasificación de todos los elementos que son de la misma clase y para ello se utiliza la siguiente formulación matemática.

$$S = \frac{Reales\ Positivos}{Reales\ Positivos + Falsos\ Positivos}$$

$$S = \frac{14}{20} = 0.7$$

Se obtuvo la sensibilidad de 70% de reconocimiento para los cuatro comandos de voz según la matriz de confusión establecida anteriormente.

CAPITULO V: PROPUESTA DE INVESTIGACIÓN

Para el desarrollo de la propuesta de investigación se propone un método que cuenta con 5 etapas tal como se puede evidenciar en la figura. En la primera etapa de la adquisición de la voz humana se ha generado cada audio, con cada característica de cada archivo de voz. Luego en la segunda etapa se ha filtrado la voz con LPC y MFCC, luego extraer las características de la voz, en la cuarta etapa se clasifica a través de la Red Neuronal y al final en la quinta etapa se evalúa los resultados como vemos en la Grafica 15.

El **sistema de reconocimiento de voz** comienza con la base de datos de audios de comando de voz, luego procesa cada audio grabado en .wav, para luego extraer las características, luego de haber procesado todos los audios, los datos resultantes se pasan a una red neuronal y se evalúa su precisión mediante el método de backpropagation. Para la elaboración del software, se ha utilizado el lenguaje de programación C#,

Grafica15: Propuesta de la Investigación para el reconocimiento de comando de voz

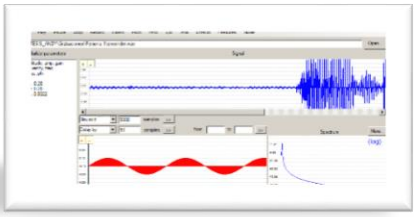
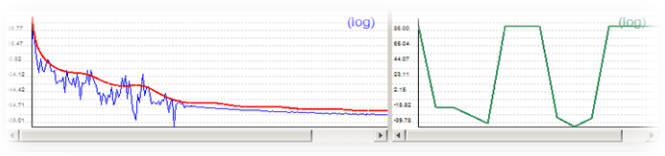


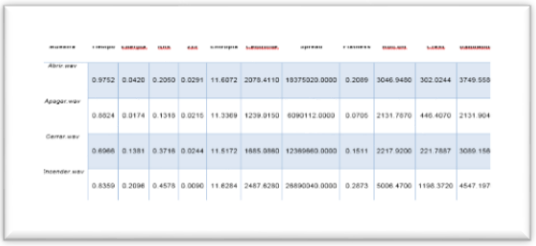
Tabla 5: Características de cada archivo de voz..

Velocidad de Transmisión	128 Kbps.
Tamaño de muestra de sonido	16 bits.
Tipo de canal	Monofónico.
Velocidad de muestreo de sonido	11 Khz
Formato de audio	*.Wav

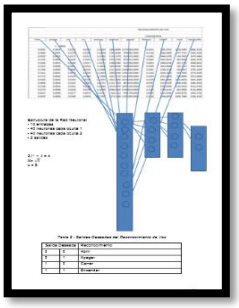
1. Adquisición de la voz humana



2. Filtro de la voz



3. Extracción de características de la voz



4. Clasificación a través de la Red Neuronal

Tabla 11: Cantidad de Entrenamientos por Error de 2 Capa

	2 capa oculta de 10 neuronas	2 capa oculta de 20 neuronas	2 capa oculta de 50 neuronas
10	0.003635941	0.002537659	0.000262604
100	0.000274606	0.000199598	0.000030093
1000	0.000023742	0.000018153	0.000003961
10000	0.000002163	0.000001694	0.000000435

Cantidad de Entrenamientos por Error

5. Resultados

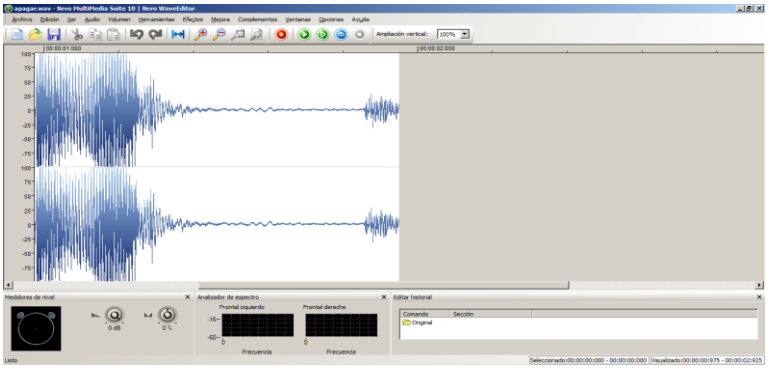
Fuente: Elaboración Propia

5.1. Base de datos con audios de comando de voz de personas con dislexia.

5.1.1. Creación de Base de datos

Para la creación de la base de datos se utilizó el programa Nero WaveEditor y se realizaron 80 grabaciones, las cuales se hicieron con un micrófono High Definition Audio y por ser un formato .wav de baja comprensión a 22 KHz de muestreo, 16 bits de resolución y en mono canal.

Gráfica 15: Creación de la base de datos



Fuente: Elaboración Propia

Tabla 4: Características de cada archivo de voz..

Velocidad de Transmisión	**128 Kbps.**
Tamaño de muestra de sonido	16 bits.
Tipo de canal	Monofónico.
Velocidad de muestreo de sonido	11 Khz
Formato de audio	*.Wav

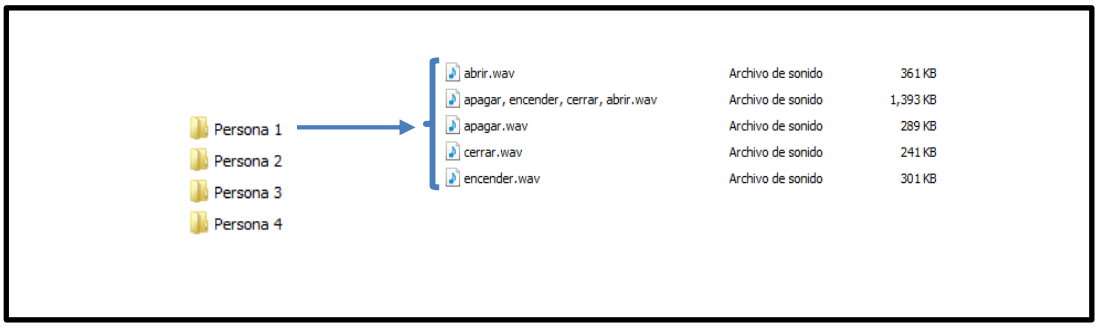
Fuente: Elaboración Propia

5.1.2. Grabación de personas con y sin dislexia

Se grabó los audios de 20 personas, para cada persona se grabó un audio en el que hablaba la palabra: abrir, apagar, cerrar, encender y un audio que une todas las palabras.

En total se tuvieron 80 archivos en formato .wav, estos audios serán utilizados para el entrenamiento del método propuesto como vemos en la Grafica 16.

Gráfica 16: Grabación de personas



Fuente: Elaboración Propia

Las características de las personas que grabaron los audios se detallan en la tabla 5.

Tabla 5 : Base de audios de Personas

Persona	Edad	Sexo	Dislexia
Persona 1	12	M	No
Persona 2	24	F	No
Persona 3	52	M	No
Persona 4	8	M	Si
Persona 5	20	F	Si
Persona 6	52	M	Si

Fuente: Elaboración Propia

5.1.2. Seleccionar Métodos de Procesamiento Y Reconocimiento de Voz

Para seleccionar el método de procesamiento en el anexo 2 se obtuvo que MFCC tiene una efectividad 98,6% y LPC con un asertividad 81%. Para ello se hizo una evaluación de dichos métodos En las siguientes páginas se presenta la selección para LPC y MFCC.

Tabla 6: Selección Métodos de Procesamiento

	Métodos	Características	Valoración
Procesamiento de Voz	Dynamic time warping (DTW) ***Alineamiento Temporal Dinámico***	El algoritmo DTW ofrece buenos resultados para un conjunto pequeño de palabras a reconocer, si se requiere reconocimiento para un vocabulario extenso, esta solución no es la más óptima computacionalmente. Las muestras se tomaron en un ambiente bastante natural, por tanto, la efectividad de la aplicación, que reside en un 84,45%, puede aumentar considerablemente. (Baquero, 2011)	3
	Linear Predictive Coding (LPC) ***Codificación Predictiva Lineal***	Que el reconocimiento de sílabas y vocales, implementado a través de un software apoyado en las teorías de WT, coeficientes LPC y BPNN, ofrece interesantes perspectivas para constituirse en una herramienta automática y amigable para complementar el aprendizaje del habla para personas que padecen problemas de audición. Empleando la totalidad de los ejemplos para entrenamiento (130 por cada vocal), se obtuvieron los coeficientes LPC que se ingresan a la red de vocales y se alcanzó un 91% de asertividad, según se aprecia en la Tabla 2. (Sánchez, 2016)	2
	Mel Frequency Cepstral Coefficient (MFCC) ***Coeficientes Cepstrales de la escala de Mel***	El método para el reconocimiento de voz "Extracción de los coeficientes cepstrales de la escala de Mel", en interfaz presenta un 98,6% de efectividad, conforme las pruebas realizadas, dicho porcentaje de efectividad puede ser mejorado aún más al elegir comandos que difieran en su pronunciación. Su proyecto comprende además el desarrollo de un algoritmo de reconocimiento de voz que permita hacer una simulación de la detección de los comandos de audio emitidos por las personas con cuadriplejia disminuyan la dependencia hacia otras personas y puedan interactuar con sus electrodomésticos, luces, puertas, ventanas y demás elementos del domicilio. (Rodríguez & Guerra, 2006)	1

5.1.2.1. *Codificación Predictiva Lineal*

LPC define los coeficientes de un predictor lineal hacia delante al minimizar el error de predicción en el sentido de mínimos cuadrados.

La fórmula matemática en la que se basa el Codificación Predictiva Lineal es la siguiente:

$$S(n)=\sum_{k=1}^{p} A_k S(n-K) + e(n) \qquad (1)$$

Donde la parte observación *S(n)* representa la señal de voz original, en la aproximación la suma de dicha señal retrasada *k* muestras pasadas desde 1 hasta *p* multiplicadas por sus amplitudes *Ak* es su aproximación artificial. Finalmente, la parte aleatoria, *e(n)* es la diferencia o error existente entre ambas.

Pre-énfasis

Para hacer menos sensible al sistema a los efectos de cuantización por longitud finita de palabra, se pasa a la señal de entrada por un filtro de bajo orden (típicamente un filtro FIR de primer orden) de manera de aplanar su espectro.

Un filtro de pre-énfasis típico es

$H(z) = 1 - az^{-1}$, $0.9 \leq a \leq 1.0$

Un valor muy usado de a es 0.95

Es importante destacar que el filtro de preénfasis se usó en este proyecto con la finalidad única de reducir el nivel de ruido de la señal original

- Segmentación:

Se realizó mediante la función framing que divide la señal en tramas de 80 muestras (que representan 29,6 mseg de voz a una frecuencia de muestreo de 16 KHz) para considerar la señal de voz como estacionaria en ese intervalo de tiempo y crea una matriz “M” que tiene en cada una de sus columnas las muestras correspondientes a cada trama en la que se dividió la señal de voz. Para la segmentación se utilizó un solapamiento entre tramas consecutivas de 80 muestras.

- Aplicación de la ventana Hamming:

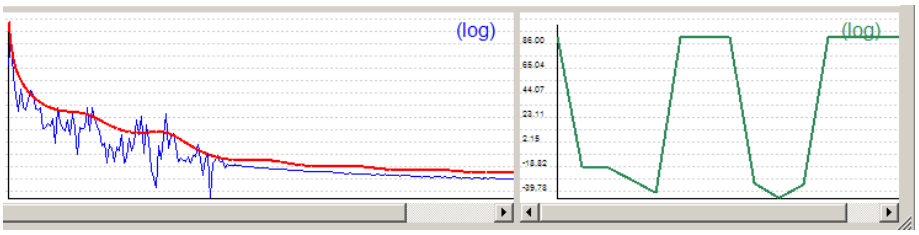
Una vez obtenida la matriz M de la etapa de segmentación, mediante la función windowing se creó una matriz h que en su diagonal contiene los valores de una ventana Hamming y que, multiplicada por M, devuelve una matriz “M2” que corresponde a la ventana aplicada a cada una de las tramas contenidas en M.

Una vez que ya se tienen las tramas correspondientes a la señal de voz con la ventana aplicada a cada una de ellas. Implementación del algoritmo

```
1.  var lpcc = new float[FeatureCount];
2.  lpcc[0] = (float)Math.Log(err);
3.  for (var n = 1; n < FeatureCount; n++)
4.  {
5.  var acc = 0.0f;
6.  for (var k = 1; k < n; k++)
7.  {
8.  acc += k * lpcc[k] * lpc[n - k];
9.  }
10. lpcc[n] = -lpc[n] - acc / n;
11. }
```

En el código en la fila número 2 obtenemos el resultado del filtro utilizando LPC.

Gráfica 18: Señal de un audio aplicando LPC



Fuente: Elaboración Propia

5.1.2.2. Coeficientes Cepstrales de la escala de Mel

Para calcular estos coeficientes, se utilizó la función del

$$mel(f) = 2595log_{10}\left(1 + \frac{f}{100}\right)$$

Comenzamos con una señal de voz, asumiremos que se muestrea a 16 kHz.

1. El marco de la señal en marcos de 20-40 ms. 25ms es estándar. Esto significa que la longitud del marco para una señal de 16 kHz es 0.025 * 16000 = 400 muestras. El paso de fotogramas suele ser algo así como 10 ms (160 muestras), lo que permite cierta superposición con los fotogramas. El primer marco de muestra 400 comienza en la muestra 0, el siguiente marco de muestra 400 comienza en la muestra 160, etc. hasta que se alcanza el final del archivo de voz. Si el archivo de voz no se divide en un número par de cuadros, rellénelo con ceros para que lo haga.

Los siguientes pasos se aplican a cada trama, se extrae un conjunto de 12 coeficientes MFCC para cada trama. Un breve aparte en notación: llamamos a nuestra señal de dominio de tiempo $S(n)$. Una

vez que se enmarca, tenemos $S_i(n)$donde n varía entre 1-400 (si nuestras tramas son 400 muestras) y i abarca la cantidad de tramas. Cuando calculamos la DFT compleja, obtenemos $S(k)$ - donde $i$denota el número de cuadro correspondiente al marco de dominio de tiempo. $P_i(k)$ Es entonces el espectro de potencia del marco i.

2. Para tomar la Transformada Discreta de Fourier del marco, realice lo siguiente:

$$S_i\,(k) = \sum_{n=!}^{N} s_i(n)h(n)e^{-j2\pi\,\mathrm{kn/N}} \qquad 1 \le k \le K$$

Donde $h(n)$ es una N ventana de análisis de muestra larga (por ejemplo, la ventana de Hamming), y Kes la longitud de la DFT. La estimación espectral de potencia basada en periodo gramas para el cuadro de voz $S_i(n)$ viene dada por:

$$P_i(k) = \frac{1}{N}|S_i(k)|^2$$

Esto se conoce como la estimación de periodograma del espectro de potencia. Tomamos el valor absoluto de la transformada de Fourier compleja y cuadramos el resultado. En general, realizaríamos una FFT de 512 puntos y conservaríamos solo los primeros 257 coeficientes.

3. Calcule el banco de filtros espaciados por Mel. Este es un conjunto de 20-40 (26 es estándar) filtros triangulares que aplicamos a la estimación espectral de potencia del periodograma del paso 2.

Nuestro banco de filtros viene en forma de 26 vectores de longitud 257 (asumiendo los ajustes de FFT para el paso 2). Cada vector es mayormente de ceros, pero no es cero para una determinada sección del espectro. Para calcular las energías del banco de filtros, multiplicamos cada banco de filtros con el espectro de potencia y luego sumamos los coeficientes. Una vez realizado esto, nos quedan 26 números que nos dan una indicación de cuánta energía había en cada banco de filtros. Para una explicación detallada de cómo calcular los bancos de filtros en la gráfica 20.

4. Tome el registro de cada una de las 26 energías del paso 3. Esto nos deja con 26 energías del banco de filtros de registro.

5. Tomar la Transformada de Coseno Discreto (DCT) de las 26 energías de banco de filtros de registro para obtener 26 coeficientes cepstrales. Para ASR, solo se mantienen los 12-13 más bajos de los 26 coeficientes.

Las características resultantes (12 números para cada cuadro) se denominan coeficientes de Cepstral de frecuencia de mel.

Cálculo del banco de filtros Mel

En esta sección, el ejemplo usará 10 bancos de filtros porque es más fácil de mostrar, en realidad, se usarían 26 a 40 bancos de filtros.

Primero tenemos que elegir una frecuencia superior e inferior. Los valores buenos son 300Hz para la frecuencia más baja y 8000Hz para la frecuencia superior. Por supuesto, si la voz se muestrea a 8000Hz, nuestra frecuencia superior está limitada a 4000Hz. Luego sigue

estos pasos:

1.Usando la ecuación 1 , convierta las frecuencias superior e inferior a Mels. En nuestro caso, 300Hz es 401.25 Mels y 8000Hz es 2834.99 Mels.

Para este ejemplo haremos 10 bancos de filtros, para los cuales necesitamos 12 puntos. Esto significa que necesitamos 10 puntos adicionales espaciados linealmente entre 401.25 y 2834.99. Esto sale a:

m (i) = 401.25, 622.50, 843.75, 1065.00, 1286.25, 1507.50,
1728.74,1949.99, 2171.24, 2392.49, 2613.74, 2834.99

Ahora usa la ecuación 2 para convertir estos de nuevo a Hertz:

h (i) = 300, 517.33, 781.90, 1103.97, 1496.04, 1973.32, 2554.33,
3261.62, 4122.63, 5170.76, 6446.70, 8000

Tenga en cuenta que nuestros puntos de inicio y fin están en las frecuencias que queríamos.

No tenemos la resolución de frecuencia requerida para colocar los filtros en los puntos exactos calculados anteriormente, por lo que necesitamos redondear esas frecuencias al contenedor FFT más cercano. Este proceso no afecta la precisión de las características. Para convertir las funciones en números de papeleras FFT, necesitamos conocer el tamaño de FFT y la frecuencia de muestreo.

$$F(i) = piso\ ((nfft + 1) * h\ (i)/\ muestra$$

Esto resulta en la siguiente secuencia:

f (i) = 9, 16, 25, 35, 47, 63, 81, 104, 132, 165, 206, 256

Podemos ver que el banco de filtros final termina en el contenedor

256, que corresponde a 8kHz con un tamaño FFT de 512 puntos.

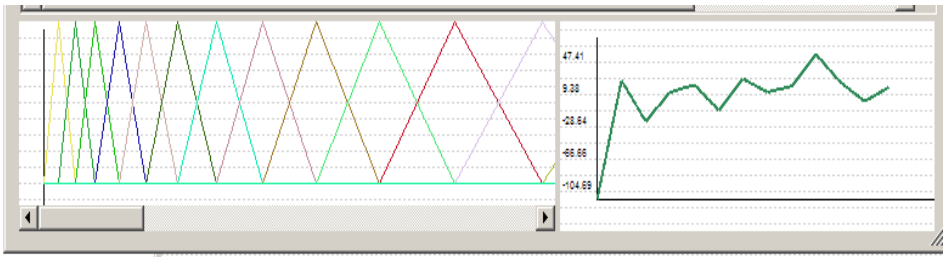
Ahora el primer banco de filtros comenzará en el primer punto, alcanzará su pico en el segundo punto, luego regresará a cero en el tercer punto. El segundo banco de filtros comenzará en el segundo punto, alcanzará su máximo en el tercero, luego será cero en el cuarto etc. Una fórmula para calcular esto es la siguiente:

$$H_m(k) = \begin{cases} 0 & k < f(m-1) \\ \dfrac{k - f(m-1)}{f(m) - f(m-1)} & f(m-1) \le k \le f(m) \\ \dfrac{f(m+1) - k}{f(m+1) - f(m)} & f(m) \le k \le f(m+1) \\ 0 & k > f(m+1) \end{cases}$$

Una muestra de audio digital de 1,5 segundos muestreada a 11 KHz en formato integer de 16 bytes son 280,5 KiB de datos. Se busca extraer características que permitan distinguir esta cadena de datos de otra, con claridad y con el menor número de datos posible para evitar sobrecargar de trabajo el procesador con información redundante o no útil.

```
1. var mfccExtractor = new MfccExtractor(13,
2. melFilterbankSize: 20,
3. preEmphasis: 0.95,
4. window: WindowTypes.Hamming);
5. _mfccVectors = mfccExtractor.ComputeFrom(_signal);
6. FillFeaturesList(_mfccVectors, mfccExtractor.FeatureDescriptions);
7. mfccListView.Items[0].Selected = true;
8. melFilterBankPanel.Groups = mfccExtractor.FilterBank;
9. mfccPanel.Line = _mfccVectors[0].Features;
```

Gráfica 20: Banco de filtros triangulares implementado.



Fuente: Elaboración Propia

5.1.3. Extracción de características

Luego de procesamiento adecuado del audio se continua a extraer las características del audio. Estas poseen un significado físico, por lo que permiten una calificación de las cualidades vocales. El objetivo de la extracción de características es tener una representación numérica precisa y compacta. Las siguientes características que se obtuvieron fueron: Time(Tiempo), Energy (Energía), Root Half Square(Raíz Media Cuadrado), Zero Crossing Ratio (La Velocidad de Cruce Cero), Entropy (La Entropía, Centroid (El centroide),Spectral Flatness Measure (Medida De Planitud Espectral),Roll-off(Frecuencia):

Las Cuales son de dominio espectral y dominio temporal estas características se utilizarán por su cuya importancia en el Backpropagation.

Aquí Extraeremos las características de la voz.

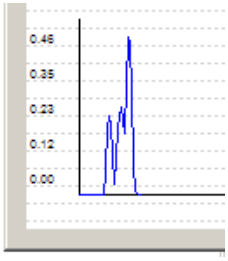
Time(Tiempo): Es el tiempo de la duración de la señal del audio

$$td = Tf - Ti$$

Energy (Energía): Es una medida de la fuerza de la señal en cualquier momento. Es capturada por un micrófono depende de la sensibilidad del transductor y la distancia que hay entre la persona y el micrófono, debido a que, a mayor distancia, menor es la amplitud de la voz, es escogida como característica teniendo en cuenta que los fonemas sordos contienen más energía que los segmentos de silencio

```
1. public float Energy(int startPos, int endPos)
2. {
3. var total = 0.0f;
4. for (var i = startPos; i < endPos; i++)
5. {
6. total += Samples[i] * Samples[i];
7. }
8. return total / (endPos - startPos);
9. }
```

Gráfica 17: Rango de la Energia de la voz

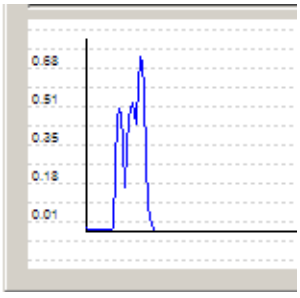


Fuente: Elaboración Propia

Root Half Square(Raíz Media Cuadrado): Se usa para estimar el tiempo de subida, el tiempo de decaimiento, la fuerza y la frecuencia de la amplitud modulación.

```
public float Rms(int startPos, int endPos)
{
return (float)(Math.Sqrt(Energy(startPos, endPos)));
}
```

Gráfica 18: Rango de Root Half Square de la voz



Fuente: Elaboración Propia

Zero Crossing Rate (La Velocidad de Cruce Cero): Es el número de veces que la señal pasa por cero por unidad de tiempo. En una señal discreta, el paso por cero se da cuando muestras consecutivas tienen signos diferentes, por lo que el cálculo de este parámetro se realiza contando el número de cambios de signo por unidad de tiempo. Un valor alto del ZCR está relacionado con una mayor cantidad de componentes de alta frecuencia en la señal.

$$Z_t = \sum_{N=1}^{N-1} \frac{|sign(x[n+1]) - sign(x[n])|}{N}$$

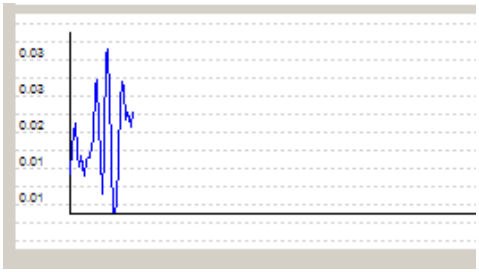
Siendo x[n] el valor de la muestra y N el número de muestras totales en la trama t.

El ZCR resulta útil para determinar si una señal contiene o no voz.

```
1.  public float ZeroCrossingRate(int startPos, int endPos)
2.  {
3.  const float disbalance = 1e-4f;
4.  var prevSample = Samples[startPos] + disbalance;
5.  var rate = 0;
6.  for (var i = startPos + 1; i < endPos; i++)
7.  {
8.  var sample = Samples[i] + disbalance;
9.  if ((sample >= 0) != (prevSample >= 0))
10. {
11. rate++;
```

```
12. }
13. prevSample = sample;
14. }
15. return (float)rate / (endPos - startPos - 1);
16.       }
```

Gráfica 19: Rango de Zero Crossing Rate de la voz

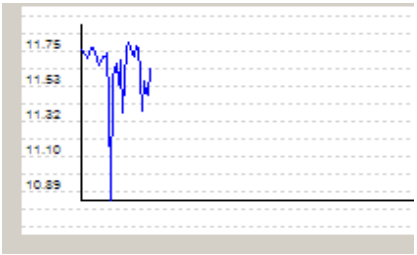


Fuente: Elaboración Propia

Entropy (La Entropía): La idea de utilizar la entropía es mejorar las falencias dejadas por el análisis la energía

```
1.  public float Entropy(int startPos, int endPos)
2.  {
3.  var sum = 0.0f;
4.  for (var i = startPos; i < endPos; i++)
5.  {
6.  sum += Math.Abs(Samples[i]);
7.  }
8.  var entropy = 0.0;
9.  for (var i = startPos; i < endPos; i++)
10. {
11. var p = Math.Abs(Samples[i]) / sum;
12. entropy -= p * Math.Log(p + float.Epsilon, 2);
13. }
14. return (float)entropy;
15. }
```

Gráfica 20: Rango de Entropy de la voz



Fuente: Elaboración Propia

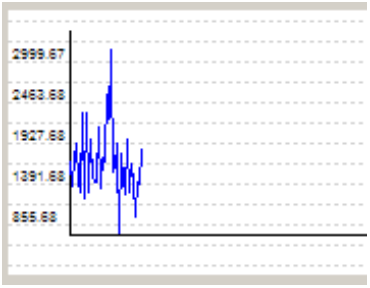
Centroid (El centroide); Este parámetro calcula el centro de gravedad espectral de una trama de la señal.

$$C_t = \frac{\sum_{K=1}^{N} |X[k]|.k}{\sum_{K=1}^{N} |X[k]|}$$

Donde X[k] representa la muestra k-ésima de la Transformada Discreta de Fourier correspondiente a la trama y N es el número de muestras de la ventana.

```
1. public static float Centroid(float[] spectrum, float[] frequencies)
2. {
3. var sum = 0.0f;
4. var weightedSum = 0.0f;
5. for (var i = 1; i < spectrum.Length; i++)
6. {
7. sum += spectrum[i];
8. weightedSum += frequencies[i] * spectrum[i];
9. }
10. return weightedSum / sum;
11. }
```

Gráfica 24: Rango del Centroide de la voz



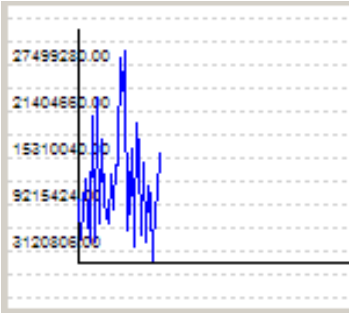
Fuente: Elaboración Propia

Spread (La Dispersión Espectral): Describe la desviación promedio de la tasa de mapa alrededor de su centroide, que se asocia comúnmente con el ancho de banda de la señal.

```
1. public static float Spread(float[] spectrum, float[] frequencies)
2. {
3. var mean = 0.0f;
4. for (var i = 1; i < spectrum.Length; i++)
5. {
6. mean += spectrum[i];
```

```
}
mean /= spectrum.Length;
var sum = 0.0f;
var weightedSum = 0.0f;
for (var i = 1; i < spectrum.Length; i++)
{
sum += spectrum[i];
weightedSum += spectrum[i] * (frequencies[i] - mean) * (frequencies[i] -
mean);
}
return weightedSum / sum;
}
```

Gráfica 21: Rango de la Dispersion Espectral de la voz

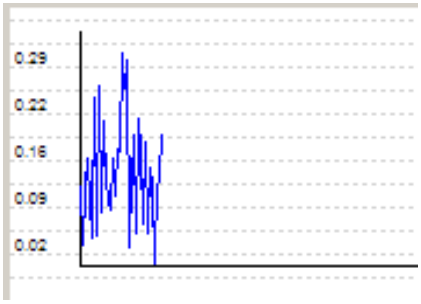


Fuente: Elaboración Propia

Spectral Flatness Measure (Medida De Planitud Espectral): Se define como la relación entre la media geométrica y la media aritmética.

```
public static float Flatness(float[] spectrum, float[] frequencies, float
minLevel = 1e-10f)
{
var sum = 0.0f;
var logSum = 0.0;
for (var i = 1; i < spectrum.Length; i++)
{
var amp = Math.Max(spectrum[i], minLevel);
sum += amp;
logSum += Math.Log(amp);
}
sum /= spectrum.Length;
logSum /= spectrum.Length;
return sum > 0 ? (float)Math.Exp(logSum) / sum : 0.0f;
      }
```

Gráfica 22: Rango de la Medida de Planitud Espectral de la voz



Fuente: Elaboración Propia

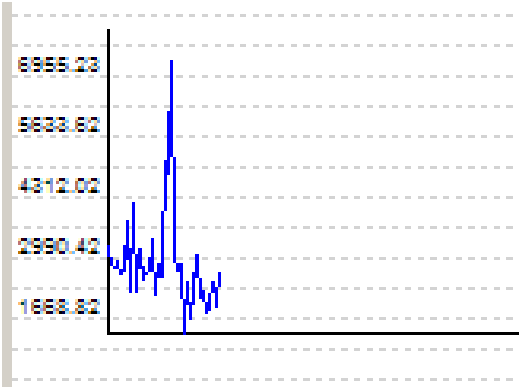
Roll-off: Es una frecuencia de corte tal que por debajo de ella reside el 85% (o 95%) de la energía total de la señal

$$\sum_{k<RF} |X[k]|^2 = 0.85 \cdot \sum_{k} |X[k]|^2$$

Siendo X[k] la Transformada Discreta de Fourier de la trama, k es el bin (unidad mínima en el dominio espectral) del eje de frecuencias discreto.

```
1.  public static float Rolloff(float[] spectrum, float[] frequencies, float
    rolloffPercent = 0.85f)
2.  {
3.  var threshold = 0.0f;
4.  for (var i = 1; i < spectrum.Length; i++)
5.  {
6.  threshold += spectrum[i];
7.  }
8.  threshold *= rolloffPercent;
9.  var cumulativeSum = 0.0f;
10. var index = 0;
11. for (var i = 1; i < spectrum.Length; i++)
12. {
13. cumulativeSum += spectrum[i];

14. if (cumulativeSum > threshold)
15. {
16. index = i;
17. break;
18. }
19. }

20. return frequencies[index];
21. }
```

Gráfica 23: Rango de Roll-Off de la Voz:

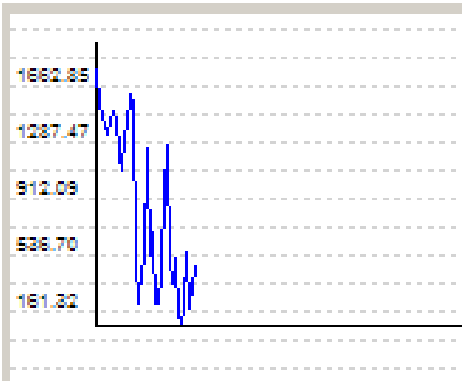


Fuente: Elaboración Propia

Crest: Una cresta es el punto más alto al que se eleva el medio y un valle es el punto más bajo al que se hunde el medio.

```
1.  public static float Crest(float[] spectrum)
2.  {
3.  var sum = 0.0f;
4.  var max = 0.0f;

5.  for (var i = 1; i < spectrum.Length; i++)
6.  {
7.  var s = spectrum[i] * spectrum[i];

8.  sum += s;

9.  if (s > max)
10. {
11. max = s;
12. }
13. }

14. return sum > 0 ? spectrum.Length * max / sum : 1.0f;
15. }
```

Gráfica 29: Rango de la cresta de la voz

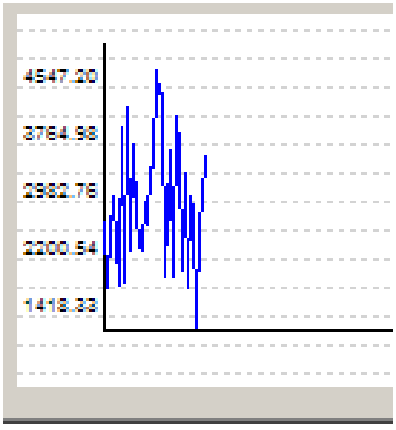


Fuente: Elaboración Propia

Bandwidth: Si se refiere a los límites del oído humano, generalmente se acepta que el límite superior es alrededor de 20 kHz más o menos. El límite inferior es probablemente de 10 Hz ... por debajo de eso, creo que el sonido probablemente se escucha como picos y valles individuales de presión y no se distingue claramente.

```
1.  public static float Bandwidth(float[] spectrum, float[] frequencies, float p
    = 2)
2.  {
3.  var centroid = Centroid(spectrum, frequencies);
4.  var norm = spectrum.Sum(s => Math.Abs(s));
5.  var sum = 0.0;
6.  for (var i = 1; i < spectrum.Length; i++)
7.  {
8.  sum += spectrum[i] / norm * Math.Pow(Math.Abs(frequencies[i] - centroid), p);
9.  }
10. return (float)Math.Pow(sum, 1/p);
11. }
```

Gráfica 30: Rango del Ancho de banda de la voz



Fuente: Elaboración Propia

5.1.4. Seleccionar el Método de Reconocimiento De Voz

El algoritmo “back-propagation” es el empleado en el entrenamiento de las redes del tipo perceptrón multicapa de 3 o más capas.

Tabla 7: Selección Métodos de Reconocimiento

	Métodos	Características	Valoración

Reconocimiento de Voz	Redes Neuronales	El algoritmo RNN ofrece buenos resultados para un conjunto pequeño de palabras a reconocer, si se requiere reconocimiento para un vocabulario extenso, esta solución no es la más óptima computacionalmente. Las muestras se tomaron en un ambiente bastante natural, por tanto, la efectividad de la aplicación, que reside en un 84,45%, puede aumentar considerablemente. (Baquero, 2011)	1
	Redes Bayesianas	Para los conjuntos de datos reales el valor de la exactitud más alta no es conocido, por lo tanto, la efectividad de la aplicación de 78,45%	2

Se ha escogido como clasificador a la red neuronal, para la clasificación de las muestras obtenidas en base a las características extraídas, teniendo en cuenta las siguientes características: El número de neuronas de entrada es igual al número características extraídas por voz, dichas características

La Red neuronal con la estructura de una de 1 capa de entrada (11 neuronas), 2 capas ocultas y 1 capa de salida (2 neuronas).

Para determinar el número de neuronas ocultas además de seguir la Regla de la capa oculta-capa entrada, la cual indica que el número de neuronas ocultas no debe ser mayor al doble del número de neuronas de la capa de entrada.

Al final 1 capa de salida con 2 neuronas por la cantidad de 4 diagnósticos (Abrir, Apagar, Cerrar, Encender)

Las pruebas se realizaron con distintos números de neuronas ocultas

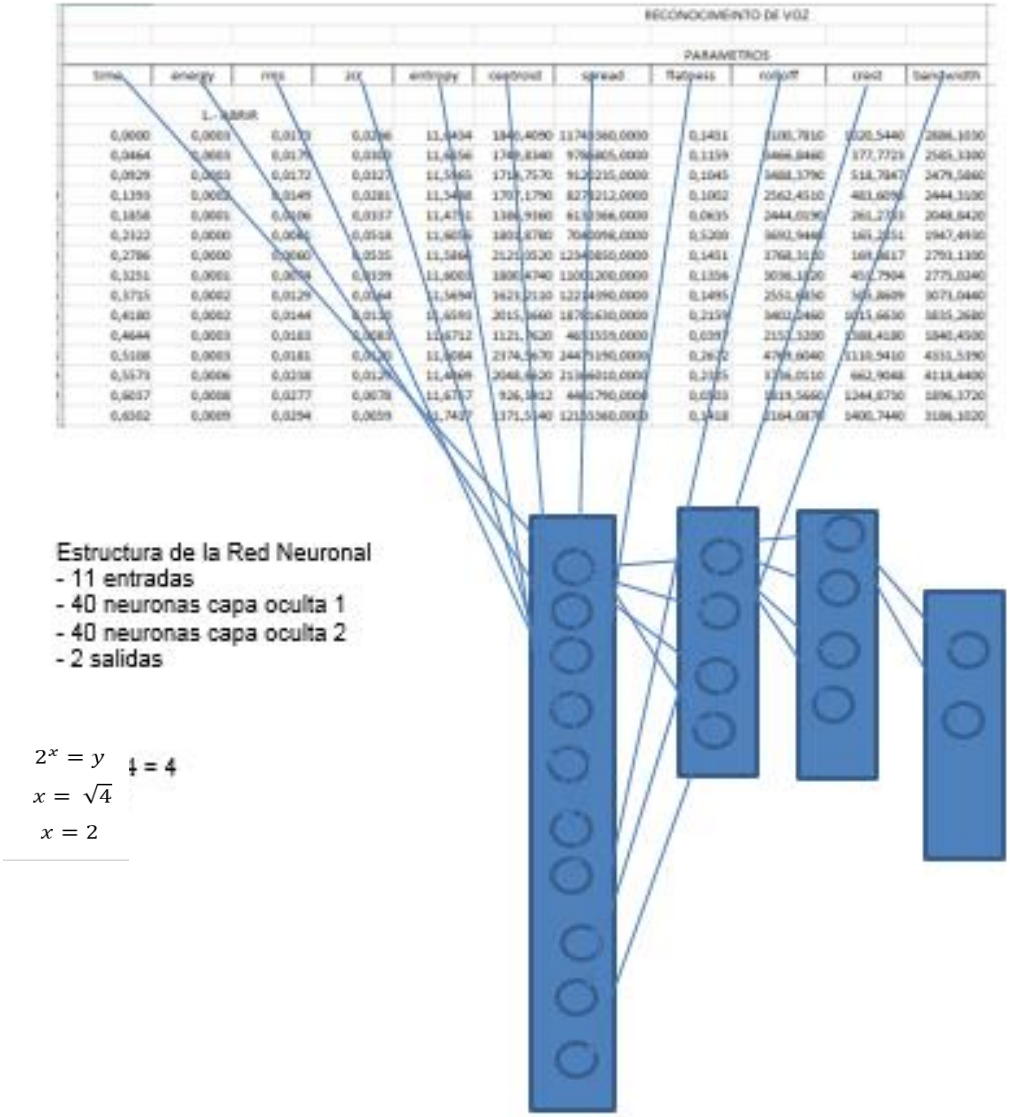
hasta llegar al número actual.

Tabla 8 : Salidas Deseadas del Reconocimiento de Voz

Salida Deseada		Reconocimiento
0	0	Abrir
0	1	Apagar
1	0	Cerrar
1	1	Encender

Fuente: Elaboración Propia.

Gráfica 24: Estructura de la Red Neuronal



$$2^x = y \quad \text{ł} = 4$$
$$x = \sqrt{4}$$
$$x = 2$$

Fuente: Elaboración Propia.

5.1.5. Arquitecturas respecto a su desempeño

Para ver el rendimiento margen de error que produce una red neuronal, se ha establecido 3 arquitecturas que se describen a continuación:

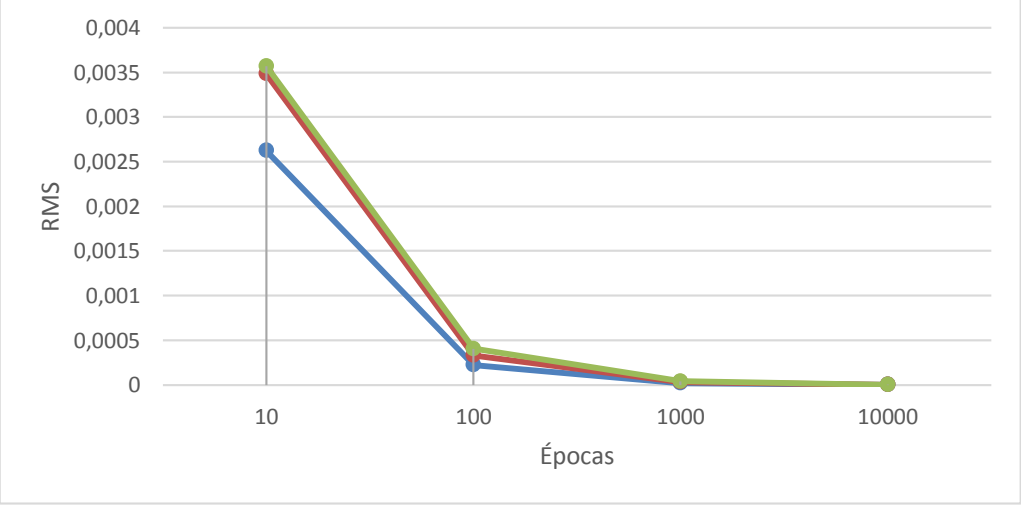
Arquitectura de Red 1: Se hizo el entrenamiento con las siguientes iteraciones, teniendo una capa de entrada de 11 neuronas, 1 capa oculta de 10,20 y 50 neuronas y la salida 2. El resultado se puede evidenciar en la siguiente tabla.

Tabla 15: Entrenamientos según el N° de Iteraciones por Error de 1 Capa

Iteraciones	1 capa oculta de 10 neuronas / tiempo	1 capa oculta de 20 neuronas / tiempo	1 capa oculta de 50 neuronas / tiempo
10	0.002623564 /3”	0.000859042/8”	0.000088102/12”
100	0.000223161 /45”	0.000106657 /48”	0.000077078/ 50”
1000	0.00002166 / 1’ 30”	0.000011484/ 1’ 33”	0.00000755/ 1’ 39”
10000	0.000002146 / 2’	0.000001154 / 2’ 5”	0.00000075/ 2’ 10”

Fuente: Elaboración Propia.

Gráfica 25: Entrenamientos según el N° de Iteraciones por con 1 Capa



Fuente: Elaboración Propia.

De la gráfica 31 se observa cuando el número de iteraciones 10 el error tiende hacer un valor 0.002623564 alto pero el tiempo que demora es de 30 seg. si se continúa analizando la gráfica se observa que cuando el número de iteraciones es 10000 tiende hacer un valor de 0.00000075, pero el tiempo que demora es de 2 min 10 seg es muy elevado.

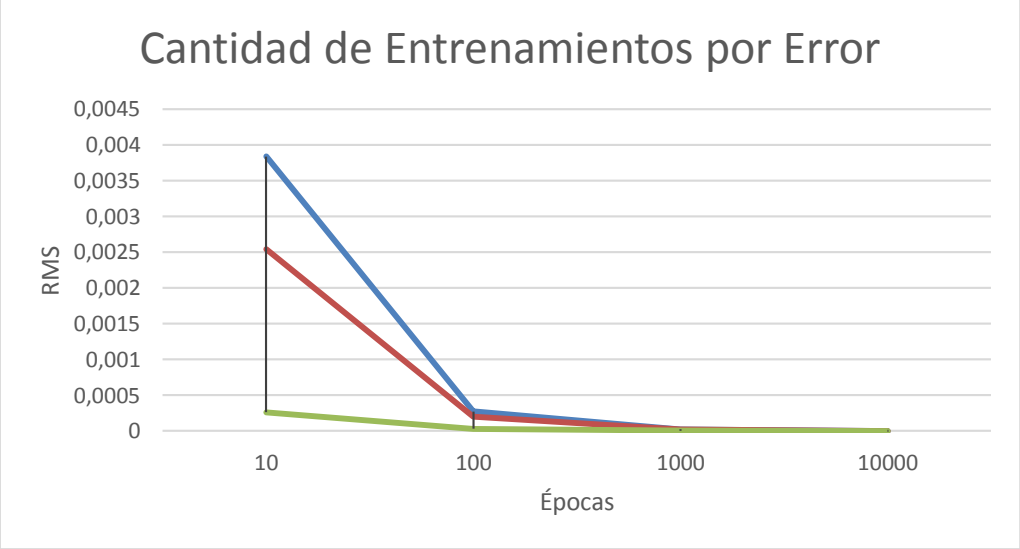
Arquitectura de Red 2: Se hizo el entrenamiento con las siguientes iteraciones, teniendo una capa de entrada de 11 neuronas, 1 capa oculta de 10,20 y 50 neuronas y la salida 2. El resultado se puede evidenciar en la siguiente tabla.

Tabla 6: Entrenamientos según el N° de Iteraciones por Error de 2 Capa

Iteraciones	2 capa oculta de 10 neuronas / tiempo	2 capa oculta de 20 neuronas / tiempo	2 capa oculta de 50 neuronas / tiempo
10	0.003835941/5”	0.002537659/10”	0.000262604/17”
100	0.000274606/1’	0.000199598/1’ 10”	0.000030093/1’ 15”
1000	0.000023742/1’ 30”	0.000018153/1’ 40”	0.000003961/1’ 44”
10000	0.000002163/ 2’	0.000001694/2’ 4”	0.000000435/2’ 10”

Fuente: Elaboración Propia.

Gráfica 26: Entrenamientos según el N° de Iteraciones por Error de 2 Capa



Fuente: Elaboración Propia.

De la gráfica 2 se observa cuando el número de iteraciones 10 el error tiende hacer un valor 0.003835941 alto pero el tiempo que demora es de 30 seg. si se continúa analizando la gráfica se observa que cuando el número de iteraciones es 10000 tiende hacer un valor de 0.000000435, pero el tiempo que demora es de 2 min 10 seg es muy elevado

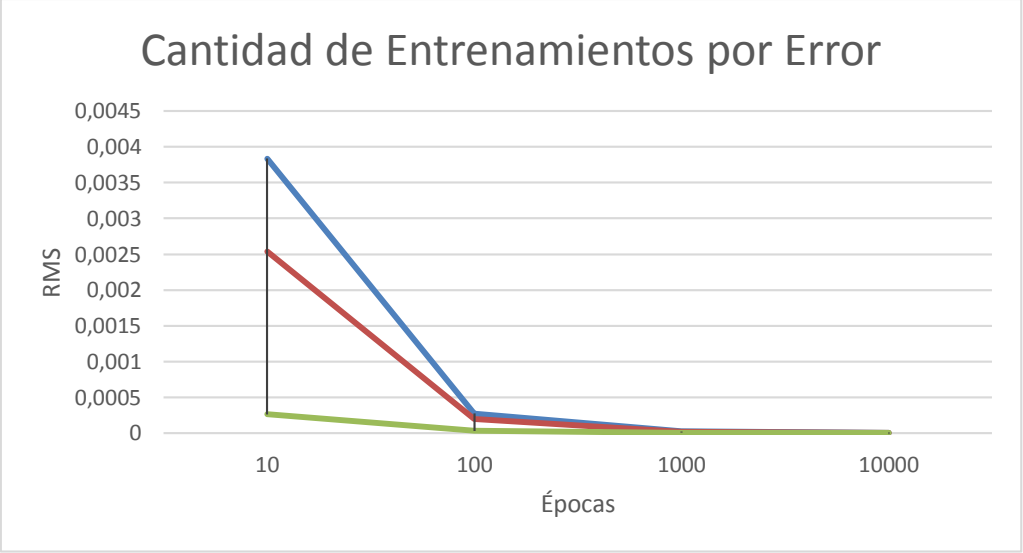
Arquitectura de Red 3: Se hizo el entrenamiento con las siguientes iteraciones, teniendo una capa de entrada de 11 neuronas, 1 capa oculta de 10,20 y 50 neuronas y la salida 2. El resultado se puede evidenciar en la siguiente tabla.

Tabla 17: Entrenamientos según el N° de Iteraciones por Error de 3 Capa

Iteraciones	3 capa oculta de 10 neuronas / tiempo	3 capa oculta de 20 neuronas / tiempo	3 capa oculta de 50 neuronas / tiempo
10	0.002975941/8”	0.002437639/10”	0.000251204/20”
100	0.000264642/1’ 4”	0.000179528/1’ 12”	0.000029091/1’ 18”
1000	0.000022742/1’ 33”	0.000017153/1’ 42”	0.000003761/1’ 46”
10000	0.000001963/ 2’ 2”	0.000001594/2’ 6”	0.00000029/2’ 12”

Fuente: Elaboración Propia.

Gráfica 277: Entrenamientos según el N° de Iteraciones por Error de 3 Capa



Fuente: Elaboración Propia.

De la gráfica 27 se observa cuando el número de iteraciones 10 el error tiende hacer un valor 0.002975941 alto pero el tiempo que demora es de 8 seg. si se continúa analizando la gráfica se observa que cuando el número de iteraciones es 10000 tiende hacer un valor de 0.00000029, pero el tiempo que demora es de 2 min 12 seg es muy elevado.

CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

- Se ha construido una base de datos de audios de 4 personas. Por cada persona contiene 4 archivos de audio en formato .wav. Cada archivo de audio tiene un tamaño promedio de 300 kb.
- Para el procesamiento de audio se ha seleccionado algoritmo Coeficientes Predicción Lineal (LPC) y Coeficientes Ceptrales de Frecuencia de Mel (MFCC). Para el reconocimiento de voz se implementó una red neuronal de tipo de backpropagation.
- Se ha implementado los métodos descritos utilizando el lenguaje de programación C# en una arquitectura de 64 bits de Windows 7 INTEL CORE i3 2.2 GHZ con una memoria RAM: 8GB
- Se aplicaron el algoritmo backpropagation, donde se obtuvo el mejor tiempo del algoritmo backpropagation 48 segundos con un error 0.000106657.
- Para el reconociendo de comando de voz se obtiene una precisión de 71%, exactitud 71% y una sensibilidad de 70%

6.2 Recomendaciones

- La investigación logro reconocer voces de audios en formato wav., pero se recomienda para futuras investigaciones realizar la detección en audio mp3.
- Se logró identificar cuatro comandos de voz en Visual Studio, pero se recomienda trabajar con otros comandos para ver el comportamiento de las redes neuronales.
- Para el reconocimiento de voz se recomienda utilizar otros algoritmos como Modelos ocultos de Markov, Alineamiento temporal dinámico, entre otros.

BIBLIOGRAFÍA

Aguilar, L. (2013). *Análisis de las principales plataformas Smart TV*. Obtenido de Teknofilo: http://www.teknofilo.com/

Bejerano, P. G. (2014). *El reconocimiento de voz: una interfaz para dominarlos a todos*. Obtenido de ThinkBig: www.blogthinkbig.com/

Bellesi , F., & Ortiz, F. (2009). *Reconocimiento de voz para aplicacion en domotica.* Argentina.

Camargo, J., Garcia, L., & Gaona, E. (2012). Reconocimiento de voz humana aplicado a la domótica. *Ingenium*, 97-106.

Fajardo C., D. (2009). *Vulnerabilidades: Sistemas biométricos tienen su talón de Aquiles*. Obtenido de El Mercurio S.A.P.: http://www.edicionesespeciales.elmercurio.com/

Lumen Vox. (s.f.). *La Historia de la Tecnología de Reconocimiento de Voz*. Obtenido de Speech Understood: http://www.lumenvox.com/espanol/resources/tips/historyOfSpeechRecognition.aspx

MIT Technology Review. (2012). *Business Impact: El futuro del reconocimiento de voz*. Obtenido de TechnologyReview: www.technologyreview.es/

Oropeza Rodriguez, J. (2006). Algoritmos y Métodos para el Reconocimiento de Voz en Español mediante sílabas. *Scielo*, 270-286.

Pérez Badillo, E. O., Poceros Martínez, F., & Villalobos Ponce, J. A. (2013). *Sistema de seguridad por reconocimiento de voz.* México.

Russell, S., & Norvig, P. (2004). *Inteligencia Artificial. Un enfoque moderno.* España: Pearson Educacion S.A.

Salcedo Cherubini, D. K., & Teixeira Gómez, A. P. (2009). *Diseño de un sistema de reconocimiento del habla para controlar dispositivos eléctricos.* Universidad Católica Andrés Bello: Venezuela.

San-José, P. P., Álvarez Alonso, E., de la Fuente Rodríguez, S., García Pérez, L., & Gutiérrez Borge, C. (2011). *Estudio sobre las tecnologías biométricas aplicadas a la seguridad.* INTECO: España.

Sigüenza Pizarro, J. A., & Tapiador Mateos, M. (2005). *Teconologías Biometricas*

aplicadas a la seguridad. Madrid: RA-MA Editorial.

Soto P., A. F., Álvarez G., C., Olavarrieta S., P., & Cañete A., L. (2012). Algoritmo para el Reconocimiento de Comandos de Voz. *Trilogía: Ciencia - Tecnologia - Sociedad*, 131-141.

UPM. (2013). *Biometría - Capacidades de I+D, soluciones tecnológicas y empresas UPM.* Universidad Politécnica de Madrid: España.

ANEXOS

Muestra	Tiempo	Energia	Rns	Zcr	Entropía	Centroide	Spread	Flatness	Roll.off	Crest	Bandwidth
Abrir.wav	0.9752	0.0420	0.2050	0.0291	11.6072	2078.4110	18375020.0000	0.2089	3046.9480	302.0244	3749.5580
Apagar.wav	0.8824	0.0174	0.1318	0.0215	11.3369	1239.0150	6090112.0000	0.0705	2131.7870	446.4070	2131.9040
Cerrar.wav	0.6966	0.1381	0.3716	0.0244	11.5172	1685.0860	12369660.0000	0.1511	2217.9200	221.7887	3089.1560
Encender.wav	0.8359	0.2096	0.4578	0.0090	11.6284	2487.6280	26890040.0000	0.2873	5006.4700	1198.3720	4547.1970

Fuente: Elaboración Propia.

1.- ABRIR

0,0464	0,0003	0,0175	0,0298	11,6252	1768,3333	10216800,0000	0,1218	3352,0020	639,0337	2650,3397
0,1858	0,0001	0,0105	0,0379	11,5432	1631,9977	7150225,3333	0,2279	2899,8047	303,3694	2146,8817
0,3251	0,0001	0,0088	0,0346	11,5854	1848,2457	11852146,6667	0,1434	3118,7260	375,8377	2880,3993
0,4644	0,0003	0,0169	0,0104	11,6463	1837,2317	15970126,3333	0,1743	3441,7233	1171,6740	3335,7523
0,6037	0,0008	0,0270	0,0089	11,6214	1448,8391	12661053,3333	0,1415	2573,2213	1102,8406	3066,9713
0,7430	0,0015	0,0372	0,0075	11,5407	1595,2395	16178886,0000	0,1770	3086,4260	1006,8462	3459,2093
0,8824	0,0217	0,1375	0,0204	11,5207	1739,8113	8340833,3333	0,1589	2282,5197	500,1531	3253,3263
1,0217	0,0364	0,1905	0,0187	11,6390	1764,6171	15394433,6667	0,1725	2928,5153	383,8883	3224,6903
1,1610	0,1371	0,3636	0,0191	11,6183	1398,8437	10093280,3333	0,1148	2655,7617	531,5585	2774,5900
1,3003	0,1265	0,3255	0,0155	11,5094	1195,8129	8012774,6667	0,0916	2390,1857	505,2687	2436,5433
1,4396	0,0023	0,0464	0,0104	11,5965	1611,8994	14473051,0000	0,1621	2716,7723	659,8780	3237,9083
1,5790	0,0010	0,0307	0,0092	11,5871	1688,0547	16243355,6667	0,1815	3072,0707	1145,9190	3507,3790
1,7183	0,0002	0,0108	0,0245	11,4711	2218,9983	20040624,3333	0,2201	4417,8957	716,5179	3683,8700
1,8576	0,0005	0,0229	0,0084	11,5747	2607,2803	28601713,3333	0,3026	5577,0997	1059,3972	4622,7883
1,9505	0,0002	0,0157	0,0129	11,5624	2382,6950	24341690,0000	0,2650	4565,0390	978,3646	4296,0260

Fuente: Elaboración Propia.

2.- APAGAR

time	energy	rms	zcr	entropy	centroid	spread	flatness	rolloff	crest	bandwidth
0.0464	0.0006	0.0237	0.0124	11.6008	1979.7363	18474341.3333	0.1963	4105.6640	946.8457	3619.8603
0.1858	0.0002	0.0110	0.0243	11.4986	1560.8127	10116923.3333	0.1231	3011.0597	572.1047	2735.6447
0.3251	0.0001	0.0078	0.0246	11.5834	1797.9143	12467124.0000	0.1483	3082.8370	608.6308	2985.5993
0.4644	0.0001	0.0093	0.0203	11.6007	1825.0047	13888803.3333	0.1682	2989.5263	620.4112	3217.5047
0.6037	0.0006	0.0212	0.0181	11.3688	1438.9647	9590864.0000	0.1081	2372.2413	361.9567	2438.1223
0.7430	0.1671	0.3999	0.0370	11.1422	1916.1403	12926911.3333	0.1506	2799.3163	278.1966	2958.6880
0.8824	0.0826	0.2563	0.0267	11.2954	1402.0887	7758093.3333	0.0911	2092.3093	279.3671	2365.5633
1.0217	0.2703	0.5194	0.0326	11.6509	1384.1510	6478878.6667	0.0750	2207.1533	244.0116	2120.7760
1.1610	0.4499	0.6642	0.0310	11.7387	1559.9413	8266793.3333	0.0960	2408.1297	641.2806	2401.0150
1.3003	0.2678	0.4876	0.0312	11.5461	1620.5017	9557777.0000	0.1111	2411.7190	278.6447	2509.7273
1.4396	0.0075	0.0747	0.0221	11.4850	1513.0067	12022346.0000	0.1395	2138.9647	251.3766	2958.0040
1.5325	0.0008	0.0286	0.0154	11.5825	1259.7750	8763469.0000	0.1081	1647.2900	912.1022	2668.6340

Fuente: Elaboración Propia.

3.- CERRAR

time	energy	rms	zcr	entropy	centroid	spread	flatness	rolloff	crest	bandwidth
0.0464	0.0076	0.0874	0.0164	11.5887	1313.7520	7553586.0000	0.0849	2332.7637	1043.2468	2267.8600
0.1858	0.0050	0.0702	0.0204	11.6112	1625.1123	11347761.3333	0.1257	2853.1497	652.1296	2729.0537
0.3251	0.0049	0.0689	0.0291	11.5266	1952.4257	15108860.0000	0.1717	3847.2660	601.1520	3294.5953
0.4644	0.0474	0.1755	0.0751	11.3426	2492.1210	17637166.6667	0.1669	5318.7013	156.3766	3329.9390
0.6037	0.1375	0.3681	0.0235	11.5798	1735.2937	14318260.0000	0.1638	2461.9630	292.3832	3228.7680
0.7430	0.2613	0.5012	0.0310	11.6103	1762.5063	11560601.6667	0.1379	2619.8733	176.5768	2784.7063
0.8824	0.1588	0.3519	0.0394	11.4838	1783.7057	11592230.0000	0.1401	2576.8067	168.9393	2888.1430
1.0217	0.0012	0.0298	0.0315	11.4869	1618.9373	10375496.6667	0.1272	2372.2413	404.7230	2790.1253
1.1610	0.0001	0.0117	0.0233	11.5942	1813.5300	14156039.3333	0.1669	3000.2933	413.0738	3165.2647
1.2539	0.0002	0.0154	0.0149	11.6704	1658.9370	13014380.0000	0.1581	2627.0510	652.5221	3190.2470

Fuente: Elaboración Propia.

4.-ENCENDER

time	energy	rms	zcr	entropy	centroid	spread	flatness	rolloff	crest	bandwidth
0.0464	0.0003	0.0168	0.0156	11.6894	1611.9373	8552032.3333	0.0948	3093.6033	1512.0693	2406.9557
0.1858	0.0002	0.0140	0.0163	11.6872	1570.1330	8880213.3333	0.1050	2906.9823	1327.9377	2477.6277
0.3251	0.0002	0.0134	0.0135	11.6856	1926.6393	15944024.3333	0.1794	3459.6683	1322.4863	3319.5620
0.4644	0.0002	0.0134	0.0145	11.6404	1606.7317	11727989.3333	0.1409	2831.6160	1222.6813	2955.3410
0.6037	0.0003	0.0174	0.0199	11.6023	1671.6280	9665760.0000	0.1156	3039.7703	1182.8928	2587.9720
0.7430	0.1615	0.3895	0.0205	11.3488	1604.8223	12589140.3333	0.1473	2709.5947	430.0728	3143.7770
0.8824	0.1220	0.3297	0.0183	11.5953	2557.7827	25815806.6667	0.2699	5566.3333	821.5346	4368.8570
1.0217	0.1778	0.4165	0.0236	11.5339	1704.4070	11151784.3333	0.1184	3039.7707	347.1996	2802.2887
1.1610	0.2180	0.4657	0.0067	11.7134	1382.1697	13336888.3333	0.1452	2077.9540	905.6019	3240.3370
1.3003	0.4092	0.6379	0.0215	11.7007	1536.3617	8738225.3333	0.1039	2670.1170	390.0416	2435.7903
1.4396	0.1353	0.3212	0.0204	11.4939	1401.0630	8471518.6667	0.0975	2228.6863	333.8131	2432.8453
1.5790	0.0014	0.0377	0.0206	11.5069	1571.7407	12018745.0000	0.1451	2555.2733	374.6023	3058.5553

Fuente: Elaboración Propia.

Anexo 1

Tiempo de Procesamiento de los Filtros LPC

Muestras	Contenido de audio	Tiempo
1	abrir	37
	apagar	28
	cerrar	25
	encender	33
2	abrir	28
	apagar	34
	cerrar	26
	encender	21
3	abrir	25
	apagar	23
	cerrar	20
	encender	22
4	abrir	25
	apagar	20
	cerrar	28
	encender	33
5	abrir	38
	apagar	28
	cerrar	33
	encender	24
6	abrir	23

Tiempo de Procesamiento de los Filtros MFCC

Muestras	Contenido de audio	Tiempo
1	abrir	98
	apagar	99
	cerrar	104
	encender	104
2	abrir	102
	apagar	99
	cerrar	100
	encender	105
3	abrir	98
	apagar	104
	cerrar	101
	encender	101
4	abrir	102
	apagar	104
	cerrar	100
	encender	103
5	abrir	98
	apagar	104
	cerrar	103
	encender	103
6	abrir	98

	apagar	27
	cerrar	28
	encender	27
7	abrir	35
	apagar	21
	cerrar	24
	encender	26
8	abrir	39
	apagar	36
	cerrar	30
	encender	23
9	abrir	38
	apagar	26
	cerrar	30
	encender	29
10	abrir	40
	apagar	21
	cerrar	40
	encender	22
11	abrir	39
	apagar	40
	cerrar	21
	encender	36
12	abrir	32
	apagar	40

	apagar	104
	cerrar	105
	encender	105
7	abrir	105
	apagar	105
	cerrar	104
	encender	105
8	abrir	98
	apagar	102
	cerrar	101
	encender	101
9	abrir	99
	apagar	101
	cerrar	98
	encender	99
10	abrir	101
	apagar	98
	cerrar	105
	encender	100
11	abrir	99
	apagar	104
	cerrar	103
	encender	98
12	abrir	99
	apagar	101

	cerrar	39
	encender	31
13	abrir	24
	apagar	40
	cerrar	26
	encender	24
14	abrir	26
	apagar	25
	cerrar	25
	encender	25
15	abrir	37
	apagar	27
	cerrar	30
	encender	37
16	abrir	23
	apagar	28
	cerrar	40
	encender	30
17	abrir	37
	apagar	38
	cerrar	30
	encender	28
18	abrir	30
	apagar	36
	cerrar	38

	cerrar	102
	encender	105
13	abrir	102
	apagar	104
	cerrar	103
	encender	103
14	abrir	103
	apagar	102
	cerrar	100
	encender	103
15	abrir	105
	apagar	100
	cerrar	101
	encender	104
16	abrir	104
	apagar	101
	cerrar	99
	encender	104
17	abrir	100
	apagar	103
	cerrar	98
	encender	104
18	abrir	102
	apagar	105
	cerrar	101

	encender	37
19	abrir	27
	apagar	39
	cerrar	27
	encender	34
20	abrir	40
	apagar	21
	cerrar	28
	encender	29
	PROMEDIO	30

	encender	98
19	abrir	104
	apagar	103
	cerrar	103
	encender	102
20	abrir	103
	apagar	102
	cerrar	98
	encender	101
	PROMEDIO	101,775

ANEXO 2

TOP: PROCESAMIENTO DE SEÑALES

CRITERIOS	Transformada de Fourier	Transformada de cosenos	Análisis Cepstrales de Mel	Análisis de Predicción Lineal
Captura de voz	x	x	x	x
Normalización de la voz	x		x	x
Transformación de tiempo a frecuencia	x	x	x	x
Segmentación en fonemas		x	x	x

Según las siguientes investigaciones:

- ✓ Gómez D.J., Simancas G.J., Acosta C.M, Meléndez P.F., Vélez Z.J.(2016) en su investigación "Algoritmo de reconocimiento de comando de voz basada en técnicas no—lineales"
- ✓ Soto P.A., Álvarez G., Olavarrieta S., y Cañete A. (2012) en su investigación "Algoritmo para el reconocimiento de comando de voz"
- ✓ Julián Camargo, Luis García, y Elvis Gaona (2012) en su investigación "Reconocimiento de voz humana aplicado a la domótica"
- ✓ Oropeza (2010) en su investigación "Algoritmos y Métodos para el reconocimiento de voz en español mediante silabas"