FECHA	OCTUBRE 22 DE 2010

NÚMERO RAE	
PROGRAMA	Ingeniería de Sonido

AUTOR (ES)	ALDANA BLANCO Andrea Lorena, PIÑEROS CASTAÑO Julián Ricardo
TÍTULO	DESARROLLO E IMPLEMENTACIÓN DE UN ALGORITMO DE RECONOCIMIENTO DE VOZ QUE PERMITA SELECCIONAR UNA IMAGEN A PARTIR DE UN BANCO DE NUEVE FOTOGRAFÍAS UTILIZANDO REDES NEURONALES

PALABRAS CLAVES

RECONOCIMIENTO DE VOZ, REDES NEURONALES, FORMANTES, RECONOCIMIENTO DE PATRONES, ENVOLVENTE DE ENERGÍA, BACKPROPAGATION, MATLAB, CAPTURA DE LA SEÑAL

DESCRIPCIÓN

El desarrollo de nuevas tecnologías ha facilitado la interacción entre las personas y los dispositivos a partir de herramientas y/o software que permitan la comunicación entre estos. El reconocimiento de voz es una de estas herramientas, ya que a partir del análisis y procesamiento de la señal se pueden establecer características particulares de la voz de una persona al decir una palabra. Esta información es interpretada por el dispositivo receptor, en este caso un computador, a través de un software desde el cual se le indica cual es el paso a seguir de acuerdo a las características reconocidas. De esta forma a través de estos comandos de voz se busca que el usuario pueda realizar tareas de la vida cotidiana de una forma diferente que le permita optimizar su tiempo.

En los procesos de reconocimiento de voz, existen 4 etapas fundamentales. La primera, es la captura de la señal en la cual se debe tener en cuenta el ruido que ingresa al sistema junto con la señal de voz para minimizarlo de la mejor forma posible sin afectar el objeto de estudio, es decir, la voz. En un segundo lugar existe un proceso de caracterización que tiene como fin extraer las características fundamentales de cada palabra para establecer las diferencias entre una y otra por cada locutor. En tercer lugar, se realiza un proceso de entrenamiento en el cual los datos tomados en la etapa de caracterización son analizados con el fin de determinar cuáles datos son característicos de una palabra en particular y de esta forma se entrena el sistema para que asocie cada palabra con unos valores específicos, además estos valores pueden ser llevados a una red neuronal u otro método con el fin de obtener un salida para cada palabra a partir de los datos ingresados. En la última etapa el sistema analiza los resultados del entrenamiento con la palabra que se quiere reconocer, si la palabra se relaciona con un patrón definido que registre el sistema, esta será reconocida.

Este algoritmo de reconocimiento es desarrollado utilizando la herramienta MATLAB y tiene como fin reconocer nueve palabras, cada una asociada a una imagen que será seleccionada cuando la palabra sea reconocida.

En este trabajo se exploran varios métodos de caracterización y entrenamiento, y finalmente se analizan los resultados para determinar con cual método se obtiene un mayor porcentaje de reconocimiento en esta aplicación.

FUENTES BIBLIOGRÁFICAS

Análisis de la señal de voz, Universidad de Extremadura, España

Articulo Sistema VLSC.
http://recedis.referata.com/wiki/Sistema_inteligente_de_reconocimiento_de_voz_para_la_traducci%C3%B3n_del_lenguaje_verbal_a_la_lengua_de_se%C3%B1as_colombiana_(VLSC**)**

Características de una Red Neuronal Artificial.
http://proton.ucting.udg.mx/posgrado/cursos/idc/neuronales2/Transferencia.htm

COLLADO, Esteban .http://poncos.freeiz.com/blog/, 2009.

MARTINEZ, Fernando; PORTALE, Gustavo; KLEIN, Hernán y OLMOS, Osvaldo. Reconocimiento de voz, apuntes de cátedra para Introducción a la Inteligencia Artificial.

MORENO, Asunción. La señal de voz, Universidad Politécnica de Cataluña, España

OROPEZA RODRIGUEZ, José Luis. Algoritmos y métodos para el reconocimiento de voz en español mediante sílabas. Computación y sistemas Vol. 9 Núm. 3, pp. 270-286, 2006.

ORTIZ RICO, Paola Milena y PARRA CARDENAS, Julie Andrea. Diseño e implementación de un Software de corrección fonética para niños con problemas de aprendizaje, 2007.

SANTOSA, Budi. Introduction to MATLAB Neural network Toolbox.

SANTOS-GARCIA, Gustavo. Inteligencia artificial y matemática aplicada, 2001

TAPIAS MERINO, Daniel. Sistemas de reconocimiento de voz en las telecomunicaciones, 1999

TREJOS POSADA, Hernando Antonio y URIBE PÉREZ, Carlos Andrés. Motor computacional de reconocimiento de voz: Principios básicos para su construcción, 2007.

Tutorial de Redes Neuronales. Universidad Tecnológica de Pereira. Facultad de ingeniería eléctrica. http://proton.ucting.udg.mx/posgrado/cursos/idc/neuronales2/

VELÁSQUEZ RAMÍREZ, Genoveva. Sistema de reconocimiento de voz, 2008

VILLORIA, Cristina. Reconocimiento y síntesis de voz.
http://observatorio.cnice.mec.es/modules.php?op=modload&name=News&file=article&sid=689

CONTENIDOS

CAPTURA DE LA SEÑAL

Implementación del algoritmo de captura de la señal de voz, exploración de resultados en distintos recintos de grabación y con diferentes micrófonos.

CARACTERIZACIÓN DE LA PALABRA
Extracción de las características fundamentales de cada palabra a través de distintos métodos.

RED NEURONAL BACKPROPAGATION
Desarrollo e implementación de una red neuronal Backpropagation para el entrenamiento de cada palabra. Determinación del número de capas ocultas y neuronas en cada una.

RECONOCIMIENTO DE PATRONES
Desarrollo e implementación del método de reconocimiento de patrones en cada una de las nueve palabras, grabación de las palabras en distintos recintos y en diferentes momentos del día, definición de los umbrales de cada uno de los formantes que caracteriza la palabra

INTERFA GRÁFICA
Realización de una interfaz gráfica que permita utilizar la aplicación de una manera más sencilla

METODOLOGÍA

1. ENFOQUE DE LA INVESTIGACION

El enfoque de esta investigación será de tipo Empírico-analítico ya que el conocimiento es construido a partir de pruebas de ensayo y error y la teoría planteada para el reconocimiento de voz, este es el punto de partida para la experimentación y desarrollo de la aplicación de acuerdo al fin de la misma y las necesidades del usuario.

2. LINEA DE INVESTIGACION DE USB / SUB-LÍNEA DE FACLTAD / CAMPO TEMATICO DEL PROGRAMA

De acuerdo a los campos de investigación determinados por la facultad de ingeniería de la Universidad de San Buenaventura sede Bogotá el proyecto está relacionado con la línea de “tecnologías actuales y sociedad” ya que es a partir de estos proyectos que se trata de dar solución a problemáticas actuales a partir de la tecnología.

En este caso la sub-línea es “análisis y procesamiento de señales” ya que se trabaja con señales de audio y es a partir de estas herramientas que se logra obtener información de la señal de voz para luego desarrollar el sistema de reconocimiento.

El campo de investigación continúa siendo “análisis y procesamiento de señales” ya que en este caso el sistema no se integra con algún tipo de hardware que permita realizar automatizaciones usando micro-controladores o tecnologías similares.

3. TÉCNICAS DE RECOLECCIÓN DE LA INFORMACIÓN

La información respecto al procesamiento de señales será obtenida de acuerdo a las guías oficiales desarrolladas para MATLAB puesto que es el software en el que se desarrollara el algoritmo, estas son obtenidas de la siguiente página web: www.mathworks.com/support.

En segundo lugar la consulta de temas relacionados con redes neuronales, procesamiento

digital de señales y reconocimiento de voz en libros y trabajos hechos previamente en esta y otras universidades del mundo.

Se extraerán los datos de la caracterización de las palabras y luego se llevaran a una hoja de cálculo con el fin de realizar pruebas y verificar que los resultados que se estén obteniendo sean verídicos y confiables. Estos valores de la caracterización de las palabras serán analizados y comparados con el propósito de lograr un mejor resultado en el reconocimiento.

4. HIPÓTESIS

Dentro de los tipos de información empleados por las tecnologías de reconocimiento de voz están los modelos acústicos, que permiten que el reconocedor identifique los sonidos pues proporcionan información sobre las propiedades y características de los mismos. Por medio del procesamiento digital de señales (DSP) en la herramienta MATLAB, se podrán encontrar estas características y de esta manera poder diferenciar las palabras para ser reconocidas.

5. VARIABLES

Variables independientes

A la hora del cálculo y extracción de las características de la señal, hay algunos factores que no pueden ser manipulados con libertad en la señal digital, como la rapidez con la que se pronuncia la palabra, pues en este caso específico, eso influye a que la división de las silabas sea impreciso.

Entorno físico ya que habrá algunos más ruidosos que otros o con distintas características acústicas. También está relacionado con la forma de hablar, la entonación y las posibles alteraciones naturales por causa de enfermedad.

Variables dependientes

Es de resaltar que los resultados estarán sujetos a variables como: el ruido de fondo y los ruidos impulsivos que se puedan presentar durante el tiempo de la captura, el algoritmo tendrá un mayor porcentaje de reconocimiento si este ha sido adecuado a una persona en específico y el tipo de micrófono utilizado, si la persona quiere hacer reconocimiento con un micrófono distinto al que se usó en la etapa inicial de extracción de formantes, el porcentaje de reconocimiento bajará.

En este caso el locutor es una variable dependiente porque es el que de cierta manera entrena el algoritmo, es decir, el número de veces que el locutor pronuncia cada una de las palabras contempla varias posibilidades y variaciones aparte de tener una forma de decirla por tratarse de un ejercicio repetitivo.

Durante el proceso de grabación y recolección de datos característicos de las palabras, es notable que para este algoritmo algunas de las palabras se pueden confundir, en especial aquellas que están formadas por silabas conteniendo las mismas vocales, por ejemplo, *gato* y *vagón*.

CONCLUSIONES

- El método de entrenamiento de la aplicación de reconocimiento de voz no debe ser determinado antes de desarrollarla ya que el desempeño del entrenamiento depende de la aplicación que se quiera implementar y a partir de las pruebas realizadas durante este proceso se podrá determinar la técnica más acertada.
- El método de reconocimiento de patrones puede ser usado en aplicaciones cuyo algoritmo de caracterización no presenta grandes diferencias en algunos de los valores extraídos de diferentes fonemas y los valores para cada hablante varían notablemente cuando se cambian las condiciones de grabación (sala, micrófono, etc.). De esta manera la determinación de umbrales permitirá obtener un mayor reconocimiento en comparación a las redes neuronales.
- El método de redes neuronales puede ser utilizado cuando el algoritmo de caracterización sea muy robusto y presente resultados realmente diferentes entre los fonemas pero mantenga unos valores relativamente constantes para cada fonema de un mismo hablante aún en diferentes condiciones de grabación. En este caso las redes neuronales podrán representar una ventaja porque aún cuando la red no ha sido entrenada con una entrada determinada se podrá producir la salida deseada ya que esta es una de las propiedades de las redes neuronales.
- En el desarrollo de esta aplicación se determinó que el método de reconocimiento de patrones permite alcanzar porcentajes de reconocimiento más altos para cada una de las nueve palabras en comparación al método de redes neuronales.
- En el caso de las redes neuronales se determinó trabajar directamente con la red Backpropagation, ya que es esta la que se menciona en los textos relacionados con este tema y así mismo es con la cual se han presentado buenos resultados en este tipo de aplicaciones.
- El algoritmo de detección de bordes con algunas modificaciones mínimas, en conjunto con el algoritmo de envolvente de energía, posibilitarían un mejor resultado al momento de separar palabras de dos silabas. En esta aplicación solo se utilizó el algoritmo de envolvente de energía ya que este era suficiente para separar las silabas en el proceso de caracterización.
- Si se decide utilizar el método de entrenamiento es necesario utilizar un gran número de entradas diferentes ya que esto aumenta notablemente el desempeño de la red. Es decir, al grabar varias veces cada palabra e ingresar estos datos a la red la probabilidad de obtener la salida deseada aumenta.
- El micrófono con el que se realiza la captura es importante para el proceso de extracción de características, ya que hay algunos que capturan más ruido que otros y esto dificulta este proceso.
- Al desarrollar una aplicación que reconozca un mayor número de palabras se necesita un algoritmo de caracterización más robusto, los resultados de esta etapa puede mejorarse añadiendo procesos a los ya utilizados, por ejemplo métodos de eliminación de ruido u otros de análisis en el dominio de la frecuencia y tiempo.
- Para poder tener un buen algoritmo y un reconocimiento por medio de redes

neuronales, es necesario tener un buen entrenamiento, esto implica tener resultados confiables en la extracción de características, se dejan como dos opciones con el método de LPC y coeficientes cepstrales.

- Es posible llegar a tener una separación de tres o más silabas realizando las investigaciones y pruebas adecuadas, de esta manera se puede lograr el reconocimiento para un mayor número de palabra a través del método de reconocimiento de patrones.
- El porcentaje de reconocimiento de la aplicación oscila entre un 70% y 80%, llegando a presentar porcentajes del 90% o 60% en algunas palabras para determinados hablantes.
- Algunas palabras tienen un mejor porcentaje de reconocimiento que otras, por esta razón el porcentaje general de la aplicación puede variar entre el 70% y 80%.
- Es posible diseñar una aplicación de reconocimiento de voz a partir de redes neuronales, pero su desempeño dependerá directamente de las necesidades de la aplicación y de la robustez del método de caracterización de la palabra. En este caso no se obtuvieron porcentajes de reconocimiento altos a partir del método de redes neuronales ya que las diferencias entre los valores de los formantes eran mínimas entre algunas palabras y por esto mismo la red tiende a confundirlas.
- Determinar el inicio y fin de una palabra o la detección de bordes es una tarea de poca importancia en ambientes controlados, casos en los que el ruido de fondo es mínimo inclusive comparándolo con los sonidos de menor nivel energético llamados fricativas o segmentos sonoros débiles.
- La interfaz gráfica permite visualizar de manera más sencilla el funcionamiento de la aplicación.

RECOMENDACIONES

- Si se desean tener resultados más precisos se pueden manejar métodos de extracción y reconocimiento más robustos.
- Para este algoritmo se pueden plantear varias mejoras: optimización del proceso de separación de silabas, extracción de características no solo en frecuencia sino también en tiempo.
- Siempre es recomendable utilizar un micrófono con patrón direccional y no los micrófonos integrados que vienen con los computadores ya que con los primeros se obtienen mejores resultados en la caracterización de las palabras y así mismo en el reconocimiento de voz.
- Es importante que cada captura de palabra se realice en campo directo ya que de esta forma no se añade información innecesaria, por ejemplo, el comportamiento acústico de una sala.
- En el desarrollo de futuros proyectos en el campo del reconocimiento de voz es recomendable limitar el número de hablantes y palabras que reconocerá la aplicación más no es recomendable limitar el método por el cual esta será desarrollada ya que durante la investigación se pueden encontrar métodos que se adapten mejor a una aplicación determinada y no a otra.
- Para obtener un mejor porcentaje de reconocimiento en un hablante

determinado es necesario entrenar el sistema para el uso de esa persona. De esta manera el sistema reconocerá las características que tienen la persona al pronunciar diferentes fonemas.

- En el momento de grabar las muestras que harán parte del entrenamiento del sistema es recomendable realizar varias grabaciones y en diferentes momentos del día o en diferentes días, esto debido a los cambios que se producen en los formantes cuando los fonemas son pronunciados de acuerdo a las distintas situaciones que hacen parte de la vida cotidiana (al despertar, al estar cansado, al estar feliz, etc.)
- Es recomendable concentrarse en un solo método de reconocimiento y extracción de características en otros espacios de investigación, por ejemplo en los proyectos integradores o grupos de investigación. De esta forma este tipo de trabajos y aplicaciones podrán ser desarrollados sobre bases más solidas, exploradas previamente con mayor profundidad.
- Es importante que la palabra que se quiera reconocer sea pronunciada de la forma más parecida posible a la utilizada durante la fase de entrenamiento.

DESARROLLO E IMPLEMENTACIÓN DE UN ALGORITMO DE RECONOCIMIENTO DE VOZ QUE PERMITA SELECCIONAR UNA IMAGEN A PARTIR DE UN BANCO DE NUEVE FOTOGRAFÍAS UTILIZANDO REDES NEURONALES

ANDREA LORENA ALDANA BLANCO

JULIAN RICARDO PIÑEROS CASTAÑO

PROYECTO DE GRADO

TUTOR:

Miguel Pérez

UNIVERSIDAD DE SAN BUENAVENTURA

INGENIERÍA DE SONIDO

BOGOTÁ, D.C.

2010

DESARROLLO E IMPLEMENTACIÓN DE UN ALGORITMO DE RECONOCIMIENTO DE VOZ QUE PERMITA SELECCIONAR UNA IMAGEN A PARTIR DE UN BANCO DE NUEVE FOTOGRAFÍAS UTILIZANDO REDES NEURONALES

ANDREA LORENA ALDANA BLANCO

JULIAN RICARDO PIÑEROS CASTAÑO

UNIVERSIDAD DE SAN BUENAVENTURA

INGENIERÍA DE SONIDO

BOGOTÁ, D.C.

2010

Agradecimientos Lorena

Quiero agradecer a mi familia por su paciencia, apoyo e incondicionalidad a través de los años y a todas las personas que hicieron parte de este proyecto.
También quiero agradecer a nuestro tutor Miguel Pérez y a Milton Mendoza asesor de TecnoParque Nodo Bogotá.
Gracias a las personas que creen, apoyan y aportan.

Agradecimientos Julián

Deseo darles las gracias a mis padres por apoyarme durante todos los años de esfuerzo, a mis compañeros de estudio que aportaron ideas al proyecto y a mi tutor de la universidad Miguel Pérez, además de agradecer a Tecnoparque Nodo Bogotá y en especial a las personas que allí nos ayudaron a sacar este trabajo adelante, entre ellos Milton Mendoza asesor Tecnoparque Bogotá.

TABLA DE CONTENIDO

Pág.

LISTA DE TABLAS

Pág.

LISTA DE FIGURAS

Pág

LISTA DE ANEXOS

Pág.

INTRODUCCION

El desarrollo de nuevas tecnologías ha facilitado la interacción entre las personas y los dispositivos a partir de herramientas y/o software que permitan la comunicación entre estos. El reconocimiento de voz es una de estas herramientas, ya que a partir del análisis y procesamiento de la señal se pueden establecer características particulares de la voz de una persona al decir una palabra. Esta información es interpretada por el dispositivo receptor, en este caso un computador, a través de un software desde el cual se le indica cual es el paso a seguir de acuerdo a las características reconocidas.

Este proyecto nace con la idea de realizar un software de reconocimiento de voz utilizando la herramienta MATLAB. Para su desarrollo fue necesario limitar el número de palabras y hablantes a reconocer, ya que los sistemas más complejos, es decir aquellos con un gran número de palabras y fase de entrenamiento de nuevos hablantes requieren algoritmos más robustos y por esto mismo un mayor tiempo de diseño y desarrollo a cargo de un grupo interdisciplinario (ingenieros, fonoaudiólogos, programadores, etc.).

Un primer paso, consistió en definir un algoritmo que permitiera establecer el inicio y fin de palabra con el fin de procesar solo la señal relevante de la misma, es decir, la voz y que a través de ciertos umbrales este determinara que parte de la grabación era ruido y por eso mismo debía eliminarse del siguiente proceso.

Posteriormente fue necesario definir el método de entrenamiento, de acuerdo a las consultas realizadas se decidió que la aplicación sería desarrollada utilizando la red neuronal Backpropagation, sin embargo aunque este método ha funcionado muy bien para una gran cantidad de aplicaciones de reconocimiento de voz, en este caso las pruebas de ensayo y error y la investigación en general llevaron a un segundo método que también fue evaluado, este es: reconocimiento de patrones, en este trabajo se exploran los dos procedimientos anteriormente mencionados.

Sin importar cual camino se tome para lograr el reconocimiento existe un paso fundamental para lograr este objetivo, esto es la caracterización de la palabra, es decir, extraer las características más importantes de cada palabra para cada hablante y que finalmente permitan diferenciar una de otra. En este trabajo se explican algunas técnicas de extracción de características, por ejemplo: FFT y formantes.

Una vez definidos el método de extracción de características y el de reconocimiento fue vital evaluar las palabras que harían parte de la aplicación final. Las pruebas realizadas permitieron establecer que hay un gran número de elementos inherentes al comportamiento del ser humano que afectan los resultados de la caracterización de la palabra. Por ejemplo, el estado de ánimo, la forma en la cual se pronuncia e inclusive el momento del día en el cual se encuentre la persona que está realizando el entrenamiento, ya que no es lo mismo decir una palabra al levantarse, al medio día o antes de ir a dormir. Por otro lado las consultas y pruebas efectuadas permitieron definir que en lo posible las nueve palabras que reconocería la aplicación deberían contener diferentes consonantes y vocales entre ellas, esto haría la tarea de reconocimiento más fácil.

Por último se realizaron las grabaciones con distintos micrófonos (diadema e internos de computadores portátiles) y una vez analizados los resultados se estableció un micrófono con el cual se realizaría el entrenamiento y con el cual se probaría la aplicación una vez finalizada. Además, se diseñó una interfaz gráfica que permitiera ver de manera más fácil el funcionamiento de la aplicación.

El proyecto se lleva a cabo mediante el desarrollo e implementación de un algoritmo de voz que permita reconocer nueve palabras a partir de un banco de nueve imágenes previamente establecidas. Se utiliza el método de formantes para la caracterización se la señal y los métodos de redes neuronales y reconocimiento de patrones para el reconocimiento, esto con el fin de determinar cuál es el procedimiento más apropiado para esta aplicación. Con este trabajo se pretenden establecer las bases de un sistema de reconocimiento de voz para que esta información pueda contribuir en el desarrollo de futuros proyectos relacionados con el tema.

1. PLANTEAMIENTO DEL PROBLEMA

1.1. ANTECEDENTES

En la década de los años 50's se desarrolló un sistema capaz de reconocer 10 dígitos, este fue implementado por los laboratorios Bell y fue la base de los sistemas de reconocimiento de voz que se conocen en la actualidad. En ésta misma década en MIT Lincoln Lab desarrollaron un programa que reconocía vocales pronunciadas por distintos hablantes. Así mismo de comenzó a tener en cuenta la normalización temporal no lineal, es decir, la capacidad de comparar los parámetros de dos palabras iguales pronunciadas a velocidad distinta esto se consiguió al determinar el inicio y fin de palabra.

A partir de este momento la creación de aplicaciones crece de forma acelerada, siendo Estados Unidos y Japón los precursores en el desarrollo de nueva tecnología y nuevos métodos de reconocimiento de voz. Es así como en los 70's se presenta el primer producto de reconocimiento de voz, el VIP100 de Treshold Technology Inc, éste sistema funcionaba con un vocabulario reducido, palabras discretas y era dependiente del locutor.

El siguiente objetivo fue crear un sistema que reconociera un número determinado de palabras cuando éstas fueran pronunciadas de forma continua y por distintos hablantes.
Por esto mismo se emprende un proyecto muy grande conocido como ARPA SUR (Advanced Research Projects Agency- SpeechUnderstanding Research / Agencia de Proyectos de Investigación Avanzados – Investigación con el fin de Entender el Habla), el cual pretendía lograr ese objetivo y aunque no se cumplió a cabalidad fue un trabajo que dejó muchos aportes que permitieron continuar con el desarrollo de proyectos.

En los 80's y los 90's se crearon aplicaciones que cumplían con esos objetivos planteados en la década anterior. Se desarrollaron aplicaciones con fines comerciales y corporativos, las cuáles consistían principalmente en sistemas de dictado que le permitían a las empresas ahorrar tiempo. Además se mejoraron los modelos ya existentes permitiendo que una aplicación reconociera un mayor número de palabras de forma aislada o continua con un porcentaje de error cada vez menor. Las compañías y productos más representativos del mercado con respecto al reconocimiento de voz son:

- Philips
- Dragon Systems
- IBM
- Speechworks
- Vocalis
- Novell
- Microsoft

Reconocimiento de voz en Colombia

De la misma forma en que se desarrollaban aplicaciones con reconocimiento de voz en el mundo, se comenzaron a crear en Colombia, inicialmente desde los trabajos en investigación de distintas Universidades, como lo son: Universidad de Manizales, Universidad del Valle, Universidad nacional, el Politécnico Colombiano, Universidad de San Buenaventura, entre otras.

En el caso de la Universidad de San buenaventura, se realizó una tesis en el 2007 cuyo nombre es: Diseño e implementación fonética para niños con problemas de aprendizaje, ésta aplicación determina si la pronunciación de la consonantes G y K y así mismo las palabras Gato y Koala es correcta o no. El reconocimiento de los fonemas y las palabras de realizó a través de la plataforma MATLAB utilizando la red neuronal Backpropagation. En el 2008 MATLAB implementó dentro de su plataforma una aplicación de nombre Neural Network Toolbox la cual hacía más sencilla la programación de redes neuronales permitiendo que el proceso de entrenamiento fuera más rápido.

En este mismo año se desarrolló una Tesis en la Universidad tecnológica de Pereira de Nombre: "Motor Computacional de reconocimiento de voz: Principios básicos para su construcción". En este trabajo el objetivo es establecer las bases necesarias para el desarrollo de un motor computacional de reconocimiento de voz por esto mismo se realiza una larga investigación con respecto a los métodos de extracción de características y entrenamiento de las mismas, finalmente se decide trabajar con la FFT y las redes neuronales. Posteriormente se estableció C++ como lenguaje de programación por su eficiencia y facilidad en la programación.

En el año 2008 Carolina Bernal Villamaría, egresada de la Universidad pedagógica Nacional desarrolló el sistema VLSC "Sistema inteligente de reconocimiento de voz para la traducción del lenguaje verbal a la lengua de señas colombiana ", este sistema fue desarrollado utilizando la plataforma MATLAB y tenía por objetivo realizar el reconocimiento de un cierto número de palabras y a partir de la información reconocida utilizar la herramienta gráfica de la plataforma para mostrar la imagen del lenguaje de señas correspondiente.

1.2. DESCRIPCION Y FORMULACION DEL PROBLEMA

El reconocimiento de voz se ha posicionado como una herramienta útil en diferentes procesos, día a día la tecnología nos va llevando a mejorar los porcentajes de reconocimiento en aplicaciones más complejas.

En Colombia existe gente capacitada para desarrollar aplicaciones adaptadas a las necesidades del país. Por esto mismo es necesario establecer las bases teóricas y prácticas que nos permitan tener un crecimiento acelerado en el desarrollo de sistemas que funcionen por reconocimiento de voz y que sean útiles para la gente de nuestro país y del mundo.

Dada la situación previamente analizada, nace la necesidad de plantear la siguiente pregunta: ¿Es posible explorar los distintos métodos planteados para el desarrollo de aplicaciones de reconocimiento de voz con el fin de implementar un algoritmo de reconocimiento de voz en el que se permita la selección de una imagen a partir de un banco de nueve fotografías y que posteriormente ayude a desarrollar otras aplicaciones de reconocimiento de voz?

1.3JUSTIFICACION

Los sistemas de reconocimiento de voz pueden facilitar el desarrollo de ciertas actividades a partir de la interacción entre el usuario y el computador. Su diseño depende directamente de las funciones que vaya a tener el sistema y así mismo el método bajo el cual se diseña la aplicación depende del porcentaje de reconocimiento que se quiera alcanzar.

El diseño de estas aplicaciones es viable ya que no representa un alto costo y por el contrario puede brindar beneficios a los usuarios en varios campos: laboral, comercial, estudiantil, etc. También es necesario tener en cuenta que el usuario final solo necesita la aplicación, un computador y un micrófono para poder acceder a los beneficios de esta tecnología.

Este proyecto es un punto de partida en el desarrollo de aplicaciones por reconocimiento de voz. En este se plantean conceptos y modelos que han sido estudiados e implementados en diversos sistemas y se busca definir el modelo más adecuado de acuerdo a las necesidades planteadas en esta aplicación.

Está dirigido a aquellas personas que deseen desarrollar aplicaciones de reconocimiento de voz y que busquen un punto de partida para llevar a cabo sus proyectos.

1.4 OBJETIVOS DE LA INVESTIGACION

1.4.1 Objetivo general

Desarrollar un algoritmo de reconocimiento de voz que permita seleccionar una de nueves imágenes de un banco preestablecido.

1.4.2 Objetivos específicos

- Elaborar un algoritmo que permita capturar una señal de audio para su posterior procesamiento.
- Desarrollar un algoritmo que calcule el inicio y el fin de palabra.
- Determinar cuál red neuronal puede funcionar mejor para el desarrollo de la aplicación.
- Diseñar el algoritmo de entrenamiento de la red neuronal Backpropagation.
- Realizar la interfaz gráfica de la aplicación.

1.5 ALCANCES Y LIMITACIONES DEL PROYECTO

El desarrollo de este algoritmo permite ejecutar una acción a través de comandos de voz en español, seleccionar una imagen a partir de un banco de nueve fotografías. Hasta este punto se desarrolla solo la parte de software, pero podría implementarse con un hardware en comunicación con MATLAB para poder realizar acciones tales como encender y apagar luces, abrir y cerrar puertas o ventanas. Comandos básicos de actividades cotidianas realizadas en el hogar o lugar de trabajo.

El proyecto estará limitado a 9 palabras y sus caracterización con el fin de poder distinguirlas entre sí (extracción de formantes), temas adicionales serán expuestos con brevedad sin llegar a profundizar en ellos.

2 MARCO TEORICO

2.1 LA HERRAMIENTA MATLAB

MATLAB es la abreviatura de MATrix LABoratory o laboratorio de matrices en español, es un software de cálculo numérico que cuenta con su propio lenguaje de programación (lenguaje M), un lenguaje de alto nivel desarrollado para mejorar la respuesta computacional en tareas intensivas más rápido que los lenguajes tradicionales como C y C++.Sus principales prestaciones son el manejo de matrices, vectores, funciones, implementación de algoritmos, desarrollo de interfaces graficas e usuario y procesamiento digital de señales en general.

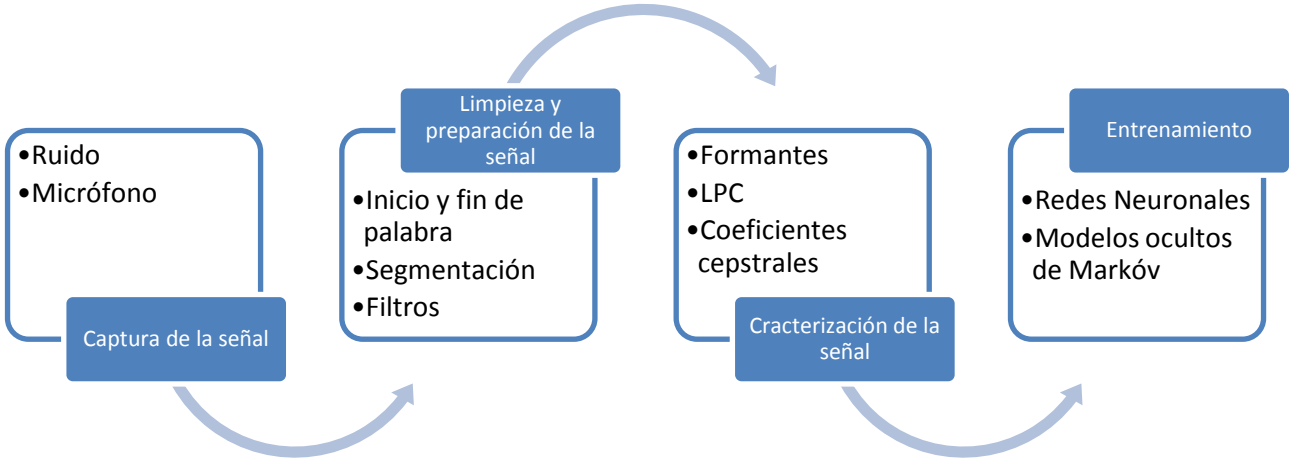
2.2 RECONOCIMIENTO DE VOZ

El reconocimiento de voz es una herramienta de comunicación entre un usuario y un computador. El porcentaje de reconocimiento depende del número de locutores, el número de palabras, el estilo del habla, entre otros.

Todos los días se desarrollan nuevos métodos y algoritmos que permitan llegar al sistema ideal, es decir, aquel que funcione de igual manera para cualquier locutor independientemente de los cambios con los cuales este pronuncie los fonemas y que tenga un muy alto porcentaje de reconocimiento sin importar el ruido propio del recinto, el micrófono y los canales de comunicación.

A continuación se presenta un esquema ordenado de los procesos que se tienen en cuenta en un sistema de reconocimiento de voz.

Figura 1: Captura, procesamiento y caracterización de la señal de voz. Entrenamiento.



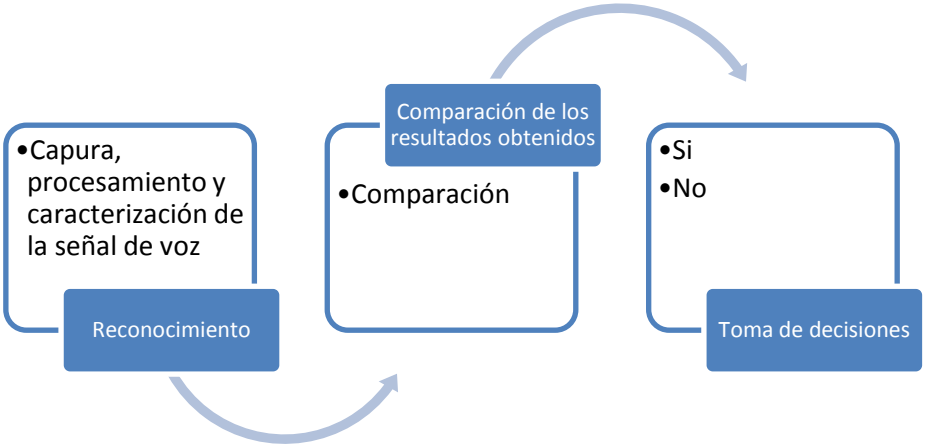
Inicialmente se realiza la captura de la señal de voz. Dos de los factores que influyen en la calidad de la muestra son el micrófono con el cual esta es tomada y el ruido de fondo del recinto o el ruido producido por los canales de comunicación.

Una vez se haya grabado la palabra, se determina el inicio y fin de la misma. Posteriormente se realiza una segmentación en la cual cada fragmento suele ser menor a 30 ms ya que en estos pequeños segmentos se puede ver el comportamiento periódico de la señal de voz. En algunos casos es necesario aplicar filtros pasa banda entre los 300 Hz y 3400 Hz aproximadamente, eso se

hace con el fin de contar solo con la señal de la voz y eliminar información en frecuencias que no sean relevantes en el análisis de la señal de voz.
Un tercer paso consiste en caracterizar la señal. Un vector de audio puede contener miles de componentes de acuerdo a la duración de la palabra que se quiere reconocer. La caracterización de la señal permite extraer parámetros que disminuyan el número de elementos que ingresan a la etapa de entrenamiento y que además sean representativos del comportamiento de la palabra. Los métodos de formantes, LPC y coeficientes cepstrales entre otros permiten caracterizar la señal y dan como resultado un número limitado de valores correspondientes a cada palabra que son utilizados en la etapa de entrenamiento.

En la fase de entrenamiento se cuenta con distintos modelos matemáticos como lo son las Redes Neuronales y los Modelos ocultos de Markóv los cuales permiten establecer patrones del comportamiento de cada palabra para que de esta forma el sistema asocie esos patrones a una palabra específica.

Figura 2: Etapa de reconocimiento y toma de decisiones



En el proceso de reconocimiento se realiza nuevamente el proceso de captura, procesamiento y caracterización de la señal de voz y el resultado es comparado con los patrones previamente establecidos en el sistema. A partir de esta comparación el sistema tomará una decisión y determinará si la palabra fue o no fue reconocida.

2.3 REDES NEURONALES

Las redes neuronales son modelos matemáticos con los cuales se pretende obtener una respuesta de acuerdo al funcionamiento del sistema neuronal biológico. Tanto en las neuronas biológicas como en las neuronas artificiales se cuenta con una entrada y una salida que a su vez está conectada a la entrada de otra neurona.

2.3.1 Funcionamiento de una neurona biológica

Figura 3: Red Neuronal Biológica

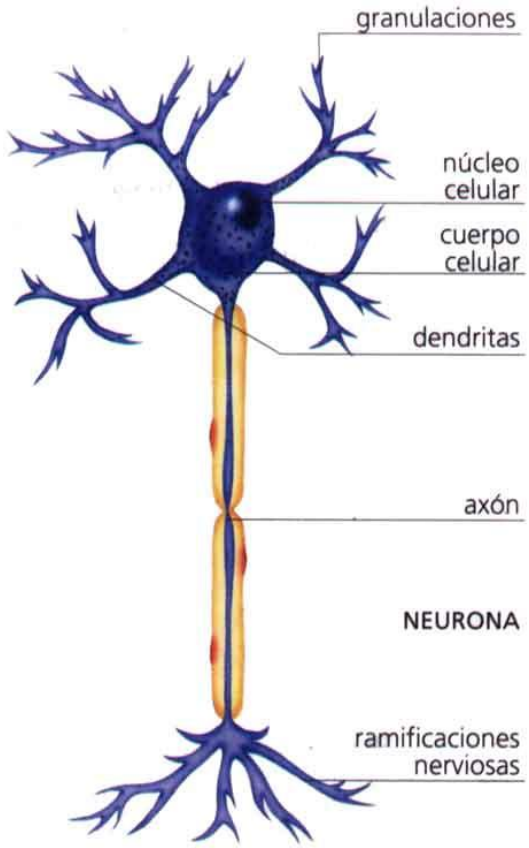


Figura tomada de: http://alejandrofatouh.blogspot.com/2010/07/neuronas-que-reconocen-famosos.html

Las neuronas biológicas cuentan con las siguientes partes:

1. Cuerpo: En el cuerpo se encuentra el núcleo, el cual recibe la información procedente de otras neuronas a partir de las dentritas y es allí donde se controlan las actividades de toda la célula.
2. Ramas de extensión conocidas como dentritas: Las dentritas Son ramificaciones encargadas de recibir la información procedente de otras célula a partir de unas conexiones conocidas como sinápticas.
3. Un Axón: El Axón es la salida de la neurona y se encarga de enviar impulsos a otras neuronas.

Un proceso químico se encarga de transmitir una señal de una neurona a otra generando en el receptor un potencial eléctrico que puede aumentar o disminuir de acuerdo a la información recibida. Una vez este potencial alcanza el umbral establecido la neurona continua transmitiendo esa información a través del Axón. En el caso de las redes neuronales artificiales este comportamiento está dado por una función de transferencia y es a partir de esta función que se transmite una señal de salida dependiendo de la entrada.

2.3.2 Ventajas de las redes neuronales artificiales

Una de las grandes ventajas de las redes neuronales es su etapa de *Aprendizaje*. En esta etapa se le indica cuál debe ser la salida a una entrada determinada y de esta manera la red "aprende" cuál es la salida esperada. Por otro lado cabe destacar que la fase descrita anteriormente le permite a la red ser más flexible y realizar el reconocimiento aun cuando la señal que está siendo comparada tenga un poco de ruido.

2.3.3 Funciones de transferencia de las redes neuronales artificiales

Función Sigmoidal

En esta función los valores de la salida oscilan entre 0 y 1, mientras que la entrada puede tomar valores entre más infinito y menos infinito.

Figura 4: Función de transferencia Log-Sigmoid

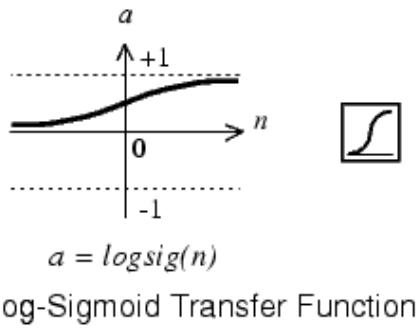


Figura tomada de: Neural Network Toolbox MATLAB Architecture Manual.

Esta dada por la ecuación:

$$a = \frac{1}{1 + e^{-n}} \tag{2.1}$$

Función Tangente – Sigmoidal

En esta función los valores de la salida oscilan entre -1 y 1, mientras que la entrada puede tomar valores entre más infinito y menos infinito.

Figura 5: Función de transferencia Tan-Sigmoid

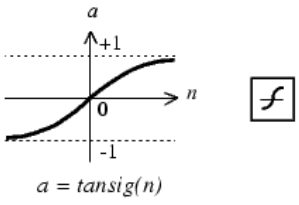


Figura tomada de: Neural Network Toolbox MATLAB Architecture Manual.

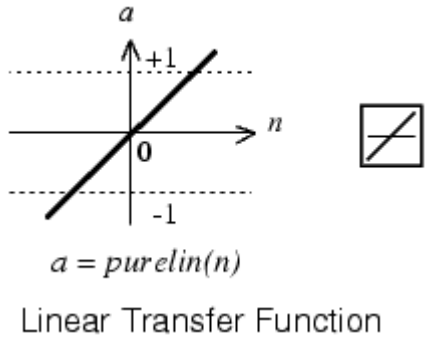
Esta dada por la ecuación:

$$a = \frac{e^{n} - e^{-n}}{e^{n} + e^{-n}} \tag{2.2}$$

Función de transferencia lineal

En este tipo de función de transferencia la salida es igual a la entrada

Figura 6: Función de transferencia lineal.



Figuratomada de: Neural Network Toolbox MATLAB Architecture Manual.

Esta dada por la ecuación:

$$a = n \tag{2.3}$$

2.3.4 Red Neuronal Backpropagation

La Red Neuronal Backpropagation, en español (propagación hacia atrás), es un método de entrenamiento de redes neuronales. En este sistema se cuenta con una capa de entrada, una de salida y una o varia capas ocultas. Al estimular la red por medio de un patrón de entrada esta información se propaga a través de las capas de la red y genera una salida, esta salida es comparada con la salida *esperada* y a partir de estos valores se calcula un *error* para cada una de las salidas. Cada vez que se realice el proceso descrito anteriormente la red habrá hecho una iteración.

Una vez se ha calculado el error para cada salida, estos valores se propagan hacía atrás, desde la salida hacía las capas ocultas permitiendo que cada neurona de la capa oculta tenga conocimiento de su aporte al error total generado en la salida y de esta forma se actualizan los pesos de conexión de cada neurona, esto con el fin de que la red converja hacia donde se quería inicialmente para poder clasificar correctamente la información.

Siguiendo este orden de ideas un sistema de entrenamiento Backpropagation cuenta con los siguientes pasos:

1. Generar los pesos de las conexiones de manera aleatoria
2. Introducir los valores de la capa de entrada

3. Determinar los valores de la salida deseada
4. Comparar la salida obtenida con la salida deseada
5. Realizar el cálculo del error en cada salida con el fin de actualizar los pesos de las conexiones entre neuronas.

Estructura de la red Backpropagation

Figura 7: Estructura de la red neuronal Backpropagation.

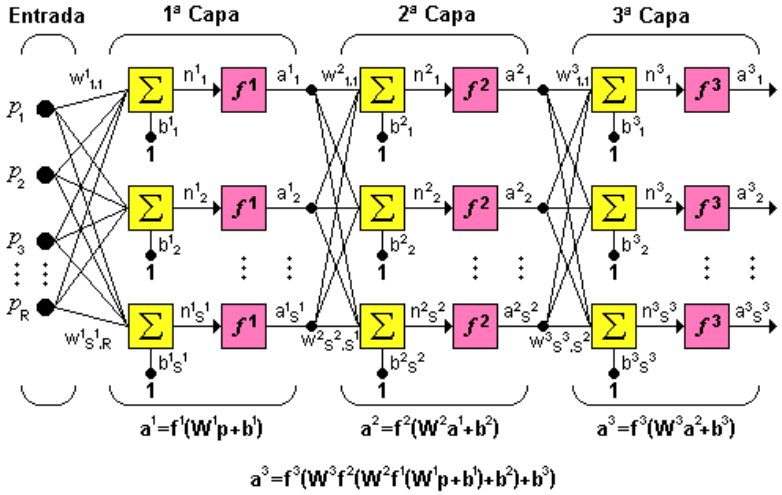


Figura tomada de: Neural Network Toolbox MATLAB Architecture Manual.

A partir de la figura 7 se puede observar que en esta red se tienen tres capas. El número de neuronas en cada capa puede ser independiente de igual manera la función de transferencia puede ser di tinta en cada capa.

Cada elemento en la figura tiene la siguiente notación:

W= Valor de los pesos de la conexiones
R= Número de elementos en el vector de entrada
S = Número de neuronas en la capa
a = Salida de la neurona

Regla de Aprendizaje

En la red Backpropagation la actualización de los pesos depende del *error medio cuadrático* (Ver sección 2.4.4). En este caso el aprendizaje es supervisado, es decir, se debe indicar a la red durante el entrenamiento la salida deseada ante cada patrón de entrada.

$$\{p_{1,}t_{1,}\}\{\mathrm{p}_2,\mathrm{t}_2\},\ldots,\{\mathrm{p}_\mathrm{Q},\mathrm{t}_\mathrm{Q}\} \tag{2.4}$$

Con base en la ecuación 2.4se puede establecer que p_Q es la entrada y t_Q es la q-ésima salida deseada para esa entrada.

Para iniciar el entrenamiento se presenta un q-ésimo número de entradas.

$$P = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_i \\ \vdots \\ p_q \end{bmatrix}$$

(2.5)

Este se propaga a través de las conexiones existentes produciendo una salida que ingresará a las neuronas de la siguiente capa.

$$n_j^o = \sum_{i=1}^{q} W_{ji}^o p_i + b_j^o$$

(2.6)

W= Valor de los pesos de la conexiones
P= Componente del vector que contiene el patrón de entrenamiento
b = Ganancia de la neurona
n = Salida de la neurona/Entrada a la siguiente neurona

La salida de las neuronas de la capa oculta está dada por la ecuación:

$$a_j^o = f^o\left(\sum_{i=1}^{q} W_{ji}^o p_i + b_j^o\right)$$

(2.7)

Donde,

$$f = \ Función\ de\ transferencia\ de\ la\ capa\ oculta$$

En la ecuación anterior se puede observar que la salida de la capa oculta es la entrada a la capa de salida.

$$n_k^s = \sum_{j=1}^{m} W_{kj}^s a_j^o + b_k^s$$

(2.8)

La salida final de la red se describe en la siguiente ecuación:

$$a_k^s = f^s\left(n_k^s\right)$$

(2.9)

En la siguiente ecuación cada salida producida en cada neurona es comparada con la salida deseada. Esto permite calcular el error en la salida de cada neurona.

$$\delta_k = \left(t_k - a_k^s\right)$$

(2.10)

El error producido por cada patrón de entrada está dado por la ecuación:

$$ep^2 = \frac{1}{2}\sum_{k=1}^{s}(\delta_k)^2$$

(2.11)

Donde,

$ep^2 = Error\ cuadrático\ medio\ para\ cada\ patrón\ de\ entrada$

El error total está dado por:

$$e^2 = \sum_{p=1}^{r} ep^2$$

(2.12)

El error para las neuronas que se encuentran en la capa de salida está dado por la siguiente ecuación:

$$\delta_k^s = \left(t_k - a_k^s\right) f'^s\left(n_k^s\right)$$

(2.13)

En el proceso de propagación inversa se busca encontrar el error de la capa oculta bajo la siguiente ecuación:

$$-\frac{\partial ep^2}{\partial W_{ji}^o} = -\frac{\partial}{\partial W_{ji}^o}\left(\frac{1}{2}\sum_{k=1}^{l}\left(t_k - a_k^s\right)^2\right) = \sum_{k=1}^{l}\left(t_k - a_k^s\right)\times\frac{\partial a_k^s}{\partial W_{ji}^o}$$

(2.14)

Inicialmente se actualizan los pesos de la capa oculta a partir de la siguiente ecuación:

$$W_{ji}\left(t+1\right) = W_{ji}\left(t\right) - 2\alpha\delta_j^o p_i$$

(2.15)

Después se actualizan las ganancias de la capa oculta a partir de la siguiente ecuación:

$$b_j\left(t+1\right) = b_j\left(t\right) - 2\alpha\delta_j^o$$

(2.16)

2.4 RED NEURONAL BACKPROPAGATION EN MATLAB

MATLAB a partir de la versión del año 2008 incluye entre sus herramientas una denominada "Neural Network Toolbox™ (Caja de herramientas de Redes Neuronales)". Utilizando esta herramienta se pueden crear diferentes tipos de redes neuronales. En el caso de la red neuronal Backpropagation, existen una serie de parámetros con los cuales esta puede ser creada, entrenada y simulada para determinar su salida y comportamiento, estos se presentan a continuación.

2.4.1 Sintaxis de una red neuronal Backpropagation en MATLAB

La sintaxis de una red neuronal Backpropagation en MATLAB es la siguiente:

net = newff([S1 S2...SNi],{TF1 TF2...TFNi},BTF,)

Donde,

SNi: Número de neuronas para cada una de las capas (entrada, salida, ocultas).
TFNi: Función de transferencia de cada una de las capas. Si no se especifica se utiliza la función sigmoidal (*tansig*) por defecto
BTF: Algoritmo de entrenamiento. Si no se especifica se utiliza *trainlm* por defecto

En el Toolbox de redes neuronales existen varios algoritmos de entrenamiento con diferentes características. A continuación se describen tres de estos algoritmos:

Algoritmo Trainbfg:

Algoritmo alternativo que emplea la técnica del gradiente conjugado, su expresión matemática se deriva del método de Newton, con la ventaja de que no es necesario computar las segundas derivadas; este algoritmo requiere más capacidad de almacenamiento que el algoritmo tradicional, pero generalmente converge en menos iteraciones. Requiere de un cálculo aproximado de la matriz Hessiana, esta es una matriz cuadrada n x n de las segundas derivadas parciales de las variables n. Suele ser usada en la resolución de problemas de optimización cuando hay funciones de varias variables. En el caso del algoritmo Trainbfg es de dimensiones n2 x n2, donde n es la cantidad de pesos y ganancias de la red; para redes que involucren una gran cantidad de parámetros es preferible emplear el algoritmo trainrp.[1]

Algoritmo Traingdm:

Este algoritmo cuenta con un coeficiente de Momentum que influye en la actualización de los pesos. En el momento en que el error de la nueva iteración exceda el error de la iteración anterior en un valor mayor al determinado por el usuario los nuevos pesos y ganancias no serán tomados en cuenta y el coeficiente de Momentum tomará el valor cero.

Estos son dos de los parámetros que se pueden definir al utilizar el Algoritmo Traingdm:

función*max_perf_inc*y: Máximo error que puede tomar la nueva iteración
net.trainParam.mc: Valor fijado para el coeficiente de Momentum.

Algoritmo Trainrp:

Las redes multicapa, utilizan típicamente una función de transferencia sigmoidal en las capas ocultas, estas funciones comprimen un infinito rango de entradas, dentro de un finito rango de salidas, además se caracterizan porque su pendiente tendera cada vez más a cero, mientras más grande sea la entrada que se le presenta a la red, esto ocasiona problemas cuando se usa un algoritmo de entrenamiento de pasos descendientes, porque el gradiente empieza a tomar valores muy pequeños y por lo tanto no habrán cambios representativos en los pesos y las ganancias, así se encuentren bastante lejos de sus valores óptimos. El propósito del algoritmo Backpropagation Resileint (RPROP) es eliminar este efecto en la magnitud de las derivadas parciales. En este algoritmo solamente el signo de la derivada es utilizado para determinar la dirección de actualización de los parámetros, la magnitud de las derivadas no tiene efecto en la actualización. La magnitud en el cambio de cada peso es determinada por separado.

Estos son dos de los parámetros que se pueden definir al utilizar el Algoritmo Trainrp:

net.trainParam.delt_inc: Valor del incremento de los pesos y ganancias
net.train.delt_dec: Valor del decremento de los pesos y ganancia

Así la derivada del error de los pesos haya cambiado de signo con respecto a la anterior iteración; si la derivada es cero, entonces el valor actualizado se conserva; si los pesos continúan cambiando en la misma dirección durante varias iteraciones, la magnitud de cambios de los pesos se decrementa.[2]

[1] http://ohm.utp.edu.co/neuronales/Anexos/AnexoA.htm
[2] http://ohm.utp.edu.co/neuronales/Anexos/AnexoA.htm

Existe otro parámetro que debe ser tenido en cuenta ya que una vez los datos son ingresados a la red estos son divididos en datos de entrenamiento, de validación y de test que por defecto tienen valores de 60%, 20% y 20% respectivamente. La función *netgrifo.divideFcn* = ''; evita está división y puede ser útil en aquellos casos donde la entrada tiene pocas neuronas y por lo tanto es mejor no realizar la división de los datos.

2.4.2 Creación y entrenamiento de la red

El siguiente comando permite crear una red neuronal Backpropagation con 20 neuronas en la capa oculta. La entrada son cuatro números pares cuya salida se espera que sean cuatro números uno.

net = newff(entrada,salida,20);

Donde,

entrada *= [2 4 8 10] (Acá se presenta la entrada de la red, cuatro entradas significan 4 neuronas en la capa de entrada)*
Salida *= [1 1 1 1] (Esta es la salida deseada, 4 valores de salida significan 4 neuronas en la capa de salida)*
net *= En esta variable se cargará la red creada*
newff *= Este comando crea la red neuronal Backpropagation*

El siguiente comando permite entrenar la red:

net = train(net,entrada,salida);

Donde,

net *= Esta es la variable de la red previamente creada*
train *= Comando que permite entrenar la red*

2.4.3 Simulación de la red

El siguiente comando permite simular la red previamente creada y entrenada. Al simular la red se obtienen unos valores en la salida que serán comparados con la salida deseada para determinar el error

y = sim(net,p);

sim = comando que simula la red una vez esta ha sido entrenada
net *= Esta es la variable de la red previamente creada*
p = Valores que se ingresan a la red una vez terminado el entrenamiento para comprobar el desempeño de la misma. Por ejemplo P = [4 8 10 12]
Y= salida de la red que ha sido entrenada

2.4.4 Error cuadrático medio

El error cuadrático medio se define como el cuadrado de la diferencia entre el estimador δ(X) y el parámetro a estimar q (θ)

$$\mathbf{ECM_{\theta}(\delta) = E(\delta(X) - q(\theta))^2}$$

Dónde,

δ(X) es un estimador y q(θ) el parámetro a estimar

El error de una red neuronal permite determinar si el tipo de red, el algoritmo de entrenamiento y el número de neuronas en la capa oculta son la mejor opción para resolver el problema planteado. A partir del cálculo del error cuadrático medio se puede determinar el desempeño de una red.

Inicialmente se debe establecer la diferencia entre la salida deseada y la salida obtenida, esto da como resultado un error. A partir de estos valores se utiliza el comando *mse* para determinar el error cuadrático medio de la red y poder verificar si este es realmente tan bajo como se espera.

```
error = salida-Y
ERROR= mse(error);
```

mse = comando para calcular el error cuadrático medio

Estos procesos se repiten para cada entrada que hará parte del entrenamiento de la red y al ser este un aprendizaje supervisado el creador de la red deberá prestar atención al error cuadrático medio para determinar si la estructura actual de la red funciona o debe ser modificada.

2.5 RECONOCIMIENTO DE PATRONES

En las últimas décadas el reconocimiento de voz se ha centrado en el desarrollo de teorías y técnicas de implementación por computador para tareas concretas de reconocimiento, Hoy en día no hay una teoría unificada utilizable para todo tipo de problemas, la mayoría de las técnicas están orientadas a cada problema.

En este método se puede dividir el problema en tres fases: adquisición de datos, análisis de datos y clasificación por decisión. La primera etapa es la digitalización del sonido, en este caso la captura por medio del micrófono, en la segunda etapa se tratan los datos para obtener un conjunto de características y la tercera fase es realmente la de reconocimiento, formado por un conjunto de funciones de decisión que permiten clasificar un sonido a partir de las características del mismo.

Figura 8: Esquema de un sistema de reconocimiento del habla

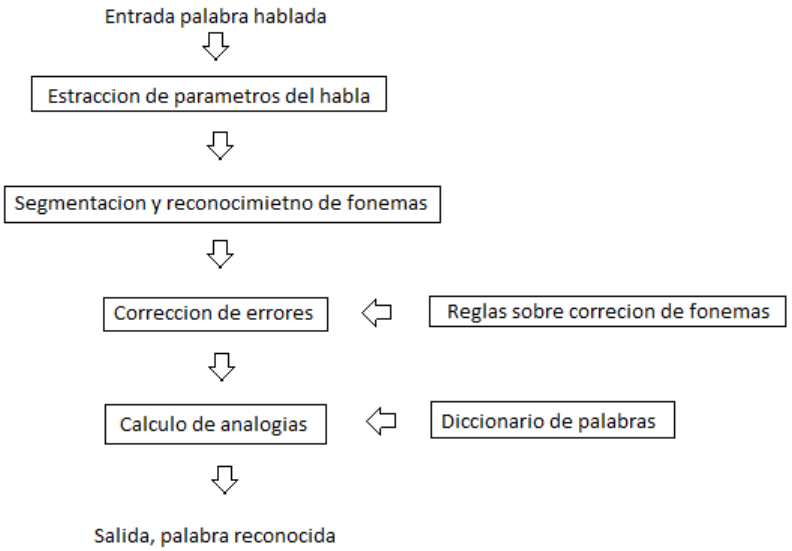
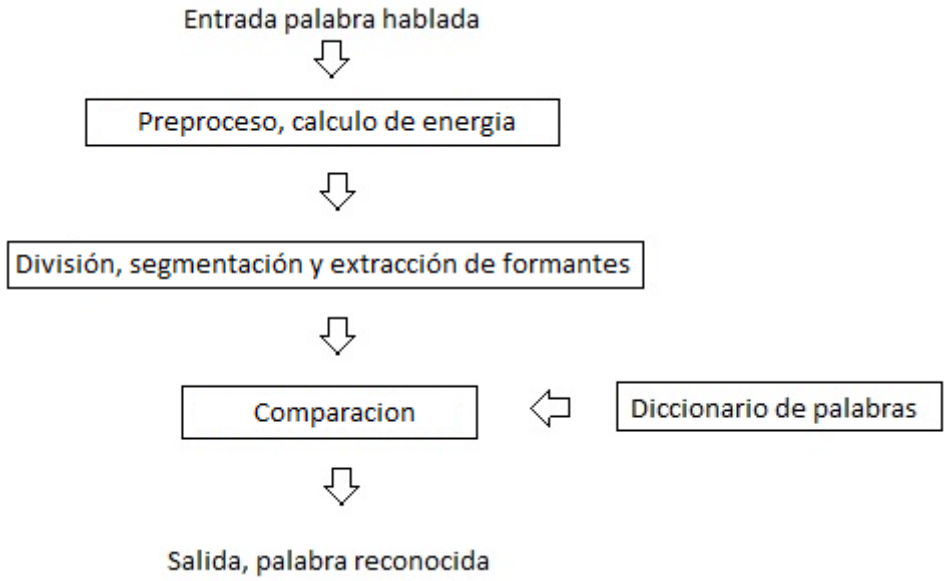


Figura tomada del libro “Inteligencia Artificial y Matemática Aplicada: Reconocimiento Automático del Habla”

Tanto la elección de características como del clasificador son decisión del diseñador del sistema. En la **figura8** se muestra el esquema básico de un sistema de reconocimiento del habla. En la **figura 9** se presenta el esquema del algoritmo utilizado en este trabajo.

Figura 9: Esquema del sistema de reconocimiento del habla implementado.



De modo que en una percepción practica "*Reconocimiento de patrones es el área de la ciencia de la computación e ingeniería que estudia conceptos, algoritmos e implementaciones que proveen a los sistemas artificiales de la capacidad perceptual de representar objetos abstractos, o patrones, en categorías, de un modo fiable y simple*"[3]

2.6 EL TRACTO VOCAL

El tracto vocal está conformado por tres cavidades acústicas y los órganos que las conforman, estas son:

- Cavidad faríngea: La cual se ubica después de la laringe.
- Cavidad nasal: Está formada por el paladar, la lengua, los dientes y los labios
- Cavidad Oral: Se ubica entre el velo del paladar y los orificios nasales.

Cuando se emiten sonidos la laringe es la encargada de excitar estas cavidades produciendo lo que se conoce como: Formantes. Estos formantes son concentraciones de energía que se producen en determinadas frecuencias.

En el tracto vocal se producen resonancias ya que al vibrar las cuerdas vocales estás producen ondas sonoras que son filtradas de acuerdo a las características del tracto vocal y finalmente atenúan o amplifican ciertas frecuencias.

[3] CÉSAR LLAMAS BELLO – VALENTÍN CARDEÑOSO PAYO. Reconocimiento automático del habla: técnicas y aplicación. Valladolid : Secretariado de publicaciones e intercambio científico universidad de Valladolid, 1997. P 19. Serie: CIENCIAS, n 16

Figura 10: Estructura del tracto vocal

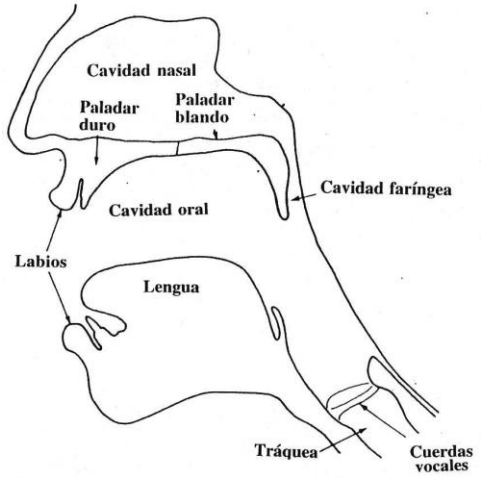


Figura tomada de: http://liceu.uab.es/~joaquim/phonetics/fon_produccio/articulacion.html

Figura 11: Órganos que constituyen el tracto vocal

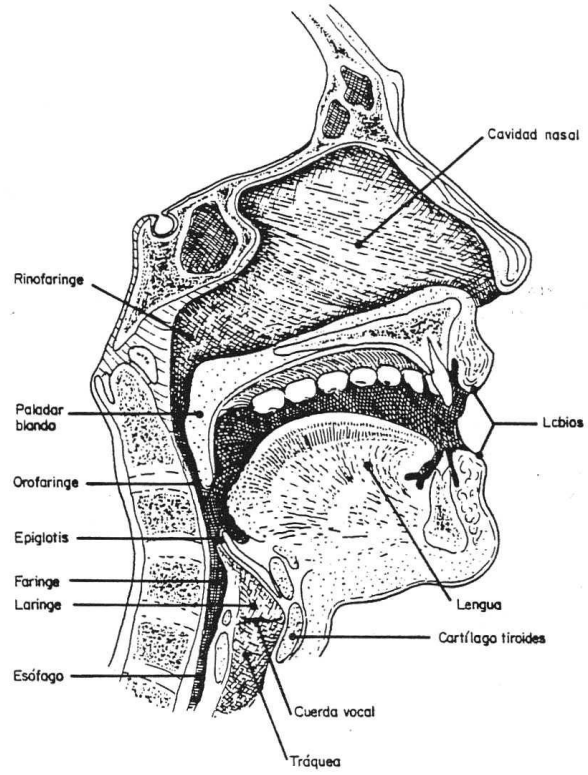


Figura tomada de: http://liceu.uab.es/~joaquim/phonetics/fon_produccio/articulacion.html

2.7 ESPECTRO DE FRECUENCIA

Así como la luz tiene un rango de frecuencias dentro del espectro electromagnético que puede ser percibido por el ojo humano conocido también como luz visible, el sentido de la audición de los seres humanos también posee un rango de frecuencias que puede percibir o escuchar, este rango se extiende desde los 20Hz a los 20.000Hz aproximadamente, siendo los sonidos de menor frecuencia los más graves como el sonido de un bajo eléctrico o un trombón, y los de mayor frecuencia los más graves como los sonidos producidos por los platillos de la batería, el chillido de un bebe o una mujer gritando.

De modo que el espectro en frecuencia lo que muestra es, la amplitud de uno o varios sonidos en la escala de los decibeles "dB" a lo largo del rango de frecuencias audibles por el ser humano. En este proyecto, la frecuencia de muestreo será de 8000Hz para cubrir el rango comprendido entre 0Hz y 4000Hz, suficiente para la transmisión de un mensaje hablado.

2.7.1 Decibel

El decibel es una unidad logarítmica empleada en acústica para expresar la relación entre dos magnitudes, la magnitud que se estudia y la magnitud de referencia, donde la magnitud que se estudia es lo que se está midiendo y la de referencia el umbral de audición humana, 0 dB equivale a una presión de 20 micro pascales, nivel en el cual el oído humano empieza a percibir sonidos. En la escala de los decibeles también se tiene un umbral de dolor, nivel al cual el oído empieza a tener la sensación de dolor, este umbral esta alrededor de los 120 dB, nivel alcanzado por los aviones en marcha o despegando.

2.7.2 Frecuencia de muestreo

Es el número de muestras por segundo que se toman de una señal continua para formar una señal discreta, en el proceso de pasar una señal de análoga a digital, de esta manera si se graba una señal de audio a 8000Hz de frecuencia de muestreo, significa que en un segundo se tendrán 8000 puntos de distintas amplitudes representando un segundo de la señal de audio, es por esta razón que entre mayor sea la frecuencia de muestreo mejor será la calidad del audio, porque se estará representando con mayor cantidad de puntos por segundo y por lo tanto tendrá mayor similitud con la señal análoga.

En las aplicaciones de reconocimiento de voz con una frecuencia de 8000Hz es suficiente para lograr el reconocimiento, este rango está determinado por el teorema de Nyquist, el cual habla de que para poder replicar una forma de onda es necesario que la frecuencia de muestreo sea superior al doble de la frecuencia que se quiere muestrear, como con el rango de 0Hz a 4000Hz es suficiente para el entendimiento de un mensaje hablado, de acuerdo a este teorema la frecuencia de muestreo será mínimo de 8000Hz.

2.8 FORMANTES

Los formantes responden a la configuración específica de los órganos propios del tracto vocal cuando se desea pronunciar un determinado sonido. De esta forma, cada sonido diferenciado conllevará una envolvente espectral característica de dicho sonido y diferenciadora respecto al resto de sonidos.[4]

[4]AGNITIO. Manual de usuario Batvox Basic 3.1. 2009. P 125

Figura 12: Espectro de un sonido y los sus formantes

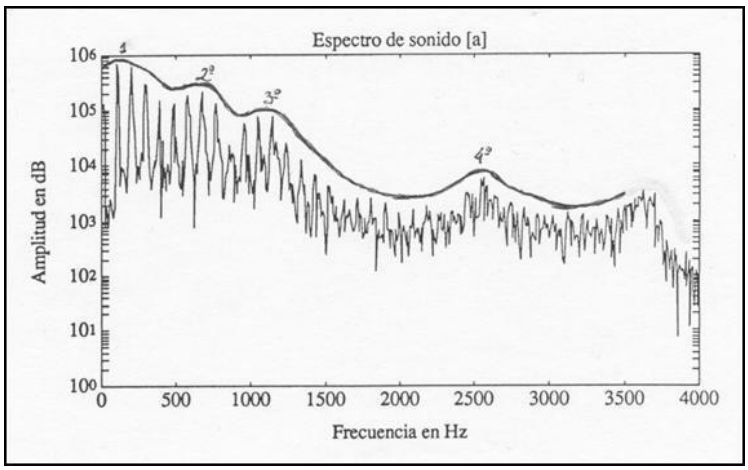


Figura tomada de: Batvox Basic 3.1: Manual de Usuario

Suelen representarse como picos de intensidad en el espectro de un sonido, dado que son concentraciones de energía en determinadas frecuencias que suelen darse en el proceso del habla por la resonancia producida en el tracto vocal. Los formantes son generalmente usados para la distinción de los sonidos del habla humana, en especial las vocales y los sonidos sonoros.

2.9 SONIDOS SORDOS Y SONOROS

Básicamentelos sonidos sonoros son los que hacen vibrar las cuerdas vocales mientras que los sonidos sordos no, por esta razón los sonidos sonoros como los de las vocales y algunas consonantes tienen una forma de onda cuasi periódica, pero los sonidos sordos por el contrario presentan formas de onda desordenadas asemejándose a la forma de onda del ruido.

Un buen ejemplo son los sonidos producidos por las vocales y consonantes como la letra g y s, sonoros y sordos respectivamente. Y se puede comprobar colocando las yemas de los dedos en la "manzana de Adán" al mismo tiempo que se producen los sonidos.

En la **figura 13** se muestra la forma de onda de un sonido sonoro y otro sordo.

Figura 13: Señal de audio con segmentos sonoros y sordos

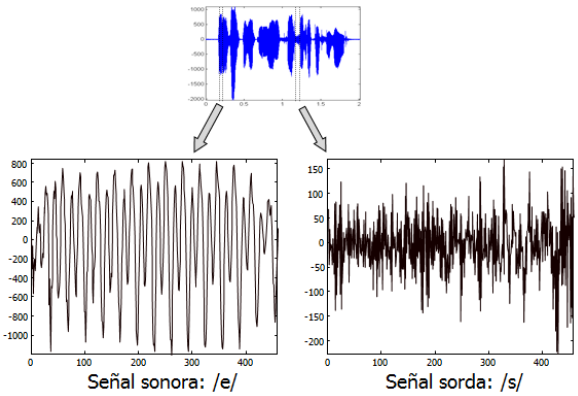


Figura tomada de "La señal de voz de Asunción Moreno. Universidad Politécnica de Cataluña".

2.10 SEGMENTACION O ENVENTANADO

La segmentación es un proceso en el cual se divide la señal en unidades más pequeñas para ser analizadas, y se utilizan ventanas en cada segmento. La función de las ventanas es disminuir la discontinuidad de información al principio y fin de los segmentos analizados ayudados del solapamiento entre las ventanas para atenuar aún más la pérdida de información, los tipos de ventanas más conocidas son: **rectangular** esta posee un valor de 1 para todo el intervalo de la ventana, de lo contrario es cero, **hanning** la característica de esta ventana es que atenúa la señal en los bordes del segmento, **hamming** es similar a la ventana de hanning pero tiene un comportamiento en frecuencia diferente.

Las ventanas más comunes y sus diferencias son mostradas en las siguientes figuras.

Figura 14: Ventana de Hann y su respuesta en frecuencia.

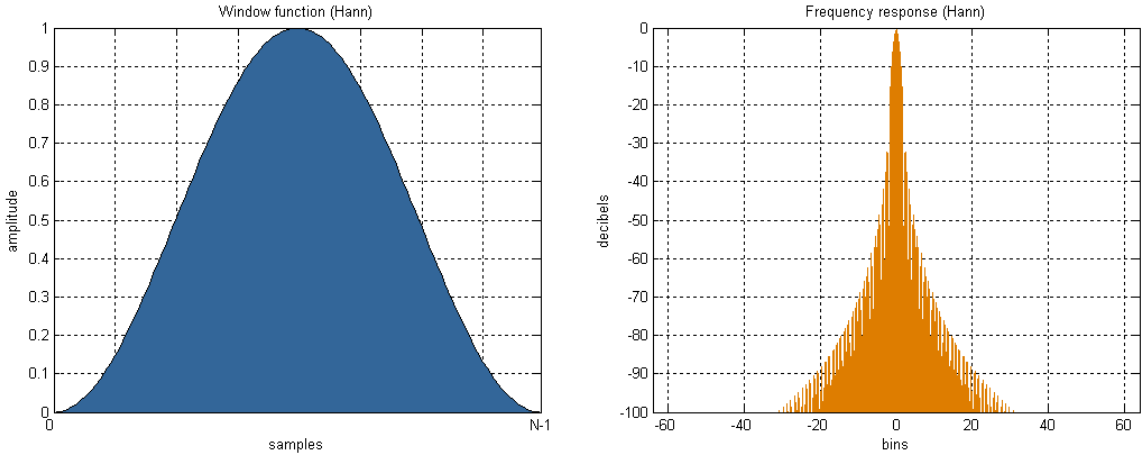
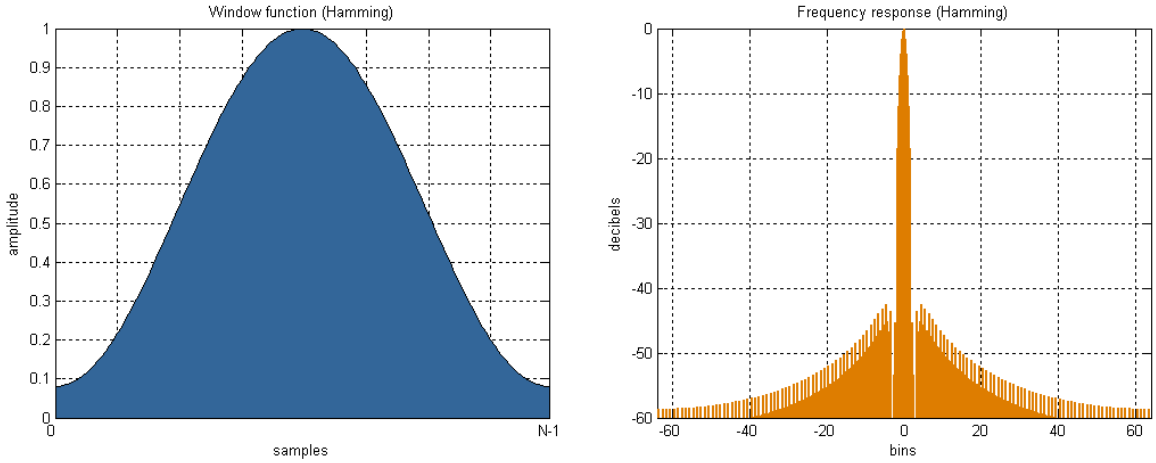


Figura 15: Ventana de Hamming y su respuesta en frecuencia.



2.11 NORMALIZAR

Es el proceso en el cual se incrementa o disminuye la amplitud de una señal entera con el fin de que el pico máximo de amplitud este en el nivel deseado.

2.12 RUIDO DE FONDO

Se entiende por ruido de fondo los sonidos no deseados mezclados con la señal útil al momento de realizar una medición acústica o una captura de señal, llegando a alterar los resultados esperados. Por ejemplo en una entrevista al aire libre, lo importante es capturar el diálogo de las personas involucradas, será ruido de fondo las voces de las personas que pasen alrededor de ellos, el sonido de los carros y sonidos de la calle.

2.13 RELACIÓN SEÑAL RUIDO

Es la relación en dB existente entre el nivel de la señal útil y el nivel del ruido de fondo.

Figura 16: Relación señal ruido.

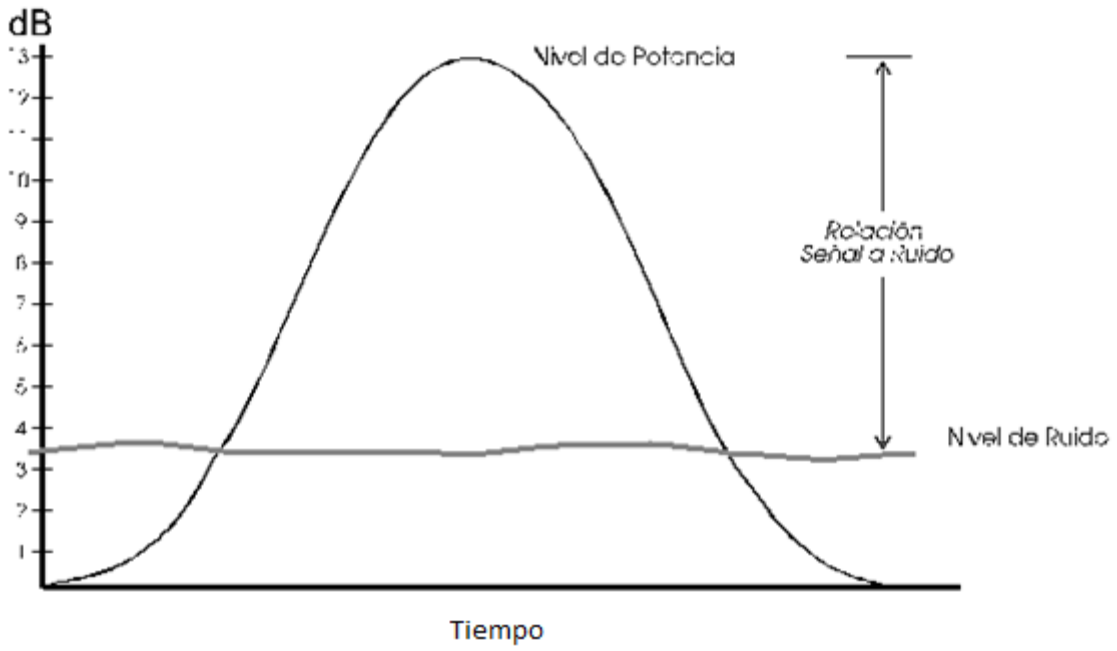


Imagen tomada de http://www.eveliux.com/mx/relacion-senal-a-ruido-snr.php

Tomando como ejemplo la **figura 16** se aprecia un nivel de ruido de fondo entre 3 y 4 dB, mientras que el nivel de la señal útil es de aproximadamente 13 dB, en este caso la diferencia será de 10 dB aproximadamente y esa será la relación señal ruido.

2.14 ENERGÍA EN TIEMPO CORTO

La energía en tiempo corto de una señal, en inglés *Short Time Energy*, es un método matemático bajo el cual se puede determinar en qué punto de la señal captada existe señal de voz y en qué punto no. Esto se hace analizando la señal al dividirla en pequeñas muestras, proceso similar al de enventanado.

2.15 UMBRAL

Un umbral es una punto clave de una escala en el cual un sistema puede comenzar o dejar de trabajar, un ejemplo es el umbral de presión sonora al que empieza a trabajar el oído humano es de 20 micro pascales, equivalentes a 0 dB. De la misma manera el ojo solo puede percibir cierto rango de frecuencias electromagnéticas.

2.16 COEFICIENTES CEPSTRALES

Los coeficientes cepstrales también conocidos como Mel Frequency Cepstrum Coefficients (MFCC) son coeficientes que representan el comportamiento del habla basándose en la percepción auditiva humana a partir de una escala logarítmica conocida como escala Mel. Según la escala Mel un tono de 1000Hz a 40 dB es comparable con 1000 Mels.

2.17 LPC

El codificador de predicción lineal en inglés Linear Predictive Coding parte de la idea de que la voz puede ser modelada ya que con una segmentación del orden de los milisegundos esta se comporta de forma periódica, por lo tanto se puede predecir el comportamiento del resto de la señal a partir de muestras anteriores. Así que a través de este método al tener una señal de habla se puede definir la función de transferencia del filtro que la generó, es decir, el tracto vocal.

2.18 REVERBERACIÓN

La reverberación es un fenómeno acústico que se produce cuando las reflexiones del sonido original permanecen en el espacio una vez este ha sido apagado. El tiempo de reverberación T60, es el tiempo en segundos que se demora en disminuir el sonido 60 dB cuando la fuente original de sonido ha sido apagada.

2.19 MATRIZ HESSIANA

La Matriz Hessiana es la Matriz cuadrada n x n de las segundas derivadas parciales. Suele ser usada en la resolución de problemas de optimización cuando hay funciones de varias variables.

2.20 CRUCES POR CERO

Un cruce por cero es el punto en el cual una onda o señal cambia de polaridad o signo.

Figura 17: Ejemplo de cruces por cero

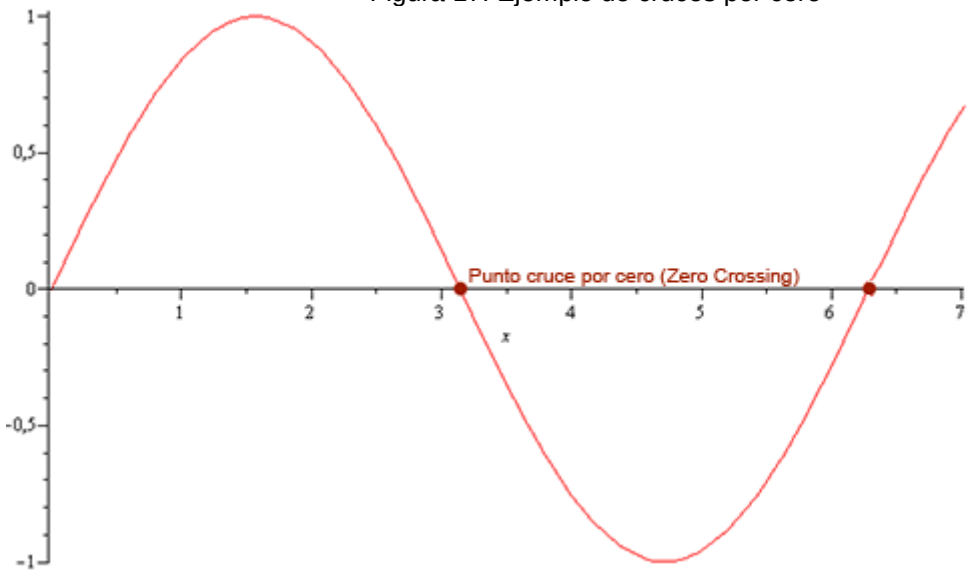
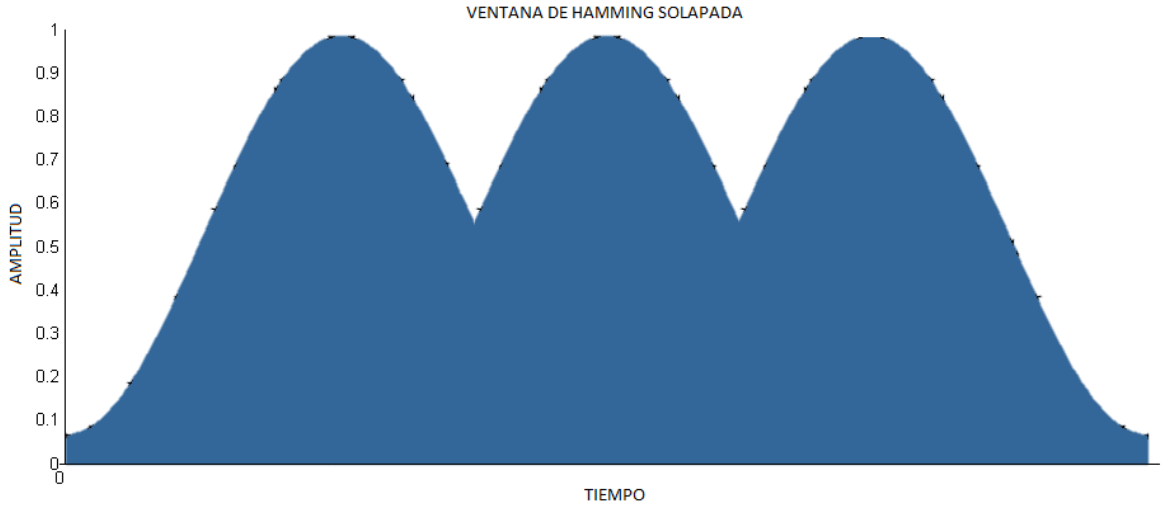


Figura tomada de: Proyecto domótica X10 chile

2.21 SOLAPAR

Solapar se refiere a poner una cosa sobre la otra, cubrir una cosa con otra parcial o totalmente.

Figura 18: Ventanas de hamming solapadas en un 50%



2.22 RUIDO IMPULSIVO

Un ruido impulsivo se caracteriza por tener un incremento repentino en la intensidad en un periodo corto de tiempo, en otras palabras son aquellos ruidos que concentran grandes cantidades de energía en instantes cortos de tiempo. Algunos ejemplos de ruidos impulsivos son: un globo al estallar, un disparo, una palmada o un aplauso.

2.23 INTERFAZ GRAFICA DE USUARIO (GUI)

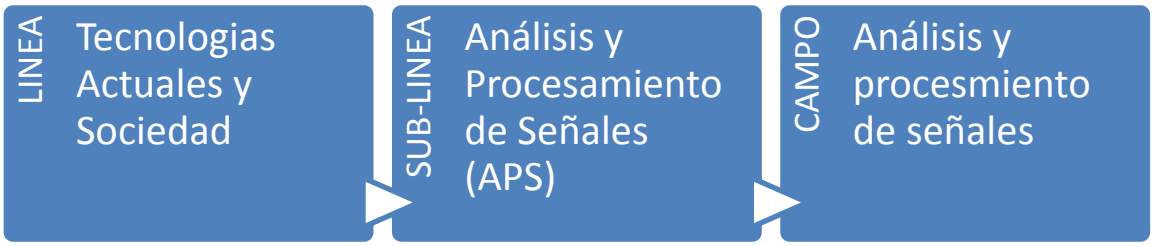
Una interfaz gráfica de usuario o (Graphical User Interface en inglés), es un programa que le facilita al usuario el uso de un algoritmo por medio de objetos texto e imágenes. Es un entorno visual con el propósito de hacer más amigable la comunicación entre una persona y un computador.

3 METODOLOGIA

3.1 ENFOQUE DE LA INVESTIGACION

El enfoque de esta investigación será de tipo Empírico-analítico ya que el conocimiento es construido a partir de pruebas de ensayo y error y la teoría planteada para el reconocimiento de voz, este es el punto de partida para la experimentación y desarrollo de la aplicación de acuerdo al fin de la misma y las necesidades del usuario.

3.2 LINEA DE INVESTIGACION DE USB / SUB-LÍNEA DE FACLTAD / CAMPO TEMATICO DEL PROGRAMA



De acuerdo a los campos de investigación determinados por la facultad de ingeniería de la Universidad de San Buenaventura sede Bogotá el proyecto está relacionado con la línea de "tecnologías actuales y sociedad" ya que es a partir de estos proyectos que se trata de dar solución a problemáticas actuales a partir de la tecnología.

En este caso la sub-línea es "análisis y procesamiento de señales" ya que se trabaja con señales de audio y es a partir de estas herramientas que se logra obtener información de la señal de voz para luego desarrollar el sistema de reconocimiento.

El campo de investigación continúa siendo "análisis y procesamiento de señales" ya que en este caso el sistema no se integra con algún tipo de hardware que permita realizar automatizaciones usando micro-controladores o tecnologías similares.

3.3 TECNICAS DE RECOLECCION DE LA INFORMACION

La información respecto al procesamiento de señales será obtenida de acuerdo a las guías oficiales desarrolladas para MATLAB puesto que es el software en el que se desarrollara el algoritmo, estas son obtenidas de la siguiente página web: www.mathworks.com/support.

En segundo lugar la consulta de temas relacionados con redes neuronales, procesamiento digital de señales y reconocimiento de voz en libros y trabajos hechos previamente en esta y otras universidades del mundo.

Se extraerán los datos de la caracterización de las palabras y luego se llevaran a una hoja de cálculo con el fin de realizar pruebas y verificar que los resultados que se estén obteniendo sean verídicos y confiables. Estos valores de la caracterización de las palabras serán analizados y comparados con el propósito de lograr un mejor resultado en el reconocimiento.

3.4 HIPÓTESIS

Dentro de los tipos de información empleados por las tecnologías de reconocimiento de voz están los modelos acústicos, que permiten que el reconocedor identifique los sonidos pues proporcionan información sobre las propiedades y características de los mismos. Por medio del procesamiento digital de señales (DSP) en la herramienta MATLAB, se podrán encontrar estas características y de esta manera poder diferenciar las palabras para ser reconocidas.

3.5 VARIABLES

3.5.1 Variables independientes

A la hora del cálculo y extracción de las características de la señal, hay algunos factores que no pueden ser manipulados con libertad en la señal digital, como la rapidez con la que se pronuncia la palabra, pues en este caso específico, eso influye a que la división de las silabas sea imprecisa.

Entorno físico ya que habrá algunos más ruidosos que otros o con distintas características acústicas. También está relacionado con la forma de hablar, la entonación y las posibles alteraciones naturales por causa de enfermedad.

3.5.2 Variables dependientes

Es de resaltar que los resultados estarán sujetos a variables como: el ruido de fondo y los ruidos impulsivos que se puedan presentar durante el tiempo de la captura, el algoritmo tendrá un mayor porcentaje de reconocimiento si este ha sido adecuado a una persona en específico y el tipo de micrófono utilizado, si la persona quiere hacer reconocimiento con un micrófono distinto al que se usó en la etapa inicial de extracción de formantes, el porcentaje de reconocimiento bajará.

En este caso el locutor es una variable dependiente porque es el que de cierta manera entrena el algoritmo, es decir, el número de veces que el locutor pronuncia cada una de las palabras contempla varias posibilidades y variaciones aparte de tener una forma de decirla por tratarse de un ejercicio repetitivo.

Durante el proceso de grabación y recolección de datos característicos de las palabras, es notable que para este algoritmo algunas de las palabras se pueden confundir, en especial aquellas que están formadas por silabas conteniendo las mismas vocales, por ejemplo, *gato* y *vagón*.

4. DESARROLLO INGENIERIL

Este capítulo está dividido en tres secciones, la primera explica los procesos generales que son comunes entre los dos métodos experimentados, redes neuronales y reconocimiento de patrones (**numerales 4.1 a 4.3**). La segunda sección expone el proceso de experimentación con el método de redes neuronales (**numeral 4.4**). Por último la sección que muestra el desarrollo del método escogido que es el de reconocimiento de patrones, con todos los pasos del algoritmo, desde la captura de la señal hasta el reconocimiento de la palabra e interfaz gráfica, (**numerales 4.5 y 4.6.**).

4.1 CAPTURA DE LA SEÑAL

Las funciones de la herramienta MATLAB (versión utilizada, MATLAB 7.9 R2009b) para la obtención de datos de un micrófono y=wavrecord(2*Fs,Fs,1), y almacenamiento de esos datos wavwrite(y,Fs,'pruebadell.wav'), hacen de la captura de la señal un proceso sencillo, pero se debe prestar especial atención al transductor de entrada que se está usando, puesto que en el mercado hay micrófonos de más calidad que otros y que capturen más ruido que otros, y esto resulta ser relevante a la hora de separar la palabra en silabas.

4.1.1 Problemas encontrados y su solución.

A continuación se muestran los resultados de las pruebas realizadas con dos micrófonos, uno es el que viene con el computador portátil y el otro es un micrófono de diadema. La prueba consistió en estimar la respuesta en frecuencia y capturar el ruido de fondo con los dos micrófonos. Para ello se utilizó el software Smaart 6.0, en las **figuras 19 y 20** se observa la respuesta en frecuencia del micrófono del computador (Dell Studio XPS, MicrophoneArray, IDT High Definition Audio CODEC) y la del micrófono de diadema (Audífonos Genius HS-04S Diadema). Ambos capturando ruido rosa al mismo nivel.

Figura 19: Respuesta en frecuencia del micrófono del computador (Dell Studio XPS, MicrophoneArray, IDT High Definition Audio CODEC)

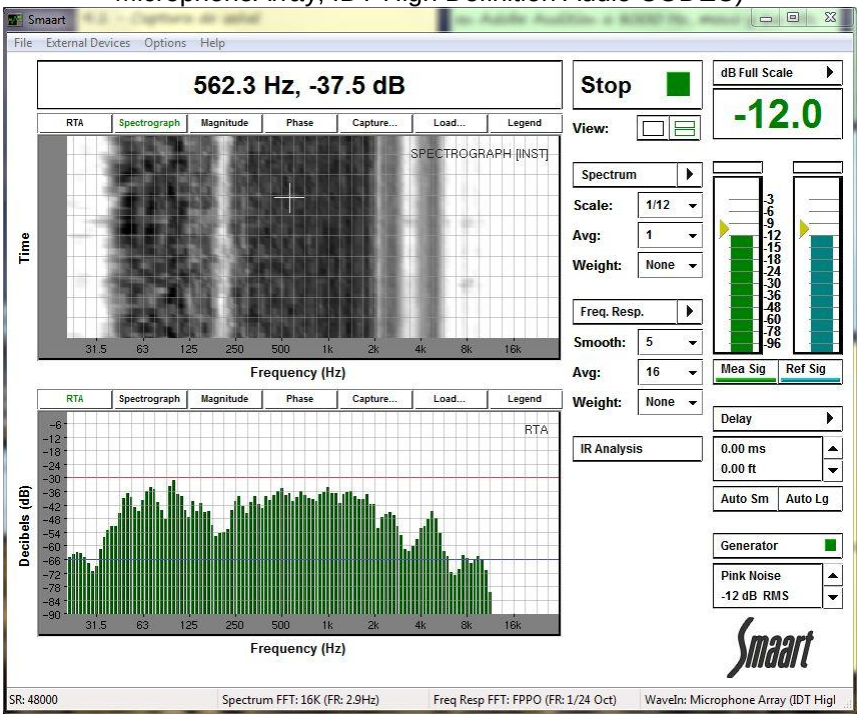


Figura 20: Respuesta en frecuencia del micrófono de diadema (Audífonos Genius HS-04S Diadema).

Las **Figuras 19** y **20** muestran los dos micrófonos cumplen con algunos requerimientos mínimos de respuesta en frecuencia, por lo menos de 250Hz a 4000Hz. Pero ahí no radica el problema, el

problema está en el ruido que cada uno de los micrófonos capta, la comparación se realizó observando la respuesta en frecuencia de los dos micrófonos con el mismo ruido de fondo, los resultados se presentan en las siguientes figuras.

Las especificaciones técnicas del micrófono de diadema ofrecidas por el fabricante son las siguientes. Las del micrófono del computador no son brindadas por la empresa fabricante y no se conoce la marca de los micrófonos que posee.

Audífonos Genius HS-04S Diadema
Micrófono:
Frecuencia de respuesta de micrófono: 50Hz - 20Hz
Impedancia de micrófono: 2.2KOhm
Sensibilidad de micrófono: -60dB * 4dBB
Micrófono con filtro: de cancelación de ruidos - Control de volumen incluido.

Figura 21: Ruido de fondo captado con el micrófono del computador.

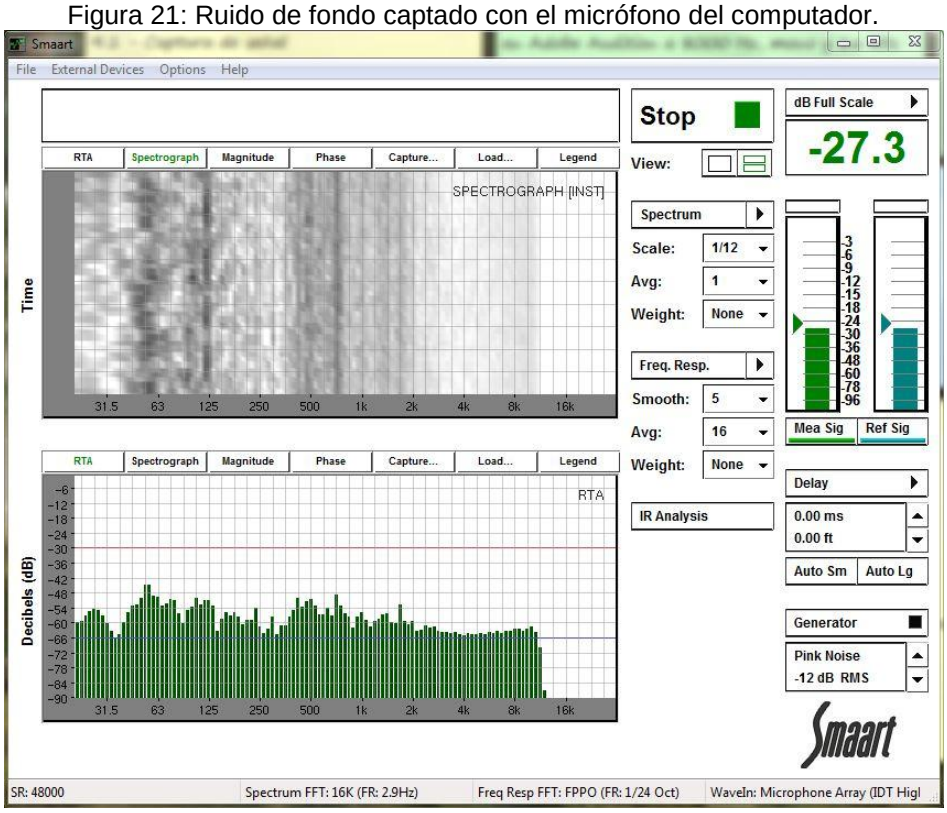
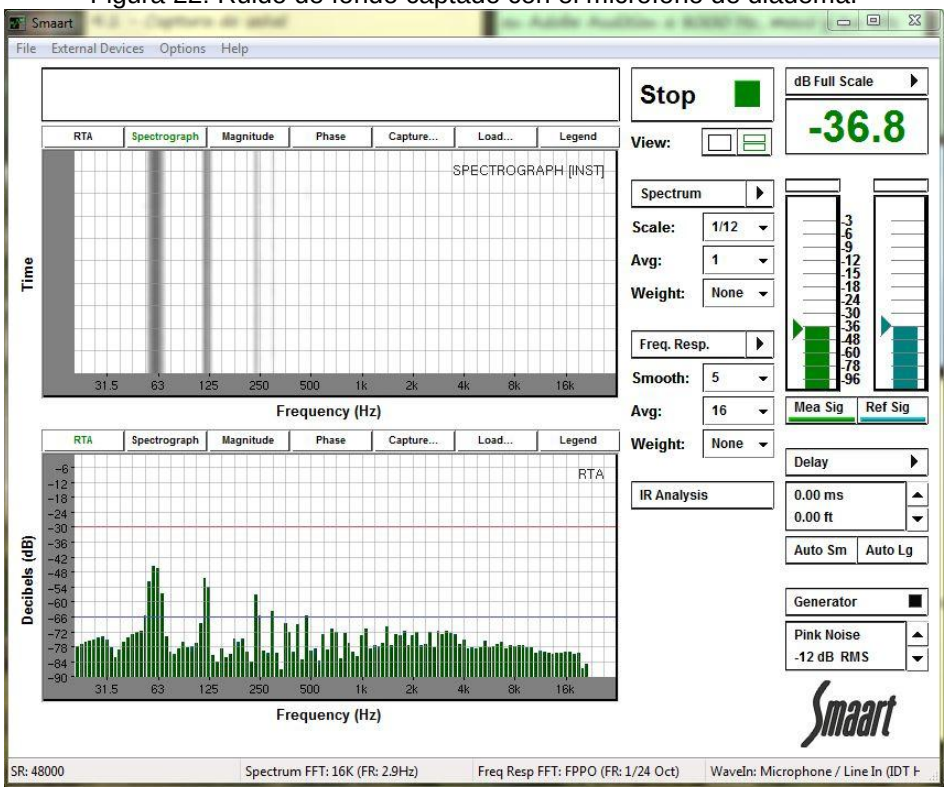
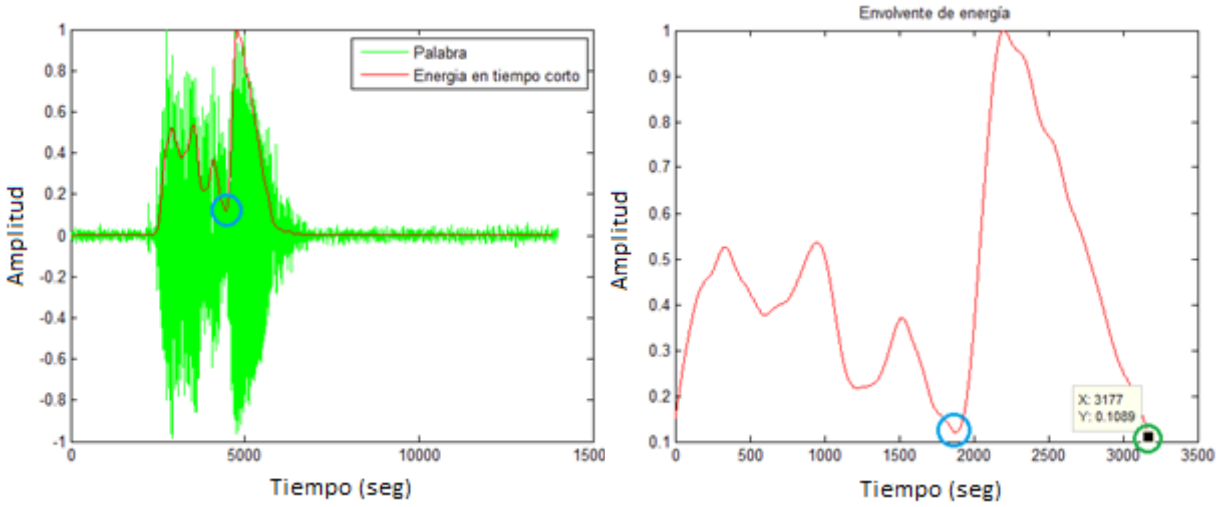


Figura 22: Ruido de fondo captado con el micrófono de diadema.



Esta diferencia de 10 dB aproximadamente es de gran importancia para que el algoritmo pueda separar las silabas con mayor facilidad y menos error. A continuación se explica el proceso de separación y porque es importante tener una señal libre de ruidos.

Figura 23: Forma de onda y energía de la palabra “abrir” (Micrófono PC)



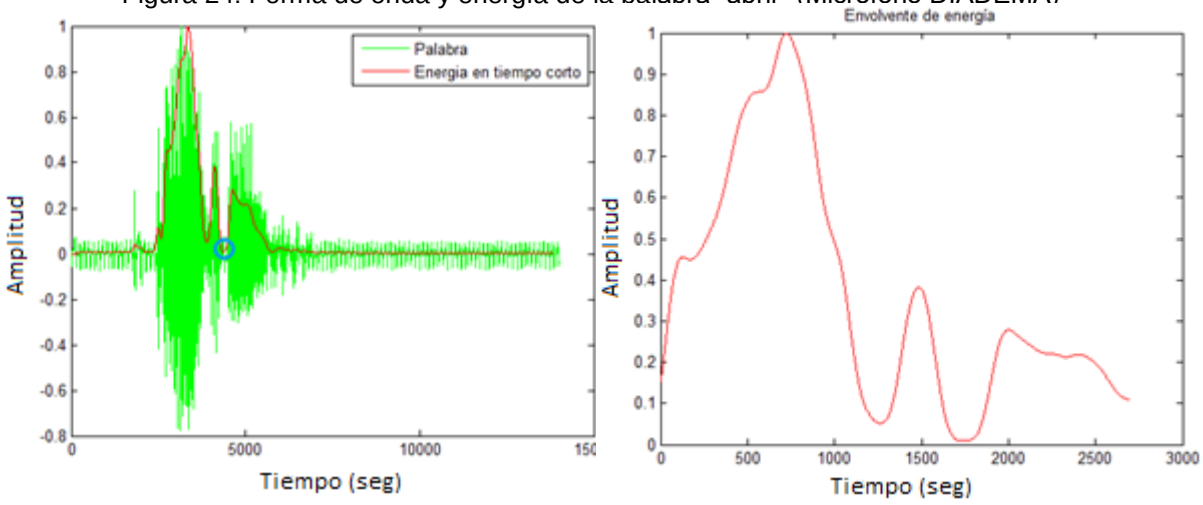
Es importante decir que estos resultados son obtenidos después de realizar todas las pruebas con el algoritmo de energía (tratado en la **sección 4.5.2**) del cual se obtienen unos valores clave por

medio de unos umbrales en amplitud para la división de las silabas. Estos valores clave son: el primero determina en qué punto del tiempo la señal deja de ser silencio y empieza la voz, el segundo establece en qué punto ya no hay más voz y continúa el silencio y el tercero (en el círculo azul),será el punto de división de las dos silabas.

El problema está en la forma como el algoritmo calcula el punto intermedio donde se dividen las silabas, ya que el algoritmo busca el punto de menor energía poniendo este como punto de división. Pero debido al ruido de fondo la energía en este punto intermedio se aumenta dejando de ser el de menor amplitud y por consiguiente ya no será este el punto de división, por el contrario este punto de menor energía será el situado al final de la señal, marcada en el círculo verde de la **figura 23**.

Una solución a este problema es cambiar el micrófono de captura, de modo que el ruido de fondo sea atenuado y sea más fácil realizar la división de silabas, en la **figura 24** se muestran los resultados del mismo ejercicio pero esta vez obtenidos con el micrófono de diadema.

Figura 24: Forma de onda y energía de la palabra “abrir” (Micrófono DIADEMA)



En esta señal tanto la división como el reconocimiento de la palabra fueron exitosos, pues el algoritmo pudo realizar la división sin ningún inconveniente gracias a que la señal esta menos contaminada de información no deseada.

Como ya se mencionó, con la información brindada por la energía, se puede hacer un buen acercamiento del inicio y fin de palabra evitando utilizar un proceso específico para la detección del inicio y fin de palabra, proceso explicado a continuación.

4.2 INICIO Y FIN DE PALABRA

El proceso de inicio y fin de palabra dentro de un sistema de reconocimiento es importante, pero en este caso no se implementó ya que al momento de usar la envolvente de energía este no era necesario, inclusive presentaba conflicto a la hora de calcular los umbrales de inicio y fin de señal de voz en la envolvente de energía.

Suprimir este paso tiene sus ventajas y sus desventajas, la principal ventaja es la eliminación de parte del código del algoritmo total y por ende una mejora en la velocidad de análisis del computador, pero la desventaja es no poder diferenciar entre una señal sonora y una señal sorda generando error al momento de la captura si se presenta un ruido impulsivo.

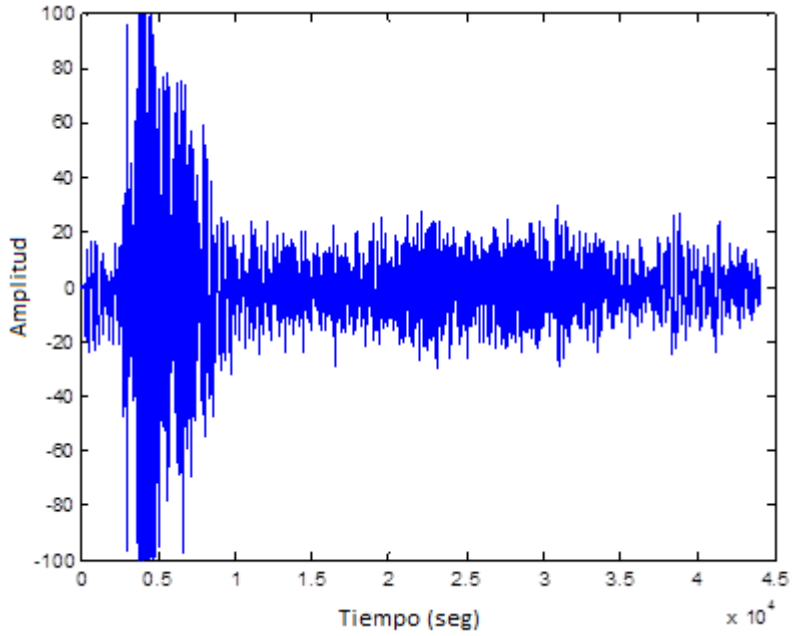
4.2.1 El Algoritmo De Inicio Y Fin

El algoritmo utilizado es muy común en este tipo de aplicaciones, y es importante porque permite distinguir la presencia de habla en entornos más o menos ruidosos, ese método se basa en dos medidas de la señal: la proporción de cruces por cero y la energía.

A continuación la explicación del proceso, en el **Anexo A** podrá consultar el código en MATLAB.

Todo parte de tener una señal digitalizada, dentro de la cual se encuentra una palabra entre un silencio inicial y un silencio final, mencionado antes como ruido de fondo ilustrado en la **figura 25**.

Figura 25: Forma de onda de la palabra “casa”.



La señal completa se divide en segmentos mucho más pequeños de una determinada cantidad de muestras correspondientes a 5.5 milisegundos aproximadamente, en los textos relacionados como “Inteligencia artificial y matemática aplicada” se menciona que estos intervalos pueden ser de entre 10 a 20 o inclusive 40 milisegundos, ya que en este periodo de tiempo se puede observar si esa parte analizada es o no cuasi periódica.

En cada uno de estos intervalos calculados se calcula la energía y el número de cruces por cero. Para después con la ayuda de estos vectores calcular unos umbrales de energía y cruces por cero, que serán utilizados en la determinación del inicio y fin de palabra.

Un punto a tener en cuenta de este algoritmo, es que para asegurar una buena detección del inicio de la palabra, la palabra no debe comenzar antes de los diez primeros intervalos, para este caso donde se tienen intervalos de 5.5 milisegundos y un solapamiento del 50% la palabra no deberá empezar antes de 30 milisegundos aproximadamente.

Una vez se tienen los vectores de energía y cruces por cero, se calculan cuatro parámetros que serán utilizados para el cálculo del inicio de palabra: ***ZCMS*** valor medio de los números de cruces por cero en los intervalos iníciales de silencio; ***DZCS*** desviación típica de los números de cruces

por cero en los intervalos iníciales de silencio; ***EMS*** valor medio de la energía en los intervalos iníciales de silencio; ***EMAX*** valor máximo de la energía en la totalidad de la señal analizada.

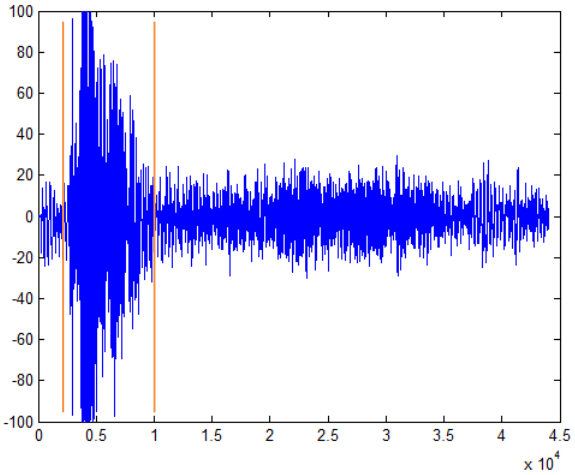
Estos parámetros permiten dar unas estimaciones del silencio inicial y proporcionan una normalización de la amplitud de la señal. A partir de estos parámetros, se definen otros tres parámetros de cálculo con los cuales se podrán hallar tres umbrales:

UZC o umbral de cruces por cero: establece un valor por encima del cual la señal es probablemente voz, sin embargo el algoritmo está diseñado para verificar que este nivel sea superado varias veces.
UEINF o umbral inferior de energía: por lo general, este valor será el cuádruple de la energía media del silencio inicial.
UESUP o umbral superior de energía: este umbral establece que la señal superior a ***UESUP*** corresponde a voz.

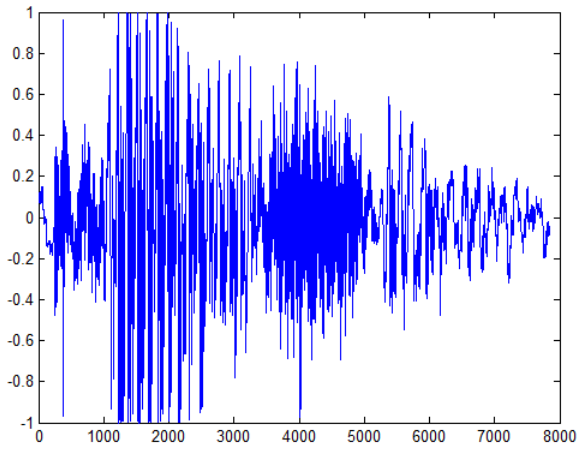
Ahora si se procede con la búsqueda del inicio de la palabra, se busca la primera ventana que supera el umbral ***UESUP***, entonces se sabe que el comienzo de la palabra esta antes de esa ventana. Se establece la ventana anterior a esta que no supere el umbral ***UEINF*** como la ventana en la cual está el inicio de palabra, Es provisional porque en caso de tener un valor bajo de energía, se hace uso del umbral de cruces por cero ***UZC***.

Finalmente para poder determinar el fin de la palabra, se procesa de igual manera que en el inicio pero esta vez se busca en la parte final de la señal, con los mismos parámetros pero calculados específicamente para el final de la palabra y se toma la señal útil, la que se encuentra entre estos dos puntos. Se muestra a continuación un ejemplo de los resultados obtenidos con este algoritmo.

Figura 26: Forma de onda de la señal completa y señal de inicio a fin



Señal de audio con voz y ruido.



***I**nicio y fin de palabra.*

En la **figura 26** se tiene la señal de la voz junto al ruido de fondo y la misma señal después de pasar por el algoritmo.

4.3 CARACTERIZACIÓN DE LA PALABRA

En el reconocimiento del habla, siempre hay un proceso de extracción de características de la señal de voz, esto con el fin de extraer información relevante de los sonidos que se están analizando. Dentro de los procesos más conocidos están coeficientes cepstrales, coeficientes de

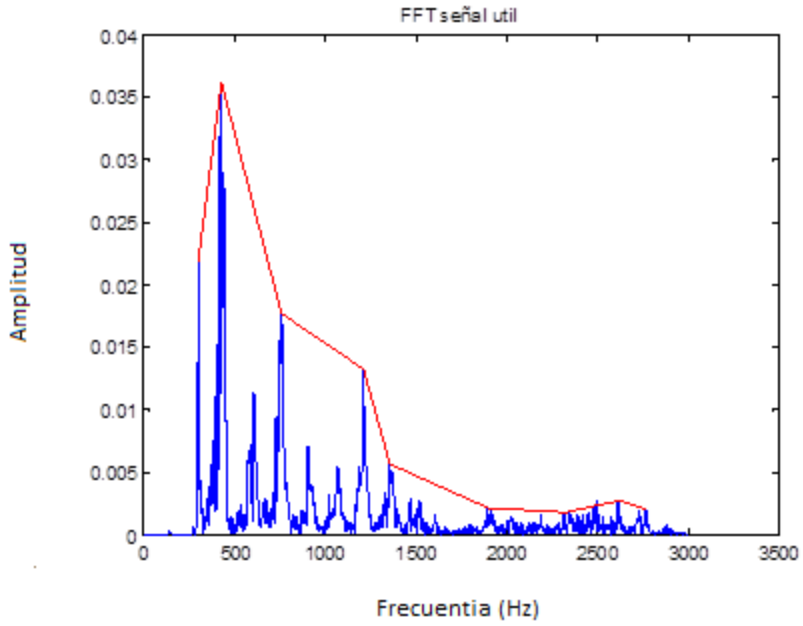
predicción lineal y formantes. Este proyecto se enfocó en la extracción de los formantes experimentando con varios algoritmos para el cálculo de los mismos, a continuación se explican los experimentos realizados para el cálculo de estos formantes y los inconvenientes encontrados. La solución final para hallar estos valores será tratada más adelante en la **sección 4.5.5**

4.3.1 Análisis en el dominio de la frecuencia

En primer lugar, era claro que se debía extraer un vector con un conjunto de valores únicos para cada palabra. De modo que se comenzaron a analizar las palabras en el dominio de la frecuencia, analizando el espectro de varias palabras tratando de identificar las concentraciones de energía y de esta manera comparar las palabras entre sí en busca de rasgos que permitieran una diferenciación.

El problema fundamental encontrado en los experimentos fue que no se podían diferenciar las palabras unas de otras, inclusive cuando estas eran totalmente distintas auditivamente. Una de las pruebas realizadas consistió en dividir en nueve partes iguales el espectro en frecuencia de la palabra y calcular el pico de mayor amplitud en cada una de estas nueve partes, pero como se comprobaría más adelante se estaban cometiendo varios errores, primero se estaba analizando la señal en su totalidad, no por segmentos más pequeños, segundo se estaba dividiendo el espectro en nueve partes de una forma arbitraria, llevando al error de tener en cuenta picos que no pertenecían a formantes ya que se analizaba la señal completa y no se tenía en cuenta una distancia mínima de separación entre los formantes, incluso picos anteriores a la frecuencia fundamental con niveles inferiores se estaban teniendo en cuenta.

Figura 27: Espectro de una palabra analizada en su totalidad con los nueve picos más altos unidos por la línea roja.



La **figura 27** es tan solo un ejemplo de los resultados obtenidos. Los datos completos son obtenidos al grabar tres personas distintas pronunciando varias palabras, donde P1 corresponde a la palabra 1, P2 a la palabra 2y P3 a la palabra 3, de igual manera L1, L2 y L3 hacen referencia a cada uno de los tres locutores, M1 a M9 son las frecuencias de los nueve máximos de amplitud hallados en cada uno de los segmentos.

Tabla 1: Frecuencias de los picos de tres locutores distintos al pronunciar tres palabras cada uno.

	Frecuencias de los picos hallados en cada intervalo								
	M1	M2	M3	M4	M5	M6	M7	M8	M9
P1 L1	303	430	743	1246	1352	1922	2213	2622	2732
P1 L2	302	521	741	1249	1349	1789	2181	2356	2963
P1 L3	329	347	710	1251	1405	1916	2150	2448	2792
P2 L1	309	434	803	1150	1460	1892	2257	2420	2814
P2 L2	301	526	667	1194	1451	1753	2191	2407	2772
P2 L3	324	534	782	1310	1366	1689	2092	2464	2740
P3 L1	302	434	734	1275	1366	1845	2278	2548	2738
P3 L2	304	555	698	1228	1534	1762	2134	2415	2685
P3 L3	329	549	724	1119	1506	1687	2035	2542	2737

Con estos valores se aprecia que no hay distinción entre las palabras, por el contrario parecen pertenecer todos a una sola palabra. Debido a esto se intenta de nuevo el mismo procedimiento pero esta vez modificando el número de divisiones de nueve a cuatro y tres; este proceso se puede asociar con la eliminación de frecuencias mencionado en la **sección 4.5.5**, ya que en ambos se experimenta para determinar cuál es el punto en el cual el algoritmo arroja los mejores resultados; los resultados obtenidos al hacer estas modificaciones mejoraron un poco pero no hasta el punto de poder encontrar diferencias claras entre las palabras y que estos resultados fueran constantes.

A esto hay que agregarle la falta de confiabilidad de los valores obtenidos, si eran correctos o no, se determinará más adelante. En la **sección 4.5.5** se presentan las pruebas con el fin de averiguar si los resultados eran verídicos.

Tabla 2: Cinco locutores pronuncian dos palabras con tres repeticiones cada una.

	Frecuencias (Hz) de los picos					Frecuencias (Hz) de los picos			
	M1	M2	M3	M4		M1	M2	M3	M4
P1 L1	448	702	1172	1362	P2 L1	625	724	1209	1452
	468	718	1199	1440		664	668	1206	1334
	468	706	1178	1405		662	676	1164	1347
P1 L2	306	433	764	1213	P2 L2	300	448	729	1076
	303	425	733	1215		303	428	729	1252
	300	433	733	1309		325	425	952	1121
P1 L3	529	765	1269	2194	P2 L3	527	668	1080	1579
	510	746	1162	1364		522	666	1311	1421
	300	525	711	1317		529	667	1190	1354
P1 L4	326	354	703	1234	P2 L4	331	333	1330	1335
	330	335	730	1292		321	852	1331	1336
	330	353	698	1227		310	629	670	1426
P1 L5	322	445	808	2020	P2 L5	326	554	666	1258
	327	427	816	1265		556	701	1279	1453
	326	398	809	1899		574	728	1303	1437

Tabla 3: Dos locutores pronuncian tres palabras con tres repeticiones cada uno.

	Frecuencias (Hz) de los picos									
	M1	**M2**	**M3**		**M1**	**M2**	**M3**		**M1**	**M2**	**M3**
P1 L1	396	762	1405	P2 L1	734	1217	1456	P3 L1	543	805	1071
	412	649	1418		769	972	1250		528	872	1399
	410	899	1302		768	1405	1677		546	774	1374
P1 L2	478	727	977	P2 L2	768	1015	1282	P3 L2	765	1031	1297
	483	792	1486		768	1005	1261		765	1559	1772
	490	730	980		745	987	1247		752	983	1728

En este nuevo experimento se comprueba que hay una leve mejora en los resultados, pero aun así, primero los resultados no son consistentes, es decir las frecuencias de los picos de máxima amplitud en ocasiones varían su rango de manera drástica, como ejemplo podemos observar en la **tabla 2**los valores resaltados en amarillo pertenecientes al cuarto máximo de las tres repeticiones realizados por el locutor 5 en la palabra 1,y segundo no tienen concordancia con las de las frecuencias de los otros locutores en la misma palabra.

Después de un largo tiempo realizando pruebas con estos números de los picos y modificando las palabras al ver que los resultados siempre eran los mismos, se toma la decisión de dividir la palabra, y es allí cuando se ensaya la extracción de características por silaba, así que se podían analizar palabras monosilábicas, pero no se hizo de esta manera ya que debido a la estructura del algoritmo este solo podría reconocer con seguridad 5 palabras, cada una con una vocal distinta, y si se escogían palabras de más de dos vocales el proceso de separación sería más complejo y se correría el riesgo de no tener un buen porcentaje de reconocimiento, sin mencionar las variables implicadas que no hacen parte del algoritmo. Por esta razón se decidió trabajar con palabras de dos silabas para asegurar un buen porcentaje de reconocimiento.

El proceso de separación de silabas es explicado en la **sección 4.5.4**.

4.3.2 Análisis en el dominio del tiempo

Cabe destacar que también se realizaron pruebas en el dominio del tiempo con el fin de encontrar parámetros adicionales que permitieran la diferenciación entre palabras. Este consistió en obtener un valor dependiendo de la duración de la palabra, y conjugarlo en un solo vector con un nuevo experimento basado en los nueve picos del espectro, antes de explicarlo es importante decir que con este método empírico los porcentajes de reconocimiento fueron bastante buenos. El código puede ser consultado en el **Anexo B**.

Este método no calcula formantes, simplemente se basa en un vector codificado en binario obtenido con la ayuda de un umbral de amplitud en el espectro que contiene los nueve picos.

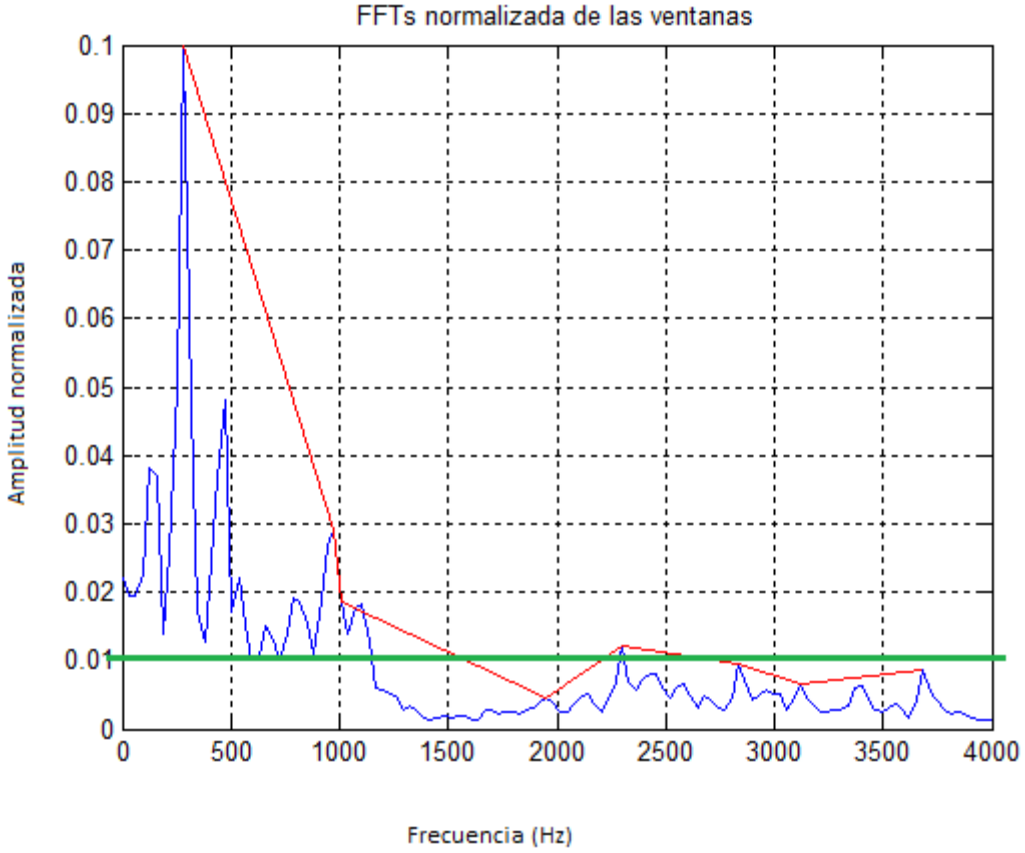
Una vez obtenidas las frecuencias de los nueve picos del espectro y después de las pruebas, se observó que los picos tenían ciertas características de amplitud similares entre la misma palabra, de nuevo se empezó a experimentar con estos valores de amplitud hasta llegar a un umbral por el cual ciertas frecuencias lo superaban y otras no, de acuerdo a estos valores, se le asignaba un uno (1) o un cero (0) a ese pico, construyendo un vector binario de nueve posiciones.

En la **figura 28** se representa con una línea de color verde el umbral por el cual los valores encima de esta amplitud serán unos y los que estén por debajo serán ceros, este valor es un punto de equilibrio clave para el correcto funcionamiento de este método, puesto que si este umbral está muy arriba, habrán más ceros que unos, pero si se encuentra muy abajo, sucederá lo contrario, de modo que para calcular este valor hacen falta muchas pruebas con distintas palabras siempre verificando los valores binarios que se calculan, hasta llegar a un punto donde se permita la distinción entre palabras, existirán algunas palabras que tengan la misma codificación que otras, es ahí donde interviene el factor calculado en el tiempo, obteniendo al final un vector como el que se muestra en la **tabla 4** .

Tabla 4: Vector final del método ayudado con análisis en el dominio del tiempo y codificación binaria para los máximos del espectro.

55	1	1	1	0	1	0	0	0

Figura 28: Números y umbral para la codificación en binario.



Siguiendo este método de caracterización se obtuvieron los siguientes resultados, las palabras usadas en este experimento son distintas a las que se usan en el algoritmo final y son solo ocho. Este vector está conformado por dos partes, la primera hace referencia a los límites dentro de los cuales puede oscilar este número (se obtiene después de trabajar con el vector de energía, específicamente con las diferencias en tiempo que existen entre los dos picos máximos de amplitud), y la segunda a las posibles combinaciones en binario para la palabra así:

Tabla 5: Segmentación del vector característico, umbrales máximo y mínimo y dos posibles codificaciones en binario.

Max	95	1	1	1	0	0	0	0	0
Min	32	1	1	0	0	0	0	0	0

Tabla 6: Resultados de las ocho palabras, vectores característicos.

Palabra "Uno"									
Max	95	1	1	1	0	0	0	0	0
Min	32	1	1	0	0	0	0	0	0
porcentaje de acierto							98%		

Palabra "Escalpelo"									
Max	104	1	1	1	1	1	1	0	0
Min	70	1	1	1	1	1	0	0	0
		1	1	1	1	1	1	1	0
		1	1	1	1	1	1	0	1
porcentaje de acierto							92%		

Palabra "Tieras"									
Max	71	1	1	0	1	1	1	1	1
Min	24	1	1	0	1	1	1	1	0
		1	1	0	1	1	1	0	0
		1	1	0	1	1	0	0	0
porcentaje de acierto							80%		

Palabra "Abrir"									
Max	64	1	1	1	1	1	1	0	0
Min	17	1	1	1	1	1	1	1	0
		1	1	1	1	1	0	1	0
porcentaje de acierto							62%		

Palabra "Mico"									
Max	96	1	1	1	0	1	1	1	0
Min	38	1	1	1	0	1	1	1	1
		1	1	1	0	1	0	1	0
		1	1	1	0	0	1	1	1
porcentaje de acierto							70%		

Palabra "Manzana"									
Max	93	1	1	1	1	0	1	0	0
Min	50	1	1	1	1	0	0	0	0
porcentaje de acierto							96%		

Palabra "Buzo"									
Max	226	1	1	0	0	0	0	1	1
Min	36	1	1	0	0	0	0	1	0
		1	1	1	0	0	0	1	1
		1	1	1	0	0	0	0	1
porcentaje de acierto							82%		

Palabra "Iris"									
Max	95	1	0	0	0	1	1	1	1
Min	32	1	0	0	0	1	0	1	1
		1	0	0	0	1	1	1	0
		1	0	0	0	1	0	0	0
porcentaje de acierto							84%		

Posteriormente para lograr hacer la identificación de la palabra, fue necesario utilizar reconocimiento de patrones, siendo esta la prueba de que reconocimiento de patrones es un método confiable. Es acá cuando se resuelve que el método de patrones es el más viable por su flexibilidad en la programación y su fácil manejo, siempre y cuando el proceso de caracterización sea confiable.

4.4 RED NEURONAL BACKPROPAGATION

Como se comentó en la **sección 4** el proceso con redes neuronales no se implementó finalmente en la aplicación, sin embargo a partir de la **sección 4.4.1** se explica el proceso de las pruebas realizadas. Cabe anotar que desde un primer momento se seleccionó la red neuronal Backpropagation para el desarrollo de estas pruebas, esto dado que en la bibliografía consultada se encontró que es el método con el cual se obtienen mejores resultados y por eso es este el más utilizado en las aplicaciones de reconocimiento de voz.

A continuación se muestra el proceso de creación, entrenamiento, simulación y cálculo de error de la palabra grifo.

4.4.1 Definición de la entrada y salida de la red

Una vez el algoritmo de formantes ha determinado los valores para el formante 1 y el formante 2 por cada sílaba estos valores son guardados y utilizados como la entrada de la red:

Tabla 7: Formantes palabra grifo hablante femenino.

Sílaba 1 (Hz)		Sílaba 2 (Hz)	
3464	3968	535	1007

A partir de los valores de la **tabla 12** se define la entrada de la red como:

entgrifo=[3464 3968   535 1007];

La salida deseada para la red será la siguiente

salgrifo=[1 1 0 0];

Nota: En el entrenamiento de esta red neuronal se utiliza la función Sigmoidal de la cual se habla en el punto 2.2.3 Funciones de transferencia de las redes neuronales artificiales. La función Sigmoidal tiene la propiedad de establecer valores de salida que oscilen entre 0 y 1 y a menos que se especifique lo contrario es la función utilizada por defecto en la red neuronal Backpropagation en MATLAB.

4.4.2 Creación y definición de los parámetros de la red Backpropagation

Una vez se han definido la entrada y la salida, se crea la red neuronal Backpropagation con dos capas ocultas, la primera con 10 neuronas y la segunda con 25 y utiliza el algoritmo de entrenamiento *trainbfg.*

```
netgrifo=newff(entgrifo,salgrifo,[10 25],{},'trainbfg');
```

Existen una serie de parámetros que se pueden definir de acuerdo con el algoritmo de entrenamiento, siendo Trainbfg el algoritmo usado en este caso, estos son algunos de los parámetros establecidos:

```
netgrifo.trainParam.epochs= 500; Número Máximo de posibles iteraciones
netgrifo.trainParam.show= 100; Muestra el Parámetro del progreso del entrenamiento
netgrifo.trainParam.goal= 1e-8; Error al que se quiere llegar
```

Teniendo en cuenta que un gradiente es un campo vectorial que indica en cada punto del campo la dirección de incremento del mismo, se utiliza la función netgrifo.trainParam.min_grad para definir el incremento del gradiente que utilizará el algoritmo. Este no puede ser muy pequeño ya que el ajuste de los pesos se tornaría muy lento, pero tampoco debe ser un valor muy grande ya que el comportamiento del entrenamiento podría ser inestable.

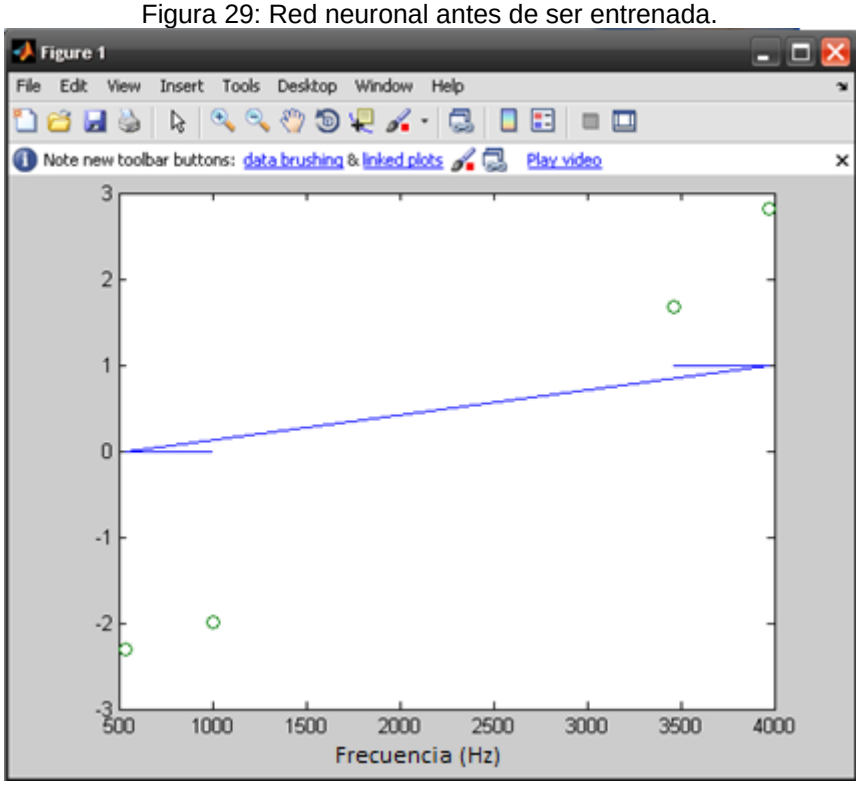
```
netgrifo.trainParam.min_grad=1e-6;
```

4.4.3 Simulación de la red antes de ser entrenada

Posteriormente se realiza una simulación con el fin de graficar el comportamiento de la red antes de ser entrenada. El comando *sim* permite realizar la simulación con las entradas y salidas de la red, en la variable Pgrifo se cargará el resultado de la salida de la red.

```
figure(1);
Pgrifo = sim(netgrifo,entgrifo);
plot(entgrifo,salgrifo,entgrifo,Pgrifo,'o')
```

Figura 29: Red neuronal antes de ser entrenada.



En la **figura 29** la línea Azul presenta la salida deseada, es decir [1 1 0 0]. Los círculos verdes representan la entrada.

4.4.4 Entrenamiento y cálculo del error de la red Backpropagation para la palabra grifo

Cuando se ha creado la red, esta se entrena utilizando la función *train* y se realiza una segunda gráfica en la que se simule el comportamiento de la red entrenada.

```
netgrifo=train(netgrifo,entgrifo,salgrifo); Entrenamiento de la red
figure(2);  En la figura dos se va a dibujar la red entrenada
Pgrifo= sim(netgrifo,entgrifo);  En Pgrifo se simula la red entrenada
plot(entgrifo,salgrifo,entgrifo,Pgrifo,'o')  Se dibuja la red entrenada.
```

Figura 30: Entrenamiento de la red

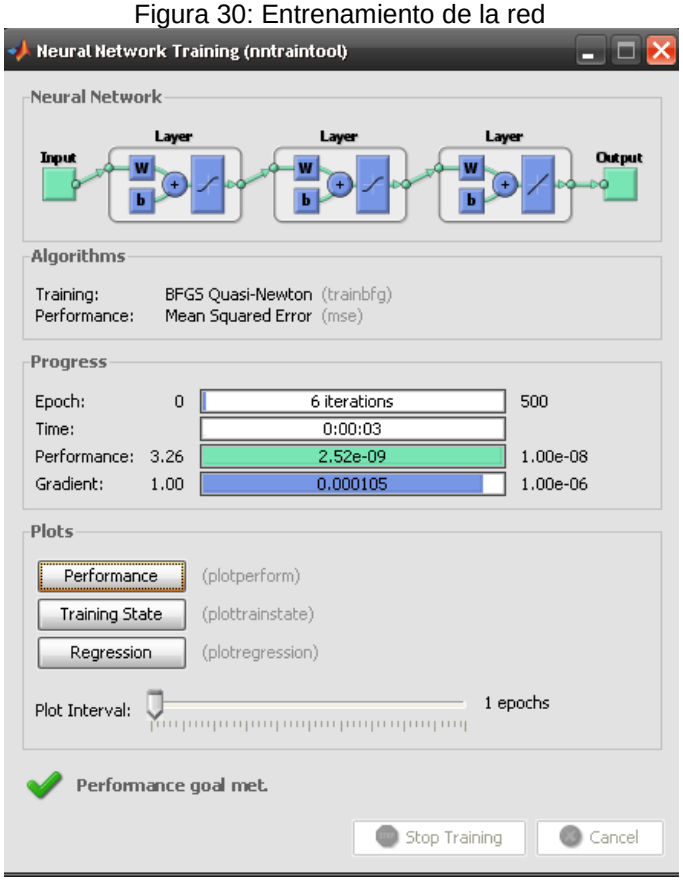
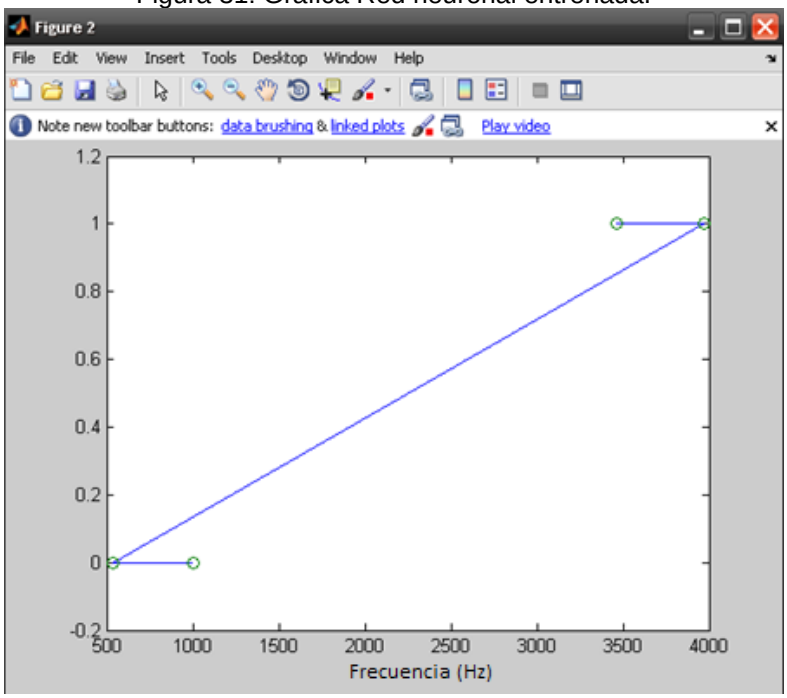


Figura 31: Gráfica Red neuronal entrenada.



La **figura 31** muestra que los valores de la entrada han alcanzado los valores de la salida deseada, es decir, [1 1 0 0] y en **la figura 30** se muestra que en 6 iteraciones se ha alcanzando un error cercano al error planteado inicialmente (1e-8).

Por último se calcula el error a partir de la salida obtenida y la salida deseada y a este valor se le aplica la función *mse* con el fin de calcular el error cuadrático medio, este error le permite a la persona que entrena la red conocer el progreso de entrenamiento de la misma.

```
errorgrifo = salgrifo-Pgrifo;
ERRORgrifo= mse(errorgrifo);
```

4.4.5 Simulación con la red entrenada

Finalmente, para comprobar el entrenamiento de la red se realiza una simulación con valores parecidos a los del entrenamiento, es decir, valores que se podrían presentar al determinar los formantes una vez grabada la palabra grifo:

En este caso la entrada para la simulación será: (3400 3953 600 1050). Por lo tanto, al utilizar el comando de simulación *sim* la simulación se programará de esta forma:

```
sim (netgrifo, [3400      3953    600     1050]);
```

Al realizar la simulación la salida presentada por la red es la siguiente:

```
[1E+00        1E+00        4E-05        -9E-05]
```

Esto quiere decir que efectivamente al simular la red con unos valores que se podrían presentar al determinar los formantes una vez grabada la palabra grifo, se puede obtener la salida deseada que se había planteado en un comienzo, es decir, [1 1 0 0]

Al analizar el valor del error cuadrático medio podemos determinar que en este caso la red ya está entrenada para valores cercanos a los presentados en el entrenamiento.

Error cuadrático medio: 2,5E-09

Para consultar el código de la red neuronal Backpropagation completo ver el **Anexo D**.

4.4.6 Salidas deseadas para cada una de las nueve palabras que hacen parte de la aplicación

Los demás parámetros de la red se pueden consultar en el código de entrenamiento de la red, ver **Anexo D**.

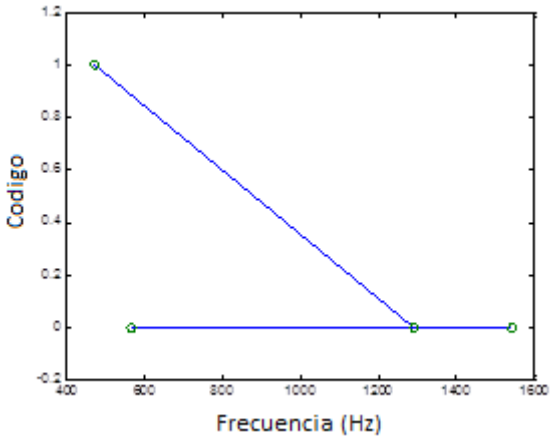
La tabla mostrada a continuación contiene la salida deseada para cada palabra, esta consiste en cuatro números entre unos y ceros, finalmente es esta salida obtenida la que indica cual palabra fue reconocida.

Tabla 8: Códigos asignados a cada palabra en el entrenamiento de la red neuronal.

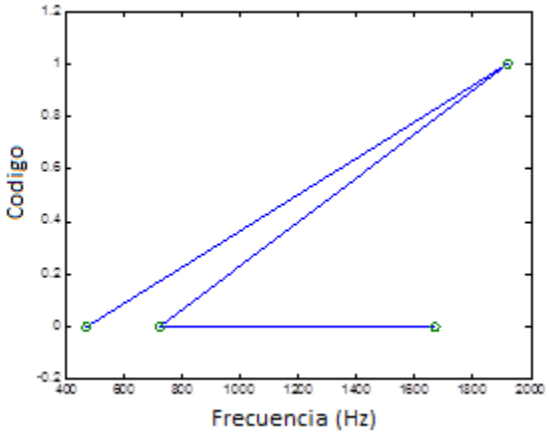
Palabra	*Código*
Fruta	1000
Señal	0100
Flecha	0010
Gato	0001
Grifo	1100
Vagón	0110
Mesa	0011
Hola	1110
Abrir	0111

A continuación se muestran las gráficas de la simulación una vez realizado el entrenamiento. Los valores alcanzados en las salidas se dan de acuerdo a las salidas deseadas definidas en la **tabla 13**. En las gráficas las frecuencias dadas por los formantes de cada palabra corresponden al eje de las abscisas y los valores de código al eje de las ordenadas. Los valores de formantes para cada palabra se muestran en la **sección 5.3.1**.

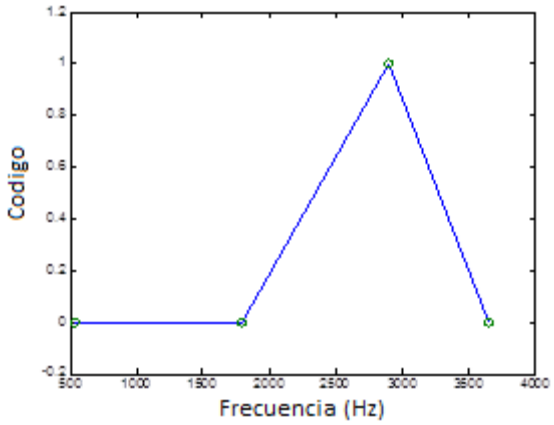
Figura 32: Graficas de las nueve palabras entrenadas por en la red neuronal.



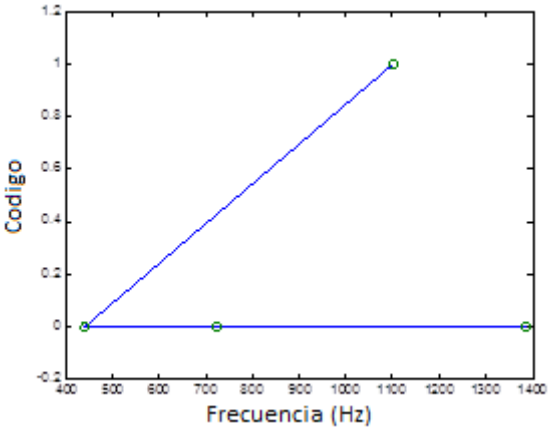
Entrenamiento de “Fruta”



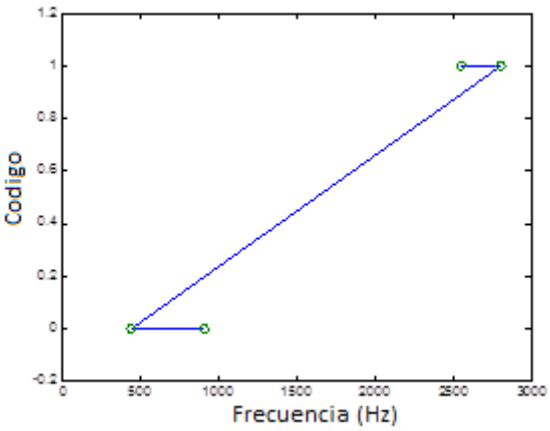
Entrenamiento de “Señal”



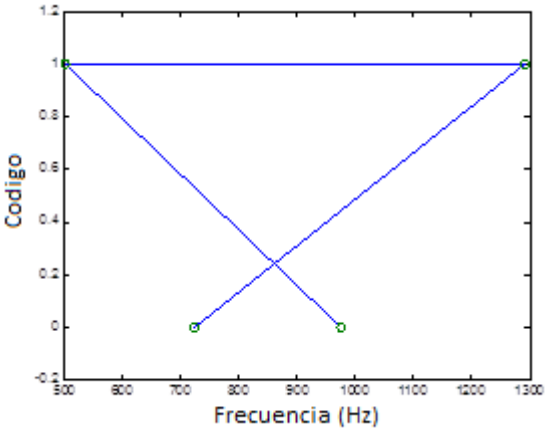
Entrenamiento de “Flecha”



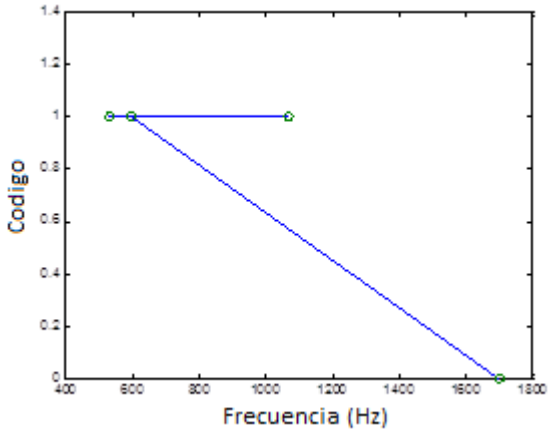
Entrenamiento de “Gato”



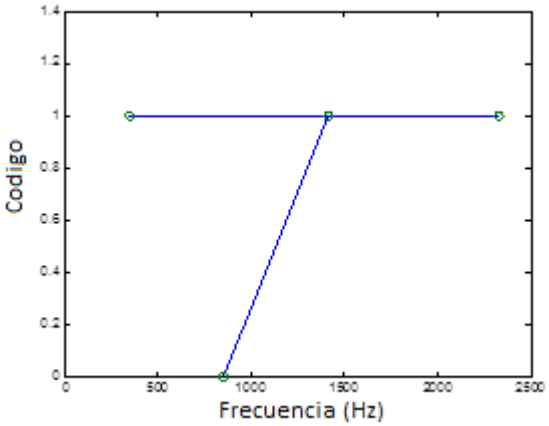
Entrenamiento de “Grifo”



Entrenamiento de “Vagón”



Entrenamiento de “Hola”



Entrenamiento de “Abrir”

Cuando la red entrenada produzca una salida estos valores serán analizados de acuerdo a unos umbrales definidos por el usuario. Si estos se encuentran entre alguno de los nueve rangos definidos para cada una de las nueve palabras, se determinará que palabra fue reconocida. Para consultar el código ver el **Anexo E.**

4.5 DESARROLLO DEL ALGORITMO FINAL Y METODO RECONOCIMIENTO DE PATRONES

En este punto se explicara cómo funciona el algoritmo final y los parámetros seguidos, retomando puntos anteriores como la captura de la señal, el cálculo de la energía, la división de las silabas, y el cálculo de las frecuencias formantes.

Se toma la decisión de desarrollar todo el algoritmo en un solo numeral para que el lector tenga la facilidad de seguir el proceso del algoritmo final con el cual se desarrolló la aplicación del proyecto.

4.5.1 Captura de la señal

El algoritmo comienza con la captura de la señal de audio por medio de un micrófono de diadema (Audífonos Genius HS-04S), se utiliza este y no el del computador porque este ayuda a eliminar gran parte del ruido de fondo y por consiguiente en la separación de las silabas.

El algoritmo es desarrollado en MATLAB (versión 7.9 R2009b), software especializado en procesamiento digital de señales, dispone de gran cantidad de funciones respecto a las señales de audio, una de ellas es usada para la captura de sonido por medio de un micrófono.

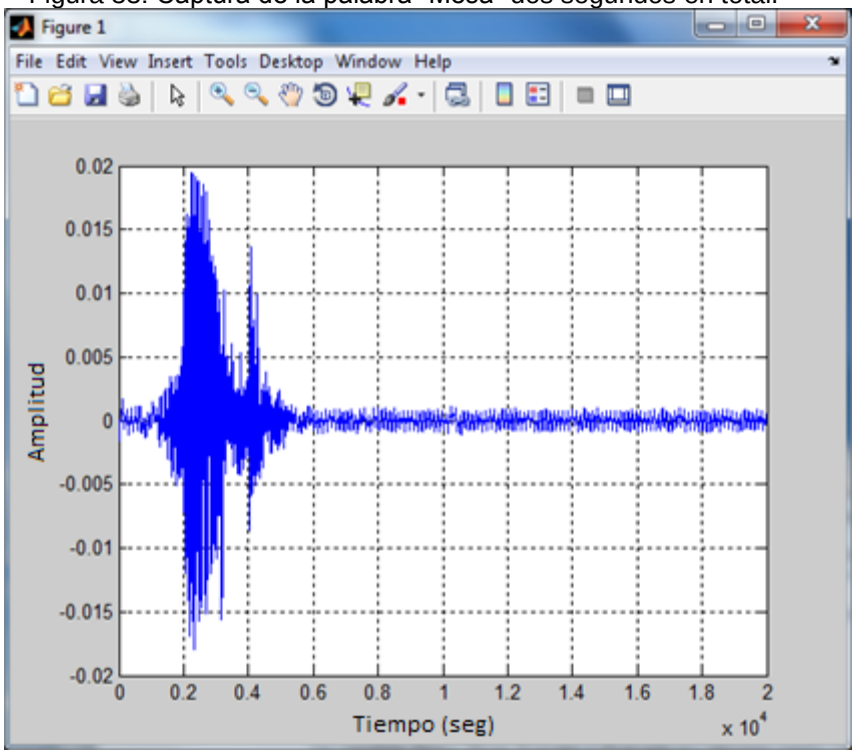
```
y=wavrecord(2*Fs,Fs,1);
```

Es de resaltar que en los dos métodos siempre se siguieron los mismos parámetros para la captura de señal, únicamente se cambió el micrófono.

Los parámetros que se siguieron en la captura fueron:

- Frecuencia de muestreo Fs=8000 Hz.
- Al grabar, se capturan 2 segundos y luego continua con el programa, este valor puede ser modificado según las necesidades.
- Por defecto graba a una resolución de 16 bits.
- Un solo canal, monofónico.

Figura 33: Captura de la palabra “Mesa” dos segundos en total.

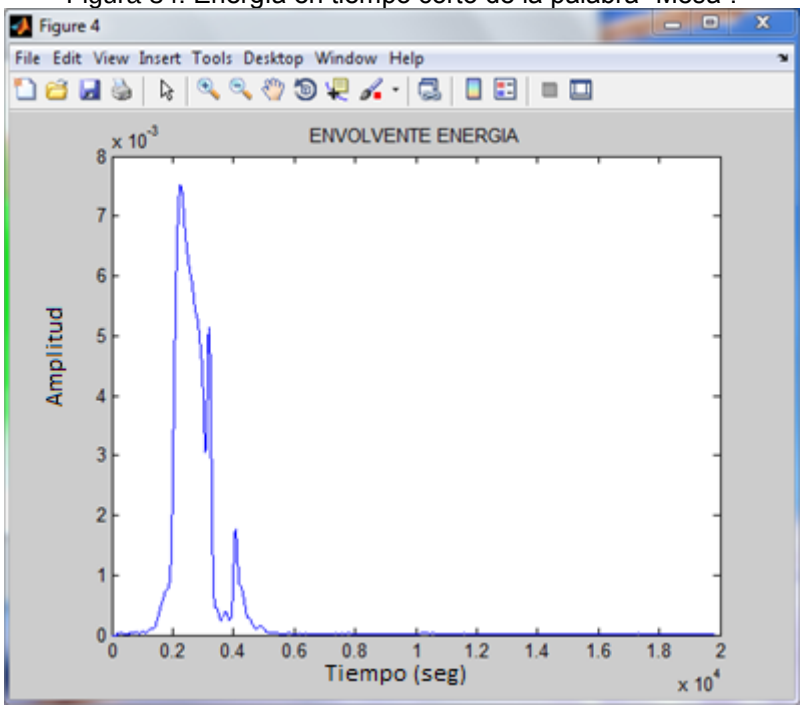


4.5.2 Calculo de envolvente de energía

Short Time Energy o Energía en tiempo corto puede ser usada para determinar voz vs. no voz en una señal, también puede ser usada para determinar la transición de señal de voz a no voz y viceversa.

Este proceso se efectúo analizando la señal por intervalos cortos de tiempo de 300 muestras o 37,5 milisegundos, y solapándolos para acoplar la información que hay entre uno y otro intervalo. El resultado de este proceso es un vector como el siguiente:

Figura 34: Energía en tiempo corto de la palabra “Mesa”.



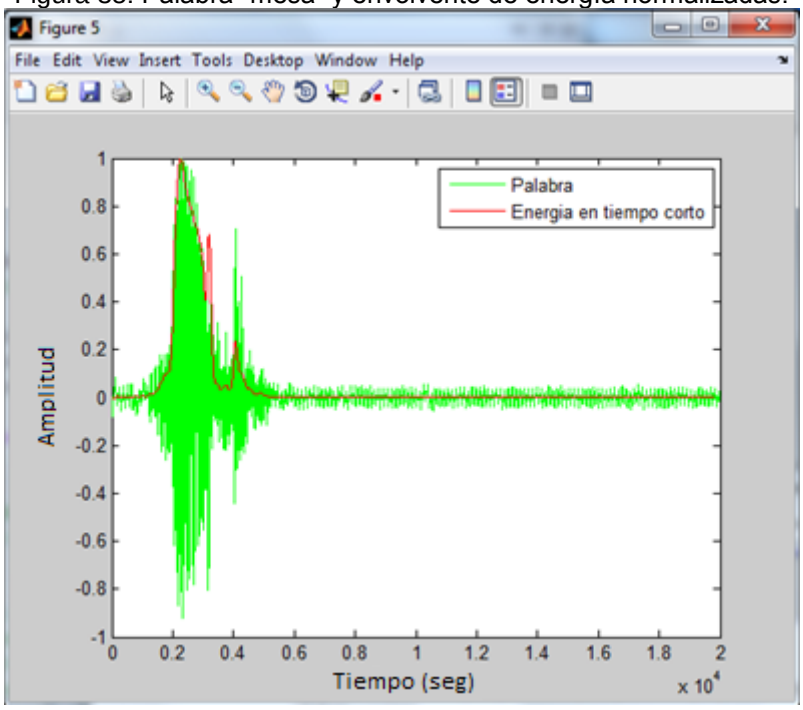
Este es un paso muy importante porque de aquí partimos para la división de la palabra en dos silabas.

4.5.3 Normalización de palabra y energía

Después de tener la señal y su respectiva envolvente de energía, se normalizó con el fin de estandarizar los niveles y poder trabajar tranquilamente con los umbrales, con la certeza de que sin importar si la palabra ingresó al sistema con un nivel alto o bajo se tendrán buenos resultados.

También se hacen coincidir en tiempo las dos señales, la palabra y la energía de la misma como se muestra en la **figura 35**.

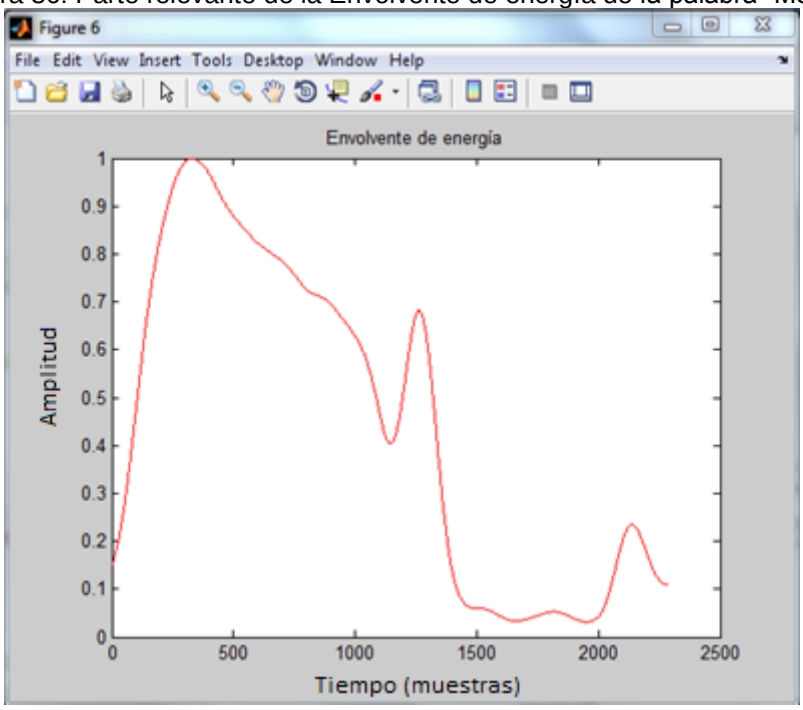
Figura 35: Palabra “mesa” y envolvente de energía normalizadas.



4.5.4 Cálculo y Extracción de silabas

Este es un procedimiento basado en la envolvente de energía. Teniendo en cuenta las envolventes de energía de las palabras, pronunciadas en repetidas ocasiones, se analizó en qué puntos la energía superaba un umbral al inicio y otro al final de la palabra delimitando la envolvente de energía a la sección donde tenemos la información de voz como se muestra en la imagen.

Figura 36: Parte relevante de la Envolvente de energía de la palabra “Mesa”.

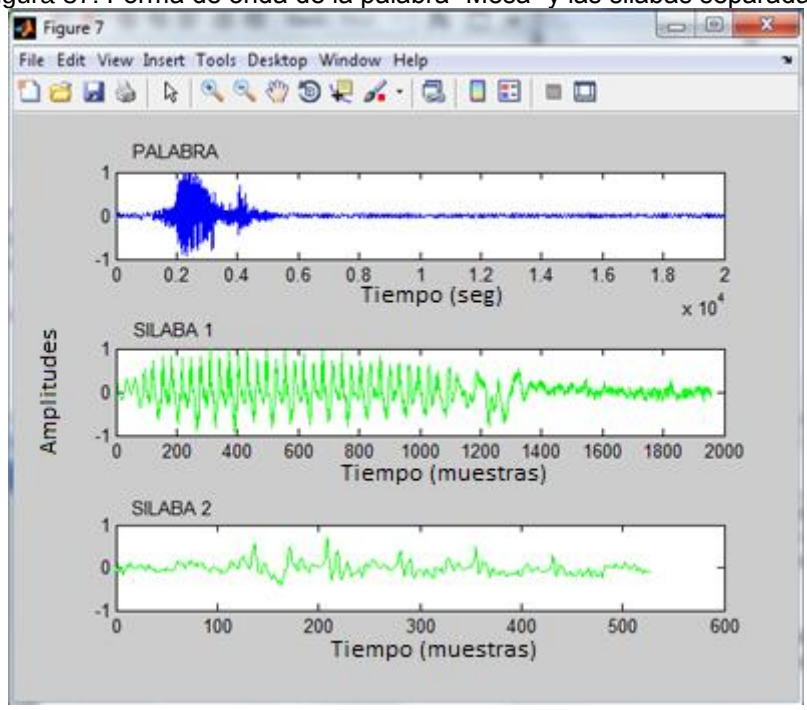


Gracias a la información que contiene esta variable, se puede determinar hasta qué punto va la primera silaba, y cual corresponde a la segunda. Es de suma importancia que las señales de energía y señal capturada estén sincronizadas en tiempo, de lo contrario los puntos hallados en la envolvente de energía no corresponderán con los hallados en la forma de onda y por consiguiente la división no será la correcta.

Cabe aclarar que es un proceso básico donde solo se están teniendo en cuenta umbrales de energía, por esta razón al ingresarle una palabra de más silabas el algoritmo buscara solo dos según los umbrales establecidos.

Después de identificar los puntos donde se superan los umbrales, podemos hacer la separación de las silabas obteniendo los siguientes resultados como lo muestra la imagen.

Figura 37: Forma de onda de la palabra “Mesa” y las silabas separadas.



4.5.5 Segmentación, FFT y Formantes

Al igual que en el cálculo de la envolvente de energía a cada silaba se le aplica una segmentación y a cada una de las ventanas se le aplica la transformada rápida de Fourier con el único propósito de calcular los formantes de la silaba, en este caso solo fueron calculados los dos primeros formantes por sílaba.

Figura 38: Análisis FFT para cada una de las silabas de la palabra “Mesa”, en los círculos rojos se muestran los formantes de cada una.

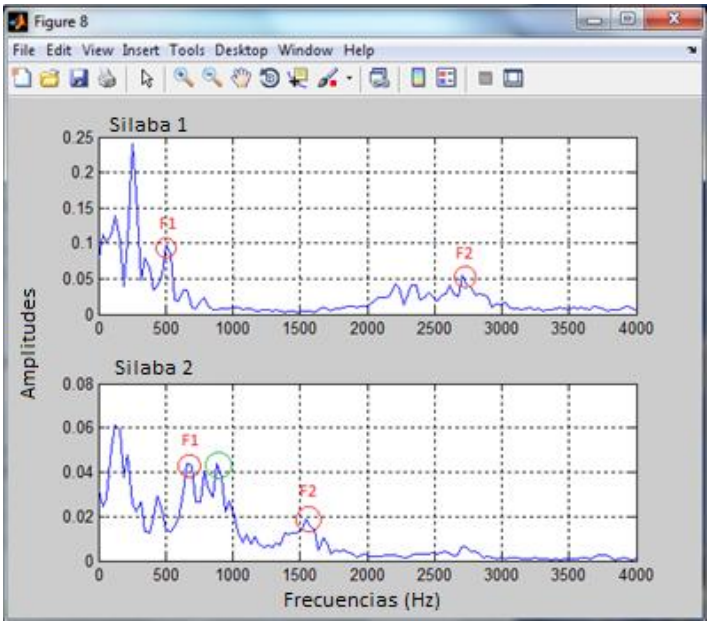


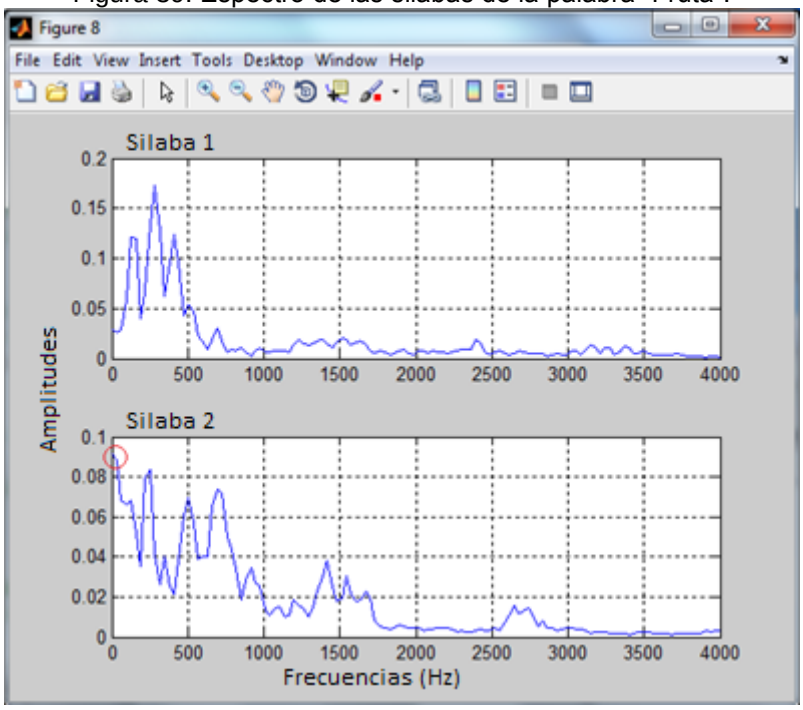
Tabla 9: Valores correspondientes a los formantes de la palabra “Mesa”.

Silaba 1		Silaba 2	
Formante 1 (Hz)	Formante 2 (Hz)	Formante 1 (Hz)	Formante 2 (Hz)
503	2708	661	1543

En la figura anterior se muestran encerrados en rojo los formantes encontrados por el algoritmo, el pico encerrado en el círculo verde junto al formante uno de la silaba 2 que se encuentra alrededor de los 850 Hz es levemente menor en amplitud respecto a F1, por esta razón el algoritmo reconoce al pico encerrado en rojo en 661 Hz y no al pico encerado en verde. El procedimiento para el cálculo de formantes se explicará más adelante.

Existen dos picos de mayor amplitud que F1, pues bien, después de muchas pruebas, en ocasiones este pico más alto no correspondía con el formante realmente, es decir se encontraba en una frecuencia muy baja, más baja que la frecuencia fundamental de la voz inclusive cero, véase la siguiente grafica de la palabra fruta:

Figura 39: Espectro de las silabas de la palabra “Fruta”.



La **figura 39** muestra que en el análisis en frecuencia de la segunda silaba, el pico más alto se encuentra en 0 Hz lo cual es un dato herrado, después de ensayo y error, se logró estabilizar el algoritmo despreciando los picos por debajo de 472 Hz en el primer formante de la primera silaba y 250 Hz para el primer formante de la segunda silaba.

PROCESO DE EXTRACCION DE FORMANTES

El método utilizado en este algoritmo está basado en una teoría muy sencilla pero eficiente si se sabe aplicar. La idea es calcular unos valores máximos a partir de la respuesta en frecuencia, el procedimiento fue calcular el pico más alto de la señal en el dominio de la frecuencia, después se eliminan un cierto número de frecuencias y se vuelve a calcular el pico más alto de la señal restante para hallar el siguiente formante, algunos autores recomiendan que este valor debe ser de

150 Hz, de este modo, si el primer pico se centra en 500 Hz, el segundo se debe buscar a partir de 650 Hz.

Desafortunadamente eliminar 150Hz no arrojó buenos resultados, algunas veces el segundo formante se podía encontrar muy cerca o muy lejos del primero, en otras palabras no era confiable. Pero se continuó con la idea y se realizaron pruebas para fijar el número de frecuencias a eliminar llegando a un valor de 473Hz.

VERIFICACION DE FORMANTES

Después de tener el algoritmo terminado hasta este punto, la forma de verificar si estos formantes que se calculaba eran confiables fue compararlos con las formantes de las vocales en el español. Los valores de formantes de la siguiente tabla fueron extraídos del texto *Inteligencia Artificial y Matemática Aplicada* de *Gustavo Santos García.*

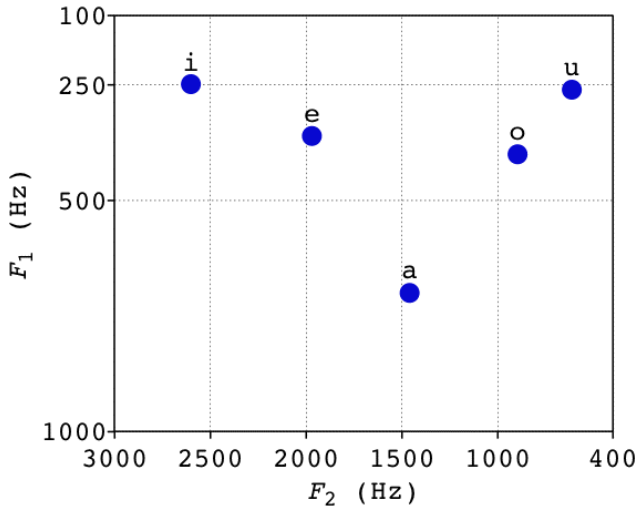
Tabla 10: Formantes de los sonidos vocálicos en el español

	Formante 1 (Hz)	Formante 2 (Hz)
Vocal a	700	1500
Vocal e	500	1800
Vocal i	400	2000
Vocal o	500	1000
Vocal u	400	700

Tomado de: Inteligencia Artificial y Matemática Aplicada.

Los valores son aproximados, pero son similares a los que se encuentran en una carta de formantes, en una carta de formantes los valores en frecuencia de los formantes se muestran en una gráfica donde cada formante es mostrado en cada uno de los dos ejes.

Figura 40: Carta de formantes de las 5 vocales españolas sintetizadas a partir de los datos de Ruiz y Soto-Barba (2005).



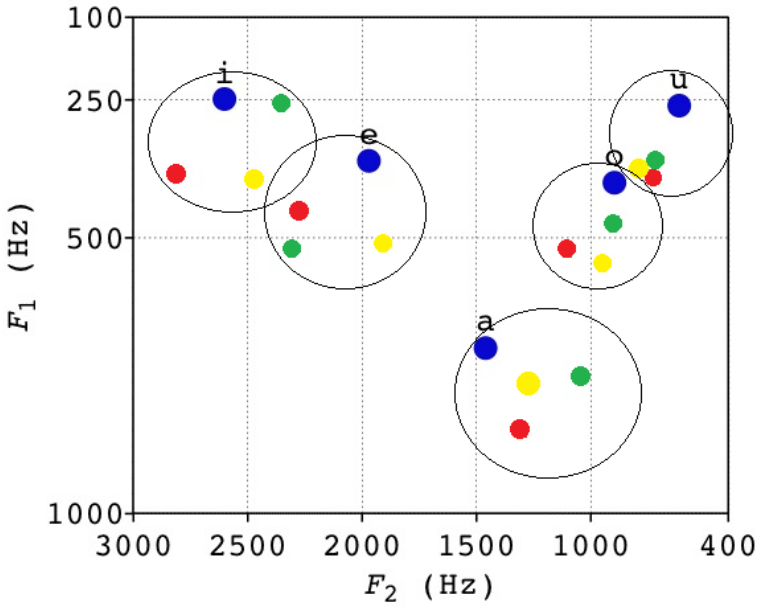
Jaime Soto-Barba y Magaly Ruiz son investigadores del (*Laboratorio de Fonética de la Universidad de Concepción)* en Chile. "Timbre vocálico en hablantes de español como segunda lengua".

A continuación se muestran los resultados obtenidos con el algoritmo de verificación de los formantes de las vocales y una distribución en la carta de formantes de distintos locutores.

Tabla 11: Datos obtenidos de dos repeticiones de las vocales por el mismo hablante masculino.

	Referencia		Prueba 1		Prueba 1	
	Formante 1 (Hz)	Formante 2 (Hz)	Formante 1 (Hz)	Formante 2 (Hz)	Formante 1 (Hz)	Formante 2 (Hz)
Vocal a	700	1500	852	1367	818	1322
Vocal e	500	1800	413	2205	409	2425
Vocal i	400	2000	300	2364	314	2425
Vocal o	500	1000	526	827	503	976
Vocal u	400	700	300	711	283	755

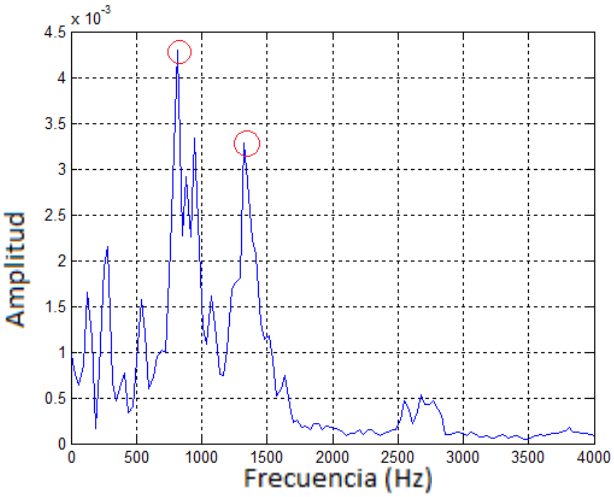
Figura 41: Carta de formantes para tres personas distintas, dos hombres una mujer.



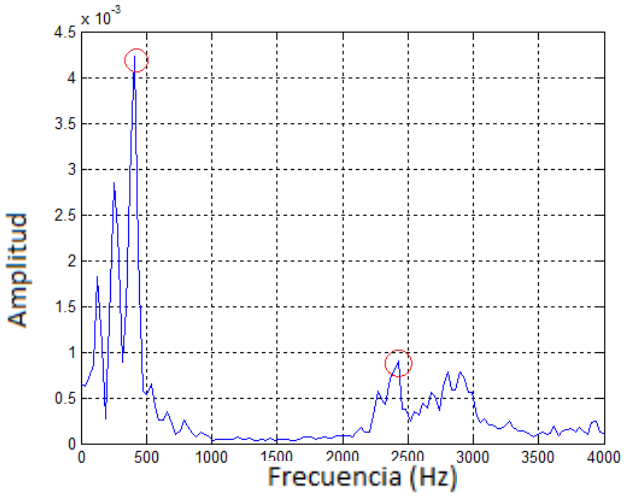
En la carta de formantes cada color representa una persona, excepto el color azul que es la de referencia. Además los círculos no delimitan por ningún motivo las frecuencias dentro de las cuales deben estar los formantes, están allí con el único propósito de agrupar los formantes de cada vocal para facilitar la lectura del gráfico.

También se muestran en las siguientes imágenes los formantes calculados por el algoritmo, los valores referentes a estos formantes son los que aparecen en la tabla anterior como *Pruebas 2*.

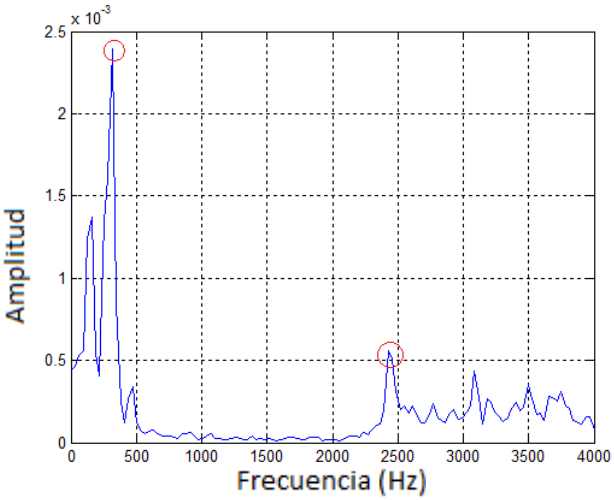
Figura 42: Espectro de las vocales y sus formantes.



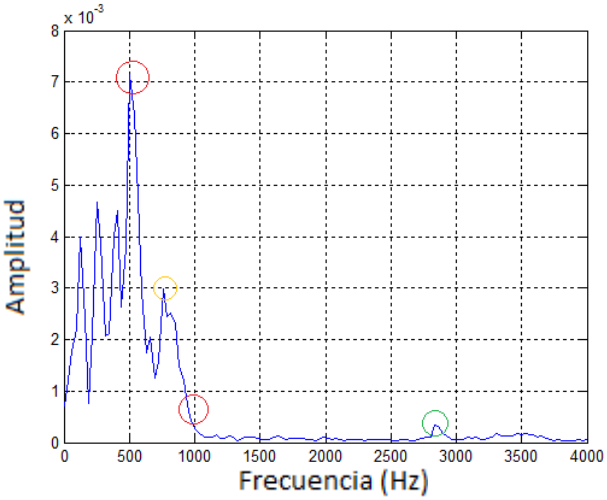
Espectro de la vocal “a” (818 y 1322)



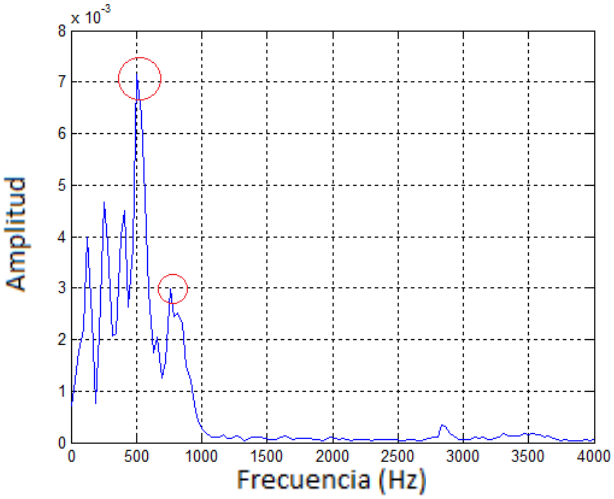
Espectro de la vocal “e” (409 y 2425)



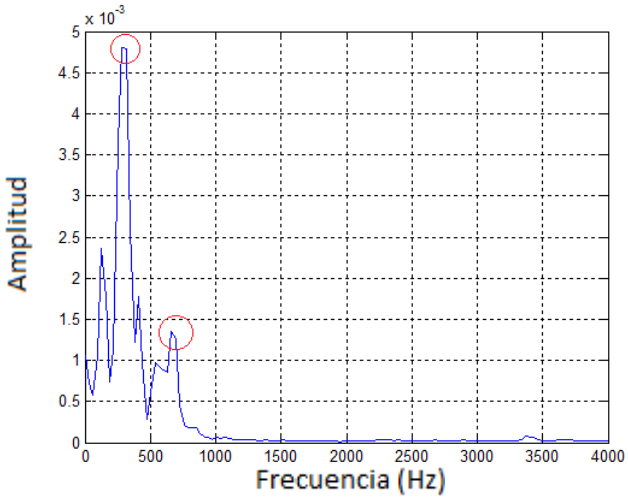
Espectro de la vocal “i” (314 y 2425)



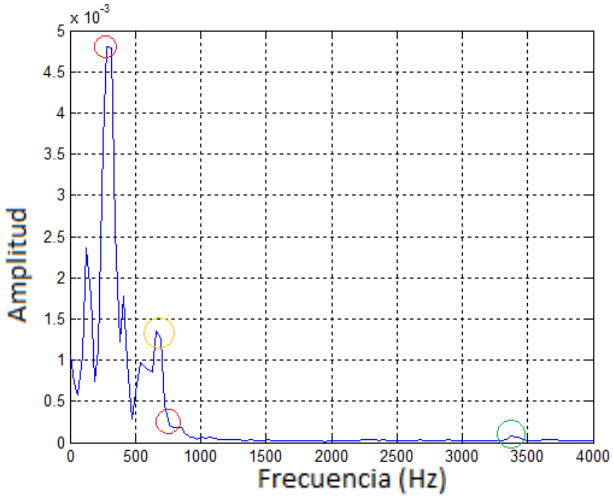
Espectro de la vocal “o” ** (503 y 976)



Espectro de la vocal “o” (503 y 818)



Espectro de la vocal “u” (283 y 661)



Espectro de la vocal “u” ** (283 y 755)

** NOTA La razón por la cual estas dos vocales tienen dos graficas es para demostrar lo que sucede cuando utilizando este método de cálculo de formantes (eliminando cierta cantidad de frecuencias y volver a calcular el pico máximo de la señal restante) se omiten más frecuencias de las necesarias. En ambas vocales el segundo formante no aparece en un pico donde se supone se encontraría (círculo amarillo), esto es debido al diseño del algoritmo ya que esa es la siguiente muestra de mayor amplitud respecto a la señal restante, lo que significa que no está tomando el siguiente pico visible (círculo verde) porque la amplitud de este es menor a la de la muestra encerrada en el círculo rojo.

Estos dos valores son arrojados por una eliminación de frecuencias de 473Hz o 15 muestras de la variable de frecuencia de la transformada de Fourier, pero al realizar un cambio en el algoritmo por una eliminación no de 15 sino de 7 muestras o lo que es igual a 189 Hz, los resultados mejoran arrojando las otras dos graficas de las vocales “o” y “u”.

Es importante decir que estos valores no siempre son los mismos, sino que tienen que ser hallados por medio de ensayo y error.

4.5.6 Reconociendo las palabras

Esta es la última etapa del proceso donde el algoritmo compara los formantes obtenidos de la señal grabada con los umbrales que se establecieron para cada formante de cada palabra, y de esta manera poder seleccionar una de las 9 palabras o decir que no es ninguna. Pero como se obtuvieron estos umbrales, es un proceso que requiere mucho tiempo y paciencia y se describe a continuación.

Después de tener el algoritmo con los parámetros establecidos y saber que se puede confiar en los cálculos del mismo, resta averiguar cuáles son los formantes para cada una de las 9 palabras y sus límites o umbrales para que no se confundan con otra palabra.

Fue necesario realizar un número significativo de grabaciones iníciales con el fin de determinar unos valores máximos y mínimos aproximados dentro de los cuales están ubicados esos formantes.

A continuación una tabla conteniendo los formantes calculados para cada silaba de la palabra “Mesa” pronunciada en 20 ocasiones.

Tabla 12: 20 repeticiones de la palabra “Mesa” con sus respectivos formantes para cada una de las dos silabas.

	Silaba 1		Silaba 2			Silaba 1		Silaba 2	
	Formante 1 (Hz)	Formante 2 (Hz)	Formante 1 (Hz)	Formante 2 (Hz)		Formante 1 (Hz)	Formante 2 (Hz)	Formante 1 (Hz)	Formante 2 (Hz)
1	535	2834	692	1511	11	503	2551	566	1322
2	535	2267	598	1574	12	503	2708	692	1417
3	535	2677	661	1417	13	503	2677	850	1574
4	535	2645	692	1480	14	503	2141	755	1511
5	535	2677	692	1448	15	472	2204	692	1511
6	535	2110	661	1480	16	535	2236	566	1574
7	535	2771	661	1448	17	503	2204	629	1543
8	503	2740	598	1480	18	503	2173	598	1511
9	535	2141	566	1511	19	503	2173	598	1574
10	503	2204	692	1511	20	535	2204	661	1511

Tabla 13: Valores máximos y mínimos para los formantes de la palabra “mesa”.

	Formante 1 (Hz)	Formante 2 (Hz)	Formante 1 (Hz)	Formante 2 (Hz)
Max	535	2834	850	1574
Min	472	2110	566	1322

Es necesario recordar que el algoritmo elimina unas frecuencias después de calcular el primer formante para luego calcular el segundo, este valor de 473 Hz se mencionó con anterioridad y ya se mencionó que es un número hallado por medio del ensayo y error, a lo que se quiere llegar es que no solo se ensayó y erró con una palabra, fue necesario calcular los formantes para varias palabras y verificar que en todas estuviera funcionando bien eliminando el mismo número de frecuencias, si por algún motivo las frecuencias no daban valores aceptables era necesario ensayar de nuevo con otro valor hasta llegar a un punto adecuado.

Después de los valores iníciales, los formantes se vuelven a verificar haciendo pruebas en entornos con más reverberación o con un mayor ruido de fondo. Finalmente cuando se tienen establecidos los umbrales de los formantes para cada palabra se procede a la verificación manual revisando que no se crucen o no se confundan unas palabras con otras.

Por último solo queda determinar los umbrales en el algoritmo y hacer las pruebas pertinentes para poder adquirir el porcentaje de reconocimiento por palabra y de la aplicación en general, este procedimiento es explicado un punto más adelante.

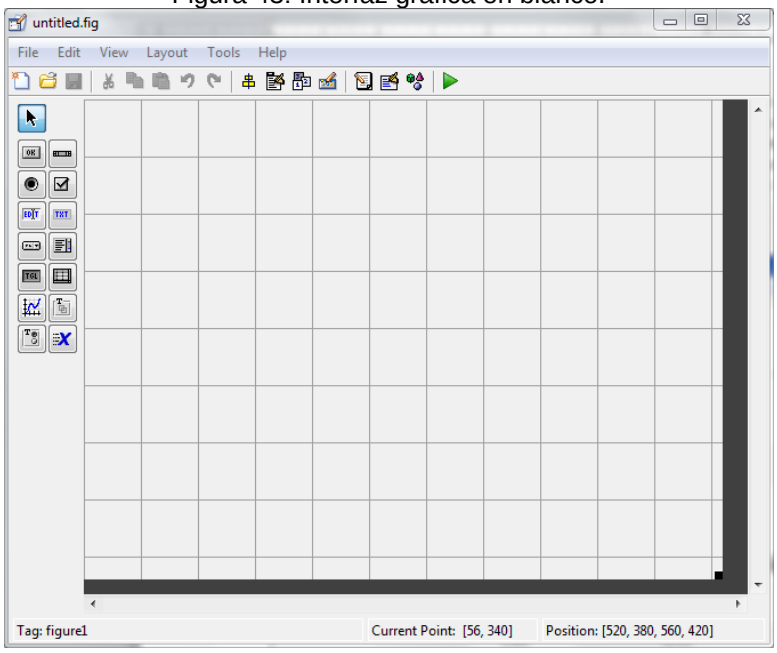
Ver el **Anexo C** para consultar el código.

El código del **Anexo C** es el algoritmo con el cual se realizaron todas las pruebas y se obtuvieron todos los resultados finales del presente trabajo. Como bien se menciona uno de los objetivos específicos es desarrollar una interfaz gráfica de usuario, esta interfaz y su desarrollo se muestran a continuación.

4.6 REALIZACION DE LA INTERFAZ GRAFICA

Gracias a la herramienta GUIDE de MATLAB un algoritmo puede ser utilizado mediante una interfaz gráfica. El primero de los pasos es crear una interfaz en blanco escribiendo el comando "guide" en la ventana de comandos de MATLAB o con el icono del mismo nombre ubicado en la barra de accesos directos, y luego se le indica crear nueva interfaz gráfica de usuario, apareciendo una ventana de edición donde se colocaran los elementos del programa. Véase **figura 43**

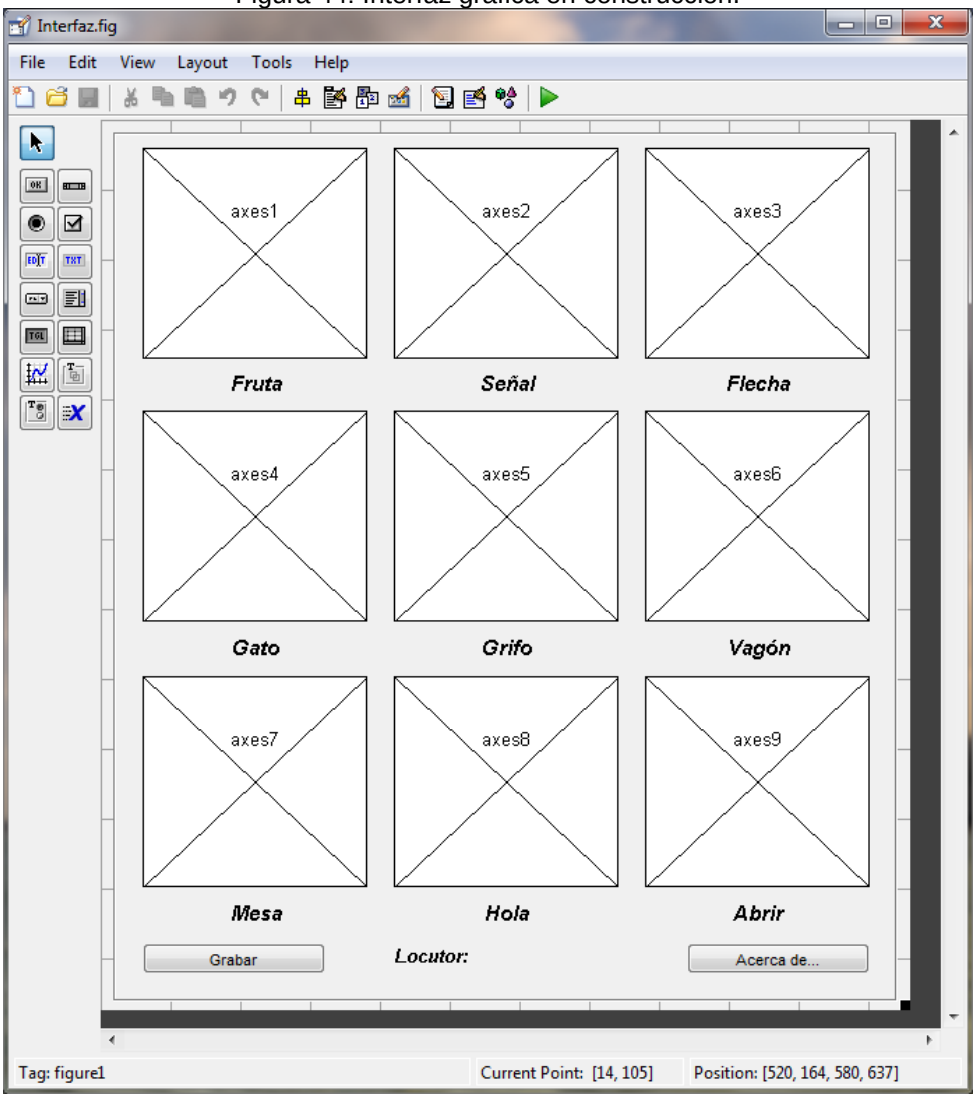
Figura 43: Interfaz gráfica en blanco.



En la cuadricula de la interfaz se ubicaran los botones, textos, imágenes y el resto de los objetos que contendrá el programa, los objetos pueden ser arrastrados desde la barra lateral izquierda hasta un espacio en blanco, donde posteriormente se le asignara un nombre y un formato, tamaño, color, color de fuente y otras propiedades específicas de cada objeto. Para realizar estos cambios basta con hacer doble clic sobre el objeto.

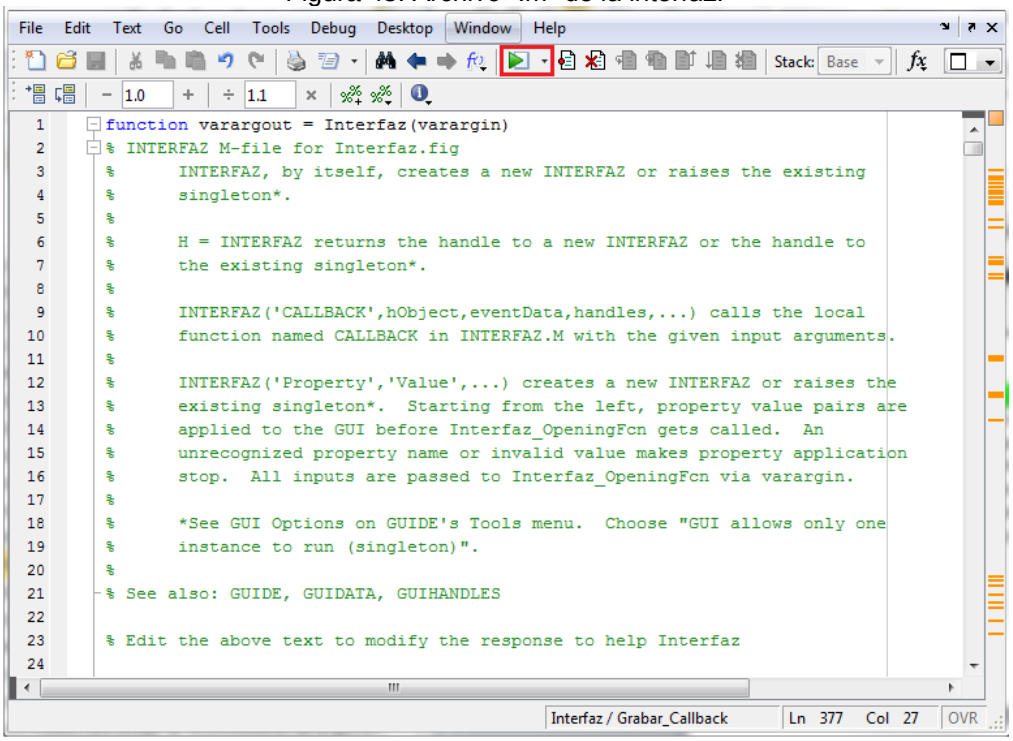
Para este caso, el resultado de la interfaz en este punto del proceso es como el que se muestra en la figura siguiente.

Figura 44: Interfaz gráfica en construcción.



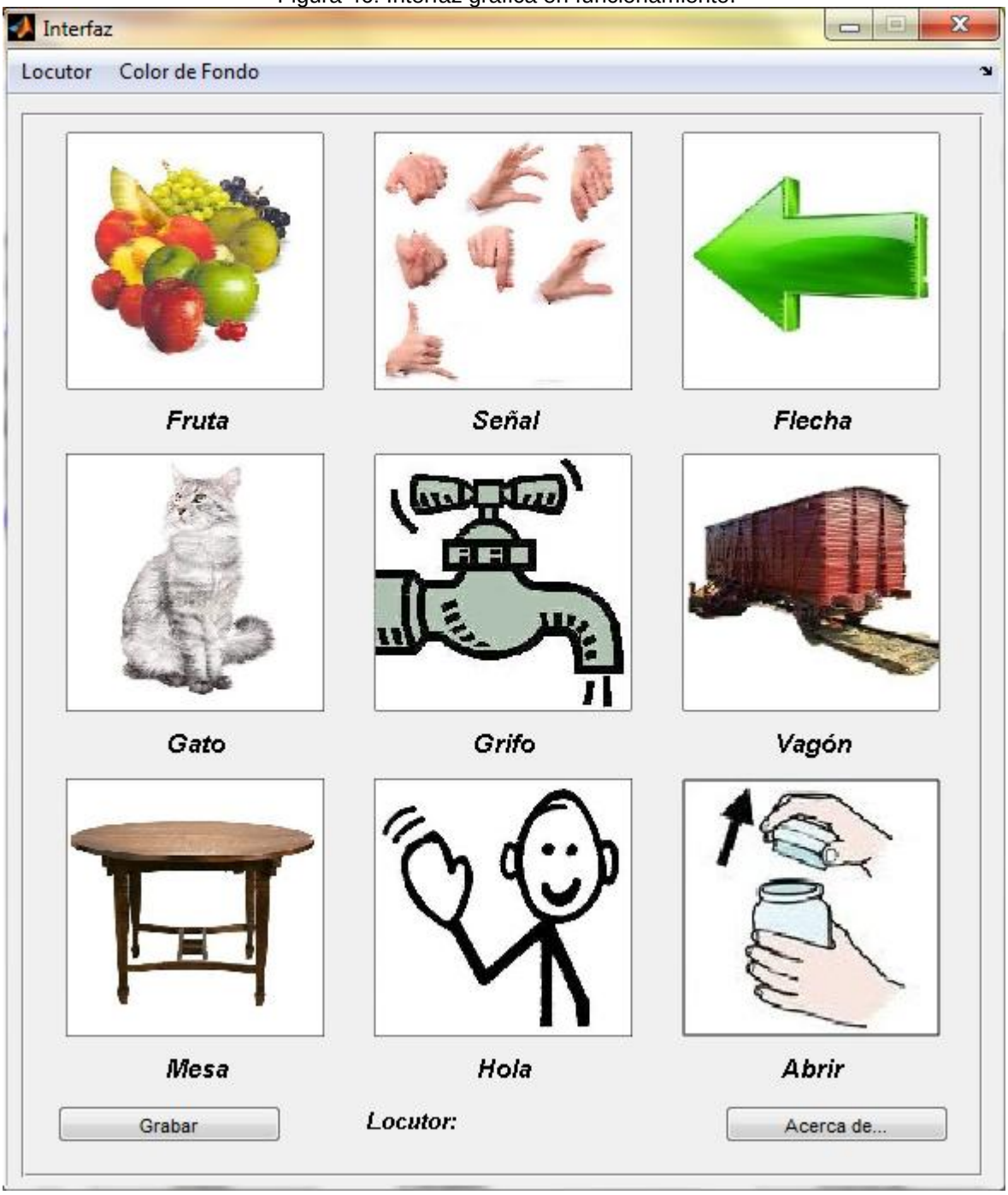
Observe que la **figura 44** tiene el nombre “interfaz” con extensión “.fig”. Esta es la extensión de los archivos de MATLAB relacionados con una interfaz gráfica y van asociados a un archivo “.m” con el mismo nombre que es creado al momento de guardar por primera vez el archivo “.fig”, estos dos archivos están enlazados y por lo tanto deben estar guardados en la misma carpeta al igual que las imágenes que se pretendan usar en la interfaz, siendo el archivo “.m” el que contiene el algoritmo que funciona por medio de la interfaz, un ejemplo de un archivo “.m” se muestra en la siguiente figura; para consultar el archivo “.m” dirigirse al **Anexo F**.

Figura 45: Archivo “.m” de la interfaz.



Para verificar como se ve la interfaz en funcionamiento, se debe lanzar o correr la aplicación oprimiendo el icono del recuadro rojo de la **figura 45** o simplemente se presiona la tecla de funcion F5, el resultado será similar a la figura siguiente.

Figura 46: Interfaz gráfica en funcionamiento.



La **figura 46** es el resultado de todos los procesos antes mencionados.

5 ANALISIS DE RESULTADOS

En los numerales 5.1 y 5.2 se presentan los resultados obtenidos por medio de redes neuronales, y en el numeral 5.3 los resultados obtenidos con el método escogido, reconocimiento de patrones siendo este el utilizado para el desarrollo del algoritmo.

5.1 ANÁLISIS DE DIFERENTES TIPOS DE ENTRENAMIENTO Y NÚMERO DE NEURONAS EN LAS CAPAS OCULTAS

Se realizaron dos pruebas con el fin de determinar cuál era el mejor algoritmo de entrenamiento (*Ver sección 2.4.1*) y el número de neuronas de las capas ocultas. Estos fueron los resultados:

Siendo las entradas y salida de la red respectivamente:

Tabla 14: Entrada de la red.

Entrada de la red			
F1 (Hz)	F2 (Hz)	F3 (Hz)	F4 (Hz)
724	1417	661	3118
692	2803	692	2866
661	1322	629	3212

Tabla 15: Salida deseada de la red.

Salida deseada de la red (código)			
1	0	0	0

Se realizó la simulación con los siguientes valores de entrada:

Tabla 16: Valores de entrada de la simulación - entrada 4.

Valores de entrada de la simulación - entrada 4 (Hz)			
F1 (Hz)	F2 (Hz)	F3 (Hz)	F4 (Hz)
692	2803	724	2960

Los resultados obtenidos con cada algoritmo de entrenamiento se presentan a continuación:

Tabla 17: Resultados con diferentes algoritmos de entrenamiento.

Pruebas con los diferentes algoritmo de entrenamiento							
Algoritmo de entrenamiento	Entrenamientos con las tres entradas y simulación	mse (error cuadrático medio)	Resultados Simulación F1	Resultados Simulación F2	Resultados Simulación F3	Resultados Simulación F4	Promedio de iteraciones durante el entrenamiento
Trainbfg	Entrenamiento 1	8,00E-09	1,00E+00	1,00E-04	-1,00E-04	-3,00E-05	8 iteraciones
	Entrenamiento 2	5,00E-09	1,00E+00	-3,00E-05	-8,00E-05	-1,00E-04	
	Entrenamiento 3	3,00E-11	1,00E+00	-6,00E-06	-6,00E-06	-8,00E-08	
	Simulación con la entrada 4		1,00E+00	-1,00E-01	1,00E+00	2,00E-01	
Traingdm	Entrenamiento 1	7,00E-05	1,00E+00	1,00E-03	1,00E-02	-1,00E-03	Más de 500 iteraciones
	Entrenamiento 2	1,00E-02	9,00E-01	-5,00E-03	1,00E-01	6,00E-03	
	Entrenamiento 3	4,00E-03	9,00E-01	-5,00E-03	9,00E-02	1,00E-02	
	Simulación con la entrada 4		1,00E+00	-2,00E-01	1,00E+00	6,00E-01	
Trainrp	Entrenamiento 1	1,00E-09	1,00E+00	2,00E-05	-3,00E-05	1,00E-05	26 iteraciones
	Entrenamiento 2	7,00E-09	1,00E+00	2,00E-04	1,00E-05	-5,00E-05	
	Entrenamiento 3	5,00E-08	1,00E+00	2,00E-04	4,00E-04	1,00E-04	
	Simulación con la entrada 4		1,00E+00	4,00E-02	1,00E+00	-2,00E-02	

Los resultados de la tabla muestran que el algoritmo Trainbfg convergió en un menor número de iteraciones alcanzando un error cuadrático medio bajo. Por otra parte al realizar las simulaciones el algoritmo que presenta una salida más parecida a la salida deseada efectivamente es Trainbfg.

Una segunda prueba se realizó con el fin de determinar el número de neuronas en las capas oculta a partir de los resultados mostrados a continuación. El resultado determinó que había un mejor funcionamiento de la red cuando en la primera capa había 10 neuronas y en la segunda 25. Salida deseada: [1 0 0 0].

Tabla 18: Resultados con diferentes números de neuronas en la capa oculta.

Pruebas diferentes números de neuronas utilizando el algoritmo Trainbfg						
Número de Neuronas en la capa 1	Número de Neuronas en la capa 2	mse (error cuadrático medio)	Resultados Simulación F1	Resultados Simulación F2	Resultados Simulación F3	Resultados Simulación F4
10	25	0	1	0	0	0
		0	1	0	0	0
		0	1	0	0	0
		Simulación entrada 4	1	0	0	0
Número de Neuronas en la capa 1	Número de Neuronas en la capa 2	mse (error cuadrático medio)	Resultados Simulación F1	Resultados Simulación F2	Resultados Simulación F3	Resultados Simulación F4
2	5	0	1	0	0	0
		0	1	0	0	0
		0,13	0,5	0	0,5	0
		Simulación entrada 4	0,5	0	0,5	0
Número de Neuronas en la capa 1	Número de Neuronas en la capa 2	mse (error cuadrático medio)	Resultados Simulación F1	Resultados Simulación F2	Resultados Simulación F3	Resultados Simulación F4
30	53	0	1	0	0	0
		0	1	0	0	0
		0	1	0	0	0
		Simulación entrada 4	1	-0,04	1,42	0,01

5.2 PORCENTAJE DE ACIERTO USANDO LA RED NEURONAL BACK PROPAGATION

En esta sección solo se muestran los resultados obtenidos en el proceso de reconocimiento de un locutor femenino con unos umbrales establecidos, las conclusiones serán dadas a conocer más adelante.

Estos resultados describen el porcentaje de acierto utilizando el método de redes neuronales. Un SI en la tabla establece que la palabra fue reconocida, un NO establece que esta no fue reconocida o fue confundida con alguna de las otras palabras.

Tabla 19: Resultados pruebas de reconocimiento con redes neuronales.

Fruta		Señal		Flecha		Gato		Grifo		Vagón		Mesa		Hola		Abrir		
1	NO	1	NO	1	NO	1	NO	1	NO	1	NO	1	NO	1	NO	1	NO	
2	SI	2	NO	2	NO	2	Si	2	NO	2	NO	2	NO	2	NO	2	NO	
3	NO	3	NO	3	NO	3	Si	3	NO	3	NO	3	Si	3	NO	3	NO	
4	NO	4	NO	4	NO	4	Si	4	NO	4	NO	4	NO	4	NO	4	NO	
5	NO	5	NO	5	NO	5	Si	5	NO	5	NO	5	Si	5	NO	5	NO	
6	NO	6	NO	6	NO	6	NO	6	NO	6	Si	6	NO	6	NO	6	NO	
7	NO	7	NO	7	NO	7	NO	7	NO	7	NO	7	NO	7	NO	7	NO	
8	NO	8	NO	8	NO	8	NO	8	NO	8	NO	8	Si	8	Si	8	NO	
9	NO	9	NO	9	Si	9	NO	9	NO	9	NO	9	Si	9	NO	9	NO	
10	NO	10	NO	10	Si	10	NO	10	NO	10	NO	10	Si	10	NO	10	NO	
																		Total
%	10%		0%		20%		40%		0%		10%		50%		10%		0%	14%

5.3 MEDIANTE EL METODO DE RECONOCMIENTO DE PATRONES

En esta sección se muestran las tablas conteniendo los valores de formantes para las nueve palabras para una persona específica. Además se muestran los resultados obtenidos por parte de dos locutores, en las pruebas de reconocimiento, uno femenino y otro masculino.

5.3.1 Formantes de las nueve palabras obtenidos por un locutor masculino

Los siguientes son los formantes obtenidos para el locutor masculino, con los máximos y mínimos donde puede encontrarse este formante.

Tabla 20: Valores de las formantes de las nueve palabras.

	Formantes (Hz)			
	F1	F2	F3	F4
GATO	755	1480	472	1385
	724	1448	440	1165
	535	1322	251	1511
	755	1354	535	1291
	724	1417	472	1417
	787	1448	472	1543
	692	1511	377	1354
	818	1511	440	1228
	629	1322	472	1322
	472	1385	472	1133
	692	1385	440	1162
	787	1417	472	1448
	724	1417	409	1228
	755	1354	472	1291
	724	1448	440	1417
	724	1574	472	1162
	724	1448	440	1165
	755	1354	472	1291
	755	1511	472	1385
	724	976	440	1102
Max	818	1574	535	1543
Min	472	976	251	1102

	Formantes (Hz)			
	F1	F2	F3	F4
MESA	535	2834	692	1511
	535	2267	598	1574
	535	2677	661	1417
	535	2645	692	1480
	535	2677	692	1448
	535	2110	661	1480
	535	2771	661	1448
	503	2740	598	1480
	535	2141	566	1511
	503	2204	692	1511
	503	2551	566	1322
	503	2708	692	1417
	503	2677	850	1574
	503	2141	755	1511
	472	2204	692	1511
	535	2236	566	1574
	503	2204	629	1543
	503	2173	598	1511
	503	2173	598	1574
	535	2204	661	1511
Max	535	2834	850	1574
Min	472	2110	566	1322

	Formantes (Hz)			
	F1	F2	F3	F4
FRUTA	472	1039	566	1511
	472	1448	503	1574
	503	1480	629	1511
	472	1196	535	1574
	472	1259	472	1543
	472	1291	661	1511
	472	1354	629	1543
	472	1354	566	1511
	472	1322	629	1511
	472	1448	598	1480
	566	1133	598	1637
	503	1259	535	1448
	535	1354	472	1700
	503	1385	535	1511
	566	1228	598	1543
	503	1448	535	1574
	472	1228	598	1574
	503	1448	629	1606
	472	1385	629	1511
	472	1291	566	1543
Max	566	1480	661	1700
Min	472	1039	472	1448

	Formantes (Hz)			
	F1	F2	F3	F4
HOLA	472	1007	535	1732
	503	976	566	1606
	503	1039	566	1637
	503	1007	535	1606
	503	1007	535	1637
	472	976	535	1606
	503	1007	566	1637
	503	1133	566	1732
	503	1133	346	1637
	503	1039	566	1669
	503	1070	566	1700
	503	1039	535	2456
	503	1039	346	2645
	503	976	346	1732
	535	1196	692	1826
	535	1007	598	1732
	535	1007	566	1543
	535	1070	692	1732
	503	1039	535	1669
	535	1070	598	1700
Max	535	1196	692	2645
Min	472	976	346	1543

	Formantes (Hz)			
	F1	F2	F3	F4
ABRIR	850	1385	346	2645
	850	1448	377	2299
	787	1385	472	2393
	850	1354	283	2299
	503	1417	377	2267
	913	1385	377	2236
	850	1291	377	2456
	881	1354	346	2173
	787	1385	346	2330
	787	1448	283	2866
	787	1417	377	1795
	503	1354	346	2456
	818	1417	346	2582
	755	1259	283	2425
	881	1354	377	2393
	850	1196	472	1826
	818	1417	346	2708
	787	1354	283	2866
	787	1354	346	2141
	850	1417	346	2330
Max	913	1448	472	2866
Min	503	1196	283	1795

	Formantes (Hz)			
	F1	F2	F3	F4
SEÑAL	503	1921	724	1669
	503	2047	724	1795
	472	2519	724	1700
	472	2582	724	1763
	472	2488	692	1826
	472	2519	692	1700
	472	2488	724	1669
	472	2582	692	1763
	472	2110	692	1606
	472	2645	692	1795
	535	2015	629	1795
	472	1826	724	1606
	566	1858	755	1637
	472	2771	724	1700
	535	2110	755	1637
	472	1952	724	1669
	503	1984	724	1732
	472	2519	724	1669
	503	2047	692	1763
	472	1921	724	1669
Max	566	2771	755	1826
Min	472	1826	629	1606

	Formantes (Hz)			
	F1	F2	F3	F4
VAGON	755	1385	503	1007
	724	1417	503	1007
	724	1385	535	976
	692	1322	472	944
	692	1354	472	944
	692	1354	503	1007
	692	1354	472	976
	692	1291	472	976
	787	1480	472	976
	692	1322	472	944
	787	1291	472	944
	724	1196	503	1007
	692	1291	472	976
	787	1291	503	1039
	629	1228	377	881
	724	1385	377	850
	692	1354	503	1007
	692	1354	503	1007
	724	1385	503	1007
	724	1291	503	976
Max	787	1480	535	1039
Min	629	1196	377	850

	Formantes (Hz)			
	F1	F2	F3	F4
GRIFO	2488	3496	440	913
	2078	2677	409	976
	2456	3496	440	913
	2267	3779	377	2676
	2204	2740	440	913
	2929	2708	440	944
	2047	2645	440	913
	1826	2803	409	1448
	2551	2803	440	787
	2236	2708	440	1448
	2110	2582	472	944
	2141	1921	472	2676
	2204	3653	409	881
	2204	1984	472	944
	2582	2740	283	944
	2110	3748	440	913
	724	2645	440	913
	472	2677	472	944
	629	2299	472	944
	598	2015	346	818
Max	2929	3779	472	2676
Min	1826	1921	283	787
Max 2	724			
Min 2	472			

	Formantes (Hz)			
	F1	F2	F3	F4
FLECHA	503	2015	2866	3905
	472	2047	2866	3874
	503	1921	2866	3685
	503	1952	2929	3779
	535	2519	2866	3716
	535	2645	2803	3685
	535	2740	2803	3842
	503	1952	2834	3779
	472	2078	2960	4000
	503	2929	2740	2866
	503	3149	2897	3622
	503	1699	2708	3433
	535	1795	2897	3653
	503	2015	2803	4000
	503	1921	2929	3748
	503	1952	283	2771
	503	2015	314	2740
	535	2110	283	2708
	535	1984	535	2960
	535	1984	377	3685
Max	535	3149	2960	4000
Min	472	1699	2708	2708
Max 2			535	
Min 2			283	

La **tabla 20** muestra que existen dos valores de máximos en dos de las palabras, estos se incluyen en el algoritmo para mejorar el porcentaje de reconocimiento siempre y cuando estos valores no intervengan con las otras palabras (valores resaltados en azul). Al igual que en el algoritmo del **Anexo C** se encuentran rangos ampliados en algunas ocasiones, estos datos adicionales son cuidadosamente añadidos para que no interfieran con las demás palabras y puedan aumentar el porcentaje de reconocimiento.

5.3.2 Pruebas acierto y error con reconocimiento de patrones

Las pruebas llevadas a cabo por el locutor masculino están divididas en dos partes, la primera es donde se evalúa el porcentaje de reconocimiento de cada palabra por separado realizando 50 repeticiones de la misma, y la segunda donde se hacen 100 repeticiones de palabras al azar para determinar el porcentaje de reconocimiento general de la aplicación.

Tabla 21: Porcentaje de reconocimiento para cada palabra.

PRUEBA FRUTA									
1	1	11	1	21	1	31	1	41	0
2	1	12	1	22	1	32	1	42	1
3	1	13	1	23	1	33	1	43	1
4	1	14	8	24	1	34	1	44	1
5	0	15	0	25	1	35	8	45	1
6	1	16	1	26	5	36	1	46	1
7	1	17	1	27	1	37	1	47	1
8	0	18	1	28	1	38	1	48	0
9	8	19	1	29	1	39	1	49	1
10	1	20	1	30	1	40	1	50	1

Bien	41	Mal	9	%	**82%**

PRUEBA SEÑAL									
1	1	11	0	21	1	31	0	41	0
2	1	12	1	22	1	32	1	42	0
3	1	13	1	23	1	33	1	43	1
4	1	14	0	24	1	34	1	44	7
5	0	15	1	25	1	35	1	45	1
6	1	16	1	26	1	36	0	46	0
7	1	17	1	27	1	37	1	47	1
8	1	18	1	28	1	38	1	48	1
9	1	19	1	29	1	39	1	49	7
10	1	20	1	30	1	40	0	50	7

Bien	38	Mal	12	%	**76%**

PRUEBA FLECHA									
1	1	11	1	21	1	31	1	41	1
2	1	12	1	22	1	32	1	42	1
3	1	13	1	23	1	33	1	43	1
4	1	14	1	24	1	34	1	44	1
5	0	15	1	25	1	35	1	45	1
6	1	16	1	26	0	36	1	46	1
7	1	17	1	27	1	37	1	47	1
8	1	18	1	28	1	38	1	48	1
9	1	19	1	29	1	39	1	49	1
10	1	20	1	30	1	40	1	50	1

Bien	48	Mal	2	%	**96%**

PRUEBA GATO									
1	1	11	1	21	6	31	1	41	1
2	0	12	1	22	1	32	1	42	1
3	0	13	0	23	0	33	1	43	1
4	1	14	1	24	6	34	0	44	1
5	1	15	1	25	1	35	1	45	6
6	1	16	1	26	6	36	1	46	1
7	1	17	0	27	0	37	1	47	9
8	1	18	1	28	6	38	1	48	1
9	1	19	1	29	1	39	0	49	1
10	1	20	1	30	1	40	0	50	1

Bien	35	Mal	15	%	**70%**

PRUEBA GRIFO									
1	1	11	1	21	1	31	1	41	0
2	1	12	1	22	1	32	1	42	1
3	1	13	0	23	1	33	1	43	1
4	1	14	1	24	1	34	1	44	0
5	1	15	1	25	1	35	1	45	1
6	1	16	0	26	0	36	1	46	1
7	1	17	0	27	1	37	0	47	1
8	1	18	1	28	1	38	1	48	1
9	1	19	1	29	1	39	1	49	1
10	1	20	3	30	1	40	1	50	1

Bien	42	Mal	8	%	**84%**

PRUEBA VAGON									
1	1	11	1	21	1	31	1	41	1
2	1	12	1	22	1	32	1	42	1
3	1	13	1	23	1	33	1	43	1
4	0	14	1	24	1	34	1	44	1
5	1	15	1	25	1	35	1	45	1
6	1	16	1	26	1	36	1	46	1
7	0	17	1	27	1	37	1	47	1
8	1	18	1	28	1	38	1	48	1
9	1	19	1	29	1	39	1	49	1
10	1	20	1	30	1	40	1	50	1

Bien	48	Mal	2	%	**96%**

PRUEBA MESA									
1	1	11	1	21	1	31	5	41	2
2	1	12	1	22	1	32	2	42	1
3	1	13	1	23	1	33	1	43	1
4	1	14	1	24	1	34	2	44	2
5	1	15	0	25	1	35	1	45	0
6	1	16	1	26	1	36	2	46	1
7	1	17	1	27	1	37	2	47	0
8	1	18	1	28	0	38	0	48	1
9	1	19	1	29	1	39	0	49	1
10	1	20	1	30	0	40	1	50	5

Bien	35	Mal	15	%	**70%**

PRUEBA HOLA									
1	1	11	1	21	0	31	1	41	1
2	1	12	1	22	1	32	0	42	1
3	1	13	0	23	0	33	1	43	1
4	0	14	0	24	1	34	1	44	0
5	1	15	1	25	1	35	1	45	0
6	1	16	1	26	0	36	1	46	1
7	1	17	1	27	1	37	1	47	0
8	1	18	1	28	0	38	1	48	1
9	1	19	1	29	0	39	0	49	1
10	1	20	1	30	1	40	0	50	1

Bien	36	Mal	14	%	**72%**

PRUEBA ABRIR									
1	1	11	1	21	1	31	1	41	1
2	1	12	1	22	1	32	1	42	1
3	1	13	1	23	1	33	1	43	1
4	1	14	1	24	1	34	1	44	1
5	1	15	1	25	1	35	1	45	0
6	1	16	0	26	1	36	1	46	1
7	1	17	0	27	1	37	1	47	1
8	4	18	1	28	1	38	1	48	0
9	1	19	1	29	1	39	1	49	1
10	1	20	1	30	1	40	1	50	1

Bien	45	Mal	5	%	**90%**

En la **tabla 21** los números 1 indican que el algoritmo acertó en el reconocimiento, los números 0 indican que el algoritmo no reconoció la palabra, y cuando se encuentra otro número en la tabla indica que la palabra fue confundida con otra que no era la correcta.

5.3.3 Prueba y porcentaje de reconocimiento de la aplicación en general.

Ahora veamos las pruebas con todas las palabras al mismo tiempo y el porcentaje general de la aplicación.

Tabla 22: Prueba final, porcentajes de la aplicación para un locutor masculino.

PRUEBA TOTAL																			
1	1	11	1	21	1	31	1	41	4	51	1	61	8	71	1	81	1	91	1
2	1	12	1	22	0	32	1	42	1	52	1	62	1	72	7	82	1	92	1
3	1	13	1	23	4	33	1	43	1	53	1	63	1	73	1	83	1	93	4
4	1	14	9	24	6	34	9	44	1	54	9	64	1	74	1	84	1	94	1
5	1	15	1	25	1	35	1	45	1	55	1	65	1	75	1	85	1	95	1
6	1	16	1	26	1	36	1	46	1	56	1	66	1	76	1	86	1	96	1
7	1	17	1	27	1	37	1	47	1	57	5	67	1	77	2	87	1	97	1
8	1	18	0	28	0	38	1	48	0	58	1	68	0	78	9	88	0	98	0
9	1	19	1	29	1	39	1	49	1	59	1	69	1	79	1	89	1	99	1
10	1	20	0	30	0	40	6	50	1	60	1	70	0	80	1	90	1	100	1

10		7		5		8		8		8		7		7		9		8	77
Palabras acertadas en grupos de diez																			

Porcentaje de reconocimiento general: 77 %

Las indicaciones son las mismas que para la prueba por cada palabra, dando un porcentaje de reconocimiento final de cerca del 77% este es un valor aproximado porque no siempre se dará el mismo valor, puede que algunas veces aumente o en otras disminuya, esto depende de las palabras que se repitan con más frecuencia ya que todas no tienen el mismo porcentaje como se mostró en la **sección 5.3.2**.

Adicionalmente se tiene el número de palabras acertadas en grupos de diez, estos datos son útiles para tener una idea del rango del porcentaje de reconocimiento, es decir, en algunas ocasiones podremos tener un 100 % (reconocer 10 de 10), pero en el peor de los casos tener un reconocimiento del 50 % (5 de 10), aunque por lo general se obtendrán porcentajes entre el 70 % y 80 % (7 u 8 de 10), y en algunos casos un 90 % (9 de 10). Esto también se puede evidenciar en los porcentajes por palabra de la **tabla 22**.

Tabla 23: Porcentajes de reconocimiento por palabra y general para un locutor femenino.

PALABRA	PORCENTAJE DE ACIERTO
Fruta	90%
Señal	75%
Flecha	75%
Gato	75%
Grifo	90%
Vagón	80%
Mesa	95%
Hola	80%
Abrir	85%
General	75%

La **tabla 23** presenta los porcentajes de reconocimiento del algoritmo en general y por palabras para un locutor femenino, en este algoritmo los valores de los formantes fueron calculados para este locutor en particular. De acuerdo a los resultados se puede observar que los porcentajes en general se mantienen entre el 70% y 80%, llegando en algunos casos al 90%.

CONCLUSIONES

- El método de entrenamiento de la aplicación de reconocimiento de voz no debe ser determinado antes de desarrollarla ya que el desempeño del entrenamiento depende de la aplicación que se quiera implementar y a partir de las pruebas realizadas durante este proceso se podrá determinar la técnica más acertada.
- El método de reconocimiento de patrones puede ser usado en aplicaciones cuyo algoritmo de caracterización no presenta grandes diferencias en algunos de los valores extraídos de diferentes fonemas y los valores para cada hablante varían notablemente cuando se cambian las condiciones de grabación (sala, micrófono, etc.). De esta manera la determinación de umbrales permitirá obtener un mayor reconocimiento en comparación a las redes neuronales.
- El método de redes neuronales puede ser utilizado cuando el algoritmo de caracterización sea muy robusto y presente resultados realmente diferentes entre los fonemas pero mantenga unos valores relativamente constantes para cada fonema de un mismo hablante aún en diferentes condiciones de grabación. En este caso las redes neuronales podrán representar una ventaja porque aún cuando la red no ha sido entrenada con una entrada determinada se podrá producir la salida deseada ya que esta es una de las propiedades de las redes neuronales.
- En el desarrollo de esta aplicación se determinó que el método de reconocimiento de patrones permite alcanzar porcentajes de reconocimiento más altos para cada una de las nueve palabras en comparación al método de redes neuronales.
- En el caso de las redes neuronales se determinó trabajar directamente con la red Backpropagation, ya que es esta la que se menciona en los textos relacionados con este tema y así mismo es con la cual se han presentado buenos resultados en este tipo de aplicaciones.
- El algoritmo de detección de bordes con algunas modificaciones mínimas, en conjunto con el algoritmo de envolvente de energía, posibilitarían un mejor resultado al momento de separar palabras de dos silabas. En esta aplicación solo se utilizó el algoritmo de envolvente de energía ya que este era suficiente para separar las silabas en el proceso de caracterización.
- Si se decide utilizar el método de entrenamiento es necesario utilizar un gran número de entradas diferentes ya que esto aumenta notablemente el desempeño de la red. Es decir, al grabar varias veces cada palabra e ingresar estos datos a la red la probabilidad de obtener la salida deseada aumenta.
- El micrófono con el que se realiza la captura es importante para el proceso de extracción de características, ya que hay algunos que capturan más ruido que otros y esto dificulta este proceso.
- Al desarrollar una aplicación que reconozca un mayor número de palabras se necesita un algoritmo de caracterización más robusto, los resultados de esta etapa puede mejorarse añadiendo procesos a los ya utilizados, por ejemplo métodos de eliminación de ruido u otros de análisis en el dominio de la frecuencia y tiempo.
- Para poder tener un buen algoritmo y un reconocimiento por medio de redes neuronales, es necesario tener un buen entrenamiento, esto implica tener resultados confiables en la extracción de características, se dejan como dos opciones con el método de LPC y coeficientes cepstrales.
- Es posible llegar a tener una separación de tres o más silabas realizando las investigaciones y pruebas adecuadas, de esta manera se puede lograr el reconocimiento para un mayor número de palabra a través del método de reconocimiento de patrones.
- El porcentaje de reconocimiento de la aplicación oscila entre un 70% y 80%, llegando a presentar porcentajes del 90% o 60% en algunas palabras para determinados hablantes.
- Algunas palabras tienen un mejor porcentaje de reconocimiento que otras, por esta razón el porcentaje general de la aplicación puede variar entre el 70% y 80%.

- Es posible diseñar una aplicación de reconocimiento de voz a partir de redes neuronales, pero su desempeño dependerá directamente de las necesidades de la aplicación y de la robustez del método de caracterización de la palabra. En este caso no se obtuvieron porcentajes de reconocimiento altos a partir del método de redes neuronales ya que las diferencias entre los valores de los formantes eran mínimas entre algunas palabras y por esto mismo la red tiende a confundirlas.
- Determinar el inicio y fin de una palabra o la detección de bordes es una tarea de poca importancia en ambientes controlados, casos en los que el ruido de fondo es mínimo inclusive comparándolo con los sonidos de menor nivel energético llamados fricativas o segmentos sonoros débiles.
- La interfaz gráfica permite visualizar de manera más sencilla el funcionamiento de la aplicación.

RECOMENDACIONES

- Si se desean tener resultados más precisos se pueden manejar métodos de extracción y reconocimiento más robustos.
- Para este algoritmo se pueden plantear varias mejoras: optimización del proceso de separación de silabas, extracción de características no solo en frecuencia sino también en tiempo.
- Siempre es recomendable utilizar un micrófono con patrón direccional y no los micrófonos integrados que vienen con los computadores ya que con los primeros se obtienen mejores resultados en la caracterización de las palabras y así mismo en el reconocimiento de voz.
- Es importante que cada captura de palabra se realice en campo directo ya que de esta forma no se añade información innecesaria, por ejemplo, el comportamiento acústico de una sala.
- En el desarrollo de futuros proyectos en el campo del reconocimiento de voz es recomendable limitar el número de hablantes y palabras que reconocerá la aplicación más no es recomendable limitar el método por el cual esta será desarrollada ya que durante la investigación se pueden encontrar métodos que se adapten mejor a una aplicación determinada y no a otra.
- Para obtener un mejor porcentaje de reconocimiento en un hablante determinado es necesario entrenar el sistema para el uso de esa persona. De esta manera el sistema reconocerá las características que tienen la persona al pronunciar diferentes fonemas.
- En el momento de grabar las muestras que harán parte del entrenamiento del sistema es recomendable realizar varias grabaciones y en diferentes momentos del día o en diferentes días, esto debido a los cambios que se producen en los formantes cuando los fonemas son pronunciados de acuerdo a las distintas situaciones que hacen parte de la vida cotidiana (al despertar, al estar cansado, al estar feliz, etc.)
- Es recomendable concentrarse en un solo método de reconocimiento y extracción de características en otros espacios de investigación, por ejemplo en los proyectos integradores o grupos de investigación. De esta forma este tipo de trabajos y aplicaciones podrán ser desarrollados sobre bases más solidas, exploradas previamente con mayor profundidad.
- Es importante que la palabra que se quiera reconocer sea pronunciada de la forma más parecida posible a la utilizada durante la fase de entrenamiento.

BIBLIOGRAFÍA

Análisis de la señal de voz, Universidad de Extremadura, España

Articulo Sistema VLSC.
http://recedis.referata.com/wiki/Sistema_inteligente_de_reconocimiento_de_voz_para_la_traducci%C3%B3n_del_lenguaje_verbal_a_la_lengua_de_se%C3%B1as_colombiana_(VLSC**)**

Características de una Red Neuronal Artificial.
http://proton.ucting.udg.mx/posgrado/cursos/idc/neuronales2/Transferencia.htm

COLLADO, Esteban .http://poncos.freeiz.com/blog/, 2009.

MARTINEZ, Fernando; PORTALE, Gustavo; KLEIN, Hernán y OLMOS, Osvaldo. Reconocimiento de voz, apuntes de cátedra para Introducción a la Inteligencia Artificial.

MORENO, Asunción. La señal de voz, Universidad Politécnica de Cataluña, España

OROPEZA RODRIGUEZ, José Luis. Algoritmos y métodos para el reconocimiento de voz en español mediante sílabas. Computación y sistemas Vol. 9 Núm. 3, pp. 270-286, 2006.

ORTIZ RICO, Paola Milena y PARRA CARDENAS, Julie Andrea. Diseño e implementación de un Software de corrección fonética para niños con problemas de aprendizaje, 2007.

SANTOSA, Budi. Introduction to MATLAB Neural network Toolbox.

SANTOS-GARCIA, Gustavo. Inteligencia artificial y matemática aplicada, 2001

TAPIAS MERINO, Daniel. Sistemas de reconocimiento de voz en las telecomunicaciones, 1999

TREJOS POSADA, Hernando Antonio y URIBE PÉREZ, Carlos Andrés. Motor computacional de reconocimiento de voz: Principios básicos para su construcción, 2007.

Tutorial de Redes Neuronales. Universidad Tecnológica de Pereira. Facultad de ingeniería eléctrica. http://proton.ucting.udg.mx/posgrado/cursos/idc/neuronales2/

VELÁSQUEZ RAMÍREZ, Genoveva. Sistema de reconocimiento de voz, 2008

VILLORIA, Cristina. Reconocimiento y síntesis de voz.
http://observatorio.cnice.mec.es/modules.php?op=modload&name=News&file=article&sid=689

REFERENCIAS EN LA WEB

COLLADO, Esteban .http://poncos.freeiz.com/blog/ , 2009.

http://recedis.referata.com/wiki/Sistema_inteligente_de_reconocimiento_de_voz_para_la_traducci%C3%B3n_del_lenguaje_verbal_a_la_lengua_de_se%C3%B1as_colombiana_(VLSC**)**

Constitución de las palabras: sonidos, fonemas y letras,
http://roble.pntic.mec.es/~msanto1/lengua/1sofolet.htm

ANEXOS

Anexo A: Código en MATLAB del algoritmo de detección de bordes o inicio y final de palabra.

Una vez capturada la señal y almacenada en una variable, en este caso “palabra”.

```
npi=122;
niis=19; %# de intervalos solapados iniciales de silencio.
largo=length(palabra);
ni=(2*fix(largo/npi))-1; %Es el # de intervalos de la palabra (en cuantas
partes se dividio)
if ni<0
error('Relación inaceptable: PALABRA PEQUEÑA O INTERVALOS GRANDES');
end
E = zeros(1,ni); %Vector para energía de los intervalos
Z = zeros(1,ni); %Vector para pasos por ceros de los intervalos
for i=1:ni;
    L1=((npi/2)*(i-1))+1; %inicio de cada intervalo
    L2=(npi/2)*(i+1); %fin de cada intervalo
interv=palabra(L1:L2);
E(i)=sum(abs(interv));
% calcula los ceros
interv2=interv(2:length(interv));
interv(length(interv))=[];interv2=interv.*interv2;
Z(i)=sum(interv2<0)+sum(interv2==0)/4;
%Fin del calculo de los ceros.
end;
silencio=Z(1:niis); %Le indicamos cuantos intervalos bamos a evalluar
ZCMS=mean(silencio);
DZCS=std(silencio);
EMS=mean(E(1:niis));
EMAX=max(E);
%CALCULAMOS LOS PARAMETROS INTERMEDIOS P1 P2 P3
%En funcion de los cuales vamos a obtener los umbrales
P1=.03*(EMAX-EMS)+EMS; P2=4*EMS; P3=EMAX/3;
UZC=ZCMS+2*DZCS;
UEINF=min(P1,P2);
UESUP=min(UEINF*100,P3);
if UESUP<UEINF, error('EL ALGORITMO HA FALLADO: UESUP<UEINF'); %puede
presentarse por el nivel bajo de la señal.
end
%BUSQUEDA DEL PRINCIPIO DE PALABRA
M=1; %M es el numero de intervalo donde se localizará el principio
while E(M)<UESUP,M=M+1;end
A=M;
if M==1, error('EUSUP se supera muy pronto.');end
M=M-1;
while E(M)>UEINF,M=M-1;end
B=M/2; %Posicion provisional
B=fix(B);
%EN CASO DE FONEMAS FRICATIVOS Y DE PORCA ENERGIA:
niz=11;
```

```
V=5;
while 1
    n=max(1,B-niz);
B1=0; %contador del n° veces que se supera el umbral UZC
for j=B:-1:n
if Z(j)>=UZC %si este intervalo supera UZC continuamos
            B1=B1+1;
Minterm=j; %ademas el principio lo movemos interinamente
end
end
if B1<V %Si en estos niz intervalos no se ha superado/igualado V
vecesUZC, no continuamos.
break;
end
    B=Minterm; %Movemos el principio de palabra.
if n==1,
break; %Se ha llegado al comienzo de la señal
end
end
% BUSQUEDA DEL FIN DE LA PALBRA
nifs=niis; %# de intervalos solapados finales de silencio.
silencio=Z(ni-nifs+1:ni);
ZCMSF=mean(silencio);
DZCSF=std(silencio);
EMSF=mean(E(ni-nifs+1:ni));
EMAXF=max(E);
%Calculamos los parametros intermedios P1 P2 P3
P1F=.03*(EMAXF-EMSF)+EMSF; P2F=4*EMSF; P3F=EMAXF/3;
UZCF=ZCMSF+2*DZCSF;
UEINFF=min(P1F,P2F);
UESUPF=min(UEINFF*100,P3F);
if UESUPF<UEINFF, error('EL ALGORITMO HA FALLADO AL FINAL:
UESUPF<UEINFF');end
MF=ni;
while E(MF)<UESUPF,MF=MF-1;end
AF=MF;
if MF==ni, error('UESUPF se supera muy pronto.');end
MF=MF+1;
while E(MF)>UEINFF,MF=MF+1;end
BF=MF/2;
BF=fix(BF);
%En caso de fonemas fricativos y de poca energia:
while 1
     n=min(ni,BF+niz);
     BF1=0; %contador del numero de veces que se supera el umbral UZCF
for j=BF:n
if Z(j)>=UZCF %si este intervalo supera UZCF
             BF1=BF1+1;
BFinterm=j; %además el final lo movemos interinamente
end
end
if B1<V %si en estos niz intervalos no se ha superado/igualado V veces
UZCF
break;
```

```
end
     BF=BFinterm; %Movemos el final de la palabra
if n==ni
break; %se ha llegado al final de la palabra.
end
end%del bucle while
```

Se han encontrado los puntos donde inicia y termina la señal de voz, ahora se puede almacenar la palabra sola en una nueva variable para continuar con el proceso de análisis.

Anexo B: Código en MATLAB de las pruebas realizadas en el dominio del tiempo.

```
clearall; closeall; clc

% // GRABAMOS Y GUARDAMOS PARA ANALISIS FUTURO.
Fs=8000;
y=wavrecord(2*Fs,Fs,1);
wavwrite(y,Fs,'prueba.wav');
palabra=wavread('mico9.wav');
figure(1); plot(palabra),gridon;

% //// FUNCION Energia.m
[energia, t, delay, energyST]=Energia(palabra);
figure(3); plot(t, palabra*3); title('vocal a'); holdon;
figure(3); plot(t(delay+1:end - delay), energyST, 'r');
xlabel('Time(muestras)'); legend({'Speech','Short-Time Energy'}); hold
off;
cleardelayenergySTt

% //// FUNCION Inicio_Fin.m
z=Inicio_Fin(palabra);
figure(4); plot(z), gridon;

% //// FUNCION Tramas_fft.m
[mayoresamp, f, normaliz, posfre, normalizamp]=Tramas_fft(z, Fs);
figure(5); plot(f,normaliz),grid on; title('FFTs normalizada de
lasventanas'); hold on;
figure(5); plot(posfre,normalizamp,'r'); hold off
clearfnormalizposfrenormalizamp

% //// FUNCION Caract.m
Caracteristica=Caract(energia, mayoresamp);
clearmayoresamp
```

Anexo C: Código en MATLAB del algoritmo final utilizado para el reconocimiento, este algoritmo no incluye la interfaz gráfica.

```
clearall; close all
clc
Fs=8000;
y=wavrecord(2*Fs,Fs,1);
wavwrite(y,Fs,'pruebadell.wav');
palabra=wavread('pruebadell.wav' );
figure(1);
plot(palabra),gridon;
z=palabra;
%Envolvente de energia
winLen=301;
winOverlap=300;
wHamm=hamming(winLen);
%dividiendo y enventanando
sigFramed=buffer(z, winLen, winOverlap, 'nodelay');
sigWindowed = diag(sparse(wHamm)) * sigFramed;
% //// FUNCION STEnergy.m
[energia, t, delay, energyST]=Energia(palabra);
figure(4)
plot(energia);title('ENVOLVENTE ENERGIA')
holdoff;
cleardelayenergySTt
%// Normalizamos "palabra" y "energia"
normalizar=1; %normalizar a este valor 1
maxpa=max(z);
difpa=maxpa-normalizar;
porpa=difpa/maxpa; %porcentaje a RESTARLE o SUMARLE a cada muestra
fortor=1:1:length(z)
normalpa(tor,1)=z(tor,1)-(z(tor,1)*porpa); %palabra de inicio a fin
normalizada
end
%// Normalizamos señal "energia"
difpa=max(energia)-normalizar;
poren=difpa/max(energia); %porcentaje a RESTARLE o SUMARLE a cada muestra
for tore=1:1:length(energia)
normalen(1,tore)=energia(1,tore)-(energia(1,tore)*poren);
%energianormalizada
end
figure(5); plot(normalpa, 'g'); hold on;
figure(5); plot(normalen, 'r');
xlabel('Time (sec)'); legend({'Palabra','Energia en tiempo corto'});
holdoff;
% SILABAS
%SE RECORTA PALABRA ENTRE LOS PUNTOS HALLADOS CON LA ENERGIA
% ahora se busca en donde supera a 0.15 en el vector energia inicio
for d=length(normalen):-1:1
ifnormalen(1,d)>0.15 %////////// modificar
enti=normalen(1,d); %valor donde se supera umbral al inicio de energia.
end
end
% buscmos la pusición de ese número
```

```
forddd=1:length(normalen)
ifenti==normalen(1,ddd)
posen=ddd;
end
end
% ahora se busca en donde supera a 0.108 en el vector energia fin
for d2=1:1:length(normalen)
ifnormalen(1,d2)>0.108 %////////// modificar
enti2=normalen(1,d2); %valor donde se supera umbral al fin de energia.
end
end
% buscmos la pusición de ese número
for ddd2=1:length(normalen)
if enti2==normalen(1,ddd2)
posen2=ddd2;
end
end
energia2=normalen(posen:1:posen2); %guardamos el vector a utilizar
cortesil=min(energia2);
for y=1:length(energia2)
ifcortesil==energia2(1,y)
posminen=y;
end
end
posminpa=posminen+posen;
silab1=normalpa(posen:posminpa);
for d2=1:1:length(normalen)
ifnormalen(1,d2)>0.05 %////////// modificar
        enti2=normalen(1,d2); %valor donde se supera umbral al fin de
energia.
end
end
% buscmos la posición de ese número
for ddd2=1:length(normalen)
if enti2==normalen(1,ddd2)
posibfin=ddd2;
end
end
silab2=normalpa(posminpa+1:posibfin);
% graficamos en una matriz de 2 filas 1 columna
figure(7)
subplot(3,1,1); plot(normalpa);title('PALABRA')
subplot(3,1,2); plot(silab1, 'g');title('SILABA 1')
subplot(3,1,3); plot(silab2, 'g');title('SILABA 2')
% ENVENTANADO Y FFT SILABAS
z=silab1;
times=30; %Tiempo de cada segmento en milisegundos
lenz=length(z);
lenven=((Fs*2)*(times/1000))/2; %# de muestras por segmento
sol=50; %porcentaje solapamiento de segmentos 50%
lensol=(lenven*sol)/100;
vens=(floor(lenz/lenven)*2)-1
v = zeros(lenven,vens);
vseg = zeros(lenven,vens);
```

```
for i=0:(vens-1)
    v(:,i+1)=z((lensol*i)+1:1:(lensol*i)+lenven);
    w = hamming(lenven);
vseg(:,i+1) = w.*v(:,i+1);
% FFT DE LA VENTANA
NFFT = 2^nextpow2(length(vseg));
Y(:,i+1) = fft(vseg(:,i+1),NFFT)/lenven;
    f = Fs/2*linspace(0,1,NFFT/2);
fourier(:,i+1)=2*abs(Y(:,i+1)); %
end
fourier2=fourier(1:128,1:vens);
fourier3=max(fourier2'); %Calcula un vector con todos los picos de la
matriz 'fourier2'
%/// FORMANTE 1 silaba 1
% / Analizamos FFT desde 500Hz
max1=max(fourier3(16:length(fourier3))); %Aca le quitamos las 16 primeras
muestras, equevalente a 500 Hz
% buscmos la pusición de ese número maximo
for x=1:length(fourier3)
if max1==fourier3(1,x)
posib1=x;
end
end
form1=f(posib1); % buscamos a que frecuencia pertenece esa posicion
%/// FORMANTE 2 silaba 1
% eliminamos 15 muestras para hallar el segundo formante "eli"
eli=15;
max2=max(fourier3(posib1+eli:length(fourier3)));
% buscamos la posición de ese número máximo
for w=1:length(fourier3)
if max2==fourier3(1,w)
posib2=w;
end
end
% buscamos a que frecuencia pertenece esa posición
form2=f(posib2);
z=silab2;
times=30; %Tiempo de cada segmento en milisegundos
lenz=length(z);
lenven=((Fs*2)*(times/1000))/2; %# de muestras por segmento
sol=50; %porcentaje solapamiento de segmentos 50%
lensol=(lenven*sol)/100;
vens=(floor(lenz/lenven)*2)-1; %# de segmentos incluyendo los solapados
%Enventanado de la señal (ventana de hamming)
v = zeros(lenven,vens);
vseg = zeros(lenven,vens);
for i=0:(vens-1)
    v(:,i+1)=z((lensol*i)+1:1:(lensol*i)+lenven);
    w = hamming(lenven);
vseg(:,i+1) = w.*v(:,i+1);
% FFT DE LA VENTANA
NFFT = 2^nextpow2(length(vseg));
Y(:,i+1) = fft(vseg(:,i+1),NFFT)/lenven;
    f = Fs/2*linspace(0,1,NFFT/2);
```

```
fourier(:,i+1)=2*abs(Y(:,i+1)); %
end
fourier2=fourier(1:128,1:vens);
fourier4 = max(fourier2');
%/// FORMANTE 1 silaba 2
max3=max(fourier4(10:length(fourier4))); %Aca le quitamos las 9 primeras
muestras.
% buscmos la pusición de ese número maximo
for x=1:length(fourier4)
if max3==fourier4(1,x)
posib3=x;
end
end
% buscamos a que frecuencia pertenece esa posicion
form3=f(posib3);
%/// FORMANTE 2 silaba 2
% eliminamos 15 muestras para hallar el segundo formante "eli"
eli=15;
max4=max(fourier4(posib3+eli:length(fourier4)));
% buscmos la pusición de ese número maximo
for w=1:length(fourier4)
if max4==fourier4(1,w)
posib4=w;
end
end
% buscamos a que frecuencia pertenece esa posicion
form4=f(posib4);
% RECONOCIMIENTO DE PATROES
reconocida=0;
% // palabra 1 FRUTA
if 471<form1 && form1<700
if 1038<form2 && form2<1614
if 471<form3 && form3<819
if 1447<form4 && form4<2701
reconocida=1;
end
end
end
end
% // palabra 2 SEÑAL
if 471<form1 && form1<567
if 1825<form2 && form2<2772
if 628<form3 && form3<855
if 1605<form4 && form4<2000
reconocida=2;
end
end
end
end
% // palabra 3 FLECHA condicion 1
if 471<form1 && form1<567
if 1695<form2 && form2<3150
if 282<form3 && form3<650 || 2706<form3 && form3<3000
if 2676<form4 && form4<4001
```

```
reconocida=3;
end
end
end
end
% // palabra 4 GATO condiciones5
if 471<form1 && form1<830
if 975<form2 && form2<1576
if 250<form3 && form3<536
if 1100<form4 && form4<1544 || 2607<form4 && form4<4001
reconocida=4;
end
end
end
end
% // palabra 5 GRIFO condiciones
if 471<form1 && form1<725 || 1825<form1 && form1<2930
if 1920<form2 && form2<3800
if 282<form3 && form3<505
if 780<form4 && form4<2677
reconocida=5;
end
end
end
end
% // palabra 6 VAGON condiciones
if 620<form1 && form1<830
if 1195<form2 && form2<1481
if 376<form3 && form3<536
if (840<form4) && (form4<1040 || form4==1385 || form4==1417 ||
form4==1448 || form4==1480 || form4==1574)
reconocida=6;
end
end
end
end
% // palabra 7 MESA opciones
if 471<form1 && form1<693 || 2600<form1 && form1<2900
if 2109<form2 && form2<2900 || 3244<form2 && form2<3748
if 525<form3 && form3<920 || 1200<form3 && form3<1600
if 1000<form4 && form4<1606
reconocida=7;
end
end
end
end
% // palabra 8 HOLA
if 471<form1 && form1<600
if 940<form2 && form2<1200
if 345<form3 && form3<820
if 1543<form4 && form4<2750
reconocida=8;
end
end
```

```
end
end
% // palabra 9 ABRIR
if 502<form1 && form1<914
if 1195<form2 && form2<1500
if 282<form3 && form3<473
if 1794<form4 && form4<2900
reconocida=9;
end
end
end
end
```

NOTA: Estos umbrales de la sección de reconocimiento de patrones son para una persona específica, si se desea un buen reconocimiento por parte de otro locutor el algoritmo tendrá que contener los valores de formantes extraídos para dicho locutor, sin embargo es posible que haya reconocimiento por parte de otro locutor sin necesidad de cambiar los umbrales.

Anexo D: Código en MATLAB del algoritmo de entrenamiento de la red neuronal Backpropagation con la palabra "Mesa".

```
clc

entMESA=[535    2677    692 1448];
salMESA=[0 0 1 1];

netMESA=newff(entMESA,salMESA,[10 25],{},'trainbfg'); %la entrada es de,
la salida es dd, tiene 2 capas cada una con 10 y 25 neuronas
respectivamente y el tipo de entrenamiento es trainbfg.
netMESA.outputs{2}.processFcns = {};    % To avoid rescaling of outputs
netMESA.trainParam.showCommandLine = 1; % To show the error on the
commandline
netMESA.divideFcn = '';                 % To avoid division of the data
into validation and test
figure(1); %en la figura 1 se va a dibujar la red ante de ser entrenada
PMESA = sim(netMESA,entMESA); % en P se simula la red antes de ser
entrenada
plot(entMESA,salMESA,entMESA,PMESA,'o') % grafica de la red antes de ser
entrenada
netMESA.trainParam.epochs= 500; % numero máximo de iteraciones
netMESA.trainParam.show= 100; % parámetro del progreso del entrenamiento
netMESA.trainParam.goal= 1e-8; % error al que se quiere llegar

netMESA.trainParam.min_grad=1e-6;
netMESA.trainParam.searchFcn= 'srchcha';


netMESA=train(netMESA,entMESA,salMESA); % entrenamiento de la red
figure(2); % en la figura nueve se va a dibujar la red entrenada
PMESA= sim(netMESA,entMESA); % en P se simula la red entrenada
plot(entMESA,salMESA,entMESA,PMESA,'o')% se dibuja la red entrenada.


errorMESA = salMESA-PMESA; %Error
ERRORMESA = mse(errorMESA);
```

Anexo E: Código en MATLAB de la implementación de redes neuronales en el código básico mostrado en el Anexo C.

```
%% CARGAR VARIABLES REDES ENTRENADAS
loadRED_ENTRENADAS
```

Sección del código del Anexo C.

```
%% CARGAR VARIABLES REDES ENTRENADAS

%SIMULACIONES
P1 =sim(netFRUTA,[form1 form2 form3 form4]);
P2 =sim(netSENAL,[form1 form2 form3 form4]);
P3 =sim(netFLECHA,[form1 form2 form3 form4]);
P4 =sim(netGATO,[form1 form2 form3 form4]);
P5 =sim(netGRIFO,[form1 form2 form3 form4]);
P6 =sim(netVAGON,[form1 form2 form3 form4]);
P7 =sim(netMESA,[form1 form2 form3 form4]);
P8 =sim(netHOLA,[form1 form2 form3 form4]);
P9 =sim(netABRIR,[form1 form2 form3 form4]);

% // REONOCIMIENTO REDES
recoredes=0;
% --- PALABRA 1 --- FRUTA
if 0.75<P1(1,1) && P1(1,1)<1.2
if P1(1,2)<0.75
if P1(1,3)<0.75
if P1(1,4)<0.75
recoredes=1;
end
end
end
end
% --- PALABRA 2 --- SEÑAL
if P2(1,1)<0.75
if 0.75<P2(1,2) && P2(1,2)<1.2
if P2(1,3)<0.75
if P2(1,4)<0.75
recoredes=2;
end
end
end
end
% --- PALABRA 3 --- ABAJO
if P3(1,1)<0.75
if P3(1,2)<0.75
if  0.75<P3(1,3) && P3(1,3)<1.2
if P3(1,4)<0.75
recoredes=3;
end
```

```
end
end
end
% --- PALABRA 4 --- GATO
if P4(1,1)<0.75
if P4(1,2)<0.75
if P4(1,3)<0.75
if 0.75<P4(1,4) && P4(1,4)<1.2
recoredes=4;
end
end
end
end
% --- PALABRA 5 --- GRIFO
if 0.75<P5(1,4) && P5(1,4)<1.2
if 0.75<P5(1,4) && P5(1,4)<1.2
if P5(1,3)<0.75
if P5(1,3)<0.75
recoredes=5;
end
end
end
end
% --- PALABRA 6 --- VAGON
if P6(1,3)<0.75
if 0.75<P6(1,4) && P6(1,4)<1.2
if 0.75<P6(1,4) && P6(1,4)<1.2
if P6(1,3)<0.75
recoredes=6;
end
end
end
end
% --- PALABRA 7 --- MESA
if P7(1,3)<0.75
if P7(1,3)<0.75
if 0.75<P7(1,4) && P7(1,4)<1.2
if 0.75<P7(1,4) && P7(1,4)<1.2
recoredes=7;
end
end
end
end
% --- PALABRA 8 --- HOLA
if 0.75<P8(1,4) && P8(1,4)<1.2
if 0.75<P8(1,4) && P8(1,4)<1.2
if 0.75<P8(1,4) && P8(1,4)<1.2
if P8(1,3)<0.75
recoredes=8;
end
end
end
end
% --- PALABRA 9 --- ABRIR
```

```
if P9(1,3)<0.75
if 0.75<P9(1,4) && P9(1,4)<1.2
if 0.75<P9(1,4) && P9(1,4)<1.2
if 0.75<P9(1,4) && P9(1,4)<1.2
recoredes=9;
end
end
end
end
```

Anexo F: Código en MATLAB, de la interfaz gráfica de usuario. En este anexo se omiten algunas partes de código que se encuentran en otros anexos, sin embarque se dejaran enunciados estos espacios a los que pertenece otro código. También se han eliminado la mayoría de comentarios para ahorrar espacio, partes de código que no son necesarias por el algoritmo, solo son utilizados como una guía para el programador.

```
functionvarargout = Interfaz(varargin)
% INTERFAZ M-file for Interfaz.fig
%      INTERFAZ, by itself, creates a new INTERFAZ or raises the existing
%      singleton*.
% Begin initialization code - DO NOT EDIT
% End initialization code - DO NOT EDIT

% --- Executes just before Interfaz is made visible.
functionInterfaz_OpeningFcn(hObject, eventdata, handles, varargin)
%IMAGENES EN LOS AXES:
axes(handles.axes1)
background = imread('Fruta.jpg');
axisoff;
imshow(background);
%*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
axes(handles.axes2)
background = imread('Señal.jpg');
axisoff;
imshow(background);
%*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
axes(handles.axes3)
background = imread('Flecha.jpg');
axisoff;
imshow(background);
%*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
axes(handles.axes4)
background = imread('Gato.jpg');
axisoff;
imshow(background);
%*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
axes(handles.axes5)
background = imread('Grifo.jpg');
axisoff;
imshow(background);
%*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
axes(handles.axes6)
background = imread('Vagon.jpg');
axisoff;
imshow(background);
%*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
axes(handles.axes7)
background = imread('Mesa.jpg');
axisoff;
imshow(background);
%*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
axes(handles.axes8)
background = imread('Hola.jpg');
axisoff;
```

```
imshow(background);
%*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
axes(handles.axes9)
background = imread('Abrir.jpg');
axisoff;
imshow(background);
%*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
% INICIAMOS VARIABLE
global p
    p=0;
% Choose default command line output for Interfaz
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.
functionvarargout = Interfaz_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in Grabar.
functionGrabar_Callback(hObject, eventdata, handles)
% hObject    handle to Grabar (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

En este espacio va el código enunciado en el ANEXO C con algunas
modificaciones menores, como agregar los datos de formantes para cada uno
de los locutores y mensajes de error.

% --- Executes on button press in Acerca.
functionAcerca_Callback(hObject, eventdata, handles)
% hObject    handle to Acerca (see GCBO)

% --------------------------------------------------------------------
functionLocutor_Callback(hObject, eventdata, handles)
% hObject    handle to Locutor (see GCBO)

% --------------------------------------------------------------------
function Untitled_1_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% PARA EL LOCUTOR LORENA
global p
set(handles.text10,'String','Lorena');     %poner texto en text10
p=1;
msgbox({'Persona seleccionada','Lorena'},'Locutor');
```

```
% --------------------------------------------------------------------
function Untitled_2_Callback(hObject, eventdata, handles)
% hObject    handle to Untitled_2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% PARA EL LOCUTOR JULIAN
global p
set(handles.text10,'String','Julian');     %ponertexto en text10
p=2;
msgbox({'Persona seleccionada','Julian'},'Locutor');

% --------------------------------------------------------------------
functionColor_de_Fondo_Callback(hObject, eventdata, handles)
% hObject    handle to Color_de_Fondo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --------------------------------------------------------------------
functionPor_Defecto_Callback(hObject, eventdata, handles)
% hObject    handle to Por_Defecto (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

set(handles.Fondo,'Backgroundcolor','factory')
set(handles.text1,'Backgroundcolor','factory')
set(handles.text2,'Backgroundcolor','factory')
set(handles.text3,'Backgroundcolor','factory')
set(handles.text4,'Backgroundcolor','factory')
set(handles.text5,'Backgroundcolor','factory')
set(handles.text6,'Backgroundcolor','factory')
set(handles.text7,'Backgroundcolor','factory')
set(handles.text8,'Backgroundcolor','factory')
set(handles.text9,'Backgroundcolor','factory')
set(handles.text10,'Backgroundcolor','factory')
set(handles.text11,'Backgroundcolor','factory')

% --------------------------------------------------------------------
functionFondo_Amarillo_Callback(hObject, eventdata, handles)
% hObject    handle to Fondo_Amarillo (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

set(handles.Fondo,'Backgroundcolor','y')
set(handles.text1,'Backgroundcolor','y')
set(handles.text2,'Backgroundcolor','y')
set(handles.text3,'Backgroundcolor','y')
set(handles.text4,'Backgroundcolor','y')
set(handles.text5,'Backgroundcolor','y')
set(handles.text6,'Backgroundcolor','y')
set(handles.text7,'Backgroundcolor','y')
set(handles.text8,'Backgroundcolor','y')
set(handles.text9,'Backgroundcolor','y')
```

```
set(handles.text10,'Backgroundcolor','y')
set(handles.text11,'Backgroundcolor','y')

% --------------------------------------------------------------------
functionFondo_Violeta_Callback(hObject, eventdata, handles)
% hObject    handle to Fondo_Violeta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

set(handles.Fondo,'Backgroundcolor','m')
set(handles.text1,'Backgroundcolor','m')
set(handles.text2,'Backgroundcolor','m')
set(handles.text3,'Backgroundcolor','m')
set(handles.text4,'Backgroundcolor','m')
set(handles.text5,'Backgroundcolor','m')
set(handles.text6,'Backgroundcolor','m')
set(handles.text7,'Backgroundcolor','m')
set(handles.text8,'Backgroundcolor','m')
set(handles.text9,'Backgroundcolor','m')
set(handles.text10,'Backgroundcolor','m')
set(handles.text11,'Backgroundcolor','m')
```