Resumen

En este Trabajo Especial de Grado se plantea realizar los cambios necesarios al sistema de reconocimiento de voz que controla a la calculadora estándar VOXAG desarrollada en el TEG titulado "*Reconocimiento del habla usando el algoritmo de coeficientes cepstrales en escala Mel para controlar una calculadora estándar bajo la plataforma PXI*" por el Ing. Alberto Rodríguez, de forma tal que se aumente su eficiencia y se corrijan los problemas relacionados con el género del locutor. Este sistema utiliza el algoritmo de Coeficientes Cepstrales en Escala Mel (MFCC) y la plataforma PXI para implementar el procesamiento. Con base en la aplicación previamente desarrollada, se busca implementar la mejora principal del sistema: aumentar el tamaño del *codeword* de 8 a 16 centroides. Para esto es necesaria la revisión, el cambio y la actualización de varios de los *subVIs* presentes dentro de la aplicación. Una vez realizado el nuevo entrenamiento del sistema y preparados los *Codebooks* necesarios, se llevan a cabo las pruebas para evaluar cuantitativamente la eficiencia, las cuales no alcanzan los resultados esperados, por lo que se determina que la longitud temporal de los comandos afecta significativamente esta nueva versión de la aplicación, y se concluye que son necesarios *codebooks* con comandos adaptados a la cantidad de centroides requeridos. Se observa también que el cambio a 16 centroides le permite al sistema tener un poco más de fortaleza ante el ruido.

Palabras Clave: Análisis Cepstral, reconocimiento de voz, estudio de centroides, patrones de voz.

Universidad Católica Andrés Bello
Facultad de Ingeniería
Escuela Ingeniería de Telecomunicaciones
Trabajo Especial de Grado

Mejora del sistema de control de una calculadora estándar usando la plataforma PXI y el algoritmo de Coeficientes Cepstrales en Escala Mel.

TRABAJO ESPECIAL DE GRADO

Presentado ante la

UNIVERSIDAD CATÓLICA ANDRÉS BELLO

Como parte de los requisitos para optar al título de

INGENIERO EN TELECOMUNICACIONES

Gabriel Guzmán

Tutor: Prof. Maria Cristi Stefanelli

Caracas, septiembre de 2012

Dedicatoria

Dedico este trabajo a las cuatro personas que más importancia han tenido en mi vida: mis padres, Meylin Rosario y Jorge Guzmán; mi hermano, Alejandro Guzmán; y mi tío, Carlos Eduardo Aponte.

Agradecimientos

Quiero agradecer a todos los que me ayudaron en la realización de este trabajo: a los profesores Javier Barrios, Trina Adrian y Maria Cristi Stefanelli, por su invaluable ayuda; a los técnicos encargados de los laboratorios de la escuela de Ingeniería en Telecomunicaciones, por soportar mi horario; a mis compañeros Marlyn Zapata, Freddy Vásquez, Francisco Contreras y Guillermo Hernández, por su ayuda y apoyo; y a todos los voluntarios que participaron en el trabajo de investigación, al prestar su tiempo y voces para grabar las muestras.

Finalmente, quiero agradecer especialmente a Juliet Cáceres por el gran apoyo que me dio desde el primer día. Sin su ayuda, la culminación de este trabajo especial de grado no hubiera sido posible.

Índice General

Índice de Figuras

Índice de Tablas

Introducción

A lo largo de los años se ha visto como con el avance de la tecnología y con tiempos de procesamiento cada vez más cortos, el desarrollo de aplicaciones que ayuden al ser humano en su día a día ha tomado cada vez más importancia. En este sentido, las aplicaciones del área de las telecomunicaciones son de las que han adquirido más relevancia en las últimas décadas, y son el fundamento de este Trabajo Especial de Grado el cual, basado en las investigaciones y conclusiones previas del ingeniero Alberto Rodríguez, busca realizar una "Mejora del sistema de control de una calculadora estándar usando la plataforma PXI y el algoritmo de Coeficientes Cepstrales en Escala Mel". Dicho trabajo se estructurará de la siguiente forma:

En el primer capítulo se plantea el problema a estudiar, las limitaciones y alcances del trabajo y los objetivos a alcanzar para lograr los resultados deseados.

En el capítulo dos se expone toda la base teórica necesaria para el desarrollo de la investigación en cuanto a: preprocesamiento de señales, adquisición de parámetros de la voz, normalización de los mismos y recursos utilizados.

En el tercer capítulo se explica la metodología seguida durante la realización del trabajo, incluyendo: la investigación teórica para comprender el funcionamiento del sistema, la implementación de la calculadora a controlar con las mejoras necesarias, el entrenamiento del sistema, la evaluación del sistema en condiciones ideales y en condiciones de ruido, y por último la actualización del manual de usuario en caso de ser necesario.

El cuarto capítulo presenta el resultado de cada una de las fases desarrolladas durante el proyecto: el diseño obtenido después de las mejoras realizadas, las etapas por las que pasa la señal durante su procesamiento, los

resultados de entrenar al sistema, el comportamiento del sistema bajo las distintas condiciones y el manual final.

En el quinto capítulo se presentan una serie de conclusiones producto de la investigación, así como recomendaciones que se podrían tomar en cuenta para profundizar aún más en la investigación.

Por último, se añaden los anexos y apéndices que contienen información complementaria creada y utilizada en el curso del desarrollo de este trabajo.

CAPÍTULO I

Planteamiento del Proyecto

I.1. Planteamiento del Problema

Estudiando el desarrollo y evolución de la tecnología a través de los años, se observa como la interacción entre el usuario y la máquina se ha simplificado bastante; no sólo porque ahora es más sencillo para las personas entender y trabajar las distintas aplicaciones gracias al mejoramiento de las interfaces, sino que a la par de esta mejora, la capacidad y la velocidad de procesamiento de las computadoras han aumentado impresionantemente. Aprovechando esto, los programadores y desarrolladores siempre están en la búsqueda de nuevas formas o mejoras que permitan comunicarse con el usuario de una forma natural y fluida, bien sea mediante teclados convencionales, pantallas táctiles, sensores de movimiento o reconocimiento de voz.

Este último caso es el que interesa en este Trabajo Especial de Grado, ya que se está trabajando con un sistema de reconocimiento automático del habla. Esto no es sencillo, ya que el computador debe tomar señales distintas a las que maneja naturalmente y traducirlas adecuadamente a su lenguaje de máquina para poder realizar la operación deseada. Para lograr dicha traducción, se utilizan muchos de los recursos del computador, y esto puede ralentizar la toma de decisiones y el tiempo de respuesta. Es por esto que al momento de crear las aplicaciones se debería buscar un equilibrio adecuado entre el tiempo de toma de decisiones y las prestaciones, de forma tal que se logre presentar resultados lo más confiables posible en tiempo real (Rodríguez, 2010).

En el curso de este trabajo, se debe disponer entonces de dos cosas fundamentales: métodos de procesamiento adecuados para las señales de audio (que pueden estar basados en predicción lineal, Transformada de Fourier,

coeficientes cepstrales, etc.), implementados en el *software* de simulación *NI LabVIEW*; y un *hardware* eficiente y con la capacidad necesaria para el trabajo.

En el caso de esta investigación, se trabaja con el equipo PXI (*PCI eXtention for Instrumentation*) el cual es una plataforma basada en PC desarrollada por *National Instruments* que ofrece una solución de despliegue de alto rendimiento y bajo costo para sistemas de medida y automatización. (National Instruments, s.f.)

En este Trabajo Especial de Grado se plantea el problema de realizar las mejoras necesarias al sistema de reconocimiento de voz que controla a la calculadora estándar VOXAG desarrollada por el ingeniero Alberto Rodríguez, de forma tal que se aumente su eficiencia y se corrijan los problemas relacionados con el género del locutor. Este sistema utiliza el algoritmo de Coeficientes Cepstrales en Escala Mel (MFCC) y la plataforma PXI para implementar el procesamiento.

I.2. Objetivos

I.2.1 Objetivo General

Mejorar el sistema de reconocimiento del habla en tiempo real usado para controlar una aplicación de calculadora estándar por medio del algoritmo de Coeficientes Cepstrales en Escala Mel (MFCC) y la plataforma PXI.

I.2.2 Objetivos Específicos

- Estudiar la plataforma desarrollada en el TEG ***Reconocimiento del habla usando el algoritmo de coeficientes cepstrales en escala Mel para controlar una calculadora estándar bajo la plataforma PXI*.**
- Seleccionar el diccionario de palabras a utilizar de forma que las formantes de las palabras no se solapen para mejorar el comportamiento del sistema.

- Buscar un equilibrio entre el desempeño del sistema y el tiempo de procesamiento necesario para dar respuesta a una búsqueda.
- Aumentar el tamaño de *Codeword*.
- Ampliar la muestra de entrenamiento para género masculino y femenino.
- Implementar las mejoras en la calculadora en el lenguaje de programación *LabVIEW*.
- Discriminar qué palabras del diccionario anteriormente definido es capaz de reconocer el algoritmo y cuáles no.
- Estudiar la influencia del ruido en el sistema.
- Actualizar el manual de usuario.

I.3. Limitaciones y Alcance

Durante el curso de la investigación se realizaron una serie de cambios al sistema desarrollado anteriormente, con el objetivo de mejorar su eficiencia general y corregir ciertos problemas presentes. Dichos cambios se hicieron tomando en cuenta las recomendaciones del trabajo previo, así como las nuevas consideraciones que fueron surgiendo a lo largo de la investigación. Al final se hace un análisis de que tanto mejoró el sistema, y si el camino seguido es ventajoso o deben buscarse otras alternativas.

Para la implementación del sistema RAH se utiliza una técnica de extracción de características de la voz mediante los Coeficientes Cepstrales en Escala Mel (MFCC), así como un método de clasificación por Distancia Euclidiana a partir de la Cuantificación Vectorial (VQ) de las características de la voz. En cuanto a *hardware*, se utiliza la plataforma modular especializada PXI de *National Instruments* y su módulo NI PXI-5142, disponibles en la Universidad Católica Andrés Bello, y que permiten realizar procesamiento de señales de audio en tiempo real con el *software* de programación *LabVIEW*.

Las pruebas de ruido se llevaron a cabo mediante la caracterización y simulación del mismo a través del *software* de programación *LabVIEW*,

permitiendo variar su intensidad y hacer pruebas inmediatas. Las condiciones de ruido ambiental fueron las presentes en el laboratorio de microondas de la UCAB.

Dado que la aplicación tiene un único comando asignado a cada una de las teclas de la misma, es necesario el uso de un diccionario limitado en el entrenamiento del sistema.

I.4. Justificación

Esta investigación busca mejorar un producto que fue concebido y diseñado para facilitar la comunicación hombre-máquina en el manejo de una aplicación de calculadora estándar a través de comandos de voz, la cual a pesar de funcionar correctamente bajo ciertas condiciones, debería alcanzar una eficiencia mayor antes de hacerla llegar a los usuarios finales. Al lograr mejorarla, esta herramienta podría ser de mayor utilidad para personas discapacitadas que requieran de una ayuda extra para cumplir con sus tareas, o simplemente para personas que busquen alternativas más cómodas para llevar a cabo una tarea común como lo es realizar cálculos relativamente simples.

CAPÍTULO II

Marco Teórico

Los sistemas de reconocimiento automático del habla (a partir de ahora, RAH) no son más que los distintos algoritmos seguidos para realizar un análisis detallado de las características de los parámetros del habla. Dichos sistemas dependerán del lenguaje empleado, así como de la frecuencia, energía y potencia de la señal acústica a procesar, entre otras cosas (San Martín & Carrillo, 2004). Estos algoritmos pueden dividirse en tres etapas fundamentales:

- El muestreo de la señal de voz, fase en la cual se obtiene dicha señal cuantificada en muestras al tomarla e introducirla en el computador utilizando un micrófono, el cual cambia la energía acústica en impulsos eléctricos.
- El preprocesamiento de las muestras obtenidas, adaptándolas al algoritmo haciéndolas pasar por distintos procesos como: el preénfasis de la señal, la escogencia del inicio y fin de la muestra útil, la segmentación y la aplicación de una ventana temporal, con lo cual se obtiene una señal lista para ser comparada e identificada (Rodríguez, 2010).
- El reconocimiento y clasificación de las características más importantes de las muestras, lo cual se logra estimando la envolvente espectral de la señal ya que su evolución en el tiempo caracteriza las distintas realizaciones acústicas (Méndez, Hernando, Nadeu, & Vallverdú, s.f.).

Todo este proceso puede verse, de forma generalizada, como el esquema presentado en la Figura 1 (Huang, Acero, & Hon, 2001).

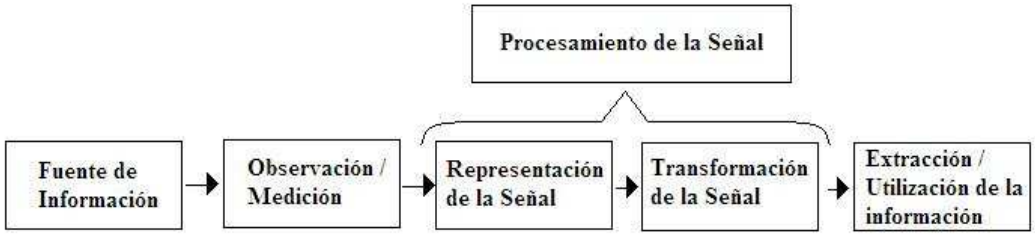

Figura 1 Extracción de las Características de una señal

Fuente: (Huang, Acero, & Hon, 2001)

Una vez realizada la parametrización de las muestras de la señal de entrada, ya se dispone de los datos y características necesarias para poder realizar comparaciones con una base de datos de muestras previamente parametrizadas (*Codebook).*

II.1. Representación de la Señal de Voz

Al estudiar como modelar la voz para llevar a cabo la parametrización, surge el planteamiento de un sistema compuesto por un filtro lineal y variante en el tiempo h[n], a través del cual pasa una fuente o señal de excitación e[n], resultando en la señal de voz x[n], como puede apreciarse en la Figura 2 (Huang, Acero, & Hon, 2001).

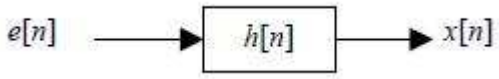

Figura 2 Modelado de la Señal de Voz

Fuente: (Huang, Acero, & Hon, 2001)

La señal de excitación e[n] sería equivalente al aire que atraviesa las cuerdas vocales, mientras que el filtro h[n] representaría el tracto vocal, todo esto como parte del aparato de producción vocal del ser humano, el cual se observa en la Figura 3 (Huang, Acero, & Hon, 2001).

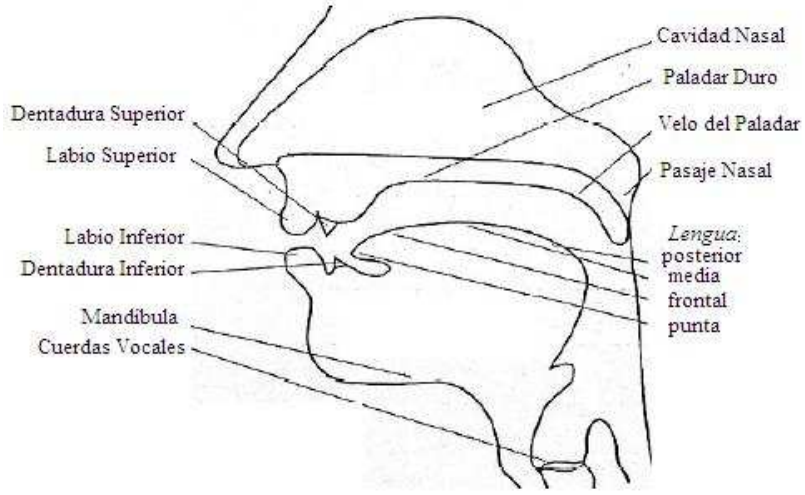

Figura 3 Diagrama del Aparato de Producción Vocal Humano

Fuente: (Huang, Acero, & Hon, 2001)

La mayoría de la información que identifica al locutor se encuentra en la señal e[n], mientras que la información relacionada a lo que este está diciendo se localiza en el filtro. Es por esto que la definición del banco de filtros tiene una gran relevancia en este trabajo (Rodríguez, 2010).

Una vez conocido el modelo, se estudiará con más detalle las fases del sistema RAH mencionadas anteriormente:

II.2. Etapa de Muestreo

El primer paso es capturar la señal de voz, mediante un micrófono que logra convertir la señal acústica en una señal eléctrica, para poder manipularla. Al pasar por el micrófono queda una señal analógica, por lo que se procede a digitalizarla con la ayuda de un conversor analógico-digital, muestreándola cada cierto Ts, o tiempo de muestreo, como se observa en la Figura 4. Con esto se obtiene una secuencia de valores discretos, recordando siempre que a la hora de muestrear se debe cumplir con el teorema de Nyquist (Goretti & González, s.f.).

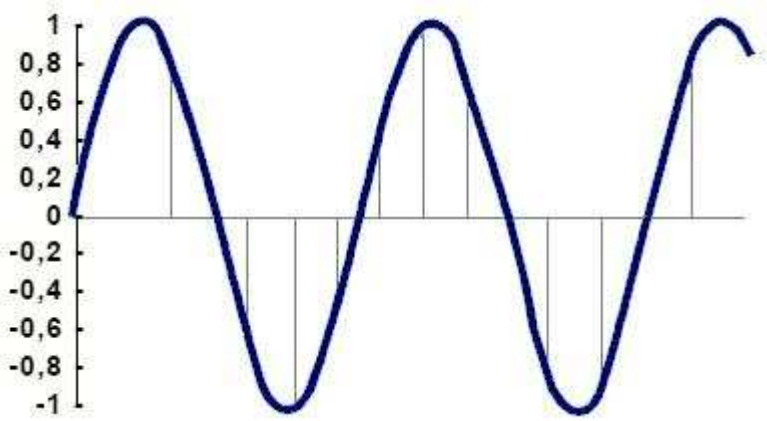

Figura 4 Señal Analógica Muestreada cada Ts

Fuente: (Goretti & González, s.f.)

II.3. Etapa de Preprocesamiento

Teniendo la señal muestreada, se prosigue con la fase de preprocesamiento donde se busca mejorar la calidad y características de la señal para obtener mejores resultados más adelante.

II.3.1. Filtro de Preénfasis

Se utiliza un filtro digital FIR (*Finite Impulse Response*) con un solo coeficiente denominado filtro de preénfasis. Está diseñado para mejorar las características de la señal, alzando su espectro aproximadamente 20 dB por década, ya que los segmentos de voz sonoros tienen una pendiente espectral negativa que se tiende a contrarrestar con este filtro. Por otro lado, la audición es más sensible en frecuencias por encima de 1 Khz, y este filtro amplifica esta zona del espectro de la señal con lo que se mejora la apreciación de sus características en las siguientes etapas (Goretti & González, s.f.).

Gracias al efecto del filtro, se observa un escalamiento gráfico del espectro en frecuencia, y su aplicación viene dada por la ecuación:

$$x[n] = e[n] + ae[n-1] \quad (1)$$

donde e[n] es la señal de entrada, x[n] es la salida del filtro de preénfasis y *a* es el coeficiente único del filtro; para los casos en que $a < 0$ se obtiene un filtro pasa bajo; en cambio, si $a > 1$ se obtiene un filtro pasa alto. Normalmente se utiliza un valor de a = 0.95, con el que se obtiene la respuesta en frecuencia aproximada de la Figura 5 (Rodríguez, 2010).

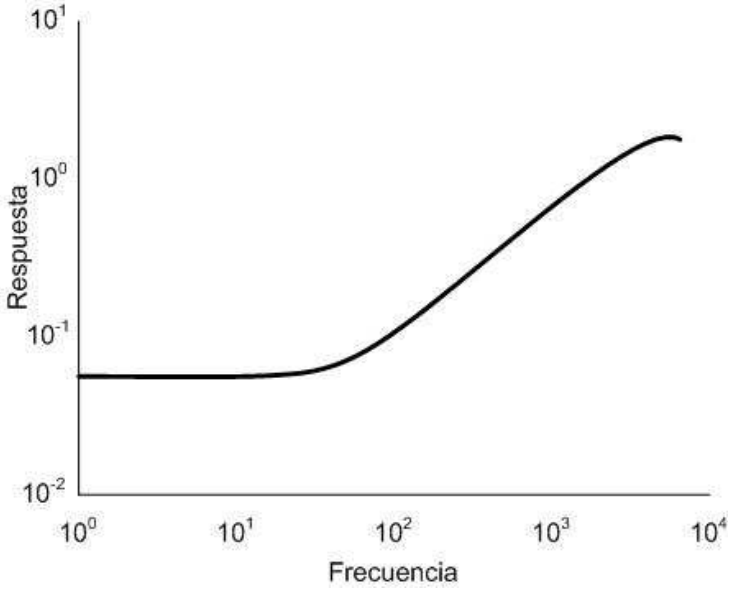

Figura 5 Respuesta en Frecuencia del Filtro de Preénfasis

Fuente: (Rodríguez, 2010)

II.3.2. Detección de *Endpoint*

Llegado a este punto del preprocesamiento, es momento de comenzar a aislar las palabras para poder procesarlas adecuadamente más adelante. Se presenta entonces el problema de determinar el inicio y fin de lo que es la muestra útil, que si no se delimita, muy difícilmente podrá ser comparada correctamente después con las muestras del *Codebook*. Por está razón, es necesario el proceso de detección de *Endpoint* (bordes o puntos finales). Esté método propone determinar ciertos umbrales a partir de la energía de la señal dada por la ecuación:

$$g[n] = 10\log\sum x^2[n] \quad \forall \quad n \qquad (2)$$

donde x[n] es la señal a procesar. Se proponen entonces cuatro umbrales cuyo fin es definir la presencia de un pulso de energía, como se ve en la Figura 6, ya que las palabras habladas contienen pulsos de energía y por lo tanto se hace necesario determinarlos para ver cual pertenece a cada palabra.

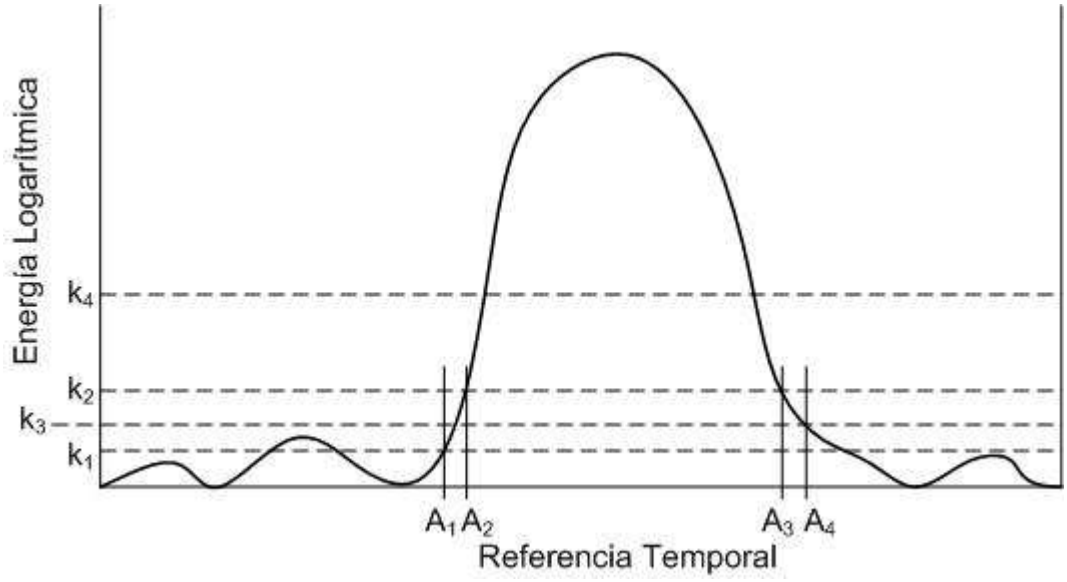

Figura 6 Umbrales de energía para la detección de Endpoint.

Fuente: (Rodríguez, 2010)

En la práctica, se escanean los valores de g[n] de izquierda a derecha, y cuando el valor de g[n] en algún punto excede el umbral k_1, se etiqueta ese punto como A_1. Si g[n] excede el umbral k_2 antes de caer por debajo de k_1 entonces A_1 será el inicio de la muestra. De igual manera se determina el final del pulso de energía utilizando los umbrales k_2 y k_3.

Otras pruebas que se hacen en este algoritmo para que el pulso de energía sea aprobado son: que el pico del pulso supere el umbral k_4 y que la duración de dicho pulso sea mayor a 75ms (Carranza, s.f.).

II.3.3. Segmentación

Para reducir la complejidad o extensión de los cálculos y operaciones que añadirían tiempo de procesamiento a la aplicación final es aconsejable realizar una

segmentación de la señal. Esta puede aplicarse teóricamente con una cantidad N de puntos cualquiera, pero para aumentar la eficiencia del algoritmo se utiliza comúnmente el N radix 2. Dichos segmentos deben tener una cantidad de puntos radix 2 y tener un intervalo de duración entre 20ms y 40ms, para que puedan ser considerados cuasi estacionarios (San Martín & Carrillo, 2004).

II.3.4. Aplicación de Ventana

Uno de los problemas que puede surgir a la hora de trabajar con la señal de voz es el fenómeno de Gibbs, el cual se basa en los valores que toma la serie de Fourier para los puntos de una señal que representan una discontinuidad (Rodríguez del Río & Zuazua, 2003). Esto se reduce con la aplicación de una ventana a cada segmento, lo cual permite suavizar la onda y evitar variaciones muy rápidas de la misma. Existen variedad de ventanas como: Rectangular, Von Hann, Hamming, Blackman, Kaiser y Dolph-Chebyshev, cada una con aplicaciones específicas (Antoniou, 2006). La ventana de Hamming es la más usada en el análisis de señales de voz pues permite una mejor resolución espectral (San Martín & Carrillo, 2004).

Dicha ventana se define con la función:

$$w(n) = \begin{cases} 0.54 + 0.46\cos\left(\frac{2\pi n}{N}\right) & para\ 0 \le |n| < N \\ 0 & para\ otros\ casos \end{cases} \quad (3)$$

II.4. Etapa de Procesamiento Cepstral

Si bien se han desarrollado diversos métodos para obtener los parámetros de las señales de voz, como los basados en LPC (*Linear Predictive Coding*), en la Transformada de Fourier o combinaciones híbridas de estas técnicas, este trabajo de investigación se enfoca en el método de los coeficientes cepstrales,

específicamente los Coeficientes Ceptrales en Escala Mel (o *Mel-frequency cepstral coefficients*, MFCC) (Rodríguez, 2010)

Primero, se debe buscar la forma de realizar la separación de la fuente y el filtro transformándolos en una suma, lo que se conoce como deconvolución homomórfica (Huang, Acero, & Hon, 2001). Según lo que se planteó con la Figura 2, la salida x[n] puede expresarse como la siguiente convolución:

$$x[n] = e[n] * h[n] \tag{4}$$

Se debe separar entonces la señal de excitación del sistema, llevando la expresión anterior (**4**) a:

$$\hat{x}[n] = \hat{e}[n] + \hat{h}[n] \tag{5}$$

Para lograr llegar a (**5**), se aplica la Transformada de Fourier a la convolución (**4)**, quedando expresada en el dominio de la frecuencia como:

$$X(f) = E(f) \cdot H(f) \tag{6}$$

Aplicando las propiedades del logaritmo, se puede transformar la multiplicación en una suma:

$$log\,[X(f)] = log\,[E(f)] + log\,[H(f)] \tag{7}$$

Por propiedades de la Transformada de Fourier, se sabe que la transformada de una suma es la suma de las transformadas, por lo que si se aplica de nuevo la transformada, se obtiene:

$$F[\log[X(f)]] = F[\log[E(f)]] + F[\log[H(f)]] \tag{8}$$

Finalmente, el cepstrum se definirá como:

$$C(q) = F[log\,[F[x(n)]]] \tag{9}$$

Esta representación tiene unidades de tiempo, y para identificar que es resultado de una transformación cepstral, a la unidad se le da el nombre de *quefrency*, y se le asigna la letra D al operador cepstrum (Rodríguez, 2010).

Ya conocido esto, se puede utilizar un filtrado homomórfico para separar la señal tanto en la fuente como en el sistema, aplicando el operador D a la señal de voz x[n], filtrando con una ventana l[n] y pasando por la operación inversa D^{-1},

recuperando así h[n] o e[n], dependiendo de la ventana l[n] utilizada. Este sistema de filtrado se observa en la Figura 7.

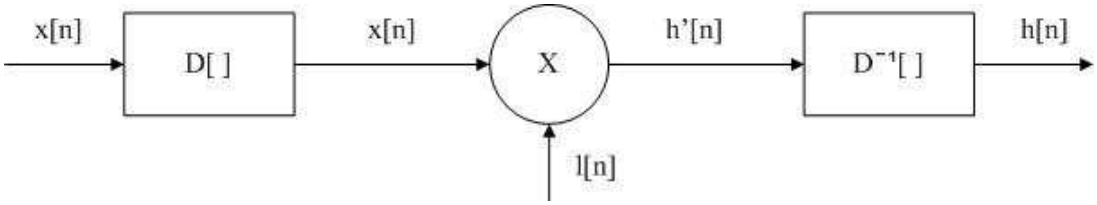

Figura 7 Sistema de Filtrado Homomórfico

Fuente: (Huang, Acero, & Hon, 2001)

De esta manera, se puede encontrar un valor de N tal que $\hat{h}[n] \approx 0$ para $n \geq N$ y que $\hat{e}[n] \approx 0$ para $n < N$, lo que permitirá recuperar y extraer cualquiera de las dos según se desee (Huang, Acero, & Hon, 2001):

Para $l[n] = \begin{cases} 1 & |n| < N \\ 0 & |n| \geq N \end{cases}$ se obtiene h[n].

Para $l[n] = \begin{cases} 1 & |n| \geq N \\ 0 & |n| < N \end{cases}$ se obtiene e[n].

Tomando en cuenta el operador cepstral es que se llega a la técnica de extracción de parámetros de señales de voz, conocida como:

II.4.1. Coeficientes Cepstrales en Escala Mel (MFCC)

Con esta técnica, a diferencia de los métodos cepstrales comunes, se utiliza una escala de frecuencia no lineal que logra modelar más eficazmente al sistema auditivo humano. Esta escala tiene como unidad el mel, y se establece que como referencia 1000 mels sean equivalentes a la frecuencia fundamental percibida (o *pitch*) de un tono a 1KHz (Rodríguez, 2010).

En la Figura 8 se observa como los 1000 mels de referencia corresponden a los 1000 Hz, tanto para una escala de frecuencia lineal como para una logaritmica.

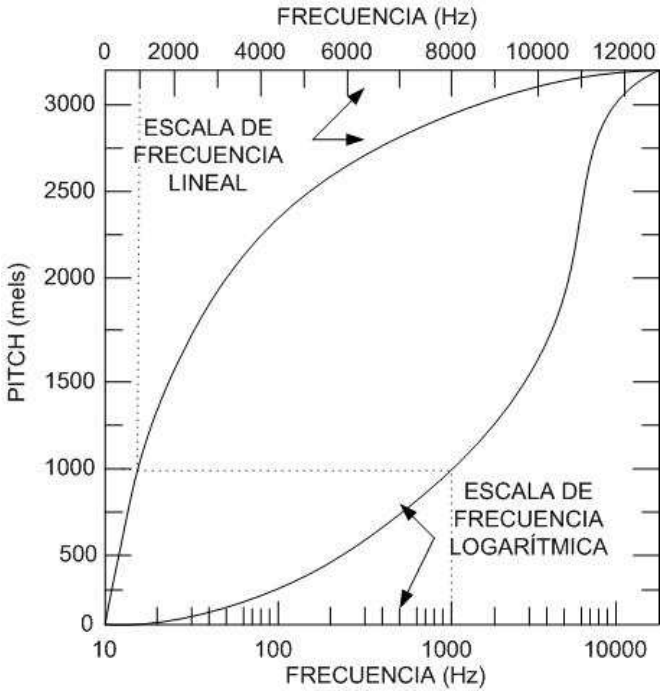

Figura 8 *Pitch* percibido en mels.

Fuente: (Rabiner & Juang, 1993)

Para comenzar con esta técnica, obtenemos la Transformada Discreta de Fourier (DFT) de la señal de entrada, lo que es igual a:

$$X_a[k] = \sum_{n=0}^{n-1} x[n] e^{-j2\pi nk/N}, \qquad 0 \leq k < N \tag{10}$$

Después de esto se define un banco de filtros con M filtros:

m=1,2,3,…,M

donde el m es el número del filtro triangular que viene dado por:

$$H_m[k] \begin{cases} 0 & k < f[m-1] \\ \dfrac{2(k - f[m-1])}{(f[m+1] - f[m-1])(f(m) - f[m-1])} & f[m-1] \leq k \leq f[m] \\ \dfrac{2(f[m+1] - k)}{(f[m+1] - f[m-1])(f(m+1) - f[m])} & f[m] \leq k \leq f[m+1] \\ 0 & k > f[m+1] \end{cases} \tag{11}$$

Dichos M filtros se muestran en la Figura 9, y se puede notar que estos calculan el espectro alrededor de cada frecuencia central, con anchos de banda que van incrementando.

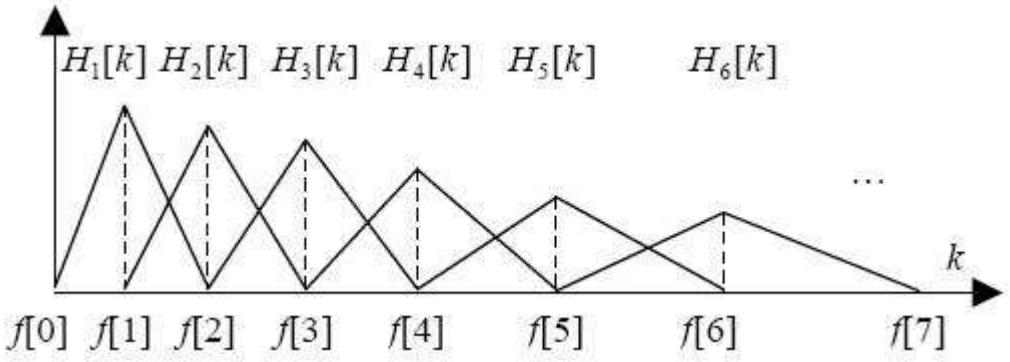

Figura 9 Filtros triangulares usados en el cálculo de Mel Cepstrum

Fuente: (Huang, Acero, & Hon, 2001)

Son diseñados con una separación asociada a la escala de frecuencia Mel, que viene dada por:

$$M = 2595 lg\left(1 + \frac{f}{700}\right) \tag{12}$$

para intentar simular el tracto vocal y dar énfasis a las frecuencias críticas que aproximan mejor el modelo (Huang, Acero, & Hon, 2001).

Una vez definido el banco de filtros, se calcula el logaritmo de la energía a la salida de cada uno de ellos, mediante:

$$S[m] = ln\left[\sum_{k=0}^{N-1} |X_a[k]|^2 H_m[k]\right], \qquad 0 \leq m \leq M \tag{13}$$

Donde N es el tamaño de la DFT de la ecuacion 10. Por último, los coeficientes cepstrales en escala Mel serán el resultado de aplicar la Transformada Coseno Discreta a cada una de estas salidas:

$$c[n] = \sum_{m=0}^{M-1} S[m] \cos\left(\frac{\pi n\left(m + \frac{1}{2}\right)}{M}\right) \qquad 0 \leq n < M \tag{14}$$

El resultado de todo esto es un conjunto de coeficientes con los que se construye una matriz que engloba los valores de todos los segmentos, la cual no es más que una representación más comprimida de la señal, y de la cual se pueden

obtener suficientes características de esta. Estas características se representan como puntos en un plano al que se conoce como vector acústico, y al agruparlas se pueden observar y comparar con otras muestras para determinar su similitud (Rodríguez, 2010).

II.5. Cuantificación Vectorial

Una vez que se tiene las características de la señal, llega el momento de reconocer cual es la palabra recibida. Si bien existen diversos métodos para lograr esto, una de las soluciones más efectivas es medir la distancia entre dos patrones o vectores de muestras (Rabiner & Juang, 1993).

Una de las formas de lograr esto es mediante la medición de la distancia o distorsión euclidiana entre las muestras de la señal entrante y las muestras almacenadas en el sistema RAH, lo que no es más que una simple resta entre ellas. Para lograr esto, se deben de normalizar primero las muestras que van a compararse ya que los vectores acústicos obtenidos siempre tendrán distintas dimensiones, producto de distintos factores como la velocidad y entonación del locutor, el tamaño de la palabra, el ruido del ambiente, etc.

Como se dijo anteriormente, el vector acústico no es más que una representación de puntos que ocupan un espacio en *n* dimensiones. Lo que hace la cuantificación vectorial es dividirlo en espacios más pequeños y le calcula el centroide de cada uno, formando con ellos lo que se denomina *codebook* de una palabra, el cual contiene sus características más relevantes en una forma más compresa (Rodríguez, 2010).

Para lograr esto se sigue lo que se conoce como el Algoritmo Generalizado de Lloyd o el Algoritmo de Agrupamiento de K-promedios en cual se realizan los siguientes pasos (Rabiner & Juang, 1993):

1. Inicialización: Se realiza una elección arbitraria de un punto en el espacio que es el centroide del *codebook*.

2. Se duplica el tamaño del *codebook* y_n dividiéndolo con la siguiente regla:

$$y_n^+ = y_n(1+\varepsilon)$$
$$y_n^- = y_n(1-\varepsilon) \quad (15)$$

donde *n* varía desde 1 hasta el tamaño del *codebook*, y ε es un parámetro de división típicamente entre 0.01 y 0.05.

3. Para cada vector se busca el vecino más cercano, es decir, el punto más cercano en términos de distancia y se asigna al grupo.
4. El vecino encontrado anteriormente es asignado como el nuevo centroide del grupo.
5. Se repiten los pasos 3 y 4 hasta que la distancia promedio esté por debajo de un umbral designado.
6. Se repiten los pasos 2, 3 y 4 hasta que el *codebook* alcance el tamaño deseado.

En la Figura 10 se presenta el diagrama de flujo de dicho algoritmo:

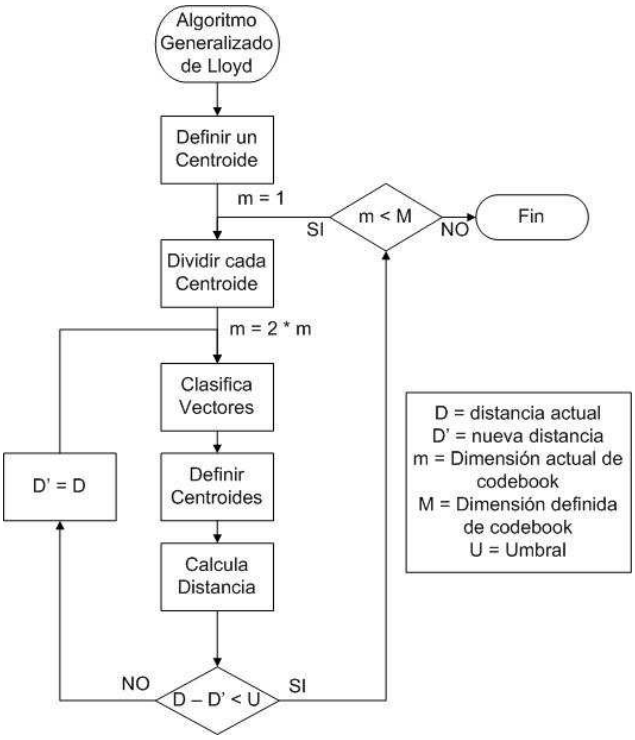

Figura 10 Algoritmo Generalizado de Lloyd

Fuente: (Rabiner & Juang, 1993)

Entre las ventajas que nos aporta este método, aparte de obtener matrices con las mismas dimensiones que permiten hacer una comparación efectiva, tenemos que se realiza la compresión de la información espectral de cada muestra, así como una reducción sustancial del cómputo necesario para calcular la distorsión y una representación discreta del discurso asociando "etiquetas" a cada *codeword* (Rodríguez, 2010).

La principal desventaja que presenta es que la cantidad de memoria necesaria para almacenar el *codebook* de todas las palabras usadas puede llegar a ser muy amplia; a medida que el *codebook* se haga más grande para reducir el error de cuantificación, aumentará su espacio en memoria. Es por esta razón que debe buscarse un equilibrio adecuado entre la eficiencia del sistema y el tiempo que tarda en dar una respuesta, tomando en cuenta los recursos de procesamiento con los que se cuente (Rabiner & Juang, 1993).

II.6. Distancia Euclidiana Generalizada

Por último, se hace necesario realizar la comparación entre la muestra entrante y las distintas muestras almacenadas en el diccionario o *codebook*, para encontrar la que concuerde y realizar el reconocimiento de la instrucción dictada. Para esto se calcula la Distancia Euclidiana Generalizada con cada uno de los *codewords*, hallando la distorsión entre ambas matrices. Esta distorsión, que es un valor numérico, indica que tan distintas son dos muestras, por lo que aquella que arroje el menor valor será tomada por el sistema como la comparación correcta y por ende, la palabra buscada (Rodríguez, 2010).

Esta distorsión o distancia entre muestras se halla mediante la formula de la Distancia Euclidiana Generalizada (Weisstein, s.f.) (Rabiner & Juang, 1993), la cual viene dada por:

$$d = |x - y| = \sqrt{\sum_{i=1}^{n} |x_i - y_i|^2} \qquad (16)$$

II.7. Plataforma PXI de *National Instruments*

PCI eXtensions for Instrumentation **(**PXI) es una plataforma desarrollada por *National Instruments* con un alto rendimiento y desempeño en un sinnúmero de aplicaciones, ya que por parte del *hardware* presenta una instrumentación modular de ranuras PCI y PCI *Express* y por parte del *software* es fácil de utilizar con programas conocidos, ya que está basada en PC; razones por las cuales tiene una gran versatilidad a la hora de utilizarla para algún tipo de proyecto (National Instruments, s.f.).

Su arquitectura, como puede verse en la Figura 11, está formada por tres componentes básicos: el chasis, el controlador del sistema y los módulos periféricos.

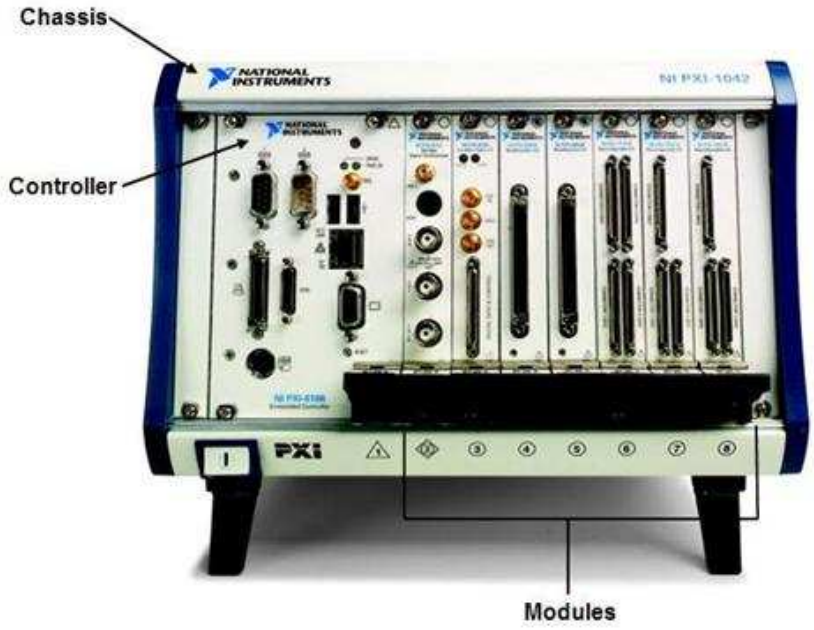

Figura 11 Arquitectura Básica del PXI de National Instruments
Fuente: (National Instruments, s.f.)

El chasis es la montura o empaque sobre la cual se mantiene todo el sistema y puede tener diversos tamaños, dependiendo de su capacidad de expansión, o mejor dicho, de cuantas tarjetas se le pueden añadir. Dicho tamaño

se mide en unidades de “U”, las cuales indican el número de tarjetas que es capaz de soportar. En la Figura 12 pueden observarse algunos ejemplos de chasis PXI.

Figura 12 Chasis del PXI de National Instruments

Fuente: (National Instruments, s.f.)

El controlador del sistema puede ser una PC común, una laptop conectada al PXI o un *hardware* especial que pueda ser acoplado al chasis, como el que se muestra en la Figura , el cual posee los componentes básicos de un computador común: CPU, disco duro, memoria RAM, Tarjeta Ethernet, controlador de video, puerto serial, USB entre otros; y con un sistema operativo, bien sea *Microsoft Windows XP/Vista* o *LabVIEW Real-Time.*

Figura 13 Controlador NI PXI-8106

Fuente: (National Instruments, s.f.)

Los módulos periféricos no son más que las tarjetas de expansión especializadas que pueden añadirse al chasis. Por ser PXI una industria de estándar abierto existe una variedad de más de 1500 módulos periféricos,

fabricados y suministrados por más de 70 proveedores, denotando la amplitud de opciones y versatilidad que presenta el sistema para la aplicación de productos.

El *software* para PXI basado en *Windows* es el mismo que el de una PC basado en *Windows*; por ello no es necesario reescribir programas especialmente para PXI, y es posible utilizar aplicaciones de desarrollo tales como *NI LabVIEW*, *LabWindows™/CVI* y *Measurement Studio*, y *Visual Basic* y *Visual C/C++*. Finalmente, la implementación de la Arquitectura de Software para Instrumentos Virtuales (VISA), la cual ha sido extensamente adoptada en el campo de la instrumentación, se requiere por PXI para la configuración y control de instrumentos VXI, GPIB, seriales y PXI (Rodríguez, 2010.).

En la Universidad Católica Andrés Bello se dispone de un equipo PXI para su uso y para la experimentación. Éste equipo tiene un chasis con una capacidad de 8 *slots*, y el controlador del sistema es el NI PXI-8106, diseñado para la adquisición de datos, y que cuenta con un procesador Intel Core 2 Duo T7400 de 2.16GHz y con memoria doble canal DDR2 de 667MHz, así como un disco duro de 60GB, un controlador Ethernet, puerto serial, puerto paralelo, puerto USB, *ExpressCard*, además de contar con el sistema operativo *Windows XP*. Los módulos disponibles son el NI PXI-5610, el cual funciona como *up_converter* para el traslado de frecuencias en sistemas de comunicaciones; el NI PXI-5441, un generador de forma de onda arbitraria con procesamiento de señal; el NI PXI-5600, un *down_coverter*; y el NI PXI-5142, que se muestra en la Figura 14, el cual es un digitalizador de 14 bits para el procesamiento de señales.

Figura 14 Módulo NI PXI 5142

Fuente: (National Instruments, s.f.)

Este último módulo es el que permite la adquisición de datos bajo altas prestaciones. Posee 2 canales de entrada con conectores BNC muestreando simultáneamente de 100MS/s a 2GM/s en tiempo real con 14 bits de resolución. Tiene 100MHz de ancho de banda, así como filtros de ruido y antisolapamiento y permite decimar con protección de solapamiento para todas las muestras, entre las cualidades útiles para otras aplicaciones (National Instruments, s.f.)(Rodríguez, 2010).

CAPÍTULO III

Metodología

A lo largo de este capítulo se describen las distintas etapas que caracterizan la metodología seguida durante la investigación y su desarrollo como trabajo especial de grado.

III.1 Revisión y familiarización con el sistema de RAH

Para la primera fase del desarrollo de este trabajo especial de grado se lleva a cabo una revisión exhaustiva de la investigación realizada previamente por el bachiller Alberto Rodríguez, titulada "*Reconocimiento del habla usando el algoritmo de coeficientes cepstrales en escala Mel para controlar una calculadora estándar bajo la plataforma PXI*". De igual manera, se busca la bibliografía en que se basó dicha investigación, de forma tal de analizar todas las fuentes disponibles.

Tomando en cuenta lo anterior, se adquieren los conocimientos para entender el diseño del sistema, el cual se resume a lo mostrado en la Figura .

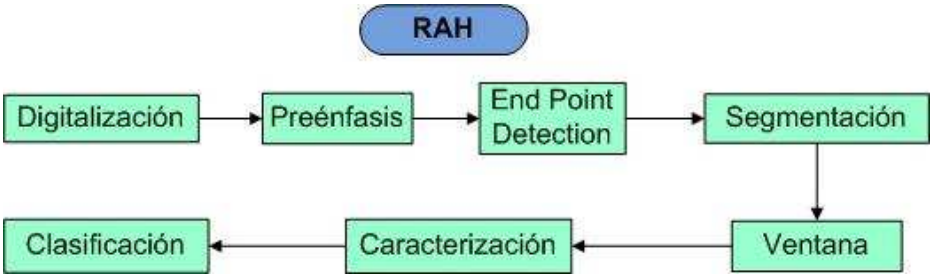

Figura 15 Fases del RAH
Fuente: (Rodríguez, 2010)

En relación con dicho esquema, y tal como ha sido expuesto anteriormente, el sistema de reconocimiento del habla busca simular las características de la voz modelándola como una señal de excitación y al tracto vocal como un sistema por el que pasa dicha señal, siendo este de vital

importancia al momento de reconocer lo que se ha dicho mas no al sujeto que lo dijo.

Para lograr implementar el sistema de RAH, deben seguirse los pasos mostrados en la Figura 15. El primer paso es digitalizar la señal de voz mediante el uso de un transductor común, en nuestro caso, un micrófono conectado a la plataforma PXI de *National Instruments*, equipo que provee ciertas ventajas en rendimiento y velocidad de procesamiento de señales.

Una vez digitalizada la señal con una frecuencia de muestreo de 10KHz, se le aplica un proceso de preénfasis con el fin de resaltar las características de la voz a alta frecuencia, logrando una amplificación de la señal y una mejor apreciación del contenido, necesaria para un procesamiento adecuado en los siguientes pasos.

Llegado a este punto, es necesario determinar que parte de la digitalización resulta ser o no información útil como producto de la vocalización de una palabra. Gracias al módulo de digitalización NI-5142 es posible iniciar la digitalización de la señal solo cuando se reciba sonido por encima de un cierto nivel de histéresis, mientras que con el algoritmo *End Point Detection* es posible observar la diferencia en función de la energía que produce un sonido en un instante dado y, mediante el uso de umbrales y etiquetas, eliminar el contenido innecesario después de la señal útil.

El próximo paso es llevar a cabo la segmentación de la señal, lo cual se hace siempre, por muy corta que sea la palabra. Con la frecuencia de muestreo utilizada y pensando en la FFT que debe aplicarse más adelante, se segmenta la señal en partes de 256 muestras, obteniendo una duración de 25.6ms por segmento, lo cual es adecuado ya que cada uno debe tener entre 20 y 40 milisegundos de duración.

A cada segmento obtenido en el paso anterior se le aplica ahora una ventana, la cual tiene como función suavizar el comportamiento de la señal, evitando así los cambios bruscos en la misma. La ventana utilizada es la de Hamming, pues es la más común para este tipo de aplicación por su buena resolución en el espectro de frecuencia para señales de voz.

Seguidamente, se realiza la extracción de características de la señal, lo cual se realiza mediante el método de Coeficientes Cepstrales en Escala Mel (MFCC), el cual se aplica a cada uno de los segmentos que componen la señal, obteniendo una matriz de coeficientes que representa las características de la voz. Solo con esta matriz no podemos realizar una comparación adecuada entre un *Codebook* (conjunto de señales de referencia con las que se entrena el sistema) y la señal entrante, por lo que surge la necesidad de llevar a cabo una Cuantificación Vectorial (VQ).

La Cuantificación Vectorial se aplica a la matriz de coeficientes que se obtienen del algoritmo MFCC, buscando obtener un grupo de centroides en el espacio de la matriz de coeficientes. Dicho grupo representa las características de una señal y es llamado *Codeword*.

Por último, se ejecuta la clasificación de la señal identificando finalmente la palabra dicha por el locutor, utilizando la distancia euclidiana entre el *Codeword* obtenido y todos los *Codewords* que conforman el *Codebook* en el mismo espacio de la matriz de coeficientes. Aquella distancia que sea la menor entre todas, será la que determina el comando de voz más adecuado a seleccionar.

III.2 Comprobación de los resultados previos del sistema desarrollado

En su trabajo de grado, el bachiller Alberto Rodríguez llegó a ciertas conclusiones después de realizar varias pruebas al sistema de RAH desarrollado, las cuales arrojaron resultados expresados en tablas, llamadas Matrices de

Confusión. En ellas se representan los aciertos y errores al estudiar cada una de las palabras del diccionario utilizado, donde las filas representan la palabra estudiada, mientras que las columnas indican cuantas veces el reconocimiento identificó determinada palabra entre 20 vocalizaciones realizadas para cada una.

Antes de implementar las mejoras al sistema, se decide realizar tres de las cuatro pruebas llevadas a cabo en la investigación anterior, de forma tal de comprobar la validez de los resultados obtenidos y hasta que punto podían llegar a variar. La primera prueba se realiza con un sujeto masculino y con un *Codebook* creado con 10 sujetos masculinos y 10 femeninos. La segunda se lleva a cabo con un sujeto masculino, comparando con un *Codebook* compuesto únicamente por 20 muestras de sujetos masculinos. Por último, se hace una prueba con un sujeto femenino y un codebook de 20 sujetos masculinos y 10 femeninos.

III.3 Diseño e implementación de mejoras del sistema de RAH

Con base en la aplicación previamente desarrollada, se busca implementar la mejora principal del sistema: aumentar el tamaño del *codeword* de 8 a 16 centroides. Para esto es necesaria la revisión, el cambio y la actualización de varios de los *subVIs* presentes dentro del sistema, como lo son:

II.3.1. act_cent_svi

Este el *subVI* donde se lleva a cabo la actualización y ubicación de los centroides, por lo que es de vital importancia dentro de la aplicación. Es aquí donde se realiza el cambio de la ampliación a 16 centroides, añadiendo 8 nuevas salidas al case principal, como se muestra en la Figura 16.

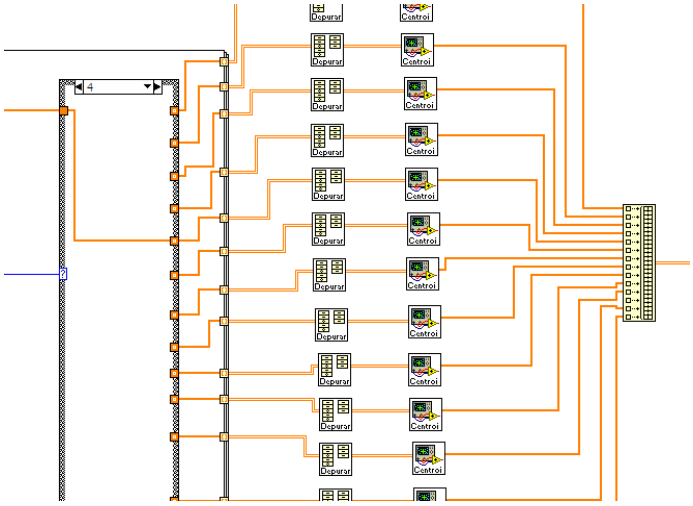

Figura 16 Aumento a 16 centroides en act_cent_svi
Fuente: Elaboración Propia

Esto amerita el cambio y actualización en otros *subVIs* que están relacionados con él. Se guarda con el nuevo nombre act_cent_svi_V3.

II.3.2. x2_cent_svi

Se encarga de duplicar el grupo de centroides, por lo que depende directamente del *subVI* act_cent_svi_V3 y es necesario actualizar el llamado del mismo, como se observa en la Figura 17.

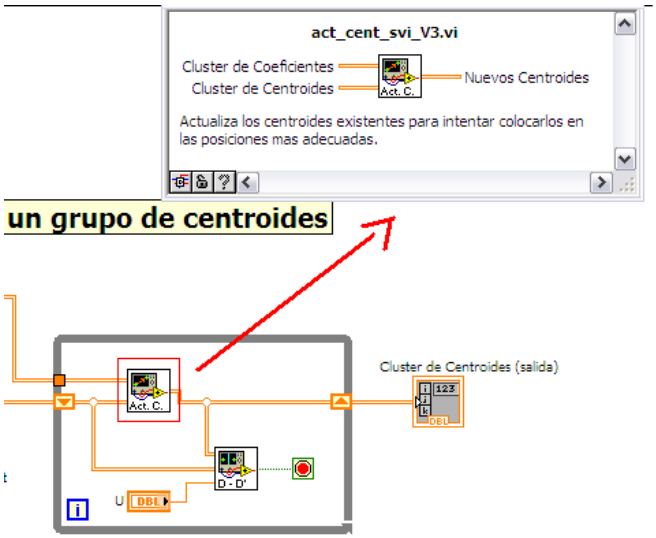

Figura 17 Actualización de llamada al *subVI* act_cent_svi_V3 en x2_cent_svi
Fuente: Elaboración Propia

Se guarda con el nuevo nombre x2_cent_svi_v3.

II.3.3. VQ8_svi

En él se agregó otro *subVI* x2_cent_svi_v3 para adaptar el proceso de cuantificación vectorial a los nuevos 16 centroides, y se actualizaron las llamadas anteriores a dicho subVi. Dicho cambio se presenta en la Figura 18.

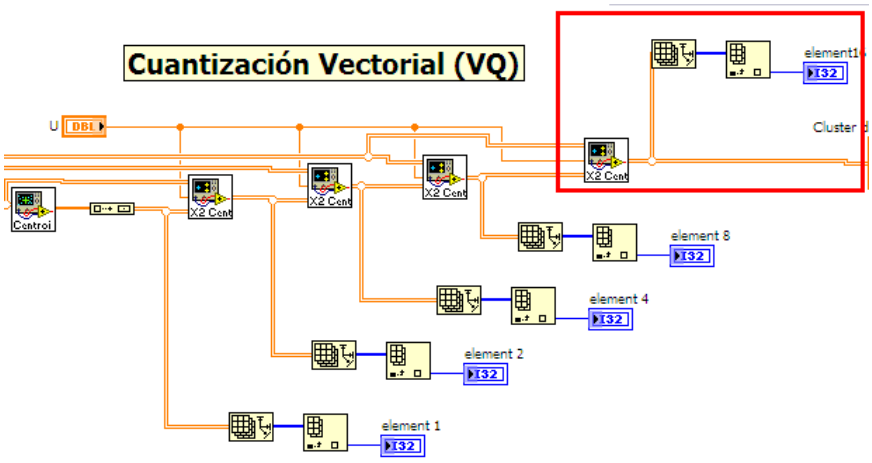

Figura 18 Ampliación y actualización del VQ8_svi
Fuente: Elaboración Propia

Se guarda con el nuevo nombre VQ16_svi, para representar el aumento del número de centroides, y se actualizan las llamadas que se hacen a este *subVI* desde el parametrizador.

II.3.4. Grabador

Se modifica el *VI* de grabación para entrenar al sistema, ya que el proceso se llevaba a cabo antes de aplicar la cuantificación vectorial, por lo que había que realizar un procesamiento aparte a las muestras que se guardaban. Para solucionarlo, se coloca el grabador después del VQ16, como se muestra en la Figura 19.

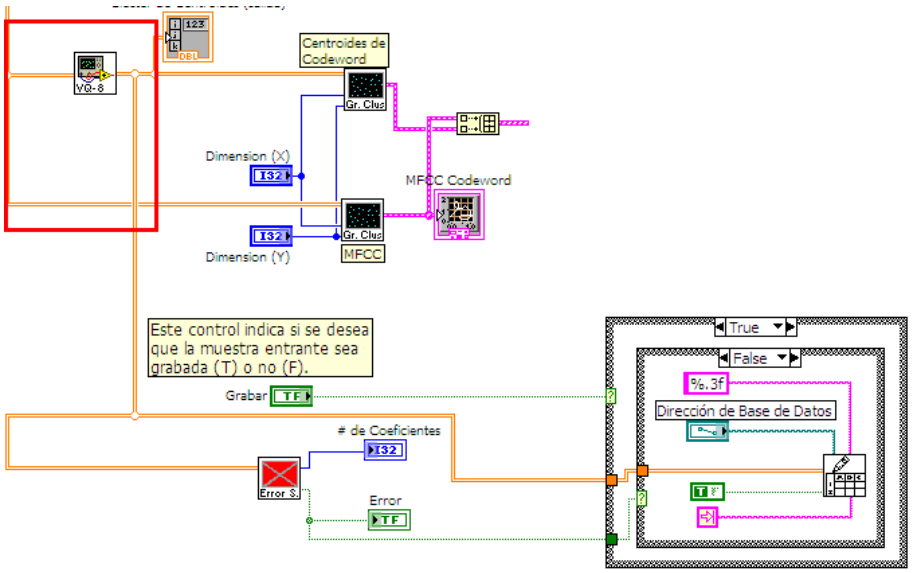

Figura 19 Modificación de *VI* Grabador
Fuente: Elaboración Propia

Se guarda con el nuevo nombre de Grabador_v3, y se actualiza la llamada al VQ16_svi.

II.3.5. Mini Grabador

Se agrega un mini grabador al programa principal para facilitar el entrenamiento y la comprobación del reconocimiento de la voz, el cual se observa en las Figuras 20 y 21.

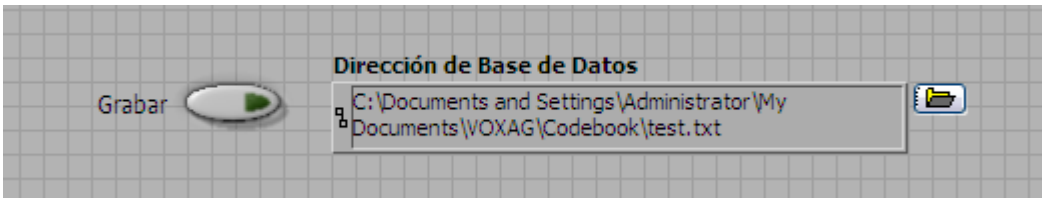

Figura 20 Controles del grabador pequeño
Fuente: Elaboración Propia

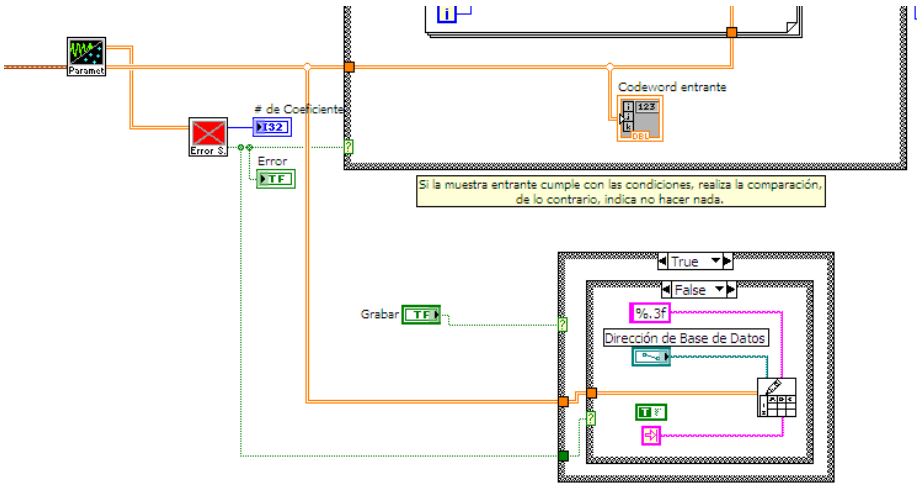

Figura 21 Modificación del *VI* principal: Mini Grabador añadido
Fuente: Elaboración Propia

II.3.6. Cambios al *VI* principal

El *VI* principal recibe una serie de cambios para adaptarlo, entre los que se cuentan:

a) Se cambia a 16 el denominador de la división que nos indica cuantas veces se repetirá la comparación de la palabra recibida con aquellas presentes en la base de datos, dicho cambio se muestra en la Figura 22.

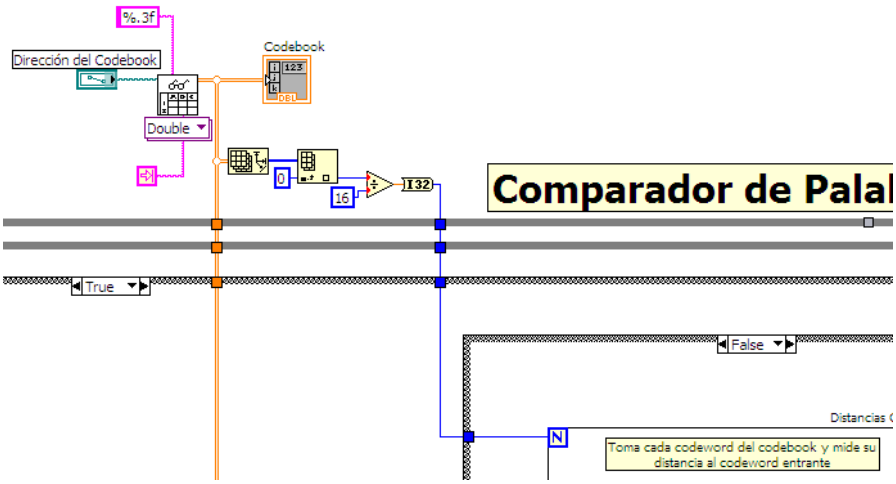

Figura 22 Modificación del *VI* principal: Cambio a ciclo de comparación
Fuente: Elaboración Propia

b) Se adaptan los *case* donde se discrimina cual es la palabra que se ha dicho después de la comparación, de forma tal que sean compatibles con la nueva longitud del *codebook*, como se ve en la Figura 23.

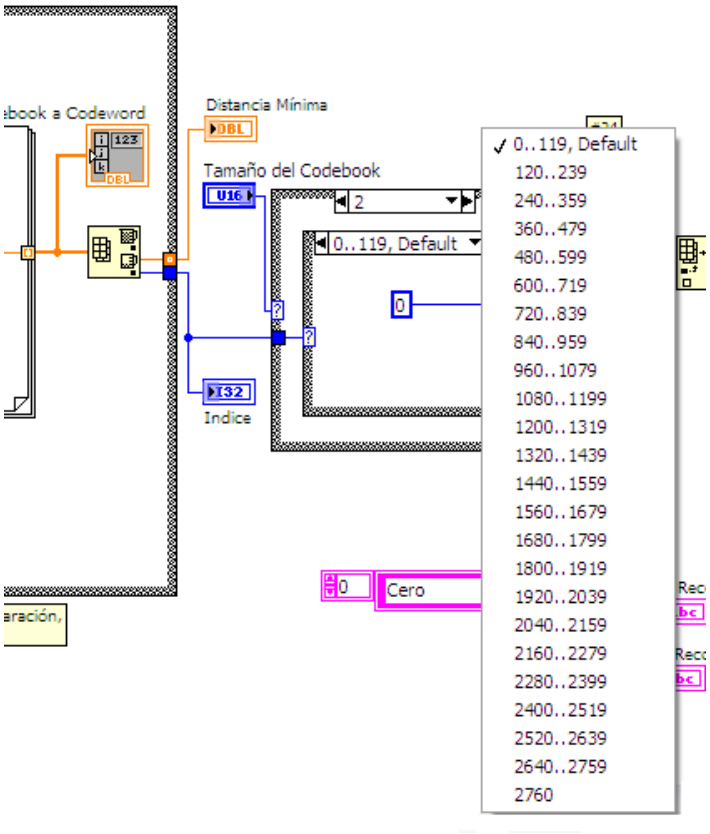

Figura 23 Modificación del *VI* principal: *Case* de selección de palabras
Fuente: Elaboración Propia

c) Se modifica el selector de elección de tamaño del *codebook*, para adaptarlo a la nueva y más amplia cantidad de sujetos utilizados para el entrenamiento. Se decide trabajar con diccionarios de 30, 40 o 60 sujetos, como se muestra en la Figura 24.

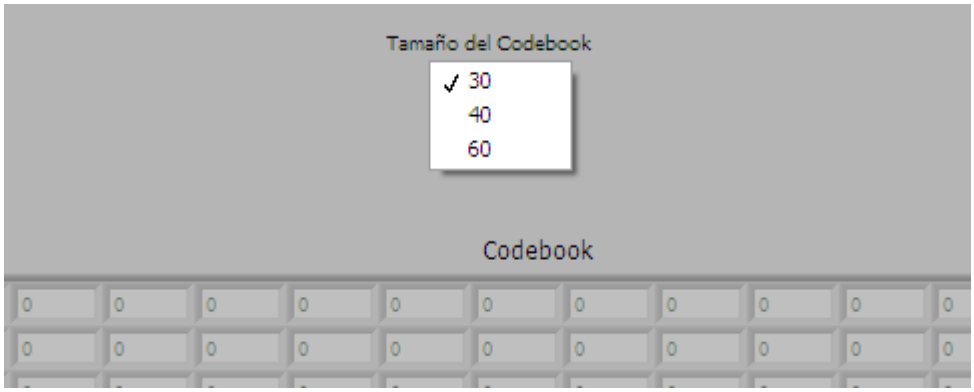

Figura 24 Modificación del *VI* principal: Selector de tamaño de *codebook*
Fuente: Elaboración Propia

III.4 Nuevo entrenamiento del sistema de RAH

Una vez implementados los cambios al sistema de RAH, se define el conjunto de comandos a utilizar. Para el caso de esta investigación se decide trabajar con dos diccionarios: el usado originalmente en la investigación anterior, y otro basado en el original pero sustituyendo cinco de las palabras, de forma tal de intentar mejorar la comparación de los *codewords*. Habiendo definido los diccionarios, se procede a realizar el nuevo entrenamiento del sistema para formar los *Codebooks* correspondientes.

Para lograr esto se aplica todo el algoritmo estudiado hasta la caracterización de la voz a las muestras de cada una de las palabras a utilizar, y el resultado es almacenado de forma permanente en una base de datos en formato de texto.

Siguiendo las recomendaciones y el objetivo planteado, se aumenta la muestra de sujetos al entrenar el sistema, de forma tal de tener una flexibilidad más amplia al momento de crear los distintos *Codebooks*. Bajo esta premisa, se decide grabar las voces de 60 sujetos, 30 masculinos y 30 femeninos, en edad adulta y contemplando todo el rango de variación de la frecuencia fundamental de la voz en adultos encontrada en la bibliografía (Anexo A).

El entrenamiento del sistema se basa en que el sujeto vocalice dos veces cada uno de los comandos a utilizar frente al sistema de RAH, siguiendo el "Formato de Control Para el Entrenamiento del Sistema de RAH" mostrado en el Apéndice A, y sus muestras sean procesadas y almacenadas en la base de datos formando el *Codebook*.

III.5 Evaluación del sistema de RAH

Una vez realizado el nuevo entrenamiento del sistema y preparados los *Codebooks* necesarios, se llevan a cabo las pruebas para evaluar cuantitativamente la eficiencia de la aplicación.

El método de evaluación es la observación. Se realizan 20 vocalizaciones de cada palabra una por una frente al sistema con un solo individuo, y se cuenta la cantidad de aciertos y fallos obtenidos para llenar una matriz de confusión (San Martín & Carrillo, 2004) y tener la capacidad de presentar resultados de forma porcentual completando así la primera evaluación del sistema.

Primero se realiza una prueba con un sujeto masculino y un *codebook* de 30 muestras masculinas, seguido de otra similar pero con sujeto y muestras femeninos. Debido a que los resultados de estas evaluaciones no arrojaron los resultados esperados, se hace otra serie de pruebas para comprobar el funcionamiento de la aplicación, así como la verdadera eficacia del cambio a 16 centroides.

Entre dichas pruebas se encuentra la vocalización ante el sistema con *codebooks* cortos, formados específicamente con la intención de comprobar el efecto del tamaño de las palabras al momento del reconocimiento, así como la influencia de las vocales que las forman.

III.6 Estudio del sistema de RAH frente al ruido

Un aspecto que no se estudió en el ámbito de la investigación anterior fue el efecto que podía llegar a tener el ruido sobre el sistema implementado. Es por esto que surge como propuesta para el trabajo actual investigar dichos efectos e implicaciones.

Para lograr esto, se simula el canal de ruido dentro de la aplicación en el *software* de programación *LabVIEW* mediante las herramientas del ruido blanco gaussiano (AWGN), y se multiplica por el nivel de ruido que decida el usuario. Posteriormente, dicho ruido se suma a la señal proveniente del micrófono, y se prosigue con todo el procesamiento y manejo de la información alterada, como puede verse en la Figura 25.

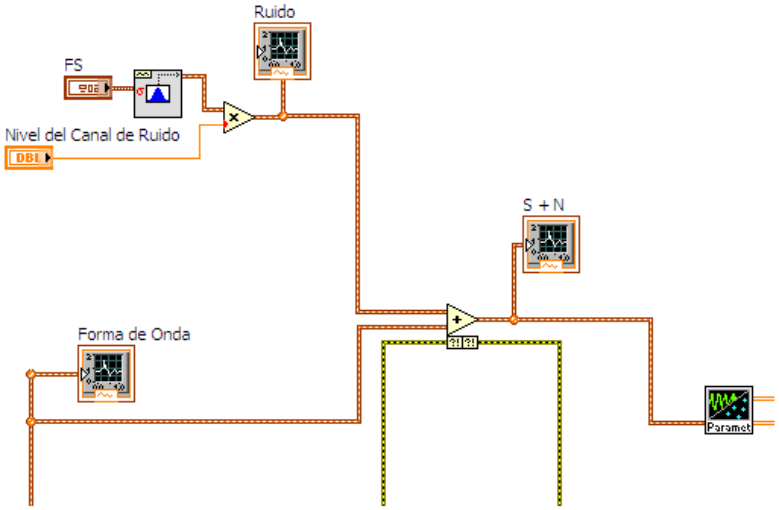

Figura 25 Código Fuente: Caracterización del Ruido
Fuente: Elaboración Propia

Una vez implementado este sistema de ruido, se vuelven a realizar algunas de las pruebas anteriores, esta vez afectadas por el canal ruidoso, y se toman los resultados para realizar comparaciones con lo obtenido previamente.

III.7 Actualización del manual de usuario

Para culminar, se realiza una actualización al Manual de Usuario que explica la correcta configuración y utilización de la calculadora VOXAG, de forma tal de corregir las referencias que pudieran hacerse a las características modificadas de la versión anterior del sistema.

Dicho manual está pensado para explicar cuidadosamente los pasos que debe seguir el usuario para usar el sistema desde la conexión del micrófono a utilizar hasta la utilización de la calculadora y sus comandos de voz, para que logre familiarizarse de forma sencilla y rápida con la aplicación.

CAPÍTULO IV

Resultados

En este capítulo se presenta los distintos resultados obtenidos tras cada fase del desarrollo del Trabajo Especial de Grado, desde la familiarización con el sistema hasta la evaluación del mismo y su resistencia contra el ruido.

IV.1 Revisión y familiarización con el sistema de RAH

Tomando en cuenta el estudio que se hace del sistema de RAH desarrollado en el trabajo de investigación anterior, se obtiene el conocimiento del funcionamiento teórico y práctico de la aplicación, el cual se representa en un diagrama de flujo.

La Figura 26 presenta dicho diagrama de flujo, donde se observa la rutina principal del programa, así como las distintas subrutinas: la que aplica el filtro de preénfasis a la señal entrante; la de Detección de *Endpoint* únicamente para detectar el final de la señal; la de segmentación de la señal aplicando un solapamiento de 100 muestras; la de extracción de coeficientes MFCC de la señal; y por último, la de Cuantificación Vectorial; todas ellas basadas en la teoría explicada en el segundo capítulo de este documente correspondiente al Marco Teórico.

.

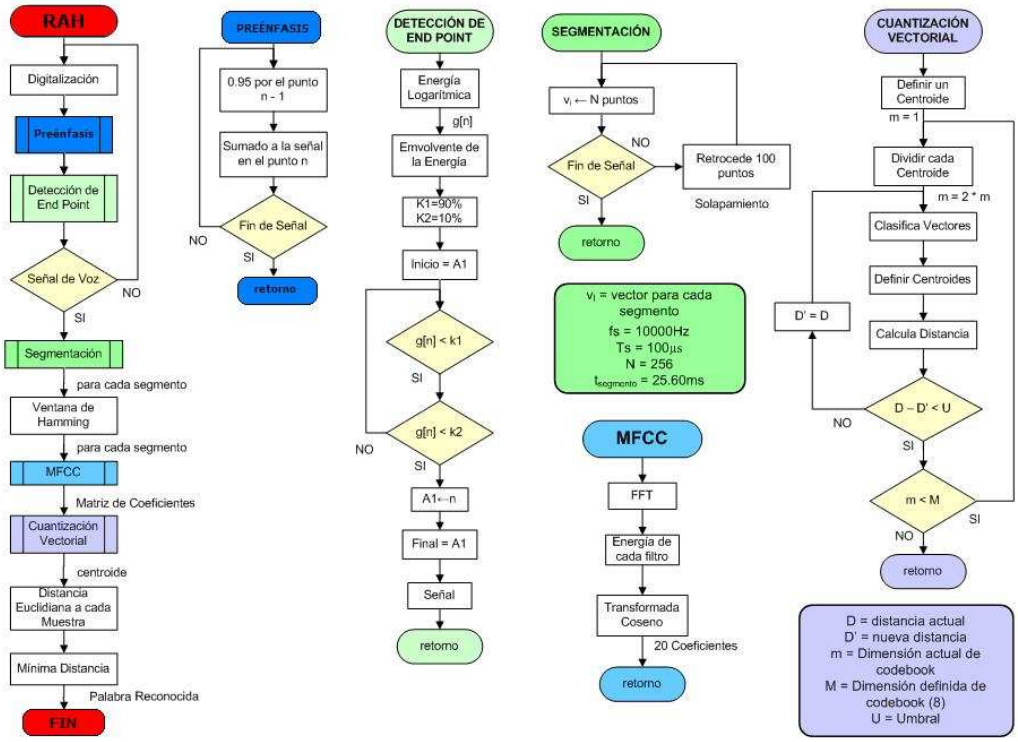

Figura 26 Diagrama de Flujo del sistema de RAH.

Fuente: Elaboración Propia.

IV.2 Comprobación de los resultados previos del sistema desarrollado

Como resultado de las pruebas de comprobación, en las que se repitió cada comando frente al sistema y se tomó nota de cuantas veces reconoció la palabra adecuadamente, se obtuvo lo siguiente:

IV.2.1. Prueba 1: Sujeto masculino, *codebook* 10 hombres y 10 mujeres

Comando	0	1	2	3	4	5	6	7	8	9	.	=	CE	<=	C	+	-	*	/	.-x	x^1/2	x^2	1/x
Aciertos	4	7	17	1	6	14	6	4	7	13	9	13	2	13	8	9	13	4	13	8	8	15	6
% de Acierto	20	35	85	5	30	70	30	20	35	65	46	65	10	65	40	45	65	20	65	40	40	75	30

Total	43.48%

Tabla 1 Prueba del Sistema, Sujeto masculino, *codebook* 10 hombres y 10 mujeres.

Fuente: Elaboración Propia.

Se observa un resultado bajo y similar a la investigación anterior, donde se obtuvo un 51,09% de acierto, como se muestra en la Tabla 2.

Recon. / Estud.	Cero	Uno	Dos	Tres	Cuatro	Cinco	Seis	Siete	Ocho	Nueve	Punto	Igual	Entrada	Borrar	Reiniciar	Mas	Menos	Por	Entre	Opuesto	Raiz	Cuadrado	Inverso
Cero	18													1								1	
Uno		12	1			1						1		1	1		1					2	
Dos			1						7			2			6			1				3	
Tres				8						2					6								4
Cuatro			1		3					1					5							10	
Cinco				1	1	10	1								4		1		1				1
Seis							16								2						2		
Siete								3							16		1						
Ocho	1		4						5	2		6						1				1	
Nueve	2									17					1								
Punto		8							1		6	1			4								
Igual		1			1							18											
Entrada													6		1							13	
Borrar					4									13	1			1				1	
Reiniciar				1			5								13								1
Mas														1	1	15						3	
Menos	2		2														13					2	1
Por		1	3		1				3									12					
Entre			3				2			1		1					3		8				2
Opuesto					2	5			1	1		2								9			
Raíz			1	2			1								9						7		
Cuadrado					1									1								18	
Inverso	2		3							2					6		3						4
% Acierto	90.0%	60.0%	5.0%	40.0%	15.0%	50.0%	80.0%	15.0%	25.0%	85.0%	30.0%	90.0%	30.0%	65.0%	65.0%	75.0%	65.0%	60.0%	40.0%	45.0%	35.0%	90.0%	20.0%
																				% TOTAL		51.09%	

Tabla 2 Matriz de confusión: Sujeto masculino. Base de Datos, 10 hombres, 10 mujeres (Sistema sin mejoras)

Fuente: (Rodríguez, 2010)

IV.2.2. Prueba 2: Sujeto masculino, *codebook* 20 hombres

Comando	0	1	2	3	4	5	6	7	8	9	.	=	CE	<=	C	+	-	*	/	.-x	x^1/2	x^2	1/x
Aciertos	8	15	9	4	11	16	13	9	15	11	18	11	10	15	7	17	4	6	6	6	7	14	9
% de acierto	40	75	45	20	55	80	65	45	75	55	90	55	50	75	35	85	20	30	30	30	35	70	45

Total	52,39 %

Tabla 3 Prueba del Sistema, Sujeto masculino, *codebook* 20 hombres.

Fuente: Elaboración Propia.

A diferencia de la investigación previa, en la cual se obtuvo una mejora llegando al 70% como se muestra en la Tabla 4, en este caso no se observa un gran avance, debido probablemente a que el sistema no estaba entrenado con la voz del locutor.

Recon. / Estud.	Cero	Uno	Dos	Tres	Cuatro	Cinco	Seis	Siete	Ocho	Nueve	Punto	Igual	Entrada	Borrar	Reiniciar	Mas	Menos	Por	Entre	Opuesto	Raíz	Cuadrado	Inverso
Cero	18																					1	1
Uno		12									3		1				2	1				1	
Dos			11						1							2		6					
Tres	2		1	13					1										1				2
Cuatro			6		8				4									1				1	
Cinco		1				13									2				4				
Seis				1			11								6						2		
Siete						2	4	5											5		3		1
Ocho			2		2	5			8							1				1			1
Nueve			2							16									1	1			
Punto		4									15						1						
Igual					1							15		1				1				2	
Entrada													13									7	
Borrar					2									16						1		1	
Reiniciar															18				1		1		
Mas	1		1						1				1			16							
Menos																	20						
Por			4															16					
Entre				1			1										2	1	14				1
Opuesto		1			1						1	1		1						15			
Raíz															3						17		
Cuadrado														3								17	
Inverso	1			2					1										1				15
% Acierto	90.0%	60.0%	55.0%	65.0%	40.0%	65.0%	55.0%	25.0%	40.0%	80.0%	75.0%	75.0%	65.0%	80.0%	90.0%	80.0%	100%	80.0%	70.0%	75.0%	85.0%	85.0%	75.0%
																				% TOTAL		70.00%	

Tabla 4 Matriz de confusión: Sujeto masculino. Base de Datos, 20 hombres (Sistema sin mejoras)

Fuente: (Rodríguez, 2010)

IV.2.3. Prueba 3: Sujeto femenino, *codebook* 20 hombres, 10 mujeres

Comando	0	1	2	3	4	5	6	7	8	9	.	=	CE	<=	C	+	-	*	/	.-x	x^1/2	x^2	1/x
Aciertos	13	6	9	6	9	11	15	4	13	6	2	15	3	3	3	3	10	9	6	3	2	6	7
% de acierto	65	30	45	30	45	55	75	20	65	30	10	75	15	15	15	15	50	45	30	15	10	30	35

Total	35,65 %

Tabla 5 Prueba del Sistema, Sujeto femenino, *codebook* 20 hombres, 10 mujeres.

Fuente: Elaboración Propia.

Al igual que en la investigación anterior, se obtiene un resultado por debajo del 50%, lo que confirma el problema del sistema para las voces femeninas.

Recon. / Estud.	Cero	Uno	Dos	Tres	Cuatro	Cinco	Seis	Siete	Ocho	Nueve	Punto	Igual	Entrada	Borrar	Reiniciar	Mas	Menos	Por	Entre	Opuesto	Raiz	Cuadrado	Inverso
Cero	10			2	1									1			1					2	3
Uno	1	2			1			1			1	1	1	8				1		1		2	
Dos			7						2							7						4	
Tres	1			11				2						1									5
Cuatro					10				2							2		1				5	
Cinco					3	7		1			2				2	1		1		1		2	
Seis							11								6						3		
Siete	1			3	1		1	5	1								1	1					6
Ocho			1		5				7							5		2					
Nueve								1		13				1					1			3	1
Punto		9	1								5	4						1					
Igual					1					3		13										3	
Entrada				1						1			8					1	2			2	5
Borrar											1		13			1		2				3	
Reiniciar				2			3								11				1		3		
Mas	1			1									2			15						1	
Menos	5			3													5		1				6
Por			5		3			1	1		1	1		1		1		6					
Entre								1											9				10
Opuesto			1		5				3	1		1		2	1			3		3			
Raíz							4						2		2		1				8	2	1
Cuadrado													6	5								9	
Inverso	1			3			1	2						2	3		1		2		1	1	3
% Acierto	50.0%	10.0%	35.0%	55.0%	50.0%	35.0%	55.0%	25.0%	35.0%	65.0%	25.0%	65.0%	40.0%	65.0%	55.0%	75.0%	25.0%	30.0%	45.0%	15.0%	40.0%	45.0%	15.0%
																				% TOTAL		41.52%	

Tabla 6 Matriz de confusión: Sujeto femenino. Base de Datos, 20 hombres, 10 mujeres (Sistema sin mejoras)

Fuente: (Rodríguez, 2010)

IV.3 Diseño e implementación de mejoras del sistema de RAH

Los cambios implementados y explicados en el tercer capítulo de este trabajo afectan tanto al programa de grabación de parámetros como al de reconocimiento de palabras. El primero, que se puede observar en la Figura 27, ahora logra mostrar la matriz de 16 centroides, de forma numérica y gráfica

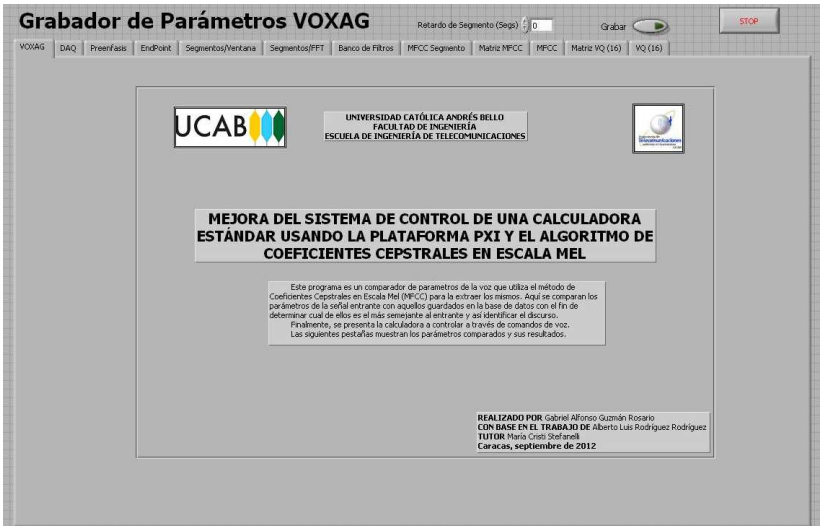

Figura 27 Interfaz del grabador de parámetros.

Fuente: Elaboración Propia.

En la Figura 28 se observa la representación numérica de la palabra "cuadrado", mientras que en la Figura 29 se aprecia la gráfica de la misma palabra, ambas con una prueba de 16 centroides.

Figura 28 Matriz VQ, 16 centroides.

Fuente: Elaboración Propia.

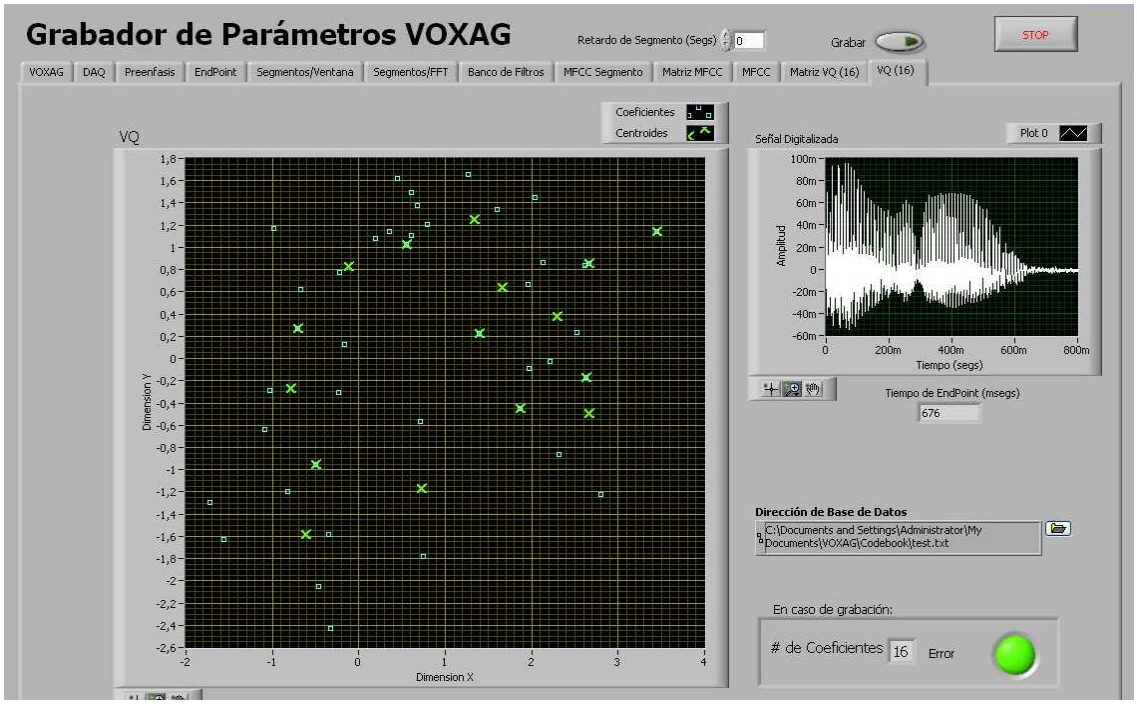

Figura 29 Gráfica VQ, 16 centroides.

Fuente: Elaboración Propia.

En cuanto al programa principal de comparación de palabras, el cual puede observarse en la Figura 30, ahora se permite trabajar con *codebooks* más largos, debido a la adaptación a *codewords* de 16 centroides.

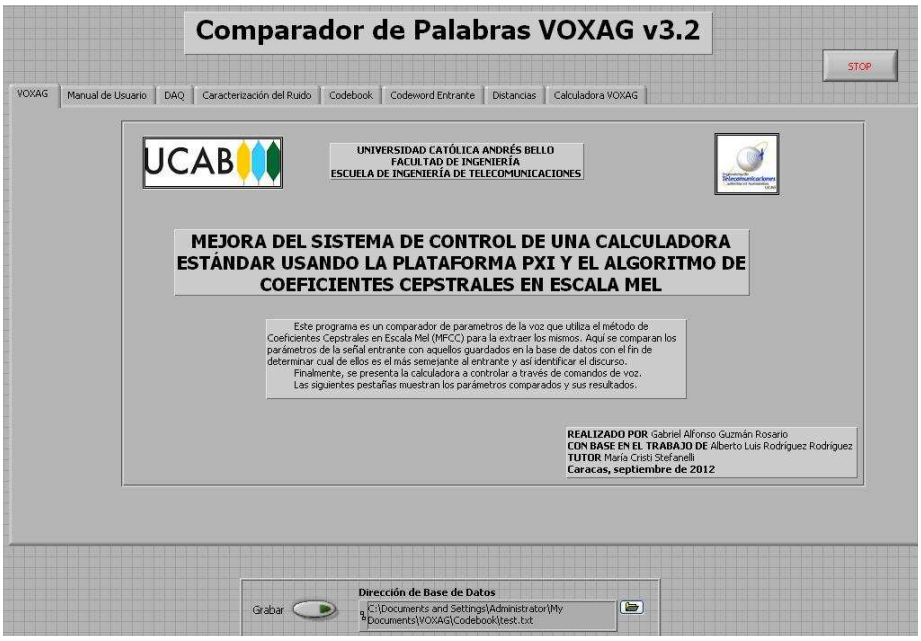

Figura 30 Interfaz del programa principal.

Fuente: Elaboración Propia.

En la Figura 31 se observa la primera palabra de la nueva matriz de centroides que representa el *codebook* completo contra el que se compararán las muestras entrantes.

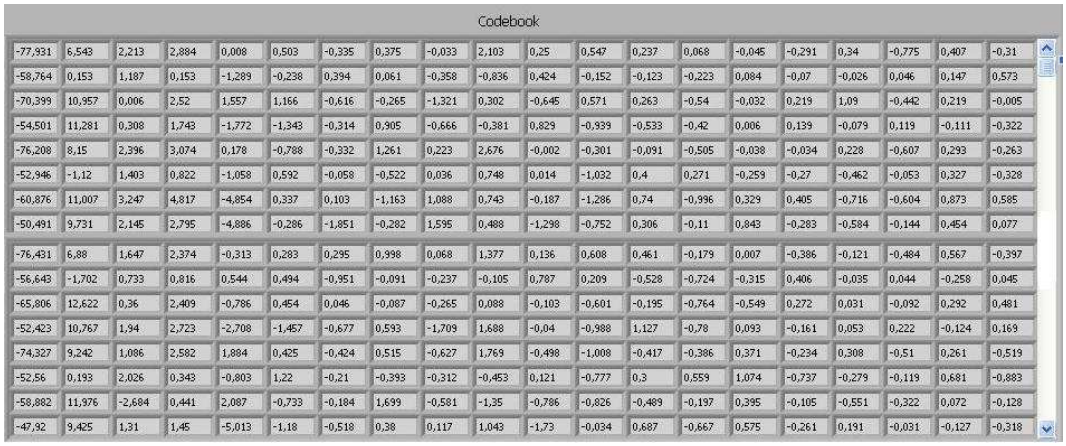

Figura 31 Interfaz: *Codebook*.

Fuente: Elaboración Propia.

En dicho *codebook* se encuentra una palabra distinta cada 16 filas, lo que representa un total de 1380 muestras para el diccionario más corto (30 individuos) y de 2760 para el más largo (60 individuos). Esto nos arroja matrices que disponen entre 22080 y 44160 filas, manteniendo 20 columnas de forma constante. A pesar de estas longitudes, no se observó un retraso significativo en el tiempo de respuesta del sistema, por lo que se comprueba que se puede trabajar con *codebooks* de mayor escala.

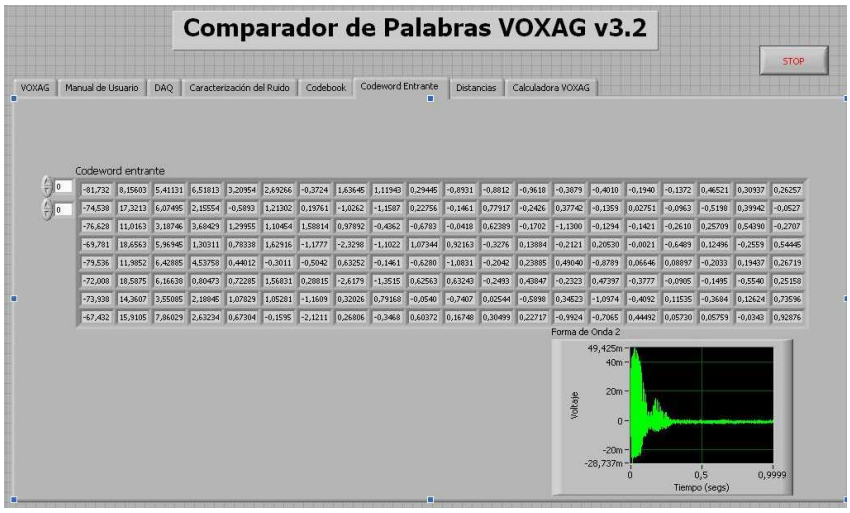

Figura 32 *Codebook* entrante, primera parte.

Fuente: Elaboración Propia.

De igual manera, se presenta la matriz de centroides de la palabra entrante, a la cual se le ha aplicado el mismo procedimiento por el que pasaron las palabras que conforman el código. Esto puede apreciarse en las figuras 32 y 33, donde se comprueba que la matriz se compone de 16 filas.

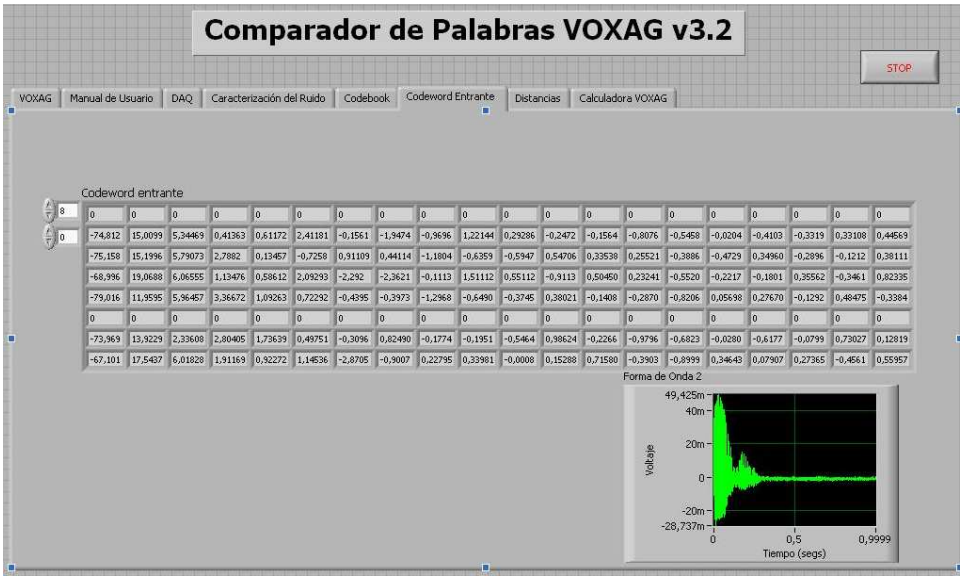

Figura 33 *Codebook* entrante, segunda parte.

Fuente: Elaboración Propia.

IV.4 Nuevo entrenamiento del sistema de RAH

Como resultado del entrenamiento del sistema, se obtiene una base de datos por cada persona, mediante la cual se describen las distintas palabras que forman los *codebooks*. Dicha base de datos queda guardada en el sistema como un archivo con extensión *.txt*, el cual se lee desde la aplicación. En la figura 34 se observa un ejemplo de esta representación.

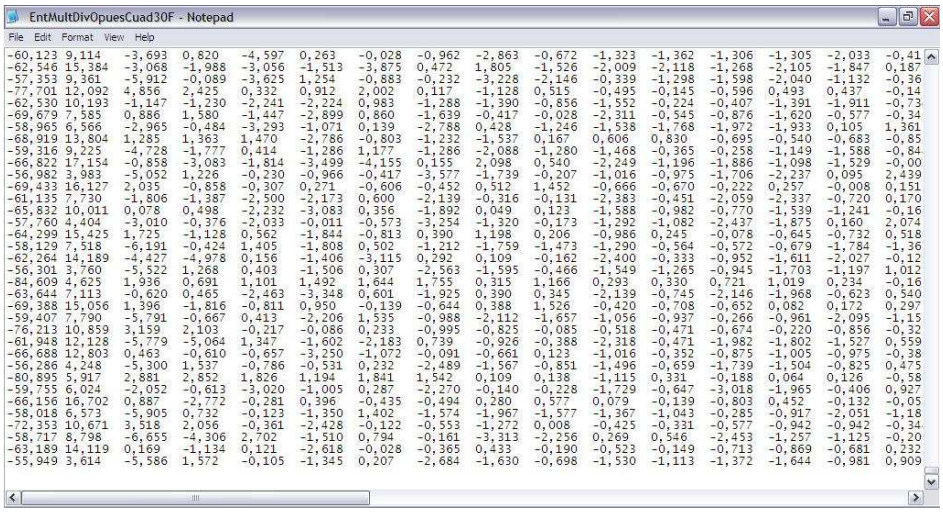

Figura 34 Ejemplo de *codebook*.

Fuente: Elaboración Propia.

Con las muestras de los 60 individuos (30 masculinos y 30 femeninos), se crearon los diversos diccionarios para el uso de la aplicación, los cuales varían principalmente en cantidad de individuos, obteniendo así *codebooks* de 30, 40 y 60 personas.

IV.5 Evaluación del sistema de RAH

Tras realizar las pruebas para llenar las matrices de confusión, se descubrió que el sistema no alcanzaba los mínimos requerimientos de eficiencia necesarios, como se observa a continuación:

IV.5.1. Prueba 4: Sujeto masculino, *codebook* 30 hombres, 16 centroides

La Tabla 7 es la matriz de confusión para la prueba realizada con un sujeto masculino y un *codebook* compuesto por 30 hombres, según los procedimientos explicados anteriormente:

	Cero	Uno	Dos	Tres	Cuatro	Cinco	Seis	Siete	Ocho	Nueve	Punto	Igual	Entrada	Borrar	Reiniciar	Más	Menos	Por	Entre	Opuesto	Raíz	Cuadrado	Inverso
Cero	4	1	1			1	3		1				1					1	3			1	3
Uno	1	5			1				1		5	3						4					
Dos	1	1	3	3	1					1	1	1	1				1	3		1			2
Tres	3	1		4		2	3	2										1	4				
Cuatro		1	1	3	6				1	1		1		1				3				2	
Cinco	2					4	5		3	1	1	2								2			
Seis	2	1		1			8								4						2	1	1
Siete	1			1		1	5	11											1				
Ocho	1				1	2	1		14									1					
Nueve	1							2		8		6						2	1				
Punto	1	5	2		2						8							1		1			
Igual		2									2	3	1	5		1		2				4	
Entrada	1												5	3	2		1		3		1	4	
Borrar		1								3			1	14									1
Reiniciar		1													10			2	7				
Más	2		1	1		1					2		5			6		1			1		
Menos	2	1		1		1				1		1	2		4		5					1	1
Por	1	1	2	1	4					2	1		2	1				4	1				
Entre				2	1	1	2	3									1		9				1
Opuesto	1				1										2					16			
Raíz				1			1								5		1				10		2
Cuadrado		1			1								1	5				1				11	
Inverso							1		1				3		4								11
% Acierto	20	25	15	20	30	20	40	55	70	40	40	15	25	70	50	30	25	20	45	80	50	55	55
																			% TOTAL			38,913	

Tabla 7 Matriz de confusión: Sujeto Masculino, *codebook*: 30 hombres.

Fuente: Elaboración Propia.

Un porcentaje de 38,91% resulta extremadamente bajo, teniendo en cuenta lo visto en la Tabla 4, donde en las pruebas con 8 centroides y un número menor de muestras se alcanzó un 70% de eficiencia.

Todas las pruebas realizadas con estos *codebooks* daban resultados bajos e ineficientes; muy aleatorios, confundiendo palabras sin mucha relación entre si como “entre” y “reiniciar”, sin importar si el locutor era hombre o mujer.

IV.5.2. Efecto de la duración de las palabras sobre el sistema

Tras una revisión exhaustiva del código y del procesamiento de la señal, se determinó que todo estaba funcionando como debería, por lo que el problema debía estar en los *codebooks*. Específicamente, se observó que existían centroides nulos, formando una gran cantidad de filas cuyos parámetros eran iguales a cero, como puede verse en la Figura 35.

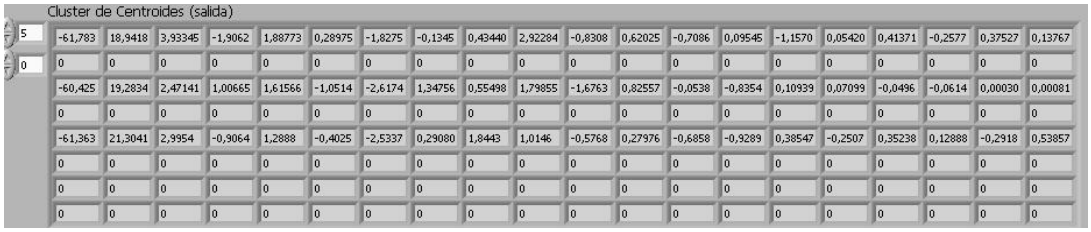

Figura 35 Centroides nulos en la palabra “dos”, 16 centroides.
Fuente: Elaboración Propia.

Viendo que esto sucedía principalmente con las palabras de corta duración, se decide llevar a cabo otras pruebas específicas tanto con el programa grabador como con el comparador, que demuestren el efecto de la duración de las palabras sobre el sistema.

Se aprovechó la capacidad del programa grabador para graficar los centroides y observar que sucedía entre palabras cortas y largas. En la Figura 36 se observan los centroides de la palabra “dos”, teniendo el sistema funcionando con 16 centroides.

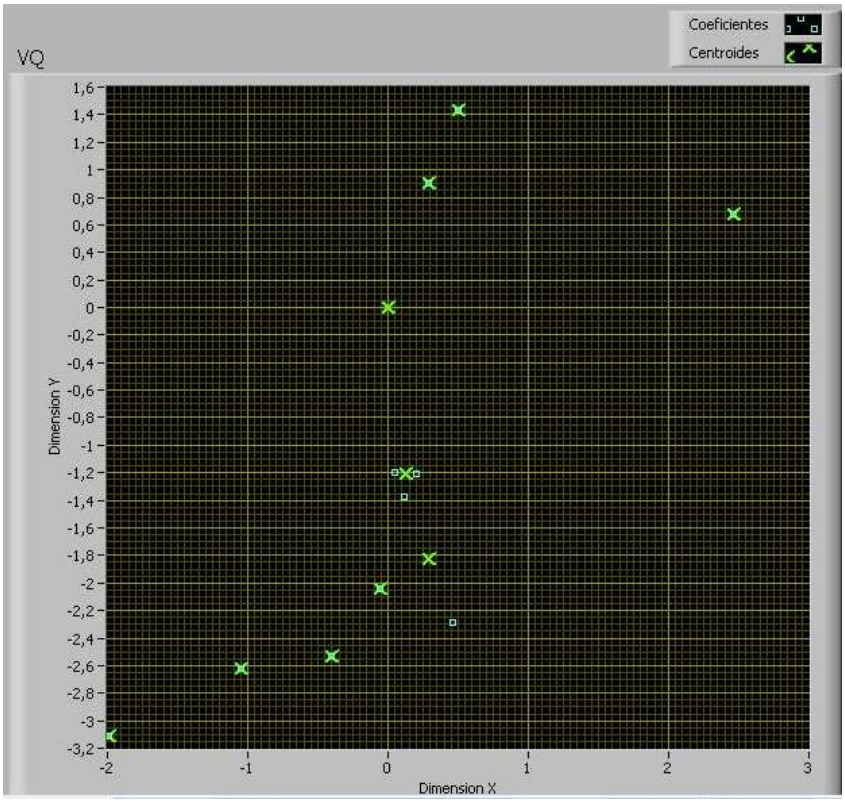

Figura 36 Gráfica de la palabra "dos", a 16 centroides.

Fuente: Elaboración Propia.

Lo primero que se puede notar es la presencia de tan solo 10 centroides, en vez de los 16 esperados, lo que corresponde con las filas nulas vistas en la Figura 35. Al hacer la misma prueba usando la aplicación con 8 centroides, se nota que las filas están completas, como se muestra en la Figura 37.

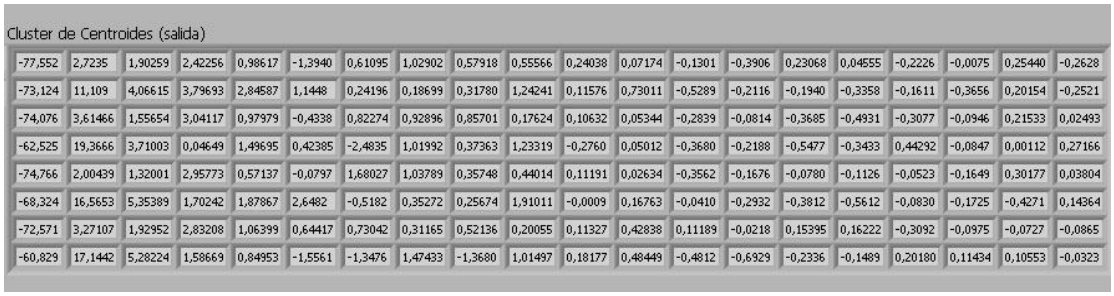

Figura 37 Centroides completos en la palabra "dos", 8 centroides.

Fuente: Elaboración Propia.

A su vez, la gráfica muestra los 8 centroides, claramente diferenciados entre si y de los coeficientes, como puede verse en la Figura 38.

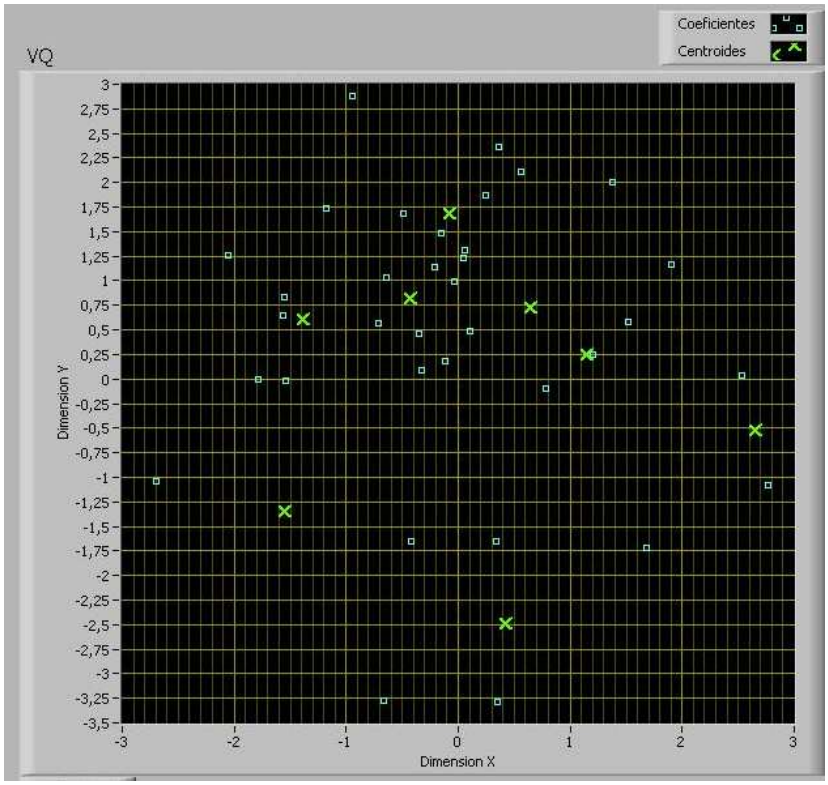

Figura 38 Gráfica de la palabra "dos", a 8 centroides.

Fuente: Elaboración Propia.

De igual manera se hace el estudio con la palabra "cuadrado", la cual no presenta filas nulas en ninguno de los dos casos (8 y 16 centroides), como se puede ver en la Figura 39.

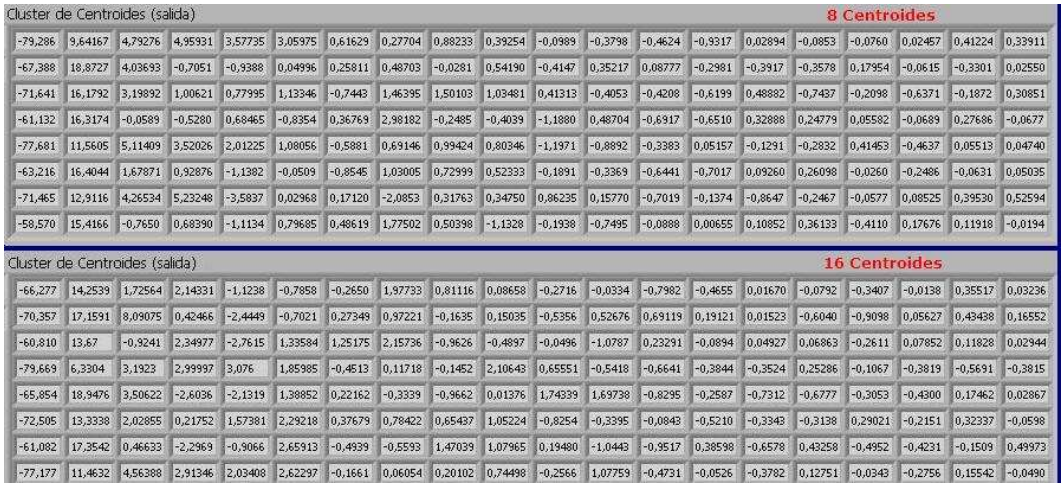

Figura 39 Centroides completos en la palabra "cuadrado", 8 y 16 centroides..

Fuente: Elaboración Propia.

Gráficamente se puede apreciar mejor como la palabra “cuadrado” queda representada adecuadamente para ambas versiones de la aplicación, tanto para 8 como para 16 centroides, como se ve en las Figuras 40 y 41, respectivamente.

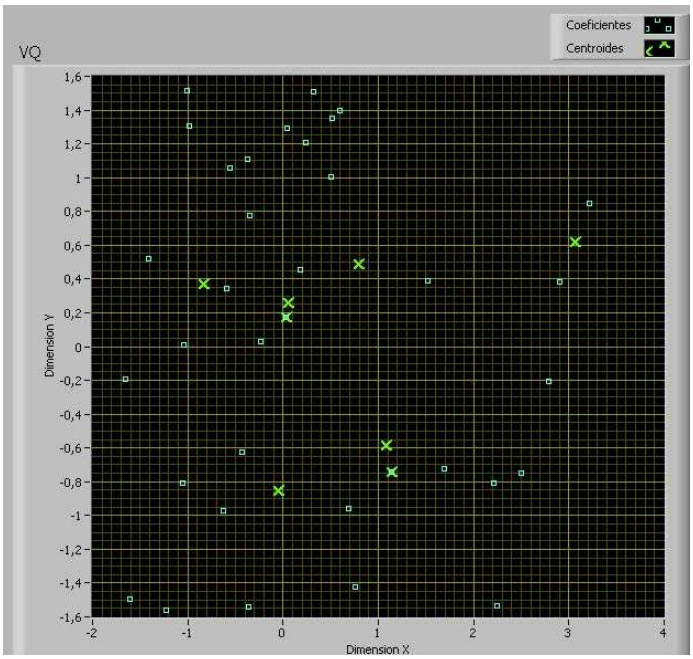

Figura 40 Gráfica de la palabra “cuadrado”, a 8 centroides.

Fuente: Elaboración Propia.

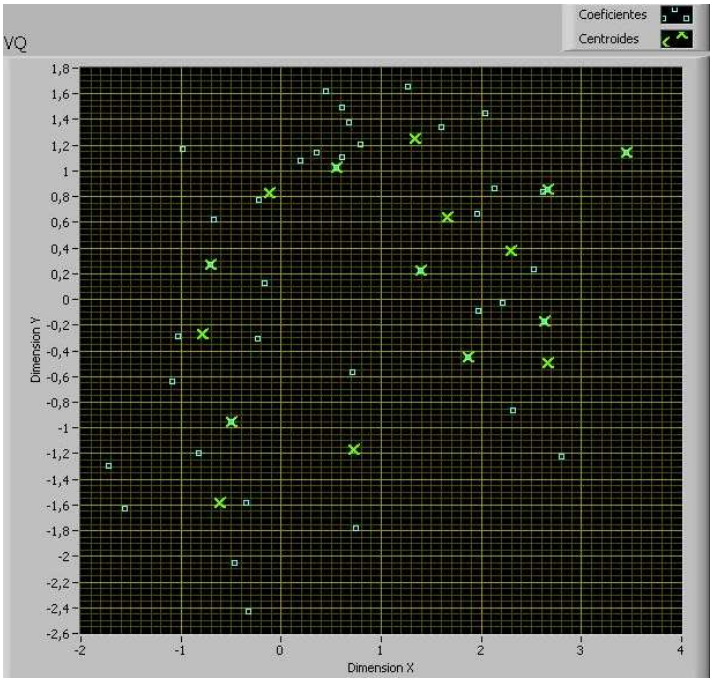

Figura 41 Gráfica de la palabra “cuadrado”, a 16 centroides.

Fuente: Elaboración Propia.

La misma prueba se realiza con las palabras “uno” y “opuesto”, obteniendo resultados similares, por lo que se plantea el problema de que las palabras cortas aportan un exceso de centroides nulos a los *codebooks*, lo cual afecta al sistema al momento de calcular la mínima distancia entre palabras.

Para demostrar la validez de este planteamiento, se crean dos nuevos *codebooks* de cinco palabras cada uno: uno compuesto por palabras largas y otro por palabras corta, de forma tal de realizar pruebas para evaluar su eficiencia.

IV.5.3. Prueba 5: Sujeto masculino, *codebook* de 5 palabras largas y 30 hombres, 16 centroides

El primer *codebook* de prueba está conformado por cinco palabras largas: entrada, multiplica, divide, opuesto y cuadrado, a partir de las muestras de 30 sujetos masculinos. El procedimiento es igual que antes, se pronuncia cada palabra 20 veces frente al sistema y se registran los resultados, los cuales se muestran en la Tabla 8.

	Entrada	Multiplica	Divide	Opuesto	Cuadrado
Entrada	18				2
Multiplica	2	17	1		
Divide			20		
Opuesto	1	1		18	
Cuadrado	1				19
% Acierto	90	85	100	90	95

% TOTAL	92

Tabla 8 Matriz de confusión: sujeto masculino, 5 palabras largas.

Fuente: Elaboración Propia.

Se observa una eficiencia del sistema de más del 90%, un resultado extremadamente favorable y más acorde a lo esperado tras implementar las mejoras.

IV.5.4. Prueba 6: Sujeto masculino, *codebook* de 5 palabras cortas y 30 hombres, 16 centroides

El segundo *codebook* se forma con cinco palabras cortas: uno, dos, seis, ocho y más, de igual manera con las muestras de 30 sujetos masculinos. Los resultados se muestran en la Tabla 9.

	Uno	Dos	Seis	Ocho	Mas
Uno	2	1	3	8	6
Dos	8	4		5	3
Seis	4	4	6	4	2
Ocho	5	4		8	3
Mas	14	2		2	2
% Acierto	10	20	30	40	10

% TOTAL	22

Tabla 9 Matriz de confusión: sujeto masculino, 5 palabras cortas.
Fuente: Elaboración Propia.

El porcentaje de acierto cae drásticamente a 22%, lo que nos demuestra que para el caso de 16 centroides, el uso de palabras largas es fundamental en esta aplicación.

IV.5.5. Prueba 7: Sujeto femenino, *codebook* de 5 palabras largas y 30 mujeres, 16 centroides

Se realizan las mismas pruebas anteriores pero esta vez con un sujeto femenino, y un *codebook* conformado por 30 mujeres Los resultados de esta evaluación se presentan en la Tabla 9.

	Entrada	Multiplica	Divide	Opuesto	Cuadrado
Entrada	14	2			4
Multiplica	2	13	3	1	1
Divide	2	2	16	1	
Opuesto	1	1		17	3
Cuadrado	3				17
% Acierto	70	65	75	85	85

% TOTAL	76

Tabla 10 Matriz de confusión: sujeto femenino, 5 palabras largas.

Fuente: Elaboración Propia.

Se observa que si bien el resultado no llega al nivel de eficiencia de la prueba 5, aun así representa una mejora significativa con respecto a lo que se obtuvo en la investigación anterior con 8 centroides.

IV.5.6. Prueba 8: Sujeto femenino, *codebook* de 5 palabras cortas y 30 mujeres, 16 centroides

Por último, se realiza la prueba de las palabras cortas con el sujeto femenino, cuyos resultados se muestran en la Tabla 11.

	Uno	Dos	Seis	Ocho	Mas
Uno	7	4	3	3	3
Dos	3	8	3	5	4
Seis	3	7	4	3	3
Ocho	1	1	2	10	6
Mas	2	3	2	3	10
% Acierto	35	40	20	50	50

% TOTAL	39

Tabla 11 Matriz de confusión: sujeto femenino, 5 palabras cortas.

Fuente: Elaboración Propia.

La eficiencia disminuye a 39%, terminando de reafirmar el efecto negativo que tienen las palabras cortas sobre el sistema.

IV.6 Estudio del sistema de RAH frente al ruido

Debido a los resultados de las pruebas anteriores, realizar un estudio del efecto del ruido sobre el sistema sería poco productivo si se hace sobre la aplicación con el cambio a 16 centroides y el diccionario completo. Es por este motivo que la prueba se llevan a cabo primeramente sobre la aplicación funcionando con 8 centroides, seguida de una evaluación del ruido sobre el *codebook* de 5 palabras largas a 16 centroides.

Como resultado del código mostrado en el tercer capítulo, el programa ahora cuenta con una interfaz donde puede controlarse el nivel de ruido, así como también verlo representado gráficamente. Dicha interfaz se observa en la Figura 42.

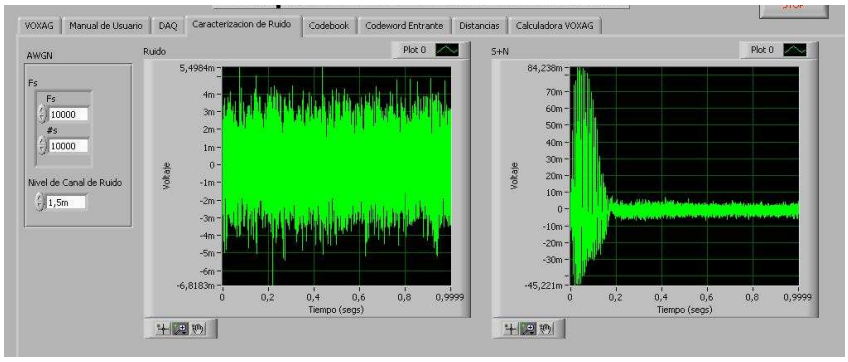

Figura 42 Interfaz de usuario: Caracterización del Ruido.

Fuente: Elaboración Propia.

IV.6.1. Prueba 9: Sujeto masculino, *codebook* de 20 hombres, 8 centroides

Para este estudio, se repite la Prueba 2, añadiendo el ruido con un nivel de 1,5 mV. Esto arroja una relación señal a ruido estimada de 239, lo que representa una magnitud de ruido relativamente pequeña con respecto a la señal de información entrante. Los resultados de esta prueba se muestran en la Tabla 12.

	Cero	Uno	Dos	Tres	Cuatro	Cinco	Seis	Siete	Ocho	Nueve	Punto	Igual	Entrada	Borrar	Reiniciar	Más	Menos	Por	Entre	Opuesto	Raíz	Cuadrado	Inverso
Cero	0	1		1		3	1								14								
Uno		0		8		1					2				8			1					
Dos			2	9			4										2	2			1		
Tres				12		4	2								2								
Cuatro			1	2	9				2	1		1						1					3
Cinco			1			11		2			1	1			4								
Seis				6			0	2							12								
Siete			2			4		7											7				
Ocho			2					6	4									4					4
Nueve				3						3					14								
Punto				6				3			1				10								
Igual	1			4				1	1	5		0			5						3		
Entrada				10		1				1			5		2								1
Borrar				8										2	6								4
Reiniciar				5											13						2		
Más				3									9		1	2					3		2
Menos				7											9		1			2			1
Por			6	2														4					8
Entre				9		3									7				1				
Opuesto				11											4					2			3
Raíz			2	14			2														2		
Cuadrado				3	3							4			8							2	
Inverso				4							1				10					4			1
% Acierto	0	0	10	60	45	55	0	35	20	15	5	0	25	10	65	10	5	20	5	10	10	10	5
																			% TOTAL		18,26		

Tabla 12 Matriz de confusión: sujeto masculino, *codebook* 20 hombres, canal ruidoso.

Fuente: Elaboración Propia.

Se observa que aun cuando el nivel de ruido añadido fue relativamente pequeño, la eficiencia cae significativamente. En este caso, de un 70% a menos de 20%, lo que indica lo poco resistente que es el sistema ante el ruido.

IV.6.2. Prueba 10: Sujeto masculino, *codebook* de 5 palabras largas y 30 hombres, 16 centroides

En esta evaluación se añade ruido con la misma magnitud de la prueba anterior (1,5 mV) y se utiliza el *codebook* de 5 palabras largas elaborado en la Prueba 5, de forma tal de poder comparar el efecto de las interferencias adversas a la señal. El resultado de esta prueba se muestra en la Tabla 13.

	Entrada	Multiplica	Divide	Opuesto	Cuadrado
Entrada	16	1			3
Multiplica	1	12	2	4	1
Divide	2	3	11		4
Opuesto	4	2		13	1
Cuadrado	3				17
% Acierto	80	60	55	65	85

% TOTAL	69

Tabla 13 Matriz de confusión: sujeto masculino, 5 palabras largas, canal ruidoso.

Fuente: Elaboración Propia.

Se puede observar como la caída de la eficiencia no es tan radical como cuando se usan 8 centroides. En este caso, el sistema presenta un porcentaje de acierto de casi 70%, una disminución del 22% con respecto a la Prueba 5. A pesar de disminuir, el resultado se mantiene muy cerca a las mejores evaluaciones obtenidas con 8 centroides, lo que demuestra que el uso de 16 centroides ayuda a fortalecer el sistema contra el ruido.

Se debe tomar en cuenta que solo se están usando 5 palabras, por lo que la comparación no puede hacerse a la ligera, pero sí se puede decir que el cambio representa una mejora en este aspecto.

IV.7 Actualización del manual de usuario

A pesar de que las pruebas no arrojaron los resultados requeridos para decir que se tiene una versión definitiva de la aplicación funcionando con 16 centroides, se decidió actualizar el manual de usuario, adecuándolo a dicha mejora. El manual actualizado es un texto sencillo que presenta el sistema al usuario y le guía a través de las etapas de instalación del micrófono en la plataforma PXI, la configuración de la tarjeta de adquisición del mismo, la configuración de la base de datos de entrenamiento a utilizar, los parámetros de control que se pueden observar para el proceso de la palabra, el uso de la calculadora en sí, y por último los comandos de voz que esta utiliza. Este manual se encuentra adjunto a este documento en el Apéndice B.

Después de realizarse el trabajo de investigación, se encontró que el valor del coeficiente del filtro de preénfasis no creaba el filtro esperado, lo que podría afectar en gran medida los resultados del sistema. Para comprobar esto, se efectuaron ciertas pruebas cuyos resultados se muestran en el Apéndice C, por haber sido realizadas después de la presentación del trabajo.

CAPÍTULO V

Conclusiones y Recomendaciones

Durante el desarrollo de esta investigación se trabajó con el sistema de Reconocimiento Automático del Habla implementado por el ingeniero Alberto Rodríguez en un estudio anterior. A dicho sistema se le aplicaron una serie de cambios, principalmente el aumento del número de centroides usados de 8 a 16, en busca de mejorar su eficiencia y de estudiar su fortaleza ante el ruido. Una vez realizadas las pruebas necesarias, se pueden concluir las ideas planteadas a continuación.

Primeramente se ha validado el correcto funcionamiento del sistema y los resultados previos obtenidos, aún así con la diferencia que existe entre los distintos locutores que probaron el sistema durante el transcurso de ambas investigaciones, demostrando el potencial que tiene la aplicación como una valiosa herramienta de procesamiento y reconocimiento de la voz.

Si bien el preprocesamiento aplicado a la señal y el algoritmo MFCC son adecuados para el reconocimiento del habla dentro de esta aplicación, la misma es muy susceptible a los cambios que pueden llegar a sufrir los *codewords* y *codebooks*. Es por esto que debe usarse un diccionario lo menos extenso posible, entrenado con una variedad de sujetos de ambos sexos. En este caso, el *codebook* más grande se compuso de 60 personas (30 mujeres y 30 hombres, 44160 filas y 20 columnas) sin que se viera afectado significativamente el tiempo de respuesta del sistema. A pesar de esto, no se recomienda un aumento del número de muestras de entrenamiento; en parte, por lo engorroso y poco práctico que puede llegar a ser conseguir más de 60 voluntarios para entrenar al sistema, y porque llegados a este punto, una excesiva cantidad de muestras no representaría una mejora proporcional a la eficiencia de la aplicación.

Observando los resultados, destaca el alto porcentaje de eficiencia que obtiene la aplicación al utilizar 16 centroides y palabras largas, demostrando que

dicho cambio sí es favorable y que es un camino que vale la pena tomar. Se recomienda entonces dirigir los esfuerzos para la mejora del sistema en dos posibles direcciones: la primera y más evidente es la creación y utilización de *codebooks* pensados para su uso con 16 centroides, ya que como se pudo comprobar, el aumento de centroides sí logra mejorar la eficiencia. Esto implica definir nuevos comandos, cuya duración temporal sea más larga que los de la mayoría de las palabras que se utilizaron. “Entrada, “opuesto” y “multiplica” son buenos ejemplos de comandos que funcionan perfectamente con el cambio a 16 centroides, mientras que “por”, “dos” o incluso las palabras bisílabas como “ocho” vuelven al sistema demasiado aleatorio. El mayor problema de esto será ingeniar comandos representativos y prácticos cuya longitud sea la adecuada, especialmente para los números del cero al nueve.

Si crear una nueva lista de comandos se torna muy dificultoso, se recomienda hacer pruebas con este sistema entrenado a una mayor frecuencia de muestreo, tomando especial cuidado en mantener la duración de los segmentos dentro del rango establecido en la teoría. Dado que en esta investigación se trabajó con un valor de Fs de 10kHz, esta podría ser aumentada a 12kHz para comprobar su funcionamiento, ya que teóricamente el aumento de la frecuencia de muestreo debería mejorar el desempeño ante las palabras cortas, debido a que el sistema toma más muestras de las mismas.

La segunda mejora que se puede realizar es un cambio o replanteamiento del algoritmo para la comparación entre palabras o cálculo de distancias. El método de la Distancia Euclidiana que se ha utilizado hasta ahora es rápido y relativamente sencillo de implementar pero requiere de condiciones ideales para un funcionamiento óptimo, las cuales al comenzar a variar (presencia de centroides nulos, aumento en los niveles de la señal) afectan en gran medida la eficiencia. Algunos de los métodos alternativos que se han planteado con anterioridad son la Distancia Espectral Logarítmica, la Distancia Cepstral y la Distorsión de Itakura-Saito, por lo que se recomienda explorar la posibilidad y factibilidad de usar alguno de ellos.

Las pruebas muestran que el desempeño que presenta el sistema ante el ruido es pobre, aún cuando el ruido es bajo. La suma de ruido a la señal entrante afecta considerablemente al método de la Distancia Euclidiana, especialmente con la cantidad de palabras que se están manejando en la aplicación. Una caída de aproximadamente 50% en la eficiencia al añadir ruido que no supera los 8 mV demuestra la gran debilidad del sistema ante condiciones adversas, a pesar del preprocesamiento que se le hace a la señal, ya que el algoritmo de detección de *endpoint* no elimina el ruido que se añade a la información útil y causa discrepancias importantes al momento de la comparación. Aún así, se puede ver una mejora de la respuesta ante el ruido al usar *codebooks* de 16 centroides, por lo que se recomienda hacer más pruebas en esa dirección.

Tomando en consideración las observaciones que se hicieron al filtro de preénfasis al presentar el trabajo, y al analizar las pruebas que se efectuaron en ese aspecto, se recomienda realizar nuevos experimentos con un diccionario formado por muestras que incorporen el cambio hecho al filtro, para constatar si se mantiene la igualdad en la eficiencia entre géneros, así como el desempeño general del sistema.

Debido a todo esto, se puede afirmar que aún no se ha logrado obtener una versión efectiva y confiable de la aplicación, pero se han dado pasos importantes para lograr esa meta, la cual requerirá algunos cambios si se quiere alcanzar: una mayor resistencia al ruido, un mejor cálculo de distancias y *codebooks* con comandos adaptados a la cantidad de centroides requeridos.

Bibliografía

- Álvarez, E. (2000). *Análisis acústico de la voz en adultos. Patrones de Normalidad.* Barquisimeto: Hospital Central Universitario "Antonio María Pineda" Barquisimeto. Recuperado el 12 marzo de 2012, de Universidad Centrooccidental Lisandro Alvarado: http://bibmed.ucla.edu.ve/DB/bmucla/edocs/textocompleto/TW4DV4A53a.pdf

- Antoniou, A. (2006). *Digital Signal Processing, Signals, Systems and Filters.* Victoria, British Columbia, Canada: McGraw-Hill.

- Carranza, F. (s.f.). *Determinación de endpoints y pitch en señales de voz.* Recuperado el 6 de enero de 2012, de Sociedad de Estudiantes de Ciencias de la Computación: http://seccperu.org/fcatho/lib/exe/fetch.php?id=speech_processing&cache=cache&media=endpoints_pitch_detection.pdf

- Goretti, I., & González, A. (s.f.). *Reconocedor de Dígitos.* Recuperado el 5 de enero de 2012, de Fundación Universitaria de Las Palmas: http://www.fulp.ulpgc.es/files/webfm/File/web/publicaciones/vectorplus/articulos/vp14_05_articulo01.pdf

- Huang, X., Acero, A., & Hon, H. (2001). *Spoken Language Processing: a Guide to Theory, Algorithm, and System Development.* New Jersey: Prentice Hall PTR.

- Méndez, J., Hernando, J., Nadeu, C., & Vallverdú, F. (s.f.). *Esquema Unificado de Parametrización de la Señal de Voz en Reconocimiento del Habla.* Recuperado el 6 de enero de 2012, de: http://www.lsi.upc.edu/~nlp/papers/hernando_esq.pdf

- National Instruments. (s.f.). *National Instruments PXI Description*. Recuperado el 3 de enero de 2012, de National Instruments: http://zone.ni.com/devzone/cda/tut/p/id/8404#toc2

- Rabiner, L., & Juang, B. (1993). *Fundamentals of Speech Recognition.* New Jersey: PTR Prentice-Hall.

- Rodríguez, A. (2010). *Reconocimiento del Habla Usando el Algoritmo de Coeficientes Ceptrales en Escala Mel para Controlar una Calculadora Estándar Bajo la Plataforma PXI.* Universidad Católica Andrés Bello, Caracas.

- Rodríguez del Río, R., & Zuazua Iriondo, E. (2003) *Series de Fourier y fenómeno de Gibbs.* Cubo. Matemática Educacional, 5 (2). pp. 185-224
- San Martín, C., & Carrillo, R. (Mayo de 2004). *Implementación de un Reconocedor de Palabras Aisladas Dependientes del Locutor*. Revista Facultad de Ingeniería, U.T.A Chile , 12 (1), págs. 9-14.
- Weisstein, E. (s.f.). *Distance*. Recuperado el 2 de febrero de 2012, de Mathworld: http://mathworld.wolfram.com/Distance.html.

Apéndice A

Formatos de Control Para el Entrenamiento del Sistema de RAH

MEJORA DEL SISTEMA DE CONTROL DE UNA CALCULADORA ESTÁNDAR USANDO LA PLATAFORMA PXI Y EL ALGORITMO DE COEFICIENTES CEPSTRALES EN ESCALA MEL.

Fecha:
___/___/____
Código:

Formato de Control Para el Entrenamiento del Sistema de RAH (Modelo 1)

Nombre: _____ Apellido: _____
Edad: _____ Sexo: _____ Archivo: _____

Al hablante: Se le pide por favor enuncie las siguiente palabras frente al micrófono una por una cuidando que el programa de grabación no de error, si esto ocurre por favor repetir la palabra. Al hablar recuerde modular claramente cada palabra.

# DE LÍNEA	PALABRA	CHEQUEO
1	CERO	
17	CERO	
33	UNO	
49	UNO	
65	DOS	
81	DOS	
97	TRES	
113	TRES	
129	CUATRO	
145	CUATRO	
161	CINCO	
177	CINCO	
193	SEIS	
209	SEIS	
225	SIETE	
241	SIETE	
257	OCHO	
273	OCHO	
289	NUEVE	
305	NUEVE	
321	PUNTO	
337	PUNTO	
353	IGUAL	

# DE LÍNEA	PALABRA	CHEQUEO
369	IGUAL	
385	ENTRADA	
401	ENTRADA	
417	BORRAR	
433	BORRAR	
449	REINICIAR	
465	REINICIAR	
481	MAS	
497	MAS	
513	MENOS	
529	MENOS	
545	POR	
561	POR	
577	ENTRE	
593	ENTRE	
609	OPUESTO	
625	OPUESTO	
641	RAÍZ	
657	RAÍZ	
673	CUADRADO	
689	CUADRADO	
705	INVERSO	
721	INVERSO	

Observaciones:

_____

MEJORA DEL SISTEMA DE CONTROL DE UNA CALCULADORA ESTÁNDAR USANDO LA PLATAFORMA PXI Y EL ALGORITMO DE COEFICIENTES CEPSTRALES EN ESCALA MEL.

Fecha:
___/___/____
Código:

Formato de Control Para el Entrenamiento del Sistema de RAH (Modelo 2)

Nombre: _____ Apellido: _____
Edad: _____ Sexo: _____ Archivo: _____

Al hablante: Se le pide por favor enuncie las siguiente palabras frente al micrófono una por una cuidando que el programa de grabación no de error, si esto ocurre por favor repetir la palabra. Al hablar recuerde modular claramente cada palabra.

# DE LÍNEA	PALABRA	CHEQUEO
1	CERO	
17	CERO	
33	UNO	
49	UNO	
65	DOS	
81	DOS	
97	TRES	
113	TRES	
129	CUATRO	
145	CUATRO	
161	CINCO	
177	CINCO	
193	SEIS	
209	SEIS	
225	SIETE	
241	SIETE	
257	OCHO	
273	OCHO	
289	NUEVE	
305	NUEVE	
321	PUNTO	
337	PUNTO	
353	IGUAL	

# DE LÍNEA	PALABRA	CHEQUEO
369	IGUAL	
385	ENTRADA	
401	ENTRADA	
417	BORRAR	
433	BORRAR	
449	INICIO	
465	INICIO	
481	SUMA	
497	SUMA	
513	RESTA	
529	RESTA	
545	MULTIPLICA	
561	MULTIPLICA	
577	DIVIDE	
593	DIVIDE	
609	OPUESTO	
625	OPUESTO	
641	RAÍZ	
657	RAÍZ	
673	CUADRADO	
689	CUADRADO	
705	INVERSO	
721	INVERSO	

Observaciones:

_____

Apéndice B

Manual de Usuario de la Calculadora VOXAG

CALCULADORA VOXAG - Vox Agnitio

Manual de usuario

Manual de uso actualizado, orientado al usuario, con instrucciones para la utilización del instrumento virtual CALCULADORA VOXAG desarrollado originalmente por Alberto Rodríguez, y mejorado por Gabriel Guzmán, bajo la tutela de la Ing. Maria Cristi Stefanelli para el trabajo especial de grado **"MEJORA DEL SISTEMA DE CONTROL DE UNA CALCULADORA ESTÁNDAR USANDO LA PLATAFORMA PXI Y EL ALGORITMO DE COEFICIENTES CEPSTRALES EN ESCALA MEL "**.

1. INSTALACIÓN DE MICRÓFONO.

El software de reconocimiento de voz requiere de un transductor que permita al equipo digitalizar la voz del usuario, este puede ser un micrófono común de 600 ohms de impedancia con plug de 3.5mm.

Conecte el micrófono a un conversor de plug 3.5mm hembra a BNC macho. Seguidamente, conecte el extremo BNC a la tarjeta de adquisición NI-5142 de National Instruments en el equipo PXI. Esta conexión puede realizarse a través del canal 0 o canal 1 pero debe tomar en cuenta que la decisión debe coincidir con la realizada en la configuración de la pestaña DAQ del programa.

2. CONFIGURACIÓN DE LA TARJETA DE ADQUISICIÓN.

Para realizar la configuración inicial, diríjase a la pestaña DAQ del programa.

2.a. BLOQUE DE MUESTREO

Seleccione la posición (Slot) en la que se encuentra la tarjeta de digitalización NI-5142 en el PXI. Esta puede ser conocida al leer la etiqueta numerada en la parte inferior del chasis del PXI debajo de tarjeta.

Seguidamente seleccione el Canal a utilizar. Debe ser Canal 0 o Canal 1, según donde haya conectado el micrófono.

Seleccione el "Timeout". Este valor es el tiempo es segundos que la tarjeta de adquisición esperará en inactividad de no escuchar nada antes de desactivar el programa.

Seleccione la frecuencia de muestreo (fs). Bajo la premisa del Teorema de Nyquist para Muestreo, esta debe estar por encima de lo 8KHz para la voz, y por razones técnicas, esta se limita a un máximo de 12kHz para este programa. Tome en cuenta que la frecuencia seleccionada debe ser la misma que la utilizada en el entrenamiento del sistema para obtener un desempeño óptimo.

Seleccione la Cantidad de Puntos a grabar por cada adquisición. La cantidad de puntos a grabar junto con la frecuencia de muestro, determinan la cantidad de tiempo que será grabada y procesada por el programa. t = Cant. de puntos * (1 / fs). Los valores aceptables se encuentran entre 6400 y 24000 puntos, permitiendo de 0.8 a 2 segundos de grabación.

2.b. BLOQUE DE TRIGGER

El Nivel de Trigger sumado a la Histéresis indica el nivel de voltaje necesario para disparar la adquisición de señales. Por defecto se configura a 0 el Nivel de Trigger y 0.001 el nivel de Histéresis para trabajar bajo las condiciones de ruido del laboratorio de Microondas de la Universidad Católica Andrés Bello, pero estos valores pueden variar en otro ambiente. Estos valores están sujetos a cumplir con la ecuación Histéresis - Trigger >= -0.5 para evitar errores de en la tarjeta de adquisición de datos.

Trigger Holdoff indica el tiempo en segundos que espera la tarjeta luego de iniciar una adquisición antes de dispararse nuevamente.

2.c. ERROR OUT

El "Error Out" indica al usuario si ocurrió alguna irregularidad con la tarjeta de adquisición de datos, el código del error, y una breve descripción del mismo.

2.d. FORMA DE ONDA

A la derecha de esta configuración se puede observar una imagen que grafica la forma de onda en el dominio de tiempo de la señal adquirida dado un disparo de la tarjeta de adquisición.

3. CONFIGURACIÓN DEL CODEBOOK

El Codebook es la base de datos que contiene los parámetros de las palabras con las que ha sido entrenado el sistema.

En la pestaña con este nombre es posible seleccionar el directorio en que este se encuentra en un formato de archivo de texto (.txt), bien sea escribiendo la ruta en la "Dirección del Codebook" o seleccionándola al hacer click sobre la carpeta a su derecha.

Adicionalmente, se le debe indicar al programa el tamaño del Codebook seleccionado para que este funcione adecuadamente. Este valor indica la cantidad de personas que entrenaron el sistema y obligatoriamente se encuentra limitado a 30, 40 ó 60 personas.

Finalmente, al correr el programa se presenta aquí una matriz que contiene los centroides almacenados en el codebook seleccionado. Cada fila indica un centroide, cada columna una dimensión, y cada 16 filas representan un codeword.

4. PARAMETROS DE CONTROL

La interfaz de usuario proporciona ciertos indicadores que muestran al usuario valores que le permiten identificar algunos parámetros del proceso de reconocimiento.

La pestaña "Codeword Entrante" contiene una matriz de 16 filas y 20 columnas con los centroides obtenidos de la última palabra procesada por el sistema.

Por otra parte, en la pestaña "Distancias" posee varios indicadores específicos del reconocimiento.

4.a. ARREGLO DISTANCIAS CODEWORD A CODEBOOK

Las distancias Codeword a Codebook representan la separación que existe entre la palabra que se acaba de procesar y todas las palabras que existen en la base de datos utilizada.

4.b. DISTANCIA MÍNIMA E ÍNDICE

La distancia mínima corresponde a aquella palabra de la base de datos que se asemeja más a la última palabra procesada, adicionalmente se indica en qué posición de la base de datos (índice) se encuentra la palabra.

4.c. ARREGLO DE RECONOCIMIENTO

El siguiente arreglo de indicadores booleanos, llamado arreglo de reconocimiento, enciende uno y solo un indicador de entre 23, correspondiente al código de palabra que fue reconocida (Ver guía de comandos).

4.d. PALABRA RECONOCIDA

Este es un indicador de texto que presenta la palabra que ha sido determinada como la más cercana a la última procesada, esperando que esta, realmente sea la misma que ha dicho el usuario.

4.e. NÚMERO DE COEFICIENTES

Para que una palabra sea procesada con éxito, se le deben poder extraer al menos 8 coeficientes. Este indicador presenta cuantos coeficientes se pudo extraer de la vocalización realizada y evita procesar un comando si se extraen menos de 8 coeficientes colocando el indicador en rojo.

La cantidad de coeficientes que el programa extrae de una grabación depende de la cantidad de tiempo que contiene información útil en la misma. Por tanto el mínimo de información útil que debe tener una palabra para ser procesada es de 180 milisegundos.

5. USO DE LA CALCULADORA

La pestaña “Calculadora VOXAG contiene la calculadora a ser operada por el usuario.

5.a. TECLADO / VOZ

El interruptor “Teclado / Voz” permite al usuario seleccionar la manera de controlar la calculadora, esta puede ser a través de comandos de voz, o presionando las teclas valiéndose del mouse del computador dependiendo de la posición de este control.

5.b. INDICADOR DE OPERANDO

En la parte superior de la calculadora se encuentra un indicador que muestra el operador que el usuario está introduciendo o el resultado de una operación.

5.c. LECTURA DE VOZ

El indicador “Lectura de Voz” cumple con la misma función del indicador de error explicado en la sección 4.e de este manual.

5.d. TECLADO

El teclado permite al usuario realizar las operaciones deseadas como son introducir dígitos y punto decimal, realizar operaciones de suma, resta, multiplicación, división, opuesto, raíz, cuadrado, inverso, borrar un dígito, borrar la entrada, borrar todo y mostrar el resultado de la operación. Este teclado puede ser controlado por comandos de voz o pisando las teclas según lo indique el control de la sección 5.a de este manual. Para conocer los comandos de voz utilizados por la calculadora diríjase a la sección 6 de este manual.

Otra forma de conocer el comando que corresponde a una tecla es colocar el mouse encima de la tecla hasta que el comando sea mostrado.

5.e. PALABRA RECONOCIDA

A la derecha de la calculadora se encuentra un indicador llamado "Palabra Reconocida" este indica al usuario que palabra ha reconocido el sistema de reconocimiento automático del habla o si ocurrió un error al procesar el comando.

6. GUÍA DE COMANDOS

CÓDIGO DE PALABRA	COMANDO	TECLA	FUNCIÓN
0	CERO	"0"	Introducir dígito
1	UNO	"1"	-
2	DOS	"2"	-
3	TRES	"3"	-
4	CUATRO	"4"	-
5	CINCO	"5"	-
6	SEIS	"6"	-
7	SIETE	"7"	-
8	OCHO	"8"	-
9	NUEVE	"9"	-
10	PUNTO	"."	Introducir punto decimal
11	IGUAL	"="	Operador igual
12	ENTRADA	"CE"	Borrar entrada
13	BORRAR	"<="	Borrar último dígito
14	REINICIAR	"C"	Borrar todo
15	MAS	"+"	Operador suma
16	MENOS	"-"	Operador resta
17	POR	"*"	Operador multiplicación
18	ENTRE	"/"	Operador división
19	OPUESTO	"-x"	Operador signo opuesto
20	RAIZ	"sqrt"	Operador raíz
21	CUADRADO	"x^2"	Operador elevar al cuadrado
22	INVERSO	"1/x"	Operador inverso

Apéndice C

Resultados de pruebas al cambiar el coeficiente del filtro de preénfasis

El coeficiente del filtro de preénfasis se cambió a -0,95 para realizar estas pruebas, ya que se planteó durante el curso de la presentación final del trabajo que el comportamiento del filtro podía estar afectando el reconocimiento de las palabras.

En las pruebas de 16 centroides con palabras largas y cortas, se observa en las tablas que disminuye la eficiencia para las palabras largas, pero a su vez la caída con las palabras cortas no es tan brusca. De igual manera, los resultados se mantienen bastante parejos sin importar el género del locutor, lo que parece indicar que este era el principal problema que había con las voces femeninas.

	Entrada	Multiplica	Divide	Opuesto	Cuadrado
Entrada	10	5	4	1	0
Multiplica	2	15	1	1	1
Divide	1	8	10	1	0
Opuesto	5	1	1	13	0
Cuadrado	3	0	2	0	15
% Acierto	50	75	50	65	75

% TOTAL	63

Tabla 14 Prueba con 5 palabras largas, femenino, filtro (-0.95)

Fuente: Elaboración Propia.

	Uno	Dos	Seis	Ocho	Mas
Uno	3	0	11	0	6
Dos	1	4	1	0	14
Seis	1	0	18	0	1
Ocho	0	3	3	14	0
Mas	0	6	1	0	13
% Acierto	15	20	90	70	65

% TOTAL	52

Tabla 15 Prueba con 5 palabras cortas, femenino, filtro (-0.95)

Fuente: Elaboración Propia.

	Entrada	Multiplica	Divide	Opuesto	Cuadrado
Entrada	13	0	0	4	3
Multiplica	2	14	4	0	0
Divide	0	2	17	0	1
Opuesto	1	1	3	14	1
Cuadrado	13	0	0	1	6
% Acierto	65	70	85	70	30

% TOTAL	64

Tabla 16 Prueba con 5 palabras largas, masculino, filtro (-0.95)

Fuente: Elaboración Propia.

	Uno	Dos	Seis	Ocho	Mas
Uno	15	1	1	1	2
Dos	0	9	1	9	1
Seis	0	0	20	0	0
Ocho	2	3	1	7	7
Mas	0	5	3	11	1
% Acierto	75	45	100	35	5

% TOTAL	52

Tabla 17 Prueba con 5 palabras cortas, masculino, filtro (-0.95)

Fuente: Elaboración Propia.

Con respecto a las pruebas del sistema con 8 centroides original, después de la vocalización de las primeras seis palabras (20 x 6 = 120 vocalizaciones) se dejó de intentar, ya que la calculadora no acertó la palabra ni una sola vez, para ninguno de los dos géneros.

Anexo A

Rango de variación de la frecuencia fundamental de la voz en adultos

Aquí se muestran parte de los resultados del estudio "Análisis Acústico de la Voz en Adultos. Patrones de Normalidad" realizado en el Hospital Central Universitario "Antonio María Pineda" de Barquisimeto, por la Dra. Elsa Álvarez.

Edad		*Fo (vocal a)*		*Fo (Vocal i)*	
(años)	*n*	*Prom.*	*DE*	*Prom.*	*DE*
20 – 29	12	162.64	29.51	205.11	55.03
30 – 39	18	177.05	39.31	202.87	37.77
40 – 49	10	159.74	34.92	181.93	30.82
50 - 59	1	165.53	0.00	172.57	0.00

Frecuencia fundamental de la voz de sexo masculino.
Fuente: (Álvarez, 2000)

Edad		*Fo (vocal a)*		*Fo (Vocal i)*	
(años)	*n*	*Prom.*	*DE*	*Prom.*	*DE*
20 – 29	17	286.48	35.33	303.47	46.91
30 – 39	26	246.71	35.48	287.06	30.70
40 – 49	6	289.78	21.70	301.85	33.95
50 - 59	1	279.67	0.00	285.86	0.00

Frecuencia fundamental de la voz de sexo femenino.
Fuente: (Álvarez, 2000)