

UNIVERSIDAT OBERTA DE CATALUNYA

Estudios de Informática, Multimedia y Telecomunicación



GRADO DE TECNOLOGÍAS DE TELECOMUNICACIÓN

***MÉTODO DE SEGURIDAD EN
UNA CARCEL MEDIANTE UN
SISTEMA DE RECONOCIMIENTO
DE VOZ EN MATLAB***

Realizado por: Silvia Herrero Torresano

Consultor: Lourdes Meler

Fecha de entrega: 10 de enero de 2018



Esta obra está sujeta a una licencia de
Reconocimiento-NoComercial-
SinObraDerivada [3.0 España de Creative
Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	Método de seguridad en una cárcel mediante un sistema de reconocimiento de voz en Matlab
Nombre del autor:	Silvia Herrero Torresano
Nombre del consultor:	Lourdes Meler
Fecha de entrega (mm/aaaa):	01/2018
Área del Trabajo Final:	Aplicaciones multimedia basadas en procesamiento de la señal
Titulación:	<i>Grado de Tecnologías de la Telecomunicación</i>

Resumen del Trabajo (máximo 250 palabras):

El trabajo que se muestra a continuación trata sobre la creación de una aplicación de seguridad la cual tiene como fin el reconocimiento de voz. Para desarrollarlo, lo primero que se ha de hacer, es guardar los datos de voz de diferentes individuos en una base de datos, mediante una grabación de audio con el micrófono del portátil, para posteriormente procesarlos y analizarlos con el fin de sacar los parámetros que caracterizan la señal y con ellos poder identificar a una persona en futuros accesos.

Para la realización de este proyecto se deberán explicar las bases de estudio para lograr entender como se llega a este objetivo. Las bases en las que se guía el proceso son los sistemas biométricos, las que a la vez son los encargados de analizar características físicas o de comportamientos de las personas y con ello identificarlas. La biometría se basa en un patrón o varios, que son únicos en los sujetos como puede ser el habla, que es en lo que se centrará este trabajo.

Todo esto tendrá como objetivo principal, el poder usar la aplicación en un futuro como método de seguridad, como por ejemplo en una cárcel, mediante la identificación y autenticación por voz. Los guardias podrán abrir las puertas de seguridad, sin necesidad de aprender claves ni llevar tarjetas identificativas que puedan ser robadas. Sería tan sencillo como registrar las palabras abre o cierra,

con ello se conseguiría manejar todas las puertas de un mismo sector. Este modelo se puede aplicar a otros lugares como empresas, organismos estatales..., en todos lo que se requieran autenticación e identificación.

Para realizar este proyecto se va a utilizar el programa MATLAB. Este programa sirve para implementar el código, la base de datos y la interfaz gráfica.

Se ha elegido este método de identificación por voz, ya que es uno de los más comerciales y en cuanto a costes, es menor que los demás.

Abstract (in English, 250 words or less):

The work that follows is about the creation of a security application which is used for voice recognition. To do this, what will be done is to save the voice data of different individuals in a database, by means of an audio recording with the microphone of the laptop, to later process and analyze them in order to extract the parameters that characterize the signal and with them be able to identify a person in future accesses.

In order to carry out this project, the first thing that will be done is to explain the bases of study to understand how this objective is achieved. These bases on which the process is guided are the biometric systems, which are responsible for analyzing physical characteristics or behaviors of people and with this to identify them. Biometrics focuses on a pattern or several that are unique in subjects such as speech that is the focus of this work.

All this will have as main objective the use of the application in a future as a method of security in a jail, through the identification and authentication by voice. The guards will be able to open the security doors, without having to learn keys or carry identification cards that can be stolen. It is as simple as registering your name and with a simple open or close you can handle all the doors of the enclosure. It should be noted that this model can be applied to other places such as companies, state agencies ... everything that requires authentication and identification.

The MATLAB program will be used to carry out the project. This program can be

used to implement the code, the database and the graphical interface. .

The method of voice identification has been chosen, since it is one of the most commercial and in terms of costs is lower than the others.

Palabras clave (entre 4 y 8):

voz, MATLAB, identificación, autenticación, sistemas biométricos, procesamiento.

Voice, MATLAB, identification, authentication, biometric system, processing.

INDICE

FICHA DEL TRABAJO FINAL	iii
INDICE	vi
INDICE DE TABLAS	viii
INDICE DE IMÁGENES	ix
1. INTRODUCCION	10
1.1 PRINCIPALES OBJETIVOS	11
1.2 BENEFICIOS	12
1.3 MOTIVACIÓN	13
2. ESTADO DEL ARTE	14
2.1 ¿QUE ES LA BIOMETRIA?	14
2.2 SISTEMA BIOMETRICO	15
2.2.1 FUNCIONAMIENTO DE UN SISTEMA BIOMETRICO	15
2.3 FUNDAMENTOS DE LA VOZ	16
2.4 RECONOCIMIENTO DE VOZ	17
2.4.1 RECOGIDA DE MUESTRAS CON EL MICROFONO. CREACIÓN DE LA BASE DE DATOS	18
2.4.2 PREPROCESAMIENTO	20
2.4.3 EXTRACCION DE CARACTERISTICAS: PARAMETRIZACIÓN.....	22
2.4.4 MEDIDA DE LA DISTANCIA. DECISIÓN.....	24
3. DISEÑO E IMPLEMENTACIÓN	26
3.1 METODOLOGÍA	26
3.2 DIAGRAMA DE BLOQUES DEL SISTEMA.....	27
3.3 MICRÓFONO	28
3.4 ENTRENAMIENTO Y BASE DE DATOS.....	28
3.4.1 ADQUISICIÓN.....	28
3.4.2 ELIMINACIÓN DE RUIDO	29
3.4.3 FILTRO	31
3.4.4 SEGMENTACIÓN	31
3.4.5 COEFICIENTES CARACTERISTICOS. PARAMETRIZACIÓN	32
3.4.6 BASE DE DATOS	33
3.5 RECONOCIMIENTO DEL HABLA.....	33
3.5.1 DECISIÓN.....	33
3.5.2 INTERFAZ GRÁFICA.....	34
3.6. PRUEBAS Y RESULTADOS	38

3.6.1 MATRIZ DE CONFUSIÓN	38
3.6.2 PRUEBAS CON PERSONAS Y PALABRAS NO IDENTIFICADAS.	43
3.6.3 RECOMENDACIONES	46
4. CONCLUSIONES.....	48
5. BIBLIOGRAFÍA	50
6. ANEXO.....	55
7. GLOSARIO.....	76

INDICE DE TABLAS

Tabla 1: Matriz de confusión de palabra apertura junto a su índice de acierto en tanto por ciento	39
Tabla 2: Matriz de confusión de palabra cierre junto a su índice de acierto en tanto por ciento	41
Tabla 3: Matriz de confusión de palabras aleatorias junto a su índice de acierto en tanto por ciento	43
Tabla 4: Matriz de confusión de la palabra apertura pronunciada por intrusos	45
Tabla 5: Matriz de confusión de la palabra cierre pronunciada por intrusos	45

INDICE DE IMÁGENES

Imagen 1: Fases de un sistema biométrico	16
Imagen 2: Aparato fonatorio	17
Imagen 3: Diagrama de bloques del sistema	21
Imagen 4: Gráfica en el dominio frecuencial del filtro pre-énfasis	21
Imagen 5: Ventana de Hamming	22
Imagen 6: Modelo de coeficientes cepstrales	23
Imagen 7: Diagrama esquemático del sistema de reconocimiento de voz	27
Imagen 8: Configuración del micrófono	28
Imagen 9: Resultado de la señal al eliminar ruido	30
Imagen 10: Diseño de la interfaz gráfica imagen de GUI	34
Imagen 11: Interfaz gráfica de Reconocedor de Voz creado en Matlab después de la ejecución	35
Imagen 12: Interfaz gráfica después de pulsar el botón grabar palabra	36
Imagen 13: Interfaz gráfica después de pulsar el botón identificar palabra	37
Imagen 14: Ventana que aparece después de presionar el botón salir	37
Imagen 15: Interfaz gráfica de reconocimiento de la palabra apertura por el usuario 1	39
Imagen 16: Interfaz gráfica de reconocimiento de la palabra apertura por el usuario 2	40
Imagen 17: Interfaz gráfica de reconocimiento de la palabra cierre por el usuario 1... ..	41
Imagen 18: Interfaz gráfica de reconocimiento de la palabra cierre por el usuario 2	42
Imagen 19: Interfaz gráfica de reconocimiento de palabra aleatoria	44
Imagen 20: Interfaz gráfica de reconocimiento de palabra pronunciada por un intruso	46

1. INTRODUCCION

A lo largo de los años las tecnologías han evolucionado a pasos agigantados, y cada día la sociedad está más conectada con los avances tecnológicos. Un área que está emergiendo, es la biometría, que es la capacidad para establecer la identidad de cualquier individuo.

Estos sistemas de identificación personal siempre estaban basados en identificaciones como el carnet de identidad, llaves, tarjetas o con conocimientos de claves secretas. Si se tiene en cuenta que cada persona tiene características y rasgos que los hacen únicos y que los distinguen de los demás, por qué no usar esto para el reconocimiento de las mismas de forma automática.

Hace unos años esto mismo lo aplicaron los científicos y crearon lo que se conoce como sistemas biométricos. Un sistema biométrico consiste en reconocer a un individuo usando una de sus características biológicas, psicológicas o de conducta.

Para que un sistema biométrico pueda funcionar se deben comparar los registros biométricos con unas muestras almacenadas previamente en una base de datos.

Hay muchas características que pueden identificar a una persona como pueden ser las huellas dactilares, la retina, la forma facial...pero este trabajo se va a centrar en el reconocimiento de voz, que es uno de los más utilizados.

1.1 PRINCIPALES OBJETIVOS

El objetivo del presente proyecto es implementar un método de seguridad para la apertura y cierre de las puertas de una cárcel. Esta aplicación hará posible que el personal de seguridad de un centro penitenciario pueda acceder a zonas de uso restringido con la mayor seguridad posible.

Para lograr este objetivo, lo primero que se ha de realizar es la creación de una base de datos, con los parámetros biométricos que se deban de utilizar. Esto serviría para los reconocimientos biométricos, que utilizarán estos registros de la base de datos con el fin de reconocer a las personas. En un futuro esa base de datos se podrá ampliar para registrar otros datos biométricos, como pueden ser las huellas dactilares, pero en un principio este proyecto solo se centrará en el uso de la voz.

Hay tres partes diferenciadas en el proyecto: la encargada de recoger muestras y almacenamiento en la BBDD, la extracción de los parámetros característicos de esas señales mediante el procesamiento de voz y por último la creación de una interfaz que simule lo anterior.

Con ello, para que los individuos que ya estén registrados puedan acceder a un área restringida, solo tendrán que pronunciar en la aplicación a través del micrófono la acción que pretendan realizar: abrir o cerrar la puerta y el sistema será capaz de reconocer a esa persona automáticamente y con ello darle acceso. Para conseguir este paso, previamente cada usuario habrá grabado su nombre completo y las palabras clave en la aplicación, que serán los registros almacenados en la base de datos.

Toda la implementación será realizada con MATLAB [28] ya que es una potente herramienta para el procesado de señales.

1.2 BENEFICIOS

El reconocimiento de voz, hoy en día, es utilizado en la vida cotidiana para muchos fines, como poder desbloquear un móvil por voz hasta para acceder a un banco a través de una palabra clave.

Pero si se analizan los beneficios, está probado que esta tecnología puede hacer la vida más fácil en distintas situaciones. A continuación, son explicados algunos ejemplos:

FIABILIDAD: Los sistemas de reconocimiento de voz son tan fiables, que a día de hoy se están utilizando en muchos organismos como sanidad, sistema judicial...no solo para accesos, sino para dictar las notas.

SEGURIDAD: Por ser un sistema que extrae características únicas de un individuo, permitirá mayor seguridad a la hora de utilizarlo.

COSTES: Tiene un coste bajo ya que el software no es complejo ni los recursos que se necesitan.

CAMBIO: Si se habla de un sistema de seguridad que no requiere tarjetas individuales intransferibles, a la hora de cambiar este método es muy cómodo porque con borrar al individuo y crear a otro no se tiene la necesidad de cambiar otras cosas.

TIEMPO: Está comprobado que produce un ahorro de tiempo, un ejemplo sería que al dictar las notas y escribirlas simultáneamente, se realizaría en poco tiempo y por eso sería muy útil.

PRODUCTIVIDAD: Permite producir una gran cantidad de información en menos tiempo.

ERRORES: Si es la máquina la que escribe mediante dictado, se evitarán los errores ortográficos

MOVILIDAD: Si se puede realizar un trabajo mediante el uso del dictado esto permitirá una mayor movilidad.

En este proyecto será muy importante como método de seguridad, ya que nadie podrá abrir ni cerrar puertas, si su voz no está registrada. Otro beneficio fundamental será el ahorro de costes, no habrá necesidad de crear tarjetas ni llaves, y si algún guardia

cambia de zona será tan fácil como quitar el acceso que tiene actualmente y registrarse en la nueva, esto ahorra tiempo y dinero.

1.3 MOTIVACIÓN

Este proyecto es la práctica de fin de Grado de Tecnologías de Telecomunicación y en el que se preponde, poner en uso los conocimientos adquiridos a lo largo de la carrera.

La decisión de realizar este proyecto es acorde a la mención que he realizado: Ingeniería de Sistemas Audiovisuales. Siempre me ha parecido interesante todo lo relacionado con el procesamiento de señal ya sea imagen, sonido...

Inclinarme por este en concreto ha sido motivado porque creo que hoy en día es muy importante el tema de la seguridad, y cuanto más efectivo y menos costoso sea mejor, y usar reconocimiento de voz es algo que reúne todas estas características, ya que realizar uno de la retina o de la huella dactilar es más complejo.

El proyecto que pretendo realizar quiero que sea accesible a todas las personas, que resulte ameno y que cualquiera sin unos conocimientos altos de programación pueda entender. Por ello, el código utilizado será MATLAB con una interfaz simple e intuitiva a simple vista.

2. ESTADO DEL ARTE

El habla es una de las características más importantes de la expresión humana, es algo que diferencia a los seres humanos del resto de seres vivos del planeta.

La biometría consta de muchas aplicaciones, pero el trabajo se va a centrar en la identificación de personas mediante el uso de la voz. Esto es algo mucho más fiable y seguro que por ejemplo la identificación por tarjetas o contraseñas.

Para realizar un sistema biométrico de reconocimiento de voz, lo primero que se tiene que hacer es un procesamiento digital de la señal de voz [33], que obtendrá los rasgos característicos de la voz para compararlos con los que guarda su base de datos, y con ello identificar a un individuo.

2.1 ¿QUE ES LA BIOMETRIA?

La biometría [5], es la ciencia que estudia la identificación de individuos a partir de sus rasgos anatómicos característicos o sus rasgos de comportamiento. Las características anatómicas no suelen variar con el tiempo como puede ser la huella dactilar o la palma de la mano. Pero sin embargo los rasgos de comportamiento pueden ser menos estables. No todas las características anatómicas sirven para ser un sistema biométrico, debe de cumplir unas reglas:

- Universalidad: toda persona tiene que tener dicha característica.
- Unicidad: dos personas deben ser diferenciadas por dicha característica.
- Permanencia: la característica debe ser más o menos estable.
- Cuantificación: La característica se puede medir.

Como características biométricas tenemos: la cara, el ADN, la voz, la huella dactilar, firma [41] ...

2.2 SISTEMA BIOMETRICO

Un sistema biométrico es un sistema de identificación automatizado, que sirve para sustituir los métodos actuales que utilizan las contraseñas y tarjetas para la autenticación de usuarios [7]. Para que dicha identificación se lleve a cabo se usan los rasgos biométricos de una persona.

Para que el sistema biométrico se considere de buena calidad debe reunir las siguientes condiciones:

- Funcionamiento: Debe ser rápido y efectivo.
- Aceptabilidad: No debe causar daño ninguno por el uso diario.
- Seguridad: El sistema debe poseer robustez.

La ventaja que tiene respecto a los demás métodos es que logra identificar a una persona por un rasgo o patrón único, que difícilmente pueden ser copiados como por ejemplo la voz que es en el que se centrará este proyecto.

Por lo tanto, un sistema biométrico es sencillamente un método de reconocimiento de patrones que utiliza ciertos rasgos identificativos de un usuario.

2.2.1 FUNCIONAMIENTO DE UN SISTEMA BIOMETRICO

Un sistema biométrico consta de tres partes principales. Por un lado, disponen de un mecanismo automático, que lee y captura una imagen digital o analógica de una característica concreta a analizar. También posee una parte que le sirve para manejar los aspectos de compresión, comparación y almacenamiento. Y, por último, ofrece una interfaz para las aplicaciones que luego lo utilizan.

El proceso de autenticación sigue unos pasos:

- Captura y lectura de los datos.
- Extracción de las características de muestra que sean necesarias.
- Comparación de las características con la base de datos.
- Toma de decisión de si un usuario es válido o no.

En la decisión es donde se puede observar si un sistema biométrico es fiable o no lo es, pueden suceder dos casos [3]:

- FRR (False Rejection Rate): Tasa de falso rechazo, que es la probabilidad de que un sistema de autenticación rechace a un usuario que es correcto, ya que no es capaz de identificarlo.
- FAR (False Acceptance Rate): Tasa de falsa aceptación, que es la probabilidad de que un sistema autentifique a un usuario ilegítimo.

En supuesto de que exista una FRR elevada provocará un descontento entre los usuarios que utilicen el sistema porque ya no será nada preciso, pero una FAR elevada provoca un fallo de seguridad muy elevado, se le está dando acceso a zonas de uso restringido a usuarios no autorizados. Cualquier sistema biométrico que se precie intentará minimizar estas probabilidades lo máximo posible.

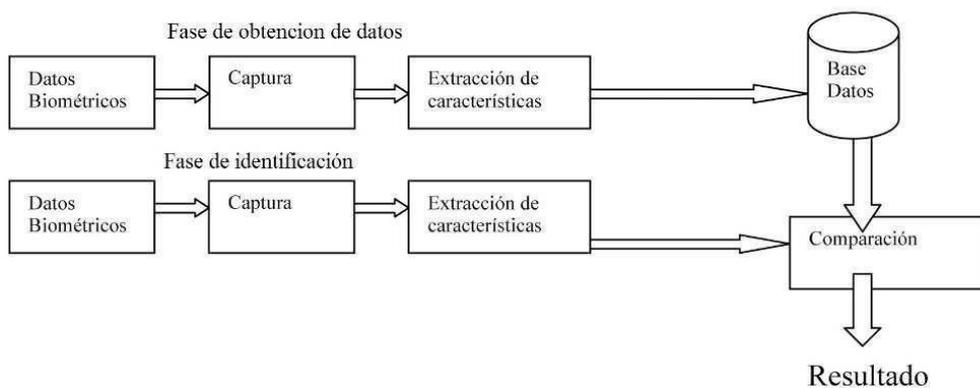


Imagen 1: Fases de un sistema biométrico

2.3 FUNDAMENTOS DE LA VOZ

La voz humana se produce por medio del aparato fonatorio. Está formado por los pulmones como fuente de energía, en forma de flujo de aire, la laringe que contiene las cuerdas vocales, la faringe, las cavidades oral y nasal y una serie de elementos articulatorios [9].

En el proceso de generación de la voz, el sonido inicial proviene de la vibración de las cuerdas vocales conocida como vibración glotal, es decir, el efecto sonoro se genera por la rápida apertura y cierre de las cuerdas vocales conjuntamente con el flujo de aire que sale desde los pulmones.

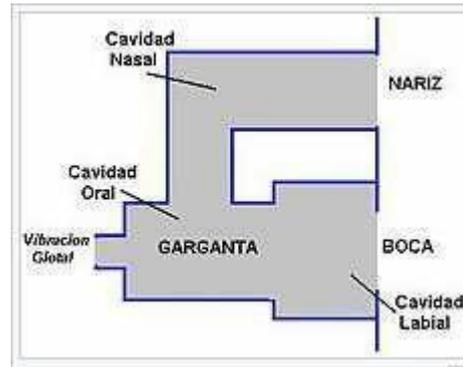


Imagen 2: Aparato fonatorio

La señal de voz está constituida por un conjunto de sonidos generados por el aparato fonador. Dicha señal acústica puede ser transformada con un micrófono en una señal eléctrica. La señal de voz en el tiempo puede ser representada en un par de ejes cartesianos. Como todos los sonidos está formado sencillamente por curvas elementales (senos y cosenos) pero las posibles combinaciones de éstas pueden ser complejas.

Habitualmente los dispositivos de reconocimiento de la voz o de la palabra tratan de identificar fonemas o palabras dentro de un repertorio o vocabulario muy limitado. Un fonema es un sonido simple o unidad del lenguaje hablado.

Las ecuaciones fundamentales que se emplean en acústica para el modelado de la voz son lineales. Y eso resulta cómodo porque el uso de sistemas no lineales resultaría más complejo.

2.4 RECONOCIMIENTO DE VOZ

Para que un sistema de reconocimiento de voz funcione se deben de realizar varios pasos que sirvan para procesar la señal de voz de entrada en el sistema. De este modo se extraen las características de la voz que resulten útiles para la tarea que se va a realizar.

2.4.1 RECOGIDA DE MUESTRAS CON EL MICROFONO. CREACIÓN DE LA BASE DE DATOS

Un micrófono es un transductor electroacústico, que transforma la presión acústica en energía eléctrica [29]. La señal de voz puede ser grabada por medio de un micrófono, el cual hace la conversión de la señal analógica en una señal digital por medio del conversor A/D en la tarjeta de sonido.

Cuando un micrófono está operando, las ondas de sonido hacen que vibre el elemento magnético del micrófono causando una corriente eléctrica hacia la tarjeta de sonido, donde el conversor A/D básicamente graba los voltajes eléctricos en intervalos específicos.

Se tienen dos factores muy importantes durante este proceso:

- Tasa de muestreo o que tan seguido los valores de voltaje son grabados.
- Bits por segundo o que tan exactamente los valores son grabados.

Se podría considerar un tercero como el número de canales mono o estéreo, pero al ser una aplicación que utiliza la voz no es necesario un canal estéreo.

Para obtener las muestras de voz a través del micrófono del ordenador y ser introducidas en la base de datos en MATLAB se cuenta con la función wavrecord (n, Fs), que es la encargada de grabar n muestras de la señal de audio, muestreadas a una frecuencia Fs. De momento la base de datos se va a crear con una serie de instrucciones básicas, que serían los comandos de voz abrir o cerrar para cada uno de los usuarios. En un futuro esta base de datos podrá ser ampliada para poder registrar muchos más usuarios.

A partir de la señal analógica obtenida se hace necesario convertir la señal a formato digital para que pueda ser procesada en la computadora y esto se realiza mediante dos procesos: muestreo y cuantificación [30]. Este proceso se compone de dos etapas y se conoce como Modulación por Código de Pulsos (PCM).

- Muestreo: consiste en el proceso de conversión de señales continuas a señales discretas en el tiempo. Este proceso se realiza midiendo la señal en momentos periódicos del tiempo. Si se aumenta el número de muestras por unidad de tiempo, la señal muestreada se parecerá más a la señal continua. El número de muestras por segundo se conoce en inglés como el *bit-rate*.

Si el *bit-rate* es lo suficientemente alto, la señal muestreada contendrá la misma información que la señal original. Con el criterio de Nyquist se asegura que para que la señal muestreada contenga la misma información que la continua, debe suceder que la frecuencia de muestreo sea mayor o igual que $2W$ donde W es el ancho de banda.

Teorema de Nyquist

Sea $x(t)$ una señal en banda limitada con $X(j\omega)=0$ para $|\omega|>w_M$ [32]. Entonces $X(t)$ se determina unívocamente mediante sus muestras $x(n)$, $n=+1,+2\dots$ si:

$$w_s > 2w_M \text{ donde } w_s = \frac{2\pi}{T}$$

Si no se cumple este teorema podría producir aliasing.

· **Cuantificación:** es la conversión de una señal discreta en el tiempo evaluada de forma continua a una señal discreta en el tiempo discretamente evaluada. El valor de cada muestra de la señal se representa como un valor elegido de entre un conjunto finito de posibles valores.

Se conoce como error de cuantificación (o *ruido*), a la diferencia entre la señal de entrada (sin cuantificar) y la señal de salida (ya cuantificada), interesa que el ruido o error de cuantificación sea lo más bajo posible.

En el caso de este proyecto se ha usado 44100 Hz de frecuencia de muestreo, y una resolución de 8 bits/ muestra por lo tanto tenemos 44100 muestras /seg * 8 bits = 352800 bits/seg para un canal mono, y en un canal estéreo se duplicaría. Para el reconecedor de voz implementado es más que suficiente para obtener una buena calidad.

2.4.2 PREPROCESAMIENTO

Es una técnica que permite extraer la información acústica que caracteriza la señal a partir de la señal de voz emitida por el usuario.

Eliminación de ruido

Una vez que la señal está en formato digital, es necesario indicar dónde está el principio y el fin de la señal o lo que es lo mismo diferenciar las partes de la señal de voz que llevan información de voz y las que no. Para ello se debe de calcular la energía promedio de la señal. Se escogen segmentos de 10 ms, si en un segmento la energía promedio es menor que un valor umbral proporcional a la energía promedio de la señal entera es descartado.

$$E_p = \sum_{k=1}^{N_n} |x[k]|^2 w[n-k]$$

E_p es la energía promedio de cada segmento.

$$E_{pe} = \frac{1}{N} \sum_{k=1}^N |x[k]|^2$$

E_{pe} es la energía promedio de la señal entera.

Filtro pre-énfasis

La etapa de pre-énfasis [6], se realiza con el propósito de suavizar el espectro y reducir las inestabilidades de cálculo asociadas con las operaciones aritméticas de precisión finita. Además, se usa para compensar la caída de -6 dB que experimenta la señal al pasar a través del tracto vocal. Se usa un filtro digital de primer orden cuya función de transferencia es:

$$H(z) = \frac{Y(z)}{X(z)} = (1 - az^{-1})$$

$$a = 0.95$$

Y la ecuación de diferencias es la siguiente:

$$y[n] = z[n] - ax[n-1]$$

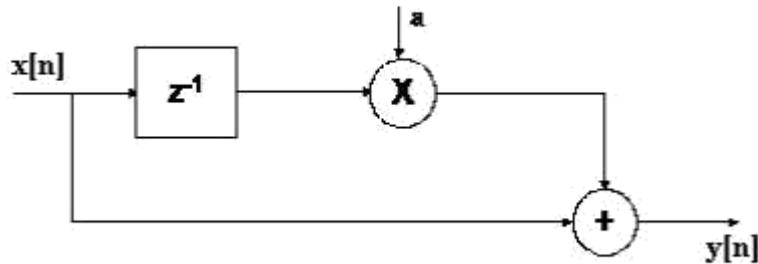


Imagen 3: Diagrama de bloques del sistema

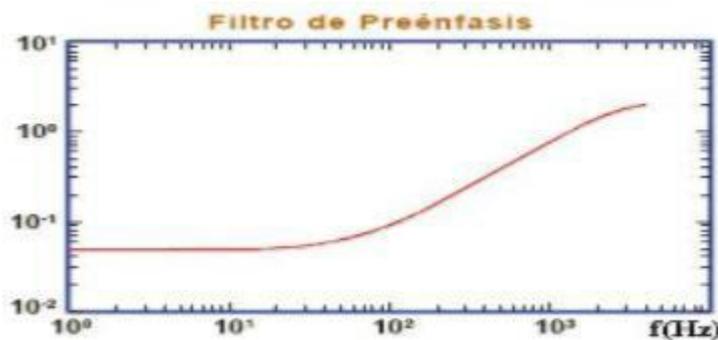


Imagen 4: Gráfica en el dominio frecuencial del filtro pre-énfasis

Segmentación

Es un proceso con el cual se corta la señal de voz en segmentos de análisis, así se evitaría coger señal que no lleve información útil y se hace menos gasto de memoria. Se asume que la señal es estacionaria en intervalos de tiempo lo suficientemente cortos.

Cuando se realiza la segmentación se guardan los datos en una matriz columna para luego ser procesados

El intervalo de tiempo en el que la señal se considera estacionaria depende de la velocidad de cambios del tracto vocal y las cuerdas vocales. Se va a establecer un valor de 30 ms.

Luego se aplica una ventana Hamming [42] a cada uno de los segmentos creados anteriormente. Esto se aplica por el motivo de tener una señal con los lóbulos secundarios pequeños y un lóbulo principal estrecho, a simple vista parece que se perderá la información correspondiente a los lóbulos secundarios, pero esto no es así debido al solapamiento entre segmentos.

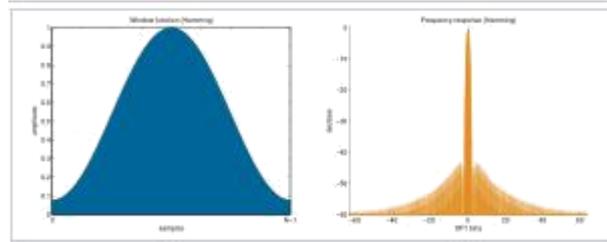


Imagen 5: Ventana de Hamming

$$v(n) = a_0 - a_1 \cos\left(\frac{2\pi n}{N}\right)$$

$$a_0 = 0,53836$$

$$a_1 = 0,46164$$

N indica el largo de cada segmento de análisis.

Se elegirá un desplazamiento de 10ms para no perder información útil, y además es el desplazamiento típico entre cada ventaneo.

Los valores que harán falta para la implementación será $N=240$ muestras que equivales a 30 ms y $M=80$ muestras que equivale a 10 ms.

2.4.3 EXTRACCION DE CARACTERISTICAS: PARAMETRIZACIÓN

En el reconocimiento de la voz, cuando una señal de voz es pre-procesada, pasa por un nuevo procesamiento y produce una nueva representación en forma de secuencias de vectores o agrupaciones de valores, estos se denominan parámetros.

El número de parámetros tiene que ser reducido para que la base de datos no se llene, ya que mientras más parámetros tenga la representación menos fiable son los resultados y es más costosa la implementación.

Existen distintos métodos de análisis para la parametrización, los más destacados son:

- Predicción lineal [12]: Es una de las técnicas más potentes de análisis de voz, y uno de los métodos más útiles para codificar voz con buena calidad. Su función es representar la envolvente espectral de una señal digital de voz en una forma comprimida, utilizando la información de un modelo lineal, con lo cual se proporcionan unas aproximaciones a los parámetros de la voz muy precisas. Se fundamenta en establecer un modelo de filtro de tipo

todo polo, para la fuente de sonido. La principal motivación del modelo todo polo viene dada porque permite describir la función de transferencia de un tubo que, sin pérdidas, está formado por diferentes secciones. El modelo recibe este nombre porque pretende extrapolar el valor de la siguiente muestra de voz $s(n)$ como la suma ponderada de muestras pasadas $s(n-1)$, $s(n-2)$, ..., $s(n-K)$.

$$\tilde{s}(n) = \sum_{k=1}^p \alpha_k \cdot s(n-k)$$

- Cepstrum: Como se sabe los sonidos de la voz se pueden representar mediante un espectrograma, que indica las componentes frecuenciales de la señal de voz. Es así entonces como el espectro proporciona información acerca de los parámetros del modelo de producción de voz, tanto de la excitación como del filtro que representa el tracto vocal.

El modelo de obtención de los coeficientes cepstrales en un sistema como MATLAB es utilizando la FFT (la transformada rápida de Fourier que es un eficiente algoritmo que permite calcular la transformada discreta de Fourier y su inversa dados vectores de longitud N) [15]. La función utilizada es la *rceps*, que proporciona el cepstrum real de la función ingresada.

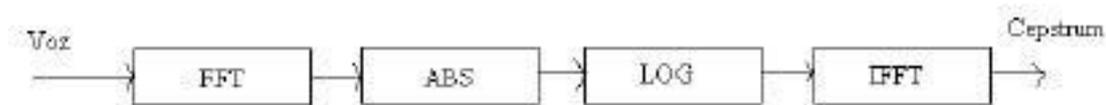


Imagen 6: Modelo de coeficientes cepstrales

En la salida del sistema se tiene:

$$c_p[n] = \frac{1}{N} \sum_{k=0}^{N-1} \log |S_p[p]| e^{j \frac{2\pi}{N} kn}$$

$c_p[n]$ son los coeficientes cepstrales y N es el número de puntos con el que se calcula la transformada de Fourier. Lo normal es utilizar solo los primeros términos $n < 20$

Para mejorar el sistema y hacerlo menos vulnerable al ruido y variabilidad en la grabación se ha utilizado una técnica llamada normalización cepstral. Los coeficientes son expandidos por medio de una representación polinomial ortogonal para poder realizar la suma de ellos. Sólo se necesitan los dos primeros coeficientes ya que esto significa que dos puntos adyacentes en el intervalo no pueden tener una convergencia muy diferente a los demás [39].

Para implementar el algoritmo de la función extracción de características se utilizan las siguientes ecuaciones:

$$P_{0i} = 1$$

$$P_{0i} = i - 5$$

Los dos primeros coeficientes son los siguientes:

$$a = \frac{\sum_i^9 x_i}{9}$$

$$b = \frac{\sum_i^9 x_i P_{0i}}{\sum_i^9 P_{0i}^2}$$

a y b representan el promedio de la función de tiempo de cada coeficiente cepstral.

Los coeficientes cepstrales son representados como $x_t(j)$ y los coeficientes polinomiales son $b_t(j)$, donde t es el número de segmento y j es el índice de los coeficientes cepstrales.

2.4.4 MEDIDA DE LA DISTANCIA. DECISIÓN

Algo fundamental en un sistema de reconocimiento por voz, es la forma en que se comparan los vectores característicos con los patrones de referencia. Para realizar esta operación se debe de definir una medida de distancia entre estos vectores característicos [8].

En MATLAB se utiliza la distancia Euclídea, inducida por la norma de espacios L_p :

$$d = \sqrt{\sum_{i=1}^D |f_i - f'_i|^2}$$

Donde D son las componentes de los vectores f y f' .

Primero se define el tamaño del mayor vector, y luego se calcula mediante la fórmula la distancia entre el vector de la palabra a reconocer y cada uno de los vectores que están en la base de datos. Por último, se realizarán unas comprobaciones para obtener la menor distancia entre las dos palabras que será la que indique que la palabra se encuentra identificada en la base de datos.

3. DISEÑO E IMPLEMENTACIÓN

3.1 METODOLOGÍA

Dentro de la metodología del proyecto se requiere:

- Generar una base de datos
- Filtrar, eliminar ruido y segmentar las señales de voz.
- Obtener los coeficientes cepstrales, normalizarlos y obtener los coeficientes característicos.
- Comparación de los coeficientes característicos para obtener las distancias.
- Abrir, cerrar o denegar acceso.

El proyecto se ha realizado con el registro de voces de diez usuarios, los cuales han grabado dos señales de voz, una para abrir y otra para cerrar, que se ha guardado en la base de datos creada en Matlab. La base de datos no tiene límite de usuarios, por lo que si en un futuro se requieren más se pueden añadir.

Posteriormente estas muestras son pasadas por una serie de procesos hasta obtener los coeficientes característicos que son los necesarios para obtener la distancia.

Por último, se realiza una comparación entre estos coeficientes característicos de las señales registradas en la base de datos y la señal obtenida por el micrófono a reconocer, a menor distancia entre palabras tenemos mayor probabilidad de parecido con la palabra buscada en la base de datos. Este procedimiento se realiza con todas las muestras que se tienen en la base de datos. Con las pruebas realizadas estos parámetros se van ajustando para mayor precisión en el reconocimiento de los usuarios.

El comando de voz que tenga la distancia menor entre las dos palabras y que este dentro de los rangos ajustados será la palabra que se busca y se dará por hecho que la voz grabada es del usuario, por lo cual el programa reconoce a la persona y le da acceso al sistema. Si por alguna razón ninguna relación entra en los parámetros mínimos que se han establecido, el programa mostrará un mensaje de no identificado.

3.2 DIAGRAMA DE BLOQUES DEL SISTEMA

El sistema de reconocimiento de voz que se ha realizado puede ser resumido en el siguiente esquema.

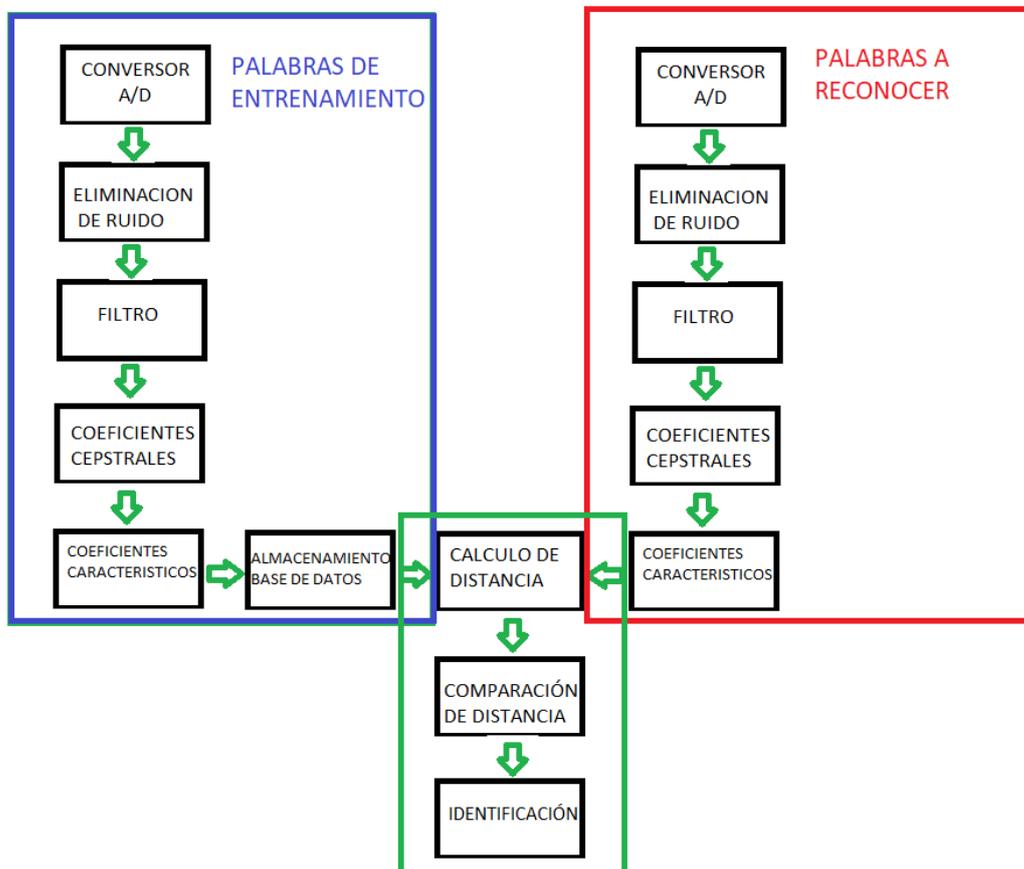


Imagen 7: Diagrama esquemático del sistema de reconocimiento de voz

El diagrama muestra a la izquierda un apartado donde se tienen las palabras de entrenamiento que equivalen a la base de datos, una vez que las palabras se han grabado y cumplan todos los procesos necesarios hasta obtener los coeficientes característicos serán almacenados en la base de datos.

A la derecha se tienen las palabras a reconocer, qué a diferencia de las palabras de entrenamiento, los coeficientes no serán guardados en una base de datos para un uso posterior, pero estas palabras si serán procesadas por las mismas funciones.

Una vez se tengan los coeficientes, tanto de las palabras de entrenamiento como de las palabras a reconocer, se puede realizar la comparación e identificar la palabra que saldría del proceso ilustrado en la parte central del esquema.

3.3 MICRÓFONO

Se ha utilizado el micrófono integrado en el portátil del proveedor Realtek, el micrófono se ha configurado a máxima potencia de audio, con 20 dB de amplificación del sonido.

Se han elegido las opciones de supresión de ruidos y cancelación de ecos acústicos para que la señal sea lo más limpia posible y el formato elegido es un mono canal, en este caso el 2, 16 bits y 44100 Hz, que es la frecuencia de muestreo utilizada en el programa.

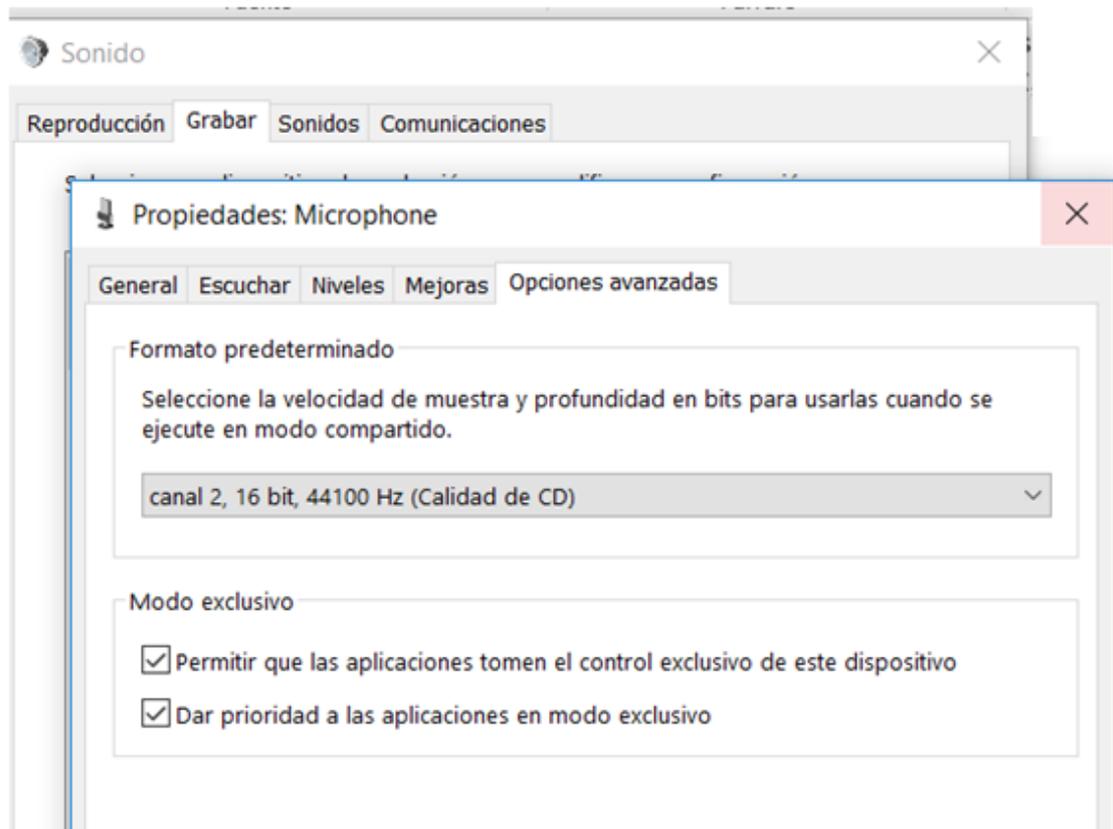


Imagen 8: Configuración del micrófono

3.4 ENTRENAMIENTO Y BASE DE DATOS

3.4.1 ADQUISICIÓN

Como se ha mencionado a lo largo de la memoria la frecuencia de muestreo que se ha elegido es el 44100 Hz, con una resolución de 8 bits y mono canal.

Cuando se han realizado las investigaciones para elaborar el proyecto, la función que era utilizada en Matlab para adquirir un audio era:

Wavrecord (n, Fs)

Pero a la hora de la implementación, las versiones más recientes de este programa no disponen de esa función, y en su reemplazo utilizan:

recorder = audiorecorder (Fs, nBits, nChannels)

Donde *Fs*. es la frecuencia de muestreo, *nBits* la resolución, y *nChannels* el número de canales siendo estéreo o mono.

Por lo que se ha modificado la forma de obtención de la señal que en un principio estaba establecida. El código mostrado adquiere una señal de audio con los parámetros que se le indican, con una duración de la grabación de 3.5 seg.

%Grabación de la primera palabra

aperturaSilvia= audiorecorder(Fs,8,1); %Grabar el audio con la Fs y canal mono

recordblocking(aperturaSilvia,3.5);%Se graba 3.5 seg

y0=getaudiodata(aperturaSilvia);%Se pasa el audio a formato para poder procesar

Una diferencia a destacar es que para poder usar la señal de audio que se ha grabado, primero se debe de crear un array con los valores, que es lo que hace la función *getaudiodata*.

3.4.2 ELIMINACIÓN DE RUIDO

Se empieza realizando un cálculo de la energía que tiene la señal de audio. Se calcula el número de muestras de la señal. Se normalizan las muestras, ya que en MATLAB la señal debe de tomar valores entre -1 y 1 para poder ser utilizado sin pérdidas. Y, por último, se elevan las muestras al cuadrado y se divide por la longitud de muestras con lo que se obtiene el valor de la energía promedio.

longitud = length(sil); % longitud del vector

pico=max(abs(sil)); %Obtener el pico máximo

sil=sil/pico; %Normalización de la señal

*promedio_senal = sum(sil.*sil)/longitud; %promedio señal entera*

Posteriormente se divide la señal normalizada en ventanas de un número determinado de muestras y si la energía es mayor que el umbral de decisión que se ha tomado de la energía promedio la ventana es conservada, en caso contrario se elimina ya que será un silencio.

$THRES = 0.2$; %Valor UMBRAL que usa para comparar

$THRES$ es el valor del umbral de decisión que corresponde a un 2% de la energía promedio de la señal entera. Las ventanas se han elegido de 80 muestras que aproximadamente son 10 ms en frecuencia de muestreo.

Se han realizado pruebas con diferentes umbrales y se detectó que con este umbral el programa realizaba mejor la eliminación de los silencios.

En las dos gráficas se observa una prueba realizada de eliminación de silencios. En la primera gráfica se representa la señal sin pasar por la función de eliminación de ruido y en la segunda gráfica se observa la señal pasada por la función silencio, donde se puede observar como se ha eliminado toda la parte que no lleva señal.

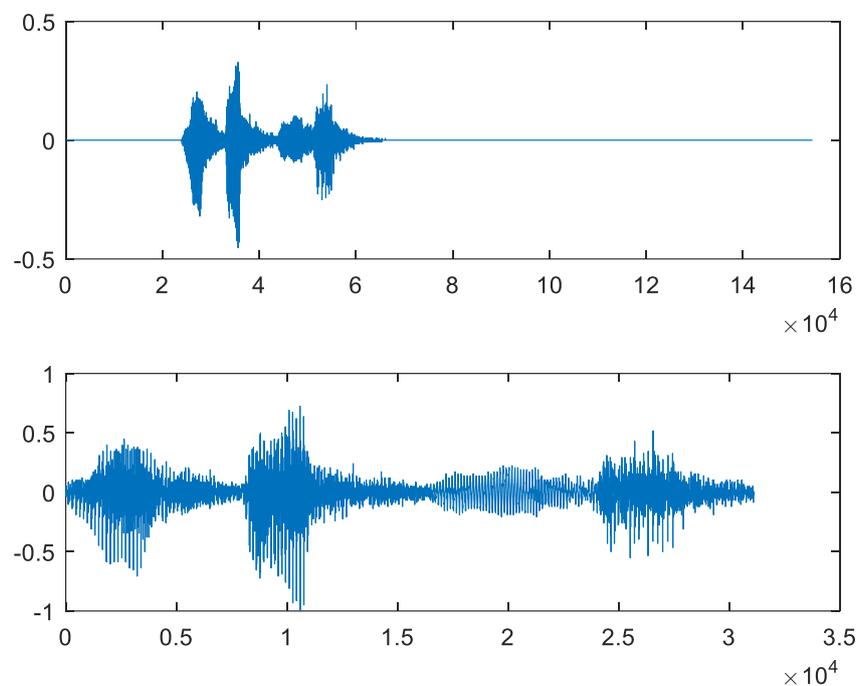


Imagen 9: Resultado de la señal al eliminar ruido

La función que se ha utilizado para la eliminación de ruido es silencio.m (ver anexo página 59):

function y = silencio(sil)

y es la variable de salida, donde se almacenan los valores que no son silencios y no tienen que ser eliminados.

sil es el parámetro de entrada de la función que en este caso es el array que devuelve la función *getaudiodata*.

3.4.3 FILTRO

El filtro pré-énfasis se ha codificado según la fórmula que se estableció en un principio, que correspondía a uno de los filtros más utilizados.

La función que se ha utilizado para filtrar la señal es `énfasis.m` (ver anexo página 60):

function [y]=énfasis(x)

y es la señal que devuelve la función después de ser filtrada.

x es la señal que le pasamos a la función después de eliminar el ruido.

Los parámetros que se han usado en el filtro son los establecidos por definición. Y para poder realizar el filtrado Matlab proporciona la función `filter`, a la cual le pasaremos los parámetros.

%Parámetros usados en el filtro

b=[1 -0.95];

A=1;

%Usamos la función filter definida en matlab

y=filter(b,A,x);

3.4.4 SEGMENTACIÓN

La segmentación se aplica con una ventana Hamming de 240 que equivale a 30 ms.

seg=zeros(numero_segmentos,240)'; %Matriz de segmentos, numero_seg 240

ventana=hamming(240);%Ventana de Hamming de 30ms

Esta función tiene como propósito cortar la señal en segmentos de análisis y guardarlos en una matriz.

La función que se ha utilizado para la segmentación es `segmentación.m` (ver anexo página 60):

function[seg]=segmentacion(y)

Seg son los segmentos retornados por la función

y es la señal que se le pasa después de ser filtrada.

3.4.5 COEFICIENTES CARACTERISTICOS. PARAMETRIZACIÓN

Para la obtención de los coeficientes que caracterizan la señal existen varios métodos que se han explicado en el capítulo anterior. En este caso, se va a elegir un análisis cepstral. Aunque el análisis de predicción lineal es una de las técnicas más potentes y útiles para codificar voz el análisis cepstral tiene la ventaja de que representa la información de una manera más eficiente por lo que proporcionará mejores resultados para el algoritmo. También cabe destacar que de los coeficientes cepstrales se pueden derivar una serie de parámetros que son invariantes, no les influyen las distorsiones que puedan surgir; por ejemplo, una de ellas puede ser ocasionada por el micrófono.

Se ha optado por la implementación completa del cálculo cepstrum, descartando las diferentes opciones que proporciona Matlab (cceps, rceps,..) al no proporcionar los resultados como precisamos para nuestro reconecedor.

Una vez calculados los coeficientes deben de ser normalizados para reducir variabilidades espectrales.

Una vez que se ha realizado la normalización los coeficientes cepstrales pasan por otra función que sirve para expandirlos por medio de una representación polinomial ortogonal en intervalos de 90ms cada 10 ms. Resultado es una función de 20 elementos característicos.

Para realizar la parametrización se han utilizado las funciones cepstrum.m (ver anexo página 60) y coeficientes.m (ver anexo página 61)

```
function coef= coeficientes(c);
```

c parámetro de entrada que coincide con los coeficientes cepstrum devueltos por la función cepstrum.

coef número de coeficientes polinomiales devueltos por la función coeficientes.

```
function c=cepstrum(segmentos)
```

Segmentos es el parámetro que devuelve la función segmentación.

c los coeficientes cepstrum retornados para convertirlos es coeficientes polinomiales.

3.4.6 BASE DE DATOS

La base de datos creada recoge muestras de voz de 10 usuarios, que han grabado las dos palabras reconocidas por la interfaz: apertura o cierre (ver anexo página 55).

Una vez se graba la palabra el registro de la señal de voz pasa por las funciones explicadas anteriormente y se guarda en el archivo de audio. Luego es reproducido para comprobar si se ha grabado correctamente.

Los usuarios serán añadidos según necesidades y en el momento que un usuario deje de estar acreditado se podrá borrar fácilmente de la aplicación. Es tan sencillo como eliminar la parte de código que corresponda a ese usuario.

Al disponer de una base de datos que puede ser modificada por el diseñador, una ventaja de este sistema es la posibilidad de poder modificar las palabras existentes semanal o mensualmente y con ello proporcionar una mayor seguridad. El inconveniente es que la base de datos actual debe de ser borrada y todos los usuarios deben volver a grabar la nueva palabra a reconocer.

3.5 RECONOCIMIENTO DEL HABLA.

3.5.1 DECISIÓN

Una característica fundamental de los sistemas de reconocimiento de voz es la forma en que las matrices características son combinadas y comparadas con los patrones de referencia. Para realizar esta operación es necesario definir una medida de distancia entre los vectores característicos. En este caso se ha usado la distancia Euclídea.

La función devuelve un vector distancia de la matriz característica respecto a la distancia de la matriz guardada.

La función que se ha utilizado para el cálculo de la distancia es `distancia.m` [8] (ver anexo página 61).

```
function dist = distancia(f_1,f_2)
```

f_1 son los coeficientes de la señal guardada en base de datos y f_2 son los coeficientes de la señal que se ha grabado para reconocer.

$dist$ es el vector que guarda la distancia de la matriz característica con respecto la matriz de los patrones guardados.

3.5.2 INTERFAZ GRÁFICA

Para la realización de la interfaz gráfica se ha utilizado GUIDE de Matlab [10]. Es un entorno que permite diseñar primero gráficamente la interfaz de usuario para posteriormente poder ser volcada en el código.

Es una interfaz muy intuitiva, se pueden ir colocando los botones, textos, gráficos... según se necesiten.

El diseño realizado en este trabajo es una interfaz con 3 botones y 2 gráficas [11], los botones indican la operación a realizar y las gráficas muestran la señal grabada y la señal en la base de datos.

A la hora de la creación de la interfaz también se deben de añadir los audios que se han creado en la base de datos.

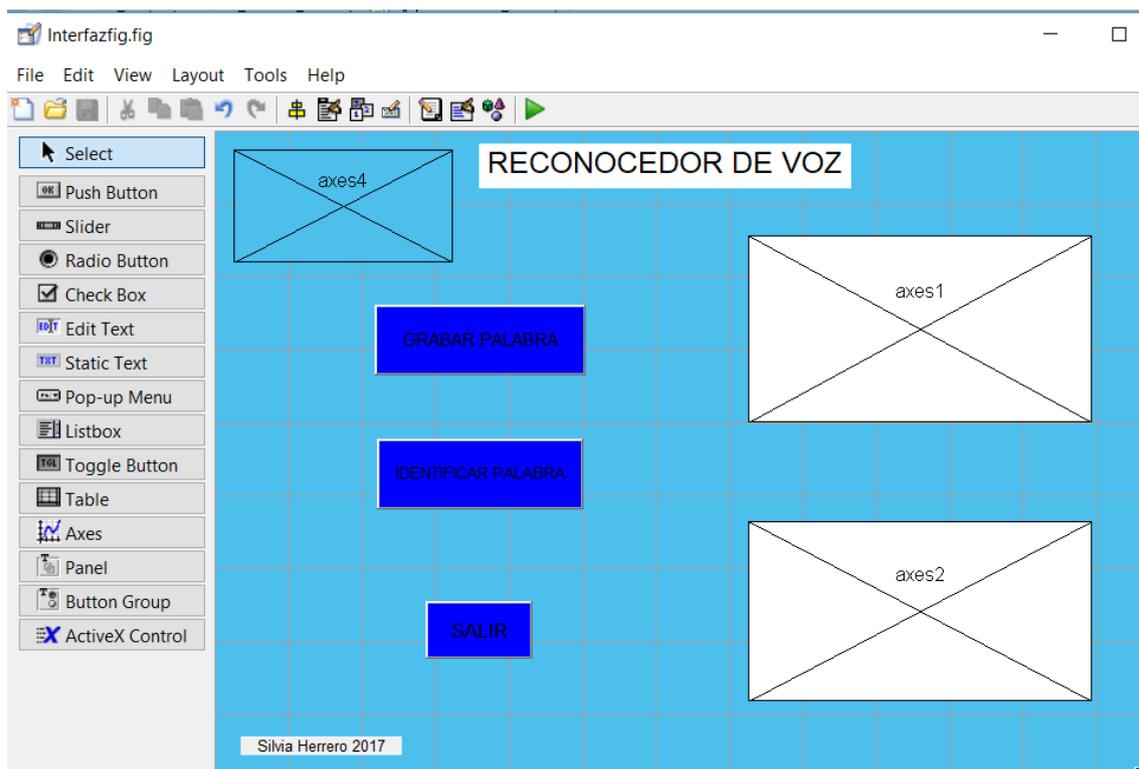


Imagen 10: Diseño de la interfaz gráfica imagen de GUI

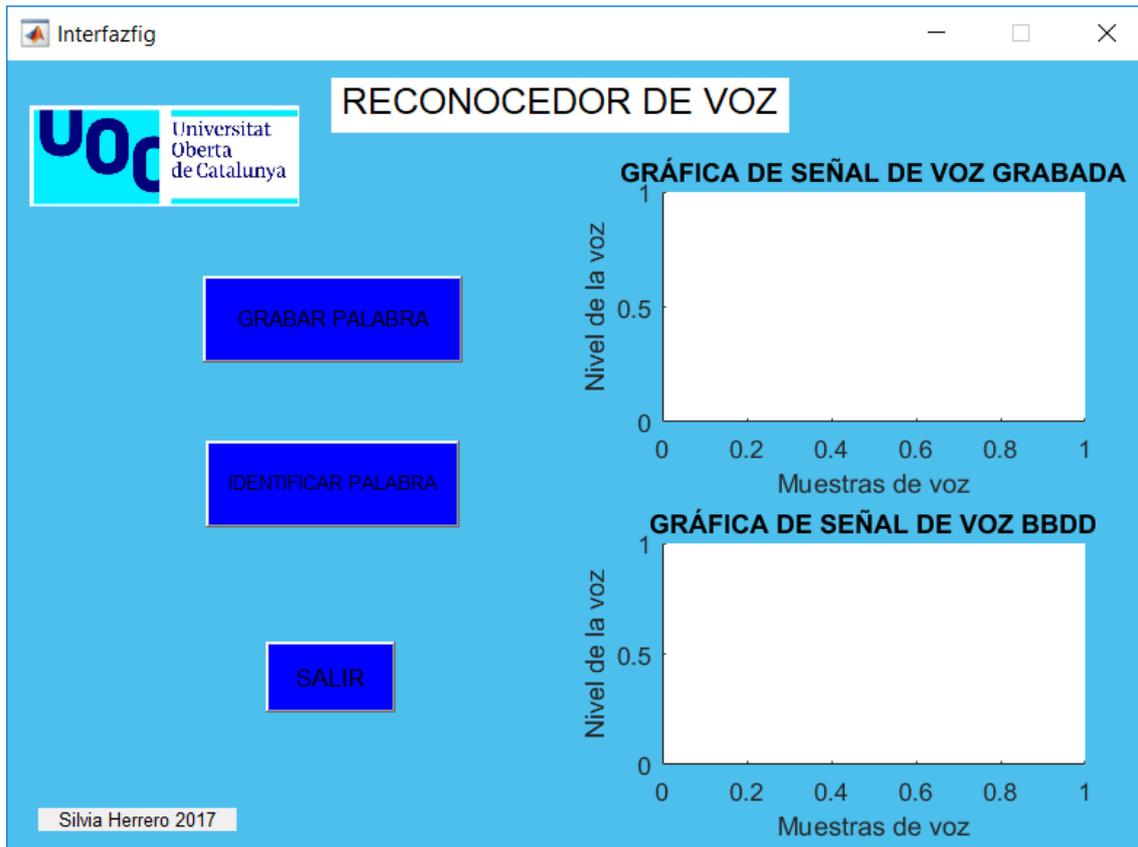


Imagen 11: Interfaz gráfica de Reconocedor de Voz creado en Matlab despues de la ejecución

Dentro de las funciones aparecen recuadros donde se avisa a la interfaz del estado del programa:

- Botón grabar palabra: El botón realiza la función de activar el micrófono y grabar durante 3.5 segundos la palabra a reconocer. Posteriormente pasará toda la palabra por los procesos descritos anteriormente para así llegar a los coeficientes característicos. También se muestra la gráfica de la señal de voz a reconocer en el primer recuadro (ver anexo página 65).

function pushbutton1_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)



Imagen 12: Interfaz gráfica después de pulsar el botón grabar palabra

- Botón identificar palabra: Este botón es el encargado de reconocer la palabra y de tomar la decisión.

Lo primero que se hace es guardar en la interfaz los coeficientes característicos de las señales de la base de datos y la señal a reconocer, luego se llama a la función distancia la cual hace la comparación, y por último va obteniendo las distancias menores de cada palabra [18].

Ya en posesión del menor elemento que es el que indica cual es la menor distancia entre las dos palabras, el programa puede reconocer el termino que se ha dicho. Posteriormente haciendo las pruebas necesarias se realiza un ajuste de mínimos de las distancias para que el programa sea lo más preciso posible a la hora de reconocer.

También posee una gráfica donde se muestra la señal de la palabra que se ha reconocido (ver anexo página 65).

```
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```



Imagen 13: Interfaz gráfica después de pulsar el botón identificar palabra

- Botón salir: Una vez que se pulsa este botón el programa muestra un mensaje de despedida y es cerrado (ver anexo página 75).

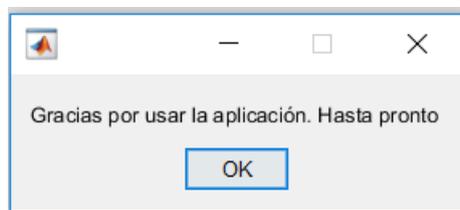


Imagen 14: Ventana que aparece después de presionar el botón salir

```
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton3 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
```

3.6. PRUEBAS Y RESULTADOS

En este capítulo se van a mostrar los resultados obtenidos al realizar las pruebas con el programa desarrollado de reconocimiento de voz para poder obtener y comparar los objetivos alcanzados, las ventajas y desventajas del proyecto.

Para analizar estas pruebas se han usado las matrices de confusión que sirven para comparar los valores reales con los valores de predicción para cada estado de predicción especificado.

Una matriz de clasificación es una herramienta importante para evaluar los resultados de la predicción, ya que hace que resulte fácil entender y explicar los efectos de las predicciones erróneas. Al ver la cantidad, podrá saber rápidamente en cuántas ocasiones ha sido exacta la predicción del modelo.

A continuación, se va a realizar un estudio de los usuarios que se han registrado en la base de datos.

3.6.1 MATRIZ DE CONFUSIÓN

Para estas pruebas se han utilizado los 10 usuarios registrados en la base de datos, hombres y mujeres adultos y un niño, se trata de acceder al sistema con alguno de estos usuarios.

La prueba se realiza en cada persona con la pronunciación 15 veces de las palabras a reconocer para ver el porcentaje de acierto del sistema.

La primera prueba se realiza con la palabra apertura. La matriz de confusión se ha realizado en Excel. El número máximo de relación se muestra en color azul.

Apertura	Silvia	Eduardo	Natalia	Emilio	Pilar	Emilio1	Angeles	Luis	Ángel	Emilio 2	No identificado	TOTAL
Silvia	13		2									15
Eduardo		9			2	3					1	15
Natalia	3		10		2							15
Emilio		4		8				3				15
Pilar	2		2		11							15
Emilio 1	2		4			7					2	15
Angeles			1		4		10					15
Luis		2		4				9				15
Ángel				4					10	1		15
Emilio 2		2		3				1		9		15
TOTAL	20	17	19	19	19	10	10	13	10	10	3	150

PRECISIÓN 66 %

Tabla 1: Matriz de confusión de la palabra apertura junto a su índice de aciertos en tanto por ciento



Imagen 15: Interfaz gráfica de reconocimiento de la palabra apertura por el usuario 1



Imagen 16: Interfaz gráfica de reconocimiento de la palabra apertura por el usuario 2

Con los resultados obtenidos, se determinó que con la palabra apertura los usuarios femeninos son mejor reconocidos, pero hay usuarios que tienen incidencias con otros como es el caso del usuario Angeles con el usuario Pilar, ya que sus señales de voz son muy parecidas. También en alguna de las grabaciones no se identificó al usuario. Otra cosa que se ha detectado es que al usuario Emilio1, que resulta ser el niño, es el usuario con menos porcentaje de acierto que todos los demás. Aun así, el porcentaje de aciertos es ligeramente superior al 50%.

La segunda prueba se realiza con la palabra cierre, con el mismo número de intentos y de usuarios.

Cierre	Silvia	Eduardo	Natalia	Emilio	Pilar	Emilio1	Angeles	Luis	Ángel	Emilio 2	No identificado	TOTAL
Silvia	12		1		2							15
Eduardo		10		3				2				15
Natalia	2		13					1				15
Emilio		3		10		1		1				15
Pilar	2				12						1	15
Emilio 1	2		2			8		1			2	15
Angeles					4		11					15
Luis		1	1	3				10				15
Ángel		1		1					12	1		15
Emilio 2		3		3						9		15
TOTAL	18	18	17	20	18	9	11	14	12	10	3	150

PRECISIÓN	73,3333333	%
-----------	------------	---

Tabla 2: Matriz de confusión de la palabra cierre junto a su índice de acierto en tanto por ciento.



Imagen 17: Interfaz gráfica de reconocimiento de la palabra cierre por el usuario 1



Imagen 18: Interfaz gráfica de reconocimiento de la palabra cierre por el usuario 2

Con los resultados obtenidos, se determinó que la palabra cierre tiene un porcentaje más alto de aciertos que la palabra apertura. En este caso no hay distinción entre hombres y mujeres. Se vuelve a observar que el usuario niño es el que menor veces es reconocido.

3.6.2 PRUEBAS CON PERSONAS Y PALABRAS NO IDENTIFICADAS

Se realizaron las pruebas con los mismos usuarios registrados en la base de datos con palabras aleatorias, se pronuncian 15 palabras aleatorias que no sean ni apertura ni cierre.

Aleatoria	Silvia	Eduardo	Natalia	Emilio	Pilar	Emilio1	Angeles	Luis	Ángel	Emilio 2	No identificado	TOTAL
Silvia	1		2								12	15
Eduardo		2		3							10	15
Natalia	2		2								11	15
Emilio			1	1							13	15
Pilar			1		1		3				10	15
Emilio 1		1		2		3					9	15
Angeles					2		1				12	15
Luis		1		1				1	2		10	15
Ángel								1	1		13	15
Emilio 2				1					1	1	12	15
TOTAL	3	4	6	8	3	3	4	2	4	1	112	150

PRECISIÓN	74,6666667	%
-----------	------------	---

Tabla 3: Matriz de confusión de palabras aleatorias junto a su índice de acierto en tanto por ciento.



Imagen 19: Interfaz gráfica de reconocimiento de palabra aleatoria

Se observa en los resultados, que muchas de las palabras tienen una señal parecida a las palabras de la base de datos ya que son pronunciadas por las mismas personas que están en ella. Por lo tanto, al pronunciar palabras aleatorias que se parecen a la forma de la señal de la palabra en la base de datos como por ejemplo cerrar o cierre las identifica. Aun así, son resultados mejores que los anteriores en cuanto a precisión.

Se realiza la prueba con usuarios que no están registrados en la base de datos, un total de 5 personas, mujeres y hombre adultos. Estas personas han pronunciado las palabras apertura y cierre 15 veces cada una.

Intruso A	1	2	3	4	5	No identificado	TOTAL
1	1					14	15
2		2				13	15
3			3			12	15
4				2		13	15
5					1	14	15
TOTAL	1	2	3	2	1	66	75

PRECISIÓN 88 %

Tabla 4: Matriz de confusión de la palabra cierre pronunciada por intrusos

Intruso C	1	2	3	4	5	No identificado	TOTAL
1	1					14	15
2		1				14	15
3			2			13	15
4				2		13	15
5					2	13	15
TOTAL	1	1	2	2	2	67	75

PRECISIÓN 89,3333333 %

Tabla 5: Matriz de confusión de la palabra cierre pronunciada por intrusos



Imagen 20: Interfaz gráfica de reconocimiento de palabra pronunciada por un intruso

Los resultados muestran un mayor índice de aciertos en personas no reconocidas por la base de datos, tanto con la pronunciación de la palabra apertura y la pronunciación de la palabra cierre. Estos valores más altos son normales ya que las personas que están en la base de datos tienen ajustados sus parámetros por las pruebas realizadas con ellos mientras que estas personas no.

3.6.3 RECOMENDACIONES

Como se puede observar el sistema no es fiable al 100% de los casos o en cualquier situación y con ello limita el alcance de la aplicación.

Se propone implementar un sistema biométrico de seguridad por reconocimiento de voz añadiendo más de un parámetro característico, como son por ejemplo los coeficientes de predicción lineal. Con esto se consigue que la probabilidad de fallo sea mucho menor, ya que en vez de utilizar solamente la distancia menor entre las dos palabras y ajustar los parámetros a las voces que se tienen, se utilizarían los dos conjuntamente.

El sistema será seguro en cuando a coeficientes ya que el software no podría ser alterado por ninguna persona, ya que cada persona contiene los coeficientes inalterables en una base de datos establecida por el diseñador.

En un futuro podría existir otra base de datos conjuntamente a la ya existente, en la que se guarde un número más extenso de palabras pronunciadas por los usuarios del sistema y semanal o mensualmente el diseñador podrá ir cambiando las palabras de una base de datos a otra sin necesidad de requerir a los usuarios de nuevo.

También, un sistema que sea capaz de reconocer una instrucción compuesta en vez de una instrucción simple será mucho menos vulnerable a fallos.

Finalmente, estas recomendaciones son algunas de las muchas, que podrían aplicarse a este sistema y que nos dejan puerta abierta a futuras revisiones y mejoras de cara a un proyecto un poco más ambicioso y en el que se disponga más tiempo para el desarrollo del mismo.

4. CONCLUSIONES

La seguridad en una cárcel es el principal objetivo que se desea alcanzar en un organismo estatal de este tipo. Un sistema de seguridad por reconocimiento de voz puede no solo aumentarla si no también tener constancia de los usuarios que accedan a él y con ello aportar cierta tranquilidad.

En un sistema de reconocimiento por voz, la toma de decisión es difícil debido a factores como la similitud de las palabras, el nivel de ruido del lugar donde se realizan las grabaciones, ya que esto puede influenciar en la señal o los estados de ánimo de los usuarios que no siempre la palabra se va a pronunciar igual.

Un sistema de seguridad será más fiable ya que pasará por diversos métodos, siendo el más importante la obtención de los coeficientes característicos y las condiciones implementadas por el programador, disminuyendo la probabilidad de error al reconocer al usuario. Aunque un aspecto muy importante a destacar es que el diseñador no puede corregir si el usuario dice una palabra en un tono de extrema alegría o extrema tristeza, ya que esto modifica las características de la voz afectando a estos coeficientes.

La seguridad por voz irá en aumento al paso de las diferentes etapas del sistema como por ejemplo filtrado y normalización, estas tecnologías pueden ir mejorando la fiabilidad y la velocidad en la toma de decisión. Estas a su vez ayudarán en la implementación de sistemas de seguridad más eficientes.

La opción de usar Matlab como herramienta principal permitió realizar una interfaz, llamada GUI, para el reconocimiento de voz de uso fácil, eficaz y con capacidad de ejecutar operaciones de procesamiento de señales. Este era uno de los objetivos en este proyecto.

Se ha aprendido que para realizar un sistema de reconocimiento de voz se necesitan muchas partes tanto de software como de hardware, que realicen sus funciones de manera adecuada, siendo el software el de mayor importancia. Gracias a este, se pueden programar los algoritmos para poder reconocer al usuario, pero el hardware también tiene su importancia debido a que si se usa un micrófono mejor se puede obtener una mejor calidad de la señal adquirida y por lo tanto dejar que el software se encargue solo del procesamiento de la señal y no tenga que desperdiciar tiempo en limpiarla, filtrarla o normalizarla.

El mayor problema que se presentó durante la realización del proyecto fue la obtención de voces para poder probar el correcto funcionamiento del programa, ya que actualmente debido a los horarios laborales y las distancias entre familiares o amigos, hacían difícil poder establecer una cita.

La solución a este problema fue el desplazamiento por mi parte con el material necesario en los horarios en los que familiares y amigos estuvieran disponibles.

Luego aparecieron problemas de codificación en Matlab ya que se requerían conceptos muy complejos nunca antes usados.

Esto se resolvió consultados documentos en internet e ir probando comandos en Matlab hasta llegar a la solución requerida, lo cual podía llevar varios días.

Otro problema importante que se detectó fue que las palabras elegidas en un principio, que eran abrir y cerrar, se tuvieron que modificar por apertura y cierre. Como consecuencia se tuvo que modificar íntegramente la base de datos. Esto fue debido a que las señales de las palabras elegidas para el proyecto tenían las gráficas muy similares. Por lo tanto, el programa registraba muchos errores.

De esta manera se pudieron solucionar los problemas que surgieron durante la realización del proyecto y así poder llegar a la solución esperada.

5. BIBLIOGRAFÍA

- [1] Análisis y síntesis de la voz mediante ordenador [Internet]. Lpi.tel.uva.es. 2017 [cited 10 November 2017]. Available from: https://www.lpi.tel.uva.es/~nacho/docencia/ing_ond_1/trabajos_05_06/io3/public_html/procesamiento/reconocimiento/reconocimiento_04.html
- [2] A.Llorca Á. Cuatro herramientas de reconocimiento de voz que permiten prescindir del teclado (entre otros) y agilizar las tareas [Internet]. Gen-beta.com. 2017 [cited 18 October 2017]. Available from: <https://www.gen-beta.com/herramientas/cuatro-herramientas-de-reconocimiento-de-voz-que-permiten-prescindir-del-teclado-entre-otros-y-agilizar-las-tareas>
- [3] Biometría [Internet]. Es.wikipedia.org. 2017 [cited 1 October 2017]. Available from: <https://es.wikipedia.org/wiki/Biometr%C3%ADa>
- [4] Apuntes de Procesado de Audio UOC [Internet]. 2017 [cited 1 October 2017]. Available from: [https://www.exabyteinformatica.com/uoc/Audio/Procesamiento_de_audio/Procesamiento_de_audio_\(Intro\).pdf](https://www.exabyteinformatica.com/uoc/Audio/Procesamiento_de_audio/Procesamiento_de_audio_(Intro).pdf)
- [5] Carlos Mauricio Galvis Traslaviña M. Introducción a la biometría (página 3) - Monografias.com [Internet]. Monografias.com. 2017 [cited 20 October 2017]. Available from: <http://www.monografias.com/trabajos43/biometria/bio-metria3.shtml>
- [6] Cite a Website Cite This For Me [Internet]. Euskalnet.net. 2017 [cited 20 November 2017]. Available from: <http://www.euskalnet.net/iosus/speech/recog2.html>
- [7] Cómo funciona un sistema de reconocimiento de voz [Internet]. eldiario.es. 2017 [cited 14 October 2017]. Available from: http://www.eldiario.es/turing/reconocimiento-voz-biometria_0_201230680.html
- [8] Elieser Ernesto Gallego Martínez M. Control mediante reconocimiento remoto de voz comprimida sobre redes IP - Monografias.com [Internet]. Monografias.com. 2017 [cited 26 November 2017]. Available from: <http://www.monografias.com/trabajos100/control-reconociiento-remoto-voz-comprimida-redes-ip/control-reconociiento-remoto-voz-comprimida-redes-ip.shtml>

- [9] Fonatorio [Internet]. [Internet]. 2017 [cited 1 October 2017]. Available from: <http://www.fceia.unr.edu.ar/prodivoz/fonatorio.pdf>
- [10] GUI de MATLAB [Internet]. Es.mathworks.com. 2017 [cited 13 November 2017]. Available from: <https://es.mathworks.com/discovery/matlab-gui.html>
- [11] Guía para interfaz de MATLAB [Internet]. 2017 [cited 1 October 2017]. Available from: http://www.utm.mx/~vero0304/HCPM/GUI_Matlab.pdf
- [12] [Internet]. 2017 [cited 16 October 2017]. Available from: http://aran-txa.ii.uam.es/~jortega/Tema3_TAPS_resto.pdf
- [13] [Internet]. Euskalnet.net. 2017 [cited 13 October 2017]. Available from: <http://www.euskalnet.net/iosus/speech/lp.html>
- [14] [Internet]. ie.itcr.ac.cr. 2017 [cited 23 October 2017]. Available from: <http://www.ie.itcr.ac.cr/palvarado/PDS/pds.pdf>
- [15] [Internet]. 2017 [cited 26 November 2017]. Available from: <http://mit.ocw.universia.net/6.345/NR/rdonlyres/Electrical-Engineering-and-Computer-Science/6-345Automatic-Speech-RecognitionSpring2003/87A83F78-E890-4F14-A1E1-B58703B63F13/0/lecture5.pdf>
- [16] [Internet]. 2017 [cited 14 November 2017]. Available from: <http://www.vision.ime.usp.br/~jorjasso/files/expo.pdf>
- [17] [Internet]. 2017 [cited 14 November 2017]. Available from: http://www.fceia.unr.edu.ar/prodivoz/banco_filtros_bw.pdf
- [18] [Internet]. 2017 [cited 17 December 2017]. Available from: <https://www.lawebdelprogramador.com/foros/Matlab/1586286-Valor-maximo-y-minimo-de-una-matriz.html>
- [19] [Internet]. 2017 [cited 18 November 2017]. Available from: http://www.utm.mx/~vero0304/HCPM/GUI_Matlab.pdf
- [20] [Internet]. 2017 [cited 13 November 2017]. Available from: https://www.dspace.espol.edu.ec/bitstream/123456789/10740/11/MATLAB_GUIDE.pdf

- [21] [Internet]. 2017 [cited 21 November 2017]. Available from: <http://webpersonal.uma.es/de/gfdc/docencia/GuiSection.pdf>
- [22] [Internet]. 2017 [cited 14 November 2017]. Available from: http://www.ehu.eus/juancarlos.gorostizaga/matell15/programas/Ej_programas.pdf
- [23] [Internet]. 2017 [cited 6 December 2017]. Available from: <https://www.youtube.com/watch?v=Rg1g0OooSao>
- [24] [Internet]. 2017 [cited 6 December 2017]. Available from: https://www.youtube.com/watch?v=GsoVFaVUF_A
- [25] [Internet]. 2017 [cited 6 December 2017]. Available from: <https://www.youtube.com/watch?v=O3yPLUgp4nk>
- [26] [Internet]. 2017 [cited 14 October 2017]. Available from: <http://bibing.us.es/proyectos/abreproy/12054/fichero/MEMORIA%252F8.Cap% C3%ADtulo+3.pdf>
- [27] Las 5 Mejores aplicaciones de reconocimiento de voz [Internet]. Profesional Review. 2017 [cited 26 November 2017]. Available from: <https://www.profesionalreview.com/2016/08/17/5-mejores-aplicaciones-reconocimiento-voz/>
- [28] MATLAB - El lenguaje del cálculo técnico [Internet]. Es.mathworks.com. 2017 [cited 16 October 2017]. Available from: <https://es.mathworks.com/products/matlab.html>
- [29] Micrófonos [Internet]. Ehu.eus. 2017 [cited 21 October 2017]. Available from: <http://www.ehu.eus/acustica/espanol/electricidad/micres/micres.html>
- [30] Muestreo y cuantificación [Internet]. Ceres.ugr.es. 2017 [cited 20 October 2017]. Available from: <http://ceres.ugr.es/~alumnos/luis/mycuan.htm>
- [31] Nuance EMEA Tienda electrónica - Dragon NaturallySpeaking 13 Home [Internet]. Shop.nuance.es. 2018 [cited 2 January 2018]. Available from: https://shop.nuance.es/store/nuanceeu/es_ES/pd/productID.313036400/pgm.9540940/OfferID.12345678910/Currency.EUR?utm_source=google&utm_medium=emea-cpc&utm_campaign=DBU+%2F+DNS+Speech+Recognition+%2F+Generic+%2F+ES+=+ES+%2F+ES+%2F+None+%2F+Broad&keyword=%2Breconocimiento+%2Bde+%2Bvoz-b

[32] Oppenheim, Alan V., Schafer, R.W., Buck, J.R. Discrete-time signal processing. 2ª edición. Prentice-Hall, Inc. New Jersey, 1989 Oskar Sánchez M.

[33] Procesamiento Digital de Señales - Monografias.com [Internet]. Monografias.com. 2017 [cited 19 October 2017]. Available from: <http://www.monografias.com/trabajos95/procesamiento-digital-de-senales/procesamiento-digital-de-senales.shtml>

[34] PROGRAMA DE TRATAMIENTO DIGITAL DE VOZ [Internet]. Gaps.ssr.upm.es. 2017 [cited 14 November 2017]. Available from: <http://www.gaps.ssr.upm.es/TDV/prelaiinfr.html>

[35] Reconocimiento del habla [Internet]. Es.wikipedia.org. 2017 [cited 1 October 2017]. Available from: https://es.wikipedia.org/wiki/Reconocimiento_del_habla

[36] Reconocimiento de Voz [Internet]. Redyseguridad.fi-p.unam.mx. 2017 [cited 10 November 2017]. Available from: <http://redyseguridad.fi-p.unam.mx/proyectos/biometria/clasificacionsistemas/aplicacionesactuales.html>

[37] Reconocimiento de Voz [Internet]. Ict.udlap.mx. 2017 [cited 26 November 2017]. Available from: <http://ict.udlap.mx/people/ingrid/Clases/IS412/in-dex.html>

[38] Replay attack [Internet]. En.wikipedia.org. 2017 [cited 14 October 2017]. Available from: https://en.wikipedia.org/wiki/Replay_attack

[39] Rey-Lancheros D, Gavilán-Acosta H, Espitia-Cuchango H. Implementación de un algoritmo para la identificación de usuarios considerando problemas fisiológicos que afectan el habla. 2017.

<https://cours.etsmtl.ca/sys828/REFS/Intro/Fundamentals%20of%20Speaker%20Recognition%20-%20Homayoon%20Beigi.pdf>

[40] Sergio d. werner M. Aplicación de Nuevas Tecnologías al Sistema Electoral – Biometría y Voto Electrónico (página 2) - Monografias.com [Internet]. Monografias.com. 2017 [cited 15 October 2017]. Available from: <http://www.monografias.com/trabajos82/biometria-y-voto-electronico/biometria-y-voto-electronico2.shtml>

[41] Types of Biometrics · About Biometrics · Biometrics Institute [Internet]. Biometricsinstitute.org. 2017 [cited 15 October 2017]. Available from: <http://www.biometricsinstitute.org/pages/types-of-biometrics.html>

[42] Ventana (función) [Internet]. Es.wikipedia.org. 2017 [cited 16 November 2017]. Available from: [https://es.wikipedia.org/wiki/Ventana_\(funci%C3%B3n\)](https://es.wikipedia.org/wiki/Ventana_(funci%C3%B3n))

[43] What is biometrics? - Definition from WhatIs.com [Internet]. SearchSecurity. 2017[cited 15 October 2017]. Available from: <http://searchsecurity.techtarget.com/definition/biometrics>

6. ANEXO

CÓDIGO PROGRAMA

% % BASE DE DATOS

%

% clc

% %

Fs= 44100; %Frecuencia de muestreo

%Grabación de la primera palabra

aperturaSilvia= audiorecorder(Fs,8,1); %Grabar el audio con la Fs y canal mono

recordblocking(aperturaSilvia,3.5);%Se graba 3.5 seg

y1=getaudiodata(aperturaSilvia);%Se pasa el audio a formato para poder procesar

y1=silencio(y1); %Eliminación de ruido

fil1=enfasis(y1);%Filtro

segmentos1=segmentacion(fil1);%Segmentamos la señal en tramas de 30ms

cep1=cepstrum(segmentos1);%Se calculan los coeficientes cepstrum

coef1=coeficientes(cep1); %Se sacan los coeficientes característicos

save('aperturaSilvia.mat','coef1','y1');%Se graba la señal de audio

play(aperturaSilvia);%Reproducir la señal grabada para ver si esta bien

%Grabación de la segunda palabra

cierreSilvia= audiorecorder(Fs,8,1); %Grabar el audio con la Fs y canal mono

recordblocking(cierreSilvia,3.5);%Se graba 3.5 seg

y2=getaudiodata(cierreSilvia);%Se pasa el audio a formato para poder procesar

y2=silencio(y2); %Eliminación de ruido

fil2=enfasis(y2); %Filtro

segmentos2=segmentacion(fil2);%Se segmenta la señal en tramas de 30ms

cep2=cepstrum(segmentos2); %Se calculan los coeficientes cepstrum

coef2=coeficientes(cep2);%SE sacan los coeficientes característicos

save('cierreSilvia.mat','coef2','y2');%Se graba la señal de audio

play(cierreSilvia);%Reproducir la señal grabada para ver si esta bien

%Grabación de la tercera palabra

aperturaEduardo= audiorecorder(Fs,8,1); %Grabar el audio con la Fs y canal mono

recordblocking(aperturaEduardo,3.5);%Se graba 3.5 seg

y3=getaudiodata(aperturaEduardo);%Se pasa el audio a formato para poder procesar

y3=silencio(y3); %Eliminación de ruido

fil3=enfasis(y3);%Filtro

segmentos3=segmentacion(fil3);%Segmentamos la señal en tramas de 30ms

cep3=cepstrum(segmentos3);%Se calculan los coeficientes cepstrum

coef3=coeficientes(cep3); %Se sacan los coeficientes característicos

save('aperturaEduardo.mat','coef3','y3');%Se graba la señal de audio

play(aperturaEduardo);%Reproducir la señal grabada para ver si esta bien

%Grabación de la cuarta palabra

cierreEduardo= audiorecorder(Fs,8,1); %Grabar el audio con la Fs y canal mono

recordblocking(cierreEduardo,3.5);%Se graba 3.5 seg

y4=getaudiodata(cierreEduardo);%Se pasa el audio a formato para poder procesar

y4=silencio(y4); %Eliminación de ruido

fil4=enfasis(y4); %Filtro

```

segmentos4=segmentacion(fil4);%Se segmenta la señal en tramas de 30ms
cep4=cepstrum(segmentos4); %Se calculan los coeficientes cepstrum
coef4=coeficientes(cep4);%SE sacan los coeficientes característicos
save('cierreEduardo.mat','coef4','y4');%Se graba la señal de audio
play(cierreEduardo);%Reproducir la señal grabada para ver si esta bien
  
```

%Grabación de la quinta palabra

```

aperturaPilar= audiorecorder(Fs,8,1); %Grabar el audio con la Fs y canal mono
recordblocking(aperturaPilar,3.5);%Se graba 3.5 seg
y5=getaudiodata(aperturaPilar);%Se pasa el audio a formato para poder procesar
y5=silencio(y5); %Eliminación de ruido
fil5=enfasis(y5);%Filtro
segmentos5=segmentacion(fil5);%Segmentamos la señal en tramas de 30ms
cep5=cepstrum(segmentos5);%Se calculan los coeficientes cepstrum
coef5=coeficientes(cep5); %Se sacan los coeficientes característicos
save('aperturaPilar.mat','coef5','y5');%Se graba la señal de audio
play(aperturaPilar);%Reproducir la señal grabada para ver si esta bien
  
```

%Grabación de la sexta palabra

```

cierrePilar= audiorecorder(Fs,8,1); %Grabar el audio con la Fs y canal mono
recordblocking(cierrePilar,3.5);%Se graba 3.5 seg
y6=getaudiodata(cierrePilar);%Se pasa el audio a formato para poder procesar
y6=silencio(y6); %Eliminación de ruido
fil6=enfasis(y6);%Filtro
segmentos6=segmentacion(fil6);%Segmentamos la señal en tramas de 30ms
cep6=cepstrum(segmentos6);%Se calculan los coeficientes cepstrum
coef6=coeficientes(cep6); %Se sacan los coeficientes característicos
save('cierrePilar.mat','coef6','y6');%Se graba la señal de audio
play(cierrePilar);%Reproducir la señal grabada para ver si esta bien
  
```

%Grabación de la séptima palabra

```

aperturaNatalia= audiorecorder(Fs,8,1); %Grabar el audio con la Fs y canal mono
recordblocking(aperturaNatalia,3.5);%Se graba 3.5 seg
y7=getaudiodata(aperturaNatalia);%Se pasa el audio a formato para poder procesar
y7=silencio(y7); %Eliminación de ruido
fil7=enfasis(y7);%Filtro
segmentos7=segmentacion(fil7);%Segmentamos la señal en tramas de 30ms
cep7=cepstrum(segmentos7);%Se calculan los coeficientes cepstrum
coef7=coeficientes(cep7); %Se sacan los coeficientes característicos
save('aperturaNatalia.mat','coef7','y7');%Se graba la señal de audio
play(aperturaNatalia);%Reproducir la señal grabada para ver si esta bien
  
```

%Grabación de la octava palabra

```

cierreNatalia= audiorecorder(Fs,8,1); %Grabar el audio con la Fs y canal mono
recordblocking(cierreNatalia,3.5);%Se graba 3.5 seg
y8=getaudiodata(cierreNatalia);%Se pasa el audio a formato para poder procesar
y8=silencio(y8); %Eliminación de ruido
fil8=enfasis(y8);%Filtro
segmentos8=segmentacion(fil8);%Segmentamos la señal en tramas de 30ms
cep8=cepstrum(segmentos8);%Se calculan los coeficientes cepstrum
coef8=coeficientes(cep8); %Se sacan los coeficientes característicos
save('cierreNatalia.mat','coef8','y8');%Se graba la señal de audio
play(cierreNatalia);%Reproducir la señal grabada para ver si esta bien
  
```

%grabación de la novena palabra

```

aperturaEmilio= audiorecorder(Fs,8,1); %Grabar el audio con la Fs y canal mono
recordingblock(aperturaEmilio,3.5);%Se graba 3.5 seg
y9=getaudiodata(aperturaEmilio);%Se pasa el audio a formato para poder procesar
y9=silencio(y9); %Eliminación de ruido
fil9=enfasis(y9);%Filtro
segmentos9=segmentacion(fil9);%Segmentamos la señal en tramas de 30ms
cep9=cepstrum(segmentos9);%Se calculan los coeficientes cepstrum
coef9=coeficientes(cep9); %Se sacan los coeficientes característicos
save('aperturaEmilio.mat','coef9','y9');%Se graba la señal de audio
play(aperturaEmilio);%Reproducir la señal grabada para ver si esta bien
  
```

%Grabación de la decima palabra

```

cierreEmilio= audiorecorder(Fs,8,1); %Grabar el audio con la Fs y canal mono
recordingblock(cierreEmilio,3.5);%Se graba 3.5 seg
y10=getaudiodata(cierreEmilio);%Se pasa el audio a formato para poder procesar
y10=silencio(y10); %Eliminación de ruido
fil10=enfasis(y10);%Filtro
segmentos10=segmentacion(fil10);%Segmentamos la señal en tramas de 30ms
cep10=cepstrum(segmentos10);%Se calculan los coeficientes cepstrum
coef10=coeficientes(cep10); %Se sacan los coeficientes característicos
save('cierreEmilio.mat','coef10','y10');%Se graba la señal de audio
play(cierreEmilio);%Reproducir la señal grabada para ver si esta bien
  
```

%grabación de la undecima palabra

```

aperturaEmilio1= audiorecorder(Fs,8,1); %Grabar el audio con la Fs y canal mono
recordingblock(aperturaEmilio1,3.5);%Se graba 3.5 seg
y11=getaudiodata(aperturaEmilio1);%Se pasa el audio a formato para poder procesar
y11=silencio(y11); %Eliminación de ruido
fil11=enfasis(y11);%Filtro
segmentos11=segmentacion(fil11);%Segmentamos la señal en tramas de 30ms
cep11=cepstrum(segmentos11);%Se calculan los coeficientes cepstrum
coef11=coeficientes(cep11); %Se sacan los coeficientes característicos
save('aperturaEmilio1.mat','coef11','y11');%Se graba la señal de audio
play(aperturaEmilio1);%Reproducir la señal grabada para ver si esta bien
  
```

%Grabación de la duodecima palabra

```

cierreEmilio1= audiorecorder(Fs,8,1); %Grabar el audio con la Fs y canal mono
recordingblock(cierreEmilio1,3.5);%Se graba 3.5 seg
y12=getaudiodata(cierreEmilio1);%Se pasa el audio a formato para poder procesar
y12=silencio(y12); %Eliminación de ruido
fil12=enfasis(y12);%Filtro
segmentos12=segmentacion(fil12);%Segmentamos la señal en tramas de 30ms
cep12=cepstrum(segmentos12);%Se calculan los coeficientes cepstrum
coef12=coeficientes(cep12); %Se sacan los coeficientes característicos
save('cierreEmilio1.mat','coef12','y12');%Se graba la señal de audio
play(cierreEmilio1);%Reproducir la señal grabada para ver si esta bien
  
```

%Grabación de la decimo tercer palabra

```

aperturaAngeles= audiorecorder(Fs,8,1); %Grabar el audio con la Fs y canal mono
recordingblock(aperturaAngeles,3.5);%Se graba 3.5 seg
y13=getaudiodata(aperturaAngeles);%Se pasa el audio a formato para poder procesar
y13=silencio(y13); %Eliminación de ruido
fil13=enfasis(y13);%Filtro
  
```

segmentos13=segmentacion(fil13);%Segmentamos la señal en tramas de 30ms
 cep13=cepstrum(segmentos13);%Se calculan los coeficientes cepstrum
 coef13=coeficientes(cep13); %Se sacan los coeficientes característicos
 save('aperturaAngeles.mat','coef13','y13');%Se graba la señal de audio
 play(aperturaAngeles);%Reproducir la señal grabada para ver si esta bien

%Grabación de la decimo cuarta palabra

cierreAngeles= audiorecorder(Fs,8,1); %Grabar el audio con la Fs y canal mono
 recordblocking(cierreAngeles,3.5);%Se graba 3.5 seg
 y14=getaudiodata(cierreAngeles);%Se pasa el audio a formato para poder procesar
 y14=silencio(y14); %Eliminación de ruido
 fil14=enfasis(y14);%Filtro
 segmentos14=segmentacion(fil14);%Segmentamos la señal en tramas de 30ms
 cep14=cepstrum(segmentos14);%Se calculan los coeficientes cepstrum
 coef14=coeficientes(cep14); %Se sacan los coeficientes característicos
 save('cierreAngeles.mat','coef14','y14');%Se graba la señal de audio
 play(cierreAngeles);%Reproducir la señal grabada para ver si esta bien

%Grabación de la palabra decimo quinta

aperturaLuis= audiorecorder(Fs,8,1); %Grabar el audio con la Fs y canal mono
 recordblocking(aperturaLuis,3.5);%Se graba 3.5 seg
 y15=getaudiodata(aperturaLuis);%Se pasa el audio a formato para poder procesar
 y15=silencio(y15); %Eliminación de ruido
 fil15=enfasis(y15);%Filtro
 segmentos15=segmentacion(fil15);%Segmentamos la señal en tramas de 30ms
 cep15=cepstrum(segmentos15);%Se calculan los coeficientes cepstrum
 coef15=coeficientes(cep15); %Se sacan los coeficientes característicos
 save('aperturaLuis.mat','coef15','y15');%Se graba la señal de audio
 play(aperturaLuis);%Reproducir la señal grabada para ver si esta bien

%Grabación de la decimo sexta palabra

cierreLuis= audiorecorder(Fs,8,1); %Grabar el audio con la Fs y canal mono
 recordblocking(cierreLuis,3.5);%Se graba 3.5 seg
 y16=getaudiodata(cierreLuis);%Se pasa el audio a formato para poder procesar
 y16=silencio(y16); %Eliminación de ruido
 fil16=enfasis(y16);%Filtro
 segmentos16=segmentacion(fil16);%Segmentamos la señal en tramas de 30ms
 cep16=cepstrum(segmentos16);%Se calculan los coeficientes cepstrum
 coef16=coeficientes(cep16); %Se sacan los coeficientes característicos
 save('cierreLuis.mat','coef16','y16');%Se graba la señal de audio
 play(cierreLuis);%Reproducir la señal grabada para ver si esta bien

%Grabación de la decimo septima palabra

aperturaAngel= audiorecorder(Fs,8,1); %Grabar el audio con la Fs y canal mono
 recordblocking(aperturaAngel,3.5);%Se graba 3.5 seg
 y17=getaudiodata(aperturaAngel);%Se pasa el audio a formato para poder procesar
 y17=silencio(y17); %Eliminación de ruido
 fil13=enfasis(y17);%Filtro
 segmentos17=segmentacion(fil17);%Segmentamos la señal en tramas de 30ms
 cep17=cepstrum(segmentos17);%Se calculan los coeficientes cepstrum
 coef17=coeficientes(cep17); %Se sacan los coeficientes característicos
 save('aperturaAngel.mat','coef17','y17');%Se graba la señal de audio
 play(aperturaAngel);%Reproducir la señal grabada para ver si esta bien

%Grabación de la decimo octava palabra

```

cierreAngel= audiorecorder(Fs,8,1); %Grabar el audio con la Fs y canal mono
recordblocking(cierreAngel,3.5);%Se graba 3.5 seg
y18=getaudiodata(cierreAngel);%Se pasa el audio a formato para poder procesar
y18=silencio(y18); %Eliminación de ruido
fil18=enfasis(y18);%Filtro
segmentos18=segmentacion(fil18);%Segmentamos la señal en tramas de 30ms
cep18=cepstrum(segmentos18);%Se calculan los coeficientes cepstrum
coef18=coeficientes(cep18); %Se sacan los coeficientes caracteristicos
save('cierreAngel.mat','coef18','y18');%Se graba la señal de audio
play(cierreAngel);%Reproducir la señal grabada para ver si esta bien
  
```

```

%Grabación de la decimo novena palabra
aperturaEmilio2= audiorecorder(Fs,8,1); %Grabar el audio con la Fs y canal mono
recordblocking(aperturaEmilio2,3.5);%Se graba 3.5 seg
y19=getaudiodata(aperturaEmilio2);%Se pasa el audio a formato para poder procesar
y19=silencio(y19); %Eliminación de ruido
fil19=enfasis(y19);%Filtro
segmentos19=segmentacion(fil19);%Segmentamos la señal en tramas de 30ms
cep19=cepstrum(segmentos19);%Se calculan los coeficientes cepstrum
coef19=coeficientes(cep19); %Se sacan los coeficientes caracteristicos
save('aperturaEmilio2.mat','coef19','y19');%Se graba la señal de audio
play(aperturaEmilio2);%Reproducir la señal grabada para ver si esta bien
  
```

```

%Grabación de la vigesima palabra
cierreEmilio2= audiorecorder(Fs,8,1); %Grabar el audio con la Fs y canal mono
recordblocking(cierreEmilio2,3.5);%Se graba 3.5 seg
y20=getaudiodata(cierreEmilio2);%Se pasa el audio a formato para poder procesar
y20=silencio(y20); %Eliminación de ruido
fil20=enfasis(y20);%Filtro
segmentos20=segmentacion(fil20);%Segmentamos la señal en tramas de 30ms
cep20=cepstrum(segmentos20);%Se calculan los coeficientes cepstrum
coef20=coeficientes(cep20); %Se sacan los coeficientes caracteristicos
save('cierreEmilio2.mat','coef20','y20');%Se graba la señal de audio
play(cierreEmilio2);%Reproducir la señal grabada para ver si esta bien
  
```

```

% Eliminación de ruido que no aporta nada a la señal
% y= variable donde se almacena los valores que no son silencios
% sil= señal adquirida pcon audio recorder
  
```

```
function y = silencio(sil)
```

```

longitud = length(sil); % longitud del vector
pico=max(abs(sil));%Obtener el pico máximo
sil=sil/pico;%Normalización de la señal
promedio_senal = sum(sil.*sil)/longitud; %promedio señal entera
THRES = 0.2; %Valor UMBRAL que usa para comparar
y = 0;
for i = 1:80:longitud-80 % cada 10ms
    segmentos = sil(i:i+79); % segmentos
    promedio_energ = sum(segmentos.*segmentos)/80; % promedio de cada segmento
    if( promedio_energ> THRES*promedio_senal) % si el promedio energetico es
mayor que la señal %completa por el valor umbral
        y=[y;segmentos(1:end)]; %lo almacena y sino es eliminado como espacio en blanco
    end
  
```

```

end
%Filtro pre-enfásis por donde pasa la señal
% y= señal filtrada
% x= señal a filtrar

function [y]=enfasis(x)
%Parámetros usados en el filtro
b=[1 -0.95];
A=1;
%Usamos la función filter definida en matlab
y=filter(b,A,x);

%Funcion segmentación. Se segmenta la señal en tramas de 30 ms
% seg= segmentos retornados
% y= resultado de la señal pasada por el filtro

function[seg]=segmentacion(y)

longitud=length(y); %Logintud del vector
numero_segmentos=floor (longitud/80)-2; %El número de segmentos se redondea

seg=zeros(numero_segmentos,240)'; %Matriz de segmentos, numero_seg 240
ventana=hamming(240);%Ventana de Hamming de 30ms

for i= 0:numero_segmentos-1
    inicio=i*80+1;
    seg(i+1,1:240)=(y(inicio:inicio+239).*ventana);%Se multiplica por una ventana
    hamming
end

% Calcular los coeficientes cepstrales de cada segmento
% Segmentos= segmentos que devuelve la funcion segmentacion
% c= coeficientes cepstrum

function c=cepstrum(segmentos)

[M,N]=size(segmentos);%Matriz de Mfilas y N columnas
%Frecuencias a las que se aplica FFT
c=zeros(M,10); % Se inicializa el vector para que se guarden los coeficientes
cepstrales,P=10

%Calculamos los coeficientes cepstrales
for i=1:M
    f = fft(segmentos(i,:));
    m = abs(f);
    l = log(m+1e-5);
    r = real(ifft(l));
    % r=rceps(segundos(i,:));
    c(i,:)=r(1:10);
end
  
```

```

%Se normaliza el valor de los coeficientes cepstrales
suma_c=sum(c);
promedio_c=suma_c/M;

%Se coge el mínimo
for i=1:M
c(i,:)=c(i,:)-promedio_c;
end

%Se obtienen los coeficientes polinomiales de los coeficientes cepstrum
%c coeficientes cepstrum devueltos por la funcion cepstrum
% coef coeficientes polinomiales

function coef= coeficientes(c);

[M,N]=size(c);
coef=zeros(M-8,20); %Valores minimo 20

%Creamos columnas del vector j y se recorre por filas

i= (1:9);
%Crear una matriz llena de coeficientes P1j
P0=repmat((i-5)',1,10);
sum_P0= sum((i-5).*(i-5)); %Se calcula la suma total

for j=1:M-8
  %coeficientes cepstrales
  coef(j,1:10)=c(j,1:10);
  b=zeros(1,10);
  %Coeficientes polinomiales para cada segmento
  for i=0:8
    b=b+P0(i+1,:).*c(j+i,:);
  end
  coef(j,11:20)=b/sum_P0;
end

%Calculo de la distancia de los parámetros característicos. Donde se
%utiliza la distancia Euclídea
%f_1 coeficientes de la matriz guardada, f_2 coeficientes de la matriz
%característica
%dist vector que guarda las distancias de la matriz característica con
%respecto a la matriz de los patrones guardados

function dist = distancia(f_1,f_2)

%tamaño de las matrices
[m_1,n_1]=size(f_1);
[m_2,n_2]=size(f_2);

%establece the guide and slave
if(m_1 < m_2 )
  guide = f_1;
  slave = f_2;

```

```

m = m_2;
n = n_1;
else
guide = f_2;
slave = f_1;
m = m_1;
n = n_2;
end

%Computa la matriz de distancia . Fórmula del calculo de la distancia
dist= zeros(m,n);
for i = 1:m
for j = 1:n
dist(i,j) = sqrt(sum(((guide(j,:)-slave(i,:)).*(guide(j,:)-slave(i,:)).^2)));
end
end

```

% Interfaz gráfica del programa de reconocimiento de voz

```

function varargout = Interfazfig(varargin)
% INTERFAZFIG MATLAB code for Interfazfig.fig
% INTERFAZFIG, by itself, creates a new INTERFAZFIG or raises the existing
% singleton*.
%
% H = INTERFAZFIG returns the handle to a new INTERFAZFIG or the handle to
% the existing singleton*.
%
% INTERFAZFIG('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in INTERFAZFIG.M with the given input arguments.
%
% INTERFAZFIG('Property','Value',...) creates a new INTERFAZFIG or raises the
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before Interfazfig_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to Interfazfig_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Interfazfig

% Last Modified by GUIDE v2.5 01-Dec-2017 19:50:27

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
'gui_Singleton', gui_Singleton, ...
'gui_OpeningFcn', @Interfazfig_OpeningFcn, ...
'gui_OutputFcn', @Interfazfig_OutputFcn, ...
'gui_LayoutFcn', [], ...
'gui_Callback', []);
if nargin && ischar(varargin{1})
gui_State.gui_Callback = str2func(varargin{1});

```

```

end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
%-----
% Se construye la interfaz gráfica y realizamos las funciones que van
% dentro del programa principal
%-----

% --- Executes just before Interfazfig is made visible.
function Interfazfig_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Interfazfig (see VARARGIN)

% Se cargan los audios que se han creao en la BBDD
load aperturaSilvia.mat;
load cierreSilvia.mat;
load aperturaEduardo.mat;
load cierreEduardo.mat;
load aperturaPilar.mat;
load cierrePilar.mat;
load aperturaNatalia.mat;
load cierreNatalia.mat;
load aperturaEmilio.mat;
load cierreEmilio.mat;
load aperturaEmilio1.mat;
load cierreEmilio1.mat;
load aperturaAngeles.mat;
load cierreAngeles.mat;
load aperturaLuis.mat;
load cierreLuis.mat;
load aperturaAngel.mat;
load cierreAngel.mat;
load aperturaEmilio2.mat;
load cierreEmilio2.mat;

%Se asignan los ejes a la imagen

axes(handles.axes1);
title('GRÁFICA DE SEÑAL DE VOZ GRABADA');
xlabel('Muestras de voz');
ylabel('Nivel de la voz');
axes(handles.axes2);
title('GRÁFICA DE SEÑAL DE VOZ BBDD');
xlabel('Muestras de voz');
ylabel('Nivel de la voz');
axes(handles.axes4);
imag = imread('logo.jpg');

```

```

imshow(imag);
axis off;

```

%Se cargan en la interfaz gráfica todos los valores característicos de las %señales de audio que están en la base de datos

```

handles.coeficientes_y1=y1;
handles.coeficientes_y2=y2;
handles.coeficientes_y3=y3;
handles.coeficientes_y4=y4;
handles.coeficientes_y5=y5;
handles.coeficientes_y6=y6;
handles.coeficientes_y7=y7;
handles.coeficientes_y8=y8;
handles.coeficientes_y9=y9;
handles.coeficientes_y10=y10;
handles.coeficientes_y11=y11;
handles.coeficientes_y12=y12;
handles.coeficientes_y13=y13;
handles.coeficientes_y14=y14;
handles.coeficientes_y15=y15;
handles.coeficientes_y16=y16;
handles.coeficientes_y17=y17;
handles.coeficientes_y18=y18;
handles.coeficientes_y19=y19;
handles.coeficientes_y20=y20;

```

```

handles.coeficientes_coef1=coef1;
handles.coeficientes_coef2=coef2;
handles.coeficientes_coef3=coef3;
handles.coeficientes_coef4=coef4;
handles.coeficientes_coef5=coef5;
handles.coeficientes_coef6=coef6;
handles.coeficientes_coef7=coef7;
handles.coeficientes_coef8=coef8;
handles.coeficientes_coef9=coef9;
handles.coeficientes_coef10=coef10;
handles.coeficientes_coef11=coef11;
handles.coeficientes_coef12=coef12;
handles.coeficientes_coef13=coef13;
handles.coeficientes_coef14=coef14;
handles.coeficientes_coef15=coef15;
handles.coeficientes_coef16=coef16;
handles.coeficientes_coef17=coef17;
handles.coeficientes_coef18=coef18;
handles.coeficientes_coef19=coef19;
handles.coeficientes_coef20=coef20;

```

% Choose default command line output for Interfazfig
 handles.output = hObject;

% Update handles structure
 guidata(hObject, handles);

```

% UIWAIT makes Interfazfig wait for user response (see UIRESUME)
% uiwait(handles.figure1);
  
```

```

% --- Outputs from this function are returned to the command line.
function varargout = Interfazfig_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
  
```

```

% Get default command line output from handles structure
varargout{1} = handles.output;
%Se pulsa el primer boton que realiza el grabado de audio
% --- Executes on button press in Grabaraudio.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
  
```

```

Fs=44100; %Frecuencia de muestreo
%Se graba la voz a reconocer
audio= audiorecorder(Fs,8,1); %Grabar el audio con la Fs y canal mono
recordblocking(audio,3.5);%Se graba durante un período de 3.5seg
y=getaudiodata(audio);%Se transforma la señal de audio en una señal que pueda
ser procesada
  
```

```

y=silencio(y); %Eliminación de ruido
filtro=enfasis(y);
segmentos=segmentacion(filtro);%Filtro
cep=cepstrum(segmentos);
coef=coeficientes(cep);
save('reconocer.mat','coef','y');
  
```

```

%Se guardan los parámetros característicos de la voz a reconocer
handles.audio_grabado=y;
handles.coef_caracteristicos=coef;
axes(handles.axes1);
plot(y);
msgbox('Hemos terminado la grabación');
  
```

```

guidata(hObject, handles);
  
```

```

% --- Executes on button press in IdentificarPersona.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
  
```

```

%coeficientes característicos de la señal de voz a reconocer
y=handles.audio_grabado;
coef= handles.coef_caracteristicos;
%coeficientes característicos de las señales de voz de la bbdd
  
```

```
y1=handles.coeficientes_y1;  
y2=handles.coeficientes_y2;  
y3=handles.coeficientes_y3;  
y4=handles.coeficientes_y4;  
y5=handles.coeficientes_y5;  
y6=handles.coeficientes_y6;  
y7=handles.coeficientes_y7;  
y8=handles.coeficientes_y8;  
y9=handles.coeficientes_y9;  
y10=handles.coeficientes_y10;  
y11=handles.coeficientes_y11;  
y12=handles.coeficientes_y12;  
y13=handles.coeficientes_y13;  
y14=handles.coeficientes_y14;  
y15=handles.coeficientes_y15;  
y16=handles.coeficientes_y16;  
y17=handles.coeficientes_y17;  
y18=handles.coeficientes_y18;  
y19=handles.coeficientes_y19;  
y20=handles.coeficientes_y20;
```

```
coef1=handles.coeficientes_coef1;  
coef2=handles.coeficientes_coef2;  
coef3=handles.coeficientes_coef3;  
coef4=handles.coeficientes_coef4;  
coef5=handles.coeficientes_coef5;  
coef6=handles.coeficientes_coef6;  
coef7=handles.coeficientes_coef7;  
coef8=handles.coeficientes_coef8;  
coef9=handles.coeficientes_coef9;  
coef10=handles.coeficientes_coef10;  
coef11=handles.coeficientes_coef11;  
coef12=handles.coeficientes_coef12;  
coef13=handles.coeficientes_coef13;  
coef14=handles.coeficientes_coef14;  
coef15=handles.coeficientes_coef15;  
coef16=handles.coeficientes_coef16;  
coef17=handles.coeficientes_coef17;  
coef18=handles.coeficientes_coef18;  
coef19=handles.coeficientes_coef19;  
coef20=handles.coeficientes_coef20;
```

%Se realiza la comparación

```
d1=distancia(coef1,coef);  
d2=distancia(coef2,coef);  
d3=distancia(coef3,coef);  
d4=distancia(coef4,coef);  
d5=distancia(coef5,coef);  
d6=distancia(coef6,coef);  
d7=distancia(coef7,coef);  
d8=distancia(coef8,coef);  
d9=distancia(coef9,coef);  
d10=distancia(coef10,coef);
```

```

d11=distancia(coef11,coef);
d12=distancia(coef12,coef);
d13=distancia(coef13,coef);
d14=distancia(coef14,coef);
d15=distancia(coef15,coef);
d16=distancia(coef16,coef);
d17=distancia(coef17,coef);
d18=distancia(coef18,coef);
d19=distancia(coef19,coef);
d20=distancia(coef20,coef);
  
```

% Se obtienen las medidas de las matrices

```

[m1,n1]=size(d1);
[m2,n2]=size(d2);
[m3,n3]=size(d3);
[m4,n4]=size(d4);
[m5,n5]=size(d5);
[m6,n6]=size(d6);
[m7,n7]=size(d7);
[m8,n8]=size(d8);
[m9,n9]=size(d9);
[m10,n10]=size(d10);
[m11,n11]=size(d11);
[m12,n12]=size(d12);
[m13,n13]=size(d13);
[m14,n14]=size(d14);
[m15,n15]=size(d15);
[m16,n16]=size(d16);
[m17,n17]=size(d17);
[m18,n18]=size(d18);
[m19,n19]=size(d19);
[m20,n20]=size(d20);
  
```

%Se recorre cada matriz para sacar su valor de distancia mínima que es el %que se necesita para comparar

```

t1=1;
for i=1:m1:size(d1,1)
    for j=i:i+n1-1
        k=d1(j,:);
        m(j)=min(k);
    end
    D1(t1)=min(m);
    t1=t1+1;
end
display(D1)
  
```

```

t2=1;
for i=1:m2:size(d2,1)
    for j=i:i+n2-1
        k=d2(j,:);
        m(j)=min(k);
    end
    D2(t2)=min(m);
    t2=t2+1;
  
```

```
end
display(D2)

t3=1;
for i=1:m3:size(d3, 1)
    for j=i:i+n3-1
        k=d3(j,:);
        m(j)=min(k);
    end
    D3(t3)=min(m);
    t3=t3+1;
end
display(D3)

t4=1;
for i=1:m4:size(d4, 1)
    for j=i:i+n4-1
        k=d4(j,:);
        m(j)=min(k);
    end
    D4(t4)=min(m);
    t4=t4+1;
end
display(D4)

t5=1;
for i=1:m5:size(d5, 1)
    for j=i:i+n5-1
        k=d5(j,:);
        m(j)=min(k);
    end
    D5(t5)=min(m);
    t5=t5+1;
end
display(D5)

t6=1;
for i=1:m6:size(d6, 1)
    for j=i:i+n6-1
        k=d6(j,:);
        m(j)=min(k);
    end
    D6(t6)=min(m);
    t6=t6+1;
end
display(D6)

t7=1;
for i=1:m7:size(d7, 1)
    for j=i:i+n7-1
        k=d7(j,:);
        m(j)=min(k);
```

```

    end
    D7(t7)=min(m);
    t7=t7+1;
  end
  display(D7)

  t8=1;
  for i=1:m8:size(d8,1)
    for j=i:i+n8-1
      k=d8(j,:);
      m(j)=min(k);
    end
    D8(t8)=min(m);
    t8=t8+1;
  end
  display(D8)

  t9=1;
  for i=1:m9:size(d9,1)
    for j=i:i+n9-1
      k=d9(j,:);
      m(j)=min(k);
    end
    D9(t9)=min(m);
    t9=t9+1;
  end
  display(D9)

  t10=1;
  for i=1:m10:size(d10,1)
    for j=i:i+n10-1
      k=d10(j,:);
      m(j)=min(k);
    end
    D10(t10)=min(m);
    t10=t10+1;
  end
  display(D10)

  t11=1;
  for i=1:m11:size(d11,1)
    for j=i:i+n11-1
      k=d11(j,:);
      m(j)=min(k);
    end
    D11(t11)=min(m);
    t11=t11+1;
  end
  display(D11)

  t12=1;
  for i=1:m12:size(d12,1)
    for j=i:i+n12-1
      k=d12(j,:);

```

```

    m(j)=min(k);
  end
  D12(t12)=min(m);
  t12=t12+1;
end
display(D12)

t13=1;
for i=1:m13:size(d13,1)
  for j=i:i+n13-1
    k=d13(j,:);
    m(j)=min(k);
  end
  D13(t13)=min(m);
  t13=t13+1;
end
display(D13)

t14=1;
for i=1:m12:size(d14,1)
  for j=i:i+n14-1
    k=d14(j,:);
    m(j)=min(k);
  end
  D14(t14)=min(m);
  t14=t14+1;
end
display(D14)

t15=1;
for i=1:m15:size(d15,1)
  for j=i:i+n15-1
    k=d15(j,:);
    m(j)=min(k);
  end
  D15(t15)=min(m);
  t15=t15+1;
end
display(D15)

t16=1;
for i=1:m16:size(d16,1)
  for j=i:i+n16-1
    k=d16(j,:);
    m(j)=min(k);
  end
  D16(t16)=min(m);
  t16=t16+1;
end
display(D16)

t17=1;
for i=1:m17:size(d17,1)
  for j=i:i+n17-1
    k=d17(j,:);

```

```

    m(j)=min(k);
  end
  D17(t17)=min(m);
  t17=t17+1;
end
display(D17)

t18=1;
for i=1:m18:size(d18,1)
  for j=i:i+n18-1
    k=d18(j,:);
    m(j)=min(k);
  end
  D18(t18)=min(m);
  t18=t18+1;
end
display(D18)

t19=1;
for i=1:m19:size(d19,1)
  for j=i:i+n19-1
    k=d19(j,:);
    m(j)=min(k);
  end
  D19(t19)=min(m);
  t19=t19+1;
end
display(D19)

t20=1;
for i=1:m20:size(d20,1)
  for j=i:i+n20-1
    k=d20(j,:);
    m(j)=min(k);
  end
  D20(t20)=min(m);
  t20=t20+1;
end
display(D18)

```

%Se realizan las comparaciones con cada usuario para ver cual es la %distancias menor

```

  if D1<D2 && D1<D3 && D1<D4 && D1< D5 && D1<D6 && D1<D7 && D1<D8 &&
  D1<D9 && D1<D10 && D1<D11 && D1<D12 && D1<D13 && D1<D14 && D1<D15
  && D1<D16 && D1<D17 && D1<D18 && D1<D19 && D1<D20 && D1<0.006
    axes(handles.axes2);
    plot(y);
    axes(handles.axes2);
    plot(y1);
    msgbox('Identificado: aperturaSilvia');
  elseif D2<D1 && D2<D3 && D2< D4 && D2<D5 && D2<D6 && D2<D7 && D2<D8
  && D2<D9 && D2<D10 && D2<D11 && D2<D12 && D2<D13 && D2<D14 && D2<D15
  && D2<D16 && D2<D17 && D2<D18 && D2<D19 && D2<D20 && D2<0.01

```

```

axes(handles.axes2);
plot(y);
axes(handles.axes2);
plot(y2);
msgbox('Identificado: CierreSilvia');
elseif D3<D1 && D3<D2 && D3<D4 && D3<D5 && D3<D6 && D3<D7 && D3<D8
&& D3<D9 && D3<D10 && D3<D11 && D3<D12 && D3<D13 && D3<D14 && D3<D15
&& D3<D16 && D3<D17 && D3<D18 && D3<D19 && D3<D20 && D3<0.01
axes(handles.axes2);
plot(y);
axes(handles.axes2);
plot(y3);
msgbox('Identificado: AperturaEduardo');
elseif D4<D1 && D4<D2 && D4<D3 && D4<D5 && D4<D6 && D4<D7 && D4<D8
&& D4<D9 && D4<D10 && D4<D11 && D4<D12 && D4<D13 && D4<D14 && D4<D15
&& D4<D16 && D4<D17 && D4<D18 && D4<D19 && D4<D20 && D4<0.2
axes(handles.axes2);
plot(y);
axes(handles.axes2);
plot(y4);
msgbox('Identificado: cierreEduardo');
elseif D5<D1 && D5<D2 && D5<D3 && D5<D4 && D5<D6 && D5<D7 && D5<D8
&& D5<D9 && D5<D10 && D5<D11 && D5<D12 && D5<D13 && D5<D14 && D5<D15
&& D5<D16 && D5<D17 && D5<D18 && D5<D19 && D5<D20 && D5<0.1
axes(handles.axes2);
plot(y);
axes(handles.axes2);
plot(y5);
msgbox('Identificado: AperturaPilar');
elseif D6<D1 && D6<D2 && D6<D3 && D6<D4 && D6<D5 && D6<D7 && D6<D8
&& D6<D9 && D6<D10 && D6<D11 && D6<D12 && D6<D13 && D6<D14 && D6<D15
&& D6<D16 && D6<D17 && D6<D18 && D6<D19 && D6<D20 && D6<0.1
axes(handles.axes2);
plot(y);
axes(handles.axes2);
plot(y6);
msgbox('Identificado: cierrePilar');
elseif D7<D1 && D7<D2 && D7<D3 && D7<D4 && D7<D5 && D7<D6 && D7<D8
&& D7<D9 && D7<D10 && D7<D11 && D7<D12 && D7<D13 && D7<D14 && D7<D15
&& D7<D16 && D7<D17 && D7<D18 && D7<D19 && D7<D20 && D7<0.1
axes(handles.axes2);
plot(y);
axes(handles.axes2);
plot(y7);
msgbox('Identificado: AperturaNatalia');
elseif D8<D1 && D8<D2 && D8<D3 && D8<D4 && D8<D5 && D8<D6 && D8<D7
&& D8<D9 && D8<D10 && D8<D11 && D8<D12 && D8<D13 && D8<D14 && D8<D15
&& D8<D16 && D8<D17 && D8<D18 && D8<D19 && D8<D20 && D8< 0.25
axes(handles.axes2);
plot(y);
axes(handles.axes2);
plot(y8);
msgbox('Identificado: cierreNatalia');

```

```

elseif D9<D1 && D9<D2 && D9<D3 && D9<D4 && D9<D5 && D9<D6 && D9<D7
&& D9<D8 && D9<D10 && D9<D11 && D9<D12 && D9<D13 && D9<D14 && D9<D15
&& D9<D16 && D9<D17 && D9<D18 && D9<D19 && D9<D20 && D9< 0.1
    axes(handles.axes2);
    plot(y);
    axes(handles.axes2);
    plot(y9);
    msgbox('Identificado: AperturaEmilio');
elseif D10<D1 && D10<D2 && D10<D3 && D10<D4 && D10<D5 && D10<D6 &&
D10<D7 && D10<D8 && D10<D9 && D10<D11 && D10<D12 && D10<D13 &&
D10<D14 && D10<D15 && D10<D16 && D10<D17 && D10<D18 && D10<D19 &&
D10<D20 && D10>0.01
    axes(handles.axes2);
    plot(y);
    axes(handles.axes2);
    plot(y10);
    msgbox('Identificado: cierreEmilio');
elseif D11<D1 && D11<D2 && D11<D3 && D11<D4 && D11<D5 && D11<D6 &&
D11<D7 && D11<D8 && D11<D9 && D11<D10 && D11<D12 && D11<D13 &&
D11<D14 && D11<D15 && D11<D16 && D11<D17 && D11<D18 && D11<D19 &&
D11<D20 && D11>0.01
    axes(handles.axes2);
    plot(y);
    axes(handles.axes2);
    plot(y11);
    msgbox('Identificado: AperturaEmilio1');
elseif D12<D1 && D12<D2 && D12<D3 && D12<D4 && D12<D5 && D12<D6 &&
D12<D7 && D12<D8 && D12<D9 && D12<D10 && D12<D11 && D12<D13 &&
D12<D14 && D12<D15 && D12<D16 && D12<D17 && D12<D18 && D12<D19 &&
D12<D20 && D12>0.1
    axes(handles.axes2);
    plot(y);
    axes(handles.axes2);
    plot(y12);
    msgbox('Identificado: CierreEmilio1');
elseif D13<D1 && D13<D2 && D13<D3 && D13<D4 && D13<D5 && D13<D6 &&
D13<D7 && D13<D8 && D13<D9 && D13<D10 && D13<D11 && D13<D12 &&
D13<D14 && D13<D15 && D13<D16 && D13<D17 && D13<D18 && D13<D19 &&
D13<D20 && D13>0.01
    axes(handles.axes2);
    plot(y);
    axes(handles.axes2);
    plot(y13);
    msgbox('Identificado: AperturaAngeles');
elseif D14<D1 && D14<D2 && D14<D3 && D14<D4 && D14<D5 && D14<D6 &&
D14<D7 && D14<D8 && D14<D9 && D14<D10 && D14<D11 && D14<D12 &&
D14<D13 && D14<D15 && D14<D16 && D14<D17 && D14<D18 && D14<D19 &&
D14<D20 && D14>0.01
    axes(handles.axes2);
    plot(y);
    axes(handles.axes2);
    plot(y14);
    msgbox('Identificado: cierreAngeles');
elseif D15<D1 && D15<D2 && D15<D3 && D15<D4 && D15<D5 && D15<D6 &&
D15<D7 && D15<D8 && D15<D9 && D15<D10 && D15<D11 && D15<D13 &&

```

```

D15<D14 && D15<D16 && D15<D17 && D15<D18 && D15<D19 && D15<D20 &&
D15<D11 && D15>0.1
    axes(handles.axes2);
    plot(y);
    axes(handles.axes2);
    plot(y15);
    msgbox('Identificado: aperturaLuis');
    elseif D16<D1 && D16<D2 && D16<D3 && D16<D4 && D16<D5 && D16<D6 &&
D16<D7 && D16<D8 && D16<D9 && D16<D10 && D16<D11 && D16<D12 &&
D15<D13 && D15<D14 && D15<D16 && D15<D17 && D14<D18 && D14<D19 &&
D14<D20 && D14>0.01
    axes(handles.axes2);
    plot(y);
    axes(handles.axes2);
    plot(y16);
    msgbox('Identificado: cierreLuis');
    elseif D17<D1 && D17<D2 && D17<D3 && D17<D4 && D17<D5 && D17<D6 &&
D17<D7 && D17<D8 && D17<D9 && D17<D10 && D17<D11 && D17<D12 &&
D17<D13 && D17<D14 && D17<D16 && D15<D18 && D15<D19 && D15<D20 &&
D15<D11 && D15>0.2
    axes(handles.axes2);
    plot(y);
    axes(handles.axes2);
    plot(y17);66
    msgbox('Identificado: aperturaAngel');
    elseif D18<D1 && D18<D2 && D18<D3 && D18<D4 && D18<D5 && D18<D6 &&
D18<D7 && D18<D8 && D18<D9 && D18<D10 && D18<D11 && D18<D12 &&
D18<D13 && D18<D14 && D18<D15 && D18<D16 && D18<D17 && D18<D19 &&
D118<D20 && D14>0.01
    axes(handles.axes2);
    plot(y);
    axes(handles.axes2);
    plot(y18);
    msgbox('Identificado: cierreAngel');
    elseif D19<D1 && D19<D2 && D19<D3 && D19<D4 && D19<D5 && D19<D6 &&
D19<D7 && D19<D8 && D19<D9 && D19<D10 && D19<D11 && D19<D12 &&
D19<D13 && D19<D14 && D19<D15 && D19<D16 && D19<D18 && D19<D20 &&
D19>0.25
    axes(handles.axes2);
    plot(y);
    axes(handles.axes2);
    plot(y19);
    msgbox('Identificado: aperturaEmilio2');
    elseif D20<D1 && D20<D2 && D20<D3 && D20<D4 && D20<D5 && D20<D6 &&
D20<D7 && D20<D8 && D20<D9 && D20<D10 && D20<D11 && D20<D12 &&
D20<D13 && D20<D14 && D20<D15 && D20<D16 && D20<D17 && D20<D18 &&
D20<D19 && D20>0.02
    axes(handles.axes2);
    plot(y);
    axes(handles.axes2);
    plot(y20);
    msgbox('Identificado: cierreEmilio2');
    else
        msgbox('No identificado');
    end
  
```

```
guidata(hObject, handles);
```

```
% --- Executes on button press in salir.
```

```
function pushbutton3_Callback(hObject, eventdata, handles)
```

```
% hObject handle to pushbutton3 (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
```

```
% handles structure with handles and user data (see GUIDATA)
```

```
msgbox('Gracias por usar la aplicación. Hasta pronto');
```

```
close(gcf)
```

7. GLOSARIO

Autenticación: Autorizar o legalizar algo.

Biometría: Estudio mensurativo o estadístico de los fenómenos o procesos biológicos.

Coeficientes: Expresión numérica de una propiedad o característica de un cuerpo, que generalmente se presenta como una relación entre dos magnitudes

Distancia: Diferencia, desemejanza notable entre unas cosas y otras.

Filtro: Dispositivo que elimina o selecciona ciertas frecuencias de un espectro eléctrico, acústico, óptico o mecánico.

Frecuencia: Número de veces que se repite un proceso periódico por unidad de tiempo.

Identificación: Reconocer si una persona o cosa es la misma que se supone o se busca.

Interfaz: Conexión o frontera común entre dos aparatos o sistemas independientes.

Implementar: Poner en funcionamiento o aplicar métodos, medidas, etc., para llevar algo a cabo.

Matriz: Conjunto de números o símbolos algebraicos colocados en líneas horizontales y verticales y dispuestos en forma de rectángulo.

Microfono: Aparato que transforma las ondas sonoras en corrientes eléctricas para su amplificación.

Procesar: Someter datos o materiales a una serie de operaciones programadas.

Segmento: Porción o parte cortada o separada de una cosa, de un elemento geométrico o de un todo.

Seguridad: Situación de tranquilidad pública y de libre ejercicio de los derechos individuales, cuya protección efectiva se encomienda a las fuerzas de orden público.

Señal: Variación de una corriente eléctrica u otra magnitud que se utiliza para transmitir información.

Sistema: Conjunto de reglas o principios sobre una materia racionalmente enlazados entre sí.

Voz: Sonido producido por la vibración de las cuerdas vocales.