REPÚBLICA DEL ECUADOR
Escuela Politécnica Nacional
"E SCIENTIA HOMINIS SALUS"

La versión digital de esta tesis está protegida por la Ley de Derechos de Autor del Ecuador.

Los derechos de autor han sido entregados a la "ESCUELA POLITÉCNICA NACIONAL" bajo el libre consentimiento del (los) autor(es).

Al consultar esta tesis deberá acatar con las disposiciones de la Ley y las siguientes condiciones de uso:

- Cualquier uso que haga de estos documentos o imágenes deben ser sólo para efectos de investigación o estudio académico, y usted no puede ponerlos a disposición de otra persona.
- Usted deberá reconocer el derecho del autor a ser identificado y citado como el autor de esta tesis.
- No se podrá obtener ningún beneficio comercial y las obras derivadas tienen que estar bajo los mismos términos de licencia que el trabajo original.

El Libre Acceso a la información, promueve el reconocimiento de la originalidad de las ideas de los demás, respetando las normas de presentación y de citación de autores con el fin de no incurrir en actos ilegítimos de copiar y hacer pasar como propias las creaciones de terceras personas.

Respeto hacia sí mismo y hacia los demás.

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DEPURACIÓN DEL CONJUNTO DE DATOS PVA-DRYRUN

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERA EN ELECTRÓNICA Y TELECOMUNICACIONES

JESSICA MAGALI FUERTES CAÑAR

jessica.fuertes@epn.edu.ec

DIRECTOR: JOSÉ ADRIÁN ZAMBRANO MIRANDA MS.c.

jose.zambrano@epn.edu.ec

Quito, febrero 2020

AVAL

Certifico que el presente trabajo fue desarrollado por Jessica Magali Fuertes Cañar, bajo mi supervisión.

JOSÉ ADRIÁN ZAMBRANO MIRANDA MS.c.
DIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo, Jessica Magali Fuertes Cañar, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración dejo constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.

JESSICA MAGALI FUERTES CAÑAR

DEDICATORIA

A la persona más importante en mi vida, mi madre, quien ha sido el pilar fundamental para que pueda culminar esta meta. Con su amor y apoyo incondicional siempre me ha motivado a continuar y a ser mejor cada día.

Soy la persona más afortunada y bendecida por tenerla.

AGRADECIMIENTO

A mi madre por su lucha constante, para regalarme la posibilidad de que pueda cumplir este sueño, a mi ángel, mi papito que desde donde este espero que se sienta muy orgulloso de mí.

A mis hermanos por darme el mejor regalo del mundo, mis sobrinas, paulita, valita y natita, ellas llenan de alegría mi vida.

A mi jesuita, que con su amor y locura, hace que me olvide de los problemas por un ratito.

A Jeampool, mi amor bonito, quien me ha apoyado siempre, en las buenas y malas, y ha sido una persona demasiado importante en este proceso.

A mi mejor amigo Cali, por tantas experiencias maravillosas que hemos compartido, tantas risas y ocurrencias.

Al MSc. Adrián Zambrano, por guiarme en la realización del presente trabajo, por su apoyo y paciencia para poder culminar este proceso.

Al Dr. Felipe Grijalva, por brindarme su conocimiento, que ha sido de mucha ayuda en este proyecto.

ÍNDICE DE CONTENIDO

AVAL I

DECLARACIÓN DE AUTORÍA II

DEDICATORIA III

AGRADECIMIENTO IV

ÍNDICE DE CONTENIDO V

RESUMEN VIII

ABSTRACT IX

1. INTRODUCCIÓN 1

1.1 OBJETIVOS 4

1.1.1 OBJETIVO GENERAL 4

1.1.2 OBJETIVOS ESPECÍFICOS 4

1.2 ALCANCE 4

1.3 MARCO TEÓRICO 5

1.3.1 TRASTORNOS DEL HABLA Y LA VOZ EN LA EP 5

1.3.1.1. Fundamentos sobre la Enfermedad de Parkinson 5

1.3.1.2 Sistema Fonatorio 6

1.3.1.3 Relación existente entre la EP y la voz 8

1.3.2 MEDIDAS DE DISFONÍA 8

1.3.2.1 Conceptos matemáticos recurrentes 8

1.3.2.2 Fonación de vocales sostenidas 9

1.3.2.3 Estimación de la frecuencia fundamental 10

1.3.2.3.1 Praat 11

1.3.2.3.2 Shrp 11

1.3.2.3.3 Swipe 12

1.3.2.3.4 Dypsa 12

1.3.2.4 Medidas de disfonía clásicas 13

1.3.2.4.1 Jitter 13
1.3.2.4.2 Shimmer 14
1.3.2.4.3 HNR y NHR 14
1.3.2.4.4 Mel Frequency Cepstral Coefficients (MFCC) 14
1.3.2.5 Medidas de disfonía modernas 16
1.3.2.5.1 Glottal to noise excitation ratio (GNE) 16
1.3.2.5.2 Detrended Fluctuation Analysis (DFA) 17
1.3.2.5.3 Recurrence Period Density Entropy (RPDE) 17
1.3.2.5.4 Pitch Period Entropy (PPE) 17
1.3.2.6 Medidas de disfonía nuevas 18
1.3.2.6.1 Wavelet 18
1.3.2.6.2 Empirical Mode Decomposition Excitation Ratios (EMD-ER) ... 19
1.3.2.6.3 Vocal Fold Excitation Ratios (VFER) 20
1.3.2.6.4 Glottal Quotient (GQ) 20
2. METODOLOGÍA 21
2.1. DESCRIPCIÓN DE LA BASE DE DATOS Y DEL TOOLBOX 21
2.1.1 Base de Datos PVA-DRYRUN 21
2.1.2 Toolbox Voice Analysis 23
2.2. EXTRACCIÓN DE LAS FÓRMULAS MATEMÁTICAS 24
2.2.1 SCRIPT voice_analysis.m 33
2.2.1.1 Jitter y Shimmer 33
2.2.1.2 Harmonics to Noise Ratio (HNR) y Noise to Harmonics Ratio (NHR).... 40
2.2.1.3 Glottal Quotient (GQ) 42
2.2.1.4 Glottal to Noise Excitation ratio (GNE) 43
2.2.1.5 Vocal Fold Excitation Ratios (VFER) 46
2.2.1.6 Empirical Mode Descomposition Excitation Ratios (EMD-ER) 49
2.2.1.7 Pitch Period Entropy (PPE) 52
2.2.2 SCRIPT melcepst.m 53
2.2.2.1 Mel Frequency Cepstral Coefficients (MFCC) 53
2.2.3 SCRIPT wavedec_features.m 56
2.2.3.1 Wavelet Measures 57

2.2.4 SCRIPT fastfda.m 62
2.2.4.1 Dentrended Fluctuation Analysis (DFA) 62
2.2.5 SCRIPT rpde.m 62
2.2.5.1 Recuerrence Period Density Entropy (RPDE) 62
2.3 DETERMINACIÓN DEL TAMAÑO MÍNIMO DE LA MUESTRA PARA LA EXTRACCIÓN DE CARACTERÍSTICAS CONFIABLE 63
2.3.1 Verificación de la Fonación de la Vocal /a/ 63
2.3.2 Eliminación de Saturación Y Silencios 66
2.3.3 Enventanar Segmentos 68
2.3.4 Reducción de Dimensionalidad 71
2.4 RECONSTRUCCIÓN DEL CONJUNTO DE DATOS PVA_DRYRUN 75
3. RESULTADOS Y DISCUSIÓN 80
4. CONCLUSIONES Y RECOMENDACIONES 85
4.1. CONCLUSIONES 85
4.2. RECOMENDACIONES 86
5. REFERENCIAS BIBLIOGRÁFICAS 87
ANEXOS 92

RESUMEN

La enfermedad de Parkinson (EP) es un trastorno neurodegenerativo que afecta al sistema nervioso de manera crónica y progresiva. Algunas investigaciones que se han enfocado en la detección de la EP, han utilizado la voz como elemento de detección de la enfermedad porque los primeros síntomas la afectan directamente.

En el presente proyecto se analizará la primera etapa para la detección de la EP; que consiste en la extracción de características de registros de voz. Para ello, se utiliza el conjunto de datos PVA-dryrun que contiene 779 registros de voz de personas que padecen la EP y el *Toolbox Voice Analysis* para la extracción de 339 características.

En el desarrollo del proyecto, primero se analiza el conjunto de datos empezando por la verificación de que cada registro de voz sea la fonación de la vocal /a/, ya que es un requisito para el correcto funcionamiento del *toolbox*. Después, se desarrolla un script en el entorno Matlab, que permite depurar los registros de voz automáticamente; la depuración consiste en eliminar silencio y saturación basándose en el concepto de energía de la señal.

Una vez depurados los registros de voz, para cada registro se toman segmentos de diferentes tamaños para extraer sus características con el fin de analizar su dispersión. El segmento que menor dispersión produjo, permitió determinar el tamaño que deben tener los registros, el cual fue de 4 segundos.

Por lo tanto, de cada registro de voz depurado, se toman 4 segundos y se extraen 339 características para la reconstrucción de la base de datos.

PALABRAS CLAVE: Parkinson, MATLAB, extracción de características, depuración, PVA-dryrun.

ABSTRACT

Parkinson's disease (PD) is a neurodegenerative disorder that it affects the nervous system in a chronic and progressive way. Some researches that have focused on the detection of PD, it has used voice as an element of disease detection because the first symptoms directly affect it.

In this project the first stage for the detection of PD will be analyzed; which consists of extracting features from voice records. For this, the PVA-dryrun data set is used, which contains 779 voice records of people suffering PD and the Toolbox Voice Analysis for the extraction of 339 features.

With the development of the project, the data set is first analyzed starting with the verification that each voice record is the phonation of the vowel /ei/, since it is a requirement for the correct operation of the toolbox. Afterwards, a script is developed in the Matlab environment, which allows you to debug voice records automatically; debugging involves eliminating silence and saturation based on the concept of signal energy.

Once the voice records have been purified, segments of different sizes are taken for each record to extract their features in order to analyze their dispersion. The segment that produced the lowest dispersion, allowed to determine the size of the records, which was 4 seconds.

Therefore, from each purified voice record, 4 seconds are taken and 339 features are extracted for reconstruction of the database.

KEYWORDS: Parkinson, MATLAB, feature extraction, debugging, PVA-dryrun.

1. INTRODUCCIÓN

La enfermedad de Parkinson (EP) es un trastorno neurodegenerativo que afecta al sistema nervioso de manera crónica y progresiva. Se caracteriza por la pérdida (o degeneración) de las células dopaminérgicas en el organismo [1]. La parte del cerebro en la que se produce esta pérdida de células se llama sustancia negra, la cual está situada en la zona media del cerebro. La falta de dopamina se ve reflejada en la alteración del control del movimiento [2].

El 90% de los pacientes con EP experimentan problemas en el habla y la voz, siendo estos problemas los primeros en manifestarse, afectando respiración, articulación, fonación y prosodia [1]. El habla de un paciente con EP está caracterizada por tener baja intensidad, disminución en la fluidez verbal y el ritmo, ser pobremente prosódica, es decir, disminución de acentos, tonos y entonación a lo que se incluye problemas en la articulación de los sonidos reduciendo significativamente su inteligibilidad [3].

En la actualidad, todavía no existen métodos concretos para diagnosticar el Parkinson. El diagnóstico realizado por los médicos es subjetivo y se enfoca en la revisión de los signos y síntomas del paciente [4]. Lamentablemente, estos síntomas solo se vuelven evidentes en etapas avanzadas de la enfermedad.

La detección de la enfermedad de Parkinson ha generado gran interés en el mundo científico, por lo que se han realizado innumerables investigaciones que han determinado una estrecha relación entre la EP y la voz de los pacientes, incluso en etapas tempranas de la enfermedad. Con estos antecedentes se han desarrollado herramientas automáticas que permitan detectar la enfermedad por medio de la voz, logrando elevadas tasas de precisión en cuanto al reconocimiento de la enfermedad. Estos métodos, por otro lado, además de ser objetivos, son también no invasivos y permiten saber si un paciente adolece de la EP mucho antes que los métodos tradicionales [1].

Para hacer uso de estas herramientas de reconocimiento automático de la enfermedad, es necesario que la voz del paciente sea inicialmente digitalizada y luego caracterizada. Una vez representada la voz mediante sus características, es necesario un proceso de selección de características que permita reducir el costo computacional. Luego de este proceso, la voz representada por un subconjunto de características debe ser clasificada para determinar si una persona está sana o enferma. En la Figura 1.1 se pueden apreciar las etapas de todo el proceso de detección de la EP a través de la voz.

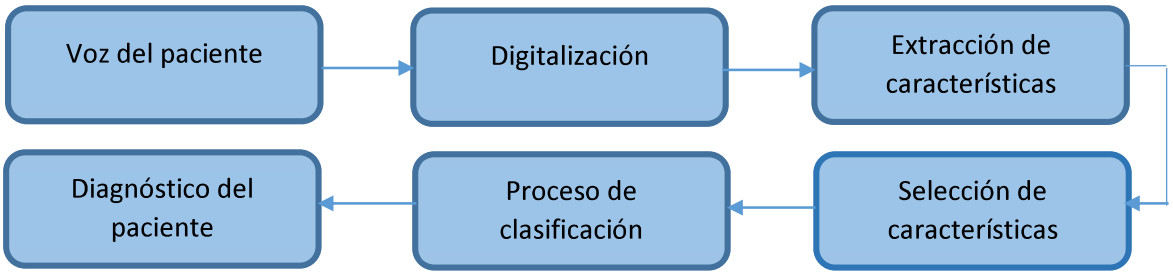

Figura 1.1. Proceso de detección de la enfermedad de Parkinson por medio de la voz

La información tabulada de las características de la voz, el estado del paciente y eventualmente los registros de audio, forman lo que se conoce como un "*conjunto de datos*". La elaboración del conjunto de datos implica altos costos de inversión debido a la cantidad de recursos que se deben utilizar en largos períodos de tiempo, pues se necesita personal que realicen las grabaciones, pacientes que estén dispuestos a colaborar en varias sesiones, médicos que los evalúen a través de la escala UPDRS o similar, equipo de grabación, etc. Por esta razón, se usan conjuntos de datos que sean asequibles gratuitamente a través de la red para las investigaciones realizadas en este campo.

Estos conjuntos de datos deben ser suficientemente grandes como para obtener resultados confiables. En la *web* se pueden encontrar pocos conjuntos de datos de libre acceso y su tamaño es bastante limitado y, a pesar de poseer información de las características de la voz de pacientes y de su estado respecto a la enfermedad (sanos o enfermos), no disponen de los registros de audio y tampoco indican de forma clara cómo se han calculado las características de los registros de voz. Hay, sin embargo, un conjunto de datos que posee una gran cantidad de registros de audio de los pacientes e indica el grado de enfermedad que poseen, este conjunto de datos es PVA-dryrun.

En la generalidad de los estudios publicados [1], [5], [6], [7] y [8], los conjuntos de datos utilizados no superan los 300 registros de voz de los pacientes. PVA-dryrun además de poseer 779 grabaciones de voz [9], tiene también la evaluación de los síntomas auto informados (PDRS – Escala de clasificación de la EP y etapa *Hoehn* & *Yahr*) que permiten evaluar la gravedad de la enfermedad; a pesar de esto, este conjunto de datos no es muy utilizado. Vale la pena analizar las razones de su poca utilización, ya que dicho conjunto de datos tiene una valiosa información que otros no la tienen, como, por ejemplo, los registros de audio, la escala de clasificación PDRS, la edad de los pacientes y el género.

Los problemas que han sido detectados están relacionados a las señales de audio, algunas de ellas están saturadas, lo que no permite extraer de forma confiable las características de la voz; sin embargo, se las reporta como buenas. En la Figura 1.2 puede apreciarse uno de estos ejemplos.

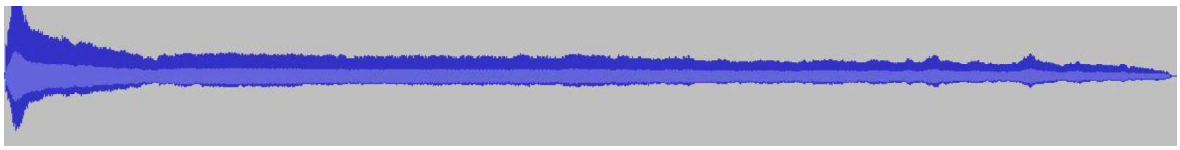

Figura 1.2. Forma de onda del archivo pva_0023336_2014-01-10-221846.wav

Otras, se las reporta como malas, pero se podría extraer la parte medible de la voz para recuperar la información adicional que poseen (PDRS entre otros), como por ejemplo el audio pva_6955455_2014-02-08-022632.wav, que puede visualizarse en la Figura 1.3.

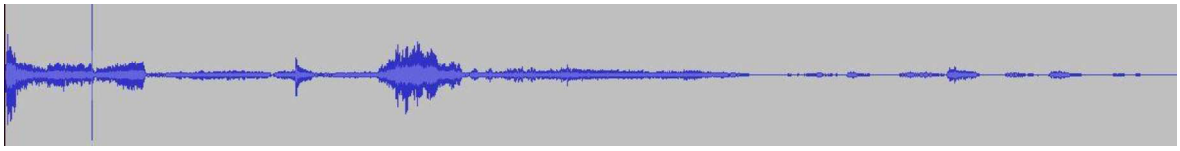

Figura 1.3. Forma de onda del archivo pva_6955455_2014-02-08-022632.wav

Estos problemas podrían ser resueltos al analizar minuciosamente cada uno de los 779 registros de audio que conforman el conjunto de datos PVA-dryrun.

En cuanto a las características de la voz de los audios contenidas en este conjunto de datos, tienen un nombre que identifica a cada una de ellas, pero no se indica cómo fueron obtenidas matemáticamente. A partir de los archivos de audio se puede recurrir a herramientas que permitan el cálculo de dichas características de la voz, no solamente de las 38 que el conjunto de datos posee, sino de más de 300 características y se puede explicitar cómo fueron calculadas matemáticamente cada una de ellas.

Los datos que se pueden recuperar de PVA-dryrun, permitirían realizar estudios para determinar, no solamente si un paciente adolece o no de la EP mediante algún método de clasificación, sino que además se podría determinar el grado de su enfermedad, permitiendo implementar algoritmos de regresión.

1.1 OBJETIVOS

1.1.1 OBJETIVO GENERAL

- Depurar la base de datos PVA DRYRUN (Patient Voice Analysis).

1.1.2 OBJETIVOS ESPECÍFICOS

- Determinar los rangos de tiempo adecuados para una extracción de características confiable.
- Analizar los registros de audio del conjunto de datos PVA-dryrun.
- Extraer y expresar matemáticamente las características de los registros de audio analizados.
- Reconstruir el conjunto de datos PVA-dryrun.

1.2 ALCANCE

Se realizará un estudio sobre el tamaño mínimo que deberían tener las muestras de voz que permita hacer una extracción de características confiables.

Se realizará el análisis de 779 registros de audio documentando los errores encontrados y especificando claramente cuáles son los rangos de las muestras que han sido considerados para el cálculo de sus características. Los archivos de audio originales se los mantendrá sin ninguna modificación.

Se utilizará el *toolbox* de análisis de voz en el entorno de MATLAB que permitirá extraer 339 características de los fragmentos de voz determinados previamente y se indicará la fórmula matemática utilizada para cada uno de estos cálculos.

Como se puede observar en la Figura 1.4, del procedimiento para la detección del Parkinson por medio de la voz, se partirá de la voz digitalizada y se llegará hasta la extracción de características.

Figura 1.4. Partes de la detección de la EP

1.3 MARCO TEÓRICO

1.3.1 TRASTORNOS DEL HABLA Y LA VOZ EN LA EP

1.3.1.1. Fundamentos sobre la Enfermedad de Parkinson

James Parkinson basándose en la observación de seis casos, en 1817 describió la enfermedad como parálisis agitante [11]. Años después Martin Charcot adoptó Parkinson como nombre de la enfermedad [10].

La Enfermedad de Parkinson (EP) es la segunda enfermedad más común después del Alzheimer [10], es la causante de la pérdida parcial o total de los reflejos motores, el habla, el comportamiento, el procesamiento mental y otras funciones vitales [17], según la historia se ha establecido la confirmación de la enfermedad desde el año 5000 a.C [16].

La EP es un trastorno neurodegenerativo que afecta al sistema nervioso de manera crónica y progresiva, caracterizándose por la pérdida (o degeneración) de las células dopaminérgicas en el organismo [1]. En los pacientes con Parkinson, el 80% o más de estas células productoras de dopamina están degeneradas, muertas o dañadas, lo que origina que se disparen incontrolablemente, dejando a los pacientes incapaces de manejar sus movimientos [16]. Estudios realizados concluyen que la edad es el factor de riesgo más importante para el inicio de la enfermedad, la cual aumenta considerablemente después de los 50 años [13], en donde aproximadamente el 1% padece la enfermedad [12].

La intervención quirúrgica y la medicación pueden retrasar la progresión de la enfermedad y aliviar algunos de los síntomas, no existe cura disponible para esta enfermedad. Por lo que, el diagnóstico temprano es crítico para mejorar la calidad de vida del paciente y prolongarlo [13].

Actualmente, para determinar si una persona padece la enfermedad se realizan exámenes físicos aplicando varias pruebas empíricas, dando lugar a un diagnóstico subjetivo el cual se obtiene una vez que la enfermedad ha empezado a mostrar problemas en el paciente y es más difícil disminuir su progreso. Con el propósito de realizar un diagnóstico en la etapa temprana de la enfermedad, surge la necesidad de desarrollar herramientas objetivas y confiables que permitan eliminar el diagnóstico deficiente que se maneja actualmente [13].

Las observaciones de las pruebas físicas se asignan a una métrica diseñada específicamente para seguir la progresión de la enfermedad, por lo general la Escala de Clasificación de la Enfermedad de Parkinson Unificada (UPDRS), ampliamente utilizada

para evaluar la gravedad de la EP. Una versión abreviada de la UPDRS es el cuestionario PDRS de 17 preguntas [13].

1.3.1.2 Sistema Fonatorio

La fonación es una de las partes más importantes de la producción de la voz [19], aquí se realiza la transformación de energía aerodinámica, generada por el aparato respiratorio, en energía acústica transmitida a los labios, que finalmente se percibe como la voz [21].

La producción del habla se origina en la corteza neuromotora lingüística (punto 1 de la Figura 1.5). La secuencia de activación neuromotora involucrada en la producción del habla se transmite a la faringe (punto 2 de la Figura 1.5), la lengua (punto 3 de la Figura 1.5), la laringe (punto 4 de la Figura 1.5), el tórax y el diafragma (punto 5 de la Figura 1.5) a través de las unidades secundarias subtalámicas. El control muscular es proporcionado por un sistema de control de retroalimentación (punto 6 de la Figura 1.3) [18].

La laringe es el órgano principal para la fonación, ya que es un conducto que sirve para la circulación del aire, en sus caras laterales se encuentra un cartílago llamado tiroides, detrás del cual están ubicadas las cuerdas vocales [20].

Las cuerdas vocales están formadas por los bordes externos que se unen a los músculos de la laringe y los bordes internos que están libres formando una apertura llamada glotis. Dependiendo de la fuerza que ejercen los músculos de la laringe se regula el tono de voz, además a mayor apertura de la glotis tenemos mayor intensidad [19].

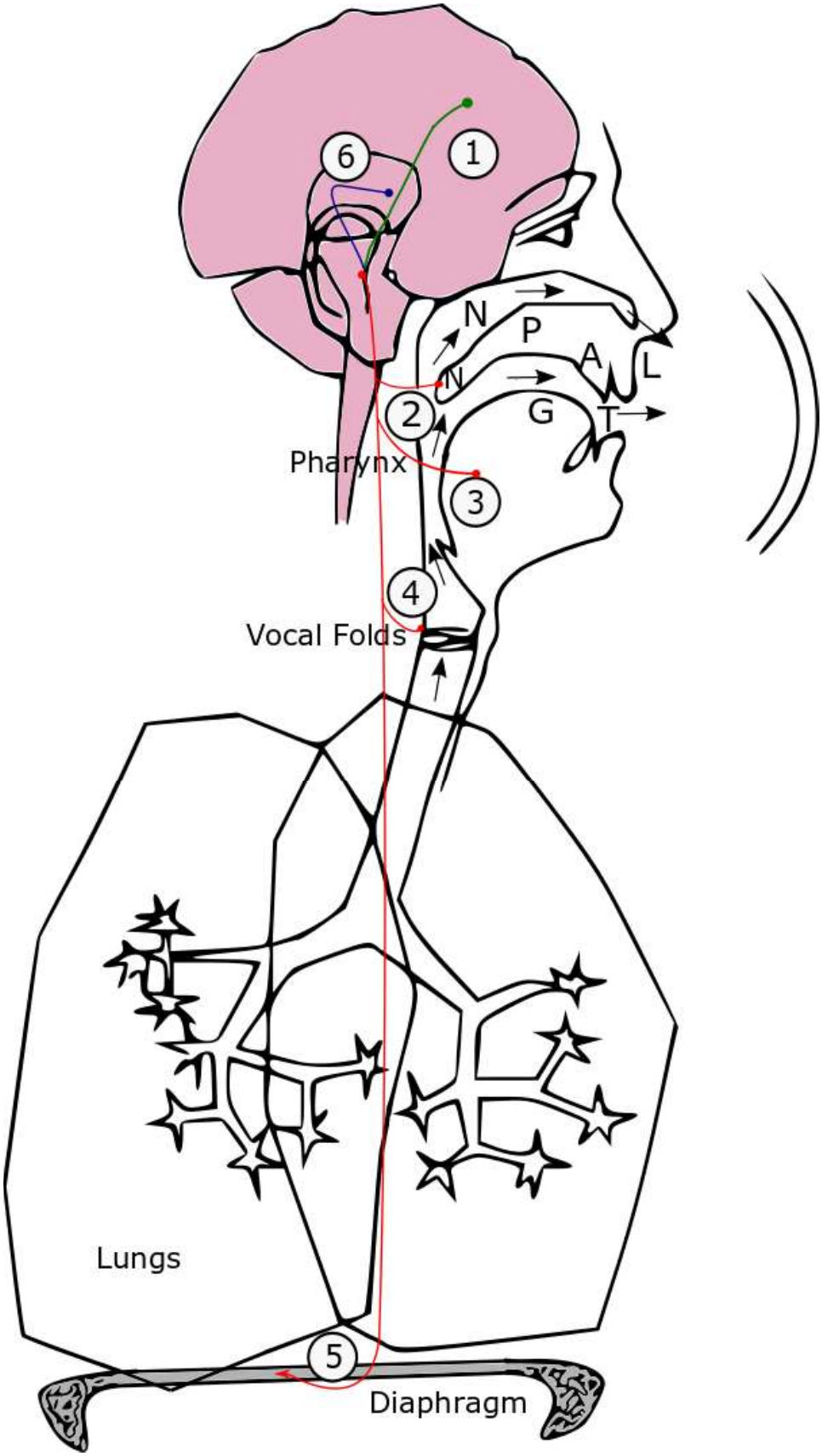

Figura 1.5. Principales caminos neuronales involucrados en la producción de la voz [18]

1.3.1.3 Relación existente entre la EP y la voz

Estudios realizados reportan que el deterioro vocal tiene una prevalencia del 70-90% después del inicio de la enfermedad. Además, que puede ser uno de los primeros indicadores de la EP, y el 29% de los pacientes lo consideran uno de sus mayores obstáculos [13].

Existe evidencia de un rendimiento degradante en la voz con la progresión de la EP, con disfonía (dificultad respiratoria, ronquera o crujido en la voz), hipofonía (volumen de voz reducido) y disartria (dificultad para articular los sonidos o sílabas) [17].

Los indicadores disfónicos de la EP hacen que las mediciones del habla sean una parte importante del diagnóstico, ya que se han propuesto como una herramienta confiable para detectar y monitorear la EP [15].

Uno de los métodos más utilizados en la detección del Parkinson por medio de la voz es la grabación de señales de voz, porque es fácil y no es invasiva, además se considera un parámetro notable para detectar y seguir la progresión de los síntomas [14]. En este método se emplean fonaciones vocales sostenidas que permiten evaluar la extensión de los síntomas vocales, donde se solicita al paciente que mantenga la frecuencia de la fonación estable durante el mayor tiempo posible [13].

Con el fin de caracterizar objetivamente los síntomas de disfonía, las señales de voz grabadas se analizan mediante algoritmos de procesamiento del habla [13].

1.3.2 MEDIDAS DE DISFONÍA

Las técnicas utilizadas para extraer características de las señales de voz son conocidas como medidas de disfonía [22]. Cada una de las medidas de disfonía tiene como objetivo, extraer distintos parámetros de la señal y reflejarlo como un solo número; ayudan a detectar patologías de las cuerdas vocales, haciendo una comparación entre los parámetros de la voz obtenidos de personas sanas respecto de los parámetros obtenidos de personas con diferentes patologías [23].

1.3.2.1 Conceptos matemáticos recurrentes

- Autocorrelación

Varios métodos de procesamiento de señales clásicos se basan en técnicas como la autocorrelación a corto plazo, que para una señal x con retraso l se define como lo indica la ecuación 1.1

$$R_{xx}(l) = \sum_{n=-\infty}^{\infty} (x_n * x_{n-l}{}^*) = x_l \otimes x_{-l}{}^* \tag{1. 1}$$

Donde $(\cdot)^*$ denota el complejo conjugado. Algunos autores se refieren a la ecuación previa como autovarianza, reservándose el término autocorrelación para el caso donde x es normalizada [22].

La autocorrelación es una herramienta para encontrar patrones repetitivos en una señal que puede estar enmascarada por el ruido y expresa la similitud entre las muestras en función de la diferencia de sus índices. Tiene un máximo global en $l = 0$, y dividiendo $R_{xx}(l)$ por $R_{xx}(0)$, lo normaliza en el rango de -1 a 1. Se asume que $-1 \leq R_{xx}(l) \leq 1$ [22].

Como información adicional tenemos que existe una relación algorítmica entre la transformada rápida de Fourier (FFT) y la autocorrelación: la autocorrelación puede ser calculada como la trasformada rápida de Fourier inversa (IFFT) de la densidad de potencia espectral $|X(w)|^2$ [22].

- Análisis de predicción lineal (LPC)

Permite calcular la envolvente espectral y los coeficientes LPC

1.3.2.2 Fonación de vocales sostenidas

Dado que la grabación de señales de voz no es invasiva y se puede integrar fácilmente en aplicaciones de telemedicina, estas pruebas son muy factibles para realizar el estudio de la detección de la enfermedad de Parkinson [13].

La interpretación vocal generalmente se logra utilizando fonaciones de vocales sostenidas, por que evitan confusiones, son muy prácticas y la información obtenida es suficiente para evaluar la extensión de los síntomas vocales, generalmente se solicita a las personas que mantengan la frecuencia de la fonación estable durante el mayor tiempo posible [24]. La duración máxima de una fonación contiene información útil, un adulto sano tiene la capacidad de sostener su voz durante 20 segundos en promedio, aunque esto depende de diferentes factores como edad, sexo, estatura del cuerpo y estado de salud.

Las vocales sostenidas se basan en la idea de que un sujeto sano puede provocar una fonación estacionaria, mientras que para una persona con alguna deficiencia vocal podría

ser imposible de realizar. Un proceso estacionario no cambia cuando se desplaza en el tiempo o el lugar, e implica periodicidad [22].

La mayoría de investigaciones se centran en la vocal sostenida /a/, ya que además de ser el sonido más simple de producir proporciona la mayor cantidad de información útil. La generación de esta vocal sostenida se produce por la combinación de una variedad de músculos en las cuerdas vocales y el tracto vocal, por lo que aumenta la probabilidad de que se pueda identificar un problema neurológico [22].

La vocal sostenida /a/ proporciona una configuración abierta del tracto vocal en la que la boca se abre al máximo en comparación con otras vocales, minimizando el pulso de aire reflejado en las cuerdas vocales; por lo tanto, el nivel de presión del sonido vocal registrado en los labios se maximiza [22].

Las perturbaciones encontradas es las señales de voz cuantifican la desviación de la periodicidad, dando lugar a algunos de los métodos más importantes para extraer información clínicamente relevante de las señales de voz, denominadas en la literatura como medidas de disfonía [22].

1.3.2.3 Estimación de la frecuencia fundamental

Varias de las medidas necesitan que se realice previamente el cálculo de la frecuencia fundamental, para lo cual existen diferentes métodos que permiten realizar el cálculo de esta característica intrínseca de las señales del voz [22].

Las cuerdas vocales son las encargadas de producir el sonido de la voz, el cual está constituido por la frecuencia fundamental y los armónicos con múltiplos enteros de dicha frecuencia; inicialmente este complejo sonido debe pasar por el tracto vocal también conocido como cavidades de resonancia que están constituidas por la cavidad nasal, cavidad oral y la cavidad formada por la faringe y la laringe, luego de este proceso será producida la voz. El tracto vocal trabaja como un filtro de las ondas de sonido que entran en él, creando máximos a ciertas frecuencias conocidas como frecuencia fundamental y frecuencias formantes [23].

La frecuencia fundamental también conocida como pitch está determinada por tres factores: la masa de las cuerdas vocales, la viscoelasticidad de las cuerdas vocales y la presión subglótica, que es la presión que hay bajo el espacio de las cuerdas vocales [23].

Existe una gran cantidad de algoritmos que permiten realizar una estimación de la frecuencia fundamental, algunos investigadores prefieren llamarlos algoritmos de

detección de tono. Dentro de la gran cantidad de algoritmos están los que funcionan ya sea en el dominio del tiempo, en el dominio de la frecuencia o algunos casos combinan información de dominio de tiempo y dominio de frecuencia para obtener estimaciones más confiables.

Entre las estrategias más utilizadas para realizar la estimación tenemos la selección de picos y la coincidencia de formas de onda; siendo esta última la más utilizada debido a su robustez contra el ruido.

Algoritmos de estimación de la frecuencia fundamental

1.3.2.3.1 Praat

Es un software de estimación de F0, propuesto inicialmente por Boersma, el cual se muestra como un enfoque de dominio de tiempo, basándose en la autocorrelación para el cálculo de F0. Por lo general, se superpone una ventana de tiempo de 40-80 ms a la señal de voz $\boldsymbol{x}$, la cual la divide en N segmentos, donde $N \geq \frac{duración\ de\ la\ señal}{ventana\ de\ tiempo}$ (la igualdad se mantiene para las ventanas que no se superponen). Para cada uno de los N segmentos, se resta el valor promedio y la señal resultante $\boldsymbol{s}$ se multiplica por una ventana apropiada $\boldsymbol{w}$ (PRAAT utiliza la ventana de Hanning o la de Gauss) dando lugar a N señales de ventana $\boldsymbol{a}$. Se calcula la autocorrelación de cada segmento de señal como $r(l) \cong \frac{R_{aa}(l)/\max(R_{aa}(l))}{R_{ww}(l)/\max(R_{ww}(l))}$, donde $R_{aa}(l)/\max(R_{aa}(l))$ es la autocorrelación normalizada de la señal de ventana $\boldsymbol{a}$, y $R_{ww}(l)/\max(R_{ww}(l))$ es la autocorrelación normalizada de la ventana $\boldsymbol{w}$, siendo $\max(R_{aa}(l))$ y $\max(R_{ww}(l))$ el instante de tiempo en donde se encuentra la mayor autocorrelación.

El índice l se encuentra entre un valor máximo y mínimo, típicamente de la frecuencia fundamental, $F_{0,min} = 50Hz$ y $F_{0,max} = 500Hz$, en consecuencia $\frac{F_s}{500} \leq l_{max} \leq \frac{F_s}{50}$.

El algoritmo original utilizo la ventana de Hanning, más tarde Boersma indico que PRAAT proporciona mejores estimaciones cuando usa la ventana de Gauss aunque sea más complicada [22].

1.3.2.3.2 Shrp

El algoritmo SHRP es conocido como uno de los métodos de estimación de F0 más precisos junto con PRAAT. SHRP trabaja en el dominio de la frecuencia utilizando la relación de subarmónicos a armónicos (SHR), que se define como la relación entre la

amplitud de subarmónicos y armónicos [22]. Fue propuesto por Sun quien mediante una serie de experimentos encontró que el tono se percibe de manera diferente cuando los subarmónicos aumentan en una señal [24].

El principal objetivo detrás de SHRP es tratar los problemas que aparecen en la estimación de F0 debido a ciclos alternos (ciclos vocales adyacentes que fluctúan en amplitud y/o periodo) que se manifiestan en el dominio de la frecuencia como subarmónicos fuertes. SHRP utiliza ventanas para realizar el análisis de la señal (por ejemplo, realiza estimaciones cada 10 ms) [22].

1.3.2.3.3 Swipe

Es un algoritmo de estimación de tono inspirado en la forma de onda *sawtooth* (diente de sierra), trabaja en el dominio de la frecuencia. SWIPE en lugar de centrarse únicamente en las ubicaciones de armónicos (picos en el espectro) como en SHRP, utiliza la información disponible en todo el espectro utilizando núcleos. Identifica los armónicos en la raíz cuadrada del espectro e impone núcleos con pesos decrecientes en las ubicaciones armónicas detectadas [24].

SWIPE funciona en segmentos de la señal de voz, tal como se define en PRAAT. En cada segmento se calcula la raíz cuadrada del espectro: utilizando la raíz cuadrada en lugar del cuadrado del espectro. Luego se identifica los armónicos en la raíz cuadrada del espectro e impone núcleos con pesos decrecientes en las ubicaciones armónicas. Estudios empíricos determinaron que se puede obtener resultados óptimos cuando se usan núcleos con peso $\frac{1}{\sqrt{k}}$, donde k corresponde al kth armónico [22].

1.3.2.3.4 Dypsa

El algoritmo DYPSA (Dynamic Programming Projected Phase-Slope) tiene como objetivo identificar los ciclos glotales directamente en lugar de proporcionar estimaciones en intervalos de tiempo previamente especificados. De los ciclos glóticos identificados se puede inferir F0. Específicamente, el algoritmo DYPSA está enfocado en detectar el cierre glótico (colisión de las cuerdas vocales) en el habla, ya que la apertura glótica (separación de las cuerdas vocales) es más difícil de detectar porque la energía de excitación es típicamente más débil y más dispersa en el tiempo en comparación con el cierre glótico [22].

DYPSA identifica los eventos de cierre glótico candidatos y utiliza la programación dinámica para recuperar eventos verdaderos. La programación dinámica en DYPSA es

efectivamente un problema de optimización, que se enfoca en determinar los cierres glóticos más probables de una lista de candidatos al penalizar una serie de atributos. Estos atributos incluyen un costo de similitud en la forma de onda del habla (se espera que los ciclos sucesivos tengan características de excitación similares), un costo de energía normalizada (para minimizar los eventos de ruido no esenciales) y un costo de desviación del tono (se espera que los tiempos de demora en los que se producen los cierres glóticos sucesivos no difieran masivamente) [22].

1.3.2.4 Medidas de disfonía clásicas

El procesamiento de señales lineales es una herramienta que domina el análisis del habla. El término lineal hace referencia a un método en donde la salida es proporcional a una combinación lineal de entradas, una propiedad conocida como el principio de superposición.

1.3.2.4.1 Jitter

El *jitter* es una medida de disfonía afectada principalmente por la falta de control de la vibración de las cuerdas vocales; las voces patologías suelen tener mayor porcentaje de *jitter* que las voces sanas [18]. Principalmente esta medida de disfonía pretende cuantificar las perturbaciones ciclo a ciclo de la frecuencia fundamental F_0, representadas como pequeñas desviaciones de la periodicidad exacta. *Jitter* carece de una definición formal e inequívoca lo que conlleva al desarrollo de muchas variantes [22].

Para realizar el cálculo del *jitter* se utiliza el contorno de F_0 o el contorno inversamente proporcional del periodo del pitch $T_0 = 1/F_0$ [22].

Entre las diferentes variantes del *jitter* tenemos: la diferencia absoluta media de F_0 estimada entre ciclos sucesivos o también conocida como variación total discreta de F_0, la diferencia absoluta media de F_0 de ciclos sucesivos dividida por la media de F_0 expresada en porcentaje (%), el cociente de perturbación usando K ciclos, el cociente de perturbación usando modelo autoregresivo, la perturbación absoluta media y la perturbación cuadrada media normalizada [22].

El contorno de F_0 también puede ser analizado utilizando el operador de energía no lineal *Teager-Kaiser* (TKEO). TKEO cuantifica la modulación de amplitud y la modulación de frecuencia de una señal oscilante, con el fin de calcular la energía no lineal de la frecuencia fundamental [22].

1.3.2.4.2 Shimmer

Es una medida de disfonía relacionada a la disminución de la resistencia glotal y a lesiones de las cuerdas vocales, factores por los cuales está vinculada a la presencia de emisión de ruido [18]. El *shimmer* mide perturbaciones de la amplitud de la señal de voz antes que la frecuencia fundamental; utilizando para sus cálculos el contorno de la amplitud A_0. El contorno de la amplitud A_0 se define utilizando el valor de amplitud máxima dentro de cada ciclo glotal, también se lo pude definir enfocándose en los segmentos de la señal (por ejemplo 40 ms) en lugar de los ciclos glotales, o usando los valores de mínima amplitud [22].

Entre las variantes del *shimmer* tenemos: la diferencia absoluta media de A_0 estimada entre ciclos sucesivos o también conocida como variación total discreta de A_0, la diferencia absoluta media de A_0 de ciclos sucesivos dividida por la media de A_0 todo esto expresado en porcentaje (%), el cociente de perturbación usando K ciclos, el cociente de perturbación usando un modelo autoregresivo, la perturbación absoluta media y la perturbación cuadrada media normalizada [22].

1.3.2.4.3 HNR y NHR

Las relaciones armónicos a ruido (HNR) y ruido a armónicos (NHR), son medidas de disfonía que tienen como objetivo cuantificar el ruido en una señal de voz que generalmente se produce por el cierre incompleto de las cuerdas vocales [22].

La relación de armónico a ruido proporciona una indicación de la periodicidad global de la señal de voz al cuantificar la relación entre los componentes periódico (parte armónica) y aperiódico (ruido). Esta medida generalmente es calculada como una característica general de la señal, y no como una función de la frecuencia [25].

1.3.2.4.4 Mel Frequency Cepstral Coefficients (MFCC)

La enfermedad de Parkinson además de afectar las cuerdas vocales, también afecta las articulaciones del tracto vocal, ya que en la producción de la voz están estrechamente relacionadas, las cuerdas vocales son las encargadas de producir el sonido, el cual posteriormente es filtrado por las cavidades de resonancia o tracto vocal, produciendo finalmente la voz, que es lo que escuchamos. El objetivo de los MFCC es detectar cambios sutiles que se producen en el movimiento de las articulaciones del tracto vocal, tales como lengua, mandíbula y labios [26].

Previo a la obtención de los coeficientes se debe considerar que la naturaleza de la señal de voz es aleatoria y no estacionaria, para evitar problemas en su análisis se trabaja con cortos lapsos de tiempo en donde la señal sea casi estacionaria. Este análisis en donde se obtienen tramas consecutivas de pocos *ms* se conoce como enventanado, para lo cual se utilizan ventanas de tipo Hamming, Hanning y Rectangular. El enventanado con bloques de muestras solapadas entre sí, se realiza para evitar pérdidas de información en la transición entre ventanas.

La teoría del filtro-fuente lineal expresa que la producción de la voz es el resultado de la convolución entre la salida de las cuerdas vocales y la respuesta impulso del tracto vocal. Para cumplir con el objetivo de los MFCC es necesario realizar la deconvolución de la señal con el fin de analizarlos por separado. La deconvolución de señales de voz generalmente se realiza mediante el análisis en el dominio cepstral, el cual transforma una convolución en el dominio del tiempo en una suma en el dominio espectral. El dominio cepstral está definido como la transformada de Fourier inversa del logaritmo del espectro de potencia de la señal de voz. Específicamente, para una señal $x = (x_1 \ldots\ldots x_N)$ se tiene la ecuación 1.2:

$$c_n = \frac{1}{N}\sum_{k=1}^{N} \log(|X(\omega)|) \cdot \exp(-j\frac{2k\pi}{N}n) \tag{1. 2}$$

Donde $X(\omega)$ es la DFT de la señal original.

El análisis cepstral es generalmente combinado con el dominio espectral usando un banco de filtros triangulares, espaciados de acuerdo a la *mel frequency scale* que se muestra a continuación en la ecuación 1.3:

$$mel(f) = 2595\log\left(1 + \frac{f}{5000}\right), \qquad 0 \leq f \leq Fs \tag{1. 3}$$

Esta escala se aproxima a la respuesta del sistema auditivo humano para sonidos de diferentes frecuencias, emulando las propiedades del filtrado efectivo del oído humano. La escala de frecuencias de Mel es una transformación no lineal de la escala de frecuencia lineal, que divide las frecuencias en bandas de frecuencias (superpuestas). Un ejemplo de banco de filtros se puede ver en la figura 1.6:

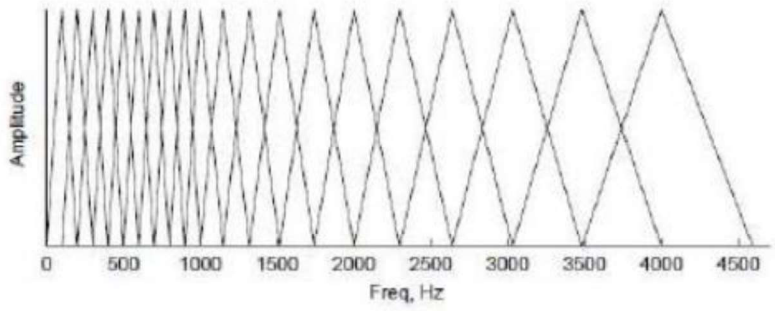

Figura 1.6. Banco de filtros [*S. B. David and P. Mermelstin, 1980*].

Finalmente, los coeficientes MFCC calculan la distribución de la energía de la señal de voz para cada banda de frecuencia como se muestra en la ecuación 1.4 [22].

$$MFCC_n = \sum_{k=1}^{K} E_k \cos\left[\frac{n(k-0.5)\pi}{K}\right], \qquad N = 0, \dots\dots, L \tag{1. 4}$$

Donde L es el número de coeficientes MFCC y E_k es la energía media de los kth filtros.

La energía media es el promedio de la amplitud al cuadrado de la transformada de Fourier, teniendo en cuenta los filtros mel triangulares. El coeficiente MFCC 0° representa la energía de la señal [22].

Generalmente se utiliza la primera y segunda derivada en el tiempo de los MFCC, que se conocen como coeficientes delta y coeficientes delta-delta, respectivamente.

1.3.2.5 Medidas de disfonía modernas

Estudios actuales ([36] y [37]) se han enfocado en explorar modelos no lineales, ya que a menudo se ha obtenido mayor precisión en la representación de características que los modelos lineales [22].

1.3.2.5.1 Glottal to noise excitation ratio (GNE)

La relación entre la excitación glótica y el ruido (GNE) es una medida que cuantifica la cantidad de ruido en la señal, basándose en que las colisiones de las cuerdas vocales generan excitación sincrónica de las bandas de diferente frecuencia, mientras que el ruido turbulento producido por un cierre incompleto de las cuerdas vocales, genera excitación asincrónica [22].

Han sido propuestos algunos pasos para el cálculo de la GNE: 1) Submuestrear la señal a 10 KHz, 2) Realizar un filtro inverso de la señal con el fin de detectar ciclos glóticos, 3) Calcular la envolvente de Hilbert (valor absoluto de la señal analítica compleja) de las diferentes bandas de frecuencia empleando un ancho de banda específico para cada ciclo glotal, 4) Un par de envolventes donde la diferencia de sus frecuencias centrales es mayor o igual que la mitad del ancho de banda son utilizadas para calcular la función de correlación cruzada, 5) Escoger el valor máximo de las correlaciones calculadas, 6) Escoger el valor máximo del paso anterior el cual representa el valor GNE para el ciclo glotal detectado [22].

1.3.2.5.2 Detrended Fluctuation Analysis (DFA)

El análisis de fluctuaciones estocásticas DFA, es una medida que tiene como objetivo caracterizar el grado de ruido turbulento en la señal del habla, cuantificando la autosimilitud estocástica del ruido producido por el flujo del aire turbulento en el tracto vocal. La transpiración u otras disfonías similares causadas por el cierre incompleto de las cuerdas vocales pueden incrementar el valor de DFA, por que las fluctuaciones alrededor de las líneas ajustadas serán más grandes que las fonaciones saludables [13].

Los procesos turbulentos se caracterizan por una escala estadística exponencial α en un rango de escalas físicas, para trastornos generales de la voz, el exponente de la escala es mayor para personas con alguna disfonía que para personas sanas [5].

1.3.2.5.3 Recurrence Period Density Entropy (RPDE)

RPDE es una medida que se enfoca en la capacidad de las cuerdas vocales para mantener una vibración estable, cuantificando las desviaciones de la periodicidad. Adicionalmente tenemos el concepto de que la señal de voz está compuesta por una componente determinista (no lineal) y una componente estocástica, el objetivo de RPDE es obtener la segunda componente [22].

Las disfonías como la ronquera o la voz chirriante suelen causar un aumento de RPDE [13].

1.3.2.5.4 Pitch Period Entropy (PPE)

La entropía de frecuencia fundamental PPE, es la encargada de medir el control deficiente del tono estable durante la fonación sostenida, el cual es un síntoma común en las personas que padecen la enfermedad. Además, es importante mencionar que esta medida

utiliza una escala de tono logarítmico y es robusta para factores de confusión, como el vibrato suave, que está presente en voces tanto sanas como disfonías [13].

La ventaja de utilizar una escala de tono logarítmico, antes que una escala de frecuencia absoluta es la relevancia que tiene para evaluar las variaciones anormales en el tono del habla [23].

1.3.2.6 Medidas de disfonía nuevas

Algunas medidas de disfonía ya presentadas se basan en métodos de procesamiento de señales lineales para señales estacionarias (por ejemplo, HNR que se basa en el cálculo de la transformada de Fourier). Sin dejar de lado la no linealidad y no estacionariedad en mecanismo de producción del habla, se han desarrollado medidas que puedan abarcar las deficiencias encontradas anteriormente. A continuación, se indican las características en las fonaciones de vocales sostenidas que las nuevas medidas de disfonía intentan cuantificar [22].

1.3.2.6.1 Wavelet

La transformada discreta de wavelet (DWT) expresa la señal de dominio de tiempo en el dominio de wavelet y proporciona una representación de tiempo-frecuencia: tiene la propiedad de cuantificar los efectos de regularidad (aspectos de escala) y los procesos transitorios (aspectos de tiempo), cualidades importantes para detectar desviaciones de escala y tiempo. El DWT analiza la señal en diferentes bandas de frecuencia con diferentes resoluciones al descomponer la señal en aproximación (coeficientes de aproximación) e información detallada (coeficientes de detalle) [22].

La descomposición de wavelets se puede considerar como una extensión del DWT, que expresa sucesivamente los coeficientes de aproximación, y si utiliza capas posteriores puede extraer nuevas señales de aproximación y detalle como se muestra en la figura 1.7 [22].

El fundamento de las medidas basadas en la descomposición de wavelets es que se espera que una persona sana sea capaz de mantener una vocal con una desviación mínima de la periodicidad, mientras que personas con trastornos vocales no pueden.

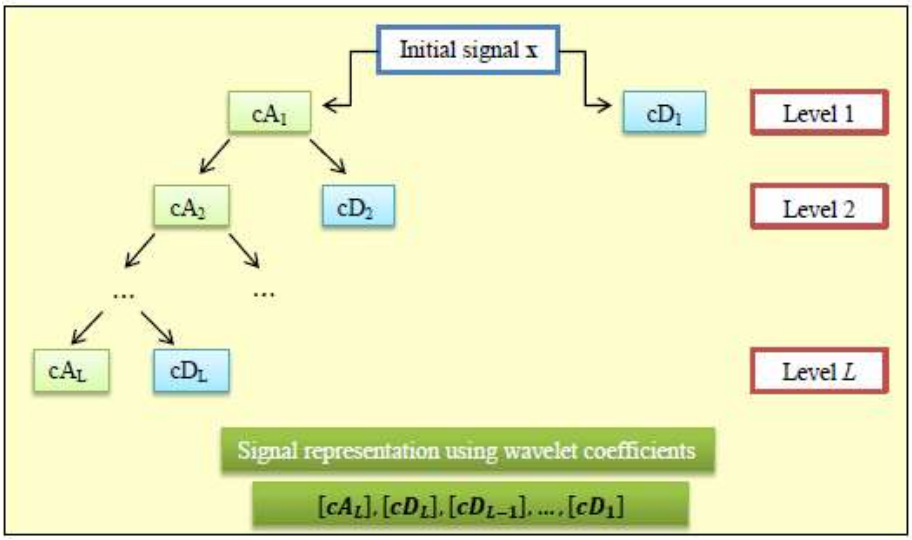

Figura 1.7. Diagrama esquemático de la descomposición de wavelets. Donde A son los coeficientes de aproximación y D los coeficientes de detalle [22].

1.3.2.6.2 Empirical Mode Decomposition Excitation Ratios (EMD-ER)

Son un grupo de medidas de disfonía que se basan en pruebas fisiológicas que sugieren que el ruido turbulento aumenta en las voces patológicas debido a un cierre incompleto de las cuerdas vocales. EMD-ER realiza un enfoque diferente para cuantificar la relación señal a ruido en la producción de voz [22].

EMD descompone una señal de múltiples componentes (señal compuesta de múltiples señales superpuestas) en componentes de señal elementales con contribuciones de AM-FM, que se conocen como funciones de modo intrínseco (IMF). Cada uno de los IMF contribuye tanto en amplitud como en frecuencia para generar la señal observada, que contiene todos los componentes superpuestos de AM Y FM [22].

El algoritmo EMD-ER se puede resumir en cinco pasos: 1) identificar los mínimos y máximos de una señal, 2) utilizar la interpolación spline cúbica para unir los mínimos generando la envoltura inferior, y unir los máximos generando la envoltura superior, 3) calcular la serie de tiempo promedio de las envolturas, 4) Restar la serie de tiempo promedio de los datos para obtener el componente IMF, 5) Repita los pasos 1-4 utilizando como señal de inicio el residuo (la señal original menos el componente IMF) del paso anterior [22].

1.3.2.6.3 Vocal Fold Excitation Ratios (VFER)

Las relaciones de excitación de las cuerdas vocales son otro grupo de medidas de disfonía similares a GNE: los ciclos glóticos conducen a la excitación sincrónica de diferentes bandas de frecuencia y el ruido turbulento conduce a una excitación no correlacionada.

VFER trabaja directamente en los ciclos de las cuerdas vocales para cuantificar las relaciones de energía durante cada ciclo, su objetivo es cuantificar los fenómenos fisiológicos interactivos no lineales en la producción del habla de las cuerdas vocales y el tracto vocal.

1.3.2.6.4 Glottal Quotient (GQ)

Para este nuevo conjunto de medidas se utilizó el algoritmo DYPSA, teniendo en cuenta la cantidad de tiempo que las cuerdas vocales están separadas o en colisión. Este conjunto de medidas de disfonía en lugar de trabajar con estimaciones calculadas utilizando alguno de los algoritmos de detección de tono con ventana, trabaja con los ciclos glóticos estimados utilizando DYPSA como se muestra en la ecuación [22].

$$[VF\_close, VF\_open] = dypsa(data,fs) \tag{1. 5}$$

Se calcula la desviación estándar de la duración cuando la glotis está abierta (cuerdas vocales separadas) y cuando la glotis está cerrada (colisión de las cuerdas vocales). La razón es que, en las voces sanas, que son casi periódicas, los ciclos de plegamiento vocal no deben diferir considerablemente, y el intervalo en que la glotis está abierta o cerrada debe permanecer aproximadamente igual en todos los ciclos vocales de las fonaciones vocales sostenidas. Sin embargo, las voces patológicas se caracterizan por un aumento de la aperiodicidad, ya que se ve afectada la vibración normal de las cuerdas vocales [22].

2. METODOLOGÍA

En esta sección se detalla el procedimiento realizado para depurar la base de datos *PVA-dryrun* y para extraer las características de cada registro de audio utilizando el *Toolbox Voice Analysis*.

Se inicia con la descripción de la base de datos y del *Toolbox,* después se realiza el análisis del *toolbox* para encontrar las fórmulas matemáticas de las características, seguido por un estudio que permite determinar el tamaño que deben tener las muestras de voz, para una extracción de características confiable.

Finalmente se reconstruye el conjunto de datos PVA-dryrun, para lo cual se toma en cuenta el tamaño de la muestra de voz obtenido previamente y a cada uno de los 779 registros de voz depurados se los normaliza a ese tamaño y se extraen sus características.

2.1. DESCRIPCIÓN DE LA BASE DE DATOS Y DEL TOOLBOX

Son dos herramientas esenciales para el desarrollo del presente proyecto, por lo que se detallan a continuación:

2.1.1 BASE DE DATOS PVA-DRYRUN

El proyecto Análisis de Voz del Paciente (PVA), fue desarrollado para evaluar la gravedad y los cambios que ocurren en los síntomas de la EP. En la página web *https://www.synapse.org/#!Synapse:syn2321745/wiki/62077*, se encuentran 7 archivos en donde está localizada la información de 620 personas que padecen la enfermedad de Parkinson, generando un total de 779 grabaciones (algunas de ellas participaron varias veces). Los 7 archivos se detallan a continuación, basados en la referencia [9]:

- **PLM-PVA-data-dictionary.doc**. - en este documento adjunto en el anexo A se muestra una tabla con la descripción de cada campo de los archivos plmpva.csv, además del tipo de dato (entero, real, string), formato (YYYY/MM/DD HH:MM: SS) y unidades (segundos, Hz, muestras) dependiendo del campo. A continuación se explica el significado de cada campo:

- call_timestamp: fecha y hora del inicio de la llamada de voz.
- recording_duration: longitud de la grabación capturada por el sistema IVR[1].
- callref: número de referencia único para la grabación de voz.
- audio_duration: longitud del audio capturado por el sistema IVR durante la llamada.

[1] Interactive Voice Response (IVR) sistema de telefonía utilizado para recopilar información.

- voice_indexstart: índice de muestra en el que se detecta que inicia la voz.
- voice_indexend: índice de muestra en el que se detecta que termina la voz.
- voice_code: calidad de la grabación de voz: “ok” grabación utilizable, “bad” grabación inutilizable.
- voice_usable_duration: duración de la voz utilizable detectada.
- feature01-feature38: características extraídas de las grabaciones de voz.
- user_id_hashed: identificador del usuario.
- pvi_created_at: fecha y hora en la que el número de referencia único fue recogido por PatientLikeMe[2].
- current_age: edad del participante.
- sex: sexo del participante.
- years_since_first_symptom: años desde el primer síntoma de Parkinson detectado por el participante.
- pdrs_date: fecha en la cual se aplicó el cuestionario PDRS a los participantes.
- pdrs_reported_at: fecha y hora en la que el PDRS fue recogido por PatientLikeMe.
- days_from_pvi_to_pdrs: número de días transcurridos entre la fecha de referencia PDRS (pdrs_date) y el número único de referencia de llamada recopilado (pvi_created_at).
- pdrs_score: suma de todas las preguntas Q1-Q17 del cuestionario PDRS.
- Memory, hallucinations, mood, motivation, speech, saliva, swallowing, handwriting, cutting_food, dressing, hygiene, turning_in_bed, falling, freezing, walking, tremors, numbness: preguntas del cuestionario PDRS.
- Hoehn_yahr: etapa Hoehn y Yahr al momento de informar PDRS.
- On_treatment_id: si el paciente esta con medicamento cuando contestó el cuestionario PDRS.
- Calls_per_user: número de llamadas por participante.

[2] PatientsLikeMe es una red social en salud utilizada para que personas puedan compartir tratamientos y síntomas de sus enfermedades.

- **plmpva_test-NoPDRS.csv**. - es el conjunto de datos de prueba, con la información de 390 grabaciones, en donde las filas corresponden a los identificadores de cada grabación, y en las columnas se tiene diferentes campos, como características de la grabación (duración de la grabación, duración del audio, etc.), 38 características extraídas de las grabaciones sin procesar, edad, sexo, número de llamadas de cada paciente, etc. Pero no se encuentra el puntaje obtenido del cuestionario PDRS.

- **plmpva_test-WithPDRS.csv**. - en este conjunto de datos de prueba se encuentra la misma información que en plmpva_test-NoPDRS.csv incluyendo el puntaje PDRS.

- **plmpva_train.csv**. - es el conjunto de datos de entrenamiento, contiene la información de 389 grabaciones, con los mismos campos indicados en plmpva_test-WithPDRS.csv.

- **pva_wav-test.zip**. - se encuentran 390 grabaciones sin procesar (con silencios, saturación, etc). Cada grabación está constituida de la fonación de la vocal /a/ sostenida, todas las grabaciones son de personas que padecen la enfermedad de Parkinson.

- **pva_wav-train.zip**. - se encuentran 389 grabaciones sin procesar.

- **PLM-PDRS-Questionaire.pdf**. - el cuestionario PDRS es una versión abreviada de la Escala Unificada de Clasificación de la Enfermedad de Parkinson (UPDRS) ya que omite la sección de observación clínica, quedando así 17 preguntas de las 42 existentes en el UPDRS. Cada pregunta se contesta en una escala de 0 a 4; 0 representa ninguna discapacidad y 4 representa peor discapacidad.

  Además de las 17 preguntas el cuestionario PDRS incluye la escala Hoehn y Yahr que también describe la progresión de los síntomas de la EP, el cuestionario se encuentra en el anexo B.

2.1.2 TOOLBOX VOICE ANALYSIS

El toolbox voice analysis desarrollado por Athanasios Tsanas ('Thanasis') ubicado en la página web *https://people.maths.ox.ac.uk/tsanas/software.html*, contiene diferentes scripts. El script principal utilizado para la extracción de características se encuentra con el nombre de voice_analysis.m, al ser ejecutado requiere un argumento de entrada que es la grabación de voz (archivo .wav), o dos argumentos de entrada que son el audio digitalizado y la frecuencia de muestreo, obteniendo a la salida 339 características, y un vector columna

de la frecuencia fundamental F0 calculada cada 10 ms. A continuación, se muestra como llamar a la función:

- *General call: [*dysphonia_measures, dysphonia_measures_names, F0] = voice_analysis(data, fs)

- *Or: [*dysphonia_measures, dysphonia_measures_names, F0] = voice_analysis('a.wav')

2.2. EXTRACCIÓN DE LAS FÓRMULAS MATEMÁTICAS

Para encontrar las fórmulas matemáticas de las 339 características, se analizó los *scripts voice_analisys.m*, *melceps.m*, *wavedec_features.m, fastfda.m y rpde.m* en los cuales se encuentran las fórmulas en forma de código constituidas por varias funciones que se van a investigar en la herramienta MathWorks/Documentación con el fin de transformar los códigos a fórmulas matemáticas, la información necesaria para llevar a cabo este proceso se encuentra en la referencia [27].

A continuación, se muestra una tabla de las características y sus variantes:

Tabla 2.1. Nombres de las características y variantes

CARACTERÍSTICAS	VARIANTES
JITTER	• F0_abs_dif • F0_dif_percent • F0_PQ3_classical_Schoentgen • F0_PQ5_classical_Schoentgen • F0_PQ11_classical_Schoentgen • F0_PQ3_classical_Baken • F0_PQ5_classical_Baken • F0_PQ11_classical_Baken • F0_PQ3_generalised_Schoentgen • F0_PQ5_generalised_Schoentgen • F0_PQ11_generalised_Schoentgen • F0_abs0th_perturb • F0_DB • F0_CV • F0_TKEO_mean • F0_TKEO_std • F0_TKEO_prc5 • F0_TKEO_prc25

	• F0_TKEO_prc75 • F0_TKEO_prc95 • F0_FM • F0range_5_95_perc
SHIMMER	• A0_abs_dif • A0_dif_percent • A0_PQ3_classical_Schoentgen • A0_PQ5_classical_Schoentgen • A0_PQ11_classical_Schoentgen • A0_PQ3_classical_Baken • A0_PQ5_classical_Baken • A0_PQ11_classical_Baken • A0_PQ3_generalised_Schoentgen • A0_PQ5_generalised_Schoentgen • A0_PQ11_generalised_Schoentgen • A0_abs0th_perturb • A0_DB • A0_CV • A0_TKEO_mean • A0_TKEO_std • A0_TKEO_prc5 • A0_TKEO_prc25 • A0_TKEO_prc75 • A0_TKEO_prc95 • A0_FM • A0range_5_95_perc
HARMONICS TO NOISE RATIO (HNR)	• HNR_mean • HNR_std
NOISE TO HARMONICS RATIO (NHR)	• NHR_mean • NHR_std
GLOTTAL QUOTIENT (GQ)	• GQ-prc5_95 • GQ-std_cycle_open • GQ-std_cycle_closed
GLOTTAL TO NOISE EXCITATION RATIO (GNE)	• GNE-mean • GNE-std • GNE-SNR_TKEO • GNE-SNR_SEO • GNE-NSR_TKEO

	• GNE-NSR_SEO
VOCAL FOLD EXCITATION RATIOS (VFER)	• VFER-mean • VFER-std • VFER-entropy • VFER-SNR_TKEO • VFER-SNR_SEO • VFER-NSR_TKEO • VFER-NSR_SEO
EMPIRICAL MODE DESCOMPOSITION EXCITATION RATIOS (EMD-ER)	• IMF-SNR_SEO • IMF-SNR_TKEO • IMF-SNR_entropy • IMF-NSR_SEO • IMF-NSR_TKEO • IMF-NSR_entropy
MEL FREQUENCY CEPSTRAL COEFFICIENTS (MFCC) En esta familia de características, algunas variantes se agrupan y se representan en una sola fórmula, cambiando únicamente el número de coeficiente. Por ejemplo: Las variantes MFCC_0th, MFCC_1st, MFCC_2nd, MFCC_3rd, MFCC_4th, MFCC_5th, MFCC_6th, MFCC_7th, MFCC_8th, MFCC_9th, MFCC_10th, MFCC_11th y MFCC_12th se representan en una sola fórmula cambiando los coeficientes desde 0 hasta 12, para el cálculo de cada una de las variantes.	• mean_Log energy • std_Log energy • MFCC_0th • MFCC_1st • MFCC_2nd • MFCC_3rd • MFCC_4th • MFCC_5th • MFCC_6th • MFCC_7th • MFCC_8th • MFCC_9th • MFCC_10th • MFCC_11th • MFCC_12th • std_MFCC_0th • std_MFCC_1st • std_MFCC_2nd • std_MFCC_3rd • std_MFCC_4th • std_MFCC_5th • std_MFCC_6th • std_MFCC_7th • std_MFCC_8th • std_MFCC_9th • std_MFCC_10th • std_MFCC_11th • std_MFCC_12th

	• mean_delta log energy • mean_0th delta • mean_1st delta • mean_2nd delta • mean_3rd delta • mean_4th delta • mean_5th delta • mean_6th delta • mean_7th delta • mean_8th delta • mean_9th delta • mean_10th delta • mean_11th delta • mean_12th delta • std_delta log energy • std_0th delta • std_1st delta • std_2nd delta • std_3rd delta • std_4th delta • std_5th delta • std_6th delta • std_7th delta • std_8th delta • std_9th delta • std_10th delta • std_11th delta • std_12th delta • mean_delta delta log energy • mean_delta delta 0th • mean_1st delta delta • mean_2nd delta delta • mean_3rd delta delta • mean_4th delta delta • mean_5th delta delta • mean_6th delta delta • mean_7th delta delta • mean_8th delta delta • mean_9th delta delta • mean_10th delta delta • mean_11th delta delta • mean_12th delta delta

	• std_delta delta log energy • std_delta delta 0th • std_1st delta delta • std_2nd delta delta • std_3rd delta delta • std_4th delta delta • std_5th delta delta • std_6th delta delta • std_7th delta delta • std_8th delta delta • std_9th delta delta • std_10th delta delta • std_11th delta delta • std_12th delta delta
WAVELET En esta familia de características, algunas variantes se agrupan y se representan en una sola fórmula, cambiando únicamente el número de coeficiente.	• Ea • Ed_1_coef • Ed_2_coef • Ed_3_coef • Ed_4_coef • Ed_5_coef • Ed_6_coef • Ed_7_coef • Ed_8_coef • Ed_9_coef • Ed_10_coef • det_entropy_shannon_1_coef • det_entropy_shannon_2_coef • det_entropy_shannon_3_coef • det_entropy_shannon_4_coef • det_entropy_shannon_5_coef • det_entropy_shannon_6_coef • det_entropy_shannon_7_coef • det_entropy_shannon_8_coef • det_entropy_shannon_9_coef • det_entropy_shannon_10_coef • det_entropy_log_1_coef • det_entropy_log_2_coef • det_entropy_log_3_coef • det_entropy_log_4_coef • det_entropy_log_5_coef • det_entropy_log_6_coef • det_entropy_log_7_coef

	• det_entropy_log_8_coef • det_entropy_log_9_coef • det_entropy_log_10_coef • det_TKEO_mean_1_coef • det_TKEO_mean_2_coef • det_TKEO_mean_3_coef • det_TKEO_mean_4_coef • det_TKEO_mean_5_coef • det_TKEO_mean_6_coef • det_TKEO_mean_7_coef • det_TKEO_mean_8_coef • det_TKEO_mean_9_coef • det_TKEO_mean_10_coef • det_TKEO_std_1_coef • det_TKEO_std_2_coef • det_TKEO_std_3_coef • det_TKEO_std_4_coef • det_TKEO_std_5_coef • det_TKEO_std_6_coef • det_TKEO_std_7_coef • det_TKEO_std_8_coef • det_TKEO_std_9_coef • det_TKEO_std_10_coef • app_entropy_shannon_1_coef • app_entropy_shannon_2_coef • app_entropy_shannon_3_coef • app_entropy_shannon_4_coef • app_entropy_shannon_5_coef • app_entropy_shannon_6_coef • app_entropy_shannon_7_coef • app_entropy_shannon_8_coef • app_entropy_shannon_9_coef • app_entropy_shannon_10_coef • app_entropy_log_1_coef • app_entropy_log_2_coef • app_entropy_log_3_coef • app_entropy_log_4_coef • app_entropy_log_5_coef • app_entropy_log_6_coef • app_entropy_log_7_coef • app_entropy_log_8_coef • app_entropy_log_9_coef

	• app_entropy_log_10_coef • app_det_TKEO_mean_1_coef • app_det_TKEO_mean_2_coef • app_det_TKEO_mean_3_coef • app_det_TKEO_mean_4_coef • app_det_TKEO_mean_5_coef • app_det_TKEO_mean_6_coef • app_det_TKEO_mean_7_coef • app_det_TKEO_mean_8_coef • app_det_TKEO_mean_9_coef • app_det_TKEO_mean_10_coef • app_det_TKEO_std_1_coef • app_det_TKEO_std_2_coef • app_det_TKEO_std_3_coef • app_det_TKEO_std_4_coef • app_det_TKEO_std_5_coef • app_det_TKEO_std_6_coef • app_det_TKEO_std_7_coef • app_det_TKEO_std_8_coef • app_det_TKEO_std_9_coef • app_det_TKEO_std_10_coef • Ea2 • Ed2_1_coef • Ed2_2_coef • Ed2_3_coef • Ed2_4_coef • Ed2_5_coef • Ed2_6_coef • Ed2_7_coef • Ed2_8_coef • Ed2_9_coef • Ed2_10_coef • det_LT_entropy_shannon_1_coef • det_LT_entropy_shannon_2_coef • det_LT_entropy_shannon_3_coef • det_LT_entropy_shannon_4_coef • det_LT_entropy_shannon_5_coef • det_LT_entropy_shannon_6_coef • det_LT_entropy_shannon_7_coef • det_LT_entropy_shannon_8_coef • det_LT_entropy_shannon_9_coef • det_LT_entropy_shannon_10_coef

	• det_LT_entropy_log_1_coef • det_LT_entropy_log_2_coef • det_LT_entropy_log_3_coef • det_LT_entropy_log_4_coef • det_LT_entropy_log_5_coef • det_LT_entropy_log_6_coef • det_LT_entropy_log_7_coef • det_LT_entropy_log_8_coef • det_LT_entropy_log_9_coef • det_LT_entropy_log_10_coef • det_LT_TKEO_mean_1_coef • det_LT_TKEO_mean_2_coef • det_LT_TKEO_mean_3_coef • det_LT_TKEO_mean_4_coef • det_LT_TKEO_mean_5_coef • det_LT_TKEO_mean_6_coef • det_LT_TKEO_mean_7_coef • det_LT_TKEO_mean_8_coef • det_LT_TKEO_mean_9_coef • det_LT_TKEO_mean_10_coef • det_LT_TKEO_std_1_coef • det_LT_TKEO_std_2_coef • det_LT_TKEO_std_3_coef • det_LT_TKEO_std_4_coef • det_LT_TKEO_std_5_coef • det_LT_TKEO_std_6_coef • det_LT_TKEO_std_7_coef • det_LT_TKEO_std_8_coef • det_LT_TKEO_std_9_coef • det_LT_TKEO_std_10_coef • app_LT_entropy_shannon_1_coef • app_LT_entropy_shannon_2_coef • app_LT_entropy_shannon_3_coef • app_LT_entropy_shannon_4_coef • app_LT_entropy_shannon_5_coef • app_LT_entropy_shannon_6_coef • app_LT_entropy_shannon_7_coef • app_LT_entropy_shannon_8_coef • app_LT_entropy_shannon_9_coef • app_LT_entropy_shannon_10_coef • app_LT_entropy_log_1_coef • app_LT_entropy_log_2_coef

	• app_LT_entropy_log_3_coef • app_LT_entropy_log_4_coef • app_LT_entropy_log_5_coef • app_LT_entropy_log_6_coef • app_LT_entropy_log_7_coef • app_LT_entropy_log_8_coef • app_LT_entropy_log_9_coef • app_LT_entropy_log_10_coef • app_LT_TKEO_mean_1_coef • app_LT_TKEO_mean_2_coef • app_LT_TKEO_mean_3_coef • app_LT_TKEO_mean_4_coef • app_LT_TKEO_mean_5_coef • app_LT_TKEO_mean_6_coef • app_LT_TKEO_mean_7_coef • app_LT_TKEO_mean_8_coef • app_LT_TKEO_mean_9_coef • app_LT_TKEO_mean_10_coef • app_LT_TKEO_std_1_coef • app_LT_TKEO_std_2_coef • app_LT_TKEO_std_3_coef • app_LT_TKEO_std_4_coef • app_LT_TKEO_std_5_coef • app_LT_TKEO_std_6_coef • app_LT_TKEO_std_7_coef • app_LT_TKEO_std_8_coef • app_LT_TKEO_std_9_coef • app_LT_TKEO_std_10_coef
PITCH PERIOD ENTROPY (PPE)	• PPE
DENTRENDED FLUCTUATION ANALYSIS (DFA)	• DFA
RECURRENCE PERIOD DENSITY ENTROPY (RPDE)	• RPED

2.2.1 SCRIPT *voice_analysis.m*

En el análisis de los scripts se detalla el procedimiento realizado por Athanasios Tsanas para determinar cada característica. En este script se encuentran las características Jitter, Shimmer, HNR, NHR, GQ, GNE, VFER, EMD-ER y PPE con sus respectivas variantes.

El script *voice_analyis.m* se encuentra en el anexo C.

2.2.1.1 Jitter y Shimmer

Son dos grupos de medidas de disfonía que se diferencian únicamente en que: el jitter trabaja con la frecuencia fundamental (F0) y el shimmer con la amplitud (A0). Las funciones de MATLAB utilizadas para el cálculo de las características son:

- *Mean.* - calcula el promedio de un vector.
- *Abs.* - encuentra el valor absoluto.
- *Diff.* - calcula la diferencia entre valores adyacentes.
- *Aryule.* - calcula los coeficientes del modelo autorregresivo utilizando las ecuaciones Yule-Walker.
- *log10.*- calcula el logaritmo en base 10.
- *Prctile.* - devuelve el percentil de un vector de datos, dado un porcentaje.

Las variantes de las características de Jitter y Shimmer son:

- ✓ F0_abs_dif y A0_abs_dif

Para el cálculo de estas características, el autor del *toolbox* encontró el promedio del valor absoluto de la diferencia entre ciclos sucesivos de F0 para Jitter y de A0 para Shimmer. Se utiliza *swipep.m* para calcular F0 y *dypsa.m* para calcular A0. El código se encuentra en la línea 108 del anexo C.

$$F0\_abs\_dif = \frac{1}{N-1}\sum_{i=1}^{N-1}\left|F_{0,i} - F_{0,i+1}\right| \tag{2.1}$$

$$A0\_abs\_dif = \frac{1}{N-1}\sum_{i=1}^{N-1}\left|A_{0,i} - A_{0,i+1}\right| \tag{2.2}$$

Donde F0 es la frecuencia fundamental, A0 es la amplitud y N es la longitud de los vectores columna F0 y A0. F0, A0 y N tendrán el mismo significado en todas las fórmulas que se los mencione.

- ✓ F0_dif_percent y A0_dif_percent

Se expresa en porcentaje la división entre el promedio del valor absoluto de la diferencia entre ciclos sucesivos de F0 y el promedio de F0, se realiza lo mismo con A0. El código se encuentra en la línea 109 del anexo C.

$$F0\_abs\_dif = 100 * \frac{\frac{1}{N-1}\sum_{i=1}^{N-1}\left|F_{0,i} - F_{0,i+1}\right|}{\frac{1}{N}\sum_{i=1}^{N} F_{0,i}} \tag{2.3}$$

$$A0\_abs\_dif = 100 * \frac{\frac{1}{N-1}\sum_{i=1}^{N-1}\left|A_{0,i} - A_{0,i+1}\right|}{\frac{1}{N}\sum_{i=1}^{N} A_{0,i}} \tag{2.4}$$

- ✓ F0_PQ3_classical_Schoentgen, F0_PQ5_classical_Schoentgen, F0_PQ11_classical_Schoentgen, A0_PQ3_classical_Schoentgen, A0_PQ5_classical_Schoentgen y A0_PQ11_classical_Schoentgen.

El cociente de perturbación (PQ) ha sido utilizado porque se ha convertido en un intento para eliminar las correlaciones de las series temporales, es una medida de dispersión que evalúa la cantidad de fluctuación de fase en un intervalo de análisis [28]. Se han usado intervalos de 3, 5 y 11 ciclos, en donde el número de ciclos es representado por la letra k, útil para calcular ciertos límites de la señal a ser analizada: $k_1 = round\ (k/2)$ y $k_2 = k - k_1$. Por ejemplo en la característica F0_PQ3_classical_Schoentgen $k = 3$, que es el número que se encuentra junto a PQ.

Se tiene el vector columna F0, del cual se van a tomar 3 valores o ciclos ($k = 3$), el primer valor está ubicado en la posición dada por k_1 donde $k_1 = 2$ y los otros dos valores están ubicados consecutivamente (posición 3 y 4), a este vector de longitud 3 se le va a restar el valor que se encuentra en el centro del mismo vector (el valor de la posición 3), después se obtiene el valor absoluto del vector resultante de la resta, y se calcula su promedio. Este promedio se suma con los promedios de las siguientes iteraciones en las que habrá un desplazamiento de una posición en cada iteración; como ejemplo, las posiciones de la siguiente iteración quedarían así: el primer valor está en la posición 3 y los dos valores consecutivos en la posición 4 y 5, y así sucesivamente hasta $N - k_2$. Finalmente, la suma total se divide para $(N - k + 1)$ por el promedio del vector F0. Se realiza el mismo procedimiento utilizando el vector columna A0.

Se tomó como ejemplo 3 ($k = 3$) ciclos, pero también se calculan las características para $k = 5$ y para $k = 11$. El código se encuentra en las líneas 136 a 145 del anexo C.

$$F0\_PQ3\_classical\_Schoentgen = \frac{\frac{1}{N-k+1}\sum_{i=k_i}^{N-k_2}\left[\frac{1}{k}\sum_{j=i-k_2}^{i+k_2}\left|F_{0,j} - F_{0,i}\right|\right]}{\frac{1}{N}\sum_{i=1}^{N} F_{0,i}} \tag{2.5}$$

$$A0\_PQ3\_classical\_Schoentgen = \frac{\frac{1}{N-k+1}\sum_{i=k_i}^{N-k_2}\left[\frac{1}{k}\sum_{j=i-k_2}^{i+k_2}\left|A_{0,j} - A_{0,i}\right|\right]}{\frac{1}{N}\sum_{i=1}^{N} A_{0,i}} \tag{2.6}$$

- ✓ F0_PQ3_classical_Baken, F0_PQ5_classical_Baken, F0_PQ11_classical_Baken, A0_PQ3_classical_Baken, A0_PQ5_classical_Baken y A0_PQ11_classical_Baken

Se tiene el vector columna F0, del cual se van a tomar 5 valores o ciclos ($k = 5$), el primer valor está ubicado en la posición dada por k_1 donde $k_1 = 3$ y los cuatro valores restantes están ubicados consecutivamente (posiciones 4, 5, 6 y 7), de éste vector de longitud 5 se saca el valor absoluto y luego el promedio, al resultado obtenido se le resta el valor ubicado en el centro del mismo vector (el valor de la posición 5), después la diferencia se suma con las diferencias resultantes de todas las iteraciones, en cada iteración habrá un desplazamiento de una posición; por ejemplo, en la siguiente iteración el primero valor estará en la posición 4 y los cuatro valores consecutivos estarán en las posiciones 5, 6, 7 y 8 y así sucesivamente hasta $N - k_2$, siendo $k_2 = k - k_1$. Finalmente, la suma total se dividirá para $(N - k + 1)$ por el promedio del vector F0. Se realiza el mismo procedimiento con el vector columna A0.

Se calculan las características para $k = 3$, $k = 5$ y $k = 11$. El código se encuentra en las líneas 146 a 150 del anexo C.

$$F0\_PQ3\_classical\_Baken = \frac{\frac{1}{N-k+1}\sum_{i=k_1}^{N-k_2}\left[\frac{1}{k}\sum_{j=i-k_2}^{i+k_2}\left|F_{0,j}\right| - F_{0,i}\right]}{\frac{1}{N}\sum_{i=1}^{N} F_{0,i}} \tag{2.7}$$

$$A0\_PQ3\_classical\_Baken = \frac{\frac{1}{N-k+1}\sum_{i=k_1}^{N-k_2}\left[\left|\frac{1}{k}\sum_{j=i-k_2}^{i+k_2} A_{0,j}\right| - A_{0,i}\right]}{\frac{1}{N}\sum_{i=1}^{N} A_{0,i}} \tag{2.8}$$

- ✓ F0_PQ3_generalised_Schoentgen, F0_PQ5_generalised_Schoentgen, F0_PQ11_generalised_Schoentgen, A0_PQ3_generalised_Schoentgen, A0_PQ5_generalised_Schoentgen y A0_PQ11_generalised_Schoentgen

Al vector columna F0 se le resta el promedio del mismo vector. De este nuevo vector resultado de la resta que se llamara new_data se van a calcular 5 coeficientes del modelo autoregresivo $\{a_j\}_{j=1}^{p}, p = 5$ utilizando la función *aryule* la cual retorna un vector de dimensión 1x6 en donde el primer valor siempre será uno, y los otros 5 valores serán los coeficientes calculados.

Del vector new_data se toman $1 + p$ valores (6 valores), empezando por el valor que se encuentra en la posición $1 + p$ (posición 6), y así en forma descendente hasta completar los 6 valores (posición 5, 4, 3, 2 y 1). Una vez seleccionados los 6 valores se los multiplica por el vector de coeficientes, los elementos del vector resultante de la multiplicación se suman y a ese valor final se le saca el valor absoluto, siendo esta la respuesta de la primera iteración, la cual se suma con las respuestas de las siguientes iteraciones. Para la segunda iteración se desplaza una posición y se seleccionan los valores del vector new_data, ahora la primera posición será 7, y las posiciones descendentes de los 6 elementos serán 6, 5 ,4, 3 y 2, así sucesivamente hasta llegar a la posición dada por la longitud del vector columna F0.

Luego de sumar los resultados de todas las iteraciones la respuesta se divide para $(N - p)$ por el promedio del vector columna F0. Se realiza el mismo procedimiento con el vector columna A0.

Las 3 características de $F0\_generalised\_Schoentgen$ tienen el mismo valor ya que p siempre va a ser 5, lo mismo pasa con las 3 características de $A0\_generalised\_Schoentgen$. El código se encuentra en las líneas 151 a 158 del anexo C.

$$F0\_PQ3\_generalised\_Schoentgen = \frac{\frac{1}{N-p}\sum_{i=p+1}^{N}\left|\sum_{j=i-p}^{i} a_j.\left(F_{0,j} - \frac{1}{N}\sum_{i=1}^{N} F_{0,i}\right)\right|}{\frac{1}{N}\sum_{i=1}^{N} F_{0,i}} \tag{2.9}$$

$$A0\_PQ3\_generalised\_Schoentgen = \frac{\frac{1}{N-p}\sum_{i=p+1}^{N}\left|\sum_{j=i-p}^{i} a_j.\left(A_{0,j} - \frac{1}{N}\sum_{i=1}^{N} A_{0,i}\right)\right|}{\frac{1}{N}\sum_{i=1}^{N} A_{0,i}} \tag{2.10}$$

- ✓ F0_abs0th_perturb y A0_abs0th_perturb

Se encuentra la diferencia entre el vector columna F0 y el promedio del mismo vector, después se calcula el valor absoluto del resultado obtenido de la diferencia y finalmente se calcula el promedio. Lo mismo se realiza con vector columna A0. El código se encuentra en la línea 122 del anexo C.

$$F0\_abs0th\_perturb = \frac{1}{N}\sum_{i=1}^{N}\left|F_{0,i} - \frac{1}{N}\sum_{J=1}^{N}F_{0,j}\right| \tag{2.11}$$

$$A0\_abs0th\_perturb = \frac{1}{N}\sum_{i=1}^{N}\left|A_{0,i} - \frac{1}{N}\sum_{J=1}^{N}A_{0,j}\right| \tag{2.12}$$

- ✓ F0_DB y A0_DB

Originalmente se tiene el vector columna F0 de dimensiones Nx1, del cual se va a tomar desde 1 hasta N-1 para el numerador y desde 2 hasta N para el denominador y se efectuará la división, para después encontrar el logaritmo en base 10 del resultado. De este nuevo resultado se obtiene el valor absoluto, se multiplicará por 20 y finalmente se calcula el promedio. Se realiza lo mismo con el vector columna A0. El código se encuentra en la línea 123 del anexo C.

$$F0\_DB = \frac{1}{N-1}\sum_{i=1}^{N-1}\left[20\left|log_{10}\frac{F_{0,i}}{F_{0,i+1}}\right|\right] \tag{2.13}$$

$$A0\_DB = \frac{1}{N-1}\sum_{i=1}^{N-1}\left[20\left|log_{10}\frac{A_{0,i}}{A_{0,i+1}}\right|\right] \tag{2.14}$$

- ✓ F0_CV y A0_CV

Para el cálculo de esta medida de disfonía o característica se encuentra la diferencia de valores adyacentes de F0, esta diferencia se eleva al cuadrado y se calcula el promedio, el cual será dividido para el promedio del vector columna F0 elevado al cuadrado. Se realiza el mismo procedimiento para el vector columna A0. El código se encuentra en la línea 124 del anexo C.

$$F0\_CV = \frac{\frac{1}{N-1}\sum_{i=1}^{N-1}\left(F_{0,i} - F_{0,i+1}\right)^2}{\left(\frac{1}{N}\sum_{i=1}^{N}F_{0,i}\right)^2} \tag{2.15}$$

$$A0\_CV = \frac{\frac{1}{N-1}\sum_{i=1}^{N-1}\left(A_{0,i} - A_{0,i+1}\right)^2}{\left(\frac{1}{N}\sum_{i=1}^{N} A_{0,i}\right)^2} \tag{2.16}$$

- ✓ F0_TKEO_mean y A0_TKEO_mean

Teager-Kaiser energy operator (TKEO), mide los cambios instantáneos de energía de las señales [22]. Para el cálculo de TKEO se tienen un vector columna de la señal x que se quiere analizar de dimensiones Nx1, el primero elemento que conforma el vector TKEO es el primer elemento de la señal x elevado al cuadrado ${x_{0,1}}^2$, para calcular el segundo elemento, se toma el segundo elemento de x se lo eleva al cuadrado y se resta el producto entre el elemento de la posición anterior $x_{0,i-1}$ y el elemento de la siguiente posición $x_{0,i+1}$, y así sucesivamente hasta N-1. El último elemento que conforma el vector TKEO es el último elemento de la señal x elevado al cuadrado ${x_{0,N}}^2$. El código se encuentra las líneas 343 a 349 del anexo C.

$$TKEO\ (x) = [{x_{0,1}}^2 \quad ({x_{0,i}}^2 - x_{0,i-1} * x_{0,i+1})_{i=2}^{N-1} \quad {x_{0,N}}^2]$$

Para encontrar la medida de disfonía en cuestión se calcula el TKEO del vector columna F0, luego se encuentra el valor absoluto y finalmente el promedio. Se realiza lo mismo con el vector columna A0. El código se encuentra en la línea 125 del anexo C.

$$F0\_TKEO\_mean = \frac{1}{N}\left({F_{0,1}}^2 + \sum_{i=2}^{N-1}\left|{F_{0,i}}^2 - F_{0,i-1} * F_{0,i+1}\right| + {F_{0,N}}^2\right) \tag{2.17}$$

$$F0\_TKEO\_mean = \frac{1}{N}\left({A_{0,1}}^2 + \sum_{i=2}^{N-1}\left|{A_{0,i}}^2 - A_{0,i-1} * A_{0,i+1}\right| + {A_{0,N}}^2\right) \tag{2.18}$$

- ✓ F0_TKEO_std y A0_TKEO_std

La desviación estándar cuantifica la variación que existe en un conjunto de datos. Para su cálculo a cada elemento del vector columna F0 se le resta el promedio del mismo vector, luego se saca el valor absoluto y se eleva al cuadrado. Del resultado obtenido se calcula el promedio y finalmente la raíz cuadrada [29].

$$desviación\ estándar = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left|F_{0,i} - \frac{1}{N}\sum_{i=1}^{N} F_{0,i}\right|^2}$$

Para el cálculo de la característica, primero se calcula el TKEO de F0, y luego la desviación estándar. Se realiza el mismo procedimiento con el vector columna A0. El código se encuentra en la línea 126 del anexo C.

$$F0\_TKEO\_sdt = \sqrt{\frac{\left|\left({F_{0,1}}^2 + \sum_{i=2}^{N-1}\left|{F_{0,i}}^2 - F_{0,i-1} * F_{0,i+1}\right| + {F_{0,N}}^2\right) - \frac{1}{N}\sum_{i=1}^{N} F_{0,i}\right|^2}{N-1}} \tag{2.19}$$

$$F0\_TKEO\_sdt = \sqrt{\frac{\left|\left({A_{0,1}}^2 + \sum_{i=2}^{N-1}\left|{A_{0,i}}^2 - A_{0,i-1} * A_{0,i+1}\right| + {A_{0,N}}^2\right) - \frac{1}{N}\sum_{i=1}^{N} A_{0,i}\right|^2}{N-1}} \tag{2.20}$$

- ✓ F0_TKEO_prc5, F0_TKEO_prc25, F0_TKEO_prc75, F0_TKEO_prc95, A0_TKEO_prc5, A0_TKEO_prc25, A0_TKEO_prc75 y A0_TKEO_prc95

La fórmula empleada para el cálculo de percentiles es:

$$percentil = \frac{N * porcentaje\ dado}{100} + 0{,}5 = K, f$$

Donde K es la parte entera de la respuesta y f es la parte decimal [30].

Para el cálculo de la característica primero se ordena de menor a mayor el vector TKEO de F0, después se calcula el percentil utilizando la fórmula mencionada previamente, para obtener K y f. Con estos dos valores se realizan dos multiplicaciones la primera es el valor de TKEO ubicado en la posición K por $(1 - f)$ y la otra multiplicación es el valor de TKEO ubicado en la posición $(K + 1)$ por f, finalmente los dos productos se suman.

Se realiza el mismo procedimiento para los porcentajes 5, 25, 75 y 95. El código se encuentra en la línea 127 del anexo C.

$$prc5 = \frac{N * 5}{100} + 0{,}5 = K, f \tag{2.21}$$

$$F0\_TKEO\_prc5 = (1 - f) * TKEO_K + f * TKEO_{K+1}$$

$$A0\_TKEO\_prc5 = (1 - f) * TKEO\_A0_K + f * TKEO\_A0_{K+1} \tag{2.22}$$

- ✓ F0_FM y A0_FM

Del vector columna F0 se encuentra el máximo valor y el mínimo valor, luego se resta el máximo menos el mínimo y ese resultado se divide para la suma entre el máximo y el

mínimo. Se realiza el mismo procedimiento con el vector columna A0. El código se encuentra en la línea 132 del anexo C.

$$F0\_FM = \frac{max(F_{0,i})_{i=1}^{N} - \min(F_{0,i})_{i=1}^{N}}{max(F_{0,i})_{i=1}^{N} + \min(F_{0,i})_{i=1}^{N}} \tag{2.23}$$

$$A0\_FM = \frac{max(A_{0,i})_{i=1}^{N} - \min(A_{0,i})_{i=1}^{N}}{max(A_{0,i})_{i=1}^{N} + \min(A_{0,i})_{i=1}^{N}} \tag{2.24}$$

- ✓ F0range_5_95_perc y A0range_5_95_perc

Se restan dos características que ya han sido obtenidas previamente, estas son: F0_TKEO_prc95 menos F0_TKEO_prc5. Se realiza lo mismo con las características A0_TKEO_prc95 y A0_TKEO_prc5. El código se encuentra en la línea 133 del anexo C.

$$F0range\_5\_95\_prc = F0\_TKEO\_prc95 - F0\_TKEO\_prc95 \tag{2.25}$$

$$F0range\_5\_95\_prc = F0\_TKEO\_prc95 - F0\_TKEO\_prc5 \tag{2.26}$$

2.2.1.2 Harmonics to Noise Ratio (HNR) y Noise to Harmonics Ratio (NHR)

Entre las funciones de Matlab más importantes para el cálculo de las características HNR y NHR se tiene:

- *Hann.* - devuelve una ventana Hann simétrica
- *Xcorr.* - estima la correlación entre dos secuencias aleatorias
- *Fft.* - calcula la transformada discreta de Fourier
- *Ifft.* - calcula la transformada discreta de Fourier inversa
- *Sort.* - ordena los elementos de un vector o una matriz de menor a mayor
- *Ceil.* - redondea al entero superior cada elemento del vector de entrada
- *Floor.* - redondea al entero inferior
- *Audioread.* - devuelve el audio digitalizado

Las variantes de las características HNR y NHR son:

- ✓ HNR_mean

Utilizando la función *audioread,* se obtiene un vector del audio digitalizado de longitud N, del valor que tenga N se resta una ventana de 640 muestras (80ms) y se divide para 80 muestras (10ms) que es el tamaño del solapamiento. El resultado indica el número de iteraciones.

Ahora, del vector que contiene el audio digitalizado se toma una trama de tamaño de la ventana y se resta el promedio de esa trama, resultado que se multiplica por una ventana hann y se lo llama segment_sig. Después se calcula la autocorrelación del vector segment_sig con la ecuación (1.1) y se lo divide para la autocorrelación de la ventana hann.

Al resultado de la división se lo llamará C, el cual se ordenará en forma descendente, y se generará un vector con las posiciones originales del vector C antes de ser ordenado, ubicadas en la misma posición que ahora se encuentran los valores ordenados. Luego se realiza una comparación, en donde se va analizando cada elemento del vector con las posiciones, el cual tiene que ser menor a 16 o mayor a 160, si uno de ellos no cumple estos límites, el ultimo valor que cumple con los límites indicará la posición del valor que se tomará del vector C.

Para calcular las características en cuestión, del vector C se toma el valor ubicado en la posición que se indique realizando el procedimiento previo y se lo divide para el resultado de restar uno menos el valor del numerador. Finalmente se multiplica 10 por el logaritmo en base 10 del resultado de la división. Para la siguiente iteración, del inicio de la primera trama se solapan 80 muestras (10ms) y se toma otra ventana de 640 muestras (80ms), y así sucesivamente. Al final de todas las iteraciones se calcula el promedio. El código se encuentra en las líneas 161 a 201 del anexo C.

$$HNR_{mean} = mean\left(10\log\left[\frac{r_{xx}(l_{max})}{1 - r_{xx}(l_{max})}\right]\right) \tag{2.27}$$

- ✓ HNR_std

La diferencia del cálculo de la característica anterior radica en que al final de todas las iteraciones se calcula la desviación estándar. El código se encuentra en la línea 202 del anexo C.

$$HNR_{std} = std\left(10\log\left[\frac{r_{xx}(l_{max})}{1 - r_{xx}(l_{max})}\right]\right) \tag{2.28}$$

- ✓ NHR_mean

De las características anteriores vamos a retomar desde la parte en la que se encuentra la posición del valor que se toma del vector C. Para el cálculo la característica se resta uno menos el valor del vector C y se dividirá para el valor del vector C. Se repite sucesivamente el mismo procedimiento para todas las iteraciones, como se explicó previamente y finalmente se calcula el promedio. El código se encuentra en la línea 203 del anexo C.

$$NHR_{mean} = mean\left(\frac{1 - r_{xx}(l_{max})}{r_{xx}(l_{max})}\right) \tag{2.29}$$

- ✓ NHR_std

Similar a la característica NHR_mean, pero en vez de calcular el promedio al final se calcula la desviación estándar. El código se encuentra en la línea 204 del anexo C.

$$NHR_{std} = std\left(\frac{1 - R_{xx}(l_{max})}{R_{xx}(l_{max})}\right) \tag{2.30}$$

2.2.1.3 Glottal Quotient (GQ)

Una de las funciones utilizadas para el cálculo de este grupo de características es:

- *Nanstd*.- que es la desviación estándar ignorando valores NAN, otras funciones que se utilizan ya se explicaron previamente.

Utilizando el script *dypsa.m* que se encuentra en el *toolbox* se calcula la apertura glótica VF_open y el cierre glótico VF_close, los cuales son necesarios para encontrar el vector $cycle_{open}$ y el vector $cycle_{closed}$.

$$cycle_{open} = \left|(VF\_open_i)_{i=2}^{end} - (VF\_close_i)_{i=1}^{end-1}\right|$$

$$cycle_{closed} = \left|(VF\_open_i)_{i=1}^{end-1} - (VF\_close_i)_{i=2}^{end}\right|$$

Del vector $cycle_{open}$ se va a analizar cada uno de los elementos que lo componen, si el elemento es mayor a 160 o menor a 16 se va a cambiar el valor por NAN caso contrario se mantiene el mismo valor del vector. Se realiza el mismo procedimiento con el vector $cycle_{closed}$. El código se encuentra en las líneas 207 a 221 del anexo C.

Después de analizar los elementos y cambiar a NAN si es necesario, se calcula el percentil con 5% y con 95% del vector $cycle_{open}$ y del vector $cycle_{closed}$.

Las variantes de la característica GQ son:

- ✓ GQ-prc5_95

Para calcular la característica se realiza una división, en el numerador se tiene el percentil del $cycle_{open}$ al 95% menos el percentil del $cycle_{open}$ al 5%, mientras que en el denominador se tiene la resta de los percentiles 95% - 5% del vector $cycle_{open}$ más la resta de los dos percentiles pero del vector $cycle_{closed}$. El código se encuentra en las líneas 222 a 226 del anexo C.

$$GQ_{prc5\_95} = \frac{X_{cycle\_open\_prc95} - X_{cycle\_open\_prc5}}{\left(X_{cycle\_open\_prc95} - X_{cycle\_open\_prc5}\right) + \left(X_{cycle\_closed\_prc95} - X_{cycle\_closed\_prc5}\right)} \tag{2.31}$$

- ✓ GQ-std_cycle_open

Se realiza el cálculo de la desviación estándar ignorando los valores NAN del vector $cycle_{open}$, una vez que se comprobó si los elementos del vector son mayores a 160 o menores a 16, como se explicó previamente. El código se encuentra en la línea 227 del anexo C.

$$GQ_{std\_cycle\_open} = nanstd(cycle\_open) \tag{2.32}$$

- ✓ GQ-std_cycle_closed

Se calcula la desviación estándar ignorando los valores NAN del vector $cycle_{closed}$. El código se encuentra en la línea 228 del anexo C.

$$GQ_{std\_cycle\_closed} = nanstd(cycle\_closed) \tag{2.33}$$

2.2.1.4 Glottal to Noise Excitation ratio (GNE)

Para realizar los cálculos de este grupo de medidas de disfonía, se utilizan algunas funciones de Matlab que son:

- *Resample*. - remuestrear la señal de entrada a una frecuencia diferente.
- *Lpc.* - encuentra los coeficientes de filtro de predicción lineal, es decir predice los valores actuales de la serie basándose en muestras pasadas.
- *Log.* - calcula el logaritmo natural.
- *Hilbert.* - es una función que permite calcular la transformada de hilbert para una secuencia de entrada real y devuelve un resultado complejo de la misma longitud.

- *Filter.* - filtra los datos de entrada utilizando una función de transferencia racional, definida por los coeficientes del numerador y del denominador.
- *fdesign.bandpass.*- sirve para construir un objeto con especificaciones de filtro, como parámetros de entrada se tiene el orden del filtro, las frecuencias de corte y la frecuencia de muestreo.
- *Design.* - diseña filtros pasabanda especificado por la función *fdesign.bandpass*.

Se van a generar en total 7 filtros, la frecuencia de corte inicia es 1, con un ancho de banda de 1000, la frecuencia de corte final es 1001, luego se desplaza 500 y se toma la frecuencia de corte inicial del segundo filtro que sería 501, y la frecuencia de corte final sería 1501 y así sucesivamente.

Las variantes de la característica GNE son:

- ✓ GNE-mean

Inicialmente se remuestrea la señal de entrada a una nueva frecuencia (10kHz), de la longitud de la señal remuestreada se resta el tamaño de la ventana de 30ms (300 muestras) y se divide para la longitud del solapamiento que es de 10ms (100 muestras), esto para encontrar el número de iteraciones que se deben realizar *steps*. Para la primera iteración, de la señal remuestreada se toma desde 1 hasta 300 muestras que es el tamaño de la ventana, después se multiplica por una ventana hann del mismo tamaño y se obtiene un vector que se lo llama segment_sig.

Ahora utilizando la función *lpc*, se calcula un vector con 13 coeficientes al cual se le añade el valor uno al inicio del vector, quedando un vector con 14 coeficientes. Coeficientes que servirán como argumento de entrada para el filtro que calcula la estimación de la señal segment_sig, después se resta de la señal segment_sig la estimación calculada y del resultado se obtiene la autorrelación.

Como siguiente paso se aplican los 7 filtros obtenidos del diseño, al vector resultado de la autorrelación, obteniendo una matriz sigBW de dimensiones 300x7. El código se encuentra en las líneas 231 a 255 del anexo C.

Continuando con la explicación de la primera iteración, se aplica la función *hilbert* a la matriz sigBW, obteniendo una matriz con valores complejos, para obtener valores reales se calcula el valor absoluto. De la matriz con valores reales se calcula la autocorrelación y se encuentra el máximo valor que se lo llamará GNEm.

Para la siguiente iteración, se realiza un solapamiento de 100 muestras, entonces de la señal remuestreada se toma desde 201 hasta 500 el tamaño de la ventana, y así sucesivamente hasta *steps*+1.

Finalmente se calcula el promedio del vector con todos los valores GNEm encontrados en cada iteración. El código se encuentra en las líneas 255 a 269 del anexo C.

$$GNE_{mean} \approx mean[\max(xcorr|hilbert(sigBW)|)]_1^{steps+1} \tag{2.34}$$

- ✓ GNE-std

Se encuentra la desviación estándar del vector con todos los valores GNEm calculados en cada iteración. El código se encuentra en la línea 270 del anexo C.

$$GNE_{std} \approx std[\max(xcorr|hilbert(sigBW)|)]_1^{steps+1} \tag{2.35}$$

- ✓ GNE-SNR_TKEO

En esta característica primero se calcula el TKEO de las 7 columnas de la matriz sigBW y luego el promedio, al resultado de todas las columnas se lo llamará sig_TKEO. Se realizará el mismo procedimiento para todas las iteraciones desde 1 hasta step+1, obteniendo una matriz de dimensiones (step+1) x7, de la cual se saca el promedio, quedando como resultado un vector de dimensiones 1x7 llamado gnTKEO.

Finalmente, para obtener la característica se realiza una división, del vector gnTKEO se toman los valores de 1 a 2 y se los suma, el resultado se divide para la suma de los valores de 4 a 7 del mismo vector gnTKEO. El código se encuentra en las líneas 271 a 273 del anexo C.

$$GNE_{SNR\_TKEO} \approx \frac{\sum\{mean[mean(TKEO(sigBW))]_1^7\}_1^2}{\sum\{mean[mean(TKEO(sigBW))]_1^7\}_{end-3}^{end}} \tag{2.36}$$

- ✓ GNE-SNR_SEO

Cada columna de la matriz sigBW se eleva al cuadrado y luego se calcula el promedio, al resultado que se lo llama sig_energy. Se repetirá el procedimiento hasta step+1, obteniendo una matriz de dimensiones (step+1)x7, de la cual se encuentra el promedio, quedando como resultado un vector de dimensiones 1x7 llamado gnSEO.

Del vector gnSEO se suman los valores ubicados en la posición 1 y 2 y el resultado se divide para la suma de los valores ubicados en la posición 4, 5, 6 y 7. El código se encuentra en las líneas 272 a 274 del anexo C.

$$GNE_{SNR\_SEO} \approx \frac{\sum\{mean[(mean(sigBW))^2]_1^7\}_1^2}{\sum\{mean[(mean(sigBW))^2]_1^7\}_{end-3}^{end}} \tag{2.37}$$

- ✓ GNE-NSR_TKEO

Se partirá del vector sig_TKEO, el cual luego de que se realicen todas las iteraciones desde 1 hasta step+1, se convertirá en una matriz de dimensiones (step+1)x7. De la cual se calcula el logaritmo natural y posteriormente el promedio, al resultado se lo llama signal_BW_TKEO2 que será un vector de dimensiones 1x7.

Finalmente se realizará una división. Del vector signal_BW_TKEO2 se toman los valores 4, 5, 6 y 7 y se los suma, el resultado se divide para la suma de los valores 1 y 2 del mismo vector signal_BW_TKEO2. El código se encuentra en la línea 275 del anexo C.

$$GNE_{NSR\_TKEO} \approx \frac{\sum\{mean[\ln(mean(TKEO(sigBW)))]_1^7\}_{end-3}^{end}}{\sum\{mean[\ln(mean(TKEO(sigBW)))]_1^7\}_1^2} \tag{2.38}$$

- ✓ GNE-NSR_SEO

Se partirá del vector sig_energy, el cual luego de que se realicen todas las iteraciones desde 1 hasta step+1, se convertirá en una matriz de dimensiones (step+1)x7. De la cual se calcula el logaritmo natural y posteriormente el promedio, encontrando el vector signal_energy2.

Finalmente se realiza una división. Del vector signal_energy2 se toman los valores 4, 5, 6 y 7 y se los suma, el resultado se divide para la suma de los valores 1 y 2 del mismo vector signal_energy2. El código se encuentra en la línea 276 del anexo C.

$$GNE_{NSR\_SEO} \approx \frac{\sum\{mean[\ln((mean(sigBW))^2)]_1^7\}_{end-3}^{end}}{\sum\{mean[ln(mean(sigBW))^2]_1^7\}_1^2} \tag{2.39}$$

2.2.1.5 Vocal Fold Excitation Ratios (VFER)

Son una familia de medidas de disfonía que cuantifican las relaciones de energía durante cada ciclo [22]. Las funciones de Matlab utilizadas para su cálculo, son las mismas ya especificadas en características anteriores.

Para encontrar las variables de VFER, se utilizan 40 filtros pasabanda creados utilizando las funciones fdesign.bandpass y design de Matlab. Para el primer filtro la frecuencia de corte inicial es 1, si se tiene un ancho de banda de 500 la frecuencia de corte final será de 500. Para el segundo filtro la frecuencia de corte inicia es de 81 y la frecuencia de corte final es de 581. Y así sucesivamente con saltos de frecuencia de 80, hasta que la frecuencia de corte inicial sea 3200 que es la frecuencia máxima, o a su vez que se tengan los 40 filtros.

Las variantes de la característica VFER son:

- ✓ VFER-mean

En esta característica también se utiliza el algoritmo dypsa.m para calcular VF_close y VF_open. La longitud del vector VF_close menos uno indica el número de iteraciones que se debe realizar el procedimiento que se especifica a continuación.

Se tiene un vector del audio digitalizado que se desea analizar (data), para la primera iteración el valor del vector VF_close ubicado en la primera posición indica desde donde se van a tomar los valores del vector data, hasta el valor del vector VF_close ubicado en la segunda posición, este segmento extraído del vector data será multiplicado por una ventana hann del mismo tamaño, al producto se lo llamará segment_sig. Si el tamaño del vector segment_sig es mayor a 40, se aplican uno por uno los 40 filtros al vector segment_sig, del resultado se toma desde 1 hasta 40 y a este nuevo vector se lo llama sigBW, el cual al finalizar las 40 iteraciones se convertirá en una matriz. Después se aplica la función *hilbert* a la matriz sigBW, se saca el valor absoluto, se calcula la autocorrelación y se encuentra el máximo valor, que se lo conocerá como NEm. Caso contrario si el tamaño del vector segment_sig no es mayor a 40, no se realiza el procedimiento detallado previamente y se pasa a la siguiente iteración. Para la siguiente iteración del vector data se toman los valores que indique el valor de VF_close ubicado en la segunda posición hasta el valor ubicado en la tercera posición, y así sucesivamente. El código se encuentra en las líneas 279 a 305 del anexo C.

Luego de realizar todas las iteraciones desde 1 hasta la longitud de VF_close menos 1. Se tiene un vector con todos los NEm resultantes, del cual se calcula el promedio. El código se encuentra en la línea 311 del anexo C.

$$VFER_{mean} \approx mean[max(xcorr|hilbert(sigVBW)|)]_1^{length(VF\_close)} \quad (2.40)$$

- ✓ VFER-std

Se calcula la desviación estándar del vector con los valores NEm obtenidos después de realizar todas las iteraciones. El código se encuentra en la línea 312 del anexo C.

$$VFER_{std} \approx std[max(xcorr|hilbert(sigVBW)|)]_1^{length(VF\_close)} \tag{2.41}$$

- ✓ VFER-entropy

Se calcula el logaritmo natural del vector NEm, el resultado se multiplica por el vector NEm, se suman los valores contenidos en el vector y al resultado final se le antepone el signo negativo. El código se encuentra en la línea 313 del anexo C.

$$VFER_{entropy} \approx -\sum([max(xcorr|hilbert(sigVBW)|)] * ln[max(xcorr|hilbert(sigVBW)|)]) \tag{2.42}$$

- ✓ VFER-SNR_TKEO

Se calcula el TKEO de las 40 columnas de la matriz sigBW y luego el promedio, generando un nuevo vector sig_TKEO, el cual luego de realizar todas las iteraciones desde 1 hasta la longitud de VF_close menos uno, se transforma en una matriz de dimensiones (longitud de VF_close -1)x40. A continuación se encuentra el promedio de la matriz y se lo llama VFTKEO, el cual es un vector de dimensiones 1x40.

Del vector VFTKEO se suman los valores que se encuentran en las posiciones del 1 al 5 y el resultado se divide para la suma de los valores ubicados en las posiciones del 6 al 10 del mismo vector VFTKEO. El código se encuentra en las líneas 306 a 317 del anexo C.

$$VFER_{SNR\_TKEO} \approx \frac{\sum\{mean[mean(TKEO(sigVBW))]\}_1^5}{\sum\{mean[mean(TKEO(sigVBW))]\}_6^{10}} \tag{2.43}$$

- ✓ VFER-SNR_SEO

Cada columna de la matriz sigBW se eleva al cuadrado y se calcula el promedio, para encontrar el vector sig_SEO, vector que después de todas las iteraciones se convertirá en una matriz, de la cual se calcula el promedio para encontrar el vector VFSEO.

Del vector VFSEO se suman los valores que se encuentran en las posiciones del 1 al 5 y el resultado se divide para la suma de los valores ubicados en las posiciones del 6 al 10 del mismo vector VFSEO. El código se encuentra en las líneas 307 a 318 del anexo C.

$$VFER_{SNR\_SEO} \approx \frac{\sum\{mean[(mean(sigVBW))^2]\}_1^5}{\sum\{mean[(mean(sigVBW))^2]\}_6^{10}} \tag{2.44}$$

- ✓ VFER-NSR_TKEO

El vector sig_TKEO después de todas las iteraciones se convierte en una matriz, de la cual se calcula el logaritmo natural y el promedio, al vector resultante se lo llamará signal_BW_TKEO2 de dimensiones 1x40.

Del vector signal_BW_TKEO2 se toman los valores ubicados en las posiciones 6, 7, 8 ,9 y 10 y se los suma, el resultado se divide para la suma de los valores ubicados en las posiciones 1, 2, 3, 4 y 5, del mismo vector signal_BW_TKEO2. El código se encuentra en las líneas 309 a 319 del anexo C.

$$VFER_{NSR\_TKEO} \approx \frac{\sum\{mean\left[ln(mean\left(TKEO(sigVBW)\right))\right]\}_6^{10}}{\sum\{mean\left[ln(mean\left(TKEO(sigVBW)\right))\right]\}_1^5} \tag{2.45}$$

- ✓ VFER-NSR_SEO

El vector sig_SEO después de todas las iteraciones se convierte en una matriz, de la cual se calcula el logaritmo natural y el promedio, al vector resultante se lo llama VFlog_SEO de dimensiones 1x40.

Finalmente, del vector VFlog_SEO se toman los valores ubicados en las posiciones 6, 7, 8 ,9 y 10 y se los suma, el resultado se divide para la suma de los valores ubicados en las posiciones 1, 2, 3, 4 y 5, del mismo vector VFlog_SEO. El código se encuentra en las líneas 307 a 320 del anexo C.

$$VFER_{NSR\_SEO} \approx \frac{\sum\{mean[ln((mean(sigVBW))^2)]\}_6^{10}}{\sum\{mean[ln((mean(sigVBW))^2)]\}_1^5} \tag{2.46}$$

2.2.1.6 Empirical Mode Descomposition Excitation Ratios (EMD-ER)

Para el cálculo de esta familia de características Matlab utiliza la función:

- *emd.* - que es la descomposición de modo empírico, la cual retorna las funciones de modo intrínseco IMF, el máximo número de IMF por default es 10.

Al vector columna del audio digitalizado (data) de dimensiones Nx1, se le aplica la función *emd*, obteniendo una matriz de dimensiones Nx10, luego de sacar la transformada tendremos una matriz llamada IMF_dec de dimensiones 10xN. N indica el número de

iteraciones que se deben realizar. El código se encuentra en las líneas 323 y 324 del anexo C.

Las variantes de la característica EMF-ER son:

- ✓ IMF-SNR_SEO

Para el cálculo de la característica se partirá de la matriz IMF_dec, en la primera iteración se toma la primera columna de la matriz, se la eleva al cuadrado, se calcula el promedio y el valor absoluto, al resultado se lo llama IMF_decEnergy, para la segunda iteración se toma la segunda columna y se realiza el mismo procedimiento, hasta la iteración N. Al final de todas las iteraciones se tendrá un vector IMF_decEnergy con todos los valores resultantes de las iteraciones de dimensiones 1XN.

Del vector IMF_decEnergy se toman los valores ubicados desde la posición 4 hasta N y se los suma, el resultado se divide para la suma de los valores ubicados en las posiciones 1, 2 y 3 del mismo vector IMF_decEnergy. El código se encuentra en las líneas 328 y 335 del anexo C.

$$IMF_{NSR\_SEO} \approx \frac{\sum|mean((emd(data)')^2)|_4^{end}}{\sum|mean((emd(data)')^2)|_1^3} \tag{2.47}$$

- ✓ IMF-SNR_TKEO

Se toma la matriz IMF_dec, en la primera iteración se toma la primera columna se calcula el TKEO, el promedio y el valor absoluto. Para la segunda iteración se tomará la segunda columna y así sucesivamente hasta la iteración N. Generando el vector IMF_decTKEO al final de todas las iteraciones.

Del vector IMF_decTKEO se toman los valores ubicados desde la posición 4 hasta N, se los suma y el resultado se divide para la suma de los valores ubicados en la posición 1, 2 y 3 del mismo vector IMF_decTKEO. El código se encuentra en las líneas 329 y 336 del anexo C.

$$IMF_{NSR\_SEO} \approx \frac{\sum|mean(TKEO(emd(data)'))|_4^{end}}{\sum|mean(TKEO(emd(data)'))|_1^3} \tag{2.48}$$

- ✓ IMF-SNR_entropy

Igual que en las anteriores características se empieza analizando la primera columna de la matriz IMF_dec de la cual se saca el logaritmo natural y se multiplica por la columna original, después se suman los valores del vector resultante y se encuentra el valor

absoluto. Para la segunda iteración se toma la segunda columna y así sucesivamente, hasta generar el vector final IMF_decEntropia.

Del vector IMF_decEntropia se suman los valores ubicados desde la posición 4 hasta N y el resultado se divide para la suma de los valores ubicados en las posiciones 1, 2 y 3 del mismo vector IMF_decEntropia. El código se encuentra en las líneas 330 y 337 del anexo C.

$$IMF_{SNR\_entropy} \approx \frac{\sum|mean(-sum((emd(data)') * ln(emd(data)')))|_4^{end}}{\sum\left|mean(-sum\left((emd(data)') * ln(emd(data)')\right)\right)\right|_1^3} \tag{2.49}$$

- ✓ IMF-NSR_SEO

Se calcula el logaritmo natural de la matriz IMF_dec, obteniendo una nueva matriz con valores complejos que se llamará IMF_dec2, la cual se analiza columna por columna. Entonces, se toma la primera columna se eleva al cuadrado, se calcula el promedio y se saca el valor absoluto, luego se toma la segunda columna y se realiza el mismo procedimiento, al final de analizar todas las columnas se tendrá un vector resultado IMF_decEnergy2.

Del vector IMF_decEnergy2 se toman los valores ubicados en las posiciones 1 y 2 y se los suma, el resultado se divide para los valores ubicados desde la posición 4 hasta N, para finalmente sacar el valor absoluto. El código se encuentra en las líneas 331 y 338 del anexo C.

$$IMF_{NSR\_SEO} \approx \left|\frac{\sum|mean((ln(emd(data)'))^2)|_1^2}{\sum|mean((ln(emd(data)'))^2)|_4^{end}}\right| \tag{2.50}$$

- ✓ IMF-NSR_TKEO

De la primera columna de la matriz IMF_dec2 se calcula el TKEO, el promedio y el valor absoluto. Luego de analizar todas las columnas de la matriz se obtiene el vector IMF_decTKEO2.

Se toman los valores de las posiciones 1 y 2 del vector IMF_decTKEO2, se los suma y el resultado se divide para la suma de los valores ubicados desde la posición 3 hasta N, finalmente se calcula el valor absoluto del resultado de la división. El código se encuentra en las líneas 332 y 339 del anexo C.

$$IMF_{NSR\_TKEO} \approx \left|\frac{\sum|mean(TKEO(ln(emd(data)')))|_1^2}{\sum|mean(TKEO(ln(emd(data)')))|_4^{end}}\right| \tag{2.51}$$

- ✓ IMF-NSR_entropy

Se toma la primera columna de la matriz IMF_dec2, se calcula el logaritmo natural, luego se multiplica por la misma columna. Obteniendo un vector resultante del cual se suman todos los valores que lo componen, de la respuesta obtenida de la suma se calcula el promedio y el valor absoluto. Al finalizar todas las iteraciones se tendrá un vector IMF_decEntropia2.

Se toman los valores de las posiciones 1 y 2 del vector IMF_decEntropia2 y se los suma, el resultado se divide para la suma de los valores ubicados desde la posición 3 hasta N del mismo vector IMF_decEntropia2. El código se encuentra en las líneas 333 y 340 del anexo C.

$$IMF_{NSR\_entropy} \approx \frac{\sum\left|mean\left(-\sum\left(\left(ln(emd(data)')\right) * ln(ln(emd(data)')\right)\right)\right|_1^2}{\sum\left|mean\left(-\sum\left(\left(ln(emd(data)')\right) * ln(ln(emd(data)')\right)\right)\right|_3^{end}} \quad (2.52)$$

2.2.1.7 Pitch Period Entropy (PPE)

Matlab utiliza algunas funciones para el cálculo de esta característica como:

- *Arcov.* - calcula coeficientes de la señal de entrada utilizando el método de covarianza.
- *Hist.*- retorna un vector fila, con el número de elementos que hay en cada grupo.

En el procedimiento realizado se utiliza el vector columna F0 el cual se divide para la frecuencia fundamental promedio de personas saludables que es 120, del resultado se calcula el logaritmo natural y se lo llama logF0signal.

Después se estiman 10 coeficientes de la señal logF0signal utilizando el método de covarianza, los cuales serán los coeficientes del numerador del filtro aplicado a la misma señal logF0signal.

Los valores del vector logF0signal se los ordena de menor a mayor y se los agrupa con el propósito de que se obtengan 100 grupos, utilizando la función *hist* se obtendrá un vector PPEd en donde se indica el número de valores en cada uno de los 100 grupos.

Finalmente se divide el vector PPEd para la suma de todos los valores del mismo vector, al total o resultado se lo llama p y de este se calcula la entropía, que es p por el logaritmo natural de p, el cálculo de la entropía será dividido para el logaritmo natural de 100. El código se encuentra en las líneas 70 a 77 del anexo C.

$$PPE \equiv \frac{-\sum_{i}^{100} p(i) * \ln(p(i))}{\ln(100)} \tag{2.53}$$

2.2.2 SCRIPT *melcepst.m*

El script *melcepst.m* que se encuentra en el *toolbox* desarrollado por Athanasios Tsanas se adjunta en el anexo D.

2.2.2.1 Mel Frequency Cepstral Coefficients (MFCC)

Para el cálculo de este grupo de características Matlab utiliza las siguientes funciones:

- *Enframe.* - divide la señal en tramas superpuestas
- *Rfft.* - calcula la transformada rápida de Fourier de valores reales
- *Melbakm.* - encuentra una matriz dispersa que contiene las amplitudes del banco de filtros
- *Conj.* - calcula la conjugada de un valor
- *Sqrt.* - calcula la raíz cuadrada
- *Rdct.* - encuentra la transformada discreta del coseno

Las variantes de la característica MFCC son:

- ✓ mean_Log energy

Se tiene un vector con el audio digitalizado (*data)* de dimensiones Nx1, a este vector se le aplica la función *enframe* para enventanar la señal, tomando ventanas de 128 muestras con solapamiento del 50%, entonces la primera trama va desde 1 hasta 128, la segunda desde 64 hasta 192, y así sucesivamente hasta que se terminen todos los valores del vector *data.* Obteniendo al final una matriz de dimensiones ((N/64)-1)x128, calculando la traspuesta de la matriz quedaría de dimensiones 128x((N/64)-1).

Luego se calcula la transformada rápida de Fourier de la matriz con la función *rfft*, lo que genera una nueva matriz de dimensiones 65x((N/64)-1), a la que se llamará f.

A continuación, se utiliza la función *melbankm* para calcular la matriz dispersa m la cual contiene las amplitudes de 27 filtros triangulares espaciados de acuerdo con la escala de frecuencias mel, además de dos límites que serán a=2 y b=64.

Ahora de la matriz f se toma desde la fila 2 hasta la fila 64 y se multiplica por la conjugada de la misma matriz, resultado al que se llamará pw de dimensiones 63x((N/64)-1). Se saca el máximo valor de pw y se multiplica por $1x10^{-20}$, a este valor se lo llamará pth.

Finalmente, para calcular la característica se suma las filas de cada columna de la matriz pw, obteniendo un vector fila de dimensiones 1x((N/64)-1). Del vector obtenido se buscan los valores menores a pth y se los multiplica por pth, para después calcular el logaritmo natural y hacer la traspuesta del vector fila generando un vector columna log_energy, del cual por último se calcula el promedio. El código se encuentra en las filas 24 a 29 y 49 del anexo D.

$$mean_{Log\ energy} \approx mean(\ln\ ((rfft(data)) * conj(rfft(data))) \tag{2.54}$$

- ✓ std_Log energy

Del vector log_energy se calcula la desviación estándar. El código se encuentra en las filas 24 a 29 y 49 del anexo D.

$$std_{Log\ energy} \approx std(\ln\ ((rfft(data)) * conj(rfft(data))) \tag{2.55}$$

- ✓ Desde mean_MFCC_0th coef hasta mean_MFCC_12th coef

Se calcula la raíz cuadrada de pth y se obtiene un valor que se llamará ath, el cual se utilizará más adelante.

De la matriz f calculada previamente, se toman los valores ubicados desde la fila 2 hasta la fila 64, se saca el valor absoluto y el resultado se multiplica por la matriz de dispersión m. De la nueva matriz resultante se buscan los valores menores que ath y de existir alguno se multiplica por ath. Luego se calcula el logaritmo natural y la transformada discreta de coseno. Generando una matriz de dimensiones 26x((N/64)-1), de la cual se sacará la traspuesta y se la llamará matriz c.

De la matriz c, se seleccionan los valores desde la columna 1 hasta la columna 13. Generando una matriz de dimensiones ((N/64)-1) x13 que se sigue llamando c, después se saca el promedio de cada columna y se obtiene un vector de dimensiones 1x13, los cuales serán los 13 coeficientes desde el coeficiente 0 hasta el coeficiente 12 respectivamente. El código se encuentra en las líneas 33 a 49 del anexo D.

$$MFCC\_n\ coef \approx mean\left(\sum_{k=1}^{K}\left(\ln\left(m * \left(rfft(data)\right)\right)\right)\cos\left[\frac{n(k-0.5)\pi}{K}\right]\right) \tag{2.56}$$

$$K = 27\ filtros\ y\ n = 0, 1, 2, \ldots\ldots, 12\ coeficientes$$

Ahora, en una misma matriz se ubica el vector columna log_energy y junto a él la matriz c. Generando una matriz de dimensiones ((N/64)-1) x14 llamada MFCC.

- ✓ Desde std_MFCC_0th coef hasta std_MFCC_12th

Se calcula la desviación estándar de cada columna de la matriz c. El código se encuentra en las líneas 33 a 49 del anexo D.

$$std\_MFCC\_n\ coef \approx std\left(\sum_{k=1}^{K}\left(\ln\left(m * \left(rfft(data)\right)\right)\right)\cos\left[\frac{n(k-0.5)\pi}{K}\right]\right) \tag{2.57}$$

$$K = 27\ filtros\ y\ n = 0, 1, 2, \ldots\ldots, 12\ coeficientes$$

- ✓ mean_delta log energy, mean_0th delta - mean_12th delta

Se generan 5 filas con los mismos valores de la primera fila de la matriz MFCC, debajo de la cual se ubica la matriz MFCC y debajo de ésta, se ubican 5 filas con los mismos valores de la última fila de la matriz MFCC. Quedando una matriz de dimensiones ((N/64)-1+10) x14, luego los valores de las 14 columnas se ubican debajo de la primera columna, obteniendo un vector columna que será filtrado utilizando una función de transferencia donde los coeficientes del numerador irán desde 4/60 disminuyendo en 1/60 hasta -4/60. Teniendo el vector columna ya filtrado se utiliza la función *reshape* que permite reorganizar el vector columna en una matriz (vx), de las mismas dimensiones que tenía antes de convertirse en un vector ((N/64)-1+10) x14. Es necesario que el número de filas de la matriz vx sea igual al número de filas de la matriz MFCC, por lo tanto, de la matriz vx se eliminan desde la fila 1 hasta la fila 8 y luego se elimina la primera fila y la última fila.

Finalmente obteniendo una matriz de las mismas dimensiones que MFCC ((N/64)-1) x14, se calcula el promedio de cada columna, y se obtiene un vector de 1x14 que son los valores de las características de mean_delta log energy y mean_0th delta hasta mean_12th delta, respectivamente. El código se encuentra en las líneas 49 a 60 del anexo D.

$$mean\ n_{delta} \approx mean\left(filter\left(\left(\sum_{k=1}^{K}\left(\ln\left(m * \left(rfft(data)\right)\right)\right)\cos\left[\frac{n(k-0.5)\pi}{K}\right]\right)\right)\right) \tag{2.58}$$

$$K = 27\ filtros\ y\ n = 0, 1, 2, \ldots\ldots, 12\ coeficientes$$

- ✓ std_delta log energy, std_0th delta - std_12th delta

De la matriz vx de las mismas dimensiones que MFCC ((N/64)-1) x14, se calcula la desviación estándar de cada columna. Generando un vector de 1x14 con los valores de las características. El código se encuentra en las líneas 49 a 60 del anexo D.

$$std\ n_{delta} \approx std\left(filter\left(\sum_{k=1}^{K}\left(\ln\left(m * \left(rfft(data)\right)\right)\right)\cos\left[\frac{n(k-0.5)\pi}{K}\right]\right)\right) \tag{2.59}$$

$$K = 27\ filtros\ y\ n = 0, 1, 2, \ldots\ldots, 12\ coeficientes$$

- ✓ mean_delta delta log energy, mean_delta delta 0th hasta mean_12th delta delta

De la matriz vx una vez eliminadas desde la fila 1 hasta la fila 8, se toman las columnas y se las ubica debajo de la primera columna, generando un vector columna que será filtrado utilizando tres coeficientes para el numerador, que irán desde 1/2 disminuyendo en 1/2 hasta -1/2. Luego se utiliza la función *reshape* para reorganizar el vector columna en una matriz ax de dimensiones ((N/64)-1+2) x14, de la cual se elimina la fila 1 y 2, para finalmente obtener una matriz de las mismas dimensiones que MFCC ((N/64)-1) x14, de la cual se calcula el promedio de cada columna obteniendo un vector de 1x14, con los valores de las características. El código se encuentra en las líneas 57 a 62 del anexo D.

$$mean\ delta - delta\ n \approx mean\left(filter\left(filter\left(\sum_{k=1}^{K}\left(\ln\left(m * \left(rfft(data)\right)\right)\right)\cos\left[\frac{n(k-0.5)\pi}{K}\right]\right)\right)\right) \tag{2.60}$$

$$K = 27\ filtros\ y\ n = 0, 1, 2, \ldots\ldots, 12\ coeficientes$$

- ✓ std_delta delta log energy, std_delta delta 0th hasta std_12th delta delta

De la matriz ax con las mismas dimensiones de MFCC ((N/64)-1) x14, se calcula la desviación estándar de cada columna. Obteniendo un vector 1x14 con los valores de las características. El código se encuentra en las líneas 57 a 62 del anexo D.

$$std\ delta - delta\ n \approx std\left(filter\left(filter\left(\sum_{k=1}^{K}\left(\ln\left(m * \left(rfft(data)\right)\right)\right)\cos\left[\frac{n(k-0.5)\pi}{K}\right]\right)\right)\right) \tag{2.61}$$

$$K = 27\ filtros\ y\ n = 0, 1, 2, \ldots\ldots, 12\ coeficientes$$

2.2.3 SCRIPT *wavedec_features.m*

El código del script *wavedec_features.m* ubicado en el *toolbox* desarrollado por Athanasios Tsanas se adjunta en anexo E.

2.2.3.1 Wavelet Measures

Para el cálculo de este grupo de características en Matlab se utilizan las siguientes funciones:

- *Wavedec.* - calcula la descomposición de la señal para 10 niveles utilizando una wavelet 'db8'.
- *Wenergy.* - retorna el porcentaje de energía correspondiente a aproximación Ea y el vector Ed que contiene los porcentajes de energía correspondiente a detalles.
- *Detcoef.* - extrae los coeficientes de detalle.
- *Wentropy.* - retorna la entropía de un vector o matriz.
- *Appcoef.* - extrae los coeficientes de aproximación.

Las variantes de la característica wavelet son:

✓ Ea

Para el cálculo de la característica, se aplica la función *wavedec* al vector columna F0, obteniendo un vector C con la descomposición wavelet y un vector L con el número de coeficientes por nivel.

Del vector C se toman los valores ubicados desde la posición 1 hasta el valor que se indique en la primera posición del vector L, después el segmento obtenido se eleva al cuadrado y se suman los valores contenidos en él, el resultado de la suma se divide para la suma de los valores del vector C elevados al cuadrado. Finalmente, el resultado se multiplicará por 100. El código se encuentra en las líneas 8 y 9 del anexo E.

$$E_a = 100 * \frac{\sum(((C)_1^{L(1)})^2)}{\sum(C^2)} \tag{2.62}$$

✓ Ed_1_coef - Ed_10_coef

Utilizando la función *detcoef* se obtiene un vector fila con 10 celdas (d), se analiza cada celda, en la primera celda hay un vector columna el cual se eleva al cuadrado, luego se suman los valores del vector y el resultado se divide para la suma del vector C elevado al cuadrado.

Finalmente, el resultado de la división se multiplica por 100, así se consigue calcular Ed_1_coef, se repite el mismo procedimiento para cada celda, hasta calcular Ed_10_coef. El código se encuentra en la línea 11 del anexo E.

$$Ed_n coef = 100 * \frac{\sum\left(\left(Cd(n)\right)^2\right)}{\sum(C^2)} \quad (2.63)$$
$$n = 1,2, \dots 10\ coeficientes$$

- ✓ det_entropy_shannon_1_coef - det_entropy_shannon_10_coef

Para determinar la primera característica det_entropy_shannon_1_coef, se toma el vector columna que se encuentra en la primera celda del vector d y se lo multiplica por el logaritmo natural del mismo vector columna, pero elevado al cuadrado. Del vector resultante se suman todos sus valores, y al total se le antepone el signo menos. Para el cálculo de la siguiente característica se analizará la segunda celda, y así sucesivamente. El código se encuentra en la línea 12 del anexo E.

$$det\ entropy\ shannon_n\ coef = -\sum\left(Cd(n) * \ln\left(\left(Cd(n)\right)^2\right)\right) \quad (2.64)$$
$$n = 1,2, \dots\dots 10\ coeficientes$$

- ✓ det_entropy_log_1_coef - det_entropy_log_10_coef

Se tienen 10 celdas obtenidas con la función *detcoef,* para el cálculo de det_entropy_log_1_coef se toma el vector columna de la primera celda, se lo eleva al cuadrado, luego se calcula el logaritmo natural y finalmente se suman todos los valores del vector resultante. Se realiza el mismo procedimiento con los vectores que se encuentran en las 9 celdas restantes. El código se encuentra en la línea 13 del anexo D.

$$det\ entropy\ log_n\ coef = \sum \ln\left(\left(Cd(n)\right)^2\right) \quad (2.65)$$
$$n = 1,2, \dots 10\ coeficientes$$

- ✓ det_TKEO_mean_1_coef - det_TKEO_mean_10_coef

Se tiene el vector d con 10 celdas, se calcula el TKEO de la primera celda y luego se obtiene el promedio. Para el cálculo del siguiente coeficiente se toma el vector de la segunda celda y se realiza el mismo procedimiento y así sucesivamente hasta calcular det_TKEO_mean_10_coef. El código se encuentra en la línea 14 del anexo D.

$$det\ TKEO\ mean_n\ coef = mean\left(TKEO\left(Cd(n)\right)\right) \quad (2.66)$$

$$n = 1,2, \ldots..10\ coeficientes$$

- ✓ det_TKEO_std_1_coef - det_TKEO_std_10_coef

Se toma el vector columna ubicado en la primera celda del vector d, se calcula el TKEO y luego la desviación estándar. Se realiza el mismo procedimiento para todos los 10 vectores que se encuentran en las celdas del vector d. El código se encuentra en la línea 15 del anexo E.

$$det\ TKEO\ std_n\ coef = std\left(TKEO\big(Cd(n)\big)\right) \tag{2.67}$$
$$n = 1,2, \ldots..10\ coeficientes$$

- ✓ app_entropy_shannon_1_coef - app_entropy_shannon_10_coef

Utilizando el vector de descomposición wavelet C y el vector con el número de coeficientes por nivel L, se calculan los coeficientes de aproximación por medio de la función *appcoef* para el primer nivel, obteniendo un vector columna Ca que se multiplica por el logaritmo natural del mismo vector Ca elevado al cuadrado, del vector resultante se suman los valores y a la respuesta obtenida de la suma se le antepone el signo menos.

Se realiza el mismo procedimiento para los 9 niveles restantes. El código se encuentra en las líneas 18 y 19 del anexo E.

$$app\ entropy\ shannon_n\ coef = -\sum\left(Ca(n) * \ln\left(\left(Ca(n)\right)^2\right)\right) \tag{2.68}$$
$$n = 1,2, \ldots..10\ coeficientes$$

- ✓ app_entropy_log_1_coef - app_entropy_log_10_coef

Se toma el vector Ca con los coeficientes de aproximación del primer nivel, se lo eleva al cuadrado, luego se calcula el logaritmo natural y finalmente se suman los valores del vector resultante. Después se calcula el vector Ca con los coeficientes de aproximación del segundo nivel y realiza el mismo procedimiento, y así sucesivamente para los 10 niveles. El código se encuentra en las líneas 18 y 20 del anexo E.

$$app\ entropy\ log_n\ coef = \sum \ln\left(\left(Ca(n)\right)^2\right) \tag{2.69}$$
$$n = 1,2, \ldots..10\ coeficientes$$

- ✓ app_det_TKEO_mean_1_coef - app_det_TKEO_mean_10_coef

Se calcula el TKEO del vector Ca del primer nivel y se encuentra el promedio. Se realiza lo mismo con los vectores Ca de los 9 niveles restantes. El código se encuentra en las líneas 18 y 21 del anexo E.

$$app\ det\ TKEO\ mean_n\ coef = mean\left(TKEO\big(Ca(n)\big)\right) \quad (2.70)$$
$$n = 1,2, \ldots\ldots 10\ coeficientes$$

- ✓ app_det_TKEO_std_1_coef - app_det_TKEO_std_10_coef

Se calcula la desviación estándar, del resultado obtenido de encontrar el TKEO del vector Ca del primer nivel, se realiza lo mismo para los 9 niveles restantes. El código se encuentra en las líneas 18 y 22 del anexo E.

$$app\ det\ TKEO\ std_n\ coef = std\left(TKEO\big(Ca(n)\big)\right) \quad (2.71)$$
$$n = 1,2, \ldots\ldots 10\ coeficientes$$

- ✓ Ea2

Para el cálculo de estas características, se obtiene el logaritmo natural del vector F0, el cual será el argumento de entrada de la función *wavedec,* para el cálculo del vector con la descomposición wavelet C2 y el vector con el número de coeficientes por nivel L2.

$$E_a 2 = 100 * \frac{\sum(((C2)_1^{L2(1)})^2)}{\sum(C2^2)} \quad (2.72)$$

Utilizando los nuevos vectores C2 y L2 y realizando el mismo procedimiento con el cual se obtuvieron las características anteriores, se calculan las siguientes características:

El código para determinar estas características se encuentra en las líneas 29 a 40 del anexo E.

- ✓ Ed2_1_coef - Ed2_10_coef

$$Ed2_n coef = 100 * \frac{\sum\left(\left(Cd2(n)\right)^2\right)}{\sum(C2^2)} \quad (2.73)$$
$$n = 1,2, \ldots 10\ coeficientes$$

- ✓ det_LT_entropy_shannon_1_coef - det_LT_entropy_shannon_10_coef

$$det\ LT\ entropy\ shannon_n\ coef = -\sum \left(Cd2(n) * \ln\left(\left(Cd2(n) \right)^2 \right) \right) \tag{2.74}$$
$$n = 1,2, \dots 10\ coeficientes$$

- ✓ det_LT_entropy_log_1_coef - det_LT_entropy_log_10_coef

$$det\ LT\ entropy\ log_n\ coef = \sum \ln\left(\left(Cd2(n) \right)^2 \right) \tag{2.75}$$
$$n = 1,2, \dots 10\ coeficientes$$

- ✓ det_LT_TKEO_mean_1_coef - det_LT_TKEO_mean_10_coef

$$det\ LT\ TKEO\ mean_n\ coef = mean\left(TKEO\left(Cd2(n) \right) \right) \tag{2.76}$$
$$n = 1,2, \dots 10\ coeficientes$$

- ✓ det_LT_TKEO_std_1_coef - det_LT_TKEO_std_10_coef

$$det\ LT\ TKEO\ std_n\ coef = std\left(TKEO\left(Cd2(n) \right) \right) \tag{2.77}$$
$$n = 1,2, \dots 10\ coeficientes$$

- ✓ app_LT_entropy_shannon_1_coef - app_LT_entropy_shannon_1_coef

$$app\ LT\ entropy\ shannon_n\ coef = -\sum \left(Ca2(n) * \ln\left(\left(Ca2(n) \right)^2 \right) \right) \tag{2.78}$$
$$n = 1,2, \dots 10\ coeficientes$$

- ✓ app_LT_entropy_log_1_coef - app_LT_entropy_log_10_coef

$$app\ LT\ entropy\ log_n\ coef = \sum \ln\left(\left(Ca2(n) \right)^2 \right) \tag{2.79}$$
$$n = 1,2, \dots 10\ coeficientes$$

- ✓ app_LT_TKEO_mean_1_coef - app_LT_TKEO_mean_10_coef

$$app\ LT\ TKEO\ mean_n\ coef = mean\left(TKEO\left(Ca2(n) \right) \right) \tag{2.80}$$
$$n = 1,2, \dots 10\ coeficientes$$

- ✓ app_LT_TKEO_std_1_coef - app_LT_TKEO_std_10_coef

$$app\ LT\ TKEO\ std_n\ coef = std\left(TKEO\left(Ca2(n)\right)\right) \tag{2.81}$$
$$n = 1,2, \dots 10\ coeficientes$$

2.2.4 SCRIPT *fastfda.m*

El código del script *fastfda.m* ubicado en *toolbox* desarrollado por Tsanas se encuentra en el anexo F.

2.2.4.1 Dentrended Fluctuation Analysis (DFA)

Inicialmente se integran las muestras de señal x: $S_n = \sum_{i=1}^{n}(x_i)$, donde n = 1, 2, …N, siendo N la longitud de la señal. El resultado obtenido se divide en intervalos o ventanas de longitud L no superpuestos y en cada ventana se realiza un ajuste de mínimos cuadrados, obteniendo la pendiente a y la intersección b.

Después de realizar el mismo procedimiento en todas las ventanas se obtiene la fluctuación del valor cuadrático medio de las tendencias locales de todas las ventanas $F(L)$.

$$F(L) = \left[\frac{1}{L}\sum_{n=1}^{L}(S_n - a_n - b)^2\right]^{1/2}$$

El exponente de escala *dfa* se determina encontrando el logaritmo en base 10 de L versus el logaritmo en base 10 de F(L) creando una línea recta que luego se lo normaliza entre 0 y 1.

Finalmente se divide uno para la suma entre 1 y el resultado de elevar e (número de Euler = 2.7183) al exponente de escala negativo [22]. El código se encuentra en las líneas 2 a 7 del anexo F.

$$DFA \equiv \frac{1}{1 + e^{-dfa}} \tag{2.62}$$

2.2.5 SCRIPT *rpde.m*

El script *rpde.m* ubicado en el *toolbox* desarrollado por Tsanas se encuentra en el anexo G.

2.2.5.1 Recurrence Period Density Entropy (RPDE)

Primero se remuestrea la señal de entrada a una frecuencia de 25k, luego se toman dos vectores de dimensión 4x1. Se parte de la muestra n_0 y se avanza en el tiempo, hasta identificar la muestra k a la cual la distancia euclidiana entre los dos vector $x_{n0} = [x_{n0}, \dots, x_{n0+m}]$ y $x_{n0+k} = [x_{n0+k}, \dots, x_{n0+m}]$ sea $\|x_{n0} - x_{n0+k}\| > r$, r = 0.2.

Después se determina la muestra n_1, para poder encontrar el tiempo de recurrencia $T = n_1 - n_2$. Se repite el procedimiento para toda la señal con el fin de formar un histograma de los tiempos de recurrencia $R(T)$, el cual es normalizado para obtener la densidad de probabilidad de tiempo de recurrencia $p(T)$.

$$p(T) = \frac{R(T)}{\sum_{i=1}^{Tmax} R(i)}$$

Donde $T_{max} = 1000$ es el tiempo de recurrencia máximo.

Finalmente, para determinar RPDE se parte de la entropía de la distribución de $p(T)$, normalizada por la entropía de una señal puramente estocástica $\ln(T_{max})$ [22]. El código se encuentra en las líneas 12 a 20 del anexo G.

$$RPDE \equiv \frac{-\sum_{i}^{Tmax} p(i) * \ln(p(i))}{\ln(T_{max})} \tag{2.63}$$

2.3 DETERMINACIÓN DEL TAMAÑO MÍNIMO DE LA MUESTRA PARA LA EXTRACCIÓN DE CARACTERÍSTICAS CONFIABLE

El procedimiento realizado para determinar el tamaño que debe tener la muestra de audio para la extracción de características confiable se muestra en la figura 2.1. El propósito de este procedimiento es analizar cómo están organizadas las características dependiendo del tamaño del audio. La única observación que se tomó en cuenta es que el *toolbox* es confiable únicamente con la fonación de la vocal /a/ [27].

A continuación, se detallan los pasos indicados en la figura 2.1:

2.3.1 VERIFICACIÓN DE LA FONACIÓN DE LA VOCAL /a/

Inicialmente se tienen 779 grabaciones de voz de diferentes tamaños obtenidas de la base de datos PVA-dryrun, cada una de ellas será analizada para verificar que durante toda su longitud se tenga la fonación de la vocal /a/, en caso de que existan partes diferentes a la vocal /a/, se utiliza la herramienta Audacity[3] para eliminarlas.

A continuación, en la tabla 2.2 se indica el segundo desde el cual han sido recortados algunos audios que tenían tos, ruido, suspiros, sonidos que no cumple con lo requerido y que son diferentes a la fonación de la vocal /a/.

[3] Audacity es un software diseñado para editar audio.

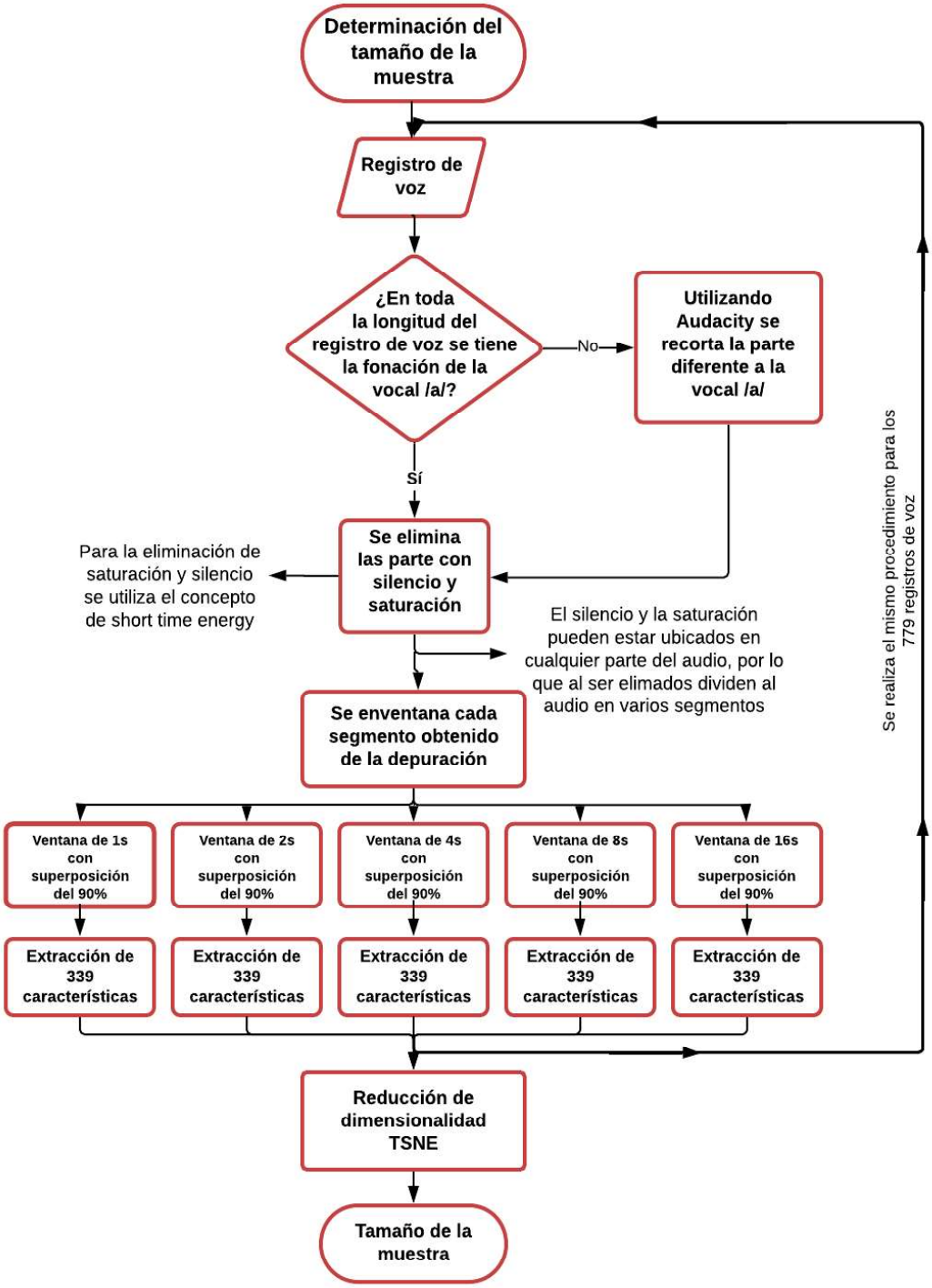

Figura 2.1. Diagrama de flujo de la determinación del tamaño de las muestras de voz

Tabla 2.2. Audios recortados utilizando Audacity

ID AUDIO	RECORTE
1.- pva_0235147_2014-02-22-195622.wav	Desde el segundo 27
2.- pva_0402368_2014-02-20-042408.wav	Desde el segundo 19
3.- pva_0503938_2014-02-19-232751.wav	Desde el segundo 22
4.- pva_0579356_2014-02-21-143347.wav	Desde el segundo 4
5.- pva_0837443_2014-03-08-114228.wav	Desde el segundo 18
6.- pva_1796096_2014-02-18-112211.wav	Desde el segundo 16.5
7.- pva_1800540_2014-02-07-060600.wav	Desde el segundo 10
8.- pva_2083540_2014-02-08-041734.wav	Desde el segundo 18
9.- pva_2671030_2014-02-20-015229.wav	Desde el segundo 18
10.- pva_3300712_2014-01-28-162411.wav	Desde el segundo 2.6
11.- pva_3444744_2014-03-05-040213.wav	Desde el segundo 18
12.- pva_4112624_2014-02-20-145142.wav	Desde el segundo 0 al 11
13.- pva_4219686_2014-02-20-015050.wav	Desde el segundo 13.5
14.- pva_4571342_2014-01-25-054932.wav	Desde el segundo 8
15.- pva_4703856_2014-02-15-192124.wav	Desde el segundo 9
16.- pva_4720430_2014-02-22-200328.wav	Desde el segundo 5
17.- pva_4866930_2014-02-15-044919.wav	Desde el segundo 23
18.- pva_4999235_2014-02-20-052124.wav	Desde el segundo 0 al 1
19.- pva_5492666_2014-02-21-210050.wav	Desde el segundo 17.5
20.- pva_5550037_2014-02-12-135310.wav	Desde el segundo 0 al 13.5
21.- pva_6762672_2014-02-20-121243.wav	Desde el segundo 15
22.- pva_7087090_2014-02-15-015104.wav	Desde el segundo 24
23.- pva_7178510_2014-01-29-205021.wav	Desde el segundo 19
24.- pva_8277921_2014-02-12-000944.wav	Desde el segundo 19
25.- pva_8465347_2014-02-20-145304.wav	Desde el segundo 4.5
26.- pva_8759528_2014-01-29-223143.wav	Desde el segundo 13
27.- pva_9433449_2014-03-07-085031.wav	Desde el segundo 4
28.- pva_9766239_2014-02-11-215751.wav	Desde el segundo 11
29.- pva_0864268_2014-01-21-020914.wav	Desde el segundo 12
30.- pva_2599463_2014-01-12-031216.wav	Desde el segundo 18
31.- pva_3495875_2014-01-08-212419.wav	Desde el segundo 8.5
32.- pva_3783094_2014-01-12-224357.wav	Desde el segundo 13
33.- pva_4264927_2014-01-18-153142.wav	Desde el segundo 15
34.- pva_4343318_2014-01-10-025106.wav	Desde el segundo 0 al 1.5
35.- pva_4434369_2014-01-22-002149.wav	Desde el segundo 15.3
36.- pva_4652942_2014-01-09-212922.wav	Desde el segundo 17

37.- pva_5182169_2014-01-11-164009.wav	Desde el segundo 6
38.- pva_5313454_2014-01-22-021408.wav	Desde el segundo 23
39.- pva_6460163_2014-01-20-022408.wav	Desde el segundo 17
40.- pva_7001403_2014-01-14-170356.wav	Desde el segundo 19
41.- pva_7147499_2014-01-14-112507.wav	Desde el segundo 12
42.- pva_7427219_2014-01-07-231044.wav	Desde el segundo 21
43.- pva_7679534_2014-01-15-214133.wav	Desde el segundo 15
44.- pva_8129427_2014-01-09-195927.wav	Desde el segundo 4.5
45.- pva_9100523_2014-01-12-055713.wav	Desde el segundo 14
46.- pva_9446433_2014-01-16-222904.wav	Desde el segundo 0 al 1
47.- pva_9699642_2014-01-14-010448.wav	Desde el segundo 9.5
48.- pva_9735253_2014-01-13-022344.wav	Desde el segundo 17

2.3.2 ELIMINACIÓN DE SATURACIÓN Y SILENCIOS

Una vez que se tiene la seguridad de que todos los audios tienen la fonación de la vocal /a/. Se eliminan las partes de silencio y saturación, para ello se desarrolló el script *depuration.m* dentro del cual lo primero que se realiza es dividir el audio en ventanas de 320 muestras con solapamiento de 160 muestras para lograr que la señal de voz sea casi estacionaria [31], y después calcular la energía de cada ventana utilizando la siguiente fórmula [32]:

$$E = \frac{1}{N}\sum_{k=1}^{N}|x[k]|^2 \quad ; N = 320$$

El proceso detallado del cálculo de la energía se indica en el script ShortTimeEnergy.m que se muestra en la figura 2.2, el cual retorna un vector con la energía de cada ventana. A este vector una vez normalizado se lo llamará Eor.

```
function E = ShortTimeEnergy(signal,win,step)
signal = signal/max(signal);                %Normalización
curPos = 1;
L = length(signal);                         %Longitud del vector con las muestras del audio
numOfFrames = floor((L-win)/step) + 1;      %Calculo del número de ventanas
E = zeros(numOfFrames,1);
for (i=1:numOfFrames)
    window = (signal(curPos:curPos+win-1)); %Selección de la ventana
    E(i) = (1/(win)) * sum(abs(window.^2)); %Cálculo de la energía
    curPos = curPos + step;
end
```

Figura 2.2. Cálculo de la energía en cada ventana

Continuando en el desarrollo del script *depuration.m*, se concluye que el vector de energía normalizada Eor deberá estar entre los umbrales de 0.05 y 0.95 (estos límites o umbrales se determinaron en base a pruebas). Si la energía es menor a 0.05, la señal se considerará como silencio y si es mayor a 0.95 se considerará como saturación.

En la figura 2.3 se muestra el ejemplo de un audio con silencio y con saturación en el cual se observan dos gráficos: en el primero se tiene la señal con la fonación de la vocal /a/, de longitud 29 segundos; y, en el segundo gráfico, se tiene la energía de la señal calculada en cada ventana Eor, además de los dos umbrales representados por las líneas negras horizontales.

El número de ventanas se determina multiplicando el tiempo del audio en segundos por la frecuencia de muestreo y dividiéndolo para el tamaño del solapamiento.

$$\#ventanas = \frac{29s * 8000Hz}{160\ muestras} = 1450\ ventanas$$

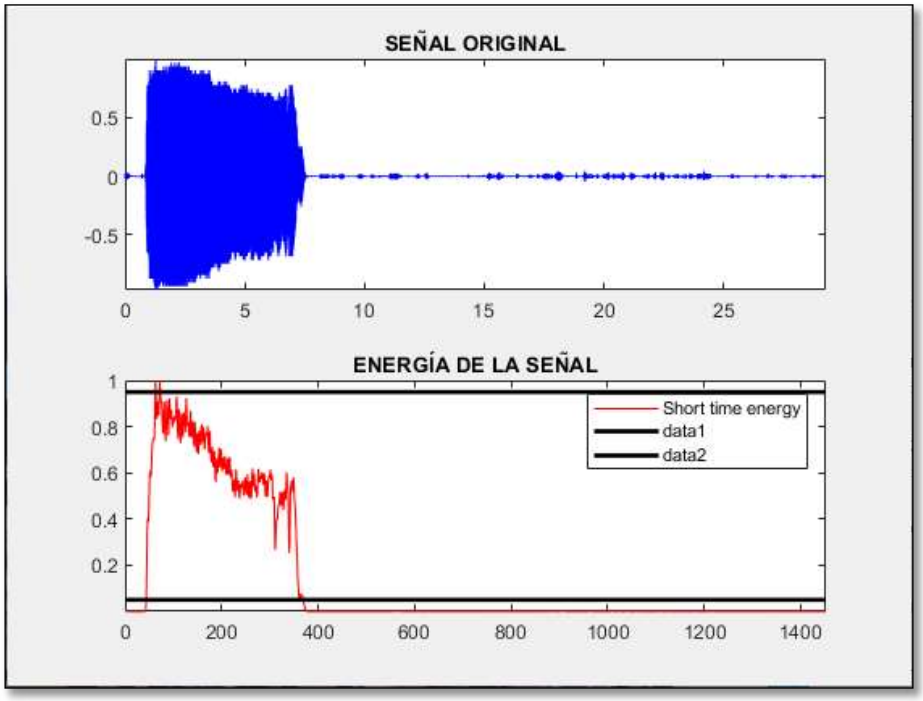

Figura 2.3. Audio pva_0478272_2014-02-16-132325.wav

En la figura 2.3 se puede observar que a partir de la muestra 400 aproximadamente, la energía es menor que el umbral de silencio, por lo cual será eliminada. Por otra parte, la energía entre la muestra 50 y 100 aproximadamente es mayor al umbral de saturación, por lo que también será eliminada.

En el gráfico “Señal depurada” de la figura 2.4 se muestra la señal depurada, en donde los segmentos en color gris son los que se eliminaron y los segmentos en color rojo son los válidos. El script *depuration.m* se encuentra en el anexo H.

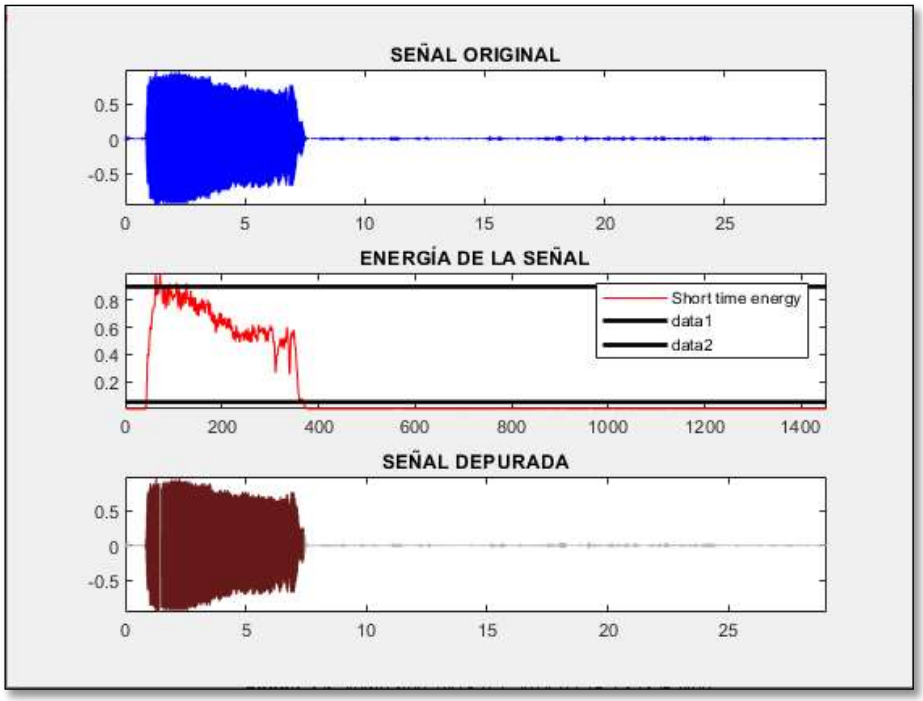

Figura 2.4. Audio pva_0478272_2014-02-16-132325.wav

2.3.3 ENVENTANAR SEGMENTOS

Después de realizar la depuración del registro de voz, se obtienen pocos o varios segmentos dependiendo del registro (segmentos de color rojo). Los cuales se van a enventanar uno por uno. Este proceso se desarrolló en el script *pruebaframe.m* que se encuentra en el anexo I.

Antes que nada, es importante saber cuál es tamaño mínimo para el cual funciona el *toolbox*, ya que éste será el primer tamaño con el que se enventanarán los segmentos. Para ello se ha tomado como ejemplo el audio pva_007588_2014-02-03-160636.wav de duración 14,25 segundos y se lo ha recortado hasta llegar a 0,908 segundos, ya que para una duración menor se genera un error que impide la obtención de las características [27].

El error obtenido, se muestra en la figura 2.5.

```
Error using aryule (line 55)
When X is a vector, it must have length greater or equal to the model order.

Error in voice_analysis>perq1 (line 341)
a = aryule(time_series-mean_tseries,p);

Error in voice_analysis>jitter_shimmer (line 273)
[Ampl_PQ3] = perq1(A0,3);

Error in voice_analysis (line 182)
[Shimmer] = jitter_shimmer(A0);
```

Figura 2.5. Error obtenido con audios de longitud menor a 0,908s

Con esto se puede concluir que el *toolbox* con el que se va a trabajar requiere que el tamaño mínimo del registro de voz sea de $0{,}908\ segundos\ \cong 1\ segundo$.

Entonces, del primer segmento se toman ventanas de 8000 muestras (1 segundo) con un solapamiento del 90% (7200 muestras), en la figura 2.6 se indica este procedimiento [33]. Después, de cada ventana se calculan las características utilizando el *toolbox.* Se realiza el mismo procedimiento con todos los segmentos que se hayan obtenido de la depuración del audio.

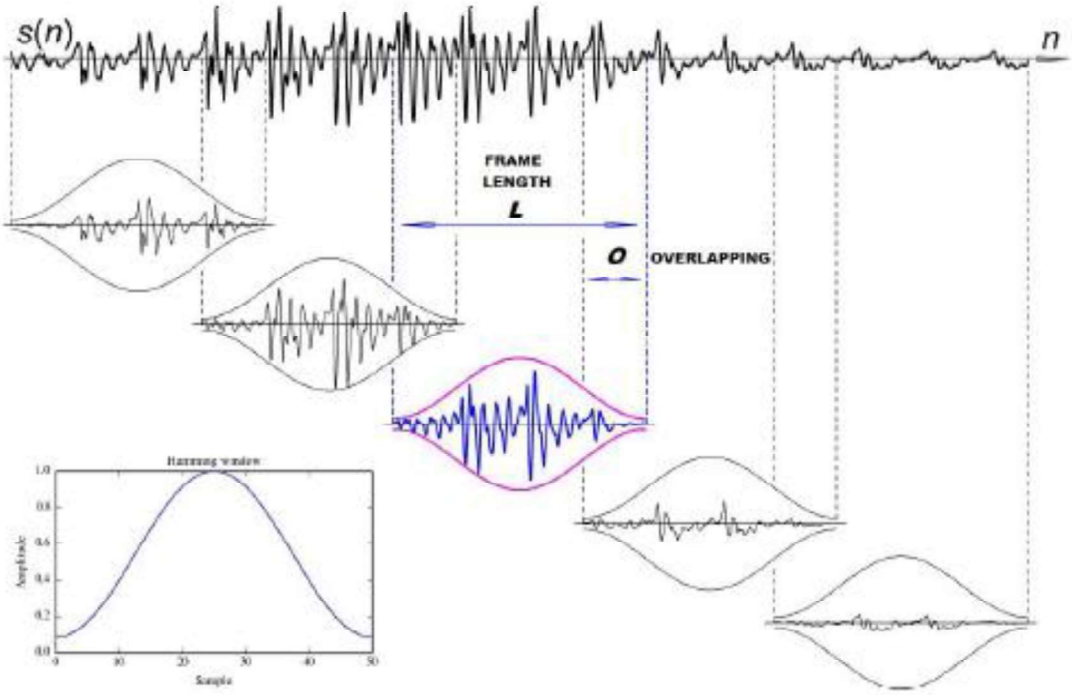

Figura 2.6. Proceso de enventanado [33]

A continuación, se sigue trabajando con el mismo segmento, y en vez de tomar ventanas de 8000 muestras (1s) se toman ventanas de 16000 muestras (2s) con el mismo solapamiento y se repite el mismo procedimiento descrito previamente, se continúa cambiando el tamaño de la ventana a 32000 muestras (4s), después a 64000 muestras (8s) y finalmente a 128000 muestras (16s). Todo dependerá del tamaño de los segmentos.

En la figura 2.7 se puede observar el procedimiento con mayor claridad, el cual se repite para los 779 registros de voz.

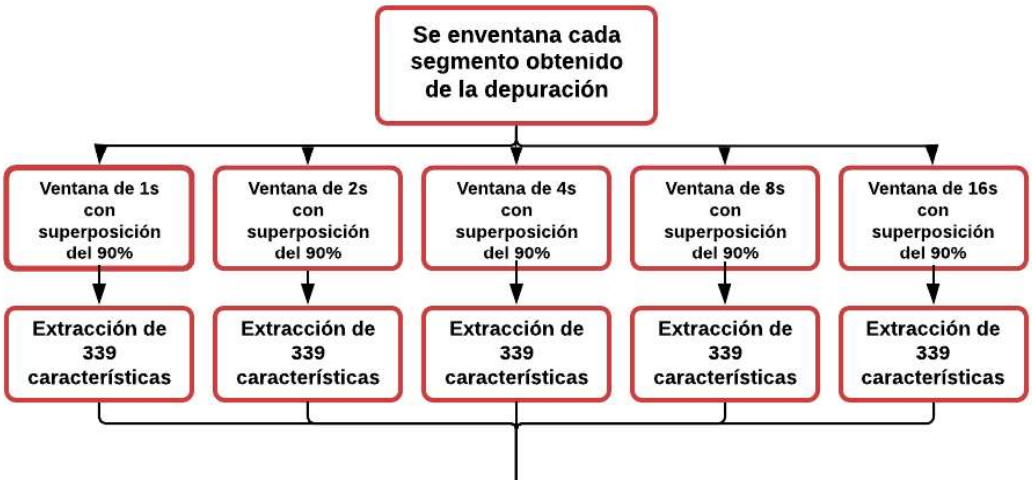

Figura 2.7. Diagrama del enventanado y extracción de características

Al final del procedimiento descrito previamente se obtiene una tabla con las 339 características de cada segmento enventanado, con el respectivo tamaño de la ventana. Así como se muestra en el esquema de tabla 2.3.

Tabla 2.3. Esquema de la tabla obtenida después del enventanado

Audio	Segmento	Tamaño de la ventana	Características
Audio1	1	1 segundo	339 características
Audio1	1	1 segundo	339 características
Audio1	1	1 segundo	339 características
Audio1	1	1 segundo	339 características
Audio1	1	1 segundo	339 características
Audio1	1	1 segundo	339 características
Audio1	1	1 segundo	339 características
Audio1	1	1 segundo	339 características

Audio1	1	1 segundo	339 características
Audio1	1	1 segundo	339 características
Audio1	1	1 segundo	339 características
Audio1	1	2 segundo	339 características
Audio1	2	1 segundo	339 características
Audio2	1	1 segundo	339 características
Audio2	1	1 segundo	339 características
Audio2	1	1 segundo	339 características
Audio3	NAN	NAN	NAN
Audio4	1	1 segundo	339 características
. . . Audio779	. . . 1	. . . 1 segundo	. . . 339 características

El procedimiento de enventanar los segmentos de audio en diferentes ventanas y extraer las características, se realiza con el propósito de obtener gran cantidad de características. Esto es necesario para poder realizar el análisis que se explica a continuación en 2.3.4.

2.3.4 REDUCCIÓN DE DIMENSIONALIDAD

Se parte de la tabla 2.3, en la cual se tienen gran cantidad de características obtenidas después del enventanado.

Para poder visualizar cómo están organizadas las características dependiendo del tamaño de la ventana, se utiliza la función tsne[4] de Matlab que permite reducir las 339 dimensiones a dos dimensiones. Para lo cual se utilizan las siguientes líneas de código:

```
TSN=tsne(zscore(fillmissing(caract,'previous')));
figure;
gscatter(TSN(:,1),TSN(:,2),time);
```

En la primera línea de código se tiene como argumento de entrada para la función tsne, la matriz *caract* de dimensiones nx339 que son las características obtenidas en la tabla final, y a la salida se obtiene una matriz TSN de dimensiones nx2.

[4] TSNE es un algoritmo que permite reducir la dimensionalidad, para la visualización de datos de alta dimensión, se basa en incrustar puntos de alta cota en dimensiones bajas de una manera que respete las similitudes entre los puntos [34].

En la tercera línea de código se grafica la primera columna de la matriz TSN en el eje de las abscisas y la segunda columna en el eje de las ordenadas, *time* es el vector columna del tiempo de la tabla final de dimensiones nx1 o también conocido como el tamaño de la muestra.

En la figura 2.9 se muestra el gráfico tsne de la tabla final, en donde los puntos de color rojo representan las características en dos dimensiones, extraídas de un segmento de tamaño 1 segundo. Las de color amarillo representan las características extraídas de un segmento de 2 segundos, las de color verde son las características extraídas de un segmento de 4 segundos, las de color azul de un segmento de 8 segundos y las de color morado de un segmento de 16 segundos.

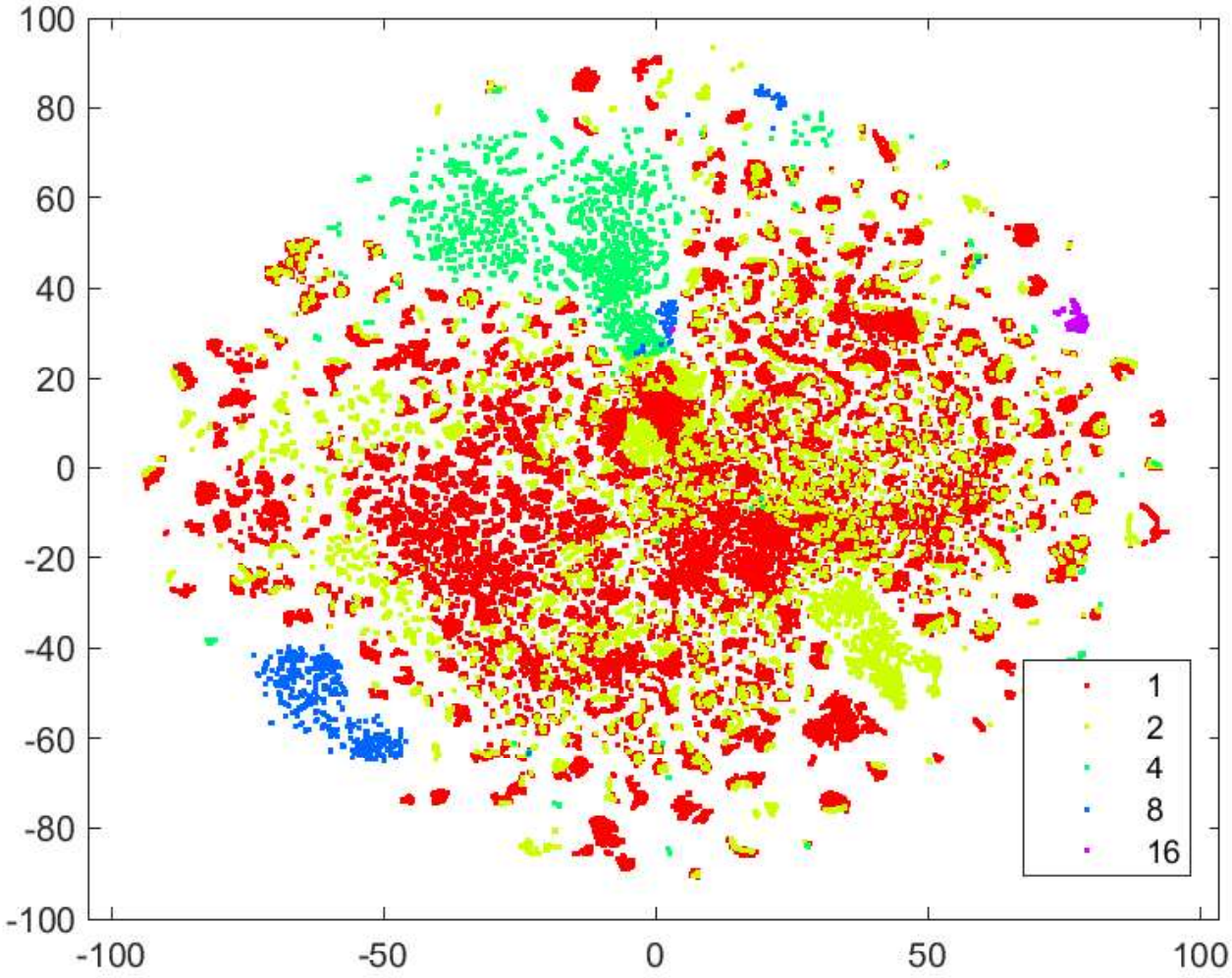

Figura 2.9. Gráfico del tsne de la tabla final

Cada agrupación de colores representa un cluster, para determinar el tamaño que permite extraer características confiables se calcula la dispersión de cada uno de los clusters, y se elige cuál de ellos presenta menor dispersión.

Para el cálculo de la dispersión se encuentra el promedio de la distancia euclidiana entre cada uno de los puntos $z_i = \{x_i, y_i\}$ y el centroide $c = (c_1, c_2)$ como se muestra en la fórmula a continuación [35]:

$$Medida\ de\ dispersión = \frac{1}{n}\sum_{i=1}^{n} \|z_i - c\|$$

$$\|z_i - c\| = \sqrt{(x_i - c_1)^2 + (y_i - c_2)^2}$$

En la figura 2.10 se muestra el cluster de las características extraídas de un segmento de 1 segundo.

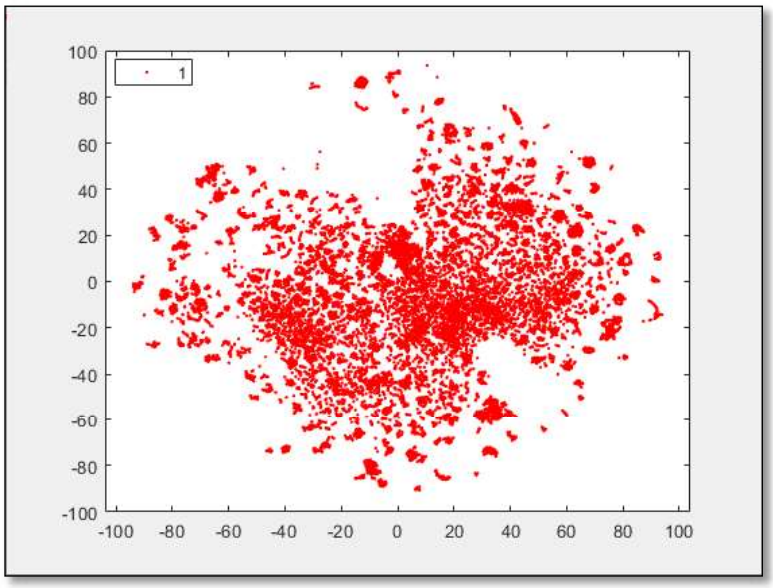

Figura 2.10. Gráfico del tsne de 1 segundo

Medida de dispersión = 48,1798

En la figura 2.11 se muestra el cluster de las características extraídas de un segmento de 2 segundos.

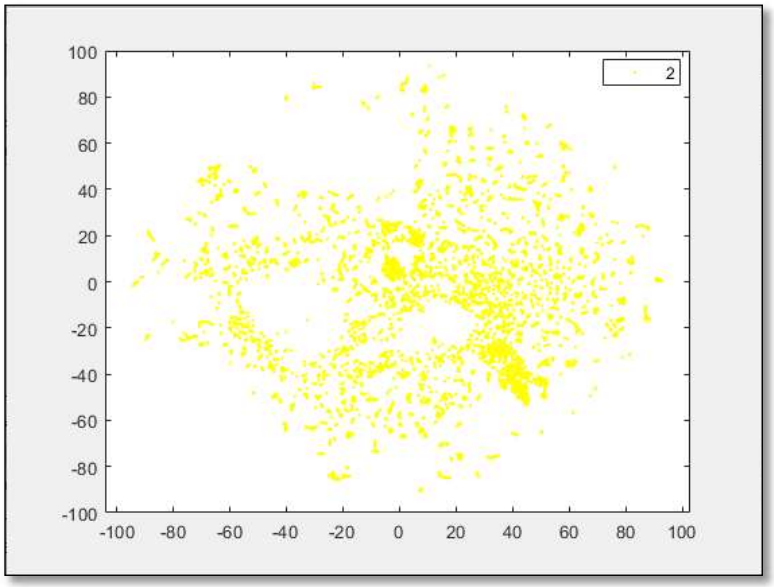

Figura 2.11. Gráfico del tsne de 2 segundo

Medida de dispersión = 50,5796

En la figura 2.12 se muestra el cluster de las características extraídas de un segmento de 4 segundos.

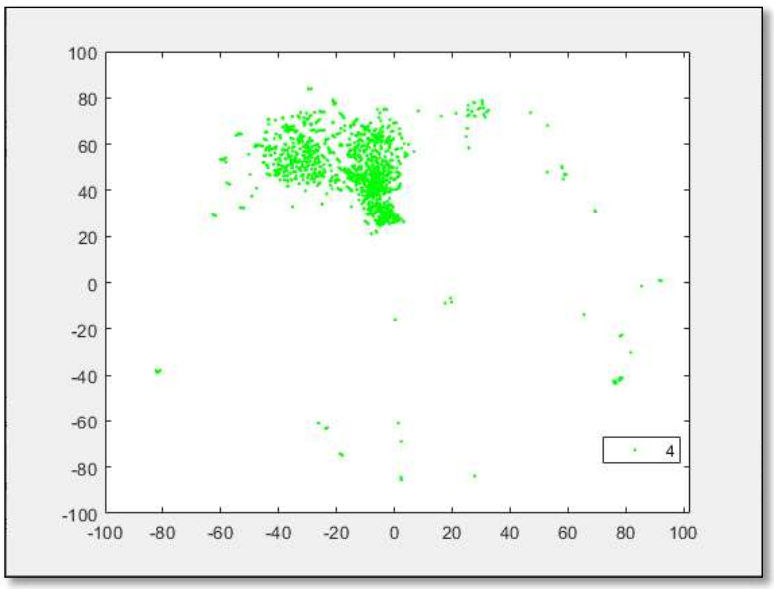

Figura 2.12. Gráfico del tsne de 4 segundo

Medida de dispersión = 24,1987

En la figura 2.13 se muestra el cluster de las características extraídas de un segmento de 8 segundos.

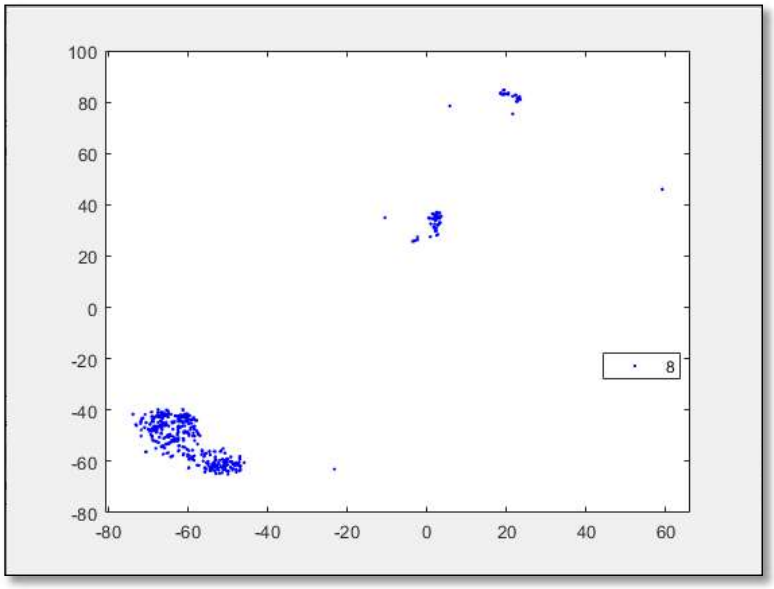

Figura 2.13. Gráfico del tsne de 8 segundo

Medida de dispersión = 38,3083

No se ha tomado en cuenta el cálculo de la dispersión del cluster con segmentos de 16 segundos por que en comparación con los otros clusters es muy pequeño, como se aprecia en la figura 2.9.

Después del análisis previo se ha determinado que el tamaño del segmento que permite extraer características confiables es de 4 segundos, por tener menor dispersión en comparación con los segmentos de los otros tamaños.

2.4 RECONSTRUCCIÓN DEL CONJUNTO DE DATOS PVA_Dryrun

Una vez que se conoce cuál es el tamaño de la muestra que permite tener una extracción de características confiables, se normaliza cada audio a ese tamaño y se extraen sus características.

Para la normalización, se parte de los segmentos que se obtienen después de depurar un registro de voz, se toma el segmento con mayor longitud y se seleccionan 4 segundos de ese segmento. Después se extraen sus características. Se realiza el mismo procedimiento para el siguiente registro de voz hasta completar los 779 registros.

Con la información de las características de todos los registros de voz, se van a reconstruir los archivos plmpva_test-WithPDRS.csv y plmpva_train.csv, los cuales se explicaron en la sección 2.1.1.

En la tabla 2.4 se indican los campos que tiene el archivo plmpva_test-WithPDRS.csv que son los mismos del archivo plmpva_train.csv, la descripción de cada campo se encuentra en el anexo A. Para la reconstrucción de los archivos que se los llamará **plmpva_test-WithPDRS-Actualizado.csv** y **plmpva_train-Actualizado.csv** se seleccionan los campos más relevantes con la información del archivo original, se remplazan algunos campos con la información obtenida en el desarrollo del presente proyecto y se eliminan algunos campos que no son importantes.

Tabla 2.4. Campos de los archivos plmpva_test-WithPDRS.csv y plmpva_train.csv

Campos de los archivos plmpva_test-WithPDRS.csv y plmpva_train.csv	**Campos de los archivos plmpva_test-WithPDRS-Actualizado.csv y plmpva_train-Actualizado.csv**
call_timestamp	El campo call_timestamp si está incluido en los archivos actualizados, porque es la fecha y hora en la que inició la llamada de voz.
recording_duration	El campo recording_duration no está incluido en los archivos actualizados, porque ahora todos los audios tienen el mismo tamaño.
Callref	El campo callref si está incluido en los archivos actualizados, ya que es el identificador del registro de voz.
audio_duration	El campo audio_duration no está incluido en los archivos acualizados, porque ahora todos los audios tienen el mismo tamaño.
voice_indexstart	El campo voice_indexstart es remplazado por el campo segment_indexstart, el cual indica la muestra en donde inicia el segmento de 4 segundos.
voice_indexend	El campo voice_indexend es remplazado por el campo segment_indexend, el cual indica la muestra en donde termina el segmento de 4 segundos.
voice_code	El campo voice_code si está incluido en los archivos actualizados, ya que indica la calidad del registro de voz si es bueno o es malo.

voice_usable_duration	El campo voice_usable_duration no está incluido en los archivos actualizados, porqué todos los audios tienen el mismo tamaño.
feature01 - feature38	Los campos feature01 – feature38 son remplazados por los campos feature01 – feature339, que son las características extraídas utilizando el toolbox.
user_id_hashed	El campo user_id_hashed si está incluido en los archivos actualizados, ya que es el identificador del paciente.
pvi_created_at	El campo pvi_created_at si está incluido en los archivos actualizados, ya que es la fecha y hora en la que el número de referencia único fue recogido por PatientLikeMe
current_age	El campo current_age si está incluido en los archivos actualizados ya que es la edad del participante.
sex	El campo sex si está incluido en los archivos actualizados ya que es el sexo del participante.
years_since_first_symptom	El campo years_since_first_symptom si está incluido en los archivos actualizados ya que indica cuantos años son desde que el paciente se dio cuenta del primer síntoma.
pdrs_date	El campo pdrs_date si está incluido en los archivos actualizados ya que es la fecha en la que se aplicó el cuestionario PDRS.
pdrs_reported_at	El campo pdrs_reported_at si está incluido en los archivos actualizados ya que es la fecha y hora en la que el PDRS fue recogido por PatientLikeMe.
days_from_pvi_to_pdrs	El campo days_from_pvi_to_pdrs si está incluido en los archivos actualizados ya que es el número de días transcurridos entre la fecha de referencia PDRS (pdrs_date) y el número único de referencia de llamada recopilado (pvi_created_at).

pdrs_score	El campo pdrs_score si está incluido en los archivos actualizados ya que es la suma de las 17 preguntas del cuestionario PDRS)
Memory	El campo memory si está incluido en los archivos actualizados ya que es la pregunta 1 del cuestionario PDRS.
hallucinations	El campo hallucinations si está incluido en los archivos actualizados ya que es la pregunta 2 del cuestionario PDRS.
Mood	El campo mood si está incluido en los archivos actualizados ya que es la pregunta 3 del cuestionario PDRS.
Motivation	El campo motivation si está incluido en los archivos actualizados ya que es la pregunta 4 del cuestionario PDRS.
Speech	El campo speech si está incluido en los archivos actualizados ya que es la pregunta 5 del cuestionario PDRS.
Saliva	El campo saliva si está incluido en los archivos actualizados ya que es la pregunta 6 del cuestionario PDRS.
Swallowing	El campo swallowing si está incluido en los archivos actualizados ya que es la pregunta 7 del cuestionario PDRS.
Handwriting	El campo handwriting si está incluido en los archivos actualizados ya que es la pregunta 8 del cuestionario PDRS.
Cutting_food	El campo cutting_food si está incluido en los archivos actualizados ya que es la pregunta 9 del cuestionario PDRS.
Dressing	El campo dressing si está incluido en los archivos actualizados ya que es la pregunta 10 del cuestionario PDRS.

Hygiene	El campo higiene si está incluido en los archivos actualizados ya que es la pregunta 11 del cuestionario PDRS.
Turning_in_bed	El campo turning_in_bed si está incluido en los archivos actualizados ya que es la pregunta 12 del cuestionario PDRS.
Falling	El campo falling si está incluido en los archivos actualizados ya que es la pregunta 13 del cuestionario PDRS.
Freezing	El campo freezing si está incluido en los archivos actualizados ya que es la pregunta 14 del cuestionario PDRS.
Walking	El campo walking si está incluido en los archivos actualizados ya que es la pregunta 15 del cuestionario PDRS.
Tremors	El campo tremors si está incluido en los archivos actualizados ya que es la pregunta 16 del cuestionario PDRS.
Numbness	El campo numbness si está incluido en los archivos actualizados ya que es la pregunta 17 del cuestionario PDRS.
hoehn_yahr	El campo hoehn_yahr si está incluido en los archivos actualizados ya que es la escala hoehn_yahr
on_treatment_id	El campo on_treatment_id si está incluido en los archivos actualizados ya que indica si la persona está en tratamiento con “true” y si no está en tratamiento con “false”.
calls_per_user	El campo call_per_user si está incluido en los archivos actualizados ya que es el número de llamadas realizadas por cada participante.

3. RESULTADOS Y DISCUSIÓN

En este capítulo se presentan los resultados más relevantes obtenidos en la depuración de la base de datos PVA-dryrun.

De cada uno de los 779 registros de voz se ha obtenido un gráfico que muestra la depuración, en esta sección se presentan solo algunos de ellos. Los gráficos de todos los registros se encuentran en los anexos digitales Resultados_Gráficos-pvaTEST.doc y Resultados_Gráficos-pvaTRAIN.doc.

Un ejemplo del gráfico que se obtiene, se muestra en la figura 3.1, la cual se divide en 4 partes, en la primera se tienen el gráfico de la señal original, en la segunda se grafica la energía de la señal, esta parte permite determinar si la señal tiene silencio y saturación, en el tercer gráfico se encuentra la señal ya depurada, la parte de color gris indica lo que ha sido eliminado ya sea por silencio o saturación y la parte de color rojo indica la parte buena de la señal, finalmente de la parte buena de la señal se seleccionan 4 segundos como se muestra en el cuarto gráfico, si no se pueden seleccionar los 4 segundos, el audio se considera como malo.

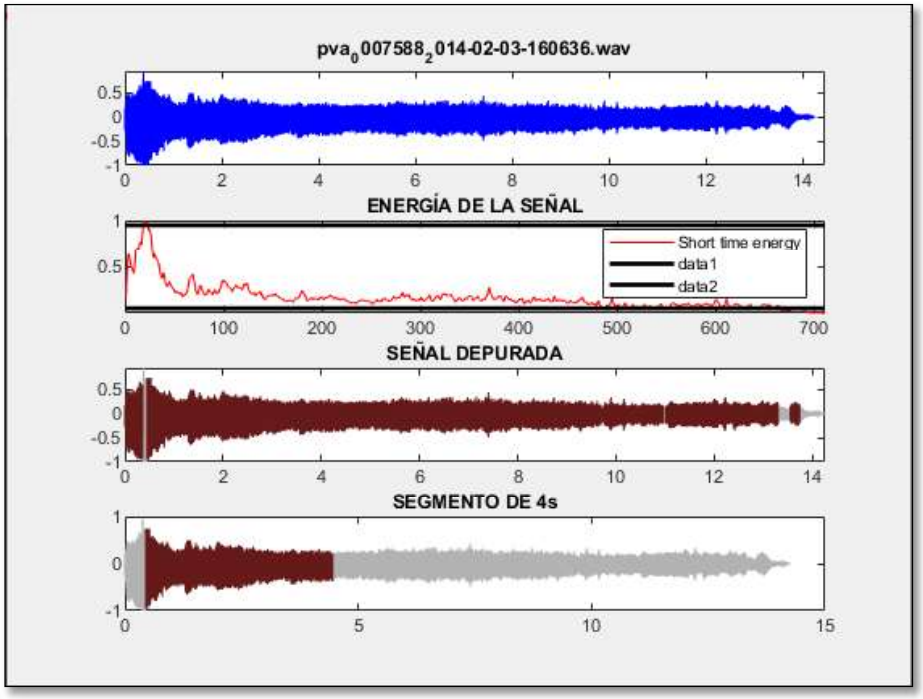

Figura 3.1. Audio pva_0007588_2014-02-30-160636.wav

Otro ejemplo se muestra en la figura 3.2, en la cual se analiza el registro de voz pva_0330573_2014-02-20-232818.wav.

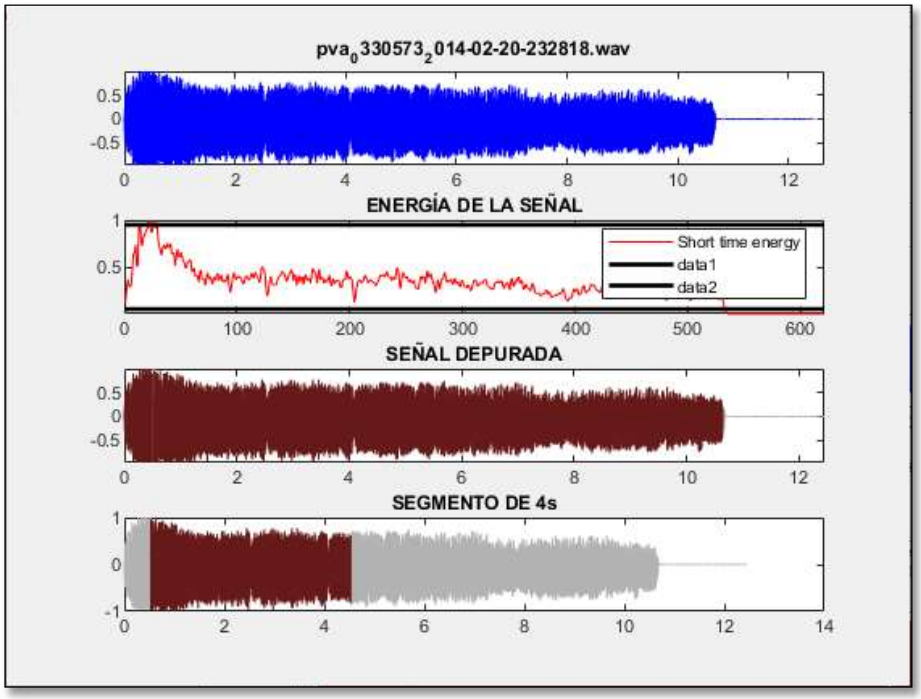

Figura 3.2. Audio pva_0330573_2014-02-20-232818.wav

En la figura 3.3 se muestra un registro de voz que es considerado como malo, porque después de eliminar silencio y saturación no hay ningún segmento que tenga 4 segundos, para poder normalizarlo. Esta situación se repite en varios registros de voz.

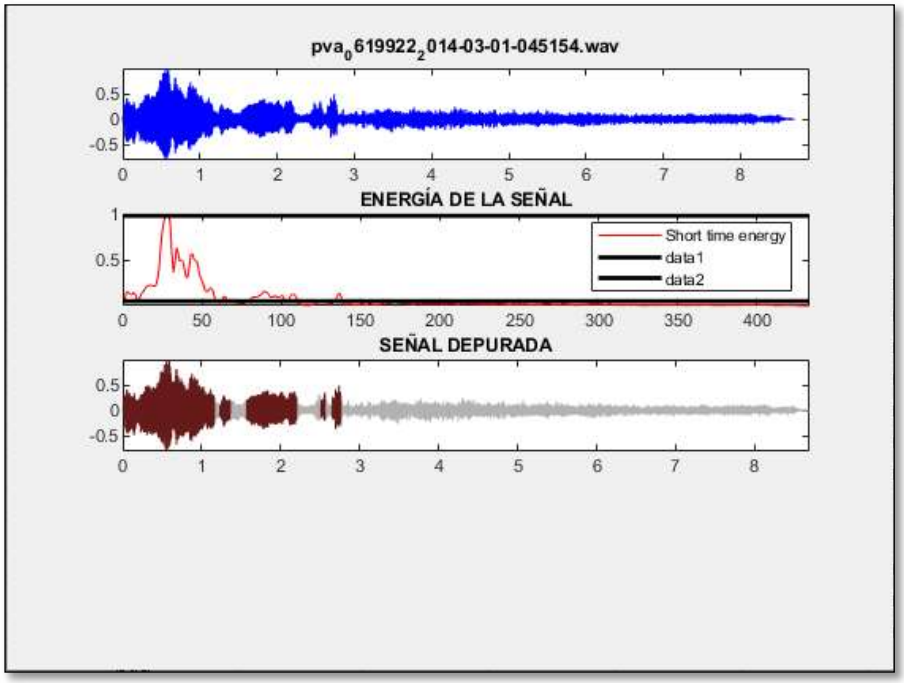

Figura 3.3. Audio pva_619922_2014-03-01-045154.wav

Los resultados que también son considerados malos, son aquellos que en la grabación original tienen una duración menor a 4 segundos, como se puede ver en las figuras 3.4 y 3.5.

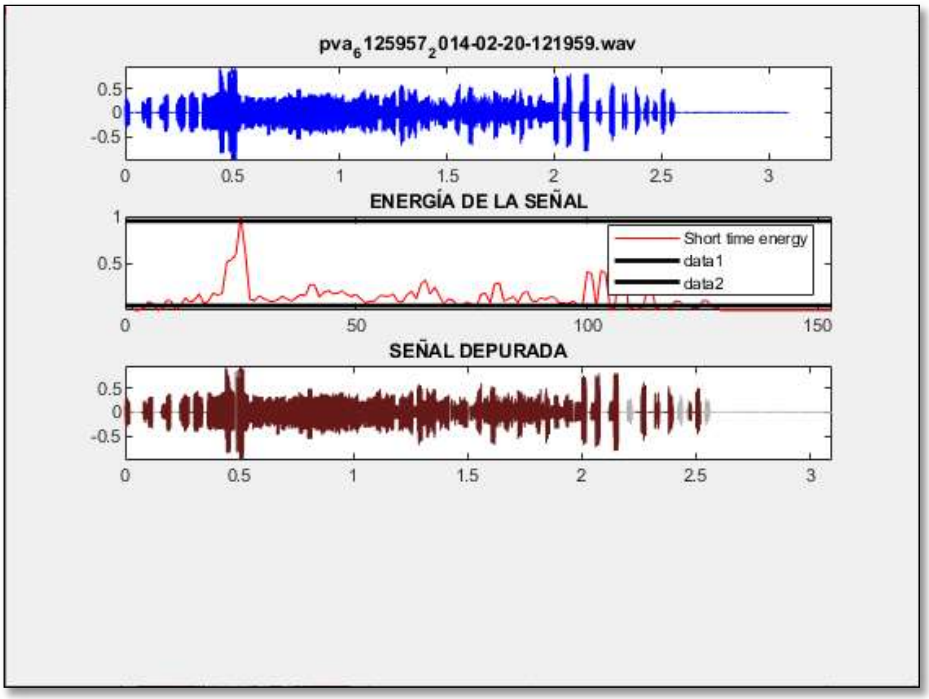

Figura 3.4. Audio pva_6125957_2014-02-20-121959.wav

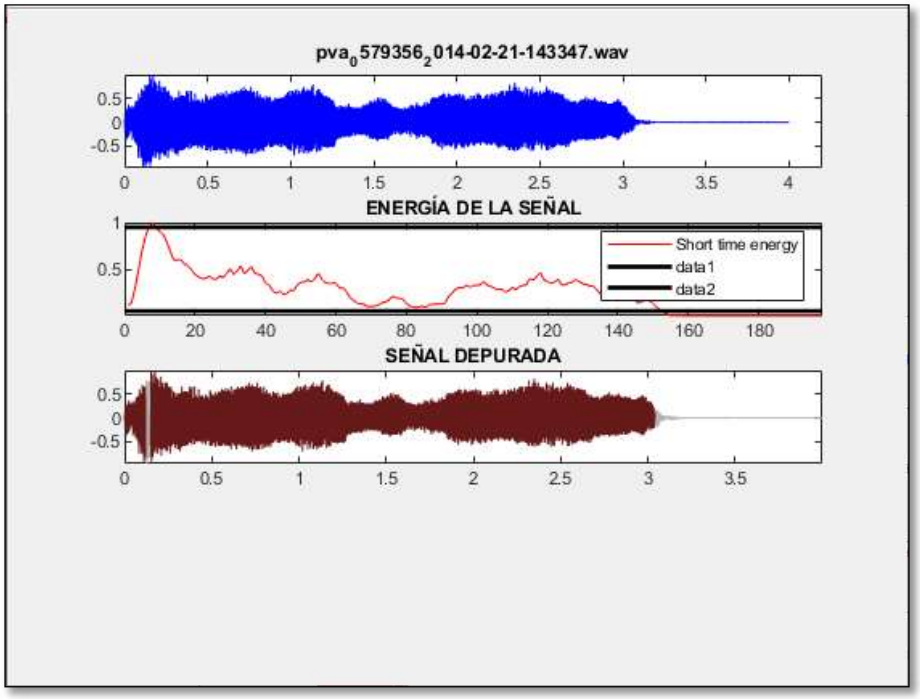

Figura 3.5. Audio pva_6125957_2014-02-20-121959.wav

Además se analizaron los resultados de las figuras 3.5 y 3.6. En la figura 3.5 a primera vista se podría pensar que la señal está saturada casi en su totalidad, pero con el análisis realizado en el presente proyecto, el cual utiliza el concepto de energía en corto tiempo se determina que la señal no está saturada. Mientras que en la figura 3.6 se podría decir a simple vista que la señal no está saturada, lo cual se observa que no es correcto.

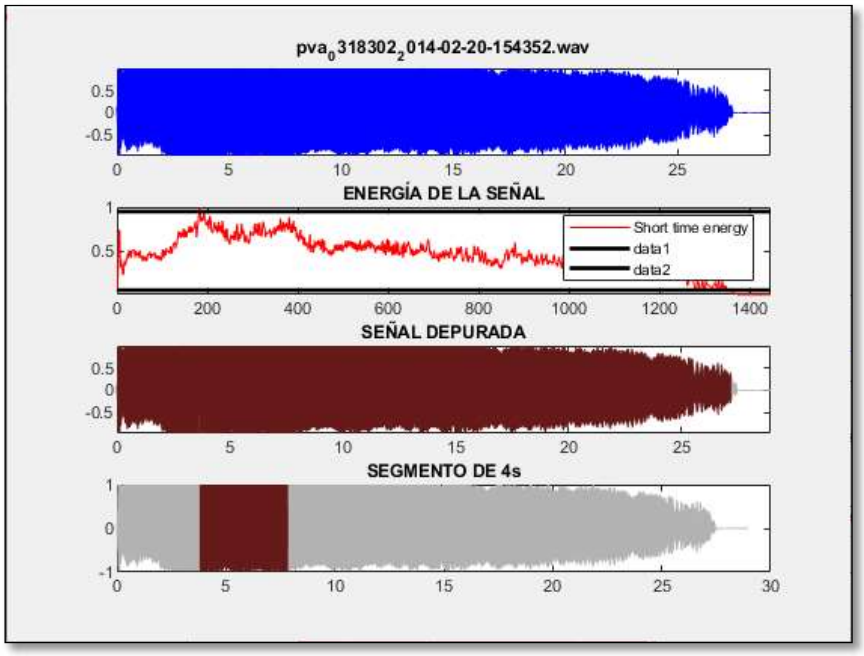

Figura 3.5. Audio pva_0318302_2014-02-20-154352.wav

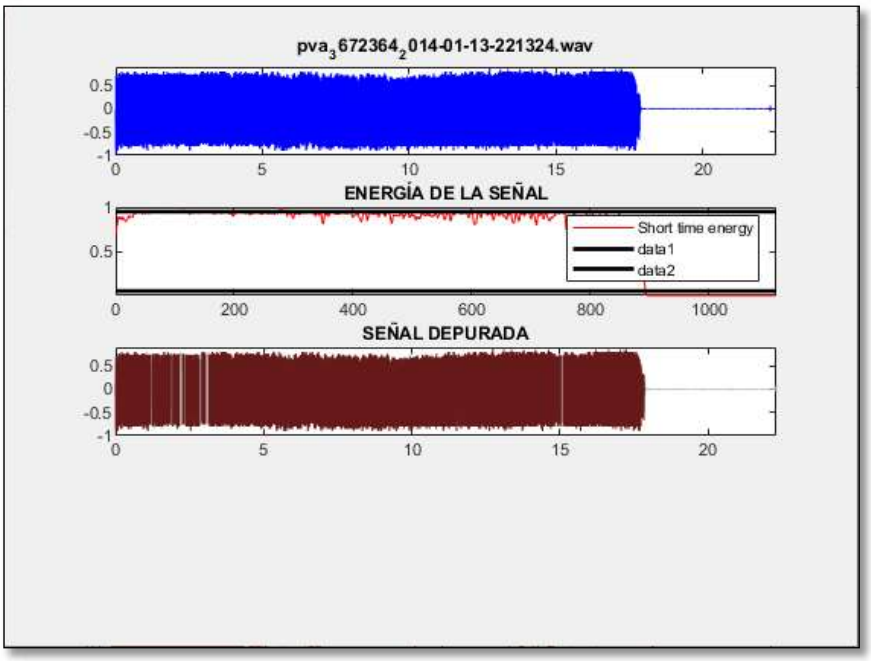

Figura 3.6. Audio pva_3672364_2014-01-13-221324.wav

Finalmente se tiene un número mucho mayor de audios malos en la base de datos actualizada que en la anterior, ya que en la anterior se utiliza control de calidad superficial para determinar si son audios buenos o malos. Un ejemplo de un audio que antes era considerado como bueno y ahora no, se muestra en la figura 3.7. Y así como éste, existen varios audios.

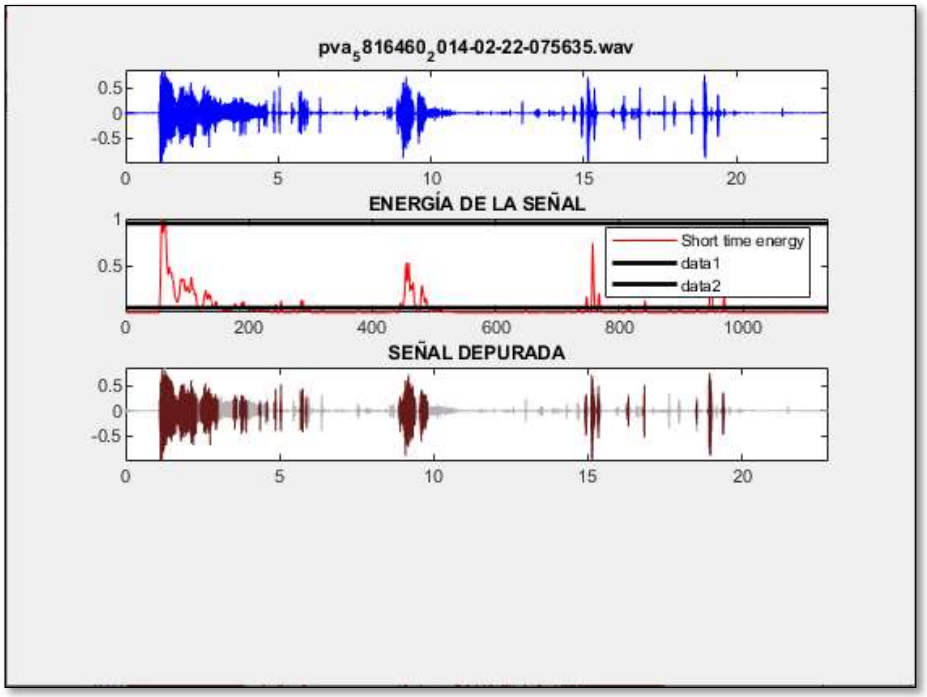

Figura 3.7. Audio pva_5816460_2014-02-22-075635.wa

Una vez que se han normalizado los audios a un tamaño de 4 segundos, se obtienen las características utilizando el *toolbox voice analysis.* Los resultados obtenidos se presentan en los anexos digitales plmpva_WithPDRS-Actualizado.xlsx y plmpva_train-Actualizado.xlsx.

4. CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

- Con el fin de realizar un aporte en la detección de la enfermedad de Parkinson (EP) por medio de la voz, en el presente proyecto se trabaja con registros de voz de personas que padecen la EP, los cuales se depuran para luego extraer sus características. Para la depuración se ha propuesto eliminar silencio y saturación, utilizando el criterio de energía en corto tiempo, y para la extracción de características se ha propuesto normalizar a 4 segundos los registros de voz depurados, ya que en ese tamaño las características presentan menor dispersión.

- La base de datos PVA-dryrun, antes de ser analizada en el presente trabajo de titulación, utilizó el método de control de calidad superficial para depurar los registros de voz, obteniendo pocos registros no válidos para la extracción de características. Como consecuencia se puede decir que al ser un método subjetivo, no mantiene el mismo criterio en todos los registros de voz, generando una depuración poco confiable. Mientras que el método de depuración utilizado en este estudio, ha sido un método objetivo basado en el criterio de energía en corto tiempo, con el cual a pesar de que se obtiene un número mayor de audios no válidos para la extracción de características, se tiene la certeza de que se aplica el mismo criterio en todos los registros de voz.

- Dentro del análisis realizado para determinar el tamaño mínimo que debe tener un registro de voz para una extracción de características confiable, se propuso analizar cómo están “clusterizadas” las características. Para lo cual, se analizaron las características extraídas de segmentos de audio de tamaño 1, 2, 4, 8 y 16 segundos, usando la función TSNE, que permite “clusterizar” las características, llegando a la conclusión que segmentos de audio de 4 segundos presentan menor distorsión, es decir menor variabilidad de sus características.

- Una vez realizada la reconstrucción del conjunto de datos PVA-dryrun, se obtiene un conjunto de datos con una cantidad de información suficientemente grande que lo sigue posicionando como uno de los mejores conjuntos de datos de acceso libre. Además, es importante recalcar, que en el conjunto de datos reconstruido cada registro de voz tiene un número mucho más grande de características, las cuales están respaldadas con su fórmula matemática, aspecto que no posee ningún otro conjunto de datos referente al tema.

4.2. RECOMENDACIONES

- Utilizar el conjunto de datos original, el cual tiene registros de voz de diferentes tamaños, de cada registro extraer 339 características utilizando el toolbox. Luego realizar los siguientes pasos para la detección de la EP (selección y clasificación de características), el resultado obtenido comparar con el resultado de realizar el mismo procedimiento de selección y clasificación de características, pero con el conjunto de datos obtenido en el presente proyecto de titulación.

  Se realiza la comparación para poder conocer si el tamaño de los registros de voz es un parámetro influyente en los valores obtenidos en la extracción de las características.

- Emplear un método de análisis diferente para la detección del tamaño mínimo confiable que debe tener un registro de voz, con el fin de confirmar el resultado obtenido en el presente proyecto de titulación.

- Verificar que la posición del segmento de voz (4 segundos en el presente proyecto), no afecta en los valores obtenidos en la extracción de características. Esto se lo podría realizar utilizando un conjunto de datos confiable, del que se tenga la certeza de los valores que se deben obtener en la extracción de características.

5. REFERENCIAS BIBLIOGRÁFICAS

[1] T. Villa, “Metodología de análisis tiempo-frecuencia para la evaluación automática de la voz de pacientes con enfermedad de Parkinson”, M.S Tesis, Universidad de Antoquia, Medellín, Colombia 2015 [Online]. Available: http://bibliotecadigital.udea.edu.co/bitstream/10495/3737/1/VillaTatiana_2016_METODOLOG%C3%8DA%20%20AN%C3%81LISIS%20TIEMPO.pdf

[2] “Espacio Parkinson”, 2017, *Parkinson Federación Española* [Online]. Available: http://www.esparkinson.es/wp-content/uploads/2017/10/1.1-CONOCE-LA-ENFERMEDAD-P%C3%A1rkinson-causas-diagn%C3%B3stico-y-evoluci%C3%B3n.pdf

[3] Martínez-Sánchez F. Trastornos del habla y la voz en la enfermedad de Parkinson. Rev Neurol 2010; 51:542 [Online]. Available: https://www.researchgate.net/profile/Francisco_Martinez-Sanchez2/publication/47556827_Speech_and_voice_disorders_in_Parkinson%27s_disease/links/58f9e00d4585152edece779c/Speech-and-voice-disorders-in-Parkinsons-disease.pdf

[4] Y. Alemami and L. Almazaydeh, “Detection of parkinson disease through voice signal features”, *The Journal of American Science*, 10(10):44, 2014.

[5] O. Aguilera, F. Sanabria, D. Escobedo, and I. Núñez (2015), “Alteración de parámetros acústicos de la voz y el habla en la enfermedad de Parkinson”. XIV Simposio Internacional de Comunicación Social. Comunicación Social: Retos y Perspectivas Vol. II. Ediciones Centro de Lingüística Aplicada, Santiago de Cuba. [Online]. Available: https://www.researchgate.net/profile/Daniel_Escobedo_Beceiro/publication/294874660_ALTERACION_DE_PARAMETROS_ACUSTICOS_DE_LA_VOZ_Y_EL_HABLA_EN_LA_ENFERMEDAD_DE_PARKINSON/links/56c4ee6d08ae7fd4625a4939.pdf

[6] A. Little, E. McSharry, J. Hunter, J. Spielman and L. Ramig. “Suitability of dysphonia measurements for telemonitoring of Parkinson´s disease”. IEEE transactions on biomedical engineering, 56(4):1015-1022, 2009, [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3051371/#R16

[7] J. Arias, E. Belalcazar, J. Orozco, J. Arias y J. Vargas, “Assessment of fourier methods and maximun entropy for automatic detection of Parkinson’s disease”,

[Online]. Avaliable: http://blade1.uniquindio.edu.co/uniquindio/revistainvestigaciones/adjuntos/pdf/59f1_67-74.pdf

[8] R. Arefi and E. Tahami, “Voice Analysis for Detectiong Parkinson’s Disease”, 18th Iranian Conference on BioMedical Engineering, 14-16 December 2011, Tehran, Iran, [Online]. Avaliable:https://www.researchgate.net/profile/Reza_Arefi3/publication/283045806_Voice_Analysis_for_Detecting_Parkinson's_Disease/links/5627609c08ae4d9e5c4f14cc.pdf

[9] Patient Voice Analysis (PVA), [Online]. Avaliable: https://www.synapse.org/#!Synapse:syn2321745/wiki/62077

[10] G. Yadav, Y. Kumar and G. Sahoo, "Predication of Parkinson's Disease Using Data Mining Methods: A Comparative Analysis Of Tree, Statistical, And Support Vector Machine Classifiers", *6th Indian Journal of Medical Science*, [Online]. Avaliable: https://www.researchgate.net/publication/235421872_Predication_of_Parkinson's_disease_using_data_mining_methods_A_comparative_analysis_of_tree_statistical_and_support_vector_machine_classifiers

[11] R. Santibáñez, "James Parkinson (1755 - 1824) y su parálisis agitante", [Online]. Avaliable: http://revecuatneurol.com/wp-content/uploads/2015/06/REVISTA-ECUATORIANA-DE-NEUROLOGIA-Volumen-13-N%C3%BAmero-3-del-2004.pdf

[12] M. Peker, B. Sen and D. Delen, "Computer-Aided Diagnosis of Parkinson's Disease Using Complex-Valued Neural Networks and mRMR Feature Selection Algorithm", *Journal of Healthcare Engineering,* vol. 6, no. 3, pp. 281-302, 2015, [Online]. Avaliable: https://www.hindawi.com/journals/jhe/2015/493865/abs/

[13] A. Tsanas, Max A. Little, "Accurate telemonitoring of Parkinson's disease progression by non-invasive speach test", *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 4, April, 2010, [Online]. Avaliable: https://ieeexplore.ieee.org/document/5339170

[14] A. Reza, «Voice Analysis for Detecting Parkinson's Disease Using Genetic Algorithm and KNN Classification Method,» de *18th Iranian Conference on BioMedical Engineering* , Iran, 2011.

[15] C. Hui, H. Chang, Y. Xin, X. Xin, S. Xin, W. Gang y W. Su, «An efficient diagnosis system for detection of Parkinson's disease using fuzzy k-nearest neighbor approach,» *Expert Systems with Applications,* vol. 40, pp. 263-271, 2013.

[16] C. Mehmet, C. Bayram y T. Inayet, «Automatic Recognition of Parkinson's Disease from Sustained Phonation Test Using ANN and Adaptive Neuro-Fuzzy Classifier,» *Engineering Science and Design ,* vol. 1, nº 2, pp. 59-64, 2010.

[17] S. Betul, I. Erdem, S. Okan, S. Ahmet, G. Fikret, D. Sakir, A. Hulya y K. Olcay, «Collection and Analysis of a Parkinson Speech Dataset With Multiple Types of Sound Recordings,» *IEEE of Biomedical and Health Informatics,* vol. 17, nº 4, pp. 828-834, 2013.

[18] P. Gómez, V. Rodellar, V. Nieto, C. Muñoz, L. Mazaira, R. Martínez, A. Álvarez, C. Ramírez y M. Fernández, «Characterizing Neurological Disease from Voice Quality Biomechanical Analysis,» *Cognitive Computation ,* vol. 5, nº 4, pp. 399-425, 2013.

[19] M. Correa, «Análisis del efecto del Parkinson en el teblor de la voz: frecuencia fundamental,» Proyecto fin de grado. Dpto. Teoría de la señal y comunicaciones. Univ. Escuela Técnica superior de ingeniería y sistemas de telecomunicaciones. Madrid, 2016.

[20] A. Ortiz y V. Arellano, «Extracción de párametros y reconocimientos de voz esofágica utilizando transformada wavelet y redes neuronales,» . Trabajo de grado. Inst. Tecnológico y de estudios superiores de Monterrey, México, 2007.

[21] I. Cobeta, F. Núñez y S. Fernández, Patología de la voz, *Sociedad Española de Otorrinolaringología y Patología Cérvico-Facial*, Barcelona: ICG Margen, SL, 2013.

[22] A. Tsanas, «Accurate telemonitoring of Parkinson´s disease symptom severity using nonlinear speech signal processing and statistical machine learning,»Tesis Ph.D, University of Oxford, 2012.

[23] A. Zambrano, «Exploración de Nuevos Métodos para la Detección de la Enfermedad de Parkinson mediante la voz,» Trabajo de fin de Máster. Univ. Politécnica de Madrid. España, 2017.

[24] A. Tsanas, M. Zañartu, M. Little, C. Fox, L. Raming y G. Clifford, «Robust fundamental frequency estimation in sustained vowels: detailed algorithmic

comparisons and information fusion with adaptive Kalman filtering,» *Acoustical Society of America,* pp. 2885-2901, 2014.

[25] P. Texeira y P. Fernandes, «Jitter, Shimmer and HNR classification within gender, tones and vowels in healthy voices,» *Proceida Technology ,* vol. 16, pp. 1228-1237, 2014.

[26] A. Tsanas, M. Little, P. McSharry, J. Spielman y L. Raming, «Novel Speech Signal Processing Algorithms for High-Accuracy Classification of Parkinson´s Disease,» *IEEE Transactions on biomedical engineering ,* vol. 59, nº 5, pp. 1264-1271, 2012.

[27] A. Tsanas, «Thanasis,» 23 Noviembre 2016. [Internet]. Available: https://people.maths.ox.ac.uk/tsanas/index.html

[28] J. Schoentgen and R. Guchteneere, «Time Series Analysis of Jitter,» *Phonetics,* vol. 23, pp. 189-201, 1995.

[29] C. Moler, «MathWorks,» 1984, Desviación Estandar [Internet]. Available: https://la.mathworks.com/help/matlab/ref/std.html.

[30] I. Robertson, «Calculating Percentiles,» 2004. [Internet]. Available: https://web.stanford.edu/class/archive/anthsci/anthsci192/anthsci192.1064/handouts/calculating%20percentiles.pdf.

[31] H. Kasuya, S. Ogawa y S. Ebihara, «An acoustic analysis of pathological voice and its application to the evaluation of laryngeal pathology,» *Speech Communication ,* vol. 5, pp. 171-181, 1986.

[32] M. Jalil, F. Awais y A. Malik, «Short-Time Energy, Marnitude, Zero Crossing Rate and Autocorrelation Measurement for Discriminating Voiced and Unvoiced segments of Speech Signals,» *Technological Advances in Electrical, Electronics and Computer Engineering ,* pp. 208-212, 2013.

[33] F. Aguirre, «Desarrollo y análisis de clasificadores de señales de audio,» MSc Tesis. Univ. Politécnica de valencia. Gandia, 2017.

[34] C. Moler, «MathWorks,» 1984, t-Distributed Stochstic Neighbor Embedding, [Internet]. Available: https://la.mathworks.com/help/stats/t-sne.html.

[35] Macro (https://stats.stackexchange.com/users/4856/macro), «StackExchange/2D analog of standard desviation?,» [Internet]. Available:

https://stats.stackexchange.com/questions/13272/2d-analog-of-standard-deviation?answertab=votes#tab-top.

[36] I.R. Titze, "Nonlinear source-filter coupling in phonation: Theory," *Journal of Acoustical Society of America*, Vol. 123(5), pp. 2733-2749, 2008

[37] H.M. Teager, "Some observations on oral air flow during phonation," *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 28(5), pp. 599-601, 1980

ANEXOS

ANEXO A: PLM-PVA Data Dictionary v1

ANEXO B: PLM-PDRS-Questionaire

ANEXO C: SCRIPT voice_analysis.m

ANEXO D: SCRIPT *melcepst.m*

ANEXO E: SCRIPT *wavedec_features.m*

ANEXO F: SCRIPT *fastfda.m*

ANEXO G: SCRIPT *rpde.m*

ANEXO H: SCRIPT *depuration.m*

ANEXO I: SCRIPT *pruebaframe.m*

ANEXO A

PLM-PVA Data Dictionary v1

Field	Description	Type, format/units/values
call_timestamp	Date and time IVR voice call initiated	string, YYYY/MM/DD HH:MM:SS UTC +0000
recording_duration	Length of recording captured by IVR system	integer, seconds
callref	Unique reference number for voice recording	integer
audio_duration	Length of audio captured by IVR system during call	real, seconds
voice_indexstart	Starting audio sample index at which voicing is detected, if at all	integer, samples
voice_indexend	Ending audio sample index at which voicing is detected, it at all	integer, samples
voice_code	Quality of voice recording: 'ok' – (part of) recording usable, 'bad' – recording unusable (no features extracted)	string, 'ok'/'bad'
voice_usable_duration	Length of usable voicing detected	real, seconds
feature01	Median F0	real, Hz
feature02	Mean absolute F0 time derivative	real, Hz^2
feature03	Median absolute F0 time derivative	real, Hz^2
feature04	Mean absolute value of time derivative of RMS power	real, Hz
feature05	Median absolute value of time derivative of RMS power	real, Hz
feature06-feature18	Median cepstral coefficients 0-12 for entire voice recording	13 x real

Feature19-feature31	Mean absolute time derivative of cepstral coefficients 0-12 across entire voice recording	13 x real
feature32	Recurrence period density entropy (RPDE) Hnorm	real
feature33	Detrended fluctuation analysis (DFA) scaling parameter alpha	real
feature34	Modified pitch period entropy (PPE)	real
feature35	Relative spectral power 0-500Hz	real
feature36	Relative spectral power 500-1kHz	real
feature37	Relative spectral power 1kHz-2kHz	real
feature38	Relative spectral power 2kHz-4kHz	real
user_id_hashed	Hashed PatientsLikeMe user ID	string
pvi_created_at	Date and time at which unique reference number for call was collected by PatientsLikeMe	string, YYYY-MM-DD HH:MM:SS.d
current_age	Age of participant when call was recorded	integer, years
sex	Sex of participant	string, 'M'/'F'/empty
years_since_first_symptom	Years since first Parkinson's symptoms detected by participant	integer, years
pdrs_date	Reference date at which PDRS applies	string, YYYY-MM-DD
pdrs_reported_at	Date and time at which PDRS was collected by PatientsLikeMe	string, YYYY-MM-DD HH:MM:SS.d
days_from_pvi_to_pdrs	Number of days elapsed between PDRS reference date (pdrs_date) and unique call	integer, days

	reference number collected (pvi_created_at)	
pdrs_score	Scaled sum of all PDRS entries Q1-Q17 below	integer, 0-100
memory	PDRS Q1	integer, 0-4
hallucinations	PDRS Q2	integer, 0-4
mood	PDRS Q3	integer, 0-4
motivation	PDRS Q4	integer, 0-4
speech	PDRS Q5	integer, 0-4
saliva	PDRS Q6	integer, 0-4
swallowing	PDRS Q7	integer, 0-4
handwriting	PDRS Q8	integer, 0-4
cutting_food	PDRS Q9	integer, 0-4
dressing	PDRS Q10	integer, 0-4
hygiene	PDRS Q11	integer, 0-4
turning_in_bed	PDRS Q12	integer, 0-4
falling	PDRS Q13	integer, 0-4
freezing	PDRS Q14	integer, 0-4
walking	PDRS Q15	integer, 0-4
tremors	PDRS Q16	integer, 0-4
numbness	PDRS Q17	integer, 0-4
hoehn_yahr	Hoehn and Yahr stage at time of reporting PDRS	integer, 1-5
on_treatment_id	Whether participant was ‘on’ vs. ‘off’ medication when PDRS was reported	string, ‘true’/’false’
calls_per_user	Number of calls made by this user	integer

ANEXO B

PLM-PDRS-Questionaire

Parkinson's Disease Rating Scale (PDRS)

This scale measures the impact of your Parkinson's disease on your day-to-day life.

Date

Today or 2013 August 19

Enter your best estimate of the year, month and day if you can't remember the exact date.

When you answer these questions, are you thinking about how you are on-treatment or off-treatment?

On-treatment Off-treatment

Hoehn & Yahr Staging

Parkinson's disease progresses through five stages, called "Hoehn & Yahr" (H&Y) stages. Please indicate one of the five statements below that best describes your status at the moment:

- I have symptoms on one side of my body only.
- I have symptoms on both sides of my body but no problems with balance or walking.
- I'm having problems with my balance and walking.
- I can still stand and walk but with great difficulty.
- I cannot stand or walk independently; I need help or a wheelchair.

1. Memory Problems

- No memory problems
- Mild memory problems; I am often forgetful and can only partially remember some events.
- Moderate memory problems that mean I get disoriented sometimes and can sometimes get confused easily.
- Severe memory problems which mean I have difficulty remembering what day it is or where I am, and often get confused.
- Very severe memory problems; I sometimes forget where I am and what day it is; I have great difficulty making decisions or solving problems.

2. Hallucinations/Visions

- I never see things which are invisible to other people ('hallucinations').
- I experience very vivid dreams.
- Sometimes I see things which are invisible to other people but I know they are not real.
- Sometimes I see things which are invisible to other people and I believe they are real.
- I see things which are invisible to other people most of the time and I believe they are real.

3. Mood

- I do not get sad for long periods.
- I sometimes get sad for longer than normal, i.e. more than a few days or a week.
- I sometimes get sad for greater than a week at a time.
- I get sad for extended periods during which I lose weight, can not sleep, and have poor appetite.
- I get sad for extended periods and have thoughts about suicide.

4. Motivation

- My levels of motivation are about normal.
- I have become less assertive and more passive.
- I have lost some of my initiative and am less interested in hobbies.
- I have lost some of my initiative and am uninterested in my daily routine.
- I am withdrawn and initiate very little activity.

5. Speech

Have you noticed any change in the way you speak? Do you have problems while talking?

- No, the way I speak has not changed.
- Yes, the way I speak has changed, but other people understand me without any problem.
- Yes, the way I speak has changed somewhat, and sometimes I have to repeat to make myself understood.
- Yes, the way I speak has changed enough, and frequently I have to repeat to make myself understood.
- Yes, the way I speak has changed so much that other people has difficulty understanding me or do not understand me at all.

6. Excessive Saliva

Have you noticed that you are producing excessive saliva?

- No, I have not noticed that I have excessive saliva and I do not drool.
- Yes, I have noticed a slight increase in the amount of saliva and on occasion I drool at night on my pillow.
- Yes, I have moderate excess of saliva and occasionally I drool during the day.
- Yes, I have a marked excess of saliva and frequently drool during the day.
- Yes, I drool so much that I have to carry a handkerchief at all times.

7. Swallowing

Do you have difficulty swallowing food? Have you choked trying to swallow food?

- No, I do not have difficulty swallowing and I do not choke.
- Yes, I do have difficulty swallowing but I rarely choke.
- Yes, I do have difficulty swallowing and occasionally choke.
- Yes, I do have difficulty swallowing and I need soft food to be able to eat.
- Yes, I am incapable of swallowing and I need nasogastric intubation or I have had a gastrostomy.

8. Handwriting

Have you noticed changes in your handwriting? Do you have difficulty while writing by hand?

- No, I have not noticed changes in the way I write.
- Yes, my handwriting is somewhat slower or my letter formation is smaller.
- Yes, now my handwriting is moderately slower and my letter formation is smaller, but everything I write can be understood.
- Yes, my handwriting is very altered and there are some words that can not be understood.
- Yes, my handwriting is deteriorated and most of the words can not be understood.

9. Cutting food

When you have to cut foods, have you noticed that it takes you longer to accomplish this task, or is it difficult to handle the utensils?

- No, I do not cut food more slowly and I have no difficulty managing my utensils.
- Yes, I am somewhat slower and more clumsy than before, but I am still capable of eating without help.
- Yes, I am slower and clumsier than before, and I need help cutting some foods.
- Yes, someone has to cut my food, but I can still eat on my own.
- I have to be fed because I can not do it on my own.

10. Dressing

Do you find it difficult to get dressed?

- No, I do not find it difficult to get dressed, nor am I slower than before.
- Yes, I get dressed more slowly, but need little help.
- Yes, I get dressed slower and sometimes need help buttoning my clothes, tying my shoes or getting my arm in the sleeves.
- Yes, I need substantial help to get dressed, but I can still do some things on my own.
- I have to be dressed by someone else.

11. Hygiene

Is it slower or more difficult when you are trying to do any task dealing with hygiene? (bathing, brushing your teeth, fixing your hair, going to the bathroom).

- No, I have not slowed down when performing these activities.
- Yes, I am a little slower dealing with my hygiene, but I do not need help.
- Yes, I am slower and need help bathing and using the facilities.
- Yes, I am slower and need help bathing, brushing my teeth, fixing my hair, and going to the bathroom.
- I need help with everything and I wear a Foley catheter.

12. Turning in bed

Do you have difficulty turning in bed or fixing the blankets?

- No I do not have difficulty turning in bed or fixing the blankets.
- Yes, I am somewhat clumsy or slower turning in bed or fixing the blankets.
- Yes, I am capable of turning in bed or fixing the blankets but with great difficulty.
- Yes, I am capable of turning in bed, but I need help to complete the task.
- I am incapable of turning in bed or fixing the blankets without help.

13. Falling

Have you experienced a fall?

- No, I have not fallen.
- Yes, I have fallen, but rarely.
- Yes, occasionally I have fallen but it happens less than once a day.
- Yes, I fall on average once a day.
- Yes, I fall every day and more than once.

14. Freezing

Have you experienced "freezing" while walking? ("freezing" is defined as being unable to walk because you feel as if your feet are stuck to the ground).

- No, I have not experienced freezing.
- Yes, I have experienced freezing when I walk, but on rare occasions. Or sometimes when I start walking I experience freezing.
- Yes, occasionally I experience freezing.
- Yes, frequently I experience freezing while walking and occasionally I have fallen because of it.
- Yes, frequently I experience freezing while walking and frequently, I have fallen because of it.

15. Walking

Has the way you walk changed? Is it difficult for you to walk?

- No, the way I walk and my arm movement has not changed.
- Yes, the way I walk has changed but it is not a problem.
- Yes, I have moderate difficulty while walking, but I do not need help.
- Yes, I have great difficulty walking and need help.
- I can not walk alone or with help.

16. Tremors

Do you have visible tremors in any part of your body?

- No, I do not have visible tremors.
- Yes, occasionally I do have visible tremors.
- Yes, I have moderate tremors that bother me.
- Yes, I have intense tremors that interfere with some activities.
- Yes, I have intense tremors that interfere with the majority of my activities.

17. Numbness

Do you feel any numbness, tingling, or discrete pain that you can attribute to Parkinson's disease?

- No, I do not feel numbness, tingling, or discrete pain, that I can attribute to my Parkinson's disease.
- Yes, on occasion, I do feel numbness, tingling, or discrete pain, that I can attribute to my Parkinson's disease.
- Yes, frequently, I do feel numbness, tingling, or pain, that I can attribute to my Parkinson's disease.
- Yes, frequently I feel painful sensations attributable to my Parkinson's disease.
- Yes, I feel extreme pain attributable to my Parkinson's disease.

ANEXO C

SCRIPT voice_analysis.m

```
function [measures_vector,measures_name,F0] = voice_analysis(data,fs,
f0_alg)

%% Parameter initialization
warning off all
addpath(genpath(pwd))
if nargin<3 || isempty(f0_alg)
   f0_alg = 'SWIPE';
end
if(nargin==1) % if only one input is provided
   [pathstr, name, ext] = fileparts(data);
   if(strcmp(ext, '.wav')) % a wav file was included as input;
        [data,fs] = audioread(data);
    else
        error('You must provide a valid *.wav file or a vector with the
speech signal and the sampling frequency!')
     end
end

% Default params for DFA
Tmax = 1000;
d = 4; % Embedding dimension
tau = 50; % Embedding delay
eta = 0.2; % RPDE close returns radius
dfa_scaling = (50:20:200)'; % DFA scaling range

f0min = 50; % min F0 possible --- Adjust depending on application!
f0max = 500; % max F0 possible --- Adjust depending on application!
flag = 1;

% pre-process data
data = data - mean(data);
data = data/max(abs(data));

% Use your favourite tool for estimating the fundamental frequency
switch (f0_alg)
   case 'SHRP'
        if (exist('shrp.m'))
          % Sun's implementation -> interested in F0
            F0 = shrp(data, fs, [f0min, f0max], 10);
        else
            disp('SHRP not downloaded; using alternative F0 estimator
instead');
            F0 = F0_Thanasis(data,fs); % avoid problems if the user has
not downloaded the SWIPE algorithm
        end

   case 'SWIPE'
        if (exist('swipep.m'))
          % SWIPE algorithm
            F0 = swipep(data, fs, [f0min, f0max], 0.01);
        else
            disp('SWIPE not downloaded; using alternative F0 estimator
instead');
```

```
38            F0 = F0_Thanasis(data,fs); % avoid problems if the user has
not downloaded the SWIPE algorithm
39        end

40    otherwise
41        error('You must specify an appropriate F0 estimation
algorithm!')
42 end

43 if (exist('dypsa.m'))
    % Work with dypsa
44    [VF_close, VF_open] = dypsa(data,fs);
    % get A0
45    N=length(VF_close);
46    for i=1:N-1
47        A0(i) = max(abs(data(VF_open(i):VF_close(i+1))));
48    end
49 else
50    data_buffered = buffer(data,0.01*fs);
51    A0 = max(data_buffered); %max value every 10 msec
52 end

53 data1=data; % for safety!
54 measures_names = []; %this will be set later!

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      Feature extraction
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% jitter variants
55 [Jitter] = jitter_shimmer(F0);

% shimmer variants
56 [Shimmer] = jitter_shimmer(A0);

% Harmonics to Noise Ratio (HNR) and Noise to Harmonics Ratio (NHR)
57 [HNR,NHR] = HNRFun(data,fs);

% DFA
58 if (exist('fastdfa.m'))
59    dfa = fastdfa(data, dfa_scaling);
60    DFA = 1/(1+exp(-dfa));
61 else
62    DFA = NaN;
63 end

%RPDE
64 if (exist('rpde.m'))
65    data_resampled = resample(data, 25000, fs);% Resample to analysis
rate
66    RPDE = rpde(data_resampled, d, tau, eta, Tmax); %single element
vector
67 else
68    RPDE = NaN;
69 end

% PPE2 (improved version compared to Max's)
```

```
70 F0mean_healthy_control = 120; % mean(F0); % strictly speaking, adjust
this for healthy controls males = 120 and females = 190
71 logF0signal = log_bb(F0/F0mean_healthy_control); % log transform F0 is
beneficial - see my ICASSP2010 paper, also controlling for F0 in healthy
controls
72 ARcoef = arcov(logF0signal, 10); % identify AR coefficients
73 sig_filtered = filter(ARcoef, 1, logF0signal);
74 sig_filtered = sig_filtered(round(0.001*fs):end);
% Obtain measure of dispersion - use simple histogramic means for now
75 PPEd = hist(sig_filtered, linspace(min(sig_filtered),
76 max(sig_filtered), 100));
77 PPE = H_entropy(PPEd/sum(PPEd))/log_bb(length(PPEd));

% Glottis open/closed quotient [Glottis Quotient (GQ) in the JRSI paper]
78 if (exist('dypsa.m'))
79    [GQ] = glottis_quotient(VF_close,VF_open, fs, f0min, f0max, flag);
80 else
81    GQ = NaN*ones(1,3);
82 end

% Glottal to Noise Excitation (GNE) related measures
83 [GNE] = GNE_measure(data,fs);

% Vocal Fold Excitation Ratios (VFER) in the JRSI2011 paper
84 if (exist('dypsa.m'))
85    [VFER] = VFER_measure(data,fs);
86 else
87    VFER = NaN*ones(1,7);
88 end

% Empirical Mode Decomposition Excitations Ratios (EMD-ER) in the JRSI
paper
89 if (exist('emd.m'))
90    [EMD_ER] = IMF_measure(data1);
91 else
92    EMD_ER = NaN*ones(1,6);
93 end

% MFCCs
94 if (exist('melcepst.m'))
95    mfcc = melcepst(data, fs, 'E0dD');
96    MFCCs_mean = mean(mfcc);
97    MFCCs_std = std(mfcc);
98 else
99    MFCCs_mean = NaN*ones(1,42);
100    MFCCs_std = NaN*ones(1,42);
101 end

% Wavelet measures (my NOLTA2010 paper)
102 [wavelet_features, wavelet_feature_names] = wavedec_features(F0);

% Summarize ALL dysphonia measures in a single vector
103 measures_vector = [Jitter, Shimmer, HNR, NHR, GQ, GNE, VFER, EMD_ER,
104 MFCCs_mean, MFCCs_std, wavelet_features, PPE, DFA, RPDE];

% Get the names of the variables correctly
104 load ('measures_names.mat')
```

```
105 end % end of main function

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            Additional functions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

106 function [measures] = jitter_shimmer(A0)
107 mean_Ampl = mean(A0);
% Mean absolute difference of successive cycles
108 measures(1) = mean(abs(diff(A0)));

% Mean absolute difference of successive cycles - expressed in percent(%)
109 measures(2) = 100*mean(abs(diff(A0)))/mean_Ampl;

% Perturbation quotient
110 [Ampl_PQ3] = perq1(A0,3);
111 measures(3) = Ampl_PQ3.classical_Schoentgen;
112 measures(4) = Ampl_PQ3.classical_Baken;
113 measures(5) = Ampl_PQ3.generalised_Schoentgen;

114 [Ampl_PQ5]=perq1(A0,5); % Use 5 cycle samples (Schoentgen)
115 measures(6)=Ampl_PQ5.classical_Schoentgen;
116 measures(7)=Ampl_PQ5.classical_Baken;
117 measures(8)=Ampl_PQ5.generalised_Schoentgen;

118 [Ampl_PQ11]=perq1(A0,11); % Use 11 cycle samples (Schoentgen)
119 measures(9)=Ampl_PQ11.classical_Schoentgen;
120 measures(10)=Ampl_PQ11.classical_Baken;
121 measures(11)=Ampl_PQ11.generalised_Schoentgen;

% zeroth order perturbation
122 measures(12) = mean(abs(A0-mean_Ampl));

% Shimmer(dB)
123 measures(13) = mean(20*(abs(log10((A0(1:end-1))./(A0(2:end))))));

% CV
124 measures(14)=mean((diff(A0)).^2)/(mean_Ampl)^2;

% TKEO
125 measures(15) = mean(abs(TKEO(A0)));
126 measures(16) = std(TKEO(A0));
127 Ampl_TKEO_prc = prctile(TKEO(A0),[5 25 75 95]);
128 measures(17)=Ampl_TKEO_prc(1);
129 measures(18)=Ampl_TKEO_prc(2);
130 measures(19)=Ampl_TKEO_prc(3);
131 measures(20)=Ampl_TKEO_prc(4);

% FM
132 measures(21) = (max(A0)-min(A0))/(max(A0)+min(A0));

133 measures(22) = Ampl_TKEO_prc(4) - Ampl_TKEO_prc(1);

134 end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 Calculate the PQ using the classical PQ formula
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function PQ = perq1(time_series, K)
N = length(time_series);
mean_tseries = mean(time_series);
K1=round(K/2);
K2=K-K1;
p = 5;
sum1=0;
for i = K1:N-K2
   sum1 = sum1+mean(abs([time_series(i-K2:i+K2)]-time_series(i)));
end
PQ.classical_Schoentgen = (sum1/(N-K+1))/(mean_tseries);

sum2=0;
for i = K1:N-K2
   sum2 = sum2+mean(abs([time_series(i-K2:i+K2)]))-time_series(i);
end
PQ.classical_Baken = (sum2/(N-K+1))/(mean_tseries);

% perturbation quotient of the residue
time_series=time_series(:);
sum3=0;
% calculate the AR coefficients (I use the Yule-Walker equations)
new_tseries=(time_series-mean_tseries)';
a = aryule(time_series-mean_tseries,p);
for i = 1+p:N
   sum3 = sum3+abs(sum(a.*(new_tseries(i:-1:i-p))));
end

PQ.generalised_Schoentgen = (sum3/(N-p))/(mean_tseries);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
          Calculate the HNR
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [HNR, NHR] = HNRFun(data,fs)
f0max=500; %Hz -- max value, possibly adjust for other applications
f0min=50; %Hz
tstep=0.01*fs;%M saltos cada 10ms
x=0.08*fs;%M ventanas de 80ms es decir de 640
steps=(length(data)-x)/tstep; %M en cuantas partes de 10ms se va a dividir la señal y se resta 80ms para eliminar partes de silencion
for i=1:steps
tseries = data(i*tstep:i*tstep+x);
   tseries = tseries-mean(tseries);
   Dwindow = hann(length(tseries));
   segment_sig = tseries.*Dwindow;

   %% HNR computation process
   ACF = xcorr(segment_sig,'coeff');
   ACF2 = ACF(length(segment_sig):end);
   aa=fft(segment_sig);
   aa=ifft(abs(aa).^2);
   ACF_Dwindow = xcorr(Dwindow,'coeff');
   ACF_Dwindow2 = ACF_Dwindow(length(Dwindow):end);
   bb=fft(Dwindow);
   bb=ifft(abs(bb).^2);
   ACF_signal = ACF2./ACF_Dwindow2;
   ACF_signal = ACF_signal(1:round(length(ACF_signal)/3));
```

```
    rho=aa./bb;
    rho=rho(1:length(rho)/2);
    rho=rho/max(rho); %M normaliza
    [rx_value,rx_index] = sort(ACF_signal,'descend');
    [d1 d2] = sort(rho, 'descend');
    low_lim=ceil(fs/f0max);  % round towards positive sample number
    up_lim=floor(fs/f0min);  % round towards negative sample number
    k=2;
    while ((rx_index(k)<low_lim) || rx_index(k)>up_lim)
        k=k+1;
    end

    m=2;
    while ((d2(m)<low_lim) || d2(m)>up_lim)
        m=m+1;
    end
    ll(i)=d2(m);
    mm=d2(m);
    HNR_dB_Praat(i) = 10*log10(rho(mm)/(1-rho(mm)));
    NHR_Praat(i) = (1-rho(mm))/rho(mm);

end

%% Summarize data
HNR(1)=mean(HNR_dB_Praat);
HNR(2)=std(HNR_dB_Praat);
NHR(1)=mean(NHR_Praat);
NHR(2)=std(NHR_Praat);

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      Calculate the glottis quotients
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [GQ] = glottis_quotient(VF_close, VF_open, fs, f0min, f0max,
flag)

cycle_open=abs(VF_open(2:end)-VF_close(1:end-1));
cycle_closed=abs(VF_open(1:end-1)-VF_close(1:end-1));

% remove erroneous cycles
if flag
    low_lim=fs/f0max; % lower limit
    up_lim=fs/f0min;  % upper limit
    N=length(cycle_open);
    for i=1:N-1
        if((cycle_open(i) > up_lim) || (cycle_open(i) < low_lim))
            cycle_open(i)=NaN;
        end
        if((cycle_closed(i) > up_lim) || (cycle_closed(i) < low_lim))
            cycle_closed(i)=NaN;
        end
    end
end

%statistics in time
prc1=prctile(cycle_open,[5 95]);
```

```
cycle_open_range_5_95_perc=prc1(2)-prc1(1);
prc2=prctile(cycle_closed,[5 95]);
cycle_closed_range_5_95_perc=prc2(2)-prc2(1);

GQ(1) =
(cycle_open_range_5_95_perc/(cycle_open_range_5_95_perc+cycle_closed_rang
e_5_95_perc));
GQ(2) = (nanstd(cycle_open));
GQ(3) = (nanstd(cycle_closed));

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   Calculate the glottal to noise excitation ratio
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [GNE] = GNE_measure(data, fs)
filt_order=100;
new_fs=10000; % M Nueva frecuencia de muestreo 10k Hz
x = 0.03*new_fs; % se transforman 30ms a muestras =300
tstep=0.01*new_fs; %M saltos de 10ms que en muestras seran 100
BW = 1000; % bandwidth
Fshift = 500; % shift fr
data = resample(data, new_fs, fs);

% cut-off1
Fc1=1:Fshift:(new_fs/2-BW-500); % not to cross Nq freq!
% cut-off2
Fc2=Fc1+BW;

for j=1:length(Fc1);
   d(j)
=fdesign.bandpass('n,fc1,fc2',filt_order,Fc1(j),Fc2(j),new_fs);
   hd(j) = design(d(j));
end
steps=(length(data)-x)/tstep;
for i=1:steps+1
   tseries = data(1+(i-1)*tstep:(i-1)*tstep+x);
   Dwindow = hann(length(tseries));
   segment_sig = tseries.*Dwindow;
   a = lpc(segment_sig,13);
   est_x = filter([0 -a(2:end)],1,segment_sig);    % Estimated signal
   e = segment_sig - est_x;
   LPE = xcorr(e,'coeff');   % LPES
   LPE=LPE(length(LPE)/2:end);
   for ii=1:length(hd)
       sigBW(:,ii)=filter(hd(ii),LPE);
       sig_TKEO(ii) = mean(TKEO(sigBW(:,ii)));
       sig_energy(ii) = mean(sigBW(:,ii)).^2;
   end
   Hilb_tr = hilbert(sigBW);
   Hilb_env = abs(Hilb_tr);
   c = xcorr(Hilb_env);
   [cval,cidx] = max(c);
   GNEm(i) = max(cval);
   signal_BW_TKEO(i,:) = sig_TKEO;
   signal_BW_energy(i,:) = sig_energy;
end
```

```
signal_BW_TKEO2 = mean(log(signal_BW_TKEO)); % used for getting the noise to signal ratio
signal_energy2 = mean(log(signal_BW_energy));

% Set outputs

GNE(1) = mean(GNEm);
GNE(2) = std(GNEm);
gnTKEO = mean(signal_BW_TKEO);
gnSEO = mean(signal_BW_energy);
GNE(3) = sum(gnTKEO(1:2))/sum(gnTKEO(end-3:end));
GNE(4) = sum(gnSEO(1:2))/sum(gnSEO(end-3:end));
GNE(5) = sum(signal_BW_TKEO2(end-3:end))/sum(signal_BW_TKEO2(1:2));
GNE(6) = sum(signal_energy2(end-3:end))/sum(signal_energy2(1:2));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   Calculate the vocal fold excitation ratios
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [VFER] = VFER_measure(data, fs)
filt_order=100;
BW = 500; % bandwidth
Fmax = (fs/2-BW-300); % Max frequency to check
Fshift = floor(fs/100);

% get VF action
[VF_close,VF_open] = dypsa(data,fs);

% cut-off1
Fc1=1:Fshift:Fmax;
% cut-off2
Fc2=Fc1+BW;

for j=1:length(Fc1);
   d(j) = fdesign.bandpass('n,fc1,fc2',filt_order,Fc1(j),Fc2(j),fs);
   hd(j) = design(d(j));
end

for i=1:length(VF_close)-1
   tseries = data(VF_close(i):VF_close(i+1));
   Dwindow = hann(length(tseries)); %Use Hanning window
   segment_sig = tseries.*Dwindow;
   if (length(tseries)>40)
       for ii=1:length(hd)
           thanasis = filter(hd(ii),segment_sig);
           sigBW(:,ii) = thanasis(1:40);
           sig_TKEO(ii) = mean(TKEO(sigBW(:,ii)));
           sig_SEO(ii) = mean(sigBW(:,ii)).^2;
       end
       Hilb_tr = hilbert(sigBW);
       Hilb_env = abs(Hilb_tr);
       c = xcorr(Hilb_env);
       [cval,cidx] = max(c);
       NEm(i) = max(cval);

       signal_BW_TKEO(i,:) = sig_TKEO;
       signal_BW_SEO(i,:) = sig_SEO;
   end
```

```
end

signal_BW_TKEO2 = mean(log(signal_BW_TKEO)); % used for getting the noise to signal ratio

% Set outputs
VFER(1) = mean(NEm);
VFER(2) = std(NEm);
VFER(3) = -sum(NEm.*log_bb(NEm));
VFTKEO = mean(signal_BW_TKEO);
VFSEO = mean(signal_BW_SEO);
VFlog_SEO = mean(log(signal_BW_SEO));

% Get 'signal to noise' ratios
VFER(4) = sum(VFTKEO(1:5))/sum(VFTKEO(6:10));
VFER(5) = sum(VFSEO(1:5))/sum(VFSEO(6:10));
VFER(6) = sum(signal_BW_TKEO2(6:10))/sum(signal_BW_TKEO2(1:5));
VFER(7) = sum(VFlog_SEO(6:10))/sum(VFlog_SEO(1:5));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   Calculate the empirical mode descomposition excitation ratios
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [IMF] = IMF_measure(data)
IMF_dec = emd(data);
IMF_dec=IMF_dec';
IMF_dec2=log_bb(IMF_dec); %Log transformation
[N,M]=size(IMF_dec);

for i=1:M
    IMF_decEnergy(i) = abs(mean((IMF_dec(:,i)).^2));
    IMF_decTKEO(i) = abs(mean(TKEO(IMF_dec(:,i))));
    IMF_decEntropia(i) = abs(mean(-sum(IMF_dec(:,i).*log_bb(IMF_dec(:,i)))));
    IMF_decEnergy2(i) = abs(mean((IMF_dec2(:,i)).^2));
    IMF_decTKEO2(i) = abs(mean(TKEO(IMF_dec2(:,i))));
    IMF_decEntropia2(i) = abs(mean(-sum(IMF_dec2(:,i).*log_bb(IMF_dec2(:,i)))));
end

% Get 'signal to noise' ratio measures
IMF(1) = sum(IMF_decEnergy(4:end))/sum(IMF_decEnergy(1:3));
IMF(2) = sum(IMF_decTKEO(4:end))/sum(IMF_decTKEO(1:3));
IMF(3) = sum(IMF_decEntropia(4:end))/sum(IMF_decEntropia(1:3));
IMF(4) = abs(sum(IMF_decEnergy2(1:2))/sum(IMF_decEnergy2(4:end)));
IMF(5) = abs(sum(IMF_decTKEO2(1:2))/sum(IMF_decTKEO2(3:end)));
IMF(6) = sum(IMF_decEntropia2(1:2))/sum(IMF_decEntropia2(3:end));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      Calculate the TKEO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [energy] = TKEO(x)
data_length=length(x);
energy=zeros(data_length,1);
energy(1)=(x(1))^2; % first sample in the vector sequence
for n=2:data_length-1
```

```
   energy(n)=(x(n))^2-x(n-1)*x(n+1); % classical TKEO equation
end
energy(data_length)=(x(data_length))^2; % last sample in the vector sequence
end % end of TKEO function

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      Calculate the entropy of a discrete distribution
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function H = H_entropy(f)
% Usage: H = entropy(f)
N = length(f);
H = 0;
for j = 1:N
  H = H - f(j) * log_bb(f(j));
end
end % end of H_entropy function

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      Calculate the algorithm
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function pout = log_bb(pin, method)
% Function that computes the algorithm depending on the user specified
% base; if the input probability is zero it returns zero.

if nargin<2
   method = 'Nats';
end
switch (method)
   case 'Hartson' % using log10 for the entropy computation
       log_b=@log10;
   case 'Nats' % using ln (natural log) for the entropy computation
       log_b=@log;
   otherwise % method -> 'Bits'using log2 for the entropy computation
       log_b=@log2;
end
if pin==0
    pout=0;
else
    pout=log_b(pin);
end
end
```

ANEXO D

SCRIPT *melcepst.m*

```
function [c,tc]=melcepst(s,fs,w,nc,p,n,inc,fl,fh)
if nargin<2 fs=11025; end
if nargin<3 w='M'; end
if nargin<4 nc=12; end
if nargin<5 p=floor(3*log(fs)); end
if nargin<6 n=pow2(floor(log2(0.03*fs))); end
if nargin<9
   fh=0.5;
  if nargin<8
     fl=0;
     if nargin<7
        inc=floor(n/2);
     end
end
end
if isempty(w)
  w='M';
end
if any(w=='R')
  [z,tc]=enframe(s,n,inc);
elseif any (w=='N')
%    [z,tc]=enframe(s,hanning(n),inc);
  [z,tc]=enframe(s,0.5-0.5*cos(2*pi*(1:n)'/(n+1)),inc);%Hanning window
else
%    [z,tc]=enframe(s,hamming(n),inc);
  [z,tc]=enframe(s,0.54-0.46*cos(2*pi*(0:n-1)'/(n-1)),inc);%Hamming window
end
f=rfft(z.');
[m,a,b]=melbankm(p,n,fs,fl,fh,w);
pw=f(a:b,:).*conj(f(a:b,:));
pth=max(pw(:))*1E-20;
if any(w=='p')
  y=log(max(m*pw,pth));
else
  ath=sqrt(pth);
  y=log(max(m*abs(f(a:b,:)),ath));
end
c=rdct(y).';
nf=size(c,1);
nc=nc+1;
if p>nc
  c(:,nc+1:end)=[];
elseif p<nc
  c=[c zeros(nf,nc-p)];
end
if ~any(w=='0')
  c(:,1)=[];
  nc=nc-1;
end
if any(w=='E')
  c=[log(max(sum(pw),pth)).' c];
  nc=nc+1;
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
      calculate derivative
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if any(w=='D')
vf=(4:-1:-4)/60;
af=(1:-1:-1)/2;
ww=ones(5,1);
cx=[c(ww,:); c; c(nf*ww,:)];
vx=reshape(filter(vf,1,cx(:)),nf+10,nc);
vx(1:8,:)=[];
ax=reshape(filter(af,1,vx(:)),nf+2,nc);
ax(1:2,:)=[];
vx([1 nf+2],:)=[];
if any(w=='d')
   c=[c vx ax];
else
   c=[c ax];
end
elseif any(w=='d')
vf=(4:-1:-4)/60;
ww=ones(4,1);
cx=[c(ww,:); c; c(nf*ww,:)];
vx=reshape(filter(vf,1,cx(:)),nf+8,nc);
vx(1:8,:)=[];
c=[c vx];
end
```

ANEXO E

SCRIPT *wavedec_features.m*

```
function [features, feature_names] = wavedec_features(data, wname, dec_levels)
if nargin<3 || isempty(dec_levels);
    dec_levels = 10; % the user can use more levels; I used 10 in the NOLTA paper
end
if nargin<2 || isempty(wname);
    wname = 'db8'; % experiment with the wavelet base
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
               Wavelet decomposition
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[C, L] = wavedec(data, dec_levels, wname); % compute decomposition
[F0c.Ea, F0c.Ed] = wenergy(C, L);
for k = 1:dec_levels
    d = detcoef(C,L,k); % Detail coefficients in levels 1...dec_levels
    F0c.det_entropy_shannon(k) = wentropy(d, 'shannon');
    F0c.det_entropy_log(k) = wentropy(d, 'log energy');
    F0c.det_TKEO_mean(k) = mean(TKEO(d));
    F0c.det_TKEO_std(k) = std(TKEO(d));
end
for k = 1:dec_levels
    a = appcoef(C,L,wname, k); %Approximation coefficients in levels 1...dec_levels
    F0c.app_entropy_shannon(k) = wentropy(a, 'shannon');
    F0c.app_entropy_log(k) = wentropy(a, 'log energy');
    F0c.app_det_TKEO_mean(k) = mean(TKEO(a));
    F0c.app_TKEO_std(k) = std(TKEO(a));
end
Ldata = log(data);
clear C L d a
[C, L] = wavedec(Ldata, dec_levels, wname); % compute decomposition
[F0c.Ea2, F0c.Ed2] = wenergy(C, L);
for k = 1:dec_levels
    d = detcoef(C,L,k);
    F0c.det_LT_entropy_shannon(k) = wentropy(d, 'shannon');
    F0c.det_LT_entropy_log(k) = wentropy(d, 'log energy');
    F0c.det_LT_TKEO_mean(k) = mean(TKEO(d));
    F0c.det_LT_TKEO_std(k) = std(TKEO(d));
end
for k = 1:dec_levels
    a = appcoef(C,L,wname,k);
    F0c.app_LT_entropy_shannon(k) = wentropy(a, 'shannon');
    F0c.app_LT_entropy_log(k) = wentropy(a, 'log energy');
    F0c.app_LT_TKEO_mean(k) = mean(TKEO(a));
    F0c.app_LT_TKEO_std(k) = std(TKEO(a));
end
```

ANEXO F

SCRIPT *fastfda.m*

```
function [alpha, intervals, flucts] = fastdfa(x, varargin)
[xpts, ypts] = fastdfa_core(x, varargin{:});
% Sort the intervals, and produce a log-log straight line fit
datapts   = sortrows([xpts ypts],1);
intervals = datapts(:,1);
flucts    = datapts(:,2);
coeffs    = polyfit(log10(xpts), log10(ypts), 1);
alpha     = coeffs(1);
end
```

ANEXO G

SCRIPT *rpde.m*

```
function [H_norm, rpd] = rpde(x, m, tau, epsilon, T_max)
if ((nargin < 3) | (nargin > 5))
   help rpde;
   return;
end
if (nargin < 4)
   epsilon = 0.12;
end
if (nargin < 5)
    T_max = -1;
end
rpd = close_ret(x, m, tau, epsilon);
if (T_max > -1)
    rpd = rpd(1:T_max);
end
rpd = rpd/sum(rpd);
N = length(rpd);
H = 0;
for j = 1:N
  H = H - rpd(j) * logz(rpd(j));
end
H_norm = H/log(N);

function y = logz(x)
if (x > 0)
  y = log(x);
else
  y = 0;
end
```

ANEXO H

Script *depuration.m*

```
function tabcar = depuration(data)
%EJEMPLO: tabcar = depuration('a.wav')
[pathstr, name, ext] = fileparts(data);
[x,fs]=audioread(data);
%TAMAÑO DE LA VENTANA Y DE LA SUPERPOSICIÓN
win = 0.04;
step = 0.02;
%CÁLCULO DE LA ENERGÍA
Eor = ShortTimeEnergy(x, win*fs, step*fs);
Eor = Eor / max(Eor);
%UMBRALES DE SILENCIO Y SATURACIÓN
T_E = 0.05;
T_S = 0.95;
% GRÁFICOS DE LA ENERGÍA Y UMBRALES
clf;
subplot(2,1,1);
time = 1/fs:1/fs:length(x)/fs;
plot(time, x,'b');
title('SEÑAL ORIGINAL');
axis([0 length(x)/fs+0.2 min(x) max(x)]);
subplot(2,1,2);
plot(Eor,'r'); title('ENERGÍA DE LA SEÑAL'); hold on; legend('Short time
energy');
L = line([0 length(Eor)],[T_E T_E]); set(L,'Color',[0 0 0]); set(L,
'LineWidth', 2);
axis([0 length(Eor) min(Eor) max(Eor)]);
L = line([0 length(Eor)],[T_S T_S]); set(L,'Color',[0 0 0]); set(L,
'LineWidth', 2);
axis([0 length(Eor) min(Eor) max(Eor)]);
%
Flags1 = (Eor>=T_E);%obtenemos 1 si la energia es mayor que el umbral o 0
si la energia es menor que el umbral
Flags2 = (Eor<=T_S);%obtenemos 1 si la energia es menor que el umbral o 0
si la energia es mayor que el umbral
flags = Flags1 & Flags2; %deben cumplirse las dos

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     DETECCIÓN DE LÍMITES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
count = 1;
WIN = 1.5; %para mayor precision de los limites
Limits = [];
while (count < length(flags))
    % inicialización:
    countTemp = 1;
    while ((flags(count)==1) && (count < length(flags)))
        if (countTemp==1)
            Limit1 = round((count-WIN)*step*fs);
            if (Limit1<1)   Limit1 = 1; end
        end
        count = count + 1;
        countTemp = countTemp + 1;
    end
```

```
    if (countTemp>1) % if at least one segment has been found in the
current loop:
        Limit2 = round((count-1)*step*fs);      % set end counter
        if (Limit2>length(x))
            Limit2 = length(x);
        end

        Limits(end+1, 1) = Limit1;
        Limits(end,   2) = Limit2;
    end
    count = count + 1; % increase overall counter
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%         ENVENTANAR
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
f={};
signal={};
medidas={};
cont=0;
 for i=1:size(Limits,1)
    tst(i)=Limits(i,2)-Limits(i,1);
    if tst(i) >= 8000
      cont=cont+1;
      signal{i} = x(Limits(i,1):Limits(i,2)); %si es mayor a 8000 tomamos
de la señal x ese segmento
      C{i} = pruebaframe(signal{i}); % enventana y ya me retorna las
caracteristicas
    end
 end
 if cont == 0
    fprintf('Audio no valido\n');
    for j=1:1
  nombre{j}=name;
 end
nombre;
    tabcar=table(nombre,NaN*ones(1,340));
 else
 C;
 caracteris=vertcat(C{:});
 d=size(caracteris,1);
 nombre=cell(d,1);
 for j=1:d
  nombre{j}=name;
 end
nombre;
tabcar=table(nombre,caracteris);
end
```

ANEXO I

Script *pruebaframe.m*

```
function C=pruebaframe(data1)
% [data1,fs]=audioread('pva_0007588.wav');
fs=8000;
f={};
medidas={};
news={};
v={};
tiempo={};
ok={};
for i=1:6
    n{i}=(((2^i)*fs)/2); % obtenemos 8000, 16000, 32000, 64000, etc
    f{i}=(enframe(data1,hamming(n{i},'periodic'),0.1*n{i}))'; %e
    c{i}=size(f{i},2); % me da el numero de filas y de columnas que tiene
la matriz f{i}(enventanado)
    if c{i} > 0
     v=cell(c{i},1); %ojo para el almacenamiento
     medidas=cell(c{i},1);
     seg=cell(c{i},1);
     VF_close=cell(c{i},1);
     ser=cell(c{i},1);
    for j=1:c{i} %el numero de columnas es decir cuantos pedazos de 8000
salieron
        v{j}=f{i}(:,j);%ok
%       ok{i}=v;
        VF_close{j}=dypsa(v{j},fs);
        tseries=cell(length(VF_close{j})-1,1);
        cont=0;
             for w=1:length(VF_close{j})-1
             tseries{w} = v{j}(VF_close{j}(w):VF_close{j}(w+1));
             z{w}=length(tseries{w});
                 if z{w} > 40
                  cont =cont+1;
                 end
             end
         if cont>=1
             medidas{j}=voice_analysis(v{j},fs);
             ok{i}=medidas;
             seg{j}=n{i}/8000;
             tiempo{i}=seg;
         else
            medidas{j}=NaN*ones(1,339);
            ok{i}=medidas;
            seg{j}=n{i}/8000;
            tiempo{i}=seg;
         end
    end
  end
  ok;
  news=vertcat(ok{:});
  caracteristicas=cell2mat(news);
  tiempo;
  segundos=vertcat(tiempo{:});
  segun=cell2mat(segundos);
  C=horzcat(segun,caracteristicas);
end
end
```

ORDEN DE EMPASTADO