



# **UNIVERSIDAD NACIONAL “PEDRO RUIZ GALLO”**



## **FACULTAD DE INGENIERIA CIVIL, SISTEMAS Y ARQUITECTURA**

*Escuela Profesional de Ingeniería de Sistemas*

---

### **TESIS**

Para obtener el título Profesional de:

Ingeniera de Sistemas

### **TÍTULO**

MODELOS OCULTOS DE MARKOV PARA EL DESARROLLO DE UN  
SISTEMA DE AYUDA AL HABLA PARA PERSONAS QUE SUFREN  
DE DISARTRIA

### **PRESENTADO POR:**

GUEVARA VELEZ, CLARA MERCEDES

### **ASESOR:**

Mg. Ing. VILLEGAS CUBAS JUAN ELIAS

**Lambayeque, ABRIL 2018**

## **TESIS**

# **“MODELOS OCULTOS DE MARKOV PARA EL DESARROLLO DE UN SISTEMA DE AYUDA AL HABLA PARA PERSONAS QUE SUFREN DE DISARTRIA”**

PRESENTADA POR:

---

Bach. Clara Mercedes Guevara Velez

AUTOR

---

Mg. Ing. Juan Elias Villegas Cubas

PATROCINADOR

APROBADO POR:

---

M. Sc. Ing. Gilberto Martin Ampuero Pasco

PRESIDENTE DEL JURADO

---

Ing. Omar Wilton Saavedra Salazar

SECRETARIO DEL JURADO

---

Mg.Ing. Jesus Bernardo Olavarria Paz

VOCAL DEL JURADO

## **DEDICATORIA**

***A Dios, por darme siempre la  
fuerza para seguir adelante,  
iluminándome y guiándome por el  
camino del bien.***

***A mi padre Henry, por ser el pilar  
más importante en nuestra  
familia, por darme siempre su  
apoyo para cumplir mis metas  
profesionales.***

***A mi madre Mercedes, por su  
entrega, dedicación, apoyo y  
sobre todo amor y cariño en lo  
bueno y malo.***

***A mis hermanos, Sara y Henry,  
por todo su cariño desde siempre.***

## **AGRADECIMIENTO**

***A mi abuelita Columba, por  
regalarme lo mejor de ella a pesar  
de su enfermedad y por ser la  
fuente de inspiración de este  
trabajo de Tesis.***

***A mi asesor y docente, Ing. Juan  
Villegas, por sus enseñanzas y  
guía para la realización de este  
proyecto.***

***A los miembros del jurado por  
permitir que se haga realidad el  
presente proyecto.***

## **Resumen**

La voz o el habla es una de las formas básicas de intercambiar información entre los seres humanos. Daños neuronales ocasionados por un infarto, una embolia o trauma cerebral, pueden afectar a la voz del individuo, alterando su articulación, resonancia y respiración. A este cuadro patológico de la voz se le conoce como disartria.

Investigaciones realizadas en el extranjero han demostrado los beneficios de la tecnología computacional para la comunicación y asistencia de personas con disartria, especialmente de Sistemas de Reconocimiento Automático del Habla (SRAH). Sin embargo, no hay desarrollo de dicha tecnología que aborde el tema de la disartria, especialmente, no hay desarrollo de dicha tecnología que aborde el tema de la disartria en Lambayeque, y particularmente, de personas cuya lengua materna sea el español peruano.

El diseño de un sistema de RAH para voz disártrica implicar solucionar los siguientes problemas: (1) tasas variables de precisión de reconocimiento de voz (25-95%) para usuario con disartria; (2) los síntomas asociados a la disartria dificultan la recopilación de muestras de voz (corpus) para un entrenamiento supervisado robusto del sistema.

La presente tesis describe el diseño y desarrollo de un reconocedor automático de voz disártrica en español y una interfaz gráfica para su uso, el sistema en su totalidad será denominados “*SISTEMA DE AYUDA AL HABLA (SAH)*”. Esto para realizar funciones de apoyo en el nivel de comunicación de

personas con el trastorno de disartria. La metodología de desarrollo abordó los problemas mencionados anteriormente, y como solución se propuso el desarrollo del sistema bajo la perspectiva de Dependencia del Usuario.

Para la función de Reconocimiento del sistema se utilizó la biblioteca HTL toolkit, siendo la técnica de modelado acústico los Modelos Ocultos de Markov (HMM). En las pruebas con un vocabulario de 5 palabras, se tuvo un aumento del nivel de comunicación de un 46,67%, y una tasa de reconocimiento de voz de 65%. Estos resultados fueron comparándose la precisión del SAH a la precisión del reconocimiento humano.

▪



# INDICE

<b>Resumen.....</b>	<b>7</b>
<b>Introducción .....</b>	<b>14</b>
<b>CAPITULO I: MARCO LÓGICO .....</b>	<b>17</b>
1.1    Situación Problemática .....	17
1.2    Formulación del problema .....	18
1.3    Hipótesis .....	18
1.4    Limitaciones de la investigación .....	18
1.5    Justificación e importancia de la investigación.....	19
1.6    Objetivos de la investigación .....	19
1.5.1    Objetivo general.....	19
1.5.2    Objetivos específicos.....	19
<b>CAPITULO II: MARCO TEÓRICO .....</b>	<b>22</b>
2.1    Antecedentes de la investigación .....	22
2.1.1    Investigaciones sobre Reconocimiento Automático de Voz .....	22
2.1.2    Investigaciones sobre Reconocimiento de Voz en personas que sufren de Disartria .....	22
2.2    Bases Teóricas .....	24
2.2.1    Sistema de Reconocimiento Automático del Habla (SRAH).....	24
2.2.2    Modelos Ocultos de Markov (HMM) .....	27

2.2.3	Algoritmos HMM .....	28
2.2.4	Biblioteca HTK Toolkit .....	30
2.2.5	Disartria .....	36
2.2.6	Comunicación Oral .....	39
2.2.7	Vocabulario .....	41
<b>CAPITULO III: MARCO METODOLÓGICO .....</b>		<b>44</b>
3.1	Tipo y Diseño del Estudio .....	44
3.1.1	Tipo de la Investigación .....	44
3.1.2	Población y Muestra .....	44
3.1.3	Operacionalización de Variables .....	45
3.2	Metodología de Recolección de Datos .....	46
3.2.1	Técnica e Instrumento de Recolección de Datos .....	46
3.2.2	Diseño de Contrastación de Hipótesis .....	47
<b>CAPITULO IV: DESARROLLO DEL RECONOCEDOR DEL HABLA ....</b>		<b>49</b>
4.1	Requisitos y Preparación .....	50
4.1.1	Requisitos Previos .....	50
4.1.2	Preparación del entorno de trabajo .....	50
4.2	Diccionario fonético equilibrado .....	51
4.2.1	Archivo prompts.txt .....	51
4.2.2	Archivo wlist .....	52

4.3	Grabación de los datos .....	52
4.4	Creación de los archivos de transcripción .....	52
4.5	Codificación de los audios .....	52
4.5.1	Archivo codetrain.scp .....	54
4.5.2	Archivo de Configuración .....	54
4.6	Entrenamiento Supervisado de los Modelos Acústicos .....	55
4.7	Definición de la gramática.....	64
4.7.1	Gramática .....	64
4.7.2	Diccionario .....	65
4.7.3.	Red Gramática.....	65
4.8	Reconocimiento .....	66
<b>CAPITULO V: DESARROLLO DE LA INTERFAZ.....</b>		<b>69</b>
5.1	Diagrama de Flujo.....	69
5.1.1	Archivos globales del SAH .....	69
5.2	Arquitectura.....	70
5.3	Interfaz Gráfica.....	73
5.3.1	Etapas de Grabación.....	73
5.3.2.	Etapas de Reconocimiento .....	74
5.3.2	Etapas de Síntesis.....	74
5.3.3	Implementación .....	74

<b>CAPITULO VI: RESULTADOS</b>	<b>79</b>
6.1 Participantes	79
6.1.1 Pacientes Disártrico	79
6.1.2 Participantes Oyente	81
6.2 FASE I: ANTES DE SAH	81
6.2.1 Descripción del Escenario	81
6.2.2 Pruebas	82
6.2.3 Observaciones	84
6.3 FASE II: DESPUÉS DEL SAH	85
6.3.1 Descripción del Escenario	85
6.3.2 Pruebas	85
6.3.3 Observaciones	88
6.4 Contratación de Fases	88
6.5 Tasa de Reconocimiento del Sistema	89
<b>CAPITULO VII: CONCLUSIONES</b>	<b>91</b>
<b>BIBLIOGRAFÍA</b>	<b>94</b>
<b>ANEXO A: Archivo prompts.txt</b>	<b>98</b>
<b>ANEXO B: Archivo prompts2wlist.pl</b>	<b>100</b>
<b>ANEXO C: Archivo wlist</b>	<b>101</b>
<b>ANEXO D: Archivo prompts2mlf.pl</b>	<b>102</b>
<b>ANEXO E: Archivo words.mlf</b>	<b>103</b>

<b>ANEXO F: Archivo codetrain.scp.....</b>	<b>104</b>
<b>ANEXO G: Archivo analisis.conf.....</b>	<b>105</b>
<b>ANEXO H: Archivo hmmdefs.....</b>	<b>106</b>
<b>ANEXO I: Archivo vFloors .....</b>	<b>107</b>
<b>ANEXO J: Archivo config2.conf.....</b>	<b>108</b>

## **Introducción**

Dentro del campo de la Inteligencia Artificial en Ingeniería de Sistemas, se busca desarrollar una aplicación directa para personas con discapacidades, en especial del habla. De acuerdo, a las estadísticas mostradas por el MINSA, en el año 2016, aproximadamente el total de personas con enfermedades y trastornos que ocasionaron deficiencias son 117 593 personas, de las cuales, un 6% tienen trastornos de la comunicación, en números esto se traduce a 7 062 peruanos registrados en el MINSA con problemas de comunicación

Dentro de las discapacidades del habla y comunicación se considera a la disartria, que se puede definir como el trastorno de la expresión verbal causado por una alteración del control muscular del mecanismo del habla, siendo este un problema del habla y no un problema del lenguaje.

Se obtuvo toda la información posible de bibliografías y linkografías, para conocer acerca de las patologías de lenguaje y motoras en general de las personas con disartria. Se reconoció la utilidad práctica de sistemas computacionales basados en inteligencia artificial que ayudan a los pacientes a comunicarse por medio de la voz en otros países.

Al tener dicha información se realizó investigación en el campo de aplicaciones de RAH, encontrándose proyecto de otros países enfocados en el desarrollo de sistemas para mejorar la comunicación de personas con disartria. Esto llevó a identificar los siguientes problemas relacionados con el desarrollo de un SRAH para voz disártrica:

- Tasas Variables de precisión en el reconocimiento de voz (25-95%) para usuarios disártricos.
- El rango de anormalidades en la voz disártrica es muy amplio, variando entre personas afectadas.

- Conforme el tamaño del vocabulario del sistema aumenta, el nivel de precisión de reconocimiento disminuye.
- Los síntomas asociados a la disartria dificultan la recopilación de muestras de voz (corpus) para un entrenamiento supervisado robusto del sistema
- No hay un corpus de voz disártrica en español latino para la realización de análisis o modelado acústico para la construcción de un SRAH.
- No hay proyecto en SRAH para el idioma español similares que sirvan como base de comparación, la mayoría están desarrollados para el idioma inglés.

La presente investigación pretende desarrollar un sistema de reconocimiento automático de voz disártrica denominado Sistema de Ayuda al Habla (SAH), superando dichos problemas mencionados anteriormente para lograr el aumento del nivel de la comunicación por medio de voz de las personas que sufren del trastorno de disartria.

# **CAPITULO I:** **MARCO LÓGICO**



## CAPITULO I: MARCO LÓGICO

### 1.1 Situación Problemática

(Duffy, 2012) señaló que, la disartria es un tipo de desorden neurológico que daña el control de las articulaciones motoras del habla. Este tipo de desorden es causado por distrofia muscular, daño cerebral tumores, enfermedad de Parkinson, esclerosis múltiple. Algunas de las características del habla disártrica incluyen voz hinchada, nasal o congestionada, monótona, articulación imprecisa, reducción o eliminación de consonantes, habla intermitente o murmurado, ritmo más lento o rápido. (Bunton, 2001; Kent, Vorperian, Kent, & Duffy, 2003; Ramig, 1992) Indicaron que, debido a estos problemas articulatoria, no hay consistencia en la pronunciación. Todas estas características decrecen la inteligibilidad del hablante con Disartria (que tan bien el discurso del hablante es entendido por otros) y reducen las interacciones verbales que disminuyen seriamente su calidad de vida.

En el Perú, según MINSA, existen aproximadamente 7062 personas con trastornos que producen deficiencia en la comunicación.

<i>Enfermedades y Trastornos que ocasionaron Deficiencias</i>	<i>Total</i>	<i>%</i>
- EN LAS FUNCIONES NEURO- MUSCULOESQUELETICAS Y RELACIONADAS CON EL MOVIMIENTO(incluye amputaciones congénitas)	86,703	73.7
- EN LAS FUNCIONES MENTALES SUPERIORES	11,303	9.%
- EN LA COMUNICACION	7,062	6.%
- DEFICIENCIAS SENSORIALES	833	0.%
- ANOMALIAS CONGENITAS	954	0.%
- OTRAS ENFERMEDADES O TRASTORNOS QUE OCASIONARON DEFICIENCIAS:	1,702	1.4
- OTROS DIAGNOSTICOS	2,156	1.8
- ATENCION PARA DESCARTE	1,229	1.0
- DIAGNOSTICOS DEFICIENTEMENTE INFORMADOS	5,651	4.8
<i>Total General:</i>	117,593	100.0

*Figura 1: Distribución de las principales enfermedades y trastornos que ocasionaron deficiencias en los pacientes atendidos en los servicios de Rehabilitación Perú 2016.*

*Fuente: HIS-DIS / MINSA: OEI-OEE-INR*

Según (“Lambayeque,” 2015), en el hospital de EsSalud Almanzor Aguinaga de Chiclayo, en la región Lambayeque, la jefa del servicio de Medicina Física y Rehabilitación, puntualizó que se ha realizado un total de 6259 procedimientos de terapia de Lenguaje para pacientes disártricos en el último año.

Una consecuencia directa de este trastorno son las dificultades que genera para hacerse entender mediante el habla y estos disminuye su capacidad de comunicación eficaz, como consecuencia de dichos problemas de comunicación pueden verse alteradas las relaciones sociales. En muchos casos esta afección puede llegar a provocar depresión y aislamiento social.

## **1.2 Formulación del problema**

¿Cómo mejorar el inadecuado nivel de comunicación entre una persona Disártrica y una persona sin discapacidad?

## **1.3 Hipótesis**

El “Sistema de Ayuda al Habla para personas con problemas de Disartria” mejorará el inadecuado nivel de comunicación entre una persona disártrica y una persona sin discapacidad

## **1.4 Limitaciones de la investigación**

El sistema desarrollado en el presente trabajo de tesis está enmarcado bajo las siguientes limitaciones:

- Reconocimiento de palabras aisladas.
- Reconocimiento de palabras de un locutor único.
- Reconocimiento de palabras está definido por comparación contra modelos ocultos de Markov previamente entrenados.
- Reconocimiento de palabras está restringido a la cantidad de palabras que se encuentran el diccionario del sistema.

## **1.5 Justificación e importancia de la investigación**

Disartria es una discapacidad neurológica que afecta el control de los articuladores motores del habla; este impedimento es causado por la falta de control sobre los músculos relacionados al habla, falta de coordinación entre ellos, o parálisis de los mismos. Es frecuentemente asociado con fonación y amplitud irregular, que resulta en señales de voz comprometidas y reducida inteligibilidad del habla, lo que conlleva a un bajo nivel de comunicación con otras personas. Como las personas discapacitadas son frecuentemente impedidas físicamente e incapaces de usar un teclado, el Reconocimiento de Voz Automático (ASR) puede ser muy útil para personas con problemas de disartria. Este sistema identifica la palabra dicha representada como señales acústicas. Los sistemas ASR tienen aplicaciones en muchos dominios incluyendo salud, servicio militar, telefonía y otros.

Sistemas ASR que son diseñados para hablantes normales usualmente proveen un bajo rendimiento para individuos que sufren de disartria, ya que el habla de ambos es totalmente diferente. Es por ello, que es necesario proponer sistemas ASR específicamente contruidos para usuarios con disartria que entregue la adecuada exactitud.

## **1.6 Objetivos de la investigación**

### **1.5.1 Objetivo general**

Desarrollar un Sistema de Ayuda al Habla para mejorar el nivel de comunicación entre una persona disártrica y una persona sin discapacidad.

### **1.5.2 Objetivos específicos**

- a. Identificar las dificultades de comunicación de personas con problemas de disartria.
- b. Describir los componentes y funciones del Sistema de Ayuda al Habla para personas con problemas de Disartria.

- c. Evaluar la eficacia del sistema desarrollado con una persona disártrica.
- d. Evaluar el nivel de comunicación de la persona disártrica con una persona sin discapacidad.

## **CAPITULO II: MARCO TEÓRICO**

## **CAPITULO II: MARCO TEÓRICO**

### **2.1 Antecedentes de la investigación**

#### **2.1.1 Investigaciones sobre Reconocimiento Automático de Voz**

(Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. r, Jaitly, N., Kingsbury, B, 2012) proponen una forma alternativa de evaluar el ajuste utilizando una red neuronal de alimentación hacia adelante, que toma varios marcos de coeficientes como entrada y produce probabilidades posteriores sobre estados de los modelos de Markov como salida. El estudio muestra que reemplazando el uso de los Modelos de Mezcla Gaussiana por Redes Neuronales Profundas en el modelado acústico para el reconocimiento de voz se pueden lograr significativas ganancias.

(Samira, Fateh, Smaine, & Mohamed, 2013) proponen un nuevo enfoque de reconocimiento de voz basado en el modelado de múltiple Modelos Ocultos de Markov. A diferencia de los tradicionales enfoques Markovianos donde un solo modelo está asociado a todas las palabras, en este enfoque varios modelos, provenientes de diferentes inicializaciones aleatorias, son combinados para cada palabra. Los resultados experimentales con un conjunto de datos de dígitos numéricos Árabicos mostraron la eficiencia de este enfoque, cuando se comparó con un enfoque clásico basado en un solo modelado (se incrementó la exactitud del reconocimiento en más del 9%).

#### **2.1.2 Investigaciones sobre Reconocimiento de Voz en personas que sufren de Disartria**

(Kadi, Selouani, Boudraa, & Boudraa, 2016) nos presentan nuevos enfoques para mejorar el análisis y clasificación del habla desordenada. Primero, proponen un enfoque de reconocimiento automático del hablante especialmente adaptado en identificar hablantes disártricos. Luego, sugieren un método para la evaluación automática de los niveles de severidad de disartria. Para este propósito, se presenta un modelo de simulación de las

partes externas, media e internas del oído. Este modelo proporciona pistas relevantes basadas en la audición que se combinan con los coeficientes Cepstral de la Frecuencia Mel (MFCC) para representar emisiones de habla atípicas. Los experimentos son llevados a cabo mediante el uso de las bases de datos de Nemours y Torgo. Se prueban modelos Gaussianos de Mezcla (GMMs), Máquinas de Vector de Soporte (SVMs) y sistemas GMM/SVM híbridos, y son comparados en el contexto de la identificación y evaluación de disártricos. Los resultados lograron una tasa de identificación correcta del hablante del 97.2%, la cual puede ser considerada prometedora para este nuevo enfoque. También se superaron los sistemas de evaluación ya existentes con una tasa de clasificación correcta de 93.2% de los niveles de severidad de la disartria.

(Chandrakala & Rajeswari, 2016) se enfocan en una representación de aprendizaje robusta para el reconocimiento de habla disártrica que involucra el reconocimiento secuencial de patrones de enunciados de distintas longitudes. Se propone un marco híbrido que usa ejemplos Específicos de los Modelos Ocultos de Markov (ESHMMs) para obtener las puntuaciones de probabilidad logarítmica para un enunciado de habla disártrica para formar una representación vectorial de puntuaciones de dimensión fija. Esta representación es usada como una entrada para el clasificador discriminativo tal como una Máquina de Vector de Soporte. Este enfoque propuesto es evaluado usando la base de datos UA-Speech. La exactitud del reconocimiento es mucho mejor que el enfoque convencional basado en los Modelos Ocultos de Markov y que el modelo híbrido de Red Neuronal Profunda- Modelos Ocultos de Markov (DNN-HMM). La eficiencia de la naturaleza discriminativa de la representación vectorial de puntuaciones es probada con palabras de “muy baja” inteligibilidad.

## 2.2 Bases Teóricas

### 2.2.1 Sistema de Reconocimiento Automático del Habla (SRAH)

(Jurafsky & Martin, 2008) nos dice que en los orígenes de las tecnologías del habla se encuentra el interés científico por conseguir que un ordenador entendiera la voz humana y fuera capaz de convertir a texto las locuciones dictadas o de realizar acciones en consonancia con la interpretación de dichas locuciones. Es lo que denominamos reconocimiento automático del habla.

De manera general un SRAH se puede catalogar como:

- Dependiente del usuario (DU): Aplicación sólo para un usuario. Se construye tomando en cuenta sus características acústicas particulares, es un sistema personalizado.
- Independiente del Usuario (IU): Aplicación para más de un usuario. Se construye tomando en cuenta las características de muchos usuarios y después se personalizan mediante técnicas de adaptación de usuario.

Según (Álvarez, 2001), el esquema general que debe seguir un sistema automático de reconocimiento del habla como el nuestro comprende las siguientes etapas:

- **Representación de la señal:** Tras obtener los datos de entrada al sistema por medio de la digitalización de la señal de voz, las muestras del paso anterior se transforman en un conjunto de medidas o características útiles. La realización de esta tarea se lleva a cabo a un ritmo constante, usualmente de entre 10 y 20 ms. Como producto de salida se obtienen una serie de vectores que de alguna manera contienen en esencia la señal original, pero con un tamaño mucho menor. Un vector de características deberá contener toda la información relevante para realizar el proceso de reconocimiento, eliminando el resto de los contenidos redundantes o que no sean útiles para el sistema.



- **Búsqueda de patrones:** En la fase de búsqueda de patrones las medidas de la etapa anterior se emplean para encontrar la palabra candidata que proporcione el mejor encaje. Para ello se tienen en cuenta las restricciones impuestas por los modelos acústico, léxico y de lengua (gramática) disponibles.
- **Modelado y clasificación:** Los sistemas de reconocimiento intentan estimar y configurar las fuentes de variación descritas anteriormente de diversas formas en lo que constituye la fase de modelado y clasificación. En el nivel de representación de la señal, los desarrolladores emplean representaciones que enfatizan las características importantes desde un punto de vista perceptual; por otra parte, se intenta des enfatizar aquellas otras que son dependientes de determinados hablantes. La adaptación a un determinado hablante o a las condiciones del entorno de funcionamiento del sistema se realiza a través de un conjunto de procesos de normalización. La normalización tiene como finalidad restaurar las características del vector, es decir, conseguir medidas con valores lo más próximos posibles a los que se obtendrían en condiciones neutras o a las presentes cuando se entrenó el reconocedor.
- **Proceso de entrenamiento:** El entrenamiento constituye una etapa previa a la entrada en funcionamiento del conjunto y tiene como finalidad establecer los diferentes conjuntos de modelos empleados durante el proceso de búsqueda:
  - Modelos acústicos, que recojan realizaciones dependientes del género del locutor, entonaciones, variantes dialectales, etc.
  - Modelos léxicos, que contemplen pronunciaciones alternativas de las palabras, con objeto de permitir que los algoritmos de búsqueda encuentren diferentes caminos.
  - Modelos de lengua, que permitan estimar la frecuencia de ocurrencia de determinadas secuencias de palabras.

Además, durante el entrenamiento se puede obtener también la información necesaria para ajustar los procesos de modelado y normalización.

En los sistemas de reconocimiento del habla basados en los modelos ocultos de Markov, se modela la evolución temporal de la secuencia de espectros obtenida de la señal de voz mediante un HMM con el fin de contemplar aleatoriamente las diversas fuentes de variabilidad de la señal. Este modelado consiste en la asociación de los estados del HMM a los diferentes tramos de la señal, de forma que las probabilidades de generación de observaciones modelan la variabilidad estadística de las características espectrales de cada tramo, mientras que las probabilidades de transición modelan su secuenciamiento y duración. En el siguiente gráfico se representa el proceso de reconocimiento mediante estos modelos:

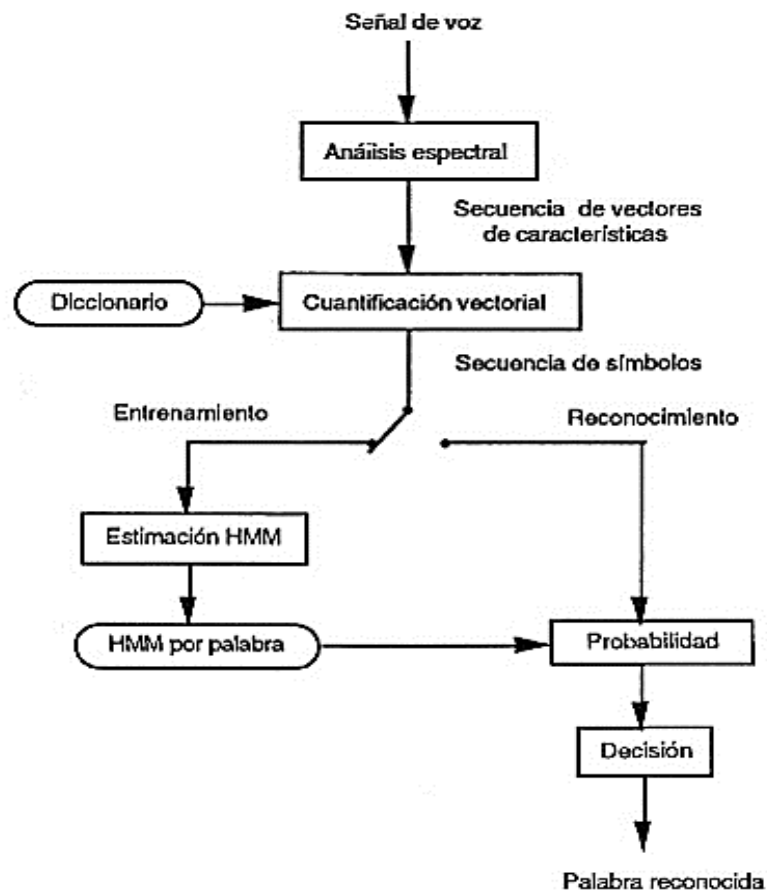


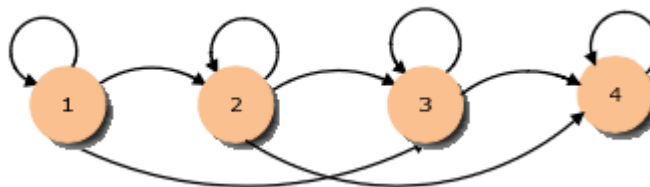
Figura 2. Proceso de Reconocimiento de palabra

### 2.2.2 Modelos Ocultos de Markov (HMM)

Los modelos ocultos de Markov (HMM), es un proceso estocástico que consta de un proceso de Markov no observado (oculto) y un proceso observado  $O$  cuyos estados son dependientes estocásticamente de los estados ocultos. La tarea fundamental consiste en determinar los parámetros ocultos a partir de los parámetros observados. La diferencia fundamental respecto a un modelo de Markov habitual consiste en que los estados no son directamente visibles para el observador, pero sí lo son las variables influenciadas por el estado. Cada estado tiene una distribución de probabilidad asociada sobre el conjunto de posibles valores de salida. La secuencia de valores de salida generados a partir de un HMM nos dará cierta información sobre la secuencia de estados. Una de las aplicaciones más utilizadas de estos modelos, es el reconocimiento del habla, esta técnica ha permitido modelar adecuadamente la gran variabilidad en el tiempo de la señal de voz

#### 2.2.2.1 Arquitectura de un HMM

Esta arquitectura se da de acuerdo con el número de estados que lo componen y las transiciones permitidas entre dichos estados. Se usará el modelo de Bakis, también se conoce como modelo izquierda-derecha, pues la secuencia de estados oculta tiene la propiedad de que, conforme el tiempo se incrementa, el estado se incrementa (o permanece constante). Se utiliza en el modelado del habla porque se ajusta a señales cuyas propiedades cambian a lo largo del tiempo, como la voz.



*Figura 3. Modelo de Bakis de 4 estados*

Cuando se aplican los modelos ocultos de Markov al reconocimiento de la voz:

- Los estados se interpretan como modelos acústicos, indicando las ocurrencias de sonidos que son más probables durante los correspondientes segmentos de habla.
- Las transiciones incorporan restricciones de tipo temporal acerca de cómo son las secuencias de aparición de esos sonidos. Estados y transiciones pueden ser utilizados para modelar distintas jerarquías del proceso del habla: desde fonemas hasta oraciones pasando por palabras

### 2.2.3 Algoritmos HMM

#### 2.2.3.1 Algoritmo de Viterbi

El algoritmo de Viterbi nos permite encontrar la secuencia de estados más probable en un Modelo Oculto de Markov (MOM) a partir de una observación, éste obtiene la secuencia óptima que mejor explica la secuencia de observaciones.

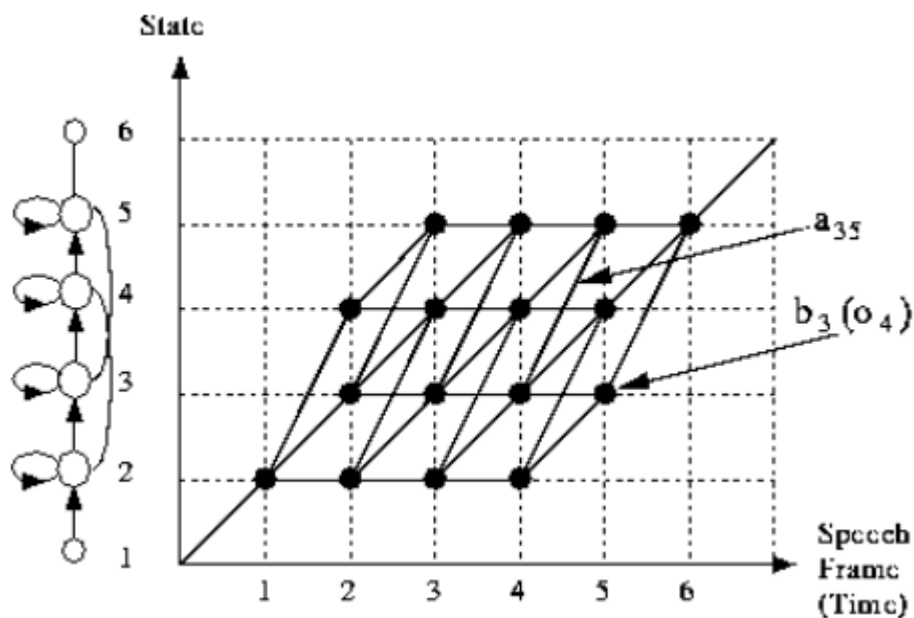


Figura 4. Algoritmo de Viterbi para reconocimiento de Palabra Aislada

Como se muestra en la figura 4, este algoritmo se puede visualizar como la mejor ruta a través de una matriz donde la dimensión vertical representa los estados de los HMM y la dimensión horizontal representa los marcos del habla (es decir, el tiempo). Cada punto grande en la imagen representa la probabilidad de registro de la observación de ese marco en ese momento y cada arco entre los puntos corresponde a una probabilidad de transición de registro. La probabilidad de registro de cualquier ruta se calcula simplemente sumando las probabilidades de transición de registro y las probabilidades de salida de registro a lo largo de esa ruta. Las rutas crecen de izquierda a derecha columna por columna.

#### **2.2.3.2 Algoritmo Baum Welch**

Se define como la reestimación de los parámetros de un HMM sobre la base de otro HMM. Uno de los problemas relacionados con los Modelos Ocultos de Márkov (MOM) es el de encontrar un modelo que maximice la probabilidad de una secuencia de observaciones, es decir, determinar el modelo que mejor explica tal secuencia. El problema es que no es posible encontrar tal modelo analíticamente y por ello es necesario un algoritmo iterativo como el de Baum y Welch, que permite estimar los parámetros de un modelo que hacen máxima la probabilidad de una secuencia de observables.

El funcionamiento del procedimiento iterativo es básicamente el siguiente:

- Se parte de un modelo inicial que se puede seleccionar aleatoriamente.
- Se realiza el cálculo de las transiciones y símbolos de emisión que son más probables según el modelo inicial escogido.
- Se construye un nuevo modelo en el que se incrementa la probabilidad de las transiciones y símbolos determinados en el paso anterior. Para la secuencia de observables en cuestión, el modelo tendrá ahora una probabilidad mayor que el modelo anterior.

Este proceso de entrenamiento se repite varias veces hasta que no exista mejora entre un modelo y el siguiente revisado.

#### 2.2.4 Biblioteca HTK Toolkit

(Young et al., 2002) El diseño de esta herramienta tiene el propósito principal de construir Modelos Ocultos de Markov para el procesamiento de voz, específicamente para el diseño de reconocedores de voz. Como primer paso las herramientas del HTK se utilizan para estimar los parámetros del conjunto de HMMs, utilizando archivos de entrenamiento que contienen la pronunciación y su transcripción asociada. Posteriormente, las pronunciaciones desconocidas se transcriben a través de las herramientas del HTK para su reconocimiento. La Figura 5., muestra las etapas de procesamiento para el diseño de Reconocedor de Voz.

La mayoría de las funcionalidades del HTK están construidas en módulos de librerías. Las herramientas del HTK están diseñadas para correr bajo el estilo de comandos en línea. Proporcionando el nombre de la herramienta, los archivos y argumentos opcionales; los valores opcionales están siempre separados del nombre de la opción por un espacio. Como ejemplo, consideremos la siguiente herramienta hipotética:

*HFcc -T 1 -f 34.3 -a -s archivo archivo1 archivo2*

Se tienen dos argumentos principales *archivo1* y *archivo2* y cuatro argumentos opcionales. Las opciones siempre se escriben con el nombre de la opción que corresponde a una letra seguida del valor de la opción. Por ejemplo, la opción *-f* es un número real, el valor de la opción *T* es un número entero y el valor *-s* es una cadena. En el caso de la opción *-a* no sigue ningún valor, ya que corresponde a una bandera para habilitar o deshabilitar alguna característica de la herramienta. Las opciones cuyo nombre es una letra mayúscula tienen el mismo significado para todas las herramientas. Por ejemplo, *-T* se utiliza para controlar la ubicación de salida de la herramienta

HTK. La operación de la herramienta se puede controlar por parámetros almacenados en un archivo de configuración, por ejemplo:

*HFcc -C config -f 34.3 -a -s archivo archivo1 archivo2*

En este caso a través de la opción *-C*, la herramienta hipotética *HFxx* cargará los parámetros que se encuentran en el archivo de configuración llamado *config* durante el procedimiento de inicialización.

#### **2.2.4.1 Preparación de datos**

Para construir el conjunto de HMMs, debemos tener un conjunto de archivos de datos (señales de voz) en el formato apropiado y su transcripción. Así mismo se deben tener los archivos etiquetados.

En caso de requerir realizar el grabado de los archivos de voz, se puede utilizar la herramienta *HSLab*.

La herramienta *HCopy* se utiliza para parametrizar los datos, se utiliza para copiar uno o más archivos fuentes en un archivo de salida (concatena archivos), colocando las variables de configuración apropiadas.

*HList* se puede usar para checar el contenido de cualquier archivo de voz y se puede usar para verificar los resultados de cualquier conversación antes de procesar una gran cantidad de datos.

La herramienta *HLed* es un editor de etiquetas diseñado para realizar las transformaciones requeridas de los archivos etiquetados. Esta herramienta puede colocar los archivos de salida en un solo Archivo de etiquetas maestro “*MLF*” (*Master Label File*), el cual es conveniente para los procesos posteriores.

#### 2.2.4.2 Herramientas de entrenamiento

El siguiente paso en la construcción del sistema consiste en definir la topología requerida para cada HMM, escribiendo una definición del prototipo. Se puede utilizar cualquier topología deseada, la definición del prototipo. Se puede utilizar cualquier topología deseada, la definición de HMM se puede almacenar en forma externa como un texto simple a través de un editor de texto. El propósito de la definición del prototipo es solo especificar las características generales de la topología del HMM. Los parámetros serán calculados posteriormente por las herramientas de entrenamiento. El proceso de entrenamiento se lleva a cabo en etapas. Primero, se debe crear un conjunto de modelos inicial.

La herramienta *HInit* y *HRest* proporcionan un entrenamiento del estilo de palabras aisladas. Cada uno de los HMMs requeridos se genera individualmente. *HInit* lee en todos los datos de entrenamiento y deja todos los ejemplos del fonio requerido. Después calcula iterativamente un conjunto inicial de valores de parámetros utilizando el procedimiento *k-media* segmentada. En el primer ciclo, los datos de entrenamiento están segmentados uniformemente, cada estado del modelo se empata con el correspondiente segmento de datos y se estima la media y la varianza. En el segundo ciclo, así como en los sucesivos, la segmentación uniforme se reemplaza por el alineamiento usando el algoritmo de Viterbi. Los valores de los parámetros iniciales calculados por *HInit* son más adelante reestimados por la herramienta *HRest*. Nuevamente los datos etiquetados se utilizan, pero esta vez el procedimiento *k-media* segmentada se reemplaza por el procedimiento de reestimación de Baum-Welch.

Una vez creado el conjunto inicial de modelos, la herramienta *HERest* se utiliza para realizar un entrenamiento integrado utilizando el conjunto de entrenamiento completo. *HERest* realiza una simple reestimación Baum Welch del conjunto total de modelos de fonios HMM simultáneamente. Para cada expresión de entrenamiento, los modelos de fonios correspondientes se



concatenan y después el algoritmo forward-backward se utiliza para acumular las estadísticas de ocupación de estados, medias y varianzas para cada HMM en la secuencia. Cuando se han procesado todos los datos de entrenamiento, las estadísticas acumuladas se utilizan para calcular la reestimación de los parámetros de HMM.

La filosofía de la construcción de un sistema en HTK consiste en que los HMMs deben ser refinados incrementalmente. Es decir, una progresión típica se inciaa con un cimple conjunto de Gaussianas de modelos de fonios independientes del contexto, y después iterativamente se refinan expandiéndolas para incluir dependencia del contexto y usar múltiples mezclas de componentes de distribución Gausiana. La herramienta *HHed* es un editor de definición de HMM la cual clonará los modelos en conjuntos dependientes del contexto, aplicando una variedad de parámetros uniendo e incrementando el número de componentes mezclados en distribuciones específicas. El proceso consiste en modificar un conjunto de HMMs en estados utilizando *HHed* y después reestimar los parámetros del conjunto modificado utilizando *HERest* después de cada estado. Para mejorar el desempeño para hablantes específicos, la herramienta *HEAdapt* y *HVite* se puede utilizar para adaptar HMMs a mejores modelos de características de hablantes particulares utilizando pequeñas cantidades de datos de entrenamiento o adaptación. El resultado final será un sistema adaptado.

#### **2.2.4.3 Herramienta de Reconocimiento**

HTK proporciona una herramienta de reconocimiento llamada *HVite*, la cual usa un algoritmo de 'token passing' para realizar el reconocimiento de voz basado en Viterbi. *HVite* toma como entrada una red describiendo la secuencia de palabras permitida, la definición del diccionario definiendo cómo se debe pronunciar cada palabra y un conjunto HMMs. Su forma de operación consiste en convertir la red de palabras para una red de fonios y después unir a la definición de HMM adecuada para cada instancia de fonio. El reconocimiento se realiza para un conjunto de archivos de voz almacenados.

Las redes de palabras necesarias para manejar *HVite* son generalmente ciclos (*loops*) simples de palabras en las cuales cualquier palabra puede seguir de cualquier otra. La herramienta *Hparse* se utiliza para convertir el lenguaje de especificación de la gramática en una notación equivalente de red de palabras. Finalmente, se requiere de la construcción de un diccionario a través de la herramienta *HDMan*.

#### **2.2.4.4 Herramienta de Análisis**

Cuando ya se ha construido un reconocedor con base en HMM, se necesita evaluar su desempeño. Esto generalmente se realiza utilizando el reconocedor para transcribir algunas oraciones de prueba pregrabadas y conjuntando la salida del reconocedor con las referencias correctas de las transcripciones. Esta comparación se realiza con la herramienta *HResults*, la cual utiliza programación dinámica para alinear las dos transcripciones, después cuenta errores de sustituciones, eliminaciones e inserciones. Esta herramienta presenta la opción de generar una matriz de confusión.

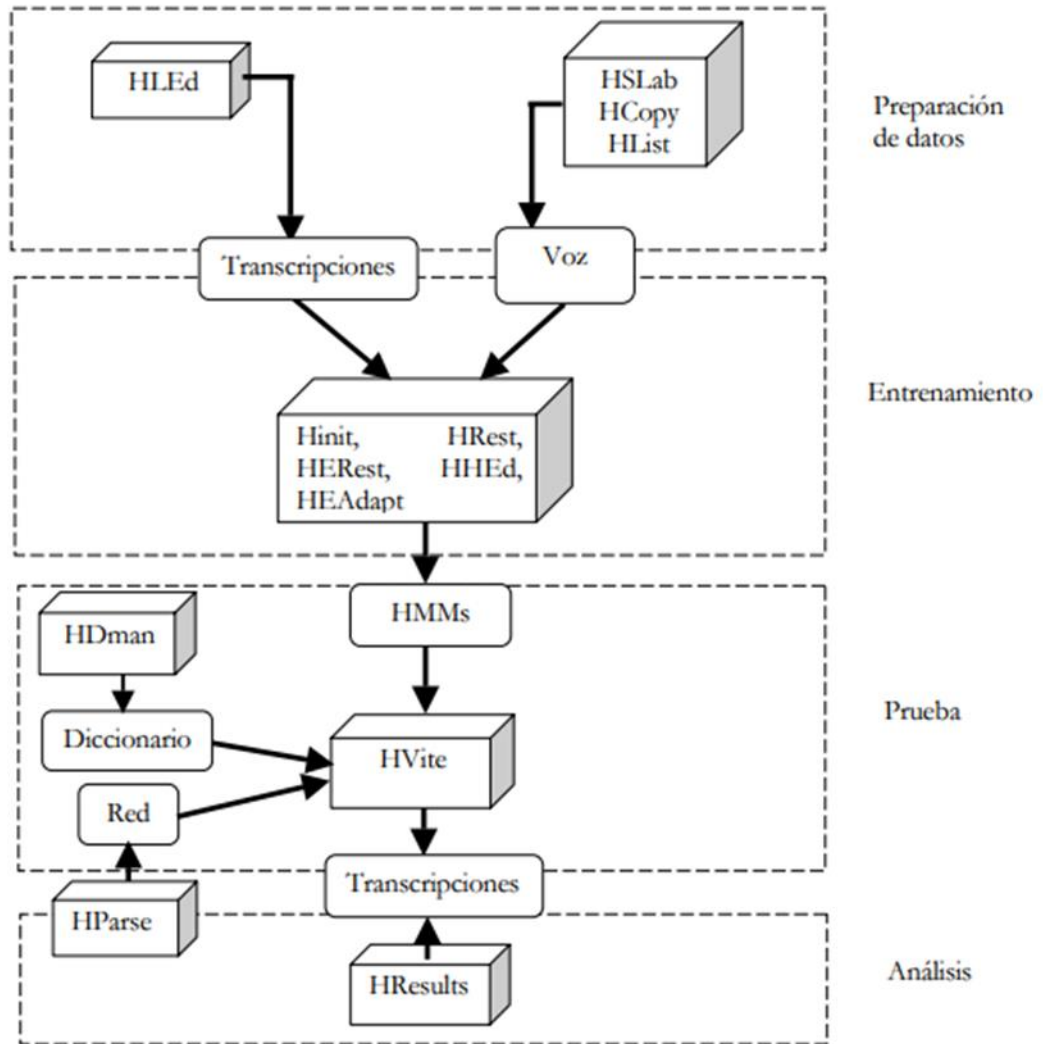


Figura 5. Arquitectura de la biblioteca HTK.

Fuente:

[http://newton.azc.uam.mx/mcc/01\\_esp/11\\_tesis/tesis/terminada/050701\\_garcia\\_galvan\\_maria\\_antonieta.pdf](http://newton.azc.uam.mx/mcc/01_esp/11_tesis/tesis/terminada/050701_garcia_galvan_maria_antonieta.pdf)

### 2.2.5 Disartria

De entre las diferentes formas que el ser humano tiene para comunicarse, el habla es la más significativa. El habla se utiliza para llevar a cabo la comunicación de forma verbal, y dar o recibir un mensaje.

Por comunicación se entiende cualquier interacción que transmite información. Relatar, informar, explicar, y expresarse son funciones de una comunicación, lo cual implica enviar y recibir mensajes con significado (Peña-Casanova, 2013). Como se presentó en el Capítulo 1, muchas personas en Perú sufren de alguna discapacidad que no les permite llevar a cabo este proceso.

(Lizandra Laplaza, n.d.) nos dice, la disartria se refiere a un grupo de desórdenes motores del habla que resulta del déficit en el control muscular de los mecanismos del habla debido a un daño en el sistema nervioso periférico o central. En este caso, el problema del habla es debido a un estado neuromuscular anormal (parálisis, atrofia, espasticidad) o el resultado de la disrupción de los movimientos de esos músculos (debilidad o falta de coordinación). Por lo tanto, a menudo hay un rango reducido en los movimientos y flexibilidad, siendo las características que generalmente se observan en el habla disártrica: hipernasalidad, consonantes imprecisas, distorsiones vocales, y problemas en el control de la velocidad.

Es por esto por lo que la disartria comprende las disfunciones motoras de la respiración, fonación, resonancia, articulación y prosodia.

Según (Sanchez, n.d.), esta discapacidad puede ser causada por lesiones o enfermedades, como las que se mencionan a continuación:

- Por un daño cerebral debido a un tumor, accidente cerebrovascular, o lesión traumática.

- Por un daño a los nervios que inervan los músculos faciales como un traumatismo facial o cervical, cirugía para cáncer de cabeza y cuello (extirpación parcial o total de la lengua o la laringe).
- Por enfermedad que afecte a nervios y músculos (enfermedades neuromusculares) como son la parálisis cerebral, esclerosis múltiple, distrofia muscular, Mal de Parkinson.
- Por otras causas: intoxicación con alcohol, prótesis dentales mal-ajustadas, efectos secundarios de medicamentos que actúan sobre el sistema nervioso central, como narcóticos, fenitoína o carbamazepina.

De entre las secuelas que pueden dejar las lesiones en el sistema nervioso (y que afectan a la expresión del lenguaje del sujeto disártrico), se pueden citar las siguientes:

- Deformaciones en la articulación por la dificultad motriz que se presenta y la falta de coordinación y control en los movimientos, pudiendo llegar a tener una expresión casi ininteligible.
- Trastornos respiratorios, con falta de sincronía entre la respiración y la fonetización, presentándose en algunos casos contracciones y espasmos que entorpecen el acto de la respiración y de la fonetización.
- Alteraciones en el tono de hipertonía o distonía, dificultando la articulación de la palabra cuando cualquiera de estos síntomas afecta a la zona bucofacial.

#### **2.2.5.1 Sintomatología**

Según (Moriana, 2009), la sintomatología se denomina de los distintos errores que se dan en el habla de la persona, los cuales se definen a partir de la raíz griega del fonema afectado: rota (/r/), sigma (/s/), lambda (/l/), etc. En donde el sufijo “*tismo*” o “*cismo*” se utiliza cuando el fonema no se articula correctamente (por ejemplo, sigmatismo o sigmacismo = dificultad para articular la /s/). Con la misma raíz y el prefijo “para” se define el error existente cuando el fonema es sustituido por otro (por ejemplo, pararrotacismo o

pararrotatismo = cuando se sustituye el fonema /r/ por otro, generalmente /g/, /d/ o /l/).

Los síntomas específicos son la sustitución, omisión, inserción y distorsión de los fonemas:

- **Sustitución:** es el error de articulación por el cual un sonido es reemplazado por otro. La persona no puede realizar una articulación y la suple por otra más fácil o, de entrada, percibe mal el sonido y lo reproduce tal como él lo discrimina (como lo emite). Es el error más frecuente dentro de las disartrias funcionales y el que presenta más dificultades para su corrección. Las formas más frecuentes son la sustitución de /r/ por /d/ o /g/, de /s/ por /z/, y del sonido /k/ por /t/.
- **Omisión:** se omite el fonema (se pronuncia “iño” por “niño”) o toda la sílaba en que se encuentra dicho fonema (se pronuncia “loj” por “reloj”).
- **Inserción:** se intercala un sonido que no corresponde a esa palabra para apoyar y resolver la articulación que se dificulta (se pronuncia “Enerique” por “Enrique”).
- **Distorsión:** se articula el sonido de forma incorrecta pero aproximada a la adecuada y sin llegar a ser una sustitución.

De esta manera, las formas y variedades de la sintomatología de la disartria se presentan a continuación:

- Rotacismo, disartria del fonema /r/ (vibrante múltiple).
- Lambdacismo, disartria del fonema /l/
- Gammacismo, disartria de los fonemas laterales /g/, /x/, y /k/.
- Deltacismo, disartria de los fonemas dentales /t/ y /d/
- Rinoartria, disartria de los fonemas nasales /m/, /n/ y /ñ/.
- Pararrotacismo, sustitución del fonema /r/ por otro como /g/, /t/, /d/, /l/, etc.
- Parasigmatismo, sustitución del fonema /s/ por otro como /t/, /z/, etc.

- Paralambdacismo, sustitución del fonema /l/ por otro.
- Paragammacismo, sustitución de los fonemas guturales por otros.
- Paradeltacismo, sustitución de los fonemas dentales por otros.

## **2.2.6 Comunicación Oral**

(Pérez, Fernández, & Torres, 2006) expresan que la comunicación oral implica el crecimiento del aspecto social, así como también tiene una influencia esencial en el desarrollo de la personalidad.

Este crecimiento individual del hombre se expresa en todo su sistema de comunicación lo que implica nuevas exigencias en sus distintas relaciones y el dominio pleno de los aspectos que integran este proceso.

Todo esto demuestra que la comunicación es una condición esencial en la formación de la conciencia de la personalidad, además sirve de estímulo al desarrollo y es un elemento fundamental en su motivación. En la relación personalidad-motivación, la comunicación es un eslabón imprescindible pues a través de ella se forma la personalidad.

El hombre desde que nace es capaz de comunicar a los que lo rodean sus estados de ánimos, sus necesidades y en la medida que va tomando experiencia en sus relaciones con otras personas el niño perfecciona su sistema de comunicación que le permite adaptarse al medio que lo rodea.

En la medida que el niño se desarrolla es capaz de interiorizar la experiencia de los adultos y transmitir esas ideas y criterios a sus coetáneos y esto lo puede lograr a través de las diferentes funciones que posee la comunicación.

### **2.2.6.1 Medición de la calidad de Comunicación**

Se necesitó medir la calidad de la comunicación entre un paciente disártrico y una persona sin discapacidad, para ello, se decidió tomar como

referente la escala analítica en Expresión e Interacciones Orales de los exámenes DELE.

(“Qué son los DELE | Exámenes - Instituto Cervantes,” n.d.) nos dice que, los diplomas de español DEL son títulos oficiales acreditativos del grado de competencia y dominio del idioma español, que otorga el Instituto Cervantes en nombre del Ministerios de Educación, Cultura y Deporte de España.

Para esta medición se adaptó la escala DELE a contexto de “conversación bilateral”. A continuación, se detallan los criterios de evaluación que se escogieron.

FLUIDEZ	
4	Se comunica con relativa fluidez, es capaz de mantener el ritmo eficazmente.
3	Se comunicada con relativa facilidad, aunque no lleva el ritmo de la conversación.
2	Las pausas son claras ya que tiene problemas para formular un discurso fácilmente entendible
1	Se le es imposible expresarse
PRONUNCIACIÓN	
4	Su pronunciación es clara e inteligible
3	Su pronunciación está distorsionada pero el receptor si llega a entender
2	Su pronunciación está demasiado distorsionada haciendo que el receptor se esfuerce para intentar comprenderlo
1	Su pronunciación es totalmente ininteligible



INTERACCIÓN	
4	Conversa con relativa facilidad y eficacia, colabora con su receptor
3	Mantiene la conversación en forma adecuada, aunque en ocasiones el receptor le pide repetir lo dicho
2	El receptor pide aclaraciones o repetir lo dicho constantemente.
1	No hay ningún tipo de entendimiento del mensaje dicho por parte del receptor

### 2.2.7 Vocabulario

(Maggiolo, 2017) nos presenta el Test de articulación a la repetición (TAR), de la profesora fonoaudióloga Edith Schwalm, el cuál es comúnmente usado para medir el nivel fonético de los pacientes, y así detectar posibles trastornos del lenguaje.

En este proyecto, nos hemos basado en la clasificación de las palabras usadas en el TAR, para crear un vocabulario que se adecue a las palabras más usadas en la vida diaria del usuario elegido. A continuación, se presenta el vocabulario desarrollado:

- CABEZA.
  - Fonemas: Bilabial “B”, Alverolares “S”, Velares “K”
  - Palabra polisilábica de 3 sílabas medias.
- COLUMBA.
  - Fonemas: Velares “K”, Alveolares “L”. Bilabiales “B, M”
  - Palabra polisilábica de 2 sílabas medias y 1 sílaba trabada.
- ELEFANTE
  - Fonemas: Alveolares “L,N”, Labio-Dentales “F”, Dentales “T”
  - Palabra polisilábica de 3 sílabas medias y 1 sílaba trabada

- GONZALES
  - Fonemas: Velares “G”, Alveolares “N,S,L”
  - Palabra polisilábica de 1 sílaba media y 2 sílabas trabadas
- MICRÓFONO
  - Fonemas: Bilabiales “M”, Velares “K”, Alveolares “R,N”, Labio-Dentales “F”
  - Dífonos Consonánticos: “KR”
  - Palabra polisilábica de 3 sílabas medias y 1 sílaba trabada.

Se trató de crear un vocabulario pequeño, para evitar el aburrimiento y frustración del paciente, pero a la vez completo, que use la mayor cantidad de fonemas y todas las vocales el alfabeto.

## **CAPITULO III:** **MARCO** **METODOLÓGICO**

## **CAPITULO III: MARCO METODOLÓGICO**

### **3.1 Tipo y Diseño del Estudio**

#### **3.1.1 Tipo de la Investigación**

(Tamayo, 2004) hace referencia que la investigación aplicada se le denomina también activa o dinámica, y depende de la investigación pura, ya que depende de sus descubrimientos y aportes teóricos. Busca confrontar la teoría con la realidad. Esta forma de investigación se dirige a su aplicación inmediata y no al desarrollo de teorías. Se refiere a resultados inmediatos y se halla interesada en el perfeccionamiento de los individuos implicados en el proceso de la investigación.

En este proyecto de investigación el tipo de investigación es de tipo Aplicada, ya que se pretende un modelo de Aprendizaje Automático en la implementación de un sistema con la finalidad de mejorar la calidad de comunicación de los pacientes disártricos.

#### **3.1.2 Población y Muestra**

.(Adams, 2017) nos describe los tipos de Sistemas de Reconocimiento de Voz, entre ellos, el Reconocimiento de Voz Dependiente del Hablante. Nos dice que es un tipo de reconocimiento de voz que depende de la persona que habla. Requiere entrenamiento para ser más preciso en la conversión de voz a texto. El entrenamiento se realiza a menudo a través de una serie de muestras de conversión que luego son corregidas por el hablante.

Debido a que en este proyecto se desarrollará un Sistemas de Reconocimiento de Voz de tipo dependiente del hablante, se usarán 5 personas para entrenar y testear el sistema.

En la fase de testeo, cada persona repetirá 5 palabras, 4 veces cada una.  
En total se evaluarán 100 palabras emitidas.

### 3.1.3 Operacionalización de Variables

Variable	Dimensiones	Indicadores
<b>Variables Independiente:</b>  Sistema de Reconocimiento Automático de Voz	Eficiencia del Sistema	Tasa de Reconocimiento de Palabra.
<b>Variable Dependiente:</b>  Calidad de la Comunicación	Fluidez  Pronunciación  Iteracción	Escala analítica en Expresión e Interacciones Orales de los exámenes DELE

#### 3.1.3.1 Variable Independiente

Se toma el Sistema de Reconocimiento Automático de Voz, como variable independiente, ya que es la variable que influirá en el cálculo de la variable dependiente.

(Young et al., 2002) nos dice, que para calcular la eficiencia de un SRAH, se usa la *Tasa de Reconocimiento de Palabra*, por ello viene a ser la dimensión de la variable independiente. Se calcula, haciendo uso de la siguiente fórmula:

$$Tasa\ de\ Reconocimiento = \frac{\#palabras\ correctamente\ reconocidas}{\#Total\ de\ Palabras} \times 100\%$$

### **3.1.3.2 Variable Dependiente**

Se toma la Calidad de la Comunicación entre el paciente Disártrico y otra persona, como variable dependiente, ya que es el factor que queremos mejorar haciendo uso del Sistema de Ayuda al Habla.

(“Qué son los DELE | Exámenes - Instituto Cervantes,” n.d.) nos dice, que para evaluar la calidad de la comunicación se hace uso de Escala analítica en Expresión e Interacciones Orales de los exámenes DELE, y se toma en cuenta tres dimensiones: Fluidez, Pronunciación e Interacción.

## **3.2 Metodología de Recolección de Datos**

### **3.2.1 Técnica e Instrumento de Recolección de Datos**

La técnica para la recolección de datos consiste en que el paciente disártrico emitirá las palabras que pertenecen al vocabulario de entrenamiento, sin exagerar en la articulación ni acentuación, lo más cercano a como habla en su vida diaria.

Se usará la escala analítica con 3 categorías, que constan respectivamente de cuatro bandas ordinales que se extienden en un rango único que va de 0 a 3 puntos. A cada una de las bandas corresponde un único descriptor ilustrativo con el que se compara la actuación del emisor y el receptor.

Se realizará dos evaluaciones para cada palabra del vocabulario de entrenamiento, la primera sin el Sistema de Ayuda al Habla, y la segunda con el sistema, y se obtendrán los puntajes de ambos escenarios.

### **3.2.2 Diseño de Contrastación de Hipótesis**

Tomaremos como variable el puntaje obtenido en las evaluaciones con la escala analítica de los exámenes DELE, la cuál será medida en dos momentos temporales distintos, que en general llamaremos “Antes del SAH” y “Después el SAH”. Estas variables serán comparadas entre sí, se comprobará la hipótesis siempre y cuando la variable “Después del SAH” sea mayor a la variable “Antes del SAH”

## **CAPITULO IV: DESARROLLO DEL RECONOCEDOR DEL HABLA**



## CAPITULO IV: DESARROLLO DEL RECONOCEDOR DEL HABLA

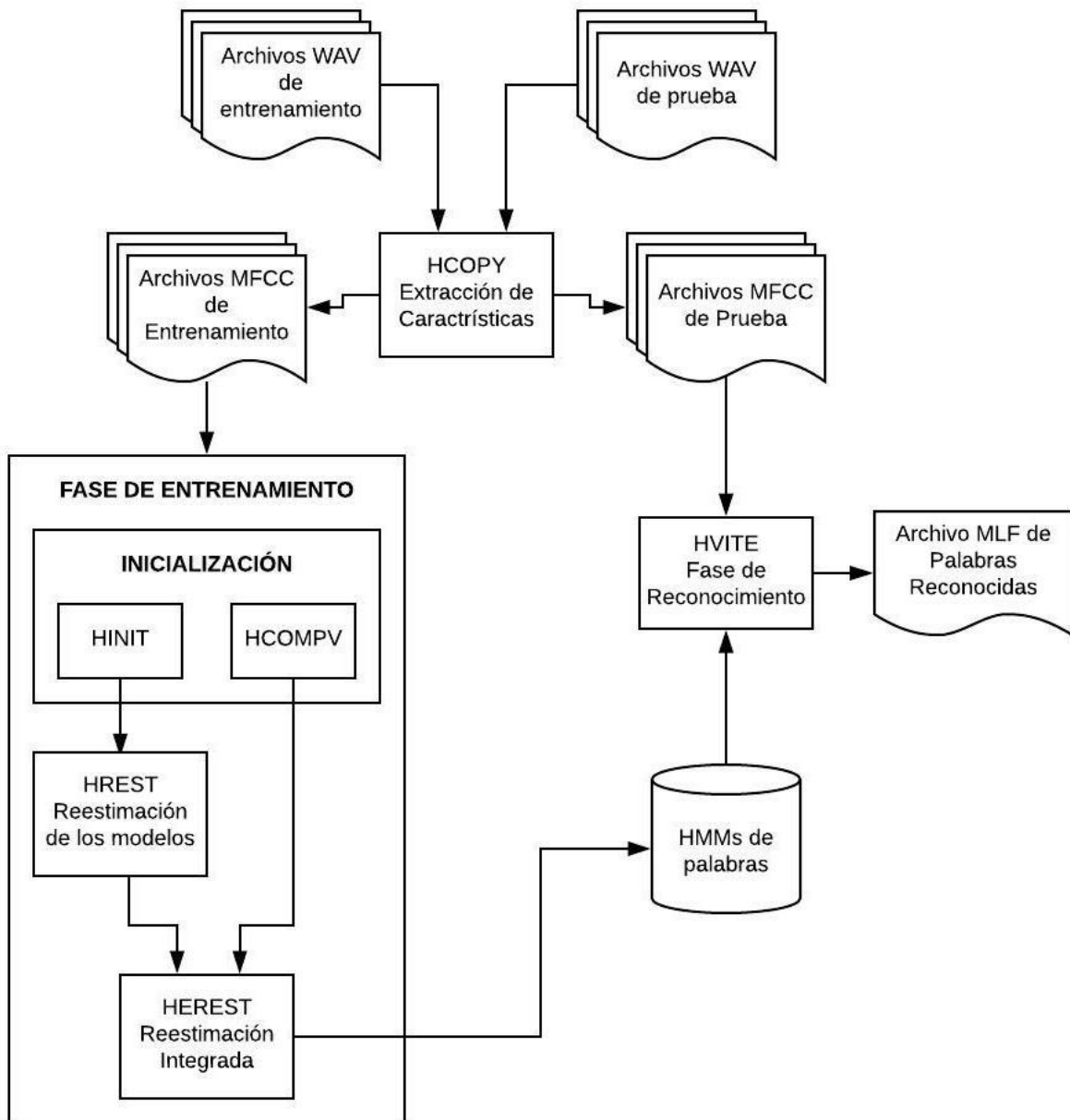


Figura 6: Diagrama de flujo para la creación de un Reconocedor de Voz

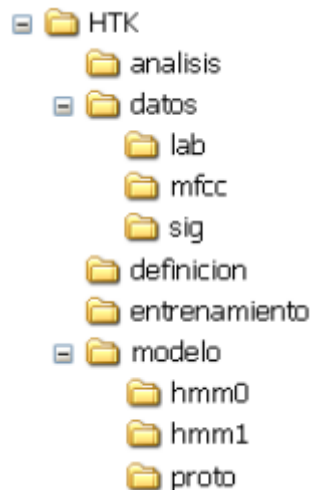
## 4.1 Requisitos y Preparación

### 4.1.1 Requisitos Previos

Para la realización de este sistema se necesitó un ordenador con Windows y MS-DOS, micrófono configurado y altavoces o auriculares. También se debió disponer de los ejecutables binarios de HTK 3.3, que pueden obtenerse de <http://htk.eng.cam.ac.uk/> previo registro.

### 4.1.2 Preparación del entorno de trabajo

Se usó la siguiente estructura de directorios:



*Figura 7. Estructura de directorios*

Las carpetas *.sig* y *.lab* almacenan los archivos de voz ya grabados y sus respectivas transcripciones.

La carpeta *. analisis* contiene los archivos de configuración necesaria para el análisis de los datos de entrada.

La carpeta *proto* almacena los archivos *hmm\_x*, donde *x* es una palabra. Estos archivos contienen la definición y estructura de los HMMs que representan las palabras que se reconocen.

La carpeta *datos* almacena los archivos de lista con la correspondencia entre las señales de voz grabadas y sus archivos de coeficientes MFCC.

La carpeta *entrenamiento* contiene la lista de archivos que se usan para entrenar los modelos.

## **4.2 Diccionario fonético equilibrado**

Por lo general, para construir un Diccionario de Pronunciación es crear la lista ordenada de las palabras contenidas en nuestra gramática, una por línea, con pronunciaciones (los fonemas que componen cada palabra). En nuestro caso, al ser un sistema desarrollado para personas disártricas, el diccionario será sólo a nivel ortográfico, debido a la omisión de fonemas cuando el paciente emite las frases.

Para crear un diccionario de pronunciación en HTK, seguiremos estos pasos:

1. Crear un archivo `prompts.txt`, que es la lista de palabras que registraremos en el siguiente paso,
2. Derivar un archivo `wlist` del archivo `prompts.txt` - el archivo `wlist` es una lista ordenada de las palabras únicas que parecen en el archivo `prompts.txt`.

### **4.2.1 Archivo `prompts.txt`**

Primero necesitamos crear un archivo `prompts.txt` que incluya nuestras palabras gramaticales. Este archivo contiene básicamente la lista de palabras que deben grabarse y los nombres de los archivos de audio que se guardarán, uno por línea.

La primera columna del archivo `prompts.txt` (Ver contenido en Anexo A), contiene el nombre del archivo de audio que se creará, y la siguiente columna contiene las transcripciones de texto de lo que se grabará en el archivo de audio.

#### 4.2.2 Archivo wlist

Se ejecuta el archivo Perl, *prompts2wlist.pl* (Ver contenido en Anexo B), el cual toma el archivo *prompts.txt*, elimina el nombre del archivo en la primera columna e imprime cada palabra en una línea en un archivo de lista de palabras, *wlist* (Ver contenido en Anexo C).

#### 4.3 Grabación de los datos

Las muestras de voz que serán utilizadas para entrenar y probar los modelos se graban en formato WAV monoaural con velocidad de muestreo 16 kHz. Actualmente hay diversidad de herramientas de uso libre para la grabación de muestra de voz, sin embargo, para este proyecto se hace referencia al software *Audacity*. Con este programa, además de poderse grabar voz, se puede hacer el etiquetado de la misma.

Se trató de que estas grabaciones sean lo más limpias posibles, minimizando el ruido de fondo, pero a la vez cuidando que la pronunciación se asemeje lo más posible al modo habitual en el que habla el paciente.

#### 4.4 Creación de los archivos de transcripción

HTK toolkit no puede procesar el archivo *prompts.txt* directamente, por lo que se crea un archivo Master Label File (*MLF*), que es un único archivo que contiene una entrada de etiqueta para cada línea del archivo *prompts.txt*.

Se utiliza el archivo de Perl, *prompts2mlf.pl* (ver contenido en Anexo D), para realizar la conversión. Esto generará un archivo *words.mlf* (ver contenido en Anexo E), el cual contiene la transcripción a nivel de palabra de los archivos de voz.

#### 4.5 Codificación de los audios

Ya que se tiene el corpus oral completo, es necesario extraer información de las muestras de voz de tal manera que se optimice el proceso de RAH. HTK

llama a este último paso en la preparación de datos la “Parametrización de las formas de onda de voz brutas en secuencias de vectores de características”. Todo esto significa que HTK no es tan eficiente en el procesamiento de archivos WAV como lo es con su formato interno. Por lo tanto, los archivos de audio WAV grabados previamente debieron ser convertidos a otro formato denominado MFCC (que se refiere a Mel Frequency Cepstral Coefficients). (Jurafsky & Martin, 2008; Young et al., 2002) nos dice que los MFCCs, son denominados “vectores de características” y es el formato más usados para SRAHs). Se derivan de la Transformada de Fourier (FT) o de la Transformada de Coseno Discreto (DCT). (Davis & Mermelstein, 1980) nos dicen que los MFCCs se consideran vectores de información espectral de la señal de voz, los cuales son la entrada para el SRAH.

Al experimentar con señales, lo primero que debemos lograr es disponer de una base de datos parametrizada, sólo así, posteriormente, podremos realizar múltiples experimentos. Para construirla, HTK dispone de la herramienta HCopy. Las entradas de HCopy son la señal en banda base (en formato wav) y un archivo de configuración de la parametrización, el cual le indica a HTK la manera de analizar los datos. HCopy entrega como salida un archivo de voz parametrizada convertida en una matriz o vector de vectores.

```
HCopy -T1 -C analisis.conf -S codetrain.scf
```

Por defecto HCopy realiza una simple copia del fichero de origen en el fichero destino. Dicho comportamiento puede modificarse con las opciones especificadas en el fichero de configuración. En este fichero pueden detallarse un conjunto de transformaciones que se aplican sobre el fichero origen y se guardan en el fichero destino. La idea es que exista un tipo de parámetros origen y un tipo de parámetros destino de forma que, especificando dichos tipos en el fichero de configuración, HCopy realice la conversión entre ambos formatos.

Este proceso consta de los siguientes pasos:

- La señal es segmentada en tramas sucesivas (cuya longitud es elegida normalmente entre 20 y 40 mseg)
- Cada trama se multiplica por una ventana de Hamming
- De cada trama enventanada se extrae un vector coeficientes MFCC.

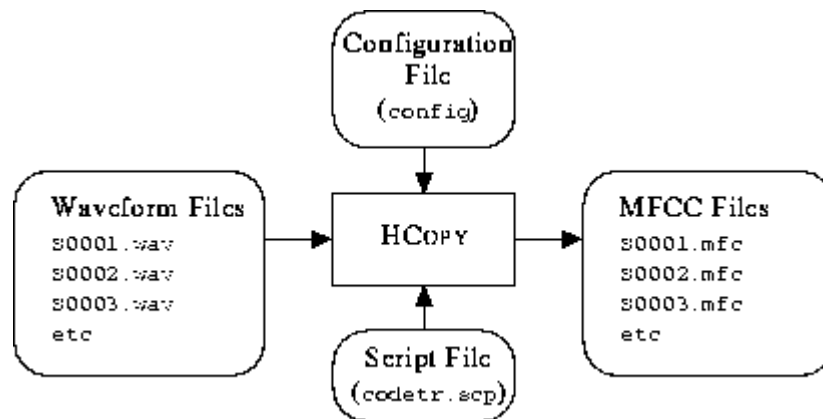


Figura 8. Procesamiento de Archivos HCOPY

#### 4.5.1 Archivo codetrain.scf

Se creó el archivo *codetrain.scf* (Ver contenido en Anexo F) que contiene una lista de cada archivo de audio fuente y el nombre del archivo MFCC en el que se convertirá. Los archivos de salida parametrizada deben poseer extensión *.mfcc*, indicando con ello que los datos corresponden a vectores codificados según coeficientes ceptrales de frecuencia MEL.

#### 4.5.2 Archivo de Configuración

Para realizar la conversión de formato WAV a MFCC, se requiere un archivo de configuración *analysis.conf* (Ver contenido en Anexo G), que especifique todos los parámetros de conversión necesario.

- Las características de los MFCCs (TARGETKIND = MFCC\_0\_D\_A)
- El formato de origen del archivo de voz en WAV (SOURCEFORMAT = WAV)

- Usualmente se especifica que la señal de muestree cada 10 ms para su codificación y que el tamaño de la ventana para el análisis sea de 25 ms. (WINDOWSIZE = 250000.0, TARGETRATE = 100000.0)
- Se especifica que se usarán 12 MFCCs (NUMCEPS = 12)

#### 4.6 Entrenamiento Supervisado de los Modelos Acústicos

Al realizar el entrenamiento de un modelo acústico involucra dos procesos: inicialización y reestimación. A continuación, se presenta, de manera general, la secuencia de ejecución y configuración de biblioteca HTK para el entrenamiento de HMM's, según (Young et al., 2002).

La creación de HMM's, es la técnica más común para el desarrollo de SRAHs. Los modelos acústicos para palabras se construyeron siguiendo una topología estándar de cinco estados (con dos no emisores) con secuencia izquierda-a-derecha.

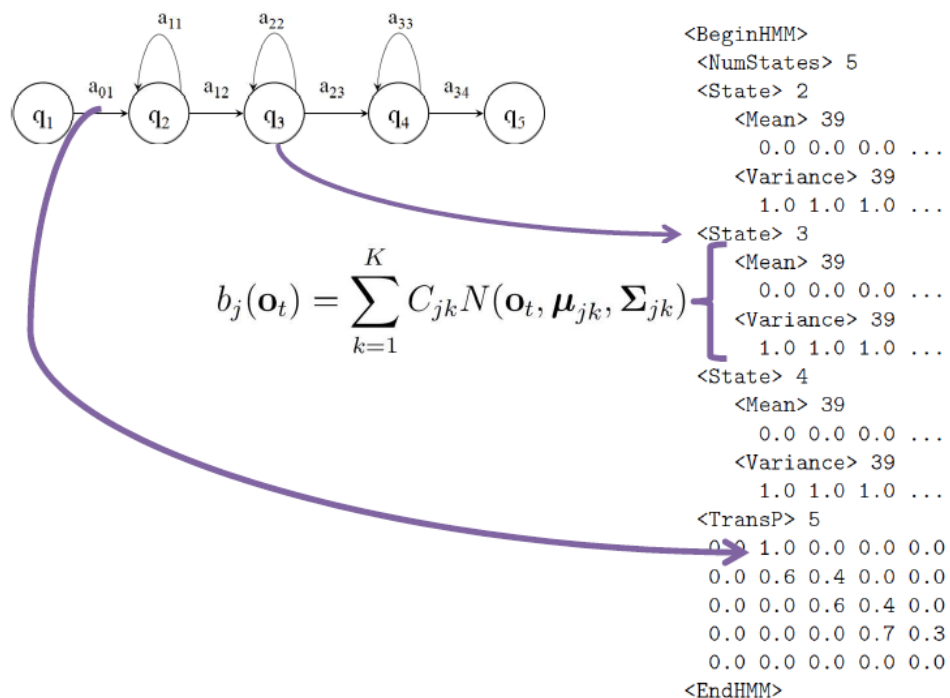


Figura 98. Declaración de un HMM en HTK con un solo componente gaussiano.  
Archivo proto.

Debemos modelar tantos eventos como palabras distintas tengamos en el vocabulario, sin olvidar el silencio. El primer paso para diseñar el HMM es elegir las características de la topología a usar:

- **Número de estados.** Entre 4 y 8 estados activos, pues trataremos con palabras cortas. Además, se añaden un primer y último estado que no generan observación por motivos de implementación en HTK.
- **Forma de la función de observación asociada a cada estado.** Son funciones densidades de probabilidad gaussianas unidimensionales (con matrices diagonales de 39 elementos no nulos), inicializadas con media cero y varianza uno. Cada elemento de la diagonal representa a un coeficiente MFCC.
- **Transiciones entre estados.** Una matriz cuadrada, de dimensión igual al número de estados. De cada estado se puede volver a él mismo, o transitar a los dos estados siguientes (excepto el primero y último estado, que son transitorios).

La Figura 9., muestra el contenido del archivo proto. Se expondrá el significado de cada línea:

- `~o <VecSize> 39 <MFCC_0_D_A>` indica la longitud del vector de coeficientes y el tipo de coeficientes.
- `~h "x" <BeginHMM> (...) <EndHMM>` encierra la definición del HMM llamado "x"
- `<NumStates> 5` especifica el número total de estados, los cuatro activos y los dos transitorios.
- `<State> i` introduce a la especificación de las características del estado i.ésimo. La matriz diagonal que representa la probabilidad de generación en ese estado puede especificarse poniendo simplemente los elementos de la diagonal: un vector de medias y otro de varianzas para las 39 gaussianas. Los estados 1 y 6 no se describen, pues no tienen función de observación.



- *<Mean>* 39 especifica el vector de medias de 39 elementos de la función de observación del estado al que acompaña. Estos coeficientes serán entrenados posteriormente y su valor cambiará, pero ahora, que estamos definiendo la estructura, son inicializados a cero.
- *<Variance>* 39 especifica el vector de varianzas de 39 elementos de la función de observación del estado al que acompaña. Son inicializados a uno, aunque serán entrenados posteriormente.
- *<TransP>* *N* especifica la matriz de transición de estados, de dimensiones *N* x *N*
- Los valores nulos indican transiciones no permitidas entre estados. El resto de valores pueden ser inicializados arbitrariamente, pero los elementos de cada línea deben sumar 1. Se recomienda, es la medida de lo posible, que todos los elementos no nulos de cada línea tengan el mismo valor para que ninguna transición parta con ventaja en el proceso de entrenamiento.
- Este prototipo debe copiarse para cada palabra que vayamos a usar, simplemente cambiando el nombre del archivo (proto) y la cabecera (*~h "x"*). todos estos elementos se almacenarán en *modelo/protobase* que será el punto de partida para entrenar los modelos.

La Inicialización de los modelos acústicos con el corpus se realiza mediante la herramienta HInit de la siguiente manera:

```
HInit -A -D -T 1 -C analisis/config.conf -S entrenamiento/train.scp -M
modelo/hmm0 -H modelo/protobase/palabra -I etiqueta -L datos/lab etiqueta
```

En donde:

- *config.conf* es un archivo de configuración igual a *analisis.conf* pero sin la línea SOURCEFORMAT = WAV. Esto dado que el archivo se usará para configurar la lectura de archivos en lugar de su codificación a otro formato.

- *train.scp*, es un archivo en donde se especifica la lista de archivo de sonido para el entrenamiento. Similar a *codetrain.scp*, pero sin la columna de datos en donde se especifican los archivos de origen a codificar.

Después de configurar un prototipo apropiado, se puede inicializar un HMM utilizando la herramienta HTK HINIT. El principio básico de HINIT depende del concepto de un HMM como generador de vectores de voz. Cada ejemplo de entrenamiento se puede ver como el resultado del HMM cuyos parámetros se deben estimar. Por lo tanto, si se conociera el estado que generó cada vector en los datos de entrenamiento, entonces las medias y variaciones desconocidas podrían estimarse promediando todos los vectores asociados con cada estado. De forma similar, la matriz de transición podría estimarse simplemente contando el número de intervalos de tiempo que ocupaba cada estado.

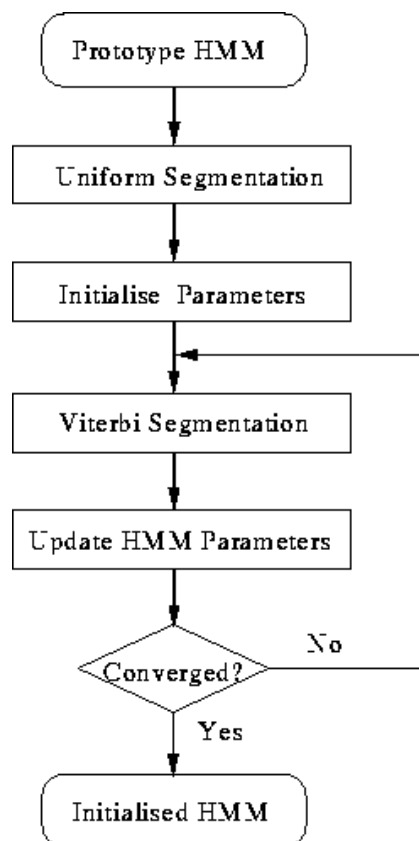


Figura 10. Operación HINIT

La idea anterior puede implementarse mediante un esquema iterativo como se muestra en la figura 10. En primer lugar, el algoritmo de Viterbi se usa para encontrar la secuencia de estado más probable correspondiente a cada ejemplo de entrenamiento, luego se estiman los parámetros de HMM. Como efecto secundario de encontrar la alineación de estado de Viterbi, se puede calcular la probabilidad logarítmica de los datos de entrenamiento. Por lo tanto, todo el proceso de estimación se puede repetir hasta que no se obtenga un aumento adicional en la probabilidad.

Este proceso requiere algunos parámetros iniciales de HMM para comenzar. Para evitar este problema, HINIT comienza segmentando de forma uniforme los datos y asociando cada segmento sucesivo con estados sucesivos.

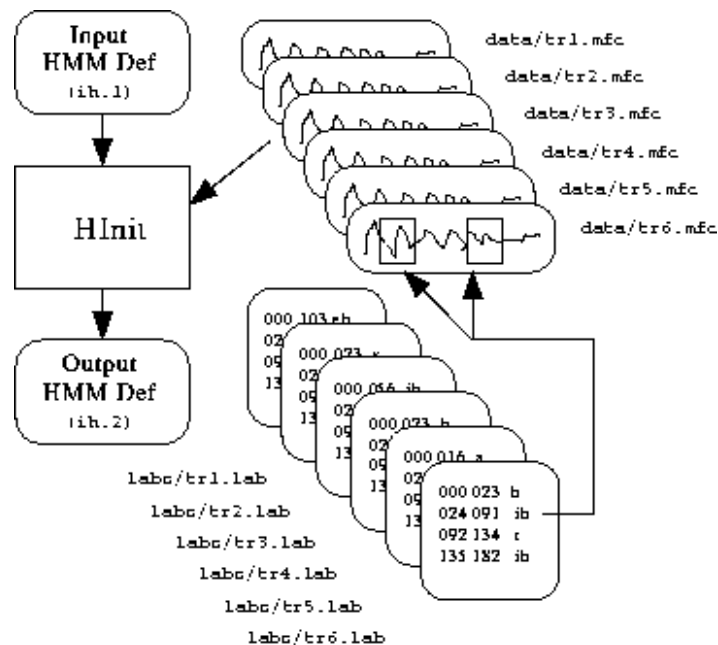


Figura 11. Procesamiento de archivos en HINIT

HInit aprovecha las facilidades del segmentado de etiquetas para detectar las diferentes palabras. Las busca en cada modelo y posteriormente utiliza el algoritmo de Viterbi para la estimación de las medias y varianzas. El proceso se repite hasta que el algoritmo ya no pueda mejorar la semejanza,

que es calculada en paralelo. Inicializa los HMM alineando en el tiempo los estados con los datos de entrenamiento, usando el algoritmo de Viterbi.

Este proceso se repite para cada una de las palabras del vocabulario. Después de este paso se estima un valor global inicial para la varianza de los HMM's, el cual se guarda en el archivo *vFloors*. Esto se realiza con la herramienta HCompV de la siguiente manera:

```
HCompV -A -V -D -T 1 -C analisis/config.conf -S entrenamiento/train.scp -M
modelo/hmm0 -H modelo/protobase/palabra -f 0.01 palabra
```

La entrada/salida de archivos de definición HMM y archivos de capacitación en HCOMPV funciona exactamente de la misma manera que la descrita anteriormente para HINIT. Lee en una definición prototipo de HMM y algunos datos de entrenamiento y genera una nueva definición en la que cada media y covarianza es igual a la media y la covarianza del habla global.

El siguiente paso consiste en realizar una iteración del proceso de reestimación de los modelos. La herramienta HRest estima los valores óptimos de los parámetros de los HMM (probabilidades de transición y vectores de medias y varianzas de cada función de observación). Cada vez que la ejecutemos, se producirán varias iteraciones del algoritmo de Baum-Welch (reestimación de los parámetros de los modelos aislados) hasta que converja.

Antes de ejecutar el comando, en el directorio *modelo/* se crea la carpeta *hmm1*, que almacenará los archivos de descripción resultantes.

```
HRest -A -D -T 1 -C analisis/config.conf -S entrenamiento/train.scp -M
modelo/hmm1 -v min_varianza -H modelo/hmm0/palabra -I palabra -L datos/lab
palabra
```

HREST es la herramienta final en el conjunto diseñado para manipular unidades aisladas HMM. Su funcionamiento es muy similar al de HINIT, excepto que, como se muestra en la figura 12, espera que la definición de

HMM de entrada se haya inicializado y utiliza la reestimación de Baum-Welch en lugar del entrenamiento de Viterbi.

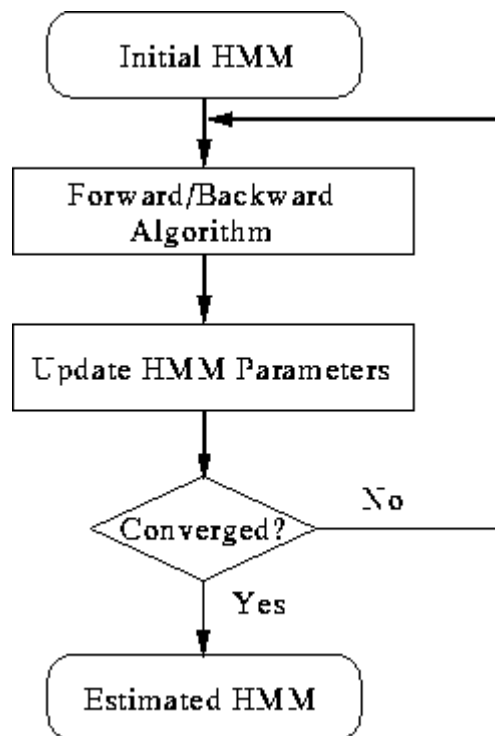


Figura 129. Operación HREST

Esto implica encontrar la probabilidad de estar en cada estado en cada marco de tiempo usando el algoritmo Baum-Welch. Esta probabilidad luego se usa para formar promedios ponderados para los parámetros de HMM. Por lo tanto, mientras que el entrenamiento de Viterbi toma una decisión difícil en cuanto a qué estado fue "generado" por cada vector de entrenamiento, Baum-Welch toma una decisión suave. Esto puede ser útil al estimar los HMM basados en fonemas, ya que no existen límites duros entre los fonemas en el habla real y el uso de una decisión suave puede dar mejores resultados.

Este proceso se repite tres veces, en donde los modelos reestimados de *hmm0* se guardan en *hmm1*, de *hmm1* a *hmm2*, y de *hmm2* a *hmm3*. La figura 13, muestra el flujo de archivos de entrada y de salida en HRest.

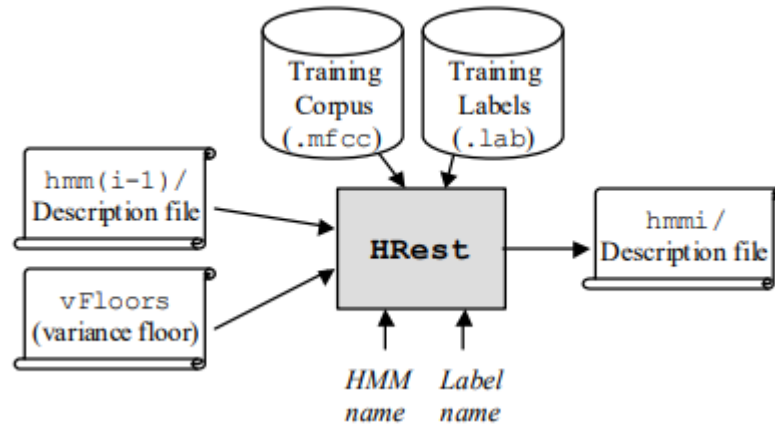


Figura 103. Procesamientos de Archivos en HRest

En este punto se tienen en *hmm3* un archivo con un HMM para cada palabra. Para hacer más manejable el uso de modelos, éstos se unen en un solo archivo de definición de modelos acústicos llamado *hmmdefs* (Ver contenido en Anexo H), y *vFloors* (Ver contenido en Anexo I) se renombra como *macros*.

El archivo de HMMs final *hmmdefs*, y el que lleva información global de la varianza de los datos *macros*, se copian al directorio *hmm4* para su reestimación con la biblioteca HERest. Se usa de la siguiente manera:

```

HERest -C analisis/config.conf -L datos/lab -t 250 150 1000 -S
entrenamiento/train.scp -H modelo/hmm4/macros -H modelo/hmm4/hmmdefs -M
modelo/hmm5 definicion/wlist

```

Mientras que el entrenamiento aislado de la unidad es suficiente para construir modelos de palabras y la inicialización de modelos usando datos de arranque etiquetados a mano, los principales procedimientos de entrenamiento HMM para construir sistemas de palabras giran en torno al concepto de entrenamiento integrado. A diferencia de los procesos descritos hasta ahora, el entrenamiento integrado actualiza simultáneamente todos los HMM en un sistema utilizando todos los datos de entrenamiento. Es realizado por HEREST que, a diferencia de HREST, realiza una sola iteración.

En resumen, HEREST funciona de la siguiente manera. Al inicio, HEREST carga en un conjunto completo de definiciones de HMM. Cada archivo de entrenamiento debe tener un archivo de etiqueta asociado que proporcione una transcripción para ese archivo. Sin embargo, HEREST utiliza únicamente la secuencia de etiquetas, y se ignora cualquier información de ubicación del límite. Por lo tanto, estas transcripciones se pueden generar automáticamente a partir de la ortografía conocida de lo que se dijo y un diccionario de pronunciación.

HEREST procesa cada archivo de entrenamiento a la misma vez. Después de cargarlo en la memoria, usa la transcripción asociada para construir un HMM compuesto que abarca todo el enunciado. Este HMM compuesto se hace mediante la concatenación de instancias de los HMM de las palabras correspondientes a cada etiqueta en la transcripción. El algoritmo Baum-Welch se aplica luego y las sumas necesarias para formar los promedios ponderados se acumulan de la manera normal. Cuando se hayan procesado todos los archivos de entrenamiento, las nuevas estimaciones de parámetros se forman a partir de las sumas ponderadas y se genera el conjunto de HMM actualizado.

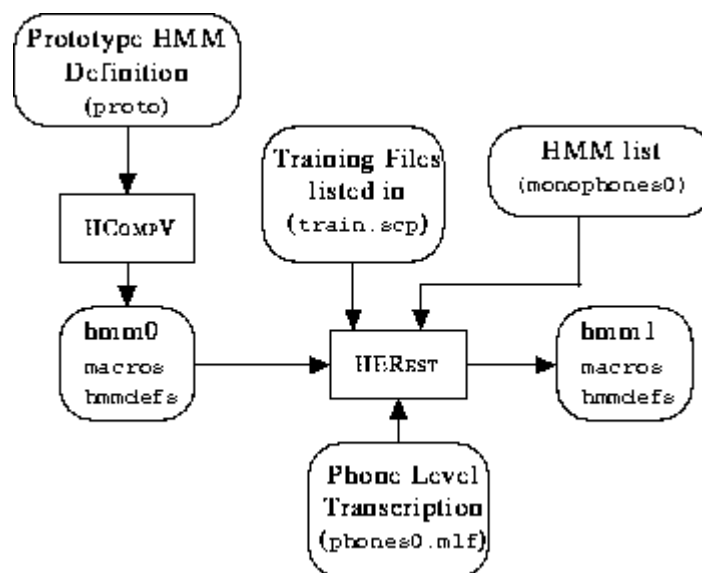


Figura 114. Procesamientos de archivos en HEREST

Sin embargo, repetir la reestimación a la convergencia puede llevar un tiempo increíblemente largo. Peor aún, puede llevar a un exceso de entrenamiento ya que los modelos pueden estar demasiado igualados a los datos de entrenamiento y no se pueden generalizar bien en datos de prueba no vistos. Por lo tanto, en la práctica, entre 2 y 5 ciclos de reestimación incorporada son normalmente suficientes cuando se entrena un modelo de palabra.

#### 4.7 Definición de la gramática

Ahora, se especifica a HTK para qué tipo de problema hemos diseñado el reconocedor. Cada archivo relativo a esta tarea debe almacenarse en el directorio *definición*.

##### 4.7.1 Gramática

Antes de usar los modelos de las palabras, tenemos que definir la arquitectura básica al reconocedor. El reconocedor solo trabajará con palabras aisladas o series de palabras. Esto implica que la entrada será del tipo: silencio inicial, palabra (o palabras) y silencio final.

En HTK, la gramática se escribe en un fichero de texto, acorde con ciertas reglas sintácticas. En nuestro caso será bastante simple, y se almacena en el archivo *definición/gramatica.txt*, y contendrá lo siguiente:

```
$palabra = palabra1 | palabra2 | palabra3 | (...);  
( { silencio } [$palabra ] { silencio } )
```

Esto indica a HTK que la variable palabra puede reemplazarse por *palabra1*, *palabra2*, *palabra3*, .... Las llaves ( { } ) denotan cero o varias repeticiones (sin silencio, un silencio corto o largo). Los corchetes ( [ ] ) alrededor de *\$palabra* denotan cero o una ocurrencia (si no se pronuncia ninguna palabra, es posible reconocer sólo silencio).



### 4.7.2 Diccionario

El sistema debe saber a qué HMM corresponde cada una de las variables del archivo de gramática. Estas variables se almacenan en un archivo de texto llamado *definición/diccionari.txt*. cuando la estructura es simple, esta correspondencia es obvia, y el diccionario simplemente contiene:

<i>palabra1</i>	[ <i>palabra1</i> ]	<i>HMM_palabra1</i>
<i>palabra2</i>	[ <i>palabra2</i> ]	<i>HMM_palabra2</i>
<i>palabra3</i>	[ <i>palabra3</i> ]	<i>HMM_palabra3</i>
(...)	(...)	(...)
<i>silencio</i>	[ ]	<i>HMM_silencio</i>

Los elementos de cada columna son:

- Los elementos de la columna izquierda se refieren a los nombres de las variables en el fichero gramática.
- Los elementos de la columna derecha se refieren a los nombres de los HMM (introducido por los *~h* en los archivos de definición de los HMM).
- Los elementos entre corchetes de la columna central son opcionales, e indican los símbolos que serán transcritos por el reconocedor (por defecto, los nombres de las variables del archivo de gramática serán usados). Lo normal es poner los nombres de las etiquetas asociadas.

### 4.7.3. Red Gramática

La gramática descrita en *definición/gramatica.txt* no es inteligible por HTK y antes de usarla para ejecutar el reconocedor, debe ser compilada con la herramienta HParse, para obtener la red *definición/red.slf*. el comando por ejecutar es el siguiente:

```
HParse definición/gramatica.txt definición/red.slf
```

*En este momento, nuestro reconocedor de voz está completamente definido por su red (definición/red.slf), el diccionario y el conjunto de HMMs*

## 4.8 Reconocimiento

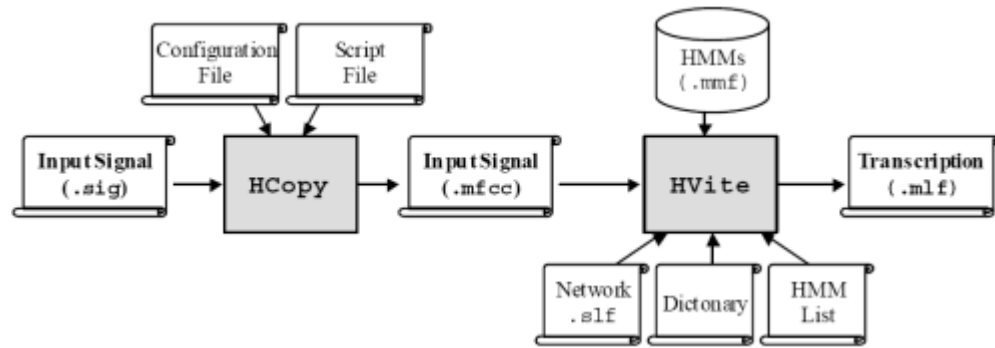
El reconocedor se probará mediante bases de datos de prueba, con el uso de HVite. Es la herramienta que maneja el algoritmo de Viterbi. Debe proporcionar el camino de mayor probabilidad para la secuencia óptima de estados. Cuanto más grande sea la red, más procesamiento requerirá el algoritmo y, por lo tanto, más lento resultará el reconocimiento.

Suponiendo que test.scp contiene una lista de los archivos de prueba codificados, cada archivo de prueba será reconocido y su salida de transcripción a un MLF llamado recout.mlf ejecutando lo siguiente:

```
HVite -T 1 -H modelo/hmm6/hmmdefs -H modelo/hmm6/macros -S test.scp -l *  
-i recout.mlf -w definición/wdnet -p 0.0 .s 5.0 dict tiedlist
```

Las opciones -p y -s establecen la penalización de inserción de palabra y el factor de escala de gramática, respectivamente. La penalización por inserción de palabras es un valor fijo agregado a cada token cuando transita desde el final de una palabra hasta el inicio de la siguiente. El factor de escala gramatical es la cantidad por la cual se escala la probabilidad del modelo de lenguaje antes de agregarla a cada token mientras transita desde el final de una palabra hasta el comienzo de la siguiente. Estos parámetros pueden tener un efecto significativo en el rendimiento del reconocimiento y, por lo tanto, vale la pena realizar algunos ajustes en los datos de las pruebas de desarrollo.

Ahora comienza el proceso de reconocimiento propiamente dicho. Cuando vamos a pasarle al programa una secuencia previamente grabada, dividimos el proceso en dos pasos:



*Figura 12 Procesamiento de Archivos en HVITE*

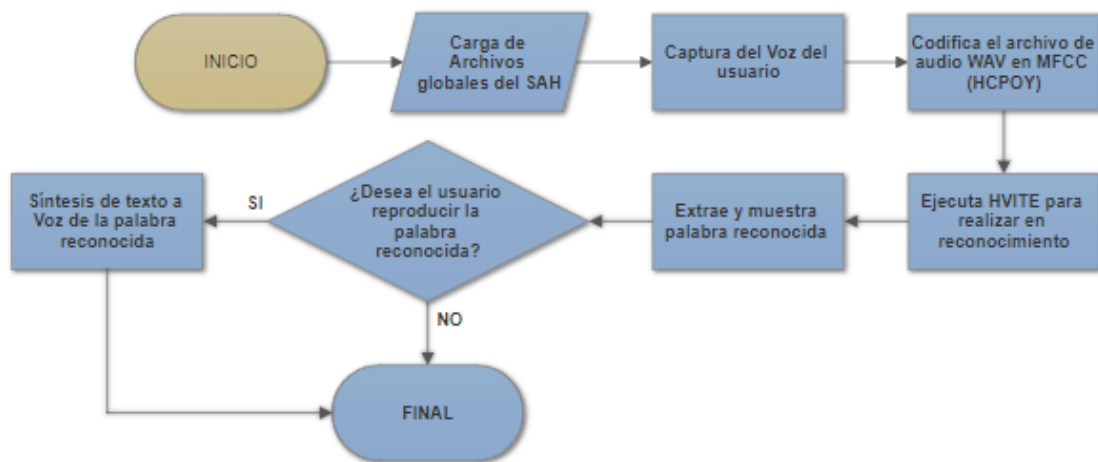
- El archivo grabado, prueba\sig\entrada.sig, es transformado en una serie de vectores acústicos (MFCC) al igual que hicimos para las secuencias de entrenamiento (con HCopy). No etiquetamos la señal, pues vamos a usarla para que el reconocedor la transcriba. El resultado se almacena en prueba\mfcc\entrada.mfcc.
- La observación de entrada se procesa mediante el algoritmo de Viterbi, calculando la probabilidad de que cada HMM produzca dicha observación.

## **CAPITULO V: DESARROLLO DE LA INTERFAZ**

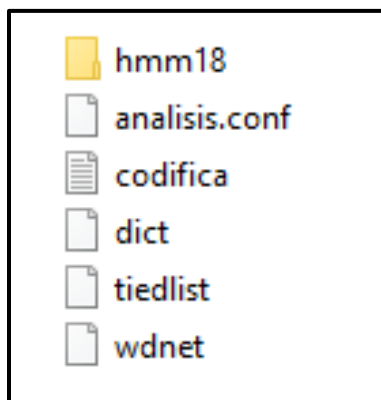
## CAPITULO V: DESARROLLO DE LA INTERFAZ

Una vez que se han construido los modelos acústicos de SAH, éste ya puede comenzar a ser usado. Para ello es diseñada una interfaz de usuario-máquina para lograr la comunicación entre ambos que se fácil de entender, se utiliza Python para crear una GUI.

### 5.1 Diagrama de Flujo



#### 5.1.1 Archivos globales del SAH



1. Carpeta *hmm18*: Carpeta que contiene los últimos modelos resultantes del Entrenamiento Supervisado de la fase de Desarrollo del Reconocedor de Voz.

2. Archivo analisis.conf: Archivo que contiene la configuración usada por la herramienta HCOPY en la interfaz.
3. Archivo codifica: Archivo con el nombre del archivo de audio y su mfcc que tendrá que ser reconocido por la herramienta HVITE en la interfaz.
4. Archivos dict, tiedlist, wdnnet: Archivos de gramática usados por la herramienta HVITE en la interfaz.

## 5.2 Arquitectura

La Figura 17., representa el prototipo de la interfaz del Sistema de Ayuda al Habla para personas Disártricas. El diagrama muestra el flujo enviado por el usuario A, de lengua española y diagnosticado con Disartria, hacia un usuario B.

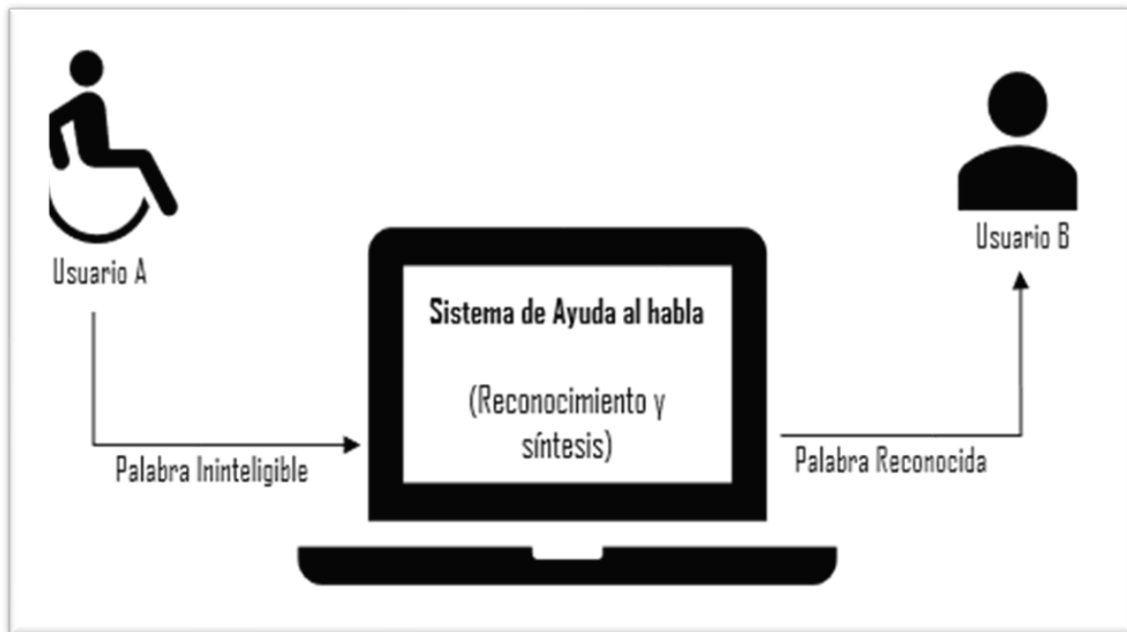


Figura 13. Prototipo del SAH

La figura 18., representa un modelo simplificado del flujo de información dentro del prototipo.

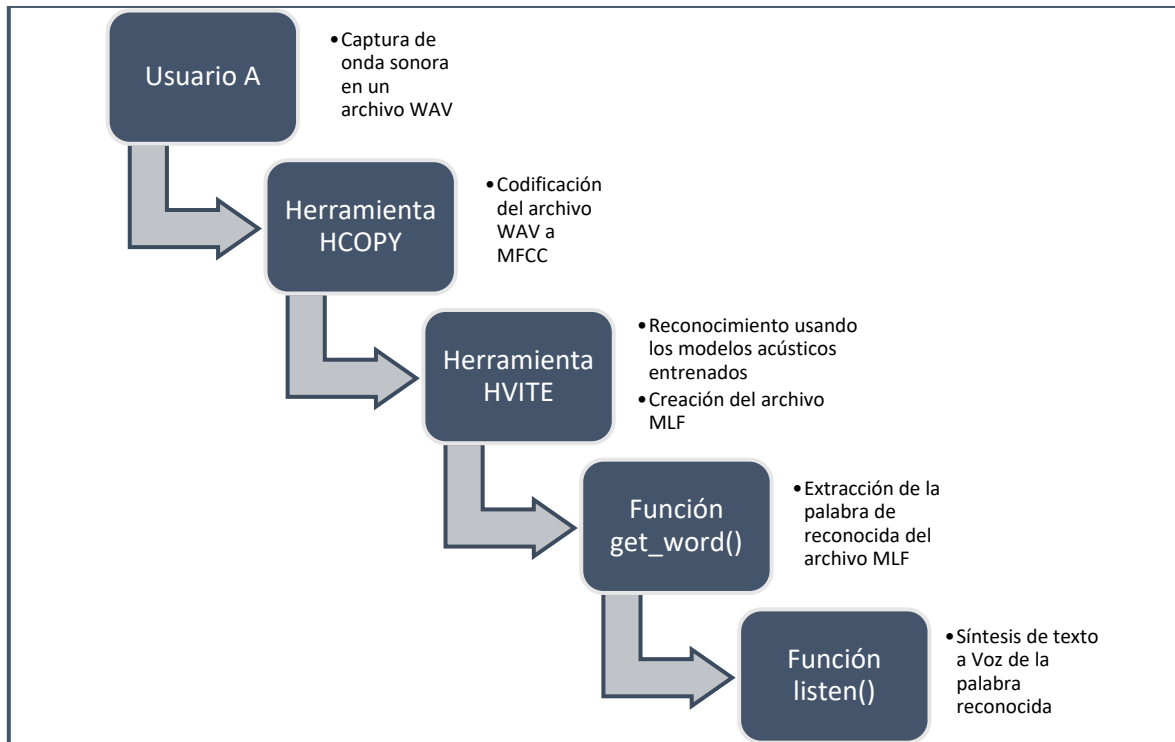


Figura 14. Flujo de información del SAH

Las bibliotecas empleadas para desarrollar la aplicación de reconocimiento de voz fueron:

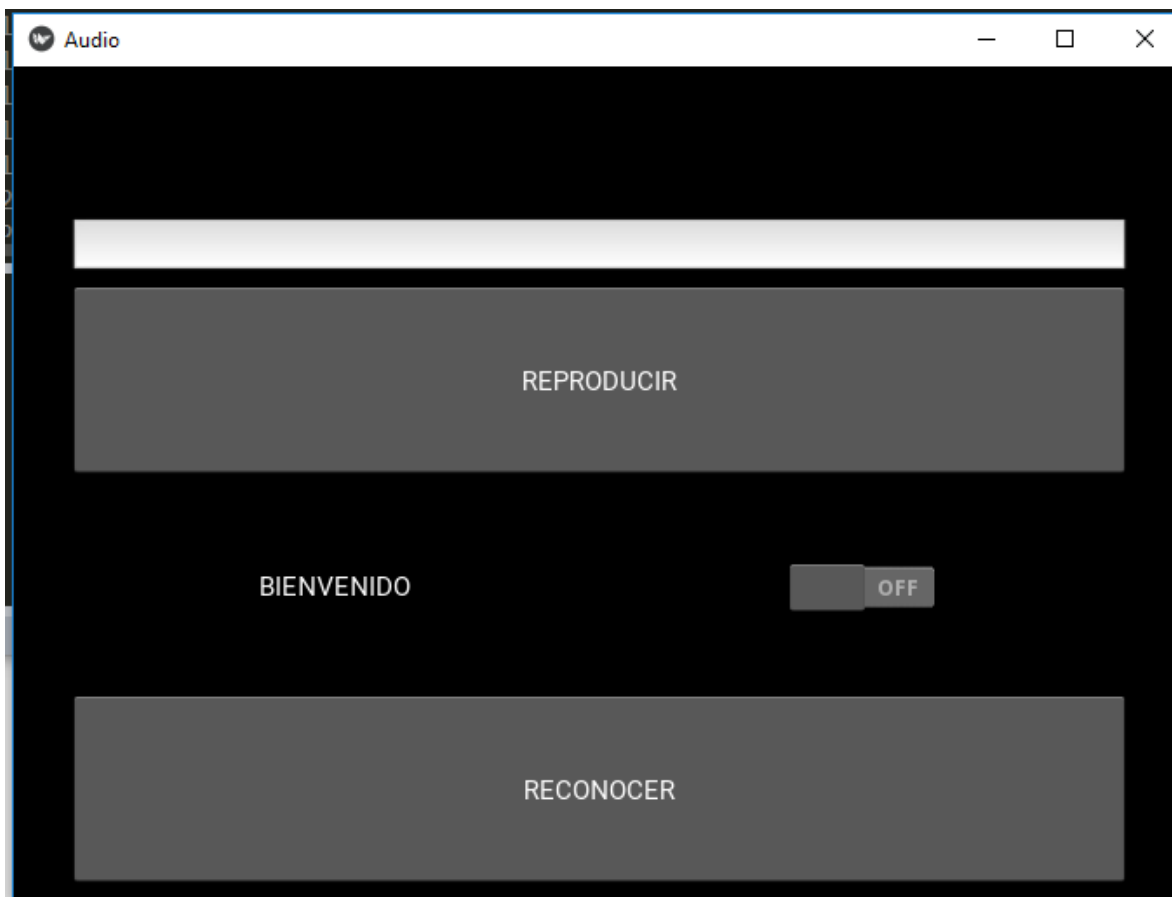
- Módulo PyAudio. Según (“PyAudio Documentation — PyAudio 0.2.11 documentation,” n.d.), PyAudio proporciona enlaces de Python para PortAudio, la biblioteca de E/S de audio multiplataforma. Con PyAudio, se puede usar Python fácilmente para reproducir y grabar audio en una variedad de plataformas. En SAH, se usa PyAudio para capturar la señal de voz del usuario A en un archivo de formato WAV.
- HTK Toolkit. (Young et al., 2002) nos dice que, HTK es un conjunto de herramientas para el diseño y desarrollo de Modelos Ocultos de Markov. Utilizamos las siguientes herramientas:
  - **HCOPY**. Usa como entrada los archivos, (1) análisis.conf, el cuál le indica los parámetros para realizar la codificación del archivo WAV a formato MFCC, y (2) codifica.scf, el cuál contiene el nombre con el que se creará el archivo codificado.

- **HVite.** Usa como entrada los archivos (1) los modelos acústicos previamente entrenados, (2) archivo codifica.scp, y (3) las opciones -p y -s establecen la penalización de inserción de palabras y el factor de escala de gramática, respectivamente. La penalización por inserción de palabras es un valor fijo agregado a cada token cuando transita desde el final de una palabra hasta el inicio de la siguiente. El factor de escala gramatical es la cantidad por la cual se escala la probabilidad del modelo de lenguaje antes de agregarla a cada token mientras transita desde el final de una palabra hasta el comienzo de la siguiente. Se genera el archivo MLF llamado recout.mlf, el cual contiene la transcripción del reconocimiento.
- **Módulo Subprocess.** Proporciona una interfaz consistente para crear y trabajar con procesos adicionales. Se utilizó la clase Popen, el constructor Popen toma argumentos para configurar el nuevo proceso para que el padre pueda comunicarse con él a través de las tuberías. Es usado en la implementación del sistema para ejecutar las herramientas binarias de HTK Toolkit.
- **Módulo Pyttsx.** Permite reproducir texto directamente desde el código de Python sin necesidad de ejecutar un comando externo. Es usado en la implementación para realizar la Síntesis de Texto a Voz de la palabra reconocida.
- **Kivy.** Es un framework Python de código abierto y plataforma para el desarrollo de aplicaciones que hacen uso de interfaces de usuario innovadoras y multitáctiles. Se usó para crear una GUI con un diseño de interacción fácil y rápido, para el práctico uso del usuario.



### 5.3 Interfaz Gráfica

En la Figura 13., se presenta la pantalla de la interfaz:



*Figura 15. Interfaz Gráfica del Sistema de ayuda al Habla*

La interfaz del Sistema de ayuda al Habla consta de 3 etapas: etapa de grabación, etapa de reconocimiento y etapa de síntesis.

#### 5.3.1 Etapa de Grabación

La etapa de grabación le permitirá al usuario realizar la captura de la señal de voz y archivarla en un archivo de formato WAV.

Está compuesta del Interruptor On/Off y el cronómetro que se encuentra a su lado izquierdo. El usuario deberá activar el interruptor, posicionándolo en

el estado ON, automáticamente se dará inicio al cronómetro y se empezará a grabar la señal de voz. Cuando el usuario haya terminado de hablar, deberá desactivar el interruptor, posicionándolo en el estado OFF. Automáticamente, se dará termino a la grabación y se creará el archivo WAV que contendrá la señal de audio capturada.

### **5.3.2. Etapa de Reconocimiento**

La etapa de reconocimiento, como lo dice su nombre, realiza el proceso de reconocimiento, ejecuta las herramientas HCopy y HVite de la biblioteca HTK Toolkit para obtener la transcripción de la palabra reconocida en el audio previamente grabado. Consta del botón RECONOCER, ubicado en la parte inferior de la interfaz, el cual, al ser pulsado, muestra en el área de texto, que se encuentra en la parte superior de la venta, la palabra reconocida.

### **5.3.2 Etapa de Síntesis**

La etapa de Síntesis se encarga de realizar la Síntesis de Texto a Voz, de la palabra mostrada en el área de texto. Si el usuario es incapaz de leer la palabra impresa o desea que sea reproducida, deberá pulsar el botón REPRODUCIR, el cual se encuentra ubicado debajo del área de texto donde se imprime la palabra reconocida.

### **5.3.3 Implementación**

Para el desarrollo del software se buscó un lenguaje que sea compatible con la biblioteca HTK Toolkit y además de que sea multiplataforma y libre acceso, con un nivel lo más cercano posible al sistema operativo para obtener un mejor rendimiento. Por estas razones se optó por implementar el prototipo en Python 2.7.14.

### 5.3.3.1. Script Record.py

Creada para realizar la grabación de la señal de audio del usuario, importa las librerías PyAudio, wave y threading. Define las variables globales que indican los parámetros que exige la biblioteca HTK Toolkit para la grabación de audios, tales como, número de canales y tasa de muestreo. Contiene la Clase *Record* (), en la cual se definen los siguientes métodos:

Método	Descripción
<code>__init__()</code>	Crea una instancia de PyAudio
<code>find_input_device()</code>	Busca un micrófono disponible para ser usado
<code>initMicrophone()</code>	Activa el micrófono seleccionado con las variables globales definidas
<code>stream_thread_new()</code>	Crea un hilo para ejecutar el subproceso de grabación
<code>stream_readchunk()</code>	Captura la señal de voz con la frecuencia dada
<code>stream_close()</code>	Detiene el proceso de grabación, crea el archivo WAV donde almacena la señal grabada.

### 5.3.3.2 Script Recognizer.py

Creada para realizar el proceso de reconocimiento. Utiliza las librerías subprocess, threading y re. Contiene la clase *Recognizer()* con los siguientes métodos:

Método	Descripción
<code>__init__()</code>	Inicia llamando al método <code>subprocess_thread_new()</code>
<code>subprocess_thread_new()</code>	Crea un hilo para ejecutar el subproceso de grabación
<code>recognize()</code>	Utiliza la clase Popen de la librería subprocess para ejecutar las herramientas HCopy y HVite de la biblioteca HTK Toolkit
<code>get_word()</code>	Utiliza la librería re para extraer la palabra reconocida del archivo MLF creado por la herramienta HVite

### 5.3.3.3 Script Main.py

Creada para implementar la interfaz gráfica del SAH, importa todas las librerías necesarias de la biblioteca Kivy, las clases Record y Recognizer, y la librería Pyttsx. Contiene dos clases:

- AudioApp (App): La clase base de la aplicación, contiene el método `build()`, esta es la función donde debe inicializar y devolver el widget raíz.
- AudioTool(Boxlayout):

Método	Descripción
<code>__init__()</code>	<p>Inicializa la clase pyttsx para la síntesis de texto a voz en español.</p> <p>Crea una instancia de la clase Record y llama al método para iniciar el micrófono.</p>
<code>start()</code>	<p>Inicia el cronómetro y empieza el proceso de grabación</p>
<code>stop()</code>	<p>Detiene el cronómetro y el proceso de grabación</p>
<code>updateDisplay()</code>	<p>Avanza el cronómetro segundo a segundo</p>
<code>callback()</code>	<p>Ejecuta <code>start()</code> o <code>stop()</code> de acuerdo a como cambie el estado del interruptor</p>
<code>reconocer()</code>	<p>Hace un llamado a la clase Recognizer y sus métodos.</p>
<code>listen()</code>	<p>Realizar la síntesis de texto a voz</p>

## **CAPITULO VI: RESULTADOS**

## **CAPITULO VI: RESULTADOS**

### **6.1 Participantes**

#### **6.1.1 Pacientes Disártrico**

##### **6.1.1.1 Paciente CG**

Señora CG, 93 años. CG ha sufrido derrame cerebral parcial del hemisferio izquierdo, dejándole como secuela una parálisis de la mitad derecha de todo su cuerpo. Recibió terapia física y del lenguaje durante los 3 primeros años de su enfermedad. Para su participación de este proyecto se platicó con ella y con los familiares para obtener el consentimiento de realizar las grabaciones de voz en su casa, ya que es su ambiente más habitual.

Al platicar con la paciente, se pudo percatar del vocabulario y/o la forma de interactuar con sus familiares y otras personas, al igual de la comodidad del paciente con la presencia del micrófono. También se le consultó al respecto de cómo se sentía anímica y físicamente FG, esto es muy importante, ya que todo trabajo con usuarios, especialmente si tienen alguna discapacidad, debe basarse en la confianza y comprensión de la persona.

Aunque CG y sus familiares aceptaron colaborar en la realización del proyecto, CG comenzó a perder la paciencia en la realización de las grabaciones por sus propias limitaciones físicas, mostrando molestia, irritación y dolor al realizarse las repeticiones de las frases del vocabulario. En este punto, al observar la situación de CG, se optó en cambiar las frases por palabras cortas que se le sean más cómodo repetir varias veces.

##### **6.1.1.2 Paciente ER**

Señor ER, 68 años, ER ha sufrido derrame cerebral parcial del hemisferio izquierdo, el cuál le originó una hemiplejía del lado derecho de todo su cuerpo. Recibió terapia física, pero de una manera inconsistente durante 2 años al

inicio de su enfermedad. Para su colaboración en este proyecto se platicó sólo con él y la persona encargada de él, debido a la falta de familiares, para obtener el consentimiento de realizar las grabaciones de voz en su casa, ya que es su ambiente habitual.

Al platicar con el paciente y su cuidador, se pudo observar que al principio el paciente no se sentía cómodo en presencia de micrófono y de personas extrañas, pero a medida que se realizaban las repeticiones, se fue adaptando.

#### **6.1.1.3 Paciente MS**

Señor MS, 65 años, MS ha sufrido derrame cerebral, el cuál le originó una parálisis total, pero a medida que avanzaba en sus terapias, pudo recuperar parte de la movilidad de su cuerpo, incluyendo la movilidad de las articulaciones del habla, por lo que se puede expresar con cierto grado de disartria.

Conversamos con el paciente y su hija, quienes accedieron a participar de este proyecto, el paciente mostró una buena disposición a realizar las repeticiones frente al micrófono y a realizar las pruebas del sistema.

#### **6.1.1.4 Paciente BD**

Señorita BD, 32 años, BD ha sufrido un traumatismo facial que le originó daños en los músculos motores del habla, indica que asistió un tiempo a terapias, pero no terminó el tratamiento. Conversamos con ella, para su participación en el proyecto, y accedió a realizarlo. Fue la única paciente que, por motivos personales, no se utilizó su casa como ambiente de grabación, si no, un ambiente público, pero sin mucho ruido de fondo. Se mostró cómoda frente al micrófono, otorgando una buena disposición para la realización del proyecto.



#### **6.1.1.5 Paciente MV**

Señor MV, 55 años, MV sufre la enfermedad de Parkinson, por la cuál se han degenerado los músculos y articulaciones responsables del habla. MV accedió a cooperar con el proyecto, sin brindar mucha información sobre su estado de salud. Se mostró cómodo frente al micrófono y frente a personas extrañas.

#### **6.1.2 Participantes Oyente**

Se decidió que los Participantes Oyente, sean personas ajenas a la familia o conocidos de los Participantes Paciente, ya que los familiares al estar constantemente con ellos ya han aprendido a identificar las palabras que dicen. La Señorita Sara Guevara Velez, Interna del Hospital Regional Docente Las Mercedes, colaboró como participante oyente en la fase “Antes del Sistema”, y el joven Henry Guevara Velez, Estudiante de Ingeniería Civil de la UNPRG, colaboró como participante oyente en la fase “Después del Sistema”.

### **6.2 FASE I: ANTES DE SAH**

#### **6.2.1 Descripción del Escenario**

Los pacientes disártricos emitirán una por una las palabras del vocabulario para que el participante Oyente, Sara Guevara, las escuche y trate de interpretarlas. Por cada palabra se medirá la fluidez, pronunciación e interacción de la conversación.

## 6.2.2 Pruebas

### 6.2.2.1 Paciente CG

<b>PALABRAS</b>	<b>FLUIDEZ</b>	<b>PRONUNCIACIÓN</b>	<b>INTERACCIÓN</b>	<b>NIVEL DE COMUNICACIÓN</b>
Columba	2	2	2	6
Gonzales	1	1	2	4
Elefante	2	2	2	6
Cabeza	2	2	1	5
Micrófono	1	1	1	3

Como se observa en la matriz de resultados, después de usar la Escala Analítica en Expresión e Interacción Orales de los Exámenes DELE, el puntaje de cada palabra se encuentra muy por debajo del puntaje total. Por lo que, podemos decir que el NIVEL DE COMUNICACIÓN PROMEDIO ES DE 5 PUNTOS en el paciente CG.

### 6.2.2.2 Paciente ER

<b>PALABRAS</b>	<b>FLUIDEZ</b>	<b>PRONUNCIACIÓN</b>	<b>INTERACCIÓN</b>	<b>NIVEL DE COMUNICACIÓN</b>
Columba	2	1	1	4
Gonzales	2	3	2	7
Elefante	2	2	2	6
Cabeza	3	3	3	9
Micrófono	1	1	1	3

Como se observa en la matriz de resultados, después de usar la Escala Analítica en Expresión e Interacción Orales de los Exámenes DELE, el puntaje de 3 de las 5 palabras es menor o igual a la mitad del puntaje total. Por lo que, podemos decir que el NIVEL DE COMUNICACIÓN PROMEDIO ES DE 5.8 PUNTOS.

### 6.2.2.3 Paciente MS

PALABRAS	FLUIDEZ	PRONUNCIACIÓN	INTERACCIÓN	NIVEL DE COMUNICACIÓN
Columba	2	1	1	4
Gonzales	2	1	1	4
Elefante	2	1	1	4
Cabeza	2	2	2	6
Micrófono	2	2	3	7

Como se observa en la matriz de resultados, después de usar la Escala Analítica en Expresión e Interacción Orales de los Exámenes DELE, solo en una palabra se logra pasar por encima de la mitad del puntaje total. Por lo que, podemos decir que el NIVEL DE COMUNICACIÓN PROMEDIO ES DE 5 PUNTOS.

### 6.2.2.4 Paciente BD

PALABRAS	FLUIDEZ	PRONUNCIACIÓN	INTERACCIÓN	NIVEL DE COMUNICACIÓN
Columba	2	2	2	6
Gonzales	2	2	2	6
Elefante	2	2	2	6
Cabeza	3	3	3	9
Micrófono	2	2	2	6

Como se observa en la matriz de resultados, después de usar la Escala Analítica en Expresión e Interacción Orales de los Exámenes DELE, el puntaje de 4 palabras logra llegar a la mitad del puntaje total y una sola palabra lo logra superar. Por lo que, podemos decir que el NIVEL DE COMUNICACIÓN PROMEDIO ES DE 6.6 PUNTOS.

#### 6.2.2.2 Paciente MV

<b>PALABRAS</b>	<b>FLUIDEZ</b>	<b>PRONUNCIACIÓN</b>	<b>INTERACCIÓN</b>	<b>NIVEL DE COMUNICACIÓN</b>
Columba	3	1	1	5
Gonzales	2	2	2	6
Elefante	2	2	2	6
Cabeza	3	1	1	5
Micrófono	2	1	1	4

Como se observa en la matriz de resultados, después de usar la Escala Analítica en Expresión e Interacción Orales de los Exámenes DELE, el puntaje de todas las palabras es menor o igual a la mitad del puntaje total. Por lo que, podemos decir que el NIVEL DE COMUNICACIÓN PROMEDIO ES DE 5.2 PUNTOS.

#### 6.2.3 Observaciones

Mientras se realizaban las pruebas se fueron obteniendo las siguientes observaciones en la comunicación de los pacientes:

- Los pacientes tenían como un límite de repeticiones por palabra un promedio de 3 veces, repetir más veces los llevaba a la frustración por no ser entendidos y se negaban a seguir intentándolo. Truncando el proceso de comunicación.
- Muchas veces la oyente tenía que interpretar adivinando o tanteando en palabras similares a las que escuchaba hasta que diera con la palabra correcta.
- Los pacientes se quejaban de dolor al tener que articular con exageración o tener que aumentar el volumen de su voz para lograr ser entendidos.

## 6.3 FASE II: DESPUÉS DEL SAH

### 6.3.1 Descripción del Escenario

Los pacientes emitirán una por una las palabras del vocabulario, estas primero serán reconocidas por el sistema, y el participante oyente, Henry Guevara, escuchará lo emitido por el SAH. Por cada palabra se medirá la fluidez, pronunciación e interacción de la conversación. Cada palabra se repetirá 4 veces, el puntaje interacción será puesto de acuerdo con la cantidad de veces que el sistema reconoce correctamente las palabras. El puntaje de Fluidez y Pronunciación siempre será de 4 puntos, ya que el sistema realiza la síntesis de Voz de una persona sin ninguna discapacidad del habla.

### 6.3.2 Pruebas

#### 6.3.2.1 Paciente CG

PALABRAS	FLUIDEZ	PRONUNCIACIÓN	INTERACCIÓN	NIVEL DE COMUNICACIÓN
Columba	4	4	4	12
Gonzales	4	4	2	10
Elefante	4	4	3	11
Cabeza	4	4	4	12
Micrófono	4	4	2	10

Como se observa en la matriz de resultados después de usar la Escala Analítica de Expresión e Interacción Orales de los Exámenes DELE, el puntaje de las palabras son iguales y/o muy cercanas al puntaje total. Por lo que, podemos decir que, el NIVEL DE COMUNICACIÓN PROMEDIO ES DE 11 PUNTOS.

### 6.3.2.2 Paciente ER

<b>PALABRAS</b>	<b>FLUIDEZ</b>	<b>PRONUNCIACIÓN</b>	<b>INTERACCIÓN</b>	<b>NIVEL DE COMUNICACIÓN</b>
Columba	4	4	2	10
Gonzales	4	4	4	12
Elefante	4	4	4	12
Cabeza	4	4	3	11
Micrófono	4	4	2	10

Como se observa en la matriz de resultados después de usar la Escala Analítica de Expresión e Interacción Orales de los Exámenes DELE, el puntaje de las palabras son iguales y/o muy cercanas al puntaje total. Por lo que, podemos decir que, el NIVEL DE COMUNICACIÓN PROMEDIO ES DE 11 PUNTOS.

### 6.3.2.3 Paciente MS

<b>PALABRAS</b>	<b>FLUIDEZ</b>	<b>PRONUNCIACIÓN</b>	<b>INTERACCIÓN</b>	<b>NIVEL DE COMUNICACIÓN</b>
Columba	4	4	2	10
Gonzales	4	4	2	10
Elefante	4	4	3	11
Cabeza	4	4	3	11
Micrófono	4	4	4	10

Como se observa en la matriz de resultados después de usar la Escala Analítica de Expresión e Interacción Orales de los Exámenes DELE, el puntaje de las palabras son iguales y/o muy cercanas al puntaje total. Por lo que, podemos decir que, el NIVEL DE COMUNICACIÓN PROMEDIO ES DE 10.2 PUNTOS.

#### 6.3.2.4 Paciente BD

PALABRAS	FLUIDEZ	PRONUNCIACIÓN	INTERACCIÓN	NIVEL DE COMUNICACIÓN
Columba	4	4	4	12
Gonzales	4	4	4	12
Elefante	4	4	4	12
Cabeza	4	4	4	12
Micrófono	4	4	4	12

Como se observa en la matriz de resultados después de usar la Escala Analítica de Expresión e Interacción Orales de los Exámenes DELE, el puntaje de las palabras son iguales y/o muy cercanas al puntaje total. Por lo que, podemos decir que, el NIVEL DE COMUNICACIÓN PROMEDIO ES DE 12 PUNTOS.

#### 6.3.2.5 Paciente MV

PALABRAS	FLUIDEZ	PRONUNCIACIÓN	INTERACCIÓN	NIVEL DE COMUNICACIÓN
Columba	4	4	3	11
Gonzales	4	4	4	12
Elefante	4	4	4	12
Cabeza	4	4	3	11
Micrófono	4	4	3	11

Como se observa en la matriz de resultados después de usar la Escala Analítica de Expresión e Interacción Orales de los Exámenes DELE, el puntaje de las palabras son iguales y/o muy cercanas al puntaje total. Por lo que, podemos decir que, el NIVEL DE COMUNICACIÓN PROMEDIO ES DE 11.4 PUNTOS.

### 6.3.3 Observaciones

Mientras se realizaban las pruebas se fueron obteniendo las siguientes observaciones en la comunicación de los pacientes:

- Los pacientes mostraron mucha afinidad con el uso del SAH, se sentían más motivados a intentar emitir más palabras.
- La interacción con el oyente fue mayor, ya que tomaba menos tiempo para el oyente entender lo que el paciente le decía.
- No hubo ningún tipo de frustración por parte de los pacientes, por lo que en ningún caso la comunicación fue interrumpida.

### 6.4 Contrastación de Fases

El Puntaje Total de la Escala Analítica de Expresión e Interacción Orales de los Exámenes DELE, es de 12 puntos. En la siguiente tabla, se muestra la variación de resultados, en las dos fases: antes y después de la implementación del Sistema de Ayuda al Habla para Personas Disártricas:

	Nivel de Comunicación	
	Antes de SAH	Después de SAH
Paciente CG	41.67%	91.67%
Paciente ER	48.33%	91.67%
Paciente MS	41.67%	85.00%
Paciente BD	55.00%	100.00%
Paciente MV	43.33%	95.00%

Como se observa en la tabla, los porcentajes muestran que todos los pacientes presentaron un mayor nivel de comunicación, después de la implementación del Sistema de Ayuda al Habla.



El nivel de Comunicación Promedio en la fase “Antes de SAH” es de 46%, pero con la implementación del Sistema de Ayuda al Habla, esta cifra en la fase “Después de SAH” aumentó a 92.67%, mostrando una mejora del 46.67%.

## 6.5 Tasa de Reconocimiento del Sistema

Se utiliza la siguiente fórmula para calcular la Tasa de reconocimiento de un Sistema de Reconocimiento de Voz automático:

$$Tasa\ de\ Reconocimiento = \frac{\#Palabras\ correctamente\ reconocidas}{\#Total\ de\ Palabras} \times 100\%$$

Mostramos la siguiente tabla, con la cantidad de palabras correcta e incorrectamente reconocidas por el sistema, con cada paciente:

	Correctos	Incorrectos
Paciente CG	15	5
Paciente ER	15	5
Paciente MS	14	6
Paciente BD	20	0
Paciente MV	17	3
<b>Total</b>	<b>81</b>	<b>19</b>

Al aplicar la fórmula anterior, se calcula una tasa de reconocimiento del 81%, el cuál es un porcentaje cercano a las tasas de reconocimiento de otros Sistemas de Reconocimiento de Voz Comerciales.

## **CAPITULO VII: CONCLUSIONES**

## CAPITULO VII: CONCLUSIONES

En los capítulos anteriores, se explicó el desarrollo de esta tesis y se mostró cómo se fueron logrando uno a uno los objetivos. El “Desarrollo de un Sistema de Ayuda al Habla para mejorar el nivel de comunicación entre una persona disártrica y una persona sin discapacidad”, fue el objetivo general que engloba este proyecto.

Se logró probar la hipótesis, EL NIVEL DE COMUNICACIÓN TUVO UN AUMENTO DE 46.67% PUNTOS, ES DECIR, TUVO UN INCREMENTO DEL 46% AL 92.67% DE LA ESCALA ANÁLITICA DE LOS INTERACCIÓN Y EXPRESIÓN ORAL DE LOS EXAMENES DELE, DESPUÉS DE LA IMPLEMENTACIÓN DEL SISTEMA DE AYUDA AL HABLA.

El SAH, consta de los modelos acústicos, modelo de lenguaje (ML), corpus textual/oral, diccionario fonético, y algoritmo de búsqueda (decodificación). La interfaz involucra el medio para la administración y configuración de estos elementos (en tiempo de ejecución y de manera automática) para cubrir las funciones de comunicación del usuario con disartria.

Para el desarrollo del SAH se abordó el enfoque de Dependiente de Usuario (DU), en los trabajos citados se ha argumentado que el desarrollo de SRAH's dependientes de Usuario son mejores para usuarios con disartria, aunque requieren más tiempo y trabajo de cerca con el paciente para ser desarrollados. Esto porque este enfoque implica el desarrollar un corpus de entrenamiento con la voz del usuario que va a usar el sistema, el cual debe estar etiquetado correctamente.

De esta manera se considera este sistema, con las funciones implementadas, contribuye al campo del desarrollo de tecnologías para personas con discapacidad en el habla. Siendo que este campo no se ha explorado de manera significativa para el español latino.

## **RECOMENDACIONES:**

Se considera los siguientes puntos como recomendaciones para trabajo futuro:

- Extender el diseño gráfico y usabilidad del módulo para comunicación de voz disártrica.
- Evaluar el rendimiento del sistema con un vocabulario mucho más extenso.
- Adaptación del sistema para su implementación e dispositivos móviles.
- Explorar el enfoque Independiente de Usuario para llegar a más usuarios.
- Incrementar el tamaño del corpus de entrenamiento con usuarios disártricos de diferentes edades y ambos géneros.
- Se recomienda para futuros trabajos en el tema, investigar Sistemas de autoaprendizaje con modelos híbridos de Modelos Ocultos de Markov y redes neuronales

## **BIBLIOGRAFÍA**

## BIBLIOGRAFÍA

- Adams, C. (2017, March 17). Explaining the Different Types of Voice Recognition. Retrieved November 30, 2017, from <https://www.thoughtco.com/types-of-voice-recognition-1205856>
- Álvarez, A. (2001). *Apuntes de Fundamentos del Reconocimiento Automático de la Voz*. Madrid: Departamento de Arquitectura y Tecnología de Sistemas Informáticos. Facultad de Informática. UPM.
- Davis, S., & Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4), 357–366. <https://doi.org/10.1109/TASSP.1980.1163420>
- Hatzis, A. (2003). Automatic speech recognition with sparse training data for dysarthric speakers. *Proc. European Conference on Speech Communication Technology*, 1189–1192.
- Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. r, Jaitly, N., ... Kingsbury, B. (2012). Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. *IEEE Signal Processing Magazine*, 29(6), 82–97. <https://doi.org/10.1109/MSP.2012.2205597>
- Jurafsky, D., & Martin, J. H. (2008). *Speech and Language Processing, 2nd Edition* (2nd edition). Upper Saddle River, N.J: Prentice Hall.

Kadi, K. L., Selouani, S. A., Boudraa, B., & Boudraa, M. (2016). Fully automated speaker identification and intelligibility assessment in dysarthria disease using auditory knowledge. *Biocybernetics and Biomedical Engineering*, 36(1), 233–247.

Lambayeque: Más de 100 mil tratamientos en medicina física se realizaron en hospital. (2015, October 26). Retrieved January 20, 2018, from <https://diariocorreo.pe/edicion/lambayeque/lambayeque-mas-de-100-mil-tratamientos-en-medicina-fisica-se-realizaron-en-hospital-628286/>

Lizandra Laplaza, R. (n.d.). Dificultades en el Desarrollo del Lenguaje Oral e Intervención. Retrieved November 26, 2017, from <https://www.yumpu.com/es/document/view/14250103/dificultades-en-el-desarrollo-del-lenguaje-oral-e-intervencion/14>

Maggiolo, M. (2017). Test de articulación a la repetición (TAR): un legado de la profesora fonoaudióloga Edith Schwalm. *Revista Chilena de Fonoaudiología*, 16. <https://doi.org/10.5354/0719-4692.2017.47557>

Moriana, M. J. (2009). La Disartria. *Revista Digital Innivación Y Experiencias Educativas*. Retrieved from <https://fonologizatte.wordpress.com/trastornos-del-habla/>

Peña-Casanova, J. (2013). *Manual de logopedia*. Amsterdam; Barcelona: Elsevier.

Pérez, A. M. S., Fernández, M. D. V., & Torres, I. H. (2006). La comunicación oral, sus características generales. *Ciencias Holguín*, XII(2), 1–6.

PyAudio Documentation — PyAudio 0.2.11 documentation. (n.d.). Retrieved January 14, 2018, from <https://people.csail.mit.edu/hubert/pyaudio/docs/>

Qué son los DELE | Exámenes - Instituto Cervantes. (n.d.). Retrieved January 21, 2018, from <https://exámenes.cervantes.es/es/dele/que-es>

Samira, H., Fateh, B., Smaine, M., & Mohamed, B. (2013). A novel speech recognition approach based on multiple modeling by hidden Markov models. In *2013 International Conference on Computer Applications Technology (ICCAT)* (pp. 1–6). <https://doi.org/10.1109/ICCAT.2013.6522028>

Sanchez, M. G. (n.d.). Desórdenes Motores del Habla y PROMPT (Parte II). *EspacioLogopedico.com*. Retrieved from <https://www.espaciologopedico.com/revista/articulo/1692/desordenes-motores-del-habla-y-prompt-parte-ii.html>

Tamayo, M. T. y. (2004). *El proceso de la investigación científica*. Editorial Limusa.

Young, S., Evermann, G., Gales, M. J. F., Hain, T., Kershaw, D., Liu, X., ... Woodland, P. (2002). *The HTK book*.



## **ANEXOS**

## **ANEXO A: Archivo prompts.txt**

\*/0101 COLUMBA  
\*/0102 COLUMBA  
\*/0103 COLUMBA  
\*/0104 COLUMBA  
\*/0105 COLUMBA  
\*/0106 COLUMBA  
\*/0107 COLUMBA  
\*/0108 COLUMBA  
\*/0109 COLUMBA  
\*/0110 COLUMBA  
\*/0201 GONZALES  
\*/0202 GONZALES  
\*/0203 GONZALES  
\*/0204 GONZALES  
\*/0205 GONZALES  
\*/0206 GONZALES  
\*/0207 GONZALES  
\*/0208 GONZALES  
\*/0209 GONZALES  
\*/0210 GONZALES  
\*/0301 MICROFONO  
\*/0302 MICROFONO  
\*/0303 MICROFONO  
\*/0304 MICROFONO  
\*/0305 MICROFONO  
\*/0306 MICROFONO

\*/0307 MICROFONO  
\*/0308 MICROFONO  
\*/0309 MICROFONO  
\*/0310 MICROFONO  
\*/0401 ELEFANTE  
\*/0402 ELEFANTE  
\*/0403 ELEFANTE  
\*/0404 ELEFANTE  
\*/0405 ELEFANTE  
\*/0406 ELEFANTE  
\*/0407 ELEFANTE  
\*/0408 ELEFANTE  
\*/0409 ELEFANTE  
\*/0410 ELEFANTE  
\*/0501 CABEZA  
\*/0502 CABEZA  
\*/0503 CABEZA  
\*/0504 CABEZA  
\*/0505 CABEZA  
\*/0506 CABEZA  
\*/0507 CABEZA  
\*/0508 CABEZA  
\*/0509 CABEZA  
\*/0510 CABEZA

## ANEXO B: Archivo prompts2wlist.pl

```
1  #!/usr/local/bin/perl -w
2  use strict;
3
4  my ($prompts, $wlist, $line, @line_array, $w, %words, @word_list);
5
6  # check usage
7  if (@ARGV != 2) {
8      print "usage: $0 prompts wordlist\n\n";
9      exit (0);
10 }
11
12 # read in command line arguments
13 ($prompts, $wlist) = @ARGV;
14
15 # open files
16 open (PROMPTS, "$prompts") || die ("Unable to open prompts $prompts
file for reading");
17 open (WLIST, ">$wlist") || die ("Unable to open word list $wlist file
for writing");
18
19 # process each prompt one at a time
20 while ($line = <PROMPTS>) {
21     chomp ($line);
22     @line_array = split(/\s+/, $line);
23     shift (@line_array); # discard fileID
24
25     foreach $w (@line_array) {
26         $words{$w} = 1;
27     }
28 }
29
30 # put the words in order
31 @word_list = sort (keys (%words));
32
33 #and finally write out to word list file
34 foreach $w (@word_list) {
35     printf(WLIST "%s\n", $w);
36 }
37
38 close(WLIST);
39 close(PROMPTS);
```

## ANEXO C: Archivo wlist

1	CABEZA
2	COLUMBA
3	ELEFANTE
4	GONZALES
5	MICROFONO
6	SILENCE
7	

## ANEXO D: Archivo prompts2mlf.pl

```
1  #!/usr/bin/perl -w
2  use strict;
3  my ($mlf, $prompt, $line, $fname, @labs, $w);
4
5  # this script makes an mlf out of a list of file names and
6  # corresponding prompts - ie in the format
7  # fileid prompt
8  # fileid prompt
9  # "      "
10 # The prompts are automatically converted to upper case.
11
12 if (@ARGV != 2) {
13     print "usage: $0 mlf promptlist\n\n";
14     exit (0);
15 }
16
17 # read in command line arguments
18 ($mlf, $prompt) = @ARGV;
19
20 # open MLF file
21 open (MLF, ">$mlf") || die ("Unable to open mlf $mlf file for writing");
22
23 print "writing to mlf file $mlf\n";
24
25 print MLF ("\#!MLF!\#\n");
26 # open prompt file
27 open (LAB, "$prompt") || die ("Unable to open prompt file $prompt");
28 while ($line = <LAB>) {
29     chomp ($line);
30     ($fname, @labs) = split(/\s+/, $line);
31     $fname =~ s/\.mfc//g;
32     $fname =~ s/\.lab//g;
33     print MLF ("\$fname.lab\n\n");
34
35     foreach $w (@labs) {
36         printf(MLF "%s\n", $w);
37     }
38     print MLF (".\n");
39 }
40
41 close (LAB);
42 close(MLF);
43 print "writing to $mlf file done\n";
```

## ANEXO E: Archivo words.mlf

#!MLF!#	GONZALES	"*/0404.lab"
"*/0101.lab"	.	ELEFANTE
COLUMBA	"*/0208.lab"	.
.	GONZALES	"*/0405.lab"
"*/0102.lab"	.	ELEFANTE
COLUMBA	"*/0209.lab"	.
.	GONZALES	"*/0406.lab"
"*/0103.lab"	.	ELEFANTE
COLUMBA	"*/0210.lab"	.
.	GONZALES	"*/0407.lab"
"*/0104.lab"	.	ELEFANTE
COLUMBA	"*/0301.lab"	.
.	MICROFONO	"*/0408.lab"
"*/0105.lab"	.	ELEFANTE
COLUMBA	"*/0302.lab"	.
.	MICROFONO	"*/0409.lab"
"*/0106.lab"	.	ELEFANTE
COLUMBA	"*/0303.lab"	.
.	MICROFONO	"*/0410.lab"
"*/0107.lab"	.	ELEFANTE
COLUMBA	"*/0304.lab"	.
.	MICROFONO	"*/0501.lab"
"*/0108.lab"	.	CABEZA
COLUMBA	"*/0305.lab"	.
.	MICROFONO	"*/0502.lab"
"*/0109.lab"	.	CABEZA
COLUMBA	"*/0306.lab"	.
.	MICROFONO	"*/0503.lab"
"*/0110.lab"	.	CABEZA
COLUMBA	"*/0307.lab"	.
.	MICROFONO	"*/0504.lab"
"*/0201.lab"	.	CABEZA
GONZALES	"*/0308.lab"	.
.	MICROFONO	"*/0505.lab"
"*/0202.lab"	.	CABEZA
GONZALES	"*/0309.lab"	.
.	MICROFONO	"*/0506.lab"
"*/0203.lab"	.	CABEZA
GONZALES	"*/0310.lab"	.
.	MICROFONO	"*/0507.lab"
"*/0204.lab"	.	CABEZA
GONZALES	"*/0401.lab"	.
.	ELEFANTE	"*/0508.lab"
"*/0205.lab"	.	CABEZA
GONZALES	"*/0402.lab"	.
.	ELEFANTE	"*/0509.lab"
"*/0206.lab"	.	CABEZA
GONZALES	"*/0403.lab"	.
.	ELEFANTE	"*/0510.lab"
"*/0207.lab"	.	CABEZA

## ANEXO F: Archivo codetrain.scf

datos/sig/0101.wav	datos/mfcc/0101.mfcc
datos/sig/0102.wav	datos/mfcc/0102.mfcc
datos/sig/0103.wav	datos/mfcc/0103.mfcc
datos/sig/0104.wav	datos/mfcc/0104.mfcc
datos/sig/0105.wav	datos/mfcc/0105.mfcc
datos/sig/0110.wav	datos/mfcc/0110.mfcc
datos/sig/0201.wav	datos/mfcc/0201.mfcc
datos/sig/0202.wav	datos/mfcc/0202.mfcc
datos/sig/0203.wav	datos/mfcc/0203.mfcc
datos/sig/0204.wav	datos/mfcc/0204.mfcc
datos/sig/0205.wav	datos/mfcc/0205.mfcc
datos/sig/0301.wav	datos/mfcc/0301.mfcc
datos/sig/0302.wav	datos/mfcc/0302.mfcc
datos/sig/0303.wav	datos/mfcc/0303.mfcc
datos/sig/0304.wav	datos/mfcc/0304.mfcc
datos/sig/0305.wav	datos/mfcc/0305.mfcc
datos/sig/0306.wav	datos/mfcc/0306.mfcc
datos/sig/0401.wav	datos/mfcc/0401.mfcc
datos/sig/0402.wav	datos/mfcc/0402.mfcc
datos/sig/0403.wav	datos/mfcc/0403.mfcc
datos/sig/0404.wav	datos/mfcc/0404.mfcc
datos/sig/0405.wav	datos/mfcc/0405.mfcc
datos/sig/0501.wav	datos/mfcc/0501.mfcc
datos/sig/0502.wav	datos/mfcc/0502.mfcc
datos/sig/0503.wav	datos/mfcc/0503.mfcc
datos/sig/0504.wav	datos/mfcc/0504.mfcc
datos/sig/0505.wav	datos/mfcc/0505.mfcc



## **ANEXO G: Archivo analisis.conf**

SOURCEFORMAT = WAV

TARGETKIND = MFCC\_0\_D\_A

WINDOWSIZE = 250000.0

TARGETRATE = 100000.0

NUMCEPS = 12

USEHAMMING = T

PREEMCOEF = 0.97

NUMCHANS = 26

CEPLIFTER = 22

## ANEXO H: Archivo hmmdefs

```

1 ~h "cabeza"
2 <BEGINHMM>
3     <NUMSTATES> 5
4     <STATE> 2
5         <MEAN> 39
6         6.215101e+000 -9.747718e+000 -1.323526e+001 -5.123218e+000 -9
7         <VARIANCE> 39
8         9.168633e+000 3.199880e+001 1.028013e+001 3.682796e+001 3.797
9         <GCONST> 1.151191e+002
10    <STATE> 3
11        <MEAN> 39
12        1.849758e+000 -8.262193e+000 -1.198433e+001 -6.245712e+000 -1
13        <VARIANCE> 39
14        9.244641e+000 2.182999e+001 1.775698e+001 2.891289e+001 2.846
15        <GCONST> 9.030519e+001
16    <STATE> 4
17        <MEAN> 39
18        5.514269e+000 -4.458220e+000 -8.038857e+000 1.605698e+000 -9.
19        <VARIANCE> 39
20        1.312963e+001 4.487305e+001 1.974440e+001 2.829346e+001 6.295
21        <GCONST> 1.170537e+002
22    <TRANSP> 5
23        0.000000e+000 1.000000e+000 0.000000e+000 0.000000e+000 0.000000e
24        0.000000e+000 9.412814e-001 5.871854e-002 0.000000e+000 0.000000e
25        0.000000e+000 0.000000e+000 9.723546e-001 2.764539e-002 0.000000e
26        0.000000e+000 0.000000e+000 0.000000e+000 9.519166e-001 4.808337e
27        0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e
28 <ENDHMM>
29 ~h "columba"
30 <BEGINHMM>
31     <NUMSTATES> 5
32     <STATE> 2
33         <MEAN> 39
34         6.530310e+000 -2.943567e+000 -1.091871e+001 -8.461262e+000 -8
35         <VARIANCE> 39
36         1.331445e+001 3.002930e+001 1.704093e+001 2.640816e+001 1.781
37         <GCONST> 1.159326e+002
38     <STATE> 3
39         <MEAN> 39
40         6.472068e+000 -4.305195e+000 -1.484663e+001 -1.408695e+001 -5

```

## ANEXO I: Archivo vFloors

```
1 ~v varFloor1
2 <Variance> 39
3 4.311925e-001 6.704976e-001 3.887933e-001 5.279268e-001
  5.892228e-001 6.341548e-001 6.376315e-001 7.791879e-001
  8.040231e-001 6.689551e-001 4.798837e-001 4.132726e-001
  6.594285e-001 1.058604e-002 2.522819e-002 1.892456e-002
  2.311118e-002 2.573603e-002 2.874618e-002 3.519536e-002
  3.848452e-002 3.993808e-002 3.472049e-002 2.906679e-002
  2.277890e-002 1.601414e-002 1.653864e-003 3.255786e-003
  3.144490e-003 3.949583e-003 4.441103e-003 4.952125e-003
  5.980016e-003 6.769460e-003 6.999257e-003 6.213301e-003
  5.401439e-003 4.302710e-003 2.001975e-003
```

4

## ANEXO J: Archivo config2.conf

```
1  SOURCERATE=625.0
2  SOURCEKIND=HAUDIO
3  SOURCEFORMAT=HTK
4  ENORMALISE=F
5  MEASURESIL=F
6  OUTSILWARN=T
7  AUDIOSIG=-1
8
9  TARGETKIND = MFCC_0_D_A
10 WINDOWSIZE = 250000.0
11 TARGETRATE = 100000.0
12
```

## ACTA DE SUSTENTACIÓN.

En la sala de sustentaciones de la Facultad de Ingeniería Civil de Sistemas y de Arquitectura, siendo las 10:30 am. del día 18 de Abril del 2018, se reunieron los miembros del Informe de Tesis titulada: "MODELOS OCULTOS DE MARKOV PARA EL DESARROLLO DE UN SISTEMA DE AYUDA AL HABLA PARA PERSONAS QUE SUFREN DE DISORTIA", conformado por los siguientes docentes:

M.Sc. Ing. Gilberto Martín Ampuero Pasco	Presidente.
Ing. Omar Wilton Sarmendra Salazar	Miembro
Mg. Ing. Jesús Bernardo Obarría Paz	Miembro
Mg. Ing. Juan Elías Villegas Cubas	Procurador

Actuando como presidente el M.Sc. Ing. Gilberto Martín Ampuero Pasco y como secretario el Ing. Omar Wilton Sarmendra Salazar, se procedió a recepcionar la sustentación del Informe de tesis a cargo de la bachiller:  
CLARA MERCEDES QUEVEDO VELEZ.

Durante la sustentación se procedió a realizar las preguntas que consideró pertinentes, las cuales fueron absorbidas por la sustentante.

Concluida la sustentación, el jurado procedió a deliberar libre y reservadamente acordando **APROBADO** con el calificativo de **MUY BUENO** y nota 18.

Finalmente se dio lectura a la presente acta, firmando en señal de conformidad los docentes que intervinieron, siendo las 12:30.

M. Sc. Ing. GILBERTO MARTÍN AMPUERO PASCO  
Presidente del Jurado

Ing. OMAR WILTON SARMIENDA SALAZAR  
Miembro del Jurado.

Mg. Ing. JESÚS BERNARDO OBARRÍA PAZ  
Miembro del Jurado.

Mg. Ing. JUAN ELÍAS VILLEGAS CUBAS  
Procurador.



“Año de la universalización de la salud”.

**CONSTANCIA DE APROBACION DE ORIGINALIDAD DE TESIS**

Según Res. N° 659-2020-R

Yo, Juan Elias, Villegas Cubas , **asesor de tesis de la bachiller:**

**GUEVARA VELEZ, CLARA MERCEDES**

**TITULADA:**

MODELOS OCULTOS DE MARKOV PARA EL DESARROLLO DE UN SISTEMA DE AYUDA AL HABLA  
PARA PERSONAS QUE SUFREN DE DISARTRIA

Luego de la revisión exhaustiva del documento constato que la misma tiene un índice de similitud de **18%** verificable en el reporte de similitud del programa TURNITIN.

El suscrito analizó dicho reporte y concluyó que cada una de las coincidencias detectadas NO CONSTITUYEN PLAGIO. A mi leal saber y entender la tesis cumple con todas las normas para el uso de citas y referencias establecidas por la Universidad Nacional Pedro Ruiz Gallo.

Se expide la presente según lo dispuesto en la Resolución N° 659-2020-R, para la obtención de Grados y Títulos de la UNPRG:

Lambayeque, 15 de julio del 2022

ATENTAMENTE,

Ing. Juan Elías Villegas Cubas  
DNI. 80103991

Se adjunta:

Recibo digital de Turnitin

Revisión de informe en Turnitin

# MODELOS OCULTOS DE MARKOV PARA EL DESARROLLO DE UN SISTEMA DE AYUDA AL HABLA PARA PERSONAS QUE SUFREN DE DISARTRIA

## INFORME DE ORIGINALIDAD

18%	17%	1%	8%
INDICE DE SIMILITUD	FUENTES DE INTERNET	PUBLICACIONES	TRABAJOS DEL ESTUDIANTE

## FUENTES PRIMARIAS

1	<a href="http://scielo.conicyt.cl">scielo.conicyt.cl</a> Fuente de Internet	1%
2	<a href="http://foro.univision.com">foro.univision.com</a> Fuente de Internet	1%
3	<a href="http://cableroto.blogspot.com">cableroto.blogspot.com</a> Fuente de Internet	1%
4	<a href="http://www.redalyc.org">www.redalyc.org</a> Fuente de Internet	1%
5	<a href="http://repositorio.unprg.edu.pe">repositorio.unprg.edu.pe</a> Fuente de Internet	1%
6	Submitted to Universidad de Lima Trabajo del estudiante	1%
7	<a href="http://alicia.concytec.gob.pe">alicia.concytec.gob.pe</a> Fuente de Internet	1%
8	Submitted to Víaa Trabajo del estudiante	1%

9	<a href="http://digibug.ugr.es">digibug.ugr.es</a> Fuente de Internet	1 %
10	<a href="http://es.wikipedia.org">es.wikipedia.org</a> Fuente de Internet	1 %
11	<a href="http://myslide.es">myslide.es</a> Fuente de Internet	1 %
12	<a href="http://repositorio.unicauca.edu.co:8080">repositorio.unicauca.edu.co:8080</a> Fuente de Internet	1 %
13	<a href="http://stutzartists.org">stutzartists.org</a> Fuente de Internet	1 %
14	<a href="http://www.espaciologopedico.com">www.espaciologopedico.com</a> Fuente de Internet	1 %
15	<a href="http://www.tdx.cat">www.tdx.cat</a> Fuente de Internet	1 %
16	<a href="http://bigdatalabblog.wordpress.com">bigdatalabblog.wordpress.com</a> Fuente de Internet	<1 %
17	<a href="http://tamarisco.datsi.fi.upm.es">tamarisco.datsi.fi.upm.es</a> Fuente de Internet	<1 %
18	<a href="http://luisenriqueelgato.blogspot.com">luisenriqueelgato.blogspot.com</a> Fuente de Internet	<1 %
19	<a href="http://cybertesis.uach.cl">cybertesis.uach.cl</a> Fuente de Internet	<1 %
20	<a href="http://studylib.es">studylib.es</a> Fuente de Internet	<1 %



21	tesis.ipn.mx Fuente de Internet	<1 %
22	exámenes.cervantes.es Fuente de Internet	<1 %
23	Submitted to Universidad Nacional de Trujillo Trabajo del estudiante	<1 %
24	ddd.uab.cat Fuente de Internet	<1 %
25	creativecommons.org Fuente de Internet	<1 %
26	livrosdeamor.com.br Fuente de Internet	<1 %
27	diariocorreo.pe Fuente de Internet	<1 %
28	learntutorials.net Fuente de Internet	<1 %
29	c024.wzu.edu.tw Fuente de Internet	<1 %
30	Submitted to Universidad Nacional Pedro Ruiz Gallo Trabajo del estudiante	<1 %
31	oa.upm.es Fuente de Internet	<1 %
32	www.ccadip.com Fuente de Internet	

<1 %

33

docplayer.es

Fuente de Internet

<1 %

34

Submitted to Universidad de Salamanca

Trabajo del estudiante

<1 %

35

repositorio-anterior.ulima.edu.pe

Fuente de Internet

<1 %

36

repositorio.espe.edu.ec

Fuente de Internet

<1 %

37

www4.congreso.gob.pe

Fuente de Internet

<1 %

Excluir citas

Activo

Excluir coincidencias < 20 words

Excluir bibliografía

Activo




## Recibo digital


Este recibo confirma que su trabajo ha sido recibido por **Turnitin**. A continuación podrá ver la información del recibo con respecto a su entrega.

La primera página de tus entregas se muestra abajo.

Autor de la entrega: Clara Mercedes Guevara Velez  
Título del ejercicio: TESIS  
Título de la entrega: MODELOS OCULTOS DE MARKOV PARA EL DESARROLLO DE ...  
Nombre del archivo: TESIS-MERCEDES-GUEVARA.docx  
Tamaño del archivo: 956.56K  
Total páginas: 108  
Total de palabras: 15,833  
Total de caracteres: 86,779  
Fecha de entrega: 15-jul.-2022 01:02p. m. (UTC-0500)  
Identificador de la entrega... 1870944039



**UNIVERSIDAD NACIONAL**  
**"PEDRO RUIZ GALLO"**



**FACULTAD DE INGENIERIA CIVIL, SISTEMAS Y**  
**ARQUITECTURA**

*Escuela Profesional de Ingeniería de Sistemas*

---

**TESIS**

Para obtener el título Profesional de:

Ingeniero de Sistemas

**TÍTULO**

MODELOS OCULTOS DE MARKOV PARA EL DESARROLLO DE UN  
SISTEMA DE AYUDA AL HABLA PARA PERSONAS QUE SUFREN  
DE DISARTRIA

**PRESENTADO POR:**

GUEVARA VELEZ, CLARA MERCEDES

**ASESOR:**

Mg. Ing. VILLEGAS CUBAS JUAN ELIAS

Lambayeque, ABRIL 2018