



UNIVERSIDAD TECNOLÓGICA CENTROAMERICANA

FACULTAD DE INGENIERÍA

PROYECTO DE GRADUACIÓN

ALGORITMO DE TRADUCCIÓN DE VOZ A TEXTO COMO HERRAMIENTA EDUCATIVA PARA

PERSONAS CON SORDERA

PRESENTADO POR:

11511160

DAVID ABRAHAM AMADOR GUEVARA

ASESOR METODOLÓGICO: ING. JUAN SÁNCHEZ

CAMPUS TEGUCIGALPA;

ABRIL, 2020

AGRADECIMIENTOS

Agradezco al Jardín y Escuela para niños sordos Manos Felices por ser de vital ayuda para la realización de esta tesis. En especial a la Lic. Oneyda Aguilar por su colaboración e incentivo a la correcta ejecución de este.

A Oliver Gerónimo Tome por su interés y gran aportación al proyecto en la parte de la programación del código y recomendaciones en otros aspectos del proyecto.

Al asesor, Ing. Juan José Sánchez por sus consejos y por compartir su conocimiento conmigo para ejecutar de una mejor manera cada proceso que llevó esta tesis.

EPÍGRAFE

“La causa de la libertad se convierte en burla si el precio a pagar es la destrucción de quienes deberían disfrutarla.”

Mohatma Gandhi.

RESUMEN EJECUTIVO

El siguiente documento detalla el trabajo realizado para el desarrollo de un algoritmo para la traducción de palabras comúnmente utilizadas en el diario vivir y poder traducirlas a texto en el contexto del Lenguaje de señas de Honduras (LESHO). La población hondureña que tienen discapacidades auditivas está restringida a acoplarse a las oportunidades que el sistema les ofrece, que no es nada comparado a los derechos que todos se merecen sin importar su condición física. Por lo que por medio de diferentes programas de edición de código e interpretación del mismo, se logró la creación de un algoritmo de traducción de voz a texto de este lenguaje. El enfoque de este algoritmo es el reconocimiento de las palabras captadas por un micrófono que en este caso es nuestra entrada, esto con la ayuda del importe de distintas librerías de apoyo; realizar un diccionario con algunas palabras de oraciones preseleccionadas y realizar un reordenamiento de la oración para que la oración este estructurada en forma de Glosas.

El método fue evaluado por medio de distintas pruebas a diferentes personas, esto con el fin de lograr el reconocimiento de las palabras con diferentes frecuencias de voz. La efectividad de la traducción fue evaluada por la Lic. Oneyda Aguilar experta en el área de interpretación del LESHO. El algoritmo representa una potencial base para futuros proyectos de traducción del LESHO, esto como solución a las distintas problemáticas a las que se ven afectadas estas personas.

ABSTRACT

The following document details the work carried out for the development of an algorithm for the translation of words commonly used in the daily life and to be able to translate them into text in the context of the Sign Language of Honduras (LESHO). The Honduran population with hearing disabilities is restricted to adapt to the opportunities that the system offers them, which is nothing compared to the rights that everyone deserves regardless of their physical condition. Therefore, through different code editing and interpretation programs, the creation of a speech-to-text translation algorithm for this language was achieved. The focus of this algorithm is the recognition of the words captured by a microphone, which in this case is our input, this with the help of the amount of different support libraries; make a dictionary with some pre-selected sentence words and reorder the sentence so that the sentence is structured in the form of Glosses.

The method was evaluated by means of different tests on different people, this in order to achieve the recognition of words with different voice frequencies. The effectiveness of the translation was evaluated by Lic. Oneyda Aguilar, expert in the area of interpretation of LESHO. The algorithm represents a potential basis for future LESHO translation projects, this as a solution to the different problems these people are affected by.

ÍNDICE DE CONTENIDO

I.	INTRODUCCIÓN.....	1
II.	PLANTEAMIENTO DEL PROBLEMA.....	2
	2.1 PRECEDENTES DEL PROBLEMA.....	2
	2.2 DEFINICIÓN DEL PROBLEMA.....	2
	2.3 JUSTIFICACIÓN	3
	2.4 PREGUNTAS DE INVESTIGACIÓN	3
	2.5 OBJETIVOS.....	3
	2.5.1 OBJETIVO GENERAL.....	3
	2.5.2 OBJETIVOS ESPECÍFICOS.....	3
III.	MARCO TEÓRICO	4
	3.1 ESTADO DEL ARTE.....	4
	3.1.1 TECNOLOGÍA DE ASISTENCIA PARA PERSONAS SORDAS BASADA EN LA PLATAFORMA DE ANDROID....	4
	3.1.2 SHOWLEAP	5
	3.1.3 MOTIONSAVVY	6
	3.1.4 AULA VIRTUAL	7
	3.1.5 SEÑAS DE HONDURAS.....	8
	3.2 LENGUAJE DE SEÑAS DE HONDURAS (LESHO)	9
	3.3 RECONOCIMIENTO DE VOZ POR MEDIO DE PYTHON.....	10
	3.3.1 EXPLICACIÓN DE CÓDIGO	10
	3.4 WORD PROCESING.....	12
	3.5 Sistema de Glosas	14
IV.	METODOLOGÍA.....	18

4.1 ENFOQUE.....	18
4.2 VARIABLES DE INVESTIGACIÓN.....	18
4.3 TÉCNICAS E INSTRUMENTOS APLICADOS.....	19
4.4 METODOLOGÍA DE ESTUDIO.....	19
4.4.1 CÓDIGO.....	20
4.5 Reconocimiento.....	24
4.6 STRING EXTRACTION.....	25
4.7 ESTRUCTURACIÓN DEL LENGUAJE.....	25
4.7.1 Sistema de Glosas.....	25
4.7.2 EXPLICACIÓN DE ESTRUCTURA EN CÓDIGO (LISTAS Y POSICIONES).....	26
4.8 INTERFAZ GRÁFICO.....	34
4.8.1 Creación.....	34
4.8.2 Ejecución.....	35
4.9 Metodología de validación.....	35
4.10 CRONOGRAMA DE ACTIVIDADES.....	36
V. RESULTADOS Y ANÁLISIS.....	37
5.1 VALIDACIÓN.....	37
5.1.1 PRIMERA PRUEBA (RECONOCIMIENTO).....	37
5.1.2 PRUEBA 2 (ESTRUCTURACIÓN DEL LENGUAJE).....	38
VI. CONCLUSIONES.....	40
VII. RECOMENDACIONES.....	41
VIII. APLICABILIDAD E IMPLEMENTACIÓN.....	42
IX. EVOLUCIÓN DE TRABAJO ACTUAL.....	43

ÍNDICE DE ILUSTRACIONES

Ilustración 1. Visualización de la Aplicación.....	4
Ilustración 2. Dispositivo de Traducción Showleap.	5
Ilustración 3. Tablet con el software de MOTIONSAVVY.....	6
Ilustración 4. Aula virtual para personas sordas Telefónica.....	7
Ilustración 5. Logo de Señas Honduras.....	8
Ilustración 6. Diagrama básico del Speech to text.....	11
Ilustración 7. Estructura del Lenguaje del país.....	16
Ilustración 8. Estructura del Lenguaje en Glosas.....	16
Ilustración 9. Variables de investigación.....	18
Ilustración 10. Proceso de creación del programa.....	20
Ilustración 11. Flujograma del funcionamiento del Logaritmo.....	22
Ilustración 12. Flujograma del funcionamiento del código.....	23
Ilustración 13. Diagrama de Metodología para traducción de voz a texto de Glosas.....	24
Ilustración 14. Captura del interfaz realizada por Visual Studio Code.....	34

ÍNDICE DE TABLAS

Tabla 1. Ejemplo del sistema de Glosas.....	14
Tabla 2. Diccionario de oraciones en glosas.....	17
Tabla 3. Funciones utilizadas en el Algoritmo	31
Tabla 4. Porcentaje de Efectividad de Reconocimiento.....	38
Tabla 5. Precisión de Estructuración en Glosas.....	39

ÍNDICE DE ANEXOS

Anexo 1. Pruebas a sujeto 1	44
Anexo 2. Pruebas a sujeto 2	44
Anexo 3. Pruebas a sujeto 3	44
Anexo 4. Pruebas a sujeto 4	44
Anexo 5. Pruebas a sujeto 5	44
Anexo 6. Pruebas a sujeto 6	44
Anexo 7. Pruebas a sujeto 7	44
Anexo 9. Captura de pruebas realizadas	44
Anexo 8. Pruebas a sujeto 8	44

LISTA DE SIGLA

Congénita	Para la Medicina, las enfermedades o malformaciones congénitas son aquellas innatas en un individuo, que se han presentado durante el periodo de desarrollo intrauterino, y que pueden ser consecuencia de radiaciones sobre el feto, fármacos o infecciones. (<i>Significado de Congénito, 2015</i>)
ASH:	Asociación de Sordos de Honduras. (Previamente ANSH Asociación Nacional de Sordos de Honduras) (A. Martínez, comunicación personal, 31 de enero de 2020)
Discapacidad auditiva:	Se considera como discapacidad auditiva o las deficiencias auditivas como aquellas alteraciones cuantitativas en una correcta percepción de la audición. (<i>Discapacidad Auditiva, 2019</i>)
Inclusión social:	La inclusión social es la tendencia a posibilitar que personas en riesgo de pobreza o de exclusión social tengan la oportunidad de participar de manera plena en la vida social, y así puedan disfrutar de un nivel de vida adecuado. (<i>Significado de Inclusión social, 2017</i>)
Lengua de señas:	Es la lengua o sistema lingüístico de carácter visual, gráfico espacial, gestual y manual en cuya conformación intervienen factores históricos, culturales, lingüísticos y sociales, utilizadas tradicionalmente como lenguas por las personas sordas, con discapacidad auditiva y sordo ciegas en Honduras. (LEY DE LA LENGUA DE SEÑAS HONDUREÑA (LESHO), 2014, art. 3)
LESHO:	Lengua de Señas Hondureñas (<i>Ley de la Lengua de Señas Hondureña (LESHO), 2014</i>).
Software:	Software es un término informático que hace referencia a un programa o conjunto de programas de cómputo, así como datos, procedimientos y pautas que permiten realizar distintas tareas en un sistema informático. (<i>Significado de Software (Qué es, Concepto y Definición) — Significados, 2019</i>)
Sordera:	Se denomina sordera o hipoacusia al déficit funcional que ocurre cuando una persona pierde capacidad auditiva en menor o mayor grado. Puede presentarse en forma unilateral, cuando afecta a un solo oído, o bilateral, cuando afecta ambos oídos (<i>Sordera y pérdida de la audición, 2018</i>).
Sordo/a:	Este vocabulario se dice especialmente a una persona, que padece una pérdida auditiva en mayor o menor grado, que ha privado del sentido del oído o no se escucha ni oye en su percepción (<i>Significado y definicion de sordo, etimologia de sordo, 2018</i>).

- API:** Las API son un conjunto de comandos, funciones y protocolos informáticos que permiten a los desarrolladores crear programas específicos para ciertos sistemas operativos (*¿Qué es una API y para qué sirve?*, 2015).
- Ágrafa** Que es incapaz de escribir o no sabe hacerlo (ASALE & RAE, 2019)
- Transliteración:** Representación de sonidos o caracteres de una lengua con signos del alfabeto de otra (*¿Qué significa Transliteración?*, 2019).
- Script:** El script es un documento que contiene instrucciones, escritas en códigos de programación. El script es un lenguaje de programación que ejecuta diversas funciones en el interior de un programa de computador (*Significado de Script*, 2018).
- Innato:** Connatural y como nacido con la persona misma (ASALE & RAE, 2019).
- Tupla:** En Python, una tupla es un conjunto ordenado e inmutable de elementos del mismo o diferente tipo. Las tuplas se representan escribiendo los elementos entre paréntesis y separados por comas (*Tuplas: listas inmutables. Python. Bartolomé Sintés Marco. www.mclibre.org*, 2016).
- Bucle:** Las repeticiones forman parte de cualquier lenguaje de programación, al menos en la inmensa mayoría. Se les llama estructuras de control (en realidad son un tipo de estructuras de control) porque nos permiten controlar el flujo de ejecución de los programas. En estas estructuras de control lo que hacemos es repetir la ejecución de un código un número de veces (*Repeticiones o bucles*, 2015).

I. INTRODUCCIÓN

Para la física, el sonido es todo aquello que implica un fenómeno vinculado a la difusión de una onda con propiedades elásticas que produce vibraciones en un cuerpo. El sonido audible para los humanos se conforma por las diferencias de presiones del aire, que el oído transforma en ondas mecánicas para poder ser percibidas por el cerebro y así mismo procesadas por él.

Sin embargo existen casos en los cuales personas carecen de la capacidad auditiva, ya sea por causas congénitas o adquiridas.

Se dice que alguien que sufre daño en la audición cuando no es apto para oír tan bien como una persona cuyo sentido de la audición es habitual, es decir, cuyo umbral de audición de ambos oídos debe ser igual o mayor a 25dB.

El medio por el cual se comunican las personas que sufren de sordera es mediante un lenguaje de señas; sin embargo es importante destacar que cada país cuenta con un lenguaje de señas distinto, por lo tanto nuestra nación cuenta con un Lenguaje de señas de Honduras o más conocido como LESHO.

Actualmente existen algunas instituciones las cuales ofrecen la enseñanza de este lenguaje como ser la Universidad Nacional Autónoma de Honduras (UNAH) y la escuela para sordos "Amor en Acción", en donde hay personas expertas en el área de la lengua de señas, y ellos imparten el conocimiento a través de varios módulos de aprendizaje.

El proyecto consiste en la implementación de un algoritmo que pueda convertir el audio a texto, el cual pueda extraer palabras de una conversación o en este caso las palabras de la clase impartida en LESHO. Estas librerías estarán enfocadas en captar el audio que emita el sujeto para que así el algoritmo de Python pueda procesarlo eliminando lo que son ruidos externos para una mejor captación de ello. Toda esta librería será para transformar el audio a texto. También se creará una librería la cual tenga palabras selectas para el lenguaje en señas que se implementará en el traductor.

II. PLANTEAMIENTO DEL PROBLEMA

2.1 PRECEDENTES DEL PROBLEMA

De acuerdo a la asociación de sordos de Honduras (ASH), en el país existen aproximadamente alrededor de 34 mil habitantes que presentan problemas de sordera ya sea parcial o total, los cuales carecen de una ayuda para su educación por parte del gobierno. (A. Martínez, comunicación personal, 31 de enero de 2020)

Actualmente solo existen 2 instituciones educativas que brindan educación superior a estas personas, las cuales son la universidad Autónoma de Honduras (UNAH) y la Universidad Pedagógica Nacional Francisco Morazán (UPNFM); y únicamente 1 escuela para dar educación media (Escuela Amor en acción).

Actualmente en el país no existen aplicaciones o programas que beneficien a las personas con discapacidades auditivas, a excepción de la app "Señas Honduras" creada por el Ing. Shamir Valdivia en el año 2019. (Valdivia, 2019)

2.2 DEFINICIÓN DEL PROBLEMA

Actualmente en el país la mayoría de las personas sordas no tienen acceso a escuelas especiales o a la facilidad de tener un traductor del lenguaje LESHO, la mayoría de la población desconoce la existencia del LESHO, sin embargo es la herramienta más útil de comunicación de las personas con discapacidad auditiva; por lo que debería de estar disponible para que cada una de estas personas pueda tener acceso a lo que todos los individuos tenemos.

Actualmente en el país no existe ninguna aplicación que sirva como herramienta para la educación de las personas sordas, la curva de aprendizaje del LESHO es bastante alta. En el extranjero existen dispositivos y aplicaciones que sirven de apoyo para la educación de estas personas, sin embargo el lenguaje de señas es único en nuestro país, distinto al de cualquier otro país en el extranjero, por lo que no nos brindaría una comunicación eficaz.

2.3 JUSTIFICACIÓN

Esta solución podrá dar un beneficio a la población sorda en el área educativa, esto por medio de un interface entre máquina y alumno; el cual el alumno utilizará durante la clase para poder entender al maestro cuando este imparta su clase siendo traducida a texto en LESHO. La creación de un algoritmo de traducción es el principio para que la tecnología se incluya y esta pueda hacer más llevadera la vida de los sordos y también poder tener una inclusión social para ellos en la comunidad estudiantil.

2.4 PREGUNTAS DE INVESTIGACIÓN

¿Será este algoritmo un recurso de utilidad para el área educativa?

¿Existen comercialmente dispositivos que funcionen como interfaz?

¿Comparten características los lenguajes de señas de otros países o son únicos para cada lugar?

2.5 OBJETIVOS

2.5.1 OBJETIVO GENERAL

Crear un algoritmo traductor de voz a texto en la estructura gramatical utilizada por los sordos, el cual sirva como herramienta de apoyo en el área Educativa; con la utilización de un micrófono y una computadora.

2.5.2 OBJETIVOS ESPECÍFICOS

- Programar con la ayuda del programa Visual Studio y con el uso del intérprete de Python, un código con el cual se pueda interpretar la voz y que lo imprima en forma de texto.
- Implementar en el programa un interface que sea de fácil entendimiento para el usuario.
- Integrar palabras del lenguaje LESHO en la base de datos que utilizará la librería.
- Lograr la interpretación necesaria para realizar una correcta traducción.

III. MARCO TEÓRICO

3.1 ESTADO DEL ARTE

3.1.1 TECNOLOGÍA DE ASISTENCIA PARA PERSONAS SORDAS BASADA EN LA PLATAFORMA DE ANDROID

Zarqa, Jordania.

El poder de esta aplicación aparece en dos aspectos: en primer lugar, la capacidad de las personas normales para comunicarse con las personas objetivo sin tener ningún conocimiento previo sobre el lenguaje de señas. Esto se puede lograr mediante el reconocimiento de voz de las palabras o escribiendo las palabras en el idioma árabe. La aplicación muestra las imágenes apropiadas en el lenguaje de señas.

En segundo lugar, y lo que es más importante, las personas con necesidades especiales se comunican con personas normales al elegir las imágenes de los signos en sus teléfonos de las numerosas categorías almacenadas en las bases de datos que expresan sus ideas y pensamientos. En consecuencia, el conjunto de imágenes se transforma en un párrafo de texto. Se evaluó aquí la aplicación probándola en usuarios reales sordos. Se creó cuidadosamente escenarios en situaciones realistas. Los primeros resultados fueron prometedores en este estudio, ya que todos los sordos encontraron útil la tecnología propuesta y el 90% de ellos quería usarla a diario. (Abdallah & Fayyumi, 2016)



Ilustración 1. Visualización de la Aplicación.

Fuente: (Abdallah & Fayyumi, 2016)

3.1.2 SHOWLEAP

Madrid, España.

Uno de los problemas que se encuentran los usuarios del Lenguaje de Señas es que únicamente pueden comunicarse con aquellas personas que igualmente la utilizan. Otra opción es tener un intérprete de Lengua de Signos. Sin embargo en la vida diaria, el trabajo o cualquier gestión burocrática, el individuo sordo usuaria de Lengua de Signos se puede hallar incomunicada. Por ello, existen varias soluciones tecnológicas para que las personas sordas puedan comunicarse. Esto es una completa revolución y, sin embargo varios proyectos tratan de remediar esa brecha de comunicación, Showleap lo hace integrando un alto componente tecnológico para un funcionamiento óptimo. Showleap es un diseño español de software que traduce la lengua de signos a voz. El beneficiado signante utiliza unos brazaletes que reconocen el movimiento de sus manos y dedos, y lo convierte en audio mediante un sintetizador de voz, y lenguaje escrito en una pantalla. (*Tecnología que cambiará la vida de las personas con pérdida auditiva*, 2018)



Ilustración 2. Dispositivo de Traducción Showleap.

Fuente: («Showleap Technologies», 2018)

3.1.3 MOTIONSAVVY

New York, USA.

Este recurso, que aún está en proceso de progreso, trata asimismo de descartar la brecha de comunicación entre personas signantes y oyentes, pero a partir de un ángulo más intuitivo y encaminado a un público más joven, incluso infantil. La persona signante utiliza una tablet con el software MotionSavvy, que es capaz de reconocer los signos que realiza con sus manos y convertirlos a voz. El beneficiado oyente le habla a la tablet, que convierte su mensaje en texto para el individuo sordo. Este proyecto, que se desarrolló en Nueva York, fue fundado por cuatro personas sordas inspirados por los problemas de inclusión social que veían en algunos niños sordos signantes. Seleccionado como uno de los proyectos de la plataforma de start-ups de Leap Motion (sociedad que ha creado un software de reconocimiento de movimientos), va cumpliendo hitos antes de su salida al mercado: han conseguido una financiación a través de la plataforma de crowdfunding IndieGogo, y siguen trabajando en el software para amp. *(Tecnología que cambiará la vida de las personas con pérdida auditiva, 2018)*



Ilustración 3. Tablet con el software de MOTIONSAVVY.

Fuente: (Iván, 2011)

3.1.4 AULA VIRTUAL

Lima, Perú.

El proyecto "Aula Móvil" es un plan innovador de la Fundación Telefónica que integra herramientas tecnológicas, el acceso a medios educativos y al aprendizaje a docentes, cuyo objetivo es optimizar el aprendizaje de escolares, así como subyugar la brecha digital en escuelas públicas rurales y urbano-marginales del país mediante el uso didáctico de los recursos tecnológicos disponibles en las escuelas. "La Fundación Telefónica ha ratificado su compromiso con los niños que sufren de sordera, promoviendo la rutina del uso de herramientas pedagógicas innovadoras indispensables, que al ser visuales se adaptan espléndidamente a las necesidades educativas de esta población", afirmó Lillian Moore, Gerente de Fundación Telefónica. Alumnos participantes con este proyecto, se beneficiaron un total de 61 niños y adolescentes, 5 maestros oyentes y 4 instructores sordos que son parte de la Asociación de Señas de Perú, un programa que busca amplificar la habilidad lingüística y cultural en los participantes sordos de su país.(Telefónica, 2014)



Ilustración 4. Aula virtual para personas sordas Telefónica.

Fuente: (Telefónica, 2014)

3.1.5 SEÑAS DE HONDURAS

Tegucigalpa, Honduras

Señas de Honduras es una aplicación descargable para teléfonos Android, la cual brinda al usuario que lo descarga una serie de palabras básicas con las cuales cualquier persona puede lograr aprender un poco del LESHO.

En esta app se puede aprender palabras como: Primeras palabras o palabras básicas, el alfabeto, los números, los colores, los miembros de la familia, algunos saludos, el nombre de alimentos, preguntas básicas, palabras relacionadas a la salud y como es de esperarse los departamentos los cuales conforman nuestro país Honduras.

Este proyecto fue llevado a cabo a finales de 2018 por el ingeniero en Biomédica Shamir Valdivia junto con la colaboración de la Asociación de Sordos de Honduras (ASH).

La aplicación puede ser encontrada tanto en Play Store como en páginas como APK.(Valdivia, 2019)



Ilustración 5. Logo de Señas Honduras.

Fuente: (*Señas Honduras - Apps en Google Play*, 2019)

3.2 LENGUAJE DE SEÑAS DE HONDURAS (LESHO)

El lenguaje LESHO es la lengua o sistema lingüístico en el cual interactúan aspectos visuales, gestos y manuales en los cuales están comprometidos factores históricos, culturales, lingüísticos y sociales utilizados usualmente por las personas con discapacidades auditivas o sordo ciegas de Honduras.

Este lenguaje fue decretado como oficial en el año 2014 en el diario "La Gaceta".

Según información de la ASH existen alrededor de 34 mil casos de sordera en el país, entre casos congénitos y adquiridos. (A. Martínez, comunicación personal, 31 de enero de 2020)

Actualmente en las distintas instituciones que brindan educación de nivel superior a personas sordas, tienen que contratar a una persona profesional en el área de la Lengua de Señas de Honduras; por lo que con la ayuda de la creación de un código e interpretación de este mismo con la plataforma de interpretación de Python, podría ayudar a la agilización de los procesos.

La ASH tiene alrededor de 860 miembros registrados en 9 distintos departamentos de Honduras. Como Asociación brinda clases de LESHO para personas oyentes y personas sordas en Tegucigalpa durante todo el año y en ciertos municipios mediante proyectos se les enseña la lengua a los sordos y sus familias.

Dentro de las instituciones que brindan apoyo están La escuela para sordos Manos Felices, La escuela Taller Amor en Acción, Centro de Educación básica CIRE en la ciudad de Tegucigalpa, y en la ciudad de San Pedro Sula La escuela Esmirna y La escuela municipal de San Pedro Sula.

En la actualidad la única tecnología relacionada con el lenguaje LESHO es la aplicación disponible para los teléfonos Android "Señas Honduras"; creada por el Ingeniero en Biomédica Shamir Valdivia. Sin embargo ninguna herramienta educativa la cual pueda utilizarse en manera de traducción (A. Martínez, comunicación personal, 31 de enero de 2020).

3.3 RECONOCIMIENTO DE VOZ POR MEDIO DE PYTHON

El reconocimiento de voz es un proceso en el que una computadora o dispositivo graba el habla humana y lo convierte a formato de texto.

También se conoce como reconocimiento automático de voz (ASR), reconocimiento de voz por computadora o voz a texto (STT).

La lingüística, la informática y la ingeniería eléctrica son algunos campos asociados con el reconocimiento de voz.

En Python se necesitan de 2 librerías principales para realizar la traducción de voz a texto, las cuales son:

- SpeechRecognition
- PyAudio

El reconocimiento de voz en python se realiza por medio del uso de la librería SpeechRecognition. Esta biblioteca es compatible con varios engines y API's en línea o fuera de línea como Google Cloud Speech API, Microsoft Bing Voice Recognition, IBM Speech to Text, y algunos otros más. (*Speech Recognition Python - Converting Speech to Text*, 2018)

3.3.1 EXPLICACIÓN DE CÓDIGO (RECONOCIMIENTO DE VOZ)

En primer lugar, se importará `speech_recognition` como `sr`.

Tenga en cuenta que se tiene `speech_recognition` en dicho formato, mientras que anteriormente lo se instaló de esta manera `SpeechRecognition`, por lo que debe echar un vistazo a los casos porque distingue entre mayúsculas y minúsculas.

Ahora lo hemos usado como notación porque escribir discursos de reconocimiento completos a la vez no es una buena manera.

Ahora se tiene que inicializar `r = sr.Recognizer ()`, esto funcionará como un reconocedor para realizar el reconocimiento de voz del usuario.

Entonces, con `sr.Microphone ()` como fuente: lo que significa que se está inicializando la fuente a `sr.Microphone`, también se puede usar algunos archivos de audio para convertirlos en texto, pero en este tutorial estoy usando la voz de Micrófono.

A continuación, se imprimirá una declaración simple que recomienda al usuario hablar cualquier cosa.

Luego se utilizó el comando `r.listen (fuente)` y proceder a escuchar la fuente, por lo que escuchará la fuente y la almacenará en el audio.

Puede suceder que en algún momento el audio no esté claro y que no lo obtenga correctamente, por lo que podemos ponerlo dentro del bloque `try` y `except`.

Entonces, dentro del bloque de prueba, el texto será `text = r.recognize_google (audio)`, ahora se tiene varias opciones como `Recognize_bing ()`, `Rcognize_google_cloud ()`, `Recognize_ibm ()`, etc. Pero para este se usó `Recognize_bing ()`. Y finalmente hay que pasar el audio.

Y esto convertirá el audio en texto.

Después para imprimir el texto se utilizó `print ("Usted dijo: {}". Formató (texto))`, esto imprimirá lo que haya dicho.

En el bloque `excepto` se puede simplemente escribir `print ("Lo siento, no te entendí")`, esto le enviará un mensaje si su voz no está grabada claramente.

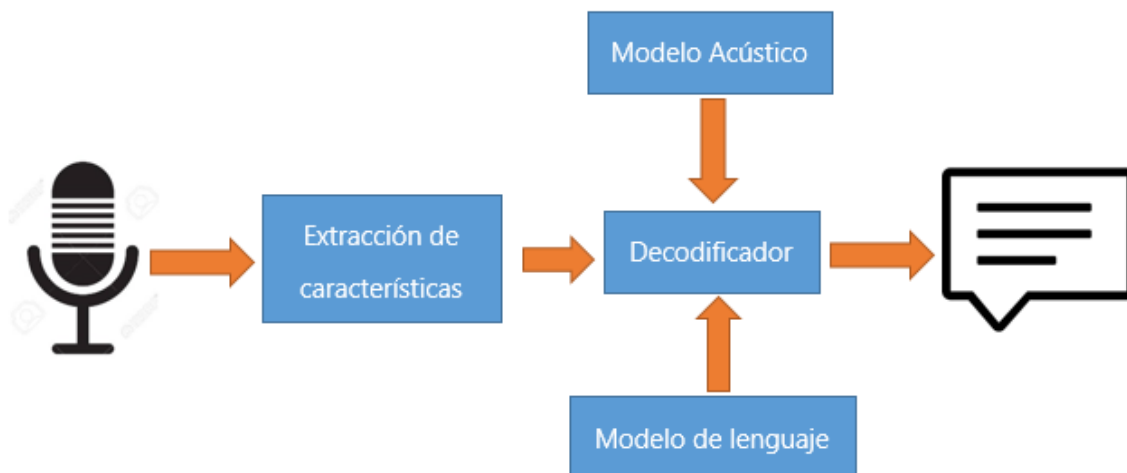


Ilustración 6. Diagrama básico del Speech to text.

Fuente: (Realizada por el autor).

3.4 WORD PROCESING.

Los primeros procesadores de texto fueron máquinas independientes similares a las máquinas de escribir eléctricas que se estrenaron en la década de 1960. La gran ventaja de estas primeras máquinas sobre el uso de una máquina de escribir era que podía hacer cambios sin tener que volver a escribir todo el documento. Con el tiempo, los dispositivos adquirieron características más avanzadas, como la capacidad de guardar documentos en un disco, opciones de formateo elaboradas y corrección ortográfica.

Si bien todavía hay algunos procesadores de texto independientes en uso hoy en día, el procesamiento de texto comenzó a trasladarse a las computadoras personales en la década de 1980. En los primeros días de la PC, un procesador de textos llamado WordPerfect se convirtió en una de las aplicaciones más utilizadas de cualquier tipo. Sin embargo, con el tiempo, los procesadores de texto que mostraban a los usuarios exactamente lo que imprimirían en sus documentos finales se hicieron más populares. Uno de esos procesadores de texto WYSISWG, Microsoft Word, se hizo dominante en la década de 1990.

Los procesadores de texto permiten crear un documento, almacenarlo electrónicamente, y mostrar el documento en una pantalla. Además, los procesadores de texto permiten modificaciones de documentos ya sea por mecanografiadas, activadas por voz o seleccionadas por comandos y permitir la impresión de texto también. El tipo de procesador de texto permitió que el procesador variara, ya sea solo un procesador de texto dedicado, teclado con almacenamiento o computadora con software de procesamiento de texto, por lo que siempre que poseerá las características descritas anteriormente.

El procesamiento de textos mejora la escritura de los estudiantes que experimentan dificultades para aprender a escribir o leer. Los efectos positivos del procesamiento de textos en la escritura indican de manera sólida una mejora longitud, desarrollo / organización y calidad del texto, así como un número reducido de errores mecánicos en los trabajos de los alumnos. Igual de importante, el procesamiento de texto aumenta la motivación de los estudiantes de bajo rendimiento para escribir. Quizás el más intrigante a partir de esta revisión se encontró que los programas de procesamiento de texto que brindan retroalimentación sobre la calidad del texto o

incitar a los estudiantes a participar en actividades específicas que la planificación, la redacción o la revisión de textos con apoyo son particularmente efectivas. La implicación es que los escritores / lectores más débiles deberían usar la palabra procesamiento como su herramienta principal para componer, ya que aumenta la cantidad de estudiantes causando motivación para escribir y produce resultados de escritura superiores en comparación de estar escribiendo a mano.

En un artículo publicado en 2003 (Armenta et al., 2003) informa los resultados de una investigación sobre tipos de error en español que se encuentra en textos de la vida real. Estos errores producidos por el ser humano se pueden agrupar en dos tipos principales: errores cognitivos (es decir, conceptos erróneos sobre reglas ortográficas o falta de conocimiento del lenguaje) y errores relacionados con errores de escritura (errores motores cuando se usa un teclado de computadora). Los resultados obtenidos en esta investigación se aplicaron a la implementación de un corrector ortográfico que detecta errores ortográficos independientes del contexto en textos generales sin restricciones. El artículo (Armenta et al., 2003) habla acerca de un corrector ortográfico que proporciona la corrección de errores de palabras aisladas al ofrecer un conjunto de correcciones candidatas que están cerca de la palabra mal escrita. La técnica que subyace a la corrección ortográfica. El algoritmo es el de la distancia mínima de edición (número de adiciones, sustituciones, omisiones y transposiciones de dos caracteres adyacentes). Cada operación se califica con una puntuación. La mejor sugerencia es la que tiene la puntuación más baja. En este contexto, la elaboración de la tipología de error tenía como objetivo (a) encontrar una categorización basada en la frecuencia de las cuatro operaciones de edición mencionadas anteriormente en español, y (b) descubrir los pares de error / corrección más frecuentes. Estos dos factores nos permitieron mejorar la clasificación de sugerencias ofrecidas por el corrector ortográfico como correcciones candidatas para errores ortográficos. Este artículo (Bustamante & Díaz, 2006) discute las generalizaciones más comunes sobre los patrones de error ortográfico y presenta resultados adicionales sobre la frecuencia y los tipos de errores ortográficos que se encuentran en los corpus de Reallife. Sus hallazgos confirman los resultados de Damerau en que la mayoría de los errores tienden a ser instancias únicas de inserciones, eliminaciones, sustituciones y transposiciones. También encontraron que la mayoría de los errores de ortografía en español son:

1. Omisiones (principalmente, de acento o un carácter)
2. Sustituciones de minúsculas por mayúsculas al comienzo de un nombre propio.
3. Errores cognitivos.
4. Adición.
5. Sustitución de un personaje y
6. Transposición.

Otras conclusiones son: a) el porcentaje de errores en la primera letra de una palabra es mayor que el informado en otros estudios, b) los efectos de adyacencia del teclado son menos importantes que otros factores, y c) no existe necesariamente una correlación directa entre frecuencia de un personaje en un corpus y sus posibilidades de ser un error (Bustamante & Díaz, 2006).

3.5 Sistema de Glosas

La lengua de señas es ágrafa, no tiene escritura, lo cual no afecta su estatus de lengua, pero para el estudio y análisis de la misma, se ha creado un sistema o sistemas, según el cual, usando palabras escritas del idioma nacional se acerca a la estructura gramatical que usan los sordos.

A este sistema o sistemas de representación escrita son conocidas como "Glosas", bajo este concepto se realiza la traducción de voz a texto en Glosas.

La glosa se puede describir como la transcripción de las señas a un medio escrito representando así su concepto. Algunos ejemplos de este:

Tabla 1. Ejemplo del sistema de Glosas

Palabra en estructura del idioma local	Palabra estructurada en glosa
Me gusta ese carro rojo	CARRO ROJO GUSTAR
No me gusta ese carro rojo	CARRO ROJO GUSTAR NO

Fuente: entrevista con directora de Escuela Manos felices.

Esto se debe a la Teoría Lingüística de Chomsky, el cual afirma que existe un dispositivo innato ubicado en el cerebro, el Dispositivo para la Adquisición del Lenguaje, que permite aprender y utilizar el lenguaje de forma casi instintiva, comprobando además que los principios generales de la gramática son universales para todos los seres humanos, defendiendo así la existencia de una Gramática Universal (Iván, 2011).

Las estructuras para el desarrollo del Lenguaje pueden ser Profunda o Superficial.

La Estructura Profunda es como pensamos, y se compone básicamente de palabras como sustantivos, adjetivos, verbos etc.

La Estructura Superficial es como transmitimos esa idea en que pensamos.

La estructura profunda puede ser idéntica entre una persona oyente y una que no lo es a la hora de crear una idea, sin embargo la forma de transmitirla es totalmente distinta entre ambas personas.



Ilustración 7. Estructura del Lenguaje del país.

- CARRO ROJO GUSTAR.
- CARRO ROJO GUSTAR NO.
- CARRO ROJO (EL) PAPÁ MÍO



Ilustración 8. Estructura del Lenguaje en Glosas.

Fuente:(O. Aguilar, comunicación personal, 12 de marzo de 2020).

El diccionario de palabras en estructura de Glosas implementado en el algoritmo incluye 10 oraciones básicas como prueba para realizar un correcto reordenamiento de las palabras en el contexto del Lenguaje local y dejándolo en la estructura de LESH0 (O. Aguilar, comunicación personal, 12 de marzo de 2020), las oraciones implementadas son las siguientes:

Tabla 2. Diccionario de oraciones en glosas.

Oración en Lengua local	Oración en Glosa	Observación
Las personas de la ciudad, ellos son amables.	CIUDAD PERSONA++ ELLOS AMABLE	Las ++ implica duplicación de la seña
La tarea está difícil.	TAREA DÍFICIL	
No conozco ese lugar de allí.	LUGAR ALLÍ NO-CONOCER	ALLÍ, la seña es un señalamiento con el índice. NO-CONOCER es una sola seña, la negación es simultánea
¿Cuándo es tu cumpleaños?	SU CUMPLEAÑOS ¿CUÁNDO?	
¿Cuál es tu color favorito?	COLOR SU FAVORITO ¿CUÁL?	
¿Dónde está ese lugar?	LUGAR ESE ¿DÓNDE?	ESE, la seña es un señalamiento con el índice
¡Que emoción verte a ti hoy!	HOY VER-A-TÚ EMOCIÓN	VER-A-TÚ, es una sola seña, ya que es un verbo direccional.
¡Ahí está mi papá!	MI PAPÁ ALLÍ (a) MI PAPÁ ALLÍ ESTÁ (b) MI PAPÁ ALLÍ TENER (c)	Estas son tres opciones para la misma oración. La más común es la (a), luego está la (b) y en lengua de señas el verbo tener equivale a estar, por eso está la opción (c)
Ojalá pueda venir mi mamá con nosotros	MAMÁ MI CON (JUNTO) OJALÁ.	CON, la seña es la misma de JUNTO A, lleva un giro en el movimiento para indicar el "nosotros"
Ojalá funcione nuestro proyecto.	PROYECTO PROPIO OJALÁ BIEN (a) PROYECTO NUESTRO OJALÁ BIEN (b)	Se colocaron dos opciones de este.

Fuente (O. Aguilar, comunicación personal, 12 de marzo de 2020)

IV. METODOLOGÍA

4.1 ENFOQUE

El enfoque de la investigación es de carácter cualitativo, ya que esta no se basará en demostrar hipótesis mediante un estudio estadístico. Habrá definición de las variables de investigación pero en este caso no se manipulará ninguna de ellas.

4.2 VARIABLES DE INVESTIGACIÓN

- Algoritmo utilizado: El algoritmo utilizado brindará la capacidad de traducción y el tiempo de reacción necesario para lograr el mismo.
- Palabras seleccionadas: Las palabras seleccionadas para la traducción se extraerán del catálogo de palabras del LESH0.

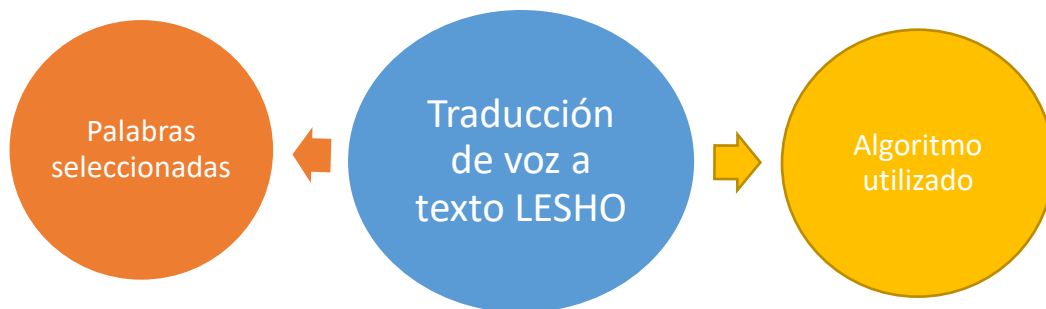


Ilustración 9. Variables de investigación.

Fuente: (Realizada por el autor)

4.3 TÉCNICAS E INSTRUMENTOS APLICADOS

Se utilizó Python 3.7.6 para realizar la ejecución de un script donde se desarrolla la detección de audio y su respectiva conversión a texto, esto con la ayuda de las librerías speech recognition que este programa nos ofrece. Para la programación se utilizó Visual Studio Code versión 1.41.1 en la cual se realizó el script para luego ser ejecutado por Python.

4.4 METODOLOGÍA DE ESTUDIO

Por medio de la edición por parte de Visual Studio Code y la ejecución del intérprete de Python, se desarrolló un programa en donde se implementó un algoritmo de conversión de voz a texto el cual ejecuta una librería de reconocimiento de voz, que utiliza como recurso para captar el sonido el micrófono de la computadora, a esta se le conoce como Speech Recognition; este sonido se transforma a texto y para eso se necesita una librería de Google Cloud Speech API la cual ya se encuentra predeterminada y la cual permite extraer las palabras a reconocer en el primer paso de la traducción, también permite incorporar el lenguaje del español de Honduras. La principal función de esta librería de Google es tomar el audio captado y transformarlo a texto en el lenguaje de la elección del usuario. Existen otros algoritmos con la misma función que el que se está desarrollando, pero se eligió este como base por su fácil interpretación al leer el código y su rapidez de respuesta al ser ejecutado. Se eligió la librería de Google por el motivo que ya estaba creada y que además de eso Google ya tiene amplia experiencia en lo que es transformar audio a texto por su traductor.

Para la segunda parte de la traducción se hizo una búsqueda de las funciones que cumplieran con los requisitos dados en el programa, en este caso el reordenamiento de la oración para estructurarla gramaticalmente de la forma que los sordos la utilizan.

Utilizando Visual Studio Code se creó un interfaz de fácil entendimiento para los usuarios que lo utilizan. Se realizaron varias pruebas para ambas partes de la traducción tanto para el reconocimiento como para ordenamiento estructural de las oraciones.

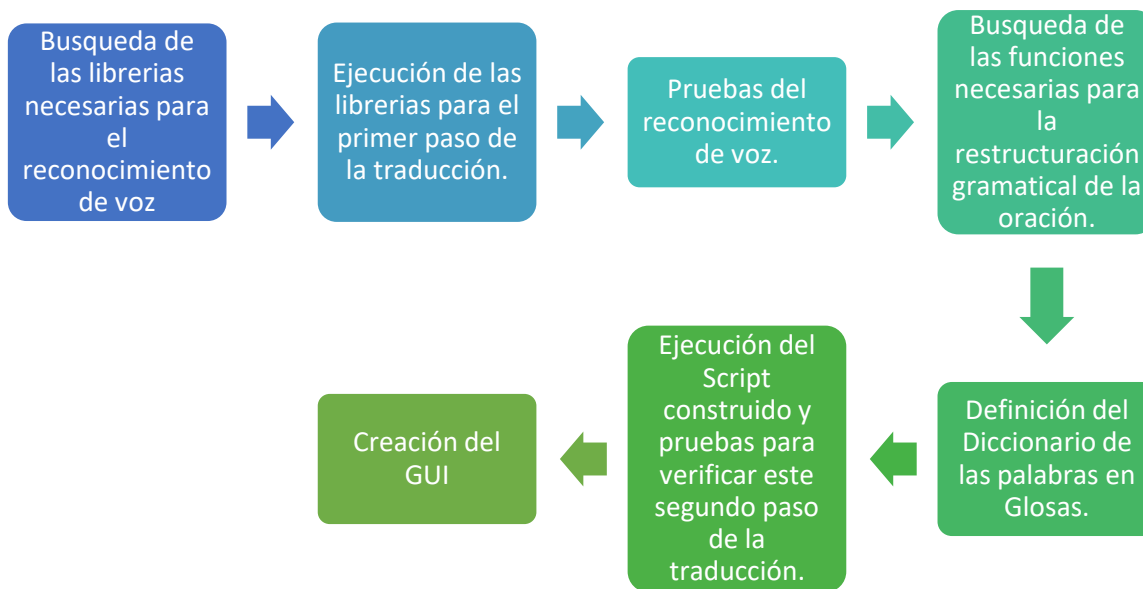


Ilustración 10. Proceso de creación del programa.

Fuente: (Realizado por el autor).

4.4.1 CÓDIGO

Como primer paso se ejecuta el programa, este realiza un importe de la librería **Speech Recognition** por medio de la sentencia **import** que en Python se trata de una declaración simple la cual nos permite importar códigos de otros ficheros. Luego de esto el programa imprimirá en la terminal el mensaje "Di algo", por lo que el algoritmo procederá a "escuchar" por así decirlo, el audio emitido por el usuario y este se introduce en un bloque **Try Except;** con el propósito de encontrar un error en el proceso de reconocimiento de voz, el error en este caso es que no se pueda reconocer el audio. En el caso que no se reconozca el audio pues el algoritmo nos imprimirá el mensaje "Lo siento, no te entendí" y finalizará la ejecución del programa.

En el caso que se pueda reconocer de una manera efectiva la voz del sujeto por medio de la utilización de la librería de Google, pues el algoritmo procederá a imprimir el mensaje "Listo" y proceder a imprimir lo que es el texto reconocido en esta primer parte.

Luego de esta primer parte empieza lo que es el reordenamiento de la oración para dejarla estructurada en el sistema de Las Glosas; para esto se comienza ingresando la oración en una lista

e iterando por todos los elementos de la lista para poder extraer la posición de las palabras en la oración. La iteración de estos elementos se realiza por medio de una sentencia llamada **Def** la cual es una definición de función que se utiliza para encontrar las posiciones de las palabras. Por medio de la función **enumerate()** se extraen estos elementos, ella toma como argumento un objeto iterable y nos regresa otro en forma de tuplas de 2 objetos los cuales son la posición de la palabra en la oración y la palabra misma.

En el caso que no se encuentre el elemento a buscar, pues el algoritmo no regresará ningún elemento y finalizará la ejecución del programa.

Luego se procede a definir lo que es el diccionario de las Glosas agregando lo que son las palabras que formarán las nuevas oraciones, esto por medio de la función `deppend` la cual podrá editar la lista y poder agregar cada palabra.

Después el programa "lee" por así decirlo el string impreso en el reconocimiento de voz y lo separa en una lista por medio de la función **Split**. El algoritmo procederá a realizar una búsqueda por medio de un bucle continuo para encontrar las palabras que conformarán la oración estructurada en el sistema de Las Glosas en el diccionario implementado; si esta palabra no se encuentra pues la traducción se verá como un espacio en blanco. En el caso que si se encuentre pues se guardará las nuevas palabras en una lista de manera desordenada.

El ordenamiento de esta palabra corre por medio de la función **sorted**, la cual devuelve una nueva lista con todos los elementos del diccionario que se encuentren en la oración, ordenados de manera ascendente. Como último paso se imprime la oración ordenada y finaliza la ejecución del algoritmo.

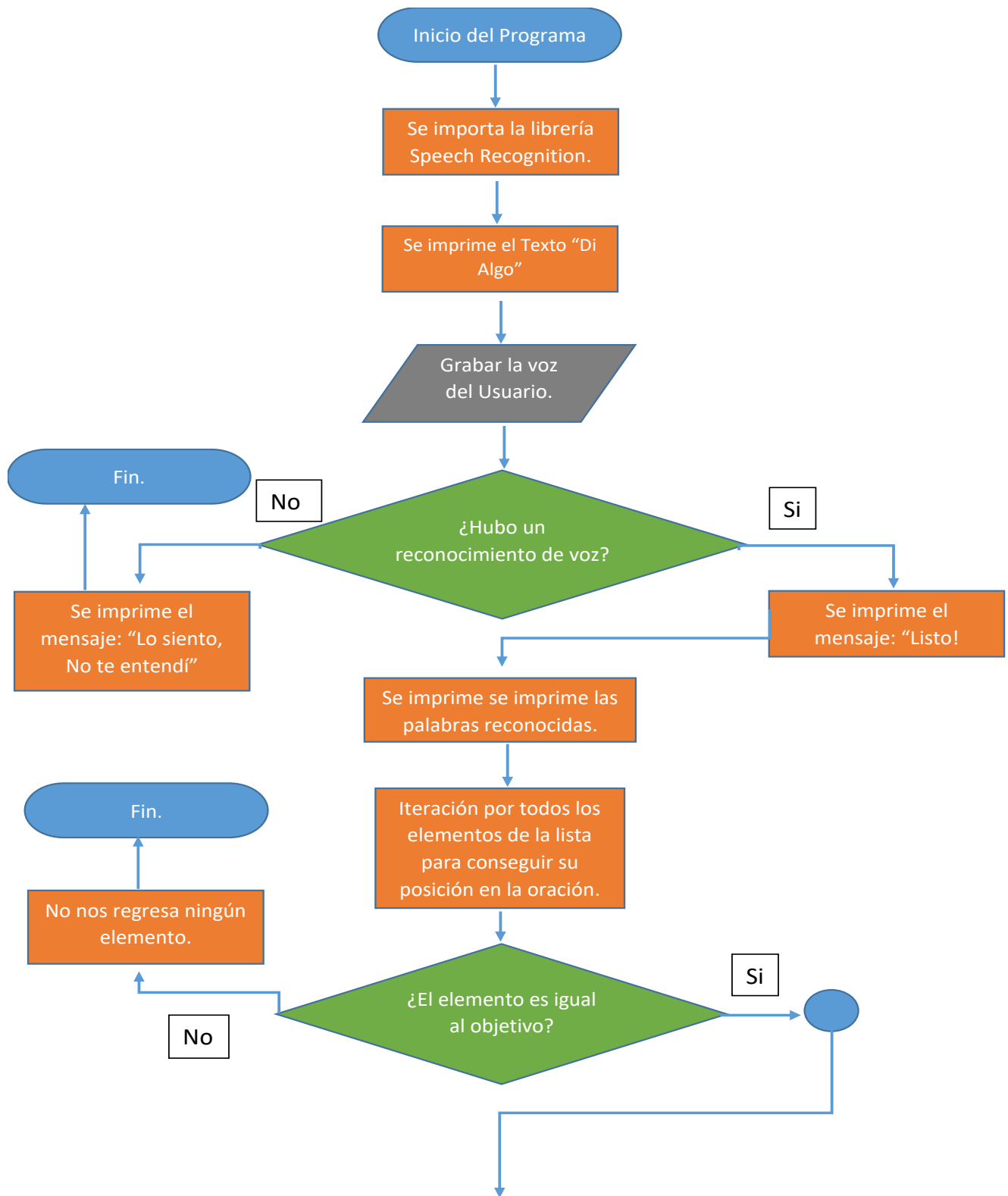


Ilustración 11. Flujograma del funcionamiento del Logaritmo.

Fuente: (Realizada por el autor)

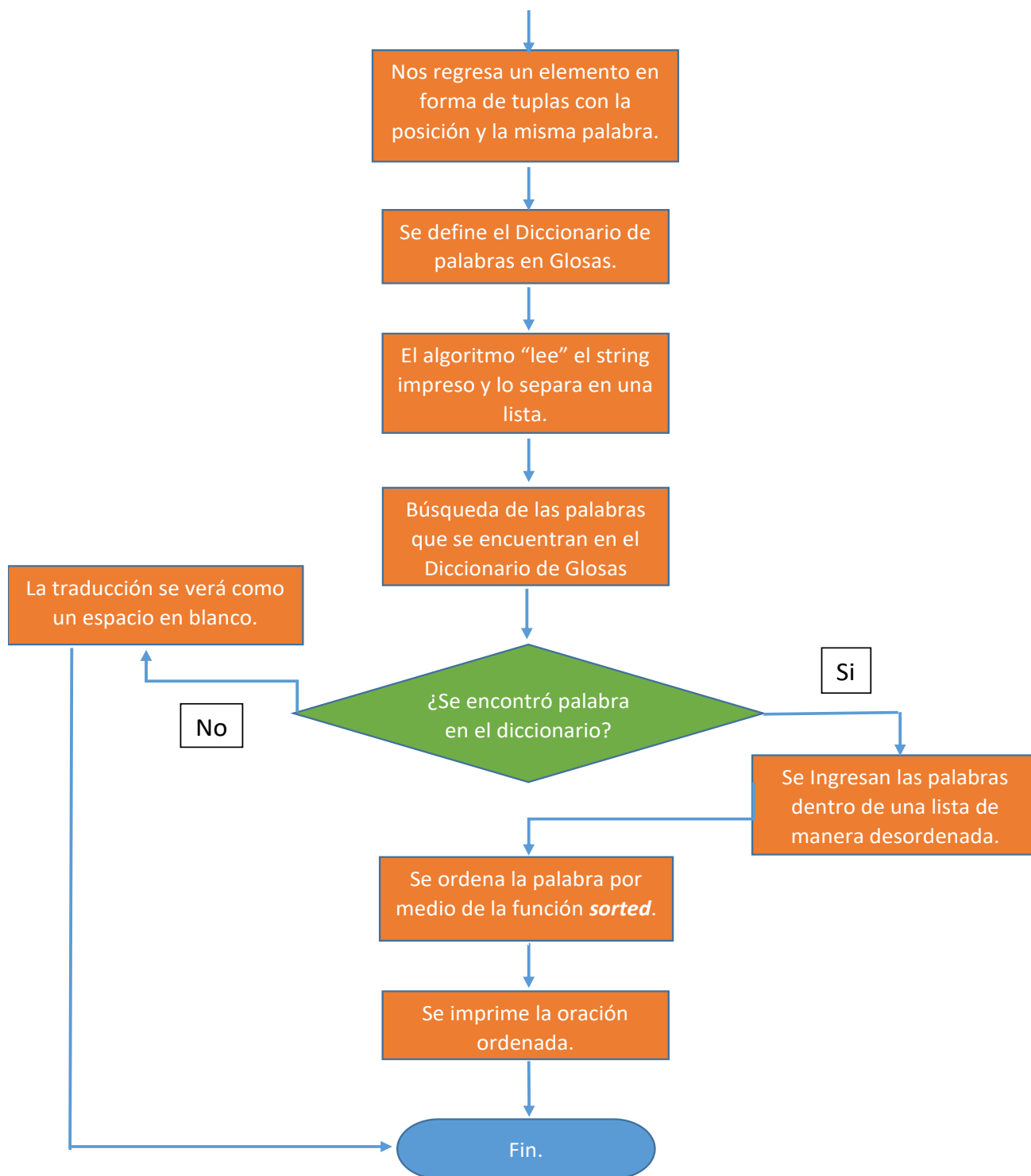


Ilustración 12. Flujograma del funcionamiento del código.

Fuente: (Realizada por el autor)

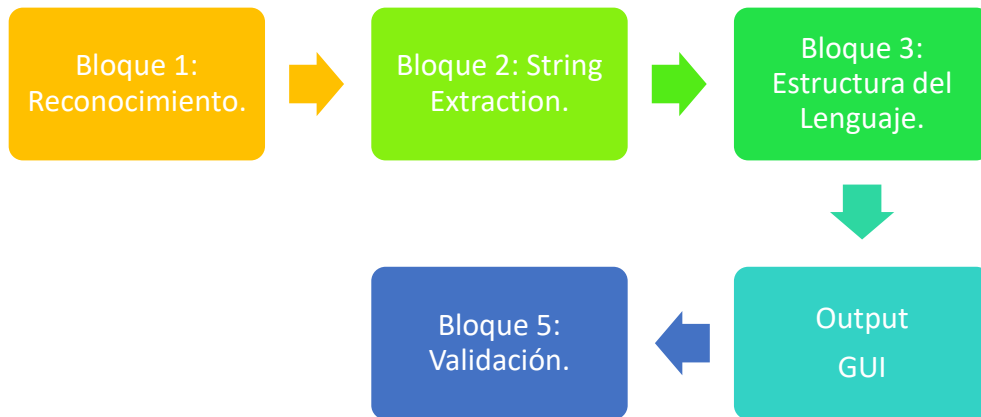


Ilustración 13. Diagrama de Metodología para traducción de voz a texto de Glosas.

Fuente: (Realizada por el autor)

4.5 Reconocimiento

El reconocimiento de voz es el primer paso en el proceso de traducción de palabras a texto en la estructura del lenguaje LESH0. El reconocimiento de voz es un proceso en el que una computadora o dispositivo graba el habla humana y lo convierte a formato de texto.

Para poder realizar el reconocimiento de voz se hace uso de un programa de creación del código llamado Visual Studio Code, y otro programa de interpretación que en este caso es Python.

La ejecución del reconocimiento está sustentada por el uso de 2 librerías una llama **pyAudio** la cual ayuda a poder escuchar el sonido mediante el uso de un micrófono, seguidamente se utiliza la librería de **speech_Recognition** que es la encargada de realizar el reconocimiento de las palabras captadas por la entrada que es el micrófono.

4.6 STRING EXTRACTION

Se le conoce como String Extraction a la extracción de las frases en texto de las palabras reconocidas en el primer paso de la traducción. Para desarrollar el algoritmo se necesita de un diccionario de traducción del cual podamos extraer las palabras reconocidas, este "diccionario" es en este caso la librería de **Google**.

En este proceso de extracción del string es importante destacar que la ejecución de este ignorará los signos de puntuación ya sea una coma o un punto, el texto se irá escribiendo de manera consecutiva únicamente separando las palabras por un espacio.

4.7 ESTRUCTURACIÓN DEL LENGUAJE

Se podría pensar que la estructura de la Lengua de señas de cada país es una transliteración de la estructura de la Lengua hablada en el mismo país, esto es un concepto erróneo; el lenguaje de señas de Honduras es totalmente indistinto al hablado comúnmente en el país, por lo tanto la estructura también lo es.

4.7.1 Sistema de Glosas

La lengua de señas es ágrafa, no tiene escritura, lo cual no afecta su estatus de lengua, pero para el estudio y análisis de la misma, se ha creado un sistema o sistemas, según el cual, usando palabras escritas del idioma nacional se acerca a la estructura gramatical que usan los sordos.

A este sistema o sistemas de representación escrita son conocidas como "Glosas", bajo este concepto se realiza la traducción de voz a texto en Glosas.

4.7.2 EXPLICACIÓN DE ESTRUCTURA EN CÓDIGO (LISTAS Y POSICIONES).

```
1 nombre = input("Cual es tu nombre?")
2 edad = input("Cual es tu edad?")
3 print("Bienvenido a la prueba del proyecto fase 1")
4 from datetime import datetime
5 current_date= datetime.now()
6
7 print(current_date)
8
9 import speech_recognition as sr # importar libreria
10
11 r = sr.Recognizer() #inicia el reconocimiento
12 with sr.Microphone() as source:
13 print("Di algo :")
14 audio = r.listen(source)
15 print("Listo!")
16 try:
17     text = r.recognize_google(audio, language= "es-HN")
18     print("Dijiste : {}".format(text))
19     print("Fin de la traduccion!!!")
20
21 except:
22
23     print("Lo siento no te entendi")
24
25 def find(objetivo, lista):
26     for posicion, elemento in enumerate(lista):
27         if elemento.lower() == objetivo.lower():
28             return (posicion,elemento)
29     return(None,None)
30 diccionario= []
31
```

```
32 diccionario.append('difícil')
33 diccionario.append('color')
34 diccionario.append('su')
35 diccionario.append('lugar')
36 diccionario.append('proyecto')
37 diccionario.append('tarea')
38 diccionario.append('cumpleaños')
39 diccionario.append('favorito')
40 diccionario.append('ese')
41 diccionario.append('nuestro')
42 diccionario.append('cuando')
43 diccionario.append('cuál')
44 diccionario.append('dónde')
45 diccionario.append('ojalá')
46 diccionario.append('bien')
47
48 oracion = text.split()
49
50 posiciones = []
51
52 oracion = [palabra for palabra in oracion if find(palabra,
    diccionario) != (None, None)]
53 for palabra in oracion:
54     pos, _ = find(palabra, diccionario)
55     posiciones.append(pos);
56
57 oracion_ord = [palabra for _, palabra in sorted(zip(posiciones,
    oracion))]
58
59 print(oracion_ord)
60
```

En primer lugar se define **nombre = input("¿Cuál es tu nombre?")** esto se da en la línea 1, con la intención de poder tener un registro de quien está siendo el sujeto de prueba.

También en la línea 2 se definió **edad = input("¿Cuál es tu edad?")** para colocar la edad del sujeto y así poder demostrar que el audio puede procesar diferentes frecuencias

Se coloca en la línea 3 del código **print("Bienvenido a la prueba del proyecto fase 1")** para dar la bienvenida.

Como se puede visualizar en la línea 4, para obtener la fecha actual se importa de la librería `datetime` a través de **from datetime import datetime.**

Ahora en la línea 5 se creó una variable **current_date= datetime.now()** la cual nos permitirá obtener la fecha del día de hoy e imprimirla con la función **print(current_date)** en la línea 7.

Se importa **speech_recognition** como `sr.` en la línea 9.

Hay que tener en cuenta que se tiene **speech_recognition** en dicho formato, mientras que anteriormente se instaló con el nombre de **SpeechRecognition**, por lo que se debe echar un vistazo a los casos porque Python distingue entre mayúsculas y minúsculas.

Ahora se ha utilizado como notación porque escribir **discursos de reconocimiento** completos no es la manera más práctica de realizarlo.

Ahora en la línea 11 se tiene que inicializar **r = sr.Recognizer ()**, esto funcionará como un reconocedor de voz.

Entonces en la siguiente línea, la línea 12, se define **con sr.Microphone () como fuente:** lo que significa que se ha inicializando la fuente a `sr.Microphone`, también se puede usar algunos archivos de audio para convertirlos en texto, pero en este proyecto se utilizó la voz de Micrófono.

A continuación, se imprimirá una declaración simple que recomienda al usuario hablar cualquier cosa, esto en la línea 13.

Ahora en la línea 14 se utiliza el **comando r.listen (fuente)** y el algoritmo procede a escuchar la fuente, por lo que hará este proceso en la fuente y la almacenará en el audio.

Una vez que el que termine de escuchar a la fuente se utiliza **print ("¡Listo!")** para notificar que el audio fue escuchado, esto definido en la línea 15.

Puede suceder que en algún momento el audio no esté claro y que no lo obtenga correctamente, por lo que se debe poner dentro del bloque **try** y **except**.

Entonces en la línea 17, dentro del bloque de prueba, el texto será **text = r.recognize_google (audio)**, ahora se tienen varias opciones como **Recognize_bing ()**, **Recognize_google_cloud ()**, **Recognize_ibm ()**, etc. Pero para este está usando **Recognize_google ()**. Y finalmente se procede a pasar nuestro audio. Y esto convertirá el audio en texto.

Ahora solo se tiene que imprimir **print ("Usted dijo: {}". Formató (texto))**, esto imprimirá lo que haya dicho, dicha función se encuentra en la línea 18.

En el bloque excepto se puede simplemente escribir **print ("Lo siento, no te entendí")**, esto le enviará un mensaje si su voz no está grabada claramente, definida así en la línea 23.

Como primer paso de la parte del reordenamiento de la oración en la línea 25, se utilizó la sentencia **def**, la cual es una definición de función que nos sirve como usuarios a crear objetos funciones definidas por nosotros mismos, en este se realiza una iteración por todos los elementos de la lista, y pidiendo su posición en ella por medio de la función **find(objetivo, lista)**.

Como se puede visualizar en la línea 26 tenemos un bucle **for** y la palabra **posición**, en este caso es el número correspondiente de posición de la palabra dentro de la oración. El elemento sería la palabra como tal, la función de **enumerate()** nos realiza una iteración de un elemento en nuestra lista para poder regresar otro en forma de tuplas en el cual se encuentran en este caso, el elemento o palabra y la posición de la misma.

La línea 27 indica que si el elemento es igual al objetivo de la búsqueda esta se termina y nos regresará como resultado al valor y su posición como se indica en la línea 28.

En la línea 29 **return(None, None)** esto significa que el elemento no está en la lista, por lo que no regresará ningún elemento.

En la línea 30 se define lo que es diccionario como una lista el cual se llena con palabras que se encuentran dentro del lenguaje LESH0, las palabras fueron agregadas con una función llamada

append el cual agrega un solo elemento a la lista existente. No devuelve una nueva lista de elementos, pero modificará la lista original al agregar el elemento al final de la lista. Después de ejecutar el método **append** en la lista, el tamaño de la lista aumenta en uno.

En la línea 48 la variable de **oración=** leerá lo que se imprimió durante el proceso de captación de audio a texto y separará el string en una lista por medio del método **split()**. Puede especificar el separador, el separador predeterminado es cualquier espacio en blanco.

La lista donde se guardarán las posiciones de las palabras encontradas en el diccionario LESH0 se dan por la definición de la variable **posición=[]** en la línea 50.

En la línea 52 se definirá la oración realizando un bucle continuo haciendo una búsqueda de palabra por palabra para conformar la nueva oración, esto si la búsqueda de la palabra en el diccionario es exitosa. Las palabras son extraídas en esta parte sin ningún orden.

En la línea 53 se empieza a realizar el ordenamiento de las palabras, esto realizando un bucle en la oración para realizar la búsqueda de la palabra por medio de **pos,_=find(palabra, diccionario)**. Y luego introduciendo las palabras en la lista de posición en la línea 55.

En la línea 57, aquí es donde se le da el orden a la oración dándole la variable **oracion_ord** que cumplirá con la condición que colocará cada palabra en orden por la función **sorted** que esta función devuelve una nueva lista que todos los elementos del diccionario en un orden ascendente. Devuelve un objeto zip cuyo método **.next ()** devuelve una tupla donde el elemento i-ésimo proviene del argumento iterativo i-ésimo. El método **.next ()** continúa hasta que se agota el iterativo más corto de la secuencia de argumentos y luego genera StopIteration.

Finalmente después de esto se construye la oración ordenada y por último el programa imprimirá la palabra ordenada a través de **print(oración_ord)** en la línea 59.

Tabla 3. Funciones utilizadas en el Algoritmo

Función	Descripción
Input()	La función input() permite al usuario introducir datos de distintos tipos desde la entrada estándar usualmente se utiliza la entrada del teclado para introducir texto.
Print()	En los programas, para que python nos muestre texto o variables hay que utilizar la función print(). La función print() permite mostrar texto en pantalla.
Datatime.now	Esta función nos da la fecha actual y la hora en la que estamos actualmente, con el formato YY/mm/dd y la hora en 24hrs.
Import	La sentencia import en Python es una declaración simple que permite importar código de otros ficheros. Este código se puede importar de ficheros que hayamos creado nosotros o paquetes externos, por ejemplo los de la librería estándar o los instalados con el administrador de paquetes (pip).
Try Except	El bloque de prueba le permite probar un bloque de código en busca de errores. El bloque except permite manejar el error.
Def return	La sentencia def es una definición de función usada para crear objetos funciones

	<p>definidas por el usuario. La sintaxis es de la sig. Manera:</p> <pre>def NOMBRE(LISTA_DE_PARAMETROS): """DOCSTRING_DE_FUNCION""" SENTENCIAS RETURN [EXPRESION]</pre> <p>RETURN, es la sentencia return en código Python.</p> <p>EXPRESION, es la expresión o variable que devuelve la sentencia return.</p>
Find()	<p>El método find () devuelve el índice de la primera aparición de la subcadena (si se encuentra). Si no se encuentra, devuelve -1.</p> <p>La sintaxis de esta es str.find (sub [, inicio [, fin]])</p> <p>Sub: es la subcadena que se debe buscar en la cadena str.</p> <p>Inicio y fin (opcional): la subcadena se busca dentro de str[start:end].</p>
Enumerate()	<p>La función incorporada (i.e. no necesita importarse) enumerate() toma como argumento un objeto iterable y retorna otro cuyos elementos son tuplas de dos objetos, el primero de los cuales indica la posición de un elemento perteneciente a it y el segundo, el elemento mismo.</p>
append()	<p>El método append () en python agrega un solo elemento a la lista existente. No devuelve una nueva lista de elementos, pero</p>

	<p>modificará la lista original al agregar el elemento al final de la lista.</p> <p>Después de ejecutar el método append en la lista, el tamaño de la lista aumenta en uno.</p> <p>La sintaxis de esta es: nombredeLista.append()</p>
split()	<p>El método Split se utiliza para separar un string. Puede especificar un separador, el separador predeterminado es cualquier espacio en blanco. La sintaxis de esta es: nombredeString.split()</p>
sorted()	<p>La función sorted que esta función devuelve una nueva lista que todos los elementos del diccionario en un orden ascendente. La sintaxis es de la siguiente manera: sorted(iterable, key=key, reverse=reverse)</p> <p>Iterable: La secuencia para ordenar, listar, diccionario, tupla, etc.</p> <p>Key: Una función a ejecutar para decidir el orden. El valor predeterminado es Ninguno.</p> <p>Reverse: Opcional. Un booleano. Falso ordenará ascendente, Verdadero ordenará descendente. El valor predeterminado es falso</p>

Fuente: (Realizada por el autor)

4.8 INTERFAZ GRÁFICO

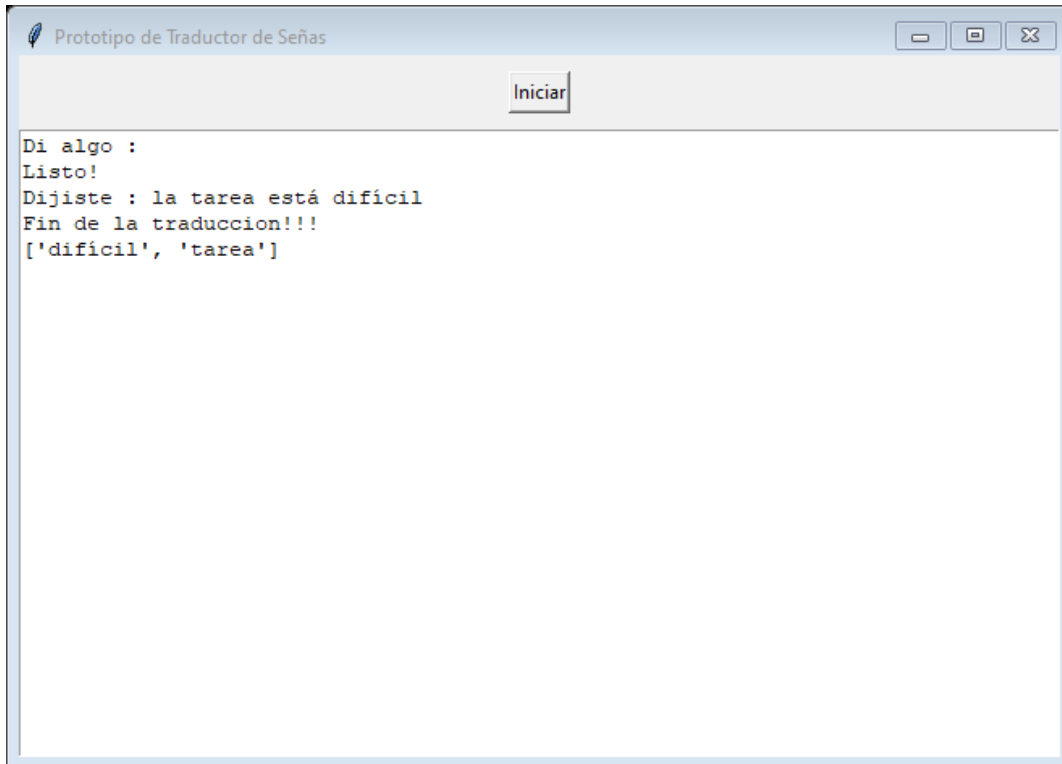


Ilustración 14. Captura del interfaz realizada por Visual Studio Code.

Fuente: (Realizada por el autor).

4.8.1 Creación

Para la realización de este interfaz se hizo uso de Visual Studio Code versión 1.41.1

Se importa la librería **tkinter** como *.

Se define la variable **raíz=TK()** en el que TK es un widget que representa principalmente la ventana principal de una aplicación. Tiene un intérprete Tcl asociado.

Utilizando el **raíz.title()** se le otorgo un nombre a la ventana generada.

Se creó el Frame para generar un widget de marco que puede contener otros widgets y puede tener un borde 3D. Y luego se empaquetó.

Se creó el objeto de texto y se empaquetó para que leyera el resultado que se genera en la terminal y así mostrarlo en el Frame.

Se definió una variable en la cual se colocó el código completo para el reconocimiento de voz y pasarlo a texto para así lograr obtener la traducción de la oración obtenida.

Se creó el botón dándole los diferentes parámetros para que se el que inicie el proceso del código de traducción y eso logro definiendo **comand=traduccion** la cual es la variable en la que se almacenó todo el código de tracción.

Y se cerró el con **raíz.mainloop()** el cual genera un flujo de control de alto nivel en un programa controlado por eventos. Es el bucle primario que recibe y distribuye los sucesos que actúan sobre el programa.

4.8.2 Ejecución

La ejecución de esta interfaz es muy amigable con el usuario, únicamente tendrá que iniciar el programa presionando **“iniciar”**, esto hará que el programa esté listo para escuchar las oraciones preseleccionadas, se procede a hacerlo y a esperar a que el programa ejecute el reordenamiento de las palabras.

4.9 Metodología de validación

Se realizaron 10 distintas pruebas de reconocimiento de voz mediante la lectura de 10 distintos párrafos de aproximadamente 75 palabras cada uno, encontrados aleatoriamente en internet.




Hablando del tiempo de respuesta el programa tarda un aproximado de en reconocer las palabras mediante voz y entregarnos el texto deseado.

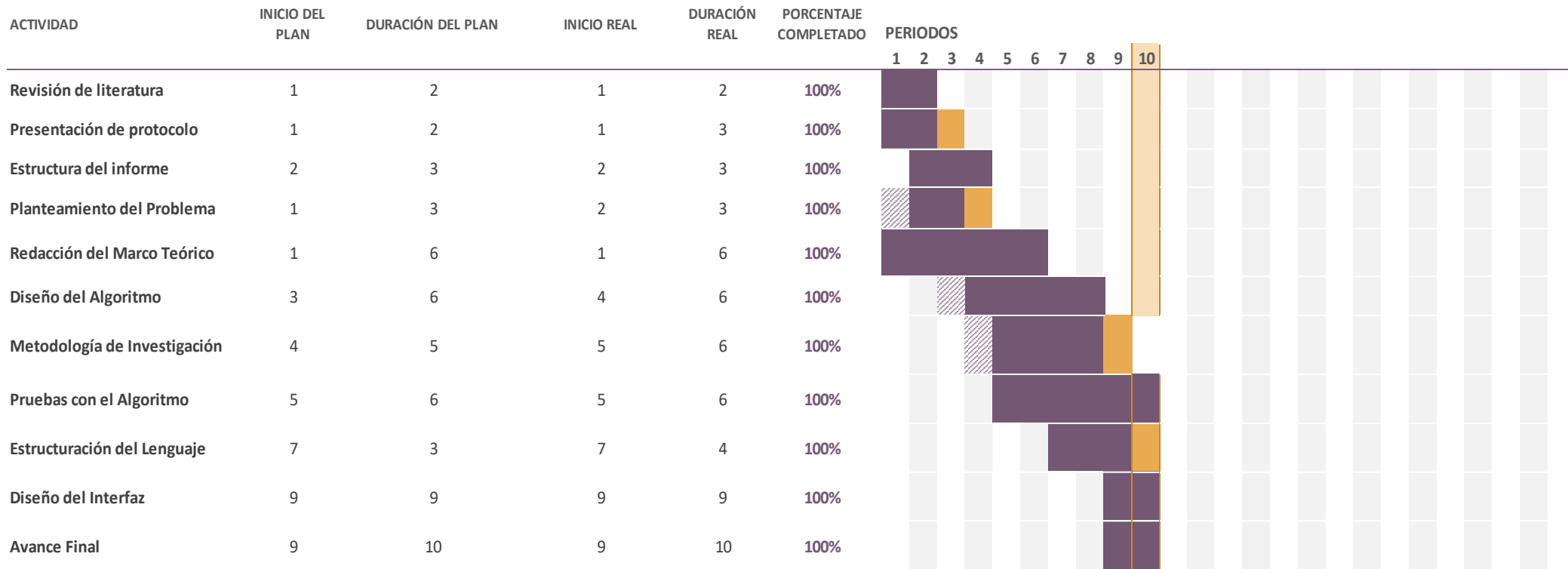
Durante la realización de estas pruebas se colocó a las personas en ambientes ruidosos y en ambientes con música para verificar la capacidad de detección bajo estas condiciones.

4.10 CRONOGRAMA DE ACTIVIDADES

Proyecto de Graduación Fase 1

ALGORITMO DE TRADUCCIÓN DE VOZ A TEXTO COMO HERRAMIENTA EDUCATIVA PARA PERSONAS CON SORDERA

Periodo resaltado 10  Duración del plan  % Completado  % Completado (fuera del plan)



Fuente: (Realizada por Autor)

V. RESULTADOS Y ANÁLISIS

5.1 VALIDACIÓN

Por la dificultad en la que está pasando el país y gran parte del planeta se hizo algo trabajoso encontrar sujetos de prueba para poder verificar la efectividad de los procesos de traducción, por lo que la muestra de prueba se tuvo que definir con 10 sujetos, se tomará en cuenta realizar más pruebas cuando la situación social y de salud del país se estabilice.

5.1.1 PRIMERA PRUEBA (RECONOCIMIENTO)

Se realizaron 10 distintas pruebas de reconocimiento de voz mediante la lectura de 10 distintos párrafos de aproximadamente 75 palabras cada uno, encontrados aleatoriamente en internet.

Durante la realización de estas pruebas se colocó a las personas en ambientes ruidosos y en ambientes con música para verificar la capacidad de detección bajo estas condiciones.

En esta prueba solo en 2 ocasiones hubo errores en una palabra, en distintos individuos, esto da una efectividad en el algoritmo de reconocimiento muy alta según estas pruebas, sin embargo habrá que realizar el resto de las pruebas para comprobar la veracidad de esta. La baja cantidad de palabras que no fueron reconocidas tienen su razón; se notó que al realizar estos ensayos los sujetos de prueba no realizaron una correcta pronunciación de las palabras, por lo que este factor es una variable para que el funcionamiento del programa pueda fallar.

Sin embargo se realizó posteriormente una correcta pronunciación de las palabras y el funcionamiento del algoritmo en el reconocimiento de voz pudo tener muy pocas fallas o ninguna según las pruebas realizadas.

Como se podrá notar después de la ejecución de las distintas pruebas, la elección de las librerías para este primer proceso de traducción tuvo un resultado muy positivo; por lo que Speech Recognition, Py Audio y Google Cloud Speech API cumplieron con las expectativas de ejecutar un correcto Reconocimiento de voz y su respectivo proceso de conversión a texto.

Tabla 4. Porcentaje de Efectividad de Reconocimiento.

Promedio de palabras en el párrafo:	75
Sujetos de prueba	Palabras Reconocidas
Sujeto 1	75
Sujeto 2	75
Sujeto 3	75
Sujeto 4	75
Sujeto 5	75
Sujeto 6	75
Sujeto 7	74
Sujeto 8	75
Sujeto 9	74
Sujeto 10	75
Total	748

Fuente: (Realizado por el autor).

5.1.2 PRUEBA 2 (ESTRUCTURACIÓN DEL LENGUAJE)

Se realizaron 10 pruebas para verificar la estructuración de las oraciones ingresadas anteriormente en nuestro diccionario de palabras para comprobar su correcto reordenamiento en la estructura de Glosas. Para la ejecución de este proceso se realizó la búsqueda de distintas funciones las cuales sirvieran para poder ordenar la oración identificada en nuestro primer proceso de traducción y lograr estructurarlo gramaticalmente a como lo expresan las personas con discapacidades auditivas en su estructura superficial. Para este proceso de reordenamiento de las palabras se ejecutan algunas funciones como ser: **enumerate()**, **.split**, **.deppend**, **sorted()**, entre otras funciones y sentencias para realizar el ordenamiento de la oración.

Durante estas pruebas no se reportaron fallas a la hora de probarlas con 10 distintas personas diciendo todas las oraciones preestablecidas en la Tabla 2. Estas pruebas nos dan un alto rango de efectividad en la traducción de estas palabras a la estructura de Glosas.

Luego de esto se hicieron un par de pruebas para determinar el tiempo de respuesta de nuestro algoritmo el cual oscila el valor según la cantidad de palabras a reconocer y ordenar, los valores varían entre 1.56 y 6.77 seg. Por lo que resulta ser un tiempo considerable para la respuesta a la respectiva traducción.

La efectividad de estas pruebas se da ya que las palabras mencionadas por las personas estaban incluidas y ordenadas de una manera correcta en el diccionario implementado en el algoritmo, de igual forma hubo una correcta pronunciación de las mismas, en caso contrario que el usuario mencionara una palabra que no se encuentre en el diccionario o se encuentre ingresada de una forma incorrecta en el diccionario, la traducción no se vería o se ejecutaría de una manera errónea. La ejecución del programa utilizando una correcta pronunciación y las palabras ingresadas en el diccionario ofrece una traducción a estructuración de Glosas bastante efectiva.

Tabla 5. Precisión de Estructuración en Glosas.

Sujetos de prueba	Oraciones: 10	
	Oraciones reconocidas	Errores durante la prueba
Sujeto 1	10	Ninguno
Sujeto 2	10	Ninguno
Sujeto 3	10	Ninguno
Sujeto 4	10	Ninguno
Sujeto 5	10	Ninguno
Sujeto 6	10	Ninguno
Sujeto 7	10	Ninguno
Sujeto 8	10	Ninguno
Sujeto 9	10	Ninguno
Sujeto 10	10	Ninguno
Total	100	

Fuente: (Realizada por el autor)

VI. CONCLUSIONES

- Se realizó una herramienta de traducción de voz a texto en el contexto del Lenguaje de señas de Honduras mediante la utilización de una computadora y un micrófono. Este algoritmo no funciona como apoyo en el área educativa para personas con discapacidades auditiva en la actualidad, sin embargo sirve como base para la creación de un traductor de voz a LESHO mediante algún programa de simulación de movimientos en 3D.
- Se diseñó un algoritmo de traducción a estructura del LESHO con la ayuda del editor de código Visual Studio Code versión 1.42.1 y el intérprete de Python versión 3.7.6, utilizando en estas las librerías que ofrece Python para realizar el reconocimiento de palabras y convertirlas en texto; para realizar este proceso se utiliza la librería de Google para extraer las palabras en el idioma de cada país, siendo en este caso el Español hablado en Honduras.
- Se implementó un interfaz para un uso más cómodo y versátil del algoritmo para el usuario. Para este se utilizó Visual Studio Code.
- En el algoritmo se creó un diccionario introduciendo 10 oraciones básicas para poder estructurarlas en forma de Glosas, estas se especificaron en la Tabla 2.
- Se comprobó su efectividad de interpretación y estructuración mediante pruebas a distintos usuarios resultando positivamente. Estas oraciones fueron preseleccionadas por el autor y con la ayuda de la Escuela Manos Felices se pudieron estructurar en Glosas.

VII. RECOMENDACIONES

Para futuros trabajos se recomienda considerar el LESHO como una lengua totalmente independiente del Lenguaje utilizado en el país, al intentar pasar las palabras comúnmente utilizadas al LESHO se debería considerar cómo una traducción de un lenguaje a otro, no como una transliteración del Lenguaje mismo, por lo que se debe tomar en cuenta ello.

Se debe de trabajar en conjunto con personas expertas en la Lengua de Señas del país, ya que de no ser así no se podrá comprobar la efectividad de los resultados obtenidos por futuros proyectos a realizar.

Debe escogerse el programa de edición y/o interpretación con el que pueda haber mayor comodidad para el autor del proyecto. Ya que al utilizar programas que resulten difíciles de manejar pueden incurrir en una pérdida de tiempo para el autor.

VIII. APLICABILIDAD E IMPLEMENTACIÓN

Este proyecto tiene aplicabilidad en el área de Educación para personas con discapacidades auditivas, podría implementarse en cada escuela del país para que todos los niños y niñas del país que sufren de esta discapacidad adquirida o congénita, tenga la oportunidad de recibir la misma calidad de educación que reciben los demás niños.

Se puede implementar en programas de televisión como noticieros o cadenas nacionales; esto para no depender de la disponibilidad de un traductor.

IX. EVOLUCIÓN DE TRABAJO ACTUAL

Como proyectos futuros puede implementarse en algún traductor de Voz a lenguaje de señas utilizando un programa de simulación de movimientos, con el cual pueda programar los movimientos con los que este Lenguaje cuenta.

Este programa sería una simulación de un avatar en 3D el cual pueda realizar los movimientos de las palabras ingresadas en el contexto de las glosas y así poder realizar una traducción exitosa de la Lengua nacional al de Lenguaje de señas de Honduras (LESHO). Existen varios Software que ofrecen el modelado de un avatar entre ellos se podía considerar Maya de Autodesk.

Se ampliará el diccionario para poder desarrollar más oraciones en Glosas, debido a la dificultad de disponibilidad por parte de los traductores y la problemática mundial, esto no pudo ser posible, sin embargo hay interés por parte de la Escuela Manos Felices y la ASH para el desarrollo del proyecto por lo cual han ofrecido su ayuda para la evolución del trabajo.

BIBLIOGRAFÍA

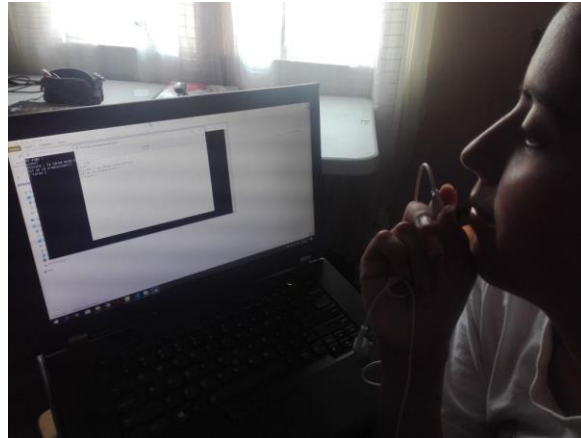
- 1) Abdallah, E. E., & Fayyoubi, E. (2016). Assistive Technology for Deaf People Based on Android Platform. *Procedia Computer Science*, 94, 295-301. <https://doi.org/10.1016/j.procs.2016.08.044>
- 2) Aguilar, O. (2020, marzo 12). *Entrevista con Escuela Manos Felices*. [Presencial].
- 3) Armenta, A., Escalada, J. G., Garrido, J. M., & Rodríguez, M. Á. (s. f.). Desarrollo de un corrector ortográfico para aplicaciones de conversión texto-voz. *2003*, 8.
- 4) ASALE, R.-, & RAE. (s. f.-a). *Ágrafo, ágrafa | Diccionario de la lengua española*. «Diccionario de la lengua española» - Edición del Tricentenario. Recuperado 5 de abril de 2020, de <https://dle.rae.es/ágrafo>
- 5) ASALE, R.-, & RAE. (s. f.-b). *Innato, innata | Diccionario de la lengua española*. «Diccionario de la lengua española» - Edición del Tricentenario. Recuperado 7 de mayo de 2020, de <https://dle.rae.es/innato>
- 6) Bustamante, F. R., & Díaz, E. L. (Mayo 2006con). *Spelling Error Patterns in Spanish for Word Processing Applications*.
- 7) *Discapacidad Auditiva*. (2019, mayo 30). Discapnet. <https://www.discalpnet.es/areas-tematicas/salud/discapacidades/auditivas/discapacidad-auditiva>
- 8) Iván. (2011, agosto 31). *Teorías sobre la adquisición y desarrollo del lenguaje en el bebé: El innatismo*. Bebés y más. <https://www.bebesymas.com/desarrollo/teorias-sobre-la-adquisicion-y-desarrollo-del-lenguaje-en-el-bebe-el-innatismo>

- 9) *Ley de la Lengua de Señas Hondureña (LESHO)*. (s. f.). Recuperado 7 de mayo de 2020, de <https://www.tsc.gob.hn/biblioteca/index.php/leyes/562-ley-de-la-lengua-de-senas-hondurena-lesho>
- 10) Martínez, A. (2020, enero 31). *Entrevista con personal de ASH* [Comunicación personal].
- 11) *¿Qué es una API y para qué sirve?* (2015, febrero 15). abc. <https://www.abc.es/tecnologia/consultorio/20150216/abci--201502132105.html>
- 12) *¿Qué significa Transliteración?* (2019, junio 5). Definiciones-de.com. <https://www.definiciones-de.com/Definicion/de/transliteracion.php>
- 13) *Repeticiones o bucles*. (2015, febrero 11). <https://desarrolloweb.com/articulos/repeticiones-bucles.html>
- 14) *Señas Honduras—Apps en Google Play*. (s. f.). Recuperado 30 de marzo de 2020, de https://play.google.com/store/apps/details?id=com.deskdust.hnlesho&hl=es_CL
- 15) Showleap Technologies. (2018, septiembre 24). *Guiadeprensa.com*. <https://www.guiadeprensa.com/suplementos/excelencia-empresarial-septiembre-2018-razon/showleap-technologies/>
- 16) *Significado de Congénito*. (2015, septiembre 7). Significados. <https://www.significados.com/congenito/>
- 17) *Significado de Inclusión social*. (2017, octubre 30). Significados. <https://www.significados.com/inclusion-social/>
- 18) *Significado de Script*. (2018, noviembre 15). Significados. <https://www.significados.com/script/>

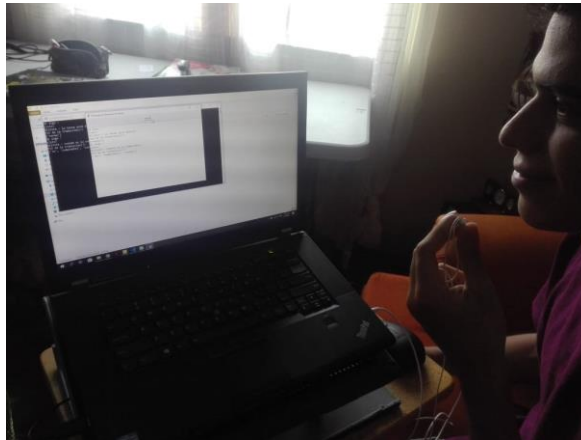
- 19) *Significado de Software (Qué es, Concepto y Definición)—Significados.* (2019, agosto 1).
<https://www.significados.com/software/>
- 20) *Significado y definicion de sordo, etimologia de sordo.* (s. f.). Recuperado 7 de mayo de 2020, de <https://definiciona.com/sordo/>
- 21) *Sordera y pérdida de la audición.* (s. f.). Recuperado 28 de enero de 2020, de <https://www.who.int/es/news-room/fact-sheets/detail/deafness-and-hearing-loss>
- 22) *Speech Recognition Python—Converting Speech to Text.* (s. f.). Recuperado 5 de abril de 2020, de <https://www.simplifiedpython.net/speech-recognition-python/>
- 23) *Tecnología que cambiará la vida de las personas con pérdida auditiva.* (2018, septiembre 6). <https://www.visualfy.com/es/cuatro-tecnologias-cambian-vida-personas-sordas/>
- 24) Telefónica, F. (s. f.). *Niños con discapacidad auditiva mejorarán su educación con tecnología.* Fundación Telefónica. Recuperado 9 de febrero de 2020, de <https://www.fundaciontelefonica.com.pe/2016/10/14/ninos-con-discapacidad-auditiva-mejoraran-su-educacion-con-tecnologia/>
- 25) *Tuplas: Listas inmutables. Python. Bartolomé Sintés Marco.* *Www.mclibre.org.* (2016, febrero 6). <https://www.mclibre.org/consultar/python/lecciones/python-tupla.html>
- 26) Valdivia, S. (2019). *HERRAMIENTA VIRTUAL PEDAGÓGICA DE LENGUA DE SEÑAS DE HONDURAS A NIVEL BÁSICO.*

ANEXOS

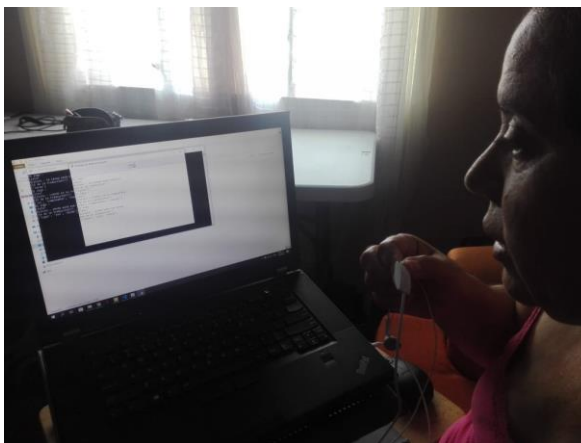
Anexo 1. Pruebas a sujeto 1



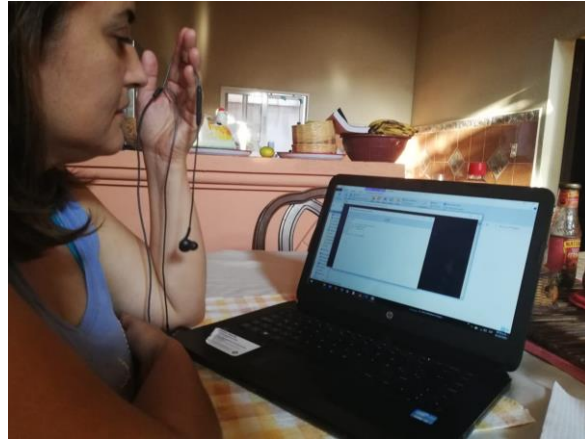
Anexo 4. Pruebas a sujeto 2



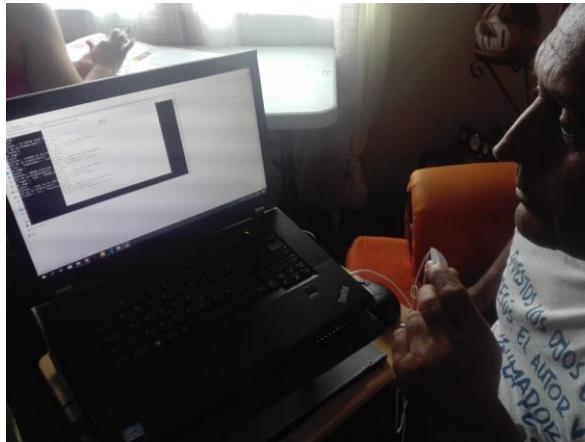
Anexo 7. Pruebas a sujeto 3



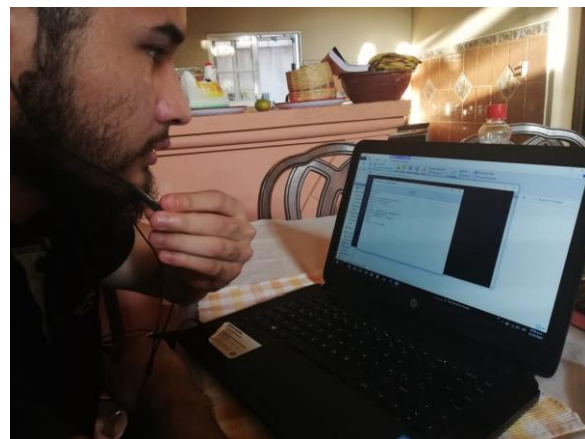
Anexo 16. Pruebas sujeto 4



Anexo 13. Pruebas a sujeto 5



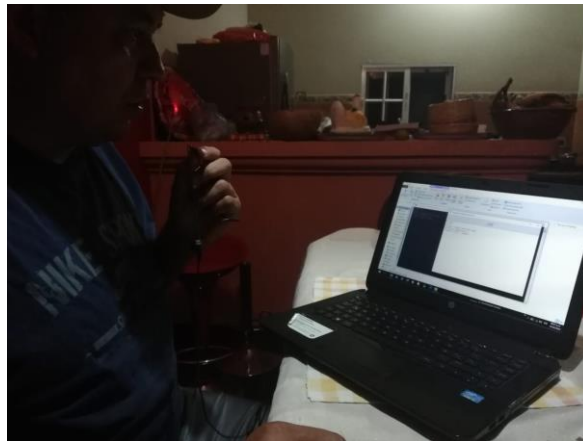
Anexo 10. Pruebas a sujeto 6



Anexo 19. Pruebas a sujeto 7



Anexo 25. Pruebas a sujeto 8



Anexo 22. Captura de pruebas realizadas

```
EXPLORER
C:\WINDOWS\py.exe
Prototipo de Traductor de Señas

Di algo :
Listo!
Dijiste : la tarea está difícil
Fin de la traducción!!!
['tarea']
Di algo :
Listo!
Dijiste : cuando es su cumpleaños
Fin de la traducción!!!
['su', 'cumpleaños', 'cuando']
Di algo :
Listo!
Dijiste : dónde está ese lugar
Fin de la traducción!!!
['lugar', 'ese', 'dónde']
Di algo :
Listo!
Dijiste : Cuál es su color favorito
Fin de la traducción!!!
['color', 'su', 'favorito', 'Cuál']
Di algo :
Listo!
Dijiste : Ojalá nuestro proyecto funcione bien
Fin de la traducción!!!
['proyecto', 'nuestro', 'Ojalá', 'bien']

Prototipo de Traductor de Señas
Iniciar

Di algo :
Listo!
Dijiste : la tarea está difícil
Fin de la traducción!!!
['tarea']
Di algo :
Listo!
Dijiste : cuando es su cumpleaños
Fin de la traducción!!!
['su', 'cumpleaños', 'cuando']
Di algo :
Listo!
Dijiste : dónde está ese lugar
Fin de la traducción!!!
['lugar', 'ese', 'dónde']
Di algo :
Listo!
Dijiste : Cuál es su color favorito
Fin de la traducción!!!
['color', 'su', 'favorito', 'Cuál']
Di algo :
Listo!
Dijiste : Ojalá nuestro proyecto funcione bien
Fin de la traducción!!!
```