Implementación de una inteligencia artificial capaz de crear una conversación en la plataforma del rostro animatrónico

Diego Sebastián Mazariegos Guzmán

UVG UNIVERSIDAD DEL VALLE DE GUATEMALA

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería

Implementación de una inteligencia artificial capaz de crear una conversación en la plataforma del rostro animatrónico

Trabajo de graduación en modalidad de Tesis presentado por Diego Sebastián Mazariegos Guzmán para optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

5 de diciembre 2023

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería

Implementación de una inteligencia artificial capaz de crear una conversación en la plataforma del rostro animatrónico

Trabajo de graduación en modalidad de Tesis presentado por Diego Sebastián Mazariegos Guzmán para optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

5 de diciembre 2023

Vo.Bo.:

(f) 

Ing. Kurt Kellner

Tribunal Examinador:

(f) 

Ing. Kurt Kellner

(f)) 

MAEB. Pablo Mazariegos

(f) 

M. Sc. Carlos Esquit

Fecha de aprobación: Guatemala, 13 de enero de 2024

Prefacio

La presente tesis, titulada *Implementación de una inteligencia artificial capaz de crear una conversación en la plataforma del rostro animatrónico*, representa una interacción más realista entre el rostro animatrónico y el usuario. Es importante señalar que no se desarrolló una inteligencia artificial desde cero, sino que se utilizó una ya existente, adaptándola y optimizándola para este propósito específico.

Este proyecto no solo representa un avance en la tecnología de animatrónicos, sino también un esfuerzo por integrar y aplicar conocimientos adquiridos durante mi formación académica. A lo largo de este proceso, he tenido el privilegio de contar con el apoyo incondicional de varias personas e instituciones, a quienes deseo expresar mi más profundo agradecimiento.

En primer lugar, quiero agradecer a mis padres, cuyo apoyo emocional y financiero fue fundamental para que pudiera llevar a cabo este proyecto. Su fe en mí y su constante aliento fueron pilares fundamentales en este camino.

Agradezco también a mi asesor, el ingeniero Kurt Kellner, cuyo conocimiento y guía fueron indispensables para obtener los resultados deseados. Su paciencia, orientación y disposición para ayudarme en todo momento han sido clave en la realización de este trabajo.

Finalmente, quiero extender mi agradecimiento a la Universidad del Valle de Guatemala, que me ha brindado una educación de calidad, permitiéndome desarrollar y aplicar los conocimientos adquiridos durante mi formación en este trabajo.

Tabla de contenidos

Lista de figuras

Lista de cuadros

Resumen

El objetivo de este trabajo de investigación es crear un método que permita la interacción entre el usuario y la inteligencia artificial mediante un micrófono. Para ello, se desarrolló un programa en Python utilizando las librerías *PyAudio*, *SpeechRecognition*, *pyttsx3*, *pydub* y *whisper*, siendo esta última fundamental para el reconocimiento de voz y la transcripción del habla a texto. Se logró implementar este método generando un archivo de tipo *Waveform* (WAV), que fue leído por la inteligencia artificial, comprobando así su comprensión del habla del usuario.

Además, se buscó que la inteligencia artificial accediera a datos relevantes de la Universidad del Valle de Guatemala y su Departamento de Ingeniería Electrónica, Mecatrónica y Biomédica. Para ello, se utilizó un programa con las librerías *LangChain* y *OpenAI*, destacando *LangChain* por su capacidad de leer y vectorizar archivos de texto, permitiendo que el modelo de lenguaje comprendiera el contenido. Esto facilitó el entrenamiento de la inteligencia artificial para responder preguntas sobre la información relevante.

Finalmente, se implementó un método de reentrenamiento a través de una interfaz amigable, desarrollada con la librería *TKinter*. La interfaz consta de un espacio para que el usuario introduzca información escrita y un botón "terminar"que genera un archivo de texto, permitiendo a la inteligencia artificial responder preguntas relacionadas con el contenido ingresado.

PALABRAS CLAVE: *Inteligencia artificial, micrófono, reconocimiento de voz, transcripción, librerías, entrenamiento, interfaz gráfica.*

Abstract

The objective of this research work is to create a method that enables interaction between the user and the artificial intelligence using a microphone. To achieve this, a program was developed in Python utilizing the libraries *PyAudio*, *SpeechRecognition*, *pyttsx3*, *pydub*, and *whisper*, with the latter being essential for voice recognition and transcribing speech to text. This method was successfully implemented by generating a *Waveform* (WAV) file, which was read by the artificial intelligence, thereby verifying its understanding of the user's speech.

Additionally, the goal was to allow the artificial intelligence to access relevant data from the University of the Valley of Guatemala and its Department of Electronic, Mechatronic, and Biomedical Engineering. To accomplish this, a program was used with the libraries *LangChain* and *OpenAI*, highlighting *LangChain* for its ability to read and vectorize text files, enabling the language model to understand the content. This facilitated the training of the artificial intelligence to answer questions regarding the relevant information.

Finally, a re-training method was implemented through a user-friendly interface developed with the *TKinter* library. The interface consists of a space for the user to input written information and a "finish"button that generates a text file, allowing the artificial intelligence to respond to questions related to the input content.

KEYWORDS: *Artificial intelligence, microphone, voice recognition, transcription, libraries, training, graphical interface.*

CAPÍTULO 1

Introducción

El objetivo principal de esta investigación consistió en la integración de una inteligencia artificial con la que el usuario sera capaz de interactuar a través del rostro animatrónico. Para lograrlo, fue necesario seguir una serie de pasos. El primer paso comprobó que la inteligencia artificial que se desea implementar es capaz de escuchar y comprender lo que el usuario le comunicó y de esta manera respondió de manera adecuada a cualquier solicitud del usuario.

Para lograr que la inteligencia artificial fuera capaz de comprender al usuario, fue necesario tomar dos puntos en cuenta. El primer paso consistió en la evaluación de la comprensión textual, para ello se realizó una transcripción de voz a texto también conocida como *speech-to-text*. De esta manera fue posible confirmar que la pregunta efectuada por el usuario fue captada palabra por palabra por la inteligencia artificial. El segundo paso generó una respuesta acorde al primero por medio de la voz de la inteligencia artificial, para ello se realizó una transcripción de texto a voz (también conocida como *text-to-speech*) de la respuesta textual generada. Es de esta manera que se comprobó por completo que la inteligencia artificial era capaz de comprender y responder acorde a la pregunta efectuada por el usuario.

Con los dos primeros pasos completados, fue posible continuar con el tercer punto en el cual se realizó el reentrenamiento de la inteligencia artificial con información de la Universidad del Valle de Guatemala y el Departamento de Electrónica, Mecatrónica y Biomédica. Para ello fue necesario agregarle la información de la Universidad a la inteligencia artificial a través de un archivo de texto. El objetivo consitió en tener a disposición la información de la Universidad del Valle y la información con la que la inteligencia artificial ya cuenta. Finalmente, como último punto se encontró el método más práctico con el que el usuario era capaz de realizar un reentrenamiento de la inteligencia artificial. Esto a través de la implementación de una interfaz gráfica en la cual el usuario fue capaz de agregar información en forma textual que a su vez debido al uso de la librería "LangChain". Fue posible una lectura por medio de la vectorización del archivo tipo texto que como se menciono anteriormente se hizo parte de la información de la que la inteligencia artificial tiene a su disposición.

CAPÍTULO 2

Antecedentes

En este capitulo se mencionan trabajos realizados por profesionales de la Universidad del Valle de Guatemala pero también por parte de personas externas a la Universidad. Dichos trabajos fueron de indispensable ayuda como referencia para la implementación de una inteligencia artificial al rostro animatrónico. Asimismo, algunos proporcionan ideas que podrían implementarse a manera de perfeccionar la interacción con el usuario.

2.1. *Sophia por Hanson Robotic's*

La página oficial de Hason Robotics (2023) afirma que Sophia es un robot con apariencia humana, que aplica robótica avanzada e investigación en inteligencia artificial creado por Hanson-Robotics. Su objetivo es hacer de Sophia un agente que explora la experiencia humano-robot en aplicaciones de servicio y entretenimiento aplicando inteligencia artificial la cual combina trabajos de redes neuronales, percepción artificial, procesamiento conversacional del lenguaje natural, entre otros.

La combinación de redes neuronales, percepción artificial y el procesamiento conversacional del lenguaje natural le permite a Sophia reconocer rostros humanos, ver expresiones emocionales y reconocer diversos gestos con las manos, tener respuestas únicas según la situación o la interación, estimar los sentimientos del usuario durante una conversación, tener emociones propias y controlar manos, mirada y estrategia de locomoción (solucionadores IK). Esta complejidad permite que Sophia sea capaz de establecer conexiones emocionales y mantener conversaciones significativas con las personas, brindando una percepción que sea más fácil de asimilar para los seres humanos (Hanson Robotics, 2023, Sophia's Artificial Intelligence).

2.2. Un asistente de voz de ChatGPT con quien puedes comunicarte - Código abierto: Vivy

En la implementación de este proyecto se usó el modelo *GPT-3.5 Turbo* de *OpenIA*, de esta manera fue posible entablar conversaciones personalizadas con los usuarios tomando en consideración su estilo de comunicación preferido. Además, es importante mencionar que durante la implementación de este trabajo GPT-3.5 Turbo fue la base para *ChatGPT*, este proyecto comparte esencialmente su modelo subyacente (Canal Jarods Journey, 2023).

El objetivo principal de este proyecto consistió en el desarrollo de un asistente personal capaz de emular interacciones similares a las humanas para una interacción más amigable al usuario. Además, se consideró que el asistente personal desarrollado fue de voz, es decir, que se generó una voz propia. Por lo tanto, no solo fue posible escuchar a ChatGPT comunicarse con el usuario a través de voz sino que el usuario también se comunicó con ChatGPT por medio de la voz propia (Canal Jarods Journey, 2023).

2.3. Crea tu propio clon de voz de texto a voz *(Text-to-Speech, TTS, por sus siglas en inglés)*

En este proyecto se muestra la manera de realizar grabaciones de voz para entrenar un modelo de texto a voz pero debe tomarse en cuenta que para lograrlo se necesitaron muestras de voz, por lo que el usuario que desarrolló este proyecto recomienda utilizar *Mycroft Mimic Studio*. Las tecnologías de Mimic de código abierto *Mycroft* son motores de conversión de texto a voz que toman un fragmento de texto escrito y lo convierten en audio hablado. Utilizando esta herramienta se simplifica la recolección de datos de entrenamiento de individuos, donde cada uno de los cuales se pueden utilizar para producir una voz distinta para *Mimic* (Canal Thorsten-Voice, 2021).

Ahora bien, una vez se toman las muestras de voz, el autor exportó la base de datos utilizando una herramienta llamada *DBeaver*, la cual se puede describir brevemente como un conector de bases de datos universal. A través de ella es posible por lo que podemos conectar a cualquier base de datos. Se realiza el post procesamiento de audio donde se configura el tipo de señal auditiva que queremos. En este caso se configura una señal mono fónica, así como la frecuencia de muestreo fue de 22,050 Hz (Canal Thorsten-Voice, 2021).

La configuración tanto de la frecuencia de muestreo como del tipo de señal de audio se realizan a través de un programa de audio que en este proyecto en específico es *Audacity*. Después de ello, se hace uso de una librería llamada *Coqui's TTS*, la cual ofrece la posibilidad de incluso hasta 20 idiomas en la generación avanzada de texto a voz. Por lo que se genera la configuración respectiva para la librería para la estadísticas de conjunto de datos informativos (Canal Thorsten-Voice, 2021).

Se corre el entrenamiento utilizando una herramienta llamada *Tacotron 2* que usa 2 redes neuronales, una encargada de convertir el texto en un espectrograma, es decir, una representación visual de frecuencias de audio en el tiempo y la otra apodada *WaveNet* encargada de leer el espectrograma y generar la reproducción del audio correspondiente. Una vez se termi-

ne de usar *Tacotron 2*, se procede a utilizar otra herramienta llamada *TensorBoard* (kit de herramientas de visualización de *TensorFlow*) para observar los espectrogramas generados por *TensorBoard* (Canal Thorsten-Voice, 2021).

Al final, se sintetiza el texto con el modelo entrenado a través de la interfaz de línea de comandos donde se configuró el texto que se deseaba sintetizar, la ubicación deseada para guardar la sintentización del texto, entre otras cosas. Una vez realizadas esas configuraciones, se adquiere una dirección web donde se podrá escuchar la sintetización del texto respectivo (Canal Thorsten-Voice, 2021).

2.4. Diseño e Implementación de un Sistema para Reconocimiento Facial, de Gestos y de Voz para un Rostro Animatrónico por el ingeniero Keneth Daniel Gonzáles Gonzáles

En la investigación de Keneth Gonzáles (2020) se tuvo como objetivo general la implementación mediante software un programa que tuvo dos modos de operación para un rostro animatrónico, detección de emociones y modo informativo.

Primero definió las expresiones faciales para un grupo específico de emociones (alegría, sorpresas, ira, tristeza, asco y miedo). Luego, se implementó un algoritmo para la identificación del rostro. Para la captura de marcas de rostro empleó un algortimo de 68 puntos junto con una base de datos *shape predictor 68 face landmarks.dat*. Para el reconocimiento y síntesis de voz empleó una librería de habla a texto (Gonzáles Gonzáles, 2020).

Logró la detección del rostro mediante la captura de video en tiempo real, que a su vez, mediante un código implementó las marcas de rostro a la imagen capturada a través de una cámara web donde el algoritmo y la base de datos detectaron las marcas de rostro. A través de programa que mostraba en la pantalla la emoción identificada y utilizando un modelo entrenado, obtuvo el resultado correcto con excepción de la emoción disgusto (Gonzáles Gonzáles, 2020).

Entonces, mediante una librería de habla a texto, se identificaron palabras u oraciones que a su vez fueron transcritas para que la comprensión del computador. La síntesis de voz se empleó la librería *puttsx3*, la cual se limita a leer frases, tomando un paquete de voz instalado (Gonzáles Gonzáles, 2020).

Con respecto al modo informativo logró apreciar la compresión de palabras y la función de situar y transmitir la posición de los ojos. Pero respecto al modo reactivo, el rostro reconoce una expresión facial acompañada de alguna frase o palabra y reaccionará reproduciendo algunas de las respuesta predefinidas. Se logró así la implementación de reconocimiento facial, de gestos y voz, por medio de algoritmos a través de software (Gonzáles Gonzáles, 2020).

2.5. Implementación de un chatbot a través de reconocimiento de voz en tiempo real entre el usuario y el rostro animatrónico de la Universidad del Valle de Guatemala por el ingeniero Daniel Eduardo Fuentes Oajaca

El objetivo de Daniel Fuentes (2021) fue desarrollar e implementar un sistema de interación verbal en tiempo real entre un usuario y el rostro animatrónico de la Universidad del Valle de Guatemala.

Primero implementó un Chatbot que usaba software con inteligencia artificial en un dispositivo, sitio web u otras redes para medir las necesidades de los consumidores y así ayudarlos a realizar una tarea en particular. Se emplearon redes neuronales para el procesamiento de información, y el aprendizaje supervisado como tipo de *machine learning*, tomando a *Python* como la herramiento de programación. El reconocimiento de voz utilizó un módulo llamado *speech recognition* de Python que identifica la información proveniente del micrófono. Sucesivamente fue necesario el uso de varias herramientas para machine learning: *TensorFlow* y *Keras* (Fuentes Oajaca, 2021).

Los resultados empezaron con un modelo de redes neuronales. Para ello se empleó un archivo que actuó como base de datos con respuestas generadas, gracias al proceso de lematización para el entrenamiento del modelo. Consecuentemente, para un manejo más sencillo de la data se empleó una serialización de objetos (listas). Para poder agilizar el proceso de entrenamiento sin importar la cantidad de texto, se empleó una lista de entrenamiento. Se definió el modelo de red neuronal del tipo secuencial así como también se determinó el tiempo de entrenamiento del modelo según el número de tópicos. Ahora bien para el uso de texto a voz se utiliza la librería *Pyttsx3*, utilizando funciones específicas para el procesamiento de entrada y la predicción de clase. El resultado fue la generación de una respuesta (Fuentes Oajaca, 2021).

Sucesivamente se desarrollá una interfaz gráfica utilizando como lenguaje de programación *Python* (librería Kivy). Se programaron algunos movimientos para representar el momento en que habla el bot. De este modo el autor logró los objetivos específicos ejecutando la reproducción de la voz y los movimientos del rostro; la implementación de una interfaz gráfica atractiva y amigable; mejorar la solidez de la estructura del rostro, así como la mejora de aspectos realizados desde la primera investigación; el reconocimiento de temas de conversación, tiempo requerido para el entrenamiento del modelo y la implementación de una librería para la sintetización de voz (Fuentes Oajaca, 2021).

CAPÍTULO 3

Justificación

La implementación de una inteligencia artificial al rostro animatrónico será un paso significativo tanto para la Universidad de Valle de Guatemala como para los futuros trabajos que esta desee implementar. El usuario interactuará de forma práctica, amigable y real, lo que le brindará aún más realismo al rostro animatrónico. Esta implementación será un acercamiento a una nueva tecnología, que brindará una amplia cantidad de conocimiento al usuario. Asimismo, algo que debe tomarse en cuenta es que existe una manera específica de interactuar con la inteligencia artificial, algo que también se explicará, con el fin de obtener la mejor respuesta a aquello que el usuario desee conocer.

En otras palabras, es importante adaptar las nuevas tecnologías lo antes posible, ya que estas presentarán algunas fallas al inicio. Por lo tanto, una implementación temprana ayudará a detectarlas y mejorar a las nuevas tecnologías. De este modo, el presente trabajo resolverá algunas de las dificultades que otras investigaciones presentaron. Una de ellas siendo la cantidad de posibles respuestas, acorde a la conversación con el usuario, debido a una limitación sobre una base de datos preexistente.

En síntesis, la implementación de una inteligencia artificial al rostro animatrónico le brindará una mayor capacidad de interacción con el usuario sin las limitaciones mencionadas.

CAPÍTULO 4

Objetivos

Objetivo general

Integrar una inteligencia artificial que sea capaz de interactuar con el usuario a través del rostro animatrónico.

Objetivos específicos

- Crear un método que permita al usuario interactuar con la inteligencia artificial implementada por medio de un micrófono.
- Crear un método por el cual la inteligencia artificial implementada sea capaz de responder a la interacción del usuario por medio de unas bocinas.
- Entrenar a la inteligencia artificial con datos relevantes de la Universidad y el Departamento de Ingeniería Electrónica, Mecatrónica y Biomédica demostrando la capacidad que tiene esta de aprender.
- Implementar un método para reentrenar la inteligencia artificial a través de una interfaz amigable, mostrando su capacidad de adquirir todo tipo de información que el usuario desee.

CAPÍTULO 5

Alcance

Con el presente trabajo se lográ la implementación de un programa que buscaba la unificación entre un rostro animatrónico y una inteligencia artificial ya existente, que en este caso es Chat GPT. En esta investigación no se creó una inteligencia artificial desde cero, sino se entrenó con información de la Universidad del Valle de Guatemala y el Departamento de Ingeniería Electrónica, Mecatrónica y Biomédica. Esto no significa que esta sea toda la información a la que se tiene acceso sino que dicha información se suma a toda la información de la que ya dispone Chat GPT.

Para lograrlo se desarrollo un método que permitió el reentrenamiento de la inteligencia artificial con información a elección del usuario. Sin embargo, en un inicio la capacidad para el reentrenamiento estaba limitada a usuarios específicos. Por ello, con el cuarto y último objetivo específico se buscó ampliar la accesibilidad a cualquier usuario. Esto fue posible gracias al desarrollo de una interfaz en la cual el usuario pueda colocar la información que desee.

La manera en la que el usuario puede aplicar el método de reentrenamiento a la inteligencia artificial es por medio de un archivo de texto. Es decir, que no podrá aplicarse este método por medio del habla sino únicamente a través de la interfaz gráfica en la que debe colocarse de forma escrita la información deseada. Finalmente, el alcance del presente trabajo consistió en la implementación de una inteligencia artificial ya existente y el reentrenamiento por medio de una interfaz gráfica en el que la información debe ser agregada por escrito, lo que a su vez es mucho más amigable para el usuario que administrar la información deseada por medio del programa desarrollado.

CAPÍTULO 6

Marco teórico

Inteligencia Artificial

La inteligencia artificial es la ciencia e ingeniería de fabricar máquinas inteligentes, especialmente programas informáticos inteligentes. Este concepto está directamente relacionado con la tarea de usar computadoras con el fin de comprender la inteligencia humana, sin embargo, no tiene por qué limitarse a métodos que sean biológicamente observables. Debe tomarse en cuenta que en la definición de este concepto no se especifica que se desea simular la inteligencia humana, sin embargo, sí es posible hacer que las máquinas resuelvan problemas observando a otras personas (McCarthy, 2007).

Los investigadores o desarrolladores de inteligencia artificial son libres de usar métodos que no necesariamente se observan en las personas o incluso llegan a involucrar mucha más computación de la que una persona promedio puede hacer. Consecuentemente, la aplicación de la inteligencia artificial a este proyecto tiene como objetivo ampliar la capacidad de respuestas generadas dependiendo de la interacción con el usuario. No se busca la dependencia directa de una base de datos preexistente, sino ampliar la capacidad de responder a diferentes incógnitas por parte del usuario (McCarthy, 2007).

Por otro lado, debe conocerse que la inteligencia artificial simula redes neuronales con el propósito de aprender pero este concepto también abarca otro tema como lo es el aprendizaje automático (*machine learning*). En este punto se manejan dos conceptos nuevos; las redes neuronales y el aprendizaje automático. Estos conceptos se explicaran de una forma más detalla más adelante (Code.org, 2023).

Aprendizaje automático *(machine learning)*

Aprendizaje automático abarca una serie de áreas como las matemáticas, la probabilidad y estadística, información tecnológica e incluso física con el objetivo de encontrar respuesta a incógnitas complejas. La aplicación de todas estas genera que dicha tecnología examine

considerablemente rápido la cantidad de datos recolectados. El funcionamiento del aprendizaje automático radica en el uso de algoritmos para analizar grandes cantidades de datos, aprender patrones y predecir resultados de los datos administrados. Por lo tanto, se puede describir al aprendizaje automático como la forma en la que las computadoras reconocen patrones y toman decisiones que no necesariamente deben estar programadas (Alpaydin, 2020).

Al usar el aprendizaje automático, en lugar de programar una computadora paso por paso, es posible programar una computadora para que aprenda justo como lo hace una persona promedio por medio de la prueba y error en conjunto con mucha práctica. Asimismo, el aprendizaje viene de la experiencia y tomando en cuenta lo anterior se sabe que esto también aplica para el aprendizaje automático (Code.org, 2023).

En el caso del aprendizaje automático, la experiencia viene de muchos datos, que pueden ser de cualquier tipo: imágenes, vídeos, audio o texto. Una vez se le proporcionen los datos deseados, comenzará a reconocer patrones en dichos datos; al aprender a reconocer patrones en los datos, también será capaz de realizar predicciones basándose en los mismos patrones encontrados anteriormente. Por ejemplo, logrando reconocer o distinguir la diferencia entre dos imágenes completamente distintas (Code.org, 2023).

Se dice que el aprendizaje automático será tan bueno y preciso dependiendo de la cantidad de datos que se le administren, pero debe tomarse en cuenta la calidad de los mismos. Existe otro aspecto que debe entenderse y es la proveniencia de los datos que se busca administrarle al aprendizaje automático, para ello debemos estar consientes que existen computadoras que están recolectando constantemente datos de entrenamiento (*training data*) de todas las personas. Por ejemplo: actualmente muchos servicios de *streaming* toman en cuenta el contenido adquirido de cada usuario, reconociendo patrones para que después puedan realizar recomendaciones de contenido que podría gustarle al usuario (Code.org, 2023).

Existen ocasiones en las que podría a considerarse que se le pide ayuda directamente al usuario, un ejemplo de esto es que algunas veces ciertos sitios web pide que se reconozcan señales de tránsito. Lo que realmente hace el usuario cuando realiza estas acciones es proveer datos para ayudar al aprendizaje automático. Otro caso que puede mencionarse, es que los investigadores médicos pueden usar imágenes médicas como datos de entrenamiento (*training data*) para enseñar a las computadoras como reconocer y diagnosticar enfermedades. En otras palabras, el aprendizaje automático necesita de cientos de miles de imágenes y orientación médica de un doctor que sabe que debe buscar exactamente antes de reconocer de forma correcta una enfermedad. No debe olvidarse que aún con una cantidad considerable de datos pueden llegar a surgir problemas con las predicciones que realiza la computadora (Code.org, 2023).

Por otro lado, si llegase a darse el caso en que los datos administrados pertenezcan a hombres únicamente, entonces las predicciones solo aplicarían para hombres. Entonces puede llegar a darse el caso de no reconocer enfermedades para las mujeres. A este punto ciego en los datos de entrenamiento (*training data*) se le conoce como BIAS. Los datos BIAS favorecen algunas cosas y despriorizan o excluyen otras. Dependiendo de cómo los datos de entrenamiento estén siendo recolectados, quién esté realizando la recolección y cómo se esté alimentando a la computadora con estos datos, se obtendrán predicciones

correctas o incorrectas. Las consecuencias de que el aprendizaje automático aprende con datos BIAS, radica simplemente en que la computadora hará predicciones erróneas o sesgadas que dependiendo del escenario significarán problemas graves (Code.org, 2023).

Como recomendación para evitar esta posibilidad, cuando se este buscando o recolectando datos de entrenamiento debe tomarse en cuenta dos aspectos: primero lo es la cantidad de datos, que regirá la satisfacción y precisión de nuestro aprendizaje automático. El segundo aspecto es el rango que abarcan los datos recolectados, es decir, si el rango cubre todos los escenarios y usuarios posibles sin datos BIAS. Con base en esto, se sabe que depende del usuario recolector de datos de entrenamiento no darle a nuestro aprendizaje automático datos BIAS. Finalmente, no debemos olvidar que lo que en realidad se está haciendo cuando se está buscando y escogiendo datos para el aprendizaje automático es programar el algoritmo pero usando datos de entrenamiento en lugar de código. Este aspecto es aplicable a la presente investigación, ya que como se mencionó anteriormente, debe de administrarse la cantidad y calidad correcta tomando en cuenta todos los escenarios posibles para que podamos obtener los resultados esperados. Finalmente, existen tres tipos de aprendizaje automático (Code.org, 2023).

Tipos de aprendizaje automático:

- **Supervisado:** el algoritmo de aprendizaje automático se entrena en datos etiquetados, sin embargo, los datos deben etiquetarse con precisión para que este método funcione. El método funciona tras brindarle datos de entrenamiento al algoritmo y este tiene la función de darle una idea básica del problema, la solución y los puntos de datos a tratar.
- **No supervisado:** En este tipo se trabaja con los datos sin etiquetar que permite la creación de estructuras ocultas, es decir, que no se requiere trabajo humano para poder crear el conjunto de datos legible por máquina, por lo que puede trabajarse con conjuntos de datos más grandes. Asimismo, la relación entre los puntos de datos son percibidas por el algoritmo de manera abstracta, por lo que la participación de los seres humanos no es necesaria.
- **Por refuerzo:** Este se inspira directamente en cómo los seres humanos aprenden de los datos en sus vidas. Cuenta con un algoritmo que es capaz de mejorarse a sí mismo y aprende de nuevas situaciones con el método de prueba y error. Tomando en cuenta el concepto psicológico de condicionamiento, donde el algoritmo se coloca en un entorno de trabajo con un intérprete y un sistema de recompensas, se decide si el resultados es favorable o no.

Redes Neuronales

Con el fin de crear aprendizaje automático, los científicos realizaron estudios que les permitieron delimitar qué era lo mejor en el ámbito del aprendizaje y notaron que no hay nada mejor en este ámbito que el cerebro. Este está hecho de células especiales llamadas neuronas, las cuales tienen dos terminaciones, una encargada de recibir las señales y otra en la que salen las señales como una sola. Consecuentemente, se debe entender que las millones

de neuronas en nuestro cerebro están conectadas entre sí en lo que se conoce como una red neuronal biológica, esta es la manera en la que el cerebro procesa la información y reconoce patrones (Code.org, 2023).

Es de esta forma en la que se crearon las neuronas artificiales con software recibiendo múltiples señales de entrada, pasando a través de la neurona, combinándose y siendo procesadas a través de matemática simple que se convierte en una nueva señal. Sin embargo, una sola neurona no puede lograr demasiado, por lo tanto todo el potencial puede sacarse cuando las neuronas artificiales están conectadas creando una red neuronal artificial. Es esta tecnología la que hace capaz a las computadoras de reconocer imágenes e incluso crear arte extraño, pero aquello que las hace poderosas es la composición de neuronas en capas. Por ejemplo, hay capas de entrada, capas escondidas y capas de salida. Esta última representa la entrada de otra; muchos medios de música, compras, entre otros, usan este tipo de sistema. Este será un aspecto que se tendrá que se generó; una red neuronal artificial asociada a la conversión del habla a texto y de texto al habla, tomando en cuenta el objetivo general de esta investigación para que el aprendizaje automático sea correcto y por lo tanto proporcione un resultado satisfactorio para la inteligencia artificial (Code.org, 2023).

De Texto a Voz *(Text-to-Speech, TTS, por sus siglas en inglés)*

El significado de texto a voz es tan simple como se escucha: es la tecnología capaz de leer texto por medio de voz, la cual puede ser automatizada. Esta tecnología puede llegar a ser de gran utilidad para oyentes con discapacidad visual o discapacidades de aprendizaje basadas en el lenguaje. Además puede aumentar la eficiencia al permitir que los usuarios que dan uso a esta tecnología puedan realizar múltiples tareas. Asimismo, se puede observar a esta tecnología cuando en algunas ocasiones se interactúa con un asiste virtual y se recibe una respuesta verbal por parte de este, es precisamente la tecnología que impulsa esta respuesta de voz generada a la cual se le conoce como texto a voz *(Text-to-Speech, TTS)* por sus siglas en inglés. Esta tecnología está creciendo de manera significativa, ya que es capaz de ejecutar una canalización de texto a voz de extremo a extremo en unos pocos milisegundos para interacciones naturales, personalizar modelos y canalizadores de IA en el momento de la interferencia para generar una voz sintética expresiva, entre otros (Balajthy, 2005).

En una catalización de texto a voz de extremo a extremo, existen ciertos modelos y módulos clave que hacen posible la conversación entre un asisten virtual y el usuario. Uno de ellos es la normalización y preprocesamiento de texto, el cual se encarga de convertir números y abreviaturas en palabras. Otro es la codificación de texto, el cual convierte el texto en un vector codificado que es utilizado como una entrada para un generador de espectrogramas. El generador de espectrograma es otro de ellos, este se encarga de generar un espectrograma a partir de un vector de texto codificado. Finalmente, el modelo vocoder, que toma los espectrogramas como entrada y consecuentemente genera una voz sintética que puede reproducirse y percibirse por el usuario. En otras palabras, puede decirse que la tecnología texto a voz es la última etapa en aplicaciones como los son los asistentes virtuales, humanos digitales y robots de servicio (Balajthy, 2005).

En este caso, la aplicación de esta tecnología se centra en tomar una inteligencia artificial como base para la investigación. Dicha inteligencia generará las respuestas a las dudas de los

usuarios por medio de un texto, por lo que se necesitaba una herramienta capaz de convertir un texto en habla, como es el caso de la tecnología mencionada anteriormente (Balajthy, 2005).

De Voz a Texto *(speech-to-text, STT, por sus siglas en inglés)*

Este es un software de reconocimiento de voz que permite el reconocimiento y la traducción del lenguaje hablado a texto utilizando la lingüística computacional. Esta tecnología desarrollada por software también es conocida como reconocimiento de voz o reconocimiento de voz por ordenador, existen ciertos dispositivos con la capacidad de transcribir flujos de audio en tiempo real para mostrar texto y actuar con respecto a esto. El funcionamiento de este se desarrolla percibiendo un audio y entregando una transcripción literal, la cual puede ser editada en un dispositivo determinado. Esto lo hace gracias a un programa que se basa en algoritmos lingüísticos con el objetivo de clasificar las señales auditivas de las palabras habladas y transferir las señales percibidas por medio de este audio a señales de texto mediante caracteres llamados *Unicode* (Trivedi, Pant, Shah, Sonik y Agrawal, 2018).

La conversión de voz a texto funciona utilizando un complejo modelo de aprendizaje automático *(machine learning)* el cual consta de varios pasos. Para entender esto un poco mejor, se tomaron una serie de aspectos con mayor profundidad. Cuando los sonidos salen de la boca de una persona se crean palabras que a su vez producen una serie de vibraciones. La tecnología de conversión de voz funciona captando estas vibraciones para después traducirlas a un lenguaje digital a través de un convertidor analógico a digital. Entonces, el convertidor de analógico a digital toma los sonidos de un archivo de audio, mide las ondas a detalle y las filtra para distinguir los sonidos relevantes. Una vez se termina este proceso, los sonidos se segmentan en ya sean centésimas o milésimas de segundo para combinarse con fonemas. Se entiende a un fonema como una unidad de sonido que distingue a una palabra de otra con un idioma determinado. Estos fonemas se ejecutan por medio de una red a través de un modelo matemático donde se realizan comparaciones con oraciones, palabras y frases conocidas. Finalmente, se presenta como un texto o una demanda computacional basada en la versión más probable del audio (Trivedi, Pant, Shah, Sonik y Agrawal, 2018).

La integración directa con esta investigación toma en cuenta que para que la inteligencia artificial pueda generar un respuesta a las preguntas o dudas del usuario, debe percibir estas por medio de texto. Es en este punto en que la tecnología de voz a texto entra, ya que será la encargada de transcribir la voz del usuario a texto y que la inteligencia artificial consecuentemente pueda responder de forma adecuada (Trivedi, Pant, Shah, Sonik y Agrawal, 2018).

Transformers

Los denominados *transformers* son modelos de *deep* learning y su funcionamiento está enfocado en el procesamiento de lenguaje natural NLP (Natural/Language/Processi). En otras palabras, es capaz de reconocer las palabras que conforman una oración, de recordar el contexto o bien la relación que existe entre las palabras. La forma en la que los *transformers* hacen esto es por medio de las representaciones vectoriales de cada una de las palabras,

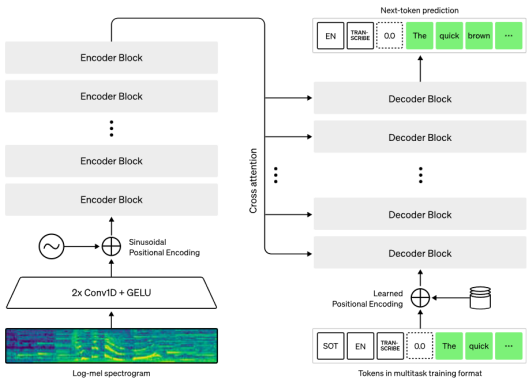
y por medio de varias redes neuronales realiza productos escalares de los dos principales vectores dentro de la oración. Un vector llamado *query* y otro llamado *key*, donde el vector *query* representa a la primera palabra de la oración y este se multiplica con cada uno de los vectores *key* dentro de las palabras de la oración. Para finalmente, obtener un número que mientras más grande sea más grande será la relación que exista entre esa palabra y el resto de la oración (Alpaydin, 2020).

Whisper

Whisper es un sistema de reconocimiento automático de voz (ASR) entrenado usando 680,000 horas de datos supervisados plurilingües y multitarea recopilados de la web. Es importante mencionar que permite la transcripción de varios idiomas, así como la traducción de estos al idioma inglés. Este sistema usa modelos de código abierto, así como código de interferencia, los cuales sirven como base para crear aplicaciones útiles y para futuras investigaciones sobre el procesamiento de voz más sólido. Además, alrededor de un tercio del conjunto de datos de audio de whisper no está en inglés y alternativamente se le asigna la tarea de transcribir en el idioma original o traducir al inglés (OpenAI, 2023).

Ahora bien, se describirá la Figura 1, que muestra una estructura de Whisper. Como se observa, primero se necesita un audio o un texto. Este será transformado para poder obtener una representación vectorial de cada una de las palabras. Para entonces calcular una codificación posicional dentro de cada palabra, los cuales son números binarios que representan la posición de cada palabra dentro del texto transcrito (OpenAI, 2023).

Figura 1: **Estructura de Whisper**



Nota: Adaptado de OpenAI (2023)

Estos números pasan por una capa de codificadores y el resultado es enviado a una capa de decodificadores, los cuales se encargan de calcular las probabilidades de nuevas palabras que se generan en base a las palabras originales codificadas. Tomando esto en cuenta, podría decirse que los decodificadores se encargan de realizar de las tareas más importantes, como lo son:

- Identificar el idioma y traducir el texto
- Identificar cuando hay patrones de ruido

- Identificar cuanto se está hablando y cuando sea el programa el que se esté comunicando con el usuario
- Transcribir texto en el caso de que el input sea un audio

Whisper es capaz de realizar estas cuatro tareas, mientras que puede que en otros casos se use un modelo enfocado en cada una estas tareas. Además, una ventaja que es importante mencionar es que Whisper es capaz de reconocer signos de puntuación, cosa que en muchas ocasiones era posible pero no con el grado detallado con el que cuenta Whisper (OpenAI, 2023).

Finalmente, es importante dar una explicación más detallada sobre la variedad de modelos de reconocimiento de voz. Dichos modelos se pueden encontrar en el repositorio de la librería donde se nos muestran 5 modelos diferentes, cuyas principales diferencias consisten en la cantidad de parámetros, el lenguaje o lenguajes en el que fueron entrenados los modelos, la vram rquerida y la velocidad de respuesta (OpenAI, 2023). Como se podra ver en la Figura 2:

Figura 2: **Modelos de Whisper**

Size	Parameters	English-only model	Multilingual model	Required VRAM	Relative speed
tiny	39 M	`tiny.en`	`tiny`	~1 GB	~32x
base	74 M	`base.en`	`base`	~1 GB	~16x
small	244 M	`small.en`	`small`	~2 GB	~6x
medium	769 M	`medium.en`	`medium`	~5 GB	~2x
large	1550 M	N/A	`large`	~10 GB	1x

Nota: Adaptado de repositorio OpenAI Whisper (2023)

La cantidad de parámetros que tiene el modelo representan las características de la señal de voz como lo son: el espectro de frecuencias, la envolvente espectral, etc. La cantidad de parámetros representa la complejidad del modelo, donde mientras mayor sea la cantidad de parámetros, mayor será la capacidad del modelo para representar las características de la señal de voz. Lo que a su vez también está directamente relacionado con el rendimiento del modelo en el reconocimiento de voz (OpenAI, 2023).

Un modelo de reconocimiento de voz responderá mejor cuando esté entrenado con un único idioma, por ejemplo, el inglés si se precisa de una única interacción con este. Existen otros modelos fueron entrenados con una variedad de lenguajes que si bien tienen una exactitud considerable no serán tan exactos como un modelo que fue entrenado unicamente con un lenguaje (OpenAI, 2023).

La cantidad de *VRAM* (Video Random Access Memory) requerida hace alusión a que esta librería usa recursos del equipo con el que se esté trabajando. Este es un factor que debe considerarse antes de usarla de esta. Debe evaluarse según los recursos de cada equipo el modelo ideal que debe utilizarse (OpenAI, 2023).

CAPÍTULO 7

Comunicación con inteligencia artificial por medio de un micrófono

7.1. Preparación para el desarrollo del programa

En este apartado se explica la forma en la que se desarrolló el código por medio del software *Visual Studio Code*. Lo primero que se hizo fue instalar la librería de *virtualenv* en un archivo de Visual Studio Code para crear un entorno virtual, lo cual se prefiere de esta forma para evitar conflictos con la instalación de las librerías de otros proyectos y además no dejar ningún residuo en el sistema al momento de eliminarlo. Para poder hacer uso del entorno virtual, este se activó y una vez activado se realizó la instalación de librerías. A continuación, se enlistan las librerías y una descripción del uso que se les dio en el programa:

- **Pyaudio:** permitió generar audio.
- **SpeechRecognition:** permitió reconocer al micrófono de nuestra computadora como fuente de audio.
- **Pyttsx3:** permitió el uso de las voces instaladas en el sistema operativo.
- **Pydub:** permitió editar el audio que se otorga del micrófono.
- **Whisper:** fue el motor para el reconocimiento de voz.

Es importante mencionar algunas observaciones durante este proceso. En el caso específico de entorno virtual, fue necesario verificar que la instalación se realizara correctamente, para ello, se verificó que en la dirección del archivo que se estaba trabajando existiera una carpeta con el nombre que se le dio al entorno virtual. Sin embargo, si se diera el caso de no tener dicha carpeta debe verificarse que la dirección en la que se esté buscando sea la correcta. Asimismo, también puede llegar a causar confusión la funcionalidad que se le dará

a la librería SpeechRecognition, ya que muchos casos es utilizada como el motor para el reconocimiento de voz, pero que este caso se consideró como una mejor opción otra librería para esta función.

En el caso de la instalación de la librería whisper fue necesaria la instalación de un gestor de dependencias *chocolately* y una dependencia llamada *ffmpeg*. La instalación del gestor de dependencias se debe a que al momento de la instalación de un componente de software este se encarga de verificar automáticamente las dependencias requeridas. Si alguna de estas aún no estaba instalada, es el propio gestor el que se encargará de descargarlas e instalarlas de forma automática antes de instalar el componente de software, que en este caso es la dependencia o también denominada colección de software.

Con respecto al caso de la dependencia "ffmpeg", se instaló utilizando al gestor de dependencias, esto con el fin de verificar que todas las librerías o componentes de software se instasen para su correcto funcionamiento. La instalación de la dependencia se debe a la conversión y manipulación de archivos multimedia, que en este caso son audios. Aunque también pueden mencionarse otras tareas que esta dependencia puede realizar:

Instalación de librerías:

- **Extracción de fragmentos:** recorta o extrae partes específicas de un archivo multimedia.
- **Conversión de fragmentos:** conversión de archivos de vídeo o audio de un formato a otro.
- **Captura de pantalla o grabación de pantalla:** captura pantallas de computadoras o graba videos de pantalla en tiempo real.
- **Cambio de códecs y ajustes:** permite cambiar los códecs de audio o video, ajustar la calidad, cambiar la velocidad de bits, entre otras modificaciones en la configuración de archivos multimedia.

Finalmente, a razón de verificar que todas las librerías estuvieran instaladas, se ejecutó una línea de código que muestra todas las librerías instaladas.

7.2. Desarrollo del programa

Lo primero que se hizo fue importar algunas librerías y módulos, por lo que a continuación se pondrán cada uno de estos y se explicará la utilidad que se le dará en el programa.

Módulos:

- **io:** permitió trabajar con archivos para guardarlos y manipularlos.
- **tempfile:** permitió generar archivos y directorios temporales.
- **os:** fue necesario para ciertos factores del sistema operativo.

Librerías:

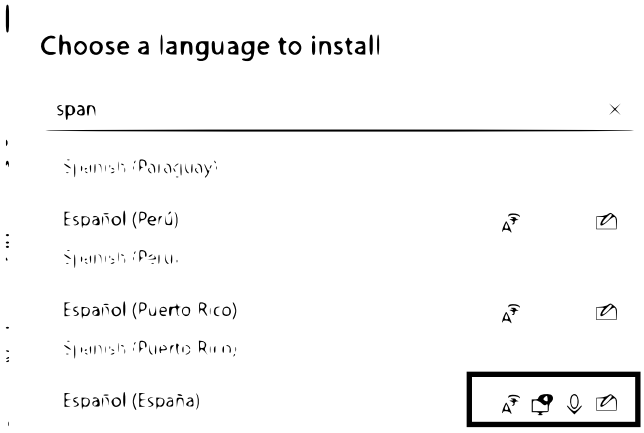
- **pydub (clase)**: tomó os datos generados por el micrófono (audio) para transformarlos en un archivo temporal.
- **SpeechRecognition**: fue la fuente de audio para capturar lo que se vaya a decir.
- **Whisper**: fue el motor el reconocimiento de voz.
- **pyttsx3**: permitió generar o producir la voz instalada en el sistema operativo Windows 11.

Como siguiente paso, se creó un directorio temporal a través de una variable que utiliza el módulo *tempfile* en el que se busca almacenar el archivo de audio que contendrá el audio del usuario que interactúe con el programa. Dicho directorio temporal debe generarse como un *string* con el fin de guardar el archivo de tipo audio en la ruta del directorio temporal. A partir de esto se creó una nueva variable, generando en ella un string usando el módulo .ºs", el módulo *path* y el módulo *join*. Teniendo este último la tarea de unir la variable del directorio temporal y el nombre que tendrá el archivo temporal de audio.

Se creó una nueva variable que a su vez crea un objeto de la librería SpeechRecognition el cual tiene la función de capturar audio del micrófono y realizar el reconocimiento de voz. Seguido de esto, se crea una nueva variable que se encarga de la inicialización de la síntesis de voz utilizando la librería de "pyttsx3". Después, se obtienen las voces disponibles creando, una nueva variable y con la variable que inicializó la síntesis de voz se configura el idioma español y la velocidad de la voz.

Algo que debe mencionarse es que para la instalación o desinstalación de voces en el sistema operativo Windows 11, basta con buscar la "configuración de lenguaje" en el buscador principal y ahí se visualizarán todas las voces que podemos instalar. Sin embargo, solamente aquellas voces que tengan el ícono de un monitor son las que tienen "Text-To-Speech", es decir, que tienen una voz generada. A continuación, podrá observar en la Figura 3:

Figura 3: **Configuración de lenguaje en Windows 11**



Nota: Adaptado de Windows (2023)

Llegados a este punto, se crea la primera función, la cual toma un argumento y el motor de síntesis de voz para generar la voz y hablar con el texto proporcionado. La segunda función se encarga de utilizar el objeto para capturar audio desde el micrófono, ajustar automáticamente el nivel de ruido ambiente, guardar el audio capturado desde el micrófono en un archivo tipo wav en una variable creada anteriormente y devolver la ruta del archivo de audio guardado de forma temporal.

La tercera función se encarga de utilizar la librería Whisper con el fin de cargar un modelo de reconocimiento de voz previamente definido. Es importante mencionar que OpenAI entrenó diferentes modelos siendo la potencia la diferencia entre ellos, los modelos son: *base*, *medium* y *large*. Su uso debe seleccionarse dependiendo de la potencia de nuestra computadora, debido a que el procesamiento lo hace nuestra propia máquina. También tiene a su cargo transcribir el audio contenido en el archivo tipo wav utilizando el modelo de reconocimiento de voz en el idioma español como fue definido y finalmente devolver el texto reconocido.

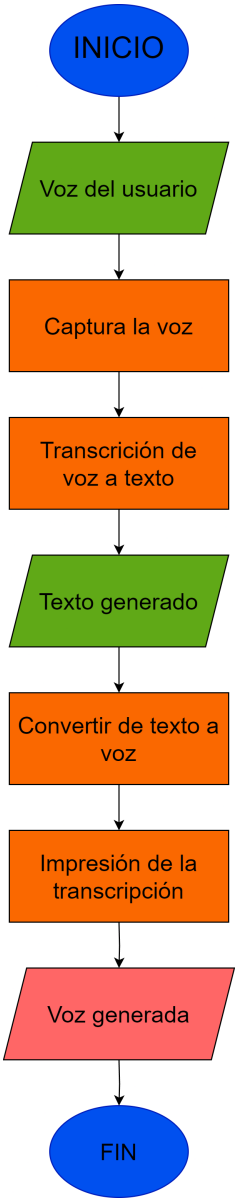
La interacción con el programa a modo de comprobar que se captó cada una de las palabras del usuario se logró tomando en cuenta funciones como lo fueron la segunda función encargada de captar el audio desde el micrófono y guardarlo en un archivo temporal. Consecutivamente, se llamó a una tercera función que tuvo la tarea de obtenerla transcripción del audio capturado a texto y por ultimo se llamó a la primera función para sintetizar la respuesta en forma de voz e imprimirla en la consola. La última parte del programa se encarga de ejecutar la función principal si el *script* se ejecuta como un programa independiente.

7.3. Resultados

En la sección de anexos se encuentran evidencias que respaldan todo lo descrito en este capítulo, en el anexo número 1 se encuentra un repositorio donde se podrá observar el código descrito, en el anexo número 2 se encuentran un videos que muestran cómo dicho programa realiza todo lo descrito y en el anexo número 6 se encuentra la documentación de todos los avances relacionados a los objetivos.

En la Figura 4 se puede observar el diagrama de flujo que explica la lógica que se siguió al desarrollar el programa. La explicación es la siguiente, se captura la voz utilizando un micrófono, se transcribe en texto utilizando un modelo de reconocimiento de voz y se reproduce la respuesta en forma de voz utilizando una voz sintetizada en español. Es decir, que se comprueba de manera correcta el habla del usuario por parte del programa desarrollado.

Figura 4: **Diagrama de flujo de comunicación con inteligencia artificial por medio de un micrófono**



Nota: Elaboración propia

CAPÍTULO 8

Respuesta por parte de la inteligencia artificial por medio de bocinas

En este apartado se explica la forma en la que se desarrolló el código por medio del software Visual Studio Code. Nuevamente, se hace uso de un entorno virtual. Una vez activado, se realizó la instalación de librerías. A continuación, se enlistarán las librerías y una descripción del uso que se les dio en el programa:

- **Pyaudio:** permite generar audio.
- **SpeechRecognition:** permite reconocer al micrófono de nuestra computadora como fuente de audio.
- **pyttsx3:** permite utilizar las voces instaladas en el sistema operativo.
- **pydub:** permite editar el audio que se otorga del micrófono.
- **Whisper:** es el motor para el reconocimiento de voz.

Como se podrá observar en esta parte, las librerías son las mismas que se instalaron para el programa anterior. Esto se debe a que no es necesario el uso de otra librería. A continuación, se dará una explicación del desarrollo de este programa y cuáles fueron las modificaciones necesarias para obtener una interacción por parte de la inteligencia artificial.

8.1. Desarrollo del programa

Lo primero que se hizo fue importar algunas librerías y módulos, por lo que a continuación se pondrán cada uno de estos y se explicará la utilidad que se le dio al programa.

Módulos:

- **sys**
- **os**

Librerías:

- **LangChain:** carga un documento de tipo texto, vectoriza para analizar y estructurar la información del documento tipo texto y para hacer uso de la *API* de ChatGPT.

Lo más relevante con respecto al desarrollo del programa consistió en que el texto transcrito por parte del modelo de Whisper fuera comunicado a Whisper. Para que este chatbot fuera capaz de generar una respuesta acorde a lo procesado por parte del modelo de la librería, el texto de esta respuesta era reproducido por parte de la voz descargada que se mencionó.

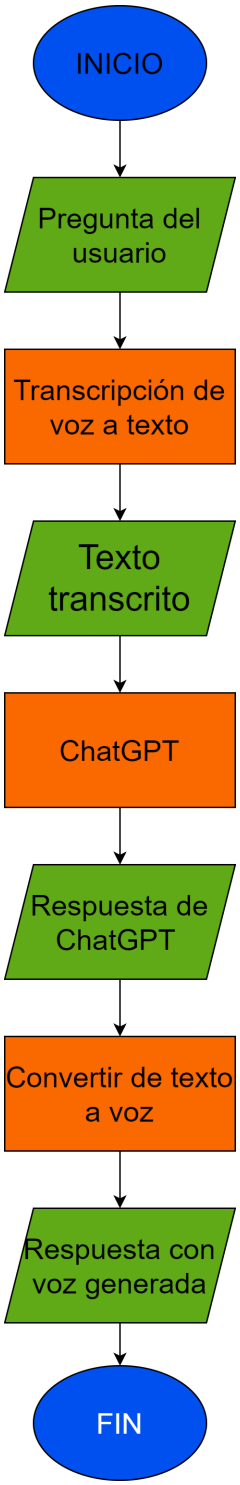
El cambio más importante se debe a la implementación temprana de la librería LangChain, que fue la encargada de obtener una respuesta por parte del Chatbot ChatGPT y que finalmente, por medio de una función con la tarea específica que antes reproducía el texto transcrito, ahora reproducía la respuesta generada por ChatGPT.

8.2. Resultados

En la sección de anexos se encuentran las evidencias que respaldan todo lo descrito en este capítulo, en el anexo número 1 se encuentra un repositorio donde se podrá observar el código descrito, en el anexo número 3 se encuentra un video que muestra cómo dicho programa realiza todo lo descrito anteriormente y en el anexo número 6 se encuentra la documentación entera de todo lo realizado para la realización de todos los avances relacionados a todos los objetivos.

En la Figura número 5 se puede observar el diagrama de flujo que explica la lógica que se siguió al desarrollar el programa. La explicación es la siguiente, se captura el habla utilizando un micrófono, se transcribe en texto utilizando un modelo de reconocimiento de voz y se reproduce la respuesta generada por ChatGPT en forma de voz utilizando una voz sintetizada en español. Es decir, que se comprueba de manera correcta que la inteligencia artificial, que en este caso es ChatGPT, genera una respuesta acorde a la interacción con el usuario y la reproduce por medio de unas bocinas.

Figura 5: **Diagrama de flujo de comunicación con inteligencia artificial por medio de bocinas**



Nota: Elaboración propia

CAPÍTULO 9

Entrenamiento de inteligencia artificial con datos relevantes de la Universidad del Valle de Guatemala

9.1. Preparación para el desarrollo del programa

En este apartado se explica la forma en la que se desarrolló el código por medio del software Visual Studio Code. Nuevamente se hace uso de un entorno virtual. Una vez activado, se realizó la instalación de librerías. A continuación, se enlistaran las librerías y una descripción del uso que se les dio en el programa:

Instalación de librerías:

- La primera librería será OpenAI, en ella se encuentra toda la información relacionada a la "API" creada por OpenAI para crear programas que permitan interactuar con esta inteligencia artificial (ChatGPT).
- La segunda librería será "LangChain" y esto se logrará con la siguiente línea de código: *pip install langchain*. Es importante mencionar que toda la documentación necesaria para hacer un uso correcto de la librería se extrajo de la página oficial creada por los mismo autores de LangChain. El link es el siguiente: `https://python.langchain.com/docs/get_started/quickstart`

Es importante tomar en cuenta que para el uso de la librería OpenAI debe crearse una cuenta en su sitio oficial y colocar un método de cobro, ya que el uso de esta API no es gratuito. Luego debe generarse una "llave" en el mismo sitio. Este es el método que los autores de OpenAI crearon para darles acceso a las personas a su "API". Cabe mencionar que dicha llave no debe compartirse con nadie.

9.2. Desarrollo del programa

Lo primero que se hizo fue importar algunas librerías y módulos, por lo que a continuación se pondrán cada uno de estos y se explicará la utilidad que se le dará en el programa.

Módulos:

- **sys**
- **os**

Librerías:

- **LangChain:** que en este caso tiene varias utilidades entre ellas cargar un documento de tipo texto, vectorizar para analizar y estructurar la información del documento tipo texto y para hacer uso de la API de ChatGPT.

Para concluir las importancias en el porgrama, también se importa un archivo que se creó aparte. En este existe una clave. En el archivo original se realizaron ciertas importaciones de LangChain. Dichas funciones fueron: *TextLoader*, *VectorIndexCreator*, *OpenAI* y *ChatOpenAI*.

Sucesivamente, se establece la clave de API de OpenAI utilizando los datos que se almacenaron el archivo que se mencionó anteriormente, esto es necesario para autenticarse en el servicio de OpenAI. Como siguiente paso, se crea una variable en la que se obtiene la pregunta que el usuario desea hacer. Para ello, en el momento en el que se ejecute el programa, el usuario debe proporcionar la pregunta como el primer argumento.

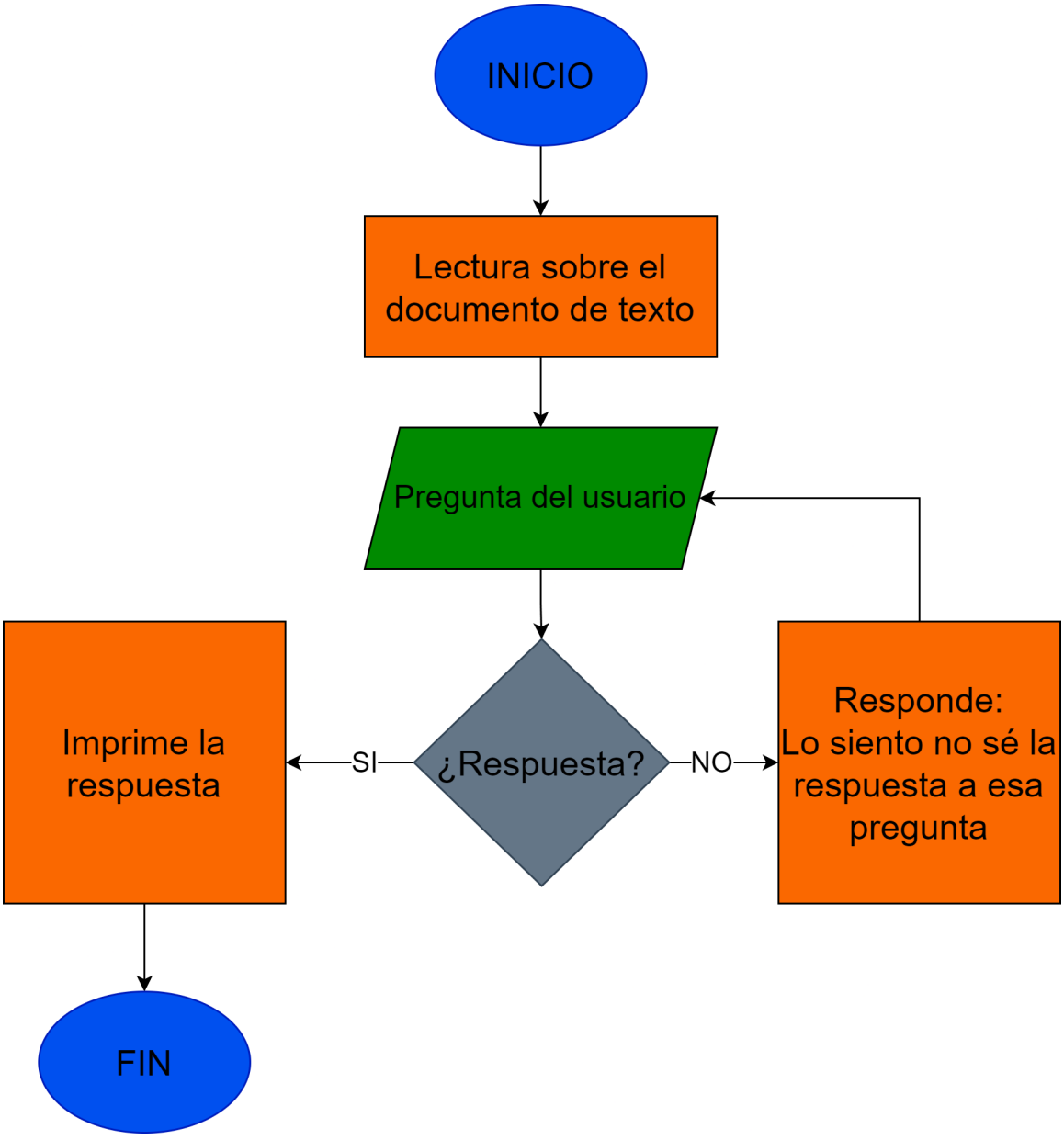
Después, se crea una variable nueva que tiene la función principal de cargar un documento de texto en el que se colocó la información relevante de la Universidad Del Valle de Guatemala. La última variable que se define crea un índice de vectorización tomando el documento de texto previamente cargado. Básicamente, es un proceso que convierte el texto en una representación numérica que el modelo de lenguaje pueda comprender. Finalmente, consulta la pregunta escrita por el usuario al modelo de lenguaje GPT-3 usando información proporcionada por el archivo de tipo texto, para entonces imprimirla.

9.3. Resultados

En la sección de anexos se encuentran evidencias que respaldan de todo lo descrito en este capítulo, en el anexo número 1 se encuentra un repositorio donde se podrá observar el código descrito, en el anexo número 4 se encuentra un video que muestra cómo dicho programa realiza todo lo descrito anteriormente y en el anexo número 6 se encuentra la documentación entera de todo lo realizado para la realización de todos los avances relacionados a todos los objetivos.

En la Figura número 6 podemos observar el diagrama de flujo que explica la lógica que se siguió al desarrollar el programa. La explicación es la siguiente: este programa utiliza el modelo de lenguaje GPT-3 de OpenAI para poder responder a las preguntas que el usuario haga usando también como herramienta el documento de texto proporcionado, esto como un método de reentreno a la inteligencia artificial.

Figura 6: **Diagrama de flujo de comunicación con inteligencia artificial por medio de un micrófono**



Nota: Elaboración propia

CAPÍTULO 10

Implementación de una interfaz para el reentrenamiento de la inteligencia artificial

10.1. Preparación para el desarrollo del programa

En este apartado se explica la forma en la que se desarrolló el código por medio del software Visual Studio Code. Nuevamente se hace uso de un entorno virtual. Una vez activado, se realizó la instalación de la librería *Tkinter*, esta herramienta permitió crear una interfaz gráfica.

10.2. Desarrollo del programa

Lo primero que se realizó en el programa fue la definición de una función la cual se encargó de tomar el texto ingresado por el usuario en el espacio correspondiente para escribirlo en un archivo de tipo texto y luego cerrar la ventana principal de la aplicación, todo esto después de presionar un botón disponible con la etiqueta "terminar". En otras palabras, esta función fue indispensable para definir el funcionamiento de la interfaz gráfica, sin embargo, para ello fue necesario crear de una serie de variables, las cuales se explican a continuación.

La primera variable se encargó de crear la ventana principal. En dicha ventana se almacena una variable a la cual se le asignará el titulo de Reentrenar IA. A continuación, se creo una etiqueta que muestra un texto (Escribe la información para reentrenar a la IA), dicha etiqueta se empaqueta en la ventana.

Después se creo una variable encargada de crear un cuadro de entrada de texto con 40 caracteres. El cuadro de entrada permite al usuario escribir texto y este se empaqueta en la ventana.

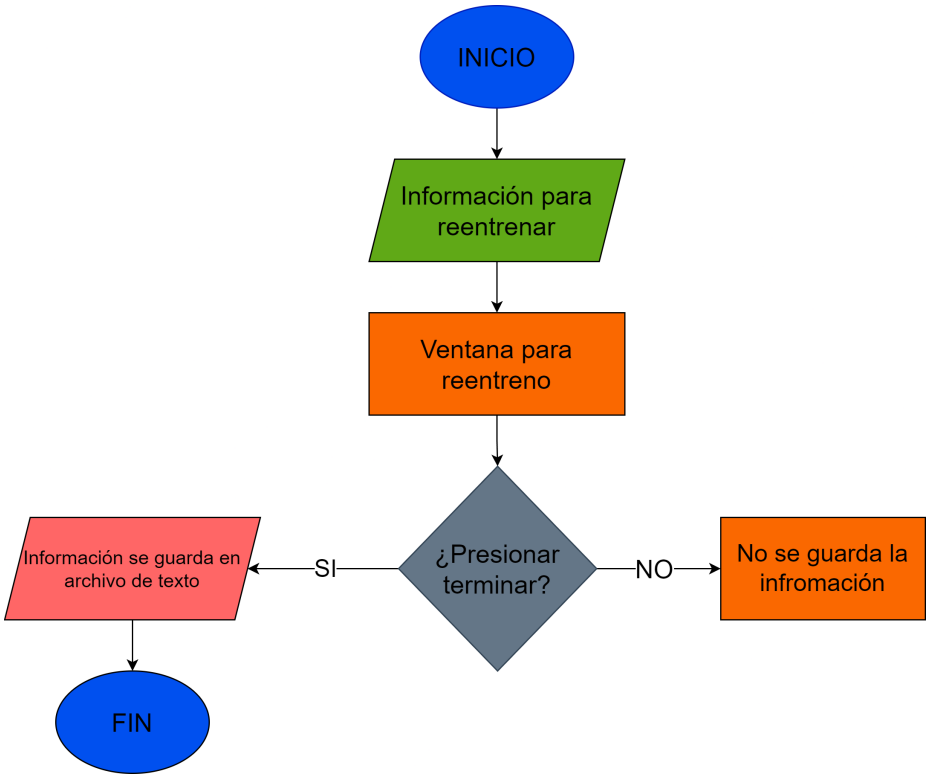
La última variable definida fue la encargada de crear un botón con el texto "terminar"que además esta asociado a la función que se encargó de guardar el texto de entrada en un documento al momento de presionar dicho botón. Se inició el bucle principal de la interfaz gráfica que espera eventos del usuario, como lo es un texto en el espacio de entrada y presionar el botón de "terminar"para poder guardar esta información en un archivo de texto.

10.3. Resultados

En la sección de anexos se encuentra evidencias que respaldan todo lo descrito en este capítulo, en el anexo número 1 se encuentra un repositorio donde se podrá observar el código descrito, en el anexo número 5 se encuentra un video que muestra cómo dicho programa realiza todo lo descrito anteriormente y en el anexo número 6 se encuentra la documentación entera de todo lo realizado para la realización de todos los avances relacionados a todos los objetivos.

En la Figura número 7 podemos observar el diagrama de flujo que explica la lógica que se siguió al desarrollar el programa. La explicación es la siguiente: se creó una interfaz gráfica capaz de brindarle un espacio de entrada al usuario en el que puede colocar la información que desee para entonces poder guardar dicha información en un archivo de texto del cual podrá tener disposición la inteligencia artificial.

Figura 7: **Diagrama de flujo de comunicación con inteligencia artificial por medio de un micrófono**



Nota: Elaboración propia

CAPÍTULO 11

Unificación del código

Primero debe mencionarse el orden que se siguió para poder conectar todos los programas que anteriormente se habían hecho. Se colocaron todas las librerías en un documento tipo texto que en este caso está identificado con el nombre *requirements.txt*. Este documento es importante ya que para instalar todas las librerías que se encuentran dentro del archivo de tipo texto era necesario usar una línea de código que se encontrará en el material de apoyo en el anexo número 6.

Ahora bien, el orden del programa consiste en la identificación de la palabra clave, tomando en cuenta que existen únicamente dos palabras clave (gato y entreno). Las palabras clave tienen funciones asignadas, explicadas a continuación. Es importante mencionar que para la identificación de palabras clave es necesaria la transcripción del habla a texto; para esta tarea existe una función dedicada.

11.1. Palabra clave Gato

Una vez se realice la transcripción de la palabra clave, esta se buscará en un programa que tiene asignada la tarea de actuar como un asistente. Al momento de activarse el asistente, se iniciará una transcripción de texto a voz de una frase que ya está determinada dentro del programa, es decir, que el asistente hablará. Para la transcripción de texto a voz también existe una función dedicada. Sucesivamente, se despliega un texto que indica al usuario que puede empezar a hablar y realizar la pregunta que desea.

Nuevamente, se realiza una transcripción de voz a texto que será enviada a la inteligencia artificial, la cual genera una respuesta en texto, que posteriormente transcribe la voz para poder comunicarle al usuario la respuesta deseada. El asistente personal también es capaz de responder con la información que se encuentra en un archivo de tipo texto, ya que el

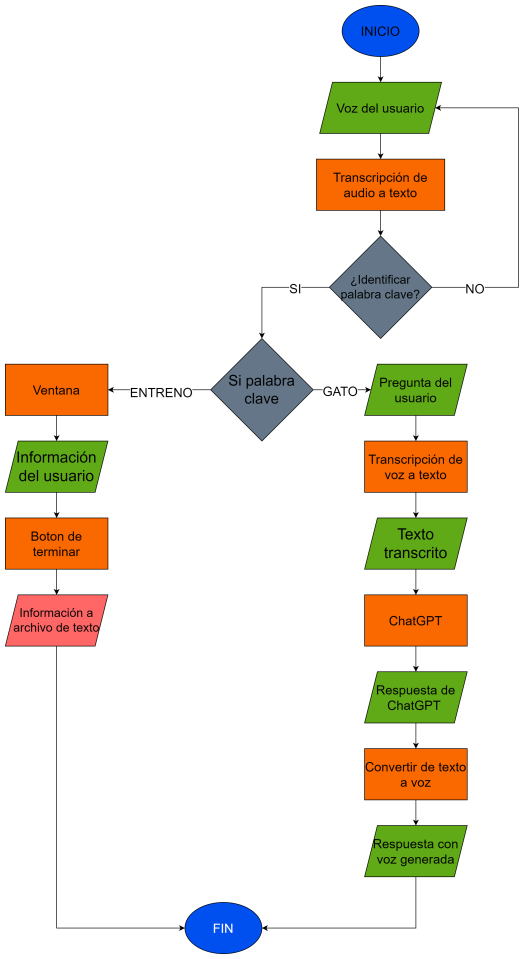
programa con información relevante de la Universidad Del Valle de Guatemala. Es importante mencionar que cuando el asistente personal termina de dar la respuesta, el programa imprime la respuesta completa en forma de texto.

11.2. Palabra clave Entreno

Una vez se realice la transcripción de la palabra clave, esta se buscará en un programa donde la tarea asignada será abrir una ventana en la que habrá un cuadro de texto donde se colocará toda la información a la que se desea que el asistente tenga acceso. Dicha información se almacenará en un archivo de tipo texto. Cada vez que se haga uso de esta palabra clave, la información se irá almacenando.

A continuación, en la Figura 8 se podrá observar una diagrama de flujo que explica la lógica que sigue el programa unificado:

Figura 8: **Diagrama de flujo del programa unificado**



Nota: Elaboración propia

11.3. Resultados

Para determinar el correcto funcionamiento del programa, se realizaron una serie de pruebas que identifican los factores que podrían afectar su funcionamiento. Para ello, se consideraron dos escenarios tomando como factor principal el ruido. El escenario uno consiste en un lugar que sea considerablemente silencioso y para el segundo escenario se consideró lo contrario, un lugar que contara con una considerable cantidad de ruido. Fue necesario el uso de una aplicación que pudiera medir el volumen de los diferentes factores que lo generan. La aplicación llamada *Decibel X* es para teléfonos inteligentes, disponible tanto para *Android* como para para *iOS*.

Se tomó en cuenta el uso de todos los modelos disponibles por la librería de Whisper *(tiny, base, small, medium y large)*, con el fin de identificar cuál sería el modelo ideal. Las pruebas se realizaron utilizando el micrófono de la computadora y el micrófono de unos auriculares con cada uno de los modelos así como en ambos escenarios, con el fin de poder encontrar las condiciones ideales para el funcionamiento correcto del programa. Por último se hizo mención de la velocidad de internet, ya que este factor afecta directamente la velocidad de respuesta por parte de los servicios de ChatGPT. Por lo tanto, esto muestra que para el uso de este programa se debe tener una buena conexión a Internet.

11.4. Escenario 1

Este escenario tiene la característica de ser considerablemente silencioso. En esta parte de la investigación se explicaran todos los resultados obtenidos gracias a la implementación del programa con los diferentes modelos así como de ciertas explicaciones de algunas gráficas obtenidas gracias a la aplicación mencionada anteriormente. Ahora bien, cada escenario se dividirá en la implementación del micrófono así como la conexión a *Internet* que se utilizó.

11.4.1. Receptor de sonido computadora para escenario 1

En el caso de este escenario, las condiciones de las pruebas fueron un lugar considerablemente silencioso, conexión a Internet residencial y el uso de un micrófono de computadora, además el orden establecido tomó como primer modelo a prueba al tiny y como último large. Para cada prueba fue necesario grabar un video en el cual se pudieran visualizar las pruebas que se realizaron con cada uno de los modelos, los vídeos de prueba se encuentran en el anexo número 10. A continuación en el Cuadro 1 podrá observar una tabla con los diferentes tiempos de respuestas entre cada uno de los modelo, siendo el mejor el obtenido por el modelo tiny y el peor obtenido por el modelo large.

Cuadro 1: **Tiempo de respuesta en segundos para el escenario 1 Micrófono Computadora**

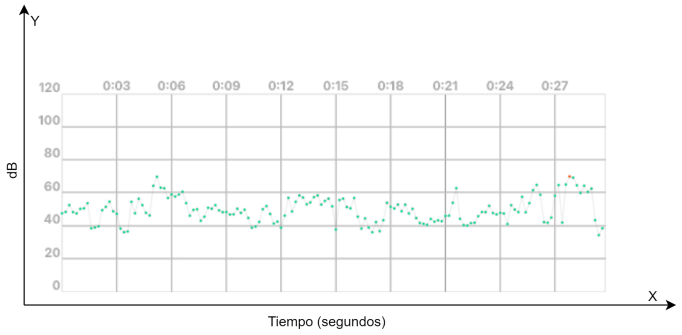
Micrófono computadora					
Modelo	**Tiny**	**Base**	**Small**	**Medium**	**Large**
P1	20.62	20.90	24.79	37.03	54.70
P2	21.22	24.11	25.06	35.94	62.67
P3	17.92	21.32	25.52	36.89	63.58
P4	19.38	20.74	25.16	37.80	57.59
P5	21.33	22.36	24.49	37.06	54.20
Promedio	20.094	21.886	25.004	36.944	58.548

Nota: Elaboración propia

Durante las pruebas se realizó una misma pregunta: ¿Podrías explicarme cuáles son las tres Leyes de Newton, por favor?. Con el fin de verificar que la respuesta fuera similar o igual. El objetivo era comprobar que la inteligencia artificial estuviera respondiendo correctamente con cada uno de los modelos empleados con las condiciones del escenario número 1. Además, se verifico el tiempo de respuesta con cada modelos, ya que este era un factor indispensable para determinar al mejor modelo, que en este caso el que mejor resultados mostró fue el modelo tiny.

Se podrá observar a continuación en la Figura número 9 una gráfica que muestra las variaciones de decibelios durante las pruebas con las condiciones mencionadas. Se comprobó que el mejor escenario para recibir las respuestas es un lugar silencioso.

Figura 9: **Gráfica de decibelios escenario 1 micrófono de computadora**



Nota: Elaboración propia

11.4.2. Receptor de sonido audífonos para escenario 1

En el caso de este escenario, las condiciones de las pruebas fueron un lugar considerablemente silencioso, conexión a Internet residencial y el uso de un micrófono de unos auriculares, además el orden establecido tomó como primer modelo a prueba al tiny y como último large. Para cada prueba fue necesario grabar un video en el cual se pudieran visualizar las pruebas que se realizaron con cada uno de los modelos, los vídeos de prueba se encuentran en el anexo número 11. A continuación en el Cuadro 2 podrá observar una tabla con los diferentes

tiempos de respuestas entre cada uno de los modelos,. siendo el mejor el obtenido por el modelo tiny y el peor obtenido por el modelo large.

Cuadro 2: **Tiempo de respuesta en segundo para el escenario 1 Micrófono de Auriculares**

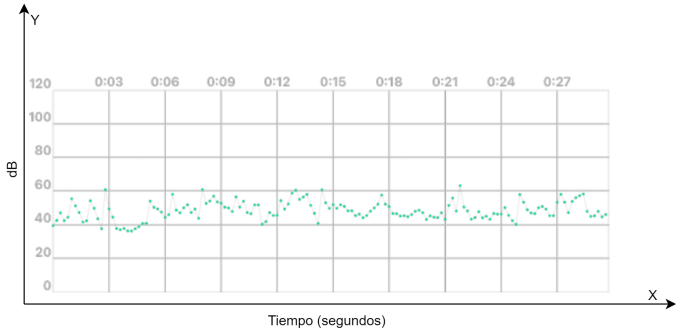
Micrófono Auriculares					
Modelo	**Tiny**	**Base**	**Small**	**Medium**	**Large**
P1	19.47	14.92	23.79	38.19	58.92
P2	19.79	20.78	24.29	36.78	60.00
P3	20.55	21.27	26.29	37.63	61.43
P4	18.81	20.94	24.78	38.38	59.66
P5	20.18	21.80	25.45	38.57	59.43
Promedio	19.760	19.942	24.920	37.910	59.888

Nota: Elaboración propia

Durante las pruebas se realizó una misma pregunta: ¿Podrías explicarme cuáles son las tres Leyes de Newton, por favor?. Con el fin de verificar que la respuesta fuera similar o igual. El objetivo era comprobar que la inteligencia artificial estuviera respondiendo correctamente con cada uno de los modelos empleados con las condiciones del escenario número 1. Además, se verifico el tiempo de respuesta con cada modelos, ya que este era un factor indispensable para determinar al mejor modelo, que en este caso el que mejor resultados mostró fue el modelo tiny.

Se podrá observar a continuación en la Figura número 10 una gráfica que muestra las variaciones de decibelios durante las pruebas con las condiciones mencionadas. Se comprobó que el mejor escenario para recibir las respuestas es un lugar silencioso.

Figura 10: **Gráfica de decibelios escenario 1 micrófono de auriculares**



Nota: Elaboración propia

Por último pero no menos importante se mostrará una imagen de una prueba de la velocidad de Internet residencial, que como puede observarse en la Figura 11 fue una prueba realizada desde el sitio web de Ookla.

Figura 11: **Prueba de internet conexión residencial**

Nota: Adaptado de SpeedTest (2023)

11.5. Escenario 2

En el escenario número 2, la condición a la que estuvieron expuestas las siguientes pruebas consisten en un lugar considerablemente ruidoso. En esta parte de la investigación se explicarán todos los resultados obtenidos gracias a la implementación del programa con los diferentes modelos así como de ciertas explicaciones de algunas gráficas obtenidas gracias a la aplicación mencionada anteriormente. Ahora bien, cada escenario se dividirá en la implementación del micrófono así como la conexión a Internet que se utilizó.

11.5.1. Micrófono computadora para escenario 2

En el caso de este escenario, las condiciones de las pruebas fueron un lugar considerablemente ruidoso, conexión a Internet compartida desde un teléfono inteligente y el uso de un micrófono de computadora, además el orden establecido tomó como primer modelo a prueba al tiny y como último large. Para cada prueba fue necesario grabar un video en el cual se pudieran visualizar las pruebas que se realizaron con cada uno de los modelos, los vídeos de prueba se encuentran en el anexo número 12. A continuación en el Cuadro 3 podrá observar una tabla con los diferentes tiempos de respuestas entre cada uno de los modelos, siendo el mejor el obtenido por el modelo tiny y el peor obtenido por el modelo large.

Cuadro 3: **Tiempo de respuesta en segundos para el escenario 2 Micrófono Computadora**

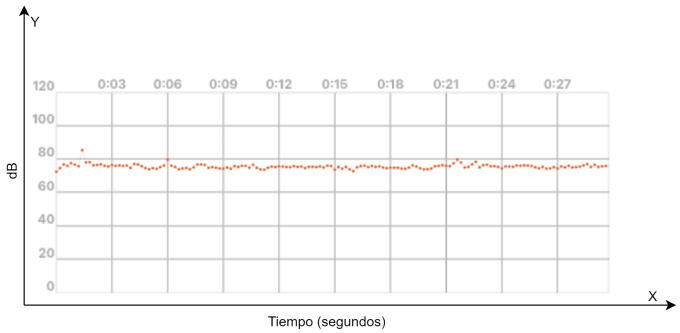
Micrófono Computadora					
Modelo	**Tiny**	**Base**	**Small**	**Medium**	**Large**
P1	19.48	19.24	25.18	34.75	53.04
P2	20.05	18.24	23.37	36.27	56.01
P3	18.78	18.64	22.34	38.76	53.82
P4	18.01	19.14	22.47	32.49	56.27
P5	18.42	18.97	22.71	34.47	57.04
Promedio	18.948	18.846	23.214	35.348	55.236

Nota: Elaboración propia

Durante las pruebas se realizó la mismo pregunta, : ¿Podrías explicarme cuáles son las tres Leyes de Newton, por favor?. Con el fin de verificar que la respuesta fuera similar o igual. El objetivo era comprobar que la inteligencia artificial estuviera respondiendo correctamente con cada uno de los modelos empleados con las condiciones del escenario número 2. Además, se verifico el tiempo de respuesta con cada modelos, ya que este era un factor indispensable para determinar al mejor modelo, que en este caso el que mejor resultados mostró fue el modelo tiny.

Asimismo, se podrá observar a continuación una gráfica en la Figura número 12 que muestras las variaciones de decibelios durante las pruebas con las condiciones mencionadas anteriormente comprobando así el escenario siendo un lugar considerablemente silencioso.

Figura 12: **Gráfica de decibelios escenario 2 micrófono de computadora**



Nota: Elaboración propia

11.5.2. Micrófono auriculares para escenario 2

En el caso de este escenario, las condiciones de las pruebas fueron un lugar considerablemente ruidoso, conexión a Internet compartida desde un teléfono inteligente y el uso de un micrófono de unos auriculares, además el orden establecido tomó como primer modelo a prueba al tiny y como último large. Para cada prueba fue necesario grabar un video en el cual se pudieran visualizar las pruebas que se realizaron con cada uno de los modelos, los vídeos de prueba se encuentran en el anexo número 13. A continuación en el Cuadro 4 podrá observar una tabla con los diferentes tiempos de respuestas entre cada uno de los modelos, siendo el mejor el obtenido por el modelo tiny y el peor obtenido por el modelo large.

Cuadro 4: **Tiempo de respuesta en segundos para el escenario 2 Micrófono de Auriculares**

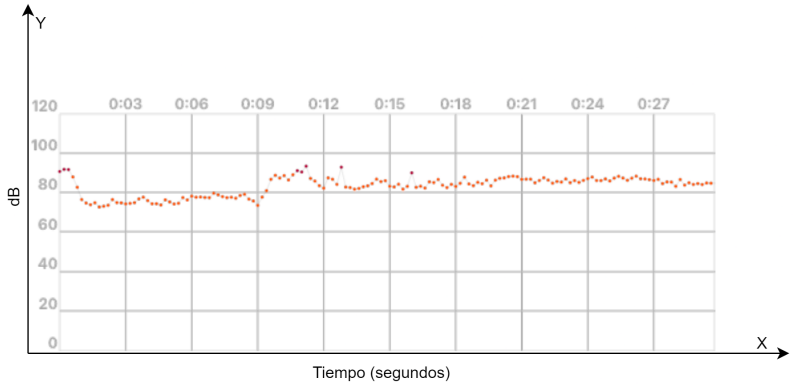
Micrófono Auriculares					
Modelo	**Tiny**	**Base**	**Small**	**Medium**	**Large**
P1	16.60	16.46	26.18	35.56	58.23
P2	15.78	17.43	24.47	36.30	57.95
P3	16.97	17.29	23.75	37.95	58.12
P4	14.25	18.44	23.56	37.86	57.63
P5	15.73	17.44	22.89	36.42	57.04
Promedio	15.866	17.412	24.170	36.818	57.794

Nota: Elaboración propia

Durante las pruebas se realizó la mismo pregunta, : ¿Podrías explicarme cuáles son las tres Leyes de Newton, por favor?. Con el fin de verificar que la respuesta fuera similar o igual. El objetivo era comprobar que la inteligencia artificial estuviera respondiendo correctamente con cada uno de los modelos empleados con las condiciones del escenario número 2. Además, se verifico el tiempo de respuesta con cada modelos, ya que este era un factor indispensable para determinar al mejor modelo, que en este caso el que mejor resultados mostró fue el modelo tiny.

Asimismo, se podrá observar a continuación una gráfica en la Figura número 13 que muestras las variaciones de decibelios durante las pruebas con las condiciones mencionadas anteriormente comprobando así el escenario siendo un lugar considerablemente silencioso.

Figura 13: **Gráfica de decibelios escenario 2 micrófono de auriculares**



Nota: Elaboración propia

Por último pero no menos importante se mostrará una imagen de una prueba de la velocidad de conexión de Internet compartida desde un teléfono inteligente, como puede observarse en la Figura 14 una prueba realizada desde el sitio web de Ookla.

Como se mencionó anteriormente para estas pruebas se consideraron dos escenarios, el escenario número uno consistió en un lugar considerablemente silencioso y una conexión de Internet residencial que puede observarse en la Figura número 11 con una velocidad descarga de 30.37 Mbps, mientras que el escenario número dos consistió en un lugar consi-

Figura 14: **Prueba de internet conexión compartida por teléfono inteligente**

Nota: Adaptado de SpeedTest (2023)

derablemente ruidoso y con una conexión a Internet compartida por un teléfono inteligente que puede observarse en la Figura número 14 con una velocidad de descarga de 76.73 Mbps. Para ambos escenarios se utilizaron dos micrófonos, el micrófono de una computadora y un micrófono de un par de audífonos. Los mejores resultados independientemente del escenario se obtuvieron utilizando el micrófono de par de audífonos, dato que fue posible corroborar por los promedios obtenidos en los cuadros uno, dos, tres y cuatro.

La diferencia de velocidad de descarga del Internet residencial y el Internet compartido por un teléfono inteligente significa que la velocidad en que se recibió la respuesta fue más veloz para el escenario dos. Como puede observarse al comparar los resultados promedio entre los cuadros del escenario número uno y los cuadros del escenario número dos, donde se obtuvo una diferencia de aproximadamente dos segundos a favor de los cuadros del escenario número dos. El modelo que obtuvo los resultados con menor tiempo de respuesta fue el modelo tiny, convirtiéndose en el modelo más rápido.

Finalmente se concluye que los mejores resultados en cuanto a tiempo de respuesta, se obtuvieron tomando en consideración un lugar notablemente silencioso, una conexión a Internet compartida por un teléfono inteligente, el uso de un micrófono de un par de audífonos y el uso del modelo tiny. Un lugar considerablemente silencioso y el uso de un micrófono de un par de audífonos tuvieron ventaja porque le brindaron menos problemas que intervinieran en la recepción de la pregunta que efectuara el usuario. Mientras que la velocidad de descarga de la conexión a Internet compartida por un teléfono inteligente al ser mayor que la del Internet residencial, permiten que las respuestas se recibían con mayor velocidad, ya que los servidores de ChatGPT son los encargados de generar la respuesta y enviarla, sin embargo el programa desarrollado solo se encarga de recibirla a través de la velocidad de descarga de la conexión de Internet. El modelo tiny consumió la menor cantidad de recurso de la computadora, resultados que son posibles de corroborar en los anexos nueve, diez, once y doce. Por lo tanto con las condiciones anteriormente mencionadas del escenario número dos, fue posible obtener los mejores promedios de tiempo de respuesta del programa.

CAPÍTULO 12

Conclusiones

- Se logró convertir la información recibida a través de un micrófono en un archivo tipo wav que posteriormente se entregó a una inteligencia artificial de texto para su procesamiento.
- Se implementó un método de comunicación de la inteligencia artificial por medio de bocinas como medio de transmisión.
- Se logró que la inteligencia artificial implementada respondiera de manera congruente con respecto a las preguntas realizadas por el usuario.
- Se logró la implementación de un método de reentrenamiento para la inteligencia artificial con información relevante de la Universidad Del Valle de Guatemala.
- Se desarrolló una interfaz amigable para el usuario capaz de reentrenar a la inteligencia artificial con la información que este desee.
- La conexión a Internet afecta la velocidad de respuesta del programa en cada uno de los modelos que tiene a su disposición la librería de Whisper.
- El uso de un micrófono que esté más cerca del usuario afecta la velocidad de respuesta del programa en cada uno de los modelos que tiene a su disposición la librería de Whisper.
- El uso de la librería Whisper consume recursos del equipo en el que se esté ejecutado un programa que haga uso de esta.

CAPÍTULO 13

Recomendaciones

- La implementación de un método que pueda aumentar la velocidad de transcripción por parte de los modelos de la librería de Whisper. Para ello es posible usar *Fast-Whisper*, que usa un motor de interferencias rápido para modelos Transformer. Con esto, también debería ser posible aumentar la velocidad en la que Whisper obtiene una respuesta de la solicitud por parte del usuario.

- Podría ser posible la implementación de otras voces para el uso de esta inteligencia artificial. Es decir, el uso de la voz en esta investigación podría ser mejorada con el fin de proporcionar una experiencia más humana para el usuario haciendo uso de software que se dedican a la síntesis de voz humanas realistas. Además también es posible realizar un entrenamiento con nuestra propia voz brindando de esta manera aún más realismo a la voz que podría usar la inteligencia artificial.Podría realizarse una investigación que tenga como objetivo mejorar la síntesis de la voz, humanizando cada vez más la interacción con una inteligencia artificial.

- Implementar otro método para el reentrenamiento de la inteligencia artificial, se recomienda la consideración de diferentes métodos que puedan ser más cómodos para cualquier tipo de usuario. Uno de ellos podría ser que al momento de realizar el método de reentrenamiento, la inteligencia artificial pueda indicar los diferentes métodos que se pueden utilizar. Entre ellos podría existir uno que ofrezca ingresar la información del reentrenamiento por medio del habla. Pueden existir diferentes detalles que hagan de esta experiencia algo práctico. Por ejemplo, que pueda ir transcribiendo en tiempo real todo lo que el usuario está diciendo, de esta manera el usuario se asegurará de que toda la información se está ingresando de la manera correcta.

  Debido a que el ruido del ambiente interfiere en la calidad de los resultados, sería una buena práctica que antes de que la inteligencia artificial empiece a transcribir toda la información, le indique al usuario que debe hablar con claridad y a una velocidad moderada.

- Debido a la creciente importancia de la inteligencia artificial en los recientes años, se considera importante realizar una evaluación sobre las diferentes inteligencias artificia-

les existentes. Por tanto, debe realizarse una implementación de inteligencias artificiales como Bing AI o Bard AI al programa ya desarrollado, para entonces realizar una serie de evaluaciones para las que deben considerarse la velocidad y la exactitud en las respuestas generadas. Para la implementación de estas, debe tomarse en cuenta que debe realizarse una serie de modificaciones al programa, lo que mayormente se reduce al archivo llamado brain.py donde se realizó la conexión respectiva con el archivo tipo texto con el que se reentrenó a la inteligencia artificial y la información con la que ya contaba.

- El reentreno es otro aspecto que debe tomarse en cuenta al momento de considerar las inteligencias artificiales, ya que probablemente existan diferencias significativas al momento de aplicar este proceso. Una vez estas consideraciones sean implementadas se podrá determinar si ChatGPT es la mejor inteligencia artificial para la plataforma del rostro animatrónico existente, o bien si procede un cambio en esta. Si este fuera el caso, entonces debe especificarse cuál y cuáles son las razones. Asimismo, también debe especificarse si se decidieron realizar otras pruebas, tomando en cuenta que estas pruebas deben ejecutarse no solo con las inteligencias artificiales propuestas sino también con ChatGPT.

CAPÍTULO 14

Bibilografia

E. Alpaydin. Introduction to machine learning [Introducción al aprendeizaje automático]. MIT Press, 2020.

E. Balajthy. "Text-to-speech software for helping struggling readers" [Software de texto a voz para ayudar a los lectores con dificultades], Reading Online, vol. 8, n.o 4, págs. 1-9, 2005.

Code.org. "How AI Works" [Como funciona la IA]. Code. Dirección: https://code.org/curriculum/how-ai-works

D. Fuentes. "Implementación de un chatbot a través de reconocimiento de voz en tiempo real entre el usuario y el rostro animatrónico de la Universidad del Valle de Guatemala," Tesis de Licenciatura, Universidad del Valle de Guatemala, 2021.

K. Gonzáles. "Diseño e Implementación de un Sistema para Reconocimiento Facial, de Gestos y de Voz para un Rostro Animatrónico," Tesis de Licenciatura, Universidad del Valle de Guatemala, 2020.

H. Robotics. "Sophia, Página oficial," 2023. Dirección: https://www.hansonrobotics.com/sophia/.

J. Journey. "A ChatGPT Voice Assistant You Can Talk To - Open Source: Vivy" [Un asistente de voz ChatGPT con el que puedes hablar - Código abierto: Vivy], Es un proyecto que está en desarrollo por lo que puede presentar una serie de mejoras significativas en el futuro.

J. McCarthy. What is Artificial Intelligence? [¿Qué es la Inteligencia Artificial?] Stanford University, 2007.

Thorsten-Voice. "Create your own Text to Speech voice clone | FREE | LOCAL" [Crea tu propio clon de voz de Texto a Voz | GRATIS | LOCAL], Este proyecto considera el hecho de realizar un Text To Speech tomando en cuenta al idioma alemán, ya que para este caso en específico esto puede representar una mayor dificultad al no ser un idioma precisamente con gran demanda.

A. Trivedi, N. Pant, P. Shah, S. Sonik, y S. Agrawal. "Speech to text and text to speech recognition systems-A review" [Sistemas de reconocimiento de voz a texto y texto a voz: una revisión], IOSR J. Comput. Eng, vol. 20, n.o 2, págs. 36-43, 2018.

CAPÍTULO 15

Anexos

En el Anexo 1, se encuentra el repositorio que contiene todo el código del programa así como las diferentes modificaciones que se realizaron en el transcurso de su desarrollo.

15.1. Anexo 1: Link del repositorio

https://github.com/kekellner/2023-rostro-animatronico.git

En el Anexo 2, se encuentra una grabación en la que se verifica que el programa cumplió con lo descrito en el objetivo específico número uno.

15.2. Anexo 2: Link de prueba Objetivo específico 1

https://youtu.be/EI-Wyg_v_Sc

En el Anexo 3, se encuentra una grabación en la que se verifica que el programa cumplió con lo descrito en el objetivo específico número dos.

15.3. Anexo 3. Link de prueba Objetivo específico 2

https://www.youtube.com/watch?v=OooHrz2yR-A

En el Anexo 4, se encuentra una grabación en la que se verifica que el programa cumplió con lo descrito en el objetivo específico número tres.

15.4. Anexo 4: Link de prueba Objetivo específico 3

https://youtu.be/uBMisdLkYCA

En el Anexo 5, se encuentra una grabación en la que se verifica que el programa cumplió con lo descrito en el objetivo específico número cuatro.

15.5. Anexo 5: Link de prueba Objetivo específico 4

https://youtu.be/wlWu9PqvA_w

En el Anexo 6, se encuentra un documento en formato de texto en el que se describe todo el proceso que se realizó para el desarrollo del programa.

15.6. Anexo 6: Link de documentación completa

https://docs.google.com/document/d/1hANYfE2e4hnQfqhsye8Zx99mSYYoLaBR8jSXlGSxbKk/edit?pli=1

En el Anexo 7, se encuentra la página oficial de Ookla para realizar pruebas en las que se determinó la velocidad de la conexión a Internet.

15.7. Anexo 7: Link de prueba de internet

https://www.speedtest.net/es

En el Anexo 8, se encuentra un enlace de Drive proporcionado por la universidad en el que se encuentra todo tipo de información sobre la presente tesis.

15.8. Anexo 8: Link de documento en drive

https://drive.google.com/drive/u/0/folders/1DZYYc0UGwFQhWvwO6gWP6wjxUHYNT6_-8

En el Anexo 9, se encuentra una lista de reproducción de todas las grabaciones que se realizaron con las condiciones del escenario número uno, con todos los modelos, conexión a Internet residencial y un micrófono de computadora.

15.9. Anexo 9: Escenario 1 Pruebas Micrófono Computadora

https://www.youtube.com/playlist?list=PLqIbJR6nYKfg14yPbOyMG_b2o8rva-uZw

En el Anexo 10, se encuentra una lista de reproducción de todas las grabaciones que se realizaron con las condiciones del escenario número uno, con todos los modelos, conexión a Internet residencial y un micrófono de uno auriculares.

15.10. Anexo 10: Escenario 1 Pruebas Micrófono Auriculares

https://www.youtube.com/playlist?list=PLqIbJR6nYKfjJjZiTrs_ycI_VjA2n0uev

En el Anexo 11, se encuentra una lista de reproducción de todas las grabaciones que se realizaron con las condiciones del escenario número dos, con todos los modelos, conexión a Internet compartida por un teléfono inteligente y un micrófono de computadora.

15.11. Anexo 11: Escenario 2 Pruebas Micrófono Computadora

https://www.youtube.com/playlist?list=PLqIbJR6nYKfi-RGPsmpjPHGHNv1XOzmCR

En el Anexo 12, se encuentra una lista de reproducción de todas las grabaciones que se realizaron con las condiciones del escenario número dos, con todos los modelos, conexión a Internet compartida por un teléfono inteligente y un micrófono de unos auriculares.

15.12. Anexo 12: Escenario 3 Pruebas Micrófono Auriculares

https://www.youtube.com/playlist?list=PLqIbJR6nYKfj2adICZtQcwxD2Sm9sHZWF