



# GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

## TRABAJO FIN DE GRADO IA PARA LA ACCESIBILIDAD

Autor: Sergio Rodríguez García

Director: Dr. Michael Thompson

Madrid, España - Waco, Texas, EE. UU.



Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

IA para la Accesibilidad

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2023/24 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido

tomada de otros documentos está debidamente referenciada.



Fdo.: Sergio Rodríguez García

Fecha: 26 / 08 / 2024

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO



Fdo.: Dr. Michael Thompson

Fecha: 20 / 08 / 2024





# GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO  
IA PARA LA ACCESIBILIDAD

Autor: Sergio Rodríguez García

Director: Dr. Michael Thompson

Madrid, España - Waco, Texas, EE. UU.



## **Agradecimientos**

Quiero dedicar este trabajo a mis padres, cuyo apoyo incondicional y fe en mí han sido una fuente constante de motivación. Su esfuerzo, dedicación y amor han sido fundamentales para alcanzar este logro. Gracias por creer siempre en mí y por estar a mi lado en cada paso de este camino.





# **IA PARA LA ACCESIBILIDAD**

**Autor: Rodríguez García, Sergio.**

Director: Thompson, Michael.

Entidad Colaboradora: ICAI – Universidad Pontificia Comillas

## **RESUMEN DEL PROYECTO**

En este proyecto, se ha desarrollado una página web cuyo objetivo es ayudar a personas con problemas del habla mediante diferentes APIs de inteligencia artificial (IA). La IA guía al usuario a través de preguntas sencillas, permitiéndole compartir una historia o idea que tenga en mente. Estas respuestas se procesan para generar una historia en formato de audio utilizando la síntesis de voz personalizada.

**Palabras clave:** API, Inteligencia Artificial, Asistencia, Accesibilidad, Síntesis

### **1. Introducción**

La tecnología juega un papel crucial en la mejora de la calidad de vida de las personas, especialmente para aquellos con necesidades especiales. El proyecto "IA para la Accesibilidad" se centra en proporcionar soluciones tecnológicas innovadoras que mejoren la comunicación de personas con dificultades del habla. A través de una página web, este proyecto utiliza inteligencia artificial para guiar a los usuarios en la estructuración de sus pensamientos, lo que resulta en la generación de narrativas en formato de audio, facilitando así su integración social y autonomía.

### **2. Definición del proyecto**

Este proyecto tiene como objetivo principal desarrollar una página web interactiva que ayude a personas con problemas del habla a expresar sus ideas y sentimientos de manera efectiva. La página utiliza APIs avanzadas para reconocer y procesar el texto introducido por el usuario y convertirlo en un archivo de audio personalizado. Este proceso no solo mejora la capacidad de comunicación de los usuarios, sino que también les proporciona una herramienta accesible y fácil de usar en su vida diaria.

### 3. Descripción del sistema

El sistema desarrollado consta de varias capas que incluyen frontend, backend, base de datos y servicios de integración de terceros. El backend, implementado con Flask, maneja la lógica de la aplicación y la integración con APIs como OpenAI para la generación de texto y Azure para la síntesis de voz. El frontend, desarrollado con HTML, CSS y Bootstrap, proporciona una interfaz accesible y responsiva que guía al usuario a través de una serie de preguntas que ayudan a estructurar la narrativa deseada. La base de datos SQL almacena temporalmente la información del usuario para su posterior procesamiento.

#### Arquitectura del Sistema:

- **Frontend:** HTML, CSS, Bootstrap
- **Backend:** Flask, SQLAlchemy, Integración con APIs de OpenAI (asistente de conversación) y Azure (síntesis de voz)
- **Base de Datos:** SQLite

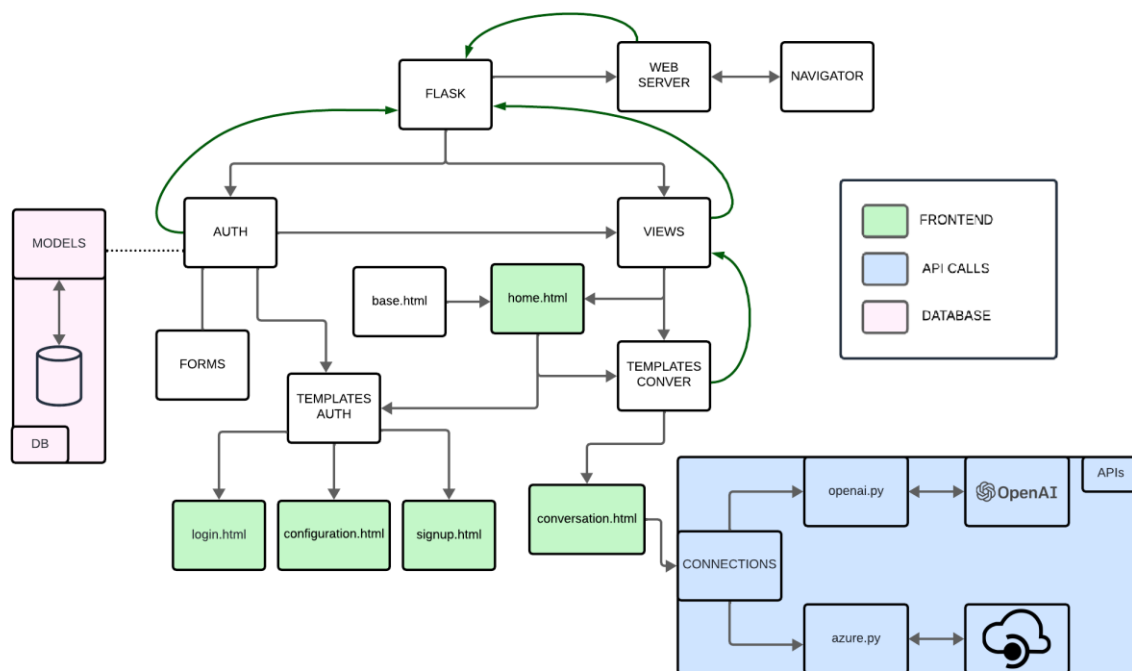


Figura 1 Arquitectura del Sistema

#### **4. Resultados**

El sistema permite a los usuarios crear y escuchar historias personalizadas a partir de sus respuestas a preguntas guiadas. Los resultados demuestran que el sistema es capaz de manejar múltiples interacciones y generar archivos de audio de alta calidad que reflejan la voz del usuario. Además, el sistema ha sido probado en diferentes escenarios, asegurando su funcionalidad y accesibilidad para usuarios con diversas necesidades.

#### **5. Conclusiones**

Este proyecto demuestra el potencial de la inteligencia artificial para mejorar la accesibilidad y la comunicación de personas con dificultades del habla. El sistema desarrollado no solo cumple con los objetivos técnicos propuestos, sino que también ofrece una herramienta valiosa que puede ser utilizada en diversas situaciones para mejorar la calidad de vida de los usuarios. El proyecto abre la puerta a futuras investigaciones y mejoras, como la inclusión de más idiomas y la capacidad de guardar historias para su uso posterior.

# **AI FOR ACCESSIBILITY**

**Author: Rodriguez Garcia, Sergio.**

Supervisor: Thompson, Michael.

Collaborating Entity: ICAI – Pontifical University of Comillas

## **ABSTRACT**

In this project, a website has been developed with the goal of assisting individuals with speech difficulties using various artificial intelligence (AI) APIs. The AI guides the user through simple questions, allowing them to share a story or idea they have in mind. These responses are processed to generate an audio story using personalized voice synthesis.

**Keywords:** API, Artificial Intelligence, Assistance, Accessibility, Synthesis

## **1. Introduction**

Technology plays a crucial role in improving the quality of life for individuals, particularly those with special needs. The project "AI for Accessibility" focuses on providing innovative technological solutions that enhance communication for individuals with speech difficulties. Through a website, this project utilizes artificial intelligence to guide users in structuring their thoughts, resulting in the generation of narratives in audio format, thereby facilitating their social integration and autonomy.

## **2. Project Definition**

The primary objective of this project is to develop an interactive website that helps individuals with speech difficulties express their ideas and feelings effectively. The website utilizes advanced APIs to recognize and process the text input by the user and convert it into a personalized audio file. This process not only enhances the users' communication abilities but also provides them with an accessible and easy-to-use tool in their daily lives.

### 3. System Description

The developed system consists of multiple layers, including frontend, backend, database, and third-party service integrations. The backend, implemented with Flask, handles the application logic and integration with APIs such as OpenAI for text generation and Azure for voice synthesis. The frontend, developed with HTML, CSS, and Bootstrap, provides an accessible and responsive interface that guides the user through a series of questions to help structure the desired narrative. The SQL database temporarily stores the user's information for further processing.

#### System Architecture:

- **Frontend:** HTML, CSS, Bootstrap
- **Backend:** Flask, SQLAlchemy, Integration with OpenAI (assistant) and Azure (voice synthesis) APIs
- **Database:** SQLite

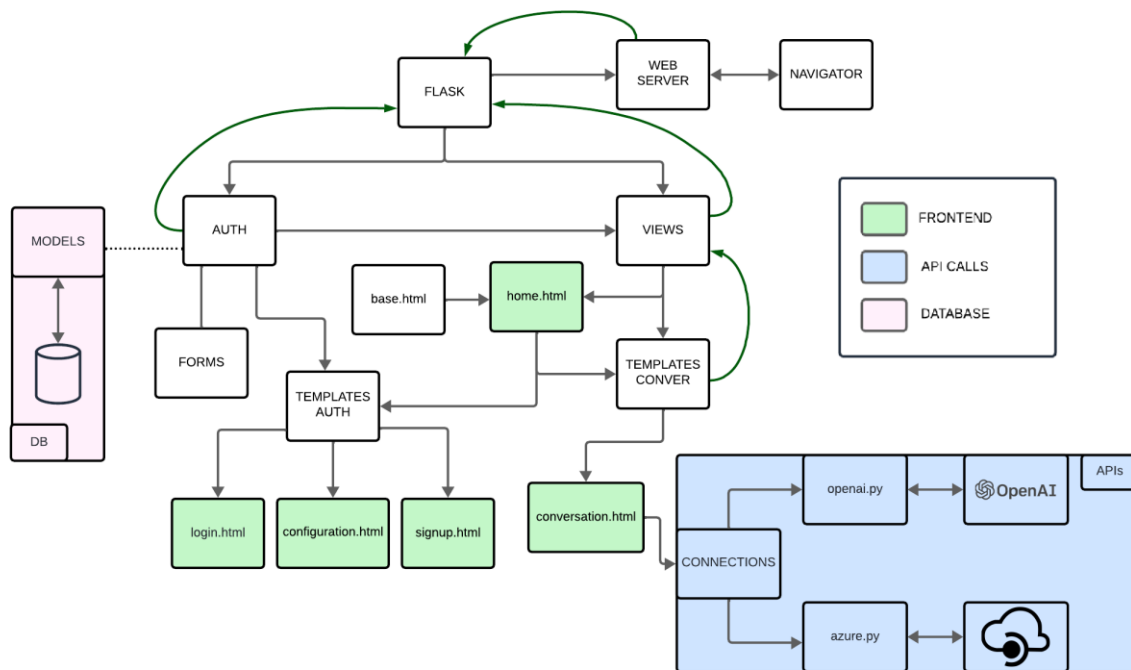


Figura 2 System Architecture

#### **4. Results**

The system allows users to create and listen to personalized stories based on their responses to guided questions. The results demonstrate that the system can handle multiple interactions and generate high-quality audio files that reflect the user's voice. Additionally, the system has been tested in various scenarios, ensuring its functionality and accessibility for users with different needs.

#### **5. Conclusions**

This project demonstrates the potential of artificial intelligence to improve accessibility and communication for individuals with speech difficulties. The developed system not only meets the proposed technical objectives but also provides a valuable tool that can be used in various situations to enhance users' quality of life. The project paves the way for future research and improvements, such as the inclusion of more languages and the ability to save stories for later use.



## Índice de la memoria

<b>Capítulo 1. Introducción .....</b>	<b>7</b>
1.1 Enfoque general.....	8
1.2 Motivaciones .....	9
1.2.1 Motivaciones Sociales .....	9
1.2.2 Motivaciones Tecnológicas .....	10
<b>Capítulo 2. Descripción del Trabajo .....</b>	<b>12</b>
2.1 Justificación.....	13
2.2 Objetivos .....	14
2.2.1 Objetivos Sociales y de Impacto.....	14
2.2.2 Objetivos Técnicos.....	15
2.3 Metodología.....	17
2.3.1 Introducción a Agile y Scrum .....	17
2.3.2 Estructura del Proyecto Scrum.....	18
<b>Capítulo 3. Estado de la Cuestión .....</b>	<b>20</b>
3.1 Trastornos del Habla y la Comunicación .....	21
3.1.1 Condiciones que Pueden Generar Trastornos del Habla.....	22
3.1.2 Impacto Social y Personal de los Trastornos del Habla .....	24
3.1.3 Estadísticas y Prevalencia.....	24
3.2 Beneficios de Guiar una Conversación .....	25
3.2.1 Inclusión Social y Educativa .....	26
3.3 Aplicaciones Similares .....	27
3.3.1 Aplicaciones de Comunicación Asistida.....	28
3.3.2 Aplicaciones de IA Para la Asistencia .....	30
3.3.3 Predictable .....	34
3.3.4 Innovaciones y Diferenciadores del Proyecto.....	36
<b>Capítulo 4. Tecnologías.....</b>	<b>38</b>
4.1 Tecnologías de Reconocimiento y Síntesis del Habla.....	39
4.1.1 Tecnologías de Reconocimiento del Habla .....	41
4.1.2 Tecnologías de Síntesis del Habla.....	42



4.2	Herramientas de Desarrollo .....	44
4.2.1	<i>Visual Studio Code</i> .....	45
4.2.2	<i>GitHub</i> .....	46
4.3	Lenguajes de Programación .....	48
4.3.1	<i>Python</i> .....	48
4.3.2	<i>HTML</i> .....	50
4.3.3	<i>CSS</i> .....	51
4.4	APIs .....	52
4.4.1	<i>APIs de Reconocimiento y Síntesis del Habla</i> .....	53
4.4.2	<i>APIs de Asistencia de Conversación</i> .....	54
4.5	Librerías y FrameWorks Utilizados .....	57
4.5.1	<i>Flask</i> .....	57
4.5.2	<i>SQLAlchemy</i> .....	58
4.5.3	<i>Bootstrap</i> .....	59
4.5.4	<i>Otras Librerías y Sub-Librerías</i> .....	61
<b>Capítulo 5. Sistema Desarrollado .....</b>		<b>64</b>
5.1	Diseño del Sistema.....	64
5.1.1	<i>Arquitectura General</i> .....	64
5.1.2	<i>Casos de Uso</i> .....	68
5.1.3	<i>Requisitos Funcionales y No Funcionales</i> .....	70
5.2	Desarrollo Backend .....	71
5.2.1	<i>Estructura</i> .....	72
5.2.2	<i>Funcionalidades Implementadas</i> .....	73
5.2.3	<i>Flask</i> .....	80
5.2.4	<i>SQLAlchemy</i> .....	85
5.3	Desarrollo Frontend.....	91
5.3.1	<i>Diseño de Interfaz</i> .....	91
5.3.2	<i>Funcionalidades Implementadas</i> .....	95
5.4	Integración de APIs .....	103
5.4.1	<i>OpenAI</i> .....	103
5.4.2	<i>Azure</i> .....	110
<b>Capítulo 6. Conclusiones y Trabajos Futuros .....</b>		<b>113</b>

6.1	Usabilidad de la Aplicación .....	113
6.1.1	<i>Diseño de la Interfaz de Usuario</i> .....	114
6.1.2	<i>Fluidez de la Experiencia del Usuario</i> .....	114
6.1.3	<i>Accesibilidad</i> .....	114
6.1.4	<i>Pruebas de Usabilidad</i> .....	115
6.2	Limitaciones.....	115
6.2.1	<i>Dependencia de APIs Externas</i> .....	116
6.2.2	<i>Latencia en la Respuesta</i> .....	116
6.2.3	<i>Escalabilidad</i> .....	117
6.2.4	<i>Limitación en la Personalización</i> .....	118
6.2.5	<i>Pruebas de Usabilidad Limitadas</i> .....	118
6.3	Trabajos Futuros.....	119
6.3.1	<i>Personalización de la Voz</i> .....	119
6.3.2	<i>Publicar la Aplicación y Servidor para la Base de Datos</i> .....	120
6.3.3	<i>Incluir más idiomas</i> .....	121
6.3.4	<i>Respuestas Personalizadas en el Foro</i> .....	121
6.3.5	<i>Traducción de Comentarios en el Foro</i> .....	122
6.3.6	<i>Gestión de Noticias</i> .....	123
6.3.7	<i>Integración de Reconocimiento de Voz</i> .....	123
6.4	Conclusiones .....	124
	<b>Capítulo 7. Bibliografía</b> .....	<b>126</b>
	<b>ANEXO I: ALINEACIÓN DEL PROYECTO CON LOS ODS</b> .....	<b>131</b>

## *Índice de figuras*

Figura 1 Arquitectura del Sistema .....	10
Figura 2 System Architecture .....	13
Figura 3 Metodología Agile .....	17
Figura 4 Proceso Scrum.....	19
Figura 5 Prologo2Go .....	28
Figura 6 TalkTablet .....	29
Figura 7 Cough Drop.....	30
Figura 8 Dragon NaturallySpeaking.....	31
Figura 9 Otter.ai.....	32
Figura 10 IBM Watson .....	33
Figura 11 Amazon Polly.....	34
Figura 12 Predictable.....	36
Figura 13 VS Code Logo.....	46
Figura 14 Github Logo .....	47
Figura 15 Python Logo .....	50
Figura 16 HTML Logo .....	51
Figura 17 CSS Logo .....	52
Figura 18 OpenAI Logo .....	57
Figura 19 Flask Logo.....	58
Figura 20 SQLAlchemy Logo .....	59
Figura 21 Bootstrap Logo.....	60
Figura 22. Arquitectura del Sistema .....	65
Figura 23 Casos de Uso .....	68
Figura 24 Registro de Usuario.....	95
Figura 25 Inicio de Sesión .....	97
Figura 26 Pantalla de Inicio.....	98

---

Figura 27 Configuración de Usuario .....	99
Figura 28 Selección de Idioma .....	99
Figura 29 Conversación.....	100
Figura 30 Resultados de la Conversación.....	101
Figura 31 Foro de la Aplicación .....	102



## Capítulo 1. INTRODUCCIÓN

En la sociedad actual, la tecnología juega un papel crucial en la mejora de la calidad de vida de las personas, especialmente para aquellos con necesidades especiales. La innovación tecnológica ofrece oportunidades significativas para abordar desafíos que antes parecían insuperables. En este contexto, la accesibilidad se ha convertido en un área de gran interés, donde la inteligencia artificial (IA) puede desempeñar un papel transformador. Este capítulo presenta el marco general del proyecto, destacando su relevancia y las motivaciones detrás de su desarrollo.

El proyecto "IA para la Accesibilidad" surge de la necesidad de crear herramientas que faciliten la comunicación para personas con dificultades del habla, permitiéndoles expresar sus ideas y sentimientos de manera efectiva. Las barreras en la comunicación pueden tener un impacto profundo en la vida diaria y la integración social de las personas

afectadas. La tecnología puede mitigar estas barreras, ofreciendo soluciones que no solo son innovadoras, sino también empáticas y centradas en el usuario.

Se describen los objetivos y el enfoque adoptado para desarrollar una solución tecnológica innovadora que aproveche las capacidades avanzadas de la IA para guiar a los usuarios en la estructuración de sus pensamientos y la generación de narrativas. La página web resultante no solo busca ser una herramienta técnica eficiente, sino también proporcionar una experiencia de usuario intuitiva y accesible, adaptándose a las necesidades individuales de cada usuario

## **1.1 ENFOQUE GENERAL**

El proyecto tiene como objetivo principal desarrollar una página web que asista a personas con dificultades del habla, facilitándoles la comunicación mediante el uso de tecnologías de inteligencia artificial. La página web se construye sobre un conjunto de APIs de IA que guían al usuario a través de una serie de preguntas sencillas, ayudando a estructurar sus ideas y generar una narrativa coherente que puede ser convertida en un archivo de audio en la propia voz del usuario.

El enfoque del proyecto se centra en crear una herramienta accesible y fácil de usar que pueda integrarse en la vida cotidiana de los usuarios. La web no solo está diseñada para ser técnicamente eficiente, sino también para ofrecer una experiencia de usuario intuitiva y amigable. Para lograrlo, se emplean tecnologías modernas como Python y HTML para el desarrollo del backend y frontend respectivamente, así como APIs avanzadas de procesamiento de lenguaje natural (NLP) y síntesis de voz.

Además, este proyecto se enmarca en un contexto social relevante, ya que proporciona una solución innovadora para mejorar la calidad de vida de las personas con trastornos del habla. El uso de la IA en este ámbito representa una combinación poderosa de

tecnología y empatía, ofreciendo a los usuarios una nueva forma de expresar sus pensamientos y sentimientos de manera más fluida y natural.

La plataforma está diseñada para ser escalable y adaptable, permitiendo futuras expansiones y mejoras basadas en los comentarios de los usuarios y los avances tecnológicos. En resumen, el proyecto no solo busca resolver un problema técnico, sino también hacer una contribución significativa a la inclusión social y la accesibilidad.

## **1.2 MOTIVACIONES**

El proyecto tiene varias motivaciones que abarcan aspectos sociales, tecnológicos y económicos. Desde un punto de vista social, se busca mejorar la calidad de vida de las personas con dificultades del habla. En el ámbito tecnológico, se pretende aprovechar los últimos avances en inteligencia artificial para crear una solución innovadora y accesible, demostrando así todo el potencial que pueden llegar a tener estas nuevas tecnologías en el futuro. Finalmente, desde una perspectiva económica, el proyecto busca reducir los costos asociados al tratamiento de estas dificultades y generar nuevas oportunidades en el sector tecnológico.

### **1.2.1 MOTIVACIONES SOCIALES**

Las motivaciones sociales del proyecto se centran en la mejora de la calidad de vida de las personas con dificultades del habla. Estas dificultades pueden surgir debido a diversas condiciones, como trastornos del desarrollo, enfermedades neurológicas o lesiones. La incapacidad de comunicarse eficazmente puede llevar al aislamiento social, baja autoestima y dificultades en el ámbito laboral y educativo. Según la Asociación Americana del Habla-Lenguaje-Audición [1], las dificultades del habla afectan significativamente la interacción social y la calidad de vida de las personas.



Las personas con dificultades del habla a menudo enfrentan barreras significativas en su vida diaria, incluyendo la comunicación con familiares, amigos y compañeros de trabajo. Esto puede llevar a sentimientos de frustración y aislamiento, afectando negativamente su salud mental y bienestar general. La intervención temprana y el uso de tecnologías asistivas pueden mejorar las habilidades de comunicación y, por ende, la calidad de vida de estas personas [2].

Además, en un contexto educativo, los niños con trastornos del habla y del lenguaje pueden tener dificultades para seguir el ritmo de sus compañeros, lo que puede afectar su rendimiento académico y limitar sus oportunidades futuras. Al proporcionar herramientas que apoyen la comunicación, el proyecto puede contribuir a una mayor inclusión educativa y ayudar a estos niños a alcanzar su máximo potencial [3].

En el ámbito laboral, las personas con dificultades del habla pueden enfrentar desafíos adicionales para encontrar y mantener empleo. Las tecnologías que facilitan la comunicación pueden abrir nuevas oportunidades laborales y permitir una participación más plena en el entorno laboral. Esto no solo beneficia a las personas afectadas, sino que también contribuye a una fuerza laboral más diversa e inclusiva [4].

## **1.2.2 MOTIVACIONES TECNOLÓGICAS**

Desde una perspectiva tecnológica, el proyecto busca aprovechar los avances recientes en inteligencia artificial (IA) y procesamiento del lenguaje natural (PLN) para crear una solución innovadora y eficiente. La inteligencia artificial ha evolucionado significativamente en la última década, principalmente el último año, permitiendo el desarrollo de modelos de lenguaje avanzados que pueden comprender y generar texto de manera coherente y contextualmente adecuada. Estos modelos han demostrado ser útiles en diversas aplicaciones, desde asistentes virtuales hasta herramientas de generación de contenido [5].

El uso del procesamiento del lenguaje natural permite que las máquinas entiendan y respondan al lenguaje humano de manera más natural y precisa. Este avance es crucial para nuestro proyecto, ya que facilita la creación de una herramienta interactiva que puede guiar a los usuarios a través de conversaciones estructuradas, adaptándose a sus necesidades individuales y mejorando su experiencia de usuario. El PLN también permite personalizar las respuestas y generar contenido que sea relevante y significativo para cada usuario [6].

Además, el desarrollo de tecnologías de síntesis de voz ha alcanzado un nivel de sofisticación que permite la creación de voces sintéticas que suenan naturales y expresivas. Estas tecnologías no solo son capaces de imitar la voz humana, sino que también pueden adaptarse para reflejar las características vocales específicas de un usuario, proporcionando una experiencia más personalizada y auténtica. Esto es particularmente importante para las personas con dificultades del habla, ya que pueden escuchar sus propias voces generadas digitalmente, lo que puede tener un impacto positivo en su autoestima y motivación para comunicarse [6].

La integración de estas tecnologías en una plataforma web accesible refuerza la importancia de la accesibilidad tecnológica para todos los usuarios. Las plataformas basadas en la web ofrecen la ventaja de ser accesibles desde cualquier dispositivo con conexión a Internet, eliminando barreras geográficas y económicas. Esto es especialmente relevante para nuestro proyecto, ya que busca llegar a una amplia audiencia de personas con dificultades del habla, proporcionando una herramienta que sea fácil de usar y ampliamente disponible.

La adopción de tecnologías como la nube y el aprendizaje automático también desempeña un papel crucial en la escalabilidad y eficiencia del sistema desarrollado. Las soluciones basadas en la nube permiten un procesamiento más rápido y una mayor capacidad de almacenamiento, lo que es esencial para manejar grandes volúmenes de datos y garantizar una experiencia de usuario fluida. El aprendizaje automático, por su parte, mejora continuamente el rendimiento del sistema a medida que procesa más datos, permitiendo una personalización y adaptación constantes a las necesidades de los usuarios [7].

## **Capítulo 2. DESCRIPCIÓN DEL TRABAJO**

Esta sección detalla la justificación, los objetivos, la metodología y el análisis económico del proyecto. La justificación se basa en la necesidad imperiosa de proporcionar herramientas que ayuden a personas con dificultades del habla a comunicarse de manera efectiva, utilizando los últimos avances en IA para superar estas barreras. Los objetivos del proyecto se enfocan en desarrollar un sitio web accesible que facilite la estructuración de pensamientos y la generación de narrativas que puedan ser convertidas en audio utilizando la voz del usuario.

## 2.1 JUSTIFICACIÓN

El proyecto surge de la necesidad urgente de ofrecer herramientas tecnológicas que faciliten la comunicación para personas con dificultades del habla. Estas dificultades pueden deberse a diversas condiciones, como trastornos del desarrollo, enfermedades neurológicas o lesiones, y tienen un impacto significativo en la vida diaria de quienes las padecen [1]. La incapacidad de comunicarse de manera efectiva puede llevar a un aislamiento social, baja autoestima y problemas en el ámbito educativo y laboral [2].

Las personas con dificultades del habla enfrentan barreras significativas para integrarse plenamente en la sociedad. Estas barreras afectan su capacidad para interactuar con familiares, amigos y colegas, lo que puede resultar en frustración y aislamiento [2]. Además, en el contexto educativo, los niños con trastornos del habla pueden tener dificultades para seguir el ritmo de sus compañeros, lo que impacta negativamente su rendimiento académico y sus oportunidades futuras [3]. En el ámbito laboral, estas personas también encuentran obstáculos adicionales para encontrar y mantener empleo, limitando su participación en el entorno laboral y contribuyendo a la exclusión social [4].

El desarrollo de tecnologías basadas en inteligencia artificial, como el procesamiento del lenguaje natural y la síntesis de voz, ofrecen una solución prometedora para superar estas barreras. La IA ha avanzado significativamente en los últimos años, permitiendo la creación de modelos de lenguaje que pueden comprender y generar texto de manera coherente y contextualmente adecuada [5]. Estas tecnologías no solo facilitan la interacción usuario-sistema, sino que también permiten la personalización de las respuestas y la generación de contenido relevante para cada usuario [6].

La integración de tecnologías de síntesis de voz ha alcanzado un nivel de sofisticación que permite la creación de voces sintéticas naturales y expresivas. Estas voces pueden adaptarse para reflejar las características vocales específicas de un usuario, proporcionando una experiencia más auténtica y personalizada [8]. Esto es particularmente importante para las personas con dificultades del habla, ya que escuchar su propia voz

generada digitalmente puede tener un impacto positivo en su autoestima y motivación para comunicarse.

El proyecto también se justifica por su potencial para mejorar la accesibilidad tecnológica. Las plataformas basadas en la web son accesibles desde cualquier dispositivo con conexión a Internet, eliminando barreras geográficas y económicas [7]. Esto es esencial para alcanzar a una amplia audiencia de personas con dificultades del habla, proporcionando una herramienta fácil de usar y ampliamente disponible.

En resumen, el proyecto no solo busca resolver un problema técnico, sino también hacer una contribución significativa a la inclusión social y la accesibilidad. La combinación de tecnologías avanzadas de IA y un enfoque centrado en el usuario puede mejorar significativamente la calidad de vida de las personas con dificultades del habla, facilitando su comunicación y promoviendo su integración social.

## **2.2 OBJETIVOS**

### **2.2.1 OBJETIVOS SOCIALES Y DE IMPACTO**

Como ya se ha mencionado anteriormente el proyecto tiene como objetivo principal claro, mejorar la calidad de vida de las personas con dificultades del habla. A través de la tecnología, se busca facilitar la comunicación de estos individuos, permitiéndoles expresar sus pensamientos y sentimientos de manera más efectiva. Esto puede tener un impacto significativo en su autoestima y motivación para interactuar con los demás, promoviendo una mayor inclusión social y reduciendo el aislamiento al que a menudo se enfrentan [2].

Otro objetivo social es promover la inclusión educativa. Los niños con trastornos del habla a menudo encuentran dificultades para seguir el ritmo de sus compañeros en el entorno escolar, lo cual puede afectar negativamente su rendimiento académico y limitar sus oportunidades futuras [3]. Al proporcionar una herramienta que apoye la comunicación, se

espera facilitar su integración en el entorno educativo, ayudándolos a alcanzar su máximo potencial.

En el ámbito laboral, el proyecto busca abrir nuevas oportunidades para las personas con dificultades del habla. Estas personas suelen enfrentar obstáculos adicionales para encontrar y mantener empleo debido a sus problemas de comunicación [4]. La implementación de tecnologías asistivas puede mejorar su capacidad para interactuar en el entorno laboral, contribuyendo a una fuerza laboral más diversa e inclusiva.

Además, se pretende ofrecer una herramienta accesible y gratuita que pueda ser utilizada por un amplio espectro de usuarios, independientemente de su situación económica. La accesibilidad financiera es crucial para asegurar que todos los individuos que puedan beneficiarse de la tecnología tengan acceso a ella, eliminando barreras económicas y promoviendo la equidad [7].

El proyecto también tiene como objetivo recoger feedback continuo de los usuarios para mejorar la herramienta de manera iterativa. Esto no solo garantizará que el producto final se adapte a las necesidades reales de los usuarios, sino que también fomentará un sentido de comunidad y colaboración entre los desarrolladores y los usuarios [9].

Por último, el proyecto busca contribuir a la investigación en tecnologías de accesibilidad y comunicación asistida. Al documentar y compartir los resultados y aprendizajes del proyecto, se espera inspirar y apoyar otros esfuerzos similares, avanzando en el campo de la accesibilidad tecnológica y promoviendo un entorno más inclusivo para todos [6].

### **2.2.2 OBJETIVOS TÉCNICOS**

Este proyecto tiene varios objetivos técnicos esenciales para su éxito. En primer lugar, se busca desarrollar un sistema backend robusto. Python es un lenguaje de programación altamente versátil y popular en el ámbito de la inteligencia artificial y el

desarrollo web debido a su simplicidad y la amplia disponibilidad de bibliotecas y frameworks [10]. Este backend debe ser capaz de manejar de manera eficiente las interacciones del usuario y las peticiones a las APIs de IA.

En segundo lugar, se pretende implementar un frontend accesible y amigable utilizando HTML y CSS. La accesibilidad web es crucial para garantizar que personas con diversas discapacidades puedan utilizar el sitio sin dificultades. Una buena accesibilidad web implica diseñar interfaces que sean utilizables por personas con discapacidades visuales, auditivas, motoras y cognitivas [11].

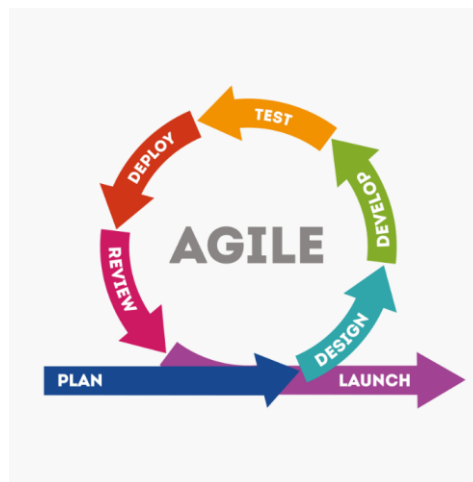
La integración de APIs de NLP y síntesis de voz es otro objetivo técnico importante. Estas APIs permiten transformar el texto en habla y viceversa, facilitando una comunicación más natural y efectiva para los usuarios [5]. Específicamente, se utilizarán APIs avanzadas que pueden personalizarse para adaptarse a las características vocales específicas de cada usuario, mejorando así la experiencia de usuario.

La seguridad y privacidad de los datos de los usuarios también son aspectos críticos. Se implementarán medidas de seguridad avanzadas para proteger la información personal y garantizar que los datos de los usuarios se manejen de manera confidencial y segura, cumpliendo con las normativas de protección de datos vigentes [12].

Finalmente, se desarrollará un prototipo funcional del sitio web, el cual será sometido a pruebas de usabilidad y accesibilidad con usuarios finales. Estas pruebas permitirán identificar y corregir posibles problemas antes de lanzar el producto final, asegurando que sea intuitivo y fácil de usar [9]. Además, se documentará detalladamente todo el proceso de desarrollo y las decisiones técnicas tomadas, lo cual es esencial para futuras mejoras y expansiones del proyecto.

## 2.3 METODOLOGÍA

La metodología empleada para el desarrollo del proyecto ha sido Agile, específicamente el marco de trabajo Scrum. Este enfoque se eligió por su flexibilidad y capacidad para adaptarse a los cambios, lo cual es crucial en proyectos que implican desarrollo tecnológico avanzado y retroalimentación constante de usuarios.



*Figura 3 Metodología Agile*

### 2.3.1 INTRODUCCIÓN A AGILE Y SCRUM

**Agile** es una metodología de gestión de proyectos que se centra en la entrega incremental e iterativa de productos. Se basa en principios como la colaboración estrecha con los clientes, la adaptabilidad al cambio y la entrega continua de valor. Dentro de Agile, Scrum es uno de los marcos más populares y estructurados. **Scrum** divide el trabajo en sprints, que son ciclos cortos de desarrollo, generalmente de dos a cuatro semanas [7].



### **2.3.2 ESTRUCTURA DEL PROYECTO SCRUM**

En este proyecto, el trabajo se organizó siguiendo los principios de **Scrum**, adaptando la estructura para un solo desarrollador bajo la supervisión de un tutor. La supervisión se realizó mediante reuniones semanales, donde se revisaba el progreso, se discutían los obstáculos y se planificaban las siguientes tareas.

**Roles y Responsabilidades:** Aunque tradicionalmente Scrum incluye varios roles como el Product Owner, el Scrum Master y el Equipo de Desarrollo, en este proyecto, se asumieron todos estos roles con la orientación del tutor. Se actuó como Product Owner, definiendo y priorizando los requisitos del proyecto; como Scrum Master, facilitando el proceso y eliminando impedimentos; y como Equipo de Desarrollo, implementando las soluciones técnicas necesarias.

**Ceremonias:** A lo largo del proyecto, se llevaron a cabo varias ceremonias clave de Scrum adaptadas a un entorno de trabajo individual. Al inicio de cada sprint semanal, se realizaba una reunión de planificación con el tutor para definir los objetivos y tareas específicas de la semana. Durante las reuniones semanales, que funcionaron como una combinación de Daily Stand-up y Sprint Review, se discutía el progreso de las tareas, se demostraban las funcionalidades completadas y se recogía feedback. Al finalizar cada sprint, se realizaba una breve retrospectiva para reflexionar sobre el proceso y discutir posibles mejoras.

**Artefactos:** Se utilizó un Product Backlog, una lista priorizada de todas las tareas y requisitos del proyecto, que se actualizaba y refinaba continuamente en colaboración con el tutor. Cada sprint contaba con un Sprint Backlog, que incluía las tareas seleccionadas para ser completadas durante la semana. Los incrementos del producto, es decir, las funcionalidades completadas, se revisaban semanalmente con el tutor para asegurar que cumplieran con los objetivos establecidos y se alinearan con la visión del proyecto.

En resumen, la metodología Agile y el marco de trabajo Scrum proporcionaron una estructura flexible y adaptable que facilitó la gestión eficiente del proyecto, permitiendo iteraciones rápidas y una mejora continua basada en la retroalimentación constante del tutor.

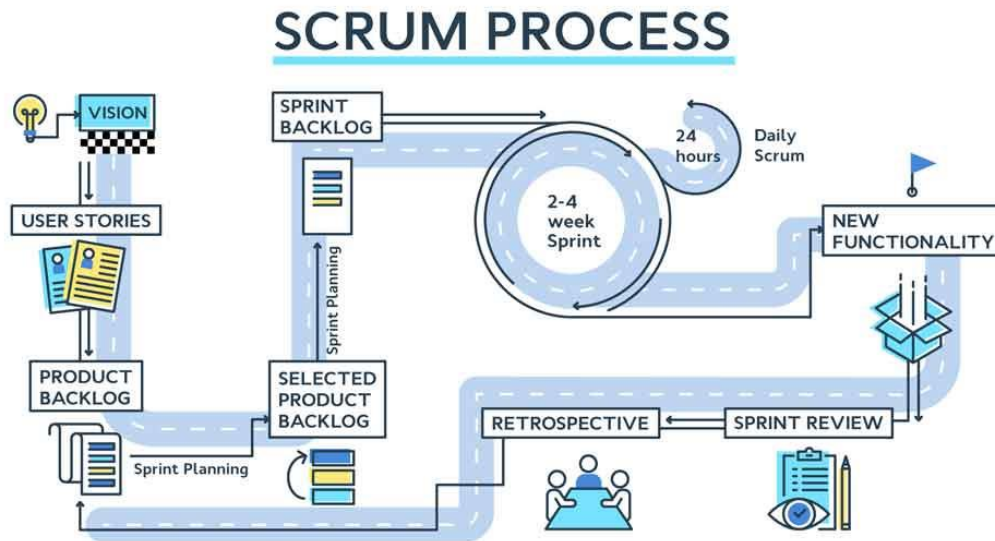


Figura 4 Proceso Scrum

## **Capítulo 3. ESTADO DE LA CUESTIÓN**

La comprensión del contexto y las bases sobre las cuales se asienta el proyecto es fundamental para apreciar su relevancia y potencial impacto. Este capítulo se dedica a explorar y analizar el estado actual del conocimiento y la práctica en el ámbito de los trastornos del habla y la comunicación, así como las soluciones tecnológicas existentes que abordan estos desafíos.

En primer lugar, se proporcionará una definición y clasificación de los trastornos del habla, destacando las diversas formas en que pueden manifestarse y afectar a los individuos. A continuación, se examinarán las enfermedades y condiciones que pueden generar estos trastornos, incluyendo tanto trastornos del desarrollo como enfermedades neurológicas y lesiones traumáticas.

Se discutirá también el impacto social y personal de los trastornos del habla, ilustrando cómo estas condiciones pueden afectar la calidad de vida, la autoestima y la integración social de las personas afectadas. Se presentarán estadísticas y datos sobre la prevalencia de estos trastornos para proporcionar una perspectiva cuantitativa de su alcance.

El capítulo continuará con una revisión de los beneficios de las conversaciones guiadas como una intervención para mejorar la comunicación y la expresión personal, subrayando cómo estas técnicas pueden potenciar la autoestima y la inclusión social y educativa.

Finalmente, se analizarán las aplicaciones similares existentes, revisando herramientas tecnológicas que han sido desarrolladas para asistir a personas con dificultades del habla. Se compararán sus funcionalidades y tecnologías empleadas, y se evaluarán sus fortalezas y debilidades, proporcionando un marco de referencia para el desarrollo del proyecto actual.

### **3.1 TRASTORNOS DEL HABLA Y LA COMUNICACIÓN**

Los trastornos del habla y la comunicación son condiciones que afectan la capacidad de una persona para producir sonidos que forman palabras o para entender y expresar el lenguaje hablado. Estos trastornos pueden manifestarse de diversas maneras, afectando la articulación, la fluidez, la resonancia y la voz [1].

- **Trastornos de la articulación:** Dificultades para producir sonidos correctamente, lo que puede hacer que el habla sea difícil de entender. Ejemplos incluyen la dislalia y la disartria [2].
- **Trastornos de la fluidez:** Problemas con el ritmo y la fluidez del habla, como el tartamudeo [2].

- **Trastornos de la voz:** Alteraciones en la calidad, tono, volumen o duración de los sonidos vocales, que pueden ser causadas por problemas en las cuerdas vocales o por un uso inadecuado de la voz [1].
- **Trastornos del lenguaje:** Dificultades para entender o usar palabras en contextos apropiados. Estos pueden ser receptivos (problemas para entender el lenguaje) o expresivos (problemas para usar el lenguaje) [3].

La clasificación de los trastornos del habla y la comunicación se basa en la naturaleza y la causa subyacente del problema. Los trastornos pueden ser congénitos, presentes desde el nacimiento, o adquiridos, desarrollándose a lo largo de la vida debido a una variedad de factores. Además, pueden ser primarios, sin otra condición asociada, o secundarios, resultantes de otra condición médica [4].

Las siguientes secciones explorarán en detalle las enfermedades y condiciones que pueden causar estos trastornos, el impacto social y personal de los mismos, y estadísticas sobre su prevalencia en la población.

### **3.1.1 CONDICIONES QUE PUEDEN GENERAR TRASTORNOS DEL HABLA**

Los trastornos del habla y la comunicación pueden ser causados por una variedad de enfermedades y condiciones que afectan el desarrollo, la función neurológica y la estructura física de las áreas relacionadas con la producción y comprensión del lenguaje. A continuación, se describen algunas de las principales causas.

Los trastornos del desarrollo, como el Trastorno del Espectro Autista (TEA), la dislexia y el síndrome de Down, son causas comunes de dificultades en la comunicación. El TEA se asocia frecuentemente con problemas significativos en la comunicación y el habla, incluyendo retrasos en el habla, problemas con la comunicación no verbal y 16 dificultades para mantener conversaciones [1]. Por otro lado, la dislexia, aunque principalmente afecta la capacidad de leer y escribir, también puede impactar la fluidez del habla y la capacidad para encontrar las palabras adecuadas durante la conversación [13]. El síndrome de Down,

una condición genética que causa retrasos en el desarrollo, también puede llevar a problemas de habla y lenguaje, incluyendo dificultades en la articulación y en la construcción de frases complejas [14].

Las enfermedades neurológicas son otra causa importante de trastornos del habla. La esclerosis múltiple, una enfermedad autoinmune que afecta el sistema nervioso central, puede causar problemas neurológicos variados, incluyendo la disartria, que se caracteriza por un habla lenta y descoordinada [15]. La parálisis cerebral, que afecta las habilidades motoras finas y gruesas, también puede causar dificultades en la articulación y coordinación del habla [16].

Las lesiones cerebrales traumáticas (TBI) son otra causa significativa de problemas de comunicación. Dependiendo de la gravedad y la localización de la lesión, las TBI pueden resultar en dificultades con el habla, el lenguaje, la lectura, la escritura y la comprensión del lenguaje [17].

Enfermedades degenerativas como la enfermedad de Parkinson y la enfermedad de Huntington también pueden generar trastornos del habla. La enfermedad de Parkinson, que afecta principalmente el control motor, puede llevar a una voz suave y monótona, además de problemas con la articulación [18]. La enfermedad de Huntington, una enfermedad genética que causa la degeneración progresiva de las neuronas en el cerebro, también afecta las capacidades motoras y del habla [19].

Además, otras condiciones como los accidentes cerebrovasculares y los tumores cerebrales pueden causar trastornos del habla. Los accidentes cerebrovasculares pueden resultar en afasia, una condición que afecta la capacidad de una persona para hablar, entender el habla, leer y escribir [20]. Los tumores en áreas del cerebro relacionadas con el lenguaje pueden generar trastornos del habla y del lenguaje, dependiendo de su tamaño y ubicación [21].

### **3.1.2 IMPACTO SOCIAL Y PERSONAL DE LOS TRASTORNOS DEL HABLA**

Los trastornos del habla y la comunicación tienen un impacto profundo tanto en la vida personal como social de las personas afectadas. A nivel personal, estas dificultades pueden generar una gran frustración y ansiedad, ya que la capacidad de comunicarse de manera efectiva es fundamental para expresar necesidades, deseos y emociones. La incapacidad para hacerlo puede llevar a sentimientos de aislamiento, baja autoestima y, en algunos casos, a la depresión [2].

En el ámbito social, los trastornos del habla pueden dificultar la interacción con familiares, amigos y colegas, lo que a menudo resulta en aislamiento social. Las personas con dificultades del habla pueden encontrarse excluidas de conversaciones y actividades sociales, lo que agrava aún más su sensación de aislamiento. En entornos educativos, los niños con trastornos del habla pueden tener dificultades para seguir el ritmo de sus compañeros, afectando negativamente su rendimiento académico y limitando sus oportunidades futuras [3].

En el entorno laboral, las dificultades del habla pueden limitar las oportunidades de empleo y promoción, ya que la comunicación efectiva es una habilidad crucial en la mayoría de los trabajos. Esto puede resultar en una menor participación en la fuerza laboral y una mayor dependencia de la asistencia social [4].

### **3.1.3 ESTADÍSTICAS Y PREVALENCIA**

Los trastornos del habla y la comunicación son relativamente comunes y pueden afectar a personas de todas las edades. Aproximadamente el 7-10% de la población mundial tiene algún tipo de trastorno del habla o del lenguaje [1].

En niños, la prevalencia de estos trastornos es notablemente alta. Estudios indican que alrededor del 5-8% de los niños en edad preescolar tienen algún tipo de trastorno del habla o del lenguaje, siendo más común en niños que en niñas. En el ámbito educativo, los

trastornos de la comunicación son una de las razones más frecuentes para la provisión de servicios de educación especial [3].

En adultos, los trastornos del habla a menudo resultan de condiciones adquiridas como accidentes cerebrovasculares, lesiones cerebrales traumáticas y enfermedades neurodegenerativas. Por ejemplo, aproximadamente el 25-40% de los sobrevivientes de accidentes cerebrovasculares presentan afasia, una condición que afecta la capacidad de hablar y entender el lenguaje [20]. La prevalencia de trastornos del habla en enfermedades neurodegenerativas como la enfermedad de Parkinson y la enfermedad de Huntington también es significativa, afectando a una proporción sustancial de pacientes [18].

Estas estadísticas subrayan la importancia de desarrollar y proporcionar intervenciones efectivas para apoyar a las personas con trastornos del habla y la comunicación, mejorando su calidad de vida y facilitando su inclusión social y económica.

### **3.2 BENEFICIOS DE GUIAR UNA CONVERSACIÓN**

Las conversaciones guiadas ofrecen múltiples beneficios para las personas con dificultades del habla, especialmente en términos de mejorar su capacidad de comunicación y expresión personal.

Uno de los principales desafíos que enfrentan las personas con trastornos del habla es la dificultad para estructurar sus pensamientos de manera coherente. Las conversaciones guiadas proporcionan un marco estructurado que ayuda a los usuarios a organizar sus ideas de manera lógica. Mediante preguntas específicas y prompts, el sistema guía al usuario a través de un proceso paso a paso, facilitando la organización de sus pensamientos antes de verbalizarlos. Esto no solo mejora la claridad de la comunicación, sino que también reduce la ansiedad asociada con el acto de hablar [2].



La claridad en la expresión es fundamental para una comunicación efectiva. Las conversaciones guiadas ayudan a los usuarios a enfocarse en puntos clave y a eliminar la ambigüedad en sus respuestas. Al recibir indicaciones claras y estructuradas, los usuarios pueden formular sus ideas de manera más precisa y directa. Esto es especialmente útil para aquellos que pueden tener dificultades para encontrar las palabras adecuadas o para aquellos que tienden a desviarse del tema principal [4].

La ansiedad al hablar es un problema común entre las personas con dificultades del habla. Las conversaciones guiadas ofrecen un entorno seguro y controlado en el que los usuarios pueden practicar sus habilidades de comunicación sin el miedo al juicio o la crítica inmediata. Saber que tienen una estructura y apoyo durante la conversación puede aliviar significativamente la ansiedad, permitiéndoles concentrarse en el contenido de su mensaje en lugar de preocuparse por posibles errores o malentendidos. Además, la práctica repetida en un entorno seguro puede aumentar la confianza de los usuarios en sus habilidades de comunicación, lo que se traduce en una mayor fluidez y seguridad en situaciones de la vida real [3].

### **3.2.1 INCLUSIÓN SOCIAL Y EDUCATIVA**

Las conversaciones guiadas no solo mejoran la comunicación individual, sino que también tienen un impacto significativo en la inclusión social y educativa de las personas con dificultades del habla.

Los estudiantes con dificultades del habla a menudo enfrentan desafíos adicionales en el entorno educativo, desde la comprensión de las instrucciones hasta la participación en actividades de grupo. Las conversaciones guiadas pueden ser una herramienta valiosa para estos estudiantes, ayudándoles a estructurar sus respuestas y participar más activamente en clase. Al proporcionar un marco claro y apoyo estructurado, las conversaciones guiadas facilitan la comprensión y la expresión, permitiendo que los estudiantes sigan el ritmo de sus compañeros y se sientan más incluidos en el aula [3]. Además, estas herramientas pueden

ser utilizadas por los profesores para adaptar sus métodos de enseñanza y apoyar mejor a los estudiantes con necesidades especiales [2].

La capacidad de comunicarse efectivamente es crucial para la participación en la vida social. Las personas con dificultades del habla a menudo se sienten excluidas de las conversaciones y actividades sociales debido a sus limitaciones de comunicación. Las conversaciones guiadas pueden reducir estas barreras, proporcionando a los usuarios las herramientas necesarias para interactuar con los demás de manera más efectiva. Esto no solo mejora su capacidad para formar y mantener relaciones, sino que también aumenta su confianza y autoestima, promoviendo una mayor inclusión social [4].

La comunicación efectiva es una habilidad clave en el ámbito laboral. Las personas con dificultades del habla pueden encontrar obstáculos significativos al buscar empleo o intentar avanzar en sus carreras debido a sus limitaciones comunicativas. Las conversaciones guiadas pueden ayudar a estos individuos a preparar respuestas claras y coherentes para entrevistas de trabajo y situaciones laborales cotidianas. Al mejorar su capacidad de comunicación, estas herramientas abren nuevas oportunidades laborales y permiten una mayor participación en el entorno profesional. Esto no solo beneficia a los individuos afectados, sino que también contribuye a una fuerza laboral más diversa e inclusiva [4].

### **3.3 APLICACIONES SIMILARES**

En el ámbito de las tecnologías asistivas y la inteligencia artificial, existen diversas aplicaciones y plataformas diseñadas para facilitar la comunicación de personas con dificultades del habla y para asistir a los usuarios en diversas tareas mediante el uso de IA. Este apartado examina tanto los sistemas de comunicación aumentativa y alternativa 20 (AAC) que ayudan directamente con la producción del habla, como las aplicaciones que emplean inteligencia artificial para ofrecer soporte en otras áreas, proporcionando un contexto valioso para entender el panorama tecnológico en el que se inserta el proyecto. Al

analizar estas aplicaciones, se identificarán tanto las similitudes como las diferencias clave en relación con el proyecto, destacando las innovaciones y ventajas competitivas que hacen única esta propuesta.

### 3.3.1 APLICACIONES DE COMUNICACIÓN ASISTIDA

#### 3.3.1.1 PROLOQUO2GO

**Proloquo2Go** es una aplicación de comunicación aumentativa y alternativa (AAC) diseñada para ayudar a personas con dificultades del habla a comunicarse de manera efectiva. Desarrollada por AssistiveWare, esta aplicación es especialmente popular entre niños y adultos con autismo, parálisis cerebral, síndrome de Down y otras condiciones que afectan la capacidad de hablar. La aplicación ofrece una interfaz altamente personalizable que permite a los usuarios seleccionar palabras y frases a través de pictogramas y símbolos, que luego son convertidos en habla por la aplicación. Incluye una amplia biblioteca de vocabulario con opciones para diferentes niveles de competencia lingüística, desde principiantes hasta usuarios avanzados. La aplicación también permite la personalización del vocabulario y la disposición de los botones para adaptarse a las necesidades individuales de cada usuario [22].



*Figura 5 Proloquo2Go*

### 3.3.1.2 TALKTABLET

**TalkTablet** es una aplicación AAC diseñada para ofrecer una solución de comunicación eficaz a personas con dificultades del habla. Es compatible con dispositivos iOS, Android y Windows, proporcionando una opción multiplataforma para los usuarios (TalkTablet, 2024). Permite a los usuarios crear y personalizar tableros de comunicación con símbolos y texto. La aplicación ofrece voces sintéticas naturales y una interfaz fácil de usar, lo que facilita la creación de mensajes complejos a partir de combinaciones de símbolos y texto. Además, incluye una biblioteca de símbolos extensa y opciones de personalización para adaptarse a las preferencias y necesidades específicas de cada usuario [23].



*Figura 6 TalkTablet*

### 3.3.1.3 COUGHDROP

CoughDrop es una aplicación AAC basada en la nube que está diseñada para facilitar la comunicación de personas con dificultades del habla. Su enfoque en la accesibilidad y la colaboración la hace adecuada tanto para usuarios individuales como para equipos de terapeutas y educadores. CoughDrop permite a los usuarios crear tableros de comunicación personalizados utilizando una amplia biblioteca de pictogramas y símbolos. La aplicación se destaca por su capacidad de sincronización en la nube, lo que permite a los usuarios acceder

a sus tableros desde cualquier dispositivo con conexión a Internet. Además, ofrece herramientas de seguimiento y análisis para supervisar el uso y el progreso del usuario, facilitando la colaboración entre terapeutas, educadores y familias [24].



*Figura 7 Cough Drop*

### **3.3.2 APLICACIONES DE IA PARA LA ASISTENCIA**

#### **3.3.2.1 DRAGON NATURALLY SPEAKING**

**Dragon NaturallySpeaking**, desarrollado por Nuance, es una de las aplicaciones de reconocimiento de voz más avanzadas y precisas disponibles en el mercado. Esta herramienta está diseñada para convertir el habla en texto en tiempo real, lo que permite a los usuarios dictar documentos, correos electrónicos y navegar por sus computadoras utilizando solo su voz. Es ampliamente utilizada por profesionales que necesitan transcribir grandes volúmenes de texto rápidamente, así como por personas con discapacidades físicas que encuentran difícil usar un teclado y un ratón. Dragon NaturallySpeaking ofrece una precisión de reconocimiento de voz excepcional, permitiendo a los usuarios dictar texto con gran exactitud, lo que es especialmente útil para la redacción de documentos largos. Los usuarios pueden controlar diversas aplicaciones de su computadora mediante comandos de voz, incluyendo la navegación por internet, el uso de procesadores de texto, y la gestión de correos electrónicos. Además, los usuarios pueden crear comandos de voz personalizados para realizar tareas específicas, mejorando así la eficiencia y la productividad. La aplicación

se adapta a la voz del usuario con el tiempo, mejorando su precisión. También puede aprender palabras y frases específicas para mejorar aún más la transcripción. Dragon NaturallySpeaking también puede transcribir grabaciones de audio previamente realizadas, convirtiéndolas en texto editable [25].



*Figura 8 Dragon NaturallySpeaking*

### 3.3.2.2 OTTER.AI

**Otter.ai** es una aplicación de transcripción que utiliza inteligencia artificial para convertir voz en texto en tiempo real. Esta herramienta es ideal para tomar notas durante reuniones, entrevistas y conferencias, proporcionando una transcripción precisa y editable que puede ser compartida y revisada fácilmente. Otter.ai es utilizado por profesionales de diversos campos, incluyendo periodistas, investigadores y ejecutivos, que necesitan capturar y analizar conversaciones de manera eficiente. Otter.ai proporciona transcripción en tiempo real, permitiendo a los usuarios ver el texto generado a medida que hablan. La aplicación es capaz de identificar diferentes altavoces, lo que facilita la organización y comprensión de las conversaciones. Otter.ai se integra con otras 22 herramientas de productividad como Zoom, Google Meet y Dropbox, permitiendo una sincronización y colaboración fluidas. Los usuarios pueden buscar palabras clave y frases dentro de las transcripciones, haciendo que la revisión y el análisis de las conversaciones sean más eficientes. Además, Otter.ai ofrece

la capacidad de exportar transcripciones en varios formatos, incluyendo texto y archivos de audio [26].



*Figura 9 Otter.ai*

### 3.3.2.3 IBM WATSON TEXT TO SPEECH

**IBM Watson Text to Speech** es una aplicación de síntesis de voz que convierte texto en voz natural utilizando tecnología avanzada de inteligencia artificial. Este servicio está diseñado para ayudar a las empresas y desarrolladores a crear aplicaciones y dispositivos que puedan interactuar con los usuarios a través de una voz generada digitalmente. Es utilizado en una variedad de industrias, incluyendo atención al cliente, educación y salud, para mejorar la accesibilidad y la experiencia del usuario. IBM Watson Text to Speech permite la personalización de voces, permitiendo a las empresas elegir entre una amplia gama de voces masculinas y femeninas en múltiples idiomas y acentos. Los usuarios pueden ajustar la pronunciación y entonación de las voces para que suenen más naturales y adecuadas al contexto de uso. La aplicación se integra fácilmente con otros servicios y aplicaciones web, permitiendo una implementación flexible en diversas plataformas. IBM Watson Text to Speech también ofrece la capacidad de generar voz en tiempo real, lo que es útil para aplicaciones que requieren interacción instantánea con los usuarios. La tecnología

subyacente está constantemente mejorando, lo que asegura que las voces generadas suenen cada vez más humanas y naturales [27].



*Figura 10 IBM Watson*

### 3.3.2.4 AMAZON POLLY

**Amazon Polly** es un servicio de conversión de texto a voz que utiliza inteligencia artificial para generar voz de alta calidad. Este servicio es parte de la suite de productos de Amazon Web Services (AWS) y está diseñado para desarrollar aplicaciones y dispositivos que necesiten capacidades de síntesis de voz. Amazon Polly es ampliamente utilizado en diversas aplicaciones, desde lectores de pantalla hasta asistentes virtuales y servicios de atención al cliente. Amazon Polly ofrece una amplia selección de voces y idiomas, permitiendo a los desarrolladores elegir la voz que mejor se adapte a sus necesidades específicas. Los usuarios pueden personalizar la pronunciación de palabras y frases para que la salida de voz sea más precisa y adecuada al contexto. La integración con otras aplicaciones y dispositivos es sencilla, gracias a la flexibilidad de la API de Amazon Polly. Además, Amazon Polly puede generar voz en tiempo real, lo que es crucial para aplicaciones interactivas que requieren respuestas instantáneas. La tecnología de síntesis de voz de Amazon Polly está diseñada para ser escalable, permitiendo su uso en aplicaciones que deben manejar grandes volúmenes de solicitudes de voz [28].





*Figura 11 Amazon Polly*

### 3.3.3 PREDICTABLE

**Predictable** es una aplicación de comunicación aumentativa y alternativa (AAC) diseñada para personas con dificultades del habla. Esta herramienta utiliza tecnología avanzada para facilitar la comunicación a través de la predicción de texto y la conversión de texto a voz. Su principal objetivo es mejorar la calidad de vida de sus usuarios, permitiéndoles expresar sus pensamientos y necesidades de manera más eficiente y efectiva.

El objetivo público de Predictable incluye personas de todas las edades que tienen dificultades para comunicarse verbalmente. Esto puede incluir individuos con trastornos del desarrollo como el autismo, parálisis cerebral, síndrome de Down, así como aquellos que han sufrido lesiones neurológicas o enfermedades degenerativas que afectan el habla. La aplicación está diseñada para ser accesible y fácil de usar, lo que la hace adecuada tanto para niños como para adultos.

Ofrece una función de predicción de texto que sugiere palabras y frases a medida que el usuario escribe, facilitando una comunicación más rápida y precisa. Esta característica es especialmente útil para personas con dificultades motoras o con un vocabulario limitado, ya que reduce la cantidad de escritura necesaria para formar oraciones completas.

Una de las principales funcionalidades de Predictable es la conversión de texto a voz, permitiendo a los usuarios escuchar sus mensajes en voz alta. Esto no solo ayuda a las personas con dificultades del habla a comunicarse más fácilmente, sino que también les permite interactuar de manera más natural en conversaciones cotidianas.

Predictable permite la importación de voces personalizadas, lo que significa que los usuarios pueden tener una voz que se asemeje a la suya propia o elegir una voz que prefieran. Esta personalización es crucial para la identidad y la autoestima de los usuarios, ya que les permite comunicarse de una manera que se siente más auténtica y personal.

La aplicación ofrece amplias opciones de personalización de la interfaz de usuario, permitiendo a cada individuo adaptar el diseño y la funcionalidad de la aplicación a sus necesidades específicas. Esto incluye la posibilidad de ajustar el tamaño de los botones, los colores y el diseño general de la pantalla, haciendo que la aplicación sea más accesible y fácil de usar para personas con diversas discapacidades.

Predictable está disponible en múltiples plataformas, incluyendo iOS y Android, lo que permite a los usuarios acceder a la aplicación desde una variedad de dispositivos. Esta accesibilidad multiplataforma asegura que los usuarios puedan utilizar la aplicación en cualquier momento y lugar, proporcionando una mayor flexibilidad y conveniencia [29].

El modo de escaneo es una característica diseñada para usuarios con movilidad extremadamente limitada. Este modo permite navegar por la aplicación mediante la utilización de dispositivos de entrada alternativos, como interruptores o dispositivos de seguimiento ocular, facilitando la comunicación para personas con discapacidades motoras severas.

De la comparación con Predictable, se ha aprendido la importancia de ofrecer una interfaz altamente personalizable y accesible. También hemos visto el valor de la predicción de texto para mejorar la eficiencia de la comunicación. Estas lecciones nos han ayudado a identificar áreas clave para el desarrollo y mejora de nuestro propio proyecto, asegurando

que ofrecemos una herramienta que no solo es técnicamente avanzada, sino también centrada en las necesidades y preferencias de los usuarios.



*Figura 12 Predictable*

### **3.3.4 INNOVACIONES Y DIFERENCIADORES DEL PROYECTO**

**IA para la Accesibilidad** se destaca notablemente en comparación con las aplicaciones existentes en el ámbito de la comunicación asistida gracias a sus características innovadoras y diferenciadoras.

En términos de funcionalidades de comunicación asistida, como ya se ha observado, existen aplicaciones como Proloquo2Go, TalkTablet y CoughDrop, que ofrecen interfaces personalizables y amplias bibliotecas de pictogramas y símbolos para facilitar la comunicación. Sin embargo, estas soluciones tienden a limitarse a funciones básicas y no siempre aprovechan al máximo las capacidades de la IA.

El proyecto integra tecnologías avanzadas de IA para ofrecer una experiencia de usuario más natural y personalizada. A diferencia de herramientas como Dragon NaturallySpeaking y Otter.ai, que se centran en la transcripción y dictado para un público más amplio, el proyecto utiliza modelos de NLP y síntesis de voz de última generación. Esto permite adaptar las respuestas y la voz generada a las características y preferencias individuales de cada usuario, mejorando así la interacción y empatía.

La interfaz de usuario intuitiva y adaptable es otro diferenciador clave. Diseñada específicamente para personas con trastornos del habla, la interfaz facilita una navegación

sencilla y permite personalizar las opciones según las necesidades del usuario. Esta característica es crucial para garantizar la accesibilidad y usabilidad, aspectos en los que el proyecto se enfoca especialmente para superar las limitaciones de otras aplicaciones existentes.

Una innovación significativa es la integración de conversaciones guiadas. En lugar de ofrecer herramientas estáticas de comunicación, el proyecto guía a los usuarios a través de preguntas y prompts estructurados que ayudan a organizar sus pensamientos y generar narrativas coherentes. Esto no solo mejora la claridad y efectividad de la comunicación, sino que también reduce la ansiedad asociada con el acto de hablar, un beneficio que no se encuentra comúnmente en otras aplicaciones.

Finalmente, la accesibilidad y disponibilidad multiplataforma asegura que la aplicación pueda ser utilizada desde cualquier dispositivo con conexión a Internet. Esto 25 elimina barreras geográficas y económicas, permitiendo que una amplia gama de usuarios se beneficie de la herramienta, independientemente de su ubicación o recursos financieros.

## Capítulo 4. TECNOLOGÍAS

En este capítulo se detallarán las diversas tecnologías utilizadas en el desarrollo del proyecto. El objetivo principal de este proyecto es crear una herramienta accesible y eficiente que permita a las personas con dificultades del habla comunicarse de manera más efectiva. Para lograrlo, se ha seleccionado un conjunto de tecnologías avanzadas y modernas, que aseguran no solo la funcionalidad deseada, sino también la escalabilidad y la adaptabilidad a futuras necesidades y mejoras.

El desarrollo del proyecto se ha basado en tecnologías de vanguardia en el ámbito del reconocimiento y la síntesis del habla, las cuales permiten que el sistema comprenda y genere lenguaje de manera natural y precisa. Estas tecnologías son fundamentales para garantizar una interacción fluida y personalizada con los usuarios.

Además, se han empleado lenguajes de programación robustos y versátiles, como Python y HTML, que facilitan tanto el desarrollo del backend como del frontend de la aplicación. Python, conocido por su simplicidad y potencia, ha sido crucial para implementar la lógica del servidor y las interacciones con las APIs de inteligencia artificial, mientras que HTML ha permitido construir una interfaz de usuario accesible y amigable.

Las llamadas a APIs de IA de proveedores líderes como OpenAI y Azure han sido integrales para incorporar capacidades avanzadas de NLP y síntesis de voz. Estas APIs proporcionan los recursos necesarios para personalizar la experiencia del usuario y asegurar una alta calidad en la generación de contenido de audio.

Por último, se destacarán las librerías y frameworks utilizados, como Flask, que han sido esenciales para el desarrollo ágil y eficiente del proyecto. Esta, en particular, ha sido esencial para facilitar la creación de una aplicación web ligera y escalable, adecuada para manejar las complejas interacciones necesarias en este proyecto. Este capítulo proporcionará una visión exhaustiva de las tecnologías empleadas, explicando cómo cada una de ellas contribuye al éxito del proyecto y cómo se integran para ofrecer una solución innovadora y accesible para las personas con dificultades del habla.

## **4.1 TECNOLOGÍAS DE RECONOCIMIENTO Y SÍNTESIS DEL HABLA**

Las tecnologías de reconocimiento y síntesis del habla juegan un papel fundamental en el desarrollo de aplicaciones que buscan mejorar la accesibilidad y la comunicación para personas con dificultades del habla. Estas tecnologías permiten que las máquinas comprendan y generen lenguaje hablado de manera natural, facilitando la interacción humano-computadora de una manera más intuitiva y eficiente.

En el contexto del proyecto, el uso de tecnologías avanzadas de habla es crucial para crear una herramienta que no solo sea funcional, sino que también ofrezca una experiencia

de usuario adaptada y personalizada. La capacidad de reconocer el habla permite al sistema entender y procesar las entradas verbales de los usuarios, mientras que la síntesis del habla facilita la generación de respuestas auditivas claras y coherentes.

El reconocimiento del habla se basa en convertir el audio hablado en texto, lo que implica un complejo proceso de análisis de señales acústicas, modelado lingüístico y decodificación. Esta tecnología es esencial para aplicaciones como asistentes virtuales, transcripción automática y herramientas de accesibilidad, proporcionando una forma natural de interactuar con dispositivos digitales [5].

Por otro lado, la síntesis del habla se encarga de convertir texto en habla, generando voces sintéticas que pueden imitar la entonación y el ritmo del habla humana. Esta tecnología permite la creación de sistemas que pueden leer texto en voz alta, asistiendo a personas con discapacidades visuales o dificultades de lectura, y mejorando la accesibilidad general de diversas aplicaciones [6].

El uso de tecnologías avanzadas de OpenAI y Microsoft Azure en este proyecto garantiza que los usuarios se beneficien de las últimas innovaciones en inteligencia artificial y procesamiento del lenguaje natural. Estas plataformas ofrecen modelos de reconocimiento y síntesis del habla altamente precisos y personalizables, que pueden adaptarse a las necesidades específicas de cada usuario, mejorando significativamente su experiencia de comunicación.

En los siguientes apartados, se explorarán en detalle las principales tecnologías de reconocimiento y síntesis del habla disponibles, con un enfoque particular en las soluciones ofrecidas por OpenAI y Microsoft Azure, así como su funcionamiento, aplicaciones y la forma en que se integran en el proyecto.

## **4.1.1 TECNOLOGÍAS DE RECONOCIMIENTO DEL HABLA**

### **4.1.1.1 OPENAI WISPER**

**OpenAI Whisper** es una tecnología de reconocimiento del habla avanzada que utiliza modelos de aprendizaje profundo para transcribir el lenguaje hablado con alta precisión. Este sistema se destaca por su capacidad para manejar múltiples idiomas y diferentes acentos, adaptándose de manera eficaz a una variedad de contextos y usuarios. Whisper emplea redes neuronales convolucionales y modelos de atención para analizar y decodificar las señales de audio, proporcionando transcripciones precisas y rápidas [30].

### **4.1.1.2 MICROSOFT AZURE SPEECH-TO-TEXT**

Microsoft Azure ofrece una solución robusta para el reconocimiento del habla a través de su servicio Speech-to-Text. Esta tecnología utiliza modelos avanzados de procesamiento del lenguaje natural (NLP) y aprendizaje automático para convertir el habla en texto. Azure Speech-to-Text es altamente configurable y puede ser entrenado para reconocer vocabularios específicos, mejorar la precisión en contextos particulares y adaptarse a diferentes idiomas y dialectos. Su integración con otros servicios de Azure facilita el desarrollo de aplicaciones escalables y eficientes [31].

### **4.1.1.3 OTRAS TECNOLOGÍAS**

Otras tecnologías relevantes pueden ser Google Speech-to-Text o IBM Watson Speech to Text. Google Speech-to-Text es conocida por su alta precisión y capacidad para manejar una amplia variedad de idiomas y contextos. Utiliza modelos de aprendizaje profundo similares a los de OpenAI y Microsoft. IBM Watson Speech to Text ofrece una solución sólida que se integra con su suite de servicios cognitivos, proporcionando transcripciones precisas y herramientas de análisis de texto.



#### **4.1.1.4 PROCESO Y APLICACIONES**

El proceso de reconocimiento del habla implica varias etapas clave. El procesamiento de señales comienza con la captura del audio por un micrófono, que se digitaliza y segmenta en pequeñas unidades llamadas frames. Estos se analizan para extraer características acústicas relevantes. Los modelos acústicos comparan las características extraídas con patrones de sonido preestablecidos, mientras que los modelos lingüísticos ayudan a predecir la probabilidad de ciertas palabras y frases basándose en el contexto. Los algoritmos de decodificación combinan la información de los modelos acústicos y lingüísticos para generar una transcripción coherente del audio. Técnicas como la Viterbi y la búsqueda de haces (beam search) son comunes en este proceso [5].

El reconocimiento del habla tiene numerosas aplicaciones que benefician a diversos sectores. En los asistentes virtuales, tecnologías como OpenAI Whisper y Microsoft Azure Speech-to-Text se utilizan para comprender y responder a comandos de voz. Las herramientas de transcripción automática utilizan el reconocimiento del habla para convertir discursos, entrevistas y reuniones en texto escrito, facilitando la documentación y el análisis. Además, personas con discapacidades físicas o visuales se benefician de estas tecnologías, ya que les permiten interactuar con dispositivos digitales y aplicaciones mediante comandos de voz [6].

#### **4.1.2 TECNOLOGÍAS DE SÍNTESIS DEL HABLA**

La síntesis del habla es una tecnología que convierte texto escrito en habla sintetizada, imitando la entonación y el ritmo del habla humana. Esta tecnología es fundamental para mejorar la accesibilidad y proporcionar una comunicación más natural en aplicaciones que interactúan con los usuarios mediante voz. En este apartado, se explorarán las principales tecnologías de síntesis del habla, con un enfoque en OpenAI y Microsoft Azure, además de una breve mención a otras tecnologías relevantes.

#### **4.1.2.1 OPENAI TEXT TO SPEECH**

**OpenAI** ofrece una tecnología avanzada de síntesis del habla que utiliza modelos de aprendizaje profundo para generar voz natural a partir de texto. Esta tecnología se destaca por su capacidad de producir voces que suenan humanas, con entonación y prosodia realistas. OpenAI Text-to-Speech se basa en técnicas como las redes neuronales recurrentes y los modelos de atención, que permiten una conversión precisa y expresiva del texto en habla [32].

#### **4.1.2.2 MICROSOFT AZURE TEXT-TO-SPEECH**

Microsoft Azure proporciona un servicio robusto de síntesis del habla a través de su Azure Cognitive Services. Azure Text-to-Speech utiliza modelos de redes neuronales avanzados para generar voz natural en múltiples idiomas y dialectos. Esta tecnología 30 permite la personalización de las voces para adaptarse a las necesidades específicas de los usuarios, mejorando la experiencia de usuario mediante una interacción más personalizada. La integración con otros servicios de Azure facilita el desarrollo de aplicaciones complejas que requieren capacidades de síntesis del habla de alta calidad [33].

#### **4.1.2.3 OTRAS TECNOLOGÍAS**

Google Text-to-Speech es conocido por su capacidad de generar voz de alta calidad en una amplia gama de idiomas y estilos. Utiliza técnicas de aprendizaje profundo similares a las de OpenAI y Microsoft para producir voces naturales y expresivas. Amazon Polly es otra tecnología notable que convierte texto en voz, proporcionando múltiples opciones de voces y estilos que se pueden integrar fácilmente en diversas aplicaciones. IBM Watson Text to Speech también ofrece una solución sólida que se integra con su suite de servicios cognitivos, proporcionando una síntesis de voz de alta calidad para aplicaciones empresariales.

#### **4.1.2.4 PROCESO Y APLICACIONES**

El proceso de síntesis del habla implica varias etapas clave. El análisis y preprocesamiento del texto son los primeros pasos, donde el texto se convierte en una representación fonética adecuada para la síntesis. Los modelos de síntesis de voz, que pueden incluir técnicas de concatenación y modelos basados en redes neuronales, generan la señal de audio correspondiente. La prosodia y la entonación se ajustan para asegurar que el habla sintetizada suene natural y expresiva [5].

La síntesis del habla tiene diversas aplicaciones que benefician a muchos sectores. En los asistentes virtuales, tecnologías como OpenAI Text-to-Speech y Microsoft Azure Text-to-Speech se utilizan para proporcionar respuestas habladas que suenan naturales, mejorando la interacción con los usuarios. Los lectores de pantalla utilizan la síntesis del habla para ayudar a las personas con discapacidades visuales a acceder al contenido textual. Además, la generación de audio personalizado permite a las empresas crear experiencias de usuario más inmersivas y adaptadas a las necesidades individuales de sus clientes [6].

## **4.2 HERRAMIENTAS DE DESARROLLO**

En el desarrollo del proyecto, la elección de herramientas de desarrollo ha sido crucial para asegurar una gestión eficiente del código, la colaboración fluida y el mantenimiento del proyecto. Este apartado se centra en las principales herramientas utilizadas, destacando sus características y cómo han contribuido al éxito del desarrollo del sitio web.

**Visual Studio Code** y **GitHub** han sido seleccionados como las herramientas principales debido a su popularidad, robustez y la variedad de funcionalidades que ofrecen. Estas herramientas no solo facilitan la escritura y depuración del código, sino que también permiten un control de versiones eficiente y la colaboración en tiempo real, aspectos

esenciales para cualquier proyecto de software moderno. A continuación, se explorará en detalle cómo cada una de estas herramientas ha sido utilizada en el proyecto.

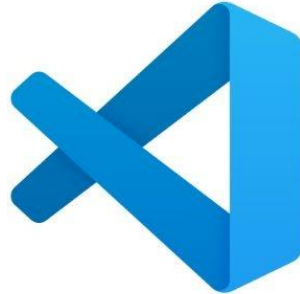
#### **4.2.1 VISUAL STUDIO CODE**

Visual Studio Code (VS Code) es un editor de código fuente desarrollado por Microsoft que se ha convertido en una herramienta esencial para el desarrollo de software gracias a su flexibilidad, extensibilidad y rendimiento. En el proyecto se comenzó utilizando Jupyter Anaconda [34], pero pronto se cambió por VS Code, el cual ha sido fundamental en la escritura y depuración del código, aportando múltiples beneficios en diversas etapas del desarrollo.

Una de las principales ventajas de VS Code es su extensibilidad. La posibilidad de añadir extensiones permite personalizar el entorno de desarrollo según las necesidades específicas del proyecto. Para este proyecto en particular, se han utilizado extensiones para lenguajes como Python, HTML, CSS y JavaScript.

La depuración integrada en VS Code es otra característica clave. Esta funcionalidad permite identificar y solucionar errores de manera eficiente, lo cual es crucial para mantener un flujo de desarrollo sin interrupciones. La capacidad de depurar código directamente en el editor ha sido especialmente útil para manejar las APIs de inteligencia artificial y la integración de diversas tecnologías.

VS Code proporciona un entorno de desarrollo unificado que soporta una amplia gama de lenguajes y herramientas. Esto permite trabajar en diferentes aspectos del proyecto dentro de un solo entorno, desde el desarrollo del backend en Python hasta el diseño del frontend con HTML y CSS, así como la gestión de bases de datos y el despliegue en la nube. Este aspecto fue un aspecto muy importante al igual que beneficioso, debido a la diversidad de lenguajes empleados en el proyecto [35].



*Figura 13 VS Code Logo*

#### 4.2.2 GITHUB

GitHub es una plataforma de alojamiento de código que utiliza el sistema de control de versiones Git. Esta plataforma ha sido fundamental en el desarrollo del proyecto, proporcionando herramientas avanzadas para la gestión del código fuente, control de versiones y seguimiento del proyecto. A continuación, se describen las principales características y beneficios de GitHub en el contexto de este proyecto.

GitHub facilita la gestión del código mediante el uso de repositorios, donde el código fuente del proyecto se almacena y organiza. Cada cambio realizado en el código se registra, permitiendo ver el historial de modificaciones, revertir a versiones anteriores si es necesario, y trabajar en diferentes ramas del proyecto sin interferencias. Esta capacidad de gestionar múltiples versiones del código es esencial para mantener la estabilidad del proyecto mientras se desarrollan nuevas funcionalidades.

Una de las características clave de GitHub es su sistema de pull requests. Aunque, en el caso de este proyecto, actúe como único desarrollador, este sistema permitió mantener un registro claro y detallado de cada modificación propuesta en el código. Cada cambio pudo ser revisado y documentado exhaustivamente, lo cual mejoró la calidad del código y evitó errores que podrían afectar el funcionamiento del proyecto.

GitHub también ha sido crucial para permitir el trabajo en distintos dispositivos y ubicaciones. Al ser una plataforma basada en la nube, GitHub permitió acceder al repositorio del proyecto desde cualquier dispositivo con conexión a Internet. Esto fue especialmente útil, ya que permitió trabajar desde diferentes lugares y dispositivos sin problemas de compatibilidad o acceso. La sincronización automática aseguró que siempre se trabajase con la versión más actualizada del código.

Además, GitHub proporcionó herramientas valiosas para que el tutor pudiera realizar un seguimiento del proyecto sin necesidad de contacto constante. Mediante el uso de GitHub Issues y el tablero de proyectos de GitHub, pude documentar tareas, errores y mejoras necesarias. Estas herramientas permitieron a mi tutor revisar el estado actual del proyecto, los avances realizados y las tareas pendientes de una manera organizada y transparente. Las notificaciones automáticas y la capacidad de comentar directamente en las issues facilitaron una comunicación eficiente y efectiva, reduciendo la necesidad de reuniones frecuentes y correos electrónicos [36].



*Figura 14 Github Logo*

## 4.3 LENGUAJES DE PROGRAMACIÓN

En el desarrollo del proyecto, la elección de los lenguajes de programación ha sido fundamental para garantizar la funcionalidad, eficiencia y escalabilidad del sistema. Los lenguajes seleccionados no solo permiten el desarrollo de una interfaz de usuario amigable y accesible, sino que también facilitan la integración con tecnologías avanzadas de inteligencia artificial y procesamiento del lenguaje natural.

En este capítulo, se describirán los lenguajes de programación utilizados en el proyecto, destacando sus características principales y el papel que desempeñan en la implementación del sitio web. Se explorarán los lenguajes Python como programa principal, HTML como programa empleado para desarrollar las ventanas de la web y CSS para el diseño de estas.

Cada uno de estos lenguajes ha sido seleccionado por sus fortalezas específicas y su capacidad para cumplir con los requisitos técnicos del proyecto. En los siguientes apartados, se detallará cómo se han implementado estos lenguajes en el desarrollo del sitio web y se explicará su contribución a la creación de una herramienta accesible y eficiente para personas con dificultades del habla.

### 4.3.1 PYTHON

Python es uno de los lenguajes de programación más populares y ampliamente utilizados en el desarrollo de software, especialmente en proyectos relacionados con la inteligencia artificial y el análisis de datos. Su sintaxis clara y su estructura intuitiva lo hacen ideal tanto para desarrolladores experimentados como para aquellos que están comenzando. En el proyecto, Python ha sido el lenguaje elegido para el desarrollo del backend, debido a varias razones.

Una de las mayores ventajas de Python es su legibilidad y simplicidad. El código en Python es fácil de escribir y leer, lo que facilita el mantenimiento y la supervisión de código.

Esta característica es especialmente útil en proyectos de investigación y desarrollo, donde la claridad del código puede acelerar el proceso de desarrollo y reducir errores.

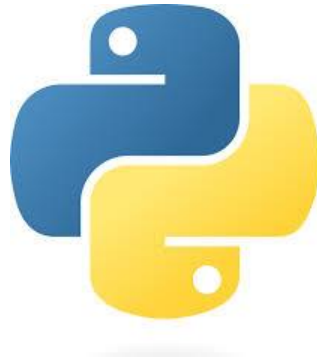
Python cuenta con una extensa colección de bibliotecas y frameworks que simplifican el desarrollo de aplicaciones complejas. Para este proyecto, se han utilizado varias bibliotecas específicas que permiten la integración con tecnologías avanzadas de inteligencia artificial como las propias de OpenAI y de Azure y el manejo eficiente de datos como la librería de SQLAlchemy. Además, la capacidad de Python para interactuar con APIs externas ha sido crucial para incorporar funcionalidades avanzadas en el sistema.

En el contexto del proyecto, Python es responsable de gestionar todos los archivos HTML. Esto implica servir las páginas web a los usuarios, manejar las solicitudes y respuestas del servidor, y garantizar que la interfaz de usuario sea dinámica y responsiva. Python permite la generación y manipulación de contenido HTML de manera eficiente, asegurando que las páginas web se actualicen en tiempo real según las interacciones del usuario.

Aunque la biblioteca Flask tiene su propio apartado en este informe, es importante mencionar que Python es el motor detrás de Flask, facilitando el desarrollo de aplicaciones web ligeras y escalables. La combinación de Python y Flask permite una integración fluida entre el backend y el frontend, garantizando que las aplicaciones web sean rápidas y fiables.

Python es particularmente efectivo en la gestión del backend de aplicaciones web. Maneja la lógica del servidor, las bases de datos y la autenticación de usuarios, entre otras funciones esenciales. Su capacidad para trabajar con bases de datos, manejar solicitudes HTTP y procesar datos de manera segura y eficiente lo convierte en una opción ideal para el desarrollo backend del proyecto [37].





*Figura 15 Python Logo*

### 4.3.2 HTML

HTML (HyperText Markup Language) es el lenguaje estándar utilizado para crear y estructurar el contenido en la web. HTML juega un papel crucial en la construcción del frontend, es decir, la parte de la aplicación que interactúa directamente con los usuarios. Este lenguaje es esencial para definir la estructura y el contenido de las páginas web, asegurando que la interfaz de usuario sea clara, accesible y fácil de navegar.

HTML permite organizar el contenido de manera lógica y jerárquica mediante el uso de elementos y etiquetas. Cada etiqueta tiene un propósito específico y contribuye a la estructura general del documento. Por ejemplo, etiquetas como `<header>`, `<nav>`, `<section>`, `<article>` o `<footer>` ayudan a dividir el contenido en secciones coherentes, facilitando tanto la navegación como la comprensión del sitio web.

La accesibilidad es un aspecto fundamental en el diseño de este proyecto, que está dirigido a personas con dificultades del habla. HTML proporciona varias herramientas y atributos que ayudan a crear un sitio web accesible. Atributos como el uso correcto de etiquetas semánticas mejoran significativamente la usabilidad del sitio para personas con discapacidades.

Una de las mayores ventajas de HTML es su compatibilidad universal con todos los navegadores web modernos. Esto garantiza que el sitio web desarrollado sea accesible para

cualquier usuario, independientemente del dispositivo o navegador que utilice. Además, HTML sigue siendo compatible con versiones anteriores, lo que ayuda a mantener la funcionalidad en sistemas más antiguos.

HTML es altamente adaptable y escalable, lo que permite que el sitio web crezca y evolucione según las necesidades del proyecto. Es fácil de mantener y actualizar, y su estructura flexible permite la integración de nuevas tecnologías y funcionalidades sin necesidad de una reestructuración completa del sitio [38].



*Figura 16 HTML Logo*

### **4.3.3 CSS**

CSS (Cascading Style Sheets) es el lenguaje utilizado para describir la presentación de un documento HTML. En este proyecto, CSS es fundamental para diseñar una interfaz de usuario atractiva y accesible. Permite controlar el layout, los colores, las fuentes y otros aspectos visuales de la página web.

CSS facilita la aplicación de estilos a los elementos HTML, asegurando una apariencia consistente y profesional en todo el sitio. Esto mejora la experiencia del usuario al hacer la navegación más intuitiva y visualmente agradable.

CSS contribuye significativamente a la accesibilidad del sitio web, permitiendo ajustar la legibilidad del texto y el contraste de colores. Además, soporta el diseño responsivo, adaptando el contenido a diferentes tamaños de pantalla y dispositivos, lo cual es crucial para usuarios con diversas necesidades.

El uso de CSS permite actualizar el diseño del sitio sin modificar el HTML subyacente, lo que facilita el mantenimiento y la evolución del proyecto. Esta separación de contenido y presentación es clave para el desarrollo ágil y eficiente [38].



*Figura 17 CSS Logo*

## **4.4 APIs**

Una API (Interfaz de Programación de Aplicaciones, por sus siglas en inglés) es un conjunto de definiciones y protocolos que permiten la comunicación entre diferentes sistemas de software. Las APIs actúan como intermediarios, permitiendo que diversas aplicaciones interactúen entre sí y compartan datos y funcionalidades sin necesidad de que el usuario intervenga directamente. Estas interfaces facilitan el desarrollo de aplicaciones complejas al permitir a los desarrolladores aprovechar servicios y funcionalidades externas, ahorrando tiempo y esfuerzo en el desarrollo de nuevas características [39].

En el contexto del proyecto, las llamadas a APIs son fundamentales para integrar capacidades avanzadas de inteligencia artificial y procesamiento del lenguaje natural. Por ejemplo, las APIs de reconocimiento y síntesis del habla permiten convertir el habla en texto y viceversa, habilitando interacciones más naturales y accesibles para los usuarios. Estas tecnologías son esenciales para desarrollar una herramienta que apoye a personas con dificultades del habla, facilitando su comunicación de manera efectiva.

Este capítulo explora las principales APIs empleadas en el proyecto, explicando cómo se utilizan para mejorar la accesibilidad y la funcionalidad del sistema. Se detallarán las APIs de reconocimiento y síntesis del habla, así como las APIs dedicadas a la gestión de la conversación con el usuario. La integración adecuada de estas APIs garantiza que el proyecto no solo sea técnicamente sólido, sino también capaz de ofrecer una experiencia de usuario intuitiva y accesible.

#### **4.4.1 APIs DE RECONOCIMIENTO Y SÍNTESIS DEL HABLA**

Las tecnologías de reconocimiento y síntesis del habla son fundamentales para permitir una interacción natural y accesible entre los usuarios y el sistema. Estas tecnologías hacen posible que los usuarios con dificultades del habla puedan utilizar el sistema de manera eficaz, facilitando tanto la conversión de voz en texto como de texto en voz. Para implementar estas capacidades, se han integrado dos APIs principales: Microsoft Azure Text-to-Speech y OpenAI Whisper, cada una proporcionando soluciones avanzadas para la síntesis y el reconocimiento del habla, respectivamente.

##### **4.4.1.1 MICROSOFT AZURE TEXT-TO-SPEECH**

Microsoft Azure Text-to-Speech se ha implementado en el proyecto para convertir texto en voz, ofreciendo a los usuarios la posibilidad de escuchar las respuestas generadas por el sistema de manera clara y natural. Esta API es ampliamente reconocida por su

capacidad de producir voces sintéticas que suenan naturales, lo que es esencial para una experiencia de usuario inclusiva y agradable.

Azure Text-to-Speech proporciona una variedad de opciones para personalizar la salida de voz, incluyendo la selección de voces en múltiples idiomas y la posibilidad de ajustar parámetros como la velocidad, el tono y la pronunciación. Esta flexibilidad permite adaptar la síntesis de voz a las necesidades y preferencias específicas de cada usuario, mejorando así la accesibilidad del sistema.

La integración de esta API en el proyecto fue facilitada por las herramientas y SDKs proporcionados por Microsoft, lo que permitió una implementación rápida y eficiente. Además, Azure Text-to-Speech es altamente escalable, lo que garantiza un rendimiento óptimo incluso cuando se manejan múltiples solicitudes simultáneas. Esto es particularmente importante en entornos donde varios usuarios interactúan con el sistema al mismo tiempo.

En el contexto del proyecto, la API de Microsoft Azure Text-to-Speech se utiliza principalmente para leer en voz alta las narrativas generadas por el sistema, permitiendo a los usuarios escuchar sus historias o ideas de una manera que les resulte familiar y cómoda. Esta funcionalidad es clave para garantizar que el sistema no solo sea accesible, sino también efectivo en su objetivo de asistir a personas con dificultades del habla.

#### **4.4.2 APIs DE ASISTENCIA DE CONVERSACIÓN**

El asistente de conversación es un componente crucial en el proyecto. Este sistema, basado en tecnologías avanzadas de inteligencia artificial, tiene la capacidad de interactuar con los usuarios de manera natural y eficiente, facilitando la comunicación y proporcionando respuestas coherentes y contextualmente adecuadas. Utilizando APIs de IA, el asistente puede comprender y procesar el lenguaje natural, adaptándose a las necesidades específicas de cada usuario y mejorando su experiencia de uso.

La implementación del asistente de conversación en este proyecto no solo implica la capacidad de comprender y generar texto, sino también la integración con otros sistemas de procesamiento del lenguaje natural y síntesis de voz. Esto permite que el asistente no solo interprete las entradas del usuario, sino que también convierta las respuestas en audio, ofreciendo una solución completa y accesible para personas con dificultades del habla.

En el siguiente apartado, se explorará en detalle cómo se ha utilizado la API de OpenAI para desarrollar y optimizar el asistente de conversación, destacando las funcionalidades clave.

#### **4.4.2.1 OPENAI – CHATGPT**

OpenAI proporciona una de las APIs más avanzadas para desarrollar asistentes de conversación basados en inteligencia artificial. Esta API permite la creación de modelos de lenguaje capaces de comprender y generar texto de manera coherente y contextualmente relevante. En el proyecto, se ha utilizado la API de OpenAI para construir el asistente de conversación que guía a los usuarios a través de interacciones naturales y significativas.

La API de OpenAI estructura las conversaciones en torno a tres roles principales: "system", "user" y "assistant". Cada uno de estos roles tiene una función específica que contribuye a la dinámica de la interacción y asegura que las respuestas generadas por el asistente sean relevantes y útiles.

- **System**

El rol "system" es fundamental para establecer el contexto y las reglas de la interacción. Al principio de una conversación, este rol se utiliza para proporcionar instrucciones y configuraciones que guían el comportamiento del asistente a lo largo de la interacción. Por ejemplo, el mensaje del sistema puede especificar el tono que debe adoptar el asistente, los temas que debe evitar o cualquier otra directriz que afecte cómo debe

responder. Esta configuración asegura que el asistente mantenga un enfoque consistente y apropiado durante toda la conversación.

- **User**

El rol "user" representa las entradas del usuario, es decir, las preguntas, comandos o cualquier otro tipo de información que el usuario proporciona al asistente. Estas entradas son cruciales ya que determinan la dirección y el contenido de la conversación. El asistente utiliza estas entradas para generar respuestas adecuadas y útiles. Las 38 interacciones del usuario guían al asistente para proporcionar respuestas que ayuden al usuario a estructurar su narrativa de manera efectiva.

- **Assistant**

El rol "assistant" corresponde a las respuestas generadas por el modelo de lenguaje, basadas en el contexto proporcionado por los roles "system" y "user". Este rol es responsable de producir el contenido que el usuario verá y con el que interactuará. Las respuestas del asistente deben ser coherentes, contextualmente adecuadas y alineadas con las instrucciones del sistema. Utilizando la entrada del usuario y las directrices del sistema, el asistente puede proporcionar orientación clara y útil.

El **Playground** es una herramienta interactiva proporcionada por OpenAI que permite a los desarrolladores experimentar con diferentes configuraciones y modelos de lenguaje en tiempo real. Esta interfaz facilita la prueba de prompts y ajustes de parámetros del modelo, permitiendo entender mejor cómo responderá el asistente en diversas situaciones. Los desarrolladores pueden utilizar el Playground para iterar rápidamente sobre diferentes configuraciones y encontrar la combinación óptima que ofrezca la mejor experiencia de usuario.

El uso de la librería de Python "OpenAI" ha sido fundamental para integrar la API en el proyecto. Esta biblioteca permite enviar solicitudes a la API de OpenAI y gestionar las interacciones entre el usuario y el asistente de manera eficiente. A través de esta integración, se pueden procesar las respuestas para proporcionar una experiencia de conversación fluida

y natural, adaptada a las necesidades específicas de los usuarios con dificultades del habla [40].



*Figura 18 OpenAI Logo*

## 4.5 LIBRERÍAS Y FRAMEWORKS UTILIZADOS

En el desarrollo, la elección de librerías y frameworks ha sido crucial para asegurar una implementación eficiente y escalable. Estas herramientas proporcionan funcionalidades predefinidas y estructuras sólidas que simplifican el proceso de desarrollo, permitiendo centrarse en la lógica específica del proyecto y en la integración de tecnologías avanzadas. Las librerías y frameworks seleccionados, como Flask, SQLAlchemy o Bootstrap, además de otras las cuales han tenido menos importancia o las librerías propias de OpenAI y Azure para la llamada a APIs respectivamente, han facilitado la creación de una aplicación web robusta, con un backend potente y un frontend accesible y atractivo. A continuación, se detallarán las principales librerías y frameworks utilizados, destacando sus características y cómo han contribuido al éxito del proyecto.

### 4.5.1 FLASK

**Flask** es un microframework para Python que permite desarrollar aplicaciones web de forma rápida y sencilla. Ha sido elegido para este proyecto por su flexibilidad y su capacidad para integrarse fácilmente con otras tecnologías y bibliotecas. Flask proporciona un entorno liviano que se adapta perfectamente a las necesidades de desarrollo de



aplicaciones web dinámicas, permitiendo una configuración mínima, pero con la opción de ampliar sus funcionalidades mediante extensiones.

Una de las características principales de Flask es su diseño modular, lo que permite a los desarrolladores construir aplicaciones utilizando una estructura de componentes reutilizables. Esto facilita el mantenimiento del código y la escalabilidad de la aplicación a medida que se agregan nuevas características. Además, su simplicidad de uso no compromete el poder de personalización, lo que hace que Flask sea una herramienta ideal tanto para proyectos pequeños como para aquellos que requieren un enfoque más detallado y controlado en su desarrollo.

En este proyecto, Flask ha sido esencial para manejar las solicitudes HTTP, gestionar rutas y renderizar plantillas, lo que nos permite crear una experiencia de usuario fluida y eficiente. Su compatibilidad con herramientas de desarrollo populares como SQLAlchemy para la gestión de bases de datos y Jinja2 para el templating ha facilitado la construcción de una aplicación robusta y escalable [41].



*Figura 19 Flask Logo*

#### 4.5.2 SQLALCHEMY

**SQLAlchemy** es una biblioteca de Python que facilita el trabajo con bases de datos mediante el uso de un enfoque de mapeo objeto-relacional (ORM). Esta herramienta permite a los desarrolladores interactuar con las bases de datos utilizando objetos Python, lo que

simplifica significativamente la manipulación de datos y el diseño de la base de datos dentro de una aplicación.

En este proyecto, SQLAlchemy ha sido implementado para gestionar la persistencia de datos, asegurando que todas las transacciones con la base de datos sean seguras y eficientes. Su capacidad para trabajar con múltiples sistemas de gestión de bases de datos proporciona flexibilidad, permitiendo cambios en el backend sin necesidad de modificar la lógica de la aplicación.

SQLAlchemy ofrece una capa de abstracción sobre SQL, permitiendo construir consultas complejas utilizando un lenguaje de consultas basado en Python. Esto no solo mejora la legibilidad del código, sino que también reduce la posibilidad de errores comunes asociados con la escritura manual de consultas SQL. Además, SQLAlchemy facilita la definición de esquemas de bases de datos mediante el uso de modelos, lo que permite un diseño más intuitivo y un mantenimiento más sencillo del esquema de datos.

Gracias a SQLAlchemy, la aplicación ha logrado una gestión eficiente de los datos, permitiendo operaciones como la creación, lectura, actualización y eliminación (CRUD) de forma optimizada. Su integración en el proyecto ha sido clave para asegurar la integridad de los datos y mejorar el rendimiento general de la aplicación [42].



*Figura 20 SQLAlchemy Logo*

### **4.5.3 BOOTSTRAP**

Bootstrap es uno de los frameworks de front-end más populares y ampliamente utilizados para el desarrollo de interfaces web responsivas y modernas. Este framework

facilita la creación de sitios web atractivos y funcionales con una estructura de código limpia y reutilizable. Bootstrap ha sido una herramienta esencial para diseñar y desarrollar la base que sigue la interfaz de usuario del sitio web.

Bootstrap proporciona un sistema de grid (rejilla) flexible y responsivo que permite diseñar layouts de página adaptables a diferentes tamaños de pantalla. Esto es crucial para asegurar que la página web sea accesible y fácil de usar en dispositivos móviles, tabletas y computadoras de escritorio.

El framework incluye una amplia variedad de componentes pre-diseñados, como botones, formularios, tablas, alertas, modales y mucho más. Estos componentes facilitan la implementación rápida de elementos interactivos y visualmente coherentes en la página web. El uso de estos componentes asegura una apariencia consistente y profesional en toda la aplicación web, mejorando la experiencia del usuario y haciendo que la navegación sea intuitiva y agradable.

Bootstrap también permite una gran personalización. Aunque proporciona un estilo predeterminado, es posible ajustar los estilos a través de hojas de estilo CSS personalizadas o utilizando las variables Sass de Bootstrap para adaptar el diseño a las necesidades específicas del proyecto. Esta capacidad de personalización ha sido fundamental para adaptar la apariencia de la web a la identidad del proyecto.

La integración de Bootstrap en este proyecto ha permitido crear una interfaz de usuario accesible, responsiva y fácil de usar, alineada con los objetivos. La utilización de Bootstrap ha simplificado el desarrollo del front-end, permitiendo centrarse en la funcionalidad y la experiencia del usuario sin sacrificar la calidad del diseño [43].



*Figura 21 Bootstrap Logo*

## **4.5.4 OTRAS LIBRERÍAS Y SUB-LIBRERÍAS**

### **4.5.4.1 WERKZEUG.SECURITY**

Werkzeug.security es una sub-librería de Werkzeug, una biblioteca integral para WSGI (Web Server Gateway Interface) que forma la base de Flask, el marco de trabajo web principal utilizado en este proyecto. `werkzeug.security` proporciona herramientas cruciales para garantizar la seguridad de la aplicación, especialmente en lo que respecta a la gestión de contraseñas. Las funciones principales, `generate_password_hash` y `check_password_hash`, se utilizan para crear hashes seguros de contraseñas y para verificar la autenticidad de las contraseñas proporcionadas por los usuarios, respectivamente. Esto garantiza que las contraseñas nunca se almacenen en texto plano, mitigando el riesgo de compromisos de seguridad.

### **4.5.4.2 AZURE.COGNITIVESERVICES.SPEECH**

`Azure.cognitiveservices.speech` es una librería proporcionada por Microsoft Azure que integra capacidades avanzadas de reconocimiento y síntesis de voz en la aplicación. Esta librería es fundamental para convertir texto en voz y viceversa, lo cual es especialmente importante para mejorar la accesibilidad del proyecto, permitiendo que los usuarios con dificultades del habla interactúen con la aplicación de manera más natural. El uso de `azure.cognitiveservices.speech` permite que las respuestas generadas por la aplicación sean presentadas en formato de audio, mejorando significativamente la experiencia del usuario.

### **4.5.4.3 OPENAI**

La librería `openai` es el puente que permite la comunicación con los modelos de lenguaje desarrollados por OpenAI, como GPT-3. En este proyecto, `openai` se utiliza para generar respuestas de texto inteligentes y contextualmente relevantes basadas en las entradas del usuario. Esta capacidad de generar contenido textual dinámico y coherente es una de las

pedras angulares del proyecto, ya que facilita que los usuarios estructuren y expresen sus ideas de manera efectiva. La flexibilidad de openai permite personalizar las interacciones, ajustando las respuestas a las necesidades específicas de cada usuario, lo que mejora la eficacia de la aplicación en su objetivo de ayudar a personas con dificultades del habla.

#### **4.5.4.4 BLUEPRINTS**

En el contexto de Flask, los Blueprints son una poderosa herramienta para organizar el código de una aplicación en módulos reutilizables y bien estructurados. Un Blueprint en Flask permite definir rutas, controladores y otros aspectos de la aplicación en módulos separados, lo que facilita la gestión y escalabilidad del proyecto. En este proyecto, se utilizan Blueprints para dividir la aplicación en componentes más pequeños y manejables, cada uno enfocado en un conjunto específico de funcionalidades, como la gestión de usuarios, la integración con APIs externas, o la generación de contenido. Esto no solo mejora la organización del código, sino que también permite un desarrollo más modular y colaborativo, donde diferentes partes del proyecto pueden ser desarrolladas y mantenidas de manera independiente.

#### **4.5.4.5 FLASK-LOGIN**

Flask-Login es una extensión de Flask diseñada para gestionar la autenticación y las sesiones de usuario en aplicaciones web. Esta librería ha sido seleccionada para el proyecto debido a su integración fluida con Flask y sus herramientas para manejar sesiones de usuario de manera segura. Flask-Login permite autenticar a los usuarios, mantener su estado a través de las sesiones y restringir el acceso a rutas específicas utilizando decoradores como `@login_required`. Además, el objeto `current_user` proporciona un acceso sencillo a la información del usuario autenticado, lo que facilita la personalización de la interfaz y el contenido según el perfil del usuario.

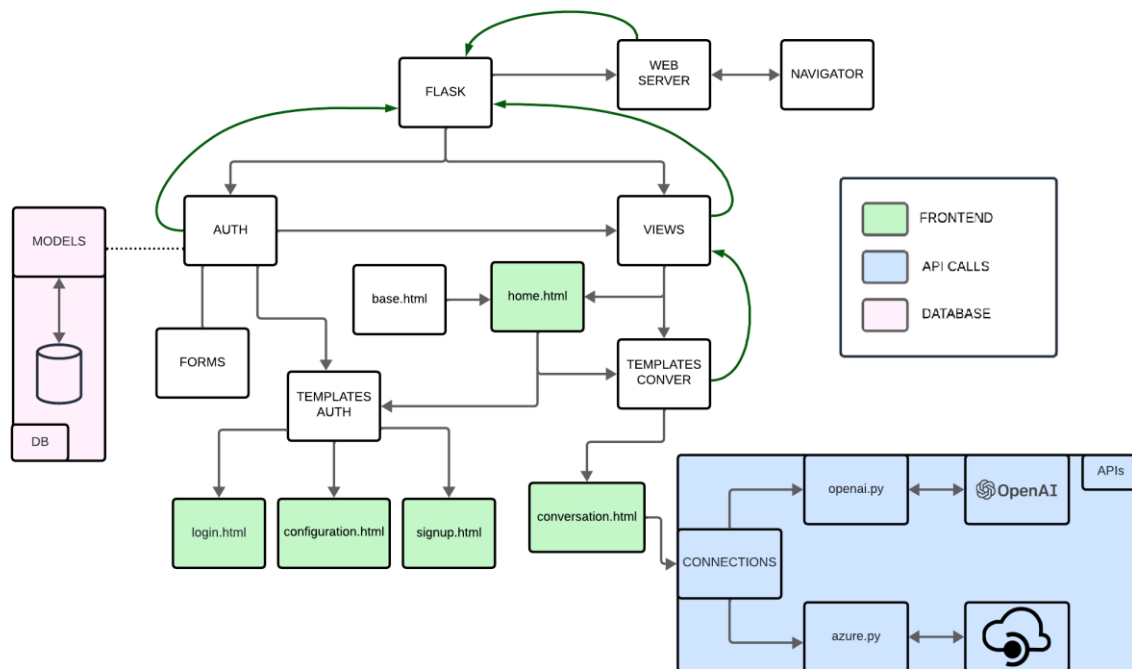
La implementación de Flask-Login asegura que cada usuario solo tenga acceso a los recursos pertinentes, lo que no solo mejora la seguridad, sino también la usabilidad de la aplicación. Se han implementado prácticas de seguridad adicionales, como la regeneración de tokens de sesión, para prevenir ataques de secuestro de sesiones, lo que refuerza aún más la seguridad del sistema.

## **Capítulo 5. SISTEMA DESARROLLADO**

### **5.1 DISEÑO DEL SISTEMA**

#### **5.1.1 ARQUITECTURA GENERAL**

La arquitectura de la aplicación ha sido diseñada para optimizar la interacción entre el usuario y las tecnologías de inteligencia artificial, garantizando una experiencia accesible y eficiente. Como se puede ver en la figura 5.1, el sistema se estructura en varias capas que abarcan el frontend, el backend, la base de datos, y la integración con servicios de terceros (APIs), logrando una solución completa y cohesiva.



*Figura 22. Arquitectura del Sistema*

- **Backend**

El backend se implementa principalmente utilizando el framework **Flask**, que ofrece un entorno ligero y flexible para el desarrollo de aplicaciones web. Aunque se ha mencionado previamente, más adelante se detallará su implementación específica en este proyecto. Los principales componentes del backend son:

**Controladores (Views):** Son responsables de manejar las solicitudes HTTP del navegador, procesar la lógica de la aplicación, y devolver las respuestas adecuadas. Estos controladores son el núcleo de la interacción entre el frontend y el backend, asegurando que cada acción del usuario se procese correctamente.

**Autenticación (Auth):** Gestiona la seguridad del sistema, incluyendo el inicio de sesión, el registro de usuarios y el mantenimiento de sesiones activas. Este componente garantiza que solo los usuarios autorizados puedan acceder a ciertas funcionalidades, protegiendo así los datos personales y la privacidad del usuario.



**Conexiones a APIs (Connections):** Encargadas de manejar las llamadas y las respuestas de las APIs de terceros, que son fundamentales para el procesamiento de lenguaje natural y síntesis de voz. Este componente permite integrar y aprovechar tecnologías avanzadas de inteligencia artificial, facilitando una comunicación más natural y eficiente con los usuarios.

**Base de Datos (DB):** Utiliza una base de datos SQL para almacenar información relevante, como datos de usuario y registros de conversación. La interacción con la base de datos se realiza a través de **SQLAlchemy**, un ORM (Object-Relational Mapping) que facilita la manipulación de datos mediante objetos de Python, llamados modelos (Models).

- **Frontend**

Para el desarrollo del frontend de la aplicación, se ha utilizado principalmente HTML y CSS, que forman la base estructural y estilística de las diferentes páginas y componentes de la interfaz. HTML se encarga de la estructura básica de las páginas web, permitiendo una organización clara de los elementos de contenido, mientras que CSS se utiliza para definir el diseño visual, incluyendo colores, tipografía y disposición de los elementos, asegurando una experiencia de usuario coherente y atractiva.

El enfoque modular del desarrollo frontend se ha logrado mediante el uso de un archivo de plantilla base llamado 'base.html'. Este archivo actúa como la plantilla maestra de la cual heredan otras páginas, lo que asegura consistencia en elementos comunes como la barra de navegación, el pie de página y el diseño general de la página. Esta metodología no solo facilita el mantenimiento del código, sino que también simplifica la implementación de cambios globales, mejorando la eficiencia del desarrollo.

La combinación de HTML y CSS, junto con el uso estratégico de plantillas y el framework Bootstrap, permite crear una interfaz de usuario accesible y responsiva, que se adapta a diferentes dispositivos y tamaños de pantalla, ofreciendo a los usuarios una experiencia fluida y uniforme en toda la aplicación.

### **5.1.1.1 FLUJO DE TRABAJO**

El flujo de trabajo de la aplicación está diseñado para proporcionar una experiencia de usuario fluida y eficiente, facilitando la interacción con el sistema mediante el uso de tecnologías avanzadas de inteligencia artificial. A continuación, se describen los pasos principales del flujo de trabajo desde el inicio de sesión hasta la generación de la historia de audio.

#### **1. Inicio de Sesión**

El usuario inicia sesión en la aplicación utilizando sus credenciales. Este proceso se gestiona a través del módulo de autenticación del backend, que verifica la identidad del usuario y otorga acceso a las funcionalidades de la aplicación.

#### **2. Interacción con el Asistente Conversacional**

Una vez autenticado, el usuario interactúa con el asistente conversacional, el cual está integrado con la API de OpenAI. Esta interacción se lleva a cabo a través de una interfaz de chat diseñada para guiar al usuario en la recopilación de información necesaria para generar una historia personalizada.

#### **3. Recopilación de Información**

Durante la conversación, el asistente solicita al usuario detalles específicos que se utilizarán para crear la historia. Estos detalles se almacenan temporalmente en la base de datos para su posterior procesamiento.

#### **4. Procesamiento de Datos y Generación de Historia**

Una vez recopilada toda la información necesaria, el backend procesa los datos y utiliza las APIs de Azure para la síntesis de voz. Este procesamiento convierte la historia generada en texto a un archivo de audio que refleja la voz del usuario.

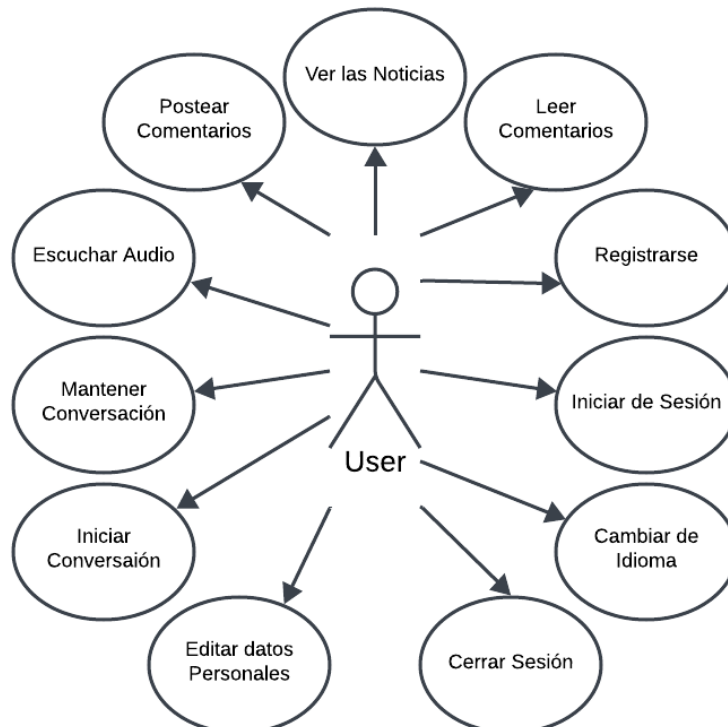
## 5. Reproducción y Almacenamiento de la Historia

El usuario puede escuchar la historia generada directamente desde la aplicación. Este archivo de audio se guarda internamente en el programa, el cual se va sobrescribiendo. En un futuro la idea sería, poder guardar estos audios para poder reproducirlos de nuevo.

## 6. Cierre de Sesión y Seguridad

Finalmente, el usuario puede cerrar sesión de manera segura, asegurándose de que su información personal y las historias generadas están protegidas conforme a las políticas de seguridad del sistema.

### 5.1.2 CASOS DE USO



*Figura 23 Casos de Uso*

- **Registrarse:** Para comenzar a utilizar la aplicación, el usuario debe registrarse proporcionando un correo electrónico, un nombre y una contraseña. Este proceso asegura que cada usuario tenga su propia cuenta personalizada.
- **Iniciar sesión:** Si el usuario ya dispone de una cuenta, puede iniciar sesión utilizando su correo electrónico y contraseña. Esto le permitirá acceder a todas las funcionalidades de la aplicación.
- **Cambiar idioma:** Una vez creada la cuenta, el usuario tiene la opción de modificar el idioma de la aplicación. De forma predeterminada, el idioma es inglés, pero puede ser cambiado a cualquiera de las otras opciones disponibles.
- **Editar datos personales:** Los usuarios tienen la capacidad de actualizar su información personal en cualquier momento. Esto incluye la modificación del correo electrónico, nombre y contraseña, así como el ajuste del idioma preferido.
- **Cerrar sesión:** Para mantener la seguridad de su cuenta, el usuario puede cerrar sesión una vez que haya terminado de utilizar la aplicación.
- **Ver las noticias:** En la página de inicio, los usuarios pueden acceder a una serie de noticias relacionadas con el mundo de los trastornos del habla, manteniéndose informados sobre las últimas novedades y avances.
- **Iniciar conversación:** El usuario puede comenzar una conversación guiada que le permitirá proporcionar la información necesaria para generar un audio que exprese lo que desea transmitir.
- **Mantener conversación:** Durante la conversación, el usuario sigue respondiendo preguntas y proporcionando datos que serán utilizados para generar una narrativa coherente y completa.
- **Escuchar audio:** Al finalizar la conversación, se genera un archivo de audio que reproduce todo lo que el usuario ha transmitido, utilizando una voz sintetizada basada en la información proporcionada.
- **Postear comentarios:** Los usuarios pueden compartir sus opiniones o comentarios en el foro de la aplicación, fomentando la interacción y el intercambio de ideas dentro de la comunidad.

- **Leer comentarios:** Además de publicar sus propios comentarios, los usuarios pueden leer las opiniones y experiencias compartidas por otros usuarios, enriqueciendo su experiencia en la aplicación.

### 5.1.3 REQUISITOS FUNCIONALES Y NO FUNCIONALES

#### 5.1.3.1 FUNCIONALES

- **Autenticación de usuarios:** El sistema debe permitir que los usuarios se registren, inicien sesión y gestionen sus credenciales de manera segura.
- **Gestión de perfiles de usuario:** Los usuarios deben poder actualizar su información personal, incluyendo nombre, correo electrónico y preferencias de idioma.
- **Interacción con el asistente conversacional:** El sistema debe facilitar la comunicación entre el usuario y el asistente de IA, permitiendo la recopilación de información de manera guiada.
- **Generación de historias en formato de audio:** El sistema debe convertir las historias generadas en texto a un archivo de audio utilizando tecnologías de síntesis de voz.
- **Almacenamiento temporal y seguro de datos:** Los datos relacionados con las conversaciones y las historias generadas deben almacenarse de manera temporal y con medidas de seguridad para proteger la privacidad del usuario.
- **Soporte multilingüe:** El sistema debe ofrecer soporte para varios idiomas, permitiendo que los usuarios seleccionen su idioma preferido durante la interacción.

#### 5.1.3.2 NO FUNCIONALES

- **Seguridad y protección de datos:** El sistema debe implementar medidas de seguridad robustas para proteger la información personal y sensible de los usuarios, garantizando la confidencialidad, integridad y disponibilidad de los datos.

- **Escalabilidad del sistema:** El sistema debe estar diseñado de manera que pueda manejar un incremento en la carga de trabajo, tanto en términos de número de usuarios como de volumen de datos, sin degradar su rendimiento.
- **Rendimiento y tiempo de respuesta:** El sistema debe ser capaz de procesar solicitudes y devolver respuestas en un tiempo adecuado para garantizar una experiencia de usuario fluida y sin retrasos perceptibles.
- **Accesibilidad web:** La interfaz de usuario del sistema debe ser accesible para personas con diferentes discapacidades, cumpliendo con las normativas de accesibilidad web como las WCAG (Web Content Accessibility Guidelines).
- **Mantenibilidad y modularidad del código:** El sistema debe estar diseñado de manera modular para facilitar el mantenimiento, la corrección de errores y la implementación de futuras mejoras sin afectar otras partes del sistema.
- **Compatibilidad con múltiples dispositivos y navegadores:** El sistema debe funcionar de manera consistente en diferentes dispositivos (como computadoras, tablets y smartphones) y navegadores web (como Chrome, Firefox, Safari), asegurando una experiencia uniforme para todos los usuarios.

## 5.2 DESARROLLO BACKEND

El desarrollo backend del proyecto se centra en construir una infraestructura robusta y eficiente que sustente todas las funcionalidades críticas de la aplicación web. Esta capa del sistema es responsable de manejar la lógica del servidor, las interacciones con las bases de datos y la integración con diversas APIs de IA.

El backend se ha implementado principalmente utilizando Python, un lenguaje de programación conocido por su versatilidad y amplia adopción en el desarrollo de aplicaciones web y de inteligencia artificial. Python, junto con el framework Flask, proporciona un entorno ligero y flexible para desarrollar y desplegar aplicaciones web rápidas y seguras.

En esta sección, se detallarán los componentes principales del backend, incluyendo la estructura del servidor, las funcionalidades implementadas, y la manera en que se ha integrado con APIs externas. También se abordará la gestión de datos del usuario, asegurando que la información se maneje de manera segura y eficiente conforme a las normativas de protección de datos.

El desarrollo backend no solo busca cumplir con los requisitos funcionales del proyecto, sino también garantizar una arquitectura escalable que permita futuras expansiones y mejoras. La elección de herramientas y tecnologías modernas asegura que la aplicación pueda adaptarse rápidamente a nuevas demandas y continuar ofreciendo una experiencia de usuario óptima.

### **5.2.1 ESTRUCTURA**

La estructura del backend se ha diseñado para ser modular y escalable, facilitando el mantenimiento y la expansión futura del sistema. La arquitectura sigue el patrón de diseño de microservicios, donde cada componente maneja una parte específica de la lógica del negocio, asegurando así una separación clara de responsabilidades y una mejor organización del código. Los componentes principales son:

#### **1. Servidor Web (Flask)**

El núcleo del backend está construido con Flask, un microframework de Python que permite desarrollar aplicaciones web de manera sencilla y eficiente. Flask se encarga de manejar las solicitudes HTTP, enrutando las peticiones a las funciones correspondientes y gestionando las respuestas hacia el cliente.

#### **2. Gestión de Autenticación**

Un módulo dedicado gestiona el registro e inicio de sesión de los usuarios, implementando protocolos de autenticación seguros. Se utilizan tokens JWT (JSON Web Tokens) para mantener sesiones activas de manera segura y eficiente.

### 3. Interacción con APIs Externas

La aplicación se integra con diversas APIs de inteligencia artificial, como las ofrecidas por OpenAI y Azure. Estos servicios son cruciales para el procesamiento de lenguaje natural y la síntesis de voz, permitiendo personalizar las interacciones con los usuarios y generar contenido de audio de alta calidad.

### 4. Base de Datos (SQLAlchemy)

El backend utiliza SQLAlchemy como ORM (Object-Relational Mapping) para interactuar con la base de datos. Esta herramienta facilita la creación, modificación y consulta de los datos, proporcionando una capa de abstracción que simplifica el manejo de las operaciones SQL.

## 5.2.2 FUNCIONALIDADES IMPLEMENTADAS

### 5.2.2.1 AUTENTICACIÓN Y GESTIÓN DE USUARIOS

La autenticación y la gestión de usuarios son componentes esenciales del desarrollo de la aplicación. Estos procesos aseguran que el acceso a las funcionalidades del sistema este restringido a usuarios autorizados, protegiendo así la información personal y garantizando una experiencia personalizada. El proceso que se emplea para la autenticación es el siguiente:

- **Registro de usuario**

Durante el registro, los nuevos usuarios proporcionan sus datos personales, como nombre, dirección de correo electrónico y contraseña. Para proteger las contraseñas de los usuarios, se utiliza un enfoque de **encriptación por hash**. Específicamente, se emplea el algoritmo de hashing bcrypt, conocido por su seguridad y capacidad de incorporar sal (salt) para prevenir ataques de diccionario y de fuerza bruta.



Antes de crear un nuevo usuario, el sistema valida la información ingresada para asegurarse de que cumple con los criterios de seguridad establecidos, como longitud mínima de la contraseña y formato del correo electrónico.

Al registrarse, la contraseña del usuario se transforma en un hash utilizando bcrypt. Este proceso implica generar una cadena aleatoria llamada "salt" que se combina con la contraseña antes de aplicar el algoritmo de hashing. El resultado es una cadena única que se almacena en la base de datos, asegurando que incluso si la base de datos fuera comprometida, las contraseñas no serían fácilmente descifrables.

```
from werkzeug.security import generate_password_hash
user.password = generate_password_hash(password, method='pbkdf2:sha256')
```

En este fragmento de código, `generate_password_hash` es la función utilizada para crear el hash de la contraseña del usuario. El parámetro 'method' especifica el método de hashing a utilizar, en este caso, `pbkdf2:sha256`, que es un algoritmo robusto y seguro. Esta función genera un hash de la contraseña combinada con un "salt" generado automáticamente. El hash resultante se almacena en la base de datos en lugar de la contraseña original.

- **Inicio de sesión**

Para el inicio de sesión, los usuarios deben ingresar sus credenciales (correo electrónico y contraseña) que se comparan con los registros almacenados.

El sistema recupera el hash de la contraseña asociado con el correo electrónico proporcionado y aplica el mismo proceso de hashing a la contraseña ingresada. Si el hash resultante coincide con el almacenado, el acceso es concedido.

```
from werkzeug.security import check_password_hash
check_password_hash(user.password, password)
```

Para verificar la contraseña durante el inicio de sesión, se utiliza la función `check_password_hash`, que compara la contraseña ingresada por el usuario con el hash

almacenado en la base de datos. Esta función, como argumentos, recibe el hash almacenado (user.password) y la contraseña ingresada. Si la contraseña es correcta, la función retorna True; de lo contrario, retorna False. Este enfoque garantiza que las contraseñas nunca se almacenen ni se transmitan en texto plano, mejorando así significativamente la seguridad de la aplicación.

Una vez autenticado, se inicia una sesión segura para el usuario. Se utilizan tokens de sesión para mantener la autenticación activa y permitir que el usuario acceda a las funcionalidades de la aplicación sin tener que volver a iniciar sesión repetidamente.

Los usuarios pueden cerrar su sesión de manera segura, lo que invalida el token de sesión y protege su cuenta de accesos no autorizados.

Posteriormente la gestión de los usuarios incluye funcionalidades de actualización de varios elementos, como el correo electrónico, la contraseña, el nombre de usuario o el idioma, el cual no se establece al crear el usuario, únicamente modificándolo.

Este sistema de autenticación y gestión de usuarios no solo garantiza la seguridad de la información personal, sino que también ofrece una experiencia de usuario robusta y confiable.

### **5.2.2.2 INTERACCIÓN CON LAS APIS**

Como se ha mencionado en repetidas ocasiones, la integración de APIs de inteligencia artificial es fundamental para el desarrollo y funcionamiento del proyecto, permitiendo una interacción fluida y efectiva con el usuario. Las APIs utilizadas no solo enriquecen la experiencia del usuario, sino que también son esenciales para la implementación de las funcionalidades básicas de la aplicación, como la conversación guiada y la síntesis o reconocimiento de voz.

- **OpenAI**

En primer lugar, la API de OpenAI es esencial para manejar las conversaciones dentro de la aplicación. Cada mensaje enviado y recibido constituye una llamada a la API, lo que permite mantener una conversación dinámica y coherente. Para gestionar estas conversaciones, los datos se estructuran y envían en formato JSON. Este enfoque permite almacenar el contexto de la conversación, incluyendo mensajes previos, en un diccionario interno que facilita la manipulación y actualización de los datos. Esto es crucial para asegurar que el modelo de IA comprenda el contexto completo de la conversación y pueda generar respuestas relevantes y coherentes. Además, se utilizan variables específicas para almacenar las últimas preguntas y respuestas, lo que facilita tanto la visualización del chat en la interfaz del usuario como el seguimiento del flujo de la conversación. Esto garantiza que el usuario pueda ver toda la interacción de manera clara y ordenada, y no suponga un problema seguirla y responder lo necesario.

- **Azure**

Por otro lado, la API de Azure se emplea para la conversión de texto a voz, permitiendo que el texto generado se convierta en audio que puede ser reproducido al usuario. El proceso de integración con Azure es relativamente sencillo. Se desarrolla una función que recibe un texto, generalmente el resultado de una conversación generada por la API de OpenAI, y devuelve un archivo de audio. Este archivo se reproduce automáticamente al finalizar la conversación, ofreciendo al usuario una experiencia completa de comunicación. Esta funcionalidad no solo mejora la accesibilidad para usuarios con dificultades del habla, sino que también proporciona una experiencia más inmersiva.

Estas llamadas a las APIs se gestionan mediante archivos de lenguaje Python específicos, donde se desarrollan las funciones necesarias para interactuar con cada servicio. Estos archivos son utilizados posteriormente por otros componentes del sistema que administran los aspectos visuales de la aplicación, generando así una comunicación efectiva entre el backend y el frontend. Esta arquitectura modular permite mantener una separación

clara de responsabilidades, facilitando tanto el mantenimiento del código como la escalabilidad del sistema para futuras ampliaciones y mejoras.

### **5.2.2.3 PROCESAMIENTO Y ALMACENAMIENTO DE DATOS**

El procesamiento y almacenamiento de datos en este proyecto se enfoca en la gestión eficiente y segura de la información generada por cada usuario, con especial atención a los comentarios y la participación en el foro de la aplicación. Cada vez que un usuario interactúa en el foro, ya sea publicando un comentario, respondiendo a otro usuario o actualizando su perfil, la información es procesada y almacenada para garantizar su integridad, disponibilidad y fácil acceso.

La base de datos SQL, gestionada mediante SQLAlchemy, es la encargada de almacenar todos los comentarios y la información relacionada con cada usuario, incluyendo su historial de publicaciones y actividades en el foro. Este sistema de almacenamiento está diseñado para permitir una recuperación rápida y eficiente de los datos, facilitando que los usuarios puedan acceder y revisar su historial de actividad en cualquier momento.

Cada comentario está vinculado de manera única al perfil del usuario que lo creó, lo que permite a los usuarios gestionar su contenido de forma personalizada. Además de poder ver y administrar sus propios comentarios, los usuarios tienen acceso a los comentarios de otros, ya que todos los comentarios en el foro son públicos y visibles para todas las cuentas, sin anonimato. Esta transparencia fomenta un entorno de interacción abierto y responsable dentro de la comunidad de usuarios.

### **5.2.2.4 CAMBIO DE IDIOMA**

Para implementar la funcionalidad de cambio de idioma en la aplicación web, se exploraron diferentes enfoques. Inicialmente, se intentó utilizar la librería Flask-Babel, una herramienta integrada en Flask. Babel gestiona múltiples idiomas mediante la creación de

archivos de traducción en los que se definen las traducciones literales de las diferentes cadenas de texto que se emplean en la aplicación.

El funcionamiento se basa en la creación de archivos de traducción para cada idioma que se desea soportar en la aplicación. Estos archivos se generan a partir del código fuente de la aplicación, donde las cadenas de texto que requieren traducción son marcadas explícitamente. Posteriormente, se utiliza una herramienta de extracción que analiza el código de la aplicación y genera un archivo de catálogo de mensajes, comúnmente en formato .pot. Este archivo contiene todas las cadenas marcadas para traducción y sirve como base para crear archivos específicos de cada idioma, en los que se asignan traducciones a las cadenas originales.

Sin embargo, debido a problemas con la actualización y descarga de las librerías necesarias, no fue posible seguir este enfoque. Como solución alternativa, se decidió utilizar diccionarios en los que se definen los diferentes idiomas. Estos diccionarios permiten determinar el idioma preferido del usuario mediante una función personalizada y seleccionar las traducciones correspondientes de manera dinámica.

La función implementada, **getlanguage()**, es responsable de determinar el idioma preferido del usuario y devolver una lista con las traducciones adecuadas. En caso de que el usuario esté autenticado, la función verifica el idioma que ha sido configurado como preferido en su perfil. Si el idioma es inglés, se selecciona la lista correspondiente a las traducciones en inglés; si es español, se selecciona la lista en español, y así sucesivamente. Si el usuario no está autenticado, la función devuelve por defecto el inglés.

El código de la función `getlanguage()` es el siguiente::

```
def getlanguage():
    if current_user.is_authenticated:
        if current_user.language == 'en':
            language = translations['English']
        elif current_user.language == 'es':
            language = translations['Spanish']
        .... #Resto de idiomas
    else:
        language = translations['English']

    return language
```

Esta función devuelve un diccionario con las traducciones utilizadas en los archivos HTML. El archivo Python donde se define la función maneja la selección del idioma, asegurando que, al llamar a una posición específica en el archivo HTML, el contenido traducido se corresponda con la preferencia de idioma del usuario. Por ejemplo, en un archivo HTML, para mostrar una cadena traducida, se haría referencia a la posición dentro del diccionario de idioma seleccionado como en el siguiente ejemplo:

```
<div class="input-group-append">
  <button
    type="submit"
    class="send-button">
    <strong>{{ language[5] }}</strong>
  </button>
</div>
```

Aquí, `language[5]` sería reemplazado por el texto correspondiente en el idioma preferido del usuario, lo que permite que la interfaz de usuario se adapte automáticamente al idioma configurado por el usuario. Este enfoque, aunque menos automatizado que el uso de Flask-Babel, garantiza que el sitio sea multilingüe sin depender de librerías externas que podrían no estar completamente actualizadas o ser incompatibles con la configuración del entorno de desarrollo.

### 5.2.3 FLASK

Flask es un microframework de Python ampliamente utilizado para el desarrollo de aplicaciones web. En el contexto de este proyecto, Flask desempeña un papel fundamental como el framework backend, proporcionando la estructura necesaria para manejar las interacciones del servidor y las solicitudes del cliente de manera eficiente.

Flask facilita la integración de APIs externas y el manejo de rutas HTTP, lo que es esencial para implementar las funcionalidades de conversación y síntesis de voz en nuestra aplicación. Además, su compatibilidad con diversas extensiones permite añadir características avanzadas, como la gestión de sesiones de usuario, la autenticación y la conexión con bases de datos, sin complicaciones.

En este proyecto, Flask no solo actúa como el puente entre el frontend y los servicios de backend, sino que también garantiza que las solicitudes de los usuarios sean procesadas de manera rápida y segura. Su capacidad para manejar múltiples peticiones simultáneamente y su soporte para pruebas unitarias hacen de Flask una elección ideal para asegurar que la aplicación mantenga un alto rendimiento y una experiencia de usuario fluida.

#### 5.2.3.1 INTEGRACIÓN DE FLASK EN EL PROYECTO

Para la integración del framework Flask en este proyecto, ha sido necesario desarrollar dos archivos principales: 'main.py' y '\_\_init\_\_.py'. El archivo '\_\_init\_\_.py' es crucial ya que funciona como el núcleo del programa, donde se realiza la creación de la aplicación Flask y la configuración de sus componentes esenciales. Dentro de este archivo, se inicializa la aplicación, se configura la base de datos (que será detallada más adelante), y se registran tanto los Blueprints como el LoginManager. Este archivo contiene la función **createApp()**, que encapsula todo el proceso de configuración y ensamblaje de la aplicación. Esta función es fundamental, ya que permite la creación de una instancia de la aplicación lista para ser ejecutada.

La parte del código correspondiente a la inicialización de Flask dentro de la función `createApp()` es el siguiente:

```
from flask import Flask
def createApp():
    app = Flask(__name__)
    app.config['SECRET_KEY'] = 'hello word'
    #Aquí se añadirán más configuraciones
    return app
```

Se define la función `createApp()`, que es la encargada de configurar y devolver una instancia de la aplicación Flask. Este patrón de diseño es ampliamente utilizado en Flask para estructurar la configuración de manera modular y reutilizable. Dentro de esta función, se crea una instancia de Flask con `app = Flask(__name__)`, donde '`__name__`' indica el nombre del módulo actual, lo que permite que Flask determine la ubicación de la aplicación y sus recursos asociados, como plantillas y archivos estáticos. Además, se configura una clave secreta mediante `app.config['SECRET_KEY'] = 'hello word'`, la cual es utilizada para asegurar sesiones, gestionar tokens de autenticación y otras operaciones sensibles dentro de la aplicación; en un entorno de producción, esta clave debe ser segura y confidencial. Finalmente, la función concluye con `return app`, que devuelve la instancia de la aplicación Flask completamente configurada y lista para su uso.

Posteriormente, en el archivo `main.py`, se crea y se lanza la aplicación llamando directamente a la función `createApp()`:

```
from website import createApp

app = createApp()

if __name__ == '__main__':
    app.run(debug=True)
```



El archivo `main.py` es el punto de entrada de la aplicación. Al ejecutar este archivo, se inicia el programa y la aplicación Flask se pone en marcha, escuchando las peticiones en el servidor local.

### 5.2.3.2 BLUEPRINTS

Como se mencionó en el Capítulo 4, los Blueprints son una herramienta poderosa en Flask para organizar el código de una aplicación en módulos reutilizables y bien estructurados. Esta funcionalidad permite dividir la aplicación en secciones más pequeñas y manejables, lo que facilita el mantenimiento y la escalabilidad del proyecto.

En este proyecto, se han implementado principalmente dos Blueprints: `auth` y `views`.

- **auth:** Este Blueprint se encarga de gestionar todas las rutas relacionadas con la autenticación de usuarios, incluyendo el inicio de sesión, registro y gestión de cuentas.
- **views:** Este Blueprint maneja el resto de las rutas de la aplicación, incluyendo las relacionadas con la interacción principal del usuario, como la conversación guiada por la inteligencia artificial.

La definición de estos Blueprints se realiza de la siguiente manera:

```
from flask import Blueprint
views = Blueprint('views', __name__)
auth = Blueprint('auth', __name__)
```

Un ejemplo práctico del uso de un Blueprint es la definición de la ruta para el inicio de sesión. Primero, en el archivo base de la plantilla HTML, se añade un enlace en la barra de navegación que apunta a la ruta del inicio de sesión:

```
<a class="nav-item nav-link" id="login" href="/login">Login</a>
```

Posteriormente, en el archivo `auth.py`, se define la ruta correspondiente utilizando el Blueprint `auth`:

```
@auth.route('/login', methods=['GET', 'POST'])
def login():
    #Aquí se define la función que maneja el inicio de sesión
```

En este ejemplo, la ruta `/login` se asocia con la función `login()`, que maneja tanto las solicitudes GET como POST. Este enfoque modular permite que cada parte de la aplicación esté claramente separada, facilitando su desarrollo y mantenimiento.

Finalmente, para que estos Blueprints sean reconocidos por la aplicación, es necesario registrarlos en la instancia de Flask en el archivo `__init__.py`. Esto se hace de la siguiente manera:

```
from .views import views
from .auth import auth

app.register_blueprint(views, url_prefix='/')
app.register_blueprint(auth, url_prefix='/')
```

- **Registro de Blueprints:** Con estas líneas de código, se registra cada Blueprint en la aplicación principal Flask. El método `register_blueprint` se utiliza para conectar el Blueprint con la aplicación, permitiendo que las rutas definidas dentro de cada Blueprint sean accesibles.
- **url\_prefix:** El argumento `url_prefix` define un prefijo que se añadirá a todas las rutas dentro del Blueprint. En este caso, el Blueprint `views` está registrado con el prefijo `/`, lo que significa que sus rutas estarán disponibles directamente en la raíz del sitio web. Por ejemplo, si se define una ruta `/home` dentro del Blueprint `views`, estará disponible en `{ruta_sitio_web}/home`.

El registro de Blueprints en ‘\_\_init\_\_.py’ es un paso crucial para asegurarse de que todas las rutas y funcionalidades definidas en los diferentes módulos de la aplicación estén correctamente integradas en la aplicación principal. Esto permite que la aplicación sea extensible y modular, facilitando futuras expansiones y mejoras.

### 5.2.3.3 FLASK-LOGIN

La extensión Flask-Login se utiliza en este proyecto para gestionar el inicio de sesión y la autenticación de usuarios, permitiendo que la aplicación sepa en todo momento qué usuario está conectado y personalizando las funciones en función de la cuenta activa. Su implementación se desarrolla en varios puntos clave de la aplicación, como en el archivo \_\_init\_\_.py, dentro de los Blueprints, y en el archivo Models.py.

En el archivo \_\_init\_\_.py, se configura el LoginManager, que se encarga de gestionar las sesiones de los usuarios. Este componente se integra en la aplicación durante su creación y define, entre otras cosas, cuál es la vista de inicio de sesión predeterminada (login\_view) y cómo se carga un usuario desde la base de datos cuando se inicia una sesión. Esta configuración asegura que el sistema pueda autenticar y mantener la sesión de un usuario activo a lo largo de su interacción con la aplicación. El código relevante es el siguiente:

```
def createApp():
    login_manager = LoginManager()
    login_manager.login_view = 'auth.login'
    login_manager.init_app(app)

    @login_manager.user_loader
    def load_user(user_id):
        return User.query.get(int(user_id))
```

```
from flask_login import login_user, login_required, logout_user, current_user
```

Dentro del Blueprint auth, se encuentran varias funciones específicas para manejar las sesiones de usuario. Estas funciones incluyen `login_user`, que inicia la sesión de un usuario, y `logout_user`, que la cierra. Un aspecto importante de esta configuración es el uso de `login_required`, una función que restringe el acceso a ciertas rutas o funciones solo a aquellos usuarios que están autenticados. Esto garantiza que operaciones sensibles, como la configuración de la cuenta o el cierre de sesión, solo puedan ser realizadas por usuarios que han iniciado sesión. Además, la función `current_user` se utiliza para acceder al usuario actualmente conectado. Esta función es también utilizada por las views para personalizar la experiencia del usuario en tiempo real, ya que permite obtener información del usuario activo y adaptar la interfaz o las funciones disponibles en función de sus datos y permisos, además de obtener atributos específicos del usuario conectado, como el correo y la contraseña, lo cual ha sido esencial para el desarrollo.

```
from flask_login import UserMixin
```

Finalmente, en el archivo `Models.py`, se utiliza la clase `UserMixin` en el modelo `User`. `UserMixin` es una clase proporcionada por `Flask-Login` que añade métodos y propiedades esenciales para gestionar usuarios autenticados, como la capacidad de verificar si un usuario está autenticado, obtener su ID y manejar sesiones de manera efectiva. Este enfoque facilita la integración del sistema de autenticación con la lógica de la aplicación y asegura que la gestión de usuarios sea consistente y segura en todo el proyecto. El modelo `User` se discutirá en mayor detalle en su respectivo apartado.

#### 5.2.4 SQLALCHEMY

En el desarrollo de aplicaciones web, la gestión eficiente de bases de datos es fundamental para asegurar que los datos se almacenen, recuperen y manipulen de manera efectiva. Para el proyecto, se ha optado por utilizar **SQLAlchemy** como la herramienta principal para la interacción con la base de datos. **SQLAlchemy** es una potente biblioteca de Python que facilita la manipulación de bases de datos relacionales a través de un enfoque de

mapeo objeto-relacional (ORM). Esto permite a los desarrolladores interactuar con la base de datos utilizando objetos y clases de Python en lugar de escribir consultas SQL directamente. En este apartado, se explorará cómo SQLAlchemy ha sido implementado en el proyecto, destacando sus características clave y cómo contribuye a la robustez y escalabilidad del sistema.

#### 5.2.4.1 INTEGRACIÓN DE SQLALCHEMY EN EL PROYECTO

En esta sección se detalla el proceso específico seguido para configurar SQLAlchemy, tomando en cuenta la estructura y necesidades del proyecto.

El primer paso fue la instalación de SQLAlchemy y la extensión Flask-SQLAlchemy para facilitar su integración con Flask. Esta instalación se realizó a través del gestor de paquetes pip, utilizando el siguiente comando:

```
pip install SQLAlchemy Flask-SQLAlchemy
```

Esto aseguró que todas las dependencias necesarias estuvieran disponibles para el correcto funcionamiento de SQLAlchemy dentro del proyecto.

La configuración de SQLAlchemy en la aplicación se realizó en el archivo principal de la misma, donde se definieron las configuraciones clave. A continuación, se muestra el código utilizado:

```
db = SQLAlchemy()
DB_NAME = "database.db"

def createApp():

    app.config['SQLALCHEMY_DATABASE_URI'] = f'sqlite:///{DB_NAME}'
    db.init_app(app)
```

En este fragmento de código, primero se define una instancia de SQLAlchemy mediante `db = SQLAlchemy()`, que será utilizada para interactuar con la base de datos dentro de la aplicación Flask. A continuación, se establece el nombre del archivo de la base de datos en `DB_NAME`, que en este caso es "database.db".

Dentro de la función `createApp()`, se configura la aplicación Flask para utilizar SQLite como su sistema de gestión de bases de datos, asignando la URI de la base de datos con `app.config['SQLALCHEMY_DATABASE_URI'] = f'sqlite:/// {DB_NAME}'`. Esta URI indica a SQLAlchemy que debe utilizar SQLite y que la base de datos se almacenará en un archivo local con el nombre especificado.

Finalmente, se inicializa la instancia de SQLAlchemy con la aplicación Flask mediante `db.init_app(app)`. Esto integra SQLAlchemy con la aplicación, permitiendo que se realicen operaciones en la base de datos a través de la instancia `db`. Este paso es esencial para asegurar que la base de datos esté correctamente configurada y lista para ser utilizada en toda la aplicación.

#### 5.2.4.2 MODELS

En el contexto de una aplicación web, un **Model** es una representación estructurada de los datos de la aplicación que interactúa directamente con la base de datos. Los modelos en Flask-SQLAlchemy se utilizan para definir las tablas de la base de datos y sus columnas, así como las relaciones entre diferentes conjuntos de datos. En este proyecto, se han declarado dos modelos principales: **User** y **Post**.

El modelo **User** se utiliza para gestionar la información relacionada con los usuarios de la aplicación. La estructura del modelo es la siguiente:

```
class User(db.Model, UserMixin):
    id = db.Column(db.Integer, primary_key=True)
    email = db.Column(db.String(150), unique=True)
    password = db.Column(db.String(150))
    first_name = db.Column(db.String(150))
    language = db.Column(db.String(50), default='en')
    posts = db.relationship('Post', backref='user', lazy=True)
```

- **id**: Es la clave primaria del modelo User, que identifica de manera única a cada usuario en la base de datos. Se define como un entero (Integer).
- **email**: Este atributo almacena la dirección de correo electrónico del usuario. Es una cadena de hasta 150 caracteres (String(150)) y es único (unique=True), lo que significa que no puede haber dos usuarios con el mismo correo electrónico en la base de datos.
- **password**: Este campo almacena la contraseña del usuario en forma de una cadena de hasta 150 caracteres. La seguridad de las contraseñas se gestiona a través de un hash, aunque esto no se refleja directamente en el modelo.
- **first\_name**: Aquí se guarda el nombre de pila del usuario, también como una cadena de hasta 150 caracteres.
- **language**: Este atributo define el idioma preferido del usuario, almacenado como una cadena de hasta 50 caracteres, con un valor predeterminado de 'en' (inglés).
- **posts**: Esta es una relación entre el modelo User y el modelo Post. Define que un usuario puede tener múltiples posts asociados. La relación se establece mediante `db.relationship('Post', backref='user', lazy=True)`, lo que permite acceder a todos los posts de un usuario desde este atributo. El `backref='user'` crea un atributo en el modelo Post que permite acceder al usuario propietario de cada post, y `lazy=True` indica que SQLAlchemy cargará los posts relacionados solo cuando sean accedidos.

El modelo **Post** representa los comentarios que cada usuario publica en el foro de la aplicación. Su estructura es la siguiente:

```
class Post(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    data = db.Column(db.String, nullable=False)
    date = db.Column(db.DateTime(timezone=True), default=func.now())
    user_id = db.Column(db.Integer, db.ForeignKey('user.id'),
nullable=False)
```

- **id**: Similar al modelo User, este es el identificador único de cada post en la base de datos.
- **data**: Este campo almacena el contenido del post, que es una cadena de texto. Es obligatorio (`nullable=False`), lo que significa que no se puede guardar un post sin contenido.
- **date**: Aquí se almacena la fecha y hora en la que se creó el post. Se utiliza el tipo de dato `DateTime` con soporte de zona horaria (`timezone=True`), y el valor predeterminado es la hora actual en que se crea el post (`default=func.now()`).
- **user\_id**: Este campo es una clave foránea (`ForeignKey`) que conecta cada post con un usuario específico. Hace referencia a la columna `id` del modelo User (`'user.id'`) y establece una relación de muchos a uno, indicando que muchos posts pueden pertenecer a un solo usuario. Es obligatorio (`nullable=False`).

La relación entre los modelos User y Post es una relación de uno a muchos. Un usuario puede tener múltiples posts, pero cada post está asociado a un único usuario. Esta relación se gestiona a través de la clave foránea `user_id` en el modelo Post y la relación definida en el modelo User mediante el atributo `posts`. Esto permite, por ejemplo, recuperar todos los posts realizados por un usuario específico, y también permite identificar fácilmente al usuario que creó un post determinado. Esta relación es fundamental para mantener la integridad de los datos y facilitar la navegación entre los usuarios y sus publicaciones dentro de la aplicación.



### 5.2.4.3 CREACIÓN DE BBDD

La creación de la base de datos es un proceso crucial en el desarrollo de aplicaciones web, ya que garantiza la persistencia y la organización adecuada de la información que maneja la aplicación. En este proyecto, la creación de la base de datos se realiza dentro de la función `createApp()`, que centraliza la configuración e inicialización de la aplicación. A continuación, se explica en detalle el proceso:

```
db = SQLAlchemy()
DB_NAME = "database.db"

def createApp():

    create_database(app)

def create_database(app):
    if not path.exists('website/' + DB_NAME):
        with app.app_context():
            db.create_all()
            print('Created Database!')
```

La función `create_database(app)` se encarga de verificar si el archivo de la base de datos ya existe en la ruta especificada (`'website/' + DB_NAME`). Si el archivo no existe, la función crea la base de datos y todas las tablas necesarias utilizando `db.create_all()` dentro del contexto de la aplicación (`app.app_context()`), lo cual es esencial para que SQLAlchemy pueda acceder correctamente al contexto de la aplicación al crear las tablas. Este proceso garantiza que las tablas definidas en los modelos de SQLAlchemy se generen adecuadamente en la base de datos. Al final, se imprime el mensaje de confirmación "Created Database!" en la consola para indicar que la base de datos ha sido creada con éxito.

Este enfoque garantiza que la base de datos esté siempre disponible y correctamente configurada cuando se inicie la aplicación, proporcionando una base sólida para la gestión de datos en la aplicación web.

## 5.3 DESARROLLO FRONTEND

El desarrollo frontend del proyecto se centra en la creación de una interfaz de usuario (UI) intuitiva, accesible y eficiente que permita a los usuarios interactuar de manera sencilla y directa con el sistema. Dado que el objetivo principal del proyecto es facilitar la comunicación para personas con dificultades del habla, se ha puesto un énfasis especial en la usabilidad y en la accesibilidad, asegurando que la interfaz sea fácil de navegar y que cumpla con las normativas de accesibilidad web.

El frontend del proyecto ha sido desarrollado utilizando una combinación de tecnologías modernas como HTML y CSS, junto con frameworks como Bootstrap para el diseño responsivo. Esto ha permitido crear una interfaz que no solo es visualmente atractiva, sino también funcional en una variedad de dispositivos y tamaños de pantalla.

En este punto, se detalla la estructura del frontend, el diseño de la interfaz, y las funcionalidades implementadas. Se exploran las decisiones de diseño tomadas para optimizar la experiencia del usuario, así como las integraciones realizadas para asegurar que la interfaz funcione de manera fluida con el backend del sistema. Además, se abordarán las pruebas de usabilidad y las mejoras continuas realizadas en base a la retroalimentación de los usuarios.

### 5.3.1 DISEÑO DE INTERFAZ

El diseño de la interfaz es un aspecto crítico para garantizar que los usuarios puedan interactuar de manera intuitiva y accesible con la aplicación. Este diseño debe considerar varios componentes clave, como la elección del tipo de fuente, la paleta de colores y la disposición de los elementos en la pantalla, todo lo cual contribuye a la usabilidad y la estética del sitio web. Para asegurar la coherencia y facilitar el mantenimiento del código, se ha decidido centralizar todos estos aspectos comunes en un archivo de plantilla base denominado base.html. Este archivo actúa como la estructura principal de la cual heredan el

resto de las páginas del sitio, lo que simplifica la implementación de cambios globales y asegura una experiencia de usuario uniforme.

El archivo `base.html` incluye una barra de navegación fija en la parte superior de la pantalla, que permite a los usuarios moverse fácilmente entre las diferentes secciones de la aplicación. Además, se han definido estilos básicos que afectan a la tipografía, el espaciado y otros elementos visuales que son consistentes en todas las páginas.

Un ejemplo de como el resto de los archivos extienden `base.html` es el siguiente:

```
{% extends "base.html" %}
```

- Tipo de fuente: "Helvetica Neue", Helvetica, Arial, sans-serif.
- Paleta de colores:
  - Fondo de la barra de navegación: #343a40 (gris oscuro).
  - Texto de la barra de navegación: #ffffff (blanco).
  - Alertas de error: #dc3545 (rojo).
  - Alertas de éxito: #28a745 (verde).

### 5.3.1.1 USO DE BOOTSTRAP

El framework Bootstrap ha sido una herramienta esencial en el desarrollo de la interfaz de usuario, proporcionando una base sólida para crear un diseño responsivo y moderno. Bootstrap facilita la creación de una estructura coherente a lo largo de todas las páginas del sitio mediante el uso de su sistema de rejillas, componentes predefinidos y utilidades CSS.

En el archivo `base.html`, Bootstrap se utiliza para implementar la barra de navegación (navbar) que se encuentra en la parte superior de la página. Esta barra incluye enlaces de navegación y se adapta automáticamente a diferentes tamaños de pantalla gracias a las clases `navbar-expand-lg`, `navbar-dark`, y `bg-dark`. Además, se han incluido estilos personalizados

para elementos específicos como la marca del sitio (navbar-brand), asegurando que el título del sitio sea prominente y fácilmente reconocible.

Otro componente importante es la inclusión de mensajes de alerta, que se gestionan dinámicamente según las acciones del usuario (como el inicio de sesión o la configuración de su perfil). Estos mensajes se muestran utilizando las clases de alerta de Bootstrap, lo que asegura que se presenten de manera clara y visualmente consistente, con diferentes estilos para errores y confirmaciones (alert-danger y alert-success respectivamente).

Finalmente, Bootstrap también se utiliza para gestionar el espaciado y el posicionamiento del contenido dentro de la página, utilizando contenedores (container) que garantizan que los elementos se mantengan centrados y con un espaciado adecuado, independientemente del dispositivo utilizado para acceder al sitio.

Para utilizar Bootstrap, primero se deben incluir los enlaces a los estilos necesarios en el archivo base.html:

```
<link
  rel="stylesheet"
  href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css"
  integrity="sha384-
Vkoo8x4CGs03+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
  crossorigin="anonymous"
/>
<link
  rel="stylesheet"
  href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-
awesome.min.css"
  crossorigin="anonymous"
/>
```

Una vez incluidos los estilos, se pueden utilizar los componentes predefinidos de Bootstrap. Por ejemplo, la barra de navegación en la parte superior de la página, específicamente el nombre de la página como un botón, se define de la siguiente manera:

```
<nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top">
  <a class="navbar-brand" href="/">AI for Accessibility</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarSupportedContent">
    <span class="navbar-toggler-icon"></span>

    <!-- Elementos adicionales de la barra de navegación -->

</button>
```

## 5.3.2 FUNCIONALIDADES IMPLEMENTADAS

### 5.3.2.1 CREACIÓN DE USUARIOS



The screenshot shows a web interface for user registration. At the top left, it says "AI for Accessibility". At the top right, there are links for "Login" and "Sign Up". The main heading is "Sign Up". Below this, there are four text input fields: "Email Address" (with placeholder "Enter email"), "First Name" (with placeholder "Enter first name"), "Password" (with placeholder "Enter password"), and "Password (Confirm)" (with placeholder "Confirm password"). At the bottom left of the form is a blue "Submit" button.

*Figura 24 Registro de Usuario*

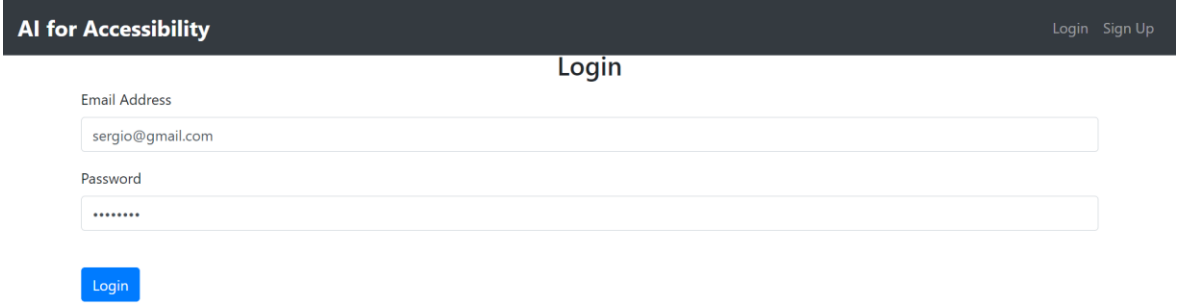
La funcionalidad de **Creación de Usuarios** es esencial en la aplicación web, ya que permite a los nuevos usuarios registrarse y crear una cuenta personalizada.

La interfaz de registro, como se muestra en la figura 5.4, sigue un diseño simple y funcional para asegurar que el proceso de creación de cuentas sea accesible y fácil de usar. La página de registro incluye los siguientes campos:

- **Email Address:** Campo de entrada de texto para que el usuario ingrese su dirección de correo electrónico. Este campo es obligatorio, ya que el correo electrónico se utilizará como identificador único para el usuario.
- **First Name:** Campo de entrada de texto donde el usuario debe ingresar su nombre. Este dato es importante para personalizar la experiencia del usuario en la aplicación.

- **Password:** Campo de entrada de texto para que el usuario ingrese su contraseña. Este campo es de tipo "password" para asegurar que los caracteres ingresados se muestren como asteriscos o puntos, protegiendo la privacidad del usuario.
- **Password (Confirm):** Un segundo campo de contraseña para que el usuario confirme su contraseña. Este paso es crucial para evitar errores de escritura y así asegurar que el usuario recuerde la contraseña ingresada.

### 5.3.2.2 INICIO DE SESIÓN



AI for Accessibility Login Sign Up

## Login

Email Address

Password

*Figura 25 Inicio de Sesión*

La funcionalidad de **Inicio de Sesión** en la aplicación es un componente crítico que permite a los usuarios acceder a sus cuentas previamente creadas. Este proceso garantiza que cada usuario pueda interactuar de manera segura con la plataforma, accediendo a sus datos y configuraciones personales.

La interfaz de inicio de sesión, como se puede observar en la figura 5.5, está diseñada para ser clara y sencilla, asegurando una experiencia de usuario intuitiva. La página contiene únicamente dos campos principales:

- **Email Address:** Campo de entrada de texto donde el usuario ingresa su dirección de correo electrónico, que se utiliza como identificador único para su cuenta.
- **Password:** Campo de entrada de texto para que el usuario ingrese su contraseña. Este campo es de tipo "password" para asegurar que los caracteres ingresados no sean visibles, protegiendo así la privacidad del usuario.



### 5.3.2.3 PANTALLA DE INICIO

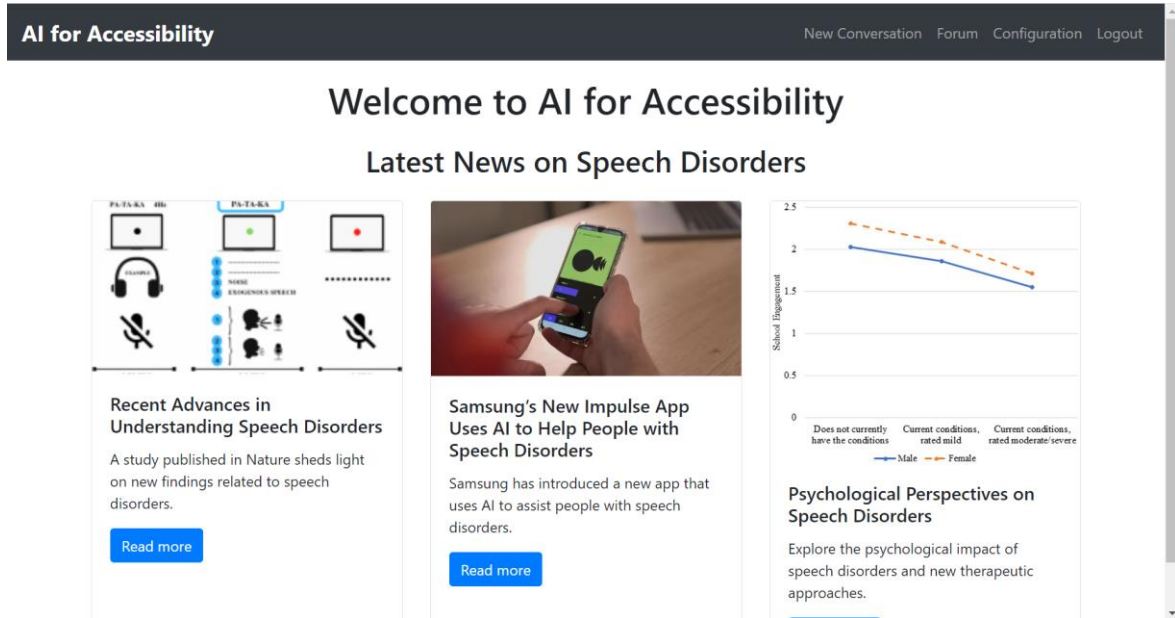
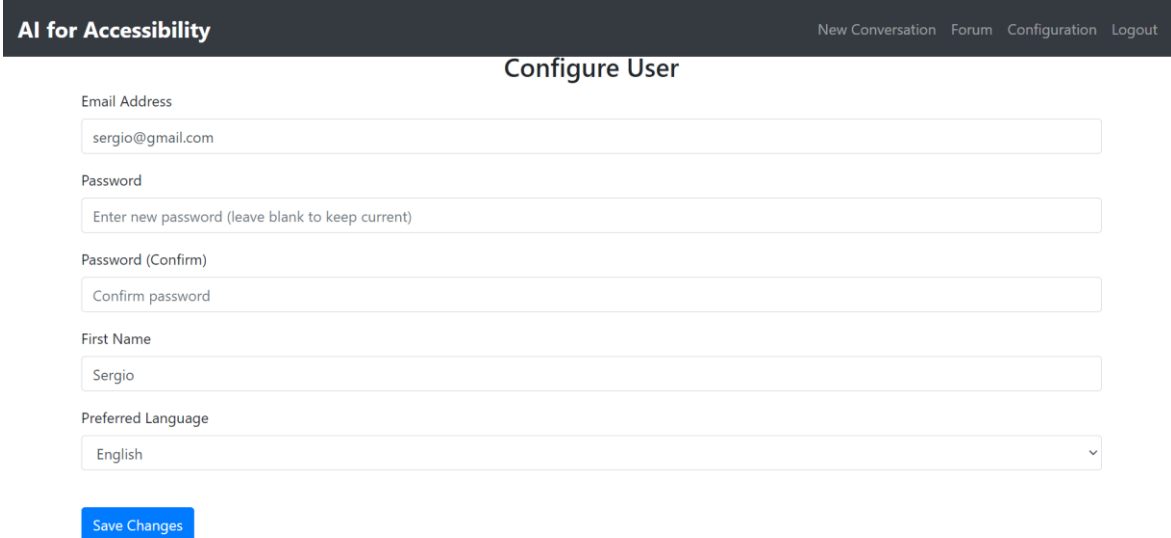


Figura 26 Pantalla de Inicio

La **Pantalla de Inicio** es lo primero que se ve al iniciar sesión y muestra las últimas noticias relacionadas con los trastornos del habla. Actualmente, se presentan las tres noticias mostradas en la figura 5.6, pero en el futuro, se designará a una persona responsable de mantenerlas actualizadas. En cada una de las noticias se puede ver una imagen relacionada con la imagen, el título de la noticia, un mini resumen y un botón “Read more”, que al hacer clic en él, se abrirá una pestaña que lleva al usuario a la fuente original de la noticia en la web de la que proviene.

### 5.3.2.4 CONFIGURACIÓN DE USUARIO



**AI for Accessibility** [New Conversation](#) [Forum](#) [Configuration](#) [Logout](#)

#### Configure User

Email Address  
sergio@gmail.com

Password  
Enter new password (leave blank to keep current)

Password (Confirm)  
Confirm password

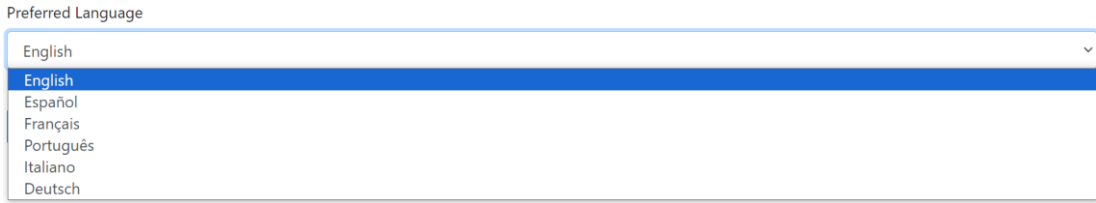
First Name  
Sergio

Preferred Language  
English

[Save Changes](#)

*Figura 27 Configuración de Usuario*

La funcionalidad de **Configuración de Usuario** permite a los usuarios modificar los datos proporcionados durante el registro, así como ajustar las preferencias de la aplicación, como el idioma preferido. Una vez que un usuario ha creado su cuenta, puede acceder a esta sección para actualizar su información personal y asegurar que la plataforma se adapte a sus necesidades. específicas.



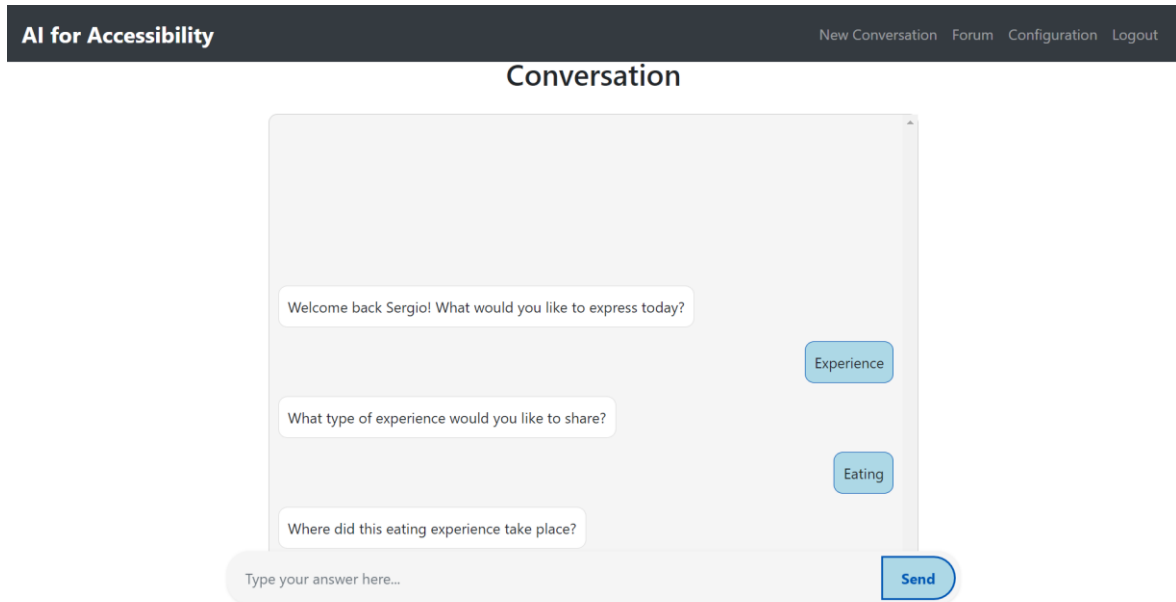
Preferred Language

- English
- English**
- Español
- Français
- Português
- Italiano
- Deutsch

*Figura 28 Selección de Idioma*

La interfaz presenta los mismos campos que la página de registro, con la adición de un campo desplegable para la selección del idioma (**Preferred Language**), el cual se puede ver en la figura 5.8. Esta opción permite a los usuarios cambiar el idioma de la aplicación, facilitando su uso en diferentes entornos lingüísticos.

### 5.3.2.5 CONVERSACIÓN PARA LA ENTRADA DE DATOS



*Figura 29 Conversación*

La funcionalidad de **Conversación** es el núcleo del sistema, ya que se encarga de recopilar los datos necesarios para generar posteriormente el archivo de audio. En la figura 5.9 se muestra un ejemplo de esta interacción, donde el usuario es guiado a través de preguntas simples y específicas, como "¿Qué tipo de experiencia te gustaría compartir?" o "¿Dónde tuvo lugar esta experiencia de comida?". Estas preguntas tienen como objetivo estructurar la información proporcionada por el usuario, facilitando la creación de una historia coherente y personalizada que será convertida en audio. Este flujo de conversación no solo asegura que los detalles relevantes sean capturados de manera eficaz, sino que también ofrece una interfaz accesible y fácil de usar para personas con dificultades en el habla.

### 5.3.2.6 RESULTADOS Y REPRODUCCIÓN DE AUDIO

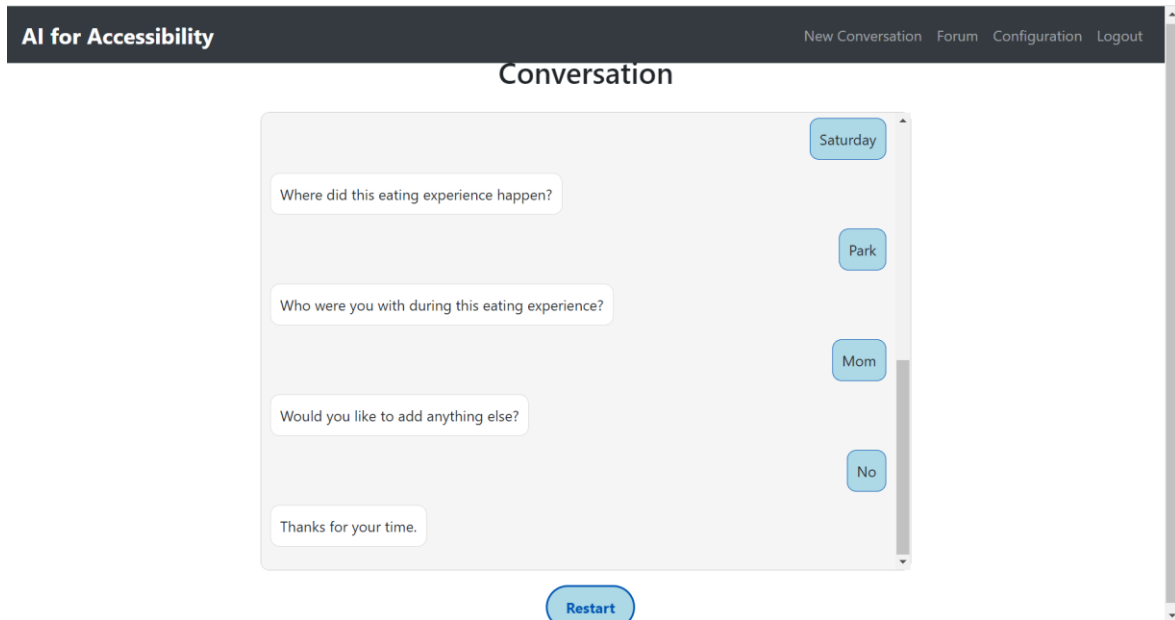
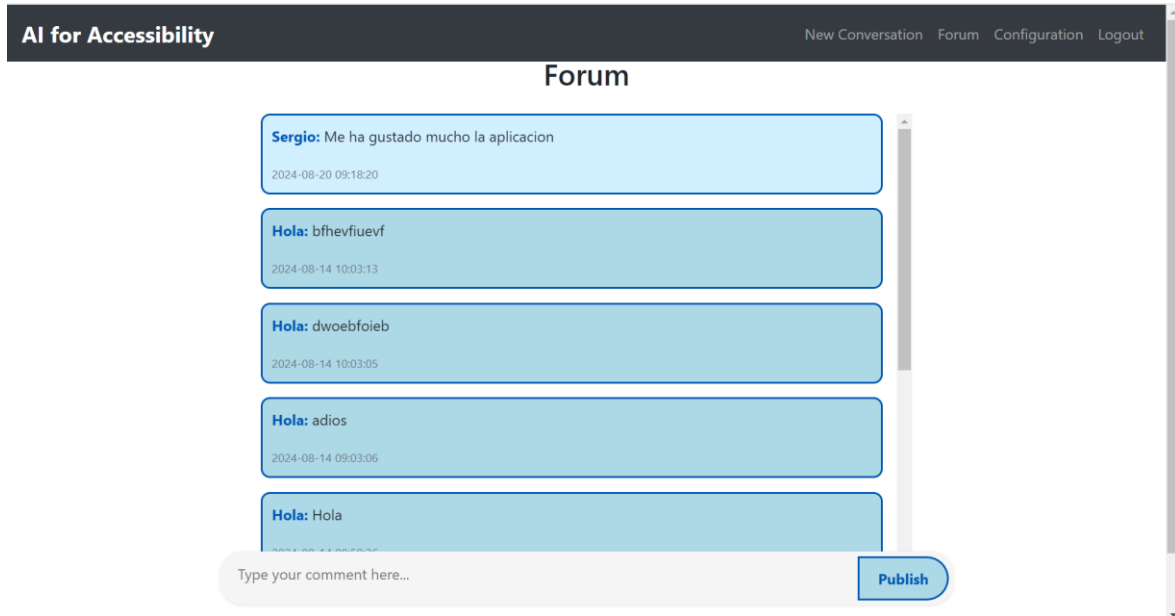


Figura 30 Resultados de la Conversación

La Figura 5.10 muestra la pantalla final de la funcionalidad de **Conversación**, donde se presentan los resultados de la interacción. En esta etapa, el sistema ha recopilado toda la información proporcionada por el usuario y está listo para generar el archivo de audio correspondiente. Se incluyen las respuestas a las preguntas, asegurando que todos los detalles relevantes sean capturados. Una vez finalizada la conversación, se devuelve el mensaje final, y se reproduce automáticamente el audio generado, una vez el audio se ha reproducido, se presenta un botón de "Restart" para reiniciar la conversación, permitiendo al usuario comenzar un nuevo proceso desde el principio si lo desea. Esta pantalla cierra el ciclo de interacción, proporcionando una manera simple y directa de revisar y reutilizar la funcionalidad principal de la aplicación.

### 5.3.2.7 INTERACCIÓN CON EL FORO



*Figura 31 Foro de la Aplicación*

La aplicación incluye una funcionalidad de foro donde los usuarios pueden interactuar publicando y leyendo comentarios. En la Figura 5.11 se muestra cómo los comentarios se despliegan en su idioma original, permitiendo a los usuarios compartir y recibir opiniones sin barreras lingüísticas. Los comentarios del propio usuario se destacan en un color azul más claro, diferenciándolos visualmente de los comentarios de otros usuarios, lo que facilita su identificación en la conversación. Además, la interfaz incluye un campo de texto en la parte inferior para que los usuarios puedan escribir y publicar nuevos comentarios de manera rápida y sencilla, promoviendo así la interacción dentro de la comunidad de la aplicación.

## 5.4 INTEGRACIÓN DE APIS

Como se ha señalado previamente, la integración de APIs externas ha sido un componente crucial para el desarrollo de las funcionalidades clave del proyecto. En esta sección, se detallará cómo estas APIs han sido incorporadas al sistema, explorando su integración y destacando las funciones principales que cada una aporta. Este análisis no solo demostrará la importancia de estas herramientas en la construcción del proyecto, sino que también proporcionará una visión clara de cómo potencian las capacidades de la aplicación.

### 5.4.1 OPENAI

#### 5.4.1.1 FUNCIONES DESARROLLADAS

El archivo `openai_connection.py` es crucial para la integración y manejo de la API de OpenAI dentro del proyecto. Este archivo contiene varias funciones estáticas que permiten la comunicación efectiva con la API de OpenAI, facilitando tanto la gestión de la conversación como la generación de narrativas a partir de las respuestas del usuario. A continuación, se describen las principales funciones desarrolladas en este archivo:

- **initialize(language)**

La función `initialize` es utilizada al comienzo de una nueva conversación para configurar las estructuras iniciales necesarias. Recibe un parámetro `language`, que determina el idioma en el que se llevará a cabo la conversación.

Dentro de esta función, se definen dos listas de mensajes: `json` y `final_json`. La lista `json` contiene un mensaje inicial que instruye al modelo sobre cómo proceder con la recopilación de información del usuario, asegurando que las preguntas sean concisas y específicas. Por otro lado, `final_json` contiene un mensaje que instruye al modelo para sintetizar la información en un párrafo narrativo al final de la conversación. Los mensajes utilizados en cada uno de los `json` son los siguientes:

- json:

```
f"To effectively gather information on an idea or experience, it's important to analyze the preceding conversation for continuity. Develop questions that encourage specific and detailed yet concise responses. Structure your inquiries to receive answers limited to one or two words. Never ask for descriptions of any kind; instead, ask questions designed to be answered in no more than two words. Following the initial question, \"What would you like to express today?\", proceed with \"What type of (previous answer) would you like to share?\". If the user is describing an experience, always make sure to ask when and where (in different questions). Always have in mind that these questions are meant to facilitate the creation of a narrative based on the shared experiences or ideas (the narrative creation itself is not required here, don't do it). Gather as much information as you can, names, places, feelings etc. Continuously evaluate the need for additional information. If no further details are necessary, end with the question, \"Would you like to add anything else?\" if the answer is yes, you will eventually have to ask this again until the answer is no, ensure to ask at least twenty questions (counter=(count number of previous answers)), or more if necessary. If the answer to the final question is no, respond with \"{language[9]}\", just this anything else. Don't include the previous answers in the questions. The first question is going to set the language of the conversation."
```

- final\_json

```
"In this task, you'll find a conversation where the assistant is collecting details about a specific topic. Please synthesize all the discussed information into a first-person narrative paragraph as if you were the one answering the questions. Begin the paragraph directly, without introductory phrases like 'based on the information' or 'with the provided information,' etc., The paragraph has to be in the language of the conversation."
```

La función retorna ambas listas (json y final\_json), que luego son utilizadas a lo largo de la interacción para gestionar la conversación y generar la narrativa final.

- **last\_question(json)**

La función `last_question` es utilizada para generar la próxima pregunta en la conversación en función de las respuestas anteriores del usuario. Recibe como parámetro `json`, que es una lista de mensajes donde se ha almacenado el historial de la conversación.

Se hace una llamada a la API de OpenAI para generar una respuesta basada en los mensajes anteriores. La API devuelve una respuesta que contiene la siguiente pregunta que debe ser presentada al usuario. Esta pregunta es extraída y retornada por la función, permitiendo que la conversación fluya de manera continua y natural.

- **conversation(conversation, json, final\_json, question, answer)**

La función `conversation` es responsable de gestionar la lógica interna de la conversación. Recibe varios parámetros: `conversation`, que es un diccionario que almacena las preguntas y respuestas de la sesión actual; `json` y `final_json`, que son listas que contienen los mensajes que se envían a la API para generar preguntas y el párrafo final, respectivamente; `question`, que es la pregunta actual presentada al usuario; y `answer`, que es la respuesta del usuario.

Esta función agrega la pregunta y la respuesta al diccionario `conversation` y a las listas `json` y `final_json`. Luego, llama a la función `last_question` para obtener la siguiente pregunta de la conversación. De esta manera, se asegura que la conversación continúe de forma fluida y que cada nueva pregunta esté basada en el contexto completo de la interacción anterior.

- **create\_paragraph(messages)**

Esta función tiene como objetivo principal generar un párrafo narrativo basado en la conversación previamente sostenida con el usuario. Se encarga de enviar una petición a la API de OpenAI utilizando el modelo "gpt-3.5-turbo". La función recibe como parámetro `messages` (`final_json`), que es una lista de mensajes en formato JSON acumulados durante la



interacción con el usuario. Estos mensajes incluyen tanto las preguntas generadas por el asistente como las respuestas proporcionadas por el usuario.

La función configura la llamada a la API especificando varios parámetros, como `temperature`, `max_tokens`, `top_p`, `frequency_penalty`, y `presence_penalty`, que influyen en el estilo y la cantidad de texto generado. Una vez que OpenAI devuelve la respuesta, se extrae el contenido del primer bloque de respuestas (`choices[0].message.content`) y se retorna como el párrafo resultante.

Esta función es esencial al final del proceso de conversación, ya que sintetiza toda la información recopilada en un formato coherente que luego será convertido a un archivo de audio.

- **Ejemplo de llamada a la API**

```
response = OpenAIConnection.client.chat.completions.create(  
    model= "gpt-3.5-turbo",  
    messages=json,  
    temperature=1,  
    max_tokens=4096,  
    top_p=1,  
    frequency_penalty=0,  
    presence_penalty=0  
)  
  
OpenAI_response = response.choices[0].message.content
```

Este código realiza una solicitud a la API de OpenAI utilizando el modelo "gpt-4o" para generar una respuesta de chat. Primero, se envía un conjunto de mensajes en formato JSON al modelo, configurando parámetros como la temperatura (que controla la creatividad de la respuesta), el número máximo de tokens en la respuesta, y otros ajustes como `top_p`, `frequency_penalty`, y `presence_penalty`, que afinan cómo se genera el texto. Una vez que la API devuelve una respuesta, se accede al contenido del primer resultado de la lista de opciones (`choices`), y se extrae el texto de la respuesta generada (`message.content`) para su uso posterior.

### 5.4.1.2 INTEGRACIÓN EN EL PROYECTO

En el proceso de interacción con el usuario, la estructura de la conversación es cuidadosamente diseñada para garantizar una experiencia fluida y coherente. Dos elementos clave en este proceso son la primera y la última pregunta de la conversación, que están predefinidas y juegan un papel fundamental en el flujo de la interacción. A continuación, se detalla cómo se gestiona esta estructura y personalización dentro del sistema.

La primera pregunta en la conversación es siempre la misma y está diseñada para ser personalizada con el nombre del usuario. Esta pregunta tiene como objetivo establecer el contexto inicial de la interacción y se genera cuando el usuario accede a la ruta `/newconversation`. Dentro de la función `new_conversation` en `views.py`, se verifica si la conversación ya existe en la sesión. Si no existe, se inicializa una nueva sesión y se define la primera pregunta utilizando el nombre del usuario y elementos del idioma seleccionado:

```
language[7] = 'Welcome back '  
language[8] = '! What would you like to express today?'  
  
session['question']="{language1}{first_name}{language2}".format(language1=  
language[7], first_name=current_user.first_name, language2=language[8])
```

En este fragmento de código, `language[7]` y `language[8]` son elementos del diccionario de idioma que rodean el nombre del usuario (`first_name`) para formar una pregunta natural y personalizada. Esto asegura que cada vez que un usuario inicia una conversación, la primera interacción se sienta directa y relevante.

La última pregunta en la conversación también es predefinida y tiene un rol crucial en indicar que la interacción ha llegado a su fin. Durante el flujo de la conversación, el sistema sigue acumulando preguntas y respuestas hasta que se llega a la pregunta final, que es siempre la misma y es generada automáticamente por la API de OpenAI:

```
language[9] = 'Thanks for your time.'  
  
if question == language[9]:  
    #Funciones de creación de párrafo y audio
```

En este caso, `language[9]` corresponde a la pregunta final en el diccionario de idioma, que como se puede ver en el código proporcionado, es igual a gracias por tu tiempo, esto se devolverá siempre que el asistente pregunte si se desea añadir algo mas y la respuesta sea “No”, el asistente cierra la conversación y ya se genera el párrafo final y el audio que se reproducirá.

- **Gestión de la Conversación: Diccionarios, JSONs y Sesiones**

Toda la conversación se maneja mediante diccionarios, estructuras JSON y sesiones, lo que permite una gestión ordenada y eficiente de las preguntas y respuestas.

El diccionario `conversation` almacena cada pregunta como una clave y cada respuesta como su valor correspondiente. Este diccionario se actualiza en cada iteración de la conversación:

```
conversation[question] = answer
```

Además, dos listas de JSON (`json` y `final_json`) se utilizan para almacenar los mensajes que se envían a la API de OpenAI. `json` mantiene el historial completo de preguntas y respuestas para continuar generando nuevas preguntas, mientras que `final_json` se utiliza para generar el párrafo final que sintetiza toda la conversación:

```
json.append({
    "role": "assistant",
    "content": question
})

json.append({
    "role": "user",
    "content": answer
})
```

Las sesiones juegan un papel crucial en mantener el estado de la conversación a través de múltiples interacciones del usuario. En un entorno web, las sesiones permiten que el servidor retenga información entre las solicitudes HTTP, que son naturalmente independientes entre sí. Esto es esencial para que la conversación con el usuario pueda continuar de manera coherente sin perder el contexto.

En este sistema, cada elemento de la conversación (como `conversation`, `json`, `final_json`, y `question`) se almacena en la sesión. Esto asegura que, incluso si el usuario refresca la página o realiza varias solicitudes, el estado de la conversación se mantiene:

```
session['conversation'] = conversation
session['json'] = json
session['final_json'] = final_json
session['question'] = question
```

Al inicio de una nueva conversación, si ya existe una sesión activa con estos datos, se reutiliza; de lo contrario, se inicializan los valores necesarios. Al finalizar la conversación, la sesión se limpia para permitir un nuevo ciclo de interacción:

```
session.pop('conversation', None)
session.pop('json', None)
session.pop('final_json', None)
session.pop('question', None)
```

Este manejo de sesiones es fundamental para proporcionar una experiencia de usuario consistente y sin interrupciones, permitiendo que el sistema mantenga un flujo de conversación coherente y personalizado.

## 5.4.2 AZURE

### 5.4.2.1 FUNCIONES DESARROLLADAS

En el desarrollo del proyecto, se ha implementado una función principal para integrar la API de Azure Cognitive Services, específicamente la funcionalidad de síntesis de voz (Text-to-Speech). Esta función permite convertir el texto generado durante la conversación en un archivo de audio que puede ser reproducido por el usuario al final de la interacción.

- **text\_to\_speech(text)**

Se creó una clase AzureConnection en Python que maneja la conexión con el servicio de síntesis de voz de Azure. La función principal dentro de esta clase es `text_to_speech(text)`, la cual recibe como parámetro un texto (generado por la ya mencionada función `create_paragraph(messages)`) se conecta a la API de Azure usando la clave de suscripción y la región del servicio, y devuelve el audio generado en formato binario.

La función realiza la conversión del texto a audio de manera asíncrona y, si la conversión es exitosa, retorna los datos de audio. En caso de que ocurra algún error, se maneja el problema de manera adecuada, devolviendo un mensaje de error específico y retornando `None`.

Una vez generado el audio, se almacena en el servidor en un directorio específico dentro de la carpeta `static`, para luego ser accesible y reproducido desde el frontend de la aplicación. Esto se hace utilizando funciones estándar de manejo de archivos en Python.

```
class AzureConnection:
    @staticmethod
    def text_to_speech(text):
        speech_key = "28953521c5604f22909c512ddf41d2f3"
        service_region = "centralus"
        speech_config = speechsdk.SpeechConfig(subscription=speech_key,
        region=service_region)
        synthesizer =
speechsdk.SpeechSynthesizer(speech_config=speech_config)
        result = synthesizer.speak_text_async(text).get()
        if result.reason ==
speechsdk.ResultReason.SynthesizingAudioCompleted:
            return result.audio_data
        else:
            print(f'Text-to-speech conversion failed: {result.reason}')
            return None
```

#### 5.4.2.2 INTEGRACIÓN EN EL PROYECTO

La integración de la API de Azure en el proyecto se llevó a cabo con el objetivo de mejorar y complementar la funcionalidad de generación de historias personalizadas. Este proceso comienza con la recopilación de información durante la interacción con el usuario, donde el sistema, a través de preguntas guiadas, organiza los datos en un formato de texto coherente que refleja la historia o idea que el usuario desea expresar.

Una vez que se ha generado el texto, se utiliza la función `text_to_speech` de la clase `AzureConnection` para convertir este texto en un archivo de audio. Este archivo de audio se guarda en el servidor, y su URL se envía al frontend para que el usuario pueda reproducirlo directamente desde la aplicación. En el código, esto se realiza utilizando `OpenAIConnection.create_paragraph(final_json)` para generar el texto, seguido de `AzureConnection.text_to_speech(paragraph)` para convertirlo en audio. Si el proceso de conversión es exitoso, el archivo de audio se guarda en la ruta especificada (`static/audio/output.wav`) y se escribe en el servidor para su posterior acceso.

```
paragraph = OpenAIConnection.create_paragraph(final_json)
audio_data = AzureConnection.text_to_speech(paragraph)
if audio_data:
    audio_path = os.path.join('static', 'audio', 'output.wav')
    os.makedirs(os.path.dirname(audio_path), exist_ok=True)
    with open(audio_path, 'wb') as audio_file:
        audio_file.write(audio_data)
```

El audio generado se integra en la interfaz de usuario en la pantalla final de la conversación. Al llegar a esta etapa, el sistema reproduce automáticamente el archivo de audio y proporciona al usuario la opción de reiniciar la conversación si lo desea. Este proceso garantiza que la aplicación sea accesible y efectiva para personas con dificultades del habla, ofreciendo una experiencia de usuario fluida y sin barreras. El código correspondiente en la interfaz incluye un contenedor de audio y un botón de reinicio:

```
<div class="audio-restart-container">
  <audio id="audioPlayer" controls class="hidden" autoplay>
    <source src="{{ audio_url }}" type="audio/wav">
    Your browser does not support the audio element.
  </audio>
  <a
    class="restart-button"
    id="newConversation"
    href="/newconversation">
    <strong>{{ language[6] }}</strong>
  </a>
</div>
```

La integración de la API de Azure es crucial para el sistema, ya que convierte el texto generado en una representación audible, mejorando de manera significativa la accesibilidad y la experiencia del usuario en general.

## **Capítulo 6. CONCLUSIONES Y TRABAJOS FUTUROS**

### **6.1 USABILIDAD DE LA APLICACIÓN**

La usabilidad de la aplicación desarrollada es un aspecto crucial para garantizar que los usuarios, especialmente aquellos con dificultades del habla, puedan interactuar con el sistema de manera eficiente y satisfactoria. Durante el desarrollo, se ha priorizado la creación de una interfaz intuitiva y accesible, aunque ciertos desafíos y observaciones han surgido en este proceso.



### **6.1.1 DISEÑO DE LA INTERFAZ DE USUARIO**

La interfaz de la aplicación ha sido diseñada para ser clara y sencilla, con una estructura que permite a los usuarios navegar fácilmente a través de las diferentes funciones. Como ya se ha explorado anteriormente, se han empleado tecnologías como HTML, CSS y Bootstrap para asegurar que la interfaz sea responsiva y se adapte a distintos tamaños de pantalla y dispositivos.

Sin embargo, durante las pruebas de usuario, se identificaron áreas donde la experiencia podría mejorar. Por ejemplo, algunos usuarios reportaron que ciertas funciones, como la configuración del idioma, no eran lo suficientemente intuitivas, lo que indica que hay margen para optimizar la organización de los elementos de la interfaz.

### **6.1.2 FLUIDEZ DE LA EXPERIENCIA DEL USUARIO**

La aplicación guía a los usuarios a través de un flujo conversacional que recopila información para generar historias de audio personalizadas. Este proceso ha sido diseñado para ser lo más natural y directo posible, minimizando la necesidad de interacción complicada.

No obstante, la latencia en las respuestas debido a la dependencia de APIs externas, como OpenAI y Azure, ha afectado la fluidez en algunos casos. Esto puede causar demoras perceptibles entre la entrada del usuario y la generación de respuestas o archivos de audio, lo que podría impactar negativamente la experiencia del usuario, especialmente en conexiones a internet más lentas.

### **6.1.3 ACCESIBILIDAD**

Se ha puesto un enfoque significativo en asegurar que la aplicación cumpla con los estándares de accesibilidad, facilitando el uso para personas con diferentes discapacidades.

Elementos como el contraste de color, la navegación por teclado y la compatibilidad con lectores de pantalla han sido considerados en el diseño.

A pesar de estos esfuerzos, es posible que algunos usuarios con discapacidades más complejas aún encuentren dificultades al interactuar con la aplicación. Por ejemplo, la ausencia de funcionalidad de reconocimiento de voz (speech to text) limita la accesibilidad para usuarios que podrían beneficiarse de esta característica.

#### **6.1.4 PRUEBAS DE USABILIDAD**

Se realizaron pruebas de usabilidad con un grupo reducido de usuarios, que ayudaron a identificar tanto fortalezas como áreas de mejora en la aplicación. Los comentarios recibidos indicaron que, si bien la aplicación es generalmente fácil de usar, existen oportunidades para hacerla aún más accesible y eficiente.

Las pruebas también revelaron que los usuarios apreciaron la simplicidad del diseño, pero sugirieron que algunas funciones podrían beneficiarse de una mayor personalización y control por parte del usuario. Estas observaciones serán fundamentales para futuras iteraciones y mejoras de la aplicación.

## **6.2 LIMITACIONES**

Durante el desarrollo de la aplicación, se han identificado ciertas limitaciones que podrían haber influido en el alcance y los resultados obtenidos. Estas limitaciones no solo reflejan los desafíos técnicos y operativos enfrentados, sino que también proporcionan un contexto crucial para comprender el rendimiento y las capacidades actuales de la aplicación. A continuación, se describen las principales limitaciones encontradas, con el objetivo de reconocer áreas donde el proyecto podría beneficiarse de mejoras o ajustes en el futuro.

### **6.2.1 DEPENDENCIA DE APIS EXTERNAS**

Una de las principales limitaciones del proyecto es la fuerte dependencia de las APIs externas, las proporcionadas por OpenAI para la generación de texto y Azure para la síntesis de voz. Aunque estas herramientas han sido fundamentales para el éxito del proyecto, su uso conlleva ciertos riesgos y restricciones.

La principal preocupación radica en la disponibilidad y estabilidad de estos servicios. Cualquier cambio en las políticas de acceso, costos o en la estabilidad de las APIs podría afectar directamente la funcionalidad de la aplicación. Por ejemplo, un aumento en los costos de las APIs podría hacer que el mantenimiento de la aplicación sea insostenible a largo plazo sin un ajuste en el presupuesto o en el modelo de negocio.

Además, la latencia y la velocidad de respuesta de las APIs pueden variar dependiendo de la carga del servicio o de la calidad de la conexión a internet del usuario, lo que podría afectar negativamente la experiencia del usuario. Este tipo de dependencia también significa que cualquier interrupción del servicio por parte del proveedor puede resultar en una funcionalidad limitada o en la inoperabilidad total de la aplicación durante el tiempo que dure la interrupción.

### **6.2.2 LATENCIA EN LA RESPUESTA**

Otro desafío significativo identificado durante el desarrollo de la aplicación es la latencia en la respuesta, especialmente en las interacciones que dependen de las APIs externas para la generación de texto y síntesis de voz. Aunque se ha trabajado para optimizar la comunicación entre la aplicación y estos servicios, existen momentos en los que los tiempos de respuesta son perceptiblemente lentos.

Esta latencia puede tener diversas causas, incluyendo la calidad de la conexión a internet del usuario, la carga en los servidores de las APIs externas, y la complejidad de las solicitudes enviadas. En particular, usuarios con conexiones a internet más lentas pueden

experimentar demoras que afectan la fluidez de la interacción, lo que podría resultar en una experiencia de usuario menos satisfactoria.

El impacto de esta limitación es especialmente relevante en aplicaciones donde la inmediatez y la fluidez de las respuestas son cruciales para mantener una experiencia de usuario positiva. En un entorno donde la aplicación es utilizada por personas con dificultades del habla, cualquier retraso adicional puede aumentar la frustración y dificultar el uso efectivo de la herramienta.

### **6.2.3 ESCALABILIDAD**

La escalabilidad de la aplicación es otra limitación significativa que debe considerarse, especialmente si se proyecta un aumento en el número de usuarios o en la cantidad de datos manejados por el sistema. Actualmente, la arquitectura de la aplicación ha sido diseñada para manejar un volumen moderado de usuarios y datos, lo que podría no ser suficiente si la aplicación comienza a ser utilizada por un número mucho mayor de personas.

La infraestructura actual, que incluye la base de datos y los servidores, podría experimentar un rendimiento reducido o incluso fallos si se enfrenta a una carga significativamente mayor. Esto podría resultar en tiempos de respuesta más lentos, errores en el procesamiento de solicitudes, o incluso caídas del sistema en casos extremos.

Además, la falta de un sistema robusto para manejar un aumento en la carga de trabajo significa que, sin una reconfiguración o mejora significativa de la infraestructura, la aplicación podría no ser capaz de soportar la expansión o la implementación a gran escala. Esto representa una barrera importante para el crecimiento y la adopción más amplia de la aplicación.

#### **6.2.4 LIMITACIÓN EN LA PERSONALIZACIÓN**

La aplicación, en su estado actual, ofrece opciones limitadas para la personalización de la experiencia de usuario, lo que puede ser una restricción importante para algunos usuarios que buscan adaptar la interfaz y las funcionalidades a sus necesidades y preferencias específicas.

Por ejemplo, aunque la aplicación permite seleccionar el idioma y ajustar configuraciones básicas, no proporciona opciones avanzadas de personalización, como la posibilidad de modificar la apariencia de la interfaz, ajustar el tamaño de la fuente, o cambiar los colores para mejorar la accesibilidad. Además, las opciones para personalizar la interacción con el asistente de inteligencia artificial, como la configuración del tono o estilo de las respuestas generadas, también son limitadas.

Esta falta de flexibilidad puede hacer que la aplicación no satisfaga completamente las expectativas o necesidades de todos los usuarios, particularmente aquellos con discapacidades que requieren ajustes específicos en la interfaz o en la interacción. La personalización es clave para garantizar una experiencia inclusiva y satisfactoria, especialmente en aplicaciones diseñadas para usuarios con diversas necesidades.

#### **6.2.5 PRUEBAS DE USABILIDAD LIMITADAS**

Otra limitación significativa del proyecto es que las pruebas de usabilidad realizadas hasta ahora han sido limitadas en alcance y diversidad. Aunque se han llevado a cabo pruebas con un grupo reducido de usuarios, estas no han abarcado una gama suficientemente amplia de perfiles, dispositivos y escenarios de uso, lo que podría dejar sin identificar ciertos problemas potenciales en la experiencia del usuario.

Las pruebas realizadas han proporcionado información valiosa para ajustar y mejorar la interfaz y las funcionalidades, pero el tamaño reducido de la muestra y la falta de diversidad en los perfiles de usuarios significa que es posible que no se hayan detectado

todas las posibles dificultades que podrían surgir en condiciones de uso reales y variadas. En particular, no se ha realizado un análisis exhaustivo de cómo la aplicación funciona en diferentes dispositivos, navegadores o entornos de red, lo que podría revelar problemas de compatibilidad o rendimiento no anticipados.

Además, la aplicación no ha sido probada de manera intensiva con usuarios que tienen diversas discapacidades, lo que significa que algunas barreras de accesibilidad podrían haber pasado desapercibidas. La falta de pruebas más amplias limita la capacidad de garantizar que la aplicación sea verdaderamente inclusiva y eficiente para todos los posibles usuarios.

## **6.3 TRABAJOS FUTUROS**

El desarrollo de esta aplicación ha sido un primer paso significativo hacia la creación de una herramienta accesible y útil para personas con dificultades del habla. Sin embargo, como ocurre con cualquier proyecto tecnológico, siempre hay margen para mejorar y expandir sus capacidades. En esta sección, se delinearán una serie de posibles trabajos futuros que podrían abordar las limitaciones actuales, incorporar nuevas funcionalidades y optimizar la experiencia del usuario. Estas propuestas están orientadas no solo a mejorar el rendimiento y la accesibilidad de la aplicación, sino también a ampliar su alcance y su utilidad en un contexto más amplio.

### **6.3.1 PERSONALIZACIÓN DE LA VOZ**

Una de las mejoras más significativas que se podría implementar en futuras versiones de la aplicación es la personalización avanzada de la voz generada. Actualmente, la síntesis de voz ofrece una experiencia básica que, aunque funcional, carece de opciones para ajustar aspectos como el tono, la velocidad, la entonación, el estilo de la voz o incluso hacer que el

texto generado se escuchase con su propia voz. La capacidad de personalizar estos parámetros permitiría a los usuarios adaptar la salida de voz a sus preferencias personales o necesidades específicas, mejorando significativamente la experiencia de uso.

Durante el desarrollo del proyecto, se investigó la posibilidad de utilizar la tecnología de Azure, que ya está integrada en la aplicación, para ofrecer opciones avanzadas de personalización de la voz. Aunque introducir esta funcionalidad no habría supuesto un esfuerzo técnico considerable, se decidió no implementarla en esta fase debido al coste asociado con el uso extendido de estas capacidades. Sin embargo, en futuras versiones, esta función podría ser introducida como una opción premium, permitiendo a los usuarios acceder a una personalización más profunda de la voz a cambio de un pequeño costo adicional. Esto no solo cubriría el costo de la tecnología, sino que también podría generar un beneficio económico extra para el mantenimiento y desarrollo continuo de la aplicación.

### **6.3.2 PUBLICAR LA APLICACIÓN Y SERVIDOR PARA LA BASE DE DATOS**

La primera mejora para realizar sería la publicación de la aplicación en línea y la contratación de un servidor dedicado para la gestión de la base de datos. Actualmente, la aplicación opera de manera local, lo que limita su accesibilidad y utilidad, ya que los usuarios necesitan acceder a ella desde un entorno específico. Publicar la aplicación permitiría que un mayor número de usuarios la utilice desde cualquier lugar y dispositivo, expandiendo su impacto y alcance.

La principal razón por la cual esta mejora no se ha implementado hasta ahora es el costo asociado. La contratación de un servidor dedicado, junto con los gastos recurrentes de mantenimiento, representan una inversión significativa que no se puede justificar sin una estrategia clara de monetización. Sin embargo, al publicar la aplicación, sería posible introducir un modelo de suscripción o tarifas de uso que no solo cubrirían estos costos, sino que también podrían generar ingresos adicionales para el proyecto.

En un futuro cercano, esta medida no solo ampliaría la accesibilidad de la aplicación, sino que también abriría la puerta a nuevas oportunidades económicas, permitiendo reinvertir en el desarrollo y la mejora continua de la aplicación.

### **6.3.3 INCLUIR MÁS IDIOMAS**

En la fase actual del proyecto, se han incluido los idiomas que eran factibles según los recursos disponibles y las necesidades inmediatas de los usuarios objetivo. Sin embargo, la ambición a futuro es expandir significativamente la cantidad de idiomas soportados por la aplicación, permitiendo que un público aún más diverso pueda beneficiarse de sus funcionalidades.

La inclusión de más idiomas no solo aumentaría la accesibilidad global de la aplicación, sino que también respondería a la demanda de usuarios de diferentes regiones que podrían encontrar útil esta herramienta. Aunque en esta fase inicial se ha priorizado la integración de los idiomas más utilizados, el plan a largo plazo es seguir ampliando esta oferta lingüística, conforme a la disponibilidad de recursos y la viabilidad técnica.

Esta expansión implicará no solo traducir la interfaz de usuario, sino también garantizar que las tecnologías de procesamiento de lenguaje natural y síntesis de voz sean igualmente efectivas en cada nuevo idioma. La implementación de esta mejora hará que la aplicación sea más inclusiva y adaptable a un contexto global, lo que podría abrir nuevas oportunidades de crecimiento y expansión del mercado.

### **6.3.4 RESPUESTAS PERSONALIZADAS EN EL FORO**

Otra mejora importante que se planea implementar es la capacidad de que los usuarios puedan responder directamente a otros usuarios dentro del foro de la aplicación. Actualmente, el foro permite a los usuarios publicar comentarios de manera general, pero no existe la opción de crear hilos de conversación o responder directamente a un comentario



específico. Esta limitación reduce la interactividad y la cohesión de las discusiones, haciendo que las conversaciones sean menos dinámicas y organizadas.

Incorporar la funcionalidad de respuestas directas no solo mejoraría la organización del contenido en el foro, sino que también fomentaría una mayor participación y colaboración entre los usuarios. Esta mejora haría que las discusiones fueran más fluidas y enfocadas, permitiendo a los usuarios interactuar de manera más significativa.

Además, esta característica podría atraer a más usuarios interesados en una comunidad interactiva y comprometida, lo que, a su vez, podría aumentar la retención de usuarios y el valor de la aplicación como plataforma social. En futuras versiones, la implementación de respuestas directas en el foro será un paso clave para mejorar la experiencia de usuario y fortalecer la comunidad dentro de la aplicación.

### **6.3.5 TRADUCCIÓN DE COMENTARIOS EN EL FORO**

Una mejora que se planea implementar en futuras versiones de la aplicación es la traducción de las publicaciones en el foro. Actualmente, las publicaciones realizadas por los usuarios permanecen en el idioma original en que fueron escritas, lo que puede limitar la comprensión y la interacción entre usuarios que hablan diferentes idiomas.

Un posible enfoque para implementar esta funcionalidad sería utilizar la API externa de OpenAI. Con este método, cada comentario que se desee traducir se enviaría a la API, que se encargaría de generar la traducción al idioma deseado. Esta solución permitiría una traducción automática y eficiente, facilitando la comunicación entre usuarios de distintas lenguas y reduciendo significativamente las barreras lingüísticas dentro del foro.

La implementación de esta funcionalidad no solo promovería una mayor interacción entre usuarios de diferentes regiones, sino que también enriquecería las discusiones, permitiendo que una diversidad más amplia de perspectivas sea compartida y comprendida por todos. Esta mejora es esencial para crear un entorno verdaderamente inclusivo donde

todos los usuarios, independientemente de su idioma nativo, puedan participar plenamente en las discusiones del foro.

### **6.3.6 GESTIÓN DE NOTICIAS**

Otra mejora importante que se planea implementar es la designación de una persona encargada de gestionar las noticias que se publican en la aplicación. Actualmente, la actualización y el manejo del contenido de noticias no están centralizados, lo que puede llevar a una falta de coherencia, relevancia y actualización oportuna del contenido presentado a los usuarios.

Al asignar a una persona o equipo específico para esta tarea, se aseguraría que las noticias publicadas sean siempre pertinentes, actuales y alineadas con los intereses de la comunidad de usuarios, pudiendo incluso a llegar a crear un periódico interno sin tener que acudir a noticias externas. Esta gestión centralizada permitiría una curación del contenido que no solo mantendría a los usuarios informados sobre temas relevantes, sino que también podría aumentar su compromiso con la aplicación al ofrecerles contenido de alta calidad de manera regular.

Además, esta mejora abriría la posibilidad de monetizar la sección de noticias, ya sea a través de contenido patrocinado o mediante la promoción de productos y servicios relacionados con los temas tratados. De esta manera, la gestión centralizada de las noticias no solo mejoraría la experiencia del usuario, sino que también podría generar ingresos adicionales para el sostenimiento y crecimiento de la aplicación.

### **6.3.7 INTEGRACIÓN DE RECONOCIMIENTO DE VOZ**

Una mejora clave que se planea implementar en la aplicación es la integración de una funcionalidad de reconocimiento de voz (speech-to-text). Esta característica permitiría a los usuarios interactuar con la aplicación hablando directamente, en lugar de tener que escribir

sus respuestas o comandos. Esto sería particularmente beneficioso para personas con dificultades para escribir o que prefieren la comunicación verbal, aumentando así la accesibilidad y comodidad de la aplicación.

Durante el desarrollo, se exploró la idea de utilizar tecnologías avanzadas como OpenAI Whisper para implementar esta funcionalidad. Sin embargo, debido a diversas limitaciones técnicas y de recursos, no se logró desarrollar esta característica en la versión actual. En el futuro, se podría reconsiderar este enfoque o explorar nuevamente el uso de otras tecnologías como Azure Speech to Text, para ofrecer una experiencia más inclusiva y eficiente para los usuarios.

Además, la aplicación ya está diseñada para integrar esta funcionalidad de manera efectiva, ya que la conversación guiada por OpenAI está estructurada para hacer preguntas que pueden ser respondidas con una o dos palabras. Este diseño tiene en cuenta las dificultades del habla y busca facilitar al máximo la interacción del usuario, asegurando que incluso aquellos con limitaciones en la expresión verbal puedan participar sin dificultad.

La implementación de esta funcionalidad no solo mejoraría la usabilidad de la aplicación, sino que también podría ampliar su atractivo, atrayendo a un público más amplio que busca herramientas más intuitivas y adaptadas a sus necesidades.

## **6.4 CONCLUSIONES**

El desarrollo de la aplicación "IA para la Accesibilidad" ha sido una experiencia profundamente enriquecedora que ha permitido no solo la creación de una herramienta útil para personas con dificultades del habla, sino también un proceso de aprendizaje integral en diversas áreas del desarrollo tecnológico. A lo largo de este proyecto, se han abordado múltiples aspectos del desarrollo web, la integración de APIs externas, el manejo de bases de datos, y se ha profundizado en el entendimiento y la aplicación de tecnologías emergentes como la inteligencia artificial.

Uno de los principales aprendizajes de este proyecto ha sido la integración de APIs externas. A través de la implementación de servicios como OpenAI para la generación de texto y Azure para la síntesis de voz, se ha demostrado cómo las APIs pueden expandir significativamente las capacidades de una aplicación. Este proceso me ha permitido darme cuenta y apreciar que no es necesario empezar todo desde cero, ya que, gracias a las APIs, es posible acceder e implementar herramientas sumamente útiles que ya han sido desarrolladas por otros. Estas herramientas permiten a los desarrolladores acceder a tecnologías avanzadas sin la necesidad de desarrollarlas desde cero, lo cual es fundamental para crear soluciones innovadoras de manera eficiente. Sin embargo, también se ha aprendido sobre las limitaciones y los desafíos asociados a esta dependencia, como los costos y la latencia, y cómo estos factores deben ser considerados cuidadosamente en el diseño y mantenimiento de una aplicación.

Un aspecto crucial del proyecto ha sido el aprendizaje y la aplicación de tecnologías de inteligencia artificial. En un contexto donde la IA a menudo se percibe con recelo o desconfianza, este proyecto ha demostrado cómo estas tecnologías pueden ser utilizadas como herramientas poderosas para mejorar la calidad de vida de las personas. La integración de la IA no solo ha permitido la creación de historias de audio personalizadas, sino que también ha abierto un nuevo abanico de posibilidades para futuras mejoras y expansiones de la aplicación. Este proyecto ha reafirmado la idea de que no debemos temer a las nuevas tecnologías, sino más bien adoptarlas y aprender a utilizarlas de manera ética y responsable para resolver problemas reales.

En resumen, el desarrollo del proyecto ha sido un viaje de aprendizaje continuo y aplicación práctica. Se ha obtenido un conocimiento profundo sobre la integración de APIs, el desarrollo web, la gestión de bases de datos y la implementación de tecnologías de inteligencia artificial. Este proyecto no solo ha resultado en una herramienta valiosa para sus usuarios, sino que también ha servido como un puente para el desarrollador hacia el entendimiento y aprovechamiento de las tecnologías emergentes. En el futuro, el conocimiento adquirido en este proyecto será invaluable para afrontar nuevos desafíos y continuar innovando en el campo de la tecnología accesible.

## Capítulo 7. BIBLIOGRAFÍA

- [1] A. S.-L.-H. A. (ASHA), «Impact of Speech and Language Disorders on Quality of Life.,» 2020. [En línea]. Available: <https://www.asha.org>.
- [2] S. W. Y. B. I. G. S. & L. G. Roulstone, The preferred outcomes of children with speech, language and communication needs and their parents., Department for Education., 2012.
- [3] S. & M. D. H. McLeod, International Journal of Language & Communication Disorders., 2007.

- [4] D. N. C. A. & P. B. B. Bryen, *Journal of Vocational Rehabilitation.*, 2007.
- [5] D. & M. J. H. Jurafsky, *Speech and Language Processing (3rd ed.)*, Pearson, 2019.
- [6] I. B. Y. & C. A. Goodfellow, «Deep Learning,» *MIT Press*, 2016.
- [7] D. S. J. S. K. A. I. H. A. G. A. .. & Silver, «Mastering the game of Go without human knowledge,» *Nature*, 2017, pp. 354-359.
- [8] A. D. S. Z. H. S. K. V. O. G. A. Van den Oord, «Kavukcuoglu, K.,» de *WaveNet: A generative model for raw audio.*, 2016.
- [9] J. Nielsen, «Usability engineering,» Elsevier, 1994.
- [10] G. & D. F. L. Van Rossum, *Python 3 Reference Manual.*, CreateSpace, 2009.
- [11] W3C, «World Wide Web,» de *Web Content Accessibility Guidelines (WCAG) 2.1.* , 2018.
- [12] R. Gellman, «Available at SSRN,» de *Fair information practices: A basic history.*, 2017.
- [13] F. M. C. R. R. S. & v. d. L. H. K. Ramus, de *Phonological deficits in specific language impairment and developmental dyslexia: Towards a multidimensional model. Brain*, 2013, pp. 136(3), 630-645.
- [14] M. B. Pereira, de *Speech and language development in children with Down*, 2015, pp. 58, 1-10.
- [15] R. A. H. R. C. G. T. T. C. D. & V. T. Marrie, de *The burden of mental comorbidity in multiple sclerosis: Frequent, underdiagnosed, and undertreated. Multiple Sclerosis Journal*, 2015, pp. 21(3), 305-312.

- [16] M. T. C. & F. O. Bax, de *Clinical and MRI correlates of cerebral*, JAMA, 2005, pp. 293(6), 705-714.
- [17] R. L. P. M. P. A. J. L. & J. K. Tate, de *Predicting clinically significant cognitive problems after traumatic brain injury. Journal of Neurology, Neurosurgery & Psychiatry*, 2014, pp. 85(10), 1150-1157.
- [18] N. N. E. J. D. & B. D. Miller, de *Hard to swallow: Dysphagia in*, 2006, pp. 35(6), 614-618.
- [19] A. Rosenblatt, de *Huntington's disease: A clinical review. Neurologist*, 2007, pp. 13(4), 2341.
- [20] A. Kertesz, de *What do we learn from closed-head injury? Brain Injury*, 1988, pp. 2(3), 191-208.
- [21] H. Duffau, de *The huge plastic potential of adult brain and the role of connectomics: New insights provided by serial mappings in glioma surgery. Cprtex*, 2014, pp. 58, 325-337.
- [22] AssistiveWare, «Proloquo2Go,» 2024. [En línea]. Available: <https://www.assistiveware.com/products/proloquo2go>.
- [23] TalkTablet, «TalkTablet,» 2024. [En línea]. Available: <https://usaspeechtablets.com/pages/talktabletpro>.
- [24] CoughDrop, «CoughDrop,» 2024. [En línea]. Available: <https://www.coughdrop.com/>.
- [25] Nuance, «Dragon NaturallySpeaking,» 2024. [En línea]. Available: <https://www.nuance.com/dragon.html>.

- [26] Otter.ai., «Otter.ai.,» 2024. [En línea]. Available: <https://otter.ai/>.
- [27] IBM, «IBM Watson Text to Speech,» 2024. [En línea]. Available: <https://www.ibm.com/products/text-to-speech>.
- [28] Amazon, «Amazon Polly,» 2024. [En línea]. Available: <https://aws.amazon.com/es/polly/>.
- [29] T. B. Ltd., «Predictablej,» 2024. [En línea]. Available: <https://therapy-box.co.uk/predictable-english>.
- [30] OpenAI, «OpenAI Whisper,» 2023. [En línea]. Available: <https://openai.com/index/whisper/>.
- [31] Microsoft, «Azure Speech-to-Text,» 2023. [En línea]. Available: <https://azure.microsoft.com/en-us/products/ai-services/speech-to-text/>.
- [32] OpenAI, «OpenAI Text to Speech,» 2024. [En línea]. Available: <https://platform.openai.com/docs/guides/text-to-speech>.
- [33] Microsoft, «Azure Text-to-Speech,» 2023. [En línea]. Available: <https://azure.microsoft.com/en-us/products/ai-services/text-to-speech>.
- [34] I. Anaconda, «Jupyter Notebooks,» 2024. [En línea]. Available: <https://www.anaconda.com/download>.
- [35] Microsoft, «Visual Studio Code,» 2024. [En línea]. Available: <https://code.visualstudio.com/>.
- [36] I. GitHub, «GitHub,» 2024. [En línea]. Available: <https://github.com/>.



- [37] G. & D. F. L. Van Rossum, Python 3 Reference Manual, Python 3 Reference Manual, 2009.
- [38] W3C, «HTML5: A vocabulary and associated APIs for HTML and XHTML.,» World, 2014. [En línea]. Available: <https://html.spec.whatwg.org/multipage/>.
- [39] IBM, «What is an API (Application Programming Interface)?,» 2024. [En línea]. Available: <https://www.ibm.com/cloud/learn/api>.
- [40] OpenAI, «OpenAI API,» 2024. [En línea]. Available: <https://platform.openai.com/docs/api-reference/introduction>.
- [41] P. P. (n.d.), «Flask documentation,» [En línea]. Available: <https://flask.palletsprojects.com/en/latest/>.
- [42] T. S. A. (n.d.), «SQLAlchemy documentation,» [En línea]. Available: <https://www.sqlalchemy.org/>.
- [43] M. O. y. J. Thornton., « Bootstrap. GitHub,» 2011. [En línea]. Available: <https://getbootstrap.com/>.
- [44] F. D., JavaScript: The Definitive Guide (7th ed.), O'Reilly Media, 2020.

# **ANEXO I: ALINEACIÓN DEL PROYECTO CON LOS ODS**

- **ODS 3: Salud y Bienestar**

El ODS 3 se centra en garantizar una vida sana y promover el bienestar para todos en todas las edades. La aplicación desarrollada contribuye directamente a este objetivo al proporcionar una herramienta que mejora la comunicación para personas con dificultades del habla. Esto no solo mejora su calidad de vida al facilitar su integración social, sino que también apoya su bienestar emocional y mental, al ofrecerles una manera efectiva de expresarse y ser comprendidos por los demás.

- **ODS 4: Educación de Calidad**

El ODS 4 busca garantizar una educación inclusiva, equitativa y de calidad, y promover oportunidades de aprendizaje durante toda la vida para todos. La aplicación puede ser utilizada como una herramienta educativa para personas con trastornos del habla, facilitando su participación en entornos de aprendizaje y mejorando su capacidad de comunicación. Al ayudar a estas personas a superar las barreras del habla, la aplicación contribuye a una educación más inclusiva y equitativa.

- **ODS 10: Reducción de las Desigualdades**

El ODS 10 tiene como objetivo reducir las desigualdades dentro y entre los países. Este proyecto promueve la inclusión social al ofrecer una herramienta accesible que ayuda a las personas con discapacidades del habla a interactuar de manera más efectiva en la sociedad. Al reducir las barreras de comunicación, se fomenta la igualdad de oportunidades y se combate la discriminación basada en la capacidad de comunicación.

- **ODS 9: Industria, Innovación e Infraestructura**

El ODS 9 promueve la construcción de infraestructuras resilientes, la promoción de la industrialización inclusiva y sostenible, y el fomento de la innovación. Este proyecto se alinea con este objetivo al desarrollar una solución innovadora basada en inteligencia artificial que aprovecha la infraestructura digital existente para abordar un problema social crítico. La integración de tecnologías avanzadas como las APIs de OpenAI y Azure demuestra el potencial de la innovación para mejorar la accesibilidad y el bienestar de las personas.

- **ODS 17: Alianzas para Lograr los Objetivos**

El ODS 17 busca fortalecer los medios de implementación y revitalizar la alianza mundial para el desarrollo sostenible. Este proyecto refleja la importancia de las alianzas entre diferentes tecnologías y plataformas para crear una solución efectiva y escalable. Al integrar APIs externas y colaborar con tecnologías desarrolladas por otras entidades, se demuestra

cómo la colaboración puede llevar a soluciones más efectivas y sostenibles para los desafíos globales.

## **ANEXO II: REPOSITORIO DEL PROGRAMA**

Enlace a repositorio en GitHub:

[https://github.com/sergiorodriguez/tfg\\_code](https://github.com/sergiorodriguez/tfg_code)