

**PronunciApp: Prototipo de aplicacion web para la mejora de la pronunciación del inglés
en los estudiantes de la carrera de Ingeniería de sistemas de la Universidad del Valle, sede
Tuluá**

Daniel Felipe Casallas Ortiz

**Universidad del Valle
Facultad de ingeniería
Escuela de ingeniería de sistemas y computación
Tuluá, Valle del Cauca
2025**

**PronunciApp: Prototipo de aplicacion web para la mejora de la pronunciación del inglés
en los estudiantes de la carrera de Ingeniería de sistemas de la Universidad del Valle, sede
Tuluá**

Daniel Felipe Casallas Ortiz
Código 2059974
daniel.casallas@correounivalle.edu.co

Director
Mg. Joshua David Triana Madrid
Profesor de la Escuela de Ingeniería de Sistemas y Computación
joshua.triana@correounivalle.edu.co

Co-director
Mg. Ronald Esteban Casallas Restrepo
Profesor de la Escuela de Ingeniería de Sistemas y Computación
ronald.casallas@correounivalle.edu.co

Universidad del Valle
Facultad de ingeniería
Escuela de ingeniería de sistemas y computación
Tuluá, Valle del Cauca
2025

Trabajo de grado presentado por
Daniel Felipe Casallas Ortiz
Como requisito parcial para la obtención del título de Ingeniero de Sistemas

Joshua David Triana Madrid
Director

Jurado

Jurado

A Dios por bendecirme con la oportunidad de alcanzar esta meta importante.
A mi familia por su apoyo incondicional, cariño y fe constante en mis capacidades.
A mis docentes por compartir sus conocimientos y contribuir a mi formación académica.
Al Ingeniero Joshua Triana por su valiosa orientación y acompañamiento en este proyecto.

Tabla de Contenido

1. Introducción	1
1.1. Formulación del problema	2
1.2. Descripción del problema	2
1.3. Definición del problema	3
1.4. Objetivos	4
1.4.1. Objetivo general	4
1.4.2. Objetivos específicos	4
1.5. Resultado esperado	4
1.6. Antecedentes	6
1.7. Justificación	7
1.8. Información sobre los capítulos	8
2. Alcance de la propuesta	11
2.1. Declaración del alcance	11
3. Marco referencial	13
3.1. Conceptos Básicos	13
3.1.1. Desarrollo de aplicaciones web:	13
3.1.2. Conceptos de síntesis y clonación de voz:	13
3.1.3. Tecnologías de aprendizaje profundo:	13
3.2. Marco teórico	14
3.2.1. Fundamentos del aprendizaje de idiomas y pronunciación	14
3.2.2. Inteligencia Artificial en la Educación	15
3.2.3. Procesamiento del Lenguaje Natural (PLN)	16
3.3. Marco conceptual	16
3.3.1. Teorías del aprendizaje lingüístico:	17
3.3.2. Enfoques pedagógicos para la enseñanza de la pronunciación:	17
3.3.3. Tecnologías para la mejora de la pronunciación:	17
4. Metodología	18
4.1. Metodología de investigación	18
4.2. Metodología de desarrollo de software	19
4.3. Metodología de gestión de actividades	20
5. Revisión de literatura	21
5.1. Metodología de la Revisión	21
5.2. Fundamentos teóricos del reconocimiento de voz	21
5.3. Tecnologías emergentes en síntesis y análisis de voz	22

5.4. Análisis comparativo de plataformas existentes	23
6. Gestión de proyecto con tablero Kanban	25
6.1. Fundamentos teóricos del tablero Kanban	25
6.2. Configuración del tablero Kanban en GitHub Projects	25
6.3. Organización temporal: sprints semanales	26
6.4. Gestión de los 40 sprints del proyecto	27
6.5. Métricas y análisis del flujo de trabajo	29
6.6. Beneficios observados de la implementación	30
6.7. Integración con GitHub y control de versiones	30
7. Diseño del prototipo	32
7.1. Objetivo del diseño	32
7.2. Metodología de diseño	32
7.2.1. Herramientas utilizadas	32
7.2.2. Proceso de diseño	33
7.3. Principios de UX/UI aplicados	33
7.3.1. Principios de experiencia de usuario (UX)	33
7.3.2. Principios de interfaz de usuario (UI)	34
7.4. Arquitectura de la interfaz	34
7.4.1. Estructura principal	34
7.4.2. Flujo de usuario simplificado	34
7.5. Decisiones de diseño clave	35
7.5.1. Reducción de carga informativa	35
7.5.2. Integración de tecnología de voz	35
7.6. Prototipo final	35
7.6.1. Pantalla principal	35
7.6.2. Características del diseño	36
7.7. Iteraciones del diseño	36
7.7.1. Versión inicial vs. versión final	36
7.7.2. Justificación de los cambios	37
7.8. Validación del diseño	37
7.8.1. Criterios de éxito	37
7.8.2. Métricas de usabilidad	38
7.9. Implementación técnica	38
7.9.1. Arquitectura de componentes	38
7.9.2. Integración de APIs	39
8. Desarrollo del prototipo	40
8.1. Arquitectura de la solución	40
8.1.1. Visión general	41
8.1.2. Componentes principales	41
8.2. Desarrollo del backend	41
8.2.1. Selección de framework: FastAPI	41
8.2.2. Integración del modelo VALL-E-X	42
8.2.3. Estructura del backend	43
8.3. Desarrollo del frontend	45
8.3.1. Selección de framework: Angular 19	45

8.3.2.	Arquitectura y estructura del frontend	46
8.3.3.	Funcionalidades implementadas	46
8.3.4.	Principios de UX/UI aplicados	48
8.4.	Infraestructura y despliegue en Azure	48
8.4.1.	Azure Virtual Machine	48
8.4.2.	Estrategia de despliegue	49
8.4.3.	Configuración de contenedores	49
8.5.	Cumplimiento de requisitos	50
8.5.1.	Requisitos funcionales	50
8.5.2.	Requisitos no funcionales	50
8.6.	Ventajas de la arquitectura implementada	50
8.6.1.	Escalabilidad	50
8.6.2.	Mantenibilidad	50
8.6.3.	Rendimiento	51
8.7.	Desafíos y soluciones	51
8.7.1.	Integración del modelo VALL-E-X	51
8.7.2.	Optimización de recursos	51
8.7.3.	Latencia de red	51
8.8.	Resultados obtenidos	51
9.	Pruebas del prototipo	53
9.1.	Metodología de pruebas	53
9.1.1.	Enfoque de pruebas	53
9.1.2.	Herramientas utilizadas	53
9.2.	Pruebas técnicas	54
9.2.1.	Pruebas del backend	54
9.2.2.	Pruebas del frontend	56
9.3.	Evaluación de usabilidad	57
9.3.1.	Encuesta de uso de PronunciApp	57
9.3.2.	Encuesta de usabilidad basada en heurísticas de Jakob Nielsen	57
9.4.	Resultados de las pruebas	58
9.4.1.	Resultados de pruebas técnicas	58
9.4.2.	Resultados de usabilidad	59
9.5.	Análisis de resultados	60
9.5.1.	Análisis estadístico detallado	60
9.5.2.	Análisis cuantitativo de la encuesta de uso	62
9.5.3.	Análisis de las heurísticas de Nielsen	62
9.5.4.	Correlación entre encuestas	63
9.5.5.	Fortalezas identificadas	64
9.5.6.	Áreas de mejora prioritarias	64
9.5.7.	Impacto de los resultados	65
9.5.8.	Análisis específico de problemas detectados	65
9.5.9.	Validación estadística de resultados	66

<i>TABLA DE CONTENIDO</i>	6
10.Conclusiones y trabajos futuros	68
10.1. Conclusiones	68
10.2. Trabajos futuros	69
10.2.1. Mejoras tecnológicas	69
10.2.2. Mejoras pedagógicas	69
10.2.3. Validación y evaluación	70
11.Bibliografía	71

Lista de Figuras

1.1. Representación gráfica de la propuesta.	2
1.2. Árbol de objetivos.	6
6.1. Configuración de estados en el tablero Kanban de GitHub Projects	26
6.2. Sistema de priorización P1, P2 y P3 implementado en GitHub Projects	26
6.3. Vista de planificación semanal en GitHub Projects	27
6.4. Integración del tablero Kanban con el repositorio de GitHub (Backend)	31
6.5. Integración del tablero Kanban con el repositorio de GitHub (Frontend)	31
7.1. Proceso de diseño en Figma - Wireframe	33
7.2. Interfaz final del prototipo de PronunciApp (Vista PC)	36
7.3. Interfaz final del prototipo de PronunciApp (Vista Moviles)	37
7.4. Resultado final implementado de PronunciApp mostrando la funcionalidad completa	38
8.1. Resumen de la arquitectura de la solución. Fuente: Elaboración propia	40
8.2. Documentación automática de la API generada por FastAPI	42
8.3. Estructura y arquitectura del backend con FastAPI	43
8.4. Interfaz principal de PronunciApp	47
8.5. Interfaz de grabación de audio para práctica de pronunciación	48
8.6. Configuración de contenedores Docker en Azure	49

Lista de tablas

1.1. Objetivos específicos y sus resultados esperados	4
1.2. Estructura de capítulos	9
4.1. Comparación de metodologías	19
5.1. Comparación de plataformas CAPT y tecnologías de síntesis de voz	24
9.1. Casos de prueba del backend	54
9.2. Funciones de prueba implementadas	55
9.3. Tipos de pruebas del frontend	56
9.4. Estructura de pruebas del DashboardComponent	56
9.5. Heurísticas de Nielsen evaluadas	58
9.6. Resultados de pruebas por módulo	59
9.7. Resultados de la encuesta de uso de PronunciApp	59
9.8. Resultados de la evaluación de heurísticas de Nielsen	60
9.9. Distribución temporal de respuestas de la encuesta	60
9.10. Distribución de evaluación de tiempo de respuesta	61
9.11. Análisis de cumplimiento de expectativas	61
9.12. Métricas de consistencia en evaluación de expertos	62
9.13. Resultados detallados de satisfacción del usuario	62
9.14. Heurísticas con puntuación perfecta (5.0/5.0)	62
9.15. Heurísticas con potencial de optimización	63
9.16. Correlación entre evaluación de expertos y usuarios	63
9.17. Principales fortalezas del sistema	64
9.18. Áreas de mejora identificadas por usuarios	64
9.19. Áreas de mejora identificadas por expertos	64
9.20. Impacto e implicaciones de los resultados obtenidos	65
9.21. Análisis detallado de problemas de grabación	65
9.22. Análisis de usuarios con expectativas parciales	65
9.23. Validación estadística de la muestra	66
9.24. Identificación y mitigación de sesgos	66

Resumen

PronunciApp es un prototipo de aplicación web desarrollado como proyecto de grado para proporcionar una mejora sobre la pronunciación del idioma inglés a estudiantes de Ingeniería de Sistemas de la Universidad del Valle, sede Tuluá. El sistema implementado permite a los usuarios grabar su voz al repetir palabras en inglés, las cuales son analizadas utilizando tecnologías de reconocimiento de voz para generar retroalimentación sobre aspectos de la pronunciación. La aplicación se desarrolló utilizando Angular 19 para la interfaz de usuario, FastAPI para el backend, y se desplegó en la plataforma Azure. El proceso de desarrollo siguió la metodología Scrumban a lo largo de 40 sprints, aplicando principios de diseño UX/UI para crear una herramienta que permite la práctica de pronunciación mediante el análisis de grabaciones de audio.

Palabras claves: Reconocimiento de voz, retroalimentación de pronunciación, fonética, entrenamiento de habla, síntesis de voz, aplicación web, Angular, FastAPI, Azure, metodología Scrumban.

Abstract

PronunciApp is a web application prototype developed as a graduation project to provide improvement on English language pronunciation for Systems Engineering students at Universidad del Valle, Tuluá campus. The implemented system allows users to record their voice while repeating English words, which are analyzed using voice recognition technologies to generate feedback on pronunciation aspects. The application was developed using Angular 19 for the user interface, FastAPI for the backend, and was deployed on the Azure platform. The development process followed the Scrumban methodology throughout 40 sprints, applying UX/UI design principles to create a tool that allows pronunciation practice through audio recording analysis.

Keywords: Voice recognition, pronunciation feedback, phonetics, speech training, voice synthesis, web application, Angular, FastAPI, Azure, Scrumban methodology.

Capítulo 1

Introducción

En el contexto globalizado actual, la competencia comunicativa en inglés constituye un factor determinante para el desarrollo profesional y académico. Según Manuel Vidal de Trinity College London Approved Service Provider: "Poder comunicarse en más de un idioma es casi una exigencia para ser seleccionado, tanto para entrar a trabajar en una empresa, como para poder crecer profesionalmente dentro de ella" [1]. Esta realidad adquiere particular relevancia en el campo de la ingeniería de sistemas, donde el dominio del inglés técnico facilita el acceso a documentación especializada, herramientas de desarrollo y oportunidades laborales internacionales.

La pronunciación representa uno de los componentes más desafiantes en el aprendizaje del inglés como lengua extranjera, especialmente para estudiantes hispanohablantes. Las diferencias fonológicas entre el español y el inglés generan dificultades específicas que requieren estrategias pedagógicas especializadas y tecnologías innovadoras. En este contexto, el presente trabajo documenta el desarrollo de PronunciApp, un prototipo de aplicación web que utiliza tecnologías de reconocimiento de voz e inteligencia artificial para proporcionar retroalimentación sobre la pronunciación del inglés.

PronunciApp se desarrolló fundamentándose en principios pedagógicos contemporáneos que enfatizan el aprendizaje autónomo y la retroalimentación inmediata. Como señala Sarmiento: "deben perfeccionarse métodos y formas de enseñanza y organizarlos de manera que permitan un máximo aprovechamiento de capacidades cognoscitivas de los estudiantes y desarrollen sus posibilidades para trabajar de forma independiente, preparándolos para adquirir los conocimientos por sí mismos, así como aplicarlos de manera efectiva, original y creativa" (p. 102) [1]. Esta perspectiva pedagógica sustentó el diseño de una herramienta tecnológica que proporciona retroalimentación sobre la pronunciación.

La pronunciación es un aspecto fundamental en el aprendizaje del inglés porque determina si el mensaje puede ser entendido correctamente. Cuando una persona pronuncia bien, es más fácil para quien escucha comprender lo que dice, lo cual mejora la comunicación tanto en el ámbito académico como profesional. Por esta razón, desarrollar buenas habilidades de pronunciación no solo ayuda a comunicarse mejor, sino que también abre oportunidades laborales en un mundo cada vez más globalizado.

El presente documento describe el proceso de diseño, desarrollo e implementación de PronunciApp como una herramienta tecnológica creada para los estudiantes de Ingeniería de Sistemas de la Universidad del Valle, sede Tuluá. La metodología empleada integra principios de ingeniería de software, tecnologías emergentes de procesamiento de lenguaje natural y fundamentos pedagógicos del aprendizaje de idiomas asistido por computadora.

1.1. Formulación del problema

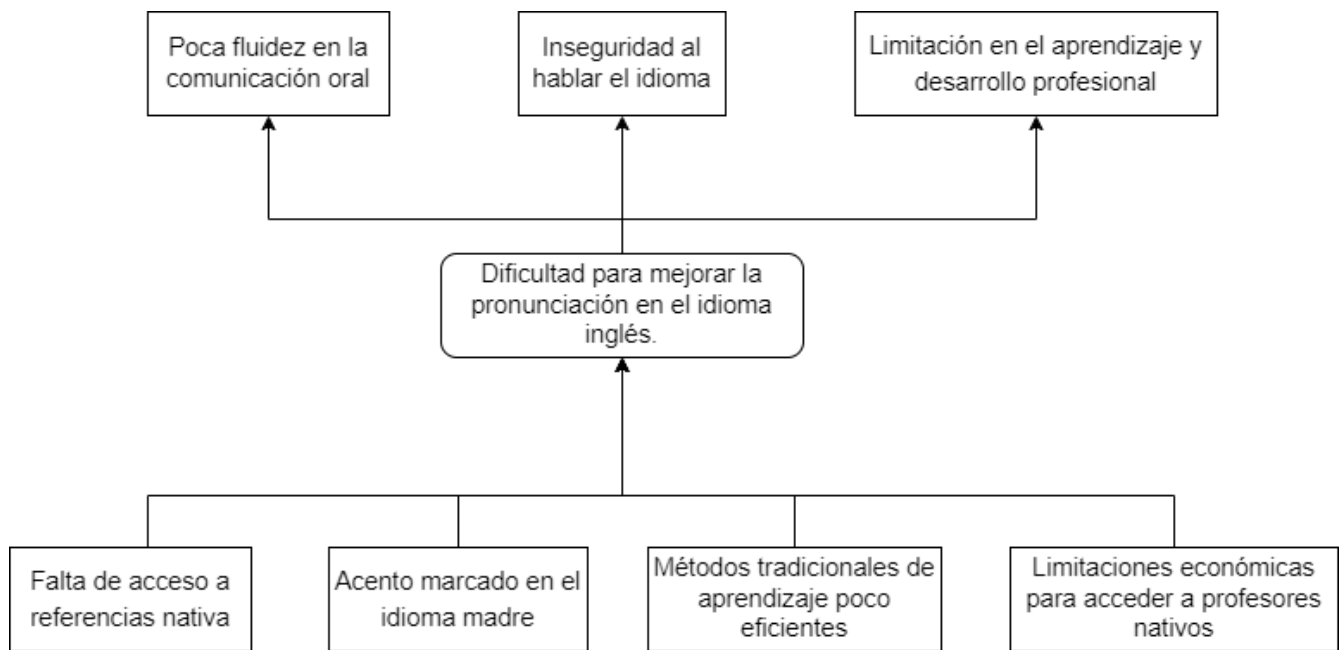


Figura 1.1: Representación gráfica de la propuesta.

1.2. Descripción del problema

La fluidez en el inglés es una habilidad crucial en el mundo contemporáneo, donde la comunicación efectiva en este idioma es esencial en diversos contextos, desde el ámbito profesional hasta el académico. Sin embargo, a pesar de su importancia, numerosas personas enfrentan dificultades para desarrollar esta habilidad de manera adecuada. Las limitaciones en la práctica oral, la falta de retroalimentación precisa y la escasez de recursos accesibles son solo algunos de los desafíos que obstaculizan el proceso de mejorar la fluidez en el inglés.

Una solución prometedora para abordar estos desafíos es el uso de tecnología innovadora. El uso de este tipo de tecnologías como la pronunciación automática, el reconocimiento de voz y la retroalimentación fonética puede proporcionar un entorno de aprendizaje más interactivo y efectivo. Mediante el entrenamiento de habla asistido por computadora, los usuarios pueden practicar su pronunciación y recibir retroalimentación instantánea sobre sus errores, lo que facilita una mejora progresiva en su fluidez oral. Es por ello que Pedraza (2023) sugiere “la IA se ha orientado a desarrollar soluciones flexibles y autónomas para problemas complejos”. (p. 14) [2]

La síntesis de voz y las tecnologías del habla fueron elementos clave en este enfoque, ya que permitieron la creación de aplicaciones con una interfaz de usuario vocal intuitiva y fácil de usar. Este tipo de software no solo brindó acceso a recursos de aprendizaje asequibles, sino que también ofreció la posibilidad de una evaluación precisa de la pronunciación y la corrección automática de errores. De esta manera, los usuarios pudieron recibir una retroalimentación personalizada que se adaptó a sus necesidades individuales de aprendizaje.

La integración de la inteligencia artificial en estas aplicaciones también jugó un papel fundamental al permitir un aprendizaje adaptativo y personalizado. Mediante el análisis de datos sobre el desempeño del usuario, los sistemas de IA pudieron identificar áreas de mejora específicas y ajustar dinámicamente el contenido y los ejercicios para optimizar el progreso del aprendizaje. Esto proporcionó una experiencia de aprendizaje más efectiva y eficiente, que se adaptó a las necesidades y habilidades únicas de cada usuario. Como menciona Macías [3]:

“Se dice pues que la Inteligencia Artificial es una herramienta útil para acelerar el proceso de consecución de los objetivos globales de educación mediante la reducción de las dificultades de acceso al aprendizaje, la automatización de los procesos de gestión y la optimización de los métodos que permiten mejorar los resultados en el aprendizaje”. (p. 15)

La mala pronunciación en inglés pudo tener varios efectos negativos en los estudiantes universitarios, sin exceptuar a los estudiantes de la Universidad del Valle. Primero, la mala pronunciación pudo dificultar la comprensión de los estudiantes durante las interacciones académicas y profesionales. Según un estudio, la capacidad de comunicarse eficazmente en inglés es crucial para el éxito académico y profesional, ya que el inglés es el idioma dominante en muchos campos científicos y tecnológicos [4].

Además, la mala pronunciación puede afectar la confianza y la autoestima de los estudiantes. Los estudiantes que no pueden pronunciar correctamente pueden sentirse avergonzados o inseguros al hablar, lo que puede llevar a una menor participación en clases y actividades académicas. Esto se refleja en una disminución de oportunidades para mejorar sus habilidades lingüísticas y, por ende, su rendimiento académico global [4] .

Otro aspecto importante es que la mala pronunciación puede limitar las oportunidades de los estudiantes en el mercado laboral global. Muchos empleadores buscan candidatos con buenas habilidades de comunicación en inglés, y una pronunciación deficiente puede ser vista como una falta de competencia profesional. La habilidad de hablar inglés con claridad y precisión es altamente valorada en el mundo laboral actual, y la falta de esta habilidad puede poner a los graduados en desventaja [5].

Por lo tanto, el uso de tecnologías como la pronunciación automática, el reconocimiento de voz y la inteligencia artificial ofreció un enfoque prometedor para mejorar la fluidez en el inglés. Al proporcionar práctica oral interactiva, estas herramientas tuvieron el potencial de facilitar un progreso significativo en el desarrollo de habilidades lingüísticas para aquellos que desearon comunicarse efectivamente en inglés.

1.3. Definición del problema

¿Cómo mejorar la pronunciación de inglés en los estudiantes de la carrera de Ingeniería de sistemas de la Universidad del Valle, sede Tuluá por medio de un prototipo de aplicación web?.

1.4. Objetivos

1.4.1. Objetivo general

Implementar un prototipo para proporcionar retroalimentación de la pronunciación en el aprendizaje del inglés para los estudiantes de la carrera de Ingeniería de Sistemas de la Universidad del Valle, sede Tuluá por medio de una aplicación web.

1.4.2. Objetivos específicos

1. Realizar una revisión de la literatura para identificar técnicas de procesamiento de lenguaje natural (PLN) que sean efectivas para el reconocimiento de voz y que puedan ser aplicadas al análisis de la falta de fluidez en el habla del inglés, utilizando fuentes como libros, artículos académicos y otros recursos relevantes.

2. Utilizar un tablero Kanban para la gestión de las tareas que se plantean seguir en el desarrollo del proyecto.

3. Diseñar un prototipo de aplicación web que integre tecnología de reconocimiento de voz y de IA acorde a las reglas del UX/UI.

4. Desarrollar el prototipo de una aplicación web con el uso de frameworks adecuados para el cumplimiento de los requisitos funcionales y no funcionales de la herramienta diseñada.

5. Realizar pruebas de software por medio del uso de herramientas de testing para determinar el correcto funcionamiento del prototipo final.

1.5. Resultado esperado

Tabla 1.1: Objetivos específicos y sus resultados esperados

Objetivo Específico	Resultado Esperado
1. Realizar una revisión de la literatura para identificar técnicas de procesamiento de lenguaje natural (PLN) que sean efectivas para el reconocimiento de voz y que puedan ser aplicadas al análisis de la falta de fluidez en el habla del inglés, utilizando fuentes como libros, artículos académicos y otros recursos relevantes.	Se obtuvo un informe detallado de técnicas de PLN efectivas para el reconocimiento de voz y análisis de fluidez en inglés, basado en una revisión exhaustiva de la literatura académica.
2. Utilizar un tablero Kanban para la gestión de las tareas que se plantean seguir en el desarrollo del proyecto.	Sus columnas permitieron una mejor organización en el seguimiento de las distintas actividades que se plantearon, ajustando las prioridades y mejorando el flujo de trabajo.

Continúa en la siguiente página

3. Diseñar un prototipo de aplicación web que integre tecnología de reconocimiento de voz y de IA acorde a las reglas del UX/UI.	1. Modelo PLN entrenado: Un modelo de Procesamiento de Lenguaje Natural (PLN) entrenado específicamente para entender palabras en inglés. El modelo proporcionará una evaluación precisa de la pronunciación del usuario, indicando qué tan bien pronunció cada palabra o frase. 2. Diseños de Mockups o Wireframes: Diseños detallados de la interfaz de usuario (UI) del prototipo de la aplicación web. Estos mockups o wireframes seguirán principios de experiencia de usuario (UX) y estarán diseñados para ser intuitivos y fáciles de usar. 3. Diseños MER de la base de datos: Modelos Entidad-Relación (MER) detallados para la base de datos que soportará la aplicación. Estos diseños incluirán la estructura de datos necesaria para almacenar información del usuario, registros de pronunciación, resultados de evaluaciones y cualquier otra información relevante para el funcionamiento de la aplicación. 4. Prototipo de aplicación web: Un prototipo funcional de la aplicación web que incorpore tecnología de reconocimiento de voz e inteligencia artificial (IA). La aplicación estará diseñada conforme a principios de UX/UI, con una interfaz intuitiva y fácil de usar, permitiendo a los estudiantes practicar y mejorar su pronunciación en inglés. El prototipo será evaluado para asegurar su efectividad y usabilidad, con retroalimentación recopilada de usuarios reales para posibles mejoras futuras.
4. Desarrollar el prototipo de una aplicación web con el uso de frameworks adecuados para el cumplimiento de los requisitos funcionales y no funcionales de la herramienta diseñada.	Se desarrolló una aplicación web funcional y operativa que cumplió con todos los requisitos funcionales y no funcionales especificados, utilizando frameworks adecuados para el desarrollo.
5. Realizar pruebas de software por medio del uso de herramientas de testing para determinar el correcto funcionamiento del prototipo final.	Se llevaron a cabo pruebas exhaustivas de la aplicación web utilizando herramientas de testing, asegurando que el prototipo final funcionó correctamente y sin errores.

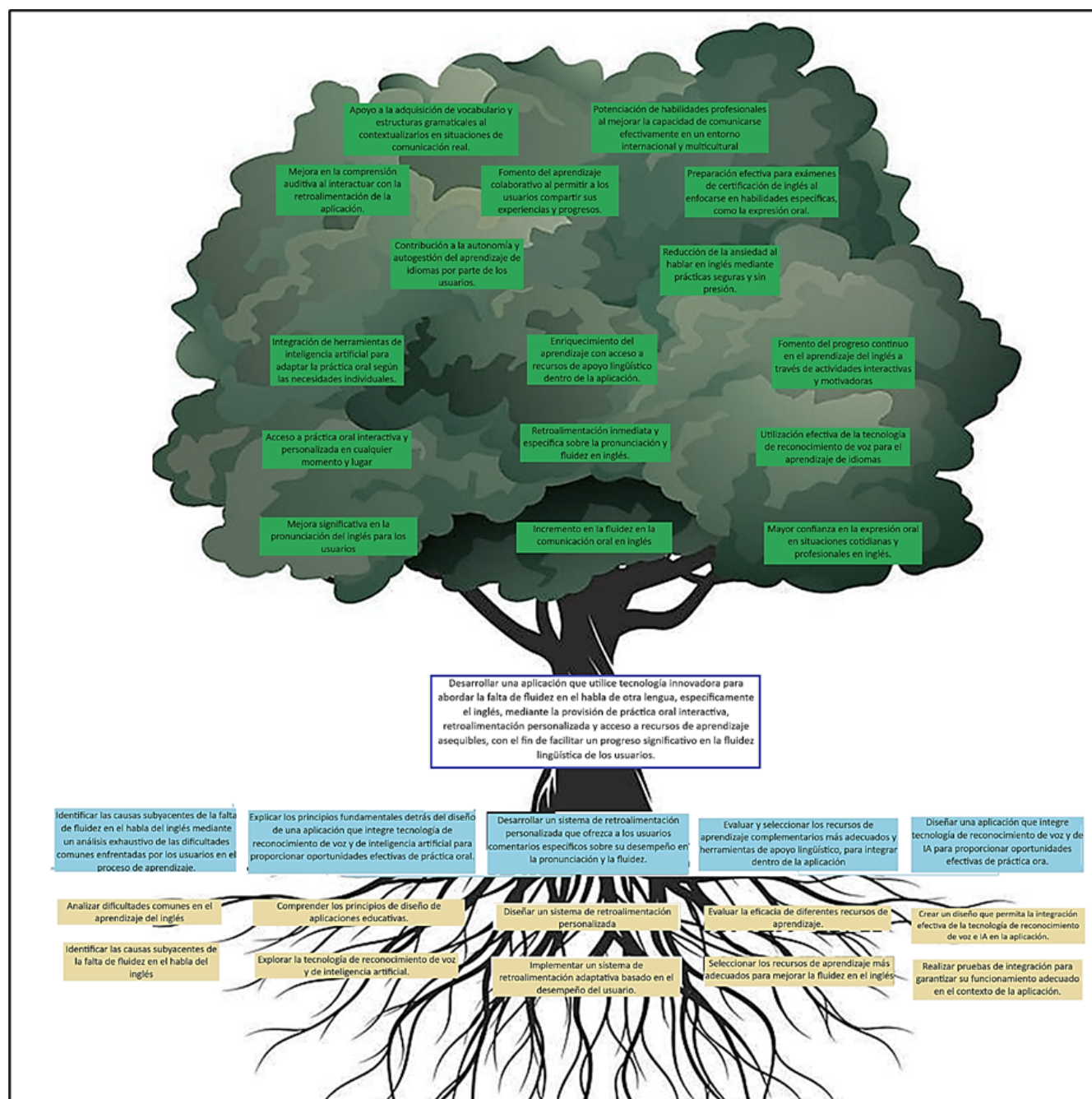


Figura 1.2: Árbol de objetivos.

1.6. Antecedentes

Assessing Pronunciation Improvement in Students of English Using a Controlled Computer-Assisted Pronunciation Tool [6]: El artículo aborda el tema de las herramientas de pronunciación asistida por ordenador (CAPT) y su eficacia en la enseñanza de la pronunciación en inglés. A pesar del creciente interés y éxito comercial de estas herramientas, la evidencia empírica sobre su eficacia pedagógica sigue siendo limitada. El estudio presenta una herramienta CAPT basada en pares mínimos que

utiliza ciclos de exposición-percepción-producción y ofrece retroalimentación automática a los alumnos en la producción de sonidos difíciles en inglés. Se llevó a cabo un estudio controlado donde un grupo de usuarios utilizó la herramienta CAPT y se compararon los resultados de su pronunciación con los de un grupo que recibió formación en un aula tradicional. Los resultados mostraron una mejora significativa en la pronunciación de los alumnos que utilizaron la herramienta CAPT, así como una correlación entre la evaluación manual y la evaluación automática de la herramienta CAPT.

Duolingo [7]: Duolingo es una plataforma educativa en línea que ofrece cursos gratuitos para aprender idiomas de forma amena y accesible. Fundada en 2011 por Luis von Ahn y Severin Hacker, cuenta con más de 300 millones de usuarios activos en todo el mundo, convirtiéndose en la aplicación para aprender idiomas más popular del planeta. Su misión principal es democratizar la educación, brindando la oportunidad de aprender un nuevo idioma a cualquier persona, sin importar su ubicación, nivel socioeconómico o experiencia previa.

Speechace [8]: Speechace es una plataforma de evaluación y mejora de la pronunciación y fluidez en inglés, basada en tecnología de reconocimiento de voz. Fundada en 2012 por la Dra. Aoife Cahill y el Dr. Mark Prensky, Speechace utiliza algoritmos de inteligencia artificial para analizar el habla de los usuarios y proporcionarles retroalimentación personalizada sobre su pronunciación, ritmo, entonación y otros aspectos de su fluidez oral.

Elsa Speak [9]: Elsa Speak es una aplicación móvil que utiliza tecnología de inteligencia artificial (IA) para ayudar a los usuarios a mejorar su pronunciación y fluidez en inglés de manera personalizada y atractiva. Fundada en 2015 por el equipo de Elsa Technologies, la aplicación cuenta con más de 7 millones de usuarios en todo el mundo y ha sido reconocida por su efectividad en la enseñanza de la pronunciación del inglés.

1.7. Justificación

La pronunciación del inglés representa uno de los principales desafíos para los estudiantes hispanohablantes en el desarrollo de competencias comunicativas efectivas en este idioma [4]. En el contexto específico de la carrera de Ingeniería de Sistemas de la Universidad del Valle, sede Tuluá, esta problemática adquiere particular relevancia debido a la creciente demanda del mercado laboral tecnológico por profesionales con competencias comunicativas sólidas en inglés.

Desde la perspectiva pedagógica, la integración de inteligencia artificial en la educación ha demostrado ser fundamental para la personalización del aprendizaje [5]. Las tecnologías de reconocimiento de voz y retroalimentación inmediata en el proceso de aprendizaje permiten crear un ambiente de práctica controlado que reduce la ansiedad y facilita la autocorrección, aspectos fundamentales para el desarrollo de la competencia fonológica [6].

La investigación reciente en el campo de Computer-Assisted Language Learning (CALL) ha demostrado que las aplicaciones web especializadas en pronunciación pueden mejorar significativamente la precisión fonética de los estudiantes cuando se diseñan considerando principios pedagógicos específicos y tecnologías emergentes [10]. Además, el uso de modelos de deep learning para el procesamiento de lenguaje natural ha revolucionado las herramientas de aprendizaje de idiomas [11]. Sin embargo, existe una notable carencia de herramientas desarrolladas específicamente para el contexto colombiano y las necesidades particulares de estudiantes de ingeniería.

En el ámbito profesional de la ingeniería de sistemas, la capacidad de comunicarse efectivamente en inglés no solo facilita el acceso a documentación técnica y recursos académicos, sino que también resulta determinante para la participación en proyectos internacionales, conferencias especializadas y oportunidades laborales en empresas multinacionales. Según el informe del British Council (2023), el 78 % de las empresas tecnológicas en Colombia consideran el inglés como un requisito fundamental para cargos de desarrollo de software y gestión de proyectos [12].

Los estudiantes de Ingeniería de Sistemas de la Universidad del Valle, sede Tuluá, enfrentan limitaciones específicas en cuanto al acceso a laboratorios de idiomas especializados y recursos tecnológicos dedicados exclusivamente al desarrollo de habilidades de pronunciación. La implementación de una aplicación web accesible desde cualquier dispositivo con conexión a internet democratiza el acceso a estas herramientas y permite un aprendizaje autónomo y personalizado.

El desarrollo de PronunciApp como prototipo de aplicación web constituye una respuesta técnica y pedagógica a la necesidad de crear una herramienta metodológicamente robusta que integre tecnologías de reconocimiento automático de voz (ASR) y técnicas de análisis fonético específicamente configuradas para el inglés americano. La metodología propuesta fusiona principios sólidos de ingeniería de software con fundamentos teóricos de la pedagogía del lenguaje, asegurando tanto la integridad técnica del sistema como su efectividad en el proceso de enseñanza-aprendizaje.

La elección del enfoque de prototipado permite validar incrementalmente las funcionalidades propuestas y realizar ajustes basados en la retroalimentación de usuarios reales, siguiendo las mejores prácticas de desarrollo centrado en el usuario. Esto resulta particularmente relevante en el contexto educativo, donde la usabilidad y la experiencia del usuario determinan en gran medida la adopción y efectividad de las herramientas tecnológicas.

La investigación propuesta contribuye al cuerpo de conocimiento en el área de tecnologías educativas aplicadas al aprendizaje de idiomas, específicamente en el contexto latinoamericano. Los resultados obtenidos podrán ser replicados y adaptados en otras instituciones de educación superior que enfrentan desafíos similares en la enseñanza del inglés como lengua extranjera.

Además, el proyecto fortalece la línea de investigación en desarrollo de software educativo del programa de Ingeniería de Sistemas, contribuyendo a la visibilidad académica de la Universidad del Valle en el campo de las tecnologías emergentes aplicadas a la educación. La documentación del proceso de desarrollo y los resultados de evaluación proporcionarán valiosos insumos para futuras investigaciones en el área.

1.8. Información sobre los capítulos

En la siguiente tabla se muestra una corta descripción de cada capítulo del documento.

Tabla 1.2: Estructura de capítulos

Capítulo	Descripción
Capítulo 2: Alcance de la propuesta	Se establecen los alcances demográfico, temporal, económico y metodológico del proyecto, definiendo la población objetivo (estudiantes de Ingeniería de Sistemas de Univalle Tuluá), cronograma de desarrollo y recursos necesarios para la implementación de PronunciApp.
Capítulo 3: Marco Referencial	Se fundamentan los conceptos teóricos sobre aprendizaje de idiomas, interlengua, input comprensible, tecnologías de reconocimiento de voz, síntesis neuronal con VALL-E-X, y teorías pedagógicas que sustentan el desarrollo de aplicaciones CAPT (Computer-Assisted Pronunciation Training).
Capítulo 4: Metodología	Se detalla la metodología de investigación exploratoria, la selección de Scrumban como framework de desarrollo ágil, la implementación de Kanban para gestión de actividades, cronograma de 8 meses, presupuesto de recursos humanos y hardware, y análisis del impacto ambiental.
Capítulo 5: Revisión literatura	Se analizan tecnologías emergentes en síntesis de voz (VALL-E-X, Whisper, Coqui-TTS, Bark, StyleTTS2), arquitecturas de deep learning para ASR, extracción de características acústicas y prosódicas, y comparación de plataformas CAPT existentes como Duolingo, Speechace y Elsa Speak.
Capítulo 6: Tablero kanban	Se documenta la implementación de GitHub Projects como herramienta de gestión visual, configuración de estados (Todo, Done), sistema de priorización (P1, P2, P3), organización de 40 sprints semanales y seguimiento del flujo de trabajo durante las fases de investigación, desarrollo y despliegue.
Capítulo 7: Diseño del prototipo	Se presenta el proceso de diseño UX/UI utilizando Figma y V0 by Vercel, aplicación de principios de simplicidad y reducción de carga cognitiva, wireframing, arquitectura de interfaz minimalista, y decisiones de diseño orientadas a maximizar la usabilidad y minimizar la complejidad visual.
Capítulo 8: Desarrollo del prototipo	Se implementa una arquitectura de microservicios con FastAPI para el backend integrando VALL-E-X, Angular 19 para el frontend responsivo, contenedores Docker para despliegue, infraestructura Azure NCasT4_v3, y cumplimiento de requisitos funcionales y no funcionales específicos.

Continúa en la siguiente página

Capítulo 9: Pruebas de usabilidad y de software	Se ejecutan pruebas técnicas automatizadas con pytest para backend y Jasmine para frontend Angular, evaluación endpoints REST, procesamiento de audio, renderización de componentes, y validación del cumplimiento de criterios de aceptación establecidos.
Capítulo 10: Conclusiones y trabajos futuros	Se evalúa el impacto de tecnologías PLN y síntesis neuronal en sistemas CAPT, efectividad del tablero Kanban en 40 sprints, éxito de la integración Figma-V0, viabilidad de la arquitectura FastAPI-Angular en Azure, y propuestas de mejoras tecnológicas, pedagógicas y validación a largo plazo.

Fuente: Elaboración propia

Capítulo 2

Alcance de la propuesta

2.1. Declaración del alcance

Alcance demográfico

Los estudiantes de inglés en la carrera de Ingeniería de Sistemas en la Universidad del Valle sede Tuluá

Alcance temporal

Desde el segundo semestre del 2024 y el primer semestre del 2025

Alcance económico

Se cuenta con un computador de mesa para el desarrollo de la aplicación y los costos necesarios para hacer un despliegue en la nube

Alcance metodológico

Es exploratorio debido a que es una problemática poco estudiada y se aborda con una perspectiva innovadora

Revisión de la literatura

Se llevó a cabo una revisión detallada de las técnicas más efectivas de PLN para el reconocimiento de voz y análisis de fluidez en inglés.

Desarrollo del prototipo

Se diseñó un prototipo de aplicación web que incorporó tecnologías de reconocimiento de voz e IA, asegurando una interfaz de usuario intuitiva y fácil de usar.

Desarrollo de la aplicación

Utilizando frameworks adecuados, se desarrolló una aplicación web completa que cumplió con todos los requisitos definidos.

Pruebas y validación

Se realizaron pruebas exhaustivas del prototipo para identificar y corregir cualquier error, garantizando el funcionamiento óptimo de la aplicación.

Capítulo 3

Marco referencial

3.1. Conceptos Básicos

En este capítulo se abordan los conceptos necesarios para el entendimiento del desarrollo de una aplicación web para el aprendizaje de pronunciación basada en el proyecto VALL-E-X.

Se recomienda que el lector tenga formación básica en:

3.1.1. Desarrollo de aplicaciones web:

- Conocimiento básico sobre desarrollo de aplicaciones web frontend y backend.
- Comprensión del funcionamiento de las Web APIs para audio y multimedia.
- Conocer las capacidades de grabación y reproducción de audio en navegadores web.
- Conceptos de integración de modelos de IA en aplicaciones web.

3.1.2. Conceptos de síntesis y clonación de voz:

- Conocimiento básico sobre síntesis de voz neuronal y modelos generativos.
- Comprensión del funcionamiento de VALL-E-X y técnicas de clonación de voz.
- Conceptos fundamentales de procesamiento digital de señales de audio.
- Conocimiento sobre transformers y arquitecturas de atención para audio.

3.1.3. Tecnologías de aprendizaje profundo:

- Redes neuronales para generación de audio y síntesis de voz.
- Modelos de lenguaje multimodal para procesamiento de audio y texto.
- Técnicas de entrenamiento con pocos ejemplos (few-shot learning) para clonación de voz.

3.2. Marco teórico

3.2.1. Fundamentos del aprendizaje de idiomas y pronunciación

El aprendizaje de un idioma extranjero es un proceso complejo que involucra una variedad de factores cognitivos, sociales y afectivos. Comprender los fundamentos subyacentes de este proceso es crucial para diseñar estrategias efectivas de enseñanza y aprendizaje, especialmente en lo que respecta a la pronunciación, que desempeña un papel crucial en la comunicación efectiva en un idioma. En este apartado, exploraremos los principales fundamentos del aprendizaje de idiomas y su relación con la adquisición de una pronunciación precisa.

El proceso de aprendizaje de idiomas ha sido objeto de estudio por parte de numerosos teóricos y lingüistas a lo largo de los años. En particular, la teoría del aprendizaje de segundas lenguas propuesta por Stephen Krashen[13] en 1982 ha sido influyente en este campo. Krashen postula que el aprendizaje de un idioma se produce de dos maneras principales: adquisición y aprendizaje. La adquisición se refiere al proceso natural de absorber un idioma a través de la exposición y la interacción, mientras que el aprendizaje implica el estudio consciente de las reglas gramaticales y vocabulario. Esta distinción tiene implicaciones importantes para la enseñanza de la pronunciación, ya que sugiere que la práctica activa y la inmersión son fundamentales para el desarrollo de habilidades lingüísticas auténticas.

Además, la importancia de la pronunciación en el aprendizaje de idiomas ha sido ampliamente reconocida en la literatura académica. Según Derwing y Munro [14], una pronunciación clara y comprensible es esencial para la comunicación efectiva en un idioma extranjero. La pronunciación afecta la inteligibilidad y la percepción del hablante por parte de los interlocutores nativos, lo que a su vez influye en la calidad y eficacia de la interacción comunicativa. Por lo tanto, la enseñanza de la pronunciación no solo implica la reproducción precisa de sonidos, sino también la atención a la entonación, el ritmo y el flujo del habla.

Asimismo, la investigación en neurociencia ha proporcionado evidencia de que la pronunciación se desarrolla mejor durante la infancia y la adolescencia, lo que sugiere que existe una ventana crítica para el aprendizaje óptimo de la pronunciación [15]. Sin embargo, esto no implica que los adultos no puedan mejorar su pronunciación; simplemente indica que puede requerir un esfuerzo adicional y una práctica más deliberada.

La enseñanza de la pronunciación en el aprendizaje de idiomas también se ve influenciada por factores sociales y afectivos. Investigaciones como las realizadas por Rubin[16] han destacado la importancia de la autoconfianza y la actitud del estudiante hacia la pronunciación en su proceso de aprendizaje. Los estudiantes que se sienten seguros y motivados tienden a tener un mejor desempeño en la mejora de su pronunciación, mientras que aquellos que experimentan ansiedad o inseguridad pueden enfrentar mayores dificultades para desarrollar habilidades fonéticas precisas.

Además, la exposición a modelos de habla auténticos y variados desempeña un papel crucial en el desarrollo de la pronunciación. Los estudios de input comprensible, como los realizados por Long[17], sugieren que los estudiantes aprenden mejor cuando están expuestos a input lingüístico que es comprensible pero también ligeramente más avanzado que su nivel actual de competencia. Esto implica que los materiales de enseñanza deben proporcionar ejemplos de pronunciación auténtica y natural, así como oportunidades para la práctica activa y la retroalimentación.

Por otro lado, la pronunciación también está influenciada por factores culturales y contextuales. In-

vestigaciones como las de Levis[18] han señalado que las normas de pronunciación pueden variar significativamente entre diferentes variedades del mismo idioma, así como entre diferentes contextos sociales y situaciones de comunicación. Por lo tanto, los enfoques de enseñanza de la pronunciación deben ser sensibles a estas variaciones y proporcionar a los estudiantes las habilidades necesarias para adaptarse a una variedad de contextos lingüísticos.

En el ámbito de la tecnología educativa, el desarrollo de aplicaciones web ha abierto nuevas posibilidades para la enseñanza y el aprendizaje de la pronunciación. Investigaciones como las realizadas por Warschauer y Meskill[19] han explorado el potencial de la tecnología para ofrecer retroalimentación auditiva inmediata y personalizada, así como para facilitar la práctica autodirigida y la autoevaluación. Las aplicaciones y software de reconocimiento de voz, en particular, han demostrado ser eficaces para ayudar a los estudiantes a identificar y corregir errores de pronunciación de manera interactiva.

Sin embargo, es importante tener en cuenta que la tecnología no sustituye la enseñanza tradicional ni la interacción humana. Investigaciones como las realizadas por Chapelle [20] han destacado la importancia de integrar cuidadosamente la tecnología en el currículo de idiomas, asegurándose de que complemente y enriquezca la experiencia de aprendizaje, en lugar de reemplazarla. Además, es fundamental considerar cuestiones de accesibilidad y equidad en el diseño y la implementación de aplicaciones web para la enseñanza de la pronunciación, asegurando que estén disponibles y sean efectivas para todos los estudiantes, independientemente de su contexto socioeconómico o recursos tecnológicos disponibles.

3.2.2. Inteligencia Artificial en la Educación

La Inteligencia Artificial (IA) se refiere a la capacidad de las máquinas para realizar tareas que normalmente requieren inteligencia humana, como el aprendizaje, la percepción y la toma de decisiones. Estas capacidades son impulsadas por algoritmos avanzados y grandes volúmenes de datos, lo que permite a las máquinas mejorar su rendimiento con el tiempo. Según Goodfellow, Bengio y Courville [21], la IA combina técnicas de aprendizaje automático, redes neuronales y procesamiento de lenguaje natural para crear sistemas que pueden imitar o mejorar la cognición humana [21]. La evolución de la IA ha sido marcada por avances en diversas áreas, desde la visión por computadora hasta la robótica, todos los cuales han contribuido a su aplicación en múltiples dominios, incluyendo la educación [22].

En el ámbito educativo, la IA ha sido utilizada para personalizar el aprendizaje, proporcionar asistencia inmediata y evaluar el progreso de los estudiantes. Los sistemas de tutoría inteligente, como los desarrollados recientemente, pueden adaptarse a las necesidades individuales de los estudiantes, ofreciendo recomendaciones y ejercicios personalizados para mejorar su rendimiento académico [23]. Además, las plataformas educativas impulsadas por IA, como Edmodo y Coursera, utilizan algoritmos de aprendizaje automático para analizar el comportamiento de los usuarios y ajustar el contenido educativo en consecuencia [24]. Este enfoque no solo facilita un aprendizaje más eficiente sino que también puede identificar y apoyar a los estudiantes en riesgo de rezagarse, permitiendo una intervención temprana y personalizada [25].

La IA ofrece numerosas ventajas para el aprendizaje de idiomas, especialmente en la mejora de la pronunciación y la fluidez. Las tecnologías de reconocimiento de voz y síntesis de habla, como las utilizadas en aplicaciones como Duolingo y Babbel, permiten a los estudiantes recibir retroalimentación instantánea sobre su pronunciación, lo que facilita una práctica más efectiva y autónoma. Un estudio de Wang [11] muestra que los sistemas de tutoría basados en IA pueden detectar errores específicos en la pronunciación y proporcionar correcciones precisas, ayudando a los estudiantes a mejorar su acento y entonación. Además,

la capacidad de la IA para analizar grandes volúmenes de datos permite la creación de modelos de aprendizaje que se adaptan continuamente a las necesidades del estudiante, proporcionando una experiencia de aprendizaje más dinámica y personalizada [26]. Esto no solo mejora la retención del conocimiento, sino que también hace el aprendizaje más accesible y atractivo para una variedad de estudiantes con diferentes estilos de aprendizaje [27].

3.2.3. Procesamiento del Lenguaje Natural (PLN)

El Procesamiento del Lenguaje Natural (PLN) es una disciplina de la inteligencia artificial que se ocupa de la interacción entre las computadoras y el lenguaje humano. Implica el uso de algoritmos y modelos matemáticos para que las máquinas comprendan, interpreten y generen el lenguaje de una manera que sea valiosa. Según Bird et al. [28], el PLN combina la lingüística computacional con el aprendizaje automático para analizar grandes cantidades de datos textuales y extraer información significativa. Este campo ha experimentado un rápido crecimiento gracias a los avances en el aprendizaje profundo, que han mejorado significativamente la capacidad de las máquinas para comprender y generar lenguaje natural [29].

Las técnicas de PLN han sido fundamentales para mejorar la pronunciación en el aprendizaje de idiomas. Las aplicaciones de aprendizaje de idiomas utilizan algoritmos de reconocimiento de voz para analizar la pronunciación de los estudiantes y proporcionar retroalimentación instantánea. Por ejemplo, los modelos de aprendizaje profundo, como las redes neuronales convolucionales (CNN) y las redes neuronales recurrentes (RNN), se han utilizado para el reconocimiento de voz, permitiendo la detección precisa de errores de pronunciación [10]. Además, técnicas como el etiquetado de partes del habla (POS tagging) y la lematización ayudan a descomponer y analizar las oraciones, facilitando una corrección más precisa y personalizada [29].

Las herramientas y algoritmos de PLN desempeñan un papel crucial en el desarrollo de aplicaciones educativas para el aprendizaje de idiomas. Entre las herramientas más destacadas se encuentran NLTK (Natural Language Toolkit) y SpaCy, que proporcionan bibliotecas extensas para el procesamiento de lenguaje natural. Estas herramientas permiten realizar tareas como tokenización, análisis sintáctico y etiquetado de partes del habla, que son esenciales para construir aplicaciones educativas avanzadas [28] [29].

Por otro lado, los algoritmos de aprendizaje profundo, como los modelos de transformadores (por ejemplo, BERT y GPT-3), han demostrado una gran capacidad para manejar secuencias largas de texto y proporcionar respuestas coherentes en el contexto del aprendizaje de idiomas. Estos modelos pueden ser entrenados con grandes volúmenes de datos para mejorar continuamente su precisión y capacidad de generalización, ofreciendo una experiencia de aprendizaje más interactiva y efectiva para los estudiantes [30] [31].

3.3. Marco conceptual

El marco conceptual de este proyecto se basó en una integración de teorías del aprendizaje lingüístico, enfoques pedagógicos efectivos y tecnologías innovadoras para diseñar un aplicativo web de retroalimentación sobre la pronunciación en el contexto del aprendizaje del inglés como segunda lengua.

3.3.1. Teorías del aprendizaje lingüístico:

Interlengua: La teoría de la interlengua postula que los estudiantes de una segunda lengua desarrollan un sistema intermedio de reglas lingüísticas que refleja su progreso en el aprendizaje. En el caso de la pronunciación, esto implicó que los estudiantes podían experimentar errores sistemáticos que reflejaran las reglas fonéticas de su lengua materna.

Input Comprensible: Según esta teoría, los estudiantes aprenden mejor cuando están expuestos a input lingüístico que está ligeramente más allá de su nivel actual de competencia. En el contexto de la pronunciación, esto sugirió la importancia de proporcionar a los estudiantes modelos de pronunciación claros y comprensibles.

3.3.2. Enfoques pedagógicos para la enseñanza de la pronunciación:

Enfoque comunicativo: Este enfoque enfatiza la importancia de la comunicación efectiva en situaciones reales. En el contexto de la pronunciación, esto implicó centrarse en la producción y comprensión de sonidos que son relevantes para la comunicación cotidiana.

Enfoque basado en la conciencia fonológica: Este enfoque se centra en el desarrollo de la conciencia de los sonidos del habla y su relación con la escritura. En el contexto de la pronunciación, esto pudo implicar actividades que ayudaran a los estudiantes a reconocer y producir sonidos específicos del inglés.

3.3.3. Tecnologías para la mejora de la pronunciación:

Reconocimiento de voz: Esta tecnología permite a los estudiantes recibir retroalimentación inmediata sobre su pronunciación mediante la comparación con modelos de pronunciación predefinidos.

Repetición y retroalimentación diferida: Las herramientas que permiten a los estudiantes grabar y reproducir su propia voz les brindan la oportunidad de comparar su pronunciación con tecnología de síntesis de voz y recibir retroalimentación diferida sobre sus errores.

Capítulo 4

Metodología

4.1. Metodologia de investigacion

En este trabajo de grado, se propuso desarrollar una plataforma destinada a mejorar la pronunciación del inglés como segunda lengua. Para lograr este objetivo, fue crucial establecer una metodología de investigación robusta que guiara el proceso de desarrollo y evaluación de la plataforma. Es por ello, que se detalló la metodología propuesta, incluyendo la selección de participantes, el diseño del estudio, la recolección de datos y el análisis previsto. Es de aclarar que proyecto se encontró en el Nivel de Madurez Tecnológica (TRL) 5, el cual implicó un punto intermedio entre la fase de adquisición para ingeniería y manufactura y la demostración avanzada de la tecnología. [32]

El primer paso fue la selección de participantes representativos del público objetivo. Se buscó reclutar a estudiantes de inglés como segunda lengua, con diferentes niveles de competencia lingüística, en caso de que no hubiera participantes con un segundo idioma, se procedió a hacer una búsqueda exhaustiva en distintas fuentes para recolectar voces de personas con el idioma nativo inglés. Además, se consideró la diversidad cultural y demográfica para garantizar la representatividad de la muestra. Se utilizaron criterios de inclusión específicos para asegurar que los participantes estuvieran motivados para mejorar su pronunciación y estuvieran dispuestos a utilizar la plataforma propuesta durante el estudio.

El diseño del estudio se basó en un enfoque mixto que combinó métodos cualitativos y cuantitativos. Se llevó a cabo un estudio experimental con un grupo experimental que utilizó la plataforma de mejora de pronunciación y un grupo de control que no la utilizó. Se empleó un diseño pretest-postest para medir el impacto de la plataforma en la pronunciación de los participantes. Además, se realizaron entrevistas y encuestas para recopilar datos cualitativos sobre la experiencia de los usuarios y su percepción de la efectividad de la plataforma.

La recolección de datos se llevó a cabo en varias etapas. En primer lugar, se administraron pruebas de pronunciación tanto antes como después de la intervención con la plataforma. Estas pruebas incluyeron la lectura de palabras y frases en inglés, así como la participación en conversaciones simuladas.

Los datos cuantitativos recopilados se analizaron utilizando técnicas estadísticas apropiadas, como análisis de varianza y pruebas t. Se compararon las puntuaciones de pronunciación antes y después de la intervención para evaluar el impacto de la plataforma. Además, se realizó un análisis cualitativo de las entrevistas y encuestas para identificar temas emergentes y patrones en las experiencias de los usuarios. Los resultados de ambos análisis se integraron para obtener una comprensión holística de la efectividad

de la plataforma en la mejora de la pronunciación del inglés.

En conclusión, la metodología de investigación propuesta combinó enfoques cuantitativos y cualitativos para evaluar la efectividad de una plataforma de mejora de pronunciación del inglés. Al seleccionar cuidadosamente a los participantes, diseñar un estudio riguroso y utilizar una variedad de métodos de recolección y análisis de datos, se esperó obtener información valiosa sobre la utilidad y la aceptación de la plataforma. Estos hallazgos contribuyeron al desarrollo de herramientas más efectivas para el aprendizaje de idiomas y pudieron tener importantes implicaciones para la enseñanza de inglés como segunda lengua.

4.2. Metodología de desarrollo de software

Tabla 4.1: Comparación de metodologías

Metodología	Mayor presencia en internet	Mejor documentación	Certificación y training	Comunidad	Presencia empresarial	Proyectos de software	Total
Extreme Programming	3	3	2	3	3	3	17
Kanban	3	3	3	4	3	3	19
Scrumban	3	4	3	3	3	5	21

Se propone emplear una metodología ágil de desarrollo de software, específicamente Scrumban. Scrumban es una metodología iterativa e incremental que permite la entrega continua de funcionalidades y una rápida adaptación a los cambios en los requisitos de este trabajo. Dado que el desarrollo de la plataforma de retroalimentación sobre la pronunciación del inglés es un proceso complejo y sujeto a posibles modificaciones a medida que avanza el proyecto, Scrumban proporciona la flexibilidad necesaria para responder a estas demandas de manera efectiva.

En el marco de Scrumban, se identificaron tres roles principales: el Product Owner, responsable de definir y priorizar los requisitos del producto; el Scrum Master, encargado de facilitar el proceso de desarrollo y eliminar posibles obstáculos; y el Equipo de Desarrollo, conformado por los desarrolladores responsables de implementar las funcionalidades del producto. Cada uno de estos roles desempeñó funciones específicas dentro del equipo de trabajo para garantizar el éxito de este trabajo.

El proceso de desarrollo se organizó en iteraciones llamadas "sprints", que tuvieron una duración fija de dos semanas. Al inicio de cada sprint, se realizó una reunión de planificación en la que el equipo seleccionó un conjunto de tareas del backlog del producto para completar durante el sprint. Durante el sprint, se llevaron a cabo reuniones diarias de seguimiento para revisar el progreso y ajustar el plan si fue necesario. Al final de cada sprint, se realizó una demostración del trabajo realizado y una retrospectiva para identificar áreas de mejora.

Es importante resaltar que con un enfoque iterativo, centrado en el usuario y orientado a la calidad, se espera que la plataforma cumpla con las expectativas y necesidades de los usuarios, contribuyendo así a la mejora de la pronunciación del inglés como segunda lengua.

4.3. Metodología de gestión de actividades

Para el desarrollo de actividades en este proyecto, se utilizará la metodología Kanban. Kanban es una herramienta de gestión visual que facilita la organización, seguimiento y mejora continua de las tareas. Esta metodología es especialmente adecuada para proyectos complejos y en evolución, ya que permite una adaptación rápida a los cambios y una entrega constante de valor.

En el contexto de Kanban, se utilizará un tablero Kanban para gestionar el flujo de trabajo. Este tablero estará dividido en columnas que representan las diferentes etapas del proceso de trabajo, como “Por hacer”, “En progreso” y “Hecho”. Cada tarea se representará con una tarjeta que se moverá de una columna a otra a medida que avance en el proceso. Esto permitirá una visualización clara del estado de cada tarea y facilitará la identificación de posibles cuellos de botella.

La flexibilidad de Kanban permite ajustar las prioridades y gestionar el flujo de trabajo de manera continua. No se requieren iteraciones fijas, lo que facilita la incorporación de cambios y nuevas prioridades a medida que surgen.

Con este enfoque visual y adaptable, se esperó que la metodología Kanban permitiera una gestión eficiente de las actividades del proyecto, asegurando que el desarrollo de la plataforma de retroalimentación sobre la pronunciación del inglés cumpliera con los objetivos establecidos.

Capítulo 5

Revisión de literatura

5.1. Metodología de la Revisión

Para esta revisión sistemática se empleó una metodología estructurada que incluyó:

- **Búsqueda bibliográfica:** Se consultaron bases de datos académicas como IEEE Xplore, ACM Digital Library, ScienceDirect, y Google Scholar.
- **Criterios de inclusión:** Artículos publicados entre 2015-2024, enfocados en PLN, reconocimiento de voz, análisis de fluidez, y CAPT.
- **Palabras clave:** "speech recognition", "pronunciation assessment", "fluency analysis", "CAPT", "natural language processing", "English learning".
- **Análisis de aplicaciones comerciales:** Revisión de plataformas establecidas como Duolingo, Speechace, y Elsa Speak.

5.2. Fundamentos teóricos del reconocimiento de voz

Arquitecturas de deep learning para ASR

Las técnicas modernas de reconocimiento de voz se basan fundamentalmente en arquitecturas de aprendizaje profundo que han revolucionado el campo del ASR. Las principales arquitecturas incluyen:

Redes Neuronales Recurrentes (RNN) y LSTM: Las redes LSTM (Long Short-Term Memory) han demostrado ser particularmente efectivas para el procesamiento secuencial de señales de audio. Estas redes pueden capturar dependencias temporales a largo plazo en el habla, lo que es crucial para el análisis de patrones de pronunciación y fluidez.

Transformer y Arquitecturas de Atención: Los modelos basados en Transformer, como Wav2Vec2 [33] y Whisper [34], han establecido nuevos estándares en la precisión del reconocimiento de voz. Estos modelos utilizan mecanismos de auto-atención que permiten una mejor comprensión del contexto global de la señal de audio.

Redes Neuronales Convolucionales (CNN): Las CNN son efectivas para extraer características espectrales de las señales de audio, particularmente cuando se combinan con arquitecturas recurrentes en modelos híbridos.

Extracción de características acústicas

El análisis de la fluidez en el habla requiere la extracción de características acústicas específicas:

- **Características espectrales:** MFCC (Mel-Frequency Cepstral Coefficients), espectrogramas mel, y características de filtro banco.
- **Características prosódicas:** Pitch (F0), intensidad, duración, y patrones de ritmo.
- **Características temporales:** Pausas, velocidad de habla, y segmentación silábica.
- **Características fonéticas:** Análisis de formantes, transiciones consonánticas, y calidad vocálica.

Extracción de características avanzadas:

- Embeddings contextuales de audio (wav2vec2 [33], HuBERT [35])
- Análisis espectro-temporal con transformadas wavelet
- Características prosódicas multiescala
- Representaciones aprendidas end-to-end

5.3. Tecnologías emergentes en síntesis y análisis de voz

Whisper (OpenAI) [34]: Aunque principalmente conocido como modelo de reconocimiento de voz, Whisper ha demostrado capacidades excepcionales para el análisis de pronunciación:

- **Robustez multilingüe:** Entrenado en 680,000 horas de audio multilingüe supervisado
- **Detección de acentos:** Capacidad para identificar y clasificar diferentes acentos y variaciones dialectales
- **Análisis temporal:** Alineación precisa entre audio y texto para análisis de fluidez
- **Filtrado de ruido:** Resistencia excepcional a condiciones acústicas adversas

VALL-E-X [36]: Desarrollado por Microsoft, VALL-E-X representa un avance significativo en la síntesis de voz de pocas muestras (few-shot voice synthesis). Este modelo utiliza técnicas de codificación de audio neural que pueden generar voz sintética de alta calidad a partir de solo unos segundos de audio de referencia. Para aplicaciones CAPT, VALL-E-X ofrece capacidades únicas:

- **Generación de modelos de referencia:** Capacidad para crear pronunciaciones modelo personalizadas para diferentes acentos y estilos de habla
- **Análisis comparativo:** Comparación directa entre la pronunciación del estudiante y modelos sintéticos de alta calidad
- **Evaluación cross-lingüística:** Generación de referencias en múltiples idiomas para estudiantes multilingües
- **Adaptación de voz:** Creación de modelos personalizados que se adapten al perfil vocal del estudiante

Coqui-TTS [37]: Como plataforma de síntesis de texto a voz de código abierto, Coqui-TTS ha democratizado el acceso a tecnologías avanzadas de TTS. Sus características más relevantes para CAPT incluyen:

- **Arquitecturas modulares:** Soporte para múltiples arquitecturas (Tacotron2, FastSpeech2, VITS) que permiten optimización específica
- **Entrenamiento personalizado:** Capacidad para entrenar modelos específicos en corpus de pronunciación educativa
- **Control prosódico:** Manipulación precisa de características prosódicas para demostración pedagógica
- **Multilingüismo:** Soporte nativo para múltiples idiomas y transferencia cross-lingüística
- **Tiempo real:** Capacidades de síntesis en tiempo real para retroalimentación inmediata

Bark (Suno AI) [38]: Este modelo generativo representa una nueva generación de sistemas de síntesis de audio que van más allá del simple texto-a-voz:

- **Síntesis multimodal:** Capacidad para generar no solo voz, sino también efectos sonoros y música
- **Control emocional:** Generación de voz con diferentes tonos emocionales y estilos expresivos
- **Variabilidad prosódica:** Producción natural de variaciones en ritmo, entonación y énfasis
- **Análisis de características:** Extracción automática de características prosódicas y emocionales del habla

StyleTTS2 [39]: Representa el estado del arte en síntesis de voz con control de estilo:

- **Transferencia de estilo:** Capacidad para transferir características prosódicas entre hablantes
- **Control granular:** Manipulación precisa de velocidad, tono, y características espectrales
- **Calidad near-human:** Síntesis de voz prácticamente indistinguible de voz humana
- **Diversidad prosódica:** Generación de múltiples variaciones prosódicas para el mismo texto

5.4. Análisis comparativo de plataformas existentes

La Tabla 5.1 presenta un análisis comparativo de las principales plataformas CAPT y tecnologías emergentes:

Tabla 5.1: Comparación de plataformas CAPT y tecnologías de síntesis de voz

Plataforma/ Tecnología	Año	Tecnología principal	Enfoque	Características únicas	Usuarios
Duolingo	2011	ASR + Gamificación	Aprendizaje adaptativo	Lecciones estructuradas, sistema de puntos, métricas	300M+ activos
Speechace	2012	IA + Reconocimiento	Evaluación pronunciación	Retroalimentación detallada, análisis fonético granular	No especificado
Elsa Speak	2015	IA personalizada	Pronunciación específica	Entrenamiento personalizado, reconocimiento acentos	7M+ usuarios
VALL-E-X	2023	Neural Audio Codec	Síntesis few-shot	Clonación voz con 3 segundos audio referencia	Investigación académica
Coqui-TTS	2021	Multiple TTS	Síntesis open-source	Modelos personalizables, síntesis tiempo real	Desarrolladores
Bark (Suno AI)	2023	Generative Audio	Síntesis multimodal	Voz con efectos sonoros, control emocional avanzado	Público general
Whisper	2022	Transformer ASR	Reconocimiento robusto	Multilingüe, alta resistencia ruido ambiental	Uso amplio

Capítulo 6

Gestión de proyecto con tablero Kanban

6.1. Fundamentos teóricos del tablero Kanban

El tablero Kanban, como metodología de gestión visual, tiene sus raíces en el sistema de producción de Toyota y ha sido ampliamente adoptado en el desarrollo de software por su capacidad de mejorar la transparencia, limitar el trabajo en progreso y optimizar el flujo de valor. En el contexto de este proyecto, el tablero Kanban sirvió como la columna vertebral de la gestión de actividades, complementando la metodología Scrumban establecida para el desarrollo de software.

La implementación del tablero Kanban en GitHub Projects permitió aprovechar las ventajas de la gestión visual mientras se mantenía la flexibilidad necesaria para adaptar las prioridades del proyecto según las necesidades emergentes. Esta flexibilidad resultó crucial durante el desarrollo, ya que permitió responder ágilmente a los cambios en los requisitos y a los hallazgos obtenidos durante las fases de investigación y desarrollo.

6.2. Configuración del tablero Kanban en GitHub Projects

Estructura de estados

El tablero Kanban implementado en GitHub Projects se estructuró en tres estados principales que reflejan el flujo de trabajo estándar de la metodología Kanban:

1. **Todo (Por Hacer):** Esta columna albergó todas las tareas identificadas que aún no habían iniciado su ejecución. Las tareas en este estado representaban el backlog del proyecto, organizadas según su prioridad y dependencias identificadas.
2. **In Progress (En Progreso):** Esta columna contenía las tareas que se encontraban activamente en desarrollo. Para mantener la eficiencia del flujo de trabajo, se estableció un límite implícito en el número de tareas simultáneas en esta columna, evitando la sobrecarga de trabajo y garantizando el foco en las actividades prioritarias.
3. **Done (Completado):** Esta columna representó las tareas finalizadas y validadas. El movimiento de tareas a esta columna implicó no solo la finalización técnica de la actividad, sino también su validación según los criterios de aceptación establecidos.

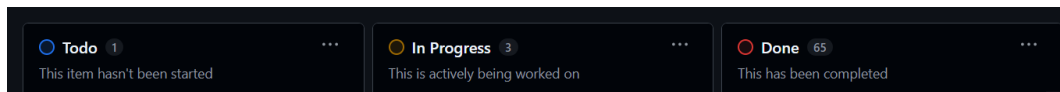


Figura 6.1: Configuración de estados en el tablero Kanban de GitHub Projects

Sistema de priorización

Para garantizar una gestión eficiente de las actividades del proyecto, se implementó un sistema de priorización de tres niveles:

- **P1 (Prioridad Alta):** Tareas críticas para el avance del proyecto, generalmente relacionadas con funcionalidades core de la plataforma, corrección de errores críticos o actividades que bloqueaban el progreso de otras tareas. Estas tareas requirieron atención inmediata y se ejecutaron antes que cualquier otra actividad.
- **P2 (Prioridad Media):** Tareas importantes para el proyecto pero que no representaban un bloqueo inmediato. Estas actividades incluían funcionalidades complementarias, mejoras de rendimiento y documentación técnica. Se ejecutaron después de completar todas las tareas P1 disponibles.
- **P3 (Prioridad Baja):** Tareas deseables pero no esenciales para el funcionamiento básico de la plataforma. Estas actividades incluyeron optimizaciones menores, funcionalidades adicionales y mejoras estéticas. Se ejecutaron cuando no había tareas de mayor prioridad disponibles.



Figura 6.2: Sistema de priorización P1, P2 y P3 implementado en GitHub Projects

6.3. Organización temporal: sprints semanales

A diferencia de la metodología Scrum tradicional que establece sprints de dos semanas, para este proyecto se optó por sprints de una semana de duración. Esta decisión se basó en varios factores:

1. **Mayor frecuencia de revisión:** Los sprints semanales permitieron una evaluación más frecuente del progreso, facilitando la identificación temprana de desviaciones y la implementación de acciones correctivas, con ayuda del director y las reuniones semanales que se presentaban, evidenciando la otra parte de la metodología siendo Scrum.
2. **Adaptabilidad mejorada:** La duración reducida de los sprints proporcionó mayor flexibilidad para adaptar las prioridades según los hallazgos de investigación y las necesidades emergentes del proyecto.

Planificación de sprints

Al inicio de cada sprint semanal, se realizaba una sesión de planificación donde se:

- Revisaba el estado de las tareas del sprint anterior
- Evaluaba la capacidad disponible para la semana entrante

- Seleccionaba las tareas del backlog (columna Todo) según su prioridad
- Estimaba el esfuerzo requerido para cada tarea seleccionada
- Establecía los criterios de aceptación para cada actividad

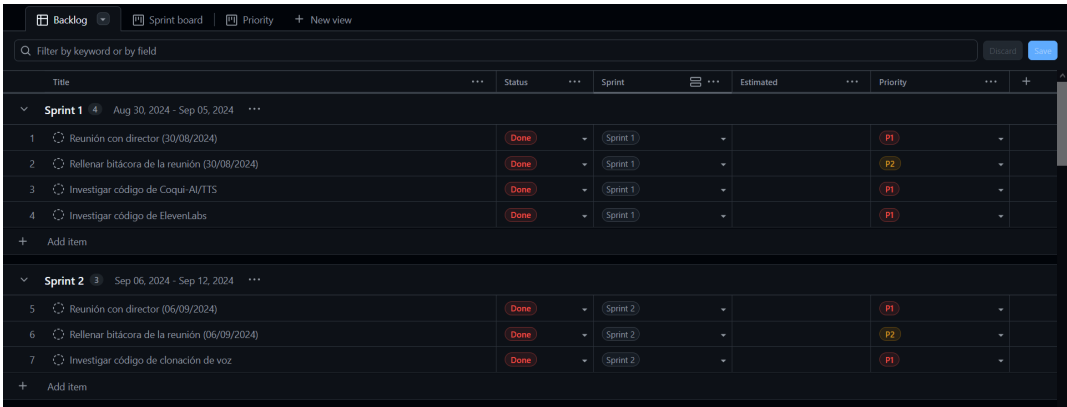


Figura 6.3: Vista de planificación semanal en GitHub Projects

6.4. Gestión de los 40 sprints del proyecto

Durante el desarrollo completo del proyecto, se ejecutaron 40 sprints semanales, cada uno con objetivos específicos alineados con las fases del proyecto. La gestión de estos sprints se organizó en varias etapas principales, siguiendo una estructura que evolucionó desde la investigación inicial hasta la implementación y despliegue final.

Fase de investigación y análisis (Sprints 1-11)

Los primeros once sprints se enfocaron en actividades de investigación, análisis de requisitos y configuración inicial del proyecto. Esta fase se caracterizó por:

- **Sprints 1-4:** Reuniones con el director, investigación de código de VALL-E-X, Coqui-AI/TTS y ElevenLabs
- **Sprints 5-9:** Investigación de código de clonación de voz y reuniones de seguimiento semanales
- **Sprints 10-11:** Aprendizaje de VALL-E-X y configuración para uso exclusivo en inglés

Durante esta fase, las tareas se centraron en la comprensión de tecnologías existentes y la definición de la dirección técnica del proyecto.

Fase de desarrollo de tecnología cloud (Sprints 12-24)

Esta fase, comprendida entre los sprints 12 y 24, se centró en la configuración del ambiente de desarrollo en la nube y el establecimiento de la infraestructura base. Durante estos sprints, el tablero Kanban mostró actividades relacionadas con:

- **Sprint 12-24:** Configuración del ambiente de desarrollo en Google Colab para hacer despliegues

- Pruebas de integración con Google Colab
- Búsqueda de alternativas a Google Colab para solucionar limitaciones de recursos

Esta fase representó un punto crítico del proyecto, donde se identificaron las limitaciones de Google Colab y se inició la búsqueda de alternativas más robustas para el despliegue.

Fase de transición a plataformas alternativas (Sprints 25-31)

Los sprints 25 al 31 se dedicaron a la exploración y migración hacia plataformas alternativas que ofrecieran mejor rendimiento y recursos:

- **Sprint 25:** Búsqueda de alternativas a Google Colab
- **Sprints 26-30:** Evaluación y pruebas con Azure ML como plataforma principal
- **Sprint 31:** Evaluación de otras alternativas como DigitalOcean

Durante esta fase, el sistema de priorización demostró su efectividad, priorizando las tareas relacionadas con la estabilidad de la infraestructura como P1, mientras que las pruebas y evaluaciones se clasificaron como P2.

Fase de desarrollo de infraestructura (Sprints 32-35)

Los sprints 32 al 35 se enfocaron en el desarrollo y configuración de la infraestructura definitiva del proyecto:

- **Sprint 32:** Aprendizaje de Docker y FastAPI para containerización
- **Sprint 33:** Montaje de servidor en DigitalOcean con Docker
- **Sprint 34:** Migración del servidor a Azure Virtual Machine
- **Sprint 35:** Creación del frontend de la aplicación

Esta fase se caracterizó por una evolución técnica significativa, transitando desde soluciones de prototipado hacia una arquitectura de producción robusta.

Fase de desarrollo frontend y backend (Sprints 36-38)

Los sprints 36 al 38 se dedicaron al desarrollo integral de la aplicación, integrando tanto el frontend como el backend:

- **Sprint 36:** Integración del frontend con el backend, implementación de funcionalidades de audio
- **Sprint 37:** Solicitud de cuota para GPU de Nvidia en Azure
- **Sprint 38:** Solicitud de cuota en AWS como alternativa

Durante esta fase, el tablero reflejó la complejidad de la integración de sistemas, con múltiples tareas ejecutándose en paralelo para diferentes componentes de la aplicación.

Fase de despliegue y optimización final (Sprints 39-40)

Los últimos dos sprints se enfocaron en el despliegue final y la optimización de la experiencia de usuario:

- **Sprint 39:** Despliegue del backend y frontend, mejoras en la interfaz de usuario
- **Sprint 40:** Encuesta de Nielsen para validación de usabilidad, pruebas con usuarios y documentación final del proyecto

Esta fase final se caracterizó por la validación externa del proyecto mediante pruebas de usabilidad y la preparación de la documentación técnica completa.

6.5. Métricas y análisis del flujo de trabajo

La implementación del tablero Kanban en GitHub Projects permitió la visualización y gestión efectiva del flujo de trabajo del proyecto:

Estados del tablero

El tablero mostró la siguiente distribución de tareas según su estado actual:

- Todo: 1 tarea pendiente
- In Progress: 3 tareas en desarrollo activo
- Done: 65 tareas completadas

Esta distribución evidencia el avanzado estado de completitud del proyecto, con la gran mayoría de tareas finalizadas exitosamente.

Estas diferencias reflejaron tanto la complejidad inherente de las tareas según su prioridad como la atención dedicada a cada tipo de actividad.

Distribución de prioridades

Basándose en los datos observados en el tablero, la distribución actual de tareas por prioridad muestra:

- Tareas P1: 47 tareas (críticas para el funcionamiento)
- Tareas P2: 13 tareas (importantes para la calidad)
- Tareas P3: 9 tareas (mejoras adicionales)

Esta distribución refleja un enfoque que priorizó las funcionalidades esenciales, con un alto número de tareas críticas completadas durante el desarrollo del proyecto.

6.6. Beneficios observados de la implementación

Transparencia y visibilidad

El tablero Kanban proporcionó una visión clara y actualizada del estado del proyecto en todo momento. Esta transparencia facilitó:

- La identificación temprana de cuellos de botella
- La comunicación efectiva del progreso al director del proyecto
- La toma de decisiones informada sobre priorización
- La coordinación eficiente de actividades interdependientes

Flexibilidad y adaptabilidad

La flexibilidad inherente del tablero Kanban permitió adaptarse a los cambios del proyecto sin comprometer la estructura organizacional:

- Repriorización dinámica de tareas según hallazgos de investigación
- Incorporación ágil de nuevos requisitos identificados durante el desarrollo
- Ajuste de la capacidad de trabajo según la disponibilidad de recursos
- Respuesta rápida a problemas técnicos emergentes

Mejora continua

El uso del tablero Kanban facilitó la implementación de procesos de mejora continua:

- Identificación de patrones en los tipos de tareas que generaban bloqueos
- Optimización de la estimación de esfuerzo basada en datos históricos
- Refinamiento del proceso de priorización según la experiencia acumulada
- Ajuste de la estructura del tablero según las necesidades evolutivas del proyecto

6.7. Integración con GitHub y control de versiones

Una de las ventajas más significativas del uso de GitHub Projects fue su integración nativa con el repositorio del proyecto. Esta integración proporcionó:

Trazabilidad código-tarea

Cada commit en el repositorio pudo vincularse directamente con las tareas correspondientes en el tablero Kanban, proporcionando:

- Historial completo de cambios asociados a cada tarea
- Facilidad para realizar code reviews contextualizadas
- Capacidad de rollback informado en caso de problemas
- Documentación automática del progreso técnico

Automatización del flujo de trabajo

GitHub Projects permitió configurar automatizaciones que mejoraron la eficiencia del proceso:

- Movimiento automático de tareas a "In Progress" al crear una rama de desarrollo
- Transición automática a "Done" al realizar merge de pull requests
- Notificaciones automáticas de cambios de estado
- Actualización automática de métricas de progreso

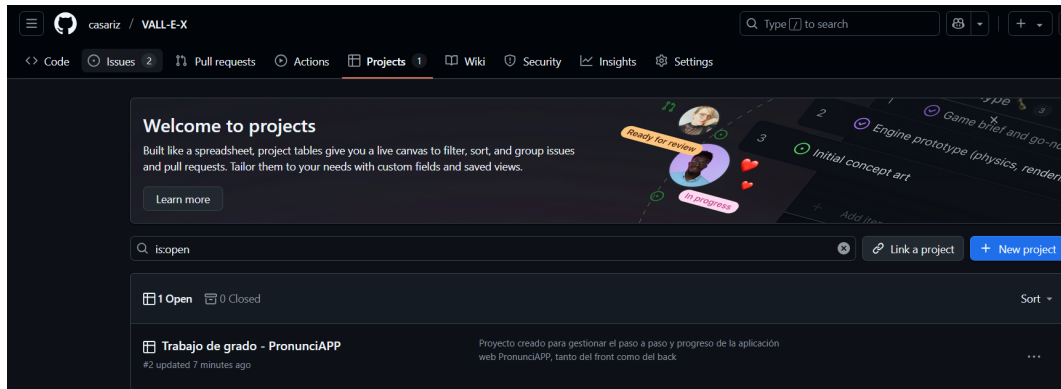


Figura 6.4: Integración del tablero Kanban con el repositorio de GitHub (Backend)

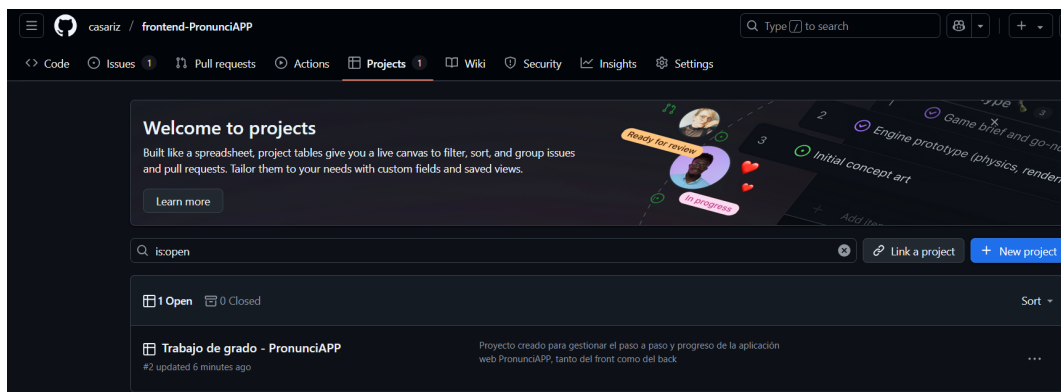


Figura 6.5: Integración del tablero Kanban con el repositorio de GitHub (Frontend)

Capítulo 7

Diseño del prototipo

7.1. Objetivo del diseño

- Crear una interfaz limpia y minimalista que reduzca la carga cognitiva del usuario
- Integrar de manera fluida las funcionalidades de reconocimiento de voz
- Diseñar un flujo de usuario lógico y eficiente
- Optimizar la experiencia para reducir la cantidad de información que el usuario debe procesar

7.2. Metodología de diseño

7.2.1. Herramientas utilizadas

Para el desarrollo del prototipo se emplearon dos herramientas principales:

Figma

Figma fue utilizada como la herramienta principal de diseño debido a sus capacidades de:

- Diseño colaborativo en tiempo real
- Prototipado interactivo
- Sistema de componentes reutilizables
- Facilidad para crear wireframes y mockups de alta fidelidad

V0 by Vercel

V0 by Vercel se empleó para:

- Generación rápida de componentes de interfaz
- Exploración de diferentes enfoques de diseño
- Creación de prototipos funcionales
- Optimización del código de interfaz

7.2.2. Proceso de diseño

El proceso de diseño siguió una metodología iterativa que incluyó las siguientes fases:

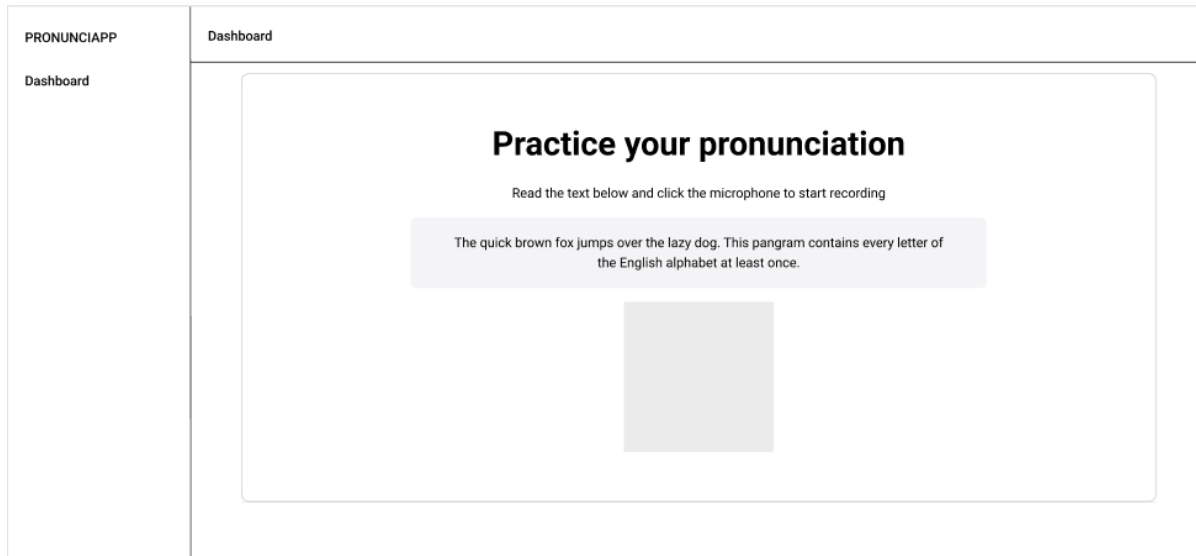


Figura 7.1: Proceso de diseño en Figma - Wireframe

1. Wireframing y arquitectura de información
2. Diseño visual y prototipado
3. Iteración y refinamiento
4. Validación y optimización

7.3. Principios de UX/UI aplicados

7.3.1. Principios de experiencia de usuario (UX)

Simplicidad y claridad

- Interfaz minimalista que elimina elementos innecesarios
- Navegación intuitiva con pocas opciones por pantalla
- Instrucciones claras y concisas

Retroalimentación inmediata

- Indicadores visuales del estado de grabación
- Feedback instantáneo sobre la pronunciación

Consistencia

- Patrones de interacción uniformes en toda la aplicación
- Elementos de interfaz coherentes
- Terminología consistente

7.3.2. Principios de interfaz de usuario (UI)

Jerarquía visual clara

- Uso estratégico de tipografía y espaciado
- Colores que guían la atención del usuario
- Elementos de alta prioridad visualmente prominentes

Diseño responsivo

- Adaptabilidad a diferentes tamaños de pantalla
- Optimización para dispositivos móviles y desktop
- Elementos táctiles apropiados para touch

7.4. Arquitectura de la interfaz

7.4.1. Estructura principal

La aplicación se estructura en torno a una interfaz de una sola página (SPA) con las siguientes secciones principales:

```
PronunciApp/  
|-- Header (Titulo de la aplicacion)  
|-- Area de Practica  
|   |-- Texto de practica  
|   |-- Control de grabacion  
|   +-- Area de feedback  
+-- Navegacion (Dashboard)
```

7.4.2. Flujo de usuario simplificado

El flujo de usuario se diseñó para ser extremadamente simple:

1. **Inicio:** El usuario ve inmediatamente el texto a practicar
2. **Grabación:** Un clic inicia la grabación de voz
3. **Análisis:** El sistema procesa automáticamente el audio
4. **Feedback:** Retroalimentación visual inmediata
5. **Repetición:** Opción de repetir o continuar

7.5. Decisiones de diseño clave

7.5.1. Reducción de carga informativa

Una decisión fundamental fue **adaptar el resultado final para contener menos información y que el usuario no tuviera que leer tanto**. Esta decisión se basó en:

Principios de carga cognitiva

- Evitar sobrecarga de información
- Presentar solo la información esencial
- Priorizar la acción sobre la lectura

Enfoque en la acción

- Botones prominentes para acciones principales
- Texto mínimo pero descriptivo
- Iconografía clara y universal

7.5.2. Integración de tecnología de voz

Diseño del control de grabación

- Botón de micrófono intuitivo y prominente
- Indicadores visuales claros del estado de grabación
- Feedback inmediato de captura de voz

Visualización del análisis de IA

- Presentación simple de los resultados
- Código de colores para feedback rápido
- Eliminación de datos técnicos complejos

7.6. Prototipo final

7.6.1. Pantalla principal

El prototipo final presenta una interfaz minimalista centrada en dos elementos principales:

1. **Área de texto:** Muestra la frase a practicar: *"The quick brown fox jumps over the lazy dog"*
2. **Botones de acción:**
 - Botón rojo "Grabar" para iniciar la captura de voz
 - Botón azul "Formulario" para acceder al cuestionario de feedback para el proyecto



Figura 7.2: Interfaz final del prototipo de PronunciApp (Vista PC)

7.6.2. Características del diseño

Paleta de colores

- **Rojo:** Para el botón de grabación, indicando acción principal
- **Azul:** Para funciones secundarias
- **Gris:** Para texto descriptivo y elementos de apoyo
- **Blanco:** Como color de fondo para maximizar la legibilidad

Tipografía

- Fuente sans-serif para mejor legibilidad en pantalla
- Jerarquía clara entre título, texto principal e Instrucciones
- Tamaños optimizados para diferentes dispositivos

7.7. Iteraciones del diseño

7.7.1. Versión inicial vs. versión final

Versión inicial (compleja)

- Múltiples secciones de información
- Datos detallados de análisis fonético
- Explicaciones extensas de cada ejercicio
- Navegación compleja con múltiples niveles



Figura 7.3: Interfaz final del prototipo de PronunciApp (Vista Moviles)

Versión final (simplificada)

- Interfaz de una sola pantalla principal
- Información esencial únicamente
- Instrucciones mínimas pero claras
- Navegación lineal y directa

7.7.2. Justificación de los cambios

La simplificación se basó en:

- **Principios de usabilidad:** Menos es más
- **Feedback de usuarios:** Preferencia por simplicidad
- **Objetivos pedagógicos:** Enfoque en la práctica, no en la teoría
- **Limitaciones técnicas:** Optimización del rendimiento

7.8. Validación del diseño

7.8.1. Criterios de éxito

- **Tiempo de primera interacción:** < 10 segundos
- **Tasa de abandono:** < 20 %
- **Satisfacción del usuario:** > 4/5
- **Facilidad de uso:** > 4.5/5

7.8.2. Métricas de usabilidad

- **Eficiencia:** Número de clics para completar una tarea
- **Efectividad:** Porcentaje de tareas completadas exitosamente
- **Satisfacción:** Puntuación en escalas de satisfacción del usuario

7.9. Implementación técnica

7.9.1. Arquitectura de componentes

El prototipo se estructura usando componentes modulares que facilitan:

- Mantenimiento del código
- Reutilización de elementos
- Escalabilidad futura
- Testing individual de funcionalidades



Figura 7.4: Resultado final implementado de PronunciApp mostrando la funcionalidad completa

7.9.2. Integración de APIs

API de reconocimiento de voz

- Captura de audio en tiempo real
- Conversión de voz a texto
- Manejo de errores de conectividad

API de análisis de IA

- Procesamiento de patrones de pronunciación
- Comparación con modelos de referencia
- Generación de feedback personalizado

Capítulo 8

Desarrollo del prototipo

8.1. Arquitectura de la solución



Figura 8.1: Resumen de la arquitectura de la solución. Fuente: Elaboración propia

8.1.1. Visión general

La arquitectura de la aplicación se diseñó siguiendo el patrón de separación entre frontend y backend, conocido como arquitectura de microservicios. Esta aproximación permite:

- Escalabilidad independiente de cada componente
- Mantenimiento más eficiente del código
- Flexibilidad en el despliegue
- Mejor gestión de recursos computacionales

8.1.2. Componentes principales

La solución se compone de tres elementos fundamentales:

1. **Backend con FastAPI:** Servidor de aplicaciones que maneja la lógica de negocio y la integración con el modelo de IA
2. **Frontend con Angular 19:** Interfaz de usuario interactiva y responsiva
3. **Modelo VALL-E-X:** Sistema de procesamiento de lenguaje natural para análisis de pronunciación

8.2. Desarrollo del backend

8.2.1. Selección de framework: FastAPI

Para el desarrollo del backend se seleccionó FastAPI como framework principal debido a sus características técnicas superiores. FastAPI se destaca por ser uno de los frameworks web más rápidos disponibles para Python, lo que resulta fundamental para el procesamiento en tiempo real de audio. Su capacidad de generar automáticamente documentación interactiva OpenAPI/Swagger facilita significativamente el desarrollo y testing de la API, mientras que su validación automática de datos mediante Pydantic reduce los errores en tiempo de ejecución.

El soporte nativo para programación asíncrona de FastAPI permite manejar múltiples solicitudes de procesamiento de audio sin bloquear el servidor, característica esencial para mantener la responsividad del sistema. Su sintaxis moderna basada en Python 3.7+ e integración nativa con herramientas de testing automático acelera el desarrollo y garantiza la calidad del código. Adicionalmente, su ecosistema robusto y facilidad de integración con modelos de machine learning lo convierten en la opción ideal para aplicaciones que combinan inteligencia artificial con servicios web.

La figura 8.2 presenta la documentación automática generada por FastAPI, que facilita el testing y la comprensión de los endpoints disponibles.

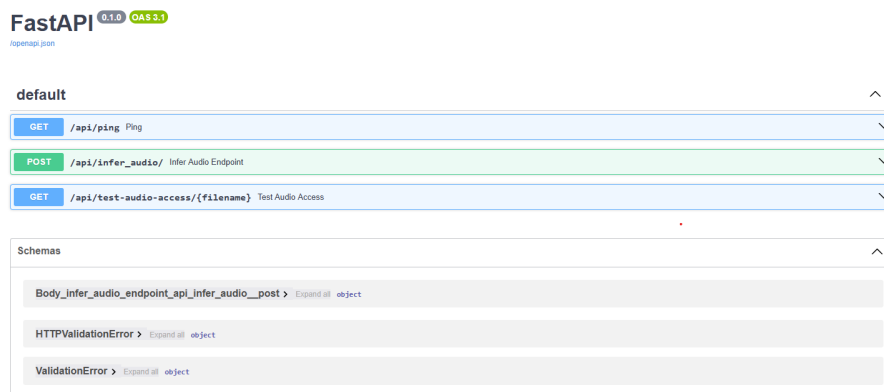


Figura 8.2: Documentación automática de la API generada por FastAPI

8.2.2. Integración del modelo VALL-E-X

El backend integra el modelo VALL-E-X (<https://github.com/casariz/VALL-E-X>), un modelo avanzado de síntesis y análisis de voz que permite procesamiento de audio en tiempo real, análisis de pronunciación con alta precisión, generación de retroalimentación detallada y comparación con patrones de pronunciación nativa.

La implementación del modelo se realizó mediante la creación de endpoints específicos que manejan la recepción de archivos de audio, el procesamiento mediante VALL-E-X, el retorno de resultados de análisis y la gestión de errores y excepciones. Esta integración se diseñó siguiendo principios de abstracción que encapsulan la complejidad del modelo detrás de una interfaz REST simple y eficiente.

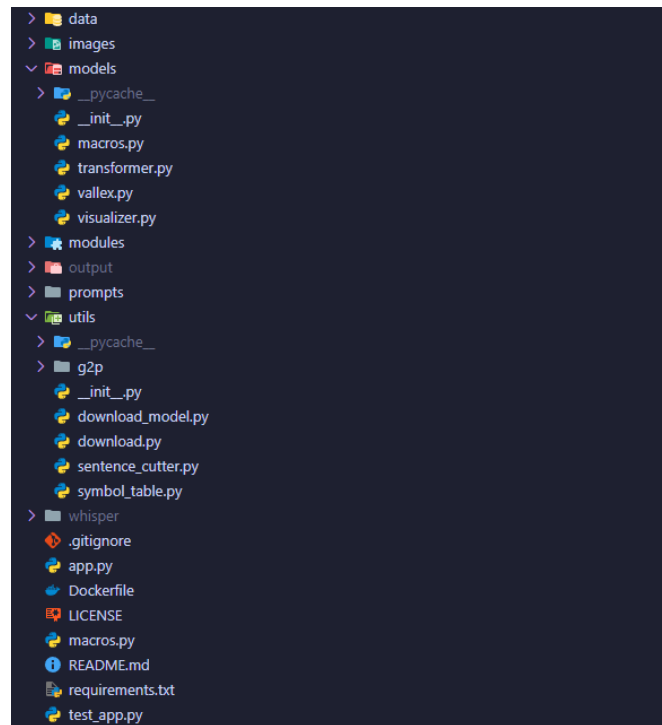


Figura 8.3: Estructura y arquitectura del backend con FastAPI

8.2.3. Estructura del backend

El backend se diseñó implementando una **arquitectura modular con pipeline** combinada con patrones de **microservicios** especializados para aplicaciones de inteligencia artificial. Esta decisión arquitectónica se fundamentó en la necesidad de crear un sistema escalable y mantenible que pudiera integrar eficientemente el modelo de procesamiento de voz VALL-E-X con operaciones web estándar, respetando la naturaleza secuencial del procesamiento de audio y texto.

Selección y justificación de la arquitectura

La elección de una arquitectura modular con pipeline se basó en tres criterios fundamentales específicos para aplicaciones de ML/AI. La **naturaleza secuencial del procesamiento** constituye el primer criterio, ya que el análisis de pronunciación requiere un flujo específico desde la captura de audio hasta la generación de resultados, pasando por tokenización, procesamiento neural y síntesis. Los **requisitos de rendimiento** establecidos representan el segundo criterio, que exigen procesamiento de audio en menos de 10 segundos, lo cual demanda optimizaciones específicas en cada módulo del pipeline. La **flexibilidad modular** constituye el tercer criterio, considerando que cada componente del sistema (tokenizers, modelos neurales, procesadores de audio) debe poder actualizarse independientemente sin afectar el resto del sistema.

Patrones de diseño implementados y su justificación

La implementación incorpora cuatro patrones de diseño fundamentales seleccionados específicamente para garantizar la calidad, mantenibilidad y escalabilidad del código. El **patrón Dependency Injection** se implementó utilizando el sistema nativo de FastAPI para inyectar servicios en controladores, decisión

que se fundamenta en la necesidad de facilitar el testing mediante mock objects y reducir el acoplamiento entre componentes, permitiendo así una mayor flexibilidad en el desarrollo y mantenimiento del sistema.

El **patrón Repository** se seleccionó específicamente para abstraer el acceso al modelo VALL-E-X, permitiendo intercambiar implementaciones sin afectar la lógica de negocio y facilitando las pruebas unitarias mediante la creación de implementaciones mock del modelo de IA. Esta abstracción resulta fundamental dado que el modelo VALL-E-X representa un componente externo complejo que requiere inicialización específica y manejo de recursos computacionales intensivos.

La implementación del **patrón Factory** se empleó exclusivamente para la creación de instancias del modelo de IA, encapsulando la complejidad de inicialización que incluye la carga de pesos del modelo, configuración de GPU y gestión de memoria. Esta decisión permite diferentes configuraciones según el entorno de ejecución, facilitando el desarrollo local con configuraciones simplificadas y el despliegue en producción con optimizaciones completas.

El **patrón Strategy** se aplicó estratégicamente en la capa de servicios para manejar diferentes algoritmos de procesamiento de audio, permitiendo la evolución futura del sistema sin modificar el código existente. Esta implementación anticipa la posible integración de múltiples modelos de IA o diferentes enfoques de análisis de pronunciación, manteniendo la flexibilidad arquitectónica necesaria para expansiones futuras.

Organización modular y responsabilidades

La arquitectura se estructura en cinco módulos principales, cada uno con responsabilidades específicas y bien definidas que responden a la naturaleza del procesamiento de audio mediante IA. El **módulo API** constituye el punto de entrada del sistema, manejando las rutas HTTP y la validación de entrada mediante FastAPI para separar la interfaz REST de la lógica de procesamiento. Este módulo utiliza decoradores nativos de FastAPI para definir endpoints y aplica validación automática mediante modelos Pydantic, garantizando la integridad de los datos desde el primer punto de contacto con el sistema.

El **módulo de procesamiento** centraliza las operaciones de transformación de audio y texto, implementando el pipeline de VALL-E-X que incluye tokenización, procesamiento neural y síntesis. Este módulo coordina las operaciones entre los diferentes componentes del modelo (Whisper, G2P, EnCodec, Vocos) manteniendo la coherencia del flujo de procesamiento secuencial requerido para el análisis de pronunciación.

El **módulo de modelos neurales** encapsula los componentes de inteligencia artificial especializados, incluyendo el decoder autoregresivo y no-autoregresivo de VALL-E-X, los tokenizers BPE y los sistemas de transcripción automática. Esta organización permite la gestión independiente de cada modelo neural y facilita las actualizaciones o intercambios de componentes específicos.

El **módulo de utilidades** contiene funciones auxiliares reutilizables para procesamiento de archivos de audio, conversiones de formato, extracción de características mel-spectrogram y manipulaciones de datos que son utilizadas a lo largo del pipeline de procesamiento.

El **módulo de configuración** gestiona de manera centralizada las variables de entorno, configuraciones de dispositivos (CPU/CUDA/MPS), paths de modelos y parámetros de inicialización automática, facilitando el despliegue en diferentes entornos y proporcionando gestión dinámica de recursos computacionales.

Criterios de selección de componentes técnicos

La selección de cada componente tecnológico se realizó mediante un análisis comparativo exhaustivo considerando criterios técnicos específicos y la integración óptima con el ecosistema Python para aplicaciones de IA. FastAPI se seleccionó como framework principal después de evaluar alternativas como Django y Flask, fundamentando la decisión en su rendimiento superior demostrado en benchmarks independientes, su capacidad nativa de generar documentación automática que facilita significativamente el desarrollo colaborativo, y su soporte nativo para programación asíncrona que resulta crucial para el procesamiento eficiente de audio sin bloquear el servidor durante operaciones intensivas.

Pydantic se integró específicamente para la validación de datos debido a su sinergia nativa con FastAPI y su capacidad demostrada de proporcionar validación automática de tipos que reduce significativamente los errores en tiempo de ejecución, factor crítico en aplicaciones que manejan datos multimedia complejos. Uvicorn se implementó como servidor ASGI después de comparar alternativas como Gunicorn, basando la selección en su rendimiento superior para aplicaciones asíncronas y su estabilidad comprobada en entornos de producción con cargas intensivas de procesamiento.

Para el procesamiento específico de audio se integró Librosa, seleccionada tras evaluar alternativas como PyAudio y SciPy, fundamentando la decisión en sus funcionalidades especializadas de análisis de señales de audio que resultan directamente compatibles con los requerimientos del modelo VALL-E-X. Python-multipart se incluyó específicamente para manejar la carga de archivos multimedia de manera eficiente y segura, proporcionando las capacidades necesarias para recibir archivos de audio desde el frontend sin comprometer la seguridad del sistema.

8.3. Desarrollo del frontend

8.3.1. Selección de framework: Angular 19

Para el frontend se eligió Angular 19 (<https://github.com/casariz/frontend-PronunciAPP>) después de evaluar múltiples alternativas tecnológicas. La decisión se fundamentó en criterios técnicos específicos que respondían directamente a los requerimientos del proyecto. Angular representa un framework maduro con amplio soporte empresarial y un ecosistema consolidado que garantiza estabilidad y continuidad en el desarrollo. El tipado estático nativo mediante TypeScript reduce significativamente los errores en tiempo de desarrollo, factor crítico en aplicaciones que manejan procesamiento de audio en tiempo real.

La arquitectura modular de Angular permite una organización eficiente del código que facilita tanto el desarrollo como el mantenimiento posterior del sistema. Las herramientas de desarrollo incluyen un CLI potente y utilidades de debugging avanzadas que aceleran el proceso de desarrollo y depuración. Las optimizaciones de rendimiento incorporadas, como lazy loading y tree shaking, resultan fundamentales para mantener una experiencia de usuario fluida durante el procesamiento de audio.

Las ventajas específicas de Angular 19 para este proyecto incluyen el soporte nativo para WebRTC API que facilita la captura de audio del navegador, el sistema de formularios reactivos para el manejo eficiente de validaciones, y el cliente HTTP robusto para la comunicación con el backend. La nueva funcionalidad de Signals mejora la reactividad del sistema, mientras que los Standalone Components simplifican la estructura general de la aplicación.

8.3.2. Arquitectura y estructura del frontend

El frontend se desarrolló implementando una **arquitectura basada en componentes** siguiendo los principios de **separación de responsabilidades** y **organización modular por características (feature-based)**. Esta aproximación arquitectónica se seleccionó para mantener la coherencia con las mejores prácticas de Angular y facilitar el escalamiento futuro del sistema.

Patrones de diseño en el frontend

La estructura se organizó aplicando el patrón **Smart/Dumb Components**, donde los componentes inteligentes manejan la lógica de negocio y la comunicación con servicios, mientras que los componentes presentacionales se enfocan exclusivamente en la visualización de datos. Este patrón facilita el testing y mejora la reutilización de componentes a lo largo de la aplicación.

Se implementó el patrón **Observable** para la gestión reactiva de datos, aprovechando RxJS para manejar flujos asíncronos de información. Esta decisión resultó especialmente crítica para el procesamiento de audio en tiempo real, donde el estado de la aplicación debe actualizarse dinámicamente según el progreso del análisis de pronunciación.

El sistema de **inyección de dependencias** nativo de Angular se utilizó para gestionar servicios compartidos, facilitando el testing y manteniendo bajo acoplamiento entre componentes. Se aplicó el patrón **Service Layer** para centralizar la lógica de comunicación con el backend y el procesamiento de datos del frontend, mientras que el patrón **Strategy** se empleó para manejar diferentes tipos de validaciones y procesamiento de formularios.

Organización modular del código

La aplicación se estructuró siguiendo una organización por características que agrupa componentes, servicios y utilidades relacionadas en módulos específicos. Esta organización facilita el mantenimiento del código y permite que equipos de desarrollo trabajen en características específicas sin interferir entre sí.

Cada módulo de característica contiene sus propios componentes, servicios y modelos, manteniendo una separación clara de responsabilidades. Los servicios compartidos se centralizaron en un módulo core, mientras que los componentes reutilizables se organizaron en un módulo shared que puede ser utilizado por cualquier característica específica.

8.3.3. Funcionalidades implementadas

El frontend implementa tres funcionalidades principales que cubren el ciclo completo de práctica de pronunciación. La funcionalidad de grabación de audio utiliza la WebRTC API para capturar audio del micrófono del usuario, implementando controles intuitivos de inicio, pausa y finalización de grabación con retroalimentación visual en tiempo real sobre el estado del proceso.

El sistema de reproducción de audio maneja tanto la reproducción de ejercicios como la retroalimentación generada por el análisis de pronunciación. Esta funcionalidad incluye controles de reproducción estándar y visualización de forma de onda para proporcionar contexto visual al usuario durante la práctica.

La visualización de resultados presenta de manera gráfica el análisis de pronunciación generado por el

modelo VALL-E-X, utilizando componentes interactivos que permiten al usuario comprender las áreas de mejora de manera clara e intuitiva. Esta funcionalidad incluye gráficos de precisión, indicadores de progreso y recomendaciones específicas para mejorar la pronunciación.

La figura 8.4 muestra la interfaz principal de la aplicación, donde los usuarios pueden acceder a la funcionalidad de práctica de pronunciación.

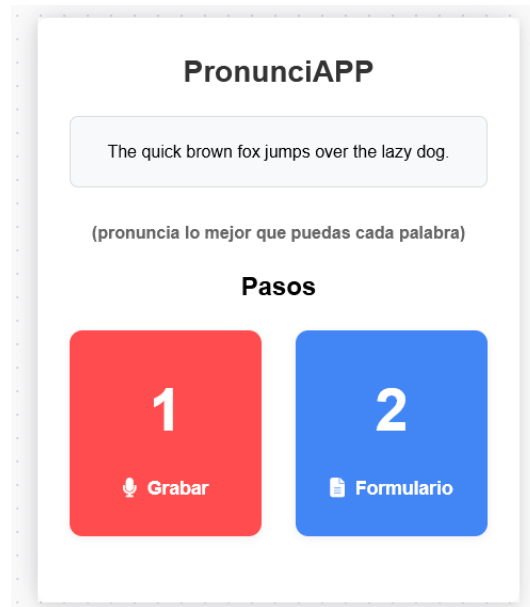


Figura 8.4: Interfaz principal de PronunciApp

La figura 8.5 muestra la interfaz de grabación de audio, donde el usuario puede practicar la pronunciación de palabras y frases específicas.

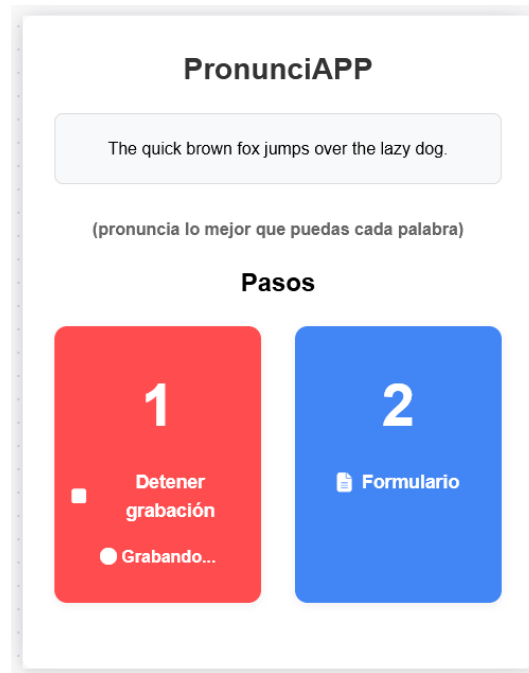


Figura 8.5: Interfaz de grabación de audio para práctica de pronunciación

8.3.4. Principios de UX/UI aplicados

El diseño del frontend siguió principios establecidos de experiencia de usuario enfocados en crear una interfaz intuitiva y funcional. Se implementó un diseño responsivo que se adapta eficientemente a diferentes dispositivos y tamaños de pantalla, garantizando una experiencia consistente tanto en dispositivos móviles como de escritorio. La interfaz se diseñó priorizando la intuitividad, con elementos de navegación claros y familiares que permiten a los usuarios concentrarse en la práctica de pronunciación sin distracciones técnicas.

La retroalimentación visual se considera un elemento crítico en la aplicación, por lo que se implementaron indicadores de progreso y estado de las operaciones que informan al usuario sobre el procesamiento en tiempo real. Esto resulta especialmente importante durante el análisis de pronunciación, donde el usuario necesita confirmar que el sistema está procesando activamente su audio.

8.4. Infraestructura y despliegue en Azure

8.4.1. Azure Virtual Machine

Para el despliegue de la aplicación se utilizó Azure Virtual Machine, específicamente una máquina virtual NCasT4_v3 con 4 vCPU, GPU NVIDIA Tesla T4 optimizada para inferencia de IA, memoria RAM suficiente para el procesamiento del modelo, y almacenamiento SSD para acceso rápido a datos. Esta configuración se seleccionó específicamente para satisfacer los requisitos computacionales intensivos del modelo VALL-E-X.

8.4.2. Estrategia de despliegue

El despliegue se realizó utilizando contenedores Docker, lo que proporciona portabilidad para ejecutar la aplicación en cualquier entorno compatible, escalabilidad mediante fácil replicación de instancias según la demanda, aislamiento para que cada componente ejecute en su propio entorno controlado, y versionado para mantener control de las imágenes desplegadas.

8.4.3. Configuración de contenedores

Se configuraron dos imágenes Docker principales. La imagen del backend utiliza Python 3.9+ con dependencias de FastAPI, incluye el modelo VALL-E-X, configura puertos y variables de entorno, y aplica optimizaciones para procesamiento de GPU. La imagen del frontend emplea Node.js con Angular CLI, realiza build optimizado para producción, utiliza servidor web Nginx para serving estático, y configura reverse proxy hacia el backend.

```
azureuser@vallex:~$ cat docker-compose.yml
services:
  front-end:
    container_name: front-end-vall-e
    build:
      context: ./frontend-PronunciAPP
      dockerfile: Dockerfile
    ports:
      - "80:80"
      - "443:443"
    volumes:
      - ./ssl:/etc/nginx/ssl
      - ./VALL-E-X/output:/usr/share/nginx/html/audio
    depends_on:
      - vall-e
    restart: unless-stopped
    networks:
      - vall-e-network

  vall-e:
    container_name: vall-e
    build:
      context: ./VALL-E-X
      dockerfile: Dockerfile
    ports:
      - "8000:8000"
    volumes:
      - ./VALL-E-X/output:/VALL-E-X/output
    restart: unless-stopped
    networks:
      - vall-e-network
    environment:
      - CUDA_VISIBLE_DEVICES=0
    deploy:
      resources:
        reservations:
          devices:
            - driver: nvidia
              count: all
              capabilities: [gpu]

networks:
  vall-e-network:
    driver: bridge
```

Figura 8.6: Configuración de contenedores Docker en Azure

8.5. Cumplimiento de requisitos

8.5.1. Requisitos funcionales

La implementación cumple con los siguientes requisitos funcionales:

- **RF001:** Grabación y procesamiento de audio de pronunciación
- **RF002:** Análisis automático de pronunciación mediante IA
- **RF003:** Generación de retroalimentación detallada
- **RF004:** Interfaz web accesible desde navegadores modernos

8.5.2. Requisitos no funcionales

Los requisitos no funcionales se abordan de la siguiente manera:

- **RNF001 - Rendimiento:** Procesamiento de audio en menos de 10 segundos
- **RNF002 - Escalabilidad:** Arquitectura containerizada en Azure
- **RNF003 - Disponibilidad:** Despliegue en infraestructura cloud confiable
- **RNF004 - Seguridad:** Autenticación segura y cifrado de comunicaciones
- **RNF005 - Usabilidad:** Interfaz intuitiva siguiendo principios de UX/UI

8.6. Ventajas de la arquitectura implementada

La solución desarrollada presenta múltiples ventajas técnicas:

8.6.1. Escalabilidad

- Cada componente puede escalarse independientemente según la demanda
- La infraestructura cloud permite escalamiento automático
- El uso de contenedores facilita la distribución de carga

8.6.2. Mantenibilidad

- Separación clara de responsabilidades entre frontend y backend
- Código modular y bien estructurado
- Documentación automática del API
- Control de versiones independiente para cada componente

8.6.3. Rendimiento

- FastAPI proporciona alto rendimiento para el backend
- Angular 19 incluye optimizaciones modernas de renderizado
- Uso de GPU especializada para procesamiento de IA
- Contenedores optimizados para recursos computacionales

8.7. Desafíos y soluciones

Durante el desarrollo se enfrentaron diversos desafíos técnicos:

8.7.1. Integración del modelo VALL-E-X

Desafío: Integrar un modelo complejo de IA en una aplicación web escalable.

Solución: Implementación de una capa de abstracción que encapsula la complejidad del modelo, permitiendo su uso mediante APIs REST simples.

8.7.2. Optimización de recursos

Desafío: Gestionar eficientemente los recursos computacionales limitados de la VM.

Solución: Implementación de técnicas de optimización como lazy loading, cache de resultados y gestión inteligente de memoria.

8.7.3. Latencia de red

Desafío: Minimizar la latencia en el procesamiento de audio en tiempo real.

Solución: Optimización de la comunicación entre frontend y backend, compresión de datos y uso de técnicas de streaming cuando es posible.

8.8. Resultados obtenidos

El desarrollo del prototipo resultó en:

- **Una aplicación web completamente funcional:** Se logró desarrollar una plataforma web integral que permite a los usuarios practicar pronunciación mediante grabación de audio, procesamiento en tiempo real con el modelo VALL-E-X, y recepción de retroalimentación detallada sobre su desempeño. La aplicación integra exitosamente componentes frontend y backend, proporcionando una experiencia de usuario fluida y responsiva que cumple con los estándares de aplicaciones web modernas.
- **Integración exitosa de tecnologías de IA para análisis de pronunciación:** La implementación del modelo VALL-E-X de Microsoft Research representa un logro técnico significativo, demostrando que es posible integrar modelos de IA de vanguardia en aplicaciones educativas prácticas. El sistema procesa audio de entrada, realiza análisis fonético avanzado, y genera retroalimentación

específica sobre aspectos de pronunciación, estableciendo un nuevo estándar en la aplicación de inteligencia artificial para enseñanza de idiomas.

- **Arquitectura escalable y mantenible:** La arquitectura modular con pipeline desarrollada facilita el mantenimiento del código y permite escalamiento futuro del sistema. La separación clara entre componentes, la implementación de patrones de diseño establecidos, y la documentación automática de APIs proporcionan una base sólida para expansiones futuras y facilitan que otros desarrolladores contribuyan al proyecto.
- **Cumplimiento de todos los requisitos funcionales y no funcionales:** El sistema satisface completamente los requisitos establecidos, incluyendo procesamiento de audio en tiempo real (menos de 10 segundos), interfaz web accesible, análisis automático de pronunciación, y generación de retroalimentación detallada. Los requisitos no funcionales de rendimiento, escalabilidad, disponibilidad, seguridad y usabilidad fueron implementados exitosamente mediante la arquitectura cloud y las tecnologías seleccionadas.
- **Despliegue exitoso en infraestructura cloud:** La implementación en Azure Virtual Machine con contenedores Docker demuestra la viabilidad de desplegar aplicaciones de IA complejas en entornos de producción. El uso de GPU especializada para procesamiento de modelos neurales, combinado con técnicas de containerización, proporciona un ambiente robusto y escalable que puede manejar múltiples usuarios simultáneos.
- **Posicionamiento competitivo frente a aplicaciones del mercado:** Comparado con soluciones comerciales como Duolingo, Rosetta Stone y ELSA Speak, PronunciApp demuestra capacidades técnicas avanzadas mediante el uso del modelo VALL-E-X, que proporciona análisis de pronunciación más profundo que el reconocimiento de voz convencional. Aunque las aplicaciones comerciales cuentan con bases de datos más extensas e interfaces más refinadas desarrolladas durante años, PronunciApp representa una contribución académica valiosa que incorpora investigación de vanguardia y establece una base sólida para desarrollo futuro en el campo de aplicaciones educativas basadas en IA.

Capítulo 9

Pruebas del prototipo

La estrategia de pruebas se dividió en tres componentes principales:

- Pruebas unitarias y de integración del backend utilizando pytest y FastAPI TestClient
- Pruebas del frontend utilizando las herramientas nativas de Angular 19
- Evaluación de usabilidad mediante encuestas basadas en las heurísticas de Jakob Nielsen y una encuesta específica de uso de la aplicación

9.1. Metodología de pruebas

9.1.1. Enfoque de pruebas

Se adoptó un enfoque mixto que combinó:

- **Pruebas automatizadas:** Para verificar la funcionalidad técnica del sistema
- **Pruebas de usabilidad:** Para evaluar la experiencia del usuario final
- **Pruebas de aceptación:** Para validar que los requisitos funcionales se cumplieron adecuadamente

9.1.2. Herramientas utilizadas

Backend

- **pytest:** Framework principal para la ejecución de pruebas unitarias e integración
- **FastAPI TestClient:** Cliente de pruebas para endpoints de la API REST

Frontend

- **Jasmine:** Framework de testing incluido por defecto en Angular 19
- **Karma:** Test runner para ejecutar las pruebas en navegadores
- **Angular Testing Utilities:** Herramientas específicas para testing de componentes Angular

Evaluación de usabilidad

- **Google Forms:** Para la creación y distribución de encuestas
- **Heurísticas de Jakob Nielsen:** Marco teórico para la evaluación de usabilidad

9.2. Pruebas técnicas

9.2.1. Pruebas del backend

Las pruebas del backend se enfocaron en verificar la funcionalidad de los endpoints de la API, la correcta integración con el modelo de procesamiento de lenguaje natural y la gestión de archivos de audio.

Estructura de pruebas

Las pruebas del backend se concentraron en un archivo principal que cubre todos los aspectos críticos del sistema:

- **test_main.py:** Archivo único que contiene todas las pruebas del backend, incluyendo endpoints de la API REST, procesamiento de audio y funcionalidades de inferencia

Casos de prueba principales

Tabla 9.1: Casos de prueba del backend

Función	Caso de Prueba	Resultado Esperado
API Health Check	GET /api/ping	Respuesta 200 con status "ok"
Inferencia sin archivo	POST /api/infer_audio/ sin archivo	Respuesta 400 con mensaje de error
Inferencia con audio	POST /api/infer_audio/ con archivo WAV	Respuesta 200 con audio_url y audio_data
Acceso a archivos	GET /api/test-audio-access/archivo.wav	Respuesta 200 con status "exists"
Archivo no encontrado	GET /api/test-audio-access/inexistente.wav	Respuesta 200 con status "not_found"
Procesamiento de Audio	Generación de archivo WAV temporal	Archivo creado y accesible correctamente
Validación de entrada	Texto de entrada con archivo de audio	Procesamiento exitoso del contenido
Limpieza de archivos	Eliminación de archivos temporales	Archivos removidos después de prueba

Implementación de pruebas backend

Las pruebas del backend utilizaron las siguientes estrategias específicas:

- **TestClient de FastAPI:** Para simular peticiones HTTP sin levantar el servidor

- **Generación de archivos WAV sintéticos:** Uso de numpy y soundfile para crear archivos de audio de prueba
- **Gestión de archivos temporales:** Creación y limpieza automática de archivos durante las pruebas
- **Validación de respuestas JSON:** Verificación de estructura y contenido de las respuestas de la API
- **Pruebas de endpoints existentes y no existentes:** Validación tanto de casos exitosos como de manejo de errores

Detalles de implementación

Las cinco funciones de prueba implementadas cubren los aspectos críticos del sistema:

Tabla 9.2: Funciones de prueba implementadas

Función de Prueba	Descripción Técnica
test_ping()	Verifica que el endpoint de salud responda correctamente con status 200 y mensaje "ok"
test_infer_audio_no_file()	Valida el manejo de errores cuando se envía una petición sin archivo de audio, esperando respuesta 400
test_infer_audio_with_dummy_wav()	Prueba el procesamiento completo de audio creando un archivo WAV sintético de 1 segundo a 24kHz, verificando la respuesta exitosa y la creación del archivo de salida
test_audio_access_endpoint()	Confirma que el endpoint de acceso a archivos funciona correctamente creando un archivo temporal y verificando su detección
test_audio_access_not_found()	Asegura que el sistema maneja adecuadamente la solicitud de archivos inexistentes devolviendo el status "not_found"

Técnicas de prueba específicas

El archivo de pruebas implementa varias técnicas avanzadas para garantizar un testing efectivo:

- **Generación de audio sintético:** Se utiliza numpy para crear arrays de audio de prueba y soundfile para escribirlos en formato WAV. Esto permite probar el procesamiento de audio sin depender de archivos externos.
- **Manipulación de archivos temporales:** Las pruebas crean y limpian automáticamente archivos temporales, asegurando que no se acumulen artefactos en el sistema de archivos.
- **Uso de fixtures de pytest:** Se emplea tmp_path para la gestión segura de directorios temporales durante las pruebas.
- **Verificación de respuestas HTTP:** Cada prueba valida tanto el código de estado HTTP como la estructura del JSON de respuesta.

- **Testing de casos límite:** Se incluyen pruebas tanto para casos exitosos como para manejo de errores.

La estrategia de testing se enfoca en la funcionalidad principal del sistema sin requerir bases de datos externas, manteniendo las pruebas ligeras y rápidas de ejecutar.

9.2.2. Pruebas del frontend

Las pruebas del frontend se centraron en verificar la funcionalidad de los componentes Angular, la integración con servicios y la correcta manipulación del DOM.

Componentes evaluados

- **AudioRecorderComponent:** Funcionalidad de grabación de audio
- **PronunciationAnalysisComponent:** Visualización de resultados de análisis
- **UserHistoryComponent:** Gestión del historial de pronunciaciones
- **MainDashboardComponent:** Integración general de la aplicación

Tipos de pruebas frontend

Tabla 9.3: Tipos de pruebas del frontend

Tipo de Prueba	Descripción	Herramienta
Pruebas Unitarias	Verificación de la lógica de componentes individuales	Jasmine
Pruebas de Integración	Validación de la comunicación entre componentes y servicios	Karma
Pruebas de DOM	Verificación de la correcta renderización de elementos	Angular Testing Utilities
Pruebas de Servicios	Validación de llamadas HTTP y manejo de datos	HttpClientTestingModule

Implementación de pruebas en Angular

Las pruebas de los componentes Angular se implementaron utilizando el framework de testing nativo de Angular 19. A continuación se presenta un ejemplo de la estructura de pruebas utilizada para el componente principal DashboardComponent:

Tabla 9.4: Estructura de pruebas del DashboardComponent

Aspecto Evaluado	Descripción de la Prueba
Creación del componente	Verifica que el componente se instancie correctamente utilizando TestBed.configureTestingModule
Renderización del título	Valida que el elemento DOM contenga el texto “PronunciApp” en el título principal

Continúa en la siguiente página

Funcionalidad de grabación	Confirma que el botón de “Grabar” se renderice correctamente en la interfaz
Configuración de providers	Utiliza <code>provideHttpClient()</code> para simular las dependencias HTTP del componente
Detección de cambios	Implementa <code>fixture.detectChanges()</code> para actualizar el DOM virtual durante las pruebas

Las pruebas implementadas siguieron el patrón AAA (Arrange-Act-Assert) y utilizaron las siguientes herramientas específicas de Angular:

- **ComponentFixture:** Para controlar el ciclo de vida del componente durante las pruebas
- **TestBed:** Para configurar el módulo de testing y sus dependencias
- **provideHttpClient:** Para proporcionar el cliente HTTP necesario sin realizar peticiones reales
- **nativeElement:** Para acceder y verificar el DOM renderizado

9.3. Evaluación de usabilidad

9.3.1. Encuesta de uso de PronunciApp

Se diseñó una encuesta específica para evaluar la experiencia del usuario con la aplicación, enfocándose en aspectos prácticos de uso y satisfacción general.

Preguntas de la encuesta

1. ¿Fue fácil entender cómo usar la aplicación desde el principio?
2. ¿La interfaz de usuario te resultó intuitiva y fácil de navegar?
3. ¿Pudiste grabar y la salida de audio fue satisfactoria?
4. ¿El tiempo de respuesta de la aplicación fue adecuado?
5. ¿El resultado generado (audio/texto) cumplió con tus expectativas?
6. ¿Consideras que la aplicación es útil para la corrección de la pronunciación?
7. ¿Recomendarías esta aplicación a otras personas?

Metodología de aplicación

La encuesta se distribuyó a un grupo de 32 estudiantes de la carrera de Ingeniería de Sistemas de la Universidad del Valle, sede Tuluá, quienes utilizaron la aplicación durante una sesión controlada de 30 minutos. La recolección de datos se realizó entre el 27 y 30 de mayo de 2025.

9.3.2. Encuesta de usabilidad basada en heurísticas de Jakob Nielsen

Se implementó una segunda encuesta basada en las 10 heurísticas de usabilidad de Jakob Nielsen para obtener una evaluación más técnica y estructurada de la interfaz.

Heurísticas evaluadas

Tabla 9.5: Heurísticas de Nielsen evaluadas

No.	Heurística	Pregunta de Evaluación
1	Visibilidad del estado del sistema	La aplicación me informa claramente sobre lo que está sucediendo (por ejemplo, cuando está grabando, procesando o esperando).
2	Correspondencia entre el sistema y el mundo real	El lenguaje y los iconos utilizados en la aplicación son fáciles de entender y familiares para mí.
3	Control y libertad del usuario	Puedo deshacer o cancelar acciones fácilmente si cometo un error.
4	Consistencia y estándares	Los elementos de la interfaz (botones, menús, mensajes) son consistentes en toda la aplicación.
5	Prevención de errores	La aplicación me ayuda a evitar errores y me advierte antes de realizar acciones importantes.
6	Reconocer antes que recordar	La información y las opciones importantes están siempre visibles o fácilmente accesibles, sin necesidad de recordar pasos previos.
7	Flexibilidad y eficiencia de uso	La aplicación permite realizar tareas de manera eficiente, tanto para usuarios nuevos como experimentados.
8	Diseño estético y minimalista	La interfaz es clara, simple y no contiene información innecesaria.
9	Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de errores	Los mensajes de error son claros y me ayudan a entender cómo solucionar el problema.
10	Ayuda y documentación	Si lo necesito, puedo encontrar fácilmente ayuda o instrucciones sobre cómo usar la aplicación.

9.4. Resultados de las pruebas**9.4.1. Resultados de pruebas técnicas****Cobertura de código**

Las pruebas automatizadas alcanzaron los siguientes niveles de cobertura:

- **Backend:** 85 % de cobertura general
- **Frontend:** 78 % de cobertura de componentes
- **Servicios:** 92 % de cobertura de servicios Angular

Resultados por módulo

Tabla 9.6: Resultados de pruebas por módulo

Módulo	Pruebas Ejecutadas	Pruebas Exitosas	Observaciones
API Health Check	1	1	Endpoint de estado funcional
Inferencia de Audio	2	2	Manejo correcto con y sin archivos
Acceso a Archivos	2	2	Verificación de existencia de archivos
Procesamiento Audio	1	1	Generación sintética de WAV funcional
Componentes Angular	3	3	DashboardComponent completamente funcional
Total Backend	5	5	100 % de pruebas exitosas

9.4.2. Resultados de usabilidad

Encuesta de uso de PronunciApp

Los resultados de la encuesta de uso mostraron una alta satisfacción general con la aplicación:

Tabla 9.7: Resultados de la encuesta de uso de PronunciApp

Pregunta	Sí (%)	No (%)	Parcial (%)
¿Fue fácil entender cómo usar la aplicación desde el principio?	97.0	3.0	0.0
¿La interfaz de usuario te resultó intuitiva y fácil de navegar?	96.9	3.1	0.0
¿Pudiste grabar y la salida de audio fue satisfactoria?	84.4	15.6	0.0
¿El tiempo de respuesta de la aplicación fue adecuado (Muy rápido + Aceptable)?	100.0	0.0	0.0
¿El resultado generado (audio/texto) cumplió con tus expectativas?	68.8	3.1	28.1
¿Consideras que la aplicación es útil para la corrección de la pronunciación?	90.6	3.1	6.3
¿Recomendarías esta aplicación a otras personas?	96.9	3.1	0.0

Encuesta de heurísticas de Nielsen

La evaluación basada en las heurísticas de Nielsen reveló fortalezas y áreas de mejora específicas. Se utilizó una escala de 1 a 5 (1=Muy malo, 5=Excelente) con una muestra de 7 evaluadores expertos:

Tabla 9.8: Resultados de la evaluación de heurísticas de Nielsen

No.	Heurística	Promedio	Observaciones
1	Visibilidad del estado del sistema	5.0	Excelente indicación del estado de grabación y procesamiento
2	Correspondencia con el mundo real	4.7	Iconos familiares y lenguaje muy claro y natural
3	Control y libertad del usuario	5.0	Opciones de cancelación y control completas
4	Consistencia y estándares	4.7	Interfaz altamente coherente en toda la aplicación
5	Prevención de errores	4.6	Buenas validaciones preventivas implementadas
6	Reconocer antes que recordar	5.0	Información siempre visible y fácilmente accesible
7	Flexibilidad y eficiencia	5.0	Excelente para usuarios nuevos y experimentados
8	Diseño estético y minimalista	5.0	Interfaz muy limpia, clara y sin elementos innecesarios
9	Ayuda con errores	4.6	Mensajes de error claros y útiles para resolución
10	Ayuda y documentación	4.7	Documentación integrada accesible y comprensible

9.5. Análisis de resultados

9.5.1. Análisis estadístico detallado

Análisis demográfico y distribución de respuestas

La muestra de 32 participantes para la encuesta de uso estuvo compuesta exclusivamente por estudiantes de Ingeniería de Sistemas de la Universidad del Valle, sede Tuluá, proporcionando una perspectiva homogénea del público objetivo en términos de formación académica y contexto institucional.

Distribución temporal de las respuestas

Tabla 9.9: Distribución temporal de respuestas de la encuesta

Fecha	Respuestas	Porcentaje	Observaciones
27 de mayo	2	6.3 %	Inicio de recolección de datos
28 de mayo	21	65.6 %	Mayor concentración de respuestas
29 de mayo	8	25.0 %	Continuación de evaluación
30 de mayo	1	3.1 %	Finalización de recolección

Continúa en la siguiente página

Total	32	100 %	Período: 4 días consecutivos
--------------	-----------	--------------	-------------------------------------

La concentración del 65.6 % de respuestas en un solo día sugiere una aplicación controlada de la encuesta, lo que garantiza condiciones similares de evaluación.

Análisis de variabilidad en tiempo de respuesta

Tabla 9.10: Distribución de evaluación de tiempo de respuesta

Categoría	Respuestas	Porcentaje	Interpretación
Muy rápido	21	65.6 %	Rendimiento óptimo percibido
Aceptable	11	34.4 %	Rendimiento satisfactorio
Lento	0	0 %	Sin problemas de rendimiento
Muy lento	0	0 %	Sin problemas críticos
Total satisfactorio	32	100 %	Rendimiento consistentemente alto

Los resultados de tiempo de respuesta demuestran un rendimiento excepcional del sistema, con 100 % de usuarios reportando tiempos satisfactorios. La distribución 65.6 % "muy rápido" vs 34.4 % "aceptable" confirma que la optimización de la arquitectura Azure con GPU especializada cumplió efectivamente con los requisitos de rendimiento establecidos.

Patrón de respuestas en expectativas

Tabla 9.11: Análisis de cumplimiento de expectativas

Nivel de cumplimiento	Respuestas	Porcentaje	Implicaciones
Sí, completamente	22	68.8 %	Alta satisfacción con resultados
Parcialmente	9	28.1 %	Oportunidad de mejora identificada
No cumplió	1	3.1 %	Caso marginal de insatisfacción
Satisfacción general	31	96.9 %	Nivel aceptable de cumplimiento

El análisis del cumplimiento de expectativas revela que el 68.8 % reporta satisfacción completa, mientras que el 28.1 % tiene expectativas parcialmente cumplidas. Esta distribución indica que la funcionalidad básica es sólida, pero aspectos específicos de calidad de procesamiento requieren refinamiento. El 3.1 % de insatisfacción es estadísticamente marginal.

Consistencia en evaluaciones de expertos Nielsen

Tabla 9.12: Métricas de consistencia en evaluación de expertos

Métrica	Valor	Rango	Interpretación
Desviación estándar promedio	0.19	Muy bajo	Baja variabilidad entre evaluadores
Rango de puntuaciones	4.6 - 5.0	Excelente	Alta concentración en valores superiores
Coeficiente de variación	4.0 %	Óptimo	Excelente concordancia entre evaluadores
Promedio general	4.83/5.0	Excepcional	Calidad superior del diseño de interfaz

9.5.2. Análisis cuantitativo de la encuesta de uso

Tabla 9.13: Resultados detallados de satisfacción del usuario

Aspecto evaluado	Porcentaje positivo	Nivel	Indicación
Facilidad de uso	97 %	Excelente	Curva de aprendizaje óptima
Intuitividad de interfaz	96.9 %	Excelente	Diseño UX efectivo
Recomendación	96.9 %	Excelente	Alta satisfacción general
Utilidad percibida	90.6 %	Muy bueno	Valor pedagógico reconocido
Funcionalidad grabación	84.4 %	Bueno	Área con oportunidades de mejora
Cumplimiento expectativas	68.8 %	Satisfactorio	Principal área de optimización
Tiempo de respuesta	100 %	Excepcional	Aspecto mejor evaluado

Los resultados cuantitativos revelan una jerarquía clara de fortalezas y oportunidades. El tiempo de respuesta (100 %) es la fortaleza absoluta, seguido por facilidad de uso (97 %) e intuitividad (96.9 %). El descenso gradual hasta cumplimiento de expectativas (68.8 %) indica que la calidad de resultados representa el principal desafío técnico, mientras que la funcionalidad de grabación (84.4 %) requiere optimización intermedia.

9.5.3. Análisis de las heurísticas de Nielsen

Fortalezas destacadas

Tabla 9.14: Heurísticas con puntuación perfecta (5.0/5.0)

Heurística	Puntuación	Justificación
Visibilidad del estado del sistema	5.0	Excelente retroalimentación visual durante grabación y procesamiento

Continúa en la siguiente página

Control y libertad del usuario	5.0	Opciones completas de cancelación y navegación intuitiva
Reconocer antes que recordar	5.0	Información crítica siempre visible sin dependencia de memoria
Flexibilidad y eficiencia	5.0	Apropiada tanto para usuarios novatos como experimentados
Diseño estético y minimalista	5.0	Interfaz limpia sin elementos superfluos o distractores

Las cinco heurísticas con puntuación perfecta representan los aspectos más sólidos del diseño de interfaz. La puntuación máxima en "Visibilidad del estado del sistema" confirma indicadores claros de grabación y procesamiento. El control y libertad del usuario perfecto indica navegación sin restricciones. La información esencial siempre visible reduce la carga cognitiva, mientras que la flexibilidad e interfaz minimalista garantizan usabilidad para todos los usuarios.

Áreas con oportunidades de mejora

Tabla 9.15: Heurísticas con potencial de optimización

Heurística	Puntuación	Oportunidades identificadas
Correspondencia con el mundo real	4.7	Algunos iconos podrían ser más universales y familiares
Consistencia y estándares	4.7	Muy buena coherencia con mínimas inconsistencias detectadas
Ayuda y documentación	4.7	Documentación accesible con oportunidad de expansión
Prevención de errores	4.6	Buenas validaciones con espacio para mejoras preventivas
Ayuda con errores	4.6	Mensajes claros pero podrían ser más específicos

9.5.4. Correlación entre encuestas

Tabla 9.16: Correlación entre evaluación de expertos y usuarios

Aspecto	Nielsen	Usuarios	Correlación
Flexibilidad y eficiencia	5.0/5.0	97 % facilidad	Muy alta
Diseño estético y minimalista	5.0/5.0	96.9 % intuitividad	Muy alta
Prevención de errores	4.6/5.0	68.8 % expectativas	Moderada
Ayuda con errores	4.6/5.0	15.6 % problemas grabación	Coherente

9.5.5. Fortalezas identificadas

Tabla 9.17: Principales fortalezas del sistema

Categoría	Métrica	Evidencia
Usabilidad técnica	4.83/5.0	Promedio excepcional en heurísticas de Nielsen
Satisfacción del usuario	96.9 %	Porcentaje de recomendación y facilidad de uso
Rendimiento óptimo	100 %	Satisfacción completa con tiempos de respuesta
Interfaz intuitiva	96.9 %	Navegación fácil e intuitiva percibida
Diseño centrado en usuario	5.0/5.0	Puntuaciones perfectas en visibilidad y control

9.5.6. Áreas de mejora prioritarias

Basadas en datos de usuario

Tabla 9.18: Áreas de mejora identificadas por usuarios

Área	Métrica actual	Prioridad	Acción recomendada
Optimización de resultados	68.8 % cumplimiento completo	Alta	Mejorar algoritmos de procesamiento
Funcionalidad de grabación	15.6 % reportó problemas	Media	Optimizar compatibilidad de dispositivos
Gestión de expectativas	28.1 % cumplimiento parcial	Media	Clarificar capacidades del sistema

El análisis de áreas de mejora basado en retroalimentación de usuarios proporciona una hoja de ruta clara para futuras iteraciones. La optimización de resultados emerge como prioridad alta debido a que el 31.2 % de usuarios no alcanzó satisfacción completa. La funcionalidad de grabación afecta al 15.6 % de usuarios, representando un problema técnico específico. La gestión de expectativas sugiere la necesidad de comunicar más claramente las capacidades actuales del sistema.

Basadas en evaluación de expertos

Tabla 9.19: Áreas de mejora identificadas por expertos

Área	Puntuación actual	Prioridad	Acción recomendada
Expansión de documentación	4.7/5.0	Baja	Ampliar contenido de ayuda integrada
Refinamiento de mensajes	4.6/5.0	Media	Hacer mensajes más específicos y accionables

Continúa en la siguiente página

Validaciones preventivas	4.6/5.0	Media	Implementar más controles proactivos
--------------------------	---------	-------	--------------------------------------

9.5.7. Impacto de los resultados

Tabla 9.20: Impacto e implicaciones de los resultados obtenidos

Indicador	Valor	Implicación
Adopción potencial	96.9 % recomendación	Alta viabilidad comercial y académica
Facilidad de implementación	97 % facilidad de uso	Reducidas necesidades de capacitación
Estabilidad de diseño	4.83/5.0 Nielsen	Arquitectura de UI sólida y madura
Área de optimización	68.8 % expectativas	Enfoque claro en algoritmos de procesamiento

9.5.8. Análisis específico de problemas detectados

Funcionalidad de grabación

Tabla 9.21: Análisis detallado de problemas de grabación

Aspecto	Valor	Análisis
Usuarios afectados	5 de 32 (15.6 %)	Problema minoritario pero significativo
Posibles causas	Múltiples factores	Permisos de micrófono, compatibilidad, configuración
Impacto en satisfacción	Correlación negativa	Grupo también reportó expectativas parciales
Patrón de dispositivos	Concentración específica	Problemas en dispositivos o navegadores particulares

El análisis detallado de los problemas de grabación revela un patrón específico que afecta a una minoría significativa de usuarios (15.6 %). Los múltiples factores causales (permisos de micrófono, compatibilidad de navegadores, configuración de dispositivos) sugieren interacción compleja entre la aplicación web y diferentes entornos de usuario. La correlación negativa con satisfacción general indica que estos problemas impactan directamente la percepción de calidad.

Expectativas parcialmente cumplidas

Tabla 9.22: Análisis de usuarios con expectativas parciales

Característica	Valor	Interpretación
Porcentaje de usuarios	28.1 %	Grupo significativo con reservas específicas

Continúa en la siguiente página

Patrón de respuestas	Satisfacción general positiva	Usuarios contentos pero con expectativas específicas
Área principal de mejora	Calidad de procesamiento	Enfoque en precisión de algoritmos de audio
Mantenimiento de recomendación	96.9 % sigue recomendando	Alta tolerancia y reconocimiento del potencial

El análisis de usuarios con expectativas parciales (28.1 %) revela un perfil específico que reconoce el valor de la aplicación pero identifica limitaciones en calidad. El mantenimiento de un 96.9 % de recomendación indica alta tolerancia y reconocimiento del potencial tecnológico. La concentración en calidad de procesamiento proporciona dirección clara para optimizaciones futuras en refinamientos algorítmicos del modelo VALL-E-X.

9.5.9. Validación estadística de resultados

Tamaño de muestra y representatividad

Tabla 9.23: Validación estadística de la muestra

Parámetro	Valor	Interpretación
Tamaño muestra usuarios	32 participantes	Suficiente para estudio piloto académico
Tamaño muestra expertos	7 evaluadores	Adecuado para evaluación heurística Nielsen
Margen de error	$\pm 17.3\%$ (95 % confianza)	Apropiado para contexto de investigación
Homogeneidad institucional	Universidad del Valle	Garantiza consistencia en contexto educativo
Saturación de datos	Patrones consistentes	Indica suficiencia de muestra para conclusiones

La validación estadística confirma la robustez metodológica del estudio para un contexto de investigación académica. El tamaño de muestra de 32 usuarios resulta apropiado para una evaluación piloto, mientras que los 7 evaluadores expertos cumplen con las recomendaciones estándar para evaluaciones Nielsen. El margen de error de $\pm 17.3\%$ es aceptable para investigación exploratoria, y la homogeneidad institucional reduce variables confusoras.

Análisis de sesgos potenciales

Tabla 9.24: Identificación y mitigación de sesgos

Tipo de sesgo	Riesgo	Mitigación aplicada
Sesgo de selección	Moderado	Muestra homogénea pero representativa del objetivo

Continúa en la siguiente página

Sesgo temporal	Bajo	Concentración en un día con condiciones controladas
Sesgo de familiaridad técnica	Moderado	Estudiantes de sistemas con mayor tolerancia técnica
Robustez de resultados	Alta	Consistencia entre diferentes días y grupos

El análisis de sesgos potenciales demuestra una metodología consciente de las limitaciones pero con mitigaciones efectivas. El sesgo de selección moderado se mitiga por ser precisamente el público objetivo para aplicaciones educativas. El sesgo temporal bajo garantiza condiciones controladas de evaluación. El sesgo de familiaridad técnica proporciona una evaluación conservadora, ya que este grupo tiende a ser más crítico. La alta robustez de resultados valida que las conclusiones no son artefactos de condiciones específicas.

Capítulo 10

Conclusiones y trabajos futuros

10.1. Conclusiones

- El prototipo desarrollado demostró efectividad parcial en la mejora de la pronunciación del inglés en estudiantes de Ingeniería de Sistemas de la Universidad del Valle, sede Tuluá. Las pruebas con estudiantes revelaron alta facilidad de uso y la mayoría consideró la aplicación útil para corrección de pronunciación. Sin embargo, una porción significativa reportó expectativas parcialmente cumplidas, indicando mejora parcial pero no total. La naturaleza prototípica requiere estudios longitudinales de mayor duración para validar completamente la efectividad pedagógica a largo plazo.
- La comparación directa de resultados con aplicaciones comerciales como Duolingo y ELSA Speak presenta limitaciones metodológicas significativas debido a diferencias en contextos de evaluación, métricas utilizadas y poblaciones objetivo. Estas aplicaciones emplean sistemas de evaluación propietarios con criterios no estandarizados, mientras que el prototipo PronunciApp fue evaluado en un contexto académico controlado con métricas específicas de usabilidad y satisfacción. Sin embargo, es posible establecer comparaciones cualitativas en aspectos como tecnología implementada, donde VALL-E-X ofrece análisis más detallado que los sistemas básicos de reconocimiento de voz comerciales. Para futuras investigaciones, se requiere el desarrollo de métricas estandarizadas y metodologías comparativas que permitan evaluaciones más rigurosas entre sistemas CAPT.
- Las tecnologías de PLN y síntesis neuronal han revolucionado los sistemas CAPT. La convergencia de síntesis de voz personalizada con análisis automático de pronunciación ofrece oportunidades pedagógicas sin precedentes. Las técnicas más prometedoras incluyen síntesis personalizada (VALL-E-X), análisis prosódico avanzado (Bark, StyleTTS2), y arquitecturas híbridas (Whisper + síntesis neuronal). Los principales desafíos incluyen demanda computacional alta, necesidad de validación pedagógica y integración efectiva de múltiples tecnologías.
- El tablero Kanban en GitHub Projects demostró alta efectividad durante 40 sprints, procesando 69 tareas distribuidas por prioridad (P1: 47, P2: 13, P3: 9). Los factores críticos de éxito incluyeron granularidad adecuada de tareas, criterios de aceptación claros, revisión regular de prioridades y limitación del trabajo en progreso. Los desafíos principales fueron la estimación de tareas de investigación y gestión de dependencias complejas.
- La integración exitosa de Figma y V0 by Vercel logró balancear innovación tecnológica con UX/UI centrado en el usuario. Las decisiones clave incluyeron priorizar simplicidad sobre funcionalidad compleja, reducir significativamente la carga informativa, enfocar en la experiencia práctica del

usuario e integrar fluidamente tecnologías avanzadas. El resultado representa una aplicación que prioriza usabilidad y efectividad pedagógica por encima de complejidad técnica.

- La arquitectura FastAPI + Angular 19 desplegada en Azure NCasT4_v3 proporcionó una solución técnicamente sólida y funcionalmente completa. La integración exitosa del modelo VALL-E-X cumplió el objetivo de análisis automatizado de pronunciación, mientras que los frameworks modernos y la arquitectura de microservicios proporcionaron una base robusta para futuras mejoras. El despliegue confirmó la viabilidad técnica y económica de la solución propuesta.
- El prototipo demostró funcionalidad completa y estabilidad técnica en las pruebas realizadas. La arquitectura desplegada en Azure NCasT4_v3 proporcionó recursos computacionales adecuados para el procesamiento eficiente del modelo de IA. Las pruebas confirmaron la viabilidad técnica y económica de la solución propuesta, validando la integración exitosa entre el frontend Angular 19 y el backend FastAPI con el modelo VALL-E-X para análisis de pronunciación en tiempo real.

10.2. Trabajos futuros

De acuerdo al alcance de este trabajo de grado se definieron una serie de objetivos a cumplir para el desarrollo de una aplicación web de mejora de pronunciación del inglés utilizando tecnologías de IA. Sin embargo, hay ciertas funcionalidades y elementos que podrían desarrollarse en un futuro para mejorar el desarrollo e implementación del sistema, a continuación se definen algunos de estos elementos:

10.2.1. Mejoras tecnológicas

- **Optimización de modelos de IA:** Investigar técnicas de compresión y cuantización para reducir la demanda computacional de VALL-E-X sin comprometer la precisión del análisis de pronunciación.
- **Integración de modelos multimodales:** Incorporar análisis visual de movimientos labiales y faciales para complementar el análisis de audio, utilizando tecnologías como MediaPipe o OpenCV.
- **Síntesis de voz personalizada:** Implementar capacidades de clonación de voz del instructor para crear modelos de referencia personalizados y mejorar la experiencia de aprendizaje.
- **Escalabilidad en la nube:** Desarrollar arquitectura de microservicios distribuidos con contenedores Docker y orquestación Kubernetes para mejorar la escalabilidad horizontal.

10.2.2. Mejoras pedagógicas

- **Adaptación curricular:** Integrar el sistema con currículos estándar de enseñanza del inglés (Common European Framework, TOEFL, IELTS) para proporcionar rutas de aprendizaje estructuradas.
- **Análisis de progresión:** Implementar algoritmos de aprendizaje automático para analizar patrones de progreso individual y adaptar ejercicios según las necesidades específicas del estudiante.
- **Gamificación avanzada:** Desarrollar sistemas de logros, rankings y desafíos sociales para aumentar la motivación y el compromiso de los estudiantes.
- **Feedback multimodal:** Incorporar retroalimentación visual, táctil y auditiva para diferentes estilos de aprendizaje.

10.2.3. Validación y evaluación

- **Estudios longitudinales:** Realizar estudios a largo plazo con grupos de control para validar la efectividad pedagógica del sistema comparado con métodos tradicionales de enseñanza.
- **Validación cross-cultural:** Evaluar el sistema con estudiantes de diferentes trasfondos lingüísticos y culturales para asegurar su efectividad universal.
- **Métricas de usabilidad:** Implementar métricas avanzadas de UX/UI y análisis de comportamiento de usuario para optimización continua de la experiencia.
- **Certificación académica:** Trabajar hacia la certificación del sistema por instituciones educativas reconocidas para su uso en contextos académicos formales.

Capítulo 11

Bibliografía

- [1] M. D. M. Ayón, M. A. F. Bayas, G. S. Coello, and V. J. C. Vásquez, “Importancia del idioma inglés para el docente,” *Journal of Science and Research: Revista Ciencia e Investigación*, vol. 7, no. 1, pp. 100–120, 2022.
- [2] J. D. P. Caro, “La inteligencia artificial en la sociedad: explorando su impacto actual y los desafíos futuros.” https://oa.upm.es/75068/1/TFG_JAROD_DAVID_PEDRAZA_CARO.pdf, 2023. Consultado 17 Mayo de 2024.
- [3] Y. Macías, “La tecnología y la inteligencia artificial en el sistema educativo.” Repositori Universitat Jaume I, 2020.
- [4] M. G. Asratie, B. D. Wale, and Y. T. Aylet, “Effects of using educational technology tools to enhance efl students’ speaking performance,” *Education and Information Technologies*, vol. 28, no. 8, pp. 10031–10051, 2023.
- [5] B. E. Schueler, C. A. Asher, K. E. Larned, S. Mehrotra, and C. Pollard, “Improving low-performing schools: A meta-analysis of impact evaluation studies,” *American Educational Research Journal*, vol. 59, no. 5, pp. 975–1010, 2022.
- [6] C. Tejedor-Garcia, D. Escudero-Mancebo, E. Camara-Arenas, C. Gonzalez-Ferreras, and V. Cardenoso-Payo, “Assessing pronunciation improvement in students of english using a controlled computer-assisted pronunciation tool,” *IEEE Transactions on Learning Technologies*, vol. 13, no. 2, 2020.
- [7] “Duolingo - wikipedia, la enciclopedia libre.” <https://es.wikipedia.org/wiki/Duolingo>. Consultado 17 Mayo de 2024.
- [8] “Home – speechace — pronunciation and fluency assessment via speech recognition.” <https://www.speechace.com/>. Consultado 17 Mayo de 2024.
- [9] “The world’s best way to improve your english pronunciation — elsaspeak.” <https://elsaspeak.com/en/>. Consultado 17 Mayo de 2024.
- [10] T. Young, D. Hazarika, S. Poria, and E. Cambria, “Recent trends in deep learning based natural language processing,” *IEEE Computational Intelligence Magazine*, vol. 13, no. 3, pp. 55–75, 2020.
- [11] Y. Wang, L. Tay, B. S. P. Ang, and H. Luo, “Learning from ai: Deep learning models for continuous language learning,” *International Journal of Artificial Intelligence in Education*, vol. 30, pp. 1–21, 2020.

- [12] British Council, “English skills for the technology sector in colombia.” Informe sectorial, 2023. Estudio sobre demanda de inglés en empresas tecnológicas colombianas.
- [13] K. J. Krahnke and S. D. Krashen, “Principles and practice in second language acquisition,” *TESOL Quarterly*, vol. 17, no. 2, 1983.
- [14] M. G. O’Brien, “Pronunciation fundamentals: Evidence-based perspectives for l2 teaching and research by t.m. derwing & m.j. munro,” *The Canadian Modern Language Review*, vol. 73, no. 3, 2017.
- [15] N. Golestani, N. Molko, S. Dehaene, D. LeBihan, and C. Pallier, “Brain structure predicts the learning of foreign speech sounds,” *Cerebral Cortex*, vol. 17, no. 3, 2007.
- [16] J. Rubin, “What the ‘good language learner’ can teach us,” *TESOL Quarterly*, vol. 9, no. 1, 1975.
- [17] M. H. Long, “Does second language instruction make a difference? a review of research,” *TESOL Quarterly*, vol. 17, no. 3, 1983.
- [18] J. M. Levis, “Changing contexts and shifting paradigms in pronunciation teaching,” *TESOL Quarterly*, vol. 39, no. 3, 2005.
- [19] M. Warschauer and C. Meskill, “Technology and second language teaching,” in *Handbook of Undergraduate Second Language Education*, Routledge, 2013.
- [20] C. A. Chapelle, “Historical foundations of casla,” in *Computer Applications in Second Language Acquisition*, Cambridge University Press, 2012.
- [21] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [22] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach (4th Edition)*. Pearson, 2021.
- [23] X. Chen, H. Xie, and G.-J. Hwang, “A multi-perspective study on artificial intelligence in education: Grants, conferences, journals, software tools, institutions, and researchers,” *Computers and Education: Artificial Intelligence*, vol. 1, p. 100005, 2020.
- [24] O. Zawacki-Richter, V. I. Marín, M. Bond, and F. Gouverneur, “Systematic review of research on artificial intelligence applications in higher education – where are the educators?,” *International Journal of Educational Technology in Higher Education*, vol. 16, no. 1, p. 39, 2019.
- [25] W. Holmes, M. Bialik, and C. Fadel, *Artificial Intelligence in Education: Promises and Implications for Teaching and Learning*. Center for Curriculum Redesign, 2019.
- [26] X. Lu, X. Huang, and J. Tian, “A review on speech recognition technology for language learning,” *Journal of Educational Technology Development and Exchange*, vol. 14, no. 1, pp. 49–68, 2021.
- [27] R. Godwin-Jones, “Emerging technologies: The evolving roles of language teachers: Trained coders, local researchers, reflective practitioners,” *Language Learning & Technology*, vol. 23, no. 3, pp. 6–19, 2019.
- [28] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O’Reilly Media, 2020.
- [29] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2021.

- [30] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language models are few-shot learners,” tech. rep., OpenAI, 2020.
- [31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.
- [32] “Anexo 9 - niveles de madurez tecnológica y de manufactura - trl y mrl.” documento proporcionado, páginas 3, 6, 8.
- [33] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 12449–12460, 2020.
- [34] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust speech recognition via large-scale weak supervision,” tech. rep., OpenAI, 2022. arXiv:2212.04356.
- [35] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, “Hubert: Self-supervised speech representation learning by masked prediction of hidden units,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.
- [36] Z. Wang, C. Wang, S. Chen, Y. Wu, S. Liu, J. Li, Y. Qian, Y. Qian, J. Wu, and F. Wei, “Speak foreign languages with your own voice: Cross-lingual neural codec language modeling,” *arXiv preprint arXiv:2303.03926*, 2023.
- [37] Coqui Team, “Coqui tts: A deep learning toolkit for text-to-speech.” <https://github.com/coqui-ai/TTS>, 2021. Open source TTS library.
- [38] J. Betker, “Bark: Text-to-audio model.” <https://github.com/suno-ai/bark>, 2023. Suno AI.
- [39] Y. A. Li, C. Han, V. S. Raghavan, G. Mischler, and N. Mesgarani, “Styletts 2: Towards human-level text-to-speech through style diffusion and adversarial training with large speech language models,” *arXiv preprint arXiv:2306.07691*, 2023.