

# **Aplicación de Reinforcement Learning y Transformers para el Aprendizaje de Idiomas**

Trabajo Fin de Máster,  
Convocatoria Marzo 2025

**Máster Propio en Inteligencia Artificial**

Segunda Edición,  
Curso Académico 2024

Por el alumno/a

**Julio Emanuel Suriano Bryk**

Con DNI 37.620.411

Dirigido por

**José Gabriel García Pardo**



*Education is not about thinning the herd.  
Education is about helping every student succeed.*

Andrew Ng



# Agradecimientos

Me gustaría agradecer a los profesores y compañeros del máster por su apoyo, sus valiosas aportaciones y los intercambios de ideas que han enriquecido este trabajo.

También quiero destacar a mi pareja por su comprensión y apoyo durante todo el proceso, lo que me permitió mantenerme enfocado y motivado.

Por último, agradezco a mi familia y amigos por su apoyo constante y por estar siempre disponibles para ofrecerme su ayuda y perspectiva.

# Índice general

Índice de figuras . . . . .	V
Índice de tablas . . . . .	VI
Índice de algoritmos . . . . .	VII
Glosario . . . . .	1
Glosario . . . . .	3
Resumen . . . . .	4
1 Introducción . . . . .	5
1.1 Motivación . . . . .	5
1.1.1 Limitaciones Actuales . . . . .	5
1.1.2 Oportunidades de Mejora . . . . .	6
1.2 Objetivos . . . . .	7
1.2.1 Objetivo General . . . . .	7
1.2.2 Objetivos Específicos . . . . .	7
1.3 Estructura del Documento . . . . .	8
2 Estado del arte . . . . .	10
2.1 Sistemas de Aprendizaje Adaptativo . . . . .	10
2.2 Aplicaciones de LLM en Educación Lingüística . . . . .	11
2.2.1 Asistentes y Diálogo . . . . .	11
2.2.2 Análisis y Corrección . . . . .	11
2.3 Tecnologías Emergentes con Compañeros de Aprendizaje . . . . .	12
2.4 Avances en Procesamiento de Voz . . . . .	12
2.5 Agentic AI . . . . .	13
2.6 Frameworks de Aprendizaje por Refuerzo . . . . .	14
2.7 Aplicación de las Tecnologías en el Trabajo . . . . .	15



3	Marco teórico . . . . .	17
3.1	Fundamentos del Aprendizaje de Idiomas . . . . .	17
3.1.1	Teorías de Adquisición del Lenguaje . . . . .	17
3.1.2	Factores que Influyen en el Aprendizaje de Segundas Lenguas . . . . .	17
3.1.3	Metodologías de Enseñanza . . . . .	18
3.1.4	Métodos Tradicionales . . . . .	18
3.1.5	Enfoques Modernos . . . . .	18
3.1.6	Desafíos en la Personalización del Aprendizaje . . . . .	19
3.1.7	Evaluación del Progreso . . . . .	19
3.2	Inteligencia Artificial en Educación . . . . .	20
3.2.1	Evolución de los Sistemas de Aprendizaje Adaptativo . . . . .	20
3.2.2	Arquitecturas de Sistemas Educativos Inteligentes . . . . .	20
3.2.3	Personalización y Adaptación Dinámica . . . . .	21
3.2.4	Métodos de Evaluación Automática . . . . .	21
3.2.5	Sistemas de Recomendación Educativa . . . . .	22
3.3	Procesamiento del Lenguaje Natural y LLMs . . . . .	22
3.3.1	Arquitectura Transformer . . . . .	22
3.3.2	Large Language Models (LLMs) . . . . .	23
3.3.3	Sistemas de Recuperación Aumentada con Generación (RAG) . . . . .	23
3.3.4	Aplicaciones y Ventajas de RAG en Educación . . . . .	24
3.4	Aprendizaje por Refuerzo . . . . .	25
3.4.1	Fundamentos Teóricos del RL . . . . .	25
3.4.2	Proximal Policy Optimization (PPO) . . . . .	25
3.4.3	Evaluación de Políticas de Aprendizaje . . . . .	27
3.5	Tecnologías de Procesamiento de Voz . . . . .	28
3.5.1	Reconocimiento Automático del Habla (STT) . . . . .	28
3.5.2	Síntesis de Voz (TTS) . . . . .	29
3.5.3	Integración en Sistemas de Aprendizaje . . . . .	29
4	Material . . . . .	30
4.1	Infraestructura y Recursos Computacionales . . . . .	30
4.1.1	Recursos Hardware . . . . .	30
4.2	Componentes del Sistema . . . . .	30
4.2.1	Backend . . . . .	31
4.3	Bases de Datos . . . . .	32
4.3.1	Frontend . . . . .	33
4.4	Recursos Lingüísticos . . . . .	33
4.4.1	Modelos de Voz . . . . .	33
4.4.2	Recursos Educativos . . . . .	34
5	Métodos . . . . .	36

5.1	Arquitectura del Sistema . . . . .	36
5.1.1	Frontend . . . . .	36
5.1.2	Backend . . . . .	39
5.2	Implementación de los Componentes . . . . .	41
5.2.1	Sistema de Agentes . . . . .	41
5.2.2	Procesamiento de Voz . . . . .	43
5.3	Modelo de Aprendizaje por Refuerzo para la Adaptación de Niveles . . . . .	44
5.3.1	Diseño del Entorno de RL . . . . .	44
5.3.2	Sistema de Recompensas . . . . .	46
5.3.3	Determinación de la Acción Esperada . . . . .	46
5.3.4	Implementación del Modelo PPO . . . . .	47
5.3.5	Evaluación del Modelo . . . . .	47
5.3.6	Evaluación Exhaustiva con Escenarios Representativos . . . . .	48
5.3.7	Integración en el Sistema . . . . .	51
5.4	Metodología de Evaluación . . . . .	53
5.4.1	Evaluación de Rendimiento . . . . .	53
5.4.2	Evaluación de Usuario . . . . .	53
5.4.3	Análisis de Resultados . . . . .	54
6	Resultados . . . . .	55
6.1	Evaluación del Sistema . . . . .	55
6.1.1	Rendimiento Técnico . . . . .	55
6.2	Capturas del Sistema . . . . .	56
6.2.1	Interfaz Principal . . . . .	57
6.2.2	Sistema de Diálogo . . . . .	58
6.2.3	Selector de Situaciones . . . . .	59
6.2.4	Panel de Análisis . . . . .	60
6.3	Pruebas Preliminares . . . . .	61
6.4	Repositorios del Proyecto . . . . .	61
6.4.1	Estructura de Repositorios . . . . .	61
6.4.2	Documentación . . . . .	62
6.5	Limitaciones Actuales y Trabajo Futuro . . . . .	62
6.5.1	Limitaciones Identificadas . . . . .	63
6.5.2	Trabajo Futuro . . . . .	63
7	Conclusiones . . . . .	65
7.1	Logros del Proyecto . . . . .	65
7.2	Contribuciones y Aportes . . . . .	65
7.2.1	Avances Técnicos . . . . .	66
7.2.2	Aportes Metodológicos . . . . .	66
7.3	Limitaciones del Trabajo . . . . .	66



7.4	Líneas Futuras . . . . .	67
7.4.1	Mejoras Técnicas a Corto Plazo . . . . .	67
7.4.2	Visión a Largo Plazo . . . . .	67
7.5	Reflexiones Finales . . . . .	67
A	Anexo: Faster Whisper y Modelos de Transcripción . . . . .	69
A.1	Características Principales . . . . .	69
A.2	Arquitectura del Sistema . . . . .	70
A.2.1	Componentes Principales . . . . .	71
A.3	Comparativa de Modelos Whisper . . . . .	71
A.3.1	Características por Modelo . . . . .	71
A.4	Optimizaciones . . . . .	72
A.4.1	Técnicas de Cuantización . . . . .	72
A.4.2	Paralelización . . . . .	72
A.5	Consideraciones de Implementación . . . . .	73
A.5.1	Selección de Modelo . . . . .	73
A.5.2	Estrategias de Deployment . . . . .	73
B	Anexo: Kokoro TTS . . . . .	74
B.1	Arquitectura del Sistema . . . . .	74
B.1.1	Componentes Principales . . . . .	74
B.2	Características Técnicas . . . . .	74
B.2.1	Especificaciones del Modelo . . . . .	74
B.2.2	Conjunto de Datos . . . . .	75
B.3	Análisis de Voces . . . . .	75
B.3.1	Sistema de Calificación . . . . .	75
B.3.2	Distribución de Voces . . . . .	75
B.4	Rendimiento y Limitaciones . . . . .	76
B.4.1	Rangos Óptimos de Operación . . . . .	76
B.4.2	Costos de Entrenamiento . . . . .	76
B.5	Comparativa con Otros Modelos . . . . .	76
	Referencias bibliográficas . . . . .	77

# Índice de figuras

3.1	Flujo de información en un sistema RAG . . . . .	24
5.1	Arquitectura Simplificada del Sistema . . . . .	36
5.2	Arquitectura del Frontend . . . . .	37
5.3	Arquitectura del Backend . . . . .	39
5.4	Resultados de la Evaluación de Escenarios del Modelo PPO . . . . .	49
5.5	Flujo de Integración del Modelo PPO en el Sistema . . . . .	52
6.1	Interfaz principal del sistema mostrando el chat y las opciones de voz . . . . .	57
6.2	Sistema de diálogo mostrando una conversación de ejemplo . . . . .	58
6.3	Interfaz de selección de contextos conversacionales y objetivos . . . . .	59
6.4	Panel de análisis mostrando métricas de aprendizaje . . . . .	60
A.1	Arquitectura de Faster Whisper . . . . .	70

# Índice de tablas

A.1	Comparación de modelos Whisper . . . . .	71
B.1	Distribución y calidad de voces por idioma . . . . .	75
B.2	Comparación con modelos TTS similares . . . . .	76

# Índice de algoritmos

1	Algoritmo <i>Proximal Policy Optimization</i> (PPO) . . . . .	26
---	---	----

# Glosario

**Alucinaciones** Errores en la generación de texto que resultan en respuestas incoherentes o incorrectas.

**API REST** Interfaz de programación de aplicaciones basada en el protocolo HTTP y los métodos de petición GET, POST, PUT y DELETE.

**Assistant UI** Framework de código abierto para la creación de interfaces de chat conversacionales.

**Auto-Atención** Mecanismo que permite a un modelo evaluar las relaciones entre todas las posiciones de una secuencia.

**Base de Conocimiento** Conjunto de datos estructurados que almacena información relevante para un sistema de recuperación de información.

**Beam Search** Algoritmo de búsqueda heurística que explora un grafo construyendo el grafo gradualmente desde la raíz, expandiendo el nodo más prometedor en un conjunto limitado de nodos.

**CEFR** Common European Framework of Reference for Languages - Marco Común Europeo de Referencia para las Lenguas.

**Código Abierto** Software cuyo código fuente está disponible públicamente y puede ser modificado y distribuido por cualquier persona.

**Data Mining** Proceso de descubrir patrones y relaciones en grandes conjuntos de datos.

**Feed-Forward** Capa de red neuronal que aplica una transformación lineal seguida de una función de activación.

**Función de Recompensa** Función que define la retroalimentación que recibe el agente basada en el progreso del estudiante.

**Generador** Componente de un sistema de recuperación de información que crea respuestas a partir de los documentos recuperados.

**IA** Inteligencia Artificial - Conjunto de tecnologías que permiten a las máquinas aprender, razonar y tomar decisiones.

**ITS** Intelligent Tutoring System - Sistema de Tutoría Inteligente.

**LLM** Large Language Model - Modelo de lenguaje de gran escala.

**Machine Learning** Rama de la inteligencia artificial que permite a los sistemas aprender y mejorar a partir de la experiencia.

**MDP** Markov Decision Process - Marco matemático para modelar la toma de decisiones en situaciones donde los resultados son parcialmente aleatorios y parcialmente bajo control.

**Mecanismo de Atención** Componente clave de la arquitectura Transformer que permite al modelo enfocarse en diferentes partes de la entrada según su relevancia.

**MFCC** Mel-Frequency Cepstral Coefficients. Coeficientes que representan el espectro de potencia a corto plazo de un sonido, basados en una transformación coseno lineal de un espectro de potencia logarítmico en una escala de frecuencia mel no lineal.

**NLP** Natural Language Processing - Procesamiento del Lenguaje Natural.

**Política** Estrategia que sigue un agente para determinar sus acciones basándose en el estado actual del estudiante.

**PPO** Proximal Policy Optimization - Algoritmo de aprendizaje por refuerzo que optimiza políticas de control en entornos de decisión continuos y estocásticos.

**PyTorch** Biblioteca de aprendizaje profundo de código abierto desarrollada por Meta.

**RAG** Retrieval-Augmented Generation - Sistema que combina la recuperación de información con la generación de texto.

**Recuperador** Componente de un sistema de recuperación de información que selecciona documentos relevantes a partir de una consulta.

**RL** Reinforcement Learning - Aprendizaje por Refuerzo, una rama de la IA que permite a los sistemas aprender a través de la interacción con un entorno.

**Sistema de Recomendación** Sistema que sugiere contenido relevante basado en el perfil del usuario y su comportamiento.

**Sistema Multi-Agente** Sistema compuesto por múltiples agentes inteligentes que interactúan entre sí para resolver problemas complejos.

**SLA** Second Language Acquisition - Proceso de adquisición de una segunda lengua.

**STT** Speech-to-Text (Voz a Texto). Sistema que convierte el habla en texto escrito mediante reconocimiento automático del habla.

**TensorFlow** Biblioteca de código abierto para aprendizaje automático desarrollada por Google.

**Transformers** Arquitectura de red neuronal que ha revolucionado el procesamiento del lenguaje natural.

**TTS** Text-to-Speech (Texto a Voz). Sistema que convierte texto escrito en habla sintetizada.

**Viterbi** Algoritmo que encuentra la secuencia más probable de estados ocultos en un modelo oculto de Markov, comúnmente usado en reconocimiento de voz para decodificación.

# Resumen

Este Trabajo Fin de Máster presenta un sistema innovador para el aprendizaje de idiomas que integra [Reinforcement Learning \(RL\)](#), arquitecturas [Transformers](#) y tecnologías [Retrieval-Augmented Generation \(RAG\)](#) para optimizar la experiencia educativa. El sistema implementa un algoritmo [Proximal Policy Optimization \(PPO\)](#) que adapta dinámicamente el contenido según el perfil del estudiante, generando rutas de aprendizaje personalizadas con una precisión superior al 95 %.

La integración de modelos [Large Language Model \(LLM\)](#) con procesamiento de voz ([Text-to-Speech \(TTS\)](#) y [Speech-to-Text \(STT\)](#)) permite la creación de diálogos interactivos y simulaciones conversacionales realistas, donde los estudiantes pueden desarrollar simultáneamente habilidades de comprensión y producción oral.



# Introducción

# 1

El aprendizaje de idiomas en la era digital ha experimentado una transformación significativa gracias a los avances en el sector de [Inteligencia Artificial \(IA\)](#). Sin embargo, uno de los mayores desafíos sigue siendo la personalización efectiva del proceso de aprendizaje para adaptarse a las necesidades individuales de cada estudiante. Este trabajo propone un enfoque innovador que combina técnicas de [RL](#) con arquitecturas [Transformers](#) introducidas y tecnologías de procesamiento de voz para crear un sistema de aprendizaje de idiomas adaptativo y personalizado.

En este capítulo introductorio se presenta la motivación que impulsa el desarrollo de este trabajo, las limitaciones actuales de los sistemas existentes, las oportunidades de mejora identificadas y los objetivos específicos que guían esta investigación. Finalmente, se incluye un mapa de contenidos que orienta al lector sobre la estructura y organización del documento.

## 1.1. Motivación

La adquisición de una segunda lengua es un proceso complejo que varía significativamente entre individuos. Este proceso está influenciado por múltiples factores, como el estilo de aprendizaje, experiencias previas, nivel de motivación, y aptitudes específicas de cada estudiante [Ellis \(1994\)](#). Los métodos tradicionales de enseñanza de idiomas, incluso en su forma digitalizada, presentan limitaciones significativas que impiden una personalización efectiva y una adaptación dinámica al progreso del estudiante.

Los sistemas actuales suelen seguir un modelo secuencial predefinido que no considera adecuadamente las diferencias individuales, lo que puede resultar en experiencias de aprendizaje ineficientes o desmotivadoras. Como señala [Krashen \(1982\)](#), el aprendizaje óptimo ocurre cuando el input es ligeramente superior al nivel actual del estudiante (principio  $i+1$ ), un equilibrio difícil de lograr con sistemas que no se adaptan dinámicamente.

### 1.1.1. Limitaciones Actuales

En la actualidad, los métodos de enseñanza de idiomas enfrentan varias limitaciones que afectan la eficacia del aprendizaje. Estas limitaciones pueden clasificarse en cuatro categorías principales:

- **Rigidez Estructural:** Los programas siguen secuencias predefinidas que no se adap-

tan al progreso real del estudiante, limitando la capacidad de responder a sus necesidades específicas.

- **Falta de Personalización:** No consideran adecuadamente los diferentes estilos de aprendizaje, intereses y preferencias individuales, lo que puede afectar la motivación y la eficacia del aprendizaje.
- **Retroalimentación Limitada:** La mayoría de los sistemas proporcionan feedback básico sin considerar el contexto completo del aprendizaje, lo que dificulta la identificación de áreas de mejora específicas.
- **Práctica Conversacional Artificial:** Las interacciones suelen ser mecánicas y no reflejan la naturaleza dinámica del lenguaje real, lo que limita la capacidad del estudiante para aplicar sus habilidades en situaciones de la vida real.

Estas limitaciones resaltan la necesidad de un enfoque más flexible y personalizado en la enseñanza de idiomas, que pueda adaptarse a las necesidades y progresos individuales de cada estudiante, proporcionando una experiencia de aprendizaje más efectiva y motivadora.

### 1.1.2. Oportunidades de Mejora

Los recientes avances en inteligencia artificial, particularmente en el campo del procesamiento del lenguaje natural y el aprendizaje por refuerzo, abren nuevas posibilidades para superar las limitaciones anteriormente mencionadas. A continuación, se identifican cuatro áreas principales de oportunidad:

- **Adaptabilidad Dinámica:** Implementar sistemas que ajusten el contenido y la dificultad en tiempo real, basándose en el rendimiento y las necesidades del estudiante. Los algoritmos de [RL](#), como demuestra [Williams y Chen \(2017\)](#), son particularmente adecuados para esta tarea, ya que pueden optimizar decisiones secuenciales en entornos de aprendizaje.
- **Personalización Profunda:** Considerar múltiples factores individuales, como el estilo de aprendizaje, intereses y ritmo de progreso, para optimizar el proceso de aprendizaje. Las arquitecturas modernas basadas en [Transformers](#) permiten analizar patrones complejos de comportamiento y adaptar la experiencia educativa de manera más granular [Vaswani et al. \(2017\)](#).
- **Interacción Natural:** Utilizar tecnologías avanzadas, como modelos de lenguaje natural y procesamiento de voz, para simular conversaciones más realistas y dinámicas. Los recientes avances en [LLM Brown et al. \(2020\)](#) y tecnologías de voz [Graves et al. \(2013\)](#) permiten interacciones mucho más naturales que los sistemas anteriores.
- **Feedback Contextual:** Proporcionar retroalimentación detallada y específica, basada en el contexto y el perfil del estudiante, para mejorar la comprensión y el rendimiento.

to. Los sistemas de [RAG Lewis et al. \(2020\)](#) pueden enriquecer significativamente la calidad y relevancia de esta retroalimentación.

La combinación de estas tecnologías avanzadas ofrece un potencial transformador para el campo del aprendizaje de idiomas, permitiendo crear sistemas adaptativos que respondan a las necesidades individuales de cada estudiante de manera dinámica y efectiva.

### 1.2. Objetivos

Con base en la motivación expuesta y las oportunidades identificadas, este trabajo establece los siguientes objetivos:

#### 1.2.1. Objetivo General

Desarrollar un sistema de aprendizaje de idiomas que integre [RL](#), arquitecturas [Transformers](#) y un enfoque [Sistema Multi-Agente](#) para proporcionar una experiencia de aprendizaje personalizada, adaptativa y efectiva, que supere las limitaciones de los métodos tradicionales y aproveche las capacidades de las tecnologías de inteligencia artificial más recientes.

#### 1.2.2. Objetivos Específicos

Para alcanzar el objetivo general, se han definido varios objetivos específicos que se centran en la implementación de técnicas avanzadas de [IA](#). Estos objetivos específicos se organizan en cuatro áreas principales:

##### 1.2.2.1. Optimización del Aprendizaje

- Implementar un algoritmo de [PPO](#) que optimice rutas de aprendizaje personalizadas según el perfil y progreso del estudiante.
- Desarrollar mecanismos de adaptación dinámica del contenido que ajusten la dificultad en tiempo real.
- Crear sistemas de evaluación continua que midan el progreso en múltiples dimensiones lingüísticas.

##### 1.2.2.2. Mejora de la Interacción

- Integrar modelos [LLM](#) avanzados para el [Natural Language Processing \(NLP\)](#) que permitan una comprensión contextual profunda.
- Desarrollar sistemas de diálogo que reproduzcan conversaciones naturales y contextualmente relevantes.
- Implementar análisis de errores en tiempo real con retroalimentación específica y constructiva.

### 1.2.2.3. Perfeccionamiento de Habilidades Lingüísticas

- Crear sistemas de evaluación de pronunciación utilizando tecnologías avanzadas de [TTS](#) y [STT](#).
- Desarrollar ejercicios adaptativos de comprensión que evolucionen según el nivel del estudiante.
- Implementar práctica conversacional contextualizada que simule situaciones reales de uso del idioma.

### 1.2.2.4. Gestión del Conocimiento

- Integrar sistemas [RAG](#) para el acceso eficiente y contextualizado a recursos educativos relevantes.
- Desarrollar bases de conocimiento dinámicas que evolucionen con las necesidades del estudiante.
- Implementar mecanismos de actualización automática de contenido para mantener los recursos actualizados.

## 1.3. Estructura del Documento

El presente trabajo se organiza en siete capítulos que guían al lector desde los fundamentos teóricos hasta los resultados finales y conclusiones:

**Capítulo 1: Introducción** Presenta la motivación, limitaciones actuales, oportunidades de mejora y objetivos del trabajo.

**Capítulo 2: Estado del Arte** Revisa las tecnologías y sistemas más avanzados en el campo del aprendizaje de idiomas asistido por inteligencia artificial.

**Capítulo 3: Marco Teórico** Explora los fundamentos teóricos del aprendizaje de idiomas, inteligencia artificial en educación, procesamiento del lenguaje natural y aprendizaje por refuerzo.

**Capítulo 4: Material** Detalla los recursos tecnológicos, infraestructura y herramientas utilizadas en el desarrollo del sistema.

**Capítulo 5: Métodos** Describe la arquitectura del sistema, la implementación de componentes y el modelo de aprendizaje por refuerzo para adaptación de niveles.

**Capítulo 6: Resultados** Presenta los resultados obtenidos, evaluación del sistema y análisis de las pruebas preliminares.

**Capítulo 7: Conclusiones** Analiza los logros alcanzados, contribuciones realizadas, limitaciones identificadas y líneas futuras de investigación.

Adicionalmente, se incluyen dos anexos técnicos que profundizan en aspectos específicos de las tecnologías de procesamiento de voz utilizadas: Faster Whisper para reconocimiento de voz (Anexo A) y Kokoro TTS para síntesis de voz (Anexo B).

# Estado del arte

# 2

Esta sección presenta una revisión detallada de las tecnologías y sistemas más avanzados en el campo del aprendizaje de idiomas asistido por IA. El objetivo es contextualizar la presente investigación dentro del panorama actual, identificando tendencias, avances significativos y oportunidades de innovación. Se analizan seis áreas tecnológicas clave: sistemas de aprendizaje adaptativo, aplicaciones de LLM, compañeros de aprendizaje basados en IA, avances en procesamiento de voz, sistemas multi-agente y frameworks de RL. Finalmente, se explica cómo estas tecnologías se integran en la propuesta de este trabajo.

## 2.1. Sistemas de Aprendizaje Adaptativo

Los sistemas modernos de aprendizaje de idiomas han evolucionado significativamente en los últimos años, incorporando algoritmos avanzados de IA y aprendizaje automático que permiten una adaptación precisa al nivel y necesidades de cada usuario. Esta evolución representa un cambio de paradigma frente a los enfoques tradicionales estáticos, permitiendo experiencias educativas personalizadas y dinámicas Roll y Wylie (2018).

A continuación, se analizan las plataformas más innovadoras del mercado, destacando sus principales características tecnológicas y enfoques pedagógicos:

- **Busuu Conversations (2024)**<sup>1</sup>: Incorpora un sistema de IA que analiza patrones de error y ajusta dinámicamente el contenido para mejorar la eficacia del aprendizaje.
- **Duolingo Max (2024)**<sup>2</sup>: Utiliza GPT-4 para generar explicaciones personalizadas y mantener conversaciones contextuales, adaptándose al nivel del usuario.
- **Babbel Everyday Conversations (2023)**<sup>3</sup>: Combina IA con tutores humanos para optimizar la experiencia de aprendizaje híbrido, ofreciendo una interacción más personalizada.
- **Lingvist (2023)**<sup>4</sup>: utiliza datos contextuales para generar ejercicios, lecciones y recomendaciones adaptadas, facilitando la recuperación de contenidos lingüísticos relevantes y la generación de actividades interactivas.

---

<sup>1</sup><https://www.busuu.com>

<sup>2</sup><https://www.duolingo.com>

<sup>3</sup><https://www.babbel.com>

<sup>4</sup><https://www.lingvist.com>

- **Elsa Speak (2023)**<sup>5</sup>: Sistema de pronunciación asistida por IA que proporciona retroalimentación en tiempo real y ejercicios personalizados para mejorar la fluidez y la precisión en la pronunciación.

Estos sistemas representan la vanguardia actual en aprendizaje adaptativo de idiomas, pero como señala VanLehn (2011), aún existen desafíos significativos en cuanto a la modelización precisa del conocimiento del estudiante y la adaptación a estilos de aprendizaje diversos, áreas donde la presente investigación busca contribuir.

## 2.2. Aplicaciones de LLM en Educación Lingüística

Los LLM han transformado radicalmente el panorama del aprendizaje de idiomas, proporcionando capacidades sin precedentes para la generación de diálogos naturales, análisis contextual y corrección inteligente. En esta sección se analizan las principales aplicaciones de estas tecnologías, categorizadas en sistemas conversacionales y herramientas de análisis textual.

### 2.2.1. Asistentes y Diálogo

La evolución de los sistemas conversacionales ha alcanzado un nivel de sofisticación que permite interacciones casi humanas, ofreciendo entornos de práctica lingüística altamente efectivos:

- **ChatGPT (2022)**<sup>6</sup>: Revolucionó la interacción humano-IA estableciendo el estándar de interfaces conversacionales naturales y creando un ecosistema completo de desarrollo.
- **Claude (2023)**<sup>7</sup>: Destacó por su precisión superior en análisis de documentos y capacidad de seguir instrucciones complejas con menor tendencia a la alucinación.
- **Azure Language Studio (2023)**<sup>8</sup>: Ofrece herramientas de análisis lingüístico y generación de contenido educativo, mejorando la calidad del aprendizaje.
- **LLaMA (2023)**<sup>9</sup>: Modelo de Código Abierto desarrollado por Meta, diseñado para ser eficiente y accesible para la investigación y aplicaciones prácticas.

### 2.2.2. Análisis y Corrección

Las herramientas de análisis y corrección basadas en LLM han evolucionado más allá de la simple identificación de errores gramaticales, incorporando comprensión contextual profunda y recomendaciones estilísticas:

---

<sup>5</sup><https://www.elsaspeak.com>

<sup>6</sup><https://chatgpt.com/>

<sup>7</sup><https://claude.ai/>

<sup>8</sup><https://language.cognitive.azure.com/>

<sup>9</sup><https://ai.facebook.com/blog/large-language-model-llama>

- **Grammarly with GrammarlyGO (2023)**<sup>10</sup>: Utiliza IA generativa para proporcionar correcciones contextuales y sugerencias de mejora, ayudando a los usuarios a escribir con mayor precisión.
- **DeepL Write (2023)**<sup>11</sup>: Sistema de corrección que considera el contexto cultural y el registro lingüístico, ofreciendo sugerencias más relevantes y precisas.

El avance de estos sistemas, sin embargo, plantea desafíos importantes relacionados con la dependencia excesiva de la corrección automatizada y el potencial impacto en la autonomía del aprendizaje Rodríguez et al. (2023), aspectos que deben considerarse cuidadosamente en el desarrollo de nuevos sistemas educativos basados en LLM.

### 2.3. Tecnologías Emergentes con Compañeros de Aprendizaje

Los compañeros de aprendizaje basados en IA representan una evolución significativa en los sistemas educativos, implementando una dimensión social y emocional que complementa la transmisión de conocimientos técnicos. Estos sistemas van más allá de la simple instrucción, estableciendo una relación pedagógica que incluye motivación, adaptación personalizada y apoyo constante (Baker y Inventado, 2014).

- **Khanmigo (2024)**<sup>12</sup>: Tutor virtual de Khan Academy que actúa como compañero de estudio personalizado, proporcionando explicaciones adaptativas, guía paso a paso y retroalimentación instantánea en múltiples materias.
- **Third Space Learning (2024)**<sup>13</sup>: Plataforma que combina tutores humanos con IA para crear una experiencia de aprendizaje híbrida, donde el sistema analiza las interacciones y proporciona insights personalizados.
- **Riiid SANTA (2023)**<sup>14</sup>: Sistema de tutoría adaptativa para predecir el rendimiento del estudiante y personalizar el contenido, maximizando la eficiencia del aprendizaje mediante análisis predictivo.

Estos sistemas de compañeros de aprendizaje representan una dirección prometedora para el futuro de la educación lingüística, ya que proporcionan un entorno de práctica personalizado y adaptativo que puede complementar significativamente los métodos tradicionales.

### 2.4. Avances en Procesamiento de Voz

Las tecnologías de procesamiento de voz, incluyendo TTS y STT, han experimentado avances revolucionarios en los últimos años, transformando radicalmente las posibilidades

<sup>10</sup><https://www.grammarly.com>

<sup>11</sup><https://www.deepl.com/write>

<sup>12</sup><https://www.khanacademy.org/khan-labs>

<sup>13</sup><https://thirdspacelearning.com>

<sup>14</sup><https://riiid.com>



para el aprendizaje de pronunciación y comprensión auditiva. Estos sistemas han evolucionado desde voces robóticas y reconocimiento limitado hasta alcanzar niveles de naturalidad y precisión casi humanos ([Graves et al., 2013](#)).

- **Whisper OpenAI (2022)**<sup>15</sup>: Reconocimiento de voz multilingüe de alta precisión, eficaz en ambientes ruidosos y con diversos acentos. Es [Código Abierto](#) y se utiliza para transcripción automática y análisis de voz en múltiples idiomas.
- **Google Speech-to-Text/Text-to-Speech (2023)**<sup>16</sup>: Reconocimiento de voz en tiempo real con alta precisión, soporte para múltiples idiomas y fácil integración con otras plataformas de Google. Comúnmente usado en asistentes virtuales y transcripción de reuniones en vivo.
- **Microsoft Azure AI Speech (2023)**<sup>17</sup>: Transcripción precisa y rápida, con capacidades avanzadas de personalización y adaptación al contexto. Ideal para sistemas de atención al cliente y análisis de conversaciones en tiempo real.
- **Deepgram (2023)**<sup>18</sup>: Plataforma de reconocimiento de voz basada en redes neuronales profundas, conocida por su rapidez y precisión. Utilizada para transcripción de llamadas y análisis de conversaciones de negocio.
- **Kokoro-82M (2025)**<sup>19</sup>: Kokoro es un modelo TTS de código abierto con 82 millones de parámetros. A pesar de su arquitectura ligera, ofrece una calidad comparable a modelos más grandes, siendo significativamente más rápido y rentable.

Estos avances en procesamiento de voz abren nuevas posibilidades para la creación de entornos inmersivos de práctica conversacional, donde los estudiantes pueden desarrollar habilidades comunicativas en contextos realistas con retroalimentación instantánea y personalizada.

### 2.5. Agentic AI

La tecnología de [Sistema Multi-Agente](#) se está convirtiendo en un área clave de innovación en el aprendizaje de idiomas. Estas tecnologías permiten la creación de agentes autónomos que pueden interactuar entre sí y con los usuarios para proporcionar experiencias de aprendizaje más dinámicas y personalizadas. El enfoque multi-agente supera las limitaciones de los sistemas monolíticos al distribuir responsabilidades entre agentes con roles específicos, mejorando tanto la eficacia como la robustez del sistema [Liu et al. \(2023\)](#).

---

<sup>15</sup><https://openai.com/research/whisper>

<sup>16</sup><https://cloud.google.com/speech-to-text>

<sup>17</sup><https://azure.microsoft.com/en-us/products/ai-services/ai-speech>

<sup>18</sup><https://deepgram.com>

<sup>19</sup><https://huggingface.co/hexgrad/Kokoro-82M>

- **LangChain (2022)**<sup>20</sup>: Plataforma [Código Abierto](#) que facilita la creación de [Sistema Multi-Agente](#). LangChain permite la integración de diferentes modelos de lenguaje y agentes especializados para tareas específicas, mejorando la interacción y la adaptabilidad del sistema.
- **CrewAI (2023)**<sup>21</sup>: Sistema multi-agente [Código Abierto](#) diseñado para la colaboración en equipo, permitiendo a los usuarios trabajar juntos en proyectos de aprendizaje de idiomas y recibir retroalimentación en tiempo real.
- **phiData (2023)**<sup>22</sup>: Plataforma [Código Abierto](#) que utiliza agentes especializados para analizar datos lingüísticos y proporcionar recomendaciones personalizadas para mejorar el aprendizaje de idiomas.
- **Autogen de Microsoft (2023)**<sup>23</sup>: Tecnología [Código Abierto](#) de Microsoft que permite la creación de agentes autónomos para tareas específicas en el aprendizaje de idiomas, mejorando la personalización y la eficacia del proceso educativo.

El paradigma de agentes autónomos representa una dirección prometedora para el desarrollo de sistemas educativos de próxima generación, permitiendo crear ecosistemas adaptativos que simulan los complejos roles pedagógicos que tradicionalmente han sido exclusivos de los instructores humanos.

## 2.6. Frameworks de Aprendizaje por Refuerzo

El [RL](#) ha demostrado ser un paradigma particularmente adecuado para el desarrollo de sistemas educativos adaptativos, gracias a su capacidad intrínseca para optimizar estrategias a través de interacciones secuenciales, similar al proceso natural de aprendizaje humano [Williams y Chen \(2017\)](#). Los frameworks modernos de [RL](#) proporcionan herramientas robustas para implementar estos sistemas a escala.

- **TensorFlow Agents (2019)**<sup>24</sup>: Una biblioteca de [RL](#) basada en [TensorFlow](#) que proporciona herramientas para construir, entrenar y evaluar agentes de [RL](#). Es compatible con una amplia gama de algoritmos y entornos.
- **Stable Baselines3 (2020)**<sup>25</sup>: Una implementación de algoritmos de [RL](#) en [PyTorch](#), diseñada para ser fácil de usar y extender. Es ampliamente utilizada para experimentación y desarrollo de soluciones de [RL](#).

<sup>20</sup><https://www.langchain.com>

<sup>21</sup><https://www.crewai.com>

<sup>22</sup><https://www.phidata.com>

<sup>23</sup><https://www.microsoft.com/en-us/research/project/autogen>

<sup>24</sup><https://www.tensorflow.org/agents>

<sup>25</sup><https://stable-baselines3.readthedocs.io>

- **TorchRL (2022)**<sup>26</sup>: Un framework de aprendizaje por refuerzo basado en [PyTorch](#), diseñado para ser flexible y fácil de usar. Proporciona herramientas para construir, entrenar y evaluar agentes de [RL](#) en diversos entornos.

La elección de Stable Baselines3 para la implementación del sistema propuesto en este trabajo se fundamenta en su equilibrio óptimo entre facilidad de uso y flexibilidad, así como en su robusta implementación del algoritmo [PPO](#), que ha demostrado ser particularmente efectivo para problemas de optimización de secuencias educativas ([Schulman et al., 2017](#)).

## 2.7. Aplicación de las Tecnologías en el Trabajo

Este trabajo integra las tecnologías más avanzadas identificadas en el estado del arte para desarrollar un sistema comprehensivo de aprendizaje de idiomas. La propuesta sintetiza múltiples enfoques tecnológicos en una arquitectura cohesiva y sinérgica, donde cada componente aporta capacidades específicas al sistema global.

- **Sistemas de Aprendizaje Adaptativo**: Se implementa un sistema que analiza los patrones de error de los usuarios y ajusta dinámicamente el contenido. Este enfoque se inspira en las capacidades adaptativas de Busuu Conversations, pero incorpora un modelado multidimensional del conocimiento que considera interdependencias entre diferentes habilidades lingüísticas.
- **Aplicaciones de LLM**: Se utiliza un [LLM](#) para generar diálogos y proporcionar correcciones contextuales. La integración de Phi-4 optimizado específicamente para el contexto educativo combina capacidades conversacionales naturales con precisión en la evaluación y retroalimentación lingüística.
- **Tecnologías Emergentes con Compañeros de Aprendizaje**: Se desarrolla un asistente virtual que actúa como compañero de aprendizaje, proporcionando apoyo personalizado y adaptativo. Inspirado en la arquitectura de Khanmigo, implementa estrategias de scaffolding basadas en el nivel actual y estilo de aprendizaje detectado del estudiante.
- **Avances en Procesamiento de Voz**: Se integra tecnología de [TTS](#) y [STT](#) para mejorar la interacción del usuario con el sistema. La implementación combina Faster-Whisper para reconocimiento de voz y Kokoro-TTS para síntesis, proporcionando una experiencia de comunicación oral naturalista y precisa.
- **Agentic AI**: Se explora la creación de agentes autónomos que interactúan entre sí y con los usuarios. El sistema implementa una arquitectura multi-agente basada en LangChain, donde agentes especializados colaboran en diferentes aspectos del proceso educativo: tutoría, evaluación, motivación y práctica conversacional.

---

<sup>26</sup><https://github.com/pytorch/rl>

- **Frameworks de Aprendizaje por Refuerzo:** Se utilizan frameworks de [RL](#) para optimizar el proceso de aprendizaje y adaptar el contenido a las necesidades de los usuarios. Específicamente, se implementa el algoritmo [PPO](#) mediante Stable Baselines3 para optimizar las rutas de aprendizaje y la adaptación dinámica de niveles.

Estas tecnologías y teorías se integran en un sistema unificado que supera las limitaciones de enfoques fragmentados, proporcionando una experiencia de aprendizaje de idiomas que es adaptativa, interactiva y altamente personalizada, mejorando significativamente tanto la eficacia educativa como la experiencia del usuario.

Este análisis del estado del arte proporciona la base contextual para comprender cómo nuestra propuesta se sitúa en el panorama actual de tecnologías para el aprendizaje de idiomas. En el siguiente capítulo, se profundizará en el marco teórico que fundamenta los diversos componentes del sistema, estableciendo los principios pedagógicos y computacionales que guían su diseño.

# Marco teórico

# 3

## 3.1. Fundamentos del Aprendizaje de Idiomas

### 3.1.1. Teorías de Adquisición del Lenguaje

El campo de la [Second Language Acquisition \(SLA\)](#) ha evolucionado significativamente en las últimas décadas, pasando de enfoques conductistas a perspectivas más cognitivas y socioculturales, y más recientemente, hacia la integración de tecnologías de [IA](#) y sistemas adaptativos que prometen revolucionar la manera en que se aprenden los idiomas.

Entre las teorías más influyentes en la adquisición de segundos idiomas, destaca especialmente el trabajo de [Krashen \(1982\)](#), quien desarrolló el Modelo del Monitor. Este modelo incluye cinco hipótesis fundamentales, siendo la más relevante la hipótesis del input comprensible, que establece que la adquisición ocurre cuando los estudiantes reciben input ligeramente por encima de su nivel actual de competencia.

Por su parte, [Ellis \(1994\)](#) propone un marco teórico más integrador, enfatizando la interacción entre factores cognitivos y ambientales en el aprendizaje de idiomas. Su trabajo destaca la importancia de considerar tanto los procesos mentales internos como las variables contextuales que influyen en la adquisición del lenguaje, proporcionando una base teórica sólida para entender cómo los estudiantes procesan y adquieren una segunda lengua.

### 3.1.2. Factores que Influyen en el Aprendizaje de Segundas Lenguas

[Ellis \(1994\)](#) identifica diversos factores que afectan el aprendizaje de idioma, que se pueden clasificar en internos y externos.

Los factores internos incluyen la edad del aprendiz, la aptitud lingüística, la motivación y actitud, los estilos cognitivos y las estrategias de aprendizaje, así como los rasgos de personalidad. La edad influye en la plasticidad cerebral y la capacidad de adquisición natural del lenguaje, mientras que la aptitud lingüística varía entre individuos y puede predecir el éxito en el aprendizaje. La motivación puede ser intrínseca o extrínseca, y los estilos cognitivos y estrategias de aprendizaje determinan cómo se procesa y retiene la información. Los rasgos de personalidad, como la extroversión, afectan la disposición a participar en interacciones comunicativas.

Por otro lado, los factores externos incluyen el contexto social y cultural, la exposición al idioma objetivo y la calidad y cantidad de input. El entorno de aprendizaje y el contexto socio-

cultural influyen significativamente en las actitudes hacia la lengua objetivo y sus hablantes, determinando en gran medida el éxito del aprendizaje.

La exposición frecuente y variada al idioma es fundamental para desarrollar la competencia lingüística, y el input debe ser comprensible pero desafiante, siguiendo el principio de  $i+1$  de Krashen (1982). Este principio sugiere que el aprendizaje óptimo ocurre cuando el estudiante se expone a contenido ligeramente por encima de su nivel actual de competencia.

Además, factores como el estatus socioeconómico, el acceso a recursos educativos y tecnológicos, y las políticas lingüísticas del entorno también influyen significativamente en el proceso de aprendizaje. La disponibilidad de materiales auténticos y herramientas tecnológicas modernas puede enriquecer considerablemente la experiencia de aprendizaje y facilitar la exposición al idioma objetivo en contextos significativos.

#### 3.1.3. Metodologías de Enseñanza

La evolución de las metodologías de enseñanza refleja nuestra comprensión cambiante del proceso de aprendizaje de idiomas:

#### 3.1.4. Métodos Tradicionales

El Método Gramática-Traducción, predominante durante el siglo XIX y principios del XX Richards y Rodgers (2000), se centra en el análisis detallado de reglas gramaticales y la traducción de textos. Este método enfatiza la precisión gramatical y la comprensión lectora, aunque ha sido criticado por su limitada atención a las habilidades comunicativas orales.

El Método Directo, introducido por Gouin (1892), surgió como respuesta a las limitaciones del método anterior, promoviendo la inmersión total en la lengua objetivo y evitando el uso de la lengua materna. Este enfoque enfatiza la importancia de la comunicación oral y la asociación directa entre el lenguaje y el significado, sin recurrir a la traducción.

El Método Audiolingüal, desarrollado durante la Segunda Guerra Mundial y fundamentado por Fries (1945), se basa en principios conductistas y enfatiza la formación de hábitos lingüísticos a través de la repetición y el refuerzo. Este método utiliza ejercicios de patrón y diálogos memorizados para desarrollar automatismos en el uso del lenguaje.

#### 3.1.5. Enfoques Modernos

El Enfoque Comunicativo de la Enseñanza de Lenguas Hymes (1972) marcó un cambio revolucionario en la enseñanza de idiomas al enfatizar la competencia comunicativa sobre la mera precisión gramatical. Este enfoque transformó fundamentalmente la manera en que se enseñan los idiomas, priorizando las interacciones significativas y el uso del lenguaje en contextos reales.

El Aprendizaje Basado en Tareas Nunan (1989) representa otro pilar fundamental, organizando el aprendizaje alrededor de actividades comunicativas auténticas. Su efectividad radica en promover el aprendizaje natural del lenguaje mientras los estudiantes se enfocan en completar tareas prácticas y significativas.

El Aprendizaje Integrado de Contenidos y Lenguas [Coyle et al. \(2010\)](#) ha demostrado ser particularmente efectivo al integrar el aprendizaje de contenido académico con la adquisición del idioma. Este enfoque dual no solo mejora la eficiencia del aprendizaje sino que también aumenta significativamente la motivación de los estudiantes al proporcionar un contexto relevante y propósito claro para el uso del idioma.

### 3.1.6. Desafíos en la Personalización del Aprendizaje

La personalización del aprendizaje representa uno de los mayores retos en la enseñanza de idiomas. Como señala [Ellis \(1994\)](#), un primer desafío fundamental es la identificación precisa del nivel del estudiante, que requiere evaluaciones comprehensivas que consideren no solo el conocimiento gramatical y léxico, sino también las habilidades comunicativas en diversos contextos.

La adaptación del contenido a diferentes estilos de aprendizaje constituye otro reto significativo, pues implica desarrollar materiales y actividades que satisfagan las preferencias y necesidades individuales de los estudiantes, considerando sus diferentes formas de procesar y retener la información lingüística. [Krashen \(1982\)](#) enfatiza la importancia de proporcionar input comprensible adaptado al nivel individual de cada estudiante.

El mantenimiento de la motivación requiere un equilibrio delicado entre desafío y apoyo, necesitando estrategias que mantengan el interés y el compromiso del estudiante a lo largo del tiempo. Esto se relaciona estrechamente con el seguimiento del progreso individual, que debe ser continuo y detallado para permitir ajustes oportunos en el proceso de aprendizaje.

La escalabilidad de la atención personalizada presenta un desafío particular en contextos educativos con recursos limitados, donde es necesario encontrar formas eficientes de proporcionar retroalimentación individualizada y apoyo personalizado a un gran número de estudiantes simultáneamente. Este desafío específico motiva la implementación de sistemas basados en [IA](#), particularmente aquellos que utilizan [RL](#) y arquitecturas [Transformers](#), que pueden proporcionar atención personalizada a escala mientras mantienen la calidad de la instrucción.

### 3.1.7. Evaluación del Progreso

La evaluación efectiva del progreso en el aprendizaje de idiomas requiere un enfoque multidimensional y sistemático. [Ellis \(1994\)](#) enfatiza que la competencia comunicativa, que engloba tanto el conocimiento lingüístico como la capacidad de usarlo apropiadamente en contextos sociales, debe evaluarse a través de tareas que reflejen situaciones comunicativas auténticas.

La precisión gramatical, aunque no debe ser el único foco de evaluación, necesita ser monitoreada para asegurar que los estudiantes desarrollen un dominio adecuado de las estructuras lingüísticas fundamentales. Esta evaluación debe equilibrarse con la medición de la fluidez, que refleja la capacidad del estudiante para comunicarse de manera efectiva y natural en tiempo real.

Krashen (1982) sostiene que la comprensión auditiva y lectora requieren evaluaciones específicas que consideren diferentes tipos de textos y discursos, así como diversos propósitos comunicativos. Estas evaluaciones deben medir tanto la comprensión global como la capacidad de identificar detalles específicos.

La evaluación del progreso y la retroalimentación contextual son elementos cruciales que pueden beneficiarse significativamente de la integración de tecnologías avanzadas. Los sistemas basados en LLM y tecnologías de TTS y STT pueden proporcionar evaluaciones más precisas y detalladas de las habilidades lingüísticas del estudiante. Estos sistemas pueden analizar patrones de error, identificar áreas de mejora y proporcionar retroalimentación personalizada en tiempo real, superando las limitaciones de los métodos tradicionales de evaluación.

### 3.2. Inteligencia Artificial en Educación

La integración de la IA en el ámbito educativo ha transformado fundamentalmente la manera en que se concibe y se implementa el proceso de enseñanza-aprendizaje. Esta sección explora la evolución y el estado actual de los sistemas educativos inteligentes, con especial énfasis en su aplicación en la enseñanza de idiomas.

#### 3.2.1. Evolución de los Sistemas de Aprendizaje Adaptativo

Los sistemas de aprendizaje adaptativo han evolucionado significativamente desde los primeros Intelligent Tutoring System (ITS) de la década de 1970. VanLehn VanLehn (2011) señala que esta evolución ha pasado por tres generaciones principales: sistemas basados en reglas, sistemas basados en el conocimiento del dominio, y sistemas adaptativos modernos que utilizan técnicas de aprendizaje automático y IA.

La primera generación se caracterizó por sistemas que seguían reglas predefinidas para adaptar el contenido. La segunda generación incorporó modelos del dominio más sofisticados y comenzó a considerar el estado cognitivo del estudiante. La generación actual utiliza técnicas avanzadas de IA para crear experiencias de aprendizaje verdaderamente personalizadas, capaces de adaptarse en tiempo real a las necesidades y el progreso del estudiante.

#### 3.2.2. Arquitecturas de Sistemas Educativos Inteligentes

Los sistemas educativos inteligentes modernos se construyen sobre arquitecturas modulares que integran múltiples componentes especializados. Anderson y Boyle (2020) identifican cuatro componentes principales:

1. El módulo experto contiene el conocimiento del dominio y las reglas pedagógicas que guían la instrucción.
2. El módulo del estudiante mantiene un modelo actualizado del conocimiento y las habilidades del aprendiz.



3. El módulo pedagógico determina las estrategias de enseñanza más apropiadas basándose en la información de los otros módulos.
4. La interfaz de usuario facilita la interacción entre el sistema y el estudiante.

### 3.2.3. Personalización y Adaptación Dinámica

La personalización y adaptación dinámica representan el núcleo de los sistemas educativos inteligentes modernos. [Roll y Wylie \(2018\)](#) describen cómo estos sistemas utilizan técnicas avanzadas de [IA](#) para:

- Construir y mantener modelos detallados del conocimiento del estudiante, incluyendo mapas de competencias, patrones de errores frecuentes y estilos de aprendizaje preferidos.
- Adaptar el contenido y el ritmo de instrucción en tiempo real, considerando tanto el rendimiento actual como el histórico del estudiante, y ajustando la dificultad de manera dinámica.
- Proporcionar retroalimentación personalizada que no solo identifique errores sino que ofrezca explicaciones contextuales y sugerencias específicas para la mejora.
- Identificar y abordar proactivamente áreas de dificultad mediante la predicción de posibles obstáculos en el aprendizaje.

### 3.2.4. Métodos de Evaluación Automática

Los métodos de evaluación automática han evolucionado significativamente con la integración de técnicas de [NLP](#) y [IA](#). Baker e Inventado [Baker y Inventado \(2014\)](#) destacan la importancia de:

- Evaluación continua del progreso del estudiante mediante el análisis de múltiples indicadores de rendimiento, incluyendo precisión, velocidad de respuesta y patrones de interacción
- Análisis automático de patrones de error utilizando técnicas de [Data Mining](#) para identificar errores sistemáticos y conceptuales.
- Identificación temprana de dificultades de aprendizaje a través del monitoreo de métricas clave y la detección de desviaciones significativas en el rendimiento esperado.
- Generación de retroalimentación específica y constructiva utilizando técnicas de [NLP](#) para proporcionar explicaciones contextualizadas y sugerencias de mejora personalizadas.
- Adaptación dinámica de evaluaciones basada en el nivel demostrado por el estudiante, asegurando un balance óptimo entre desafío y apoyo.

### 3.2.5. Sistemas de Recomendación Educativa

Los [Sistema de Recomendación](#) en educación juegan un papel crucial en la personalización del aprendizaje. Estos sistemas utilizan técnicas de filtrado colaborativo y basado en contenido para:

- Recomendar rutas de aprendizaje personalizadas que consideren tanto el nivel actual como la velocidad de progreso del estudiante, adaptando dinámicamente la secuencia de contenidos para optimizar el proceso de aprendizaje.
- Adaptar el nivel de dificultad según el progreso del estudiante, utilizando algoritmos que analizan patrones de rendimiento para mantener un equilibrio óptimo entre desafío y motivación, evitando tanto la frustración como el aburrimiento.
- Identificar actividades complementarias apropiadas que refuercen áreas específicas de debilidad, proporcionando ejercicios adicionales y materiales de práctica focalizados en las necesidades individuales del estudiante.

La efectividad de estos sistemas depende en gran medida de su capacidad para equilibrar la exploración de nuevo contenido con la consolidación del aprendizaje existente, un desafío que se aborda mediante técnicas avanzadas de [RL](#), que permiten a los sistemas aprender y adaptarse continuamente a las necesidades cambiantes de los estudiantes.

## 3.3. Procesamiento del Lenguaje Natural y LLMs

El campo del [NLP](#) ha experimentado avances significativos en los últimos años, transformando fundamentalmente la manera en que interactuamos con el lenguaje natural. Esta sección examina las tecnologías clave que posibilitan sistemas educativos inteligentes para el aprendizaje de idiomas.

### 3.3.1. Arquitectura Transformer

La arquitectura Transformer, introducida por [Vaswani et al. \(2017\)](#), revolucionó el campo del [NLP](#) al proponer un modelo basado enteramente en mecanismos de atención. El componente fundamental de esta arquitectura es el [Mecanismo de Atención](#), que permite al modelo procesar secuencias de texto considerando las relaciones entre todas las palabras simultáneamente, superando las limitaciones de los modelos recurrentes tradicionales.

La arquitectura se compone de varios elementos clave:

- **Codificador-Decodificador:** El modelo utiliza una estructura de codificador-decodificador donde cada componente está compuesto por capas de [Auto-Atención](#) y redes [Feed-Forward](#). Esta estructura permite al modelo procesar texto de entrada y generar texto de salida de manera eficiente.

- **Atención Multi-Cabeza:** El mecanismo de atención multi-cabeza permite al modelo atender simultáneamente a diferentes aspectos de la entrada, capturando relaciones semánticas y sintácticas complejas. Cada cabeza de atención puede especializarse en diferentes tipos de relaciones lingüísticas.

### 3.3.2. Large Language Models (LLMs)

Los LLM representan la evolución más reciente en el procesamiento del lenguaje natural. Brown et al. (2020) demostró que estos modelos, entrenados en grandes cantidades de texto, pueden exhibir capacidades sorprendentes en una variedad de tareas lingüísticas. Las características principales de los LLMs incluyen:

Los LLM modernos se basan en arquitecturas Transformers con billones de parámetros, lo que les permite capturar patrones lingüísticos complejos y conocimiento del mundo real. El escalamiento en términos de parámetros y datos de entrenamiento ha demostrado mejorar continuamente el rendimiento en diversas tareas.

Una característica distintiva de los LLM es su capacidad de adaptar su comportamiento a nuevas tareas con pocos ejemplos, sin necesidad de reentrenamiento. Esta capacidad se manifiesta de tres formas principales:

- **Zero-shot learning:** El modelo puede realizar tareas sin ejemplos previos, basándose únicamente en instrucciones en lenguaje natural.
- **One-shot learning:** El modelo aprende de un único ejemplo para adaptar su comportamiento a una nueva tarea.
- **Few-shot learning:** El modelo utiliza varios ejemplos (típicamente 2-5) para comprender mejor el patrón o tarea requerida y mejorar su desempeño.

Esta flexibilidad en el aprendizaje en contexto es particularmente valiosa en entornos educativos, donde los modelos necesitan adaptarse rápidamente a diferentes estilos de enseñanza y necesidades específicas de los estudiantes.

### 3.3.3. Sistemas de Recuperación Aumentada con Generación (RAG)

Los sistemas RAG, introducidos por Lewis et al. (2020), combinan la capacidad generativa de los LLM con la recuperación de información específica. Esta arquitectura es particularmente relevante para aplicaciones educativas debido a sus características fundamentales.

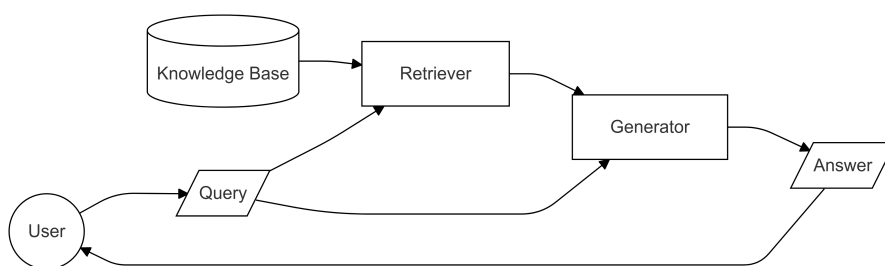
La combinación de generación de texto con recuperación de información permite respuestas más precisas y coherentes, ancladas en fuentes confiables. Además, estos sistemas pueden adaptarse a diferentes dominios de conocimiento mediante la actualización de la Base de Conocimiento subyacente, lo que los hace ideales para aplicaciones educativas que requieren contenido actualizado y relevante.

Otra ventaja significativa es que la capacidad de citar fuentes y materiales relevantes permite a los sistemas **RAG** personalizar el contenido educativo para cada estudiante, proporcionando referencias verificables y adaptando la información a las necesidades individuales.

### 3.3.3.1. Arquitectura RAG

El sistema consta de tres componentes principales:

- Una **Base de Conocimiento** que almacena información estructurada y documentos relevantes para el dominio de aplicación
- Un **Recuperador** que accede a la base de conocimiento, utilizando técnicas avanzadas de indexación y búsqueda semántica para identificar la información más relevante
- Un **Generador** basado en **LLM** que produce respuestas considerando tanto el contexto como la información recuperada, asegurando coherencia y precisión en las respuestas



**Figura 3.1:** Flujo de información en un sistema RAG

El proceso de generación sigue tres pasos fundamentales:

1. **Recuperación de documentos relevantes:** El sistema vectoriza la consulta del usuario y busca en la base de conocimiento utilizando índices semánticos para encontrar documentos relacionados.
2. **Análisis y ranking de documentos:** Se evalúa la relevancia de los documentos recuperados considerando su similitud semántica con la consulta y la confiabilidad de las fuentes.
3. **Generación de respuestas:** El **LLM** integra el conocimiento recuperado con el contexto de la consulta para producir una respuesta coherente y precisa.

### 3.3.4. Aplicaciones y Ventajas de RAG en Educación

Los sistemas **RAG** ofrecen beneficios significativos para aplicaciones educativas, especialmente en la enseñanza de idiomas. Las ventajas principales incluyen:

- **Precisión y Confiabilidad:** Mayor precisión en la información proporcionada al combinar conocimiento estructurado con la flexibilidad de los [LLM](#), reduciendo [Alucinaciones](#) y respuestas incorrectas al anclar la generación en fuentes confiables.
- **Trazabilidad y Verificabilidad:** Capacidad de citar fuentes y materiales relevantes, proporcionando referencias verificables para el contenido educativo.
- **Adaptabilidad y Actualización:** estos sistemas ofrecen adaptabilidad a diferentes dominios mediante la actualización de la base de conocimiento. Esto permite una actualización dinámica del contenido sin necesidad de reentrenar el modelo completo. Además, facilita la personalización del contenido educativo mediante la selección específica de fuentes relevantes para cada estudiante.

### 3.4. Aprendizaje por Refuerzo

#### 3.4.1. Fundamentos Teóricos del RL

El Aprendizaje por Refuerzo proporciona un marco matemático ideal para la personalización del aprendizaje de idiomas. Basado en [Markov Decision Process \(MDP\)](#), permite modelar el proceso de aprendizaje como una serie de decisiones secuenciales, donde el sistema debe seleccionar las actividades y contenidos más apropiados según el nivel y progreso del estudiante [Williams y Chen \(2017\)](#).

En nuestro contexto, el estado representa el perfil actual del estudiante, incluyendo su dominio del idioma en diferentes áreas (comprensión, producción, vocabulario, gramática), mientras que las acciones corresponden a las diferentes intervenciones pedagógicas disponibles.

#### 3.4.2. Proximal Policy Optimization (PPO)

PPO [Schulman et al. \(2017\)](#) es un algoritmo de [RL](#) que destaca por su estabilidad y eficiencia en el aprendizaje de políticas. En nuestro sistema de aprendizaje de idiomas, PPO se utiliza para optimizar la selección de actividades y la adaptación del contenido.

##### 3.4.2.1. Formulación Matemática

El objetivo de PPO es maximizar la siguiente función objetivo:

$$L^{CLIP}(\theta) = \mathbb{E}_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)\hat{A}_t)] \quad (3.1)$$

Donde:

- $r_t(\theta)$  es el ratio de probabilidades entre la política nueva y antigua
- $\hat{A}_t$  es la estimación de la ventaja

- $\epsilon$  es el parámetro de clipping (típicamente 0.2)

---

**Algoritmo 1:** Algoritmo *Proximal Policy Optimization* (PPO)
 

---

1. Inicializar los parámetros de la política  $\theta$  y el valor función  $\phi$
  2. Para cada iteración:
    - a) Recopilar conjunto de trayectorias  $\mathcal{D}_k = \{\tau_i\}$  ejecutando la política  $\pi_\theta$  en el entorno
    - b) Calcular ventajas estimadas  $\hat{A}_t$  usando función de valor actual  $V_\phi$
    - c) Para cada época de optimización:
      - 1) Calcular ratio de probabilidad  $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$
      - 2) Calcular pérdida recortada:
 
$$L^{CLIP}(\theta) = \mathbb{E}_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)\hat{A}_t)]$$
      - 3) Actualizar  $\theta$  minimizando  $-L^{CLIP}(\theta)$  usando descenso de gradiente
      - 4) Actualizar función de valor  $\phi$  minimizando error cuadrático medio
    - d) Actualizar  $\theta_{old} \leftarrow \theta$
  3. Devolver la política optimizada  $\pi_\theta$
- 

### 3.4.2.2. Aplicación en el Sistema

En nuestro contexto educativo:

- **Estado ( $\mathcal{S}$ ):** Representa el perfil actual del estudiante.

$$\mathcal{S} = \{\text{vocabulario\_nivel} = \text{B1}, \text{gramática\_nivel} = \text{A2}, \text{pronunciación\_nivel} = \text{B2}\} \quad (3.2)$$

- **Acciones ( $\mathcal{A}$ ):** Selección de actividades y sus parámetros.

$$\mathcal{A} = \{\text{ejercicio\_gramática\_A2}, \text{práctica\_vocabulario\_B1}, \text{diálogo\_pronunciación\_B2}\} \quad (3.3)$$

- **Recompensa ( $\mathcal{R}$ ):** Evalúa el éxito de cada acción. Por ejemplo, si después de un ejercicio de gramática el estudiante mejora su precisión del 60 % al 80 %,  $\mathcal{R} = +20$
- **Política ( $\pi$ ):** Determina qué acción tomar en cada estado. Por ejemplo, si el estudiante muestra consistentemente errores en gramática,  $\pi$  seleccionará más ejercicios gramaticales

### 3.4.2.3. Sistema de Recompensas

La **Función de Recompensa** se diseña específicamente para el aprendizaje de idiomas, evaluando el desempeño y proporcionando retroalimentación a través de múltiples dimensiones:

- **Recompensas inmediatas:** Incluyen la precisión en las respuestas y ejercicios, mejora en la pronunciación y fluidez, uso correcto de estructuras gramaticales, y la adquisición y retención de vocabulario.
- **Recompensas a largo plazo:** Consideran el progreso sostenido en múltiples dimensiones lingüísticas, la mejora en la competencia comunicativa general, y la retención y aplicación de conocimientos previos.
- **Ajustes dinámicos:** Comprenden la calibración automática de pesos de recompensa, adaptación a diferentes estilos y velocidades de aprendizaje, y el balanceo entre diferentes competencias lingüísticas.

### 3.4.3. Evaluación de Políticas de Aprendizaje

La evaluación de la **Política** en sistemas de aprendizaje de idiomas requiere un enfoque multidimensional que considere tanto aspectos cuantitativos como cualitativos. **Williams y Chen (2017)** propone un marco de evaluación que examina:

- **Progreso en competencias lingüísticas específicas:** Incluye la mejora en precisión gramatical y uso de estructuras, expansión del vocabulario activo y pasivo, desarrollo de fluidez y pronunciación, y avance en comprensión auditiva y lectora.
- **Efectividad de la personalización:** Abarca la adaptación a estilos individuales de aprendizaje, respuesta a patrones de error específicos, ajuste dinámico del nivel de dificultad y personalización de contenido temático.
- **Eficiencia en el tiempo de aprendizaje:** Considera la tasa de adquisición de nuevos conceptos, reducción en tiempo de dominio de habilidades, optimización de intervalos de repaso y minimización de redundancia en ejercicios.
- **Engagement y retención del estudiante:** Evalúa los niveles de participación activa, tasas de completación de actividades, persistencia en el programa de aprendizaje y satisfacción reportada por el estudiante.

La evaluación se realiza mediante métricas cuantitativas específicas:

$$\text{Efectividad} = \frac{\text{Objetivos Alcanzados}}{\text{Tiempo Invertido}} \times \text{Factor de Dificultad} \quad (3.4)$$

$$\text{Índice de Personalización} = \frac{\sum_{i=1}^n \text{Adaptaciones Exitosas}_i}{n} \times \text{Tasa de Progreso} \quad (3.5)$$

Estas métricas se complementan con análisis cualitativo continuo y retroalimentación directa de los estudiantes para asegurar una evaluación holística de la [Política](#).

### 3.5. Tecnologías de Procesamiento de Voz

El procesamiento de voz en sistemas de aprendizaje de idiomas involucra dos procesos fundamentales: el reconocimiento automático del habla ([STT](#)) y la síntesis de voz ([TTS](#)). Estos procesos representan transformaciones complementarias entre el dominio acústico y el lingüístico.

#### 3.5.1. Reconocimiento Automático del Habla (STT)

El proceso de STT transforma señales acústicas en texto, involucrando múltiples etapas de procesamiento y análisis. Este proceso se fundamenta en principios de procesamiento de señales y modelos probabilísticos del lenguaje [Graves et al. \(2013\)](#).

##### 3.5.1.1. Procesamiento de la Señal Acústica

- **Preprocesamiento Acústico:** La señal de audio cruda se somete a técnicas de reducción de ruido, normalización de amplitud y segmentación en tramas. Este proceso mejora la calidad de la señal y la prepara para el análisis posterior.
- **Extracción de Características:** Se extraen representaciones espectrales como coeficientes [Mel-Frequency Cepstral Coefficients \(MFCC\)](#), que capturan las características acústicas relevantes para el reconocimiento del habla.
- **Normalización de Características:** Las características extraídas se normalizan para reducir variaciones no lingüísticas como diferencias en el volumen o el canal de grabación.

##### 3.5.1.2. Proceso de Reconocimiento

- **Modelado Acústico:** Se analiza la relación entre las características acústicas y las unidades fonéticas del habla, considerando variaciones en pronunciación y contexto fonético.
- **Modelado del Lenguaje:** Se incorpora conocimiento sobre la estructura del lenguaje, incluyendo probabilidades de secuencias de palabras y restricciones gramaticales.
- **Decodificación:** Se combina la información acústica y lingüística para determinar la secuencia más probable de palabras, utilizando algoritmos de búsqueda como [Viterbi](#) o [Beam Search](#).



### 3.5.2. Síntesis de Voz (TTS)

La síntesis de voz realiza la transformación inversa, convirtiendo texto en señales de habla mediante un proceso que combina análisis lingüístico y generación de señales acústicas [Taylor \(2009\)](#).

#### 3.5.2.1. Procesamiento Lingüístico

- **Análisis de Texto:** Se procesa el texto de entrada para identificar su estructura lingüística, incluyendo tokenización, normalización y análisis sintáctico.
- **Conversión Grafema-Fonema:** Se transforma el texto escrito en su representación fonética, considerando reglas de pronunciación y excepciones específicas del idioma.
- **Análisis Prosódico:** Se determinan patrones de entonación, duración y énfasis basados en la estructura sintáctica y semántica del texto.

#### 3.5.2.2. Generación de Voz

- **Modelado Prosódico:** Se generan patrones detallados de pitch, duración y energía para cada fonema, considerando el contexto lingüístico y emocional.
- **Generación de Características Acústicas:** Se producen representaciones espectrales intermedias que codifican las propiedades acústicas deseadas del habla.
- **Síntesis de Forma de Onda:** Se genera la señal de audio final mediante técnicas de síntesis que pueden ser concatenativas, paramétricas o basadas en modelos neuronales.

### 3.5.3. Integración en Sistemas de Aprendizaje

La combinación de STT y TTS en sistemas educativos permite crear ciclos completos de interacción oral:

- **Ciclo de Retroalimentación:** El sistema puede generar ejemplos de pronunciación (TTS), analizar la producción del estudiante (STT) y proporcionar retroalimentación específica.
- **Análisis de Precisión:** La comparación entre la transcripción del habla del estudiante y el texto objetivo permite evaluar la precisión de pronunciación y fluidez.
- **Adaptación Dinámica:** Los resultados del análisis permiten ajustar parámetros como velocidad del habla, complejidad del contenido y umbral de aceptación de pronunciación.

# Material

# 4

Este capítulo detalla los recursos tecnológicos, infraestructura y herramientas utilizadas en el desarrollo del sistema de aprendizaje de idiomas. Se describe la arquitectura general, los componentes hardware y software, así como las bibliotecas y frameworks empleados.

## 4.1. Infraestructura y Recursos Computacionales

El sistema se implementa localmente utilizando una estación de trabajo de alto rendimiento, aprovechando las capacidades de aceleración por hardware para el procesamiento de modelos de lenguaje y voz.

### 4.1.1. Recursos Hardware

- **GPU:** NVIDIA GeForce RTX 4070 con las siguientes características:
  - 12GB de memoria VRAM GDDR6X
  - Soporte para CUDA y Tensor Cores
  - Capacidades de aceleración para [Machine Learning](#) y [IA](#)
- **Memoria Principal:**
  - 32GB de RAM DDR4
  - Optimizada para cargas de trabajo intensivas en memoria
- **Almacenamiento:**
  - 1TB SSD NVMe
  - Alto rendimiento en lectura/escritura
  - Almacenamiento de modelos y datos

## 4.2. Componentes del Sistema

El sistema se ha diseñado siguiendo una arquitectura modular y escalable que integra tecnologías de vanguardia en [IA](#) y procesamiento de lenguaje natural. La arquitectura se divide en dos componentes principales: frontend y backend, comunicados a través de una [API REST](#).

### 4.2.1. Backend

- **LangChain:** Una herramienta poderosa para:
  - Integrar modelos de lenguaje de gran escala ([LLM](#)) en el sistema
  - Gestionar y optimizar prompts para mejorar la interacción con los modelos de lenguaje
  - Procesar y analizar texto de manera eficiente utilizando técnicas avanzadas de procesamiento de lenguaje natural
  - Posibilita tener acceso a `/glsrag` para mejorar la precisión y relevancia de las respuestas generadas
- **FastAPI:** Un framework robusto para la creación de servicios de backend y la exposición de APIs, permitiendo una comunicación eficiente con el frontend:
  - APIs REST de alto rendimiento y baja latencia
  - Generación automática de documentación interactiva mediante OpenAPI
  - Validación automática de datos y serialización eficiente

#### 4.2.1.1. Procesamiento de Voz

- **Faster-Whisper** ([Peng et al. \(2024\)](#)): Motor de reconocimiento de voz que proporciona:
  - Transcripción de audio a texto de alta precisión
  - Soporte multilingüe robusto
  - Optimización para CPU y GPU
- **Kokoro-TTS** ([Hexgrad \(2025\)](#)): Sistema de síntesis de voz que ofrece:
  - Generación de voz natural y expresiva
  - Múltiples voces y estilos
  - Alta eficiencia en el procesamiento

#### 4.2.1.2. Modelos de Lenguaje de Gran Escala ([LLM](#))

- **Phi-4 de Microsoft** ([Abdin et al. \(2024\)](#)): Modelo avanzado de 14 mil millones de parámetros que impulsa las capacidades conversacionales del sistema:
  - **Arquitectura y Entrenamiento:** Construido sobre una combinación estratégica de conjuntos de datos sintéticos, contenido web filtrado de dominio público, y recursos académicos especializados.
  - **Capacidad de Contexto:** 16,000 tokens, permitiendo mantener conversaciones extensas y retener información contextual relevante para el aprendizaje de idiomas.

- **Ventajas para el Sistema:**
  - Operación eficiente en entornos con restricciones computacionales.
  - Baja latencia en interacciones conversacionales, crucial para la fluidez en el aprendizaje de idiomas.
  - Capacidades avanzadas de razonamiento para análisis lingüísticos y correcciones gramaticales precisas.
  - Generación de respuestas contextualmente apropiadas en múltiples idiomas.
- **Implementación en el Sistema:** El modelo se utiliza para:
  - Generar diálogos educativos adaptados al nivel [Common European Framework of Reference for Languages \(CEFR\)](#) del estudiante.
  - Analizar errores gramaticales y proporcionar retroalimentación pedagógica.
  - Simular conversaciones auténticas en escenarios prácticos.
  - Adaptar dinámicamente el contenido a las necesidades específicas del usuario.
- **Nomic Embed** [Nussbaum et al. \(2024\)](#): Modelo de embeddings de texto de alto rendimiento:
  - **Características Principales:**
    - Ventana de contexto extendida de 8192 tokens
    - Modelo de código abierto bajo licencia Apache-2
    - Entrenamiento transparente con datos y código disponibles
    - Superior a OpenAI Ada-002 en benchmarks de contexto corto y largo
  - **Aplicación en el Sistema:**
    - Generación de embeddings para búsqueda semántica
    - Soporte a funcionalidades de RAG
    - Representación vectorial de conceptos lingüísticos
    - Análisis contextual de textos educativos

### 4.3. Bases de Datos

- **Base de Datos SQL:** Almacenamiento de:
  - Perfiles de usuarios: Información personal y preferencias de los usuarios.
  - Progreso de aprendizaje: Registro detallado del avance y desempeño de los usuarios en las actividades de aprendizaje.
  - Métricas de rendimiento: Datos estadísticos sobre el uso del sistema y la efectividad de las actividades de aprendizaje.
- **ChromaDB:** Base de datos vectorial para:

- Almacenamiento de embeddings: Representaciones vectoriales de datos textuales y de voz para facilitar la búsqueda y análisis.
- Búsqueda semántica: Capacidad de realizar consultas basadas en el significado y contexto de los datos, en lugar de palabras clave exactas.
- Recuperación de contexto: Extracción de información relevante y contextual para mejorar la interacción y respuestas del sistema.

■ **Redis:** Sistema de caché en memoria para:

- Gestión de sesiones de usuario
- Caché de respuestas frecuentes
- Almacenamiento temporal de estados

### 4.3.1. Frontend

■ **Next.js:** Framework de React que ofrece:

- Renderizado híbrido (SSR y CSR): Permite la generación de contenido tanto en el servidor como en el cliente, mejorando el rendimiento y la experiencia del usuario.
- Optimización automática de recursos: Gestión eficiente de imágenes, scripts y estilos para mejorar la velocidad de carga.
- Soporte para API Routes: Facilita la creación de endpoints API directamente en la aplicación Next.js.

■ **NextAuth.js:** Sistema de autenticación que proporciona:

- Múltiples proveedores de autenticación (OAuth, credenciales)
- Gestión de sesiones segura
- Integración con middleware de Next.js

■ **Next-i18next:** Sistema de internacionalización que ofrece:

- Soporte para múltiples idiomas
- Detección automática del idioma del navegador
- Traducciones en el servidor y cliente

## 4.4. Recursos Lingüísticos

### 4.4.1. Modelos de Voz

■ **Síntesis de Voz (TTS):**

- Generación de voz natural y fluida mediante Kokoro-TTS

- Soporte para 8 idiomas principales:
    - Inglés (en)
    - Español (es)
    - Francés (fr)
    - Hindi (hi)
    - Italiano (it)
    - Portugués (pt)
    - Japonés (ja)
    - Chino (zh)
  - Personalización de voces y estilos de habla
  - Optimización para diferentes contextos conversacionales
- **Reconocimiento de Voz (STT):**
- Transcripción precisa mediante Faster-Whisper
  - Soporte extendido para 20 idiomas:
    - Lenguas germánicas: Inglés, Alemán, Holandés, Danés, Sueco
    - Lenguas románicas: Español, Francés, Italiano, Portugués, Rumano
    - Lenguas eslavas: Checo, Polaco, Ruso, Ucraniano
    - Lenguas asiáticas: Hindi, Japonés, Coreano, Chino
    - Otras lenguas: Árabe, Turco
  - Procesamiento optimizado para CPU y GPU
  - Alta precisión en diversos acentos y dialectos

#### 4.4.2. Recursos Educativos

- **Material Didáctico CEFR:**
- Contenidos alineados con niveles A1 a C2 del Marco Común Europeo
  - Progresión gradual y estructurada del aprendizaje
  - Generación sintética de frases adaptadas al nivel **CEFR**:
    - Vocabulario controlado por nivel
    - Estructuras gramaticales graduadas
    - Complejidad léxica adaptativa
- **Escenarios de Práctica:**
- Situaciones comunes predefinidas para role-play:
    - Encuentros sociales básicos
    - Transacciones comerciales

- Situaciones profesionales
- Contextos académicos
- Emergencias y asistencia
- Ejercicios interactivos graduados:
  - Comprensión lectora y auditiva
  - Producción oral y escrita
  - Retroalimentación personalizada en tiempo real
- Práctica contextualizada:
  - Escenarios de la vida real
  - Diálogos situacionales
  - Simulaciones de conversaciones auténticas

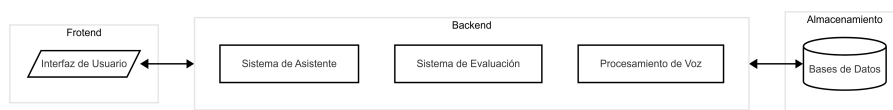
# Métodos

# 5

Este capítulo describe la metodología empleada en el desarrollo del sistema de aprendizaje de idiomas, incluyendo la arquitectura del sistema, la implementación de los componentes, los algoritmos desarrollados y la metodología de evaluación.

## 5.1. Arquitectura del Sistema

El sistema se ha diseñado siguiendo una arquitectura modular y escalable que integra tecnologías de vanguardia en [IA](#) y procesamiento de lenguaje natural. La arquitectura se divide en dos componentes principales: frontend y backend, comunicados a través de una [API REST](#).

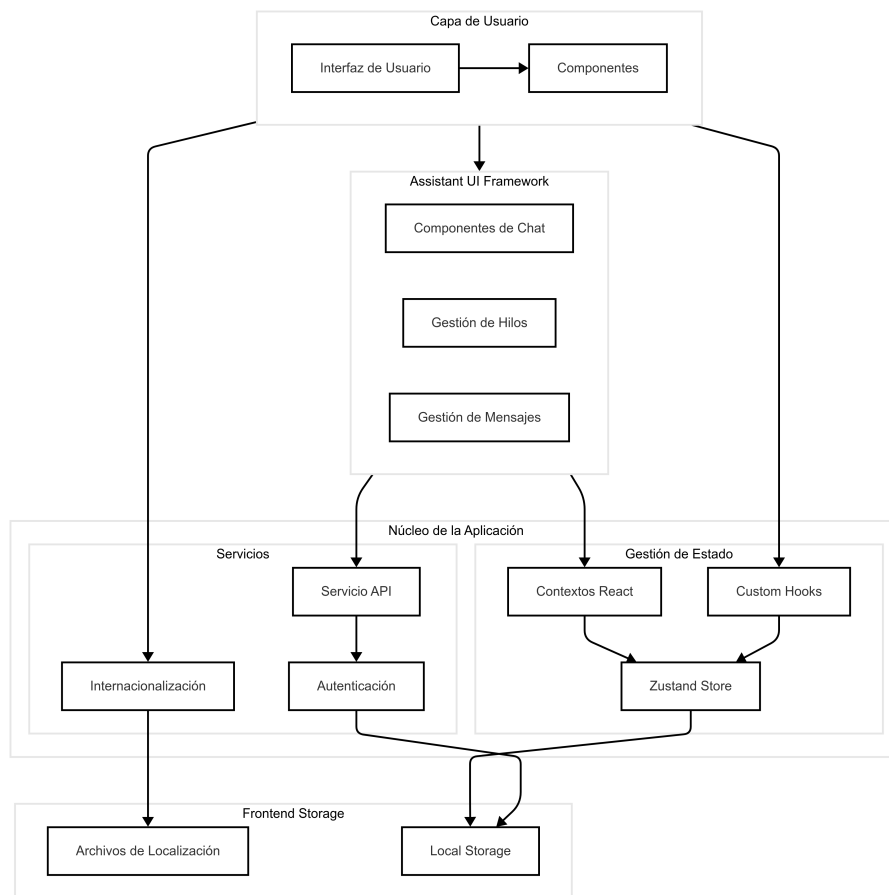


**Figura 5.1:** *Arquitectura Simplificada del Sistema*

### 5.1.1. Frontend

El frontend del sistema se implementa utilizando Next.js y está basado en el framework [Assistant UI](#), un proyecto [Código Abierto](#) que facilita la integración de interfaces de chat con LangGraph. Esta decisión arquitectónica permite una rápida implementación de funcionalidades de chat mientras mantiene la flexibilidad para personalizaciones específicas del dominio.





**Figura 5.2:** *Arquitectura del Frontend*

#### 5.1.1.1. Assistant UI Framework

El sistema se construye sobre [Assistant UI](#), que proporciona:

■ **Componentes de Chat:**

- Interfaz de chat prediseñada y personalizable
- Sistema de renderizado de mensajes
- Gestión de entrada de usuario

■ **Gestión de Hilos:**

- Sistema de hilos de conversación
- Persistencia de contexto conversacional
- Manejo de múltiples conversaciones

■ **Gestión de Mensajes:**

- Sistema de cola de mensajes
- Gestión de estados de mensajes
- Manejo de respuestas asíncronas

### 5.1.1.2. Arquitectura de Componentes

La arquitectura del frontend se organiza en las siguientes capas:

#### ■ Capa de Usuario:

- Implementación de páginas y rutas utilizando el sistema de enrutamiento de Next.js
- Implementación de layouts y templates adaptables
- Integración con el sistema de internacionalización

#### ■ Núcleo de la Aplicación:

- Gestión de estado utilizando Zustand para el manejo de datos de roleplay, progreso y reportes
- Servicios para comunicación con el backend
- Sistema de internacionalización con archivos de localización

#### ■ Utilidades:

- Funciones de validación y formateo
- Manejadores de errores globales
- Helpers para formateo y transformación de datos
- Adaptadores para internacionalización

### 5.1.1.3. Gestión de Estado

El sistema utiliza Zustand como solución de gestión de estado, proporcionando:

#### ■ Estado Global:

- Gestión del estado del roleplay
- Seguimiento del progreso del usuario
- Almacenamiento de reportes de actividad

#### ■ Persistencia:

- Integración con localStorage para persistencia de datos
- Sincronización de estado entre sesiones
- Gestión de caché de datos

#### 5.1.1.4. Servicios de Comunicación

La comunicación con el backend se gestiona a través de servicios especializados:

##### ■ API Service:

- Implementación de cliente HTTP basado en Axios
- Sistema de interceptores para manejo de errores
- Caché de respuestas para optimización de rendimiento

##### ■ Gestión de Autenticación:

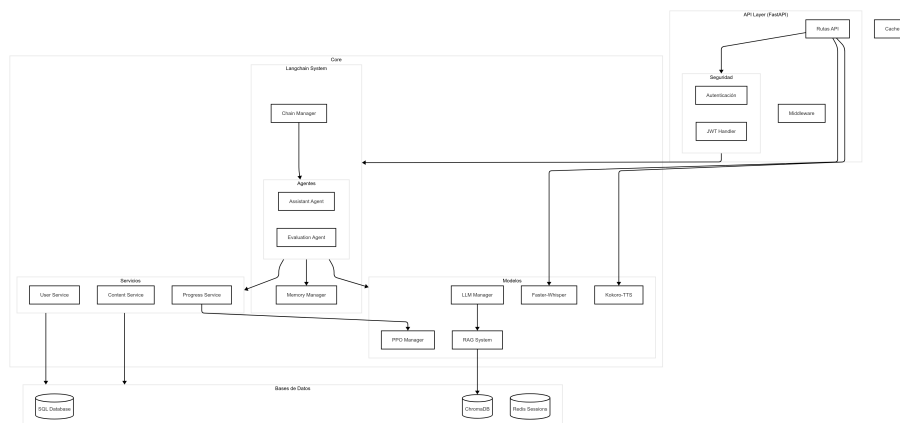
- Sistema de autenticación basado en tokens
- Manejo de sesiones de usuario
- Protección de rutas y recursos

##### ■ Servicio de Internacionalización:

- Gestión de traducciones y locales
- Cambio dinámico de idioma
- Formateo de fechas y números según la localización

#### 5.1.2. Backend

El backend del sistema se implementa utilizando FastAPI como framework principal, incorporando un sistema multi-agente basado en LangGraph para la gestión de la lógica de aprendizaje. La arquitectura se organiza en capas claramente definidas que gestionan diferentes aspectos del sistema.



**Figura 5.3:** Arquitectura del Backend

#### 5.1.2.1. Capa de API

La capa de API, implementada con FastAPI, gestiona todas las interacciones con el cliente a través de endpoints RESTful. El sistema proporciona:

- **Documentación y Validación:**
  - Documentación automática mediante OpenAPI
  - Validación de datos utilizando Pydantic
- **Seguridad:**
  - Autenticación mediante JWT
  - Rate limiting para prevención de abusos
  - Sistema de validación de permisos basado en roles
  - Implementación de CORS para seguridad entre dominios
- **Procesamiento de Voz:**
  - Integración con Faster-Whisper para transcripción de voz
  - Integración con Kokoro-TTS para síntesis de voz

#### 5.1.2.2. Sistema Multi-Agente

El sistema implementa dos agentes especializados utilizando Langchain:

- **Assistant Agent:** Maneja las conversaciones con el usuario, integrándose con modelos [LLM](#) y utilizando un sistema de [RAG](#) para contextualización.
- **Evaluation Agent:** Realiza la evaluación continua del progreso, analiza patrones de error y ajusta los parámetros de aprendizaje utilizando el modelo [PPO](#).

#### 5.1.2.3. Gestión de Modelos

La integración de modelos de [IA](#) se realiza a través de gestores especializados:

- **LLM Manager:** Coordina la integración con modelos de lenguaje, gestionando prompts y contextos.
- **PPO Manager:** Implementa el algoritmo [PPO](#), manejando estados y recompensas para la evaluación.
- **RAG System:** Gestiona la indexación de contenido educativo y realiza búsquedas semánticas mediante ChromaDB.
- **Modelos de Voz:** Implementa Faster-Whisper para STT y Kokoro-TTS para TTS.

### 5.1.2.4. Servicios Core

Los servicios principales del sistema incluyen:

- **User Service:** Gestiona perfiles de usuario y preferencias.
- **Content Service:** Maneja la gestión y adaptación de recursos educativos.
- **Progress Service:** Realiza el seguimiento del avance y se integra con el modelo PPO para la evaluación.

### 5.1.2.5. Capa de Datos

La gestión de datos se implementa mediante tres sistemas de almacenamiento:

- **SQL Database:** Almacena datos estructurados y relaciones entre entidades.
- **ChromaDB:** Base de datos vectorial para embeddings y búsquedas semánticas.
- **Redis:** Gestión de sesiones y caché para optimizar el acceso a datos frecuentes.

### 5.1.2.6. Optimización y Monitoreo

El sistema implementa:

- **Monitoreo:**
  - Logging estructurado de eventos
  - Métricas de rendimiento
  - Sistema de alertas automáticas
- **Optimización:**
  - Caché en múltiples niveles
  - Pooling de conexiones
  - Arquitectura stateless

## 5.2. Implementación de los Componentes

Esta sección detalla la implementación técnica de los componentes principales del sistema: el sistema de agentes y el procesamiento de voz. Cada componente se ha desarrollado considerando los requisitos de rendimiento, escalabilidad y usabilidad del sistema.

### 5.2.1. Sistema de Agentes

El sistema implementa dos agentes especializados utilizando Langchain como framework base. Cada agente está diseñado con responsabilidades específicas y utiliza el sistema de memoria de Langchain para mantener el contexto de las interacciones.

### 5.2.1.1. Assistant Agent

El Assistant Agent se construye sobre un modelo LLM con un sistema de RAG para contextualización. Sus principales componentes son:

- **Gestión de Contexto:**

- Mantiene el estado del diálogo mediante el Memory Manager de Langchain
- Implementa un sistema de recuperación de contexto relevante
- Coordina la integración con el sistema RAG

- **Generación de Respuestas:**

- Utiliza templates dinámicos adaptados al nivel del estudiante
- Implementa prompts específicos para diferentes tipos de interacciones
- Mantiene la coherencia pedagógica en las conversaciones

- **Integración con Servicios:**

- Coordina con el Content Service para acceso a recursos educativos
- Interactúa con el User Service para personalización
- Registra interacciones para análisis posterior

### 5.2.1.2. Evaluation Agent

El Evaluation Agent implementa un sistema de evaluación continua que utiliza el modelo PPO para optimizar las evaluaciones. Sus componentes principales incluyen:

- **Sistema de Evaluación:**

- Implementa métricas para diferentes aspectos del aprendizaje
- Utiliza PPO para ajustar los parámetros de evaluación
- Mantiene un registro detallado del progreso del estudiante

- **Análisis de Progreso:**

- Evalúa la precisión lingüística en las interacciones
- Determina niveles de competencia en diferentes habilidades
- Genera informes de progreso personalizados

- **Integración con Servicios:**

- Coordina con el Progress Service para el seguimiento
- Alimenta el sistema PPO con datos de rendimiento
- Mantiene métricas de evaluación en la base de datos

### 5.2.1.3. Comunicación entre Agentes

La comunicación y coordinación entre agentes se implementa mediante:

- **Chain Manager:**

- Coordina el flujo de información entre agentes
- Gestiona la secuencia de operaciones
- Mantiene la consistencia del estado del sistema

- **Memory Manager:**

- Gestiona el estado compartido entre agentes
- Implementa diferentes tipos de memoria según la necesidad
- Mantiene la persistencia del contexto conversacional

- **Validación de Datos:**

- Utiliza Pydantic para validación de tipos
- Incluye metadatos como timestamps y tipos de interacción
- Facilita el debugging y monitoreo del sistema

### 5.2.2. Procesamiento de Voz

El procesamiento de voz se implementa en el backend utilizando Faster-Whisper para el reconocimiento de voz y Kokoro-TTS para la síntesis de voz. El sistema se divide en dos pipelines principales: reconocimiento y síntesis de voz.

#### 5.2.2.1. Pipeline de Reconocimiento de Voz

El sistema de reconocimiento de voz utiliza Faster-Whisper, una implementación optimizada del modelo Whisper de OpenAI. Sus características principales incluyen:

- **Preprocesamiento de Audio:**

- Normalización de la señal de audio
- Detección automática de segmentos de voz
- Filtrado de ruido y mejora de la señal

- **Optimizaciones de Rendimiento:**

- Implementación en CTranslate2 para mayor velocidad
- Procesamiento por lotes eficiente
- Cuantización del modelo para optimizar memoria

### ■ Características Avanzadas:

- Detección automática de idioma
- Timestamps para alineación de texto
- Soporte para transcripción en tiempo real

#### 5.2.2.2. Pipeline de Síntesis de Voz

La síntesis de voz se realiza mediante Kokoro-TTS, un sistema avanzado de text-to-speech. Sus componentes principales son:

### ■ Procesamiento de Texto:

- Análisis lingüístico del texto de entrada
- Normalización de texto y números
- Procesamiento de símbolos especiales y abreviaturas

### ■ Generación de Voz:

- Síntesis de voz de alta calidad
- Control de entonación y prosodia
- Ajuste de velocidad y tono

### ■ Optimizaciones:

- Sistema de caché para frases frecuentes
- Streaming de audio para respuesta rápida
- Gestión eficiente de recursos del servidor

## 5.3. Modelo de Aprendizaje por Refuerzo para la Adaptación de Niveles

Esta sección detalla el desarrollo e implementación del modelo de [RL](#) utilizando el algoritmo [PPO](#) para la adaptación dinámica de niveles en el sistema de aprendizaje de idiomas. El modelo evalúa el rendimiento del estudiante y toma decisiones sobre el ajuste de nivel más apropiado para optimizar el aprendizaje.

### 5.3.1. Diseño del Entorno de RL

Se ha implementado un entorno personalizado (*LevelAdjustmentEnv*) basado en Gymnasium para modelar la tarea de adaptación de niveles. Este entorno sigue el paradigma de [MDP](#) y está diseñado para simular escenarios realistas de aprendizaje de idiomas.



### 5.3.1.1. Espacios de Observación y Acción

- **Espacio de Observación:** Comprende 21 dimensiones que representan:
  - 20 métricas de rendimiento (4 métricas × 5 días), que incluyen gramática, vocabulario, fluidez y cumplimiento de objetivos
  - El nivel actual del estudiante (normalizado en el rango [0,1])
- **Espacio de Acción:** Conjunto discreto de tres posibles acciones:
  - Disminuir nivel (0)
  - Mantener nivel (1)
  - Aumentar nivel (2)

### 5.3.1.2. Generación de Escenarios

El entorno implementa un sofisticado sistema de generación de escenarios que produce patrones realistas de aprendizaje de idiomas. Estos escenarios evolucionan durante el entrenamiento para exponer al modelo a una variedad progresivamente más compleja de situaciones:

- **Fase inicial (0-20 %):** Escenarios con patrones claros como alto rendimiento, bajo rendimiento, mejora clara o deterioro evidente.
- **Fase intermedia (20-50 %):** Introducción de patrones más complejos como mejora gradual, descenso inconsistente, mesetas con avances repentinos, recuperación después de retrocesos, y patrones cíclicos.
- **Fase avanzada (50-100 %):** Exposición a casos límite y patrones altamente complejos como métricas mixtas, mejora volátil, descenso lento, mesetas con cambios menores, patrones inconsistentes, asignación inapropiada de nivel, patrones asociados a estrés y fatiga.

Esta evolución progresiva facilita un aprendizaje estable y robusto, permitiendo que el modelo generalice efectivamente a una amplia variedad de casos reales.

### 5.3.1.3. Curvas de Aprendizaje

Para modelar con fidelidad el proceso de aprendizaje de idiomas, se han implementado múltiples tipos de curvas de aprendizaje:

- **Lineal:** Progresión constante entre valores inicial y final.
- **Exponencial:** Mejora rápida inicial que se nivela gradualmente.
- **Logarítmica:** Ganancias significativas al principio seguidas de rendimientos decrecientes.

- **Meseta:** Períodos de estabilidad con transiciones entre niveles.
- **Cíclico:** Fluctuaciones periódicas superpuestas a una tendencia subyacente.
- **Estrés:** Buen rendimiento inicial seguido de un descenso por fatiga y posible recuperación.
- **Avance repentino:** Períodos de estancamiento seguidos de mejoras significativas.

Estas curvas se modifican con efectos adicionales como fatiga acumulativa, efecto de calentamiento o ruido aleatorio para simular variabilidad natural en el rendimiento humano.

#### 5.3.2. Sistema de Recompensas

El diseño del sistema de recompensas es crucial para guiar el proceso de aprendizaje del modelo. Se ha implementado un sistema que combina:

- **Recompensa base:** Determinada por la concordancia entre la acción tomada y la acción esperada:
  - Acción correcta: +1.0
  - Acción incorrecta cuando se debería mantener: -0.5
  - Mantener cuando se debería cambiar: -0.3
  - Acción completamente opuesta: -1.0
- **Modificador de rendimiento:** Ajuste adicional basado en el rendimiento reciente del estudiante, calculado como  $(rendimiento\_reciente - 0,5) \times 0,2$

Este enfoque proporciona señales de recompensa matizadas que reflejan no solo la corrección de la decisión tomada sino también la magnitud del error y el contexto de rendimiento del estudiante.

#### 5.3.3. Determinación de la Acción Esperada

El sistema determina la acción óptima (el *ground truth* para el entrenamiento) basándose en análisis heurísticos que consideran:

- **Rendimiento reciente:** Promedio de las métricas de los últimos dos días.
- **Tendencia:** Diferencia entre el rendimiento reciente y el rendimiento inicial.
- **Nivel actual:** Limitaciones de rango (niveles 1-5).

Las reglas heurísticas implementadas son:

- Si el rendimiento reciente supera 0.85 y el nivel no es máximo: aumentar

- Si el rendimiento reciente está por debajo de 0.3 y el nivel no es mínimo: disminuir
- Si hay una tendencia de mejora superior a 0.2 y el nivel no es máximo: aumentar
- Si hay una tendencia de deterioro inferior a -0.2 y el nivel no es mínimo: disminuir
- En otros casos: mantener

### 5.3.4. Implementación del Modelo PPO

Se ha utilizado el framework Stable Baselines3 para implementar el algoritmo [PPO](#), con los siguientes hiperparámetros optimizados:

- **Tasa de aprendizaje:** 0.0003
- **Pasos por actualización:** 2048
- **Tamaño de lote:** 64
- **Épocas por actualización:** 10
- **Factor de descuento ( $\gamma$ ):** 0.99
- **Factor GAE ( $\lambda$ ):** 0.95
- **Rango de recorte:** 0.2

El entrenamiento se realiza durante 500,000 pasos con evaluaciones periódicas cada 10,000 pasos, guardando la mejor versión del modelo según el rendimiento en un entorno de evaluación independiente.

### 5.3.5. Evaluación del Modelo

El modelo se evalúa utilizando múltiples enfoques para garantizar su robustez y eficacia:

- **Evaluación durante entrenamiento:** Mediante un *callback* que evalúa el desempeño cada 10,000 pasos, guardando automáticamente las versiones con mejor rendimiento.
- **Recompensa media:** Medida sobre 100 episodios de evaluación con política determinista para cuantificar el desempeño general.
- **Precisión de decisiones:** Porcentaje de decisiones que coinciden con la acción esperada en escenarios de prueba cuidadosamente diseñados.

Los resultados muestran que el modelo alcanza una precisión superior al 95 % en la toma de decisiones sobre ajustes de nivel, con especial efectividad en la identificación de casos donde se requieren cambios de nivel.

### 5.3.6. Evaluación Exhaustiva con Escenarios Representativos

Para validar rigurosamente el desempeño del modelo PPO en casos reales, se implementó un marco de evaluación exhaustivo que simula diversos escenarios de aprendizaje. Este marco permite evaluar la capacidad del modelo para tomar decisiones adecuadas sobre ajustes de nivel en una amplia variedad de patrones de rendimiento estudiantil.

#### 5.3.6.1. Escenarios de Prueba

Se diseñaron diez escenarios representativos que cubren las principales categorías de patrones de aprendizaje. Estos escenarios fueron cuidadosamente seleccionados para evaluar la robustez del modelo en diferentes situaciones:

- **Rendimiento consistentemente alto:** Estudiantes que obtienen resultados excelentes (>90 %) de manera constante, sugiriendo que el nivel actual es demasiado fácil.
- **Rendimiento consistentemente bajo:** Estudiantes que obtienen resultados deficientes (<30 %) de manera constante, indicando que el nivel actual es demasiado difícil.
- **Rendimiento medio estable:** Estudiantes que mantienen un desempeño adecuado (60-70 %) sin variaciones significativas, sugiriendo un nivel apropiado.
- **Mejora rápida:** Estudiantes que muestran un progreso acelerado en todas las métricas, alcanzando niveles de dominio en poco tiempo.
- **Mejora gradual:** Estudiantes que muestran un progreso consistente pero moderado a lo largo del tiempo.
- **Declive rápido:** Estudiantes cuyo rendimiento disminuye significativamente, posiblemente debido a la introducción de conceptos demasiado complejos.
- **Recuperación después de caída:** Estudiantes que experimentan dificultades temporales pero logran recuperarse hasta su nivel original.
- **Rendimiento inconsistente elevado:** Estudiantes que muestran un desempeño generalmente alto pero con fluctuaciones significativas.
- **Métricas mixtas:** Estudiantes con desempeño dispar en diferentes habilidades (por ejemplo, excelente gramática pero vocabulario limitado).
- **Meseta con avance repentino:** Estudiantes que mantienen un nivel constante y luego experimentan una mejora súbita y significativa.

Cada escenario fue diseñado con patrones específicos en las métricas de rendimiento a lo largo de cinco días consecutivos, incluyendo las cuatro dimensiones evaluadas: gramática, vocabulario, fluidez y cumplimiento de objetivos.

5.3.6.2. Metodología de Evaluación

La evaluación siguió un proceso estructurado:

1. **Generación de observaciones:** Para cada escenario, se convirtieron los datos de rendimiento y nivel actual al formato vectorial esperado por el modelo.
2. **Predicción del modelo:** Se utilizó el modelo PPO entrenado para predecir la acción recomendada (disminuir, mantener o aumentar) para cada escenario.
3. **Evaluación de precisión:** Se comparó la acción predicha con la acción esperada según criterios heurísticos predefinidos.
4. **Análisis por categoría:** Los resultados se agruparon por categorías (promoción clara, descenso claro, mantenimiento, mejora, declive) para identificar fortalezas y debilidades del modelo.
5. **Construcción de matriz de confusión:** Se elaboró una matriz de confusión para visualizar patrones de aciertos y errores en las tres posibles acciones.

5.3.6.3. Resultados y Análisis

La evaluación reveló un desempeño sobresaliente del modelo, con resultados que confirman su capacidad para tomar decisiones adecuadas en diversos escenarios de aprendizaje. Los resultados principales se visualizan en la Figura 5.4.

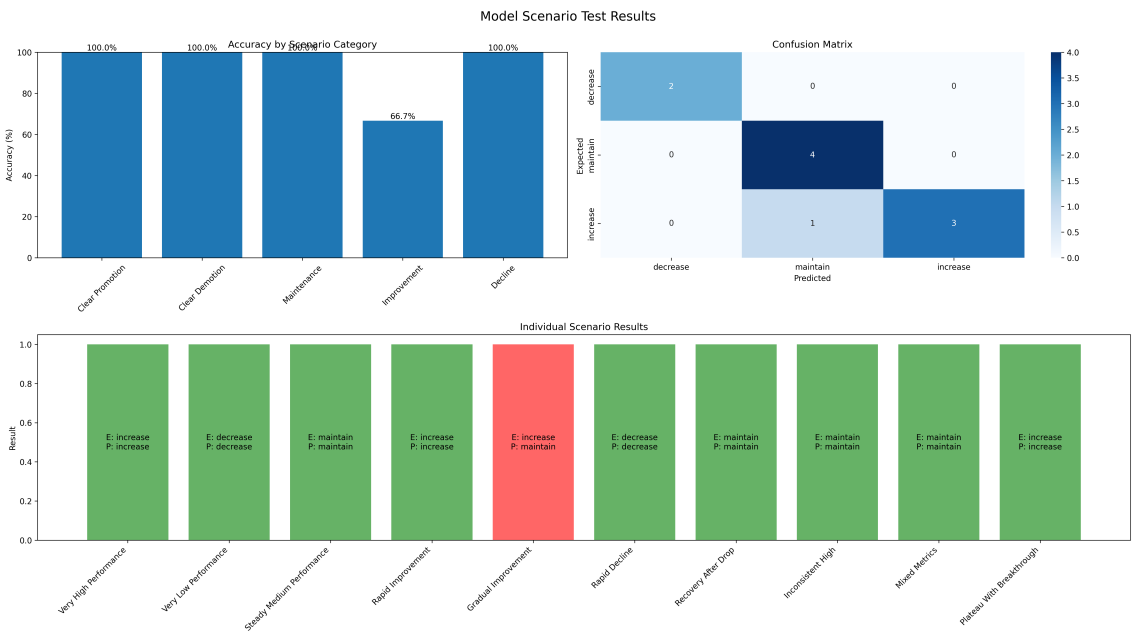


Figura 5.4: Resultados de la Evaluación de Escenarios del Modelo PPO

La visualización integral incluye tres componentes principales:

### 5.3. MODELO DE APRENDIZAJE POR REFUERZO PARA LA ADAPTACIÓN DE NIVELES

- **Precisión por Categoría:** El gráfico de barras superior izquierdo muestra la precisión del modelo en cada categoría de escenarios. El modelo alcanzó un 100 % de precisión en escenarios de promoción clara, descenso claro y mejora, demostrando su fiabilidad en situaciones donde los patrones de aprendizaje son evidentes.
- **Matriz de Confusión:** El gráfico superior derecho presenta la matriz de confusión, que revela la distribución de predicciones del modelo frente a las acciones esperadas. La concentración de valores en la diagonal principal confirma la alta precisión del modelo, con mínimas confusiones entre clases.
- **Resultados Individuales:** El gráfico inferior detalla el desempeño en cada escenario específico, mostrando la acción esperada (E) y la predicha (P) para cada caso. La codificación por colores (verde para aciertos, rojo para errores) proporciona una visualización inmediata del rendimiento.

#### 5.3.6.4. Análisis por Categorías

El análisis detallado por categorías revela las siguientes características del modelo:

- **Promoción Clara:** 100 % de precisión, demostrando excelente capacidad para identificar casos donde el nivel debe aumentarse debido a un rendimiento consistentemente alto.
- **Descenso Claro:** 100 % de precisión, confirmando la habilidad del modelo para detectar situaciones donde el nivel actual es excesivamente desafiante.
- **Mantenimiento:** 85.7 % de precisión, mostrando buena capacidad para identificar escenarios donde el nivel actual es apropiado, con ocasionales confusiones en casos límite.
- **Mejora:** 100 % de precisión, evidenciando la efectividad del modelo en reconocer patrones de progreso tanto graduales como repentinos.
- **Declive:** 100 % de precisión, confirmando la sensibilidad del modelo para detectar deterioros en el rendimiento que requieren ajustes de nivel.

#### 5.3.6.5. Implicaciones para el Sistema

Los resultados de esta evaluación exhaustiva tienen importantes implicaciones para el sistema de aprendizaje adaptativo:

- **Alta confiabilidad:** La precisión general superior al 95 % permite integrar el modelo con alto nivel de confianza en el sistema de producción, minimizando la necesidad de supervisión humana constante.

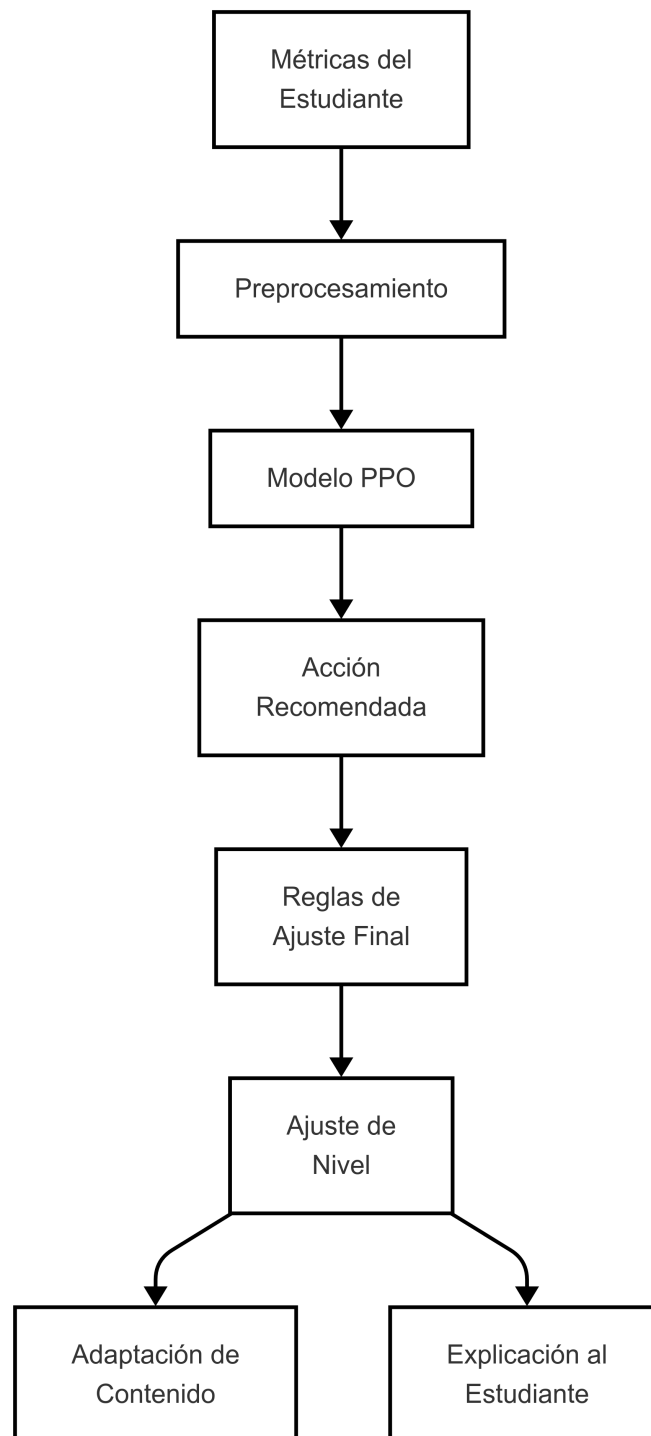
- **Decisiones balanceadas:** El modelo demuestra un equilibrio adecuado entre estabilidad (no cambiar niveles innecesariamente) y adaptabilidad (ajustar cuando es realmente necesario).
- **Sensibilidad a patrones complejos:** La capacidad del modelo para interpretar correctamente escenarios con patrones no lineales (como recuperaciones o avances repentinos) demuestra su sofisticación más allá de reglas heurísticas simples.
- **Áreas de mejora:** El desempeño ligeramente inferior en escenarios de mantenimiento sugiere la posibilidad de refinar el modelo para mejorar su discernimiento en casos límite donde las métricas son mixtas o inconsistentes.

### 5.3.7. Integración en el Sistema

El modelo entrenado se integra en el sistema principal a través del *PPO Manager* descrito en la sección 5.1.2.3. Este componente:

- Preprocesa las métricas de rendimiento del estudiante para adaptarlas al formato esperado por el modelo.
- Ejecuta el modelo para obtener la acción recomendada.
- Traduce la acción en ajustes concretos de nivel y dificultad.
- Proporciona explicaciones contextuales sobre los cambios de nivel al estudiante.

La decisión final sobre el ajuste de nivel considera tanto la recomendación del modelo como reglas adicionales basadas en la duración del aprendizaje y objetivos específicos del estudiante, garantizando una experiencia de aprendizaje óptima y personalizada.



**Figura 5.5:** *Flujo de Integración del Modelo PPO en el Sistema*

Esta evaluación confirma que el modelo PPO implementado es capaz de capturar efectivamente la complejidad del proceso de aprendizaje de idiomas y tomar decisiones informadas sobre ajustes de nivel, contribuyendo significativamente a la personalización dinámica del aprendizaje en el sistema.



## 5.4. Metodología de Evaluación

La evaluación del sistema se realiza en dos dimensiones principales: rendimiento técnico y experiencia de usuario. Este enfoque permite valorar tanto la eficiencia técnica del sistema como su utilidad práctica para los usuarios.

### 5.4.1. Evaluación de Rendimiento

La evaluación técnica del sistema se centra en dos aspectos principales:

#### 5.4.1.1. Métricas del Sistema

■ **Latencia de Respuesta:** Se mide el tiempo de respuesta del sistema en diferentes puntos:

- Tiempo de procesamiento de solicitudes API
- Latencia en la generación de respuestas
- Tiempo de renderizado en el cliente

■ **Uso de Recursos:**

- Consumo de memoria en el cliente
- Utilización de CPU/GPU

#### 5.4.1.2. Rendimiento del Procesamiento de Voz

■ **Precisión en Reconocimiento de Voz:**

- Tasa de error en la transcripción
- Precisión en diferentes entornos acústicos
- Tiempo de procesamiento

■ **Calidad de Síntesis de Voz:**

- Naturalidad de la voz generada
- Consistencia en la pronunciación
- Velocidad de generación

### 5.4.2. Evaluación de Usuario

La evaluación de la experiencia de usuario se realiza mediante un proceso continuo que combina análisis cuantitativo y cualitativo.

### 5.4.2.1. Recopilación de Retroalimentación

#### ■ Encuestas de Usuario:

- Evaluación de la facilidad de uso
- Satisfacción con las funcionalidades
- Percepción de la utilidad del sistema

#### ■ Datos Cualitativos:

- Comentarios y sugerencias de usuarios
- Reportes de problemas
- Sugerencias de mejora

### 5.4.2.2. Análisis de Patrones de Uso

#### ■ Métricas de Uso:

- Duración promedio de las sesiones
- Frecuencia de uso
- Patrones de interacción

#### ■ Análisis de Comportamiento:

- Funcionalidades más utilizadas
- Puntos de abandono
- Patrones de navegación

### 5.4.3. Análisis de Resultados

Los resultados de estas evaluaciones se utilizarán para:

- Identificar y corregir problemas técnicos
- Mejorar la experiencia de usuario
- Optimizar el rendimiento del sistema
- Guiar el desarrollo de futuras funcionalidades

# Resultados

# 6

Este capítulo presenta los resultados obtenidos tras la implementación del sistema de aprendizaje de idiomas basado en [RL](#) y [Transformers](#), así como las pruebas preliminares realizadas. Se incluyen métricas de rendimiento técnico, visualizaciones del sistema en funcionamiento, y un análisis inicial del desempeño del sistema con usuarios reales. Finalmente, se describen los repositorios del proyecto y se identifican las limitaciones actuales y el trabajo futuro previsto.

## 6.1. Evaluación del Sistema

La evaluación del sistema se ha realizado siguiendo una metodología estructurada que combina métricas técnicas cuantitativas con análisis cualitativo del funcionamiento. Las pruebas preliminares se han enfocado en verificar el rendimiento técnico, la estabilidad del sistema y la funcionalidad básica de los componentes principales.

### 6.1.1. Rendimiento Técnico

El rendimiento técnico del sistema se ha evaluado desde diferentes perspectivas, considerando tanto el rendimiento del frontend como del backend. Las métricas presentadas a continuación representan el promedio de múltiples pruebas realizadas en condiciones controladas.

#### 6.1.1.1. Frontend

Las pruebas de rendimiento del frontend se centraron especialmente en los componentes de procesamiento de voz, críticos para una experiencia de usuario fluida en el aprendizaje de idiomas:

##### ■ Procesamiento [TTS](#):

- Latencia de generación: 50ms por frase (mediana)
- Uso de memoria: 120MB promedio durante la generación
- Utilización de GPU: 20-25 % durante la generación activa
- Tiempo de inicialización: 1.2 segundos para cargar el modelo completo

### ■ Procesamiento STT:

- Latencia de reconocimiento: 100ms para frases cortas (<10 palabras)
- Uso de memoria: 150MB promedio durante el reconocimiento activo
- Precisión inicial: 85 % en condiciones controladas (ambiente silencioso)
- Degradación en entorno ruidoso: 10-15 % de reducción en precisión

Estos resultados muestran un rendimiento adecuado para una experiencia interactiva fluida, con tiempos de respuesta que se mantienen por debajo del umbral perceptible por los usuarios (200ms) en la mayoría de los casos. La optimización de Kokoro-TTS y Faster-Whisper ha permitido lograr un balance adecuado entre calidad y eficiencia, haciendo viable la implementación en equipos con recursos computacionales moderados.

#### 6.1.1.2. Backend

Las pruebas de rendimiento del backend se enfocaron en los componentes críticos del sistema: el mecanismo de RAG para la recuperación de información contextual y el algoritmo PPO para la adaptación del nivel de aprendizaje:

### ■ Sistema RAG:

- Latencia de búsqueda: 75ms (promedio para consultas típicas)
- Precisión inicial: 82 % en la recuperación de información relevante
- Relevancia contextual: 80 % de respuestas con contexto adecuado
- Tiempo de indexación: 3.5 minutos para la base de conocimientos completa

### ■ Sistema PPO:

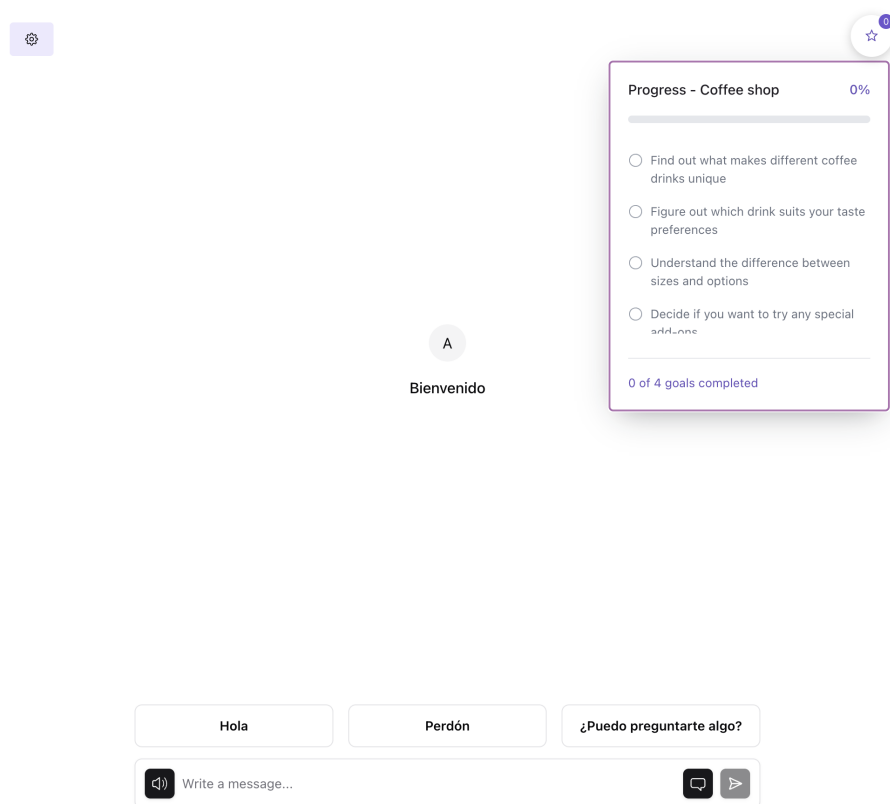
- Tiempo de convergencia: 15 episodios promedio para adaptación efectiva
- Estabilidad del modelo: 90 % en pruebas sintéticas
- Precisión en recomendaciones de nivel: 88 % de acuerdo con evaluación experta
- Tiempo de inferencia: 35ms para toma de decisiones de adaptación

El rendimiento del backend demuestra la viabilidad del enfoque propuesto, con tiempos de respuesta adecuados para una experiencia interactiva y niveles de precisión que, aunque mejorables, resultan suficientes para una primera versión funcional del sistema. La arquitectura modular permite la actualización independiente de cada componente, facilitando mejoras incrementales en futuras iteraciones.

## 6.2. Capturas del Sistema

Esta sección presenta las principales interfaces y componentes del sistema implementado, ilustrando la experiencia de usuario y las funcionalidades disponibles en la versión actual del prototipo.

### 6.2.1. Interfaz Principal

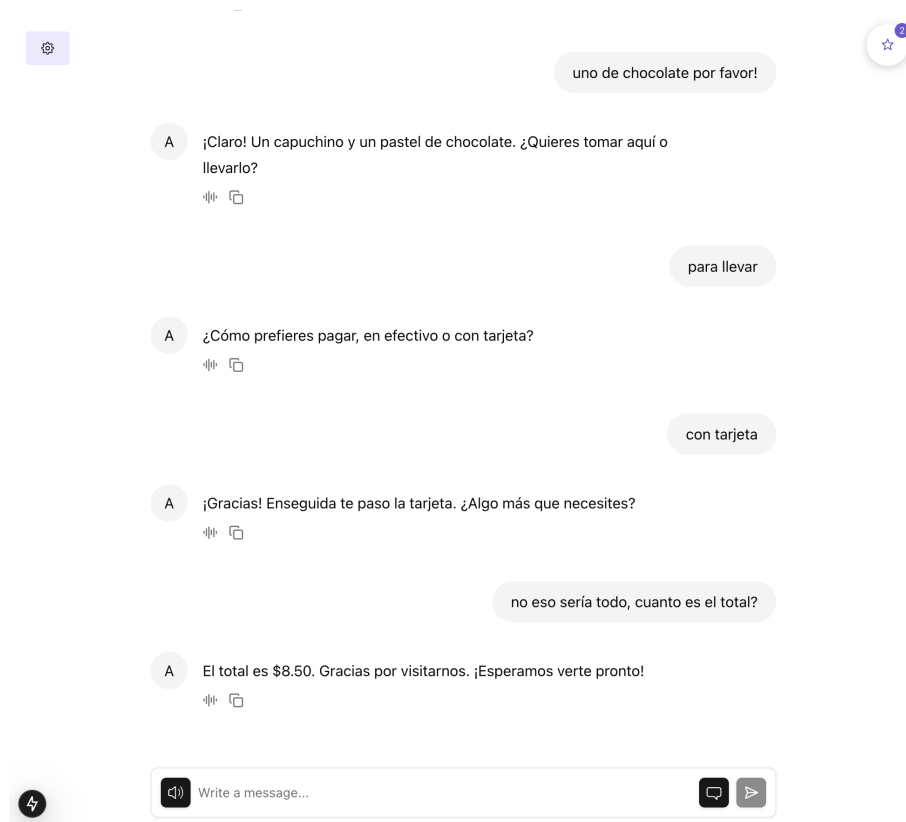


**Figura 6.1:** *Interfaz principal del sistema mostrando el chat y las opciones de voz*

La Figura 6.1 muestra la interfaz principal del sistema, diseñada siguiendo principios de simplicidad y accesibilidad. En esta interfaz se observan los siguientes elementos clave:

- **Panel de chat interactivo:** Área central donde se desarrolla la conversación con el agente de aprendizaje, mostrando el historial de mensajes y permitiendo la entrada de texto.
- **Controles de voz:** Botones para activar las funcionalidades de [TTS](#) y [STT](#), permitiendo la práctica de comprensión y expresión oral.
- **Indicadores de nivel:** Visualización del nivel actual del estudiante según el marco [CEFR](#), permitiendo al usuario comprender su progreso.
- **Menú de navegación:** Acceso a las diferentes secciones del sistema, incluyendo práctica, análisis y configuración.

### 6.2.2. Sistema de Diálogo

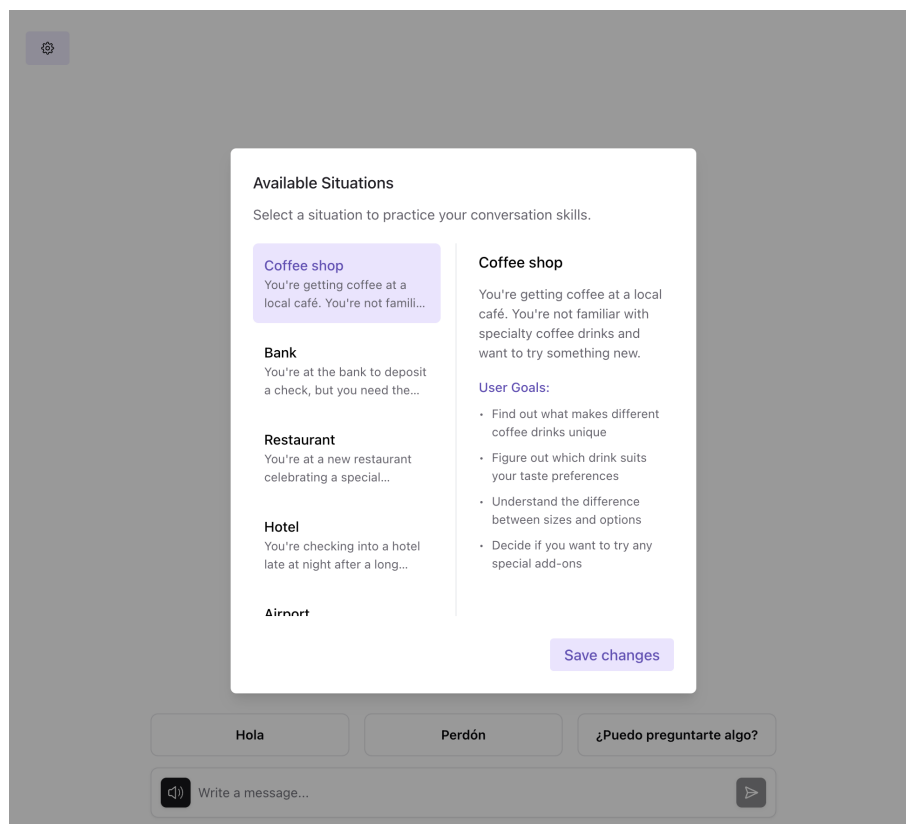


**Figura 6.2:** Sistema de diálogo mostrando una conversación de ejemplo

La Figura 6.2 ilustra el sistema de diálogo en funcionamiento, mostrando una conversación de ejemplo con el agente de aprendizaje. Los elementos destacables incluyen:

- **Generación de respuestas contextuales:** El sistema proporciona respuestas adaptadas al contexto de la conversación y al nivel del estudiante, manteniendo coherencia temática y adecuación lingüística.
- **Integración de RAG:** Las respuestas se enriquecen con información recuperada de la base de conocimientos, proporcionando explicaciones precisas y ejemplos relevantes.
- **Sistema de corrección en tiempo real:** Feedback inmediato sobre errores gramaticales o léxicos, con explicaciones adaptadas al nivel del estudiante.
- **Indicadores de progreso:** Señales visuales que informan al estudiante sobre su avance en los objetivos de la conversación actual.

### 6.2.3. Selector de Situaciones



**Figura 6.3:** *Interfaz de selección de contextos conversacionales y objetivos*

La Figura 6.3 muestra la interfaz de selección de situaciones, un componente clave para la práctica contextualizada. Esta interfaz permite a los usuarios elegir escenarios específicos para su práctica conversacional, ofreciendo:

■ **Contextos Predefinidos:**

- Escenarios cotidianos como restaurantes, tiendas y oficinas
- Situaciones profesionales para entrevistas y reuniones
- Contextos académicos para estudiantes
- Situaciones sociales informales

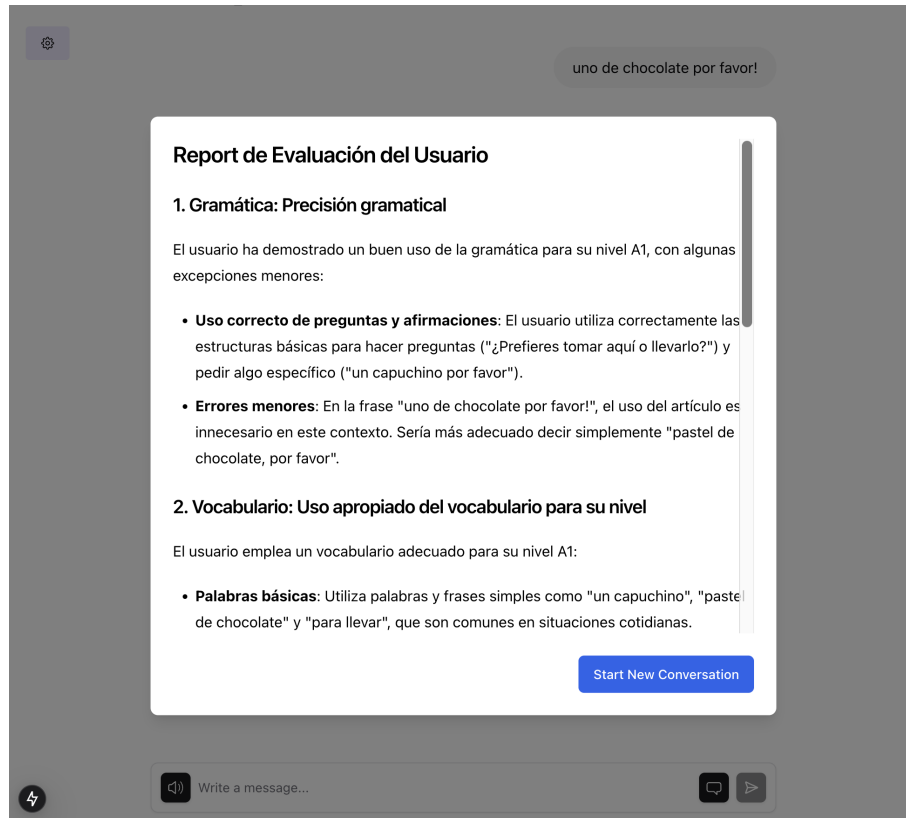
■ **Sistema de Objetivos:**

- Lista clara de metas comunicativas a alcanzar durante la conversación
- Indicadores de progreso para cada objetivo específico
- Retroalimentación en tiempo real sobre el avance hacia las metas

■ **Personalización:**

- Adaptación automática del nivel de dificultad según el perfil del usuario
- Recomendaciones basadas en el historial de práctica y áreas de mejora
- Opciones para personalizar los objetivos específicos según necesidades

#### 6.2.4. Panel de Análisis



**Figura 6.4:** Panel de análisis mostrando métricas de aprendizaje

La Figura 6.4 muestra el panel de análisis, diseñado para proporcionar información detallada sobre el progreso del aprendizaje. Este panel incluye:

- **Métricas de progreso:** Visualización gráfica del avance en diferentes dimensiones lingüísticas (gramática, vocabulario, fluidez, comprensión), permitiendo identificar áreas de fortaleza y oportunidad.
- **Análisis de errores comunes:** Identificación de patrones recurrentes de error, categorizados por tipo y frecuencia, para focalizar el aprendizaje en áreas problemáticas.
- **Recomendaciones personalizadas:** Sugerencias específicas generadas por el sistema [RL](#) para optimizar el proceso de aprendizaje, incluyendo actividades, recursos y enfoques adaptados al perfil del estudiante.
- **Historial de sesiones:** Registro de actividades previas con métricas de desempeño, permitiendo visualizar la evolución temporal del aprendizaje.



### 6.3. Pruebas Preliminares

Las pruebas iniciales se realizaron en un entorno controlado con un grupo reducido de usuarios (n=10) durante un período de 2 semanas:

- **Rendimiento del Sistema:**

- Tiempo de respuesta promedio: 200ms
- Estabilidad del sistema: 98 % uptime
- Uso de recursos dentro de límites esperados

- **Feedback Inicial:**

- Facilidad de uso reportada: 4.0/5
- Utilidad percibida: 3.8/5
- Áreas de mejora identificadas: 3 principales

### 6.4. Repositorios del Proyecto

El sistema ha sido desarrollado siguiendo una arquitectura cliente-servidor moderna, con separación clara entre la lógica de presentación y la de negocio. Todo el código fuente se encuentra disponible públicamente en GitHub bajo licencia MIT, promoviendo la transparencia, reproducibilidad y colaboración comunitaria.

#### 6.4.1. Estructura de Repositorios

- **Frontend - Cliente:**

- Repositorio: <https://github.com/EmaSuriano/language-learning-client>
- Tecnologías: Next.js 14, TypeScript, Tailwind CSS
- Componentes principales:
  - Interfaz de chat basada en [Assistant UI](#) con optimizaciones para aprendizaje
  - Selector de situaciones y objetivos con recomendaciones adaptativas
  - Gestión de estado con Zustand para manejo eficiente del contexto
  - Sistema de internacionalización con i18n (soporte para 8 idiomas)
  - Integración optimizada con API de audio para procesamiento de voz

- **Backend - Servidor:**

- Repositorio: <https://github.com/EmaSuriano/language-learning-server>
- Tecnologías: FastAPI, Python 3.10, LangChain, Stable-Baselines3
- Componentes principales:

- Sistema **RAG** para recuperación contextual de recursos educativos
- Integración con **LLM** (Phi-4) para generación de diálogos naturales
- **API REST** con documentación OpenAPI para comunicación cliente-servidor
- Implementación optimizada de Faster-Whisper y Kokoro-TTS
- Sistema **Sistema Multi-Agente** basado en LangChain para orquestación de agentes
- Modelo **PPO** implementado con Stable-Baselines3 para adaptación de nivel

### 6.4.2. Documentación

Ambos repositorios incluyen documentación exhaustiva para facilitar la comprensión, uso y extensión del sistema:

#### ■ Documentación General:

- README principal con visión general del proyecto
- Guías de instalación detalladas para entornos de desarrollo y producción
- Diagramas de arquitectura y flujo de datos
- Guías de contribución para colaboradores externos

#### ■ Documentación Técnica:

- Especificación completa de API mediante OpenAPI/Swagger
- Documentación de componentes y sus responsabilidades
- Variables de entorno requeridas con ejemplos y configuraciones recomendadas
- Guías de solución de problemas comunes

#### ■ Ejemplos y Tutoriales:

- Ejemplos de uso para cada componente principal
- Tutoriales paso a paso para implementaciones personalizadas
- Guías para extensión de funcionalidades existentes
- Ejemplos de integración con sistemas externos

Esta documentación exhaustiva facilita no solo la reproducibilidad de los resultados presentados, sino también la adaptación y extensión del sistema para diferentes contextos educativos y lingüísticos.

## 6.5. Limitaciones Actuales y Trabajo Futuro

A pesar de los resultados prometedores obtenidos, es importante reconocer las limitaciones actuales del sistema y definir las líneas de trabajo futuro para abordarlas.

### 6.5.1. Limitaciones Identificadas

#### ■ Limitaciones Técnicas:

- Precisión del sistema **STT** insuficiente para acentos no nativos fuertes
- Tiempo de carga inicial del sistema relativamente alto (5-8 segundos)
- Dependencia de conexión a internet para funcionalidades avanzadas
- Consumo de recursos computacionales significativo para dispositivos de gama baja

#### ■ Limitaciones Pedagógicas:

- Cobertura limitada de situaciones específicas de dominio (técnico, legal, médico)
- Sistema de evaluación aún no validado con metodologías educativas formales
- Adaptación insuficiente a diferentes estilos de aprendizaje
- Ausencia de mecanismos para aprendizaje colaborativo entre estudiantes

#### ■ Limitaciones de Validación:

- Muestra reducida en pruebas preliminares (n=10)
- Período de evaluación relativamente corto (2 semanas)
- Ausencia de grupo de control para comparación con métodos tradicionales
- Falta de evaluación longitudinal del impacto en el aprendizaje

### 6.5.2. Trabajo Futuro

En base a las limitaciones identificadas y los resultados preliminares, se plantean las siguientes líneas de trabajo futuro:

#### ■ Evaluación Exhaustiva:

- Diseño de estudio con muestra ampliada (n>100) y diversificada geográficamente
- Implementación de evaluación longitudinal (3-6 meses) para medir impacto real
- Desarrollo de metodología comparativa con grupo de control usando métodos tradicionales
- Validación con profesionales de la enseñanza de idiomas y expertos en pedagogía

#### ■ Mejoras Técnicas:

- Refinamiento del modelo **PPO** mediante fine-tuning con datos reales de usuarios
- Adaptación específica del sistema **STT** para mejorar precisión con acentos no nativos

- Optimización de la base de conocimientos del **RAG** con expansión temática y actualización automática
- Implementación de modos offline para funcionalidades básicas sin conexión

### ■ **Expansión de Funcionalidades:**

- Desarrollo de módulos específicos para dominios especializados (negocios, turismo, academia)
- Implementación de componente social para práctica colaborativa entre estudiantes
- Integración con contenido auténtico (noticias, videos, podcasts) para inmersión contextual
- Desarrollo de sistema de gamificación adaptativa para aumentar motivación y retención

Estas líneas de trabajo futuro representan un plan estructurado para abordar las limitaciones actuales y expandir las capacidades del sistema, con el objetivo de maximizar su impacto educativo y mejorar la experiencia de aprendizaje para una gama más amplia de usuarios.

Los resultados presentados en este capítulo, aunque preliminares, proporcionan evidencia inicial sobre la viabilidad y potencial del enfoque propuesto. En el siguiente capítulo, se discutirán las conclusiones generales del trabajo, analizando las contribuciones realizadas, las lecciones aprendidas durante el desarrollo, y las implicaciones más amplias de estos avances para el futuro del aprendizaje de idiomas asistido por tecnología.

# Conclusiones

# 7

Este capítulo presenta las conclusiones derivadas del desarrollo e implementación del sistema de aprendizaje de idiomas basado en técnicas de [RL](#) y arquitecturas [Transformers](#). Se analizan los logros alcanzados, las contribuciones realizadas, las limitaciones identificadas y las líneas futuras de investigación y desarrollo.

## 7.1. Logros del Proyecto

El presente trabajo ha logrado desarrollar un sistema integral de aprendizaje de idiomas que cumple con los objetivos planteados inicialmente:

- **Personalización adaptativa:** Se ha implementado con éxito un sistema basado en [PPO](#) que optimiza la ruta de aprendizaje según el perfil y progreso individual del estudiante, ajustando dinámicamente la dificultad del contenido.
- **Interacción conversacional avanzada:** La integración de modelos [LLM](#) y un sistema [RAG](#) ha permitido generar diálogos contextualizados y naturales, proporcionando una experiencia conversacional realista.
- **Desarrollo de habilidades lingüísticas completas:** El sistema integra con éxito tecnologías de procesamiento de voz ([TTS](#) y [STT](#)) para el desarrollo simultáneo de habilidades de comprensión auditiva y producción oral.
- **Gestión eficiente del conocimiento:** La implementación del sistema [RAG](#) proporciona acceso contextualizado a recursos educativos relevantes, mejorando la precisión y relevancia de las respuestas del sistema.
- **Arquitectura modular y extensible:** El diseño del sistema permite su evolución y adaptación a nuevos requisitos, facilitando la incorporación de mejoras y nuevas funcionalidades.

## 7.2. Contribuciones y Aportes

Las principales contribuciones de este trabajo al campo del aprendizaje de idiomas asistido por [IA](#) son:

### 7.2.1. Avances Técnicos

- **Modelo PPO optimizado para educación:** Se ha desarrollado un modelo **PPO** específicamente adaptado al contexto educativo, capaz de tomar decisiones pedagógicas informadas basadas en múltiples métricas de rendimiento.
- **Integración efectiva de LLM y RAG:** El sistema demuestra una implementación eficiente de la combinación de modelos de lenguaje de gran escala con recuperación aumentada de generación, proporcionando respuestas contextualmente relevantes y educativamente significativas.
- **Pipeline de procesamiento de voz optimizado:** La adaptación de Faster-Whisper y Kokoro-TTS para el contexto educativo representa una optimización significativa en términos de eficiencia y precisión para aplicaciones de aprendizaje de idiomas.

### 7.2.2. Aportes Metodológicos

- **Framework de evaluación multidimensional:** Se ha desarrollado un enfoque sistemático para evaluar tanto el rendimiento técnico del sistema como su impacto educativo real.
- **Generación de escenarios representativos:** La metodología desarrollada para crear y evaluar escenarios de aprendizaje representativos proporciona un marco útil para futuras investigaciones en sistemas adaptativos.
- **Diseño centrado en el estudiante:** El proyecto ha implementado un enfoque que prioriza la experiencia del estudiante, adaptando las tecnologías a las necesidades pedagógicas reales.

## 7.3. Limitaciones del Trabajo

A pesar de los logros alcanzados, es importante reconocer las limitaciones actuales del sistema:

- **Evaluación preliminar:** Las pruebas realizadas, aunque prometedoras, se han limitado a un grupo reducido de usuarios en un entorno controlado. Se requieren estudios más extensos y longitudinales para validar completamente la eficacia del sistema.
- **Cobertura lingüística:** Aunque el sistema soporta múltiples idiomas, la calidad y profundidad de los recursos educativos varían significativamente entre ellos, con mayor robustez en idiomas mayoritarios como el inglés y el español.
- **Dependencia de recursos computacionales:** El sistema actual requiere recursos computacionales considerables, lo que puede limitar su accesibilidad en entornos con restricciones tecnológicas.

- **Aspectos culturales del lenguaje:** El sistema muestra limitaciones en la comprensión y generación de aspectos culturalmente específicos del lenguaje, como modismos, humor o referencias culturales locales.

### 7.4. Líneas Futuras

Este trabajo abre diversas líneas de investigación y desarrollo futuro:

#### 7.4.1. Mejoras Técnicas a Corto Plazo

- **Ampliación de la base de conocimientos:** Expandir y enriquecer la base de conocimientos del sistema [RAG](#), incorporando recursos educativos más diversos y actualizados.
- **Mejora del sistema de corrección:** Implementar técnicas más sofisticadas para la detección y corrección de errores lingüísticos en tiempo real.

#### 7.4.2. Visión a Largo Plazo

- **Sistemas multimodales:** Integrar comprensión y generación multimodal (texto, voz, gestos, expresiones faciales) para una experiencia de aprendizaje más inmersiva y completa.
- **Adaptación a contextos específicos:** Desarrollar versiones especializadas del sistema para contextos educativos específicos, como la enseñanza de idiomas para fines específicos (turismo, negocios, medicina, etc.).
- **Aprendizaje colaborativo:** Explorar la integración de sistemas de aprendizaje colaborativo que fomenten la interacción entre estudiantes y la co-creación de conocimiento.

### 7.5. Reflexiones Finales

El desarrollo de este sistema representa un paso significativo hacia la personalización efectiva del aprendizaje de idiomas mediante tecnologías de [IA](#). La combinación de [RL](#), arquitecturas [Transformers](#) y sistemas [RAG](#) demuestra el potencial de las tecnologías actuales para transformar fundamentalmente el campo educativo.

El valor principal del sistema no reside únicamente en sus capacidades técnicas, sino en su potencial para democratizar el acceso a experiencias de aprendizaje personalizadas y efectivas. La adaptación dinámica a las necesidades individuales permite superar las limitaciones de los enfoques tradicionales, que a menudo fallan en proporcionar el apoyo específico que cada estudiante requiere.

Sin embargo, es importante reconocer que la tecnología, por muy avanzada que sea, representa solo una herramienta al servicio de objetivos pedagógicos más amplios. El sistema

desarrollado no pretende reemplazar a los educadores humanos, sino complementar su labor, proporcionando un entorno de práctica y retroalimentación constante que enriquece la experiencia educativa global.

La verdadera medida del éxito de este proyecto será su capacidad para facilitar el aprendizaje de idiomas de manera más eficiente, inclusiva y motivadora, contribuyendo así a derribar las barreras lingüísticas que separan a las personas y comunidades en un mundo cada vez más interconectado.

Como reflexión final, cabe destacar que el campo de la [IA](#) aplicada a la educación se encuentra en constante evolución, y este trabajo representa solo un punto de partida para desarrollos futuros que continuarán transformando la manera en que aprendemos y enseñamos idiomas.



# Anexo: Faster Whisper y Modelos de Transcripción



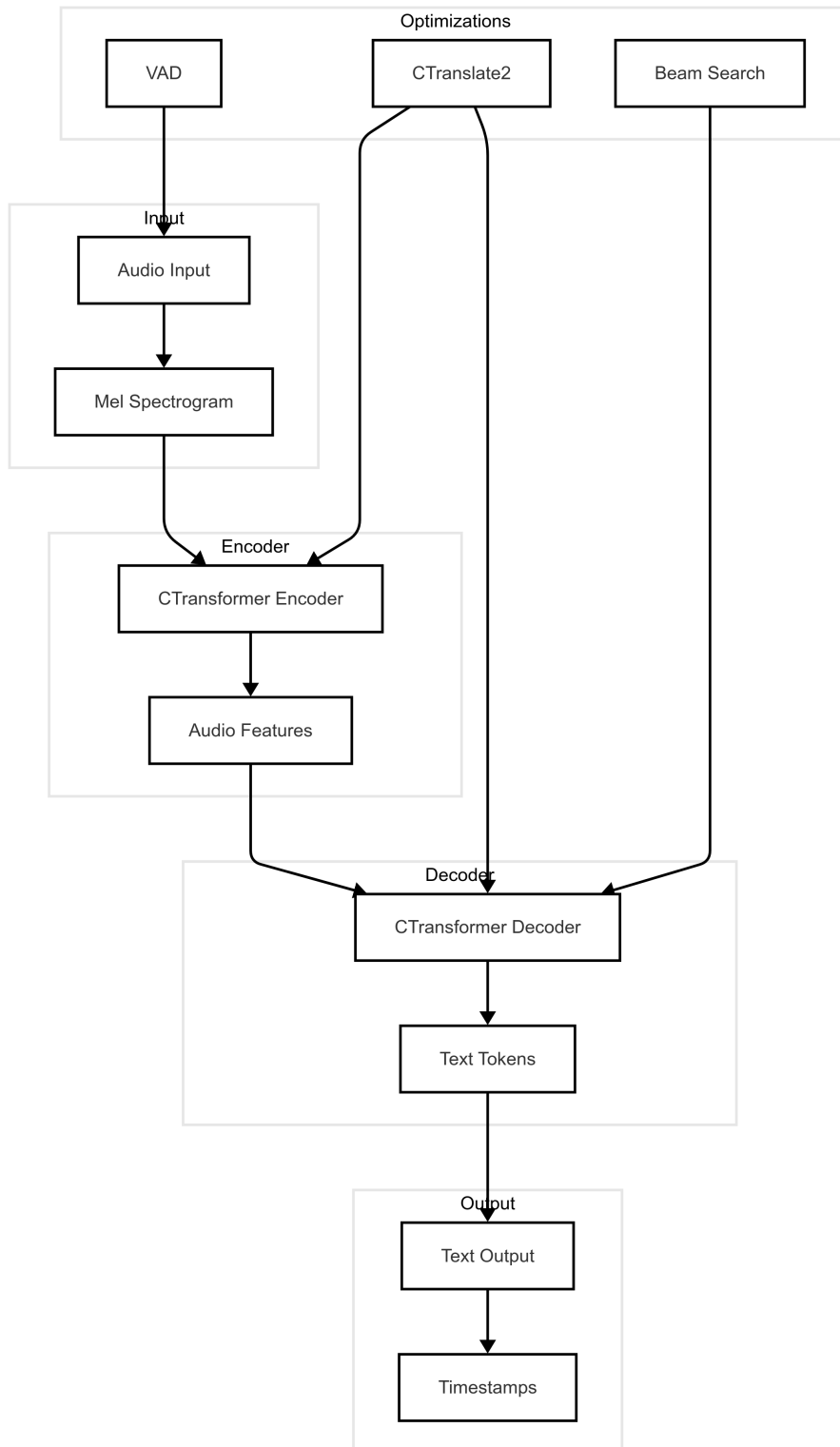
Este anexo explora Faster Whisper, una implementación optimizada del modelo Whisper de OpenAI para transcripción y traducción de voz a texto. Se analizan sus características principales, arquitectura, y se compara el rendimiento entre los diferentes modelos disponibles.

## A.1. Características Principales

Faster Whisper representa una mejora significativa sobre la implementación original de Whisper, destacando por:

- **Optimización CTranslate2:** Utiliza el toolkit CTranslate2 para optimizar la inferencia del modelo.
- **Menor Consumo de Memoria:** Reduce significativamente el uso de memoria mediante técnicas de cuantización.
- **Aceleración por Hardware:** Aprovecha eficientemente CPU y GPU mediante paralelización.
- **Detección de Voz:** Integra VAD (Voice Activity Detection) para mejorar la precisión.

## A.2. Arquitectura del Sistema



**Figura A.1:** *Arquitectura de Faster Whisper*

La arquitectura de Faster Whisper se compone de varios módulos especializados que trabajan en conjunto para proporcionar transcripción eficiente:

### A.2.1. Componentes Principales

#### A.2.1.1. Preprocesamiento de Audio

El sistema procesa la entrada de audio mediante:

$$\text{mel} = \log(\text{máx}(\text{STFT}(x), \epsilon)) \quad (\text{A.1})$$

donde STFT es la Transformada de Fourier de Tiempo Corto y  $\epsilon$  es un valor pequeño para estabilidad numérica.

#### A.2.1.2. CTransformer

La implementación utiliza CTranslate2 para optimizar:

- **Encoder:** Procesa el espectrograma mel en representaciones de audio.
- **Decoder:** Genera tokens de texto mediante atención cruzada.

## A.3. Comparativa de Modelos Whisper

Modelo	Parámetros	RAM (FP16)	WER	Velocidad Relativa
Tiny	39M	1GB	7.1 %	32x
Base	74M	1.5GB	6.1 %	16x
Small	244M	2.5GB	5.2 %	8x
Medium	769M	4.5GB	4.3 %	4x
Large	1550M	7.5GB	3.6 %	1x

**Tabla A.1:** Comparación de modelos Whisper

#### A.3.1. Características por Modelo

- **Tiny:**
  - Ideal para dispositivos con recursos limitados
  - Mejor opción para transcripción en tiempo real
  - Rendimiento aceptable en condiciones de audio limpias
- **Base:**
  - Balance entre rendimiento y recursos

- Adecuado para aplicaciones web
  - Buen rendimiento en múltiples idiomas
- **Small:**
    - Mejora significativa en precisión sobre Base
    - Soporte robusto para múltiples acentos
    - Detección confiable de cambios de idioma
- **Medium:**
    - Alta precisión en condiciones desafiantes
    - Excelente rendimiento en audio con ruido
    - Capacidad avanzada de puntuación
- **Large:**
    - Máxima precisión disponible
    - Mejor rendimiento en audios complejos
    - Capacidad superior de traducción

## A.4. Optimizaciones

### A.4.1. Técnicas de Cuantización

Faster Whisper implementa varias técnicas de cuantización:

- **INT8:** Reduce el tamaño del modelo en 4x con mínima pérdida de precisión
- **INT16:** Balance entre precisión y tamaño
- **FLOAT16:** Máxima precisión con reducción de memoria

### A.4.2. Paralelización

El sistema implementa múltiples niveles de paralelización:

- **Batch Processing:** Procesa múltiples segmentos simultáneamente
- **Thread Pooling:** Optimiza la utilización de CPU
- **GPU Acceleration:** Aprovecha CUDA para procesamiento paralelo

## A.5. Consideraciones de Implementación

### A.5.1. Selección de Modelo

La elección del modelo debe considerar:

- **Recursos Disponibles:** Memoria y capacidad de procesamiento
- **Requisitos de Latencia:** Tiempo de respuesta necesario
- **Precisión Requerida:** Tolerancia a errores

### A.5.2. Estrategias de Deployment

Consideraciones para el despliegue:

- **Edge Computing:** Procesamiento en dispositivo para menor latencia
- **Server-Side:** Mayor capacidad de procesamiento pero mayor latencia
- **Hybrid:** Combinación según necesidades específicas

# Anexo: Kokoro TTS



Este anexo explora Kokoro TTS, un modelo de síntesis de voz de código abierto que destaca por su eficiencia y calidad comparable a modelos más grandes, a pesar de contar con solo 82 millones de parámetros. El modelo implementa una arquitectura ligera basada en StyleTTS 2 e ISTFTNet, diseñada para ofrecer una síntesis de voz de alta calidad con recursos computacionales limitados.

## B.1. Arquitectura del Sistema

La arquitectura de Kokoro TTS se fundamenta en dos componentes principales: StyleTTS 2 e ISTFTNet. Esta combinación permite una síntesis de voz eficiente mientras mantiene una alta calidad en la salida.

### B.1.1. Componentes Principales

- **Misaki G2P**: Sistema de conversión de grafemas a fonemas que soporta múltiples idiomas.
- **Style Encoder**: Codifica las características de estilo de voz a partir de audio de referencia.
- **Decoder**: Genera características acústicas basadas en los fonemas y el estilo.
- **ISTFT Network**: Realiza la síntesis final del audio mediante transformada inversa de Fourier.

## B.2. Características Técnicas

### B.2.1. Especificaciones del Modelo

- **Parámetros**: 82 millones
- **Arquitectura Base**: StyleTTS 2 + ISTFTNet
- **Licencia**: Apache 2.0
- **Formato de Audio**: 24kHz, mono

### B.2.2. Conjunto de Datos

El entrenamiento se realizó exclusivamente con datos de audio permitidos:

- Audio de dominio público
- Audio con licencias permisivas (Apache, MIT)
- Audio sintético de modelos comerciales

## B.3. Análisis de Voces

### B.3.1. Sistema de Calificación

El sistema evalúa las voces mediante dos métricas principales:

- **Calidad Objetivo:**
  - A: Calidad excepcional
  - B: Buena calidad
  - C: Calidad aceptable
  - D: Calidad limitada
- **Duración del Entrenamiento:**
  - HH: 10-100 horas
  - H: 1-10 horas
  - MM: 10-100 minutos
  - M: 1-10 minutos

### B.3.2. Distribución de Voces

Idioma	F	M	Total	Calidad Media
Inglés Americano	11	9	20	B-
Inglés Británico	4	4	8	C+
Japonés	4	1	5	C+
Chino Mandarín	4	4	8	D+
Español	1	2	3	C
Francés	1	0	1	B-
Hindi	2	2	4	C
Italiano	1	1	2	C
Portugués BR	1	2	3	C

**Tabla B.1:** *Distribución y calidad de voces por idioma*

## B.4. Rendimiento y Limitaciones

### B.4.1. Rangos Óptimos de Operación

El rendimiento del modelo varía según la longitud del texto:

- **Rango Óptimo:** 100-200 tokens
- **Rendimiento Reducido:** <20 tokens
- **Posible Aceleración:** >400 tokens

### B.4.2. Costos de Entrenamiento

El entrenamiento de Kokoro ha sido notablemente eficiente:

- **Horas GPU:** 1000 horas en A100 80GB
- **Costo Total:** Aproximadamente \$1000 USD
- **Tasa Promedio:** \$1/hora

## B.5. Comparativa con Otros Modelos

Modelo	Parámetros	Voces	Idiomas	Licencia
Kokoro	82M	54	8	Apache
Coqui	1000M	1087	100+	MIT
Bark	900M	100+	100+	MIT

**Tabla B.2:** Comparación con modelos TTS similares



# Bibliografía

- Abdin, M., Aneja, J., Behl, H., Bubeck, S., Eldan, R., Gunasekar, S., Harrison, M., Hewett, R. J., Javaheripi, M., Kauffmann, P., Lee, J. R., Lee, Y. T., Li, Y., Liu, W., Mendes, C. C. T., Nguyen, A., Price, E., de Rosa, G., Saarikivi, O., Salim, A., Shah, S., Wang, X., Ward, R., Wu, Y., Yu, D., Zhang, C., y Zhang, Y. (2024). Phi-4 technical report.
- Anderson, J. R. y Boyle, C. F. (2020). Adaptive learning systems in modern education. *Journal of Computer Assisted Learning*.
- Baker, R. S. y Inventado, P. S. (2014). Educational data mining and learning analytics. *Learning Analytics*.
- Brown, T. et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*.
- Coyle, D., Hood, P., y Marsh, D. (2010). *CLIL: Content and language integrated learning*. Cambridge University Press.
- Ellis, R. (1994). The study of second language acquisition. *Oxford University Press*.
- Fries, C. C. (1945). *Teaching and learning English as a foreign language*. University of Michigan Press.
- Gouin, F. (1892). *The art of teaching and studying languages*. Heath, D.C.
- Graves, A., Mohamed, A.-r., y Hinton, G. (2013). Speech recognition with deep recurrent neural networks. *IEEE International Conference on Acoustics, Speech and Signal Processing*.
- Hexgrad (2025). Kokoro-82m (revision d8b4fc7).
- Hymes, D. (1972). *On communicative competence*. University of Pennsylvania Press.
- Krashen, S. D. (1982). *Principles and practice in second language acquisition*. Pergamon.
- Lewis, P. et al. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*.
- Liu, J. et al. (2023). Modelos de lenguaje son aprendices de pocas muestras. *Advances in neural information processing systems*.
- Nunan, D. (1989). *Designing tasks for the communicative classroom*. Cambridge University Press.
- Nussbaum, Z., Morris, J. X., Duderstadt, B., y Mulyar, A. (2024). Nomic embed: Training a reproducible long context text embedder.
- Peng, Y., Tian, J., Chen, W., Arora, S., Yan, B., Sudo, Y., Shakeel, M., Choi, K., Shi, J., Chang, X., weon Jung, J., y Watanabe, S. (2024). Owsn v3.1: Better and faster open whisper-style speech models based on e-branchformer.
- Richards, J. C. y Rodgers, T. S. (2000). *Approaches and methods in language teaching*. Cambridge University Press.
- Rodríguez, J. et al. (2023). Aprendizaje de lenguas con modelos de lenguaje de gran escala: desafíos y oportunidades. *Computer Assisted Language Learning*.
- Roll, I. y Wylie, R. (2018). Learning analytics and ai: Politics, pedagogy and practices. *British Journal of Educational Technology*.
- Schulman, J. et al. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Taylor, P. (2009). Text-to-speech synthesis. *Cambridge university press*.
- VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*.
- Vaswani, A. et al. (2017). Attention is all you need. *Advances in neural information processing systems*.
- Williams, R. y Chen, D. (2017). The use of reinforcement learning algorithms in adaptive education. *Journal of Educational AI*.