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

Índice de tablas

Índice de algoritmos

Glosario

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

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.

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

El sistema propuesto en este Trabajo de Fin de Máster integra Reinforcement Learning (RL), arquitecturas Transformers y un sistema de Retrieval-Augmented Generation (RAG) para optimizar el aprendizaje de idiomas. El sistema recomienda ejercicios personalizados y genera diálogos interactivos con retroalimentación en tiempo real. Además, genera rutas de aprendizaje adaptativas y simulaciones de conversación, integrando tecnologías de Text-to-Speech (TTS) y Speech-to-Text (STT) para el desarrollo de habilidades auditivas y de pronunciación. El sistema evoluciona continuamente mediante el análisis de datos de interacción, registros de audio y diálogos simulados, adaptándose a las necesidades específicas de cada estudiante.

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.

1.1. Motivación

La adquisición de una segunda lengua es un proceso complejo que varía significativamente entre individuos. 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.

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 adaptan 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

A pesar de los avances en la enseñanza de idiomas, existen varias áreas donde se pueden realizar mejoras significativas:

- **Adaptabilidad Dinámica**: Implementar sistemas que ajusten el contenido y la dificultad en tiempo real, basándose en el rendimiento y las necesidades del estudiante.
- **Personalización Profunda**: Considerar múltiples factores individuales, como el estilo de aprendizaje, intereses y ritmo de progreso, para optimizar el proceso de aprendizaje.
- **Interacción Natural**: Utilizar tecnologías avanzadas, como modelos de lenguaje natural y procesamiento de voz, para simular conversaciones más realistas y dinámicas.
- **Feedback Contextual**: Proporcionar retroalimentación detallada y específica, basada en el contexto y el perfil del estudiante, para mejorar la comprensión y el rendimiento.

1.2. Objetivos

1.2.1. Objetivo General

Desarrollar un sistema de aprendizaje de idiomas que utilice RL, Transformers y un Sistema Multi-Agente para una experiencia de aprendizaje personalizada, adaptativa y efectiva.

1.2.2. Objetivo Específicos

Para alcanzar el objetivo general, se han definido varios objetivos específicos que se centran en la implementación de técnicas avanzadas de inteligencia artificial. Estos objetivos específicos se enfocan en la aplicación de RL para optimizar el proceso de aprendizaje, el uso de modelos Large Language Model (LLM) para mejorar la interacción, el desarrollo de herramientas basadas en TTS y STT para perfeccionar las habilidades lingüísticas, y la integración de sistemas RAG para una gestión eficiente del conocimiento. A continuación, se detallan los objetivos específicos:

1.2.2.1. Optimización del Aprendizaje

El objetivo de esta sección es implementar estrategias que optimicen el proceso de aprendizaje mediante la personalización y la evaluación continua. Para ello, se propone implementar un sistema de RL que optimice rutas de aprendizaje personalizadas, desarrollar mecanismos de adaptación dinámica del contenido y crear sistemas de evaluación continua del progreso.

1.2.2.2. Mejora de la Interacción

El objetivo de esta sección es mejorar la interacción entre el sistema y el usuario mediante el uso de LLM. Para ello, se propone integrar modelos LLM para el Natural Language Processing (NLP), desarrollar sistemas de diálogo contextuales e implementar análisis de errores en tiempo real. Estas mejoras permitirán una comunicación más fluida y natural, facilitando una experiencia de aprendizaje más efectiva y personalizada.

1.2.2.3. Perfeccionamiento de Habilidades Lingüísticas

El objetivo aquí es desarrollar herramientas que ayuden a los usuarios a mejorar sus habilidades lingüísticas, especialmente en pronunciación y comprensión. Para lograr esto, se propone crear sistemas de evaluación de pronunciación usando TTS y STT, desarrollar ejercicios adaptativos de comprensión e implementar práctica conversacional contextual.

1.2.2.4. Gestión del Conocimiento

Esta sección se dedica a la integración y gestión de recursos educativos para proporcionar un acceso eficiente y actualizado a la información. Se busca integrar sistemas RAG para el acceso a recursos educativos, desarrollar bases de conocimiento dinámicas e implementar mecanismos de actualización de contenido.

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. Se exploran los sistemas de aprendizaje adaptativo, destacando innovaciones recientes en plataformas populares. Además, se analizan las aplicaciones LLM en la generación de diálogos y corrección de textos. También se presentan tecnologías emergentes con compañeros de aprendizaje basados en IA, avances en procesamiento de voz (TTS y STT), y la implementación de tecnologías de Sistema Multi-Agente. Finalmente, se revisan los frameworks de RL más utilizados en la industria.

2.1. Sistemas de Aprendizaje Adaptativo

Los sistemas modernos de aprendizaje de idiomas han evolucionado significativamente con la integración de IA y aprendizaje automático. A continuación, se destacan algunas innovaciones recientes:

- **Busuu Conversations (2024)**[1]: Incorpora un sistema de IA que analiza patrones de error y ajusta dinámicamente el contenido para mejorar la eficacia del aprendizaje.
- **Duolingo Max (2024)**[2]: Utiliza GPT-4 para generar explicaciones personalizadas y mantener conversaciones contextuales, adaptándose al nivel del usuario.
- **Babbel Everyday Conversations (2023)**[3]: Combina IA con tutores humanos para optimizar la experiencia de aprendizaje híbrido, ofreciendo una interacción más personalizada.
- **Lingvist (2023)**[4]: utiliza datos contextuales para generar ejercicios, lecciones y recomendaciones adaptadas, facilitando la recuperación de contenidos lingüísticos relevantes y la generación de actividades interactivas.
- **Elsa Speak (2023)**[5]: Sistema de pronunciación asistida por IA que proporciona retroalimentación en tiempo real y ejercicios personalizados para mejorar la fluidez y la

[1] https://www.busuu.com
[2] https://www.duolingo.com
[3] https://www.babbel.com
[4] https://www.lingvist.com
[5] https://www.elsaspeak.com

precisión en la pronunciación.

2.2. Aplicaciones de LLM

Los LLM han revolucionado el aprendizaje de idiomas en múltiples aspectos, proporcionando herramientas avanzadas para la generación de diálogos y el análisis y corrección de textos.

2.2.1. Asistentes y Diálogo

- **ChatGPT (2022)**[6]: Revolucionó la interacción humano-IA estableciendo el estándar de interfaces conversacionales naturales y creando un ecosistema completo de desarrollo.
- **Claude (2023)**[7]: 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)**[8]: Ofrece herramientas de análisis lingüístico y generación de contenido educativo, mejorando la calidad del aprendizaje.
- **LLaMA (2023)**[9]: 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

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

2.3. Tecnologías Emergentes con Compañeros de Aprendizaje

Los compañeros de aprendizaje basados en IA están emergiendo como una herramienta fundamental en la educación moderna. Estos asistentes virtuales proporcionan apoyo personalizado y adaptativo, actuando como tutores disponibles 24/7. A continuación, se presentan algunas implementaciones destacadas:

[6] https://chatgpt.com/
[7] https://claude.ai/
[8] https://language.cognitive.azure.com/
[9] https://ai.facebook.com/blog/large-language-model-llama
[10] https://www.grammarly.com
[11] https://www.deepl.com/write

- **Khanmigo (2024)**[12]: 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)**[13]: 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)**[14]: Sistema de tutoría adaptativa para predecir el rendimiento del estudiante y personalizar el contenido, maximizando la eficiencia del aprendizaje mediante análisis predictivo.

2.4. Avances en Procesamiento de Voz

Las tecnologías de TTS y STT han mejorado en naturalidad y expresividad, proporcionando voces más humanas y adaptativas, así como una transcripción precisa y rápida. A continuación, se presentan algunas de las tecnologías más destacadas en este campo:

- **Whisper OpenAI (2022)**[15]: 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)**[16]: 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)**[17]: 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)**[18]: 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)**[19]: 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.

[12] https://www.khanacademy.org/khan-labs
[13] https://thirdspacelearning.com
[14] https://riiid.com
[15] https://openai.com/research/whisper
[16] https://cloud.google.com/speech-to-text
[17] https://azure.microsoft.com/en-us/products/ai-services/ai-speech
[18] https://deepgram.com
[19] https://huggingface.co/hexgrad/Kokoro-82M

2.5. Agentic AI

La tecnología de Sistema Multi-Agente se esta 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.

- **LangChain (2022)**[20]: 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)**[21]: 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)**[22]: 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)**[23]: 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.

2.6. Frameworks de Aprendizaje por Refuerzo

El RL ha ganado popularidad en la industria debido a su capacidad para resolver problemas complejos mediante la optimización de políticas a través de la interacción con el entorno. A continuación, se presentan algunos de los frameworks de RL más utilizados en la industria, todos ellos Código Abierto:

- **TensorFlow Agents (2019)**[24]: 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)**[25]: 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.

[20] https://www.langchain.com
[21] https://www.crewai.com
[22] https://www.phidata.com
[23] https://www.microsoft.com/en-us/research/project/autogen
[24] https://www.tensorflow.org/agents
[25] https://stable-baselines3.readthedocs.io

- **TorchRL (2022)**[26]: 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.

2.7. Aplicación de las Tecnologías en el Trabajo

En este trabajo, se toman en cuenta las siguientes tecnologías y teorías para desarrollar un sistema de aprendizaje de idiomas asistido por IA:

- **Sistemas de Aprendizaje Adaptativo**: Se implementa un sistema que analiza los patrones de error de los usuarios y ajusta dinámicamente el contenido.
- **Aplicaciones de LLM**: Se utiliza un LLM para generar diálogos y proporcionar correcciones contextuales.
- **Tecnologías Emergentes con Compañeros de Aprendizaje**: Se desarrolla un asistente virtual que actúa como compañero de aprendizaje, proporcionando apoyo personalizado y adaptativo.
- **Avances en Procesamiento de Voz**: Se integra tecnología de TTS y STT para mejorar la interacción del usuario con el sistema.
- **Agentic AI**: Se explora la creación de agentes autónomos que interactúan entre sí y con los usuarios para proporcionar experiencias de aprendizaje más dinámicas.
- **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.

Estas tecnologías y teorías son fundamentales para el desarrollo de un sistema de aprendizaje de idiomas que es adaptativo, interactivo y altamente personalizado, mejorando significativamente la experiencia del usuario.

[26] https://github.com/pytorch/rl

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

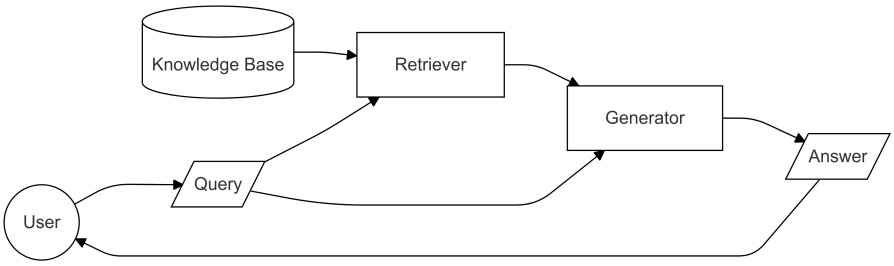


Figura 3.1: *Flujo de información en un sistema RAG*

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:

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

Donde:

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

- ϵ es el parámetro de clipping (típicamente 0.2)

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

1. Inicializar los parámetros de la política θ y el valor función ϕ
2. Para cada iteración:
   a) Recopilar conjunto de trayectorias $\mathcal{D}_k = \{\tau_i\}$ ejecutando la política π_θ en el entorno
   b) Calcular ventajas estimadas $\hat{A}_t$ usando función de valor actual V_ϕ
   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:

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

      3) Actualizar θ minimizando $-L^{CLIP}(\theta)$ usando descenso de gradiente
      4) Actualizar función de valor ϕ minimizando error cuadrático medio
   d) Actualizar $\theta_{old} \leftarrow \theta$
3. Devolver la política optimizada π_θ

3.4.2.2. Aplicación en el Sistema

En nuestro contexto educativo:

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

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

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

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

- **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 (π):** Determina qué acción tomar en cada estado. Por ejemplo, si el estudiante muestra consistentemente errores en gramática, π 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:

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

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

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
  - Arquitectura Ada Lovelace
  - 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**: 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**: Sistema de síntesis de voz que ofrece:
  - Generación de voz natural y expresiva
  - Múltiples voces y estilos
  - Alta eficiencia en el procesamiento

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.

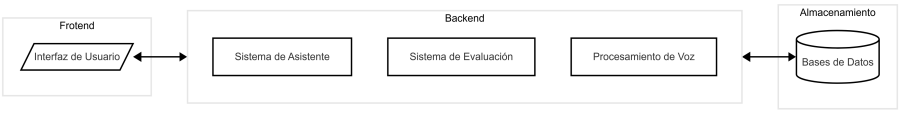
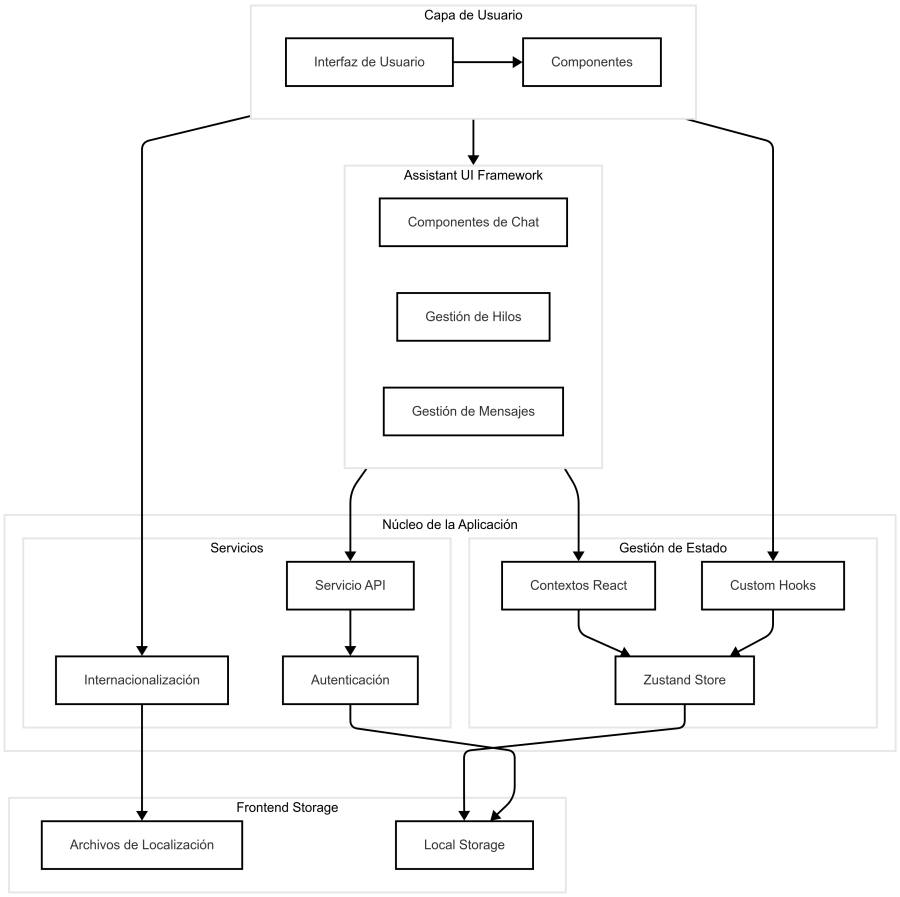


Figura 5.1: *Arquitectura Simplificada del Sistema*

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

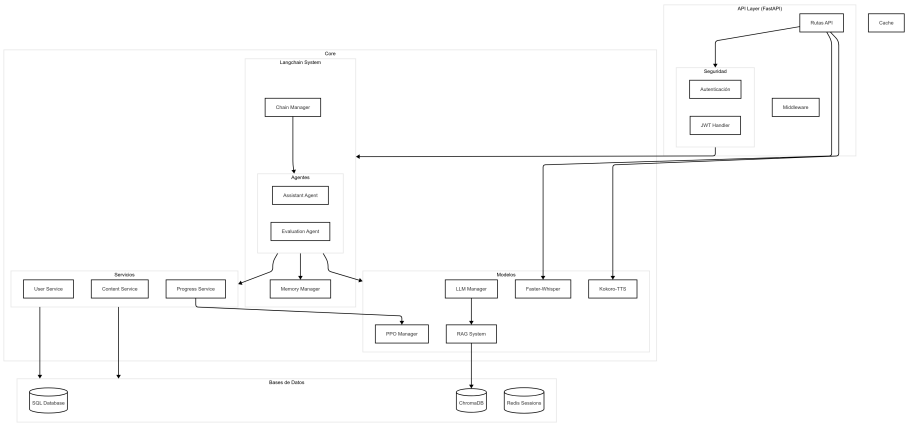
La comunicación con el backend se gestiona a través de servicios especializados:

- **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 Proximal Policy Optimization (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

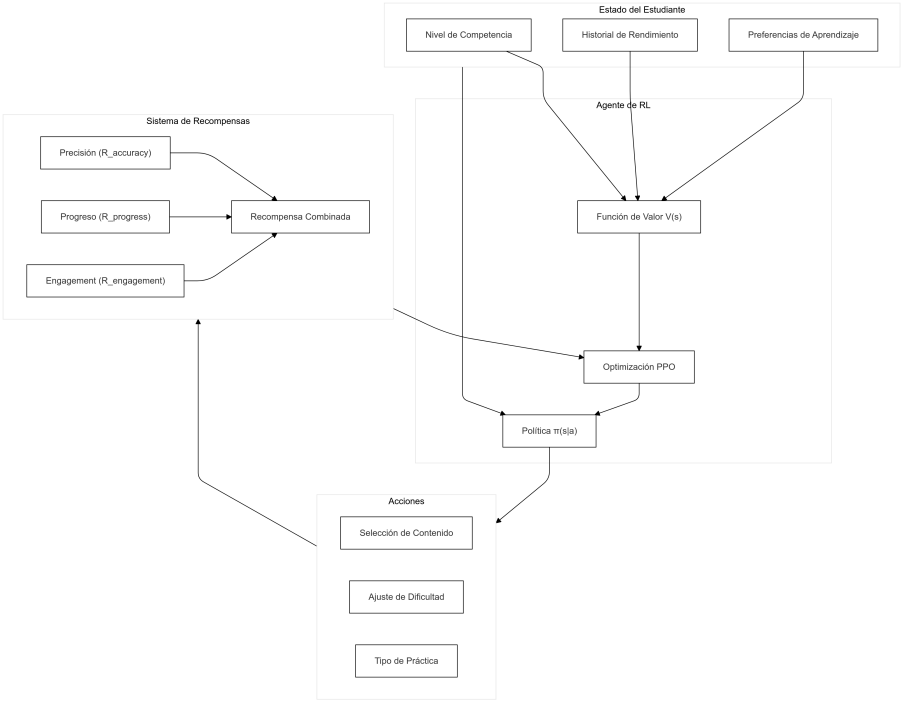
La síntesis de voz se realiza mediante Kokoro-TTS, un sistema avanzado de text-to-speech. Sus componentes principales son:

- **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. Algoritmos Desarrollados

Esta sección detalla los algoritmos principales desarrollados para la personalización del aprendizaje, incluyendo el sistema de RL y el mecanismo de recompensas.

Figura 5.4: *Flujo del Algoritmo de RL y Sistema de Recompensas*



5.3.1. Algoritmo de Personalización

El sistema implementa un algoritmo de RL basado en PPO Schulman et al. (2017) para optimizar las rutas de aprendizaje. El objetivo es maximizar el aprendizaje a largo plazo mientras se mantiene un nivel apropiado de desafío y engagement.

5.3.1.1. Formulación del Problema

El problema se formula como un MDP donde:

- **Estado (s_t):** Vector que representa el estado actual del estudiante:

$$s_t = [c_1, ..., c_n, h_1, ..., h_m, p_1, ..., p_k] \tag{5.1}$$

  donde c_i son los niveles de competencia en diferentes habilidades, h_i es el historial de rendimiento, y p_i son las preferencias de aprendizaje.

- **Acciones (a_t):** Vector de decisiones pedagógicas:

$$a_t = [d, t, c] \tag{5.2}$$

  donde d es el nivel de dificultad, t es el tipo de ejercicio, y c es el contenido específico.

- **Política (π_θ):** La política que mapea estados a acciones:

$$\pi_\theta(a_t|s_t) = P(a_t|s_t; \theta) \tag{5.3}$$

5.3.1.2. Algoritmo PPO

El algoritmo PPO optimiza la política mediante la siguiente función objetivo:

$$L^{CLIP}(\theta) = \mathbb{E}_t[\text{mín}(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t)] \tag{5.4}$$

donde:

- $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ es el ratio de probabilidades
- A_t es la estimación de ventaja
- ϵ es el parámetro de recorte (típicamente 0.2)

La actualización de la política se realiza mediante descenso de gradiente:

$$\theta_{new} = \theta + \alpha \nabla_\theta L^{CLIP}(\theta) \tag{5.5}$$

5.3.2. Sistema de Recompensas

Se implementa un sistema de recompensas multiobjetivo que considera tres componentes principales:

$$R = w_1 R_{accuracy} + w_2 R_{progress} + w_3 R_{engagement} \tag{5.6}$$

5.3.2.1. Componentes de Recompensa

- **Precisión ($R_{accuracy}$):** Evalúa la corrección de las respuestas:

$$R_{accuracy} = \frac{\text{respuestas\_correctas}}{\text{total\_respuestas}} \cdot \gamma \tag{5.7}$$

  donde γ es un factor de dificultad que aumenta la recompensa para ejercicios más desafiantes.

- **Progreso ($R_{progress}$):** Mide el avance en el dominio de habilidades:

$$R_{progress} = \sum_{i=1}^{n} \Delta c_i \cdot \beta_i \tag{5.8}$$

  donde Δc_i es el cambio en el nivel de competencia de la habilidad i, y β_i es su peso relativo.

- **Engagement ($R_{engagement}$):** Evalúa la participación activa:

$$R_{engagement} = \alpha_t t_{session} + \alpha_c c_{completion} + \alpha_i i_{interaction} \tag{5.9}$$

  donde $t_{session}$ es la duración de la sesión, $c_{completion}$ es la tasa de finalización, y $i_{interaction}$ es la frecuencia de interacción.

5.3.2.2. Adaptación de Pesos

Los pesos w_i se ajustan dinámicamente según el perfil del estudiante mediante un algoritmo de adaptación:

$$w_i^{new} = w_i + \eta(\bar{R}_i - R_{target}) + \lambda \Delta w_i \tag{5.10}$$

donde:

- η es la tasa de adaptación
- $\bar{R}_i$ es la recompensa promedio reciente para el componente i
- R_{target} es el valor objetivo
- $\lambda \Delta w_i$ es un término de momentum para estabilizar los cambios

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 y las pruebas preliminares realizadas. Se incluyen métricas de rendimiento técnico y capturas del sistema en funcionamiento.

6.1. Evaluación del Sistema

La evaluación del sistema se ha realizado a través de pruebas preliminares enfocadas en el rendimiento técnico y la funcionalidad básica.

6.1.1. Rendimiento Técnico

6.1.1.1. Frontend

El sistema muestra los siguientes resultados en pruebas internas:

- **Procesamiento TTS:**
  - Latencia de generación: 50ms por frase
  - Uso de memoria: 120MB promedio
  - Utilización de GPU: 20-25 % durante la generación
- **Procesamiento STT:**
  - Latencia de reconocimiento: 100ms
  - Uso de memoria: 150MB promedio
  - Precisión inicial: 85 % en condiciones controladas

6.1.1.2. Backend

Las pruebas preliminares del sistema RAG muestran:

- **Sistema RAG:**
  - Latencia de búsqueda: 75ms
  - Precisión inicial: 82 %

- Relevancia contextual: 80 %

- **Sistema PPO:**
  - Tiempo de convergencia: 15 episodios promedio
  - Estabilidad del modelo: 90 % en pruebas sintéticas

6.2. Capturas del Sistema

Esta sección presenta las principales interfaces y componentes del sistema implementado.

6.2.1. Interfaz Principal

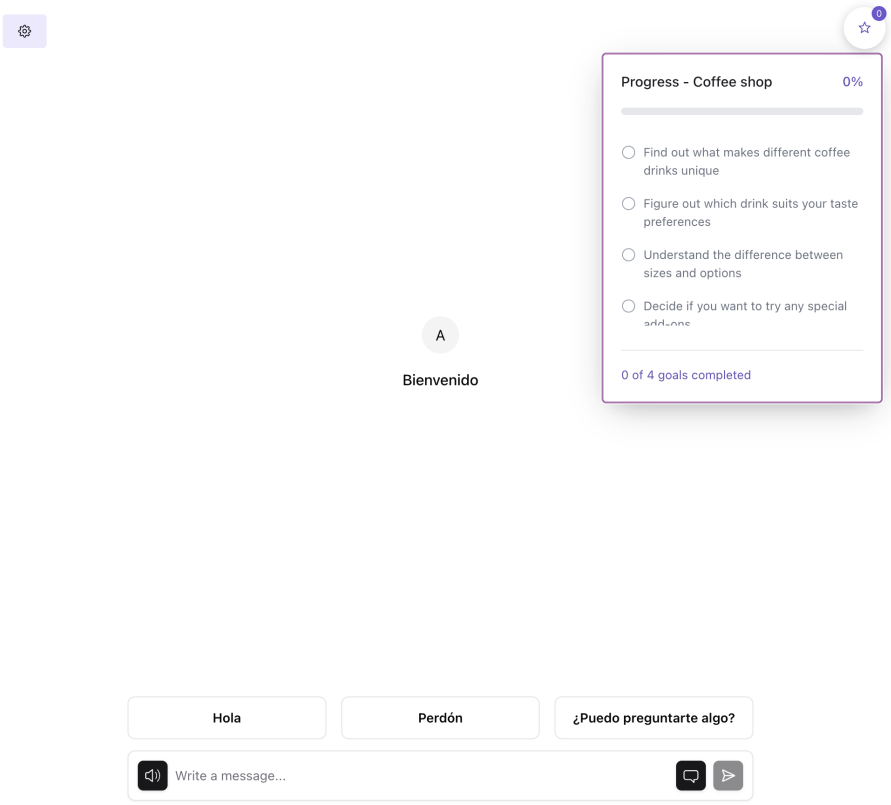


Figura 6.1: *Interfaz principal del sistema mostrando el chat y las opciones de voz*

La Figura 6.1 muestra la interfaz principal del sistema, donde se pueden observar:

- Panel de chat interactivo
- Controles de voz para TTS y STT
- Indicadores de nivel y progreso

6.2.2. Sistema de Diálogo

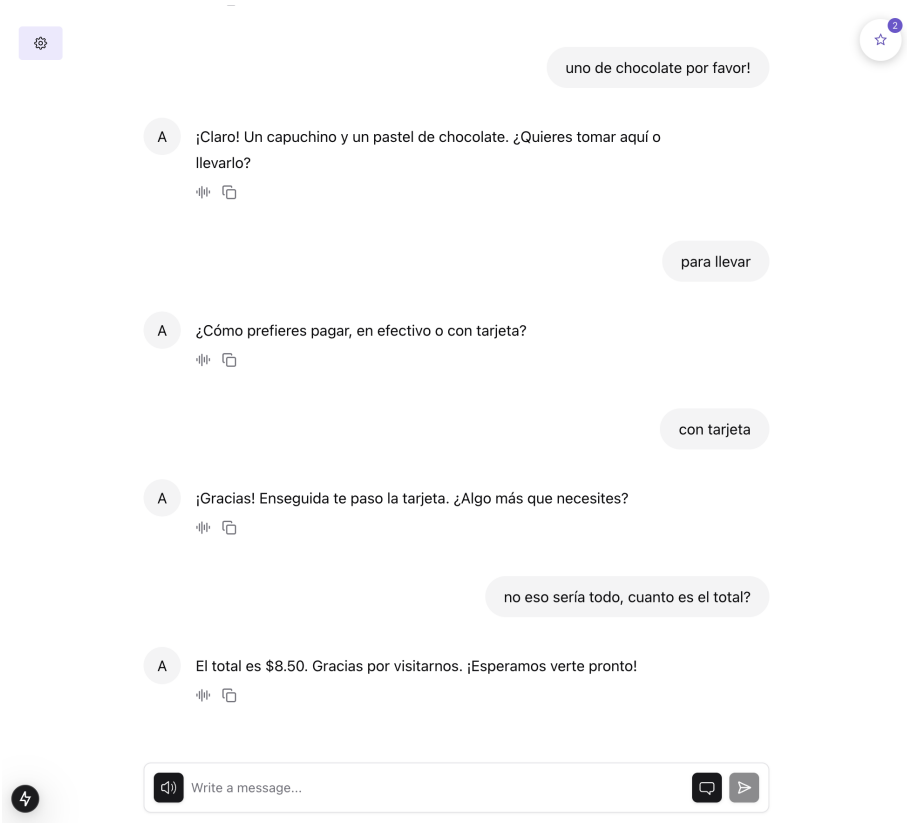


Figura 6.2: *Sistema de diálogo mostrando una conversación de ejemplo*

La Figura 6.2 ilustra el sistema de diálogo en acción, destacando:

- Generación de respuestas contextuales
- Integración de RAG para respuestas precisas
- Sistema de corrección en tiempo real

6.2.3. Selector de Situaciones

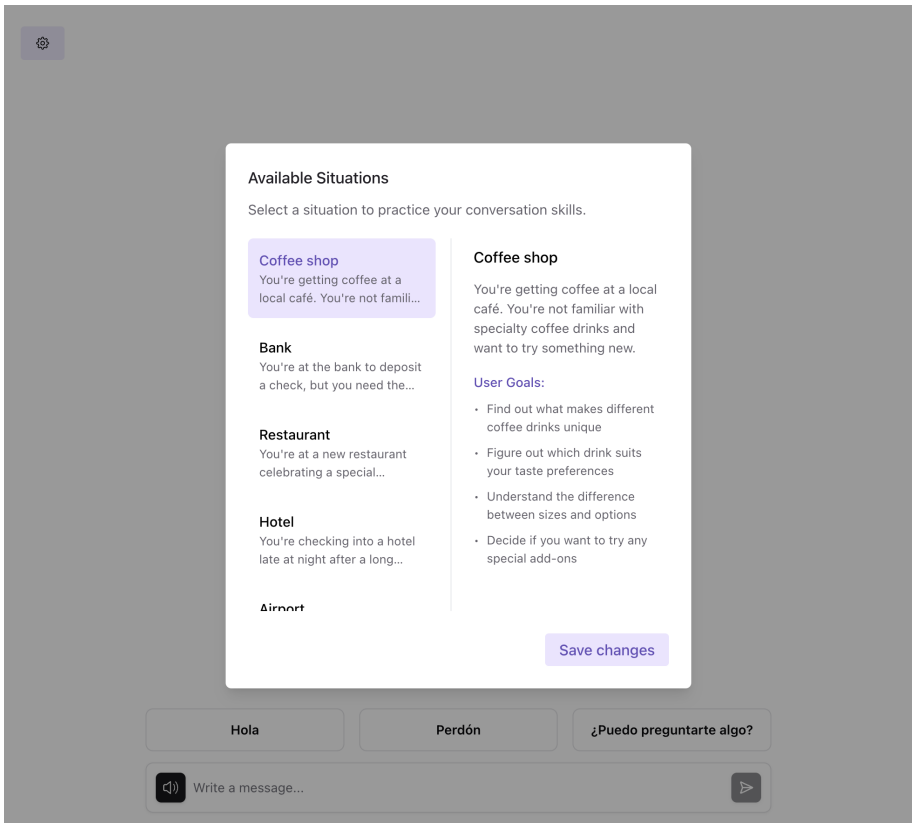


Figura 6.3: *Interfaz de selección de contextos conversacionales y objetivos*

La Figura 6.3 muestra la interfaz de selección de situaciones, donde los usuarios pueden elegir el contexto específico para su práctica conversacional. El sistema ofrece:

- **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 a alcanzar durante la conversación
  - Indicadores de progreso para cada objetivo
  - Retroalimentación en tiempo real sobre el avance
- **Personalización:**
  - Adaptación del nivel de dificultad según el perfil del usuario

- Recomendaciones basadas en el historial de práctica
- Opciones para personalizar los objetivos específicos

6.2.4. Panel de Análisis

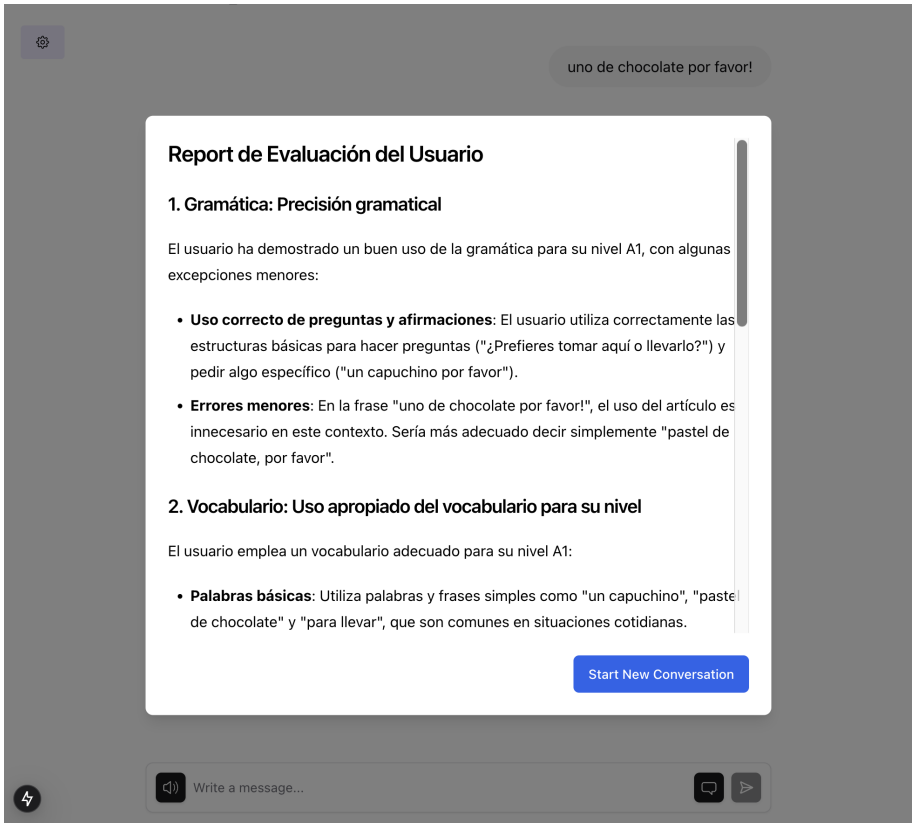


Figura 6.4: *Panel de análisis mostrando métricas de aprendizaje*

La Figura 6.4 muestra el panel de análisis, que incluye:

- Métricas de progreso
- Análisis de errores comunes
- Recomendaciones personalizadas

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, con el código fuente disponible públicamente en GitHub bajo licencia MIT.

6.4.1. Estructura de Repositorios

- **Frontend - Cliente:**
  - Repositorio: https://github.com/EmaSuriano/language-learning-client
  - Tecnologías: Next.js, TypeScript, Tailwind CSS
  - Componentes principales:
    - Interfaz de chat basada en Assistant UI
    - Selector de situaciones y objetivos
    - Gestión de estado con Zustand
    - Multilenguaje con i18n

- **Backend - Servidor:**
  - Repositorio: https://github.com/EmaSuriano/language-learning-server
  - Tecnologías: FastAPI, Python, LangChain
  - Componentes principales:
    - Sistema RAG para recuperación de contexto
    - Integración con LLM
    - API REST para comunicación con el cliente
    - Sistema de procesamiento de voz con Faster-Whisper y Kokoro-TTS

6.4.2. Documentación

Ambos repositorios incluyen:

- README con instrucciones detalladas de instalación y configuración
- Documentación de endpoints y componentes principales

- Variables de entorno requeridas
- Ejemplos de uso

6.5. Limitaciones y Trabajo Futuro

Se identifican las siguientes áreas para desarrollo futuro:

- **Evaluación Exhaustiva:**
  - Pruebas con una muestra más amplia de usuarios
  - Evaluación longitudinal del progreso
  - Análisis comparativo con otros sistemas
- **Mejoras Técnicas:**
  - Optimización del modelo PPO
  - Mejora en la precisión del STT
  - Ampliación de la base de conocimientos del RAG
- **Funcionalidades Adicionales:**
  - Implementación de más escenarios de práctica
  - Expansión del sistema de evaluación
  - Mejoras en la interfaz de usuario

Conclusiones

7

Anexo: Faster Whisper y Modelos de Transcripción

A

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.

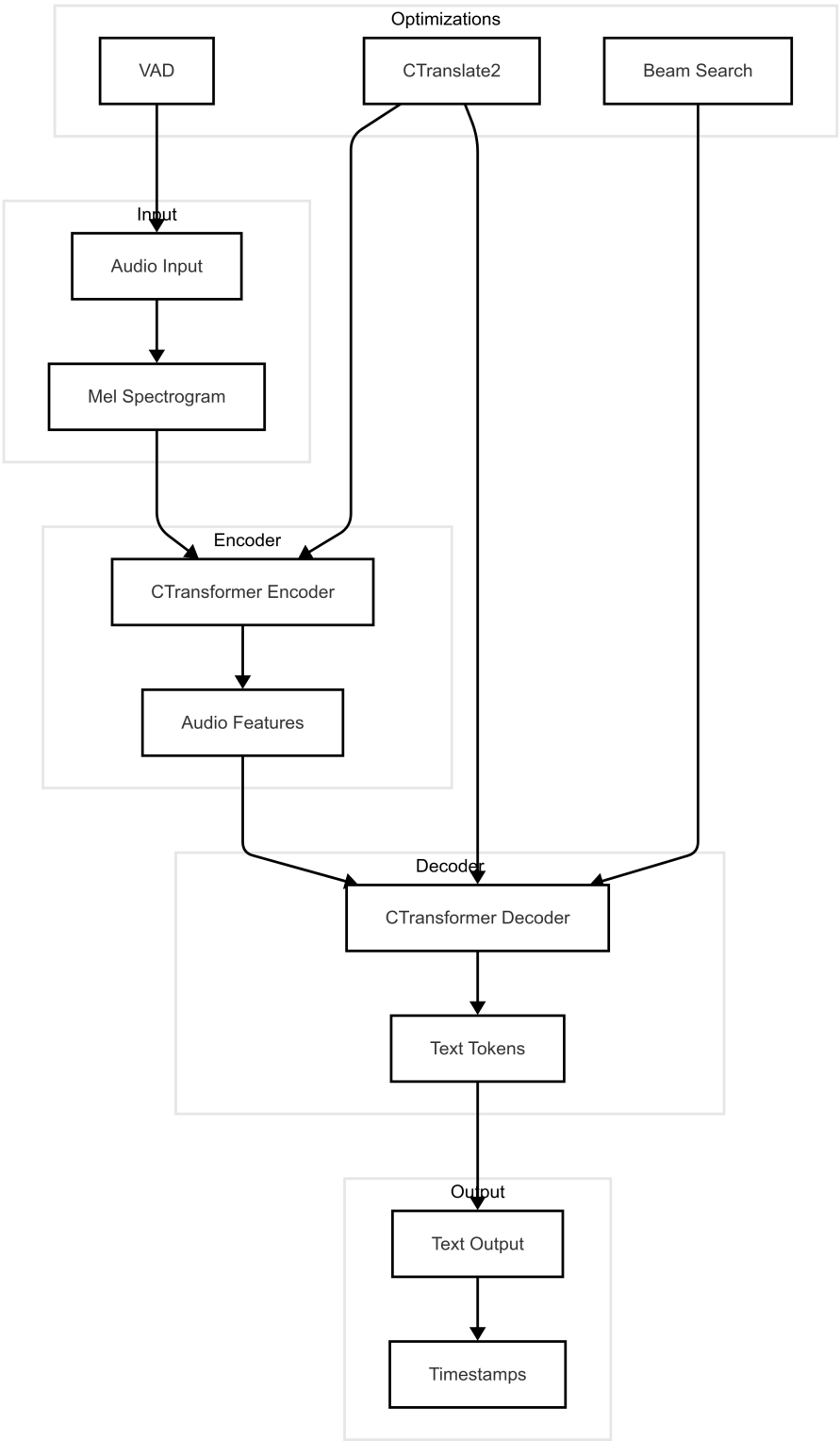
A.1. Características Principales

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:

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

donde STFT es la Transformada de Fourier de Tiempo Corto y ϵ 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

Tabla A.1: *Comparación 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

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

B

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

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

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

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

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

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

Bibliografía

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*.

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*.

Nunan, D. (1989). *Designing tasks for the communicative classroom*. Cambridge University Press.

Richards, J. C. y Rodgers, T. S. (2000). *Approaches and methods in language teaching*. Cambridge University Press.

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*.