

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE HÁBITAT, INFRAESTRUCTURA Y

CREATIVIDAD

CARRERA INGENIERÍA EN SISTEMAS DE LA INFORMACIÓN



TRABAJO DE TITULACIÓN

Tema: DESARROLLO DEL PROTOTIPO DE UN TUTOR UTILIZANDO
UN MODELO DE IA PARA EL APRENDIZAJE DEL IDIOMA INGLÉS
EN EL NIVEL A1.

CASO DE ESTUDIO: ACADEMIA “ENGLISH VALLEY”.

AUTOR:

JUAN NICOLÁS ROJAS MARTÍNEZ

DIRECTOR:

JORGE ALEJANDRO ALARCÓN MENA

QUITO, SEPTIEMBRE DE 2026

DEDICATORIA

Este trabajo de titulación está dedicado a:

En primer lugar, a Dios, a su mano y a su guía constante que he sentido a lo largo de toda mi vida. Su enseñanza ha sido mi fortaleza en cada paso, recordándome que todo lo que emprendo debo hacerlo con entrega y excelencia como para ÉL.

A mi madre, por su presencia dulce y su ejemplo inalcanzable de constancia y superación ante cualquier adversidad. Gracias por instruirme e inspirarme cada día a construir la mejor versión de mí mismo.

A mi familia, especialmente a mis abuelos, tíos y primos. Gracias por enseñarme que la familia se construye con amor y presencia, mucho más allá de los lazos sanguíneos, y por brindarme su apoyo único e incondicional en cada etapa de este proceso.

AGRADECIMIENTO

A mi padre, por haber permanecido pendiente de mi crecimiento y formación a pesar de las dificultades. Le agradezco profundamente sus enseñanzas y el haberme permitido ser autodidacta, dándome las herramientas necesarias para siempre buscar una solución.

A mis "Colegas", por recorrer a mi lado en esta etapa y por cada momento compartido. Gracias por demostrarme que es posible superar la exigencia de la carrera y generar lazos de amistad; de cada uno de ustedes me llevo una enseñanza valiosa para la vida.

A mis profesores, agradecerles por su exigencia, por desafiarnos a alcanzar la excelencia y por compartir también sus consejos profesionales y personales.

Finalmente, a cada persona que cruzó mi camino durante estos años. Miro hacia atrás y agradezco cada palabra de aliento, cada gesto y cada momento compartido.

Estoy convencido de que nadie llega por casualidad; de todos aprendí algo, y su presencia fue el tejido que hizo posible este sueño.

ABSTRACT

The present project proposes the development of a English Language learning tutor prototype based on Artificial Intelligence (AI) for A1 level, applied to English Valley academy. The main objective is to design and implement a prototype system with the main feature of generating interactive exercises, providing immediate feedback, and adapting to each student's progress according to the descriptors established by the Common European Framework of Reference for Languages (CEFR). The prototype will use Large Language Models (LLM) to simulate learning contexts by each skill that fosters autonomous and continuous practice.

The project adopts an applied research approach and is developed under the Agile Scrum methodology, which allows iterative progress, validation of functional components, and continuous improvement through user feedback. This methodological framework ensures that the system evolves in accordance with the pedagogical objectives of the academy and the learning needs of its students.

The theoretical foundation highlights the transformative role of AI in education, emphasizing its capacity to personalize learning, promote self-regulation, and reduce linguistic anxiety through interactive tools such as chatbots and intelligent tutors. Additionally, it underscores the importance of maintaining the pedagogical coherence between technological innovation and also the educational objectives of the academy in order to ensure meaningful learning outcomes and not only interesting tools but without a methodological background.

The resulting prototype represents an innovative, accessible and scalable solution that does not replace but complements the active learning methodology implemented at English Valley. The integration of AI-driven feedback and adaptive learning strategies, results in the proposed system contributing to enhancing language practice, improving learner engagement, and strengthening the overall quality of English instruction at the foundational level.

Keywords: Artificial Intelligence, Language Learning, Intelligent Tutor, Natural Language Processing, CEFR, A1 Level, Agile Methodology, Autonomous Learning.

TABLA DE CONTENIDOS

CAPÍTULO I: INTRODUCCIÓN.....	1
1.1 JUSTIFICACIÓN.....	1
1.2 PLANTEAMIENTO DEL PROBLEMA.....	5
1.3 OBJETIVOS.....	8
1.3.1 OBJETIVO GENERAL	8
1.3.2 OBJETIVOS ESPECÍFICOS	8
1.4 METODOLOGÍA.....	9
1.4.1 ENFOQUE METODOLÓGICO	9
1.5 ALCANCE	10
CAPÍTULO II: MARCO TEÓRICO	11
2.1 ANTECEDENTES	11
2.2 FUNDAMENTACIÓN TEÓRICA	13
2.2.1 INTELIGENCIA ARTIFICIAL APLICADA A LA EDUCACIÓN Y AL APRENDIZAJE DE LENGUAS	13
2.2.2 CHATBOTS Y TUTORES CONVERSACIONALES COMO HERRAMIENTAS EDUCATIVAS	14
2.2.3 APRENDIZAJE AUTÓNOMO Y PERSONALIZACIÓN MEDIADA POR IA	14
2.2.4 ALINEACIÓN PEDAGÓGICA Y DESAFÍOS ACTUALES.....	15
2.3 FUNDAMENTACIÓN TÉCNICA	16
2.3.1 MODELO DE INTELIGENCIA ARTIFICIAL Y API DE LLM	16
2.3.1.1 HERRAMIENTAS.....	18
2.3.1.2 COMPARACIÓN DE LAS HERRAMIENTAS	19
2.3.2 FRONTEND.....	20
2.3.2.1 HERRAMIENTAS.....	21
2.3.2.2 COMPARACIÓN DE LAS HERRAMIENTAS	23
2.3.3 BACKEND.....	24
2.3.3.1 HERRAMIENTAS.....	25
2.3.3.2 COMPARACIÓN DE LAS HERRAMIENTAS	27
2.3.4 BASE DE DATOS RELACIONAL (SQL).....	28
2.3.4.1 HERRAMIENTAS.....	29
2.3.4.2 COMPARACIÓN DE LAS HERRAMIENTAS	31
2.3.5 BASE DE DATOS NO RELACIONAL (NOSQL).....	32

2.3.5.1	HERRAMIENTAS.....	33
2.3.5.2	COMPARACIÓN DE LAS HERRAMIENTAS	35
2.3.6	AUTENTICACIÓN Y GESTIÓN DE USUARIOS	36
2.3.6.1	HERRAMIENTAS.....	37
2.3.6.2	COMPARACIÓN DE LAS HERRAMIENTAS	39
2.4	SELECCIÓN DE LA METODOLOGÍA ÁGIL	40
2.4.1	SCRUM	41
2.4.2	EXTREME PROGRAMMING (XP).....	42
2.4.3	COMPARACIÓN DE LAS METODOLOGÍAS	43
CAPÍTULO III: ANÁLISIS Y DISEÑO		45
3.1	CASOS DE USO.....	45
3.1.1	CASO DE USO 1 (CU1): INICIO DE SESIÓN EN EL SISTEMA.....	46
3.1.2	CASO DE USO 2 (CU2): GENERAR Y RESOLVER EJERCICIO.	47
3.1.3	CASO DE USO 3 (CU3): CONSULTAR PROGRESO ACADÉMICO....	48
3.2	ANÁLISIS DE REQUERIMIENTOS DEL PROTOTIPO.....	50
3.2.1	REQUERIMIENTOS FUNCIONALES	51
3.2.2	REQUERIMIENTOS NO FUNCIONALES	52
3.3	ARQUITECTURA DEL PROTOTIPO	53
3.3.1	ANÁLISIS DE LA ARQUITECTURA DEL PROTOTIPO	53
3.3.2	DISEÑO DE LA ARQUITECTURA DEL PROTOTIPO.....	55
3.4	MODELADO DE DATOS.....	57
3.4.1	MODELO DE DOMINIO	58
3.4.2	MODELO CONCEPTUAL.....	59
3.4.3	MODELO LÓGICO	60
3.4.4	MODELO FÍSICO	61
3.5	DISEÑO DE LA INTERFAZ	62
CAPÍTULO IV: DESARROLLO E IMPLEMENTACIÓN		63
4.1	PREPARACIÓN DEL ENTORNO DE DESARROLLO.....	63
4.1.1	SQL SERVER MANAGEMENT STUDIO (SSMS).....	63
4.1.2	VISUAL STUDIO (.NET FRAMEWORK)	63
4.1.3	VISUAL STUDIO CODE (REACT)	64
4.1.4	MONGODB COMPASS.....	64
4.2	DESARROLLO DE LAS FUNCIONALIDADES	65
4.2.1	SPRINT 1 (SP1): IMPLEMENTACIÓN Y VALIDACIÓN DE LAS BASES DE DATOS	65

4.2.1.1	OBJETIVO:	65
4.2.1.2	ACTIVIDADES	65
4.2.1.3	RESULTADOS ESPERADOS	66
4.2.1.4	EVIDENCIAS	66
4.2.1.5	TABLA DE TERMINACIÓN	68
4.2.2	SPRINT 2 (SP2): DESARROLLO DEL BACKEND Y CONEXIÓN CON SERVICIOS EXTERNOS (LLM Y AUTENTICACIÓN).....	69
4.2.2.1	OBJETIVO	69
4.2.2.2	ACTIVIDADES	69
4.2.2.3	RESULTADOS ESPERADOS	70
4.2.2.4	EVIDENCIAS	70
4.2.2.5	TABLA DE TERMINACIÓN	72
4.2.3	SPRINT 3 (SP3): DESARROLLO DE LA INTERFAZ WEB.....	73
4.2.3.1	OBJETIVO	73
4.2.3.2	ACTIVIDADES	73
4.2.3.3	RESULTADOS ESPERADOS	74
4.2.3.4	EVIDENCIAS	74
4.2.3.5	TABLA DE TERMINACIÓN	76
4.3	PRUEBAS	76
4.3.1	Pruebas funcionales del backend (API REST).....	76
4.3.2	Pruebas de integración de la interfaz de usuario con la API REST	78
4.3.3	Pruebas de validación funcional del modelo de lenguaje (API LLM)	78
CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES		81
5.1	CONCLUSIONES.....	81
5.2	RECOMENDACIONES	82
BIBLIOGRAFÍA		83
ANEXOS.....		87
	ACTA DE INICIACIÓN SPRINT 1 (SP1)	87
	ACTA DE INICIACIÓN SPRINT 2 (SP2)	88
	ACTA DE INICIACIÓN SPRINT 3 (SP3)	89
	ACTA DE ACEPTACIÓN	90

CONTENIDO DE FIGURAS

Figura 1 <i>Caso de uso 1</i>	46
Figura 2 <i>Caso de uso 2</i>	47
Figura 3 <i>Caso de uso 3</i>	48
Figura 4 <i>Diagrama de arquitectura del prototipo</i>	55
Figura 5 <i>Diagrama de modelo de dominio</i>	58
Figura 6 <i>Diagrama de modelo conceptual</i>	59
Figura 7 <i>Diagrama de modelo lógico</i>	60
Figura 8 <i>Diagrama de modelo físico</i>	61
Figura 9 <i>Interfaz de ejemplo</i>	62
Figura 10 <i>Evidencia 1 - Sprint 1</i>	66
Figura 11 <i>Evidencia 2 - Sprint 1</i>	67
Figura 12 <i>Evidencia 3 - Sprint 1</i>	67
Figura 13 <i>Evidencia 1 - Sprint 2</i>	70
Figura 14 <i>Evidencia 2 - Sprint 2</i>	71
Figura 15 <i>Evidencia 3 – Sprint 2</i>	71
Figura 16 <i>Evidencia 1 - Sprint 3</i>	74
Figura 17 <i>Evidencia 2 - Sprint 3</i>	75
Figura 18 <i>Evidencia 3 - Sprint 3</i>	75
Figura 19 <i>Prueba funcional 1</i>	77
Figura 20 <i>Prueba funcional 2</i>	77
Figura 21 <i>Prueba integración 1</i>	78
Figura 22 <i>Prueba validación 1</i>	79
Figura 23 <i>Prueba validación 2</i>	80

CONTENIDO DE TABLAS

Tabla 1 <i>Comparativa de LLM's</i>	19
Tabla 2 <i>Comparación de frameworks Frontend</i>	23
Tabla 3 <i>Comparación de frameworks Backend</i>	27
Tabla 4 <i>Comparación de bases de datos relacionales</i>	31
Tabla 5 <i>Comparación de bases de datos NoSQL</i>	35
Tabla 6 <i>Comparación de servicios de autenticación</i>	39
Tabla 7 <i>Comparación de metodologías ágiles</i>	43
Tabla 8 <i>Tabla de terminación - Sprint 1</i>	68
Tabla 9 <i>Tabla de terminación - Sprint 2</i>	72
Tabla 10 <i>Tabla de terminación - Sprint 3</i>	76

CAPÍTULO I: INTRODUCCIÓN

1.1 JUSTIFICACIÓN

En nuestro contexto actual, la inteligencia artificial juega un rol fundamental en una gran cantidad de ámbitos debido a que tiene una variedad de usos prácticos y a que genera facilidad y comodidad para el usuario, este hecho abre una ventana enorme para innovar con los usos de esta herramienta. No es para nada un secreto que, desde el lanzamiento de las primeras aplicaciones masivas de IA, uno de los sectores que ha sido altamente revolucionado es la educación. La inteligencia artificial, lejos de tener una presencia débil, se presenta como una fuerza disruptiva que genera esperanza, inquietud y tensión. Este hecho inquieta las formas tradicionales de la enseñanza y el aprendizaje (UNESCO, 2025). Esto debido al reto que implica adaptar o combinar la educación tradicional, a la que gran parte de la población está acostumbrada, con las infinitas ventajas que brinda la inteligencia artificial tanto a estudiantes como también a maestros.

La facilidad que tienen los modelos de lenguaje “LLM” de generar ideas, cumplir con tareas específicas o asistir a los usuarios en procesos creativos generan un reto para las instituciones educativas y sus metodologías de enseñanza. Hace algunos años, al cursar la escuela, solíamos imaginar un “lápiz mágico” que hiciera nuestras tareas por nosotros. En ese entonces parecía algo idílico, pero hoy en día con la fuerte presencia de la tecnología, la inteligencia artificial hace de esa idea prácticamente una realidad. Según la UNESCO (2025), a lo que se enfrenta la educación es mucho más que una actualización con las herramientas y medios tecnológicos a los que estábamos acostumbrados, sino más bien a un cambio estructural; que de adaptarse correctamente acarrearía no solo el poder modernizar las aulas, sino que se evolucionaría a un mejor modelo de aprendizaje. Es en este sentido que la intervención de las herramientas de IA en los procesos de enseñanza, instan a que la educación tenga un enfoque más práctico y priorizando el razonamiento de los alumnos.

En América Latina, se evidencia que la región se encuentra en una etapa de creciente adopción de la IA en varios contextos, específicamente en la educación, pero que enfrenta barreras estructurales, tales como brechas digitales de los usuarios,

formación del personal docente e infraestructura tecnológica que limitan el impacto efectivo que esta pueda tener (Rivas, 2025). Pese a eso, se reconoce también la oportunidad de que la inteligencia artificial permite una personalización educativa significativa y que incluso con los limitantes mencionados contribuya a reducir las desigualdades en el acceso a la educación de calidad en la región.

Entrando a un tema más específico, de manera global, la inteligencia artificial está siendo parte activa de la enseñanza de idiomas. Uno de los usos más comunes son las tutorías conversacionales, chatbots adaptativos y retroalimentación automática, herramientas que aprovechan el Procesamiento del Lenguaje Natural para simular una experiencia real de aprendizaje. Estudios recientes muestran que estas herramientas no solo mejoran la práctica del habla y la escritura, sino que también tienen un impacto positivo al reducir la presión que genera la comunicación en un idioma en el que los estudiantes son inexpertos y de esta manera aumentando su disposición a comunicarse en inglés. Por ejemplo, Du & Daniel (2024) realizan una revisión muy detallada sobre el uso de los chatbots con Inteligencia Artificial para práctica del habla como parte del proceso de aprendizaje del inglés como una lengua extranjera, y en su investigación encuentran efectos positivos en la motivación, la competencia comunicativa percibida y la reducción de la ansiedad lingüística.

Teniendo en cuenta todos estos aspectos, la academia English Valley, dedicada a la enseñanza del idioma inglés, se enfoca en prestar sus servicios a sus clientes, impartiendo sus clases con una metodología activa basada en experiencias reales, sin descuidar los contenidos estructurales del Marco Común Europeo de Referencia (MCER). En este enfoque, la práctica continua y efectiva en las cuatro destrezas del idioma; como lo son Listening o Escuchar, Speaking o Hablar, Reading o Leer y Writing o Escribir, llevadas de la mano del aprendizaje de las estructuras gramaticales, son fundamentales para lograr una adquisición efectiva del idioma. Si bien la academia, por medio de la implementación de su metodología, busca generar actividades lúdicas e inspiradas en contextos reales para que los estudiantes lleven un proceso de aprendizaje más orgánico; se ha identificado que existen ciertos casos de alumnos que enfrentan limitaciones debido a diversos factores como la disponibilidad de tiempo fuera del aula, condiciones personales, o barreras sociales

que dificultan la participación activa en clases presenciales y que truncan el progreso del alumno generando un sentimiento de frustración en su proceso de aprendizaje.

En este contexto, el desarrollo de un prototipo de un tutor basado en inteligencia artificial orientado a fomentar la práctica del idioma representa una solución tecnológica innovadora, accesible y escalable. Este proyecto utilizará modelos de IA orientados al procesamiento de lenguaje natural (NLP) para simular conversaciones reales con el estudiante y generar ejercicios según la destreza seleccionada adaptando el contenido al nivel A1 del MCER. Dicho tutor será capaz de interpretar respuestas, brindar correcciones, sugerencias y mantener la coherencia conversacional, facilitando la práctica autónoma del idioma en un entorno accesible y seguro.

“La IA permite personalizar el aprendizaje, con condiciones y límites bien marcados” (UNESCO, 2025). En el ámbito del aprendizaje de idiomas es siempre esencial tener el componente práctico y vivencial, por lo que el hecho de saber dirigir el potencial de los modelos de inteligencia artificial en aspectos como la práctica por medio de la generación de algunos ejercicios dinámicos, la retroalimentación oportuna y específica de las preguntas realizadas y por último la capacidad de comunicarse en el idioma natal del estudiante, presentan una ventaja estratégica en este proceso de aprendizaje. Un estudio exploratorio sobre evaluación escrita automatizada *Automated Writing Evaluation AWE* encontró que los alumnos que usan estas herramientas mejoran más rápido en precisión gramatical y tienen más oportunidades para revisar y aprender de sus errores que los que solo reciben corrección humana más esporádica (Wang, y otros, 2024).

Este tipo de tecnologías fomentan la autonomía del estudiante, permitiendo prácticas fuera del aula en cualquier momento, sin depender del horario del profesor. Esta libertad no solo mejora la cantidad de práctica, sino también la motivación intrínseca, tal como revela la revisión sistemática de chatbots aplicados al speaking de la lengua inglesa (Du & Daniel, 2024). El componente práctico asincrónico combinado con la guía metodológica y vivencial impartida por la academia, brindan una ruta de aprendizaje clara y consistente que aprovecha las ventajas de la inteligencia artificial y elimina barreras o limitaciones al momento de utilizar este tipo de herramientas en el aprendizaje de un idioma. Es en este sentido que la implementación de un prototipo de tutor que fomente la práctica es conveniente para mejorar la calidad académica de

los estudiantes de English Valley. La implementación de herramientas de IA no significa reemplazar a los profesores, sino quitarles el peso de tareas repetitivas para darles la capacidad de enfocarse en los aspectos verdaderamente profundos de la enseñanza como lo son inspirar, ser mentores y construir el aprendizaje desde la calidez humana (UNESCO, 2025). Los directivos de la academia son conscientes de esta ventaja estratégica que se apega a los valores de la marca.

Desde el punto de vista técnico, este prototipo busca validar la viabilidad de integrar modelos de lenguaje actuales con funcionalidades clave dentro del proceso académico de la institución como lo son la generación de ejercicios específicos parametrizados según la destreza que se decida practicar, detección de errores gramaticales, evaluación de los ejercicios para cuantificar el progreso del alumno, y generación de diálogos temáticos alineados al currículo A1. El proyecto se desarrollará como una aplicación web, lo que favorece su implementación escalable y sostenible en la academia. Esta propuesta no solo representa un avance en la digitalización del proceso clave de English Valley, el cual es la enseñanza, sino que también abre la posibilidad de integrar herramientas de IA educativa en diferentes niveles y contextos, permitiendo ofrecer experiencias de aprendizaje más personalizadas, accesibles y efectivas.

1.2 PLANTEAMIENTO DEL PROBLEMA

Según EF Education First (2024) en los resultados del informe del English Proficiency Index EPI, el Ecuador ocupa el puesto 84 de los 116 países evaluados con un promedio de 465 sobre 700, lo cual demuestra un nivel de aptitud bajo en el idioma. El EPI es un informe que clasifica a países y regiones según las habilidades en inglés de sus adultos, y que se basa en los resultados del examen gratuito online EF SET. Dicho examen evalúa habilidades de lectura, escucha y gramática de los examinados. Nuestro país se encuentra entre los últimos países del ranking en Latinoamérica, específicamente ocupa el puesto 19 de 21 en la región. Este tipo de indicadores nos dan una idea de la realidad que vive nuestro país con respecto a la enseñanza del idioma inglés como una lengua extranjera. Existen varios factores y limitaciones que generan este tipo de resultados, pero como mencionan Quizhpe & Yanella (2024) uno de los desafíos al momento de aprender inglés como lengua extranjera, es que existen muy pocas oportunidades para practicar el idioma de manera activa.

En este sentido, la academia de inglés “English Valley”, la cual si bien mediante su metodología activa y vivencial ha logrado solucionar en gran manera el tema de generar práctica continua en el proceso de enseñanza del idioma con sus estudiantes. En este momento atraviesa por un periodo de transición e innovación en el cual se busca implementar nuevas formas de generar práctica del idioma para sus alumnos, priorizando dar solución a las limitaciones que se han identificado al momento de impartir las clases y que afectan directamente al ritmo de aprendizaje de los alumnos. En el marco de buscar formas en las cuales los estudiantes puedan tener un recurso de práctica constante para mejorar su rendimiento en las habilidades propias del idioma, surge la idea de aplicar la inteligencia artificial como recurso complementario y de apoyo al aprendizaje. Es importante destacar que el enfoque de este proyecto busca generar una síntesis entre la metodología vivencial de la academia y los beneficios específicos enfocados a la práctica del idioma; es decir, no se quiere caer en la descentralización de estos componentes.

Como plantea UNESCO (2025), al momento de integrar IA en instituciones educativas, siempre existe la fundamentación de cómo hacerlo de manera segura,

efectiva y con un enfoque futuro. Es entonces donde la mayoría de las instituciones educativas; sin importar su tamaño, público objetivo o enfoque educativo, quedan atascadas en un modo reactivo en el cual prueban aplicaciones atractivas y se enfocan en las demostraciones de estas, cayendo en un círculo vicioso de probar herramientas sin ningún norte claro. Es por eso por lo que es esencial, tener una proyección y planificación de cómo se integrarán las herramientas dentro del proceso educativo de la academia. La generación de un aplicativo que utilice la inteligencia artificial debe ser parametrizado y alineado a los contenidos y resultados de aprendizaje formales del idioma, en este caso regulados por el Marco Común Europeo de Referencia para las lenguas y al enfoque metodológico de la academia. Es imperativo mencionar que gran parte del éxito de la implementación de la IA, radica en el desarrollo de programas de formación integral que involucren a los usuarios finales y que aborden no solo temas técnicos sino también la integración pedagógica efectiva de este tipo de herramientas (Sanz Manzanedo, 2025).

Con todo este contexto previo la academia English Valley, plantea el desarrollo de un aplicativo propio que complemente su servicio, siempre de la mano de su guía metodológica y académica. De esta manera aumentando la posibilidad de que los alumnos de la academia puedan practicar y mejorar su desempeño en los tiempos de aprendizaje autónomo fuera del aula de una manera más dinámica y con un recurso flexible que haga la práctica entretenida haciendo que los alumnos puedan relacionar tópicos necesarios para su formación en el idioma con temas de su interés. Así solucionando ciertas restricciones operativas como el tiempo de las clases, el número de estudiantes por curso y, además, solucionando posibles problemas personales de inseguridad o complejos en aspectos esenciales del aprendizaje, dado a que se ha identificado que los estudiantes se limitan a participar en las sesiones de clases presenciales, principalmente por temor a equivocarse; incluso cuando la academia prioriza un ambiente de empatía y crecimiento de los alumnos.

Este proyecto, por ende, sobresale y se distingue del uso de tecnologías tradicionales, como ejercicios grabados, videos estáticos, ejercicios de textos tradicionales, que no logran replicar la experiencia de práctica real ni permite adaptación a los intereses, nivel o desempeño del estudiante. Este tipo de proyectos han emergido como herramientas innovadoras para el aprendizaje de idiomas, ofreciendo interacción en

tiempo real y práctica conversacional. Dado a que ofrecen una práctica conversacional que es crucial para el desarrollo de competencias comunicativas, permitiendo a los estudiantes interactuar en tiempo real y recibir retroalimentación inmediata, lo cual es difícil de conseguir en un entorno tradicional de enseñanza de idiomas como el aula (Sanz Manzanedo, 2025). El permitir a los estudiantes utilizar recursos adicionales para complementar la formación impartida por la academia, no solo incrementa sus competencias comunicativas, sino que aumenta su confianza y seguridad en sus conocimientos. De esa manera la academia y bajo la premisa apoyar la práctica autónoma del idioma inglés, espera mejorar no solo el rendimiento de sus alumnos sino agilizar el ritmo de las clases para, en posterior, enfocarse en una formación más profunda.

Frente a esta problemática, se propone el desarrollo de un prototipo de tutor de aprendizaje basado en IA, que combine procesamiento de lenguaje natural, reconocimiento de voz y generación de contenido de manera dinámica, para brindar una experiencia de práctica realista y personalizada. Se plantea aprovechar las ventajas respecto a flexibilidad que ofrece la IA para poder generar ejercicios que permitan a los estudiantes practicar los temas en los que consideren que tengan falencias o en los cuales quieran mejorar, complementando con la retroalimentación alineada a MCER sobre la resolución de dichos ejercicios. Además, el prototipo cuantificará las respuestas de los alumnos en los ejercicios planteados con el objetivo de guardar y analizar el progreso del usuario en las destrezas del idioma. Haciendo visibles las áreas y oportunidades de mejora en las cuales se debería invertir más tiempo de práctica. Este tutor podrá integrarse como una herramienta complementaria a la enseñanza presencial, enfocándose inicialmente en el nivel A1 y expandiéndose posteriormente a otros niveles.

1.3 OBJETIVOS

1.3.1 OBJETIVO GENERAL

- Desarrollar un prototipo funcional de un tutor para el aprendizaje del idioma inglés basado en inteligencia artificial, mediante la implementación de modelos de procesamiento del lenguaje natural, que genere ejercicios alineados a los contenidos del Marco Común Europeo de Referencia para las Lenguas (MCER) y a los intereses del usuario, con el fin de facilitar una experiencia de práctica personalizada. El prototipo permitirá evaluar las respuestas del estudiante, medir su progreso y proporcionar retroalimentación en tiempo real, orientándose al nivel A1 para favorecer la práctica autónoma y el fortalecimiento de las destrezas lingüísticas de los estudiantes de la academia English Valley.

1.3.2 OBJETIVOS ESPECÍFICOS

- Analizar los posibles requerimientos funcionales y no funcionales, alineados con la metodología pedagógica de la academia, que permitan diferenciar el proyecto de las aplicaciones genéricas de IA para el aprendizaje del idioma.
- Determinar una metodología de desarrollo que sea adecuada para el proyecto.
- Diseñar el flujo de funcionamiento de la aplicación para que sea atractiva y entretenida de utilizar para el usuario.
- Implementar funcionalidades básicas para la práctica por medio de ejercicios, la corrección gramatical y lingüística acordes al nivel A1.
- Desarrollar una interfaz web interactiva y accesible para que los estudiantes puedan practicar las destrezas del idioma y recibir feedback de sus respuestas.

1.4 METODOLOGÍA

Para el desarrollo del presente proyecto se busca adoptar un enfoque metodológico orientado a la construcción de un prototipo funcional de un tutor de aprendizaje del idioma inglés basado en inteligencia artificial para el nivel A1 del Marco Común Europeo de Referencia para las Lenguas. Dado el carácter tecnológico, educativo y con requerimientos con tendencia al cambio que tiene el proyecto, de preferencia se debe emplear una metodología ágil que permite el desarrollo iterativo del sistema, facilitando la incorporación de mejoras y ajustes conforme se avanza en el desarrollo del prototipo, en coherencia con los requerimientos técnicos y pedagógicos de la academia English Valley.

1.4.1 ENFOQUE METODOLÓGICO

El proyecto busca desarrollar una solución tecnológica concreta que responde a una necesidad real, que ha sido identificada dentro del proceso de enseñanza perteneciente a la academia English Valley. Para el desarrollo del mismo, empleamos la metodología ágil Scrum, que es un marco de trabajo flexible, además es iterativo y orientado a la colaboración activa entre las partes involucradas. Esta metodología es ideal para proyectos en los que los requerimientos pueden llegar a modificarse conforme se va obteniendo retroalimentación del usuario, como es el caso en el entorno educativo, especialmente con un proyecto que representa una innovación para la academia. A través de Scrum, el desarrollo del prototipo se organizará en sprints cortos que permitirán desarrollar, evaluar y mejorar las funcionalidades del tutor de aprendizaje, de esta manera asegurando que cada etapa aporte valor tangible al proyecto.

La elección de Scrum corresponde a la naturaleza dinámica del entorno académico de English Valley, donde las funcionalidades del proyecto pueden evolucionar conforme se avanza en el proceso de desarrollo, dado a que es un proyecto en el que se incursionará y experimentará. Este enfoque facilita la incorporación de ajustes derivados de la retroalimentación con el personal de la academia, garantizando que el sistema conserve su pertinencia pedagógica y su alineación con los objetivos de aprendizaje del nivel A1. Además, el enfoque colaborativo de Scrum promueve una

comunicación constante entre el generador del proyecto y las partes interesadas, favoreciendo que las decisiones técnicas se tomen con base en criterios educativos y funcionales de los profesionales a cargo de la academia.

En conjunto, la aplicación de esta metodología nos permitirá que el desarrollo del tutor avance de manera estructurada pero manteniendo la flexibilidad necesaria para poder adaptarse a los hallazgos que surjan durante el desarrollo. De este modo, se asegura que el producto final no solo cumpla con los requerimientos técnicos analizados y previstos, sino que también contribuya de manera significativa al fortalecimiento del proceso de enseñanza-aprendizaje en English Valley.

1.5 ALCANCE

El alcance de este proyecto se enfoca en el desarrollo de un prototipo funcional, enfocado en estudiantes de nivel A1 del idioma inglés en la academia English Valley:

- Un flujo que genera ejercicios y actividades cuantificables basadas en destrezas lingüísticas y temas de preferencia para el usuario.
- Un sistema básico de corrección y sugerencia gramatical adaptado a errores comunes que se cometen en las actividades. De esta manera garantizando tener una retroalimentación de los ejercicios.
- Una interfaz responsiva accesible desde computadoras (Web).
- Ejercicios temáticos limitados a contextos relevantes del currículo A1 basadas en las habilidades pertinentes al nivel.

CAPÍTULO II: MARCO TEÓRICO

2.1 ANTECEDENTES

“El aprendizaje de una segunda lengua debe entenderse como un sistema dinámico, no simplemente lineal, en el que múltiples factores interactúan y emergen de forma no predecible” (Larsen-Freeman, 1997). El proceso de aprendizaje de un idioma tiene múltiples aristas y es inherentemente complejo, si bien no debemos verlo de manera estricta si debemos considerar que muchas veces es estandarizado o parametrizado lo que exige un enfoque riguroso y bien diseñado. En varios contextos educativos, especialmente como sucede en Ecuador, la complejidad que mencionábamos muchas veces conduce a métodos rígidos y muy limitantes para el proceso de aprendizaje de los alumnos, lo que puede generar rechazo hacia el aprendizaje de idiomas, especialmente del inglés. “Esta tendencia de aplicar modelos tradicionales de enseñanza se contrasta con la necesidad de adaptarse a diferentes niveles de competencia, pues investigaciones recientes muestran que la complejidad del proceso de enseñanza lingüístico varía significativamente según el nivel del estudiante y las condiciones de instrucción” (Kuiken, 2022). De manera cultural, en Ecuador, existe cierto temor de aprender inglés; en gran parte por malas experiencias relacionadas a la impartición de este.

Ahora bien, no se puede negar que el inglés es un idioma global con una gran presencia y que suma varios beneficios a quienes lo manejan de manera solvente. De ahí la proliferación de academias o institutos que se comprometen a impartirlo enmarcados en el Marco Común Europeo de Referencia para las Lenguas.

En el caso de la academia English Valley, el aprendizaje del inglés está regulado conforme al estándar internacional más reconocido Marco Común Europeo de Referencia para las Lenguas MCER y los servicios se imparten mediante un enfoque pedagógico formal y estructurado dada la naturaleza del estándar utilizado. Sin embargo, la institución busca simultáneamente y con un esfuerzo mayor, implementar metodologías vivenciales que transformen la experiencia de aprendizaje, integrando actividades que conecten los contenidos teóricos con situaciones comunicativas reales.

La planificación académica de la institución promueve el desarrollo activo del alumno y su participación constante dentro del aula de clases; ahora bien, existen restricciones operativas que dificultan la aplicación completa de las estrategias metodológicas diseñadas para fomentar la interacción y colaboración. Entre las principales limitaciones mencionadas se encuentra la duración reducida de las clases, lo que impide revisar con profundidad los contenidos gramaticales, su explicación y por último su práctica vivencial posterior. Ante este escenario, la academia ha optado por incorporar actividades autónomas que incentiven al estudiante a poner en práctica lo aprendido fuera del aula. Estas actividades constituyen un complemento importante del proceso formativo, pero al realizarse de manera individual y sin retroalimentación inmediata, presentan el desafío de mantener la motivación.

Por esta razón, se ha identificado la necesidad de fortalecer los espacios de práctica independiente mediante herramientas que acompañen al estudiante y que permitan evaluar y cuantificar los resultados de sus ejercicios. Esta información resulta esencial para monitorear el progreso, ajustar estrategias pedagógicas y garantizar un aprendizaje más integral y medible. Dejando libre al maestro para enfocarse en poner en práctica las metodologías planificadas y estudiadas por la academia enfocadas en promover la convivencia de los alumnos.

“En los últimos años, la integración de tecnologías basadas en inteligencia artificial en la educación ha mostrado un crecimiento exponencial, gracias a su capacidad de personalizar y adaptar los contenidos de aprendizaje al ritmo y nivel de cada estudiante” (Sasmayunita & Assiddiq, 2025). No obstante, según Sasmayunita & Assiddiq (2025) muchas de las soluciones disponibles carecen de una base pedagógica sólida y no se encuentran alineadas al MCER, lo que limita su aplicabilidad en contextos académicos formales. “Estudios recientes demuestran que la instrucción mediada por IA puede mejorar significativamente el rendimiento en inglés, incrementar la motivación intrínseca y fortalecer la autorregulación del aprendizaje en estudiantes de segunda lengua” (Wei, 2023).

2.2 FUNDAMENTACIÓN TEÓRICA

“El aprendizaje de lenguas asistido por tecnología ha evolucionado hacia la integración de sistemas inteligentes capaces de adaptar la enseñanza a las necesidades de cada estudiante. En este contexto, la Inteligencia Artificial se ha consolidado como una herramienta transformadora, al permitir la personalización, automatización y retroalimentación continua en los procesos educativos” (Wei, 2023). La capacidad que tiene para analizar patrones y responder de forma adaptativa, abre la puerta a experiencias de aprendizaje más dinámicas y centradas en el estudiante.

2.2.1 INTELIGENCIA ARTIFICIAL APLICADA A LA EDUCACIÓN Y AL APRENDIZAJE DE LENGUAS

En las últimas décadas, la inteligencia artificial ha pasado de ser utilizada como una herramienta experimental a ser utilizada con un papel protagónico en varios entornos, entre ellos la educación. La capacidad que tiene para poder procesar grandes volúmenes de datos y aprender de las interacciones con los usuarios permite generar experiencias de aprendizaje con un enfoque personalizado y que son altamente interactivas. En el ámbito de la enseñanza de idiomas específicamente, la IA facilita la retroalimentación inmediata y la evaluación automática de las respuestas de los usuarios permitiendo cuantificar su progreso, elementos que promueven la autonomía, la motivación y la autorregulación de los estudiantes (Wei, 2023).

La evidencia obtenida a partir de estudios recientes muestra que la implementación de sistemas educativos potenciados por IA no se limita al incremento del rendimiento académico, sino que también incide de manera significativa en el fortalecimiento de la motivación estudiantil. Los estudios de Wei (2023) demostraron que los estudiantes que participaron en entornos apoyados por IA alcanzaron mejores resultados en comprensión lectora, gramática y escritura que aquellos que utilizaron métodos tradicionales, además de manifestar mayor compromiso y autoeficacia en su proceso formativo. Estos hallazgos demuestran que la IA no actúa únicamente como una herramienta de apoyo, sino como un agente que transforma la dinámica educativa al adaptar los contenidos al ritmo y nivel de cada estudiante.

2.2.2 CHATBOTS Y TUTORES CONVERSACIONALES COMO HERRAMIENTAS EDUCATIVAS

La aplicación de la inteligencia artificial en la enseñanza del idioma inglés se ha consolidado de manera significativa a través del desarrollo de chatbots y tutores conversacionales, gracias en gran parte al desarrollo de este tipo de tecnologías, estas herramientas permiten sostener interacciones coherentes, interpretar errores lingüísticos y proporcionar retroalimentación personalizada orientada al proceso de aprendizaje comunicativo. Du & Daniel (2024), en sus estudios destacan que estos sistemas permiten simular interacciones reales, favoreciendo la práctica del habla y de la escritura, además de reducir la ansiedad comunicativa y fortalecer la autoconfianza de los aprendices. La revisión de los hallazgos presentados por estos autores nos permite identificar que la utilización de este tipo de herramientas incide positivamente en la mejora de la competencia comunicativa, la motivación y la frecuencia de práctica oral de los estudiantes debido a que se genera un entorno libre de juicios sociales, que habitualmente limitan el desempeño académico del alumno. Estas condiciones son especialmente relevantes en niveles iniciales como el nivel A1, en donde, si bien las bases gramaticales son determinantes, la exposición al idioma para permitir que el alumno se adapte y por supuesto, la repetición es esencial para el desarrollo de la fluidez. Además, los tutores conversacionales poseen la capacidad de registrar y analizar las respuestas de los usuarios, generando métricas que facilitan la evaluación de forma cuantitativa del aprendizaje y disminuyendo la ambigüedad en el proceso de formación usando agentes de inteligencia artificial.

2.2.3 APRENDIZAJE AUTÓNOMO Y PERSONALIZACIÓN MEDIADA POR IA

Uno de los aportes más significativos de la inteligencia artificial en el ámbito educativo es que nos da la posibilidad de fortalecer la parte autónoma del aprendizaje. Estas tecnologías permiten a los estudiantes planificar, monitorear y evaluar su propio progreso mediante indicadores personalizados. “La retroalimentación inmediata y la posibilidad de repetir ejercicios con variaciones controladas fortalecen la autorregulación y la motivación intrínseca, factores esenciales para el aprendizaje efectivo de una segunda lengua” (Wei, 2023).

Lo que buscamos con el tutor de aprendizaje propuesto en este proyecto es replicar e implementar este tipo de ventajas mediante la integración de modelos de IA que se parametricen con el nivel y progreso del estudiante. A diferencia de las aplicaciones genéricas de aprendizaje, el prototipo se diseñará bajo los parámetros del Marco Común Europeo de Referencia para las Lenguas (MCER) y bajo la influencia académica de la academia English Valley, garantizando así que las actividades y correcciones respondan a los descriptores específicos del nivel A1. De esta forma, se asegura la coherencia pedagógica del prototipo y se promueve la posibilidad de medir avances tangibles en las destrezas lingüísticas trabajadas.

2.2.4 ALINEACIÓN PEDAGÓGICA Y DESAFÍOS ACTUALES

A pesar de los avances agigantados y constantes en el campo de la inteligencia artificial enfocada al aspecto educativo, varios autores nos advierten sobre una brecha persistente entre el potencial tecnológico de estas herramientas y su integración pedagógica efectiva. Muchas de las soluciones disponibles carecen de una base instruccional sólida y no consideran principios didácticos ni estándares curriculares, lo que puede limitar su impacto y reducirlas a simples recursos complementarios sin dirección pedagógica clara (Du & Daniel, 2024).

Por este motivo, el desarrollo de un tutor de aprendizaje basado en IA debe priorizar la coherencia entre la innovación tecnológica y la fundamentación educativa. En este sentido, resulta esencial integrar los descriptores del MCER, las metodologías activas y los principios de evaluación formativa, garantizando que el sistema mantenga su relevancia pedagógica. En el caso de la academia English Valley, esta alineación permitirá que el tutor funcione como un recurso complementario al trabajo docente, potenciando las prácticas autónomas y fortaleciendo el aprendizaje sin reemplazar la interacción humana que caracteriza su modelo metodológico.

2.3 FUNDAMENTACIÓN TÉCNICA

El desarrollo del tutor de aprendizaje propuesto se sustenta en varios conceptos técnicos que orientan su diseño e implementación. En esta sección detallaremos la estructura y las herramientas necesarias para llevar a cabo el proyecto.

2.3.1 MODELO DE INTELIGENCIA ARTIFICIAL Y API DE LLM

La Inteligencia Artificial y en particular el *Procesamiento del Lenguaje Natural NLP*, por medio de las API de los *Large Language Model LLM* constituyen la base tecnológica del tutor de aprendizaje. Este tipo de tecnologías permiten que las máquinas comprendan, generen y respondan usando nuestro lenguaje, esto nos da ventajas enormes en la generación de contenido. De acuerdo con Russell y Norvig (2022), el NLP representa una de las ramas que están más avanzadas con respecto a la IA moderna, está orientada a interpretar y producir lenguaje natural a partir de modelos y técnicas computacionales que se han entrado con grandes volúmenes de datos de texto.

En la actualidad, según OpenAI (2024a), el avance de los LLM ha permitido un salto cualitativo en el desarrollo de aplicaciones conversacionales, esto es gracias a que estos modelos pueden no solo generar texto, sino también adaptarse a diferentes contextos, niveles de complejidad y estilos comunicativos. Los LLM funcionan mediante redes neuronales profundas entrenadas sobre enormes conjuntos de datos lingüísticos, esto es lo que les otorga la capacidad de generar respuestas que imitan el razonamiento humano y que mantienen la coherencia dentro de una conversación.

“Una API de LLM (Application Programming Interface for Large Language Models) es una interfaz que permite acceder a las capacidades de un modelo de lenguaje pre entrenado sin necesidad de entrenar o alojar dicho modelo localmente. Estas APIs exponen endpoints que reciben solicitudes (prompts) en formato de texto y devuelven respuestas generadas por el modelo, lo que simplifica su integración dentro de aplicaciones web o móviles” (Brown, y otros, 2020).

El funcionamiento general consiste en enviar una petición al endpoint del modelo, junto con los parámetros de configuración y el “prompt” que es la instrucción que deseamos que se ejecute. Entre los principales parámetros se encuentran:

- **Temperature:** controla el nivel de creatividad o aleatoriedad en las respuestas. Aquí los valores bajos (0.2–0.5) generan respuestas más determinantes. Mientras que, los valores altos (0.7–1.0) permiten mayor creatividad.
- **Max tokens:** define la extensión máxima de la respuesta generada.
- **Top-p y frequency penalty:** ajustan la probabilidad de palabras y la repetición de términos, respectivamente.
- **System prompt:** permite establecer el contexto o rol del modelo antes de iniciar la conversación (por ejemplo, instruir al modelo a comportarse como un tutor de inglés nivel A1).

Estos parámetros permiten parametrizar el comportamiento del modelo para ajustarlo a necesidades pedagógicas concretas, como mantener el lenguaje dentro del nivel A1 del Marco Común Europeo de Referencia para las Lenguas (MCER), generar ejercicios controlados o brindar retroalimentación específica al estudiante.

En el caso del proyecto, el enfoque se basa en el uso de modelos de lenguaje pre entrenados mediante API para generar y evaluar ejercicios del nivel A1, ofrecer retroalimentación personalizada y simular conversaciones que favorezcan la práctica autónoma del inglés. El tutor no se limita a generar texto, sino que utiliza las respuestas del modelo para interpretar las respuestas del estudiante y retroalimentar con sugerencias de mejora, aciertos que haya realizado el alumno y pasos siguientes.

Este enfoque permite combinar personalización y escalabilidad, ya que los modelos se ejecutan en servidores externos de los proveedores (OpenAI o Google), lo que evita los altos costos de entrenamiento y mantenimiento de modelos propios, garantizando al mismo tiempo acceso a tecnología de vanguardia (Du & Daniel, 2024).

2.3.1.1 HERRAMIENTAS

Para la integración del modelo de IA dentro del sistema, se consideran dos alternativas principales:

- **OpenAI:** Según OpenAI (2024a), ofrece una de las APIs más avanzadas y ampliamente documentadas para el uso de modelos de lenguaje. Sus modelos GPT (Generative Pre-trained Transformer) se caracterizan por su capacidad de mantener coherencia conversacional y generar respuestas con precisión gramatical y contextual. Además, proporciona una API REST bien estructurada con SDKs disponibles en lenguajes como C#, Python y JavaScript, lo que facilita su integración con el backend del tutor de aprendizaje.
- **Google Gemini:** “Google Gemini representa la propuesta más reciente de modelos de lenguaje generativo de Google DeepMind. Destaca por su capacidad multimodal avanzada, integrando texto, imagen y audio en una misma interacción” (Google DeepMind, 2024). Su API, disponible a través de Google AI Studio, ofrece múltiples funcionalidades y una interfaz intuitiva para el manejo de prompts y parámetros.

2.3.1.2 COMPARACIÓN DE LAS HERRAMIENTAS

Tabla 1

Comparativa de LLM's

Criterio técnico	OpenAI	Google Gemini
Calidad output	Alta, soporta contextos extensos	Alta, con buen manejo multimodal
Control de parámetros	Amplio y flexible	Moderado, dependiente del entorno de Google AI Studio
TTS/STT	Incluido en la API	APIs complementarias de Google Cloud Speech
Integración	SDKs oficiales y documentación extensa	SDKs en expansión, documentación más reciente
Costo y escalabilidad	Uso por tokens con costos accesibles.	Coste variable según servicios asociados (Google Cloud)
Soporte y comunidad	Amplia comunidad global y soporte técnico	Comunidad global, centrada en entorno Google

Nota: Comparativa de capacidades lingüísticas y parámetros configurables en APIs LLM.
 Autor: Nicolás Rojas, 2025.

Ambas plataformas ofrecen capacidades avanzadas para el desarrollo de un tutor de aprendizaje, pero OpenAI se selecciona como la opción principal para este proyecto. Esta decisión se fundamenta en los siguientes aspectos:

1. Mayor madurez tecnológica y documentación: OpenAI cuenta con guías de implementación detalladas en su plataforma, lo que facilita la integración con el backend desarrollado.
2. Control de parámetros y personalización: su API nos permite configurar el modelo con un alto grado de detalle, ajustando prompts y temperatura para mantener la producción lingüística dentro del nivel A1 del MCER.
3. Compatibilidad y escalabilidad: Tiene integración y compatibilidad directa con proyectos de .NET y es muy adaptable con JS.
4. Funcionalidades multimodales: la nueva versión GPT-4o permite futuras extensiones del sistema con reconocimiento y generación de voz, coherente con los objetivos de práctica oral de la academia English Valley.

Por su parte, Google Gemini se considera una alternativa viable para versiones futuras del proyecto, especialmente si se busca una mayor integración con los servicios del ecosistema Google Cloud o funcionalidades multimodales avanzadas. (Brown, y otros, 2020)

En resumen, la elección de OpenAI como API de modelo de lenguaje (LLM) para el tutor de aprendizaje garantiza precisión lingüística, control de personalización y facilidad de integración con la arquitectura tecnológica seleccionada, factores que la convierten en la opción más adecuada para cumplir los objetivos pedagógicos y técnicos del proyecto.

2.3.2 FRONTEND

El Frontend se define como la capa de presentación del sistema. Esto quiere decir que representa la interfaz gráfica a través de la cual el usuario interactúa directamente con el prototipo. El desarrollo de este componente está enfocado directamente en la experiencia del usuario y en la presentación visual de los datos procesados por el backend, almacenados en la base de datos o consumidos de un servicio externo. Según W3Schools (2025), el frontend combina tecnologías de interfaz como HTML, CSS y JavaScript junto con frameworks modernos que permiten construir aplicaciones web interactivas, dinámicas y adaptables a distintos dispositivos.

En el contexto del tutor de aprendizaje propuesto, el frontend cumple un rol esencial, ya que será la interfaz mediante la cual los estudiantes interactúen con el modelo de inteligencia artificial para practicar las distintas destrezas del idioma, recibir retroalimentación, visualizar su progreso y acceder a las distintas funcionalidades del sistema. El diseño de dicha interfaz debe priorizar la usabilidad, la fluidez de interacción y la accesibilidad, ofreciendo una experiencia intuitiva tanto para usuarios con conocimientos tecnológicos básicos como para usuarios con menos experiencia.

Como concepto, este componente debe integrarse eficientemente con la capa de lógica (Backend) consumiendo servicios REST y manejando o renderizando componentes gráficos y multimedia como texto, audio y elementos visuales, priorizando que la información presentada al usuario sea sencilla y clara de digerir.

2.3.2.1 HERRAMIENTAS

Antes de detallar las herramientas seleccionadas, es pertinente definir lo que es un framework de frontend. “Un framework frontend es un conjunto de estructuras, componentes reutilizables y herramientas que permiten crear interfaces web de forma organizada, escalable y eficiente. Estos frameworks proporcionan una base estructural sobre la cual construir componentes visuales dinámicos, gestionar estados de la aplicación, manipular el DOM (Document Object Model) de manera reactiva y optimizar la comunicación con el backend” (Mozilla Developer Network, 2024).

El uso de frameworks modernos aporta ventajas técnicas relevantes. Como por ejemplo, mejora la modularidad del código mediante componentes reutilizables, facilita el enrutamiento interno, además permite una mejor gestión del estado de la interfaz, y ofrece herramientas integradas para el control del ciclo de vida de los componentes. La arquitectura basada en componentes favorece la integración con librerías externas, como aquellas necesarias para el manejo de voz o visualización de gráficos en el tutor de aprendizaje.

A continuación, se describen las herramientas candidatas consideradas para el desarrollo del frontend del sistema:

- **React:** “Su funcionamiento se basa en el uso del Virtual DOM, una representación ligera del DOM real que optimiza la actualización de los elementos en pantalla, incrementando el rendimiento en aplicaciones altamente interactivas” (Meta, 2024). Este enfoque declarativo facilita la construcción de interfaces, ya que los componentes se actualizan automáticamente cada vez que el estado de la aplicación cambia, así se reduce la complejidad del código y se vuelve más manipulable. React tiene la ventaja de que su arquitectura es modular y basada en componentes, esto permite dividir la interfaz en secciones independientes y que son reutilizables. También destaca su compatibilidad con TypeScript, lo que mejora la mantenibilidad y la detección temprana de errores en proyectos de gran escala. Además, cuenta con un extenso ecosistema de librerías y herramientas, como Redux para la gestión del estado global, React Router para el enrutamiento de vistas y diversos paquetes para la manipulación de

audio y video. A esto se suma su sencilla integración con APIs REST y GraphQL, lo que facilita la comunicación con el backend y el consumo eficiente de datos en sistemas complejos.

- **Vue.js:** Vue.js es un framework progresivo de JavaScript diseñado para crear interfaces de usuario reactivas y de alto rendimiento. Su principal ventaja radica en que es adaptable, es decir, puede integrarse gradualmente en proyectos existentes o emplearse para desarrollar aplicaciones completas del tipo SPA (*Single Page Applications*). “Vue utiliza una arquitectura basada en componentes; sin embargo, se distingue por su sintaxis más clara y su curva de aprendizaje más accesible, lo que lo convierte en una herramienta ideal para equipos en formación o proyectos con tiempos de desarrollo ajustados” (You, 2023).

2.3.2.2 COMPARACIÓN DE LAS HERRAMIENTAS

Tabla 2

Comparación de frameworks Frontend

Criterio técnico	React	Vue.js
Tipo	Biblioteca de UI	Framework
Arquitectura	Componentes y Virtual DOM	Componentes y reactividad bidireccional
Curva de aprendizaje	Media	Baja
Integración REST	Muy buena, compatible con Axios y Fetch	Muy buena, similar con Axios
Soporte para multimedia	Ecosistema de librerías JS	Mediante plugins externos
Rendimiento	Alto, gracias al Virtual DOM	Alto, con menor peso inicial
Comunidad y soporte	Amplia, respaldada por Meta	Amplia, con enfoque open source
Flexibilidad y escalabilidad	Alta (apto para grandes proyectos)	Media (apto para proyectos medianos o rápidos)

Nota: Comparativa del rendimiento, arquitectura y ecosistema de herramientas frontend.

Autor: Nicolás Rojas, 2025.

Ambas herramientas ofrecen un entorno moderno, eficiente y ampliamente adoptado para el desarrollo de aplicaciones web interactivas. Sin embargo, para el caso del proyecto propuesto, se selecciona React como la herramienta principal de desarrollo del frontend.

Esta elección se fundamenta en los siguientes aspectos:

1. Rendimiento y modularidad: React ofrece una arquitectura orientada a componentes que facilita la actualización dinámica de la interfaz y la reutilización de elementos, ideal para el manejo de la retroalimentación y las respuestas en tiempo real.
2. Ecosistema y soporte: Su integración con bibliotecas multimedia y APIs REST lo convierte en una opción ideal para conectar con el backend y con las funcionalidades de IA.

3. Escalabilidad y mantenimiento: React es ampliamente utilizado en entornos académicos y empresariales, lo que garantiza documentación, soporte y una base sólida para futuras ampliaciones del sistema (por ejemplo, añadir nuevas funcionalidades o niveles del MCER).

Por otro lado, Vue.js se mantiene como una alternativa viable para implementaciones más ligeras o prototipos funcionales, especialmente en etapas tempranas de validación de la interfaz o cuando se busque una curva de aprendizaje más rápida.

En conclusión, la elección de React como framework principal de frontend permite al tutor de aprendizaje ofrecer una experiencia de usuario fluida, personalizable y adaptable, en coherencia con los objetivos del proyecto y las necesidades pedagógicas de la academia English Valley.

2.3.3 BACKEND

El backend constituye la capa de lógica y es la que opera “detrás de la escena”, es decir, realiza todo aquello que no ve el usuario pero que hace posible que la aplicación funcione. Por ejemplo, recibe las peticiones del frontend, ejecuta la lógica del negocio, accede y gestiona los datos, consume servicios externos como APIs (en este caso concreto API's de IA), y garantiza la seguridad, consistencia y disponibilidad del sistema.

Para el proyecto del tutor de aprendizaje propuesto, el backend funcionará como intermediario entre la interfaz del estudiante, la API de modelo de lenguaje, los módulos de autenticación y las bases de datos. Es decir, tiene un rol fundamental y se convierte en el eje central como orquestador de los componentes de la aplicación. Por lo tanto, debe ofrecer una estructura organizada, modular y escalable que permita adaptarse al crecimiento funcional de la aplicación con miras al futuro (por ejemplo, abarcar más niveles del MCER) y también responder a los requisitos de tiempos de respuesta, integración de voz/texto y almacenamiento de historiales. Este enfoque concuerda con la definición general de desarrollo backend, que lo describe como “la parte del desarrollo web que se enfoca en el servidor, las bases de datos, los servicios y la lógica que los conecta” (Intelivita, 2025).

2.3.3.1 HERRAMIENTAS

Antes de pasar a describir cada una de las herramientas que se analizarán para esta parte del proyecto, es importante definir lo que entendemos como un framework de backend. Un framework de backend es un conjunto de librerías, convenciones, estructuras de proyecto y herramientas que facilitan la creación de servicios del lado del servidor. Como, por ejemplo, rutas/endpoint, middleware (pre-procesamiento de peticiones, autenticación, logging), conexión con las bases de datos, control de errores, seguridad y otras utilidades repetitivas. Utilizar un framework robusto dentro de un proyecto de desarrollo trae ventajas técnicas como la estandarización de la arquitectura; lo que favorece mantenimiento y escalabilidad, la integración más fluida con servicios externos (APIs, bases de datos, autenticación) y un ecosistema de extensiones y comunidad que acelera el desarrollo. Tal como señalan las revisiones de frameworks de desarrollo backend, estos permiten construir aplicaciones del lado del servidor de manera más segura, escalable y mantenible (Intelivita, 2025).

A continuación, se describen las herramientas candidatas específicas para este proyecto:

- **ASP.NET Core:** Es un framework multiplataforma de código abierto de Microsoft, diseñado para construir aplicaciones web modernas, APIs REST, microservicios y servicios en la nube. Su ORM asociado, Entity Framework Core (EF Core), permite interactuar con bases de datos relacionales de forma tipada, con migraciones, LINQ y control de integración. Entre sus características destacan el soporte nativo para inyección de dependencias, middleware configurable (logging, autenticación, CORS), alto rendimiento, y compatibilidad con contenedores y Linux/Windows/macOS (Microsoft, 2024). “EF Core permite mapear modelos de dominio a tablas de la base de datos, realizar consultas con LINQ expresivo, y mantener la consistencia de datos en entornos relacionales. Para el proyecto del tutor de aprendizaje, esta herramienta ofrece una base confiable para gestionar la lógica del negocio, la autenticación, el almacenamiento de usuarios y progreso, y la integración con servicios externos” (Microsoft, 2024).

- **NestJS:** “Es un framework progresivo para Node.js escrito en TypeScript, que adopta una arquitectura modular inspirada en patrones de diseño como MVC, inyección de dependencias, interceptores, guards y otros, lo que promueve la creación de aplicaciones server-side bien estructuradas y mantenibles” (Kumar, 2022). Permite la creación de endpoints REST, el uso de módulos independientes (por ejemplo, módulo de autenticación, módulo de LLM, módulo de datos), y la integración con ORMs como TypeORM o Prisma para manejar bases de datos relacionales o NoSQL. Al estar basado en TypeScript, ofrece tipado estático y mejor soporte para errores en tiempo de desarrollo. Dada su compatibilidad con el ecosistema JavaScript/TypeScript, resulta muy adecuado si el equipo va a compartir lógica o tipos con el frontend (por ejemplo, React).

2.3.3.2 COMPARACIÓN DE LAS HERRAMIENTAS

Tabla 3

Comparación de frameworks Backend

Criterio técnico	ASP.NET Core	NestJS
Lenguaje principal	C#	TypeScript
Paradigma arquitectónico	Tipado fuerte, OO, middleware/inyección	Tipado fuerte (TS), estructura modular/decoradores
Estructura del framework	Controladores + middleware nativo, integración con Visual Studio.	Módulos, controladores, providers, interceptors
ORM o acceso a datos	Entity Framework Core integrado	TypeORM/Prisma como librería externa
Autenticación y seguridad	Middleware nativo para OAuth2 y JWT	Sin integración directa con OAuth2
Rendimiento	Alto, multihilo	Alto, basado en event loop
Curva de aprendizaje	Media	Media-baja
Escalabilidad	Alta (microservicios, contenedores)	Alta (modular, distribuida)
Integración con APIs LLM externas	SDKs oficiales C#, integración fluida	SDKs oficiales JS/TS, buena integración
Mantenimiento y comunidad	Amplia comunidad .NET, soporte empresarial	Amplia comunidad JS/TS, open source activa. Documentación amplia.

Nota: Comparativa de rendimiento, estructura modular y ecosistema de desarrollo backend. Autor: Nicolás Rojas, 2025.

Ambas herramientas cumplen los requisitos fundamentales del proyecto: exposición de APIs REST para el frontend, integración con servicios externos (LLM, bases de datos), autenticación con dominio institucional, y almacenamiento híbrido (SQL + NoSQL).

Sin embargo, para el presente proyecto de tutor de aprendizaje, la selección recomendada es el framework ASP.NET Core por las siguientes razones específicas al contexto:

1. Poseo experiencia previa con el uso de esta herramienta, lo que reduce la curva de aprendizaje, acelera el desarrollo del prototipo y mejora la probabilidad de entregar una solución de calidad en el plazo académico planteado.
2. La gestión de datos relacionales (usuarios, progreso, roles) requiere integridad, tipado fuerte y operaciones transaccionales seguras; Entity Framework Core junto con ASP.NET Core ofrecen una solución madura para el desarrollo de lo planteado.
3. Dado que el sistema deberá integrarse con módulos complejos (IA, autenticación, bases de datos híbridas), contar con un ecosistema robusto y herramientas de desarrollo maduras facilita debugging, pruebas, monitoreo y mantenibilidad a largo plazo.

Por su parte, **NestJS** se mantiene como una alternativa viable, especialmente si en el futuro la academia plantea una extensión del sistema orientada hacia el lado del frontend o se requiere una integración más estrecha con frameworks como React o Vue. Asimismo, esta opción resulta favorable si se busca desplegar la aplicación en un entorno productivo basado en Linux, ya que simplifica la instalación de dependencias JavaScript frente a las propias de Microsoft en servidores Ubuntu.

La elección de .NET como framework backend permite al proyecto del tutor de aprendizaje combinar rendimiento, seguridad y mantenibilidad lo que lo hace ideal para el desarrollo de un MVP o Prototipo y por otro lado el aprovechamiento de la experiencia persona técnica previa con la herramienta, estos elementos son clave para alcanzar los objetivos establecidos en este trabajo de titulación.

2.3.4 BASE DE DATOS RELACIONAL (SQL)

La base de datos relacional (SQL) constituye la capa de persistencia de datos dentro del prototipo, esta es la encargada básicamente de almacenar, organizar y gestionar la información estructurada que genere y utilice el tutor de aprendizaje. En esta capa se almacena información crítica como los usuarios, sus intentos y progreso. Además, se almacenan los datos para parametrizar el tutor y las configuraciones generales del sistema que garantizan la integridad y coherencia de los datos en cada transacción.

Según Elmasri y Navathe (2020), una base de datos relacional es un conjunto de datos estructurados almacenados en tablas que están interrelacionadas. Cada tabla está compuesta por filas y columnas que representan registros y atributos en ese orden, respectivamente. Este modelo permite mantener la integridad de los datos mediante claves primarias y foráneas, y aplicar restricciones que aseguran la integridad referencial entre las distintas entidades.

La mayor ventaja de usar una BDD relacional radica en su capacidad para mantener relaciones lógicas entre los datos, esto es lo que facilita consultas complejas y actualizaciones coherentes sin que haya inconsistencias en la data. “Además, las bases de datos relacionales utilizan el Lenguaje de Consulta Estructurado (SQL, por sus siglas en inglés) como estándar para la manipulación y recuperación de información, garantizando interoperabilidad y compatibilidad con una amplia gama de tecnologías” (Date, 2019).

En el contexto del tutor de aprendizaje, la base de datos relacional cumple un papel esencial al permitir registrar la información de cada estudiante, sus ejercicios realizados, la retroalimentación recibida y su progreso en las distintas competencias lingüísticas del nivel A1. Esta estructura permite generar reportes y análisis sobre el avance académico, así como garantizar la seguridad e integridad de los datos almacenados.

2.3.4.1 HERRAMIENTAS

Antes de abordar las herramientas específicas, es necesario definir el concepto de un Sistema Gestor de Base de Datos SGBD o DBMS por sus siglas en inglés. “Un SGBD es un conjunto de programas que permiten la creación, administración y manipulación de bases de datos. Su función principal es actuar como intermediario entre el usuario o la aplicación y los datos almacenados, garantizando que las operaciones se realicen de manera segura, eficiente y consistente. Además, proporciona mecanismos para controlar el acceso, asegurar la integridad de los datos, gestionar transacciones y optimizar el rendimiento de las consultas” (Silberschatz, Korth, & Sudarshan, 2020).

Los SGBD relacionales son los más utilizados en aplicaciones, ya que soportan transacciones ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad), esto es lo que asegura la fiabilidad y credibilidad en la gestión de la data. En el caso del proyecto, el SGBD seleccionado debe integrarse de manera eficiente la base de datos que se seleccione para el tutor.

A continuación, se describen las herramientas candidatas seleccionadas para esta capa del proyecto propuesto:

- **PostgreSQL:** “PostgreSQL es un sistema gestor de bases de datos relacional de código abierto ampliamente reconocido por su solidez, estabilidad y apego a los estándares del lenguaje SQL. Su arquitectura robusta lo convierte en una opción ideal tanto para entornos empresariales como académicos, ya que ofrece un alto nivel de escalabilidad, soporte completo para transacciones ACID y la posibilidad de ampliar sus capacidades mediante módulos personalizados” (PostgreSQL Global Development Group, 2024). Al ser una herramienta open source el uso de PostgreSQL nos da un margen muy amplio en cuestión de escalabilidad y flexibilidad con respecto a las decisiones técnicas de desarrollo; ya que es compatible con cualquier tipo de servicio, lenguaje de programación o framework. Además, es más sencillo de implementar en un ambiente productivo.
- **Microsoft SQL Server:** “Microsoft SQL Server es un sistema gestor de bases de datos relacional desarrollado por Microsoft, diseñado para ofrecer un equilibrio óptimo entre rendimiento, seguridad e integridad de los datos. Su estrecha integración con el ecosistema de desarrollo de la compañía —especialmente con .NET y Entity Framework Core— lo convierte en una herramienta muy eficiente para entornos corporativos y académicos” (Microsoft, 2024). Una de las grandes ventajas de usar SQLServer es la perfecta integración que tiene con todos los servicios y suite de Microsoft, en este caso sería una decisión inteligente si se utiliza el Backend con .NET, gracias a la complementación que estos tienen.

2.3.4.2 COMPARACIÓN DE LAS HERRAMIENTAS

Tabla 4

Comparación de bases de datos relacionales

Criterio técnico	PostgreSQL	SQL Server
Licenciamiento y costo	Gratuito y de código abierto	Requiere licencia, versión Express gratuita con limitaciones
Compatibilidad con .NET	Compatible mediante drivers Npgsql y EF Core	Nativa
Compatibilidad multiplataforma	Windows, Linux y macOS	Principalmente Windows (pero disponible en Linux)
Herramientas administrativas	pgAdmin, DBeaver	SQL Server Management Studio
Comunidad y soporte	Amplia comunidad open source	Soporte empresarial por Microsoft

Nota: Comparativa técnica de RDBMS enfocada en integridad, escalabilidad y compatibilidad con .NET. Autor: Nicolás Rojas, 2025.

Ambas herramientas son opciones muy viables y robustas como bases de datos relacionales y cumplen los requerimientos del proyecto. Sin embargo, considerando la compatibilidad con el backend se selecciona Microsoft SQL Server como la base de datos relacional principal para el tutor de aprendizaje. Esta decisión se justifica por los siguientes motivos:

1. Integración nativa con el entorno .NET y herramientas de Microsoft.
2. Soporte empresarial y estabilidad, documentación bien planteada.
3. Herramientas de administración y monitoreo avanzadas y nativas del entorno propio de SQL Server, en este caso SSMS.

La elección de SQL Server como SGBD relacional refuerza la coherencia y selección del stack tecnológico del proyecto, esto porque se integra perfectamente con el backend en .NET y así garantizamos la integridad, consistencia y seguridad de los datos almacenados.

2.3.5 BASE DE DATOS NO RELACIONAL (NOSQL)

Las bases de datos no relacionales (NoSQL) son una alternativa moderna al modelo relacional tradicional al que estamos acostumbrados, este tipo de bases de datos están orientadas al almacenamiento y procesamiento de datos no estructurados o en todo caso semiestructurados. “A diferencia de las bases de datos SQL, que organizan la información en tablas con relaciones predefinidas, las bases de datos NoSQL almacenan los datos en estructuras más flexibles, como documentos, pares clave-valor, grafos o columnas distribuidas” (Han, Haihong, Le, & Du, 2011).

Para el presente proyecto, esta capa se emplea como un complemento a la base de datos relacional y al flujo general de las funcionalidades, esta actúa como una capa de persistencia adicional pero enfocado en persistir la información transaccional y dinámica del tutor de aprendizaje. Esto es útil porque aquí se almacenarán los historiales de conversación, los ejercicios generados, las respuestas y retroalimentaciones del modelo de IA; lo que corresponde a lo que llamamos un “Intento”.

Según Sadalage y Fowler (2012), el uso de bases de datos NoSQL resulta especialmente útil en aplicaciones que manejan grandes volúmenes de datos cambiantes o en las que la estructura de los datos puede variar con el tiempo, como los sistemas conversacionales. Su diseño flexible las hace idóneas para registrar interacciones dinámicas entre el usuario y el sistema, sin necesidad de definir esquemas rígidos.

2.3.5.1 HERRAMIENTAS

Antes de abordar las herramientas específicas, es necesario definir el concepto de un Sistema Gestor de Base de Datos No relacionales. Los sistemas gestores de bases de datos NoSQL son motores diseñados para almacenar, recuperar y administrar datos utilizando estructuras no tabulares. “A diferencia de los sistemas relacionales, estos gestores están orientados a escenarios donde se requiere flexibilidad estructural, alto rendimiento y capacidad para gestionar grandes volúmenes de datos distribuidos. Su arquitectura permite escalar horizontalmente mediante la distribución de datos en múltiples nodos, y soportan acceso de baja latencia, lo que los hace ideales para aplicaciones modernas con alta demanda de procesamiento” (Cattell, 2011).

Una de sus principales ventajas radica en que no ocupa un esquema rígido, y esto es lo que facilita la incorporación de nuevos atributos sin necesidad de modificar tablas o esquemas. Además, destacan por su capacidad para manejar eficientemente operaciones de lectura y escritura fuertes. Tienen además una muy buena integración con servicios backend modernos y su implementación es sencilla.

En el caso específico del tutor de aprendizaje, usar bases de datos no relacionales da una ventaja al momento de guardar la información no estructurada y transaccional de los intentos que puede llegar a ser muy variante en cuestión de atributos por el tema de la personalización propia del prototipo. A continuación, se describen dos opciones propuestas para el almacenamiento de datos no relacionales dentro del proyecto.

- **MongoDB:** “MongoDB es una base de datos NoSQL de tipo documental que utiliza estructuras JSON o BSON para representar la información. Esta característica le brinda una gran flexibilidad, ya que cada documento puede seguir un esquema diferente según las necesidades de los datos almacenados. Su arquitectura está diseñada para ofrecer alto rendimiento, escalabilidad horizontal mediante técnicas como sharding y una integración fluida con aplicaciones web modernas” (Chodorow, 2020). Ofrece soporte nativo para múltiples lenguajes de programación, como C#, Python, Java y JavaScript, así como compatibilidad con frameworks backend gracias a controladores oficiales como lo es MongoDB.Driver para .NET.

- **Firestore:** “Cloud Firestore, integrado en el ecosistema Firebase de Google, es una base de datos documental NoSQL basada en la nube. Su principal fortaleza reside en su capacidad para sincronizar información en tiempo real entre clientes y servidores, lo que la convierte en una alternativa atractiva para aplicaciones web y móviles que requieren actualizaciones instantáneas” (Google Firebase, 2024). Además, su esquema basado en colecciones y documentos comparte similitudes con MongoDB, lo que simplifica la organización de los datos. Firestore destaca por su escalabilidad automática y por tratarse de un servicio completamente administrado, reduciendo considerablemente el mantenimiento necesario por parte del desarrollador. Este como tal es más orientado a ser un servicio, que provee de más funcionalidades útiles pero que en ciertos casos no son del todo adaptables o accesibles.

2.3.5.2 COMPARACIÓN DE LAS HERRAMIENTAS

Tabla 5

Comparación de bases de datos NoSQL

Criterio técnico	MongoDB	Firestore
Modelo de datos	Documentos JSON/BSON	Documentos y colecciones JSON
Gestión de esquema	Dinámico, sin restricciones fijas	Dinámico, basado en colecciones
Integración backend	Compatible mediante controlador oficial MongoDB.Driver	SDK oficial de Firebase para .NET y JS
Búsquedas y agregaciones	Potente sistema de índices y consultas avanzadas	Limitado a consultas simples y compuestas
Sincronización en tiempo real	Posible mediante streams o triggers	Integrada nativamente
Costo	Gratuito (open source)	Modelo pago por uso.
Comunidad y soporte	Amplia comunidad open source	Soporte oficial de Google y comunidad

Nota: Comparativa orientada al almacenamiento flexible y escalabilidad horizontal. Autor: Nicolás Rojas, 2025.

Ambas herramientas son altamente compatibles con arquitecturas web modernas y ofrecen ventajas significativas. Sin embargo, por una cuestión de integración oficial y costos, se selecciona MongoDB como la base de datos no relacional para el tutor de aprendizaje. Esta elección se justifica por las siguientes razones:

1. Integración directa con .NET gracias a controladores oficiales.
2. Flexibilidad estructural para manejar intentos.
3. Capacidad de búsqueda avanzada, útil para devolver historial de intentos.
4. Escalabilidad y disponibilidad multiplataforma, además de ser gratuito.

En conclusión, la elección de MongoDB como base de datos no relacional fortalece el ecosistema técnico del tutor de aprendizaje

2.3.6 AUTENTICACIÓN Y GESTIÓN DE USUARIOS

La autenticación de usuarios constituye una funcionalidad esencial e indispensable en cualquier aplicación web moderna; ya que, garantiza la seguridad y el control de acceso al sistema como tal. Mediante la verificación de la identidad de los usuarios mantenemos a salvo los recursos y la información que sea sensible del sistema. Según Stallings (2018), la autenticación es el mecanismo mediante el cual un sistema valida que una entidad (usuario o servicio) es quien afirma ser, utilizando credenciales como contraseñas, certificados o tokens digitales.

En el contexto del tutor de aprendizaje, la plataforma de autenticación y gestión de usuarios cumple un papel clave dentro del flujo de interacción. Esta se encarga del registro, inicio de sesión y control de acceso de los usuarios. Además, tiene la particularidad de que el acceso estará restringido exclusivamente a las cuentas institucionales de la academia, es decir que las únicas cuentas aceptadas serán las que se autentican por medio del dominio verificado de la organización (@englishvalley.edu.ec).

La integración de este módulo consumiendo la api OAuth 2.0 de Google permite implementar un flujo de autenticación seguro y moderno, y verificado por una empresa internacional de la talla de Google. Esta autenticación está basada en tokens de acceso lo que permite tener un enfoque de backend stateless y restringir los recursos expuestos desde el backend solo para usuarios autenticados. “Este protocolo, ampliamente adoptado por las principales plataformas tecnológicas, proporciona un marco que posibilita que los usuarios se autentican mediante sus cuentas institucionales sin necesidad de almacenar credenciales directamente en el sistema” (Hardt, 2012).

Desde una perspectiva de arquitectura, el módulo de autenticación se comunica con el backend para validar los tokens emitidos por el proveedor y enlaza las credenciales con los registros almacenados en la tabla de Usuario. Este enfoque garantiza una gestión centralizada y segura de las sesiones, manteniendo la trazabilidad de los usuarios y su progreso académico.

2.3.6.1 HERRAMIENTAS

Antes de analizar las opciones específicas consideradas para el proyecto, es necesario que comprendamos el funcionamiento general de los servicios de autenticación externos. Este tipo de servicios consisten en una plataforma o conjunto de herramientas que fueron diseñadas para implementar mecanismos seguros de verificación de identidad dentro de las aplicaciones. “Estos sistemas se encargan de gestionar el almacenamiento de credenciales, la emisión de tokens de acceso, la recuperación de contraseñas y la autorización de permisos, utilizando estándares ampliamente adoptados como OAuth 2.0 y OpenID Connect” (Grassi, Garcia, & Fenton, 2020).

El uso de servicios de autenticación externos presenta importantes ventajas técnicas. En primer lugar, refuerzan la seguridad, ya que delegan el proceso de verificación a proveedores especializados con infraestructura robusta y actualizaciones constantes. Además, reducen las vulnerabilidades al evitar que las aplicaciones directamente se encarguen de gestionar contraseñas o datos sensibles, en las que muchas veces se cometen errores. Así se disminuyen los riesgos y brechas de seguridad de mantener la gestión de autenticación y seguridad desde el backend Finalmente, este tipo de servicios permiten el acceso con cuentas institucionales como Google Workspace o Microsoft 365, simplificando el proceso de inicio de sesión. A continuación, se describen dos opciones propuestas para desarrollar esta funcionalidad:

- **Google OAuth 2.0:** “Google OAuth 2.0 es un servicio de autenticación y autorización desarrollado por Google que implementa el estándar abierto OAuth 2.0. Su principal función es permitir que los usuarios accedan a aplicaciones externas utilizando sus credenciales de cuenta de Google, sin exponerlas directamente ante la aplicación que solicita acceso. Este sistema utiliza un flujo seguro basado en tokens de autenticación y actualización, garantizando la protección de las credenciales durante todo el proceso” (Google Identity Platform, 2024). Entre sus características más destacadas se encuentra el soporte para dominios verificados, que posibilita restringir el acceso únicamente a correos institucionales, por ejemplo, del dominio

@englishvalley.edu.ec. Además, su flujo seguro de autenticación mediante Access Tokens y Refresh Tokens asegura la validez temporal de las sesiones. Google OAuth 2.0 también es compatible con múltiples frameworks backend, incluidos ASP.NET Core y Node.js, y permite gestionar roles y permisos a través de claims incluidos en los JSON Web Tokens (JWT).

- **Firebase Authentication:** “Firebase Authentication, parte del ecosistema Firebase de Google, es un servicio de autenticación en la nube que ofrece una solución completa para la gestión de usuarios, credenciales y sesiones. Su diseño simplifica la implementación de mecanismos de acceso mediante correo electrónico y contraseña, número telefónico o proveedores externos como Google, Facebook y Apple” (Google Firebase, 2024). Entre sus principales ventajas se destaca su implementación rápida gracias a los SDKs oficiales disponibles para múltiples plataformas. Además, al ser un servicio se sincroniza de forma nativa con otros servicios del ecosistema Firebase, como Firestore y Cloud Functions, con este tipo de herramientas puedes prácticamente comenzar manejando el servicio de autenticación y extenderte hasta poder incluso manejar tu backend como servicio, lamentablemente este tipo de situaciones genera dependencia y suele truncar la escalabilidad en caso de querer usar otro stack tecnológico.

2.3.6.2 COMPARACIÓN DE LAS HERRAMIENTAS

Tabla 6

Comparación de servicios de autenticación

Criterio técnico	Google OAuth 2.0	Firebase Authentication
Tipo de servicio	Autenticación y autorización federada	Servicio completo de autenticación gestionada
Control de dominio institucional	Soporte nativo, dominios verificados de Google Workspace	Requiere validación manual o reglas personalizadas
Integración backend	SDKs y bibliotecas oficiales, amplia documentación	SDKs nativos, integración directa con Firebase
Gestión de roles y permisos	Configurable mediante JWT y claims	Configurable mediante reglas de seguridad y roles internos
Seguridad	Alto nivel, tokens firmados y cifrados	Alto nivel, control de sesiones y verificación de tokens
Costo	Gratuito, límites de uso generosos	Gratuito primero, modelo de pago por uso

Nota: Evaluación técnica de autenticación basada en estándares OAuth 2.0 y servicios gestionados. Autor: Nicolás Rojas, 2025.

Ambas herramientas ofrecen un nivel de seguridad elevado, soporte multiplataforma y facilidad de integración con el stack tecnológico del proyecto. Se define que la opción de autenticación será Google OAuth 2.0 por las siguientes justificaciones técnicas.

1. Permite limitar el acceso exclusivamente a usuarios con correos institucionales mediante la verificación del dominio en Google Workspace, alineándose con las políticas de seguridad de la academia.
2. Proporciona mayor control y personalización sobre los flujos de autenticación, permitiendo que el backend gestione la validación y asignación de roles.
3. Evita la dependencia total de servicios gestionados, manteniendo la independencia tecnológica del proyecto frente a plataformas cerradas.

En conclusión, la elección de Google OAuth 2.0 como plataforma de autenticación permite al tutor de aprendizaje garantizar la seguridad y la autenticidad institucional.

2.4 SELECCIÓN DE LA METODOLOGÍA ÁGIL

En el desarrollo de proyectos tecnológicos que integran componentes de inteligencia artificial, resulta esencial adoptar metodologías que permitan una planificación flexible, una entrega iterativa de resultados y la validación continua de los avances. Las metodologías ágiles surgen como un enfoque idóneo frente a los modelos tradicionales de desarrollo, ya que promueven la adaptabilidad, la comunicación constante y la mejora continua (Pressman & Maxim, 2020). Estas metodologías se sustentan en valores y principios definidos en el Manifiesto Ágil (Beedle, Beck, & Van Bennekum, 2001), el cual prioriza la colaboración con el cliente, la respuesta ante el cambio y la entrega continua de software funcional por encima de la documentación exhaustiva.

En el caso del tutor de aprendizaje propuesto, la naturaleza experimental del proyecto al combinar inteligencia artificial, procesamiento del lenguaje natural y objetivos pedagógicos requiere un marco de trabajo que facilite ajustes frecuentes y valide los resultados con los usuarios finales (en este caso, la dirección académica de la academia English Valley). Dado que el desarrollo será realizado por una sola persona, los roles tradicionales de las metodologías ágiles se adaptan a una estructura individual, en la que el autor asumirá los papeles de desarrollador, planificador y tester, mientras que la gerencia académica actuará como cliente o Product Owner.

En este contexto se han seleccionado dos metodologías ágiles que se adaptan y que tienen más afinidad con el proyecto propuesto, las cuales se detallarán a continuación.

2.4.1 SCRUM

Scrum es uno de los marcos de trabajo ágiles más utilizados en el desarrollo de software. Fue diseñado para gestionar proyectos complejos mediante ciclos iterativos denominados sprints, los cuales permiten entregar incrementos funcionales del producto en periodos cortos y controlados (Schwaber & Sutherland, 2020). En este enfoque, cada sprint tiene objetivos específicos y concluye con la entrega de una versión parcial del sistema que puede ser evaluada por los interesados. Entre sus características más relevantes se destacan:

- División del proyecto en iteraciones cortas, lo que facilita la planificación y control del progreso.
- Revisión y retroalimentación continua, garantizando que los entregables se alineen con los requerimientos del cliente.
- Flexibilidad para incorporar cambios, permitiendo adaptar las funcionalidades conforme se identifican nuevas necesidades.
- Gestión visual del flujo de trabajo, usualmente mediante tableros Kanban o herramientas digitales como Jira o Trello.

Scrum define tres roles principales: Product Owner, encargado de definir las prioridades; Scrum Master, facilitador del proceso; y Development Team, responsable de la implementación. En el contexto de este proyecto, el autor asume los tres roles técnicos (planificación, desarrollo y evaluación), mientras que la gerencia de English Valley cumple la función de cliente, brindando la validación funcional y pedagógica del prototipo.

La aplicación de Scrum en proyectos individuales ha demostrado ser efectiva, ya que su estructura permite mantener un ritmo de avance medible, documentar iteraciones y controlar versiones del sistema sin necesidad de equipos grandes (Rising & Janoff, 2000). En este caso, cada sprint abarcará fases de análisis, desarrollo, pruebas e integración progresiva del tutor de aprendizaje.

2.4.2 EXTREME PROGRAMMING (XP)

Extreme Programming (XP) es otra metodología ágil orientada a maximizar la calidad del software y responder de manera rápida a los cambios en los requerimientos del cliente. Su enfoque se basa en prácticas técnicas rigurosas, como la programación en pares, el desarrollo guiado por pruebas (TDD), la integración continua y la retroalimentación frecuente (Beck, 2004).

XP promueve un ciclo de desarrollo corto y la mejora continua del código mediante refactorización constante, priorizando la simplicidad y la comunicación directa con el cliente. Entre sus principios destacan:

- Simplicidad del diseño, evitando sobreingeniería.
- Liberaciones frecuentes, que aseguran la entrega temprana de valor.
- Pruebas automatizadas, que garantizan la calidad y evitan regresiones.
- Escucha activa al cliente, adaptando el sistema a sus necesidades reales.

No obstante, XP requiere una alta disciplina técnica y una colaboración permanente entre desarrollador y cliente, lo que puede representar una carga significativa en proyectos unipersonales, donde las prácticas colaborativas (como programación en pares o revisiones cruzadas de código) no son viables. Sin embargo, ciertos elementos de XP como el desarrollo iterativo, la refactorización y el enfoque en pruebas pueden incorporarse de manera complementaria dentro de un marco Scrum.

2.4.3 COMPARACIÓN DE LAS METODOLOGÍAS

Tabla 7

Comparación de metodologías ágiles

Criterio técnico	Scrum	Extreme Programming (XP)
Enfoque principal	Gestión iterativa del proyecto y entrega incremental	Mejora continua del código y calidad técnica
Duración de los ciclos	Sprints de 1 a 4 semanas	Iteraciones cortas (1-2 semanas)
Énfasis en roles	Product Owner, Scrum Master, Development Team	Cliente, Programador, Tester
Flexibilidad frente al cambio	Alta, mediante backlog priorizado	Muy alta, mediante retroalimentación constante
Prácticas clave	Planificación, revisión, retrospectiva, backlog	TDD, refactorización, integración continua
Adecuación a proyectos individuales	Alta: adaptable, simple de implementar	Media: requiere colaboración y revisiones externas
Enfoque en documentación	Moderado, orientado a resultados funcionales	Bajo, prioriza código sobre documentación
Madurez y adopción global	Muy alta, estándar industrial	Moderada, más técnica y especializada

Nota: Comparación de roles, iteraciones, prácticas técnicas y madurez metodológica.
 Autor: Nicolás Rojas, 2025.

Tras el análisis comparativo, se selecciona la metodología Scrum como la más adecuada para el desarrollo del tutor de aprendizaje basado en inteligencia artificial. Esta elección se fundamenta en los siguientes aspectos:

1. **Alineación con el contexto del proyecto:** al ser un proyecto académico individual, Scrum permite gestionar el tiempo y las entregas de forma flexible mediante sprints cortos y controlados.
2. **Validación progresiva con el cliente:** la academia (como Product Owner) podrá revisar avances en cada iteración y proponer mejoras según las necesidades pedagógicas detectadas.

3. **Integración de buenas prácticas técnicas:** aunque XP aporta un fuerte enfoque en la calidad del código, sus principios de integración continua y refactorización pueden incorporarse dentro del marco de Scrum, sin requerir una estructura colaborativa completa.
4. **Madurez y documentación:** Scrum cuenta con una extensa base de guías, estudios de caso y herramientas de soporte, lo que facilita su implementación y seguimiento en entornos académicos y educativos.
5. **Facilidad de implementación unipersonal:** su estructura puede adaptarse a un solo desarrollador, simplificando la planificación y la gestión de tareas sin comprometer la calidad del proceso.

En conclusión, Scrum será la metodología ágil utilizada para el desarrollo del proyecto, debido a su equilibrio entre control, adaptabilidad y enfoque en la entrega continua de valor. Su implementación garantizará que el tutor de aprendizaje evolucione de manera coherente con los requerimientos técnicos y pedagógicos de la academia English Valley, permitiendo una evaluación continua y un producto final alineado a los objetivos educativos y tecnológicos establecidos.

CAPÍTULO III: ANÁLISIS Y DISEÑO

3.1 CASOS DE USO

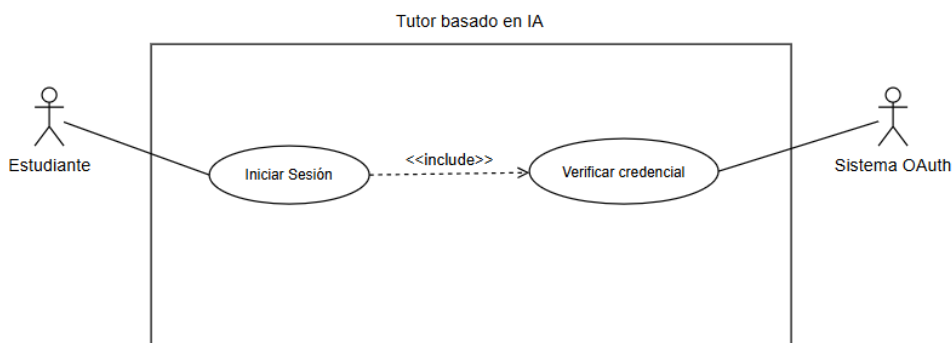
Los casos de uso son una herramienta fundamental dentro de la etapa de análisis y diseño de software, mediante estos diagramas modelamos las interacciones entre los actores, como pueden ser usuarios o sistemas externos con el sistema desarrollado. Los diagramas de casos de uso describen las funciones que debe realizar el sistema desde la perspectiva del usuario final. Para el prototipo propuesto, la formulación de los casos de uso nos permite identificar las funcionalidades principales del prototipo de manera más visible y basándonos en el proceso de toma de requerimientos y con el contexto que tenemos de lo que vamos a desarrollar. Por medio de esta fase lo que hacemos es plasmar de manera más formal lo previamente definido en la etapa de planificación con la información que teníamos disponible. De esta manera podemos ir estableciendo las relaciones que se producirán entre el estudiante y los componentes del prototipo.

El uso de esta técnica es esencial para garantizar que el prototipo cumpla con los requerimientos, ya que aquí planificamos una visión clara de las operaciones esperadas y de cómo los distintos módulos se integran para generar una experiencia coherente de aprendizaje. Así, los casos de uso se convierten en una guía para el desarrollo e implementación del prototipo. Además, de que sirven como base para las siguientes fases del proceso de análisis, como el planteamiento de requerimientos funcionales y no funcionales, la arquitectura y flujo del sistema, etc. A continuación, se presentan los tres casos de uso principales identificados para el sistema tutor de aprendizaje.

3.1.1 CASO DE USO 1 (CU1): INICIO DE SESIÓN EN EL SISTEMA

Figura 1

Caso de uso 1



Nota: Diagrama de caso de uso 1. Autor: Nicolás Rojas, 2025.

Actor: Usuario (Estudiante)

Propósito: Verificar la identidad del usuario mediante su cuenta institucional o credenciales registradas para acceder al sistema.

Flujo principal:

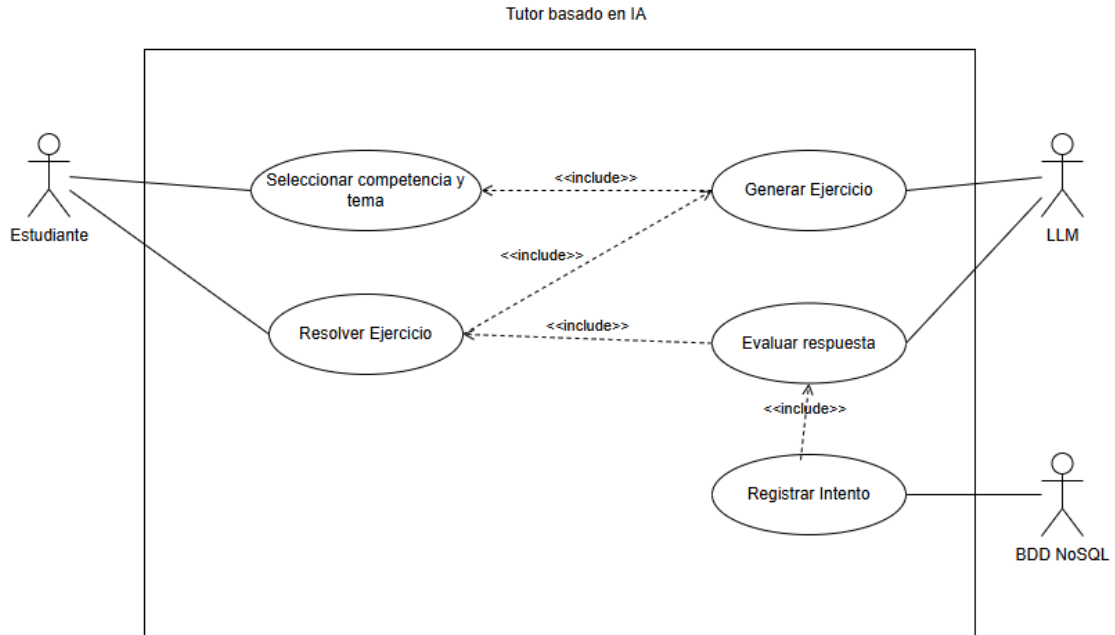
1. El usuario accede a la página principal del tutor.
2. Selecciona la opción “Iniciar sesión”.
3. El sistema presenta las opciones de autenticación (Google Workspace o usuario/contraseña).
4. El usuario se autentica.
5. El sistema valida las credenciales y verifica que el correo pertenezca a un dominio autorizado.
6. Si las credenciales son válidas, el usuario accede a la interfaz principal.

Resultado esperado: El usuario es autenticado exitosamente y obtiene acceso a su panel personal.

Excepciones: Si la autenticación falla, el sistema muestra un mensaje de error y no permite el ingreso.

3.1.2 CASO DE USO 2 (CU2): GENERAR Y RESOLVER EJERCICIO.

Figura 2
Caso de uso 2



Nota: Diagrama de caso de uso 2. Autor: Nicolás Rojas, 2025.

Actor: Usuario (Estudiante)

Propósito: Permitir al usuario seleccionar una habilidad y tema para generar un ejercicio adaptado al nivel A1 y recibir retroalimentación automática.

Flujo principal:

1. El usuario accede al módulo de práctica.
2. Selecciona la habilidad (por ejemplo, *Speaking* o *Writing*), la destreza y el tema deseado.
3. El sistema genera un ejercicio mediante la integración con el modelo de lenguaje (LLM).
4. El usuario responde el ejercicio o participa en la conversación.
5. El backend envía la respuesta del usuario al modelo de IA para su evaluación.

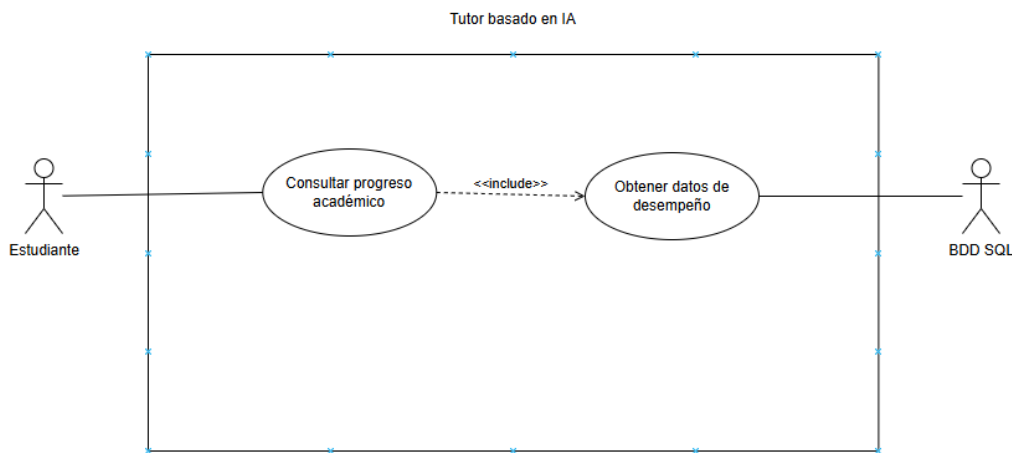
6. El sistema muestra la retroalimentación y calificación obtenida.
7. Se registra el intento en la base de datos NoSQL.

Resultado esperado: El usuario completa la actividad y recibe una evaluación automatizada con retroalimentación textual.

Excepciones: Si la actividad no se completa, se guarda la información ejecutada, pero se debe marcar como incompleto o abandonado.

3.1.3 CASO DE USO 3 (CU3): CONSULTAR PROGRESO ACADÉMICO

Figura 3
Caso de uso 3



Nota: Diagrama de caso de uso 3. Autor: Nicolás Rojas, 2025.

Actor: Usuario (Estudiante)

Propósito: Permitir al estudiante visualizar su avance global en cada habilidad y destreza trabajada.

Flujo principal:

1. El usuario accede al módulo “Mi progreso”.
2. El sistema recupera los registros de ejercicios realizados y sus calificaciones.

3. Calcula el nivel de progreso por habilidad y destreza.
4. Muestra un resumen gráfico o tabular con el avance académico.

Resultado esperado: El usuario visualiza su desempeño global y puede identificar sus áreas de mejora.

Excepciones: Si el usuario no ha completado ejercicios, el sistema muestra un mensaje informativo.

La modelación de los casos de uso permitió estructurar las funcionalidades esenciales del sistema, estableciendo las relaciones entre el usuario, el modelo de inteligencia artificial y los servicios de almacenamiento de datos. A través de estos diagramas se identificaron los flujos principales de interacción que guiarán el desarrollo técnico del prototipo, garantizando que las funcionalidades implementadas respondan directamente a los requerimientos funcionales definidos. Este proceso, además de servir como referencia para el diseño arquitectónico, proporciona una base sólida para validar el cumplimiento de los objetivos pedagógicos y tecnológicos del tutor de aprendizaje propuesto.

3.2 ANÁLISIS DE REQUERIMIENTOS DEL PROTOTIPO

En la presente sección nos enfocamos en continuar con el análisis y diseño del prototipo, con el propósito de transformar los objetivos generales y los casos de uso previamente definidos del proyecto en un conjunto de reglas formales que debe cumplir el sistema, a esto es a lo que llamamos “requerimientos”. En este sentido, se busca dar forma a una solución integral que resuelva de manera eficiente la problemática planteada para poder pasar del modelo mental de “lo que queremos que haga” a un conjunto de requerimientos formales que definan “cómo va a funcionar”. De esta manera vamos describiendo las funcionalidades principales, los requerimientos del sistema y la estructura arquitectónica que sustentará el desarrollo del prototipo del tutor de aprendizaje basado en inteligencia artificial.

En este sentido, el análisis de requerimientos constituye una parte fundamental del diseño, ya que define las funciones y características que debe cumplir el prototipo para responder a las necesidades del cliente, teniendo en mente los requerimientos pedagógicos y técnicos de *English Valley*. Debemos entender también que estos requerimientos se dividen en requerimientos funcionales, los cuales describen las operaciones concretas que se deben realizar. Y los no funcionales, que son los que establecen los atributos de calidad que garantizarán su rendimiento, seguridad y mantenibilidad.

3.2.1 REQUERIMIENTOS FUNCIONALES

Los requerimientos funcionales determinan las acciones específicas que el prototipo de sistema realizará. Para el prototipo del tutor de aprendizaje se identifican los siguientes:

Autenticación y Gestión de Usuarios

- RF-01. El sistema debe permitir el inicio y cierre de sesión mediante cuentas institucionales de Google Workspace, con la opción alterna de autenticación mediante usuario y contraseña.
- RF-02. El sistema debe permitir al usuario actualizar su información básica (nombre, idioma de la interfaz y modo visual) desde su perfil personal.

Configuración Académica

- RF-03. El sistema debe gestionar los niveles, habilidades, destrezas, contenidos estandarizados por MCER y también los temas que servirán como base para la generación de ejercicios.
- RF-04. Cada habilidad deberá vincularse con uno o varios contenidos pedagógicos (gramática, vocabulario o funciones comunicativas), así como con las destrezas lingüísticas correspondientes.

Práctica y Evaluación

- RF-05. El sistema debe permitir al usuario seleccionar una habilidad, destreza y tema para generar ejercicios personalizados de práctica.
- RF-06. El backend deberá generar y evaluar los ejercicios automáticamente mediante la integración con el modelo de lenguaje (LLM), incorporando los contenidos pedagógicos asociados.
- RF-07. El sistema debe registrar cada intento del usuario, incluyendo la calificación obtenida, el tiempo de ejecución y la retroalimentación proporcionada, almacenando esta información en la base de datos NoSQL.

Seguimiento y Reportes

- RF-08. El sistema debe calcular y almacenar el progreso del usuario en cada competencia (habilidad y destreza), con base en los ejercicios completados.

3.2.2 REQUERIMIENTOS NO FUNCIONALES

Los requerimientos no funcionales establecen las condiciones enfocadas a la calidad o puntos mínimos que el prototipo debe cumplir para asegurar una experiencia óptima, segura y estable durante su uso. Estos se agrupan en las siguientes categorías:

Usabilidad y Experiencia de Usuario

- RNF-01. La interfaz del sistema debe ser responsiva, intuitiva y accesible, garantizando su correcto funcionamiento en su enfoque de desarrollo web.

Seguridad y Privacidad

- RNF-02. Las credenciales de los usuarios deben gestionarse de manera segura, aplicando protocolos de autenticación o contraseñas cifradas. Es con el objetivo de proteger los datos personales y académicos de los usuarios conforme a los lineamientos de protección de datos personales vigentes y las políticas institucionales.

Rendimiento y Disponibilidad

- RNF-03. El tiempo de respuesta promedio del backend no debe superar los 60 segundos por cada solicitud, esto es para garantizar la fluidez en la interacción.

Arquitectura y Mantenibilidad

- RNF-04. El sistema debe implementarse bajo una arquitectura RESTful, con separación entre las capas de frontend, backend y bases de datos (relacional y NoSQL).

Los requerimientos funcionales y no funcionales definidos constituyen la base estructural del sistema, garantizando que el tutor de aprendizaje cumpla con los objetivos pedagógicos y técnicos planteados. Además, estos requerimientos servirán como punto de partida para el modelado de los casos de uso, los diagramas de arquitectura y el modelo de datos, asegurando coherencia entre la fase de análisis y el diseño técnico del sistema.

3.3 ARQUITECTURA DEL PROTOTIPO

El diseño de la arquitectura de un sistema corresponde a una de las etapas más importantes y definitorias dentro del proceso de desarrollo y en especial en la fase de análisis y diseño, esto es porque en esta etapa definimos la estructura general del software. Es decir, aquí es donde plasmamos como va a ser la forma en que los componentes van a interactuar entre sí para cumplir con los requerimientos funcionales y no funcionales que han sido planteados previamente en la sección 3.2. “Una arquitectura bien diseñada garantiza la coherencia del sistema, facilita su mantenimiento, mejora la eficiencia del procesamiento de datos y asegura la escalabilidad futura del producto” (Tu, 2023).

En el caso del prototipo del tutor basado en inteligencia artificial, el diseño arquitectónico cumple un rol fundamental al ser el punto de integración entre los componentes analizados previamente en la *Fundamentación Técnica*. Esta arquitectura tiene como objetivo principal establecer una comunicación fluida entre la interfaz de usuario, el backend, los servicios de inteligencia artificial y los sistemas de almacenamiento de datos, garantizando su correcto funcionamiento.

3.3.1 ANÁLISIS DE LA ARQUITECTURA DEL PROTOTIPO

El análisis de la arquitectura permite identificar cómo se organizarán y como se interconectarán los componentes del prototipo, determinando el flujo de trabajo ideal en el cual se muestre la comunicación entre las capas del sistema. En este sentido, el funcionamiento del tutor se percibe directamente con un enfoque modular, priorizando que los componentes tengan “independencia” en su funcionamiento, esto principalmente para facilitar la mantenibilidad y la escalabilidad del software.

El flujo de interacción inicia cuando el estudiante accede a la interfaz web del tutor; es decir, esta es la “punta de la lanza”. En primera instancia, el usuario se autentica utilizando su cuenta institucional provista por la academia, se validan sus credenciales antes de permitir el acceso. Una vez autenticado y dentro de la interfaz, el usuario puede seleccionar una habilidad, una destreza y un tema de práctica desde el FrontEnd, que constituye la capa de presentación del sistema.

La solicitud generada se envía al BackEnd, el cual cumple el rol de ser la capa de lógica del prototipo. El componente perteneciente a la capa de lógica es lo “central” y lo que definiríamos como el núcleo del sistema. Aquí es donde se coordinan las operaciones principales, como la comunicación con el modelo de Inteligencia Artificial (LLM API) para la generación de ejercicios, la evaluación automática de las respuestas y la gestión de los datos provenientes de la interacción del usuario, básicamente se encarga de orquestar y mantener el flujo funcional para que el prototipo haga lo que debe hacer. Posteriormente, los resultados obtenidos se almacenan en dos tipos de bases de datos:

- Una base de datos relacional, la cual es utilizada para mantener la información estructurada (usuarios, niveles, habilidades, destrezas, etc)
- Una base de datos NoSQL, la cual se emplea para almacenar los detalles de los intentos de práctica, resultados de evaluación y retroalimentación generada dinámicamente por la IA.

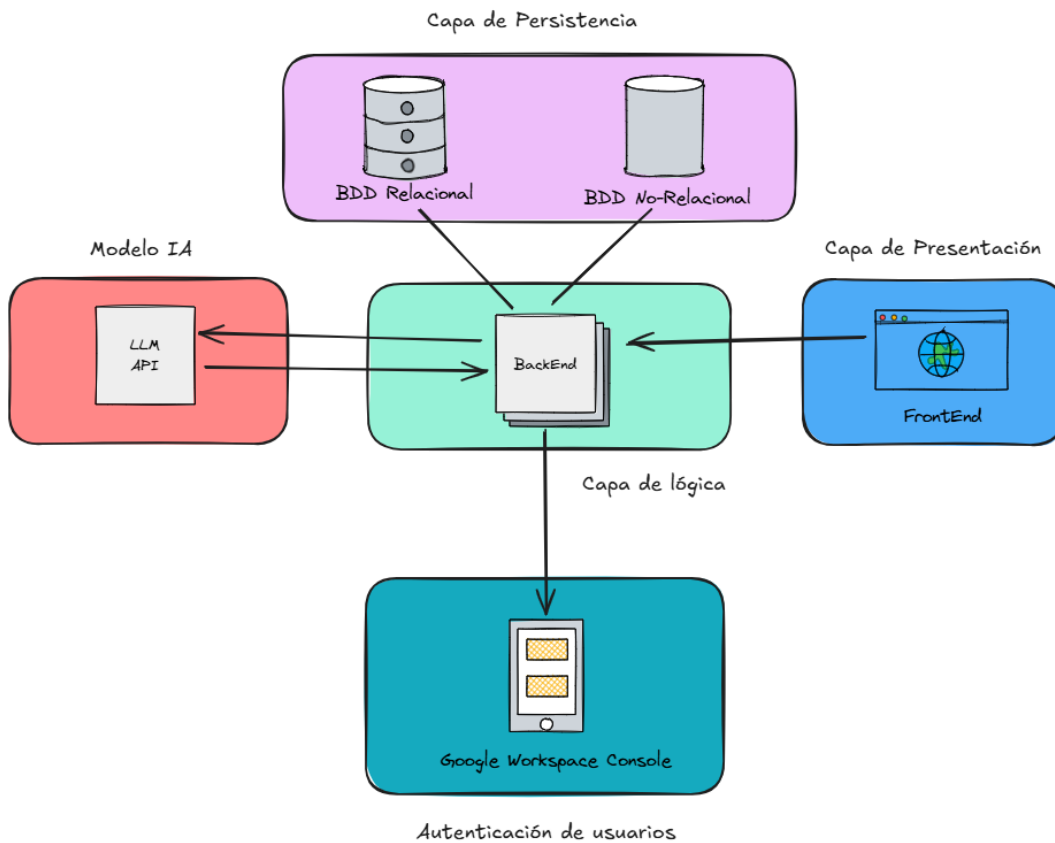
Esta arquitectura en capas garantiza que cada módulo funcione de forma independiente, pero coordinada, asegurando la trazabilidad y consistencia de los datos durante todo el proceso de interacción entre el estudiante y el sistema.

3.3.2 DISEÑO DE LA ARQUITECTURA DEL PROTOTIPO

El diseño arquitectónico adoptado para el prototipo se basa en una arquitectura enfocada a múltiples capas, que organizan los componentes del prototipo. Esta estructura favorece el funcionamiento individual de los módulos, también permite la reutilización de componentes en otras integraciones y da la posibilidad de realizar mantenimientos o mejoras sin afectar la integridad general del sistema (Söylemez, Tekinerdogan, & Kolukisa Tarhan, 2022).

Figura 4

Diagrama de arquitectura del prototipo



Nota: Diagrama de arquitectura del prototipo de tutor con inteligencia artificial. Autor: Nicolás Rojas, 2025.

En la Figura 4 se presenta el diagrama de arquitectura del prototipo a desarrollar. El diseño propuesto consta de 5 capas, cada una con una responsabilidad propia y bien definida como se detallará ahora. La capa de presentación está representada por el componente del FrontEnd que está encargado de proporcionar la interfaz gráfica y las funcionalidades de interacción con el usuario. La capa de lógica se encuentra compuesta por el componente del BackEnd, este es el que actúa como “orquestador” de las distintas capas y componentes, esta capa asume todo el proceso operativo y se encarga de hacer posible que las funcionalidades trabajen correctamente. En ese sentido se encarga de gestionar las peticiones del usuario, el flujo de datos y la comunicación con el modelo de inteligencia artificial, los servicios de autenticación y las bases de datos de la capa de persistencia.

La capa de persistencia se conforma por una base de datos relacional y una base de datos no relacional, las cuales se encargan de garantizar el correcto almacenamiento, procesamiento y distribución de los datos tanto a lo correspondiente de almacenamiento estructurado de la información académica, como también la flexibilidad en el manejo de registros dinámicos de ejercicios y retroalimentaciones a manera de log. Finalmente, se integran dos servicios externos que son esenciales. El primero es el modelo de IA (LLM API), que es responsable de la generación, evaluación y retroalimentación de los ejercicios. El segundo es el servicio de autenticación que se relaciona a la consola de Google Workspace de la academia, que gestiona el acceso seguro de los usuarios generados bajo el dominio de la academia al sistema.

Esta distribución permite que cada componente se comunique mediante servicios bien definidos, utilizando una arquitectura RESTful para la gestión del BackEnd, lo que proporciona una comunicación ligera y estandarizada entre las capas del sistema. Gracias a esta organización, el prototipo mantiene una alta cohesión interna y un bajo acoplamiento, lo que facilita la escalabilidad y futuras integraciones con nuevos servicios, modelos de inteligencia artificial o incluso interfaces (Söylemez, Tekinerdogan, & Kolukisa Tarhan, 2022).

3.4 MODELADO DE DATOS

La etapa del modelado de datos constituye un componente esencial dentro del diseño del sistema; ya que, define la estructura lógica mediante la cual se almacenará, organizará y relacionará la información que sea propia del prototipo. Una correcta definición del modelo de datos garantiza la coherencia y trazabilidad de la información, así como la eficiencia en las consultas y operaciones de lectura o escritura realizadas durante la ejecución del sistema (Connolly & Begg, 2021).

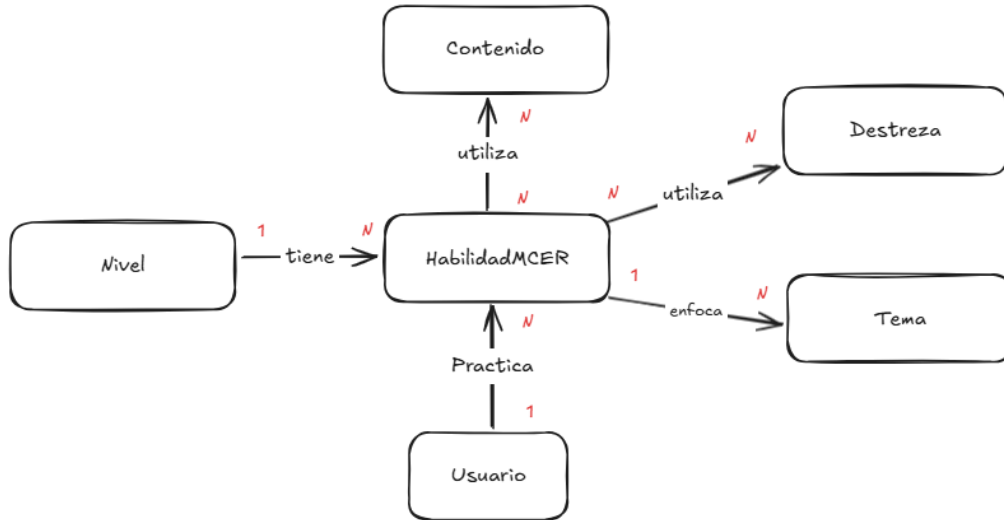
Una vez identificados los requerimientos funcionales, no funcionales y las especificaciones de la problemática que se debe solucionar se siguió un proceso para poder definir, en un primer avance, un pseudo modelo entidad relación con las entidades y sus relaciones. En concreto para el desarrollo del prototipo del tutor se debe tomar en cuenta el manejo de datos académicos estandarizados; como lo son los niveles, las habilidades, contenidos y destrezas. Para esto, principalmente se tomaron en cuenta las definiciones formales que provee el MCER, con base a este gran punto de referencia se identificaron dichas primeras entidades y se sintetizó la información que manejaría cada una y en base a esto su funcionalidad.

Otro punto crucial que se debe tomar en cuenta es el manejo transaccional de la funcionalidad principal del prototipo. Aquí se identifican las entidades como usuarios, intentos de práctica, temas de preferencia para dar una experiencia personalizada de aprendizaje. En este punto se debe tomar en cuenta la gestión de datos dinámicos o no estructurados que son resultado de las ejecuciones de la funcionalidad principal del prototipo, esto refiere a los intentos, ejercicios, respuestas, calificaciones y retroalimentaciones generadas por la IA y que se guardarán en el esquema no relacional.

3.4.1 MODELO DE DOMINIO

Figura 5

Diagrama de modelo de dominio

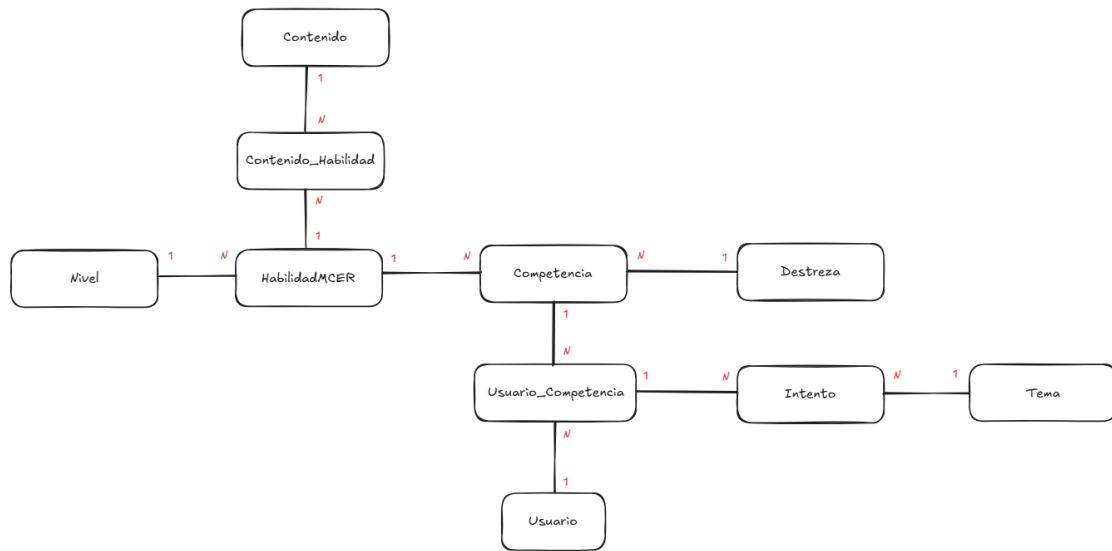


Nota: Diagrama de modelo de dominio del prototipo de tutor con inteligencia artificial.
Autor: Nicolás Rojas, 2025.

El modelo de dominio representado en la Figura 5 muestra las entidades principales y las relaciones entre los componentes pedagógicos y operativos del proceso que se busca sistematizar. Este modelo representa la abstracción de la funcionalidad del sistema enfocado al proceso académico de la academia. A partir de este modelo, se identifican entidades fuertes y partir de eso generamos los conceptos claves para el funcionamiento del prototipo y que se expanden en el diagrama entidad relación que se muestra a continuación.

3.4.2 MODELO CONCEPTUAL

Figura 6
Diagrama de modelo conceptual



Nota: Diagrama de modelo conceptual del prototipo de tutor con inteligencia artificial.
Autor: Nicolás Rojas, 2025.

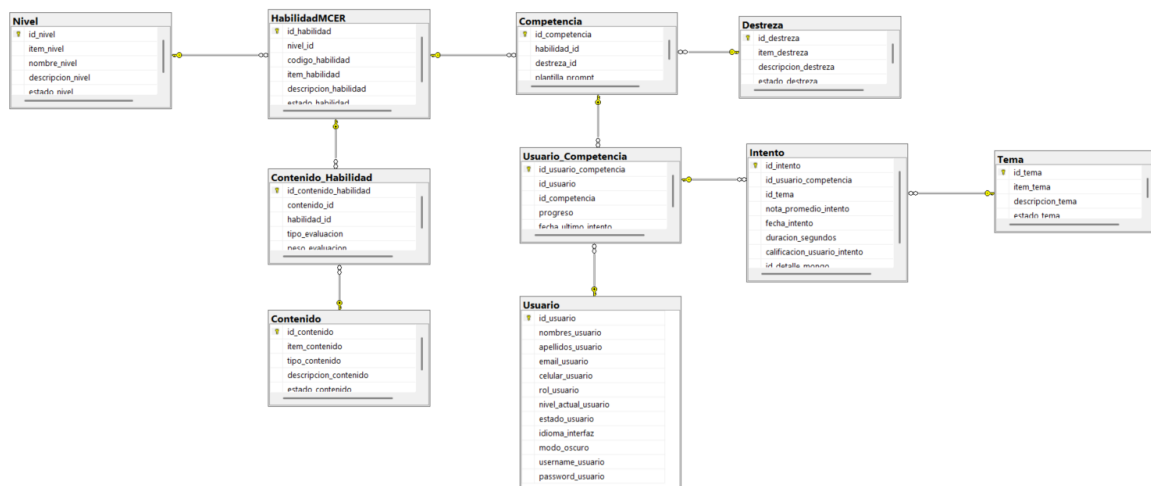
En el presente modelo conceptual, comenzamos a enfocar los conceptos resultantes del modelo de dominio a tablas o entidades. Es así como llegamos a identificar tanto entidades fuertes como entidades débiles (tablas intersección), añadiendo entidades como “Competencia” o “Intento”. Es así que podemos ver que la entidad *Usuario* se relaciona con *Competencia* a través de la tabla intermedia *Usuario_Compentencia* que permite almacenar el desempeño individual por habilidad. Cada *Competencia* está asociada a una *HabilidadMCER* vinculada a un Nivel, siguiendo los descriptores pedagógicos del Marco Común Europeo de Referencia (MCER).

En este sentido, las entidades *Contenido*, *HabilidadMCER* y *Contenido_Habilidad* estructuran los catálogos que definen los temas y ejercicios de práctica, mientras que la entidad *Intento* almacena los resultados de cada interacción del estudiante con el tutor, incluyendo la relación con *Tema* y *Destreza*. Este modelo fue diseñado para asegurar la consistencia referencial, evitando redundancias y permitiendo trazabilidad completa del progreso del usuario (Elmasri & Navathe, 2020).

Durante el proceso de diseño se presentaron dos problemáticas principales: la necesidad de integrar datos estructurados y no estructurados y la flexibilidad para almacenar múltiples intentos por usuario. La primera se resolvió adoptando un enfoque dual (SQL + NoSQL) que permite mantener información académica formal en un entorno relacional y, al mismo tiempo, registrar la información dinámica generada por la IA sin restricciones de esquema. La segunda se solucionó mediante la entidad Intento, que actúa como contenedor de los resultados y retroalimentaciones de cada sesión de práctica, facilitando la evaluación y análisis de desempeño individual.

3.4.3 MODELO LÓGICO

Figura 7
Diagrama de modelo lógico



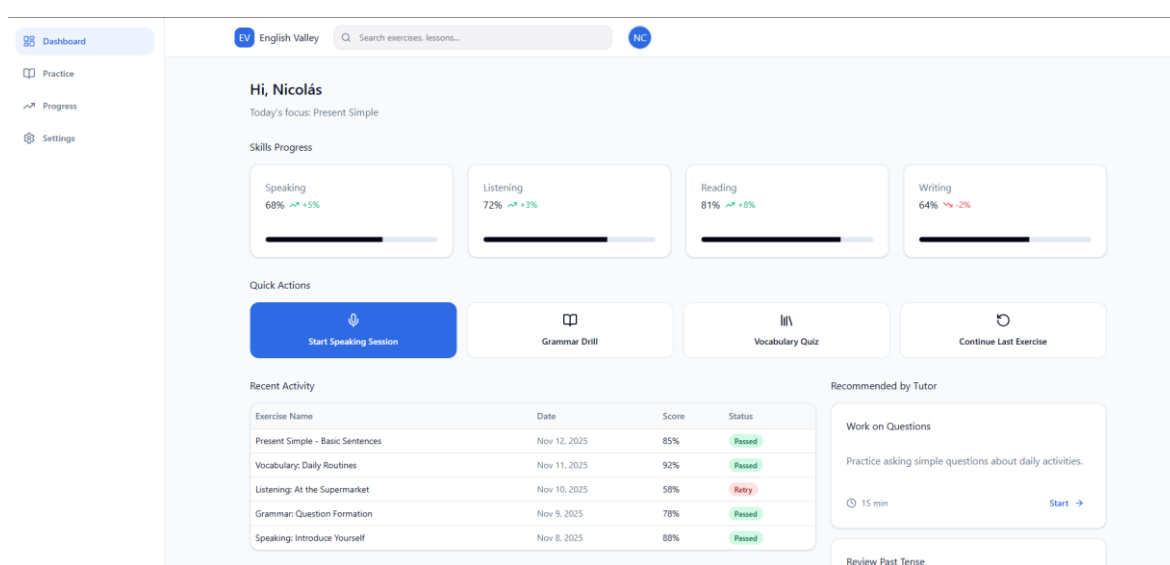
Nota: Diagrama de modelo lógico del prototipo de tutor con inteligencia artificial. Autor: Nicolás Rojas, 2025.

Posteriormente se procedió a identificar los atributos, claves primarias, foráneas y revisar las relaciones para cumplir con las formas normales de una base de datos, se procedió a la construcción del modelo lógico. Este paso es crucial para identificar los campos que se asociarán en cada entidad, esto está directamente relacionado con las funcionalidades del prototipo.

3.5 DISEÑO DE LA INTERFAZ

El diseño de la interfaz del prototipo constituye un componente clave para garantizar una experiencia de usuario intuitiva, coherente y atractiva. En este proyecto, la interfaz se desarrolló siguiendo principios de usabilidad, consistencia visual y accesibilidad, asegurando que el estudiante pueda interactuar con el tutor de aprendizaje de manera fluida y sin distracciones.

Figura 9
Interfaz de ejemplo



Nota: Diseño de interfaz del prototipo de tutor con inteligencia artificial. Autor: Nicolás Rojas, 2025.

El diseño sigue una línea gráfica limpia y moderna, basada en la paleta cromática institucional de la academia (*primary #2D6AE3, text #0F172A, surface white*), con sombras suaves y esquinas redondeadas de 12 px para transmitir una sensación de ligereza y profesionalismo. El color primario se utiliza únicamente en los llamados a la acción principales (CTA) para mantener jerarquía visual y evitar sobrecarga cromática (Tidwell, Brewer, & Valencia, 2020).

Este enfoque visual y funcional permite que el estudiante se centre en la práctica del idioma, con una interfaz minimalista que favorece la interacción, el seguimiento del progreso y la motivación continua. Además, su estructura modular facilita la integración de nuevas funcionalidades o niveles del MCER en futuras versiones del sistema, manteniendo coherencia con la arquitectura general del prototipo.

CAPÍTULO IV: DESARROLLO E IMPLEMENTACIÓN

4.1 PREPARACIÓN DEL ENTORNO DE DESARROLLO

Para el desarrollo del prototipo del tutor de aprendizaje se estableció un entorno de trabajo que integra herramientas especializadas para cada uno de los componentes del sistema: base de datos relacional, backend, frontend y base de datos no relacional. La selección de dichas herramientas se fundamenta en los criterios técnicos y de compatibilidad definidos en el Capítulo II, garantizando coherencia con la arquitectura propuesta y facilitando la implementación modular del sistema.

4.1.1 SQL SERVER MANAGEMENT STUDIO (SSMS)

“SQL Server Management Studio es un entorno integrado de Microsoft que permite administrar y desarrollar bases de datos SQL Server de forma visual, facilitando la ejecución de consultas, la creación de tablas y la gestión de relaciones entre entidades” (Microsoft, 2025). Su uso en este proyecto responde a la necesidad de implementar la base de datos relacional definida para el sistema, asegurando la integridad de los datos estructurados (usuarios, competencias, niveles, etc.) mediante la ejecución del script de creación del modelo físico.

4.1.2 VISUAL STUDIO (.NET FRAMEWORK)

“Visual Studio es un entorno de desarrollo integrado (IDE) desarrollado por Microsoft que permite compilar, depurar e implementar aplicaciones basadas en el framework .NET” (Microsoft, 2024). Su selección se justifica por la necesidad de implementar el backend del prototipo, desarrollado en ASP.NET Core, que actúa como la capa de lógica del sistema, gestionando la comunicación entre el frontend, las bases de datos y la API de inteligencia artificial. La compatibilidad nativa entre Visual Studio, .NET y SQL Server optimiza el flujo de desarrollo y depuración del proyecto.

4.1.3 VISUAL STUDIO CODE (REACT)

“Visual Studio Code es un editor de código multiplataforma desarrollado por Microsoft, que ofrece soporte para múltiples lenguajes de programación y frameworks modernos como React” (Microsoft, s.f.). En el presente proyecto se utiliza para el desarrollo del frontend del tutor de aprendizaje, encargado de la interfaz web con la que interactúan los usuarios. La elección de VS Code responde a su flexibilidad, su integración con librerías de React y su capacidad de gestionar proyectos web modulares mediante control de versiones, extensiones y herramientas de desarrollo de interfaz.

4.1.4 MONGODB COMPASS

MongoDB Compass es la interfaz gráfica oficial para la gestión de bases de datos MongoDB, permitiendo visualizar, consultar y analizar colecciones de documentos sin necesidad de comandos de consola (MongoDB Inc, s.f.). En este proyecto se emplea para la administración del almacenamiento no relacional, donde se guardan los historiales de conversación, retroalimentaciones y registros dinámicos generados por el modelo de IA. Su uso garantiza una manipulación eficiente y visual de los datos no estructurados, complementando la persistencia relacional del sistema.

4.2 DESARROLLO DE LAS FUNCIONALIDADES

El desarrollo del prototipo del tutor de aprendizaje se llevó a cabo siguiendo la metodología ágil Scrum, estructurada en tres iteraciones o sprints. Cada sprint se planificó con un conjunto definido de objetivos, actividades y entregables, priorizando el avance incremental y verificable del sistema.

Este enfoque permitió validar los componentes del prototipo en etapas, facilitando la detección temprana de mejoras y asegurando la integración coherente entre los módulos del sistema: base de datos, backend y frontend.

4.2.1 SPRINT 1 (SP1): IMPLEMENTACIÓN Y VALIDACIÓN DE LAS BASES DE DATOS

4.2.1.1 OBJETIVO:

Diseñar, implementar y validar las bases de datos del sistema relacional y no relacional a partir del modelo físico definido en el capítulo anterior, garantizando la integridad, coherencia y disponibilidad de la información estructurada que servirá como soporte para el funcionamiento del proyecto.

4.2.1.2 ACTIVIDADES

- Conversión del modelo lógico a modelo físico, verificando la correcta normalización y las relaciones entre entidades.
- Implementación del script de creación de tablas en Microsoft SQL Server Management Studio (SSMS), según la estructura establecida en el archivo con el script de la base de datos.
- Registro de datos de prueba en las entidades principales (Nivel, HabilidadMCER, Contenido, Competencia y Usuario) para validar integridad referencial y funcionamiento de claves foráneas.
- Ejecución de consultas SQL para comprobar la relación entre las tablas, el rendimiento de las búsquedas y la consistencia de los datos.

- Identificación y ajuste de posibles mejoras en el modelo físico, tales como la inclusión de atributos complementarios o restricciones adicionales para optimizar la estructura.
- Configuración inicial de la base de datos MongoDB mediante la herramienta MongoDB Compass, creando las colecciones destinadas al almacenamiento no estructurado (por ejemplo: intentos, respuestas y retroalimentaciones).
- Pruebas de conexión y validación entre el entorno de desarrollo local y los servicios de base de datos, verificando la comunicación con el backend a través de drivers y controladores nativos de .NET y MongoDB.

4.2.1.3 RESULTADOS ESPERADOS

- Base de datos relacional TutorIA_EV creada correctamente con todas las entidades, relaciones y restricciones funcionales.
- Datos de prueba insertados y validados para garantizar la consistencia y completitud del modelo.
- Configuración inicial del entorno NoSQL en MongoDB lista para almacenar datos dinámicos del sistema.

4.2.1.4 EVIDENCIAS

Figura 10

Evidencia 1 - Sprint 1

	TABLE_SCHEMA	TABLE_NAME
1	dbo	Competencia
2	dbo	Contenido
3	dbo	Contenido_Habilidad
4	dbo	Destreza
5	dbo	HabilidadMCER
6	dbo	Intento
7	dbo	Nivel
8	dbo	sysdiagrams
9	dbo	Tema
10	dbo	Usuario
11	dbo	Usuario_Compentencia

Nota: Evidencia de la creación de las tablas en el DBMS seleccionado. Autor: Nicolás Rojas, 2025.

Figura 11
Evidencia 2 - Sprint 1

id_nivel	item_nivel	nombre_nivel	descripcion_nivel	estado_nivel
1	A1	Beginner	Nivel introductorio de usuario básico según el MC...	Activo

id_habilidad	nivel_id	codigo_habilidad	item_habilidad	descripcion_habilidad	estado_habilidad
1	1	A1-01	Introducirse a si mismo y a otros	Decir su nombre, país, ocupación y saludos básicos.	Activo
2	1	A1-02	Hacer y responder preguntas personales	Formular preguntas sobre familia, edad, dirección ...	Activo
3	1	A1-03	Usar saludos y expresiones sociales básicas	Emplear frases comunes como Hello, Thanks, Sorr...	Activo
4	1	A1-04	Describir rutinas y actividades diarias	Expresar acciones cotidianas con Present Simple.	Activo
5	1	A1-05	Expresar emociones y estados físicos	Decir cómo se siente o describir su estado físico (f...	Activo

id_contenido	item_contenido	tipo_contenido	descripcion_contenido	estado_contenido
1	Present simple	Gramática	Uso afirmativo, negativo e interrogativo para hab...	Activo
2	Verbo to be (am, is, are)	Gramática	Estructuras básicas para presentarse y describir...	Activo
3	Vocabulary: family and friends	Vocabulario	Palabras y frases para hablar sobre la familia y ...	Activo
4	Vocabulary: daily routines	Vocabulario	Vocabulario sobre actividades diarias y hábitos.	Activo
5	Expressing likes and dislikes	Función comunicativa	Uso de I like / I don't like para expresar preferen...	Activo

Nota: Evidencia del ingreso de los datos correctamente en las tablas fuertes. Autor: Nicolás Rojas, 2025.

Figura 12
Evidencia 3 - Sprint 1

The screenshot shows the MongoDB Compass interface. On the left, the 'Connections' panel shows a connection to 'Docker_mongo_tutorEv' with a database named 'TutorIA_EV_NoSQL' and a collection named 'intentos_detalle'. The main area displays the 'Documents' tab for this collection, showing two JSON documents. The first document has a state of 'INICIADO' and the second has a state of 'FINALIZADO'. Both documents include fields for 'sql', 'nivel', 'estado', 'createdAt', 'updatedAt', 'items', 'evaluacionFinal', and 'llm'.

Nota: Evidencia de configuración de BDD NoSQL e ingreso de datos de prueba. Autor: Nicolás Rojas, 2025.

4.2.1.5 TABLA DE TERMINACIÓN

Tabla 8

Tabla de terminación - Sprint 1

Actividad	Estado	Fecha	Observaciones
Implementación de base de datos relacional SQL Server	Completada	15/11/2025	Se debe añadir un campo “estado” en Intento.
Ingreso de datos de prueba y pruebas de consultas	Completada	15/11/2025	Sin observaciones.
Creación de colecciones NoSQL (MongoDB)	Completada	20/11/2025	Sin observaciones.
Pruebas de conexión con backend (preliminar)	Completada	20/11/2025	Sin observaciones.

Nota: Tabla de terminación referente al componente de la capa de Persistencia (Base de datos SQL y NoSQL). Autor: Nicolás Rojas, 2025.

4.2.2 SPRINT 2 (SP2): DESARROLLO DEL BACKEND Y CONEXIÓN CON SERVICIOS EXTERNOS (LLM Y AUTENTICACIÓN)

4.2.2.1 OBJETIVO

Implementar la capa de lógica del sistema mediante el desarrollo del backend en ASP.NET Core, estableciendo la comunicación entre la base de datos relacional, la base de datos NoSQL, el servicio de autenticación y la API del modelo de lenguaje (LLM). Este sprint busca consolidar la estructura funcional del sistema, garantizando la integridad de los procesos internos y la interoperabilidad entre los distintos componentes tecnológicos.

4.2.2.2 ACTIVIDADES

- Configuración del proyecto BackEnd utilizando ASP.NET Core Web API, estableciendo la estructura por capas.
- Implementación de la conexión a la base de datos relacional SQL Server mediante Entity Framework, generación de modelos y archivos de Datos con funciones de manipulación de datos estándar (CRUD).
- Integración de la base de datos NoSQL MongoDB para el almacenamiento de información no estructurada relacionada con intentos.
- Desarrollo del módulo de autenticación y autorización mediante el servicio de OAuth, incluyendo la generación y validación de tokens JWT para la gestión de sesiones seguras.
- Desarrollo de la capa lógica con servicios de negocio para la gestión de usuarios, flujo de Intentos (Start, Answer, Finish), reportería y catálogos.
- Integración del servicio externo (API) del modelo de inteligencia artificial, definiendo la estructura de prompts y la parametrización basada en el contexto académico.
- Exposición de endpoints REST para el consumo de datos desde el FrontEnd, validando parámetros, control de errores y respuestas estandarizadas.
- Ejecución de pruebas funcionales de los endpoints mediante herramientas como Swagger y Postman, verificando la correcta comunicación entre capas y la integridad de la información procesada.

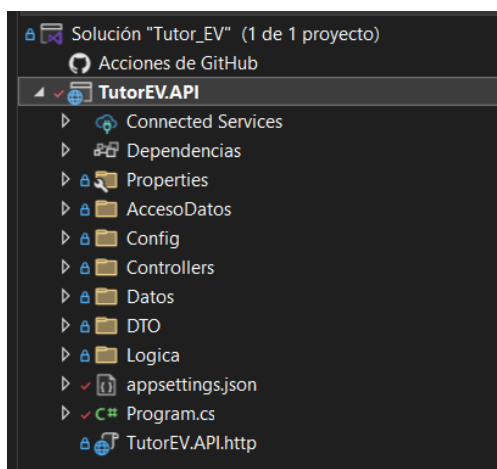
4.2.2.3 RESULTADOS ESPERADOS

- BackEnd funcional con API REST implementada y operativa.
- Conexión establecida entre el BackEnd y las bases de datos SQL Server y MongoDB.
- Integración exitosa con el servicio de autenticación de Google y la API de OpenAI.
- Validación de la lógica de negocio y del flujo de datos entre las capas del sistema.

4.2.2.4 EVIDENCIAS

Figura 13

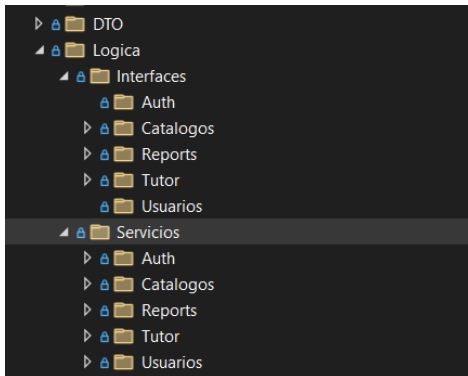
Evidencia 1 - Sprint 2



Nota: Evidencia de la creación del proyecto y la estructura del BackEnd. Autor: Nicolás Rojas, 2025.

Figura 14

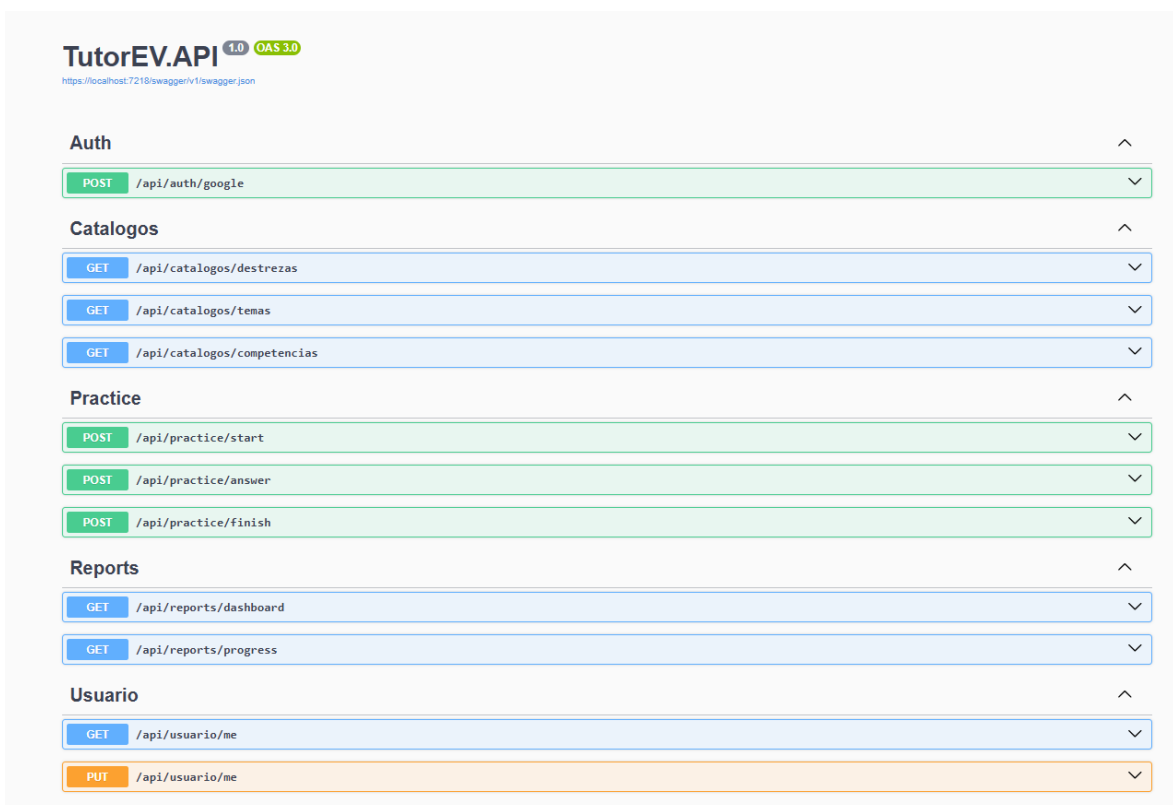
Evidencia 2 - Sprint 2



Nota: Evidencia de la creación de la capa de Lógica de negocio con Interfaces y Servicios de las funcionalidades principales del proyecto. Autor: Nicolás Rojas, 2025.

Figura 15

Evidencia 3 – Sprint 2



Nota: Evidencia de la exposición de los endpoints finales en swagger. Autor: Nicolás Rojas, 2025.

4.2.2.5 TABLA DE TERMINACIÓN

Tabla 9

Tabla de terminación - Sprint 2

Actividad	Estado	Fecha	Observaciones
Configuración del proyecto backend y arquitectura por capas	Completada	30/11/2025	Sin observaciones.
Integración con bases de datos SQL y MongoDB. Capa de Acceso a datos y Datos.	Completada	05/11/2025	Sin observaciones.
Implementación de autenticación con Google y JWT	Completada	11/12/2025	Sin observaciones.
Integración de servicio de OpenAI y servicios de la capa de lógica de negocio.	Completada	16/12/2025	Sin observaciones.

Nota: Tabla de terminación referente al componente de BackEnd. Autor: Nicolás Rojas, 2025.

4.2.3 SPRINT 3 (SP3): DESARROLLO DE LA INTERFAZ WEB

4.2.3.1 OBJETIVO

Diseñar e implementar la interfaz web del tutor de aprendizaje utilizando React, integrando los servicios desarrollados en el BackEnd y asegurando una experiencia de usuario intuitiva, accesible y coherente con los lineamientos visuales definidos en el diseño de la interfaz. Este sprint busca consolidar la interacción completa del usuario con el sistema y verificar el correcto funcionamiento de las funcionalidades desde la capa de presentación.

4.2.3.2 ACTIVIDADES

- Configuración del proyecto FrontEnd utilizando tecnologías modernas de desarrollo web, estableciendo la estructura base de componentes y servicios.
- Diseño e implementación de las pantallas principales del sistema, incluyendo autenticación, selección de destrezas, módulos de práctica y visualización de progreso.
- Integración del FrontEnd con la API REST del BackEnd, consumiendo los endpoints para autenticación, obtención de datos académicos y registro de intentos.
- Implementación de manejo de estado y control de sesiones mediante el uso de tokens JWT almacenados de forma segura en el cliente.
- Desarrollo de componentes interactivos para la presentación de ejercicios generados por el tutor inteligente, adaptados al nivel y destreza del estudiante.
- Aplicación de criterios básicos de usabilidad y diseño responsivo para asegurar compatibilidad con diferentes dispositivos.
- Pruebas funcionales del flujo completo del sistema desde la interfaz, validando la correcta comunicación con el BackEnd y la experiencia de usuario.

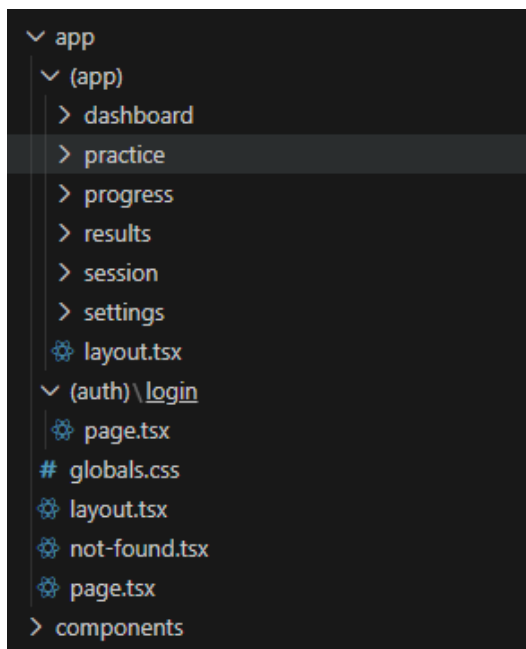
4.2.3.3 RESULTADOS ESPERADOS

- Prototipo de interfaz gráfica (Aplicación web) funcional e integrada completamente con el BackEnd.
- Flujo de usuario operativo desde el inicio de sesión hasta la ejecución de ejercicios académicos.

4.2.3.4 EVIDENCIAS

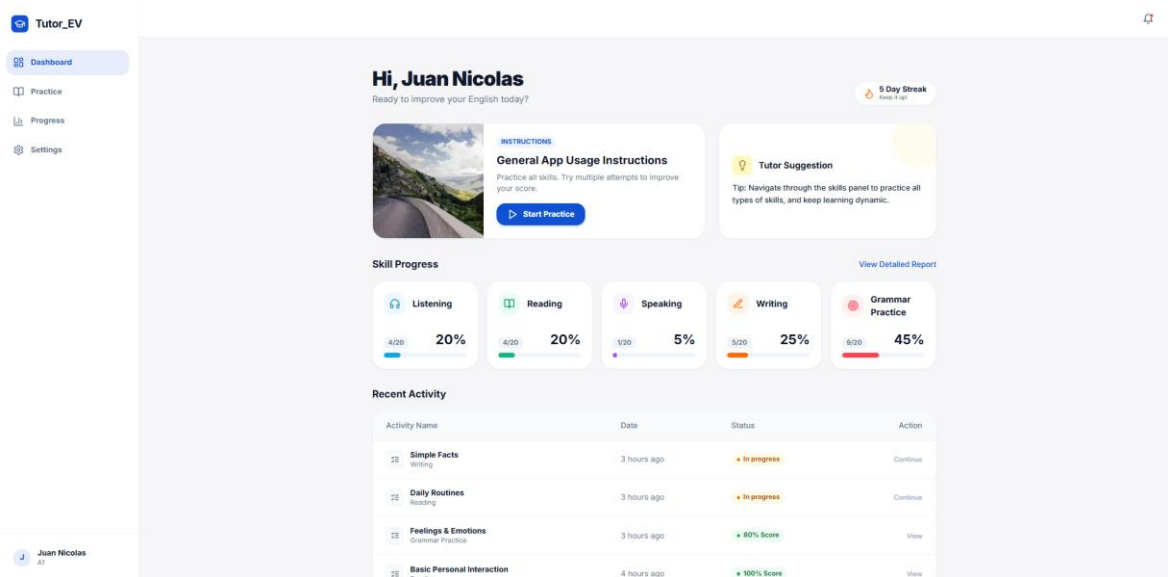
Figura 16

Evidencia 1 - Sprint 3



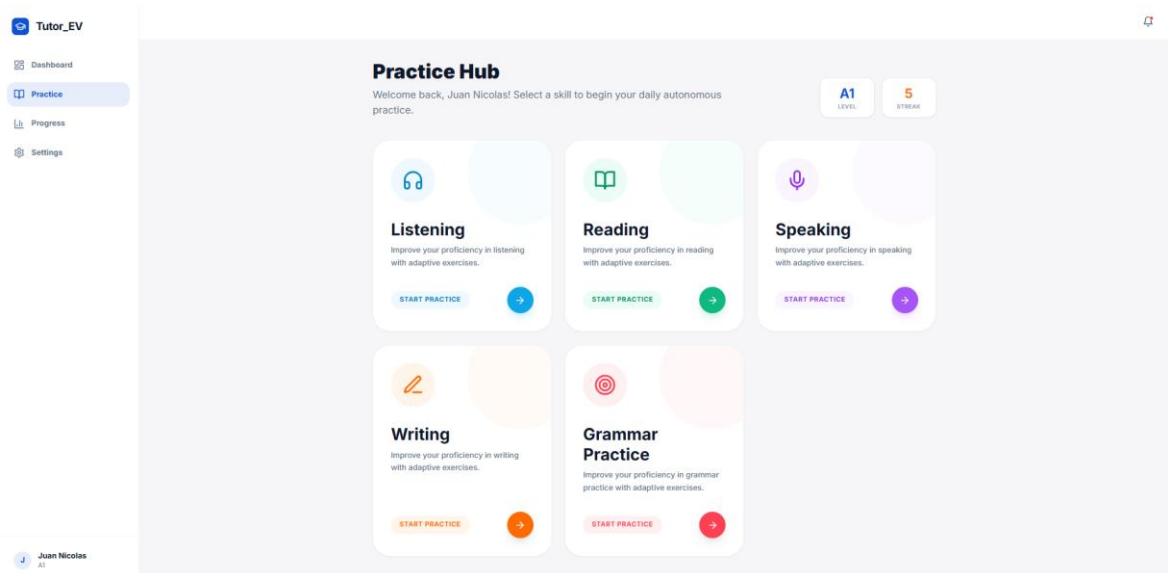
Nota: Evidencia de la creación del proyecto correspondiente al componente de FrontEnd.
Autor: Nicolás Rojas, 2025.

Figura 17
Evidencia 2 - Sprint 3



Nota: Evidencia del resultado de la pantalla de “Dashboard” de la interfaz. Autor: Nicolás Rojas, 2025.

Figura 18
Evidencia 3 - Sprint 3



Nota: Evidencia del resultado de la pantalla de “Practice” de la interfaz. Autor: Nicolás Rojas, 2025.

4.2.3.5 TABLA DE TERMINACIÓN

Tabla 10

Tabla de terminación - Sprint 3

Actividad	Estado	Fecha	Observaciones
Configuración y estructura del proyecto frontend	Completada	18/12/2025	Sin observaciones.
Desarrollo de pantallas principales	Completada	22/12/2025	Sin observaciones.
Integración con backend y manejo de sesiones	Completada	26/12/2025	Sin observaciones.
Corrección de elementos visuales y pantallas finales	Completada	2/01/2026	Sin observaciones.

Nota: Tabla de terminación referente al componente de FrontEnd. Autor: Nicolás Rojas, 2025.

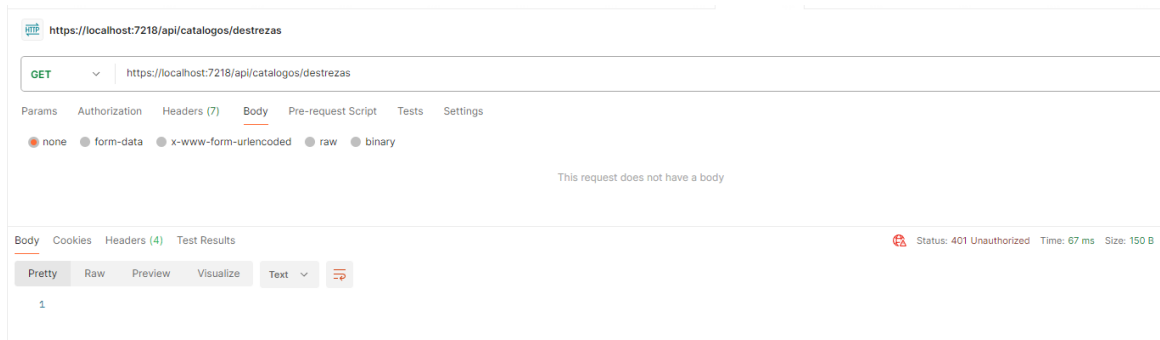
4.3 PRUEBAS

4.3.1 Pruebas funcionales del backend (API REST)

El objetivo de estas pruebas fue verificar el correcto funcionamiento de los endpoints que fueron desarrollados en el sprint 2, resultados del componente del backend. Mediante estas pruebas lo que se buscaba es asegurarse de recibir respuestas válidas, control de acceso y manejo adecuado de errores. Las pruebas se realizaron mediante la herramienta Postman, enviando solicitudes HTTP a los servicios REST, incluyendo endpoints protegidos con autenticación JWT y revisando las respuestas recibidas y el manejo de excepciones en caso de llamadas incorrectas a los endpoints. Este proceso permitió validar la lógica de negocio, la correcta integración con las bases de datos y detectar ajustes necesarios antes del consumo desde la interfaz.

Figura 19

Prueba funcional 1

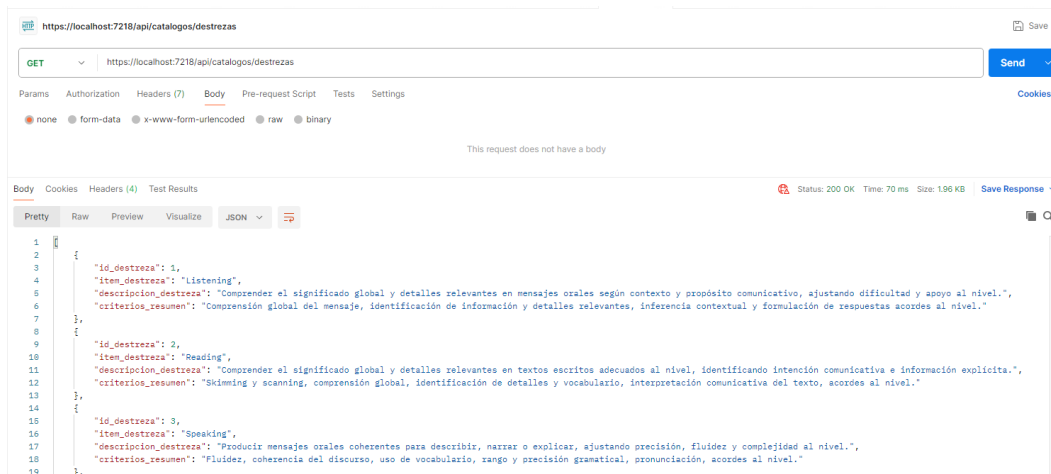


Nota: Prueba sobre endpoint con JWT inválido. Autor: Nicolás Rojas, 2025.

En este caso se realizó un request a los endpoints con un JWT inválido para validar que no se retorne nada y que de un error 401 Unauthorized. En todas las pruebas realizadas funcionó correctamente, en la imagen se muestra el endpoint de `catalogos/destrezas`.

Figura 20

Prueba funcional 2



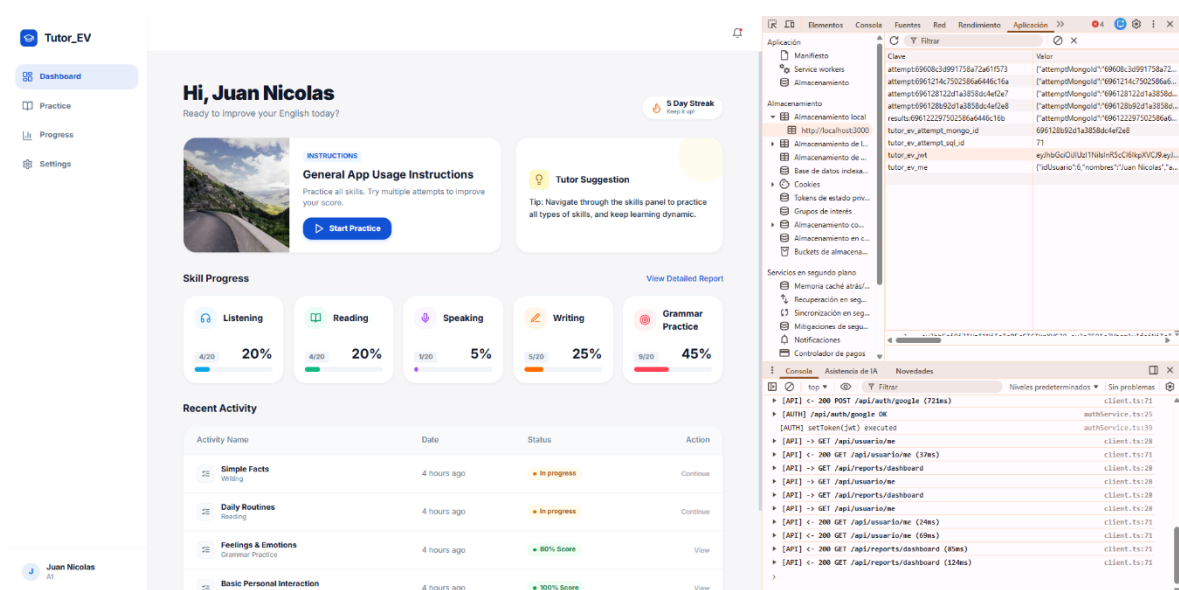
Nota: Prueba sobre endpoint con JWT válido. Autor: Nicolás Rojas, 2025.

En este caso se realizó un request a los endpoints con un JWT válido y se validó que se retorne la información específica de cada endpoint. En todas las pruebas realizadas funcionó correctamente, en la imagen se muestra el endpoint de `catalogos/destrezas`.

4.3.2 Pruebas de integración de la interfaz de usuario con la API REST

Estas pruebas tuvieron como objetivo comprobar la correcta comunicación entre el frontend y el backend del sistema. Se validaron flujos completos desde la interfaz, como autenticación, carga de información y visualización de datos, verificando el uso de tokens JWT almacenados en local storage y cookies. Las pruebas permitieron confirmar la correcta integración de los endpoints y garantizar la consistencia de la información presentada al usuario final.

Figura 21
Prueba integración 1

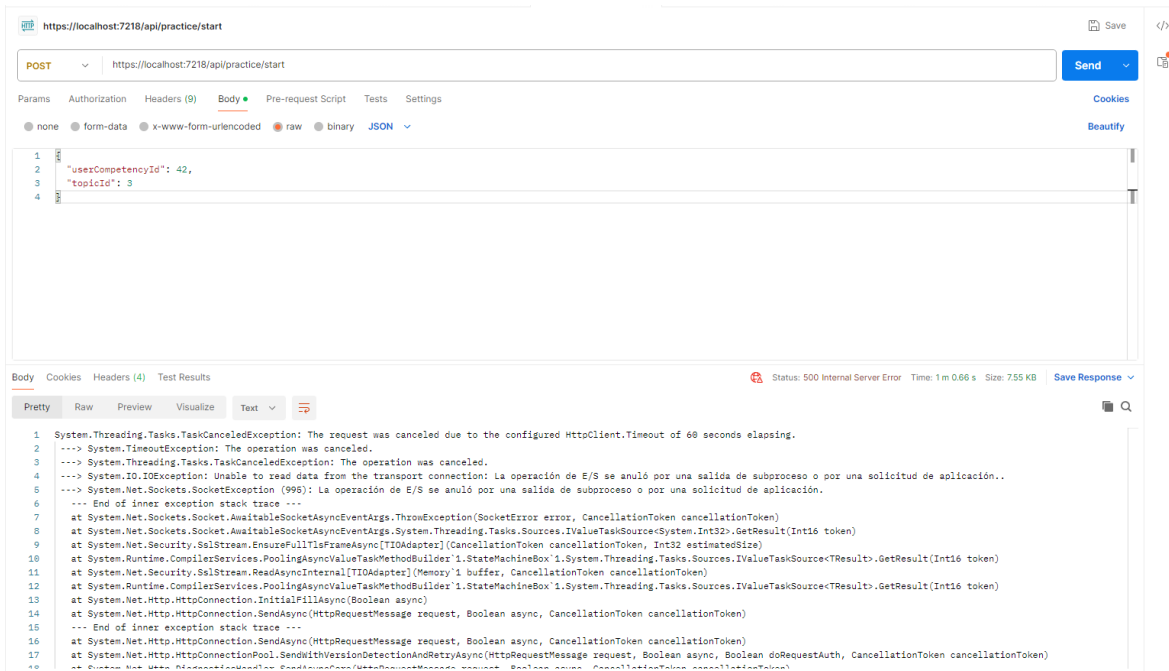


Nota: Prueba sobre la integración de los endpoints con los datos dinámicos en la interfaz.
Autor: Nicolás Rojas, 2025.

4.3.3 Pruebas de validación funcional del modelo de lenguaje (API LLM)

El objetivo de estas pruebas fue validar el comportamiento funcional del modelo de lenguaje utilizado para la generación de ejercicios académicos. Las pruebas se realizaron mediante llamadas directas a la API del LLM usando Postman, evaluando la estructura, coherencia y adecuación de las respuestas generadas según el prompt definido. Este proceso permitió ajustar los parámetros y asegurar que las respuestas cumplan con los requisitos académicos del sistema.

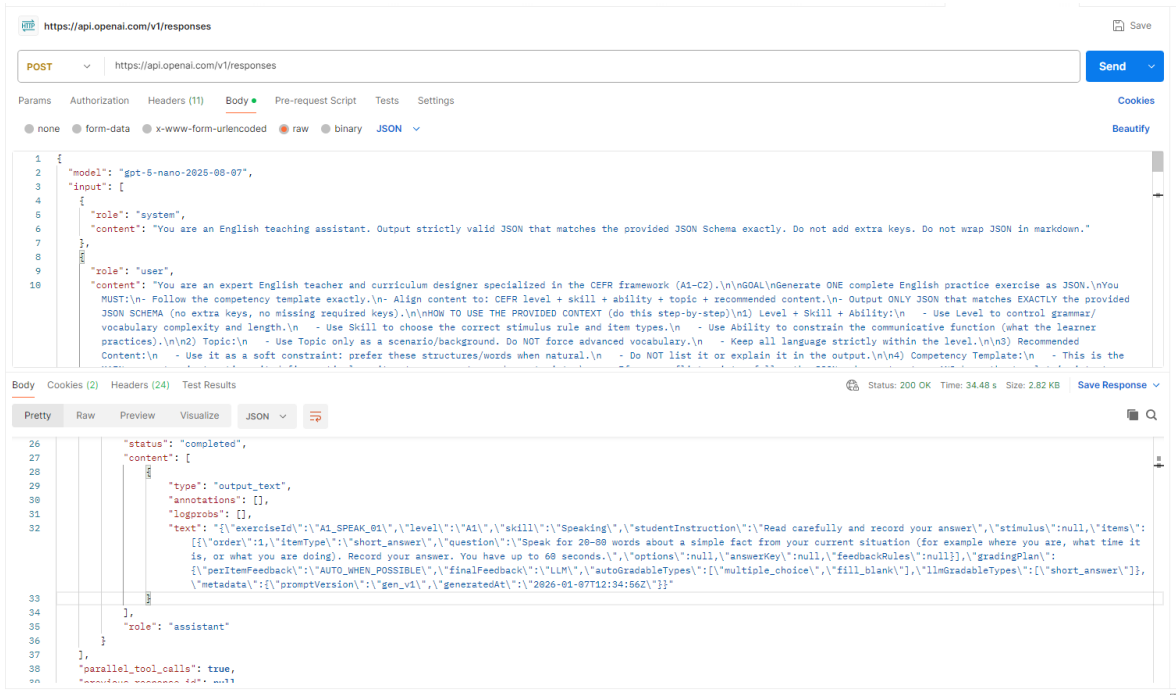
Figura 22
Prueba validación 1



Nota: Prueba sobre la API del LLM en el cual se genera un fallo por timeout. Autor: Nicolás Rojas, 2025.

Aquí podemos verificar que la API del servicio de LLM suele tener fallos por timeout, al fallar devuelve un error 500, el cual menciona que el servicio backend no recibió resultados devueltos por la IA en el lapso admitido (60 segundos).

Figura 23
Prueba validación 2



Nota: Prueba sobre la API del LLM para validar calidad del prompt y resultado devuelto.
Autor: Nicolás Rojas, 2025.

Aquí podemos verificar una prueba a la API del servicio de LLM de manera externa para poder verificar que tipo de response envía y probar la calidad tanto del prompt enviado como también de los resultados.

CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES

El presente capítulo expone las conclusiones y recomendaciones derivadas del análisis, diseño e implementación inicial del prototipo de tutor basado en inteligencia artificial desarrollado para la academia English Valley. Estos resultados se fundamentan exclusivamente en el trabajo realizado durante el proceso investigativo y de construcción del sistema, considerando los avances alcanzados en el modelado de datos, la arquitectura, la integración tecnológica y la validación técnica de los componentes del prototipo. Las recomendaciones propuestas buscan orientar futuras mejoras y consolidar el potencial pedagógico y operativo de la solución dentro del contexto académico institucional.

5.1 CONCLUSIONES

1. El desarrollo del prototipo permitió comprobar la viabilidad técnica de integrar modelos de lenguaje actuales con una arquitectura web moderna basada en ASP.NET Core, SQL Server y MongoDB. El funcionamiento del sistema evidenció que es posible operar de manera consistente, integrar servicios externos como la API de inteligencia artificial y gestionar datos estructurados y no estructurados sin comprometer la coherencia del flujo funcional. Este resultado confirma que el tutor de aprendizaje puede implementarse de forma escalable y eficiente dentro del entorno tecnológico de la academia.
2. El análisis y modelado de datos basado en los descriptores del nivel A1 del MCER permitió estructurar una solución pedagógicamente coherente y alineada a las necesidades de la academia. Por medio del prototipo se pudo utilizar la IA para representar con fidelidad las habilidades, destrezas, contenidos y competencias del proceso formativo, preservando la trazabilidad del progreso y la lógica de evaluación. De este modo, se concluye que la integración tecnológica puede realizarse sin afectar la fundamentación educativa ni la metodología activa de la academia English Valley.

3. La arquitectura diseñada en capas confirmó que el prototipo posee una estructura sólida, funcional y escalable que puede adaptarse a futuras ampliaciones del sistema. Esta estructura modular permite la incorporación de nuevas funcionalidades sin afectar el funcionamiento central del prototipo, facilitando su futura expansión hacia otros niveles del MCER y la integración de módulos adicionales de retroalimentación y evaluación. En consecuencia, el prototipo desarrollado establece una base técnica que permite el crecimiento progresivo de la plataforma.

5.2 RECOMENDACIONES

1. Dado que la arquitectura técnica y el modelo de datos fueron diseñados con un enfoque escalable, se recomienda ampliar el prototipo hacia niveles superiores del MCER una vez finalizada la implementación completa del nivel A1. Esto permitirá aprovechar la estructura ya establecida, generar continuidad pedagógica en el aprendizaje y fortalecer la utilidad del sistema dentro del proceso formativo de la academia.
2. Se recomienda que la academia English Valley establezca lineamientos internos que regulen el uso pedagógico y técnico de herramientas de inteligencia artificial. Estos lineamientos deberían contemplar criterios sobre evaluación, retroalimentación, privacidad de datos, límites de uso y orientaciones metodológicas para integrar el tutor de aprendizaje como un recurso complementario al trabajo de aula. Este enfoque permitirá una implementación segura, responsable y coherente con los valores institucionales.
3. Como evolución natural del prototipo, se recomienda integrar un módulo orientado a los docentes que permita visualizar reportes de desempeño, historial de práctica y métricas por habilidad o destreza. Esto facilitaría la toma de decisiones pedagógicas y permitiría personalizar la enseñanza según las necesidades de cada estudiante. Adicionalmente, se sugiere incorporar opciones de configuración académica como selección de contenidos, creación de actividades o ajustes de criterios de evaluación, para que el personal docente pueda adaptar el funcionamiento del tutor a los objetivos académicos de cada curso.

BIBLIOGRAFÍA

- Beck, K. (2004). *Extreme Programming Explained: Embrace Change* (2nd ed.). Addison-Wesley.
- Beedle, M., Beck, K., & Van Bennekum, A. (2001). *Manifesto for Agile Software Development*. Obtenido de Agile Alliance: <https://agilemanifesto.org/>
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., & Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 1877-1901.
- Cattell, R. (2011). Scalable SQL and NoSQL Data Stores. *ACM SIGMOD Record*, 39, 12-27.
- Chodorow, K. (2020). *MongoDB: The Definitive Guide* (3rd ed.). O'Reilly Media.
- Connolly, T., & Begg, C. (2021). *Database systems: A practical approach to design, implementation, and management* (7th ed.). Pearson Education.
- Consejo de Europa. (2021). *Marco común europeo de referencia para las lenguas: aprendizaje, enseñanza, evaluación. Volumen complementario*. Ministerio de Educación y Formación Profesional e Instituto Cervantes. Obtenido de https://cvc.cervantes.es/ensenanza/biblioteca_ele/marco_complementario/anejo_01.htm
- Council of Europe. (20 de Octubre de 2025). *Common European Framework of Reference for Languages (CEFR)*. Obtenido de The CEFR Levels: <https://www.coe.int/en/web/common-european-framework-reference-languages/level-descriptions>
- Date, C. (2019). *An Introduction to Database Systems* (8th ed.). Pearson Education.
- Du, J., & Daniel, B. K. (2024). Transforming language education: A systematic review of AI-powered chatbots for English as a foreign language speaking practice. *Computers and Education: Artificial Intelligence*. doi:10.1016/j.caeai.2024.100230
- EF Education First. (2024). *EF English Proficiency Index (EPI) 2024: A Ranking of 116 Countries and Regions by English Skills*. EF Education First. Obtenido de <https://www.ef.com/wwen/epi/>
- Elmasri, R., & Navathe, S. (2020). *Fundamentals of Database Systems* (8th ed.). Pearson.
- Elmasri, R., & Navathe, S. B. (2020). *Fundamentals of database systems* (7th ed.). Pearson Education.
- Feng, S., Magana, A. J., & Kao, D. (2021). A Systematic Review of Literature on the Effectiveness of Intelligent Tutoring Systems in STEM. *2021 IEEE Frontiers in Education Conference (FIE)*, (págs. 1-9). doi:10.1109/FIE49875.2021.9637240
- Google DeepMind. (2024). *Introducing Gemini 1.5*. Obtenido de Google DeepMind: <https://deepmind.google/>

- Google Firebase. (2024). *Cloud Firestore Documentation*. Obtenido de Google Firebase: <https://firebase.google.com/docs/firestore>
- Google Firebase. (2024). *Firestore Authentication Documentation*. Obtenido de Google Firebase: <https://firebase.google.com/docs/auth>
- Google Identity Platform. (2024). *Using OAuth 2.0 for Web Server Applications*. Obtenido de Google for Developers: <https://developers.google.com/identity/protocols/oauth2>
- Grassi, P. A., Garcia, M. E., & Fenton, J. L. (2020). *Digital Identity Guidelines: Authentication and Lifecycle Management*. National Institute of Standards and Technology.
- Han, J., Haihong, E., Le, G., & Du, J. (2011). Survey on NoSQL Database. *Proceedings of the 6th International Conference on Pervasive Computing and Applications*, (págs. 363-366).
- Hardt, D. (2012). *The OAuth 2.0 Authorization Framework*. Internet Engineering Task Force. Obtenido de <https://www.rfc-editor.org/rfc/rfc6749>
- <https://learn.microsoft.com/es-es/ssms/>. (2025). *What is Visual Studio?* Obtenido de Microsoft Learn: <https://learn.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2022>
- Intelivita. (2025). *Frontend vs Backend Development: Key Differences Explained*. Obtenido de Intelivita: <https://www.intelivita.com/blog/frontend-vs-backend-development/>
- Kuiken, F. (2022). Linguistic complexity in second language acquisition. *Linguistics Vanguard*, 83-93. doi:<https://doi.org/10.1515/lingvan-2021-0112>
- Kumar, A. (2022). *Introduction to NestJS – A Progressive Node.js Framework*. Obtenido de GeeksforGeeks: <https://www.geeksforgeeks.org/javascript/what-is-nestjs/>
- Larsen-Freeman, D. (1997). Chaos/Complexity Science and Second Language Acquisition. *Applied Linguistics*, 2, 141-165. doi:<https://doi.org/10.1093/applin/18.2.141>
- Li, Y., & Manoharan, S. (2013). A performance comparison of SQL and NoSQL databases. *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, (págs. 15-19).
- Meta. (2024). *React Documentation*. Obtenido de react.dev: <https://react.dev/>
- Microsoft. (2024). *Entity Framework Core overview*. Obtenido de Microsoft Learn: <https://learn.microsoft.com/en-us/ef/core/>
- Microsoft. (2024). *Introduction to ASP.NET Core*. Obtenido de Microsoft Learn: <https://learn.microsoft.com/en-us/aspnet/core/introduction-to-aspnet-core>
- Microsoft. (2024). *SQL Server Documentation*. Obtenido de Microsoft Learn: <https://learn.microsoft.com/en-us/sql/sql-server/>

- Microsoft. (2024). *What is Visual Studio?* Obtenido de Microsoft Learn: <https://learn.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2022>
- Microsoft. (2025). *SQL Server Management Studio (SSMS)*. Obtenido de Microsoft Learn: <https://learn.microsoft.com/es-es/ssms/>
- Microsoft. (s.f.). *Visual Studio Code documentation*. Obtenido de Visual Studio Code: <https://code.visualstudio.com/docs>
- MongoDB Inc. (s.f.). *What is MongoDB Compass?* Obtenido de MongoDB Docs: <https://www.mongodb.com/docs/compass/>
- Mozilla Developer Network. (2024). *Frontend Web Development Overview*. Obtenido de Mozilla Developer Network: https://developer.mozilla.org/en-US/docs/Learn/Front-end_web_developer
- OpenAI. (2024). *API Reference: Completions and Chat Endpoints*. Obtenido de OpenAI Platform: <https://platform.openai.com/docs/api-reference>
- OpenAI. (2024). *GPT-4o: OpenAI API Documentation*. Obtenido de OpenAI Platform: <https://platform.openai.com/docs/models/gpt-4o>
- PostgreSQL Global Development Group. (2024). *PostgreSQL Documentation*. Obtenido de PostgreSQL: <https://www.postgresql.org/docs/>
- Pressman, R. S., & Maxim, B. R. (2020). *Software Engineering: A Practitioner's Approach* (9th ed.). McGraw-Hill Education.
- Quizhpe, R., & Yanella, R. (2024). The Use of Self-Produced Videos to Improve Speaking Skills: A Complete Protocol. *Universidad San Francisco de Quito (USFQ)*. Obtenido de <https://repositorio.usfq.edu.ec/handle/23000/14052>
- Rising, L., & Janoff, N. S. (2000). The Scrum Software Development Process for Small Teams. *IEEE Software*, 17, 26-32.
- Rivas, A. (2025). *La llegada de la IA a la educación en América Latina: en construcción*. Organización de Estados Iberoamericanos (OEI) & Fundación ProFuturo, Madrid. Obtenido de <https://oei.int/oficinas/secretaria-general/publicaciones/la-llegada-de-la-ia-a-la-educacion-en-america-latina-en-construccion>
- Russell, S., & Norvig, P. (2022). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.
- Sadalage, P., & Fowler, M. (2012). *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*. Addison-Wesley.
- Sanz Manzanedo, M. (2025). La IA en la enseñanza de idiomas: chatbots y formación del profesorado. *European Public & Social Innovation Review*. doi:10.31637/epsir-2025-513

- Sasmayunita, & Assiddiq, M. A. (2025). The Role of Artificial Intelligence (AI) in Promoting Autonomous Learning in Language Education. *The 5th International Conference on Linguistics and Cultural Studies 5 (ICLC-5 2024)*, (págs. 349-363). doi:10.2991/978-2-38476-394-8_40
- Schwaber, K., & Sutherland, J. (2020). *The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game*. Obtenido de Scrum.org
- Silberschatz, A., Korth, H., & Sudarshan, S. (2020). *Database System Concepts* (7th ed.). McGraw-Hill Education.
- Söylemez, M., Tekinerdogan, B., & Kolukisa Tarhan, A. (2022). Challenges and solution directions of microservice architectures: A systematic literature review. *Applied Sciences*, 5507. doi:https://doi.org/10.3390/app12115507
- Stallings, W. (2018). *Cryptography and Network Security: Principles and Practice* (8th ed.). Pearson Education.
- Tidwell, J., Brewer, C., & Valencia, A. (2020). *Designing interfaces: Patterns for effective interaction design* (3rd ed.). O'Reilly Media.
- Tu, Z. (2023). Research on the application of layered architecture in computer software development. *Journal of Computing and Electronic Information Management*, 11, 34-38. doi:https://doi.org/10.54097/jceim.v11i3.08
- UNESCO. (2025). *AI and the future of education: disruptions, dilemmas and directions*. Obtenido de UNESDOC Digital Library: <https://unesdoc.unesco.org/ark:/48223/pf0000395236?posInSet=3&queryId=d1650d47-8561-4475-855e-53a54b31cd36>
- W3Schools. (2025). *Frontend Development Introduction*. Obtenido de W3Schools: https://www.w3schools.com/whatis/whatis_frontenddev.asp
- Wang, I. X., Wu, X., Coates, E., Zeng, M., Kuang, J., Liu, S., . . . Park, J. (2024). *Neural Automated Writing Evaluation with Corrective Feedback*. Obtenido de arXiv: <https://arxiv.org/abs/2402.17613>
- Wei, L. (2023). Artificial intelligence in language instruction: impact on English learning achievement, L2 motivation, and self-regulated learning. *Frontiers in Psychology*, 14. doi:https://doi.org/10.3389/fpsyg.2023.1261955
- You, E. (2023). *Vue.js Documentation*. Obtenido de vuejs.org: <https://vuejs.org/>

ANEXOS

ACTA DE INICIACIÓN SPRINT 1 (SP1)



PREPARATION CENTRE FOR INTERNATIONAL CERTIFICATES

Tutor EV - Acta de inicio Sprint 1 (SP1)

Proyecto	Prototipo de tutor basado en IA – English Valley		
Nombre Producto	Implementación y Validación de Bases de Datos		
Producto No:	P_001	Acta No:	001
Fecha Inicio:	10/11/2025	Fecha Fin:	21/11/2025
Responsables			
Nombre	Rol	Institución	Firma
Claudia Martínez	Product Owner	English Valley	
Nicolás Rojas	Scrum Master / Developer	PUCE	

1. Objetivo del Sprint: Implementar y validar la base de datos relacional y el entorno NoSQL del prototipo, garantizando integridad, coherencia y disponibilidad de los datos que soportan las funcionalidades posteriores del sistema.

2. Cronograma:

Fecha	Actividad
10/11/2025	Implementación BD SQL Server.
11/11/2025	Datos de prueba y consultas.
17/11/2025	Configuración MongoDB.
19/11/2025	Pruebas de conexión backend.

gerencia@englishvalley.edu.ec

+593 99 423 0643

<https://englishvalley.edu.ec>

ACTA DE INICIACIÓN SPRINT 2 (SP2)



PREPARATION CENTRE FOR INTERNATIONAL CERTIFICATES

Tutor EV - Acta de inicio Sprint 2 (SP2)

Proyecto	Prototipo de tutor basado en IA – English Valley		
Nombre Producto	Desarrollo del Backend y Conexión con Servicios Externos		
Producto No:	P_002	Acta No:	002
Fecha Inicio:	24/11/2025	Fecha Fin:	16/12/2025
Responsables			
Nombre	Rol	Institución	Firma
Claudia Martínez	Product Owner	English Valley	
Nicolás Rojas	Scrum Master / Developer	PUCE	

1. Objetivo del Sprint: Implementar la capa de lógica mediante el desarrollo del backend en ASP.NET Core, estableciendo la comunicación con SQL Server, MongoDB, Google OAuth 2.0 y la API del Modelo de Lenguaje. El objetivo es consolidar la estructura funcional interna del sistema.

2. Cronograma:

Fecha	Actividad
30/11/2025	Creación del proyecto backend y estructura multicapa.
05/12/2025	Integración con bases de datos SQL y MongoDB. Capa de Acceso a datos y Datos.
11/12/2025	Integración de autenticación con Google OAuth 2.0 y JWT.
16/12/2025	Consumo preliminar de la API LLM y validación de endpoints.

gerencia@englishvalley.edu.ec

+593 99 423 0643

<https://englishvalley.edu.ec>

ACTA DE INICIACIÓN SPRINT 3 (SP3)



PREPARATION CENTRE FOR INTERNATIONAL CERTIFICATES

Tutor EV - Acta de inicio Sprint 3 (SP3)

Proyecto	Prototipo de tutor basado en IA – English Valley		
Nombre Producto	Desarrollo de Interfaz Web (Frontend)		
Producto No:	P_003	Acta No:	003
Fecha Inicio:	17/12/2025	Fecha Fin:	03/01/2026
Responsables			
Nombre	Rol	Institución	Firma
Claudia Martínez	Product Owner	English Valley	
Nicolás Rojas	Scrum Master / Developer	PUCE	

1. Objetivo del Sprint: Diseñar e implementar la interfaz web del tutor utilizando React, integrando las funcionalidades del backend para habilitar la interacción completa del usuario con el sistema de práctica, autenticación, generación de ejercicios y consulta de progreso.

2. Cronograma:

Fecha	Actividad
18/12/2025	Configuración y estructura del proyecto frontend
22/12/2025	Desarrollo de pantallas principales
26/12/2025	Integración con backend y manejo de sesiones
02/01/2026	Corrección de elementos visuales y pantallas finales

gerencia@englishvalley.edu.ec

+593 99 423 0643

<https://englishvalley.edu.ec>

ACTA DE ACEPTACIÓN



Acta de Aceptación del producto

Tutor EV

En la ciudad de Quito, a los 05 días del mes de Enero de 2026, se suscribe la presente Acta de Aceptación del Producto Final del proyecto titulado "Desarrollo del prototipo de un tutor basado en Inteligencia Artificial para el aprendizaje del idioma inglés", desarrollado para la academia English Valley, como trabajo de titulación de la carrera de Ingeniería en Sistemas de Información.

El producto final entregado corresponde a un prototipo funcional de tutor basado en IA, el cual integra componentes de autenticación, gestión de usuarios, persistencia de datos, servicios de backend y una interfaz de usuario orientada al refuerzo del aprendizaje autónomo del idioma inglés.

Luego de la revisión integral del producto, la Product Owner certifica que el prototipo (MVP) ha sido evaluado y aceptado a satisfacción, verificando su correcto funcionamiento, coherencia con la metodología de la academia y pertinencia como solución tecnológica propuesta.

Para constancia de lo actuado, se firma la presente acta en señal de conformidad.

Nombre: Claudia Martínez

Rol: Product Owner

Institución: English Valley

Nombre: Nicolás Rojas

Rol: Scrum Master / Developer

Institución: PUCE