

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE HÁBITAT, INFRAESTRUCTURA Y

CREATIVIDAD

CARRERA DE SISTEMAS DE LA INFORMACIÓN



TRABAJO DE TITULACIÓN

Tema: DESARROLLO DE UN PROTOTIPO WEB Y MÓVIL PARA EL
APRENDIZAJE DE KICHWA E INGLÉS MEDIANTE JUEGOS
EDUCATIVOS.

AUTORES:

ANDY AROON CONTENTO JAPÓN

OSCAR RONALDO FIGUEROA URIBE

DIRECTOR:

JORGE ALEJANDRO ALARCÓN MENA

QUITO DM, ENERO DE 2026

Dedicatoria

Este trabajo de disertación está dedicado a:

A nuestros padres por el apoyo en cada paso de nuestra vida universitaria.

A nuestros docentes por guiarnos en nuestra vida personal y académica.

A nuestros amigos y próximos colegas de trabajo.

Agradecimiento

Quisiéramos agradecer principalmente a nuestros padres, quienes siempre estuvieron presentes en esta etapa universitaria.

De igual manera, quisiera agradecer a nuestros amigos por apoyarnos en cada obstáculo presentado en este camino que recorrimos juntos.

Por último, quisiera agradecer a los docentes de la universidad por ser excelentes guías y personas que verdaderamente se interesan por el bienestar de sus alumnos.

Abstract

The following dissertation was undertaken to develop a web and mobile prototype for learning Kichwa and English through educational games, with the aim of facilitating learning for children between the ages of 4 and 12. The prototype allows users to visualize their progress and receive feedback, all in an interactive and customizable way. Once the prototype was completed, unit and integration tests were performed to verify its correct operation.

To carry out the prototype development, the most suitable development methodology and tools were first analyzed, comparing the possible options to arrive at the optimal decision that best aligned with the project requirements. The necessary functional and non-functional requirements were established to proceed with the development; in addition, use case diagrams were created to help gain a better understanding of how the prototype works.

The prototype was developed in Python for back-end management, due to its ease of use, clear syntax, and built-in frameworks that simplified the creation of technical documentation. Flutter was used for the front-end, as it is a cross-platform framework that speeds up both web and mobile development, in addition to having high performance and a widget architecture that facilitated the creation of custom interfaces.

Tabla de Contenidos

Dedicatoria.....	II
Agradecimiento	III
Abstract.....	IV
CAPÍTULO 1: Introducción.....	11
1.1 Justificación	11
1.2 Planteamiento del Problema	12
1.3 Objetivos.....	13
1.3.1 Objetivo General.....	13
1.3.2 Objetivos Específicos	13
1.4 Metodología	13
1.5 Alcance	14
1.6 Antecedentes	16
1.7 Marco Teórico.....	17
1.8 Marco Conceptual.....	19
CAPÍTULO 2: Selección de Metodología y Herramientas	25
2.1 Metodología de Desarrollo	25
2.1.1 SCRUM.....	25
2.1.2 Extreme XP	26
2.1.3 Análisis de la Metodología de Desarrollo	28
3.2 Análisis de las Herramientas de Desarrollo	30

3.2.1 PostgreSQL.....	31
3.2.2 DBeaver.....	34
3.2.3 Power Designer	37
3.2.4 Visual Studio Code.....	40
3.2.5 Lucidchart.....	43
3.2.6 Figma.....	46
3.2.7 FastAPI.....	49
3.2.8 Flutter	52
CAPÍTULO 3: Diseño del Prototipo del Sistema Web y Móvil	56
3.1 Análisis de Requerimientos	56
3.2 Casos de Uso.....	57
3.2.1 Requerimientos generales.....	57
3.2.2 Requerimiento F7	58
3.2.3 Requerimiento F10	60
3.3 Requerimientos Funcionales.....	62
3.4 Requerimientos No Funcionales	63
3.5 Planeación de la Arquitectura y Diseño.....	65
3.5.1 Diseño de la arquitectura del prototipo del sistema (microservicios y por capas).	65
3.5.2 Diseño de la Base de Datos del Prototipo del Sistema.....	67
3.5.3 Diseño de la interfaz de usuario del prototipo.....	72

CAPÍTULO 4: Desarrollo y pruebas del prototipo web y móvil	77
4.1 Desarrollo del prototipo	77
4.1.1 Diseño de la base de datos y desarrollo del backend.....	77
4.1.2 Diseño de la interfaz y módulo de autenticación.....	78
4.1.3 Población de BD-Lex y Desarrollo del Módulo de Juegos Interactivos	79
4.1.4 Módulo de configuración para el Usuario, Clasificación Global, Logros y Biblioteca de Aprendizaje	80
4.1.5 Módulo de Configuración Dinámica Administrativa	82
4.2 Pruebas del prototipo web y móvil	83
4.2.1 Pruebas Unitarias	83
4.2.2 Pruebas de Integración	85
4.3 Entrega del Prototipo	91
CAPÍTULO 5: Conclusiones y Recomendaciones.....	92
5.1 Conclusiones	92
5.2 Recomendaciones	92
Bibliografía.....	95
Anexos	102

Contenido de Tablas

Tabla 1	Comparativa de características de metodologías de desarrollo de software	28
Tabla 2	Comparativa de metodologías de desarrollo de software	30
Tabla 3	Comparativa de características de bases de datos	32
Tabla 4	Comparativa de bases de datos	33
Tabla 5	Comparativa de características entre sistemas gestores de bases de datos.....	35
Tabla 6	Comparativa de sistemas gestores de bases de datos	36
Tabla 7	Comparativa de características entre herramientas de diseño de bases de datos... 38	
Tabla 8	Comparativa de herramientas de modelado de bases de datos	39
Tabla 9	Comparativa de características entre entornos de desarrollo integrado (IDE)	41
Tabla 10	Comparativa de entornos de desarrollo integrado	42
Tabla 11	Comparativa de características entre herramientas de diagramación	44
Tabla 12	Comparativa de herramientas de diagramación.....	45
Tabla 13	Comparativa de características entre herramientas de diseño de interfaces	47
Tabla 14	Comparativa de herramientas de diseño de interfaz.....	48
Tabla 15	Comparativa de características frameworks para el desarrollo del backend	50
Tabla 16	Comparativa de frameworks para el desarrollo del backend.....	51
Tabla 17	Comparativa de características de frameworks para el desarrollo del frontend..	53
Tabla 18	Comparativa de frameworks para el desarrollo del frontend	54

Contenido de Figuras

Figura 1 Diagrama de caso de uso general.....	57
Figura 2 Caso de uso requerimiento F7.....	58
Figura 3 Caso de uso requerimiento F10.....	60
Figura 4 Modelo Conceptual BD-Lex.....	68
Figura 5 Modelo lógico BD-Lex.....	68
Figura 6 Modelo físico BD-Lex.....	69
Figura 7 Modelo conceptual BD-Game (parte 1).....	69
Figura 8 Modelo conceptual BD-Game (parte 2).....	70
Figura 9 Modelo lógico BD-Game (parte 1).....	70
Figura 10 Modelo lógico BD-Game (parte 2).....	71
Figura 11 Modelo físico BD-Game (parte 1).....	71
Figura 12 Modelo físico BD-Game (parte 2).....	72
Figura 13 Paleta de colores de la interfaz del prototipo.....	74
Figura 14 Jugar nivel (web).....	74
Figura 15 Jugar nivel (móvil).....	75
Figura 16 Agregar nivel (web).....	75
Figura 17 Agregar nivel (móvil).....	76
Figura 18 API del prototipo.....	78
Figura 19 Registro e inicio de sesión del prototipo.....	78
Figura 20 Población BD-Lex.....	79
Figura 21 Módulo de juegos interactivos.....	79
Figura 22 Módulo de Configuración para el usuario.....	80
Figura 23 Módulo de clasificación global.....	81

Figura 24 Módulo de logros y recompensas	81
Figura 25 Módulo de biblioteca de aprendizaje	82
Figura 26 Módulo de configuración dinámica administrativa	82
Figura 27 Registro Exitoso	83
Figura 28 Registro Duplicado	84
Figura 29 Login Exitoso	84
Figura 30 Login incorrecto	85
Figura 31 Registro desde la interfaz (parte 1)	86
Figura 32 Registro desde la interfaz (parte 2)	87
Figura 33 Registro de usuario en la base de datos.....	87
Figura 34 Registro de usuario con correo ya existente	88
Figura 35 Inicio de sesión desde la interfaz (parte 1).....	89
Figura 36 Inicio de sesión desde la interfaz (parte 2).....	90
Figura 37 Inicio de sesión con credenciales incorrectas	90

CAPÍTULO 1: Introducción

1.1 Justificación

Actualmente, la pérdida progresiva del idioma kichwa es una clara amenaza contra la diversidad cultural y lingüística del Ecuador, Pomboza-Tamaquiza, Paucar Pomboza, & Ulcuango Ulcuango en su artículo ‘Idioma Kichwa y resistencia cultural en los pueblos indígenas de Tungurahua-Ecuador’ publicado en 2024, muestra en su estudio cómo el kichwa ha sido desplazado por el español en ámbitos familiares, educativos y comunitarios, y que esto no solo implica la posible desaparición de un idioma, sino también de todo un conjunto de saberes ancestrales y cosmovisiones que son parte de la cultura del país. A pesar de ser reconocida formalmente como lengua ancestral oficial, su transmisión de generación en generación se ha limitado, dado que no se han desarrollado recursos educativos accesibles, actualizados y adaptados a las nuevas generaciones. En un contexto actual, donde las lenguas que predominan son el español e inglés, se evidencia el desplazamiento de las lenguas ancestrales, en los entornos familiares y digitales.

Vista la realidad y contexto actual, el dominio del idioma inglés se ha convertido en un factor determinante para acceder a oportunidades educativas, profesionales o tecnológicas. Por lo que una propuesta educativa que combine aprendizaje del kichwa e inglés puede no solo fortalecer el multilingüismo, también puede promover el respeto y la valorización de la identidad cultural desde la infancia.

El desarrollo del prototipo web y móvil aprovechó el potencial de los juegos educativos como método de enseñanza. Se incorporaron herramientas interactivas como el uso de juegos de la memoria, la determinación de niveles progresivos y la aplicación de recompensas; se pretendió ofrecer a los niños un aprendizaje atractivo, intuitivo y significativo.

El prototipo se centró en un público de rango etario de 4 a 12 años, ya que gracias a la revisión del artículo ‘Gamificación y tecnología en la educación infantil: una revisión sistemática’, publicado en la revista de InveCom por Buendía, Tasayco, & Menacho en 2025 se puede afirmar que es un periodo en donde los niños presentan una notable plasticidad cognitiva y lingüística en la adquisición de vocabulario y pronunciación; además de que la gamificación en la educación infantil promueve y mejora la autonomía en el aprendizaje.

1.2 Planteamiento del Problema

El aprendizaje e instrucción del Kichwa en Ecuador está limitado por la ausencia de recursos didácticos que lo integren desde una mirada accesible, actual e interactiva, ya que el aprendizaje e instrucción del Kichwa se reduce en el aula por recursos didácticos estáticos y sin estrategias pedagógicas que muestren a las nuevas generaciones que es un idioma aprensible y significativo para la identificación de la cultura de su pueblo. Es decir, por la escasa diversidad en el manejo de los recursos se ha incrementado la pérdida de la lengua sobre todo para niños y jóvenes.

A pesar de la proximidad de las tecnologías educativas, no existen aplicaciones web y móvil que, paralelamente, enseñen el kichwa y el inglés por medio de juegos educativos de carácter interactivo, adaptados a la infancia, en forma de memoria, actividades visuales, etc. y que hacen ver que la brecha se acentúa tanto en la práctica adecuada de lenguas originarias, como en la incorporación de estrategias de recursos didácticos modernos.

En este sentido, se plantea la necesidad de diseñar una herramienta tecnológica que sea accesible e inclusiva y que sirva para enseñar simultáneamente el kichwa y el inglés por medio de la motivación de juego, potenciando así la lengua y la identidad cultural.

1.3 Objetivos

1.3.1 Objetivo General

- Desarrollar un prototipo web y móvil (Android) que utilice juegos educativos interactivos para facilitar el aprendizaje simultáneo de kichwa e inglés en niños, promoviendo la preservación lingüística mediante herramientas tecnológicas accesibles y atractivas.

1.3.2 Objetivos Específicos

- Documentar requerimientos funcionales y no funcionales del prototipo, asegurando su alineación con las necesidades del aprendizaje bilingüe kichwa–inglés.
- Analizar herramientas tecnológicas clave justificando su elección en base a criterios de compatibilidad, rendimiento y seguridad.
- Diseñar una arquitectura de prototipo en capas, un modelo de base de datos y el diseño de la interfaz de usuario.
- Desarrollar los módulos principales del prototipo, incluyendo autenticación, juegos interactivos, seguimiento de experiencia de usuario, biblioteca de aprendizaje y configuración dinámica del contenido.

1.4 Metodología

Una metodología ágil es idónea para realizar el desarrollo del prototipo. Según Laoyan (2025), esta clase de metodologías son iterativas, realizan entregas cíclicas, entregan valor y reciben feedback constantemente a largo de todo el desarrollo. Además, la metodología ágil cuenta con cuatro valores principales que son:

- Individuos e interacciones sobre procesos y herramientas: se valora más a los individuos y las interacciones entre ellos que a los procesos y herramientas, fomentando una mejor colaboración en equipo.
- Software funcionando sobre documentación extensa: se valora más un software en funcionamiento que una documentación extensa.
- Colaboración con el cliente sobre negociación contractual: se valora más la colaboración entre las partes interesadas y el cliente, se debe dejar que el cliente marque oriente la dirección que seguirá el software.
- Respuesta ante el cambio sobre seguir un plan: se valora más la respuesta ante cambios que seguir un plan, lo que permite a los equipos cambiar rápido de estrategia sin tener repercusiones con el proyecto entero.

En resumen, se puede decir que una metodología ágil permite realizar ajustes rápidos basados en las necesidades que surgen a lo largo del desarrollo y promueve el desarrollo incremental de un producto. Por estos motivos, se tomaron en consideración para el desarrollo del prototipo las metodologías de desarrollo SCRUM y XP, las cuales fueron analizadas con mayor detenimiento más adelante, con el fin de escoger la opción que más se adaptaba a las necesidades del proyecto.

1.5 Alcance

El presente trabajo de titulación se centró en el desarrollo de un prototipo de aplicación educativa web y móvil orientado al aprendizaje del idioma kichwa e inglés mediante el uso de juegos interactivos. El prototipo está dirigido a niños de entre 4 y 12 años, considerando principios de usabilidad, accesibilidad digital y autoaprendizaje. Se establecieron dos roles dentro del prototipo:

- Jugador: quién puede acceder a los juegos, ganar recompensas y visualizar su progreso.
- Administrador: quién se encarga de gestionar el contenido interactivo de cada modo de juego y de contenidos adicionales como reversos de cartas y avatares.

El prototipo cuenta con una configuración dinámica solamente gestionada por el rol de administrador.

El prototipo incluye dos modos de juego:

- Memorama bilingüe (kichwa e inglés).
- Emparejamiento de palabras (relación con su traducción correcta).

El juego tiene 10 niveles, organizados en secciones temáticas, por ejemplo:

- Niveles del 1 al 5: Números.
- Niveles del 6 al 10: Animales.

Se incluye un módulo de 'Biblioteca' dividido en dos secciones:

- Colección: donde se muestran las cartas y palabras revisadas por el usuario.
- Errores: donde se guardan las palabras en las que el usuario tuvo problemas al pasar los niveles, ordenadas de mayor a menor número de fallos.

Entre las funcionalidades que están incluidas en el prototipo se contempla:

- Registro e inicio de sesión mediante autenticación.
- Acceso a un módulo principal de juegos con palabras en kichwa e inglés.
- Implementación de niveles de dificultad progresiva.
- Sistema de recompensas (monedas virtuales, desbloqueo de reversos de cartas, avatares).

- Panel de clasificación global con los mejores jugadores.
- Visualización del perfil y avance del usuario.

El prototipo se desarrolló utilizando un framework que permitió su ejecución tanto en navegadores web como en dispositivos móviles Android. El backend se apoyó en un sistema para la autenticación de usuarios, en conjunto con una base de datos relacional para el manejo y almacenamiento de datos.

Es importante aclarar que este proyecto se limitó a funcionalidades orientadas exclusivamente al aprendizaje de vocabulario básico y asociación visual. No se contempló en esta fase la integración con herramientas de reconocimiento de voz, traducción automática ni contenidos gramaticales avanzados.

Asimismo, el sistema se diseñó como un prototipo funcional, no como una aplicación lista para producción. Por tanto, se validó mediante pruebas unitarias y de integración, pero no se realizó una publicación oficial en tiendas de aplicaciones durante el alcance de este proyecto.

1.6 Antecedentes

La enseñanza de lenguas originarias como el kichwa, y también la práctica del inglés como lengua global, son llevadas a cabo en entornos escolares a partir de enfoques tradicionales que limitan la interacción dinámica entre el docente y el alumno con recursos impresos, pizarra, ejercicios a repetir que suelen ser funcionales e interesantes, pero resultan poco óptimos para las nuevas generaciones de niños y jóvenes. La práctica de la enseñanza de idiomas necesita de estrategias que propicien el uso de tecnologías de contexto cultural que faciliten la práctica constante del conocimiento del idioma. Por otra parte, el kichwa como lengua ancestral ecuatoriana presenta un alto riesgo de pérdida progresiva al no estar

integrada en procesos de enseñanza modernos, lo que deja entrever una falta de herramientas tecnológicas accesibles para el aprendizaje de kichwa mediante juegos interactivos.

1.7 Marco Teórico

En estos últimos años, la utilización de herramientas tecnológicas en la educación ha supuesto un recurso apasionante y eficaz para conseguir la motivación y la retención del conocimiento, sobre todo en el enfoque de aprendizajes de lengua. En el caso de las lenguas de origen como el kichwa, son escasas las iniciativas que implementan digitalmente las técnicas pedagógicas, culturales y tecnológicas en un único recurso educativo. No obstante, hay algunos estudios que ponen a prueba las aplicaciones móviles y los juegos interactivos como un medio para la conservación y para la enseñanza del kichwa en el ámbito formal e informal de aprendizaje. Por ejemplo, Pilicita-Garrido y Cevallos-Duque (2019) desarrollaron una aplicación para Android que hace las veces de guía interactiva para la enseñanza de kichwa. Su propuesta incluía la gamificación a partir del uso de trivias, imágenes y materiales auditivos buscando lograr corresponder palabras y significados. Los autores terminan argumentando que este tipo de aplicaciones suponen un medio eficaz para la promoción del uso de la lengua como recurso para reforzar la enseñanza desde edades tempranas, muy especialmente en esos puntos donde el acceso a los recursos educativos es limitado, haciendo mención a la gamificación como un recurso que propicia el incremento de la participación del usuario en su propio proceso de aprendizaje.

De igual manera, los trabajos de investigación más recientes como el de Cárdenas-Valdivia, Floreas-Alvines y Cabanillas-Carbonell, (2023) en su artículo *Augmented Reality for Quechua Language Teaching-Learning: A Systematic Review* publicado en la revista *International Journal of Interactive Mobile Technologies*, ha evidenciado que el uso de las

tecnologías emergentes (en este caso la realidad aumentada) puede ser un excelente facilitador del aprendizaje de lenguas originarias, como por ejemplo el quechua. A partir de una revisión sistemática, los autores indican que tales prototipos de realidad aumentada exhiben un aumento de la motivación, comprensión visual y conexión emocional a la lengua, la cultura y su ancestralidad. A pesar de que el objetivo del estudio se centró en la lengua quechua, dicho estudio también puede ser extrapolado al kichwa, ya que ambas lenguas son andinas y comparten raíces.

La idea en torno a la que gira el desarrollo del prototipo es la enseñanza del kichwa e inglés, por lo que se realizó una búsqueda de material fidedigno de diccionarios que proporcionen términos y su traducción al castellano; el Ministerio de Educación del Ecuador (2009) ofreció uno, llamado ‘Yachakukkunapa Shimi Yuk Kamu: Runa Shimi – Mishu Shimi’ y es el material que se usó para añadir términos de kichwa en el prototipo. Este material se creó con la idea de rescatar, normalizar y fortalecer el kichwa dentro del sistema educativo ecuatoriano y es por ese motivo que no solo es un diccionario, también ofrece información sobre fonemas y sintaxis, lo que nos da como resultado una buena herramienta pedagógica.

El prototipo al ser una aplicación web y móvil enfocado en la enseñanza a través de la gamificación, lo que busca es ser óptima en la experiencia del usuario, y es por eso que se tomaron en cuenta algunos de los principios o reglas heurísticas de Nielsen para el desarrollo. Según la Universidad Internacional de La Rioja (2024) estas heurísticas son 10 pautas creadas por Jakob Nielsen, quien buscaba optimizar la experiencia de los usuarios, lo que brinda una mejora en la interacción. Revisemos cada regla:

- **Visibilidad del estado del sistema:** se debe informar al usuario sobre lo que esté sucediendo en cada etapa del uso del servicio con el fin de evitar confusiones.

- **Relación entre el sistema y el mundo real:** usar referencias del mundo real, pero en el entorno digital, usando conceptos que resulten familiares a los usuarios.
- **Control y libertad del usuario:** brindar control y libertad a través de opciones disponibles como “volver atrás” o “deshacer”.
- **Consistencia y estándares:** los elementos visuales y comportamientos que se usen en la interfaz deben ser uniformes y consistentes en todo el sistema.
- **Prevención de errores:** reducir la posibilidad de la aparición de errores por parte del usuario, ya sea con instrucciones claras, confirmaciones previas, entre otros.
- **Reconocimiento en lugar de memorización:** proporcionar contextos claros para facilitar el reconocimiento de elementos y evitar el tener que recordarlos.
- **Flexibilidad y eficiencia:** ofrecer distintas maneras para realizar una misma acción o tarea, ya sea para usuarios avanzados o con poca experiencia.
- **Diseño estético y minimalista:** evitar el exceso de información para lograr que los elementos claves sean más fáciles de identificar.
- **Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de errores:** incluir mensajes de error claros y concisos, y ayudar a los usuarios a que reconozcan y/o solucionen estos errores.
- **Ayuda y documentación:** ofrecer ayuda al usuario, siempre es útil proporcionar una guía práctica y/o una sección de preguntas frecuentes para solventar dudas.

1.8 Marco Conceptual

A continuación, se describen los conceptos clave que se utilizarán en el desarrollo del sistema:

Gamificación: es el proceso por el cual se incorporan las técnicas propias de los juegos a cualquier otro proceso para que sea más fácil poder realizarlo y/o comprenderlo. De esta manera, la gamificación en educación utiliza las mecánicas de los juegos para enriquecer y facilitar el aprendizaje (Smartmind, 2025).

Aplicación Web Progresiva (PWA): tipo de aplicación que combina lo mejor de las páginas web y de las aplicaciones móviles, siendo accesible desde navegadores web, pero también adaptable para ejecutarse en dispositivos móviles, con capacidad de instalación local y funcionamiento offline (García, 2025).

Lenguaje de Programación: es un sistema formal compuesto por símbolos y reglas sintácticas y semánticas que permite a los programadores escribir instrucciones comprensibles para una computadora. Estas instrucciones se traducen en código fuente que define algoritmos y procesos lógicos, facilitando la creación de software y aplicaciones (Concepto.de, 2025).

Base de Datos Relacional: una base de datos relacional organiza la información en tablas compuestas por filas y columnas, donde cada fila representa una entidad y cada columna un atributo. Las relaciones entre tablas se establecen mediante claves primarias y externas, lo que permite consultas eficientes y garantiza la integridad de los datos (IBM, 2021).

Arquitectura por Capas: el sistema estará estructurado bajo el enfoque de arquitectura multicapa, donde cada capa tiene una responsabilidad específica. Las capas principales incluyen: presentación, lógica del negocio, y acceso a datos. Este modelo promueve la mantenibilidad, reutilización del código y separación de responsabilidades (Pressman & Maxim, 2021). Además, facilita el trabajo en equipo al permitir que los integrantes trabajen en capas distintas de forma paralela.

Metodología de Desarrollo: es un conjunto de prácticas, técnicas y herramientas usadas por los equipos de desarrollo para planificar, diseñar, construir, probar e implementar software de una manera eficiente y efectiva. Además, se definen roles y responsabilidades dentro del equipo (Valtx, 2021).

Léxico: es el conjunto de palabras que son conocidas de un idioma (su vocabulario), dentro de un léxico también pueden encontrarse palabras provenientes de otras lenguas o incluso inventadas, con el objetivo de cubrir una necesidad expresiva (Fariás, 2024).

Interactivo: se refiere a la capacidad que tiene un sistema informático para involucrar la comunicación entre un usuario y el sistema, puede darse mediante la entrada de datos, la manipulación de elementos visuales y la respuesta del sistema antes estas acciones. Aquí el usuario no solo recibe información pasivamente, sino que también influye en el comportamiento del sistema mediante sus acciones (Glosarix, 2025).

Interfaz de Usuario (UI): es el conjunto de elementos visuales y funcionales con los que un usuario puede interactuar con un sistema o aplicación digital, estos elementos incluyen botones, menús, iconos, barras de desplazamiento, entre otros. La UI busca garantizar interacciones intuitivas y eficientes entre los usuarios y la tecnología (Aula CM, s.f.).

Diseño Responsivo: es un enfoque de diseño que hace que las páginas web se vean bien y funcionen de manera correcta en una variedad de dispositivos y tamaños de pantalla diferentes, ya sea en laptops, tabletas, teléfonos móviles, entre otros; y usando un solo diseño (Lenovo, s.f.).

API: o por sus siglas, interfaz de programación de aplicaciones, es un conjunto de reglas o protocolos que permiten a las aplicaciones informáticas una comunicación entre sí para poder intercambiar datos, características o funcionalidades. Las API permiten compartir solo

la información que se desee y mantener ocultos detalles internos del sistema, lo cual ayuda en términos de seguridad (Goodwin, s.f.).

API Key: es una cadena alfanumérica que los desarrolladores usan para controlar el acceso a las API, estas claves son utilizadas para restringir y supervisar el acceso, identificando su uso autorizado (Amazon Web Services, s.f.).

Python: es un lenguaje de programación orientado a objetos de alto nivel, tiene una sintaxis fácil de leer, pero depende de la sangría, por lo que obliga a escribir código limpio. Python permite crear scripts, aplicaciones web, programas de aprendizaje automático, desarrollo de software, entre otros (Oracle, s.f.).

Pydantic: es un paquete de validación de datos para Python, permite definir modelos y verificar que los datos se ajusten al formato esperado antes de que sean procesados o almacenados (Allen, 2024).

Entorno de Desarrollo Integrado (IDE): es una aplicación de software que combina en un solo lugar todas las herramientas que son necesarias para desarrollar software. Ofrece un editor, depurador, compilador y funcionalidades de gestión de código genérico (DataScientest, 2022).

Backend: se refiere a la parte de una aplicación o sitio web que el usuario final no ve, pero que es esencial para su funcionamiento. Algunas de las responsabilidades del backend son la gestión de datos, lógica del negocio, seguridad, rendimiento, mantenimiento, administración y monitoreo (Ken, Backend: ¿Qué es y para qué sirve?, 2023).

Frontend: es la parte del desarrollo de una aplicación web o móvil que interactúa directamente con el usuario, se encarga de todo lo que los usuarios ven y tocan en una aplicación, usualmente tiene tres componentes principales: HTML (HyperText Markup Language), CSS (Cascading Style Sheets) y JavaScript (Initium Software, 2024).

Serialización: es el proceso de convertir un objeto en un flujo de bytes, con el objetivo de almacenar el objeto o transmitirlo a otro lugar, su propósito es mantener el estado de un objeto para cuando sea necesario, volver a crearlo (Microsoft, 2025).

Deserialización: es el proceso de convertir una estructura de datos o estado de un objeto almacenado en formato JSON, XML o binario en un objeto que sea utilizable en memoria, es una técnica usada para la transmisión de datos o almacenamiento de objetos (Imperva, s.f.).

OAuth2: es un estándar abierto de autorización delegada que permite a un usuario conceder a una aplicación de terceros, acceso a una información almacenada sin compartir credenciales, todo por medio de tokens de acceso (IBM, s.f.).

Formato RTF: es un formato de archivo para textos que es capaz de abrirse con varios programas de procesamiento de textos, es un formato avanzado que puede almacenar información y tipos de datos adicionales como estilo de fuente, formato e imágenes (Adobe, s.f.).

Object Relational Mapping (ORM): es una técnica de programación que permite usar código orientado a objetos para interactuar con bases de datos relacionales. Mediante un sistema de mapeo, los objetos se corresponden con tablas y campos de la base de datos. De esta forma un ORM automatiza operaciones de almacenamiento, consulta y recuperación de datos sin escribir consultas SQL explícitas (Oppermann & Whitfield, 2025).

Data Transfer Object (DTO): un patrón DTO tiene como finalidad crear objetos planos con una serie de atributos que puedan ser enviados o recuperados en una sola invocación. Es decir, es un patrón que es efectivo para transmitir información, permite crear estructuras de datos independientes de nuestro modelo de datos y permite controlar el formato, nombre y

tipos de datos que vamos a transmitir según algún requerimiento determinado (Reactive Programming, s.f.).

Requerimientos Funcionales: son pautas específicas que describen los comportamientos, funciones y operaciones que debe realizar un software o sistema. Se realizan para garantizar la claridad y alineación entre los equipos de desarrollo, las partes que estén interesadas y los usuarios finales (Visure Solutions, s.f.).

Requerimientos No Funcionales: son características y criterios relacionados a cómo debe ser en rendimiento, seguridad, usabilidad y otros aspectos, un sistema; todo más allá de su funcionalidad básica, es decir, se centran más en la calidad (Ken, 2023).

Unified Modeling Language (UML): es un lenguaje de modelado estandarizado que consiste en todo un conjunto de diagramas para ayudar a desarrolladores software a especificar, visualizar y documentar artefactos de los sistemas. Es una colección de mejores prácticas que utiliza principalmente notaciones gráficas para expresar el diseño de proyectos de software y ayuda a mejorar la comunicación entre los equipos de proyectos (Visual Paradigm, s.f.).

Diagrama de Casos de Uso: es una representación gráfica que describe cómo los usuarios interactúan con un sistema, estos diagramas muestran las relaciones que existen entre los actores (usuarios u otros sistemas) y casos de uso (funciones o servicios del sistema) (Gaskin, 2024).

Diagrama de Secuencia: es una representación gráfica que muestra la interacción de objetos de un sistema a lo largo del tiempo, se captura la secuencia de mensajes intercambiados entre objetos y orden en la que ocurren, presentándolos como líneas de vida verticales y flechas que son horizontales (Miro, s.f.).

CAPÍTULO 2: Selección de Metodología y Herramientas

2.1 Metodología de Desarrollo

Se consideraron dos metodologías de desarrollo para analizarlas y escoger trabajar con una, son:

- SCRUM
- XP (Extreme Programming)

2.1.1 SCRUM

Scrum se seleccionó como una opción puesto que es una metodología ágil, flexible, con enfoque iterativo y gran capacidad para maximizar la productividad y colaboración en el equipo. Scrum permite garantizar la calidad del producto y responder con eficacia a los posibles cambios que surjan durante el desarrollo. Según Schwaber y Sutherland, autores de La Guía Scrum, una buena implementación de esta metodología requiere de la correcta implementación de los siguientes principios:

- **Transparencia:** todos los aspectos del desarrollo, como avances en los módulos de juego, integración con Firebase o pruebas de usabilidad, deben ser visibles para cada uno de los miembros del equipo.
- **Inspección:** siempre se deben realizar revisiones periódicas del progreso de cada sprint, evaluando la funcionalidad, rendimiento, etc. Estas inspecciones lo que nos permiten es detectar errores o incluso desviaciones de los objetivos planteados.
- **Adaptación:** el equipo debe mantenerse preparado para ajustar los requerimientos y cada cambio se incorpora al backlog según el impacto que tenga en los objetivos del desarrollo.

2.1.1.1 Implementación de Scrum

- **Definición del Product BackLog:** se deben recopilar todas las características del prototipo, modos de juego, biblioteca de aprendizaje, autenticación de usuarios, etc.
- **Sprints:** dividir el desarrollo en ciclos en los que se entregan funcionalidades del producto, en este caso, conexión a la base de datos, creación de la interfaz principal, lógica de los juegos, etc.
- **Revisión de Sprint:** al final de cada sprint el equipo presenta el avance, mostrando el estado del prototipo y haciendo los ajustes necesarios según una retroalimentación.
- **Retrospectiva del Sprint:** el equipo reflexiona y analiza sobre los puntos a mejorar y se planifican acciones para los próximos sprints.

2.1.1.2 Beneficios de aplicar Scrum en el Desarrollo del Prototipo

- **Flexibilidad:** permite adaptarse a cambios que surgen en el desarrollo.
- **Entrega incremental:** se obtienen versiones funcionales del sistema en etapas tempranas.
- **Mejora continua:** debido a la retroalimentación de cada sprint se puede mejorar y refinar el prototipo con cada entrega.

2.1.2 Extreme XP

Según Ben (2024), Extreme Programming (XP) es una metodología de desarrollo con énfasis en la calidad técnica, colaboración estrecha y rápida adaptación a cambios en los requerimientos. Los valores que son fundamentales para la implementación de XP son la comunicación, simplicidad y retroalimentación; lo que permite centrarse en la mejora continua y en la entrega frecuente de versiones funcionales.

2.1.2.1 Puntos Clave de Aplicación de XP en el Prototipo

- **Comunicación:** el equipo debe mantener un diálogo constante, validando historias de usuario. Esta comunicación guía en el diseño funcional y técnico del sistema.
- **Simplicidad:** lo que se busca es priorizar el desarrollo de funcionalidades mínimas viables, evitando características no mandatorias.
- **Retroalimentación:** cada iteración realizada debe incluir sesiones de validación para que los comentarios obtenidos se integren en el siguiente ciclo.
- **Coraje:** el equipo de trabajo es capaz de tomar decisiones audaces como, por ejemplo, refactorizar módulos de juego. Dando mayor importancia a la calidad sobre la comodidad técnica.
- **Respeto:** cada integrante del equipo debe valorar el trabajo del otro, fomentando la programación en pareja y la revisión constante del código.

2.1.2.2 Implementación de XP

- **Historias de usuario:** se deberán documentar las funcionalidades del prototipo desde la perspectiva del usuario final, como “registrarse” o “jugar nivel”. Lo que ayuda a planificar y estimar mejor el esfuerzo.
- **Iteraciones cortas:** se organiza el desarrollo en ciclos cortos de 1 a 2 semanas, donde siempre se entrega una versión funcional, ya sea el módulo de la biblioteca de aprendizaje, el módulo de autenticación y control de acceso, etc.
- **Programación en pareja:** dos desarrolladores colaboran en tiempo real para poder escribir y revisar el código, esto permite mejorar la calidad y transferir el conocimiento más fácilmente.

2.1.2.3 Beneficios de Aplicar XP en el Desarrollo del Prototipo

- **Alta calidad del código:** la práctica de programación en pareja y la refactorización continua garantiza un sistema robusto.
- **Adaptabilidad:** los ciclos cortos permiten responder de manera rápida ante cambios en los requerimientos.
- **Colaboración efectiva:** la programación en pareja y su beneficio de retroalimentación constante fortalece la comunicación y cohesión del equipo de trabajo.
- **Entrega continua:** se obtienen versiones funcionales desde etapas bastante tempranas, lo que permite validar la funcionalidad.

2.1.3 Análisis de la Metodología de Desarrollo

Se realizó un análisis de las metodologías tentativas para el desarrollo del prototipo, destacando sus características, detalladas en la Tabla 1. El objetivo fue tomar una decisión óptima sobre cuál es la más adecuada para nuestro caso.

Tabla 1

Comparativa de características de metodologías de desarrollo de software

Características	Scrum	XP
Enfoque	Ágil e iterativo, se centra en la gestión y la productividad del equipo.	Ágil, centrado en la calidad técnica y mejora continua del código.
Aplicación	Proyectos de desarrollo con entregas incrementales.	Proyectos de pequeños a medianos.
Ciclos de Trabajo	Sprints de 2 a 4 semanas.	Iteraciones de 1 a 2 semanas.

Roles Principales	Product owner, scrum master, equipo de desarrollo.	Equipo de desarrollo, cliente y coach técnico.
Artefactos Principales	Product backlog, sprint backlog, incremento.	Historias de usuario, código aprobado, entregas funcionales.
Reuniones clave	Sprint planning, daily scrum, sprint retrospective.	Reuniones de planificación, pruebas continuas, revisiones de código y retroalimentación.
Técnica de modelado	Entidad-Relación y diagramas UML.	Casos de uso, diagramas simples y pruebas unitarias.
Personalización	Alta.	Alta.
Etapas	Inicio, planificación, implementación, revisión y retrospectiva.	Planificación de la versión, diseño simple, codificación, pruebas y refactorización.
Nivel de documentación	Moderado, con reportes de avance por sprint.	Bajo a moderado, prioriza el código.
Orientación al Cliente	Alta.	Muy alta.

Nota. En esta tabla se muestra una comparativa entre las características de las metodologías SCRUM y XP para tener una mejor idea de cómo es el funcionamiento de cada una. Autores: Andy Contento y Oscar Figueroa (2025).

Con el fin de determinar la mejor metodología para el desarrollo del prototipo, se procedió a puntuar cada una de las características de ambas metodologías en la Tabla 2.

Tanto SCRUM como XP resultan bastante compatibles con el desarrollo que se quiere lograr, ambas tienen entregas frecuentes, flexibilidad y trabajo colaborativo, por ende, se debe hilar más profundo. SCRUM ofrece una buena organización y seguimiento de los avances mediante los llamados sprints y una muy buena revisión, lo que facilita

enormemente el control de los tiempos; XP, por su parte, se destaca mayormente por su gran énfasis en la calidad técnica y la programación en pareja, lo cual también es ventajoso, pero acaba siendo demasiado centrado en el código.

Tabla 2

Comparativa de metodologías de desarrollo de software

Criterio	Scrum	XP
Fiabilidad	5	5
Agilidad	5	5
Calidad técnica del código	4	5
Complejidad de implementación	3	2
Flexibilidad antes cambios	5	5
Escalabilidad (web y móvil)	5	4
Relación con el cliente/tutor	5	5
Documentación requerida	3	2
Colaboración y comunicación	5	5
Adecuación a equipos pequeños	4	5
TOTAL	44	43

Nota. Escala de valoración: 1 = Muy baja, 5 = Muy alta. Autores: Andy Contento y Oscar Figueroa (2025).

Puesto que el equipo para el desarrollo del prototipo consta de dos personas y la metodología debe equilibrar calidad, organización, entrega continua y simplicidad, la metodología que se seleccionó fue SCRUM, ya que presenta un gran balance entre todo lo mencionado.

3.2 Análisis de las Herramientas de Desarrollo

Para iniciar con el desarrollo del prototipo de manera eficiente, se optó por las siguientes herramientas:

- PostgreSQL (Base de datos).
- DBeaver (Sistema de gestión de bases de datos).

- Power Designer (Modelado de base de datos)
- Visual Studio Code (IDE).
- Lucidchart (Diagramas de casos de uso, diagramas de secuencia).
- Figma (Diseño de interfaz)
- FastAPI (Framework de desarrollo para backend)
- Flutter (Framework de desarrollo para frontend)

3.2.1 PostgreSQL

Según Google Cloud (2025), PostgreSQL es una base de datos relacional de código abierto, fue diseñada para ofrecer rendimiento, fiabilidad y escalabilidad. Con el tiempo, PostgreSQL se ha consolidado como una solución robusta y altamente extensible; además, cumple con los principios ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad) y con la mayoría de estándares SQL, lo que garantiza validez en los datos incluso frente a fallos de hardware o red.

Se escogió PostgreSQL como base de datos porque ofrece una variedad de ventajas tales como:

- **Fiabilidad y recuperación ante desastres:** mantiene la durabilidad de los datos con mecanismos como WAL (Write-Ahead Logging) es decir, registra los datos antes de aplicarlos a la base de datos, replicación lógica y recuperación a un punto en el tiempo.
- **Seguridad avanzada:** tiene una autenticación segura y control granular de acceso.
- **Rendimiento y concurrencia:** optimiza las consultas con índices avanzados y soporta múltiples CPUs en paralelo.
- **Cumplimiento normativo:** cumple con la mayoría de los estándares SQL y requisitos ACID, lo que lo posiciona como una de las bases de datos más confiables.

Estas características hacen que PostgreSQL sea una excelente opción como base de datos para el prototipo.

3.2.1.1 Base de Datos

En esta sección lo que se realizó es una comparación entre dos bases de datos PostgreSQL y MySQL. Ambas opciones son sólidas, pero se analizaron con el objetivo de determinar cuál es la más adecuada para el desarrollo del prototipo, se detallaron sus características en la Tabla 3.

Tabla 3

Comparativa de características de bases de datos

Características	PostgreSQL	MySQL
Tipo de sistema	ORDBMS (Object-Relational Database Management System).	RDBMS (Relational Database Management System).
Modelo de datos	Soporta datos relacionales y objetos complejos.	Relacional, con soporte limitado a JSON.
Licencia	Open Source.	Open Source.
Lenguaje de consultas	SQL estándar con extensiones avanzadas.	SQL estándar con funciones simplificadas.
Soporte de concurrencia	MVCC (Multiversion Concurrency Control) nativo.	MVCC a través de InnoDB, menor eficiencia en cargas pesadas.
Integridad referencial	Alta, con soporte para claves foráneas, triggers y reglas.	Media, depende del motor de almacenamiento utilizado.
Escalabilidad y rendimiento	Ideal para proyectos complejos con alto volumen de datos y transacciones.	Muy eficiente en lectura y consultas simples, tiene

		menor rendimiento en operaciones masivas.
Seguridad	Soporta autenticación por roles, cifrado SSL, certificados y políticas granulares.	Autenticación básica, soporte SSL y seguridad dependiente del servidor.
Compatibilidad con frameworks	Compatible con ORM modernos.	Compatible, en especial con PHP, Laravel y WordPress.

Nota. Fuente: Adaptado de PostgreSQL Global Development Group (2025) y Oracle (2025).

Continuando con el análisis y con el objetivo de determinar la mejor base de datos para el desarrollo del prototipo, se procedió a puntuar cada una de las características de ambas en la Tabla 4.

Tabla 4

Comparativa de bases de datos

Criterio	PostgreSQL	MySQL
Rendimiento en lecturas y escrituras	5	4
Manejo de concurrencia	5	4
Escalabilidad y carga de datos	5	4
Compatibilidad con frameworks modernos	5	4
Soporte para JSON y estructuras complejas	5	3
Seguridad y control de acceso	5	4
Fiabilidad y cumplimiento ACID	5	4
Mantenimiento y extensibilidad	5	3
Facilidad de uso inicial	3	5
Comunidad y Documentación	5	5
TOTAL	48	40

Nota. Escala de valoración: 1 = Muy baja, 5 = Muy alta. Fuente: Adaptado de PostgreSQL

Global Development Group (2025) y Oracle (2025).

En base a la evaluación que se realizó y bajo el criterio de los puntajes, PostgreSQL se posiciona como la mejor opción con 48 puntos, y todo debido a su soporte avanzado para datos relacionales, su compatibilidad con frameworks modernos y la capacidad que tiene en cuanto a rendimiento. De esta manera, se escogió a PostgreSQL para el desarrollo del prototipo.

3.2.2 DBeaver

Según DBeaver (s.f), esta herramienta de gestión de bases de datos es gratuita y de código abierto, diseñada para quienes requieren una interfaz que sea potente, flexible y multiplataforma. Su objetivo esencial es el de la usabilidad, lo que queda evidenciado en su interfaz cuidadosamente diseñada para facilitar la administración de datos y estructuras complejas. Su arquitectura extensible permite a los usuarios personalizar y ampliar la herramienta según las necesidades de cada usuario, lo que lo convierte en una herramienta robusta para entornos técnicos y académicos que demandan eficiencia, versatilidad y control.

De acuerdo con DBeaver (s.f.), la herramienta ofrece diversas ventajas:

- **Compatibilidad con múltiples fuentes de datos:** DBeaver permite la conexión con bases de datos como PostgreSQL, MongoDB, Cassandra, etc. Lo cual es ideal para trabajar con distintos tipos de almacenamiento.
- **Diagramas ER automáticos:** la herramienta genera diagramas entidad-relación, lo que ayuda a comprender mejor la estructura de las bases de datos.
- **Editor SQL:** DBeaver ayuda con la escritura y organización de scripts SQL y permite construir visualmente consultas complejas sin necesitar conocimientos avanzados en SQL.

- **Seguridad avanzada:** la herramienta incluye soporte para autenticación, almacenamiento cifrado de credenciales, lo que permite una gestión de accesos segura.

Estas son las características por las cuales se eligió DBeaver como sistema de gestión de base de datos para el desarrollo del prototipo.

3.2.2.1 Sistema gestor de bases de datos.

En esta sección se realizó un análisis de los sistemas gestores de bases de datos para el desarrollo del prototipo, comparando sus características, detalladas en la Tabla 5. El objetivo fue tomar una decisión acertada.

Tabla 5

Comparativa de características entre sistemas gestores de bases de datos

Características	DBeaver	pgAdmin
Tipo de herramienta	Cliente universal de administración de bases de datos.	Herramienta oficial de administración de PostgreSQL.
Compatibilidad con sistemas	Multibase de datos (PostgreSQL, Oracle, MongoDB, etc.).	Exclusiva para PostgreSQL y servidores derivados.
Licencia	Open Source.	Open Source bajo licencia de PostgreSQL.
Interfaz de usuario	Moderna, intuitiva y multiplataforma.	Interfaz web y de escritorio, no muy moderna, aunque estable,
Gestión de conexiones	Permite múltiples conexiones al mismo tiempo a diferentes SGBD.	Conexión única enfocada en entornos que sean PostgreSQL.

Visualización de datos	Panel avanzado con filtros, exportaciones, diagramas y resultados tabulares que son dinámicos.	Visualización básica de datos tabulares.
Generación de diagramas	Incluye herramientas ERD y visualización de relaciones entre tablas.	Generación de diagramas que es limitado y menos personalizable.
Soporte para SQL y scripting	Compatible con SQL, PL/pgSQL, Python y scripts personalizados.	Orientado a PL/pgSQL.
Rendimiento	Optimizado para grandes volúmenes de datos.	Rendimiento adecuado pero limitado a PostgreSQL.
Requisitos de instalación	Ligero, instalación rápida y configurable.	Pesado, requiere dependencias y configuración adicional.
Nivel de personalización	Alto: admite temas, extensiones y configuraciones avanzadas.	Medio: interfaz rígida con menos opciones visuales.

Nota. Fuente: Adaptado de pgAdmin Development Team (2025) y DBeaver (s.f.).

Continuando con el análisis y con el objetivo de determinar el mejor sistema gestor de base de datos para el desarrollo del prototipo, se procedió a puntuar cada una de las características de ambos en la Tabla 6.

Tabla 6

Comparativa de sistemas gestores de bases de datos

Criterio	DBeaver	pgAdmin
Compatibilidad con múltiples bases de datos	5	3
Interfaz y usabilidad	5	4
Rendimiento y estabilidad	5	4

Gestión visual de datos y relaciones	5	4
Generación de diagramas ERD	5	3
Soporte para scripting y consultas SLQ	5	4
Personalización y extensibilidad	5	3
Curva de aprendizaje	4	5
Exportación e importación de datos	5	4
Integración con PostgreSQL	5	5
TOTAL	49	39

Nota. Escala de valoración: 1 = Muy baja, 5 = Muy alta. Fuente: Adaptado de pgAdmin Development Team (2025) y DBeaver (s.f.).

Ambas herramientas son potentes para la gestión de bases de datos, pero bajo los puntajes de la comparación DBeaver se destaca con 49 puntos, debido a su interfaz moderna, flexible, compatible con múltiples motores de bases de datos y facilidad para gestionar diferentes conexiones al mismo tiempo. Estas características hicieron que se escogiera a DBeaver como sistema gestor de base de datos para el desarrollo del prototipo.

3.2.3 Power Designer

Según Novalys (2024), Power Designer es una herramienta de modelización de datos que permite diseñar, visualizar y gestionar arquitecturas de información que es compleja. La herramienta facilita la creación de modelos conceptuales, lógicos y físicos. La característica que más distingue a Power Designer es su enfoque en la modelización en la modelización estructurada de datos, en la que cada entidad, relación y atributo se representa de manera visual y formal.

Se escogió Power Designer para modelar la base de datos de este proyecto porque ofrece diversas ventajas, como:

- Modelización completa: la herramienta soporta modelos de datos conceptuales, lógicos y físicos, lo que permite representar estructuras que sean complejas con mayor precisión.
- Generación de documentación: ofrece la posibilidad de crear reportes HTML y RTF, lo que permite documentar de forma estructurada.
- Soporte multiplataforma: es compatible con más de 60 sistemas de gestión de bases de datos.

Estas razones hicieron que Power Designer se use para modelar la base de datos del prototipo.

3.2.3.1 Herramienta para el modelado de base de datos

En esa sección se realizó el análisis de las herramientas para diseñar la base de datos para el desarrollo del prototipo, comparando sus características, detalladas en la Tabla 7.

Tabla 7

Comparativa de características entre herramientas de diseño de bases de datos

Características	Power Designer	ER/Studio
Tipo de herramienta	Herramienta integral para el modelado empresarial y de datos.	Plataforma profesional para el modelado de datos.
Compatibilidad de bases de datos	Amplia, incluye: PostgreSQL, SQL Server, Oracle, entre otras.	Amplia y enfocada en SQL Server, Oracle y DB2.
Modelos soportados	Conceptual, lógico, físico y arquitecturas empresariales complejas.	Conceptual, lógico, físico.

Documentación automática	Genera reportes en Word, HTML, PDF.	Genera reportes con menor personalización.
Interfaz de usuario	Moderna, intuitiva y vistas que se pueden personalizar.	Tradicional, técnica y menos intuitiva.
Generación de código SQL	Automática, con plantillas adaptables.	Generación correcta, no muy flexible
Integración con otras herramientas	Compatible con IDEs, repositorios Git y herramientas de arquitectura empresarial.	Enfocada en entornos IDERA.
Requisitos del sistema	Ligero y estable.	Demandante en recursos e instalación compleja.

Nota. Fuente: Adaptado de Novalys (2024) e IDERA (2025).

Para continuar con el análisis y determinar la mejor herramienta para el diseño de la base de datos para el desarrollo del prototipo, se procedió a puntuar cada una de las características de ambas herrameintas en la Tabla 8.

Tabla 8

Comparativa de herramientas de modelado de bases de datos

Criterio	Power Designer	ER/Studio
Compatibilidad con SGDB	5	4
Interfaz y facilidad de uso	5	4
Ingeniería directa e inversa	5	4
Generación automática de código SQL	5	4
Capacidad de documentación técnica	5	4
Análisis de impacto y trazabilidad	5	4
Colaboración de quipo y control de versiones	5	4

Integración con arquitecturas empresariales	5	3
Rendimiento y estabilidad	5	4
Disponibilidad académica y licencias educativas	5	3
TOTAL	50	38

Nota. Escala de valoración: 1 = Muy baja, 5 = Muy alta. Fuente: Adaptado de Novalys (2024) e IDERA (2025).

En base a las características y puntuaciones dadas, Power Designer toma la delantera con 50 puntos, debido a su amplia compatibilidad con diferentes sistemas de gestión de bases de datos, su enfoque visual y colaborativo. Por estos motivos, se escogió a Power Designer como la herramienta para el diseño de la base de datos para el desarrollo del prototipo.

3.2.4 Visual Studio Code

De acuerdo con Cimas (2022), Visual Studio Code es un editor de código desarrollado por Microsoft que ofrece una experiencia de desarrollo ágil, ligera y personalizada. La herramienta permite adaptar el entorno de acuerdo con las necesidades específicas de de cada usuario.

Se escogió Visual Studio Code como IDE de desarrollo puesto que tiene características bastante completas, de las cuales podemos destacar:

- **Ligereza y rapidez:** es más liviano que otros IDEs, lo que permite una ejecución ágil incluso con un hardware limitado. Todo esto permite reducir los tiempos de espera en proyectos medianos y grandes.
- **Depuración integrada:** permite ejecutar y depurar el código directamente desde el editor, y además soporta lenguajes como JavaScript, Python, C++, entre otros.

- Control de versiones con Git: tiene integración nativa con Git, lo que permite gestionar nuestros repositorios, realizar commits y visualizar los cambios en el propio IDE.
- Interfaz intuitiva: posee un diseño claro y personalizable, lo que facilita mucho la curva de aprendizaje, permitiendo familiarizarse rápidamente con el entorno.

Estas características llevaron a elegir como IDE a Visual Studio Code para el desarrollo del prototipo, una opción optimizada para distintos tipos de experiencia de usuarios.

3.4.1.1 Entorno de Desarrollo Integrado (IDE)

En esta sección se realizó una comparación entre dos IDEs de desarrollo, Visual Studio Code y Eclipse. Se analizaron con el objetivo de determinar qué herramienta es la más adecuada para el desarrollo del prototipo, mencionando sus características en la Tabla 9.

Tabla 9

Comparativa de características entre entornos de desarrollo integrado (IDE)

Características	Visual Studio Code	Eclipse IDE
Tipo de herramienta	Editor de código ligero, extensible y multiplataforma.	Entorno de desarrollo orientado a JAVA y proyectos empresariales.
Lenguajes compatibles	Más de 30: JavaScript, TypeScript, Python, Dart, etc.	Java, C/C++, Python y otros mediante plugins.
Licencia	Open Source	Open Source
Interfaz de usuario	Moderna, personalizable y minimalista.	Tradicional, técnica y no muy intuitiva.
Extensibilidad	Amplia variedad de extensiones.	Menor catálogo de plugins con instalación compleja.

Rendimiento	Ligero y rápido.	Pesado, consume más recursos en memoria.
Depuración	Depurador integrado de múltiples lenguajes y frameworks.	Excelente depurador para Java y limitado en otros entornos.
Integración con control de versiones	Integración nativa con Git y GitHub.	Compatible mediante plugins.
Compatibilidad multiplataforma	Windows, macOS, Linux.	Windows, macOS, Linux.
Integración con frameworks modernos	Compatible con Flutter, Node.js, Angular, etc.	Orientado a Java y J2EE.
Terminal integrada	Sí y con múltiples sesiones.	Sí.
Soporte para desarrollo web y móvil.	Excelente, compatible con frameworks como Flutter, React Native, entre otros.	Limitado para desarrollo móvil o multiplataforma.
Curva de aprendizaje	Baja, interfaz que es intuitiva.	Media, requiere configuración inicial.
Comunidad y documentación	Muy activa, con soporte global.	Activa, pero orientada a Java.

Nota. Fuente: Adaptado de Microsoft (2025) y Eclipse Foundation (2025).

Para continuar con el análisis y con el objetivo de determinar la mejor opción, se procedió a puntuar ambas herramientas en la Tabla 10.

Tabla 10

Comparativa de entornos de desarrollo integrado

Criterio	Visual Studio Code	Eclipse
Compatibilidad con lenguajes y frameworks	5	4

Interfaz y facilidad de uso	5	3
Rendimiento y consumo de recursos	5	3
Integración con Git y control de versiones	5	4
Extensiones y personalización	5	4
Depuración y pruebas	5	4
Compatibilidad con Flutter y FastAPI	5	3
Curva de aprendizaje	5	4
Comunidad y soporte	5	4
Adaptabilidad al entorno multiplataforma (web/móvil)	5	3
TOTAL	50	36

Nota. Fuente: Adaptado de Microsoft (2025) y Eclipse Foundation (2025).

Después de haber analizado las características de ambas herramientas, Visual Studio Code, junto con sus 50 puntos en la tabla comparativa, es la que más se alinea con los necesidades para el desarrollo del prototipo, debido a su rendimiento superior, flexibilidad, compatibilidad multiplataforma y soporte extendido; todas estas características hicieron que sea el IDE elegido.

3.2.5 Lucidchart

De acuerdo con Oyarzún (2023), Lucidchart es una herramienta de diagramación colaborativa que permite crear representaciones de manera visual de los procesos o sistemas y permite a múltiples usuarios editar diagramas desde distintos dispositivos.

Uno de los motivos por las que se escogió a Lucidchart para el desarrollo del prototipo fue por su gran capacidad para generar diagramas de secuencia y casos de uso de manera rápida y comprensible. Este tipo de diagramas son de importancia porque permiten visualizar el

flujo de interacción de los componentes y actores involucrados, y esta herramienta ofrece plantillas, facilitando el desarrollo de estos diagramas. Asimismo, Lucidchart emplea el Lenguaje Unificado de Modelado (UML), lo que garantiza que los diagramas elaborados tengan una estructura comprensible y estandarizada.

Por estas razones, se seleccionó a Lucidchart como la herramienta a utilizar en la elaboración de diagramas del prototipo, debido a su enfoque colaborativo, su estandarización y su capacidad para representar visual y claramente los componentes de un sistema.

3.2.5.1 Herramienta de diagramación

En esta sección se realizó una comparación entre dos herramientas para diagramar los casos de uso para el desarrollo del prototipo. Entonces, con el objetivo de determinar cuál se ajusta más a las necesidades, se detallaron las características en la Tabla 11.

Tabla 11

Comparativa de características entre herramientas de diagramación

Características	Lucidchart	Draw.io
Tipo de herramienta	Plataforma profesional para diagramación en línea.	Herramienta gratuita de diagramación en línea y escritorio.
Licencia	Comercial con una versión gratuita limitada.	Gratuita.
Acceso y disponibilidad	Es 100% web accesible desde cualquier navegador.	Tiene versión web y de escritorio.
Colaboración en tiempo real	Sí, posee edición simultánea y con comentarios.	Limitada, solo permite compartir archivos.
Integración con servicios en la nube.	Integración con Google Drive, OneDrive, Slack, Jira, Github, etc.	Compatible con Google Drive y Github.

Plantillas y recursos visuales	Posee una amplia biblioteca de plantillas.	Repositorio básico y menos estandarizado.
Soporte UML	Completo y actualizado	Notación UML parcial.
Interfaz de usuario	Moderna, intuitiva y orientada a la colaboración.	Interfaz funcional pero no muy fluida, simple.
Velocidad y rendimiento	Fluido con diagramas grandes y colaborativos.	No optimizado para diagramas complejos.
Exportación de diagramas	PNG, PDF, SVG, JPEG, VDX, HTML interactivo.	PNG, PDF, SVG, XML.
Gestión de proyectos y carpetas	Permite organizar en carpetas y etiquetas.	Se basa en archivos individuales.
Integración académica y profesional.	Ampliamente utilizada en entornos educativos y corporativos, con versión académica.	Entornos personales o académicas informales.

Nota. Fuente: Adaptado de Lucid Software (2025) y draw.io (2025).

Para continuar con el análisis y con el objetivo de determinar a la herramienta más adecuada para el desarrollo, se procedió a puntuar las características de ambas en la Tabla 12.

Tabla 12

Comparativa de herramientas de diagramación

Criterio	Lucidchart	Draw.io
Facilidad de uso e interfaz	5	4
Colaboración en tiempo real	5	3
Compatibilidad con UML	5	3
Disponibilidad multiplataforma	5	4
Integración en la nube y herramientas externas	5	4
Plantillas y recursos visuales	5	4
Rendimiento con diagramas grandes	5	4

Exportación de formatos	5	4
Gestión y organización de proyectos	5	3
Uso académico y profesional	5	4
Total	50	38

Nota. Fuente: Adaptado de Lucid Software (2025) y draw.io (2025).

En base al análisis de sus características y la puntuación obtenida, Lucidchart obtiene 50 puntos, debido a la visión colaborativa que ofrece, sus estándares y su optimización en línea.

Por estos motivos se escogió a Lucidchart como herramienta para la diagramación para el desarrollo del prototipo.

3.2.6 Figma

Stevens (2024) explica que Figma es una herramienta de diseño y prototipado colaborativo.

La plataforma permite trabajar en línea y en tiempo real, lo que es de gran ayuda para equipos en la construcción de interfaces. Se escogió a Figma como herramienta de diseño de la interfaz del prototipo porque ofrece características muy completas, como por ejemplo:

- **Colaboración en tiempo real:** varios usuarios pueden colaborar editando y comentando al mismo tiempo, lo que agiliza la comunicación entre equipos.
- **Accesibilidad multiplataforma:** es una aplicación basada en la nube, lo que permite acceder desde cualquier dispositivo con conexión a internet.
- **Prototipado interactivo:** es capaz de permitir la creación de flujos de navegación para simular la experiencia de usuario, lo que permite validar ideas antes de implementarlas.
- **Interfaz intuitiva:** tiene un diseño claro y adaptable que reduce la curva de aprendizaje, por ende, permite familiarizarse rápidamente con la herramienta.

Todas las características mencionadas llevaron a escoger a Figma como herramienta del diseño de la interfaz del usuario.

3.2.6.1 Herramienta para el Diseño de la Interfaz

En esta sección se realizó una comparación entre dos herramientas para el diseño de la interfaz del prototipo, Figma e InVision. Se analizarán con el objetivo de determinar cuál es la más adecuada para el desarrollo, de detallaron sus características en la Tabla 13.

Tabla 13

Comparativa de características entre herramientas de diseño de interfaces

Características	Figma	InVision
Tipo de herramienta	Plataforma colaborativa de diseño y para prototipado de interfaces UI/UX en la nube.	Herramienta de diseño y para prototipado orientada a la presentación de interfaces.
Licencia	Freemium.	Freemium.
Acceso y disponibilidad	Es 100% en la nube accesible desde cualquier navegador.	Basado en la nube y con aplicación de escritorio opcional.
Colaboración en tiempo real	Sí, edición al mismo tiempo y comentarios en línea.	Permite comentarios, pero la edición colaborativa es limitada.
Prototipado interactivo	Soporta animaciones, transiciones, flujo de pantallas.	Se enfoca en la presentación, con animaciones que son básicas.
Compatibilidad con frameworks de desarrollo	Integración con Flutter, React, Vue y Android Studio.	Integración con Sketch y Adobe XD.
Componentes reutilizables y bibliotecas	Creación de componentes, estilos compartidos y bibliotecas colaborativas.	Permite componentes, pero no se sincronizan

		dinámicamente entre proyectos.
Interfaz de usuario	Moderna, fluida y fácil de usar.	Técnica, centrada en la revisión y presentación.
Integraciones	GitHub, Slack, Zeplin, Trello, entre otros.	Sketch, Slack, Jira, Trello.
Velocidad y rendimiento	Alta y fluida con múltiples colaboradores.	Depende del tamaño del proyecto.
Herramientas complementarias	FigJam	No cuenta con herramientas nativas.
Exportación de recursos	PNG, SVG, PDF, CSS, XML.	PNG, JPG, PDF.
Orientación de uso	Equipos de diseño y desarrollo colaborativo.	Presentaciones de prototipos y validación visual.

Nota. Fuente: Adaptado de Figma (2025) e InVision (2025).

Para continuar con el análisis y con el objetivo de determinar la mejor herramienta para el diseño de la interfaz de usuario, se procedió a puntuar las características de ambas en la Tabla 14.

Tabla 14

Comparativa de herramientas de diseño de interfaz

Criterio	Figma	InVision
Facilidad de uso e interfaz	5	4
Colaboración en tiempo real	5	3
Prototipado interactivo	5	4
Compatibilidad con frameworks de desarrollo	5	3

Creación de componentes reutilizables	5	3
Integración con herramientas externas	5	4
Rendimiento y estabilidad	5	4
Bibliotecas de diseño compartidas	5	4
Curva de aprendizaje	5	4
Uso académico y profesional	5	4
TOTAL	50	38

Nota. Fuente: Adaptado de Figma (2025) e InVision (2025).

3.2.7 FastAPI

De acuerdo con FastAPI (s.f.-a), esta herramienta es un framework web moderno de alto rendimiento para construir APIs con Python, sus características son diversas, se puede mencionar su rapidez tanto en rendimiento, como en programación, ya que aumenta la velocidad de desarrollo de funcionalidades desde un 200% a 300%, es fácil de usar y aprender, minimiza la duplicación de código, es robusto y reduce en un 40% los errores cometidos por desarrolladores humanos.

Por otro lado, se tiene sus funcionalidades, FastAPI (s.f.-b) nombra a las siguientes:

- **Documentación automática:** interfaces web de documentación y exploración de APIs interactivas.
- **Soporte de editor:** es un framework fácil e intuitivo. Se proporciona un autocompletado en todas partes para no tener que regresar a ver documentación.
- **Seguridad y autenticación:** incluye esquemas varios esquemas de seguridad como http básico, OAuth2 y API keys.

Las características y funcionalidades de FastAPI hicieron que se escoja a esta herramienta como el framework para el desarrollo del backend del prototipo.

3.2.7.1 Framework de desarrollo para backend

En esta sección se compararon dos frameworks de desarrollo para backend, FastAPI y Sanic, listando sus características en la Tabla 15, con el objetivo de determinar cuál es la mejor opción para el desarrollo del prototipo.

Tabla 15

Comparativa de características frameworks para el desarrollo del backend

Características	FastAPI	Sanic
Lenguaje base	Python 3.7+	Python 3.7+
Paradigma	Framework moderno, asíncrono, se basa en tipado estático y una validación automática.	Framework asíncrono que se enfoca en la velocidad y la simplicidad.
Arquitectura	Basada en ASGI (Asynchronous Server Gateway Interface).	Basado en ASGI de manera minimalista.
Velocidad de ejecución	Muy alta, es compatible con Node.js y Go.	Muy alta, superior en operaciones simples.
Definición de rutas	Decoradores intuitivos y soporta	Decoradores simples sin sistema de dependencias.
Validación de datos	Automática mediante Pydantic.	Manual o con el uso de librerías externas.
Documentación automática	Generación automática interactiva con Swagger UI y ReDoc.	No genera documentación por defecto.
Manejo de errores y respuestas	Integración con modelos de respuesta y una validación automática.	Implementación manual.

Integración con bases de datos	Compatible con ORMs modernos.	Compatible con ORMs pero con configuración manual.
Autenticación y seguridad	Incluye OAuth2, JWT, CORS y dependencias de seguridad.	Se requiere de una implementación manual o librerías de terceros.
Compatibilidad con servicios externos	Alta, con Firebase, Docker, GraphQL y microservicios.	Alta pero poco soporte comunitario.
Escalabilidad y mantenimiento	Muy buena, con estructura modular y soporte nativo para microservicios.	Buena, aunque no tan estructurada para proyectos grandes.
Comunidad y documentación oficial	Amplia, documentación detallada y actualizada.	Comunidad limitada y documentación técnica.
Licencia	MIT (open source).	MIT (open source).

Nota. Fuente: Adaptado de FastAPI (s.f.) y Sanic Community (2025).

Continuando con el análisis y con el objetivo de seleccionar un framework de desarrollo para el backend del prototipo, se procedió a puntuar las características de ambos en la Tabla 16.

Tabla 16

Comparativa de frameworks para el desarrollo del backend

Criterio	FastAPI	Sanic
Velocidad y rendimiento	5	5
Facilidad de uso e implementación	5	4
Validación y tipado de datos	5	3
Documentación automática	5	2
Compatibilidad con bases de datos y ORMs	5	4
Autenticación y seguridad	5	3
Estructura y mantenibilidad del código	5	4
Escalabilidad y microservicios	5	4
Integración con otros servicios	5	4

Comunidad y soporte técnico	5	4
TOTAL	50	37

Nota. Fuente: Adaptado de FastAPI (s.f.) y Sanic Community (2025).

En base a las características vistas y la puntuación, FastAPI con 50 puntos es la que más se alinea con las necesidades del desarrollo del prototipo, debido a su alto rendimiento, validación automática de datos, generación de documentación en tiempo real y su gran facilidad para integrarlo con tecnologías modernas. Todos estos aspectos hicieron que se escoja a FastAPI como el framework para el desarrollo del backend del prototipo.

3.2.8 Flutter

Amazon Web Services (s.f.) explica que Flutter es un framework de código abierto desarrollado y compatible con Google, utilizado para crear interfaces de usuario (UI) para varias plataformas con un único código base. La herramienta utiliza el lenguaje de programación de código abierto llamado Dart, desarrollado por Google, y siendo un lenguaje perfecto para la detección de los llamados errores de nulos.

Flutter tiene ventajas muy claras:

- Rendimiento casi nativo: utiliza como lenguaje a Dart y compila el código en máquina, lo cual garantiza un rendimiento rápido y eficaz.
- Rendimiento rápido, consistente y personalizable: utiliza una biblioteca gráfica de código abierto llamada Skia de Google para poder renderizar la interfaz, lo proporciona visuales consistentes sin importar la plataforma que se utilice para acceder a la aplicación.
- Herramientas para desarrolladores: el objetivo de Flutter es que sea fácil de usar, es por eso que permite realizar recarga, es decir, previsualizar cambios de código sin perder el estado, y otras herramientas como el inspector de widgets, lo que permite visualizar y solucionar problemas con los diseños de la interfaz de usuario.

Las ventajas que ofrece Flutter hicieron que escogiéramos esta herramienta para el desarrollo del frontend del prototipo.

3.2.8.1 Framework para el desarrollo del frontend

En esta sección, se realizó una comparación entre dos frameworks de desarrollo para frontend, Flutter y React Native. Se analizaron con el objetivo de determinar cuál se ajusta más para el desarrollo del prototipo, detallando sus características en la Tabla 17.

Tabla 17

Comparativa de características de frameworks para el desarrollo del frontend

Características	Flutter	React Native
Lenguaje de programación	Dart	JavaScript / TypeScript
Tipo de framework	SDK completo con motor de renderizado propio.	Framework basado en JavaScript.
Arquitectura	Hace uso del motor Skia para renderizar su propia interfaz.	Depende de un bridge que conecta JS con componentes nativos.
Rendimiento	Muy alto: renderiza sin dependencias del sistema operativo.	Alto: con posibilidad de retrasos.
Interfaz de usuario	Cuenta con widgets personalizables y consistentes de alto rendimiento.	Depende de componentes nativos y con diferencias entre plataformas.
Compatibilidad multiplataforma	Android, iOS, web, escritorio.	Android, iOS y web.
Curva de aprendizaje	Media: se debe familiarizar con Dart.	Baja: se usan conocimientos en JavaScript.
Comunidad y soporte	Rápido crecimiento impulsado por Google.	Muy amplia y consolidada.

Ecosistema de librerías	Amplio y en expansión.	Muy extenso.
Hot reload	Inmediato y estable.	Funcional pero menos consistente.
Integración con APIs externas	Sencilla y compatible con REST, GraphQL y Firebase.	Amplia, ecosistema de Node.js y JavaScript.
Soporte para animaciones y gráficos	Avanzado, personalizable y fluido.	Limitado al rendimiento del bridge JS.
Empaquetado y despliegue	Produce binarios nativos que son de alto rendimiento.	Genera apps híbridas que dependen del bridge.
Actualizaciones y mantenimiento	Gestionadas por Google-	Mantenimiento por la comunidad y Meta.

Nota. Fuente: Adaptado de Flutter (2025) y Meta Platforms (2025).

Continuando con el análisis de los frameworks y con objetivo de determinar el mejor para el desarrollo del frontend del prototipo, se procedió a puntuar sus características en la Tabla 18.

Tabla 18

Comparativa de frameworks para el desarrollo del frontend

Criterio	Flutter	React Native
Rendimiento y estabilidad	5	4
Diseño visual y personalización UI	5	4
Compatibilidad multiplataforma	5	4
Integración con backend y servicios externos	5	5
Curva de aprendizaje	4	5
Comunidad y soporte	4	5
Mantenimiento y escalabilidad del código	5	4
Soporte para animaciones y gráficos	5	4
Empaquetado y despliegue	5	4
TOTAL	43	39

Nota. Fuente: Adaptado de Flutter (2025) y Meta Platforms (2025).

En base al análisis de las características y puntuación de los frameworks, Flutter con 43 puntos fue el mejor para el desarrollo del frontend del prototipo, debido a su rendimiento nativo, consistencia visual entre plataformas y soporte avanzado para animaciones, entre otros aspectos. Todas estas características hicieron que escogiéramos a Flutter para el desarrollo.

CAPÍTULO 3: Diseño del Prototipo del Sistema Web y Móvil

3.1 Análisis de Requerimientos

La identificación de requerimientos se llevó a cabo mediante la revisión de los objetivos del desarrollo prototipo planteados al inicio del documento. Adicionalmente, y como ya se mencionó a lo largo del documento, la gamificación es una buena herramienta para motivar el aprendizaje, por lo que se estableció que el prototipo debía integrar mecánicas como recompensas, niveles, logros y retroalimentación, lo cual se tomó en cuenta para definir los requerimientos funcionales.

En cuanto al enfoque técnico, los requerimientos no funcionales plantearon aspectos como la compatibilidad, escalabilidad y seguridad. Además, se basó en criterios de usabilidad, específicamente en las heurísticas de Nielsen número 1, 4, 5 y 8:

- Visibilidad del estado del sistema (heurística 1): el usuario debe recibir una retroalimentación visual y/o sonora sobre sus acciones.
- Consistencia y estándares (heurística 4): todos los íconos, botones y/o colores deben mantener una lógica visual uniforme a lo largo de todas las pantallas.
- Prevención de errores (heurística 5): el prototipo debe evitar errores mediante mensajes de confirmación antes de realizar acciones críticas.
- Estética y diseño minimalista (heurística 8): las pantallas deben mostrar solamente la información necesaria para evitar sobrecargar visual o textualmente.

Tras la identificación y documentación de los requerimientos tanto funcionales como no funcionales, se elaboró un acta de reunión bajo la metodología SCRUM (véase Anexo A y Anexo B) con el propósito de validar la conformidad. En el acta se listaron los requerimientos propuestos, los cuales fueron revisados con el ingeniero Jorge Alarcón. Posterior a la revisión y discusión, el ingeniero manifestó su conformidad con el documento

y firmó el acta como evidencia de aprobación de los requerimientos funcionales y no funcionales definidos.

3.2 Casos de Uso

En esta sección, se mostró con más claridad el funcionamiento del prototipo mediante diagramas de casos de uso elaborados con Lucidchart.

3.2.1 Requerimientos generales

Figura 1

Diagrama de caso de uso general



Nota. Autores: Andy Contento y Oscar Figueroa (2025).

3.2.1.1 Actores

- **Jugador:** usuario que se registra, juega niveles, ve la biblioteca y sus logros.

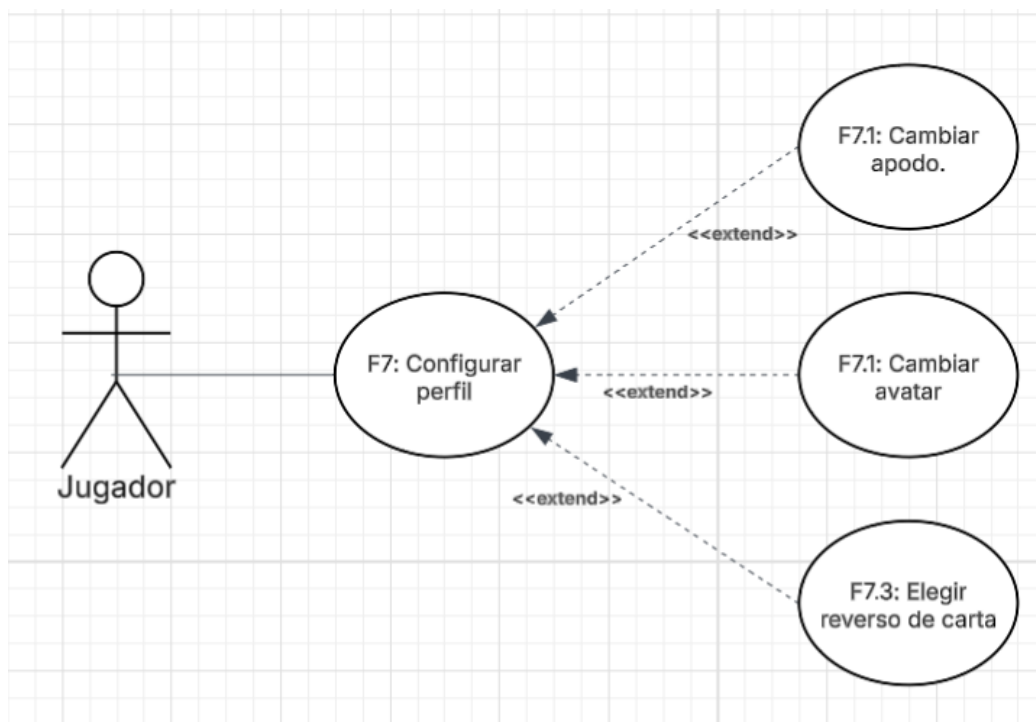
3.2.1.2 Descripción

Caso de uso general que define qué hace el prototipo desde la perspectiva de los actores externos como el jugador.

3.2.2 Requerimiento F7

Figura 2

Caso de uso requerimiento F7



Nota. Autores: Andy Contento y Oscar Figueroa (2025).

3.2.2.1 Actor

- Jugador.

3.2.2.2 Descripción

El caso de uso “Configuración de perfil (F7)” lo que muestra es cómo se permite al usuario realizar operaciones como cambiar apodo, cambiar avatar y elegir el reverso de carta.

3.2.2.3 F7.1: Cambiar Apodo

- **Actor:** jugador

- **Descripción:** permite modificar el apodo que es visible para el usuario en el juego.
- **Precondiciones:** usuario autenticado.
- **Flujo Principal:**
 1. El usuario ingresa a la “configuración de perfil”.
 2. El prototipo muestra el apodo actual y entrada para editar.
 3. El jugador ingresa su nuevo apodo.
 4. El prototipo valida el formato.
 5. El prototipo actualiza el apodo y confirma el cambio.

3.2.2.4 F7.2: Cambiar Avatar

- **Actor:** jugador
- **Descripción:** permite seleccionar un nuevo avatar entre los desbloqueados-
- **Precondiciones:** usuario autenticado.
- **Flujo Principal:**
 1. El jugador ingresa a la “configuración de perfil”.
 2. El prototipo muestra los avatares disponibles.
 3. El jugador selecciona un avatar.
 4. El prototipo guarda el nuevo avatar y confirma el cambio.

3.2.2.5 F7.3: Elegir Reverso de Carta

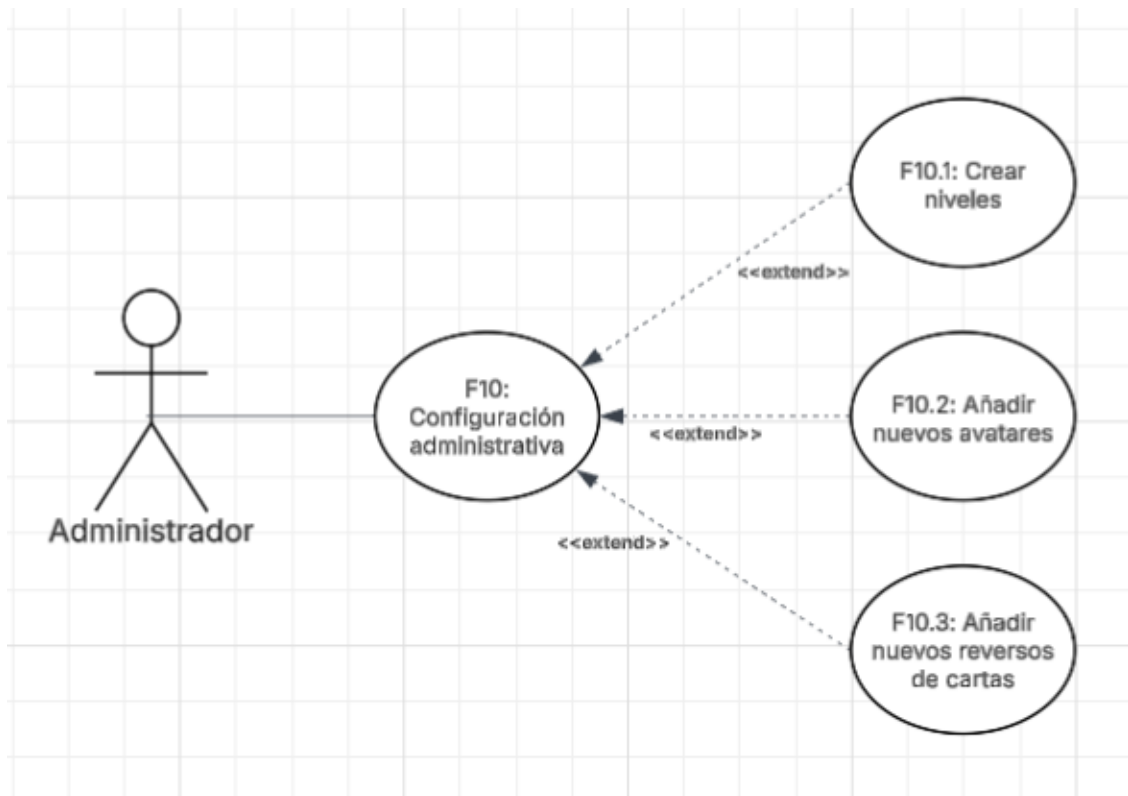
- **Actor:** jugador
- **Descripción:** permite seleccionar un nuevo avatar entre los desbloqueados-
- **Precondiciones:** usuario autenticado.
- **Flujo Principal:**
 1. El jugador ingresa a la “configuración de perfil”.
 2. El prototipo muestra los reversos de carta disponibles.

3. El jugador selecciona un reverso.
4. El prototipo registra la preferencia y confirma el cambio.

3.2.3 Requerimiento F10

Figura 3

Caso de uso requerimiento F10



Nota. Autores: Andy Contento y Oscar Figueroa (2025).

3.2.3.1 Actor

- Jugador

3.2.3.2 Descripción

El caso de uso “Configuración administrativa (F10)” lo que se muestra es cómo se permite al administrador gestionar el contenido del prototipo: niveles, avatares y reversos de carta.

3.2.3.3 F10.1: Crear niveles

- **Actor:** administrador.

- **Descripción:** permite crear nuevos niveles.
- **Precondiciones:** usuario autenticado con rol “administrador”.
- **Flujo Principal:**
 1. El administrador ingresa al módulo de administración y selecciona “Crear nivel”.
 2. El prototipo solicita los datos del nivel (nombre, modo de juego, dificultad, experiencia a ganar, monedas a ganar, número de vidas/intentos, límite de tiempo opcional).
 3. El administrador ingresa los datos solicitados y selecciona el conjunto de palabras que estarán en el nivel.
 4. El prototipo valida y confirma la operación.

3.2.3.4 F10.2: Añadir Nuevos Avatares

- **Actor:** administrador.
- **Descripción:** permite crear nuevos avatares.
- **Precondiciones:** usuario autenticado con rol “administrador”.
- **Flujo Principal:**
 1. El administrador ingresa al módulo de administración y selecciona “Crear Avatares”.
 2. El prototipo solicita los datos del avatar (nombre, seleccionar imagen y tipo de desbloqueo por monedas o nivel de experiencia).
 3. El administrador ingresa los datos solicitados.
 4. El prototipo valida y confirma la operación.

3.2.3.5 F10.3: Añadir Nuevos Reversos de Cartas

- **Actor:** administrador.

- **Descripción:** permite crear nuevos reversos de cartas.
- **Precondiciones:** usuario autenticado con rol “administrador”.
- **Flujo Principal:**
 1. El administrador ingresa al módulo de administración y selecciona “Crear Reversos de Cartas”.
 2. El prototipo solicita los datos del reverso de carta (nombre, seleccionar imagen y tipo de desbloqueo por monedas o nivel de experiencia).
 3. El administrador ingresa los datos solicitados.
 4. El prototipo valida y confirma la operación.

3.3 Requerimientos Funcionales

En esta sección se definieron los requerimientos funcionales que cumple el prototipo, son los siguientes:

- F1: El prototipo debe permitir registro e inicio de sesión mediante correo electrónico.
- F2: El prototipo debe permitir al usuario jugar niveles diseñados para el aprendizaje de vocabulario en kichwa e inglés.
- F3: El prototipo debe permitir al usuario pausar y reanudar una partida.
- F4: El prototipo debe mostrar retroalimentación visual y sonora al completar correcta o erróneamente un nivel.
- F5: El prototipo debe mostrar panel de clasificación global de los mejores jugadores, basado en su nivel de experiencia.
- F6: El prototipo debe permitir desbloquear reversos de cartas y avatares con monedas virtuales o puntos de experiencia.

- F7: El prototipo debe permitir al usuario acceder al módulo de configuración de perfil.
 - F7.1: El prototipo debe permitir al usuario cambiar su apodo.
 - F7.2: El prototipo debe permitir al usuario cambiar su avatar personal.
 - F7.3: El prototipo debe permitir elegir el reverso de carta que se utilizará.
- F8: El prototipo debe permitir la visualización de la biblioteca de aprendizaje, donde se muestran las cartas vistas y los errores cometidos.
- F9: El prototipo debe permitir visualizar los logros obtenidos por el usuario.
- F10: El prototipo debe permitir acceso al módulo de configuración administrativa.
 - F10.1: El prototipo debe permitir crear o niveles.
 - F10.2: El prototipo debe añadir nuevos avatares.
 - F10.3: El prototipo debe permitir añadir nuevos reversos de cartas.

3.4 Requerimientos No Funcionales

En esta sección se definieron los requerimientos no funcionales que cumple el prototipo:

Compatibilidad

- NF1: El prototipo debe ser compatible con dispositivos Android de la versión 5.0 en adelante.
- NF2: El prototipo debe tener un diseño responsivo, adaptando su interfaz correctamente a distintas resoluciones de pantalla.

Usabilidad

- NF3: El prototipo debe tener una interfaz intuitiva y colorida para niños de entre 4 y 12 años, empleando elementos interactivos.

- NF4: El prototipo debe cumplir con las heurísticas de usabilidad de Nielsen N.º 1, 4, 5 y 8:
 - Visibilidad del estado del sistema (heurística 1): el usuario debe recibir una retroalimentación visual y/o sonora sobre sus acciones.
 - Consistencia y estándares (heurística 4): todos los íconos, botones y/o colores deben mantener una lógica visual uniforme a lo largo de todas las pantallas.
 - Prevención de errores (heurística 5): el prototipo debe evitar errores mediante mensajes de confirmación antes de realizar acciones críticas.
 - Estética y diseño minimalista (heurística 8): las pantallas deben mostrar solamente la información necesaria para evitar sobrecargar visual o textualmente.
- NF5: El prototipo debe permitir que las acciones principales se completen en un máximo de tres clics, haciendo más fácil la navegación de los usuarios.

Seguridad

- NF6: El acceso al módulo administrativo debe ser restringido solo para usuarios con un rol autorizado para evitar modificaciones no deseadas.

Escalabilidad

- NF7: El prototipo debe ser escalable, permitiendo la futura incorporación de nuevos niveles o contenido adicional como nuevos avatares o reversos de cartas.

Como parte del proceso que define la metodología Scrum, se elaboró el Product Backlog del prototipo, en el cual se incluye una lista priorizada de las funcionalidades, características y requisitos que guiaron el desarrollo incremental. El backlog fue elaborado a partir de los requerimientos funcionales y no funcionales que se definieron, y desarrollado bajo el formato de historias de usuario (véase Anexo C y Anexo D para más información).

3.5 Planeación de la Arquitectura y Diseño

3.5.1 Diseño de la arquitectura del prototipo del sistema (microservicios y por capas).

3.5.1.1 Visión general

- Front-end: UI del usuario y panel del administrador.
- MS-Game (Microservicio del juego): propietario de toda la jugabilidad, conectado a la BD-Game.
- MS-Lex (Microservicio de léxico kichwa-inglés): propietario del diccionario de términos y traducciones, conectado a la BD-Lex.

3.5.1.2 Backend

Dentro del back, arquitectura del back tenemos las siguientes capas:

- Capa de presentación (API/Controllers)
- Capa de lógica de negocio (servicios)
- Capa de contratos/validación (DTOs)
- Capa de acceso a datos (ORM/modelos)
- Capa de Infraestructura/Core

3.5.1.2.1 Capa de presentación

Sus funciones consisten en:

- Recibir y validar peticiones HTTP con estructura genérica.
- Llamar a la capa de servicios.
- Retornar respuestas con estructura genérica.
- Gestionar códigos HTTP.

3.5.1.2.2 Capa de Lógica de Negocio

Sus funciones consisten en:

- Implementar reglas de negocio.
- Procesar lógica.
- Retornar DTOs.
- Interactuar con la capa de datos.

3.5.1.2.3 Capa de contratos/validación

Sus funciones son:

- Definir contratos de datos con Pydantic.
- Validar automáticamente los datos.
- Serializar y deserializar.
- Separar entre entrada y salida.

3.5.1.2.4 Capa de acceso a datos (OMR/modelos)

Sus funciones son:

- Mapeo objeto-relacional
- Definir relaciones entre tablas.
- Gestionar restricciones

3.5.1.2.5 Capa de Infraestructura

Sus funciones son:

- Configuración global.
- Gestión de conexiones a la BD.
- Seguridad y autenticación.

3.5.1.3 Frontend

3.5.1.3.1 Capa de Presentación

Sus funciones son:

- Manejar la interacción con el usuario.
- Escucha eventos, emite resultados y orquesta la navegación.

3.5.1.3.2 Capa de Dominio

Sus funciones son:

- Contener las reglas del negocio: entidades, interfaces de repositorios y casos de uso.
- Define qué se debe hacer y cuáles contratos deben cumplirse.

3.5.1.3.3 Capa de Datos

Sus funciones son:

- Implementar los repositorios declarados en la capa de dominio.
- Convertir los DTOs a entidades.

3.5.1.3.4 Capa Núcleo

Sus funciones son:

- Actuar como “caja de herramientas” compartida, sin ligarse a casos de uso específicos.
- Configuración, constantes, almacenamiento, errores, navegación.

3.5.2 Diseño de la Base de Datos del Prototipo del Sistema

Se decidió separar el almacenamiento en dos bases de datos porque el prototipo contiene dos dominios diferentes, el dominio de la jugabilidad (cartas, avatares, biblioteca de aprendizaje, etc.) y el dominio del léxico kichwa-inglés (catálogo de términos y traducciones). Los motivos por los que se decidió hacer la separación fueron, por la facilidad para escalar de manera independiente y para separar responsabilidades.

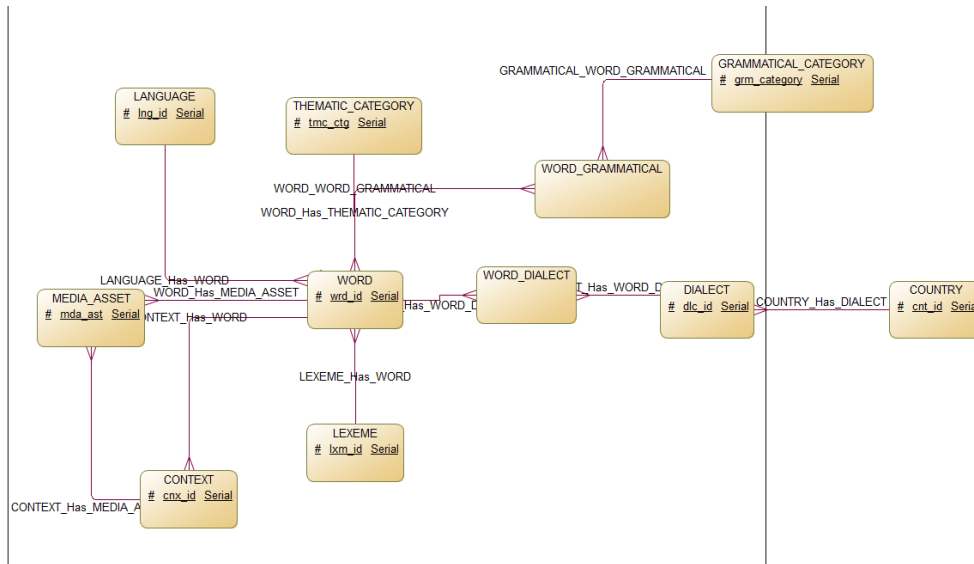
Se refiere a la base de datos de jugabilidad como BD-Game y a la base de datos del léxico como BD-Lex.

3.5.2.1 BD-Lex

3.5.2.1.1 Modelo Conceptual

Figura 4

Modelo Conceptual BD-Lex

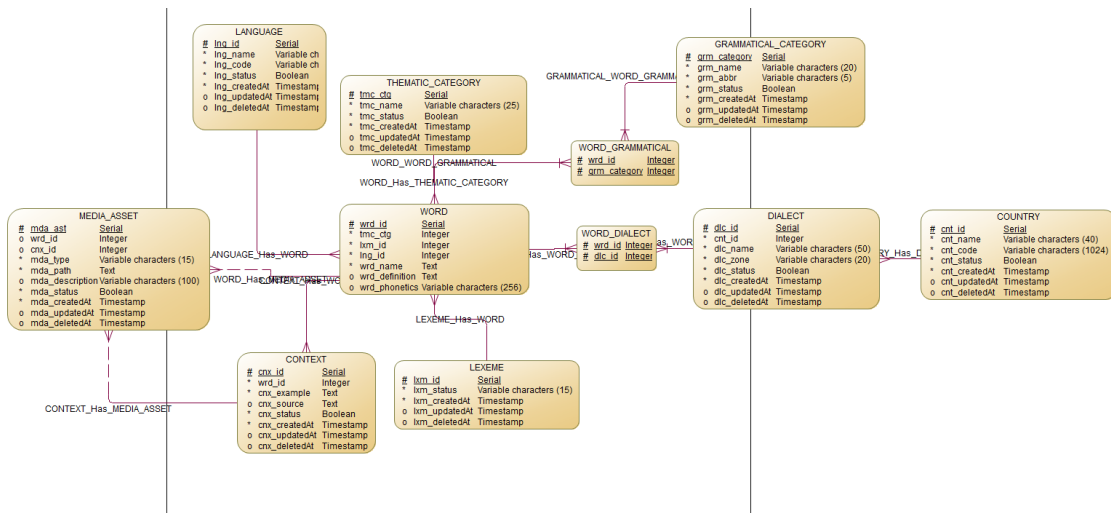


Nota. Autores: Andy Contento y Oscar Figueroa (2025).

3.5.2.1.2 Modelo Lógico

Figura 5

Modelo lógico BD-Lex

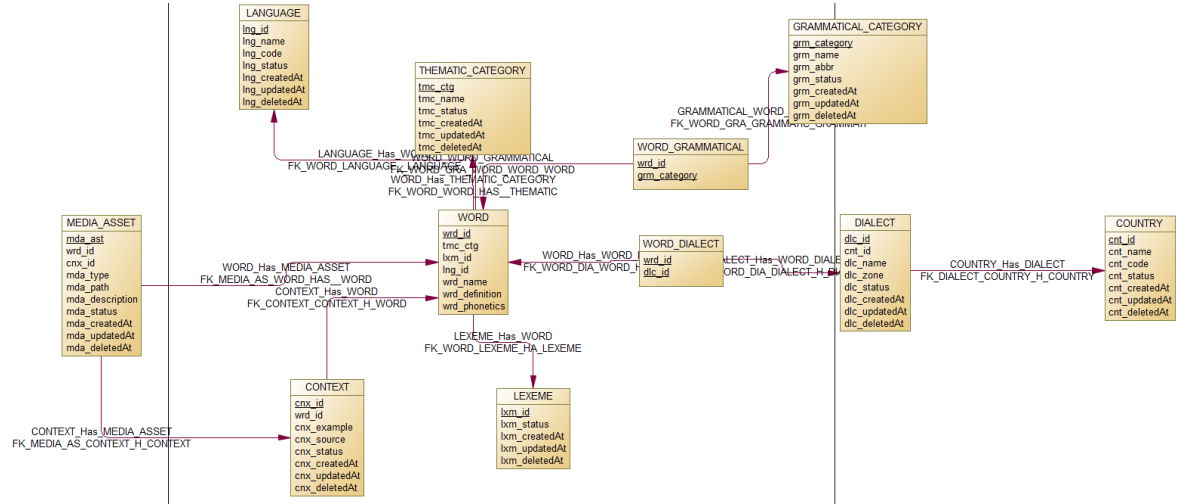


Nota. Autores: Andy Contento y Oscar Figueroa (2025).

3.5.2.3 Modelo Físico

Figura 6

Modelo fisico BD-Lex



Nota. Autores: Andy Contento y Oscar Figueroa (2025).

3.5.2.2 BD-Game

3.5.2.2.1 Modelo Conceptual

Figura 7

Modelo conceptual BD-Game (parte 1)

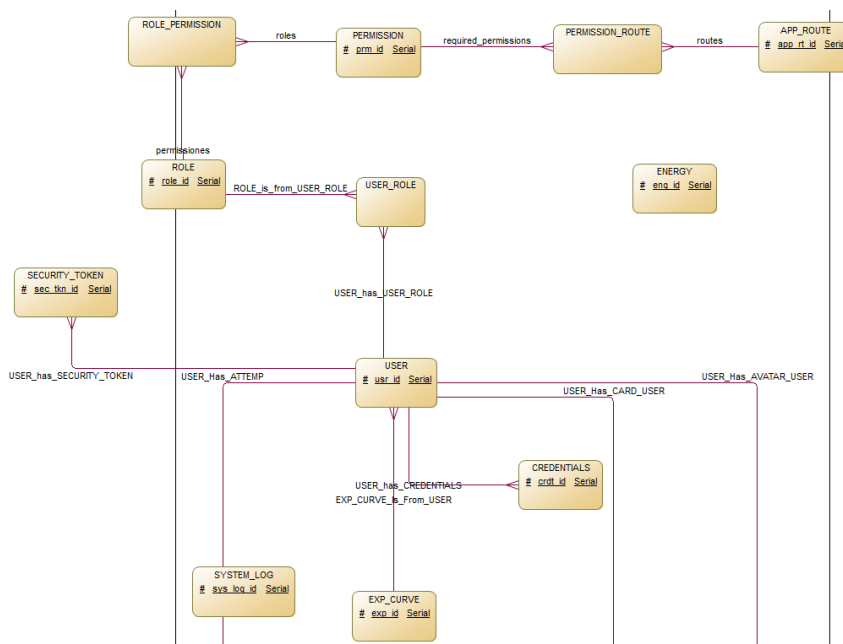
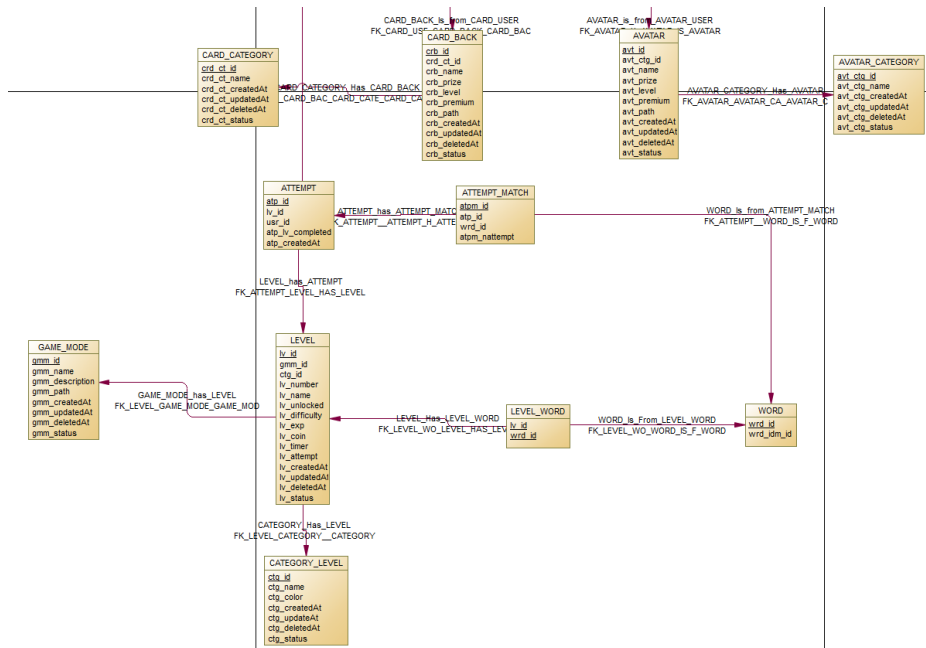


Figura 12

Modelo físico BD-Game (parte 2)



Nota. Autores: Andy Contento y Oscar Figueroa (2025).

3.5.3 Diseño de la interfaz de usuario del prototipo

En esta sección se realizó una descripción de cómo están organizadas las interfaces del prototipo y la elección de la paleta de colores. Se elaboró un acta de reunión bajo la metodología SCRUM (véase el Anexo E y Anexo F) con el propósito de validar la conformidad. En el acta se mostró la interfaz, la cual fue revisada con el ingeniero Jorge Alarcón. Posterior a la revisión y discusión, el ingeniero manifestó su conformidad y firmó el acta como evidencia de aprobación de la interfaz.

3.5.3.1 Estructura Visual de la Interfaz

La estructura visual que tiene el prototipo se realizó siguiente una disposición jerárquica, para permitir al usuario identificar con facilidad elementos interactivos y áreas informativas.

3.5.3.1.1 Distribución general

- **Zona superior:** se encarga de mostrar la información general del usuario, aquí se visualiza el nombre de usuario, avatar, nivel, monedas y energía que posea.
- **Zona central:** es el núcleo visual de la interfaz, el cual despliega la información o actividades del usuario. En las diferentes pantallas, la zona se adapta para mostrar el contenido correspondiente, ya sean cartas del memorama, formularios de registro o configuraciones del usuario.
- **Zona inferior:** contiene el menú de navegación principal con las secciones principales, como perfil, mapa y recompensas. Los íconos son ilustrativos y mantienen un espaciado uniforme pensado precisamente para que sean fácilmente seleccionables. Esta barra se mantiene fija durante la navegación, y así el flujo de uso es consistente.

3.5.3.1.2 Elementos interactivos

- Los botones principales son redondeados y con una iconografía clara.
- Al presionar un botón se activa una ligera animación o cambio de color para indicar la interacción.
- Los mensajes de confirmación o error se aparecen brevemente en la parte inferior.

3.5.3.2 Selección de Paleta de Colores

Según EspacioUX (2025), la selección de colores juega un papel muy importante en el diseño de las interfaces. En el caso del prototipo, se decidió ir por tonos de color azul, ya que representan seguridad, serenidad, inteligencia y calma, y por tonos de color violeta, asociados a la creatividad, autenticidad y verdad.

La paleta de colores elegida se conformó por:

- Azul oscuro.

- Índigo oscuro.
- Morado oscuro.
- Morado vibrante.

Para una mejor visualización, se muestran los colores y sus respectivos códigos en la Figura 13.

Figura 13

Paleta de colores de la interfaz del prototipo



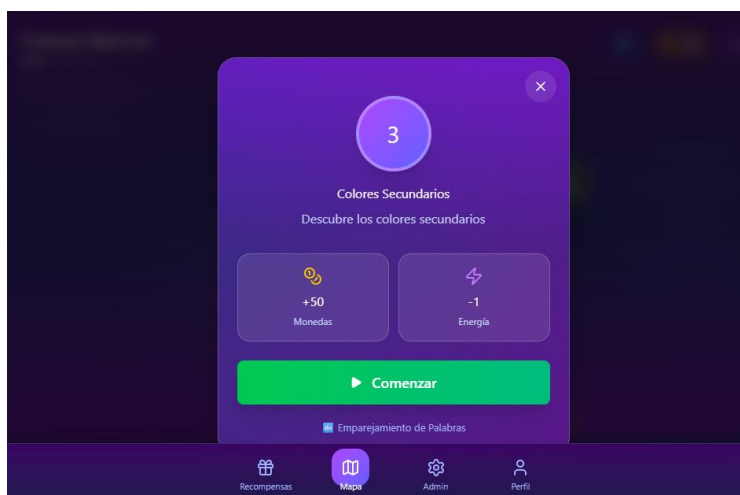
Nota. Autores: Andy Contento y Oscar Figueroa (2025).

3.5.3.3 Pantallas Principales de la Interfaz

En esta sección se muestran algunas de las pantallas principales del prototipo.

Figura 14

Jugar nivel (web)



Nota. Andy Contento y Oscar Figueroa (2025).

Figura 15

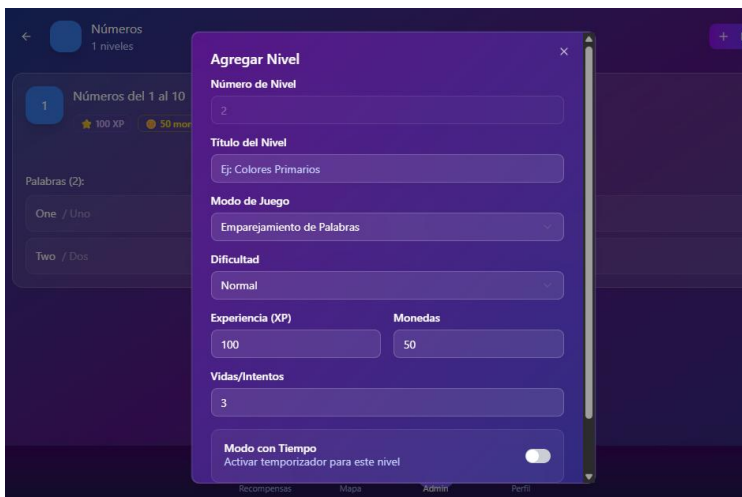
Jugar nivel (móvil)



Nota. Andy Contento y Oscar Figueroa (2025).

Figura 16

Agregar nivel (web)



Nota. Andy Contento y Oscar Figueroa (2025).

Figura 17

Agregar nivel (móvil)



Nota. Andy Contento y Oscar Figueroa (2025).

CAPÍTULO 4: Desarrollo y pruebas del prototipo web y móvil

4.1 Desarrollo del prototipo

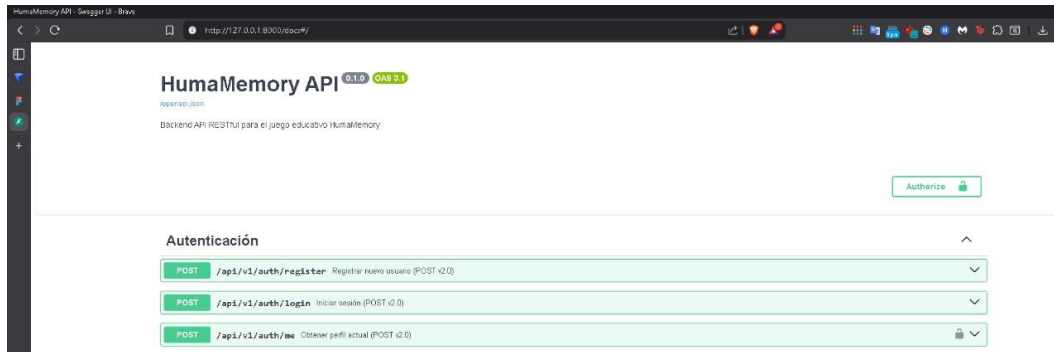
El desarrollo del prototipo se llevó a cabo mediante sprints, siguiendo los principios que establece la metodología ágil de Scrum; cada sprint fue gestionado siguiendo los eventos que indica Scrum. Al inicio de cada ciclo se realizó un Sprint Planning de manera informal basado en el Product Backlog, donde se seleccionaron las historias a desarrollar. Durante cada momento del proceso se mantuvo una comunicación continua entre ambos desarrolladores, lo que es equivalente a un Daily Scrum. Y al finalizar cada sprint, se llevó a cabo un Sprint Review, documentado a través de sus actas correspondientes. Al final de cada sprint se llevó a cabo una retroalimentación de manera interna (Sprint Retrospective) que permitió ajustar el enfoque a llevar para el próximo sprint.

4.1.1 Diseño de la base de datos y desarrollo del backend

Durante el primer sprint, desarrollado entre el 6 y 17 de octubre de 2025, se realizó el diseño de la base de datos y el desarrollo de los servicios de backend. En esta fase se elaboraron los modelos conceptuales, lógicos y físicos de las bases de datos BD-Lex y BD-Game, utilizando la herramienta de Power Designer. En este ciclo también se implementaron los servicios de backend para ambas bases de datos, haciendo uso de FastAPI, estructurando en capas. Al finalizar con el sprint, se validó el cumplimiento mediante el Acta de Sprint 1 (véase Anexo G y Anexo H).

Figura 18

API del prototipo

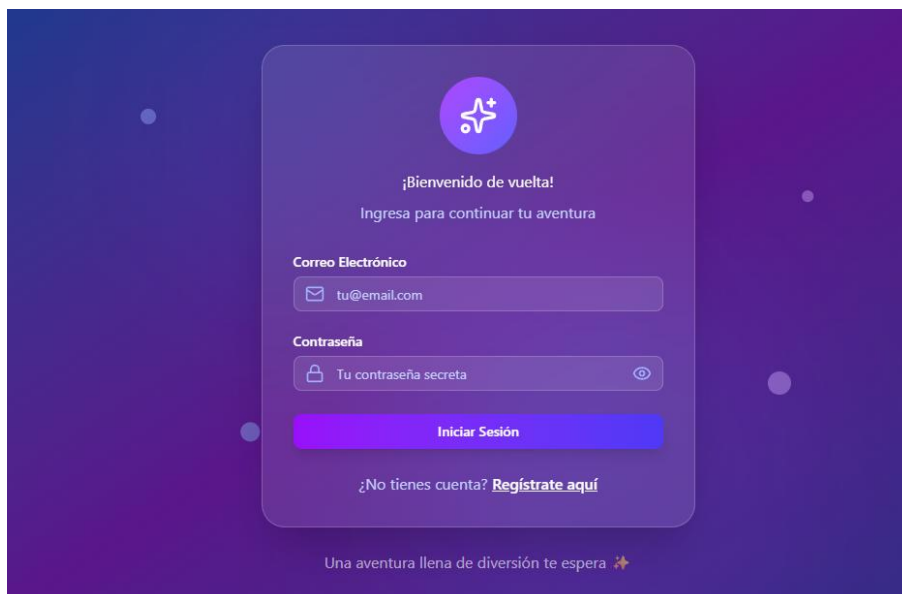


4.1.2 Diseño de la interfaz y módulo de autenticación

Durante el segundo sprint, desarrollado entre el 20 y 31 de octubre de 2025, se realizó el diseño de la interfaz del prototipo y la implementación del módulo de autenticación junto con su control de acceso. Las interfaces fueron desarrolladas en Figma, aplicando la paleta de colores seleccionados y la jerarquía mencionada. En este ciclo también se avanzó con el desarrollo del frontend con Flutter, implementando el registro e inicio de sesión. Al finalizar el sprint, se validó el cumplimiento mediante el Acta de Sprint 2 (véase Anexo I y Anexo J).

Figura 19

Registro e inicio de sesión del prototipo

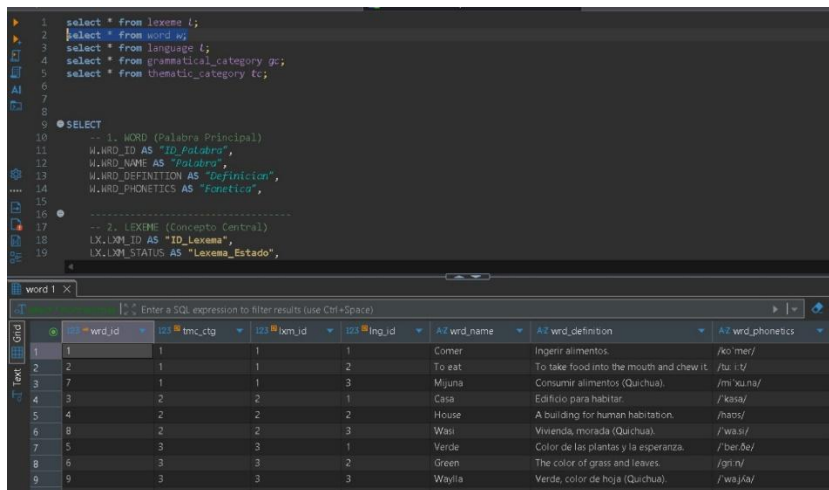


4.1.3 Población de BD-Lex y Desarrollo del Módulo de Juegos Interactivos

Durante el tercer sprint, desarrollado entre el 3 y 21 de noviembre de 2025, se desarrolló la carga inicial de datos en la base de datos BD-Lex, introduciendo vocabulario en kichwa e inglés. En este ciclo también se desarrolló el *Módulo de Juegos Interactivos*, incluyendo niveles en el modo memorama y en el de emparejamiento de palabras. Al finalizar el sprint se validó el cumplimiento mediante el Acta de Sprint 3 (véase Anexo K y Anexo L).

Figura 20

Población BD-Lex

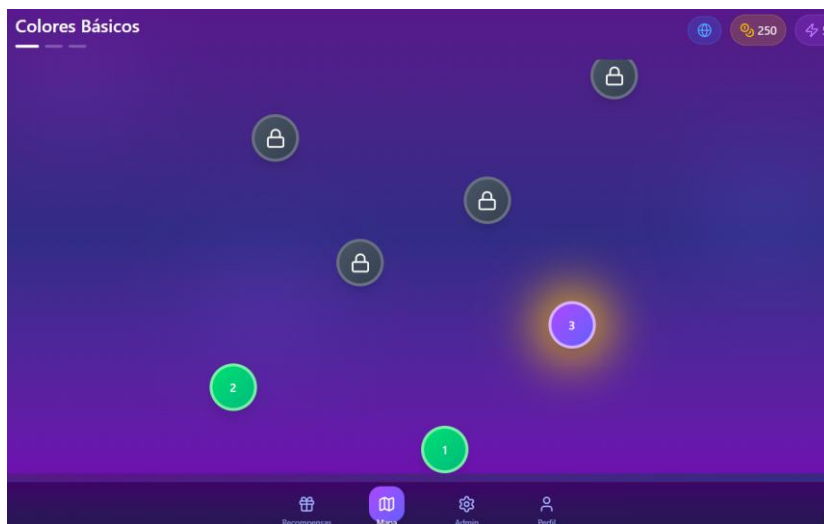


```
1 select * from lexeme l;
2 select * from word w;
3 select * from language l;
4 select * from grammatical_category gc;
5 select * from thematic_category tc;
6
7
8
9
10
11
12
13
14
15
16
17
18
19
```

WRD_ID	WRD_NAME	WRD_DEFINITION	WRD_PHONETICS
1	Comer	Ingerir alimentos.	/ko mer/
2	To eat	To take food into the mouth and chew it.	/tu i:t/
7	Miyuna	Consumir alimentos (Quichua).	/mi xu na/
3	Casa	Edificio para habitar.	/kasa/
4	House	A building for human habitation.	/haʊs/
8	Wasi	Vivienda, morada (Quichua).	/wasɪ/
5	Verde	Color de las plantas y la esperanza.	/ber de/
6	Green	The color of grass and leaves.	/grɪ n/
3	Waylla	Verde, color de hoja (Quichua).	/wa jʌ/

Figura 21

Módulo de juegos interactivos



4.1.4 Módulo de configuración para el Usuario, Clasificación Global, Logros y Biblioteca de Aprendizaje

Durante el cuarto sprint, desarrollado entre el 24 de noviembre y 4 de diciembre de 2025, se desarrollaron los módulos necesarios para consolidar el funcionamiento del prototipo para el jugador.

En este periodo se desarrolló el *Módulo de Configuración para el Usuario*, para permitir la edición del nombre, selección de avatar y elección del reverso de carta de preferencia. Se implementó el *Módulo de Clasificación Global*, que muestra el ranking que los jugadores tienen en base a su experiencia acumulada en el prototipo. De igual manera, se construyó el *Módulo de Logros y Recompensas*, que permite consultar los logros que se han desbloqueado, recompensas obtenidas y gestión de obtención de ítems mediante monedas del juego. Finalmente, se desarrolló el *Módulo de Biblioteca de Aprendizaje*, sección en la que se registran todas las cartas y los errores que el usuario a cometido al jugar niveles.

Al finalizar el sprint se validó el cumplimiento mediante el Acta de Sprint 4 (véase Anexo M y Anexo N).

Figura 22

Módulo de Configuración para el usuario

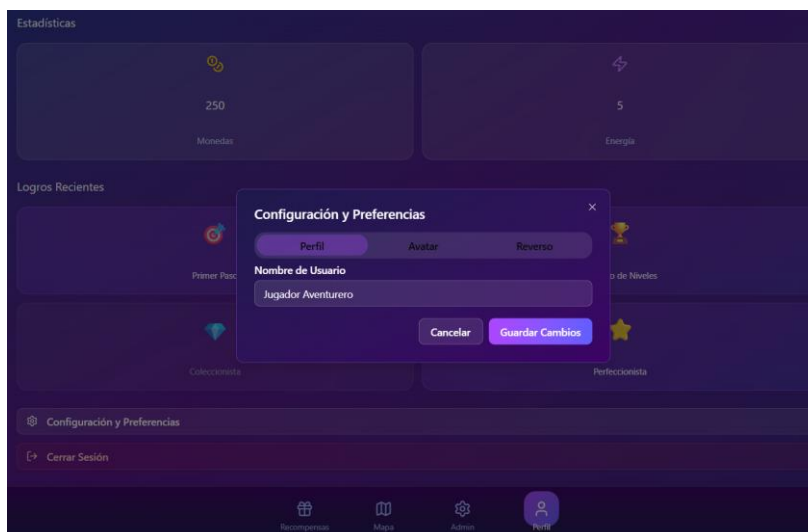


Figura 23

Módulo de clasificación global

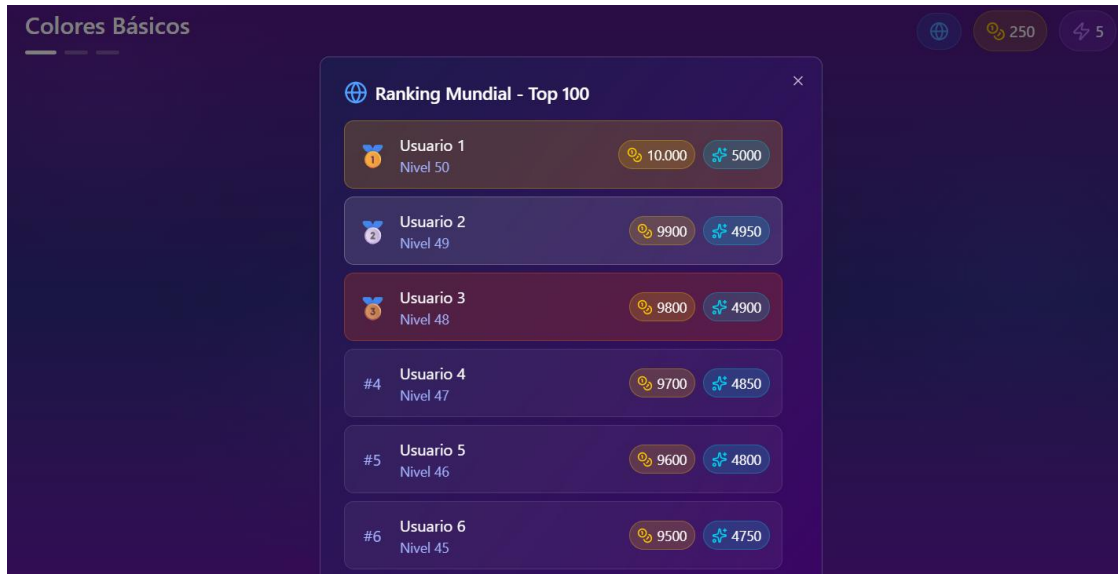


Figura 24

Módulo de logros y recompensas

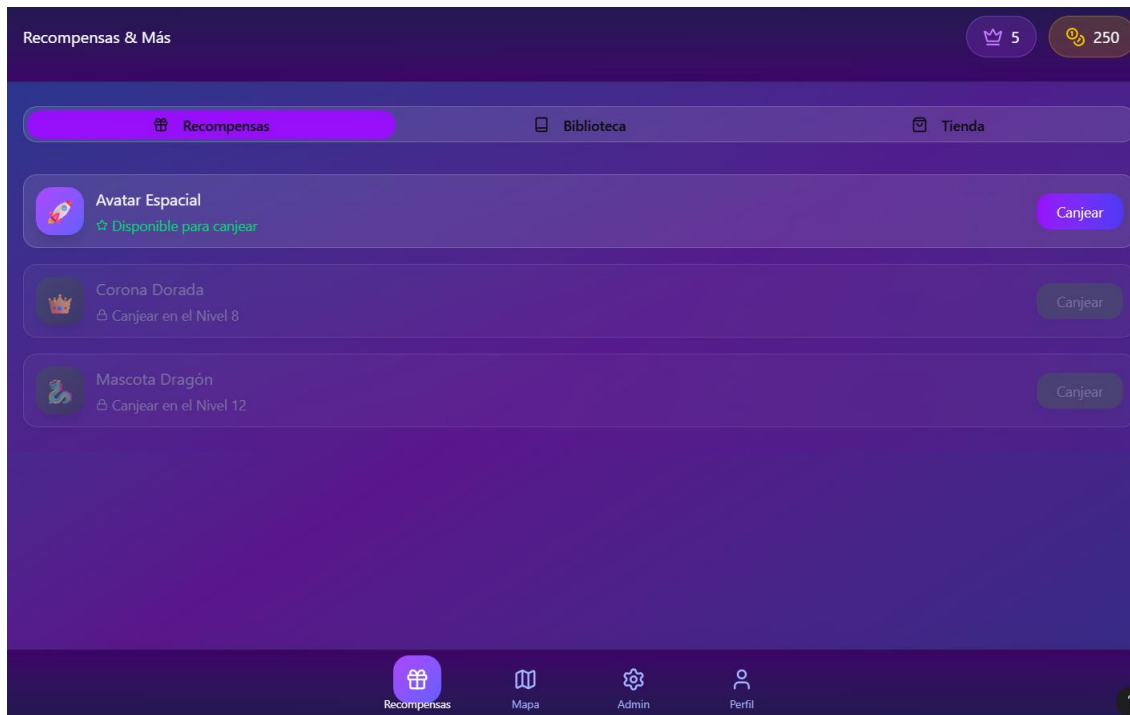
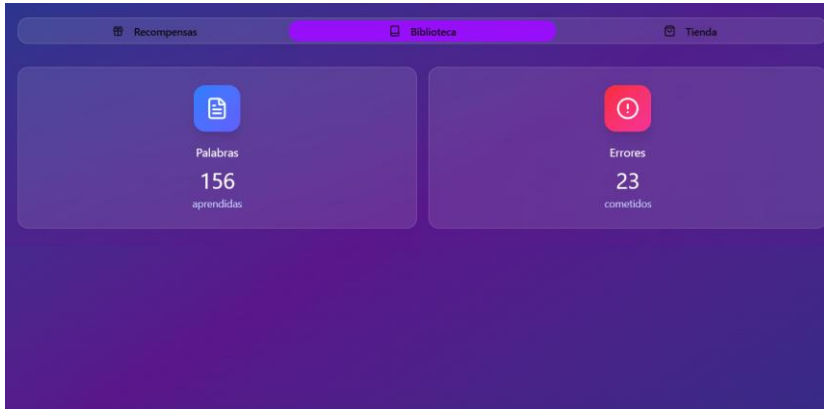


Figura 25

Módulo de biblioteca de aprendizaje

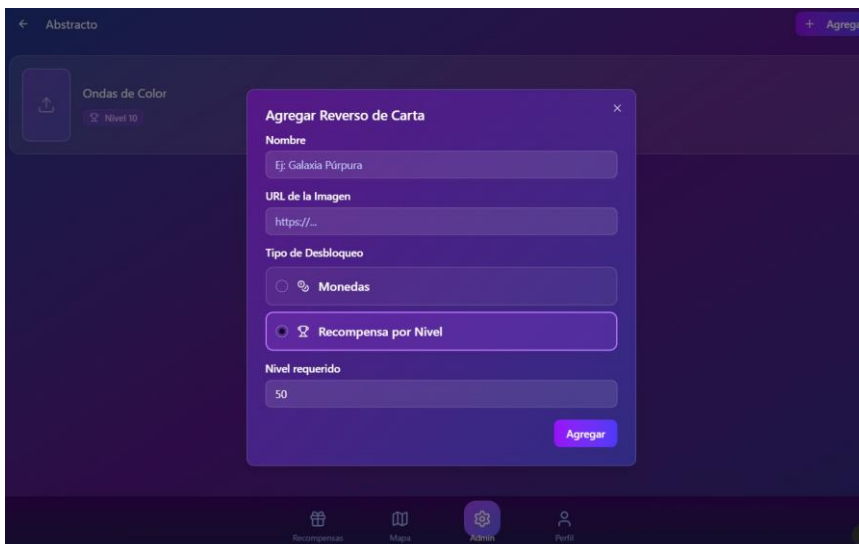


4.1.5 Módulo de Configuración Dinámica Administrativa

Durante el quinto sprint, desarrollado entre el 8 y el 12 de diciembre de 2025, se desarrolló el *Módulo de Configuración Dinámica Administrativa*, siendo el que permite gestionar los contenidos del prototipo. Este módulo permite que el administrador crear niveles, añadir nuevos reversos de cartas y registrar nuevos avatares; lo que permitió completar la parte administrativa del prototipo. Al finalizar el sprint se validó el cumplimiento mediante el Acta de Sprint 5 (véase Anexo O y Anexo P).

Figura 26

Módulo de configuración dinámica administrativa



4.2 Pruebas del prototipo web y móvil

Durante el quinto sprint, también se realizaron pruebas unitarias y de integración a los componentes que se desarrollaron. Estas pruebas permitieron identificar y corregir aspectos menores, garantizando una versión estable y final para el prototipo. Al finalizar el sprint se validó el cumplimiento mediante el Acta de Sprint 5 (véase Anexo O y Anexo P).

4.2.1 Pruebas Unitarias

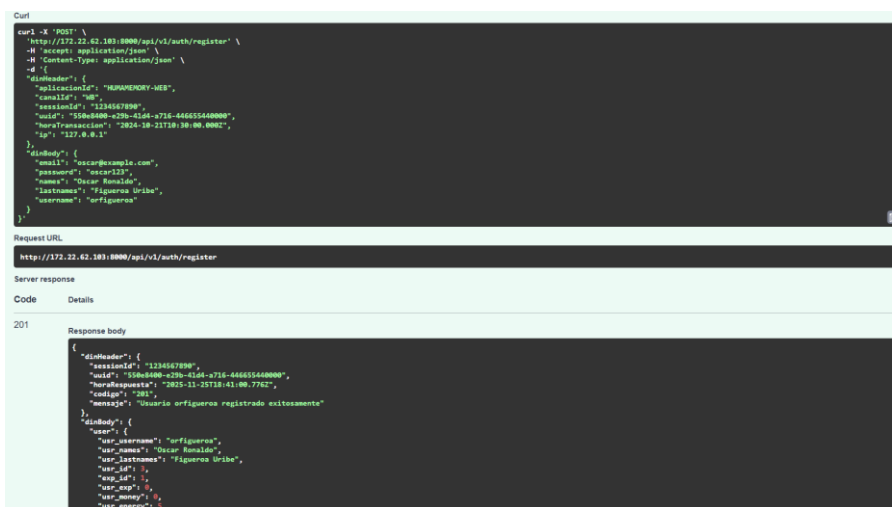
Para validar el funcionamiento del backend del prototipo, se realizaron pruebas unitarias que se enfocaron en el módulo de autenticación. Estas pruebas permitieron comprobar que la lógica de registro e inicio de sesión estaba funcionando adecuadamente. Los casos de prueba permitieron confirmar que cada servicio se comportaba de acuerdo con lo esperado.

4.2.1.1 Registro Exitoso

Se envió una solicitud POST en el endpoint /auth/register con un correo y una contraseña válidos. El backend creó correctamente el usuario y respondió con un mensaje, confirmando que se creó.

Figura 27

Registro Exitoso



```
curl -X 'POST' \
  'http://172.22.42.183:8000/api/v1/auth/register' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "deviceId": "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX",
    "sessionId": "1234567890",
    "tokenId": "528d808-226-4144-371E-446555448888",
    "horaTransaccion": "2024-10-21T18:30:00.000Z",
    "ip": "127.0.0.1"
  },
  "idHeader": {
    "email": "oscar@example.com",
    "password": "oscar123",
    "name": "Oscar Romulo",
    "lastnames": "Figueras Uribe",
    "username": "orfigueras"
  }
}
```

Request URL

```
http://172.22.42.183:8000/api/v1/auth/register
```

Server response

Code Details

201

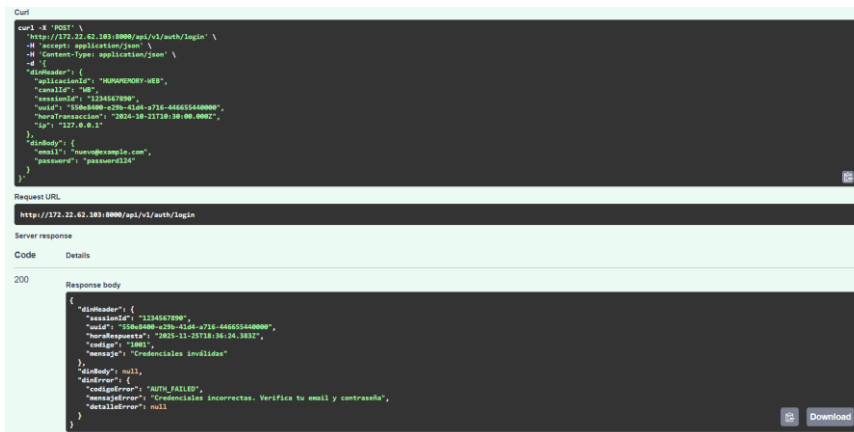
```
Response body
{
  "idHeader": {
    "sessionId": "1234567890",
    "tokenId": "528d808-226-4144-371E-446555448888",
    "horaRespuesta": "2024-11-25T18:41:00.776Z",
    "codigo": "201",
    "mensaje": "Usuario orfigueras registrado exitosamente"
  },
  "idBody": {
    "user": {
      "usr_username": "orfigueras",
      "usr_name": "Oscar Romulo",
      "usr_lastnames": "Figueras Uribe",
      "usr_id": 1,
      "exp_id": 1,
      "usr_age": 0,
      "usr_mage": 0,
      "usr_energy": 1,
      "usr_energy": 1,
    }
  }
}
```


4.2.1.4 Login Incorrecto

Al enviar credenciales no válidas en una solicitud POST al endpoint auth/login, el backend respondió con un mensaje, indicando que las credenciales fueron incorrectas, y confirmando de esta manera un buen manejo de errores.

Figura 30

Login incorrecto



4.2.2 Pruebas de Integración

Las pruebas de integración se realizaron con el objetivo de verificar que las diferentes partes del prototipo trabajen de manera conjunta y mantengan un flujo correcto de información entre la interfaz de usuario, los servicios desarrollados en el backend y la base de datos. En esta etapa lo que se verificó es el proceso de la autenticación, evaluando que el registro e inicio de sesión funcionen adecuadamente. Estas pruebas permitieron demostrar que los componentes desarrollados por separado se integran correctamente dentro del prototipo funcional.

4.2.2.1 Registro Completo de Usuario

Esta prueba de integración verificó que el proceso de registro de un nuevo usuario funcionó de extremo a extremo, desde el ingreso de los datos en la interfaz hasta su almacenamiento en la base de datos y su posterior navegación a la pantalla de inicio del prototipo.

4.2.2.1.1 Objetivo

Comprobar que, al registrar un usuario desde el frontend, el backend cree la cuenta en la base de datos y el prototipo redirija al usuario a la pantalla correspondiente.

4.2.2.1.2 Procedimiento

1. Acceder a la pantalla de registro del prototipo.
2. Ingresar un correo electrónico válido y una contraseña.
3. Presionar el botón de registro.
4. Esperar la respuesta del backend y la redirección automática de la aplicación.

4.2.2.1.3 Resultado

- El backend respondió con un código de éxito indicando que el usuario ha sido creado.
- El nuevo usuario quedó registrado en la base de datos.
- La aplicación mostró un mensaje de confirmación de registro y redireccionó a la pantalla de inicio de sesión.

Figura 31

Registro desde la interfaz (parte 1)

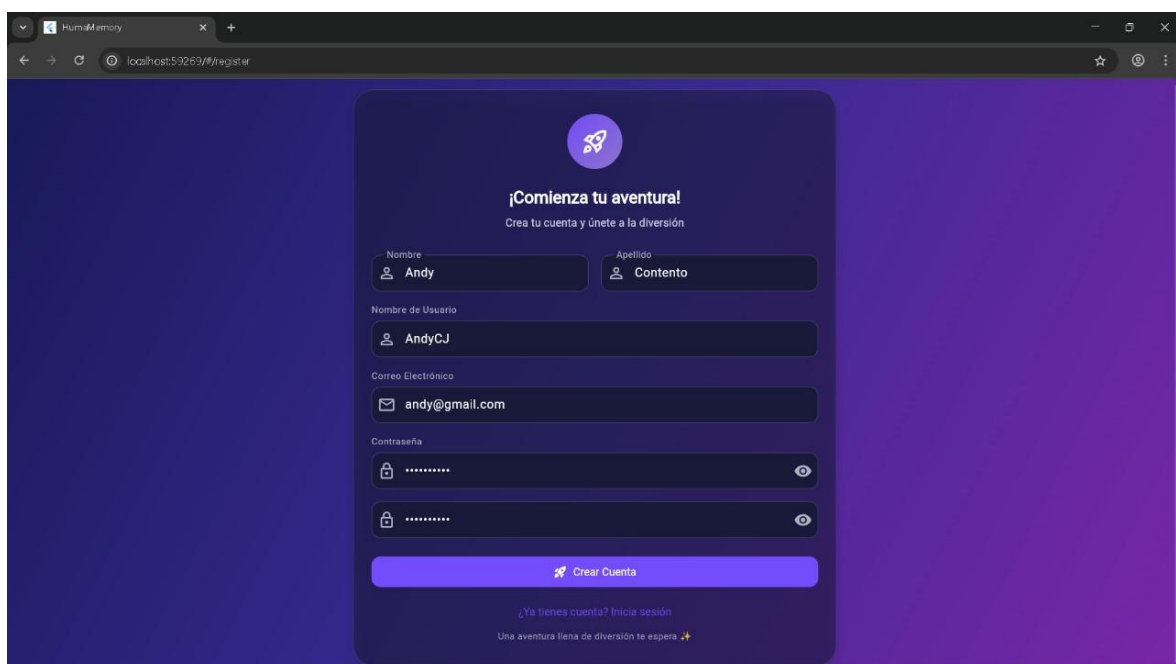


Figura 32

Registro desde la interfaz (parte 2)

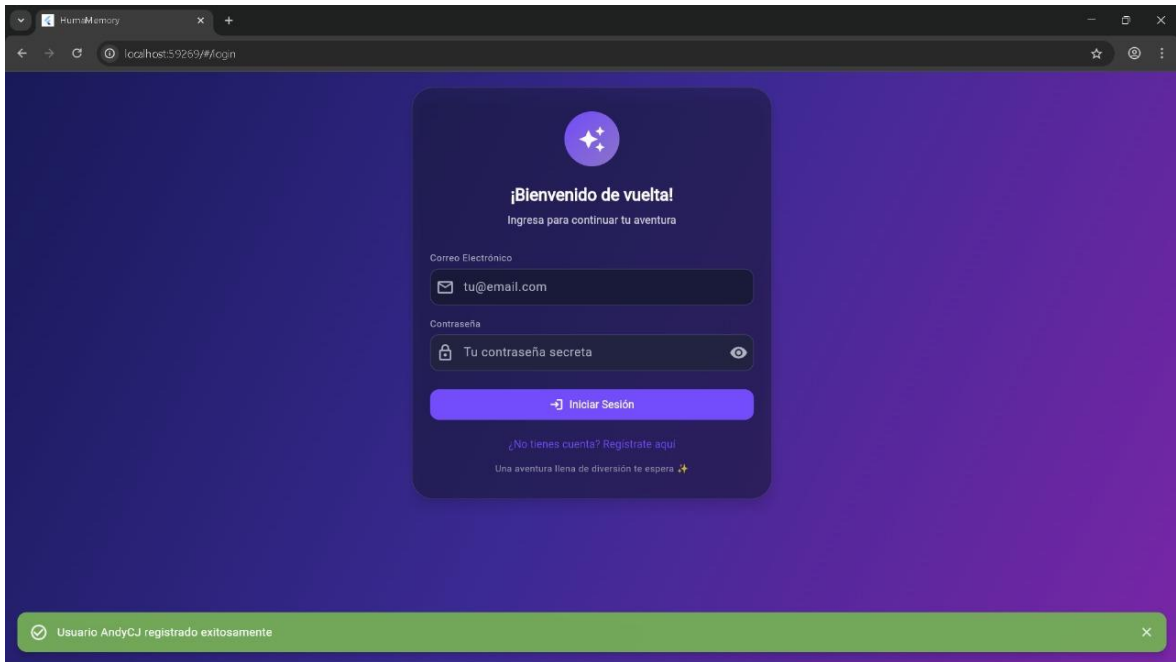


Figura 33

Registro de usuario en la base de datos

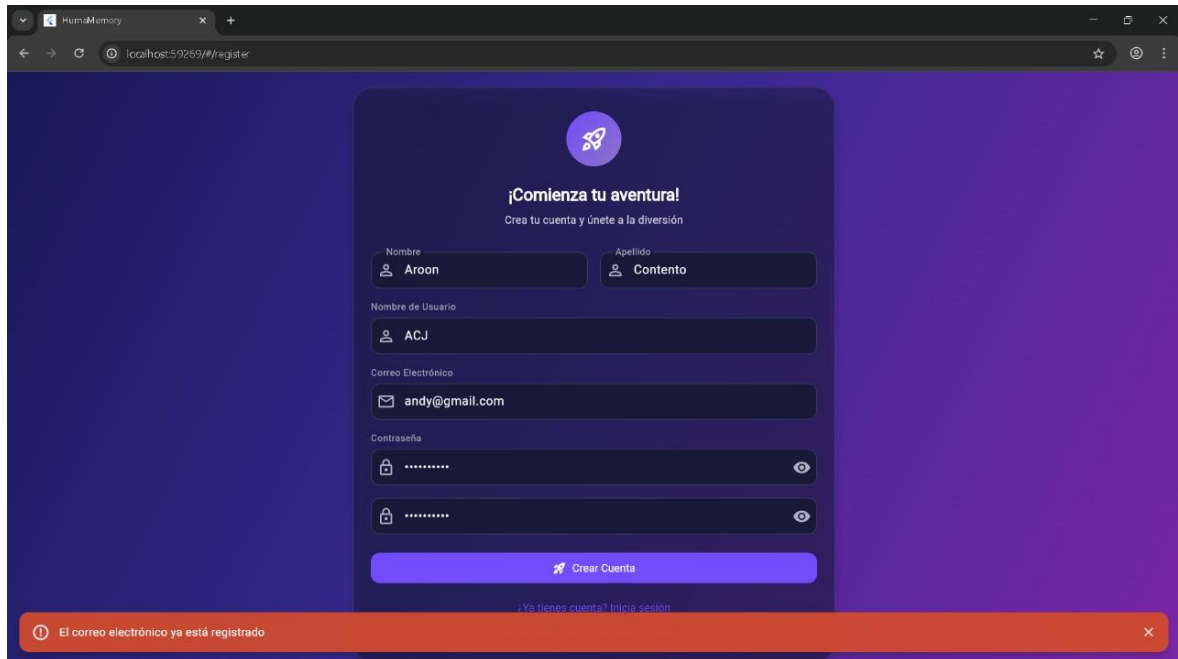
	usr_id	exp_id	usr_names	usr_lastnamees	usr_username	usr_exp	usr_money	usr_energy
1	3	1	Oscar Ronaldo	Figueroa Uribe	ronaldo_updated	0	0	2
2	2	1	Juan Carlos	Pérez García	juanperez_updated	0	200	4
3	12	1	Andy	Contento	AndyCJ	0	0	5

4.2.2.1.4 Caso de Error

Adicionalmente, se realizó una prueba de integración negativa, un intento de registro con un usuario ya existente, verificando que el backend devuelve el error correspondiente y la interfaz muestra el mensaje de retroalimentación adecuado.

Figura 34

Registro de usuario con correo ya existente



4.2.2.2 Inicio de Sesión

Esta prueba de integración validó el flujo de inicio de sesión, desde el ingreso de las credenciales en la interfaz hasta la validación en el backend y la correspondiente redirección de pantalla del prototipo.

4.2.2.2.1 Objetivo

Verificar que el usuario previamente registrado pueda autenticarse correctamente, que el backend valide las credenciales en la base de datos y que el frontend maneje la navegación según el resultado del inicio de sesión.

4.2.2.2.2 Procedimiento

1. Acceder a la pantalla de inicio de sesión.
2. Ingresar el correo electrónico y contraseña de un usuario previamente registrado.
3. Presionar el botón de inicio de sesión.

4. Esperar la respuesta del backend y la navegación hacia la pantalla principal del prototipo.

4.2.2.2.3 Resultado

- El backend respondió con un código de éxito e incluyó los datos básicos y token de autenticación del usuario.
- La aplicación guardó la información que es necesaria para mantener la sesión activa.
- Se mostró la pantalla principal con el mapa de los niveles disponibles.
- No se presentaron errores de comunicación entre el frontend, backend y base de datos.

Figura 35

Inicio de sesión desde la interfaz (parte 1)

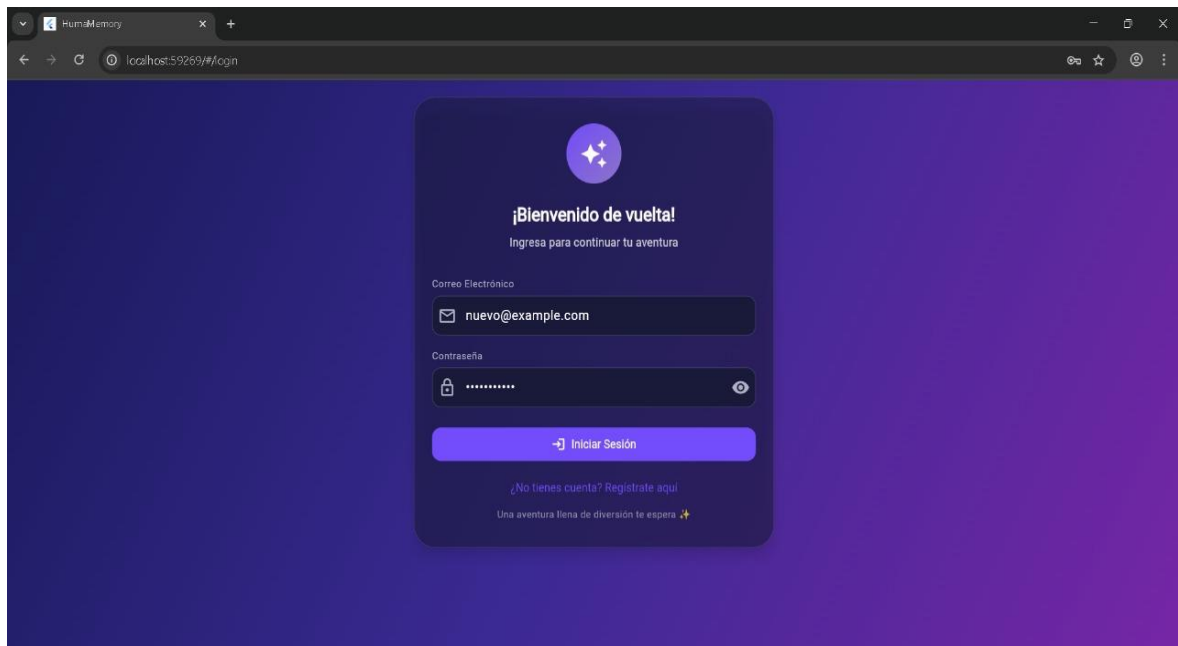
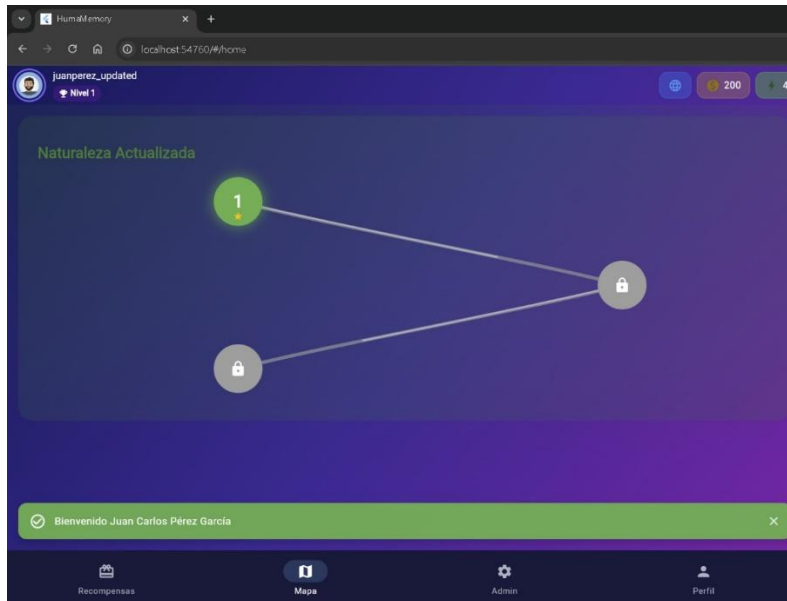


Figura 36

Inicio de sesión desde la interfaz (parte 2)

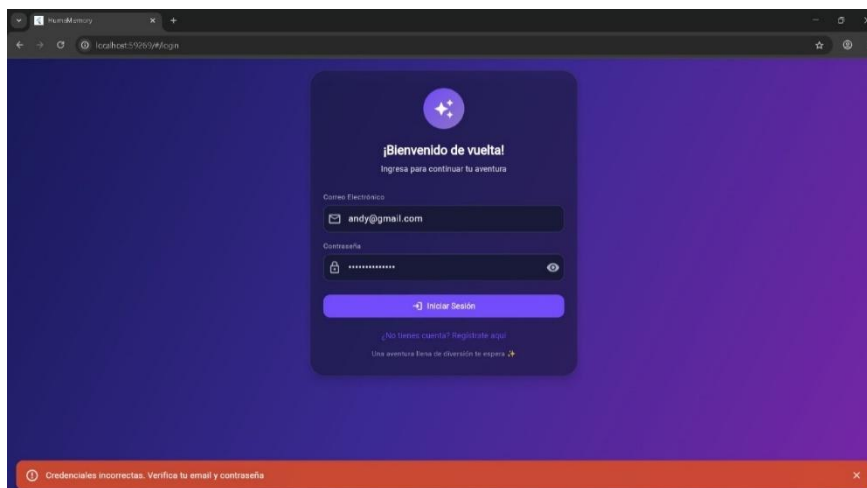


4.2.2.2.4 Caso de Error

Adicionalmente, se realizó una prueba de integración negativa, un intento de inicio de sesión con credenciales incorrectas, verificando que el backend devuelve el error correspondiente y la interfaz muestra el mensaje de retroalimentación adecuado sin permitir que se acceda al sistema.

Figura 37

Inicio de sesión con credenciales incorrectas



4.3 Entrega del Prototipo

Como cierre del proceso de desarrollo, se procedió a realizar la entrega formal del prototipo web y móvil al docente tutor, conforme a lo que se planificó a lo largo del proceso. La entrega incluyó todos los módulos implementados y un manual de usuario (véase *Anexo Q* y *Anexo R*), que explica el uso y navegación del prototipo. La entrega quedó documentada en el acta de entrega y aceptación del prototipo (véase *Anexo S*), donde se deja constancia de la revisión y conformidad por parte del ingeniero Jorge Alarcón con respecto al prototipo desarrollado.

CAPÍTULO 5: Conclusiones y Recomendaciones

5.1 Conclusiones

- La implementación de Flutter para el frontend, FastAPI para el backend y la base de datos PostgreSQL, permitió poder construir un prototipo estable y escalable. El diseño modular y las pruebas realizadas confirmaron que estas herramientas fueron adecuadas para garantizar mantenibilidad y un correcto funcionamiento.
- El prototipo logró alcanzar los objetivos propuestos al ofrecer un entorno dinámico para el aprendizaje de vocabulario básico en kichwa e inglés. Se limitó a contenidos iniciales como números y animales, pero se comprobó que la estructura del prototipo permite ampliar fácilmente las categorías y niveles, lo que evidencia la viabilidad del proyecto como un recurso educativo.
- Las funcionalidades implementadas, tales como la autenticación, el módulo de juegos interactivos, la biblioteca de aprendizaje, la clasificación global, los logros y la configuración dinámica administrativa funcionaron de manera coherente. Esto evidenció que el diseño, la construcción de la interfaz y la lógica planteada, fueron correctamente implementados en el prototipo funcional.

5.2 Recomendaciones

- Es recomendable definir la estructura visual de la interfaz, estableciendo la ubicación de los elementos informativos, interactivos y de navegación, ya que facilita el desarrollo del frontend. Al contar con una distribución previamente pensada, reduce la necesidad de tomar decisiones improvisadas sobre aspectos como colores o ubicación de elementos, permitiendo enfocar los esfuerzos en la funcionalidad del prototipo.

- Es recomendable utilizar herramientas para el desarrollo multiplataforma como Flutter, que demuestra ser eficiente para el desarrollo simultáneo en web y móvil, lo que asegura que proyectos de desarrollo puedan completarse sin duplicar esfuerzos de programación. De esta manera se reducen tiempos de desarrollo y se asegura que futuras versiones del prototipo tengan un mismo código base.
- Es recomendable incorporar un módulo administrativo para gestionar contenido desde la interfaz, dado que permite actualizar elementos del prototipo sin recurrir a cambios en la programación. Este enfoque provee ventajas en la organización y control del contenido, siendo una buena práctica para prototipos que requieren actualizaciones frecuentes.

Bibliografía

- Adobe. (s.f.). *Formato RTF: ¿qué es un archivo RTF, cómo crearlos y editarlos?* Obtenido de Adobe: <https://www.adobe.com/es/acrobat/resources/document-files/text-files/rtf-file.html>
- Allen, J. (14 de Febrero de 2024). *¿Qué es Pydantic? Validación de datos en Python.* Obtenido de Prefect: <https://www.prefect.io/blog/what-is-pydantic-validating-data-in-python>
- Amazon Web Services. (s.f.). *¿Qué es Flutter?* Obtenido de AWS: <https://aws.amazon.com/es/what-is/flutter/>
- Amazon Web Services. (s.f.). *¿Qué es una clave de API?* Obtenido de AWS: <https://aws.amazon.com/es/what-is/api-key/>
- Aula CM. (s.f.). *UI (User Interface) o Interfaz de Usuario: Qué es, Significado y Ejemplos.* Obtenido de Aula CM: <https://aulacm.com/que-es/ui-user-interface-interfaz-de-usuario-significado-ejemplos/>
- Ben, A. (13 de Marzo de 2024). *<https://scrum-master.org/en/extreme-programming-xp-a-beginners-guide-to-the-agile-method/>.* Obtenido de Scrum-Master: <https://scrum-master.org/en/extreme-programming-xp-a-beginners-guide-to-the-agile-method/>
- Buendía, G., Tasayco, A., & Menacho, A. (2025). Gamificación y tecnología en la educación. *InveCom*, 5(3), 1-20. Obtenido de <https://doi.org/10.5281/zenodo.14549138>
- Cárdenas-Valdivia, J., Floreas-Alvines, I.-V. O., & Cabanillas-Carbonell, M. (2023). Augmented Reality for Quechua Language. *International Journal of Interactive Mobile Technologies*, 4-21. doi:<https://doi.org/10.3991/ijim.v17i06.37793>

Cimas, G. (22 de Julio de 2022). *Visual Studio Code: Editor de código para desarrolladores*.
Obtenido de OpenWebinars: <https://openwebinars.net/blog/que-es-visual-studio-code-y-que-ventajas-ofrece/>

Concepto.de. (2025). *Lenguaje de Programación*. Obtenido de Concepto.de:
<https://concepto.de/lenguaje-de-programacion/>

DataScientest. (2 de Septiembre de 2022). *IDE : ¿Qué es un Entorno de Desarrollo Integrado?* Obtenido de DataScientest: <https://datascientest.com/es/ide-que-es>

DBeaver. (s.f.). *Acerca de*. Obtenido de DBeaver: <https://dbeaver.io/about/>

DBeaver. (s.f.). *Main features*. Obtenido de DBeaver: <https://dbeaver.com/features/>

draw.io. (2025). *draw.io Documentation*. Obtenido de draw.io:
<https://www.drawio.com/doc/>

Eclipse Foundation. (2025). *Eclipse Documentation*. Obtenido de Eclipse Foundation:
<https://www.eclipse.org/documentation/>

EspacioUX. (2025). *Uso del color en interfaces UX/UI (y por qué los necesitamos tanto)*.
Obtenido de EspacioUX: <https://www.espacioux.com/blog/uso-del-color-en-interfaces-ux-ui>

Farías, G. (31 de Octubre de 2024). *Léxico*. Obtenido de Concepto.de:
<https://concepto.de/lexico/>

FastAPI. (s.f.). *FastAPI*. Obtenido de FastAPI: <https://fastapi.tiangolo.com/es/>

FastAPI. (s.f.). *Funcionalidades*. Obtenido de FastAPI:
<https://fastapi.tiangolo.com/es/features/>

- Figma. (2025). *Figma: La herramienta de diseño de interfaz colaborativo*. Obtenido de Figma: <https://www.figma.com/es-la/>
- Flutter. (2025). *Flutter documentation*. Obtenido de Flutter: <https://docs.flutter.dev/>
- García, A. (2025). *Aplicación Web Progresiva (PWA) – Definición y características principales*. Obtenido de SensaciónWeb: <https://sensacionweb.com/diccionario/aplicacion-web-progresiva-pwa-definicion-y-caracteristicas-principales/>
- Gaskin, J. (18 de Septiembre de 2024). *El diagrama de casos de uso: una guía completa y práctica*. Obtenido de Venngage: <https://es.venngage.com/blog/diagrama-de-caso-de-uso/>
- Glosarix. (24 de Febrero de 2025). *Interactivo*. Obtenido de Glosarix: <https://glosarix.com/glossary/interactivo/>
- Goodwin, M. (s.f.). *¿Qué es una API (interfaz de programación de aplicaciones)?* Obtenido de IBM: <https://www.ibm.com/es-es/think/topics/api>
- Google Cloud. (2025). *¿Qué es PostgreSQL?* Obtenido de Google Cloud: <https://cloud.google.com/discover/what-is-postgresql?hl=es>
- IBM. (2021). *¿Qué es una base de datos relacional?* Obtenido de IBM: <https://www.ibm.com/mx-es/think/topics/relational-databases>
- IBM. (s.f.). *OAuth 2.0*. Obtenido de IBM Documentation: <https://www.ibm.com/docs/es/cics-ts/6.x?topic=cics-oauth-20>
- IDERA. (2025). *ER/Studio Data Architect Technical Specifications*. Obtenido de IDERA: <https://wiki.idera.com/spaces/TechSpecs/pages/1266066408/ER+Studio+Data+Architect+Technical+Specifications>

Imperva. (s.f.). *Deserialización*. Obtenido de Imperva:
<https://www.imperva.com/learn/application-security/deserialization/>

Initium Software. (20 de Noviembre de 2024). *Front End: Qué es y su importancia en el desarrollo web*. Obtenido de Initium Software:
https://www.initiumsoft.com/blog_initium/front-end-que-es/

InVision. (2025). *Design better. Faster. Together*. Obtenido de InVision: <https://invision.webflow.io/>

Ken, A. (3 de Octubre de 2023). *Backend: ¿Qué es y para qué sirve?* Obtenido de Gluo:
<https://www.gluo.mx/blog/backend-que-es-y-para-que-sirve>

Ken, A. (16 de Agosto de 2023). *Requisitos no funcionales: ¿Por qué son importantes?* Obtenido de Gluo: <https://www.gluo.mx/blog/requisitos-no-funcionales-por-que-son-importantes>

Laoyan, S. (8 de Enero de 2025). *¿Qué es la Metodología Agile y cómo revoluciona la gestión de proyectos?* Obtenido de Asana: <https://asana.com/es/resources/agile-methodology>

Lenovo. (s.f.). *¿Qué es el diseño responsive?* Obtenido de Lenovo:
https://www.lenovo.com/mx/es/glosario/que-es-el-diseno-responsive/?orgRef=https%253A%252F%252Fwww.google.com%252F&srsltid=AfmBOopbl5wCZFavakqrM4Og71N5dEdhPYtD_KBtXuEQcCtDrfzpNdw1

Lucid Software. (2025). *Diagramming powered by intelligence*. Obtenido de Lucidchart:
<https://www.lucidchart.com/pages>

Meta Platforms. (2025). *Introduction*. Obtenido de React Native:
<https://reactnative.dev/docs/getting-started>

- Microsoft. (18 de Junio de 2025). *Serialización (Visual Basic)*. Obtenido de Microsoft Learn: <https://learn.microsoft.com/es-es/dotnet/visual-basic/programming-guide/concepts/serialization/>
- Microsoft. (2025). *Visual Studio Code documentation*. Obtenido de Visual Studio Code: <https://code.visualstudio.com/docs>
- Ministerio de Educación del Ecuador. (2009). *Yachakukkunapa Shimiyuk Kamu: Runa Shimi – Mishu Shimi [Diccionario Kichwa–Castellano / Castellano–Kichwa]*. Quito: Ministerio de Educación del Ecuador. Obtenido de https://educacion.gob.ec/wp-content/uploads/downloads/2013/03/RK_diccionario_kichwa_castellano.pdf
- Miro. (s.f.). *Diagrama de secuencia UML*. Obtenido de Miro: <https://miro.com/es/diagrama/que-es-diagrama-secuencia-uml/>
- Novalys. (2024). *Funcionalidades principales de PowerDesigner*. Obtenido de PowerDesigner: <https://www.powerdesigner.biz/ES/powerdesigner/powerdesigner-features.html>
- Oppermann, A., & Whitfield, B. (31 de Julio de 2025). *¿Qué es el mapeo relacional de objetos (ORM)?* Obtenido de Built In: <https://builtin.com/data-science/object-relational-mapping>
- Oracle. (s.f.). *¿Qué es Python?* Obtenido de Oracle: <https://www.oracle.com/latam/developer/what-is-python-for-developers/>
- Oracle. (2025). *MySQL documentation*. Obtenido de MySQL: <https://dev.mysql.com/doc/>
- Oyarzún, G. (14 de Marzo de 2023). *Qué es Lucidchart y por qué facilita la gestión de proyectos*. Obtenido de Blog ComparaSoftware: <https://blog.comparasoftware.com/lucidchart/>

- pgAdmin Development Team. (2025). *pgAdmin 4 9.9 documentation*. Obtenido de pgAdmin: <https://www.pgadmin.org/docs/pgadmin4/9.9/index.html>
- Pilicita-Garrido, A., & Cevallos-Duque, D. (2019). Aplicación con sistema operativo Android para el aprendizaje del idioma Kichwa. *Revista Cátedra*, 54-68. doi:<https://doi.org/10.29166/catedra.v2i3.1715>
- Pomboza-Tamaquiza, P. P., Paucar Pomboza, C. M., & Ulcuango Ulcuango, K. d. (2024). Idioma kichwa y resistencia cultural en los pueblos indígenas de Tungurahua-Ecuador. *Diálogo Andino*, 74, 89–99. Obtenido de <http://dx.doi.org/10.4067/S0719-26812024000200089>
- PostgreSQL Global Development Group. (2025). *Acerca de PostgreSQL*. Obtenido de PostgreSQL: <https://www.postgresql.org/about/>
- Pressman, R., & Maxim, B. (2021). *Ingeniería del software: Un enfoque práctico*. Ciudad de México, México: McGraw-Hill.
- Reactive Programming. (s.f.). *Data Transfer Object (DTO)*. Obtenido de Reactive Programming: <https://reactiveprogramming.io/blog/es/patrones-arquitectonicos/dto>
- Sanic Community. (2025). *Build fast. Run fast*. Obtenido de Sanic: <https://sanic.dev/en/>
- Schwaber, K., & Sutherland, J. (2020). *La guía Scrum: La guía definitiva de Scrum: Las reglas del juego*. Obtenido de <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf>
- Smartmind. (2025). *¿Qué es la Gamificación?* Obtenido de Smartmind: <https://www.smartmind.net/blog/que-es-la-gamificacion/>

Stevens, E. (23 de Agosto de 2024). *¿Qué es Figma? Usos, beneficios y características clave*. Obtenido de AND Academy: <https://www.andacademy.com/resources/blog/graphic-design/what-is-figma/>

Universidad Internacional de La Rioja. (8 de Noviembre de 2024). *10 Reglas Heurísticas de Nielsen y su aplicación en Diseño*. Obtenido de UNIR Ecuador: <https://ecuador.unir.net/actualidad-unir/heurísticas-de-nielsen/>

Valtx. (2021). *Metodologías de desarrollo de software: ¿Qué son y para qué sirven?* Obtenido de Valtx: <https://www.valtx.pe/blog/metodologias-para-el-desarrollo-de-software-que-son-y-para-que-sirven>

Visual Paradigm. (s.f.). *¿Qué es el lenguaje de modelado unificado (UML)?* Obtenido de Visual Paradigm: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/>

Visure Solutions. (s.f.). *¿Qué son los requisitos funcionales? Ejemplos y plantillas*. Obtenido de Visure Solutions: <https://visuresolutions.com/es/alm-guide/functional-requirements/>

Anexos

Anexo A

Acta de reunión: validación de requerimientos del prototipo (parte 1).

Acta de Reunión: Validación de Requerimientos del Prototipo

Proyecto: Prototipo web y móvil para el aprendizaje de Kichwa e inglés mediante juegos educativos.

Metodología: Scrum

Fecha: 17 de octubre de 2025

Hora de inicio: 10h00

Hora de finalización: 10h30

Lugar: Pontificia Universidad del Ecuador (Quito), Facultad de Ingeniería, oficina 21.

1. Objetivo de la Reunión

Validar los requerimientos funcionales y no funcionales definidos para el prototipo Huma Memory, asegurando su alineación con los objetivos del proyecto y la aprobación formal por parte del ingeniero Jorge Alarcón, supervisor del trabajo de titulación.

2. Participantes

Nombre Completo	Rol dentro del proyecto	Firma
Andy Aroon Contento Japón	Desarrollador / Product Owner	
Oscar Ronaldo Figueroa Uribe	Desarrollador / Scrum Master	
Ing. Jorge Alarcón	Tutor	

3. Desarrollo de la reunión

Durante la sesión se revisaron y discutieron los requerimientos documentados en la fase de análisis. Se expuso la justificación de cada requisito funcional y no funcional, los cuales fueron contrastados con los objetivos del proyecto.


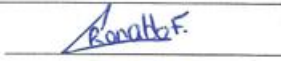

Anexo B

Acta de reunión: validación de requerimientos del prototipo (parte 2).

6. Conclusiones

- Todos los requerimientos funcionales y no funcionales fueron revisados y aprobados por el ingeniero Jorge Alarcón.
- Se validó la coherencia entre los requerimientos y los objetivos del proyecto.
- El acta se firma como evidencia formal de validación dentro del marco metodológico Scrum.

7. Firmas de Conformidad

Participante	Rol	Firma
Andy Aroon Contento Japón	Product Owner	
Óscar Ronaldo Figueroa Uribe	Scrum Master	
Ing. Jorge Alarcón	Tutor / Revisor técnico	

Anexo C

Product backlog - prototipo (parte 1).

Product Backlog – Prototipo

Fecha de creación: 6 de octubre de 2025

Metodología: Scrum

Estructura de cada ítem:

- **HU:** Historia de Usuario
- **Nombre:** título breve
- **Descripción:** necesidad expresada desde la perspectiva del usuario
- **Criterios de aceptación:** condiciones que validan que la historia está completa
- **Prioridad:** Alta / Media / Baja

HU1 – Registro e inicio de sesión

Como usuario del prototipo

Quiero poder registrarme e iniciar sesión con mi correo

Para guardar mi avance y acceder a mis recompensas

Criterios de aceptación:

- Se permite crear cuenta con correo y contraseña.
- Se valida credenciales correctamente.
- El sistema redirige a la pantalla principal al autenticar.

Prioridad: Alta

Anexo D

Product backlog - prototipo (parte 2).

Resumen del Product Backlog

HU	Funcionalidad	Prioridad
HU1	Registro e inicio de sesión	Alta
HU2	Niveles bilingües	Alta
HU3	Pausa y reanudar	Media
HU4	Retroalimentación visual/sonora	Alta
HU5	Ranking global	Media
HU6	Recompensas	Media
HU7	Configuración perfil	Media
HU8	Biblioteca	Media
HU9	Logros	Media
HU10	Configuración administrativa	Alta

Anexo E

Acta de reunión: validación de interfaces y paleta de colores (parte 1).

Acta de Reunión: Validación de Interfaces y Paleta de Colores

Proyecto: Prototipo web y móvil para el aprendizaje de Kichwa e inglés mediante juegos educativos.

Metodología: Scrum

Fecha: 17 de octubre de 2025

Hora de inicio: 10h00

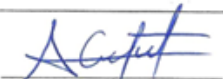
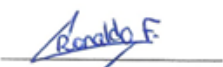

Hora de finalización: 10h30

Lugar: Pontificia Universidad del Ecuador (Quito), Facultad de Ingeniería, oficina 21.

1. Objetivo de la Reunión

Validar el diseño visual del prototipo, incluyendo la estructura de las interfaces creadas en Figma y la paleta de colores seleccionada, garantizando la coherencia estética, la usabilidad y la alineación con los objetivos del proyecto.

2. Participantes

Nombre Completo	Rol dentro del proyecto	Firma
Andy Aroon Contento Japón	Desarrollador / Product Owner	
Oscar Ronaldo Figueroa Uribe	Desarrollador / Scrum Master	
Ing. Jorge Alarcón	Tutor	

3. Desarrollo de la reunión

Durante la sesión, se mostró al ingeniero Jorge Alarcón la totalidad de las interfaces del prototipo diseñadas en Figma, explicando la disposición de los elementos, la jerarquía visual y la navegación entre pantallas. Asimismo, se presentó la paleta de colores oficial definida para el prototipo.

Anexo F




Acta de reunión: validación de interfaces y paleta de colores (parte 2).

El ingeniero revisó las propuestas, sugirió leves ajustes de contraste y tipografía, los cuales fueron incorporados inmediatamente en el entorno Figma. Luego de la retroalimentación final, manifestó su conformidad y aprobación tanto con las interfaces como con la paleta de colores, validando el cumplimiento de los principios de usabilidad y consistencia visual.

4. Conclusiones

- Se aprueba el diseño visual presentado en Figma.
- Se valida la paleta de colores y su aplicación en todas las pantallas.
- El acta se firma como evidencia de aprobación de la interfaz bajo la metodología Scrum.

5. Firmas de Conformidad

Participante	Rol	Firma
Andy Aroon Contento Japón	Product Owner	
Óscar Ronaldo Figueroa Uribe	Scrum Master	
Ing. Jorge Alarcón	Tutor / Revisor técnico	

Anexo G

Acta de Sprint 1 (parte 1).

Acta de Sprint 1: 6 al 17 de octubre de 2025

Proyecto: Prototipo web y móvil para el aprendizaje de Kichwa e inglés mediante juegos educativos.

Metodología: Scrum

Sprint: Sprint 1: Arquitectura de Datos y Backend.

Fecha: 17 de octubre de 2025

Hora de inicio: 10h00

Hora de finalización: 10h30

Lugar: Pontificia Universidad del Ecuador (Quito), Facultad de Ingeniería, oficina 40.

1. Objetivo del Sprint

Diseñar las bases de datos BD-Lex y BD-Game, e implementar los servicios backend necesarios para soportar los requerimientos del prototipo.

2. Actividades Realizadas

- Modelado conceptual, lógico y físico en PowerDesigner.
- Definición de entidades, atributos, claves primarias y foráneas.
- Desarrollo de API REST en FastAPI.
- Configuración de entorno, controladores, servicios y modelos.

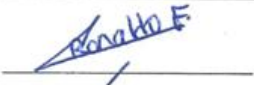

3. Validación

El ingeniero Jorge Alarcón revisó los modelos de datos y la estructura del backend y aprobó su conformidad.

Anexo H

Acta de Sprint 1 (parte 2).

4. Participantes

Nombre Completo	Rol dentro del proyecto	Firma
Andy Aroon Contento Japón	Desarrollador / Product Owner	
Oscar Ronaldo Figueroa Uribe	Desarrollador / Scrum Master	
Ing. Jorge Alarcón	Tutor	

Anexo I

Acta de sprint 2 (parte 1)

Acta de Sprint 2: 20 al 31 de octubre de 2025

Proyecto: Prototipo web y móvil para el aprendizaje de Kichwa e inglés mediante juegos educativos.

Metodología: Scrum

Sprint: Sprint 1: Arquitectura de Datos y Backend.

Fecha: 31 de octubre de 2025

Hora de inicio: 10h00

Hora de finalización: 10h30

Lugar: Pontificia Universidad del Ecuador (Quito), Facultad de Ingeniería, oficina 40.

1. Objetivo del Sprint

Diseñar las interfaces en Figma e implementar el módulo de autenticación y control de acceso en Flutter.

2. Actividades Realizadas

- Diseño UI en Figma.
- Presentación y validación de paleta cromática.
- Programación del registro, inicio de sesión y validación de credenciales.
- Integración con backend de FastAPI.

3. Validación

El ingeniero Jorge Alarcón revisó el diseño en Figma y probó el módulo de autenticación, aprobando su funcionamiento.

Anexo J

Acta de sprint 2 (parte 2)

4. Participantes

Nombre Completo	Rol dentro del proyecto	Firma
Andy Aroon Contento Japón	Desarrollador / Product Owner	
Oscar Ronaldo Figueroa Uribe	Desarrollador / Scrum Master	
Ing. Jorge Alarcón	Tutor	

Anexo K

Acta de sprint 3 (parte 1)

Acta de Sprint 3: 3 al 21 de noviembre de 2025

Proyecto: Prototipo web y móvil para el aprendizaje de Kichwa e inglés mediante juegos educativos.

Metodología: Scrum

Sprint: Sprint 3: Población BD-Lex y módulo de juegos interactivos.

Fecha: 21 de noviembre de 2025

Hora de inicio: 10h00

Hora de finalización: 10h30

Lugar: Pontificia Universidad del Ecuador (Quito), Facultad de Ingeniería, oficina 40.

1. Objetivo del Sprint

Incorporar vocabulario en BD-Lex y desarrollar el módulo de juegos interactivos.

2. Actividades Realizadas

- Carga de datos léxicos (kichwa–inglés).
- Desarrollo del memorama bilingüe.
- Implementación de retroalimentación visual y sonora.
- Sistema de puntaje y finalización de niveles.

3. Validación

El ingeniero Jorge Alarcón validó el funcionamiento del módulo y la consistencia del vocabulario ingresado.

Anexo L

Acta de sprint 3 (parte 2).

4. Participantes

Nombre Completo	Rol dentro del proyecto	Firma
Andy Aroon Contento Japón	Desarrollador / Product Owner	
Oscar Ronaldo Figueroa Uribe	Desarrollador / Scrum Master	
Ing. Jorge Alarcón	Tutor	

Anexo M

Acta de sprint 4 (parte 1)

Acta de Sprint 4: 24 de noviembre al 4 de diciembre de 2025

Proyecto: Prototipo web y móvil para el aprendizaje de Kichwa e inglés mediante juegos educativos.

Metodología: Scrum

Sprint: Sprint 4: Módulos del Sistema.

Fecha: 4 de diciembre de 2025

Hora de inicio: 10h00

Hora de finalización: 10h30

Lugar: Pontificia Universidad del Ecuador (Quito), Facultad de Ingeniería, oficina 21.

1. Objetivo del Sprint

Desarrollar los módulos:

- Módulo de Configuración para el Usuario
- Módulo de Clasificación Global
- Módulo de Logros y Recompensas
- Módulo de Biblioteca de Aprendizaje

2. Actividades Realizadas

- Implementación del Módulo de Configuración: edición de nombre de usuario, avatar y reverso de carta
- Desarrollo del Módulo de Clasificación Global: ranking de jugadores por experiencia
- Construcción del Módulo de Logros y Recompensas
- Desarrollo del Módulo de Biblioteca de Aprendizaje (Palabras vistas y errores)

Anexo N

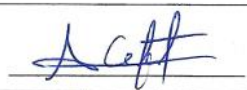
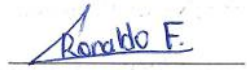

Acta de sprint 4 (parte 2)

3. Validación

El ingeniero Jorge Alarcón revisó y validó:

- Correcto funcionamiento de los módulos implementados
- Flujo de juego completo del prototipo
- Integración con bases de datos y backend

4. Participantes

Nombre Completo	Rol dentro del proyecto	Firma
Andy Aroon Contento Japón	Desarrollador / Product Owner	
Oscar Ronaldo Figueroa Uribe	Desarrollador / Scrum Master	
Ing. Jorge Alarcón	Tutor	

Anexo O

Acta de sprint 5 (parte 1)

Acta de Sprint 5: 8 al 12 de diciembre de 2025

Proyecto: Prototipo web y móvil para el aprendizaje de Kichwa e inglés mediante juegos educativos.

Metodología: Scrum

Sprint: Sprint 5: Configuración Administrativa y Pruebas

Fecha: 8 de diciembre de 2025

Hora de inicio: 10h00

Hora de finalización: 10h30

Lugar: Pontificia Universidad del Ecuador (Quito), Facultad de Ingeniería, oficina 21.

1. Objetivo del Sprint

Desarrollar el Módulo de Configuración Dinámica Administrativa y ejecutar las pruebas unitarias y de integración del prototipo.

2. Actividades Realizadas

- Desarrollo de pantallas administrativas para creación de niveles
- Implementación del registro dinámico de avatares
- Implementación del registro dinámico de reversos de cartas
- Pruebas unitarias.
- Pruebas de integración.
- Corrección de fallos menores detectados durante las pruebas.

Anexo P


Acta de sprint 5 (parte 2)

3. Validación

El ingeniero Jorge Alarcón revisó y validó:

- La funcionalidad completa del módulo administrativo.
- La correcta restricción de acceso según roles.
- La estabilidad del prototipo posterior a las pruebas.

4. Participantes

Nombre Completo	Rol dentro del proyecto	Firma
Andy Aroon Contento Japón	Desarrollador / Product Owner	
Oscar Ronaldo Figueroa Uribe	Desarrollador / Scrum Master	
Ing. Jorge Alarcón	Tutor	

Anexo Q

Manual de usuario (parte 1)

Manual de Usuario del Prototipo Web y Móvil para el Aprendizaje de Kichwa e inglés Mediante Juegos Educativos.

1. Introducción

Prototipo educativo diseñado como un juego interactivo que facilita el aprendizaje bilingüe de kichwa e inglés. La aplicación utiliza el formato tradicional del memorama y la asociación de palabras, lo que combina con recursos niveles de dificultad progresiva y logros, con el fin de ofrecer una experiencia de aprendizaje entretenida, intuitiva y significativa para el usuario.

El prototipo funciona en dos plataformas:

- Aplicación web, accesible desde navegadores.
- Aplicación móvil, desarrollada para dispositivos Android.

El prototipo integra varios módulos principales que organizan su funcionamiento: el módulo de autenticación, el módulo de juegos interactivos, el sistema de puntuación y recompensas, la biblioteca de aprendizaje, la clasificación global, módulo de configuración para el usuario y módulo de configuración dinámica administrativa.

2. Requisitos del Sistema

Aquí se detallan las condiciones mínimas para usar la app.

2.1 Requisitos Técnicos

- Dispositivo Android (versión 5.0 en adelante).
- Conexión a internet para autenticación y sincronización.

Anexo R

Manual de usuario (parte 2)

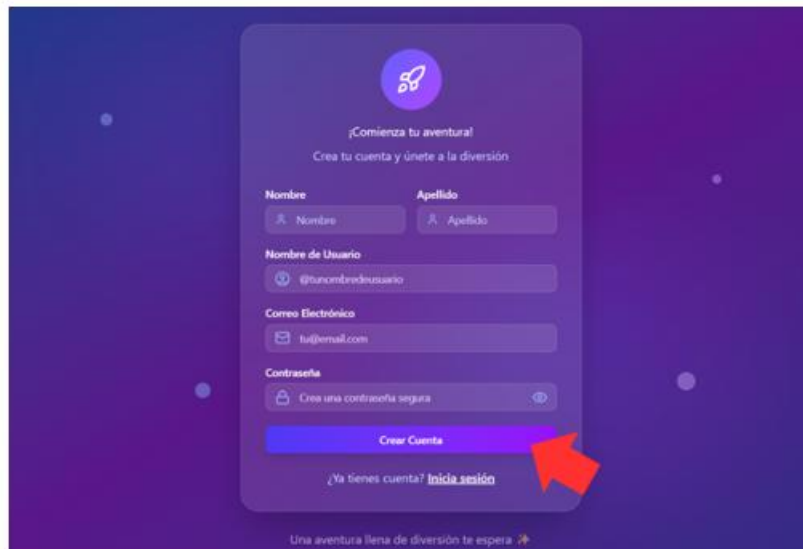
2.2 Requisitos de usuario

- Tener una cuenta válida (o crearla).

3. Acceso al Sistema

3.1 Registro

1. Ingrese los datos que se solicitan.
2. Haga clic en el botón "Crear Cuenta".



Nota. Se presenta una parte del manual de usuario realizado para el prototipo.

Anexo S

Acta de entrega y aceptación del prototipo

Acta de Entrega y Aceptación del Prototipo

Proyecto: Prototipo web y móvil para el aprendizaje de Kichwa e inglés mediante juegos educativos.


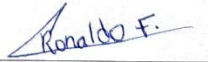
En la ciudad de Quito, el día 8 del mes de enero del 2026, se deja constancia de la entrega formal del desarrollo del prototipo, llevado a cabo en el marco del Plan de Integración Curricular de la Pontificia Universidad Católica del Ecuador, Facultad de Hábitat, Infraestructura y Creatividad, Carrera de Sistemas de la Información.

Los estudiantes Oscar Ronaldo Figueroa Uribe, con cédula de ciudadanía 3040032280 y Andy Aroon Contento Japón, con cédula de ciudadanía 1150039467, proceden a realizar la entrega oficial del prototipo al docente tutor, Ing. Jorge Alarcón, docente de la Carrera de Sistemas de la Información.

Se hace constar también que, junto con el prototipo, se entregó y explicó el Manual de Usuario, documento que detalla el uso y navegación del prototipo.

El docente tutor revisa, recibe y acepta el prototipo entregado, verificando que cumple con los lineamientos académicos establecidos y con las funcionalidades acordadas.

Firmas de conformidad:

Nombre	Firma
Andy Aroon Contento Japón	
Oscar Ronaldo Figueroa Uribe	
Ing. Jorge Alarcón	