

Desarrollo de un videojuego educativo 3D tipo Zig-Zag con algoritmos de navegación autónoma y persecución basados en Inteligencia Artificial para el desarrollo del pensamiento y la estimulación de funciones ejecutivas en niños de 7 a 12 años.

Saltos Palacios Jean Michael y Velasquez Males Tupac Yupanqui

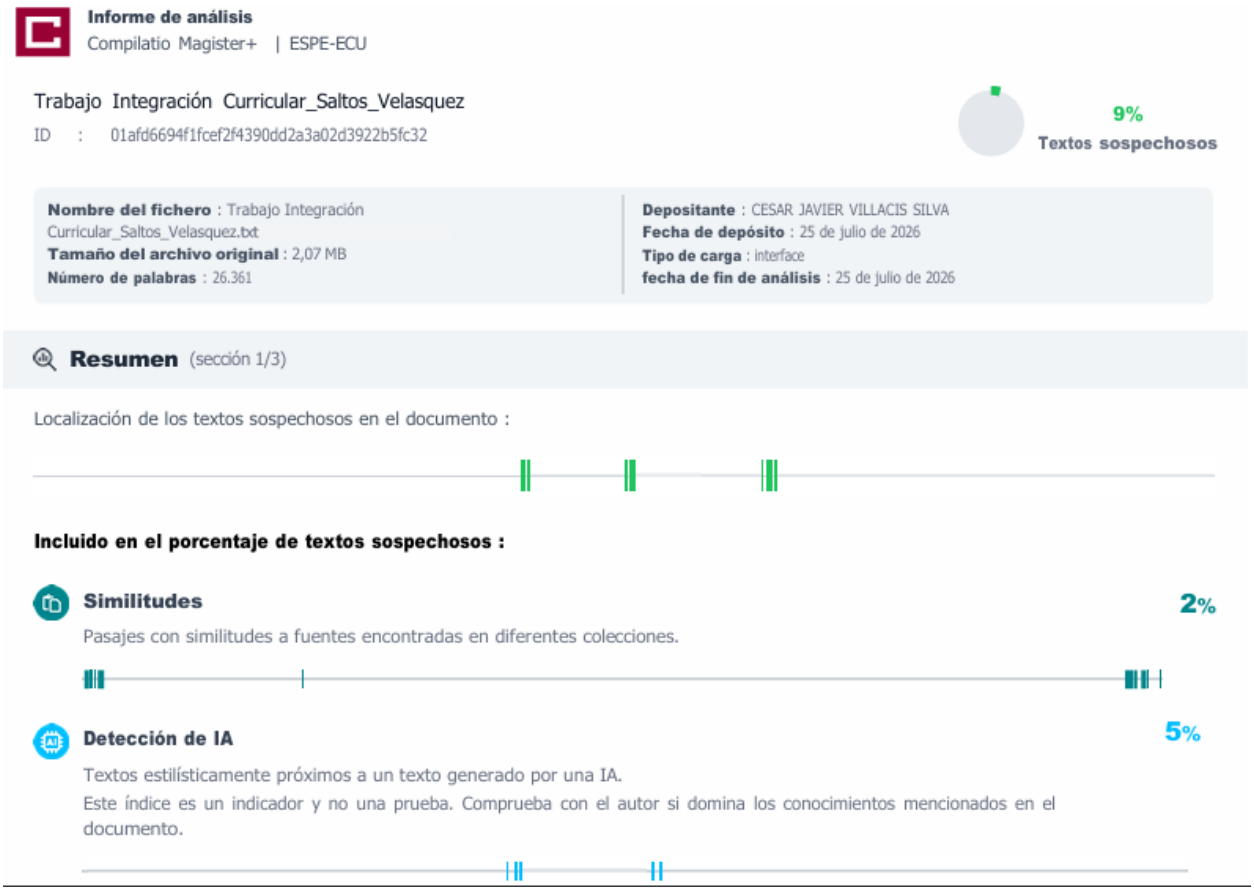
Departamento de Ciencias de la Computación

Carrera de Tecnologías de la Información

Trabajo de integración curricular, previo a la obtención del título de Ingeniero en Tecnologías de la Información.

Mgs. Villacís Silva César Javier

Sangolquí, 24 de Julio del 2026



Informe de análisis
Compilatio Magister+ | ESPE-ECU

Trabajo Integración Curricular_Saltos_Velasquez
ID : 01afd6694f1fcef2f4390dd2a3a02d3922b5fc32

9%
Textos sospechosos

Nombre del fichero : Trabajo Integración Curricular_Saltos_Velasquez.txt
Tamaño del archivo original : 2,07 MB
Número de palabras : 26.361

Depositante : CESAR JAVIER VILLACIS SILVA
Fecha de depósito : 25 de julio de 2026
Tipo de carga : interface
fecha de fin de análisis : 25 de julio de 2026

Resumen (sección 1/3)

Localización de los textos sospechosos en el documento :

Incluido en el porcentaje de textos sospechosos :

Similitudes 2%
Pasajes con similitudes a fuentes encontradas en diferentes colecciones.

Detección de IA 5%
Textos estilísticamente próximos a un texto generado por una IA.
Este índice es un indicador y no una prueba. Comprueba con el autor si domina los conocimientos mencionados en el documento.

Validar únicamente en FirmaEC.
Firmado electrónicamente por:
CESAR JAVIER
VILLACIS SILVA

Mgs. Villacís Silva César Javier

CC: 1704892726

Departamento de Ciencias de la Computación

Carrera de Tecnologías de la Información

Certificación

Certifico que el trabajo de integración curricular: "**Desarrollo de un videojuego educativo 3D tipo Zig-Zag con algoritmos de navegación autónoma y persecución basados en Inteligencia Artificial para el desarrollo del pensamiento y la estimulación de funciones ejecutivas en niños de 7 a 12 años**" fue realizado por los señores **Velasquez Males Tupac Yupanqui, Saltos Palacios Jean Michael,** el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE; habiéndose efectuado una revisión integral de forma y de fondo conforme a los lineamientos y formatos institucionales vigentes. Además, fue revisado y analizada en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Sangolquí, 24 de Julio del 2026

Validar únicamente en FirmaEC.
Firmado electrónicamente por:
CESAR JAVIER VILLACIS SILVA

Mgs. Villacís Silva César Javier

CC: 1704892726

Departamento de Ciencias de la Computación

Carrera de Tecnologías de la Información

Responsabilidad de Autoría

Nosotros, **Saltos Palacios Jean Michael,** con cédula de ciudadanía n° 1350196158 y **Velasquez Males Tupac Yupanqui,** con cédula de ciudadanía n° 1050332921, declaramos que el contenido, ideas y criterios del trabajo de integración curricular**: "Desarrollo de un videojuego educativo 3D tipo Zig-Zag con algoritmos de navegación autónoma y persecución basados en Inteligencia Artificial para el desarrollo del pensamiento y la estimulación de funciones ejecutivas en niños de 7 a 12 años"** es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Sangolquí, 24 de Julio del 2026

Firmado electrónicamente por:
TUPAC YUPANQUI
VELASQUEZ MALES

Velasquez Males, Tupac Yupanqui

CC: 1050332921

Firmado electrónicamente por:
JEAN MICHAEL
SALTOS PALACIOS

Saltos Palacios, Jean Michael

CC: 1350196158

Departamento de Ciencias de la Computación

Carrera de Tecnologías de la Información

Autorización de Publicación

Nosotros, **Saltos Palacios Jean Michael,** con cédula de ciudadanía n° 1350196158 y **Velasquez Males Tupac Yupanqui,** con cédula de ciudadanía n° 1050332921, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de integración curricular: "**Desarrollo de un videojuego educativo 3D tipo Zig-Zag con algoritmos de navegación autónoma y persecución basados en Inteligencia Artificial para el desarrollo del pensamiento y la estimulación de funciones ejecutivas en niños de 7 a 12 años"** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi/nuestra responsabilidad.

Sangolquí, 24 de Julio del 2026

Firmado electrónicamente por:
TUPAC YUPANQUI
VELASQUEZ MALES

Velasquez Males, Tupac Yupanqui

CC: 1050332921

Firmado electrónicamente por:
JEAN MICHAEL
SALTOS PALACIOS

Saltos Palacios, Jean Michael

CC: 1350196158

Índice de contenido

Certificación 3
Responsabilidad de Autoría 4
Resumen 14
Abstract....15
Capítulo I 16
Introducción 16
Planteamiento del problema....17
Formulación del problema o pregunta de investigación....18
Objetivos....19
Objetivo general....19
Objetivos específicos 19
Justificación 20
Alcance....22
Delimitaciones....23
Capítulo II 24
Estado del arte....24
Marco teórico 27
Videojuegos educativos y simulaciones 3D....27
Funciones ejecutivas como referente de diseño....28
Relación entre las demandas del juego y las funciones ejecutivas 28
Interacción niño-computador....30
Inteligencia artificial en videojuegos 31
Seek....31
Pursuit....32
Trayectoria segura, histéresis y salto autónomo....32

Motor Unity, física y renderizado 33
Persistencia local y analítica de juego 34
Ingeniería de requisitos 35
Scrum y XP en el desarrollo iterativo 35
Calidad del producto según ISO/IEC 25010 36
Modelo Tetrad Modificado 37
Estética (Aesthetics) 38
Control e Interfaces (Control and Interfaces) 38
Pruebas del Juego (Game Testing) 38
Generación de Sonido (Sound Generation) 39
Historia / Diseño Narrativo (Story) 39
Diseño del Juego (Game Design) 39
Guión Gráfico / Progresión de Flujo (Storyboard) 40
Tecnología (Technology) 41
Diseño Gráfico (Graphic Design) 41
Animación (Animation) 42
Manejo de Imágenes y de Datos (Image Handling & Data Management) 42
Mecánicas (Mechanics) 43
Algoritmos del Juego (Game Algorithms) 43
Algoritmos de Inteligencia Artificial (Artificial Intelligence Algorithms) 44
Aplicación en la Fase de Diseño del Sistema con XP 45
Aplicación del Modelo Tetrad Modificado en el videojuego 45
Capítulo III 47
Metodología y desarrollo de la solución 47
Enfoque de investigación 47
Tipo y diseño de investigación 47

Población, población accesible y muestra....48
Técnicas e instrumentos....49
Observación y telemetría....49
Encuesta de experiencia....50
Consistencia interna del instrumento....51
Procedimiento....51
Consideraciones éticas y protección de datos....52
Estrategia de análisis de datos....52
Proceso iterativo de desarrollo con Scrum y XP....53
Criterios de cierre y control de cada incremento....54
Especificación de requisitos funcionales....55
Requisitos no funcionales....56
Historias de usuario y Product Backlog....57
Trazabilidad entre objetivos, requisitos y pruebas....58
Arquitectura del videojuego....59
Flujo de escenas y operación....60
Controles, movimiento y física....61
Diseño de niveles, obstáculos, JumpPads y recolectables....62
Materialización de los nodos Tetrad en el desarrollo....63
Integración del flujo de juego, datos y retroalimentación....64
Implementación de Seek, Pursuit y navegación segura....66
Persistencia, métricas y diccionario operativo....67
Configuración y despliegue en Windows y Android....69
Diseño de pruebas....69
Indicadores de calidad ISO/IEC 25010....71
Documentación técnica y operativa....71

Modelado funcional consolidado y mantenibilidad....72
Secuencia de acceso y preparación de una partida....74
Colaboración entre clases implementadas....75
Interfaces operativas y evidencia de responsabilidad....76
Despliegue verificable en Windows y Android....77
Capítulo IV....78
Evaluación, resultados y discusión....78
Organización de los resultados....78
Calidad y cobertura de los datos....78
Resultados de telemetría de la población infantil....80
Resultados del grupo técnico complementario de participantes adultos....80
Comprobación mediante promedios por participante....81
Resultados por nivel....82
Lectura integrada de la progresión por bloques de niveles....85
Resultados de la encuesta por ítem....88
Coherencia entre contenido, polaridad y comportamiento de respuesta....89
Dimensiones y consistencia interna....91
Resultados de las respuestas abiertas....92
Resultados de pruebas de caja negra....93
Resultados de caja blanca y compilación....94
Resultados de Seek, Pursuit y navegación segura....95
Resultados por plataforma....96
Evaluación según ISO/IEC 25010....96
Resultados según el Modelo Tetrad Modificado....97
Discusión....98
Implicaciones para diseño, uso educativo y evaluación futura....99

Limitaciones 101

Capítulo V 102

Conclusiones 102

Recomendaciones 104

Referencias 106

Índice de tablas

Tabla 1 *Relación entre objetivos y evidencia del trabajo* 19

Tabla 2 *Delimitaciones del estudio* 23

Tabla 3 *Síntesis de antecedentes relacionados* 26

Tabla 4 *Criterios de ingeniería de requisitos aplicados* 35

Tabla 5 *Correspondencia entre nodos Tetrad y componentes del videojuego* 46

Tabla 6 *Diseño metodológico aplicado* 48

Tabla 7 *Distribución de participantes por grupo y plataforma* 49

Tabla 8 *Correspondencia de los ítems con las dimensiones analíticas* 50

Tabla 9 *Secuencia del procedimiento de evaluación* 52

Tabla 10 *Incrementos principales del desarrollo* 53

Tabla 11 *Requisitos funcionales de registro y flujo* 55

Tabla 12 *Requisitos funcionales de mecánicas e inteligencia artificial* 55

Tabla 13 *Requisitos funcionales de datos y plataformas* 56

Tabla 14 *Requisitos no funcionales del producto* 56

Tabla 15 *Historias de usuario de interacción y gameplay* 57

Tabla 16 *Historias de usuario de IA, datos y mantenimiento* 58

Tabla 17 *Trazabilidad de objetivos, requisitos, componentes y evidencia* 58

Tabla 18 *Responsabilidades de los componentes principales* 59

Tabla 19 *Mapa de controles y validaciones* 62

Tabla 20 *Función de diseño de los niveles 1 a 8* 63

Tabla 21 *Función de diseño de los niveles 9 a 16* 63

Tabla 22 *Decisiones del agente autónomo* 66

Tabla 23 *Diccionario operativo de métricas de nivel* 67

Tabla 24 *Configuración comparativa de plataformas* 69

Tabla 25 *Casos de prueba de caja negra* 70

Tabla 26 *Casos de prueba de caja blanca* 70

Tabla 27 *Indicadores seleccionados para evaluar la calidad*.....71

Tabla 28 *Catálogo de casos de uso de acceso, configuración y navegación* 73

Tabla 29 *Catálogo de casos de uso de partida, inteligencia artificial, resultados y datos* 73

Tabla 30 *Interfaces operativas y responsabilidades verificadas* 76

Tabla 31 *Resultados de integridad y cobertura* 79

Tabla 32 *Telemetría de la población infantil* 80

Tabla 33 *Telemetría del grupo técnico adulto*.....81

Tabla 34 *Promedios por participante de tiempo, intentos y muertes* 82

Tabla 35 *Promedios por participante de recolección, entradas aceptadas e influencia del enemigo*.....82

Tabla 36 *Resultados infantiles por nivel, niveles 1 a 8* 83

Tabla 37 *Resultados infantiles por nivel, niveles 9 a 16* 83

Tabla 38 *Resultados del grupo técnico adulto por nivel, niveles 1 a 8*.....83

Tabla 39 *Resultados del grupo técnico adulto por nivel, niveles 9 a 16*.....84

Tabla 40 *Resultados de encuesta por ítem, ítems 1 a 8*.....88

Tabla 41 *Resultados de encuesta por ítem, ítems 9 a 15*.....88

Tabla 42 *Resultados por dimensión y confiabilidad*.....91

Tabla 43 *Síntesis temática de respuestas abiertas* 93

Tabla 44 *Resultados ejecutados de caja negra*.....94

Tabla 45 *Resultados de verificación de caja blanca*.....95

Tabla 46 *Resultados del producto según características de calidad* 97

Índice de figuras

Figura 1 *Secuencia de navegación Seek/Pursuit y validación de seguridad* 33

Figura 2 *Modelo de la Tétrada Elemental Modificado (ETGDMM)* 37

Figura 3 *Distribución de los participantes efectivos*.....49

Figura 4 *Arquitectura lógica de componentes del videojuego* 59

Figura 5 *Flujo de escenas, progreso y resultado* 61

Figura 6 *Secuencia de captura y persistencia de métricas JSON*.....68

Figura 7 *Panorama consolidado de casos de uso del videojuego*.....72

Figura 8 *Secuencia validada de acceso, tutorial y carga de nivel* 74

Figura 9 *Colaboración entre clases de interfaz, partida y persistencia*.....75

Figura 10 *Estructura de despliegue y persistencia por plataforma* 77

Figura 11 *Cobertura de registros por nivel y grupo* 79

Figura 12 *Perfil de tiempo y muertes por nivel en la población infantil* 85

Figura 13 *Valoración media de los 15 ítems por grupo* 89

Figura 14 *Medias de las dimensiones de experiencia por grupo*.....92

Resumen

El trabajo tuvo como propósito diseñar, desarrollar, implementar y evaluar un videojuego educativo tridimensional tipo Zig-Zag para niños de 7 a 12 años, integrando navegación autónoma y persecución mediante inteligencia artificial. Se aplicó un enfoque mixto, aplicado y evaluativo, con comparación descriptiva de condiciones de juego configuradas por nivel. El producto se construyó en Unity 2022.3.62f3 con 16 niveles, tutorial, controles de escritorio y móviles, progresión, recolectables, obstáculos, plataformas de salto, persistencia local y un agente que combina Seek, Pursuit, validación de terreno, seguimiento de trayectoria y salto autónomo. La evaluación efectiva reunió 50 participantes: 44 niños de 9 a 12 años y un grupo técnico complementario de 6 adultos. Se analizaron 248 registros de nivel y una encuesta de 15 ítems con tres preguntas abiertas. En la población infantil se obtuvo un tiempo medio de 63.20 segundos por registro, 5.45 intentos, 4.30 muertes, 12.04 recolectables y 97.27 % de entradas aceptadas. La valoración media de la encuesta fue 4.28 sobre 5; las dimensiones presentaron consistencia interna entre .905 y .948. La diversión, la intención de continuar y el desafío del enemigo fueron bien valorados, mientras los controles y los cambios de orientación concentraron las principales dificultades cualitativas. Las verificaciones funcionales, estructurales y de integridad de datos mostraron operación en Windows y Android, cobertura de los 16 niveles y registros completos. Los resultados respaldan el cumplimiento de los objetivos de desarrollo y evaluación del producto, sin interpretar la telemetría o la percepción como evidencia causal de mejora clínica o cognitiva.

Palabras clave: videojuego educativo, inteligencia artificial, funciones ejecutivas, Seek y Pursuit, Unity

Abstract

This study aimed to design, develop, implement, and evaluate a three-dimensional Zig-Zag educational video game for children aged 7 to 12, integrating autonomous navigation and artificial-intelligence pursuit. An applied mixed-methods and evaluative approach was used, with descriptive comparisons of level-based game conditions. The product was built in Unity 2022.3.62f3 and includes 16 levels, a tutorial, desktop and mobile controls, progression, collectibles, obstacles, jump platforms, local persistence, and an agent that combines Seek, Pursuit, terrain validation, trail following, and autonomous jumping. The effective evaluation involved 50 participants: 44 children aged 9 to 12 and a complementary technical group of 6 adults. A total of 248 level records and a 15-item survey with three open-ended questions were analyzed. In the child population, the mean time was 63.20 seconds per record, with 5.45 attempts, 4.30 deaths, 12.04 collectibles, and 97.27% accepted inputs. The survey mean was 4.28 out of 5, and dimension-level internal consistency ranged from .905 to .948. Fun, willingness to continue, and the challenge provided by the enemy received favorable ratings, whereas controls and changes in orientation concentrated the main qualitative difficulties. Functional, structural, and data-integrity checks showed operation on Windows and Android, coverage of all 16 levels, and complete records. The findings support achievement of the product development and evaluation objectives, while gameplay telemetry and perceived experience are not interpreted as causal evidence of clinical or cognitive improvement.

Keywords: educational video game, artificial intelligence, executive functions, Seek and Pursuit, Unity

Capítulo I

Introducción

La incorporación de recursos digitales interactivos en contextos educativos ha ampliado las posibilidades de practicar atención sostenida, toma de decisiones, coordinación visomotora y resolución de problemas dentro de entornos seguros. Un videojuego educativo no se define solo por presentar contenidos o recompensas; su valor depende de la relación entre propósito, reglas, interacción, retroalimentación y evidencia de uso. Esa relación exige integrar diseño de juegos, ingeniería de software, interacción niño-computador y evaluación del producto.

Las funciones ejecutivas comprenden procesos como control inhibitorio, memoria de trabajo y flexibilidad cognitiva, que permiten sostener metas y adaptar la conducta ante cambios del entorno (Diamond, 2013; Miyake et al., 2000). Los juegos pueden plantear demandas compatibles con estos procesos, pero la presencia de un reto no demuestra por sí misma una mejora cognitiva. Para sostener una afirmación de cambio se requerirían mediciones válidas antes y después, condiciones comparables y un diseño que controle explicaciones alternativas. Por ello, este trabajo evalúa el funcionamiento y la experiencia de uso del videojuego y evita convertir indicadores de partida en diagnósticos psicológicos.

El proyecto se concretó como un videojuego 3D tipo Zig-Zag. El participante guía un personaje por plataformas suspendidas, decide cambios de dirección, recolecta objetos, evita caídas y responde a la presión de un agente perseguidor. La solución combina reglas de juego, física, interfaces, progresión, persistencia y una inteligencia artificial basada en comportamientos de dirección. El agente no emplea aprendizaje automático: calcula objetivos a partir de posición, velocidad, trayectoria y seguridad del terreno, seleccionando entre Seek y Pursuit.

Planteamiento del problema

En el ámbito educativo, todavía es frecuente que las actividades digitales dirigidas a niños se limiten a presentaciones lineales, cuestionarios estáticos o ejercicios con poca variación. Estas soluciones pueden transmitir información, pero ofrecen menos oportunidades para practicar decisiones bajo presión, explorar consecuencias y recibir retroalimentación inmediata. La literatura sobre juegos serios muestra resultados prometedores, aunque también advierte heterogeneidad en diseños, edades, instrumentos y duración de las intervenciones (Kou et al., 2024; Rodríguez Timaná et al., 2024). La calidad del juego y de la evaluación condiciona cualquier interpretación.

Un segundo desafío reside en la representación técnica del comportamiento autónomo. Un enemigo que sigue la posición actual del jugador puede acortar caminos, abandonar la plataforma o actuar de manera predecible. En una pista angosta y discontinua, el perseguidor debe anticipar sin sacrificar seguridad. Esto demanda combinar predicción, validación geométrica, seguimiento de la ruta recorrida, control de movimiento físico y mecanismos de recuperación. La conducta debe ser comprensible para el jugador y suficientemente estable para no convertir el reto en un evento arbitrario.

En Ecuador se han desarrollado videojuegos educativos orientados a biodiversidad, reciclaje, patrimonio cultural y salud. Los proyectos nacionales recientes confirman la viabilidad de Unity y de las experiencias tridimensionales, pero sus objetivos, públicos y formas de evaluación son diversos. Algunos valoran percepción mediante escalas Likert; otros incorporan pruebas antes y después o evaluación de expertos. Esta diversidad refuerza la necesidad de documentar con precisión qué mide cada instrumento y qué alcance tienen los resultados.

Para la formación en Tecnologías de la Información, el problema no se limita a producir un prototipo visual. Es necesario especificar requisitos, mantener trazabilidad, diseñar componentes, verificar plataformas, registrar datos y relacionar los resultados con características de calidad. IEEE 830-1998 proporciona una estructura conocida para la

especificación de requisitos y la norma ISO/IEC/IEEE 29148:2018 actualiza los principios de ingeniería de requisitos. ISO/IEC 25010:2023, por su parte, permite organizar la evaluación de la calidad del producto.

El problema específico se basa en diseñar una experiencia de navegación Zig-Zag que combinara controles breves, progresión por niveles, presión de persecución y registro de métricas, manteniendo una interacción comprensible para niños. La pista suspendida introduce riesgo de caída; los cambios de orientación exigen recordar la relación entre entrada y movimiento; los obstáculos y plataformas de salto amplían el repertorio de acciones; el perseguidor obliga a mantener continuidad.

La solución debe operar en Windows y Android, conservar progreso, presentar tutorial, informar el resultado y almacenar telemetría local. Además, el agente debía seguir al jugador sin atravesar el vacío, anticipar cuando la velocidad ofreciera una predicción útil, retroceder a una conducta segura cuando la predicción no fuera válida y superar discontinuidades mediante salto autónomo. La evaluación debía separar a la población infantil del grupo técnico adulto y vincular cada afirmación con datos de interacción, encuesta o prueba técnica.

La necesidad identificada fue disponer de un videojuego educativo 3D que superara la interacción estática mediante decisiones de navegación, persecución autónoma segura, dificultad por niveles, retroalimentación y evaluación basada en datos, sin confundir la experiencia de juego con una medición clínica.

Formulación del problema o pregunta de investigación

¿De qué manera el diseño, desarrollo, implementación y evaluación de un videojuego educativo 3D tipo Zig-Zag, respaldado por algoritmos de Inteligencia Artificial para navegación autónoma y persecución, permite estructurar un entorno interactivo de demandas de decisión y coordinación para niños de 7 a 12 años, sustentando su funcionamiento mediante telemetría local, pruebas de calidad y percepción de uso?

Objetivos

Objetivo general

Diseñar, desarrollar, implementar y evaluar un videojuego educativo 3D tipo Zig-Zag con algoritmos de navegación autónoma y persecución basados en Inteligencia Artificial para el desarrollo del pensamiento y la estimulación de funciones ejecutivas en niños de 7 a 12 años.

Objetivos específicos

- Fundamentar teóricamente el diseño de simulaciones 3D y sistemas de navegación autónoma para su aplicación en el entrenamiento cognitivo.
- Establecer las especificaciones técnicas y funcionales del simulador y del comportamiento del agente inteligente, alineadas al estándar IEEE-830.
- Implementar los componentes del software (motor físico, IA y gestión de datos) mediante un proceso iterativo incremental basado en Scrum y XP.
- Evaluar el desempeño del sistema y la efectividad de la IA mediante pruebas de calidad basadas en la norma ISO/IEC-25010.
- Generar la documentación técnica y los manuales operativos que soporten la mantenibilidad y uso del entorno virtual.

Tabla 1

Relación entre objetivos y evidencia del trabajo

Objetivo	Evidencia principal	Resultado esperado
General	Producto, pruebas, telemetría y encuesta	Videojuego implementado y evaluado
Específico 1	Estado del arte y marco teórico	Fundamento de diseño y navegación
Específico 2	Requisitos, historias y trazabilidad	Especificación verificable
Específico 3	Arquitectura, componentes y niveles	Integración funcional en Unity
Específico 4	Pruebas y resultados ISO/IEC 25010	Evidencia de calidad del producto
Específico 5	Descripción técnica y guía operativa	Soporte para uso y mantenimiento

Nota. La tabla identifica la evidencia desarrollada en los capítulos II a V y no sustituye la formulación literal de los objetivos.

Justificación

La propuesta ofrece una experiencia breve y repetible en la que el participante debe atender el camino, seleccionar una dirección, inhibir respuestas impulsivas y adaptar su ritmo ante cambios de trazado. Estas demandas son compatibles con componentes de las funciones ejecutivas, pero el aporte educativo se formula como oportunidad de práctica y no como tratamiento. La distinción protege la validez de las conclusiones y permite que futuras investigaciones incorporen instrumentos estandarizados.

La retroalimentación inmediata —recolectables, estrellas, mensajes, resumen y reintento— permite comprender consecuencias y ajustar la estrategia. La progresión mantiene una secuencia de aprendizaje desde controles básicos hasta obstáculos y saltos. Las respuestas de los usuarios aportan información para valorar claridad, diversión, reto, concentración percibida y comprensión del resultado.

La navegación de un agente sobre plataformas suspendidas constituye un problema técnico relevante porque una predicción geométricamente correcta puede ser inválida para el terreno. La solución combina steering behaviors, muestreo de suelo, control de altura, validación de segmentos, historial de trayectoria, histéresis, límites de predicción, salto y respawn. Este conjunto demuestra la aplicación de inteligencia artificial para juegos sin depender de aprendizaje automático.

El producto integra arquitectura por componentes, física mediante Rigidbody, persistencia JSON/CSV y configuración multiplataforma. La telemetría se incorpora desde el propio videojuego y permite describir tiempo, intentos, muertes, recolectables, entradas y eventos asociados al enemigo. La especificación y la evaluación con estándares fortalecen la trazabilidad entre propósito, software y resultados.

El enfoque mixto combina la precisión de los registros automáticos con la perspectiva de los participantes. La telemetría muestra lo que ocurrió durante las partidas; la encuesta muestra cómo fue percibida la interacción; las respuestas abiertas explican dificultades y

preferencias que una media no revela. La separación entre población infantil y grupo técnico adulto evita presentar cohortes de distinta edad como si fueran equivalentes.

Socialmente, el trabajo aporta una herramienta local, ejecutable sin conexión y adaptable a contextos educativos. También aporta criterios para desarrollar software dirigido a niños con lenguaje claro, retroalimentación visible, control de datos y límites interpretativos. La documentación permite que docentes, investigadores y desarrolladores comprendan el alcance del producto antes de reutilizarlo.

Alcance

El foco del Trabajo de Integración Curricular es el diseño, desarrollo e implementación de un videojuego educativo 3D tipo Zig-Zag, enfocado en plantear demandas de decisión y coordinación para niños de 7 a 12 años, junto con la evaluación empírica de su funcionamiento, rendimiento de Inteligencia Artificial y percepción de uso. Por lo cual, se abarca el desarrollo de una solución integral en el motor Unity, la cual comprende las siguientes directrices descritas de forma general:

- **Interfaz Gráfica de Usuario (GUI) e Interacción en Unity:** Desarrollo de pantallas de registro local de usuario (nombre y edad), tutorial guiado obligatorio, menú principal, selección y desbloqueo progresivo de niveles, pausa, pantalla de resultados con sistema de estrellas y tabla de clasificación local.
- **Mecánicas de Juego 3D y Control:** Implementación de mecánicas de navegación tipo Zig-Zag con controles lateralizados (escritorio y táctil), recolectables, obstáculos físicos, zonas de caída y plataformas de salto (*JumpPads*).
- **Navegación Autónoma e Inteligencia Artificial:** Implementación de algoritmos de dirección (*Steering Behaviors*) que combinan los modos *Seek* y *Pursuit* condicionado, integrando validación geométrica de terreno seguro, seguimiento de trayectoria histórica y salto autónomo del agente persecutor.
- **Persistencia y Analítica de Juego:** Manejo y almacenamiento de estados mediante archivos planos locales (JSON por nivel para telemetría detallada de interacción y CSV para el resumen acumulado de sesión).
- **Validación de Plataforma y Evaluación Empírica:** Despliegue funcional para computadoras de escritorio (.exe en Windows) y dispositivos móviles (.apk en Android), respaldado por pruebas de software (caja negra y blanca), evaluación de calidad bajo el estándar ISO/IEC 25010:2023, telemetría de uso y encuestas de experiencia aplicadas a una muestra de participantes.

Delimitaciones

Las delimitaciones que orientan la interpretación se presentan en la tabla siguiente.

Tabla 2

Delimitaciones del estudio

Dimensión	Delimitación	Implicación
Población	Público objetivo de 7 a 12 años	La evaluación infantil se interpreta dentro de su cobertura efectiva
Intervención	Sesiones de juego y encuesta posterior	No estima cambio longitudinal
Diseño	Comparativo, transversal y evaluativo	No atribuye causalidad
Datos	Telemetría local por participante y nivel	Cobertura desigual entre niveles
IA	Steering behaviors deterministas	No utiliza aprendizaje automático
Calidad	Indicadores disponibles del producto	No sustituye certificación formal

Nota. Las delimitaciones se establecen a partir del propósito, el diseño aplicado, la estructura de la telemetría y las características implementadas.

Capítulo II

Estado del arte

La revisión se organizó en antecedentes nacionales e internacionales relacionados con videojuegos educativos, interacción infantil, funciones ejecutivas, diseño de juegos e inteligencia artificial. La comparación no busca equiparar productos distintos, sino identificar decisiones metodológicas transferibles: definición del público, construcción iterativa, medición de experiencia, uso de grupos comparables y límites de generalización.

En cada antecedente se examinaron objetivo, método, población, tecnología, resultado, limitación y relación con el presente trabajo. Esta estructura evita reducir el estado del arte a una lista de títulos y permite explicar qué aporta cada estudio a la formulación de la solución.

Villacís et al. (2016) desarrollaron un juego educativo de tres en raya que combinó análisis semiótico, Extreme Programming, inteligencia artificial y una modificación de la Tétrada Elemental. El trabajo fue probado con niños de 7 a 11 años y organizó el diseño en nodos relacionados con estética, historia, tecnología y mecánicas. Su aporte principal para este TIC es el Modelo Tetrad Modificado y la integración entre diseño, algoritmos e iteración. La diferencia radica en que Zig-Zag trabaja navegación física continua, persecución y telemetría por nivel.

Loja Marín (2025) desarrolló un videojuego 3D de plataformas para divulgar la biodiversidad de la Amazonía ecuatoriana. Aplicó Lean UX y encuestas Likert para valorar la percepción del carácter educativo y lúdico. El proyecto demuestra la utilidad de ciclos iterativos y entornos 3D contextualizados. Su límite para la comparación es que el registro público no detalla una medición de funciones ejecutivas ni un agente de persecución; su relación se concentra en diseño de plataformas, experiencia y valoración posterior.

Padilla Hinojosa (2026) creó EcoAventuras, un videojuego 3D en Unity para educación ambiental. Empleó un enfoque aplicado y cualitativo, niveles progresivos, modelado low-poly, evaluación con un experto ambiental y sesiones con docentes. Obtuvo un prototipo de tres

niveles y valoración favorable de pertinencia. El antecedente evidencia la importancia de integrar contenido, progresión y validación de especialistas, aunque su población evaluadora y sus indicadores difieren de la evaluación directa con niños realizada en Zig-Zag.

González Campoverde y Vintimilla Torres (2026) diseñaron una experiencia de realidad virtual para hábitos de higiene bucal en 16 participantes de 8 a 15 años. Utilizaron Design Thinking, Blender, Unity, pretest, postest y evaluación de experiencia. El conocimiento aumentó de 3.38 a 4.81 sobre 6 y la experiencia obtuvo 6.19 sobre 7. El tamaño no probabilístico limita la generalización, pero su diseño muestra que una afirmación de efectividad requiere mediciones previas y posteriores, condición que no se reemplaza con telemetría de juego.

Arce Maldonado y Chabla Muñoz (2026) desarrollaron un sistema gamificado para cultura Cañari con narrativas generadas por un modelo multimodal. La solución combinó servicios, Kotlin, Flask e inteligencia artificial generativa, y fue valorada con SUS y TAM. Su resultado demuestra otra forma de integrar IA y gamificación en Ecuador. Sin embargo, su arquitectura, objetivo cultural y tipo de IA son distintos; la relación con este TIC se ubica en la trazabilidad entre tecnología, usabilidad y propósito educativo.

Parong et al. (2017) diseñaron Alien Game para practicar cambio entre tareas. Tres experimentos con universitarios compararon un juego focalizado con un juego de control o con inactividad. Dos horas distribuidas en cuatro sesiones produjeron mejoras en pruebas de shifting, mientras una hora no mostró el mismo efecto. El antecedente demuestra que el tiempo, la dificultad y la medida externa son condiciones centrales para estudiar transferencia; su población adulta impide extrapolar el efecto a niños.

Zeng et al. (2023) realizaron un ensayo cruzado aleatorizado con 34 niños de edad preescolar. Compararon 12 semanas de exergaming con práctica habitual y utilizaron Dimensional Change Card Sort y acelerometría. La diferencia en funciones ejecutivas fue pequeña y el intervalo incluyó cero, por lo que los autores solicitaron ensayos mayores. El estudio ilustra la necesidad de comparadores, instrumentos externos y seguimiento temporal.

Kou et al. (2024) sintetizaron 37 ensayos aleatorizados con 2771 niños y adolescentes. Encontraron efectos sobre flexibilidad cognitiva, control inhibitorio, cognición global y habilidades motoras, pero no un efecto concluyente sobre memoria de trabajo. La heterogeneidad de poblaciones y actividades limita la transferencia directa. Para Zig-Zag, el antecedente justifica tratar por separado cada función y no convertir una experiencia interactiva en evidencia general de entrenamiento.

Rodríguez Timaná et al. (2024) revisaron 16 estudios sobre juegos serios y funciones ejecutivas en población neurodiversa. Reportaron resultados prometedores en atención, memoria de trabajo y flexibilidad, junto con variabilidad en muestras, edades, tecnologías y calidad. El trabajo respalda la relación conceptual entre juegos y funciones ejecutivas, pero también exige evaluación específica por producto.

Saiger et al. (2023) analizaron 47 estudios sobre participación de niños y jóvenes en el diseño de juegos aplicados. Identificaron 45 actividades y señalaron comprensión, cohesión, confianza, accesibilidad y tiempo como factores de participación. El presente TIC no se define como codiseño: los niños participaron en la evaluación del producto. El antecedente orienta el lenguaje, el tutorial y la separación de roles.

Tabla 3

Síntesis de antecedentes relacionados

Estudio	Método y población	Resultado y limitación	Relación con Zig-Zag
Villacís et al. (2016)	XP; niños de 7–11; juego con IA	Modelo modificado y prototipo; dominio distinto	Marco Tetrad e integración iterativa
Loja Marín (2025)	Lean UX; encuesta Likert	Videojuego 3D educativo; detalle muestral limitado	Plataformas, percepción y diseño local
Padilla Hinojosa (2026)	Aplicado cualitativo; experto y docentes	Tres niveles en Unity; sin medición infantil directa	Progresión y evaluación de pertinencia
González y Vintimilla (2026)	Pre/post; n = 16, 8–15 años	Aumento de conocimiento; muestra no probabilística	Criterio para hablar de efectividad
Parong et al. (2017)	Tres experimentos con universitarios	Transferencia tras dos horas; población adulta	Práctica focalizada y medida externa

Estudio	Método y población	Resultado y limitación	Relación con Zig-Zag
Zeng et al. (2023)	Ensayo cruzado; n = 34 preescolares	Efecto pequeño; requiere mayor muestra	Comparación y medición validada
Kou et al. (2024)	Meta-análisis de 37 ensayos	Efectos diferenciados; alta diversidad	Interpretación por función
Rodríguez Timaná et al. (2024)	Revisión sistemática de 16 estudios	Resultados prometedores; heterogeneidad	Juegos serios y funciones ejecutivas
Saiger et al. (2023)	Scoping review de 47 estudios	Factores de participación; reporte desigual	Interacción y evaluación con niños

Nota. La comparación utiliza publicaciones científicas y repositorios universitarios. Los resultados se presentan en su propio diseño y población; no se transfieren como efectos del videojuego desarrollado en este trabajo.

Marco teórico

Videojuegos educativos y simulaciones 3D

Un videojuego educativo articula objetivos de aprendizaje o práctica con una estructura de juego capaz de mantener participación. La mecánica debe constituir el núcleo de la experiencia y no un adorno añadido a contenido expositivo. Grey et al. (2017) sostienen que los resultados de aprendizaje deben incorporarse a las mecánicas para evitar que el comportamiento emergente se aleje del propósito. La relación entre reglas, dinámica y experiencia exige revisar el juego como un sistema de decisiones conectado (Hunicke et al., 2004; Schell, 2019). En Zig-Zag, el propósito se expresa mediante decisiones continuas de dirección, control del ritmo, observación de la ruta y respuesta al perseguidor.

La tridimensionalidad aporta profundidad, perspectiva, física y variación espacial. También introduce exigencias de cámara, escala, colisiones y orientación. Una pista suspendida hace visible la consecuencia de una decisión incorrecta, mientras plataformas, recolectables y obstáculos crean puntos de atención. El diseño requiere que el error sea comprensible y recuperable: el participante debe saber por qué cayó o fue alcanzado y cómo reintentar.

La progresión por niveles distribuye la carga. Los primeros escenarios consolidan relación entre entrada y giro; los intermedios combinan velocidad, curvas y recolectables; los avanzados agregan obstáculos y salto. La dificultad es predefinida por escena. No se modifica

automáticamente según el rendimiento individual, por lo que debe denominarse escalonada y no adaptativa.

Funciones ejecutivas como referente de diseño

Diamond (2013) organiza las funciones ejecutivas básicas en control inhibitorio, memoria de trabajo y flexibilidad cognitiva. El control inhibitorio permite resistir una respuesta dominante; la memoria de trabajo mantiene información relevante; la flexibilidad permite cambiar de regla o perspectiva. Estas funciones se relacionan, pero conservan componentes diferenciables (Miyake et al., 2000). Las intervenciones revisadas por Diamond y Lee (2011) muestran que la práctica debe analizarse según tarea, duración y medida, sin asumir transferencia automática.

La mecánica Zig-Zag plantea demandas compatibles con estos procesos. Recordar el mapeo de controles se vincula con mantenimiento de reglas; esperar el momento de giro o salto se vincula con inhibición; adaptar la respuesta a una orientación distinta se vincula con flexibilidad; atender simultáneamente al camino, objetos y enemigo se vincula con atención ejecutiva. La relación es funcional y teórica.

Las medidas capturadas por el juego —tiempo, intentos, muertes, clics y recolectables— describen desempeño dentro del sistema. No sustituyen pruebas como Trail Making Test o Torre de Londres. Estas herramientas se reconocen como referentes de evaluación neuropsicológica, pero no fueron aplicadas y, por tanto, no generan resultados en este estudio.

Relación entre las demandas del juego y las funciones ejecutivas

La relación entre diseño y función ejecutiva se estableció en el nivel de la tarea. Cada acción de juego presenta información, solicita una decisión y produce una consecuencia observable. Cuando el personaje se aproxima a un cambio de dirección, el participante debe conservar la regla de entrada, atender la geometría inmediata y ejecutar la respuesta en una

ventana temporal. Si además se aproxima el perseguidor, la decisión incorpora presión, pero continúa siendo una conducta situada dentro del videojuego. Esta descripción permite fundamentar el diseño sin atribuir a una sola métrica la medición directa de un constructo psicológico.

El control inhibitorio se representa como demanda cuando una respuesta apresurada conduce a una caída o cuando es necesario evitar una entrada adicional. El sistema registra clics, entradas aceptadas, intentos y muertes, pero esos valores también dependen de experiencia previa, comprensión del tutorial, dispositivo, velocidad del nivel y características del trazado. Por ello, una muerte no se interpreta como ausencia de inhibición y un porcentaje elevado de entradas aceptadas tampoco se interpreta como capacidad ejecutiva. El indicador conserva su significado operacional: describe cómo procesó el software las entradas realizadas.

La memoria de trabajo interviene como demanda al mantener activa la correspondencia entre entrada y movimiento, recordar el objetivo del nivel y actualizar la posición relativa de obstáculos, recolectables y enemigo. A diferencia de una prueba estandarizada, el videojuego no aísla esta función: las claves permanecen visibles y la acción se apoya en información perceptiva continua. El diseño utiliza esa combinación para sostener la jugabilidad. La evaluación, en cambio, se limita a describir desempeño y percepción dentro del entorno, sin asignar una puntuación de memoria de trabajo.

La flexibilidad cognitiva aparece como demanda en los cambios de orientación, en la transición entre desplazamiento y salto y en la necesidad de reorganizar prioridades cuando el perseguidor reduce la distancia. La dificultad cualitativa asociada a los controles indica que la regla no siempre fue inmediata. Ese resultado es útil para mejorar tutorial y retroalimentación, pero admite explicaciones motrices y de familiaridad con videojuegos. La interpretación adecuada relaciona la demanda de cambio con la mecánica y contrasta esa relación con lo que expresaron los participantes.

La atención ejecutiva se distribuye entre ruta, objetos, información del HUD y agente. Los niveles avanzados introducen más elementos y, en los niveles 15 y 16, una acción de salto. El aumento no se considera una progresión cognitiva automática; es una progresión de condiciones de juego. Su análisis requiere observar tiempo, muertes, intentos, recolección y comentarios en conjunto. Una variación aislada puede surgir de un trazado más largo, de menor cobertura o de una ejecución especialmente extensa, de modo que las medidas se presentan con tendencia central y dispersión.

Esta fundamentación determina el alcance de la palabra estimulación en el proyecto: el videojuego propone situaciones compatibles con el ejercicio de decisiones, atención y adaptación durante la partida. No demuestra transferencia a tareas no jugadas, permanencia del cambio ni efecto clínico. Las revisiones de Kou et al. (2024) y Rodríguez Timaná et al. (2024) muestran que esas conclusiones demandan diseños comparativos, instrumentos válidos y control de heterogeneidad. La presente evaluación aporta evidencia de funcionamiento, exposición a las demandas diseñadas y experiencia de uso, que constituye una base verificable para estudios posteriores.

Interacción niño-computador

El diseño para niños exige considerar etapa de desarrollo, comprensión del lenguaje, experiencia previa, motricidad, atención y contexto. Lehnert et al. (2022) identifican una diversidad de métodos de interacción infantil y señalan la necesidad de adaptar procesos y herramientas. Saiger et al. (2023) muestran que la participación puede adoptar roles distintos, desde informante hasta socio de diseño.

En este producto, el tutorial utiliza instrucciones breves y demostración práctica; el HUD prioriza tiempo, recolectables y progreso; los estados de victoria y derrota ofrecen acciones directas; los controles táctiles replican las acciones esenciales. La encuesta pregunta por comprensión, aprendizaje de controles, respuesta, claridad, utilidad del tutorial y

reconocimiento de objetos, lo que permite examinar la interacción sin asumir que una interfaz diseñada por adultos es automáticamente adecuada.

La accesibilidad se abordó mediante texto visible, iconografía, controles limitados, realimentación visual y audio configurable. No se evaluaron lectores de pantalla, contraste clínico, diversidad motora ni necesidades educativas específicas. Por ello, accesibilidad se usa como criterio de diseño general y no como certificación de conformidad.

Inteligencia artificial en videojuegos

La inteligencia artificial de juegos busca producir conductas eficaces, legibles y coherentes con las reglas. Puede utilizar planificación, búsqueda, máquinas de estados, árboles de comportamiento, steering behaviors o aprendizaje automático. La elección depende del problema. En una persecución continua sobre plataformas, un comportamiento de dirección ofrece control directo de velocidad y objetivo.

Reynolds (1999) formalizó steering behaviors como fuerzas o velocidades deseadas que permiten a personajes autónomos buscar, huir, llegar, perseguir y evitar. Millington (2019) diferencia el objetivo cinemático del control que lo materializa. Zig-Zag emplea Seek para dirigirse a un objetivo actual o seguro y Pursuit para predecir una posición futura acotada.

Seek

Seek calcula un vector desde la posición del agente hacia el objetivo actual. Si p_a representa la posición del agente y p_o la posición objetivo, la dirección deseada se expresa como $d = \text{normalizar}(p_o - p_a)$. La velocidad deseada resulta de multiplicar d por una rapidez máxima. En el videojuego, el vector se proyecta al plano horizontal y se integra con la física del cuerpo rígido.

Seek es robusto cuando la velocidad del jugador no está disponible, la distancia es corta o una predicción no dispone de suelo. Su desventaja es que sigue el punto presente y

puede reaccionar tarde ante cambios rápidos. Por ello funciona como modo base y como mecanismo de recuperación.

Pursuit

Pursuit estima una posición futura $\hat{p}$ mediante $\hat{p} = p_j + v_j t$, donde p_j es la posición del jugador, v_j su velocidad y t un horizonte de predicción. El videojuego limita t entre valores configurables y acota la magnitud del desplazamiento previsto. Si la velocidad es muy baja o los datos no son finitos, la predicción se descarta.

La anticipación no garantiza transitabilidad. Una posición futura puede encontrarse sobre el vacío, después de una esquina o en otra altura. Por eso el sistema busca suelo, valida la diferencia vertical y muestrea el segmento. Cuando el candidato no es seguro, reduce el desplazamiento o selecciona un objetivo de la trayectoria recorrida.

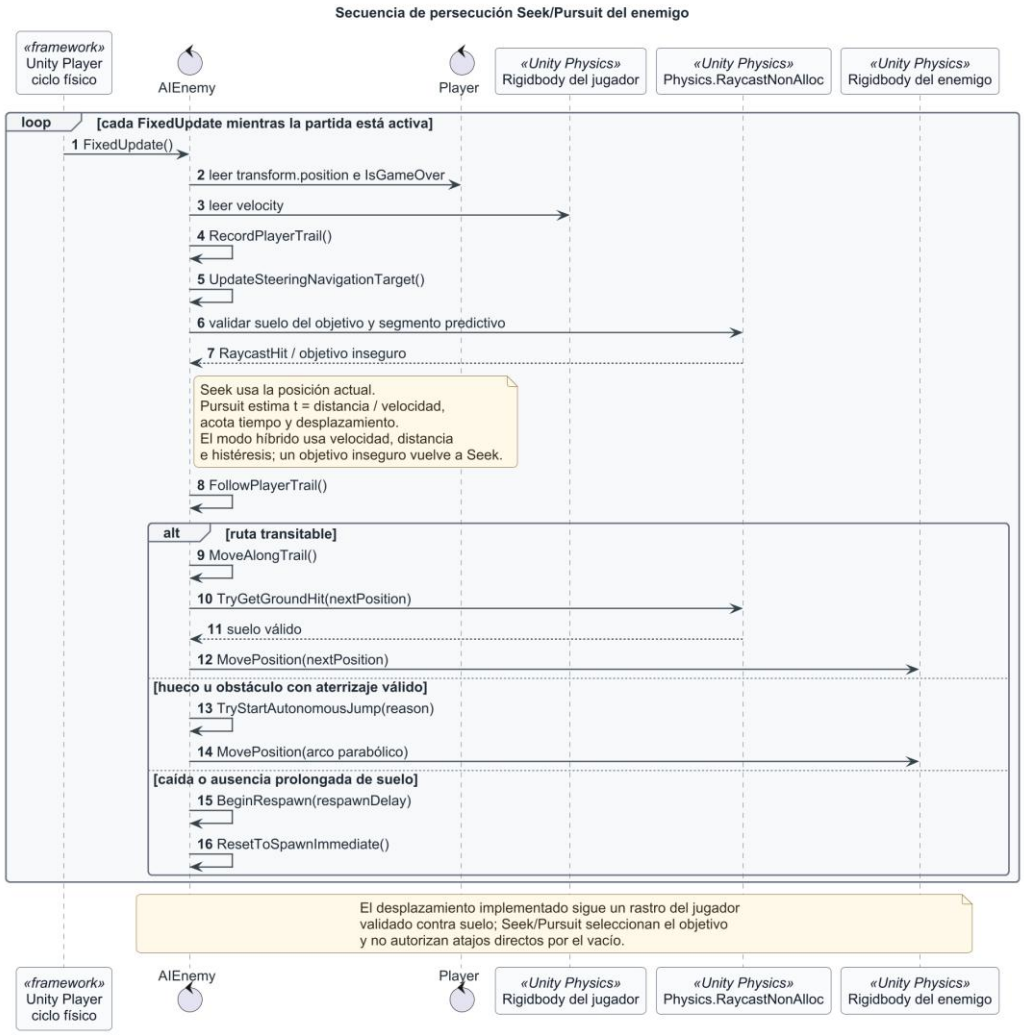
Trayectoria segura, histéresis y salto autónomo

El historial de trayectoria almacena muestras del recorrido del jugador separadas por una distancia mínima. Las muestras se validan contra el suelo y se acumulan por distancia de ruta. El agente puede seguir puntos y esquinas en lugar de cortar diagonalmente. Este mecanismo es especialmente importante cuando la distancia euclidiana es pequeña, pero la ruta segura exige rodear una esquina.

La histéresis evita alternancias rápidas entre Seek y Pursuit. El umbral de cambio se desplaza según el modo activo, de modo que una variación mínima de distancia no provoque oscilación. El suavizado limita cambios bruscos de dirección. El salto autónomo se activa cuando la trayectoria indica una discontinuidad o falta de suelo, siempre que exista un punto de aterrizaje válido y se cumplan tiempos de reutilización.

Figura 1

Secuencia de navegación Seek/Pursuit y validación de seguridad



Nota. Diagrama de la lógica implementada en el componente del agente. Incluye selección de modo, predicción acotada, validación de suelo y segmento, seguimiento de trayectoria, salto autónomo y retorno a Seek cuando el objetivo predictivo no es transitable.

Motor Unity, física y renderizado

Unity organiza el producto mediante escenas, GameObjects y componentes. La versión 2022.3.62f3 corresponde a una edición LTS del motor. Los componentes de movimiento emplean Rigidbody y ejecutan operaciones físicas en FixedUpdate, mensaje que Unity destina a cálculos independientes de la frecuencia de renderizado (Unity Technologies, 2022a).

El ciclo Update recibe entradas y actualiza interfaz; FixedUpdate aplica velocidad, impulso y rotación física. Esta separación reduce la dependencia del número de cuadros. Los

colliders y triggers diferencian suelo, recolectables, obstáculos, bandera y contacto con el enemigo. Raycasts y SphereCasts apoyan detección de suelo, validación de altura y búsqueda de aterrizaje.

Las escenas contienen cámara, iluminación, plataformas, personaje, agente, HUD y objetos. Los prefabs favorecen consistencia, mientras los parámetros serializados permiten ajustar velocidad, fuerza de salto, horizonte de predicción, tolerancias y capas sin reescribir lógica. La orientación horizontal y los controles táctiles permiten trasladar la experiencia a Android.

Persistencia local y analítica de juego

La persistencia combina PlayerPrefs para identidad, tutorial, niveles y estrellas, JSON para métricas detalladas de nivel y CSV para el resumen de sesión. Unity proporciona persistentDataPath como ubicación estable entre ejecuciones y JsonUtility para serializar campos públicos (Unity Technologies, 2022b, 2022c). El diseño es local: no existe servidor, API o base central.

La analítica de juegos requiere definir cada indicador, su fórmula y su unidad. Daoudi (2022) destaca que la analítica puede apoyar la usabilidad de juegos serios cuando se integra con propósito y contexto. En Zig-Zag, un registro JSON corresponde a una combinación participante–nivel. El tiempo se mide en segundos; intentos y muertes son conteos; los recolectables se expresan frente a un total; el primer clic y el intervalo tras respawn son duraciones.

El campo almacenado como click_accuracy_percentage se interpreta como porcentaje de entradas aceptadas por la lógica de juego: entradas efectivas divididas para clics del nivel y multiplicadas por 100. No mide exactitud cognitiva. La precisión terminológica evita transformar una condición de software en una capacidad del participante.

Ingeniería de requisitos

IEEE 830-1998 describe cualidades y estructuras de una especificación de requisitos de software. Aunque fue sustituida, se conserva porque forma parte del objetivo aprobado. ISO/IEC/IEEE 29148:2018 actualiza la ingeniería de requisitos y enfatiza que un buen requisito debe ser necesario, apropiado, inequívoco, completo, singular, factible, verificable y trazable.

En el proyecto, los requisitos se expresan con identificador, prioridad, descripción, criterio de aceptación y relación con componentes y pruebas. Esta estructura permite distinguir necesidad de diseño, comportamiento implementado y evidencia de evaluación. Las historias de usuario traducen necesidades a lenguaje operativo, y el backlog ordena incrementos.

La relación entre IEEE 830 y la especificación aplicada se resume en la tabla siguiente.

Tabla 4

Criterios de ingeniería de requisitos aplicados

Criterio	Aplicación	Evidencia
Corrección	Enunciados coherentes con el producto	Comportamiento observable
No ambigüedad	Un resultado principal por requisito	Criterio de aceptación
Completitud	Entradas, proceso y salida	Dominio funcional y no funcional
Consistencia	Reglas comunes de progreso y datos	Trazabilidad entre componentes
Verificabilidad	Condición observable o medible	Caso de prueba o indicador
Trazabilidad	Vínculo con objetivo y prueba	Matriz integrada en el capítulo III
Modificabilidad	Identificadores y agrupación por dominio	Backlog priorizado

Nota. Los criterios combinan la estructura solicitada por IEEE 830-1998 con las propiedades de requisitos definidas por ISO/IEC/IEEE 29148:2018.

Scrum y XP en el desarrollo iterativo

Scrum organiza el trabajo mediante un Product Backlog, objetivos, incrementos y revisión frecuente (Schwaber & Sutherland, 2020). XP complementa con prácticas de ingeniería como diseño simple, integración continua, refactorización y retroalimentación rápida (Beck & Andres, 2004). En un proyecto académico de dos desarrolladores, estas ideas se adaptaron a incrementos funcionales sin reproducir una estructura organizacional de gran escala.

El desarrollo avanzó desde un estado funcional inicial hacia registro, tutorial, niveles, progresión, estrellas, telemetría, versión móvil y navegación explícita Seek/Pursuit. Cada incremento integró escenas y componentes antes de extender el alcance. Las correcciones de cámara, niveles, datos y persecución responden a la retroalimentación obtenida durante uso y pruebas.

La combinación Scrum–XP permite explicar dos perspectivas: Scrum prioriza qué valor incorporar en el siguiente incremento; XP orienta cómo mantener el código ejecutable mientras se incorporan cambios. La trazabilidad con requisitos evita que la iteración se reduzca a cambios visuales sin criterio de aceptación.

Calidad del producto según ISO/IEC 25010

ISO/IEC 25010:2023 define un modelo de calidad de producto con nueve características: adecuación funcional, eficiencia de desempeño, compatibilidad, capacidad de interacción, fiabilidad, seguridad, mantenibilidad, flexibilidad y seguridad operacional. El estándar se utiliza como referencia para especificar objetivos, seleccionar indicadores y organizar resultados, no como declaración de certificación.

La adecuación funcional se relaciona con requisitos y escenarios; la eficiencia con tiempos y estabilidad de ejecución; la capacidad de interacción con tutorial, controles, claridad y retroalimentación; la fiabilidad con persistencia y manejo de estados; la mantenibilidad con componentes y compilación; la flexibilidad con despliegue en Windows y Android. La seguridad se aborda mediante almacenamiento local y pseudonimización del análisis.

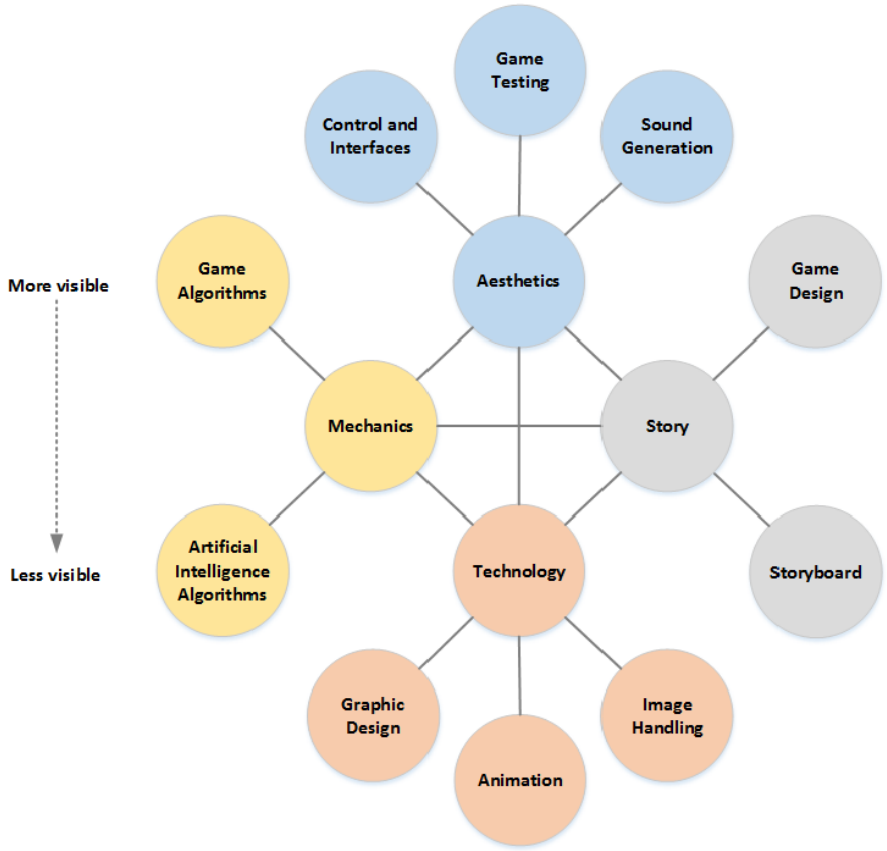
Cada resultado conserva su denominador. La encuesta informa percepción; la telemetría informa eventos; la compilación informa corrección sintáctica y de dependencias; la presencia de archivos válidos informa integridad del registro. Ninguna fuente, por sí sola, demuestra todas las características del estándar.

Modelo Tetrad Modificado

El Modelo de la Tétrada Elemental Modificado (Elemental Tetrad Game Design Model Modified - ETGDMM) es un marco de trabajo estructural utilizado en la Ingeniería de Software y el Diseño de Videojuegos (especialmente en Serious Games) para guiar y armonizar todos los componentes del sistema durante la fase de diseño.

Figura 2

Modelo de la Tétrada Elemental Modificado (ETGDMM)



Nota. La figura organiza los nodos Estética, Historia, Tecnología y Mecánicas junto con control e interfaces, pruebas, sonido, diseño del juego, storyboard, diseño gráfico, animación, manejo de imágenes, algoritmos del juego y algoritmos de inteligencia artificial.

Este modelo organiza los elementos del juego en cuatro nodos centrales interconectados, divididos según su visibilidad ante el usuario (desde los más visibles en la superficie hasta los menos visibles en la lógica interna del sistema):

Estética (Aesthetics)

Este nodo maestro define la identidad perceptiva del videojuego. Además, coordina la forma en que el jugador experimenta el entorno visual, auditivo y operativo, transformando las reglas abstractas en una experiencia sensorial fluida, accesible, validada y altamente interactiva.

Control e Interfaces (Control and Interfaces)

Se encarga de estructurar la interacción directa entre el usuario y el sistema a través de menús, paneles informativos y controles específicos:

- Registro Inicial y Menú Principal: Al ingresar por primera vez, el sistema despliega un formulario de login obligatorio que solicita Nombre y Edad (validada entre 5 y 30 años). Una vez registrado, se accede al Menú Principal que centraliza los botones de Jugar, Seleccionar Nivel (para escenarios ya desbloqueados), Tabla de Mejores Puntajes (Leaderboard), Opciones, Salir y un Tutorial instructivo.
- HUD en Pantalla (Heads-Up Display): Interfaz activa durante la partida que muestra en tiempo real las métricas clave para el jugador: el tiempo transcurrido, la cantidad de banderas/cofres recogidos y el puntaje acumulado.
- Pantallas de Estado (Pausa y Game Over): El menú de Pausa interrumpe el juego permitiendo Reanudar, Ajustar Configuración o Volver al Menú. La pantalla de Game Over se activa al caer al vacío o ser alcanzado por el NPC, mostrando el conteo de intentos, cofres recolectados y las opciones de reinicio.
- Panel de Opciones: Incluye interruptores analógicos o digitales para el encendido/apagado del sonido (Toggle ON/OFF) y un botón crítico de “Resetear” diseñado para borrar por completo el progreso guardado y devolver al usuario al Nivel 1.

Pruebas del Juego (Game Testing)

Dedicado al control de calidad, calibración y verificación del comportamiento del software bajo la experiencia del usuario:

• Validación de Progresión: Evalúa que el flujo secuencial y lineal del juego funcione correctamente (comprobar que los niveles superiores permanezcan bloqueados hasta superar los anteriores).

• Calibración de Dificultad Adaptativa: Testea la relación de velocidad entre el jugador y el NPC enemigo para garantizar un reto justo pero desafiante.

• Efectividad del Tutorial: Verifica de forma práctica que el Tutorial Básico introductorio prepare adecuadamente al usuario en las mecánicas de clic y control lateralizado antes de someterlo a los niveles reales.

Generación de Sonido (Sound Generation)

Sostiene la atmósfera del videojuego y aporta retroalimentación auditiva a las acciones del jugador:

• Efectos de Sonido (SFX): Implementa alertas sonoras de éxito al recolectar cada uno de los 18 cofres/monedas, así como frecuencias específicas para los eventos de derrota (Game Over).

• Música Ambiental: Provee una banda sonora de fondo (BGM) que acompaña el ritmo del juego de forma continua.

• Control de Audio: Integrado directamente con el menú de opciones a través de un switch que permite silenciar o activar el entorno sonoro a preferencia del usuario.

Historia / Diseño Narrativo (Story)

Este nodo maestro articula el propósito de la experiencia del juego, estableciendo el viaje del usuario a través de una progresión estructurada. Define el marco contextual, las reglas de avance lineal y la lógica de progresión que justifican y motivan cada partida.

Diseño del Juego (Game Design)

Define las bases teóricas, mecánicas pedagógicas y el sistema de evaluación que rigen la experiencia del juego:

• **Concepto Pedagógico y Estimulación Cognitiva:** El juego no posee una narrativa fantástica tradicional, sino una justificación funcional centrada en la estimulación cerebral. Está diseñado para activar de forma independiente los hemisferios cerebrales mediante un esquema de control lateralizado (el clic izquierdo acciona un movimiento hacia la izquierda/arriba activando un lóbulo, mientras que el clic derecho acciona el movimiento opuesto activando el otro).

• **Estructura de Progreso (Secuencialidad Estricta):** El diseño establece un avance lineal rígido e inalterable. El jugador se enfrenta a una curva de aprendizaje progresiva y obligatoria donde no se permite omitir escenarios; para acceder a un nivel superior (por ejemplo, del Nivel 1 al Nivel 2, o del Nivel 2 al Nivel 3), se debe haber superado con éxito el mapa predecesor.

• **Sistema de Calificación por Estrellas:** Cada nivel contiene hasta 18 cofres, con un total detectado dinámicamente por escena distribuidos estratégicamente en la pista de zigzag. El desempeño y éxito del jugador se miden cuantitativamente al final del nivel bajo la siguiente escala de recompensas:

• **Tres Estrellas:** Recolección perfecta de los 18 cofres/monedas.

• **Dos Estrellas:** Recolección alta de 12 a 17 cofres/monedas.

• **Una Estrella:** Recolección mínima aceptable de 6 a 11 cofres/monedas.

• **Cero Estrellas (Fallo):** Recolección deficiente (menos de 6 cofres), lo que impide considerar el nivel como superado.

Guión Gráfico / Progresión de Flujo (Storyboard)

Mapea de forma secuencial los estados de la experiencia por los que transita el jugador, dividiéndose en fases lógicas de interacción:

• **Flujo Secuencial de Pantallas:** Modela el recorrido visual y operativo del usuario desde el arranque de la aplicación: Pantalla de Registro (Login) → Tutorial Práctico de Movimiento → Menú Principal → Ejecución del Nivel Activo (1, 2, 3 o superiores) → Pantallas de

Transición (Despliegue de Estrellas obtenidas en caso de victoria, o Pantalla de Game Over en caso de caída o captura).

• **Evolución del Entorno Narrativo/Visual:** Establece el cambio de ritmo en el guión del juego a través de los escenarios. El storyboard contempla que los primeros niveles sirvan para dominar el zigzag básico y la presión de la IA, mientras que los últimos 3 niveles introducen un giro visual y mecánico fundamental al incorporar obstáculos físicos alrededor de las plataformas, alterando la estética del entorno y obligando al jugador a adaptar su ritmo y añadir la acción de salto a su repertorio.

Tecnología (Technology)

Este nodo maestro representa el motor de soporte técnico y la infraestructura informática que materializa el videojuego. Engloba las herramientas de renderizado visual, el procesamiento de dinámicas de movimiento y la gestión de bases de datos persistentes que sustentan el comportamiento de la simulación en tiempo real.

Diseño Gráfico (Graphic Design)

Se ocupa del modelado y la construcción estética tridimensional del entorno interactivo, en caso sobre la plataforma Unity:

• **Entorno y Estilo Visual:** Estilo gráfico tridimensional (3D) que da forma a una serie de plataformas suspendidas estructuradas geométricamente en un patrón de Zig-Zag.

• **Modelado de Personajes (TPS):** Modelado en perspectiva de tercera persona para las dos entidades móviles del juego: el avatar del usuario (Gamer) y el enemigo controlado por el ordenador (NPC).

• **Assets e Interactivos:** Diseño gráfico de los elementos clave del mapa, como los 18 cofres distribuidos estratégicamente, las banderas de progreso y los obstáculos físicos específicos añadidos alrededor de las plataformas en los últimos 3 niveles.

Animación (Animation)

Define los ciclos de movimiento físico y los estados de transición que dan vida a las entidades del juego:

- **Ciclos de Movimiento Locomotor:** Animaciones en tercera persona para las acciones básicas de desplazamiento (caminar, correr y girar) tanto del Gamer como del NPC perseguidor.
- **Mecánica de Salto:** Desarrollo de animaciones de salto fluido para el jugador, diseñadas exclusivamente para superar con éxito los obstáculos implementados en los últimos 3 niveles del mapa.
- **Feedback Visual y de Entorno:** Animaciones de respuesta, como la apertura o recolección de los cofres al entrar en contacto con el usuario, y la fluidez en las transiciones de las interfaces y menús interactivos.

Manejo de Imágenes y de Datos (Image Handling & Data Management)

Encargado del procesamiento de texturas, elementos gráficos en pantalla y la persistencia técnica de la información del juego:

- **Manejo de Imágenes de Interfaz:** Carga y optimización de sprites y elementos visuales para el HUD en pantalla (visualización de las estrellas ganadas, banderas e iconos decorativos de los menús).
- **Persistencia mediante Archivos Planos JSON:** Arquitectura técnica de almacenamiento de datos. El sistema gestiona un motor de escritura de archivos JSON para guardar de forma local la telemetría detallada del jugador. Cada registro almacena estrictamente las siguientes variables técnicas:
- **Identificación básica:** Nombre del jugador y Edad (validada de 5 a 30 años).
- **Métricas de rendimiento:** Nivel jugado, tiempo empleado en completarlo, número de intentos, monedas/cofres recolectados y total de clics efectuados.

• **Métricas analítico-cognitivas:** Porcentaje de acciones correctas respecto al total de clics registrados, tiempo exacto en realizar el primer clic, tiempo promedio de supervivencia tras reaparecer, y número de muertes provocadas directamente por el enemigo.

• **Temporalidad:** Fecha y hora exacta del registro guardado.

Mecánicas (Mechanics)

Este nodo maestro define el motor de reglas lógicas, las restricciones, las interacciones y los sistemas matemáticos que rigen el mundo virtual. Traduce los comandos del jugador en respuestas físicas y controla las dinámicas de juego, coordinando tanto las acciones del avatar como las conductas autónomas de los enemigos.

Algoritmos del Juego (Game Algorithms)

Representa la infraestructura lógica que procesa los comandos físicos de entrada y calcula de manera constante los estados del juego:

• **Algoritmo de Control Lateralizado y Estímulo Hemisférico:** Mapea directamente las entradas (inputs) del ratón para guiar el movimiento del TPS.

• **Clic Izquierdo:** Ejecuta un vector de movimiento exclusivo hacia la Izquierda o Arriba.

• **Clic Derecho:** Ejecuta un vector de movimiento exclusivo hacia la Derecha o Abajo. Este algoritmo fuerza de forma alterna la toma de decisiones motrices para estimular los lóbulos cerebrales correspondientes de los niños.

• **Algoritmo de Detección de Colisiones y Triggers:** Administra el procesamiento de eventos físicos en tiempo real:

• **Recolección:** Al detectar la colisión entre el Gamer y un cofre, se activa un disparador (trigger) que destruye el objeto, suma el puntaje y actualiza el HUD.

• **Obstáculos y Salto:** En los últimos 3 niveles hay algoritmos específicos que evalúan si el jugador activa a tiempo la mecánica de salto al aproximarse a las colisiones de los obstáculos periféricos.

• **Algoritmo de Caída Libre y Reseteo:** Monitorea las coordenadas de altura (Y) del avatar del jugador. Si el usuario realiza un clic erróneo o pierde el rumbo de la plataforma de Zig-Zag, el algoritmo detecta la pérdida de sustentación, activa inmediatamente la pantalla de Game Over por caída libre y habilita las opciones lógicas de reiniciar el nivel actual o volver al menú.

• **Algoritmo de Secuencialidad y Desbloqueo:** Controla el árbol de niveles. Mantiene un estado booleano de bloqueo en los niveles superiores y solo permite cambiar el bit a "desbloqueado" una vez que el algoritmo de recolección valida que el nivel anterior fue superado con éxito.

Algoritmos de Inteligencia Artificial (Artificial Intelligence Algorithms)

Gobierna el comportamiento del NPC enemigo a través de reglas autónomas de persecución, asedio y dificultad dinámica:

• **Algoritmos de Dirección y Persecución (Pursuit Steering Behaviors):** El NPC (también en tercera persona) no se mueve de forma aleatoria; utiliza algoritmos de persecución predictiva. Calcula de forma matemática el vector de posición futura y la velocidad actual del jugador (Gamer) para trazar una ruta óptima de intercepción. El objetivo de este algoritmo es estorbar, hostigar o seguir de cerca al usuario a lo largo de la pista en Zig-Zag.

• **Algoritmo de Ajuste y Dificultad por Velocidad Relativa:** Controla la tasa de aproximación del enemigo. La IA está programada con una velocidad adecuada que actúa como un temporizador biológico. Si el algoritmo detecta que el usuario disminuye su ritmo de juego o se detiene, la distancia relativa se acorta drásticamente.

• **Algoritmo de Condición de Captura:** Si las cajas de colisión del NPC y del jugador se intersecan, la IA activa el estado de Game Over, detiene la simulación y registra en memoria la muerte influenciada por el enemigo. Este valor se incorpora a PlayerMetricsData cuando se ejecuta el guardado de métricas implementado.

Aplicación en la Fase de Diseño del Sistema con XP

Durante la fase de diseño, dentro de metodología ágil XP (Extreme Programming), el Modelo de Diseño de Juegos de la Tétrada Elemental Modificado (Elemental Tetrad Game Design Model Modified - ETGDMM) actúa como una metáfora del sistema. Permite al equipo de ingeniería y diseño desglosar las historias de usuario en tareas técnicas claras, asegurando que un cambio en un nodo (como añadir un obstáculo en Story) se traduzca de forma coherente en los algoritmos correspondientes (Mechanics), su representación visual (Aesthetics) y su soporte de interfaz (Technology).

Aplicación del Modelo Tetrad Modificado en el videojuego

El modelo se materializó como una red de decisiones interdependientes. La Estética integra la experiencia visible y audible: registro, menú, tutorial, HUD, pausa, resumen, controles, pruebas de interacción, música y efectos. La Historia articula una progresión funcional, sin necesidad de una trama fantástica extensa: el usuario aprende, supera niveles, obtiene estrellas y avanza. La Tecnología sostiene escenas, física, renderizado, animación, datos y plataformas. Las Mecánicas contienen reglas de movimiento, colisión, recolección, progreso y persecución.

La implementación final conserva la intención de los controles lateralizados como alternancia de decisiones, pero no atribuye activación independiente de hemisferios. El registro de edad admite el rango operativo configurado en la interfaz, mientras el público educativo del trabajo permanece entre 7 y 12 años. Las estrellas se calculan de forma proporcional a los recolectables detectados por escena. Completar la meta permite avanzar aun cuando la cantidad recolectada sea baja; las estrellas expresan rendimiento y no constituyen una barrera adicional de progreso.

La dificultad se materializa mediante configuración predefinida de escenas, velocidad, trazado, obstáculos y presión del enemigo. El agente no ajusta automáticamente la dificultad según cada usuario. Las plataformas de salto se verifican en los niveles 15 y 16, y el sistema

de salto puede coexistir con obstáculos en la fase avanzada. La persistencia combina PlayerPrefs, JSON y CSV, manteniendo los datos en el dispositivo.

El agente usa Pursuit cuando la velocidad, la distancia y la seguridad permiten una predicción útil. Seek actúa como comportamiento base y como retorno seguro. La ruta del jugador, la validación de suelo, el muestreo del segmento, la altura, el salto y el respawn amplían el nodo Mecánicas. Así, la aplicación mantiene la estructura conceptual del tutor y expresa cómo cada componente se concretó en el software.

La correspondencia aplicada de los nodos se resume en la tabla siguiente.

Tabla 5

Correspondencia entre nodos Tetrad y componentes del videojuego

Nodo	Componentes del modelo	Materialización en el producto
Estética	Control e interfaces	Registro, tutorial, menú, HUD, pausa y resumen
Estética	Pruebas del juego	Pruebas funcionales, encuesta y telemetría
Estética	Generación de sonido	Música, efectos y control de audio
Historia	Diseño del juego	Propósito, progresión, estrellas y niveles
Historia	Storyboard	Registro → tutorial → menú → nivel → resultado
Tecnología	Diseño gráfico	Escenarios 3D, plataformas, avatar y objetos
Tecnología	Animación	Movimiento, giro, salto y transiciones
Tecnología	Imágenes y datos	HUD, iconos, JSON, CSV y PlayerPrefs
Mecánicas	Algoritmos del juego	Entradas, física, colisiones, recolección y progreso
Mecánicas	Algoritmos de IA	Seek, Pursuit, ruta segura, salto y respawn

Nota. Los componentes se presentan como relaciones de diseño, no como módulos aislados.

Capítulo III

Metodología y desarrollo de la solución

Enfoque de investigación

El trabajo empleó un enfoque mixto de predominio cuantitativo. El componente cuantitativo integró telemetría automática, respuestas ordinales, frecuencias, medidas descriptivas e indicadores técnicos. El componente cualitativo examinó tres preguntas abiertas sobre preferencias, dificultades y cambios propuestos. La integración se realizó durante la interpretación: los comentarios explican patrones observados en ítems y métricas, estrategia coherente con la integración interpretativa de métodos mixtos (Creswell & Plano Clark, 2018).

La investigación fue aplicada porque se produjo una solución funcional; tecnológica porque integró componentes de software; y evaluativa porque contrastó funcionamiento, percepción y características de calidad. El alcance fue descriptivo y comparativo. Los participantes estuvieron expuestos a configuraciones predefinidas por nivel en condiciones naturales de uso, con medidas repetidas incompletas y cobertura desigual. Este diseño permite caracterizar diferencias observadas, pero no estimar causalidad ni cambio cognitivo.

Tipo y diseño de investigación

El diseño fue transversal respecto de la encuesta, aplicada después de la experiencia, y longitudinal corto respecto de la telemetría, porque algunos participantes generaron varios registros durante su recorrido por niveles. La unidad de análisis de la encuesta fue el participante. La unidad de análisis de la telemetría fue el registro participante–nivel.

Las medias de registros ponderan más a quienes jugaron más niveles. Para controlar esa característica, se calcularon también resúmenes por participante. Los resultados por nivel se presentan con su n específico. No se compararon variantes controladas de IA porque el modo de steering no se almacenó en los archivos y los participantes no fueron asignados a condiciones experimentales.

Tabla 6

Diseño metodológico aplicado

Componente	Definición	Aplicación
Enfoque	Mixto con predominio cuantitativo	Telemetría, Likert y respuestas abiertas
Finalidad	Aplicada y tecnológica	Desarrollo de un videojuego funcional
Alcance	Descriptivo y comparativo	Grupos y niveles sin inferencia causal
Encuesta	Corte transversal	Una respuesta posterior por participante
Telemetría	Medidas repetidas incompletas	Varios niveles por participante
Calidad	Evaluación de producto	Pruebas e indicadores ISO/IEC 25010

Nota. El diseño se define por el procedimiento ejecutado y la estructura de los datos. Las comparaciones describen el desempeño observado y no equivalen a un experimento con asignación de condiciones.

Población, población accesible y muestra

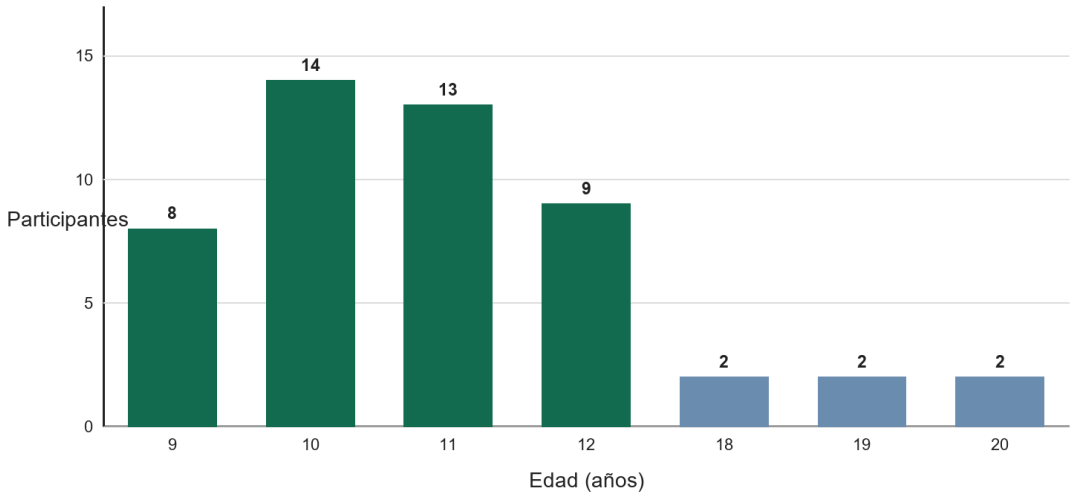
La población objetivo estuvo constituida por niños de 7 a 12 años, rango definido en la nota conceptual y en el objetivo general. La población accesible correspondió a estudiantes disponibles en la institución educativa donde se realizaron las sesiones, con acceso a Windows o Android y posibilidad de completar la interacción guiada.

La muestra efectiva estuvo integrada por 50 participantes. La población infantil incluyó 44 niños y seis participantes entre 18 a 20 años, conformaron un grupo técnico complementario de participantes adultos. Sus resultados se analizaron por separado y no se utilizaron como contrafactual de la población infantil.

El muestreo fue no probabilístico por accesibilidad. Los criterios de inclusión fueron completar la experiencia, disponer de una respuesta de encuesta y contar con identificación pseudonimizada compatible con los registros. El análisis no publica nombres, contactos ni respuestas que permitan identificar a menores.

Figura 3

Distribución de los participantes efectivos



Nota. Frecuencias de las 50 respuestas utilizadas: 44 participantes de la población infantil y 6 del grupo técnico adulto. Los datos provienen del campo edad de la encuesta y se presentan sin identificadores personales.

Tabla 7

Distribución de participantes por grupo y plataforma

Grupo	n	Windows	Android	Uso analítico
Población infantil	44	29	15	Resultados principales
Grupo técnico adulto	6	4	2	Comprobación técnica complementaria
Total	50	33	17	Cobertura de la evaluación

Nota. La plataforma se obtuvo de la encuesta. El grupo técnico adulto se conserva separado en todos los análisis y no constituye un grupo equivalente ni contrafactual.

Técnicas e instrumentos

Observación y telemetría

La observación se apoyó en los eventos que el videojuego registra durante la partida. Cada archivo JSON contiene participante, edad, nivel, tiempo, intentos, muertes, recolectables, clics, porcentaje de entradas aceptadas, demora del primer clic, intervalo medio entre respawn y muerte, muertes influidas por el enemigo y marca temporal. La persistencia CSV resume variables de sesión.

La telemetría reduce el sesgo de recuerdo porque registra eventos en el momento de uso. No obstante, refleja exclusivamente aquello que fue programado. El archivo no incluye una variable explícita de resultado final ni el modo Seek/Pursuit activo, y conserva un único archivo por combinación participante–nivel. Estas condiciones se consideran al interpretar cobertura y dificultad.

Encuesta de experiencia

La encuesta fue revisada por el tutor en claridad, pertinencia y correspondencia con los objetivos. Contiene 15 afirmaciones con escala de 1 a 5: 1, Muy en desacuerdo; 2, En desacuerdo; 3, No estoy seguro/a; 4, De acuerdo; y 5, Muy de acuerdo. Todos los ítems tienen orientación positiva, por lo que una puntuación alta representa una valoración favorable del aspecto preguntado.

Los ítems se analizaron individualmente y se agruparon por contenido en cuatro dimensiones: usabilidad e interacción; experiencia y motivación; reto y demanda percibida; y retroalimentación del desempeño. La agrupación es teórica y se usa para síntesis interna; no se presenta como validación factorial o psicométrica completa. Las tres preguntas abiertas indagan qué gustó, qué resultó difícil y qué cambiaría el participante.

Tabla 8

Correspondencia de los ítems con las dimensiones analíticas

Ítem	Contenido	Dimensión	Polaridad
1	Comprensión de la tarea	Usabilidad e interacción	Positiva
2	Aprendizaje de controles	Usabilidad e interacción	Positiva
3	Respuesta de controles	Usabilidad e interacción	Positiva
4	Claridad visual	Usabilidad e interacción	Positiva
5	Utilidad del tutorial	Usabilidad e interacción	Positiva
6	Diversión	Experiencia y motivación	Positiva
7	Deseo de continuar	Experiencia y motivación	Positiva
8	Dificultad adecuada	Reto y demanda percibida	Positiva

Ítem	Contenido	Dimensión	Polaridad
9	Concentración requerida	Reto y demanda percibida	Positiva
10	Decisión rápida	Reto y demanda percibida	Positiva
11	Aporte del enemigo al reto	Reto y demanda percibida	Positiva
12	Reconocimiento de objetos	Usabilidad e interacción	Positiva
13	Comprensión del resultado	Retroalimentación	Positiva
14	Utilidad del resumen	Retroalimentación	Positiva
15	Intención de volver a jugar	Experiencia y motivación	Positiva

Nota. La asignación se basa en el contenido literal de los 15 ítems. No se invirtieron respuestas porque todas las afirmaciones están redactadas en sentido favorable.

Consistencia interna del instrumento

La consistencia se estimó con alfa de Cronbach por dimensión. Para k ítems, el coeficiente compara la suma de varianzas de los ítems con la varianza de la puntuación total. Un valor alto indica covariación interna, pero no demuestra unidimensionalidad, validez de constructo ni estabilidad temporal (Cronbach, 1951; Tavakol & Dennick, 2011). La interpretación conjunta de contenido, estructura y respuesta sigue las cautelas para desarrollo y evaluación de escalas descritas por Boateng et al. (2018).

El coeficiente se calculó para la muestra total y la población infantil. Los valores del grupo técnico adulto no se utilizaron como evidencia de confiabilidad debido a su tamaño reducido. Para la dimensión de dos ítems, alfa depende directamente de la correlación entre ambos y se interpreta con especial cautela.

Procedimiento

El procedimiento se organizó en cinco momentos. Primero, se preparó el producto y se verificó el acceso a las plataformas. Segundo, se registró a cada participante con un código y edad. Tercero, se presentó el tutorial y se inició el recorrido por niveles. Cuarto, el sistema almacenó telemetría local durante la interacción. Quinto, se aplicó la encuesta posterior y se consolidaron los datos para el análisis.

La cobertura de niveles dependió del recorrido realizado por cada participante, por lo que no todos aportaron el mismo número de registros. La encuesta sí aporta una respuesta completa por participante efectivo.

Tabla 9

Secuencia del procedimiento de evaluación

Etapa	Actividad	Registro	Control
Preparación	Comprobar ejecución y dispositivos	Plataforma	Versión funcional
Ingreso	Registrar código y edad	Perfil local	Pseudonimización
Aprendizaje	Completar tutorial	Estado de progreso	Instrucciones breves
Interacción	Jugar niveles disponibles	JSON por nivel	Registro automático
Percepción	Responder 15 ítems y 3 preguntas	Encuesta	Respuesta completa
Integración	Unir por código	Base analítica	Separación por grupo

Nota. La secuencia corresponde a la aplicación ejecutada. La telemetría y la encuesta se vincularon mediante el código pseudonimizado del participante.

Consideraciones éticas y protección de datos

La evaluación se realizó con finalidad académica y coordinación institucional. En el análisis, los nombres fueron sustituidos por códigos. El documento presenta agregados y no incorpora contactos, firmas de menores ni respuestas textuales con información identificable. El grupo de adultos se reporta por su función técnica y no como equivalente de los niños.

El videojuego almacena datos localmente. Esta decisión reduce transferencia por red, pero requiere administrar el dispositivo y el acceso a los archivos. El botón de restablecimiento elimina identidad, progreso, estrellas y métricas locales. La documentación distingue el rango operativo del formulario del público educativo y recomienda conservar únicamente la información necesaria para cada sesión.

Estrategia de análisis de datos

Se verificó la lectura de los archivos, la presencia de campos, rangos, duplicados por participante–nivel y coherencia básica. Para telemetría se calcularon n, media, desviación

estándar, mediana, cuartiles, mínimo y máximo. Los análisis por nivel conservaron su denominador. También se promedió primero dentro de cada participante para comprobar que los resultados globales no dependieran exclusivamente del número desigual de niveles jugados.

Para la encuesta se calcularon medias, desviaciones estándar y porcentaje favorable —respuestas 4 o 5— por ítem. Las dimensiones se obtuvieron como media de sus ítems y se estimó alfa de Cronbach. Las respuestas abiertas se codificaron temáticamente con categorías no excluyentes; una respuesta podía aportar a más de una categoría.

No se realizaron pruebas de hipótesis entre niños y adultos porque la función del grupo técnico, su tamaño y su composición no sostienen una comparación inferencial. Tampoco se modeló dificultad causal por nivel debido a la cobertura desigual y a la falta de asignación. El análisis se orientó a describir funcionamiento, experiencia y puntos de mejora.

Proceso iterativo de desarrollo con Scrum y XP

El Product Backlog se priorizó por valor de uso y dependencia técnica. Registro, movimiento y escenas constituyeron la base; tutorial y progreso organizaron la experiencia; estrellas, persistencia y telemetría agregaron retroalimentación; Android amplió el contexto; y la IA explícita Seek/Pursuit completó el comportamiento autónomo.

XP se aplicó mediante integración frecuente, correcciones pequeñas, reutilización de prefabs y prueba continua del incremento. Las correcciones de cámara, niveles, métricas y persecución se incorporaron sin separar el diseño de la implementación. La tabla siguiente presenta los incrementos principales.

Tabla 10

Incrementos principales del desarrollo

Incremento	Contenido integrado	Criterio de cierre
I1 Base jugable	Proyecto, jugador, nombre, edad y nivel inicial	Escena ejecutable y control básico

Incremento	Contenido integrado	Criterio de cierre
I2 Aprendizaje	Tutorial, secuencia, niveles y cámara	Recorrido comprensible
I3 Progreso	Estrellas, puntaje, niveles 10–16	Progreso persistente
I4 Datos y móvil	Métricas, personajes, Android y reinicio	Registro local y control móvil
I5 IA	Seek, Pursuit y seguridad de navegación	Persecución explícita y estable

Nota. Cada incremento reúne funcionalidades relacionadas y un criterio observable de término.

Criterios de cierre y control de cada incremento

El cierre de un incremento exigió una versión ejecutable y un recorrido verificable, no únicamente la existencia de componentes. Para la base jugable se comprobó carga de escena, registro, movimiento y recuperación. Para aprendizaje se revisó la continuidad entre tutorial, menú y primer nivel. Para progreso se comprobó desbloqueo, puntuación y persistencia. Para datos y móvil se verificó creación de registros, controles táctiles y restablecimiento. Para IA se recorrieron decisiones Seek/Pursuit, seguridad del objetivo y recuperación.

La integración frecuente permitió localizar efectos entre nodos del Tetrad. Un cambio de movimiento podía modificar animación y sensación de control; un cambio de escena podía afectar progresión y persistencia; un cambio de persecución podía alterar reto y tiempo. El criterio de cierre incluyó, por tanto, la función principal y sus relaciones inmediatas. Este enfoque es consistente con incrementos pequeños de Scrum y prácticas técnicas de XP (Beck & Andres, 2004; Schwaber & Sutherland, 2020).

Las incidencias detectadas se resolvieron en el incremento correspondiente o se registraron como limitación. La compilación correcta confirmó que los componentes y referencias de código podían construirse, mientras las pruebas de flujo examinaron operación visible. La compilación no sustituyó PlayMode ni la sesión con participantes; cada evidencia respondió a una pregunta distinta y se conserva así en los resultados.

La definición de término también protegió la trazabilidad. Ninguna historia se consideró cubierta solo por su descripción: debía relacionarse con requisito, componente y caso de

prueba. Esta práctica facilita mantenimiento, porque una modificación futura permite identificar qué flujo y qué dato deben comprobarse nuevamente.

Especificación de requisitos funcionales

Los requisitos funcionales describen servicios observables del videojuego. Se priorizaron con MoSCoW: Must para el flujo esencial, Should para funciones necesarias de soporte y Could para complementos. Los criterios de aceptación se redactaron como resultados visibles o datos persistidos.

Para mantener legibilidad, los requisitos se dividen por dominio. Las tres tablas siguientes contienen registro y flujo, mecánicas e inteligencia artificial, y datos y plataformas.

Tabla 11

Requisitos funcionales de registro y flujo

ID	Prioridad	Requisito	Criterio de aceptación
RF-01	Must	Registrar un nombre local	El perfil conserva un nombre no vacío
RF-02	Must	Registrar y validar la edad	El sistema acepta solo el intervalo operativo
RF-03	Must	Presentar tutorial inicial	El primer acceso conduce al tutorial
RF-04	Must	Mostrar menú principal	Se ofrecen jugar, niveles, opciones y salir
RF-05	Must	Desbloquear niveles secuencialmente	Solo se habilita progreso alcanzado
RF-06	Should	Seleccionar niveles habilitados	La selección carga la escena válida
RF-07	Should	Pausar y reanudar	El estado detiene y restaura la partida
RF-08	Should	Configurar audio	El estado se conserva localmente
RF-09	Should	Restablecer datos	Se eliminan perfil, progreso y métricas
RF-10	Could	Mostrar clasificación local	Se ordenan resultados disponibles

Nota. La tabla concentra RF-01 a RF-10: registro, tutorial, menú, progreso y servicios de apoyo; cada fila indica prioridad y resultado verificable.

Tabla 12

Requisitos funcionales de mecánicas e inteligencia artificial

ID	Prioridad	Requisito	Criterio de aceptación
RF-11	Must	Mover al jugador por entradas lateralizadas	Cada entrada válida cambia dirección
RF-12	Must	Aceptar controles táctiles y de escritorio	Las acciones esenciales operan en ambas plataformas

ID	Prioridad	Requisito	Criterio de aceptación
RF-13	Must	Ejecutar salto	El impulso se aplica con condición de suelo
RF-14	Must	Recolectar objetos	El objeto se desactiva y el contador aumenta
RF-15	Must	Detectar caída u obstáculo	Se registra muerte y se ofrece reintento
RF-16	Must	Detectar contacto con el enemigo	Se activa derrota una sola vez
RF-17	Must	Detectar meta	Se guarda el nivel y se muestra el resultado
RF-18	Should	Calcular estrellas	Se usa proporción de recolectables
RF-19	Should	Mostrar resumen final	Se presentan indicadores de la partida
RF-20	Should	Reintentar nivel	Se restablecen estados necesarios
RF-21	Must	Aplicar Seek	El agente busca un objetivo actual o seguro
RF-22	Must	Aplicar Pursuit	El agente predice una posición futura acotada
RF-23	Must	Alternar Seek/Pursuit con histéresis	El modo no oscila por cambios mínimos
RF-24	Must	Validar suelo y segmento	Se descartan objetivos no transitables
RF-25	Should	Seguir trayectoria del jugador	El agente respeta esquinas y plataformas
RF-26	Should	Saltar de forma autónoma	Se usa un aterrizaje válido
RF-27	Should	Recuperar al agente tras caída	El respawn conserva la persecución

Nota. La tabla reúne RF-11 a RF-27: entradas, física, resultado de nivel y navegación Seek/Pursuit; los criterios describen la conducta observable.

Tabla 13

Requisitos funcionales de datos y plataformas

ID	Prioridad	Requisito	Criterio de aceptación
RF-28	Must	Guardar métricas JSON por nivel	Se crea un registro con campos definidos
RF-29	Should	Guardar resumen CSV de sesión	Se agrega una fila al cierre
RF-30	Must	Ejecutar en Windows y Android	El flujo esencial opera en ambas plataformas

Nota. La tabla resume RF-28 a RF-30: persistencia JSON/CSV y operación en Windows y Android; la aceptación exige creación del registro y flujo funcional.

Requisitos no funcionales

Tabla 14

Requisitos no funcionales del producto

ID	Característica	Requisito	Indicador
RNF-01	Adecuación funcional	Las escenas incluidas deben ejecutar el flujo definido	Compilación y cobertura funcional
RNF-02	Eficiencia	La física debe actualizarse en paso fijo	Movimiento mediante FixedUpdate
RNF-03	Interacción	El tutorial debe explicar acciones esenciales	Ítems 1, 2 y 5 de encuesta

ID	Característica	Requisito	Indicador
RNF-04	Interacción	El HUD debe ser legible durante la partida	Ítems 4, 12, 13 y 14
RNF-05	Fiabilidad	Una derrota no debe registrarse dos veces	Guardas de estado
RNF-06	Fiabilidad	Los datos del nivel deben persistir una vez	Control de guardado
RNF-07	Compatibilidad	Entradas de escritorio y táctiles deben coexistir	Uso en Windows y Android
RNF-08	Mantenibilidad	Las responsabilidades deben distribuirse por componentes	Arquitectura y componentes
RNF-09	Mantenibilidad	Los parámetros de IA deben ser configurables	Campos serializados
RNF-10	Flexibilidad	Las escenas deben compartirse entre plataformas	Mismo conjunto de niveles
RNF-11	Seguridad	Los datos deben permanecer locales	Persistencia sin servicios de red
RNF-12	Integridad	Los campos obligatorios no deben quedar vacíos	Validación de 248 JSON
RNF-13	Trazabilidad	Cada registro debe incluir participante, nivel y fecha	Campos de identificación
RNF-14	Recuperación	El usuario debe poder reintentar o volver al menú	Flujos de resultado y pausa

Nota. Las características se alinean con ISO/IEC 25010:2023. Los indicadores identifican evidencia disponible y no representan certificación del estándar.

Historias de usuario y Product Backlog

Las historias expresan valor desde la perspectiva de participante, docente o equipo técnico. Se redactaron con la forma «Como…, quiero…, para…» y se vincularon con requisitos. Las historias se agruparon en dos tablas para mantener legibilidad.

Tabla 15

Historias de usuario de interacción y gameplay

ID	Rol	Necesidad	Beneficio	Vínculo
HU-01	Participante	Registrar nombre y edad	Conservar identidad de sesión	RF-01–02
HU-02	Participante	Completar un tutorial	Aprender los controles	RF-03
HU-03	Participante	Entrar al menú	Elegir una acción	RF-04
HU-04	Participante	Ver niveles habilitados	Conocer mi progreso	RF-05–06
HU-05	Participante	Pausar la partida	Interrumpirla de forma segura	RF-07
HU-06	Participante	Activar o silenciar audio	Adaptar la experiencia	RF-08
HU-07	Participante	Restablecer datos	Comenzar de nuevo	RF-09

ID	Rol	Necesidad	Beneficio	Vínculo
HU-08	Participante	Moverme con pocas acciones	Concentrarme en la ruta	RF-11–12
HU-09	Participante	Saltar obstáculos	Superar niveles avanzados	RF-13
HU-10	Participante	Recolectar objetos	Mejorar mi resultado	RF-14
HU-11	Participante	Entender una derrota	Reintentar con otra estrategia	RF-15–16
HU-12	Participante	Ver estrellas y resumen	Conocer mi desempeño	RF-17–19

Nota. Las historias HU-01 a HU-12 conectan registro, aprendizaje de controles, progresión y retroalimentación con RF-01 a RF-19.

Tabla 16

Historias de usuario de IA, datos y mantenimiento

ID	Rol	Necesidad	Beneficio	Vínculo
HU-13	Participante	Ser perseguido por un agente	Mantener el reto	RF-21–23
HU-14	Participante	Observar un perseguidor estable	Percibir reglas justas	RF-24–27
HU-15	Docente	Usar el juego sin conexión	Facilitar la aplicación local	RNF-11
HU-16	Docente	Revisar resultados	Comprender la interacción	RF-28–29
HU-17	Equipo técnico	Guardar métricas por nivel	Analizar funcionamiento	RF-28
HU-18	Equipo técnico	Configurar parámetros de IA	Calibrar la persecución	RNF-09
HU-19	Equipo técnico	Reutilizar escenas en dos plataformas	Reducir duplicación	RF-30
HU-20	Equipo técnico	Separar componentes	Facilitar mantenimiento	RNF-08
HU-21	Equipo técnico	Validar archivos	Evitar registros incompletos	RNF-12–13
HU-22	Participante	Consultar clasificación local	Comparar resultados	RF-10

Nota. Las historias HU-13 a HU-22 relacionan persecución segura, telemetría, configurabilidad, plataformas y mantenimiento con los requisitos técnicos.

Trazabilidad entre objetivos, requisitos y pruebas

La trazabilidad integrada se presenta en la tabla siguiente.

Tabla 17

Trazabilidad de objetivos, requisitos, componentes y evidencia

Objetivo	Requisitos	Componentes	Evidencia
Específico 1	Base conceptual	Modelo Tetrad, IA, Unity	Capítulo II
Específico 2	RF-01–30; RNF-01–14	Backlog y criterios	Tablas de requisitos e historias
Específico 3	RF-11–30	Jugador, agente, datos y escenas	Arquitectura y desarrollo

Objetivo	Requisitos	Componentes	Evidencia
Específico 4	RNF-01–14	Pruebas y métricas	Capítulo IV
Específico 5	RNF-08–14	Descripción técnica y operación	Capítulos III–V

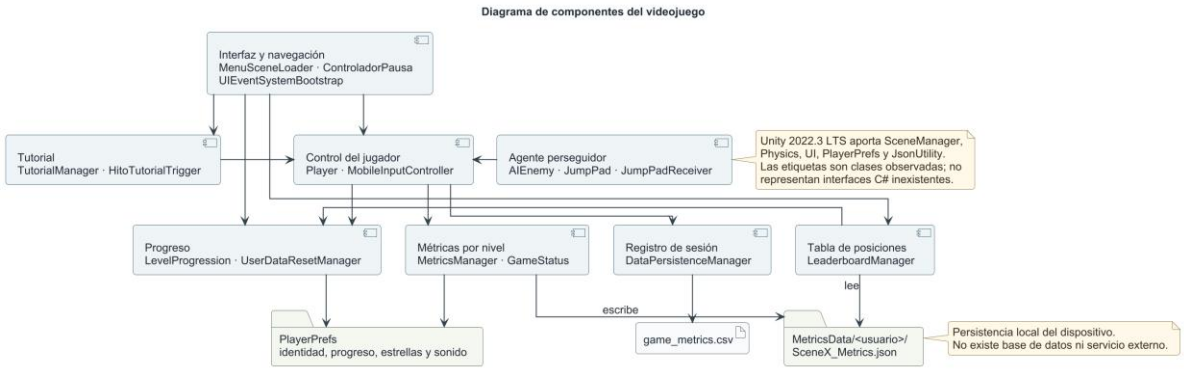
Nota. Esta tabla muestra vínculos operativos para localizar evidencia.

Arquitectura del videojuego

La arquitectura es local, orientada a componentes y organizada por escenas. La capa de presentación contiene registro, menú, tutorial, HUD, pausa y resumen. La capa de gameplay contiene jugador, enemigo, plataformas, objetos, meta y reglas. La capa de persistencia contiene progreso, estrellas, JSON y CSV. La configuración de plataforma adapta entrada y empaquetado sin duplicar las reglas centrales.

Figura 4

Arquitectura lógica de componentes del videojuego



Nota. La figura representa dependencias implementadas entre interfaz, control de escenas, jugador, agente, progreso y persistencia. Las flechas describen colaboración en tiempo de ejecución; no representan servicios de red.

Tabla 18

Responsabilidades de los componentes principales

Componente	Responsabilidad	Entradas	Salidas
MenuSceneLoader	Registro, menú y carga	Perfil y selección	Escena activa
LevelProgression	Tutorial y desbloqueo	Nivel completado	Progreso local
Player	Control y reglas de jugador	Entradas y colisiones	Movimiento y estados
AIEnemy	Persecución autónoma	Jugador, ruta y terreno	Steering y movimiento
MetricsManager	Telemetría de nivel	Eventos de partida	Archivo JSON
GameStatus	Acumulados de sesión	Eventos globales	Resumen temporal

Componente	Responsabilidad	Entradas	Salidas
DataPersistenceManager	Persistencia de sesión	GameStatus	Fila CSV
MobileInputController	Adaptación táctil	Botones de pantalla	Solicitudes de acción
ControladorPausa	Pausa y audio	Entrada de usuario	Estado e interfaz
JumpPad	Trayectoria de salto	Contacto de entidad	Impulso o destino

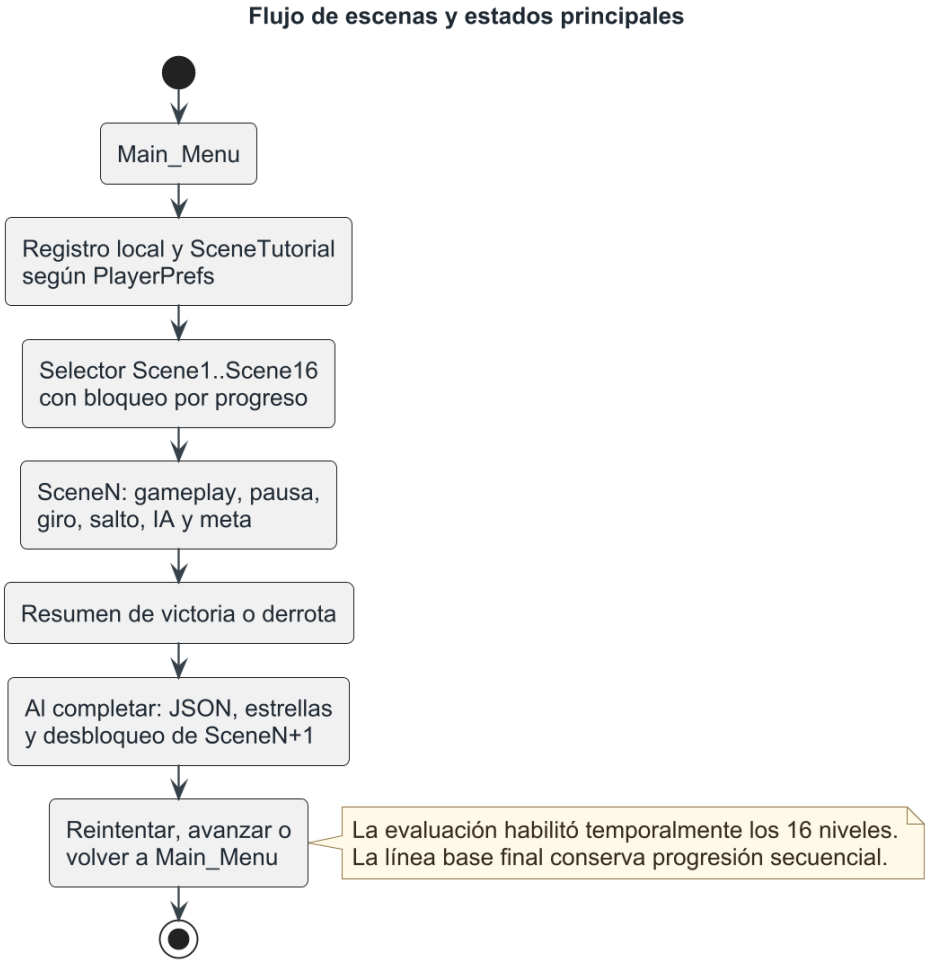
Nota. Las responsabilidades se describen desde la interacción entre componentes.

Flujo de escenas y operación

El producto registra al usuario, presenta el tutorial, habilita el menú y carga un nivel. Durante la partida se actualizan HUD, física, IA y métricas. Una derrota abre Game Over y permite reintentar o salir; alcanzar la bandera calcula estrellas, persiste el nivel y habilita el siguiente.

Figura 5

Flujo de escenas, progreso y resultado



Nota. Flujo implementado desde registro y tutorial hasta menú, niveles y resultado. El progreso se conserva localmente y el cierre del nivel activa el registro de métricas y el cálculo de estrellas.

Controles, movimiento y física

El control lateralizado asigna acciones discretas a entrada izquierda y derecha. La intención de diseño es obligar a seleccionar una dirección en vez de mantener un eje analógico continuo. La acción se valida antes de contabilizarse como efectiva. La versión móvil transmite solicitudes equivalentes desde botones táctiles.

El movimiento se aplica sobre Rigidbody. La velocidad horizontal se actualiza en paso fijo y la rotación visual refleja el desplazamiento. El salto exige contacto o margen temporal de suelo, admite buffer de entrada y aplica impulso vertical. Esta combinación mejora la tolerancia sin eliminar la necesidad de decidir.

Las colisiones distinguen objetos, meta, enemigo y obstáculos. Una guarda evita que la misma derrota se procese varias veces. El reintento restablece posición y estados; el cierre guarda métricas una sola vez.

Tabla 19

Mapa de controles y validaciones

Acción	Escritorio	Android	Validación	Efecto
Dirección A	Clic/tecla izquierda	Botón táctil	Partida activa	Giro y movimiento
Dirección B	Clic/tecla derecha	Botón táctil	Partida activa	Giro opuesto
Salto	Tecla configurada	Botón táctil	Suelo, buffer y margen	Impulso vertical
Pausa	Tecla/botón	Botón táctil	Escena jugable	Detener/reanudar
Reintento	Botón de interfaz	Botón de interfaz	Estado de derrota	Reiniciar nivel
Menú	Botón de interfaz	Botón de interfaz	Pausa o resultado	Cambiar escena

Nota. Las acciones de escritorio y Android convergen en solicitudes comunes. El porcentaje de entradas aceptadas contabiliza las solicitudes que superan estas validaciones.

Diseño de niveles, obstáculos, JumpPads y recolectables

El producto contiene menú, tutorial y 16 escenas jugables incluidas en la configuración de compilación. Cada nivel combina una ruta, punto de inicio, meta, recolectables, jugador, agente e interfaz. La variación proviene del trazado, longitud, frecuencia de giros, obstáculos, saltos y presión temporal.

Los recolectables cumplen funciones de orientación y puntuación. El sistema detecta el total de la escena y calcula una proporción para las estrellas. Los obstáculos producen derrota cuando se activa la condición correspondiente. Los JumpPads de los niveles avanzados pueden impulsar tanto al jugador como al agente mediante un receptor común.

La tabla siguiente organiza los niveles por función de diseño. La agrupación no sustituye la configuración específica de cada escena, pero explica la progresión pedagógica y mecánica.

Tabla 20

Función de diseño de los niveles 1 a 8

Nivel	Fase	Componente principal	Demanda de interacción
1	Introducción	Ruta inicial y persecución	Dirección básica
2	Introducción	Continuidad de giros	Ritmo
3	Introducción	Variación de trayecto	Anticipación
4	Consolidación	Ruta con baja mortalidad observada	Control estable
5	Consolidación	Transición breve	Precisión
6	Consolidación	Mayor repetición de intentos	Persistencia
7	Consolidación	Ritmo intermedio	Coordinación
8	Consolidación	Cobertura menor	Exploración

Nota. La fase resume la progresión funcional. Los resultados cuantitativos se presentan en el capítulo IV con el denominador de cada nivel.

Tabla 21

Función de diseño de los niveles 9 a 16

Nivel	Fase	Componente principal	Demanda de interacción
9	Avanzado	Ruta y recolectables	Atención dividida
10	Avanzado	Tiempo alto observado	Planificación
11	Avanzado	Combinación de giros	Flexibilidad
12	Avanzado	Mayor presión observada	Control del ritmo
13	Avanzado	Escenario complejo	Adaptación
14	Avanzado	Presión del perseguidor	Decisión rápida
15	Salto	Obstáculos y JumpPads	Sincronización
16	Salto	Cierre con JumpPad y alta dificultad	Integración de acciones

Nota. Los niveles 15 y 16 contienen referencias verificadas de JumpPads. La dificultad se configura por escena y no cambia automáticamente por usuario.

Materialización de los nodos Tetrad en el desarrollo

El Tetrad orientó decisiones durante los incrementos. Una nueva mecánica exigió revisar su representación visual, su lugar en la progresión y su soporte técnico. Al incorporar JumpPads, por ejemplo, se integraron objeto, feedback, animación, colisión, trayectoria, recepción por el agente y niveles avanzados.

La Estética se implementó mediante interfaces y sonido; la Historia mediante secuencia y progresión; la Tecnología mediante Unity, componentes, datos y plataformas; y las Mecánicas mediante reglas e IA. Esta aplicación evita que un nodo evolucione sin considerar los demás. La trazabilidad con historias y pruebas extiende esa relación al proceso de ingeniería.

Integración del flujo de juego, datos y retroalimentación

El flujo integrado comienza con la creación o recuperación del perfil local. El participante ingresa nombre y edad, y el sistema valida el rango configurado antes de habilitar la navegación principal. La selección de jugar conduce al tutorial o al nivel disponible, mientras la selección de escenario respeta el progreso conservado. Esta secuencia evita que las escenas funcionen como archivos independientes: menú, perfil, progresión y estado de juego forman una operación continua.

Al iniciar un nivel, el estado global prepara tiempo, intentos, recolectables y referencias de interfaz. Player recibe entradas de teclado o controles táctiles y las traduce a decisiones de dirección y salto. La física se actualiza en el ciclo fijo para conservar consistencia con Rigidbody, mientras los elementos visuales reflejan el estado ya calculado. Separar entrada, física y presentación reduce dependencias y permite que Windows y Android compartan la misma lógica central.

AIEnemy obtiene la referencia del jugador y la información necesaria para construir su objetivo. El componente no consulta respuestas de encuesta ni modifica la dificultad de acuerdo con la edad. Su conducta depende de variables del nivel: distancia, velocidad, trayectoria, límites de predicción, terreno y estado de persecución. Esta independencia es importante para la trazabilidad, porque permite asociar la IA con requisitos técnicos concretos y no con inferencias sobre la persona.

Los eventos de caída, contacto con el enemigo, recolección y llegada a meta actualizan el estado correspondiente. Una caída incrementa las medidas de intento y muerte, ejecuta

recuperación y conserva la posibilidad de continuar. La influencia del enemigo se registra cuando la lógica del juego la atribuye a ese agente. La recolección incrementa el contador hasta el total disponible; la meta detiene el tiempo, calcula estrellas, presenta el resultado y habilita la progresión prevista. Cada evento tiene una consecuencia visible para que el usuario pueda comprender el vínculo entre acción y resultado.

Al finalizar, MetricsManager reúne los valores operacionales y forma un registro serializable. DataPersistenceManager gestiona la escritura local mediante JsonUtility y el directorio persistente que Unity asigna a la plataforma. El archivo conserva un estado por participante y nivel; por ello, una nueva ejecución del mismo par puede reemplazar la anterior. Esta decisión favorece una lectura resumida del último estado, pero impide recuperar el historial completo de intentos. El análisis respeta esa granularidad y denomina a cada fila registro participante–nivel.

La retroalimentación se distribuye en tres momentos. Durante el nivel, HUD, sonido y animaciones informan estado inmediato. Después de la meta, estrellas, tiempo y recolectables condensan el resultado. Al cerrar la sesión de evaluación, la encuesta recoge comprensión, respuesta, motivación, reto y claridad del desempeño. Estas capas corresponden a nodos distintos del Tetrad: la interfaz participa en Estética, la secuencia en Historia, la persistencia en Tecnología y las reglas de cálculo en Mecánicas. La integración evita interpretar la encuesta como sustituto de la telemetría o viceversa.

El tratamiento de errores también pertenece al flujo. Las referencias se comprueban antes de usarse, las predicciones no finitas se descartan y el perseguidor vuelve a una conducta segura cuando no dispone de un objetivo válido. El jugador y el agente cuentan con mecanismos de respawn. La persistencia opera con estructuras definidas, y la interfaz conserva acciones para pausar, reanudar o salir. Estas decisiones sostienen fiabilidad y seguridad operacional dentro del alcance del producto.

La trazabilidad del flujo se verificó desde ambos extremos. Desde el requisito se identifica la historia, el componente, el evento y el caso de prueba que lo cubre. Desde un resultado se identifica la regla que lo produjo y el campo que lo conserva. Así, tiempo, muertes o recolectables no aparecen como cifras aisladas: proceden de estados definidos del software. Esta relación respalda la reproducibilidad de los cálculos descriptivos y la mantenibilidad de la solución.

Implementación de Seek, Pursuit y navegación segura

AIEnemy actualiza la trayectoria del jugador, resuelve un objetivo de steering y aplica movimiento. En Seek, el objetivo es la posición del jugador o una proyección segura. En Pursuit, el componente obtiene una velocidad de referencia, calcula un horizonte, limita la distancia prevista y busca una posición con suelo.

La selección de Pursuit depende de distancia, velocidad y estado anterior. La histéresis utiliza umbrales diferentes para entrar y salir del modo. La predicción se invalida si contiene valores no finitos, si la velocidad es insuficiente, si el desplazamiento excede el máximo o si el terreno no es transitable. En esos casos se conserva Seek.

La seguridad combina búsqueda de suelo, diferencia de altura, muestreo del segmento y trayectoria histórica. Si la ruta exige una esquina, el agente sigue puntos acumulados. Si falta suelo delante, evalúa salto autónomo. Una caída bajo el umbral activa respawn en una posición segura. La tabla siguiente reúne las decisiones.

Tabla 22

Decisiones del agente autónomo

Condición	Decisión	Control de seguridad	Resultado
Velocidad insuficiente	Seek	Objetivo con suelo	Seguimiento estable
Distancia corta	Seek	Histéresis	Evita predicción innecesaria
Velocidad y distancia válidas	Intentar Pursuit	Horizonte y desplazamiento máximos	Anticipación acotada

Condición	Decisión	Control de seguridad	Resultado
Predicción sin suelo	Reducir o descartar	Muestreo de candidato	Retorno seguro
Segmento no transitable	Usar trayectoria	Muestras acumuladas	Respeto de esquinas
Hueco próximo	Evaluar salto	Aterrizaje y tiempo de reutilización	Salto autónomo
Agente bajo umbral	Respawn	Posición segura	Reanudación de persecución
Cambio mínimo de distancia	Conservar modo	Histéresis	Menor oscilación

Nota. Las decisiones se derivan del flujo implementado. Seek actúa como modo base y Pursuit se utiliza únicamente cuando la predicción cumple las condiciones.

Persistencia, métricas y diccionario operativo

MetricsManager inicia contadores al cargar un nivel y recibe eventos del jugador. Al guardar, calcula tiempo, entradas aceptadas, primera interacción, intervalos y eventos del enemigo. JsonUtility serializa una estructura pública y el sistema escribe el archivo en almacenamiento persistente. El resumen CSV se guarda desde el estado acumulado de la sesión.

El grano es indispensable: un JSON describe un nivel, mientras una fila CSV describe una sesión. Por tanto, no se suman como si fueran registros equivalentes. La tabla siguiente define las métricas utilizadas en resultados.

Tabla 23

Diccionario operativo de métricas de nivel

Campo	Definición	Unidad	Grano
time_spent_seconds	Duración del registro de nivel	Segundos	Inicio a guardado
total_attempts	Intentos acumulados	Conteo	Nivel
deaths	Derrotas registradas	Conteo	Nivel
coins_collected	Recolectables obtenidos	Conteo	Nivel
total_coins_available	Total detectado en escena	Conteo	Nivel
total_clicks	Entradas registradas	Conteo	Nivel
click_accuracy_percentage	Entradas aceptadas / total × 100	Porcentaje	Nivel

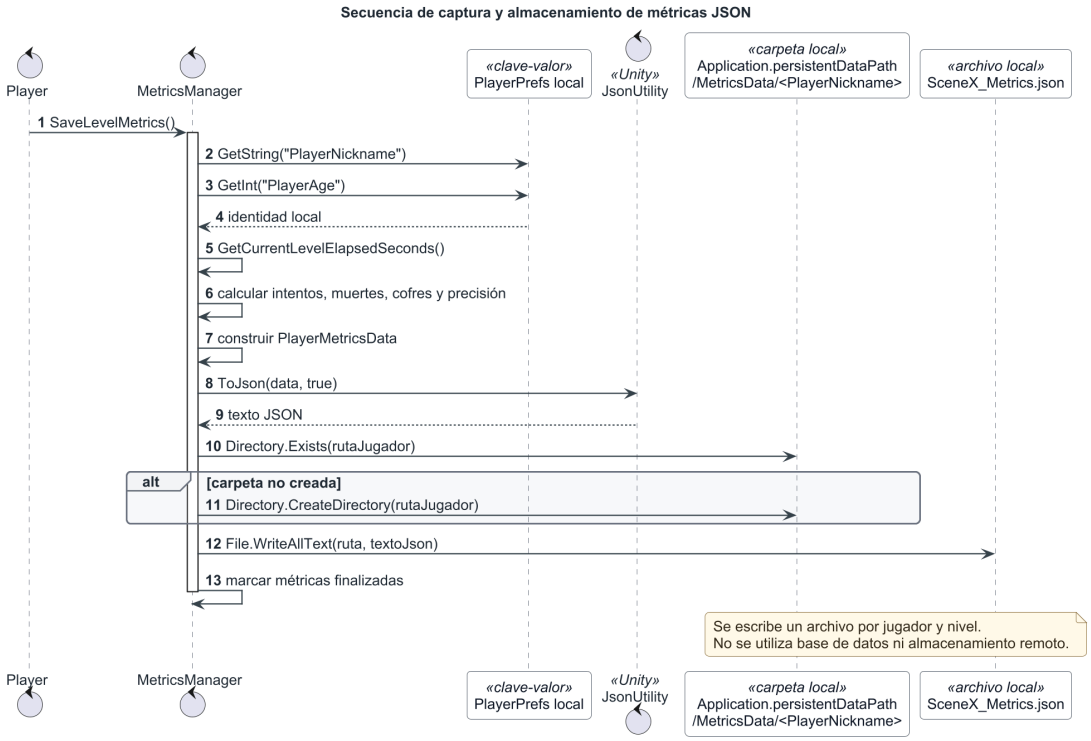
Campo	Definición	Unidad	Grano
first_click_delay_seconds	Demora hasta la primera entrada	Segundos	Nivel
avg_time_between_respawn_and_death	Intervalo medio posterior a respawn	Segundos	Nivel
enemy_influence_deaths	Muertes asociadas a proximidad/contacto del enemigo	Conteo	Nivel
level_name	Escena jugada	Categoría	Nivel
timestamp	Fecha y hora de guardado	Fecha-hora	Nivel

Nota. Las definiciones corresponden al cálculo del sistema. El campo denominado click_accuracy_percentage se reporta como porcentaje de entradas aceptadas y no como precisión cognitiva.

La Figura 6 muestra la secuencia de persistencia por nivel.

Figura 6

Secuencia de captura y persistencia de métricas JSON



Nota. La secuencia inicia con eventos de Player, actualiza MetricsManager, serializa los campos mediante JsonUtility y escribe un archivo por combinación de participante y nivel en el almacenamiento persistente.

Configuración y despliegue en Windows y Android

Las dos plataformas comparten escenas, reglas, IA, progreso y persistencia. Windows utiliza teclado y ratón; Android utiliza botones táctiles y orientación horizontal. El sistema selecciona la ubicación persistente que Unity asigna a cada plataforma.

La configuración móvil incorpora arquitectura ARM64 y controles adaptados. El producto no requiere conexión para el gameplay ni para guardar resultados. La compatibilidad se evaluó mediante uso reportado en ambas plataformas y la presencia de registros completos.

Tabla 24

Configuración comparativa de plataformas

Aspecto	Windows	Android	Elemento común
Entrada	Ratón y teclado	Botones táctiles	Solicitudes de dirección y salto
Orientación	Ventana horizontal	Paisaje	Cámara de gameplay
Persistencia	Directorio local de aplicación	Directorio local de aplicación	PlayerPrefs, JSON y CSV
Escenas	Menú, tutorial y 16 niveles	Menú, tutorial y 16 niveles	Build Settings compartido
Conectividad	No requerida	No requerida	Operación local
Evaluación	33 participantes	17 participantes	Encuesta y telemetría

Nota. Las cifras de evaluación corresponden a la plataforma declarada por los 50 participantes. La telemetría no guarda la plataforma dentro de cada JSON.

Diseño de pruebas

Las pruebas de caja negra examinaron entradas y resultados visibles: registro, tutorial, movimiento, recolección, derrota, meta, persistencia y plataformas. Las pruebas de caja blanca examinaron rutas de decisión: guardas, Seek/Pursuit, validación de terreno, salto, respawn y guardado único. La compilación por lotes verificó sintaxis, referencias y dependencias del proyecto.

Un plan de prueba se considera resultado únicamente cuando existe una ejecución identificable. Por ello, el capítulo IV separa la prueba ejecutada de los aspectos que solo pueden inferirse o que requieren instrumentación adicional. No se declara cobertura automatizada de PlayMode.

Tabla 25

Casos de prueba de caja negra

ID	Flujo	Entrada	Resultado esperado
CN-01	Registro	Nombre y edad válidos	Perfil local creado
CN-02	Tutorial	Primer acceso	Instrucciones y práctica disponibles
CN-03	Movimiento	Entradas izquierda/derecha	Cambio de dirección
CN-04	Recolección	Contacto con objeto	Contador y HUD actualizados
CN-05	Caída	Salir de plataforma	Derrota y reintento
CN-06	Persecución	Contacto del agente	Derrota asociada al enemigo
CN-07	Meta	Contacto con bandera	Resumen, estrellas y progreso
CN-08	Persistencia	Cierre de nivel	JSON completo
CN-09	Windows	Sesión de escritorio	Flujo esencial operativo
CN-10	Android	Sesión táctil	Flujo esencial operativo

Nota. Los casos se vinculan con eventos observables y registros de campo. Los resultados ejecutados se presentan en el capítulo IV.

Tabla 26

Casos de prueba de caja blanca

ID	Ruta interna	Condición	Criterio
CB-01	Guardado de derrota	Evento repetido	Procesar una sola vez
CB-02	Guardado de nivel	Múltiples llamadas	Crear un registro
CB-03	Selección de Pursuit	Velocidad y distancia válidas	Activar predicción
CB-04	Retorno a Seek	Predicción inválida	Mantener objetivo seguro
CB-05	Validación de suelo	Objetivo sobre vacío	Rechazar candidato
CB-06	Validación de segmento	Ruta corta no transitable	Usar trayectoria
CB-07	Salto autónomo	Hueco y aterrizaje válido	Aplicar salto
CB-08	Respawn del agente	Caída bajo umbral	Reubicar y continuar
CB-09	Progreso	Nivel completado	Desbloquear siguiente
CB-10	Restablecimiento	Confirmación del usuario	Eliminar datos locales

Nota. Los casos de caja blanca cubren decisiones críticas del código y se verifican mediante inspección estructural, compilación y evidencia producida por el uso.

Indicadores de calidad ISO/IEC 25010

Tabla 27

Indicadores seleccionados para evaluar la calidad

Característica	Indicador	Fuente	Criterio de interpretación
Adecuación funcional	Cobertura de flujos y niveles	Escenas, pruebas y registros	Funciones esenciales observables
Eficiencia	Tiempo y estabilidad de interacción	Telemetría	Magnitudes plausibles; sin benchmark FPS
Compatibilidad	Uso en Windows y Android	Encuesta	Sesiones completas en ambas
Capacidad de interacción	Medias por ítem y dimensión	Encuesta	Valoración favorable y comentarios
Fiabilidad	JSON legibles y completos	Telemetría	Sin campos obligatorios vacíos
Seguridad	Persistencia local y pseudonimización	Arquitectura y análisis	Sin transferencia de red
Mantenibilidad	Compilación y separación de componentes	Proyecto	Dependencias resueltas
Flexibilidad	Escenas y reglas compartidas	Configuración	Mismo núcleo en dos plataformas
Seguridad operacional	Recuperación ante caída	Gameplay	Reintento y respawn controlados

Nota. Los indicadores se seleccionan por disponibilidad y relación con el producto. ISO/IEC 25010:2023 se usa como marco de organización y no como certificado.

Documentación técnica y operativa

La documentación técnica queda integrada en este capítulo: requisitos, historias, arquitectura, responsabilidades, controles, física, niveles, IA, persistencia, plataformas, pruebas e indicadores. Esta organización permite localizar decisiones con una lectura autosuficiente dentro de los cinco capítulos.

Operativamente, el usuario debe iniciar la aplicación, registrar un nombre y edad, completar el tutorial, seleccionar Jugar, utilizar las entradas de dirección, aplicar salto cuando aparezca, recolectar objetos, observar el perseguidor y alcanzar la meta. El menú de pausa permite reanudar, configurar audio o salir. La pantalla de resultado permite revisar estrellas y continuar.

Para restablecer la experiencia se usa la opción de reinicio de datos, que elimina perfil, progreso, estrellas y métricas locales. Para conservar resultados, el responsable debe copiar

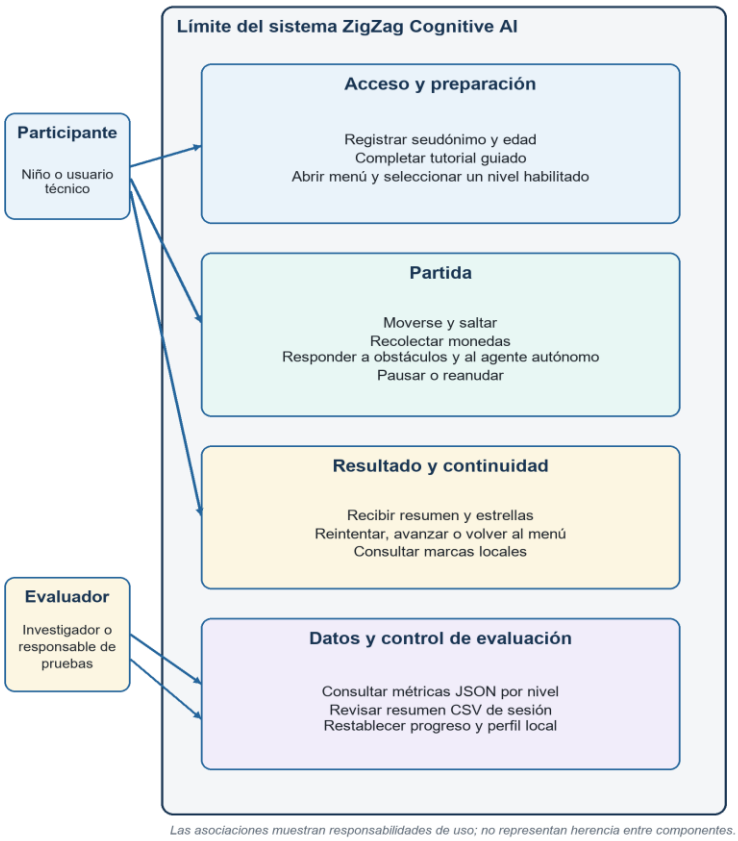
los registros antes de restablecer. La interpretación de métricas debe respetar las fórmulas y los denominadores definidos en esta sección.

Modelado funcional consolidado y mantenibilidad

La revisión conjunta de requisitos, escenas, scripts y artefactos de compilación permitió consolidar modelos que complementan la arquitectura ya descrita sin sustituir los diagramas vigentes de Seek/Pursuit ni el Modelo de la Tétrada Elemental Modificado. Los modelos siguientes representan únicamente responsabilidades y dependencias comprobables en la versión evaluada del producto.

Figura 7

Panorama consolidado de casos de uso del videojuego



Nota. Elaboración propia a partir de MenuSceneLoader, Player, AIEnemy, MetricsManager, DataPersistenceManager, LeaderboardManager y LevelProgression. El evaluador consulta o restablece datos locales; no se representa un servicio remoto ni una base de datos externa.

Tabla 28

Catálogo de casos de uso de acceso, configuración y navegación

ID	Caso y actor	Precondición	Flujo y resultado	RF
CU-01	Registrar perfil Participante	Primera apertura o perfil restablecido.	Ingresar seudónimo y confirmar una edad válida. Resultado: perfil local disponible.	RF-01–02
CU-02	Restablecer perfil Evaluador	Menú accesible.	Confirmar el borrado de progreso y datos. Resultado: flujo inicial disponible.	RF-09
CU-03	Completar tutorial Participante	Perfil válido y tutorial pendiente.	Ejecutar hitos de movimiento, giro y salto. Resultado: tutorial completado.	RF-03
CU-04	Usar menú Participante	Acceso validado.	Elegir jugar, nivel, opciones o tabla local. Resultado: vista solicitada activa.	RF-04, 08, 10
CU-05	Seleccionar nivel Participante	Menú activo.	Consultar disponibilidad y solicitar Scene1…Scene16. Resultado: nivel habilitado cargado.	RF-05–06
CU-06	Controlar personaje Participante	Partida activa.	Moverse, girar y saltar con teclado o tacto. Resultado: acción aplicada por Player.	RF-11–13
CU-07	Recolectar moneda Participante	Partida activa y moneda disponible.	Entrar en el disparador. Resultado: contador y métricas actualizados.	RF-14
CU-08	Pausar y reanudar Participante	Partida activa.	Abrir pausa y reanudar o volver al menú. Resultado: tiempo de juego controlado.	RF-07

Nota. La precondición y el resultado se formularon con el flujo de Main_Menu, SceneTutorial y Scene1…Scene16.

Tabla 29

Catálogo de casos de uso de partida, inteligencia artificial, resultados y datos

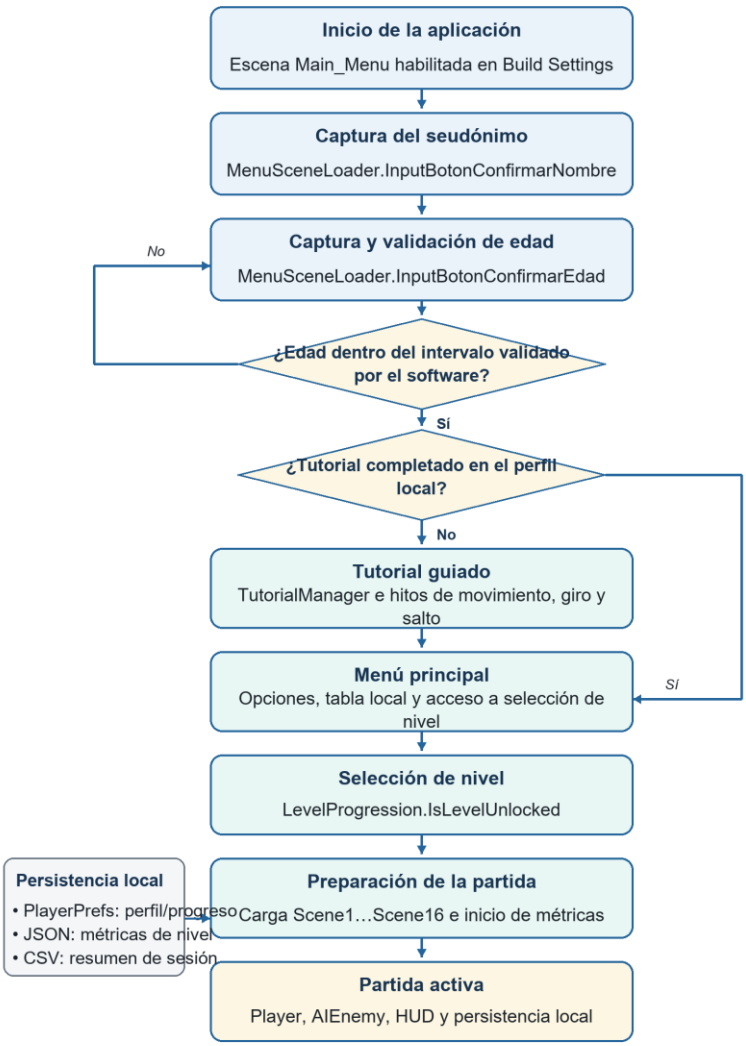
ID	Caso y actor	Precondición	Flujo y resultado	RF
CU-09	Finalizar nivel Participante	Bandera alcanzada.	Calcular tiempo, monedas y estrellas; cerrar métricas. Resultado: resumen y progreso actualizados.	RF-17–19, 28
CU-10	Gestionar derrota Participante	Caída, obstáculo o contacto mortal.	Registrar muerte y presentar Game Over. Resultado: reintento o salida disponibles.	RF-15–16, 19
CU-11	Continuar partida Participante	Resumen o Game Over visible.	Elegir reintento, siguiente nivel o menú. Resultado: escena solicitada cargada.	RF-20
CU-12	Perseguir al jugador Agente IA	Jugador y enemigo activos.	Alternar Seek/Pursuit y validar terreno. Resultado: movimiento limitado al suelo transitable.	RF-21–27
CU-13	Resolver salto asistido Sistema	Contacto con JumpPad.	Bloquear control, trasladar por arco y devolver control. Resultado: objeto reposicionado.	RF-13, 26
CU-14	Guardar métricas Sistema	Evento o cierre de nivel.	Registrar interacciones y serializar. Resultado: JSON local actualizado.	RF-28
CU-15	Guardar sesión Sistema	Cierre lógico de sesión.	Consolidar datos y escribir CSV. Resultado: resumen de sesión disponible.	RF-29
CU-16	Consultar tabla local Participante	Registros JSON existentes.	Ordenar por monedas y tiempo y mostrar cinco marcas. Resultado: clasificación visible.	RF-10, 28
CU-17	Ejecutar en dos plataformas Participante	Compilación instalada.	Usar teclado en Windows o tacto en Android. Resultado: misma lógica de partida operativa.	RF-12, 30

Nota. El caso CU-12 describe parámetros configurables de persecución y validación del terreno; no implica dificultad adaptativa. LeaderboardManager consulta y ordena archivos JSON locales, mientras que DataPersistenceManager escribe el resumen CSV de sesión.

Secuencia de acceso y preparación de una partida

Figura 8

Secuencia validada de acceso, tutorial y carga de nivel

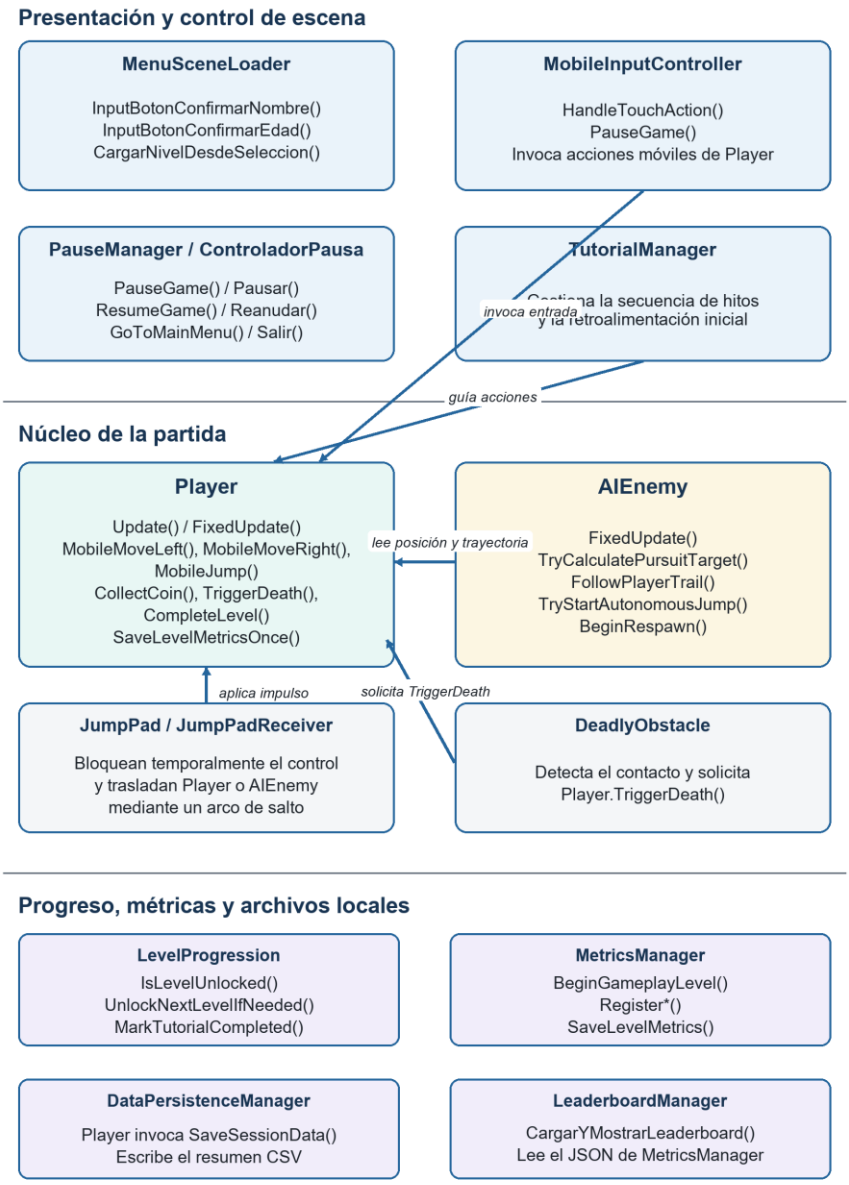


Nota. Elaboración propia a partir de MenuSceneLoader, TutorialManager, LevelProgression y EditorBuildSettings. La marca de tutorial y el progreso se conservan en PlayerPrefs; el inicio de métricas ocurre al cargar una escena de juego.

Colaboración entre clases implementadas

Figura 9

Colaboración entre clases de interfaz, partida y persistencia



Nota. La figura muestra dependencias de colaboración y métodos en el código.

Interfaces operativas y evidencia de responsabilidad

Tabla 30

Interfaces operativas y responsabilidades verificadas

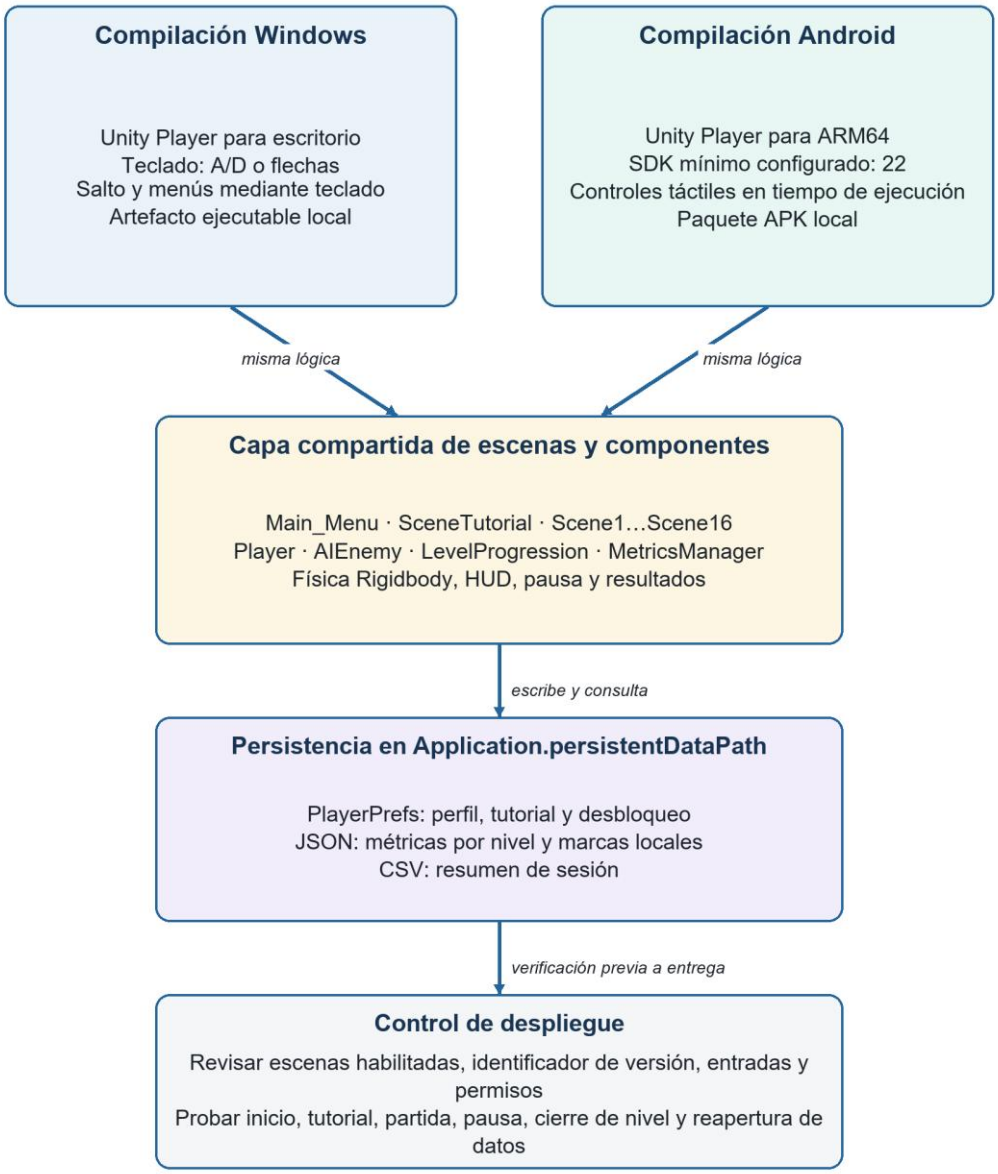
Interfaz	Propósito y activación	Responsable verificado	Evidencia
NamePanel	Capturar seudónimo mediante confirmación de texto.	MenuSceneLoader	InputBotonConfirmarNombre()
AgePanel	Validar la edad mediante confirmación numérica.	MenuSceneLoader	InputBotonConfirmarEdad()
Tutorial	Guiar acciones mediante hitos de SceneTutorial.	TutorialManager	Secuencia de objetivos y mensajes
MenuPanel	Centralizar navegación mediante botones.	MenuSceneLoader	Carga de vistas y escenas
LevelPanel	Elegir Scene1…Scene16 según disponibilidad.	MenuSceneLoader / LevelProgression	CargarNivelDesdeSeleccion(); IsLevelUnlocked()
HUD	Mostrar estado durante la partida.	Player	UpdateHud()
PausePanel	Detener o reanudar por tecla o tacto.	PauseManager / ControladorPausa	PauseGame(), ResumeGame(), Pausar(), Reanudar()
Game Over	Informar derrota tras TriggerDeath().	Player	BotonAccionReiniciar() y menú
SummaryPanel	Presentar resultado al finalizar.	Player	FinalizarYMostrarResumen()
Leaderboard	Mostrar marcas locales desde el menú.	LeaderboardManager	CargarYMostrarLeaderboard()

Nota. Relaciona las vistas que intervienen en el recorrido del usuario con el componente que activa o actualiza cada estado..

Despliegue verificable en Windows y Android

Figura 10

Estructura de despliegue y persistencia por plataforma



Nota. Diferencia las entradas y los artefactos de cada plataforma, y conserva como capa común las escenas, la lógica de partida y los archivos locales.

Capítulo IV

Evaluación, resultados y discusión

Organización de los resultados

Los resultados se presentan en cuatro bloques: calidad y cobertura de datos; telemetría de la población infantil y del grupo técnico adulto; encuesta cuantitativa y cualitativa; y pruebas de software e indicadores de calidad. Cada bloque separa resultado, interpretación y discusión para evitar que una expectativa de diseño se confunda con un hallazgo.

La población infantil constituye la evidencia principal. El grupo técnico adulto aporta una referencia de operación con usuarios de mayor edad, pero no se utiliza para estimar efecto ni como condición equivalente.

Calidad y cobertura de los datos

Se analizaron 248 archivos JSON de 50 participantes y los 16 niveles. Todos los archivos pudieron leerse. No se detectaron valores ausentes en los campos obligatorios (0 celdas faltantes), duplicados por participante–nivel (0) ni valores fuera de los rangos básicos comprobados (0). El total de recolectables disponible fue 18 en los registros analizados.

La encuesta aportó 50 respuestas completas, una por participante, sin valores ausentes en los 15 ítems o las tres preguntas abiertas.

La ausencia de duplicados es coherente con el diseño de un archivo por combinación participante–nivel. También implica que una repetición posterior del mismo nivel puede reemplazar el archivo anterior; por ello, la base representa el registro conservado y no necesariamente todos los intentos históricos como archivos independientes.

Tabla 31

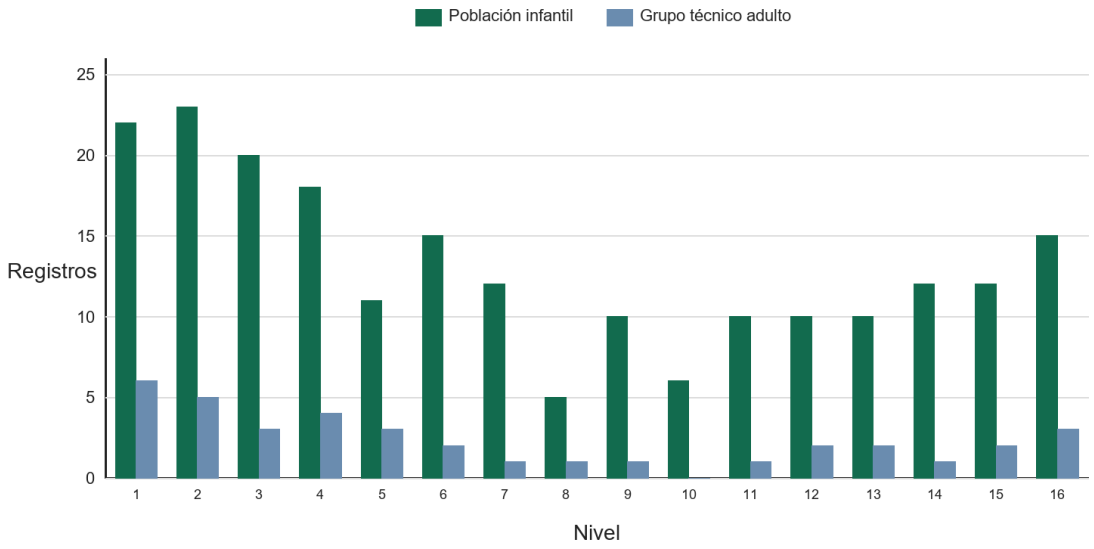
Resultados de integridad y cobertura

Control	Resultado	Interpretación
Archivos JSON legibles	248 de 248	Cobertura completa de la base utilizada
Participantes con telemetría	50	Coinciden con respuestas de encuesta
Niveles representados	16 de 16	Cobertura del producto
Campos obligatorios ausentes	0	Estructura completa
Duplicados participante–nivel	0	Un registro conservado por combinación
Valores fuera de rango básico	0	Consistencia aritmética
Encuestas completas	50 de 50	Sin imputación

Nota. Los controles se aplicaron a 248 registros JSON y 50 encuestas. La verificación incluyó lectura, campos obligatorios, duplicados y rangos de tiempo, intentos, muertes, recolectables y porcentaje.

Figura 11

Cobertura de registros por nivel y grupo



Nota. Cada barra contabiliza combinaciones participante–nivel conservadas. La población infantil aporta 211 registros y el grupo técnico adulto 37.

Resultados de telemetría de la población infantil

La población infantil generó 211 registros, con una media de 4.80 niveles distintos por participante. El tiempo medio fue 63.20 s y la mediana 54.63 s, lo que indica asimetría producida por partidas prolongadas. Los intentos tuvieron media 5.45 y mediana 4; las muertes, media 4.30 y mediana 3.

El promedio de recolectables fue 12.04 de 18. El porcentaje de entradas aceptadas alcanzó 97.27 %, con mediana 98.00 %. Este valor significa que la mayoría de entradas fueron procesadas por la lógica, no que las decisiones fueran correctas desde una perspectiva cognitiva. La demora media del primer clic fue 1.76 s.

Las muertes influidas por el enemigo promediaron 1.31 por registro y presentaron mediana 1. El resultado confirma que la persecución produjo eventos observables sin explicar todas las derrotas; caídas y obstáculos también contribuyeron.

Tabla 32

Telemetría de la población infantil

Métrica	Media	DE	Mediana	Q1–Q3	Mín.–máx.
Tiempo (s)	63.20	44.49	54.63	37.30–74.63	8.48–330.82
Intentos	5.45	5.36	4.00	2.50–6.00	1.00–44.00
Muertes	4.30	5.35	3.00	1.00–5.00	0.00–43.00
Recolectables	12.04	3.02	12.00	10.00–14.00	3.00–18.00
Entradas	49.39	48.53	37.00	26.50–53.50	7.00–446.00
Entradas aceptadas (%)	97.26	3.07	98.00	95.19–100.00	84.41–100.00
Demora primer clic (s)	1.76	1.15	1.43	1.04–2.12	0.00–6.82
Intervalo tras respawn (s)	4.95	4.26	3.67	2.11–7.24	0.00–21.33
Muertes influidas por enemigo	1.31	1.56	1.00	0.00–2.00	0.00–13.00

Nota. Resultados calculados sobre 211 registros de 44 niños. Un participante puede aportar varios niveles; por eso las filas no son observaciones independientes entre sí.

Resultados del grupo técnico complementario de participantes adultos

El grupo técnico adulto generó 37 registros, con una media de 6.17 niveles por participante. El tiempo medio fue 73.12 s y la mediana 34.64 s; la diferencia entre ambas

muestra la influencia de registros prolongados. Los intentos presentaron media 5.65 y mediana 2, y las muertes media 4.65 y mediana 1.

Los recolectables promediaron 12.84, el porcentaje de entradas aceptadas 99.11 % y la demora del primer clic 1.55 s. Las muertes influidas por el enemigo promediaron 1.16. Estos datos confirman la operación del producto con adultos, pero el tamaño y la función del grupo no permiten inferir superioridad ni equivalencia con los niños.

Tabla 33

Telemetría del grupo técnico adulto

Métrica	Media	DE	Mediana	Q1–Q3	Mín.–máx.
Tiempo (s)	73.12	79.25	34.64	27.71–86.69	9.21–390.41
Intentos	5.65	8.49	2.00	1.00–6.00	1.00–44.00
Muertes	4.65	8.49	1.00	0.00–5.00	0.00–43.00
Recolectables	12.84	3.52	13.00	11.00–15.00	4.00–18.00
Entradas	65.89	83.25	33.00	24.00–62.00	16.00–447.00
Entradas aceptadas (%)	99.10	1.76	100.00	98.53–100.00	92.93–100.00
Demora primer clic (s)	1.55	1.18	0.94	0.77–2.10	0.40–4.73
Intervalo tras respawn (s)	3.93	4.55	2.30	0.00–7.22	0.00–16.68
Muertes influidas por enemigo	1.16	2.29	0.00	0.00–1.00	0.00–12.00

Nota. Resultados calculados sobre 37 registros de 6 participantes adultos. Las magnitudes se presentan como comprobación técnica complementaria y no como comparación causal.

Comprobación mediante promedios por participante

Para reducir la ponderación desigual, se promedió cada métrica dentro de cada participante y luego se resumieron los promedios individuales. En la población infantil, el tiempo medio de los promedios fue 63.72 s, los intentos 5.52, las muertes 4.36 y las muertes influidas por el enemigo 1.31. Estos valores son cercanos a los obtenidos por registro.

En el grupo técnico adulto, los promedios individuales fueron 73.75 s, 5.74 intentos, 4.74 muertes y 1.10 muertes influidas por el enemigo. La cercanía confirma que el patrón general no surge únicamente de uno o dos participantes con mayor cobertura, aunque la variabilidad y el tamaño del grupo se mantienen.

Tabla 34

Promedios por participante de tiempo, intentos y muertes

Grupo	n	Tiempo (s)	Intentos	Muertes
Población infantil	44	63.72	5.51	4.36
Grupo técnico adulto	6	73.75	5.74	4.74

Nota. Primero se calculó la media de cada participante y después la media del grupo. Esta estrategia asigna el mismo peso a cada persona, con independencia del número de niveles conservados.

Tabla 35

Promedios por participante de recolección, entradas aceptadas e influencia del enemigo

Grupo	n	Recolectables	Entradas aceptadas (%)	Muertes influidas por el enemigo
Población infantil	44	11.96	97.35	1.31
Grupo técnico adulto	6	12.53	99.08	1.10

Nota. Los valores se obtuvieron con el mismo procedimiento de ponderación por participante. Entradas aceptadas describe procesamiento de software y no exactitud cognitiva.

Resultados por nivel

En la población infantil, los niveles 1, 10, 12, 15 y 16 presentaron tiempos medios relativamente altos. El nivel 12 mostró 10.60 intentos y 8.70 muertes; el nivel 1, 9.41 intentos y 8.27 muertes. Los niveles 4, 5 y 7 presentaron menor mortalidad media. La serie no es monotónica: la dificultad emerge de la combinación de trazado, experiencia previa, obstáculos y cobertura.

Los recolectables disminuyeron en los niveles 15 y 16, con medias de 9.25 y 8.07 en niños. En paralelo, las muertes influidas por el enemigo alcanzaron 3.30 en el nivel 12 y 2.47 en el 16. Estos resultados indican concentración de presión en algunos escenarios y respaldan la función de la IA como componente del reto.

En el grupo técnico adulto hubo niveles con uno o dos registros y ausencia de cobertura en el nivel 10. Sus valores por nivel se presentan para transparencia, pero no se interpretan como una curva estable.

Tabla 36

Resultados infantiles por nivel, niveles 1 a 8

Nivel	n	Tiempo (s)	Intentos	Muertes
1	22	95.62	9.41	8.27
2	23	50.22	4.65	3.65
3	20	73.40	6.10	5.00
4	18	51.33	2.00	0.94
5	11	39.44	2.82	1.73
6	15	66.59	6.40	4.73
7	12	39.00	2.58	1.58
8	5	55.22	5.60	4.60

Nota. Medias por registro de la población infantil. El n se conserva por nivel para mostrar la cobertura desigual.

Tabla 37

Resultados infantiles por nivel, niveles 9 a 16

Nivel	n	Tiempo (s)	Intentos	Muertes
9	10	52.95	3.50	2.10
10	6	103.14	5.33	4.17
11	10	47.26	3.80	2.80
12	10	73.18	10.60	8.70
13	10	55.15	5.10	4.10
14	12	55.54	5.92	4.92
15	12	75.95	5.58	4.58
16	15	68.32	6.13	5.13

Nota. Medias por registro de la población infantil. Los denominadores desiguales limitan la lectura como una secuencia balanceada.

Tabla 38

Resultados del grupo técnico adulto por nivel, niveles 1 a 8

Nivel	n	Tiempo (s)	Muertes
1	6	111.42	7.83
2	5	72.69	7.60
3	3	39.60	1.67
4	4	56.46	1.00
5	3	23.86	0.33
6	2	75.60	6.00
7	1	29.27	0.00

Nivel	n	Tiempo (s)	Muertes
8	1	32.40	0.00

Nota. Medias por registro del grupo técnico complementario de participantes adultos. Los valores son una comprobación de operación y no un comparador infantil.

Tabla 39

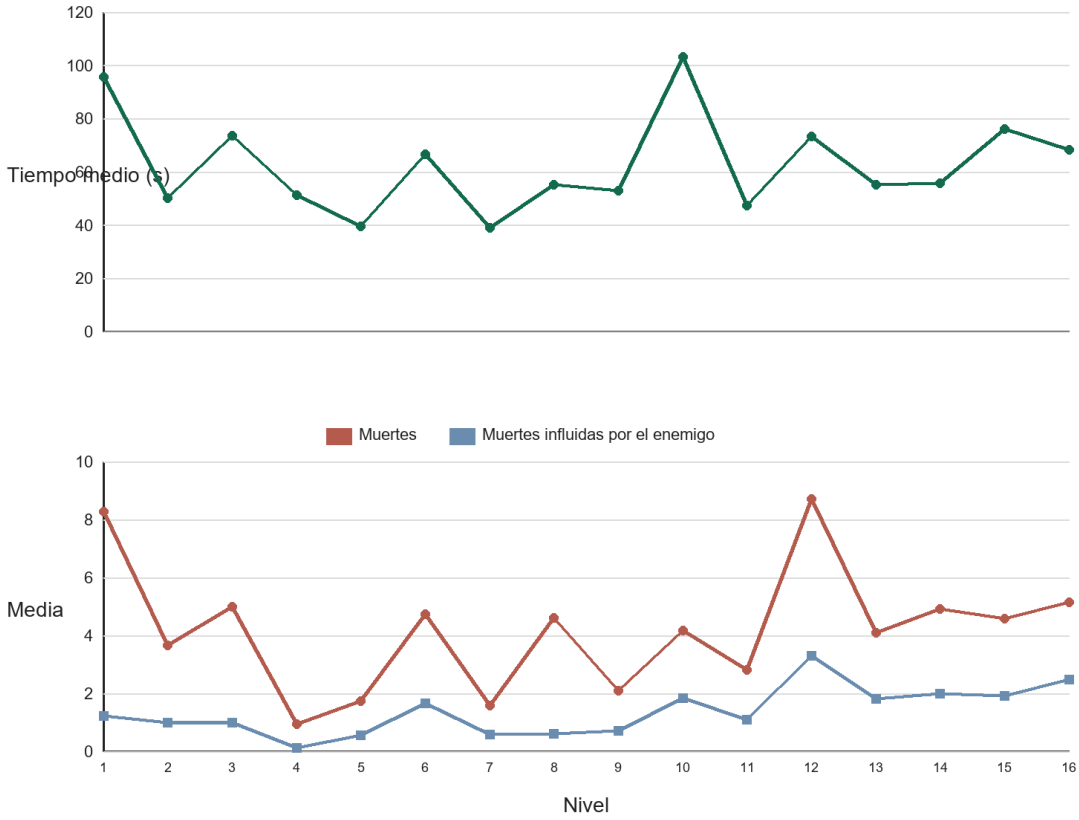
Resultados del grupo técnico adulto por nivel, niveles 9 a 16

Nivel	n	Tiempo (s)	Muertes
9	1	31.99	0.00
10	0	—	—
11	1	92.96	1.00
12	2	52.55	3.50
13	2	18.46	0.00
14	1	27.47	0.00
15	2	138.70	6.00
16	3	157.47	15.00

Nota. Medias por registro del grupo técnico complementario de participantes adultos. Un guion indica ausencia de registros en el nivel.

Figura 12

Perfil de tiempo y muertes por nivel en la población infantil



Nota. Medias calculadas sobre 211 registros infantiles. El panel superior presenta tiempo; el inferior, muertes totales y muertes influidas por el enemigo. Cada nivel conserva un n diferente, informado en las tablas anteriores.

Lectura integrada de la progresión por bloques de niveles

Los niveles 1 a 4 forman el bloque de incorporación. El nivel 1 concentra un tiempo medio de 95.62 segundos, 9.41 intentos y 8.27 muertes en la población infantil, valores superiores a los de los tres niveles siguientes. La disminución observada después del primer escenario es compatible con familiarización con controles, cámara y relación entre entrada y giro. No se atribuye exclusivamente a aprendizaje, porque la geometría cambia entre escenas y los participantes no aportaron la misma cantidad de registros. El valor principal del bloque consiste en mostrar que la introducción requiere apoyo y que el tutorial debe anticipar el primer contacto con la ruta real.

En ese primer bloque, el nivel 4 registra 2.00 intentos, 0.94 muertes y 15.11 recolectables en niños. La reducción de mortalidad junto con una recolección alta indica que fue posible atender a objetivos secundarios cuando el trazado ofrecía una condición más estable. Este contraste interno confirma que tiempo, muerte y recolección deben leerse de manera conjunta. Un nivel breve no es necesariamente más comprensible, y un nivel con pocos eventos del enemigo puede seguir exigiendo orientación o precisión.

Los niveles 5 a 8 constituyen un bloque intermedio con cobertura infantil variable. Los tiempos medios se mantienen entre 39.00 y 66.59 segundos; el nivel 6 presenta 6.40 intentos, 4.73 muertes y 1.67 muertes influidas por el enemigo. El nivel 7, en cambio, desciende a 2.58 intentos y 1.58 muertes. La alternancia muestra que la progresión no fue un aumento aritmético de dificultad. Cada escena combina longitud, cambios de dirección, distribución de objetos y presión del agente, de modo que un escenario posterior puede resultar operacionalmente más estable.

En los niveles 9 a 12 se observa una segunda concentración de demanda. El nivel 10 alcanza 103.14 segundos en promedio infantil, aunque con seis registros, mientras el nivel 12 alcanza 10.60 intentos, 8.70 muertes y 3.30 muertes influidas por el enemigo en diez registros. El nivel 12 constituye el punto de mayor presión media del agente en la serie infantil. El resultado respalda que la persecución participa en el reto, pero las muertes totales son mayores que las atribuidas al agente; por ello, plataforma, decisiones de dirección y otros obstáculos conservan una contribución importante.

Los niveles 13 y 14 muestran tiempos medios cercanos a 55 segundos, con 5.10 y 5.92 intentos, respectivamente. La mortalidad y la influencia del enemigo se mantienen por debajo del pico del nivel 12. Esta transición resulta relevante para la narrativa de progresión: después de un escenario exigente aparecen condiciones que permiten consolidar movimiento antes de introducir la acción adicional de salto. El patrón sugiere que la secuencia puede conservar variedad sin exigir que cada nivel supere todos los indicadores del anterior.

Los niveles 15 y 16 integran JumpPads y representan el bloque avanzado verificado. En niños, el tiempo aumenta de 75.95 a 68.32 segundos y la recolección desciende a 9.25 y 8.07. El nivel 16 registra 6.13 intentos, 5.13 muertes y 2.47 muertes influidas por el enemigo. La acción de salto añade una decisión que compite con orientación, recolección y vigilancia del perseguidor. La menor recolección es coherente con una redistribución de atención dentro del juego, aunque no permite identificar por sí sola cuál proceso individual produjo el cambio.

La cobertura adulta es menor en todos los bloques y está ausente en el nivel 10. En el nivel 16, tres participantes adultos generan un promedio de 157.47 segundos, 16.00 intentos y 15.00 muertes. Este resultado no se usa como referencia normativa ni para inferir superioridad de un grupo; muestra que edad adulta no elimina automáticamente las dificultades de mecánica, familiaridad o trazado. Su utilidad técnica consiste en revelar que la operación avanzada puede producir ejecuciones extensas también fuera de la población infantil.

La lectura por bloques orienta acciones de mejora. El inicio requiere reforzar la transición tutorial–nivel 1; el nivel 6 y especialmente el 12 requieren revisar equilibrio entre trazado y presión del agente; los niveles 15 y 16 requieren anticipar visualmente el salto y su oportunidad. Estas prioridades derivan de la convergencia entre datos por nivel, observaciones abiertas sobre controles y el diseño implementado. No suponen modificar los resultados obtenidos, sino convertir la evaluación en decisiones mantenibles.

También se desprende una implicación para futuras mediciones. El sistema debería conservar cada intento, registrar resultado final, plataforma y modo de steering por evento. Con esa granularidad sería posible separar longitud del nivel, experiencia acumulada, intervención del enemigo y acción de salto. En la versión evaluada, el archivo participante–nivel resume el último estado conservado, por lo que la interpretación permanece descriptiva y se acompaña del n de cada escenario.

Resultados de la encuesta por ítem

La media global de los 15 ítems fue 4.28 en la población infantil y 4.57 en el grupo técnico adulto. En los niños, diversión alcanzó 4.48, deseo de continuar 4.45 e intención de volver a jugar 4.36. El enemigo como elemento interesante y desafiante obtuvo 4.32, lo que respalda su función en la experiencia.

Las medias infantiles más bajas fueron aprendizaje de controles (4.02), comprensión de la tarea (4.07) y concentración requerida (4.09). Aunque siguen por encima del punto neutral, presentan mayor dispersión y menor porcentaje favorable. Este patrón coincide con comentarios sobre orientación e inversión de controles.

El grupo técnico adulto valoró especialmente respuesta de controles, reconocimiento de objetos y comprensión del resultado, con medias de 4.83. Debido a n = 6, una respuesta representa 16.7 puntos porcentuales; las diferencias deben leerse como descripción.

Tabla 40

Resultados de encuesta por ítem, ítems 1 a 8

Ítem	Contenido	Media infantil	Fav. %	Media adulta	Fav. %
1	Comprensión de la tarea	4.07	72.7	4.50	100.0
2	Aprendizaje de controles	4.02	72.7	4.17	83.3
3	Respuesta de controles	4.39	81.8	4.83	100.0
4	Claridad visual	4.32	81.8	4.33	83.3
5	Utilidad del tutorial	4.39	77.3	4.50	83.3
6	Diversión	4.48	88.6	4.67	100.0
7	Deseo de continuar	4.45	88.6	4.67	100.0
8	Dificultad adecuada	4.18	75.0	4.17	83.3

Nota. Población infantil n = 44; grupo técnico adulto n = 6. Favorable corresponde a respuestas 4 o 5 en la escala de 1 a 5.

Tabla 41

Resultados de encuesta por ítem, ítems 9 a 15

Ítem	Contenido	Media infantil	Fav. %	Media adulta	Fav. %
9	Concentración requerida	4.09	75.0	4.67	100.0

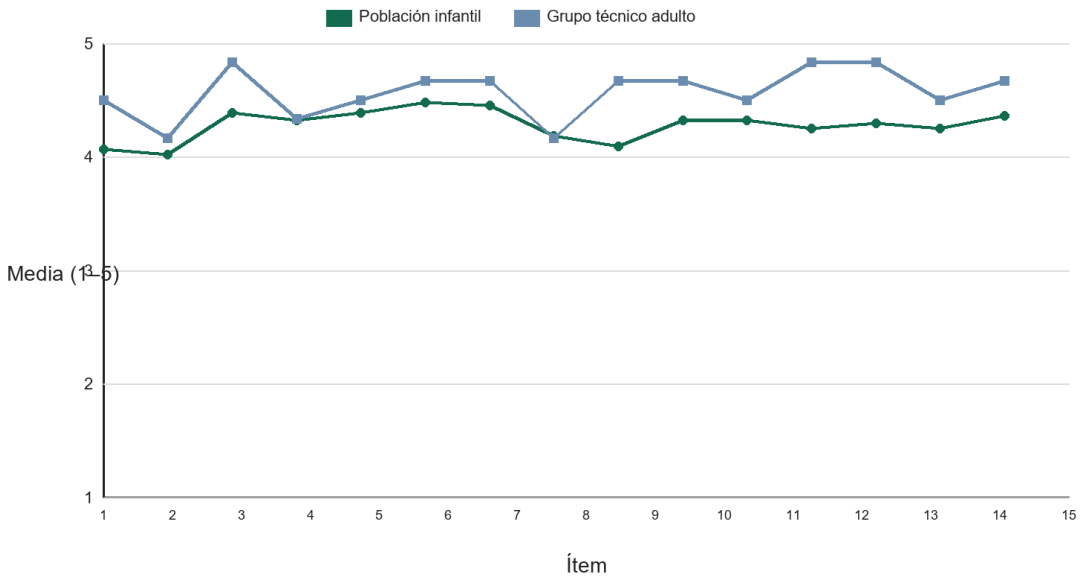
Ítem	Contenido	Media infantil	Fav. %	Media adulta	Fav. %
10	Decisión rápida	4.32	84.1	4.67	100.0
11	Aporte del enemigo al reto	4.32	86.4	4.50	83.3
12	Reconocimiento de objetos	4.25	81.8	4.83	100.0
13	Comprensión del resultado	4.30	81.8	4.83	100.0
14	Utilidad del resumen	4.25	75.0	4.50	83.3
15	Intención de volver a jugar	4.36	84.1	4.67	100.0

Nota. Población infantil n = 44; grupo técnico adulto n = 6. Las medias resumen percepción y no constituyen una prueba objetiva de funciones ejecutivas.

La Figura 13 permite comparar el perfil de los 15 ítems.

Figura 13

Valoración media de los 15 ítems por grupo



Nota. Medias de escala 1–5. La población infantil aporta 44 respuestas y el grupo técnico adulto 6. Los ítems se conservan en el orden presentado en las dos tablas anteriores.

Coherencia entre contenido, polaridad y comportamiento de respuesta

Los 15 ítems están redactados en polaridad positiva: una puntuación alta representa una valoración favorable de la característica enunciada. Esta uniformidad facilita la comprensión para participantes infantiles y evita invertir respuestas durante el cálculo. También

puede favorecer patrones de asentimiento; por ello, el análisis no depende únicamente de la media total y examina cada ítem, los porcentajes favorables y las respuestas abiertas.

Los ítems 1 a 5 forman un bloque coherente de incorporación y control. Comprensión de la tarea, aprendizaje, respuesta, claridad y utilidad del tutorial se refieren a momentos próximos de la experiencia, pero no son equivalentes. La respuesta de controles obtuvo 4.39 en niños, mientras aprendizaje alcanzó 4.02. El contraste indica que el sistema podía responder correctamente, aunque la regla demandara familiarización. Esta diferencia coincide con comentarios sobre orientación y justifica separar desempeño técnico de facilidad percibida.

Los ítems 6, 7 y 15 describen experiencia y motivación: diversión, deseo de continuar e intención de volver a jugar. Sus medias infantiles entre 4.36 y 4.48 y sus porcentajes favorables entre 84.1 % y 88.6 % forman un patrón estable. La consistencia no demuestra aprendizaje, pero sí apoya que el producto mantuvo interés durante la sesión evaluada. La intención declarada se interpreta como percepción inmediata y no como conducta futura observada.

Los ítems 8 a 11 abordan reto y demanda percibida. Dificultad adecuada, concentración y decisión rápida describen la experiencia del jugador; el aporte del enemigo identifica un elemento de diseño. Sus medias infantiles oscilan entre 4.09 y 4.32. La concentración presenta la media menor del bloque, mientras el enemigo alcanza 86.4 % de respuestas favorables. Esta combinación muestra que el reto fue reconocido y generalmente valorado, aunque algunos participantes encontraron exigente la coordinación de acciones.

Los ítems 12 a 14 se relacionan con reconocimiento de objetos y comprensión de resultados. Las medias infantiles de 4.25 a 4.30 indican percepción favorable de la retroalimentación. El contenido de estos ítems conecta con HUD, objetos, pantalla de resultado y resumen posterior. La evidencia permite sostener claridad percibida de esos elementos, pero no verificar cuánto recuerda cada participante después de la sesión.

La revisión conjunta de polaridad, contenido y respuesta conserva el vínculo con los objetivos de evaluación. Los ítems de interacción informan mejoras de tutorial y controles; los

de motivación describen aceptabilidad; los de reto se relacionan con mecánicas e IA; los de retroalimentación examinan comprensión del estado. La lectura por dimensión organiza estas relaciones sin ocultar la información individual que resulta necesaria para decidir cambios del producto.

Dimensiones y consistencia interna

La dimensión infantil con mayor media fue experiencia y motivación (4.43), seguida de retroalimentación (4.27), reto y demanda percibida (4.23) y usabilidad e interacción (4.24). Las diferencias son pequeñas y el perfil general es favorable.

La consistencia interna infantil fue .945 para usabilidad e interacción, .948 para experiencia y motivación, .905 para reto y demanda percibida y .928 para retroalimentación. Los valores altos indican respuestas covariantes dentro de las agrupaciones. No demuestran que la escala mida un único constructo ni que sea válida para diagnóstico.

El patrón de respuesta del primer participante fue bajo en la mayoría de ítems y contribuye a la covariación. La confiabilidad debe leerse junto con contenido, número de ítems y tamaño muestral. No se utiliza un alfa global para sostener una puntuación total homogénea.

Tabla 42

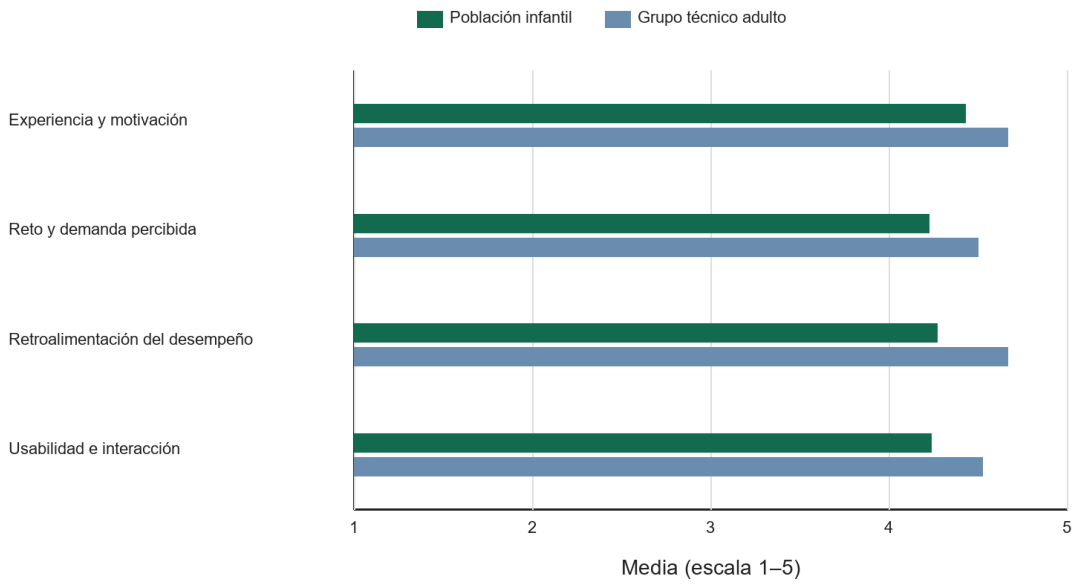
Resultados por dimensión y confiabilidad

Dimensión	Ítems	Media infantil	α infantil	Media adulta
Usabilidad e interacción	1, 2, 3, 4, 5, 12	4.24	0.945	4.53
Experiencia y motivación	6, 7, 15	4.43	0.948	4.67
Reto y demanda percibida	8, 9, 10, 11	4.23	0.905	4.50
Retroalimentación del desempeño	13, 14	4.27	0.928	4.67

Nota. Alfa de Cronbach se calcula para la población infantil. No se informa alfa del grupo técnico adulto por su tamaño reducido. Las dimensiones son agrupaciones teóricas de contenido y no una validación factorial.

Figura 14

Medias de las dimensiones de experiencia por grupo



Nota. Las dimensiones se calcularon como media de los ítems definidos en la tabla de correspondencia ítem–dimensión. La escala va de 1 a 5. El grupo adulto se presenta únicamente como referencia técnica descriptiva.

Resultados de las respuestas abiertas

Las respuestas abiertas se codificaron con categorías no excluyentes. En la población infantil, las preferencias se concentraron en progresión, reto y diversión (21 menciones), controles y mecánicas (16), diseño visual y escenarios (13), persecución (5) y recolección o exploración (2). Los participantes valoraron tanto la estructura del reto como la forma particular de control.

La dificultad más repetida fue controles y orientación (28 menciones), seguida de obstáculos, salto o niveles avanzados (13) y reacción o ritmo (7). Este resultado explica por qué comprensión y aprendizaje de controles quedaron entre las medias más bajas, aun cuando la valoración general fue favorable.

En las propuestas de cambio infantiles aparecieron controles, guía y claridad (14), contenido y presentación (14), rendimiento o correcciones (9), ajuste del reto o perseguidor (4) y ausencia de cambios importantes (10). La coexistencia de solicitudes opuestas —enemigo

más rápido o más lento— indica diversidad de habilidad y apoya una futura selección de dificultad, no una única velocidad universal.

En el grupo técnico adulto destacaron contenido y presentación, mecánicas y correcciones de rendimiento. Su función complementaria permitió identificar aspectos técnicos sin mezclar estas respuestas con la experiencia infantil.

Tabla 43

Síntesis temática de respuestas abiertas

Pregunta	Tema	Infantil	Adulto técnico	Lectura
Lo que gustó	Progresión, reto y diversión	21	2	Participación favorable
Lo que gustó	Controles y mecánicas	16	2	Identidad de gameplay
Lo que gustó	Diseño y escenarios	13	1	Aporte estético
Dificultad	Controles y orientación	28	3	Principal fricción
Dificultad	Obstáculos, salto o niveles	13	2	Demanda avanzada
Dificultad	Reacción y ritmo	7	1	Presión temporal
Cambios	Controles, guía y claridad	14	0	Refuerzo del tutorial
Cambios	Contenido y presentación	14	3	Variedad y feedback
Cambios	Rendimiento o correcciones	9	1	Estabilidad percibida
Cambios	Sin cambios importantes	10	2	Satisfacción declarada

Nota. Frecuencias de menciones en tres preguntas abiertas. La codificación es no excluyente; una respuesta puede aportar a más de un tema, por lo que las frecuencias no deben sumarse como participantes únicos.

Resultados de pruebas de caja negra

Los flujos de registro, interacción y persistencia produjeron 50 perfiles vinculados, 50 encuestas y 248 registros de nivel. Los 16 niveles aparecen en la telemetría. Hubo 201 registros con al menos una muerte y 156 con al menos una muerte influida por el enemigo, lo que aporta evidencia de ejecución de estados de derrota y persecución.

La recolección aparece en todos los registros y ningún valor supera el total disponible. El uso declarado incluyó 33 participantes en Windows y 17 en Android. La valoración de respuesta de controles fue 4.39 en niños y 4.83 en adultos, y la comprensión del resultado 4.30 y 4.83, respectivamente.

Los resultados confirman ejecución de los flujos esenciales. No se infiere que todos los caminos internos hayan sido cubiertos por cada usuario; la evidencia de campo se complementa con verificación estructural.

Tabla 44

Resultados ejecutados de caja negra

Caso	Evidencia	Resultado	Interpretación
CN-01 Registro	50 perfiles vinculados	Conforme	Identidad pseudonimizada disponible
CN-02 Tutorial	Ítem 5: 4.39 infantil	Conforme	Ayuda percibida favorable
CN-03 Movimiento	248 registros con entradas	Conforme	Gameplay ejecutado
CN-04 Recolección	248 valores dentro de 0–18	Conforme	Contador consistente
CN-05 Caída/derrota	201 registros con muertes	Conforme	Estado de derrota ejecutado
CN-06 Persecución	156 registros con influencia	Conforme	Agente produjo presión observable
CN-08 Persistencia	248 JSON legibles	Conforme	Guardado por nivel
CN-09 Windows	33 participantes	Conforme	Operación reportada
CN-10 Android	17 participantes	Conforme	Operación reportada

Nota. Los resultados combinan telemetría y encuesta.

Resultados de caja blanca y compilación

La compilación por lotes del proyecto en Unity 2022.3.62f3 finalizó con código de salida 0 y sin errores de compilación. Este resultado verifica sintaxis, referencias y dependencias del conjunto de scripts; no equivale a una cobertura automática de escenarios en PlayMode.

La inspección estructural confirmó guardas para procesar derrota y guardado una vez, selección de Seek/Pursuit, límites de predicción, validación de suelo y segmento, historial de trayectoria, salto autónomo, respawn, desbloqueo y restablecimiento. La existencia de datos completos y eventos del enemigo es coherente con la ejecución de varios de estos caminos.

Los casos que dependen de geometrías específicas se valoraron por su condición y retorno seguro. Cuando un objetivo predictivo no dispone de suelo o el segmento no es válido, el algoritmo reduce la predicción, sigue la trayectoria o usa Seek. Este comportamiento evita que Pursuit tenga prioridad sobre la transitabilidad.

Tabla 45

Resultados de verificación de caja blanca

Ruta	Evidencia estructural	Resultado	Alcance
Guardado único	Banderas de estado	Conforme	Evita doble procesamiento
Selección Pursuit	Velocidad, distancia e histéresis	Conforme	Predicción condicionada
Retorno Seek	Validaciones y fallback	Conforme	Objetivo seguro
Suelo y segmento	Consultas físicas y muestreo	Conforme	Descarta vacío
Trayectoria	Muestras y distancia acumulada	Conforme	Respeta ruta
Salto autónomo	Aterrizaje y cooldown	Conforme	Supera discontinuidades válidas
Respawn	Umbral y posición segura	Conforme	Recuperación
Compilación	Unity batch, salida 0	Conforme	Sin errores de scripts

Nota. La verificación de caja blanca combina estructura del código y compilación.

Resultados de Seek, Pursuit y navegación segura

La implementación contiene dos comportamientos diferenciados. Seek usa el objetivo actual o una proyección segura; Pursuit calcula un objetivo futuro con velocidad y tiempo acotados. La decisión híbrida conserva Pursuit solo cuando las condiciones son válidas. Esta separación satisface la especificación técnica del agente.

La navegación segura agrega valor respecto de una persecución puramente geométrica. El agente comprueba suelo del jugador y del candidato, tolerancia de altura y

transitabilidad del segmento. El historial permite seguir el recorrido sobre esquinas. El salto y el respawn controlan discontinuidades. Los 277 eventos de influencia del enemigo acumulados en los registros infantiles muestran que el agente intervino en la dificultad, mientras la encuesta infantil de aporte del enemigo alcanzó 4.32.

La telemetría no registra en qué cuadros estuvo activo cada modo. Por ello, la efectividad de Pursuit se sostiene mediante funcionamiento estructural, compilación, eventos del agente y percepción del reto; no mediante una comparación cuantitativa Seek versus Pursuit.

Resultados por plataforma

Windows concentró 66 % de participantes y Android 34 %. Ambos grupos de edad utilizaron las dos plataformas. La disponibilidad de respuestas y telemetría indica que el flujo esencial pudo completarse en ambos contextos.

La encuesta no fue diseñada para comparar plataformas y el JSON no guarda esa variable. Por ello, no se atribuyen diferencias de desempeño a Windows o Android. La evidencia sustenta compatibilidad funcional básica y señala la necesidad de incluir plataforma, dispositivo y versión en futuras telemetrías.

Evaluación según ISO/IEC 25010

La adecuación funcional se respalda con 16 niveles, requisitos implementados y flujos de campo. La capacidad de interacción se respalda con medias favorables y comentarios específicos. La fiabilidad se respalda con 248 JSON legibles y completos. La compatibilidad y flexibilidad se respaldan con sesiones en Windows y Android.

La mantenibilidad se apoya en componentes identificables y compilación sin errores, aunque clases centrales extensas pueden beneficiarse de mayor separación. La eficiencia se describe mediante tiempos de interacción, no mediante un benchmark de FPS o consumo. La

seguridad se limita al almacenamiento local y pseudonimización; no se realizaron pruebas de penetración.

El resultado es suficiente para evaluar el producto dentro del alcance académico, pero cada característica conserva su límite. Esta forma de reporte evita convertir un marco de calidad en una calificación única sin medidas.

Tabla 46

Resultados del producto según características de calidad

Característica	Resultado disponible	Valoración	Límite
Adecuación funcional	16 niveles y flujos esenciales	Evidencia favorable	Sin matriz exhaustiva de todas las combinaciones
Eficiencia	Tiempos y operación de campo	Evidencia descriptiva	Sin FPS, memoria o energía
Compatibilidad	33 Windows; 17 Android	Evidencia favorable	Sin comparación por dispositivo
Capacidad de interacción	Media infantil 4.28/5	Evidencia favorable	Percepción posterior
Fiabilidad	248 JSON completos	Evidencia favorable	Un archivo conservado por nivel
Seguridad	Persistencia local	Control básico	Sin prueba de penetración
Mantenibilidad	Componentes y compilación	Evidencia favorable	Clases centrales extensas
Flexibilidad	Núcleo compartido en dos plataformas	Evidencia favorable	No se evaluaron otros sistemas
Seguridad operacional	Reintento, salto y respawn	Evidencia funcional	No es software de seguridad crítica

Nota. La valoración se fundamenta en datos, pruebas y arquitectura.

Resultados según el Modelo Tetrad Modificado

En Estética, la claridad visual obtuvo 4.32, la utilidad del tutorial 4.39 y la respuesta de controles 4.39 en la población infantil. La experiencia y motivación alcanzó 4.43. Los comentarios favorables sobre diseño y escenarios respaldan el nodo, mientras la fricción de controles orienta su refinamiento.

En Historia, la progresión, el deseo de continuar y la intención de volver a jugar alcanzaron 4.45 y 4.36. El flujo funcional produjo una secuencia comprensible sin depender de una narrativa extensa. En Tecnología, Windows y Android generaron sesiones completas y la

persistencia produjo archivos íntegros. En Mecánicas, los 16 niveles, las muertes, la recolección, el salto y la influencia del enemigo muestran ejecución de reglas e IA.

La lectura integrada confirma la utilidad del Tetrad: una dificultad de controles no pertenece únicamente a interfaz, porque afecta percepción estética, progresión, reglas y tecnología de entrada. De forma similar, la persecución influye en reto, historia funcional, mecánica y calidad.

Discusión

Los resultados muestran una experiencia favorable y un reto real. Diversión, continuidad e intención de volver a jugar se sitúan entre las valoraciones más altas, mientras controles y orientación concentran dificultades. Este patrón es coherente con un juego cuya identidad depende de una asignación de acciones poco convencional: la misma mecánica que genera reto puede elevar la curva de aprendizaje.

La influencia del enemigo aparece en telemetría y encuesta. El agente no explica todas las muertes, lo cual es consistente con un sistema donde caída, obstáculos y persecución coexisten. El diseño híbrido Seek/Pursuit aporta anticipación sin renunciar a seguridad. Este resultado se relaciona con Reynolds (1999) y Millington (2019), pero añade controles de terreno específicos para plataformas suspendidas.

Frente a Parong et al. (2017), Zeng et al. (2023) y Kou et al. (2024), el estudio comparte el uso de juegos que plantean demandas ejecutivas, pero difiere en diseño y medida. Aquellos trabajos utilizaron comparadores, pre/post o síntesis de ensayos. Zig-Zag describe interacción y percepción después del desarrollo. En consecuencia, los resultados apoyan viabilidad, aceptabilidad y funcionamiento, no transferencia cognitiva.

Los antecedentes ecuatorianos muestran que Unity y las experiencias 3D se aplican a biodiversidad, reciclaje, salud y patrimonio. Zig-Zag aporta un problema distinto: navegación

autónoma segura sobre una pista discontinua, telemetría por nivel y evaluación separada por grupo. La coincidencia está en la necesidad de integrar diseño, tecnología y evaluación local.

La alta consistencia de las dimensiones indica respuestas relacionadas, pero no elimina la necesidad de examinar contenido. La experiencia/motivación y el reto muestran valores altos, mientras la usabilidad contiene los ítems más bajos. Esta diferenciación es más informativa que una puntuación global única.

La cobertura por nivel revela que dificultad no aumenta de forma lineal. Los niveles 1 y 12 pueden demandar muchos intentos por razones diferentes: aprendizaje inicial y complejidad posterior. Los niveles 15 y 16 agregan salto y muestran menor recolección, lo que sugiere competencia entre atención a ruta, obstáculos, objetos y enemigo.

Según el Tetrad, los resultados se explican mejor como relaciones. Una mejora del tutorial modifica Estética y Tecnología; una selección de dificultad modifica Historia y Mecánicas; un registro de modo IA mejora Tecnología y evaluación de Mecánicas. El modelo, por tanto, no solo organizó el diseño, sino también la discusión.

Implicaciones para diseño, uso educativo y evaluación futura

La primera implicación concierne a la curva de aprendizaje. Los controles constituyen parte de la identidad del juego, pero fueron la dificultad cualitativa más repetida y los ítems de comprensión y aprendizaje obtuvieron medias menores que diversión o continuidad. La mejora prioritaria no consiste necesariamente en eliminar el reto, sino en hacerlo legible. Un tutorial escalonado puede presentar dirección, caída, recolección, persecución y salto en momentos separados, solicitar una demostración breve y ofrecer repetición voluntaria antes del primer nivel.

La segunda implicación se relaciona con dificultad. Los datos muestran picos en escenarios específicos, no una pendiente uniforme. La calibración debe considerar longitud, cambios de dirección, ubicación de recolectables, velocidad del agente y presencia de

obstáculos. Un panel de parámetros por escena facilitaría ajustar estos factores sin modificar la lógica central. Antes de aplicar nuevos valores, conviene definir criterios observables —tiempo esperado, mortalidad y porcentaje de finalización— y probarlos con participantes del rango objetivo.

La tercera implicación corresponde al perseguidor. El híbrido Seek/Pursuit consiguió intervenir en el reto sin explicar todas las muertes. Para profundizar la evaluación, la telemetría debe registrar el modo activo, distancia al jugador, objetivo previsto, validación de terreno y causa del respawn. Con esos eventos puede compararse la conducta del mismo algoritmo bajo configuraciones planificadas. Una evaluación controlada de IA necesitaría asignar condiciones equivalentes —por ejemplo, Seek solo frente a Seek/Pursuit— y conservar una medida de resultado común.

En uso educativo, el videojuego debe presentarse como una actividad interactiva con demandas de atención, decisión y adaptación, no como tratamiento ni diagnóstico. El acompañante puede observar estrategias, pedir que el niño explique una decisión y utilizar la retroalimentación de la partida para conversar sobre planificación y corrección del error. Cualquier afirmación de mejora fuera del juego requerirá instrumentos apropiados, medición temporal y un protocolo aprobado para esa finalidad.

La cuarta implicación aborda inclusión. La población prevista cubre edades de 7 a 12 años. Las instrucciones, tamaño de objetivos, duración de sesiones y precisión exigida deben probarse con el tramo inferior antes de generalizar. También se requieren verificaciones específicas de contraste, lectura, audición, motricidad y diversidad educativa. La arquitectura de controles separada favorece incorporar alternativas, pero la disponibilidad técnica no demuestra accesibilidad hasta ser evaluada con usuarios pertinentes.

La quinta implicación se refiere a la encuesta. La revisión del tutor respaldó claridad, pertinencia y correspondencia con objetivos, y las dimensiones presentaron consistencia interna alta en la muestra infantil. Aun así, futuras aplicaciones deben conservar la

identificación de cada ítem, revisar posibles redundancias y estudiar estructura con una muestra suficiente. La escala puede acompañarse de preguntas cognitivas apropiadas al desarrollo para comprender por qué un participante eligió una respuesta, sin inducir valoraciones favorables.

En ingeniería, la mantenibilidad se beneficia de conservar la relación entre requisito, componente, prueba y métrica. Los cambios de escena deben registrar qué parámetro fue modificado y qué criterio motivó la decisión. Las pruebas automatizadas pueden cubrir serialización, cálculo de porcentajes, desbloqueo y reglas de puntuación; PlayMode puede cubrir cambios de escena, colisiones, respawn y meta. Las pruebas manuales con usuarios seguirán siendo necesarias para claridad, experiencia y adecuación de la dificultad.

Finalmente, una evaluación posterior puede organizarse en dos fases. La primera confirmaría usabilidad y estabilidad con representación de todas las edades previstas, registros de sesión completos y criterios de desempeño definidos. La segunda, solo después de esa confirmación, podría estudiar efectos mediante pretest, postest, comparador, asignación justificada y medidas válidas de funciones ejecutivas. Esta secuencia conserva la contribución actual: un producto implementado y evaluado en su funcionamiento y experiencia, con una base técnica clara para preguntas de mayor alcance.

Limitaciones

La muestra fue no probabilística y la cobertura etaria efectiva restringe la generalización a todo el público previsto. El grupo técnico adulto fue pequeño y se usó únicamente como apoyo de operación. La cobertura de niveles fue desigual y cada participante aportó un número distinto de registros.

El diseño no incluyó pretest, postest, asignación aleatoria ni condiciones controladas de dificultad o IA. La encuesta fue revisada por contenido, pero no cuenta con una validación

psicométrica completa. Los valores de alfa no sustituyen evidencia de estructura factorial, validez convergente o estabilidad.

La telemetría no almacena resultado final, plataforma, dispositivo ni modo Seek/Pursuit por evento. Un archivo por participante–nivel puede reemplazar ejecuciones previas. No se midieron FPS, memoria, consumo energético ni cobertura automatizada de PlayMode. Estas limitaciones acotan las conclusiones sin anular la evidencia funcional y de experiencia obtenida.

Capítulo V

Conclusiones

Se diseñó, desarrolló, implementó y evaluó un videojuego educativo 3D tipo Zig-Zag en Unity 2022.3.62f3, organizado en un flujo de acceso, tutorial, menú y dieciséis escenas de juego, con compilaciones para Windows y Android. La solución integra movimiento, salto, obstáculos, recolectables, retroalimentación, progreso local y un agente autónomo cuya persecución alterna Seek y Pursuit bajo validación del terreno, seguimiento de trayectoria, salto autónomo y recuperación. La evaluación reunió telemetría local, encuesta de experiencia y pruebas funcionales y estructurales, lo que permitió comprobar la operación del producto y describir su comportamiento en la población infantil evaluada y en el grupo técnico complementario. Por tanto, el trabajo alcanzó el alcance de ingeniería definido: existe un producto funcional, trazable y documentado, con evidencia de calidad y uso. Este resultado no demuestra por sí mismo una mejora causal de funciones ejecutivas; la muestra disponible, el tiempo de exposición y la ausencia de un diseño experimental controlado limitan la interpretación a viabilidad técnica, interacción observada e indicadores de experiencia.

La fundamentación teórica de la propuesta relacionó simulación 3D, diseño de interacción infantil, funciones ejecutivas, aprendizaje basado en juego, modelo Tetrad modificado y navegación autónoma. Esta revisión orientó decisiones concretas: controles breves, progresión por niveles, retroalimentación inmediata, separación entre indicadores de

interacción y resultados clínicos, y combinación de Seek/Pursuit con restricciones de seguridad sobre plataformas suspendidas. La correspondencia entre teoría y producto permitió justificar por qué el videojuego plantea demandas de atención, inhibición, planificación y flexibilidad sin afirmar que las mide de forma diagnóstica ni que produce efectos terapéuticos. El grado de cumplimiento es satisfactorio para sustentar el diseño; como límite, la eficacia educativa de tales decisiones requiere estudios posteriores con exposición longitudinal, comparación y medidas externas validadas.

La especificación del sistema transformó el problema en requisitos funcionales y no funcionales, criterios de aceptación, historias de usuario, casos de uso y una matriz de trazabilidad que conecta objetivos, componentes y pruebas. La especificación delimita registro, tutorial, menú, niveles, controles, resultados, persistencia y comportamiento del agente, y distingue la configuración de persecución de una supuesta dificultad adaptativa que no está implementada. Los modelos integrados de acceso, clases e interfaces se contrastaron con escenas, métodos y archivos vigentes, por lo que reducen ambigüedades entre la documentación y el software. La especificación ofrece una base suficiente para verificación y mantenimiento en el alcance actual; deberá actualizarse en la misma versión del código cada vez que cambien escenas, entradas, esquema de datos o parámetros de IA.

La implementación del software se ejecutó mediante un proceso iterativo e incremental que integró motor físico, inteligencia artificial y gestión de datos en componentes con responsabilidades identificables. Player concentra entrada, colisiones, HUD y cierre del nivel; AIEnemy administra steering, predicción, validación de terreno, seguimiento de trayectoria, salto y reaparición; MetricsManager y DataPersistenceManager separan el registro JSON por nivel del resumen CSV de sesión; y MenuSceneLoader, LevelProgression y los controladores de interfaz gestionan acceso y continuidad. La evidencia de escenas habilitadas y artefactos de compilación confirma la materialización de los incrementos en Windows y Android. El cumplimiento es funcional dentro de la arquitectura implementada, aunque la mantenibilidad

futura mejoraría con pruebas automatizadas de regresión, integración continua y una reducción adicional de responsabilidades concentradas en Player y MenuSceneLoader.

La evaluación del producto se ejecutó mediante pruebas de caja negra, revisión de caja blanca, verificación de compilaciones, análisis de telemetría y valoración con características seleccionadas de ISO/IEC 25010. La inspección de Seek/Pursuit corroboró que la anticipación se aplica solo cuando existen condiciones válidas y que la seguridad del terreno, la trayectoria registrada, el salto autónomo y la recuperación limitan el movimiento del enemigo. Los resultados de uso permitieron comparar cobertura, tiempo, muertes, entradas y percepciones entre grupos sin confundir esos indicadores con una prueba clínica o causal de efectividad cognitiva. En consecuencia, se comprobó el desempeño operativo y se obtuvieron evidencias iniciales de interacción y calidad; permanecen como limitaciones la muestra no probabilística, la falta de un grupo de comparación, la ausencia de cobertura automatizada cuantificada y la necesidad de ensayos prolongados en contexto educativo.

Recomendaciones

Se recomienda realizar una evaluación longitudinal con niños de 7 a 12 años, grupo de comparación, criterios de inclusión explícitos y medidas externas validadas antes de atribuir al videojuego efectos sobre funciones ejecutivas. El protocolo debería registrar tiempo de exposición, experiencia previa, condiciones de aplicación y posibles pérdidas de seguimiento, además de conservar el consentimiento y la protección de datos exigibles para población infantil.

Conviene ampliar y estratificar la muestra por edad, plataforma y familiaridad con videojuegos, manteniendo separado cualquier grupo técnico adulto utilizado para encontrar defectos de interacción. Esta decisión permitiría estimar si las diferencias de tiempo, muertes o valoración responden a la edad, al dispositivo, al aprendizaje entre niveles o a características del trazado, sin generalizar a partir de promedios agregados.

En una siguiente versión se debería automatizar la regresión de navegación, progresión y persistencia mediante Unity Test Framework. Las pruebas prioritarias incluyen la validez de puntos Pursuit, la respuesta ante ausencia de suelo, la recuperación del enemigo, la transición entre Scene1 y Scene16, el bloqueo y desbloqueo de niveles, la serialización JSON y CSV y la reapertura de datos en Windows y Android; una canalización de compilación ayudaría a detectar cambios incompatibles antes de distribuir el producto.

Se recomienda versionar el esquema de telemetría y documentar para cada campo su unidad, evento de origen, denominador y tratamiento de valores ausentes. La recolección debe conservar seudonimización, almacenamiento mínimo y reglas de retención; si en el futuro se incorpora sincronización remota, esta no debería añadirse sin un modelo de consentimiento, seguridad, control de acceso y eliminación adecuado para datos de menores.

La persecución actual debe describirse como un comportamiento configurable, no como dificultad adaptativa. Si se desea adaptar el reto en trabajos futuros, será necesario diseñar y validar una política explícita que utilice indicadores estables, establezca límites de seguridad y evite aumentar simultáneamente velocidad, anticipación y penalización. La nueva política tendría que compararse con la configuración fija mediante pruebas controladas y métricas de experiencia, abandono y desempeño.

Referencias

Arce Maldonado, E. P., & Chabla Muñoz, A. P. (2026). Desarrollo de un sistema educativo basado en dinámicas gamificadas para la enseñanza de la cultura Cañari con generación automática de narrativas usando un modelo multimodal [Trabajo de titulación, Universidad Politécnica Salesiana]. https://dspace.ups.edu.ec/handle/123456789/32028

Beck, K., & Andres, C. (2004). Extreme programming explained: Embrace change (2nd ed.). Addison-Wesley.

Boateng, G. O., Neilands, T. B., Frongillo, E. A., Melgar-Quiñonez, H. R., & Young, S. L. (2018). Best practices for developing and validating scales for health, social, and behavioral research: A primer. Frontiers in Public Health, 6, 149. https://doi.org/10.3389/fpubh.2018.00149

Creswell, J. W., & Plano Clark, V. L. (2018). Designing and conducting mixed methods research (3rd ed.). SAGE.

Cronbach, L. J. (1951). Coefficient alpha and the internal structure of tests. Psychometrika, 16(3), 297–334. https://doi.org/10.1007/BF02310555

Daoudi, I. (2022). Learning analytics for enhancing the usability of serious games in formal education: A systematic literature review and research agenda. Education and Information Technologies, 27, 11237–11266. https://doi.org/10.1007/s10639-022-11087-4

Diamond, A. (2013). Executive functions. Annual Review of Psychology, 64, 135–168. https://doi.org/10.1146/annurev-psych-113011-143750

Diamond, A., & Lee, K. (2011). Interventions shown to aid executive function development in children 4 to 12 years old. Science, 333(6045), 959–964. https://doi.org/10.1126/science.1204529

González Campoverde, W. F., & Vintimilla Torres, M. N. (2026). Juego en realidad virtual como herramienta educativa para mejorar los hábitos de higiene bucal en niños y adolescentes [Trabajo de titulación, Universidad Politécnica Salesiana]. https://dspace.ups.edu.ec/handle/123456789/32141

Grey, S., Grey, D., Gordon, N., & Purdy, J. (2017). Using formal game design methods to embed learning outcomes into game mechanics and avoid emergent behaviour. International Journal of Game-Based Learning, 7(3), 63–73.

Hunicke, R., LeBlanc, M., & Zubek, R. (2004). MDA: A formal approach to game design and game research. Proceedings of the AAAI Workshop on Challenges in Game AI, 1–5.

IEEE Computer Society. (1998). IEEE recommended practice for software requirements specifications (IEEE Std 830-1998). https://standards.ieee.org/ieee/830/1222/

International Organization for Standardization, International Electrotechnical Commission, & Institute of Electrical and Electronics Engineers. (2018). Systems and software engineering—Life cycle processes—Requirements engineering (ISO/IEC/IEEE 29148:2018). https://standards.ieee.org/standard/29148-2018.html

International Organization for Standardization. (2023). Systems and software engineering—Systems and software Quality Requirements and Evaluation (SQuaRE)—Product quality model (ISO/IEC 25010:2023). https://www.iso.org/standard/78176.html

Kou, R., Zhang, Z., Zhu, F., Tang, Y., & Li, Z. (2024). Effects of exergaming on executive function and motor ability in children: A systematic review and meta-analysis. PLOS ONE, 19(9), e0309462. https://doi.org/10.1371/journal.pone.0309462

Lehnert, F. K., Niess, J., Lallemand, C., Markopoulos, P., Fischbach, A., & Koenig, V. (2022). Child–computer interaction: From a systematic review towards an integrated understanding of interaction design methods for children. International Journal of Child-Computer Interaction, 32, 100398. https://doi.org/10.1016/j.ijcci.2021.100398

Loja Marín, J. S. (2025). Videojuego 3D del género plataformas educativo para promover el conocimiento de la biodiversidad de la naturaleza ecuatoriana [Trabajo de titulación, Universidad Politécnica Salesiana]. https://dspace.ups.edu.ec/handle/123456789/31175

Millington, I. (2019). AI for games (3rd ed.). CRC Press.

Miyake, A., Friedman, N. P., Emerson, M. J., Witzki, A. H., Howerter, A., & Wager, T. D. (2000). The unity and diversity of executive functions and their contributions to complex frontal lobe tasks: A latent variable analysis. Cognitive Psychology, 41(1), 49–100. https://doi.org/10.1006/cogp.1999.0734

Padilla Hinojosa, E. J. (2026). EcoAventuras: El reto del reciclaje en plástico [Trabajo de titulación, Universidad Politécnica Salesiana]. https://dspace.ups.edu.ec/handle/123456789/32019

Parong, J., Mayer, R. E., Fiorella, L., MacNamara, A., Homer, B. D., & Plass, J. L. (2017). Learning executive function skills by playing focused video games. Contemporary Educational Psychology, 51, 141–151. https://doi.org/10.1016/j.cedpsych.2017.07.002

Reynolds, C. W. (1999). Steering behaviors for autonomous characters. Proceedings of the Game Developers Conference, 763–782. https://www.red3d.com/cwr/papers/1999/gdc99steer.html

Rodríguez Timaná, L. C., Castillo García, J. F., Bastos Filho, T., Ocampo González, A. A., Hincapié Monsalve, N. R., & Valencia Jimenez, N. J. (2024). Use of serious games in interventions of executive functions in neurodiverse children: Systematic review. JMIR Serious Games, 12, e59053. https://doi.org/10.2196/59053

Saiger, M. J., Deterding, S., & Gega, L. (2023). Children and young people's involvement in designing applied games: Scoping review. JMIR Serious Games, 11, e42680. https://doi.org/10.2196/42680

Schell, J. (2019). The art of game design: A book of lenses (3rd ed.). CRC Press. https://doi.org/10.1201/b22101

Schwaber, K., & Sutherland, J. (2020). The Scrum guide: The definitive guide to Scrum. https://scrumguides.org/scrum-guide.html

Tavakol, M., & Dennick, R. (2011). Making sense of Cronbach's alpha. International Journal of Medical Education, 2, 53–55. https://doi.org/10.5116/ijme.4dfb.8dfd

Unity Technologies. (2022a). MonoBehaviour.FixedUpdate. Unity Scripting API 2022.3. https://docs.unity3d.com/2022.3/Documentation/ScriptReference/MonoBehaviour.FixedUpdate.html

Unity Technologies. (2022b). Application.persistentDataPath. Unity Scripting API 2022.3. https://docs.unity3d.com/2022.3/Documentation/ScriptReference/Application-persistentDataPath.html

Unity Technologies. (2022c). JsonUtility. Unity Scripting API 2022.3. https://docs.unity3d.com/2022.3/Documentation/ScriptReference/JsonUtility.html

Villacís, C., Fuertes, W., Santillán, M., Aules, H., Tacuri, A., Zambrano, M., & Salguero, E. (2016). On the development of strategic games based on a semiotic analysis: A case study of an optimized Tic-Tac-Toe. In Proceedings of the 18th International Conference on Enterprise Information Systems (Vol. 1, pp. 425–432). SCITEPRESS. https://doi.org/10.5220/0005772904250432

Zeng, N., Lee, J. E., & Gao, Z. (2023). Effects of home-based exergaming on preschool children's cognition, sedentary behavior, and physical activity: A randomized crossover trial. Brain, Behavior, and Immunity—Integrative, 1, 100002. https://doi.org/10.1016/j.bbii.2023.100002