



UNIVERSIDAD
CATÓLICA
DE CUENCA

UNIVERSIDAD CATÓLICA DE CUENCA

Comunidad Educativa al Servicio del Pueblo

**FACULTAD DE INFORMÁTICA, CIENCIAS DE LA
COMPUTACIÓN E INNOVACIÓN TECNOLÓGICA
CARRERA DE REALIDAD VIRTUAL Y VIDEOJUEGOS**

**Desarrollo de un entorno de Realidad Virtual para la validación de aplicaciones de
aprendizaje lógico-matemático en niños de educación inicial**

**PROYECTO DE INTEGRACIÓN CURRICULAR PREVIO A LA
OBTENCIÓN DEL TÍTULO DE INGENIERO EN REALIDAD
VIRTUAL Y VIDEOJUEGOS**

AUTOR: KAROL GABRIEL BERRÚ JIMÉNEZ

DIRECTOR: ING. JOHN LUIS ESTRADA GUAYTA, M.S.

CUENCA - ECUADOR

2026

DIOS, PATRIA, CULTURA Y DESARROLLO



UNIVERSIDAD CATÓLICA DE CUENCA

Comunidad Educativa al Servicio del Pueblo

**FACULTAD DE INFORMÁTICA, CIENCIAS DE
LA COMPUTACIÓN E INNOVACIÓN
TECNOLÓGICA**

**CARRERA DE REALIDAD VIRTUAL Y
VIDEOJUEGOS**

Desarrollo de un entorno de Realidad Virtual para la validación de aplicaciones de
aprendizaje lógico-matemático en niños de educación inicial

**PROYECTO DE INTEGRACIÓN CURRICULAR PREVIO A LA
OBTENCIÓN DEL TÍTULO DE INGENIERO EN REALIDAD
VIRTUAL Y VIDEOJUEGOS**

AUTOR: KAROL GABRIEL BERRÚ JIMÉNEZ

DIRECTOR: ING. JOHN LUIS ESTRADA GUAYTA, M.S.

CUENCA - ECUADOR

2026

DIOS, PATRIA, CULTURA Y DESARROLLO



Universidad
Católica
de Cuenca

DECLARATORIA DE AUTORÍA Y RESPONSABILIDAD

Declaratoria de Autoría y Responsabilidad

Karol Gabriel Berrú Jimenez portador(a) de la cédula de ciudadanía N° **1900760750**. Declaro ser el autor de la obra: **"Desarrollo de un entorno de Realidad Virtual para la validación de aplicaciones de aprendizaje lógico-matemático en niños de educación inicial."**, sobre la cual me hago responsable sobre las opiniones, versiones e ideas expresadas. Declaro que la misma ha sido elaborada respetando los derechos de propiedad intelectual de terceros y eximo a la Universidad Católica de Cuenca sobre cualquier reclamación que pudiera existir al respecto. Declaro finalmente que mi obra ha sido realizada cumpliendo con todos los requisitos legales, éticos y bioéticos de investigación, que la misma no incumple con la normativa nacional e internacional en el área específica de investigación, sobre la que también me responsabilizo y eximo a la Universidad Católica de Cuenca de toda reclamación al respecto.

Cuenca, **31 de agosto de 2026**

F:

Karol Gabriel Berrú Jimenez

C.I. 1900760750



Universidad
Católica
de Cuenca

CERTIFICADO

Certificado

Certifico que la presente Propuesta Tecnológica fue desarrollado por **Karol Gabriel Berrú Jimenez** portador(a) de la cedula de ciudadanía N.º **1900760750** con el tema **"Desarrollo de un entorno de Realidad Virtual para la validación de aplicaciones de aprendizaje lógico-matemático en niños de educación inicial."**, bajo mi supervisión.

Cuenca, **31 de agosto de 2026**

F: 

Ing. John Luis Estrada Guayta, M.S.

Docente - Tutor

Dedicatoria

Dedico este trabajo de titulación a mis padres, por su apoyo, esfuerzo y confianza durante toda mi formación académica. Sus consejos y enseñanzas me ayudaron a continuar adelante y a cumplir una de mis metas profesionales.

A mi familia, por acompañarme en los momentos buenos y difíciles, por comprender el tiempo dedicado a mis estudios y por motivarme a terminar esta etapa.

A mis amigos y compañeros de carrera, con quienes compartí conocimientos, experiencias y momentos importantes durante mi vida universitaria.

Agradecimiento

Agradezco principalmente a Dios por darme salud, fortaleza y sabiduría para superar las dificultades y culminar mi formación universitaria.

A mis padres, por su apoyo económico y emocional, por confiar en mis capacidades y por brindarme las oportunidades necesarias para alcanzar esta meta. También agradezco a mi familia por sus consejos, comprensión y palabras de ánimo.

A mi tutor, por orientarme durante el desarrollo del proyecto, revisar mi trabajo y ayudarme a corregir los errores encontrados durante el proceso.

A mi compañero de proyecto por compartir sus conocimientos y trabajar conmigo en la integración de la programación y el diseño visual. Su colaboración fue importante para completar los objetivos planteados.

Finalmente, agradezco a los docentes, amigos y compañeros de carrera que compartieron sus conocimientos y experiencias conmigo durante estos años de formación académica.

Resumen

El objetivo general fue desarrollar y evaluar, en términos de funcionamiento, usabilidad e interacción, una aplicación de realidad virtual con mecánicas interactivas y retroalimentación en tiempo real orientada a la práctica de destrezas lógico-matemáticas en niños de 4 a 5 años. La investigación fue aplicada, de enfoque cuantitativo y alcance descriptivo; el software se construyó mediante un proceso iterativo en Unity para Meta Quest 3 con seguimiento de manos. Crokiando integra tres niveles: Planeta Dulzura, para reconocimiento de figuras; Parque de Figuras, para clasificación por forma; y Tren de los Patrones, para continuación de secuencias. El material suministrado reporta resultados agregados de 20 participantes en diez preguntas: siete sobre usabilidad e interacción y tres sobre aceptación. Las respuestas favorables oscilaron entre 75 % y 100 %, y la aceptación inicial entre 85 % y 95 %. Sin embargo, no se suministraron la caracterización completa de participantes y sesiones, el instrumento original, la base anonimizada ni los respaldos éticos previos. En consecuencia, la implementación y la verificación funcional quedan documentadas; la evaluación con usuarios solo cuenta con soporte parcial y no demuestra una mejora del aprendizaje ni una validación cerrada.

Palabras clave: realidad virtual, aprendizaje lógico-matemático, educación inicial, interacción, usabilidad.

Abstract

The general objective was to develop and evaluate, in terms of functionality, usability, and interaction, a virtual reality application with interactive mechanics and real-time feedback designed to support the practice of logical-mathematical skills in children aged 4 to 5. This was an applied study with a quantitative approach and a descriptive scope; the software was built iteratively in Unity for Meta Quest 3 using hand tracking. Crokiando integrates three levels: Planeta Dulzura for shape recognition, Parque de Figuras for classification by shape, and Tren de los Patrones for pattern continuation. The supplied material reports aggregate results from 20 participants across ten questions: seven on usability and interaction and three on acceptance. Favorable responses ranged from 75% to 100%, while initial acceptance ranged from 85% to 95%. However, complete participant and session characteristics, the original instrument, an anonymized dataset, and prior ethical records were not supplied. Therefore, implementation and functional verification are documented, whereas the user evaluation is only partially supported and does not demonstrate learning improvement or a closed validation process.

Keywords: *virtual reality, logical-mathematical learning, early childhood education, interaction, usability.*

ÍNDICE DE CONTENIDOS

Certificado	III
Dedicatoria.....	IV
Agradecimiento	V
Resumen	VI
Abstract.....	VII
Índice de tablas	XIII
Índice de ilustraciones	XIV
Introducción.....	1
1. Marco referencial.....	4
1.1 Contextualización del problema	4
1.2 Planteamiento del problema	7
1.3 Formulación del problema.....	10
1.4 Justificación	10
1.5 Objetivo general	12
1.6 Objetivos específicos	13
1.7 Alcance del proyecto	13
1.8 Delimitaciones	16
1.8.1 Delimitación temporal	16
1.8.2 Delimitación geográfica e institucional.....	17
1.8.3 Delimitación poblacional.....	17
1.8.4 Delimitación tecnológica.....	17
1.8.5 Delimitación metodológica	18
1.8.6 Delimitación ética y de seguridad	18
1.9 Beneficiarios	19
1.10 Metodología general	20
1.11 Resultados esperados.....	22
1.12 Relación con el protocolo de investigación.....	23
2. Marco teórico.....	27
2.1 Antecedentes de investigación	27

2.2	Bases teóricas	31
2.2.1	Pensamiento lógico-matemático en educación inicial.....	31
2.2.2	Aprendizaje activo, corporal y progresivo	32
2.2.3	Aprendizaje basado en juegos y gamificación	33
2.2.4	Inmersión, presencia e interactividad	33
2.2.5	Carga cognitiva y aprendizaje multimedia	34
2.2.6	Retroalimentación formativa y refuerzo positivo	34
2.2.7	Usabilidad, experiencia de usuario y diseño centrado en el niño	35
2.2.8	Seguridad, comodidad y supervisión.....	35
2.3	Bases metodológicas	36
2.3.1	Naturaleza aplicada y alcance descriptivo.....	36
2.3.2	Ciencia del diseño y construcción del artefacto	37
2.3.3	Ingeniería de requisitos y trazabilidad.....	37
2.3.4	Gestión ágil con Scrum	38
2.3.5	Diseño iterativo, prototipado y playtesting.....	38
2.3.6	Pruebas y evaluación	39
2.4	Bases tecnológicas	40
2.4.1	Motor Unity y modelo basado en componentes	40
2.4.2	Programación en C#	41
2.4.3	XR Interaction Toolkit	42
2.4.4	OpenXR y compatibilidad de plataforma.....	42
2.4.5	Meta Quest 3 y seguimiento de manos	43
2.4.6	Física, colisiones y zonas de validación	43
2.4.7	Interfaz y retroalimentación dentro del espacio XR.....	44
2.4.8	Rendimiento y comprobación en dispositivo	44
2.5	Arquitecturas, modelos y estándares	45
2.5.1	Arquitectura modular por capas	45
2.5.2	Modularidad de los tres minijuegos.....	47
2.5.3	Modelo de estados	48
2.5.4	Comunicación basada en eventos	49
2.5.5	Modelo de mecánica educativa.....	50

2.5.6	Estándares aplicables	50
2.6	Herramientas de desarrollo	51
2.7	Marco conceptual	54
2.8	Relación entre la teoría y la propuesta tecnológica	58
3.	Manual del usuario	62
3.1	Presentación de la propuesta y perfil del usuario	62
3.1.1	Componentes de la propuesta	63
3.1.2	Perfil de los usuarios	63
3.1.3	Requisitos mínimos para utilizar la experiencia	64
3.1.4	Condiciones de uso seguro	64
3.1.5	Lista de comprobación de seguridad y registro	65
3.2	Acceso e inicio de la experiencia	66
3.2.1	Preparación previa	66
3.2.2	Encendido, ajuste y centrado del visor	67
3.2.3	Opción A: abrir la versión instalada en Meta Quest 3.....	67
3.2.4	Opción B: ejecutar mediante Meta Quest Link	68
3.2.5	Ejecución en cualquiera de las dos computadoras de desarrollo.....	69
3.2.6	Inicio de una sesión de juego.....	71
3.2.7	Cierre de la aplicación	72
3.2.8	Solución de problemas de acceso	73
3.3	Navegación mediante seguimiento de manos.....	73
3.3.1	Interacción principal con las manos	74
3.3.2	Selección directa con el índice	75
3.3.3	Toma, movimiento y colocación de objetos	76
3.3.4	Interpretación de las señales del sistema	77
3.3.5	Pausa, regreso y recentrado supervisados	78
3.4	Funciones e interacciones disponibles.....	78
3.4.1	Selector de nivel y guía de Croki.....	80
3.4.2	Nivel 1: Planeta Dulzura.....	81
3.4.3	Nivel 2: Parque de Figuras	83
3.4.4	Nivel 3: Tren de los Patrones	84

3.4.5	Validación, reintento y retroalimentación	86
3.4.6	Progreso y finalización	86
3.5	Evidencia de uso del prototipo	87
4.	Manual del programador	98
4.1	Arquitectura general del sistema	98
4.1.1	Estructura arquitectónica	98
4.1.2	Distribución de la lógica y flujo de sesión	98
4.2	Tecnologías, motores y herramientas utilizadas.....	100
4.3	Organización técnica del proyecto	102
4.3.1	Estructura de carpetas y recursos.....	102
4.3.2	Correspondencia de escenas	103
4.4	Arquitectura de clases y componentes	104
4.4.1	Modelo basado en componentes.....	104
4.4.2	Comunicación y control de estado	105
4.4.3	Diagrama de componentes y dependencias	106
4.4.4	Modelo de estados de una actividad	107
4.4.5	Secuencia de interacción y validación.....	107
4.5	Desarrollo de sistemas principales	108
4.5.1	Navegación e inicio de las actividades	108
4.5.2	Audio, accesibilidad y medición	109
4.6	Desarrollo de mecánicas del videojuego	109
4.6.1	Nivel 1: Planeta Dulzura.....	109
4.6.2	Nivel 2: Parque de Figuras	111
4.6.3	Nivel 3: Tren de los Patrones	113
4.7	Integración de sistemas.....	115
4.8	Integración entre diseño, arte y programación	116
4.8.1	Aporte individual y trazabilidad de recursos.....	120
4.9	Gestión de datos y persistencia.....	121
4.10	Configuración de escenas y flujo del videojuego.....	122
4.11	Pruebas técnicas y validación	124
4.11.1	Verificación funcional	124

4.11.2	Testeo con participantes	126
4.12	Optimización y rendimiento	127
4.13	Despliegue y compilación	129
4.14	Escalabilidad y mantenimiento.....	130
4.15	Procedimiento para ampliar el sistema	131
4.16	Evidencias técnicas relevantes.....	132
4.17	Resultados, discusión y limitaciones	135
4.17.1	Alcance del análisis	135
4.17.2	Funcionamiento técnico.....	135
4.17.3	Usabilidad y aceptación inicial.....	136
4.17.4	Discusión	139
4.17.5	Limitaciones metodológicas y documentales	139
4.17.6	Trazabilidad ética	140
4.17.7	Rendimiento en dispositivo	141
4.18	Conclusiones y recomendaciones.....	141
4.18.1	Conclusiones.....	141
4.18.2	Recomendaciones	142
	Referencias	144

ÍNDICE DE TABLAS

Tabla 1 Síntesis de causas y consecuencias del problema.....	9
Tabla 2 Alcance funcional de los tres minijuegos.....	14
Tabla 3 Beneficiarios de la propuesta tecnológica.....	19
Tabla 4. Fases generales del proceso de desarrollo.....	21
Tabla 5. Productos y criterios generales de comprobación.....	23
Tabla 6. Correspondencia entre el protocolo y el desarrollo de la tesis.....	24
Tabla 7. Síntesis comparativa de antecedentes relevantes.....	30
Tabla 8. Relación entre las bases metodológicas y sus evidencias.....	40
Tabla 9. Bases tecnológicas del prototipo.....	45
Tabla 10. Modelo funcional de estados de una actividad.....	49
Tabla 11. Estándares y modelos de referencia utilizados.....	51
Tabla 12. Herramientas de desarrollo y justificación de selección.....	53
Tabla 13. Conceptos educativos y de diseño de juego.....	55
Tabla 14. Conceptos de realidad virtual y desarrollo técnico.....	56
Tabla 15. Correspondencia entre fundamentos y decisiones tecnológicas.....	58
Tabla 16. Componentes de la propuesta.....	63
Tabla 17. Perfil de los usuarios.....	63
Tabla 18. Requisitos mínimos para utilizar la experiencia.....	64
Tabla 19. Ejecución en diferentes computadoras.....	70
Tabla 20 Solución de problemas de acceso.....	73
Tabla 21 Interacción principal con las manos.....	74
Tabla 22 Interpretación de las señales del sistema.....	77
Tabla 23. Validación, reintento y retroalimentación.....	86
Tabla 24. Tecnologías principales del prototipo.....	101
Tabla 25. Correspondencia entre los niveles académicos y las escenas internas ...	103
Tabla 26. Clases y componentes principales.....	104
Tabla 27. Gestión y permanencia de los datos.....	122
Tabla 28. Escenas y función dentro del flujo.....	123
Tabla 29. Matriz de verificación funcional.....	124
Tabla 30. Resultados de usabilidad e interacción.....	126
Tabla 31. Estrategias de optimización aplicadas.....	128
Tabla 32 Datos del artefacto de compilación.....	129
Tabla 33. Criterios de escalabilidad y mantenimiento.....	130
Tabla 34. Relación de evidencias técnicas incorporadas.....	134
Tabla 35. Síntesis de funcionamiento técnico y evidencia disponible.....	136
Tabla 36. Resultados agregados de usabilidad, interacción y aceptación inicial ...	138
Tabla 37. Trazabilidad de información disponible y pendiente.....	140

ÍNDICE DE ILUSTRACIONES

Ilustración 1 Flujo funcional de las mecánicas interactivas del prototipo.....	15
Ilustración 2 Ciclo iterativo aplicado al desarrollo del prototipo	39
Ilustración 3 Arquitectura conceptual por capas del entorno inmersivo	47
Ilustración 4 Vista general de Crokiando	65
Ilustración 5 Conexión del Meta Quest 3 con la computadora	69
Ilustración 6 Verificación de Crokiando en el entorno de desarrollo	71
Ilustración 7 Selector principal de minijuegos	72
Ilustración 8 Representación de las manos mediante seguimiento de manos	74
Ilustración 9 Selección de un botón con el índice	75
Ilustración 10 Proceso de tomar una figura	76
Ilustración 11 Proceso de colocar una figura.....	77
Ilustración 12 Secuencia de interacción mediante seguimiento de manos.....	79
Ilustración 13 Explicación de Croki antes de un minijuego.....	81
Ilustración 14 Reconocimiento de figuras en Planeta Dulzura.....	82
Ilustración 15 Validación correcta en Parque de Figuras	84
Ilustración 16 Resolución de una secuencia en Tren de los Patrones	85
Ilustración 17 Proyecto abierto en la computadora 1	87
Ilustración 18 Proyecto abierto en la computadora 2	88
Ilustración 19 Apertura de Crokiando	88
Ilustración 20 Entorno principal y selección de nivel	89
Ilustración 21 Explicación animada de Croki.....	90
Ilustración 22 Reto de reconocimiento de figuras en Planeta Dulzura.....	91
Ilustración 23 Respuesta y avance de ronda en Planeta Dulzura	92
Ilustración 24 Toma de una figura en Parque de Figuras	93
Ilustración 25 Colocación y validación en Parque de Figuras.....	94
Ilustración 26 Patrón incompleto en Tren de los Patrones	95
Ilustración 27 Patrón resuelto y progreso del tren	96
Ilustración 28 Progreso general	97
Ilustración 29 Gestores, servicios compartidos y ciclo de sesión	99
Ilustración 30 Paquetes Meta XR instalados en el proyecto	101
Ilustración 31 Organización de prefabs, escenas y scripts del proyecto (1).....	102
Ilustración 32 Organización de prefabs, escenas y scripts del proyecto (2).....	103
Ilustración 33 Estructura de la mano virtual utilizada para el seguimiento	105
Ilustración 34 Componentes y dependencias principales del prototipo Crokiando	106
Ilustración 35 Modelo de estados de una actividad.....	107
Ilustración 36 Secuencia de selección, validación y respuesta.....	108
Ilustración 37 Configuración del gestor de Planeta Dulzura.....	110
Ilustración 38 Configuración de una respuesta geométrica de Planeta Dulzura	111
Ilustración 39 Gestión y validación de cofres en Parque de Figuras, parte 1	112

Ilustración 40 Gestión y validación de cofres en Parque de Figuras, parte 2	113
Ilustración 41 Configuración del patrón en Tren de los Patrones, parte 1	114
Ilustración 42 Configuración del patrón en Tren de los Patrones, parte 2	114
Ilustración 43 Componentes de agarre y transformación manual	116
Ilustración 44 Escenarios implementados para Planeta Dulzura.....	117
Ilustración 45 Escenarios implementados para Parque de Figuras.....	118
Ilustración 46 Escenarios implementados para Tren de los Patrones.....	118
Ilustración 47 Configuración del sistema centralizado de audio.....	119
Ilustración 48 Material y recursos del sistema de accesibilidad cromática	120
Ilustración 49 Paquete APK generado para la instalación.....	129
Ilustración 50 Resultados descriptivos del testeo del prototipo	137

Introducción

La educación inicial constituye una etapa decisiva para el desarrollo de habilidades de observación, comparación, clasificación y reconocimiento de patrones. En Ecuador, el Currículo de Educación Inicial incorpora las relaciones lógico-matemáticas en el eje de descubrimiento del medio natural y cultural y, para niños de 4 a 5 años, contempla destrezas como clasificar objetos por atributos, identificar semejanzas y diferencias y continuar patrones simples. Estas capacidades preparan aprendizajes matemáticos posteriores y pueden fortalecerse mediante actividades concretas, progresivas y comprensibles.

El aprendizaje basado en juegos ofrece oportunidades para favorecer la participación y la motivación infantil, aunque su aporte depende de la calidad del diseño y de la evaluación aplicada. En este contexto, la realidad virtual permite situar al usuario dentro de escenarios tridimensionales en los que puede observar, seleccionar, tomar y colocar objetos. La literatura reporta resultados educativos prometedores, pero también advierte que la inmersión, por sí sola, no garantiza el aprendizaje y que sus efectos dependen de la organización de la actividad, la duración, la interactividad y la carga cognitiva.

A partir de esta consideración, el problema abordado en la tesis es la insuficiente implementación de mecánicas interactivas estructuradas en experiencias de realidad virtual orientadas al aprendizaje lógico-matemático. Esta dificultad es especialmente relevante en niños de 4 a 5 años, quienes requieren instrucciones breves, acciones reconocibles y una relación clara entre su intervención y la respuesta del sistema. Si no existen objetivos, reglas, condiciones de acierto y error, retroalimentación inmediata y control del progreso, el entorno puede convertirse en una exploración atractiva, pero sin una finalidad educativa técnicamente verificable.

Para atender esta necesidad se desarrolló en Unity un entorno de realidad virtual para Meta Quest 3, con interacción mediante seguimiento de manos. El prototipo, denominado Crokiando, integra tres niveles: Planeta Dulzura, dedicado al reconocimiento de figuras geométricas; Parque de Figuras, orientado a la clasificación por forma; y Tren de los Patrones, centrado en la continuación de secuencias. En los tres casos se aplica un flujo común de acción, validación, retroalimentación y actualización del progreso, con el propósito de relacionar cada destreza con una mecánica observable y comprobable.

La investigación es aplicada, de enfoque cuantitativo y alcance descriptivo. El desarrollo del software es iterativo y se verifica mediante pruebas funcionales, observación de la interacción, registro de errores y sesiones de prueba con usuarios (playtesting). Este diseño permite describir el comportamiento del prototipo y los resultados agregados reportados; por sí solo no demuestra una relación causal entre el uso del videojuego y una mejora estable del aprendizaje.

Los resultados agregados suministrados para 20 participantes muestran respuestas favorables de usabilidad e interacción entre 75 % y 100 %, con 100 % en el reconocimiento de aciertos y errores. La aceptación inicial alcanzó 85 % en Planeta Dulzura y 95 % tanto en Parque de Figuras como en Tren de los Patrones. Estos datos aportan evidencia preliminar sobre el funcionamiento percibido, la usabilidad y la aceptación del prototipo; sin embargo, no demuestran una mejora del aprendizaje ni permiten generalizar, debido a la ausencia de la caracterización completa de participantes y sesiones, el instrumento original, la base anonimizada, la trazabilidad ética completa y mediciones repetidas de rendimiento en Meta Quest 3.

La tesis se organiza en seis capítulos. El primero presenta el marco referencial, el problema, la justificación, los objetivos, el alcance y la metodología general; el segundo desarrolla las bases teóricas, metodológicas y tecnológicas; el tercero expone el manual de usuario y las pautas de interacción con los tres niveles; el cuarto documenta la arquitectura, la programación, la integración y las pruebas; el quinto analiza los resultados, la discusión y las limitaciones; y el sexto formula las conclusiones y recomendaciones. Al final se incluyen las referencias.

Capítulo I.

1. Marco referencial

1.1 Contextualización del problema

La educación inicial constituye una etapa decisiva para el desarrollo de habilidades que permiten a los niños reconocer semejanzas y diferencias, organizar objetos, establecer relaciones y resolver situaciones sencillas. En Ecuador, el Currículo de Educación Inicial ubica las relaciones lógico-matemáticas dentro del eje de descubrimiento del medio natural y cultural. Para los niños de 4 a 5 años, el documento plantea, entre otras destrezas, clasificar objetos según atributos como forma, color o tamaño, identificar semejanzas y diferencias y continuar patrones simples mediante objetos concretos o representaciones gráficas [1].

Estas destrezas no se reducen al aprendizaje de números. También comprenden la capacidad de observar, comparar, ordenar y anticipar qué elemento continúa dentro de una secuencia. Un estudio longitudinal realizado con niños desde los 4 hasta los 11 años encontró que conocimientos tempranos como el conteo, la cantidad y el reconocimiento de patrones se relacionan con aprendizajes matemáticos posteriores [2]. En consecuencia, trabajar formas, colores y patrones durante la educación inicial tiene una función preparatoria para aprendizajes de mayor complejidad.

El juego es una vía adecuada para presentar estas actividades porque permite que el niño aprenda mientras toma decisiones y observa el resultado de sus acciones. Una revisión sistemática y metaanálisis sobre aprendizaje basado en juegos en educación infantil informó efectos favorables en resultados cognitivos, motivación y participación, aunque también señaló la necesidad de mejorar la calidad y consistencia de las investigaciones [3]. Esto indica

que el uso de un videojuego puede apoyar la experiencia educativa, pero su aporte depende de que las actividades, reglas y respuestas del sistema estén bien diseñadas.

En este escenario, la realidad virtual ofrece la posibilidad de pasar de una actividad observada en una pantalla plana a una experiencia situada dentro de un espacio tridimensional. Mediante el visor y el seguimiento de manos, el niño puede observar los elementos desde distintos ángulos, seleccionarlos, tomarlos, moverlos y colocarlos sin sostener controladores. Un metaanálisis centrado en educación K-6 encontró un efecto positivo general de la realidad virtual sobre el aprendizaje y señaló resultados favorables en intervenciones cortas; sin embargo, estos hallazgos corresponden a estudios diversos y no demuestran por sí solos la efectividad de una aplicación concreta [4].

La investigación específica sobre realidad virtual en educación infantil todavía es limitada. Una revisión publicada en 2025 identificó un crecimiento del interés en este campo, pero únicamente ocho trabajos cumplieron los criterios de selección para el análisis final [5]. Esta cantidad reducida confirma que existe una oportunidad de desarrollo, pero también obliga a evitar afirmaciones generales que no hayan sido comprobadas. Por ello, el presente proyecto no parte de la idea de que la inmersión garantiza el aprendizaje, sino de la necesidad de construir y evaluar una experiencia con objetivos claros.

Las revisiones sobre realidad virtual educativa muestran resultados prometedores, pero también advierten que la calidad del aprendizaje depende del diseño de la actividad, de la duración de la experiencia y de los instrumentos empleados para evaluarla [6]. El modelo CAMIL explica que la inmersión y la interactividad pueden influir en la presencia, la sensación de control, la motivación y la carga mental del usuario; estos factores pueden facilitar o dificultar el aprendizaje según la forma en que se organice la experiencia [7]. En

un proyecto para niños pequeños, esto significa que el escenario no debe limitarse a ser llamativo: debe indicar qué hacer, aceptar acciones comprensibles y responder con claridad.

La sensación de presencia tampoco depende únicamente de disponer de un visor. Un metaanálisis incluido en la bibliografía del anteproyecto encontró que determinadas características inmersivas influyen en la presencia, pero su aporte varía según la combinación de recursos y el contexto de uso [8]. En consecuencia, la selección tecnológica debe estar subordinada a la actividad educativa y no reemplazar la claridad de las reglas ni la orientación del usuario.

Con base en esta necesidad se planteó el desarrollo de un videojuego educativo de realidad virtual en Unity. El prototipo se organiza en tres niveles: Nivel 1, Planeta Dulzura, centrado en el reconocimiento de figuras geométricas; Nivel 2, Parque de Figuras, orientado a la clasificación por forma; y Nivel 3, Tren de los Patrones, dedicado a completar secuencias por forma. Los tres niveles comparten un flujo compuesto por la acción del usuario, la comprobación de la respuesta, la retroalimentación, el refuerzo positivo y la actualización del progreso.

Dentro del trabajo colaborativo, Programación, Desarrollo e Integración de Sistemas, se ocupa de convertir ese diseño educativo en un sistema funcional. Esto implica programar la interacción con los objetos, detectar selecciones y colocaciones, comparar la respuesta con la regla de cada actividad, activar sonidos o animaciones, controlar los intentos y permitir el avance entre niveles. Los recursos visuales y sonoros aportados por el equipo se integran en Unity para que respondan de manera coherente con la lógica programada.

1.2 Planteamiento del problema

El problema central identificado es la insuficiente implementación de mecánicas interactivas estructuradas en experiencias educativas de realidad virtual dirigidas al aprendizaje lógico-matemático. Una aplicación puede presentar objetos tridimensionales y un escenario atractivo, pero si no define objetivos, reglas, condiciones de acierto, tratamiento del error y una respuesta inmediata, la interacción se convierte en exploración sin una finalidad educativa claramente controlada.

Esta dificultad adquiere mayor importancia cuando el público está formado por niños de 4 a 5 años. A esa edad, una instrucción extensa, un botón poco visible o una respuesta ambigua pueden impedir que el niño comprenda la actividad. El sistema debe reducir acciones innecesarias, utilizar elementos fáciles de reconocer y mostrar una relación directa entre lo que hace el usuario y lo que ocurre después. La retroalimentación formativa debe ser oportuna, específica y útil para que el usuario relacione su acción con el resultado [9]. Además, una investigación sobre retroalimentación en juegos matemáticos para niños de edad preescolar mostró que la forma, la voz y el tipo de respuesta pueden generar reacciones emocionales diferentes [10]. Por tanto, la retroalimentación no es un elemento decorativo, sino una decisión de diseño que debe adaptarse a la edad.

Desde la programación, el problema se manifiesta cuando la aplicación no reconoce de forma confiable la acción del usuario o cuando los sistemas trabajan de manera aislada. Por ejemplo, un objeto puede verse correctamente, pero no ser tomado; puede llegar a un recipiente, pero no ser identificado; o la respuesta puede validarse, pero no actualizar el progreso. También pueden aparecer fallos por la configuración de colisiones, cuerpos físicos,

referencias entre componentes, estados de animación, eventos de interfaz o cambios de escena.

La ausencia de una arquitectura funcional clara produce consecuencias para el usuario y para el equipo de desarrollo. Para el niño, puede generar confusión, repetición de errores, pérdida de interés o dificultad para saber si completó el reto. Para el proyecto, provoca código difícil de mantener, respuestas inconsistentes entre niveles, mayor tiempo de corrección y riesgo de integrar recursos que no cumplen ninguna función. Por ello, el problema no consiste solamente en la falta de una aplicación de realidad virtual, sino en la necesidad de conectar correctamente el objetivo de aprendizaje con una mecánica verificable.

La experiencia del usuario en un entorno inmersivo reúne dimensiones como presencia, participación, facilidad de uso, emoción y consecuencias de la experiencia. El cuestionario propuesto por Tcha-Tokey et al., incluido en el anteproyecto, organiza estas dimensiones para estudiar experiencias inmersivas [11]. Sin embargo, ese instrumento no fue diseñado específicamente para niños de 4 a 5 años; por ello, no debe aplicarse sin una adaptación autorizada y comprensible para la población del proyecto.

Los puntos, recompensas y niveles pueden apoyar la participación, pero no deben reemplazar el objetivo educativo. La revisión de Hamari et al. mostró que los resultados de la gamificación dependen del contexto y de la forma de implementación [12]. Un metaanálisis posterior encontró efectos positivos pequeños en resultados cognitivos, motivacionales y conductuales [13]. En este proyecto, la recompensa se utiliza después de una acción correcta y el progreso representa el avance real dentro de la actividad; no se plantea como un estímulo separado del aprendizaje.

La calidad del sistema también requiere considerar la usabilidad. La norma ISO 9241-11:2018 entiende la usabilidad como un resultado del uso de un sistema dentro de un contexto definido y proporciona un marco para analizar si dicho uso es efectivo, eficiente y satisfactorio [14]. Aplicado al prototipo, esto exige observar si el usuario comprende la tarea, puede completarla con las interacciones disponibles y recibe información suficiente para continuar.

Tabla 1
Síntesis de causas y consecuencias del problema

Causa identificada	Manifestación en el prototipo	Consecuencia principal
Reglas de interacción poco definidas	El usuario no sabe qué objeto tomar, dónde colocarlo o qué opción seleccionar.	Confusión y abandono de la actividad.
Validación incompleta o aislada	La aplicación no distingue correctamente entre una acción válida y una incorrecta.	Avance equivocado o repetición sin orientación.
Retroalimentación tardía o ambigua	El acierto o el error no produce una señal clara y relacionada con la acción.	Dificultad para corregir y aprender de la respuesta.
Integración técnica inconsistente	Objetos, animaciones, sonidos, interfaz y progreso no comparten el mismo estado.	Fallos funcionales y mayor costo de mantenimiento.
Evaluación limitada	Se afirma que la experiencia es útil sin separar funcionamiento, usabilidad y aprendizaje.	Conclusiones que no están respaldadas por la evidencia recogida.

Nota: elaboración propia a partir del anteproyecto y del proceso de desarrollo (2026).

La Tabla 1 relaciona las principales debilidades de definición, orientación y comprobación con sus manifestaciones en el prototipo y con el efecto esperado sobre la experiencia; esta síntesis delimita el problema que aborda la propuesta.

En consecuencia, se requiere desarrollar una solución que organice las actividades como módulos, establezca reglas comprobables, integre los componentes del entorno y permita realizar pruebas sobre el comportamiento del sistema. El aporte individual de

programación evaluado en esta tesis responde a esta necesidad mediante la programación e integración de las mecánicas principales del videojuego.

1.3 Formulación del problema

¿Cómo desarrollar e integrar en Unity un entorno inmersivo de realidad virtual con mecánicas de reconocimiento de figuras, clasificación por forma y continuación de patrones que permita a niños de 4 a 5 años interactuar, recibir retroalimentación inmediata y completar una progresión de niveles de manera comprensible y técnicamente estable?

1.4 Justificación

La propuesta se justifica, en primer lugar, por su relación con el aprendizaje lógico-matemático en educación inicial. Las mecánicas seleccionadas corresponden a destrezas reconocidas en el currículo ecuatoriano para el grupo de 4 a 5 años: clasificar, comparar atributos y continuar patrones [1]. El proyecto no sustituye la orientación del docente ni las actividades con objetos reales; ofrece un recurso complementario para presentar esas destrezas mediante acciones dentro de un entorno virtual.

Desde la perspectiva científica, el trabajo aporta evidencia de desarrollo en un campo que todavía presenta pocos estudios específicos en educación infantil [5]. Las revisiones disponibles muestran beneficios posibles de la realidad virtual y del aprendizaje basado en juegos, pero también insisten en que la tecnología debe estar acompañada por un diseño pedagógico y una evaluación rigurosa [3], [6]. En consecuencia, el valor del proyecto no reside únicamente en utilizar un visor, sino en documentar cómo se transforma una destreza en una secuencia de interacción que puede ser comprobada.

En el ámbito tecnológico, Unity se utiliza como motor para reunir escenas tridimensionales, objetos físicos, animaciones, sonido, interfaz y lógica del videojuego [15].

El XR Interaction Toolkit aporta un sistema de componentes para seleccionar, activar y manipular objetos en experiencias VR y AR [16]. OpenXR proporciona una interfaz estándar para acceder a distintas plataformas y dispositivos XR [17], mientras que C# se emplea para programar las reglas, los estados y la comunicación entre componentes [18]. Meta Quest 3 constituye el dispositivo de referencia para la compilación y las pruebas; la documentación oficial de Meta describe la preparación de Unity y OpenXR para visores Quest [19]. Estas herramientas permiten construir el prototipo de forma modular.

La justificación práctica se relaciona con la necesidad de obtener un flujo claro y repetible. En Planeta Dulzura, el niño selecciona la forma que coincide con la figura presentada; en Parque de Figuras, toma un objeto con la mano y lo coloca en el contenedor de la misma forma; y en Tren de los Patrones, toma la figura que completa la secuencia y la coloca en el espacio vacío. En todos los casos, la programación detecta la acción, la compara con una condición y presenta una respuesta inmediata.

Desde la calidad de software, el proyecto requiere comprobar que las funciones previstas estén presentes, que las interacciones respondan de forma confiable, que el rendimiento sea adecuado al dispositivo y que el código pueda mantenerse. ISO/IEC 25010:2023 proporciona un modelo de calidad de producto aplicable al software y permite organizar características como adecuación funcional, eficiencia de desempeño, fiabilidad, capacidad de interacción y mantenibilidad [20]. El proyecto no pretende certificar el prototipo bajo esta norma, pero utiliza sus categorías como referencia para ordenar las pruebas técnicas.

En el ámbito académico, el desarrollo reúne conocimientos de programación en C#, arquitectura de componentes, física en tiempo real, interacción XR, interfaz, control de

estados, pruebas y documentación. También permite evidenciar el aporte individual de programación dentro de un trabajo en equipo: mientras los recursos visuales y sonoros son producidos o preparados por otros integrantes, la programación los conecta con las reglas y convierte el conjunto en una experiencia funcional.

La dimensión social del proyecto se relaciona con la búsqueda de recursos educativos que promuevan una participación activa. Sin embargo, al tratarse de niños pequeños, el beneficio potencial debe acompañarse de medidas de seguridad, consentimiento del representante, supervisión adulta y control del tiempo de exposición. La bibliografía del anteproyecto señala la relación negativa que puede existir entre presencia y cibermareo [21]. Asimismo, una revisión sobre seguridad de realidad virtual en menores de 14 años encontró evidencia limitada sobre daños durante exposiciones breves y supervisadas, pero destacó la falta de datos y de instrumentos consistentes para registrar efectos adversos [22]. Por esta razón, cualquier prueba con niños debe realizarse dentro de un protocolo institucional y detenerse ante molestias visuales, mareo o desorientación.

Finalmente, la propuesta es pertinente porque produce un resultado concreto: un prototipo que puede ser ejecutado, observado y evaluado. Su contribución se medirá primero por el correcto funcionamiento de las mecánicas, la claridad del flujo y la calidad de la interacción. Una mejora del aprendizaje solo podrá afirmarse si se aplican instrumentos adecuados y se obtienen resultados suficientes con la población objetivo.

1.5 Objetivo general

Desarrollar y evaluar, en términos de funcionamiento, usabilidad e interacción, una aplicación de realidad virtual con mecánicas interactivas y retroalimentación en tiempo real orientada a la práctica de destrezas lógico-matemáticas en niños de 4 a 5 años.

1.6 Objetivos específicos

- Analizar los principios del aprendizaje lógico-matemático y del desarrollo cognitivo en niños de 4 a 5 años para definir los requerimientos pedagógicos del entorno de realidad virtual.
- Diseñar la estructura funcional de los minijuegos y las mecánicas interactivas del entorno de realidad virtual, incorporando criterios de interacción y gamificación.
- Implementar la aplicación en Unity mediante mecánicas interactivas y recursos tridimensionales que permitan la manipulación de objetos y la retroalimentación en tiempo real.
- Integrar los recursos tridimensionales y componentes físicos en el entorno de realidad virtual, configurando propiedades como colliders y rigidbodies, con el fin de garantizar la coherencia funcional, la correcta interacción con las mecánicas diseñadas y un comportamiento consistente dentro del sistema.
- Evaluar la usabilidad, la interacción y la experiencia de usuario del entorno desarrollado mediante pruebas funcionales y sesiones de playtesting.

1.7 Alcance del proyecto

El proyecto comprende el diseño, la programación, la integración, la prueba y la evaluación de un prototipo de videojuego educativo de realidad virtual dirigido a niños de 4 a 5 años. El hardware de referencia es Meta Quest 3 y el desarrollo se realiza en Unity con C#, XR Interaction Toolkit y OpenXR [15], [19]. La solución está orientada a una experiencia individual y local, sin depender de conexión a Internet para ejecutar las actividades principales. La evaluación se limita a la evidencia verificable disponible: las pruebas funcionales documentan el comportamiento del prototipo, mientras que los resultados

agregados con usuarios se consideran preliminares hasta cerrar el instrumento, la base y la trazabilidad ética.

El alcance funcional se concentra en tres minijuegos. Cada uno representa una destreza distinta, pero todos comparten el mismo principio: presentar un reto, recibir una acción, comprobarla, responder y actualizar el progreso. La Tabla 2 resume la función de cada nivel.

Tabla 2
Alcance funcional de los tres minijuegos

Nivel	Destreza	Acción principal	Respuesta del sistema
1. Planeta Dulzura	Reconocimiento de figuras	Seleccionar con el índice la forma que coincide con la figura presentada.	Validar la elección, mostrar retroalimentación y avanzar de ronda.
2. Parque de Figuras	Clasificación por forma	Tomar la figura con la mano y colocarla en el contenedor de la misma forma.	Validar la colocación, mostrar retroalimentación y actualizar el progreso.
3. Tren de los Patrones	Continuación de patrones	Tomar la figura que completa la secuencia y colocarla en el espacio vacío.	Comprobar la regla, informar el resultado y completar el desafío.

Nota: elaboración propia con base en el anteproyecto y el diagrama de mecánicas (2026).

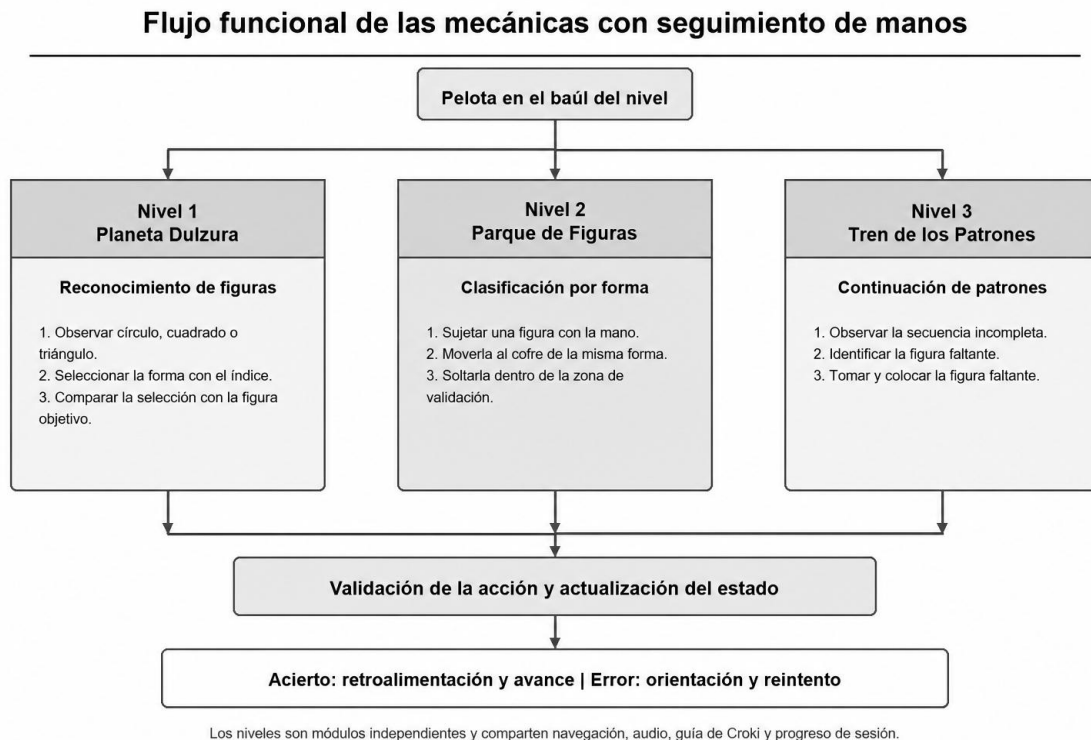
La tabla diferencia los tres minijuegos según la destreza trabajada, la acción manual requerida y la respuesta del sistema. Aunque las mecánicas cambian entre selección, clasificación y continuación de patrones, todas conservan la secuencia común de interacción, validación, retroalimentación y avance.

Como se observa en la Ilustración 1, el flujo funcional general separa los tres módulos. La colocación de una pelota en el baúl correspondiente abre uno de ellos; por eso, las acciones internas de un nivel no se conectan directamente con las de otro. Los módulos

solo convergen en el sistema común de progreso y recompensas. Esta ilustración documenta la lógica de interacción y no reemplaza los diagramas de clases.

Ilustración 1

Flujo funcional de las mecánicas interactivas del prototipo



Nota: elaboración propia a partir del diagrama de mecánicas del proyecto (2026).

La Ilustración 1 sintetiza la selección del nivel mediante la colocación de una pelota en el baúl correspondiente, la ejecución independiente de las tres mecánicas y el flujo común de validación, retroalimentación y actualización del progreso.

El alcance técnico incluye:

- Configuración del entorno XR, seguimiento de manos y objetos interactivos.
- Programación de selección directa, agarre natural, movimiento, colocación y activación de botones virtuales mediante el índice.

- Validación de figuras, destinos y secuencias mediante reglas definidas para cada nivel.
- Retroalimentación visual y sonora ante aciertos y errores, junto con animaciones asociadas.
- Control de intentos, puntos, recompensas, finalización de nivel y progreso general.
- Integración de modelos, materiales, colliders, rigidbodies, animaciones, audio e interfaz aportados por el equipo.
- Pruebas funcionales, corrección de errores y revisión del rendimiento en el dispositivo de referencia.

El proyecto no incluye:

- Sistemas multijugador, servicios en línea o intercambio de datos entre usuarios.
- Inteligencia artificial avanzada o adaptación automática del nivel de dificultad.
- Producción completa de todos los modelos y animaciones desde cero como responsabilidad de Programación, Desarrollo e Integración de Sistemas.
- Un sistema clínico, terapéutico o de diagnóstico del desarrollo infantil.
- Una investigación experimental a gran escala que permita generalizar mejoras del aprendizaje a toda la población infantil.

1.8 Delimitaciones

1.8.1 Delimitación temporal

El desarrollo se organizó durante seis meses e incluyó análisis, diseño de mecánicas, configuración de Unity, programación, integración, pruebas, optimización y documentación. La versión identificable en la evidencia disponible es Crokiando.apk, documentada el 27 de julio de 2026 con un tamaño de 124 592 KB. Esta referencia no se presenta como constancia

de entrega oficial; cualquier correspondencia con una entrega institucional debe comprobarse contra su registro de recepción.

1.8.2 Delimitación geográfica e institucional

El prototipo se desarrolla en el contexto académico de la Universidad Católica de Cuenca, matriz Cuenca. Las pruebas técnicas pueden realizarse en el laboratorio o en espacios controlados que permitan usar el visor con seguridad. Los resultados obtenidos en este contexto no deben generalizarse automáticamente a todas las instituciones de educación inicial del Ecuador.

1.8.3 Delimitación poblacional

El público objetivo corresponde a niños de 4 a 5 años de Inicial 2. Los docentes, representantes y especialistas cumplen una función de acompañamiento y observación. Debido a la edad de los participantes, los instrumentos deben utilizar lenguaje sencillo, apoyos visuales y tiempos breves; además, no se debe interpretar el desempeño en el juego como un diagnóstico individual.

1.8.4 Delimitación tecnológica

La solución se limita al motor Unity, la programación en C#, la interacción XR, la compatibilidad mediante OpenXR y la ejecución en Meta Quest 3 [15], [19]. El prototipo depende de la configuración del seguimiento de manos, del área segura de juego y del rendimiento disponible en el visor. No se asegura compatibilidad con otros dispositivos sin pruebas adicionales. El manual técnico registra la versión de Unity y los paquetes utilizados en la versión final.

1.8.5 Delimitación metodológica

La investigación es aplicada y de alcance descriptivo. El desarrollo del software es iterativo y se valida mediante pruebas funcionales, observación de la interacción, registro de errores y playtesting. Este diseño permite describir el comportamiento del prototipo y detectar problemas de uso; por sí solo no demuestra una relación causal entre el uso del videojuego y una mejora estable del aprendizaje.

1.8.6 Delimitación ética y de seguridad

Las pruebas con menores requieren autorización institucional, consentimiento informado del representante, asentimiento del niño cuando corresponda, supervisión adulta permanente, higiene del visor, un espacio libre de obstáculos y suspensión inmediata ante cualquier señal de malestar. La Academia Americana de Pediatría recomienda límites razonables, descansos y orientación del adulto, debido a que los niños pueden tener dificultad para explicar molestias visuales o retirar el visor por sí mismos [23].

Existe, además, una diferencia entre el público educativo del proyecto y las condiciones comerciales del dispositivo. Meta señala que las funciones de supervisión de Quest se ofrecen para niños y adolescentes desde los 10 años, aunque la edad puede variar según la región [24]. Por ello, un niño de 4 o 5 años no debe utilizar de forma autónoma una cuenta infantil convencional de la plataforma. Cualquier evaluación con el visor debe plantearse como una actividad institucional controlada, con un perfil administrado por un adulto y con la revisión previa del tutor y del comité o autoridad que corresponda. Antes de cualquier nueva sesión con menores deben constar una autorización institucional previa, la evaluación de riesgos y un protocolo que defina supervisión adulta uno a uno, área despejada, higiene del visor, pausas y suspensión inmediata ante malestar.

En los materiales suministrados para esta revisión no constan la aprobación institucional previa, el consentimiento informado, el asentimiento cuando corresponda ni la autorización de uso de imagen. Estos respaldos no pueden generarse de forma retrospectiva. Por ello, los resultados agregados se informan únicamente como datos preliminares reportados y se excluyen de cualquier afirmación de validación ética o evaluación cerrada.

1.9 Beneficiarios

Los beneficiarios se dividen en directos e indirectos. Esta clasificación se refiere a las personas que pueden utilizar, observar o reutilizar el resultado del proyecto; no supone que el beneficio educativo ya haya sido demostrado.

Tabla 3 Beneficiarios de la propuesta tecnológica

Tipo	Beneficiarios	Beneficio esperado
Directos	Niños de 4 a 5 años que participen en pruebas autorizadas.	Acceder a actividades de reconocimiento de figuras, clasificación por forma y continuación de patrones mediante interacción, juego y retroalimentación inmediata.
	Docentes o acompañantes de educación inicial.	Disponer de un prototipo que permita observar el desempeño del niño y complementar actividades presenciales.
	Equipo de desarrollo, especialmente el Programación, Desarrollo e Integración de Sistemas.	Aplicar y documentar programación XR, integración, validación y pruebas de un sistema funcional.
Indirectos	Universidad Católica de Cuenca y comunidad académica.	Contar con evidencia de un proceso tecnológico relacionado con realidad virtual y educación inicial.
	Futuros estudiantes y desarrolladores.	Reutilizar la arquitectura, las reglas y la documentación para ampliar niveles o crear propuestas similares.
	Representantes e instituciones educativas.	Conocer las posibilidades y limitaciones de la realidad virtual como recurso complementario, siempre bajo supervisión.

Nota: elaboración propia (2026).

La Tabla 3 distingue beneficiarios directos e indirectos y delimita el beneficio esperado para cada grupo. Estas formulaciones expresan el alcance potencial de la propuesta y no constituyen evidencia de un efecto educativo ya demostrado.

1.10 Metodología general

El proyecto adopta una investigación aplicada porque busca resolver una necesidad mediante la construcción de un producto tecnológico. Su enfoque es cuantitativo y su alcance descriptivo: se sistematizan frecuencias y porcentajes agregados, se caracterizan las actividades y se documenta el comportamiento del prototipo. El término iterativo corresponde al proceso de desarrollo del software: cada versión de una mecánica se implementa, prueba, corrige y vuelve a integrar. Por tanto, el enfoque y el alcance de investigación se distinguen del proceso iterativo de desarrollo. La interpretación cuantitativa permanece limitada hasta disponer del instrumento original, la base anonimizada y la caracterización de la muestra.

La definición de requisitos parte del anteproyecto, del currículo de educación inicial y del flujo de cada minijuego. Como referencia, ISO/IEC/IEEE 29148:2018 describe procesos e información relacionados con la ingeniería de requisitos a lo largo del ciclo de vida de sistemas y software [25]. En este trabajo, los requisitos se expresan mediante objetivos de aprendizaje, acciones del usuario, respuestas del sistema, condiciones de éxito y error, recursos necesarios y criterios de prueba.

Para organizar las tareas se utilizaron períodos cortos de trabajo, listas de actividades, revisión de avances y ajustes del prototipo. Estas prácticas son compatibles con Scrum, marco que promueve inspección y adaptación mediante ciclos de trabajo y revisión [26]. Sin

embargo, Scrum se emplea como apoyo para gestionar el desarrollo; no sustituye el diseño de investigación ni las fases técnicas descritas en el anteproyecto.

Tabla 4. Fases generales del proceso de desarrollo

Fase	Actividades principales	Evidencia o producto
1. Análisis y requisitos	Revisar currículo, anteproyecto, público objetivo, recursos disponibles y restricciones del visor. Definir acciones, entradas, salidas y criterios de aceptación.	Lista de requisitos, alcance actualizado y reglas de cada actividad.
2. Diseño de mecánicas	Representar el flujo de los tres niveles, establecer aciertos, errores, reintentos, puntos, recompensas y cambio de nivel.	Diagrama funcional, esquemas de interacción y diseño de niveles.
3. Configuración XR	Preparar el proyecto Unity, OpenXR, XR Interaction Toolkit, seguimiento de manos, cámara, área de juego y objetos interactivos.	Proyecto base ejecutable en el dispositivo de referencia.
4. Programación	Crear componentes en C# para tomar, seleccionar, colocar, validar, emitir retroalimentación y controlar el progreso.	Scripts y prefabs funcionales por mecánica.
5. Integración	Vincular modelos, colliders, rigidbodies, botones, animaciones, audio, interfaz y cambios de escena con la lógica programada.	Niveles integrados y flujo completo del videojuego.
6. Pruebas y ajustes	Ejecutar casos correctos e incorrectos, revisar referencias, colisiones, eventos, reintentos, rendimiento y comprensión de la interacción.	Matriz de pruebas, registro de errores y versiones corregidas.
7. Validación y cierre	Realizar pruebas funcionales y, únicamente con autorización previa y expediente ético completo, evaluar la usabilidad, la interacción y la experiencia de usuario; consolidar evidencias, analizar los resultados según el instrumento aplicado y documentar la versión final.	Prototipo final, informe de evaluación y validación con alcance declarado, y documentación técnica.

Nota: elaboración propia con base en la metodología del anteproyecto (2026).

La Tabla 4 organiza el desarrollo en siete fases consecutivas, desde el análisis de requisitos hasta la validación y el cierre, y vincula cada fase con actividades y productos observables. Esta correspondencia permite comprobar el avance mediante requisitos, diagramas, scripts, escenas integradas, matrices de prueba y documentación final.

Las pruebas técnicas se organizan por casos. Cada caso registra el estado inicial, la acción realizada, el resultado esperado, el resultado observado y la evidencia. Los escenarios principales incluyen seleccionar figuras correctas e incorrectas, colocar figuras en destinos correctos e incorrectos, completar o fallar un patrón, repetir un intento, terminar un nivel y regresar al selector.

La evaluación con usuarios debe separar tres dimensiones. La primera es el funcionamiento: si la acción es detectada y la regla se cumple. La segunda es la usabilidad: si el usuario comprende y completa la tarea con la ayuda prevista. La tercera es el aprendizaje: si existen cambios en el desempeño lógico-matemático. Las dos primeras pueden evaluarse con observación, errores, tiempo, finalización y una escala visual sencilla. La tercera requiere un diseño educativo y mediciones antes y después o una comparación equivalente; por ello, no debe confundirse una encuesta de agrado con evidencia de aprendizaje.

El análisis combina datos simples de funcionamiento, como cantidad de aciertos, errores, tareas completadas y estabilidad del sistema, con observaciones cualitativas sobre dudas, ayudas solicitadas, reacciones y comentarios. Debido a la edad del público, las preguntas deben ser breves, leídas por un adulto y apoyadas con pictogramas o expresiones faciales. La interpretación debe limitarse a lo que realmente mide cada instrumento.

1.11 Resultados esperados

Al finalizar el proyecto se espera obtener un prototipo funcional de realidad virtual que integre los tres minijuegos y permita ejecutar el recorrido completo desde la selección del nivel hasta la actualización del progreso. El resultado principal del aporte individual de programación será la lógica programada que conecta las acciones del usuario con las reglas, respuestas y estados del videojuego.

Tabla 5. Productos y criterios generales de comprobación

Resultado esperado	Criterio de comprobación
Tres minijuegos funcionales	Cada nivel inicia, recibe la interacción prevista, valida aciertos y errores y termina según sus reglas.
Interacción VR comprensible	Los objetos y botones responden a las acciones configuradas y el usuario puede completar el flujo con ayudas acordes a la edad.
Retroalimentación inmediata	Toda respuesta relevante activa una señal visual, sonora o animada coherente con el resultado.
Progreso y recompensas	Los aciertos, la finalización del nivel y el avance general se actualizan sin duplicaciones ni pérdidas de estado.
Integración estable	Modelos, física, animaciones, audio, interfaz y scripts trabajan de manera coordinada en el dispositivo de referencia.
Evidencia de pruebas	Existe una matriz con casos ejecutados, errores encontrados, correcciones y resultado de la versión final.
Documentación reutilizable	Los diagramas, reglas, estructura técnica e indicaciones permiten mantener o ampliar el sistema.

Nota: elaboración propia (2026).

Como resultado educativo esperado, se plantea que las actividades permitan practicar reconocimiento de figuras, clasificación por forma y continuación de patrones. Esta formulación expresa la finalidad de la propuesta, no una conclusión anticipada. La tesis solo podrá afirmar una mejora del aprendizaje si los datos obtenidos y el diseño de evaluación permiten sostenerla. En caso contrario, el resultado deberá presentarse como viabilidad técnica, usabilidad observada y aceptación del prototipo.

1.12 Relación con el protocolo de investigación

El desarrollo mantiene la correspondencia con el anteproyecto mediante la relación entre los objetivos aprobados, las actividades realizadas y las evidencias producidas. La Tabla 6 presenta esta trazabilidad y permite reconocer el aporte individual de programación evaluado en esta tesis sin separar el prototipo de su propósito educativo.

Tabla 6.
Correspondencia entre el protocolo y el desarrollo de la tesis

Elemento del protocolo	Correspondencia en el desarrollo	Evidencia verificable	Estado
Problema: mecánicas interactivas insuficientemente estructuradas.	Flujo común de acción, validación, retroalimentación, recompensa y progreso.	Ilustración 1, reglas de interacción y pruebas funcionales.	Delimitado
Objetivo general: desarrollar y evaluar la aplicación en términos de funcionamiento, usabilidad e interacción, como apoyo a la práctica de destrezas lógico-matemáticas.	Se desarrolló Crokiando en Unity para Meta Quest 3; la evaluación con usuarios no está documentalmente cerrada.	APK documentado, arquitectura, capturas y resultados agregados reportados.	Parcial
Objetivo específico 1: analizar principios y definir requerimientos pedagógicos.	Se revisaron destrezas y se definieron reconocimiento, clasificación y patrones.	Anteproyecto, marco teórico, requisitos y descripción de mecánicas.	Cumplido
Objetivo específico 2: diseñar estructura y mecánicas interactivas.	Cada nivel se definió con inicio, acción, validación, respuesta y cierre.	Diagramas, diseño de niveles, escenas y estados.	Cumplido
Objetivo específico 3: implementar la aplicación en Unity.	Se programaron manipulación, selección, validación, retroalimentación y progreso.	Scripts descritos, componentes, prefabs y capturas.	Cumplido
Objetivo específico 4: integrar recursos tridimensionales y componentes físicos.	Se configuraron modelos, colliders, rigidbodies, animadores, audio e interfaz.	Escenas integradas, configuración y pruebas de interacción documentadas.	Cumplido documentalmente

Elemento del protocolo	Correspondencia en el desarrollo	Evidencia verificable	Estado
Objetivo específico 5: evaluar usabilidad, interacción y experiencia de usuario.	Se reportan resultados agregados, pero faltan instrumento, base, caracterización y respaldos éticos previos.	Resultados agregados de 20 participantes; originales y expediente ético pendientes.	Parcial
Metodología: investigación aplicada, enfoque cuantitativo y alcance descriptivo; desarrollo iterativo.	Implementación, prueba, corrección e integración por fases; análisis descriptivo de datos agregados.	Documentación técnica y tablas de resultados; trazabilidad metodológica incompleta.	Aplicada con limitaciones documentales

Nota: elaboración propia a partir del anteproyecto (2026).

La Tabla 6 establece la trazabilidad entre el problema, los objetivos y la metodología del protocolo, las actividades realizadas y la evidencia prevista. Su lectura permite comprobar que cada objetivo se traduce en una decisión de diseño, implementación, integración o evaluación del prototipo.

Durante el desarrollo se refinó el alcance amplio del anteproyecto. El documento inicial menciona conteo, cantidad, clasificación, asociación y patrones como posibles actividades; la versión actual concentra los tres niveles en reconocimiento de figuras, clasificación por forma y continuación de patrones por forma. Este ajuste mantiene el núcleo lógico-matemático del proyecto, se registró como una decisión de alcance y fue aprobado por el tutor.

También es necesario distinguir la validación técnica de la efectividad educativa. La primera comprueba que el prototipo funciona y puede utilizarse; la segunda exige datos sobre

cambios en el aprendizaje. Esta precisión evita presentar como resultado lo que aún es un propósito.

CAPÍTULO II.

2. MARCO TEÓRICO

2.1 Antecedentes de investigación

Los antecedentes se seleccionaron por su relación directa con cuatro aspectos del proyecto: aprendizaje lógico-matemático temprano, aprendizaje basado en juegos, realidad virtual educativa e interacción segura con usuarios infantiles. La revisión es narrativa y orientada al diseño del prototipo; por tanto, no se presenta como una revisión sistemática propia. Su finalidad es comparar hallazgos existentes, reconocer sus límites y determinar qué vacío concreto aborda la propuesta.

En el ámbito del aprendizaje matemático temprano, Rittle-Johnson et al. siguieron el desarrollo de niños desde los 4 hasta los 11 años y encontraron que distintos conocimientos matemáticos iniciales contribuyen a trayectorias posteriores de aprendizaje [2]. Sarama y Clements organizaron la enseñanza temprana mediante trayectorias compuestas por una meta matemática, una progresión del pensamiento infantil y actividades ajustadas a esa progresión [27]. Estos aportes respaldan la selección de tareas observables y graduadas, pero no indican que una aplicación de realidad virtual produzca por sí sola una mejora.

Alotaibi analizó el aprendizaje basado en juegos durante la educación infantil y reportó resultados favorables en dimensiones cognitivas, motivacionales y de participación, junto con diferencias metodológicas entre los estudios revisados [3]. Plass, Homer y Kinzer explican que un juego educativo debe integrar perspectivas cognitivas, motivacionales, afectivas y socioculturales, en lugar de reducirse a puntos o recompensas [28]. Para el proyecto, esto significa que cada nivel debe contener una acción vinculada con una destreza, una regla comprensible y una respuesta que ayude a continuar.

En educación con realidad virtual, Hamilton et al. hallaron resultados generalmente prometedores, aunque destacaron diferencias en los diseños experimentales y poca evidencia sobre retención a largo plazo [6]. Radianti et al. identificaron que muchas aplicaciones inmersivas de educación superior se concentran en la experiencia tecnológica y no explican suficientemente sus fundamentos de aprendizaje [29]. Aunque esa revisión no estudia educación inicial, su observación es relevante: una experiencia inmersiva necesita alinear objetivos, interacción y evaluación.

El metaanálisis de Villena-Taranilla et al. informó un efecto positivo general de la realidad virtual en educación K-6 [4]. Sin embargo, la población, los contenidos, los dispositivos y la duración de las intervenciones fueron diversos. La revisión de Adhe et al. sobre realidad virtual en educación infantil localizó pocos estudios que cumplieran sus criterios de análisis [5]. En conjunto, estas investigaciones muestran una oportunidad de desarrollo, pero también una insuficiencia de evidencia específica para niños de 4 a 5 años.

El diseño de la interacción también modifica el valor educativo de la experiencia. Johnson-Glenberg propone que la realidad virtual educativa aproveche gestos y acciones corporales congruentes con el contenido, con un grado de movimiento que aporte significado y no solo espectáculo [30]. En contraste, Makransky, Terkildsen y Mayer observaron en una simulación científica mayor presencia, pero menor aprendizaje y mayor carga cognitiva en la condición inmersiva [31]. Este resultado impide asumir que más inmersión siempre produce más aprendizaje y respalda el uso de escenas simples, objetivos breves y estímulos controlados.

La retroalimentación constituye otro antecedente central. Shute sostiene que la retroalimentación formativa debe ofrecer información útil, oportuna y relacionada con la

tarea [9]. Hwang y Kang comprobaron que la manera de formular la retroalimentación dentro de un juego educativo genera respuestas emocionales distintas en niños de edad preescolar [10]. Por ello, el prototipo utiliza respuestas inmediatas, evita mensajes punitivos y permite un nuevo intento cuando la acción no coincide con la regla.

Las investigaciones sobre seguridad obligan a considerar la edad del usuario. Weech et al. describen la relación entre presencia y cibermareo [21], mientras que la revisión de Bexson et al. señala que la evidencia disponible sobre uso de realidad virtual en niños sigue siendo heterogénea [22]. La American Academy of Pediatrics recomienda atender la supervisión, las reacciones del niño y el equilibrio con otras actividades [23]. Estas fuentes no establecen un tiempo universal para todos los casos, de modo que el proyecto prioriza sesiones breves, pausas, observación adulta y suspensión inmediata si aparece malestar.

Como referentes de producto, Khan Academy Kids, Bini Bambini y ABCmouse presentan actividades digitales dirigidas a educación inicial, con contenidos de matemáticas, formas, colores, secuencias o progreso [32], [33], [34]. Se utilizan únicamente como comparación funcional, no como evidencia de efectividad. Su principal diferencia con la propuesta es que operan principalmente mediante interfaces bidimensionales, mientras que el prototipo transforma las tareas en selección, manipulación y colocación dentro de un espacio tridimensional.

Tabla 7.
Síntesis comparativa de antecedentes relevantes

Antecedente	Aporte principal	Aplicación al proyecto	Limitación para esta tesis
Rittle-Johnson et al. [2]	Relaciona conocimientos matemáticos tempranos con trayectorias posteriores.	Justifica trabajar clasificación, relaciones y patrones desde los 4 años.	No estudia realidad virtual ni evalúa este prototipo.
Sarama y Clements [27]	Organiza metas, progresiones de aprendizaje y actividades.	Orienta tareas breves con dificultad y ayuda progresivas.	El proyecto no implementa una trayectoria curricular completa.
Alotaibi [3]	Sintetiza aprendizaje basado en juegos en educación infantil.	Sustenta el uso del juego como contexto de práctica.	Los estudios revisados son metodológicamente diversos.
Hamilton et al. [6]	Revisa resultados y diseños de realidad virtual educativa.	Respalda evaluar diseño y resultados, no solo inmersión.	Predominan otros niveles educativos y contenidos.
Villena-Taranilla et al. [4]	Metaanálisis de resultados de aprendizaje en K-6.	Aporta evidencia general favorable para actividades breves.	No permite generalizar directamente a niños de 4 a 5 años.
Adhe et al. [5]	Revisión específica de VR en educación infantil.	Confirma el interés y la escasez de estudios específicos.	El número final de trabajos analizados fue reducido.
Johnson-Glenberg [30]	Propone interacción corporal congruente con el contenido.	Sustenta tomar, colocar y seleccionar como acciones significativas.	Es un marco de diseño, no una validación del videojuego.
Makransky et al. [31]	Muestra mayor presencia, pero posible sobrecarga y menor aprendizaje.	Justifica reducir estímulos y mantener objetivos claros.	El experimento se realizó con otro contenido y población.
Hwang y Kang [10]	Analiza cómo suena el feedback constructivo para niños.	Orienta respuestas breves, positivas y no punitivas.	No estudia interacción inmersiva.
Bexson et al. [22]	Revisa la seguridad del uso de VR en población infantil.	Sustenta supervisión, pausas y registro de malestar.	La evidencia disponible es heterogénea.

Nota: elaboración propia a partir de [2], [3], [4], [5], [6], [10], [22], [27], [29] y [30].

La Tabla 7 compara diez antecedentes por aporte, aplicación y limitación para la tesis. En conjunto, respaldan las decisiones educativas, interactivas y de seguridad, pero también muestran que sus resultados no pueden transferirse directamente al prototipo ni a niños de 4 a 5 años sin una evaluación propia.

La comparación permite identificar el vacío que atiende el proyecto: existe evidencia general sobre matemáticas tempranas, juegos educativos y realidad virtual, pero todavía son limitados los trabajos que documentan cómo convertir reconocimiento de figuras, clasificación por forma y continuación de patrones en mecánicas inmersivas simples para niños de 4 a 5 años. La contribución esperada no consiste en afirmar anticipadamente una mejora del aprendizaje, sino en construir, justificar y evaluar un prototipo cuya lógica pueda ser observada y repetida.

2.2 Bases teóricas

2.2.1 Pensamiento lógico-matemático en educación inicial

El pensamiento lógico-matemático comprende procesos de comparación, clasificación, relación, orden y reconocimiento de regularidades. En el Currículo de Educación Inicial del Ecuador, estas destrezas forman parte del ámbito de relaciones lógico-matemáticas y, para el grupo de 4 a 5 años, incluyen clasificar objetos con atributos definidos, reconocer semejanzas y diferencias y continuar patrones simples [1]. El proyecto toma estas destrezas como referencia curricular y las transforma en acciones que pueden ejecutarse dentro del entorno virtual.

La clasificación requiere identificar un atributo relevante y agrupar elementos que lo comparten. En Parque de Figuras, el atributo es la forma; por ello, el usuario debe reconocer círculo, cuadrado o triángulo y colocar el objeto en el recipiente correspondiente. La

asociación también compara atributos, pero establece una correspondencia entre dos representaciones. En Planeta Dulzura, se presenta una figura geométrica y el usuario selecciona el botón que representa esa misma forma.

Un patrón es una regularidad que puede reconocerse y continuarse. En Tren de los Patrones, la secuencia presenta una posición incompleta y el usuario debe tomar la figura que respeta la regla y colocarla en el espacio vacío. Esta actividad se relaciona con la seriación, pero no equivale a evaluar todas sus formas; el alcance se limita a la continuación de patrones visuales por forma. La teoría de trayectorias de aprendizaje sugiere relacionar la meta, el nivel de pensamiento del niño y las actividades que apoyan el siguiente avance [27].

2.2.2 Aprendizaje activo, corporal y progresivo

El prototipo adopta una orientación de aprendizaje activo: el niño no recibe únicamente una explicación, sino que observa, decide y actúa. La acción debe mantener correspondencia con la idea que se practica. Tomar una figura y colocarla según su forma representa físicamente la clasificación; seleccionar la forma que coincide con el estímulo representa el reconocimiento; y colocar el elemento faltante completa una estructura visual. Esta congruencia entre gesto y contenido coincide con los principios de aprendizaje corporal propuestos para experiencias inmersivas [30].

La progresión no se limita a pasar de una escena a otra. También se manifiesta dentro de cada actividad mediante instrucciones iniciales, intentos, retroalimentación y refuerzo. Las ayudas deben aparecer cuando son necesarias y disminuir cuando el usuario comprende la tarea. El recorrido de tres minijuegos organiza contenidos diferentes, pero no pretende representar por sí solo una secuencia curricular completa ni sustituir la intervención de un docente.

2.2.3 Aprendizaje basado en juegos y gamificación

El aprendizaje basado en juegos utiliza el juego completo como medio para alcanzar un objetivo educativo. La gamificación, en cambio, incorpora elementos como puntos, niveles o recompensas en una actividad que no necesariamente es un juego. Plass et al. explican que el diseño de un juego educativo debe considerar de manera conjunta los procesos cognitivos, la motivación, la emoción y el contexto social [28]. Esta distinción evita presentar las recompensas como el núcleo del proyecto.

En los tres niveles, la mecánica principal es la tarea lógico-matemática y la recompensa aparece después de resolverla. Los estudios de Hamari et al. y Sailer y Homner muestran que los efectos de la gamificación dependen del contexto y de su forma de integración [12], [13]. Por ello, los puntos y estímulos visuales se vinculan con una acción correcta, mientras que el error conduce a información y reintento, no a una penalización que interrumpa la práctica.

2.2.4 Inmersión, presencia e interactividad

La inmersión describe las propiedades objetivas con las que un sistema presenta y actualiza el entorno virtual, como el campo visual, el seguimiento y la respuesta en tiempo real. La presencia es la experiencia subjetiva de sentirse ubicado dentro del entorno, mientras que la interactividad representa la capacidad de modificarlo mediante acciones. Jerald reúne estos conceptos dentro de un enfoque de diseño de realidad virtual centrado en el usuario [35], y Cummings y Bailenson muestran que determinadas características inmersivas influyen en la presencia [8].

El modelo CAMIL plantea que la inmersión y el control de la interacción pueden influir en presencia, agencia, interés, motivación y carga cognitiva, variables que

posteriormente se relacionan con el aprendizaje [7]. En el prototipo, la presencia se utiliza para favorecer atención y participación, pero no se trata como resultado educativo suficiente. Una experiencia puede sentirse inmersiva y, al mismo tiempo, presentar instrucciones confusas o una mecánica irrelevante.

2.2.5 Carga cognitiva y aprendizaje multimedia

El aprendizaje multimedia parte de que las personas procesan información verbal y visual mediante capacidades limitadas. Mayer propone principios para seleccionar, organizar y presentar palabras e imágenes sin añadir procesamiento innecesario [36]. En una experiencia infantil de realidad virtual, esto se traduce en mostrar pocos elementos relevantes a la vez, mantener instrucciones breves, utilizar señales visibles y evitar que sonidos, animaciones o decoraciones compitan con el objetivo.

El hallazgo de Makransky et al. demuestra que la presencia elevada no compensa automáticamente una carga cognitiva excesiva [31]. Por ello, cada nivel trabaja una regla principal: reconocimiento de figuras, clasificación por forma o continuación de patrones. El sistema debe separar la instrucción, la acción y la respuesta; si todos los mensajes se presentan simultáneamente, el usuario puede atender al espectáculo visual sin comprender qué debe resolver.

2.2.6 Retroalimentación formativa y refuerzo positivo

La retroalimentación formativa informa al usuario sobre la relación entre su acción y la meta. Debe ser oportuna, comprensible y útil para corregir [9]. En el prototipo, una respuesta correcta activa una señal positiva y permite avanzar; una respuesta incorrecta comunica que la relación no coincide y habilita un nuevo intento. El sistema no debe limitarse

a indicar rojo o verde: siempre que sea posible, la señal debe relacionarse con el objeto, el recipiente, la figura o la posición que se está evaluando.

Debido a la edad del público, la voz, la duración y el tono del mensaje forman parte de la mecánica. Hwang y Kang muestran que diferentes formulaciones de retroalimentación generan reacciones emocionales distintas en niños [10]. En consecuencia, se priorizan mensajes cortos, lenguaje positivo y ausencia de burlas o sanciones. El refuerzo positivo reconoce el logro, pero no debe ocultar la regla que el usuario aplicó.

2.2.7 Usabilidad, experiencia de usuario y diseño centrado en el niño

ISO 9241-11:2018 organiza la usabilidad mediante efectividad, eficiencia y satisfacción dentro de un contexto de uso [14]. La efectividad corresponde a completar la tarea correctamente; la eficiencia relaciona el resultado con los recursos empleados, como tiempo, intentos o ayuda; y la satisfacción recoge la percepción del usuario. En niños de 4 a 5 años, estas dimensiones deben observarse con instrumentos sencillos y con mediación adulta.

ISO 9241-210:2019 propone comprender a los usuarios, especificar sus necesidades, producir soluciones de diseño y evaluarlas de forma iterativa [14]. El usuario infantil no debe tratarse como un adulto con textos más grandes. La interfaz requiere objetos reconocibles, distancias alcanzables, objetivos visibles, acciones consistentes y apoyo de un responsable. El cuestionario de Tcha-Tokey et al. aporta dimensiones de experiencia inmersiva [11], pero no debe aplicarse de forma directa sin adaptación y autorización.

2.2.8 Seguridad, comodidad y supervisión

El cibermareo puede incluir malestar, fatiga visual, desorientación, náusea o pérdida de equilibrio. Su aparición depende de factores personales, del contenido, del movimiento y

del dispositivo [21]. En actividades dirigidas a niños pequeños se deben evitar desplazamientos artificiales innecesarios, giros bruscos y estímulos que obliguen a mantener posturas incómodas. Las tareas principales del prototipo pueden ejecutarse desde una posición estable y dentro de un área controlada.

La evidencia de seguridad en población infantil todavía no permite establecer una regla única para todas las edades y aplicaciones [22]. Por ello, se adopta un criterio preventivo: consentimiento del representante, supervisión adulta continua, ajuste correcto del visor, área libre de obstáculos, sesiones breves, pausas y suspensión ante cualquier señal de malestar. Las herramientas de supervisión de Meta permiten establecer límites y descansos [24], pero no reemplazan la observación del adulto ni el protocolo del estudio [23].

2.3 Bases metodológicas

2.3.1 Naturaleza aplicada y alcance descriptivo

La investigación es aplicada porque utiliza conocimientos educativos y de ingeniería para resolver una necesidad concreta mediante un producto tecnológico. El Manual de Frascati diferencia la investigación aplicada del desarrollo experimental y define este último como un trabajo sistemático que aprovecha conocimiento existente para producir o mejorar productos y procesos [37]. La construcción del prototipo combina ambas orientaciones: analiza fundamentos relevantes y los transforma en un sistema funcional que debe probarse.

El alcance es descriptivo porque documenta los requisitos, las mecánicas, los componentes, el comportamiento observado y los resultados de las pruebas. El término iterativo corresponde al proceso de desarrollo, no a una categoría única de investigación. En consecuencia, la expresión aplicada, descriptiva e iterativa debe leerse como tres decisiones complementarias: finalidad práctica, alcance de análisis y forma de construir el software.

2.3.2 Ciencia del diseño y construcción del artefacto

La metodología de ciencia del diseño propuesta por Peffers et al. organiza la investigación tecnológica alrededor de la identificación del problema, los objetivos de una solución, el diseño y desarrollo del artefacto, su demostración, evaluación y comunicación [38]. Esta estructura es compatible con la tesis porque el resultado principal del área de programación es un artefacto: el sistema de mecánicas interactivas integrado en un entorno de realidad virtual.

El proyecto no adopta la ciencia del diseño como una etiqueta añadida sin evidencia. Cada actividad debe dejar un producto verificable: requisitos, diagrama, versión ejecutable, caso de prueba, registro de error, medición o conclusión. La demostración consiste en recorrer los tres minijuegos; la evaluación comprueba funcionamiento y usabilidad; y la comunicación se realiza mediante la documentación de la tesis.

2.3.3 Ingeniería de requisitos y trazabilidad

ISO/IEC/IEEE 29148:2018 describe procesos e información para la ingeniería de requisitos [25]. En el prototipo, un requisito debe relacionar la destreza, la acción del usuario, la condición que evalúa el sistema, la respuesta esperada y la evidencia de prueba. Por ejemplo: si el usuario coloca un círculo en el recipiente destinado a círculos, el sistema debe reconocer la coincidencia, activar retroalimentación positiva y actualizar el avance una sola vez.

La trazabilidad evita que una función exista únicamente porque resulta llamativa. Cada objeto interactivo, panel, sonido o animación debe corresponder a una instrucción, un estado o una respuesta. También permite comprobar cambios de alcance: el anteproyecto mencionaba conteo, cantidad, clasificación, asociación y patrones; la versión actual

concentra los tres niveles en reconocimiento de figuras, clasificación por forma y continuación de patrones. Esta decisión debe mantenerse registrada.

2.3.4 Gestión ágil con Scrum

Scrum se utiliza como apoyo para organizar el trabajo en períodos breves, revisar incrementos y adaptar el backlog. La Guía Scrum establece ciclos de inspección y adaptación alrededor de un producto que evoluciona [26]. En el proyecto, un incremento puede ser una interacción de agarre funcional, una regla de validación integrada o un nivel que completa su flujo sin errores.

Scrum no sustituye la metodología de investigación ni convierte automáticamente el desarrollo en un estudio experimental. Su función es gestionar tareas, prioridades, revisiones y decisiones del equipo. Las actividades académicas, la recolección de datos y la interpretación de resultados deben conservar sus propios criterios metodológicos.

2.3.5 Diseño iterativo, prototipado y playtesting

Fullerton propone un enfoque centrado en el jugador que combina prototipado, playtesting y revisión continua [39]. Aplicado al proyecto, una mecánica se diseña primero como flujo; luego se implementa con objetos básicos, se comprueba su regla y finalmente se integra con modelos, interfaz, sonido y animación. Este orden reduce el riesgo de invertir tiempo en acabado visual antes de demostrar que la interacción es comprensible y estable.

El playtesting observa qué hace realmente el usuario, no solo lo que el desarrollador espera que haga. Debe registrar si reconoce el objeto, encuentra la zona correcta, comprende la respuesta, solicita ayuda, repite un error o completa la actividad. Las observaciones se convierten en ajustes de distancia, escala, tiempo, señal visual, audio o regla de validación.

2.3.6 Pruebas y evaluación

Las pruebas técnicas verifican funciones, estados y condiciones límite. ISO/IEC 25010:2023 proporciona características de calidad como adecuación funcional, eficiencia de desempeño, fiabilidad, capacidad de interacción y mantenibilidad [20]. El prototipo no pretende certificarse bajo la norma, pero utiliza estas categorías para ordenar los casos de prueba y evitar que la evaluación se limite a afirmar que el juego abre o se ve bien.

La evaluación debe separar tres resultados. El funcionamiento determina si el sistema detecta y procesa la acción. La usabilidad determina si el usuario puede completar la tarea de forma efectiva, eficiente y satisfactoria [14]. El aprendizaje requiere evidencia adicional, como una medición antes y después o una comparación equivalente. Una encuesta de agrado o una sesión completada no demuestran por sí solas una mejora lógico-matemática.

Ilustración 2
Ciclo iterativo aplicado al desarrollo del prototipo



Cada iteración conserva lo verificado, corrige errores y actualiza la documentación.

Nota: elaboración propia con base en los puntos del apartado 2.3.

La Ilustración 2 representa el ciclo iterativo seguido en el proyecto: análisis, diseño, implementación, integración, pruebas y ajustes se repiten hasta consolidar un incremento funcional respaldado por evidencia de prueba.

Tabla 8.
Relación entre las bases metodológicas y sus evidencias

Base metodológica	Función dentro del proyecto	Evidencia mínima
Investigación aplicada y desarrollo experimental [37]	Transformar conocimientos existentes en un prototipo que atiende una necesidad.	Producto ejecutable y descripción de la solución.
Alcance descriptivo	Caracterizar requisitos, funcionamiento, problemas y resultados observados.	Registros, métricas simples y análisis sin inferencias no comprobadas.
Ciencia del diseño [38]	Organizar problema, objetivos, diseño, demostración, evaluación y comunicación.	Trazabilidad entre necesidad, artefacto y evaluación.
Ingeniería de requisitos [25]	Definir entradas, reglas, salidas y criterios de aceptación.	Matriz de requisitos y casos de prueba.
Scrum [26]	Gestionar tareas, prioridades, incrementos y revisiones.	Backlog, revisiones y decisiones por ciclo.
Diseño iterativo y playtesting [39]	Probar temprano, observar la interacción y ajustar.	Versiones, observaciones y correcciones justificadas.
Calidad y usabilidad [14], [20], [40].	Evaluar funcionamiento, desempeño, interacción y mantenimiento.	Casos ejecutados, resultados y evidencia en dispositivo.

Nota: elaboración propia (2026).

La Tabla 8 traduce cada base metodológica en una función y una evidencia verificable, por lo que establece el vínculo entre el enfoque de investigación y los productos que deben documentarse.

2.4 Bases tecnológicas

2.4.1 Motor Unity y modelo basado en componentes

Unity funciona como el entorno central en el que se organizan escenas, objetos tridimensionales, materiales, física, animación, audio, interfaz y scripts [15]. Su modelo se

basa en GameObjects que adquieren comportamiento mediante componentes [41]. Esta estructura permite que una figura combine representación visual, Collider, Rigidbody, componente interactivo y script de identificación sin reunir toda la lógica del proyecto en una sola clase.

La selección de Unity se justifica por la integración de herramientas XR, el desarrollo en C#, la creación de prefabs reutilizables y la compilación para Meta Quest. La versión exacta del Editor utilizada en la entrega final debe registrarse desde Unity Hub o Help > About; no se debe reemplazar por la versión más reciente de la documentación si no coincide con el proyecto.

2.4.2 Programación en C#

C# se utiliza para programar reglas, estados, eventos y comunicación entre componentes [18]. La programación orientada a objetos permite representar responsabilidades diferentes, como identificar una figura, administrar un contenedor, validar una respuesta, controlar un nivel o actualizar el progreso. La modularidad facilita corregir una regla sin modificar el resto del sistema y permite reutilizar componentes en objetos semejantes.

En Unity, los scripts suelen extender MonoBehaviour y responder a eventos del ciclo de vida o de otros componentes. Para evitar dependencias difíciles de mantener, las referencias expuestas en el Inspector deben validarse, los nombres deben expresar su responsabilidad y los eventos no deben registrar varias veces el mismo acierto. La teoría de patrones de diseño aporta soluciones reutilizables para separar estados, observadores y responsabilidades [42], aunque la tesis debe documentar únicamente los patrones realmente implementados.

2.4.3 XR Interaction Toolkit

XR Interaction Toolkit 3.0.11 es un sistema de interacción de alto nivel y basado en componentes para experiencias VR y AR [15]. Su arquitectura distingue Interactors, que representan el medio que inicia la interacción; Interactables, que representan los objetos que responden; y XR Interaction Manager, que coordina la relación entre ambos [16]. En Crokiando, los interactores corresponden a las manos rastreadas y permiten seleccionar, tomar, soltar y activar elementos virtuales.

Para la manipulación directa, los objetos incorporan componentes interactivos y físicos que responden al agarre de la mano. La interfaz se activa acercando el índice o mediante un gesto de selección. Esta configuración evita que el niño tenga que memorizar botones y mantiene la acción visible dentro del entorno.

2.4.4 OpenXR y compatibilidad de plataforma

OpenXR proporciona una interfaz estándar entre una aplicación XR y el tiempo de ejecución del dispositivo [17]. Su propósito es reducir la dependencia de APIs específicas y ofrecer conceptos comunes para sistemas, sesiones, espacios, vistas, acciones y dispositivos de entrada. En el proyecto, OpenXR constituye la capa que comunica Unity con las capacidades XR habilitadas para Meta Quest.

La compatibilidad estándar no elimina la necesidad de configurar la plataforma. Deben habilitarse el proveedor OpenXR, los perfiles de interacción, las funciones requeridas y el perfil de compilación para Quest. La documentación de Meta indica el procedimiento de preparación de Unity y de selección de la plataforma [19]. El registro final debe incluir capturas de XR Plug-in Management, Package Manager y Build Profiles.

2.4.5 Meta Quest 3 y seguimiento de manos

Meta Quest 3 es el dispositivo de referencia para ejecutar y probar el prototipo. Sus cámaras y servicios de seguimiento permiten representar la cabeza, las manos y sus movimientos dentro del entorno. Unity expone una interfaz para acceder a datos de seguimiento de manos [15], mientras Meta XR y OpenXR conectan esas poses con los componentes interactivos [17], [19]. De este modo, el niño puede acercar la mano, señalar con el índice, sujetar una figura, moverla y soltarla.

Para conservar una interacción cómoda, los elementos se ubican dentro del alcance de los brazos, con tamaños y separaciones adecuados. La iluminación del espacio, la visibilidad de las manos y el ajuste del visor se comprueban antes de cada sesión. Estas condiciones favorecen una detección estable y una experiencia comprensible para el usuario infantil.

2.4.6 Física, colisiones y zonas de validación

Los Colliders definen límites utilizados por el sistema físico y los Rigidbodies permiten que un objeto participe en la simulación. Un trigger detecta superposición sin producir necesariamente una respuesta física. Unity establece que, para generar eventos de trigger, al menos uno de los objetos del par debe poseer un cuerpo físico adecuado [43]. Este principio explica por qué una figura puede verse y moverse, pero no activar un contenedor si su configuración está incompleta.

En Parque de Figuras, el contenedor funciona como zona de detección y compara la identificación del objeto. La validación se ejecuta una sola vez para evitar duplicar el progreso y define la respuesta cuando una figura correcta o incorrecta entra en la zona. El

Collider aproxima el destino visible y mantiene coherencia entre lo que el niño observa y la regla programada.

2.4.7 Interfaz y retroalimentación dentro del espacio XR

La interfaz de un entorno XR se configura normalmente en Canvas de espacio mundial y requiere componentes de entrada compatibles con dispositivos rastreados [15]. Los botones deben mantener una distancia, escala y separación que permitan seleccionarlos sin precisión excesiva. En Planeta Dulzura, la opción seleccionada se envía a la lógica de validación; en Tren de los Patrones, se valida la figura colocada.

La retroalimentación combina cambios visuales, contornos, partículas, animaciones, audio o vibración. Cada recurso debe activarse desde el mismo resultado lógico para evitar contradicciones; por ejemplo, no debe reproducirse un sonido correcto si el contador registró un error. El contorno u outline se considera una señal de posibilidad de interacción, no una confirmación de respuesta.

2.4.8 Rendimiento y comprobación en dispositivo

La realidad virtual necesita actualizar la imagen de forma estable para conservar comodidad y respuesta. El Unity Profiler permite obtener información de CPU, GPU, memoria, renderizado y otros subsistemas, tanto en el Editor como mediante conexión con el dispositivo objetivo [45]. La medición válida para la tesis debe realizarse también en Meta Quest 3, porque el rendimiento del computador no representa el comportamiento de una compilación autónoma.

Las optimizaciones deben responder a un problema medido. Entre los aspectos a revisar se encuentran cantidad de objetos activos, complejidad de Colliders, materiales, luces, sombras, animaciones, audio y llamadas frecuentes de scripts. El objetivo no es reducir

recursos de manera indiscriminada, sino mantener la interacción, la legibilidad y el desempeño dentro de los criterios definidos.

Tabla 9.
Bases tecnológicas del prototipo

Tecnología	Función en el prototipo	Configuración o comprobación requerida
Unity [15], [41]	Integra escenas, recursos, física, UI, audio y lógica.	Registrar versión exacta del Editor y módulos instalados.
C# [19]	Implementa reglas, estados, eventos y progreso.	Revisar modularidad, referencias y duplicación de eventos.
XR Interaction Toolkit 3.0.11 [16]	Proporciona interacción directa con manos, UI e interactivables.	Conservar versión y validar muestras/Starter Assets utilizados.
OpenXR 1.1 [17]	Estandariza la comunicación con el runtime XR.	Verificar proveedor, perfiles y funciones habilitadas.
Meta Quest 3 [19], [46]	Dispositivo objetivo y entrada 6DoF.	Probar alcance, seguimiento, comodidad y compilación.
Colliders y Rigidbody [43]	Detectan contacto, superposición y comportamiento físico.	Comprobar capas, trigger, cuerpo físico y eventos únicos.
UI en espacio mundial [44]	Presenta instrucciones y opciones seleccionables.	Configurar Canvas, EventSystem, XR UI Input Module y escala.
Unity Profiler [45]	Mide comportamiento y uso de recursos.	Capturar datos en una compilación ejecutada en Quest 3.

Nota: elaboración propia a partir de documentación oficial.

La Tabla 9 identifica las tecnologías que sostienen el prototipo y la comprobación requerida para cada una; así distingue la función técnica de la evidencia necesaria para reproducir la configuración.

2.5 Arquitecturas, modelos y estándares

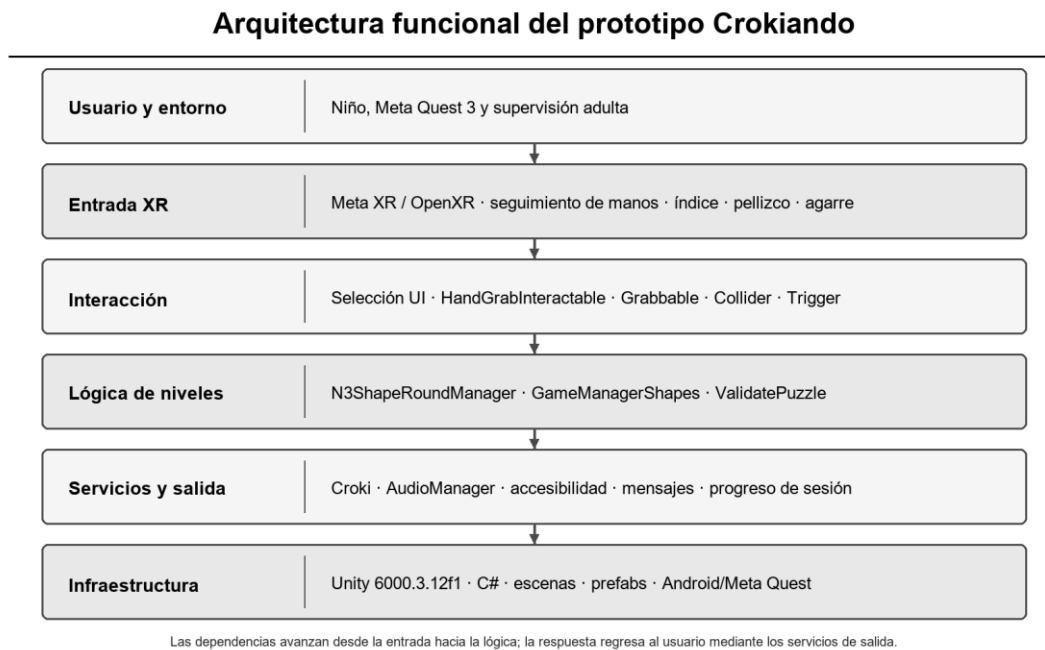
2.5.1 Arquitectura modular por capas

La arquitectura conceptual separa el sistema en capas con responsabilidades reconocibles. La capa de entrada recibe datos del visor y de las manos; la capa de interacción transforma esos datos en eventos sobre objetos; la capa de dominio aplica las reglas lógico-matemáticas; la capa de estado administra el progreso; y la capa de presentación activa

interfaz, audio, animación y recompensas. Unity y OpenXR sostienen estas capas como infraestructura.

La separación evita que la respuesta visual decida si una acción es correcta. Primero se obtiene la entrada, después se valida la regla y finalmente se notifica el resultado a los componentes de retroalimentación. De esta forma, cambiar un sonido o una animación no altera el criterio educativo. El modelo basado en componentes de Unity favorece esta distribución [41].

Ilustración 3
Arquitectura conceptual por capas del entorno inmersivo



Nota: elaboración propia a partir de la arquitectura propuesta (2026).

La Ilustración 3 organiza la solución en capas, desde el usuario y la entrada XR hasta la infraestructura de Unity, y muestra que la interacción y la lógica de los niveles dependen de servicios compartidos para producir la respuesta al usuario.

2.5.2 Modularidad de los tres minijuegos

Parque de Figuras, Planeta Dulzura y Tren de los Patrones se modelan como módulos independientes. Cada uno posee contenido, entrada y regla propias, pero comparte servicios generales de inicio, retroalimentación, puntuación, recompensa, finalización y cambio de nivel. Esta decisión permite probar un nivel sin ejecutar los demás y evita conexiones directas entre reglas que no guardan relación.

La reutilización debe producirse en comportamientos comunes, no en una clase única que conozca todos los detalles. Un validador puede recibir una respuesta y devolver un resultado; un gestor de nivel decide cuándo termina; y un sistema de progreso registra la

finalización. Las figuras, destinos y secuencias permanecen como datos específicos de cada módulo.

2.5.3 Modelo de estados

Una mecánica interactiva cambia de comportamiento según su estado. Antes de iniciar, prepara objetos y referencias; durante la instrucción limita acciones; en interacción acepta entradas; en validación compara la respuesta; en retroalimentación bloquea temporalmente nuevas acciones; y al completar registra el resultado. El patrón State descrito por Gamma et al. constituye una referencia para encapsular comportamientos dependientes del estado [42], aunque puede implementarse mediante enumeraciones o una estructura equivalente.

El control explícito de estados evita problemas frecuentes: validar antes de que la actividad inicie, sumar dos veces, aceptar entradas mientras se reproduce una respuesta, cambiar de escena antes de guardar el progreso o dejar objetos activos después de completar. La Tabla 10 representa el modelo funcional esperado; el capítulo de implementación deberá mostrar cómo se codificó realmente.

Tabla 10.
Modelo funcional de estados de una actividad

Estado	Acción permitida	Condición de salida	Control necesario
Preparación	Cargar datos, crear objetos y comprobar referencias.	Todos los componentes requeridos están disponibles.	Detener el nivel y registrar referencias faltantes.
Instrucción	Mostrar objetivo, ejemplo o señal de inicio.	El usuario inicia o finaliza el tiempo de guía.	Evitar que una entrada accidental cuente como intento.
Interacción	Tomar, colocar o seleccionar una opción.	Se recibe una respuesta evaluable.	Ignorar objetos u opciones que no pertenecen al reto.
Validación	Comparar entrada con la regla del nivel.	Se obtiene resultado correcto o incorrecto.	Ejecutar una sola validación por respuesta.
Retroalimentación	Mostrar señal, audio, animación y orientación.	Termina la respuesta o se habilita el reintento.	Bloquear entradas contradictorias durante el feedback.
Completado	Actualizar progreso y habilitar continuidad.	Se cambia de nivel, escena o menú.	Impedir duplicar puntos o perder el estado guardado.

Nota: elaboración propia con base en el flujo de mecánicas y en [42].

La Tabla 10 define los estados de preparación, instrucción, interacción, validación, retroalimentación y completado, junto con sus condiciones de salida y controles. Esta secuencia evita entradas accidentales, validaciones duplicadas y cambios de estado antes de concluir la respuesta del sistema.

2.5.4 Comunicación basada en eventos

La arquitectura de XR Interaction Toolkit utiliza eventos de hover, selección, activación y cancelación para comunicar Interactors e Interactables mediante el Interaction Manager [47]. Los scripts del proyecto pueden suscribirse a estos eventos para iniciar una

validación o actualizar una señal. Este enfoque evita consultar continuamente todos los objetos cuando solo es necesario responder a una acción concreta.

El patrón Observer aporta un principio similar: un componente emite una notificación y otros componentes interesados reaccionan sin que el emisor necesite conocer sus detalles [42]. En el prototipo, el resultado de una validación puede notificar al audio, la animación y el progreso. Para mantener fiabilidad, las suscripciones deben retirarse cuando el objeto se deshabilita y no deben acumularse tras reiniciar una escena.

2.5.5 Modelo de mecánica educativa

La unidad funcional de diseño se expresa mediante seis elementos: objetivo de aprendizaje, estímulo presentado, acción del usuario, regla de validación, retroalimentación y consecuencia. Esta estructura combina el enfoque de aprendizaje basado en juegos [28] con el diseño iterativo centrado en el jugador [39]. También facilita la trazabilidad porque cada elemento puede relacionarse con un requisito y un caso de prueba.

En Parque de Figuras, el estímulo es una forma y la acción consiste en colocarla; en Planeta Dulzura, el estímulo es una figura geométrica y la acción consiste en seleccionar círculo, cuadrado o triángulo; en Tren de los Patrones, el estímulo es una secuencia incompleta y la acción consiste en tomar la figura faltante y colocarla en el espacio vacío. Las tres mecánicas comparten validación inmediata, reintento y avance, pero no comparten la misma regla.

2.5.6 Estándares aplicables

Los estándares se utilizan como marcos de referencia, no como certificaciones obtenidas. OpenXR define interoperabilidad técnica; ISO/IEC/IEEE 29148 orienta los requisitos; ISO 9241-11 y 9241-210 orientan usabilidad y diseño centrado en las personas; e

ISO/IEC 25010 organiza la calidad del producto. La aplicación debe documentar qué partes de cada estándar fueron utilizadas y qué evidencias las respaldan.

Tabla 11.
Estándares y modelos de referencia utilizados

Estándar o modelo	Aspecto que orienta	Aplicación concreta	Alcance declarado
OpenXR 1.1 [17]	Interfaz entre aplicación y runtime XR.	Configuración de acciones, espacios y perfiles compatibles.	Uso de la especificación; no certificación de conformidad.
ISO 9241-11:2018 [14]	Efectividad, eficiencia y satisfacción en contexto.	Criterios de usabilidad y observación de tareas.	Referencia para evaluación; instrumento adaptado.
ISO 9241-210:2019 [40]	Proceso de diseño centrado en el usuario.	Comprender al niño, diseñar, probar y ajustar.	Aplicación de principios, no auditoría de cumplimiento.
ISO/IEC 25010:2023 [20]	Modelo de calidad del producto de software.	Casos sobre función, desempeño, fiabilidad e interacción.	Selección de características pertinentes.
ISO/IEC/IEEE 29148:2018 [25]	Ingeniería y trazabilidad de requisitos.	Relación entre destreza, acción, regla y prueba.	Guía documental para el proyecto.

Nota: elaboración propia a partir de [14], [17], [20], [25] y [40].

La Tabla 11 presenta OpenXR y cuatro normas ISO/IEC como marcos para la interoperabilidad, la usabilidad, el diseño centrado en el usuario, la calidad y la trazabilidad de requisitos. Su aplicación se limita a aspectos pertinentes del proyecto y no implica certificación o conformidad integral.

2.6 Herramientas de desarrollo

La selección de herramientas responde a cinco criterios: compatibilidad con Meta Quest 3, soporte para interacción XR, posibilidad de programación modular, integración con recursos tridimensionales y disponibilidad de mecanismos de prueba. La Tabla 12 distingue

las herramientas utilizadas directamente en la programación de aquellas que apoyan la integración o la gestión del proyecto.

Las versiones indicadas como confirmadas corresponden a datos ya establecidos en el Capítulo I. Cuando la versión exacta no aparece en las fuentes disponibles, se marca como pendiente de registro. Esta decisión evita inventar información y facilita que el capítulo de implementación se complete con capturas de Unity Hub, Package Manager, Visual Studio, Blender y el cliente de control de versiones.

Tabla 12.
Herramientas de desarrollo y justificación de selección

Herramienta	Uso dentro del proyecto	Justificación de selección	Versión o evidencia
Unity Engine [15]	Creación de escenas, lógica, física, UI, audio y compilación.	Integra el ciclo completo y dispone de soporte XR.	Registrar versión exacta del Editor.
XR Interaction Toolkit [16]	Agarre, selección directa, botones e interacción con las manos.	Sistema oficial de alto nivel basado en componentes.	3.0.11 confirmada.
OpenXR [17]	Conexión estandarizada con el runtime del visor.	Reduce dependencia de una API de entrada exclusiva.	Especificación 1.1.61 citada.
C# [18]	Programación de reglas, estados, eventos y progreso.	Lenguaje integrado con Unity y orientación a objetos.	Registrar versión asociada al Editor.
Visual Studio [18]	Edición, IntelliSense, análisis y depuración de scripts.	Incluye herramientas específicas para proyectos Unity.	Registrar edición y versión instalada.
Meta Quest 3 [19], [46]	Ejecución autónoma y prueba de entrada 6DoF.	Dispositivo objetivo definido para el prototipo.	Modelo confirmado; conservar evidencia de build.
Blender [48]	Preparación o ajuste de recursos antes de importarlos.	Admite modelado, rigging, animación y exportación 3D.	Registrar versión usada por el equipo.
Git / GitLab / SourceTree [49]	Historial de versiones, ramas y colaboración.	Permite recuperar cambios y documentar incrementos.	Registrar repositorio, ramas y commits pertinentes.
Unity Profiler [45]	Medición de CPU, GPU, memoria y subsistemas.	Permite perfilar el Editor y el dispositivo conectado.	Adjuntar capturas y condiciones de medición.
Matrices de prueba	Registro de entrada, resultado esperado, resultado real y evidencia.	Vinculan requisitos con verificación repetible.	Plantilla propia controlada por versión.

Nota: elaboración propia con base en [15], [16], [17], [18], [19], [44] y [49].

La Tabla 12 identifica las herramientas de producción, interacción XR, programación, modelado, control de versiones, perfilado y prueba, y justifica su función dentro del flujo de desarrollo. La columna de versión o evidencia también señala los datos que deben conservarse para reproducir la configuración.

Blender se incluye como herramienta. Su responsabilidad es importar recursos, comprobar escala, orientación, pivote, materiales, animadores y compatibilidad con la interacción. Esta delimitación mantiene coherencia con el alcance aprobado.

Git, GitLab y SourceTree permiten conservar el historial del proyecto. Un commit útil debe representar un cambio comprobable y describir su propósito; por ejemplo, corrección de validación, ajuste de Collider o integración de retroalimentación. El control de versiones no sustituye el registro de pruebas, pero aporta evidencia temporal y facilita recuperar una versión estable [49].

2.7 Marco conceptual

El marco conceptual establece el significado operativo de los términos utilizados en la tesis. Las definiciones no pretenden agotar cada campo; delimitan cómo se emplea cada concepto dentro del proyecto. Esta precisión permite distinguir términos cercanos, como aprendizaje basado en juegos y gamificación, inmersión y presencia, o clasificación y asociación.

Tabla 13.
Conceptos educativos y de diseño de juego

Concepto	Definición operativa	Aplicación en la propuesta
Pensamiento lógico-matemático	Procesos para comparar, relacionar, clasificar, ordenar y reconocer regularidades.	Finalidad educativa común de los tres niveles [1], [2].
Clasificación	Agrupación de objetos según un atributo o criterio compartido.	Colocar cada figura en el contenedor correspondiente a su forma.
Asociación	Establecimiento de correspondencia entre dos representaciones que comparten un atributo.	Relacionar la figura mostrada con el botón de la misma forma.
Seriación	Organización de elementos según una relación de orden.	Concepto amplio relacionado; el alcance se limita a continuar patrones.
Patrón	Regularidad formada por una unidad o regla que permite anticipar el elemento siguiente.	Tomar la figura faltante y colocarla en la posición correspondiente del tren.
Trayectoria de aprendizaje	Meta, progresión del pensamiento y actividades ajustadas al avance del niño [27].	Orienta dificultad, ayudas y secuencia de tareas; no se implementa como currículo completo.
Mecánica de juego	Regla que define qué puede hacer el usuario y cómo responde el sistema.	Tomar y colocar, reconocer una figura o completar una secuencia.
Minijuego	Actividad breve con objetivo, reglas, inicio, desarrollo y finalización propios.	Cada nivel funciona como módulo independiente.
Aprendizaje basado en juegos	Uso de una experiencia de juego completa para practicar o alcanzar un objetivo educativo [28].	La resolución lógico-matemática constituye el núcleo del reto.
Gamificación	Uso de puntos, niveles, recompensas u otros elementos de juego en una actividad [12], [13].	Refuerzo, progreso y recompensas después de cumplir la regla.
Retroalimentación formativa	Información que ayuda a comprender el resultado y orientar la siguiente acción [9].	Respuesta inmediata ante acierto o error, seguida de avance o reintento.

Concepto	Definición operativa	Aplicación en la propuesta
Refuerzo positivo	Señal que reconoce una respuesta o conducta esperada.	Animación, sonido, punto o recompensa vinculados con el acierto.
Objetivo de aprendizaje	Desempeño específico que la actividad busca practicar.	Clasificar por forma, reconocer figuras o continuar un patrón.
Evidencia de aprendizaje	Dato que permite sostener un cambio en conocimiento o desempeño.	Requiere medición adecuada; no equivale a agrado o finalización.

Nota: elaboración propia con base en [1], [2], [3], [9], [12], [13], [27] y [28].

La Tabla 13 delimita el significado operativo de los conceptos educativos y de diseño de juego utilizados en la tesis y muestra su aplicación específica en Crokiando. Esta distinción evita equiparar clasificación, asociación, seriación, gamificación, retroalimentación y evidencia de aprendizaje.

Tabla 14.
Conceptos de realidad virtual y desarrollo técnico

Concepto	Definición operativa	Aplicación en la propuesta
Realidad virtual	Entorno digital tridimensional presentado y actualizado según la acción del usuario.	Medio de ejecución de los tres minijuegos.
Inmersión	Propiedades objetivas del sistema que envuelven perceptivamente al usuario [8], [35].	Visor, seguimiento, imagen estereoscópica y respuesta en tiempo real.
Presencia	Experiencia subjetiva de sentirse ubicado dentro del entorno virtual [7], [8].	Variable de experiencia; no se asume como aprendizaje.
Interactividad	Capacidad del usuario para producir cambios mediante acciones.	Tomar, mover, colocar y seleccionar.
Usabilidad	Grado de efectividad, eficiencia y satisfacción en un contexto de uso [14].	Criterio para observar la realización de cada tarea.
Experiencia de usuario	Percepciones y respuestas resultantes del uso o uso anticipado de un sistema.	Comprensión, comodidad, emoción, participación y reacción al feedback.

Concepto	Definición operativa	Aplicación en la propuesta
Cibermareo	Malestar asociado a la exposición a entornos virtuales, con síntomas visuales o vestibulares [21].	Motiva sesiones breves, pausas y suspensión ante síntomas.
Playtesting	Prueba de una experiencia jugable mediante observación de usuarios y revisión del diseño [39].	Detecta dudas, errores de interacción y necesidades de ajuste.
Prototipo funcional	Versión que permite ejecutar y comprobar las funciones principales.	Recorrido completo de los tres niveles con validación y progreso.
6DoF	Seguimiento de rotación y desplazamiento en tres ejes espaciales [47].	Representa la cabeza y las manos dentro del espacio virtual.
GameObject	Objeto fundamental de una escena Unity que actúa como contenedor de componentes [41].	Representa figuras, contenedores, cámaras, UI y gestores.
Componente	Unidad funcional que añade datos o comportamiento a un GameObject [41].	Collider, Rigidbody, Interactable, Animator o script.
Prefab	Configuración reutilizable de un GameObject y sus componentes.	Permite instanciar objetos interactivos con una estructura consistente.
Interactor	Componente que intenta iniciar o mantener una interacción XR [16], [47].	Mano directa, índice o gesto de selección.
Interactable	Objeto que puede recibir hover, selección o activación [16], [47].	Figura, botón u opción del minijuego.
Collider	Volumen utilizado para detectar contacto o superposición.	Delimita figuras y zonas de respuesta.
Rigidbody	Componente que integra un objeto en el sistema físico.	Permite movimiento y generación de eventos con Colliders.
Trigger	Collider configurado para detectar superposición sin respuesta física obligatoria [43].	Activa la validación al ingresar en un contenedor.
Estado	Condición del sistema que determina acciones y transiciones válidas.	Preparación, interacción, validación, feedback y completado.
Evento	Notificación emitida cuando ocurre una acción o cambio.	Conecta selección, validación, feedback y progreso.

Concepto	Definición operativa	Aplicación en la propuesta
OpenXR	Estándar de acceso a plataformas y dispositivos XR [17].	Capa de compatibilidad entre Unity y el runtime de Quest.

Nota: elaboración propia con base en [7], [8], [14], [15], [17], [21], [35], [41], [42], [43], [44].

La Tabla 14 reúne los conceptos de realidad virtual y los componentes técnicos empleados para describir el prototipo, desde inmersión, presencia y usabilidad hasta GameObject, Collider, eventos y OpenXR. Las definiciones permiten interpretar la interacción y la arquitectura sin atribuir aprendizaje a variables de experiencia o funcionamiento.

Estas definiciones delimitan la interpretación de los resultados. Si el usuario completa una tarea, se puede informar efectividad de uso; si expresa agrado, se puede informar satisfacción; y si el sistema mantiene una ejecución estable, se puede informar desempeño técnico. Ninguno de estos datos, por separado, permite afirmar una mejora del aprendizaje.

2.8 Relación entre la teoría y la propuesta tecnológica

El marco teórico respalda decisiones concretas del prototipo. La relación no es decorativa: cada fundamento debe traducirse en una regla de diseño, un componente o una evidencia de evaluación. La Tabla 15 presenta esta correspondencia y permite identificar qué parte de la propuesta depende de cada base.

*Tabla 15.
Correspondencia entre fundamentos y decisiones tecnológicas*

Fundamento	Decisión incorporada en la propuesta	Evidencia prevista
Currículo de Educación Inicial [1]	Seleccionar reconocimiento de figuras, clasificación por forma y patrones simples.	Reglas y contenido de los tres niveles.
Trayectorias matemáticas [2], [27]	Organizar metas observables, ayuda y progresión de tareas.	Secuencia de instrucciones, intentos y dificultad.
Aprendizaje corporal [30]	Representar la clasificación mediante tomar y colocar objetos.	Observación de manipulación directa en Parque de Figuras.

Fundamento	Decisión incorporada en la propuesta	Evidencia prevista
Aprendizaje basado en juegos [3], [28]	Hacer que la tarea educativa sea la mecánica principal.	Cada reto exige aplicar la regla antes de obtener recompensa.
Gamificación [12], [13]	Vincular puntos, niveles y estímulos con logros reales.	Registro de progreso sin duplicación de puntuación.
CAMIL y presencia [7], [8]	Usar inmersión e interacción para favorecer participación, sin tratarlas como aprendizaje.	Medidas separadas de experiencia y desempeño.
Carga cognitiva y multimedia [31], [36]	Mostrar una regla por nivel, pocas opciones y señales breves.	Revisión de escenas, UI e instrucciones.
Feedback formativo [9], [10]	Responder inmediatamente, explicar el resultado y permitir reintento.	Eventos visuales/sonoros asociados con cada validación.
Diseño centrado en el usuario [14], [40]	Ajustar escala, alcance, lenguaje y ayuda al público infantil.	Playtesting, observación y correcciones documentadas.
Seguridad infantil [21]-[24]	Mantener posición estable, supervisión, pausas y criterio de suspensión.	Protocolo, consentimiento y registro de malestar.
Arquitectura modular [41], [42], [43]	Separar entrada, reglas, estados, feedback y progreso.	Componentes reutilizables y pruebas independientes por nivel.
Calidad y requisitos [20], [25]	Definir criterios de aceptación y casos de prueba trazables.	Matriz de requisitos, errores y resultados.
Desarrollo iterativo [26], [38], [39]	Construir, probar, corregir e integrar por incrementos.	Backlog, versiones, commits y evidencias de revisión.
Unity, XRI y OpenXR [15]-[19]	Implementar interacción y compilar para Meta Quest 3.	Configuración de paquetes y build ejecutable.

Nota: elaboración propia (2026).

La Tabla 15 vincula cada fundamento curricular, pedagógico, interactivo, técnico y de calidad con una decisión concreta y con la evidencia prevista. Esta correspondencia muestra cómo la teoría se transforma en reglas de los niveles, criterios de interfaz, controles de seguridad, arquitectura modular y pruebas trazables.

En el Nivel 1, el reconocimiento de figuras se transforma en una selección de interfaz que debe ser visible y fácil de accionar. En el Nivel 2, la teoría de clasificación y la interacción corporal convergen en una mecánica de tomar y colocar. En el Nivel 3, el niño toma la figura faltante y la coloca en el espacio vacío de la secuencia. Los tres niveles utilizan la misma estructura de validación, retroalimentación y progreso, pero conservan reglas independientes.

La arquitectura tecnológica permite materializar estas decisiones. XR Interaction Toolkit captura la interacción; los scripts en C# convierten la entrada en una respuesta evaluable; los validadores aplican la regla; y los componentes de presentación comunican el resultado. OpenXR y Meta Quest 3 proporcionan la plataforma de ejecución, mientras que las pruebas comprueban si la cadena completa funciona.

La teoría también establece límites. La realidad virtual no garantiza aprendizaje, la presencia no sustituye una medición educativa, la recompensa no reemplaza la mecánica y una encuesta de satisfacción no demuestra efectividad pedagógica. Por ello, las conclusiones deberán corresponder a los datos obtenidos. Si la evaluación solo demuestra funcionamiento y usabilidad, la tesis debe informar esos resultados sin convertirlos en una afirmación de mejora cognitiva.

En síntesis, el marco teórico sostiene una propuesta en la que el valor educativo depende de la relación entre objetivo, acción, regla, retroalimentación y evaluación. El

siguiente desarrollo de la tesis deberá mostrar cómo estos fundamentos se concretaron en requisitos, scripts, componentes, escenas, pruebas y resultados del prototipo.

CAPÍTULO III.

3. MANUAL DEL USUARIO

3.1 Presentación de la propuesta y perfil del usuario

Crokiando es un prototipo educativo de realidad virtual desarrollado en Unity para apoyar el aprendizaje lógico-matemático de niños de 4 a 5 años. La experiencia permite observar, seleccionar, tomar, mover y colocar elementos tridimensionales mediante seguimiento de manos, mientras el sistema comprueba cada acción y responde de forma inmediata.

El contenido se organiza en tres minijuegos independientes. El usuario selecciona cualquiera de ellos al colocar una pelota en el baúl correspondiente; no necesita terminar un nivel para abrir otro. Antes de cada actividad, el personaje guía Croki presenta una explicación breve mediante animación. Después de la explicación comienza el reto, se registran las respuestas y, al finalizar, se actualiza el progreso del jugador.

Objetivo del manual: explicar cómo preparar el equipo, iniciar la aplicación, interactuar con las manos, resolver los tres minijuegos, interpretar la retroalimentación y cerrar correctamente la sesión.

Modalidad de uso: experiencia individual, local y sin conexión de red durante la actividad, con acompañamiento permanente de un adulto.

Dispositivo principal: visor Meta Quest 3 con seguimiento de manos activado.

3.1.1 Componentes de la propuesta

Tabla 16.
Componentes de la propuesta

Nivel	Actividad	Acción principal del niño	Forma de avance
Nivel 1	Planeta Dulzura	Observar la figura y seleccionar con el índice la opción geométrica coincidente.	Responder las rondas mostradas por el sistema.
Nivel 2	Parque de Figuras	Tomar círculos, cuadrados y triángulos con la mano y colocarlos en el contenedor de la misma forma.	Completar la clasificación dentro del tiempo disponible.
Nivel 3	Tren de los Patrones	Observar la secuencia, tomar la figura faltante y colocarla en el espacio vacío del tren.	Completar las secuencias y formar el progreso del tren.

Nota: elaboración propia a partir del Proyecto.

La Tabla 16 compara los tres niveles por actividad, acción del niño y condición de avance, y confirma que cada minijuego conserva una regla propia dentro de un flujo común.

3.1.2 Perfil de los usuarios

Tabla 17.
Perfil de los usuarios

Perfil	Responsabilidad	Características y necesidades
Usuario infantil	Realizar las actividades.	Niño de 4 a 5 años. No requiere experiencia previa en realidad virtual. Necesita instrucciones cortas, objetos visibles, tiempo para responder y refuerzo positivo.
Adulto facilitador	Preparar y supervisar la sesión.	Docente, investigador, padre, madre o responsable. Ajusta el visor, confirma el seguimiento de manos, vigila el área y ayuda únicamente cuando el niño lo necesita.
Operador técnico	Iniciar y comprobar el prototipo.	Persona que verifica la aplicación, la conexión del visor, la escena inicial y el correcto funcionamiento en la computadora utilizada para la demostración.

Nota: elaboración propia a partir del Proyecto.

La Tabla 17 diferencia al usuario infantil, al adulto acompañante y al equipo responsable, junto con las necesidades que condicionan la interacción y la supervisión.

3.1.3 Requisitos mínimos para utilizar la experiencia

Tabla 18.
Requisitos mínimos para utilizar la experiencia

Elemento	Requisito
Visor	Meta Quest 3 cargado y configurado.
Seguimiento de manos	Hand tracking activado y manos visibles para las cámaras del visor. Las actividades no requieren controladores.
Área	Zona despejada, iluminada y sin muebles, cables, personas u objetos dentro del alcance de los brazos.
Aplicación	Versión de Crokiando instalada en el visor o proyecto preparado para ejecución mediante Meta Quest Link.
Computadora, si se usa Link	Equipo con Windows, Meta Quest Link, puerto USB de datos y el proyecto de Unity disponible.
Versión de desarrollo	Unity 6000.3.12f1. Paquetes instalados: Meta XR All-in-One SDK, MR Utility Kit, Core SDK, Haptics SDK, Interaction SDK, Interaction SDK Essentials y Platform SDK (201.0.0); Meta XR Audio SDK y Meta - Voice SDK - Immersive Voice Commands (85.0.0); Quick Outline (1.1).
Conectividad	No se necesita Internet durante el juego una vez que el visor y la aplicación están configurados.
Supervisión	Presencia permanente de un adulto durante toda la experiencia.

Nota: elaboración propia a partir del Proyecto.

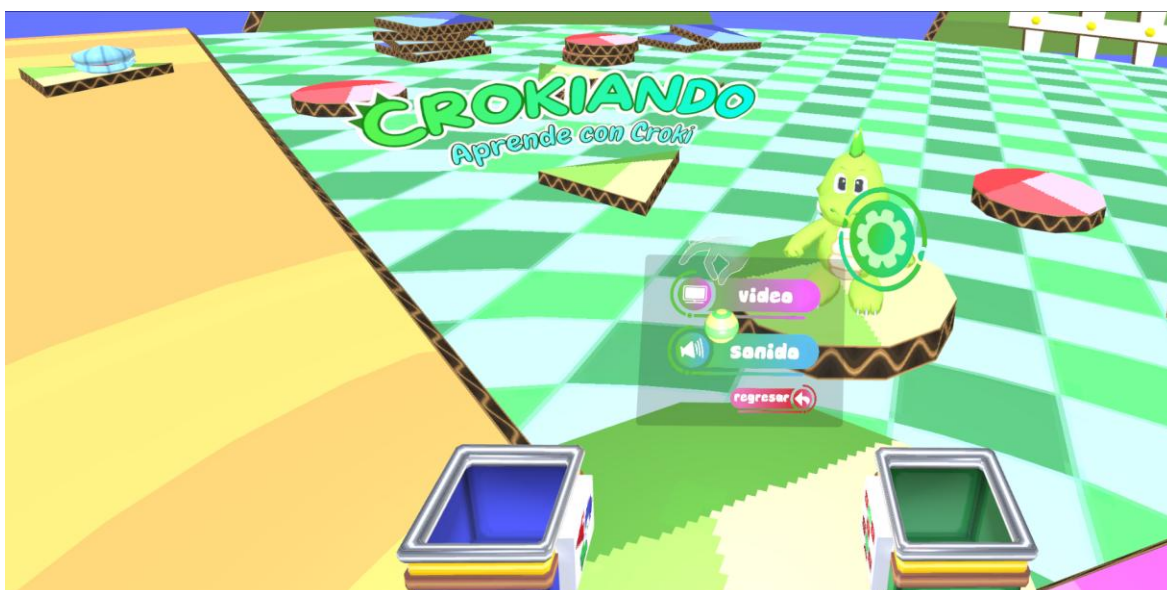
La Tabla 18 reúne los requisitos de visor, espacio, seguimiento de manos, audio y supervisión que deben verificarse antes de iniciar una sesión.

3.1.4 Condiciones de uso seguro

- El adulto debe despejar el área antes de colocar el visor al niño.
- El visor debe quedar firme, pero sin apretar la frente, las mejillas o la parte posterior de la cabeza.
- Antes de iniciar, el adulto debe confirmar que las cámaras externas estén limpias y que las manos aparezcan correctamente dentro del entorno.

- El niño debe permanecer en la zona marcada y evitar correr, saltar o realizar movimientos bruscos.
- El volumen debe permitir escuchar las indicaciones sin impedir que el niño oiga al adulto.
- La sesión debe detenerse si aparece cansancio, mareo, dolor de cabeza, visión borrosa, miedo o desorientación.
- Al retirar el visor, el adulto debe sostenerlo para evitar que caiga al suelo.

Ilustración 4
Vista general de Crokiando



Nota: elaboración propia a partir de capturas del prototipo Crokiando (2026).

La captura presenta el entorno principal de Crokiando, con el personaje guía, los accesos a los minijuegos y los controles audiovisuales que conforman el punto de inicio de la experiencia.

3.1.5 Lista de comprobación de seguridad y registro

La preparación y el cierre se organizan por roles. La duración máxima, los descansos y los criterios de exclusión deben definirse en el protocolo institucional aprobado; no se asigna un valor no documentado en los materiales disponibles.

- ADULTO — Antes: comprobar autorización institucional y del representante, ausencia de malestar, limpieza del visor, carga disponible, iluminación suficiente, área libre de obstáculos y reconocimiento estable de las manos.
- ADULTO — Antes: ajustar el visor sin presión excesiva, explicar la señal de pausa y confirmar que el niño puede escuchar y responder al acompañante.
- NIÑO — Antes: permanecer en el área indicada, comunicar cualquier incomodidad y realizar únicamente la acción explicada por la guía.
- ADULTO — Durante: mantener supervisión continua, evitar ayudas que revelen la respuesta, registrar duración, pausas, orden de niveles, pérdidas de seguimiento, agarres fallidos, falsas selecciones e incidencias.
- NIÑO — Durante: señalar, sujetar y soltar con movimientos lentos; detenerse cuando el adulto lo indique o cuando aparezca malestar.
- ADULTO — Después: cerrar la aplicación, retirar y limpiar el visor, comprobar el estado del niño y completar el registro de sesión sin datos identificables innecesarios.

3.2 Acceso e inicio de la experiencia

El acceso se divide en dos partes: primero, el adulto prepara el visor y el espacio; después, el operador abre la aplicación y confirma el seguimiento de manos. Crokiando puede ejecutarse como una aplicación instalada en Meta Quest 3 o desde una computadora mediante Meta Quest Link.

3.2.1 Preparación previa

- Cargar el visor y comprobar que el seguimiento de manos esté habilitado.
- Limpiar suavemente los lentes y las cámaras externas con un paño adecuado.
- Retirar obstáculos del área de juego y dejar espacio para extender ambos brazos.

- Encender la iluminación del lugar y evitar luces directas o reflejos intensos sobre las cámaras.
- Mantener las manos libres y visibles frente a las cámaras del visor.
- Mantener una silla cercana únicamente para el adulto; no debe invadir el límite de juego.

3.2.2 Encendido, ajuste y centrado del visor

- Mantener presionado el botón de encendido del Meta Quest 3 hasta que aparezca la imagen del sistema.
- El adulto sostiene el visor, afloja las correas y lo coloca con cuidado sobre la cabeza del niño.
- Ajustar primero la correa superior y después las correas laterales hasta obtener una imagen clara.
- Pedir al niño que abra y cierre las manos frente al visor para confirmar que la representación virtual responde a sus movimientos.
- Crear o confirmar el límite de seguridad que muestra el visor. El niño debe comenzar cerca del centro.
- Si la vista aparece desplazada, el adulto debe utilizar la opción de recentrado del sistema antes de iniciar la actividad.

Comprobación antes de continuar. El niño debe ver con claridad, reconocer sus manos virtuales y extender los brazos sin tocar objetos reales. Si alguna condición no se cumple, el adulto debe ajustar nuevamente el visor y la posición inicial.

3.2.3 Opción A: abrir la versión instalada en Meta Quest 3

- Abrir la biblioteca de aplicaciones del visor.

- Buscar Crokiando. Si se trata de una compilación de prueba, revisar la sección de aplicaciones de desarrollo u orígenes desconocidos.
- Señalar el nombre o icono de Crokiando con la mano virtual.
- Acercar el índice o realizar el gesto de selección para abrir la aplicación.
- Esperar sin retirar el visor mientras carga el entorno principal.
- Comprobar que se escucha el audio, que Croki aparece y que el selector de niveles está visible.

3.2.4 Opción B: ejecutar mediante Meta Quest Link

- Encender el Meta Quest 3 y abrir Meta Quest Link en la computadora.
- Conectar el visor con un cable USB-C que transmita datos. Un cable que solo carga no permite la comunicación.
- Colocarse el visor y aceptar los permisos de conexión que aparezcan.
- Activar Quest Link desde el visor y esperar a que la computadora lo reconozca.
- Abrir la misma versión del proyecto Crokiando en Unity 6000.3.12f1.
- Confirmar que la escena inicial o el menú principal sea la escena abierta.
- Verificar que OpenXR y el perfil de Meta Quest estén activos, y que XR Interaction Toolkit no muestre errores.
- Presionar Play en Unity. No mover, desconectar ni cerrar el visor mientras se carga la experiencia.
- Comprobar dentro del visor que aparecen las manos virtuales, el audio y el menú de selección.

Ilustración 5
Conexión del Meta Quest 3 con la computadora



Nota: elaboración propia a partir de capturas del prototipo Crokiando (2026).

La fotografía documenta la conexión física del Meta Quest 3 con la computadora de desarrollo, condición necesaria para ejecutar, supervisar y depurar el prototipo desde Unity.

3.2.5 Ejecución en cualquiera de las dos computadoras de desarrollo

Para que la demostración sea igual en las dos computadoras, ambas deben abrir la misma versión del proyecto. Antes de cambiar de equipo, el operador debe cerrar Unity en la computadora anterior, guardar los cambios y comprobar que el proyecto utilizado en la segunda computadora esté actualizado.

Tabla 19.
Ejecución en diferentes computadoras

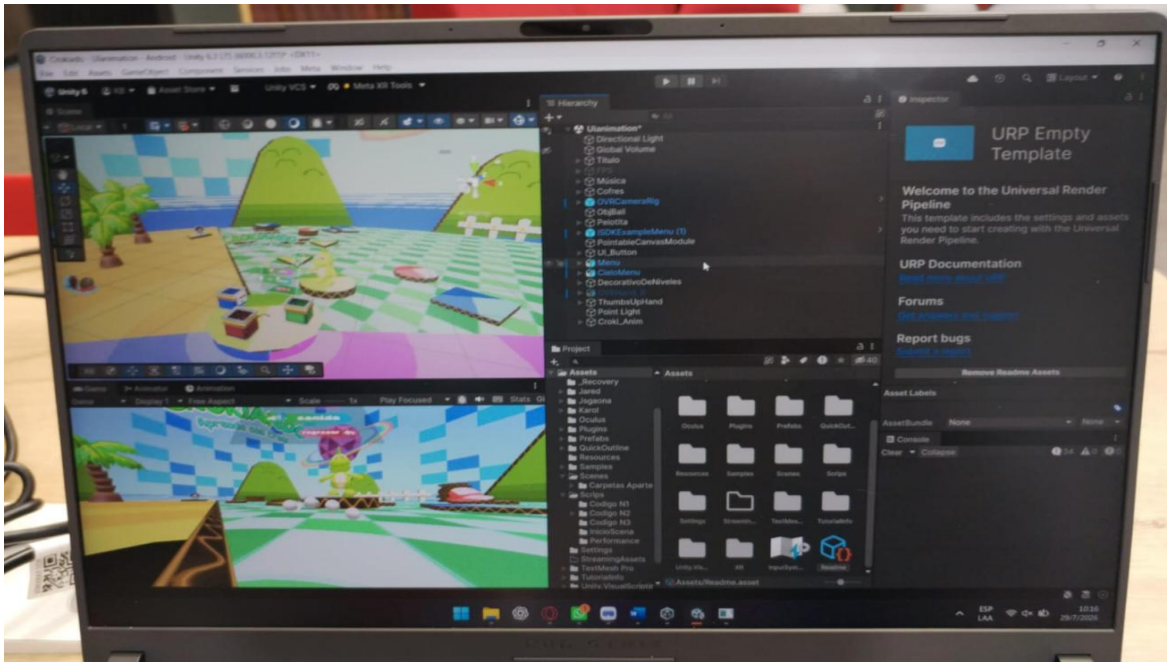
Comprobación	Computadora 1	Computadora 2
Proyecto	Abrir la copia actual de Crokiando.	Abrir exactamente la misma copia o versión sincronizada.
Unity	Usar Unity 6000.3.12f1.	Usar Unity 6000.3.12f1.
Paquetes XR	Meta XR 201.0.0; Meta XR Audio y Voice 85.0.0; Quick Outline 1.1.	Las mismas versiones y configuración.
Escena	Abrir la escena inicial definida para la demostración.	Abrir la misma escena inicial.
Plataforma	Comprobar Android y la configuración de Meta Quest.	Repetir la misma comprobación.
Visor	Conectar y verificar seguimiento de manos, audio y escena inicial.	Conectar y realizar la misma comprobación.

Nota: elaboración propia a partir de datos del proyecto

La Tabla 19 establece las comprobaciones que deben repetirse en ambas computadoras para asegurar una ejecución equivalente: misma copia del proyecto, versión de Unity, paquetes XR, escena inicial, plataforma y visor. Cualquier diferencia en estos elementos puede alterar el comportamiento observado durante la demostración.

Carpetas del proyecto. Al trasladar o sincronizar el proyecto se deben conservar Assets, Packages y ProjectSettings. Las carpetas temporales generadas por Unity, como Library, Temp y Logs, pueden volver a crearse y no deben utilizarse para decidir cuál es la versión más reciente.

Ilustración 6
Verificación de Crokiando en el entorno de desarrollo



Nota: elaboración propia a partir de capturas del prototipo Crokiando (2026).

La imagen muestra el proyecto abierto en Unity, con las vistas de escena y de juego activas; esta disposición permite verificar simultáneamente la configuración del entorno y su respuesta durante la ejecución.

3.2.6 Inicio de una sesión de juego

- Esperar a que el entorno principal termine de cargar.
- Escuchar o mirar el saludo inicial de Croki.
- Observar las tres opciones: Planeta Dulzura, Parque de Figuras y Tren de los Patrones.
- Tomar una pelota de selección con la mano.
- Colocar la pelota en el baúl correspondiente al nivel deseado y esperar la carga.
- Mirar la explicación animada de Croki antes de tocar los objetos del reto.

- Cuando la explicación finalice, comenzar la actividad indicada.

Ilustración 7
Selector principal de minijuegos



Nota: elaboración propia a partir de capturas del prototipo Crokiando (2026).

La captura muestra el selector principal, donde las opciones de nivel se representan mediante objetos diferenciados que el usuario manipula para iniciar la actividad elegida.

3.2.7 Cierre de la aplicación

- Terminar la actividad actual o regresar al selector principal tocando con la mano el botón disponible en la interfaz.
- Para una pausa, el adulto puede abrir el menú del sistema mediante el gesto habilitado o el botón físico del visor.
- Seleccionar la opción para cerrar o salir de Crokiando.
- Si la sesión se ejecuta desde Unity, detener Play en el editor después de cerrar la experiencia.
- Retirar el visor sujetándolo con ambas manos, apagarlo si no volverá a utilizarse y guardarlo.

3.2.8 Solución de problemas de acceso

Tabla 20
Solución de problemas de acceso

Problema	Comprobación	Acción recomendada
El visor no aparece en la computadora.	Cable, puerto USB y permiso de datos.	Usar un cable de datos, cambiar de puerto, volver a aceptar el permiso y reiniciar Meta Quest Link.
Quest Link no inicia.	Visor encendido y función Link habilitada.	Desconectar, reiniciar visor y computadora, conectar de nuevo y activar Link.
Las manos virtuales no aparecen.	Iluminación, cámaras externas y posición de las manos.	Mejorar la iluminación, limpiar las cámaras y mantener las manos abiertas frente al visor.
La imagen está desplazada.	Posición inicial del usuario.	Volver al centro del área y utilizar la opción de recentrado del sistema con ayuda del adulto.
No se puede seleccionar un botón.	Índice sobre el botón y panel visible.	Acercar el índice al centro, esperar el resaltado y tocar o realizar un pellizco breve.
No se puede tomar un objeto.	Mano virtual cerca del objeto.	Acercar la mano abierta y cerrarla sobre la figura o realizar el gesto de pellizco.
La aplicación queda en negro.	Carga, escena inicial y errores en Unity.	Esperar unos segundos; si continúa, cerrar, verificar la escena inicial y abrir nuevamente.
Se pierde el seguimiento.	Iluminación y cámaras externas.	Detener la sesión, mejorar la luz, limpiar las cámaras y volver a definir el límite.

Nota: elaboración propia a partir de datos del proyecto

La Tabla 20 vincula los problemas de acceso más frecuentes con su comprobación y la acción correctiva correspondiente, para apoyar la puesta en marcha del prototipo.

3.3 Navegación mediante seguimiento de manos

La interacción se basa en movimientos naturales de cabeza y manos. Para completar las actividades no es necesario utilizar mandos, joysticks ni combinaciones de botones. El niño mira el entorno, señala, toca, toma, mueve y suelta los elementos según la indicación de Croki.

3.3.1 Interacción principal con las manos

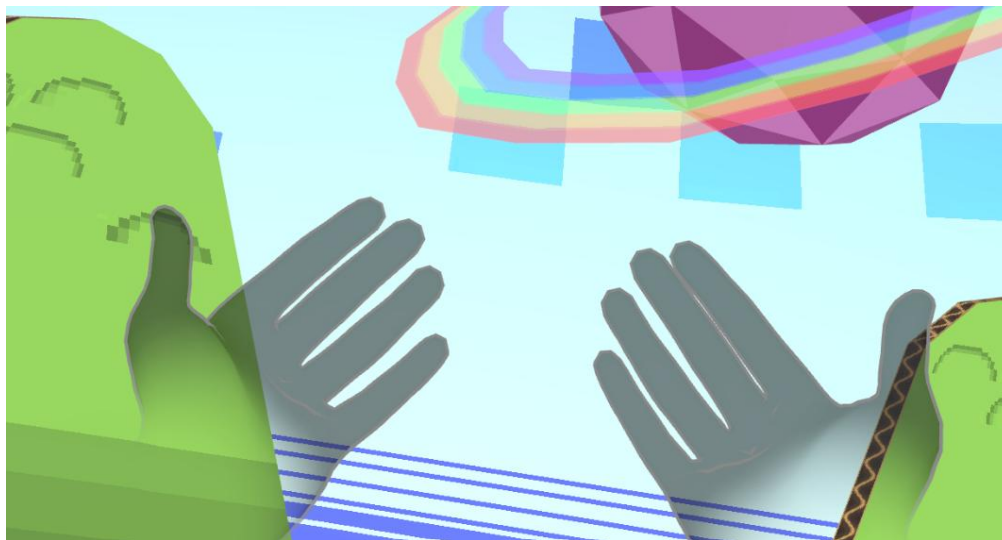
Tabla 21
Interacción principal con las manos

Gesto o movimiento	Acción	Uso dentro de Crokiando
Movimiento de la cabeza	Mirar alrededor.	Localizar a Croki, los paneles y los objetos.
Movimiento de las manos	Acercar y orientar las manos.	Señalar, tocar y trasladar figuras.
Cerrar la mano o pellizcar	Sujetar un objeto.	Tomar una figura interactiva.
Abrir la mano	Soltar el objeto.	Depositar la figura en un contenedor.
Acercar el índice	Tocar o seleccionar.	Activar botones de respuesta en Planeta Dulzura.
Menú del sistema	Acción supervisada por el adulto.	Pausar, salir o recentrar mediante el gesto disponible o el botón físico del visor.

Nota: elaboración propia a partir de datos del proyecto

La Tabla 21 explica la correspondencia entre gestos, acciones del sistema y uso dentro de Crokiando, y funciona como guía operativa para la interacción sin controles.

Ilustración 8
Representación de las manos mediante seguimiento de manos



Nota: elaboración propia a partir de capturas del prototipo Crokiando (2026).

La imagen evidencia la representación virtual de las manos detectadas por el visor y su correspondencia espacial con los elementos interactivos del entorno.

3.3.2 Selección directa con el índice

- Levantar la mano y mantenerla dentro del campo de visión del visor.
- Extender el índice y acercarlo lentamente al centro del botón virtual.
- Comprobar que el botón cambia de estado, se ilumina o muestra un resaltado.
- Tocar el botón con el índice o realizar un pellizco breve y esperar la respuesta.
- Si no responde, retirar la mano, volver a acercarla con un movimiento lento y repetir una sola vez.

Ilustración 9
Selección de un botón con el índice



Nota: elaboración propia a partir de capturas del prototipo Crokiando (2026).

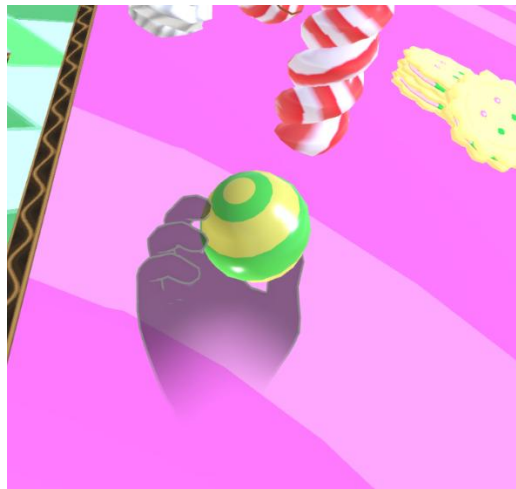
La captura ilustra la selección directa de un botón virtual mediante el índice, mecanismo utilizado para activar opciones sin controles físicos.

3.3.3 Toma, movimiento y colocación de objetos

- Mirar la figura que se desea mover.
- Acercar la mano abierta hasta tocar o rodear la figura virtual.
- Cerrar la mano o realizar el gesto de pellizco para sujetarla.
- Mover la mano despacio; la figura debe acompañar el movimiento.
- Llevar la figura hasta el contenedor o espacio indicado.
- Cuando la figura esté dentro del área correcta, abrir la mano para soltarla.
- Esperar la señal del sistema antes de tomar otra figura.

Señal de ayuda. En Parque de Figuras, el contorno visual permite reconocer el área de colocación. El usuario debe acercar y soltar el objeto suavemente dentro del contenedor, sin lanzarlo.

Ilustración 10
Proceso de tomar una figura



Nota: elaboración propia a partir de capturas del prototipo Crokiando (2026).

La imagen muestra la fase de agarre, en la que la mano virtual sujeta una figura y permite trasladarla dentro del espacio de juego.

Ilustración 11
Proceso de colocar una figura



Nota: elaboración propia a partir de capturas del prototipo Crokiando (2026).

La captura muestra la liberación de la figura dentro del contenedor de destino, acción que desencadena la comprobación de correspondencia.

3.3.4 Interpretación de las señales del sistema

Tabla 22
Interpretación de las señales del sistema

Señal	Significado	Qué debe hacer el usuario
Contorno o resaltado	El objeto o destino puede utilizarse.	Acercar la figura o apuntar al elemento resaltado.
Marca verde, sonido alegre o animación positiva	La respuesta es correcta.	Esperar el avance y continuar con el siguiente elemento.
Marca roja, sonido breve o animación de corrección	La respuesta no coincide.	Intentar nuevamente; no existe una penalización que impida continuar.
Croki animado	Presenta una instrucción, recordatorio o felicitación.	Mirar y escuchar antes de seguir.
Temporizador	Tiempo restante de Parque de Figuras.	Continuar clasificando sin lanzar los objetos.
Indicador de ronda	Avance dentro de Planeta Dulzura o Tren de los Patrones.	Resolver la ronda actual para abrir la siguiente.
Estrella, punto, trofeo o elemento del tren	Recompensa o progreso alcanzado.	Reconocer que el reto se completó y esperar el siguiente menú.

Nota: elaboración propia a partir de datos del proyecto

La Tabla 22 interpreta las señales visuales, sonoras y animadas y precisa la respuesta esperada del usuario ante cada una.

3.3.5 Pausa, regreso y recentrado supervisados

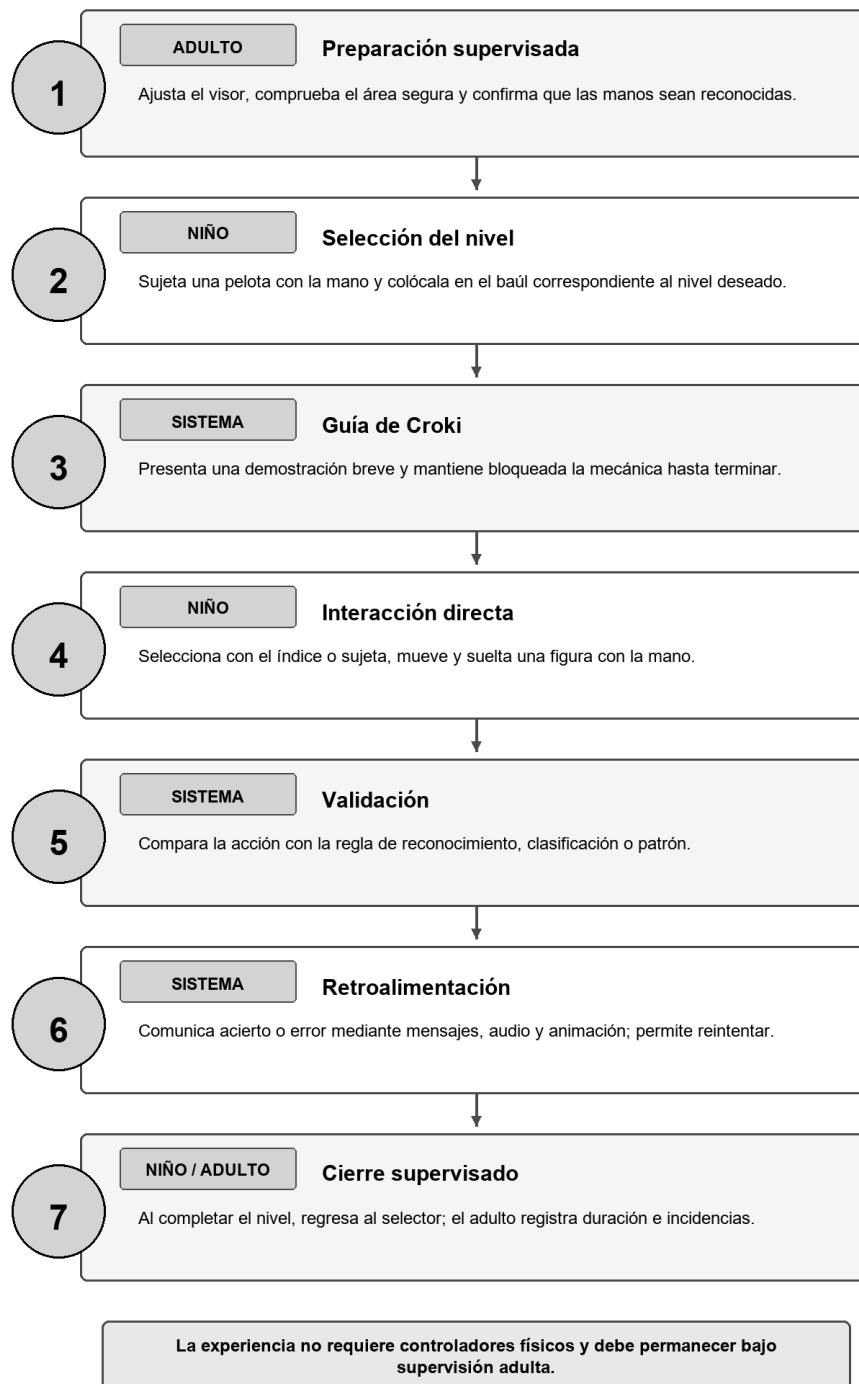
- Para cambiar de minijuego, utilizar con la mano el botón de regreso disponible en la interfaz.
- Para una pausa inmediata, el adulto puede abrir el menú del sistema o retirar cuidadosamente el visor.
- Para corregir una vista desplazada, colocar al niño en el centro del área y utilizar la opción de recentrado del visor.
- Después de recentrar, verificar que los objetos principales queden al frente y a una altura cómoda.
- No retirar el visor mientras el niño sostiene un objeto virtual; primero debe abrir la mano, soltarlo y bajar los brazos.

3.4 Funciones e interacciones disponibles

Las funciones del sistema siguen una secuencia común: el usuario selecciona un nivel al colocar una pelota en el baúl correspondiente, Croki explica la regla, el niño realiza una acción, la aplicación comprueba el resultado, presenta retroalimentación y actualiza el progreso. La respuesta del sistema aparece inmediatamente para que el niño relacione su acción con el resultado.

Ilustración 12
Secuencia de interacción mediante seguimiento de manos

Secuencia de interacción mediante seguimiento de manos



Nota: elaboración propia a partir de capturas del prototipo Crokiando (2026).

El diagrama resume la secuencia supervisada de uso: preparación, selección del nivel, guía de Croki, interacción, validación, retroalimentación y cierre con registro.

3.4.1 Selector de nivel y guía de Croki

Función: permitir que el usuario elija libremente Planeta Dulzura, Parque de Figuras o Tren de los Patrones.

Interacción: tomar una pelota y colocarla en el baúl correspondiente al nivel deseado. Antes de iniciar, Croki muestra una animación corta con la acción que se debe realizar.

Resultado esperado: la actividad comienza únicamente después de presentar la instrucción, y el niño identifica qué objeto debe tomar o qué opción debe seleccionar.

Ilustración 13
Explicación de Croki antes de un minijuego



Nota: elaboración propia a partir de capturas del prototipo Crokiando (2026).

La captura muestra a Croki presentando la tarea y los objetos relevantes antes de habilitar la interacción, con el fin de orientar la acción esperada.

3.4.2 Nivel 1: Planeta Dulzura

- Objetivo: reconocer la forma de la figura presentada y seleccionar la opción correspondiente: círculo, cuadrado o triángulo.
- Modo de juego: selección directa con la mano, organizada por rondas.
- Observar la figura u objeto mostrado en el centro de la actividad.

- Identificar si corresponde a un círculo, un cuadrado o un triángulo antes de elegir una respuesta.
- Revisar las opciones geométricas disponibles.
- Acercar el índice a la opción que coincide con la figura.
- Tocar el botón virtual o realizar un pellizco breve y esperar la validación.
- Si la selección es correcta, se activa una señal positiva y comienza la siguiente ronda.
- Si la selección es incorrecta, se presenta una señal de orientación y la ronda continúa disponible.
- Continuar hasta completar las rondas indicadas por la interfaz.
- Ayuda para el adulto. En caso necesario, pedir al niño que compare la figura con cada opción sin indicarle directamente la respuesta.

Ilustración 14
Reconocimiento de figuras en Planeta Dulzura



Nota: elaboración propia a partir de capturas del prototipo Crokiando (2026).

La escena presenta una figura de referencia y varias opciones geométricas; el niño debe reconocer su forma y seleccionar la alternativa coincidente.

3.4.3 Nivel 2: Parque de Figuras

Objetivo: clasificar objetos según su forma: círculo, cuadrado o triángulo.

- Modo de juego: toma y colocación mediante seguimiento de manos, con tiempo visible.
- Observar las figuras disponibles y los contenedores identificados con cada forma.
- Elegir una figura, acercar la mano abierta y cerrar los dedos o realizar un pellizco para sujetarla.
- Buscar el contenedor que tiene la misma forma. El contorno visual ayuda a identificar el destino.
- Introducir la figura en el contenedor y abrir la mano para soltarla.
- Esperar la validación. Una colocación correcta activa una señal positiva y registra el elemento.
- Si el contenedor no corresponde, la figura no se contabiliza y el niño puede volver a intentarlo.
- Repetir la acción hasta completar la clasificación o hasta que finalice el tiempo.
- Al terminar, esperar la felicitación y la recompensa antes de regresar al selector.
- Comportamiento del sistema. La zona de validación compara la forma de la figura con el contenedor y actualiza el progreso una sola vez por objeto.

Ilustración 15
Validación correcta en Parque de Figuras



Nota: elaboración propia a partir de capturas del prototipo Crokiando (2026).

La marca verde y la animación de Croki comunican que la figura fue colocada en el destino correcto, mientras los indicadores actualizan el resultado de la ronda.

3.4.4 Nivel 3: Tren de los Patrones

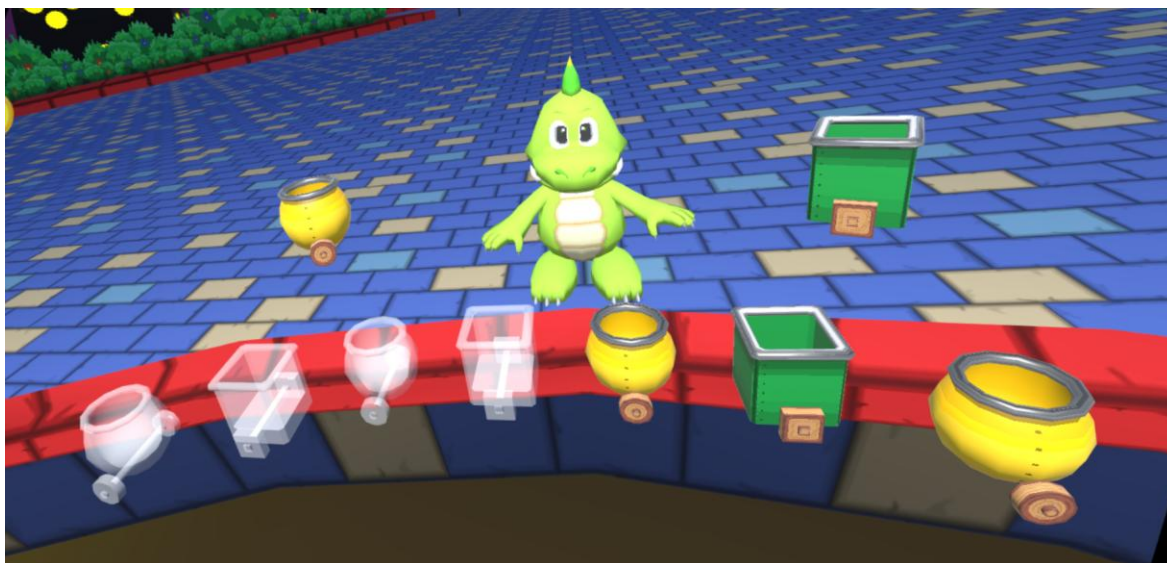
- Objetivo: reconocer la regla de una secuencia de figuras e identificar el elemento que debe continuarla.
- Modo de juego: toma y colocación de la figura faltante por rondas.
- Observar de izquierda a derecha la secuencia presentada.
- Nombrar o reconocer mentalmente las figuras que se repiten, por ejemplo: círculo, cuadrado, círculo, cuadrado.
- Identificar el espacio vacío o el signo de interrogación.

- Revisar las figuras disponibles como respuesta.
- Tomar con la mano la figura que continúa la secuencia, llevarla al espacio vacío del tren y soltarla.
- Si la respuesta es correcta, el sistema completa el espacio, muestra una señal positiva y avanza a la siguiente ronda.
- Si la respuesta es incorrecta, el patrón no se completa y el niño puede tomar otra figura e intentar colocarla.
- Continuar hasta completar todas las secuencias y observar la recompensa final del tren.

Ejemplo. Si la secuencia muestra círculo - cuadrado - círculo - ?, la respuesta esperada es cuadrado porque las dos formas se alternan.

Ilustración 16

Resolución de una secuencia en Tren de los Patrones



Nota: elaboración propia a partir de capturas del prototipo Crokiando (2026).

La captura muestra una secuencia de objetos sobre el tren y el espacio que debe completarse, lo que permite identificar la regularidad y escoger la pieza faltante.

3.4.5 Validación, reintento y retroalimentación

Tabla 23. Validación, reintento y retroalimentación

Situación	Respuesta de Crokiando	Continuación
Acción correcta	Señal visual positiva, sonido o animación; se registra el progreso.	Se habilita el siguiente objeto, pregunta o ronda.
Acción incorrecta	Señal de corrección clara y breve.	La misma acción puede repetirse sin bloquear la actividad.
Objeto fuera del destino	No se valida la colocación.	El usuario vuelve a tomarlo y lo lleva al contenedor correcto.
Ronda completada	Se actualiza el indicador y aparece el nuevo reto.	El usuario escucha la instrucción y continúa.
Nivel completado	Croki felicita, se muestra una recompensa y se guarda el progreso de la sesión.	El usuario vuelve al selector o elige otra actividad.

Nota: elaboración propia a partir de la configuración del proyecto (2026).

La Tabla 23 diferencia la respuesta del sistema ante una acción correcta, incorrecta o incompleta y muestra cómo cada resultado conduce al avance o al reintento.

La retroalimentación visual, auditiva y animada comunica el resultado, pero no decide la respuesta. La lógica de cada nivel comprueba la figura, el destino o el patrón; después, el sistema de presentación muestra el mensaje correspondiente. Esta separación evita que una animación active por error un avance.

3.4.6 Progreso y finalización

Parque de Figuras muestra el tiempo y los elementos correctamente clasificados.

Planeta Dulzura y Tren de los Patrones muestran el avance por rondas.

Cada acierto puede activar una estrella, un punto, una animación o un sonido de refuerzo.

El nivel termina únicamente cuando se cumple su condición de finalización.

La pantalla o animación final confirma que el reto fue superado.

Después del resultado, el usuario puede regresar al selector y elegir otro minijuego.

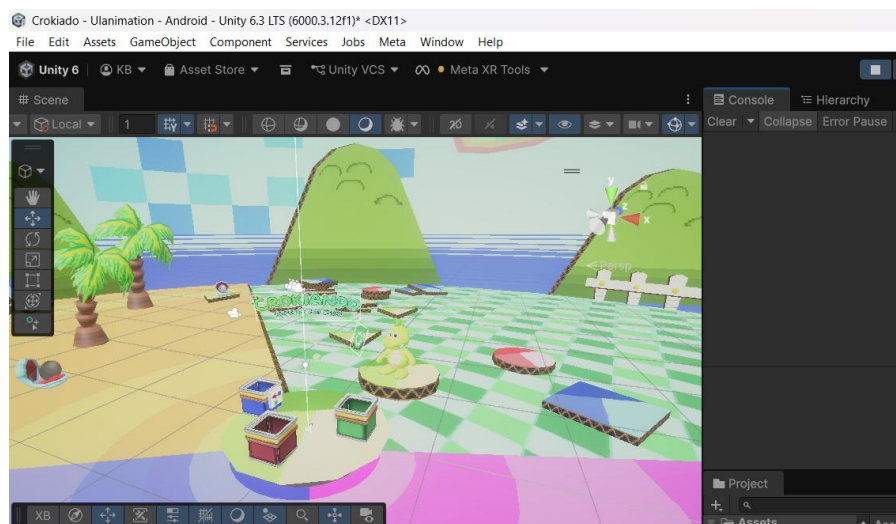
3.5 Evidencia de uso del prototipo

Las capturas siguientes documentan el entorno de desarrollo, la apertura del prototipo, la selección del nivel, la guía de Croki y el funcionamiento de las tres mecánicas. Se conservaron imágenes legibles y centradas en la acción que se desea demostrar.

La secuencia visual permite comprobar la correspondencia entre la explicación del manual y la respuesta del sistema: selección con la mano, manipulación directa, validación, retroalimentación y progreso.

Ilustración 17

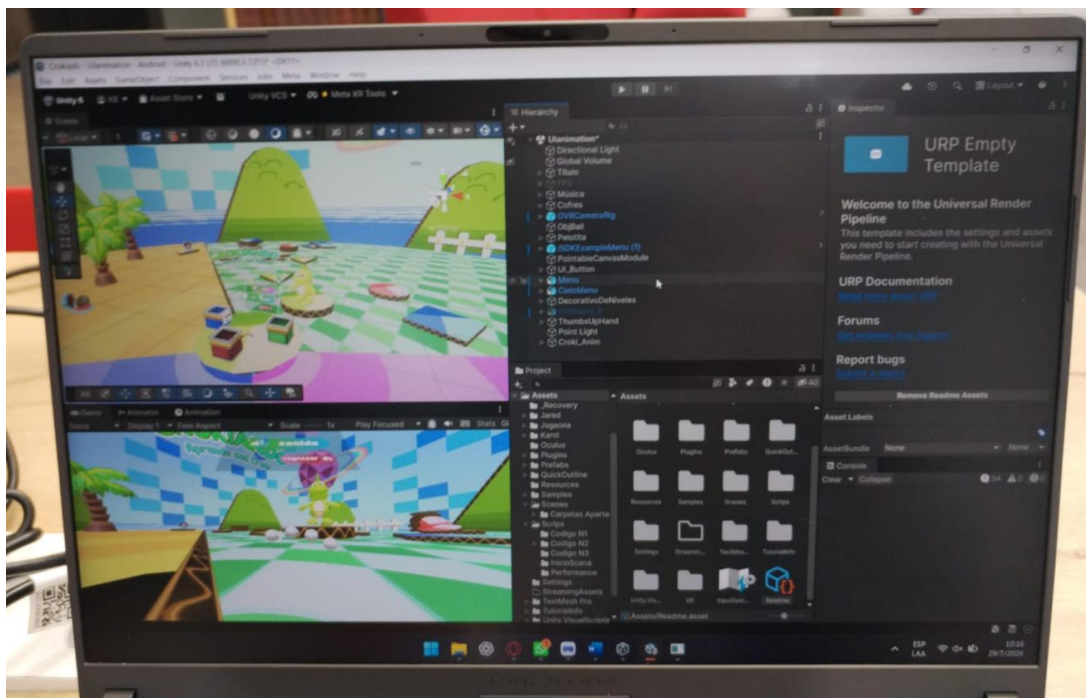
Proyecto abierto en la computadora 1



Nota: elaboración propia a partir de capturas del prototipo Crokiando (2026).

La imagen documenta la ejecución del proyecto en la primera computadora mediante el Editor de Unity, incluyendo la jerarquía, el Inspector y la vista del prototipo.

Ilustración 18
Proyecto abierto en la computadora 2



Nota: elaboración propia a partir de capturas del prototipo Crokiando (2026).

La fotografía documenta la apertura del mismo proyecto en la segunda computadora y permite comprobar la disponibilidad de las mismas escenas y recursos de desarrollo.

Ilustración 19
Apertura de Crokiando



Nota: elaboración propia a partir de capturas del prototipo Crokiando (2026).

La captura registra la apertura del entorno principal de Crokiando y confirma la carga de la interfaz, el personaje guía y los accesos a las actividades.

Ilustración 20
Entorno principal y selección de nivel



Nota: elaboración propia a partir de capturas del prototipo Crokiando (2026).

La imagen muestra el espacio de selección de nivel y los objetos manipulables asociados a cada minijuego, desde los cuales se inicia la navegación.

Ilustración 21
Explicación animada de Croki



Nota: elaboración propia a partir de capturas del prototipo Crokiando (2026).

La captura evidencia la guía animada de Croki, que presenta visualmente la acción antes de que el gestor habilite la mecánica.

Ilustración 22
Reto de reconocimiento de figuras en Planeta Dulzura



Nota: elaboración propia a partir de capturas del prototipo Crokiando (2026).

La escena muestra el reto de reconocimiento de formas de Planeta Dulzura, con una figura objetivo, opciones de respuesta e indicadores de aciertos y errores.

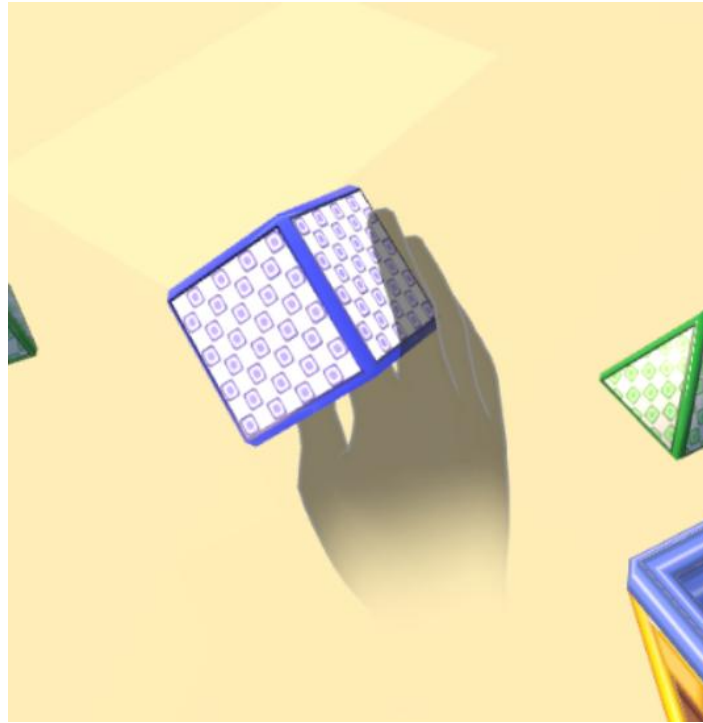
Ilustración 23
Respuesta y avance de ronda en Planeta Dulzura



Nota: elaboración propia a partir de capturas del prototipo Crokiando (2026).

La captura evidencia la respuesta del sistema tras una selección y la actualización del tiempo, los contadores y la animación de retroalimentación.

Ilustración 24
Toma de una figura en Parque de Figuras



Nota: elaboración propia a partir de capturas del prototipo Crokiando (2026).

La imagen muestra el agarre de una figura en Parque de Figuras y su traslado hacia los cofres que funcionan como destinos de clasificación.

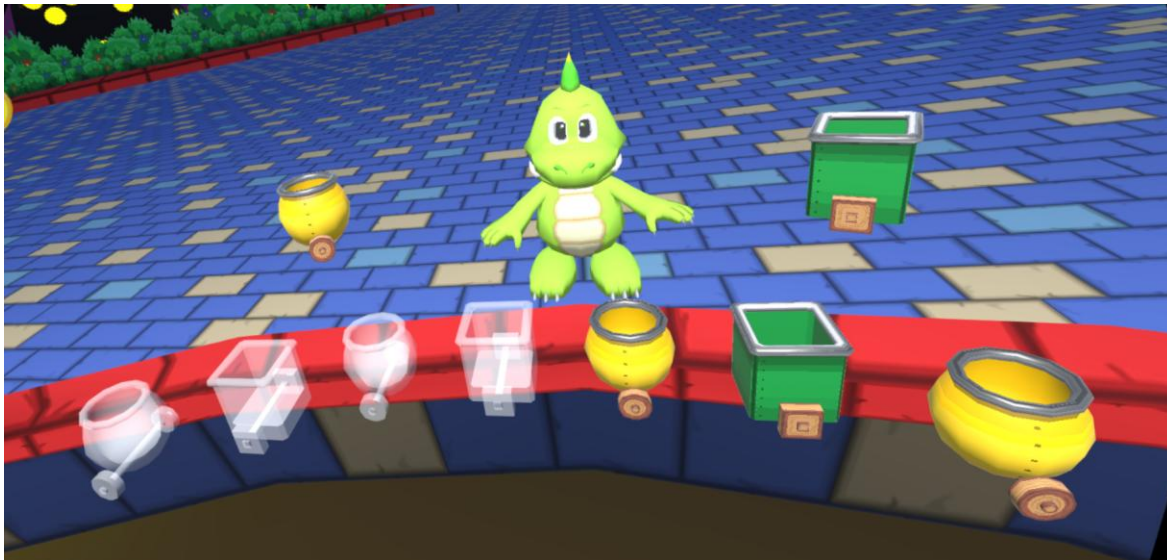
Ilustración 25
Colocación y validación en Parque de Figuras



Nota: elaboración propia a partir de capturas del prototipo Crokiando (2026).

La marca de confirmación y la celebración de Croki indican que la figura fue depositada en el cofre correspondiente y que la validación fue correcta.

Ilustración 26
Patrón incompleto en Tren de los Patrones



Nota: elaboración propia a partir de capturas del prototipo Crokiando (2026).

La escena presenta un patrón incompleto en los vagones, de modo que el usuario debe observar la alternancia e identificar la figura necesaria.

Ilustración 27
Patrón resuelto y progreso del tren



Nota: elaboración propia a partir de capturas del prototipo Crokiando (2026).

La captura muestra el patrón completado, la distribución de las piezas en el tren y la retroalimentación positiva que confirma el avance.

Ilustración 28
Progreso general



Nota: elaboración propia a partir de capturas del prototipo Crokiando (2026).

La imagen resume el progreso acumulado mediante los contadores de respuestas correctas e incorrectas y la reacción del personaje guía.

CAPÍTULO IV.

4. MANUAL DEL PROGRAMADOR

4.1 Arquitectura general del sistema

4.1.1 Estructura arquitectónica

Crokiando se desarrolló mediante una arquitectura basada en escenas y componentes, propia de Unity. Cada escena reúne modelos tridimensionales, interfaces, zonas de colisión, animaciones, fuentes de audio y scripts necesarios para ejecutar una actividad. Una parte importante del comportamiento se configura desde el Inspector mediante referencias serializadas; así, un recurso visual, un punto de aparición o un sonido puede sustituirse sin reescribir la regla educativa.

La arquitectura se organiza en cuatro capas funcionales. La primera captura la posición y el movimiento de las manos mediante Meta XR y OpenXR. La segunda convierte los gestos en selección, agarre, desplazamiento y liberación. La tercera contiene la lógica de cada nivel y compara la acción con la respuesta esperada. La cuarta comunica el resultado por medio de Croki, sonidos, animaciones, mensajes, contadores y cambios visuales.

La arquitectura funcional general se presenta en el marco teórico; en el capítulo técnico se complementa con componentes, estados y secuencias verificables.

4.1.2 Distribución de la lógica y flujo de sesión

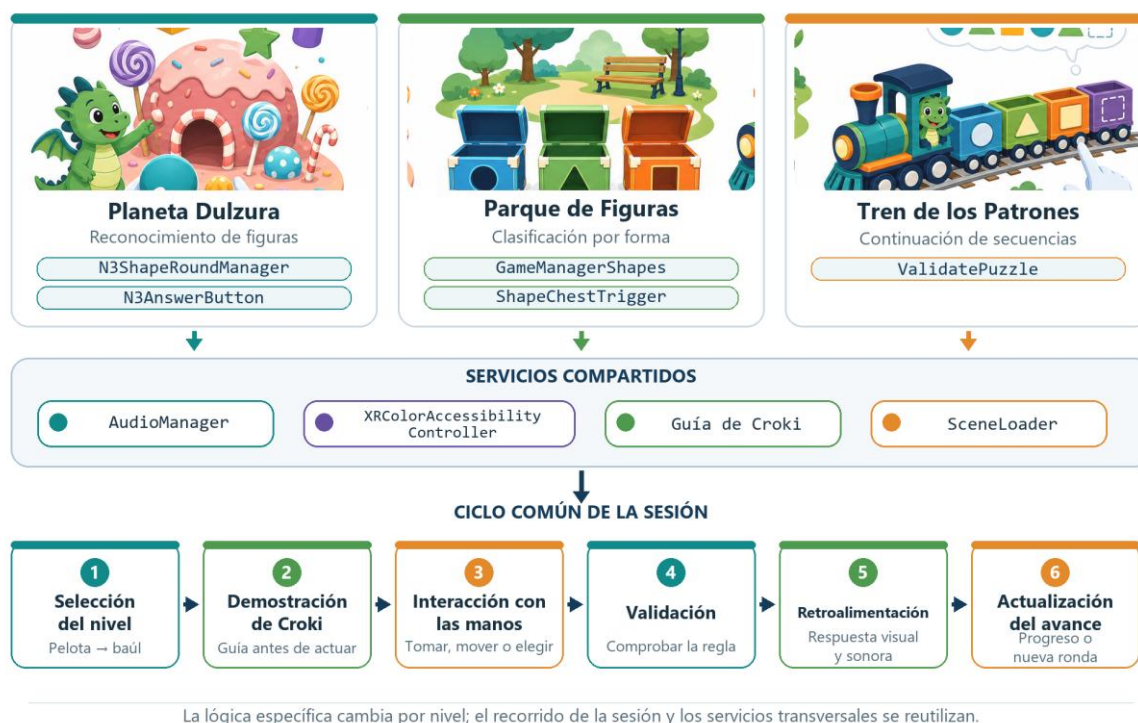
Cada nivel posee un gestor independiente. Planeta Dulzura utiliza N3ShapeRoundManager y N3AnswerButton; Parque de Figuras se coordina mediante GameManagerShapes y valida los destinos con ShapeChestTrigger; y Tren de los Patrones emplea ValidatePuzzle para organizar rondas y comprobar la secuencia. Los servicios

compartidos —AudioManager, XRColorAccessibilityController, los controladores de guía y SceneLoader— permanecen separados de esas reglas.

Una sesión sigue el mismo ciclo general: colocación de una pelota en el baúl del nivel, demostración de Croki, habilitación de la interacción, acción con las manos, validación, retroalimentación y actualización del avance. Esta regularidad reduce el número de instrucciones nuevas entre actividades y permite que cada gestor se concentre en su propia regla.

La Ilustración 1 presenta el flujo funcional general del prototipo; la ilustración siguiente sintetiza la distribución de responsabilidades y el ciclo común de sesión, que después se detalla mediante los diagramas técnicos del capítulo.

Ilustración 29
Gestores, servicios compartidos y ciclo de sesión



Nota: elaboración propia a partir de la arquitectura y del flujo funcional implementados en Crokiando (2026).

La ilustración relaciona cada nivel con el gestor que contiene su regla específica y separa los servicios transversales de audio, accesibilidad, guía y navegación. El ciclo inferior muestra que las tres actividades reutilizan la misma secuencia, desde la selección del nivel hasta la actualización del avance.

4.2 Tecnologías, motores y herramientas utilizadas

El prototipo se desarrolló en Unity 6, versión 6000.3.12f1, con C# para la lógica de juego, navegación, retroalimentación y servicios. El proyecto integra Meta XR All-in-One SDK 201.0.0 y OpenXR como base de la experiencia inmersiva. La representación visual se apoya en Universal Render Pipeline, mientras que uGUI, TextMesh Pro, Animator, Audio Source y Audio Mixer resuelven la interfaz, las animaciones y el sonido.

Meta XR proporciona los componentes que representan y reconocen las manos; OpenXR establece la comunicación con el entorno XR. Algunos prefabs del paquete conservan nombres internos como Controller and Hand, pero esa denominación no implica el uso de controles en la actividad. Los componentes activos de la experiencia corresponden a las manos virtuales y a la interacción directa.

Tabla 24.
Tecnologías principales del prototipo

Tecnología	Versión o configuración	Aplicación
Unity	6000.3.12f1	Motor, escenas, físicas, animación, audio, interfaz y compilación.
C#	Scripts MonoBehaviour	Lógica de los niveles y servicios generales.
Meta XR All-in-One SDK	201.0.0	Seguimiento de manos e interacción inmersiva.
OpenXR	Configurado para Meta XR	Comunicación estandarizada con el sistema XR.
Universal Render Pipeline	Perfil móvil del proyecto	Renderizado, materiales, iluminación y efectos.
uGUI y TextMesh Pro	Integrados en Unity	Botones, paneles, temporizadores y contadores.
Animator y Audio Mixer	Configurados por escena	Guías, reacciones, música y efectos.

Nota: elaboración propia a partir de la configuración del proyecto (2026).

La Tabla 24 resume la configuración tecnológica efectivamente documentada del prototipo y la función de cada componente en escenas, lógica, seguimiento de manos, renderizado, interfaz, animación y audio. Las versiones registradas permiten relacionar las capturas y el comportamiento con un entorno de desarrollo concreto.

Ilustración 30
Paquetes Meta XR instalados en el proyecto

Meta - Voice SDK - Immersive Voice Commands	85.0.0	🔗
Meta MR Utility Kit	201.0.0	🔗
Meta XR All-in-One SDK	201.0.0	🔗
Meta XR Audio SDK	85.0.0	🔗
Meta XR Core SDK	201.0.0	🔗
Meta XR Haptics SDK	201.0.0	🔗
Meta XR Interaction SDK	201.0.0	🔗
Meta XR Interaction SDK Essentials	201.0.0	🔗
Meta XR Platform SDK	201.0.0	🔗
Quick Outline	1.1	🔗

Nota: captura del proyecto Crokiando en Unity (2026).

La captura del Package Manager registra los paquetes Meta XR y sus versiones, evidencia de la configuración tecnológica utilizada para seguimiento de manos e interacción inmersiva.

4.3 Organización técnica del proyecto

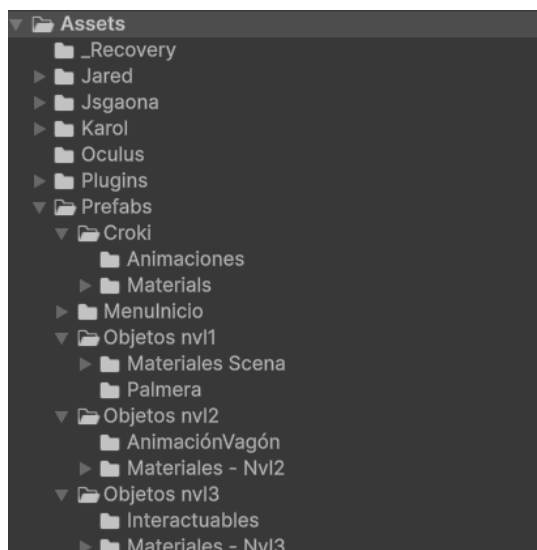
4.3.1 Estructura de carpetas y recursos

Dentro de Assets, la carpeta Scenes contiene el menú y las escenas educativas; Scripts agrupa el código de los niveles, el inicio y la medición de rendimiento; Prefabs reúne a Croki, los objetos del menú y los elementos reutilizables; y Settings, XR y Resources concentran configuraciones y recursos compartidos. Los materiales y animaciones se mantienen próximos a los prefabs que los utilizan.

Las carpetas Codigo N1, Codigo N2 y Codigo N3 reflejan identificadores creados durante el desarrollo. Para documentar el resultado sin romper referencias serializadas, se conserva esa organización interna y se explicita su correspondencia con el orden académico.

Ilustración 31

Organización de prefabs, escenas y scripts del proyecto (1)

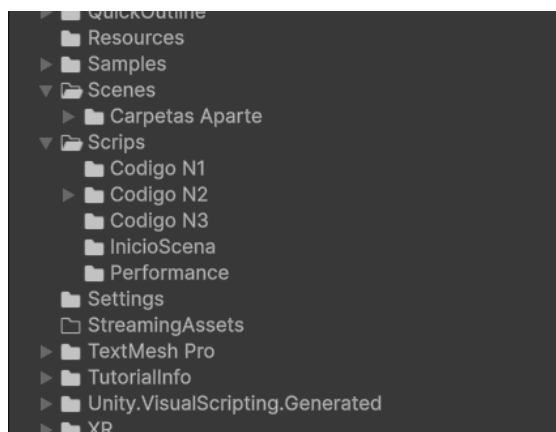


Nota: captura de la estructura del proyecto Crokiando (2026).

La captura muestra la jerarquía de Assets y la organización de los prefabs de Croki, objetos de los niveles y materiales, lo que evidencia la separación de recursos reutilizables.

Ilustración 32

Organización de prefabs, escenas y scripts del proyecto (2)



Nota: capturas de la estructura del proyecto Crokiando (2026).

La imagen complementa la estructura del proyecto al mostrar las carpetas de escenas, scripts y recursos, incluida la división del código por nivel y de los servicios compartidos.

4.3.2 Correspondencia de escenas

Tabla 25.

Correspondencia entre los niveles académicos y las escenas internas

Orden académico	Nombre del nivel	Escena interna	Gestor principal
Nivel 1	Planeta Dulzura	Nivel N3	N3ShapeRoundManager
Nivel 2	Parque de Figuras	Nivel N1	GameManagerShapes
Nivel 3	Tren de los Patrones	Nivel N2	ValidatePuzzle

Nota: elaboración propia a partir de las escenas y componentes del proyecto (2026).

La Tabla 25 resuelve la diferencia entre el orden académico de los niveles y los nombres internos de las escenas y gestores, referencia necesaria para interpretar las capturas de Unity.

La escena UAnimation contiene el menú, la configuración y la selección de actividades. La equivalencia de la Tabla 25 se utiliza en todo el capítulo para interpretar correctamente las capturas reales de Unity; no se renombraron escenas porque sus nombres forman parte de referencias y eventos ya configurados.

4.4 Arquitectura de clases y componentes

4.4.1 Modelo basado en componentes

La implementación sigue el modelo de composición de Unity: un `GameObject` obtiene su comportamiento mediante componentes. Una figura interactiva puede reunir un modelo visual, `Collider`, `Rigidbody`, un componente de agarre y un script de identidad. La detección de las manos, el movimiento físico y la regla educativa permanecen diferenciados, aunque colaboren sobre el mismo objeto.

La manipulación directa se configura con componentes como `HandGrabInteractable`, `Grabbable` y `GrabFreeTransformer`. El primero reconoce el agarre; `Grabbable` vincula la interacción con el cuerpo rígido; y `GrabFreeTransformer` permite trasladar y orientar el objeto mientras permanece sujeto. Cuando la figura entra en una zona configurada como trigger, el componente de dominio recibe el evento y valida su tipo.

Tabla 26.
Clases y componentes principales

Sistema	Clase o componente	Responsabilidad
Interacción manual	<code>HandGrabInteractable</code> , <code>Grabbable</code>	Detectar y mantener el agarre con la mano.
Transformación	<code>GrabFreeTransformer</code>	Trasladar y orientar objetos sujetos.
Planeta Dulzura	<code>N3ShapeRoundManager</code> , <code>N3AnswerButton</code>	Presentar figuras y comprobar la opción seleccionada.
Parque de Figuras	<code>GameManagerShapes</code> , <code>ShapeChestTrigger</code>	Generar figuras y validar su clasificación.
Tren de los Patrones	<code>ValidatePuzzle</code> , <code>Figure</code> , <code>ContainerFigure</code>	Crear patrones y comprobar la colocación.
Servicios	<code>SceneLoader</code> , <code>AudioManager</code>	Navegación, música, ambiente y efectos.
Accesibilidad	<code>XRCOLORAccessibilityController</code>	Aplicar la configuración cromática.

Nota: elaboración propia a partir de los componentes del proyecto (2026).

La Tabla 26 asigna responsabilidades a las clases y componentes principales: los elementos XR gestionan el agarre y la transformación, los gestores controlan las reglas de cada nivel y los servicios compartidos atienden navegación, audio y accesibilidad. Esta separación evidencia una arquitectura modular por función.

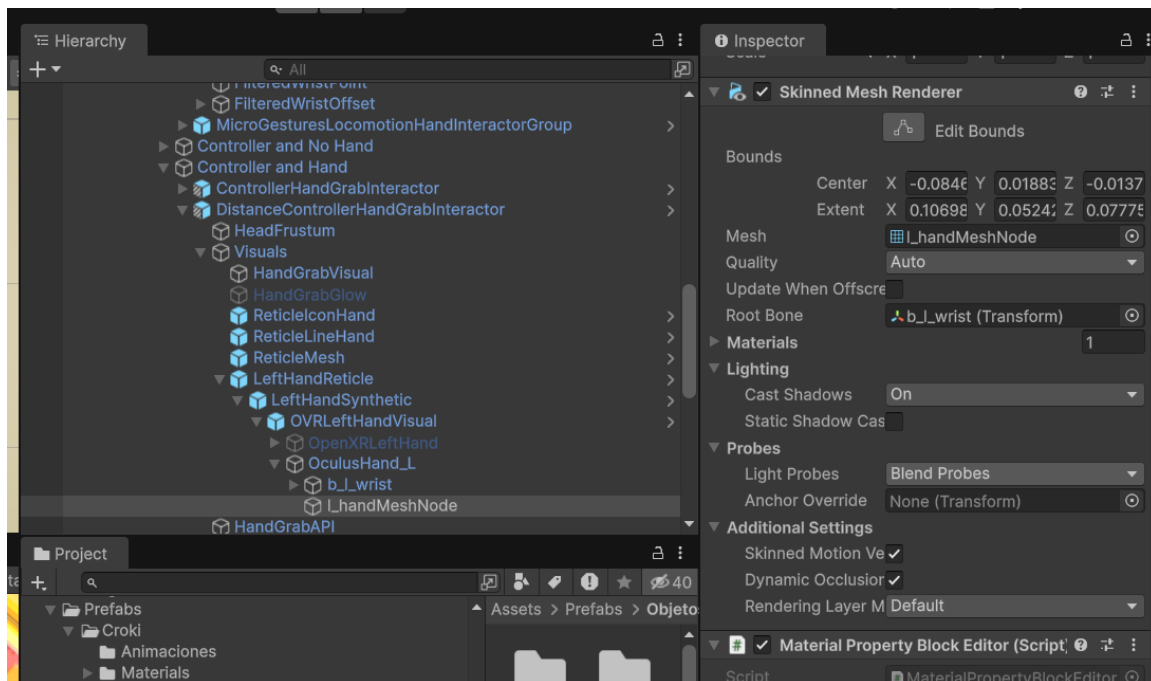
4.4.2 Comunicación y control de estado

Los componentes se comunican mediante referencias serializadas, métodos públicos, eventos y corrutinas. Los controladores de Croki notifican el comienzo y el final de la guía; los gestores esperan esa señal antes de habilitar la actividad. Las corrutinas coordinan animaciones, pausas, aparición de objetos y transiciones sin detener la ejecución de la escena.

Las variables de estado evitan que una respuesta, un inicio o una reposición se procese más de una vez. De este modo, la retroalimentación puede concluir antes de aceptar una nueva acción y cada objeto contabiliza una única validación.

Ilustración 33

Estructura de la mano virtual utilizada para el seguimiento



Nota: captura del proyecto Crokiando en Unity (2026).

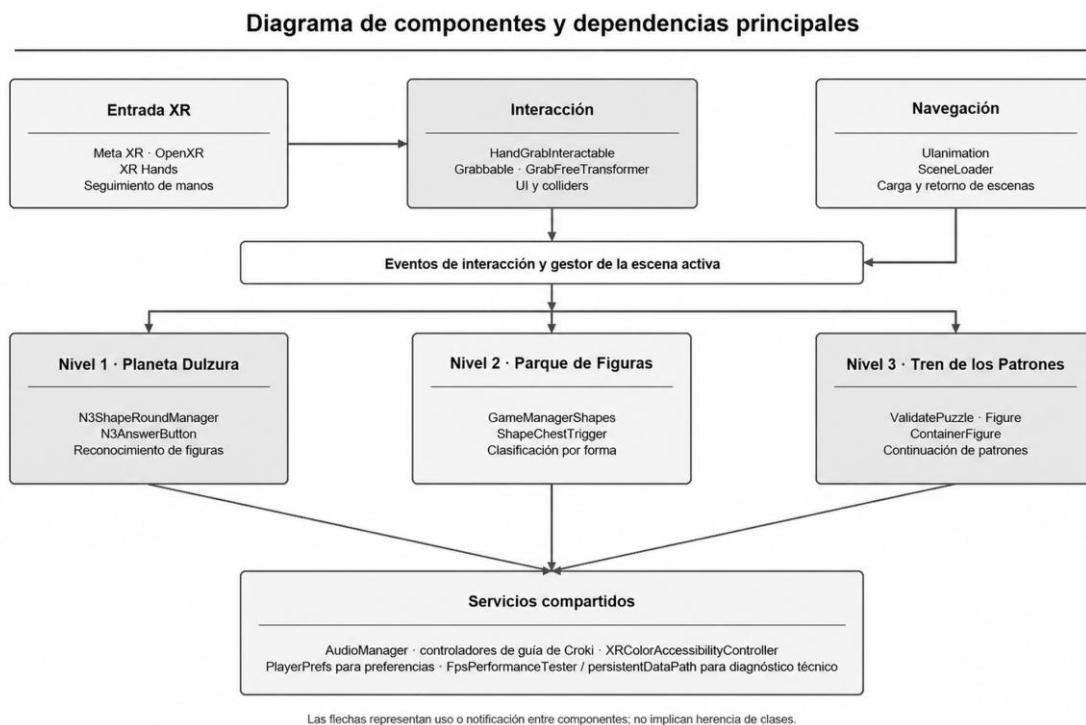
La captura presenta la configuración de la mano virtual en Unity y los componentes asociados a su representación y seguimiento dentro de la escena XR.

4.4.3 Diagrama de componentes y dependencias

El diagrama distingue la entrada XR, la interacción, la navegación, los gestores de cada nivel y los servicios compartidos. Las flechas representan uso o notificación entre componentes y no herencia de clases.

Ilustración 34

Componentes y dependencias principales del prototipo Crokiando



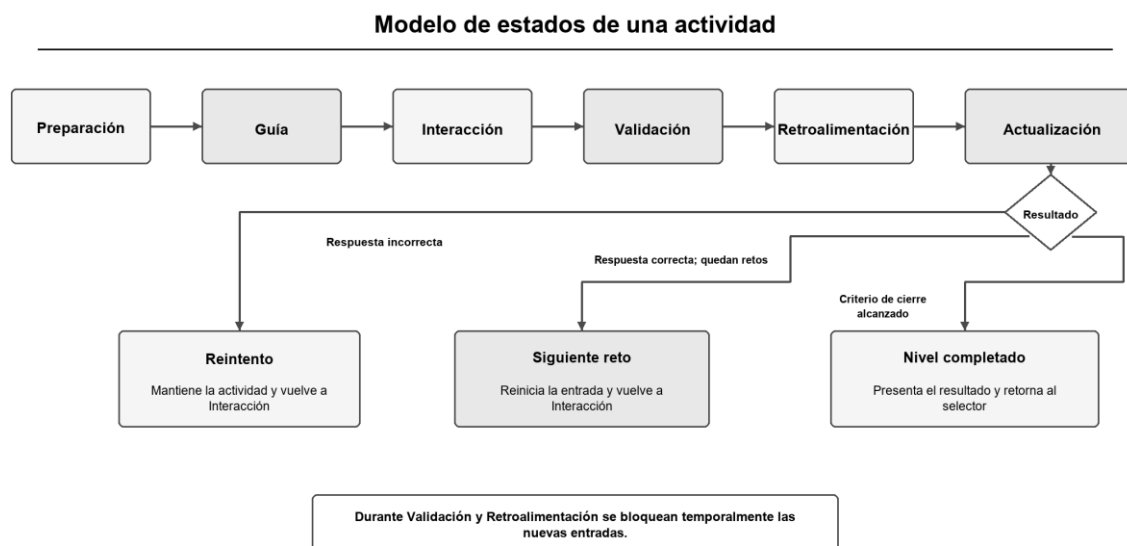
Nota: elaboración propia a partir de la estructura y clases documentadas del proyecto (2026).

El diagrama muestra las dependencias entre entrada XR, interacción, navegación, gestores de nivel y servicios compartidos; las flechas expresan uso o notificación entre componentes.

4.4.4 Modelo de estados de una actividad

La actividad avanza desde la preparación hasta la actualización del resultado. Después de la validación, el sistema bloquea nuevas entradas durante la retroalimentación y decide entre reintento, siguiente reto o cierre del nivel.

Ilustración 35
Modelo de estados de una actividad



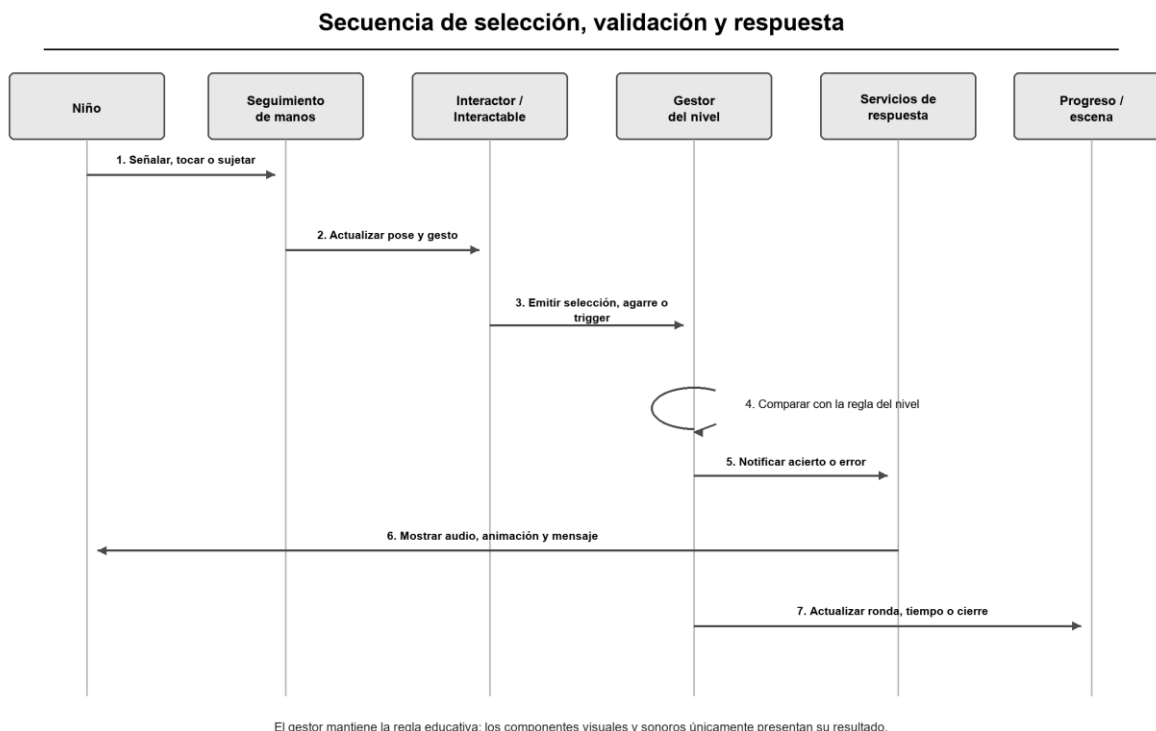
Nota: elaboración propia a partir del flujo implementado y documentado (2026).

El modelo representa la transición desde la preparación hasta la actualización del resultado y distingue las salidas de reintento, siguiente reto y finalización del nivel.

4.4.5 Secuencia de interacción y validación

La secuencia separa las responsabilidades del usuario, el seguimiento de manos, los objetos interactivos, el gestor del nivel, los servicios de respuesta y el control de progreso o escena.

Ilustración 36
Secuencia de selección, validación y respuesta



Nota: elaboración propia a partir de la comunicación entre los componentes documentados (2026).

El diagrama de secuencia detalla cómo la acción del niño pasa por el seguimiento de manos y el objeto interactivo hasta el gestor, que valida la regla y ordena la respuesta y la actualización del progreso.

4.5 Desarrollo de sistemas principales

4.5.1 Navegación e inicio de las actividades

El flujo comienza en UAnimation. El niño toma con la mano una pelota de selección y la coloca en el baúl correspondiente al nivel deseado. CofresScenas comprueba el activador y solicita el cambio de escena; si la pelota cae fuera del espacio previsto, RespawnPelotaMenu la devuelve a su posición. Al finalizar una actividad, SceneLoader permite regresar al menú.

Cada escena incorpora una guía animada de Croki. El controlador activa los objetos demostrativos, reproduce la explicación y mantiene bloqueada temporalmente la mecánica. Cuando la guía termina, el gestor prepara los elementos, acepta la interacción, comprueba la respuesta, comunica el resultado y continúa la actividad.

4.5.2 Audio, accesibilidad y medición

AudioManager organiza el mezclador MixSound en los grupos Master, Music y SFX. El servicio aplica transiciones entre pistas, puede conservarse al cambiar de escena y centraliza los parámetros de volumen. XRColorAccessibilityController aplica el material Daltonismo y ajusta el modo y la intensidad del filtro cromático. FpsPerformanceTester registra métricas técnicas para apoyar la revisión del comportamiento del prototipo.

Estos sistemas son transversales: no contienen la regla de una actividad, pero preparan el entorno en el que la mecánica se ejecuta. Esta separación permite que una modificación de audio o accesibilidad se aplique a varios niveles sin duplicar la lógica.

4.6 Desarrollo de mecánicas del videojuego

Las tres mecánicas siguen un ciclo común: presentación del reto, acción con las manos, comprobación, retroalimentación y actualización del avance. La lógica de juego determina el resultado, mientras Croki, la interfaz, los sonidos y las animaciones lo comunican.

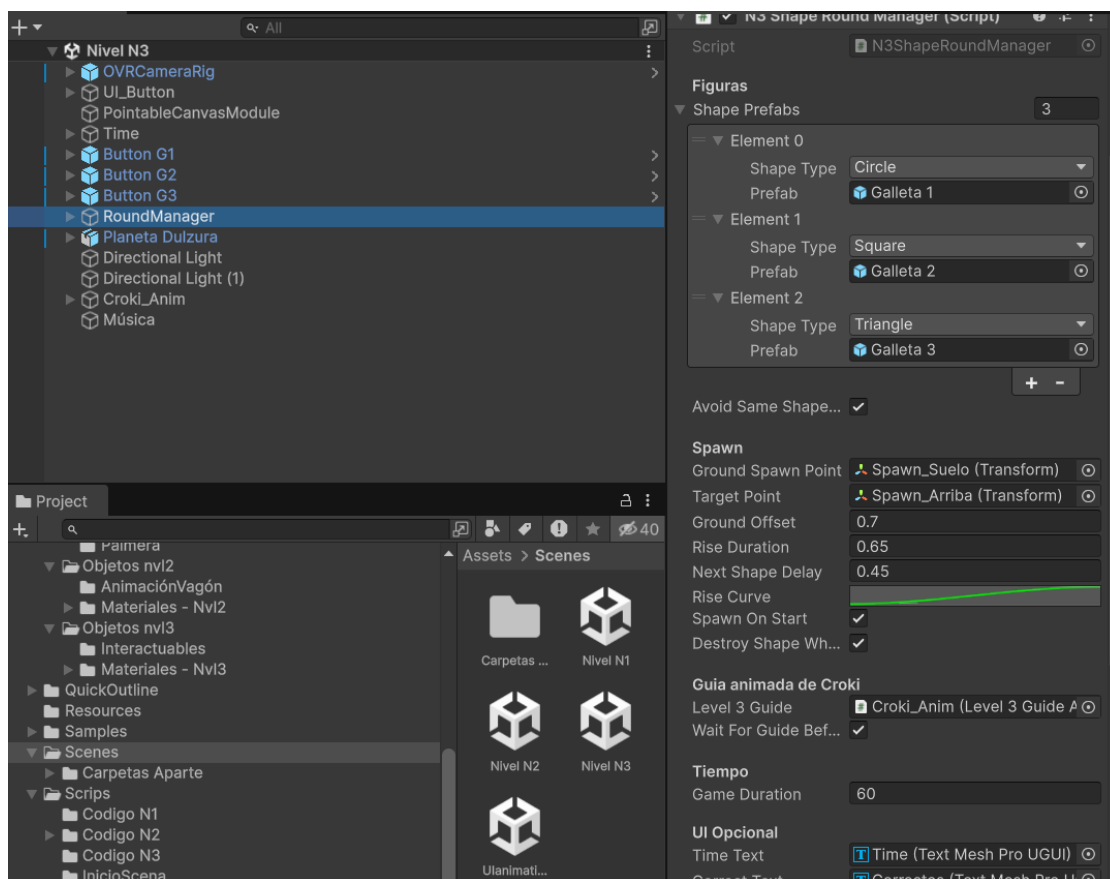
4.6.1 Nivel 1: Planeta Dulzura

Planeta Dulzura desarrolla el reconocimiento de figuras geométricas. El sistema presenta un círculo, un cuadrado o un triángulo y habilita opciones para que el niño identifique la forma correspondiente. La selección se realiza acercando el índice y activando el botón virtual mediante el gesto reconocido por el seguimiento de manos.

En la escena interna Nivel N3, N3ShapeRoundManager elige la figura y evita repetir inmediatamente la misma alternativa. La configuración muestra un desplazamiento vertical de 0,7 unidades durante 0,65 s y una espera de 0,45 s antes de la siguiente figura. La actividad dispone de 60 segundos. Cada opción utiliza N3AnswerButton, cuyo tipo se compara con la figura activa.

Por ejemplo, si aparece un círculo, el niño observa las alternativas y selecciona círculo con el índice. El gestor comprueba la coincidencia, actualiza el contador, activa la reacción de Croki y prepara el siguiente estímulo. Si la elección no coincide, presenta una orientación y permite continuar.

Ilustración 37
Configuración del gestor de Planeta Dulzura

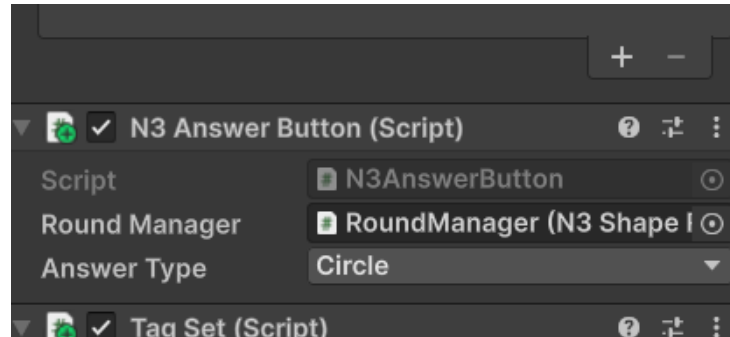


Nota: captura del proyecto Crokiando en Unity (2026).

La captura del Inspector muestra la configuración del gestor de Planeta Dulzura, incluidos prefabs, puntos de aparición, tiempo, interfaz y recursos de retroalimentación.

Ilustración 38

Configuración de una respuesta geométrica de Planeta Dulzura



Nota: captura del proyecto Crokiando en Unity (2026).

La imagen muestra cómo una respuesta geométrica se vincula con el gestor de ronda, su tipo de forma y las etiquetas que permiten comprobar la opción seleccionada.

4.6.2 Nivel 2: Parque de Figuras

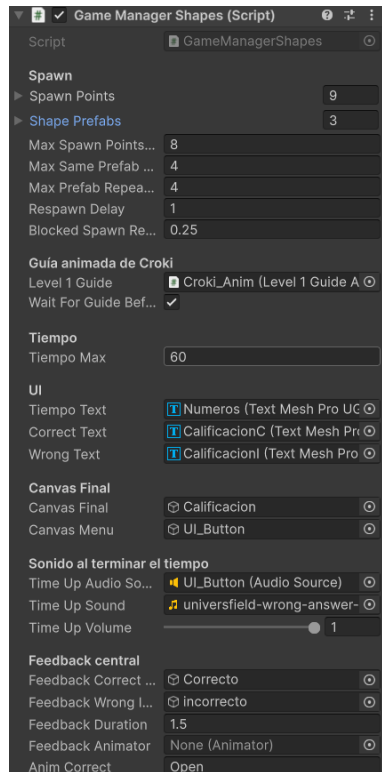
Parque de Figuras trabaja la clasificación por forma. El niño sujeta círculos, cuadrados y triángulos, los desplaza por el escenario y los deposita en el cofre correspondiente. El objeto acompaña el movimiento de la mano hasta que el usuario abre los dedos o libera el gesto de sujeción.

En la escena interna Nivel N1, GameManagerShapes administra tres prefabs y nueve puntos de aparición, con un máximo de ocho posiciones activas y una duración de 60 segundos. Cada cofre incorpora ShapeChestTrigger, donde se define el tipo admitido. Al entrar en la zona de validación, el componente compara la figura con el destino y activa el texto, la animación y el sonido correspondientes.

Como ejemplo, el niño toma un cuadrado, lo conduce hasta el cofre de esa forma y abre la mano para soltarlo. Si existe coincidencia, el sistema registra el acierto y presenta una señal positiva; si no coincide, la figura continúa disponible para otro intento.

Ilustración 39

Gestión y validación de cofres en Parque de Figuras, parte 1

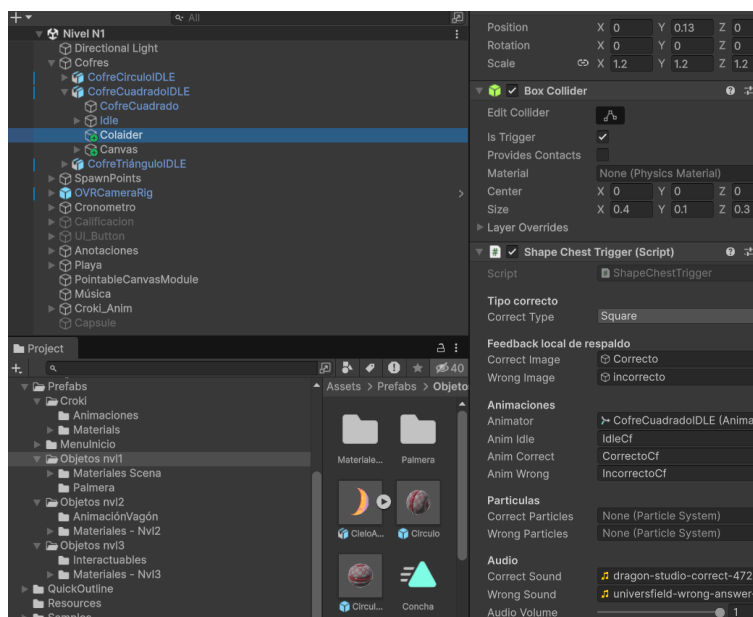


Nota: captura del proyecto Crokiando en Unity (2026).

La captura presenta el gestor de Parque de Figuras y sus parámetros de generación, tiempo, interfaz, audio y retroalimentación, responsables de coordinar la mecánica.

Ilustración 40

Gestión y validación de cofres en Parque de Figuras, parte 2



Nota: capturas del proyecto Crokiando en Unity (2026).

La imagen muestra la configuración de un cofre como zona de validación, con collider, tipo de forma esperado y respuestas diferenciadas para acierto y error.

4.6.3 Nivel 3: Tren de los Patrones

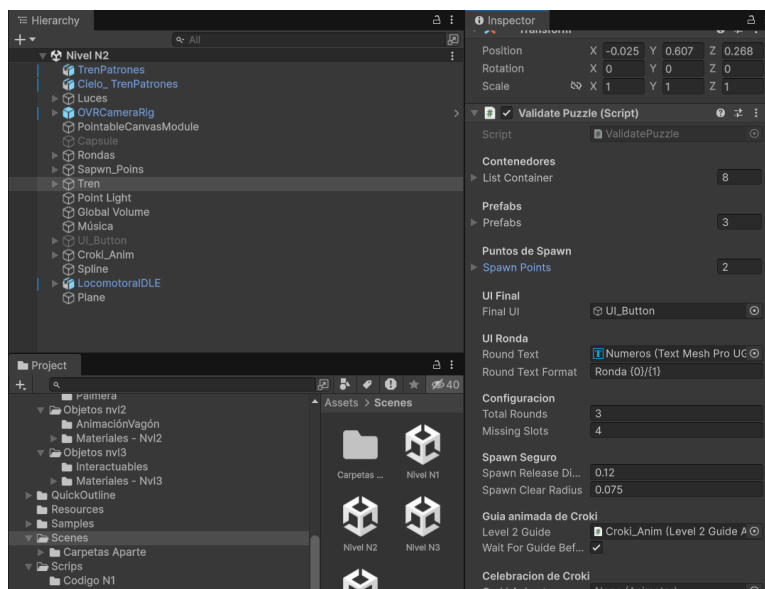
Tren de los Patrones propone completar secuencias alternadas. La escena utiliza ocho posiciones y tres rondas; en cada una se combinan dos tipos de figuras y se dejan espacios incompletos para que el niño reconozca la regularidad.

En la escena interna Nivel N2, ValidatePuzzle administra los contenedores, los tres prefabs, dos puntos de aparición, el número de rondas y los espacios faltantes. El niño sujeta una pieza, la traslada al vagón elegido y la suelta. Si la figura coincide con la posición esperada se integra al patrón y se actualiza el avance; de lo contrario, el espacio queda disponible para otro intento.

Por ejemplo, ante la secuencia círculo–cuadrado–círculo–?, la respuesta esperada es un cuadrado. La colocación correcta completa ese tramo, activa la celebración de Croki y permite continuar hasta resolver las tres rondas.

Ilustración 41

Configuración del patrón en Tren de los Patrones, parte 1

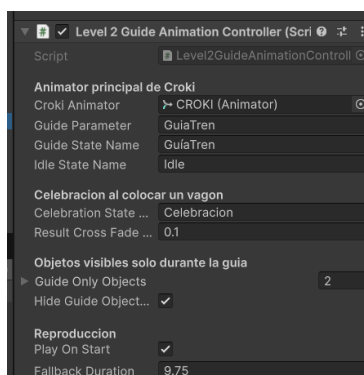


Nota: captura del proyecto Crokiando en Unity (2026).

La captura presenta ValidatePuzzle y los parámetros que controlan contenedores, prefabs, puntos de aparición, rondas y espacios faltantes del Tren de los Patrones.

Ilustración 42

Configuración del patrón en Tren de los Patrones, parte 2



Nota: capturas del proyecto Crokiando en Unity (2026).

La imagen muestra el controlador de la guía animada de Croki, con los estados de explicación y celebración que bloquean o habilitan la interacción del nivel.

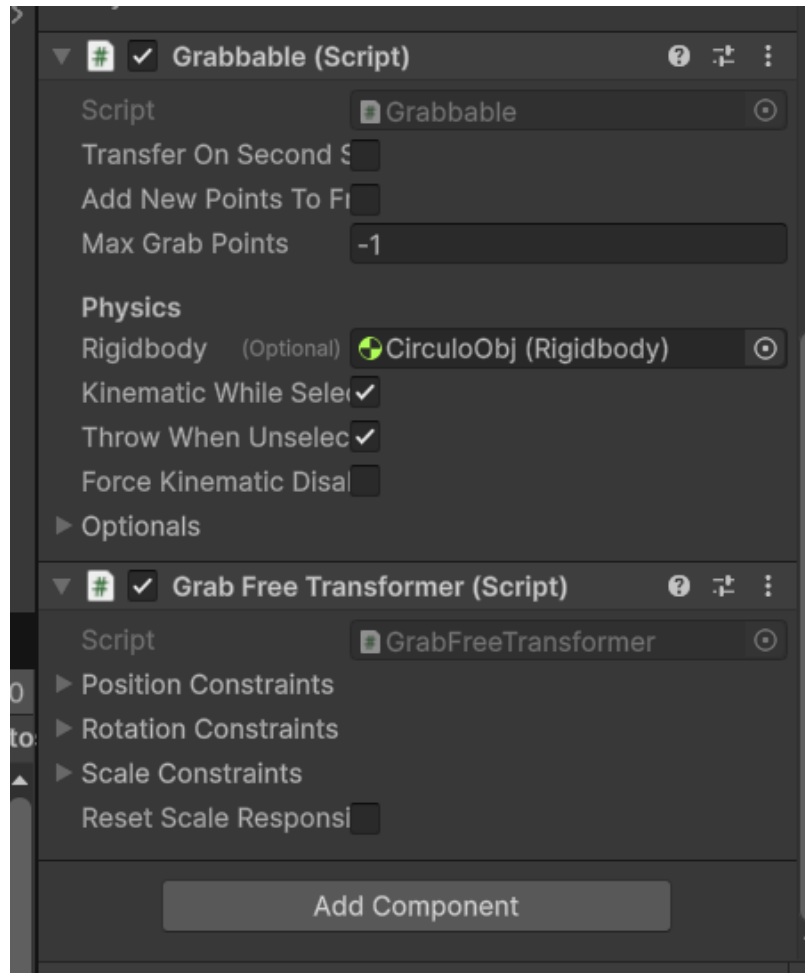
4.7 Integración de sistemas

La integración conecta la entrada mediante seguimiento de manos, los componentes interactivos, la lógica de las mecánicas, los servicios compartidos y la presentación audiovisual. El visor obtiene la posición de las manos; los interactores reconocen contacto, pulsación o agarre; el objeto informa la acción al gestor; y este comprueba la regla educativa antes de actualizar la interfaz, el audio y el Animator de Croki.

1. El visor determina la posición y orientación de la mano.
2. El componente de interacción reconoce contacto, sujeción, liberación o selección.
3. El objeto o botón comunica la acción al gestor del nivel.
4. El gestor compara la acción con la regla de la actividad.
5. La interfaz, el audio y Croki presentan el resultado.
6. El nivel habilita el siguiente intento o muestra la finalización.

La guía animada participa en la misma cadena. Mientras Croki demuestra la tarea, el gestor mantiene bloqueado el gameplay; al recibir la señal de finalización, oculta los objetos auxiliares y habilita la mecánica una sola vez. Audio y accesibilidad se aplican al cargar la escena, mientras los contadores y rondas permanecen bajo la autoridad del gestor específico.

Ilustración 43
Componentes de agarre y transformación manual



Nota: captura del proyecto Crokiando en Unity (2026).

La captura identifica los componentes Grabbable, cuerpo rígido y transformador libre que permiten sujetar, mover y soltar una figura con la mano.

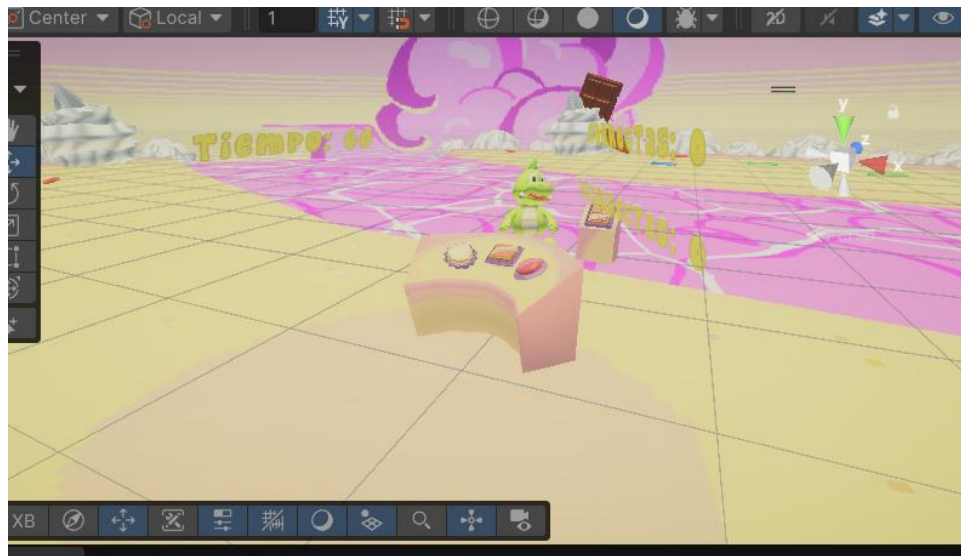
4.8 Integración entre diseño, arte y programación

Los recursos visuales se incorporan como prefabs funcionales. Un objeto combina modelo, material, collider, cuerpo rígido, componentes de Meta XR y script de dominio. Así, un cambio de modelo o material puede realizarse sin reescribir toda la mecánica, mientras la identidad y la validación permanecen en sus componentes.

Croki aporta continuidad narrativa entre escenarios. Sus estados de espera, guía, celebración y orientación se activan desde los gestores en momentos definidos. Parque de Figuras utiliza una ambientación de parque con cofres; Planeta Dulzura presenta un escenario de dulces y figuras; y Tren de los Patrones integra locomotora, vagones y una composición urbana nocturna. Los tres conservan una interfaz coherente y retroalimentación multimodal.

Ilustración 44

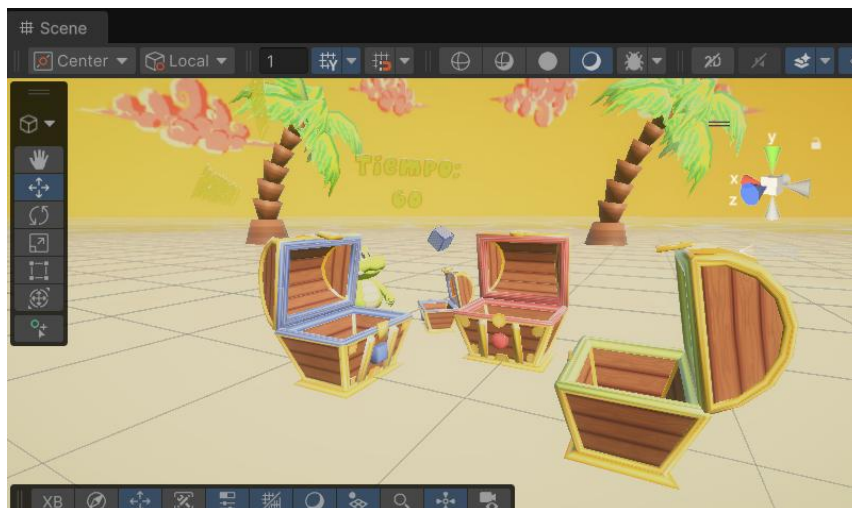
Escenarios implementados para Planeta Dulzura



Nota: captura de la escena del prototipo Crokiando en Unity (2026).

La imagen documenta en Unity el escenario de Planeta Dulzura, su ambientación, objetos interactivos y organización en la jerarquía de la escena.

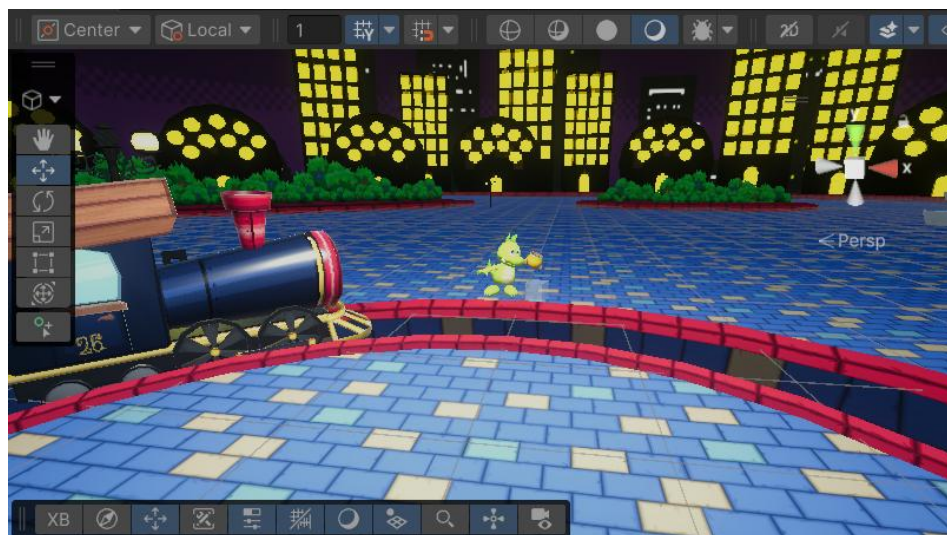
Ilustración 45
Escenarios implementados para Parque de Figuras



Nota: captura de la escena del prototipo Crokiando en Unity (2026).

La captura documenta el escenario de Parque de Figuras, incluidos los cofres y elementos de clasificación visibles tanto en la escena como durante la ejecución.

Ilustración 46
Escenarios implementados para Tren de los Patrones

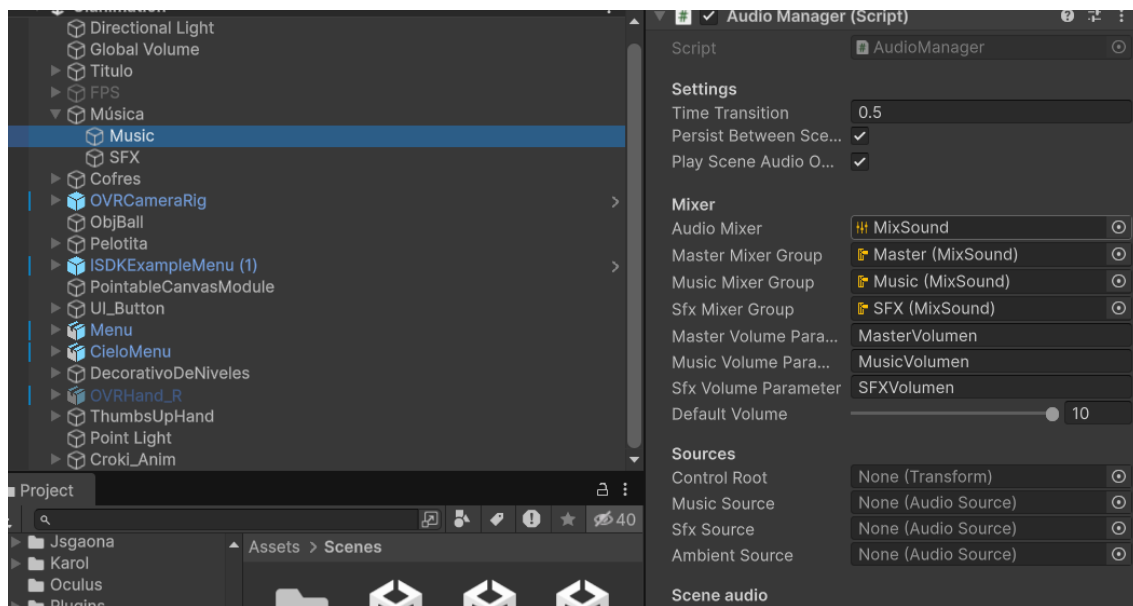


Nota: capturas de las escenas del prototipo Crokiando en Unity (2026).

La imagen documenta el escenario de Tren de los Patrones y la disposición de locomotora, vagones, piezas y personaje guía durante la actividad.

El sonido se distribuye en los grupos Master, Music y SFX del mezclador MixSound. La música de fondo se dirige a Music y las respuestas de la interfaz o de las mecánicas a SFX. Esta separación permite regular cada categoría desde un punto central.

Ilustración 47
Configuración del sistema centralizado de audio

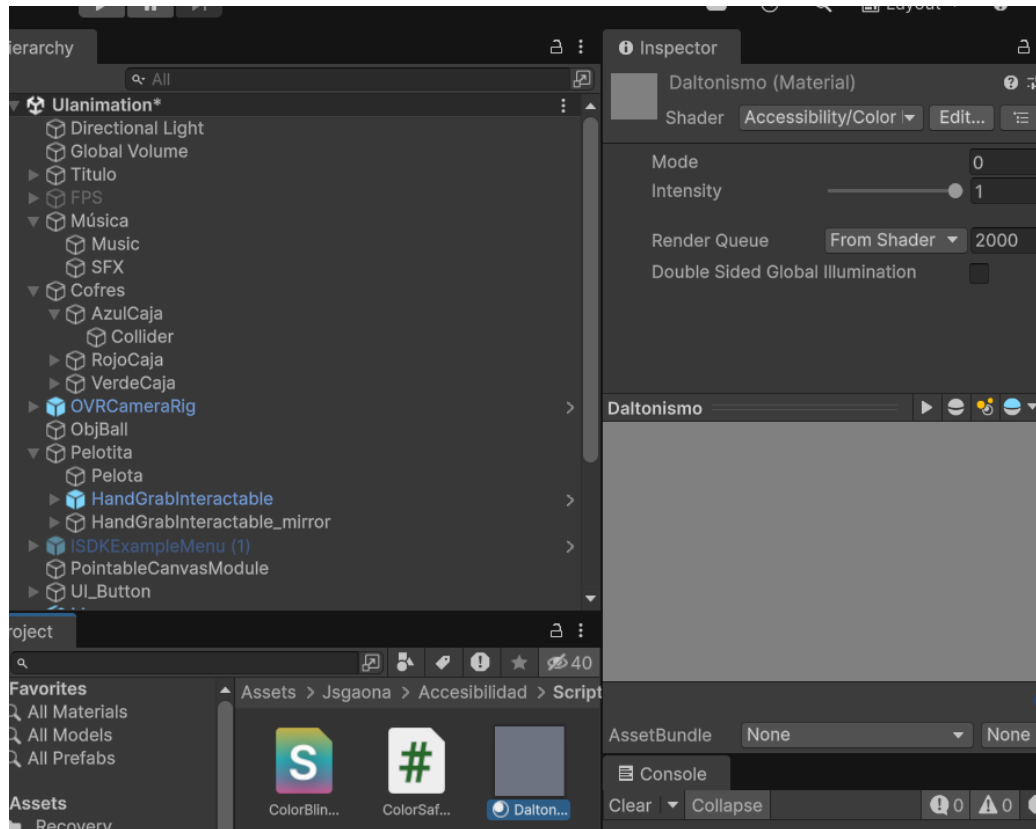


Nota: captura del proyecto Crokiando en Unity (2026).

La captura muestra el AudioManager y la asignación de los grupos Master, Music y SFX, que centralizan la mezcla y los niveles de volumen.

La accesibilidad cromática se integra como un efecto global. El material Daltonismo utiliza el shader Accessibility/Color y expone los parámetros Mode e Intensity. XRColorAccessibilityController aplica el ajuste sin modificar por separado cada textura o material de las escenas.

Ilustración 48
Material y recursos del sistema de accesibilidad cromática



Nota: captura del proyecto Crokiando en Unity (2026).

La imagen muestra el material Daltonismo, el shader de accesibilidad cromática y los recursos que permiten aplicar el modo seleccionado de forma global.

4.8.1 Aporte individual y trazabilidad de recursos

La contribución individual evaluada en esta tesis comprende la configuración XR, la programación en C#, el control de estados, la validación de respuestas, la retroalimentación, el progreso, la navegación, las pruebas funcionales y la documentación técnica. Los modelos, texturas, animaciones y audios se reconocen como aportes del equipo o de terceros cuando corresponda; no se atribuyen de forma exclusiva al autor sin un registro verificable.

- Motor documentado: Unity 6000.3.12f1. Estado: identificado en la tesis; comprobante del proyecto pendiente de adjuntar.
- Paquetes documentados: Meta XR All-in-One SDK, MR Utility Kit, Core, Haptics, Interaction y Platform SDK 201.0.0. Estado: versión declarada; origen y licencia deben cotejarse con el manifiesto del proyecto.

- Paquete documentado: Meta XR Audio & Voice SDK 85.0.0. Estado: versión declarada; manifiesto y licencia pendientes de adjuntar.
- Recurso documentado: Quick Outline 1.1. Estado: uso declarado; origen, autor, licencia y modificaciones pendientes de respaldar.
- Recursos audiovisuales y tridimensionales. Estado: inventario incompleto; debe identificarse por archivo el origen, autor, licencia, modificación y responsable de integración.

Antes del depósito final debe anexarse el repositorio o historial de cambios, el manifiesto de paquetes y una matriz completa de scripts, modelos, texturas, animaciones y audios con origen, autor, licencia, modificación y aporte individual, validada con el equipo y el tutor. Esta exigencia constituye una brecha de cierre y no se presenta como ya cumplida.

4.9 Gestión de datos y persistencia

Crokiando conserva únicamente preferencias pequeñas del dispositivo. AudioManager utiliza PlayerPrefs para recuperar y guardar los valores de volumen maestro, música y efectos. XRColorAccessibilityController utiliza la clave ColorBlindMode para mantener el modo cromático seleccionado y aplicarlo nuevamente al iniciar.

Los datos propios de las mecánicas tienen un alcance temporal. El tiempo restante, los aciertos, los errores, la ronda actual y las figuras colocadas se mantienen durante la escena y se reinician al comenzar una nueva sesión. El prototipo no almacena perfiles, nombres, desbloques ni historiales individuales de los niños. Los resultados del testeo se analizaron de forma agregada para la investigación y no forman parte del almacenamiento interno del videojuego.

FpsPerformanceTester puede escribir métricas técnicas en Application.persistentDataPath/PerformanceTests. Esos archivos se destinan al diagnóstico del rendimiento y no representan progreso académico ni información personal del participante.

Tabla 27.
Gestión y permanencia de los datos

Tipo de dato	Mecanismo	Permanencia	Uso
Volumen maestro, música y efectos	PlayerPrefs y AudioManager	Entre escenas y ejecuciones	Preferencia de audio.
Modo cromático	PlayerPrefs: ColorBlindMode	Entre escenas y ejecuciones	Accesibilidad visual.
Aciertos, errores, tiempo y ronda	Variables del gestor	Durante la escena	Retroalimentación y avance local.
Figuras colocadas o pendientes	Estado del nivel	Durante la ronda	Validación de la mecánica.
Métricas técnicas	persistentDataPath	Archivo local de diagnóstico	Evaluación de rendimiento.
Datos personales de participantes	No se almacenan	No aplica	Protección de la identidad.

Nota: elaboración propia a partir del código y de la configuración del proyecto (2026).

La Tabla 27 distingue los datos persistentes de las variables temporales de cada escena. Solo las preferencias de audio y accesibilidad se conservan entre ejecuciones; el progreso de las mecánicas se reinicia y no se almacenan datos personales de los participantes.

4.10 Configuración de escenas y flujo del videojuego

La navegación adopta un flujo de centro y radios: UAnimation funciona como punto central y permite acceder de manera independiente a cualquiera de las tres actividades. Cada escena incorpora OVRCameraRig, módulos de interacción, Croki, el gestor de la mecánica, interfaz y audio.

Tabla 28.
Escenas y función dentro del flujo

Escena interna	Nombre utilizado en la tesis	Función
UAnimation	Menú principal	Presentación, configuración, selección mediante pelota y baúl, y regreso.
Nivel N3	Nivel 1: Planeta Dulzura	Reconocimiento de círculo, cuadrado y triángulo.
Nivel N1	Nivel 2: Parque de Figuras	Clasificación de objetos según su forma.
Nivel N2	Nivel 3: Tren de los Patrones	Continuación de secuencias alternadas.

Nota: elaboración propia a partir de las escenas del proyecto (2026).

La Tabla 28 traduce los nombres internos de las escenas al orden académico utilizado en la tesis y precisa la función de cada una. Esta equivalencia evita confundir UAnimation, Nivel N1, Nivel N2 y Nivel N3 al interpretar la navegación, el código y las capturas de Unity.

1. Unity carga la escena y resuelve sus referencias configuradas en el Inspector.
2. Los servicios compartidos aplican el volumen y la opción cromática.
3. Croki presenta la guía y la mecánica permanece temporalmente bloqueada.
4. Al terminar la explicación se habilita la interacción mediante las manos.
5. El gestor procesa respuestas, tiempo o rondas y actualiza el avance.
6. Al cumplirse la condición de cierre aparece la interfaz final.
7. El usuario selecciona la salida con la mano y regresa al menú.

Por ejemplo, al elegir Parque de Figuras, el menú solicita la escena interna Nivel N1. Después de aplicar las preferencias y reproducir la guía de Croki, el gestor habilita las figuras. Al terminar, el botón de salida devuelve a UAnimation sin trasladar los contadores temporales a la siguiente actividad.

4.11 Pruebas técnicas y validación

4.11.1 Verificación funcional

La verificación funcional se organiza mediante ocho casos que explicitan precondition, pasos, resultado esperado, resultado obtenido, estado, versión y evidencia. Como no se suministraron bitácoras originales de ejecución ni capturas enlazadas a cada corrida, la matriz distingue la comprobación respaldada por la documentación técnica de la réplica reproducible que aún debe ejecutarse y conservarse.

Tabla 29.
Matriz de verificación funcional

Código	Precondición y pasos	Resultado esperado y obtenido	Estado, versión y evidencia
PF-01	Escena instalada. Abrir el nivel y esperar la guía.	Esperado: Cargar objetos, interfaz, ronda y guía de Croki. Obtenido: La arquitectura documentada inicializa esos componentes.	Estado: Cumple según evidencia documental; réplica pendiente Versión y evidencia: Crokiando.apk, 27-07-2026; jerarquías y gestores de escena.
PF-02	Seguimiento habilitado. Iniciar y mover ambas manos.	Esperado: Representar el movimiento sin controles físicos. Obtenido: La configuración muestra OVRCameraRig y manos OpenXR/Meta XR.	Estado: Cumple según evidencia documental; réplica pendiente Versión y evidencia: Crokiando.apk, 27-07-2026; configuración XR.
PF-03	Objeto agarrable activo. Tomarlo, moverlo y liberarlo.	Esperado: El objeto acompaña la mano y queda libre al soltarlo. Obtenido: Los componentes Grabbable y GrabFreeTransformer están documentados.	Estado: Cumple según evidencia documental; réplica pendiente Versión y evidencia: Crokiando.apk, 27-07-2026; componentes de agarre.

Código	Precondición y pasos	Resultado esperado y obtenido	Estado, versión y evidencia
PF-04	Planeta Dulzura iniciado. Elegir opciones correcta e incorrecta.	Esperado: Comparar la forma, informar el resultado y avanzar. Obtenido: N3ShapeRoundManager y N3AnswerButton implementan el flujo descrito.	Estado: Cumple según evidencia documental; réplica pendiente Versión y evidencia: Crokiando.apk, 27-07-2026; scripts y capturas del nivel.
PF-05	Parque de Figuras iniciado. Depositar figuras en cofres correcto e incorrecto.	Esperado: Validar el cofre, responder y actualizar el progreso. Obtenido: GameManagerShapes y ShapeChestTrigger implementan la regla descrita.	Estado: Cumple según evidencia documental; réplica pendiente Versión y evidencia: Crokiando.apk, 27-07-2026; scripts y capturas del nivel.
PF-06	Tren de los Patrones iniciado. Colocar piezas correcta e incorrecta.	Esperado: Validar cada posición y completar la secuencia. Obtenido: ValidatePuzzle y los contenedores implementan la comprobación descrita.	Estado: Cumple según evidencia documental; réplica pendiente Versión y evidencia: Crokiando.apk, 27-07-2026; scripts y capturas del nivel.
PF-07	Un nivel activo. Provocar acierto y error.	Esperado: Diferenciar ambos resultados con señales coherentes. Obtenido: Textos, Audio Source y Animator documentan los canales de respuesta.	Estado: Cumple según evidencia documental; réplica pendiente Versión y evidencia: Crokiando.apk, 27-07-2026; interfaz, audio y animación.
PF-08	Completar un nivel. Revisar resultado y volver al menú.	Esperado: Presentar el cierre y permitir la navegación. Obtenido: La interfaz final y SceneLoader documentan el flujo de salida.	Estado: Cumple según evidencia documental; réplica pendiente Versión y evidencia: Crokiando.apk, 27-07-2026; interfaz final y SceneLoader.

Nota: elaboración propia a partir de la documentación técnica disponible (2026); la réplica de ejecución y sus evidencias originales permanecen pendientes.

Esta matriz cubre inicio, seguimiento de manos, agarre, reglas de los tres niveles, retroalimentación y cierre. El estado consignado no reemplaza un registro reproducible: antes del depósito deben asociarse a cada caso las capturas o bitácoras originales, la persona responsable, la fecha de ejecución y la versión exacta.

4.11.2 Testeo con participantes

El material suministrado reporta una muestra de 20 participantes del público objetivo, una explicación inicial, interacción con las actividades y respuesta a diez preguntas agregadas. No incluye la distribución exacta por edad, institución y reclutamiento, experiencia previa, orden y duración de los niveles, pausas, ayudas, abandonos o incidencias; tampoco incorpora el instrumento original, la escala visual utilizada ni la base anonimizada. Por ello, los resultados se presentan de forma descriptiva y no permiten análisis por subgrupos ni generalización.

Tabla 30.
Resultados de usabilidad e interacción

Indicador	Respuestas favorables	Porcentaje
Comprendió la actividad desde el inicio	15 de 20	75 %
Consideró fácil agarrar y mover los objetos	15 de 20	75 %
Percibió una respuesta rápida de los objetos	17 de 20	85 %
Consideró fácil colocar los objetos correctamente	16 de 20	80 %
Reconoció cuándo acertó y cuándo se equivocó	20 de 20	100 %
Consideró útiles los sonidos, luces y animaciones	15 de 20	75 %
Indicó que la actividad funcionó sin interrupciones o pérdida de objetos	17 de 20	85 %

Nota: instrumento aplicado durante el testeo del prototipo (n = 20), 2026.

La Tabla 30 presenta siete indicadores agregados de usabilidad e interacción para 20 participantes. Las respuestas favorables oscilan entre 75 % y 100 %; el reconocimiento de aciertos y errores alcanza el valor más alto, mientras que la comprensión inicial, el agarre y la utilidad de los recursos audiovisuales registran 75 %.

En los datos agregados suministrados, 20/20 participantes reconocieron cuándo acertaron y cuándo se equivocaron. La rapidez de respuesta y el funcionamiento sin interrupciones o pérdida de objetos registraron 17/20; la facilidad para colocar objetos, 16/20; y la comprensión inicial, el agarre y la utilidad de los recursos audiovisuales, 15/20. Estos valores describen el desempeño percibido en la muestra reportada y orientan ajustes del prototipo.

En la aceptación inicial, Planeta Dulzura registró 17/20 respuestas favorables; Parque de Figuras, 19/20; y Tren de los Patrones, 19/20. Además, 12/20 participantes eligieron Planeta Dulzura como favorito, 17/20 destacaron agarrar y mover objetos y 17/20 indicaron que volverían a jugar. La falta del instrumento y de la base anonimizada impide reconstruir otras distribuciones o asociaciones.

Los resultados constituyen evidencia descriptiva preliminar de funcionamiento percibido, usabilidad y aceptación inicial dentro de la muestra reportada. No miden cambios de aprendizaje ni sustituyen la caracterización de participantes, el instrumento original, la base anonimizada o la trazabilidad ética.

4.12 Optimización y rendimiento

La optimización se abordó mediante escenas independientes, reutilización de prefabs, generación controlada y parámetros ajustables desde el Inspector. Cada actividad mantiene

activos sus propios recursos, mientras los servicios compartidos evitan duplicar configuraciones de audio y accesibilidad.

Parque de Figuras limita la cantidad y repetición de objetos y utiliza intervalos para reponerlos. Planeta Dulzura controla una figura por turno y temporiza su aparición. Tren de los Patrones restringe el número de rondas, contenedores y espacios pendientes. El proyecto utiliza una configuración de renderizado móvil, compilación IL2CPP para ARM64 y opciones de Meta XR orientadas al visor.

*Tabla 31.
Estrategias de optimización aplicadas*

Estrategia	Aplicación	Aporte
Separación por escenas	Un entorno independiente por actividad.	Carga solamente los recursos del nivel activo.
Reutilización de prefabs	Figuras, cofres, vagones y elementos interactivos.	Facilita ajustes comunes y reduce duplicación.
Generación controlada	Límites, intervalos y puntos de aparición.	Evita acumulación innecesaria de objetos.
Validación por eventos	Triggers, componentes de agarre y gestores.	Evita comprobaciones generales innecesarias.
Audio centralizado	Grupos Master, Music y SFX.	Concentra el control y la mezcla del sonido.
Parámetros serializados	Duraciones, cantidades y referencias.	Permite ajustar el comportamiento desde el Inspector.
Medición técnica	FpsPerformanceTester y archivos de diagnóstico.	Apoya pruebas repetibles en el dispositivo.

Nota: elaboración propia a partir de la implementación (2026).

La Tabla 31 relaciona las estrategias de separación por escenas, reutilización, generación controlada, validación por eventos, audio centralizado, parámetros serializados y medición técnica con su aporte al rendimiento y al mantenimiento. Estas medidas describen decisiones de optimización, pero no sustituyen mediciones repetidas en el visor.

Las lecturas obtenidas dentro del Editor se consideran diagnósticas porque el entorno de desarrollo añade una carga que no representa exactamente la ejecución del APK en el visor. Por ello, el documento registra las estrategias y el procedimiento de medición sin afirmar una tasa fija de cuadros por segundo que no haya sido comprobada mediante una prueba repetida en el dispositivo.

4.13 Despliegue y compilación

La aplicación se preparó para Android y Meta Quest. El proceso incluye seleccionar la plataforma, habilitar OpenXR y el grupo de características de Meta XR, registrar el menú y las tres escenas en el perfil de compilación, configurar IL2CPP y ARM64, generar el paquete e instalarlo en el visor. Antes de compilar se comprueba que el seguimiento de manos se encuentre habilitado y que todas las escenas necesarias estén incluidas.

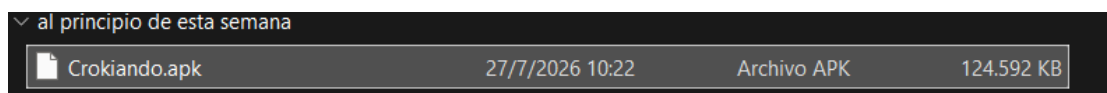
Tabla 32
Datos del artefacto de compilación

Elemento	Descripción
Motor	Unity 6
Plataforma de salida	Android para Meta Quest
Sistema XR	OpenXR y Meta XR
Modalidad de entrada	Seguimiento directo de manos
Archivo generado	Crokiando.apk
Tamaño mostrado	124 592 KB
Fecha de la evidencia	27 de julio de 2026

Nota: captura del artefacto generado (2026).

La Tabla 32 registra la configuración de compilación y los metadatos observables del APK; demuestra la existencia del artefacto, pero no acredita por sí sola su instalación o publicación.

Ilustración 49
Paquete APK generado para la instalación



Nota: evidencia de compilación del proyecto Crokiando (2026).

La captura evidencia la generación de Crokiando.apk y registra el nombre, la fecha y el tamaño del artefacto disponible para instalación.

La evidencia confirma la generación del archivo APK. La instalación, firma o publicación en una tienda constituyen etapas posteriores y no se presentan como acciones demostradas por esta captura.

4.14 Escalabilidad y mantenimiento

En el contexto de Crokiando, la escalabilidad se refiere a incorporar figuras, rondas, retroalimentaciones y niveles adicionales. La separación por escenas, prefabs y componentes permite extender el contenido sin sustituir el sistema completo de interacción. Los gestores concentran las reglas y exponen cantidades, duraciones, sonidos y referencias en el Inspector.

El mantenimiento se favorece al conservar la organización por nivel, documentar la equivalencia entre nombres internos y académicos, fijar las versiones de los paquetes XR, respaldar cada versión funcional y repetir la matriz de pruebas después de una modificación. Los servicios transversales de manos, audio y accesibilidad pueden reutilizarse en nuevas actividades.

Tabla 33.
Criterios de escalabilidad y mantenimiento

Criterio	Aplicación en el proyecto
Modularidad	Una escena y un gestor principal por actividad.
Reutilización	Prefabs y servicios compartidos para interacción, audio y accesibilidad.
Configuración	Parámetros serializados ajustables desde el Inspector.
Trazabilidad	Correspondencia documentada entre escenas internas y orden académico.
Control de cambios	Respaldo de versiones y repetición de pruebas funcionales.
Ampliación futura	Nuevos catálogos de contenido o interfaces pueden incorporarse de manera incremental.

Nota: elaboración propia (2026).

La Tabla 33 sintetiza los criterios que permiten mantener y ampliar el prototipo: modularidad, reutilización, configuración, trazabilidad, control de cambios y crecimiento incremental. Su aplicación reduce la necesidad de modificar componentes no relacionados al incorporar nuevo contenido.

4.15 Procedimiento para ampliar el sistema

La incorporación de una nueva función o contenido puede seguir el procedimiento siguiente:

1. Definir el objetivo educativo, la acción solicitada y la respuesta esperada.
2. Crear o duplicar la escena base y configurar cámara, seguimiento de manos, interfaz y guía.
3. Preparar prefabs con modelo, material, Collider, Rigidbody y componente de interacción.
4. Implementar o ampliar el gestor encargado de la validación, las rondas y el tiempo.
5. Relacionar sonidos, animaciones, textos y reacciones de Croki.
6. Registrar la actividad en el menú y en el perfil de compilación.
7. Ejecutar los casos funcionales de inicio, interacción, validación, cierre y regreso.
8. Actualizar diagramas, tablas, capturas y registro de versión.

Por ejemplo, para incorporar un rectángulo a Parque de Figuras se añade el tipo al catálogo, se crea un prefab con su configuración física y de agarre, se registra en GameManagerShapes y se incorpora un cofre cuyo ShapeChestTrigger acepte esa forma. Finalmente, se verifica aparición, manipulación, colocación, contabilización y retroalimentación. El mismo principio puede aplicarse a una nueva pareja de figuras en Tren de los Patrones o a una nueva opción en Planeta Dulzura.

4.16 Evidencias técnicas relevantes

Las capturas anteriores relacionan la descripción con configuraciones reales del proyecto. Como complemento, los siguientes fragmentos muestran decisiones de programación verificadas en el código fuente: inicio controlado después de la guía, reposición sin duplicados, generación de patrones, integración con la carga de escenas y selección de figuras sin repetición inmediata.

```
bool guideWillRun = waitForGuideBeforeStarting &&
                    level1Guide != null &&
                    (level1Guide.WillPlayOnStart ||
                     level1Guide.IsPlayingGuide);

if (!guideWillRun)
    IniciarJuego();
```

Nota: código fuente de GameManagerShapes del proyecto Crokiando (2026).

El gestor comprueba si la guía se ejecutará antes de iniciar. Cuando no existe una demostración pendiente, concentra el comienzo en IniciarJuego(); así evita depender del orden accidental de activación de otros componentes.

```
if (!juegoActivo || container == null ||
    pendingRespawns.ContainsKey(container))
    return;

pendingRespawns[container] =
    StartCoroutine(RespawnAfterDelay(container, delay));
```

Nota: código fuente de GameManagerShapes del proyecto Crokiando (2026).

La colección pendingRespawns mantiene una sola corrutina pendiente por punto de aparición. Esta comprobación impide generar reemplazos simultáneos en el mismo contenedor.

```

for (int i = 0; i < pattern.Length; i++)
{
    pattern[i] = i % 2 == 0 ? figureA : figureB;
    if (emptySlots.Contains(i))
        pendingMissingFigures.Enqueue(pattern[i]);
}

```

Nota: código fuente de ValidatePuzzle del proyecto Crokiando (2026).

La posición par o impar determina cuál de las dos figuras corresponde. Las posiciones vacías se registran en una cola para presentar las piezas que el usuario debe colocar.

```

private void OnEnable()
{
    SceneManager.sceneLoaded += OnSceneLoaded;
}

private void OnDisable()
{
    SceneManager.sceneLoaded -= OnSceneLoaded;
}

```

Nota: código fuente del sistema de integración entre escenas de Crokiando (2026).

La suscripción se realiza al habilitar el componente y se elimina al deshabilitarlo. Esta simetría permite responder a la nueva escena sin conservar escuchas duplicadas.

```

do
{
    selectedIndex = Random.Range(0, shapePrefabs.Length);
}
while (shapePrefabs.Length > 1 &&
    selectedIndex == lastShapeIndex);

```

Nota: código fuente de N3ShapeRoundManager del proyecto Crokiando (2026).

Mientras exista más de una alternativa, el gestor evita que el nuevo estímulo repita inmediatamente el índice anterior. La variación mantiene la dinámica de Planeta Dulzura sin alterar el catálogo configurado.

Tabla 34.
Relación de evidencias técnicas incorporadas

Evidencia	Contenido documentado	Sección
Arquitectura y flujo	Capas funcionales y ciclo de interacción con las manos.	4.1
Tecnologías	Unity, Meta XR, OpenXR y herramientas de presentación.	4.2
Organización	Prefabs, scripts, escenas y correspondencia de niveles.	4.3
Componentes	Manos, agarre, gestores, triggers y validadores.	4.4–4.7
Diseño y servicios	Escenarios, AudioManager y accesibilidad cromática.	4.8
Persistencia	Preferencias guardadas y estado temporal de las mecánicas.	4.9
Validación	Matriz funcional, fotografía y resultados de 20 participantes.	4.11
Despliegue	Archivo Crokiando.apk generado.	4.13
Código	Inicio, reposición, patrones, escenas y no repetición.	4.16

Nota: elaboración propia (2026).

La Tabla 34 integra las evidencias de arquitectura, configuración, código, ejecución y prueba con la sección donde se documentan, facilitando la trazabilidad del aporte técnico.

En conjunto, las evidencias muestran la relación entre la arquitectura, la configuración del editor, el código, la ejecución de las mecánicas, la interacción con participantes y el artefacto de compilación. El capítulo documenta así el aporte individual de programación evaluado en esta tesis mediante elementos verificables y vinculados con el funcionamiento real de Crokiando.

4.17 Resultados, discusión y limitaciones

4.17.1 Alcance del análisis

Este capítulo separa cuatro dimensiones que no deben confundirse: funcionamiento técnico, usabilidad, aceptación inicial y aprendizaje. Los datos disponibles permiten describir las tres primeras dentro de la muestra reportada; no incluyen una medición pretest–posttest, grupo de comparación u otro diseño que permita atribuir cambios de aprendizaje al prototipo.

La unidad de análisis corresponde al prototipo Crokiando y a los resultados agregados suministrados para 20 participantes. La interpretación se mantiene descriptiva y se limita a la evidencia contenida en la tesis y en sus capturas, tablas y fragmentos de código.

4.17.2 Funcionamiento técnico

La verificación funcional documenta el inicio de los niveles, la recepción de acciones mediante seguimiento de manos, la validación de respuestas, la retroalimentación y el cierre o retorno. Esta evidencia respalda la existencia del flujo implementado, pero no reemplaza los registros de ejecución, los casos negativos ni las mediciones repetidas en el dispositivo.

Tabla 35.
Síntesis de funcionamiento técnico y evidencia disponible

Dimensión	Evidencia disponible	Alcance o pendiente
Flujo de niveles	Matriz funcional, capturas, escenas y gestores documentados.	Faltan registros completos de pasos, resultado observado y regresión por versión.
Interacción	Selección, agarre, colocación y validación descritas con componentes XR.	Falta una bitácora de pérdidas de seguimiento, falsas selecciones y recuperación.
Despliegue	Archivo Crokiando.apk de 124 592 KB, con evidencia fechada el 27 de julio de 2026.	Faltan package ID, número de build, ABI, hash, firma e instalación limpia documentada.
Rendimiento	Estrategias de optimización y procedimiento de medición propuesto.	No se suministró una medición válida y repetida de FPS, frame time, CPU/GPU o memoria en Meta Quest 3.
Datos	No se describen perfiles infantiles ni persistencia de progreso personal.	Falta evidencia del ciclo de vida de preferencias y diagnósticos técnicos tras cierre y reinicio.

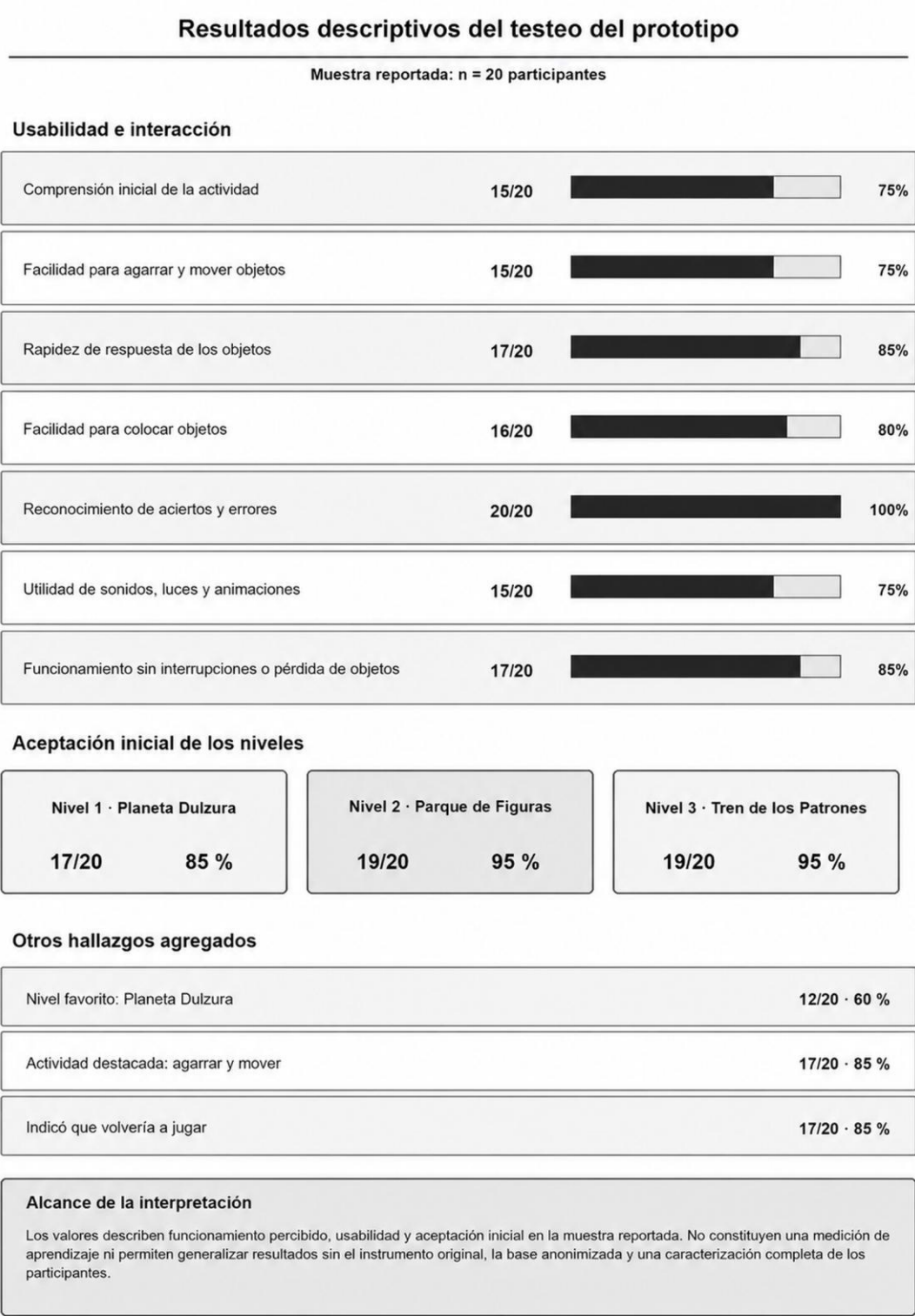
Nota: elaboración propia a partir de las evidencias técnicas disponibles en el Capítulo IV (2026).

La Tabla 35 contrasta la evidencia disponible sobre flujo, interacción, despliegue, rendimiento y datos con las comprobaciones todavía pendientes. Aunque documenta el funcionamiento y la existencia del APK, identifica la falta de registros de regresión, incidencias de seguimiento, metadatos del build y mediciones válidas en Meta Quest 3.

4.17.3 Usabilidad y aceptación inicial

Los resultados agregados muestran respuestas favorables entre 15/20 y 20/20 en los siete indicadores de usabilidad e interacción. El reconocimiento de aciertos y errores obtuvo 20/20, mientras que la comprensión inicial, el agarre y la utilidad de los recursos audiovisuales registraron 15/20. Estos valores describen percepciones y observaciones de la muestra reportada; no equivalen a una prueba de eficacia educativa.

Ilustración 50
Resultados descriptivos del testeo del prototipo



Nota: elaboración propia a partir de los resultados agregados reportados para n = 20 (2026).

La Ilustración 50 representa los resultados agregados de 20 participantes mediante frecuencias y porcentajes de usabilidad, aceptación por nivel y otros hallazgos descriptivos; estos datos no equivalen a una medición de aprendizaje.

Tabla 36.
Resultados agregados de usabilidad, interacción y aceptación inicial

Dimensión	Indicador	Respuestas favorables	Porcentaje
Usabilidad/interacción	Comprensión inicial de la actividad	15/20	75 %
Usabilidad/interacción	Facilidad para agarrar y mover objetos	15/20	75 %
Usabilidad/interacción	Rapidez de respuesta de los objetos	17/20	85 %
Usabilidad/interacción	Facilidad para colocar objetos	16/20	80 %
Usabilidad/interacción	Reconocimiento de aciertos y errores	20/20	100 %
Usabilidad/interacción	Utilidad de sonidos, luces y animaciones	15/20	75 %
Usabilidad/interacción	Funcionamiento sin interrupciones o pérdida de objetos	17/20	85 %
Aceptación inicial	Planeta Dulzura	17/20	85 %
Aceptación inicial	Parque de Figuras	19/20	95 %
Aceptación inicial	Tren de los Patrones	19/20	95 %

Nota: elaboración propia a partir de los resultados agregados reportados para n = 20 (2026).

La Tabla 36 consolida los indicadores de usabilidad e interacción y la aceptación inicial de los tres niveles para una muestra agregada de 20 participantes. Los porcentajes se sitúan entre 75 % y 100 % en usabilidad y entre 85 % y 95 % en aceptación por nivel, por lo que deben interpretarse como resultados descriptivos de la muestra y no como evidencia de aprendizaje.

En otros hallazgos agregados, 12/20 participantes eligieron Planeta Dulzura como nivel favorito, 17/20 destacaron la acción de agarrar y mover objetos y 17/20 indicaron que volverían a jugar. Sin la base anonimizada y el instrumento original no es posible reconstruir distribuciones, valores faltantes, subgrupos o asociaciones entre respuestas.

4.17.4 Discusión

La aceptación inicial observada describe agrado e intención de volver a jugar, pero no permite demostrar aprendizaje ni retención. En consecuencia, estos resultados deben interpretarse únicamente como indicadores preliminares de aceptación dentro de la muestra evaluada.

El 100 % reportado para el reconocimiento de aciertos y errores sugiere que la retroalimentación fue comprensible en la muestra evaluada. Sin embargo, la claridad de una respuesta no demuestra por sí sola la adquisición, retención o transferencia de una destreza.

La usabilidad debe interpretarse respecto de los usuarios, los objetivos y el contexto de uso específicos. Los porcentajes obtenidos aportan una primera descripción del contexto evaluado, pero la capacidad de interacción debe complementarse con verificaciones de fiabilidad, eficiencia de desempeño y mantenibilidad. Esta distinción permite conservar separados los resultados de percepción, las pruebas funcionales y las métricas de rendimiento.

4.17.5 Limitaciones metodológicas y documentales

La revisión académica identificó brechas de cierre relacionadas con participantes, instrumento, ética, pruebas técnicas y licencias. Mientras no se incorporen evidencias auténticas, estas brechas quedan fuera del alcance demostrado, limitan la evaluación y obligan a moderar las conclusiones; no se tratan únicamente como ausencias documentales menores.

Tabla 37.
Trazabilidad de información disponible y pendiente

Elemento	Disponible en la tesis	No suministrado	Implicación
Participantes	Tamaño agregado n = 20 y grupo objetivo de 4 a 5 años.	Distribución exacta de edad, institución, reclutamiento, inclusión/exclusión y experiencia previa.	No permite caracterización completa ni análisis por subgrupos.
Sesiones	Descripción general de explicación, interacción y respuesta al instrumento.	Orden, duración, pausas, ayudas, abandonos e incidencias por sesión.	No permite evaluar efecto de orden ni reproducir el procedimiento.
Instrumento	Diez preguntas agrupadas en usabilidad/interacción y aceptación.	Formulario original, escala visual, instrucciones, revisión de contenido y regla de respuesta favorable.	Limita la validez y la auditabilidad de los porcentajes.
Datos	Conteos agregados para indicadores principales.	Base anonimizada, valores faltantes y observaciones codificadas.	Impide recalcular resultados o estimar dispersión.
Rendimiento	Optimización y procedimiento propuesto.	Logs o capturas de Profiler/OVR Metrics ejecutados en Quest 3.	No permite afirmar una tasa estable de cuadros o desempeño final.
Ética	Requisitos de seguridad y supervisión descritos.	Aprobación institucional, consentimientos, asentimientos y autorización de imagen.	La trazabilidad ética no puede considerarse cerrada.

Nota: elaboración propia a partir de la auditoría del documento y del reporte de revisión académica (2026).

La Tabla 37 diferencia la información presente de la no suministrada sobre participantes, sesiones, instrumento, datos, rendimiento y ética, y explica la consecuencia de cada ausencia. Esta trazabilidad limita las inferencias posibles y señala los documentos necesarios para reproducir, auditar y cerrar la validación.

4.17.6 Trazabilidad ética

El documento establece supervisión adulta, higiene, área segura y suspensión ante malestar. No obstante, no se suministraron la aprobación institucional, los consentimientos,

los asentimientos ni la autorización de imagen. Por ello, la fotografía se omitió y los resultados permanecen como evidencia preliminar.

Antes del depósito o de una nueva sesión deben codificarse los respaldos éticos, limitarse los datos personales y documentarse duración, pausas, suspensión y responsables. Estas ausencias no se subsanan con una declaración retrospectiva.

4.17.7 Rendimiento en dispositivo

Sin una medición repetida en Meta Quest 3, la tesis no informa FPS ni frame time. La prueba pendiente debe adjuntar registros de rendimiento, memoria, errores y seguimiento del dispositivo.

4.18 Conclusiones y recomendaciones

4.18.1 Conclusiones

Objetivo específico 1. El análisis de antecedentes, currículo y requisitos permitió traducir tres destrezas observables en actividades inmersivas: reconocimiento de figuras, clasificación por forma y continuación de patrones. La evidencia documental respalda la definición de requerimientos pedagógicos, pero no demuestra un efecto de aprendizaje.

Objetivo específico 2. La estructura funcional se organizó en niveles independientes con un flujo común de preparación, guía, interacción, validación, retroalimentación y actualización. Los diagramas de componentes, estados y secuencia respaldan el diseño de mecánicas y responsabilidades.

Objetivo específico 3. El prototipo fue implementado en Unity mediante scripts en C#, seguimiento de manos, gestores de nivel, zonas de validación, audio, animación y navegación. Las capturas, componentes y descripciones técnicas respaldan la implementación del recorrido funcional documentado.

Objetivo específico 4. La integración de objetos tridimensionales, colliders, rigidbodies, interactables y servicios compartidos conectó las acciones con las reglas de cada actividad. Este cumplimiento está respaldado documentalmente; el inventario completo de recursos y licencias sigue siendo una condición previa al depósito.

Objetivo específico 5. El material suministrado reporta resultados agregados de 20 participantes, con respuestas favorables de usabilidad e interacción entre 75 % y 100 % y aceptación inicial entre 85 % y 95 %. El cumplimiento es parcial porque faltan el instrumento original, la base anonimizada, la caracterización de participantes y sesiones y los respaldos éticos previos; los datos no demuestran una mejora del aprendizaje.

Objetivo general. La aplicación fue desarrollada y su funcionamiento queda respaldado por la evidencia técnica descrita. La evaluación no puede considerarse cerrada mientras falten los registros metodológicos, éticos y reproducibles indicados. Por ello, el objetivo general se considera cumplido en su componente de desarrollo y parcialmente cumplido en su componente de evaluación; no se concluye que la aplicación haya favorecido el aprendizaje.

4.18.2 Recomendaciones

- Incorporar al expediente la aprobación institucional, consentimientos, asentimientos, autorizaciones de imagen y registros de supervisión antes de presentar los resultados como validación concluida.
- Adjuntar el instrumento original, justificar su adaptación al grupo etario, documentar la revisión de contenido y entregar una base anonimizada que permita recalculer los resultados.

- Registrar por sesión la edad exacta, contexto, experiencia previa, orden de niveles, duración, pausas, ayudas, incidencias, pérdidas de seguimiento y criterio de cierre, respetando la minimización de datos personales.
- Ejecutar un protocolo repetible de rendimiento en Meta Quest 3 y conservar capturas o exportaciones originales de FPS, frame time, CPU/GPU, memoria y errores asociados a una versión identificable del APK.
- Ampliar la matriz funcional con precondiciones, pasos, resultado esperado y observado, estado, versión, evidencia, pruebas negativas, interrupciones, cambios de escena, regresión y recuperación.
- Completar el inventario de scripts, paquetes, modelos, texturas, animaciones, audio y licencias; identificar material propio, adaptado y de terceros, y validar las contribuciones con el equipo y el tutor.
- Evaluar la comprensión del manual con un adulto no desarrollador y comprobar contraste, legibilidad, volumen y alternativas de interacción cuando el seguimiento de manos falle.

Referencias

- [1] Ministerio de Educación del Ecuador, “Currículo de Educación Inicial 2014”, Ministerio de Educación, Quito, Ecuador, 978-9942-07-625-0, 2014. [En línea]. Disponible en: <https://recursos.educacion.gob.ec/red/curriculo-inicial/>
- [2] B. Rittle-Johnson, E. R. Fyfe, K. G. Hofer, y D. C. Farran, “Early Math Trajectories: Low-Income Children’s Mathematics Knowledge From Ages 4 to 11”, *Child Dev.*, vol. 88, núm. 5, pp. 1727–1742, 2017, doi: 10.1111/cdev.12662.
- [3] M. S. Alotaibi, “Game-based learning in early childhood education: a systematic review and meta-analysis”, *Front. Psychol.*, vol. 15, p. 1307881, 2024, doi: 10.3389/fpsyg.2024.1307881.
- [4] R. Villena-Taranilla, S. Tirado-Olivares, R. Cózar-Gutiérrez, y J. A. González-Calero, “Effects of virtual reality on learning outcomes in K-6 education: A meta-analysis”, *Educ. Res. Rev.*, vol. 35, p. 100434, 2022, doi: 10.1016/j.edurev.2022.100434.
- [5] K. R. Adhe, M. A. Al Ardha, H. Herawati, S. S. Bikalawan, G. D. Lestari, y C. B. Yang, “Research Publication on Virtual Reality in Early Childhood Education: A Systematic Review”, *TEM J.*, vol. 14, núm. 1, pp. 654–660, 2025, doi: 10.18421/TEM141-58.
- [6] D. Hamilton, J. McKechnie, E. Edgerton, y C. Wilson, “Immersive virtual reality as a pedagogical tool in education: a systematic literature review of quantitative learning outcomes and experimental design”, *J. Comput. Educ.*, vol. 8, pp. 1–32, 2021, doi: 10.1007/s40692-020-00169-2.
- [7] G. Makransky y G. B. Petersen, “The Cognitive Affective Model of Immersive Learning (CAMIL): a Theoretical Research-Based Model of Learning in Immersive Virtual Reality”, *Educ. Psychol. Rev.*, vol. 33, pp. 937–958, 2021, doi: 10.1007/s10648-020-09586-2.
- [8] J. J. Cummings y J. N. Bailenson, “How Immersive Is Enough? A Meta-Analysis of the Effect of Immersive Technology on User Presence”, *Media Psychol.*, vol. 19, núm. 2, pp. 272–309, 2016, doi: 10.1080/15213269.2015.1015740.
- [9] V. J. Shute, “Focus on Formative Feedback”, *Rev. Educ. Res.*, vol. 78, núm. 1, pp. 153–189, 2008, doi: 10.3102/0034654307313795.

- [10] D. Hwang y Y. Kang, “How Does Constructive Feedback in an Educational Game Sound to Children?”, *Int. J. Child-Comput. Interact.*, vol. 36, p. 100581, 2023, doi: 10.1016/j.ijcci.2023.100581.
- [11] K. Tcha-Tokey, O. Christmann, E. Loup-Escande, y S. Richir, “Proposition and Validation of a Questionnaire to Measure the User Experience in Immersive Virtual Environments”, *Int. J. Virtual Real.*, vol. 16, núm. 1, pp. 33–48, 2016, doi: 10.20870/IJVR.2016.16.1.2880.
- [12] J. Hamari, J. Koivisto, y H. Sarsa, “Does Gamification Work?—A Literature Review of Empirical Studies on Gamification”, presentado en 2014 47th Hawaii International Conference on System Sciences, Waikoloa, HI, USA, 2014, pp. 3025–3034. doi: 10.1109/HICSS.2014.377.
- [13] M. Sailer y L. Homner, “The Gamification of Learning: a Meta-analysis”, *Educ. Psychol. Rev.*, vol. 32, núm. 1, pp. 77–112, 2020, doi: 10.1007/s10648-019-09498-w.
- [14] International Organization for Standardization, “ISO 9241-11:2018 Ergonomics of human-system interaction—Part 11: Usability: Definitions and concepts”, ISO, Geneva, Switzerland, ISO 9241-11:2018, 2018. [En línea]. Disponible en: <https://www.iso.org/standard/63500.html>
- [15] Unity Technologies, “Unity Engine”. Consultado: el 23 de julio de 2026. [En línea]. Disponible en: <https://unity.com/products/unity-engine>
- [16] Unity Technologies, “XR Interaction Toolkit 3.0.11”. Consultado: el 23 de julio de 2026. [En línea]. Disponible en: <https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@3.0/manual/index.html>
- [17] The Khronos Group, “The OpenXR 1.1.61 Specification”, OpenXR 1.1.61, 2026. Consultado: el 23 de julio de 2026. [En línea]. Disponible en: <https://registry.khronos.org/OpenXR/specs/1.1/html/xrspec.html>
- [18] Microsoft, “C# language documentation”, Microsoft Learn. Consultado: el 23 de julio de 2026. [En línea]. Disponible en: <https://learn.microsoft.com/en-us/dotnet/csharp/>
- [19] Inc. Meta Platforms, “Set up Unity for VR development”, Meta Horizon OS Developers. Consultado: el 23 de julio de 2026. [En línea]. Disponible en: <https://developers.meta.com/horizon/documentation/unity/unity-project-setup/>

- [20] International Organization for Standardization y International Electrotechnical Commission, “ISO/IEC 25010:2023 Systems and software engineering—Systems and software Quality Requirements and Evaluation (SQuaRE)—Product quality model”, ISO, Geneva, Switzerland, ISO/IEC 25010:2023, 2023. [En línea]. Disponible en: <https://www.iso.org/standard/78176.html>
- [21] S. Weech, S. Kenny, y M. Barnett-Cowan, “Presence and Cybersickness in Virtual Reality Are Negatively Related: A Review”, *Front. Psychol.*, vol. 10, p. 158, 2019, doi: 10.3389/fpsyg.2019.00158.
- [22] C. Bexson, G. Oldham, y J. Wray, “Safety of virtual reality use in children: a systematic review”, *Eur. J. Pediatr.*, vol. 183, núm. 5, pp. 2071–2090, 2024, doi: 10.1007/s00431-024-05488-5.
- [23] American Academy of Pediatrics, “Virtual Reality and Children”. Consultado: el 23 de julio de 2026. [En línea]. Disponible en: <https://www.aap.org/en/patient-care/media-and-children/center-of-excellence-on-social-media-and-youth-mental-health/qa-portal/qa-portal-library/qa-portal-library-questions/virtual-reality-use-and-children/>
- [24] Inc. Meta Platforms, “Set time limits for Meta Horizon and schedule breaks with Supervision”, Meta Quest Help Center. Consultado: el 23 de julio de 2026. [En línea]. Disponible en: <https://www.meta.com/help/quest/196832126660700/>
- [25] International Organization for Standardization, International Electrotechnical Commission, y Institute of Electrical and Electronics Engineers, “ISO/IEC/IEEE 29148:2018 Systems and software engineering—Life cycle processes—Requirements engineering”, ISO, Geneva, Switzerland, ISO/IEC/IEEE 29148:2018, 2018. [En línea]. Disponible en: <https://www.iso.org/standard/72089.html>
- [26] K. Schwaber y J. Sutherland, “La Guía Scrum: La guía definitiva de Scrum: Las reglas del juego”, nov. 2020. [En línea]. Disponible en: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf>
- [27] J. Sarama y D. H. Clements, *Early Childhood Mathematics Education Research: Learning Trajectories for Young Children*. New York, NY: Routledge, 2009. doi: 10.4324/9780203883785.

- [28] J. L. Plass, B. D. Homer, y C. K. Kinzer, “Foundations of Game-Based Learning”, *Educ. Psychol.*, vol. 50, núm. 4, pp. 258–283, 2015, doi: 10.1080/00461520.2015.1122533.
- [29] J. Radianti, T. A. Majchrzak, J. Fromm, y I. Wohlgenannt, “A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda”, *Comput. Educ.*, vol. 147, p. 103778, 2020, doi: 10.1016/j.compedu.2019.103778.
- [30] M. C. Johnson-Glenberg, “Immersive VR and Education: Embodied Design Principles That Include Gesture and Hand Controls”, *Front. Robot. AI*, vol. 5, p. 81, 2018, doi: 10.3389/frobt.2018.00081.
- [31] G. Makransky, T. S. Terkildsen, y R. E. Mayer, “Adding immersive virtual reality to a science lab simulation causes more presence but less learning”, *Learn. Instr.*, vol. 60, pp. 225–236, 2019, doi: 10.1016/j.learninstruc.2017.12.007.
- [32] Age of Learning, Inc., “ABCmouse: Early Learning Academy”. Consultado: el 23 de julio de 2026. [En línea]. Disponible en: <https://www.abcmouse.com/>
- [33] Bini Bambini, “Educational games for kids”. Consultado: el 23 de julio de 2026. [En línea]. Disponible en: <https://binibambini.com/>
- [34] Khan Academy, “Khan Academy Kids”. Consultado: el 23 de julio de 2026. [En línea]. Disponible en: <https://learn.khanacademy.org/khan-academy-kids/>
- [35] J. Jerald, *The VR Book: Human-Centered Design for Virtual Reality*. New York, NY: ACM Books, 2015. doi: 10.1145/2792790.
- [36] R. E. Mayer, *Multimedia Learning*, 3a ed. Cambridge, United Kingdom: Cambridge University Press, 2021. doi: 10.1017/9781316941355.
- [37] Organisation for Economic Co-operation and Development, *Frascati Manual 2015: Guidelines for Collecting and Reporting Data on Research and Experimental Development*. Paris, France: OECD Publishing, 2015. doi: 10.1787/9789264239012-en.
- [38] K. Peffers, T. Tuunanen, M. A. Rothenberger, y S. Chatterjee, “A Design Science Research Methodology for Information Systems Research”, *J. Manag. Inf. Syst.*, vol. 24, núm. 3, pp. 45–77, 2007, doi: 10.2753/MIS0742-1222240302.

- [39] T. Fullerton, *Game Design Workshop: A Playcentric Approach to Creating Innovative Games*, 4a ed. Boca Raton, FL: CRC Press, 2018. [En línea]. Disponible en: <https://www.routledge.com/Game-Design-Workshop-A-Playcentric-Approach-to-Creating-Innovative-Games/Fullerton/p/book/9781138098770>
- [40] International Organization for Standardization, “Ergonomics of human-system interaction — Part 210: Human-centred design for interactive systems”, ISO, Geneva, Switzerland, ISO 9241-210:2019, 2019. Consultado: el 23 de julio de 2026. [En línea]. Disponible en: <https://www.iso.org/standard/77520.html>
- [41] Unity Technologies, “Introduction to components”. 2026. Consultado: el 23 de julio de 2026. [En línea]. Disponible en: <https://docs.unity3d.com/6000.5/Documentation/Manual/Components.html>
- [42] E. Gamma, R. Helm, R. Johnson, y J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading, MA: Addison-Wesley, 1994.
- [43] Unity Technologies, “Create and configure a trigger collider”. 2026. Consultado: el 23 de julio de 2026. [En línea]. Disponible en: <https://docs.unity3d.com/6000.5/Documentation/Manual/collider-interactions-create-trigger.html>
- [44] Unity Technologies, “Set up UI Canvases for XR”. 2026. Consultado: el 23 de julio de 2026. [En línea]. Disponible en: <https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@3.0/manual/ui-setup.html>
- [45] Unity Technologies, “Unity Profiler”. 2026. Consultado: el 23 de julio de 2026. [En línea]. Disponible en: <https://docs.unity3d.com/6000.5/Documentation/Manual/Profiler.html>
- [46] Meta Platforms, Inc., “Controllers”. el 13 de marzo de 2026. Consultado: el 23 de julio de 2026. [En línea]. Disponible en: <https://developers.meta.com/horizon/design/controllers/>
- [47] Unity Technologies, “XR Interaction Manager”. 2026. Consultado: el 23 de julio de 2026. [En línea]. Disponible en: <https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@3.0/manual/xr-interaction-manager.html>
- [48] Blender Foundation, “Blender”. 2026. Consultado: el 23 de julio de 2026. [En línea]. Disponible en: <https://www.blender.org/>

- [49] Git Project, “git Documentation”. 2026. Consultado: el 23 de julio de 2026. [En línea]. Disponible en:
<https://git-scm.com/docs/git>