



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 recursos tridimensionales para una guía interactiva
orientada al aprendizaje guiado del entorno de Unity**

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

AUTOR: JOSE ALBERTO GUILLEN PARRA

DIRECTOR: ING. JHEYSON STEVEN GAONA PINEDA, MTR.

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 recursos tridimensionales para una guía interactiva
orientada al aprendizaje guiado del entorno de Unity

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

AUTOR: JOSE ALBERTO GUILLEN PARRA

DIRECTOR: ING. JHEYSON STEVEN GAONA PINEDA, MTR.

CUENCA - ECUADOR

2026

DIOS, PATRIA, CULTURA Y DESARROLLO



Universidad
Católica
de Cuenca

DECLARATORIA DE AUTORÍA Y RESPONSABILIDAD

Declaratoria de Autoría y Responsabilidad

José Alberto Guillén Parra portador de la cédula de ciudadanía N.º 0302459748. Declaro ser el autor de la obra: **"Desarrollo de recursos tridimensionales para una guía interactiva orientada al aprendizaje guiado del entorno de Unity."**, 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, 26 de agosto de 2026

F: 

José Alberto Guillén Parra

C.I. 0302459748



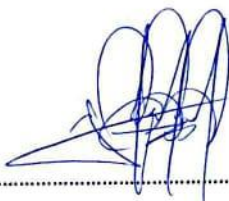
Universidad
Católica
de Cuenca

CERTIFICADO

Certificado

Certifico que el presente Trabajo de Investigación fue desarrollado por José Alberto Guillén Parra, bajo mi supervisión portador(a) de la cédula de ciudadanía N.º 0302459748 con el tema **"Desarrollo de recursos tridimensionales para una guía interactiva orientada al aprendizaje guiado del entorno de Unity"**, bajo mi supervisión.

Cuenca, 26 de agosto de 2026

F: 

Ing. Jheyson Steven Gaona Pineda, Mtr.

Docente - Tutor

Dedicatoria

Este trabajo está dedicado, por sobre todo a mi abuelo, Tito, quien estuvo conmigo desde que entre a la universidad, pero que lamentablemente no pudo estar para verme salir. El siempre soñaba con ver mi graduación y sé que desde el cielo me acompañara siempre.

A mi abuela, Pia, quien me acompañó durante toda esta etapa, por cuidarme y por velar siempre por mí, y sé que siempre estarás en las próximas etapas de mi vida.

A mis padres, quienes, a pesar de encontrarse lejos, siempre me han apoyado y guiado durante todo este proceso y me van a seguir guiando de aquí en adelante.

A mis gatos, Thomas y Mercedes, por ser mis compañeros más fieles durante estos años, y verme en mis buenos y malos momentos, y por siempre acompañarme a pesar de todo.

Todos ellos y muchas más personas han sido parte de cómo me he construido durante estos años en la universidad, y lo van a seguir siendo el resto de mi vida.

José Alberto Guillén Parra

Agradecimiento

Agradezco a Dios por todas las oportunidades que me ha dado en la vida, y por estar siempre conmigo a pesar de no ser alguien perfecto.

A mis abuelos, por apoyarme y guiarme desde que nací, y por el esfuerzo que hacían para que yo pudiera salir adelante en la vida.

A mis padres, por apoyarme a pesar de la distancia, y por aconsejarme para ser una mejor persona, por su perseverancia y la paciencia que siempre han tenido conmigo.

A mis hermanos por haberme inculcado cada uno un poco de sus valores y principios, y por haberme ayudado cuando los necesitaba.

A mi compañera de tesis, Britney, por haberme ayudado a desarrollar este proyecto y por haberme enseñado en momentos de confusión, le deseo lo mejor de aquí en adelante.

Por último, agradezco a mi tutor Jheyson, y al resto de mis maestros, por habernos acompañado en este y en todos los procesos que hemos atravesado, por habernos inculcado cada uno su granito de arena que nos llevara a un mejor futuro como profesionales.

A todos ellos les deseo los mejores éxitos en su vida, tanto personal como profesional.

José Alberto Guillén Parra

Resumen

El modelado tridimensional Low Poly constituye una técnica de representación gráfica orientada a optimizar la geometría mediante el uso controlado de polígonos, permitiendo desarrollar recursos visuales estilizados y eficientes para entornos interactivos en tiempo real, sin comprometer la coherencia estética establecida. El presente proyecto aborda la creación de recursos tridimensionales hechos en Blender para el desarrollo de una guía interactiva implementada en Unity. El proceso comprende el modelado Low Poly del personaje jugable, objetos interactivos y elementos decorativos, seguido del mapeado UV, aplicación de mapas de color y configuración de materiales reutilizables para optimizar los recursos gráficos. Para el personaje se implementa un sistema de rigging mediante armature y weight painting, posibilitando deformaciones controladas y la generación de animaciones básicas. Los recursos producidos se integran posteriormente en Unity mediante prefabs, materiales, colisionadores y un Animator Controller para gestionar las interacciones y estados de animación. De manera complementaria, se desarrolla la propuesta gráfica de un libro guía, considerando criterios de composición, jerarquía visual, distribución espacial y coherencia cromática. El desarrollo emplea la metodología en cascada, estructurando secuencialmente las fases de planificación, diseño, producción, integración y validación, obteniendo como resultado una guía interactiva funcional, coherente y visualmente consistente. El rendimiento de la aplicación se analiza mediante Unity Profiler, evaluando su comportamiento durante la ejecución. Complementariamente, se aplica una escala de Likert para valorar el modelado, estilo artístico, selección cromática, legibilidad y coherencia temática, determinando así la factibilidad técnica y visual de la propuesta desarrollada.

Palabras clave: *Modelado Low Poly, Diseño visual, Guía interactiva, Blender.*

Abstract

Low Poly three-dimensional modeling is a graphic representation technique aimed at optimizing geometry through the controlled use of polygons, enabling the development of stylized and efficient visual assets for real-time interactive environments without compromising the established aesthetic coherence. This project focuses on the creation of three-dimensional assets in Blender for the development of an interactive guide implemented in Unity. The process involves low-poly modeling of the playable character, interactive objects, and decorative elements, followed by UV mapping, the application of color maps, and the configuration of reusable materials to optimize graphic resources. For the character, a rigging system is implemented using an armature and weight painting, enabling controlled deformations and the creation of basic animations. The resulting assets are subsequently integrated into Unity through prefabs, materials, colliders, and an Animator Controller to manage interactions and animation states. Additionally, there is a graphic design proposal for a guidebook, considering criteria such as composition, visual hierarchy, spatial distribution, and color consistency. The development follows the waterfall methodology, sequentially structuring the planning, design, production, integration, and validation phases, resulting in a functional, coherent, and visually consistent interactive guide. The application's performance is analyzed with Unity Profiler, evaluating its behavior during operation. Additionally, a Likert scale is applied to assess the modeling, artistic style, color selection, readability, and thematic coherence, thereby determining the technical and visual feasibility of the developed proposal.

Keywords: *Low Poly modeling, visual design, interactive guide, Blender*

Índice De Contenido

Declaratoria de autoría y responsabilidad	II
Certificado	III
Dedicatoria	IV
Agradecimiento	V
Resumen	VI
Abstract.....	VII
Índice de Tablas.....	XI
Índice de Ilustraciones.....	XII
Introducción.....	1
Capítulo I.....	3
1. Marco Referencial	3
1.1. Contextualización del Problema	3
1.2. Planteamiento del problema.....	4
1.3. Formulación del problema	5
1.4. Justificación	5
1.5. Objetivo General.....	8
1.6. Objetivos Específicos	8
1.7. Alcance del proyecto.....	8
1.8. Delimitaciones	10
1.9. Beneficiarios	11
1.9.1. Beneficiarios Directos:	11
1.9.2. Beneficiarios Indirectos:.....	11
1.10. Metodología general.....	12
1.11. Resultados esperados	14
1.11.1. Modelado 3D de Personajes y Plataformas Modulares:	14
1.11.2. Ensamblaje del Escenario y Colisionadores:.....	14
1.11.3. Integración de Texturas, Materiales e Iluminación:	15
1.11.4. Sistema de Animaciones y Controlador del Personaje:.....	15
Capítulo II.....	16
2. Marco Teórico	16
2.1. Antecedentes de investigación.....	16
2.1.1. Educación y Competencias de Programación	16
2.1.2. Visualización y Gráficos en Tiempo Real.....	16

2.1.3.	Impacto Social y Realidad Virtual.....	17
2.1.4.	Metodologías de Diseño y Creación.....	17
2.1.5.	Cuadro Comparativo de trabajos relacionados.....	18
2.2.	Bases teóricas.....	19
2.2.1.	Modelado.....	19
2.2.2.	Texturizado.....	21
2.2.3.	Rigging.....	22
2.3.	Bases metodológicas.....	25
2.3.1.	Scrum.....	25
2.3.2.	Waterfall (Modelo en Cascada).....	26
2.3.3.	Design Thinking.....	27
2.4.	Bases tecnológicas.....	31
2.4.1.	Blender.....	31
2.4.2.	Autodesk Maya.....	32
2.4.3.	Maxon Zbrush.....	33
2.4.4.	Krita.....	34
2.4.5.	Adobe Photoshop.....	35
2.4.6.	Cuadro comparativo de Herramientas.....	36
2.5.	Arquitecturas, modelos y estándares.....	37
2.5.1.	Arquitectura de Personaje.....	38
2.5.2.	Arquitectura de Escenarios.....	38
2.5.3.	Modelo basado en referencias.....	39
2.5.4.	Modelo de validación técnica.....	40
2.5.5.	Estandar de escala y unidades.....	41
2.5.6.	Estandar de animación.....	41
2.6.	Herramientas de desarrollo.....	42
2.7.	Marco conceptual.....	43
Capítulo III		46
3.	Manual del Programador.....	46
3.1.	Dirección de arte y propuesta visual.....	46
3.2.	Diseño de personajes, escenarios y elementos gráficos.....	46
3.3.	Producción de assets 2D y 3D.....	49
3.4.	Pipeline de producción artística.....	52
3.4.1.	Desarrollo del personaje jugable.....	52

3.4.2.	Diseño modular del escenario y elementos del entorno	54
3.4.3.	Colisionadores personalizados y geometría base	56
3.5.	Preparación y animación de personajes y objetos.....	58
3.5.1.	Construcción y adaptación de la armadura (Rigging)	58
3.5.2.	Vinculación y depuración de pesos (Skinning)	59
3.5.3.	Generación de los clips de animación del personaje	61
3.5.4.	Animación técnica de elementos del escenario en Unity	62
3.6.	Materiales, texturas, iluminación y renderización	63
3.6.1.	Generación de mapas de textura.....	63
3.6.2.	Configuración e integración de materiales en Unity	65
3.7.	Creación de efectos visuales	66
3.7.1.	Efectos de Postprocesamiento (Bloom y Color Adjustments)	67
3.7.2.	Configuración de la Atmósfera y Entorno (Lighting y Fog)	68
3.8.	Integración de recursos visuales en el motor	69
3.8.1.	Integración y Configuración del Personaje Jugable	69
3.8.2.	Integración del Escenario y Plataformas Modulares	71
3.8.3.	Integración de Elementos Decorativos	73
3.9.	Optimización gráfica.....	74
3.9.1.	Optimización Geométrica y de Recursos	74
3.9.2.	Optimización de Iluminación y Sombras	75
3.9.3.	Optimización de Materiales y Texturas.....	75
3.9.4.	Optimización de Animaciones y Físicas	75
3.10.	Pruebas de rendimiento en motor grafico – Unity Profiler	76
3.11.	Validación técnica de recursos visuales	77
3.12.	Conclusiones y recomendaciones	80
3.12.1.	Conclusiones.....	80
3.12.2.	Recomendaciones	80
	Referencias	82
	Anexos	90

Índice de Tablas

Tabla 1 Comparación de los trabajos relacionados	18
Tabla 2 Comparación de las metodologías Fuente: Autor Propio	29
Tabla 3 Comparación de Herramientas Tecnológicas	36
Tabla 4 Modelos 3D creados	51
Tabla 5 Análisis de Datos obtenidos en la encuesta	78

Índice de Ilustraciones

Ilustración 1 Metodología Cascada aplicada al proyecto.	12
Ilustración 2 Personaje principal en el escenario.	47
Ilustración 3 Escenario en Unity.	48
Ilustración 4 Modelos de plataformas.	49
Ilustración 5 Modelos de elementos decorativos.	49
Ilustración 6 Ficha de personaje.	53
Ilustración 7 Modelo de Personaje.	54
Ilustración 8 Modelos base de Plataformas.	55
Ilustración 9 Modelos base de elementos decorativos.	56
Ilustración 10 Colliders Personalizados.	57
Ilustración 11 Paredes en Unity.	57
Ilustración 12 Armature en personaje principal.	59
Ilustración 13 Errores en Weight Paint Fuente.	60
Ilustración 14 Weight Paint arreglado a mano.	60
Ilustración 15 Clips de animación en Blender.	62
Ilustración 16 Animación de UV.	63
Ilustración 17 Texturas en Unity.	64
Ilustración 18 Material en Unity.	66
Ilustración 19 Efectos de PostProcesado.	67
Ilustración 20 Efectos de Iluminación del Entorno.	68
Ilustración 21 Integración de Personaje y animaciones dentro de las carpetas de Unity....	70
Ilustración 22 Configuración de Collider de personaje.	71
Ilustración 23 Jerarquía y efecto de Dash en Unity.	71
Ilustración 24 Importación y edición de una plataforma modular en Unity.	72
Ilustración 25 Importación y edición de un Collider Personalizado en Unity.	73
Ilustración 26 Importación de elementos decorativos a Unity.	74
Ilustración 27 Escena en Unity con profiler.	77

Introducción

El desarrollo de la tecnología educativa mediante entornos interactivos ha transformado de manera significativa los procesos de enseñanza y aprendizaje contemporáneos. En este ámbito, la estructuración y la representación visual de la información desempeñan un rol fundamental, dado que contribuyen a reducir la carga cognitiva del estudiante y facilitan la asimilación de conceptos abstractos. Sin embargo, durante la creación de proyectos tecnológicos en el entorno académico, es habitual que se prioricen los aspectos puramente funcionales del software, relegando la dirección de arte a un plano secundario.

Esta falta de integración metodológica genera una desconexión en la construcción del proyecto. Cuando los modelos tridimensionales, los componentes de la interfaz de usuario y los demás recursos gráficos se desarrollan de manera aislada y carecen de una guía estética unificada, la experiencia del usuario se fragmenta. Como consecuencia, la incoherencia visual dificulta la navegación fluida dentro de la aplicación y disminuye la claridad didáctica, obstaculizando la comprensión efectiva de contenidos técnicos y abstractos relacionados con la lógica y la programación.

Con el fin de dar respuesta a esta problemática, el presente trabajo aborda el diseño, desarrollo e integración de un sistema visual coherente adaptado a un entorno interactivo educativo. Para alcanzar este propósito, se aplican técnicas de modelado 3D bajo la estética low poly y un flujo de trabajo optimizado en el motor de desarrollo Unity. El objetivo

principal reside en evaluar el impacto directo que genera la coherencia estética en la percepción de claridad y calidad comunicativa del entorno por parte de los usuarios. De este modo se busca demostrar que el diseño visual no actúa únicamente como un complemento estético, sino como un pilar pedagógico clave para potenciar la experiencia educativa digital.

Capítulo I

1. Marco Referencial

1.1. Contextualización del Problema

En el desarrollo de entornos digitales con fines educativos, la forma en que se presenta la información influye directamente en la comprensión del usuario. No se trata únicamente del contenido, sino de cómo este se encuentre estructurado, representado y percibido a través de elementos visuales. Diversos estudios en el campo del aprendizaje multimedia han demostrado que una inadecuada organización visual puede incrementar la carga cognitiva del estudiante, dificultando la interpretación de conceptos abstractos y reduciendo la efectividad del proceso de aprendizaje [1].

En este sentido, tanto los entornos tridimensionales como los recursos gráficos bidimensionales deben ser diseñados bajo criterios que faciliten la construcción de significado. En contextos donde se combinan entornos interactivos con contenido educativo, como en el desarrollo de videojuegos formativos, la problemática se intensifica debido a la coexistencia de múltiples tipos de representación: escenarios 3D, personajes, interfaces visuales, diagramas y materiales de apoyo. Investigaciones sobre aprendizaje multimedia señalan que la integración coherente de estos elementos permite al usuario relacionar información visual con conceptos teóricos, favoreciendo la comprensión [2].

Sin embargo, cuando estos componentes se diseñan de forma aislada o sin una intención comunicativa clara, se genera fragmentación en la experiencia del usuario, dificultando la interpretación del contenido. En relación con el desarrollo del sistema propuesto, se evidencia que los componentes visuales actuales no establecen una relación clara entre la representación gráfica y el contenido que se busca transmitir. Los personajes,

entornos y elementos gráficos no cuentan con una dirección del contexto local, particularmente en entornos de formación tecnológica en la ciudad de Cuenca, se observa que los proyectos académicos orientados al desarrollo de aplicaciones interactivas suelen centrarse en los aspectos funcionales, dejando en segundo plano el diseño visual como herramienta pedagógica. Como resultado, tanto los modelos tridimensionales como los elementos gráficos como diagramas o materiales de lectura carecen de una estructura visual coherente, lo que limita la capacidad del usuario para interpretar la información presentada. Esta desconexión ha sido identificada en estudios de diseño instruccional como un factor que reduce la retención del conocimiento y la comprensión en entornos digitales [3].

1.2. Planteamiento del problema

En relación con el desarrollo del sistema propuesto, se evidencia que los componentes visuales actuales no establecen una relación clara entre la representación gráfica y el contenido que se busca transmitir. Los personajes, entornos y elementos gráficos no cuentan con una dirección de diseño unificado, lo que dificulta el poder identificar funciones, jerarquías visuales y relaciones espaciales dentro del entorno. Asimismo, la ausencia de criterios en la organización de información en recursos como textos y diagramas limita la comprensión de los contenidos presentados al usuario. Las causas de esta problemática se relacionan con la fragmentación del proceso de diseño, donde el modelado 3D, el diseño gráfico y la estructuración de la información se desarrollan de manera independiente, sin un enfoque integrador. Esta separación impide que los elementos visuales funcionen como un sistema coherente de comunicación. Desde una perspectiva contemporánea, el diseño visual en entornos digitales debe entenderse como un proceso estratégico orientado a facilitar la

comprensión, donde cada elemento ya sea tridimensional o gráfico cumple una función dentro de la experiencia del usuario [4].

Si esta problemática no es abordada, el entorno desarrollado corre el riesgo de presentar información de manera desarticulada, lo que puede derivar en una experiencia de aprendizaje superficial, caracterizada por la interacción sin comprensión. Esto afecta directamente la retención del conocimiento y la capacidad del usuario para aplicar lo aprendido en otros contextos.

1.3. Formulación del problema

En este sentido, el problema se delimita dentro del campo del diseño visual aplicado a entornos interactivos educativos, específicamente en el área de desarrollo de videojuegos, enfocado en la creación de sistemas visuales que articulen modelado 3D, diseño gráfico y organización de la información. Bajo este contexto, se identifica como variable independiente la coherencia del sistema visual (modelado 3D, diseño gráfico y organización de la información), y como variable dependiente la percepción de claridad, navegación y calidad comunicativa del entorno por parte del usuario. En consecuencia, se desconoce en qué medida la coherencia de un sistema visual influye en la percepción y comprensión de la estructura de un entorno interactivo educativo, lo que plantea la necesidad de investigar: ¿cómo la coherencia del sistema visual en un entorno interactivo influye en la percepción de claridad, navegación y calidad comunicativa por parte del usuario?

1.4. Justificación

La necesidad de desarrollar el presente trabajo se fundamenta en la importancia de mejorar, mediante un sistema visual coherente, la forma en que se presenta y se percibe la información dentro de entornos interactivos educativos. En procesos de formación inicial,

especialmente en carreras vinculadas al desarrollo de software y videojuegos, los estudiantes enfrentan dificultades para interpretar conceptos abstractos cuando estos no cuentan con un soporte visual adecuado. Diversos estudios han demostrado que el uso de representaciones visuales estructuradas permite reducir la carga cognitiva y facilita la comprensión de información compleja, favoreciendo el aprendizaje significativo [1]. En este sentido, el diseño visual deja de ser un componente estético para convertirse en un recurso pedagógico clave dentro de los entornos digitales. Desde el contexto local, particularmente en instituciones de educación superior orientadas a la formación tecnológica, se ha identificado que los proyectos académicos relacionados con el desarrollo de videojuegos suelen centrarse en los aspectos funcionales, relegando el diseño visual a un rol secundario. Esto genera entornos interactivos donde los elementos tridimensionales, interfaces visuales y materiales de apoyo no cuentan con una estructura coherente que facilite la interpretación del contenido por parte del estudiante. Como consecuencia, se dificulta la comprensión de conceptos fundamentales en etapas iniciales de aprendizaje, lo que puede afectar el rendimiento académico y la motivación. Esta problemática evidencia la necesidad de incorporar enfoques de diseño visual que respondan a las características reales del entorno educativo en el que se desarrollan estos proyectos. Desde una perspectiva técnica, el desarrollo de un sistema visual integral que combine modelado 3D, diseño gráfico y organización de la información permite optimizar tanto la experiencia del usuario como el rendimiento del entorno interactivo. El uso de técnicas como el modelado low poly contribuye a mantener un equilibrio entre calidad visual y eficiencia en tiempo real, aspecto fundamental en aplicaciones educativas donde se requiere estabilidad y accesibilidad. Estudios sobre optimización en entornos gráficos en tiempo real destacan que la reducción de geometría y la correcta gestión de recursos visuales influyen directamente en el rendimiento del sistema [2]. Asimismo, la integración de

elementos gráficos bien estructurados, como diagramas y materiales visuales, refuerza la claridad en la transmisión de información. En el ámbito académico y social, este trabajo aporta al fortalecimiento de metodologías de aprendizaje basadas en entornos interactivos, las cuales han demostrado incrementar la motivación, la participación activa y la retención del conocimiento en los estudiantes. Al proponer un sistema visual que articule de manera coherente los diferentes elementos del entorno como personajes, escenarios y recursos gráficos se contribuye a mejorar la calidad de las herramientas educativas utilizadas en procesos formativos.

Los beneficiarios directos de este proyecto son estudiantes en formación inicial en áreas relacionadas con programación y desarrollo de videojuegos, quienes podrán interactuar con un entorno diseñado para facilitar la comprensión de los contenidos desde una perspectiva visual.

Desde una perspectiva científica, este trabajo aporta al estudio del diseño visual aplicado a entornos interactivos educativos, al analizar la relación entre la coherencia de los elementos visuales y la comprensión del usuario. A diferencia de enfoques centrados únicamente en la implementación técnica, la presente investigación busca generar evidencia sobre cómo la organización visual, la jerarquía gráfica y la consistencia en el diseño influyen en la interpretación de contenidos dentro de entornos tridimensionales. Este aporte permite establecer criterios y lineamientos que pueden ser utilizados en futuros desarrollos educativos interactivos, contribuyendo a la consolidación del diseño visual como un componente fundamental dentro de los procesos de enseñanza-aprendizaje en medios digitales.

1.5. Objetivo General

Diseñar e implementar un sistema de recursos tridimensionales estilizados bajo la técnica Low Poly e integrarlos en Unity, evaluando la percepción visual, usabilidad y coherencia estética del escenario por parte del público objetivo.

1.6. Objetivos Específicos

- Determinar los criterios de coherencia visual y organización informativa necesarios para el diseño de recursos tridimensionales para un escenario de Unity.
- Diseñar los recursos tridimensionales, (personaje, plataformas y elementos decorativos) mediante técnicas de modelado tridimensional de baja complejidad y herramientas de diseño digital en Blender.
- Implementar los recursos 3D, animaciones, iluminación y colisionadores dentro del motor gráfico Unity.
- Evaluar la calidad visual, la fluidez al momento de navegar por el entorno interactivo mediante herramientas de profiling y pruebas con usuarios.

1.7. Alcance del proyecto

El presente trabajo se enfoca en el diseño y desarrollo de un sistema visual aplicado a un entorno interactivo educativo, considerando la creación de personajes, escenarios y recursos gráficos mediante técnicas de modelado 3D low poly, diseño gráfico y su integración en un motor de tiempo real. Este proceso contempla la definición de una coherencia visual basada en criterios de organización, jerarquía y consistencia, orientados a facilitar la interpretación de los elementos dentro del entorno.

Adicionalmente, el alcance del proyecto no se limita únicamente a la producción de los elementos visuales, sino que incluye la evaluación de su impacto en la comprensión del usuario.

Seguidamente se enlista los elementos a realizar como gestor del proyecto:

- **Modelado 3D:** Creación de modelos low-poly estilizado para el proyecto, el modelado del personaje principal y escenarios.
- **Texturizado:** Se crean texturas para el material de los modelos, El UV Mapping permite distribuir correctamente las texturas sobre los modelos tridimensionales, asegurando una adecuada correspondencia entre la geometría y su apariencia visual.
- **Animación:** Animaciones desarrolladas para la mejor comprensión visual del proyecto, la creación de armature permite dotar al modelo de una estructura ósea necesaria para que pueda ser animado correctamente; posteriormente, el weight paint se utiliza para definir cómo se deforman las mallas durante la animación, asegurando movimientos naturales y coherentes. A partir de ello, se desarrollan animaciones básicas que facilitan la interpretación de las acciones del personaje dentro del entorno interactivo.
- **Diseño de Entorno (Level Design):** Construcción de un escenario dividido en zonas funcionales delimitadas por 100x100 unidades, la construcción del entorno consiste en crear un espacio coherente donde se desarrollan las actividades del usuario.
- **Integración en Unity:** Configuración de materiales, prefabs y colisionadores, la importación de assets consiste en trasladar los modelos desde Blender a Unity manteniendo su estructura y configuración original, asegurando su correcta integración en el entorno.

- **Optimización Visual:** Aplicación de técnicas de iluminación y post-procesado para lograr una estética neon futurista, las técnicas de iluminación, se aplican con el objetivo de equilibrar la calidad visual y el rendimiento según las necesidades del proyecto.

1.8. Delimitaciones

El desarrollo del sistema visual y técnico para el entorno interactivo sigue las siguientes delimitaciones artísticas, de animación y de optimización:

- **Delimitación Estilística y de Modelado 3D:** La creación gráfica de personajes y objetos del entorno se limita estrictamente a un estilo Low Poly de baja poligonización (reducida cantidad de vértices y triángulos). No se contempla el uso de acabados o detalles hiperrealistas.
- **Delimitación de Materiales y Texturizado:** Las texturas se restringen al uso de colores planos (flat colors) de baja resolución. Se omite por completo la implementación de mapas de normales (normal maps) o micro-relieves, y los efectos lumínicos locales (como brillos o neones) se simulan mediante máscaras emisivas directamente en las texturas, evitando el uso de luces físicas puntuales o geometría adicional.
- **Delimitación del Diseño de Entorno (Level Design):** La construcción del espacio tridimensional no abarca arquitecturas complejas ni escenarios expansivos. El diseño del nivel se limita a estructuras geométricas sencillas compuestas por primitivas modificadas (cubos editados para conformar paredes) dentro de zonas funcionales delimitadas.
- **Delimitación de Iluminación y Físicas en Motor:** Se evita el uso de luces direccionales principales y del cálculo de sombras dinámicas en tiempo real para todos los modelos del juego, resolviendo la atmósfera espacial únicamente con gradientes de luz ambiental

(Environment Lighting) y niebla (Fog). En cuanto a las físicas, la interacción espacial se delimita al uso de colisionadores primitivos sencillos (Box Colliders), restringiendo los colisionadores de malla (Mesh Colliders) únicamente a casos excepcionales de geometría irregular.

1.9. Beneficiarios

El impacto y aprovechamiento de la presente propuesta tecnológica se estructura en dos niveles de beneficiarios:

1.9.1. Beneficiarios Directos:

- **Estudiantes y usuarios en entornos de formación tecnológica:** Son los principales destinatarios de la propuesta. Al interactuar con un entorno interactivo diseñado bajo criterios de coherencia visual, se benefician de una mayor claridad visual y de una interpretación más intuitiva de la información presentada en el entorno. La correcta organización espacial, jerarquía gráfica e integración en Unity facilitan la comprensión de funciones dentro del videojuego y optimizan la retención del conocimiento.

1.9.2. Beneficiarios Indirectos:

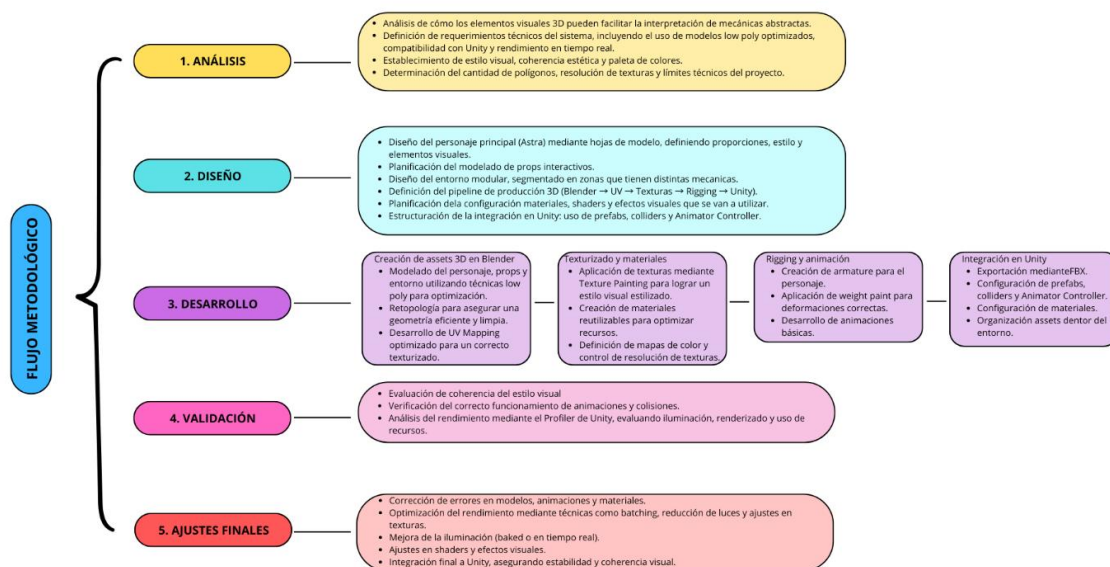
- **Desarrolladores y diseñadores de videojuegos formativos:** Desarrolladores, modeladores y usuarios que desarrollen proyectos académicos (especialmente dentro del contexto de formación tecnológica local) que obtienen una referencia metodológica sobre cómo integrar de manera coordinada el modelado 3D low poly, la interfaz gráfica y las técnicas de optimización en motores en tiempo real.

1.10. Metodología general

La metodología aplicada se basa en un enfoque secuencial estructurado bajo el modelo en cascada, el cual organiza el desarrollo en fases progresivas que permiten un control ordenado del proceso, desde la conceptualización hasta la validación final. Este enfoque se adapta a la naturaleza del proyecto, ya que permite definir con claridad los requerimientos visuales antes de su implementación, asegurando coherencia entre los elementos diseñados y los objetivos planteados. Cada fase se desarrolla de manera consecutiva, utilizando como base los resultados obtenidos en la etapa anterior, lo que garantiza consistencia en el sistema visual.

Ilustración 1

Metodología Cascada aplicada al proyecto.



Nota. Elaboración propia.

En la fase de análisis, se identifican los requerimientos visuales del entorno educativo, considerando la interpretación de conceptos abstractos a través de elementos gráficos y tridimensionales. En esta etapa se definen criterios como el estilo visual, la paleta de colores,

la jerarquía de elementos y las limitaciones técnicas relacionadas con el rendimiento en tiempo real. Además, se establecen los lineamientos para la construcción del entorno, personajes y recursos gráficos, asegurando que todos los componentes respondan a una intención comunicativa clara.

Durante la fase de diseño, se desarrollan las propuestas visuales que darán forma al sistema, incluyendo la creación conceptual del personaje, el entorno modular y los elementos gráficos de apoyo como diagramas y materiales visuales. Se elaboran bocetos, guías visuales y esquemas de organización de la información que definen la estructura del contenido dentro del entorno interactivo. Asimismo, se planifica el flujo de trabajo para la producción 3D, estableciendo el pipeline que abarca modelado, texturizado, rigging y animación, junto con los criterios de integración en el motor gráfico.

En la fase de desarrollo, se ejecuta la producción de los activos visuales definidos previamente. Esto incluye el modelado 3D de personajes y objetos mediante técnicas low poly en Blender, la creación de texturas y materiales, así como la configuración de estructuras de animación. De forma paralela, se desarrollan los elementos gráficos necesarios para la presentación de información, asegurando coherencia visual en todos los componentes. Posteriormente, los activos son integrados en Unity, donde se configuran materiales, animaciones y organización de escena, garantizando su correcto funcionamiento desde el punto de vista visual.

En la fase de validación, se evalúa la coherencia del sistema visual desarrollado, verificando que los elementos diseñados cumplan con su función comunicativa dentro del entorno. Se realizan pruebas orientadas a analizar la claridad visual, la correcta interpretación de los elementos por parte del usuario y el rendimiento del sistema en tiempo real, utilizando

herramientas de análisis como el profiler del motor gráfico. Esta etapa permite identificar posibles inconsistencias o dificultades en la comprensión del entorno.

Finalmente, en la fase de ajustes finales, se corrigen los aspectos detectados durante la validación, optimizando modelos, texturas y configuraciones visuales para mejorar tanto el rendimiento como la claridad del sistema. Se aplican técnicas de optimización como reducción de geometría, mejora de iluminación y ajustes en materiales, asegurando un equilibrio entre calidad visual y eficiencia. Esta fase concluye con la consolidación del sistema visual integrado, listo para su implementación en el entorno interactivo.

1.11. Resultados esperados

Al concluir el proyecto, se obtendrá un conjunto de productos finales que conforman un sistema visual e interactivo completamente estructurado e integrado en el motor de juego Unity, los cuales se detallan a continuación:

1.11.1. Modelado 3D de Personajes y Plataformas Modulares:

- Modelos tridimensionales finalizados bajo la técnica low poly, correspondientes al personaje principal y a un conjunto de plataformas modulares y elementos decorativos adaptados a la línea estética del proyecto.

1.11.2. Ensamblaje del Escenario y Colisionadores:

- Escenario tridimensional estructurado y armado en Unity, compuesto por la disposición coordinada de paredes, plataformas modulares y objetos decorativos.
- Implementación y asignación de colisionadores (colliders) en paredes, plataformas y elementos del entorno, garantizando la física básica y la navegación espacial fluida del usuario.

1.11.3. Integración de Texturas, Materiales e Iluminación:

- Configuración completa de materiales y asignación de texturas en todos los objetos tridimensionales dentro del motor.
- Configuración del esquema de iluminación ambiental y efectos estéticos aplicados al nivel para consolidar la atmósfera visual deseada.

1.11.4. Sistema de Animaciones y Controlador del Personaje:

- Integración del Animator Controller en el personaje principal junto con la vinculación de sus respectivos clips de animación cortos para las acciones dentro del entorno.
- Implementación de animaciones específicas e indicadores dinámicos en los elementos del escenario y plataformas que lo requieran para reforzar el comportamiento del sistema visual.

Capítulo II

2. Marco Teórico

2.1. Antecedentes de investigación

En esta sección se presentan los trabajos relacionados con el objetivo de investigación, con el fin de identificar antecedentes y obtener insumos teóricos y metodológicos que sirvan de base para el desarrollo de la presente investigación.

2.1.1. Educación y Competencias de Programación

Investigaciones recientes demuestran que el desarrollo de videojuegos es una herramienta pedagógica poderosa que fomenta competencias de programación en estudiantes universitarios de manera más efectiva que los métodos tradicionales [4]. El uso de herramientas de programación visual, como el Visual Scripting en Unity o los bloques de lógica en Blender, reduce las barreras de entrada para principiantes que encuentran dificultades en los lenguajes basados en texto [5]. Estos entornos permiten a los estudiantes centrarse en la lógica del juego y la resolución de problemas antes de enfrentarse a la sintaxis compleja del código [6]. Además, la transición de bloques visuales a código profesional es un área de estudio crítica para mejorar el aprendizaje en ciencias de la computación [4].

2.1.2. Visualización y Gráficos en Tiempo Real

El trabajo de Galán (2021) sobre la producción de escenarios óptimos destaca que la integración de escultura digital, retopología y texturizado profesional es necesaria para alcanzar estándares de la industria como los de "The Rookies" [2]. Por otro lado, la creación de museos virtuales para la difusión de patrimonio, como modelos 3D de vestimenta histórica, demuestra la versatilidad de estas tecnologías más allá del entretenimiento [7]. La

precisión en la recreación de objetos históricos requiere un manejo avanzado de herramientas CAD y simulación de telas para garantizar la fidelidad visual [7].

2.1.3. Impacto Social y Realidad Virtual

Los videojuegos han evolucionado de sistemas simples a entornos sociales complejos donde la influencia mutua entre jugadores es un factor determinante de la experiencia [8]. En el ámbito de la salud y el deporte, se ha comprobado que el uso de Realidad Virtual (RV) mejora significativamente las intenciones de participación deportiva en adolescentes en comparación con los videojuegos tradicionales [9]. La RV ofrece una percepción de requerimiento de habilidad que motiva más al usuario en disciplinas como el béisbol o el tenis [10].

2.1.4. Metodologías de Diseño y Creación

El aprendizaje del dibujo y el concept art en grados de videojuegos se ve potenciado por el uso de andamiajes conceptuales que vinculan la teoría con la producción visual profesional [11]. Estos flujos de trabajo imitan los procesos industriales donde el dibujo analítico precede a la creación del modelo 3D final [12]. Asimismo, la optimización para dispositivos móviles sigue siendo un reto técnico donde la gestión de la memoria y la reducción de llamadas de dibujo son prioridades constantes para los desarrolladores [13]. Finalmente, la programación visual se ha consolidado como un referente en la industria, permitiendo a artistas y diseñadores implementar mecánicas sin depender exclusivamente de programadores de sistemas, facilitando la creación de cinemáticas y entornos interactivos de alta calidad [2].

2.1.5. Cuadro Comparativo de trabajos relacionados

Tabla 1

Comparación de los trabajos relacionados

Eje Temático / Área de Estudio	Aportes y Hallazgos Principales	Aplicación y Relación con el Proyecto
1. Educación y Competencias de Programación	Fomenta habilidades de programación más eficazmente que la enseñanza tradicional. El scripting visual reduce la barrera de entrada al evitar sintaxis complejas. Facilita la transición de bloques visuales a código profesional.	Sustento Pedagógico: Fundamenta el uso de interfaces y entornos visuales para enseñar lógica y programación de forma intuitiva
2. Visualización y Gráficos en Tiempo Real	Define el pipeline industrial: escultura digital, retopología y texturizado Demuestra el uso de herramientas 3D y CAD para la recreación precisa de entornos virtuales.	Pipeline 3D: Brinda pautas técnicas para la creación de escenarios y modelos low poly estilizados y optimizados.
3. Impacto Social y Realidad Virtual	La inmersión en entornos virtuales incrementa la motivación y la participación activa. La percepción de requerimiento de habilidad mejora la retención y la respuesta del usuario.	Experiencia de Usuario (UX): Justifica la necesidad de crear un entorno interactivo atractivo que mantenga la motivación del estudiante.
4. Metodologías de Diseño y Optimización	Los andamiajes conceptuales (dibujo analítico y concept art) preceden al modelo 3D. La optimización técnica (gestión de memoria, reducción de draw calls) es vital para el rendimiento. Permite a los artistas implementar mecánicas y cinemáticas sin depender solo de código.	Metodología y Rendimiento: Aplica hojas de diseño previas (character sheets) y técnicas de optimización visual (batching e iluminación) en Unity.

Nota. Elaboración propia con base en Salvatierra y Palacios, Puerto y Meier, Ewell y Wilkins, Chao y López

2.2. Bases teóricas

2.2.1. Modelado

El modelado 3D surge de la necesidad de representar objetos en un espacio matemático tridimensional mediante geometría compuesta por vértices, aristas y caras [13]. Con la evolución de los motores de renderizado en tiempo real, la industria del videojuego adoptó el estilo Low Poly, el cual prioriza la síntesis geométrica manteniendo la legibilidad formal del objeto. Inicialmente adoptado por limitaciones de hardware, el Low Poly ha evolucionado hacia una corriente estética estilizada que permite optimizar el rendimiento y enfocar la atención del usuario en los elementos interactivos clave.

- **Modelado 3D y Estructura de Malla**

El modelado 3D es el proceso fundamental mediante el cual se genera la representación matemática de cualquier superficie u objeto tridimensional en un entorno digital. La estructura base que compone a la gran mayoría de estos objetos se conoce como malla (mesh), la cual se encuentra delimitada por elementos geométricos esenciales: vértices, aristas y caras [14]. La correcta disposición espacial de estos componentes dentro del software de modelado (como Blender) determina no solo la silueta y el volumen de la figura, sino también su comportamiento estructural al ser procesada en motores de tiempo real [2].

Para la creación de personajes y activos orgánicos complejos, se suele emplear la escultura digital, una técnica avanzada de modelado que permite trabajar con mallas de muy alta densidad poligonal. Este procedimiento actúa como una etapa inicial indispensable en la producción profesional [14], ya que facilita el esculpido de volúmenes detallados, pliegues y arrugas sin las restricciones inmediatas del flujo topológico tradicional. Posterior a esta fase,

la malla de alta densidad es sometida a procesos de simplificación geométrica para generar una versión funcional compatible con motores interactivos [2].

Una vez construidas, estas mallas se transfieren al motor gráfico (como Unity), donde se organizan mediante contenedores denominados GameObjects. Los GameObjects integran los componentes de renderizado de malla (Mesh Filter y Mesh Renderer) junto con scripts lógicos y colisionadores, transformando la geometría estática creada en el software de modelado en un elemento interactivo dotado de propiedades espaciales y comportamientos dentro de la escena.

- **Low Poly y Optimización**

La técnica Low Poly se define por el uso deliberado de una densidad poligonal reducida para representar objetos tridimensionales, sintetizando las formas complejas sin perder su legibilidad visual ni sus rasgos identitarios [13]. Este enfoque artístico y técnico resulta indispensable en aplicaciones que requieren renderizado en tiempo real, garantizando que el entorno mantenga tasas de fotogramas estables y una navegación fluida sin sobrecargar la capacidad de procesamiento gráfico del dispositivo [2].

En el ámbito de la optimización en motores de videojuegos, la reducción de polígonos disminuye directamente la carga computacional en la tarjeta gráfica y la memoria RAM [2]. Adicionalmente, la eficiencia del sistema se refuerza mediante técnicas como el batching o agrupamiento de llamadas de dibujo (draw calls), las cuales permiten que el motor procese e instancie múltiples objetos idénticos o que comparten material en un solo paquete de procesamiento, reduciendo drásticamente las peticiones enviadas por la CPU a la GPU [15].

2.2.2. Texturizado

El texturizado abarca el conjunto de procesos técnicos y artísticos destinados a conferir color, relieve y propiedades lumínicas a la superficie de los modelos tridimensionales [16]. Esta etapa inicia con el mapeado UV, una proyección indispensable que despliega la malla 3D sobre un plano bidimensional para garantizar la correcta aplicación de mapas de textura [17]. Gracias al texturizado, se logran representar detalles complejos y características físicas de los materiales sin aumentar la densidad poligonal de la escena, garantizando un equilibrio óptimo entre la fidelidad visual y el rendimiento técnico del sistema.

- **UV Mapping y Texturizado**

El mapeado UV (UV Mapping) es el proceso crítico que consiste en proyectar una superficie bidimensional sobre una malla tridimensional para permitir la aplicación precisa de texturas e imagen de mapa de bits. Mediante este procedimiento, la superficie del objeto 3D se "despliega" en un plano de coordenadas U y V, definiendo cómo se alinearán las texturas sobre cada cara poligonal y evitando deformaciones o estiramientos indeseados en el aspecto final del modelo [18].

En los flujos de trabajo de este proyecto, el texturizado se fundamenta en un enfoque estilizado y coherente con la dirección de arte basada en el uso de colores planos y paletas cromáticas controladas, el sistema de materiales busca potenciar la legibilidad visual y la identidad gráfica tanto del personaje como de los escenarios de la guía interactiva, asegurando una lectura limpia y uniforme bajo cualquier condición de iluminación ambiental.

Asimismo, el texturizado actúa como una herramienta clave de optimización al no depender de mapas de relieve complejos o simulaciones de alta densidad física. En su lugar, el diseño gráfico se apoya directamente en la geometría limpia de los modelos low-poly,

complementada con el uso estratégico de máscaras emisivas y texturas simplificadas que aportan contraste, profundidad visual y dinamismo a elementos clave de la escena (como plataformas, interfaces o detalles energéticos) sin comprometer el rendimiento técnico en el motor gráfico.

- **Materiales y Shader**

Los materiales son activos encargados de definir la apariencia visual de la superficie de un objeto 3D dentro del motor de renderizado, actuando como el enlace entre los mapas de textura y las propiedades cromáticas de la malla. A través de un material, se especifican parámetros como el color base, el grado de brillo, el contraste y la transparencia, determinando cómo se percibirá visualmente cada activo dentro de la escena interactiva [20].

Por su parte, los shaders son programas o scripts especializados que contienen los cálculos matemáticos necesarios para determinar la iluminación, sombras y el color final de cada píxel en la pantalla. Mientras el material actúa como el contenedor de parámetros, el shader procesa en tiempo real las fórmulas de atenuación de luz y reflejos, permitiendo crear desde efectos fotorrealistas hasta acabados estilizados con estéticas de neón o cel-shading acordes a la dirección de arte del proyecto [21].

2.2.3. Rigging

El rigging es el proceso de articulación técnica que transforma una malla 3D estática en un activo tridimensional dinámico capaz de deformarse y moverse de forma orgánica [23]. Consiste en la creación de una armadura u osamenta interna (armature) y en el ajuste de influencias mediante el pintado de pesos (weight paint), determinando con precisión cómo se deforma la geometría al reaccionar ante cada hueso [24]. Esta estructura resulta imprescindible para la posterior creación de animaciones y su posterior gestión mediante

controladores en el motor gráfico, permitiendo transmitir comportamientos, estados y mecánicas interactivas dentro del entorno educativo.

- **Armature**

El Armature o armadura es la estructura ósea virtual compuesta por una jerarquía de huesos (bones) que se integra dentro de una malla tridimensional para permitir su movilidad, proceso conocido técnicamente como rigging [25]. Este esqueleto actúa como el sistema articular del personaje u objeto dinámico, estableciendo los puntos pivote, las cadenas de jerarquía y las restricciones mecánicas que dictaminarán cómo se trasladará y rotará la geometría dentro del espacio tridimensional.

Durante la fase de articulación, el diseñador debe disponer y alinear estratégicamente cada hueso dentro del volumen de la malla low poly, replicando la estructura anatómica o funcional requerida por el activo. Una correcta configuración de la armature en el software de modelado (como Blender) es fundamental para garantizar que las extremidades y articulaciones del avatar respondan con fidelidad a las cinemáticas planteadas, sentando las bases para una deformación natural de la superficie [25].

Finalmente, el armature sirve como el puente técnico que vincula la geometría pura del modelo con los sistemas de animación en tiempo real. Al exportar el activo hacia el motor gráfico, la estructura de huesos es reconocida por el sistema de rigging del motor (como el sistema Humanoid o Generic en Unity), permitiendo reutilizar animaciones, aplicar dinámicas físicas y preparar al personaje para responder de forma interactiva ante las acciones del usuario [26].

- **Weight Paint**

El Weight Paint (pintado de pesos) es la técnica mediante la cual se define y ajusta la influencia específica que ejerce cada hueso de la armature sobre los vértices de la malla

tridimensional [27]. Este procedimiento asigna a cada vértice valores numéricos de peso que oscilan entre 0 y 1, determinando en qué porcentaje responderá dicho vértice al movimiento de un hueso determinado cuando el modelo sea articulado.

En el entorno de trabajo, esta asignación de influencias se representa de forma visual mediante un gradiente cromático de color sobre el modelo 3D (variando comúnmente desde el azul para cero influencias hasta el rojo para influencia máxima) [27]. El equilibrio en el pintado de pesos en zonas de flexión (como codos, rodillas y hombros) es imprescindible para evitar deformaciones antinaturales, pliegues defectuosos, colapsos de volumen o rupturas en la geometría durante la ejecución de los movimientos [28].

En el marco del proyecto, un correcto desarrollo del weight paint garantiza que los personajes y avatares mantengan una apariencia estética coherente y fluida mientras interactúan dentro de la guía interactiva. La precisión en el pulido de estas influencias, lograda a través del uso técnico de grupos de vértices [29], asegura que la geometría de baja densidad se deforme orgánicamente, protegiendo la legibilidad del diseño visual sin añadir carga poligonal extra a la escena.

- **Animación 3D**

La animación 3D comprende el proceso de dotar de vida, dinamismo y movimiento a la malla articulada mediante la creación de fotogramas clave (keyframes) [30] y ciclos de acción estandarizados [31]. En el desarrollo de activos para entornos interactivos, se generan clips de animación específicos (tales como estados de reposo, caminata, gestos o interacciones con objetos) que representan visualmente la conducta y estado del personaje u objeto dentro del entorno virtual.

Una vez elaboradas las secuencias de movimiento en Blender, estas se integran en Unity [32] y se gestionan a través de controladores de animación (Animator Controllers)

[33]. Estos controladores emplean máquinas de estados (Animation State Machines) para definir la lógica de transición entre un movimiento y otro en función de las variables de entrada del sistema, garantizando cambios de estado suaves, coherentes y sin interrupciones bruscas en el flujo visual.

En el contexto de entornos interactivos educativos, la animación 3D trasciende la función estética para convertirse en un canal de retroalimentación comunicativa esencial. Las respuestas animadas claras facilitan la interpretación de las mecánicas por parte del estudiante, permitiéndole comprender inmediatamente los cambios de estado y las relaciones lógicas del sistema, lo que reduce la ambigüedad y favorece una navegación intuitiva por la guía de aprendizaje.

2.3. Bases metodológicas

En el desarrollo de proyectos interactivos y la producción de activos tridimensionales, la selección de la metodología de trabajo adecuada es un factor determinante para estructurar el flujo de producción, organizar los tiempos de entrega y garantizar que el producto final responda tanto a los requerimientos técnicos como a las necesidades del usuario. A continuación, se examinan tres enfoques metodológicos clave: un marco ágil e incremental (Scrum), un modelo secuencial tradicional (Waterfall) y una metodología enfocada en la experiencia de usuario (Design Thinking).

2.3.1. Scrum

Scrum es un marco de trabajo ágil e incremental diseñado para gestionar proyectos complejos a través de ciclos de desarrollo iterativos de corta duración denominados sprints (por lo general de una a cuatro semanas) [34]. Esta metodología promueve la colaboración continua, la autogestión y la capacidad de adaptarse rápidamente a cambios en las

especificaciones del sistema. Bajo esta premisa, en el contexto del desarrollo gráfico y 3D, Scrum permite descomponer la creación del entorno interactivo en bloques funcionales y entregables (como la creación del personaje principal en un sprint, el diseño del escenario en otro y la implementación de la interfaz en el siguiente), posibilitando evaluar e iterar constantemente la calidad visual y el rendimiento técnico sin tener que esperar a las etapas finales de la producción [35].

Características principales:

- Estructura iterativa e incremental: Organiza el trabajo en bloques temporales fijos (sprints) donde se generan entregables funcionales al final de cada ciclo [36].
- Roles definidos: Se fundamenta en tres roles clave: Product Owner (define las prioridades del proyecto), Scrum Master (facilita el proceso) y el Development Team (ejecuta el trabajo de diseño y programación).
- Eventos estandarizados: Incluye ceremonias sistemáticas como la planificación del sprint (Sprint Planning), reuniones diarias de seguimiento (Daily Standup), revisión de entregables (Sprint Review) y retrospectivas de mejora continua [37].
- Gestión dinámica de requerimientos: Utiliza una lista priorizada de tareas (Product Backlog) que permite reajustar los requerimientos visuales o técnicos según las necesidades emergentes.
- Flexibilidad y adaptación: Facilita la incorporación de cambios de forma rápida a partir de la retroalimentación constante durante el proceso.

2.3.2. Waterfall (Modelo en Cascada)

El modelo en cascada o Waterfall es un enfoque de desarrollo tradicional, lineal y secuencial en el cual el proyecto avanza de forma estricta a través de fases jerárquicas

consecutivas: análisis, diseño, desarrollo, validación y ajustes finales [38]. Cada fase debe completarse por completo y ser aprobada antes de dar inicio a la siguiente, lo que exige una planificación detallada y una definición rigurosa de los requerimientos desde el inicio del proyecto [39]. Siguiendo este enfoque, en el contexto del desarrollo gráfico y 3D, esta metodología resulta especialmente sólida en el ámbito del diseño y la producción de arte 3D, ya que permite "congelar" la dirección de arte, el estilo visual (low poly), la paleta de colores y las restricciones poligonales en la fase inicial, evitando reestructuraciones costosas o cambios imprevistos durante la fase de producción masiva e integración en el motor gráfico.

Características principales:

- Progresión lineal y secuencial: El flujo de trabajo sigue una secuencia unidireccional y ordenada de etapas delimitadas [38].
- Documentación exhaustiva: Requiere la definición y congelamiento previo de las especificaciones técnicas y artísticas desde la fase inicial de análisis [39].
- Dependencia estricta entre fases: Los resultados de una etapa previa se convierten en los insumos indispensables para la ejecución de la fase siguiente.
- Alto nivel de control y predictibilidad: Facilita la estimación de cronogramas, la verificación de hitos y el cumplimiento de entregables académicos formales.
- Baja permeabilidad al cambio tardío: Un cambio estructural en etapas avanzadas (como en la fase de integración) suele requerir reiniciar fases completas del proceso.

2.3.3. Design Thinking

Design Thinking es una metodología de resolución de problemas centrada en el ser humano, enfocada en comprender las necesidades profundas de los usuarios para generar soluciones innovadoras y efectivas. Se estructura en cinco fases principales no

necesariamente lineales: empatizar, definir, idear, prototipar y evaluar [40]. En el desarrollo de la guía interactiva educativa, esta metodología aporta un marco fundamental para garantizar que el sistema visual, la jerarquía de los elementos gráficos y los modelos 3D no respondan únicamente a criterios estéticos, sino a las capacidades perceptivas y de aprendizaje del estudiante. Bajo este enfoque estratégico, al centrar el diseño en la usabilidad y en la reducción de la carga cognitiva, Design Thinking asegura que los elementos tridimensionales faciliten directamente la claridad visual y la navegación coherente dentro del entorno interactivo [41].

Características principales:

- Enfoque centrado en el usuario (UX): Prioriza la empatía con las necesidades, dificultades y comportamiento del usuario final.
- Estructura en cinco fases interconectadas: Explora el problema a través de las etapas de Empatizar, Definir, Idear, Prototipar y Evaluar [40].
- Fomento de la innovación visual: Estimula la generación de múltiples alternativas conceptuales a través de la ideación y el arte conceptual.
- Prototipado rápido e iterativo: Promueve la elaboración temprana de bocetos, moodboards y maquetas 3D para validar la dirección visual sin consumir grandes recursos técnicos [42].
- Validación basada en la experiencia: Evalúa la efectividad del diseño mediante pruebas directas con usuarios, midiendo aspectos clave como la perceptibilidad y la usabilidad.

Tabla 2*Comparación de las metodologías Fuente: Autor Propio*

Metodología	Principios Clave	Aplicabilidad al Proyecto de Diseño	Enfoque Principal	Gestión del Cambio y Entregables
Scrum	Desarrollo iterativo e incremental organizado en sprints. Roles definidos (Product Owner, Scrum Master, Dev Team). Eventos estandarizados (Dailies, Planning, Review, Retrospective). Backlog dinámico y priorizado.	Moderada-Alta: Facilita la organización del tiempo y la división por bloques de arte 3D (personaje, props, escenario) en ciclos temporales fijos.	Gestión ágil de proyectos: Enfocado en la velocidad de adaptación, la entrega continua de valor y la autogestión.	Alta flexibilidad: Permite redefinir tareas o ajustar requerimientos visuales al inicio de cada sprint. Entregables: Módulos visuales o mecánicas funcionales e incrementales al final de cada ciclo.
Waterfall (Cascada)	Congelamiento de requerimientos iniciales. Documentación técnica exhaustiva previa.	Muy Alta: Ideal para la producción 3D, ya que exige definir el estilo visual, paleta cromática y límites poligonales antes de proceder al modelado, rigging e integración en Unity.	Planificación y control del proceso: Enfocado en la previsibilidad, el rigor metodológico, el cumplimiento estricto del cronograma y el control de calidad.	Baja flexibilidad: Los cambios en fases avanzadas resultan costosos e implican reestructurar etapas previas. Entregables: Un producto final integrado y validado tras completar todas las etapas.
Design Thinking	Prototipado e iteración constante basada en pruebas.	Muy Alta: Indispensable para garantizar que el sistema	Experiencia de usuario (UX): Enfocado en resolver	Adaptabilidad guiada: Modifica la propuesta visual

	visual, la jerarquía de la interfaz y la estética low poly reduzcan la carga cognitiva y mejoren la usabilidad (evaluada mediante escala SUS).	problemas perceptivos y pedagógicos comprendiendo las capacidades visuales del estudiante.	según los resultados obtenidos en las pruebas con usuarios. Entregables: Moodboards, hojas de personaje, maquetas de interfaz y prototipos evaluables
--	--	--	---

Nota. Elaboracion propia con base en Martins, Martiño y Alfonso, Cifuentes, Laoyan, SafetyCulture, El Ecosistema Startup, Design Thinking España

2.4. Bases tecnológicas

En esta sección se describen las principales herramientas utilizadas para la construcción de modelos 3D y el proceso de texturizado. En primer lugar, se presentan las herramientas de modelado 3D seleccionadas a partir de una revisión basada en su popularidad, adopción y aceptación dentro de la comunidad de desarrollo de videojuegos y diseño 3D.

2.4.1. Blender

Blender es un programa de creación tridimensional integrada, de código abierto y completamente gratuita, ampliamente reconocida en la industria del desarrollo de videojuegos por su capacidad para cubrir la totalidad del pipeline de producción 3D. Su entorno unificado integra herramientas para modelado poligonal, escultura digital, desempaquetado UV, texturizado, creación de armaduras (rigging), pintura de pesos (weight painting) y animación por fotogramas clave (keyframes) [43]. Gracias a su arquitectura ligera y constante actualización, Blender permite desarrollar activos tridimensionales low poly con un flujo de trabajo ágil, facilitando la exportación directa de mallas articuladas y secuencias animadas hacia motores en tiempo real como Unity mediante formatos estándares como FBX y OBJ. Seguidamente se describen algunas funciones que ofrece esta herramienta para el modelado 3D.

- De acuerdo con el manual técnico, los modificadores de geometría (Mirror, Subdivision Surface, Decimate): permiten la construcción simétrica y la optimización no destructiva de mallas poligonales [44].

- El conjunto de herramientas de edición poligonal (Extrude, Loop Cut, Bevel): es indispensable para la definición limpia de la topología low poly de los avatares y elementos del escenario [43].
- Editor de desempaquetado UV (Unwrap, Smart UV Project): Facilita la proyección y ordenamiento eficiente de coordenadas U y V para evitar distorsiones en las texturas [43].
- Sistema de Armatures y Restricciones (FK/IK Constraints): Permite la construcción de esqueletos articulados jerárquicos y la configuración de cinemática inversa para codos y rodillas [43].
- Modo Weight Paint: Ofrece pinceles graduados de influencia cromática para calibrar la flexión suave de la malla sobre cada hueso [43].
- Dope Sheet y Graph Editor: Permiten crear, ajustar y pulir las curvas de interpolación de los clips de animación antes de su integración en el motor gráfico [43].
- Modelo de Licencia: Software Libre y de Código Abierto.

2.4.2. Autodesk Maya

Autodesk Maya es un software propietario de modelado, simulación y animación 3D considerado un estándar consolidado en la industria audiovisual y de videojuegos AAA [45]. Destaca principalmente por su robusta arquitectura orientada a la caracterización técnica, articulación (rigging) y animación de personajes complejos, respaldada por un motor de evaluación en paralelo que optimiza el uso de CPU y GPU. Tal como se detalla en las guías de usuario, en la producción de activos para la guía interactiva, Maya aporta herramientas de precisión industrial para el control de deformadores, el retargeting de animaciones y la organización rigurosa de esqueletos (skeletons), asegurando la máxima compatibilidad

técnica al exportar archivos hacia Unity [46]. Seguidamente se describen algunas funciones que ofrece esta herramienta para el modelado 3D.

- Toolset de Modelado Poligonal: Facilita la edición fluida de caras y bordes reduciendo la necesidad de limpieza geométrica manual [47].
- UV Editing Toolkit: Editor avanzado de coordenadas UV con algoritmos de empaquetado optimizado para maximizar la resolución del espacio de textura [48].
- HumanIK y Retargeting: Sistema especializado para adaptar y transferir animaciones preexistentes entre diferentes jerarquías óseas [46].
- Time Editor y Graph Editor: Entorno de edición de animación no lineal que permite combinar y editar secuencias clave de movimiento como si fuesen clips de video [49].
- Modelo de Licencia: Software Propietario de Pago.

2.4.3. Maxon Zbrush

Maxon ZBrush es la herramienta referente en la industria para la escultura digital 3D y el modelado orgánico de alta resolución [50]. Fundamentado en la tecnología Pixol, ZBrush permite esculpir y manipular mallas compuestas por millones de polígonos en tiempo real con una sensibilidad y respuesta similar a la del modelado en arcilla tradicional [51]. En la fase conceptual del proyecto, ZBrush resulta clave para dar forma a los detalles orgánicos de los personajes sin restricciones iniciales de topología; posteriormente, sus algoritmos integrados permiten convertir esas esculturas densas en mallas low poly totalmente funcionales para entornos interactivos. Seguidamente se describen algunas funciones que ofrece esta herramienta para el modelado 3D.

- DynaMesh: De acuerdo con las guías de usuario, este es un modo de generación geométrica que redistribuye los polígonos de forma uniforme durante la escultura libre, eliminando estiramientos en la malla [52].
- ZRemesher: Algoritmo de retopología automática que procesa esculturas complejas para generar mallas de baja densidad con un flujo de aristas optimizado [53].
- Polypaint: Permite pintar color directamente sobre la superficie poligonal del modelo 3D antes de generar la disposición final de coordenadas UV [54].
- Organización por SubTools y Booleans: El sistema facilita la división del personaje en partes compuestas (cabeza, ropa, accesorios) y el corte de geometrías mediante operaciones booleanas en tiempo real [55], [56].
- Modelo de Licencia: Software Propietario de Pago.

A continuación, se describen las principales herramientas empleadas en el proceso de texturizado de modelos 3D. Para su selección se consideró su amplia utilización en la industria de los videojuegos, la calidad de los resultados que ofrecen y la disponibilidad de funciones orientadas a la creación de materiales realistas y estilizados.

2.4.4. Krita

Krita es una aplicación libre y de código abierto orientada a la ilustración digital, la pintura de arte conceptual y la creación de texturas estilizadas 2D [57]. Diseñada por y para artistas, la plataforma cuenta con una interfaz personalizable, motores de pincel avanzados y sistemas de estabilización de trazo que facilitan el desarrollo de bocetos ortográficos (character sheets), concept art y mapas de color pintados a mano (hand-painted textures) [57]. Su ligereza la convierte en una opción idónea para la etapa previa de diseño estético de la

guía interactiva. Seguidamente se describen algunas funciones que ofrece esta herramienta para el texturizado.

- **Wrap-around Mode (Modo Envolvente):** Esta herramienta repite automáticamente el lienzo en los ejes X e Y para pintar y verificar en tiempo real la continuidad de texturas patronales sin costuras (seamless textures) [58].
- **Varios Pinceles:** El programa dispone varios estilos de pinceles para simular acabados artísticos [59].
- **Estabilizadores de Trazo:** Se integran modos de suavizado que eliminan las imprecisiones del trazo manual, útiles para definir siluetas e íconos de interfaz [60].
- **Modelo de Licencia:** Software Libre y de Código Abierto.

2.4.5. Adobe Photoshop

Adobe Photoshop es el software estándar en edición ráster, composición digital y retoque de imágenes. En la producción de videojuegos y entornos 3D, Photoshop es una pieza clave para la elaboración de arte conceptual, el diseño de elementos para la interfaz de usuario (UI) y el procesamiento de mapas de materiales bajo el estándar PBR [61] para proyectos de mayor peso gráfico. Su sistema de trabajo basado en capas, máscaras y objetos inteligentes garantiza una edición no destructiva, facilitando el empaquetado de canales de textura e igualando la coherencia cromática de los recursos gráficos antes de importarse a Unity [62].

- **Sistema de Capas y Máscaras de Capa:** Permite aislar y ajustar de forma no destructiva componentes individuales de una textura o interfaz [63].
- **Gestión de Canales de Color (RGBA):** De acuerdo con la documentación oficial. Esta función facilita la combinación de mapas en canales individuales de una sola imagen (por ejemplo, empaquetar rugosidad, oclusión ambiental y metalicidad) [64].

- Pinceles personalizados y texturizado ráster: Permiten agregar detalles de superficie como polvo, desgaste o para texturas realistas [65].
- Herramientas de Selección y Ajuste Tonal: Estas utilidades permiten homogeneizar la paleta estético-visual de todos los activos del proyecto [62].
- Modelo de Licencia: Software Propietario de Pago.

2.4.6. Cuadro comparativo de Herramientas

Tabla 3
Comparación de Herramientas Tecnológicas

Herramienta	Ventajas Principales	Desventajas	Pertinencia en el Proyecto
Blender	Software libre y gratuito (GPL). Pipeline todo-en-uno (modelado, UV, rigging, animación).	Curva de aprendizaje inicial pronunciada en animación y weight paint. Menos estandarizado en estudios AAA tradicionales.	Eje central 3D: Creación completa de modelos low poly, articulación (rigging), pintado de pesos y ciclos de animación.
Autodesk Maya	Estándar de la industria en animación y rigging industrial. Herramientas de deformación de precisión y retargeting avanzado.	Licencia comercial de pago muy costosa. Software pesado que requiere más recursos de hardware.	Referente técnico: Consulta de estándares de articulación ósea y deformación topológica
Maxon ZBrush	Liderazgo en escultura digital de alta densidad y detalle orgánico. Algoritmos avanzados como ZRemesher y DynaMesh..	Costo de suscripción o licencia perpetua elevado. Interfaz no convencional y compleja para flujos puramente 2D/Low Poly.	Fase conceptual: Modelado libre inicial de personajes orgánicos antes de su retopología.

Krita	Software libre y gratuito (GPL). Modo Wrap-around para texturas continuas sin costuras. Interfaz ligera y optimizada para pintura e ilustración.	Menor cantidad de herramientas avanzadas para edición fotográfica ráster. Exportación sencilla para imágenes planas.	Eje central 2D: Diseño de arte conceptual (character sheets), bocetos de interfaz (UI) y texturizado estilo hand-painted.
--------------	--	--	---

Adobe Photoshop	Estándar en composición digital y manipulación de canales RGBA. Extenso ecosistema de pinceles y herramientas de ajuste tonal.	Modelo de suscripción mensual obligatoria (Creative Cloud). Mayor consumo de memoria RAM al trabajar con archivos pesados.	Procesamiento visual: Edición final de texturas PBR y retoque de elementos de interfaz gráfica.
------------------------	--	--	---

Nota. Blender, Autodesk Maya, Maxon Zbrush, Krita y Adobe Photoshop

2.5. Arquitecturas, modelos y estándares

En el desarrollo de entornos interactivos tridimensionales, la definición formal de arquitecturas de activos, modelos de producción y estándares técnicos constituye la base metodológica para garantizar la escalabilidad, la cohesión visual y el rendimiento óptimo del software. Estos lineamientos estructuran la labor del artista 3D y facilitan una transición fluida hacia motores de tiempo real como Unity, asegurando que cada componente cumpla con las especificaciones técnicas necesarias para evitar problemas de optimización. Bajo esta premisa, la adopción de flujos de trabajo orientados al rendimiento permite alcanzar una escalabilidad técnica superior [66], fundamentada en la definición rigurosa de formatos y ajustes de importación desde la fase de diseño [67].

2.5.1. Arquitectura de Personaje

La arquitectura de personajes organiza el desarrollo integral de los avatares para el personaje jugable del entorno, segmentándolo en partes de forma modular. Su propósito es separar lógicamente los componentes anatómicos, textiles y esqueléticos para facilitar la personalización, el texturizado y la vinculación con los sistemas de animación en el motor gráfico.

- **Modelado base y topología:** Construcción de la malla poligonal optimizada (low poly) asegurando bucles de aristas limpios en las zonas de articulación para una correcta deformación.
- **Desglose modular y vestimenta:** Separación de los elementos de ropa, accesorios y rostro en mallas independientes para optimizar la jerarquía de materiales y el intercambio de piezas.
- **Rigging e integración:** Incorporación del esqueleto virtual (armature), asignación de pesos de vértices (skinning) y pruebas de deformación previas a la exportación en formato FBX.

Este enfoque asegura una topología eficiente capaz de soportar deformaciones complejas [68], garantizando transformaciones predecibles durante la integración de los activos en el motor gráfico [69].

2.5.2. Arquitectura de Escenarios

La arquitectura de escenarios establece la segmentación estructural del entorno virtual para optimizar tanto el diseño de niveles como el rendimiento de renderizado. Al dividir el espacio en categorías funcionales, se facilita el uso de piezas modulares y se optimiza la carga de elementos estáticos e interactivos en memoria.

- **Bloqueo espacial y terreno:** Definición de la topografía base y las proporciones del escenario mediante volúmenes primitivos (blockout) para validar la escala y la jugabilidad.
- **Ensamblaje modular:** Distribución ordenada de estructuras arquitectónicas, elementos de vegetación y utilería (props) empleando sistemas de ajuste a rejilla (Grid Snapping).
- **Configuración de colisiones e iluminación:** Asignación de mallas de colisión simplificadas y preparación de los activos para el horneado de luz estática (lightmapping).

En este sentido, la segmentación modular del escenario facilita la colaboración técnica entre disciplinas, mientras que el uso de iluminación precalculada optimiza significativamente el procesamiento por cuadro en tiempo real [67].

2.5.3. Modelo basado en referencias

El modelo basado en referencias estipula que ningún recurso tridimensional debe crearse de forma intuitiva o arbitraria, sino fundamentado en fuentes documentables y visuales precisas. Este enfoque metodológico garantiza la fidelidad estética, la coherencia proporcional y la verosimilitud de los elementos dentro del contexto interactivo.

- **Recopilación y análisis visual:** Búsqueda de fotografías, planos arquitectónicos, bocetos conceptuales y láminas de proporciones anatómicas para establecer las directrices estéticas.
- **Proyección de planos de referencia:** Importación de las imágenes de apoyo en las vistas ortogonales del software de modelado 3D para alinear las siluetas con precisión milimétrica.

- **Validación de proporciones:** Comparación constante del modelo tridimensional con las referencias recopiladas durante las etapas de bloqueo y refinamiento de volúmenes.

En este sentido, la combinación de fotografía y bocetos conceptuales permite fundamentar el diseño antes de la escultura digital [70], asegurando que el flujo de bordes siga la anatomía natural de la estructura [68].

2.5.4. Modelo de validación técnica

El modelo de validación técnica comprende el protocolo de control de calidad aplicado a cada activo digital antes de su inserción definitiva en el motor de desarrollo. Su finalidad es certificar que los modelos cumplan estrictamente con los umbrales de rendimiento, topología y compatibilidad establecidos para evitar caídas de fotogramas.

- **Auditoría geométrica y de escalas:** Verificación del recuento de triángulos, eliminación de vértices huérfanos y comprobación con respecto a la escala métrica el motor de Unity, equivalente a 1 unidad = 1 metro.
- **Revisión de materiales y UVs:** Control de la correcta proyección de coordenadas UV dentro del atlas de color y verificación de que se emplee el número mínimo de materiales únicos.
- **Prueba de rendimiento en Unity:** Importación preliminar del activo en la escena de pruebas para medir su impacto en las llamadas de renderizado (SetPass Calls) y lotes (Batches).

Bajo este protocolo, es fundamental auditar métricas como el recuento de triángulos y las llamadas SetPass para garantizar la estabilidad del software [67]. Asimismo, la

configuración precisa de los factores de escala y ejes en la exportación FBX previene errores de transformación en la fase de integración [71].

2.5.5. Estandar de escala y unidades

El estándar de escala y unidades rige las directrices métricas universales que deben respetar todos los elementos modelados en el proyecto. Mantener un sistema de medidas homogéneo evita distorsiones físicas, errores de colisión y problemas de proporción al interactuar con los sistemas de físicas del motor gráfico.

- **Definición del sistema métrico:** Configuración inicial del software de modelado (Blender) utilizando el sistema métrico decimal, donde 1 unidad de Blender equivale estrictamente a 1 metro en Unity.
- **Congelación de transformaciones:** Aplicación obligatoria de la operación de reinicio (Apply Scale/Rotation) en los objetos 3D para asegurar que sus valores de escala locales sean exactamente 1.0.
- **Verificación de colisionadores:** Comprobación de que las dimensiones de los activos coincidan de manera proporcional con los componentes de colisión generados en el entorno de desarrollo.

Para garantizar la interoperabilidad técnica, es fundamental que el factor de escala se mantenga en 1.0 mediante el congelamiento de transformaciones, evitando así desajustes de rotación y escala durante la importación [71], [69].

2.5.6. Estandar de animación

El estándar de animación establece las especificaciones técnicas y los parámetros de formato que deben cumplir todos los movimientos exportados para los personajes y objetos

interactivos. Esto garantiza una transición fluida hacia los controladores de animación (Animator Controllers) y previene desincronizaciones en la ejecución del software.

- **Definición de fotogramas y ciclos:** Estandarización de la frecuencia de muestreo a 30 o 60 fotogramas por segundo, asegurando que las animaciones de ciclo continuo posean un punto de enlace perfecto.
- **Nomenclatura y etiquetado:** Aplicación de un convenio de nombres estricto que identifique claramente la acción, el estado y el orden del ciclo (ej. ANIM_Personaje_Caminar).
- **Exportación y configuración de clips:** Importación del archivo con la jerarquía esquelética limpia, configurando la opción de bucle (Loop Time) directamente en Unity.

La implementación de convenciones de nomenclatura rigurosas facilita la organización de los activos, mientras que el horneado de las animaciones antes de la exportación FBX asegura una reproducción predecible y sincronizada en el motor de juego [69].

2.6. Herramientas de desarrollo

La selección de las herramientas con las que se realizaron este proyecto no responde a criterios de popularidad comercial, sino a una evaluación técnica orientada a la eficiencia y la interoperabilidad. Esta elección se fundamenta en su alta compatibilidad de exportación y su accesibilidad, elementos clave para mantener la integridad del flujo de trabajo y la calidad de los activos gráficos.

Varias herramientas fueron descartadas durante este proceso al no garantizar la fluidez necesaria en la cadena de producción low poly o al presentar barreras de integración

que complicaban la gestión colaborativa del proyecto. A continuación, se presenta un cuadro comparativo que sintetiza las capacidades técnicas de las herramientas analizadas, justificando por qué la combinación seleccionada representa la solución más equilibrada.

La adopción y apego a Blender y Krita como tecnologías principales del proyecto se justifica por la optimización de recursos, la sostenibilidad académica y la eficiencia en el pipeline de desarrollo técnico-artístico. Al ser herramientas de código abierto y distribución gratuita, eliminan la barrera económica de las licencias comerciales, garantizando la viabilidad y replicabilidad del trabajo en el contexto universitario. En la dimensión técnica, Blender permite centralizar todo el flujo de trabajo 3D (modelado low poly, mapeado UV, rigging, weight paint y animación) en una sola plataforma, facilitando una integración nativa y sin pérdida de datos con el motor Unity 6. Por su parte, Krita aporta un entorno especializado para el diseño de concept art, recursos de interfaz (UI) y texturas estilizadas, garantizando un rendimiento ligero que agiliza la producción de activos y respalda directamente la coherencia visual exigida por la propuesta pedagógica.

2.7. Marco conceptual

Este apartado describe los pilares técnicos fundamentales para el desarrollo de activos tridimensionales y su implementación en motores de juego.

- **Modelado 3D y Estructura de Malla** El modelado 3D es el proceso de crear una representación matemática de cualquier superficie de un objeto. En entornos como Blender, una malla (mesh) constituye la base de la mayoría de los objetos y se compone de vértices, aristas y caras que definen su forma en un espacio tridimensional [1]. Para el desarrollo de personajes, se suele emplear la escultura digital, una técnica que permite trabajar con una alta densidad poligonal para capturar

detalles orgánicos complejos, actuando como una pieza clave en la producción profesional [2]. Los bloques fundamentales en motores como Unity son los GameObjects, los cuales actúan como contenedores para los componentes de malla y otros elementos lógicos [3].

- **Low Poly y Optimización** La técnica Low Poly se caracteriza por el uso de un número reducido de polígonos para representar objetos, priorizando el rendimiento sin sacrificar la legibilidad visual [72]. Este enfoque es esencial para el renderizado en tiempo real, especialmente en dispositivos con recursos limitados como los móviles, donde la optimización geométrica dictamina la fluidez del juego [73]. La reducción de la carga computacional se logra también mediante la técnica de batching o agrupamiento de llamadas de dibujo (draw calls), lo que permite al motor procesar múltiples objetos de manera eficiente [74].
- **UV Mapping y Texturizado** El mapeado UV es el proceso crítico de proyectar una malla 2D sobre la superficie de un objeto 3D para permitir la aplicación precisa de texturas [76]. En el flujo de trabajo para proyectos de mayor escala, se utiliza el Physically Based Rendering (PBR), un sistema que busca simular la interacción de la luz con los materiales de manera realista basándose en propiedades físicas [2]. El texturizado permite añadir detalles visuales profundos, sin embargo, para la realización de este proyecto se van a utilizar texturas de color base sin detalles que sobrecarguen al motor gráfico.
- **Armature, Rigging y Animación 3D** Para que un modelo 3D pueda moverse, requiere de una Armature o armadura, que consiste en una estructura de huesos que deforma la malla mediante un proceso conocido como rigging [78]. En Unity, el sistema de

animación se gestiona a través de controladores de animación (Animator Controllers), que permiten organizar clips de animación y definir transiciones lógicas entre diferentes estados del personaje [79].

- **Materiales y Shaders** Los materiales definen cómo se ve la superficie de un objeto vinculando texturas y propiedades de color, mientras que los shaders son scripts que contienen los cálculos matemáticos necesarios para determinar la iluminación y el color de cada píxel [80].
- **Carga cognitiva** se refiere a la cantidad de esfuerzo mental que una persona utiliza al procesar información. En entornos de aprendizaje, un diseño visual inadecuado puede sobrecargar la memoria de trabajo del usuario, dificultando la comprensión de los contenidos. Según Sweller, la organización clara de la información y la reducción de elementos innecesarios permiten optimizar el proceso de aprendizaje, facilitando la construcción de conocimiento significativo [7]. En el contexto de entornos interactivos, la coherencia visual contribuye a disminuir la carga cognitiva al presentar la información de manera estructurada y comprensible.
- **Percepción visual** es el proceso mediante el cual el usuario interpreta y organiza la información que recibe a través de estímulos visuales. Este proceso es fundamental en entornos digitales, donde la forma, el color, la jerarquía y la disposición de los elementos influyen directamente en la interpretación del contenido. Investigaciones en el área del diseño visual destacan que una adecuada organización perceptiva permite identificar relaciones, funciones y jerarquías dentro de un sistema, facilitando la comprensión del usuario [8]. En entornos 3D, la percepción visual juega un papel clave en la identificación de objetos, espacios y dinámicas.

- Experiencia de usuario (UX) se refiere a la percepción global que tiene una persona al interactuar con un sistema digital, incluyendo aspectos como la facilidad de uso, la claridad de la información y la satisfacción durante la interacción. En entornos educativos interactivos, un diseño visual coherente contribuye a mejorar la experiencia del usuario al facilitar la navegación, la interpretación de los elementos y la comprensión de los contenidos. De acuerdo con estudios en diseño centrado en el usuario, una experiencia bien estructurada favorece el aprendizaje al reducir la confusión y permitir una interacción más intuitiva con el entorno [9].

Capítulo III

3. Manual del Programador

3.1. Dirección de arte y propuesta visual

La dirección de arte del proyecto se fundamenta en una estética de ciencia ficción futurista con un enfoque estilizado de baja poligonización (Low Poly). La identidad visual prioriza la claridad formal, el uso de contrastes lumínicos intensos y una paleta cromática basada en tonos metálicos oscuros (grises y azulados) complementados con líneas de emisión de luz en colores vibrantes (cian, verde, rojo y magenta). Esta propuesta visual busca generar un entorno inmersivo de temática espacial que resulte visualmente limpio, estético y funcional para la jugabilidad.

3.2. Diseño de personajes, escenarios y elementos gráficos

Esta sección detalla la propuesta estética y conceptual del videojuego, la cual se orienta hacia una línea visual de ciencia ficción estilizada basada en un enfoque Low Poly y de colores planos. A lo largo de este apartado se exponen y demuestran los diferentes recursos gráficos modelados e integrados para dotar de identidad, atmósfera y cohesión al mundo

virtual, estructurándose a través de los componentes principales que conforman la experiencia interactiva:

- **Personaje jugable:** Se diseña un modelo humanoide, el cual es un astronauta estilizado con proporciones amigables y compactas, representado con su baja estatura, compuesto por un traje espacial blanco, un visor oscuro reflectante y una mochila grande en tono azul brillante, estos colores son interactúan con facilidad ante la iluminación del entorno.

Ilustración 2

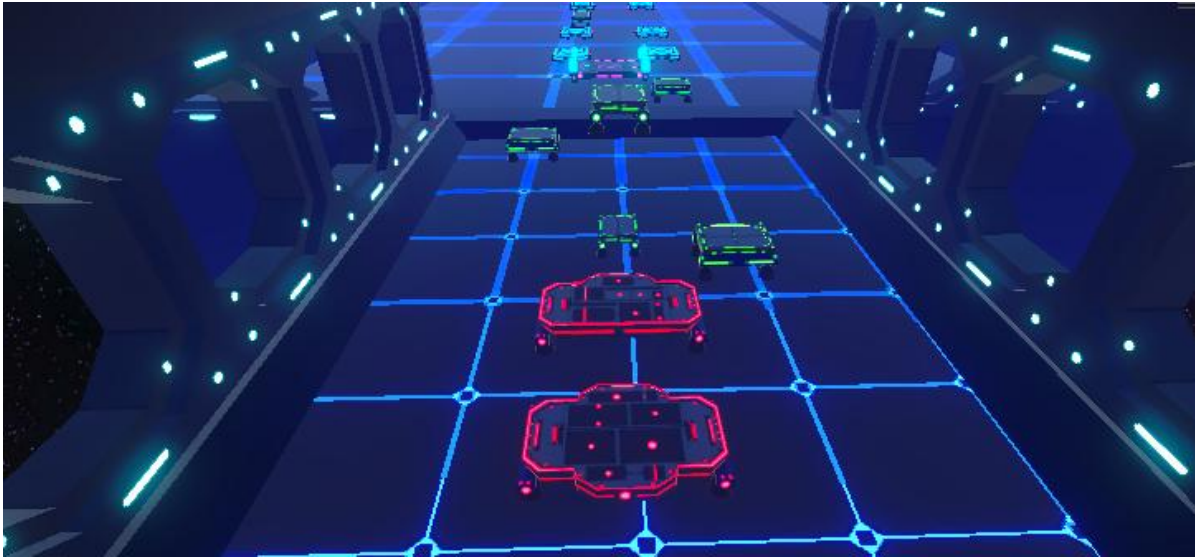
Personaje principal en el escenario.



Nota. Elaboración Propia en Unity

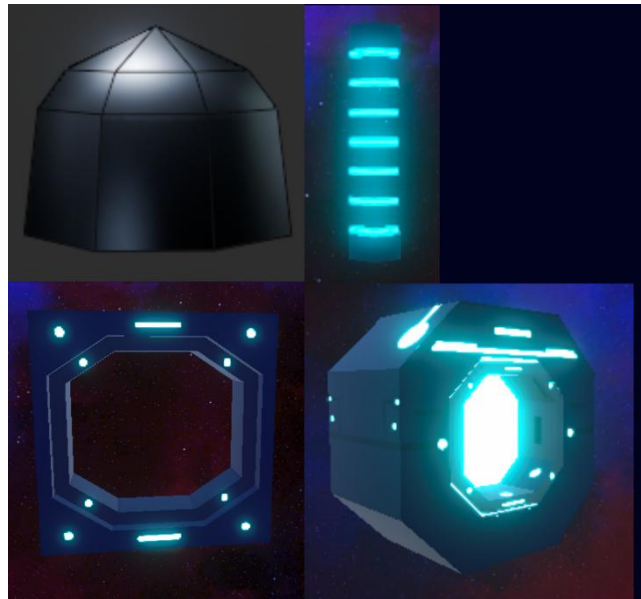
- **Escenario:** El entorno se compone de una estación espacial modular con ventanas grandes metálicas con patrones de rejilla y paneles en el techo equipados con luces de neón, además de un suelo está diseñado con un patrón en cuadrados emitivos de luz neón, si el jugador toca el suelo al caer pierde, además para avanzar en los niveles se diseñaron plataformas, estáticas y movibles, en donde el jugador tendrá que caminar y las cuales deberá saltar para evitar caer al suelo.

Ilustración 3
Escenario en Unity.



Nota. Elaboración propia en Unity.

- **Elementos gráficos:** Se incorporaron portales dimensionales circulares, ventanas con efectos de luz neón y barras energéticas que indican que plataformas delimitan las distintas zonas, además de las plataformas con luces neón a sus costados.

Ilustración 4*Modelos de plataformas.**Nota.* Elaboración propia con base de Blender.**Ilustración 5***Modelos de elementos decorativos.**Nota.* Elaboración propia con base de Blender y Unity.**3.3. Producción de assets 2D y 3D**

La producción de los recursos tridimensionales del proyecto se centró en lograr un equilibrio óptimo entre el atractivo visual y el rendimiento técnico. Para el personaje

principal y los elementos del entorno se optó por un estilo de baja poligonización (Low Poly), lo que permitió diseñar figuras limpias, con formas bien definidas y sin una cantidad excesiva de detalles geométricos que pudieran sobrecargar el sistema. Esta elección artística no solo facilitó el proceso de modelado, sino que también aseguró que el personaje principal se integrara de manera fluida y ligera dentro del espacio virtual, manteniendo una estética moderna y estilizada acorde a la temática espacial de la propuesta.

Para la construcción de los escenarios se implementó una estrategia de diseño modular. En lugar de modelar una estructura única e invariable para cada sector del mapa, se crearon piezas versátiles, como plataformas y ventanas que pueden combinarse y ensamblarse de distintas formas para armar diferentes secciones del nivel sin necesidad de fabricar decenas de modelos tridimensionales diferentes. Asimismo, para aquellas plataformas que presentaban formas más complejas o irregulares, se modelaron colisionadores personalizados adaptados a su silueta exacta, los cuales fueron importados al motor de juego para asegurar que el personaje pudiera caminar, saltar e interactuar con total precisión sobre ellas sin atravesar superficies o vacíos.

En cuanto al aspecto visual de las superficies, se recurrió a técnicas eficientes mediante el uso de texturas emisivas que funcionan como máscaras. Esto significa que, en lugar de añadir más geometría o luces independientes en cada rincón, se configuraron los materiales para que únicamente ciertas zonas específicas de los modelos brillen con intensidad creando los característicos efectos de neón y paneles luminosos de la estación espacial. De este modo, se logró una apariencia detallada y futurista que resalta en los entornos oscuros sin comprometer el rendimiento técnico ni encarecer el procesamiento gráfico del proyecto.

Finalmente, para dotar al escenario de mayor dinamismo visual, se incorporó una animación especial en las texturas de ciertos elementos clave, como las barras y las plataformas que contienen una flecha. Mediante una técnica de desplazamiento de coordenadas de textura en bucle, se logró el efecto óptico de que las líneas de luz sobre estos objetos se muevan de forma constante y fluida. Este detalle visual ayuda a captar la atención del usuario y guía de manera intuitiva su recorrido a través de la experiencia interactiva sin requerir complejos sistemas de animación geométrica.

Tabla 4
Modelos 3D creados

Nombre modelo	Descripción
Astra	Modelo tridimensional y posable del personaje jugable del videojuego.
barra_energia	Módulo vertical equipado con textura de segmentos de luz cian brillante, diseñado para elementos energéticos del escenario.
barra_energia_bordes	Componentes cilíndricos de refuerzo y cierre situados en los extremos de la barra de energía.
plataforma_1_centro	Módulo central de suelo plano decorado con líneas de luz lateral en tonos magenta.
Plataforma_1_esquina	Pieza angular de plataforma diseñada para unirse de forma modular con los módulos centrales magenta.
Plataforma_2_centro	Módulo central de plataforma caracterizado por incorporar franjas de iluminación verde.
Plataforma_2_esquinas	Pieza angular complementaria de plataforma con detalles lumínicos en tono verde.
Plataforma_3_centro1	Módulo central de suelo con estructura lineal y detalles de emisión roja.
Plataforma_3_centro2	Variante geométrica de módulo central de plataforma equipada con iluminación roja.
Plataforma_4_centro	Módulo central de plataforma adaptado para la expansión la longitud de las plataformas.
Plataforma_4_esquina	Pieza angular correspondiente a la serie de plataformas con detalles luminosos rojos.
Plataformas-Cohete	Estructuras modulares con diseño curvo simulando cohetes espaciales.
Rampa_cohete	Estructura curva con diseño metálico.
Rampa_flecha	Superficie plana texturizada con flechas luminosas en color cian para guiar la trayectoria del jugador.

Rampa_metal	componente de aspecto metálico limpio con un borde perimetral iluminado en tono azul, sirve para contener dentro.
Rampa_tubos	Plano inclinado de aspecto metálico limpio con un borde perimetral iluminado en tono azul.
Portal	Estructura angular espacial con paneles oscuros, pernos metálicos y un marco central con luces emisivas.
Ventana	Marco estructural cuadrado decorativo dotado de puntos luminosos en sus esquinas para ambientación espacial.

Nota. Elaboración propia.

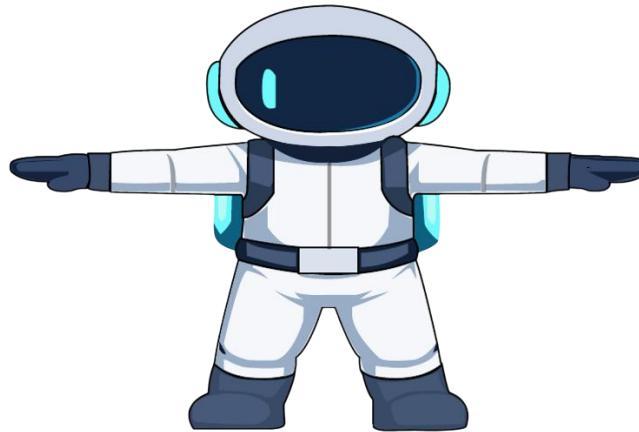
3.4. Pipeline de producción artística

El flujo de trabajo (pipeline) de producción artística estableció una secuencia metódica y ordenada para transformar los conceptos visuales iniciales en recursos tridimensionales totalmente funcionales dentro del entorno virtual. Este proceso se estructuró y dividió en dos líneas de trabajo principales: la creación del personaje principal y el diseño modular del escenario, asegurando una identidad estética coherente y un rendimiento técnico eficiente a lo largo de todo el proyecto.

3.4.1. Desarrollo del personaje jugable

- **Concepción visual y plantilla de referencia:** El diseño inició con la definición gráfica de la identidad del personaje mediante un *model sheet* en postura neutra (*T-pose*). Esta ilustración definió un astronauta de proporciones estilizadas, compactas y amigables. La imagen se importó como referencia de fondo en las vistas ortogonales de Blender para guiar la construcción geométrica con precisión dimensional.

Ilustración 6
Ficha de personaje.

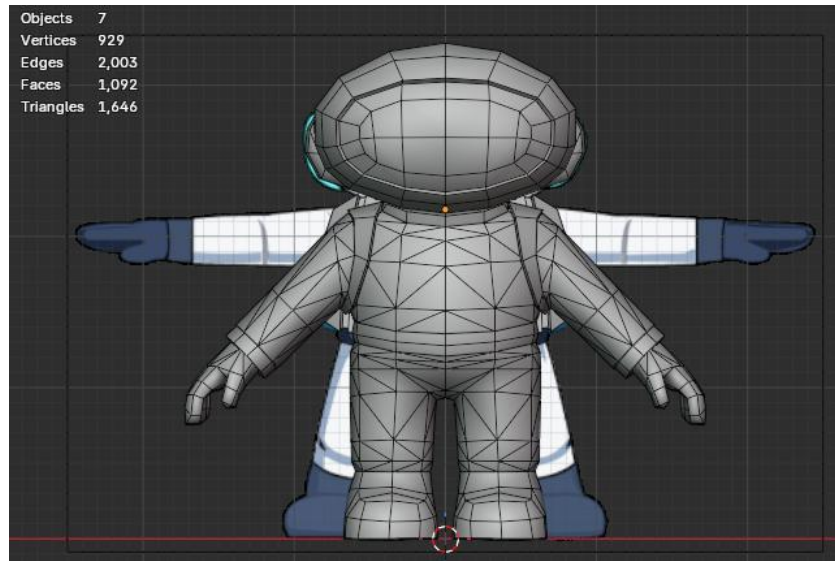


Nota. Elaboración propia.

- **Modelado tridimensional (Low Poly):** La geometría se construyó manteniendo un presupuesto estricto de baja poligonización (con un límite máximo de 1000 vértices y 2000 triángulos y un resultado final de 929 vértices y 1,646 triángulos). El proceso siguió las siguientes etapas:
 - **Cabeza y casco:** Partiendo de un cubo base con el modificador Mirror (para garantizar simetría), se aplicaron biselados (Bevel) para suavizar esquinas y cortes de bucle (Loop Cut) para definir la curvatura. Las herramientas de extrusión (Extrude) e inserción de caras (Insert Faces) permitieron dar forma al visor y a las estructuras laterales del casco.
 - **Cuerpo y extremidades:** A partir de un cilindro base simétrico, se extrajeron brazos y piernas, ubicando cortes estratégicos en articulaciones (codos y rodillas) para facilitar la deformación durante la posterior fase de animación. Bajo esta misma técnica se integraron botas, guantes, cinturón y correas.
 - **Mochila y detalles emisivos:** Se modeló como una pieza independiente basada en un cubo. Se aplicaron biselados en sus bordes para simular un

acabado metálico y pequeñas extrusiones hacia el interior para alojar las superficies que emitirán luz LED.

Ilustración 7
Modelo de Personaje.



Nota. Elaboración propia con base de Blender.

- **Unificación de escala y limpieza:** Una vez ensambladas las piezas en la posición requerida, se verificó la topología y se aplicó la transformación de escala (Ctrl + A > Apply Scale) para fijar el valor de escala en 1 en todos los ejes, garantizando la consistencia de transformación antes del etiquetado y exportación.

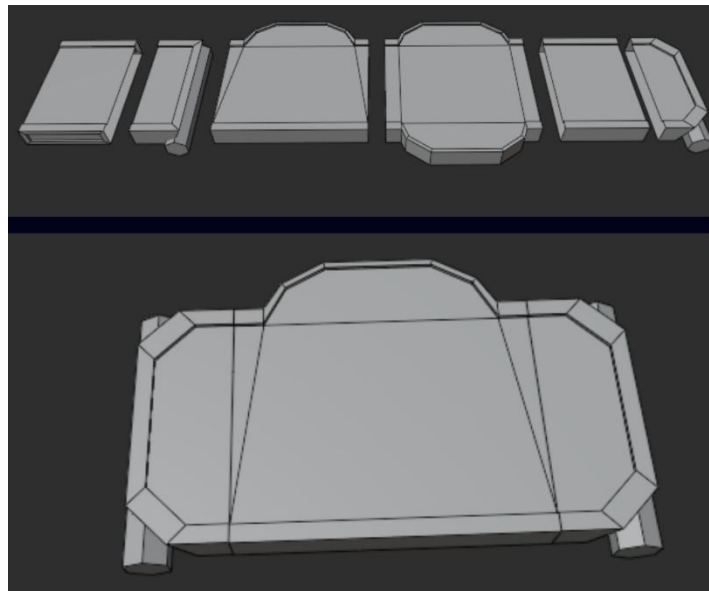
3.4.2. Diseño modular del escenario y elementos del entorno

Para optimizar el tiempo de producción y la carga de procesamiento, el escenario no se concibió como un bloque rígido, sino como un sistema modular de piezas reutilizables que pueden combinarse, rotarse y acoplarse estratégicamente en Blender y Unity.

- **Estructura de las plataformas:**
 - **Piezas únicas (Inicio y Fin):** Diseñadas a partir de cubos base y extrusiones para delimitar los extremos de cada nivel. Incorporan orificios y elementos cilíndricos distintivos.
 - **Módulos estándar (Estáticos y Movibles):** Comparten bordes de contacto con geometría y dimensiones idénticas. Para asegurar que los vértices de los bordes entre distintas piezas coincidan perfectamente al ensamblarse, se utilizó la herramienta de ajuste magnético (Snap).

Ilustración 8

Modelos base de Plataformas.



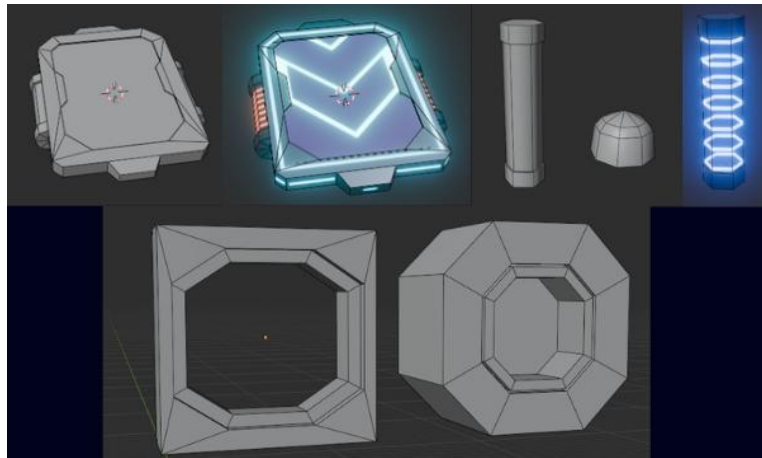
Nota. Elaboración propia con base de Blender.

- **Objetos funcionales y decorativos:**
 - Se crearon elementos complementarios como la Rampa Flecha (plano con guía direccional), mini-cohetes, ventanas panorámicas, portales dimensionales y barras de energía.

- **Limpieza topológica:** Durante el modelado de estos objetos, los vértices sueltos o superpuestos se fusionaron mediante la función Merge ($M > By$ Distance o At Last), manteniendo mallas limpias sin geometría redundante.

Ilustración 9

Modelos base de elementos decorativos.

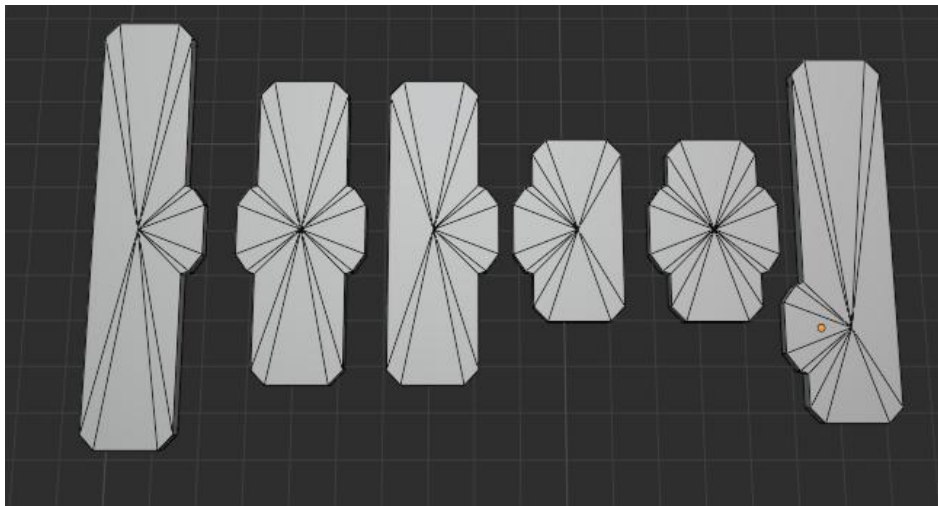


Nota. Elaboración Propia con base de Blender.

3.4.3. Colisionadores personalizados y geometría base

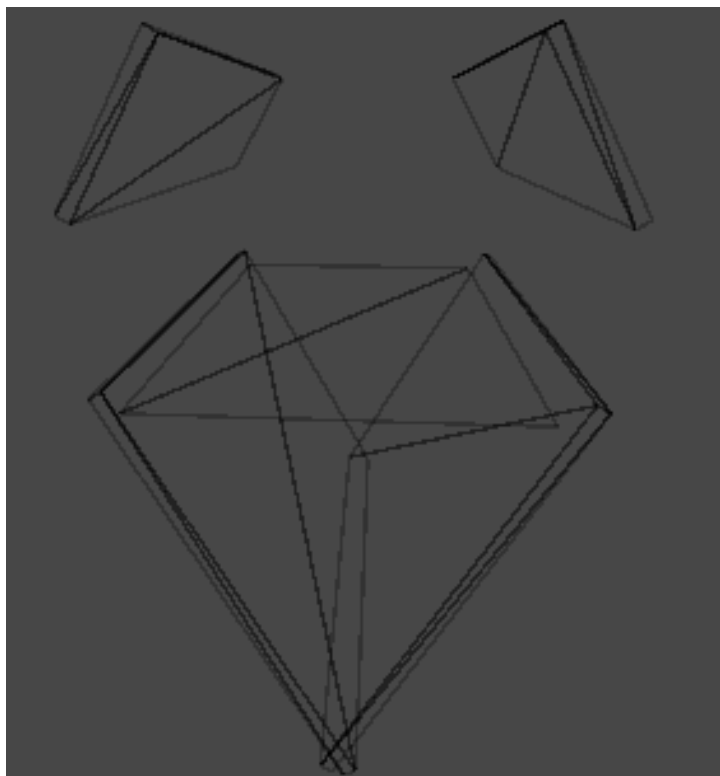
- **Colisionadores de física (Custom Colliders):** Debido a que las mallas decorativas complejas sobrecargan el sistema de físicas de Unity, se modelaron 6 mallas de colisión simplificadas en Blender. Estas estructuras respetan la silueta general de las plataformas, pero omiten detalles menores (como orificios o adornos), sin requerir mapas UV ni texturas.
- **Estructuras primarias en el motor:** Las paredes de gran tamaño y los suelos base del mapa no se exportaron desde Blender; en su lugar, se construyeron directamente en Unity utilizando primitivas 3D (cubos y planos) para simplificar el flujo técnico de integración.

Ilustración 10
Colliders Personalizados.



Nota. Elaboración propia con base de Blender.

Ilustración 11
Paredes en Unity.



Nota. Elaboración propia con base de Unity.

3.5. Preparación y animación de personajes y objetos

El proceso de animación se diseñó para dotar de movilidad fluida al personaje jugable y dinamismo al entorno, asegurando que la geometría Low Poly se deforme de manera limpia y orgánica durante el movimiento. Este procedimiento abarcó desde la estructura ósea (rigging) y el ajuste de pesos (skinning), hasta la creación del set de animaciones en Blender y la implementación de efectos dinámicos en Unity.

3.5.1. Construcción y adaptación de la armadura (Rigging)

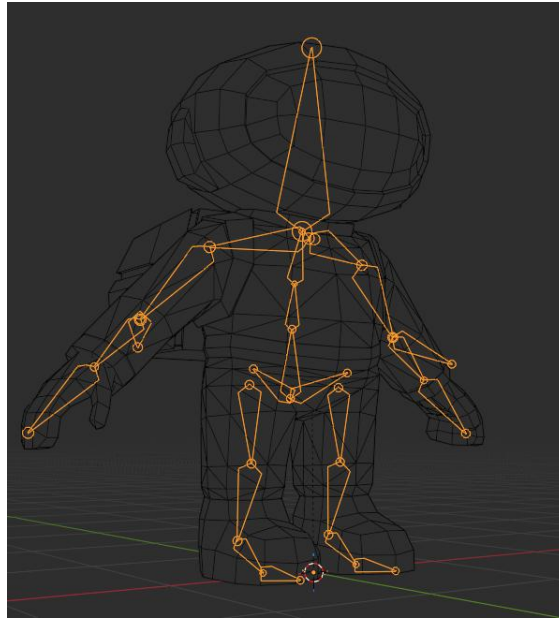
Para la estructura ósea del astronauta se utilizó la herramienta integrada **Rigify** en Blender, partiendo de un meta-esqueleto humanoide estándar:

- **Simplificación de la estructura:** Dado el diseño estilizado y compacto del personaje, se eliminaron los huesos innecesarios del rostro y se simplificó la cadena ósea de las manos, optimizando la complejidad técnica del modelo.
- **Alineación anatómica:** En el modo de edición (Edit Mode), se ajustó la posición, escala y rotación de los huesos principales (columna, hombros, codos, caderas y rodillas) para hacerlos coincidir con la geometría interna del traje espacial.

Seguidamente, se presenta el resultado del personaje principal con el armature aplicado, evidenciando que la estructura ósea se encuentra correctamente alineada con la malla (mesh) del modelo 3D, lo que garantiza una deformación adecuada durante el proceso de animación. Tal como se aprecia en la Ilustración 12.

Ilustración 12

Armature en personaje principal.



Nota. Elaboración propia con base de Blender.

3.5.2. Vinculación y depuración de pesos (Skinning)

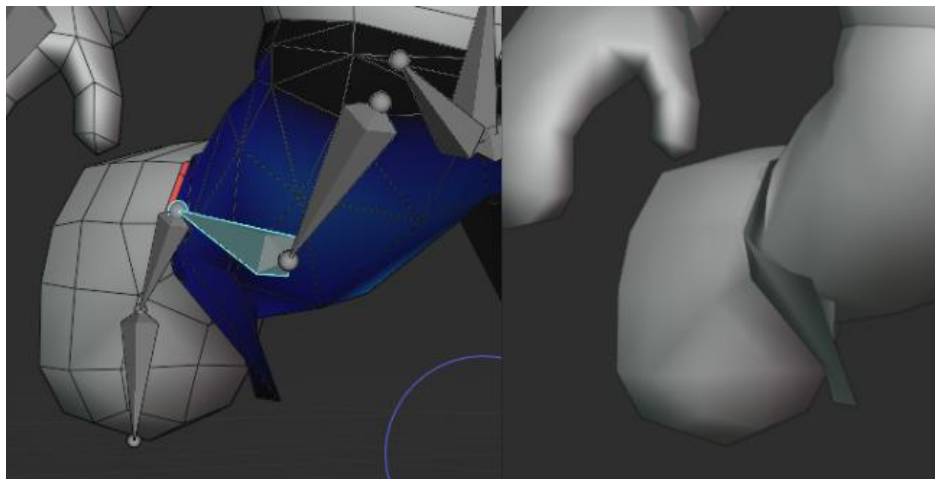
Una vez adaptado el esqueleto, se procedió a la vinculación entre la malla tridimensional y la armadura:

- **Emparentado automático:** En el modo objeto (Object Mode), se enlazó la malla a la armadura utilizando el comando de emparentado por pesos automáticos (Ctrl + P > With Automatic Weights). Esta función calcula la cercanía espacial entre los vértices y los huesos para asignar una influencia inicial de movimiento.
- **Pintado de pesos manual (Weight Paint):** Para corregir deformaciones anómalas en la geometría de baja poligonización, se refinaron las áreas críticas de flexión (rodillas, codos y cintura) mediante la modalidad Weight Paint.
 - A través de brochas de edición como Draw (añadir peso), Subtract (restar peso) y Smooth (suavizar transiciones), se ajustó la graduación de influencia

representada en el mapa cromático desde azul (sin influencia) hasta rojo (influencia máxima), evitando rupturas y pliegues bruscos al doblar las articulaciones.

Ilustración 13

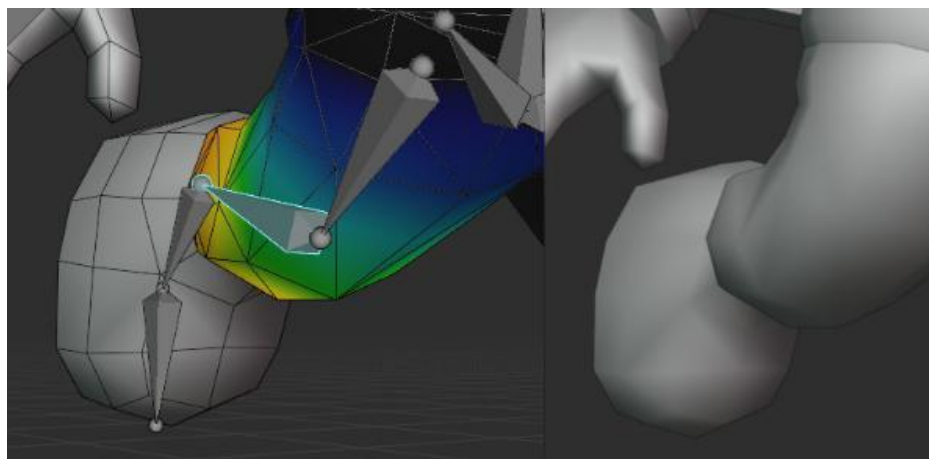
Errores en Weight Paint Fuente.



Nota. Elaboración propia con base de Blender.

Ilustración 14

Weight Paint arreglado a mano.

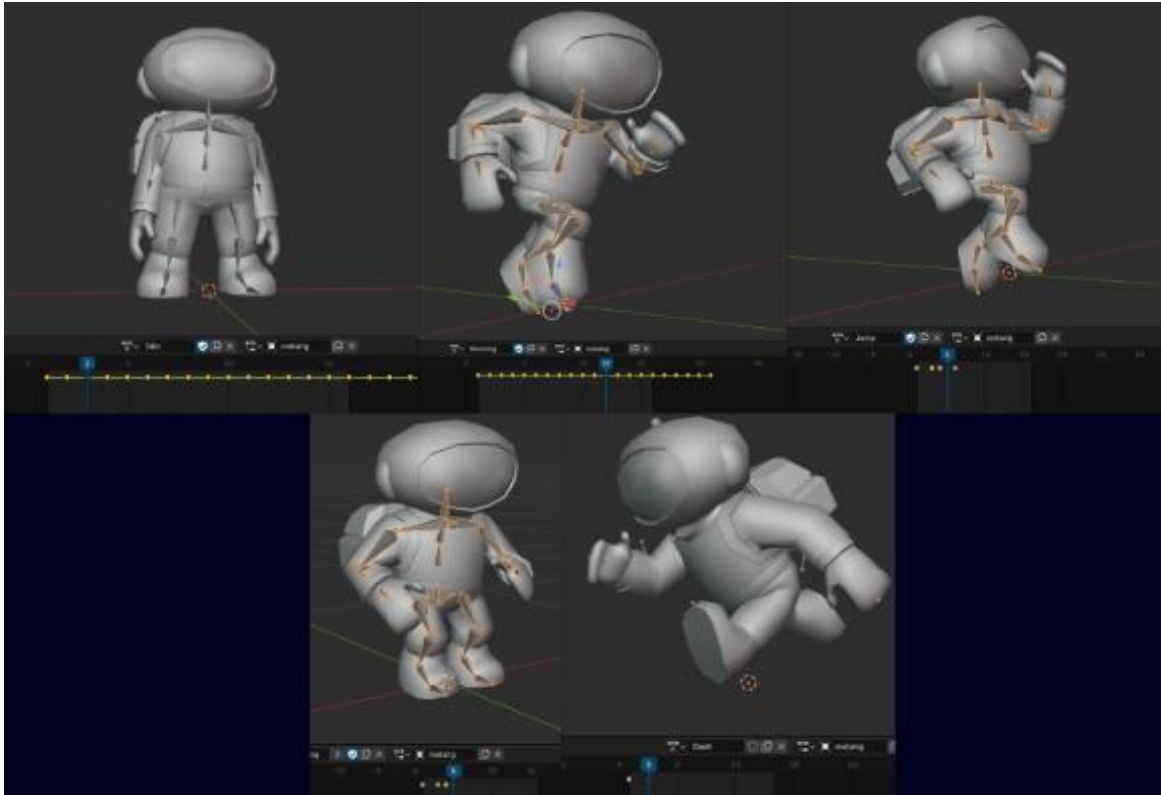


Nota. Elaboración propia con base de Blender.

3.5.3. Generación de los clips de animación del personaje

Las animaciones del astronauta se construyeron en Blender utilizando fotogramas clave (keyframes) en la línea de tiempo. Posteriormente, se exportaron al motor Unity para su integración dentro del controlador de estados de animación (Animator Controller):

- **Reposo (Idle):** Ciclo continuo (loop) que representa al personaje detenido en postura relajada, aplicando un leve balanceo corporal y respiratorio manteniendo los brazos abajo.
- **Caminata (Walk):** Ciclo de pasos rítmico y simétrico con balanceo alternado de extremidades. Sirve como base del desplazamiento estándar y su velocidad de reproducción puede acelerarse en el motor para simular la acción de correr.
- **Salto (Jump):** Movimiento de respuesta rápida que captura la elevación del personaje de forma casi inmediata, omitiendo fases prolongadas de anticipación para mantener un control ágil durante el juego.
- **Aterrizaje (Grounding):** Transición corta ejecutada al hacer contacto con el suelo; muestra una rápida flexión en rodillas y codos que simula la absorción del impacto.
- **Impulso rápido (Dash):** Acción de desplazamiento veloz con una duración menor a un segundo. Se compone de una postura inclinada hacia adelante con los brazos extendidos para transmitir aceleración.

Ilustración 15*Clips de animación en Blender.*

Nota. Elaboración propia con base de Blender.

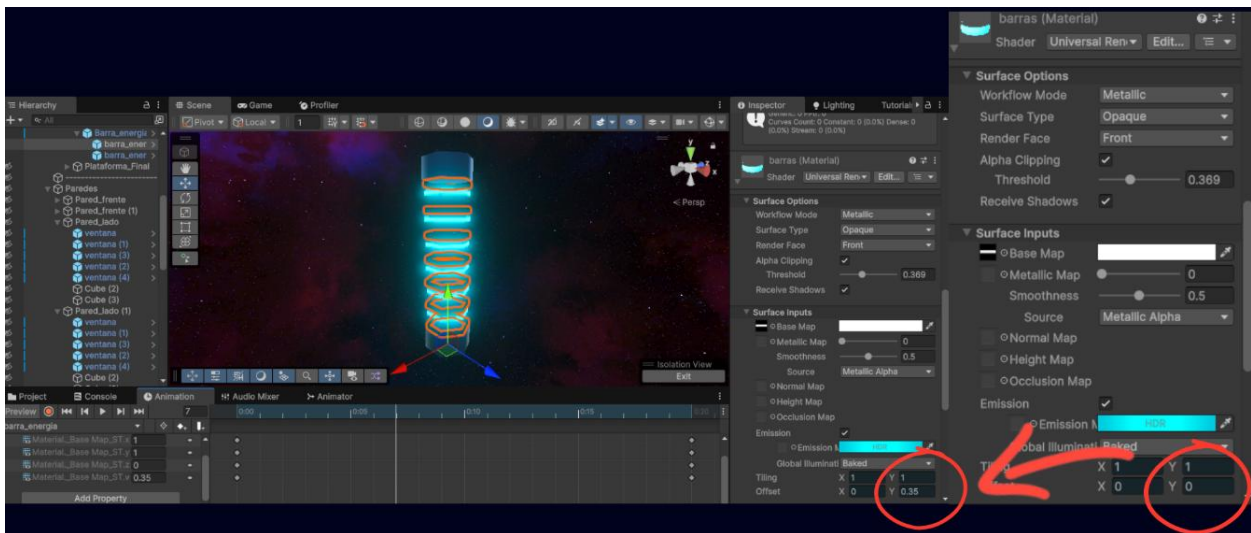
3.5.4. Animación técnica de elementos del escenario en Unity

A diferencia del personaje, los elementos interactivos y decorativos del entorno como las barras de energía y las rampas direccionales no requirieron estructuras óseas. Su movimiento se resolvió de forma directa y eficiente dentro de Unity mediante la manipulación de materiales:

- **Desplazamiento de mapa UV (UV Offset):** Se configuró una animación mediante secuencias de fotogramas que modifican continuamente las coordenadas del mapa de textura en los ejes horizontales o verticales.

- Al variar el parámetro Offset en un ciclo continuo del valor 0 a 1, se genera el efecto visual de flujo constante en las flechas indicadoras y en la emisión luminosa de las barras de energía sin añadir carga geométrica adicional.

Ilustración 16
Animación de UV.



Nota. Elaboración propia con base de Unity.

3.6. Materiales, texturas, iluminación y renderización

Una vez finalizado el modelado 3D y el desplegado de coordenadas UV, el proceso continuó con la fase de texturizado y la posterior configuración de materiales en el motor de juego. La propuesta gráfica se enfocó en una estética Low Poly estilizada, priorizando paletas de colores limpios, contrastes marcados y una alta claridad visual.

3.6.1. Generación de mapas de textura

Para vestir los modelos tridimensionales se crearon tres tipos de mapas principales utilizando un flujo de trabajo combinado entre Blender y Krita:

- **Mapa de color base (Albedo):** Define la pigmentación y los tonos primarios de las superficies.

- **Personaje jugable:** Se texturizó íntegramente en Blender mediante el modo Texture Paint, aplicando bloques de color plano sobre la malla.
- **Plataformas y escenario:** Se diseñaron en Krita para aprovechar sus herramientas de trazado recto y vectorial, facilitando la creación de patrones limpios como rejillas y bordes estructurales.
- **Mapa metálico (Metallic):** Determina qué áreas de la superficie reaccionan a la luz ambiental simulando acabados metálicos. Consiste en una máscara en escala de grises pintada en Blender mediante colores planos, donde el tono blanco asigna el valor metálico máximo y el negro conserva un acabado mate.
- **Mapa de emisión (Emission):** Funciona como una máscara en blanco y negro para delimitar las zonas que emiten luz propia (paneles luminosos, líneas de neón e indicadores direccionales). Pintadas en Krita, estas texturas permiten simular brillo ambiental sin necesidad de añadir luces físicas adicionales que sobrecarguen el rendimiento del sistema.

Ilustración 17
Texturas en Unity.



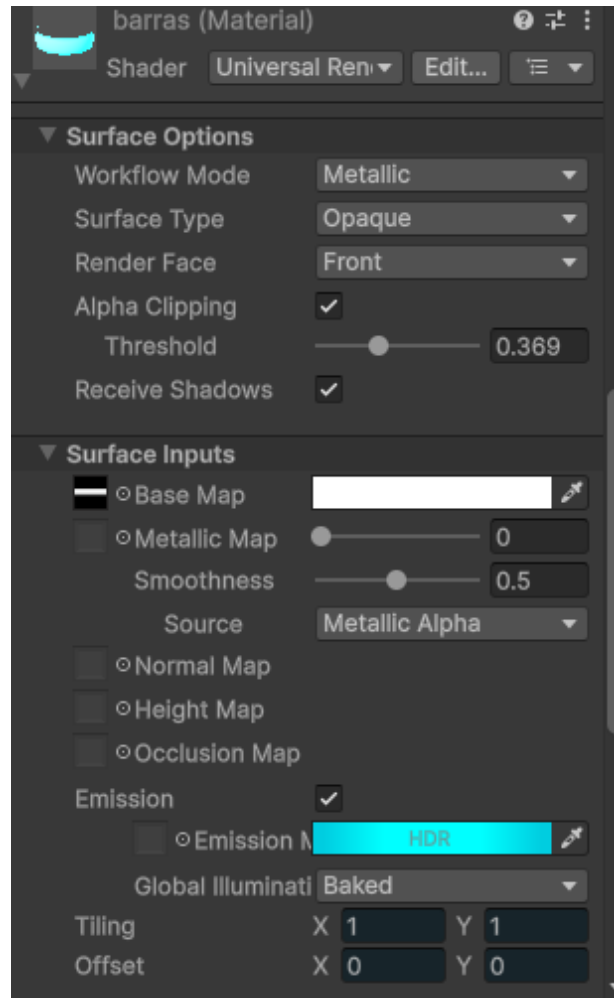
Nota. Elaboración propia con base de Unity.

3.6.2. Configuración e integración de materiales en Unity

Con las imágenes exportadas, se procedió a la construcción y parametrización de los materiales dentro del editor de Unity para su correcta renderización:

- **Creación del material:** En la ventana Project de Unity, se creó un nuevo activo de tipo Material asignado a cada elemento o conjunto modular del escenario.
- **Asignación de mapas (Slots):** Desde la ventana del Inspector, se vincularon las texturas correspondientes arrastrándolas directamente a sus respectivas casillas de control:
 - La textura cromática se asignó al canal **Albedo**.
 - La máscara de reflectividad se ubicó en la casilla **Metallic**.
 - Se activó la casilla de verificación **Emission** para habilitar el canal de luz propia y se arrastró la máscara correspondiente a su casilla de textura.
- **Ajuste de parámetros y color de emisión:** Una vez vinculados los mapas, el editor de Unity permitió ajustar la intensidad lumínica y cambiar el tono de color de la emisión (HDR Color) de forma rápida y flexible, logrando que los neones y señales resalten en el entorno virtual sin alterar la textura base.

Ilustración 18
Material en Unity.



Nota. Elaboración propia con base de Unity.

3.7. Creación de efectos visuales

La creación de los efectos visuales y la atmósfera del proyecto se desarrolló directamente dentro del motor de juego Unity, aprovechando las herramientas de post-procesamiento y configuración ambiental para realzar la estética de ciencia ficción. Este proceso se dividió en dos componentes principales: los efectos de realce visual en pantalla y la iluminación ambiental general del escenario.

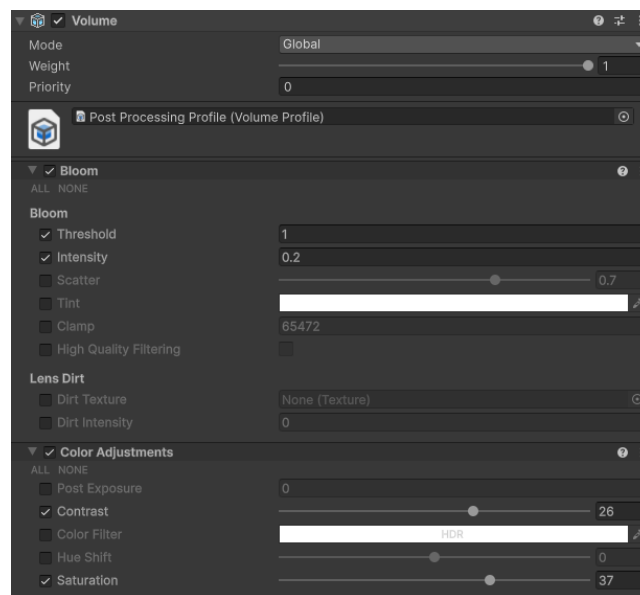
3.7.1. Efectos de Postprocesamiento (Bloom y Color Adjustments)

Mediante el uso de un componente de volumen global (*Global Volume*), se configuraron filtros de postprocesamiento orientados a mejorar la calidad estética de la escena de forma inmediata:

- **Efecto Bloom:** Se activó para interactuar directamente con los materiales emisivos de las plataformas y el personaje. Al configurar un umbral (*Threshold*) de 1 y una intensidad equilibrada, se logró que las líneas de neón y los paneles luminosos emitan un brillo suave y difuso hacia el entorno, simulando luces artificiales en el espacio.
- **Ajustes de Color (*Color Adjustments*):** Se incrementaron ligeramente los niveles de contraste (con un valor de 26) y saturación (con un valor de 37). Estos ajustes permitieron que los colores de los escenarios y el fondo estelar destaquen con mayor fuerza, logrando una presentación visual más viva, atractiva y estilizada para el usuario.

Ilustración 19

Efectos de PostProcesado.



Nota. Elaboración propia con base de Unity.

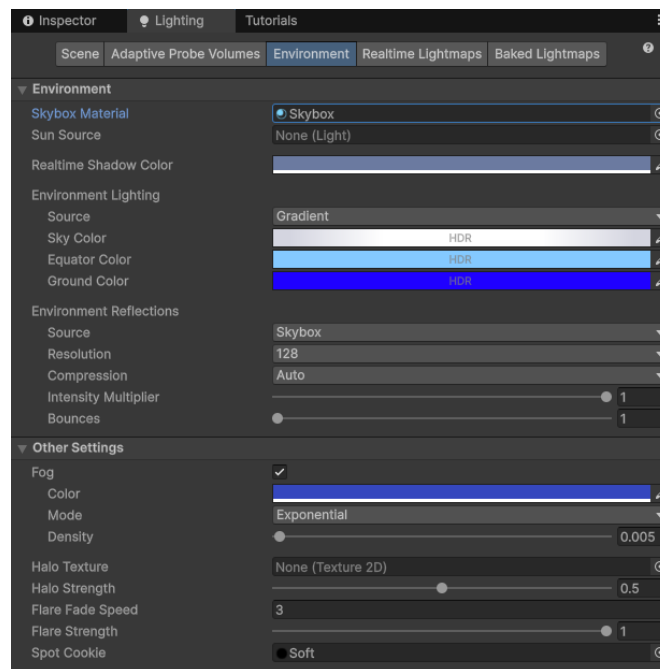
3.7.2. Configuración de la Atmósfera y Entorno (Lighting y Fog)

Para dar cohesión visual al espacio exterior y establecer la iluminación general de los niveles, se ajustaron los parámetros en la ventana de iluminación (*Lighting*) de Unity:

- **Iluminación del entorno (*Environment Lighting*):** Se configuró la fuente de luz ambiental utilizando la opción de *Gradient* (Gradiente). Se asignaron tres tonos azulados distintos para el cielo, el ecuador y el suelo, envolviendo toda la escena en una paleta cromática coherente con la ambientación espacial.
- **Efecto de Niebla (*Fog*):** Se habilitó la opción de niebla en modo *Exponential* (Exponencial) con una densidad ligera de 0.005 y un tono azulado acorde a la escena. Este ajuste genera un difuminado gradual y sutil sobre las plataformas y elementos que se encuentran más alejados del jugador, ayudando a crear una sensación de profundidad espacial y suavizando la transición visual en la lejanía.

Ilustración 20

Efectos de Iluminación del Entorno.



Nota. Elaboración propia con base de Unity.

3.8. Integración de recursos visuales en el motor

La integración de los recursos gráficos en el motor Unity se estructuró de manera metódica para garantizar que tanto el personaje como los elementos del escenario funcionaran correctamente a nivel visual, estético y físico dentro de la experiencia interactiva.

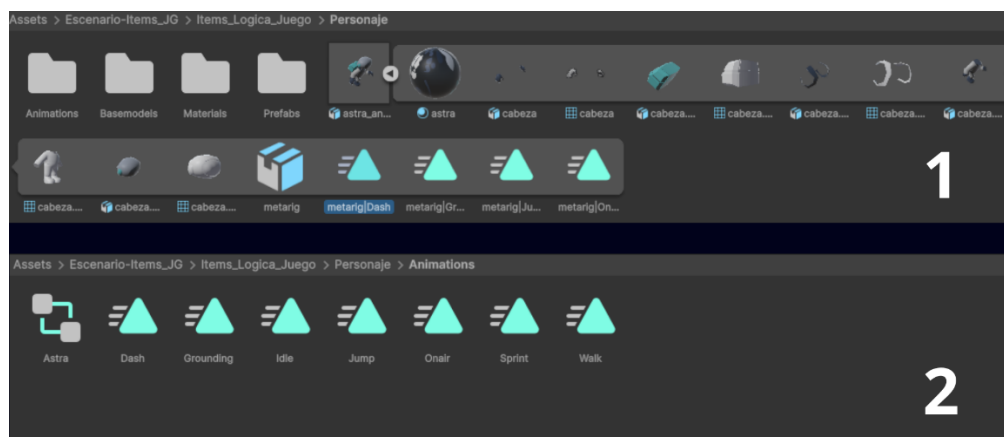
3.8.1. Integración y Configuración del Personaje Jugable

- Estructura y Escala: Se importaron los archivos en formato FBX del personaje junto con su esqueleto (*armature*) y sus respectivas animaciones, asegurando que todos compartieran la misma escala y cantidad de huesos para evitar fallos en la ejecución de los movimientos.
- Organización jerárquica: El modelo se arrastró a la escena, se utilizó la opción *Prefab Unpack* para separarlo de su estructura predefinida y se creó un objeto vacío (*Empty Object*) llamado *Player* para contenerlo, lo que facilitó la gestión centralizada de sus componentes.
- Componentes físicos: Al objeto *Player* se le añadió un *Capsule Collider* para definir el volumen de colisión del jugador, un *Rigid Body* para permitirle reaccionar ante la gravedad del entorno, y un *Physics Material* asignado al colisionador para evitar que el personaje se quede pegado o friccione incorrectamente contra las paredes.
- Iluminación de apoyo: Se incorporó una luz focal (*Spot Light*) de color amarillo en la base del personaje para proyectar un círculo luminoso que indica visualmente el lugar exacto donde va a caer.

- Efecto de estela (*Dash*): Se añadió un objeto secundario equipado con un componente *Trail Renderer* para generar una estela visual detrás del personaje, la cual se activa y desactiva mediante la lógica del juego únicamente al realizar la acción de aceleración rápida.
- Gestión de animaciones: Se creó un controlador de animaciones (*Animator Controller*) en el prefab del personaje y se extrajeron los clips de animación individuales de los archivos FBX mediante el atajo de teclado Ctrl+D, organizándolos en una carpeta específica para mantener limpio el proyecto.

Ilustración 21

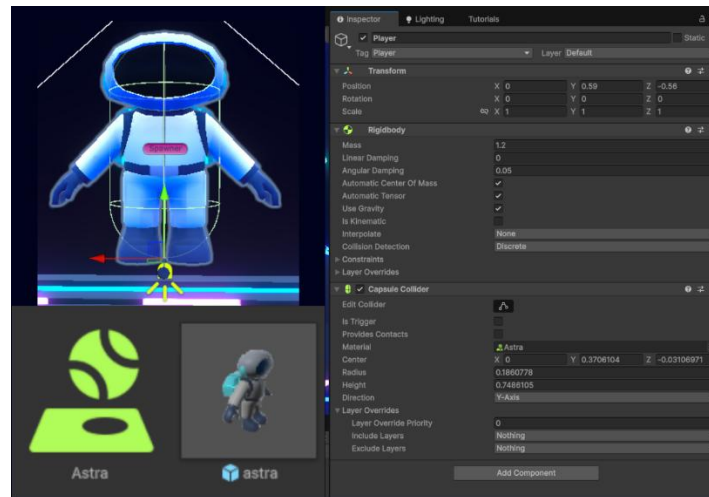
Integración de Personaje y animaciones dentro de las carpetas de Unity



Nota. Elaboración propia con base de Unity.

Ilustración 22

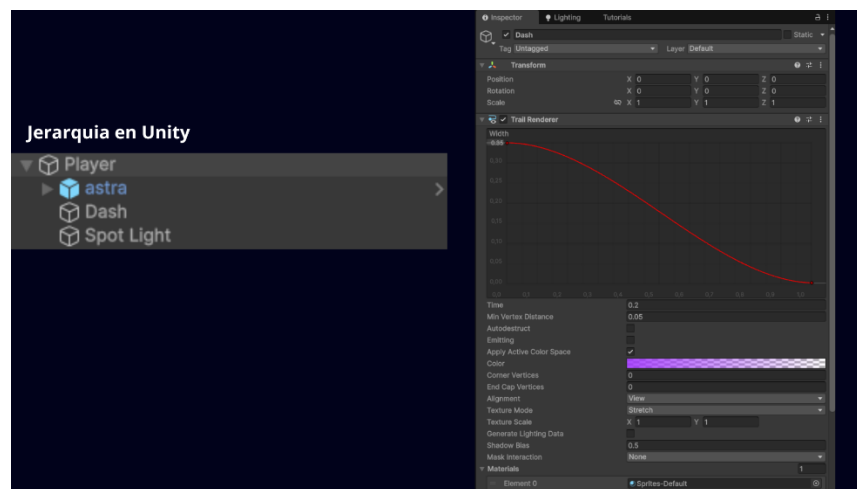
Configuración de Collider de personaje.



Nota. Elaboración propia con base de Unity.

Ilustración 23

Jerarquía y efecto de Dash en Unity.



Nota. Elaboración propia con base de Unity.

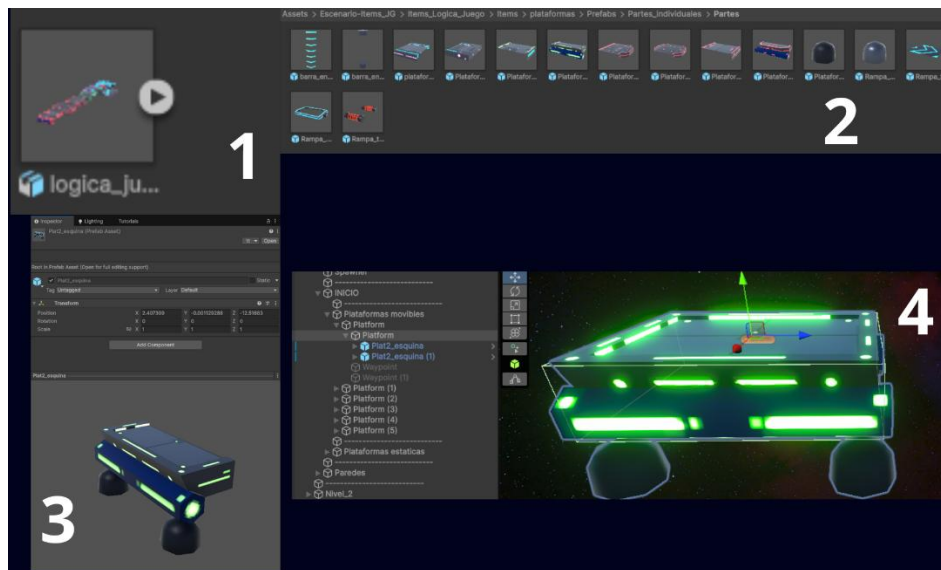
3.8.2. Integración del Escenario y Plataformas Modulares

- **Desglose de piezas:** Los archivos FBX contenedores de las plataformas se importaron al motor y se desempacaron (*Unpack*) dentro de la jerarquía para separar sus componentes individuales y convertirlos en piezas reutilizables.

- **Ensamblaje modular:** Cada plataforma compuesta se estructuró agrupando las piezas individuales dentro de un objeto vacío (*Empty Object*) situado exactamente en el centro de la plataforma, permitiendo guardarlas como nuevos prefabs para su uso repetido.
- **Colisiones estandarizadas:** Para las plataformas de formas sencillas, se aplicó un componente genérico de colisión (*Box Collider*) en el objeto contenedor, evitando saturar el sistema con colisionadores individuales para cada pequeña pieza.
- **Colisionadores personalizados:** En el caso de plataformas con diseños complejos o irregulares, se importó un archivo FBX con el colisionador específico, extrayendo su malla geométrica (*mesh*) con el atajo Ctrl+D para asignarla al componente *Mesh Collider* y asegurar una precisión milimétrica en las físicas.

Ilustración 24

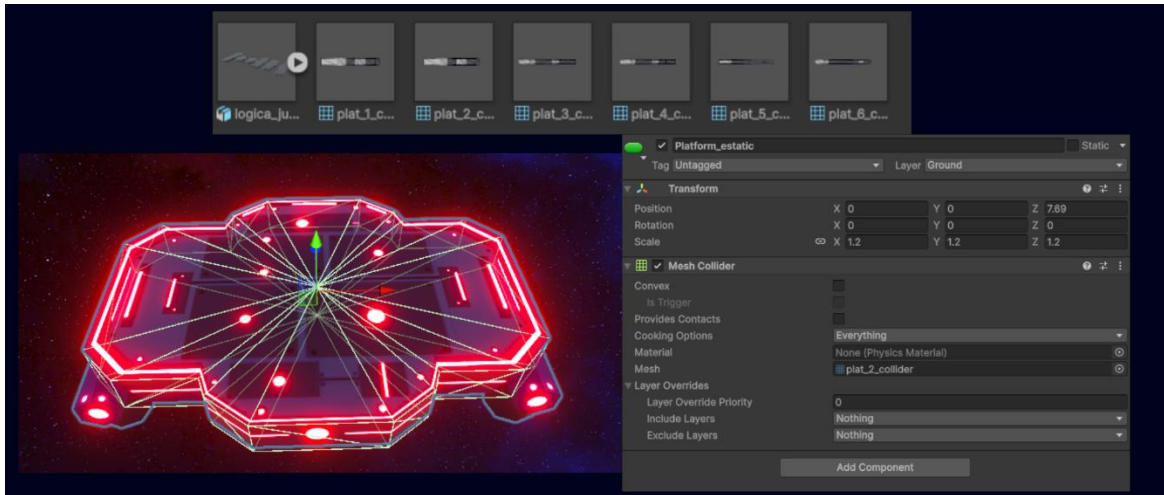
Importación y edición de una plataforma modular en Unity.



Nota. Elaboración propia con base de Unity.

Ilustración 25

Importación y edición de un Collider Personalizado en Unity.



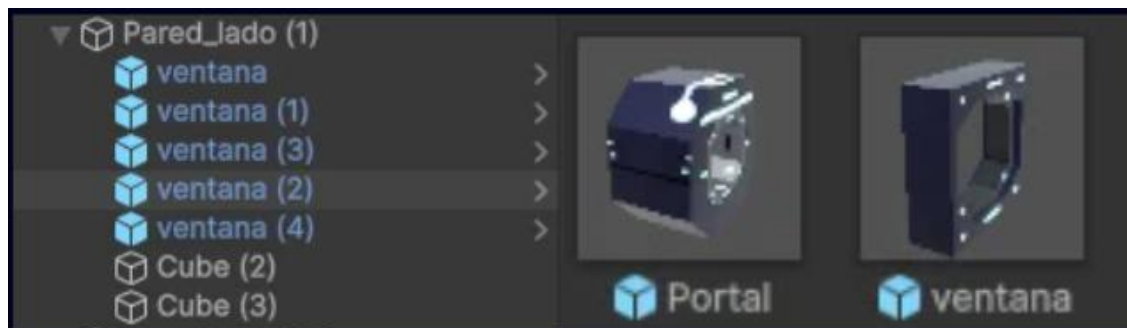
Nota. Elaboración propia con base de Unity.

3.8.3. Integración de Elementos Decorativos

- **Materiales y texturas:** Los objetos puramente estéticos (como las ventanas) se importaron mediante sus archivos FBX y se les asignaron los materiales con sus respectivas texturas correspondientes.
- **Ausencia de físicas:** Al tratarse de elementos decorativos, se omitió la incorporación de colisionadores ya que no interactúan de forma física directa con el jugador.
- **Orden en la escena:** Estos elementos se acomodaron y apilaron según la composición visual deseada, agrupándolos dentro de objetos vacíos (Empty Objects) ubicados en el centro de cada conjunto para mantener la limpieza y organización en la jerarquía del proyecto.

Ilustración 26

Importación de elementos decorativos a Unity.



Nota. Elaboración propia con base de Unity.

3.9. Optimización gráfica

La optimización gráfica del proyecto se implementó como un pilar fundamental para garantizar que el videojuego funcione de manera fluida, estable y con una tasa de cuadros por segunda constante, sin sacrificar la identidad visual de ciencia ficción propuesta. Para lograr este rendimiento óptimo, se aplicó un conjunto de estrategias técnicas y artísticas integradas desde las fases de modelado en Blender hasta su configuración final en el motor de juego.

3.9.1. Optimización Geométrica y de Recursos

- **Estilo Low Poly:** Se adoptó una baja poligonización en el personaje y los elementos del escenario, reduciendo al mínimo la cantidad de vértices y triángulos procesados por la tarjeta gráfica para mantener formas limpias y ligeras.
- **Agrupamiento en archivos FBX:** Los modelos se estructuraron agrupando múltiples piezas dentro de un mismo archivo fuente, facilitando la importación y permitiendo desempacar los elementos individuales directamente en la jerarquía de Unity.
- **Organización estática (Static):** Los objetos fijos del escenario se organizaron dentro de objetos vacíos (Empty Objects) y se marcaron con la opción de Static en el motor,

lo que permite al sistema realizar optimizaciones internas de procesamiento por lotes (batching).

3.9.2. Optimización de Iluminación y Sombras

- **Eliminación de luz direccional y sombras:** Se prescindió del uso de luces direccionales principales y se desactivó la generación de sombras en todos los modelos del juego, eliminando los costosos cálculos de sombreado en tiempo real de la GPU.
- **Iluminación ambiental (*Environment Lighting*):** Toda la iluminación de los niveles se resolvió mediante gradientes de luz ambiental y efectos de niebla (*Fog*), logrando una atmósfera espacial envolvente sin sobrecargar el rendimiento técnico.

3.9.3. Optimización de Materiales y Texturas

- **Colores planos y baja resolución:** Se utilizaron texturas de baja resolución enfocadas en colores planos.
- **Máscaras emisivas:** Los efectos de neón y brillo en las plataformas y en el traje del personaje se consiguieron mediante texturas emisivas utilizadas como máscaras, evitando la necesidad de añadir luces puntuales físicas o geometría adicional.

3.9.4. Optimización de Animaciones y Físicas

- **Animaciones de texturas (*UV Scrolling*):** Para los elementos dinámicos secundarios como las barras luminosas y las plataformas con flechas se evitaron animaciones geométricas pesadas, implementando en su lugar un desplazamiento cíclico de coordenadas UV.
- **Colisionadores optimizados:** Se priorizó el uso de colisionadores primitivos y sencillos (*Box Colliders*), reservando los colisionadores de malla (*Mesh Colliders*)

personalizados únicamente para plataformas con geometrías irregulares complejas, lo que redujo drásticamente el costo de procesamiento físico del motor.

3.10. Pruebas de rendimiento en motor grafico – Unity Profiler

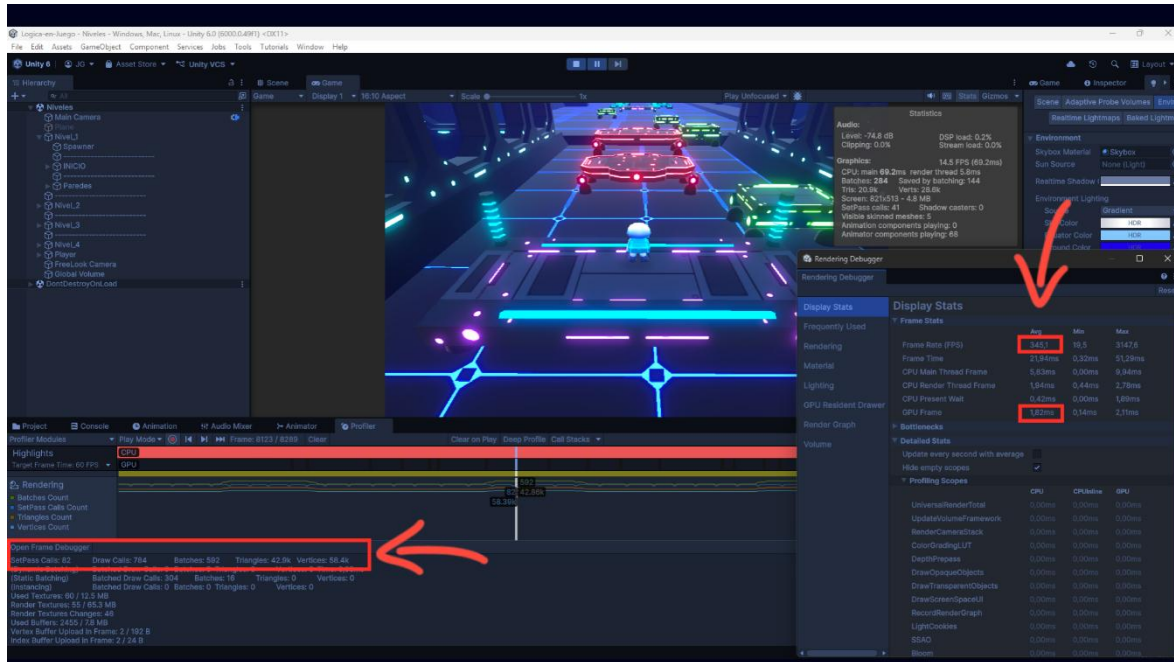
La renderización del escenario y del personaje vista a través de la cámara principal se ejecuta de forma sumamente fluida, alcanzando una velocidad promedio de hasta 345 FPS con un tiempo de renderizado de la GPU de tan solo 1.82 milisegundos (ms), lo que supera con creces el umbral mínimo de 60 FPS (16.6 ms) necesario para garantizar la máxima respuesta interactiva en PC. En cuanto a las estadísticas de renderizado, la escena registra alrededor de 42.9 mil triángulos, 58.4 mil vértices, 592 batches, 784 draw calls y 82 SetPass Calls. Según las recomendaciones no oficiales utilizadas habitualmente para proyectos de PC en Unity, mantener menos de 100 mil triángulos, menos de 800 batches, menos de 1500 draw calls y alrededor de 100 SetPass Calls o menos representa un rango adecuado para conservar un buen rendimiento. Comparando estos valores con los obtenidos en la escena, se observa que todos se encuentran dentro de los límites recomendables, destacando especialmente el bajo número de triángulos y SetPass Calls, En base a esto, el escenario diseñado para la guía cumple con las condiciones esperadas para una ejecución eficiente, demostrando que tanto el personaje como el escenario están mayormente optimizados y se mantienen dentro de un rango de jugabilidad óptimo.

- Equipo de prueba (Hardware): Procesador Intel Core i7, 64 GB RAM DDR5, Tarjeta Gráfica NVIDIA GeForce RTX 4060.
- Resolución de pantalla: 1920 x 1080.
- Configuración del Motor: Unity 6, utilizado solo en la ventana Play del Editor.

- Escena evaluada: Escena principal de la estación espacial con personaje animado en movimiento y efectos de postprocesado activos.

Ilustración 27

Escena en Unity con profiler



Nota. Elaboración propia con base de Unity.

3.11. Validación técnica de recursos visuales

Para evaluar la coherencia visual de los recursos tridimensionales integrados en la guía interactiva se aplicó un estudio de tipo descriptivo, no experimental y de enfoque cuantitativo, mediante la administración de una encuesta transversal.

La población objetivo estuvo conformada por jóvenes de entre 18 a 23 años de edad. La muestra se conformó por 10 participantes, seleccionados mediante un muestreo por conveniencia, considerando como criterio de selección familiaridad previa con entornos 3D o con Unity.

El instrumento utilizado fue una escala de Likert de 5 puntos (1 = muy en desacuerdo, 5 = muy de acuerdo), compuesta por 10 ítems (P1 a P10) orientados a valorar la identificación

del personaje, la estética general, la claridad de las animaciones, la delimitación de zonas, la navegación y la calidad de texturas y colores del escenario (véase Tabla 5). El instrumento no incluyó ítems orientados a medir la comprensión de contenidos de lógica o programación; por lo tanto, por lo que los resultados deben interpretarse como una validación de la percepción visual y la usabilidad del entorno.

El procedimiento consistió en la exploración libre del entorno interactivo por parte de cada participante, seguida de la aplicación del instrumento en línea.

Tabla 5
Análisis de Datos obtenidos en la encuesta

N. de pre-gunta	Criterio Evaluado	Media ($\bar{x}$)	Mediana	Moda	Desv. Estándar (σ)	Mín.	Máx.	Rango
P1	Identificación del personaje	4.90	5.0	5	0.32	4	5	1
P2	Estética uniforme (3D)	4.90	5.0	5	0.32	4	5	1
P3	Claridad de animaciones	4.90	5.0	5	0.32	4	5	1
P4	Delimitación de zonas	4.90	5.0	5	0.32	4	5	1
P5	Guía progresiva y lógica	4.80	5.0	5	0.42	4	5	1
P6	Navegación intuitiva	4.90	5.0	5	0.32	4	5	1
P7	Iluminación y ambiente	5.00	5.0	5	0.00	5	5	0
P8	Destaque de plataformas	4.90	5.0	5	0.32	4	5	1
P9	Calidad de texturas/colores	4.70	5.0	5	0.67	3	5	2
P10	Escala y proporción	4.90	5.0	5	0.32	4	5	1

Nota. Elaboración propia

En el ámbito de la homogeneidad estética y el diseño mallas tridimensionales, los resultados reflejan una valoración sumamente alta, alcanzando un promedio de 4.90 sobre 5.00, una moda de 5 y un rango reducido de 1 (puntuaciones entre 4 y 5). El 100 % de los encuestados coincidió en que los assets gráficos conservan un diseño y una calidad poligonal equilibrada, lo que valida que la aplicación del estilo Low Poly se mantuvo consistente entre el personaje jugable, las plataformas y los elementos decorativos del escenario sin generar rupturas visuales.

En lo respectivo a la fidelidad del texturizado y la aplicación de color sobre los modelos, se obtuvo una media de 4.70 sobre 5.00, con una moda de 5 y un rango de 2 (puntuaciones entre 3 y 5), registrando un 90 % de valoraciones altamente positivas. Esta puntuación confirma que el trabajo de desplegado de coordenadas UV en Blender y la elaboración de mapas cromáticos en Krita previnieron en gran parte la presencia de distorsiones, pixelares o estiramientos en la geometría, asegurando acabados limpios en el motor de juego.

En conjunto, la dimensión global de modelado 3D y coherencia estética consolida un rendimiento promedio de 4.80 sobre 5.00. La ausencia total de calificaciones deficientes (valores 1 y 2) demuestra que el pipeline de producción gráfica logró proyectar una identidad visual estilizada y funcional, donde la simplicidad geométrica se alinea perfectamente con la calidad de las texturas aplicadas.

3.12. Conclusiones y recomendaciones

3.12.1. Conclusiones

Se logró diseñar e implementar una línea gráfica coherente bajo la técnica Low Poly y colores planos, la cual fue evaluada positivamente por los usuarios en términos de percepción visual, estética futurista y contraste cromático.

La estructuración modular de plataformas y la aplicación de máscaras emisivas permitieron construir un entorno espacial visualmente atractivo, garantizando la claridad visual en la diferenciación de zonas transitables y obstáculos.

La integración de mallas optimizadas, rigging simplificado y materiales reutilizables en Unity permitió un rendimiento técnico fluido, respaldado por las métricas del Profiler con tasas de fotogramas estables y bajo número de llamadas de renderizado (draw calls).

La evaluación aplicada a la muestra de usuarios del público objetivo confirma una alta aceptación de la legibilidad, la calidad del modelado y la fluidez de navegación dentro de la guía interactiva.

3.12.2. Recomendaciones

Para futuras fases del proyecto, se recomienda ampliar la biblioteca de activos modulares siguiendo estrictamente la arquitectura de escenarios definida, priorizando la diversificación de elementos interactivos mientras se respetan los presupuestos poligonales establecidos.

Es aconsejable, además, continuar iterando sobre el flujo de iluminación, explorando técnicas de renderizado avanzado que se ajusten a los estándares de rendimiento ya validados, con el fin de aumentar la inmersión visual sin comprometer la estabilidad del sistema ni la tasa de fotogramas por segundo.

Finalmente, se recomienda realizar futuras validaciones con muestras de participantes más amplias y diversas, permitiendo llevar a cabo análisis estadísticos más profundos que permitan perfeccionar las métricas de usabilidad y la eficacia del entorno a largo plazo.

Referencias

- [1] L. Castillo-Salvatierra, J. Cárdenas-Cobo, C. de la Fuente-Burdiles y C. Vidal-Silva, «Programming competencies in university...,» *Frontiers in Education*, vol. 10, 2025.
- [2] G. Koulaxidis y S. Xinogalos, «Improving mobile game performance with basic optimization techniques in Unity,» *Modelling*, vol. 3, nº 2, 2022.
- [3] Á. Villagómez-Palacios, C. De la Fuente-Burdiles y C. Vidal-Silva, «Videogame Programming & Education: Enhancing Programming Skills Through Unity Visual Scripting,» *Computers*, vol. 15, nº 68, 2026.
- [4] Unity Technologies, «Animation State Machines,» [En línea]. Available: <https://docs.unity3d.com/Manual/class-State.html>. [Último acceso: 25 Julio 2026].
- [5] Unity Technologies, «Shaders,» [En línea]. Available: <https://docs.unity3d.com/Manual/Shaders.html>. [Último acceso: 25 Julio 2026].
- [6] Unity Technologies, «Lighting Overview,» [En línea]. Available: <https://docs.unity3d.com/Manual/LightingOverview.html>. [Último acceso: 26 Julio 2026].
- [7] J. Sweller, *Cognitive Load Theory*, New York: Springer, 2011.
- [8] R. E. Mayer, *Multimedia Learning*, Cambridge: Cambridge University Press, 2009.
- [9] N. Donald A, *The Design of Everyday Things*, New York: Basic Books, 2013.
- [1] C. Meier, I. Sanchez Berriel y F. Pérez Nava, «Creation of a Virtual Museum for the
- 0] Dissemination of 3D Models of Historical Clothing,» *Sustainability*, vol. 13, nº 22, 2021.
- [1] P. J. Ewell, «Changing the Game: Understanding the Role of Social Influence in
- 1] Videogames to Generate New Ideas and Answer Old Questions,» *Social and Personality Psychology Compass*, vol. 20, 2026.
- [1] L. Wilkins, A. D. Gorman, S. Merry y A. Kittel, «Comparative effects of VR and
- 2] traditional...,» *Virtual Reality*, vol. 30, nº 58, 2026.
- [1] Adobe Substance 3D, «How to create low-poly 3D models and characters,» [En línea].
- 3] Available: <https://www.adobe.com/products/substance3d/discover/how-to-create-low-poly-3d-models-and-characters.html>. [Último acceso: 26 Julio 2026].
- [1] Blender Foundation, «Introduction to Meshes,» [En línea]. Available:
- 4] <https://docs.blender.org/manual/es/latest/modeling/meshes/introduction.html>. [Último acceso: 28 Julio 2026].
- [1] Unity Technologies, «Unity 6.5 User Manual,» [En línea]. Available:
- 5] <https://docs.unity3d.com/Manual/analysis.html>. [Último acceso: 25 Julio 2026].
- [1] Blender Foundation, «Blender 5.2 LTS Manual Introduction,» [En línea]. Available:
- 6] https://docs.blender.org/manual/en/latest/sculpt_paint/texture_paint/introduction.html. [Último acceso: 25 Julio 2026].
- [1] Blender Foundation, «Blender 5.2 LTS Manual Using UV Maps,» [En línea]. Available:
- 7] https://docs.blender.org/manual/en/latest/modeling/meshes/uv/applying_image.html. [Último acceso: 25 Julio 2026].
- [1] Blender Foundation, «Blender 5.2 LTS Manual UV Overlays,» [En línea]. Available:
- 8] <https://docs.blender.org/manual/es/latest/editors/uv/overlays.html>. [Último acceso: 26 Julio 2026].

- [1 Unity Technologies, «Unity Manual Physically Based Rendering Material Validator,» [En línea]. Available: <https://docs.unity3d.com/es/2018.4/Manual/MaterialValidator.html>. [Último acceso: 24 Julio 2026].
- [2 Unity Technologies, «Unity 6.5 User Manual Material Inspector window reference,» [En línea]. Available: <https://docs.unity3d.com/Manual/class-Material.html>. [Último acceso: 24 Julio 2026].
- [2 Unity Technologies, «Unity 6.5 User Manual Custom shaders,» [En línea]. Available: <https://docs.unity3d.com/Manual/Shaders.html>. [Último acceso: 25 Julio 2026].
- [2 Unity Technologies, «Unity 6.2 User Manual Lightmapping settings in the Lighting Settings Asset reference,» [En línea]. Available: <https://docs.unity3d.com/Manual/Lightmaps-reference.html>. [Último acceso: 25 Julio 2026].
- [2 Foundation, Blender, «Animation & Rigging - Introduction,» [En línea]. Available: <https://docs.blender.org/manual/en/latest/animation/introduction.html>. [Último acceso: 26 Julio 2026].
- [2 Unity Technologies, «Unity - Manual: Using Humanoid Characters,» [En línea]. Available: <https://docs.unity3d.com/Manual/UsingHumanoidChars.html>. [Último acceso: 25 Julio 2026].
- [2 Foundation, Blender, «Armatures - Introduction,» [En línea]. Available: <https://docs.blender.org/manual/en/latest/animation/armatures/introduction.html>. [Último acceso: 26 Julio 2026].
- [2 Unity Technologies, «Unity - Manual: Importing a model with humanoid animations,» [En línea]. Available: <https://docs.unity3d.com/Manual/ConfiguringtheAvatar.html>. [Último acceso: 25 Julio 2026].
- [2 Foundation, Blender, «Weight paint - Introduction,» [En línea]. Available: https://docs.blender.org/manual/en/latest/sculpt_paint/weight_paint/introduction.html. [Último acceso: 26 Julio 2026].
- [2 Foundation, Blender, «Editing Weight Paint,» [En línea]. Available: https://docs.blender.org/manual/en/latest/sculpt_paint/weight_paint/editing.html. [Último acceso: 27 Julio 2026].
- [2 Blender Documentation Team, «Using Vertex Groups,» [En línea]. Available: https://docs.blender.org/manual/en/latest/sculpt_paint/weight_paint/usage.html. [Último acceso: 26 Julio 2026].
- [3 Blender Foundation, «Editing Keyframes,» [En línea]. Available: <https://docs.blender.org/manual/en/latest/animation/keyframes/editing.html>. [Último acceso: 27 Julio 2026].
- [3 Blender Foundation, «Animation & Rigging - Introduction,» 2026. [En línea]. Available: <https://docs.blender.org/manual/en/latest/animation/introduction.html>. [Último acceso: 27 Julio 2026].
- [3 Unity Technologies, «Unity - Manual: Introduction to animation clips,» [En línea]. Available: <https://docs.unity3d.com/Manual/AnimationClips.html>. [Último acceso: 25 Julio 2026].

- [3] Unity Technologies, «Unity - Manual: Introduction to Animator Controllers,» [En línea].
- 3] Available: <https://docs.unity3d.com/Manual/class-AnimatorController.html>. [Último acceso: 25 Julio 2026].
- [3] J. Martins, «Qué es Scrum: marco ágil, sprints, roles y artefactos,» Asana, Inc., 09 Abril
- 4] 2026. [En línea]. Available: <https://asana.com/es/resources/what-is-scrum>. [Último acceso: 28 Julio 2026].
- [3] S. I. M. y. P. L. Alfonso, «Implementación de SCRUM en el diseño del,» *Scientia et Technica*, vol. vol. 19, nº 4, 2014.
- [3] K. Schwaber y . J. Sutherland, «The Scrum Guide: The Definitive Guide to Scrum: The
- 6] Rules of the Game,» Noviembre 2020. [En línea]. Available: <https://scrumguides.org>. [Último acceso: 29 Julio 2026].
- [3] G. Cifuentes, «Scrum desde cero: conceptos, roles, artefactos y eventos que debes
- 7] conocer,» 15 Julio 2016. [En línea]. Available: <https://www.giovannycifuentes.com/scrum/>. [Último acceso: 29 Julio 2026].
- [3] S. Laoyan, «Qué es la metodología waterfall y cuándo utilizarla,» Asana, Inc., 06 Febrero
- 8] 2026. [En línea]. Available: <https://asana.com/es/resources/waterfall-project-management-methodology?exp=mm>. [Último acceso: 29 Julio 2026].
- [3] SafetyCulture Content Team, «Metodología en cascada: Ventajas e inconvenientes,»
- 9] SafetyCulture, 28 Abril 2026. [En línea]. Available: <https://safetyculture.com/es/temas/metodologia-en-cascada>. [Último acceso: 29 Julio 2026].
- [4] El Ecosistema Startup, «Lean Startup vs Design Thinking: Dos Enfoques para Crear
- 0] Productos,» [En línea]. Available: <https://ecosistemastartup.com/comparaciones/lean-startup-vs-design-thinking/>. [Último acceso: 28 Julio 2026].
- [4] A. Mogollón, «Design Thinking aplicado en lo público: El pensamiento de Diseño para el
- 1] impulso de soluciones centradas en las personas: dos casos desde lo público,» *RChD: creación y pensamiento*, vol. 6, nº 11, 2021.
- [4] Design Thinking España, «Herramientas de Design Thinking,» [En línea]. Available:
- 2] <https://designthinkingespaña.com/herramientas-de-design-thinking>. [Último acceso: 28 Julio 2026].
- [4] Blender Foundation, «Acerca de Blender - Blender 5.2 LTS Manual,» 2026. [En línea].
- 3] Available: https://docs.blender.org/manual/es/latest/getting_started/about/index.html. [Último acceso: 26 Julio 2026].
- [4] Blender Foundation, «Introduction - Blender 5.2 LTS Manual,» [En línea]. Available:
- 4] <https://docs.blender.org/manual/en/latest/modeling/modifiers/introduction.html>. [Último acceso: 27 Julio 2026].
- [4] Autodesk Inc., «Maya Getting Started: Introduction,» [En línea]. Available:
- 5] https://download.autodesk.com/us/maya/Maya2013_Getting_Started/index.html. [Último acceso: 27 Julio 2026].
- [4] Autodesk Inc., «Maya User's Guide: Retargeting character animation,» [En línea].
- 6] Available: https://download.autodesk.com/global/docs/maya2014/en_us/index.html?url=files/GUID-B2E9F8B9-F188-4E27-8820-BB0C8541701F.htm,topicNumber=d30e301216. [Último acceso: 26 Julio 2026].

- [4 Autodesk Inc., «Maya User's Guide: Introduction to polygons,» [En línea]. Available:
7] https://download.autodesk.com/global/docs/maya2014/en_us/index.html?url=files/Polygons_overview_Introduction_to_polygons.htm,topicNumber=d30e151211. [Último acceso: 26 Julio 2026].
- [4 Autodesk Inc., «Maya User's Guide: UV Texture Editor overview,» [En línea]. Available:
8] https://download.autodesk.com/global/docs/maya2014/en_us/index.html?url=files/GUID-73BF7546-3BF2-44F3-9192-15A024CEC173.htm,topicNumber=d30e219740. [Último acceso: 26 Julio 2026].
- [4 Autodesk Inc., «Maya Help - Retiming animation in the Time Editor,» [En línea].
9] Available: <https://help.autodesk.com/view/MAYAUL/2024/ENU/?guid=GUID-164A97B6-D316-4D3F-AF6B-E7BAE8F3AE11>. [Último acceso: 27 Julio 2026].
- [5 Maxon Computer, «Meet ZBrush!,» [En línea]. Available: <https://help.maxon.net/zbr/en-us/Content/html/features/features.html>. [Último acceso: 27 Julio 2026].
- [5 Maxon Computer, «Sculpting,» [En línea]. Available: <https://help.maxon.net/zbr/en-us/Content/html/user-guide/3d-modeling/sculpting/sculpting.html>. [Último acceso: 27 Julio 2026].
- [5 Maxon Computer, «DynaMesh,» [En línea]. Available: <https://help.maxon.net/zbr/en-us/Content/html/user-guide/3d-modeling/modeling-basics/creating-meshes/dynamesh/dynamesh.html>. [Último acceso: 27 Julio 2026].
- [5 Maxon Computer, «ZRemesher,» [En línea]. Available: <https://help.maxon.net/zbr/en-us/Content/html/user-guide/3d-modeling/topology/zremesher/zremesher.html>. [Último acceso: 26 Julio 2026].
- [5 Maxon Computer, «Polypaint,» [En línea]. Available: <https://help.maxon.net/zbr/en-us/Content/html/user-guide/3d-modeling/painting-your-model/polypaint/polypaint.html>. [Último acceso: 26 Julio 2026].
- [5 Maxon Computer, «Subtool,» [En línea]. Available: <https://help.maxon.net/zbr/en-us/Content/html/zbrushcore/reference-guide/tool/subtool/subtool.html>. [Último acceso: 26 Julio 2026].
- [5 Maxon Computer, «Basic Process,» [En línea]. Available: <https://help.maxon.net/zbr/en-us/Content/html/user-guide/3d-modeling/modeling-basics/creating-meshes/live-boolean/basic-process/basic-process.html>. [Último acceso: 27 Julio 2026].
- [5 Krita Foundation, «Welcome to the Krita 5.3 Manual!,» [En línea]. Available:
7] <https://docs.krita.org/en/>. [Último acceso: 25 Julio 2026].
- [5 Krita Foundation, «Introduction to Krita coming from Paint Tool SAI,» [En línea].
8] Available: https://docs.krita.org/en/user_manual/introduction_from_other_software/introduction_from_sai.html. [Último acceso: 27 Julio 2026].
- [5 Krita Foundation, «Krita 4 Preset Bundle Overview,» [En línea]. Available:
9] https://docs.krita.org/en/reference_manual/krita_4_preset_bundle.html. [Último acceso: 27 Julio 2026].
- [6 Krita Foundation, «Freehand Brush Tool,» [En línea]. Available:
0] https://docs.krita.org/en/reference_manual/tools/freehand_brush.html. [Último acceso: 27 Julio 2026].

- [6 Adobe Inc., «How to use Photoshop for beginners - learn the basics,» [En línea]. Available:
1] <https://www.adobe.com/products/photoshop/how-to-use.html>. [Último acceso: 26 Julio 2026].
- [6 Adobe Inc., «Understand Photoshop color management,» 24 Mayo 2023. [En línea].
2] Available: <https://helpx.adobe.com/photoshop/using/understanding-color-management.html>. [Último acceso: 25 Julio 2026].
- [6 Adobe Inc., «Add and edit layer masks in Photoshop,» 23 Febrero 2026. [En línea].
3] Available: <https://helpx.adobe.com/photoshop/desktop/create-masks/layer-masks/add-layer-masks.html>. [Último acceso: 27 Julio 2026].
- [6 Adobe Inc., «Photoshop channel basics,» 24 Mayo 2023. [En línea]. Available:
4] <https://helpx.adobe.com/photoshop/using/channel-basics.html>. [Último acceso: 26 Julio 2026].
- [6 Adobe Inc., «Introducción a los pinceles de Photoshop,» [En línea]. Available:
5] <https://www.adobe.com/es/products/photoshop/photoshop-brushes.html>. [Último acceso: 26 Julio 2026].
- [6 C. Monkey, «Animation System in Unity DOTS! (15 NEW Lectures),» Endless Loop
6] Studios, 2025. [En línea]. Available:
<https://unitycodemonkey.com/text.php?v=P01egjRl2cs>. [Último acceso: 04 Agosto 2026].
- [6 Unity Technologies, Optimize Your Game Performance for Consoles and PC, Unity
7] Technologies, 2024.
- [6 GarageFarm.NET, «A Comprehensive Retopology Guide to 3D Model Optimization,»
8] 2024. [En línea]. Available: <https://garagefarm.net/blog/retopology-a-comprehensive-guide-to-3d-model-optimization>. [Último acceso: 04 Agosto 2026].
- [6 V. Bonenkamp, «Blender to Unity Pipeline: FBX Export Best Practices | Blended BORIS |
9] Digital Art IP,» 20 Marzo 2026. [En línea]. Available: <https://blendedboris.com/our-blog/tpost/blender-to-unity-pipeline>. [Último acceso: 05 Agosto 2026].
- [7 J. L. Burns, «Defining the Modeling Standard for 3D Character Artists,» East Tennessee
0] State University, Johnson City, 2015.
- [7 Cluster Official, «Configuraciones recomendadas para exportar modelos de Blender a
1] Unity,» 13 Octubre 2024. [En línea]. Available:
https://medium.com/@cluster_official/recommended-settings-for-exporting-models-from-blender-into-unity-77e3e1fb3c8d. [Último acceso: 05 Agosto 2026].
- [7 Blender Foundation, «Introduction to Meshes,» 2025. [En línea]. Available:
2] <https://docs.blender.org/manual/es/latest/modeling/meshes/introduction.html>. [Último acceso: 5 Abril 2026].
- [7 M. Á. Roque López, «Entendiendo la programación visual...,» *Con A de animación*, nº 8,
3] pp. 92-101, 2018.
- [7 Adobe Substance 3D, «How to create low-poly 3D models and characters,» 2025. [En
4] línea]. Available: <https://www.adobe.com/products/substance3d/discover/how-to-create-low-poly-3d-models-and-characters.html>. [Último acceso: 4 Abril 2025].
- [7 M. Galán Puerto, «Producción de un escenario óptimo para videojuegos 3D,» Universitat
5] Politècnica de València, España, 2021.

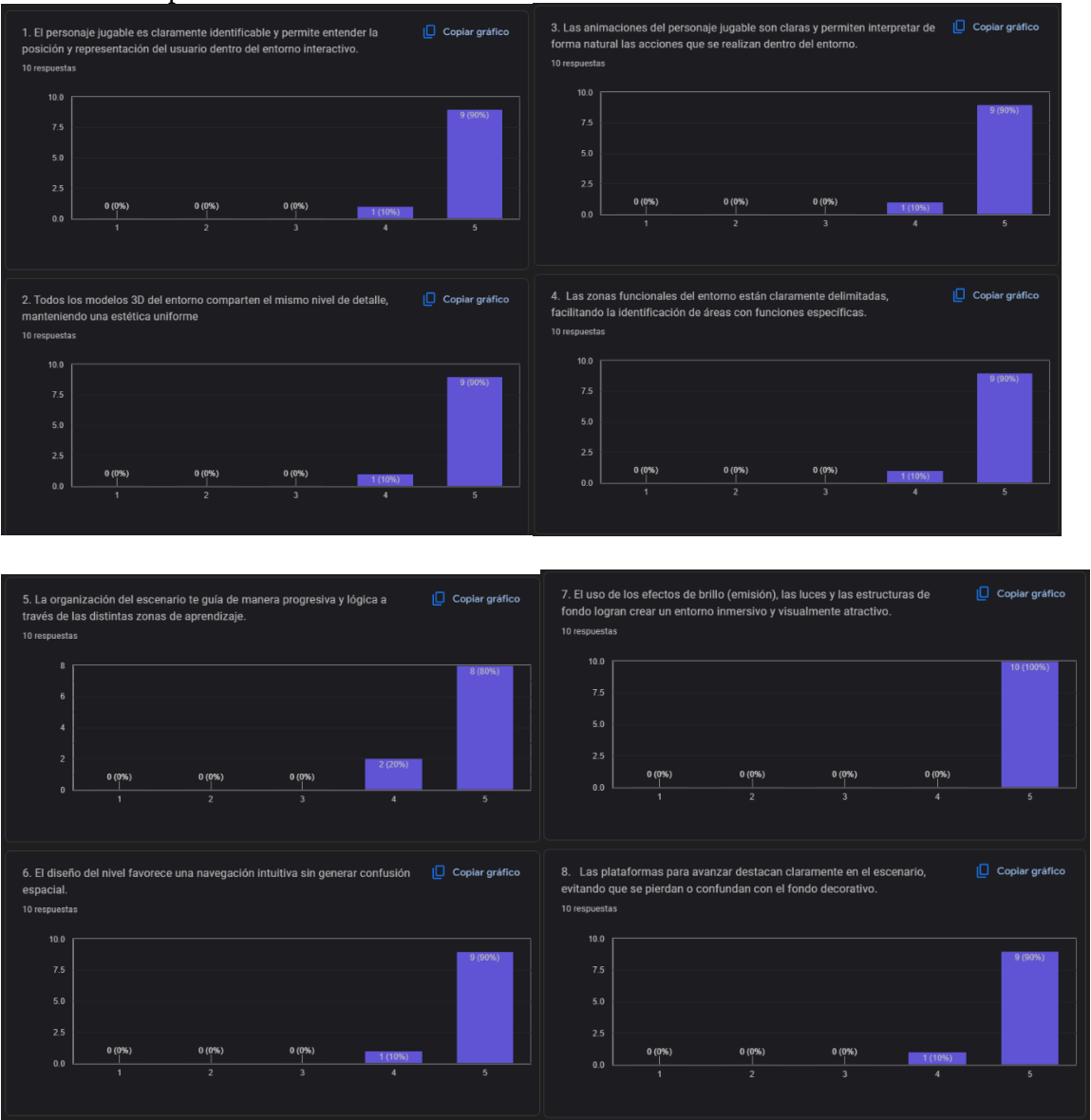
- [7] Unity Technologies, «Working with GameObjects,» 2025. [En línea]. Available:
6] <https://docs.unity3d.com/Manual/working-with-gameobjects.html>. [Último acceso: 3 Abril 2026].
- [7] Animum Creativity Advanced School, «¿Qué es el Low Poly?,» 2025. [En línea].
7] Available: <https://www.animum3d.com/blog/que-es-low-poly/>. [Último acceso: 4 Abril 2026].
- [7] Unity Technologies, «Optimizing graphics performance,» 2025. [En línea]. Available:
8] <https://docs.unity3d.com/Manual/lighting-optimize-builtin.html>. [Último acceso: 2 Abril 2026].
- [7] Blender Foundation, «UV Editing,» 2025. [En línea]. Available:
9] <https://docs.blender.org/manual/es/latest/editors/uv/index.html>. [Último acceso: 2 Abril 2026].
- [8] Blender Foundation, «Armatures Introduction,» 2025. [En línea]. Available:
0] <https://docs.blender.org/manual/es/latest/animation/armatures/introduction.html>. [Último acceso: 3 Abril 2026].
- [8] Unity Technologies, «Animation Section & Animator Controllers,» 2025. [En línea].
1] Available: <https://docs.unity3d.com/Manual/AnimationSection.html>. [Último acceso: 2 Abril 2026].
- [8] M. F. G. Trischler, «backend.orbit.dtu.dk,» 06 2022. [En línea]. Available:
2] <https://backend.orbit.dtu.dk/ws/portalfiles/portal/293194740/Thesis.pdf>.
- [8] K. P. Liu, K. W. Chung, W. Chiu y G. Chen, «La transformación digital impulsa la
3] innovación del modelo de negocio de las PYME,» *Revista de Gestión de la Información Global*, pp. 1-23, 2024.
- [8] M. Riesener, M. Kuhn, S. Schümmelfeder y G. Schuh, «Business Model Research in the
4] Age of Digitalization: A Systematic Literature Research. Journal of Economics, Business and Management,» *Journal of Economics, Business and Management*, pp. 157-168, 2023.
- [8] W. . R. Trujillo Jiménez y R. J. Suárez Venegas, «Repartidores de delivery y plataformas
5] digitales en Ecuador ¿Relación laboral o prestación autónoma de servicios?,» *Arandu UTIC*, pp. 853-868, 2025.
- [8] N. J. Foss y T. Saebi, «Modelos de negocio e innovación de modelos de negocio: entre
6] problemas perversos y paradigmáticos,» *Long Range Planning*, pp. 9-21, 2018.
- [8] . C. Ancillai, A. Sabatini, M. Gatti y A. Perna, «Tecnología digital e innovación en
7] modelos de negocio: una revisión sistemática de la literatura y una agenda de investigación futura,» *Elsevier*, 2023.
- [8] J. J. Perez Muzha, «cia.uagraria.edu.ec,» 2022. [En línea]. Available:
8] <https://cia.uagraria.edu.ec/Archivos/P%C3%89REZ%20MUZHA%20JESSICA%20YESENIA.PDF>.
- [8] J. E. Yanez Rodriguez, «repositorio.tecnologicopichincha.edu.ec,» 2023. [En línea].
9] Available:
<https://repositorio.tecnologicopichincha.edu.ec/bitstream/123456789/973/1/Yanez%20Rodriguez%20Julia%20Elena.pdf>.
- [9] H. S. Perez Vallejos, «dspace.udla.edu.ec,» 2023. [En línea]. Available:
0] <https://dspace.udla.edu.ec/bitstream/33000/14805/1/UDLA-EC-TMAE-2023-11.pdf>.

- [9 L. F. Escobar Montenegro, «repositorio.uta.edu.ec,» 03 2022. [En línea]. Available:
1] <https://repositorio.uta.edu.ec/server/api/core/bitstreams/10b8c14b-0726-4e39-9df1-242f8c1021f3/content>.
- [9 J. A. Bravo Macias, «repositorio.unesum.edu.ec,» 2024. [En línea]. Available:
2] <https://repositorio.unesum.edu.ec/bitstream/53000/7001/1/Bravo%20Macias%20Jos%C3%A9%20Alberto.pdf>.
- [9 A. J. Matute Rosales, E. Zuniga Mejia y C. E. Cotto Sanchez, «repositorio.unitec.edu,»
3] 2020. [En línea]. Available:
<https://repositorio.unitec.edu/server/api/core/bitstreams/07c7763f-1fbb-41ef-b71a-2d97d39e6e75/content>.
- [9 O. J. Alvarez Bautista , «repositorio.uta.edu.ec,» 04 2023. [En línea]. Available:
4] <https://repositorio.uta.edu.ec/server/api/core/bitstreams/b9192b83-c7b4-4855-a9c9-57efd678724b/content>.
- [9 D. I. Perero Saltos, «repositorio.unesum.edu.ec,» 2024. [En línea]. Available:
5] <https://repositorio.unesum.edu.ec/bitstream/53000/6452/1/Perero%20Saltos%20Darwin%20Israel.pdf>.
- [9 K. D. Pincay Holguin y I. G. Sanchez Balon, «cia.uagraria.edu.ec,» 2022. [En línea].
6] Available:
<https://cia.uagraria.edu.ec/Archivos/S%C3%81NCHEZ%20BAL%C3%93N%20INGRID%20GARDENIA.pdf>.
- [9 G. M. Mantilla Buenaño, «repositorio.ulvr.edu.ec,» 2022. [En línea]. Available:
7] <http://repositorio.ulvr.edu.ec/bitstream/44000/5021/1/T-ULVR-4045.pdf>.
- [9 M. A. Gualli Tenegusniay, «cia.uagraria.edu.ec,» 2021. [En línea]. Available:
8] <https://cia.uagraria.edu.ec/Archivos/GUALLI%20TENEGUSNIAY%20MARIO%20ANIBAL.pdf>.
- [9 T. Faria de Assis, V. H. Souza de Abreu, M. Gonzalez da Costa y D. A. Marcio de
9] Almeida, «Metodología para la priorización de las mejores prácticas aplicadas a la última milla sostenible: el caso de una empresa brasileña de servicios de entrega de paquetes,» *Sustainability*, pp. 1-24, 2020.
- [1 S. N. Arcos Azuero y A. A. Astudillo Hinojosa, «dspace.ups.edu.ec,» 2024. [En línea].
00 Available: <https://dspace.ups.edu.ec/bitstream/123456789/27529/1/UPS-GT005015.pdf>.
]
- [1 G. Koulaxidis y S. Xinogalos, «Improving Mobile Game Performance...,» *Modelling*, vol.
01 3, pp. 201-223, 2022.
]
- [1 Nerd for Tech (Medium), «Low-poly modelling in Blender,» 2024. [En línea]. Available:
02 <https://medium.com/nerd-for-tech/low-poly-modelling-in-blender-aaf132ef3124>. [Último
] acceso: 3 Abril 2026].
- [1 Unity Technologies, «Animation State Machines,» 2025. [En línea]. Available:
03 <https://docs.unity3d.com/Manual/class-State.html>. [Último acceso: 3 Abril 2026].
]
- [1 Unity Technologies, «Shaders,» 2025. [En línea]. Available:
04 <https://docs.unity3d.com/Manual/Shaders.html>. [Último acceso: 4 Abril 2026].
]

[1 Unity Technologies, «Lighting Overview,» 2025. [En línea]. Available:
05 <https://docs.unity3d.com/Manual/LightingOverview.html>. [Último acceso: 2 Abril 2026].
]

Anexos

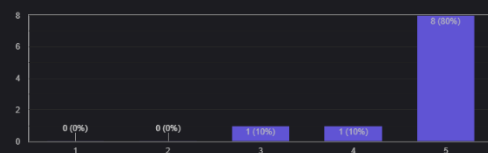
Anexo 1. Respuestas de encuesta



9. Las texturas y colores aplicados sobre las plataformas, el suelo y el personaje se visualizan de forma nítida, sin distorsiones o estiramientos en su apariencia gráfica

[Copiar gráfico](#)

10 respuestas



10. La escala general del escenario tiene una proporción lógica y equilibrada con respecto al personaje y la ambientación espacial.

[Copiar gráfico](#)

10 respuestas

