



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

Aplicación de técnicas de modelado y animación tridimensional en

Blender para el desarrollo optimizado de personajes y entornos

interactivos en tiempo real

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

AUTOR: RONNY SEBASTIAN CEVALLOS VILLAVICENCIO

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

Aplicación de técnicas de modelado y animación tridimensional en Blender
para el desarrollo optimizado de personajes y entornos interactivos en tiempo
real

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

AUTOR: RONNY SEBASTIÁN CEVALLOS VILLAVICENCIO

DIRECTOR: ING. JHEYSON STEVEN GAONA PINEDA, MTR.

CUENCA - ECUADOR

2026

DIOS, PATRIA, CULTURA Y DESARROLL



Universidad
Católica
de Cuenca

DECLARATORIA DE AUTORÍA Y RESPONSABILIDAD

Declaratoria de Autoría y Responsabilidad

Ronny Sebastián Cevallos Villavicencio portador de la cédula de ciudadanía N.º **0106328966**.

Declaro ser el autor de la obra: “Aplicación de técnicas de modelado y animación tridimensional en Blender para el desarrollo optimizado de personajes y entornos interactivos en tiempo real.”, 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, **1 de septiembre de 2026**

F: 

Ronny Sebastián Cevallos Villavicencio

C.I. 0106328966

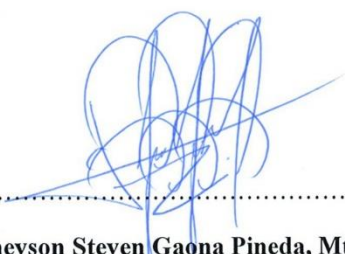


Universidad
Católica
de Cuenca

CERTIFICADO

Certificado

Certifico que el presente Trabajo de Investigación fue desarrollado por **Ronny Sebastián Cevallos Villavicencio** portador(a) de la cédula de ciudadanía N.º **0106328966** con el tema **"Aplicación de técnicas de modelado y animación tridimensional en Blender para el desarrollo optimizado de personajes y entornos interactivos en tiempo real"**, bajo mi supervisión.

F: 

Ing. Jheyson Steven Gaona Pineda, Mtr.

Docente - Tutor

Dedicatoria

Dedico este trabajo a las personas que han sido parte fundamental en mi vida y que me han acompañado durante este camino. Este logro no solamente finaliza una etapa universitaria, sino también el resultado de todo el apoyo, esfuerzo y confianza que recibí durante estos años.

A mis padres Eva y Luis, por creer en mí incluso en aquellos momentos en los que yo mismo llegué a dudar. Gracias por no dejarme caer, por hacer todo lo posible para ayudarme y por estar presentes en todo momento, tanto en los buenos y malos momentos. Gracias por su amor, sacrificio y apoyo incondicional, que fueron unas de las principales razones para seguir adelante y no rendirme.

A mis abuelitos Isaura y Raúl, por acompañarme con sus consejos, cariño y enseñanzas. Cada palabra de ustedes me ayudó a crecer no solo como estudiante, sino también como persona. Gracias por estar presentes en este camino y formar parte de los recuerdos que siempre llevaré conmigo.

A mis hermanos Jairo y Santiago, por estar a mi lado durante esta etapa, por su cariño, apoyo y por todos los momentos que hemos compartido. A mi hermana Andrea y a mis sobrinos Jeanpierre y Victoria, aunque la distancia no nos permita vernos tanto como quisiera, gracias por estar siempre pendientes de mí y por apoyarme desde lejos.

Finalmente, dedico este trabajo a toda mi familia y a las personas que estuvieron presentes durante esta etapa de mi vida. A quienes me brindaron una palabra o mensaje de ánimo, un consejo, una ayuda o simplemente estuvieron ahí cuando más lo necesitaba. A cada uno les agradezco por motivarme para que hoy pueda cerrar esta etapa de mi vida con orgullo.

Agradecimiento

En primer lugar, agradezco a Dios y a la Virgencita por darme la fuerza y la perseverancia necesarias para continuar durante este proceso de aprendizaje.

Al haber culminado esta etapa tan importante en mi vida, quiero agradecer a todas las personas que estuvieron a mi lado y me brindaron su apoyo para poder alcanzar este logro. Cada palabra de ánimo, consejo y ayuda que recibí durante este camino fue importante para seguir adelante y no rendirme.

Agradezco a mi tutor de tesis, Ing. Jheyson Steven Gaona Pineda, por su orientación y enseñanza durante el desarrollo de este trabajo. Sus conocimientos y recomendaciones fueron fundamentales para poder llevar este proyecto y culminar esta etapa de mi formación profesional.

Finalmente agradezco a todos los docentes universitarios que fueron parte de mi formación. Gracias por compartir sus conocimientos, experiencias y enseñanzas, que me permitieron crecer como estudiante y prepararme para los retos de la vida profesional.

Resumen

La inteligencia artificial aplicada al modelado 3D ha transformado los flujos de trabajo en el desarrollo de contenido digital, permitiendo generar mallas tridimensionales de forma automática y acelerar las etapas de la escultura digital; sin embargo, estas geometrías suelen presentar problemas de topología que hacen indispensable la intervención humana mediante procesos de retopología, donde el autor aplica su conocimiento artístico para dotar de expresividad y realismo a los personajes y objetos, evidenciando cómo la combinación entre inteligencia artificial y trabajo humano puede acelerar el desarrollo de contenido 3D sin comprometer su calidad final. El presente proyecto describe el modelado y retopología de personajes jugables y no jugables, así como la creación de recursos para el videojuego "Ashes of Faith", utilizando Blender bajo un estilo artístico low poly. Para los personajes, se toma como punto de partida mallas generadas mediante la herramienta "Tencent Hunyuan 3D", aplicando retopología para reducir y organizar la densidad poligonal manteniendo su apariencia visual, mientras que los recursos fueron modelados desde cero bajo el mismo estándar. Además, se trabajó en mapeo UV, materiales, texturas, rigging y animación para su correcta integración en Unity, todo planificado bajo la metodología ágil Scrum, permitiendo organizar el trabajo en sprints y reducir los tiempos de desarrollo. Como resultado se realizan pruebas en Unity utilizando el Profiler, midiendo el impacto de los recursos en renderización, confirmando que la inteligencia artificial actúa como acelerador inicial, mientras que la expresividad y el rendimiento de los modelos son producto del criterio humano validado técnicamente.

Palabras clave: Modelado 3D, Inteligencia Artificial, Retopología, Blender.

Abstract

Artificial intelligence applied to 3D modeling has transformed workflows in digital content development, enabling the automatic generation of three-dimensional meshes and accelerating the stages of digital sculpting; however, these geometries often present topology issues that make indispensable human intervention through retopology processes, where the author applies artistic knowledge to provide expressiveness and realism to characters and objects, demonstrating how the combination of artificial intelligence and human work can accelerate 3D content development without compromising its final quality. This project describes the modeling and retopology of playable and non-playable characters, as well as the creation of assets for the video game “Ashes of Faith”, using Blender under a low-poly artistic style. For the characters, meshes generated with the “Tencent Hunyuan 3D” tool was used as the starting point, with retopology applied to reduce and organize polygon density while maintaining their visual appearance, while the assets were modeled from scratch to the same standard. In addition, work was carried out on UV mapping, materials, textures, rigging, and animation for their proper integration into Unity; all activities were planned under the Scrum Agile methodology, allowing the work to be organized into sprints and reducing development time. As a result, tests were conducted in Unity using the Profiler to measure the impact of the assets on rendering, confirming that artificial intelligence acts as an initial accelerator, while the expressiveness and performance of the models are the product of technically validated human judgment.

Keywords: *3D modeling, Artificial Intelligence, Retopology, Blender*

Contenido

Declaratoria De Autoría y Responsabilidad.....	II
Certificación.....	III
Dedicatoria.....	IV
Agradecimiento.....	V
Resumen.....	VI
Índice de tablas.....	X
Índice de ilustraciones.....	XI
Introducción.....	1
Capítulo I.....	2
1. Marco referencial.....	2
1.1. Contextualización del problema.....	2
1.4. Justificación.....	5
1.5. Objetivos General.....	7
1.6. Objetivo Específico.....	7
1.7. Alcance del proyecto.....	7
1.10. Metodología general.....	9
Capítulo II.....	14
2. Marco teórico.....	14
2.1. Antecedentes de investigación.....	14
2.1.1. The Legend of Zelda: Breath of the Wild.....	14
2.1.2. Genshin Impact.....	16
2.1.3. Dark Souls III.....	17
2.1.4. Cuadro comparativo de trabajos relacionados.....	19
2.2. Bases teóricas.....	20
2.2.1. Modelado 3D en el desarrollo de videojuegos.....	20
2.2.2. Técnicas de modelado para personajes y entornos interactivos.....	21
2.2.3. Texturizado y materiales en modelos 3D.....	22
2.2.4. Rigging y animación para videojuegos.....	24
2.2.5. Optimización de modelos 3D para motores gráficos en tiempo real.....	26
2.3. Bases metodológicas.....	27
2.3.1. Scrum.....	28
2.3.2. Kanban.....	29
2.3.3. Extreme Programming (XP).....	31
2.3.4. Cuadro comparativo de Metodologías.....	33
2.4.1.1. Blender.....	35
2.4.1.2. Autodesk Maya.....	37
2.4.1.3. Autodesk 3ds Max.....	39

2.4.2.	Herramientas para texturizado	40
2.4.2.1.	Adobe Photoshop.....	40
2.4.2.2.	Krita	42
2.4.2.3.	Adobe Substance 3D Painter	43
2.5.	Arquitecturas, modelos y estándares.....	44
2.5.1.	Arquitectura de personajes.....	44
2.5.2.	Arquitectura de escenarios	45
2.5.3.	Modelos basados en referencias.....	46
2.5.4.	Modelo de validación técnica	47
2.5.5.	Estándar de escala y unidades	48
2.5.6.	Estándar de animación.....	49
2.5.7.	Cuadro comparativo de arquitecturas, modelos y estándares.....	50
2.6.	Herramientas de desarrollo.....	51
2.6.1.	Cuadro comparativo de Tecnologías.....	51
2.6.2.	Cuadro comparativo de Herramientas de texturizado.....	53
2.7.	Marco conceptual.....	54
2.8.	Relación entre la teoría y la propuesta tecnológica.....	55
Capítulo III		56
3.	Manual de usuario	56
3.1.	Especificaciones de uso	56
3.2.	Configuración de Controles	56
3.3.	Interfaz de Usuario (UI) y HUD	58
3.3.1.	Menú de inicio del juego.....	58
3.3.2.	Indicadores del HUD	59
3.3.3.	Sistema de Menú de Pausa	60
3.4.	Procedimiento de Instalación	61
Capítulo IV		63
4.	Manual de programador.....	63
4.1.	Dirección de arte y propuesta visual	63
4.2.	Diseño de personajes, escenarios y elementos gráficos	64
4.3.	Producción de <i>assets</i> 3D.....	68
4.4.	Pipeline de producción artística.....	69
4.5.	Preparación y animación de personajes y objetos.....	72
4.6.	Materiales, texturas, iluminación y renderización	73
4.7.	Integración de recursos visuales	75
4.8.	Optimización gráfica	76
4.9.	Software, formatos y organización de recursos	78
4.10.	Conclusiones	79
4.11.	Recomendaciones	79

Pruebas y resultados.....	1
----------------------------------	----------

Índice de tablas

Tabla 1 Resumen comparativo de los trabajos relacionados	19
Tabla 2 Comparación de metodologías ágiles	33
Tabla 3 Comparación entre las arquitecturas, modelos y estándares.....	50
Tabla 4 Comparación de software de modelado 3D.....	51
Tabla 5 Comparación de software de texturizado.....	53
Tabla 6 Marco conceptual	54
Tabla 7 Componentes para la ejecución del videojuego.....	56
Tabla 8 Esquema de controles e interfaz de entrada.....	56
Tabla 9 Clasificación de los modelos 3D desarrollados para el videojuego	67
Tabla 10 Desarrollo del Product Backlog.....	70
Tabla 11 Desarrollo y planificación de sprints	72

Índice de ilustraciones

Ilustración 1	Presentación visual de la jugabilidad en The Legend of Zelda: Breath of the Wild	16
Ilustración 2	Presentación visual de la jugabilidad en Genshin Impact	17
Ilustración 3	Presentación visual de la jugabilidad en Dark Souls III.....	19
Ilustración 4	Ejemplo visual del concepto de los niveles de poligonaje	21
Ilustración 5	Mapeo UV (UV mapping)	24
Ilustración 6	Presentación visual del rigging	25
Ilustración 7	Optimización de modelos 3D.....	27
Ilustración 8	Flujo de trabajo de la metodología Scrum	29
Ilustración 9	Flujo de trabajo de la metodología Kanban	31
Ilustración 10	Flujo de trabajo de la metodología Extreme Programming (XP).....	33
Ilustración 11	Principales interfaces del motor gráfico de Blender	37
Ilustración 12	Principales interfaces del motor gráfico de Autodesk Maya.....	38
Ilustración 13	Principales interfaces del motor gráfico de Autodesk 3ds Max	40
Ilustración 14	Principales interfaces del motor gráfico de Adobe Photoshop.....	42
Ilustración 15	Principales interfaces del motor gráfico de Krita.....	43
Ilustración 16	Principales interfaces del motor gráfico de Adobe Substance 3D Painter	44
Ilustración 17	Ejemplo de la Arquitectura de personajes.....	45
Ilustración 18	Ejemplo de la arquitectura de escenarios	46
Ilustración 19	Ejemplos de la definición de modelos basados en referencias.....	47
Ilustración 20	Ejemplo de modelo de validación técnica.....	48
Ilustración 21	Ejemplo de definición de estándar de escala y unidades.....	49
Ilustración 22	Ejemplo de definición de estándar de animación	50
Ilustración 23	Esquema y distribución de los botones	57
Ilustración 24	Interfaz gráfica del menú de inicio.....	58
Ilustración 25	Vista de la perspectiva de cámara e indicadores del HUD (salud y cooldown)	59
Ilustración 26	Ejecución del sistema de diálogos con el NPC Brom	60
Ilustración 27	Interfaz de pausa	61
Ilustración 28	Presentación visual del diseño 2D y el modelado 3D	64
Ilustración 29	Presentación visual de los modelos 3D para el videojuego “Ashes of Faith”	66
Ilustración 30	Desarrollo del modelado 3D	69
Ilustración 31	Desarrollo de animación	73
Ilustración 32	Desarrollo del texturizado.....	75
Ilustración 33	Exportación e Integración de los modelos 3D	76
Ilustración 34	Optimización gráfica de modelos 3D.....	78
Ilustración 35	Análisis del rendimiento gráfico de los modelos 3D en Unity.....	2
Ilustración 36	Prueba de rendimiento de la escena completa del videojuego en Unity	3

Introducción

En la actualidad, la industria de los videojuegos ha adquirido gran importancia dentro del área tecnológica. En su desarrollo, el modelado 3D permite crear personajes y escenarios que forman parte de los entornos virtuales. Sin embargo, estos recursos deben mantener una buena apariencia visual y también funcionar correctamente dentro del videojuego.

El presente proyecto se enfoca en el desarrollo de recursos 3D para un videojuego, utilizando Blender para la creación de personajes, escenarios y objetos. Para ello, se aplican procesos de modelado, texturizado, *rigging*, animación y optimización de mallas, buscando mantener un estilo visual *Low Poly* y un buen rendimiento dentro de Unity.

A continuación, se presenta una breve descripción de los capítulos que conforman este trabajo de investigación:

Capítulo I: En este capítulo se desarrolla el marco referencial, donde se presenta el problema de investigación, los antecedentes, la justificación, los objetivos, el alcance y las limitaciones del proyecto.

Capítulo II: En este capítulo se desarrolla el marco teórico, donde se presentan los principales conceptos relacionados con el modelado 3D, texturizado, *rigging*, animación, optimización y herramientas utilizadas para el desarrollo de videojuegos.

Capítulo III: En este capítulo se presenta el manual de usuario, donde se explica de manera sencilla el uso y funcionamiento de los recursos desarrollados para el proyecto.

Capítulo IV: En este capítulo se describe el proceso de desarrollo de los recursos 3D para “*Ashes of Faith*”. Se detallan las etapas de modelado, texturizado, *rigging*, animación, optimización e integración de los modelos en Unity. También se presentan las pruebas realizadas para comprobar su funcionamiento y rendimiento.

Capítulo I

1. Marco referencial

En este capítulo se presenta el contexto general del proyecto y la problemática relacionada con el desarrollo de recursos 3D para videojuegos. Se describe la importancia del modelado, la animación, y la optimización de personajes, escenarios y objetos para lograr recursos que puedan funcionar correctamente en tiempo real. Además, se presenta la situación relacionada con el desarrollo del proyecto, junto con la justificación, los objetivos, el alcance y las limitaciones que orientan el desarrollo del proyecto.

1.1. Contextualización del problema

En los últimos años, la industria de los videojuegos ha tenido un crecimiento importante y se ha convertido en un área que combina diferentes tecnologías, como los gráficos por computadora, la inteligencia artificial y la animación digital. Dentro de este contexto, el modelado 3D es una herramienta importante para la creación de recursos digitales utilizados en diferentes aplicaciones graficas, ya que permite construir modelos tridimensionales a partir de bocetos e imágenes de referencia [1], [2]. De igual manera, la construcción de y procesamiento de las mallas son aspectos importantes en la creación de personajes virtuales para videojuegos [3].

Para desarrollar este tipo de contenido 3D se utilizan diferentes programas especializados que permiten realizar procesos como el modelado, la animación y el renderizado. Sin embargo, algunas de las herramientas profesionales tienen costos elevados de licencia, lo que puede dificultar su uso en proyectos académicos o independientes. Por esta razón, existen alternativas como Blender, que es un software libre y de código abierto, lo que permite rea-

lizar actividades de modelado, animación y renderizado sin la necesidad de realizar una inversión económica. A pesar de esto, el desarrollo de contenido 3D todavía presenta dificultades ya que una mala optimización puede afectar el rendimiento del sistema[4].

Dentro del desarrollo de videojuegos, la creación de personajes 3D es uno de los procesos que requiere mayor trabajo, ya que es necesario combinar el diseño visual con mayor nivel de detalle. Algunos métodos de modelado de personajes utilizan imágenes como referencia para generar modelos 3D [5]. Además, es necesario combinar el modelado, el texturizado, el *rigging* y la animación para que el personaje pueda funcionar correctamente dentro del videojuego [6].

Diversos estudios han demostrado que el modelado 3D puede utilizar referencias de arte y animación 2D para lograr diseños visualmente atractivos [6]. En el caso de los personajes humanos, el modelado requiere de mucha atención ya que la anatomía y la forma en que se deforman las diferentes partes del cuerpo, con el fin de conseguir movimientos más naturales [7]. De igual manera, la animación cumple una función importante, ya que permite dar movimiento y vida a los personajes, por lo que se deben desarrollar movimientos fluidos y coherentes [8],[9]. La generación y animación de personajes 3D continúa siendo una área de investigación en la que se desarrollan técnicas para facilitar la creación de modelos interactivos para animarlos [10], [11].

Por otra parte, los entornos virtuales requieren diferentes procesos de construcción y procesamiento de modelos 3D para obtener una representación adecuada de los espacios y objetos que los conforman. En este caso se han desarrollado técnicas para la mejora de mallas

3D, así como métodos de reconstrucción y aplicación de texturas que permiten obtener modelos con mayor calidad visual y detalle [12], [13], [14]. Para la creación de estos espacios también se pueden utilizar técnicas como la reconstrucción de modelos a partir de imágenes y el procesamiento digital, lo cual permite generar y trabajar con recursos 3D dedicados a la construcción de personajes y escenarios [15], [16] [17]. Sin embargo, los escenarios con un alto nivel de complejidad geométrica o requieren una gran cantidad de información visual, es necesario considerar técnicas de optimización que permitan reducir el consumo de recursos y mantener un funcionamiento adecuado en tiempo real [4], [18].

1.2. Planteamiento del problema

Dentro de este contexto, uno de los principales desafíos en el desarrollo de videojuegos es la dificultad para integrar correctamente los procesos técnicos y artísticos relacionados con el modelado, el texturizado, el *rigging*, la animación y la creación de entornos virtuales. Cuando estas etapas no se organizan adecuadamente, se pueden obtener modelos con una buena apariencia visual, pero que presentan problemas al momento de funcionar en tiempo real. Esto puede afectar el rendimiento del sistema, la estabilidad del motor gráfico y la experiencia del usuario [19], [4], [5]. Además, esta situación se vuelve más complicada en proyectos que necesitan optimizar constantemente sus recursos sin perder la calidad visual ni la funcionalidad.

1.3. Formulación del problema

¿Cómo influye la aplicación de técnicas de modelado y animación en Blender en el desarrollo optimizado de personajes y entornos interactivos para videojuegos en tiempo real?

1.4. Justificación

El presente trabajo de titulación busca mejorar la creación de contenido visual 3D para el videojuego “*Ashes of Faith*”, principalmente en el desarrollo de personajes, animaciones y entornos virtuales. Durante el desarrollo de un videojuego, la falta de una buena organización en la creación de los *assets* puede generar retrasos, diferencias en el estilo visual, errores al momento de integrarlos y problemas de rendimiento. Por esta razón, este proyecto busca desarrollar recursos gráficos optimizados que puedan integrarse correctamente al videojuego y que permitan mantener una buena calidad visual y un funcionamiento adecuado [6]. Además, el modelado 3D y el uso de texturas son importantes para mejorar la apariencia de los elementos dentro de las aplicaciones interactivas [2].

Desde el punto de vista técnico, este proyecto es importante ya que busca encontrar un equilibrio entre la calidad visual de los modelos y el rendimiento del videojuego en tiempo real. Para esto, se utilizarán herramientas como Blender, ya que permite crear y trabajar con modelos 3D sin la necesidad de realizar una gran inversión económica. También se aplicarán técnicas de optimización de mallas, texturizado, *rigging* y animación, ya que una mala organización de estos elementos puede afectar el rendimiento del sistema [19], [4], [5]. De esta manera, se busca reducir errores durante la integración y mejorar el funcionamiento del videojuego.

Además, durante el desarrollo del proyecto se tomó en cuenta diferentes prácticas relacionadas con la creación de contenido 3D, como la optimización de los modelos, el uso adecuado de las texturas y la correcta aplicación del *rigging* y las animaciones. Esto permitió mantener una buena calidad visual sin afectar demasiado el rendimiento del videojuego. Por

lo tanto, el proyecto responde a una necesidad específica de “*Ashes of Faith*” y también permite aplicar procesos importantes dentro del desarrollo de contenido digital.

En el ámbito académico, este trabajo permitirá aplicar los conocimientos adquiridos durante la formación profesional en áreas como el modelado 3D, texturizado, *rigging*, animación y optimización de recursos. Estas áreas se relacionan con investigaciones enfocadas en la creación y procesamiento de modelos tridimensionales [21]. Además, el desarrollo de estos recursos permitirá fortalecer la experiencia práctica en el manejo de diferentes técnicas y herramientas utilizadas en la producción de videojuegos. De igual manera, los estudios relacionados con la generación y optimización de modelos 3D [22], [17] permiten establecer una base para comprender la importancia de desarrollar recursos que, además de cumplir con aspectos visuales, puedan adaptarse a las necesidades técnicas de un videojuego. De esta forma, el proyecto permitirá poner en práctica estos conocimientos dentro de un proyecto real de desarrollo de videojuegos.

Desde una perspectiva social, los videojuegos forman parte de una industria que combina la creatividad y la tecnología, para desarrollar experiencias interactivas. La creación de entornos y recursos tridimensionales requiere procesos de reconstrucción, procesamiento de mallas y aplicación de texturas que permiten definir las características visuales de los espacios y objetos que forman parte del videojuego [12], [13], [14]. En este sentido, el desarrollo de “*Ashes of Faith*” puede contribuir a la creación de un producto digital y también al fortalecimiento de los conocimientos y habilidades relacionados con el área multimedia.

Finalmente, esta investigación de diseño multimedia, animación y desarrollo tecnológico, ya que se enfoca en la creación de recursos digitales tridimensionales para un videojuego. De esta manera, el proyecto busca combinar los conocimientos adquiridos durante la formación profesional con la creatividad, la innovación y la aplicación de técnicas de optimización en el desarrollo de contenido 3D.

1.5. Objetivos General

Desarrollar personajes y activos tridimensionales optimizados para entornos interactivos en tiempo real, mediante técnicas de modelado 3D y optimización gráfica, considerando su calidad visual y desempeño dentro de un motor de juego.

1.6. Objetivo Específico

- Analizar las técnicas de modelado 3D, la optimización de mallas y la animación digital que se pueden usar para crear personajes y objetos para entornos interactivos.
- Diseñar personajes 3D basados en criterios de geometría, topología y coherencia visual para la integración en Unity.
- Implementar los procesos de texturizado, *rigging* y optimización gráfica de los modelos 3D para mejorar su funcionalidad y rendimiento en tiempo real.
- Evaluar la calidad visual y el rendimiento de los personajes y objetos 3D en Unity analizando métricas, conteo de *batches* y fotogramas por segundo (FPS).

1.7. Alcance del proyecto

El presente proyecto está enfocado en el desarrollo de modelos 3D de personajes, escenarios y objetos con sus respectivas animaciones para ser utilizados en el videojuego “*Ashes of Faith*”. Para lograrlo se aplicaron diferentes técnicas de creación 3D, como el modelado poligonal, el mapeo UV con textura de imagen directa y la optimización de mallas.

Con el objetivo de crear recursos visuales que mantengan una buena calidad gráfica, sean coherentes con el estilo artístico del videojuego y también que pueda ejecutarse de manera efectiva dentro de un entorno interactivo en tiempo real.

También se tiene en cuenta lograr optimizar los modelos 3D reduciendo la cantidad de polígonos y organizando mallas, ya que esto ayudará a mantener un equilibrio entre la calidad visual de los modelos y el rendimiento del videojuego. De igual manera, se aplicarán técnicas de *rigging* para que los personajes puedan realizar sus animaciones correctamente, permitiendo que sus movimientos sean más naturales y facilitando su integración dentro del motor de videojuego.

Por otra parte, dentro del alcance del proyecto no se incluirá el desarrollo completo del videojuego, ya que no abarca aspectos como la programación, la inteligencia artificial, el diseño de niveles ni la implementación de mecánicas de juego. Esto se debe a que el enfoque principal está dirigido a la creación y optimización de elementos visuales en 3D. Sin embargo, se realizará una integración básica de los modelos y sus animaciones dentro de un entorno interactivo, con el fin de que funcione correctamente.

Finalmente, este proyecto se desarrolla en el área de los gráficos computacionales aplicados al desarrollo de videojuegos, enfocándose en la creación de contenido digital en 3D.

1.8. Delimitaciones

El presente proyecto se delimita al desarrollo y optimización de personajes, escenarios y objetos 3D para el videojuego, utilizando Blender como herramienta principal de modelado y animación. La investigación se centra en el proceso de producción de recursos gráficos e integración en Unity, sin abordar el desarrollo completo del videojuego, como la programación de mecánicas, inteligencia artificial o el diseño narrativo.

1.9. Beneficiarios

Los principales beneficiarios de este proyecto son el equipo de desarrollo, al contar con personajes y escenarios 3D optimizados para su integración en Unity. De manera indirecta, este trabajo también beneficia a estudiantes y desarrolladores interesados en el modelado y animación 3D con Blender, al servir como referencia para futuros proyectos, así como a los usuarios finales, quienes disfrutan de una experiencia de juego con mejor calidad visual y rendimiento.

1.10. Metodología general

El proyecto se desarrolla mediante una metodología de investigación aplicada, orientada a la creación y evaluación de recursos tridimensionales para el videojuego. Este enfoque permite utilizar los conocimientos adquiridos durante la formación académica y aplicarlos de manera práctica en los procesos de modelado 3D, texturizado, *rigging*, animación y optimización de recursos. Para ello, se realiza una revisión de trabajos relacionados y fundamentos teóricos que permitieron establecer criterios técnicos y visuales para desarrollo de los modelos 3D.

Fase de investigación y desarrollo técnico de modelos 3D: Comprende la revisión y análisis de trabajos relacionados, fundamentos teóricos y técnicas utilizadas en la creación de modelos tridimensionales para videojuegos. A partir de esta base, se determinan los criterios técnicos y visuales necesarios para el desarrollo de personajes, objetos y otros recursos del videojuego. Posteriormente, se realizan las etapas de modelado de mallas, mapeado UV, texturizado, *rigging* y animación, aplicando procesos de optimización para adaptar los recursos a las necesidades técnicas del proyecto.

Fase de integración y evaluación de los modelos desarrollados: En esta etapa se incorporan los modelos 3D y sus animaciones en el motor Unity, donde se realizan pruebas

de funcionamiento e integración. Estas pruebas permiten verificar aspectos como la correcta deformación de los personajes y el funcionamiento de los recursos dentro del entorno del videojuego. Además, se revisan los resultados obtenidos después de aplicar los procesos de optimización, con el propósito de comprobar que los modelos mantuvieran sus características visuales y pudieran ser utilizados dentro del entorno interactivo.

Para la planificación y ejecución de estas actividades se implementa la metodología Scrum para la gestión del proyecto, lo que permite organizar el desarrollo de los recursos mediante ciclos de trabajo. Esta metodología facilita la planificación de las actividades, el seguimiento de los avances y la realización de ajustes durante las diferentes etapas del proyecto.

Scrum

La metodología Scrum se organiza en ciclos de trabajo llamados *Sprints*, donde se planifican, desarrollan y revisan las actividades del proyecto. Cada sprint tiene una duración de dos a cuatro semanas, ya que en este tiempo se pueden cumplir los objetivos específicos, dar seguimiento al proyecto y realizar los ajustes necesarios en los modelos 3D, las animaciones y los entornos.

Al finalizar cada *Sprint*, se lleva a cabo una evaluación para revisar si los resultados cumplen con los objetivos planteados. Además, esta etapa permite verificar el funcionamiento y la integración de los recursos desarrollados antes de continuar con el proyecto.

Roles

Product Owner (Dueño del Producto): Este rol es asumido por el tutor de tesis, quien se encarga de definir y priorizar los requerimientos del proyecto para que se cumplan con los objetivos académicos y técnicos establecidos. Además, revisa los avances

presentados al finalizar cada *Sprint*, brinda retroalimentación y aprueba los entregables cuando cumplen con los criterios definidos.

Scrum Master: Este rol es asumido por el tutor de tesis, quien se encarga de garantizar que la metodología Scrum se aplique correctamente durante el desarrollo del proyecto. Además, ayuda a organizar las actividades de cada sprint, también da seguimiento a los avances de las tareas y ayuda a resolver cualquier inconveniente que pueda surgir. De esta manera, se busca que el proyecto avance de forma organizada y se cumplan los objetivos establecidos.

Development Team (Equipo de desarrollo): El equipo de desarrollo está conformado por el investigador, quien será responsable del modelado, texturizado, *rigging* y animación de personajes y objetos, además de su integración en el entorno interactivo. También se encarga de cumplir los objetivos de cada *Sprint*, asegurando calidad visual y la optimización de los recursos desarrollados.

Product Backlog e historias de usuario

Como parte de la metodología Scrum, se elabora un *Product Backlog* con las historias de usuario para organizar las actividades y los requerimientos del proyecto. Además, se utiliza Blender para el modelado, texturizado, *rigging* y animación de los personajes y objetos 3D, mientras que Unity se emplea para su integración y las pruebas de funcionamiento dentro del videojuego.

1. Sprint 0: Análisis y planificación

En esta fase se planifica el proyecto mediante la revisión del estado de arte, identificación de las técnicas y herramientas que se utilizan y los requerimientos gráficos y técnicos. Además, se elabora el *Product Backlog* con las tareas del proyecto. Como

entregables se obtienen el documento de requerimientos, el *Backlog* y las referencias técnicas.

2. Sprint 1: Diseño y Modelado 3D

En esta fase se realiza el diseño conceptual de los personajes y objetos, seguido del modelado de las mallas base, también se revisa la geometría y la topología de los modelos para asegurar una estructura adecuada y se realizan los ajustes necesarios en las proporciones y los detalles. Como resultado se obtienen los primeros modelos 3D con mallas optimizadas y un registro de avances para dar seguimiento al desarrollo del proyecto.

3. Sprint 2: Texturizado, optimización y rigging

En esta fase se realiza el mapeo UV con textura de imagen directa para mejorar la apariencia de los modelos. Además, se optimizan las mallas mediante la reducción de polígonos, ya que esto ayuda a mejorar el rendimiento del videojuego. También se aplica el *rigging* y se realizan pruebas para verificar que los movimientos de los personajes sean correctos.

4. Sprint 3: Desarrollo de animaciones e integración

En esta fase se desarrollan las animaciones básicas de los personajes, como caminar, correr, atacar, y reposo. Además, se ajustan las transiciones para que los movimientos sean más fluidos y naturales. Posteriormente, los modelos y animaciones se integran en Unity para realizar pruebas de funcionamiento e interacción en tiempo real. Como resultado, se obtienen animaciones funcionales, un prototipo integrado y evidencia de las pruebas realizadas.

1.11. Resultados esperados

Como resultado del presente proyecto se espera obtener un conjunto de recursos 3D para el videojuego, conformado por personajes jugables y no jugables, enemigos, escenarios y objetos, desarrollados mediante Blender. Estos recursos estarán texturizados, animados y

optimizados para su integración en Unity. Además, se espera comprobar mediante pruebas de integración y funcionamiento de los modelos. Las texturas, el *rigging* y las animaciones puedan utilizarse correctamente dentro del entorno interactivo del videojuego.

Capítulo II

2. Marco teórico

En este capítulo se presentan los principales conceptos relacionados con el modelado y la animación 3D aplicados al desarrollo de videojuego. Se revisan temas como el modelado tridimensional, las técnicas de creación de personajes y entornos, el texturizado, los materiales, el *rigging*, la animación y la optimización de los modelos para motores gráficos. También se analizan diferentes trabajos relacionados, herramientas y la metodología utilizada en la producción del proyecto.

2.1. Antecedentes de investigación

En este apartado se revisan tres videojuegos relacionados con la creación de personajes, criaturas y elementos de entornos virtuales, considerando diferentes estilos artísticos, técnicas de modelado 3D y recursos visuales aplicados al desarrollo de videojuegos. De cada videojuego se analizaron sus características visuales, el diseño de sus personajes y enemigos, el uso de formas, proporciones, texturas y el nivel de detalle que tienen sus modelos tridimensionales.

2.1.1. The Legend of Zelda: Breath of the Wild

The Legend of Zelda: Breath of the Wild es un videojuego de acción y aventura desarrollado por Nintendo y lanzado en 2017. El videojuego presenta un mundo abierto con diversos entornos, estructuras y elementos naturales que forman parte de su diseño visual. Su estilo artístico emplea formas estilizadas, colores nítidos y una combinación de elementos naturales para construir escenarios atractivos.

Este videojuego llama la atención por su estilo visual y su diseño de niveles. La forma en que se combinan los elementos en un entorno demuestra cómo un escenario puede transmitir su una buena estructura como, objetos, vegetación y características del paisaje.

- **Estilo visual:** El videojuego utiliza una estética estilizada que combina formas simples, pero con un estilo visual atractivo. Este aspecto sirvió de referencia para analizar cómo los modelos 3D pueden mantener una buena apariencia sin depender excesivamente del realismo.
- **Diseño de niveles:** La variedad de entornos del videojuego tiene mucha importancia para la creación de espacios con características distintivas.
- **Diseño de objetos y elementos del entorno:** Los objetos que forman parte de la escena ayudan a definir la identidad del espacio. Elementos como estructuras, casas y objetos decorativos sirvieron de referencia para comprender cómo los componentes individuales pueden mejorar un entorno 3D.
- **Integración visual del entorno:** La relación entre los distintos elementos del escenario garantiza que el mundo virtual mantenga una coherencia visual.

Ilustración 1

Presentación visual de la jugabilidad en The Legend of Zelda: Breath of the Wild



Nota: Obtenido de [23]

2.1.2. Genshin Impact

Genshin Impact es un videojuego de rol y acción desarrollado por HoYoverse y lanzado en 2020. Este juego presenta personajes con diseños visuales únicos, una estética estilizada y un alto nivel de detalle en la vestimenta, los accesorios y los rasgos físicos. Además, cada personaje cuenta con rasgos propios, lo que permite reconocerlos fácilmente dentro del juego.

Este videojuego es una referencia clave, principalmente en el modelado 3D y la texturización. La forma en que cada personaje combina elementos como vestimenta, materiales y colores que tiene mucha importancia al crear modelos con rasgos visuales propios.

- **Diseño de personajes:** Cada personaje tiene rasgos visuales que lo distinguen de los demás.

- **Modelado 3D:** Los personajes se construyen definiendo claramente sus cuerpos, vestimenta y accesorios.
- **Texturización y materiales:** El uso de diferentes materiales en la vestimenta, los accesorios y los elementos del personaje genera una mayor variedad visual.
- **Identidad visual del personaje:** La combinación de modelado, texturas, colores y accesorios garantiza que cada personaje tenga una apariencia propia.

Ilustración 2

Presentación visual de la jugabilidad en Genshin Impact



Nota: obtenido de [24]

2.1.3. Dark Souls III

Dark Souls III es un videojuego de rol y acción desarrollado por FromSoftware y lanzado en 2016. Este juego es conocido por su ambientación de fantasía oscura, sus enemigos y animaciones durante el combate. Cada personaje y enemigo tiene movimientos según sus rasgos físicos, armas y comportamiento en batalla.

El juego destaca principalmente por sus animaciones, movimientos y diseño de enemigos. La forma en que los personajes atacan, esquivan y reaccionan tiene mucha importancia al crear animaciones naturales según el diseño y la función de cada personaje.

- **Animación de personajes:** Los movimientos de los personajes se relación con sus acciones y rasgos físicos.
- **Estilo de movimiento:** Cada personaje tiene movimientos según su diseño.
- **Diseño de enemigos:** El videojuego presenta enemigos con rasgos físicos y visuales distintivos. Sus formas, tamaños y movimientos ayudan a diferenciarlos dentro del entorno, asegurando que sus rasgos visuales y movimientos los distinguieran del personaje principal y de los NPC.
- **Relación entre modelo y animación:** Los personajes deben estar correctamente modelados para su deformación y el movimiento. Esto es importante para el proceso de *rigging* y animación de los personajes desarrollados para el proyecto, ya que un modelo bien estructurado facilita el movimiento y la animación es más natural.

Ilustración 3

Presentación visual de la jugabilidad en *Dark Souls III*



Nota: obtenido de [25]

2.1.4. Cuadro comparativo de trabajos relacionados

Tabla 1

Resumen comparativo de los trabajos relacionados

Trabajo relacionado	<i>The Legend of Zelda: Breath of the Wild</i>	<i>Genshin Impact</i>	<i>Dark Souls III</i>
Aspectos Analizados	Estilo visual, diseño de escenarios y objetos	Diseño de personajes, modelado 3D y texturizado	Animaciones, estilo de movimiento y diseño de enemigos
Principales aportes como referente	Uso de formas estilizadas, integración de elementos ambientales y construcción de entornos visuales	Identidad visual, proporciones, vestimenta, colores y materiales	Relación entre las características del personaje, sus movimientos y sus acciones dentro del videojuego
Relacion con “ <i>Ashes of Faith</i> ”	Inspiración para la creación de escenarios y objetos del videojuego	Referencia para la construcción tridimensional y texturizado de personajes, NPC y enemigos	Referencia para el desarrollo de animaciones, movimientos y diferenciación de enemigos

Nota: Elaboración propia

2.2. Bases teóricas

El desarrollo de modelos 3D para videojuegos incluye varios procesos que convierten una idea o concepto en un recurso digital listo para un entorno interactivo. Este trabajo tiene diferentes etapas como el modelado 3D, la aplicación de texturas y materiales, la preparación de personajes con *rigging* y animación, y la optimización de los recursos para que funcionen en motores gráficos. Cada etapa es importante ya que está conectada con la demás.

2.2.1. Modelado 3D en el desarrollo de videojuegos

El modelado 3D es importante en el desarrollo de videojuegos, ya que permite transformar una idea o un concepto en un elemento visual con el que el jugador puede interactuar dentro de un entorno virtual. Wang y Zhang señalan que la creación de personajes 3D para videojuegos requiere considerar la eficiencia del proceso de modelado, con el propósito de obtener modelos adecuados para su utilización dentro del videojuego [2]. Por lo tanto, el modelado no consiste únicamente en obtener un resultado atractivo, sino también en mantener un equilibrio entre la calidad visual y el rendimiento del videojuego.

Con el avance de la tecnología, las técnicas de modelado 3D han evolucionado considerablemente. Mientras que los primeros videojuegos utilizaban modelos simples debido a las limitaciones de hardware, actualmente se han desarrollado diferentes métodos para facilitar la creación de modelos 3D. Qiu et al. presentan un método para generar modelos 3D animables a partir de una sola imagen, demostrando la aplicación de técnicas basadas en inteligencia artificial para la creación de modelos de personajes tridimensionales [22]. Además, Arifudin et al. plantean técnicas de optimización de *assets* 3D mediante métodos de

reducción de geometría, con el propósito de reducir el tamaño de los recursos y favorecer su utilización dentro de los videojuegos [16].

Ilustración 4

Ejemplo visual del concepto de los niveles de poligonaje



Nota: Elaboración propia

La cantidad de la geometría utilizada en un modelo 3D es otro aspecto importante del proceso de desarrollo. Esto se debe a que influye en el nivel de detalle y en los recursos que se necesitan para procesarlo. Hanocka et al. presentan un método orientado al análisis de mallas 3D, mediante el cual se trabaja con la estructura geométrica de los modelos para su procesamiento [20]. En este contexto, los modelos pueden clasificarse según su densidad geométrica. La selección de niveles de detalle depende de los requerimientos del proyecto y de las características de la plataforma de destino [2].

2.2.2. Técnicas de modelado para personajes y entornos interactivos

El modelado de personajes y entornos interactivos incluye técnicas que transforman un diseño conceptual en un modelo 3D listo para uso en un videojuego. La técnica depende

de las características del proyecto, del nivel de detalle que se desea obtener y de las limitaciones del motor gráfico donde será implementado. Las técnicas más utilizadas son el modelado poligonal (es una de las técnicas más básicas para la construcción de la malla en las coordenadas x, y, z), el esculpido digital (simula el proceso de modelar con arcilla, pero los modelos tienen millones de vértices. Esto permite crear personajes con un nivel alto de polígonos) y la retopología (proceso donde se crea una nueva malla optimizada de baja o media densidad sobre un modelo con nivel alto de detalle). Kazmi et al. presentan un método de creación de personajes mediante deformaciones de malla a partir de bocetos, relacionados con la aplicación de técnicas de modelado para obtener personajes 3D [7].

En la creación de un personaje es común combinar varias técnicas en el mismo proceso de producción. Por lo general, el modelado comienza con una malla base que se pule usando herramientas de escultura digital. Una vez que se completa este proceso, se lleva a cabo la retopología que reorganiza la geometría del modelo. Pant et al. estudian la reducción de la cantidad de polígonos mediante técnicas de retopología y mapas de normales, permitiendo optimizar los modelos 3D para su utilización como *assets* [26]. Estas técnicas también permiten obtener una topología (es la estructura actual del modelo) bien organizada, lo que reduce la complejidad y facilita su utilización en diferentes procesos de modelado, además de prepararlo para etapas posteriores, como el *rigging* y la animación.

2.2.3. Texturizado y materiales en modelos 3D

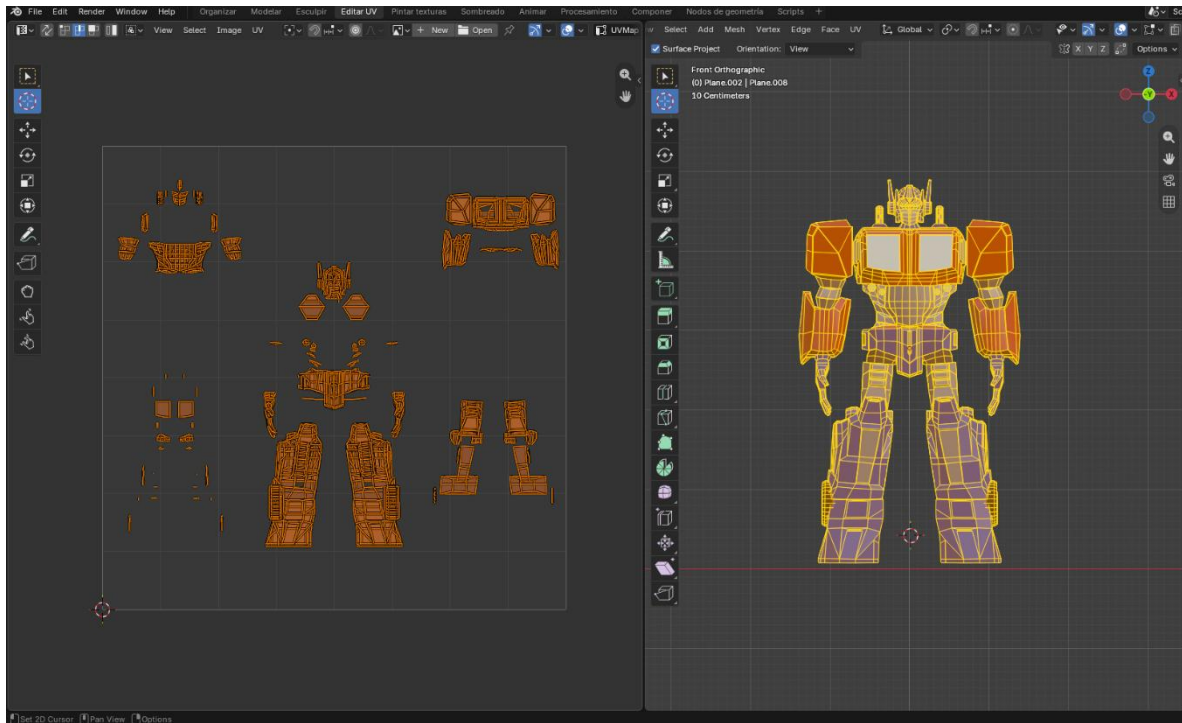
Estos elementos añaden color, detalles y realismo a los modelos 3D. Mediante este proceso es posible mostrar características como la apariencia de materiales como piedra, metal, madera o tela de este modo, los personajes y los escenarios presenten una apariencia más detallada y coherente en los videojuegos. Para realizar este proceso es importante

realizar un buen mapeo UV, ya que este permite establecer como se ven las texturas aplicadas sobre los modelos y evitar deformaciones durante su aplicación. Uccheddu et al. estudian técnicas de mapeo de texturas aplicadas a modelos 3D, destacando la importancia de establecer una adecuada implementación entre las texturas y la superficie de los modelos [14].

En el desarrollo de videojuegos, el texturizado no solo busca mejorar la apariencia visual, sino también utilizar los recursos gráficos de manera correcta para evitar errores innecesarios en el rendimiento. Usar materiales y mapas de texturas de manera correcta mantienen un buen nivel de detalle sin aumentar la complejidad del modelo, facilitando su integración en motores gráficos. Koulaxidis y Xinogalos analizan diferentes técnicas de optimización para mejorar el rendimiento de videojuegos desarrollados en Unity, destacando la importancia de optimizar los recursos utilizados durante la ejecución [4].

Ilustración 5

Mapeo UV (UV mapping)



Nota: Elaboración propia

El proceso de texturizado comprende diferentes etapas que permiten definir la apariencia visual de un modelo 3D. Una de estas etapas es el mapeo UV, que permite aplicar correctamente las texturas o colores sobre la superficie del modelo. También se utilizan materiales para presentar superficies como metal, madera, piedra o tela. Estos elementos trabajan juntos para mostrar un gran nivel de detalle y mantiene el rendimiento necesario para los videojuegos.

2.2.4. Rigging y animación para videojuegos

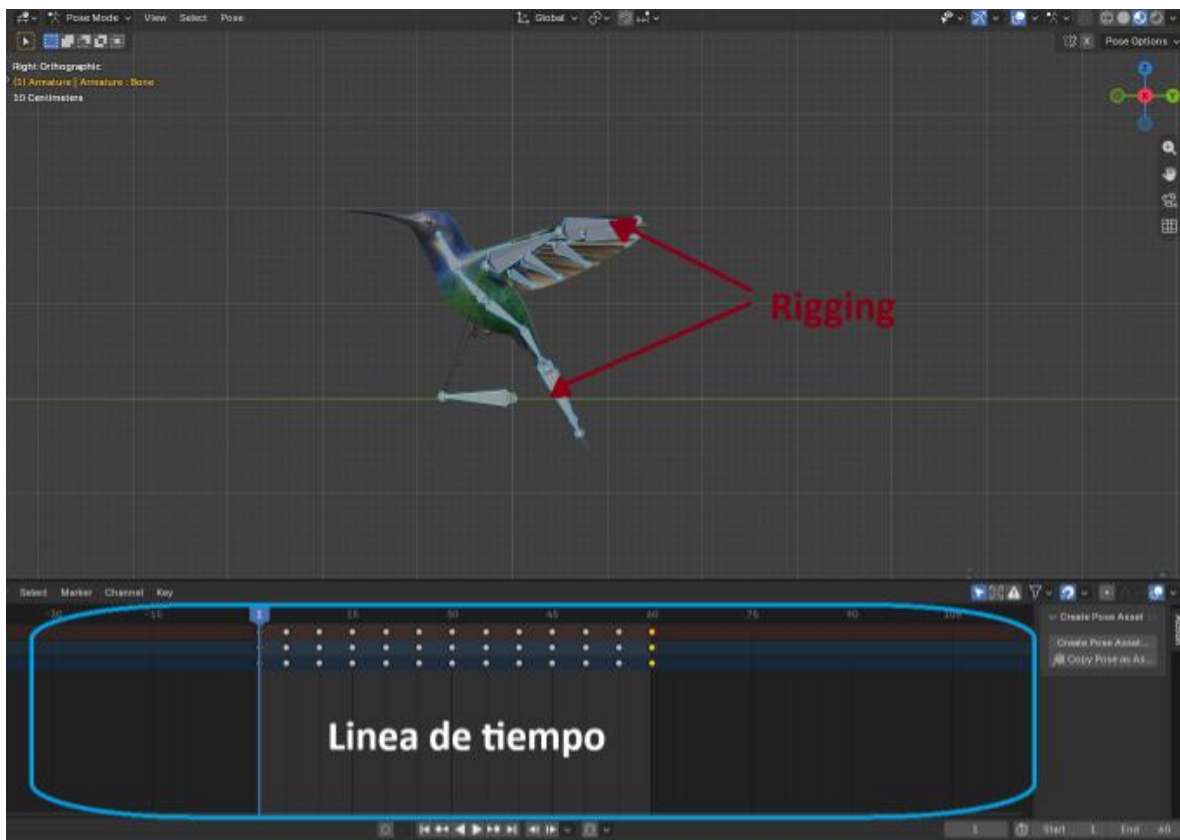
El *rigging* es el proceso de crear un esqueleto que permite mover un modelo 3D. Multon et al. señalan que la estructura facilita el control de cada parte del personaje y hace que sus movimientos sean más naturales, según las acciones del videojuego. Una vez se

completa el *rigging*, se continúa con las animaciones. En esta fase, se diseñan los movimientos para dar vida a los personajes, lo que añade mayor realismo a la experiencia del jugador [9].

En los videojuegos, una buena animación no solo mejora la apariencia de los personajes, sino que también afecta la jugabilidad y la sensación que percibe el usuario. Además, las técnicas como la captura de movimiento y los sistemas de animación automatizados han hecho que este proceso sea más eficiente, logrando resultados más precisos y reduciendo el tiempo de producción [9].

Ilustración 6

Presentación visual del rigging



Nota: Elaboración propia

El proceso de *rigging* y animación son etapas diferentes que permiten dar movimiento a los modelos 3D. Kenwright señala que el *rigging* es el proceso de crear un esqueleto que facilita el control y movimiento del personaje [8], y permite generar movimientos más naturales según las acciones del videojuego. Una vez completado se continúa con las animaciones. Multon et al. indican que trabajar con fotogramas por segundo (FPS) permite animar a los personajes y aportar mayor realismo a la experiencia del jugador [9].

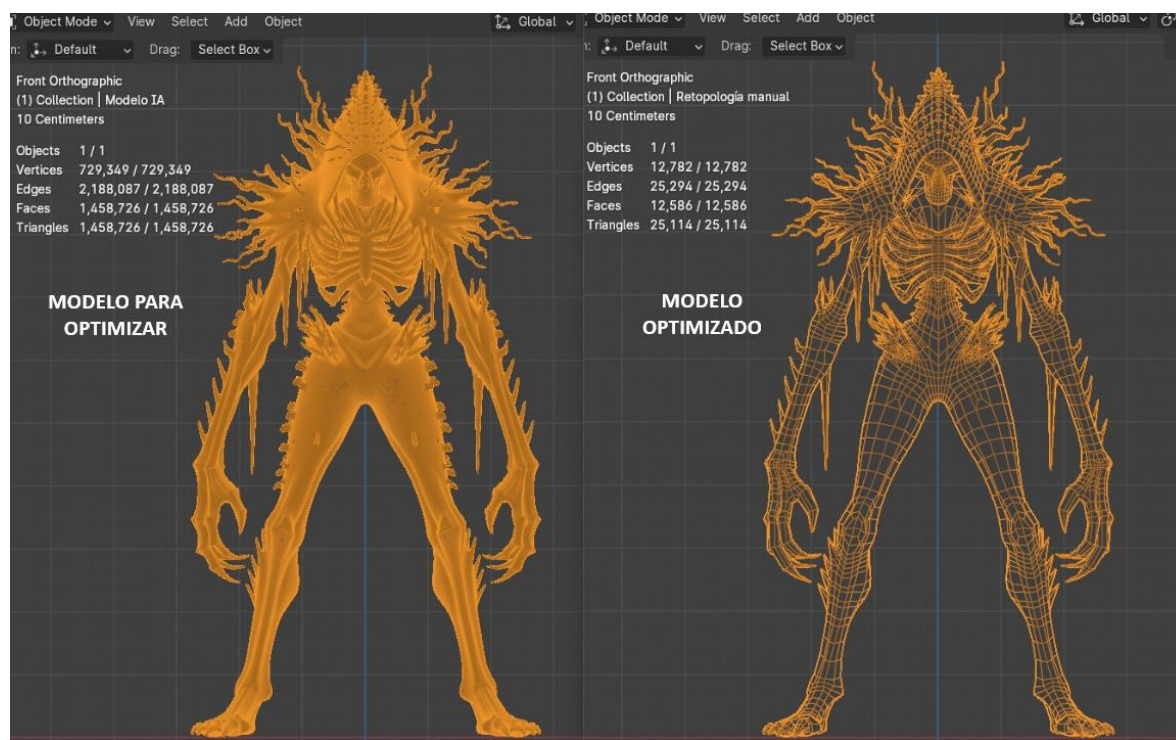
2.2.5. Optimización de modelos 3D para motores gráficos en tiempo real

La optimización de modelos es una etapa clave para el desarrollo de videojuegos, ya que permite que los personajes y entornos funcionen de manera eficiente dentro de un motor gráfico sin afectar la calidad visual. Este proceso incluye reducir la complejidad de los modelos mediante técnicas como la reducción de polígonos, reutilizar materiales y usar adecuadamente las texturas. Koulaxidis y Xinogalos señalan la importancia de aplicar técnicas de optimización para mejorar el rendimiento del videojuego y garantizar una ejecución fluida [4].

Hoy en día, la optimización forma parte del flujo de trabajo desde las primeras etapas del modelado, ya que facilita la integración de los recursos en motores gráficos, esto ayuda a reducir el consumo de memoria y procesamiento. Además de mejorar el desempeño del videojuego, estas técnicas mantienen una experiencia visual consistente para el jugador, especialmente en escenas con muchos personajes y objetos interactivos [4].

Ilustración 7

Optimización de modelos 3D



Nota: Elaboración propia

La optimización de modelos 3D es una etapa clave para el desarrollo de videojuegos, ya que permite reducir vértices, aristas y caras sin perder la calidad visual de los personajes y objetos manteniendo un buen funcionamiento dentro de un motor gráfico.

2.3. Bases metodológicas

El desarrollo de videojuegos tiene varias áreas de trabajo como programación, diseño, modelado, animación y pruebas. Por eso, es necesario usar metodologías que ayuden a planificar, organizar y seguir las actividades del equipo. Hoy en día, las metodologías ágiles son una buena opción para organizar el flujo de trabajo en el desarrollo de videojuegos. Además, estas metodologías permiten adaptarse a cambios, mejorar la comunicación entre el equipo y realizar entregas durante el desarrollo del proyecto.

2.3.1. Scrum

Scrum es una metodología ágil que gestiona proyectos de forma organizada e interactiva. Su objetivo principal es dividir el trabajo en periodos cortos de tiempo llamados *Sprints*, por lo cual el equipo realiza un conjunto de tareas planificadas. Esta metodología ayuda con la colaboración entre los miembros del equipo, la comunicación y la capacidad de adaptarse a los cambios que puede surgir durante el desarrollo del proyecto. Estos aspectos la convierten en una de las metodologías más populares en el desarrollo de videojuegos [27], [28].

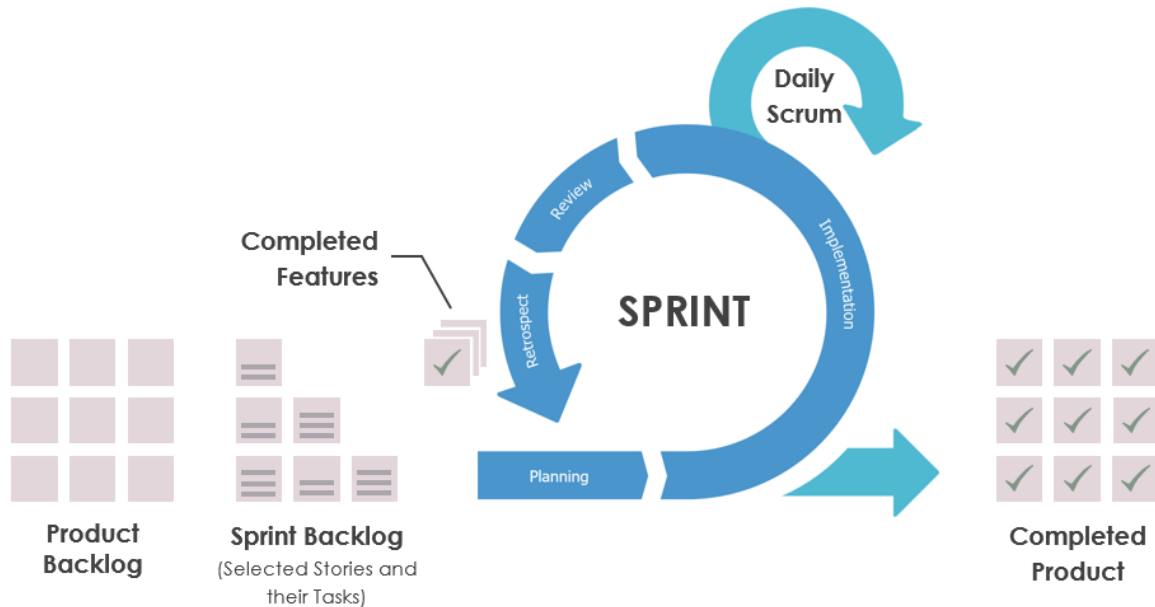
Scrum se basa en la participación de tres roles principales: el *Product Owner*, responsable de definir las necesidades del proyecto; el *Scrum Master*, quien facilita la aplicación de la metodología y apoya al equipo; y el Equipo de Desarrollo, que está encargado de ejecutar las tareas planificadas. También incluye reuniones periódicas que permiten revisar avances del proyecto y detectar problemas a tiempo [27].

El flujo de trabajo de Scrum comienza con la creación del *Product Backlog*, donde se registran todas las actividades y requerimientos del proyecto. Luego, el equipo divide las tareas que se desarrollarán durante un Sprint y las organiza en el (*Sprint Backlog*). Al iniciar el Sprint, se llevan a cabo reuniones dentro de dos semanas para revisar el progreso de las actividades. Al final de cada ciclo, se presenta el trabajo realizado mediante una revisión (*Sprint Review*) y se realiza una reunión de retrospectiva (*Sprint Retrospective*) para identificar oportunidades de mejora para los siguientes ciclos de desarrollo [27], [28].

La figura 14 muestra el flujo de trabajo de la metodología Scrum, donde las tareas se organizan de una lista general a un ciclo de trabajo llamado sprint.

Ilustración 8

Flujo de trabajo de la metodología Scrum



Nota: obtenido de [29]

2.3.2. Kanban

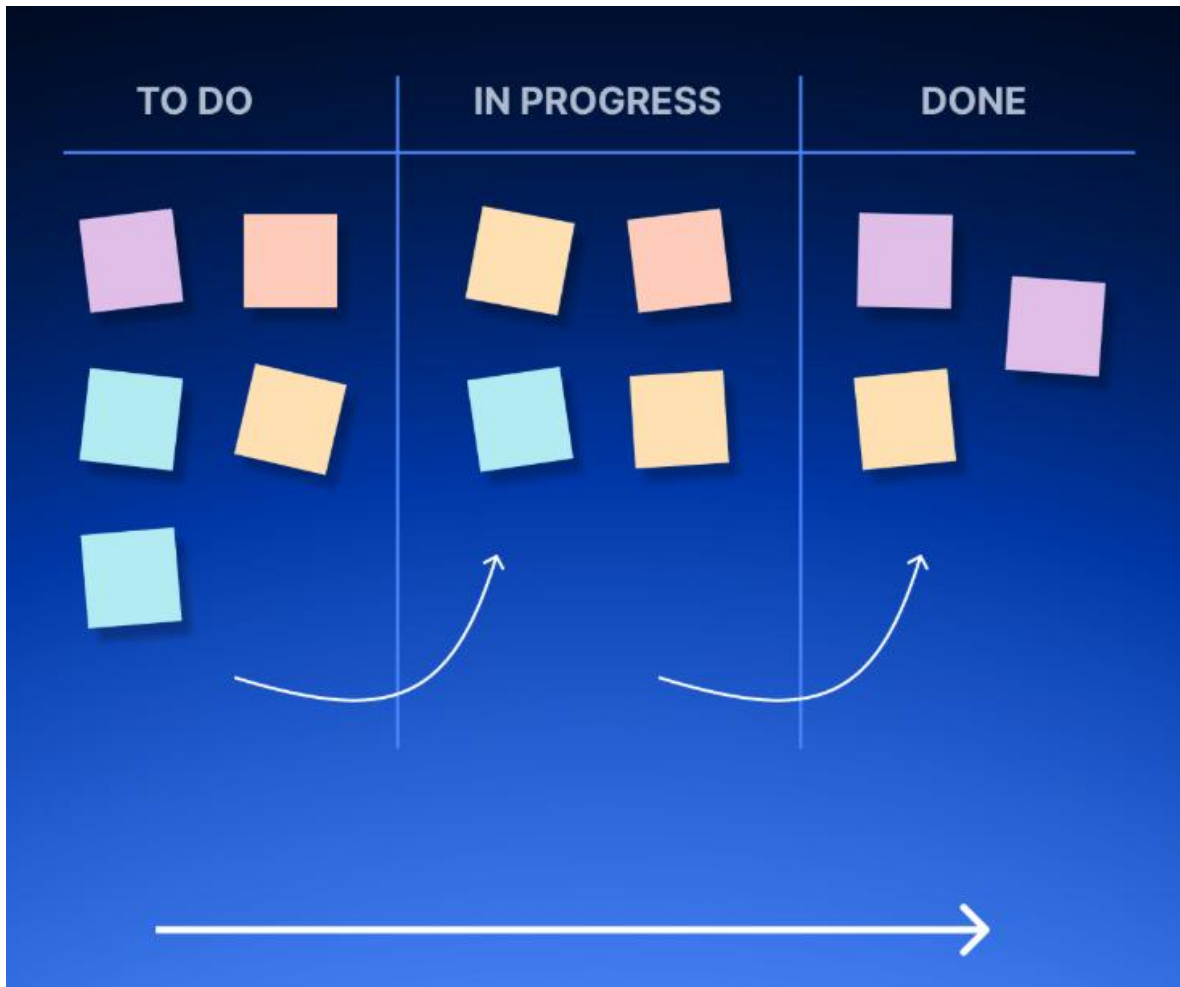
Kanban es una metodología ágil que se enfoca en la gestión visual del trabajo y en la optimización del flujo de actividades en un proyecto. Funciona con un tablero dividido en columnas que representan las diferentes etapas del proceso, permitiendo conocer en todo momento el estado de cada tarea. Esta metodología ayuda a organizar el trabajo, mejora la comunicación del equipo y ayuda a identificar retrasos durante el desarrollo del proyecto [30], [31].

Una de las principales ventajas de Kanban es su flexibilidad. No depende de ciclos de tiempo definidos como Scrum, sino que las actividades pueden incorporarse, modificarse o finalizarse según las necesidades del proyecto, lo que permite una adaptación continua a los cambios. Además, establece límites en la cantidad de tareas que pueden ejecutarse al

mismo tiempo, ayudando con una mejor distribución de la carga de trabajo y aumentando la productividad del equipo [30].

El flujo de trabajo en Kanban comienza con la creación de un tablero que registra todas las actividades pendientes. Estas tareas avanzan por diferentes columnas, generalmente etiquetadas como pendiente, en proceso y finalizado, hasta que se completan. Durante este proceso, el equipo supervisa constantemente el movimiento de las tareas para asegurar un flujo continuo y detectar problemas que puedan afectar el cumplimiento de los objetivos del proyecto [30], [31].

La figura 15 muestra la estructura básica de un tablero Kanban, una herramienta visual diseñada para gestionar el flujo de trabajo dividiendo el proceso en tres columnas.

Ilustración 9*Flujo de trabajo de la metodología Kanban*

Nota: obtenido de [32]

2.3.3. Extreme Programming (XP)

Extreme Programming (XP) es una metodología ágil que se centra en mejorar la calidad del software mediante ciclos de desarrollo cortos, retroalimentación y comunicación con el equipo de trabajo y el cliente. Su meta es responder rápidamente a los cambios en el proyecto, promoviendo prácticas que facilitan la entrega frecuente de versiones funcionales y el desarrollo de software de alta calidad [33], [34].

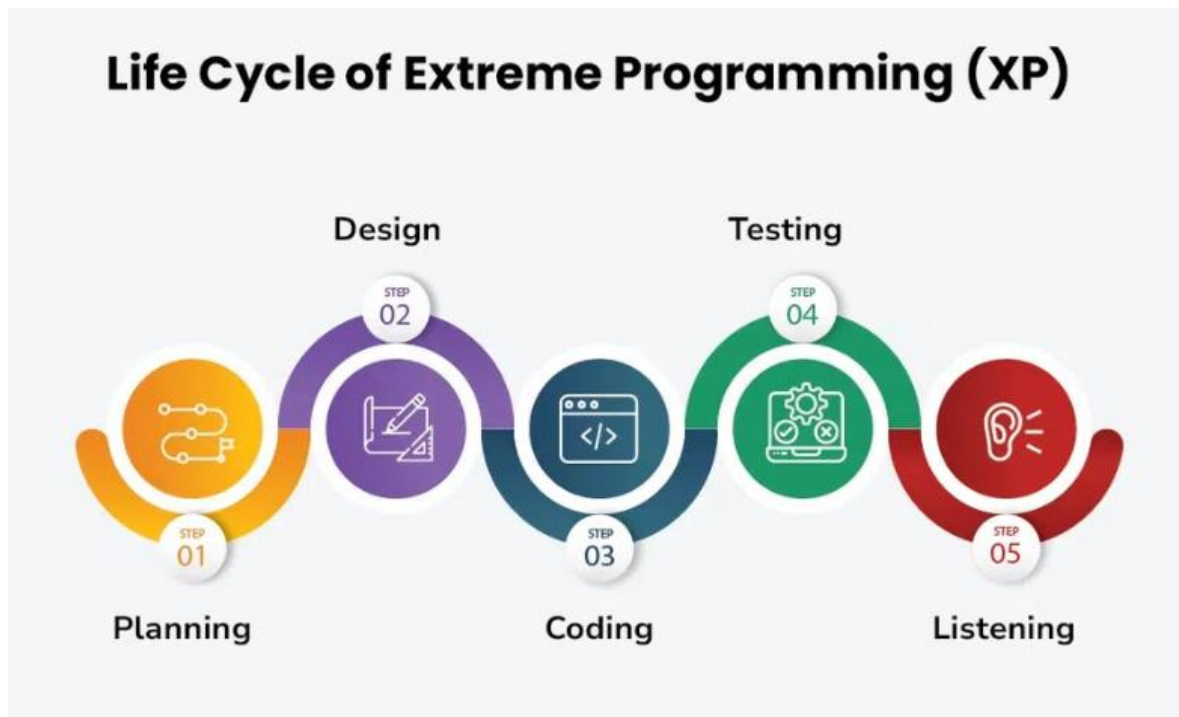
Esta metodología incluye prácticas como la programación en parejas (*Pair Programming*), el desarrollo guiado por pruebas (*Test-Driven Development*) y la integración continua del código. Estas actividades ayudan a detectar errores tempranamente, mejoran la calidad del producto y mantienen un desarrollo más organizado en todas las etapas del proyecto [33].

El flujo de trabajo de Extreme Programming (XP) comienza con la recopilación de los requerimientos del cliente a través de historias de usuario (*User Stories*). Luego, el equipo planifica las actividades, desarrolla pequeñas funcionalidades con ciclos de programación cortos y realiza pruebas continuas para verificar el correcto funcionamiento del sistema. Finalmente, el software se integra, se presenta al cliente para recibir retroalimentación y se repite el proceso hasta completar el desarrollo del proyecto [33], [34].

La figura 16 ilustra el ciclo de vida de Extreme Programming (XP) mediante una secuencia de cinco pasos interconectados que guían el desarrollo del proyecto.

Ilustración 10

Flujo de trabajo de la metodología Extreme Programming (XP)



Nota: obtenido de [35]

2.3.4. Cuadro comparativo de Metodologías

Tabla 2

Comparación de metodologías ágiles

Características	Scrum	Kanban	Extreme Programming (XP)
Tipo de metodología	Ágil e iterativa	Ágil de flujo continuo	Ágil orientado al desarrollo de software
Formato de Trabajo	Se organiza mediante Sprints con una duración definida	Flujo continuo de tareas mediante un tablero visual	Desarrollo en iteraciones cortas con programación y pruebas constantes
Organización del trabajo	Product Backlog y Sprint Backlog	Tablero Kanban dividido por estados de las tareas	Historias de usuario y planificación de iteraciones

Roles definidos	Product Owner, Scrum Master y Equipo de Desarrollo	No establece roles específicos	Cliente, desarrolladores y entrenador (Coach)
Planificación	Se realiza antes de cada Sprint	Flexible; las tareas se incorporan según las necesidades del proyecto	Planificación basada en historias de usuario y prioridades del cliente
Seguimiento del proyecto	Reuniones diarias (Daily Scrum) y revisiones al finalizar cada Sprint	Seguimiento visual del avance mediante el tablero Kanban	Reuniones frecuentes y retroalimentación continua con el cliente
Flexibilidad ante cambios	Alta, aunque los cambios suelen incorporarse en el siguiente Sprint	Muy alta, permite modificar prioridades en cualquier momento	Muy alta, se adapta continuamente a nuevos requerimientos
Control de progreso	Mediante entregas parciales y evaluación al finalizar cada Sprint	A través del estado de las tareas y el flujo de trabajo	Mediante pruebas continuas e integración frecuente del software
Ventajas	Favorece la organización, el trabajo colaborativo y la planificación del proyecto	Facilita la visualización de tareas y mejora la gestión del flujo de trabajo	Incrementa la calidad del software mediante pruebas continuas y retroalimentación constante
Limitaciones	Requiere una buena planificación y compromiso del equipo	Puede generar desorganización si no se controlan los límites de trabajo	Requiere una alta participación del cliente y mayor disciplina técnica
Aplicación en videojuegos	Coordinación de equipos multidisciplinarios y desarrollo por entregas	Organización y seguimiento de tareas durante la producción	Desarrollo y mejora continua del código del videojuego

Nota: Elaboración propia

Tras comparar las metodologías ágiles analizadas, se determinó que Scrum era la alternativa más adecuada para el desarrollo del proyecto, ya que ayuda a planificar, distribuir

y controlar las actividades del equipo. La planificación mediante *Sprints* ayuda con el seguimiento y progreso de cada integrante, facilitando la coordinación entre las áreas de programación, modelado 3D, texturizado, animación y diseño. Manteniendo un flujo de trabajo ordenado y enfocado con el cumplimiento de los objetivos del proyecto.

2.4. Bases tecnológicas

2.4.1. Herramientas de modelado

En este apartado se revisan las herramientas que sirven para desarrollar los modelos tridimensionales. La revisión se centra en los programas y recursos empleados en las distintas etapas del proceso de producción 3D, como el modelado, la optimización de mallas, el texturizado, el *rigging*, la animación y la preparación de los modelos para su integración en el videojuego.

2.4.1.1. Blender

Blender es un software libre y de código abierto, desarrollado por Blender Foundation. Está diseñado para crear contenido tridimensional. Hoy en día, es una de las herramientas más populares en el diseño 3D. Permite realizar varios procesos en un solo entorno, como modelado, texturizado, creación de materiales, *rigging*, animación, simulación y renderizado. También ofrece compatibilidad con múltiples formatos de exportación, lo que facilita la integración de modelos en motores gráficos como Unity y Unreal Engine [36].

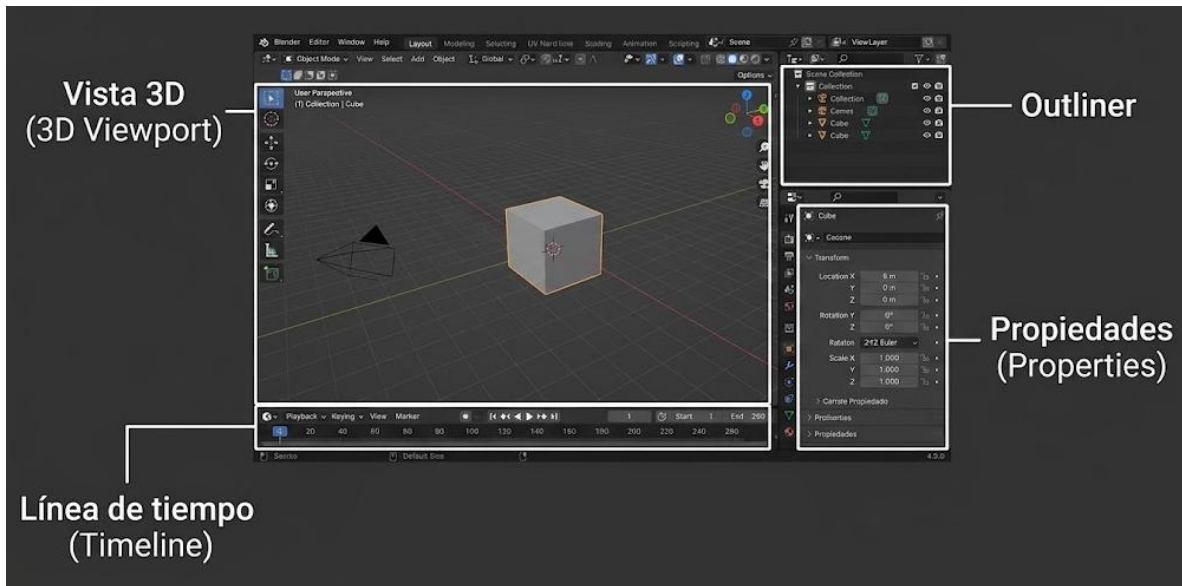
Blender se usa tanto en proyectos académicos como profesionales. Proporciona herramientas útiles para desarrollar personajes, escenarios y objetos para videojuegos, animaciones y producciones audiovisuales. Como es un software de código abierto, tiene una comunidad activa que contribuye al desarrollo de nuevas funciones, complementos y

recursos de aprendizaje. Además, cuenta con una extensa documentación oficial, tutoriales y foros que ayudan a los nuevos usuarios con su proceso de aprendizaje [36].

- **Layout:** ofrece una vista general del proyecto y permite organizar los diferentes objetos en la escena.
- **Modeling:** reúne las herramientas necesarias para crear y modificar modelos tridimensionales mediante la edición de vértices, aristas y caras.
- **Sculpting:** incluye herramientas de esculpido digital que permiten añadir detalles orgánicos y de alta resolución a los modelos.
- **UV Editing:** despliega las coordenadas UV del modelo para aplicar correctamente texturas y materiales.
- **Shading:** ayuda a crear y editar materiales usando nodos, lo que permite controlar la apariencia visual de los modelos.
- **Animation:** integra las herramientas para crear y editar animaciones mediante fotogramas clave (*Keyframes*), curvas de animación y líneas de tiempo.
- **Rendering:** permite ajustar la iluminación, cámaras y otros parámetros necesarios para generar imágenes o animaciones finales utilizando motores de render como *Eevee* y *Cycles*.

Ilustración 11

Principales interfaces del motor gráfico de Blender



Nota: Obtenido de Blender [36]

2.4.1.2. Autodesk Maya

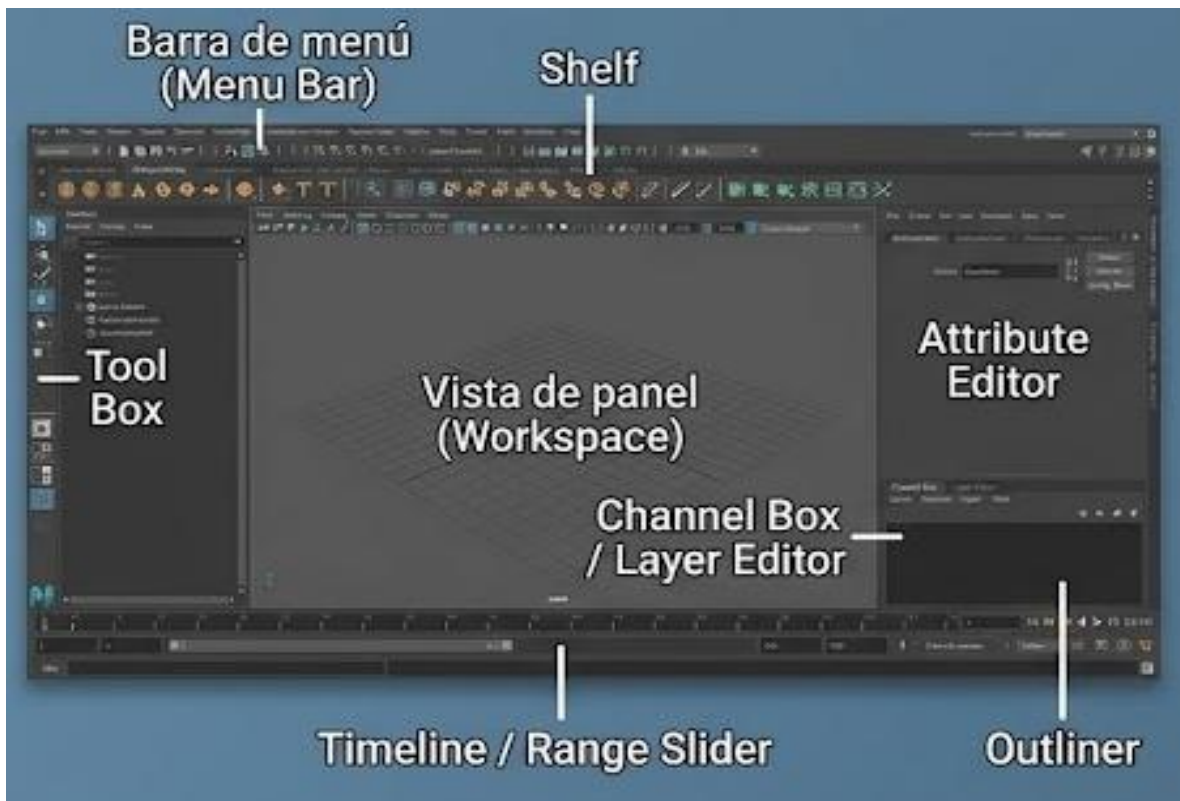
Autodesk Maya es un software profesional creado por Autodesk para hacer contenido en 3D. Se usa mucho en el cine, la animación y el desarrollo de videojuegos. Ofrece muchas herramientas para modelar, animar, *riggear*, simular y crear efectos visuales. Su capacidad para manejar proyectos complejos lo hace una de las aplicaciones más reconocidas en la producción digital. Muchos estudios lo utilizan para crear personajes, escenarios y secuencias animadas [37].

Una de las características clave de Autodesk Maya es su extensa colección de herramientas para crear y editar modelos en 3D. El programa permite desarrollar modelos usando técnicas de modelado poligonal, modelado con curvas y superficies NURBS. También tiene funciones para esculpido digital y edición de geometría. Además, cuenta con herramientas para crear materiales, iluminación, animación de personajes y simulaciones físicas. Esto facilita el desarrollo de proyectos con un alto nivel de detalle [37].

- **Viewport:** permite visualizar y editar modelos en 3D dentro de la escena.
- **Outliner:** muestra la organización jerárquica de todos los objetos en el proyecto.
- **Channel Box / Attribute Editor:** permite cambiar las propiedades y transformaciones de los objetos seleccionados.
- **Modeling Toolkit:** reúne las herramientas para crear y editar geometría.
- **UV Editor:** facilita el mapeo UV para aplicar materiales y texturas correctamente.
- **Animation Workspace:** incluye las herramientas necesarias para crear animaciones usando fotogramas clave (*Keyframes*), curvas de animación y editores de tiempo.

Ilustración 12

Principales interfaces del motor gráfico de Autodesk Maya



Nota: Obtenido de Autodesk Maya [37]

2.4.1.3. Autodesk 3ds Max

Autodesk 3ds Max es un software profesional hecho por Autodesk para crear modelos en 3D, animaciones y visualizaciones digitales. Se usa mucho en el desarrollo de videojuegos, arquitectura, diseño industrial y producción audiovisual. Ofrece funciones especializadas para modelar, aplicar materiales, iluminar, animar y renderizar. Su interfaz y flujo de trabajo permiten desarrollar proyectos de alta calidad. Por eso, es uno de los programas más conocidos en la industria del diseño 3D [38].

Una de las características clave de Autodesk 3ds Max es la amplia gama de herramientas para el modelado poligonal y la edición de geometría. También permite crear materiales y texturas con diferentes sistemas de sombreado, desarrollar animaciones con fotogramas clave y generar simulaciones para diversos tipos de proyectos. Además, es compatible con diferentes formatos de archivo. Esto facilita el intercambio de modelos con otros programas de diseño y su integración en motores gráficos como Unity y Unreal Engine [38].

- **Viewport:** permite visualizar y editar modelos en 3D desde diferentes ángulos.
- **Command Panel:** reúne las herramientas para crear, modificar y gestionar objetos en la escena.
- **Scene Explorer:** muestra la estructura jerárquica de todos los elementos del proyecto.
- **Material Editor:** permite crear, modificar y asignar materiales y texturas a los modelos en 3D.
- **Modifier Stack:** facilita la aplicación y edición de modificadores sin cambiar la geometría original del modelo.

- **Animation Controls:** incluye las herramientas necesarias para crear y editar animaciones usando líneas de tiempo y fotogramas clave.

Ilustración 13

Principales interfaces del motor gráfico de Autodesk 3ds Max



Nota: Obtenido de Autodesk 3ds Max [38]

2.4.2. Herramientas para texturizado

El proceso de texturizado es una de las etapas más importantes dentro de la creación de modelos 3D para videojuegos, ya que permite definir la calidad visual de los personajes, escenarios y objetos mediante la aplicación de colores y materiales. Para realizar este existen diferentes herramientas que ofrecen funciones para la edición de imágenes, facilitando la creación de recursos gráficos adaptados a motores gráficos.

2.4.2.1. Adobe Photoshop

Adobe Photoshop es un software creado por Adobe Inc. Especializado en la edición y manipulación de imágenes. Gracias a su gran variedad de herramientas para dibujo, pintura

y edición fotográfica, se ha vuelto uno de los programas más utilizados para crear texturas para modelos 3D y videojuegos. Además, permite trabajar con imágenes de alta resolución y varios formatos que funcionan con diferentes programas de diseño y motores gráficos [39].

Dentro del desarrollo de videojuegos, Photoshop es utilizado para crear y editar mapas de color (*Diffuse o Base Color*), mapas de transparencia y detalles gráficos. También nos ayuda con la creación de texturas personalizadas y corrección de imágenes antes de aplicar a un modelo 3D [39].

- **Capas (Layers):** organizan los diferentes elementos de una textura de manera independiente.
- **Pinceles (Brush Tool):** facilitan la pintura y edición manual de detalles en las imágenes.
- **Herramientas de selección:** permiten modificar áreas específicas de la textura con mayor precisión.
- **Filtros y ajustes:** ofrecen opciones para corregir color, iluminación, contraste y otros aspectos visuales.
- **Exportación de imágenes:** guarda texturas en formatos como PNG, JPG, TIFF y PSD para su uso en otros programas.

Ilustración 14

Principales interfaces del motor gráfico de Adobe Photoshop



Nota: Obtenido de Adobe Photoshop [39]

2.4.2.2. Krita

Krita es un software libre y de código abierto orientado a la ilustración y pintura digital, desarrollado por Krita Foundation. Aunque se centra en el arte digital, también se usa mucho para crear y editar texturas aplicadas a modelos 3D. Es gratuita y ofrece diferentes herramientas de dibujo, lo que lo convierte en una opción accesible para estudiantes, artistas independientes y desarrolladores de videojuegos [40].

- Sistema avanzado de pinceles.
- Gestión de capas y máscaras.
- Herramientas de selección y transformación.

- Soporte para tabletas gráficas.
- Exportación en múltiples formatos de imagen.

Ilustración 15

Principales interfaces del motor gráfico de Krita



Nota: Obtenido de Krita [40]

2.4.2.3. Adobe Substance 3D Painter

Adobe Substance 3D Painter es un software creado por Adobe Inc. que permite pintar directamente sobre el modelo 3D, generando mapas de color, normales, rugosidad, metálicos y otros canales necesarios para lograr materiales realistas. Hoy en día, es una de las herramientas más usadas para la creación de modelos en videojuegos, ya que permite crear materiales con buen nivel de detalle utilizando un flujo de trabajo basado en materiales PBR (*Physically Based Rendering*), que es compatible con motores gráficos modernos [41].

- Pintura directa sobre modelos 3D.
- Materiales inteligentes (*Smart Materials*).
- Generadores y máscaras procedurales.
- Compatibilidad con flujo de trabajo PBR.

- Exportación de texturas para Unity, Unreal Engine y otros motores gráficos.

Ilustración 16

Principales interfaces del motor gráfico de Adobe Substance 3D Painter



Nota: Obtenido de Adobe Substance 3D Painter [41]

2.5. Arquitecturas, modelos y estándares

2.5.1. Arquitectura de personajes

La arquitectura de personajes se organiza mediante diferentes componentes, entre ellos el cuerpo, el rostro, la vestimenta, los accesorios, el esqueleto, los materiales y las animaciones. Esta organización permite estructurar los elementos necesarios para preparar los personajes, facilitar la creación e integración de cada componente, mejorando el flujo de trabajo y garantizando una implementación adecuada dentro del motor gráfico [42] [43].

Arquitectura de personaje

*Ilustración 17**Ejemplo de la Arquitectura de personajes*

Nota: Elaboración propia

2.5.2. Arquitectura de escenarios

La arquitectura de escenarios consiste en estructurar el entorno virtual mediante la integración de diferentes recursos que conforman el videojuego como viviendas, pilares, muelles, caminos y otros objetos decorativos. Esta organización permite administrar mejor los recursos del videojuego, optimizar el rendimiento y mantener una distribución coherente de todos los elementos del escenario [44].

Ilustración 18

Ejemplo de la arquitectura de escenarios



Nota: Elaboración propia

2.5.3. Modelos basados en referencias

El modelo basado en referencias usa fotografías, bocetos, arte conceptual y referencias anatómicas como guía durante el proceso de modelado 3D. Este enfoque ayuda a mejorar las proporciones, mantener la coherencia visual y reducir errores al crear personajes y escenarios para videojuegos [44] [45].

Ilustración 19

Ejemplos de la definición de modelos basados en referencias



El uso de referencias reales, bocetos, estudios anatómicos y arte conceptual permite crear un modelo 3D coherente, creíble y visualmente consistente.

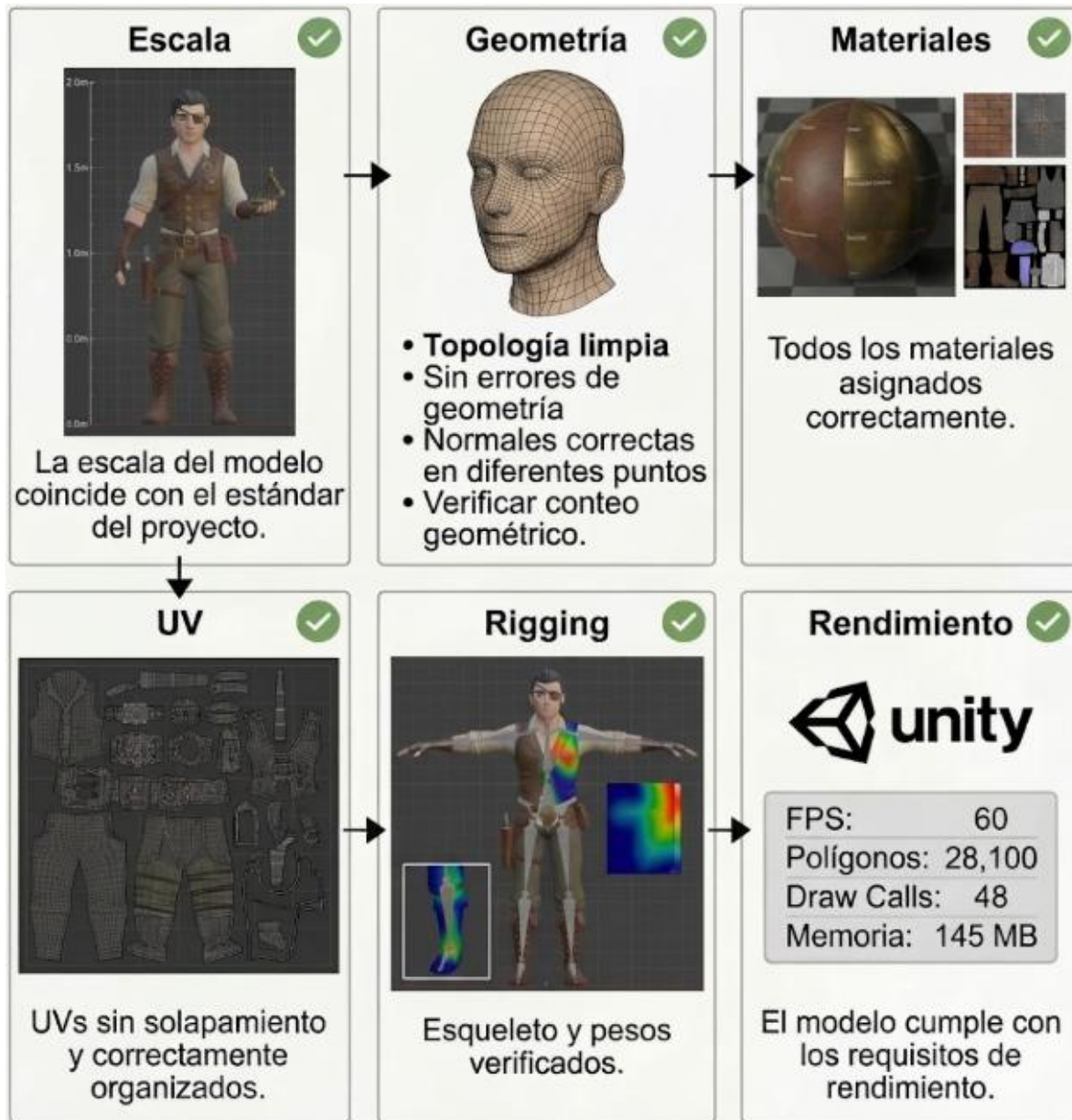
Nota: Elaboración propia

2.5.4. Modelo de validación técnica

El modelo de validación técnica comprueba que los modelos 3D cumplan con los requisitos establecidos antes de integrarlos al motor gráfico. Para esto, se revisan aspectos como la escala, la cantidad de polígonos, el mapeado UV, los materiales, el *rigging* y el rendimiento. Así se asegura la calidad y el buen funcionamiento de los modelos dentro del videojuego [44] [46].

Ilustración 20

Ejemplo de modelo de validación técnica



Nota: Elaboración propia

2.5.5. Estándar de escala y unidades

El estándar de escala y unidades dice que todos los modelos 3D deben conservar las mismas medidas durante su desarrollo. Esto asegura que sean compatibles con el motor gráfico. Seguir este estándar evita que haya diferencias entre personajes, escenarios y objetos.

También hace más fácil integrar los modelos y que las físicas y colisiones funcionen bien [44].

Ilustración 21

Ejemplo de definición de estándar de escala y unidades

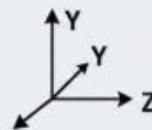


Escala del personaje y barril: 2.0 metros de altura

Estándares del proyecto	
Unidad del proyecto	1 unidad = 1 metro
Altura del personaje	2.0 m
Altura de barril	1.0 m
Altura de muro	5.0 m
Escala de objetos	Basada en metros

Objetivo

Mantener una escala consistente en todos los modelos para asegurar coherencia visual, colisiones correctas y jugabilidad adecuada dentro del motor gráfico.



Nota: Elaboración propia

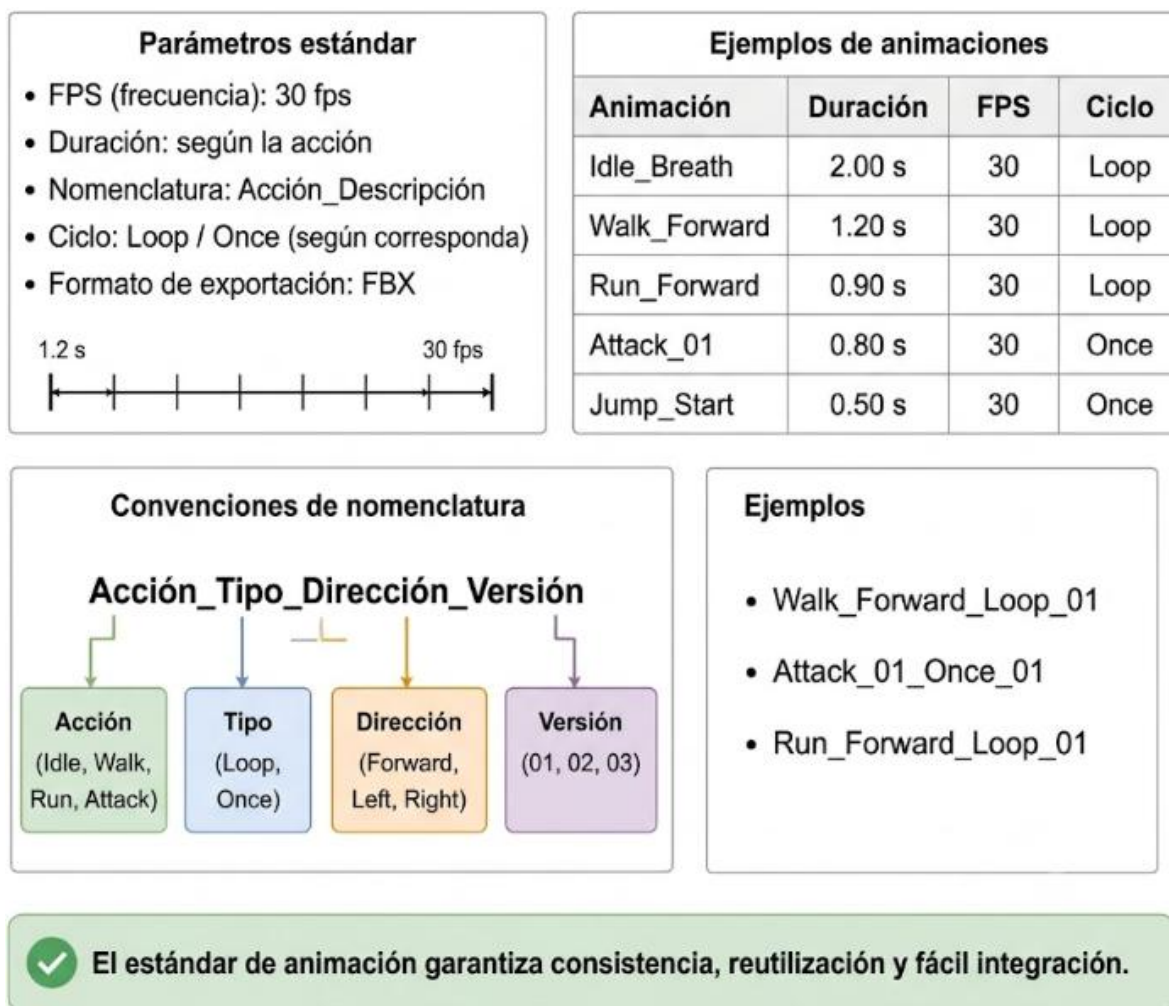
2.5.6. Estándar de animación

El estándar de animación define criterios sobre la frecuencia de fotogramas, la duración de las secuencias y los ciclos de animación. Aplicar este estándar ayuda a mantener

la uniformidad entre los distintos personajes. También mejora la organización de los recursos y facilita su integración en el motor gráfico [42] [47].

Ilustración 22

Ejemplo de definición de estándar de animación



Nota: Elaboración propia

2.5.7. Cuadro comparativo de arquitecturas, modelos y estándares

Tabla 3

Comparación entre las arquitecturas, modelos y estándares

Elemento	Propósito	Aplicación en el proyecto
Arquitectura de personajes	Organizar los componentes que conforman un personaje	Facilita el desarrollo e integración de los personajes

	3D (cuerpo, vestimenta, <i>rigging</i> , materiales y animaciones)	en el videojuego “ <i>Ashes of Faith</i> ”
Arquitectura de escenarios	Estructurar los elementos que conforman el entorno del videojuego para mantener una distribución organizada.	Permite optimizar el rendimiento y administrar eficientemente los recursos del escenario.
Modelos basados en referencias	Utilizar imágenes, bocetos y arte conceptual como guía durante el modelado 3D.	Ayuda a obtener modelos con proporciones correctas y una mayor coherencia visual
Modelo de validación técnica	Verificar que los modelos cumplan requisitos técnicos antes de integrarlos al motor gráfico.	Garantiza la calidad, compatibilidad y correcto funcionamiento de los recursos en Unity.
Estándar de escala y unidades	Mantener dimensiones uniformes en todos los modelos tridimensionales.	Evita inconsistencias de tamaño y mejora la integración con físicas y colisiones del videojuego.
Estándar de animación	Definir criterios para la creación y reproducción de animaciones.	Favorece la uniformidad de las animaciones y facilita su implementación en el motor gráfico.

Nota: Elaboración propia

2.6. Herramientas de desarrollo

2.6.1. Cuadro comparativo de Tecnologías

Tabla 4

Comparación de software de modelado 3D

Características	Blender	Autodesk Maya	Autodesk 3ds Max
Licencia	Software libre y de código abierto	Licencia comercial	Licencia comercial
Costo	Gratuito	Suscripción de pago	Suscripción de pago
Sistema operativo	Windows, macOS y Linux	Windows, macOS y Linux	Windows
Modelado 3D	Modelo poligonal, esculpido y edición de mallas	Modelo poligonal, NURBS y esculpido	Modelo poligonal y edición mediante modificadores
Texturizado y materiales	Editor UV y sistema de nodos para materiales	Herramientas avanzadas de materiales y UV	Editor de materiales y mapeo UV
<i>Rigging</i> y animación	Sistema completo de <i>rigging</i> , animación y cinemática inversa	Herramientas profesionales para	Animaciones mediante

		personajes y animación	fotogramas clave y sistemas de <i>rigging</i>
Renderizado	Cycles y Eevee integrados	Arnold Render	Arnold Render
Compatibilidad	FBX, OBJ, STL, GLTF, Alembic entre otros	FBX, OBJ, Alembic y otros formatos	FBX, OBJ, 3DS y otros formatos
Integración con motores gráficos	Unity, Unreal Engine y Godot	Unity y Unreal Engine	Unity y Unreal Engine
Curva de aprendizaje	Intermedia con amplia documentación gratuita	Intermedia-Alta	Intermedia
Comunidad y soporte	Comunidad internacional muy activa y abundante documentación	Soporte oficial Autodesk y comunidad profesional	Soporte oficial Autodesk y comunidad profesional
Uso de la industria	Videojuegos independientes, animación, educación y producción audiovisual	Estudios de videojuegos AAA, cine y animación profesional	Arquitectura, visualización, videojuegos y diseño 3D
Ventajas	Gratuito, completo, actualizaciones constantes e integración de todo el flujo de trabajo	Herramientas avanzadas para animación y producción profesional	Potente para modelado arquitectónico y visualización 3D
Limitaciones	Algunas funciones avanzadas dependen de complementos especializados	Alto costo de licencia	Disponible únicamente para Windows y con licencia comercial

Nota: Elaboración propia

Tras analizar las características de las diferentes herramientas de modelado 3D, se decidió que Blender era la opción correcta, ya que su integración de herramientas para modelado, texturizado, *rigging* y animación ayuda a optimizar el flujo de trabajo durante el proyecto. Además de ser un software libre de código abierto y actualizaciones continuas tiene una comunidad activa que ayuda con documentación y recursos de apoyo.

2.6.2. Cuadro comparativo de Herramientas de texturizado

Tabla 5

Comparación de software de texturizado

Características	Adobe Photoshop	Krita	Adobe Substance 3D Painter
Licencia	Comercial	Software libre y de código abierto	Comercial
Costo	Suscripción mensual (Adobe Creative Cloud)	Gratuito	Suscripción mensual (Adobe Substance 3D Collection)
Sistema operativo	Windows y macOS	Windows, macOS y Linux	Windows y macOS
Función principal	Edición de imágenes y creación de texturas	Ilustración digital y pintura de texturas	Texturizado 3D basado en materiales PBR
Pintura sobre modelos 3D	No	No	Si
Compatibilidad con PBR	Limitada (requiere trabajo manual)	Limitada	Completa
Capas y máscaras	Si	Si	Si
Biblioteca de materiales	No	No	Sí, incluye materiales inteligentes (Smart Materials)
Compatibilidad con motores gráficos	Compatible mediante exportación de imágenes	Compatible mediante exportación de imágenes	Exportación directa para Unity, Unreal Engine y otros motores
Facilidad de aprendizaje	Intermedia	Fácil	Intermedia
Uso en la industria	Diseño gráfico, edición de imágenes y texturas	Ilustración digital, concept art y texturas	Videojuegos, cine y producción 3D profesional
Ventajas	Gran variedad de herramientas de edición y amplia compatibilidad de formatos	Gratuito, ligero y con excelentes herramientas de pintura digital	Pintura directa sobre modelos 3D, flujo de trabajo PBR y materiales inteligentes
Limitaciones	No permite pintar directamente sobre modelos 3D	No cuenta con herramientas específicas para texturizado PBR	Requiere licencia comercial y mayor capacidad de hardware

Nota: Elaboración propia

Tras comparar las diferencias herramientas de texturizado, se decidió utilizar Adobe Photoshop debido a que esta herramienta tiene las funciones necesarias para crear, editar y mejorar las texturas para los modelos 3D. Aunque Substance 3D Painter ofrece herramientas más especializadas para el texturizado PBR y la pintura directa sobre modelos 3D, Photoshop se ajustó mejor a las necesidades del proyecto, principalmente por su compatibilidad con Blender. Además, su uso permitió mantener un flujo de trabajo sencillo para la creación y adaptación de las texturas utilizadas en los modelos.

2.7. Marco conceptual

Con el propósito de facilitar los términos usados en esta investigación, se presenta el siguiente marco conceptual. Aquí se reúnen los términos principales sobre modelado 3D, creación de personajes y escenarios, técnicas de optimización y las herramientas que se usaron durante el desarrollo del videojuego. Estos conceptos sirven como referencia para entender los temas tratados en este trabajo.

Tabla 6

Marco conceptual

Concepto	Definición aplicada al proyecto
Modelado 3D	Proceso de crear objetos 3D mediante software especializado, definiendo su forma, geometría y estructura para ser utilizados en entornos virtuales.
Blender	Software libre y de código abierto utilizado para modelado, esculpido, texturizado, <i>rigging</i> , animación y renderizado de modelos tridimensionales
Videojuego	Aplicación interactiva que integra elementos gráficos, mecánicas y narrativa para ofrecer una experiencia de entretenimiento al usuario.
Personaje 3D	Modelo digital que representa un personaje dentro del videojuego, compuesto por geometría, materiales, esqueleto y animaciones.
Escenario 3D	Entorno virtual conformado por elementos como edificios, vegetación, caminos, rocas y objetos decorativos que sirven como espacio de interacción para el jugador.

<i>Assets</i>	Elemento digital reutilizable, como personajes, escenarios, objetos o materiales, que forma parte del desarrollo del videojuego.
Low Poly	Técnica de modelado que utiliza una cantidad reducida de polígonos para optimizar el rendimiento sin perder la identidad visual del modelo.
Mapeado UV	Proceso mediante el cual la superficie de un modelo 3D se despliega en dos dimensiones para aplicar correctamente las texturas.
Texturizado	Proceso de aplicar imágenes o materiales sobre un modelo tridimensional para definir su apariencia visual, color y detalles superficiales.
<i>Rigging</i>	Técnica que consiste en crear un esqueleto digital para permitir el movimiento y la animación de un modelo tridimensional.
Animación 3D	Proceso de generar movimiento en personajes u objetos mediante la manipulación del esqueleto y la interpolación de fotogramas.
Motor gráfico	Software encargado de procesar e integrar gráficos, físicas, animaciones e interacción para ejecutar un videojuego en tiempo real.
Optimización	Conjunto de técnicas destinadas a reducir el consumo de recursos del modelo 3D sin afectar significativamente su calidad visual.
Unity	Motor gráfico utilizado para integrar los modelos 3D, animaciones y demás recursos que conforman el videojuego “ <i>Ashes of Faith</i> ”.

Nota: Elaboración propia

2.8. Relación entre la teoría y la propuesta tecnológica

Los temas desarrollados en este capítulo sirvieron como base para el desarrollo de la propuesta tecnológica, proporcionando los conocimientos necesarios para la aplicación de técnicas de modelado 3D, optimización de recursos, *rigging*, animación y el uso de motores de videojuegos. Estos fundamentos permitieron definir los procedimientos para la creación de personajes, escenarios y objetos y facilitaron la integración de los recursos a Unity, obteniendo un buen rendimiento y una mejor calidad visual del proyecto.

Capítulo III

3. Manual de usuario

Este capítulo documenta los parámetros necesarios para la correcta ejecución y navegación dentro del videojuego. Aquí se detallan las especificaciones técnicas de usabilidad, el esquema de los controles, la distribución de las interfaces de usuario (UI/HUD) y como recorrer por las diversas pantallas del videojuego.

3.1. Especificaciones de uso

Para la ejecución del videojuego y la su correcta ejecución en el entorno tridimensional, se establecen los siguientes requerimientos de hardware y periféricos:

Tabla 7

Componentes para la ejecución del videojuego

Componente / Parámetro	Detalle de Especificación
Dispositivo de Entrada	Teclado estándar QWERTY y Ratón (Mouse) de dos botones con rueda de desplazamiento (Scroll)
Entorno de Despliegue	PC
Resolución recomendada	1920 x 1080 pixeles (Relación de aspecto 16:9)

Nota: Elaboración propia

3.2. Configuración de Controles

La ejecución de acciones se realiza utilizando un teclado alfanumérico para el desplazamiento espacial del personaje jugable y la orientación del ángulo visual de la cámara, junto a los botones del ratón que sirven para las acciones de combate.

Tabla 8

Esquema de controles e interfaz de entrada

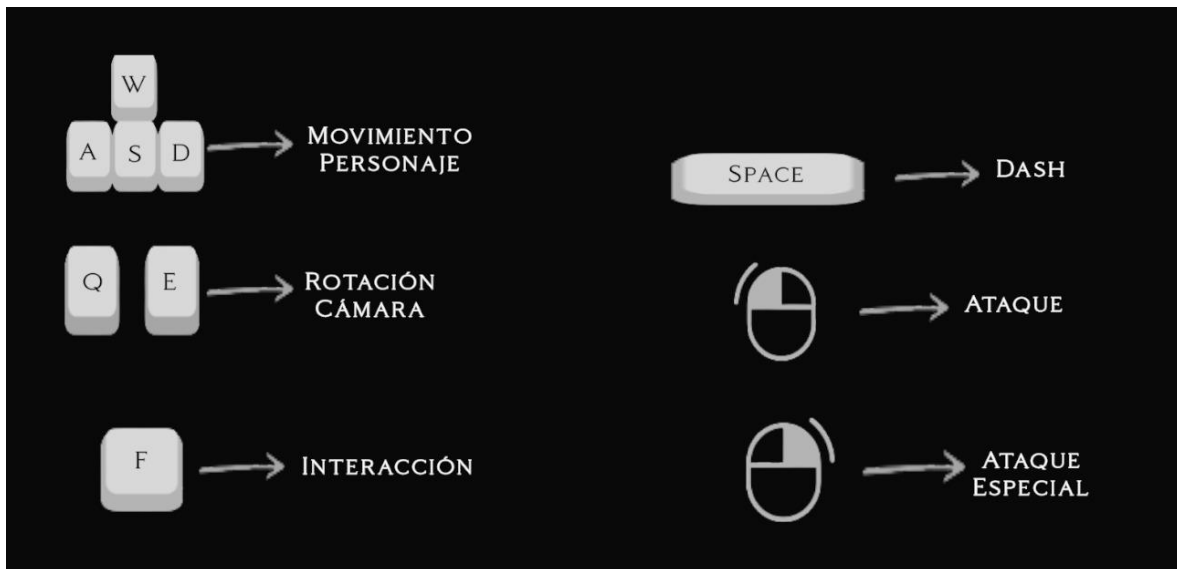
Acción Mecánica	Periférico / Tecla	Descripción Funcional dentro del Sistema
-----------------	--------------------	--

Movimiento de Personaje	W, A, S, D	Controla el movimiento del personaje jugable en el plano tridimensional (X, Z).
Rotación de Cámara	Q, E	Ajusta la perspectiva visual y la orientación angular de la cámara.
Interacción	F	Activa el diálogo con personajes no jugables (NPCs) e interacción con objetos del escenario.
Esquive / Impulso (Dash)	SPACE	Ejecuta un desplazamiento rápido para evadir ataques enemigos.
Ataque Básico	Clic Izquierdo	Activa la secuencia de ataque principal con el arma equipada
Ataque Especial	Clic Derecho	Despliega la técnica especial sujeta al tiempo de recarga (Cooldown).

Nota: Elaboración propia

Ilustración 23

Esquema y distribución de los botones



Nota: Elaboración propia

3.3. Interfaz de Usuario (UI) y HUD

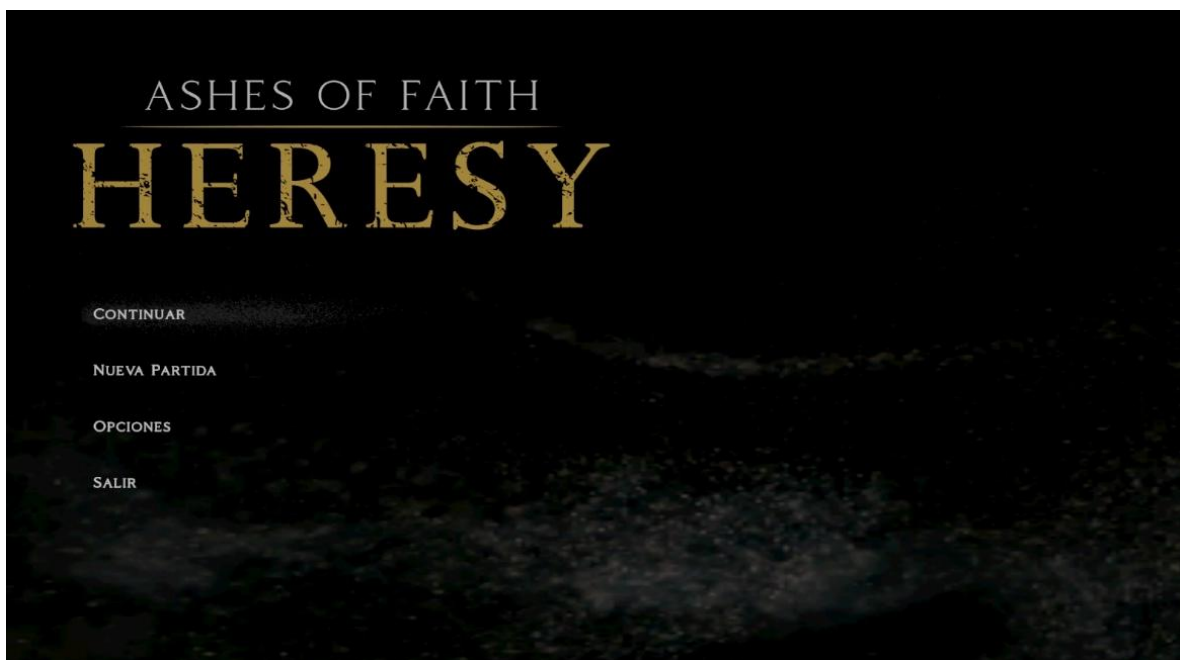
La interfaz gráfica presenta la información de las pantallas del juego desde la pantalla de inicio, los elementos permanentes en pantalla durante el *gameplay* (*Heads-Up Display* - *HUD*) y las pantallas de pausa para la gestión de información (menús).

3.3.1. Menú de inicio del juego

- **Continuar:** Permite reanudar la sesión de juego en el último punto de guardado donde se quedó el jugador la última vez que jugó.
- **Nueva Partida:** Empieza la historia del juego desde cero, comienza una nueva aventura.
- **Opciones:** Muestra un panel de ajustes técnicos configurables como sonido, video y controles.
- **Salir:** Cierra el videojuego de forma segura y vuelve al escritorio de Windows.

Ilustración 24

Interfaz gráfica del menú de inicio



Nota: Elaboración propia

3.3.2. Indicadores del HUD

- **Barra de Vida (Salud):** Se encuentra ubicada en la parte inferior izquierda de la pantalla. Muestra la salud del jugador dentro de un marco alargado oscuro y metálico.
- **Indicador de Poder (Cooldown):** Ubicado por encima de la barra de vida. Indica si el ataque especial está disponible para usarse y el tiempo de espera para volver a usarlo.

Ilustración 25

Vista de la perspectiva de cámara e indicadores del HUD (salud y cooldown)

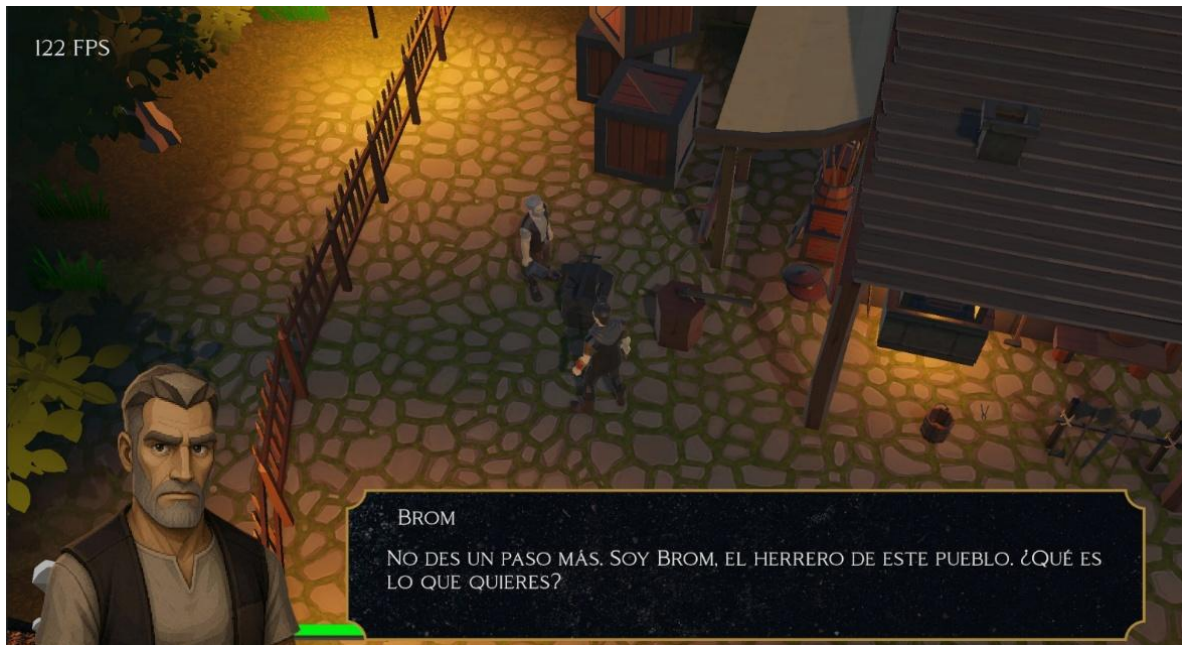


Nota: Elaboración propia

- **Panel de Diálogos:** Franja horizontal oscura con borde dorado que se despliega en la parte inferior de la pantalla al presionar la tecla de interacción (F) frente a un personaje no jugable (NPC). Su propósito es mostrar los diálogos entre el jugador y los NPCs.

Ilustración 26

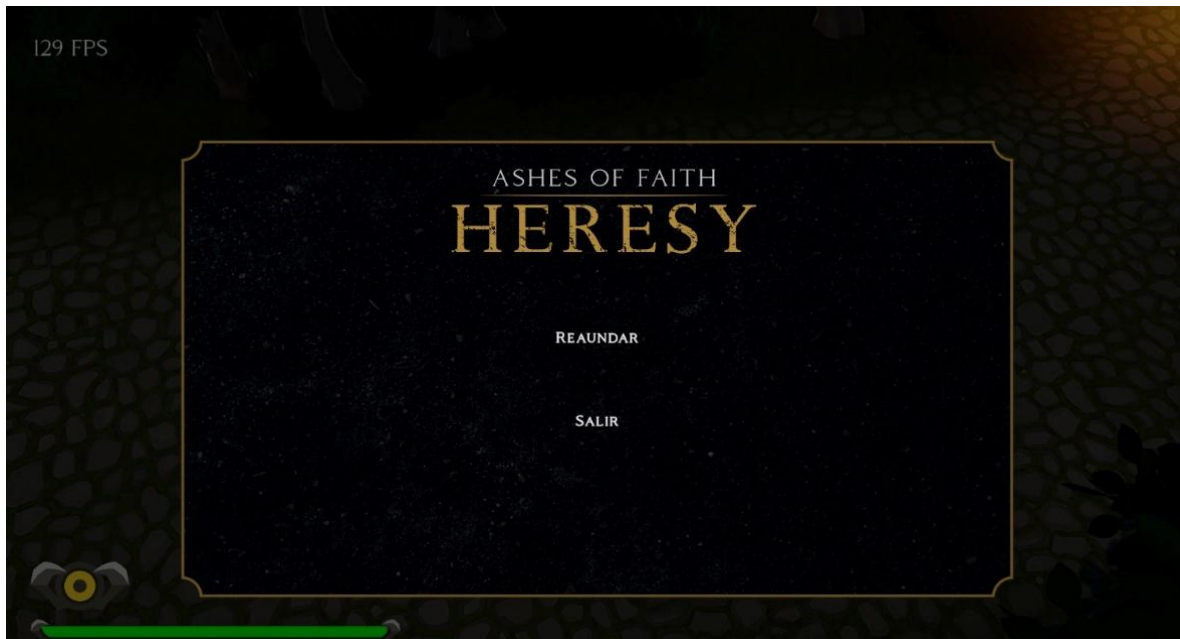
Ejecución del sistema de diálogos con el NPC Brom



Nota: Elaboración propia

3.3.3. Sistema de Menú de Pausa

- **Reanudar:** Permite volver a la pantalla de juego de forma instantánea, cerrando el menú de pausa y retomando el juego.
- **Salir:** Finaliza la partida de juego y regresa al menú principal.

*Ilustración 27**Interfaz de pausa*

Nota: Elaboración propia

3.4. Procedimiento de Instalación

- **Ejecutar el archivo de Instalación:**
- Con el archivo ejecutable de instalación *AshesOfFaith_Setup.exe*, hacer doble clic sobre él para ejecutarlo (o seleccionar la opción Ejecutar como Administrador para otorgar los permisos del sistema).
- **Configuración de la Ruta de Instalación:**
- En la pantalla de bienvenida del instalador, hacer clic en el botón Siguiente.
- Definir la ruta donde se colocarán los archivos del juego (por defecto: *C:\Program Files\Ashes of Faith: Heresy*). Si se desea cambiar la ubicación de la instalación, seleccionar Examinar y elegir la nueva ubicación en el directorio.
- **Copias de Seguridad y Accesos Directos:**

- Seleccionar la casilla Crear acceso directo en el Escritorio para crear un icono de acceso rápido en el escritorio y facilitar el ingreso posterior al videojuego.
- Hacer clic en el botón Instalar para iniciar el proceso de instalación de datos, copia de librerías DLL y registro de recursos del motor.
- **Finalización de la Instalación y Puesta en Marcha:**
- Una vez completado el proceso de instalación, la interfaz del asistente mostrará la pantalla de finalización.
- Marcar la casilla Ejecutar *Ashes of Faith: Heresy* y hacer clic en Finalizar.
- Entonces el aplicativo estará listo para su ejecución.

Capítulo IV

4. Manual de programador

En este capítulo se describe el proceso utilizado para desarrollar los recursos 3D del videojuego. Se presentan las diferentes etapas realizadas para la creación de personajes, escenarios y objetos, incluyendo el modelado texturizado, *rigging*, animación, optimización e integración en Unity. También se explica el flujo de trabajo utilizado y la aplicación de la metodología Scrum para la organización de las actividades del proyecto. Finalmente, se presentan las pruebas realizadas para comprobar el funcionamiento y rendimiento de los recursos desarrollados dentro del entorno interactivo.

4.1. Dirección de arte y propuesta visual

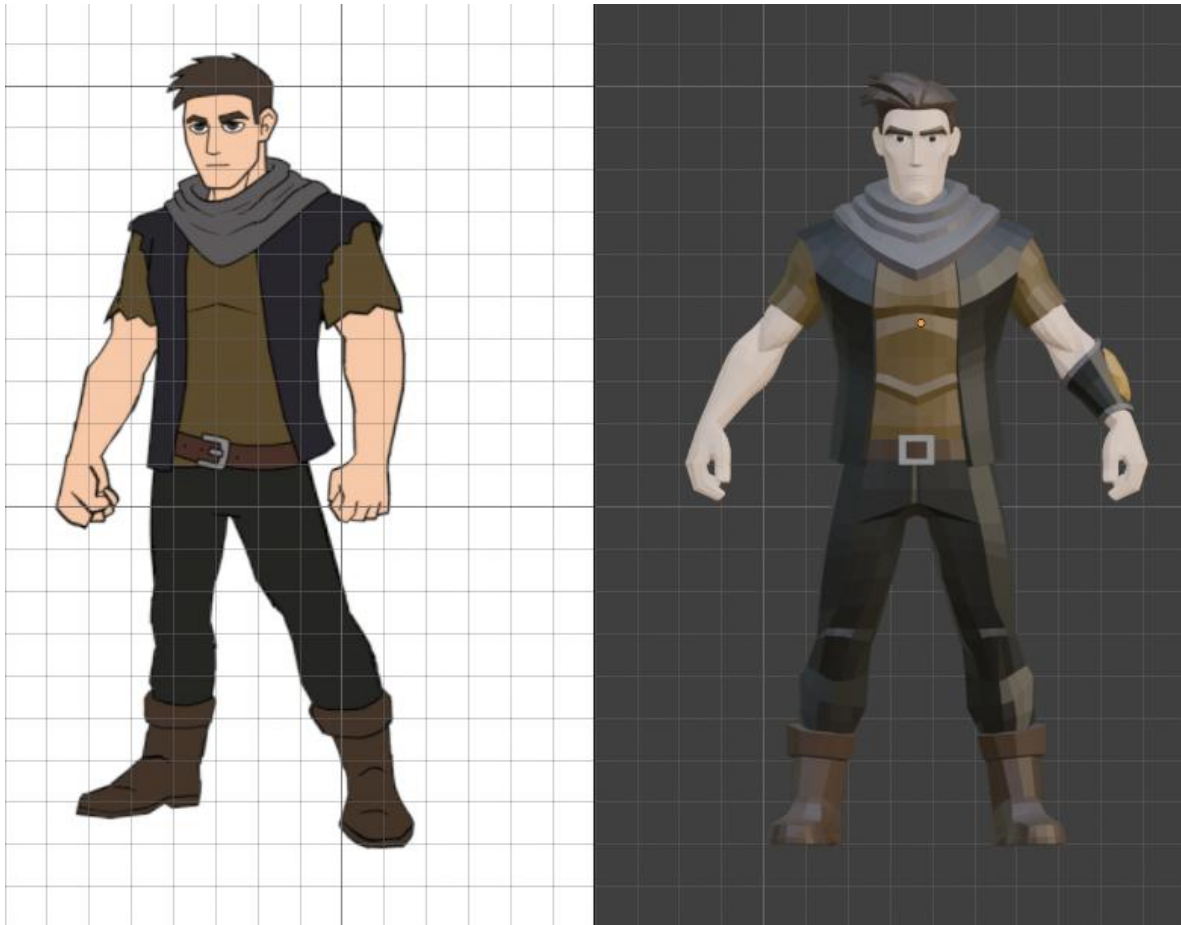
El arte del videojuego a nivel de diseño tiene el propósito de mantener una calidad visual coherente entre personajes, escenarios y objetos. Para ello se definió una propuesta visual de estilo 3D de poligonaje bajo (*Low Poly*), buscando que los recursos mantengan una apariencia uniforme y puedan integrarse correctamente en un entorno interactivo del videojuego.

La propuesta visual se orienta hacia una ambientación de fantasía con elementos medievales, utilizando formas simples y detalles que permitan diferenciar los distintos recursos del juego. Este estilo (*Low Poly*) se aplicó en la creación de los personajes, casas, escenarios y objetos, manteniendo una relación visual en todos los modelos 3D.

Además, durante el desarrollo se optimizaron varios modelos, reduciendo la cantidad de polígonos, la organización de las mallas y el nivel de detalle. De esta manera se buscó mantener un equilibrio entre calidad visual y el rendimiento técnico, para que los recursos puedan integrarse correctamente dentro del motor gráfico Unity.

Ilustración 28

Presentación visual del diseño 2D y el modelado 3D



Nota: Elaboración propia

La figura 22 muestra el estilo artístico Low Poly aplicado al personaje, manteniendo las características principales del diseño 2D y el modelado 3D.

4.2. Diseño de personajes, escenarios y elementos gráficos

El diseño visual de los personajes se realiza a partir de la propuesta visual establecida por un miembro del equipo de trabajo, con el propósito de mantener un buen estilo entre los diferentes elementos del proyecto. A partir de los diseños se planificó el desarrollo de los modelos 3D, procurando respetar la identidad visual planteada desde el inicio del proyecto.

En el diseño de personajes, se definieron sus características visuales y proporciones según la función que cumplen en el videojuego. También se desarrollaron objetos como barriles, cajas, armas, herramientas y otros elementos decorativos que complementan los espacios y enriquecen la ambientación del mundo virtual.

Durante este proceso, se busca que los personajes, escenarios y objetos mantengan una relación visual adecuada para su modelado e integración en el videojuego. Así, el diseño de los personajes sirvió como base para las siguientes etapas. Esto facilitó la organización del desarrollo de los modelos 3D.

Ilustración 29

Presentación visual de los modelos 3D para el videojuego “Ashes of Faith”



Nota: Elaboración propia

La figura 23 presenta el conjunto de personajes, escenarios y elementos gráficos desarrollado para el videojuego. Todos los modelos 3D tienen un estilo artístico Low Poly, manteniendo una identidad visual coherente.

Tabla 9*Clasificación de los modelos 3D desarrollados para el videojuego*

Categoría	Cantidad de modelos	Total de vértices	Descripción
Personaje principal	1	5,125	Personaje jugable desarrollado para el videojuego
NPCs	5	34,275	Personajes no jugables utilizados para la interacción con el jugador
Enemigos	5	14,794	Personajes enemigos utilizados para el combate
Viviendas del pueblo	9	25,728	Construcciones de estructuras para el pueblo
Viviendas del puerto	9	36,968	Construcciones de estructuras para el puerto
Viviendas con interiores	3	29,243	Viviendas con espacios interiores explorables
Iglesia	1	5,721	Edificación principal del pueblo
Barco grande y bote	2	2,392	Embarcaciones utilizadas como parte del escenario del muelle
Estructura del muelle	3	2,132	Plataforma, gradas y puente colgante de madera del muelle
Muros y puentes	1 conjunto	633	Estructura de piedra para delimitar el escenario
Grupo de armas	1 conjunto	502	Espadas, lanzas, escudos y otros elementos decorativos
Grupo de barriles, cajas, carreta, pozo de agua y una estatua	1 conjunto	8,261	Objetos utilizados para ambientar los escenarios

Grupo de mercado	1 conjunto	1,688	Puestos, telas, mesas y elementos comerciales
Grupo de cementerio	1 conjunto	2,074	Tumbas y elementos decorativos del cementerio

Nota: Elaboración propia

4.3. Producción de *assets* 3D

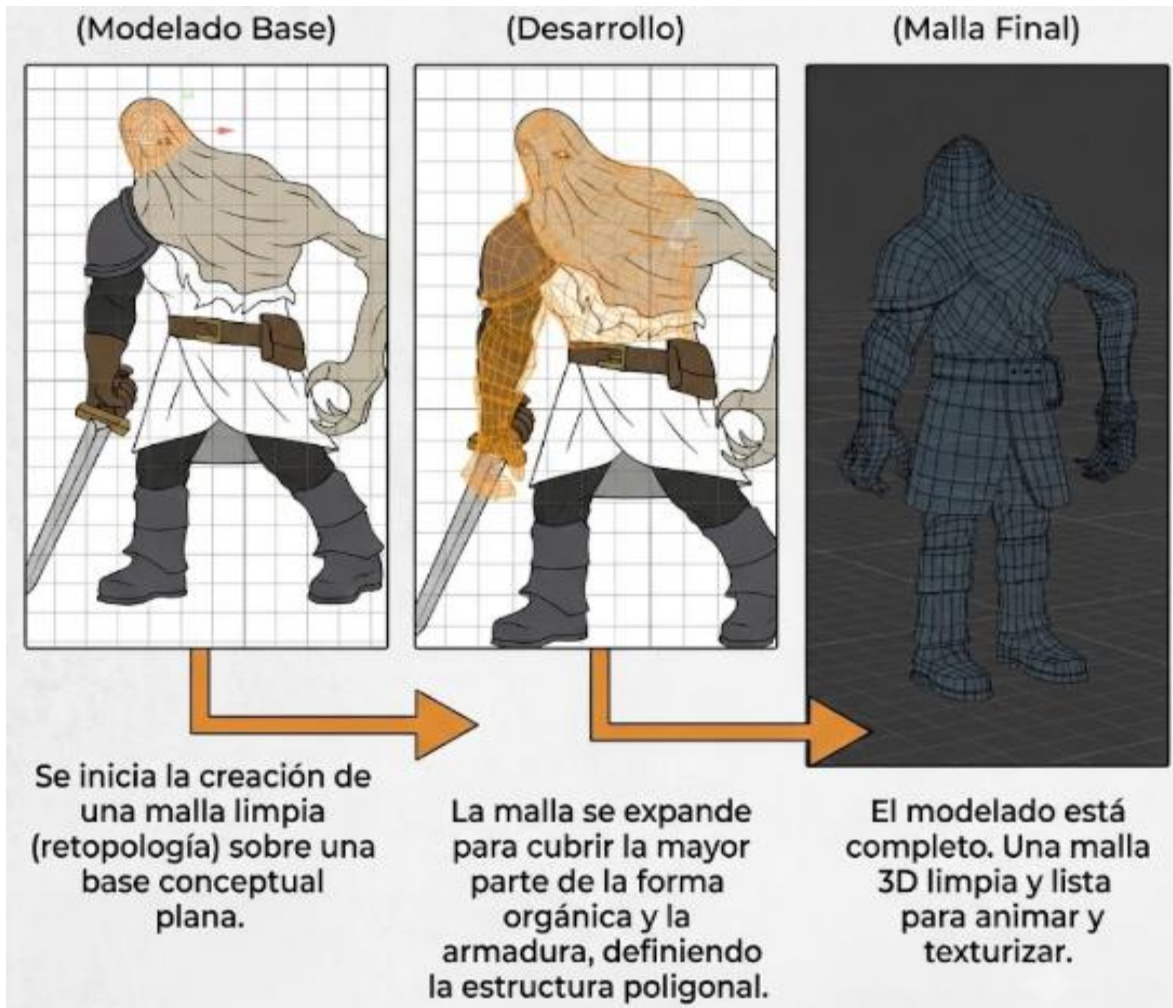
La producción de los recursos visuales comenzó a partir de los bocetos de los personajes elaborados por un miembro del equipo encargado del diseño. Estos sirvieron como referencia para desarrollar los modelos 3D en Blender, respetando las características visuales realizadas en la etapa de diseño. En el caso de los escenarios y objetos, su producción se realizó siguiendo la propuesta visual establecida para el proyecto, para mantener una identidad gráfica uniforme en el videojuego.

El proceso de producción incluyó varias etapas, como el modelado 3D, la creación de mapas UV, la aplicación de materiales y texturas, y la optimización de los modelos. En el caso de los personajes, se realizaron procesos de *rigging* y animación para permitir su uso dentro del entorno interactivo. También se desarrollaron recursos como casas y objetos decorativos para enriquecer los escenarios del videojuego.

Como resultado, se obtuvieron diferentes recursos 3D que formarán el contenido visual. Estos elementos fueron preparados para continuar con las etapas de animación, optimización e integración en el motor Unity, asegurando la calidad visual de los modelos y los requerimientos técnicos del proyecto.

Ilustración 30

Desarrollo del modelado 3D



Nota: Elaboración propia

4.4. Pipeline de producción artística

El desarrollo del proyecto se realiza mediante el flujo de trabajo organizado en diferentes etapas. El proceso comenzó con la planificación de los recursos por desarrollar, seguido del diseño y modelado 3D de personajes, escenarios y objetos. Luego, se realizó el texturizado, optimización, *rigging* y animación según la necesidad de cada recurso.

Este flujo de trabajo organizó las actividades de producción mediante *sprints* de la metodología Scrum. Cada etapa avanzó progresivamente, permitiendo revisar los avances y

hacer ajustes necesarios antes de seguir con las siguientes actividades. De esta manera se mantuvo un orden desde la creación de mallas hasta la preparación para su integración en el videojuego.

Finalmente, los modelos y animaciones desarrollados se prepararon para su exportación e integración en Unity, donde se realizaron pruebas de funcionamiento. Este flujo de trabajo mantuvo un proceso de producción ordenado y facilitó la coordinación con el equipo de trabajo para obtener recursos visuales funcionales para el proyecto.

Tabla 10

Desarrollo del Product Backlog

PRODUCT BACKLOG				
Título Del Proyecto: Aplicación de técnicas de modelado y animación tridimensional en Blender para el desarrollo optimizado de personajes y entornos interactivos en tiempo real.			Fecha de preparación: 07/05/2026	
ID	Descripción resumida	Propiedad	Historia	Estado
US-01	Como artista 3D necesito de un personaje protagonista en detallado y optimizado para un videojuego.	Alta	Modelado de personaje principal	Terminado
US-02	Como artista 3D necesito 5 NPCs para integrar a un videojuego.	Alta	Modelado de NPCs	Terminado
US-03	Como artista 3D necesito 5 enemigos modelados correctamente para integrar dentro del juego.	Alta	Modelado de enemigos	Terminado
US-04	Como artista 3D necesito optimizar la geometría del personaje protagonista mediante reducción de polígonos, control topológico y estructuración eficiente de mallas para mejorar el rendimiento gráfico.	Alta	Optimización de mallas del personaje principal	Terminado
US-05	Como artista 3D necesito optimizar la geometría de los NPCs y enemigos mediante reducción de polígonos, control topológico y estructuración eficiente de mallas para mejorar el rendimiento gráfico.	Alta	Optimización de NPCs y enemigos	Terminado
US-06	Como artista 3D necesito crear <i>assets</i> y escenarios	Alta	Creación de <i>assets</i> y escenarios 3D	Terminado

	tridimensionales modulares para su integración dentro de un entorno interactivo del videojuego.			
US-07	Como artista 3D necesito aplicar materiales y texturas PBR optimizadas para mejorar la calidad visual y coherencia estética de los modelos tridimensionales.	Alta	Texturizado y materiales PBR	Terminado
US-08	Como animador 3D necesito implementar sistemas de <i>rigging</i> funcional mediante esqueletos y controles de deformación para animar correctamente los personajes.	Alta	Sistema de <i>rigging</i>	Terminado
US-09	Como animador 3D necesito desarrollar animaciones básicas para el personaje protagonista con movimientos fluidos y coherentes para la jugabilidad.	Alta	Animaciones básicas del personaje principal	Terminado
US-10	Como animador 3D necesito desarrollar animaciones funcionales para NPCs y enemigos que permitan una correcta interacción dentro del entorno del videojuego.	Alta	Animaciones de NPCs y enemigos	Terminado
US-11	Como desarrollador necesito integrar correctamente los modelos de personajes jugables y no jugables con su respectiva animación en Unity para validar su funcionamiento.	Alta	Integración de personajes en Unity	Terminado
US-12	Como desarrollador necesito integrar <i>assets</i> y escenarios tridimensionales dentro de Unity para validar compatibilidad, rendimiento y funcionamiento visual.	Alta	Integración de <i>assets</i> en Unity	Terminado
US-13	Como desarrollador necesito realizar pruebas de optimización gráfica mediante análisis de FPS, batches y rendimiento para garantizar estabilidad dentro del entorno interactivo.	Media	Optimización y pruebas	Terminado

Nota: Elaboración propia

La tabla 10 presenta *Product Backlog* del proyecto, donde se organizan las historias de usuario que guían el desarrollo de las actividades. Esta tabla sirve como base para la planificación de los *sprints*, estableciendo las tareas, prioridades y objetivos que se ejecutan

en cada interacción hasta completar el desarrollo del proyecto. A continuación, se muestra en la tabla 11 el plan de lanzamiento y desarrollo para el proyecto.

Tabla 11

Desarrollo y planificación de sprints

Componentes	ID, Sprint	Historia
Modelado y texturizado de personajes y escenarios	Sprint 1	US-01: Modelado de personaje principal
		US-02: Modelado de NPCs y enemigos
	Sprint 2	US-03: Optimización de mallas
		US-04: Creación de escenarios 3D
		US-05: Modelado de interiores
	Sprint 3	US-06: Creación de <i>assets</i> ambientales
		US-07: Texturizado y materiales
Animación	Sprint 4	US-08: Sistema de <i>rigging</i>
		US-09: Animaciones básicas
		US-10: Animaciones de combate
Integración y optimización	Sprint 5	US-11: Integración en Unity
		US-12: Optimización y pruebas

Nota: Elaboración propia

4.5. Preparación y animación de personajes y objetos

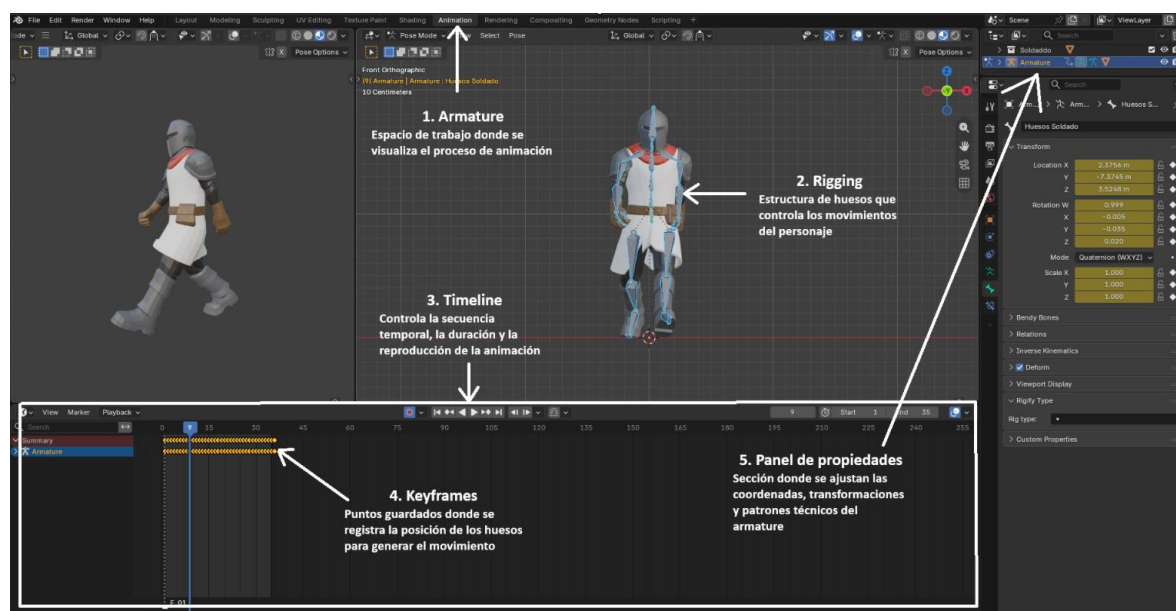
La preparación de los personajes 3D para la animación se realiza en Blender mediante la creación del *rigging* utilizando un *Armature*, la cual define la estructura de huesos que facilita el movimiento de los modelos. Después, se une el esqueleto con el modelo mediante la opción de *Parent with Automatic Weights*, generando los pesos para que la malla se deformara correctamente durante las animaciones. Para corregir estas deformaciones y mejorar el movimiento durante la animación, se usa la herramienta *Weight Paint*, esta herramienta permite ajustar manualmente el peso de cada hueso sobre la geometría del modelo, logrando deformaciones más naturales.

Una vez configurado el *rigging*, se desarrollan las animaciones básicas del personaje utilizando el (*Animation*) de Blender y se crean las animaciones en el apartado de (*Timeline*) que permite organizar la secuencia de fotogramas y controlar la duración de las animaciones como caminar, correr, atacar o mantenerse en reposo.

Durante esta etapa se realizan pruebas de movimiento y ajustes en el *rig* como en el peso de la malla para corregir deformaciones de los modelos y asegurar animaciones fluidas y naturales. Finalmente, los personajes animados se exportan en formato (FBX) para su integración en Unity, donde se verifica el funcionamiento de las animaciones.

Ilustración 31

Desarrollo de animación



Nota: Elaboración propia

4.6. Materiales, texturas, iluminación y renderización

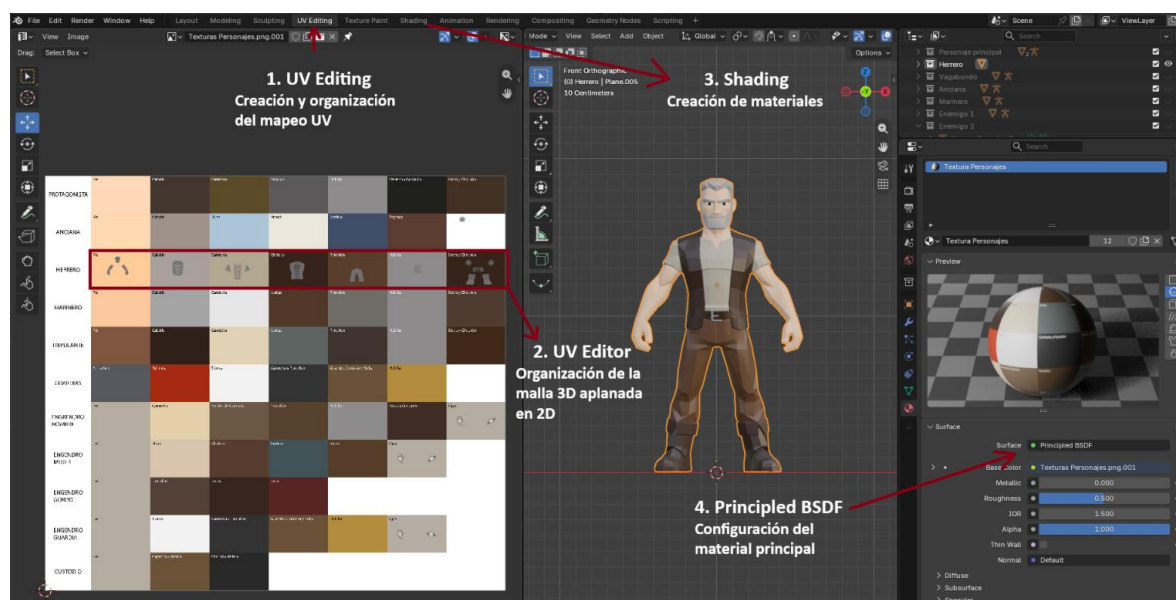
En esta etapa se aplican materiales y texturas a los modelos 3D para mejorar su estilo visual y mantener una relación coherente con el estilo artístico definido. Se utilizan distintos tipos de materiales según las características de los personajes, escenarios y objetos,

considerando elementos como madera, metal, piedra y otros materiales del entorno del videojuego.

Como primer paso se realiza el (*UV Mapping*) donde el modelo del personaje se despliega en un plano 2D mediante la herramienta de (*UV Editing*). Este proceso organiza las islas UV para colocar adecuadamente dentro del espacio de la textura y evitar distorsiones durante su aplicación.

Posteriormente, se asignan los materiales básicos al modelo utilizando el espacio de trabajo (*Shading*) y el nodo de (*Principled BSDF*), el cual permite configurar las propiedades principales del material. En esta etapa se crean materiales independientes para las diferentes partes del personaje, como la piel, la ropa, el cabello y los accesorios.

Como resultado, los modelos lograron una apariencia más definida con la propuesta visual del videojuego. La aplicación de este tipo de texturizado permitió mantener el estilo plano y estilizado de los personajes, además de preparar los modelos para las siguientes etapas del proceso de producción e integración a Unity.



Nota: Elaboración propia

4.7. Integración de recursos visuales

Una vez finalizado el modelado, texturizado y la animación de los personajes, los modelos 3D se exporta desde Blender en formato FBX, ya que este formato conserva la geometría, materiales y la animación para luego integrarse en el motor de Unity. Durante la exportación se configuran los parámetros correspondientes manteniendo la escala y la orientación correcta de los modelos.

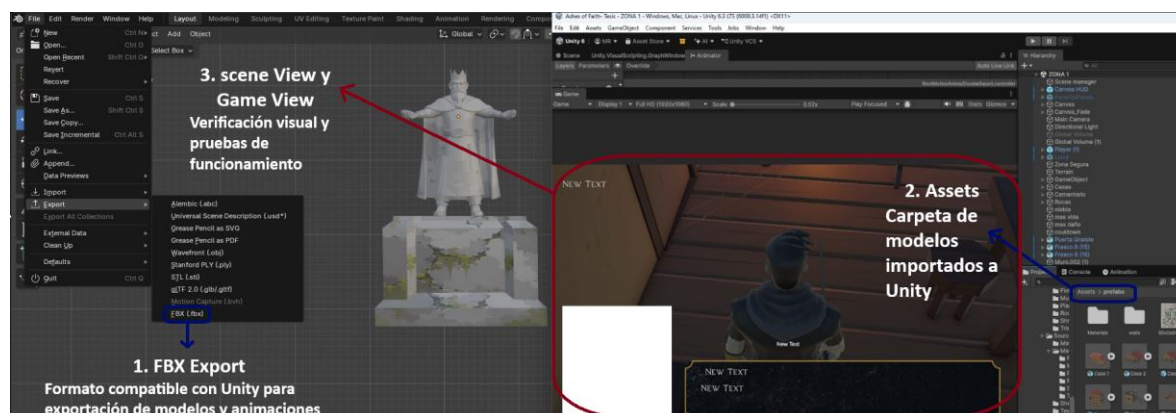
Una vez importados los archivos FBX y sus texturas, se configuran los componentes de cada personaje dentro de Unity. En el caso de los personajes animados, se asigna el tipo de *rig* correspondiente y se verifica la correcta importación de las animaciones.

Finalmente, se realizan pruebas de funcionamiento dentro del editor de Unity para verificar la correcta visualización de texturas en los modelos, la reproducción de las animaciones y la interacción de los personajes, objetos y escenarios. Además, se comprueba

que los recursos mantengan un rendimiento adecuado antes de continuar con las pruebas finales de videojuego.

Ilustración 33

Exportación e Integración de los modelos 3D



Nota: Elaboración propia

4.8. Optimización gráfica

La inteligencia artificial “*Tencent Hunyuan 3D*”, se utiliza como herramienta de apoyo para agilizar el proceso de optimización. Su función permite generar una base inicial para la retopología de los modelos, reduciendo el tiempo requerido en esta etapa. Sin embargo, los modelos obtenidos son optimizados y ajustados manualmente en Blender para garantizar una topología adecuada para la animación, el rendimiento y la integración en Unity.

Una vez importado el modelo generado por inteligencia artificial “*Tencent Hunyuan 3D*”, se realiza el proceso de retopología manual en Blender, con el objetivo de reducir la cantidad de polígonos sin afectar significativamente la apariencia visual del modelo. Este procedimiento permite obtener un modelo optimizado para su utilización en videojuegos en tiempo real.

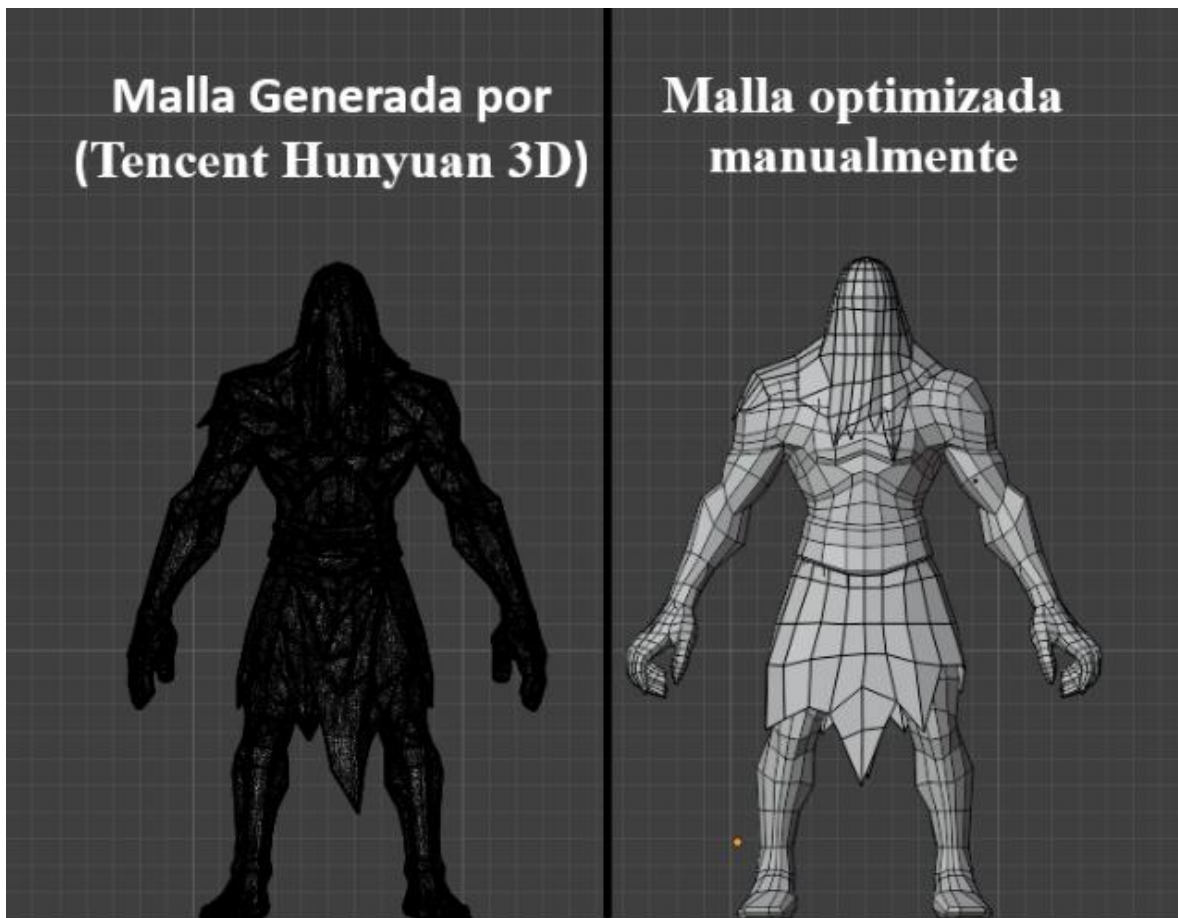
Como primer paso, el modelo original se utiliza como referencia y sirve como base para la retopología. Para mantener la malla adherida al modelo de referencia se aplica el modificador (*Shrinkwrap*), este modificador une automáticamente los vértices de la nueva malla sobre la superficie del modelo original permitiendo conservar su forma durante todo el proceso de retopología.

Finalizada la retopología, se revisa la geometría de la malla para eliminar polígonos innecesarios y mantener una malla uniforme compuesta solamente por cuadriláteros (*quads*).

Al comparar el modelo original y el modelo obtenido mediante retopología manual, se observa una reducción en la cantidad de polígonos sin afectar la apariencia visual del personaje. Además, la nueva topología mejoro la deformación durante las animaciones y facilito su integración y rendimiento dentro del motor Unity.

Ilustración 34

Optimización gráfica de modelos 3D



Nota: Elaboración propia

4.9. Software, formatos y organización de recursos

Para desarrollar los recursos visuales del videojuego “*Ashes of Faith*”, se usaron principalmente Blender, Photoshop y Unity. Blender se utiliza para crear los modelos 3D, lo que incluye el modelado, el mapeo UV, el *rigging* y la animación de personajes y objetos. Photoshop se usó para elaborar y editar las texturas, principalmente para aplicar el estilo visual plano y estilizado para los personajes y objetos del proyecto.

Finalmente, Unity se emplea para integrar los modelos 3D y sus animaciones en el entorno del videojuego. Los recursos se organizaron según su función, separando personajes,

escenarios, objetos, texturas y animaciones. Esta organización permitió mantener un flujo de trabajo ordenado y facilitó la transferencia de recursos desde Blender y Photoshop hacia su integración y validación.

4.10. Conclusiones

El desarrollo de este proyecto permitió aplicar técnicas de modelado, texturizado, *rigging* y animación en Blender para la creación de personajes, escenarios y objetos tridimensionales destinados al videojuego *Ashes of Faith*. La aplicación de estas técnicas permitió obtener recursos gráficos optimizados, logrando un equilibrio entre la calidad visual y el rendimiento necesario para su integración en Unity.

Asimismo, la investigación demostró que una adecuada planificación del proceso de desarrollo, junto con el uso de herramientas y metodologías apropiadas, contribuye a mejorar la organización del trabajo y la calidad del producto final. Los fundamentos teóricos analizados sirvieron como base para la toma de decisiones durante el desarrollo del proyecto, permitiendo cumplir con los objetivos planteados y obtener una propuesta tecnológica funcional.

4.11. Recomendaciones

Se recomienda que, en futuros proyectos de desarrollo de videojuegos, se mantenga una planificación adecuada desde las primeras etapas del modelado 3D, definiendo estándares de escala, organización de recursos y criterios de optimización. Esto permitirá facilitar la integración de los modelos en el motor gráfico y mejorar el rendimiento del videojuego.

Además, es recomendable continuar ampliando el proyecto mediante la incorporación de nuevos personajes, escenarios y animaciones, así como realizar pruebas periódicas de

rendimiento durante el proceso de desarrollo. De esta manera, será posible identificar oportunidades de mejora y garantizar una experiencia de juego más estable y de mayor calidad para los usuarios.

Bibliografía

- [1] W. Su, D. Du, X. Yang, S. Zhou, y H. Fu, «Interactive Sketch-Based Normal Map Generation with Deep Neural Networks», *Proc. ACM Comput. Graph. Interact. Tech.*, vol. 1, n.º 1, pp. 1-17, jul. 2018, doi: 10.1145/3203186.
- [2] L. Wang y J. Zhang, «A Fast and Efficiency Character Model Method for 3D Game», en *2008 International Conference on Internet Computing in Science and Engineering*, Harbin, China: IEEE, ene. 2008, pp. 118-121. doi: 10.1109/ICICSE.2008.57.
- [3] M. Tsai y T. Lu, «A Rapid Mesh Fusion Method to Create 3D Virtual Characters in Games», en *2009 Fourth International Conference on Computer Sciences and Convergence Information Technology*, Seoul, Korea: IEEE, 2009, pp. 393-398. doi: 10.1109/ICCIT.2009.276.
- [4] G. Koulaxidis y S. Xinogalos, «Improving Mobile Game Performance with Basic Optimization Techniques in Unity», *Modelling*, vol. 3, n.º 2, pp. 201-223, mar. 2022, doi: 10.3390/modelling3020014.
- [5] Z. Wang, M. Sun, G. Ren, y F. Meng, «High-Resolution Textured 3D Human Modeling from Images.», en *2010 International Conference on Multimedia Technology*, Ningbo, China: IEEE, oct. 2010, pp. 1-4. doi: 10.1109/ICMULT.2010.5631431.
- [6] Y. Ono, Bing-Yu Chen, y T. Nishita, «3D Character Model Creation from Cel Animation», en *2004 International Conference on Cyberworlds*, Tokyo, Japan: IEEE, 2004, pp. 210-215. doi: 10.1109/CW.2004.1.
- [7] I. K. Kazmi, L. You, X. Yang, X. Jin, y J. J. Zhang, «Efficient sketch-based creation of detailed character models through data-driven mesh deformations», *Comput. Animat. Virtual Worlds*, vol. 26, n.º 3-4, pp. 469-481, may 2015, doi: 10.1002/cav.1656.
- [8] B. Kenwright, «Controlled 3D Biped Stepping Animations Using the Inverted Pendulum and Impulse Constraints», en *2013 International Conference on Cyberworlds*, Yokohama, Japan: IEEE, oct. 2013, pp. 326-329. doi: 10.1109/CW.2013.20.
- [9] F. Multon, R. Kulpa, L. Hoyet, y T. Komura, «From Motion Capture to Real-Time Character Animation», en *Motion in Games*, vol. 5277, A. Egges, A. Kamphuis, y M. Overmars, Eds., en *Lecture Notes in Computer Science*, vol. 5277., Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 72-81. doi: 10.1007/978-3-540-89220-5_8.
- [10] N. Isami y Y. Sakamoto, «Interactive 3D Model Generation from Character Illustration», en *SIGGRAPH Asia 2020 Posters*, Virtual Event Republic of Korea: ACM, dic. 2020, pp. 1-2. doi: 10.1145/3415264.3425458.
- [11] Yao-Yang Tsai, Wen-Chieh Lin, K. B. Cheng, Jehee Lee, y Tong-Yee Lee, «Real-Time Physics-Based 3D Biped Character Animation Using an Inverted Pendulum Model», *IEEE Trans. Vis. Comput. Graph.*, vol. 16, n.º 2, pp. 325-337, mar. 2010, doi: 10.1109/TVCG.2009.76.
- [12] G. Arvanitis, A. S. Lalos, y K. Moustakas, «Image-Based 3D MESH Denoising Through A Block Matching 3D Convolutional Neural Network Filtering Approach», en *2020 IEEE International Conference on Multimedia and Expo (ICME)*, London, United Kingdom: IEEE, jul. 2020, pp. 1-6. doi: 10.1109/ICME46284.2020.9102938.
- [13] C. Lv, W. Lin, y B. Zhao, «Voxel Structure-Based Mesh Reconstruction From a 3D Point Cloud», *IEEE Trans. Multimed.*, vol. 24, pp. 1815-1829, 2022, doi: 10.1109/TMM.2021.3073265.
- [14] F. Ucheddu, A. Del Mastio, F. Remondino, A. Pelagotti, y V. Cappellini, «Texture mapping of flat-like 3D models», en *2009 16th International Conference on Digital Signal Processing*, Santorini, Greece: IEEE, jul. 2009, pp. 1-6. doi: 10.1109/ICDSP.2009.5201210.
- [15] D. Li, Y. Hu, J. Ji, Y. Liu, y Z. Wan, «A Limbs Combination Algorithm in Character Modeling», en *2012 International Conference on Computer Science and Service System*, Nanjing, China: IEEE, ago. 2012, pp. 1907-1910. doi: 10.1109/CSSS.2012.475.
- [16] D. Arifudin, N. A. Khasanah, y A. A. Robbani, «3D Game Asset Optimization Using Quadric Error Metrics And Decimate Modifier», *J. Sist. Komput. Dan Inform. JSON*, vol. 7, n.º 2, pp. 321-331, dic. 2025, doi: 10.30865/json.v7i2.8974.
- [17] J. Pan, X. Han, W. Chen, J. Tang, y K. Jia, «Deep Mesh Reconstruction From Single RGB Images via Topology Modification Networks», en *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South): IEEE, oct. 2019, pp. 9963-9972. doi: 10.1109/ICCV.2019.01006.
- [18] J. Wang, Y. Zhong, Y. Li, C. Zhang, y Y. Wei, «Re-Identification Supervised Texture Generation», en *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA: IEEE, jun. 2019, pp. 11838-11848. doi: 10.1109/CVPR.2019.01212.

- [19] X. Xu, C. Leng, y Z. Wu, «Rapid 3D Human Modeling and Animation Based on Sketch and Motion Database», en *2011 Workshop on Digital Media and Digital Content Management*, Hangzhou, Zhejiang, China: IEEE, may 2011, pp. 121-124. doi: 10.1109/DMDCM.2011.52.
- [20] R. Hanocka, A. Hertz, N. Fish, R. Giryes, S. Fleishman, y D. Cohen-Or, «MeshCNN: a network with an edge», *ACM Trans. Graph.*, vol. 38, n.º 4, pp. 1-12, ago. 2019, doi: 10.1145/3306346.3322959.
- [21] R. Hanocka, G. Metzger, R. Giryes, y D. Cohen-Or, «Point2Mesh: a self-prior for deformable meshes», *ACM Trans. Graph.*, vol. 39, n.º 4, ago. 2020, doi: 10.1145/3386569.3392415.
- [22] L. Qiu *et al.*, «AniGS: Animatable Gaussian Avatar from a Single Image with Inconsistent Gaussian Reconstruction», en *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA: IEEE, jun. 2025, pp. 21148-21158. doi: 10.1109/CVPR52734.2025.01970.
- [23] «Jugar a Zelda Breath of the Wild y Tears of the Kingdom en Switch 2 descubren una nueva aventura». [En línea]. Disponible en: <https://as.com/meristation/reportajes/jugar-a-zelda-breath-of-the-wild-y-tears-of-the-kingdom-en-switch-2-descubren-una-nueva-aventura-r/>
- [24] «6 Genshin Impact Tips That'll Help You Survive Teyvat». [En línea]. Disponible en: <https://uk.pcmag.com/pc-games/129390/6-genshin-impact-tips-thatll-help-you-survive-teyvati>
- [25] «DARK SOULS™ III». [En línea]. Disponible en: <https://games.gg/es/dark-souls-iii/>
- [26] S. Pant, K. Negi, y S. K. Srivastava, «3D Asset Size Reduction using Mesh Retopology and Normal Texture Mapping», en *2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, Greater Noida, India: IEEE, dic. 2021, pp. 1061-1065. doi: 10.1109/ICAC3N53548.2021.9725549.
- [27] «What is Scrum». [En línea]. Disponible en: <https://scrumguides.org/download.html>
- [28] K. Schwaber, *Agile Project Management with Scrum*. Microsoft Press.
- [29] «Corriendo hacia el éxito en Scrum: Una guía rápida». [En línea]. Disponible en: <https://guides.visual-paradigm.com/es/sprinting-towards-success-in-scrum-a-quick-guide/>
- [30] D. J. Anderson, *Kanban: Successful Evolutionary Change for Your Technology Business*. Blue Hole Press.
- [31] M. O. Ahmad, J. Markkula, y M. Oivo, «Kanban in software development: A systematic literature review», en *2013 39th Euromicro Conference on Software Engineering and Advanced Applications*, Santander: IEEE, sep. 2013, pp. 9-16. doi: 10.1109/SEAA.2013.28.
- [32] «Kanban in Project Management for Beginners: Steps + Examples». [En línea]. Disponible en: <https://plaky.com/learn/project-management/kanban-project-management/>
- [33] B. Beck y C. Andres, *Extreme Programming Explained: Embrace Change*. Addison-Wesley Professional.
- [34] K. Beck, «Embracing change with extreme programming», *Computer*, vol. 32, n.º 10, pp. 70-77, oct. 1999, doi: 10.1109/2.796139.
- [35] «Extreme Programming (XP)». [En línea]. Disponible en: <https://www.geeksforgeeks.org/software-engineering/software-engineering-extreme-programming-xp/>
- [36] «Manual de referencia de Blender 5.2 LTS». [En línea]. Disponible en: <https://docs.blender.org/manual/en/latest/>
- [37] «Autodesk Maya: Crea personajes complejos, mundos extensos y efectos deslumbrantes.» [En línea]. Disponible en: <https://www.autodesk.com/products/maya/overview>
- [38] «3ds Max: Crea mundos inmersivos y diseños de alta calidad.» [En línea]. Disponible en: <https://www.autodesk.com/products/3ds-max/overview>
- [39] «Photoshop Help». [En línea]. Disponible en: <https://helpx.adobe.com/photoshop/desktop.html>
- [40] «Welcome to the Krita 5.3 Manual». [En línea]. Disponible en: <https://docs.krita.org/en/>
- [41] «Substance 3D Painter Help». [En línea]. Disponible en: <https://helpx.adobe.com/substance-3d-painter/>
- [42] E. Van Horn, *3D Character Development Workshop: Rigging Fundamentals for Artists and Animators*. De Gruyter, 2024. doi: 10.1515/9781683922445.
- [43] J. Ma y D. Zhang, «TARig: Adaptive template-aware neural rigging for humanoid characters», *Comput. Graph.*, vol. 114, pp. 158-167, ago. 2023, doi: 10.1016/j.cag.2023.05.018.
- [44] N. Villanueva, *Beginning 3D Game Assets Development Pipeline: Learn to Integrate from Maya to Unity*. Berkeley, CA: Apress, 2022. doi: 10.1007/978-1-4842-7196-4.
- [45] H.-Y. Peng, J.-P. Zhang, M.-H. Guo, Y.-P. Cao, y S.-M. Hu, «CharacterGen: Efficient 3D Character Generation from Single Images with Multi-View Pose Canonicalization», *ACM Trans. Graph.*, vol. 43, n.º 4, pp. 1-13, jul. 2024, doi: 10.1145/3658217.

- [46] G. Bharaj, T. Thormählen, H. Seidel, y C. Theobalt, «Automatically Rigging Multi-component Characters», *Comput. Graph. Forum*, vol. 31, n.º 2pt4, pp. 755-764, may 2012, doi: 10.1111/j.1467-8659.2012.03034.x.
- [47] J. Pan, X. Yang, X. Xie, P. Willis, y J. J. Zhang, «Automatic rigging for animation characters with 3D silhouette», *Comput. Animat. Virtual Worlds*, vol. 20, n.º 2-3, pp. 121-131, jun. 2009, doi: 10.1002/cav.284.

Anexos

Pruebas y resultados

La validación de los recursos visuales se lleva a cabo mediante diferentes pruebas durante las etapas de desarrollo e integración. Para realizar estas pruebas se utilizaron principalmente Blender y Unity, herramientas que permitieron comprobar tanto las características de los modelos 3D como su comportamiento dentro del entorno interactivo.

En Blender se verificó que los modelos mantengan una estructura adecuada de geometría, materiales y texturas. También se realizaron pruebas de *rigging* y animación para comprobar que los personajes no presentaran deformaciones inadecuadas durante sus movimientos.

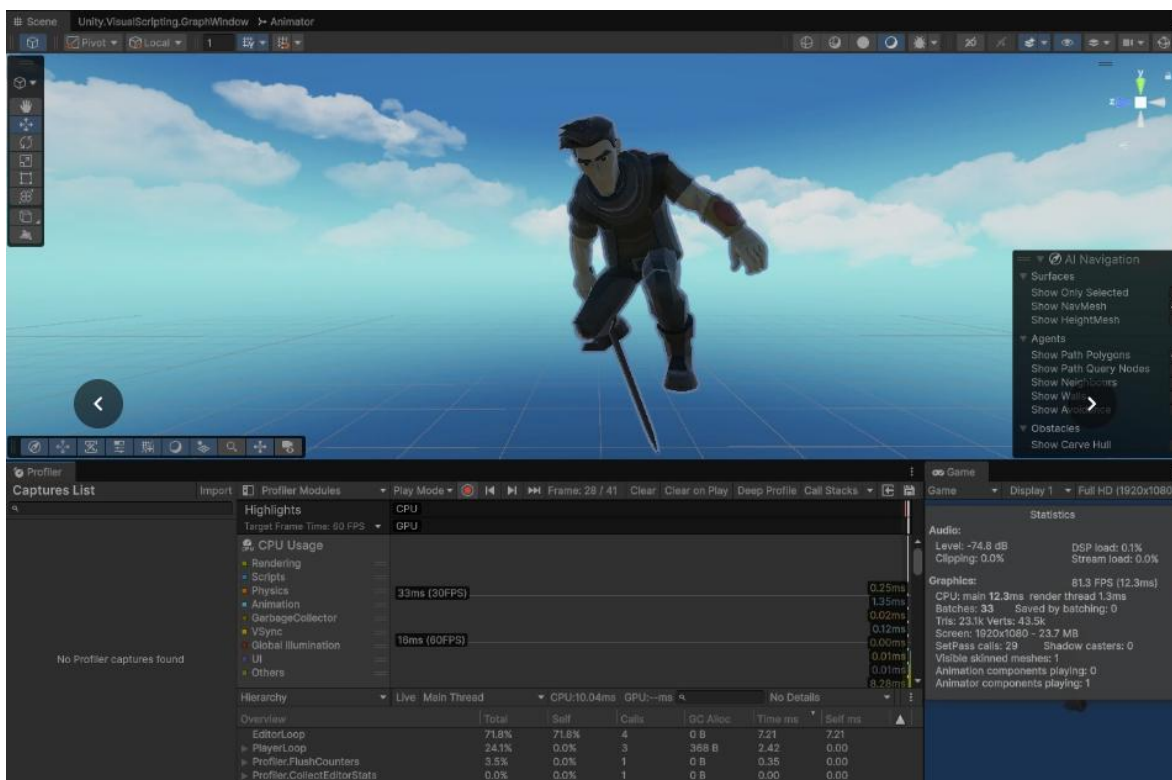
Además, se revisó el funcionamiento de los recursos integrados en Unity, considerando aspectos como la visualización de los modelos, la reproducción de las animaciones y el rendimiento de los modelos dentro del entorno interactivo. Estas pruebas permitieron identificar errores de integración, deformaciones y problemas de optimización.

Como resultado, se realizaron ajustes en los modelos y animaciones para mejorar su funcionamiento y asegurar que los recursos visuales cumplieran con las características definidas para el proyecto. La validación permitió comprobar la calidad técnica y la correcta integración de los elementos desarrollados antes de su uso en el videojuego.

Para evaluar el rendimiento de los modelos dentro de Unity, se utilizaron las estadísticas de rendimiento disponibles en el motor, mediante las cuales se registraron valores relacionados con los fotogramas por segundo (FPS), *batches* y cantidad de triángulos visibles. Estos datos permitieron analizar el comportamiento de los recursos durante la ejecución de las escenas.

Ilustración 35

Análisis del rendimiento gráfico de los modelos 3D en Unity

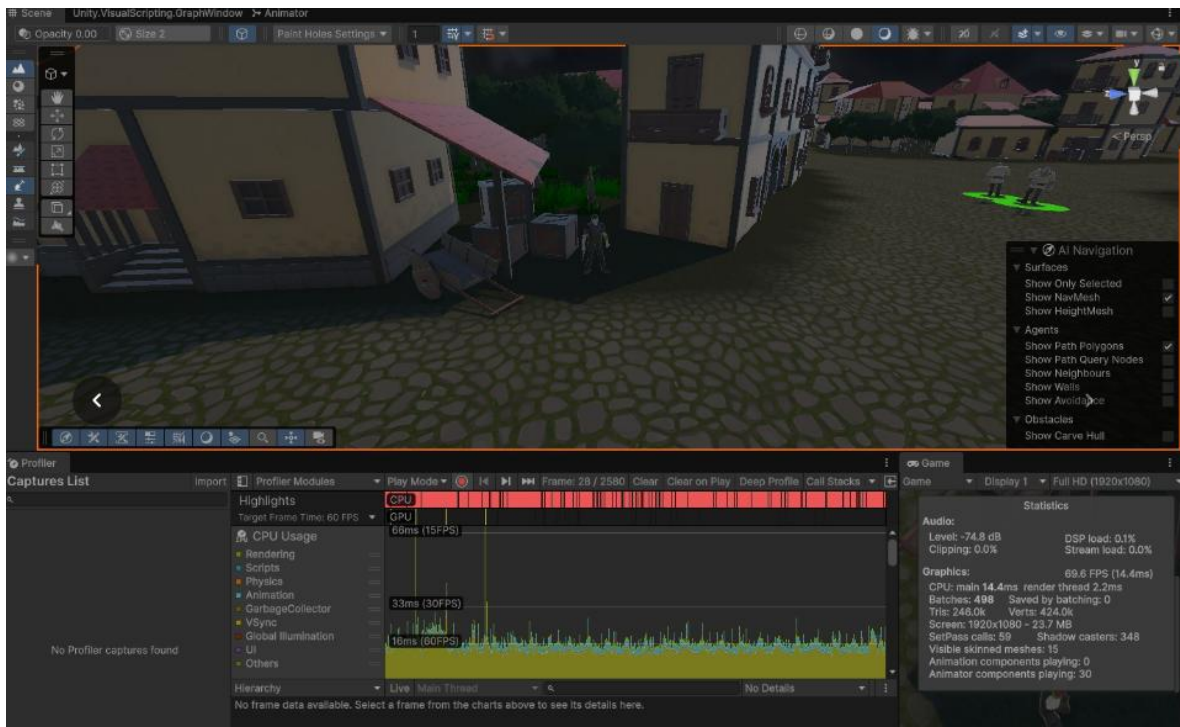


Nota: Elaboración propia

Los resultados obtenidos en Unity demuestran que los modelos desarrollados presentan un rendimiento óptimo para su implementación en el videojuego. La escena alcanza 81.3 FPS, superando el mínimo recomendado de 60 FPS, mientras que el número de *batches* (33) y la cantidad de triángulos visibles (23.1K) se mantienen muy por debajo del límite recomendado para videojuegos en tiempo real. Esto confirma que la optimización realizada permite obtener un buen desempeño sin afectar la calidad visual de los modelos.

Ilustración 36

Prueba de rendimiento de la escena completa del videojuego en Unity



Nota: Elaboración propia

La prueba de rendimiento realizada en Unity demuestra que la escena completa del videojuego mantiene un desempeño optimo, alcanzando 69.6 FPS, valor superior al mínimo recomendado de 60 FPS. Además, la cantidad de triángulos visibles (246.0k) se encuentra muy por debajo del límite de un millón de triángulos recomendado para videojuegos en tiempo real, lo que confirma que los modelos y escenarios optimizados pueden integrarse correctamente sin comprometer el rendimiento del proyecto.