



**UNIVERSIDAD
NACIONAL DE LOJA**



Facultad de la Energía, las Industrias y los Recursos Naturales No Renovables

CARRERA DE INGENIERÍA EN SISTEMAS

“Análisis Comparativo para la Evaluación de Motores de Juegos en el Desarrollo de Juegos Serios”

TRABAJO DE TITULACIÓN PREVIO A
LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN SISTEMAS

Autor:

Jennifer del Carmen Tomalá González

Director:

Ing. José Oswaldo Guamán Quinche, Mg. Sc

Loja – Ecuador

2019

Certificación del Director

Ing. José Oswaldo Guamán Quinche Mg. Sc.

DOCENTE DE LA CARRERA DE INGENIERIA EN SISTEMAS DE LA FACULTAD DE ENERGIA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES DE LA UNIVERSIDAD NACIONAL DE LOJA.

CERTIFICA:

Que la egresada Jennifer del Carmen Tomalá González, realizó el Trabajo de Titulación denominado: **“ANÁLISIS COMPARATIVO PARA LA EVALUACIÓN DE MOTORES DE JUEGOS EN EL DESARROLLO DE JUEGOS SERIOS”** bajo mi dirección y asesoramiento, mismo que fue revisado, enmendado y corregido minuciosamente. En virtud que el Trabajo de Titulación reúne, a satisfacción las cualidades de fondo y forma exigidas para un trabajo de este nivel, autorizó su presentación, sustentación y defensa ante el tribunal respectivo.

Loja, 08 de agosto del 2019

Ing. José Oswaldo Guamán Quinche Mg. Sc.

DIRECTOR DE TESIS

Autoría

Yo, **TOMALÁ GONZÁLEZ JENNIFER DEL CARMEN** declaro ser autora del presente Trabajo de Titulación y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales, por el contenido de la misma.

Adicionalmente acepto y autorizo a la Universidad Nacional de Loja, la publicación de mi trabajo de titulación en el Repositorio Institucional-Biblioteca Virtual.

Firma:

Cédula: 0603951815

Fecha: Loja, 05 de septiembre del 2019

Carta de autorización del Trabajo de Titulación por parte de la Autora, para la consulta, reproducción parcial o total y publicación electrónica del texto completo.

Yo, **JENNIFER DEL CARMEN TOMALÁ GONZÁLEZ**, declaro ser la autora del Trabajo de Titulación titulado: **ANÁLISIS COMPARATIVO PARA LA EVALUACIÓN DE MOTORES DE JUEGOS EN EL DESARROLLO DE JUEGOS SERIOS**, como requisito para optar el grado de **INGENIERO EN SISTEMAS**; autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional:

Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior, con las cuales tenga convenio la universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia de la tesis que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja a los 08 días del mes de agosto del dos mil diecinueve.

Firma:

Autor: Jennifer del Carmen Tomalá González

Cédula: 0603951815

Dirección: Loja, (Ciudad Alegría, Av. Eloy Alfaro y Medardo Alfaro, L21).

Correo Electrónico: jctomalaq@unl.edu.ec

Teléfono: 072112291 **Celular:** 0986773563

DATOS COMPLEMENTARIOS

Director de Tesis: Ing. José Oswaldo Guamán Quinche Mg. Sc.

Tribunal de Grado: Ing. Jorge Iván Tocto, Mg. Sc.

Ing. Wilman Patricio Chamba Zaragoza, Mg. Sc.

Ing. Javier Francisco Sinche Freire, Mg. Sc.

Agradecimiento

Dejo constancia de mi sincero agradecimiento a la Autoridades de la Universidad Nacional de Loja, a la Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables, en particular a los docentes de la Carrera de Ingeniería en Sistemas ya que con sus valiosos conocimientos impartidos han participado en mi formación como profesional.

De manera especial, mi gratitud más sincera al Mgs. José Oswaldo Guamán Quinche, Director de Tesis, quién con su experiencia profesional, confianza, acertadas sugerencias y recomendaciones, guío el presente trabajo investigativo en forma adecuada.

A mis familiares, amigos y demás personas que creyeron en mí, de verdad GRACIAS...

Jennifer Tomalá G.

Dedicatoria

El presente Trabajo de Titulación está dedicado principalmente a Dios quien guía mis pasos día tras día, a mis padres Gilbert y Lucila que con su amor, paciencia, sacrificio y sabios consejos me supieron motivar para seguir adelante y alcanzar esta anhelada meta, a mis hermanos Mercedes, Henry, Paul, Nicol y Cristian que siempre me han brindado su apoyo incondicional, y sobre todo a ese ser tan especial que ilumina mis días con solo existir mi hijo Joel Isaías quien es y será mi mayor motivación para nunca rendirme, para levantarme con fortaleza tras cada caída, y así poder llegar a ser un ejemplo para él; a mis sobrinos, amigos y demás personas que confiaron y creyeron en mí.

Jennifer Tomalá G.

Índice General

| | |
|-----------------------------------------------------------------------------|------|
| Certificación del Director..... | II |
| Autoría..... | III |
| Carta de autorización del Trabajo de Titulación por parte de la Autora..... | IV |
| Agradecimiento..... | V |
| Dedicatoria..... | VI |
| A. Título..... | XII |
| B. Resumen..... | XIII |
| Summary..... | XIV |
| C. Introducción..... | 1 |
| D. Revisión de literatura..... | 3 |
| 1. JUEGOS SERIOS..... | 3 |
| 1.1. ¿Qué son los Juegos Serios?..... | 3 |
| 1.2. Campos de Aplicación..... | 3 |
| 1.2.1. Militar..... | 3 |
| 1.2.2. Política..... | 4 |
| 1.2.3. Empresariales..... | 5 |
| 1.2.4. Salud..... | 6 |
| 1.2.5. Educación..... | 6 |
| 2. TECNOLOGÍA DE DESARROLLO..... | 7 |
| 2.1. Motores de Juegos..... | 7 |
| 2.2. Componentes de un motor de juegos..... | 8 |
| 2.2.1. Motor de Renderizado..... | 8 |
| 2.2.2. Motor de Audio..... | 10 |
| 2.2.3. Motor de Física y Detección de Colisiones..... | 11 |
| 2.2.4. Inteligencia Artificial..... | 12 |
| 2.2.5. Networking y multijugador..... | 13 |
| 2.2.6. Scripting..... | 13 |
| 2.3. UNITY..... | 14 |
| 2.3.1. Características..... | 15 |
| 2.3.2. Requisitos de instalación..... | 16 |
| 2.4. UNREAL ENGINE 4..... | 16 |
| 2.4.1. Características..... | 17 |
| 2.4.2. Requisitos de instalación..... | 18 |
| E. Materiales y métodos..... | 19 |
| 1. Materiales..... | 19 |
| 1.1. Talento Humano..... | 19 |
| 1.2. Hardware y Software..... | 19 |
| 1.3. Servicios..... | 20 |
| 2. Técnicas de recolección de información..... | 21 |
| 3. Métodos..... | 21 |
| 4. Metodología de desarrollo..... | 21 |

| | |
|-----------------------------------------------------------------------------------------------------------------------------------|-----|
| F. Resultados | 23 |
| 1. FASE 1: Analizar los motores de juegos y establecer los criterios de comparación..... | 23 |
| 2. FASE 2: Desarrollar un prototipo de videojuego que sirva como escenario real para la evaluación. | 25 |
| 3. FASE 3: Evaluar los motores de juegos de acuerdo a los criterios de comparación establecidos. | 49 |
| G. Discusión | 61 |
| H. Conclusiones..... | 65 |
| I. Recomendaciones | 67 |
| J. Bibliografía | 69 |
| K. Anexos | 72 |
| ANEXO 1: Revisión Sistemática de literatura (SLR) sobre Metodologías y Motores de Juegos para el desarrollo de Juegos serios..... | 72 |
| ANEXO 2: Game Design Document..... | 116 |
| ANEXO 3: Guía de Instalación de Unreal Game Engine | 128 |
| ANEXO 4: Certificado de Traducción del Resumen | 137 |

Índice de figuras

| | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Figura 1. Armed Assault 2..... | 4 |
| Figura 2. Food Force..... | 5 |
| Figura 3. Advergaming..... | 5 |
| Figura 4. Utilización de videojuegos para rehabilitación física de pacientes..... | 6 |
| Figura 5. Utilización de videojuegos para la práctica de médicos..... | 7 |
| Figura 6. Componentes de un Motor de juego..... | 8 |
| Figura 7. Capas del motor de renderización..... | 9 |
| Figura 8. Modelo Render de una mesa..... | 10 |
| Figura 9. Interfaz de usuario de Sound Cue Editor de Unreal Engine..... | 10 |
| Figura 10. Aplicación de un Capsule Collider a un brazo de un robot..... | 12 |
| Figura 11. Interacción de personajes por sí solos dentro de un laboratorio de ciencias dentro de la escuela virtual (Second Life, Linden Research Inc.)..... | 12 |
| Figura 12. World of Warcraft. Ejemplo de Juego multiplayer..... | 13 |
| Figura 13. Sistema Visual de Scripting – Blueprint de Unreal Engine..... | 14 |
| Figura 14. Logo de Unity..... | 14 |
| Figura 15. Logo de Unreal Engine 4..... | 16 |
| Figura 16. Motores de juegos utilizados en desarrollo de Juegos Serios..... | 24 |
| Figura 17. Criterios para la elección de un Motor de Juegos..... | 24 |
| Figura 18. Opciones de licencia disponibles en Unity..... | 26 |
| Figura 19. Descarga e instalación de Unity Hub Launcher..... | 26 |
| Figura 20. Crear cuenta en Unity..... | 27 |
| Figura 21. Plantillas prediseñadas para principiantes..... | 27 |
| Figura 22. Arranque del motor de juegos Unreal Engine 4..... | 28 |
| Figura 23. Formulario para la creación de cuenta en Epic Games..... | 28 |
| Figura 24. Descarga y ejecución del instalador..... | 29 |
| Figura 25. Inicio de sesión en Epic Games..... | 29 |
| Figura 26. Instalación del motor de juegos..... | 30 |
| Figura 27. Arranque del motor de juegos Unreal Engine 4..... | 30 |
| Figura 28. Diagrama de Estados del Juego..... | 32 |
| Figura 29. Boceto del menú principal..... | 33 |
| Figura 30. Boceto de la pantalla del juego..... | 33 |
| Figura 31. Boceto de la pantalla de Pause..... | 34 |
| Figura 32. Boceto de la pantalla del juego terminado..... | 34 |
| Figura 33. Pantalla de creación de proyecto..... | 39 |
| Figura 34. Pantalla de creación de nivel..... | 39 |
| Figura 35. Acción de correr sin parar..... | 40 |
| Figura 36. Determinación del grado de rotación para poder girar..... | 41 |
| Figura 37. Acción de girar..... | 41 |
| Figura 38. Acción de morir..... | 42 |
| Figura 39. Sistema de creación de plataforma..... | 43 |
| Figura 40. Sistema de creación de obstáculos..... | 43 |
| Figura 41. Sistema de destrucción de plataformas..... | 44 |
| Figura 42. Sistema de generación de objetos..... | 44 |

| | |
|-------------------------------------------------------------------------------|----|
| Figura 43. Recolección y Destrucción Objetos. | 44 |
| Figura 44. Generación Componentes GUI. | 45 |
| Figura 45. Pantalla de creación de proyecto. | 46 |
| Figura 46. Pantalla de creación de Scene. | 46 |
| Figura 47. Script configuración del movimiento de la cámara. | 47 |
| Figura 48. Script de configuración del movimiento del personaje. | 48 |
| Figura 49. Script de destrucción de objetos. | 48 |
| Figura 50. Script de efectos del objeto. | 49 |
| Figura 51. Resultados de la Evaluación Experimental. | 53 |
| Figura 52. Resultados del Rendimiento del GPU al ejecutar Unity. | 54 |
| Figura 53. Resultados del Rendimiento del GPU al ejecutar Unreal Engine. | 54 |
| Figura 54. Resultados de la Evaluación durante la construcción. | 58 |

Índice de tablas

| | |
|---------------------------------------------------------------------------------|----|
| Tabla I. Características del motor de juego Unity..... | 15 |
| Tabla II. Características del motor de juego Unreal Engine 4..... | 17 |
| Tabla III. Talento humano..... | 19 |
| Tabla IV. Hardware y Software..... | 20 |
| Tabla V. Servicios | 20 |
| Tabla VI. Determinación de características a comparar | 25 |
| Tabla VII. Determinación de valores a evaluar..... | 36 |
| Tabla VIII. determinación de valores para la evaluación experimental | 37 |
| Tabla IX. Comparativa de características básicas..... | 50 |
| Tabla X. Comparación de componentes básicos | 51 |
| Tabla XI. Evaluación experimental | 52 |
| Tabla XII. Determinación de escala de medición y pesos | 56 |
| Tabla XIII. Evaluación de los motores de juegos en la fase de construcción..... | 57 |
| Tabla XIV. Determinación de fortalezas y debilidades de Unity..... | 58 |
| Tabla XV. Determinación de fortalezas y debilidades de Unreal Engine 4 | 59 |
| Tabla XVI. Valoración económica de recursos humano | 63 |
| Tabla XVII. Valoración económica de materiales y servicios..... | 63 |
| Tabla XVIII. Valoración económica de recursos técnicos y tecnológicos | 64 |
| Tabla XIX. Costo total del Trabajo de Titulación..... | 64 |

A.Título

**“Análisis Comparativo para la
Evaluación de Motores de Juegos
en el Desarrollo de Juegos
Serios.”**

B. Resumen

El presente trabajo de titulación tiene como propósito realizar un análisis comparativo para la evaluación de Motores de Juegos en el desarrollo de Juegos Serios, permitiendo así, determinar fortalezas y debilidades de dos significativas herramientas empleadas en la creación de juegos serios, con el fin de permitirle al desarrollador realizar una correcta toma de decisiones al seleccionar la herramienta acorde a sus necesidades.

La investigación inicia con la elaboración de una Revisión Sistemática de Literatura (SLR) referente a Metodologías y Motores de juegos para la implementación de juegos serios, logrando establecer que, de la gran variedad de motores de juegos existentes en la actualidad, Unity y Unreal Engine son los más utilizados en este tipo de desarrollo, debido a que ofrecen grandes ventajas como coste, curva de aprendizaje, aceleración en el flujo de trabajo, entre otros. Partiendo de ello, se determina y categoriza criterios propuestos por varios autores a la hora de elegir este tipo de herramienta, mismos que están enmarcados en características básicas y componentes del motor.

Es así que, se procede a diseñar un prototipo de juego serio para luego crearlo en cada uno de los motores seleccionados, lo que permite analizar factores tales como: funcionalidad, usabilidad, documentación, tiempos de respuesta, y rendimiento del Pc al usar cada uno de los motores. Finalmente se realiza el análisis comparativo tanto de las características, componentes y la evaluación experimental asociada a la producción del juego serio en los dos motores de juegos utilizados en el presente estudio.

Palabras claves: Desarrollo de juegos serios, motores de juegos, Unity Engine, Unreal Engine.

Summary

The present degree work aims to carry out a comparative analysis for the evaluation of Games Engines in the Development of Serious Games, allowing to determine strengths and weaknesses of two significant tools employed in the creation of serious games, in order to allow the developer to make a correct decision making when selecting the tool according to their needs.

The research begins with the elaboration of a Systematic Literature Review (SLR) concerning Methodologies and Game Engines for the implementation of serious games, succeeding in confirming that, of the wide variety of game engines currently available, Unity and Unreal Engine are the most used in this type of development, because they offer great advantages as cost, learning curve, work flow acceleration, among others.

On the basis of this, criteria proposed by several authors is determined and categorized when choosing this type of tool, which are framed in basic characteristics and components of the engine.

So, we proceed to design a serious game prototype and then create it in each of the selected engines, which allows to analyze factors such as: functionality, usability, documentation, response times, and performance of the PC when using each of the engines. Finally, a comparative analysis of the characteristics, components and the experimental evaluation associated with the production of the serious game in the two game engines used in this study is carried out.

Keywords: Development of serious games, game engines, Unity Engine, Unreal Engine.

C.Introducción

No hay duda que los videojuegos se han convertido en uno de los productos de ocio y entretenimiento de mayor crecimiento en los últimos años, Joan Chipia (2014) nos manifiesta que el papel que desempeñan los videojuegos en la actualidad es de proveer habilidades y destrezas facilitando el aprendizaje de procesos complejos con eficacia.[1] A los Videojuegos de carácter educativo, se los denomina Juegos Serios, donde el principal objetivo no es el entretenimiento ni la competitividad, sino más bien el de aprender o reforzar conocimientos, adoptar nuevas habilidades facilitando el aprendizaje en diferentes áreas en los que se aplique como pueden ser: la educación, seguridad, defensa, política, o salud.[2]

El proceso de desarrollo de Juegos Serios o Videojuegos Educativos se acerca al de un producto de software común, necesitando la utilización de metodologías y herramientas que garanticen la calidad de los mismos, la reducción del tiempo de elaboración, y la optimización de tareas; su producción puede iniciar de 0, utilizando librerías o mediante plataformas de desarrollo que para el caso particular de los Videojuegos se denominan Motores de juegos o Games Engines (GE). [3] Si bien el desarrollo de Videojuegos es un tema interesante, son pocas las personas que asumen este reto, debido a varias dificultades que este proceso conlleva, ya que primero se debe considerar para qué plataforma van a desarrollar el videojuego, la dificultad al seleccionar la mejor herramienta de desarrollo debido a la amplia gama de motores de juegos disponibles en el mercado, la necesidad de tener un gran grupo de desarrolladores (ingenieros de sonido, animación, modelado, etc.), sus altos costos de producción, entre otras.

En base a los inconvenientes mencionados anteriormente, se ha considerado pertinente realizar un análisis comparativo para la evaluación de motores de juegos en el desarrollo de juegos serios; la misma que permita conocer características, componentes, fortalezas y debilidades de los motores de juegos y promover el uso de los mismos en el desarrollo de juegos serios, dando solución de esta manera al problema de investigación: **¿El determinar fortalezas y debilidades de los motores de juegos, mediante un análisis comparativo, facilitará al desarrollador la elección de la herramienta más adecuada para el desarrollo de Juegos Serios?**

Entre las actividades desarrolladas, se realizó una Revisión Sistemática de Literatura sobre metodologías y motores de juegos para el desarrollo de juegos serios con el objetivo de determinar cuáles son los motores de juegos más utilizados en el desarrollo de juegos serios en la actualidad, con el cual se determinó que Unity y Unreal Engine 4 son los que cuentan con mayor popularidad por las diferentes ventajas que estos presentan. A la par de este proceso se realizó un estudio de los criterios que diversos autores establecen para la elección de este tipo de herramientas, los mismos que serán categorizados para evaluar los motores de juegos antes mencionados. Posteriormente se desarrolló un demo o prototipo de aplicación que sirvió como escenario para la evaluación de los motores de juegos en base a los criterios establecidos; finalmente se realizó el análisis comparativo y se determinó fortalezas y debilidades de cada uno de los motores evaluados.

El presente trabajo de titulación está conformado por varias sesiones y anexos que permiten la comprensión del análisis realizado, las mismas que son: En la sección de **Revisión de Literatura**, se especifica la información teórica correspondiente a Juegos Serios y Motores de juegos puntualizando sobre Unity 5 y Unreal Engine 4; en **Técnicas y Métodos** se expone las técnicas de recolección de información utilizadas, además se explica el método deductivo, inductivo y el método de revisión sistemática de literatura; en **Resultados** se hace referencia a las fases seguidas para realizar este proyecto; en **Discusión** se expone la propuesta alternativa describiendo los pasos seguidos para dar cumplimiento de los objetivos planteados.

Finalmente, se redactan tanto **Conclusiones y Recomendaciones** obtenidas al finalizar el presente trabajo; como sustento científico de la investigación realizada se presenta la respectiva **Bibliografía** utilizada y se adjunta los **Anexos** como evidencia formal del trabajo realizado.

D.Revisión de literatura

A continuación, se brinda una base conceptual sobre los juegos serios, campos de aplicación, y de manera general sobre motores de Juegos y sus componentes haciendo hincapié en la descripción de cada uno de los motores de juegos seleccionados como objetos de estudio en este trabajo.

1. JUEGOS SERIOS

1.1. ¿Qué son los Juegos Serios?

El término Juegos Serios o Serious Games fue acuñado en 1970 por Clark Albert, investigador estadounidense y autor del libro Serious Games (Viking Press, 1970) en el cual los define como juegos diseñado y desarrollados primordialmente con un propósito educativo por encima del entretenimiento, a lo que Sellami (2014) acota, que deben poseer características de videojuegos para generar motivación y experiencias en el aprendizaje inmersivo.[4]

Los Juegos Serios rompen con el pensamiento generalizado de que los videojuegos sirven únicamente para el ocio. Las instituciones pueden encontrar en este medio al vehículo idóneo para capacitar a sus empleados, promocionar sus productos, simular modelos de negocios, recolectar datos, comunicar y educar sobre las enfermedades, concientizar sobre problemas sociales, ecológicos, humanitarios, y mucho más.[5]

1.2. Campos de Aplicación

1.2.1. Militar

El área Militar ha sido precursora en el uso de juegos y simuladores para el entrenamiento en habilidades tácticas y de toma de decisiones (Wayne, 2003) así como en el entrenamiento en el manejo de armas y vehículos de guerra a través de simuladores y de videojuegos que luego han sido lanzados al mercado y han tenido una buena aceptación como herramienta para la captación y entrenamiento de los soldados, adoptando mayor velocidad de reacción, mejor capacidad para diferenciar el equipo

amigo del equipo enemigo, eficiente manejo de las reglas del juego, mayor facilidad de comunicación en los juegos multijugador.[6] Los ambientes tridimensionales virtuales, representan una gran ventaja por la efectividad que se logra y la seguridad del contexto, evitando fallos y reduciendo posibles errores que tendrían consecuencias lamentables, como pérdidas humanas y el mal manejo de equipos costosos.[7]



Figura 1. Armed Assault 2 [8]

1.2.2. Política

Estos son juegos diseñados para educar al público sobre sus derechos y obligaciones como ciudadanos, para fomentar los comportamientos cívicos y las normas de convivencia. Las ONG los utilizan para denunciar injusticias o para promover la conciencia social.[7]

El Programa Mundial de Alimentos de Naciones Unidas desarrolló en 2005 un videojuego para promover la concienciación social y la lucha contra el hambre. En este videojuego (Food Force) los jugadores se ponen en la piel de un recién llegado a la ONU que tiene que lidiar con una crisis de hambruna y guerra civil en una isla del Pacífico. El objetivo del jugador es distribuir comida entre la población y ayudar al país a ser autosuficiente en el medio-largo plazo.[9]



Figura 2. Food Force[10]

1.2.3. Empresariales

La mayoría de los Juegos Serios que se utilizan en el sector empresarial son comúnmente llamados Advergaming, son herramientas de mercadotecnia y comunicación que permiten la exposición o promoción de una marca o producto específico ante el posible cliente, usualmente lo utilizan las marcas de automóviles.



Figura 3. Advergaming[11]

Dentro del sector empresarial también se destacan los simuladores con sus diversos grados de complejidad para fomentar la cultura emprendedora, potenciar habilidades estratégicas, prevenir riesgos laborales o aprender mecanismos de trabajo de una manera eficaz y más amena. [2]

1.2.4. Salud

Otro de los sectores con gran acogida que tienen los Juegos Serios es el campo de la Salud, debido a que son muy utilizados sobre todo en pacientes con discapacidades intelectuales como física. Raúl Oliveira (2017) nos manifiesta que los juegos serios son utilizados para la evaluación objetiva de la rehabilitación física cuando se hace uso de los sensores Kinect para la detección de movimientos del cual se extraen datos como velocidades de ángulos, tiempo y distancia asociados con las mecánicas del juego. [12] Martins, Carvalho y Soares concuerdan que los programas de fisioterapia o rehabilitación física son repetitivos y aburridos y que se necesita el diseño e implementación de ambientes agradables como motivación para los pacientes procurando la continuidad sus tratamientos.[13]



Figura 4. Utilización de videojuegos para rehabilitación física de pacientes[14]

1.2.5. Educación

En el campo educativo encontramos numerosos ejemplos donde son utilizados los videojuegos en el proceso de enseñanza-aprendizaje. Marcano Beatriz (2008) menciona que los Serios Games se los ha utilizado para el entrenamiento de los estudiantes de medicina, enfermería y personal paramédico mediante las simulaciones

de diferentes procedimientos médicos. [3] Un caso que se puede señalar es el que menciona Prabal, Katz et al. (2013) que trata de un Serious Game diseñado para la capacitación sobre el método aséptico para el catéter venoso central o procedimiento de colocación de línea central, el mismo que fue probado con un grupo de residentes del Centro Médico Mount Sinai.[7]

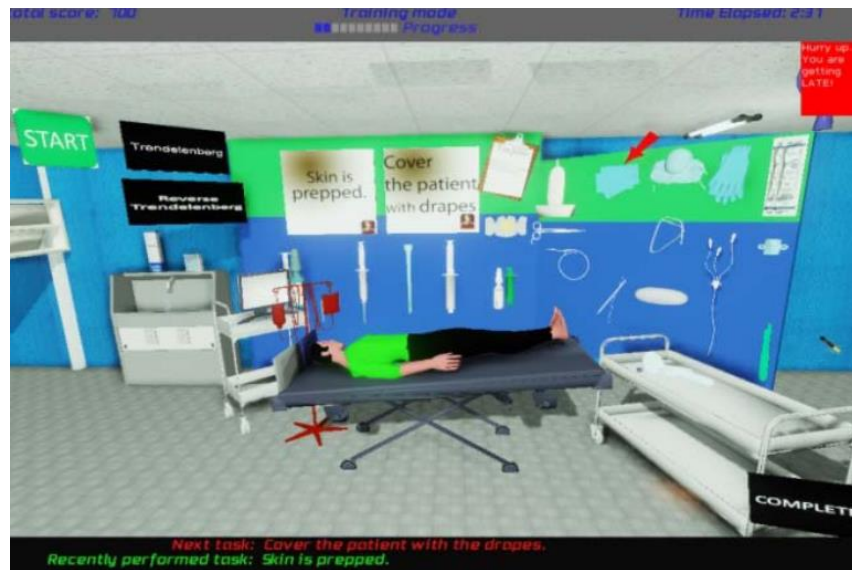


Figura 5. Utilización de videojuegos para la práctica de médicos[15]

2. TECNOLOGÍA DE DESARROLLO

2.1. Motores de Juegos

Un motor de juegos es una aplicación informática compuesta por un conjunto de sistemas de software extensibles como son: motor de gráficos o renderizado, motor de sonido, motor de Inteligencia Artificial, motor de física, que facilita el desarrollo de un videojuego sin la necesidad de una gran modificación de código. [16]



Figura 6. Componentes de un Motor de juego[17]

2.2. Componentes de un motor de juegos

2.2.1. Motor de Renderizado

El motor de render es el encargado de las gráficas, uno de los más grandes y complejos componentes en cualquier motor de juegos, el cual depende del hardware de gráficos y las librerías que se utilizan para interactuar con él, es quien permite crear mallas estáticas, pintado, animación, efectos de post-procesado, partículas, iluminación, texturas, materiales, entre otras, a partir de gráficos o modelos de 2D y 3D.[18]

El motor de renderización es el encargado de hacer que todos los objetos del videojuego sean renderizados en la pantalla de la plataforma, (Patton, 2015) define unas responsabilidades que tiene el motor, las cuales mejoran el rendimiento del videojuego; dichas responsabilidades son:

- Transformar, procesar y presentar la información del escenario en un formato digital o en un dispositivo de salida de imagen.
- Ocultar las partes de un objeto que no son visibles desde la perspectiva del observador, evitando que estas sean enviadas al proceso de renderización y por ende optimizando el rendimiento de dicho proceso.
- Controlar el LOD (Level of Detail) sobre los objetos, reduciendo la complejidad de la presentación de un objeto 3D a medida que este se aleja del observador o

de acuerdo a otras métricas como son la importancia sobre otros objetos, su velocidad o su posición relativa al observador.

El motor de renderización está compuesto por 4 subcapas básicamente, las mismas que se detallan en la figura a continuación empezando de manera ascendente:

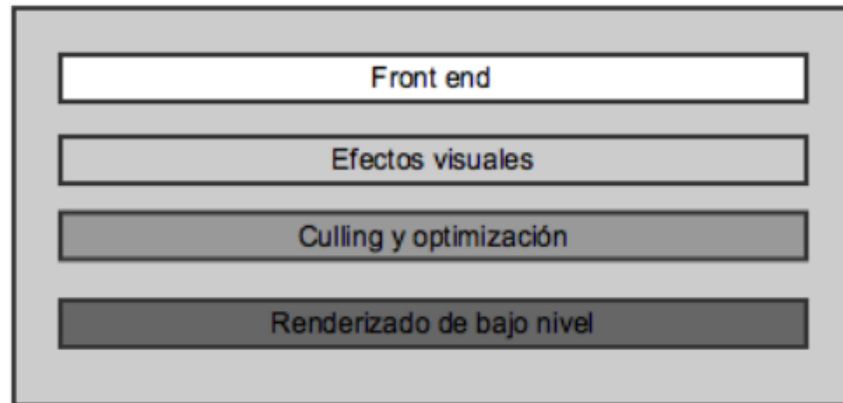


Figura 7. Capas del motor de renderización.[18]

- **Renderizado de bajo nivel:** el renderizado de bajo nivel, pretende renderizar las distintas primitivas (texturas y materiales) de nuestro mundo 3D de la manera más rápida posible sin tener en cuenta posibles optimizaciones. Simplemente todos los vértices que sean visibles por la cámara serán renderizados, aquí.
- **Culling y Optimización:** Como la palabra lo dice optimiza la cantidad de primitivas que envía el rendering de bajo nivel, esto se logra al descartar los objetos no visibles o fuera del viewport, haciendo uso del control de sistemas LOD (Level of Details).[18]
- **Efectos visuales:** Son aquellos que soportan distintos efectos visuales que posterior pueden ser integrados en los juegos desarrollados, un ejemplo de estos son los sistemas de partículas (humo, agua, etc.), los mapeados del entorno o las sombras dinámicas.
- **Front-end:** suele estar vinculada a funcionalidad relativa a la superposición de contenido 2D sobre el escenario 3D. Por ejemplo, es bastante común utilizar algún tipo de módulo que permita visualizar el menú de un juego o la interfaz gráfica que permite conocer el estado del personaje principal del videojuego (inventario, armas, herramientas, etc.).

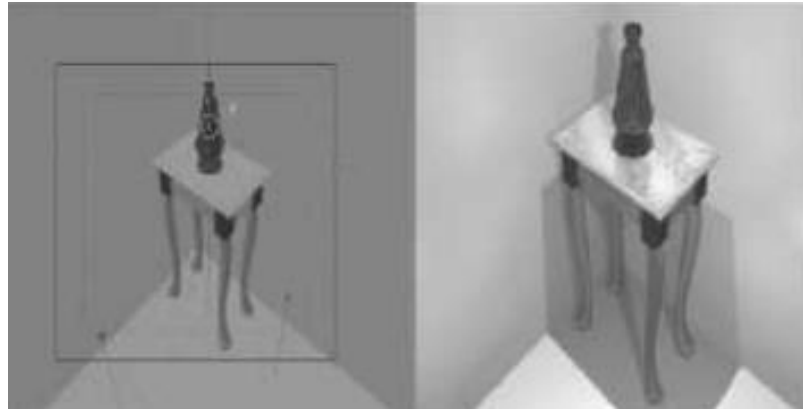


Figura 8. Modelo Render de una mesa[19]

2.2.2. Motor de Audio

El audio hace que la experiencia del jugador sea mucho más realista, permitiendo que según la situación se cree un ambiente de inmersión intenso para este, por lo cual el motor de audio ha ido cobrando más y más relevancia. Así mismo, la aparición de nuevos formatos de audio de alta definición y la popularidad de los sistemas de cine en casa han contribuido a esta evolución en el cada vez más relevante apartado sonoro. Actualmente, al igual que ocurre con otros componentes de la arquitectura del motor de juego, es bastante común encontrar desarrollos listos para utilizarse e integrarse en el motor de juego, los cuales han sido realizados por compañías externas a la del propio motor.

El motor de audio o sonidos tiene que poseer funcionalidades mínimas como: Carga, descompresión y reproducción archivos digitales, los motores más actuales incluyen efectos Doopler, ecos, ajustes de amplitud y tono, entre otros.[20]

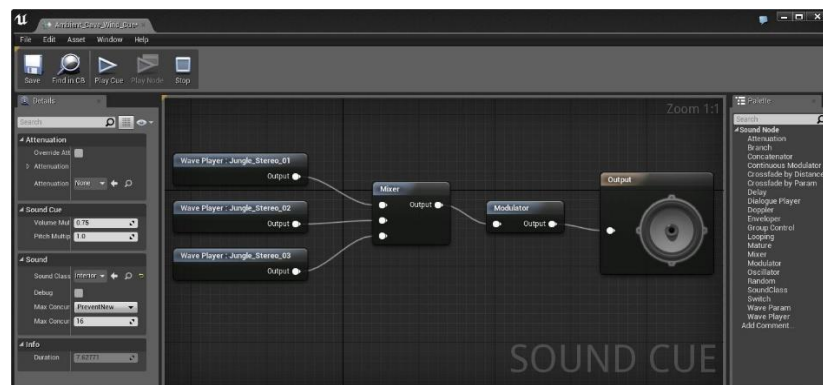


Figura 9. Interfaz de usuario de Sound Cue Editor de Unreal Engine[21]

2.2.3. Motor de Física y Detección de Colisiones

Este motor es el que permite aplicar aproximaciones físicas a los videojuegos para que luzcan más naturales y sean más ergonómicos para el jugador. En otras palabras, es el encargado de realizar los cálculos necesarios para que un objeto simule tener atributos físicos como peso, volumen, estado físico, gravedad etc.[20]

Al sistema de Física va ligado con el sistema de detección de colisiones que no es más que el mecanismo que detecta cuando dos o más objetos chocan simulando estímulos y brindando al programador la opción de reaccionar a las colisiones como lo desee, dotando de realismo al objeto. Sin un mecanismo de detección de colisiones, los objetos se traspasarían unos a otros y no sería posible interactuar con ellos. Desde un punto de vista general, el sistema de detección de colisiones es responsable de llevar a cabo las siguientes tareas:

- **La detección de colisiones**, cuya salida es un valor lógico indicando si hay o no colisión.
- **La determinación de la colisión**, cuya tarea consiste en calcular el punto de intersección de la colisión.
- **La respuesta a la colisión**, que tiene como objetivo determinar las acciones que se generarán como consecuencia de la misma.

Actualmente, la mayoría de compañías utilizan motores de colisión/física desarrollados por terceras partes, integrando estos kits de desarrollo en el propio motor. Los más conocidos en el ámbito comercial son Havok, el cual representa el estándar por defecto en la industria debido a su potencia y rendimiento, y PhysX, desarrollado por NVIDIA e integrado en motores como por ejemplo Unreal Engine 3. En el ámbito del open source, uno de los más utilizados es ODE. [22]

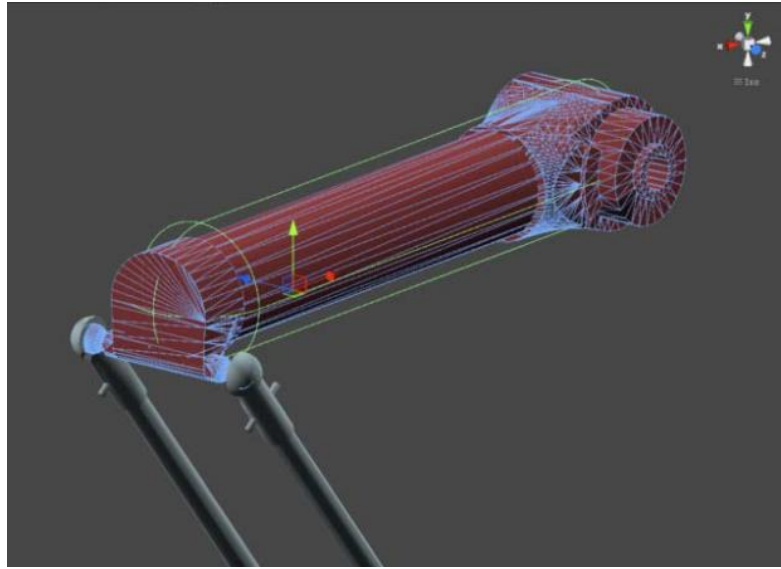


Figura 10. Aplicación de un Capsule Collider a un brazo de un robot.[23]

2.2.4. Inteligencia Artificial

El sistema de Inteligencia Artificial es el encargado de dotar a ciertos elementos del juego un comportamiento pseudointeligente. Por regla general, lo que se aplica en los juegos son técnicas muy simples de inteligencia artificial como máquinas de estados o algoritmos de búsquedas. Sin embargo, cada vez más se utilizan nuevas técnicas para conseguir una inteligencia más “realista” como redes neuronales o algoritmos genéticos.



Figura 11. Interacción de personajes por sí solos dentro de un laboratorio de ciencias dentro de la escuela virtual (Second Life, Linden Research Inc.).[23]

2.2.5. Networking y multijugador

La gestión de redes está alcanzando una progresiva mayor importancia. Hoy por hoy, casi todos los juegos tienen una componente de red. Existen juegos exclusivamente online, de los cuales, la mayoría ofrece partidas multijugador a través de red con el objetivo de incrementar la jugabilidad y duración en el mercado[18]. El módulo de red de un motor de videojuego es el encargado de mantener en constante comunicación a todos los jugadores que se encuentran conectados dentro de una misma red, tal como lo expresa (Ramez, 2015) “El módulo de red es responsable de administrar los estados de todos los actores del videojuego en una red. Otra responsabilidad es el mantener el estado de juego consistente en ambas plataformas y evitar interferencias en la unidad de movimiento de red.”. Este módulo se usa principalmente en aquellos videojuegos con multijugador, donde se debe procurar que todos los jugadores interactúen entre sí sin que se presenten interferencias en tiempo de ejecución, así como mantener información actualizada sobre todos los demás jugadores presentes. [22]



Figura 12. World of Warcraft. Ejemplo de Juego multiplayer[24]

2.2.6. Scripting

Este mecanismo posibilita al programador la creación de scripts propios cuando las funcionalidades buscadas no están implementadas en el motor o cuando los

comportamientos de algunas configuraciones requieren algunos ajustes. La mayoría de motores hoy en día, permiten realizar scripts en los lenguajes más comunes (C#, C++, Python, entre otros.). Actualmente varios motores de juegos han integrado sistemas visuales de scripting haciendo más amigable el desarrollo de videojuegos a usuarios que tienen poco conocimiento en programación, un ejemplo de este tipo de sistemas es el que implementa Unreal Engine 4 denominado Blueprint.

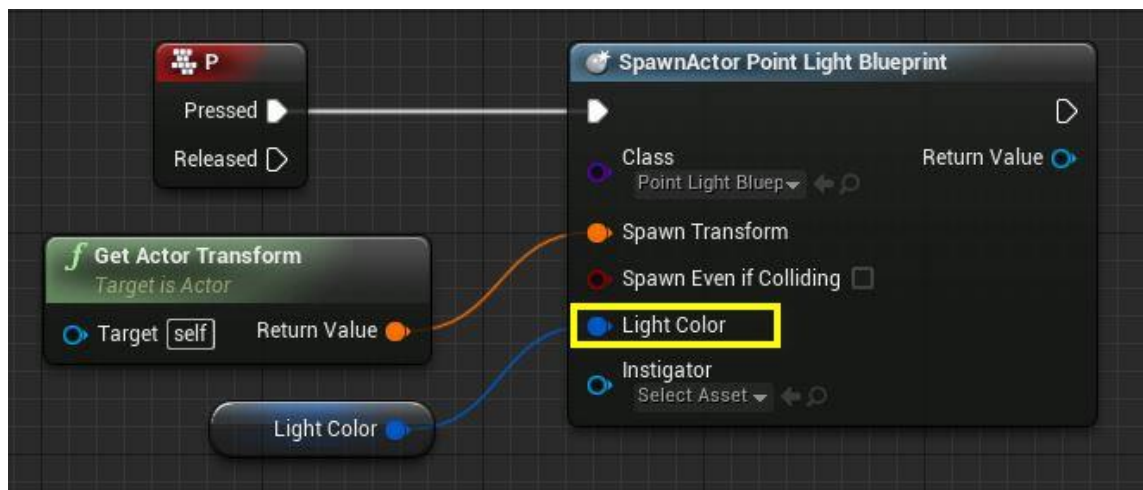


Figura 13. Sistema Visual de Scripting – Blueprint de Unreal Engine.[21]

2.3. UNITY

Unity es un motor de videojuegos desarrollado por Unity Technologies en el 2005 para su juego GooBall, el cual no tuvo éxito alguno, pero sus desarrolladores vieron la oportunidad de ofrecer a las compañías desarrolladoras un motor competente a un precio asequible. Unity fue lanzado oficialmente en el 2006 como una plataforma de desarrollo flexible y poderosa para la creación de juegos en 2D y 3D, su gran éxito se debe a que ofreció a los estudios de desarrollo indie un motor competente con un precio por licencia mucho más bajo en comparación con otros motores usados por las grandes compañías.



Figura 14. Logo de Unity

2.3.1. Características

Unity, cuenta con las siguientes características que se muestran en la tabla siguiente:

[2][25]

TABLA I. CARACTERÍSTICAS DEL MOTOR DE JUEGO UNITY.

| CATEGORÍA: | CARACTERÍSTICAS |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Licencia | <ul style="list-style-type: none"> • Personal: Es gratis, y ofrece las características mínimas con varias limitaciones. • Profesional: Es de pago mensual (Desde USD\$75), y ofrece todas las características disponibles de la versión más reciente. |
| Plataformas que Soporta | <ul style="list-style-type: none"> • Móvil: Android OS, iOS, Windows Phone 8, BlackBerry 10 y Tizen. • Escritorio: Windows, Windows Store Apps, Mac, Linux, y Steam OS. • Consolas de Videojuegos: PS3, PS4, PSVITA, Xbox 360, Xbox One, Nintendo Switch, Nintendo Wii. • Web: WebPlayer (Firefox, Chrome, IE, Safari, etc) y WebGL. • Realidad Virtual: PlayStation VR, Oculus Rift, Gear VR, HTC Vive, Samsung Gear VR, Steam VR, Google Cardboard y Google Daydream y Microsoft Hololens. • Otros: Android TV, Samsung SmartTV, tvOS y Facebook Gameroom). |
| Renderización | <ul style="list-style-type: none"> • Animaciones Retargetables. • Jerarquías y transacciones sofisticadas sobre máquinas de estados. • Iluminación global en tiempo real. • Sombreado basado en física. • Sistema modular de partículas, curvo e impulsado por gradientes. • Herramientas intuitivas para el usuario. • Utilización de Skybox para la creación de entornos lejanos. |
| Física | <ul style="list-style-type: none"> • NVidia PhysX 3.3. • Box2D con una amplia gama de efectores, las articulaciones y colisionadores. |
| Soporte de aprendizaje | <ul style="list-style-type: none"> • Documentación. • Video tutoriales. • Gran Comunidad Activa. |
| Audio y Video | <ul style="list-style-type: none"> • Mezclas y masterización de sonidos en tiempo real. • Jerarquías en las mezclas, copias instantáneas y efectos predeterminados. • Tipo de audio que soporta (.mp3, .ogg, .aiff, .wav, .mod, .it, .sm3). |
| Redes y Multijugador | <ul style="list-style-type: none"> • Networking o High level API para un manejo multijugador básico. • Network Transport API para infraestructuras de red y multijugador. |
| Scripting | <ul style="list-style-type: none"> • Se puede realizar en lenguajes como C#, JavaScript y Boo. • IA con un avanzado algoritmo de PathFinding automatizado y mallas de navegación. |
| Otros | <ul style="list-style-type: none"> • Tienda de Assets donde se pueden conseguir elementos de diseño necesarios para adicionar a cualquier proyecto que lo requiera. Se ofrecen assets gratis y de pago. • Integración total sobre Perforce y Plastic SCM, para el control de versiones. |

2.3.2. Requisitos de instalación

Requisitos para ser utilizado en el desarrollo:

- Sistema Operativo:
 - Windows 64-bit versions
 - macOS 10.12+
- CPU: Soporte para el conjunto de instrucciones SSE2.
- GPU: Tarjeta de video con capacidad para DX10 (shaders modelo 4.0).

2.4. UNREAL ENGINE 4

Unreal Engine es un paquete (Suite) de herramientas para el desarrollo de Videojuegos, desarrollado por Epic Games, Unreal Engine es un potente motor gráfico y motor de juego que permite el desarrollo de juegos de todo tipo desde plataforma 2D hasta VR y para prácticamente cualquier plataforma (PC, VR, consolas, smartphones, etc.).

La facilidad de uso y el hecho de que no es necesario pagar ningún tipo de licencia durante el desarrollo de un producto lo hacen un motor ideal tanto para estudios independientes como para estudios más grandes y consolidados. Ofrece desarrollo móvil, un lenguaje gráfico de basado en scripts sin la necesidad de escribir líneas de código de lenguaje de programación denominado Blueprint, varias herramientas de edición de aspectos visuales, acceso al código fuente en lenguaje de programación C++ y un amplio mercado para adquirir diferentes aspectos del videojuego creados por la comunidad. Unreal Engine maneja un solo flujo de trabajo, lo cual le permite un flujo único de trabajo orientado a diferentes plataformas como: iOS Android, Html5, Linux, OSX, PlayStation, Steam, Windows y Xbox.



Figura 15. Logo de Unreal Engine 4[21]

2.4.1. Características

Las principales características de este motor de videojuegos se muestran en la tabla siguiente: [2][21][25]

TABLA II. CARACTERÍSTICAS DEL MOTOR DE JUEGO UNREAL ENGINE 4.

| CATEGORÍA: | CARACTERÍSTICAS |
|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Licencia | <ul style="list-style-type: none"> • Gratuita hasta obtener 3000 dólares trimestrales en beneficios, luego de eso se pagará el 5% de compensaciones a la empresa Epic Games. |
| Plataformas que soporta | <ul style="list-style-type: none"> • Soporte multiplataforma. (Windows, Mac, Linux, PS4, Xbox One, Nintendo Switch, SteamOS, iOS, Android y HTML5). • Soporte para realidad virtual. (PlayStation VR, Oculus Rift, Samsung Gear VR, HTC Vive y Google Daydream). |
| Renderización | <ul style="list-style-type: none"> • Compatibilidad total con funcionalidades de renderizado y animación con DirectX11 y DirectX12, como reflejos HDR de escenas completas, miles de luces dinámicas, desplazamiento y mosaicos programables, sombreados, perfiles de luz IES y mucho más. • Cascade VFX. Editor de partículas completo y detallado optimizado para un bajo consumo de GPU. • Terreno y Follaje: Se puede crear ambientes grandes y extensos de exploración con el sistema de panoramas. El cual permite pintar terrenos a grandes magnitudes gracias al poderoso sistema LOD (Level of Detail) y el uso eficiente de memoria. • Efectos Visuales en cascada: Ofrece las herramientas para crear fuego, humo, nieve, polvo, escombros de forma detallada. • Segmentación (Pipeline) de Materiales: Esta característica hace uso de un sombreado físico, el cual ofrece un control sobre la apariencia de personajes u objetos, esto realizado mediante el editor de material. • Escalabilidad: Ofrece la posibilidad de adaptar el tamaño de los gráficos dependiendo de las necesidades del hardware y rendimiento de donde se encuentre el videojuego. • Animación de Personas: Se utiliza la herramienta de animación para modificar estructuras y mallas esqueléticas. Igualmente se puede modificar las propiedades de colisión de las mallas esqueléticas utilizando herramientas de edición Physics Assets (PhAT). • Efecto Post-procesamiento: La característica de post procesamiento permite ajustar la apariencia de las escenas como, por ejemplo, tecnología cubemaps para el ambiente (comúnmente utilizado en Skyboxes), el sombreado de ambiente, degradado de colores, profundidad de campo, entre otras. |
| Física | <ul style="list-style-type: none"> • SDK Physx 3.3 de Nvidia para la emulación de la física en 3D. • Paper 2D para la creación de escenas 2D. |
| Soporte de aprendizaje | <ul style="list-style-type: none"> • Cuenta con un gran soporte al aprendizaje. Documentación, una extensa bibliografía, video tutoriales, clases en directo vía Twitch y wiki. • Gran comunidad de desarrolladores activa y colaborativa. |

| | |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Audio y Video | <ul style="list-style-type: none"> • Audio: Unreal Engine ofrece un editor de audio (Sound Cue Editor) para construir la segmentación del audio, así como la definición del audio que se va a utilizar en el videojuego. |
| IA | <ul style="list-style-type: none"> • Árboles de Comportamiento: Es una herramienta poderosa que combina dos tipos de assets, el Blackboard almacena las decisiones preconfiguradas que utiliza el árbol de comportamiento y esta toma decisiones a partir de las situaciones basándose en las decisiones almacenadas. |
| Scripting | <ul style="list-style-type: none"> • Soporte para programación en lenguaje C++. • Blueprint. Lenguaje de scripting visual. |
| Otros | <ul style="list-style-type: none"> • Tienda de Assets: Unreal Engine ofrece una tienda la cual ofrece diferentes assets que se puede utilizar en la creación del Videojuego. • Contenido de Navegación: Unreal Engine permite importar, organizar, buscar, etiquetar, filtrar y modificar cantidades de componentes para el videojuego. • Acceso al código fuente, control sobre el código, editor de código fuente y función de recarga en caliente: Unreal Engine permite el acceso al código fuente de cada objeto y las características que este ofrece, igualmente permite total control sobre el código, un editor de código de C++ integrado y permite la actualización en caliente a cualquier modificación realizada en el código fuente. |

2.4.2. Requisitos de instalación

Unreal Engine tiene algunos requisitos específicos de hardware y software para ejecutar el Editor, los mismo que son los siguientes:[21]

- **Sistema operativo:** Windows 10 de 64 bits
- **Procesador:** Intel o AMD de cuatro núcleos, 2.5 GHz o más rápido
- **Memoria:** 8 GB de RAM
- **Tarjeta de video / versión DirectX:** Tarjeta gráfica compatible con DirectX 11 o DirectX 12

E. Materiales y métodos

Para el desarrollo del presente proyecto de titulación se hizo uso de varios recursos y métodos de investigación que permitieron llevar un proceso ordenado, conocer a fondo la problemática y adquirir los conocimientos necesarios para plantear una solución al problema de estudio. Para el logro de los objetivos planteados se utilizó lo siguiente:

1. Materiales

En la realización del presente proyecto se requirió recursos humanos, materiales técnicos y tecnológicos, que se detalla a continuación:

1.1. Talento Humano

El trabajo de titulación cuenta con una estructura de desarrollo, que involucró a un tesista y a dos docentes, los cuales guiaron de forma técnica y teórica cuyo costo es asumido por la Universidad Nacional de Loja. El tiempo empleado para el desarrollo del trabajo de titulación de 6 meses, de los cuales se emplearon un total de 400 horas. En la TABLA III muestra el talento humano que participo en esta investigación.

TABLA III. TALENTO HUMANO

| Recurso Humano | Descripción |
|-------------------------------------|--------------------|
| Jennifer del Carmen Tomalá González | Investigadora |
| José Oswaldo Guamán Quinche | Director de Tesis |
| Oscar Miguel Cumbicus Pineda | Tutor Académico |

1.2. Hardware y Software

En la TABLA IV muestra el hardware y software que se utilizaron para el desarrollo de este trabajo de titulación.

TABLA IV. HARDWARE Y SOFTWARE

| HARDWARE | | |
|--------------------------------------------------|--|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Equipo | | Descripción |
| Computador portátil Hp Core i5 modelo 14-bs015la | | Permitió realizar consultas bibliográficas, documentar todo el proceso de desarrollo del trabajo de titulación, la evaluación de las funcionalidades de los Motores de Juegos elegidos y el desarrollo estructural del documento final. |
| Impresora Epson L3150 | | Se la utilizó para imprimir documentos previos al documento final y el documento final. |
| Memoria de 8GB | | Empleado para guardar información relacionada al trabajo de titulación. |
| SOFTWARE | | |
| Mendeley Desktop | | Permitió la gestión bibliográfica para citar o referenciar desde el procesador de texto. |
| Unity3D | | Motor de videojuegos a Evaluar. |
| Unreal Engine 4 | | Motor de Videojuegos a Evaluar. |
| Procesador de texto | | Para la edición y corrección de texto de los documentos productos del trabajo de titulación. |

1.3. Servicios

En la TABLA V muestra los diferentes servicios básicos que fueron fundamentales para el desarrollo del trabajo de titulación.

TABLA V. SERVICIOS

| Servicios | | Descripción |
|------------------|--|-----------------------------------------------------------------------------------------------------------------------|
| Internet | | Se empleó para realizar consultas bibliográficas relacionadas al trabajo de titulación. |
| Transporte | | Se utilizó para transportarse a las revisiones de los avances del trabajo de titulación y a las tutorías de la misma. |
| Resma de papel | | Se las utilizó para las impresiones de los borradores y del informe final de la investigación. |
| Anillados | | Para presentar el informe final del trabajo de titulación. |

2. Técnicas de recolección de información

2.1 Bibliográfica

Esta técnica se la aplicó para buscar información en diferentes fuentes bibliográficas confiables, libros, revistas indexadas, artículos científicos, base de datos científicas, entre otras.

3. Métodos

3.1 Método Deductivo

Con el método deductivo se determinó el tema a investigar partiendo del referente teórico general sobre las herramientas de desarrollo de videojuegos hasta llegar a un punto en particular como los motores de juegos en el desarrollo de juegos serios.

3.2 Método Inductivo

Mediante este método, que es el razonamiento que, partiendo de casos particulares, se eleva a conocimientos generales; se lo utilizó para determinar los criterios necesarios para comparar los Motores de Juegos que actualmente emplean los desarrolladores de videojuegos

3.3 Método Científico

Con el método inductivo se pudo analizar, estructurar, sintetizar y desarrollar el estado del arte que da fundamento teórico al trabajo de titulación.

4. Metodología de desarrollo

En todo proceso investigativo es necesario seguir una secuencia de fases que conforman una metodología con el fin de ejecutar ordenadamente los procesos, razón por la cual se detalla las fases realizadas dentro del presente proyecto de investigación que nos ayudó a dar cumplimiento al objetivo general del mismo. Esta metodología consta de tres fases que son:

4.1 Análisis y Selección

Para iniciar el desarrollo del presente proyecto fue necesario conocer los conceptos que nos ayudaron en la investigación, para el cual se utilizó el método de revisión sistemática de literatura de acuerdo a lo propuesto en el artículo de Bárbara Kitchenham[26]. Esta revisión consta de tres etapas que son:

- **Planificación de Revisión Sistemática de Literatura:** esta fase se definió el objetivo de la Revisión Sistemática, se formuló la(s) pregunta(s) de investigación, se determinó las palabras claves, se realizó la selección de fuentes y estrategias de búsqueda, la cadena de búsqueda, los criterios de inclusión y los criterios de exclusión.
- **La etapa de Ejecución de Revisión Sistemática de Literatura:** En esta fase se realizó la búsqueda de los estudios, la selección de estudios y la extracción de la información.
- **La etapa de Análisis de resultados y hallazgos:** En esta fase se desarrolla la discusión de la revisión sistemática de literatura.

Al término de la revisión se obtiene como resultados los criterios de comparación a utilizar y la determinación de los motores de juegos más utilizados en el desarrollo de juegos serios, que fueron los objetos principales de estudio dentro del proyecto (Ver sección Anexo 1).

4.2 Diseño y Desarrollo

Durante esta fase se instalaron los motores de juegos seleccionados y registraron los inconvenientes presentados, luego se procede al diseño de un prototipo de videojuego mediante un GDD (ver sección Anexo 2), el mismo que posteriormente fue desarrollado en cada uno de los motores de juegos seleccionados como escenario real para la evaluación.

4.3 Comparación y Evaluación.

Finalmente, en esta etapa se analizó los resultados obtenidos tanto en la etapa de recolección de información referente a los criterios comparativos de cada motor de juegos, los cuales fueron constatados durante el proceso de desarrollo del prototipo, con el fin de determinar las fortalezas y debilidades de cada motor de juego evaluado.

F. Resultados

En este apartado se muestra las actividades que se llevaron a cabo para desarrollar el presente trabajo de titulación.

En el primer punto se determinan los motores de juegos a evaluar y los criterios de elección de los mismos, en segundo punto se crea un prototipo de videojuego que sirva como escenario de evaluación y en el tercer punto se realizan los cuadros comparativos sobre los dos motores de juegos, pudiendo determinar fortalezas y debilidades de los mismos.

1. FASE 1: Analizar los motores de juegos y establecer los criterios de comparación.

Para el desarrollo de esta fase se realizó un análisis sobre “Juegos serios: una revisión de metodologías y motores de juegos para su desarrollo”, en base a la revisión Sistemática de Literatura de acuerdo a lo propuesto en el artículo de Bárbara Kitchenham[26], con el fin de identificar cuáles son los dos motores de juegos más utilizados en el desarrollo de juegos serios, cuáles son los criterios de elección de los mismos y cuáles son las metodologías que están siendo aplicadas en este tipo de desarrollo (Ver sección Anexo 1). Del cual se obtuvo los siguientes resultados:

Los motores de juegos más utilizado en la creación de juegos serios son: Unity de la empresa Unity Technologies y Unreal Engine de Epic Games, debido a características tales como: Renderización, Modelos 3D gratuitos, recursos limitados de desarrollo como tiempo y presupuesto, facilidad de manejo, soporte, documentación y accesibilidad, entre otros.

En la Figura 16 que a continuación se presenta se puede evidenciar que, dentro de los 27 estudios analizados, 21 de ellos usó, analizó o consideró a Unity para la realización de sus proyectos y 7 de ellos hizo a Unreal Engine parte de sus proyectos.

MOTORES DE JUEGOS UTILIZADOS EN EL DESARROLLO DE JUEGOS SERIOS

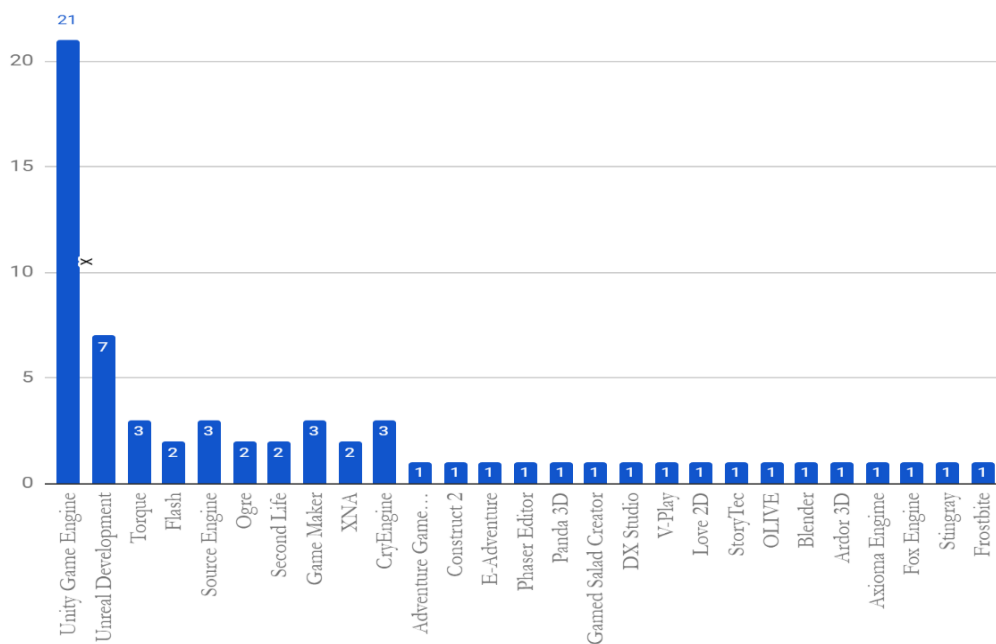


Figura 16. Motores de juegos utilizados en desarrollo de Juegos Serios

En la Figura 17 se muestran los criterios que varios autores proponen al momento de elegir el motor de juegos a utilizar, el que más consideran es que sean Multiplataforma para que los productos se adapten a las nuevas tecnologías existentes, otra característica importante es la renderización para tener un software de calidad en cuanto a imágenes.

CRITERIOS DE ELECCIÓN

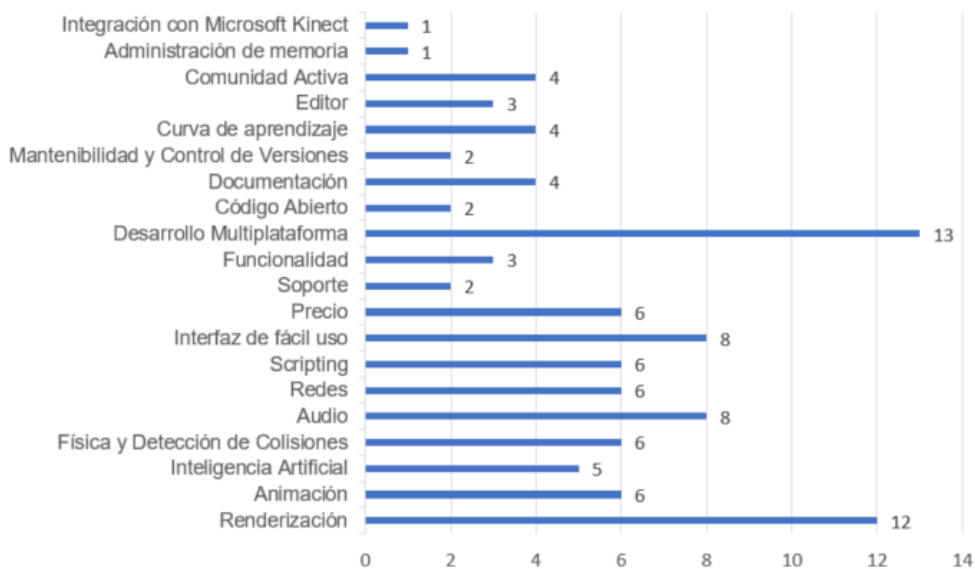


Figura 17. Criterios para la elección de un Motor de Juegos.

Los criterios expuestos en la gráfica han sido reagrupados en dos categorías, las mismas que se pueden evidenciar en la Tabla VI:

TABLA VI. DETERMINACION DE CARACTERISTICAS A COMPARAR

| COMPONENTES BÁSICOS | CARACTERÍSTICAS BÁSICAS |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none">• Renderización• Animación• Física• Audio• Inteligencia Artificial• Scripting | <ul style="list-style-type: none">• Plataformas de distribución• Documentación Disponible• Lenguaje de Programación• Tipo de Licencia• Curva de aprendizaje• Requisitos de Instalación |

2. FASE 2: Desarrollar un prototipo de videojuego que sirva como escenario real para la evaluación.

Esta fase consiste en el desarrollo de una aplicación que sirve como escenario real para la valoración de algunas características que ofrecen los motores de juegos, para lo cual se procede a realizar las siguientes actividades:

2.1 Instalar los motores de juegos seleccionados para la comparación.

2.1.1 Instalación de Unity 3D

Para instalar Unity 3D se debe ingresar a la página oficial del mismo (<https://unity.com/>) y seleccionar la opción Empezar y realizar lo siguiente:

- Seleccionar la licencia que se ajuste a nuestras necesidades, para el desarrollo de este proyecto utilizaremos la versión personal.

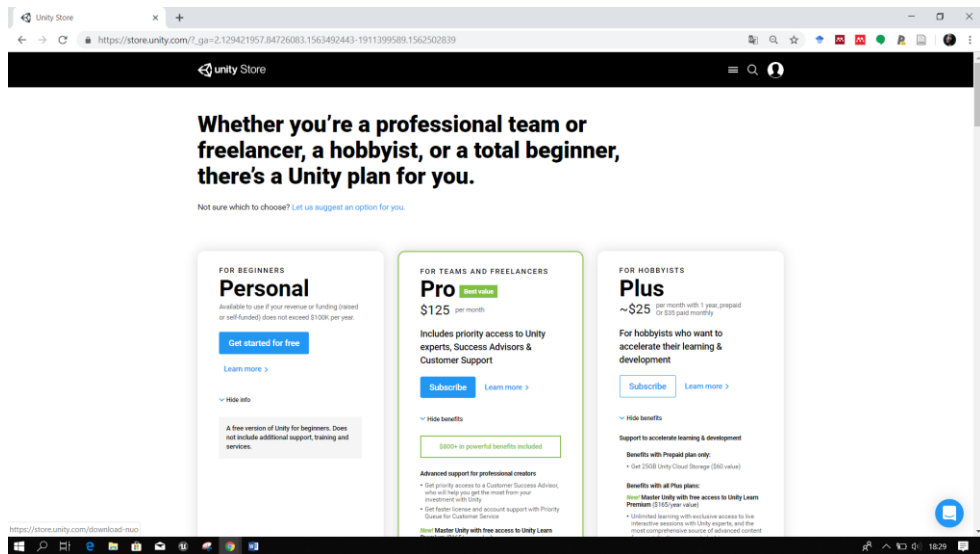


Figura 18. Opciones de licencia disponibles en Unity.

- Aceptar los términos de Licencia, y seleccionar Descargar Unity Hub, existen dos versiones la primera para principiantes y la segunda para desarrolladores con conocimientos en programación para el siguiente proyecto se optó por la primera opción, al finalizar este proceso le damos en ejecutar para que empiece a instalarse el Launcher de Unity

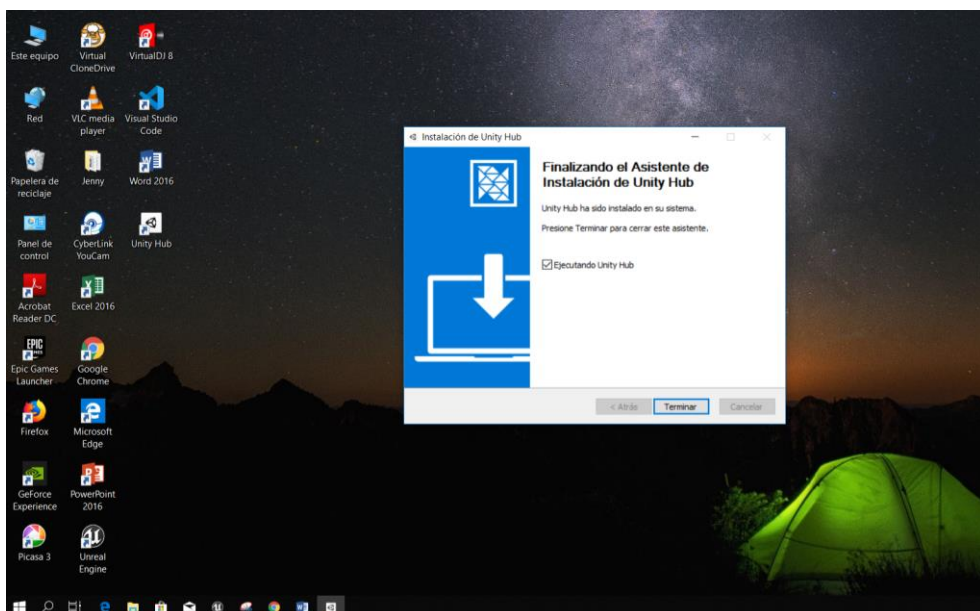


Figura 19. Descarga e instalación de Unity Hub Launcher.

- Iniciar el Unity Hub Launcher y empezara la descarga del motor de juegos Unity, a la par de este proceso se debe crear una cuenta en Unity

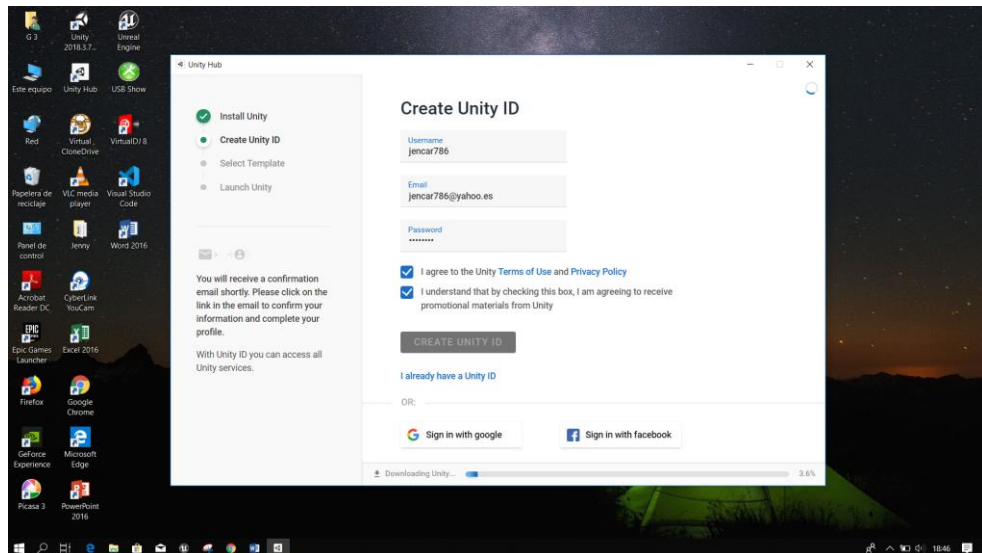


Figura 20. Crear cuenta en Unity.

- Al haber optado por la versión para principiantes, el Launcher nos permite seleccionar una plantilla de 3 opciones de juegos para tener una guía rápida de como iniciar el desarrollo de juegos en Unity.

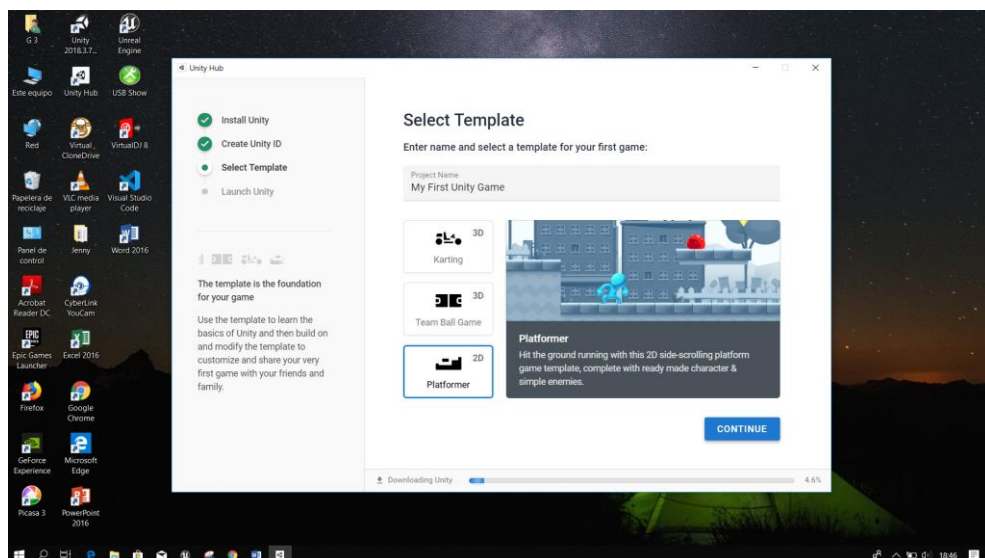


Figura 21. Plantillas prediseñadas para principiantes.

- Al término de la descarga se habilita la opción de Lanzar Unity y comenzar el desarrollo.

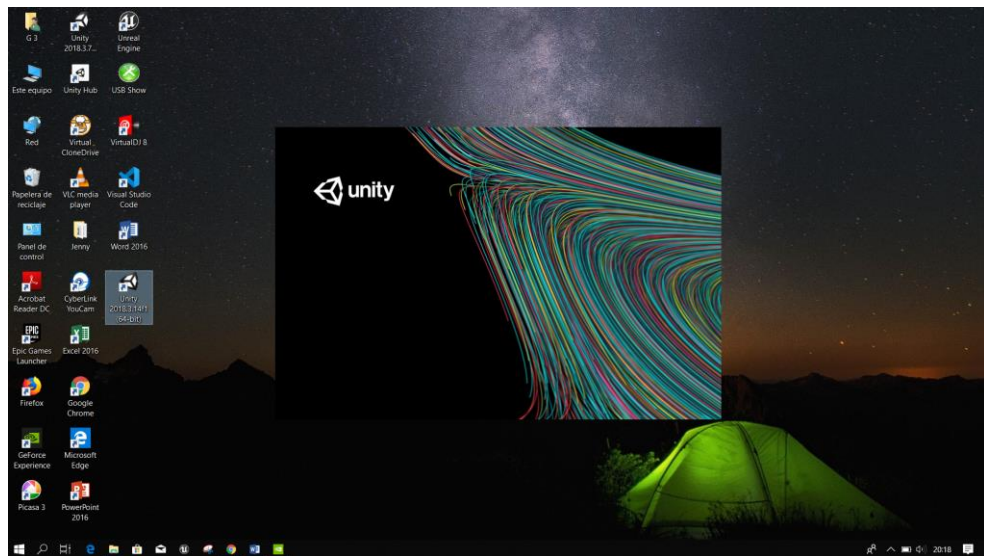


Figura 22. Arranque del motor de juegos Unreal Engine 4.

2.1.2 Unreal Engine

Para instalar Unreal Engine 4 se debe ingresar página oficial de Epic Games (<https://www.unrealengine.com/en-US/>) y realizar lo siguiente:

- Crear una cuenta de Epic Games.

A screenshot of the Epic Games account creation page. At the top, there is the Epic Games logo and the text "Crear Cuenta". Below this, there are five social media icons: PlayStation, Xbox, Nintendo Switch, Facebook, and Google. The form includes a dropdown menu for "INSCRIBIRSE" set to "ECUADOR". There are two input fields for first and last names, with "Jenny" and "Tomalá" entered. A third field contains the username "Jtomala" with an information icon. The email field contains "jctomalag@unl.edu.ec" with a link to "¿Quieres decir jctomalag@unl.edu?". A password field is shown with masked characters. A "No soy un robot" checkbox is checked, accompanied by a CAPTCHA image and links for "Privacidad" and "Condiciones". At the bottom, there are two checkboxes: "Me gustaría recibir las últimas noticias e información sobre los productos y servicios de Epic." (unchecked) and "He leído y acepto los términos de servicio." (checked). A blue "CREAR CUENTA" button is at the bottom, with a link to "Política de privacidad" below it.

Figura 23. Formulario para la creación de cuenta en Epic Games

- Descargue y ejecute el instalador (Epic Games Launcher Set-Up Program), hay que considerar las opciones de licencias para la descarga (ver Sección Anexos, Guía de instalación de Unreal Engine 4):

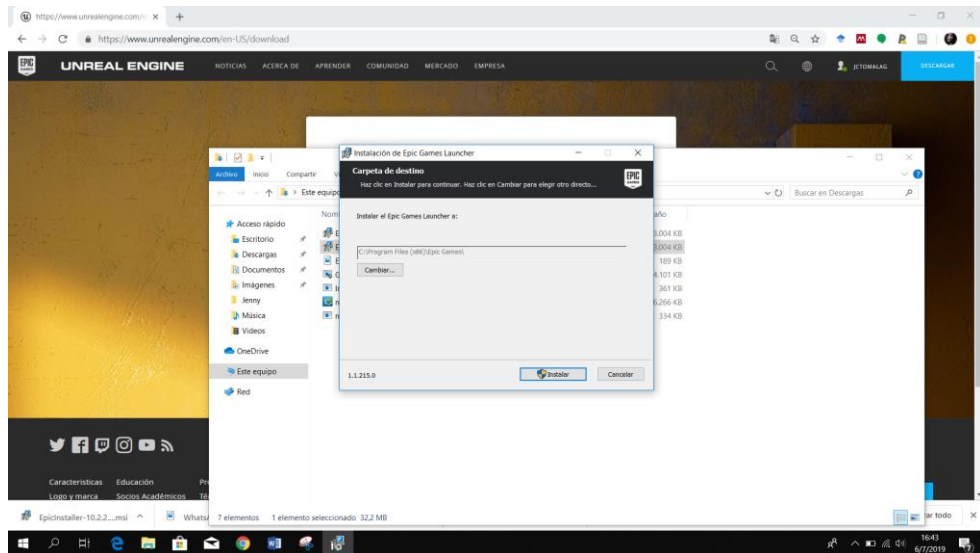


Figura 24. Descarga y ejecución del instalador

- Iniciar sesión en Epic Games Launcher

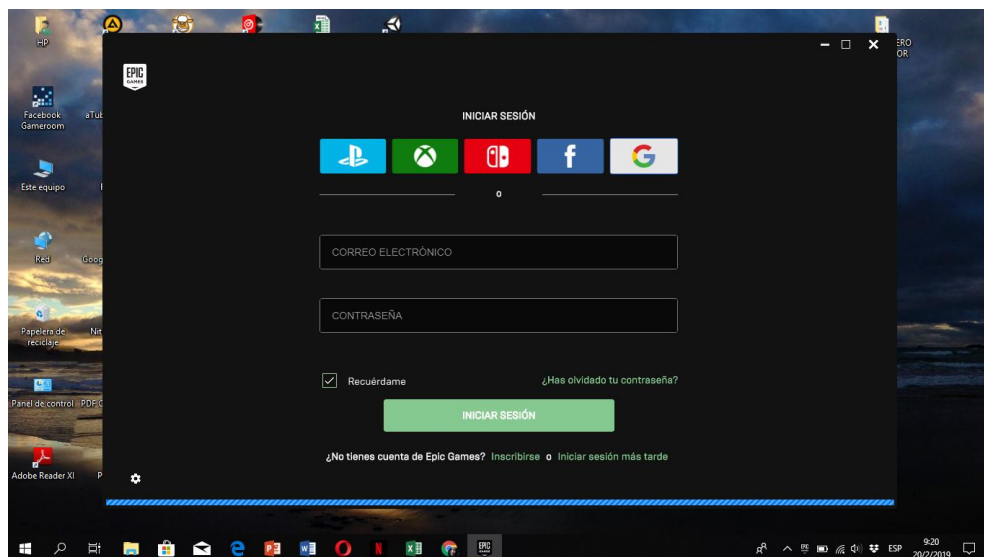


Figura 25. Inicio de sesión en Epic Games.

- Instalar el motor de Unreal Engine, por defecto saldrá la versión más reciente, pudiendo cambiar por la versión que se necesite.

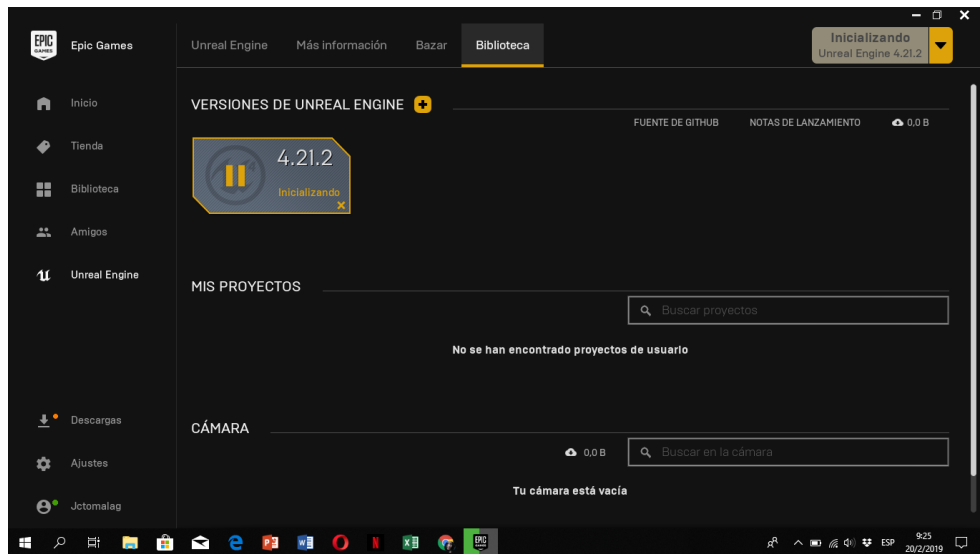


Figura 26. Instalación del motor de juegos.

- Lanzar Unreal Engine y comenzar el desarrollo.



Figura 27. Arranque del motor de juegos Unreal Engine 4.

2.2 Registrar y comparar los inconvenientes presentados al momento de instalar cada plataforma.

Al proceso de instalación es sencillo, pese a eso, se presentaron algunos inconvenientes en el motor de juegos Unreal Engine como:

- **UNREAL ENGINE 4:**

- El Proceso de instalación es lento y toma un tiempo considerable entre 5 a 8 horas como mínimo aproximadamente, se debería poseer una buena conexión a internet.
- Se necesita la instalación previa de una versión de visual estudio.
- Al momento de arrancar el motor no se puede abrir ningún archivo del paquete office.
- La descarga de los assets y la importación hacia los proyectos depende de la conexión a internet que tengamos.

2.3 Diseñar el prototipo de videojuego con la cual se realizará la evaluación.

Para el proceso de diseño del prototipo de videojuego se planteó la idea de crear “CLEAN RUNNER” es un Juego Serio de genero Endless Runner o Corredor sin fin, y su objetivo principal es: “Concienciar a los más pequeños sobre el reciclaje y la limpieza de las calles de su comunidad”. Para la documentación de este proceso, se utilizó el documento de diseño de videojuego (GDD) (Ver sección Anexo 2). A continuación, se muestran los puntos más esenciales del documento antes mencionado.

- **Concepto del juego**

Clean Runner es un juego de tipo Endless Runner (Corredor sin fin), en el que se controla con movimientos de izquierda a Derecha del personaje de la historia: el mismo que se llama Sergio, que en latín su significado es “guardián”. Sergio se caminará, a través de una serie de plataformas intentando recopilar el un mayor número establecido de botellas tiradas en el suelo, procurando que las calles queden lo más limpias posibles, y evitando los obstáculos que se le presenten en el camino de manera rápida, ya que al lograr cierta cantidad de botellas su velocidad ira incrementando paulatinamente y podría colisionar contra los mismos.

- **Jugabilidad**

Como se ha comentado anteriormente, se trata de un juego sencillo enfocado a la concientización, así que únicamente dispondrá de un nivel de juego, aunque fácilmente se podrían implementar otros niveles si fuera necesario.

- **Movilidad:** El personaje principal se moverá a través del escenario en 3ª persona, con las flechas de teclado.
- **Recompensas:** Cada vez que se recoja una botella del suelo este sumara puntos. Si las botellas no son recogidas no se descontarán los puntos obtenidos.
- **Muerte Súbita:** El jugador morirá cuando este no logre esquivar el obstáculo y choque con el mismo, en ese momento definitivamente se acaba el juego.

- **Diagrama de flujo**

El siguiente diagrama de estados muestra las pantallas presentes a lo largo de **Clean Runner** y las transiciones entre ellas. En puntos posteriores nos centraremos en ellas de forma individual.

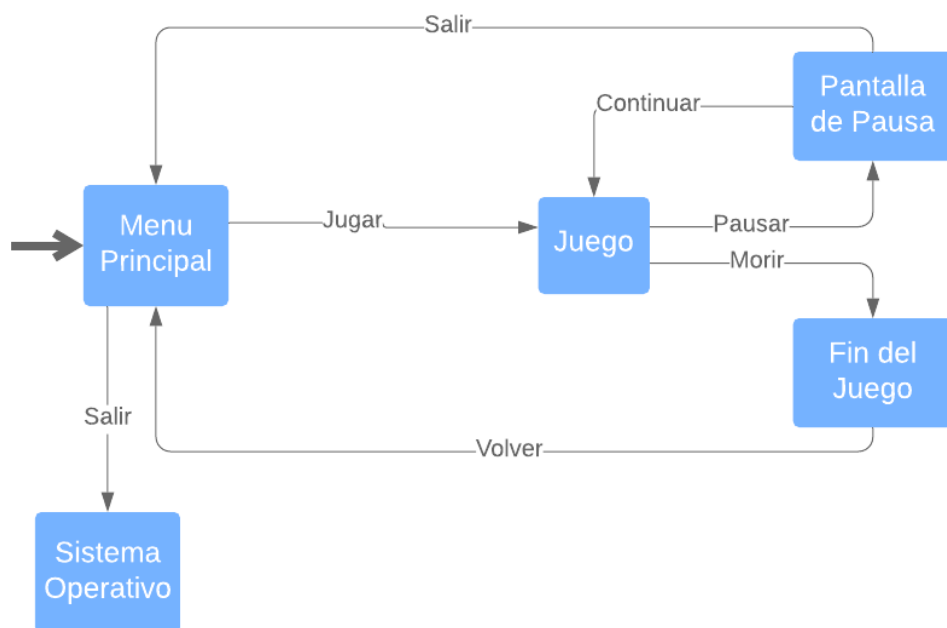


Figura 28. Diagrama de Estados del Juego.

- **Bocetos de pantallas**

A continuación, se presentan los bocetos de las pantallas principales:

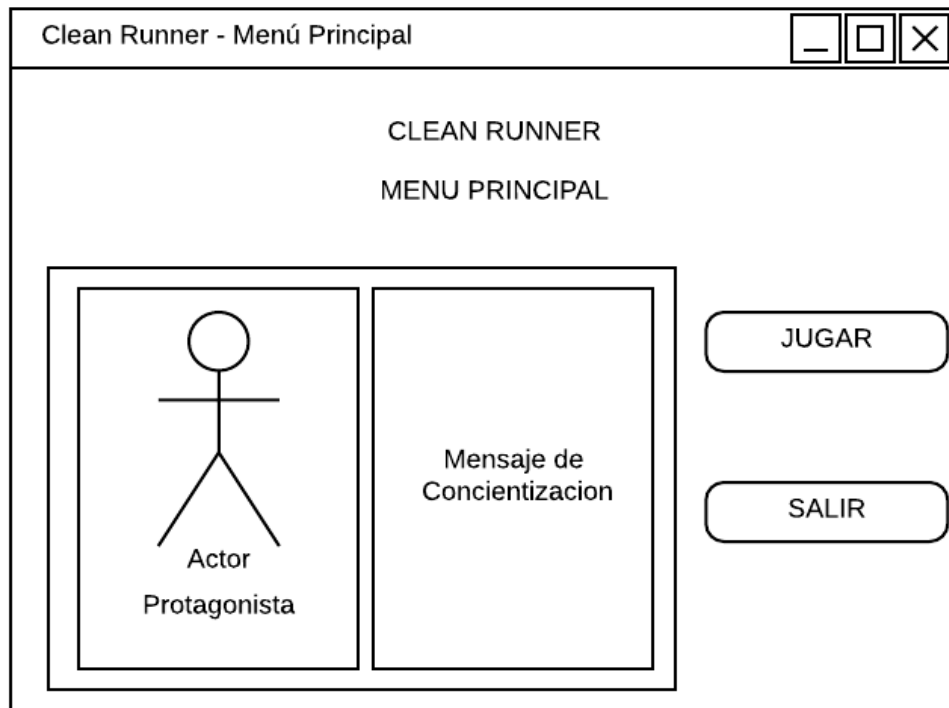


Figura 29. Boceto del menú principal.

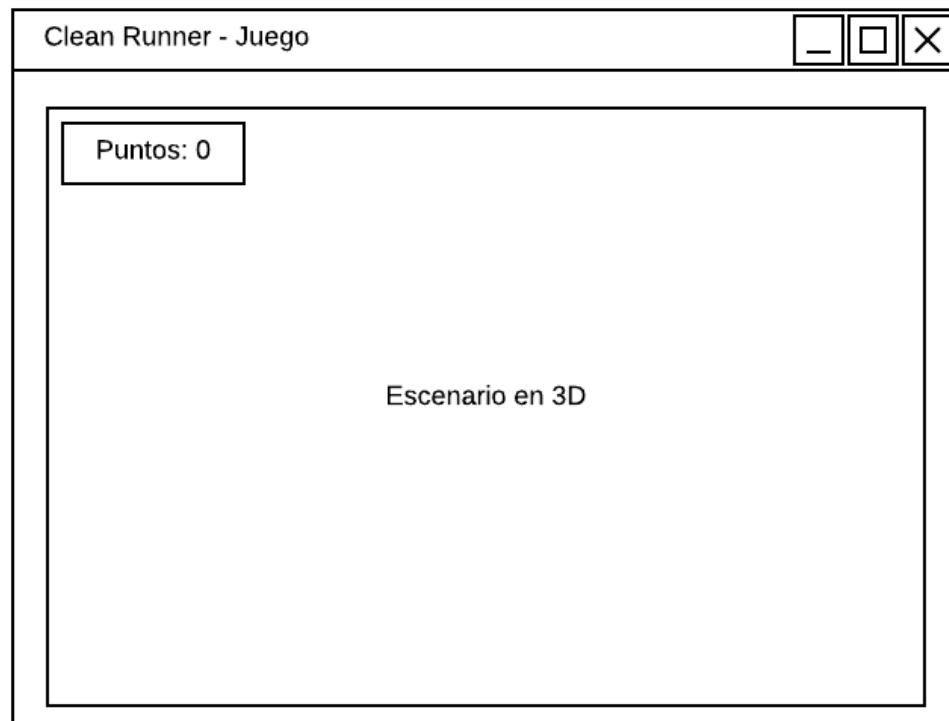


Figura 30. Boceto de la pantalla del juego.

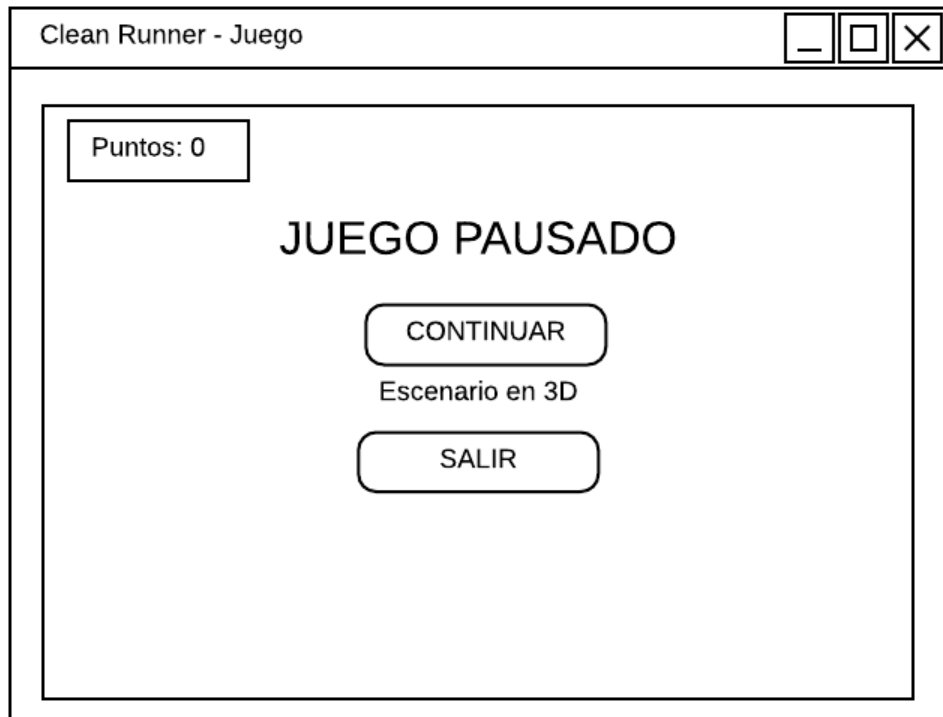


Figura 31. Boceto de la pantalla de Pause.

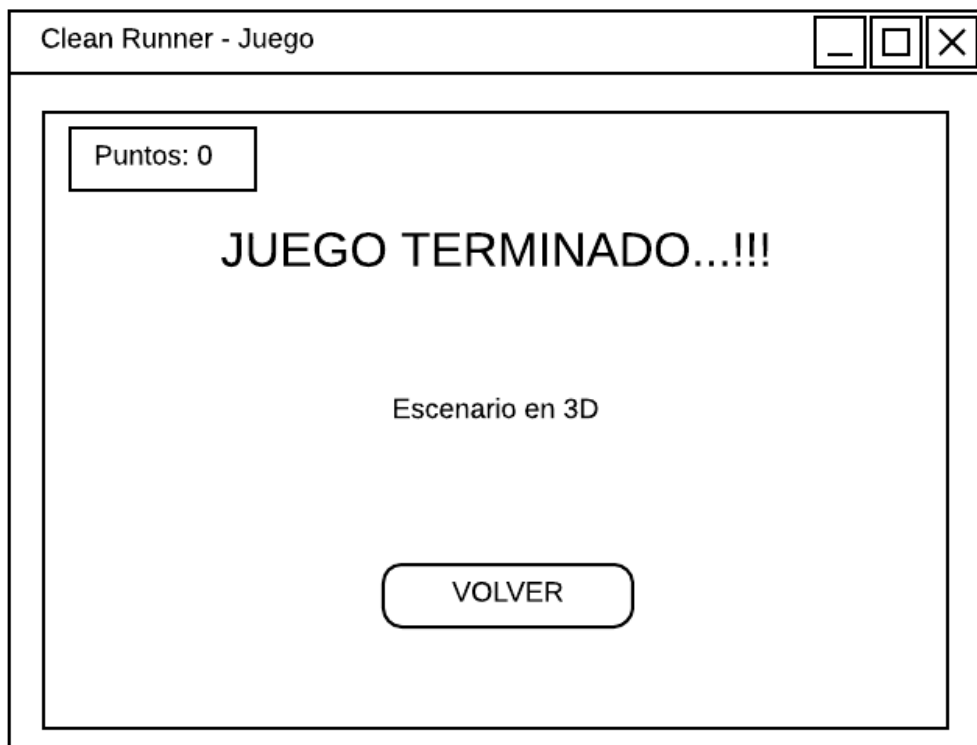


Figura 32. Boceto de la pantalla del juego terminado.

- **Arte 3D**

Todos los modelos 3D son guardados e importados en formato (.FBX) u (.OBJ) que son compatibles con los motores.

- El personaje posee animaciones tales de correr, girar, saltar y morir
- También existen objetos tales como:
 - Carretera.
 - Edificios - casas.
 - Cercas.
 - Botellas.
 - Hidrantes, etc.

2.4 Determinación de categorías y valores a evaluar durante el proceso de desarrollo del prototipo.

Al concluir el proceso de diseño, se consideró pertinente determinar varios puntos a evaluar en los motores de juegos durante el proceso de desarrollo del prototipo de manera experimental, cabe recalcar que cada categoría esta evaluada de manera cualitativa detallada a continuación:

- **Funcionalidad:** Permite medir la parte esencial del software: sus funcionalidades básicas y extras.
- **Usabilidad:** Permite evaluar el grado de facilidad que presenta la herramienta para ser manejada por el usuario/desarrollador.
- **Documentación:** Permite conocer la calidad y cantidad de información disponible acerca de la herramienta, lo cual es de gran importancia para su correcto manejo.
- **Tiempos de Respuestas:** Permite conocer los tiempos que demora el motor de juegos al responder a algunas acciones básicas.
- **Rendimiento:** Permite medir como trabaja la herramienta en términos de Hardware.

Luego de definir las categorías se procede a determinar que valores serán evaluados dentro de cada una de ellas. Estas se pueden observar en la Tabla VII:

TABLA VII. DETERMINACIÓN DE VALORES A EVALUAR

| CATEGORÍAS | VALORES |
|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Funcionalidad | <ul style="list-style-type: none">• Entorno integrado de desarrollo (IDE)• Lenguaje de desarrollo• Creación de interfaces o HUD• Importación de Contenido |
| Usabilidad | <ul style="list-style-type: none">• Facilidad de aprendizaje• Experiencia de interfaz de usuario final• Instalación completa de la herramienta |
| Documentación | <ul style="list-style-type: none">• Guía de instalación• Manuales / tutoriales• Capacitaciones |
| Tiempos de Respuestas | <ul style="list-style-type: none">• Tiempo de respuesta al abrir el motor• Tiempo de creación de un nuevo proyecto. |

Otras métricas a evaluar dentro de la categoría de rendimiento son los consumos de: GPU, memoria dedicada, y las temperaturas alcanzadas de las mismas, las cuales serán obtenidos con el software GPU-Z.

Para realizar la evaluación se ha procedido a establecer tanto la escala de medición como los valores cualificales, las mismas que se muestran en la Tabla VIII:

TABLA VIII. DETERMINACION DE VALORES PARA LA EVALUACION EXPERIMENTAL

| METRICAS | DESCRIPCION | PUNTAJE | | |
|---------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|------------------------------------------------------------|-----------------------------------------------|
| | | 1 (Inaceptable) | 2 (Aceptable) | 3 (Excelente) |
| Entorno integrado de desarrollo (IDE) | Se evalúa si el motor presenta un IDE que mejore o facilite el desarrollo. | No tiene IDE | IDE no propio del Motor | IDE propio del Motor |
| Sistema de Scripting utilizado | Se evalúa el nivel de complejidad que presentan los lenguajes con los que se programa dentro del motor. | Requiere conocimientos avanzados de programación | Requiere conocimientos de programación | No requiere conocimiento en programación |
| Creación de interfaces graficas GUI | Se evalúa la manera en que se crean las interfaces graficas dentro del motor. | | Mediante líneas de código | Mediante componentes gráficos |
| Importación de Contenido adicional | Se evalúa la disponibilidad de recursos adicionales necesarios para el prototipo. | Solo contenido de paga | Contenido gratuito y de paga | Contenido gratuito |
| Facilidad de aprendizaje | Se evalúa el tiempo /esfuerzo con la que nuevos usuarios desarrollan una interacción efectiva con la herramienta. | Más de 6 meses | De 2 a 6 meses | De 2 semanas a 1 mes. |
| Experiencia de interfaz del usuario desarrollador | Se evalúa el manejo con que se percibe la interfaz del motor (Interfaz/navegador de contenidos /panel de detalles). | Complejo, no se puede utilizar sin un manual. | Toma poco tiempo aprender, requiere cierto uso del manual. | Sencillo e intuitivo, no requiere de manual |
| Instalación completa del Motor | Se evalúa el tiempo que se tarda en conseguir la instalación completa del motor | Mas de 10 horas | De 6 a 10 horas | Menos de 6 horas |
| Guía de instalación | Se evalúa la existencia de una o varias guías de instalación del motor | No existe una guía de instalación | Guía de instalación oficial poco detallada | Guía de instalación bien detallada |
| Manuales / tutoriales | Se evalúa la existencia de manuales y tutoriales sobre el motor | No existen manuales o tutoriales del motor | Existen pocos manuales y tutoriales | Existe gran cantidad de manuales y tutoriales |
| Capacitación | Se evalúa la existencia y calidad de ofertas de capacitación sobre el motor | No existe capacitación identificada | Existen capacitaciones solo en ingles | Existen capacitaciones en inglés y español |
| Tiempo de respuesta al abrir el motor | Se evalúa el tiempo que tarda en abrir el editor del motor | Más de 5 minutos | Entre 2 a 5 minutos | Menos de 2 minuto |
| Tiempo de creación de un nuevo proyecto. | Se evalúa el tiempo que tarda en crearse un nuevo proyecto dentro del motor | Más de 5 minutos | Entre 2 y 5 minutos | Menos de 2 minuto |

2.5 Desarrollar el prototipo en cada uno de los motores de juegos a evaluar.

En este apartado se detalla las funcionalidades más importantes relacionado al prototipo de juego que se desarrolló:

DESARROLLO DEL VIDEOJUEGO CLEAN RUNNER

EN EL MOTOR UNREAL ENGINE 4

1. Creación del Proyecto

Ingresar al motor de juegos, elegir la opción crear un nuevo proyecto y tomar en cuenta las siguientes consideraciones:

- Señalar el sistema de Scripting a utilizar: lenguaje de programación C++ o mediante Blueprint, para este proyecto se manejó Blueprint.
- Optar por si se trabajará en un proyecto en blanco o usando una de las plantillas prediseñadas que trae el motor consigo, se utilizó la plantilla de tercera persona.
- Escoger la plataforma para el cual será desarrollado nuestra aplicación (en nuestro caso se eligió PC).
- Elegir la calidad gráfica que puede ser máxima calidad o escalable 2D o 3D, en el presente trabajo se lo realizo con la calidad gráfica máxima.
- Seleccionar si el proyecto comenzara con contenido preestablecido que nos proporciona el motor o sin contenido y,
- Finalmente definir la ruta donde se almacenará el proyecto.
- Asignar un nombre al proyecto en este caso Clean Runner, y daremos clic al botón Crear proyecto.

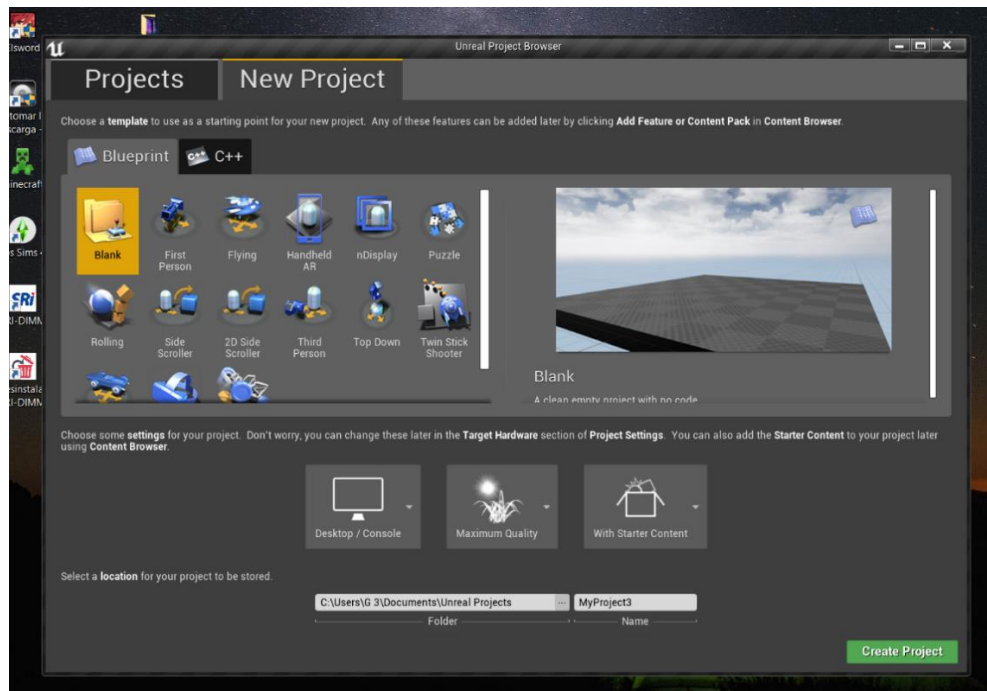


Figura 33. Pantalla de creación de proyecto.

2. Creación del Nivel del Juego

El primer paso para crear un juego es crear un nivel o mapa del nivel, para lo cual en la barra de menú nos vamos a File, y luego a New Level, en el cual podemos producir nuestro terreno, importar un landscape, o simplemente configurarlo para que inicie con un Blueprint Class, para efectos de este proyecto se utilizó un Blueprint Class.

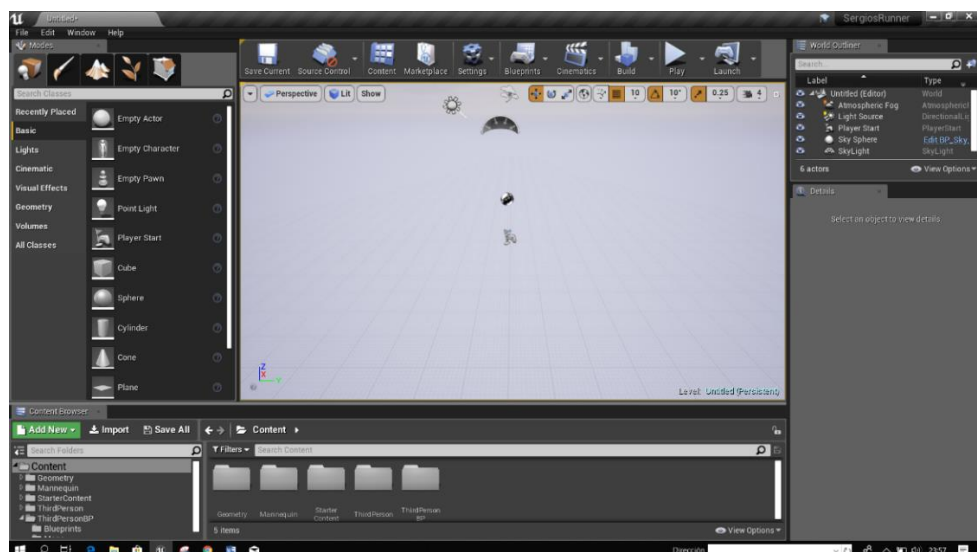


Figura 34. Pantalla de creación de nivel.

3. Acciones del Personaje

Dentro del Blueprint de personaje, nos ubicamos en el Event Graph y ahí modificaremos varias secciones de los paneles visuales para lograr que el personaje corra sin la necesidad de presionar ninguna tecla, de la misma manera hay que considerar que nuestro jugador solo puede moverse hacia la izquierda o la derecha y saltar. A continuación, se muestran los sistemas visuales de cada una de las acciones fundamentales del personaje:

– Correr sin parar:

Aquí se eliminó el Input del movimiento hacia adelante mediante y se lo remplazo por el evento Tick, es decir aquel evento que se ejecuta en cada fotograma. Si probamos podemos observar que el personaje comienza a correr sin para y que al presionar las teclas direccionales se mueve, pero no gira en su totalidad.

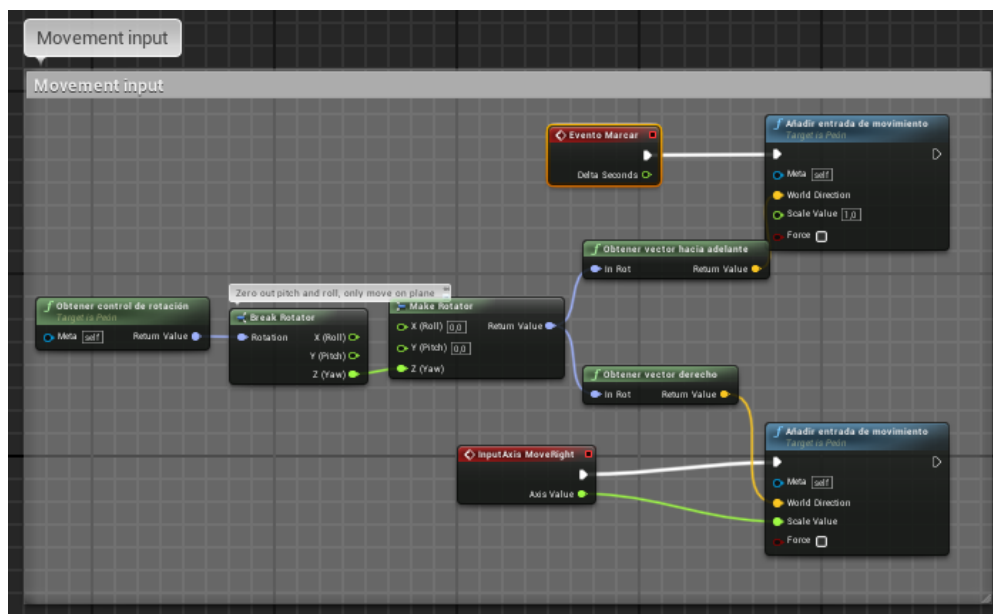


Figura 35. Acción de correr sin parar.

– Girar:

Para poder hacer que nuestro personaje gire de forma completa tenemos que configurar las entradas del teclado la A para la izquierda y D para la derecha, por lo cual nos dirigimos al apartado ajustes del proyecto y dentro de la sección Engine nos ubicamos en input, y en el action mapping crearemos e insertaremos dos mapeos una para girar

al lado derecho con la tecla D y con la A giraremos hacia la izquierda. Cabe recalcar que dentro de un mapping podemos configurar varias entradas.

Luego se procede a agregar los dos inputs creados **GirarIzquierda** y **GirarDerecha**, y procedemos a configurar la rotación que queremos que tenga el personaje al momento de girar a cualquier lado, para ello necesitaremos de una variable denominada **Rotacion** objetivo que será de tipo **Rotation**, adicional a eso necesitamos una variable de tipo **Booleano** para saber si se puede girar o no.

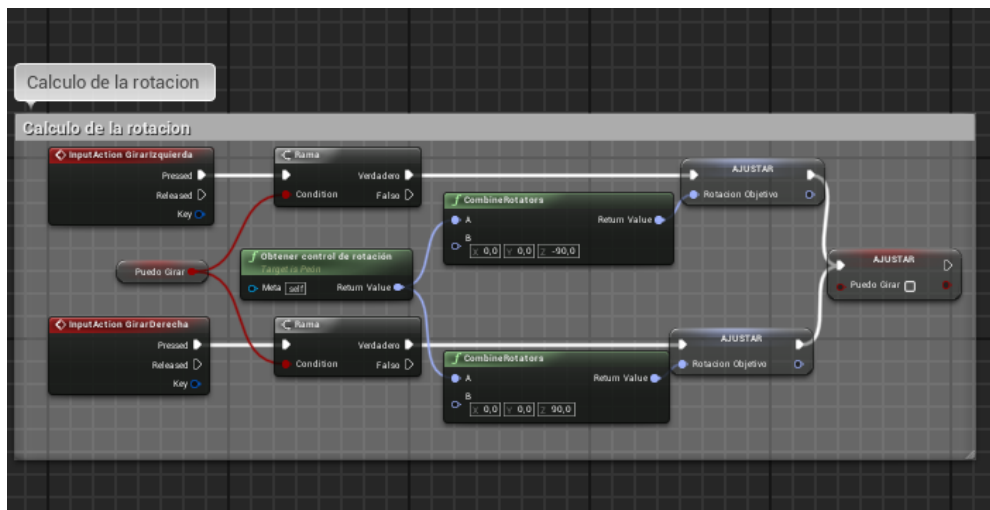


Figura 36. Determinación del grado de rotación para poder girar.

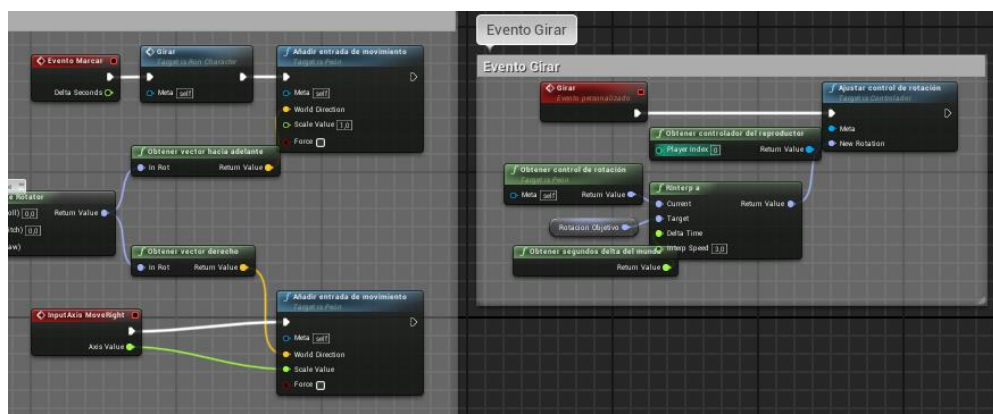


Figura 37. Acción de girar.

– **Morir**

Esta función se lleva a cabo cuando el personaje traslape ya sea con el obstáculo o choque con alguna pared. Hacemos uno de nodos tales como Ajustar visibilidad, movimiento desactivado y la habilitación del cursor del mouse para poder salir del juego.

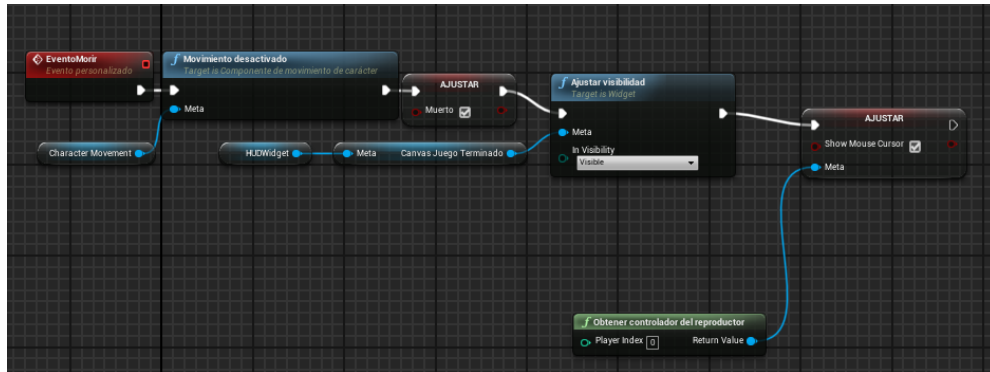


Figura 38. Acción de morir.

4. Sistema de Plataforma

Para la creación de las plataformas por el cual el personaje se desplaza, crearemos un Blueprint Class que tendrá el nombre de plataforma, en la cual añadiremos varios componentes como Static Mesh que podrán adoptar la forma y texturas que nosotros especifiquemos en el panel de detalle, acomodados todos los elementos emulando el ambiente que deseamos crear, aquí también haremos uso del sistema visual de scripting para crear algunos eventos:

– **Sistema de creación de plataforma**

Aquí determinaremos puntos finales de cada una de las plataformas para que a partir de ahí se genere otra plataforma que den como resultados un camino sin fin, cabe recalcar que la generación de plataformas es aleatoria, si ejecutamos el juego unas 5 veces, el diseño cambiara por cuanto utilizamos el nodo random.

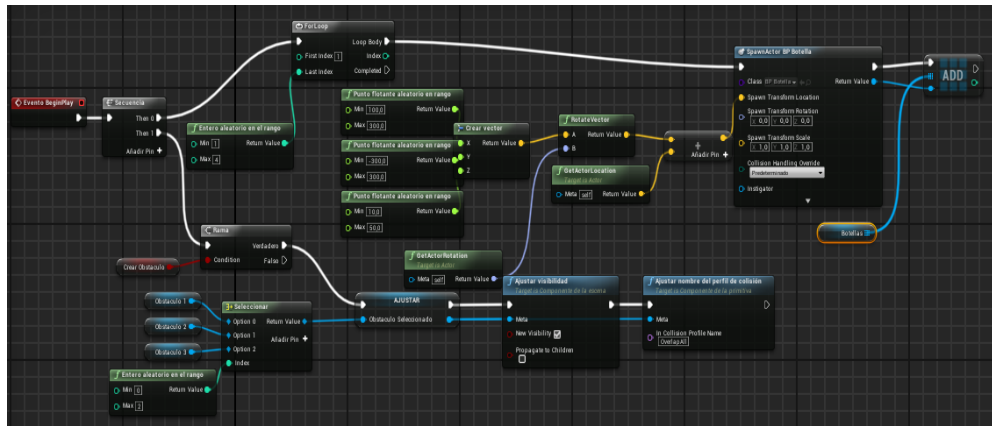


Figura 39. Sistema de creación de plataforma.

– Sistema de creación de obstáculos.

Este sistema se crea en cada una de las plataformas creadas.

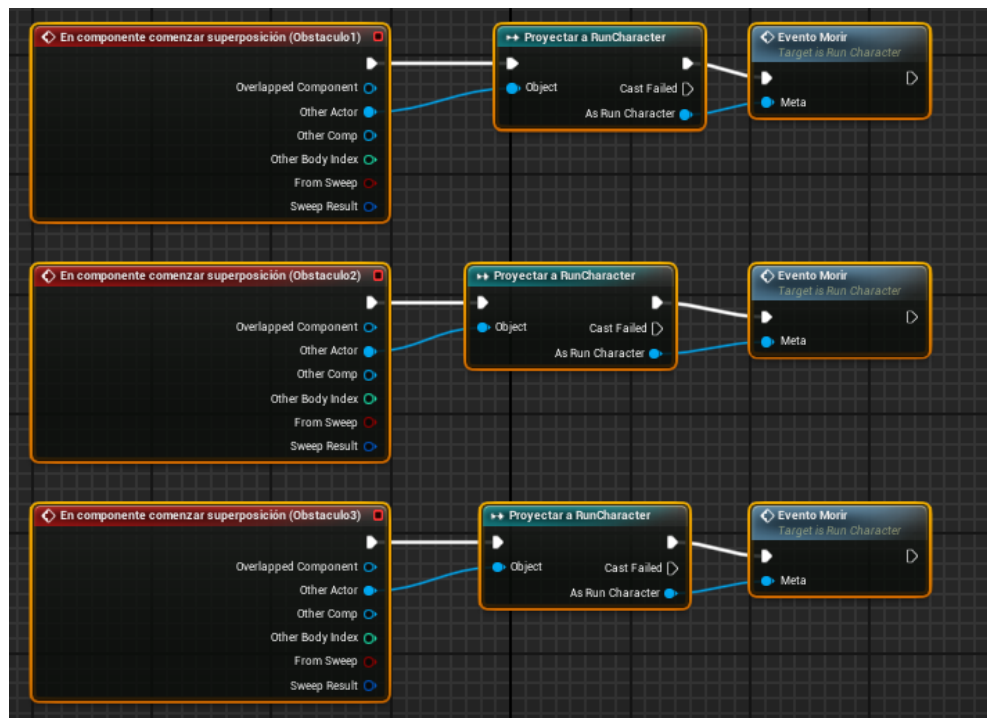


Figura 40. Sistema de creación de obstáculos.

– Sistema de destrucción de las plataformas.

Se realiza la destrucción de la plataforma con todos sus componentes luego de 1 min de tiempo para evitar que se traslapen y choquen una con otra.

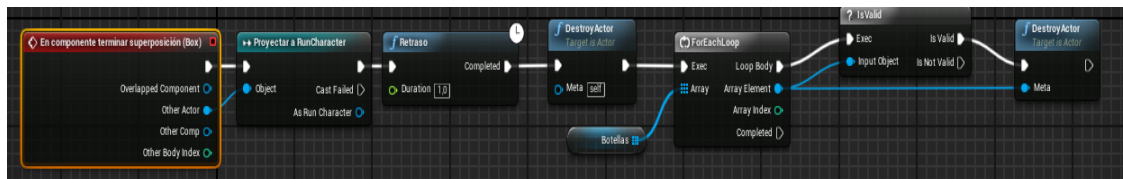


Figura 41. Sistema de destrucción de plataformas.

– **Generación – Recolección – y Destrucción de objetos:**

Se generan los objetos a recoger a partir del inicio del juego de manera aleatoria por el área de la plataforma, el mismo que tendrá un movimiento giratorio, se le accionó un sonido que se reproducirá al momento que el personaje colisiones con él y en ese instante el objeto se destruye para dar la apariencia de recolección.

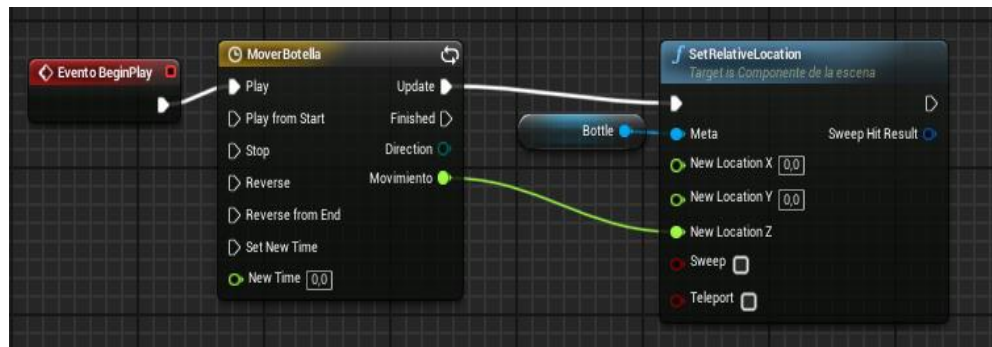


Figura 42. Sistema de generación de objetos.

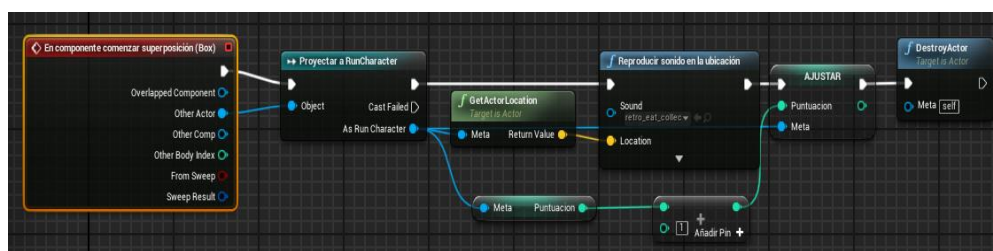


Figura 43. Recolección y Destrucción Objetos.

Generación de Componentes GUI:

Los componentes GUI son aquellas pantallas que se crean para la presentación de los videojuegos y que en su mayoría se encuentran superpuestas en el juego.



Figura 44. Generación Componentes GUI.

DESARROLLO DEL VIDEOJUEGO CLEAN RUNNER EN EL MOTOR UNITY

1. Creación del Proyecto

Ingresar al motor de Juegos y crear un nuevo proyecto, asignar el nombre al proyecto, para este caso se lo denominará Clean Runner.

Elegir la calidad gráfica 2D o 3D, para el presente trabajo se optó por trabajar con la calidad gráfica máxima.

Seleccionar si se desea que el proyecto tenga contenido inicial preestablecido, nos vamos a Add Assets Package y se selecciona los paquetes necesarios.

Finalmente se define la ruta donde se almacenará el proyecto y daremos clic al botón Crear proyecto.

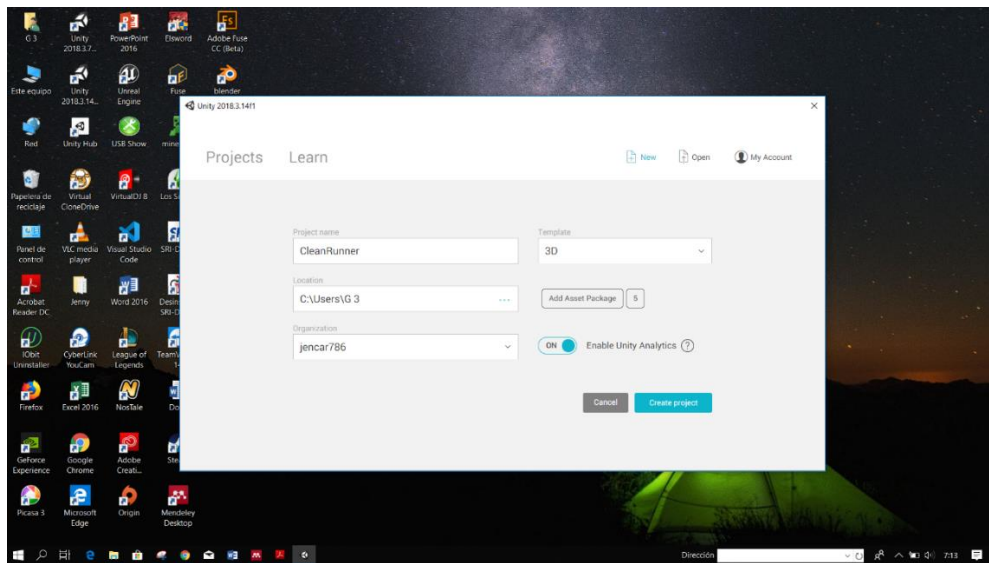


Figura 45. Pantalla de creación de proyecto.

2. Creación de la escena del Juego

El primer paso para crear un juego es crear una escena, para lo cual en la barra de menú nos vamos a File, y luego a New Scene, en el cual podemos crear nuestro terreno, o importar un landscape.

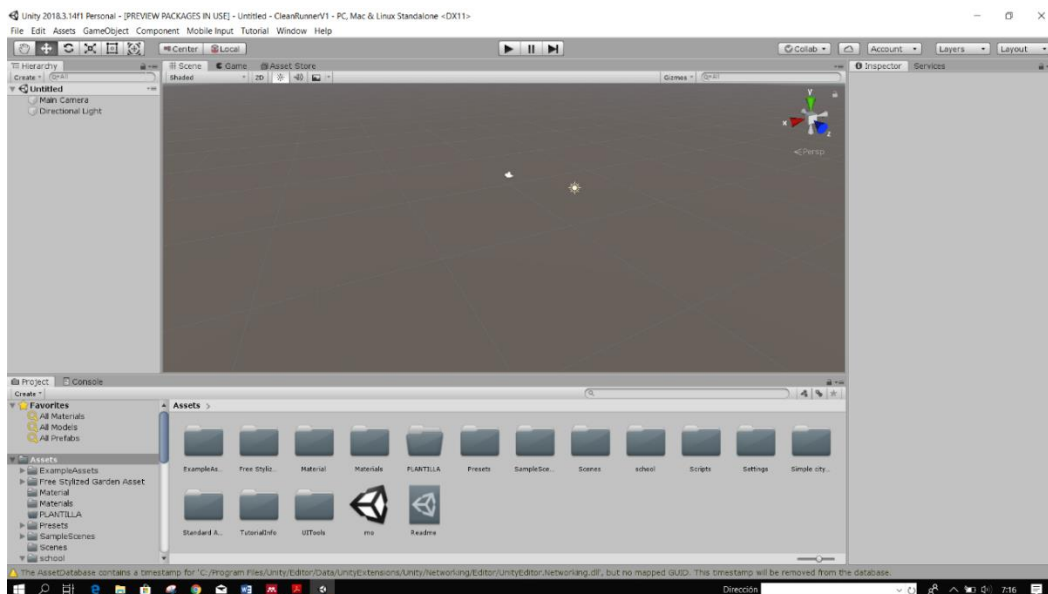
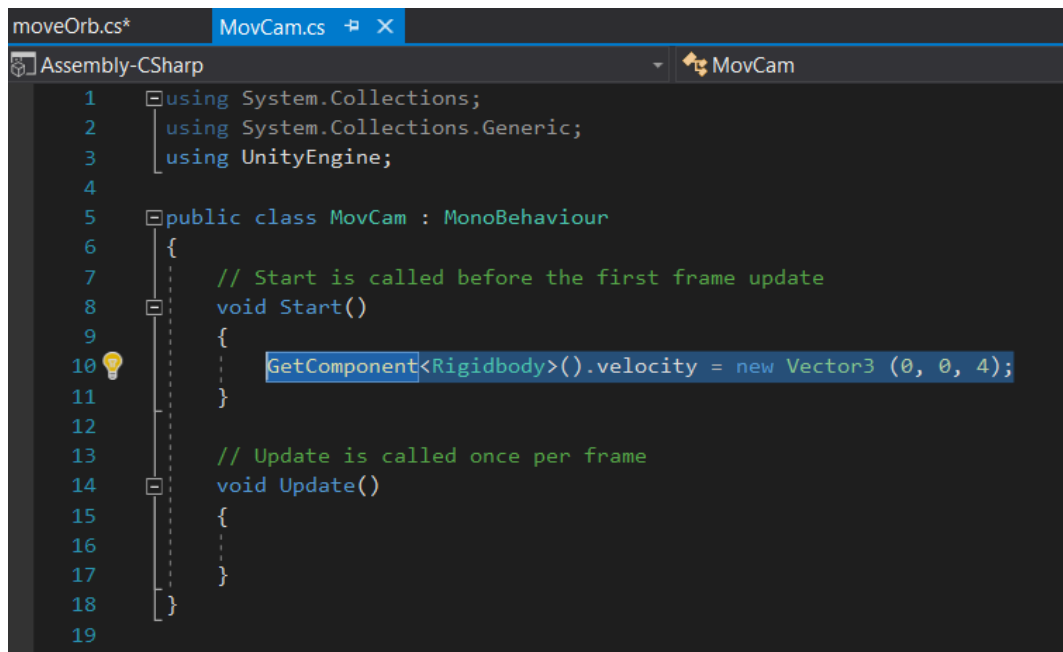


Figura 46. Pantalla de creación de Scene.

3. Acciones del Personaje

Para crear el movimiento del personaje se realiza un script para dar movimiento a la cámara, como es un proyecto de tercera persona la cámara debe tener una perspectiva desde atrás del actor, por lo cual la codificación se muestra en la figura siguiente:



```
1 using System.Collections;
2     using System.Collections.Generic;
3     using UnityEngine;
4
5     public class MovCam : MonoBehaviour
6     {
7         // Start is called before the first frame update
8         void Start()
9         {
10            GetComponent<Rigidbody>().velocity = new Vector3 (0, 0, 4);
11        }
12
13        // Update is called once per frame
14        void Update()
15        {
16            //
17        }
18    }
19
```

Figura 47. Script configuración del movimiento de la cámara.

Correr sin parar:

Como el personaje es un corredor sin fin, se procede a añadirle el movimiento hacia adelante y hacia los lados (izquierda - derecha) al presionar las teclas A o D respectivamente, todo esto mediante un script al cual se le codifico de la siguiente manera:

```
moveOrb.cs x MovCam.cs
Assembly-CSharp
moveOrb
moveL

4 public class moveOrb : MonoBehaviour
5 {
6     public KeyCode moveL;
7     public KeyCode moveR;
8     public float horizVel = 0;
9     public int laneNum = 2;
10    public string controlLocked = "n";
11
12    void Start()
13    {
14    }
15
16    void Update()
17    {
18        GetComponent<Rigidbody>().velocity = new Vector3(horizVel, 0, 4);
19        if((Input.GetKeyDown(moveL)) && (laneNum > 1) && (controlLocked == "n"))
20        {
21            horizVel = -2;
22            StartCoroutine(stopSlide());
23            laneNum -= 1;
24            controlLocked = "y";
25        }
26        if ((Input.GetKeyDown(moveR)) && (laneNum < 3) && (controlLocked == "n"))
27        {
28            horizVel = 2;
29            StartCoroutine(stopSlide());
30            laneNum += 1;
31            controlLocked = "y";
32        }
33    }
34    IEnumerator stopSlide()
35    {
36        yield return new WaitForSeconds(.5f);
37        horizVel = 0;
38        controlLocked = "n";
39    }
40 }
```

Figura 48. Script de configuración del movimiento del personaje.

– **Generación, Recolección y Destrucción de objetos:**

```
void OnCollisionEnter(Collision other)
{
    if (other.gameObject.name == "Capsule")
    {
        Destroy(other.gameObject);
    }
}
```

Figura 49. Script de destrucción de objetos.

```
efectos.cs x moveOrb.cs MovCam.cs
Assembly-CSharp efectos
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class efectos : MonoBehaviour
6 {
7     // Start is called before the first frame update
8     void Start()
9     {
10
11     }
12
13     // Update is called once per frame
14     void Update()
15     {
16         transform.Rotate(3, 0, 0);
17     }
18 }
19
```

Figura 50. Script de efectos del objeto.

3. FASE 3: Evaluar los motores de juegos de acuerdo a los criterios de comparación establecidos.


3.1 Comparar los motores de juegos de acuerdo a los criterios de evaluación determinados.

En esta etapa se realiza la comparación de los motores de juegos en base a los criterios determinados tanto en la Revisión Sistemática de Literatura, como los establecidos en la etapa de diseño, los mismos que se presentan a continuación:

COMPARATIVA DE CARACTERÍSTICAS BÁSICAS

Esta comparativa se la realizo en base a los criterios determinados en la Revisión Sistemática de Literatura realizada en el primer objetivo (Véase sección Anexo 1).



TABLA IX. COMPARATIVA DE CARACTERÍSTICAS BÁSICAS

| CRITERIOS | MOTORES DE JUEGOS | |
|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| |  |  |
| Desarrollador | Unity | Epic Games |
| Plataformas de Distribución | IOS, Android, Windows, Windows Store Apps, WindosPhone, Mac OS, Linux/Steam OS, WebGL, PS4, PS3, PlayStation Vita, Samsung, XboxOne, Xbox 360 Nintendo 3DS, Oculus Rift, Google Cardboard Android & IOS, Steam VR PC & Mac, PlayStation VR, Gear VR, Windows Mixed Reality, DayDream, Android TV, TvOs, Nintendo Switch, Apple ARKit, Google ARCore, Vuforia, WiiU | HTML5, Nintendo Switch, DayDreams, Linux, Android, PS4, XboxOne, Windows, OSX, PlayStation VR, Steam VR, IOS, Oculus, Samsung, Gear VR, y Vive |
| Lenguaje de programación | C#, Boo, Js | C++, Blueprints |
| Documentación Disponible | <ul style="list-style-type: none"> • Wiki. • Foros. • Video tutoriales. • Manual de usuario. • Blogs • Aprendizaje en Vivo | <ul style="list-style-type: none"> • Wiki. • Foros. • Blogs • Marketplace. • Video tutoriales. • Aprendizaje en Vivo |
| Curva de Aprendizaje | <ul style="list-style-type: none"> • Sencilla | <ul style="list-style-type: none"> • Complicada si se desarrolla en C++ |
| Requisitos de Instalación | <ul style="list-style-type: none"> • SO, Windows 7 SP1, 8/8.1 y 10 de 64 bits. • Procesador: Intel Core i7-3770 @ 3.4 GHz o AMD FX-8350 @ 4.0 GHz o superior. • NVIDIA GeForce GTX 780 o AMD Radeon R9 290X • Memoria de 8GB de RAM • Tarje grafica compatible con DirectX9.0c • Disco duro de 50GB | <ul style="list-style-type: none"> • SO, Windows 10 de 64 bits • Procesador: Intel o AMD de cuatro núcleos, 2.5 GHz o más rápido. • Memoria de 8GB de RAM • Tarje grafica compatible con DirectX11 o DirectX12 |
| Licencia | <ul style="list-style-type: none"> • Versión Personal gratuita • Versión de paga para desarrolladores avanzados | <ul style="list-style-type: none"> • Versión Personal gratuita • Versión de paga para desarrolladores avanzados |

COMPARATIVA DE COMPONENTES BÁSICOS

Al igual que la comparativa anterior se la realizo en base a los criterios determinados en la Revisión Sistemática de Literatura realizada en el primer objetivo (Véase sección Anexo 1).

TABLA X. COMPARACIÓN DE COMPONENTES BÁSICOS

| CRITERIOS | MOTORES DE JUEGOS | |
|-------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| |  unity |  UNREAL ENGINE |
| Renderización | <ul style="list-style-type: none"> • Iluminación Global en tiempo real. • Sombreado a base de física • UI Intuitivo • Skybox • Combinación de Figuras para expresiones Faciales • Soporte sobre LOD. • Motor de sombreado para occlusion culling. • Sistema Modular de partículas | <ul style="list-style-type: none"> • Librería Diredx11 • Iluminación global, • Sombreado diferido, traslucidez, materiales, texturas y partículas. • Animación de personas • Cinemática Matinée • Terrenos y Follaje • Efecto Post-procesamiento • Efectos Visuales en Cascada |
| Animación | <ul style="list-style-type: none"> • Animaciones retargetables. • Clips de animación. • Controladores de animaciones. • Rendimiento y optimización. • Glosario de términos de animación | <ul style="list-style-type: none"> • Máquinas de estado. • Importe de animaciones. |
| Física | <ul style="list-style-type: none"> • Controles de física por guiones • Posee dos motores de Física, uno para 2D (Rigidbody 2D) y el otro para 3D (Rigidbody) • Articulaciones • Controladores de personajes • Detección continua de colisiones. • Depuración Física | <ul style="list-style-type: none"> • Motor de Física PhysX 3.3 • Detección de Colisiones precisas. • Componente Destructible. • Componente de restricción física • Componente físico de mando |
| Audio | <ul style="list-style-type: none"> • Grabar audio desde cualquier micrófono disponible • Audio mixer • Soporta varios formatos de audio como: AIFF, WAV, MP3 y Ogg | <ul style="list-style-type: none"> • Editor Sound Cue |
| Multi-jugador / Conectividad | <ul style="list-style-type: none"> • Modo Cliente Servidor | <ul style="list-style-type: none"> • Modo cliente Servidor |
| IA | NavMesh y Path Finding | Path Finding |

COMPARATIVA EXPERIMENTAL

A la par del desarrollo de los prototipos se fueron evaluando ciertas métricas cualitativas establecidas y especificadas en la etapa de diseño (Ver Tabla VIII de este documento), esta evaluación se encuentra en la Tabla XI que se me presenta a continuación:

TABLA XI. EVALUACIÓN EXPERIMENTAL

| CATEGORÍA | MÉTRICAS | GRADO COMPLEJIDAD | | | | | |
|-----------------------|---------------------------------------------------|--------------------|--------|------------------|--------|------------------|--------|
| | | 1 (Inaceptable) | | 2 (Aceptable) | | 3 (Excelente) | |
| | | Unity | Unreal | Unity | Unreal | Unity | Unreal |
| FUNCIONALIDAD | Entorno integrado de desarrollo (IDE) | | | | | X | X |
| | Sistema de Scripting utilizado | | | X | | | X |
| | Creación de interfaces graficas GUI | | | | | X | X |
| | Importación de Contenido adicional | | | X | X | | |
| USABILIDAD | Facilidad de aprendizaje | | | X | | | X |
| | Experiencia de interfaz del usuario desarrollador | | | X | X | | |
| | Instalación completa del Motor | | | X | X | | |
| DOCUMENTACIÓN | Guía de instalación | | | | | X | X |
| | Manuales / tutoriales | | | X | | | X |
| | Capacitación | | | X | | | X |
| TIEMPOS DE RESPUESTAS | Tiempo de respuesta al ejecutar el prototipo | | | | | X | X |
| | Tiempo de creación de un nuevo proyecto. | | | X | | | X |

La Figura 51, muestra q las curvas obtenidas del resultado de la comparación experimental sobre el grado de complejidad durante el desarrollo del prototipo planteado, de un rango de 0 a 7 siendo 7 el grado más alto de complejidad y 0 de menos complejidad que poseen los motores de juegos, se puede observar que tienen una diferencia notable en ámbitos como: el sistema de Scripting utilizado, en Unity se hace uso de Scripting basados en el lenguaje de programación C# por el cual debemos tener conocimientos sobre el mencionado lenguaje, mientras que en Unreal engine nos da la opción de trabajar con un sistema visual de Scripting el mismo que está basado en nodos interconectados por lo que se requiere mínimos conocimientos de programación de manera general, es mediante este sistema que muchos desarrolladores novatos pueden realizar un aplicación básica y efectiva en aproximadamente 4 semanas, demostrando así la facilidad de uso y aprendizaje, que diferencia a Unreal de Unity.

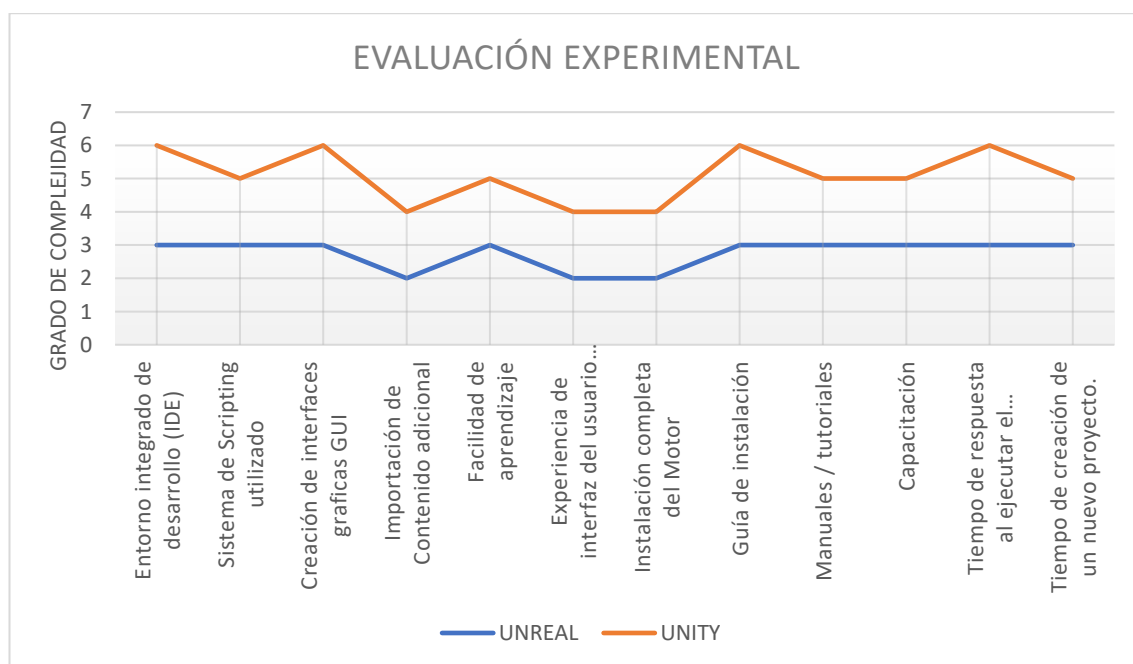


Figura 51. Resultados de la Evaluación Experimental.

Para determinar la evaluación de métricas tales como: Consumo de GPU, memoria dedicada y las temperaturas alcanzadas tanto de las mismas, se utilizó el software GPU-Z, estas pruebas de rendimiento fueron realizadas con una computadora de las siguientes características:

- Laptop HP modelo 14-bs015la
- Sistema Operativo: Windows 10 Pro 64 bits
- Procesador: Intel® Core™ i5-7200U CPU @ 2.50GHz (4 CPUs), 3.1GHz
- Memoria RAM: 8GB
- Tarjeta gráfica integrada: Intel (R) HD Graphics 620
- Versión de DirectX: DirectX 12

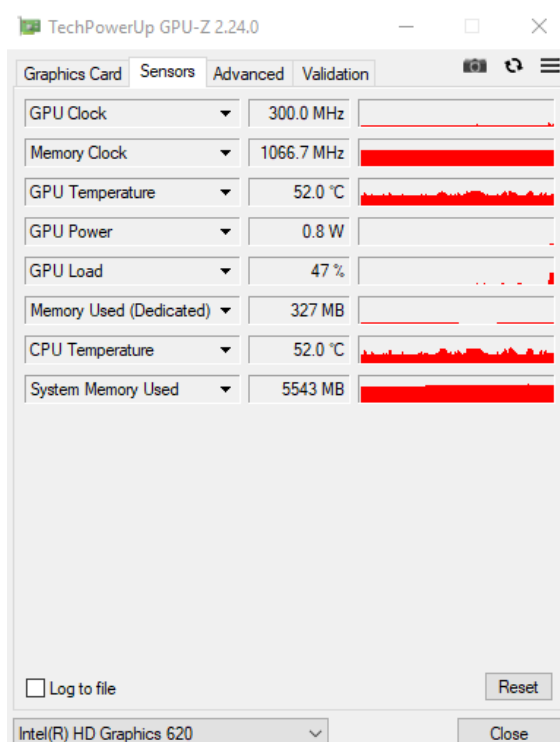


Figura 52. Resultados del Rendimiento del GPU al ejecutar Unity.

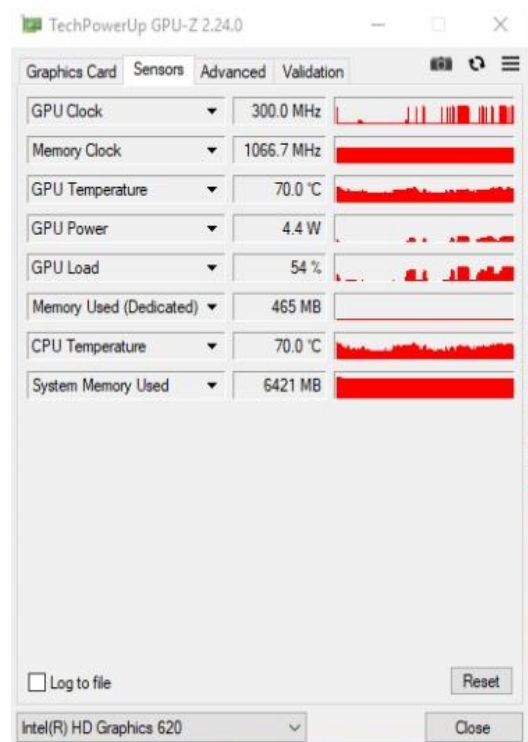


Figura 53. Resultados del Rendimiento del GPU al ejecutar Unreal Engine.

En la Figura 52 y Figura 53 se puede evidenciar el uso de recursos de GPU que realizan ambos motores al ser ejecutados, la gran diferencia que existe da como resultado que, si se necesita elegir un motor de acuerdo al equipo, obviamente tendríamos que elegir Unity, ya que el consumo de memoria es menor al de Unreal Engine con una diferencia de 1122 MB y el calentamiento que sufre el CPU dista en 18 °C entre ambos motores.

COMPARATIVA DURANTE LA CONSTRUCCION

A más de las comparativas realizadas, se ha previsto realizar una comparativa orientada a la calidad gráfica, disponibilidad e importación de contenido y a la fluidez del trabajo al momento de la producción del prototipo en los dos motores de juegos Unity y Unreal Engine, abarcando aspectos tales como:

- **Renderización en Tiempo Real:**

En Unreal Engine la renderización se puede visualizar en tiempo real al momento de llevar a cabo el desarrollo del videojuego, debido a que, si a tomamos como referencia el personaje de nuestra escena y le aplicamos un zoom este se acercará, pero en ningún momento se vera la pixelación del mismo, lo contrario que sucede con Unity que al realizar el mismo procedimiento se puede observar una gran pixelación.

- **Sombras y Cámaras:**

Unreal trae los shaders y efectos de cámara activados de manera determinada lo que permite ahorrar tiempo si desarrolla un escenario realista, facilitando así el trabajo del desarrollador, lo que no sucede con Unity donde se necesita activar este tipo de funciones mediante scripts programables y si la versión que se utiliza no trae consigo estas funciones, se las debería adquirir en el assets store y luego proceder a incorporarlas y modificarlas a nuestras necesidades.

- **Tienda de Contenidos**

Ambos cuentan con una tienda de contenidos, en Unity se la conoce como Assets Store y se encuentra dentro del editor para mayor comodidad del desarrollador, cabe mencionar que ofrece contenido gratuito y de paga que se renuevan en tiempo considerable; mientras que, en Unreal se lo conoce Marketplace, el cual esta implementado dentro del Lanzador, este ofrece de igual manera contenido gratuito y de paga pero con la diferencia que mensualmente la empresa libera nuevo contenido gratuito, y si necesitamos un contenido tendríamos que ir al lanzador, buscarlo, descargarlo y del mismo lanzador seleccionar el proyecto al cual queremos importarlo.

- **Integración de Contenido**

La integración de contenido adicional en ambos motores es fácil, aunque se debe admitir que al importar texturas Unreal la carga toma un tiempo considerable, debido a que este motor realiza el proceso de asignación de sombras e iluminación automáticamente.

- **Fluidez en el trabajo**

La rapidez para trabajar que nos ofrece Unreal es debido a dos características sobresalientes, el primero el sistema de Scripting visual que prácticamente lo realiza todo visualmente a través de arrastrar y soltar nodos y luego interconectarlos entre sí, y la segunda característica son las plantillas predeterminadas que este trae consigo, las mismas que pueden ser adaptadas a cualquier plataforma que este soporte, nada comparado con el desarrollo desde cero que Unity nos brinda en el cual debemos conocer el lenguaje de scripting que este maneja y para el cual debemos definir en primera instancia la plataforma que soportará nuestro juego.

Luego de haber obtenido esta comparativa durante la construcción del prototipo, se procede a determinar una escala de medición y métricas cuantitativas del mismo, la misma que se muestra a continuación en la Tabla XII y posteriormente se procede a realizar la evaluación, la cual se puede evidenciar en la Tabla XIII:

TABLA XII. DETERMINACIÓN DE ESCALA DE MEDICIÓN Y PESOS

| CATEGORÍA | CARACTERÍSTICAS | ESCALA DE MEDICIÓN | | PESO (%) |
|-----------------|-------------------------------------|-------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|----------|
| | | De 0 a 10 (%) | De 10 a 20 (%) | |
| Calidad Grafica | Renderización en Tiempo Real | Si permite y visualiza la renderización solo al momento de ejecutar el juego. | Si permite y visualiza la renderización en tiempo real y no solo al momento de ejecutar el juego. | 20 |
| | Sombras y Cámaras | Si el ajuste de sombras no se lo realiza automáticamente | Si permite el ajuste de sombras automáticamente | 20 |

| | | | | |
|-------------------------------------------------|---------------------------------|--------------------------------------------------------------|-----------------------------------------------------------------|--------------|
| Disponibilidad de Contenidos adicionales | Tienda de Contenidos | No integrada en el Editor | Integrada en el Editor | 20 |
| | Integración de Contenido | Se requiere de modificaciones por parte del desarrollador | Se acopla a la importación sin problema alguno | 20 |
| Agilidad en el desarrollo | Fluidez en el trabajo | Se necesita conocimientos previos para empezar el desarrollo | No se necesita conocimientos previos para empezar el desarrollo | 20 |
| TOTAL: | | | | 100 % |

TABLA XIII. EVALUACIÓN DE LOS MOTORES DE JUEGOS EN LA FASE DE CONSTRUCCIÓN

| CATEGORÍA | CARACTERÍSTICAS | PESO EVALUADO | |
|-------------------------------------------------|-------------------------------------|----------------------|---------------|
| | | UNITY | UNREAL |
| Calidad Gráfica | Renderización en Tiempo Real | 10% | 16% |
| | Sombras y Cámaras | 10% | 20% |
| Disponibilidad de Contenidos adicionales | Tienda de Contenidos | 20% | 10% |
| | Integración de Contenido | 14% | 16% |
| Agilidad en el desarrollo | Fluidez en el trabajo | 6% | 20% |
| TOTAL: | | 60% | 82% |

De la evaluación de los motores de juegos obtenida en la Tabla XIII, se aprecia que el resultado favorece con un 82% a Unreal Engine, por todas las prestaciones que se pudo evidenciar al momento de la construcción del prototipo, demostrando que el desarrollo de juegos con esta herramienta es mucho más sencillo, rápido y dinámico tanto en métricas cualitativas y cuantitativas aplicada en este estudio, la notable diferencia de esta evaluación se ve reflejada en la Figura 54 que se muestra a continuación:

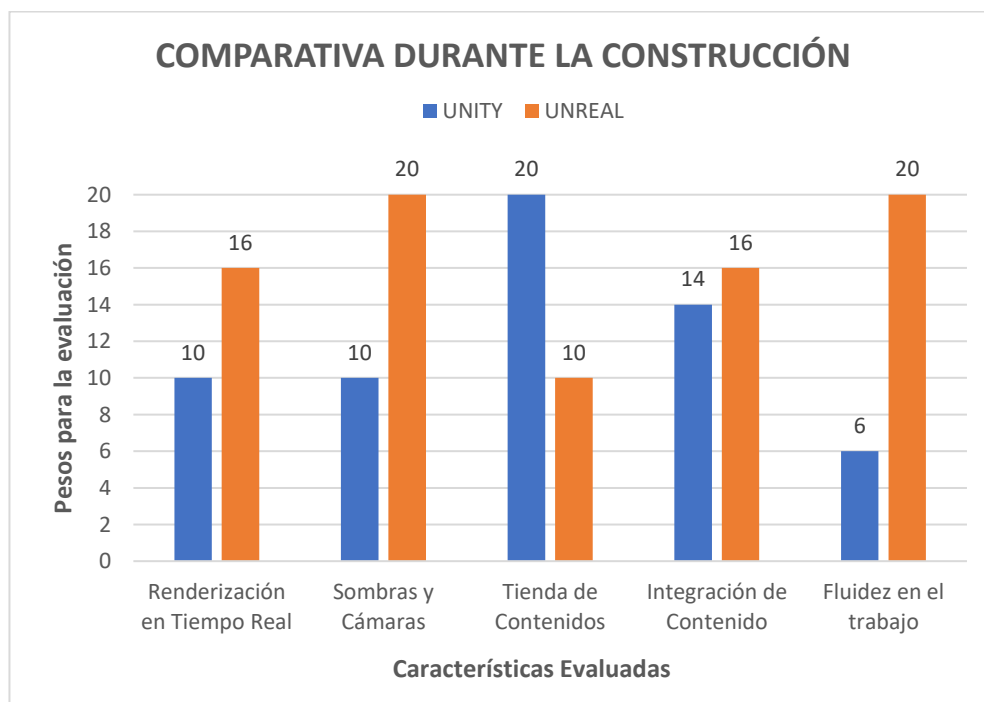


Figura 54. Resultados de la Evaluación durante la construcción.

3.2 Definir fortalezas y debilidades de cada motor de juego evaluado.

Luego del proceso de comparación de características básicas, componentes básicos y la experiencia obtenida en el desarrollo de juegos serios, se pudo proceder a la determinación de fortalezas y debilidades que cada motor de juego presenta:

TABLA XIV. DETERMINACIÓN DE FORTALEZAS Y DEBILIDADES DE UNITY

| UNITY 3D | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FORTALEZAS | DEBILIDADES |
| <ul style="list-style-type: none"> - La curva de aprendizaje es baja, debido a su facilidad de uso para alguien que sepa programar - Cantidad de documentación disponible tanto para la instalación como el desarrollo de videojuegos y sus particularidades como creación | <ul style="list-style-type: none"> - Mala gestión de la memoria - Multitud de parches - Menor calidad grafica - Exige mucho rendimiento al PC |

| | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| <p>de acciones, animaciones, texturizado, etc.</p> <ul style="list-style-type: none"> - Gran comunidad activa dispuesta a ayudarte tanto en inconvenientes propios del motor como los que se presentan en el desarrollo. - Rapidez al empezar a trabajar debido a la interfaz de usuario que posee. - Facilidad de integración de plugins de terceros. - Assets Store amplia y completa. - Acepta muchos formatos multimedia. | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|

TABLA XV. DETERMINACIÓN DE FORTALEZAS Y DEBILIDADES DE UNREAL ENGINE 4

| UNREAL ENGINE | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FORTALEZAS | DEBILIDADES |
| <ul style="list-style-type: none"> - Calidad Gráfica muy realista - El desarrollo no necesariamente puede iniciar desde 0, ya que posee plantillas de juegos prediseñadas. - Comunidad activa en español en redes sociales. - Activos gratis mensualmente desde la tienda virtual. - Sistema de scripting visual (Blueprint) que permite el desarrollo sin utilizar ninguna línea de código - Control completo del engine para adecuarlo a nuestras necesidades. | <ul style="list-style-type: none"> - Curva de aprendizaje un poco difícil cuando se programa en C++, debido a la complejidad del lenguaje. - Se necesita una maquina con altas prestaciones, RAM de 16 GB, y tarjeta de video dedicada. - Admite formatos FBX y OBJ - La opción del idioma en español trae muchos errores. - Falta de documentación en español |

| | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| <ul style="list-style-type: none">- Potencia en Iluminación, y creación de shaders y materiales.- Gran cantidad de Documentación disponible tanto para la instalación como el desarrollo.- Creación de terrenos y landscape fáciles e intuitivos. | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|

Al término de esta fase que es la culminación del presente proyecto y como desarrolladora principiante en lo referente a la creación de juegos serios, me permito recomendar el uso del motor Unreal Engine por la facilidad de manejo y el sistema de Scripting Visual hacen que el 'proceso de implementación de este tipo de aplicaciones, se vuelva más fácil e intuitiva.

G. Discusión

1. Desarrollo de la propuesta alternativa

Concluido el desarrollo del presente Trabajo de Titulación, se realiza una evaluación de los objetos planteados para determinar si se cumplió cada uno de ellos.

Objetivo específico 1: Analizar los motores de juegos y establecer los criterios de comparación.

Para realizar este objetivo con éxito, se usó el método de revisión sistemática de literatura (SLR) siguiendo las etapas planteadas por Bárbara Kitchenham [26]. En primer lugar, se determinó las preguntas de investigación, las mismas que sirvieron de pauta para obtener los estudios que fueran de utilidad. Posteriormente se seleccionó las fuentes bibliográficas y se elaboró las cadenas de búsqueda usando los operadores AND y OR. Se aplicó las cadenas de búsqueda en las diferentes bases científicas obteniendo un total de 158 estudios, de los cuales, aplicando los criterios de inclusión, exclusión y de selección se han escogido 27 artículos, con los cuales se procedió a determinar que Unity y Unreal Engine son los motores de juegos más usados en el desarrollo de juegos serios y que para la selección de los mismos se encontraron 14 criterios, clasificados en 2 categorías (Ver sección Resultados, fase 1), dando así por cumplido el objetivo planteado.

Objetivo Especifico 2: Desarrollar un prototipo de videojuego que sirva como escenario real para la evaluación.

En este objetivo se realizó la instalación de los motores de juegos seleccionados en el sistema operativo Windows, para posteriormente ir registrando los inconvenientes presentados durante este proceso, en donde se pudo evidenciar que para descargar e instalar Unity y Unreal, se deben seguir los mismos pasos: registrarse como desarrollador en las páginas oficiales de ambas empresas, instalar el lanzador o gestor de contenido de cada una de ellas, iniciar sesión con la cuenta antes mencionada, y desde ahí proceder con la instalación del motor en su versión más reciente, o si no tenemos la opción de elegir la versión que se ajuste a nuestros requerimientos (Ver

sección Resultados, fase 2, subsección 2.1), a la par de este proceso se fue registrando los inconvenientes que se presentaron al momento de instalar ambos motores, donde se evidencio que aunque la instalación de Unreal es sencilla toma un tiempo considerable de instalación aproximadamente entre 8-10 horas con una velocidad de internet de 3.47 Mbps de descarga, otro problema que se pudo evidenciar es la exigencia de recursos que necesita Unreal Engine causaba que otros programas colapsen al punto de no iniciar.

Posteriormente, se procedió a realizar el Documento de Diseño del juego a desarrollar que no es más que un documento con la síntesis de lo que va ser el videojuego donde consta de descripción, género, personajes, historia, mecánicas, arte, sonidos, entre otros (Ver sección Resultados, fase 2, subsección 2.3), y para termino de este objetivo se procede a llevar acabo el desarrollo del mismo en cada uno de los motores de juegos a evaluar, dando como resultado una aplicación con similares características pero con distinto proceso de desarrollo, en este documento se presentan puntos importantes que se realizaron durante el desarrollo del videojuego tales como: Creación del proyecto, creación del nivel del juego, Acciones del personaje, sistema de creación y destrucción de Plataformas, sistema de obstáculos, generación – recolección y destrucción de objetos, Generación de componentes GUI (Ver sección Resultados, Fase 2, subsección 2.4).

Objetivo Especifico 3: Evaluar los motores de juegos de acuerdo a los criterios de comparación establecidos

Para cumplir con objetivo se procedió a elaborar los cuadros comparativos tanto de las características básicas, componentes básicos determinados en la SLR, y la evaluación experimental obtenida durante la etapa de desarrollo del prototipo, con los cuales se pudo determinar las fortalezas y debilidades de cada motor de juegos (Ver sección Resultados, Fase 3), determinando así que Unreal es la mejor opción para un usuario principiante que desee iniciar en el desarrollo de videojuegos y que tenga escasos o nulos conocimientos en lenguajes programación.

2. Valoración técnica económica ambiental

Para el desarrollo del presente trabajo de titulación se contó con la colaboración de recursos humanos, materiales, técnicos y tecnológicos.

La Tabla XVI hace referencia a los recursos humanos empleados para realizar el desarrollo del proyecto, cabe recalcar que además se contó con el continuo asesoramiento del director del trabajo de titulación, lo cual permitió que se lleve a cabo un correcto desarrollo del proyecto.

TABLA XVI. VALORACIÓN ECONÓMICA DE RECURSOS HUMANO

| Roles | Números de Horas | Valor por hora (\$) | Valor Total (\$) |
|---------------------------------------------------|-------------------------|----------------------------|-------------------------|
| Tesista: Jennifer Tomalá González | 400 | 8.00 | 3200.00 |
| Director: José Oswaldo Guamán Quinche | 100 | - | - |
| Asesor Académico: Oscar Miguel Cumbicus Pineda | 240 | - | - |
| Subtotal: | | | 3200.00 |

La Tabla XVII presenta los recursos materiales y servicios que fueron utilizados para poder desarrollar el trabajo de titulación.

TABLA XVII. VALORACIÓN ECONÓMICA DE MATERIALES Y SERVICIOS

| Descripción | Cantidad | Valor Unitario (\$) | Valor Total (\$) |
|--------------------|-----------------|----------------------------|-------------------------|
| Resma de papel A4 | 2 | 5.00 | 10.00 |
| Copias | 400 | 0.02 | 8.00 |
| CD | 3 | 3.00 | 9.00 |
| Anillados | 4 | 5.00 | 20.00 |
| Tinta de Impresión | 2 | 27,50 | 55,00 |
| Suministros varios | - | 20.00 | 20.00 |
| Subtotal: | | | 122.00 |

La Tabla XVIII presenta los recursos técnicos y tecnológicos que fueron utilizados para dar cumplimiento a los objetivos planteados en el desarrollo del TT. Estos son de vital importancia debido que permitieron cumplir con el desarrollo de todo el proceso del proyecto.

TABLA XVIII. VALORACIÓN ECONÓMICA DE RECURSOS TECNICOS Y TECNOLOGICOS

| Recurso | Cantidad | Valor Unitario | Valor Total |
|------------------------------|----------|----------------|------------------|
| RECURSO HARDWARE | | | |
| Laptop (Depreciación) | 1 | \$ 224.98 | \$ 224.98 |
| RECURSOS SOFTWARE | | | |
| Unity 5 | 1 | Gratuito | 0,00 |
| Unreal Engine 4 | 1 | Gratuito | 0,00 |
| RECURSOS DE TELECOMUNICACIÓN | | | |
| Internet | 6 meses | 20.00 | 120,00 |
| Subtotal | | | \$ 344,98 |

Finalmente, en la tabla XIX se presenta los costos totales de cada uno de los recursos anteriormente detallados, con el fin de presentar el costo total del trabajo de titulación.

TABLA XIX. COSTO TOTAL DEL TRABAJO DE TITULACIÓN

| RECURSOS | SUBTOTALES |
|----------------------------------|----------------|
| Talento Humano | 3200,00 |
| Materiales y Servicios | 122,00 |
| Recursos Técnicos y Tecnológicos | 344,98 |
| Imprevistos 10% | 446,70 |
| TOTAL | 4113,68 |

H. Conclusiones

Al finalizar el presente Trabajo de se pudo cumplir de manera exitosa los objetivos propuestos inicialmente, y en este proceso de aprendizaje teórico y práctico se puede concluir con lo siguiente:

- En base a la Revisión Sistemática de Literatura, se pudo evidenciar que los motores de juegos más utilizados actualmente son Unity y Unreal Engine, ambos ofrecen licencias de usuario final (EULA en sus siglas en inglés), tiendas de activos gratuitos y pagados, amplia documentación en inglés y grandes comunidades de desarrolladores que brindan soporte y responden a las más pequeñas inquietudes.
- Dos características fundamentales que en general los desarrolladores consideran para la elección del motor de juegos se encuentran los atributos de ser Multiplataforma (adaptable a las nuevas tecnologías) y el componente renderización (calidad de gráficos).
- La rapidez para trabajar que nos ofrece Unreal es debido a dos características sobresalientes, el primero el sistema de Scripting visual (Blueprint) que prácticamente lo realiza todo visualmente a través de arrastrar y soltar nodos y luego interconectarlos entre sí, y la segunda característica son las plantillas predeterminadas que este trae consigo que nos ayuda a no iniciar desde cero en el desarrollo
- Unreal trae los shaders y efectos de cámara activados de manera determinada lo que permite ahorrar tiempo si desarrolla un escenario realista, facilitando así el trabajo del desarrollador, lo que no sucede con Unity donde se necesita activar este tipo de funciones mediante scripts programables
- Dentro de la comparativa realizada tanto Unity como Unreal Engine ofrecen una amplia gama de herramientas, su interfaz es intuitiva y de fácil uso; pero la mayor ventaja de Unreal Engine es la creación de gráficos de primera calidad, es de código abierto, dando la posibilidad que los desarrolladores puedan hacer los cambios necesarios que se ajusten a los requerimientos.
- Un juego serio es un videojuego que tiene un enfoque que va más allá de la pura diversión, este puede servir para alcanzar objetivos en educación, salud, marketing o cambio social.

- El crear un Juego serio no solo depende del hecho de tener una idea para crear un videojuego, se necesita además determinar las pautas correctas que se desean transmitir al jugador, siendo estas incluidas en el propósito del videojuego.
- Elegir el motor correcto va más allá de la evaluación de las características de cada uno, realmente depende del tipo de juego que desee crear, la audiencia, pero sobre todo la predisposición por aprender.
- Los motores de videojuegos son herramientas muy útiles que facilitan el trabajo de los creadores permitiéndoles centrarse en los aspectos creativos y no tanto en los aspectos estructurales.
- En el desarrollo de videojuegos se requiere a más de la preparación técnica un alto grado de imaginación y creatividad.

I. Recomendaciones

Luego de haber realizado el presente trabajo de titulación se puede recomendar lo siguiente:

- Aprovechar la gran acogida que tienen los juegos serios y videojuegos en general para concientizar a temprana edad sobre los problemas sociales que atraviesa la sociedad actual como el daño al medio ambiente, racismo, migración, bullying, accidentes de tránsito, entre otros temas.
- Hacer una adecuada investigación acerca del tema que se desea transmitir mediante el videojuego serio, con el fin de que este pueda ser incorporado adecuadamente en los lineamientos de este.
- Dar el tiempo pertinente al aprendizaje, manejo y control del motor de juego elegido, para evitar realizar un doble esfuerzo como el de aprender y desarrollar al mismo tiempo.
- Al momento de iniciar la creación del prototipo de juego serio, es recomendable la utilización de un Documento de Diseño de Juegos (GDD en sus siglas en ingles), el cual debe contener como mínimo: concepto, genero, propósito y público objetivo, elementos del juego, mecánicas, bocetos iniciales y arte gráfico, para guiarnos durante el proceso de desarrollo.
- Poseer un buen equipo computacional con prestaciones tales como: 16GB de RAM y una tarjeta gráfica dedicada (podría ser NVIDIA GeForce GTX 680) al momento de desarrollar videojuegos. Cabe acotar que trabajar con un equipo que posee requerimientos mínimos no es mala opción, pero la fluidez no será la misma.
- Al momento de desarrollar un juego serio, videojuego u otro producto de Software se debe considerar el Modelo de Calidad ISO/IEC 25010 para evaluar los criterios que este debería contener.
- Investigar constantemente las actualizaciones de los Motores de juegos expuestos en este Trabajo de Titulación, así como también de la aparición de nuevos motores; con el fin de aprovechar al máximo las prestaciones que este tipo de herramientas ofrecen.
- Utilizar el motor de juegos Unreal Engine si eres un diseñador o principiante en el desarrollo de videojuegos.

- Que dentro de la malla curricular se incluya una unidad orientada al desarrollo de videojuegos.
- Considerar plataformas virtuales tales como Udemy y Coursera, que brindan varios cursos gratuitos para principiantes en el desarrollo de videojuegos.
- A partir de la culminación del presente proyecto se pueden realizar los siguientes Trabajos Futuros:
 - Realizar un estudio al Mecanismo de Scripting Visual (Blueprint) que implementa Unreal Engine.
 - Desarrollar un videojuego multijugador para fomentar el trabajo en equipo utilizando el motor de juegos Unreal Engine.
 - Realizar un estudio comparativo de motores de juegos en base al desarrollo de juegos con realidad virtual y sensores de movimiento.

J. Bibliografía

- [1] J. F. C. Lobo, "Juegos Serios: Alternativa Innovadora," in *Conocimiento Libre y Educación (CLED)*, 2014, vol. 2, no. 2.
- [2] D. G. Vilches Antão, "Juegos serios, evaluación de tecnologías y ámbitos de aplicación," Universidad Nacional de La Plata, 2014.
- [3] F. Evans, S. E. Mileta, and J. D. Petrillo, "Proceso de selección de tecnología para el desarrollo de serious games," 2016.
- [4] E. A. Figueredo, Oscar Boude; Sosa, "Juego serio : modelo teórico para su diseño y producción," no. February, 2016.
- [5] J. V. Altamirano Martínez, "¿Esto es un juego? Juegos serios y gamificación, taxonomía y aplicación.," Feb. 2017.
- [6] A. Samčović, "Serious games in military applications," *Vojnoteh. Glas.*, vol. 66, no. 3, pp. 597–613, 2018.
- [7] B. Marcano, "Juegos Serios y Entrenamiento en la Sociedad Digital," *Teoría la Educ. Educ. y Cult. en la Soc. la Inf.*, vol. 9, no. 1138–9737, p. 16, 2008.
- [8] T. I. M. Tutenel, R. Bidarra, R. M. Smelik, and K. J. A. N. D. E. Kraker, "The Role of Semantics in Games and Simulations," vol. 6, no. 4, pp. 1–35, 2008.
- [9] M. T. Gómez del Castillo Segurado, "Videojuegos y transmisión de valores," 2007.
- [10] Naciones Unidas, "La agencia de ayuda alimentaria de la ONU ayuda a crear un juego en línea para combatir el hambre," 2018.
- [11] M. Lee, Y. Choi, E. Taylor, and R. T. Cole, "Playing With Food : Content Analysis of Food Advergimes," vol. 43, no. 1, pp. 129–154, 2009.
- [12] R. Oliveira, "Physical Rehabilitation based on Kinect Serious," *2017 Elev. Int. Conf. Sens. Technol.*, pp. 0–5, 2017.
- [13] T. Martins, V. Carvalho, and F. Soares, "A Serious Game for Rehabilitation of Neurological Disabilities : Premilinary Study," *2015 IEEE IV Encuentro Port. Bioingeniería*, pp. 2–6, 2015.
- [14] E. M. C. R. O. Moreno, Francisco; Ojeda, Jordan; Ramirez, "Un Framework para

- la Rehabilitación Física en Miembros Superiores con Realidad Virtual,” *Prim. Conf. Nac. Comput. Informática y Sist. (CoNCISa 2013)*, Club Puerto Azul, Naiguatá, Venez., pp. 77–84, 2013.
- [15] P. Khanal, D. Katz, S. DeMaria, M. Krol, and K. Kahol, “Design and development of a serious game for central line placement,” *Proc. CBMS 2013 - 26th IEEE Int. Symp. Comput. Med. Syst.*, pp. 530–531, 2013.
- [16] A. M^a Manrubia Pereira, “El proceso productivo del videojuego: fases de producción,” *Hist. y Comun. Soc.*, vol. 19, pp. 791–805, 2014.
- [17] S. Pavkov, I. Franković, and N. Hoić-Božić, “Comparison of game engines for serious games,” *2017 40th Int. Conv. Inf. Commun. Technol. Electron. Microelectron. MIPRO 2017 - Proc.*, pp. 728–733, 2017.
- [18] C. Vallejo Fernández, David; Martín Angelina, *Desarrollo de Videojuegos: Arquitectura del Motor*, Segunda. Ciudad Real, 2013.
- [19] Y. C. Buleje Castro, A. Tarqui Apaza, E. E. Vargas Carpio, M. Vargas Valverde, H. D. Calderón-Vilca, and I. Solis Soaria, “Sistema Clúster De Alto Rendimiento Para La Renderización De Escenas Tridimensionales Fotorealistas Con Trazado De Rayos Abancay 2015 High Performance Cluster System for the Rendering of Three-Photoreal Photorealist Scenes With Ray Tracing, Abancay 2015,” *Rev. Investig. La Esc. Posgrado Univ. Nac. Del Altiplano Puno - Perú*, vol. 6, no. 2, p. 2017, 2017.
- [20] A. Luciano, J. Nomdedeu, D. J. F. Díaz, and L. A. Fava, “RAICES: un juego serio social para revalorizar las culturas originarias,” Universidad Nacional de la Plata, 2015.
- [21] Epic Games, “Unreal Engine 4 Documentation,” 2017. .
- [22] G. Tirado Granados, “Introducción a OGRE 3D,” *Proy. innovación Educ. PIE07-084 subvencionado por el Serv. innovación Educ. y el Serv. Enseñanza Virtual y Lab. Tecnológicos la Univ. Málaga.*, p. 73.
- [23] J. A. Ortega, R. E. Sánchez, J. J. González, and G. Reyes, “Virtual Laboratories for Training in Industrial Robotics,” vol. 14, no. 2, pp. 665–672, 2016.
- [24] A. Pérez Clemente, “Desarrollo de un juego basado en red mediante Unity,” Escuela Técnica Superior de Ingeniería y Sistemas de Telecomunicación., 2018.

- [25] M. Chavarro, Christian; Valencia, "COMPARACIÓN DE MOTORES DE VIDEOJUEGOS PARA LA CREACIÓN DE JUEGOS SERIOS," UNIVERSIDAD TECNOLÓGICA DE PEREIRA PROGRAMA, 2015.
- [26] B. Kitchenham *et al.*, "Systematic literature reviews in software engineering-A tertiary study," *Inf. Softw. Technol.*, vol. 52, no. 8, pp. 792–805, 2010.

K. Anexos

ANEXO 1: Revisión Sistemática de literatura (SLR) sobre Metodologías y Motores de Juegos para el desarrollo de Juegos serios.

Systematic review of literature (SLR) on methodologies and game engines for the development of serious games.

José Guamán - Quinche, Jennifer Tomalá – González

Universidad Nacional de Loja, Carrera de Ingeniería en Sistemas, Av. Pío Jaramillo
Alvarado, La Argelia, Loja, Ecuador

[jose.o. guaman, jctomalag] @unl.edu.ec

Resumen: Los Juegos Serios son videojuegos que se caracterizan por estar diseñados esencialmente con fines pedagógicos (educar, concienciar, entrenar e informar), el desarrollo de un videojuego conlleva un largo y complejo proceso; para la simplificación del mismo se han creado los Games Engines o Motores de Juegos. El presente artículo especifica el proceso de revisión sistemática de literatura (SLR) sobre estudios primarios referentes al uso de motores de juegos (Game Engine) y metodologías en el desarrollo de juegos serios, siguiendo el protocolo de Barbara kitchenham[1], con el fin de identificar qué metodologías y motores de juegos son los más utilizados actualmente en este tipo de desarrollo, y cuáles son los criterios que se consideran para su elección. Para lo cual se consideró el uso de tres bibliotecas digitales como: IEEEExplorer, ACM Digital Library y Google Scholar.

Palabras claves: Juegos Serios, Videojuegos Educativos, Motores de juegos

1. Introducción:

Los videojuegos son uno de los productos de ocio y entretenimiento de mayor alcance en los últimos años [2], pero también se consideran poderosas herramientas como estrategia metodológica[3], estos videojuegos de índole educativo, también conocidos

como Serious Games (SG), son videojuegos diseñados con un propósito principal que no es el entretenimiento puro; su principal objetivo puede ser enseñar, cambiar una actitud, comportamiento o crear conciencia de un determinado tema[4].

El desarrollo de este campo es muy poco explotado, por la complejidad de ejecución, o por el gran uso de recursos que se necesita, en la actualidad existen herramientas que minimizan tanto esfuerzos como recurso, a estas herramienta se las denomina Motores de Juegos, la presente revisión tiene como objetivo principal el mostrar cuales son las metodologías y los motores de juegos utilizados actualmente en el desarrollo de Juegos Serios, así como los criterios que se consideran para la selección del motor de juego, mediante la sintetización del actual estado del arte.

2. Metodología

En el presente artículo se aplica la metodología propuesta por Bárbara Kitchenham[1], la cual parte del estudio de las pruebas disponibles sobre una determinada intervención, con el objeto de responder a cuestiones concretas, siguiendo una metodología explícita y rigurosa, descrita en la Tabla 1, donde se puede visualizar el esquema a seguir.

Tabla 1. Proceso de Revisión Sistemática de Literatura

| Fase | Paso |
|-----------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. Planificación de la Revisión Sistemática de Literatura | <ul style="list-style-type: none"> a. Objetivo de la Revisión Sistemática de Literatura b. Formulación de la pregunta de investigación c. Palabras Claves d. Selección de fuentes y estrategias de búsqueda e. Cadena de búsqueda f. Criterios de inclusión g. Criterios de exclusión h. Lista de Evaluación de la Calidad |
| 2. Ejecución de la Revisión Sistemática de literatura | <ul style="list-style-type: none"> a. Búsqueda de Estudios b. Selección de Estudios primarios c. Formato de Extracción de la información |
| 3. Análisis de resultados y hallazgos | <ul style="list-style-type: none"> a. Discusión |

3. Resultados

3.1. Planificación de la revisión literaria

En esta etapa se identifica cuáles son las necesidades de la revisión, para lo cual es necesario plantear y responder algunos puntos claves descritos a continuación.

a. Objetivo de la Revisión Sistemática de Literatura

Estos objetivos se cumplen respondiendo y verificando la pregunta de investigación.

- Realizar un estudio de Metodologías, Motores de Juegos y criterios de elección dentro del desarrollo de Juegos Serios.

b. Formulación de Pregunta de Investigación

A partir de la temática de “Revisión de literatura sobre Metodologías y Motores de Juegos para el desarrollo de Juegos serios” se plantea las siguientes preguntas de investigación referente al tema, las cuales están descritas en la Tabla 2, donde RQ1 es la Primera pregunta, RQ2 la segunda pregunta y RQ3 la tercera pregunta respectivamente.

Tabla 2. Preguntas de investigación

| Preguntas |
|--------------------------------------------------------------------------------------------------------------|
| RQ1. ¿Qué metodologías se están utilizando en el desarrollo de Juegos Serios y cuáles son sus etapas? |
| RQ2. ¿Cuáles son los motores de juegos actualmente utilizados en el desarrollo de Juegos Serios? |
| RQ3. ¿Cuáles son los criterios para la selección del motor de Juegos? |

Estas preguntas se pueden clasificar en dos áreas de interés. Por un lado, la pregunta RQ1 estudia lo concerniente a las metodologías de desarrollo utilizadas en la creación de videojuegos serios o serious games, sus características y las etapas de cada una de ellas.

Por otro lado, RQ2 y RQ3 se centran en conocer qué motores de juegos actualmente están siendo utilizados al momento de implementar juegos serios, así como los criterios por los cuales estos son elegidos entre una extensa gama de motores existentes.

c. Palabras claves

La cadena de búsqueda se obtiene de la revisión de artículos que permitan identificar las palabras claves; se describen en la Tabla 3:

Tabla 3. Artículos Preliminares y Palabras Claves

| Número | Estudio | Palabra Clave |
|--------|------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------|
| E1 | A Survey of Frameworks and Game Engines for Serious Game Development[5] | serious games; game engine; framework; review; |
| E2 | Serious Games, Gamification and Game Engines to Support Framework Activities in Engineering: Case Studies, Analysis, Classifications and Outcomes[6] | serious games; gamification; serious game engines; engineering. |
| E3 | A Serious Game for Interview Preparation[7] | Serious games; job interview; simulation; virtual environment; scenario scripting. |
| E4 | EngAGe: A link between Educational Games Developers and Educators[8] | Assessment; Assessment Engine; Learning Analytics; Serious Games; Games-Based Learning; Feedback; Adaptivity |
| E5 | Towards the Development of an Interactive 3D Coach Training Serious Game[9] | serious game, game engine; game-based learning; coach training simulator |
| E6 | A serious game development process using competency approach[10] | Game development process; serious games; software engineering; development process design. |
| E7 | Serious games and the gamification of tourism[11] | Gamification; Serious games; Engagement; Loyalty; Tourism experiences |
| E8 | A systematic literature review on serious games evaluation: An application to software project management[12] | Serious game; Evaluation; Systematic literature review; Software project management |

d. Selección de Fuentes y Estrategia de Búsqueda

- **Selección de Fuentes**

Para la realización de la revisión se eligió las siguientes bases de datos digitales que se muestran en la Tabla 4. en la cual se puede observar el nombre de la base científica y la URL de la misma:

Tabla 4. Bases de Datos Digitales a utilizar

| Nombre | URL |
|-------------------------|---------------------------------------------------------------------|
| Biblioteca digital IEEE | http://ieeexplore.ieee.org |
| Google Académico | https://scholar.google.es/ |
| ACM Digital Library | https://dl.acm.org/dl.cfm |

- **Estrategia de Búsqueda**

El objetivo de la estrategia de búsqueda es identificar los estudios primarios, que darán respuestas a las preguntas de investigación propuestas. Estos estudios deberán tener relación con una cadena de búsqueda, el mismo que estará compuesta por palabras claves en común de la sección c, y que concuerdan con los términos PICOC.

Determinación de términos PICOC

El Modelo PICOC (Población, Intervención, Comparación, Resultado, Contexto) ayuda en el proceso de análisis de los artículos para responder a las preguntas de investigación, ya que determinan los términos principales que se utilizarán como base para definición de las palabras claves; sólo se usará tres criterios como se muestra en la Tabla 5, pues sólo se trabajará con un grupo específico (Desarrollo de Juegos serios o Videojuegos Educativos), las metodologías y los motores de juegos que se aplican en este grupo específico.

Tabla 5. Términos PICOC

| | |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| P: Población | Desarrollo de Juegos Serios, Desarrollo de Videojuegos Educativos, Serious Games Development, Development of Educational Video Games |
| I: Intervención | Motores de Juegos, Games Engine |
| C: Contexto | Desarrollo de Software |

Definición de palabras claves

Para la determinación de palabras claves se realizó una búsqueda preliminar de estudios relacionados con el tema de interés, y de los documentos encontrados se extrajo las palabras principales en común y se realizó una comparación con los términos PICOC, como se muestra en la Tabla 6:

Tabla 6. Palabras Claves

| Español | | Inglés |
|-----------------------------|--------------------------------------|-------------------------------------------------------------------|
| Palabras Claves | Sinónimos | |
| Desarrollo de Juegos Serios | Desarrollo de Videojuegos Educativos | Serious Games Development, Development of Educational Video Games |
| Motor de Juegos | | Games Engine |

e. Cadena de Búsqueda

Para la determinación de la cadena de búsqueda se realiza la unión de las palabras claves mediante los operadores booleanos AND para vincular términos principales y OR para agregar ortografías alternativas y sinónimos, teniendo como resultado la siguiente cadena:

Tabla 7. Cadena de Búsqueda

| Cadena |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ("Desarrollo de Juegos Serios" OR "Desarrollo de Videojuegos Educativos" OR "Development of Serious Games" OR "Development of Education games") AND ("Motores de Juegos" OR "Games Engines") |

f. Criterios de Inclusión

En la búsqueda de artículos se seleccionará a aquellos que cumplan los criterios de inclusión, los mismos que se encuentran en la Tabla 8 del presente documento.

Tabla 8. Criterios de Inclusión

| Criterio | Valor |
|-----------------------------|----------------------------------------------------------------------------|
| Idioma | Inglés, español. |
| Motores de búsqueda | IEEEXplorer, Google Scholar y ACM Digital Library. |
| Fecha de publicación | 2013 – 2018. |
| Enfoque | Área de Computación |
| Tipo de producciones | Artículos científicos y paper de conferencias. |
| Duplicados | Si existieran duplicados, se incluirá la versión más completa del estudio. |

g. Criterios de Exclusión

Los estudios serán descartados si cumplen por lo menos uno de los criterios de exclusión que se presentan en la Tabla 9, los mismos que se muestra a continuación:

Tabla 9. Criterios de Exclusión

| Criterio | Valor |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Idioma | Diferentes al inglés, español. |
| Fecha de publicación | Estudios realizados antes del 2013. |
| Fecha de publicación | Menores al 2013. |
| Enfoque | Diferentes al Área de Computación |
| Tipo de producciones | Tesis, Folletos, Instructivos, Cursos, Talleres, etc. |
| Duplicados | Él mismo documento recuperado de diferentes bases de datos bibliográficas |
| Accesibilidad | Artículos no accesibles al contenido completo |
| Otros | Se excluirán aquellos estudios que, a pesar de contener los términos de búsqueda o combinación de ellos, no contienen información relevante sobre el tema y/o no abordan los tópicos de interés. |

h. Lista de Evaluación de la Calidad

Para determinar la lista de evaluación de la calidad que será utilizada en el desarrollo del presente trabajo se procedió a realizar un cuestionario que se encuentra en la Tabla 10, el mismo que debía aplicarse para cada documento que haya pasado el sesgo de los criterios de exclusión.

Este cuestionario consta de 5 preguntas de evaluación de calidad (QA) y se dividen en dos: Las primeras tres preguntas sirven para identificar la relevancia del documento dentro de la SLR (QA1 - QA3), la segunda parte contrasta la calidad del documento (QA4 - QA5). De acuerdo a la información anterior, las cinco preguntas de evaluación de calidad (QA) utilizadas se muestran en la tabla siguiente:

Tabla 10. Preguntas para Evaluar la Calidad de Estudios

| Cuestionario |
|-----------------------------------------------------------------------------------------------------------------------------------|
| QA1. ¿La investigación trata sobre el desarrollo de Juegos Serios o Videojuegos Educativos? |
| QA2. ¿Se utilizó motor(es) de juegos en el desarrollo de la investigación? |
| QA3. ¿Se utilizó alguna metodología para el desarrollo de videojuegos o Juegos Serios en el contenido de la investigación? |
| QA4. ¿Este artículo posee 10 o más fuentes bibliográficas confiables? |
| QA5. ¿Este artículo ha sido citado en 3 o más investigaciones? |

Cada pregunta será respondida SI o NO, y serán puntuadas con el valor de 1 para las respuestas afirmativas y 0 para Negativos, estableciendo una base mínima de 3 puntos para determinar la validez del artículo.

3.2. Ejecución de la Revisión Sistemática de literatura

Se especifica a continuación los criterios de selección de estudios, extracción de la información y las cadenas de búsquedas aplicadas en cada exploración avanzada dentro de la base de datos científicas.

La revisión sistemática se inició con la búsqueda de los artículos en las bases de datos digitales establecidas anteriormente, se encontró un total de 159 artículos distribuido de

la siguiente manera, 88 artículos de IEEEXplorer, 32 de ACM y 39 Google Scholar, para una mejor comprensión, observar Figura 1:

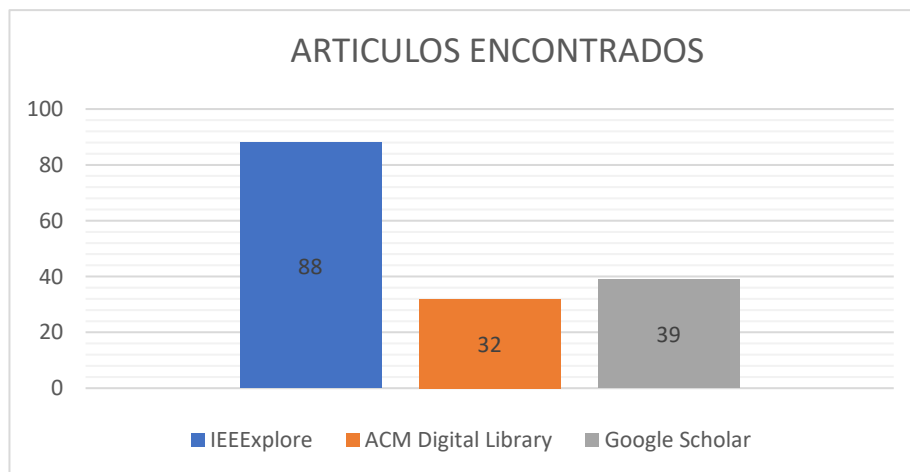


Fig. 1: Estudios encontrados en las diferentes fuentes digitales

a. Búsqueda de Estudios

Antes de obtener los estudios primarios que se utilizaran en la investigación, procedemos a realizar una preselección de artículos en cada una de las fuentes digitales establecidas.

- **Preselección de los Estudios en la fuente IEEE**

Para la búsqueda de los Estudios en la fuente Bibliográfica IEEE se procedió adaptar la cadena de búsqueda original, la misma que se encuentra en la Tabla 11 que se muestra a continuación:

Tabla 11. Cadena de Búsqueda aplicada en IEEE

| Cadena |
|-----------------------------------------------------------------------------------------|
| ((Development Educational Video Games) OR Serious Games Development) AND Games engines) |

La ejecución de la búsqueda en IEEE arrojó un total de 88 artículos encontrados, de los cuales, al aplicar los criterios de Exclusión se descartaron los siguientes:

- Año de publicación: 38
- Tipo de documento: 2 cursos
- Idioma: 1 artículo en portugués

- Enfoque: 5 artículos de áreas diferentes a computación y;
- 11 artículos que, teniendo los términos de búsqueda, no abordan tópicos de interés.

La cantidad en total de artículos descartados serian 57, por lo tanto, tenemos 31 artículos pre-seleccionados (AP) que tendrán que ser evaluados según la lista establecida para la evaluación de la calidad. Estos artículos se presentan en la tabla continuación:

Tabla 12. Artículos Pre-seleccionados (AP) de la fuente IEEE

| Cód. | Título | Número de Citas | Número de Referencias |
|-------------|--------------------------------------------------------------------------------------------------------------------------------|------------------------|------------------------------|
| AP01 | A framework for game engine selection for gamification and serious games[13] | 1 | 15 |
| AP02 | A serious game for interview preparation[7] | 0 | 2 |
| AP03 | A serious game with virtual reality for travel training with Autism Spectrum Disorder[14] | 4 | 4 |
| AP04 | A survey of frameworks and game engines for serious game development[5] | 6 | 10 |
| AP05 | A toolkit for presenting advanced mathematics in serious games[15] | 0 | 10 |
| AP06 | A virtual reality based serious games for rehabilitation of arm[16] | 2 | 17 |
| AP07 | Choosing the infrastructure for entertainment and serious computer games - A whiteroom benchmark for game engine selection[17] | 6 | 52 |
| AP08 | Comparison of game engines for serious games[18] | 1 | 21 |
| AP09 | Creation of custom-made serious games with user-generated learning content[19] | 0 | 14 |
| AP10 | Design and development of a serious game for central line placement[20] | 0 | 5 |
| AP11 | Hugo Against Dengue: A serious game to educate people about Dengue fever prevention[21] | 4 | 16 |
| AP12 | Hybrid peer to peer and server client system for limited user's multiplayer first person style games[22] | 0 | 24 |

| | | | |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------|---|----|
| AP13 | Intelligent Game Engine for Rehabilitation (IGER)[23] | 8 | 38 |
| AP14 | Multimodal reconstruction of landscape in serious games for heritage: An insight on the creation of Fort Ross Virtual Warehouse serious game[24] | 6 | 22 |
| AP15 | Physical Rehabilitation based on Kinect Serious games[25] | 3 | 15 |
| AP16 | Practices and Technologies in Computer Game Software Engineering[26] | 3 | 7 |
| AP17 | Seeker: A serious game for improving cognitive abilities[27] | 0 | 15 |
| AP18 | Semi-autonomous digitization of real-world environments[28] | 1 | 26 |
| AP19 | Serious 3D gaming research for the vision impaired[29] | 0 | 22 |
| AP20 | Serious game engines: Analysis and applications[30] | 1 | 31 |
| AP21 | Serious games for the Elicitation of way-finding behaviours in emergency situations[31] | 2 | 25 |
| AP22 | Serious games, gamification and game engines to support framework activities in engineering: Case studies, analysis, classifications and outcomes[6] | 7 | 48 |
| AP23 | Serious games-creating an ecosystem for success[32] | 0 | 27 |
| AP24 | Serious mods: A case for modding in serious games pedagogy[33] | 0 | 16 |
| AP25 | SolarSystemGO - An augmented reality-based game with astronomical concepts[34] | 0 | 8 |
| AP26 | Towards a new concept of serious games generator[35] | 0 | 8 |
| AP27 | Towards a service-oriented architecture framework for educational serious games[36] | 7 | 12 |
| AP28 | UAdventure: The eAdventure reboot: Combining the experience of commercial gaming tools and tailored educational tools[37] | 1 | 24 |
| AP29 | Using game engines for marine visualisation and collaboration[38] | 0 | 7 |
| AP30 | Using Serious Games to Create Awareness on Visual Impairments[39] | 0 | 16 |

| | | | |
|------|------------------------------------------------------------------------------------------|---|----|
| AP31 | Wanda: A Framework to Develop Card Based Games to Help Motivate Programming Students[40] | 1 | 16 |
|------|------------------------------------------------------------------------------------------|---|----|

- **Preselección de los Estudios en la fuente ACM**

Para la búsqueda de los Estudios en la fuente Bibliográfica ACM se procedió adaptar la cadena de búsqueda original, obteniendo como resultado:

Tabla 13. Cadena de Búsqueda aplicada en la fuente ACM

| Cadena |
|-----------------------------------------------------------------------------------------------------------------|
| acmdlTitle:(+"Development" "Serious Games" "Educational Videogames") AND recordAbstract:(+"Game Engine") |

La ejecución de la búsqueda en ACM arrojó un total de 32 artículos encontrados, de los cuales, al aplicar los criterios de Exclusión se descartaron los siguientes:

- Año de publicación: 11
- Artículos no accesibles: 2
- 6 artículos que, teniendo los términos de búsqueda, no abordan tópicos de interés.

La cantidad en total de artículos descartados serían 19, por lo tanto, tenemos 13 artículos que tendrán que ser evaluados según la lista establecida para la evaluación de calidad. Estos artículos se presentan en la tabla a continuación:

Tabla 14. Artículos Pre-Seleccionados (AP) de la fuente ACM

| Cód. | Título | Número de Citas | Número de Referencias |
|------|-----------------------------------------------------------------------------------------------------|-----------------|-----------------------|
| AP32 | 3D Game Development with Unity in the Computer Science Curriculum[41] | 1 | 5 |
| AP33 | A Framework for Developing Multi-Agent Systems in Ambient Intelligence Scenarios[42] | 0 | 5 |
| AP34 | An Experience-based Comparison of Unity and Unreal for a Stand-alone 3D Game Development Course[43] | 1 | 27 |

| | | | |
|------|----------------------------------------------------------------------------------------------------------------------|---|----|
| AP35 | CapStudio: An Interactive Screencast for Visual Application Development[44] | 1 | 13 |
| AP36 | Developing a Highly Automated Driving Scenario to Investigate User Intervention: When Things Go Wrong[45] | 0 | 9 |
| AP37 | Development of a Tsunami Evacuation Behavior Simulation System with Massive Evacuation Agents[46] | 0 | 4 |
| AP38 | Development of Encantasya: War of the Four Kingdoms[47] | 0 | 8 |
| AP39 | Development of Laparoscopic Cholecystectomy Simulator Based on Unity Game Engine[48] | 0 | 32 |
| AP40 | Development of Virtual Reality Based Robotic Surgical Trainer for Patient-specific Deformable Anatomy[49] | 1 | 21 |
| AP41 | Engine- Cooperative Game Modeling (ECGM): Bridge Model-Driven Game Development and Game Engine Tool-chains[50] | 0 | 35 |
| AP42 | HCore: A Game Engine Independent OO Architecture for Fast Development of Haptic Simulators for Teaching/Learning[51] | 0 | 21 |
| AP43 | Tracing the Development of a Haptically-enhanced Simulation for Teaching Phase Change[52] | 0 | 17 |
| AP44 | Using Unity to Teach Game Development: When You've Never Written a Game[53] | 5 | 22 |

- **Preselección de los estudios en la Fuente Google Scholar**

Para la búsqueda de los Estudios en la fuente Bibliográfica Google Scholar se procedió adaptar la cadena de búsqueda original, obteniendo como resultado:

Tabla 15. Cadena de Búsqueda aplicada en la fuente Google Scholar

| Cadena |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ("Desarrollo de Juegos Serios" OR "Desarrollo de Videojuegos Educativos" OR "Development Educational Video Games" OR "Serious Games Development") AND ("Games engines" OR "Motores de Juegos") |

La ejecución de la cadena de búsqueda en Google Scholar arrojó un total de 39 coincidencias, de los cuales, al aplicar los criterios de Exclusión se descartaron los siguientes:

- Año de publicación: 8
- Artículos no accesibles: 3
- Tipo de documento: 15 conformados por 1 libro, 13 tesis y 1 tesina.
- 3 artículo duplicado
- Enfoque: 1 artículo de área diferente al de computación y;
- 3 artículo que, teniendo los términos de búsqueda, no abordan tópicos de interés.

La cantidad en total de artículos descartados serían 33, por lo tanto, tenemos 6 artículos que tendrán que ser evaluados según la lista establecida para la evaluación de calidad. Estos artículos se presentan en la tabla a continuación:

Tabla 16. Artículos Pre-Seleccionados (AP) de la fuente Google Scholar

| Cód. | Título | Número de Citas | Número de Referencias |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|------------------------------|
| AP45 | A serious game for rehabilitation of neurological disabilities: Preliminary study[54] | 6 | 24 |
| AP46 | Artículo Científico-Diseño y desarrollo de un videojuego educativo con técnicas de inteligencia artificial para plataforma Android utilizando metodología OOHDM. Caso de estudio: Laberinto 3D.[55] | 0 | 10 |
| AP47 | Capítulo 3 UAdventure: El Renacer de eAdventure[37] | 0 | 24 |
| AP48 | Marco de trabajo ingenieril para el proceso de desarrollo de videojuegos.[56] | 1 | 38 |
| AP49 | Serious Games Architectures and Engines[57] | 4 | 25 |
| AP50 | Videojuego educativo como apoyo a la enseñanza de la algoritmia para los estudiantes del Programa Nacional de Formación en Sistemas e Informática[58] | 0 | 17 |

Al finalizar la preselección de artículos podemos observar en la Figura 2 una gráfica resumen del mismo, en el cual podemos observar que existe una gran variedad artículos

que concuerdan con las palabras claves o cadenas de búsquedas, pero que al realizar una revisión minuciosa de cada uno de ellos podemos ir acortando un poco más la selección de acuerdo a nuestros criterios de exclusión:

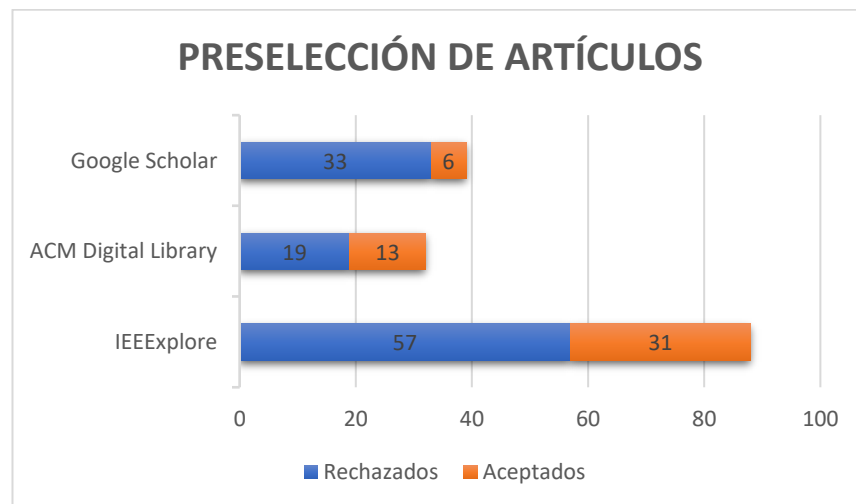


Fig. 2: Estudios preseleccionados de las diferentes fuentes digitales

b. Selección de Estudios primarios

- **Evaluación de la Calidad**

Se evaluarán 50 artículos que fueron preseleccionados (AP), con nuestro cuestionario previamente elaborado, quedando como resultado:

Tabla 17. Evaluación de la Calidad

| Código de Artículo | Puntuación: SI = 1 y NO = 0 | | | | | TOTAL |
|--------------------|-----------------------------|-----|-----|-----|-----|-------|
| | QA1 | QA2 | QA3 | QA4 | QA5 | |
| AP01 | 0 | 1 | 0 | 1 | 0 | 2 |
| AP02 | 1 | 1 | 0 | 0 | 0 | 2 |
| AP03 | 1 | 1 | 0 | 0 | 1 | 3 |
| AP04 | 0 | 1 | 0 | 1 | 1 | 3 |
| AP05 | 0 | 1 | 0 | 1 | 0 | 2 |
| AP06 | 0 | 1 | 0 | 1 | 0 | 2 |
| AP07 | 0 | 1 | 0 | 1 | 1 | 3 |
| AP08 | 0 | 1 | 1 | 1 | 0 | 3 |

| | | | | | | |
|------|---|---|---|---|---|---|
| AP09 | 0 | 1 | 0 | 1 | 0 | 2 |
| AP10 | 1 | 1 | 1 | 0 | 0 | 3 |
| AP11 | 1 | 1 | 0 | 1 | 1 | 4 |
| AP12 | 0 | 1 | 0 | 1 | 0 | 2 |
| AP13 | 0 | 1 | 0 | 1 | 1 | 3 |
| AP14 | 0 | 1 | 0 | 1 | 1 | 3 |
| AP15 | 1 | 1 | 0 | 1 | 1 | 4 |
| AP16 | 0 | 1 | 0 | 0 | 1 | 2 |
| AP17 | 1 | 1 | 0 | 1 | 0 | 3 |
| AP18 | 0 | 1 | 0 | 1 | 0 | 2 |
| AP19 | 0 | 1 | 0 | 1 | 0 | 2 |
| AP20 | 0 | 1 | 0 | 1 | 0 | 2 |
| AP21 | 1 | 1 | 0 | 1 | 0 | 3 |
| AP22 | 0 | 1 | 0 | 1 | 1 | 3 |
| AP23 | 0 | 1 | 0 | 1 | 0 | 2 |
| AP24 | 0 | 1 | 0 | 1 | 0 | 2 |
| AP25 | 1 | 1 | 0 | 0 | 0 | 2 |
| AP26 | 0 | 1 | 0 | 0 | 0 | 1 |
| AP27 | 0 | 0 | 1 | 1 | 1 | 3 |
| AP28 | 1 | 1 | 0 | 1 | 0 | 3 |
| AP29 | 0 | 1 | 0 | 0 | 0 | 1 |
| AP30 | 0 | 1 | 1 | 1 | 0 | 3 |
| AP31 | 0 | 1 | 1 | 1 | 0 | 3 |
| AP32 | 1 | 1 | 0 | 0 | 0 | 2 |
| AP33 | 0 | 1 | 0 | 0 | 0 | 1 |
| AP34 | 0 | 1 | 0 | 1 | 1 | 3 |
| AP35 | 0 | 1 | 0 | 1 | 0 | 2 |

| | | | | | | |
|------|---|---|---|---|---|---|
| AP36 | 0 | 1 | 0 | 0 | 0 | 1 |
| AP37 | 1 | 1 | 0 | 0 | 0 | 2 |
| AP38 | 1 | 1 | 1 | 0 | 0 | 3 |
| AP39 | 1 | 1 | 0 | 1 | 0 | 3 |
| AP40 | 1 | 1 | 0 | 1 | 0 | 3 |
| AP41 | 1 | 1 | 1 | 1 | 0 | 4 |
| AP42 | 0 | 1 | 0 | 1 | 0 | 2 |
| AP43 | 0 | 1 | 0 | 1 | 0 | 2 |
| AP44 | 0 | 1 | 0 | 1 | 1 | 3 |
| AP45 | 1 | 1 | 0 | 1 | 1 | 4 |
| AP46 | 1 | 1 | 1 | 1 | 0 | 4 |
| AP47 | 0 | 1 | 0 | 1 | 0 | 2 |
| AP48 | 0 | 1 | 1 | 1 | 0 | 3 |
| AP49 | 0 | 1 | 0 | 1 | 1 | 3 |
| AP50 | 1 | 1 | 1 | 1 | 0 | 4 |

Luego de realizar la evaluación de la calidad de los artículos pre-seleccionados teniendo en consideración el promedio de 3 puntos para ser catalogado como Primario, se logró obtener un total de 27 artículos primarios, con los cuales se llevará a cabo la extracción de información que den respuesta a las preguntas de investigación planteadas anteriormente.

c. Formato de Extracción de la Información

De cada documento, se extraerá algunos datos generales sobre la base de datos electrónica donde se recuperó el estudio, su título, revista o conferencia, autores, y fecha de publicación. Además, se recopilaron datos más específicos necesarios para lograr nuestros objetivos, tales como las respuestas a las preguntas de investigación que proporciona el estudio.

3.3. Análisis de Resultados y Hallazgos

Se enlistan 27 artículos con las etiquetas EP01 a EP27, que son los estudios primarios seleccionados de acuerdo a los criterios indicados, y que pasaron la evaluación de calidad preestablecida anteriormente. En la Tabla a continuación se presenta la información de cada uno de los estudios establecidos en el apartado formato de extracción de datos:

Tabla 18. Estudios seleccionados de la Revisión Sistemática de Literatura.

| Estudio primario EP01 | |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nombre: | A serious game with virtual reality for travel training with Autism Spectrum Disorder[14] |
| Revista / Conferencia: | 2015 International Conference on Virtual Rehabilitation (ICVR) |
| Fecha | 9-12 de junio de 2015 |
| Autor(es): | Bernárdez, Miguel; Barros, Fernando; Simoes, Marco and Castelo Branco, Miguel |
| Aporte al estudio: | |
| RQ1: | No se especifica la metodología utilizada en el desarrollo del videojuego |
| RQ2: | Unity Game engine |
| RQ3: | No se definen los criterios por el cual se eligió el motor de juegos. |
| Observación: | En la introducción del artículo, se menciona que éste describe el desarrollo y las pruebas de viabilidad de un juego serio que usa VR para rehabilitar individuos con ASD, pero en ningún apartado del documento se establece el desarrollo del juego creado. |
| Fuente Digital: | IEEE |

| Estudio primario EP02 | |
|-------------------------------|------------------------------------------------------------------------------------------------|
| Nombre: | A survey of frameworks and game engines for serious game development[5] |
| Revista / Conferencia: | Proceedings - IEEE 14th International Conference on Advanced Learning Technologies, ICAIT 2014 |
| Fecha | 7-10 de julio de 2014 |

| | |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Autor(es): | Cowan, Brent and Kapralos, Bill |
| Aporte al estudio: | |
| RQ1: | No se utiliza metodología de desarrollo |
| RQ2: | <ul style="list-style-type: none"> ● Unity Game engine ● Unreal Development kit ● Torque ● Flash ● Source Engine ● Ogre ● Second Life ● GameMaker ● XNA |
| RQ3: | <ul style="list-style-type: none"> ● Scripting ● Renderización ● Animación ● Inteligencia Artificial ● Física ● Audio ● Redes |
| Observación: | Este artículo proporciona una lista de los marcos y motores de juego más comúnmente identificados y un resumen de sus características proporcionando así información para aquellos que buscan herramientas para desarrollar juegos serios. |
| Fuente Digital: | IEEE |

| | |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| Estudio primario EP03 | |
| Nombre: | Choosing the infrastructure for entertainment and serious computer games - A whiteroom benchmark for game engine selection [17] |
| Revista / Conferencia: | 2013 5th International Conference on Games and Virtual Worlds for Serious Applications, VS-GAMES 2013 |
| Fecha | 11-13 de septiembre de 2013. |
| Autor(es): | Anderson, Eike Falk; McLoughlin, Leigh; Watson, Joe; Holmes, Sam; Jones, Peter; Pallett, Hayden and Smith, Brendan |
| Aporte al estudio: | |
| RQ1: | No se utiliza metodología de desarrollo |

| | |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RQ2: | <ul style="list-style-type: none"> ● Source engine ● Unity game engine ● Unreal Development kit ● CryEngine |
| RQ3: | <ul style="list-style-type: none"> ● Renderización 3D ● Inteligencia Artificial ● Física y Detección de Colisiones ● Sonido ● Scripting o Secuencia de Comandos |
| Observación: | Este documento hace un breve análisis de una posible clasificación de los motores de juego tomando como referencia el tipo de motor y su descripción. |
| Fuente Digital: | IEEE |

| Estudio primario EP04 | |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nombre: | Comparison of game engines for serious games[18] |
| Revista / Conferencia: | 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO) |
| Fecha | 22-26 de mayo de 2017. |
| Autor(es): | Sanja Pavkov; Ivona Franković; Nataša Hoić-Božić |
| Aporte al estudio: | |
| RQ1: | <p>SADDIE: es un modelo instruccional utilizado en la elaboración de proyectos eLearning, o experiencias basadas en el aprendizaje.</p> <p>Etapas:</p> <ul style="list-style-type: none"> ● Especificación ● Análisis ● Diseño ● Desarrollo ● Implementación y, ● Evaluación. |
| RQ2: | <ul style="list-style-type: none"> ● Adventure Game Studio ● Construct 2 ● e-Adventure ● GameMaker: Studio ● Phaser Editor. |

| | |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RQ3: | <ul style="list-style-type: none"> ● Lenguaje de programación ● Interfaz de fácil uso ● Precio ● Soporte |
| Observación: | A pesar de no ser un artículo explícitamente sobre la creación de Juegos Serios, este documento nos da una gran variedad de pautas para evaluar motores de juegos. |
| Fuente Digital: | IEEE |

| Estudio primario EP05 | |
|-------------------------------|--------------------------------------------------------------------------------------------------------------------|
| Nombre: | Design and development of a serious game for central line placement[20] |
| Revista / Conferencia: | 2015 International Conference on Virtual Rehabilitation (ICVR) |
| Fecha | 20-22 de junio de 2013. |
| Autor(es): | Khanal, Prabal; Katz, Daniel; De Maria, Samuel; Krol, Marina and Kahol, Kanav |
| Aporte al estudio: | |
| RQ1: | Interfaz del Juego, Base de datos, algoritmo y sistema de puntuación |
| RQ2: | Unreal Development kit |
| RQ3: | <ul style="list-style-type: none"> ● Facilidad ● Multiplataforma |
| Observación: | Este Juego Serio se trató de un plan piloto para el departamento de anestesiología que tuvo muy buenos resultados. |
| Fuente Digital: | IEEE |

| Estudio primario EP06 | |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------|
| Nombre: | Hugo Against Dengue: A serious game to educate people about Dengue fever prevention[21] |
| Revista / Conferencia: | SeGAH 2014 - IEEE 3rd International Conference on Serious Games and Applications for Health, Books of Proceedings |
| Fecha | 14-16 de mayo de 2014. |
| Autor(es): | Malheiros Porcino, Thiago; Strauss, Edilberto and Gonzalez Clua, Esteban |

| Aporte al estudio: | |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| RQ1: | <ul style="list-style-type: none"> ● Diseño de juego ● Diseño de producción y, ● Desarrollo de lógica y mecánica. |
| RQ2: | Unity Game engine |
| RQ3: | <ul style="list-style-type: none"> ● Portabilidad ● Facilidad de uso |
| Observación: | Sin observaciones. |
| Fuente Digital: | IEEE |

| Estudio primario EP07 | |
|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nombre: | Intelligent Game Engine for Rehabilitation (IGER) [23] |
| Revista / Conferencia: | IEEE Transactions on Computational Intelligence and AI in Games |
| Fecha | 26 de noviembre de 2014. |
| Autor(es): | Pirovano, Michele; Mainetti, Renato; Baud-Bovy, Gabriel; Lanzi, Pier Luca and Borghese, N. Alberto |
| Aporte al estudio: | |
| RQ1: | No se especifica la metodología utilizada en el desarrollo del videojuego |
| RQ2: | Panda 3D |
| RQ3: | Código Abierto |
| Observación: | Este videojuego incorpora un sistema difuso de manera que aprovechan el conocimiento de varios terapeutas para evitar errores en los ejercicios que los pacientes tienen que realizar, utiliza redes bayesianas para adecuar el nivel de dificultad adecuado para cada caso clínico. |
| Fuente Digital: | IEEE |

| Estudio primario EP08 | |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nombre: | Multimodal reconstruction of landscape in serious games for heritage: An insight on the creation of Fort Ross Virtual Warehouse serious game [24] |

| | |
|-------------------------------|--------------------------------------------------------------------------------------------|
| Revista / Conferencia: | 2013 Digital Heritage International Congress (Digital Heritage) |
| Fecha | 28 de octubre-1 de noviembre de 2013. |
| Autor(es): | Nicola Lercari; Maurizio; Lionel Onsurez and Joe Schultz |
| Aporte al estudio: | |
| RQ1: | No se especifica la metodología utilizada en el desarrollo del videojuego |
| RQ2: | Unity Game engine |
| RQ3: | <ul style="list-style-type: none"> ● Renderización. ● Presupuesto. |
| Observación: | Se enfoca más al desarrollo de paisajes dentro de un motor de juegos. |
| Fuente Digital: | IEEE |

| | |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Estudio primario EP09 | |
| Nombre: | Physical Rehabilitation based on Kinect Serious games [25] |
| Revista / Conferencia: | 2017 Eleventh International Conference on Sensing Technology (ICST) |
| Fecha | 4-6 diciembre. 2017 |
| Autor(es): | Oliveira, Raúl |
| Aporte al estudio: | |
| RQ1: | No se especifica la metodología utilizada en el desarrollo del videojuego |
| RQ2: | Unity Game engine |
| RQ3: | <ul style="list-style-type: none"> ● Admite gráficos 2D y 3D ● Lenguaje de programación. ● Se asocia con diferentes plataformas de computación: móviles, PC y PS4 |
| Observación: | Este proyecto es un juego de realidad virtual con interacción física mediante el sensor Kinect v2, para la realización de fisioterapia en casa. |
| Fuente Digital: | IEEE |

| Estudio primario EP10 | |
|-------------------------------|------------------------------------------------------------------------------------------------------|
| Nombre: | Seeker: A serious game for improving cognitive abilities[27] |
| Revista / Conferencia: | 2015 14th RoEduNet International Conference - Networking in Education and Research (RoEduNet NER) |
| Fecha: | 24-26 de septiembre de 2015. |
| Autor(es): | Irina Toma; Mihai Dascalu and Stefan Trausan-Matu |
| Aporte al estudio: | |
| RQ1: | No se especifica la metodología utilizada en el desarrollo del videojuego |
| RQ2: | Unity Game engine |
| RQ3: | <ul style="list-style-type: none"> ● Multiplataforma ● Interfaz de Fácil uso |
| Observación: | Sin observaciones. |
| Fuente Digital: | IEEE |

| Estudio primario EP11 | |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------|
| Nombre: | Serious games for the Elicitation of way-finding behaviours in emergency situations[31] |
| Revista / Conferencia: | Iberian Conference on Information Systems and Technologies, CISTI |
| Fecha: | 18-21 de junio de 2014. |
| Autor(es): | João Emílio Almeida; João Tiago Pinheiro Neto Jacob; Brígida Mónica Faria; Rosaldo JF Rossetti; and António Leca Coelho |
| Aporte al estudio: | |
| RQ1: | No se especifica la metodología utilizada en el desarrollo del videojuego |
| RQ2: | Unity Game engine |
| RQ3: | No se definen los criterios por el cual se eligió el motor de juegos. |
| Observación: | Sin observaciones. |
| Fuente Digital: | IEEE |

| Estudio primario EP12 | |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nombre: | Serious games, gamification and game engines to support framework activities in engineering: Case studies, analysis, classifications and outcomes [6] |
| Revista / Conferencia: | IEEE International Conference on Electro Information Technology |
| Fecha | 5-7 June 2014 |
| Autor(es): | Uskov Alexander and Sekar Bhuvana |
| Aporte al estudio: | |
| RQ1: | No se utiliza metodología de desarrollo |
| RQ2: | <ul style="list-style-type: none"> ● Unity Game engine ● Game Salad Creator ● DX Studio ● V-Play |
| RQ3: | <ul style="list-style-type: none"> ● Fidelidad audiovisual ● Fidelidad Funcional ● Precio ● Redes. |
| Observación: | A pesar de no ser un artículo explícitamente sobre la creación de Juegos Serios, este documento nos da una gran variedad de herramientas para la creación de Serious Games, así como también las pautas para la selección de motores de juegos. |
| Fuente Digital: | IEEE |

| Estudio primario EP13 | |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nombre: | Towards a service-oriented architecture framework for educational serious games [36] |
| Revista / Conferencia: | Proceedings - IEEE 15th International Conference on Advanced Learning Technologies: Advanced Technologies for Supporting Open Access to Formal and Informal Learning, ICALT 2015 |
| Fecha | 6-9 de julio de 2015. |
| Autor(es): | Carvalho Maira B.; Bellotti Francesco; Hu Jun; Hauge Jannicke Baalsrud; Berta Riccardo; Gloria Alessandro De and Rauterberg Matthias |
| Aporte al estudio: | |

| | |
|------------------------|-----------------------------------------------------------------------|
| RQ1: | ATMSG - Modelo de juegos serios basado en la teoría de actividades |
| RQ2: | Unity Game engine |
| RQ3: | No se definen los criterios por el cual se eligió el motor de juegos. |
| Observación: | Sin observaciones. |
| Fuente Digital: | IEEE |

| Estudio primario EP14 | |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| Nombre: | Using Serious Games to Create Awareness on Visual Impairments[39] |
| Revista / Conferencia: | 2015 International Conference on Developments of E-Systems Engineering (DeSE) |
| Fecha | 13-14 dic 2015 |
| Autor(es): | Melthis John; Brown Alex; Tang Stephen and Hanneghan Martin |
| Aporte al estudio: | |
| RQ1: | Desarrollo Rápido de Aplicaciones RAD |
| RQ2: | <ul style="list-style-type: none"> ● Unity Game engine ● Unreal Game Engine |
| RQ3: | <ul style="list-style-type: none"> ● Multiplataforma ● Cuentan con grandes comunidades de soporte en línea. |
| Observación: | Sin observaciones. |
| Fuente Digital: | IEEE |

| Estudio primario EP15 | |
|-------------------------------|------------------------------------------------------------------------------------------|
| Nombre: | Wanda: A Framework to Develop Card Based Games to Help Motivate Programming Students[40] |
| Revista / Conferencia: | 2014 Brazilian Symposium on Computer Games and Digital Entertainment |
| Fecha | 12-14 nov. 2014 |
| Autor(es): | Rafael Rego Drumond; André Luiz Brandão de Damasceno and Carlos de Salles Soares Neto |
| Aporte al estudio: | |

| | |
|------------------------|----------------------------------------------------------------------------------------------------------------------------|
| RQ1: | No se aplicó una metodología conocida, el proyecto se ejecutó en las siguientes fases: Diseño Conceptual e implementación. |
| RQ2: | Love2D |
| RQ3: | <ul style="list-style-type: none"> • Compatible con el lenguaje Lua. • Código abierto |
| Observación: | Sin observaciones. |
| Fuente Digital: | IEEE |

| Estudio primario EP16 | |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| Nombre: | UAdventure: The eAdventure reboot: Combining the experience of commercial gaming tools and tailored educational tools[37] |
| Revista / Conferencia: | 2017 IEEE Global Engineering Education Conference (EDUCON) |
| Fecha | 25-28 April 2017 |
| Autor(es): | Pérez Colado Iván J.; Pérez Colado Víctor M.; Martínez Ortiz Iván; Freire Moran Manuel and Fernández Manjon Baltasar |
| Aporte al estudio: | |
| RQ1: | No se especifica la metodología utilizada |
| RQ2: | Unity Game engine |
| RQ3: | <ul style="list-style-type: none"> • Documentación • Mantenibilidad |
| Observación: | Sin observaciones. |
| Fuente Digital: | IEEE |

| Estudio primario EP17 | |
|-------------------------------|---------------------------------------------------------------------------------------------------------------|
| Nombre: | Using Unity to Teach Game Development: When You've Never Written a Game[53] |
| Revista / Conferencia: | ITiCSE '15: Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education |
| Fecha | 4 al 8 de julio de 2015 |
| Autor(es): | Dickson, Paul E. |

| Aporte al estudio: | |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| RQ1: | No se especifica la metodología utilizada |
| RQ2: | Unity Game engine |
| RQ3: | <ul style="list-style-type: none"> ● Interfaz de Fácil uso ● Precio ● Multiplataforma ● Curva de aprendizaje |
| Observación: | Sin observaciones. |
| Fuente Digital: | ACM |

| Estudio primario EP18 | |
|-------------------------------|----------------------------------------------------------------------------------------------------------------------------|
| Nombre: | Development of Virtual Reality Based Robotic Surgical Trainer for Patient-specific Deformable Anatomy [49] |
| Revista / Conferencia: | AIR '15: Proceedings of the 2015 Conference on Advances in Robotics |
| Fecha | 02 - 04 de julio de 2015 |
| Autor(es): | Kalavakonda Niveditha; Chandra Sourav and Thondiyath Asokan |
| Aporte al estudio: | |
| RQ1: | No se especifica la metodología utilizada en el desarrollo del videojuego |
| RQ2: | Unity Game engine |
| RQ3: | <ul style="list-style-type: none"> ● Motor gráfico ● Motor de física |
| Observación: | Sin observaciones. |
| Fuente Digital: | ACM |

| Estudio primario EP19 | |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| Nombre: | Engine- Cooperative Game Modeling (ECGM): Bridge Model-Driven Game Development and Game Engine Tool-chains [50] |
| Revista / Conferencia: | ACE '16 Proceedings of the 13th International Conference on Advances in Computer Entertainment Technology |
| Fecha | 09 al 12 de noviembre de 2016 |

| | |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| Autor(es): | Zhu Meng; Wang Alf Inge and Trættemberg Hallvard |
| Aporte al estudio: | |
| RQ1: | Model-Driven Game Development (MDGD) |
| RQ2: | Torque 2D |
| RQ3: | <ul style="list-style-type: none"> ● Lenguaje de programación ● Programación orientada a objetos ● Editor |
| Observación: | Sin Observaciones. |
| Fuente Digital: | ACM |

| | |
|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Estudio primario EP20 | |
| Nombre: | An Experience-based Comparison of Unity and Unreal for a Stand-alone 3D Game Development Course [43] |
| Revista / Conferencia: | Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education - ITiCSE '17 |
| Fecha | 9-12 de junio de 2015 |
| Autor(es): | Bernardes, Miguel; Barros, Fernando; Simoes, Marco and Castelo Branco, Miguel |
| Aporte al estudio: | |
| RQ1: | No se utilizó ninguna metodología |
| RQ2: | <ul style="list-style-type: none"> ● Unity Game engine ● Unreal Game Engine |
| RQ3: | <ul style="list-style-type: none"> ● Características Básicas ● Curva de Aprendizaje ● Desarrollo multiplataforma ● Comunidad activa ● Documentación ● Precio |
| Observación: | Sin observaciones. |
| Fuente Digital: | ACM |

| Estudio primario EP21 | |
|-------------------------------|-------------------------------------------------------------------------------------|
| Nombre: | Development of Encantasya: War of the Four Kingdoms [47] |
| Revista / Conferencia: | ICSCA '17: Proceedings of the 6th International Conference on Software and Computer |
| Fecha | 26 al 28 de febrero de 2017 |
| Autor(es): | Arguson, Angelo C. and Aldea, III, William A. |
| Aporte al estudio: | |
| RQ1: | Scrum |
| RQ2: | Unity Game engine |
| RQ3: | Lenguaje de programación |
| Observación: | Sin Observaciones. |
| Fuente Digital: | ACM |

| Estudio primario EP22 | |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nombre: | Development of Laparoscopic Cholecystectomy Simulator Based on Unity Game Engine [48] |
| Revista / Conferencia: | CVMP '18: Proceedings of the 15th ACM SIGGRAPH European Conference on Visual Media Production |
| Fecha | 13 al 14 de diciembre de 2018 |
| Autor(es): | Jinglu Zhang, Yao Lyu, Yukun Wang, Yinyu Nie, Xiaosong Yang, Jianjun Zhang, and Jian Chang |
| Aporte al estudio: | |
| RQ1: | No se especifica la metodología utilizada en el desarrollo del videojuego |
| RQ2: | Unity Game engine |
| RQ3: | <ul style="list-style-type: none"> ● Multiplataforma ● Gran soporte de audio avanzado y efectos visuales ● Curva de aprendizaje ● Entorno de Desarrollo intuitivo |
| Observación: | Sin Observaciones. |
| Fuente Digital: | ACM |

| Estudio primario EP23 | |
|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nombre: | Serious Games Architectures and Engines[57] |
| Revista / Conferencia: | Entertainment Computing and Serious Games, 2016 |
| Fecha | 06 de octubre de 2016 |
| Autor(es): | Heinrich Söbke and Alexander Streicher |
| Aporte al estudio: | |
| RQ1: | No se utilizó metodología de desarrollo |
| RQ2: | <ul style="list-style-type: none"> ● Second Life ● Unity ● Unreal ● Flash ● XNA ● Torque ● OGRE ● GameMaker ● StoryTec ● OLIVE ● CryEngine ● SourceEngine |
| RQ3: | <ul style="list-style-type: none"> ● Facilidad de Uso ● Bajo costo ● Comunidad Activa ● Soporte ● Heterogeneidad - Multiplataforma ● Documentación ● Características gráficas ● Fidelidad audiovisual (inmersión) ● Fidelidad funcional (apoyo a objetivos de aprendizaje) ● Redes |
| Observación: | A pesar de no ser un artículo explícitamente sobre la creación de Juegos Serios, este documento nos da una gran variedad de herramientas para la creación de Serious Games, así como también las pautas para la selección de motores de juegos. |
| Fuente Digital: | Google Scholar |

| Estudio primario EP24 | |
|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nombre: | Diseño y desarrollo de un videojuego educativo con técnicas de inteligencia artificial para plataforma Android utilizando metodología OOHDM. Caso de estudio: Laberinto 3D. [55] |
| Revista / Conferencia: | GEEKS DECC - Universidad de las Fuerzas Armadas ESPE. Carrera de Ingeniería en Sistemas e Informática. |
| Fecha | 9-12 de junio de 2015 |
| Autor(es): | González Sotomayor Itamar Soledad |
| Aporte al estudio: | |
| RQ1: | Metodología de Diseño de Hipermedia Orientado a Objetos OOHDM |
| RQ2: | Unity Game engine |
| RQ3: | <ul style="list-style-type: none"> ● Graficas 3D ● Multiplataforma |
| Observación: | Sin Observaciones. |
| Fuente Digital: | Google Scholar |

| Estudio primario EP25 | |
|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nombre: | Videojuego educativo como apoyo a la enseñanza de la algoritmia para los estudiantes del Programa Nacional de Formación en Sistemas e Informática [58] |
| Revista / Conferencia: | Acuerdos de la IUI '18 Acompañante de la 23ª Conferencia Internacional sobre Interfaces de Usuarios Inteligentes Acompañante |
| Fecha | 15 noviembre 2013 |
| Autor(es): | Arocha, Elías Oswaldo Cisneros and Delgado, Ramón Collazo |
| Aporte al estudio: | |
| RQ1: | <ul style="list-style-type: none"> ● Proceso Unificado Rational RUP ● SUM |
| RQ2: | Blender |
| RQ3: | <ul style="list-style-type: none"> ● Documentación disponible ● Interfaz ● Multiplataforma |
| Observación: | Sin Observaciones. |

| | |
|------------------------|----------------|
| Fuente Digital: | Google Scholar |
|------------------------|----------------|

| Estudio primario EP26 | |
|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nombre: | Marco de trabajo ingenieril para el proceso de desarrollo de videojuegos. [56] |
| Revista / Conferencia: | Revista Antioqueña de las Ciencias Computacionales y la ingeniería de Software |
| Fecha | 19 de mayo 2017 |
| Autor(es): | Andy Hernández P., Karina Pérez T. and Omar Correa M. |
| Aporte al estudio: | |
| RQ1: | <ul style="list-style-type: none"> ● Cascada ● SCRUM ● XP ● SUM ● Huddle ● Marco de trabajo de Pressman ● Marco Edulibre |
| RQ2: | <ul style="list-style-type: none"> ● Unreal ● Unity ● Ardor3D ● Axioma Engine ● Cry engine ● Fox Engine ● Stingray ● Frostbite |
| RQ3: | <ul style="list-style-type: none"> ● Heterogeneidad ● Renderización ● Motor físico ● sonido ● Animación ● Inteligencia artificial ● redes ● streaming scripting ● administración de memoria |
| Observación: | Sin Observaciones. |
| Fuente Digital: | Google Scholar |

| Estudio primario EP27 | |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| Nombre: | A serious game for rehabilitation of neurological disabilities: Preliminary study[54] |
| Revista / Conferencia: | 2015 IEEE 4th Portuguese Meeting on Bioengineering (ENBENG) |
| Fecha | 26-28 de febrero de 2015 |
| Autor(es): | Martins, Tiago; Carvalho, Vítor and Soares, Filomena |
| Aporte al estudio: | |
| RQ1: | No se especifica la metodología utilizada en el desarrollo del videojuego |
| RQ2: | Unity Game engine |
| RQ3: | <ul style="list-style-type: none"> • Comunidad Activa • multiplataforma • integración con Microsoft Kinect |
| Observación: | Sin observaciones. |
| Fuente Digital: | Google Scholar |

a. Discusión

En esta sección, se analiza la información extraída de los estudios primarios, discute los hallazgos y se da respuestas a las tres preguntas de investigación que se formularon.

RQ1: ¿Qué metodologías se están utilizando en el desarrollo de Juegos Serios y cuáles son sus etapas?

En los artículos EP04, EP05, EP06, EP13, EP14, EP15, EP19, EP21, EP24, EP25 y EP26, nos presenta varias metodologías utilizadas en la creación de Juegos serios, cabe resaltar que todas se caracterizan por ser iterativas e incrementales, pudiendo mejorar el producto en cada etapa de desarrollo. Otro punto importante a considerar es que en los estudios EP05, EP06 y EP15, el modelo para su desarrollo fue establecido por autoría propia, ya que en estos estudios muestran una síntesis de varias metodologías y de acuerdo al mismo plantean su nuevo modelo a seguir para el desarrollo de este tipo de videojuegos.

En la Tabla 19 se muestran las metodologías encontradas que son utilizadas en el desarrollo de este tipo de software, las mismas que se encuentran con sus respectivas fases:

Tabla 19. Metodologías encontradas en los estudios primarios

| Código | Metodología | Fases |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| EP04 | SADDIE: Modelo sistemático de diseño instruccional, que se apoya en una orientación pedagógica del aprendizaje para producir con calidad una amplia variedad de materiales educativos, adecuados a las necesidades de aprendizajes | <ul style="list-style-type: none"> • Especificación • Análisis • Diseño • Desarrollo • Implementación y, • Evaluación. |
| EP05 | Modelo establecido por los autores | <ul style="list-style-type: none"> • Interfaz del Juego • Base de Datos • Algoritmo • Sistema de puntuación |
| EP06 | Modelo establecido por los autores | <ul style="list-style-type: none"> • Diseño de juego • Diseño de producción y, • Desarrollo de lógica y mecánica. |
| EP13 | ATMSG (Activity theory-based model of serious games model): El modelo tiene el objetivo de apoyar el análisis y diseño de juegos serios para dos grupos de usuarios: expertos en juegos serios y no expertos que participan en proyectos relacionados con juegos serios, como maestros y expertos en el dominio de aplicaciones. | <ul style="list-style-type: none"> • Descripción de las actividades • Representar la secuencia del juego • Identificar acciones, herramientas y marcador • Implementación |
| EP14 | RAD: El desarrollo rápido de aplicaciones, es un proceso de desarrollo de software, que comprende un desarrollo interactivo, construcción de prototipos y el uso de herramientas CASE. | <ul style="list-style-type: none"> • Modelado de Gestión • Modelado de datos • Modelado de proceso, • Generación de aplicaciones y, • Pruebas de entrega. |
| EP15 | Modelo establecido por los autores | <ul style="list-style-type: none"> • Diseño Conceptual • Implementación |
| EP19 | MDGD (Model-Driven Game Development): En el desarrollo de juegos impulsado por modelos (MDGD), los modelos se convierten | <ul style="list-style-type: none"> • Diseño del Juego (PIM) • Modelado del Juego (PSM) |

en elementos de primer orden en el desarrollo, mantenimiento y evolución de juegos. Al centrarse en modelos que especifican juegos en lugar del código que implementa juegos, se puede lograr un mayor nivel de abstracción y automatización del proceso de desarrollo

- Programación (Code)
- Pruebas (Game Binary)

| | | |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| EP21 | SCRUM: metodología ágil de desarrollo de software que puede adaptarse a distintos entornos de trabajo | <ul style="list-style-type: none"> • Preparación • Diseño • Desarrollo • Pruebas |
| EP24 | OOHDM (Object Oriented Hypermedia Design Method): Ayuda la elaboración de aplicaciones multimedia y tiene como objetivo simplificar y a la vez hacer más eficaz el diseño de aplicaciones hipermedia. | <ul style="list-style-type: none"> • Diseño Conceptual • Fase Navegacional • Fase de Interfaz Abstracta • Fase de Implementación |
| EP25 | SUM: La metodología SUM para videojuegos tiene como objetivos desarrollar videojuegos de calidad en tiempo y costo, así como la mejora continua del proceso para incrementar la eficacia y eficiencia de esta. | <ul style="list-style-type: none"> • Concepto • Planificación • Elaboración • Beta • Cierre |
| EP26 | <ul style="list-style-type: none"> • Cascada • SCRUM • XP • SUM • Huddle • Marco de trabajo de Pressman • Marco Edulibre | |

RQ2: ¿Cuáles son los motores de juegos actualmente utilizados en el desarrollo de Juegos Serios?

En la Figura 3 que se muestra a continuación, se puede observar que existe una gran gama de Motores de juegos disponibles para la creación de videojuegos educativos o Juegos serios, sin embargo; el Motor de Juegos Unity es el más utilizado en los artículos primarios que abarca esta investigación, ya que, 21 estudios consideraron el análisis de Unity Game Engine como herramienta para el desarrollo de su investigación debido a características tales como: Funcionalidad (Scripting, Renderización, Animación, Inteligencia Artificial, Física, Audio, y Redes) como lo mencionan en el EP02 y EP03;

Portabilidad e Interactividad (EP06), Modelos 3D gratuitos y recursos limitados como tiempo y presupuesto (EP08) entre otras características. De la misma manera podemos observar que el segundo motor de juegos más utilizado es Unreal Engine, debido a: la Facilidad de manejo y soporte a múltiples plataformas (EP05); soporte (EP14); Documentación, accesibilidad, entre otros (EP23).

MOTORES DE JUEGOS UTILIZADOS EN EL DESARROLLO DE JUEGOS SERIOS

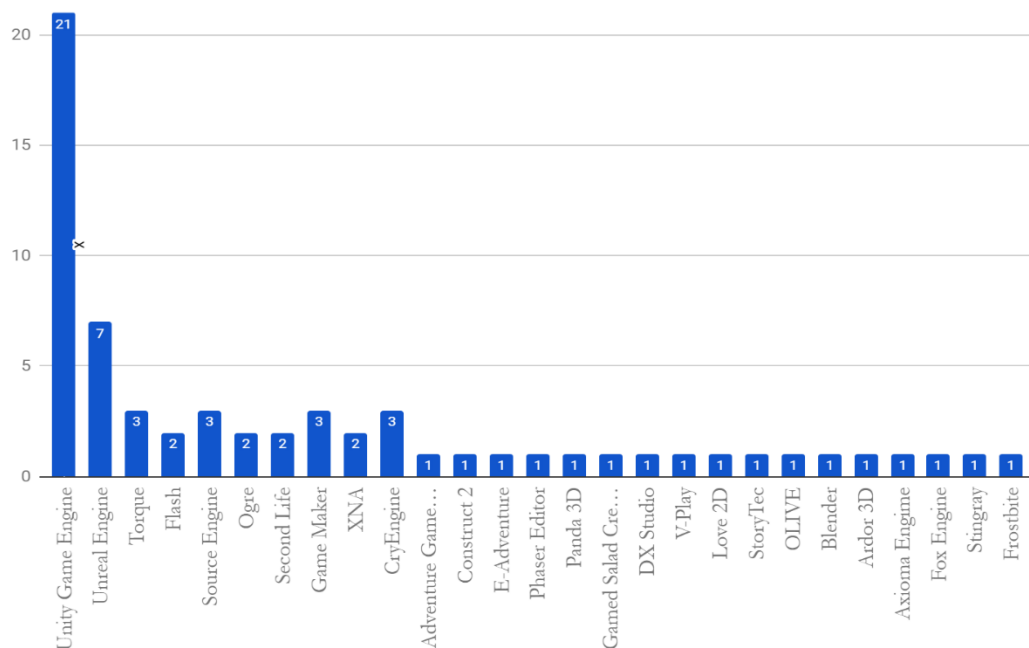


Fig. 3: Motores de juegos más utilizados en desarrollo de Juegos Serios

RQ3: ¿Cuáles son los criterios para la selección del motor de Juegos?

Si bien es cierto en el apartado anterior pudimos observar que existe un sinnúmero de motores de juegos, la elección del mismo no debería tomarse a la ligera, es por tal motivo que se ha recopilado los criterios que se han considerados para la selección de cada motor de juegos dentro de los estudios primarios y así poder determinar un listado depurado de los que considero deberían ser fundamentales para la selección del mismo.

En la Figura 4 se muestra la diversidad de criterios que se consideraron al momento de elegir un motor de juegos en los estudios primarios seleccionados para esta investigación:

CRITERIOS DE ELECCIÓN

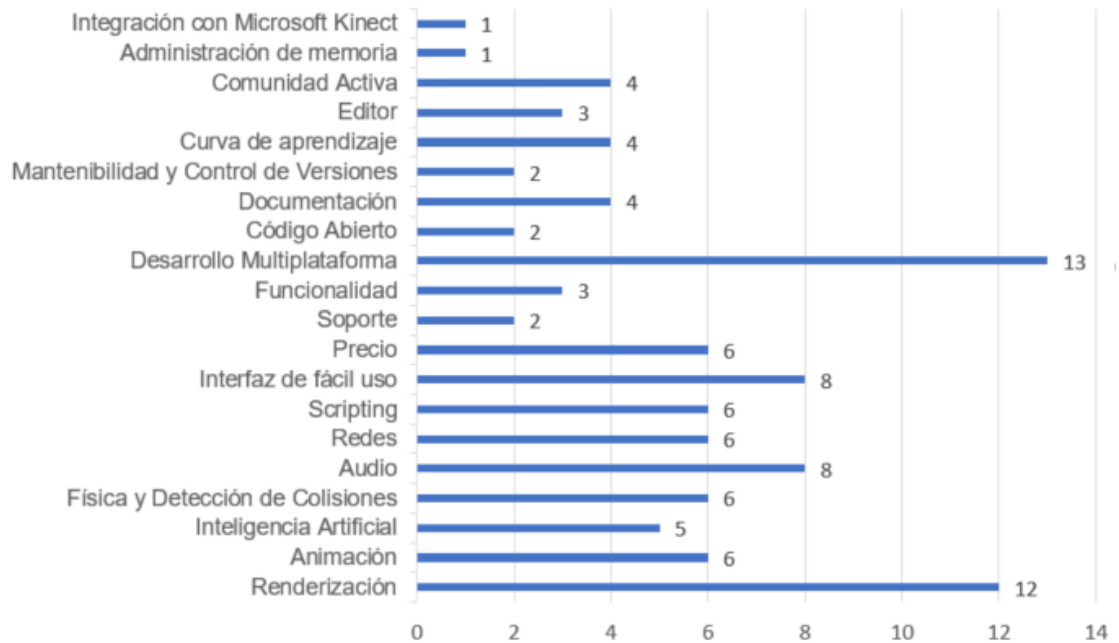


Fig. 4: Criterios para la selección del Motor de juegos utilizados en desarrollo de Juegos Serios

Como se puede observar contamos con 20 criterios encontrados que nos pueden ayudar a la selección del motor de juegos, de los cuales se depuró y agrupó de la siguiente manera:

- Componentes Básicos de un Motor de Juegos:
 - Renderización
 - Animación
 - Física
 - Audio
 - Inteligencia Artificial
 - Scripting

- Características Básicas del motor de juegos
 - Plataformas de distribución
 - Documentación Disponible
 - Lenguaje de Programación
 - Interfaz

- Tipo de Licencia
- Curva de aprendizaje
- Requisitos de Instalación

4. Conclusiones

Si bien es cierto existe un sinnúmero de metodologías de desarrollo de software que pueden ser adaptadas al desarrollo de videojuegos, la metodología SUM para videojuegos, está basada en Scrum debido al contexto con el que trabaja (programación rápida, precisa, optimizada y adaptable), tiene como objetivo desarrollar videojuegos de calidad en proyectos con escasa cantidad de recursos (tiempo, personal, costo, etc.), puede ser combinada fácilmente con otras metodologías de desarrollo para adaptarse a distintas realidades.

El proceso de creación de un videojuego es un proceso complejo y extenso en comparación con un proceso de software normal, esto debido a que dentro del proceso intervienen muchas personas de muchas áreas diferentes a la ingeniería en sistemas, las cuales toman un rol importante durante dicho proceso.

Los dos motores de juegos más utilizados actualmente son Unity y Unreal Engine, ambos ofrecen licencias de usuario final (EULA en sus siglas en inglés), tiendas de activos gratuitos y pagados, amplia documentación en inglés y grandes comunidades de desarrolladores que brindan soporte y responden a las más pequeñas inquietudes.

Unity ofrece una amplia gama de herramientas, su interfaz es intuitiva y de fácil uso; por su parte Unreal Engine tiene una curva de aprendizaje un poco más compleja, la mayor ventaja es la creación de gráficos de primera calidad, es de código abierto, dando la posibilidad que los desarrolladores puedan hacer los cambios necesarios que se ajusten a los requerimientos.

Elegir el motor correcto va más allá de la evaluación de las características de cada uno, realmente depende del tipo de juego que desee crear, la audiencia y su nivel de experiencia en el lenguaje de programación.

5. Referencias

- [1] B. Kitchenham *et al.*, "Systematic literature reviews in software engineering – A tertiary study," *Information and Software Technology*, vol. 52, no. 8. pp. 792–805, 2010.
- [2] Asociación española de empresas productoras y desarrolladoras de videojuegos y software de entretenimiento, "LIBRO BLANCO DE DESARROLLO ESPAÑOL DE VIDEOJUEGOS 2017," vol. 4, p. 126, Jan. 2018.
- [3] A. Chaves-Montero *et al.*, *Aprendizaje lúdico: Los videojuegos*, vol. 1. Google Libros, 2018.
- [4] C. Alonso-Fernandez, A. Calvo, M. Freire, I. Martinez-Ortiz, and B. Fernandez-Manjon, "Systematizing game learning analytics for serious games," *2017 IEEE Global Engineering Education Conference (EDUCON)*. 2017.
- [5] B. Cowan and B. Kapralos, "A Survey of Frameworks and Game Engines for Serious Game Development," *2014 IEEE 14th International Conference on Advanced Learning Technologies*. 2014.
- [6] A. Uskov and B. Sekar, "Serious games, gamification and game engines to support framework activities in engineering: Case studies, analysis, classifications and outcomes," *IEEE International Conference on Electro/Information Technology*. 2014.
- [7] H. Andrews, M. Bramar, L. Lupinski, and B. Kapralos, "A serious game for interview preparation," *2014 IEEE Games Media Entertainment*. 2014.
- [8] Y. Chaudy, T. M. Connolly, and T. Hainey, "EngAGe: A Link between Educational Games Developers and Educators," *2014 6th International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES)*. 2014.
- [9] C. Gatzidis, K. Parry, E. Kavanagh, A. Wilding, and D. Gibson, "Towards the Development of an Interactive 3D Coach Training Serious Game," *2009 Conference in Games and Virtual Worlds for Serious Applications*. 2009.
- [10] A. B. Saavedra, F. J. Álvarez Rodríguez, J. M. Arteaga, R. S. Salgado, and C. A. Collazos Ordoñez, "A serious game development process using competency approach," *Proceedings of the XV International Conference on Human Computer Interaction - Interacción '14*. 2014.
- [11] F. Xu, D. Buhalis, and J. Weber, "Serious games and the gamification of tourism," *Tourism Management*, vol. 60. pp. 244–256, 2017.
- [12] A. Calderón and M. Ruiz, "A systematic literature review on serious games evaluation: An application to software project management," *Computers & Education*, vol. 87. pp. 396–422, 2015.
- [13] Z. Ali and M. Usman, "A framework for game engine selection for gamification and

- serious games,” in *2016 Future Technologies Conference (FTC)*, 2016.
- [14] M. Bernardes, F. Barros, M. Simoes, and M. Castelo-Branco, “A serious game with virtual reality for travel training with Autism Spectrum Disorder,” in *2015 International Conference on Virtual Rehabilitation (ICVR)*, 2015.
- [15] K. Smith, Y. Shen, J. Shull, T. Dean, and J. Michaeli, “A toolkit for presenting advanced mathematics in serious games,” in *SoutheastCon 2016*, 2016.
- [16] M. K. Tageldeen, I. Elamvazuthi, N. Perumal, and T. Ganesan, “A virtual reality based serious games for rehabilitation of arm,” in *2017 IEEE 3rd International Symposium in Robotics and Manufacturing Automation (ROMA)*, 2017.
- [17] E. F. Anderson *et al.*, “Choosing the Infrastructure for Entertainment and Serious Computer Games - a Whiteroom Benchmark for Game Engine Selection,” in *2013 5th International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES)*, 2013.
- [18] S. Pavkov, I. Frankovic, and N. Hoic-Bozic, “Comparison of game engines for serious games,” in *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2017.
- [19] P. Mildner, B. John, A. Moch, and W. Effelsberg, “Creation of custom-made serious games with user-generated learning content,” in *2014 13th Annual Workshop on Network and Systems Support for Games*, 2014.
- [20] P. Khanal, D. Katz, S. De Maria, M. Krol, and K. Kahol, “Design and development of a serious game for central line placement,” in *Proceedings of the 26th IEEE International Symposium on Computer-Based Medical Systems*, 2013.
- [21] T. Porcino, E. Strauss, and E. Clua, “Hugo against dengue: a serious game to educate people about dengue fever prevention,” in *2014 IEEE 3rd International Conference on Serious Games and Applications for Health (SeGAH)*, 2014.
- [22] S. Cassar, M. Montebello, and S. Zammit, “Hybrid Peer to Peer and Server Client System for Limited Users Multiplayer First Person Style Games,” in *2014 6th International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES)*, 2014.
- [23] M. Pirovano, R. Mainetti, G. Baud-Bovy, P. L. Lanzi, and N. Alberto Borghese, “Intelligent Game Engine for Rehabilitation (IGER),” *IEEE Trans. Comput. Intell. AI Games*, vol. 8, no. 1, pp. 43–55, 2016.
- [24] N. Lercari, M. Forte, and L. Onsurez, “Multimodal reconstruction of landscape in serious games for heritage: An insight on the creation of Fort Ross Virtual Warehouse serious game,” in *2013 Digital Heritage International Congress (Digital Heritage)*, 2013.
- [25] D. Ferreira, R. Oliveira, and O. Postolache, “Physical rehabilitation based on Kinect serious games,” in *2017 Eleventh International Conference on Sensing Technology*

(ICST), 2017.

- [26] W. Scacchi, "Practices and Technologies in Computer Game Software Engineering," *IEEE Soft.*, vol. 34, no. 1, pp. 110–116, 2017.
- [27] I. Toma, M. Dascalu, and S. Trausan-Matu, "Seeker: A serious game for improving cognitive abilities," in *2015 14th RoEduNet International Conference - Networking in Education and Research (RoEduNet NER)*, 2015.
- [28] M. M. Shashkov, C. S. Nguyen, M. Yopez, M. Hess-Flores, and K. I. Joy, "Semi-autonomous digitization of real-world environments," in *2014 Computer Games: AI, Animation, Mobile, Multimedia, Educational and Serious Games (CGAMES)*, 2014.
- [29] J. B. Marshall, G. Tyson, J. Llanos, R. M. Sanchez, and F. B. Marshall, "Serious 3D gaming research for the vision impaired," in *2015 17th International Conference on E-health Networking, Application & Services (HealthCom)*, Boston, MA, USA, 2015, pp. 468–471.
- [30] V. B. Vasudevamurt and A. Uskov, "Serious game engines: Analysis and applications," in *2015 IEEE International Conference on Electro/Information Technology (EIT)*, 2015.
- [31] J. E. Almeida, Joao Tiago Pinheiro, B. M. Faria, R. J. F. Rossetti, and A. L. Coelho, "Serious games for the Elicitation of way-finding behaviours in emergency situations," in *2014 9th Iberian Conference on Information Systems and Technologies (CISTI)*, 2014.
- [32] B. F. Goldiez and A. Angelopoulou, "Serious Games - Creating an Ecosystem for Success," in *2016 8th International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES)*, 2016.
- [33] V. McArthur and R. J. Teather, "Serious mods: A case for modding in serious games pedagogy," in *2015 IEEE Games Entertainment Media Conference (GEM)*, 2015.
- [34] J. M. Patricio, M. C. Costa, J. A. Carranca, and B. Farropo, "SolarSystemGO — An augmented reality based game with astronomical concepts," in *2018 13th Iberian Conference on Information Systems and Technologies (CISTI)*, 2018.
- [35] L. Eaachak, A. Belhabibe, and M. Bouhorma, "Towards a new concept of serious games generator," in *2015 International Conference on Electrical and Information Technologies (ICEIT)*, 2015.
- [36] M. B. Carvalho *et al.*, "Towards a Service-Oriented Architecture Framework for Educational Serious Games," in *2015 IEEE 15th International Conference on Advanced Learning Technologies*, 2015.
- [37] I. J. Perez-Colado, V. M. Perez-Colado, I. Martinez-Ortiz, M. Freire-Moran, and B. Fernandez-Manjon, "UAdventure: The eAdventure reboot: Combining the experience of commercial gaming tools and tailored educational tools," in *2017*

IEEE Global Engineering Education Conference (EDUCON), 2017.

- [38] J. Broderick, J. Duggan, and S. Redfern, "Using game engines for marine visualisation and collaboration," in *2016 International Conference on Image, Vision and Computing (ICIVC)*, Portsmouth, United Kingdom, 2016, pp. 96–101.
- [39] J. Melthis, A. Brown, S. Tang, and M. Hanneghan, "Using Serious Games to Create Awareness on Visual Impairments," in *2015 International Conference on Developments of E-Systems Engineering (DeSE)*, 2015.
- [40] R. R. Drumond, A. L. B. de Damasceno, and C. de Salles Soares Neto, "Wanda: A Framework to Develop Card Based Games to Help Motivate Programming Students," in *2014 Brazilian Symposium on Computer Games and Digital Entertainment*, 2014.
- [41] L. Ivanov, "3D game development with unity in the computer science curriculum," *Journal of Computing Sciences in Colleges*, vol. 31, no. 1, pp. 167–173, Oct. 2015.
- [42] P. Campillo-Sanchez and J. J. Gomez-Sanz, "A Framework for Developing Multi-Agent Systems in Ambient Intelligence Scenarios," in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, 2015, pp. 1949–1950.
- [43] P. E. Dickson, J. E. Block, G. N. Echevarria, and K. C. Keenan, "An Experience-based Comparison of Unity and Unreal for a Stand-alone 3D Game Development Course," in *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education - ITiCSE '17*, 2017.
- [44] K. Fukahori, D. Sakamoto, J. Kato, and T. Igarashi, "CapStudio: an interactive screencast for visual application development," in *Proceedings of the extended abstracts of the 32nd annual ACM conference on Human factors in computing systems*, 2014, pp. 1453–1458.
- [45] S. Faltaous, T. Machulla, M. Baumann, and L. Chuang, "Developing a Highly Automated Driving Scenario to Investigate User Intervention," in *Proceedings of the 9th International Conference on Automotive User Interfaces and Interactive Vehicular Applications Adjunct - AutomotiveUI '17*, 2017.
- [46] Y. Kawai, Y. Kaizu, K. Kawahara, Y. Obuchi, S. Otsuka, and S. Tomimatsu, "Development of a Tsunami Evacuation Behavior Simulation System with Massive Evacuation Agents," in *Proceedings of the 23rd International Conference on Intelligent User Interfaces Companion - IUI 18*, 2018.
- [47] A. C. Arguson and W. A. Aldea, "Development of Encantasya: War of the Four Kingdoms," in *Proceedings of the 6th International Conference on Software and Computer Applications - ICSCA '17*, 2017.
- [48] J. Zhang *et al.*, "Development of laparoscopic cholecystectomy simulator based on unity game engine," in *Proceedings of the 15th ACM SIGGRAPH European Conference on Visual Media Production - CVMP '18*, 2018.

- [49] N. Kalavakonda, S. Chandra, and A. Thondiyath, "Development of virtual reality based robotic surgical trainer for patient-specific deformable anatomy," in *Proceedings of the 2015 Conference on Advances In Robotics*, 2015, p. 16.
- [50] M. Zhu, A. I. Wang, and H. Trættemberg, "Engine- Cooperative Game Modeling (ECGM)," in *Proceedings of the 13th International Conference on Advances in Computer Entertainment Technology - ACE2016*, 2016.
- [51] J. de Pedro, G. Esteban, M. A. Conde, and C. Fernández, "HCore: a game engine independent OO architecture for fast development of haptic simulators for teaching/learning," in *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality - TEEM '16*, Salamanca, Spain, 2016, pp. 1011–1018.
- [52] J. Minogue, D. Borland, M. Russo, and S. T. Chen, "Tracing the Development of a Haptically-enhanced Simulation for Teaching Phase Change," in *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts - CHI PLAY Companion '16*, 2016.
- [53] P. E. Dickson, "Using Unity to Teach Game Development: When You've Never Written a Game," in *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education - ITICSE '15*, Vilnius, Lithuania, 2015, pp. 75–80.
- [54] T. Martins, V. Carvalho, and F. Soares, "A serious game for rehabilitation of neurological disabilities: Preliminary study," in *2015 IEEE 4th Portuguese Meeting on Bioengineering (ENBENG)*, 2015.
- [55] I. S. González Sotomayor, "Artículo Científico - Diseño y desarrollo de un videojuego educativo con técnicas de inteligencia artificial para plataforma Android utilizando metodología OOADM. Caso de estudio: Laberinto 3D," August. 2014.
- [56] A. Hernández P, K. Pérez T., and O. Correa M., "Marco de trabajo ingenieril para el proceso de desarrollo de videojuegos," *Revista Antioqueña de las Ciencias Computacionales y la Ingeniería de Software*, May 2017.
- [57] H. Söbke and A. Streicher, "Serious Games Architectures and Engines," in *Lecture Notes in Computer Science*, 2016, pp. 148–173.
- [58] E. O. Cisneros Arocha and R. Collazo Delgado, "Videojuego educativo como apoyo a la enseñanza de la algoritmia para los estudiantes del Programa Nacional de Formación en Sistemas e Informática," 2014.

ANEXO 2: Game Design Document

CLEAN RUNNER

Game Design Document

agosto 2019

Revisión

Jennifer Tomalá González

Contenido

| | | |
|------|------------------------------------|-----|
| 1. | INTRODUCCION | 118 |
| 1.1. | Concepto del juego | 118 |
| 1.2. | Características Principales | 118 |
| 1.3. | Genero | 119 |
| 1.4. | Propósito y público objetivo | 119 |
| 1.5. | Jugabilidad | 119 |
| 1.6. | Estilo visual | 120 |
| 1.7. | Alcance | 120 |
| 2. | MECÁNICAS DEL JUEGO | 120 |
| 2.1. | Jugabilidad | 120 |
| 2.2. | Flujo del juego | 121 |
| 2.3. | Personajes | 121 |
| 2.4. | Movimientos y física | 122 |
| 3. | INTERFAZ | 122 |
| 3.1. | Diagrama de flujo | 122 |
| 3.2. | Menú principal | 123 |
| 3.3. | Nivel | 124 |
| 3.4. | Pause | 124 |
| 3.5. | Final del juego | 125 |
| 4. | ARTE | 126 |
| 4.1. | Arte 3D | 126 |
| 4.2. | Audio/Sonido | 127 |

1. INTRODUCCION

Este es el documento de diseño del Prototipo de Juego Serio: **Clean Runner**. Un Juego serio para Pc, realizado tanto con el motor de videojuegos Unreal Engine versión 4.21 y el motor de videojuegos Unity Game Engine versión 2018.3.5f1 Personal. El presente tiene como objetivo plasmar los elementos que incluirá **Clean Runner** en su versión prototipo y servir como base para que a futuro terminen su elaboración incrementando más escenarios y demás elementos, que ayuden a concientizar a los niños de los primeros años de Educación Básica sobre la importancia ambiental que tiene el reciclaje y la limpieza del entorno con el cual conviven.

1.1. Concepto del juego

Clean Runner es un juego de tipo Endless Runner (Corredor sin fin), en el que se controla con movimientos de izquierda a Derecha del personaje de la historia: el mismo que se llama Sergio, que en latín su significado es "Guardián". Sergio se encaminará, a través de una serie de plataformas intentando recopilar un mayor número establecido de botellas localizadas en el suelo, procurando que las calles queden lo más limpias posibles, y evitando los obstáculos que se le presenten en el camino de manera rápida, ya que podría colisionar contra los mismos, teniendo como objetivo primordial concientizar a los usuarios en temas de limpieza y reciclaje, adoptando cognoscitivamente un alto nivel de compromiso ambiental, especialmente en edades en donde estos hábitos se interioricen con facilidad.

1.2. Características Principales

El juego se basa en:

- **Planteamiento sencillo:** Debido a que es un prototipo de prueba para evaluar los motores de juegos antes mencionados, está enfocado a niños de edad escolar, se ha buscado un escenario sencillo con el cual el infante se sienta identificado y comprometido en la tarea ambiental propuesta.
- **Acción:** Es un juego dinámico, con cierta sensación de tensión ya que pone al usuario en una carrera de agilidad, lo que permite captar la mayor atención y concentración del infante, a la vez que se siente identificado con el personaje, apoyando directamente a la motricidad debido a que tendrá que reaccionar muy

rápido antes de que colisione con algún objeto u otros obstáculos concebidos en las reglas del juego.

1.3. Género

Atendiendo a las características del juego, **Clean Runner** se enmarca dentro de varios géneros:

- **Endless Runner:**
Un género de corredor sin fin, basado en juegos tales como Subway surfeas, que tienen como objetivo que el personaje principal Corra sin parar, recolecte objetos y evite chocar con algunos obstáculos.
- **Acción en tercera persona:**
Es un juego dinámico y directo en el que el jugador experimenta una descarga de adrenalina, la cámara se situara de tras del jugador logrando poder ver en gran parte la escena.

1.4. Propósito y público objetivo

El objetivo principal es desarrollar un gran nivel de conciencia ambiental y está enfocado a los niños de los primeros años de educación básica, debido a que durante esta etapa se propicia el desarrollo motriz, sensorial e intelectual de los mismos, facilitando la adopción de nuevos hábitos y habilidades.

1.5. Jugabilidad

Como se ha comentado anteriormente, se trata de un juego sencillo enfocado a la concientización ambiental, en su primera versión dispone de un nivel de juego sencillo con movimientos básicos de desplazamiento, al cual se le podría ampliar aún más al futuro.

- **Movilidad:** El personaje principal se desplazará través del escenario en 3ª persona, con las teclas destinadas para esa acción.
- **Sistema de Puntuación:** se dispone de un sistema que permitirá sumar números por cada botella recogida, permitiendo guardar el resultado por cada

partida del mismo, pudiendo realizar comparaciones entre las diversas partidas realizadas por cada infante. La medición de puntajes se lo hace a nivel de las botellas recogidas, como primer nivel al obtener 50 botellas alcanzara una medalla de ecologista.

- **Muerte Súbita:** El jugador ira esquivando los obstáculos que se le presenten en el transcurso de la partida propiciando ase mantener la concentración del mismo y desarrollando la motricidad fina durante la partida, al momento de no lograr esquivar el obstáculo, esté colisionará con el mismo, dando así por terminado el juego.

1.6. Estilo visual

Clean Runner tendrá un estilo sencillo en 3D, pero tratando de ser lo más realista en cuanto a texturas y colores se trata, con personajes simples y no demasiado detallista.

1.7. Alcance

El objetivo principal es lograr un alto grado de compromiso ambiental en los niños a través del desarrollo de este Juego Serio.

2. MECÁNICAS DEL JUEGO

En esta sección entraremos en detalle en lo que a las mecánicas de **Clean Runner** se refiere. Se comentarán todos los pilares que fundamentan su jugabilidad y se detallaran las acciones que podrá llevar a cabo el jugador dentro de una típica partida. Además, se ofrecerá una lista con el personaje del juego, habilidades, objetos, etc. Por último, se modelará el mundo en el plano de movimiento, físicas y detección de colisiones.

2.1. Jugabilidad

- **Niveles:** Por ser un prototipo de juego sea creado un solo nivel que consta de un paisaje urbano como ciudadela con una calle con 3 carriles por la cual el jugador se moverá de izquierda a derecha, el mismo que tendrá obstáculos sobre todo mobiliario que dificulta el tráfico por la zona donde se lo encuentre.

- **Foco:** Nuestro personaje se verá expuesto a los obstáculos que aparecen aleatoriamente en su camino, el movimiento del personaje será sobre un plano (suelo), más adelante se dará detalle de los movimientos que este puede realizar

2.2. Flujo del juego

A lo largo de esta sección se detallará el transcurso de la partida típica de prototipo a desarrollar, se comentarán los pasos que ha de seguir desde el arranque del juego hasta el fin del mismo. Poco a poco se irá desgranando el funcionamiento exacto del juego, en esta sección se describen las mecánicas.

El Jugador inicia el juego y se le presenta el menú principal, si desea iniciar la partida el jugador seleccionará la opción jugar caso contrario podrá cancelarla. El jugador inmediatamente que selecciona la opción jugar el personaje comienza a correr sin parar desde el centro del piso. Desde ese momento empezaran las botellas a aparecer en el suelo para ser recogidas, inmediatamente recoja una botella se incrementará el sistema de puntaje en 1 el cual se podrá observar al margen superior izquierdo de la pantalla, de la misma manera que aparecen las botellas aparecen los obstáculos en el camino, el jugador terminara su partida en el momento que choque con alguna pared mientras gira o algún obstáculo que se presente en el camino.

Cabe mencionar que durante el juego el jugador podrá optar por pausar la partida al presionar la tecla P y retomarla al seleccionar la opción continuar.

2.3. Personajes

- **Protagonista:** Sergio es el protagonista de este videojuego, es un ciudadano más dentro de la ciudad y es quien quiere marcar la diferencia cuidando y protegiendo la ciudad para que no se convierta en un basurero gigante, Sergio es un joven simpático, y apasionado por la limpieza, su apariencia es muy sencilla y no lleva ropas de lujo ni adornos.

2.4. Movimientos y física

- **Interacción con los elementos**

Clean Runner se desarrolla sobre un plano por el cual se desplaza, dentro del cual se presentan varios obstáculos como mobiliario o paredes que no pueden ser atravesados. Al momento que el personaje choque con uno de ellos colisiona y cae dando por terminada la partida.

El definitiva las colisiones que se producirán son:

Personaje – Obstáculo.

Personaje – Pared.

Personaje – Escenario.

- **Controles**

Movimiento: teclas direccionales o A para moverme a la izquierda y D para la derecha.

Saltar: Barra espaciadora.

Pausar: tecla P.

3. INTERFAZ

En esta sección se especificará con detalle cada una de las pantallas que componen **Clean Runner**, además se indicaran las transiciones entre ellas, así como la utilidad de cada elemento GUI (Graphical User Interface). Las imágenes adjuntas son bocetos que ilustran los componentes que debe contener cada pantalla, no obstante, se podrán realizar cambios en la apariencia y disposición de los elementos si así se lo consideran oportuno.

3.1. Diagrama de flujo

El siguiente diagrama de estados muestra las pantallas presentes a lo largo de **Clean Runner** y las transiciones entre ellas. En puntos posteriores nos centraremos en ellas de forma individual.

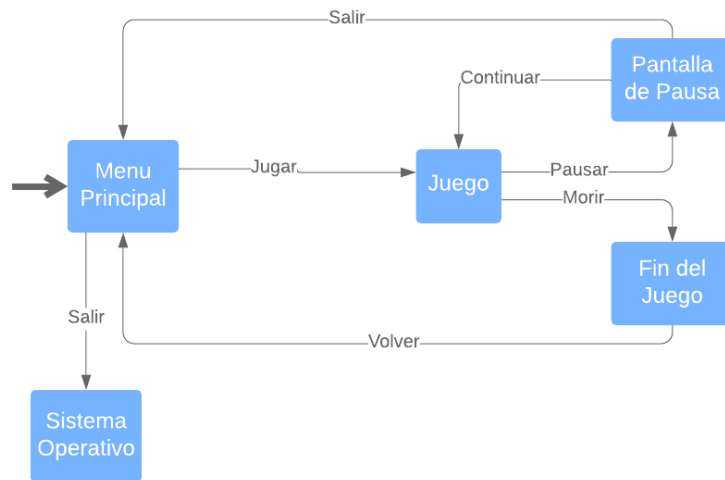


Ilustración 1: Diagrama de flujo de pantallas en el juego.

3.2. Menú principal

A continuación, el boceto de la pantalla de Menú Principal:

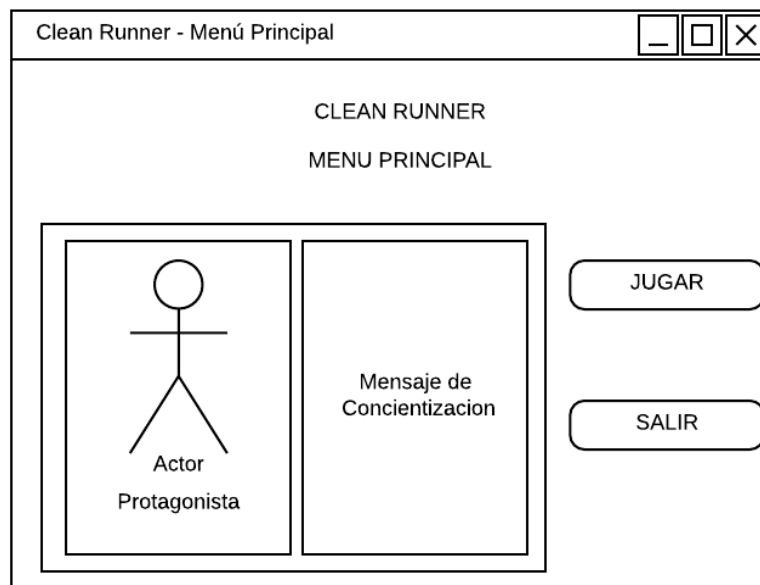


Ilustración 2: Boceto del menú principal

Lista y descripción de todos sus componentes:

- **Botón Jugar:** Al pulsarlo lleva a la pantalla de acción del juego en sí.
- **Botón Salir:** Al pulsarlo nos lleva de vuelta al Sistema Operativo.

- **Protagonista:** vista 3D del personaje principal del Juego a modo de ambientación.
- **Mensaje de Concientización:** Mensaje que impulsa el cuidado del medio ambiente y conservación de espacios libre de desechos plásticos.

3.3. Nivel

A continuación, el boceto de la pantalla del Nivel:

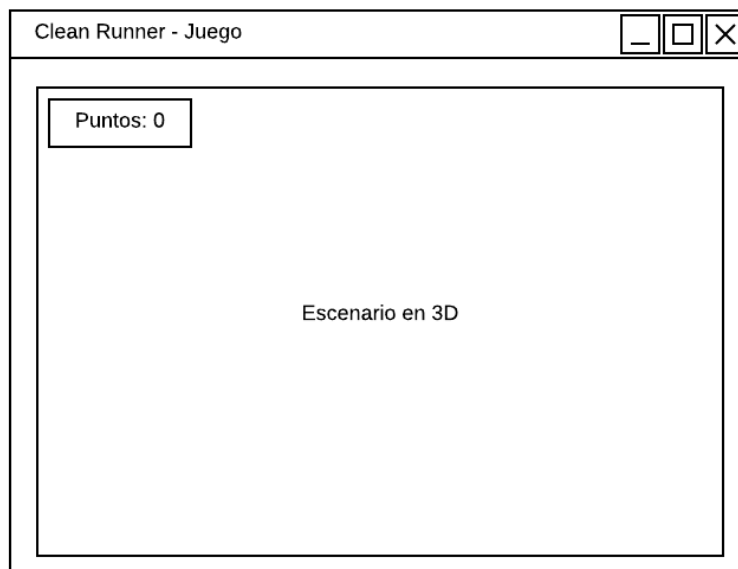


Ilustración 3: Boceto de la pantalla del juego.

Lista y descripción de todos sus componentes:

- **Escenario 3D:** Panel de juego.
- **Puntos:** Etiqueta con el sistema de puntuación de acuerdo a la cantidad de objetos recolectados.

3.4. Pause

A continuación, el boceto de la pantalla de Pause:

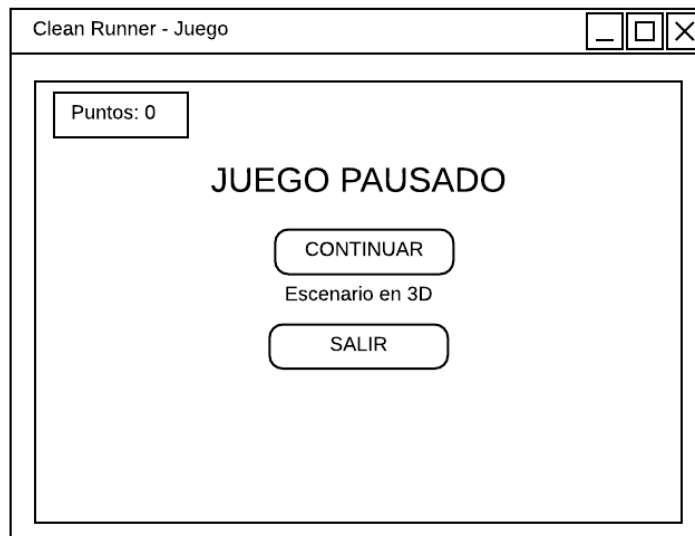


Ilustración 4: Boceto de la pantalla de Pause.

Lista y descripción de todos sus componentes:

- **Botón Continuar:** Vuelve al momento en que se quedó pausado el juego.
- **Botón Salir:** Al pulsarlo nos lleva de vuelta al Menú Principal.
- **Juego Pausado:** Etiqueta con el mensaje de aviso que el juego ha sido pausado.

3.5. Final del juego

A continuación, el boceto de la pantalla de Final de Juego:

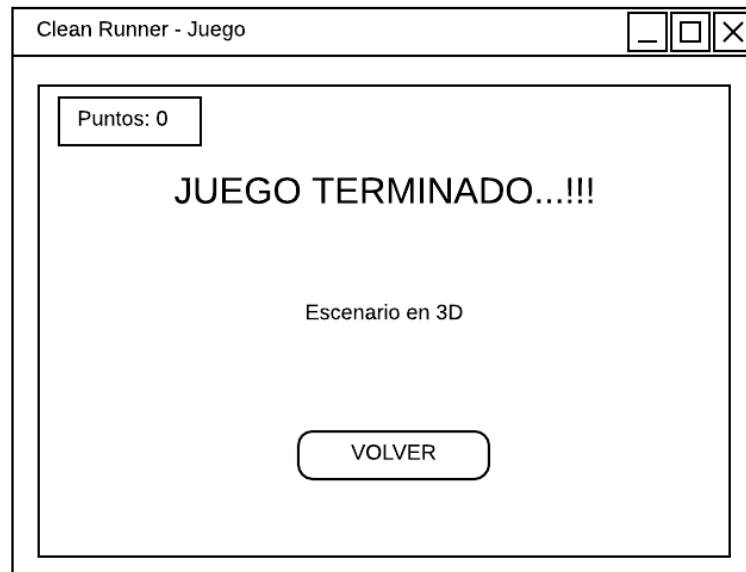


Ilustración 5: Boceto de la pantalla de Final del Juego.

Lista y descripción de todos sus componentes:

- **Botón Volver:** Retorna al menú principal.
- **Juego Terminado:** Etiqueta con el mensaje de aviso que el juego ha terminado.

4. ARTE

Sergio Limpiando la ciudad debe tener un carácter alegre, lleno de vida con un toque de realismo. Para el joven protagonista es todo un reto el tener que ir limpiando las calles recolectando las botellas que están esparcidas por todos lados. Los modelos utilizados son modelos básicos que se encuentran en cada una de las tiendas de los respectivos motores de juego y los sonidos son similares al de los juegos clásicos. Siendo este un prototipo los efectos son sencillos.

4.1. Arte 3D

Todos los modelos 3D son guardados e importados en formato (.FBX) u (.OBJ) que son compatibles con los motores.

- El personaje posee animaciones tales de correr, girar, saltar y morir
- También existen objetos tales como:

- Carretera.
- Edificios.
- Casas.
- Faros o lámparas.
- Basureros.
- Cercas.
- Botellas.
- Puestos de comidas.
- Hidrante.

4.2. Audio/Sonido

Al ser un pequeño proyecto se utilizará un sonido clásico al momento que el protagonista recoge las botellas, también se optará por insertar un fondo musical animado para que el juego de la sensación de tensión.

ANEXO 3: Guía de Instalación de Unreal Game Engine

1. En un navegador digitamos la dirección <https://www.unrealengine.com/> que es la página oficial de Unreal y en la parte superior derecha presionamos el botón DOWNLOAD.

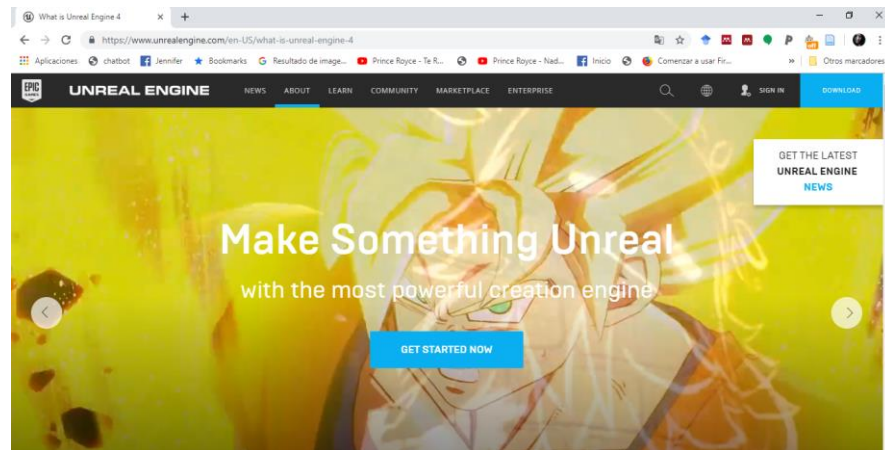


Figura 1: Pagina web de Unreal

2. Nos re direcciona a la página de registro de EPIC GAMES empresa desarrolladora de este motor de videojuegos. En el cual debemos ingresar los datos que se nos solicitan, aceptamos los términos de servicios y presionamos el botón crear usuario.

The image shows the Epic Games account creation interface. At the top is the Epic Games logo. Below it is the heading "Crear Cuenta". There are five social media icons: PlayStation, Xbox, Nintendo Switch, Facebook, and Google. The word "INSCRIBIRSE" is displayed above a dropdown menu showing "ECUADOR". Below this are two input fields for "Jenny" and "Tomalá", followed by a combined field for "Jtomala" with an information icon. The email field contains "jctomalag@unl.edu.ec" with a link below it that says "¿Quieres decir jctomalag@unl.edu?". A password field is shown with masked characters. A reCAPTCHA widget is present with the text "No soy un robot" and the reCAPTCHA logo. Below the widget are two checkboxes: one for receiving news (unchecked) and one for accepting terms of service (checked). At the bottom is a blue "CREAR CUENTA" button and a link for "Política de privacidad".

Figura 2: Formulario de Creación de cuenta en Epic Games

3. Se muestra el Acuerdo de la licencia del usuario final del cual tenemos dos opciones de licencias:
- **Desarrollador de Juegos:** Esta licencia es para los desarrolladores de juegos que pretenden producir títulos para su distribución.
 - **Enterprise:** Esta licencia libre de derechos de autor es para profesionales de la arquitectura, diseño de productos, fabricación, visualización de diseño y entretenimiento lineal, educadores y estudiantes que no tienen la intención de lanzar a los consumidores el software con tecnología Unreal. Cubre Unreal Studio y Unreal Engine 4.

De acuerdo a nuestras necesidades Seleccionaremos la opción que mejor nos convenga, y marcaremos la aceptación del acuerdo de la licencia de usuario final y presionamos el botón aceptar.

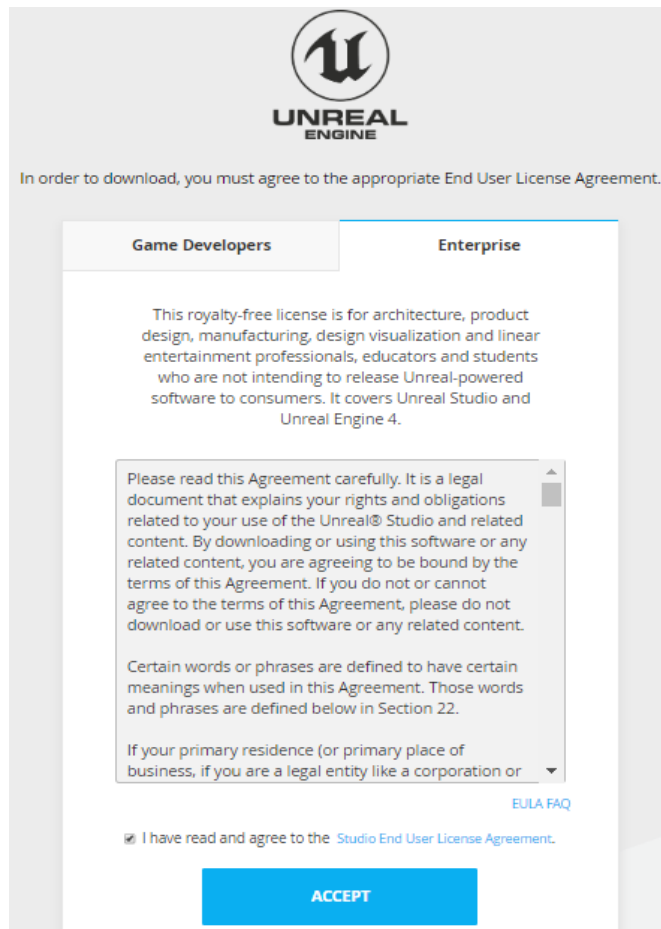


Figura 3: Selección del tipo de licencia y Aceptación del acuerdo del mismo.

4. Aparece la notificación de haber aceptado el Acuerdo de la licencia del usuario final que elegimos y una opción para continuar con la Descarga del entorno de desarrollo al cual le daremos clic.

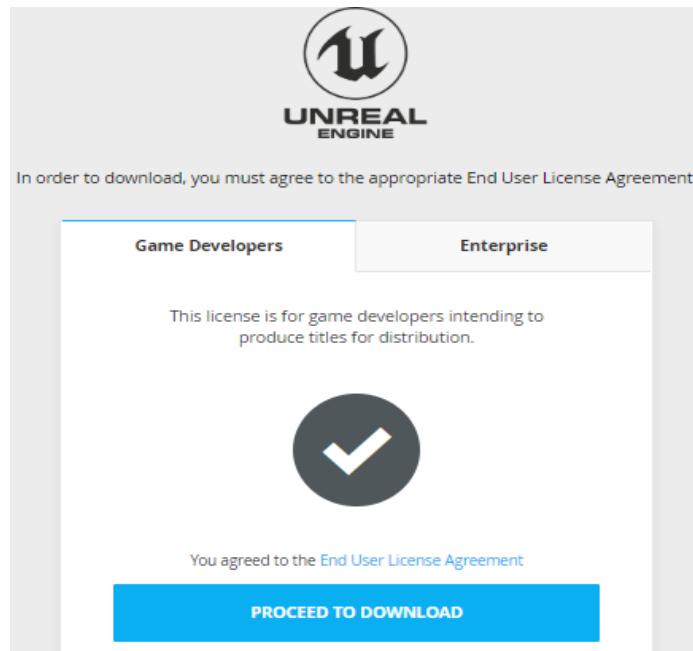


Figura 4: Link para continuar con el proceso de descarga del Epic Games

5. Inmediatamente empieza la descarga del Instalador de Epic Game y un mensaje de agradecimiento por la descarga realizada, cabe mencionar que solo existe el instalador para dos tipos de plataformas que son Windows y Mac, como se muestra a continuación:

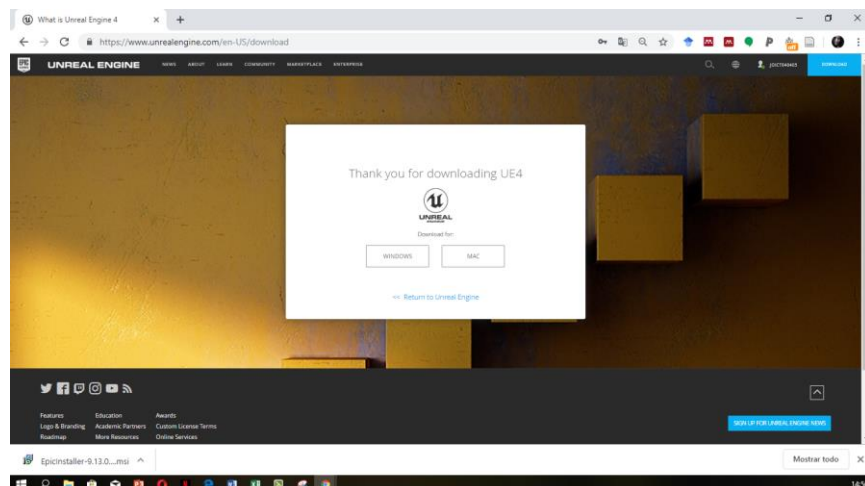


Figura 5: Descarga completa del instalador y plataformas soportadas para la instalación.

6. Una vez descargado el archivo abrimos la ubicación y damos doble clic para ejecutar el mismo, nos aparecerá una ventana que mostrará la carpeta de destino, y seleccionaremos la opción instalar y esperaremos que el proceso de instalación termine.

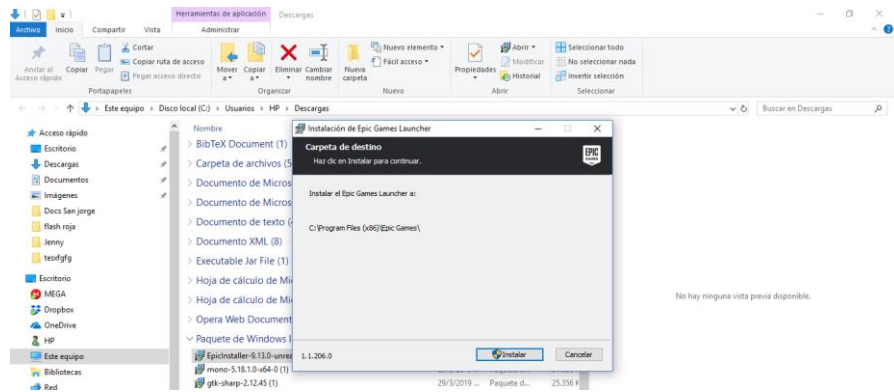


Figura 6: Dirección de la carpeta de destino para la instalación del Epic Games.

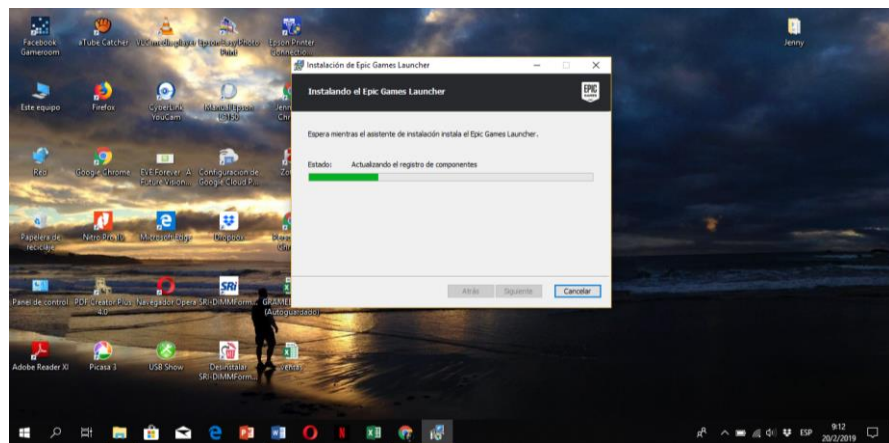


Figura 7: instalación del Epic Games.

7. El sistema nos pedirá que otorguemos permisos al programa sobre redes públicas y privadas, el mismo que es necesario para que el Epic Game funcione debe tener conexión a Internet para descargar el Unreal Editor, y seleccionaremos la opción permitir acceso.



Figura 8: Acceso a Internet.

8. Empieza la descarga de las actualizaciones del software que pesan 132.4 MB y esperaremos hasta que este proceso termine.

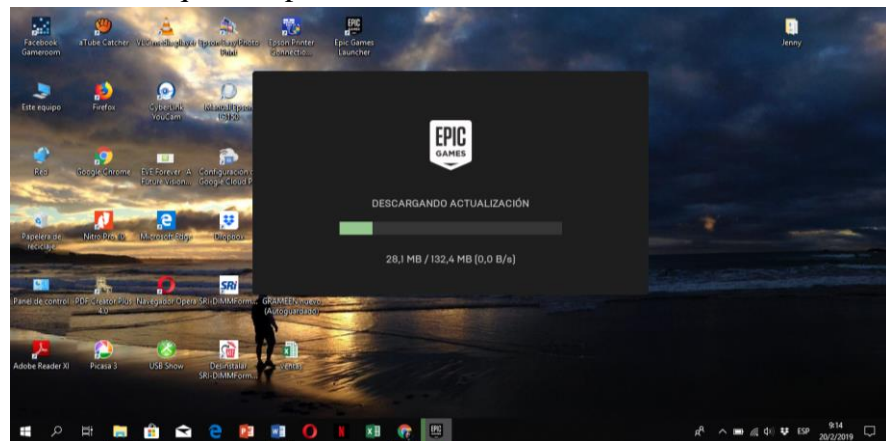


Figura 9: Descarga de Actualizaciones.

9. Al finalizar se nos despliega una nueva ventana que nos pedirá que iniciemos sesión con los datos de la cuenta creada en Epic Games y nos redireccionara al navegador predeterminado.

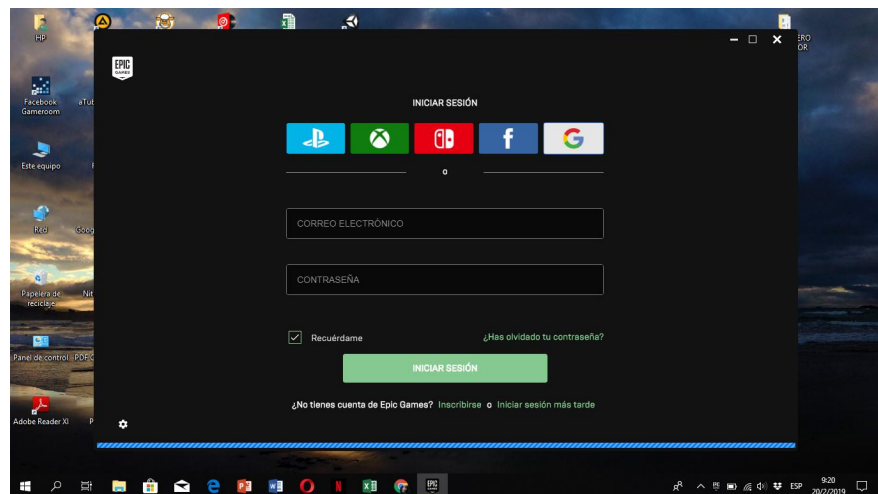


Figura 10: Inicio de Sesión en Epic Games.

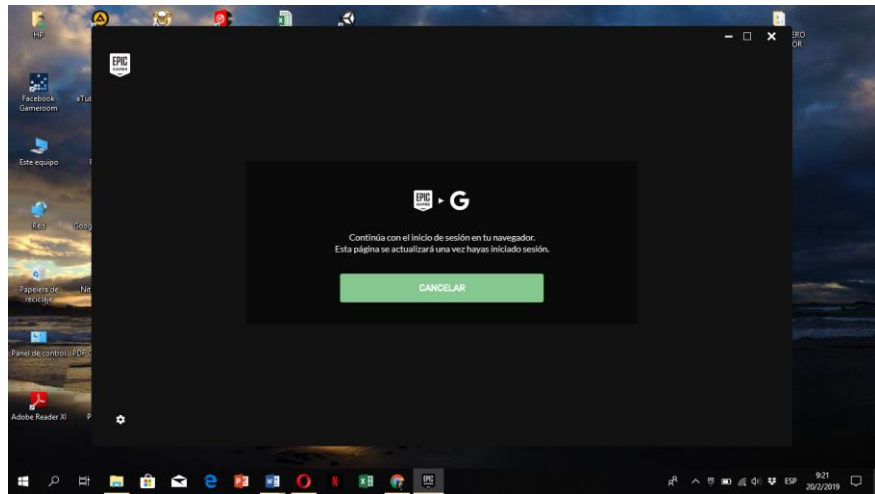


Figura 11: Re direccionamiento al navegador por defecto.

10. Elegimos la cuenta de inicio de sesión y volveremos a la interfaz de Epic Games.

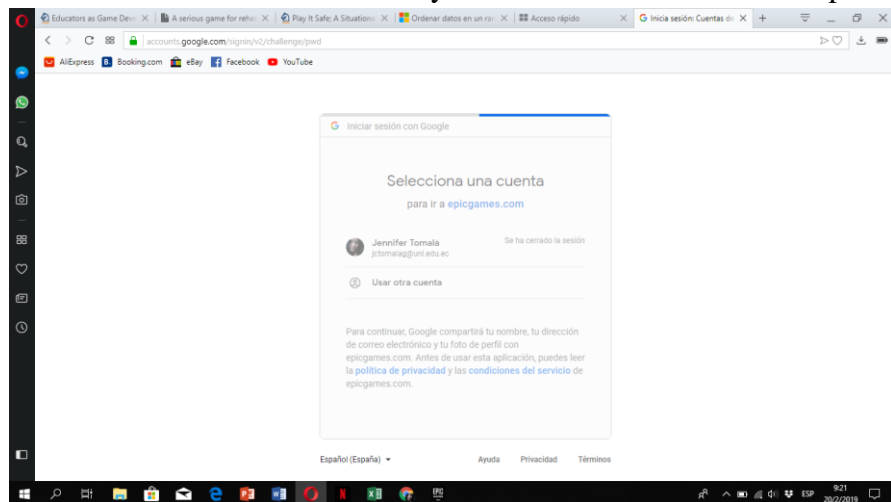


Figura 12: Selección de cuenta para inicio de sesión.

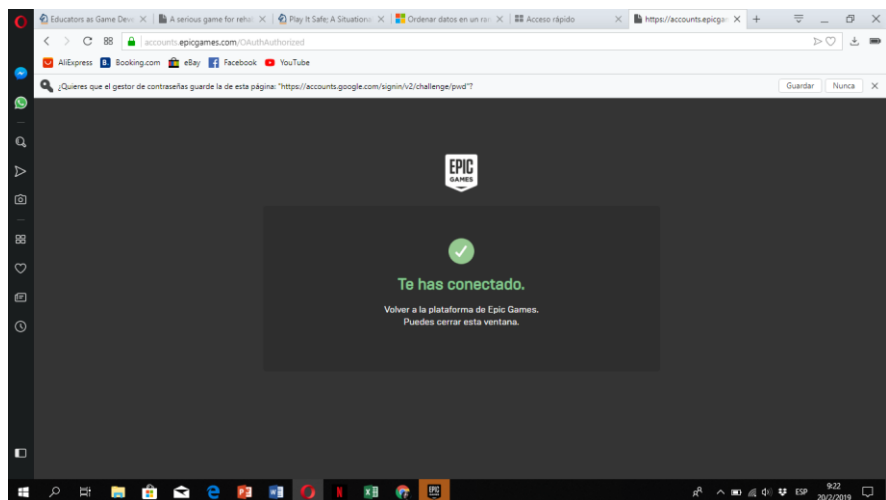


Figura 13: Establecimiento de conexión con el Epic Games.

11. Se actualiza la interfaz del Epic Game y al margen izquierdo encontramos un menú de opciones, seleccionamos la opción Unreal Engine.

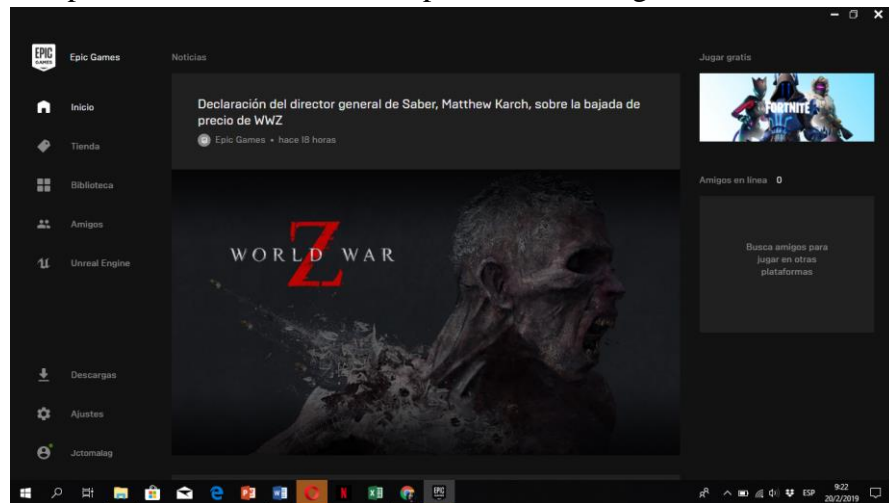


Figura 14: Epic Games iniciado.

12. Seleccionamos el botón amarillo que se encuentra al margen superior derecho e inicializa la descarga del editor.

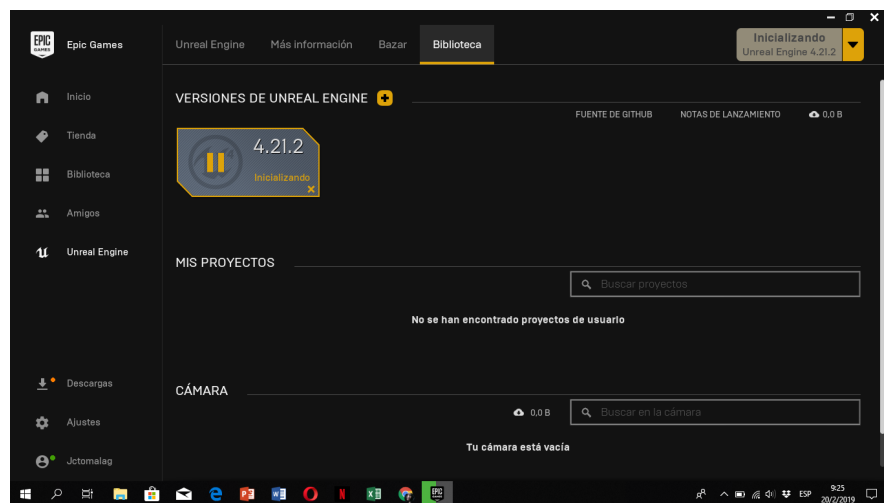


Figura15: Descarga del Editor de Unreal.

13. Al término del mismo presionamos el botón ejecutar e inicializara la carga del editor y procederemos a crear nuestros proyectos.



Figura 16: Inicialización del Editor de Unreal.

ANEXO 4: Certificado de Traducción del Resumen

English Speak Up Center

Nosotros "*English Speak Up Center*"

CERTIFICAMOS que

La traducción del documento adjunto solicitado por la señorita **JENNIFER DEL CARMEN TOMALÁ GONZÁLEZ** con cédula de ciudadanía número **0603951815** cuyo tema de investigación se titula: "**ANÁLISIS COMPARATIVO PARA LA EVALUACIÓN DE MOTORES DE JUEGOS EN EL DESARROLLO DE JUEGOS SERIOS**", ha sido realizada por el Centro Particular de Enseñanza de Idiomas "*English Speak Up Center*".

Esta es una traducción textual del documento adjunto, y el traductor es competente para realizar traducciones.

Loja, 9 de septiembre de 2019


Mgs. Elizabeth Sánchez Burneo

DIRECTORA ACADÉMICA

DIRECCION: SUCRE 207-46 ENTRE AZUAY Y MIGUEL

TELF: 2565842 - 0995263264