

UNIVERSIDAD TÉCNICA DEL NORTE



Facultad de Ingeniería en Ciencias Aplicadas
Carrera de Ingeniería en Sistemas Computacionales

Estudio de dos metodologías para la creación de un videojuego 3D educativo, enfocado en el aprendizaje de Matemáticas dirigido a niños de educación básica.

Trabajo de grado previo a la obtención del título de
Ingeniero en Sistemas Computacionales

Autor:

Sr. Anderson Fernando Aux Culcha

Director:

MSc. Carpio Agapito Pineda Manosalvas

Ibarra – Ecuador

2023



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN

A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO		
CÉDULA DE IDENTIDAD:	0402008965	
APELLIDOS Y NOMBRES:	AUX CULCHA ANDERSON FERNANDO	
DIRECCIÓN:	Av. ESPEJO Y C. 11	
EMAIL:	afauxc@utn.edu.ec	
TELÉFONO FIJO:	N/A	TELÉFONO MÓVIL: 0969123992

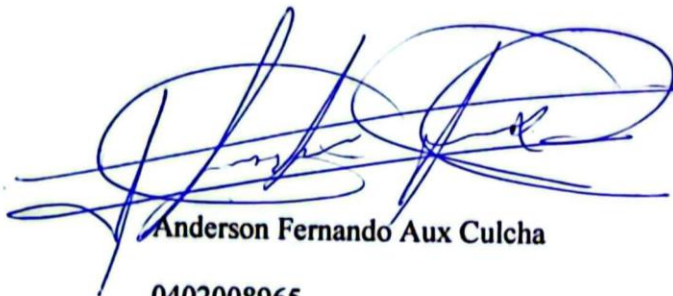
DATOS DE LA OBRA	
TÍTULO:	Estudio de dos metodologías para la creación de un videojuego 3D educativo, enfocado en el aprendizaje de Matemáticas dirigido a niños de educación básica.
AUTOR (ES):	ANDERSON FERNANDO AUX CULCHA
FECHA DE APROBACIÓN: DD/MM/AAAA	02/10/2023
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO
TITULO POR EL QUE OPTA:	INGENIERO EN SISTEMAS COMPUTACIONALES
ASESOR /DIRECTOR:	MSC. CARPIO AGAPITO PINEDA MANOSALVAS

2. CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se desarrolló sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 02 días del mes de octubre de 2023.

EL AUTOR:



Anderson Fernando Aux Culcha

0402008965



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICACIÓN DEL DIRECTOR

Certifico que la Tesis previa a la obtención del título de Ingeniero en Sistemas Computacionales con el tema: **“Estudio de dos metodologías para la creación de un videojuego 3D educativo, enfocado en el aprendizaje de Matemáticas dirigido a niños de educación básica”** ha sido desarrollado y terminado en su totalidad por el Sr. Anderson Fernando Aux Culcha, con cédula de identidad Nro. 0402008965 bajo mi supervisión para lo cual firmo en constancia.

Ing. Msc. Carpio Pineda
DIRECTOR DE TESIS



Ministerio
de Educación



Gobierno
del Ecuador

WILLERMO LARRO
PRESIDENTE



UNIDAD EDUCATIVA "CÉSAR ANTONIO MOSQUERA"

Julio Andrade - Prov. Carchi

Julio Andrade, 19 de septiembre de 2023

El suscrito Rector de la **Unidad Educativa "CÉSAR ANTONIO MOSQUERA"**, MSc. Fernando Cortez, portador de la C.C. 0401203716 a petición de la parte interesada, Señor **AUX CULCHA ANDERSON FERNANDO**, con CI: **0402008965**.

CERTIFICA:

Que, la señor: **AUX CULCHA ANDERSON FERNANDO**, portador de la C.C. **0402008965**, estudiante de la Universidad Técnica del Norte (UTN), en la carrera de Ingeniería en Sistemas Computacionales, ha dado cumplimiento a su Proyecto de Investigación denominado **"ESTUDIO DE DOS METODOLOGÍAS PARA LA CREACIÓN DE UN VIDEOJUEGO 3D EDUCATIVO, ENFOCADO EN EL APRENDIZAJE DE MATEMÁTICAS DIRIGIDO A NIÑOS DE EDUCACIÓN BÁSICA"**, en la Institución de mi regencia, mismo que lo ha culminado satisfactoriamente con todos los requerimientos que la Institución Educativa había solicitado, por lo que se extiende al mencionado estudiante nuestro más sincero agradecimiento por sus aportes brindados que serán de mucha ayuda dentro de la Unidad Educativa, augurándole toda clase de éxitos en su vida profesional.

Lo que certifico en honor a la verdad, facultando al interesado hacer uso del presente en lo que estimare conveniente.

Atentamente,


MSc. Fernando Cortez
RECTOR UECAM



Dirección: Juan Montalvo y el Oro

colegiocesarantoniosmosquera@gmail.com

Teléfono: 593-62-205-540

DEDICATORIA

Este proyecto de investigación va dedicado principalmente a mis padres Adelmo Aux e Ilia Culcha, quienes han sido mi gran inspiración y fuente de apoyo incondicional ante cualquier adversidad, sus sabias palabras y constante motivación me enseñaron a nunca darme por vencido. Gracias a ellos, he aprendido a perseguir mis metas con determinación y que todo es posible gracias a Dios.

Anderson Aux

AGRADECIMIENTOS

Agradezco a Dios por darme la fortaleza necesaria para seguir adelante en los momentos más difíciles, por guiarme por un buen camino y siempre brindarme su protección.

A mis padres, por su apoyo incondicional, por creer en mí y por inculcarme buenos valores desde que era muy pequeño.

A mi tutor MSc. Carpio pineda y opositores, por su valiosa ayuda, consejos, recomendaciones y por compartir su conocimiento durante todo este tiempo. Su apoyo fue fundamental para culminar este trabajo de investigación de manera exitosa.

A mis amigos, por haberme brindado su apoyo tanto en mi vida personal como académica, a lo largo de todos los años que compartimos juntos en la universidad.

Anderson Aux

TABLAS DE CONTENIDOS

INTRODUCCIÓN	XX
Antecedentes	XX
Situación Actual.....	XXI
Prospectiva.....	XXII
Planteamiento del Problema.	XXII
Objetivos	XXIII
Objetivo General.....	XXIII
Objetivos Específicos.....	XXIV
Alcance.	XXIV
Arquitectura.	XXV
Metodología.....	XXV
CAPÍTULO 1	1
1. Marco Teórico	1
1.1. Método	1
1.2. Clasificación de métodos de enseñanza.....	8
1.2.1. Métodos de enseñanza según el grado de participación de los sujetos.....	8
1.2.2. Métodos de enseñanza según el nivel de comprensión.....	8
1.2.3. Los métodos pasivos o reproductivos.....	8
1.2.4. Los métodos activos o productivos.....	9

1.3. Recursos didácticos como apoyo al aprendizaje.....	9
1.4. Videojuegos en la educación.....	9
1.5. Metodologías de desarrollo de software.....	10
1.5.1. Metodologías ágiles	11
1.5.2. Metodologías tradicionales	18
1.5.3. Metodologías ágiles vs metodologías tradicionales.	20
1.6. Metodología SUM y OOADM.....	21
1.6.1. Metodología OOADM.....	21
1.6.2. Metodología SUM.	23
CAPÍTULO 2	25
2.1. Características.....	25
2.1.1. Características de la metodología SUM	25
2.1.2. Características de la metodología OOADM	27
2.2. Ventajas y desventajas.....	29
2.3. Comparación.....	31
2.3.1. Comparación de Metodología SUM y Metodología OOADM por niveles.	31
2.3.2. Métricas para determinar la mejor metodología.	32
2.4. Resultados	33
CAPÍTULO 3.	35
3.1. Lugar de aplicación.....	35

3.2. Materiales e insumos.....	35
3.3. Lenguaje de programación utilizado.....	35
3.4. Unity 3D como motor de videojuegos.....	36
3.5. Metodología SUM modificada.....	37
3.5.1. <i>Desarrollo del concepto (Fase 1)</i>	38
3.5.2. <i>Planificación de la aplicación. (Fase 2)</i>	46
3.5.3. <i>Elaboración. (Fase 3)</i>	50
3.5.4. <i>Beta. (Fase 4)</i>	80
3.5.5. <i>Cierre. (Fase 5)</i>	85
CAPÍTULO 4.	89
4.1. Validación de resultados mediante la ISO 25010	89
4.2. Propósito de la evaluación.....	89
4.3. Módulos de evaluación.....	90
4.3.1. <i>Criterios de decisión para la evaluación de métricas</i>	92
4.3.2. <i>Criterios de decisión para la evaluación general del videojuego</i>	95
4.4. Evaluación por pregunta.....	97
4.4.1. <i>Análisis de los resultados</i>	98
4.5. Evaluación general del test.....	109
4.5.1. <i>Promedios a nivel de subcaracterísticas</i>	109
4.5.2. <i>Promedio final de la característica de usabilidad</i>	111

CONCLUSIONES.....	113
RECOMENDACIONES.	114
BIBLIOGRAFÍA.....	115
ANEXOS.....	121

Índice de figuras

Figura 1. Árbol del problema.....	XXIII
Figura 2. Arquitectura de software para el desarrollo de videojuegos.	XXV
Figura 3. Diagrama de flujo del proceso ingenieril.	XXVII
Figura 4. Metodología XP(eXtreme Programming)	12
Figura 5. Metodología SCRUM	13
Figura 6. Metodología DSDM (Método de Desarrollo Dinámico de Software)	14
Figura 7. Metodología ASD (Desarrollo de software adaptativo).....	15
Figura 8. Metodología Crystal	16
Figura 9. Metodología Feature Driven Development (FDD)	17
Figura 10. Metodología Kanban	18
Figura 11. Metodología waterfall o cascada.....	19
Figura 12. Metodología en Espiral.	20
Figura 13. Estructura de la metodología SUM	27
Figura 14. Estructura de la metodología OOADM.....	29
Figura 15. Metodología de desarrollo para el videojuego.	38
Figura 16. Boceto de objetos internos de la casa.....	42
Figura 17. Modelo de enemigos.	43
Figura 18. Creación de objetos para curación.....	43
Figura 19. Modelo de objetos que contienen información.	44
Figura 20. Modelo de objeto para cambio de color de texto	44
Figura 21. Boceto inicial de las paredes de la casa.....	45
Figura 22. Modelo 2D de las paredes internas de la casa.	45
Figura 23. Modelo 3D de paredes internas de la casa.	46

Figura 24. Cronograma de actividades.	48
Figura 25. Imagen digitalizada del modelo.	52
Figura 26. Ingreso a Unity Asset Store.....	53
Figura 27. Descarga de packs de audios gratuitos.	54
Figura 28. Importación de pack en el proyecto.	54
Figura 29. Modelo estructural del interior de la casa.	56
Figura 30. Eliminación de vértices duplicados.....	57
Figura 31. Creación de escenas.....	59
Figura 32. Ordenar escenas según su prioridad.	59
Figura 33. Botones de navegación.....	60
Figura 34. Métodos para manejo de escenas	61
Figura 35. Métodos para manejo del Audio Manager.	61
Figura 36. Mínimo y máximo del control de volumen.....	62
Figura 37. Imágenes utilizadas en el canvas.....	64
Figura 38. Visualización inicial de imágenes.....	65
Figura 39. Capsula contenedora de cámara y modelo de la pistola.....	67
Figura 40. Modelo de munición, enemigos y generador de vida.....	68
Figura 41. Clase implementada en el jugador.....	69
Figura 42. Clase implementada en la pistola de pelotas de tenis.....	69
Figura 43. Clase implementada en los enemigos.....	69
Figura 44. Llave activada en la UI principal.	71
Figura 45. Llave ubicada en el entorno 3D.....	71
Figura 46. Método para contabilizar objetivos.	72

Figura 47. Animación de apertura de puerta.....	74
Figura 48. Creación del componente collider.....	75
Figura 49. Clase IA.....	76
Figura 50. Quiz en la UI principal.....	77
Figura 51. Clase implementada para el control del quiz.....	78
Figura 52. Promedio de encuestados antes de utilizar el videojuego.....	86
Figura 53. Promedio de encuestados después de la implementación del videojuego.....	87
Figura 54. Porcentaje de mejora en los estudiantes de quinto grado.....	88
Figura 55. Modelo de calidad del producto software según la ISO/IEC 25010.....	89
Figura 56. Escala de Likert adecuada para evaluar niños.....	98
Figura 57. Resultados de la pregunta 1 del test de usabilidad.....	99
Figura 58. Resultados de la pregunta 2 del test de usabilidad.....	100
Figura 59. Resultados de la pregunta 3 del test de usabilidad.....	101
Figura 60. Resultados de la pregunta 4 del test de usabilidad.....	102
Figura 61. Resultados de la pregunta 5 del test de usabilidad.....	103
Figura 62. Resultados de la pregunta 6 del test de usabilidad.....	104
Figura 63. Resultados de la pregunta 7 del test de usabilidad.....	105
Figura 64. Resultados de la pregunta 8 del test de usabilidad.....	106
Figura 65. Resultados de la pregunta 9 del test de usabilidad.....	107
Figura 66. Resultados de la pregunta 10 del test de usabilidad.....	108
Figura 67. Valoración de las subcaracterísticas de usabilidad.....	109
Figura 68. Valoración final de usabilidad del videojuego 3D.....	112

Índice de Tablas.

Tabla 1. Preguntas de investigación.....	2
Tabla 2. Cadena de búsqueda y bases de datos bibliográficas I.	4
Tabla 3. Detalles de artículos de la cadena de búsqueda I.	4
Tabla 4. Cadena de búsqueda y bases de datos bibliográficas II.	6
Tabla 5. Detalles de artículos de la cadena de búsqueda II.....	6
Tabla 6. Diferencia entre metodologías tradicionales y metodologías ágiles.....	21
Tabla 7. Comparativa entre las metodologías investigadas.	24
Tabla 8. Fases de la metodología SUM.	26
Tabla 9. Etapas de la metodología OOHDM.....	28
Tabla 10. Ventajas y desventajas de las metodologías SUM Y OOHMD.....	30
Tabla 11. Comparación según sus niveles.....	31
Tabla 12. Comparación de metodologías según las métricas.....	32
Tabla 13. Materiales e insumos.....	35
Tabla 14. Identificación de necesidades para el videojuego 3D educativo.....	39
Tabla 15. Equipo de trabajo.	47
Tabla 16. Equipo de desarrollo.	47
Tabla 17. Lista de actividades asignada al equipo de desarrollo.	49
Tabla 18. Lista de tareas iteración 1.	51
Tabla 19. Digitalizar elementos necesarios para el videojuego.	52
Tabla 20. Selección de sonidos que incluye el videojuego.....	53
Tabla 21. Lista de tareas iteración 2.	55
Tabla 22. Modelado de objetos 3D.	56
Tabla 23. Lista de tareas iteración 3.....	57

Tabla 24. Creación de escenas y botones.....	58
Tabla 25. Creación y programación de scripts del videojuego.....	60
Tabla 26. Adaptación del “AudioManager”.....	62
Tabla 27. Lista de tareas iteración 4	63
Tabla 28. Visualizar imágenes en la UI.....	63
Tabla 29. Visualización inicial de UI y ubicación principal del player.....	64
Tabla 30. Lista de tareas iteración 5	66
Tabla 31. Incorporar modelos principales en la escena main.....	66
Tabla 32. Control de player, enemigos, objetos proveedores de munición y vida.....	68
Tabla 33. Lista de tareas iteración 6	70
Tabla 34. Implementar llave en la interfaz gráfica.....	71
Tabla 35. Implementar código de control para contabilizar objetivos.....	72
Tabla 36. Lista de tareas iteración 7.....	73
Tabla 37. Animación de puertas.....	74
Tabla 38. Control de colisiones y navegación de enemigos.....	75
Tabla 39. Lista de tareas iteración 8.....	76
Tabla 40. Integrar Quiz en la UI principal.....	77
Tabla 41. Creación del quiz.....	78
Tabla 42. Lista de tareas iteración 9.....	79
Tabla 43. Importación de texturas.....	79
Tabla 44. Control de explosiones.....	80
Tabla 45. Evaluación del verificador beta - Distribución 1.....	81
Tabla 46. Lista de cambios - Distribución 1.....	81

Tabla 47. Evaluación del verificador beta - Distribución 2.	82
Tabla 48. Lista de cambios - Distribución 2.	83
Tabla 49. Evaluación del verificador beta - Distribución 3.	84
Tabla 50. Lista de cambios - Distribución 3.	85
Tabla 51. Escala de calificaciones de acuerdo con MINEDU	86
Tabla 52. Subcaracterísticas de usabilidad a ser evaluadas.	90
Tabla 53. Nivel de importancia para valorar cada subcaracterística de la usabilidad.	90
Tabla 54. Valoración de las subcaracterísticas de usabilidad.	91
Tabla 55. Indicadores para la medición de usabilidad.	92
Tabla 56. Criterios de decisión aplicados para la calificación de métricas.....	93
Tabla 57. Criterios de decisión para evaluar las subcaracterísticas de usabilidad.	93
Tabla 58. Valoración a nivel de subcaracterísticas según la calificación de métricas.....	94
Tabla 59. Ecuación 1 para medir la usabilidad en productos de software.	95
Tabla 60. Criterio de decisión para evaluar la característica de usabilidad.	95
Tabla 61. Tabla de referencia para calcular la valoración final.	96
Tabla 62. Resultados del test por pregunta.	97

RESUMEN

Este trabajo de investigación presenta una manera óptima de mejorar el aprendizaje destinada a niños de educación básica mediante la utilización de un videojuego 3D. La necesidad de este proyecto surge a raíz del bajo nivel de competencias en el área de matemáticas que experimentan los estudiantes de la unidad educativa “César Antonio Mosquera”, atribuida en gran parte a la falta de interés y práctica en esta materia.

Con el propósito de abordar esta problemática, se optó por emplear un videojuego 3D como herramienta didáctica que estimule el deseo de aprendizaje en los estudiantes. Se utiliza esta estrategia como vía efectiva para potenciar el proceso de aprendizaje.

Para determinar que metodología de desarrollo de software es más adecuada para la elaboración del videojuego 3D, se establecieron métricas de comparación entre las metodologías SUM y OOHDM.

Se concluyó que la metodología SUM se adapta de manera más adecuada a este proyecto de investigación. Esto se debe a su facilidad para implementar cambios en cualquier etapa del proceso y a su alta capacidad para mitigar riesgos, lo que contribuye y asegura una buena calidad del mismo.

Finalmente se evaluaron a estudiantes antes y después de la implementación del videojuego 3D, con el fin de cuantificar el impacto que tuvo el videojuego. Mediante el cálculo de mejora porcentual se obtuvo un incremento del 37.52% en la capacidad de resolución de problemas y en el desarrollo de competencias matemáticas, esto respalda la viabilidad del videojuego 3D como herramienta educativa.

PALABRAS CLAVE: Videojuegos 3D, Gamificación, SUM, Unity 3D, Videojuego didáctico.

ABSTRACT

This research work presents an optimal way to improve learning for primary school children by using a 3D video game. The need for this project arises from the low competence in the area of mathematics that students of the educational unit "César Antonio Mosquera" experience, mainly attributed to the lack of interest and practice in this subject.

In order to address this problem, it was decided to use a 3D video game as a teaching tool that stimulates the desire to learn in students. This strategy is used as an effective way to enhance the learning process.

To determine which software development methodology is more appropriate for the creation of the 3D video game, comparison metrics were established between SUM and OOHDM methodologies.

It was concluded that the SUM methodology is the most appropriate for this research project. This is due to its ease to implement changes at any stage of the process and its high capacity to mitigate risks, which contributes and ensures a good quality of the process.

Finally, students were evaluated before and after the implementation of the 3D video game, in order to quantify the impact of the video game. By calculating the percentage improvement, an increase of 37.52% was obtained in the ability to solve problems and in the development of mathematical skills, which supports the viability of the 3D video game as an educational tool.

KEY WORDS: 3D video games, Gamification, SUM, Unity 3D, Didactic video game.

INTRODUCCIÓN

Antecedentes

Según (Gross, 2018a): Ecuador fue partícipe por primera vez en la prueba internacional PISA en el año 2018, encargada de evaluar los sistemas educativos de todo el mundo, este informe generó resultados de los estudiantes ecuatorianos y dio a conocer que los jóvenes poseen graves dificultades al momento de resolver problemas matemáticos debido a: falta de práctica, poco interés durante las clases, falta de solidez en las bases fundamentales y también falta de motivación para enriquecer el conocimiento, dando un total de 70,9% de estudiantes que no alcanzaron el nivel 2 categorizado como el nivel básico.

El informe PISA evidenció los recursos que se invierten en el sistema educativo y los entornos de aprendizaje de escuelas y comunidades, estableciendo un panorama crítico en la educación ecuatoriana debido al uso limitado de las herramientas tecnológicas, como también que el docente imparte sus clases con metodologías no apropiadas para un aprendizaje efectivo, por lo que las clases son monótonas y poco interactivas generando un aprendizaje superficial (Gross, 2018b).

Según (Chiles Bolaños, 2021): A los niños de 5to grado educación básica de la Unidad Educativa “César Antonio Mosquera” se les dificulta el aprendizaje de Matemáticas especialmente cuando las clases son impartidas en la modalidad online, debido a que los niños no asisten a las clases virtuales, no tienen disposición para aprender cosas nuevas y también la falta de compromiso con los estudios, como consecuencia el docente no puede continuar con los temas planificados en el sílabo ni alcanzar el objetivo de enseñanza, ya que los niños no adquieren una base sólida de la asignatura.

Situación Actual

En la actualidad los estudiantes aún presentan inconvenientes a la hora de resolver problemas matemáticos y esto se ve reflejado en las pruebas de ingreso a la universidad por el bajo índice de aciertos en la sección que corresponde a esta asignatura, lo cual influye en un gran porcentaje de la nota final impidiendo que los estudiantes postulen a carreras que desean y esto provoca un aumento de deserción estudiantil y que los jóvenes prefieran conseguir un trabajo y dejar de lado el estudio(Sinchi Nacipucha, 2018).

Es muy importante la interacción entre estudiante-maestro, la didáctica de la matemática es esencial para que los estudiantes aprovechen al máximo las horas de clase y de igual manera los docentes deben buscar metodologías de enseñanza que favorezca la construcción de aprendizaje(Sayas & Gonzales, 2017).

Según Quelal(2021), la enseñanza de Matemática en la Unidad Educativa “César Antonio Mosquera” continúa en estado crítico, afirmación que se verificó a través de una evaluación de diagnóstico en la cual los niños de quinto grado no demostraron tener las competencias necesarias en la mencionada área, debido a que las clases en niveles inferiores fueron impartidas con la modalidad online, se presenta una falta de disposición a la hora de aprender, realizan otras actividades mientras están en el horario de clases, los niños no practican los temas tratados y tienen miedo de participar, como consecuencia se creó un aprendizaje superficial.

También existe una gran dificultad a la hora de resolver problemas matemáticos, porque los padres de familia no permiten que se les exija demasiado a los niños, no incentivan a la práctica de las tablas de multiplicar, no hay un apoyo moral para que los infantes mejoren, tampoco

supervisan la actividad desarrollada por los niños en los horarios de clases y esto provoca que los niños pierdan completamente el interés por los estudios y también que aumente la deserción estudiantil.

Prospectiva.

Para el presente trabajo de titulación se propuso realizar una comparativa entre las metodologías SUM y OOHDM para aplicar la mejor en el desarrollo del videojuego 3D educativo, que ayudará a mejorar el aprendizaje de la matemática en los niños de educación básica, motivándolos a educarse a través de un videojuego logrando así un mayor interés en la asignatura y que de manera progresiva vaya aumentando su lógica Matemática, permitiendo que su interés académico en Matemáticas mejore y que aumente la capacidad de resolver problemas de este tipo.

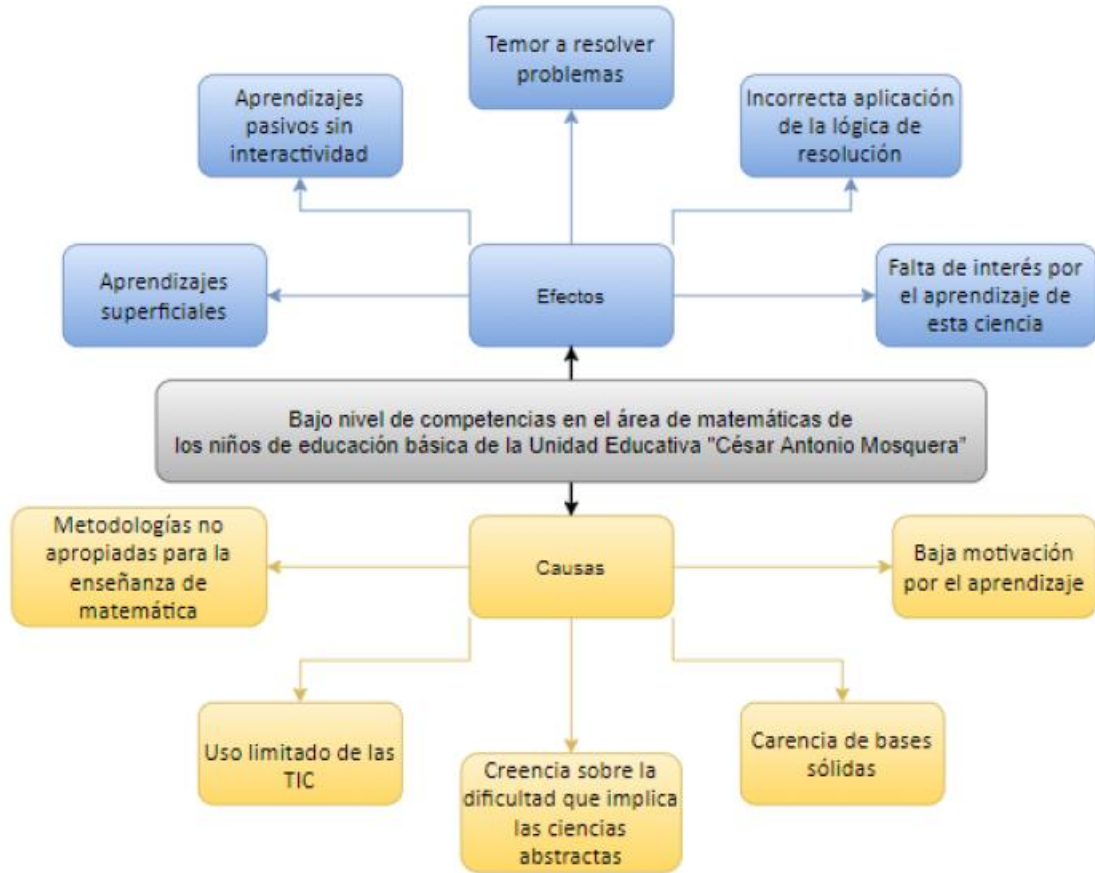
Los videojuegos educativos son especialmente utilizados como herramientas de apoyo en el proceso de enseñanza-aprendizaje en los niños, para aumentar la lógica y el desarrollo psicomotriz. Los videojuegos 3D son los más deseados por los infantes cuando se trata de entretenimiento y ocio, la industria de los videojuegos es una de las más grandes a nivel mundial y la más productiva en el área de la educación (González Sotomayor, 2014).

Planteamiento del Problema.

Establecer métricas de selección que permitan comparar las metodologías SUM y OOHDM, para implementar con ella un videojuego 3D educativo, que permita reforzar las diferentes dificultades que afrontan los niños cuando intentan resolver problemas matemáticos.

Figura 1.

Árbol del problema.



Objetivos

Objetivo General

Estudiar dos metodologías para el desarrollo de un videojuego 3D educativo, enfocado en el aprendizaje de Matemáticas dirigido a niños de educación básica de la Unidad Educativa "César Antonio Mosquera" en la ciudad de Tulcán.

Objetivos Específicos

- Elaborar un marco teórico comparando las metodologías SUM y OOHDM para el desarrollo del videojuego 3D.
- Establecer las métricas de selección para definir la mejor metodología y utilizarla en el desarrollo del videojuego.
- Desarrollar un videojuego 3D utilizando la metodología que fortalezca el aprendizaje de Matemáticas en niños de 5to grado de Educación Básica de la Unidad Educativa “César Antonio Mosquera”.
- Validar el videojuego 3D basado en la ISO/IEC 25010 con las diferentes subcaracterísticas de usabilidad.

Alcance.

Para el presente trabajo de titulación se desarrolla un videojuego 3D educativo en el motor de desarrollo de videojuegos multiplataforma Unity, el cual permite organizar y estructurar las características funcionales básicas de movilidad de un personaje mientras recorre el área de juego (casa), el videojuego cuenta con un top 10 de mejores jugadores que se posicionan de acuerdo a su puntaje, también se programaron animaciones y diversas mecánicas de objetos, finalmente se incorporaron operaciones de lógica matemática que servirán como llave para abrirse camino por la casa.

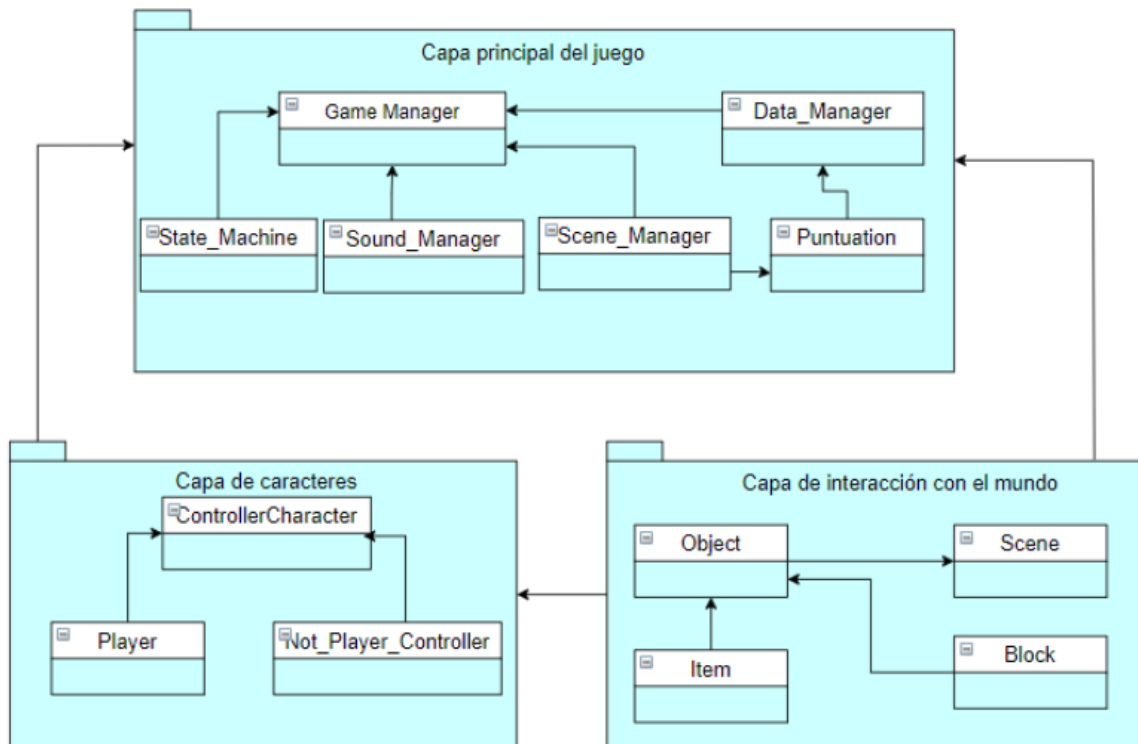
Para realizar el videojuego se identificaron los patrones de diseño para construir su arquitectura, la cual se basó en la unión de dos tipos de arquitecturas: en capas y componentes para obtener un prototipo funcional de un videojuego del género educativo enfocado en el aprendizaje de Matemáticas para los estudiantes de la Unidad Educativa. Finalmente se realizó un análisis con la norma ISO/IEC 25010 de las subcaracterísticas de usabilidad.

Arquitectura.

El juego consta con una arquitectura de software basada en capas y componentes para estructurar todos los elementos de un videojuego, permitiendo que los componentes de las capas tengan una buena comunicación con los componentes de las otras capas a través de las clases e interfaces(Páez et al., 2018a).

Figura 2.

Arquitectura de software para el desarrollo de videojuegos.



Fuente: (Páez et al., 2018b).

Metodología

En el presente trabajo de titulación se usó el método ingenieril como metodología para lograr un cambio radical aplicando las metodologías actuales, y también se planteó que la

estrategia esté asociada a una técnica o procedimiento práctico para la resolución de problemas (Markelov, 2016).

Para el objetivo uno en el cual se elaboró un marco teórico comparando las dos metodologías se procedió a realizar una búsqueda de conceptos en diversas fuentes bibliográficas como tesis, artículos y revistas, luego se definió el problema a resolver, ¿cuál es la mejor metodología?, después se seleccionó el software para el modelado y diseño del videojuego 3D y se prosiguió con la planificación del prototipado.

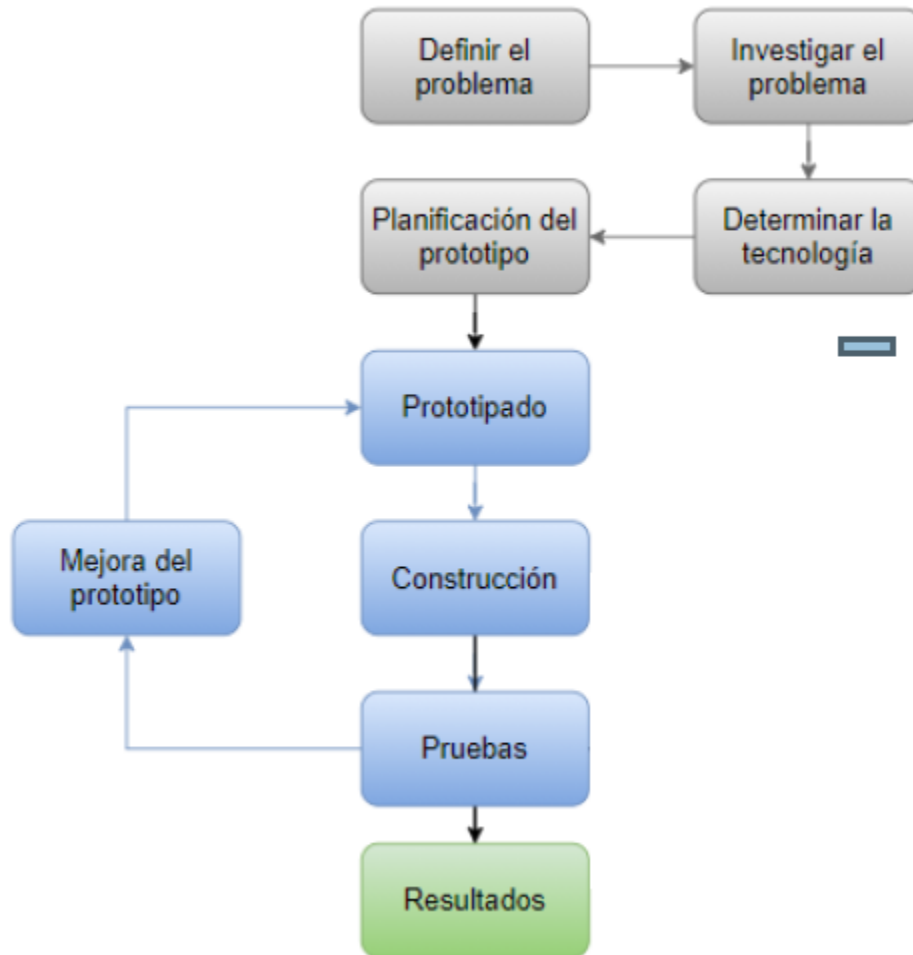
Para el cumplimiento del objetivo dos se establecieron métricas de selección que ayudaron a elegir la mejor metodología y así implementarla en el desarrollo del videojuego 3D.

Para el cumplimiento del objetivo tres se desarrolló el videojuego educativo como herramienta didáctica para fortalecer el aprendizaje de Matemáticas en los niños de la Unidad Educativa empezando por el primer prototipo del videojuego 3D, se prosiguió con su construcción y la implementación de mecánicas del juego como también de los diferentes escenarios a ser implementados.

Para cumplir el objetivo cuatro se necesitó que el prototipo de videojuego 3D esté finalizado y sin ningún tipo de falla para validarlo con la ISO/IEC 25010 en las diferentes subcaracterísticas de usabilidad.

Figura 3.

Diagrama de flujo del proceso ingenieril.



Fuente: (Villadiego, 2018).

CAPÍTULO 1

1. Marco Teórico

1.1. Método

Para el presente trabajo de titulación se optó por utilizar como método de investigación la realización de revisiones sistemáticas de literatura o SLR, la cual se enfoca en la recopilar y analizar diferentes estudios e investigaciones a través de un proceso sistemático.(García-Peñalvo, 2021)

1.1.1. *Revisión Sistemática de literatura (SLR).*

La metodología de revisión sistemática de literatura se basa en una revisión bibliográfica en artículos científicos y bases de datos bibliográficas como también se ha convertido en una metodología importante en el estudio de la ingeniería de software. Al aplicar este enfoque, se involucran aspectos importantes para lograr un adecuado diseño e implementación, así como una estrategia de búsqueda efectiva.(Carrizo et al., 2018)

Cuenta con los siguientes pasos:

I. Definición para la búsqueda:

- Definir la pregunta de investigación.
- El alcance de la revisión.
- Determinar los criterios de inclusión.
- La cadena de búsqueda.

II. Ejecución de la búsqueda:

- Definir la selección de trabajos

III. Discusión de los resultados:

- Esquemas de caracterización.
- Analizar los resultados.

Desarrollo:

I. Definición para la búsqueda:

- Definir las preguntas de investigación.

Tabla 1.

Preguntas de investigación.

Nro.	Preguntas de Investigación	Motivación
1	¿Cuál es la contribución de los videojuegos como método de aprendizaje?	Puntualizar cómo los videojuegos aportan en el aprendizaje de los estudiantes.
2	¿Cuáles son las estrategias pedagógicas para impartir una educación de calidad?	Investigar y conocer las diferentes estrategias pedagógicas más usadas por los docentes.
3	¿Cuál es la importancia del uso de herramientas didácticas en la enseñanza?	Conocer la importancia de las herramientas didácticas como apoyo en la enseñanza.
4	¿Cuál es el impacto de los videojuegos educativos como material didáctico?	Conocer la influencia que tienen los videojuegos en la educación.
5	¿Qué metodologías del desarrollo de software son adecuadas para el desarrollo de videojuego?	Conocer las metodologías apropiadas para desarrollar videojuegos.

- **El alcance de la revisión.**

El alcance de la revisión se enfoca en una investigación exploratoria en la cual se mapea un tema en específico, se pueden determinar los conceptos clave, las teorías y fuentes bibliográficas tomando en cuenta los diferentes filtros como la fecha que empieza desde el 2017 y se enfoca en el área de Ingeniería, Informática, Matemáticas. Las bases de datos bibliográficas que se utilizaron son: Science Direct, Springer Link, Google Academic, SCIELO.

- **Determinar los criterios de inclusión.**

1. Solo artículos que fueron publicados entre el 2017 y 2023
2. Solo artículos orientados a la educación, material pedagógico que incluya metodologías de enseñanza y aprendizaje.
3. Solo artículos encaminados al desarrollo de videojuegos, metodologías de desarrollo de software.
4. Solo artículos que incluyan métricas de evaluación de software para establecer una comparación entre las dos metodologías a ser estudiadas.

- **Cadenas de búsqueda.**

1. (Videogames or 3D Environment) And (Education OR pedagogy)
2. Methodology AND (Software or Videogames 3D)

II. Ejecución de la búsqueda.

- Definir la selección de trabajos

Tabla 2.*Cadena de búsqueda y bases de datos bibliográficas I.*

Base de datos	Fase I	Fase II	Fase III
Science Direct	118	27	3
Springer Link	83	74	2
Google Académico	1240	22	4
SCIELO	962	40	3

Nota: Cadena de búsqueda I (Videogames or 3D Environment) And (Education OR pedagogy)

Tabla 3.*Detalles de artículos de la cadena de búsqueda I.*

Código	Título	Autor
A01	Peace education and inclusive reasoning. Critical thinking in philosophy for children	Sara Mariscal Vega
A02	Study of the opinions of the students of the Primary Education Degree on the use of videogames as a didactic resource through a qualitative analysis	Luis Manuel Soto-Ardila, Lina Melo Niño, Ana Caballero, Ricardo Luengo.
A03	Curricular Proposal on a Blended Approach and an Informative Multimedia for Didactic Resources for the Teaching of English Course	Silvia Saborío-Taylor
A04	Design and assessment of an open repository of educational resources for the teaching of recent history.	Diego Miguel Revilla, María Sánchez Agustí, Jesús María Moro Bengoechea
A05	The impact of video games on Students' educational outcomes	Slobodan Adžić, Jarrah Al-Mansour, Hasnain Naqvi, Slobodan Stambolić,
A06	The act of playing and the logical and mathematical reasoning in digital games: The mathematical experience in the digital games	Cristiano Natal Tonéis,

A07	Learning Styles and Approaches to Studying	N.Entwistle, E.Peterson
A08	Cómo hacer una Systematic Literature Review (SLR)	García-Peñalvo, F. J.
A09	Indicadores de participación de los estudiantes en una metodología activa	Fidalgo Blanco Ángel, Sein-Echaluze Lacleta María Luisa, García-Peñalvo Francisco José.
A10	Videojuegos como recurso educativo digital para el desarrollo de las dimensiones básicas integrales en los estudiantes de sexto grado	Guido Brun Cantero, Rafaela Cardona Romero.
A11	Stimulating children's engagement with an educational serious videogame using Lean UX co-design	María C. Ramos-Vega, Victor M. Palma-Morales, Diana Pérez- Marín, Javier M. Moguerza.
A12	Designing social videogames for educational uses.	Carina González-González, Francisco Blanco-izquierdo,

Tabla 4.*Cadena de búsqueda y bases de datos bibliográficas II.*

Base de datos	Fase I	Fase II	Fase III
Science Direct	171	97	4
Springer Link	184	34	2
Google Académico	357	50	5
SCIELO	21	9	4

Nota: Cadena de búsqueda II: Methodology AND Development AND (Software or Videogames 3D)

Tabla 5.*Detalles de artículos de la cadena de búsqueda II.*

Código	Título	Autor
A01	The Relation between Software Development Methodologies and Factors Affecting Software Reliability	V. Yakovyna, M. Seniv and I. Symets
A02	Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software.	Vite Cevallos Harry, Montero Kelvin, Cuesta Jefferson
A03	Analysis of Software Development Methodologies.	SoobiaSaeed, NZ Jhanjhi, Mehmood Naqvi and Mamoona Humayun.
A04	A survey of agile software development methodologies in Ecuador	R. E. Quelal, M. Villavicencio and L. E. Mendoza
A05	Considerations about the efficiency and sufficiency of the utilization of the Scrum methodology: A survey for analyzing results for development teams	MarceloMorandini, Thiago AdrianoColetib, EdsonOliveiraJr, Pedro Luiz Pizzigatti Corr�ea.

A06	Acute and long-lasting cortical thickness changes following intensive first-person action videogame practice	Davide Momi, Carmelo Smeralda, Giulia Sprugnoli, Salvatore Ferrone, Simone Rossi, Alessandro Rossi, Giorgio Di Lorenzo, Emiliano Santarnecchi
A07	Inside Agile Family: Software Development Methodologies	Rakesh Kumar, Priti Maheshwary, Timothy Malche
A08	Adopting Agile Software Development Methodologies in Big Data Projects – a Systematic Literature Review of Experience Reports.	I. Krasteva and S. Ilieva
A09	Diseño de un e-portafolio basado en la metodología OOHDM para gestionar la información de proyectos en informática.	Díaz Pulido, Jose Arturo
A10	Análisis de metodologías de diseño de interfaz humano-computador en la página de aprendizaje en línea de la puce-e	Cedeño Cedeño Karol Ivette
A11	Learning Process of Agile Scrum Methodology With Lego Blocks in Interactive Academic Games: Viewpoint of Students.	D. A. Barcelos Bica and C. A. G. d. Silva
A12	Desarrollo de un Serious Gaming para estudiantes de Primer Año de Primaria aplicando la Metodología SUM	Kevin Maximiliano Arenas Cancapa
A13	Implementación de la metodología SUM modificada para el desarrollo de videojuegos orientados al aprendizaje en Bolivia	Murillo Sanchez Xavier, Rocha Andres, Ibañez Illanes Alan, Quiroz Perez Jorge, Sahonero Guillermo, Palacios Fabio.
A14	Agile Software Development: Methodologies and Trends.	Alsaqqa, S., Sawalha, S., & Abdel-Nabi, H
A15	Gender, videogames, and navigation in virtual space.	Suzanne de Castell, Hector Larios, Jennifer Jenson,

1.2. Clasificación de métodos de enseñanza.

Los métodos de enseñanza se clasifican según el grado de participación, nivel de comprensión o por el rol que desempeña el docente en su aplicación.

1.2.1. Métodos de enseñanza según el grado de participación de los sujetos.

Las estrategias de enseñanza más usadas, se denominan tipos básicos de enseñanza sistemática, se basan en la participación del alumno o del profesor con el tema tratado.

Las técnicas de enseñanza expositiva se diferencian de las demás en que la participación del profesor siempre tiene prioridad sobre la de los alumnos, que sólo responden ocasionalmente. Este enfoque se emplea sobre todo en las clases magistrales.

1.2.2. Métodos de enseñanza según el nivel de comprensión.

El método de enseñanza según el nivel de dominio de los contenidos didácticos o el nivel de asimilación de conocimientos y habilidades, se basa en el desarrollo de las actividades cognitivas de los estudiantes y por ende encaminado al proceso de aprendizaje como también corresponde a los objetivos pedagógicos que van a ser alcanzados, sus niveles asimilación son: familiarización, reproducción, producción o aplicación y creación, estos métodos se dividen en: pasivos o reproductivos y activos o productivos. (Gabriel & Ibarra, 2018)

1.2.3. Los métodos pasivos o reproductivos.

Esta clase de métodos se identifican porque la recepción del conocimiento es de manera pasiva y esto limita el aprendizaje debido a que es unidireccional y genera conflicto al momento de ser reproducido; es decir, que los docentes imparten conocimiento que es comprensible por los estudiantes, pero no son capaces de compartirlo con alguien más. Estos métodos se enfocan a la repetición de los contenidos ya existentes y a proseguir una serie de pasos sin visión al futuro (Vega, 2018).

1.2.4. Los métodos activos o productivos.

La metodología que permite a los estudiantes participar de manera activa en su aprendizaje se denomina activa o productiva; posee muchas ventajas y también apoya el desarrollo de capacidades cognitivas incentivando la creatividad. Otro punto para resaltar es que el aprendizaje se construye a partir de otro ya existente logrando así lo que se denomina creación del conocimiento(Fidalgo Blanco et al., 2019).

1.3. Recursos didácticos como apoyo al aprendizaje.

Actualmente la tecnología facilita considerablemente las investigaciones y el fácil acceso a la información, dando paso a nuevas tecnologías de la información y comunicación(TIC), estas tecnologías lograron un cambio significativo en las actividades diarias de todas las personas sin dejar de lado a la educación, no solo surgen como un medio de soporte en el entorno educativo sino que constituyen un factor especial en el cambio de las bases y principios de los procesos de enseñanza y aprendizaje(Saborío-Taylor & Saborío-Taylor, 2019).

La correcta implantación de la tecnología en el aula, a pesar de las dificultades que conlleva, se ha convertido en uno de los principales objetivos de la innovación educativa, porque el uso de la tecnología no se limita a una mera presencia en la formación, sino que tiene un significado pedagógico consistente y ayuda a los profesores a adaptarse a una nueva realidad (Revilla Diego Miguel et al., 2018).

1.4. Videojuegos en la educación.

Según (Soto-Ardila et al., 2019) plantea que los videojuegos pueden presentar un sinnúmero de posibilidades como recurso para la enseñanza de contenidos educativos, facilitando a los docentes impartir las clases de forma no tradicional, convirtiendo sus clases en un entorno

interactivo y emocionante orientando a que los estudiantes pueden despertar un mayor interés en aprender.

Según (Adžić et al., 2021) considera que el uso de videojuegos se ha convertido en un espacio fundamental en la vida de los niños y adolescentes debido a que los videojuegos se aplican en el área financiera, social, así como también en la educativa.

Autores como (Momi et al., 2018) evidencian que una experiencia en un entorno de videojuegos puede tener un impacto positivo en el enriquecimiento cognitivo y perceptivo, es importante resaltar que los cambios estructurales en el cerebro se basan en el tipo de videojuegos y la experiencia del usuario como jugador.

El siglo XX dio paso a la implementación de los videojuegos en el entorno educativo, creando así una versión tecnológica de los juegos tradicionales, provocando una revolución en el material didáctico afectando positivamente la forma de impartir clases en todas las instituciones educativas, cada día la tecnología da un paso hacia el futuro en este caso los videojuegos pueden aumentar el interés educativo en los estudiantes porque mientras exploran el videojuego refuerzan los conocimientos adquiridos en clase, la industria de los videojuegos ha ido creciendo constantemente con el pasar de los años y ha sido un complemento para las nuevas tecnologías de información y comunicación(TIC). (Cantero Guido Brun, 2018)

1.5. Metodologías de desarrollo de software.

El desarrollo de software siempre ha sido debatido debido a los problemas que se presentan para su finalización y existen claros ejemplos como: Los sistemas no cumplen las expectativas de usuarios, varios bugs o fallos suceden al ejecutar el programa, los costos elevados de software difíciles de prever que casi siempre superan las estimaciones, otro punto a destacar es el tiempo de entrega de una modificación al software que puede ser entregado fuera del tiempo previsto y con

menos prestaciones de las que se planteó inicialmente, todo ello debido a que esta tarea puede ser una tarea difícil y costosa; para enfrentar este tipo de circunstancias existen diversos métodos y metodologías que son utilizados como herramientas de apoyo para el desarrollo de software(Delgado & Díaz, 2021).

La calidad de software no es solo la satisfacción del cliente, también se enfoca en su velocidad, flexibilidad, seguridad, usabilidad, estabilidad, escalabilidad entre muchos otros aspectos más, por lo que para hablar de calidad de software se crearon metodologías que proponen como objetivo principal, aplicar un conjunto de técnicas o metodologías ágiles y/o tradicionales que permitan un modelado de software de calidad(Vite et al., 2018).

1.5.1. Metodologías ágiles

En la última década ha sido necesario adoptar la mayoría de las metodologías ágiles para el desarrollo de software y así generar software de calidad, debido a las diversas ventajas que proveen estas metodologías y han sido adoptadas en sistemas de seguridad, proyectos a gran escala, desarrollo distribuido, entre otros (Krasteva & Ilieva, 2020).

El término ágil ha cambiado la manera de desarrollar y preservar los sistemas de software. Las metodologías ágiles más populares son: eXtreme Programming (XP), Scrum, Dynamic System Development Method (DSDM), Adaptive Software Development (ASD), Crystal, Feature Driven Development (FDD) y Kanban. Cada una tiene diferentes arquitecturas y principios, pero su objetivo principal es: Reducir el tiempo de ciclo de desarrollo, garantizar un producto de calidad, tener mejor conocimiento de los requisitos como también una mayor flexibilidad(Quelal et al., 2018).

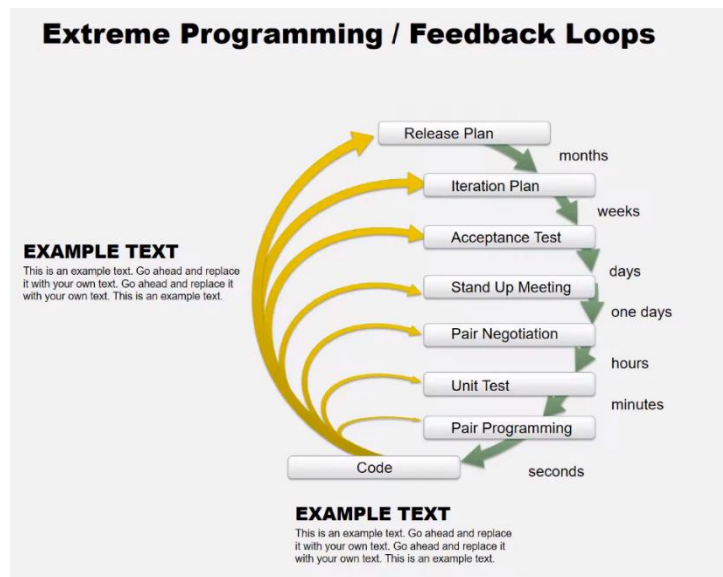
1.5.1.1. eXtreme Programming (XP).

Según (Saeed et al., 2019a) plantean que este método divide el proceso de desarrollo de software en pequeñas partes con el propósito de gestionarlas nuevamente en el proceso original, al dividir las especificaciones se reduce el costo de cambiar todo el programa y realizar las actividades durante el proceso de elaboración del software.

Según (Kumar et al., 2019) manifiestan que el trabajo en equipo de desarrolladores ha sido conocido como XP (Extreme Programming). La comunicación, la sencillez, la retroalimentación, el respeto y el valor han sido utilizados para mejorar el proyecto de software. XP es la capacidad de respuesta hacia el cliente dinámico y la mejora para un software de calidad. La adopción de los requisitos del cliente mejora la productividad.

Figura 4.

Metodología XP(eXtreme Programming)



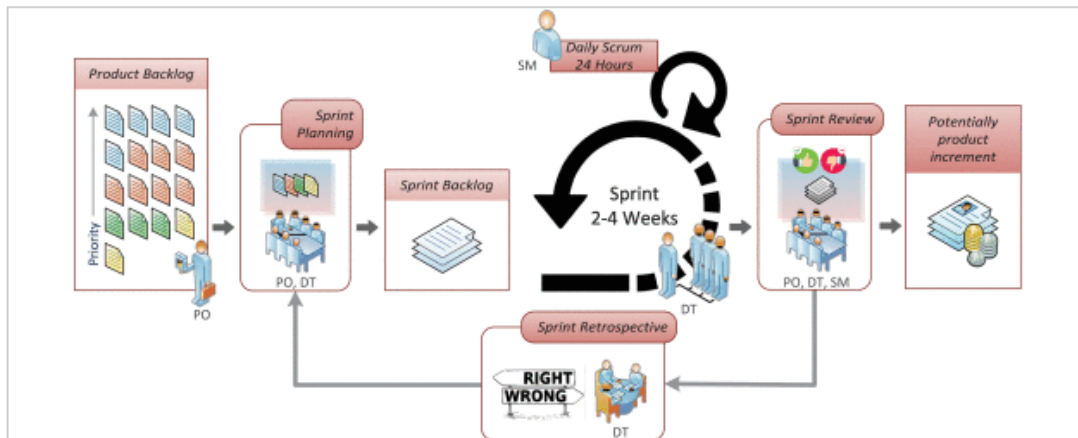
Fuente:(Saeed et al., 2019a)

1.5.1.2. Scrum

SCRUM es una metodología ágil habitualmente utilizada en el desarrollo de software, se basa en sprints que comienzan con una planificación diaria para lograr terminar el software en el periodo establecido al inicio y garantizar un producto de calidad; sin embargo, esto no significa que no existan dificultades en su proceso, los desarrolladores pueden presentar problemas a menudo cuando intentan implementar las mejores soluciones para el proyecto y esto se refleja en un tiempo más prolongado de desarrollo(Morandini et al., 2021).

Figura 5.

Metodología SCRUM



Fuente:(Bica et al., 2020).

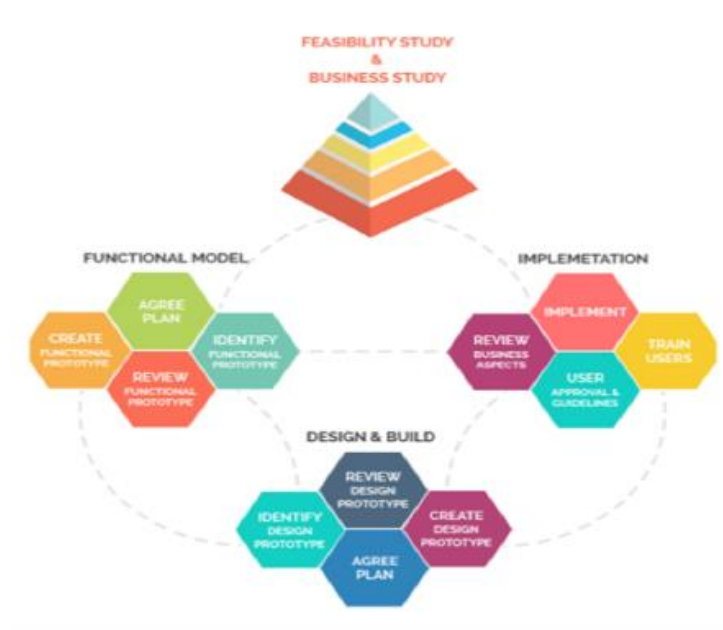
1.5.1.3. Dynamic System Development Method (DSDM).

DSDM (Método de Desarrollo Dinámico de Software) en el desarrollo de proyectos ágil es un marco de trabajo que mira tanto hacia adelante como hacia atrás. DSDM se centra en la entrega rápida del producto junto con una metodología de guía para controlar el proceso simultáneamente. La técnica que se utiliza para la priorización de los requisitos se asigna sobre la base de "Debería tener", "Podría tener" y "Quiere tener".

Cuando un proyecto se desarrolla con DSDM, el estudio de viabilidad y de negocio deben realizarse de forma secuencial. La decisión del terreno para el resto del proyecto es realizada mediante dos fases. Las tres fases finales son incrementales y de naturaleza iterativa. En realidad, el desarrollo se realiza en el proyecto durante estas fases.(Kumar et al., 2019).

Figura 6.

Metodología DSDM (Método de Desarrollo Dinámico de Software)



Fuente:(Saeed et al., 2019a).

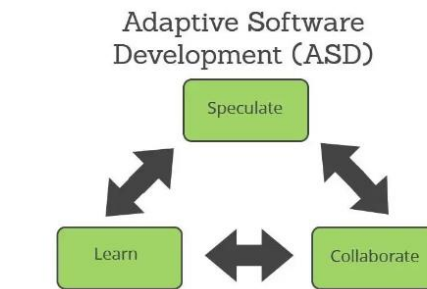
1.5.1.4. Adaptive Software Development (ASD).

El desarrollo de software adaptativo es una metodología rápida para adoptar nuevas técnicas desarrolladas por James A. Highsmith III como una respuesta a una economía que está cambiando progresivamente y evolucionando, esta filosofía de gestión se centra en la adaptación en lugar de optimización, problemas en el desarrollo de complejos y grandes sistemas y rápida creación y evolución de software. La realización de un proyecto ASD sigue tres fases, que son especular, colaborar y aprender.(Merzouk et al., 2018a)

Según (Kumar et al., 2019) la metodología ASD (Desarrollo de software adaptativo) establece que puede ser un requisito impreciso cuando se de inicio a un proyecto de comercio electrónico. Por lo que el establecimiento de la misión del proyecto, manejo e identificación requisitos, objetivos, cuadro de tiempo del proyecto que está adaptado a los requisitos del conjunto de características, estimaciones, alcance y la disponibilidad de recursos se hace por especulación.

Figura 7.

Metodología ASD (Desarrollo de software adaptativo)



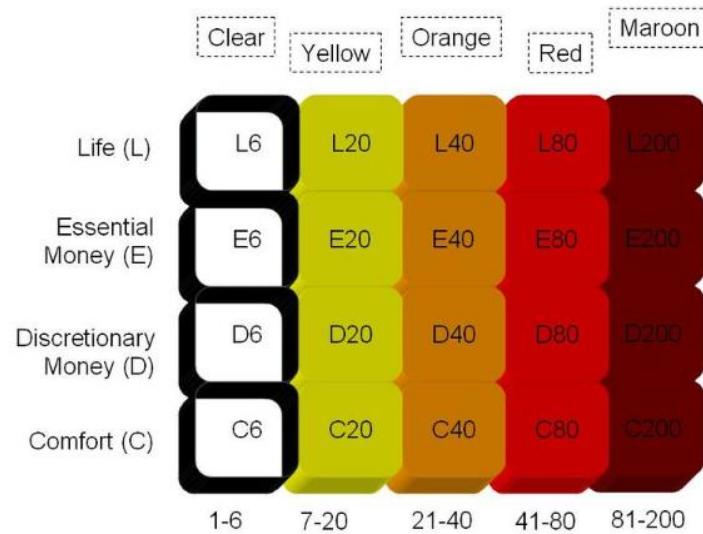
Fuente:(Kumar et al., 2019).

1.5.1.5. Crystal.

La metodología de Crystal es un conjunto de métodos y cada uno se diferencia por su color ordenado en opacidad ascendente que da lugar a la "La elección" de un método determinado que se pueda adaptar a un proyecto específico y esto se hace eligiendo el color en función del tamaño y la criticidad del proyecto. Algo a destacar es el indispensable uso de la documentación causado por la dificultad de gestionar el proceso a través de la comunicación cara a cara(Merzouk et al., 2018a).

Figura 8.

Metodología Crystal



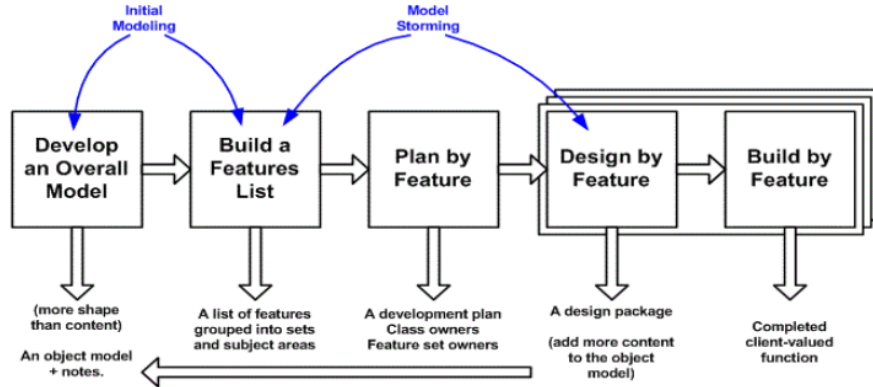
Fuente:(Kumar et al., 2019).

1.5.1.6. Feature Driven Development (FDD).

Este enfoque se centra en hechos que se proyectan en la funcionalidad del proyecto de software. Todas las características de esta metodología son comprensibles y solicitan requisitos para emprendedores de negocios reales, esto significa la descripción del verdadero valor del propósito. Durante el proyecto, el cliente y los equipos de desarrollo interactúan constantemente para obtener comentarios del propietario del proyecto cuando configura los ajustes de la aplicación. Esta metodología es adecuada para pequeñas, medianas y grandes firmas (Saeed et al., 2019b).

Figura 9.

Metodología Feature Driven Development (FDD)



Fuente:(Kumar et al., 2019).

1.5.1.7. Kanban.

La metodología Kanban se centra en el sistema de control de cadena de suministro, en lugar de un sistema de control de inventario que es usado en la fabricación. Kanban fue desarrollado para mejorar el rendimiento del sistema y mantener la producción de alto nivel, ganando popularidad como herramienta de respaldo cuando se necesita implementar sistemas de producción, esto demuestra que Kanban es una excelente manera de impulsar la innovación en el área de fabricación(Kumar et al., 2019).

Figura 10.

Metodología Kanban



Fuente: (Kumar et al., 2019).

1.5.2. Metodologías tradicionales

Las metodologías tradicionales se basan principalmente en la documentación de todo el proceso de desarrollo de software, se realiza una planificación global y se plantean los requisitos al inicio del proyecto para luego dar paso al ciclo del desarrollo de software, este tipo de metodologías no se adaptan fácilmente a los cambios debido a que son poco flexibles (Vite et al., 2018).

Este conjunto de metodologías se enfocan directamente en el trabajo sobre su proceso, con el fin de elaborar software eficiente y de calidad, debido a esto las metodologías hacen hincapié en la planificación, detallando todo el proceso a ser realizado y continuar con el control del proceso mediante la asignación de roles, de modo que esto permita su posterior modelado y respectiva documentación, es preciso resaltar que este tipo de metodologías no se ajustan a cambios por lo

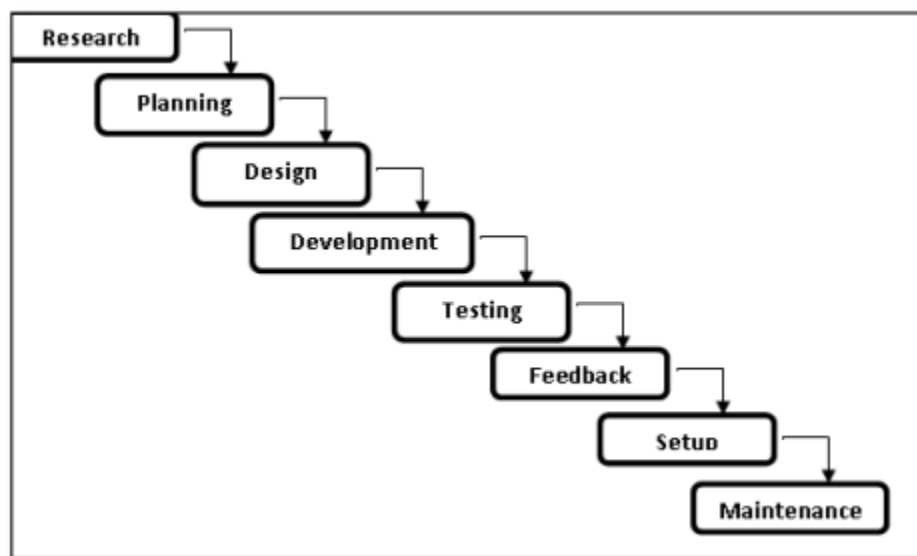
tanto no se recomienda que sean implementadas en entornos variables debido a sus estrictos procedimientos.(Mesa, 2017)

1.5.2.1. Cascada o Waterfall

El modelo en cascada o waterfall es considerado una de las metodologías de desarrollo de software más antiguas. Esta metodología se centra en las diferentes etapas del ciclo de vida de software como son: análisis, diseño, desarrollo, prueba, lanzamiento y soporte. Para dar paso a una nueva etapa, la anterior debe ser terminada por completo, se caracteriza por que en esta metodología se evalúa calidad en cada etapa, pero esto no puede garantizar que todos los proyectos sean implementados de manera secuencial debido a esto es una metodología obsoleta(Yakovyna et al., 2020).

Figura 11.

Metodología waterfall o cascada



Fuente:(Saeed et al., 2019a).

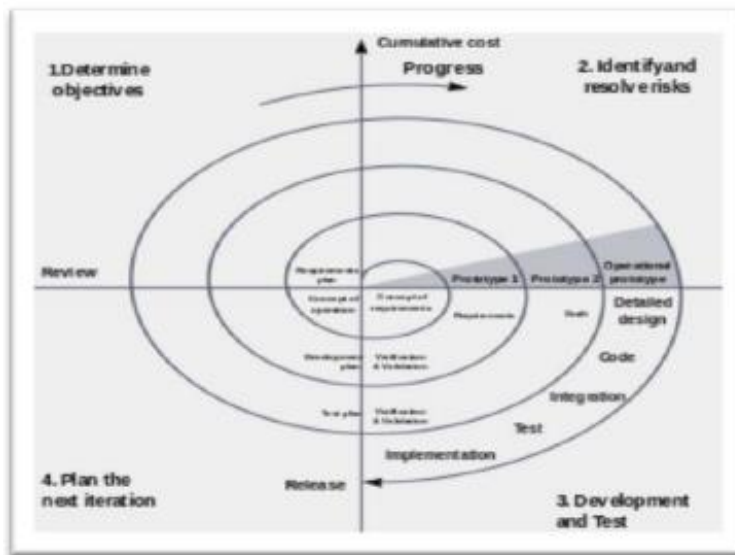
1.5.2.2. Modelo Espiral.

Esta metodología se centra en establecer objetivos y analizar otras opciones útiles para los proyectos mejor documentados. La metodología de la espiral tiene cuatro etapas: planificación, análisis de riesgos, desarrollo y evaluación. El proyecto sigue varias veces cada etapa hasta llegar a la etapa final en la que la configuración del programa se lleva a cabo en un entorno real.

La fase de análisis de riesgos utiliza varias opciones antes de continuar con el programa. Tras la finalización de la primera repetición, los propietarios del proyecto reciben comentarios. Esta metodología es adecuada para proyectos que necesitan identificar riesgos y también es adecuada para proyectos medianos y grandes proyectos.

Figura 12.

Metodología en Espiral.



Fuente:(Saeed et al., 2019a)

1.5.3. Metodologías ágiles vs metodologías tradicionales.

Ante todo, las metodologías de desarrollo tradicionales se basan en una jerarquía rígida de documentación detallada sobre todo el proceso de elaboración, se hace énfasis en toda la

planificación para dar paso al ciclo de desarrollo de software; a diferencia de las metodologías de desarrollo ágiles que se distinguen por enfocarse en el desarrollo del proyecto, buscando así un equilibrio entre el trabajo y el proceso(Cevallos et al., 2018).

Tabla 6.

Diferencia entre metodologías tradicionales y metodologías ágiles

Parámetros	Metodologías tradicionales	Metodologías ágiles
Fácil de modificar	Difícil	Fácil
Enfoque de desarrollo	Predictivo	Adaptable
Orientación del desarrollo	Orientado al proceso	Orientado al cliente
Tamaño del proyecto	Amplio	Pequeño o mediano
Escala de planificación	Equipo numeroso	A corto plazo
Estilo de gestión	Organizado y controlado	Liderazgo y colaboración
Aprendizaje	Aprendizaje y desarrollo evolutivo	El aprendizaje es secundario al desarrollo
Documentación	Alta	Bajo
Tipo de organización	Altos ingresos	Ingresos moderados y bajos
Número de empleados de la organización	Amplio	Pequeño
Presupuesto	Alto	Bajo
Número de equipos	Múltiples	Uno
Tamaño del equipo	Mediano	Pequeño

Fuente:(Al-Saqqa et al., 2020).

1.6. Metodología SUM y OOHDM.

1.6.1. Metodología OOHDM.

OOHDM (Object-Oriented Hypermedia Design Methodology) es una metodología que integra diversos tipos de medios, como imágenes, textos, videos y objetos interactivos. Esta metodología busca una representación precisa y estructurada de toda la información que se emplea

en las aplicaciones porque debe adaptarse a los requisitos del usuario, entornos de trabajo y tiempo de entrega, teniendo en cuenta posibles cambios en cada modelo(Díaz Pulido, 2017).

Según (González et al., 2014): La metodología OOHDM se utiliza para representar las características de aplicaciones hipermedia, se busca visualizar los diagramas de manera amplia y exacta para que logre contribuir a un diseño más efectivo en aplicaciones hipermedia.

La aplicación de esta metodología puede implicar mayores costos de diseño, recursos y esfuerzo, pero este enfoque puede proporcionar ventajas significativas al proyecto como: menor tiempo de desarrollo, facilitar su evolución y mantenimiento(Chamkopbi Quenta, n.d.).

La metodología OOHDM proporciona una documentación exhaustiva en cada etapa de desarrollo, esto ayuda a que exista un registro claro en las decisiones realizadas, por otro lado esta metodología dificulta la escalabilidad en proyectos de gran tamaño, ya que puede presentar fallos en la coordinación del desarrollo(Díaz Pulido, 2017).

En sus inicios esta metodología estaba orientada a una automatización excesiva, debido a esto los desarrolladores no tomaban en cuenta detalles fundamentales, esto implica que para garantizar la calidad y éxito del desarrollo de aplicaciones hipermedia debe existir un equilibrio entre automatización y optimización de errores(Chamkopbi Quenta, n.d.).

Puntos fuertes de OOHDM:

- Orientada a objetos
- Diseño hipermedia interactivo
- Reutilización de código
- Documentación muy detallada

1.6.2. Metodología SUM.

La elaboración de videojuegos es un proceso difícil y complejo debido a que depende de la colaboración de múltiples personas e incluso de diferentes áreas de trabajo. La programación es un componente esencial para este tipo de proyectos y es habitual que los equipos de trabajo utilicen metodologías ágiles que se centren en el desarrollo para agilizar el proceso dando poca prioridad a la documentación(Sánchez et al., 2018).

El desarrollo de videojuegos no está sujeto a una metodología específica, pero hay diversos procesos que facilitan el trabajo de equipo, Es habitual que las entidades encargadas del desarrollo de los proyectos empleen métodos diferentes debido a la diversidad de especificaciones y requerimientos que tiene cada uno, para el desarrollo de videojuegos SUM es una buena opción debido a su alta detección de errores, flexibilidad y adaptación(Molero et al., 2019).

Según (Jaramillo et al., 2020) Afirma que la metodología SUM para videojuegos se enfoca en crear videojuegos de alta calidad en el menor tiempo posible y tiene como propósito producir resultados predecibles sin dejar de lado la gestión de riesgos, corrección de errores y optimización de recursos.

El enfoque de SUM es basado en SCRUM porque adopta sus prácticas y principios como lo son la agilidad, adaptabilidad, colaboración y una pronta respuesta a cambios, todo esto se debe a que SUM se centra en el proceso y en las personas, asignando roles y tareas a cada miembro del equipo de desarrollo con el fin de entregar software de alta calidad de forma rápida y asequible(Arenas, 2018).

Puntos fuertes de SUM:

- Centrado en el proceso y personas
- Alta mitigación de riesgos

- Flexibilidad y adaptabilidad
- Pronta respuesta a cambios
- Gestión efectiva de tareas

Tabla 7.

Comparativa entre las metodologías investigadas.

Métodos / Criterios	SCRUM	DSDM	LeanSD	KANBAN	CRYSTAL FAMILY	AUP	OOHDM	SUM
Tamaño del equipo	5-9	2-10	-	-	Todos	-	1-4	3-7
Longitud de la iteración (semana)	4	-	-	1	Proyecto Específico	1-2	4-6	2-4
Roles y responsabilidades	✓	✓	X	✓	X	✓	X	✓
Centrado en el proceso	X	✓	✓	✓	X	X	✓	✓
Centrado en las personas	✓	X	X	X	✓	✓	X	✓
Apoyo al equipo virtual	✓	X	X	✓	✓	-	X	X
Mitigación de riesgos altos	✓	✓	X	X	✓	✓	X	✓
Mitigación de riesgos medios	X	X	✓	✓	X	X	✓	✓
Documentación	Básica	Existe	X	X	Importante	Básica	Rigurosa	Importante
Reunión diaria	✓	X	X	✓	X	X	X	✓
Intercambio de información (mediante documentos)	X	✓	X	X	X	-	✓	✓
Reuniones cara a cara	X	X	✓	X	✓	X	X	X
Basado en modelos	X	X	X	X	X	X	✓	X

Fuente: (Arenas, 2018; Cedeño, 2020; Merzouk et al., 2018).

CAPÍTULO 2

2.1. Características.

2.1.1. *Características de la metodología SUM*

La metodología SUM al ser ágil puede ser un sistema atractivo y transformador, dependiendo del nivel de compromiso de los involucrados en el proyecto, se debe considerar las características para su aplicación(Sánchez et al., 2018).

- Flexibilidad para ser combinada con otras metodologías de desarrollo para adaptarse a distintas realidades
- Desarrollar videojuegos de calidad en tiempo y costo asequible
- Mejora progresiva del proceso para el incremento de la eficiencia
- Adaptación a equipos multidisciplinarios pequeños
- Marco de referencia en la elaboración de proyectos

2.1.1.1. *Fases De la metodología SUM*

Según (Molero et al., 2019), SUM fue desarrollada con el fin de lograr adaptarse a proyectos orientados a videojuegos académicos y tiene como objetivo principal mitigar costos y tiempos de desarrollo. (Sánchez et al., 2018) enfatiza que SUM destaca por su flexibilidad y su fácil adaptación a los requerimientos del cliente. Además, su alta gestión de riesgos permite abordar y prevenir posibles errores e imprevistos a lo largo del ciclo de vida del proyecto, esta metodología tiene cinco fases detalladas en la Tabla 8.

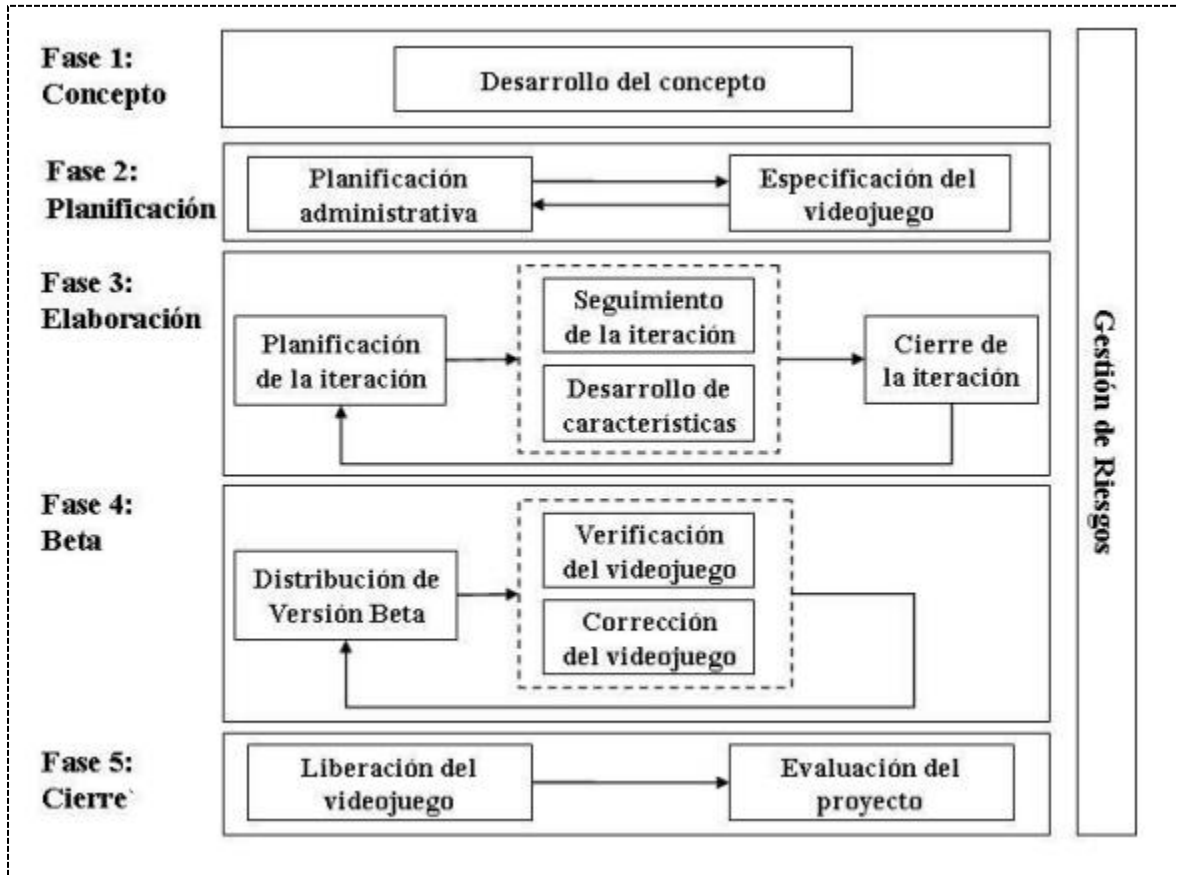
Tabla 8.*Fases de la metodología SUM.*

Concepto	En esta fase se detalla cada una de las cualidades del videojuego, también se considera todo lo que se espera construir desde una perspectiva amplia. Se enfoca en puntos como el gameplay, tipo de videojuego, historia y ambientación.
Planificación	En esta etapa se describe la planificación administrativa del videojuego, se considera alcance, presupuesto y herramientas necesarias para su desarrollo. Además, se establecen costos, plazos y roles del equipo de desarrollo para cumplir con el cronograma establecido.
Elaboración	Esta fase es la más importante de la metodología, ya que en esta etapa se establecen los objetivos del desarrollo, se asignan tareas y se controla el avance de acuerdo con el cronograma establecido en la fase anterior. El avance se da por iteraciones, esto posibilita realizar ajustes en el proyecto ante cambios no previstos y mejorar su calidad.
Beta	Esta etapa es una versión de prueba del videojuego, aquí se testean las mecánicas de juego y se verifica que el gameplay sea entretenido para el jugador; además se toma en cuenta el rendimiento en todos los aspectos, se aplican pruebas de jugabilidad para evaluar si existen errores o bugs y se retorna a la fase 3 para realizar las correcciones necesarias.
Cierre	Al finalizar la fase beta, se libera el videojuego al público objetivo y se procede a la evaluación de problemas, éxito y cumplimiento de objetivos, luego se realiza una retroalimentación con el fin de obtener información para futuras mejoras.

Fuente: (Jaramillo et al., 2020).

Figura 13.

Estructura de la metodología SUM



Fuente:(Arenas, 2018).

2.1.2. Características de la metodología 00HDM

Esta metodología propone un conjunto de tareas que pueden tener un alto costo a corto plazo, mientras que a mediano y largo plazo los tiempos se reducen considerablemente ya que al reutilizar el diseño simplifica el costo de evolución y mantenimiento, se debe considerar sus características(González et al., 2016).

- Desarrollo de aplicaciones hipermedia, con orientación a objetos en un diagrama estandarizado para clases

- Posee 4 etapas, donde cada etapa define un esquema de objeto específico donde se introducen nuevos elementos o clases
- Uso de abstracciones apropiadas para organizar el espacio de navegación.
- Promueve el uso de prototipos para validar y mejorar el proyecto.

2.1.2.1. Etapas de la metodología OOHDM

Según (Díaz Pulido, 2017), la metodología OOHDM es orientada a modelos dedicados al desarrollo de aplicaciones hipermedia, esta metodología utiliza mecanismos de abstracción que permiten una descripción concisa de los detalles necesarios para crear software, lo cual facilita la implementación de soluciones.

Esta metodología consta de cinco etapas que se detallan en la siguiente **Tabla 9**:

Tabla 9.

Etapas de la metodología OOHDM

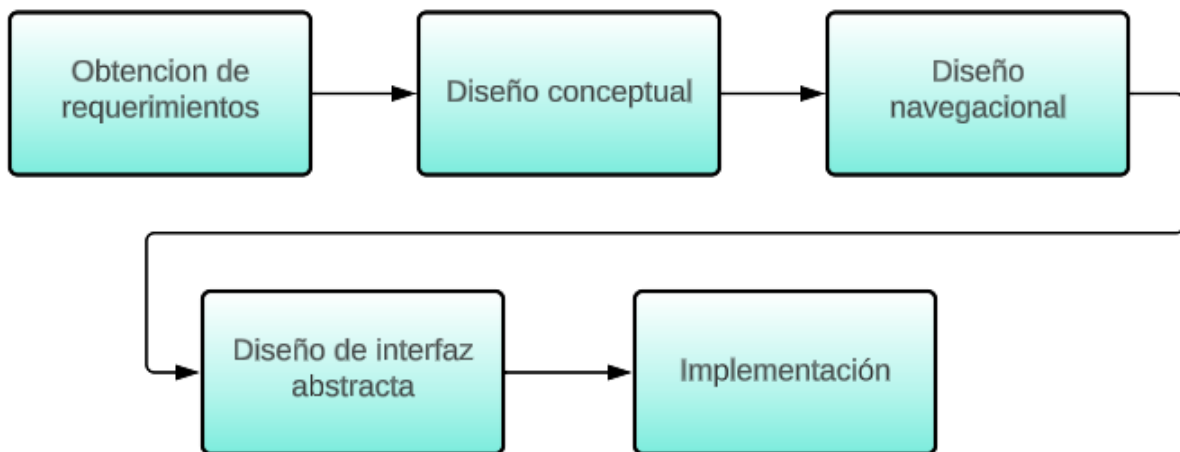
Obtención de requerimientos	En esta etapa se recolectan cuidadosamente todos los requerimientos del cliente, se identifican actores, tareas y escenarios. Se representa de manera visual por medio de diagramas de interacción para establecer una base sólida para el desarrollo del sistema.
Diseño conceptual	Se construye el modelo conceptual utilizando diagramas de clases que representen el dominio de la aplicación; el enfoque se centra en las interrelaciones de las clases y los subsistemas, aquí no se toman en cuenta los actores y las tareas, solo se elabora la representación del modelo UML.
Diseño navegacional	En esta etapa se elabora la estructura de navegación, la cual permite que el usuario pueda realizar acciones por medio de enlaces, vínculos o índices. Todo se orienta a sus necesidades y a la navegación dentro de la aplicación hipermedia.

Diseño de interfaz abstracta	En esta etapa se define como deben aparecer los objetos navegacionales y cuál va a ser su función dentro de la aplicación hipermedia, esto aborda el diseño estructural y de comportamiento con el fin de garantizar una interfaz intuitiva y funcional para los que darán uso a la aplicación.
Implementación	En la fase final de esta metodología se tiene en cuenta el entorno donde será implementada la aplicación, aquí los usuarios probarán la aplicación y evaluarán su desempeño, con el fin de detectar posibles errores y que los desarrolladores puedan mejorar la aplicación.

Fuente: (Cedeño, 2020).

Figura 14.

Estructura de la metodología OOHDM.



Fuente:(Cedeño, 2020).

2.2. Ventajas y desventajas.

La propuesta de gestión, aplicación y objetivo del proyecto marca las ventajas y desventajas de las metodologías a comparar:

Tabla 10.*Ventajas y desventajas de las metodologías SUM Y OOHDM*

	Ventajas	Desventajas
SUM	<ul style="list-style-type: none"> • Ágil, flexible, adaptable en el desarrollo a distintos requerimientos en constante evolución. • Permite obtener resultados predecibles. • Maneja de manera competente los recursos y riesgos del proyecto • Revisión temprana del producto, facilita la detección de problemas o posibles cambios a realizarse. • Plantea un sistema de planificación en las actividades que presentan prioridad frente a aquella que no poseen una mayor demanda, dentro del ciclo de vida del proyecto • El cliente es una parte importante del equipo ya que tiene conocimiento del desarrollo. 	<ul style="list-style-type: none"> • Demanda absoluta en la definición de las tareas y sus plazos • Realización de varios eventos de forma simultánea dificulta el análisis en el final de cada fase. • Dificultad en mantener el nivel de colaboración por parte de los colaboradores. • Ambigüedad en los objetivos, dificulta el desarrollo del proceso. • Dificultad de que el cliente esté presente de forma constante.
OOHDM	<ul style="list-style-type: none"> • Independencia entre lo conceptual, la navegación y lo visual facilita el mantenimiento de la aplicación • Ofrece la posibilidad de reusabilidad de diseño, simplificación en la evolución y mantenimiento. • Representación de objetos a través de diagramas estandarizados. • Utiliza herramientas diagramáticas que facilitan la representación de manera más clara y precisa. • Cada etapa de la metodología considera al usuario como integrante fundamental en la validación del producto. 	<ul style="list-style-type: none"> • Funcionalidad del sistema, al momento de plantear la metodología de desarrollo que tenga en cuenta la navegación y la interfaz. • No posee un mecanismo para tareas multidisciplinarias con varios actores. • La simplificación de las tareas en cada fase puede desencadenar el abandono de detalles fundamentales por parte del desarrollador. • Elevado costo para mantenimiento y adaptación de cambios. • La etapa de diseño navegacional es una de las más complejas ya que conlleva al desarrollo de una gran cantidad de diagramas, la redundancia de información se podría evitar graficando la información en un solo tipo de diagrama.

Fuente: (Molina et al., 2018; Torres et al., n.d.).

2.3. Comparación

2.3.1. Comparación de Metodología SUM y Metodología OOHDm por niveles.

Existen varias metodologías que son utilizadas en distintas aplicaciones, en este caso se consideran dos metodologías reconocidas, considerando criterios específicos que se detallan más adelante, en primera instancia como se puede observar en la **Tabla 11** se comparan las dos metodologías respecto de sus niveles (Cedeño, 2020).

Tabla 11.

Comparación según sus niveles

Metodología	Nivel de Concepto	Nivel Estructural	Nivel Visible
SUM	Definición del concepto <ul style="list-style-type: none"> • Público objetivo • Modelo de negocios • Características • Personajes e historia • Definición del ambiente de desarrollo • Lenguaje de programación • Herramientas de desarrollo 	Planificación de fases <ul style="list-style-type: none"> • Cronograma • Conformación del equipo • Definición de presupuesto • Identificación de funciones primordiales • Evaluación del producto • Desarrollo de características del producto • Beta • Evaluación y ajustes del prototipo • Cierre • Entrega de la versión final 	<ul style="list-style-type: none"> • Presentación del producto al potencial cliente
OOHDm	Obtención de requerimientos <ul style="list-style-type: none"> • Clases • Perspectivas • Relación orientada a objetos 	Diagramación <ul style="list-style-type: none"> • Enlaces • Clases navegacionales • Contextos navegacionales 	<ul style="list-style-type: none"> • ADV • En contextos • Modelos secuenciales

Fuente: (Ayala, 2017; Sánchez et al., 2018).

En la siguiente **Tabla 12** se detalla las métricas de evaluación para ambas metodologías y su respectiva comparación:

2.3.2. Métricas para determinar la mejor metodología.

Tabla 12.

Comparación de metodologías según las métricas

Métricas	SUM	OOHDM
Flexibilidad	<ul style="list-style-type: none"> Se basa en enfoques ágiles como Scrum, esto proporciona la adaptabilidad para hacer frente a posibles cambios durante el desarrollo del software. 	<ul style="list-style-type: none"> Su enfoque se centra más en el desarrollo debido a esto sí es necesario realizar cambios se debe ajustar el modelo conceptual, lo que aumenta costos y recursos.
Enfoque	<ul style="list-style-type: none"> Tiene un enfoque iterativo que va escalando en todo el ciclo de desarrollo de software, se centra en la entrega temprana de productos de calidad. 	<ul style="list-style-type: none"> Se centra en la creación de modelos de diseño para representar el comportamiento de la aplicación hipermedia, se basa en modelos y la reutilización de componentes.
Defectos	<ul style="list-style-type: none"> Dificultad de acoplamiento por parte del equipo desarrollador. 	<ul style="list-style-type: none"> La simplificación de tareas puede provocar la pérdida de detalles importantes.
Costo de desarrollo	<ul style="list-style-type: none"> El uso de esta metodología puede ser rentable debido a que prioriza el desarrollo de funciones principales, esto permite entregar resultados funcionales de manera temprana con el fin de reducir costos. 	<ul style="list-style-type: none"> En contraste, esta metodología requiere una estructura y navegación más compleja, esto ocasiona un esfuerzo adicional en términos de diseño y modelado lo cual provoca aumento en costos asociados a la planificación inicial.
Productividad	<ul style="list-style-type: none"> Esta metodología fomenta la colaboración y el trabajo en equipo, lo cual ayuda a mejorar la productividad y la eficiencia en el desarrollo de software. 	<ul style="list-style-type: none"> Por otra parte, la productividad de esta metodología dependerá de la familiaridad, experiencia y una planificación sólida para mejorar la productividad.

Calidad	<ul style="list-style-type: none"> Referente a esta metodología, destaca por su enfoque en la entrega iterativa, con el fin de obtener una retroalimentación rápida por parte de los usuarios, esto a su vez facilita la detección temprana de errores y una mejora significativa en el desarrollo, con el propósito de entregar un producto final de mayor calidad. 	<ul style="list-style-type: none"> En cuanto a esta metodología se enfoca en el modelado conceptual y en la arquitectura del sistema hipermedia, la calidad del software dependerá de cómo se implementen los principios de diseño orientado a objetos y de la implementación en la fase final.
Gestión de riesgos	<ul style="list-style-type: none"> SUM realiza la gestión de riesgos de manera iterativa por medio de entregas al finalizar cada iteración, esto permite una temprana identificación de errores y su respectiva mitigación ajustándose a cambios imprevistos. 	<ul style="list-style-type: none"> Debido a su enfoque centrado en el diseño y la arquitectura del sistema hipermedia, la mitigación de riesgos puede requerir un gasto adicional de tiempo y recursos.
Ciclo de iteración	<ul style="list-style-type: none"> De 2 a 4 semanas 	<ul style="list-style-type: none"> De 4 a 6 semanas.

Fuente: (Corzo Durand, 2021; Torres et al., n.d.).

2.4. Resultados

En conclusión, al realizar el análisis cualitativo se dedujo que la metodología más adecuada para desarrollar el videojuego 3D educativo que fortalezca el aprendizaje de matemáticas en niños de educación básica, es la metodología SUM. Debido a estas razones:

- SUM se enfoca en entregas iterativas y enfatiza la adaptación a cambios inesperados.
- Es una metodología orientada a videojuegos lo cual facilita la implementación de funcionalidades.
- Las entregas son incrementales, se prioriza la entrega rápida de prototipos funcionales.
- El uso de esta metodología es rentable debido a que los cambios se realizan después de cada iteración, esto evita gastos imprevistos.
- Esta metodología incentiva la colaboración de equipo, lo cual permite que mejore la productividad y fomente el compromiso para lograr desarrollar un producto de calidad.

- El punto más fuerte de esta metodología es la gestión de riesgos, debido a que se realiza después de cada iteración, el verificador beta se encarga de la detección de errores, con el fin de realizar los cambios respectivos y evitar futuras complicaciones o gastos imprevistos.
- Se enfoca en el desarrollo de productos de calidad a corto plazo.

CAPÍTULO 3.

3.1. Lugar de aplicación.

El videojuego educativo fue enfocado en estudiantes de quinto grado de Educación Básica de la Unidad Educativa “César Antonio Mosquera”, ubicado en la parroquia Julio Andrade perteneciente a la ciudad de Tulcán.

3.2. Materiales e insumos.

Materiales e insumos necesarios para el desarrollo del software.

Tabla 13.

Materiales e insumos.

Herramientas	Descripción	Costo
Blender	Diseño de modelos 3D	Libre
Microsoft Visual Studio	Herramienta para la programación del videojuego	Libre
Unity	Motor de videojuegos para el desarrollo del software.	Libre
Microsoft Paint	Editor de imágenes nativo de Windows	Libre
Git	Herramienta para el control de versiones y gestión de archivos del proyecto.	Libre
GitHub	Repositorio de código para almacenamiento en la nube.	Versión gratuita

3.3. Lenguaje de programación utilizado.

El lenguaje de programación que se utilizó para la elaboración del videojuego 3D es c#, este es un lenguaje de programación general y versátil debido a su similitud con java, también es uno de los principales lenguajes para la producción de videojuegos, C# fue específicamente creado para desarrollar aplicaciones que contengan un nivel de abstracción más alto(Layedra et al., 2022).

Sus ventajas son:

- Es multiplataforma y su idioma es de fácil comprensión.
- Similitud con el lenguaje de programación Java
- Inexistencia de punteros debido a que no son necesarios como en C.
- Rendimiento superior a java.

3.4. Unity 3D como motor de videojuegos.

¿Qué es un motor de videojuegos?

Antes del desarrollo de un videojuego es necesario tener en cuenta los comportamientos de cada uno de los elementos llamados objetos y así tener conocimiento del rol y función que desempeñarán dentro del videojuego, como también es indispensable un motor de videojuegos el cual se compone de una serie de rutinas para diseño y representación de la idea.

La principal función de un motor de videojuegos es facilitar un motor de renderizado para gráficos 2D y 3D, el entorno en donde se desenvuelve la historia, motor físico o también llamado detector de colisiones, animación, sistema de audio y gestión de almacenes para drenaje (Arenas Cancapa, 2019).

Unity 3D.

Unity, o mejor conocido como Unity 3D es un potente motor de videojuegos multiplataformas orientado al desarrollo de videojuegos 2D y 3D, es uno de los motores que lideran el mercado con más de 170.000 juegos, incluyendo juegos para Android y para PC.

Unity funciona con un motor de renderizado llamado Unity Engine. Es un motor que se basa en físicas para simular movimiento, colisión e interacción con objetos, esto permite la creación de entornos tridimensionales e iluminación en tiempo real. Además, Unity es un motor que ofrece soporte para gráficos 2D facilitando el desarrollo de videojuegos a escala para celulares.

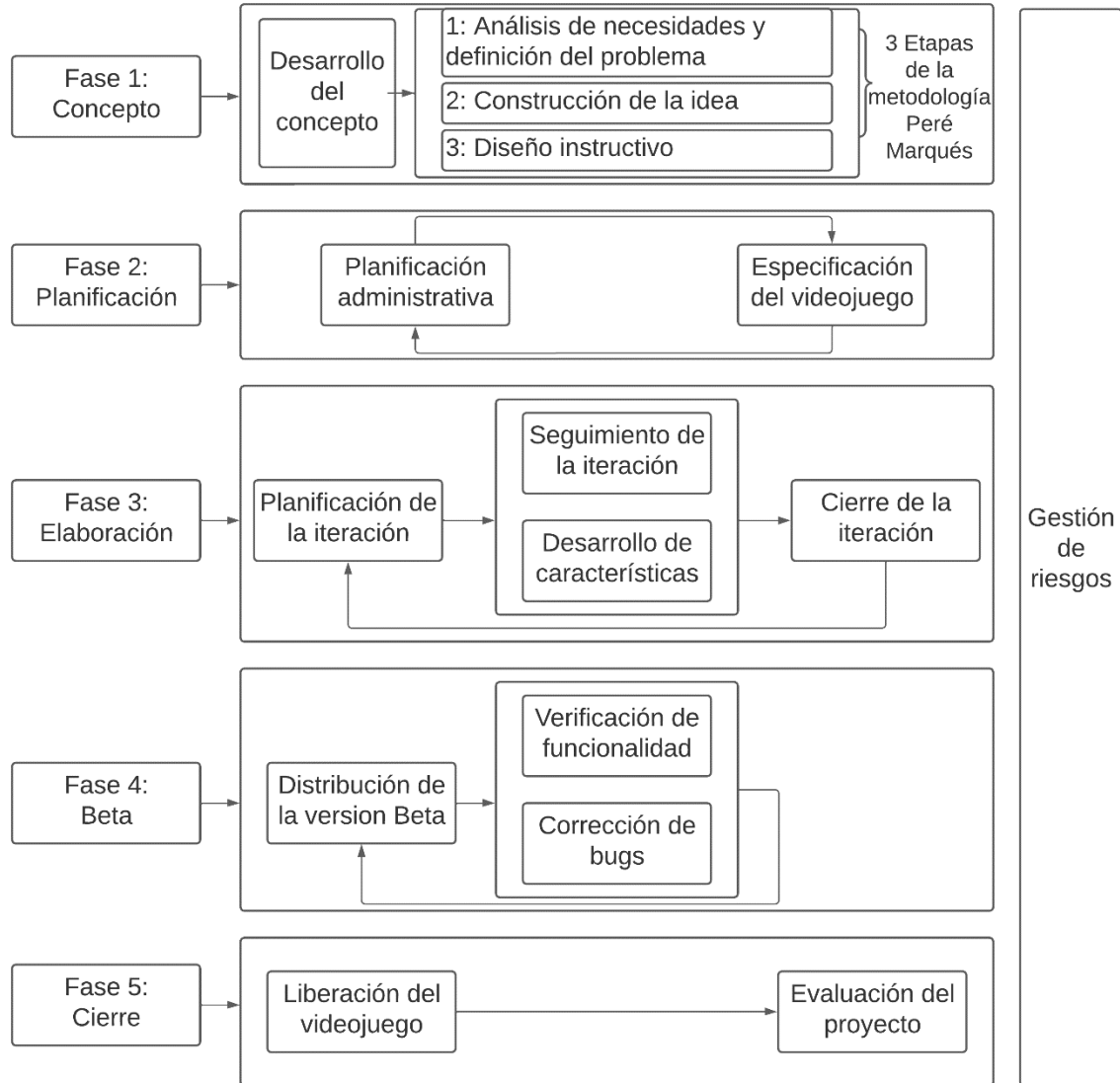
El renderizado de Unity está controlado por shaders programados en lenguaje Shaderlab, permitiendo que el programador pueda escribir sus propios shaders para lograr diversos y mejores efectos visuales(Romero González, 2018).

3.5. Metodología SUM modificada.

Para lograr un videojuego educativo se incorporó la metodología Peré Marqués en la primera fase de la metodología SUM, con el fin de construir una base sólida para que el videojuego incluya características educativas, se empleó las tres primeras etapas de la metodología Peré Marqués: Análisis de necesidades y definición del problema, Construcción de la idea y diseño instructivo. En la fase 1 o fase de concepto se definen los elementos educativos necesarios para el desarrollo del videojuego con sus respectivas mecánicas(Arenas Cancapa, 2019).

Figura 15.

Metodología de desarrollo para el videojuego.



3.5.1. Desarrollo del concepto (Fase 1).

Empleando las tres etapas de la metodología Perú Marqués se identificaron los objetivos del videojuego proyectados en destrezas o competencias que fueron la base para el diseño del videojuego 3D educativo.

1. Análisis de necesidades y definición del problema

El principal objetivo de este proyecto es reforzar el contenido académico de matemáticas a estudiantes de quinto grado de educación básica, debido a un bajo nivel de competencias en esta área. Para lograr esto es esencial que los estudiantes se sientan libres de presión, la cual se puede presentar al resolver un ejercicio matemático frente a una pizarra, entonces es necesario inculcar al estudiante una vista positiva sobre aprendizaje mediante herramientas audiovisuales o videojuegos educativos, esto da paso a que tengan un incentivo para educarse y así puedan aprender y relajarse compartiendo conocimiento.

2. Construcción de la idea.

Para una identificación más precisa de las necesidades requeridas por los estudiantes se desarrolló una tabla con una lista de preguntas realizadas a docentes de quinto nivel de educación básica. Sus respuestas permitieron construir un boceto inicial del videojuego.

Tabla 14.

Identificación de necesidades para el videojuego 3D educativo

Preguntas	Respuestas	Viabilidad para usar en videojuegos.	Ventajas o Desventajas
¿Cuáles son las competencias que más cuestan desarrollar?	La identificación de las variables a ser usadas en problemas matemáticos.	Sí se puede aplicar	El estudiante podrá aumentar su pensamiento creativo, agudizar su enfoque, memoria y mejorar su aprendizaje.
¿Qué temas son los menos llamativos para los estudiantes?	La resolución de problemas matemáticos con	Sí se puede aplicar	Los incentivos por aumento de puntos animan a los alumnos a mostrar más interés en los estudios.

	multiplicaciones o divisiones.		
¿Qué temáticas son las más difíciles de enseñar?	Las tablas de multiplicar y la resolución de problemas con multiplicaciones.	Sí se puede aplicar	Mejorar memorización de las tablas de multiplicar.
¿Cuál es el principal problema de aprendizaje?	La falta de práctica de tablas de multiplicar.	Sí se puede aplicar	La falta de práctica durante la educación virtual desencadenó un desinterés por la materia.
¿Cuál es la temática preferida por los estudiantes?	Juegos con sumas y restas.	Sí se puede aplicar	Se incluyen operaciones con sumas y restas para motivar al estudiante para que culmine todo el videojuego.

Fuente:(Arenas Cancapa, 2019).

3. *Diseño instructivo*

El PCA de la Unidad Educativa “César Antonio Mosquera”, de quinto grado de educación básica consta con las siguientes competencias:

- 1. Dominar el sistema de coordenadas cartesiana incluyendo las cuatro operaciones básicas que son: suma, resta, multiplicación y división las cuales son necesarias para resolver problemas de la vida diaria.
- 2. Trabajar en equipo para resolver problemas que surgen en la vida cotidiana, utilizando tecnología, ideas de proporcionalidad y algoritmos para operaciones matemáticas con números naturales.
- 3. Buscar patrones, secuencias numéricas y figuras poligonales en diferentes actividades infantiles, tomar en cuenta estructuras y artefactos referentes a la cultura, para disfrutar de las matemáticas e inspirar la búsqueda de soluciones a problemas diarios.

- 4. Utilizar la tecnología para resolver problemas cotidianos, por ejemplo, calcular perímetros y áreas de polígonos, tasar y medir longitudes, áreas, volúmenes y masas de cosas del entorno y transformar distintos sistemas de medida.
- 5. Utilizar las TIC para mejorar la conexión con la realidad ecuatoriana, analizar, evaluar y visualizar datos estadísticos, también generar medidas de tendencia central a partir de datos publicados en los medios de comunicación.

El videojuego está enfocado en el punto 2 “Trabajar en equipo para encontrar soluciones a problemas matemáticos”(MinEduc, 2022).

Visión del videojuego.

El videojuego 3D se creó, como herramienta de apoyo para que profesores de quinto grado profundicen en la materia de matemáticas y al mismo tiempo, se logre un entorno de aprendizaje interactivo y didáctico que potencie el interés de los alumnos por esta asignatura y esto facilite la resolución de problemas.

Género.

El videojuego educativo 3D incluye una mecánica de juego que resuelve un laberinto en primera persona recorriendo un entorno tridimensional, se clasifica en “Acción-Aventura”.

Mecánica del videojuego.

El videojuego incluye un reto, el cual requiere que el jugador recorra la casa laberíntica y para esto debe abrir varias puertas las cuales contienen preguntas de matemáticas, el ingreso en cada puerta depende de la decisión del jugador si es correcta se abre paso por la casa, caso contrario aparecen enemigos que debe derrotar con un arma la cual dispara pelotas de tenis. Además, incluye un sistema de puntuación y niveles de dificultad, el objetivo es abrir todas las puertas y conseguir la llave para finalizar el videojuego, para terminar, se implementó una zona de retroalimentación

que permite observar las preguntas que fueron contestadas correcta e incorrectamente y las puntuaciones de anteriores jugadores clasificados en top 10.

Plataforma.

El videojuego está diseñado para computadores de escritorio y portátiles con sistema operativo Windows.

Modelado del diseño 3D.

El equipo de desarrollo realizó bocetos de casa y objetos internos en el modelador 3D Blender v2.93.

Figura 16.

Boceto de objetos internos de la casa

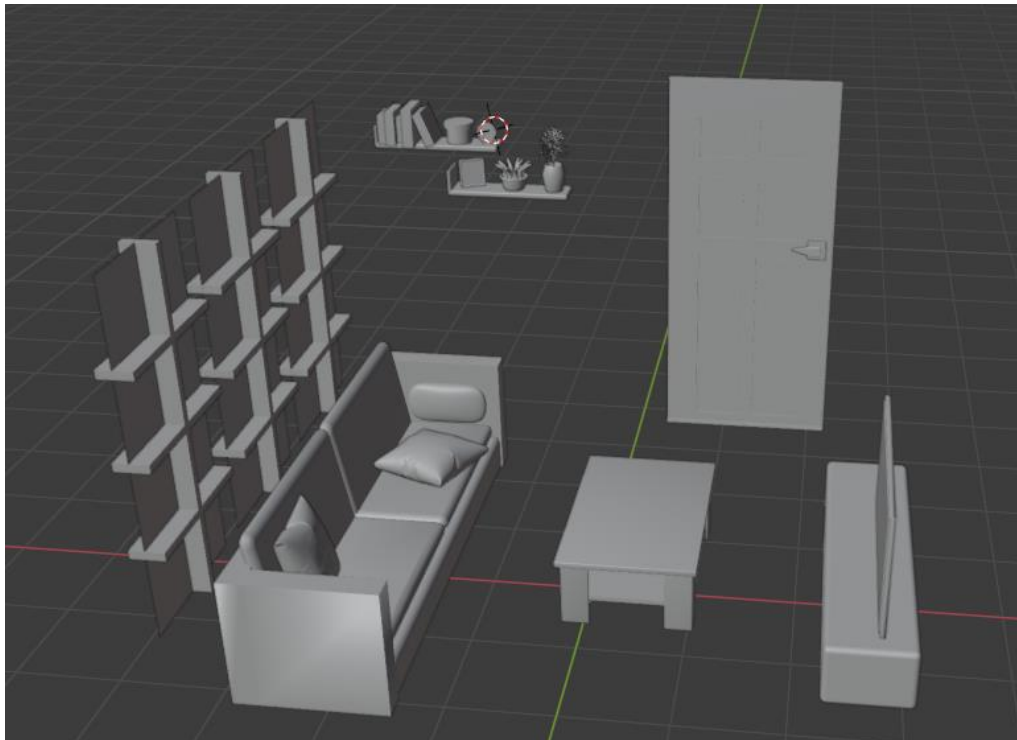


Figura 17.

Modelo de enemigos.

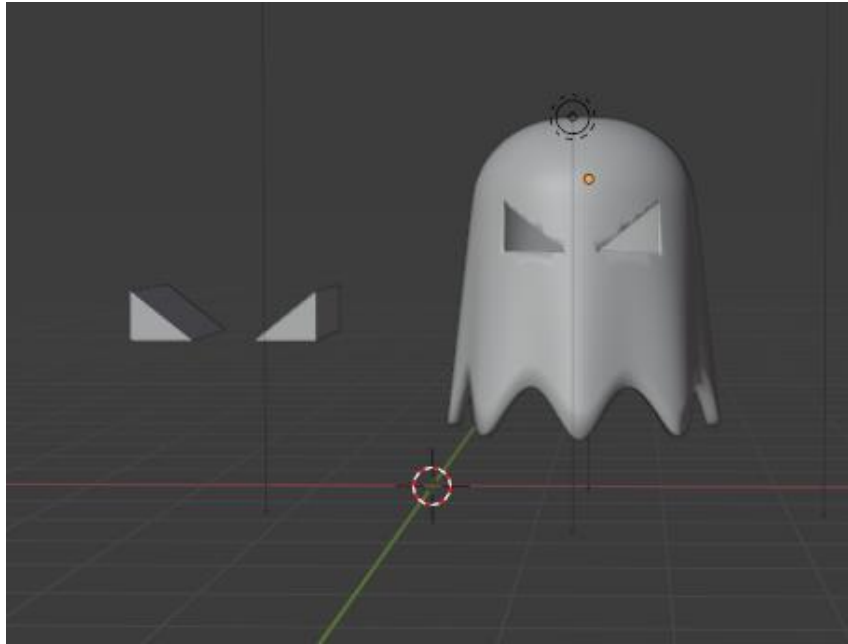


Figura 18.

Creación de objetos para curación

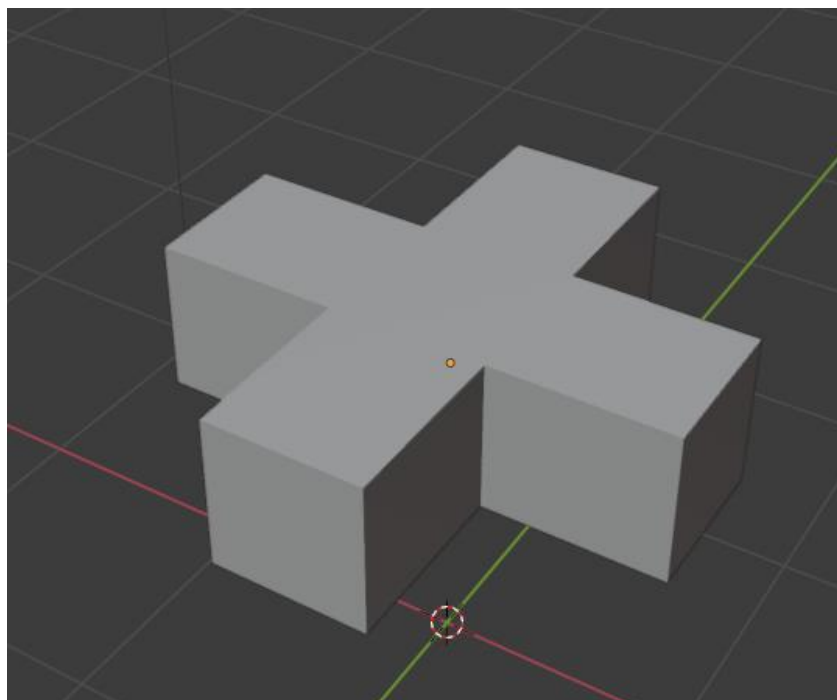


Figura 19.

Modelo de objetos que contienen información.



Figura 20.

Modelo de objeto para cambio de color de texto

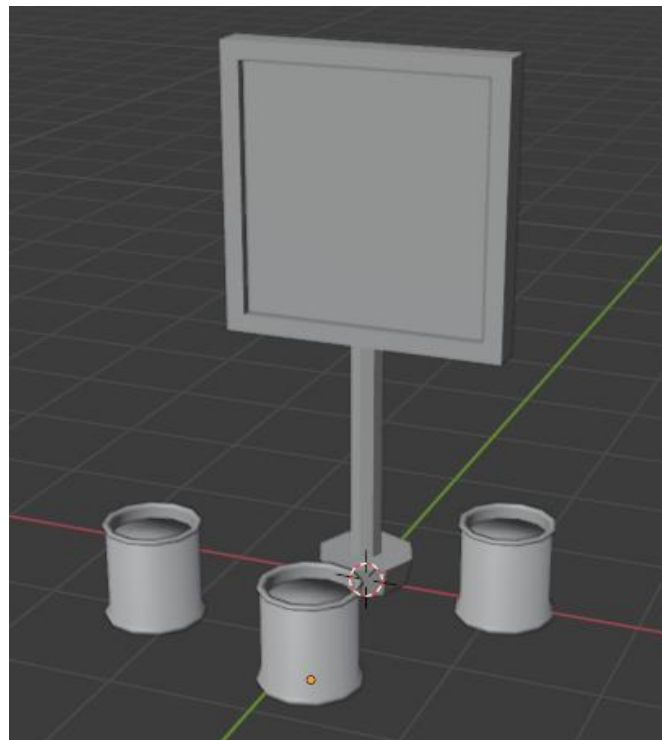


Figura 21.

Boceto inicial de las paredes de la casa.

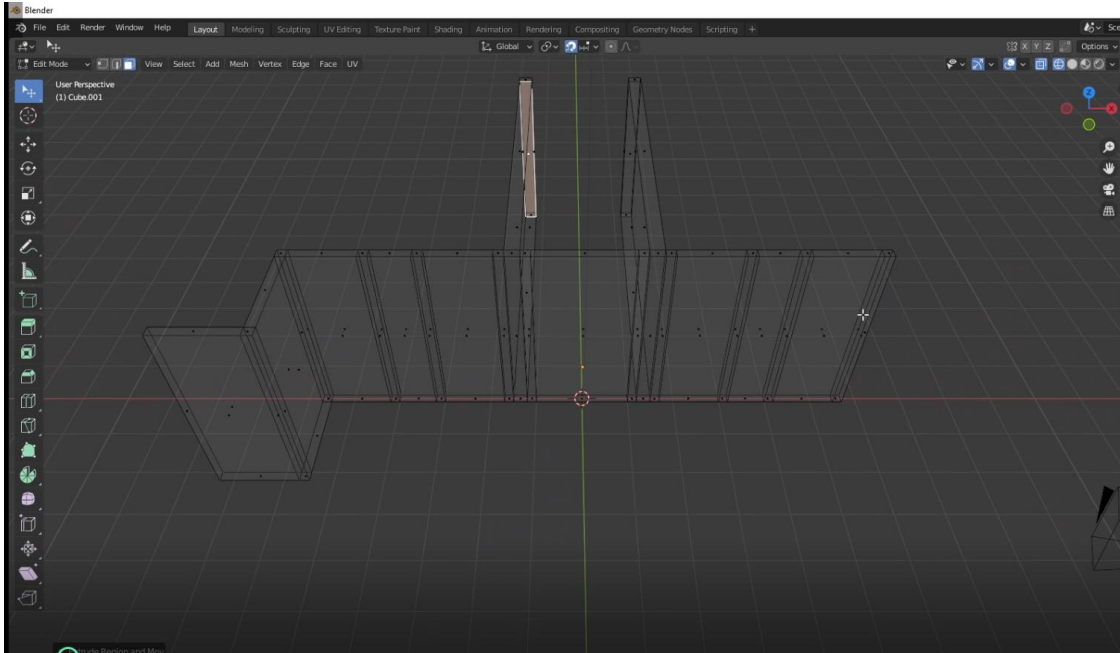


Figura 22.

Modelo 2D de las paredes internas de la casa.

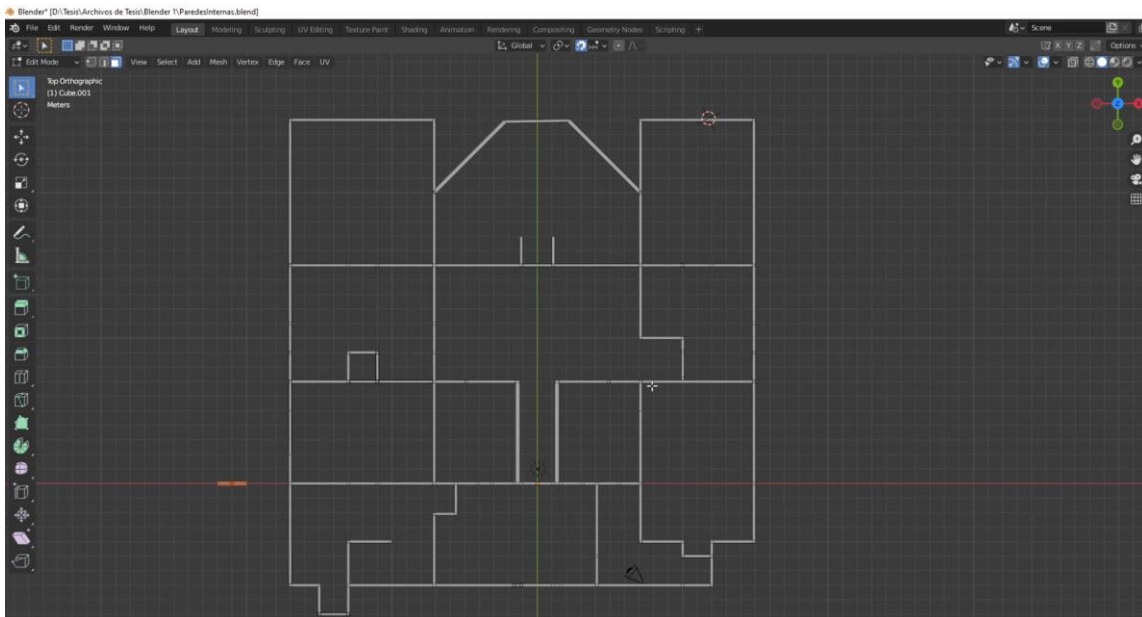
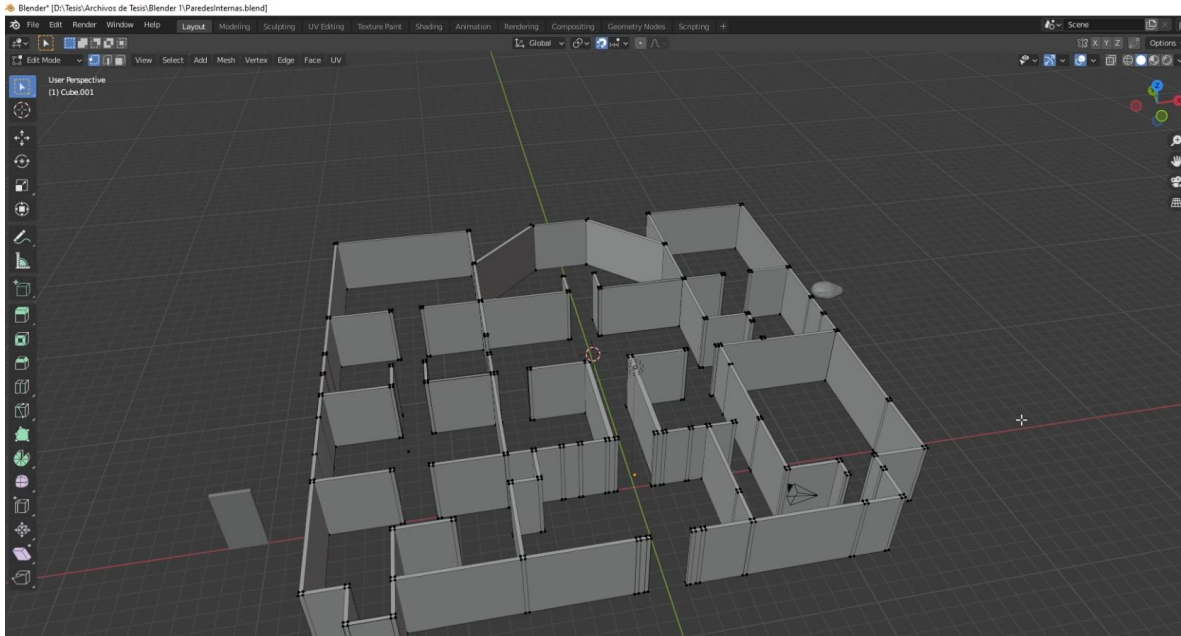


Figura 23.

Modelo 3D de paredes internas de la casa.



3.5.2. Planificación de la aplicación. (Fase 2).

Para lograr con éxito la finalización del proyecto se definieron los objetivos que debe cumplir el proyecto de titulación

3.5.2.1. Planificación Administrativa.

1) Objetivos del proyecto

a) Objetivo general.

Elaborar un videojuego 3D educativo como herramienta didáctica para mejorar las destrezas en la resolución de problemas matemáticos dirigido a niños de quinto grado de la unidad educativa.

b) Objetivos específicos

- El videojuego debe ser intuitivo.

- Reforzar el conocimiento sobre tablas de multiplicar.
- Mejorar la capacidad de los estudiantes para resolver problemas matemáticos.

2) Equipo de trabajo.

El equipo de trabajo encargado de la elaboración del proyecto estuvo compuesto por un equipo de desarrollo, stakeholders, product owner, scrum master y un verificador beta.

Tabla 15.

Equipo de trabajo.

Nro.	Rol	Nombre
1	Scrum Master	MSc. Carpio Pineda (Tutor)
2	Product owner	MSc. Guillermo Reascos. (Rector de la unidad Educativa)
3	Equipo de desarrollo	Anderson Aux
4	Stakeholder 1	MSc. Fausto Salazar
5	Stakeholder 2	MSc. Silvia Arciniega
6	Verificador beta	MSc. Carpio Pineda (Tutor)

Tabla 16.

Equipo de desarrollo.

Nro.	Rol	Nombre	Alias
1	Programador	Anderson Aux	AA
2	Diseñador Grafico	Anderson Aux	AA
3	Técnico de Sonido	Anderson Aux	AA

3.5.2.2. Especificación del videojuego.

El cronograma de actividades se detalla en la siguiente **Figura 24**, donde se puede visualizar las semanas asignadas para el desarrollo del videojuego 3D.

Figura 24.

Cronograma de actividades.

			SEMANAS 2022-2023													
Fase	Actividad	Días	42	43	44	45	46	47	48	49	50	1	2	3	4	5
Fase 1	Desarrollo del concepto	3	■													
Fase 2	Planificación	2		■	■											
Fase 3	Elaboración				■	■	■	■	■	■	■	■	■	■	■	■
	Iteración 1	5			■	■										
	Iteración 2	5				■	■									
	Iteración 3	5					■	■								
	Iteración 4	5						■	■							
	Iteración 5	5							■	■						
	Iteración 6	5								■	■					
	Iteración 7	5									■	■				
	Iteración 8	5										■	■			
	Iteración 9	5											■	■		
	Iteración 10	5												■	■	
Fase 4	Beta													■	■	
	Distribución de la version beta 1	2												■	■	
	Distribución de la version beta 2	2													■	■
	Distribución de la version beta 3	2														■
Fase 5	Cierre	2														■
	Gestión de riesgos	63	■	■	■	■	■	■	■	■	■	■	■	■	■	■

Para lograr culminar el videojuego en el tiempo previsto, se le asignan actividades a cada miembro del equipo de desarrollo, las cuales se detallan en la **Tabla 17**, con el fin de agilizar el proceso de desarrollo y prevenir futuro inconvenientes.

Tabla 17.*Lista de actividades asignada al equipo de desarrollo.*

Nro.	Descripción	Horas
Iteración 1		
1	Realizar el borrador del entorno 3D.	30
2	Organizar los elementos de la escena.	10
3	Elegir imágenes de prueba para la UI.	5
4	Elegir sonidos de prueba para la interacción de elementos.	10
5	Agregar paneles para información de retos y misiones.	40
Iteración 2(Blender)		
6	Modelar el diseño 3D, casa y objetos internos.	20
7	Modelar el arma y la pelota de tenis	10
8	Modelar el plano del laberinto interno que incluye la casa.	10
9	Formar un prediseño de la escena en Unity con los elementos modelados.	30
Iteración 3(Unity)		
10	Creación de escenas para el videojuego.	20
11	Agregar botones para navegación entre escenas.	10
12	Incluir controlador de volumen.	10
Iteración 4(Unity)		
13	Añadir imágenes necesarias en la UI para dar vida al personaje, puntuación, munición y stamina.	5
14	Crear objetos vacíos para añadir los scripts que manejen los elementos antes mencionados.	15
15	Agregar código inicial en el método start del script para el inicio del videojuego.	20
Iteración 5. (Unity)		
15	Crear el objeto "Player" y añadirle cámara y arma.	2
16	Añadir a "Player" scripts para interacción, movimiento y manejo de objetos.	20
17	Incluir nuevos objetos para el rol de enemigos, cajas de munición y objetos para curar la vida del jugador.	10

18	Añadir Scripts a los objetos antes mencionados.	20
Iteración 6(Unity UI)		
19	Añadir código a los objetos de la UI para activación y desactivación de imágenes.	15
20	Enlazar acciones entre UI y Player por medio de código.	20
21	Control de paneles dependiendo de decisiones del jugador.	20
Iteración 7(Unity)		
22	Cuando el jugador interactúa con las puertas, incluir código que permita controlar la activación y desactivación de estas.	20
23	Añadir código para la interacción entre enemigo y jugador.	30
24	Añadir código de navegación que restrinja el movimiento del enemigo a las zonas designadas.	30
Iteración 8(Unity)		
25	Diseño de la interfaz del Quiz.	10
26	Agregar código de control para preguntas y respuestas.	40
27	Validación de preguntas correctas e incorrectas y agregar acciones a la puerta según la respuesta.	60
Iteración 9(Unity)		
28	Añadir partículas cuando el jugador pierda vida.	10
29	Añadir explosiones cuando los enemigos sean derrotados y cada vez que colisionen con el jugador.	5
30	Mejora del entorno 3D, modelos y texturas.	15

3.5.3. Elaboración. (Fase 3).

El presente proyecto de titulación se completó en un lapso de 14 semanas, es necesario detallar cada actividad y subactividad, realizada en el transcurso del periodo académico, estas modificaciones son llamadas iteraciones.

3.5.3.1.Elaboración de iteración 1.

Para la primera iteración se espera obtener una base en la cual se logre desarrollar el escenario y la mecánica del videojuego 3D.

a) Planificación de la iteración.

Asignar actividades y subactividades a cada miembro del equipo de desarrollo.

Tabla 18.

Lista de tareas iteración 1.

ítem	Descripción	Encargado
1.	Realizar el borrador del entorno 3D.	AA
1.1.	Dibujar un boceto de la casa y de paredes internas.	AA
1.2.	Boceto de los enemigos.	AA
2.	Elegir imágenes de prueba para la UI.	AA
2.1.	Buscar imágenes para los objetos en canvas.	AA
2.2.	Elegir imágenes adecuadas para la visualización infantil.	AA
3.	Elegir sonidos de prueba para la interacción de elementos en Unity Asset Store.	AA
3.1.	Seleccionar una canción de fondo para el videojuego.	AA
3.2.	Descargar sonidos para el arma, explosiones, movimientos e interacciones del player y enemigos.	AA
4.	Agregar paneles para información de retos y misiones.	AA
4.1.	Incluir paneles en el canvas con su respectivo cuadro de texto.	AA

Para continuar con el proceso se detalla la actividad realizada por cada miembro del equipo de desarrollo.

- Diseñador Gráfico

Es importante seguir las etapas descritas en la **Tabla 19** para el diseño de casa, enemigos, laberinto y los diversos elementos internos de la casa:

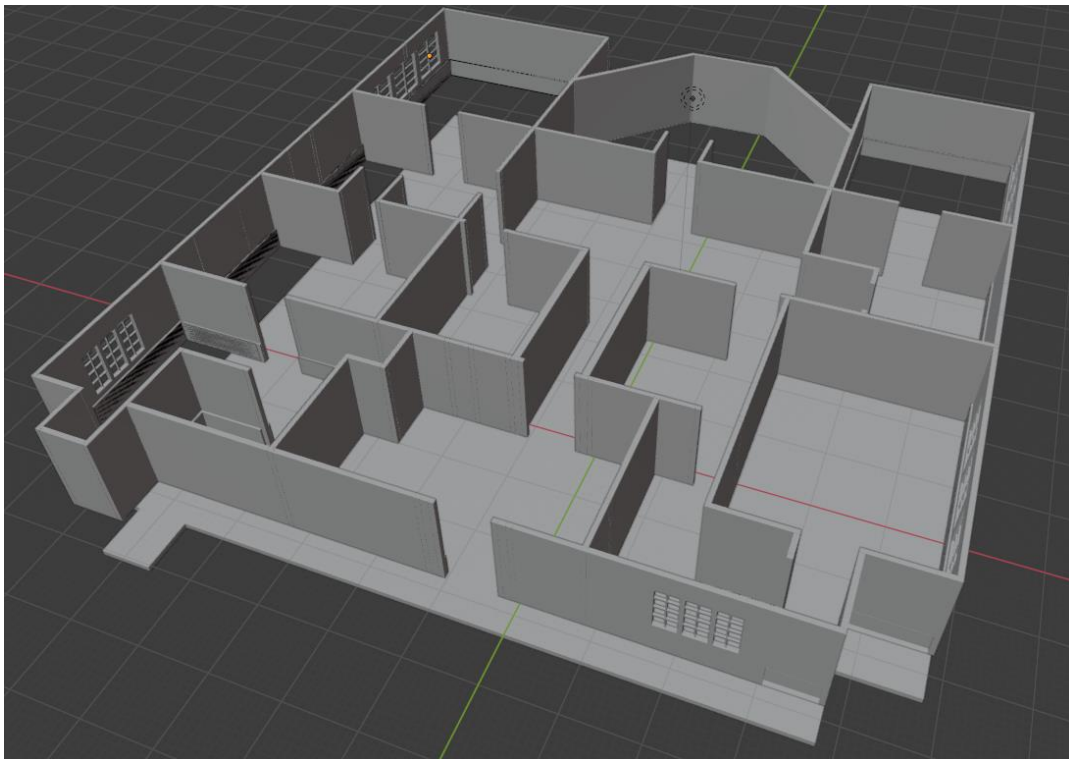
Tabla 19.

Digitalizar elementos necesarios para el videojuego.

Paso	Procedimiento
1	Según los bocetos realizados en el desarrollo del concepto realizar un modelo inicial para la base del videojuego.
2	Escanear los modelos realizados.
3	Las imágenes deben dimensionarse de acuerdo con los requerimientos del player.
4	Recortar las ilustraciones según sea necesario.
5	Posicionar las imágenes en el canvas.
6	Agregar los respectivos cuadros de texto.

Figura 25.

Imagen digitalizada del modelo.



- Técnico de sonido

El procedimiento para la obtención de sonidos de cada elemento para el videojuego se detalla en la **Tabla 20**.

Tabla 20.

Selección de sonidos que incluye el videojuego.

Paso	Procedimiento
1	Ingresar en Unity Asset Store.
2	Realizar una búsqueda de packs gratuitos de sonido.
3	Descargar el pack seleccionado.
4	Importar el pack de sonidos anteriormente descargado.
5	Seleccionar los sonidos adecuados para implementar en el videojuego.

Figura 26.

Ingreso a Unity Asset Store.

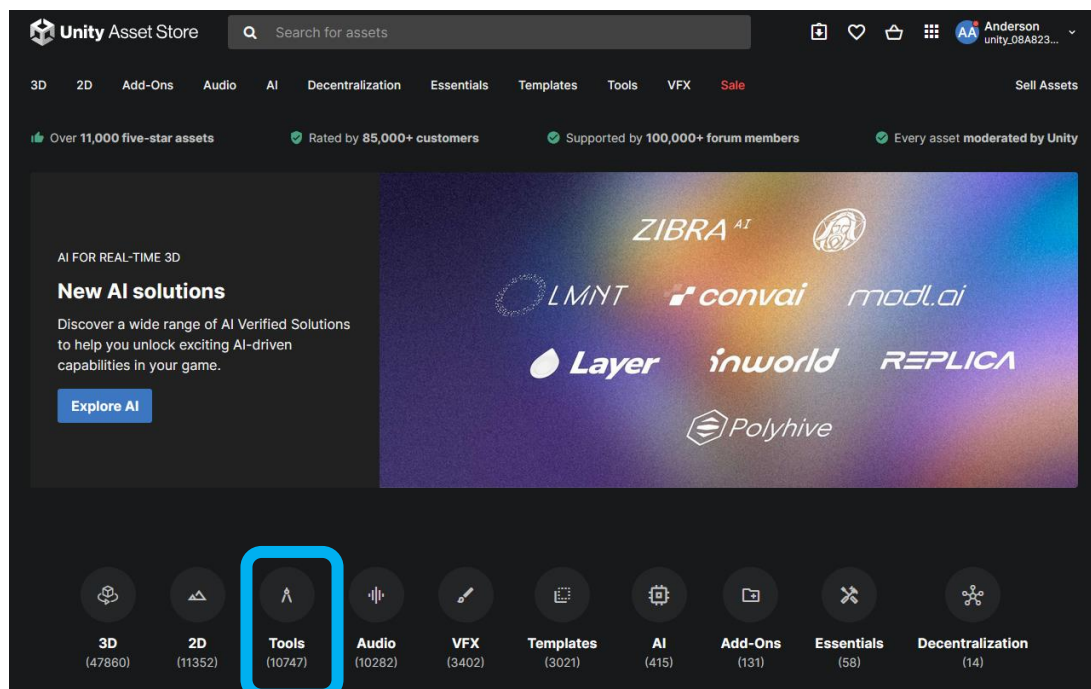


Figura 27.

Descarga de packs de audios gratuitos.

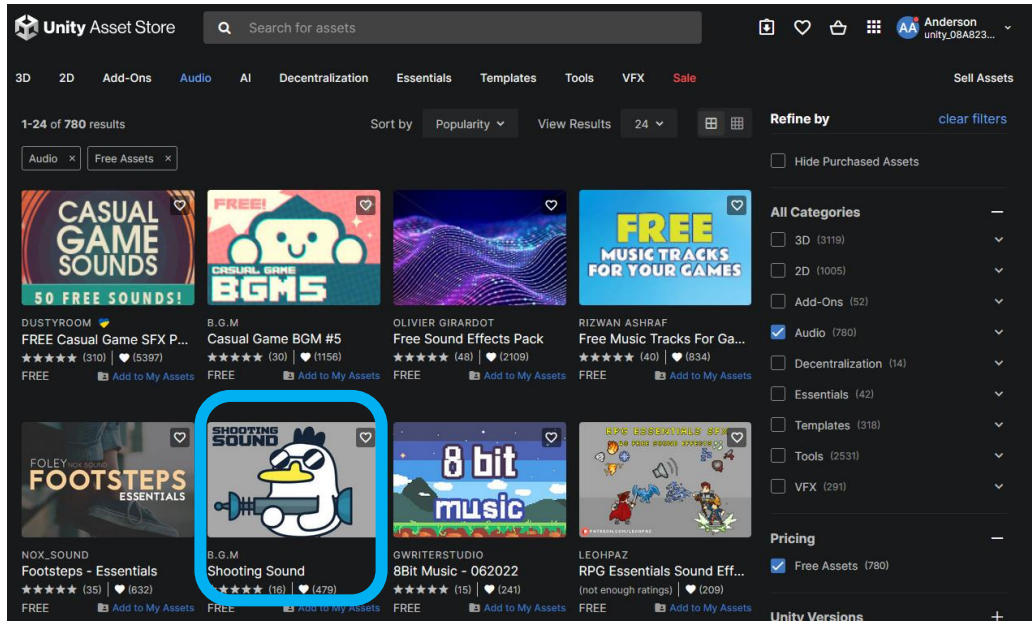
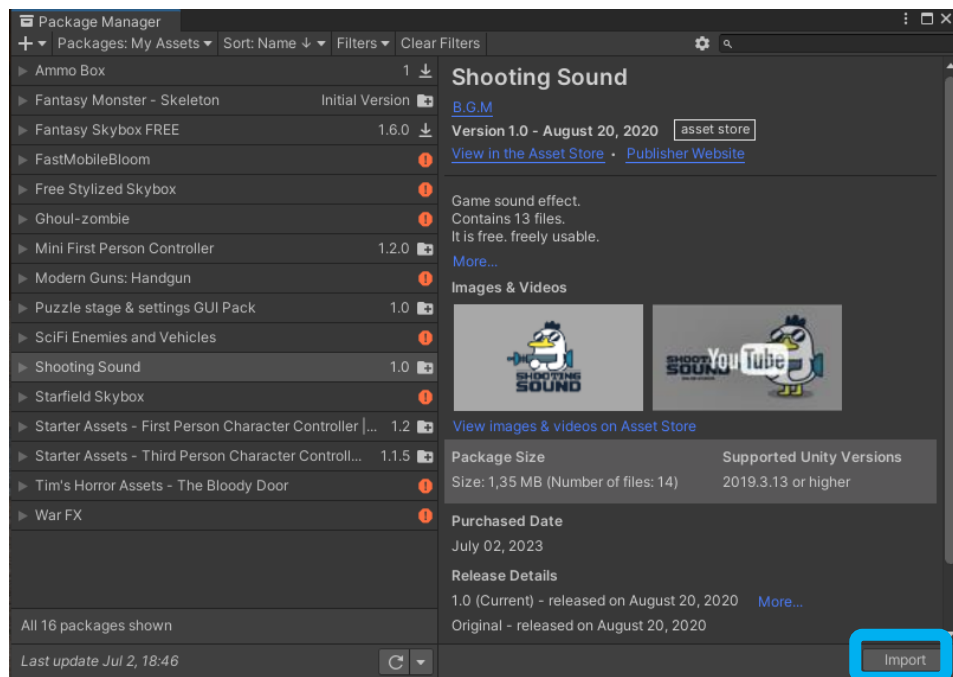


Figura 28.

Importación de pack en el proyecto.



3.5.3.2.Elaboración de iteración 2.

Como principal objetivo de esta iteración es, desarrollar modelos 3D en la plataforma Blender para implementarlos en el escenario 3D de la escena principal de Unity.

a) Planificación de la iteración.

Tabla 21.

Lista de tareas iteración 2.

Ítem	Descripción	Encargado
5.	Modelar el diseño 3D, casa y objetos internos.	AA
5.1.	Modelar el diseño externo de la casa.	AA
5.2.	Modelar cada objeto para ser incluido en el interior de la casa.	AA
6.	Modelar el arma y la pelota de tenis.	AA
6.1.	Crear polígonos y organizarlos para formar una pistola de tenis.	AA
6.2.	Modelar, moldear y texturizar una esfera para formar una pelota de tenis.	AA
7.	Modelar el plano del laberinto interno que incluye la casa.	AA
7.1.	Crear el modelo estructural	AA
7.2.	Agregar ventanas y marcos de puertas	AA
8.	Formar un prediseño de la escena en Unity con los elementos modelados.	AA
8.1.	Ordenar los elementos para ser usados en la escena 3D	AA

Desarrollo de características.

- Diseñador Gráfico.

Para realizar los modelos 3D se hizo uso de la plataforma Blender en su versión 2.93.11. A continuación se detalla el proceso realizado para moldear los elementos a partir de cubos 3D.

Tabla 22.

Modelado de objetos 3D.

Paso	Procedimiento
1	Definir objetos a ser modelados en Blender.
2	Crear polígonos de objetos internos de la casa.
3	Organizar polígonos según su función.
4	Eliminar vértices duplicados.
5	Verificar la orientación de caras en todos los modelos.
6	Organizar elementos de escena según su prioridad.

A continuación, se detalla el procedimiento realizado por el diseñador gráfico.

Figura 29.

Modelo estructural del interior de la casa.

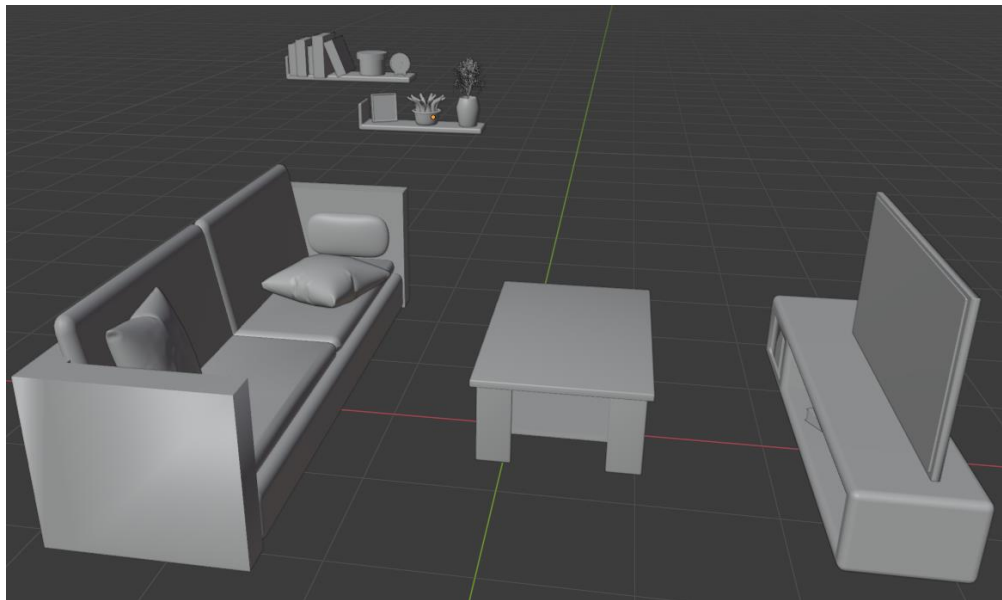
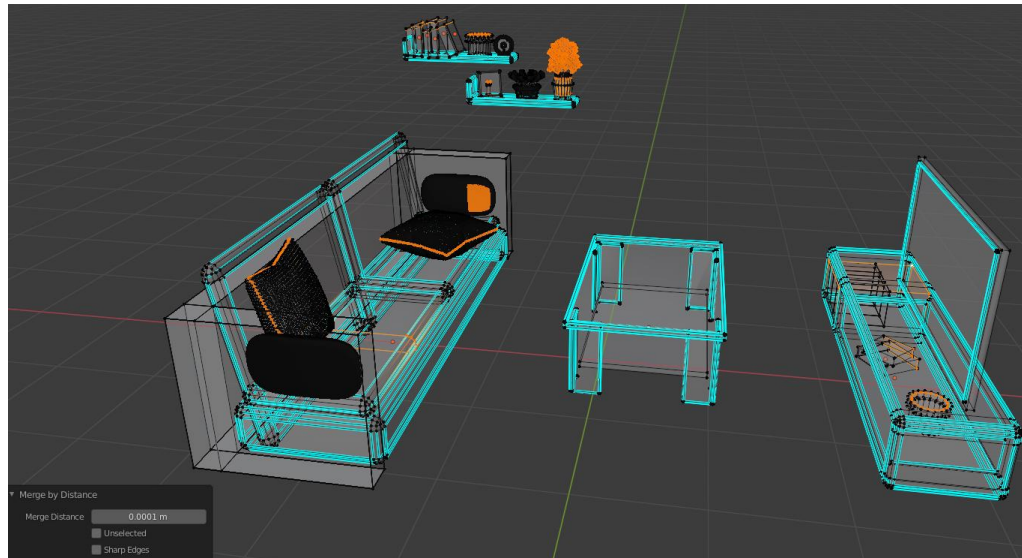


Figura 30.

Eliminación de vértices duplicados.



3.5.3.3.Elaboración de iteración 3.

El objetivo de esta iteración es implementar escenas necesarias para el entorno 3D del videojuego, como también son necesarios los botones de navegación entre escenas y un controlador de volumen.

a) Planificación de la iteración.

Tabla 23.

Lista de tareas iteración 3

Ítem	Descripción	Encargado
9.	Creación de escenas para el videojuego.	AA
9.1.	Crear una escena “MainMenu”.	AA
9.2.	Crear una escena “MainScene”.	AA
9.3.	Crear una escena “Options”	AA
9.4.	Crear una escena “Record”	AA

10.	Agregar botones para navegación entre escenas.	AA
10.1.	Ordenar botones de navegación de manera intuitiva en cada escena.	AA
10.2.	Agregar código para una correcta navegación entre escenas.	AA
11.	Incluir controlador de volumen.	AA
11.1.	Se implementa una clase “AudioManager” para controlar el volumen del juego.	AA

b) Desarrollo de características.

Para proseguir con la iteración a continuación se detalla paso a paso el trabajo realizado por cada miembro del equipo de desarrollo.

- Diseñador Gráfico.

El diseñador gráfico encargado del entorno visual realizó la implementación de botones en la escena siguiendo el siguiente proceso detallado en la **Tabla 24**:

Tabla 24.

Creación de escenas y botones.

Paso	Procedimiento
1	Dirigirse a la barra de herramientas ubicada en la parte superior.
2	Dar clic en la sección “File”.
3	Proseguir con la opción “New Scene”.
4	Escoger una plantilla vacía para crear todos los objetos necesarios.
5	Guardar la escena y establecer un nombre para posteriormente lograr un correcto funcionamiento.
6	Crear botones en cada escena según sean necesarios.
7	Ubicar cada botón de manera estratégica.

Figura 31.

Creación de escenas.

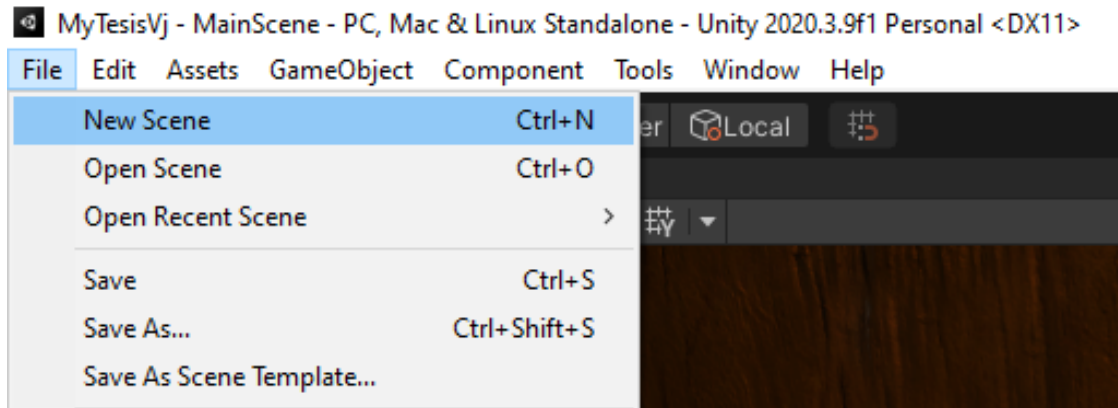


Figura 32.

Ordenar escenas según su prioridad.

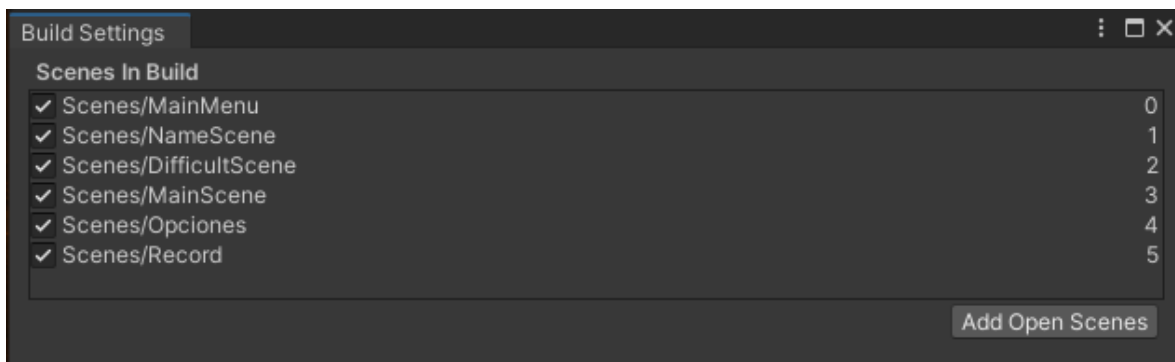


Figura 33.

Botones de navegación.



- Programador.

Se implementaron los siguientes métodos en la clase “MainMenu” que permita una correcta navegación entre escenas y que esto funcione de manera eficiente para lograr una experiencia de juego intuitiva.

Tabla 25.

Creación y programación de scripts del videojuego.

Paso	Procedimiento
1	Definir la función de cada botón en su respectiva escena.
2	Definir una opción de salida para terminar el videojuego.
3	Crear una clase para cada escena que permita controlar su navegación.
4	Crear métodos para un funcionamiento conjunto con los botones.
5	Incluir un slider para el manejo de volumen general.
6	Crear métodos necesarios para el control de sonido

Figura 34.

Métodos para manejo de escenas

```
0 referencias
public void EscenaJuego()
{
    SceneManager.LoadScene("NameScene");
}
0 referencias
public void RegresarMenuPrincipal()
{
    SceneManager.LoadScene("MainMenu");
}
0 referencias
public void CargarNivel(string nNivel)
{
    SceneManager.LoadScene(nNivel);
}
0 referencias
public void Salir()
{
    Debug.Log("Salir...");
    Application.Quit();
}
```

Figura 35.

Métodos para manejo del Audio Manager.

```
public class AudioManager : MonoBehaviour
{
    public Slider controlVolume; //Referencia a la barra de volumen
    public GameObject[] vectorAudios;
    public Image imagenMute;

    // Start is called before the first frame update
    @ Mensaje de Unity | 0 referencias
    void Start()
    {
        vectorAudios = GameObject.FindGameObjectsWithTag("audios"); //Obten
        controlVolume.value = PlayerPrefs.GetFloat("volumenSave", 0.5f); //C
    }

    @ Mensaje de Unity | 0 referencias
    private void Update()
    {
        foreach(GameObject au in vectorAudios)
        {
            au.GetComponent<AudioSource>().volume = controlVolume.value; //
            if(controlVolume.value <= 0)
            {
                imagenMute.enabled = true;
            }
            else
            {
                imagenMute.enabled = false;
            }
        }
    }
}
```


- Técnico de sonido.

En la **Tabla 26** se puede visualizar el procedimiento ejecutado por el técnico de sonido para obtener un correcto control de volumen en el videojuego.

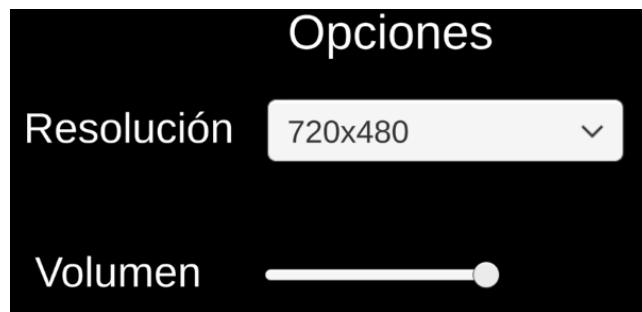
Tabla 26.

Adaptación del “AudioManager”.

Paso	Procedimiento
1	Definir las escenas que incluyen el slider para control de volumen.
2	Establecer un mínimo y un máximo que limite el sonido.
3	Adjuntar el script creado por el programador al slider de la escena.
4	Verificar el correcto funcionamiento del controlador de sonido

Figura 36.

Mínimo y máximo del control de volumen.



3.5.3.4. Elaboración de iteración 4.

El objetivo principal de esta iteración es incluir imágenes para interfaz de usuario como también crear objetos vacíos para el control de funciones generales del videojuego y agregar código para lograr un arranque inicial del videojuego.

a) Planificación de la iteración.

Tabla 27.

Lista de tareas iteración 4

Ítem	Descripción	Encargado
12.	Añadir imágenes necesarias en la UI para vida del personaje, puntuación y munición.	AA
12.1.	Agregar al objeto canvas una imagen que refleje la vida del player.	AA
12.2.	Añadir al objeto canvas una imagen que refleje la munición del arma.	AA
12.3.	Añadir al objeto canvas una imagen que indique la puntuación del player.	AA
13.	Crear objetos vacíos para añadir los scripts que manejen los elementos antes mencionados.	AA
13.1.	Crear objeto para la clase “GameManager”.	AA
13.2.	Crear un objeto para controlar la puntuación.	AA
13.3.	Crear un objeto para incluir el script de control de munición.	AA
14.	Agregar código inicial en el método start del script para el inicio del videojuego.	AA
14.1.	Implementar el código inicial para la posición del player, valor de vida, puntuación y munición de arma.	AA

b) Desarrollo de características.

- Diseñador Gráfico.

Tabla 28.

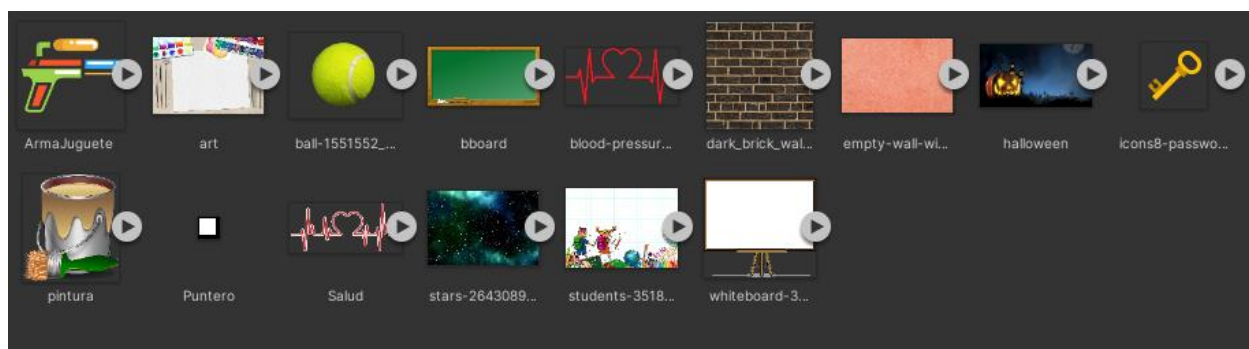
Visualizar imágenes en la UI.

Paso	Procedimiento
1	Descargar la imagen en formato PNG para la vida, puntuación y munición del player.
2	Ubicar la imagen en una carpeta dentro del proyecto.
3	En propiedades de la imagen dar clic en Texture types.
4	Seleccionar la opción “Sprite 2D and UI”
5	Agregar un objeto en el canvas que contenga el componente image.
6	Seleccionar “source image”.

7	Seleccionar la imagen adecuada para ser visualizada en la UI del objeto correspondiente.
---	--

Figura 37.

Imágenes utilizadas en el canvas.



- Programador.

Se crea la clase Game Manager, esta clase es esencial debido a que debe contener el valor de vida, puntos, munición y asignaciones principales del juego.

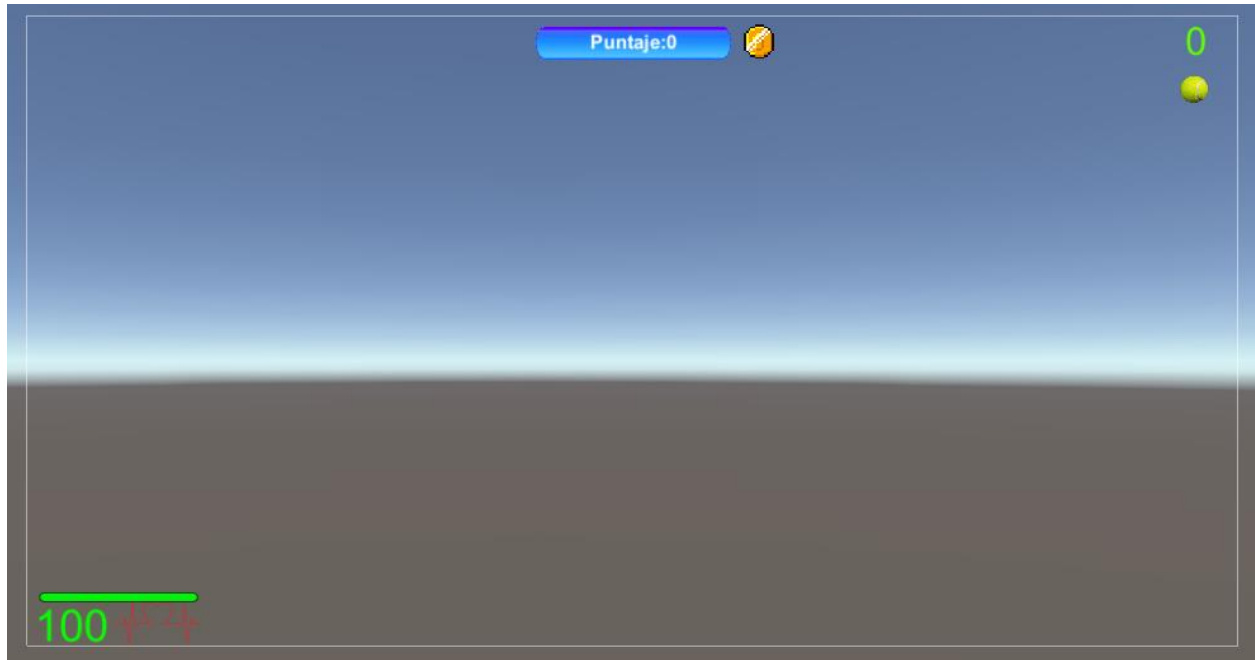
Tabla 29.

Visualización inicial de UI y ubicación principal del player.

Paso	Procedimiento
1	Crear clase “GameManager” para control general de la interacción del player.
2	Crear clase “Records” para control de puntaje.
3	Crear una clase “Gun” para control de munición y manejo del arma.
4	Enlazar los scripts antes mencionados a los objetos vacíos creados en la escena.
5	Agregar código inicial para la posición del player y componentes iniciales en la UI.

Figura 38.

Visualización inicial de imágenes.



Nota: representen el nivel de vida, puntuación y munición en la UI principal del canvas.

3.5.3.5.Elaboración de iteración 5.

El objetivo principal de esta iteración es brindar al jugador las funciones básicas de movilidad del player, así como la rotación de la cámara, capacidad de disparar a los adversarios y una función de arma especializada. Todas estas características se implementaron a través del código.

a) Planificación de la iteración.

Tabla 30.

Lista de tareas iteración 5

Ítem	Descripción	Encargado
15.	Crear el objeto “Player” y añadirle cámara y arma.	AA
15.1.	Crear una cápsula que contenga la cámara principal de visualización.	AA
15.2.	Añadir el modelo 3D del arma con un “Spawn point” y municiones.	AA
16.	Añadir a “Player” scripts para interacción, movimiento y manejo de objetos.	AA
16.1.	Crear clase “Player movement” para control de movimientos.	AA
16.2.	Crear clase “Player interactions” la cual verifique la interacción del player con los modelos 3D de la escena.	AA
17.	Incluir nuevos objetos que tendrán rol de enemigos, cajas de munición y objetos para curar la vida del jugador.	AA
17.1.	Importar los modelos realizados en la iteración 2.	AA
17.2.	Posicionar estratégicamente cada objeto en la escena.	
18.	Añadir Scripts a los objetos antes mencionados.	AA
18.1.	Crear clase “Attack Enemy”, “Follow Player Movement”, “Enemy UI”, “BillBoard”, “Enemy Shoot”.	AA
18.2.	Crear clase “Bullet”, “Weapon Sway”, “Ammo Box”.	AA
18.3.	Crear clase “Health Objects” la cual controle la recuperación de vida para el player.	AA

b) Desarrollo de características.

- Diseñador Gráfico.

El diseñador gráfico está a cargo de las siguientes actividades:

Tabla 31.

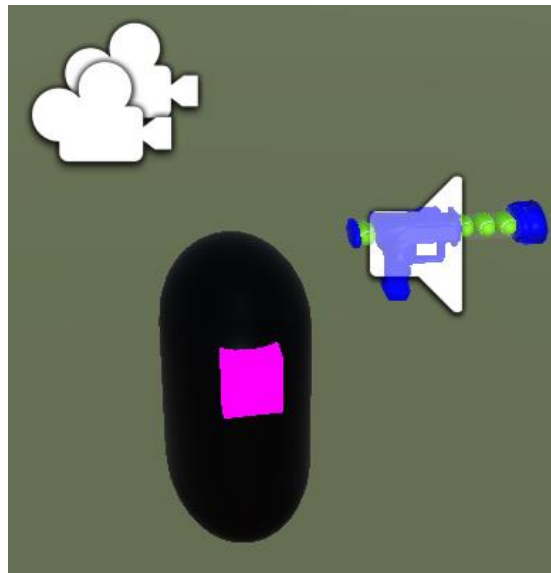
Incorporar modelos principales en la escena main.

Paso	Procedimiento
1	En la escena principal agregar un objeto cápsula que hará referencia al jugador.
2	Añadir al objeto creado una cámara y un box collider.

-
- 3 Incorporar a la cápsula, el modelo de pistola de juguete la cual contiene pelotas de tenis.
 - 4 Ubicar un punto “spawn point” en la pistola de juguete, es el lugar de donde se disparan las pelotas de tenis.
 - 5 Agregar los modelos de cajas de munición y enemigos creados en la iteración 2.
 - 6 Añadir el modelo para recuperar vida, el cual tiene forma de cruz.
 - 7 Posicionar de manera estratégica los enemigos.
-

Figura 39.

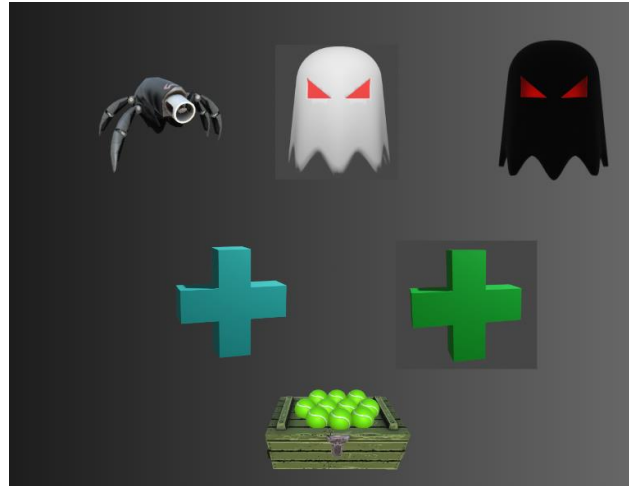
Capsula contenedora de cámara y modelo de la pistola.



Nota: Hace referencia al player.

Figura 40.

Modelo de munición, enemigos y generador de vida.



- Programador.

Tabla 32.

Control de player, enemigos, objetos proveedores de munición y vida.

Paso	Procedimiento
1	Añadir la clase “Player Movement” y “Player Interaction” a la capsula con referencia del jugador.
2	Enlazar las clases: “Atack Enemy”, “Follow Player Movement”, “Enemy UI”, “BillBoard”, “Enemy Shoot” al modelo del enemigo.
3	Enlazar las clases: “Bullet”, “Weapon Sway”, “Ammo Box” a los objetos: arma, pelota de tenis y caja de munición.
4	Enlazar la clase: “Health Objects” la cual sirvió como regeneración de vida para el jugador.
5	Añadir código para el movimiento del jugador, movimiento de cámara.
6	Implementar métodos que faciliten la mecánica desempeñada por el enemigo.
7	Generar código para la recuperación de vida.
8	Controlar la colisión de enemigos, munición y generador de vida.

Figura 41.

Clase implementada en el jugador.

```
public class PlayerInteractions : MonoBehaviour
{
    public Transform startPosition; // Ir a la ubicación inicial
    [SerializeField] GameObject explosionPrefab; // Prefab para instanciar una explosión
    float lifeTimeExplosion = 3.0f; // Tiempo de explosión
    public DoorTrigger doorToOpen; // Referencia a la clase DoorTrigger
    public GameObject imageKey; // Referencia a la imagen de la llave
    public TextMeshProUGUI textMission; // Referencia al manejo del textMission

    // Sonidos
    [SerializeField] private AudioClip healthSound;
    [SerializeField] private AudioClip AmmoSound;
    [SerializeField] private AudioClip KeySound;
    [SerializeField] private AudioClip ASound;
}
```

Figura 42.

Clase implementada en la pistola de pelotas de tenis.

```
public class Gun : MonoBehaviour
{
    public Transform spawnPoint; // Instanciar posición inicial de la bala
    public GameObject bullet; // Objeto bala
    public float shotforce = 1000f; // Velocidad de la bala
    public float shotRate = 0.5f; // Cooldown entre cada disparo
    private float shotRateTime = 0; // Contador para el shot rate
    private AudioSource audioSource; // Referencia del audio source
    public AudioClip shotSound; // Referencia del audio clip
}
```

Figura 43.

Clase implementada en los enemigos.

```
public class EnemyScriptX : MonoBehaviour
{
    [SerializeField] GameObject explosionPrefab; // Referencia a una explosión predefinida

    float lifeTimeExplosion = 3.0f; // Timer de explosión
    public EnemyUI eu; // Referencia a la clase UI del enemigo
    private Puntaje puntajePj; // Referencia a la clase puntaje
    GameObject moneda; // Referencia al objeto moneda

    int balas_disparadas = 0; // Variable como contador de balas
    [SerializeField] int balas_necesarias; // Variable para límite de balas
    [SerializeField] private AudioClip AmmoSound; // Referencia a un audio
}
```


- Técnico de sonido.

Se realizó el mismo procedimiento detallado en la iteración 1, se integraron sonidos en la pistola de pelotas de tenis, generador de vida y enemigos.

3.5.3.6. *Elaboración de iteración 6.*

Esta iteración se realiza con el objetivo de implementar código necesario para activación de imágenes en la UI, como también el control de paneles encargados de visualizar los objetivos restantes.

a) **Planificación de la iteración.**

Tabla 33.

Lista de tareas iteración 6

Ítem	Descripción	Encargado
19.	Añadir código a los objetos de la UI para activación y desactivación de imágenes.	AA
19.1.	Seleccionar una imagen que represente una llave.	AA
19.2.	Añadir código que establezca bajo que circunstancias debe estar visible y activa la llave.	AA
20.	Enlazar acciones entre UI y Player por medio de código.	AA
20.1.	Crear métodos que permitan detección de objetos cercanos.	AA
21.	Control de paneles dependiendo de decisiones del jugador.	AA
21.1.	Implementar métodos en la clase que permitan el cambio de texto en el panel principal.	AA
21.2.	Añadir código dedicado al control de objetivos, los cuales son necesarios para que aparezca la llave.	AA

b) **Desarrollo de características.**

- Diseñador Gráfico.

Tabla 34.

Implementar llave en la interfaz gráfica.

Paso	Procedimiento
1	Seleccionar imagen de la llave.
2	En propiedades de la imagen dar clic en Texture types.
3	Seleccionar la opción "Sprite 2D and UI"
4	Agregar un objeto en el canvas que contenga el componente "image".
5	Seleccionar "Source image".
6	Ubicar la llave de forma estratégica en la UI principal.

Figura 44.

Llave activada en la UI principal.



Figura 45.

Llave ubicada en el entorno 3D.



- Programador.

Tabla 35.

Implementar código de control para contabilizar objetivos.

Paso	Procedimiento
1	Implementar métodos que permitan detectar objetos cercanos.
2	Agregar un método que verifique la dificultad.
3	Crear métodos que permitan modificar el texto del panel principal.
4	Crear métodos para contabilizar los objetivos a ser cumplidos.
5	Implementar código para visualización de la llave.
6	Añadir código para que los objetivos ubicados en la casa aparezcan de forma aleatoria.

Figura 46.

Método para contabilizar objetivos.

```

if (other.gameObject.CompareTag("objetive"))//Compara si el tag es igual a objetive
{
    SoundManager.InstanceS.EjecutarSonido(esferaSound);//Ejecuta un sonido
    Destroy(other.gameObject); //Se destruye el objeto
    GameManager.Instance.numDeObjetivos--;//Decrece el número de objetivos
    moneda.GetComponent<Animator>().SetTrigger("Xm");//Activa animación de moneda
    monedaX.GetComponent<Animator>().SetTrigger("Color");//Activa animación de color
    yield return new WaitForSeconds(0.5f); //Cooldown de 0.5 Segundos
    textMision.text = "Obtén las esferas rojas" + "\n Restantes:"
        + GameManager.Instance.numDeObjetivos;//Indica las esferas restantes
}

```

3.5.3.7.Elaboración de iteración 7.

El propósito de esta iteración es permitir que el jugador tenga control sobre la activación y desactivación de puertas, además se añade la mecánica de ataque de enemigos, también se agrega la visualización de pérdida de vida y navegación de enemigos.

a) Planificación de la iteración.

Tabla 36.

Lista de tareas iteración 7.

Ítem	Descripción	Encargado
22.	Cuando el jugador interactúe con puertas, incluir código que permita controlar la activación y desactivación de estas.	AA
22.1.	Agregar método “On Trigger Stay” a la puerta para la detección de objetos según su collider.	AA
22.2.	Agregar código para la desactivación de puertas cuando el jugador se aleje de ellas.	AA
23.	Añadir código para la interacción entre enemigo y jugador.	AA
23.1.	Agregar métodos a las clases creadas en la iteración 5 que permitan detectar distancia entre el enemigo y el jugador.	AA
23.2.	Agregar métodos en la clase “Attack Enemy” que habiliten la mecánica de ataque del enemigo y permitan el decrecimiento de vida cuando colisionen con los disparos del jugador.	AA
24.	Añadir código de navegación que restrinja el movimiento del enemigo a las zonas designadas.	AA
24.1.	Crear una malla de navegación para que los enemigos recorran el mapa solo por sitios ya preestablecidos.	AA

b) Desarrollo de características.

- Diseñador Gráfico.

El diseñador gráfico es encargado de realizar el siguiente procedimiento detallado en la

Tabla 37 para facilitar la apertura de puertas.

Tabla 37.

Animación de puertas.

Paso	Procedimiento
1	Ubicar las puertas en los pasillos internos de la casa.
2	Agregar color de marco, puerta y perilla.
3	Crea animación de la puerta abriéndose.
4	Agregar la animación creada a todas las puertas.
5	Verificar funcionamiento individual de cada puerta.

Figura 47.

Animación de apertura de puerta.



- Programador.

Tabla 38.

Control de colisiones y navegación de enemigos.

Paso	Procedimiento
1	Agregar box collider a todas las puertas creadas.
2	Implementar métodos en script "On Trigger Stay" para controlar activación de puertas dependiendo si el jugador está cerca de las mismas a través de su etiqueta.
3	Crear métodos para calcular distancia entre enemigo y jugador, según la distancia el enemigo persigue al jugador.
4	Crear método que posibilite contabilizar los enemigos derrotados.
5	Agregar código que verifique si el enemigo colisiona con el jugador y este pierda vida por cada colisión recibida.
6	Incluir métodos que permita la destrucción de enemigos en la escena.
7	Para construir una malla de navegación en la que algunos oponentes naveguen constantemente, añadir métodos al script "IA".

Figura 48.

Creación del componente collider.

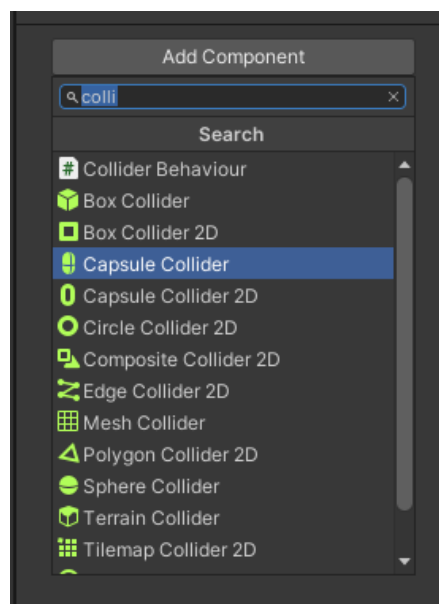


Figura 49.

Clase IA.

```
public class IA : MonoBehaviour
{
    public NavMeshAgent navMeshAgent; //Referencia para manejar el destino de navegación
    public Transform[] destinationsV; //Array para manejar destinos en el mapa
    private float distanceToFollowP = 2; //Modifica distancia de un punto a otro
    private int i = 0; //Indicador de puntos para el vector
    public bool followPlayer = false; //Bandera para activar malla
    private GameObject player; // Referencia al jugador

    [Header("-----Follow Player?-----")] //header para el inspector, indica de una mejor manera los atributos de un script

    private float distanceToPlayer; //Conoce la distancia entre enemigo y jugador
    public float distanceToFollowPlayer; //Distancia mínima para seguir al jugador
}
```

3.5.3.8. Elaboración de iteración 8.

Esta iteración se desarrolló con la finalidad de detallar las tareas y subtareas que se llevaron a cabo para lograr completar el sistema de preguntas que va a contener cada puerta.

a) Planificación de la iteración.

Tabla 39.

Lista de tareas iteración 8.

Ítem	Descripción	Encargado
25.	Diseño de la interfaz del Quiz.	AA
25.1.	Realizar boceto en papel de la interfaz gráfica del Quiz.	AA
25.2.	Diseñar interfaz “Quiz” en la UI principal.	AA
26.	Agregar código de control para preguntas y respuestas.	AA
26.1.	Crear clases para el manejo de preguntas y respuestas obtenidas en el quiz.	AA
27.	Validación de preguntas correctas e incorrectas y agregar acciones a la puerta según la respuesta.	AA
27.1.	Crear clase para verificar las preguntas correctas e incorrectas.	AA

b) Desarrollo de características.

- Diseñador Gráfico.

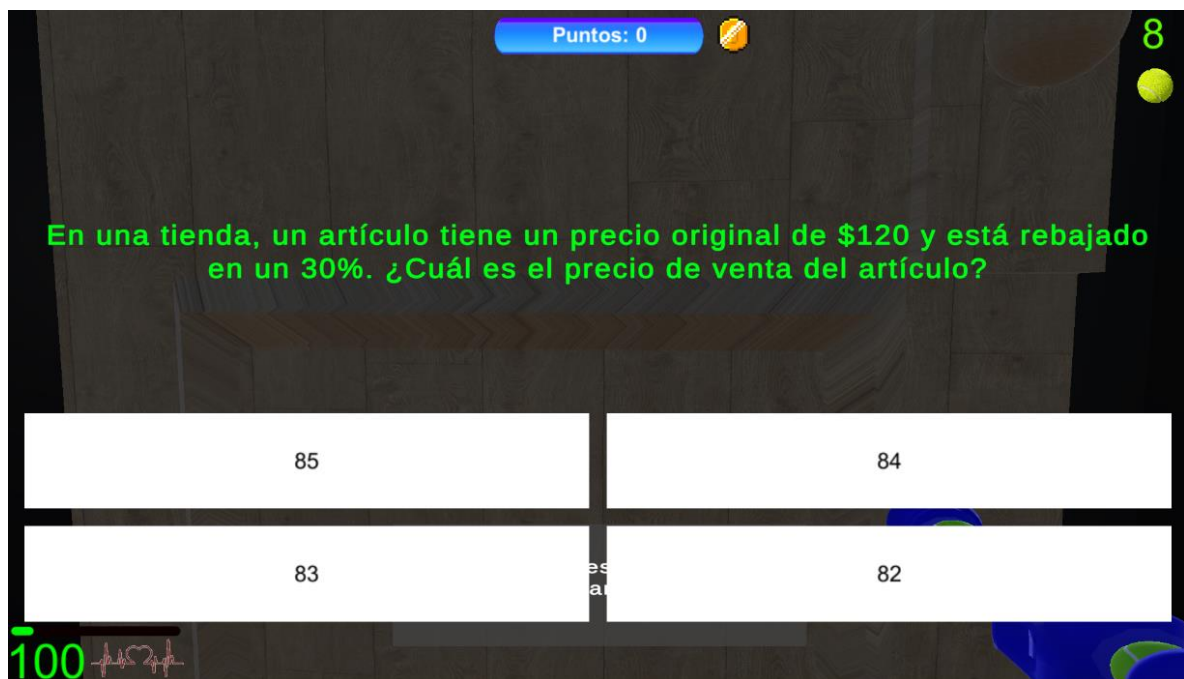
Tabla 40.

Integrar Quiz en la UI principal.

Paso	Procedimiento
1	Realizar los pasos de la tabla 19.
2	Agregar botones en la UI principal.
3	Ubicar de manera intuitiva los botones de selección para responder.
4	Agregar colores a los botones que posteriormente sean modificados con código.
5	Verificar la funcionalidad de los botones y la activación del quiz.

Figura 50.

Quiz en la UI principal.



- Programador.

El programador se encargó de escribir código que verifique respuestas correctas e incorrectas y, en función de los resultados, el jugador sea capaz de moverse por toda la casa.

Tabla 41.

Creación del quiz.

Paso	Procedimiento
1	Crear clases que permitan el manejo de preguntas.
2	Incluir código para gestionar la respuesta a la consulta de cada pregunta.
3	Agregar métodos para validar preguntas correctas e incorrectas.
4	Implementar código que se ejecute dependiendo si la respuesta del jugador fue correcta o incorrecta.
5	Validar, si la respuesta es correcta, se ejecuta la animación encargada de abrir la puerta además el cuadro de opción cambia a color verde. Todo se realiza a través de código.
6	Verificar funcionalidad de botones, activación y desactivación de quiz.

Figura 51.

Clase implementada para el control del quiz.

```
public class QuizzUI : MonoBehaviour
{
    [SerializeField] private TextMeshProUGUI questionText = null; //Referencia al texto de la pregunta
    [SerializeField] private List<OptionsButton> optionsButtonsList = null; //referencia a una lista de las opciones del botón

    public void Construct(Question q, Action<OptionsButton> callbackOB) //Retorna un valor OptionButton para verificar lo ingresado en los botones
    {
        questionText.text = q.text; //se le asigna el texto de la clase Question
        for(int n=0;n<optionsButtonsList.Count;n++) //Recorrido de la lista hasta el límite de la misma
        {
            optionsButtonsList[n].Construct(q.optionsList[n],callbackOB); //Se almacenan las preguntas en el arreglo
        }
    }
}
```

3.5.3.9.Elaboración de iteración 9.

a) Planificación de la iteración.

Tabla 42.

Lista de tareas iteración 9.

Ítem	Descripción	Encargado
28.	Añadir partículas cuando el jugador pierda vida.	AA
29.	Añadir explosiones cuando los enemigos sean derrotados y cada vez que colisionen con el jugador juntamente con sonidos.	AA
29.1.	Crear explosiones según la interacción del jugador.	AA
29.2.	Incorporar sonidos a explosiones y partículas.	AA
30.	Mejora del entorno 3D, modelos y texturas.	AA
30.1.	Importar nuevas texturas y corregir fallos en movimiento de objetos.	AA

b) Desarrollo de características.

- Diseñador Gráfico

Tabla 43.

Importación de texturas.

Paso	Procedimiento
1	Importar texturas como se detalló en la iteración 6.
2	Agregar nuevos objetos con las texturas importadas.
3	Comprobar errores en las texturas al ser cargadas en cada objeto.
4	Verificar paredes mal posicionadas en el entorno 3D.

- Programador.

Tabla 44.

Control de explosiones.

Paso	Procedimiento
1	Crear clases que permitan el manejo de explosiones.
2	Incluir código para controlar en qué momento deben aparecer las explosiones.
3	Incluir partículas para representar la pérdida de vida en el jugador.
4	Implementar código que maneje el audio de cada explosión.
5	Validar si las explosiones no se repiten y que se destruyan una vez terminada su función.

- Técnico de sonido.

Se realizó el mismo procedimiento detallado en la iteración 1, se integraron sonidos en explosiones y en la aparición de partículas.

3.5.4. Beta. (Fase 4).

Esta fase de la metodología SUM se centra en la verificación y corrección de errores del videojuego, se deben hacer pruebas y reportar todos los errores encontrados, también se verifica que el gameplay funcione de manera fluida y correcta.

3.5.4.1. Distribución 1

Aspectos para verificar:

- Errores
- Jugabilidad
- Entretenimiento
- Dificultad

Evaluación y verificación de errores.

En la **Tabla 45**, se pueden visualizar los errores encontrados y recomendaciones dadas por el verificador Beta para mejorar el prototipo.

Tabla 45.

Evaluación del verificador beta - Distribución 1.

Tipo de error	Descripción del error	Aspectos para verificar
Programación	La navegación del jugador se ve afectada por el control de cámara	Errores
Programación	Mal direccionamiento entre escenas	
Diseño	Paneles de información, mal posicionados y funcionamiento incorrecto	
Diseño	Hay objetos en el entorno 3D que carecen de colliders	
Sonido	La pista de fondo no tiene un control correcto de audio	
Diseño	El entorno 3D es muy grande para el jugador	Jugabilidad
Diseño	Las imágenes de fondo de cada escena no son llamativas	Entretenimiento
Diseño	Los carteles informativos no dan instrucciones de juego	Dificultad

Lista de cambios realizados.

A continuación, se detallan los cambios que se realizaron en la primera distribución:

Tabla 46.

Lista de cambios - Distribución 1.

Descripción	Horas	Encargado
Se implementó método para controlar cada cámara individualmente	1	AA
Cambiar el código de cada botón para mejorar la navegación entre escenas	1	AA
Se ajustó el alto y ancho de pantalla y se corrigió su funcionamiento	1	AA
Cambio de colliders, se agregó “box collider” a todos los objetos cuadrados y rectangulares.	1	AA
Se agregó un componente para controlar el audio de fondo	1	AA

Se cambió las dimensiones del objeto “Player”	1	AA
Cambio de imágenes, se integró imágenes visualmente atractivas y divertidas.	1	AA
Se agregó instrucciones para una mejor orientación del jugador	1	AA
Total	8	AA

3.5.4.2. Distribución 2

Aspectos para verificar:

- Errores
- Jugabilidad
- Dificultad
- Aprendizaje

Evaluación y verificación de errores.

En esta distribución el verificador beta identificó una lista de errores y proporcionó información detallada para la corrección y mejora del videojuego 3D, los errores se describen a continuación:

Tabla 47.

Evaluación del verificador beta - Distribución 2.

Tipo de error	Descripción del error	Aspectos para verificar
Programación	Incorrecto control de mínimo y máximo de vida del jugador.	Errores
Programación	No se suman las municiones recogidas al contador	
Programación	Los objetos restauradores de vida no agregan vida al jugador	
Diseño	La vida de los enemigos no decrece cuando reciben daño	

Programación	Cuando la vida de los enemigos es igual a 0 no desaparecen	
Sonido	No existe sonido de disparo y explosiones	
Diseño	No se diferencian los objetos internos de la casa debido a sus colores	Jugabilidad
Programación	No se visualiza el número total de objetivos	
Programación	Los enemigos se desplazan de manera muy lenta	Dificultad
Diseño	Se recomienda incluir instrucciones para manejo del personaje	Aprendizaje

Lista de cambios realizados.

A continuación, en la **Tabla 48** se detallan los cambios realizados en la distribución 2.

Tabla 48.

Lista de cambios - Distribución 2.

Descripción	Horas	Encargado
Modificar el método para control de vida	1	AA
Cambio de lógica en la clase controladora de municiones	1	AA
Cambio de lógica en la clase que controla los restauradores de vida	1	AA
Se optó por incluir un slider para representar la vida del enemigo	1	AA
Cambio de toda la clase que controla la vida del enemigo	1	AA
Se importó sonidos para disparos y explosiones	1	AA
Cambio de colores de todos los objetos internos de la casa	1	AA
Actualizar método para identificar los objetivos a ser cumplidos	1	AA
Cambio de método para que los enemigos se desplacen más rápido.	1	AA
Diseño y elaboración de carteles informativos	1	AA
Total	10	AA

3.5.4.3.Distribución 3

Aspectos para verificar:

- Errores
- Jugabilidad
- Entretenimiento
- Dificultad
- Aprendizaje

Evaluación y verificación de errores.

Posteriormente, se presenta una tabla se describen los errores o bugs encontrados en el videojuego, los cuales fueron reportados por el verificador beta durante la distribución 3.

Tabla 49.

Evaluación del verificador beta - Distribución 3.

Tipo de error	Descripción del error	Aspectos para verificar
Programación	Modificar el controlador de apertura de puertas	Errores
Programación	El enemigo no persigue correctamente al jugador	
Programación	El quiz no reconoce si la respuesta es correcta o incorrecta	
Diseño	No existe animación cuando colisionan los enemigos	
Sonido	No existe sonido de explosión	
Programación	El jugador no puede correr en el entorno 3D	Jugabilidad
Diseño	Animaciones sencillas y poco llamativas con los objetos interactivos	Entretenimiento
Ninguna	No hay sugerencias	Dificultad
Programación	Las preguntas incluidas en las puertas deben ser aleatorias	Aprendizaje

Lista de cambios realizados.

Tabla 50.

Lista de cambios - Distribución 3.

Descripción	Horas	Encargado
Cambio de lógica de la clase que permite la apertura de puertas	5	AA
Cambio de lógica en la clase que controla al enemigo	1	AA
Modificar el método que reconoce las respuestas correctas	1	AA
Se agregó animación de explosión cuando los enemigos colisionan	1	AA
Se importó sonidos para explosiones	1	AA
Se implementó un método que permita al jugador correr por un determinado tiempo	1	AA
Se mejoró las animaciones en todos los objetos interactivos	1	AA
Se cambió toda la clase que controla la visualización de preguntas en cada puerta	3	AA
Total	14	AA

3.5.5. Cierre. (Fase 5).

Con el propósito de socializar el videojuego en la Unidad Educativa “César Antonio Mosquera”, fue necesario llevar a cabo pruebas antes y después de su implementación, con el fin de evaluar el efecto ocasionado en los estudiantes revisar **Anexo B** donde se detallan las preguntas de la evaluación.

Para medir los promedios finales se tomó en cuenta la **Tabla 51** de acuerdo con el sistema de calificaciones del Ecuador.

Tabla 51.

Escala de calificaciones de acuerdo con MINEDU

Escala cualitativa	Escala cuantitativa
Domina los aprendizajes requeridos	9,00-10,00
Alcanza los aprendizajes requeridos.	7,00-8,99
Está próximo a alcanzar los aprendizajes requeridos	4,01-6,99
No alcanza los aprendizajes requeridos	≤ 4

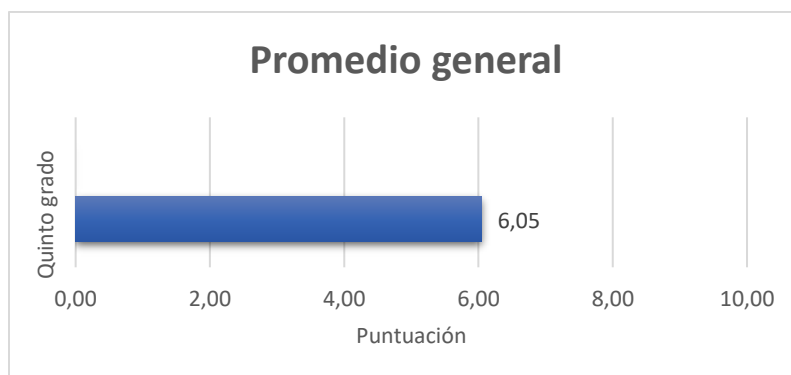
3.5.5.1.Pruebas.

- **Prueba 1**

Con el fin de medir el impacto que ha tenido el videojuego en los niños de quinto grado de educación básica de la Unidad Educativa, se realizó una evaluación de 10 preguntas (Anexo 1), la cual fue elaborada en colaboración con docentes de la Unidad Educativa, además estas preguntas se incluyeron en el videojuego, esta evaluación se aplicó a un grupo de 44 estudiantes de quinto grado y se obtuvieron los resultados de la **Figura 52**.

Figura 52.

Promedio de encuestados antes de utilizar el videojuego.



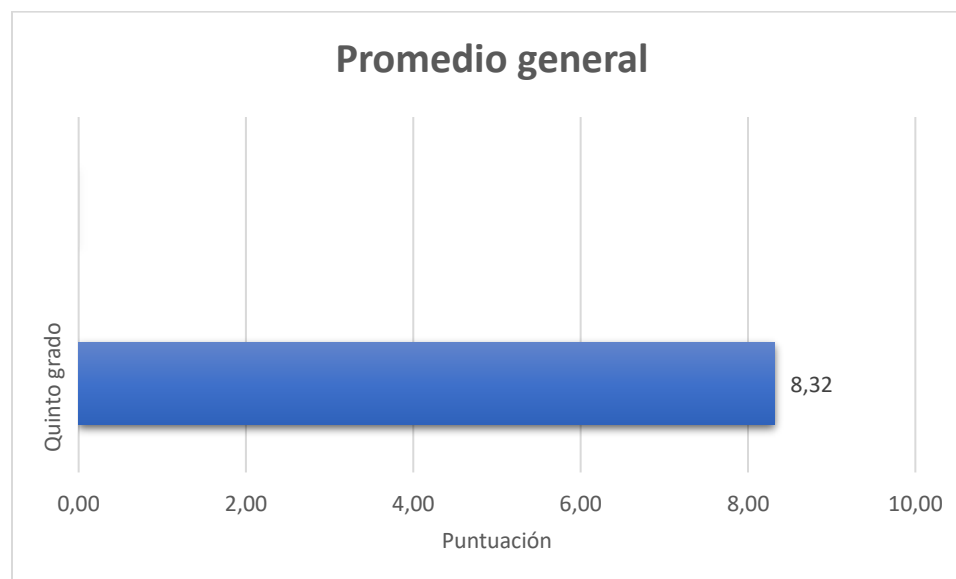
La **Figura 52** demuestra que los estudiantes tienen problemas en la resolución de problemas matemáticos y operación con números naturales, de acuerdo con la escala de valoración representada en la **Tabla 51**, el promedio de los niños de quinto grado los posiciona en: “Están próximos a alcanzar los aprendizajes requeridos”.

- **Prueba 2**

El videojuego fue utilizado alrededor de 5 días en cada salón de clase por un periodo mínimo de 20 minutos, el docente proyectó desde su computador los problemas matemáticos y fueron resueltos individualmente, además el docente hizo una retroalimentación de todas las preguntas mal respondidas. Para finalizar se evaluó nuevamente a los 44 estudiantes obteniendo lo que se presenta en la **Figura 53**.

Figura 53.

Promedio de encuestados después de la implementación del videojuego



En la **Figura 53** se visualizan los resultados obtenidos luego de realizar la implementación del videojuego 3D educativo como herramienta de aprendizaje, de acuerdo con la **Tabla 51**, el promedio de los niños de quinto grado los posiciona en: “Alcanzan los aprendizajes requeridos.”.

Conclusión

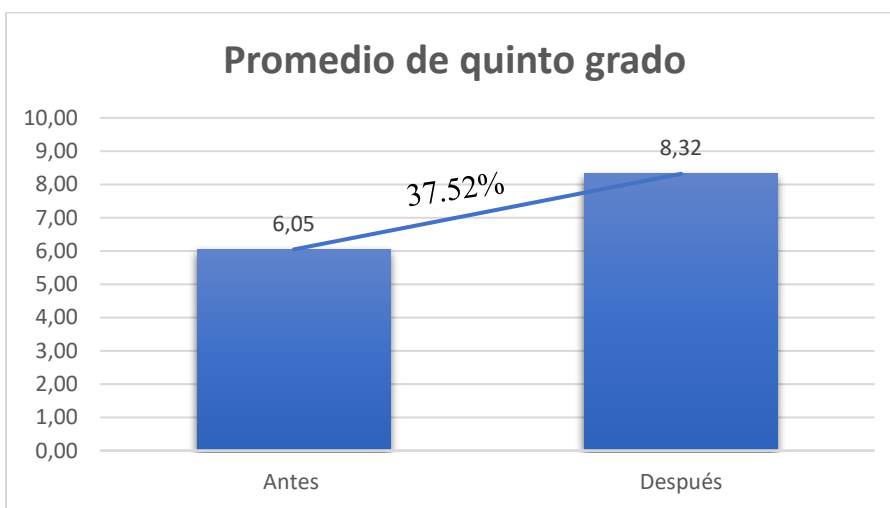
Con el fin de medir el impacto que tuvo el videojuego 3D como herramienta didáctica para fortalecer el aprendizaje de matemáticas, se compararon los resultados obtenidos en las dos evaluaciones. Es necesario resaltar que se utilizó la fórmula de mejora porcentual tomando como referencia la nota inicial y ver el cambio en la segunda nota.

Fórmula de la mejora porcentual: $((\text{Valor Final} - \text{Valor inicial}) / \text{Valor inicial}) * 100$

Al aplicar la fórmula se obtuvieron los resultados que se muestran en la Figura 54.

Figura 54.

Porcentaje de mejora en los estudiantes de quinto grado.



Finalmente, se concluye que el videojuego 3D demostró ser efectivo al mejorar las competencias matemáticas en estudiantes de educación básica en un **37,52%**, validando su viabilidad como herramienta educativa e incentivando a que los niños aprendan con videojuegos académicos.

CAPÍTULO 4.

4.1. Validación de resultados mediante la ISO 25010

Un miembro de la familia de normas ISO 25000 es ISO/IEC 25010. Se trata de una norma centrada en la usabilidad que establece los factores de calidad que deben tenerse en cuenta al evaluar los atributos de un producto de software terminado. El grado en que un producto de software satisface las necesidades de sus clientes, añadiendo valor, se caracteriza como la calidad del producto de software. Se trata de calibrar la calidad del producto de software(Andrés et al., 2017).

Figura 55.

Modelo de calidad del producto software según la ISO/IEC 25010



4.2. Propósito de la evaluación.

El propósito de la evaluación del producto de software académico, cuyo objetivo es mejorar la comprensión de las matemáticas por parte de los alumnos de educación básica, es confirmar que cumple los requisitos de calidad establecidos en la norma ISO/IEC 25010 con respecto a las subcaracterísticas de usabilidad.

Las subcaracterísticas de usabilidad que se evaluaron se muestra en la **Tabla 52**:

Tabla 52.

Subcaracterísticas de usabilidad a ser evaluadas.

Subcaracterísticas de usabilidad.	Simbología de subcaracterísticas
Capacidad para reconocer su adecuación	CapAd
Capacidad de aprendizaje	CapAp
Capacidad para ser usado	CapUs
Protección contra errores de usuario	ProtErr
Estética de la interfaz de usuario	EstInt
Accesibilidad	Acc

4.3. Módulos de evaluación.

Para realizar la evaluación de usabilidad del videojuego educativo 3D, según las subcaracterísticas de usabilidad se asignó un nivel de importancia definido en la **Tabla 53**:

Tabla 53.

Nivel de importancia para valorar cada subcaracterística de la usabilidad.

Nivel de importancia.	Simbología	Significado
Alto	A	El grado de importancia es alto en la característica y subcaracterística, por consiguiente, se procedió a realizar las mediciones.
Medio	M	El nivel de importancia de la característica y subcaracterística no es tan relevante, pero sí puede ser medida.
Bajo	B	La característica y subcaracterística carece de relevancia por lo tanto no fue medida.
No Aplica	NA	Este valor se asigna a la característica y subcaracterística que no se pueden medir dependiendo de diversos factores.

Fuente: (Sifuentes & Peralta, 2022)

A continuación, en colaboración con los docentes, se da una valoración a las subcaracterísticas de usabilidad del producto de software académico, esta evaluación sirve de base para determinar las ponderaciones necesarias para medir la usabilidad del producto de software académico, en este caso el videojuego 3D educativo.

Tabla 54.

Valoración de las subcaracterísticas de usabilidad.

Subcaracterísticas de usabilidad.	Nivel de importancia
Capacidad para reconocer su adecuación	M
Capacidad de aprendizaje	A
Capacidad para ser usado	M
Protección contra errores de usuario	A
Estética de la interfaz de usuario	A
Accesibilidad	B

Nota: Se valora según su nivel de importancia.

Se elaboró un test de usuario que sirvió como instrumento para medir usabilidad considerando cada subcaracterística y sus indicadores de acuerdo con la norma ISO/IEC 25010.

Tabla 55.*Indicadores para la medición de usabilidad.*

Subcaracterísticas de usabilidad.	Indicador
Capacidad para reconocer su adecuación	¿Cree usted que el videojuego explica de forma clara los objetivos que deben ser cumplidos? ¿Piensa usted que el videojuego tiene instrucciones de lo que debe hacer para ganar?
Capacidad de aprendizaje	¿Considera usted que las instrucciones del videojuego explicaron todo lo necesario para jugar? ¿Cree usted que el videojuego incluye pistas o ayudas para cumplir el objetivo final?
Capacidad para ser usado	¿Piensa usted que el videojuego le comunica(avisa) cuando está haciendo algo incorrecto? ¿Piensa usted que el videojuego da consejos para evitar errores y avanzar hasta el final?
Protección contra errores de usuario	¿Piensa usted que el videojuego es fácil de entender para alguien que nunca lo ha jugado antes? ¿Cree usted que, al escribir su nombre en el videojuego, se le avisa si comete un error?
Estética de la interfaz de usuario	¿Le resultó fácil cambiar brillo, volumen y tamaño de pantalla en las opciones del videojuego? ¿Le resultó fácil cambiar el color de letras que aparecen en el videojuego?
Accesibilidad	Debido a que esta subcaracterística se valoró con un nivel bajo, no se consideró en la medición.

4.3.1. Criterios de decisión para la evaluación de métricas.

Para lograr la medición de usabilidad del producto de software académico: Videojuego 3D utilizando la metodología más adecuada que fortalezca el aprendizaje de Matemáticas en niños de 5to grado de Educación Básica de la Unidad Educativa “César Antonio Mosquera”. Se aplicó un Test de usuario utilizando la escala de Likert de 1 a 4 detallada a continuación en la **Tabla 56**:

Tabla 56.

Criterios de decisión aplicados para la calificación de métricas.

Escala	Puntuación	Criterio de decisión para las métricas.
1	0 puntos	Bajo
2	2 puntos	Regular
3	3 puntos	Bueno
4	5 puntos	Excelente

Fuente: (Sifuentes & Peralta, 2022).

Para evaluar cada subcaracterística se establecieron criterios de evaluación, en base al puntaje resultante de la suma de dos indicadores que conforman cada subcaracterística. Estos criterios son detallados en la **Tabla 57**:

Tabla 57.

Criterios de decisión para evaluar las subcaracterísticas de usabilidad.

Rangos de puntuación	Criterio de decisión para evaluar subcaracterísticas.
[10 - 9] puntos	Alta Calidad
]9 - 6] puntos	Buena Calidad
]6 - 2] puntos	Regular Calidad
]2 - 0] puntos	Mala Calidad

Fuente: (Sifuentes & Peralta, 2022).

Tomando en cuenta los valores asignados a cada subcaracterística de la usabilidad y conforme a los valores obtenidos al sumar los dos indicadores, se presenta la siguiente tabla de ejemplo. Según:(Sifuentes & Peralta, 2022) se procede a elaborar la evaluación final de cada test considerando la **Tabla 58** como punto de referencia:

Tabla 58.

Valoración a nivel de subcaracterísticas según la calificación de métricas.

Subcaracterística de usabilidad	Indicador	Valoración a nivel de subcaracterísticas		
		calificación según métricas	Sumar valores de indicadores	Criterio de evaluación
Capacidad para reconocer su adecuación	¿Cree usted que el videojuego explica de forma clara los objetivos que deben ser cumplidos?	2	5	Regular Calidad
	¿Piensa usted que el videojuego tiene instrucciones de lo que debe hacer para ganar?	3		
Capacidad de aprendizaje	¿Considera usted que las instrucciones del videojuego explicaron todo lo necesario para jugar?	3	8	Buena Calidad
	¿Cree usted que el videojuego incluye pistas o ayudas para cumplir el objetivo final?	5		
Capacidad para ser usado	¿Piensa usted que el videojuego le comunica(avisa) cuando está haciendo algo incorrecto?	5	8	Buena Calidad
	¿Piensa usted que el videojuego da consejos para evitar errores y avanzar hasta el final?	3		
Protección contra errores de usuario	¿Piensa usted que el videojuego es fácil de entender para alguien que nunca lo ha jugado antes?	5	10	Alta Calidad
	¿Cree usted que, al escribir su nombre en el videojuego, se le avisa si comete un error?	5		
Estética de la interfaz de usuario	¿Le resultó fácil cambiar brillo, volumen y tamaño de pantalla en las opciones del videojuego?	5	10	Alta Calidad

¿Le resultó fácil cambiar el color de 5
letras que aparecen en el videojuego?

Nota: Se suma los valores de los indicadores de cada subcaracterística.

4.3.2. Criterios de decisión para la evaluación general del videojuego.

Con el fin de obtener la medida general de la característica de usabilidad, se procedió a calcular el valor total de acuerdo con los niveles de importancia que fueron asignados en la **Tabla 54**. Se multiplican por 1 las subcaracterísticas valoradas con un nivel de importancia Medio(M) y por 2 las subcaracterísticas valoradas con un nivel de importancia Alto(A) luego se reemplaza en la siguiente fórmula que se puede visualizar en la **Tabla 59**:

Tabla 59.

Ecuación 1 para medir la usabilidad en productos de software.

$$\text{Medida usabilidad} = 1*\text{CapAd} + 1*\text{CapAp} + 2*\text{CapUs} + 2*\text{ProtErr} + 2*\text{EstInt}$$

Fuente: (Sifuentes & Peralta, 2022).

A continuación, se detalla en la **Tabla 60** los rangos para la evaluación final del videojuego 3D, basado en los criterios de decisión y el puntaje obtenido al aplicar la ecuación 1.

Tabla 60.

Criterio de decisión para evaluar la característica de usabilidad.

Rango de puntuación	Criterio de decisión para evaluar la característica de usabilidad
[80 -70] puntos	Alta Calidad
[69 - 40] puntos	Buena Calidad
[39 - 20] puntos	Regular Calidad
[19 - 0] puntos	Mala Calidad

Nota: El videojuego se pondera en función del rango obtenido.

Tabla 61.

Tabla de referencia para calcular la valoración final.

Sub característica de usabilidad	Indicador.	Valoración a nivel de subcaracterísticas			Valoración a nivel de característica de usabilidad		
		Puntos	Suma	Valor	Nivel de	Aplicación	Valoración final de acuerdo con los criterios de evaluación general.
					Importancia	de Ecuación	
Capacidad para reconocer su adecuación	¿Cree usted que el videojuego explica de forma clara los objetivos que deben ser cumplidos?	2	5	Regular Calidad	M	69	Buena Calidad
	¿Piensa usted que el videojuego tiene instrucciones de lo que debe hacer para ganar?	3			1*5		
Capacidad de aprendizaje	¿Considera usted que las instrucciones del videojuego explicaron todo lo necesario para jugar?	3	8	Buena Calidad	A		
	¿Cree usted que el videojuego incluye pistas o ayudas para cumplir el objetivo final?	5			2*8		
Capacidad para ser usado	¿Piensa usted que el videojuego le comunica(avisa) cuando está haciendo algo incorrecto?	5	8	Buena Calidad	M		1*8
	¿Piensa usted que el videojuego da consejos para evitar errores y avanzar hasta el final?	3					
Protección contra errores de usuario	¿Piensa usted que el videojuego es fácil de entender para alguien que nunca lo ha jugado antes?	5	10	Alta Calidad	A		2*10
	¿Cree usted que, al escribir su nombre en el videojuego, se le avisa si comete un error?	5					

Estética de la interfaz de usuario	¿Le resultó fácil cambiar brillo, volumen y tamaño de pantalla en las opciones del videojuego?	5	10	Alta Calidad	A 2*10
	¿Le resultó fácil cambiar el color de letras que aparecen en el videojuego?	5			

Nota: Se aplicó la ecuación 1 y se posiciona de acuerdo con la Tabla 60.

4.4. Evaluación por pregunta.

Para realizar la medición de usabilidad del videojuego 3D se evaluó a 54 personas, 50 estudiantes y 4 profesores con el test:(Revisar Anexo A).

Obteniendo los siguientes resultados detallados en la **Tabla 62**:

Tabla 62.

Resultados del test por pregunta.

Nro. Pregunta	Bajo	Regular	Bueno	Excelente
1	0	1	12	41
2	1	2	10	41
3	0	1	13	40
4	2	1	9	42
5	0	8	10	36
6	1	1	14	38
9	2	0	12	40
8	1	4	12	37
7	0	2	13	39
10	1	3	8	42

La respuesta de cada pregunta se realizó en la escala de Likert de (1 – 4), de acuerdo con la **Tabla 56**. se asignó una imagen a cada valoración para que los niños comprendan mejor las opciones de respuesta:

Figura 56.

Escala de Likert adecuada para evaluar niños.



Tabla de ejemplo.

4.4.1. Análisis de los resultados.

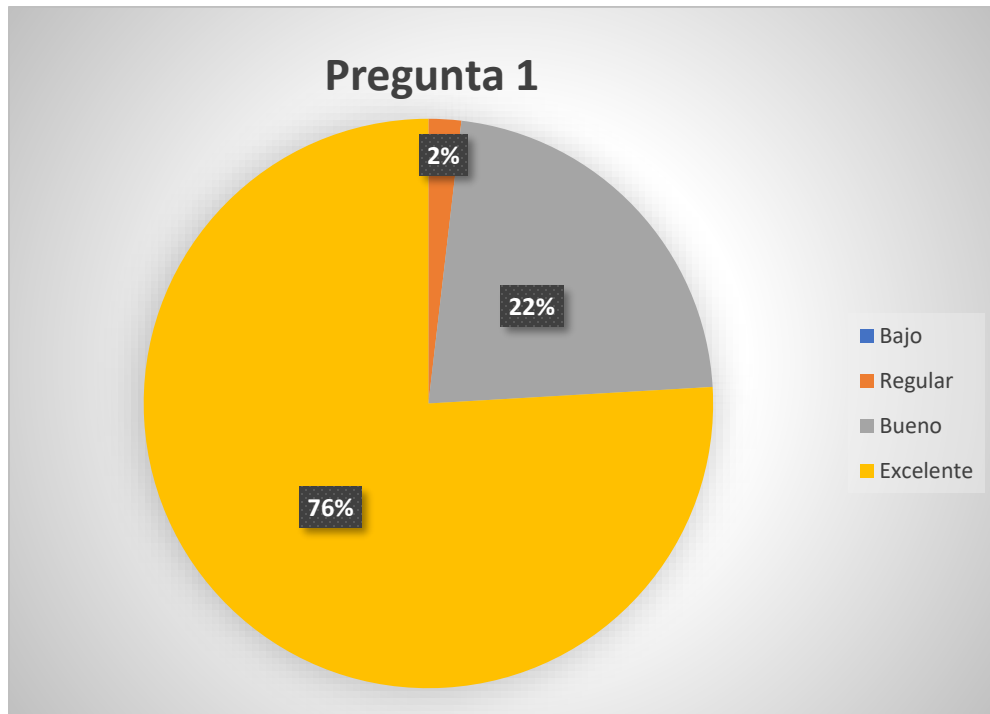
A continuación, se presentan los resultados obtenidos por cada pregunta del test aplicado a estudiantes y docentes de la unidad educativa.

Pregunta 1: ¿Cree usted que el videojuego explica de forma clara los objetivos que deben ser cumplidos?

En la **Figura 57.** correspondiente a la pregunta 1 se puede visualizar que la opción más votada fue “Excelente” con un 76% de los votos, mientras que la segunda opción votada fue con una calificación de “Bueno” con un 22% de los votos y con una valoración de “Regular” solo el 2% del total.

Figura 57.

Resultados de la pregunta 1 del test de usabilidad



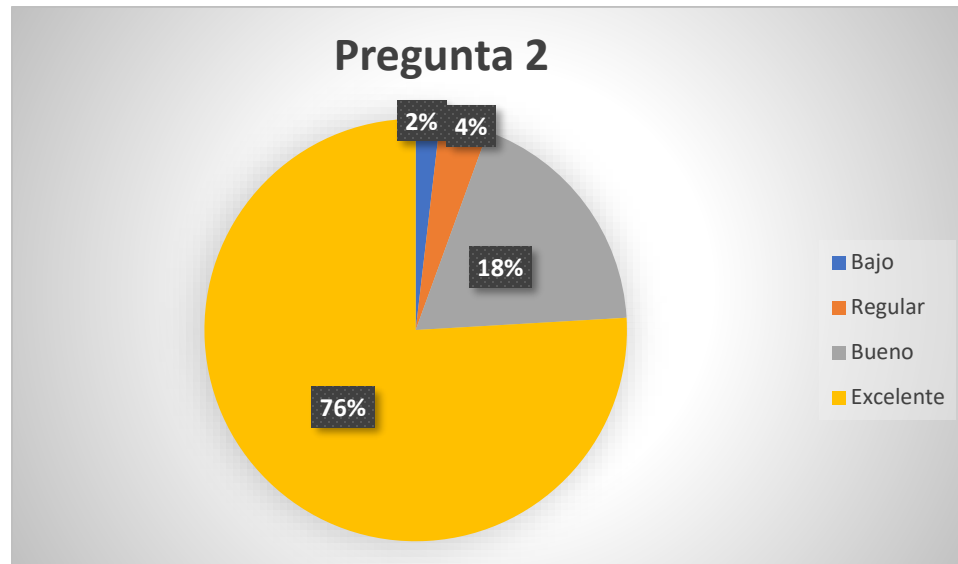
Este resultado muestra que el videojuego 3D explicó de forma clara los objetivos que deben ser cumplidos, indicando que la mayoría de usuarios no tienen problemas para comprender lo que se espera de ellos en el videojuego.

Pregunta 2: ¿Piensa usted que el videojuego tiene instrucciones de lo que debe hacer para ganar?

En la **Figura 58**, correspondiente a la pregunta 2 se puede visualizar que la opción más votada fue “Excelente” con un 76% de los votos, mientras que la segunda opción votada fue con una calificación de “Bueno” obteniendo 18% de los votos, con una valoración de “Regular” se obtuvo el 4% y el 2% restante con valoración “Baja”.

Figura 58.

Resultados de la pregunta 2 del test de usabilidad.



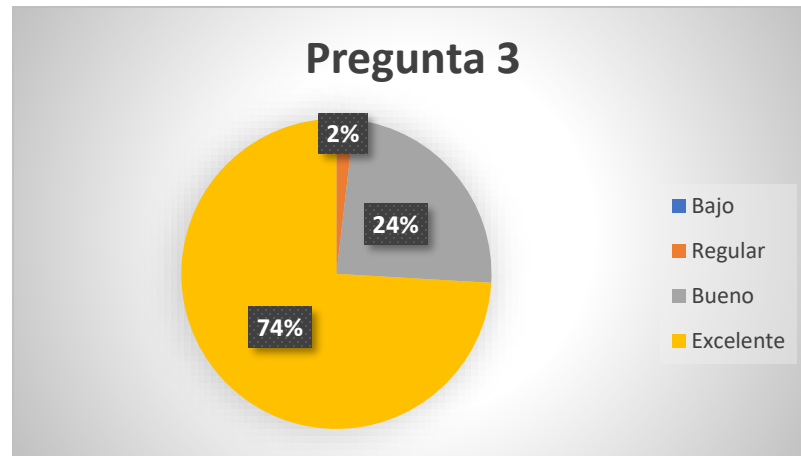
Según los resultados obtenidos, se puede interpretar que la mayoría de encuestados consideró que tiene excelentes instrucciones acerca de cómo pueden ganar el videojuego, esto sugiere que los niños no tienen problemas para comprender el objetivo del videojuego.

Pregunta 3: ¿Considera usted que las instrucciones del videojuego explicaron todo lo necesario para jugar?

En la **Figura 59**, correspondiente a la pregunta 3 se puede visualizar que la opción más votada fue “Excelente” con un 74% de los votos, mientras que la segunda opción votada fue con una calificación de “Bueno” obteniendo 24% de los votos, con una valoración de “Regular” se obtuvo el 2%.

Figura 59.

Resultados de la pregunta 3 del test de usabilidad



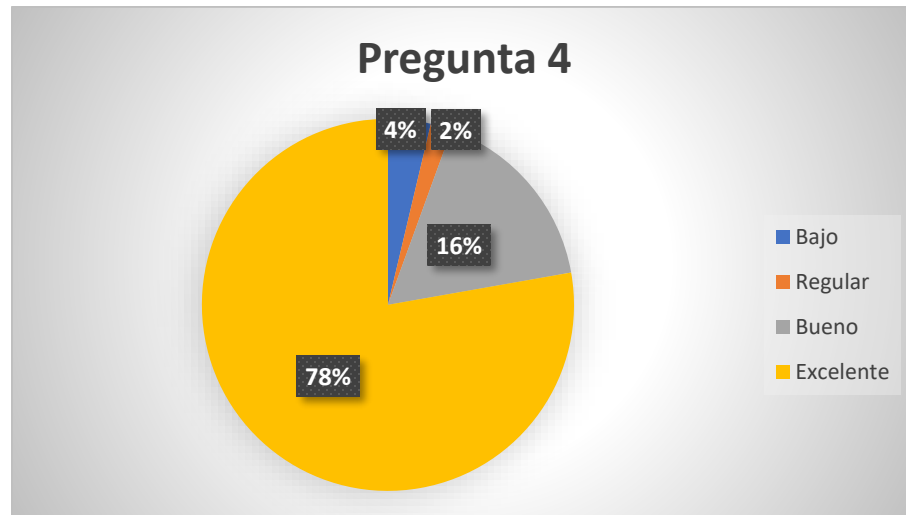
Estos resultados demostraron que la mayor parte de usuarios consideró que las instrucciones del videojuego proveen indicaciones fáciles de entender, revelando una alta satisfacción de claridad y completitud de las instrucciones.

Pregunta 4: ¿Cree usted que el videojuego incluye pistas o ayudas para cumplir el objetivo final?

En la **Figura 60**, correspondiente a la pregunta 4 visualiza que la opción más votada fue “Excelente” con un 78% de los votos, mientras que la segunda opción votada fue con una calificación de “Bueno” obteniendo 16% de los votos, con una valoración de “Regular” se obtuvo el 2% y finalmente un 4% calificó con valoración “Baja”.

Figura 60.

Resultados de la pregunta 4 del test de usabilidad.



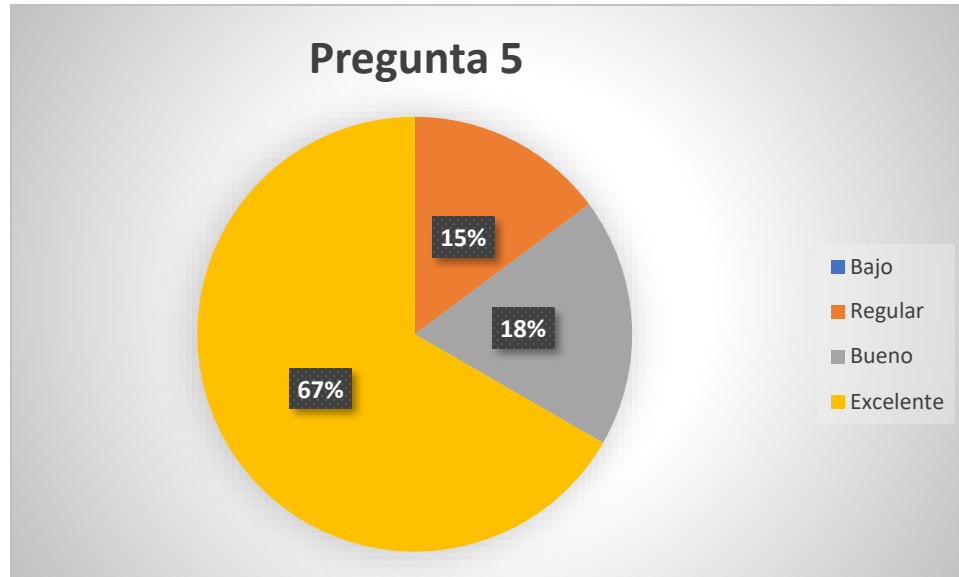
Se puede interpretar que la mayoría de usuarios encontró, que el videojuego 3D si incluye pistas o ayudas, esto demuestra que fue diseñado de manera efectiva para ayudar a los usuarios a cumplir el objetivo final.

Pregunta 5: ¿Piensa usted que el videojuego le comunica(avisa) cuando está haciendo algo incorrecto?

En la **Figura 61**, correspondiente a la pregunta 5 visualiza que la opción más votada fue “Excelente” con un 67% de los votos, mientras que la segunda opción votada fue con una calificación de “Bueno” obteniendo 18% de los votos, con una valoración de “Regular” se obtuvo el 15%.

Figura 61.

Resultados de la pregunta 5 del test de usabilidad



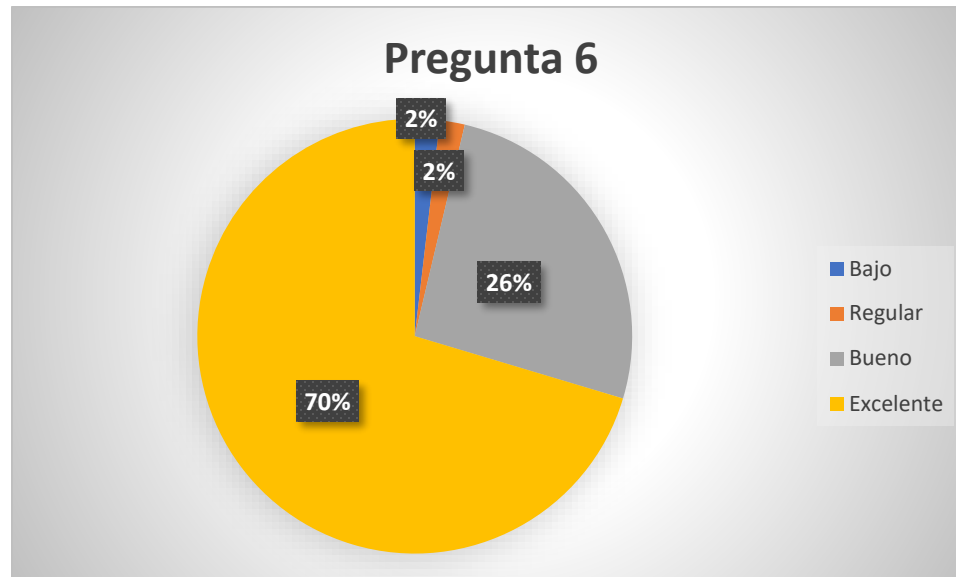
La mayor parte de encuestados considera que el videojuego ayuda a evitar acciones incorrectas mediante una guía de texto informativo indicando una buena retroalimentación, por otra parte, se presentó la minoría con un 15% indicando que se encontraron algunas deficiencias para alertar acciones inapropiadas, concluyendo que hay percepciones y niveles de satisfacción diferentes en las que el videojuego puede mejorar.

Pregunta 6: ¿Piensa usted que el videojuego da consejos para evitar errores y avanzar hasta el final?

En la **Figura 62**, correspondiente a la pregunta 6 visualiza que la opción más votada fue “Excelente” con un 70% de los votos, mientras que la segunda opción votada fue con una calificación de “Bueno” obteniendo 26% de los votos, con una valoración de “Regular” se obtuvo el 2% y finalmente un 2% calificó con valoración “Baja”.

Figura 62.

Resultados de la pregunta 6 del test de usabilidad.



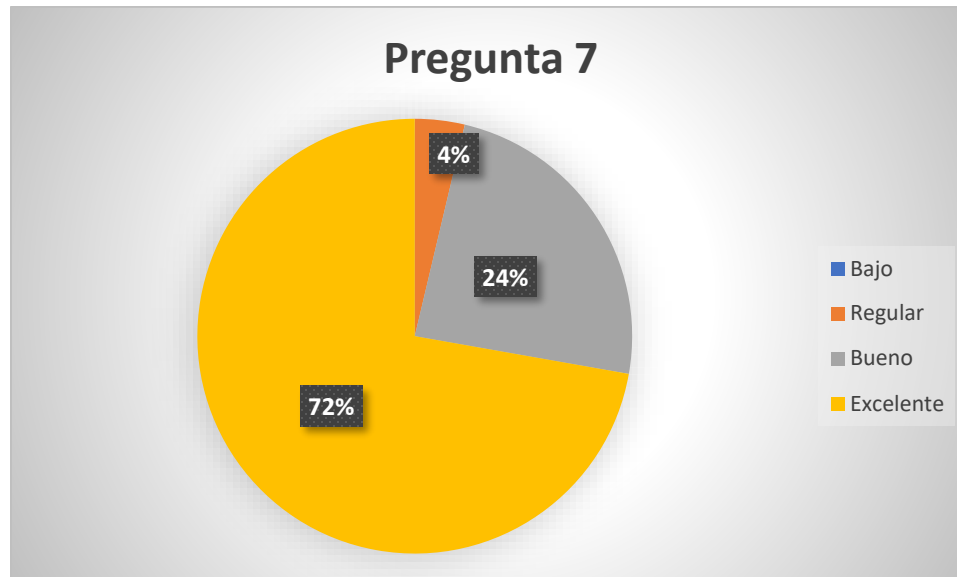
Los resultados obtenidos de esta pregunta reflejan que la mayor parte de jugadores encontraron consejos útiles y efectivos para progresar en el videojuego, existe un porcentaje menor que tienen opiniones menos favorables, pero en general, la percepción es positiva en cuanto a los consejos que facilitó el videojuego.

Pregunta 7: ¿Le resultó fácil cambiar brillo, volumen y tamaño de pantalla en las opciones del videojuego?

En la **Figura 63**, correspondiente a la pregunta 7 visualiza que la opción más votada fue “Excelente” con un 72% de los votos, mientras que la segunda opción votada fue con una calificación de “Bueno” obteniendo 24% de los votos, con una valoración de “Regular” se obtuvo el 4%.

Figura 63.

Resultados de la pregunta 7 del test de usabilidad.



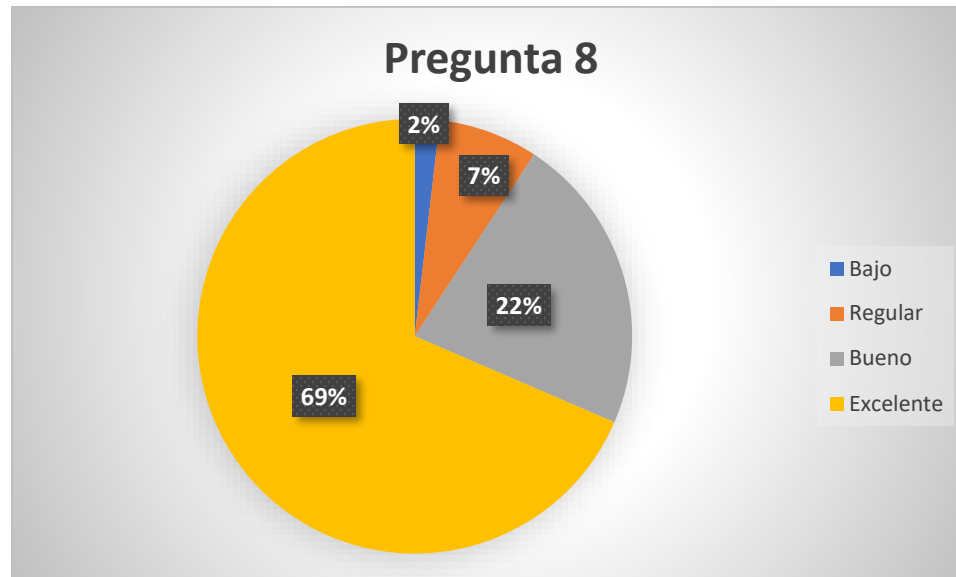
Se puede visualizar en el gráfico que la mayoría de usuarios no tuvieron dificultades para realizar ajustes en la pantalla del videojuego, esto indica que el diseño de interfaz es muy intuitivo y fácil de manipular.

Pregunta 8: ¿Cree usted que, al escribir su nombre en el videojuego, se le avisa si comete un error?

En la **Figura 64**, correspondiente a la pregunta 8 visualiza que la opción más votada fue “Excelente” con un 69% de los votos, mientras que la segunda opción votada fue con una calificación de “Bueno” obteniendo 22% de los votos, con una valoración de “Regular” se obtuvo el 7% y finalmente un 2% calificó con valoración “Baja”.

Figura 64.

Resultados de la pregunta 8 del test de usabilidad.



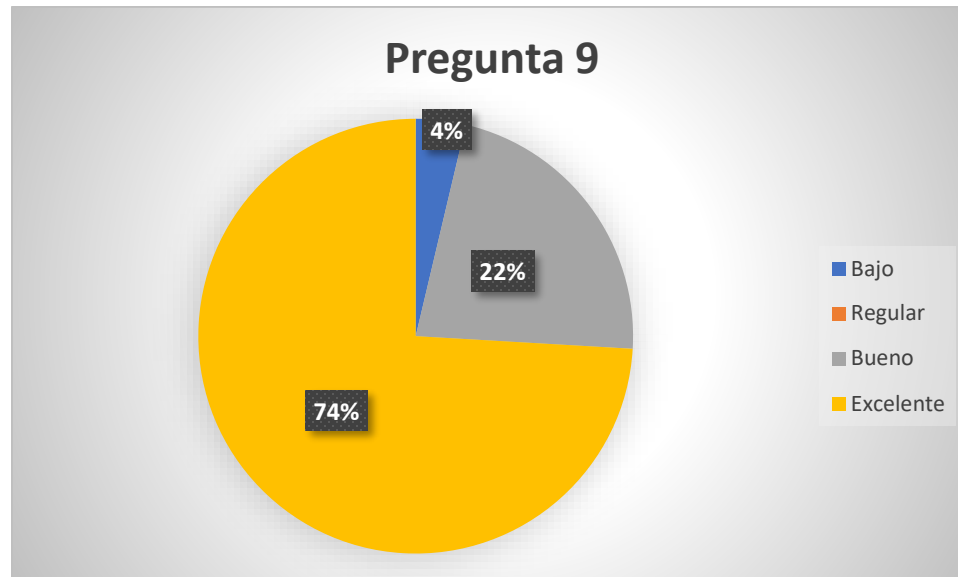
La mayor parte de encuestados encontró que el videojuego alerta eficazmente a los jugadores acerca de errores, por ejemplo, cuando se introduce un carácter mal o no se escribe el mínimo de caracteres.

Pregunta 9: ¿Piensa usted que el videojuego es fácil de entender para alguien que nunca lo ha jugado antes?

En la **Figura 65**, correspondiente a la pregunta 9 visualiza que la opción más votada fue “Excelente” con un 74% de los votos, mientras que la segunda opción votada fue con una calificación de “Bueno” obteniendo 22% de los votos, con una valoración de “Regular” se obtuvo el 4%.

Figura 65.

Resultados de la pregunta 9 del test de usabilidad.



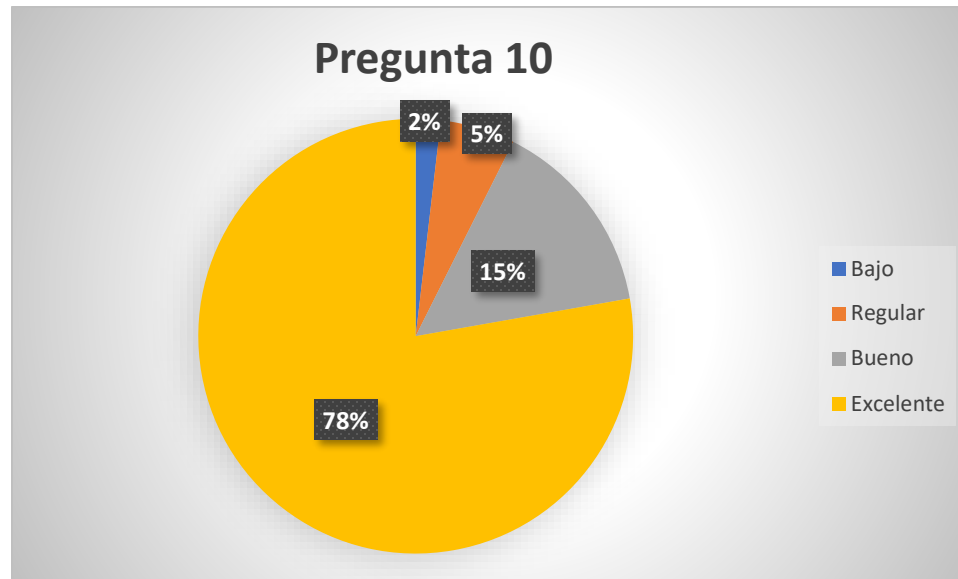
Los resultados de la encuesta reflejan que la mayoría de los participantes consideran que el videojuego es fácil de comprender para alguien sin experiencia previa, esto demuestra una buena accesibilidad y diseño intuitivo, por otro lado, también se debe considerar que hay un pequeño margen indicando que se puede mejorar claridad y comprensión de las mecánicas del videojuego, para aquellos que carecen de experiencia jugando.

Pregunta 10: ¿Le resultó fácil cambiar el color de letras que aparecen en el videojuego?

En la **Figura 66**, correspondiente a la pregunta 10 visualiza que la opción más votada fue “Excelente” con un 78% de los votos, mientras que la segunda opción votada fue con una calificación de “Bueno” obteniendo 15% de los votos, con una valoración de “Regular” se obtuvo el 5% y finalmente 2% restante valoraron con calificación “Baja”.

Figura 66.

Resultados de la pregunta 10 del test de usabilidad.



Concluyendo, la mayor parte de usuarios piensan que cambiar el color de letras es un proceso sencillo y sugieren que el videojuego ha logrado una experiencia de personalización satisfactoria.

4.5. Evaluación general del test.

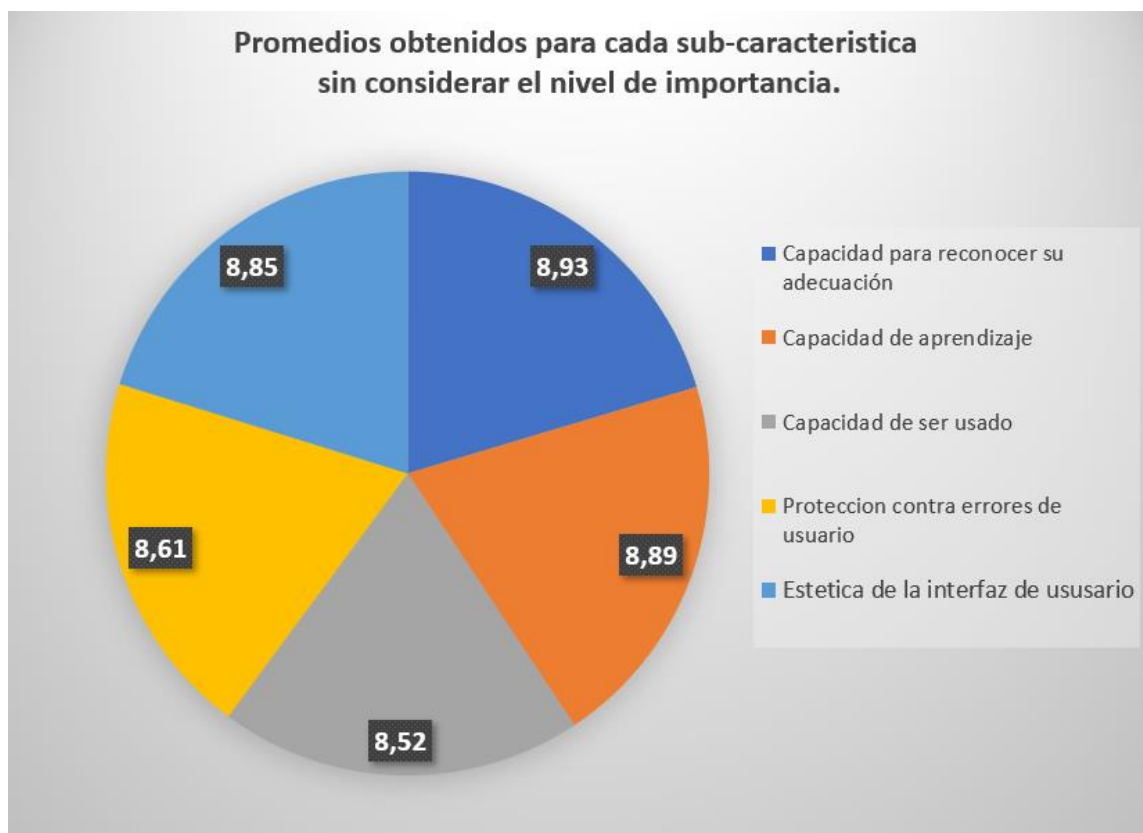
Con el propósito de evaluar la usabilidad del videojuego 3D, se obtuvieron puntuaciones individuales de las 54 personas encuestadas pertenecientes a la Unidad Educativa “César Antonio Mosquera” tomando en cuenta cada subcaracterística de acuerdo con la norma ISO/IEC 25010.

4.5.1. Promedios a nivel de subcaracterísticas

Basándose en la **Tabla 58**. A continuación en la Figura 67, se presentan los resultados obtenidos sin tomar en cuenta el nivel de importancia:

Figura 67.

Valoración de las subcaracterísticas de usabilidad.



Nota: No se consideró su nivel de importancia.

Con respecto a la **Figura 67**, se puede visualizarlas diferentes porcentajes obtenidos en cada subcaracterística y se clasifican de acuerdo con la **Tabla 57**, reflejando los siguientes puntajes:

Subcaracterísticas:

- **Capacidad para reconocer su adecuación**

Esta subcaracterística obtuvo un 8.93%, según el criterio de decisión de evaluación se posiciono al videojuego 3D como software de buena calidad, esto demuestra que el videojuego cumple con la capacidad de proporcionar una experiencia de juego completa, ofreciendo funcionalidades e instrucciones detalladas.

- **Capacidad de aprendizaje**

Se puede visualizar que esta subcaracterística obtuvo un 8.89%, según los criterios de evaluación indica que el videojuego 3D es software de buena calidad, Esta puntuación refleja que el videojuego proporciona una curva de aprendizaje gradual, lo que permite a los niños familiarizarse rápidamente con la mecánica del videojuego.

- **Capacidad para ser usado**

Los datos revelaron que esta subcaracterística registro un 8.52%, aplicando los criterios de evaluación el videojuego 3D se clasificó como software de buena calidad. Estos resultados demuestran que el videojuego presenta una interfaz intuitiva y brinda instrucciones claras para comprender y disfrutar la experiencia en el entorno 3D.

- **Protección contra errores de usuario**

Esta subcaracterística, obtuvo una puntuación del 8,61% de acuerdo con los criterios de decisión de la evaluación, sitúa al videojuego 3D como software de buena calidad. Los datos recopilados indican que el videojuego cuenta con una buena protección

contra errores de usuario, lo cual contribuye a una experiencia de juego fluida y reduce inconformidades por errores inadvertidos.

- **Estética de la interfaz de usuario**

Se puede visualizar que esta subcaracterística alcanzó un 8.85% y se posicionó al videojuego 3D como software de buena calidad según los criterios de evaluación. Esto demuestra que el videojuego incluye una buena estética de interfaz como el diseño de objetos, el uso de colores y tipografías, todo esto permite que la experiencia de juego sea satisfactoria y refuerza el interés por los videojuegos educativos.

4.5.2. Promedio final de la característica de usabilidad.

Para concluir la evaluación se tomó en cuenta los resultados anteriores, los cuales deben ser multiplicados según los criterios del nivel de importancia establecidos en la **Tabla 54**, luego se evalúa cada test individualmente de acuerdo con la **Tabla 61** y finalmente se promedia los resultados obtenidos de las 54 personas evaluadas.

Figura 68.

Valoración final de usabilidad del videojuego 3D.



Nota: Se tomó en cuenta la calificación de cada subcaracterística y se consideró su nivel de importancia.

La **Figura 68**, demuestra una alta puntuación en todas las subcaracterísticas de usabilidad tomando en cuenta los niveles de importancia, esto revela que el videojuego es capaz de ofrecer una experiencia de juego fluida, comprensible, gratificante e intuitiva. Además, proporciona una experiencia de uso de alta calidad.

Según los criterios de evaluación establecidos en la **Tabla 60**. Finalmente, el videojuego 3D educativo con un puntaje de 70,15% se clasificó como software de **Alta calidad** el cual demuestra que cumple con los estándares de usabilidad establecidos en la norma ISO/IEC 25010.

CONCLUSIONES.

Después de realizar un análisis bibliográfico, se determinó que la metodología "SUM" promueve la comunicación y colaboración constante entre miembros del equipo de trabajo, esto permite que se reduzca el tiempo de desarrollo, la iteración entre semanas es corta y se centra en la mitigación de riesgos de cada iteración, a comparación de la metodología OOHDM debido a su extensa iteración entre semanas, carece de mitigación de riesgos en cada iteración y se basa en modelos.

La metodología "SUM" demostró ser superior en términos de flexibilidad, productividad, costo de desarrollo y ciclo de iteración. Debido a sus ventajas en la alta detección de errores, tiempos de desarrollo más cortos y capacidad de ajustarse a las necesidades de los estudiantes debido a esto fue la mejor opción para el desarrollo del videojuego 3D.

El videojuego 3D demostró ser efectivo en la mejora de competencias en el área de matemáticas en estudiantes de educación básica, validando su viabilidad como herramienta educativa para mejorar el aprendizaje.

Después de evaluar las subcaracterísticas de usabilidad, se determinó que el videojuego educativo 3D obtuvo una valoración de "Alta calidad". Este resultado confirma que el videojuego cumple con los estándares de calidad establecidos en la norma ISO/IEC 25010 y se valora como software de calidad.

RECOMENDACIONES.

Para desarrollar software de manera planificada y eficiente, es recomendable utilizar una metodología de desarrollo. Es importante analizar puntos fuertes y débiles de cada metodología, así mismo se recomienda investigar opciones alternas.

Para la selección de una metodología se recomienda tomar en cuenta todos los factores, internos y externos al desarrollo, también se deben analizar las características específicas de cada proyecto y sus necesidades, debido a que pueden ser únicas, es aconsejable realizar una previa evaluación para determinar todos estos requisitos.

Antes de elaborar el videojuego 3D, se aconseja tener una reunión previa con los docentes de la unidad educativa, debido a que tienen una comprensión más profunda de las necesidades de los niños y esto puede ayudar a un mejor desarrollo del videojuego, adaptándolo específicamente a sus necesidades y así garantizar que el videojuego educativo sea efectivo para fortalecer el aprendizaje en los estudiantes.

Se recomienda que los desarrolladores de videojuegos educativos utilicen los criterios de la norma ISO/IEC 25010 para evaluar la usabilidad de sus productos de software y así garantizar que sean de buena calidad.

BIBLIOGRAFÍA

- Adžić, S., Al-Mansour, J., Naqvi, H., & Stambolić, S. (2021). The impact of video games on Students' educational outcomes. *Entertainment Computing*, 38, 100412.
<https://doi.org/10.1016/J.ENTCOM.2021.100412>
- Al-Saqqah, S., Sawalha, S., & Abdelnabi, H. (2020). Agile Software Development: Methodologies and Trends. *International Journal of Interactive Mobile Technologies (IJIM)*, 14(11), 246–270. <https://doi.org/10.3991/IJIM.V14I11.13269>
- Arenas, C. K. M. (2018). *Desarrollo de un Serious Gaming para estudiantes de Primer Año de Primaria aplicando la Metodología SUM*. UNIVERSIDAD PERUANA UNIÓN.
- Bica, B., Silva, D. A., & Gouvea Da, C. A. (2020). Learning process of agile scrum methodology with lego blocks in interactive academic games: Viewpoint of students. *Revista Iberoamericana de Tecnologías Del Aprendizaje*, 15(2), 95–104.
<https://doi.org/10.1109/RITA.2020.2987721>
- Cantero Guido Brun, R. R. C. (2018). *VIDEOJUEGOS COMO RECURSO EDUCATIVO DIGITAL 1*.
- Carrizo, D., Moller, C., Carrizo, D., & Moller, C. (2018). Estructuras metodológicas de revisiones sistemáticas de literatura en Ingeniería de Software: un estudio de mapeo sistemático. *Ingeniare. Revista Chilena de Ingeniería*, 26, 45–54.
<https://doi.org/10.4067/S0718-33052018000500045>
- Cedeño, C. K. I. (2020). *ANÁLISIS DE METODOLOGÍAS DE DISEÑO DE INTERFAZ HUMANO-COMPUTADOR EN LA PÁGINA DE APRENDIZAJE EN LÍNEA DE LA PUCE-E*. Pontífica Universidad Católica del Ecuador .

- Cevallos, H., Montero, K. M., & Cuesta, J. D. (2018). Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software. *ResearchGate*.
- Chiles Bolaños, M. G. (2021). *Entrevista a docente de 5to Grado de la Unidad Educativa “César Antonio Mosquera” [video]*.
- Delgado, O., & Díaz, L. M. A. (2021). Software Development Models. *Revista Cubana de Ciencias Informáticas, 15*.
- Díaz Pulido, J. A. (2017). *Diseño de un e-portafolio basado en la metodología OOHDM para gestionar la información de proyectos en informática*. UNIVERSIDAD SAN PEDRO.
- Fidalgo Blanco, Á., Luisa, M., Lacleta, S.-E., & García-Peñalvo, F. J. (2019, October 9). *Indicators of students’ participation in an active methodology*.
<https://doi.org/10.26754/CINAIC.2019.0121>
- Gabriel, L., & Ibarra, A. B. (2018). ANÁLISIS DOCUMENTAL DE LAS METODOLOGÍAS DE ENSEÑANZA. *DOCUMENTARY ANALYSIS OF TEACHING METHODOLOGIES*.
- García-Peñalvo, F. J. (2021, May 17). *Repositorio de GRIAL: Cómo hacer una Systematic Literature Review (SLR)*. *Cómo Hacer Una Systematic Literature Review (SLR)*.
<https://repositorio.grial.eu/handle/grial/2253>
- Gross, J. A. (2018a). *PISA-D*.
- Gross, J. A. (2018b). *PISA-D*.
- Krasteva, I., & Ilieva, S. (2020). Adopting Agile Software Development Methodologies in Big Data Projects - A Systematic Literature Review of Experience Reports. *Proceedings - 2020 IEEE International Conference on Big Data, Big Data 2020, 2028–2033*.
<https://doi.org/10.1109/BIGDATA50022.2020.9378118>

- Kumar, R., Malche, T., & Maheshwary, P. (2019). Inside Agile Family Software Development Methodologies A Review View project IoT Experimental project for Smart City View project Inside Agile Family: Software Development Methodologies. *Article in INTERNATIONAL JOURNAL OF COMPUTER SCIENCES AND ENGINEERING*.
<https://doi.org/10.26438/ijcse/v7i6.650660>
- Layedra Larrea Patricia, N. I., Alejandro Salazar Cazco III, S., Vinicio Ramos Valencia, M. I., & Alexander Baldeón Hermida, B. I. (2022). Análisis de los lenguajes de programación más utilizados en el desarrollo de aplicaciones web y móviles. *Dominio de Las Ciencias, ISSN-e 2477-8818, Vol. 8, Nº. 3, 2022 (Ejemplar Dedicado a: Julio-Septiembre 2022), 8(3), 132*.
<https://doi.org/10.23857/dc.v8i3>
- Markelov, A. (2016). Un Método Ingenieril de Identificación en Automatización | Tecnología Hoy. *Revista Tecnología Hoy, 2(1)*.
- Merzouk, S., Elhadi, S., Cherkaoui, A., Marzak, A., & Sael, N. (2018a). Agile Software Development: Comparative Study. *SSRN Electronic Journal*.
<https://doi.org/10.2139/SSRN.3186323>
- Merzouk, S., Elhadi, S., Cherkaoui, A., Marzak, A., & Sael, N. (2018b). Agile Software Development: Comparative Study. *SSRN Electronic Journal*.
<https://doi.org/10.2139/SSRN.3186323>
- Mesa, M. E. (2017). *Benchmarking metodologías híbridas para el desarrollo de software prototipo sistema didáctico*. Universidad Técnica del Norte.
- Momi, D., Smeralda, C., Sprugnoli, G., Ferrone, S., Rossi, S., Rossi, A., Di Lorenzo, G., & Santarnecchi, E. (2018). Acute and long-lasting cortical thickness changes following

- intensive first-person action videogame practice. *Behavioural Brain Research*, 353, 62–73.
<https://doi.org/10.1016/J.BBR.2018.06.013>
- Morandini, M., Coleti, T. A., Oliveira, E., & Corrêa, P. L. P. (2021). Considerations about the efficiency and sufficiency of the utilization of the Scrum methodology: A survey for analyzing results for development teams. *Computer Science Review*, 39, 100314.
<https://doi.org/10.1016/J.COSREV.2020.100314>
- Páez, A. H., Falcón, J. A. D., & Cruz, A. A. P. (2018a). *Software architecture for the development of videogames on the game engine Unity 3D* (1st ed., Vol. 14, pp. 56–57).
- Páez, A. H., Falcón, J. A. D., & Cruz, A. A. P. (2018b). *Software architecture for the development of videogames on the game engine Unity 3D* (1st ed., Vol. 14, pp. 56–57).
- Quelal, R. E., Villavicencio, M., & Mendoza, L. E. (2018). A survey of agile software development methodologies in Ecuador. *Iberian Conference on Information Systems and Technologies, CISTI, 2018-June*, 1–6. <https://doi.org/10.23919/CISTI.2018.8399186>
- Revilla Diego Miguel, Agustí María Sánchez, & Bengoechea Jesús María Moro. (2018, June). *DESIGN AND ASSESSMENT OF AN OPEN REPOSITORY OF EDUCATIONAL RESOURCES FOR THE TEACHING OF RECENT HISTORY*. *Revista Electrónica de Tecnología Educativa*. <https://www.edutec.es/revista/index.php/edutec-e/article/view/981/pdf>
- Saborío-Taylor, S., & Saborío-Taylor, S. (2019). Propuesta curricular desde un enfoque bimodal y un multimedia informativo para el curso Recursos Didácticos para la Enseñanza del Inglés. *Revista Electrónica Educare*, 23(3), 221–239. <https://doi.org/10.15359/REE.23-3.11>
- Saeed, S., Jhanjhi, N., Naqvi, M., & Humayun, M. (2019a, September). Analysis of Software Development Methodologies. *International Journal of Computing and Digital Systems*.

- Saeed, S., Jhanjhi, N. Z., Naqvi, M., & Humayun, M. (2019b). Analysis of software development methodologies. *International Journal of Computing and Digital Systems*, 8(5), 445–460. <https://doi.org/10.12785/IJCDS/080502>
- Sánchez, M. A., Rocha, A. G., Wilfor, A., Illanes, I., & Quiroz Perez, J. A. (2018). *Implementacion de la metodologia SUM modificada para el desarrollo de videojuegos orientados al aprendizaje en Bolivia.*
- Sayas, L. G., & Gonzales, D. P. (2017). *Influencia de los juegos didácticos en el aprendizaje del área de matemática en los alumnos de la Institución Educativa, Huaycán.* UNIVERSIDAD NACIONAL DE EDUCACIÓN.
- Sinchi Nacipucha, E. R. (2018). Acceso y deserción en las universidades. Alternativas de financiamiento. *Acceso y Deserción En Las Universidades.*, 13, 276.
- Soto-Ardila, L. M., Niño, L. M., Caballero, A., & Luengo, R. (2019). Estudio de las opiniones de los futuros maestros sobre el uso de los videojuegos como recurso didáctico a través de un análisis cualitativo. *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*, 33, 48–63. <https://doi.org/10.17013/RISTI.33.48-63>
- Vega, S. M. (2018, December). Educación para la paz y razón inclusiva. El pensamiento crítico en la filosofía para niños. *Peace Education and Inclusive Reasoning. Critical Thinking in Philosophy for Children.*
- Villadiego, C. (2018). *Explicación del proceso ingenieril.* Epartamento de Didácticas Específicas de La Universidad de Burgos. Facultad de Educación.
- Vite, H., Bryan Molina, M., Cevallos, & Cuesta, J. D. (2018, June). Agile methodologies against traditional methods in the software development process. *Espiraes Revista Multidisciplinaria de Investigación.*

Yakovyna, V., Seniv, M., & Symets, I. (2020). The Relation between Software Development Methodologies and Factors Affecting Software Reliability. *International Scientific and Technical Conference on Computer Sciences and Information Technologies, 1*, 377–381. <https://doi.org/10.1109/CSIT49958.2020.9321937>

ANEXOS

Anexo A. Test de usuario para medir la usabilidad.

Test de usuario dirigido a niños de educación básica de la Unidad Educativa “César Antonio Mosquera” para medir la usabilidad en productos de software académicos.

Nombre:

Edad:



Por favor marque una sola vez en los cuadros de cada pregunta según su opinión acerca del videojuego.

Bajo	Regular	Bueno	Excelente
			
—	—	—	X





Tabla de ejemplo.

Preguntas:



1. ¿Cree usted que el videojuego explica de forma clara los objetivos que deben ser cumplidos?

			
—	—	—	—

2. ¿Piensa usted que el videojuego tiene instrucciones de lo que debe hacer para ganar?

			
—	—	—	—

3. ¿Considera usted que las instrucciones del videojuego explicaron todo lo necesario para jugar?

			
—	—	—	—

4. ¿Cree usted que el videojuego incluye pistas o ayudas para cumplir el objetivo final?

			
—	—	—	—

5. ¿Piensa usted que el videojuego le comunica(avisa) cuando está haciendo algo incorrecto?



6. ¿Piensa usted que el videojuego da consejos para evitar errores y avanzar hasta el final?



7. ¿Le resultó fácil cambiar brillo, volumen y tamaño de pantalla en las opciones del videojuego?



8. ¿Cree usted que, al escribir su nombre en el videojuego, se le avisa si comete un error?



9. ¿Piensa usted que el videojuego es fácil de entender para alguien que nunca lo ha jugado antes?



10. ¿Le resultó fácil cambiar el color de letras que aparecen en el videojuego?



Gracias por su respuesta.

Anexo B. Evaluación de conocimientos para verificar el rendimiento de los estudiantes.

Evaluación para niños de 5to grado de Educación Básica de la
Unidad Educativa “César Antonio Mosquera”.

Nombre:

Edad:

1. ¿Si repartimos 50 panes a 25 personas, de cuantos panes le toca a cada uno?
 - a) 3
 - b) 6
 - c) 5
 - d) 2
2. ¿Si 7×8 es 56, cuanto es 8×7 ?
 - a) 55
 - b) 66
 - c) 65
 - d) 56
3. ¿Cuál es el resultado de la siguiente operación: 20×20 ?
 - a) 400
 - b) 40
 - c) 200
 - d) 240
4. Si tenemos 350 gramos de arroz y le agregamos 190 gramos, ¿cuánto pesará en total, en gramos?
 - a) 525
 - b) 510
 - c) 540
 - d) 530
5. Si un tren lleva 20 pasajeros, ¿cuántos pasajeros llevarán 311 trenes?
 - a) 6000
 - b) 6200
 - c) 6220
 - d) 3000

6. ¿Cuánto es el doble de 50?
- a) 26
 - b) 100
 - c) 55
 - d) 50
7. Un saco de limones pesa 89 kilogramos. ¿Cuántos kilogramos pesarán 36 sacos iguales?
- a) 3204 kg.
 - b) 3104 kg.
 - c) 3214 kg.
 - d) 3010 kg.
8. Ricardo compró 5 kilos de peras a \$24 el kilo y 4 kilos de cebollas a \$9 el kilo. ¿Cuánto gastó?
- a) 150
 - b) 157
 - c) 160
 - d) 156
9. ¿Cuál es el número mayor?
- a) 31,04
 - b) 900,02
 - c) 104,52
 - d) 480,83
10. Calcule el número resultante: $3+(5 \times 2) = ?$
- a) 10
 - b) 12
 - c) 13
 - d) 16

Anexo C. Primera evaluación para obtener el promedio inicial.



Anexo D. Segunda evaluación para obtener el promedio final luego de aplicar el videojuego.



Anexo E. Uso del videojuego por parte de los niños.



Anexo F. Participación de estudiantes.

