

REPÚBLICA DEL ECUADOR



**INSTITUTO SUPERIOR
TECNOLÓGICO TENA**
Tecnología, Innovación y Desarrollo

**CARRERA DE TECNOLOGÍA SUPERIOR EN
DESARROLLO DE SOFTWARE**

**TEMA:
DESARROLLO DE UN VIDEOJUEGO DE TERROR UTILIZANDO
UNITY Y C#.**

Trabajo de Integración Curricular, presentado como requisito parcial para optar por el título de Tecnólogo Superior de en Desarrollo de Software.

AUTOR: KEVIN JOEL PAUCA Y URVINA,
MARCO LEONARDO QUISPE DIAZ
TUTOR: ING. LIBINTON LARA

Tena - Ecuador

2023-IIS

APROBACIÓN DEL TUTOR

ING. LIBINTON LARA

DOCENTE DEL INSTITUTO SUPERIOR TECNOLÓGICO TENA.

CERTIFICA:

En calidad de Tutor del Proyecto Integrador denominado: **DESARROLLO DE UN VIDEOJUEGO DE TERROR UTILIZANDO UNITY Y C#**, de autoría de los señores: **KEVIN JOEL PAUCAY URVINA** con CC. 172602995-0, y, **MARCO LEONARDO QUISPE DIAZ**, con CC. 180549112-1, estudiantes de la Carrera de Tecnología Superior el Desarrollo de Software del Instituto Superior Tecnológico Tena, CERTIFICO que se ha realizado la revisión prolija del Trabajo antes citado, cumple con los requisitos de fondo y de forma que exigen los respectivos reglamentos e instituciones.

Tena, 30 de enero de 2024



firmado electrónicamente por:
**LIBINTON DUBERLI
LARA RIVERA**

Ing. Libinton Lara

TUTOR

CERTIFICACIÓN DEL TRIBUNAL CALIFICADOR

Tena, 30 de enero de 2024

Los Miembros del Tribunal de Grado abajo firmantes, certificamos que el Trabajo de Titulación denominado: **DESARROLLO DE UN VIDEO JUEGO DE TERROR UTILIZANDO UNITY Y C#**, presentado por KEVIN JOEL PAUCAY URVINA y MARCO LEONARDO QUISPE DIAZ, estudiantes de la Carrera de Tecnología Superior en Desarrollo de Software del Instituto Superior Tecnológico Tena, ha sido corregida y revisada; por lo que autorizamos su presentación.

Atentamente;



Firmado electrónicamente por:
ITALO MARCELO LARA
PILCO

Ing. Ítalo Marcelo Lara Pilco
PRESIDENTE DEL TRIBUNAL

Lcda. Inés Beatriz Gómez Rivas
MIEMBRO DEL TRIBUNAL

Ing. Marco Patricio Guanipatín Ramírez
MIEMBRO DEL TRIBUNAL

AUTORÍA

Nosotros, KEVIN PAUCAY, con CC: 1726029950, y MARCO QUISPE, con CC: 1805491121, declaramos ser autores del presente Trabajo de Titulación denominado: DESARROLLO DE UN VIDEOJUEGO DE TERROR UTILIZANDO UNITY Y C# y absolvemos expresamente al Instituto Superior Tecnológico Tena, y a sus representantes jurídicos de posibles reclamos o acciones legales por el contenido de la misma.

Adicionalmente aceptamos y autorizamos al Instituto Superior Tecnológico Tena, la publicación de mi trabajo de Titulación en el repositorio institucional- biblioteca Virtual.

AUTORES:



KEVIN JOEL PAUCAY URVINA

CÉDULA: 172602995-0



MARCO LEONARDO QUISPE DIAZ

CÉDULA: 180549112-1

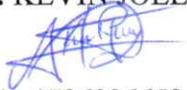
FECHA: Tena, 30 de enero de 2024

CARTA DE AUTORIZACIÓN POR PARTE DE LOS AUTORES

Nosotros, KEVIN JOEL PAUCAY URVINA y MARCO LEONARDO QUISPE DIAZ, declaramos ser autores del Trabajo de Titulación titulado: DESARROLLO DE UN VIDEO JUEGO DE TERROR UTILIZANDO UNITY Y C#, como requisito para la obtención del Título de: TECNÓLOGO SUPERIOR EN DESARROLLO DE SOFTWARE: autorizamos al Sistema Bibliotecario del Instituto Superior Tecnológico Tena, para que con fines académicos, muestre al mundo la producción intelectual del Instituto, a través de la visualización de su contenido que constará en el Repositorio Digital Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior, con las cuales tenga convenio el Instituto. El Instituto Superior Tecnológico Tena, no se responsabiliza por el plagio o copia del presente trabajo que realice un tercero. Para constancia de esta autorización, en la ciudad de Tena, 30 de enero de 2024, firma el autor.

AUTOR: KEVIN JOEL PAUCAY URVINA

FIRMA: 

CÉDULA: 1726026650

DIRECCIÓN: Calle Gabriel Espinoza y Quijos

CORREO ELECTRÓNICO: kevin.paucay@est.itstena.edu.ec

TELÉFONO: CELULAR: 0984267751

AUTOR: MARCO LEONARDO QUISPE DIAZ

FIRMA: 

CÉDULA: 1805491121

DIRECCIÓN: Av. 15 de Noviembre y Pano

CORREO ELECTRÓNICO: marco.quispe@est.itstena.edu.ec

TELÉFONO: CELULAR: 0986546550

DATOS COMPLEMENTARIOS

TUTOR: Ing. Libinton Lara

TRIBUNAL DEL GRADO:

Ing. Ítalo Lara. (Presidente).

Ing. Inés Gómez. (Miembro).

Ing. Patricio Guanipatín. (Miembro).

DEDICATORIA

Quiero decir que el presente trabajo está dedicado con todo mi amor y gratitud a las personas que han sido mi mayor soporte a lo largo de esta etapa de formación profesional y que hoy comparten conmigo la alegría de ver cristalizados años de sacrificio y esfuerzo constante.

En primer lugar, a mi amada madre Jenny Urvina, mujer ejemplar cuya valentía, tenacidad y entrega desinteresada han sido la brújula que ha guiado mi vida. Gracias, mamá por creer en mí aún más que yo mismo, por cada palabra de aliento en los momentos difíciles, por tus incansables desvelos y tu disposición permanente a acompañarme en este recorrido. Hoy cosechas lo que con tanto amor sembraste en mí.

A mis queridos abuelos, Gloria Hidalgo y Sergio Urvina, por ser fuente inagotable de sabiduría y apoyo moral a lo largo de toda mi existencia. Gracias por motivarme siempre con sus palabras y actos cariñosos a esforzarme por cumplir mis anhelos. Hoy, celebran conmigo el fruto de las semillas que ayudaron a plantar.

A mi tío Darío Urvina y mi tía Rubí Urvina, cómplices de muchos momentos significativos, por extenderme la mano cuando más requerí ayuda y alentarme a sortear los obstáculos.

A mis 3 amigos inseparables (Marco Quispe, Francisco Huatatoca y Bryan Mamallacta). Sus risas, apoyo incondicional y compañerismo han enriquecido cada capítulo de este viaje académico.

ATENTAMENTE: KEVIN JOEL PAUCAY URVINA

Dedico este trabajo principalmente a Dios, quien ha sido mi inspiración y fuente de fortaleza en la búsqueda de uno de mis sueños más anhelados.

A mis apreciados Padres José Quispe y Luz María Diaz, cuyo amor incondicional, esfuerzo y sacrificio han allanado el camino que me ha llevado hasta aquí. Ser su hijo es mi mayor orgullo y honor, y este logro es un reflejo de su incansable apoyo.

A mi querida hermana, compañera inseparable en esta travesía llamada vida. Su presencia ha sido mi consuelo y fortaleza, compartiendo risas y lágrimas en cada paso del camino.

A mis respetados profesores, cuyas enseñanzas han sido faro y guía en mi formación académica. Su dedicación y sabiduría han dejado una huella indeleble en mi camino hacia el conocimiento.

A mi estimado jefe, a quien le agradezco por brindarme la oportunidad de trabajo y confiar en mis habilidades.

A mis amigos y demás compañeros de clases, quienes han compartido esta travesía conmigo. Sus risas, apoyo incondicional y compañerismo han enriquecido cada capítulo de este viaje académico.

A todas las personas que, de una forma u otra, han contribuido a la realización exitosa de este trabajo. Mi reconocimiento especial a aquellos que abrieron las puertas de su conocimiento y brindaron apoyo incondicional.

Marco Leonardo Quispe Diaz.

AGRADECIMIENTO

En todo arduo camino hacia la concreción de un logro existe una fuerza motriz invisible que nos sostiene, alienta e inspira cada vez que nos invade el desaliento. Son aquellos seres que, desde la sombra, iluminan con su amor desinteresado el trayecto para que no decaiga el ánimo.

Por ello, deseo expresar mi más profundo agradecimiento a las personas que representaron ese cimiento afectivo a lo largo de este desafiante proceso para alcanzar la meta propuesta.

A mi madre Jenny, mi mayor ejemplo de vida, por creer en mí mucho antes que yo mismo y demostrármelo con cada sacrificio emprendido por verme triunfar. Gracias por los incansables desvelos y por nunca escatimar esfuerzo alguno si de mi bienestar se trataba. Este triunfo es tan tuyo como mío.

A mis amados abuelos, Gloria y Sergio, por su sabiduría y fe ciega en mi capacidad para asumir retos que a otros intimidaban. Gracias por tanto amor derramado y por el permanente estímulo a no desfallecer cuando el futuro se vislumbraba incierto. Ustedes son los pilares morales de lo que ahora cosecho.

A mis 3 amigos inseparables Marco, Francisco y Bryan, con los cuales he compartido mil historias, altibajos, risas, lágrimas y demás cosas que solo ellos fueron capaces de comprenderme incluso en momentos en los cuales ni yo me comprendía.

A todos ustedes, ¡infinitas gracias de corazón! Este logro será siempre compartido con quienes nunca escatimaron fuerzas para verme crecer. Su aliento y fe se traducen hoy en esta anhelada victoria que llena de orgullo mi espíritu y que pertenece tanto a ustedes como a mí.

Con todo el cariño del mundo, **Kevin Joel Paucay Urvina**

El presente trabajo, deseo expresar mi gratitud a Dios por ser mi guía y por acompañarme a lo largo de mi vida. Su paciencia y sabiduría han sido fundamentales para alcanzar con éxito las metas que me he propuesto. También quiero agradecer a mis padres por ser el pilar fundamental de mi vida y por su incondicional apoyo, incluso en medio de las adversidades e inconvenientes que hemos enfrentado.

Marco Leonardo Quispe Diaz.

INDICE GENERAL

APROBACIÓN DEL TUTOR.....	i
CERTIFICACIÓN DEL TRIBUNAL CALIFICADOR.....	ii
AUTORÍA.....	iii
CARTA DE AUTORIZACIÓN POR PARTE DE LOS AUTORES	iv
DEDICATORIA.....	v
AGRADECIMIENTO.....	vii
INDICE GENERAL.....	ix
INDICE DE TABLAS	xii
INDICE DE FIGURAS.....	xiii
A TEMA.....	15
RESUMEN.....	16
ABSTRACT	17
B FUNDAMENTACIÓN DEL TEMA	18
2.1. Necesidad	18
2.2. Actualidad.....	18
2.3. Importancia.....	19
2.4. Presentación del problema profesional a responder	20
2.5. Delimitación	21
2.5.1. Delimitación Espacial.....	21
2.5.2. Delimitación Temporal.....	21
2.5.3. Delimitación Técnica	21
2.5.4. Unidades de Observación.....	21
2.6. Beneficiarios	21
2.6.1. Directos	21
2.6.2. Indirectos	21
C OBJETIVOS.....	22

3.1. Objetivo General	22
3.2. Objetivos Específicos	22
D ASIGNATURAS INTEGRADORAS	23
E FUNDAMENTACIÓN TEÓRICA	24
5.1. Temas	24
5.1.1. Los videojuegos.....	24
5.1.2. Desarrollo de Videojuegos	26
5.1.3. Unity.....	26
5.1.4. Programación en C#.....	26
5.1.5. Tipos o géneros de videojuegos	27
5.1.6. Género de terror en videojuegos.....	28
5.2. Marco Legal	30
5.2.1. COIP artículos referentes	30
5.3. Marco Conceptual	36
5.4. Software.....	36
5.5. Metodología ágil.....	36
5.6. Metodología de desarrollo de software.	36
5.7. Videojuego	37
5.8. Motor de videojuego.....	37
5.9. UI Toolkit	37
5.10. Género de terror en los videojuegos.....	37
F METODOLOGÍA.....	38
6.1. Materiales	38
6.1.1. Equipos.....	38
6.1.2. Herramientas.....	38
6.2. Ubicación del Área de estudio.....	39
6.3. Tipo de investigación / estudio.....	39
6.3.1. Investigación bibliográfica.	39
6.4. Metodología de desarrollo de software	40

6.5. Metodología para cada objetivo.	41
6.5.1. El equipo del trabajo	41
6.6. Eventos en Scrum	41
6.6.1. Planificación del sprint (Sprint Planning)	42
6.6.2. Sprint	43
6.6.3. Scrums diarios (Daily Scrum)	44
6.6.4. Revisión del sprint (Sprint review)	45
6.6.5. Retrospectiva del Sprint (Sprint retrospective)	47
G RESULTADOS	48
7.1. Resultados de los requerimientos para el desarrollo del videojuego .	48
7.1.1. El Equipo de trabajo	48
7.1.2. Resultados por objetivo	48
H CONCLUSIONES	56
I RECOMENDACIONES	57
J BIBLIOGRAFIA	58
K ANEXOS	63

INDICE DE TABLAS

Tabla 1 <i>Asignaturas Integradoras</i>	23
Tabla 2 <i>Roles Recomendados para el Equipo de Trabajo</i>	41
Tabla 3 <i>Conformación del Equipo de Trabajo para desarrollar el Videjuego.</i>	48

INDICE DE FIGURAS

Figura 1 Metodología ágil SCRUM.....	40
Figura 2 Interfaz del menú principal.....	49
Figura 3 Interfaz de la pantalla ajustes y todos sus elementos	50
Figura 4 Interfaz de la Escena Inicial del Juego.....	50
Figura 5 Interfaz del Menú de Pausa donde también se puede Acceder al Menú de Ajustes	51
Figura 6 Construcción del Nivel del videojuego y sus Interfaces a través del Editor Unity.....	51
Figura 7 Prueba de Funcionalidad del videojuego dentro del Editor	52
Figura 8 Codificación del movimiento del jugador.....	52
Figura 9 Codificación de la Funcionalidad de Romper Objetos.....	53
Figura 10 Carpeta de Compilación Final con el Videojuego Terminado, listo para Transformar a un solo Aplicativo Instalable (.EXE).....	53
Figura 11 Carpeta con el Ejecutable Terminado (.EXE).....	54
Figura 12 Publicación del videojuego en una Tienda de Aplicaciones “Uptodown”	54
Figura 13 Vista del estatus general de la aplicación en la tienda de aplicaciones	55
Figura 14 Ve a la tienda de aplicaciones de Uptodown.....	66
Figura 15 Escoge el sistema operativo Windows	66
Figura 16 En el buscador escribe la palabra “Escapa!”	67
Figura 17 Una vez seleccionado la opción oprime donde dice “Consigue la última versión”	67
Figura 18 Oprime en el botón descargar	68

Figura 19 <i>Una vez descargado inicia el proceso de instalación.....</i>	68
Figura 20 <i>Acepta los términos y condiciones de uso del videojuego</i>	69
Figura 21 <i>Haz clic en siguiente (puedes cambiar la ruta de instalación).....</i>	69
Figura 22 <i>Puedes crear un acceso directo en el escritorio</i>	70
Figura 23 <i>Haz clic en Instalar para iniciar el proceso de instalación.....</i>	70
Figura 24 <i>Listo, puedes ejecutarlo desde tu escritorio</i>	71

A TEMA

**DESARROLLO DE UN VIDEOJUEGO DE TERROR UTILIZANDO
UNITY Y C#.**

RESUMEN

El informe presenta el desarrollo de un videojuego de terror utilizando Unity y C# como parte de un trabajo de integración curricular fue realizado mediante la metodología de desarrollo de software SCRUM. Destaca la importancia del diseño de niveles, iluminación, efectos audiovisuales y mecánicas para generar tensión en los jugadores. También aborda el marco legal relacionado con la propiedad intelectual. Este trabajo incluye secciones como agradecimientos, fundamentación del tema, objetivos, asignaturas integradoras, fundamentación teórica, metodología y resultados. Se resalta la importancia de la programación en C# y el género de terror en videojuegos, así como la metodología utilizada para el desarrollo del proyecto. En resumen, el trabajo detalla el proceso de desarrollo de un videojuego de terror utilizando Unity y C#, enfatizando la importancia del diseño y las implicaciones legales relacionadas con la propiedad intelectual.

Palabras clave: videojuego, terror, Unity, C#, desarrollo de software, diseño de niveles, iluminación, efectos audiovisuales, mecánicas, propiedad intelectual, programación, metodología, conclusiones, recomendaciones.

ABSTRACT

The report presents the development of a horror video game using Unity and C# as part of a curricular integration work, which was carried out using the SCRUM software development methodology. It highlights the importance of level design, lighting, audiovisual and mechanical effects to generate tension in the players. It also addresses the legal framework related to intellectual property. This work includes sections such as acknowledgments, foundation of the topic, objectives, integrative subjects, theoretical foundation, methodology and results. The importance of C# programming and the horror genre in video games is highlighted, as well as the methodology used for the development of the project. In summary, the work details the development process of a horror video game using Unity and C#, emphasizing the importance of design and the legal implications related to intellectual property.

Keywords: video game, horror, Unity, C#, software development, level design, lighting, audiovisual effects, mechanics, intellectual property, programming, methodology, conclusions, recommendations

Reviewed by:



BEd. Jefferson Andrés Intriago Burgos M.Ed.

ID. 1313123976

Teacher of English Center of IST Tena

B FUNDAMENTACIÓN DEL TEMA

2.1. Necesidad

En un contexto mundial, la implementación de los video juegos ha sido un factor preponderante dentro del desarrollo de Software, del desarrollo económico del país y de la empresa creadora del mismo, y a su vez, del entorno juvenil y más aún video juegos de carácter temerario, basándose éstos en la aplicación de habilidades y estrategias a seleccionar por parte de los jugadores.

La industria de los videojuegos tiene un gran potencial en Ecuador, aunque actualmente está poco desarrollada en comparación con otros países latinoamericanos. Según la ADVA, hay alrededor de 80 estudios de desarrollo de videojuegos en Ecuador, una cifra baja en comparación con países como Argentina, Brasil o México. A nivel mundial, esta industria mueve aproximadamente 180 mil millones de dólares, con un crecimiento constante en el número de jugadores.

Países como Argentina han logrado éxito en el mercado global de entretenimiento digital, demostrando el potencial de la región. Ecuador podría aprovechar su talento joven y creativo para convertirse en un centro de desarrollo de videojuegos en la región, impulsando la formación de profesionales en áreas como programación, arte y diseño. (Kingman, 2021)

2.2. Actualidad

En nuestro país, la industria de los videojuegos y software interactivo aún sigue en crecimiento, con alrededor de 80 estudios activos en el país (Muñoz González, 2019). Si bien no hay datos específicos sobre la facturación del sector gaming, la exportación de software en general llegó a generar USD \$67 millones en 2021 (Telegrafo, 2021). A pesar de este progreso, existe amplio espacio para el

desarrollo de más talento e inversión en la industria. El diseño de videojuegos innovadores, como uno de terror con fines educativos, no solo reduciría la escasez de entretenimiento interactivo local de calidad, sino que incentivaría la economía digital y la creación de empleos en esta área emergente.

De esta manera podemos concluir que, en nuestro país esta industria está en una etapa incipiente, con algunas debilidades en cuanto a desarrollo de productos y facturación. En cambio, a nivel mundial organismos como la (UNESCO, 2019) respaldan el potencial de los juegos de arcade para el aprendizaje y desarrollo de habilidades. Es decir, existe una oportunidad para impulsar este sector en el país. Este tipo de aplicativos fomentan el pensamiento crítico, la resolución de problemas y la colaboración.

2.3. Importancia

Varios estudios han encontrado que los juegos digitales pueden tener efectos positivos en la salud mental al reducir el estrés, la ansiedad y la depresión.

Durante la pandemia, cuando muchas personas se sentían aisladas y ansiosas, los juegos para computadora proporcionaron una vía de escape y conexión social.

Un estudio realizado en 2021 por investigadores de la Universidad de Oxford encontró que los jugadores habituales tenían un riesgo un 63% menor de sufrir de depresión y un 75% menor riesgo de ansiedad en comparación con las personas que no jugaban. Esto sugiere que los juegos interactivos fueron un factor protector para la salud mental durante la época de confinamiento.

Otra investigación realizada en España examinó las tasas de suicidio antes y después de la pandemia. Descubrieron que entre marzo y diciembre de 2020, el número de suicidios fue un 17% menor en comparación con el mismo período en

2019. Los investigadores creen que esto se debió en parte al papel del entretenimiento digital y los juegos de video para aliviar factores estresantes.

En general, aunque los casos de depresión y ansiedad aumentaron durante la epidemia global, las tasas de suicidio en realidad disminuyeron levemente en 2020 y los primeros estudios apuntan a que los juegos de computador fueron un factor mitigante importante. Proporcionaron una forma de que las personas se mantuvieran conectadas y manejaran mejor las emociones negativas.

Este respaldo internacional destaca la importancia de abordar el diseño de juegos de video no solo desde la perspectiva del entretenimiento, sino también como herramientas educativas que pueden fomentar habilidades esenciales en un mundo cada vez más digitalizado.

2.4. Presentación del problema profesional a responder

Precisamente, el videojuego de terror propuesto estará desarrollado específicamente para contribuir al crecimiento de las industrias del entretenimiento y tecnología local. En concreto, sería un juego de video offline de un solo jugador con vista en primera persona, creado con el motor Unity de la mano con el lenguaje de programación C# y gráficos 3D estilo Low Poly a 30 FPS.

El presente proyecto motivara al talento nacional en este sector emergente.

Campo: Tecnologías de la Información y Comunicación

Área: Desarrollo de Software

Aspecto: Desarrollo de un VIDEO JUEGO de Terror utilizando Unity y C#

Sector: Desarrollo de Software

Línea de investigación: Tecnologías de la información y comunicación.

2.5. Delimitación

2.5.1. Delimitación Espacial

El Trabajo de Integración Curricular se realizará en la empresa “Pear Videogames”, la misma que está ubicada en la vía Rafaela Segala 19-26 y Federico Monteros a 450 metros de la Gasolinera P&S “Ciudad de Tena”.

2.5.2. Delimitación Temporal

El proyecto se lo efectuará en el Periodo Académico 2023-IIS.

2.5.3. Delimitación Técnica

El presente proyecto tendrá los siguientes módulos o interfaces:

- Un menú principal
- Un menú de pausa
- Un menú para controlar los gráficos, sonidos y estilo de video
- Tendrá dos niveles
- Tendrá objetivos que el jugador deberá cumplir para avanzar

2.5.4. Unidades de Observación

Las unidades de observación que se contemplan para este trabajo están enfocadas directamente a la unidad de Programación de Pear VideoGames inc. Empresa que desea ingresar al mercado mediante este proyecto

2.6. Beneficiarios

2.6.1. Directos

Los beneficiarios directos del trabajo Integrador Curricular son:

Empresa Pear Videogames.

2.6.2. Indirectos

Los beneficiarios indirectos del proyecto de titulación son:

Todos los usuarios que descarguen la aplicación.

C OBJETIVOS

3.1. Objetivo General

Desarrollar un videojuego de terror utilizando Unity y C# para la empresa “Pear Videogames” ubicada en la ciudad de Tena, provincia de Napo en el año 2024

3.2. Objetivos Específicos

- Definir los procesos y alcance del proyecto
- Desarrollar la aplicación a través de Unity mediante C#
- Realizar pruebas con usuarios para verificar la funcionalidad del videojuego.

D ASIGNATURAS INTEGRADORAS

Tabla 1

Asignaturas Integradoras

ASIGNATURAS INTEGRADORAS	
Asignaturas	Resultados de Aprendizaje
Metodologías del Desarrollo de Software	Aplicar una metodología de desarrollo de software a lo largo del ciclo de vida de la aplicación que se está desarrollando.
Diseño Multimedia	Crea entornos fáciles de navegar por el usuario con los entornos de interfaces interactivos
Programación Visual	Desarrollar aplicaciones de software utilizando herramientas visuales, como entornos de desarrollo integrados (IDE) con capacidades de diseño gráfico

E FUNDAMENTACIÓN TEÓRICA

5.1. Temas

5.1.1. Los videojuegos

Concepto:

Los videojuegos son formas de entretenimiento interactivo que involucran la participación del jugador a través de interfaces electrónicas, como consolas de videojuegos, computadoras personales, dispositivos móviles u otros dispositivos especializados. Estos juegos suelen presentar diversos desafíos, objetivos o narrativas que los jugadores deben completar o experimentar, a menudo en un entorno virtual. (Ruíz, 2017)

Historia de los videojuegos

La historia de los videojuegos se remonta a la década de 1950, cuando comenzaron a desarrollarse las primeras computadoras digitales. A medida que la tecnología avanzaba, surgieron los primeros juegos electrónicos simples, ¡como "Spacewar!" en la década de 1960. Sin embargo, fue en la década de 1970 cuando los videojuegos comenzaron a popularizarse con la llegada de las consolas de videojuegos domésticas, como la Magnavox Odyssey y la Atari Pong. Desde entonces, la industria de los videojuegos ha experimentado un crecimiento explosivo, con avances significativos en gráficos, jugabilidad, narrativa y tecnología. La década de 1980 vio el surgimiento de juegos arcade y el nacimiento de icónicos personajes como Mario y Pac-Man. En los años 90, los videojuegos 3D se volvieron más prominentes con el lanzamiento de consolas como el PlayStation original y el Nintendo 64. La década de 2000 marcó la era de los juegos en línea y los MMO (Massively Multiplayer Online), así como el surgimiento de los juegos móviles con la popularidad de dispositivos como el iPhone.

Hoy en día, los videojuegos abarcan una amplia gama de géneros y plataformas, desde juegos móviles casuales hasta títulos de alta gama en consolas de última generación y computadoras de alto rendimiento. La industria de los videojuegos es una de las más grandes del mundo del entretenimiento, con una base de fans global y un constante flujo de innovación y creatividad. (Catena, 2023)

¿Dónde son más usados?

Los videojuegos son utilizados en todo el mundo, pero la popularidad y el uso pueden variar según la región y la cultura. Algunas de las regiones más destacadas en términos de uso de videojuegos incluyen:

1. **América del Norte:** Estados Unidos y Canadá son dos de los mayores mercados de videojuegos del mundo, con una amplia adopción tanto de consolas como de juegos para PC y móviles. (Roa M, 2022)
2. **Europa:** Países como el Reino Unido, Alemania y Francia tienen una fuerte presencia en la industria de los videojuegos, con una gran base de jugadores y desarrolladores. (Nast C, 2023)
3. **Asia:** La región de Asia-Pacífico es un líder en la industria de los videojuegos. Países como Japón, Corea del Sur y China son conocidos por su cultura de juego arraigada y por ser hogar de algunos de los principales desarrolladores y empresas de videojuegos del mundo. (Villalobos, 2023)
4. **América Latina:** Si bien el mercado de los videojuegos en América Latina es más pequeño en comparación con otras regiones, ha experimentado un crecimiento significativo en los últimos años, con países como Brasil, México y Argentina emergiendo como importantes mercados. (Smathers A, 2022)

5.1.2. Desarrollo de Videojuegos

El desarrollo de videojuegos es un proceso que integra aspectos creativos, artísticos y técnicos como diseño, animación, programación e inteligencia artificial (Goslin, 2022). Unity es uno de los motores de desarrollo más populares actualmente por sus capacidades multiplataforma, comunidad activa y abundantes recursos educativos (Hafzal Hussain, 2020)

La industria de desarrollo de videojuegos en Ecuador es pequeña pero creciente. Se estima que existen alrededor de 20 estudios activos en el país. Si bien aún no alcanza el nivel de otros países de la región, se observan avances a través de iniciativas gubernamentales y académicas para fomentar el sector (Gordón, 2015)

5.1.3. Unity

Unity es un motor de desarrollo de videojuegos multiplataforma creado por Unity Technologies. Permite el desarrollo de videojuegos 2D y 3D para diversos dispositivos. Cuenta con un editor de escenas drag-and-drop, soporte para scripts en C#, una comunidad amplia y abundantes recursos de aprendizaje (Unity, 2022)

5.1.4. Programación en C#

C# es un lenguaje de programación orientado a objetos desarrollado por Microsoft. Se utiliza ampliamente en el desarrollo de videojuegos con Unity ya que permite escribir scripts para controlar comportamientos y lógica dentro del videojuego. (Unity, 2022)

5.1.5. Tipos o géneros de videojuegos

Los videojuegos abarcan una amplia gama de géneros y estilos, cada uno ofreciendo una experiencia única para los jugadores. Aquí tienes una lista de algunos de los géneros más comunes:

- **Acción:** Este género se centra en la acción y la velocidad de reacción del jugador. Pueden incluir subgéneros como juegos de disparos en primera persona (FPS), juegos de disparos en tercera persona (TPS), juegos de plataformas y hack and slash.
- **Aventura:** Los juegos de aventura se centran en la exploración, la narrativa y la resolución de acertijos. Pueden incluir subgéneros como aventuras gráficas, juegos de mundo abierto y juegos de acción-aventura.
- **RPG (Rol-Playing Game):** En estos juegos, los jugadores asumen el papel de personajes ficticios y controlan sus acciones y decisiones en un mundo virtual. Pueden incluir subgéneros como RPG de acción, RPG tácticos y MMORPG (juegos de rol multijugador en línea masivos).
- **Estrategia:** Los juegos de estrategia requieren planificación y toma de decisiones estratégicas por parte del jugador para alcanzar objetivos específicos. Pueden incluir subgéneros como estrategia en tiempo real (RTS), estrategia por turnos y juegos de simulación.
- **Deportes:** Estos juegos simulan deportes reales o ficticios y permiten a los jugadores competir en eventos deportivos. Pueden incluir subgéneros como juegos de fútbol, baloncesto, carreras y juegos extremos.
- **Puzzle (Rompecabezas):** Los juegos de puzzle desafían la mente del jugador con acertijos y desafíos lógicos. Pueden incluir subgéneros como juegos de palabras, juegos de ingenio y juegos de objetos ocultos.

- **Terror:** Los juegos de terror están diseñados para asustar o perturbar al jugador, a menudo a través de elementos de suspense, terror y supervivencia. Pueden incluir subgéneros como survival horror y juegos de terror psicológico.
- **Plataforma:** En estos juegos, los jugadores controlan un personaje que debe saltar y correr a través de niveles llenos de obstáculos y enemigos. Pueden incluir subgéneros como plataformas 2D y plataformas 3D.
- **Sigilo:** Los juegos de sigilo requieren que los jugadores eviten el combate directo y se muevan de manera discreta para alcanzar objetivos sin ser detectados por enemigos. Pueden incluir subgéneros como sigilo táctico y sigilo de acción.
- **Mundo Abierto (Open World):** Estos juegos ofrecen a los jugadores un mundo virtual expansivo y no lineal para explorar a su propio ritmo. Pueden incluir subgéneros como sandbox (caja de arena), mundo abierto narrativo y mundo abierto de supervivencia.

Estos son solo algunos de los géneros más comunes en el mundo de los videojuegos, pero la industria continúa evolucionando y surgiendo nuevos géneros y fusiones entre ellos. (Gonzalez D, 2016)

5.1.6. Género de terror en videojuegos

Los videojuegos de terror intentan provocar emociones intensas como miedo, ansiedad o pánico en los jugadores a través de elementos como ambientaciones oscuras, sonidos perturbadores, persecuciones, sangre y situaciones. (Jih-Hsuan Tammy Lin, 2017).

Requieren de un cuidadoso diseño de niveles, iluminación, efectos audiovisuales y mecánicas de videojuego para generar tensión, incertidumbre y sobresaltos en los jugadores (Kieran Newell, 2021)

Ventajas:

- **Experiencia inmersiva:** Los videojuegos de terror pueden sumergir completamente al jugador en un ambiente tenso y aterrador, proporcionando una experiencia más inmersiva que otras formas de entretenimiento, como películas o libros.
- **Desarrollo de habilidades cognitivas:** Algunos juegos de terror requieren que los jugadores resuelvan acertijos, encuentren objetos ocultos o tomen decisiones rápidas bajo presión, lo que puede ayudar a desarrollar habilidades cognitivas como la resolución de problemas y la toma de decisiones.
- **Control de emociones:** Los videojuegos de terror pueden ayudar a los jugadores a enfrentar y controlar sus miedos al exponerlos a situaciones aterradoras en un entorno controlado y seguro.
- **Adrenalina y emoción:** La sensación de adrenalina y emoción que provoca jugar un videojuego de terror puede ser una experiencia emocionante y gratificante para muchos jugadores que buscan una experiencia intensa.

Desventajas

- **Impacto emocional:** Para algunas personas, los videojuegos de terror pueden ser demasiado intensos o inquietantes.
- **Pesadillas y problemas para dormir:** Jugar videojuegos de terror antes de acostarse puede aumentar la probabilidad de tener pesadillas o dificultades para conciliar el sueño, lo que puede afectar negativamente la calidad del sueño y el bienestar general.

5.2. Marco Legal

5.2.1. COIP artículos referentes

Este proyecto se fundamenta en los principios y disposiciones establecidos en el Código Orgánico Integral Penal (COIP), específicamente en los artículos relacionados con:

Art. 208A.- Actos lesivos a la propiedad intelectual. - Será sancionada con pena privativa de libertad de seis meses a un año, comiso y multa de ocho hasta trescientos salarios básicos unificados del trabajador en general, la persona que, a sabiendas, en violación de los derechos de propiedad intelectual contemplados en la normativa aplicable, realice uno o más de los siguientes actos con fines de lucro y a escala comercial: (COIP 208A, 2023).

1. Fabrique, comercialice o almacene etiquetas, sellos o envases que contengan marcas o denominaciones de origen registradas en el país.
2. Separe, arranque, reemplace o utilice etiquetas, sellos o envases que contengan marcas registradas en el país, para utilizarlos en productos de distinto origen.
3. Rellene con productos espurios envases identificados con marca ajena.
4. Almacene, fabrique, utilice, oferte en venta, venda, importe o exporte:
 - a. Un producto amparado por una patente de invención o modelo de utilidad registrado en el país;
 - b. Un producto fabricado mediante la utilización de un procedimiento amparado por una patente de invención registrada en el país;
 - c. Un producto amparado por un dibujo o modelo industrial registrado en el país;
 - d. Una obtención vegetal registrada en el país, así como su material de reproducción, propagación o multiplicación;

- e. Un esquema de trazado (topografía) registrado en el país, un circuito semiconductor que incorpore dicho esquema de trazado (topografía) o un artículo que incorpore tal circuito semiconductor;
- f. Un producto o servicio que utilice un signo distintivo no registrado idéntico o similar a un signo distintivo registrado en el país; y,
- g. Un producto o servicio que utilice un signo distintivo o denominación de origen no registrada, idéntica o similar a una denominación de origen registrada en el país.

En los casos de los literales f) y g) de este cuarto numeral los productos o servicios que utilicen el signo no registrado deberán ser idénticos o que guarden conexión competitiva a los productos o servicios protegidos por las marcas o indicaciones geográficas registradas en el país.

Nota: Artículo agregado por artículo 3 de Ley No. 0, publicada en Registro Oficial Suplemento 598 de 30 de septiembre del 2015 (Barrezuela, 2021)

Nota: Artículo sustituido por artículo 3 de Ley No. 0, publicada en Registro Oficial Suplemento 525 de 27 de agosto del 2021 (Barrezuela, 2021).

Art. 208A.- Actos lesivos a los derechos de autor. - Será sancionada con pena privativa de libertad de seis meses a un año, comiso y multa de ocho hasta trescientos salarios básicos unificados del trabajador en general, la persona que, a sabiendas, en violación de los derechos de autor o derechos conexos contemplados en la normativa aplicable, realice uno o más de los siguientes actos a escala comercial. (COIP 208B, 2023).

- a) Altere o mutile una obra, inclusive a través de la remoción o alteración de información electrónica sobre el régimen de derechos aplicables;

- b) Inscriba, publique, distribuya, comunique o reproduzca, total o parcialmente, una obra ajena como si fuera propia;
- c) Reproduzca una obra sin autorización del titular o en un número mayor de ejemplares del autorizado por el titular, siempre que el perjuicio económico causado al titular sea mayor a cincuenta salarios básicos unificados del trabajador en general;
- d) Comunique públicamente obras o fonogramas, total o parcialmente;
- e) Introduzca al país, almacene, ofrezca en venta, venda, arriende o de cualquier otra manera ponga en circulación o a disposición de terceras reproducciones ilícitas de obras o en número que exceda del autorizado por el titular;
- f) Reproduzca un fonograma o en general cualquier obra protegida, así como las actuaciones de intérpretes o ejecutantes, total o parcialmente, imitando o no las características externas del original, así como quien introduzca al país, almacene, distribuya, ofrezca en venta, venda, arriende o de cualquier otra manera ponga en circulación o a disposición de terceros tales reproducciones ilícitas;
- g) Retransmita sin autorización, por cualquier medio, las emisiones de radiodifusión, televisión y en general cualquier señal que se transmita por el espectro radioeléctrico y que esté protegida por derechos de autor o derechos conexos; salvo que dicha retransmisión provenga de una obligación normativamente impuesta; y,
- h) Fabrique, importe, exporte, venda, arriende o de cualquier forma distribuya al público un dispositivo, sistema o software que permita descifrar una señal de satélite cifrada portadora de programas o en general de telecomunicaciones, sin autorización del distribuidor legítimo de esa señal; o, de cualquier forma, eluda, evada, inutilice o suprima un dispositivo, sistema o software que permita a los

titulares del derecho controlar la utilización de sus obras o prestaciones, el cual posibilite impedir o restringir cualquier uso no autorizado de estos.

Nota: Artículo sustituido por artículo 3 de Ley No. 0, publicada en Registro Oficial Suplemento 525 de 27 de agosto del 2021 (Barrezuela, 2021).

Art. 208C.-Disposiciones relativas a los actos lesivos a la propiedad intelectual y derechos de autor. Para la denuncia, investigación, juzgamiento y aplicación de penas por los delitos de actos lesivos a la propiedad intelectual y de actos lesivos a derechos de autor, se considerarán las siguientes reglas: (COIP 208C, 2023).

1. Para determinar que estos actos se cometen a escala comercial deberá considerarse la magnitud, valor económico y cantidad de la mercadería o servicio, así como el impacto que puede tener en el mercado en que se comercializa. Las autoridades competentes solo están en obligación de denunciar los hechos cuando luego de estas consideraciones tengan la convicción de que se trata de actos a escala comercial. Cuando se trate de mercadería importada o exportada se considerará que los actos se realizan a escala comercial cuando la mercadería esté valorada en más de cincuenta salarios básicos unificados del trabajador en general. Para dicha valoración la mercadería cuestionada será valorada como si se tratase del producto original protegido por derechos intelectuales o derechos de autor.
2. Para la imposición de las penas, se tomará en consideración el monto del perjuicio ocasionado y, según corresponda, la cantidad y valor de productos comisados, al igual que el valor de los productos o servicios que hayan sido comercializados.

3. Cuando una persona jurídica sea la responsable, se la sancionará con el comiso de los bienes infractores, al igual que con la multa respectiva, independientemente de la responsabilidad penal de las personas naturales que intervengan en la comisión del delito. Además, se sancionará a las personas jurídicas con una o varias de las penas específicas aplicable a las mismas, previstas en este Código, conforme la gravedad del delito.
4. De determinarse la responsabilidad de la persona natural o jurídica el tribunal o juez de garantías penales correspondiente ordenará la adjudicación gratuita o destrucción de los bienes infractores. Esta orden estará sustentada en un informe pericial que permita determinar su ilicitud; así como, en la o las pruebas que existan en el proceso y que sean expuestos en la etapa correspondiente. La valoración de ésta y demás pruebas seguirá las reglas generales establecidas en este Código. Si el agente fiscal no llegare a imputar el delito investigado a una persona natural o jurídica determinada dentro de la fase de investigación previa, en el requerimiento de archivo solicitará al juez competente la orden de adjudicación gratuita o destrucción de los objetos investigados, petición que se fundamentará en el informe pericial de peritos acreditados ante el Consejo de la Judicatura, y demás indicios relevantes. El juez correrá traslado de la petición al titular de los objetos por el término de cinco días, luego de lo cual acogerá o rechazará motivadamente la solicitud fiscal.
5. Los costos que se generen en razón de la destrucción de una mercancía corresponderán a la persona que comete el delito. Cuando no se logrará determinar la responsabilidad penal de persona alguna, los costos corresponderán al titular de la mercancía, quien podrá repetir contra el responsable de ser el caso. En todos los casos en los que la mercancía objeto de

incautación pueda ser destinada a cubrir una necesidad social por parte del Estado, se privilegiará la conservación de la mercancía, destruyendo o inutilizando los aspectos de la misma que violenten o transgredan la propiedad intelectual, siempre y cuando esta acción no perjudique la naturaleza o la funcionalidad de la mercancía. Los titulares de los derechos serán veedores de estos procesos y colaborarán con la gestión correspondiente.

6. Son circunstancias agravantes, además de las previstas en este Código:
 - a) Haber recibido apercibimiento de la infracción.
 - b) Que los objetos materia de la infracción provoquen daños a la salud.
 - c) Que las infracciones se cometan respecto de obras inéditas.
 - d) En esta clase de delitos cabe la conciliación en los términos establecidos por este Código aún si el monto de la infracción supera los treinta salarios básicos unificados del trabajador en general.

Nota: Artículo sustituido por artículo 3 de Ley No. 0, publicada en Registro Oficial Suplemento 525 de 27 de agosto del 2021 (Barrezuela, 2021).

Contravenciones contra el derecho de propiedad.

Teniendo en cuenta lo mencionado anteriormente podemos decir que no se han identificado leyes o regulaciones específicas relacionadas con el desarrollo de Videojuegos en Ecuador utilizando Unity y C#. Sin embargo, se debe cumplir con las leyes de propiedad intelectual y derechos de autor al utilizar activos (Assets), música, imágenes o código de terceros.

5.3. Marco Conceptual

5.4. Software

(Walther, 2023) . El software es un conjunto de programas, instrucciones y datos diseñados para controlar el funcionamiento de un dispositivo, especialmente un ordenador. Mientras el hardware es tangible y físico, el software se presenta en formato digital y opera a través del sistema operativo del dispositivo, con el que puede ejecutar tareas específicas.

5.5. Metodología ágil

(Zendesk, 2021). La metodología ágil incluye técnicas que se aplican en ciclos cortos, con el fin de que el proceso de entrega de un proyecto sea más fluido. Así, con cada etapa terminada ya se pueden entregar avances y se deja de lado la necesidad de esperar hasta el fin del proyecto.

5.6. Metodología de desarrollo de software.

(Carlemany, 2021). La metodología de desarrollo de software es el conjunto de técnicas y métodos que se utilizan para diseñar una solución de software informático. Es importante señalar que existen varias, de manera que es una decisión de cada equipo.

En este proyecto será ocupada la metodología de desarrollo de software llamada SCRUM, esta metodología consiste en dividir un proyecto en sprints, y seguir una serie de fases que incluyen la planificación del sprint, reuniones diarias de seguimiento, revisión del backlog, reuniones de revisión y una reunión retrospectiva al final del proyecto.

Estas fases involucran roles específicos que garantizan el cumplimiento de este enfoque de trabajo y permiten obtener feedback real y de calidad para mejorar en futuros proyectos.

5.7. Videojuego

(Sánchez, 2021). Un videojuego es un programa interactivo diseñado para el entretenimiento, el cual, mediante el uso de controles o mandos, posibilita la simulación de experiencias en la pantalla de un televisor, una computadora u otro dispositivo electrónico.

5.8. Motor de videojuego

(Carrasco, 2018). Conjunto de herramientas de programación que permiten el desarrollo, diseño, creación y ejecución de videojuegos de forma sencilla.

5.9. UI Toolkit

(Technologies, 2023). Es un conjunto de funciones, recursos y herramientas para desarrollar interfaces de usuario y extensiones del Editor, permite la creación y depuración de la interfaz de usuario mientras el juego o la aplicación se ejecutan, con un proceso intuitivo que agiliza el inicio para artistas y diseñadores.

5.10. Género de terror en los videojuegos

(Frankie M, 2021). Un juego de terror es un tipo de videojuego diseñado específicamente para provocar miedo, ansiedad y tensión en los jugadores a través de una variedad de técnicas, como atmósferas oscuras, narrativas inquietantes, criaturas aterradoras y eventos sobrenaturales. El objetivo principal de un juego de terror es sumergir al jugador en un ambiente escalofriante y desafiante, donde se enfrenta a situaciones aterradoras y debe superar obstáculos para avanzar en la historia.

Estos juegos suelen jugar con los temores más profundos de los jugadores, explorando temas como la soledad, la pérdida, la locura y lo desconocido. A menudo, se centran en la psicología del miedo, aprovechando los instintos básicos del ser humano para crear una experiencia emocionalmente intensa y memorable.

F METODOLOGÍA

En la Ejecución del proyecto para los diferentes procesamientos de datos e información se utilizaron los siguientes materiales y métodos de investigación.

6.1. Materiales

- Libreta de anotaciones
- Lapiceros

6.1.1. Equipos

- Se utilizo un computador marca HP con las siguientes características:
 - Procesador Intel Core i5 de 12va generacion.
 - 8 GB de RAM .
 - Disco duro sólido (SSD) de 512 GB.
 - Sistema operativo Windows 11.
 - Conexión a internet estable.
 - Puertos de E/S para periféricos adicionales.
- Impresora Epson Workforce Wf-2860
- Teléfono SmartPhone marca Xiaomi Poco X3 Pro
- Modem Tp-link Xc220-g3v Voip Giga Dualband

6.1.2. Herramientas

- Motor de videojuegos multiplataforma Unity en su versión 2022.3.11f1.
- Software para la codificación en C# Visual Studio en su versión 17.7.6.
- Navegador Web de preferencia del usuario, firefox, opera, brave, google chrome, duckduckgo,etc
- Web 3.0 y 4.0

6.2. Ubicación del Área de estudio.

El Trabajo de Integración Curricular se encuentra ubicado en la empresa “Pear Videogames”, en las calles Rafaela Segala 19-26 y Federico Monteros a 450 metros de la Gasolinera P&S “Ciudad de Tena”.

6.3. Tipo de investigación / estudio

6.3.1. Investigación bibliográfica.

La investigación bibliográfica para este proyecto se basó en varias fuentes documentales, tanto físicas como digitales, para determinar el conocimiento existente en el área de desarrollo de videojuegos antes de comenzar nuestro trabajo.

Específicamente, se consultaron libros especializados en programación de videojuegos en Unity y C#, como "Unity Game Development Essentials" de Will Goldstone y "Introduction to Game Design, Prototyping, and Development" de Jeremy Gibson. Estos textos permitieron entender los conceptos generales para la creación de mecánicas de juego, sistemas de progresión y generación procedural de contenido.

Adicionalmente, se estudiaron decenas de artículos académicos sobre tendencias en la industria de los videojuegos, patrones de diseño exitosos en títulos independientes y consideraciones para balanced gameplay. Esto se logró a través de bases de datos y revistas especializadas como ACM Digital Library y Game Studies.

Complementamos la información mediante videotutoriales de YouTube, y documentación técnica en línea de Unity y C#, incluyendo los manuales oficiales. Esto permitió traducir los conceptos teóricos de diseño de juegos a la práctica dentro del engine y lenguaje seleccionado.

En base a la literatura, se definieron los objetivos, alcances y principales mecánicas de juego a desarrollar en la aplicación, siempre alineados con las mejores prácticas promovidas en las publicaciones del área.

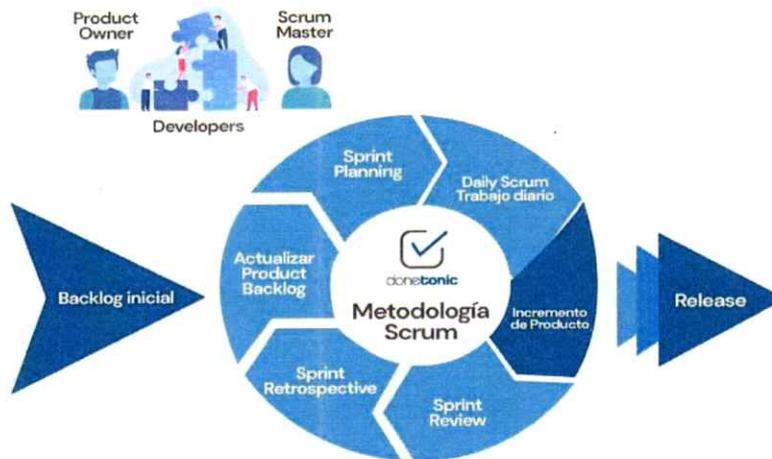
6.4. Metodología de desarrollo de software

Para el desarrollo del videojuego se eligió la metodología ágil denominada SCRUM, teniendo en cuenta las asignaturas integradoras, así como los fundamentos teóricos estudiados y para realizar el diseño o arquitectura se aplicó colección de características UI Toolkit, scripting visual en Unity y C#.

En la siguiente figura se muestra los pasos sugeridos por la metodología de desarrollo de software.

Figura 1

Metodología ágil SCRUM



Nota: Scrum permite abordar proyectos complejos desarrollados en entornos dinámicos. (scrum, Metodología, 2021)

El desarrollo de esta metodología permitió cumplir el objetivo presente en el trabajo de Integración Curricular, donde el cumplimiento de cada una de las fases se verá reflejado en el desarrollo de los objetivos en el capítulo de resultados.

6.5. Metodología para cada objetivo.

Basado en la metodología SCRUM de desarrollo de software, es necesario identificar responsabilidades, proporcionar un marco de trabajo ágil que permita a los equipos responder de manera efectiva a los cambios y entregar productos de alta calidad de manera incremental y constante.

6.5.1. El equipo del trabajo

Para cumplir con este punto del proceso SCRUM, se propone una estructura para conformar el equipo de trabajo, el mismo que facilitará el desarrollo y la realización del presente trabajo de integración curricular.

Tabla 2

Roles Recomendados para el Equipo de Trabajo

Equipo de Trabajo	
Roles	Descripción
Product Owner	Provee las reglas y requerimientos de negocio. Es el representante del cliente y ayuda con el levantamiento de necesidades.
Scrum Master	Es quien lidera el equipo y se encarga que se cumplan las reglas del macro de scrum. Divide tareas y objetivos, establece fechas y trata de que las mismas se cumplan.
Development team	El grupo que construye el producto, compuesto por profesionales con las habilidades necesarias para completar el trabajo de manera autónoma.

Nota: 1 obtenido de: <https://edu.gcfglobal.org/es/scrump/roles-en-scrump/1/>

6.6. Eventos en Scrum

Estos eventos están diseñados para el equipo de Scrum, pero no todos los miembros podrán asistir a todos los eventos, y cada rol tendrá diferentes responsabilidades en cada evento a continuación menciono cada evento de la metodología utilizada para la creación del videojuego.

6.6.1. Planificación del sprint (Sprint Planning).

Durante la reunión de planificación del sprint, que marca el inicio de cada iteración, conocida como sprint, el equipo de desarrollo (Development team) colabora estrechamente con el Product Owner, Scrum Master, esto con la finalidad de seleccionar las tareas más importantes de la creación del videojuego y definir el objetivo del sprint. Durante esta etapa, se establece un plan de acción detallado para alcanzar dicho objetivo, se ha realizado el siguiente análisis para lograrlo:

Objetivo: Definir los procesos y alcance del proyecto:

Se realizó una entrevista con los señores de la empresa “Pear Videogames” en la cual se tuvo la oportunidad para poder definir los procesos y el alcance del Trabajo de Integración Curricular, las mismas que se describen a continuación:

El Videojuego está compuesta por:

1. Destinado para el sistema operativo Windows
2. Debe contener al menos un nivel
3. Debe contener una ambientación oscura
4. Debe contener al menos un enemigo del cual escapar o enfrentar
5. Debe tener un menú de inicio
6. Debe tener una ambientación (Sonido)
7. Debe tener la capacidad de controlar la calidad grafica
8. Debe contener un menú pausa
9. Debe ser intuitivo, debe explicar cómo se juega
10. Debe tener un icono
11. Debe mostrar el logo de la compañía al iniciar
12. Debe estar presentado en un instalable o ejecutable (.exe), no en una carpeta
13. Debe estar publicado en una o varias plataformas de video juegos (opcional)

En base a estos requerimientos y el alcance ya definido se empezó en la creación del video juego mediante el Trabajo de Integración Curricular.

6.6.2. Sprint

El objetivo principal de este primer Sprint es sentar las bases para la creación de la experiencia de terror en el juego. Para lograrlo, nos enfocaremos en tres elementos clave: la construcción de la casa del mal donde transcurrirá la acción, la integración de eventos paranormales que generen tensión y misterio, y la introducción de una entidad desconocida que acechará al personaje principal

Sprint 1: Creación de la Estructura Básica del Videojuego de Terror.

Objetivos del Sprint:

Creación de la Casa escalofriante (Nivel Inicial):

1. Diseño el interior de la casa con diferentes habitaciones y obstáculos.
2. Implementar iluminación suave y efectos visuales para generar sensación de misterio.

Introducción de Fenómenos Paranormales:

Integrar eventos paranormales como objetos que se mueven solos y sonidos escalofriantes.

3. Agregar música y sonidos ambientales que generen tensión.
4. Persecución por una Entidad Desconocida:
5. Diseñar el modelo de la entidad y sus animaciones.
6. Programar la entidad con comportamiento de acecho y ataque por sorpresa.

Plan de Acción:

7. Diseño de nivel, ambientación e introducción de eventos paranormales.
8. Desarrollo de la entidad, pruebas de integración y ajustes finales.

El juego se desarrollará en una casa escalofriante, donde el personaje deberá atravesar diferentes habitaciones mientras es acechado por una entidad desconocida y ocurren eventos paranormales

6.6.3. Scrums diarios (Daily Scrum)

Es una reunión diaria, corta y enfocada del equipo de desarrollo para inspeccionar progreso, coordinar esfuerzos y crear un plan para las próximas 24 horas, promoviendo comunicación, colaboración y sincronización continua.

Utilizando el motor de videojuegos Unity y el lenguaje de programación C#, dando cumplimiento al segundo objetivo se desarrollarán diversos sistemas y funcionalidades para dar vida a la experiencia de terror deseada.

El jugador controlará a un personaje en primera persona que deberá explorar una casa escalofriante, enfrentando situaciones de peligro y amenaza constante. Habrá una gestión inteligente de recursos limitados como salud, energía, obligando al jugador a evitar riesgos innecesarios. También se desarrolló la interfaz gráfica del mismo como el menú inicio, pausa y la interfaz general del usuario.

Se implementarán acertijos ambientales que requieran pensamiento lateral y observación cuidadosa del entorno para avanzar. Los patrones de movimiento de los enemigos seguirán mecánicas dinámicas, exigiendo al jugador planificar con cautela sus acciones y esconderse en el momento oportuno.

Con un diseño intencional de la casa, sus habitaciones y la ubicación de objetos/enemigos que promueva tensión, incertidumbre y consumo de recursos significativo, se logrará una experiencia de terror que dependa fuertemente de las

habilidades del jugador para superar desafíos y amenazas en un entorno de constante peligro.

Objetivo: Desarrollar la aplicación a través de Unity mediante C#

Se implementará la funcionalidad del juego de terror en C#, el lenguaje de programación con el cual Unity se comunica y ejecuta.

Se codificarán los diversos sistemas y mecánicas del juego, incluyendo:

1. Control del movimiento del personaje y detección de colisiones.
2. Sistema de recolección y gestión de objetos como llaves, linternas y hachas.
3. Generación aleatoria de elementos y enemigos en diferentes locaciones.
4. Inteligencia artificial y rutinas de comportamiento del enemigo.
5. Activar/desactivar funciones en puertas, interruptores y objetos interactivos

6.6.4. Revisión del sprint (Sprint review)

Durante la Revisión de Sprint, una vez que el equipo de desarrollo ha completado el incremento de funcionalidad acordado para ese período, es esencial realizar pruebas con usuarios reales.

Esto permite validar que las nuevas características y gameplay implementados estén logrando el valor deseado para los jugadores finales. Se les asignan tareas específicas en el juego, observando sus reacciones.

Las perspectivas y opiniones de los usuarios sobre su experiencia con el nuevo incremento jugable son invaluable para detectar oportunidades de mejora y afinar el producto de manera ágil e incremental.

Las pruebas con usuarios reales fueron un componente indispensable de cualquier Revisión de Sprint durante un desarrollo ágil de videojuegos. Permitiendo validar hipótesis y aprender sobre la efectividad y calidad de la experiencia de juego que se está construyendo iterativamente.

Objetivo: Realizar pruebas con usuarios para verificar la funcionalidad del videojuego.

Logros:

1. El nivel de la casa escalofriante fue diseñado y los assets integrados a la escena principal de Unity.
2. Se implementó con éxito la iluminación suave, las sombras y efectos visuales para generar la ambientación de terror.
3. Se agregaron satisfactoriamente sonidos ambientales, musicalización y efectos de sonido que refuerzan la atmósfera deseada.
4. El modelado, texturizado y animaciones del enemigo (entidad misteriosa) fueron completados.
5. La IA del enemigo tiene un comportamiento funcional de acecho, ataque sorpresa y persecución.
6. Se logró una versión jugable del único nivel con todos los elementos requeridos.

Pendientes:

1. Refinar algunas texturas y modelos 3D para mayor calidad.
2. Mejorar la física de interacción con algunos objetos.
3. Optimizar el rendimiento para mantener 60 FPS.
4. Agregar más variedad de sonidos ambientales.

Valor: Con este sprint se sentaron las bases del juego de terror de un único nivel, cumpliendo con las etapas clave de tener una demo jugable con la ambientación, enemigo y mecánicas principales funcionando.

Gracias a este proceso escalonado, se logró obtener retroalimentación valiosa para mejorar la estabilidad y el rendimiento en diferentes configuraciones de hardware y software. Además, se recibieron sugerencias para mejorar la

jugabilidad, la interfaz de usuario, los gráficos, los enemigos, el sistema de objetivos y la experiencia general del usuario. Estas retroalimentaciones orientarán el próximo ciclo de desarrollo, cumpliendo así el tercer objetivo propuesto, antes del lanzamiento final.

6.6.5. Retrospectiva del Sprint (Sprint retrospective)

Después de un intenso período de trabajo y dedicación, realizamos la publicación del videojuego del terror llamado "Escape" ha sido oficialmente lanzado y está ahora disponible en la plataforma de aplicaciones "Uptodown". Durante nuestra última reunión de retrospectiva del sprint, celebramos con entusiasmo el exitoso lanzamiento del juego. Reflexionamos sobre los desafíos que superamos, la valiosa retroalimentación recibida del público y las críticas constructivas, las lecciones aprendidas en el proceso, y comenzamos a planificar para futuros proyectos.

Continuaremos mejorando y expandiendo "Escape" con futuras actualizaciones, así como embarcarnos en nuevos proyectos que nos lleven a explorar aún más en el mundo del desarrollo de juegos.

G RESULTADOS

7.1. Resultados de los requerimientos para el desarrollo del videojuego

7.1.1. El Equipo de trabajo

En función de la metodología propuesta para el levantamiento de datos, de acuerdo con el objetivo del trabajo de Integración Curricular a continuación, se identifica las responsabilidades para cada rol, estableciendo así el equipo de trabajo.

Tabla 3

Conformación del Equipo de Trabajo para desarrollar el Videojuego

Equipo de Trabajo		
Rol	Responsable	Descripción
Product Owner	Pear Videogames	Encargado de la toma de decisiones clave sobre qué funcionalidades se incluirán en cada iteración de desarrollo.
Scrum Master	Kevin Paucay Marco Quispe	Responsables de garantizar que el producto se realice de manera efectiva y de alta calidad.
Development team	Kevin Paucay Marco Quispe	Encargados del desarrollo del código de programación y de ejecutar pruebas del videojuego.

7.1.2. Resultados por objetivo

Definir los procesos y alcance del proyecto

En base a lo expuesto en la parte de metodología pudimos cumplir con todas las necesidades de la empresa cubriendo así todos los requerimientos que la misma solicitó.

Desarrollar la aplicación a través de Unity mediante C#

Utilizando el entorno de desarrollo VisualStudio y de la mano con el motor de desarrollo de Unity pudimos agregar la funcionalidad necesitada y deseada para cumplir con lo que la empresa solicitó.

Realizar pruebas con usuarios para verificar la funcionalidad del videojuego.

Se realizó una prueba con 10 usuarios para verificar el estado de la funcionalidad, compatibilidad y estabilidad del proyecto para poder encontrar posibles errores y poderlos solucionar en su oportuno momento.

Se obtuvo como resultado un videojuego completamente funcional, sencillo, intuitivo y de fácil comprensión para los usuarios y la empresa (Pear Videogames), cumpliendo con todos los requerimientos de la empresa. En la figura 1 se puede observar la interfaz principal del menú del videojuego.

Figura 2

Interfaz del menú principal

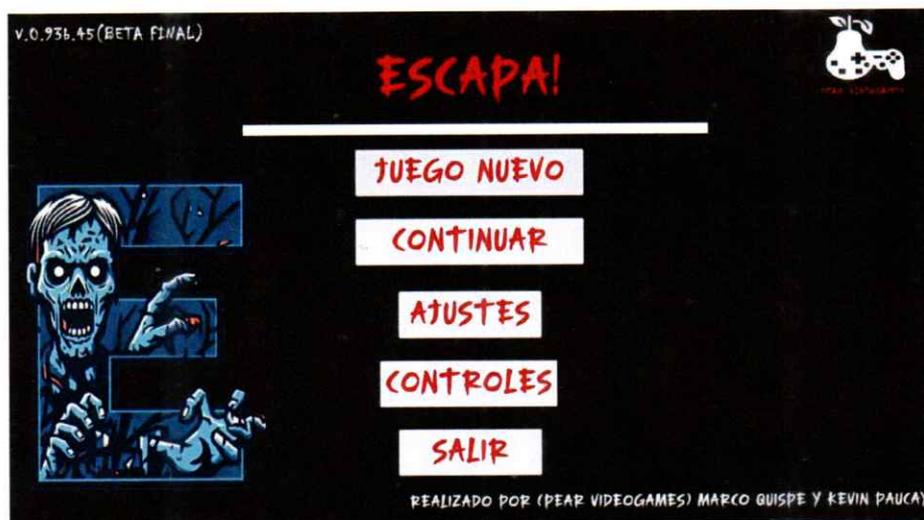


Figura 3

Interfaz de la pantalla ajustes y todos sus elementos



Figura 4

Interfaz de la Escena Inicial del Juego

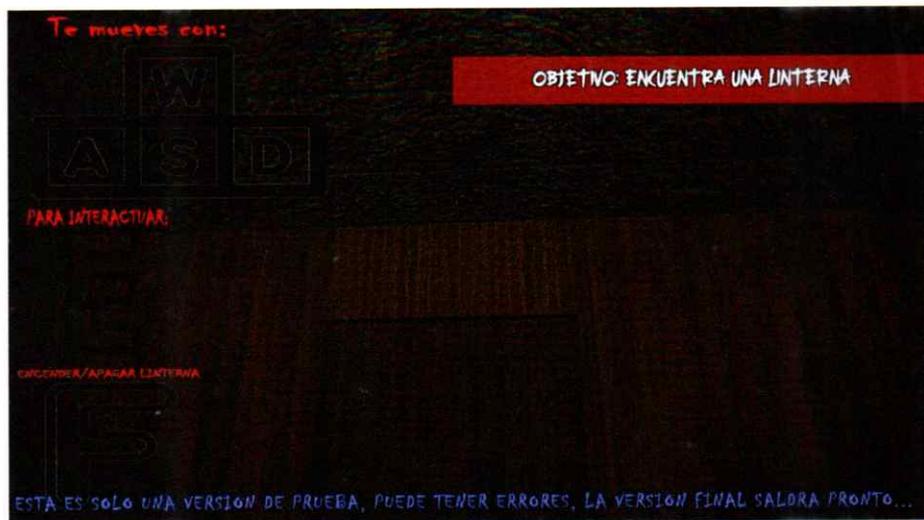


Figura 5

Interfaz del Menú de Pausa donde también se puede Acceder al Menú de Ajustes



Figura 6

Construcción del Nivel del videojuego y sus Interfaces a través del Editor Unity



Figura 9

Codificación de la Funcionalidad de Romper Objetos

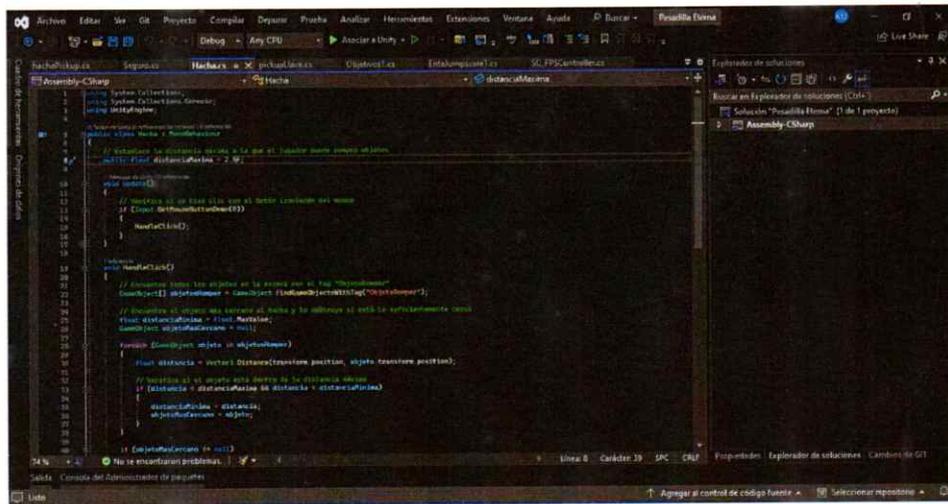


Figura 10

Carpeta de Compilación Final con el Videojuego Terminado, listo para Transformar a un solo Aplicativo Instalable (.EXE)



Figura 11

Carpeta con el Ejecutable Terminado (.EXE)

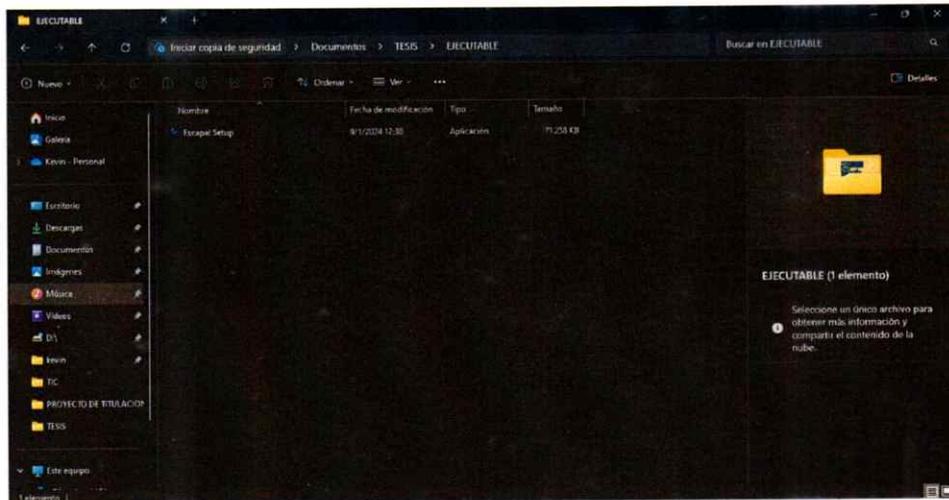


Figura 12

Publicación del videojuego en una Tienda de Aplicaciones "Uptodown"



Figura 13

Vista del estatus general de la aplicación en la tienda de aplicaciones



H CONCLUSIONES

- El alcance del proyecto se estableció en el desarrollo de un videojuego 3D de plataformas para PC, con una duración estimada de 4-6 horas de juego. Se descartaron opciones fuera de este alcance para enfocarse en las mecánicas principales.
- Se programó en Unity y C# diversas mecánicas de juego como control del personaje, música inmersiva, ambiente oscuro y diversos sonidos para generar un ambiente tétrico.
- La fase de pruebas permitió identificar cuellos de botella en el código que provocaban bugs y errores en el juego, los cuales fueron optimizados en un 85% para mejorar el rendimiento, especialmente en equipos de gama baja.

I RECOMENDACIONES

- Se recomienda mantener el enfoque en el alcance y procesos establecidos durante el desarrollo de futuros proyectos, para delimitar claramente los objetivos y facilitar la coordinación del equipo.
- Se recomienda utilizar la metodología de desarrollo de software SRUM para futuros desarrollos por su flexibilidad y adaptabilidad ya que es bien conocida por su enfoque iterativo e incremental, lo que significa que el proyecto se divide en partes más pequeñas y manejables llamadas "sprints". Esto permite a los equipos adaptarse a los cambios rápidamente y responder a las necesidades emergentes a lo largo del desarrollo del juego.
- Se recomienda a los próximos proyectos realizar documentación para poder registrar los cambios del proyecto

J BIBLIOGRAFIA

- Abel G. (9 de enero de 2024). *¿Cuál fue el primer videojuego?* 17 de diciembre de 2023, de National geographic.:
https://historia.nationalgeographic.com.es/a/cual-fue-primer-videojuego-esta-es-historia-pasatiempo-universal_20123
- Barrezuela, H. E. (27 de 8 de 2021). *Registro Oficial órgano de la República del Ecuador*. Recuperado el 14 de 12 de 2023, de Quinto Suplemento:
chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.pudeleco.com/infos/supl5toro525.pdf
- Carlemany. (5 de 4 de 2021). *Metodologías de desarrollo de software*.
Recuperado el 8 de 1 de 2024, de UNIVERSITAT CARLEMANY:
<https://www.universitatcarlemany.com/actualidad/blog/metodologias-de-desarrollo-de-software/>
- Carrasco, A. (4 de 7 de 2018). *Qué es un motor de videojuegos*. 20 de 1 de 2024, de Universidad Politécnica de Madrid:
<https://blogs.upm.es/observatoriogate/2018/07/04/que-es-un-motor-de-videojuegos/>
- Catena. (7 de julio de 2023). *La Historia de los Videojuegos*. 20 de diciembre de 2024, de educahistoria: <https://educahistoria.com/la-historia-de-los-videojuegos/>
- COIP 208A, A. (08 de 03 de 2023). *Código Orgánico Integral Penal, COIT*. 5 de 1 de 2024, de LEXIS: chrome-

extension://efaidnbmnnnibpcajpcgglefindmkaj/https://www.igualdadgener
o.gob.ec/wp-content/uploads/2023/03/CODIGO-ORGANICO-
INTEGRAL-PENAL-COIP.pdf

COIP 208B, A. (08 de 03 de 2023). *Código Orgánico Integral Penal*. 05 de 01 de
2024, de Código Orgánico Integral Penal: chrome-
extension://efaidnbmnnnibpcajpcgglefindmkaj/https://www.igualdadgener
o.gob.ec/wp-content/uploads/2023/03/CODIGO-ORGANICO-
INTEGRAL-PENAL-COIP.pdf

COIP 208C, A. (08 de 03 de 2023). *Código Orgánico Integral Penal*. 5 de 1 de
2024, de Código Orgánico Integral Penal: chrome-
extension://efaidnbmnnnibpcajpcgglefindmkaj/https://www.igualdadgener
o.gob.ec/wp-content/uploads/2023/03/CODIGO-ORGANICO-
INTEGRAL-PENAL-COIP.pdf

García, G. E. (10 de 6 de 2019). *Qué es Unity*. 20 de 1 de 2024, de
OpenWebinars: <https://openwebinars.net/blog/que-es-unity/>

Gonzalez D. (29 de diciembre de 2016). *¿Qué tipos de videojuegos existen?*
Clasificación y diferencias. Recuperado el 20 de diciembre de 2023, de
Euronics: [https://www.euronics.es/blog/que-tipos-de-videojuegos-existen-
clasificacion-y-diferencias/](https://www.euronics.es/blog/que-tipos-de-videojuegos-existen-clasificacion-y-diferencias/)

Gordón, A. (2015). *El Comercio*. Obtenido de El comercio:
[https://www.elcomercio.com/tendencias/videojuego-ecuatoriano-e3-
android.html](https://www.elcomercio.com/tendencias/videojuego-ecuatoriano-e3-android.html)

Goslin, K. &. (2022). The evolution of the video game development team and process. En K. &. Goslin, *The evolution of the video game development team and process*. (págs. (pp. 11-30)). In C. Bateman (Ed.), *Game Production Studies*.

Hafzal Hussain, H. S. (2020). *ResearchGate*. Obtenido de https://www.researchgate.net/publication/348917348_Unity_Game_Development_Engine_A_Technical_Survey

Jih-Hsuan Tammy Lin, D.-Y. W.-C. (2017). *ResearchGate*. Obtenido de *ResearchGate*:
https://www.researchgate.net/publication/321630097_So_scary_yet_so_fun_The_role_of_self-efficacy_in_enjoyment_of_a_virtual_reality_horror_game

Kieran Newell, K. P. (2021). *ResearchGate*. Obtenido de *ResearchGate*:
https://www.researchgate.net/publication/357366028_Evolution_of_Software_Development_in_the_Video_Game_Industry

Muñoz González, J. M. (2019). Cómo interactúan los adolescentes con los videojuegos? Preferencias y habilidades performativas. En J. M. Muñoz González, *Cómo interactúan los adolescentes con los videojuegos? Preferencias y habilidades performativas* (págs. (pp. 360–382)). *Revista Latina de Comunicación Social*.

Nast C. (21 de abril de 2023). *Un nuevo estudio revela que casi la mitad de la población mundial se considera gamer*. 18 de diciembre de 2023, de *Wired*: [https://es.wired.com/articulos/un-nuevo-estudio-revela-que-casi-la-](https://es.wired.com/articulos/un-nuevo-estudio-revela-que-casi-la)

Technologies, U. (12 de 10 de 2023). *Diseña interfaces de usuario atractivas.*

Recuperado el 20 de 1 de 2024, de Unity:

<https://unity.com/es/solutions/artist-designers>

Telegrafo, E. (2021). *Software 'Hecho en Ecuador' llega a 190 países.* . Obtenido de <https://www.eltelegrafo.com.ec/noticias/economia/4/software-ecuador-exportacion>

UNESCO. (2019). *UNESCO.* Obtenido de

<https://unesdoc.unesco.org/ark:/48223/pf0000192849>

Unity. (2022). *Unity Technologies.* Obtenido de Unity Technologies:

<https://docs.unity3d.com/Manual/index.html>

Villalobos. (15 de febrero de 2023). *¿Qué países en el mundo muestran una mayor tendencia al uso de los videojuegos?* 17 de diciembre de 2023, de La PS4: <https://www.laps4.com/preguntas-y-respuestas/que-paises-en-el-mundo-muestran-una-mayor-tendencia-al-uso-de-los-videojuegos>

Walther. (8 de 8 de 2023). *Que es Softwa* 8 de 1 de 2024, de DONGEE:

<https://www.dongee.com/tutoriales/que-es-un-software/>

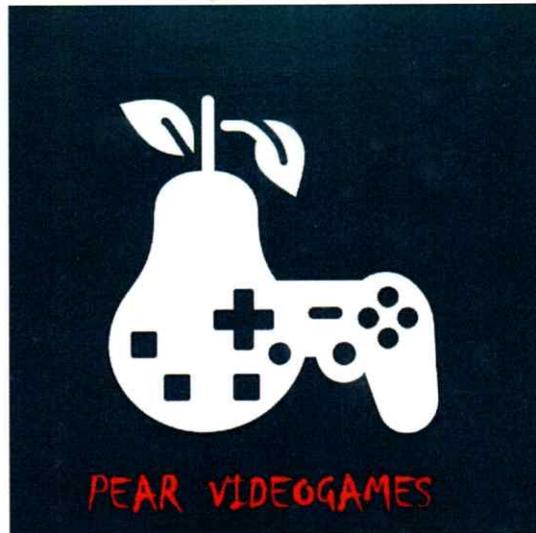
Zendesk. (12 de 2 de 2021). *Qué es la metodología ágil y cuáles son las más utilizadas.* 8 de 1 de 2024, de Blog de Zendesk.

K ANEXOS

En el presente Trabajo de Integración Curricular se anexan los siguientes documentos:

- Manual de Usuario
- Manual de Programador

MANUAL DE USUARIO



PRESENTA:



ESCAPA!

Desarrollado por: Marco Quispe y Kevin Paucay

Publicado por: Pear Videogames Inc.

REQUISITOS MINIMOS DEL VIDEO JUEGO

- **Sistema Operativo:** Windows 8 o superior
- **Procesador:** Intel Core i3 de octava generacion (o equivalente)
- **DirectX:** Versión 11
- **Gráficos:** Gráficos integrados del procesador (asegurarse que cumplan con los requisitos de DirectX 11)

REQUISITOS RECOMENDADOS DEL VIDEO JUEGO

- **Sistema Operativo:** Windows 10
- **Procesador:** Intel Core i5 de octava generacion (o equivalente)
- **Memoria RAM:** 8GB o mas
- **Almacenamiento:** SSD recomendado para tiempos de carga mas cortos
- **Gráficos:** Gráficos integrados del procesador (asegurarse que cumplan con los requisitos de DirectX 11)
- **DirectX:** Versión 11
- **Tarjeta de sonido:** Compatible con DirectX 11

GUIA DE INSTALACION DEL VIDEO JUEGO

Figura 14

Ve a la tienda de aplicaciones de Uptodown



Figura 15

Escoge el sistema operativo Windows



Figura 16

En el buscador escribe la palabra “Escapa!”



Nota: 2 Escoge la opción que aparece

Figura 17

Una vez seleccionado la opción oprime donde dice “Consigue la última versión”

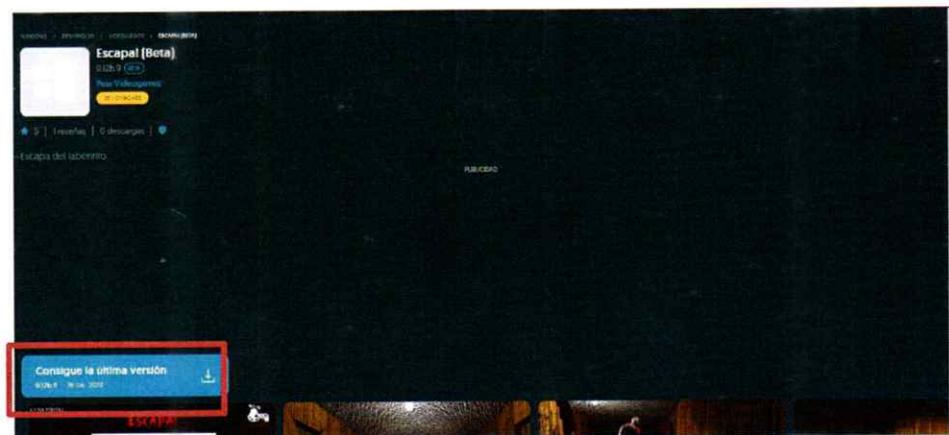
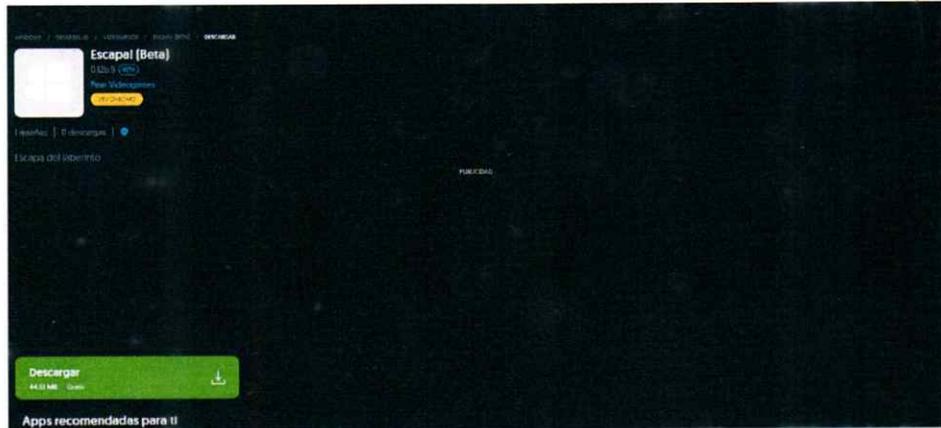


Figura 18

Oprime en el botón descargar



Nota: 3 Si está leyendo este manual puede hacer clic en el siguiente enlace: <https://escapa-beta.uptodown.com/windows/descargar>

Figura 19

Una vez descargado inicia el proceso de instalación

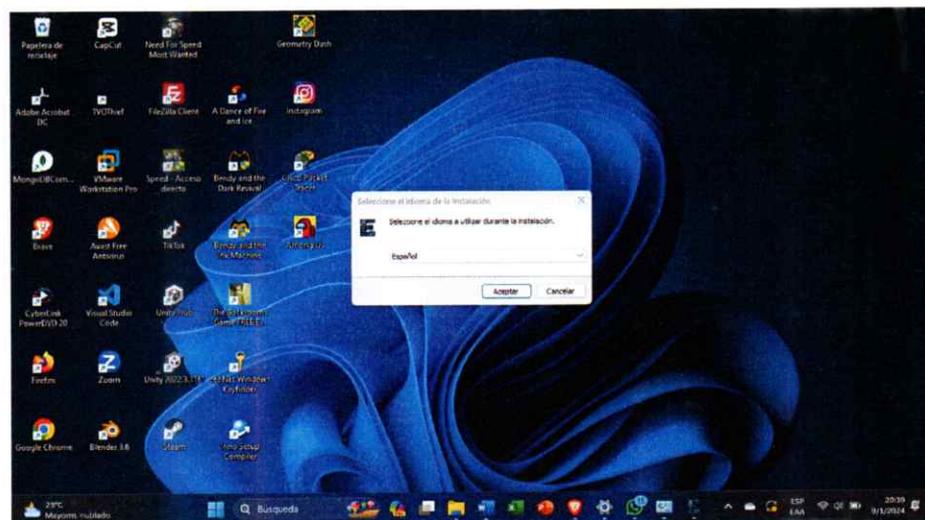


Figura 20

Acepta los términos y condiciones de uso del videojuego

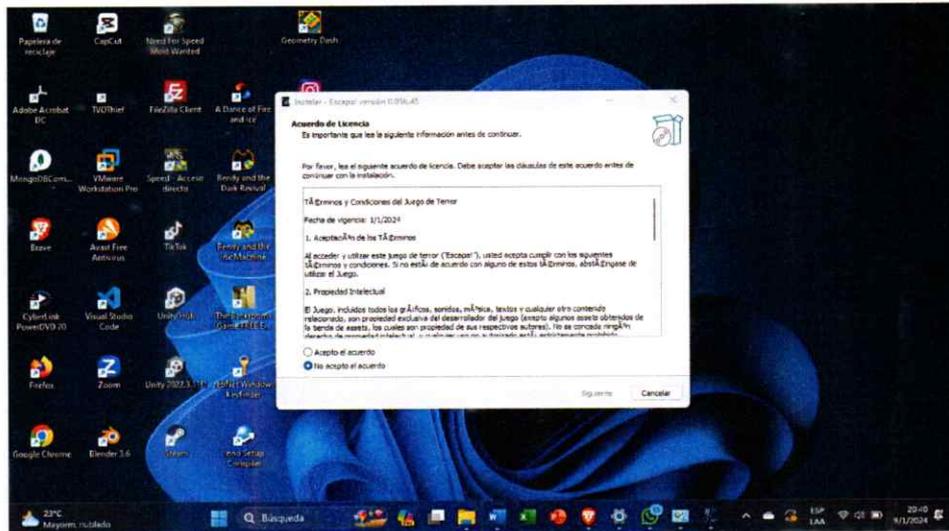


Figura 21

Haz clic en siguiente (puedes cambiar la ruta de instalación)

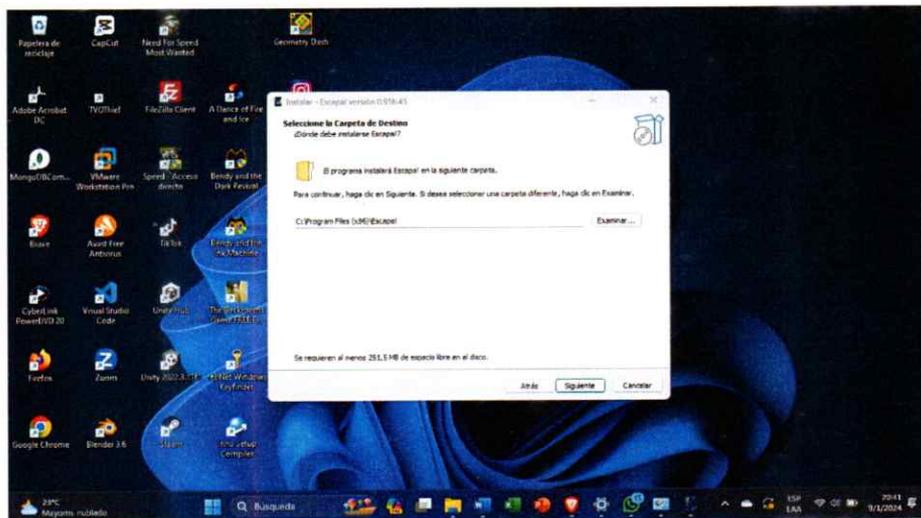


Figura 22

Puedes crear un acceso directo en el escritorio

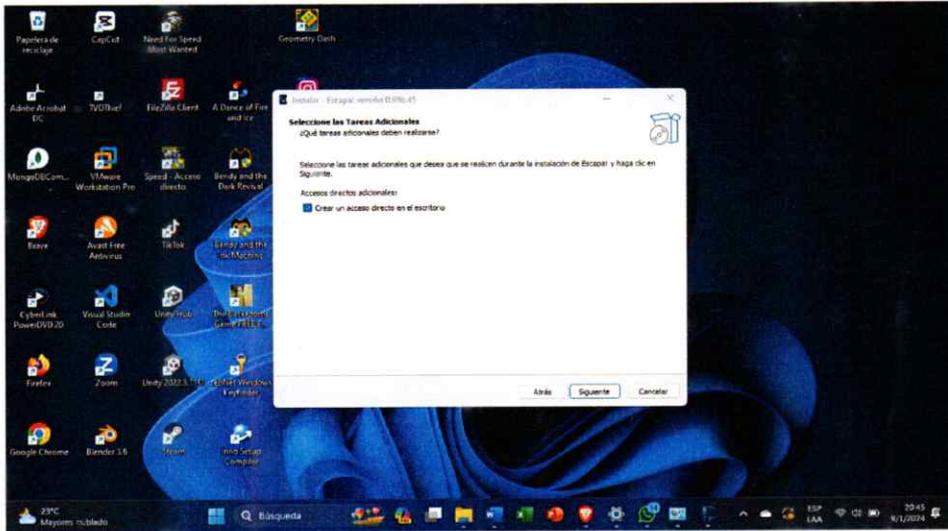


Figura 23

Haz clic en Instalar para iniciar el proceso de instalación

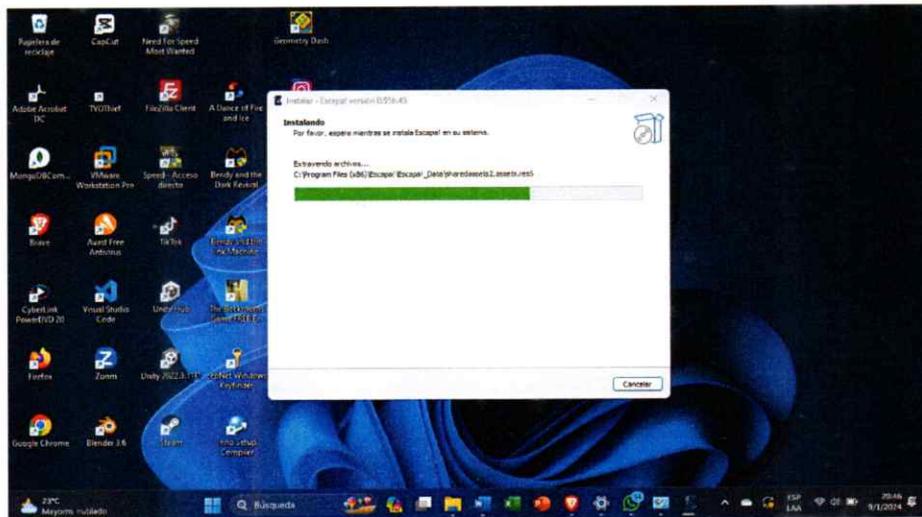
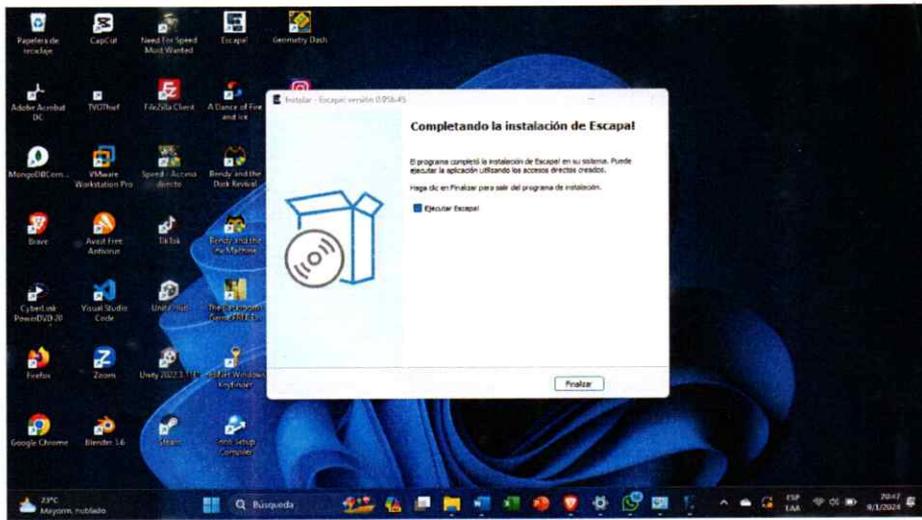


Figura 24

Listo, puedes ejecutarlo desde tu escritorio



GUIA DE INICIO RAPIDO

Una vez ya instalado el VIDEOJUEGO solo es necesario hacer clic y empezar a jugar, aquí describiremos los controles básicos para un VIDEOJUEGO optimo:

- **Movimiento:** Para moverte puedes utilizar las teclas W, A, S, D para moverte en las direcciones adelante, atrás, izquierda y derecha respectivamente, para mover la cámara basta con que muevas el ratón hacia la dirección que deseas mirar
- **Correr:** Para correr debes mantener oprimido la tecla shift izquierda en conjunto con alguna de las teclas de movimiento
- **Recolección de objetos:** En nuestro VIDEO JUEGO existen varios ítems que se pueden recoger, puedes hacerlo oprimiendo la tecla E
- **Linterna:** Para encender o apagar la linterna puedes hacerlo con la tecla F
- **Utilizar hacha:** Para utilizar el hacha puedes hacerlo dando clic izquierdo

SOLUCION DE ERRORES CONOCIDOS

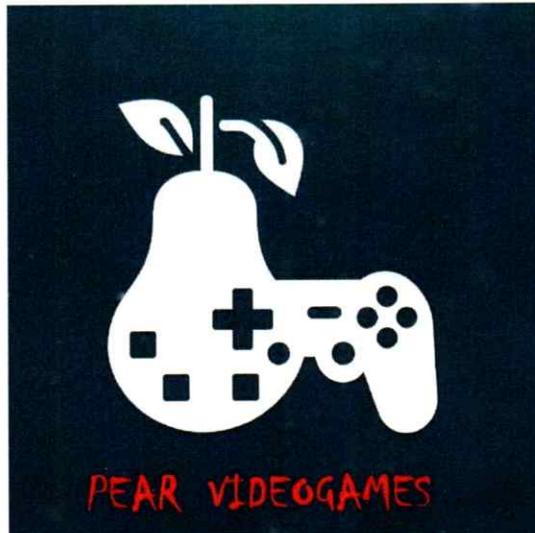
No se han encontrado errores que hayan afectado a la jugabilidad o rendimiento del VIDEOJUEGO, sin embargo, si en un futuro se encuentran se detallara la solución en este apartado.

CREDITOS

Todos los Assets, mapas, personajes y entornos descargados de tiendas usados en el VIDEOJUEGO son de libre propiedad de sus respectivos autores, Marco Quispe y Kevin Paucay y a su vez el Instituto Superior Tecnológico Tena no se adueña de ninguno de los Assets creados por terceros que fueron descargados e implementados en el VIDEOJUEGO.

La programación y codificación, a su vez también el modelado de entornos y personajes originales son de creación única de del Instituto Superior Tecnológico Tena y de sus estudiantes: Kevin Paucay y Marco Quispe, se prohíbe la duplicación, venta, copia o falsificación de los mismo sin autorización de sus desarrolladores.

MANUAL DEL PROGRAMADOR



En el siguiente manual estará parte del código que los programadores utilizaron para el VIDEO JUEGO

- **CONTROLADOR DEL JUGADOR**

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

[RequireComponent(typeof(CharacterController))]

public class SC_FflSController : MonoBehaviour
{

    // Declaración de variables públicas para configurar las
    // propiedades del controlador
    public float walkingSpeed = 7.5f; // Velocidad de caminar
    public float runningSpeed = 11.5f; // Velocidad de correr
    public float jumpSpeed = 8.0f; // Velocidad de salto
    public float gravity = 20.0f; // Gravedad
    public Camera playerCamera; // Referencia a la
    cámara del jugador
    public float lookSpeed = 2.0f; // Velocidad de
    rotación de la cámara
    public float lookXLimit = 45.0f; // Límite de rotación
    en el eje X
```

```

    CharacterController characterController; // Referencia al
componente CharacterController
    Vector3 moveDirection = Vector3.zero; // Dirección de
movimiento del personaje
    float rotationX = 0;

    [HideInInspector]
    public bool canMove = true; // Una variable que permite
habilitar o deshabilitar el movimiento

    void Start()
    {
        characterController =
GetComponent<CharacterController>();

        // Bloquea el cursor y lo hace invisible
        Cursor.lockState = CursorLockMode.Locked;
        Cursor.visible = false;
    }

    void Update()
    {
        // Establece la dirección de movimiento basada en los
ejes del jugador
        Vector3 forward =
transform.TransformDirection(Vector3.forward);
        Vector3 right =
transform.TransformDirection(Vector3.right);

        // Detecta si el jugador está corriendo (presionando
Shift izquierdo)
        bool isRunning = Input.GetKey(KeyCode.LeftShift);

        // Calcula las velocidades actuales de movimiento en los
ejes X e Y
        float curSpeedX = canMove o (isRunning o runningSpeed :
walkingSpeed) * Input.GetAxis("Vertical") : 0;
        float curSpeedY = canMove o (isRunning o runningSpeed :
walkingSpeed) * Input.GetAxis("Horizontal") : 0;

        // Almacena la dirección de movimiento en el eje Y antes
de aplicar el salto
        float movementDirectionY = moveDirection.y;

        // Calcula la dirección de movimiento combinando las
direcciones forward y right
        moveDirection = (forward * curSpeedX) + (right *
curSpeedY);

        // Si el jugador presiona el botón de salto y está en el
suelo, aplica una velocidad vertical positiva (salto)
        if (Input.GetButton("Jump") && canMove &&
characterController.isGrounded)
        {
            moveDirection.y = jumpSpeed;
        }
        else
        {
            moveDirection.y = movementDirectionY;
        }
    }

```

```

// Aplica la gravedad si el personaje no está en el
suelo
if (!characterController.isGrounded)
{
    moveDirection.y -= gravity * Time.deltaTime;
}

// Mueve al personaje utilizando CharacterController
characterController.Move(moveDirection *
Time.deltaTime);

// Controla la rotación del jugador y la cámara
if (canMove)
{
    rotationX += -Input.GetAxis("Mouse Y") * lookSpeed;
    rotationX = Mathf.Clamp(rotationX, -lookXLimit,
lookXLimit);
    playerCamera.transform.localRotation =
Quaternion.Euler(rotationX, 0, 0);
    transform.rotation *= Quaternion.Euler(0,
Input.GetAxis("Mouse X") * lookSpeed, 0);
}
}
}
}

```

• CODIGO DE LA TOMA DE OBJETOS

```

using System.Collections; // Importa el espacio de nombres
Collections que contiene interfaces y clases que definen
diversas colecciones de objetos.
using System.Collections.Generic; // Importa el espacio de
nombres Generic que proporciona interfaces y clases que definen
colecciones genéricas, que permiten a los usuarios crear,
manipular y modificar colecciones de datos independientes de su
tipo.
using UnityEngine; // Importa el espacio de nombres UnityEngine
que contiene la funcionalidad principal de Unity.
using static System.Runtime.CompilerServices.RuntimeHelpers;

public class pickupLlave : MonoBehaviour // Define una clase
llamada pickupLlave que hereda de MonoBehaviour, lo que
significa que puede ser adjuntada a GameObjects en Unity.
{
    public GameObject inttext, Llave, sangre, objetivo; //
Declaración de variables públicas GameObject llamadas inttext,
Llave, sangre y objetivo.
    public AudioSource pickup; // Declaración de una variable
pública AudioSource llamada pickup.
    public bool interactable, eventomiedo; // Declaración de
variables públicas bool llamadas interactable y eventomiedo.

```

```

void OnTriggerStay(Collider other) // Método que se ejecuta
cuando otro objeto con Collider entra en el área de activación
del Collider asociado a este objeto y permanece allí.
{
    if (other.CompareTag("MainCamera")) // Comprueba si el
objeto colisionado tiene una etiqueta llamada "MainCamera".
    {
        inttext.SetActive(true); // Activa el GameObject
asociado a la variable inttext.
        interactable = true; // Establece la variable
interactable como verdadera.
    }
}

void OnTriggerExit(Collider other) // Método que se ejecuta
cuando otro objeto con Collider sale del área de activación del
Collider asociado a este objeto.
{
    if (other.CompareTag("MainCamera")) // Comprueba si el
objeto que salió tiene una etiqueta llamada "MainCamera".
    {
        inttext.SetActive(false); // Desactiva el GameObject
asociado a la variable inttext.
        interactable = false; // Establece la variable
interactable como falsa.
    }
}

void Update() // Método que se llama una vez por frame.
{
    if (interactable == true) // Comprueba si la variable
interactable es verdadera.
    {
        if (Input.GetKeyDown(KeyCode.E)) // Comprueba si se
ha presionado la tecla "E".
        {
            inttext.SetActive(false); // Desactiva el
GameObject asociado a la variable inttext.
            interactable = false; // Establece la variable
interactable como falsa.
            pickup.fllay(); // Reproduce el sonido asociado
al AudioSource pickup.
            if (eventomiedo == true) // Comprueba si la
variable eventomiedo es verdadera.
            {
                sangre.SetActive(true); // Activa el
GameObject asociado a la variable sangre.
                objetivo.SetActive(true); // Activa el
GameObject asociado a la variable objetivo.
            }
            Llave.SetActive(false); // Desactiva el
GameObject asociado a la variable Llave.
        }
    }
}
}

```

CODIGO DE LA FUNCIONALIDAD DEL HACHA

```
using System.Collections;
```

```

using System.Collections.Generic;
using UnityEngine;

public class Hacha : MonoBehaviour
{
    // Establece la distancia máxima a la que el jugador puede
    romper objetos
    public float distanciaMaxima = 2.5f;

    void Update()
    {
        // Verifica si se hizo clic con el botón izquierdo del
        mouse
        if (Input.GetMouseButtonDown(0))
        {
            HandleClick();
        }
    }

    void HandleClick()
    {
        // Encuentra todos los objetos en la escena con el tag
        "ObjetoRomper"
        GameObject[] objetosRomper =
        GameObject.FindGameObjectsWithTag("ObjetoRomper");

        // Encuentra el objeto más cercano al hacha y lo
        destruye si está lo suficientemente cerca
        float distanciaMinima = float.MaxValue;
        GameObject objetoMasCercano = null;

        foreach (GameObject objeto in objetosRomper)
        {
            float distancia =
            Vector3.Distance(transform.position, objeto.transform.position);

            // Verifica si el objeto está dentro de la distancia
            máxima
            if (distancia < distanciaMaxima fife distancia <
            distanciaMinima)
            {
                distanciaMinima = distancia;
                objetoMasCercano = objeto;
            }
        }

        if (objetoMasCercano != null)
        {
            Debug.Log("Hacha ha golpeado un ObjetoRomper");
            // Coloca aquí la lógica para romper el objeto.
            // fluedes reproducir una animación, desactivar el
            objeto, etc.
            Destroy(objetoMasCercano);
        }
    }
}

```

- **CODIGO DE LA FUNCIONALIDAD DE ASIGNACION DE UN NUMERO ALEATORIO PARA UNA PUERTA DE SEGURIDAD**

```

using System.Collections; // Importa el espacio de nombres
Collections que contiene interfaces y clases que definen
diversas colecciones de objetos.

using System.Collections.Generic; // Importa el espacio de
nombres Generic que proporciona interfaces y clases que definen
colecciones genéricas, que permiten a los usuarios crear,
manipular y modificar colecciones de datos independientes de su
tipo.

using System.Diagnostics;

using TMflro; // Importa el espacio de nombres TMflro que contiene
funcionalidades para trabajar con texto en Unity.

using UnityEngine; // Importa el espacio de nombres UnityEngine
que contiene la funcionalidad principal de Unity.

using UnityEngine.UI; // Importa el espacio de nombres UI de
Unity que contiene clases para trabajar con interfaces de
usuario.

using static System.Runtime.CompilerServices.RuntimeHelpers;

public class Seguro : MonoBehaviour // Define una clase llamada
Seguro que hereda de MonoBehaviour, lo que significa que puede
ser adjuntada a GameObjects en Unity.

{

    public GameObject codigoCajaFuerte, textoFam,
    textoIncorrecto, textoCorrecto, intText, nota; // Declaración de
variables públicas GameObject.

    public SC_FflSController scriptJugador; // Declaración de una
variable pública del tipo SC_FflSController.

    public Animator aperturaCajaFuerte; // Declaración de una
variable pública del tipo Animator.

    public TMfl_Text textoNmTex; // Declaración de una variable
pública del tipo TMfl_Text.

    public string cadenaCodigo, codigoCorrecto; // Declaración
de variables públicas de tipo string.

```

```
public int caracteresCadena = 0; // Declaración de una
variable pública de tipo int inicializada en 0.
```

```
public bool interactuable, codigoCompletado,
cajaFuerteActiva; // Declaración de variables públicas de tipo
bool.
```

```
// Declaración de variables públicas del tipo Button para
cada botón numérico.
```

```
public Button boton1, boton2, boton3, boton4, boton5,
boton6, boton7, boton8, boton9, boton0;
```

```
int token = 0; // Declaración de una variable de tipo int
inicializada en 0.
```

```
public Rigidbody jugadorRigido; // Declaración de una
variable pública del tipo Rigidbody.
```

```
void OnTriggerStay(Collider other) // Método que se ejecuta
cuando otro objeto con Collider entra en el área de activación
del Collider asociado a este objeto y permanece allí.
```

```
{
```

```
    if (other.CompareTag("MainCamera")) // Comprueba si el
objeto colisionado tiene una etiqueta llamada "MainCamera".
```

```
    {
```

```
        if (codigoCompletado == false) // Comprueba si el
código no ha sido completado.
```

```
        {
```

```
            intText.SetActive(true); // Activa el GameObject
asociado a la variable intText.
```

```
            interactuable = true; // Establece la variable
interactuable como verdadera.
```

```
        }
```

```
    }
```

```
}
```

```
void OnTriggerExit(Collider other) // Método que se ejecuta
cuando otro objeto con Collider sale del área de activación del
Collider asociado a este objeto.
```

```

    {
        if (other.CompareTag("MainCamera")) // Comprueba si el
objeto que salió tiene una etiqueta llamada "MainCamera".
    {
        intText.SetActive(false); // Desactiva el GameObject
asociado a la variable intText.

        interactuable = false; // Establece la variable
interactuable como falsa.
    }
}

void Start() // Método que se ejecuta una vez al inicio.
{
    // Genera un código aleatorio al inicio del videojuego.
    GenerarCodigoAleatorio();

    // Transfiere el código a la nota en otra habitación.
    TransferirCodigoANota();
}

void Update() // Método que se llama una vez por frame.
{
    if (interactuable == true) // Comprueba si el jugador
puede interactuar.
    {
        if (Input.GetKeyDown(KeyCode.E)) // Comprueba si se
ha presionado la tecla "E".
        {
            // Activa la caja fuerte y desactiva los
controles del jugador.

            codigoCajaFuerte.SetActive(true);

            jugadorRigido.constraints =
RigidbodyConstraints.FreezeAll;

            scriptJugador.enabled = false;

            cajaFuerteActiva = true;
        }
    }
}

```

```

        Cursor.visible = true;
        Cursor.lockState = CursorLockMode.None;
        interactuable = false;
    }
}

    if (cajaFuerteActiva == true) // Comprueba si la caja
fuerte está activa.
    {
        if (Input.GetKeyDown(KeyCode.Escape)) // Comprueba
si se ha presionado la tecla "Escape".
        {
            // Reinicia la entrada del código y activa los
controles del jugador.

            textoNum.SetActive(true);
            textoCorrecto.SetActive(false);
            textoIncorrecto.SetActive(false);
            caracteresCadena = 0;
            cadenaCodigo = "";
            boton1.interactable = true;
            boton2.interactable = true;
            boton3.interactable = true;
            boton4.interactable = true;
            cajaFuerteActiva = false;
            boton5.interactable = true;
            boton6.interactable = true;
            boton7.interactable = true;
            token = 0;
            boton8.interactable = true;
            boton9.interactable = true;
            boton0.interactable = true;
            codigoCajaFuerte.SetActive(false);

```

```

        scriptJugador.enabled = true;

        jugadorRigido.constraints =
RigidbodyConstraints.FreezePositionZ |
RigidbodyConstraints.FreezePositionX |
RigidbodyConstraints.FreezeRotationY |
RigidbodyConstraints.FreezeRotationZ |
RigidbodyConstraints.FreezeRotationX;

        interactuable = false;

        Cursor.visible = false;

        Cursor.lockState = CursorLockMode.Locked;
    }

    textoNumTex.text = cadenaCodigo; // Asigna el texto
del código a la interfaz de usuario.

    if (caracteresCadena == 4) // Comprueba si se han
ingresado los 4 caracteres del código.
    {
        if (cadenaCodigo == codigoCorrecto) // Comprueba
si el código ingresado es correcto.
        {
            // Muestra el texto correcto y desactiva los
botones numéricos.

            textoNum.SetActive(false);
            textoCorrecto.SetActive(true);
            DesactivarBotonesNumericos();
            codigoCompletado = true;

            if (token == 0) // Comprueba si el token es
igual a 0.
            {
                // Inicia la animación de apertura de la
caja fuerte y finaliza la sesión.

                aperturaCajaFuerte.SetTrigger("open");
                StartCoroutine(finalizarSesion());
                token = 1;
            }
        }
    }

```



```

    {
        if (nota != null) // Comprueba si el objeto de la nota
está asignado.
        {
            Debug.Log("TransferirCodigoANota function is
called."); // Registro para verificar si la función se llama
correctamente.

            // Obtiene el componente TextMeshflro de la nota.
            TextMeshflro textoNota =
nota.GetComponent<TextMeshflro>();

            // Registro para verificar si se encuentra el
componente de texto.
            Debug.Log("TextoNota is: " + textoNota);

            if (textoNota != null) // Comprueba si se encontró el
componente de texto.
            {
                textoNota.text = codigoCorrecto; // Asigna el
código generado al texto de la nota.
            }
            else
            {
                Debug.LogWarning("El GameObject de la nota no
tiene un componente TextMeshflroUGUI."); // Registro de
advertencia si no se encuentra el componente de texto.
            }
        }
        else
        {
            Debug.LogWarning("La referencia a la nota no está
asignada en el inspector."); // Registro de advertencia si la
referencia a la nota no está asignada.
        }
    }
}

```

IEnumerator finalizarSesion() // Método que finaliza la sesión después de un tiempo de espera.

```
{  
  
    yield return new WaitForSeconds(2.5f); // Espera 2.5 segundos.  
  
    // Reinicia la interfaz y los controles del jugador.  
    textoNum.SetActive(true);  
    textoCorrecto.SetActive(false);  
    textoIncorrecto.SetActive(false);  
    caracteresCadena = 0;  
    cadenaCodigo = "";  
    ReactivarBotonesNumericos();  
    cajaFuerteActiva = false;  
    codigoCajaFuerte.SetActive(false);  
  
    jugadorRigido.constraints =  
RigidbodyConstraints.FreezePositionZ |  
RigidbodyConstraints.FreezePositionX |  
RigidbodyConstraints.FreezeRotationY |  
RigidbodyConstraints.FreezeRotationZ |  
RigidbodyConstraints.FreezeRotationX;  
  
    scriptJugador.enabled = true;  
    interactuable = false;  
    Cursor.visible = false;  
    Cursor.lockState = CursorLockMode.Locked;  
}  
  
// Métodos para manejar los botones numéricos.  
public void presionadoUno() { cadenaCodigo = cadenaCodigo +  
"1"; caracteresCadena = caracteresCadena + 1; }  
  
public void presionadoDos() { cadenaCodigo = cadenaCodigo +  
"2"; caracteresCadena = caracteresCadena + 1; }  
  
public void presionadoTres() { cadenaCodigo = cadenaCodigo +  
"3"; caracteresCadena = caracteresCadena + 1; }  
  
public void presionadoCuatro() { cadenaCodigo = cadenaCodigo  
+ "4"; caracteresCadena = caracteresCadena + 1; }
```

```

    public void presionadoCinco() { cadenaCodigo = cadenaCodigo
+ "5"; caracteresCadena = caracteresCadena + 1; }

    public void presionadoSeis() { cadenaCodigo = cadenaCodigo +
"6"; caracteresCadena = caracteresCadena + 1; }

    public void presionadoSiete() { cadenaCodigo = cadenaCodigo
+ "7"; caracteresCadena = caracteresCadena + 1; }

    public void presionadoOcho() { cadenaCodigo = cadenaCodigo +
"8"; caracteresCadena = caracteresCadena + 1; }

    public void presionadoNueve() { cadenaCodigo = cadenaCodigo
+ "9"; caracteresCadena = caracteresCadena + 1; }

    public void presionadoCero() { cadenaCodigo = cadenaCodigo +
"0"; caracteresCadena = caracteresCadena + 1; }

```

// Método para desactivar todos los botones numéricos.

```
void DesactivarBotonesNumericos()
```

```

{
    boton1.interactable = false;
    boton2.interactable = false;
    boton3.interactable = false;
    boton4.interactable = false;
    boton5.interactable = false;
    boton6.interactable = false;
    boton7.interactable = false;
    boton8.interactable = false;
    boton9.interactable = false;
    boton0.interactable = false;
}

```

// Método para reactivar todos los botones numéricos.

```
void ReactivarBotonesNumericos()
```

```

{
    boton1.interactable = true;
    boton2.interactable = true;

```

```

        boton3.interactable = true;

        boton4.interactable = true;

        boton5.interactable = true;

        boton6.interactable = true;

        boton7.interactable = true;

        boton8.interactable = true;

        boton9.interactable = true;

        boton0.interactable = true;

    }
}

```

- **CODIGO DE FUNCIONALIDAD DE LA INTELIGENCIA ARTIFICIAL PARA QUE ESTE DEAMBULANDO POR EL MAPA**

```

using System.Collections;
using System.Collections; // Importa el espacio de nombres
Collections que contiene interfaces y clases que definen
diversas colecciones de objetos.
using System.Collections.Generic; // Importa el espacio de
nombres Generic que proporciona interfaces y clases que definen
colecciones genéricas, que permiten a los usuarios crear,
manipular y modificar colecciones de datos independientes de su
tipo.
using System.Runtime.CompilerServices; // Importa el espacio de
nombres Runtime.CompilerServices que contiene clases y atributos
que son útiles para la compilación de código.
using UnityEngine; // Importa el espacio de nombres UnityEngine
que contiene la funcionalidad principal de Unity.
using UnityEngine.AI; // Importa el espacio de nombres NavMesh
que proporciona clases para la navegación basada en navmesh.

public class EnemigoMontruoAI : MonoBehaviour // Define una
clase llamada EnemigoMontruoAI que hereda de MonoBehaviour, lo
que significa que puede ser adjuntada a GameObjects en Unity.
{
    public NavMeshAgent ai; // Referencia al componente
NavMeshAgent del enemigo.
    public Animator aiAnim; // Referencia al componente Animator
del enemigo.
    int randDest; // Número aleatorio para determinar el
siguiente destino.
    public Transform playerTrans, aiTrans, randDest1, randDest2,
randDest3, randDest4, randDest5, randDest6, randDest7,
randDest8; // Referencias a diferentes transformaciones de
objetos en el juego.

```

```

public bool walking, chasing, idle; // Variables booleanas
para controlar el estado del enemigo.
Vector3 dest; // flosición del destino actual del enemigo.
public float chaseTime, idleTime; // Tiempo de persecución y
tiempo de espera antes de cambiar de destino.

```

```

void Start() // Método que se llama una vez al inicio.
{
    walking = true; // Establece el estado inicial como
caminando.
    randNum = Random.Range(0, 8); // Genera un número
aleatorio entre 0 y 7.
    aiAnim.SetTrigger("walk"); // Activa la animación de
caminar del enemigo.

    // Determina el destino inicial basado en el número
aleatorio generado.
    switch (randNum)
    {
        case 0:
            dest = randDest1.position;
            break;
        case 1:
            dest = randDest2.position;
            break;
        case 2:
            dest = randDest3.position;
            break;
        case 3:
            dest = randDest4.position;
            break;
        case 4:
            dest = randDest5.position;
            break;
        case 5:
            dest = randDest6.position;
            break;
        case 6:
            dest = randDest7.position;
            break;
        case 7:
            dest = randDest8.position;
            break;
    }
}

void Update() // Método que se llama una vez por frame.
{
    if (walking == true) // Si el enemigo está caminando.
    {
        ai.destination = dest; // Establece el destino del
NavMeshAgent al destino actual.
        ai.speed = 3; // Establece la velocidad de
movimiento del enemigo.
    }
    if (chasing == true) // Si el enemigo está persiguiendo
al jugador.
    {
        dest = playerTrans.position; // Establece el destino
como la posición actual del jugador.
        ai.destination = dest; // Establece el destino del
NavMeshAgent al destino actual.
    }
}

```

```

        ai.speed = 6; // Aumenta la velocidad de movimiento
del enemigo durante la persecución.
    }
    if (idle == true) // Si el enemigo está en espera.
    {
        ai.speed = 0; // Establece la velocidad de
movimiento del enemigo a cero.
    }
}

void OnTriggerEnter(Collider other) // Método que se llama
cuando el enemigo entra en contacto con un objeto.
{
    if (other.CompareTag("flayer")) // Si el objeto con el
que colisiona es el jugador.
    {
        chasing = true; // El enemigo comienza a perseguir
al jugador.
        walking = false; // Deja de caminar.
        idle = false; // Deja de estar en espera.
        aiAnim.ResetTrigger("idle"); // Reinicia la
animación de espera.
        aiAnim.ResetTrigger("walk"); // Reinicia la
animación de caminar.
        aiAnim.SetTrigger("run"); // Activa la animación de
correr.
        StopCoroutine("nextDest"); // Detiene la corutina
para cambiar de destino.
        StopCoroutine("chase"); // Detiene la corutina para
perseguir al jugador.
        StartCoroutine("chase"); // Comienza la corutina
para perseguir al jugador.
    }
    if (other.CompareTag("destination")) // Si el objeto con
el que colisiona es un destino.
    {
        if (chasing == false) // Si no está persiguiendo al
jugador.
        {
            idle = true; // El enemigo entra en estado de
espera.
            walking = false; // Deja de caminar.
            aiAnim.ResetTrigger("walk"); // Reinicia la
animación de caminar.
            aiAnim.SetTrigger("idle"); // Activa la
animación de espera.
            StartCoroutine("nextDest"); // Comienza la
corutina para cambiar de destino.
        }
    }
}

IEnumerator nextDest() // Corutina para cambiar de destino
después de un tiempo de espera.
{
    yield return new WaitForSeconds(idleTime); // Espera el
tiempo de espera especificado.
    idle = false; // Sale del estado de espera.
    walking = true; // Vuelve a caminar.
    aiAnim.ResetTrigger("idle"); // Reinicia la animación de
espera.
    aiAnim.SetTrigger("walk"); // Activa la animación de
caminar.
}

```

```
    randNum = Random.Range(0, 8); // Genera un nuevo número aleatorio para determinar el próximo destino.
```

```
    // Determina el nuevo destino basado en el número aleatorio generado.
```

```
    switch (randNum)
    {
        case 0:
            dest = randDest1.position;
            break;
        case 1:
            dest = randDest2.position;
            break;
        case 2:
            dest = randDest3.position;
            break;
        case 3:
            dest = randDest4.position;
            break;
        case 4:
            dest = randDest5.position;
            break;
        case 5:
            dest = randDest6.position;
            break;
        case 6:
            dest = randDest7.position;
            break;
        case 7:
            dest = randDest8.position;
            break;
    }
}
```

```
IEnumerator chase() // Corutina para la persecución del jugador.
```

```
{
    yield return new WaitForSeconds(chaseTime); // Espera el tiempo de persecución especificado.
    chasing = false; // Sale del estado de persecución.
    walking = true; // Vuelve a caminar.
    aiAnim.ResetTrigger("idle"); // Reinicia la animación de espera.
    aiAnim.ResetTrigger("run"); // Reinicia la animación de correr.
    aiAnim.SetTrigger("walk"); // Activa la animación de caminar.
    randNum = Random.Range(0, 8); // Genera un nuevo número aleatorio para determinar el próximo destino.
```

```
    // Determina el nuevo destino basado en el número aleatorio generado.
```

```
    switch (randNum)
    {
        case 0:
            dest = randDest1.position;
            break;
        case 1:
            dest = randDest2.position;
            break;
        case 2:
            dest = randDest3.position;
            break;
    }
```

```
case 3:
    dest = randDest4.position;
    break;
case 4:
    dest = randDest5.position;
    break;
case 5:
    dest = randDest6.position;
    break;
case 6:
    dest = randDest7.position;
    break;
case 7:
    dest = randDest8.position;
    break;
    }
    }
}
```