

**PROTOTIPO DE VIDEOJUEGO 3D DE *PUZZLE* PARA EL APRENDIZAJE DE
LOS FUNDAMENTOS DE PROGRAMACIÓN**

**JEFFERSON ROZO RAMIREZ
JHON ALEXIS GONZÁLEZ DURÁN**

**UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA
FACULTAD DE INGENIERÍA
INGENIERÍA DE SISTEMAS
BUCARAMANGA**

2023

TABLA DE CONTENIDOS

| | |
|---|----|
| INTRODUCCIÓN..... | 5 |
| 1. PROBLEMA DE INVESTIGACIÓN..... | 6 |
| 1.1 ANTECEDENTES DEL PROBLEMA..... | 6 |
| 1.2 PLANTEAMIENTO DEL PROBLEMA..... | 6 |
| 1.3 JUSTIFICACION..... | 6 |
| 1.4 PREGUNTA PROBLEMA | 7 |
| 2. OBJETIVOS..... | 8 |
| 2.1 OBJETIVO GENERAL..... | 8 |
| 2.2 OBJETIVOS ESPECÍFICOS | 8 |
| 2.3 TABLA DE PRODUCTOS | 8 |
| 3. ESTADO DEL ARTE | 11 |
| 4. MARCO DE REFERENCIA..... | 16 |
| 4.1 MARCO TEÓRICO | 16 |
| 4.2 MARCO METODOLOGICO | 19 |
| 4.2.1 METODOLOGIA DE DESARROLLO DE SOFTWARE | 21 |
| 5. PRESUPUESTO | 23 |
| 6. CRONOGRAMA..... | 25 |
| 7. BIBLIOGRAFIA..... | 26 |

LISTADO DE FIGURAS

| | |
|--|----|
| Figura 1: Juego While True: learn()..... | 10 |
| Figura 2: Juego CodeCombat..... | 11 |
| Figura 3: Juego Hack 'n' Slash..... | 12 |
| Figura 4: Logo de Unity..... | 19 |
| Figura 5: Pistola de gravedad desarrollada para el juego en blender..... | 19 |
| Figura 6: Logo de LMMS..... | 20 |
| Figura 7: Cronograma de actividades..... | 24 |

LISTADO DE TABLAS

| | |
|--|----|
| Tabla 1: Tabla de productos | 8 |
| Tabla 2: Presupuesto recursos humano..... | 23 |
| Tabla 3: Presupuesto software utilizado..... | 23 |
| Tabla 4: Presupuesto hardware utilizado..... | 24 |
| Tabla 5: Presupuesto servicios públicos..... | 24 |

INTRODUCCIÓN

En la actualidad, el conocimiento en programación se ha convertido en un requisito esencial para la mayoría de las empresas y organizaciones. Por ello, es importante que las personas interesadas en este ámbito desarrollen habilidades y conocimientos en esta área desde una edad temprana.

En el siguiente documento se enunciará la propuesta de una herramienta valiosa para enseñar y aprender programación, con el fin de despertar el interés en las personas mucho antes del inicio de sus estudios profesionales en un programa de Ingeniería de Sistemas. Se plantea un prototipo de videojuego de género puzzle que tenga un impacto beneficioso en el aprendizaje de los estudiante y futuros profesionales del área de la programación, fomentando su crecimiento y desarrollo en este campo en constante evolución.

El prototipo de videojuego de género puzzle está diseñado para ser accesible y entretenido para cualquier persona, sin tener en cuenta su nivel de conocimiento en programación. Con este videojuego, se busca no solo enseñar a los jugadores conceptos básicos de programación, sino también despertar su interés y curiosidad por esta área del conocimiento.

Para ello, se explorarán diversas metodologías y herramientas de desarrollo de videojuegos, así como también se analizarán los fundamentos esenciales de la lógica de programación y su aplicación en el diseño de videojuegos. En definitiva, el presente proyecto tiene como finalidad contribuir al aprendizaje de la programación de manera innovadora y atractiva para los estudiantes y futuros profesionales del área.

1. PROBLEMA DE INVESTIGACIÓN

1.1 ANTECEDENTES DEL PROBLEMA

- El plan de estudio de los colegios para el nivel de bachillerato no contempla la programación entre sus temáticas de enseñanza.
- La gran cantidad de información que se puede hallar en internet y lo confuso que puede ser encontrar un punto de partida puede resultar en una pérdida de interés para aquellas personas sin previa experiencia que buscaban aprender a programar.
- Para muchas personas puede resultar complicado el empezar a desarrollar una nueva habilidad, como lo es la programación, por lo que es necesario crear opciones sencillas que no hagan que la gente pierda el interés por aprender.
- Un estudio realizado por ACOFI indica que la importancia de la programación ha aumentado significativamente a lo largo de estos últimos años, sin embargo, para muchas personas sin experiencia, el aprendizaje de esta disciplina puede resultar intimidante y agobiante, debido a la enorme cantidad de información existente en línea.

1.2 PLANTEAMIENTO DEL PROBLEMA

La programación es una disciplina que cuenta con una temática muy extensa. Es por esto que, para aquellos que carecen de algún tipo de experiencia en el tema, el aprendizaje puede resultar abrumador y exigente, lo que a su vez puede generar frustración y disminuir su interés en el campo. Es por esto que es esencial contar con recursos que apoyen a los estudiantes en los conceptos **esenciales** de la programación y les permitan construir una base sólida para avanzar hacia una carrera profesional o un curso especializado en esta área.

1.3 JUSTIFICACION

En la actualidad, la programación se ha convertido en un campo de creciente relevancia en la sociedad, dado que muchas profesiones y trabajos requieren

habilidades en esta área. Sin embargo, a día de hoy no se ningún tipo de interés por parte de las instituciones de educación básica y media en instruir a sus estudiantes en esta disciplina, cosa que se ve reflejada en el plan de estudio para la etapa educativa del bachillerato. (Ministerio de Educación de Colombia, 1964)

Por otra parte, aprender programación puede resultar una tarea difícil para algunas personas, debido a la complejidad y abstracción de los conceptos técnicos que se manejan, así como a la necesidad de comprender la lógica y las matemáticas que subyacen en ellos. (Fuentes-Rosado & Moo-Medina, 2017)

Por lo tanto, en el presente trabajo se propone el desarrollo de un prototipo de videojuego *puzzle* que permita a los jugadores aprender los fundamentos de la programación de una manera accesible y divertida. La idea es utilizar un enfoque lúdico e interactivo para que el aprendizaje de la programación resulte más entretenido y menos tedioso.

Una ventaja significativa de este videojuego es que puede estimular el interés por la programación desde una etapa temprana, antes de que los estudiantes inicien sus estudios en ingeniería de sistemas. Además, el enfoque *puzzle* del videojuego puede ayudar a los jugadores a comprender los conceptos de manera más concreta y visual, lo que a su vez puede resultar beneficioso para su aprendizaje. En resumen, el desarrollo de este prototipo de videojuego puede ser una herramienta valiosa para hacer más accesible el aprendizaje de la programación y fomentar el interés en esta área de conocimiento.

1.4 PREGUNTA PROBLEMA

¿Puede un videojuego cumplir con la función de enseñar a las personas los fundamentos de programación?

2. OBJETIVOS

2.1 OBJETIVO GENERAL

Desarrollar un prototipo de videojuego de género *puzzle* en 3D que sirva como herramienta educativa que facilite el aprendizaje de lógica en los fundamentos de la programación.

2.2 OBJETIVOS ESPECÍFICOS

- Investigar sobre elementos y métodos para enseñanza de programación.
- Diseñar el documento de diseño de videojuego GDD (*Game Desing Document*).
- Construir el videojuego en el motor de desarrollo Unity Engine.
- Aplicar pruebas de usuario que permitan validen el correcto funcionamiento del videojuego.

2.3 TABLA DE PRODUCTOS

Tabla 1: Tabla de productos

| OBJETIVOS | ACTIVIDADES | PRODUCTOS |
|--|---|--|
| Investigar sobre elementos y métodos para enseñanza de programación. | <ul style="list-style-type: none">• Identificación de elementos de videojuegos que sean aplicables a la enseñanza.• Investigación sobre los temas relevantes que se deseen incluir en el videojuego. | <ul style="list-style-type: none">○ Documento con la recopilación de temas y su interacción con elementos de videojuego. |

| | | |
|---|--|--|
| <p>Diseñar el documento de diseño de videojuego GDD (Game Desing Document).</p> | <ul style="list-style-type: none"> • Identificación las mecánicas de juego que se desean incluir, como los controles, la física, los objetivos, los niveles, etc. • Elaboración de la historia del juego que involucre a personajes y elementos del mismo. • Definición diseño del estilo visual del juego, la paleta de colores, los gráficos, la OST(Música) y los SFX (Efectos de sonido) que se desean utilizar en el juego. • Elaboración del diseño de mapa del juego identificando los niveles, la progresión, los desafíos y los elementos relevantes del mismo. | <ul style="list-style-type: none"> ○ Documento de diseño de juego (Game Desing Document) |
| <p>Construir el videojuego en el motor de desarrollo Unity Engine.</p> | <ul style="list-style-type: none"> • Elaboración de los elementos del juego: los personajes, los objetos, los escenarios y cualquier otro elemento que se desee incluir en el juego. • Definición de lógicas de juego, interacción entre los personajes y objetos, físicas y mecánicas del juego. • Integración los elementos visuales y sonoros, como los gráficos, la OST y los SFX, en el juego. | <ul style="list-style-type: none"> ○ Archivos ejecutables que contengan diferentes versiones del juego para probar y ajustar. |

| | | |
|---|---|---|
| <p>Aplicar pruebas de usuario que permitan validen el correcto funcionamiento del videojuego.</p> | <ul style="list-style-type: none"> • Definición del plan de pruebas, como los niveles del juego o las mecánicas de juego específicas que se probaran. • Realización de pruebas con base en el plan de pruebas planteado. • Análisis de resultado y aplicación de correcciones. | <ul style="list-style-type: none"> ○ Documento que contenga los resultados de las pruebas de usuario, incluyendo información como la descripción del escenario de prueba, los errores encontrados y las sugerencias para mejorar el juego. |
|---|---|---|

Fuente: Elaboración propia.

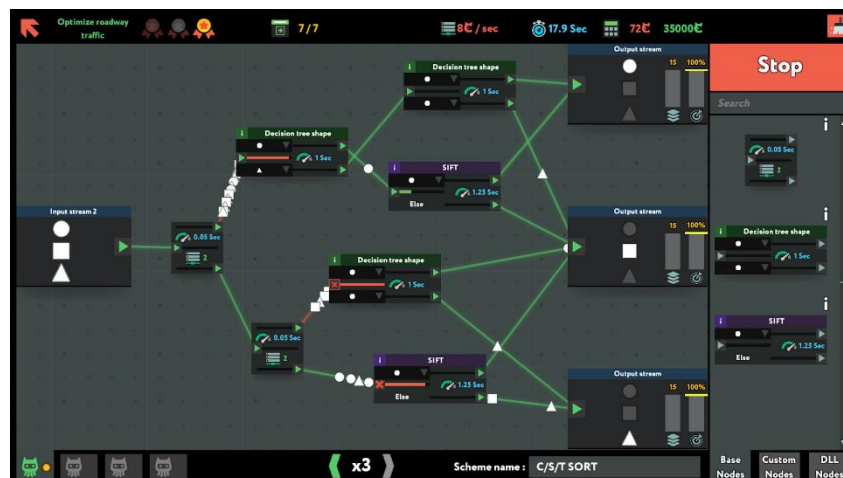
3. ESTADO DEL ARTE

3.1.1 while True: learn()

Resumen: While True: learn() es un juego orientado a la enseñanza de sistemas complejos de la programación, tales como el aprendizaje automatizado, redes neuronales e inteligencia artificial. Su mecánica consiste en diferentes niveles con objetivos que deben ser cumplidos por medio de la lógica de la programación. (Oleg Chumakov, 2019)

Aporte: Esta propuesta de juego puede brindar una valiosa inspiración para la elaboración y la construcción de un prototipo de videojuego. La aplicación de la mecánica de niveles con objetivos que requieren la aplicación de la lógica de programación puede ser una forma efectiva de crear desafíos dentro de tu propio juego. Además, si te enfocas en sistemas de programación más avanzados, como el aprendizaje automático, las redes neuronales y la inteligencia artificial, podrás diseñar niveles que enseñen conceptos más complejos de programación. En general, la implementación de estas mecánicas en tu propio juego puede ser una estrategia valiosa para fomentar el aprendizaje interactivo y práctico de la programación

Figura 1: Juego While True: learn()



Fuente: <https://luden.io/wtl/>

3.1.2 CodeCombat

Resumen: CodeCombat es un juego de aventuras 2D que cuenta con un editor de código, en el cual, los jugadores programarán los movimientos que desean que el personaje realice mientras exploran el mundo de fantasía en el que se ambienta el juego. (CodeCombat, 2013)

Aporte: Se podría contribuir al proyecto mediante la inclusión de una herramienta de programación interactiva. Al permitir que los jugadores programen los movimientos del personaje en el juego de aventuras, se les brinda una oportunidad valiosa para practicar programación de manera práctica e interactiva en un entorno lúdico. Esto puede ser especialmente beneficioso para aquellos que desean aprender programación de manera más accesible y amigable para los principiantes.

Figura 2: Juego CodeCombat



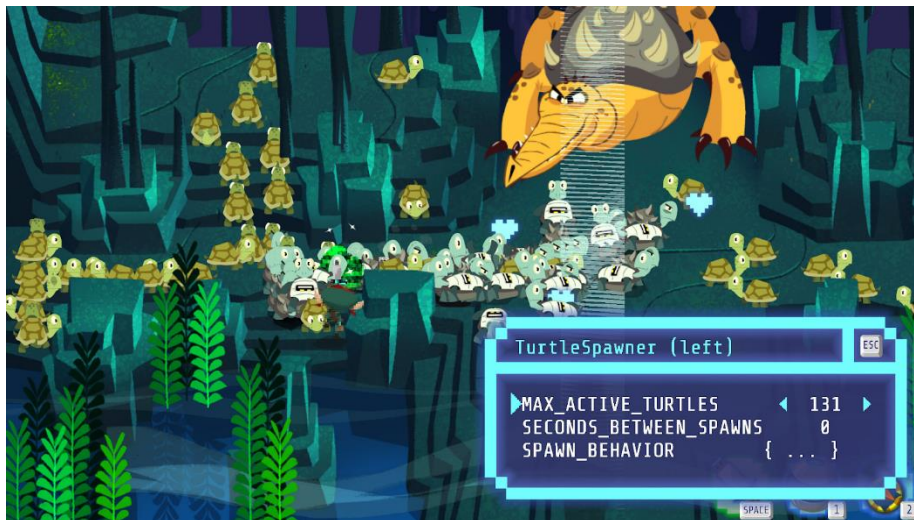
Fuente: <https://codecombat.com>

3.1.3 Hack 'n' Slash

Resumen: Hack 'n' Slash es un juego de acción y rompecabezas que consiste en la "reprogramación" de las mecánicas de los objetos, secuestrar variables globales, o incluso reescribir el código del juego, dando al jugador la sensación de que no solo controla el juego, sino que también está creando su lógica. (Double Fine, 2014)

Aporte: El aporte más destacado es la capacidad del jugador para "reprogramar" las mecánicas de los objetos, secuestrar variables globales y reescribir el código del juego puede ser una herramienta poderosa para enseñar conceptos de programación. Esta mecánica permite al jugador experimentar directamente cómo los cambios en el código afectan al juego, lo que puede ser una forma efectiva de enseñar cómo funciona la programación. Además, la sensación de control que proporciona esta mecánica podría aumentar la motivación y despertar el interés de los jugadores por aprender a programar.

Figura 3: Juego Hack 'n' Slash



Fuente: <https://www.doublefine.com/games/hack-n-slash>

3.1.4 Artículo Dificultades de aprender a programar

Resumen: El artículo se centra en los problemas que confrontan los estudiantes al aprender programación, especialmente en la comprensión de conceptos abstractos y matemáticos. Propone una metodología basada en la práctica y la visualización de los conceptos para mejorar la comprensión de los estudiantes. (Fuentes-Rosado & Moo-Medina, 2017)

Aporte: Esta metodología podría ser útil para el desarrollo del videojuego de *puzzle* 3D para enseñar programación, ya que aborda las dificultades que enfrentan los estudiantes al aprender programación y hace que el aprendizaje sea más accesible y atractivo mediante una experiencia práctica y visual.

3.1.5 Metodología para el diseño de videojuegos educativos sobre una arquitectura para el análisis del aprendizaje colaborativo

Resumen: La metodología propuesta por Natalia Padilla Zea se centra en el elaboración y desarrollo de videojuegos educativos que fomenten el aprendizaje colaborativo y que permitan el análisis del aprendizaje colaborativo en ellos, utilizando una serie de pasos que incluyen la identificación de objetivos educativos, la selección de herramientas y tecnologías, el diseño de la estructura y los aspectos pedagógicos del videojuego, la integración de elementos de aprendizaje colaborativo, la evaluación de la efectividad del videojuego educativo y el análisis del aprendizaje colaborativo. (Por et al., 2011)

Aporte: la metodología propuesta por Natalia Padilla Zea se enfoca en el diseño y construcción de videojuegos educativos que fomenten el aprendizaje colaborativo y que permitan el análisis del aprendizaje colaborativo en ellos, utilizando una serie de pasos que incluyen la identificación de objetivos educativos, la selección de herramientas y tecnologías, el diseño de la estructura y los aspectos pedagógicos del videojuego, la integración de elementos de aprendizaje colaborativo, la evaluación de la efectividad del videojuego educativo y el análisis del aprendizaje colaborativo.

3.1.6 Videojuego para el aprendizaje de lógica de programación

Resumen: El artículo presenta la importancia de enseñar programación en edades tempranas y cómo los videojuegos pueden ser una herramienta complementaria para la enseñanza de la lógica computacional. El artículo sugiere el desarrollo de un videojuego móvil con el propósito de potenciar la comprensión de la lógica computacional en los estudiantes de bachillerato. Se incluye una revisión de estudios previos sobre la enseñanza de la programación y videojuegos y presenta el videojuego desarrollado para finalmente presentar las conclusiones al respecto. (Bermón Angarita et al., 2021)

Aporte: El artículo puede servir como una fuente de apoyo para el diseño del prototipo. Se pueden tomar en cuenta las recomendaciones del artículo en cuanto al uso de los videojuegos para la enseñanza de la lógica computacional y la validación del videojuego desarrollado. Además, el artículo destaca la importancia de enseñar programación desde edades tempranas, por lo que se puede considerar el público objetivo del videojuego propuesto y cómo este puede ser atractivo para estudiantes de básica secundaria.

4. MARCO DE REFERENCIA

4.1 MARCO TEÓRICO

Conceptos generales:

- **Gamificación:** También conocida como “ludificación”, según la Real Academia Española se define cómo aplicar técnicas lúdicas a entornos o actividades no recreativos, con el objetivo de fomentar la participación y facilitar el proceso de aprendizaje de dichas actividades.
- **Teoría de aprendizaje:** Es un conjunto de ideas y enfoques que buscan explicar cómo las personas adquieren nuevos conocimientos, alegando que es importante que para se produzca un aprendizaje significativo se deben cumplir dos condiciones: La disposición a aprender y la presentación de un material potencialmente significativo. Es importante saber que el aprendizaje no se debe producir de forma mecánica, sino de una manera que involucre al aprendiz y sus conocimientos previos aplicándolos a estos nuevos aprendizajes. (Richard Matienzo, 2020)
- **Aprendizaje receptivo:** Es un método de aprendizaje donde un instructor le transmite a su aprendiz la información ya elaborada y “procesada”, por lo cual, la única función de este último es retener o memorizar dicha información. Es un método de aprendizaje ampliamente utilizado en la educación tradicional. (Laura Ruiz Mitjana, 2019)
- **Fundamentos de programación:** Son los conceptos y principios fundamentales que subyacen del desarrollo de software, cuyo conocimiento es fundamental para comprender la lógica básica de la programación y aplicarla en proyectos de desarrollo de software. Algunos de los conceptos que aborda esta materia son: variables, tipos de datos, controladores, estructuras de control, funciones y estructuras de datos regulares.
- **Algoritmos:** Se define como una secuencia de lógica de instrucciones que le indican a un programa como debe resolver un problema o realizar una tarea.

Conceptos de videojuegos

- **Puzzle:** Traducido al español como “puzle” o “rompecabezas” se trata de un problema o acertijo de difícil solución.
- **Videojuego Puzzle:** También conocidos como “videojuegos de lógica” es un género de videojuegos cuya mecánica principal consiste en el uso de desafíos lógicos que ponen a prueba la habilidad mental del jugador, estos desafíos pueden ir desde problemas de lógica o matemáticos, hasta incluso el azar.
- **Juegos 3D:** Son videojuegos que, con ayuda del uso de gráficos poligonales, crean un mundo donde se simulan las tres dimensiones que existen en el espacio (altura, anchura y profundidad), a diferencia de los juegos 2D que carecen de esta última, los videojuegos 3D no son un género, sino una decisión de diseño.
- **Juego Single-Player:** Son videojuegos que están diseñados para una sola persona, sin la posibilidad de añadir a otro participante.

Motores de desarrollo

- **Unity:** Es un motor de desarrollo creado por Unity Technologies, este software es muy conocido dentro de la comunidad de desarrollo de videojuegos independientes debido a la versatilidad del motor a la hora de crear videojuegos y otro tipo de contenido multimedia. Unity utiliza principalmente el lenguaje de programación C#. (*Plataforma de Desarrollo En Tiempo Real de Unity | Motor de VR, AR, 3D y 2D*, n.d.)
- **Unreal Engine:** Es un motor de desarrollo creado por la empresa Epic Games que utilizando programación con C++ y Blueprint. (*La Herramienta de Creación 3D En Tiempo Real Más Potente - Unreal Engine*, n.d.)

Lenguajes de programación

- **C++:** Es un lenguaje de programación de propósito general basado en el lenguaje C. Creado en la década de 1980 por Bjarne Stroustrup como una extensión de C para agregar características como la programación orientada a objetos, mejoras de eficiencia y expresividad del código. Actualmente tiene varios usos, tales como el desarrollo de sistemas operativos, motores de juegos, aplicaciones de escritorio, aplicaciones de baja latencia y alto rendimiento, controladores de dispositivos, entre otros.
- **C#:** Es un lenguaje de programación multiparadigma que combina elementos de C++ y Java, desarrollado y estandarizado por Microsoft utilizado principalmente para el desarrollo de software en el entorno de Windows. Este lenguaje se ha convertido en
- **Blueprint:** Es una forma de programación visual donde un usuario puede crear diferentes funciones y elementos interactivos sin necesidad de tener grandes conocimientos en programación, facilitando así el proceso.

Software de arte:

- **Blender:** Es un software gratuito dedicado al modelado, esculpido, renderizado y animación en 3d, también cuenta con la función de animación en 2d. (Blender.Org - Home of the Blender Project - Free and Open 3D Creation Software, n.d.)
- **Clip Studio Paint:** Es un software de ilustración para Mac OS X y Microsoft Windows desarrollado por Celsys de arte digital, a diferencia de otros programas como Photoshop, Clip Studio Paint no utiliza gráficos vectorizados en su dibujo, pues funcionalidad está más orientada a la creación de ilustraciones y comics digitales. (Clip Studio Paint - La Aplicación de Dibujo y Pintura Para Artistas, n.d.)

Software de sonido

- **LMMS:** Es un software de creación musical, de código abierto y multiplataforma que permite la composición, edición y creación de música.
- **Audacity:** Es un software de edición y grabación de audio de código abierto y gratuito que permite modificar audio, aplicar efectos y mezclar pistas.

Metodologías de desarrollo

- **Metodología Scrum:** Scrum es una metodología de trabajo ágil para el desarrollo de software. Se centra en un proceso iterativo e incremental para la gestión de proyectos, donde el equipo de desarrollo trabaja en sprints, ciclos de trabajo cortos y enfocados en un conjunto de objetivos específicos. (Trigás Gallego, 2012)
- **Metodología de desarrollo de prototipado rápido:** El prototipado rápido es una técnica que se utiliza en el desarrollo de productos para crear prototipos rápidamente y de manera económica. Los prototipos son modelos de prueba que se utilizan para probar la funcionalidad y el rendimiento de un producto antes de su producción final. (Federico Del Giorgio Solfa & Sofía Lara Marozzi, 2020)

4.2 MARCO METODOLOGICO

Se plantea utilizar una metodología de aprendizaje receptivo para el desarrollo del proyecto. En este enfoque, los jugadores del videojuego interiorizarán la información presentada en el juego para aplicarla en la resolución de puzzles más adelante. Será importante considerar la complejidad de la información proporcionada, ya que los temas de fundamentos de programación irán aumentando en complejidad progresivamente.

Se decidió utilizar el motor de juego Unity para el desarrollo del proyecto, debido a su flexibilidad y a la gran cantidad de documentación disponible, lo que facilita la creación del juego ya que es una herramienta ya usada. Si bien se consideraron

otros motores de desarrollo, se descartaron debido al tiempo que se requeriría para familiarizarse con ellos y su lenguaje de programación.

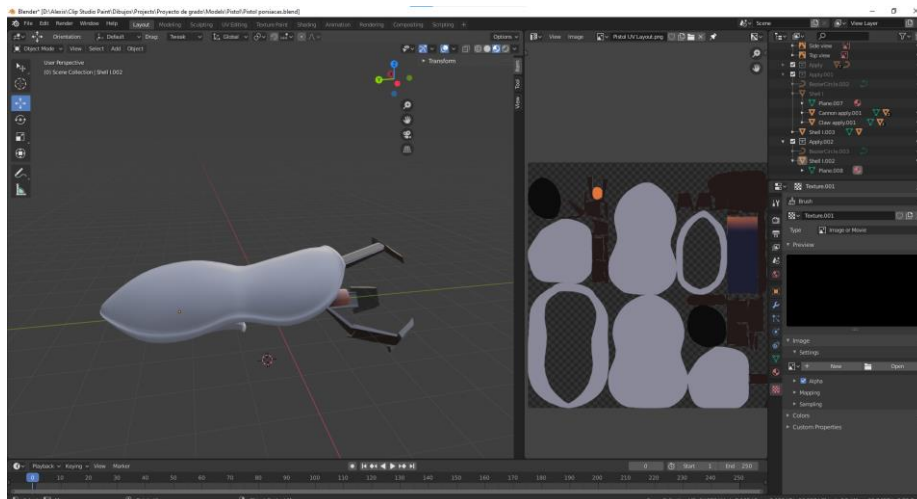
Figura 4: Logo de Unity



Fuente: <https://unity.com/>

Para la de modelos 3D elegimos Blender por ser altamente versátil y además gratuita. Blender una opción sólida para el desarrollo de videojuegos en 3D. Además de ser un software libre, Blender cuenta con una gran comunidad de usuarios activos, una integración suave con otras herramientas de desarrollo de videojuegos como Unity y actualizaciones frecuentes. Todo esto puede reducir el tiempo y la complejidad del flujo de trabajo, permitir ahorrar costos en herramientas de software y mantenerse actualizado con las últimas tendencias y tecnologías en el desarrollo de videojuegos.

Figura 5: Pistola de gravedad desarrollada para el juego en blender



Fuente: Modelo Propio en blender

Para la composición musical se decidió usar LMMS, ya que puede ser una opción sólida para la creación de música en videojuegos. Una de las ventajas de LMMS es que es gratuito, lo que puede ser especialmente útil para equipos de desarrollo con presupuestos limitados como el nuestro. Además, LMMS cuenta con una interfaz de usuario intuitiva que facilita la creación de música, incluso para aquellos que no tienen mucha experiencia en producción musical. Otra ventaja es que LMMS cuenta con una amplia variedad de herramientas de composición, como sintetizadores, cajas de ritmos, *samplers* y efectos de audio, que permiten crear una amplia gama de estilos musicales. Además, LMMS puede ser utilizado en conjunto con otros programas de producción musical, lo que permite una integración suave entre las herramientas utilizadas en el desarrollo del juego.

Figura 6: Logo de LMMS



Fuente: <https://lms.io/lsp/>

4.2.1 METODOLOGIA DE DESARROLLO DE SOFTWARE

La metodología de desarrollo de prototipado rápido es una técnica utilizada en la ingeniería de software que se enfoca en la creación de prototipos rápidos y funcionales para obtener retroalimentación temprana de los usuarios y clientes. Esta técnica permite a los desarrolladores obtener información valiosa sobre los requisitos del proyecto y ajustar el enfoque del desarrollo de manera eficiente.

Si se desea desarrollar un prototipo de videojuego 3D de *puzzle* para el aprendizaje de los fundamentos de programación, la metodología de prototipado rápido es una excelente opción. Esta técnica permite crear rápidamente prototipos funcionales que pueden ser probados por los usuarios y que pueden proporcionar

retroalimentación sobre las características y funcionalidades necesarias para el videojuego. De igual forma el desarrollo de un videojuego puede ser un proceso muy complejo y costoso. La metodología de prototipado rápido ayuda a reducir los costos y el tiempo dedicado a la producción del videojuego, ya que se enfoca en la creación de prototipos rápidos que pueden ser mejorados a través del ciclo de retroalimentación. Esto también ayuda a evitar problemas potenciales que podrían surgir en una fase posterior del proceso de desarrollo.

Ventaja:

- Retroalimentación temprana de los usuarios y clientes.
- Ayuda a reducir los costos y el tiempo dedicado al desarrollo de software.
- Permite ajustar el enfoque del desarrollo de manera eficiente.
- Facilita la toma de decisiones en el proceso de desarrollo.

Desventajas:

- Es difícil mantener la documentación actualizada durante el proceso iterativo de desarrollo.

5. PRESUPUESTO

Tabla 2: Presupuesto recursos humano

| Recursos humanos | | | | |
|------------------|-------------------|----------------|----------------|---------------|
| Personal | Ítem | Cantidad (hrs) | Valor unitario | Total |
| Desarrollador 1 | Tiempo de trabajo | \$ 320 | \$ 20,000 | \$ 6,400,000 |
| Desarrollador 2 | Tiempo de trabajo | \$ 320 | \$ 20,000 | \$ 6,400,000 |
| Director | Tiempo de trabajo | \$ 32 | \$ 20,000 | \$ 640,000 |
| Coordinador | Tiempo de trabajo | \$ 16 | \$ 20,000 | \$ 320,000 |
| Total: | | | | \$ 13,760,000 |

Fuente: Elaboración propia.

Tabla 3: Presupuesto software utilizado

| Software utilizado | | |
|--------------------|--------------------------|-------------|
| Nombre | Ítem | Valor (COP) |
| Unity Engine | Uso del software | \$ - |
| Clip Studio Paint | Licencia de un solo pago | \$ 226,352 |
| Blender | Uso del software | \$ - |
| Coordinador | Uso del software | \$ - |
| Total: | | \$ 226,352 |

Fuente: Elaboración propia.

Tabla 4: Presupuesto hardware utilizado

| Hardware utilizado | | | | |
|--------------------|------------------|------------------|----------------------|--------------|
| Ítem | Valor del activo | Vida útil (años) | Depreciación mensual | Total |
| Computador 1 | \$ 2,000,000 | 5 | \$ 33,333 | \$ 1,866,667 |
| Computador 2 | \$ 4,000,000 | 5 | \$ 66,667 | \$ 3,733,333 |
| Total: | | | | \$ 5,600,000 |

Fuente: Elaboración propia.

Tabla 5: Presupuesto servicios públicos

| Servicios públicos | | |
|--------------------------|---------------|--------------|
| Servicio de electricidad | | |
| Personal | Valor mensual | Sumatoria |
| Desarrollador 1 | \$ 360,000 | \$ 1,440,000 |
| Desarrollador 2 | \$ 90,000 | \$ 360,000 |
| Servicio de internet | | |
| Personal | Valor mensual | Sumatoria |
| Desarrollador 1 | \$ 70,000 | \$ 280,000 |
| Desarrollador 2 | \$ 105,000 | \$ 420,000 |
| Total: | | \$ 2,460,000 |

Fuente: Elaboración propia.

6. CRONOGRAMA

Figura 7: Cronograma de actividades



Fuente: Elaboración propia

7. BIBLIOGRAFIA

- Bermón Angarita, L., Prieto Taborda, A., Escobar Márquez, J. D., David, J., & Díaz, V. (2021). Videojuego para el aprendizaje de lógica de programación. *Revista Educación En Ingeniería*, 16(31), 46–56. <https://doi.org/10.26507/REI.V16N31.1141>
- blender.org - Home of the Blender project - Free and Open 3D Creation Software*. (n.d.). Retrieved May 7, 2023, from <https://www.blender.org/>
- Clip Studio Paint - La aplicación de dibujo y pintura para artistas*. (n.d.). Retrieved May 7, 2023, from <https://www.clipstudio.net/es/>
- CodeCombat. (2013). *CodeCombat - Juegos de Programación para aprender Python y JavaScript* | CodeCombat. <https://codecombat.com/>
- Double Fine. (2014). *Hack “n” Slash* | Double Fine Productions. <https://www.doublefine.com/games/hack-n-slash>
- Federico Del Giorgio Solfa, & Sofía Lara Marozzi. (2020, December). *Diseño y desarrollo de productos en base a prototipado rápido*. <https://www.aacademica.org/del.giorgio.solfa/174.pdf>
- Fuentes-Rosado, J. I., & Moo-Medina, M. (2017). Dificultades de aprender a programar. *Revista Educación En Ingeniería*, 12(24), 76–82. <https://doi.org/10.26507/REI.V12N24.728>
- La herramienta de creación 3D en tiempo real más potente - Unreal Engine*. (n.d.). Retrieved May 7, 2023, from <https://www.unrealengine.com/es-ES>
- Laura Ruiz Mitjana. (2019). *Aprendizaje receptivo: características y usos en la educación*. <https://psicologiaymente.com/psicologia/aprendizaje-receptivo>
- Ministerio de Educación de Colombia. (1964). *DECRETA: La enseñanza Media*.
- Oleg Chumakov. (2019). *while True: learn() Release or The Story About 1 Year of Weekly Updates, Machine Learning and Cats* | by Oleg Chumakov | Luden.io. <https://blog.luden.io/while-true-learn-release-or-the-story-about-1-year-of-weekly-updates-machine-learning-and-cats-4ad1884e5d8e>

Plataforma de desarrollo en tiempo real de Unity | Motor de VR, AR, 3D y 2D. (n.d.).

Retrieved May 7, 2023, from <https://unity.com/es>

Por, P., Padilla, N., Dirigida, Z., Gutiérrez Vela, F. L., & Medina, N. M. (2011).

Metodología para el diseño de videojuegos educativos sobre una arquitectura para el análisis del aprendizaje colaborativo. 1–407.

<https://digibug.ugr.es/handle/10481/19440>

Richard Matienzo. (2020). *Vista de Evolución de la teoría del aprendizaje significativo y su aplicación en la educación superior | Dialektika: Revista depppppl Investigación Filosófica y Teoría Social.*

<https://journal.dialektika.org/ojs/index.php/logos/article/view/15/14>

Trigás Gallego, M. (2012). *Metodología Scrum.* <http://hdl.handle.net/10609/17885>