



Videojuego serio para ejercitar el inglés

Documento de Tesis 5to año, Ingeniería Informática

Autor: Frank David Rico Rodríguez

Tutor: Jose Enrique Diaz Ramoz

UNIVERSIDAD DE MATANZAS
Facultad de Ciencias Técnicas
Departamento de Informática
2019

Pensamientos

“Los hombres crecen, crecen físicamente de una manera visible crecen, cuando aprenden algo, cuando entran a poseer algo, y cuando han hecho algún bien.”

José Martí.

“...La educación es como un árbol: se siembra una semilla y se abre en muchas ramas. Sea la gratitud del pueblo que se educa árbol protector, en las tempestades y las lluvias, de los hombres que hoy les hacen tanto bien. Hombres recogerá quien siembre escuelas.”

José Martí.

Dedicatoria

El éxito solo es posible gracias al esfuerzo de muchas personas que día a día comparten sus experiencias, a todas ellas dedico este trabajo de diploma, en especial:

A mi familia por su apoyo incondicional, por estar presente en este paso que doy en mi vida y por enseñarme que la perseverancia es parte del triunfo.

A mis profesores que me enseñaron a esforzarme y a crecer siempre un poco más en el estudio y como persona, por su preocupación y comprensión.

A todas las personas que se han cruzado en mi camino aportándome todo a cambio de nada, personas que te enriquecen el alma.

Agradecimientos

Agradezco a mi familia por brindarme su apoyo incondicional, siempre estuvieron conmigo en todo momento.

A mi tutor Jose Enrique por haberme ayudado tanto y siempre apoyarme en mis decisiones, por hacer ese espacio y lograr guiarme a través de toda esta tesis.

A Walfredo por aconsejarme siempre que me hizo falta.

A Noel porque a través de sus consejos esta tesis ha podido realizarse de una manera correcta.

Al colectivo de profesores y el jurado que están para brindarme apoyo y hacer de mí un mejor ingeniero.

A mi novia Gabriela por todo su apoyo, comprensión y esfuerzo incansable en lograr que todo este proyecto se realizara.

A Osniel por haberme ayudado con su opinión y su punto de vista, que siempre me hizo falta.

A mi religión que en ella guardé mis esperanzas.

A todos los que han intervenido en la realización de esta tesis, muchas gracias.

Declaración de autoría

Yo, Frank David Rico Rodríguez, declaro que soy el único autor de este trabajo y autorizo a la Universidad de Matanzas, y en especial, a la Facultad de Ciencias Técnicas, a que hagan el uso que estimen pertinente de él.

Para que así conste, firmo la presente a los _____ días del mes de junio del 2019.

Firma del autor

Frank David Rico Rodríguez

Firma del tutor

Jose Enrique Diaz Ramoz

Opinión del tutor sobre el trabajo diploma.

En esta tesis se ha trabajado con mucho esfuerzo y se han alcanzado los resultados propuestos, hemos podido enfrentar numerosas adversidades pero fuimos capaces de seguir avanzando con ánimos.

Los estudios realizados y la recopilación de información han sido significativos, las herramientas utilizadas han probado ser útiles pero sobre todo eficientes a la hora de realizar sus objetivos.

La finalidad de este proyecto cumple las expectativas, y se espera que se continúe su desarrollo en un futuro, poseer un software con la capacidad de ejercitar el idioma a temprana edad puede ser un recurso muy útil para nuestra sociedad y pienso que aportaría grandes beneficios para todos.

Resumen

En la actualidad muchos de los estudios realizados han demostrado que la mejor etapa para aprender un idioma nuevo es en la niñez (Centro de Formación Nathalie, 2019), pero a la mayoría de los niños y niñas les motiva muy poco estudiar algo pesado y poco entretenido, por ello se han realizado numerosos esfuerzos por lograr atraer la atención con didáctica, cuentos, videos e incluso juegos; y de este último es del que queremos tratar para lograr la total y suma atención del pequeño mientras aprende.

En este proyecto se logró un ambiente atractivo y didáctico con muchos métodos de apreciación, audición para mayor diversión y entendimiento metodológico, se utilizó herramientas confiables que han demostrado realizar bien su trabajo. También se contó con el plan de clases de inglés para las escuelas primarias de tercer grado, que fue la guía para llevar a cabo este proyecto.

Abstract

At present, many of the studies carried out have shown that the best stage to learn a new language is childhood (Nathalie Formation Center, 2019). But most children are very motivated, so many efforts have been made to call attention to the didactic stories, including videogames, and this is what we want to use to achieve the child maximum attention.

In this project we can include an attractive and didactic ambient, with many appreciation, audition for more fun and methodological learning.

Contenido

Pensamientos	2
Dedicatoria	3
Agradecimientos.....	4
Declaración de autoría.....	5
Opinión del tutor sobre el trabajo diploma.....	6
Resumen	7
Introducción.....	10
CAPÍTULO 1: Fundamentación Teórica	13
1.1 Juego Serio (Serious Game)	13
1.2 Clasificación de los videojuegos serios.	14
1.3 Teoría psicológica en el diseño del videojuego.	16
1.3.1 Factores cognitivos:.....	17
1.3.2 La memoria subconsciente:.....	18
1.4 Los videojuegos en la educación.....	18
1.5 Fases del proceso de Diseño y Desarrollo de un videojuego.....	20
1.5.1 Proceso de Desarrollo	20
1.5.2 Fase de Pre-producción.....	21
1.5.3 Fase de Producción	22
1.6 Metodología de desarrollo de Software.	25
1.7 Herramientas utilizadas	27
1.7.1 Gráficos 2D	27
1.7.2 Gráfica 3D	28
1.7.3 Edición de Sonido Adobe Audition CC.....	29
1.7.4 Motores de videojuegos.....	29
CAPÍTULO 2: Descripción de la solución.....	32
2.1 Documento de Diseño	32

2.2	Descripción de la Fase de Producción.....	39
2.2.1	Requisitos Software	39
2.2.2	Requisitos funcionales	41
2.2.3	Requisitos no funcionales del Videojuego.....	42
2.3	Elementos Dinámicos del Videojuego	43
2.3.1	Especificaciones de los elementos dinámicos.....	44
2.4	Implementación	46
2.4.1	Iteración 1: Diseño y composición del entorno.....	47
2.4.2	Iteración 2: Componentes	48
2.4.3	Iteración 3: Script de funcionalidades.....	50
	Conclusiones del capítulo.....	54
	CAPÍTULO 3: Pruebas de la solución	55
3.1	Pruebas de Rendimiento.....	55
3.2	Pruebas unitarias.....	57
3.3	Valoraciones Económicas y Análisis de Costo	58
	Conclusiones del capítulo.....	59
	Conclusiones	60
	Recomendaciones	61
	Bibliografía.....	62
	Anexos.....	64
	Editor Visual de Unity 3D.....	64
	Documento Diseño	65
	Diseño y composición del Entorno.....	71
	Componentes de Visión.....	76
	Componentes de Colisión.....	76
	Componentes de Animación.....	77

Introducción

A pesar de que en las escuelas primarias del país enseñan idioma y contamos con profesores clasificados para impartir las clases a veces esto no es suficiente para que nuestros pequeños se sientan atraídos hacia un nuevo idioma y mucho menos que les interese poder dominarlo, el inglés requiere de una memorización forzosa a veces un poco aburrida para los niños que no comprenden su importancia.

A muchas familias hoy en día siempre que le realizamos esta pregunta resulta que el niño desconoce el idioma o lo desarrolla muy mal, incluso cuando se les intenta enseñar les cuesta trabajo lograr la atención del muchacho, así que como una posible ayuda para incrementar el interés del muchacho hacia la nueva lengua, les brindamos este software, creado completamente siguiendo las reglas del plan de clases de las escuelas primarias de tercer grado, el cual tiene como finalidad evaluar, repasar y enseñar al niño el nuevo idioma a medida de que el mismo se divierte jugando la envolvente historia con muchas interacciones divertidas.

¿A través de qué?, a través de los ojos de nuestro personaje principal, un pionero, que tendrá que encontrar la manera de llegar al otro extremo de la isla para así por fin conquistar la gran torre y liberar la fantasía por el mundo, pero esto no será tarea fácil, pues en su camino se encontrará con obstáculos que le impedirán el paso y le pondrán a prueba su inglés con ingeniosos juegos o difíciles problemas. Será algo divertido y dinámico, la historia avanzará a medida de que el niño avance en su dominio del inglés, al igual que tendrá que interactuar constantemente con la computadora y sus periféricos, también se enamorará de personajes de la cultura cubana que serán protagonistas de esta insaciable aventura.

Existen cientos de juegos de diferentes variedades y formas para aprender inglés para niños, ¿por qué no hacer un juego con la educación que se exige en Cuba mientras es ambientado en nuestro país? Lograr un proyecto así podría iniciar nuevas formas para el aprendizaje de nuestros niños y abrir las puertas a un nuevo mundo de informatización académica donde los usuarios puedan acceder a un juego entretenido y lleno de aventuras donde se ejercita completamente gratis una nueva lengua.

El **Objetivo General**: Elaborar un juego en tercera dimensión que sirva como herramienta para ejercitar el inglés para los niños de siete a nueve años de edad.

Para dar solución al objetivo general se perfilan los siguientes **objetivos específicos**:

- 1- Desarrollar un Marco Teórico Referencial de la elaboración del juego.
- 2- Diseñar el juego utilizando una metodología que sirva para guiar el desarrollo de este tipo específico de software.
- 3- Implementar las mecánicas de juego acorde al nivel de estudio de los niños de 8 a 9 años de edad.
- 4- Realizar pruebas al videojuego garantizar un nivel óptimo en la aplicación.

Se determina como **problema científico**: ¿Cómo crear un videojuego enfocado a ejercitar el inglés de los niños entre ocho a nueve años con un énfasis didáctico y entretenido?

Se plantea la siguiente **hipótesis**: Al crear el software se podrá ayudar al incremento del aprendizaje del inglés para niños dando como resultado una mayor capacidad intelectual de la población.

Enmarcado en el **campo de acción** del desarrollo de videojuegos como herramienta de aprendizaje.

Análisis histórico-lógico: Permitted esclarecer las etapas de desarrollo histórico del problema a solucionar, indagar sobre las tecnologías a emplear y el funcionamiento de las herramientas informáticas para la gestión de los paquetes de juegos e ingeniería del mismo

Analítico-sintético: Este se precisó durante la revisión bibliográfica y el análisis de los resultados, permitiendo descomponer el proceso del negocio en partes y determinar las funciones que realiza cada uno de los roles determinados. De la misma manera se utiliza para la descomposición de cada flujo de trabajo en funcionalidades para cada rol establecido.

Inductivo-deductivo: Implicó el análisis de los sistemas existentes para encontrar similitudes y diferencias, adaptándolas a la solución a desarrollar. Su uso fue necesario tanto en la revisión bibliográfica, como en el análisis de los resultados, permitiendo arribar a conclusiones que se derivaron a partir de propiedades y relaciones existentes entre los elementos del fenómeno objeto de estudio.

En conclusión nuestro capítulo 1 abarcará las fundamentaciones teóricas explicando los conceptos de los Juegos Serios o Serious Game en inglés al igual que los factores cognitivos y beneficios del estudio del inglés a edades tempranas.

El capítulo 2 retomará toda la fase de implementación del software, con los requisitos funcionales y no funcionales, los diseños gráficos con las herramientas utilizadas y los algoritmos de programación o scripts en inglés.

Por último el capítulo 3 mostraremos las pruebas rendimiento que demuestran el flujo constante de información de nuestro programa y las pruebas unitarias para el código.

CAPÍTULO 1: Fundamentación Teórica

En el presente capítulo se hace un estudio de las diferentes clasificaciones de videojuegos serios existentes, así como los recursos psicológicos a utilizar durante la implementación del mismo. Se hace un análisis del plan de estudio de la Morfología Humana, además de un estudio de las tecnologías, herramientas y metodología a utilizar en el proceso de desarrollo de la aplicación.

1.1 Juego Serio (Serious Game)

En la actualidad se hace complejo definir el género de un videojuego, debido a que cada vez estos se interrelacionan y mezclan sus temáticas. La presente investigación se centra en el estudio de los videojuegos serios ya que son estos los que permiten representar los contenidos de una materia de manera divertida y dinámica.

En 2005, Mike Zyda abordó de una forma actualizada los conceptos referentes a que es un juego, un videojuego y un videojuego serio en un artículo publicado en la revista *"Computer"* de la *IEEE Computer Society* (Zyda, 2005).

Juego: una prueba física o mental, llevada a cabo de acuerdo con unas reglas específicas, cuyo objetivo es divertir o recompensar al participante.

Videojuego: una prueba mental, llevada a cabo frente a una computadora de acuerdo con ciertas reglas, cuyo fin es la diversión o esparcimiento, o ganar una apuesta.

Videojuego serio: una prueba mental, de acuerdo con unas reglas específicas, que usa la diversión como modo de formación gubernamental o corporativa, con objetivos en el ámbito de la educación, sanidad, política pública y comunicación estratégica.

Otra definición de renombrados profesionales del ámbito en cuanto a los videojuegos serios es la siguiente:

“Un videojuego serio es un juego en el que la educación (en sus diversas formas) es el principal objetivo, en lugar de entretenimiento (Clark, 1987).

Esta definición ayuda a aclarar la contradicción aparente, pero sin duda también genera confusión en algunas personas (tanto desarrolladores de videojuegos como educadores) que ven incompatibles al entretenimiento y la educación. La educación y el entretenimiento no son totalmente conflictivas entre sí, pues existen muchos lugares donde los dos conceptos se superponen y donde cada parte puede utilizar las herramientas de la otra para lograr sus objetivos.

1.2 Clasificación de los videojuegos serios.

Aunque la clasificación de los videojuegos serios es algo que todavía tiene que consolidarse, existen sin embargo una serie de términos que se han comenzado a utilizar en la actualidad los cuales se muestran a continuación. (Clark, 1987)

Advergaming: publicidad y juego, es la práctica de usar videojuegos para publicitar una marca, producto, organización o idea.

Edutainment: este es un término que resulta de la unión de *education* (educación) y *entertainment* (entretenimiento), es decir, educación y entretenimiento o diversión. Se aplica a los programas que enseñan mediante el uso de recursos lúdicos.

Aprendizaje basado en juegos: tienen como objetivo mejorar el aprendizaje. Están diseñados en general manteniendo un equilibrio entre la materia y la jugabilidad teniendo en cuenta la capacidad del jugador para retener y aplicar dicha materia en el mundo real. Es utilizado en el mundo empresarial para mejorar las capacidades de los empleados en temas como la atención al público y negociaciones.

Edumarket Games: cuando un juego serio combina varios aspectos (por ejemplo, los propios del *advergaming* y del *edutainment* u otros relacionados con la prensa y la persuasión), se dice que la aplicación es un juego de tipo *edumarket*, término que resulta

de la unión de *education* (educación) y *marketing* (publicidad). Un ejemplo es *Food Force*, un juego con objetivos en el ámbito de las noticias, la persuasión y el *edutainment*.

News Games: son juegos periodísticos (del inglés *news*, es decir, noticia) que informan sobre eventos recientes o expresan un comentario editorial. Un ejemplo es *September 12th*, un juego creado por Gonzalo Frasca que denuncia el uso de la violencia para resolver el problema del terrorismo.

Simuladores o videojuegos de simulación: son juegos que se emplean para adquirir o ejercitar distintas habilidades o para enseñar comportamientos eficaces en el contexto de situaciones o condiciones simuladas. En la práctica, son muy usados los simuladores de conducción de vehículos, coches, trenes, aviones, etc., como por ejemplo *FlightGear*), los simuladores de gestión de compañías (por ejemplo, *Transport Tycoon*) y los simuladores sobre negocios en general, que ayudan a desarrollar el pensamiento estratégico y enseñan a los usuarios los principios de la micro y macroeconomía y de la administración de empresas (por ejemplo, *Virtonomics*).

Militainment: este es un término que resulta de la unión de *military* y *entertainment*, es decir, militar y entretenimiento o diversión. Son juegos financiados por el ejército o que, de lo contrario, reproducen operaciones militares con un alto grado de exactitud.

Antecedentes

Durante la investigación, se encontró que existen en el mundo, pero ninguno en Cuba, videojuego serio desarrollado en tercera dimensión, sin embargo encontramos juegos que realizan el mismo objetivo con diferentes enfoques.

Entre los más destacados a nivel mundial se encuentran los títulos:

1. Scribble Nauts

2. Adjective research
3. Academy Island
4. Hangman
5. WordSpector
6. Word Mahjong
7. Vocabulary Quiz
8. Multi Popword
9. AphaVear English Word Games
10. The MOnkey Island
11. Duolingo

1.3 Teoría psicológica en el diseño del videojuego.

Todo videojuego implica el uso de factores psicológicos en mayor o menor medida para atrapar al jugador en su historia, para hacer que el usuario se enamore y quiera jugar el videojuego hasta el final. En la presente investigación se hace un estudio de varios de estos factores, pero particularmente en los que influyen directamente en la adquisición del conocimiento. (Cascales, 2007)

El ser humano almacena el conocimiento en estructuras semánticas. Las estructuras se organizan a partir de la interacción con el mundo externo usando los procesos perceptuales. Esta información se almacena en diferentes tipos de memoria y se rechaza o se mantiene según diversos criterios, lo que se puede llamar "aprendizaje". Existen dos modelos que permiten estudiar el aprendizaje de un determinado concepto por parte de una persona:

Modelo mental: A partir de las experiencias y el aprendizaje realizado por la persona, éste tiene un modelo inexacto y sólo conoce relaciones sencillas del concepto estudiado ("si se cumple esta condición, ocurre esto"). Por ejemplo, sabe que, si presiona el acelerador de un coche, se mueve, pero no sabe por qué (no entiende su principio mecánico).

Modelo conceptual: El estudio del concepto se hace de manera detallada y se conocen todos los mecanismos que actúan sobre él. Es el modelo detallado de un concepto. Siguiendo con el ejemplo anterior, un ingeniero mecánico conoce al detalle cómo está

construido un coche.

En el caso de la programación de aplicaciones, el modelo que tiene el programador sobre su aplicación sería un modelo conceptual, mientras que el modelo por el que se forman los diferentes usuarios sería un modelo mental. Para una misma aplicación, los dos modelos deben asemejarse tanto como sea posible. Cuanta más divergencia haya entre ellos, más probable será el rechazo por parte del usuario. Si se quiere que el videojuego sea intuitivo, se debe pensar en cómo crear un modelo conceptual para que se asemeje al del usuario.

De igual manera se hizo necesario revisar los estilos de aprendizajes, que no son más que los rasgos cognitivos, afectivos y fisiológicos que sirven como indicadores relativamente estables de cómo los discentes perciben, interactúan y responden a sus ambientes de aprendizaje.

1.3.1 Factores cognitivos:

Entre estos factores están:

- Las modalidades sensoriales.

La modalidad sensorial preferida por cada sujeto es, sin duda, otro elemento que debe analizarse. Los individuos se apoyan en distintos sentidos para captar y organizar la información, de forma que algunos autores la esquematizan de la siguiente forma:

- Visual o icónico lleva al pensamiento espacial.
- Auditivo o simbólico lleva al pensamiento verbal.
- Cinético o inactivo lleva al pensamiento motor.

(Cascales, 2007)

1.3.2 La memoria subconsciente:

A diario llega al cerebro mucha información a través de los órganos sensoriales; en realidad se percibe la mayoría de esta información, existiendo diferentes maneras de manejarlas. Las personas pueden ver cosas no solo de manera consciente, mientras más conscientes somos del entorno el cerebro percibe muchas más cosas de su alrededor, puesto que a través de la *visión periférica* (Cascales, 2007) se recibe mucha información del entorno, siendo el procesamiento preconscious el encargado de procesar toda la información en el subconsciente o mente interior antes de que sea registrada conscientemente. Todo esto sucede gracias a la visión periférica. No son solo las cosas que se ven las que afectan, las cosas que se ven de manera no consciente a menudo tienen más impacto que las que se observan conscientemente.

1.4 Los videojuegos en la educación.

El juego ha sido siempre una actividad tratada como un entretenimiento, tiempo éste dedicado a “distraerse” de las ocupaciones serias de la vida. Tiempo dedicado a “perderse” en cosas no productivas, incluida la educación, para proporcionar una diversión a niños y adultos. Una especie de tiempo muerto en las obligaciones productivas de la vida diaria. Desde hace muchos siglos, el juego ha sido una forma de educación e instrucción más, ejercicio básico para entender parte de ese adiestramiento social que las diferentes culturas han dotado a sus miembros.

En cuanto a la parte que tiene que ver con el aprendizaje, hay autores como Crawford que le otorgan al juego un valor inconsciente respecto al hecho de aprender. Aunque se piense que sólo es una diversión, un entretenimiento, se juega básicamente porque aprendemos y eso nos gusta. Incluso, desde un punto de vista más integral, el juego ha sido visto como un vehículo de desarrollo integral. (Crawford, 2010)

Estas definiciones se basan en todo tipo de juegos, pero si se centran en los videojuegos, se puede observar que, aparte de estas características ya mencionadas para todos los juegos, se adquieren otro tipo de habilidades. Quizás el efecto más claro de su potencial formativo se produce en la

adquisición de competencias digitales; es decir, en el contexto tecnológico y digital de las sociedades actuales, la mayoría de los niños accede por primera vez al universo digital a partir de los videojuegos. Así, desarrollan competencias propias de la alfabetización digital de manera recreativa y lúdica, las cuales les sirven para iniciarse en el manejo de las interfaces gráficas y las lógicas del mundo digital. Aun así, hay que advertir que el uso del videojuego para el entretenimiento no parece asegurar una transferencia de los aprendizajes digitales, sino que aumenta las posibilidades respecto a los que no juegan. (UCL, 2008)

Desde el comienzo del auge de los juegos serios se han comenzado a introducir en las aulas videojuegos y multimedia que se utilizan como herramienta pedagógica, centrado no tanto en la consecución de victorias o avances, sino en la resolución del problema que presenta el videojuego. Esto obligará a los alumnos a concebir estrategias de investigación y exploración conjunta para poder avanzar dentro del videojuego. (Morales, 2013)

Si todo esto lo combinamos con los beneficios que obtenemos al darle a los niños un aprendizaje temprano del inglés los cuales son: (Brains Nursey Schools, 2019)

- Mayor facilidad de aprendizaje
Desde el nacimiento hasta los 3 años el cerebro tiene su máxima plasticidad, potenciando el aprendizaje.
- Impacto positivo en la salud mental
El impacto a la salud mental se ve demostrado en una mayor capacidad de concentración, posibilidad de realizar varias tareas al mismo tiempo con eficacia, mejora la atención y la memoria.
- Acceso a un mundo Globalizado
Aumentará su interés por otras culturas y formas de vida, alimentando su desarrollo personal.
- Pérdida de la vergüenza
Los niños no tienen vergüenza de nada y aprenden sin miedo, por lo que adquieren seguridad en este campo de cara al futuro.

- Mayor empleabilidad

El saber otro idioma aumenta en un 44% las posibilidades de encontrar empleo en un futuro.

1.5 Fases del proceso de Diseño y Desarrollo de un videojuego.

A lo largo de este apartado se comentará cómo se diseña un videojuego y cómo se conforma el equipo de desarrollo. Se describirá la metodología de Ingeniería del Software usada y qué actividades se realizan para crear un videojuego. Podríamos definir diseño y desarrollo de un videojuego como las actividades multidisciplinares en la cual se diseña y se produce un videojuego, desde su concepto inicial, hasta su versión final.

1.5.1 Proceso de Desarrollo

Aunque los videojuegos se consideran un tipo de software más, no existe una metodología común y propia para su diseño y posterior desarrollo. Se puede asegurar, que realizar un videojuego no es una tarea sencilla y que requiere de mecanismos y metodologías de desarrollo de software propio, sin embargo, no existe una metodología estándar que rija dichos procesos y asegure su calidad. Son las propias compañías las que fijan cuál será su filosofía de trabajo a lo largo de la creación de un juego. En lo que sí coinciden diversos autores como Rollings (Morris & Rollings) y especialmente Bethke (Bethke, 2003), es que el desarrollo del juego, a lo largo de su ciclo de vida, se puede asemejar al de una película de cine, pudiéndose segmentar en tres fases ampliamente diferenciadas: Pre-Producción, Producción y Post-producción, cada una con sus etapas características.

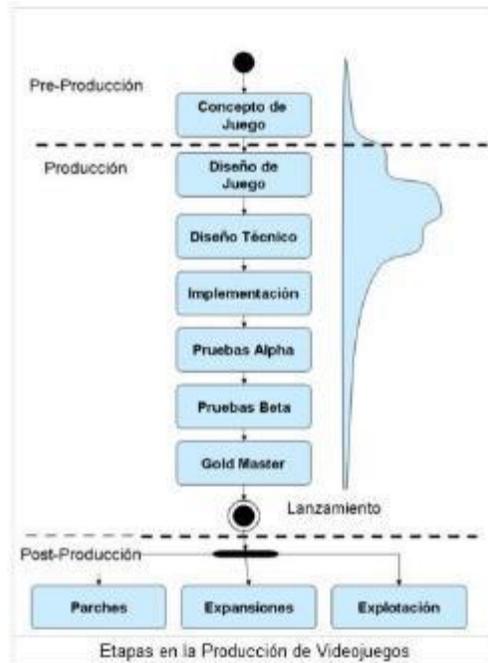


Fig. 1: Etapas en la producción de videojuegos.

1.5.2 Fase de Pre-producción

La fase de Pre-producción es la etapa inicial del proceso de desarrollo, se caracteriza sobretodo porque se realiza la concepción de la idea del videojuego, es decir, los aspectos fundamentales que conformarán el videojuego a lo largo de su construcción, a continuación, se definen cada una de las tareas a realizar en esta etapa: (Morris & Rollings)

Género: Se debe especificar el género o los géneros al que pertenece el juego para así establecer las características básicas para su posterior diseño.

Historia: Se debe realizar un esbozo de la trama o historia a desarrollar por el juego, indicando qué se quiere contar y cómo se quiere contar (storyline y storytelling).

Bocetos: Se crean bocetos o diseños preliminares de los personajes y dónde transcurrirá la acción del juego, ya sean decorados, ambientaciones, ropaje, música, movimientos entre otros.

Gameplay: Es la parte más importante de este proceso de concepción del juego. El gameplay es un concepto amplio y difuso, que se define de manera diferente en cada juego. Se podría definir como la esencia, grado o naturaleza del videojuego, incluyendo parte de la interactividad. Es aquí donde se define cómo se va a jugar, qué cosas se pueden hacer en el juego y cómo va reaccionar el entorno del juego a las acciones del jugador a través del personaje. A su vez se establece cómo será la curva de aprendizaje del jugador. Todo esto sin entrar en detalles gráficos, sonoros o de historia. (Zimmerman, 2003)

Una vez cerrada la fase de Pre-producción se debe crear la primera versión del documento de diseño del videojuego (del inglés Game Design Document, GDD) (Bethke, 2003). En esta versión se muestran las primeras especificaciones del juego, plasmando los puntos anteriores. Es la base fundamental para comenzar la fase de Producción, en especial la de diseño del videojuego, pues fija todo lo que se trabajará en ella.

1.5.3 Fase de Producción

Es la fase del proceso que más tiempo demora, es donde los mayores esfuerzos se invierten en las etapas de diseño del videojuego y del diseño técnico. Es en estas etapas donde más personas colaborarán, pues al equipo de diseño de videojuego inicial se le incorporará el resto de la plantilla asociada a la producción del juego. Concluida todas las etapas que componen la etapa de producción se tiene una primera versión del producto final. A continuación, se describen cada una de las etapas. (Morris & Rollings)

Diseño de Juego: se detallan todos los elementos que compondrán el juego, dando una idea clara a los miembros del grupo desarrollador de cómo son. Se termina el GDD diseñando en profundidad los aspectos anteriormente especificados.

Diseño Artístico: se crea la Biblia de la Historia donde se recogen todas las historias de los personajes del mundo donde sucede el videojuego, de su pasado y de los personajes secundarios que aparecen, creando el hilo argumental completo, con todos los detalles.

Sonido: Se diseñan todos los elementos sonoros del videojuego: voces, ambiente, efectos y música. Se comienza el motor de sonido.

Interfaz: Se describe la forma en que se verán los elementos GUI (del inglés Graphical User Interface) y HUD (del inglés Head-Up Display), mediante los cuales el usuario interactuará con el juego.

Gráficos: Dependiendo de si el juego es 2D o 3D se diseñarán los elementos gráficos como los *sprites*, modelos 3D, cámaras y luces a utilizar. Se da comienzo al motor gráfico.

Diseño Mecánico: Se diseña cómo se va a interactuar en el juego y las reglas que rigen a éste y las comunicaciones que deben darse en caso de jugarse on-line. Además, se diseña el comportamiento de los personajes y del mundo que les rodea, así como sus habilidades y otros detalles. Se comienza a diseñar la Inteligencia Artificial (IA) del juego y el motor asociado a ella. También se diseña el Motor Físico, encargado de generar diversos aspectos físicos de los elementos y del mundo donde se lleva a cabo el juego (explosiones, disparos, caídas).

Motor del Juego: Es el elemento más importante, y se puede decir que es la parte fundamental de este proceso tan costoso y duradero. El Motor del Juego hace referencia a una serie de rutinas que permiten la representación de todos los elementos del videojuego. Del mismo modo en que la carrocería, la pintura y los exteriores no andan sin un motor, el arte y los guiones del juego no funcionan sin un motor del juego. Es aquí donde se debe controlar cómo se representan los elementos del juego y cómo se interactúa con ellos. Se gestionará la IA del videojuego, los sonidos asociados a cada elemento y todos los aspectos gráficos asociados a éstos, incluida la *cinemática* de éste. Se puede decir que el Motor del Juego equivale a una conjunción del Motor Gráfico, Motor de Sonido, Gestor de IA y Motor Físico, más las reglas necesarias para crear el universo completo del juego.

Diseño Técnico: Esta es la etapa más relacionada con el diseño del Software, pues es donde se tratará el videojuego como un verdadero producto de software. Esta etapa describirá cómo se llevará a cabo la implementación del videojuego en una máquina real a través de una determinada metodología y mediante un lenguaje concreto. Se generarán los diagramas que describan el funcionamiento estático y dinámico, la interacción con los usuarios y los diferentes estados que atravesará el videojuego como software.

Esta fase incluye la planificación del juego y en ella se identifican las tareas necesarias para desarrollarlo, repartiéndolas entre los distintos componentes del equipo desarrollador.

Implementación: La etapa de implementación consiste en “montar todas las piezas” descrito anteriormente utilizando el Motor del Juego. Se finalizan todos los contenidos del juego: misiones, scripts, efectos e IA. Este proceso tiene pocas innovaciones y es donde se pulen algunos errores detectados en el diseño inicial. No existen dependencias y el ritmo de trabajo debe ser constante por parte del equipo de desarrollo. Se pueden desechar ideas que son buenas, pero que no encajan al final del juego y corregir imprevistos.

Pruebas Alpha: Durante las pruebas Alpha o Code Complete se tiene un producto terminado. Este producto es probado por un equipo pequeño, que ha estado involucrado en el diseño y desarrollo del juego, en busca de errores para su refinamiento. Uno de los aspectos a probar es la Jugabilidad y de la forma de probarla se hablará a lo largo de este trabajo.

Pruebas Beta: En las pruebas Beta o Content Complete se terminan todas las variaciones del contenido (decorado de misiones, gráficos, textos en diferentes idiomas, doblaje, etc.). Estas pruebas se realizarán por un equipo externo al equipo de desarrollo, ya sean externos a la empresa o pertenecientes al proyecto, intentando conseguir que el videojuego vea la luz con la menor cantidad posible de defectos.

Gold Master: Es exactamente el juego que se publicará y se enviará a la fábrica para su producción con todo el contenido de arte (diseño de portada, caja, etc.) y manuales de usuario.

1.6 Metodología de desarrollo de Software.

La industria del videojuego se aferró muchos años a utilizar la metodología cascada (Flood, 2003) para guiar los procesos de desarrollo, esta metodología no era la más óptima pues no estaba orientada a la construcción de videojuegos, lo cual hizo necesario la búsqueda de otras alternativas. En principio hay que considerar que un videojuego es una aplicación de software, esto originó la búsqueda de procesos ya existentes para el desarrollo de software; se analizaron en torno a sus actividades, roles y artefactos para ver qué tan plausibles eran para guiar proyectos de videojuegos. En la siguiente tabla se hace un análisis de algunas de las metodologías ya existentes:

Tabla 1: Metodologías de Software.

Procesos Viables al Desarrollo de Videojuegos		
Metodología	Descripción	Valoración
Modelo en Cascada	Proceso de desarrollo de software especializado secuencial en el cual el desarrollo se basa en el modelo Cascada a través de las fases de concepción, iniciación, análisis, diseño, construcción y pruebas. (Flood, 2003)	El producto final se demora más de lo esperado ya que cualquier problema que se presente en una de las etapas se tiene que regresar a una anterior para corregirlo. Requiere hacer muchos cambios a los documentos y regresarse a etapas anteriores propicia a que se vuelva un proceso muy desordenado. Sin embargo es uno de los procesos más populares en la industria de los videojuegos.

Rational Unified Process	Es un proceso de desarrollo de software iterativo, es adaptable y entallable para satisfacer las necesidades del equipo del proyecto. Comúnmente sigue una metodología pesada. (Flood, 2003)	Proceso muy completo, pero no está enfocado al desarrollo de videojuegos. Demasiada documentación permite realizar un buen producto con lo que ello conlleva (tiempo, dinero y personas involucradas). Sin embargo esta metodología genera mucha documentación innecesaria para el desarrollo de un videojuego.
Scrum Framework	Scrum es un framework de desarrollo de software iterativo-incremental utilizado en el desarrollo de software ágil. El trabajo está estructurado en ciclos conocidos como sprints. Durante cada sprint los equipos toman los requerimientos de una lista ordenada por prioridades conocidas como historias de usuario. Al terminar cada sprint, se tiene una versión potencialmente final del producto. (Scrum Alliance, 2019)	Scrum facilita la iteración, permite a los equipos entregar características pulidas para probar la calidad del juego a lo largo de su desarrollo y así incorporar la retroalimentación de jugadores. Scrum no es solo para programadores, involucra a muchas personas a un solo proyecto. Es útil debido a que los videojuegos hoy en día se vuelven más complejos e involucran a personas multidisciplinarias. Por dichas razones, se considera Scrum ideal para el desarrollo de videojuegos.

Después de un estudio de las diferentes metodologías, se desprende que no existen procesos específicos para el desarrollo de videojuegos de manera pública. En tal sentido se publica poca información, modelos, plantillas y herramientas, por lo tanto, puede considerarse que no existe una metodología estándar para el desarrollo de videojuegos.

Huddle es una de las pocas metodologías de desarrollo que está al alcance de todos, ya que cada empresa de videojuego define su propio estilo de trabajo. La misma tiene las siguientes características: es ágil, óptimo para equipos multidisciplinarios de 5 a 10 personas, iterativo, incremental y evolutivo. Huddle, sin embargo, puede utilizarse en equipos de menos de cinco elementos. Permite guiar proyectos de videojuegos, la cual está fundamentada siguiendo la analogía de SCRUM y su base es proveniente

de las fases de desarrollo propuestas por Rollings y Morris en el 2003, anteriormente tratadas en el epígrafe 1.5.

Todo el proceso se divide en 3 fases:

- Preproducción.
- Producción.
- Postmortem.

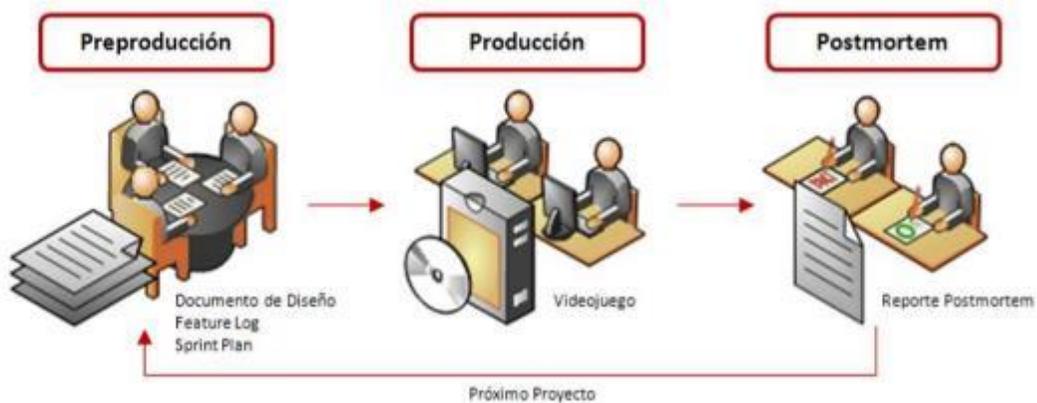


Fig. 2: Las 3 fases del proceso de Huddle.

1.7 Herramientas utilizadas

A continuación, se dará una breve explicación de cada uno de los programas utilizados durante la realización del proyecto; ya haya sido de forma exhaustiva o simplemente para momentos puntuales.

1.7.1 Gráficos 2D

- **Pixlr Express**

Pixlr Express cuenta con las opciones de tratar fotos pero además puedes importar imagen de una url, utilizar un collage o incluso usar la webcam para realizar una captura.

Podrás contar con: (The Photo Editor for Every, 2019)

- Infinidad de plantillas para decorar fotografías
- Ajustarlas en tamaño a tu antojo
- Uso de bordes
- Colores y sombreados
- Trabajar en capas
- Difuminados
- Correctores de fotos
- Reemplazar color
- Transformar objetos
- Añadir transparencias
- Importar imágenes
- Filtros
- Recortar o rotar fotografías
- Asignar colores sepias, efecto
- Recortar brillos, tonos

(Pixlr, 2019)

1.7.2 Gráfica 3D

- **Blender**

Blender proporciona potentes herramientas integradas de modelado, animación y renderización en 3D que permiten a los artistas y los diseñadores dedicar más energía a la creatividad en lugar de a las dificultades técnicas. Ofrece herramientas especializadas a los desarrolladores de juegos, creadores de efectos visuales, diseñadores de gráficos de movimiento y otros profesionales de la creatividad que trabajan en el diseño de medios.

(Blender, 2019)

1.7.3 Edición de Sonido Adobe Audition CC

El software Adobe Audition CC proporciona las herramientas profesionales que necesita para dotar del mejor sonido a sus producciones de vídeo y audio. Gestiona de forma eficaz una amplia gama de tareas de producción de audio, incluidas la grabación, mezcla y restauración de sonido.

(Adobe, 2019)

1.7.4 Motores de videojuegos

Un motor de videojuego es una herramienta con funcionalidades reutilizables (implementadas previamente), que permiten automatizar determinadas tareas en el desarrollo de videojuegos. De este modo se evita lidiar con procesos de bajo nivel de gran complejidad. Este provee al videojuego de un motor para renderizar gráficos 2D y 3D, motor físico o detector de colisiones, sonidos, *scripting*, animación, IA, redes, *streaming*, administración de memoria y un escenario gráfico.

Para el desarrollo de la solución se usará el motor de videojuego Unity 3D en su versión 2018, debido a que es la herramienta que actualmente se utiliza en los principales centros de desarrollo de videojuegos en Cuba como son el centro Vertex de la Universidad de Ciencias Informáticas, el Estudio de Animación del ICAIC y Cinesoft. Está disponible para sistemas operativos, Windows y Macintosh y viene empaquetada como una herramienta para crear videojuegos, aplicaciones interactivas, visualizaciones y animaciones en 3D y 2D en tiempo real. Implementa contenido para múltiples plataformas como Windows, Mac OS, Linux, Xbox 360, Xbox One, PlayStation 3, PlayStation 4, PlayStation Vita, Wii, Wii U, Nintendo 3DS, iOS, Android y Windows Phone, además de contar con un *plugin web* para desarrollar videojuegos para navegador. (UnityTechnologies, 2019)

En Unity, las mecánicas de videojuegos son implementadas mediante *scripts* y compiladas usando una versión de *JavaScript* o *C#*. Además, está enfocada en los bloques de construcción (*assets*) y facilita su comprensión y manipulación en tiempo de edición, para mejorar el rendimiento del videojuego. Los *assets* pueden ser: texturas, modelos 3D, archivos de audio, prefabricados (*prefabs*), materiales, animaciones, *scripts* y cualquier contenido utilizado en la creación del videojuego. Permite empaquetar y exportar en un archivo con extensión propia del motor, colecciones de *assets* que conjuntamente conforman mecánicas, componentes o recursos, que se pueden importar en futuros proyectos de manera flexible. Esto potencia la reutilización, obteniéndose como consecuencia la aceleración y optimización del tiempo de desarrollo y minimizando el esfuerzo necesario para su realización. (UnityTechnologies, 2019)

Unity ofrece un editor visual para crear videojuegos, siendo este el centro de la línea de producción. A continuación, se describen las principales áreas que lo conforman.

[Ver Figura 3. Captura de pantalla de la interfaz del editor de Unity 3D.](#)

Editor visual de Unity 3D

Está compuesto por cinco áreas principales (No 1. Ventana de escena, No 2. Ventana de juego, No 3. Ventana de jerarquía, No 4. Ventana de proyecto y No 5. Ventana de inspector), como se observa en la siguiente figura (UnityTechnologies, 2019):

Ventana de escena: es el área de construcción de Unity donde se confecciona visualmente cada escena del juego.

Ventana de juego: en esta ventana se obtiene una pre visualización del juego. Permite reproducirlo en cualquier momento del desarrollo para ir chequeando que funcione correctamente.

Ventana de jerarquía: esta ventana visualiza todos los objetos presentes en la escena actual.

Ventana de proyecto: permite manipular y visualizar el conjunto de *assets* que se emplea en el juego. En ella se pueden importar objetos 3D de distintas aplicaciones, texturas, audios, *scripts*, *prefabs*, entre otros, para usarlos en el videojuego.

Ventana de inspector: la ventana de inspector tiene varias utilidades, si se selecciona un objeto, entonces mostrará sus propiedades y podrán ser personalizadas. También permite configurar ciertas herramientas como la de terrenos si se tiene un terreno de Unity seleccionado.

Otra de las potencialidades que ofrece este motor es el sistema de mallas de navegación implementado, útil para trasladar unidades del juego hasta un destino determinado por el camino más óptimo y evadiendo obstáculos.

CAPÍTULO 2: Descripción de la solución

El presente capítulo contiene todas las especificaciones necesarias para comenzar el proyecto, éstas van desde el tema principal del videojuego hasta el número de niveles que tendrá. En el mismo se asienta toda la información detallada para la composición del guion técnico siguiendo las características de la metodología Huddle. Esta información queda registrada en un artefacto ingenieril llamado documento de diseño (tabla 3), artefacto esencial de la metodología Huddle.

2.1 Documento de Diseño

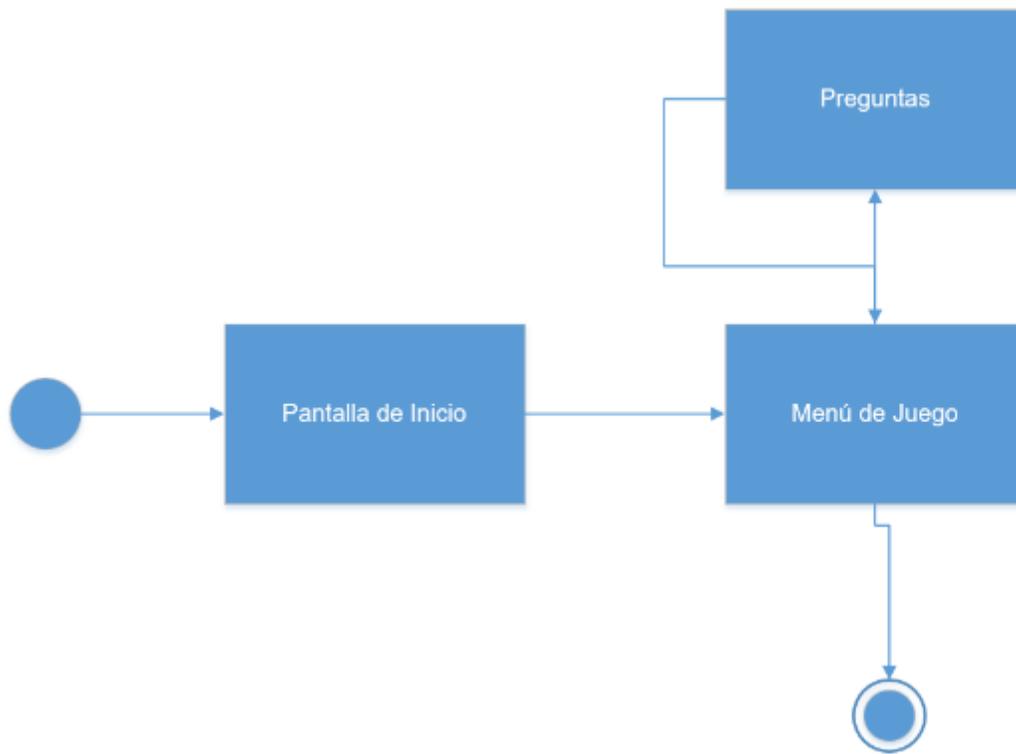
Concepto	
Título	English Journey
Estudio/Diseñadores	Frank David Rico Rodríguez
Género	Juego Serio
Plataforma	Computadora Personal
Versión	1
Sinopsis de Jugabilidad y Contenido	Eres un pionero con la misión de avanzar por el mundo venciendo las pruebas y desafíos que te encuentras en el camino, con la misión de recolectar lápices para poder enfrentar los problemas.

Categoría	Edutainment
Mecánica	El jugador puede caminar libremente por el ambiente, también puede coger objetos específicos de la escena y colocarlos en determinados lugares para completar desafíos o desactivar trampas.
Tecnología	<ul style="list-style-type: none"> • Hardware (Requerimientos mínimos para implementar el juego): <ul style="list-style-type: none"> - Tarjeta Gráfica con 64 MB. - El uso del Occlusion Culling requiere GPU • Software (Requerimientos mínimos para implementar el juego): <ul style="list-style-type: none"> - Gráficos 2D: <ul style="list-style-type: none"> • Photoshop CS6 - Gráficos 3D: <ul style="list-style-type: none"> • Autodesk 3DMax - Edición de Sonido: <ul style="list-style-type: none"> • Adobe Audition CS6 - Motor de Videojuego:
Público	Está dirigido a estudiantes de educación primaria, fundamentalmente de siete a nueve años de edad.
Historial de Versiones	

Mecánica del Juego

Mecánica del Juego	
Cámaras	Cámara 3D en 3era Persona
Periféricos	Teclado y Mouse
Controles	Para moverse se usará los Axis de Control (Up, Down, Left y Right) Para ver el menú pause la tecla Esc
Puntajes	La forma de puntuación está dada por las respuestas correctas que el estudiante es capaz de marcar satisfactoriamente. Esto al final se ve en el acumulado general.
Guardad/Cargar	Se guarda sólo durante la partida del juego

Estado del Juego



Interfaces

Menú Principal

Nombre de la Pantalla

Menú Principal

Descripción de la Pantalla

Es la interfaz inicial del juego, en ella el jugador tiene acceso a:

Comenzar Partida: para iniciar el juego.

Nombre: Poner su nombre en el juego.

Salir: Para salir del juego

Imagen



Niveles

Bosque

Título del Nivel	Bosque
Encuentro	Este es el primer escenario en el que comenzara el juego.
Descripción	Un pequeño bosque fantástico en la que se encuentran animales y personas que interactúan con el usuario.

Objetivos	Superar las pruebas para avanzar
Progresos	Después de superar cada prueba se calificará al final su estancia y se decide si puede pasar al nivel de bono.
Ítems	Lápices y globos
Personajes	Cocodrilo, niña granjera, profesora, sabio y mago.
Música y Efectos de Sonido	La música utilizada en este nivel recrea un ambiente tranquilo.
Referencias de BGM y SFX	Fugue: Música de fondo.

Personajes	
Personaje Principal	
Nombre del Personaje	Pionero
Descripción	Pionero uniformado con pañoleta azul.

Miembros del Equipo
Diseñador y desarrollador: Frank David Rico Rodríguez
Detalles de Producción

Imagen	
Encuentro	Entra en escena desde el principio.
Habilidades	Correr y saltar
Fecha de inicio	15/11/2018
Fecha de Terminación	13/05/2019

Tabla 3. Documento de Diseño

Para ver las demás tablas:

[Ver Tabla4. Documento Diseño.](#)

2.2 Descripción de la Fase de Producción

En el proceso de Producción, siguiendo la metodología Huddle, cada rol ejecuta sus tareas atendiendo a su especialidad. A continuación, se describen los procesos de ingeniería a realizar para la construcción del videojuego, los cuales permiten identificar las necesidades del producto mediante la obtención de los requisitos funcionales y no funcionales, ayudando al desarrollador a comprender la naturaleza del producto a construir, así como detectar las funcionalidades requeridas. El desarrollo de este epígrafe comenzará con la extracción de los requisitos del software, funcionales y no funcionales que se han identificado, posteriormente se continuará con el análisis detallado de los casos de uso y la creación de los diagramas de clases que guiarán la fase de implementación.

2.2.1 Requisitos Software

A continuación, se describen los requisitos software del videojuego, estos requisitos se han concretado en las diferentes reuniones establecidas según la metodología. Con el fin de facilitar la tarea de comprensión por parte del lector, en primer lugar, se explica la estructura de cada uno de los requisitos, así como los campos por los que está compuesta su definición, y posteriormente mostrar la manera en la que están agrupados.

Cada requisito estará representado por una tabla de la siguiente forma:

Identificación							
Título							
Descripción							
Prioridad	Alta	Media	Baja	Estabilidad	Sí	No	
Claridad	Alta	Media	Baja	Verificabilidad	Alta	Media	Baja
Necesidad	Esencial		Deseable		Opcional		
Fuente							

Tabla 4: Tabla ejemplo de requisitos

Donde cada campo significa lo siguiente:

- **Identificación:** Código que identifica de forma unívoca cada uno de los requisitos.
- **Título:** Título descriptivo del requisito.
- **Descripción:** Explicación clara y concisa del requisito que se está especificando.
- **Prioridad:** Orden de cumplimiento de un requisito. A mayor prioridad, mayor urgencia para realizarlo.
- **Estabilidad:** Probabilidad de cambio de un requisito. A mayor estabilidad, menor posibilidades de que dicho requisito se vea modificado.
- **Claridad:** Cataloga la no ambigüedad de un requisito. Cuanta mayor claridad tenga, menor ambigüedad contendrá.
- **Verificabilidad:** Facilidad para comprobar que el sistema cumple un requisito.
- **Necesidad:** Esencialidad. Interés del cliente en la realización de un requisito.
- **Fuente:** Origen del requisito ya sea una persona o documentación.

Por otro lado, con ánimo de ofrecer una estructura que facilite la realización y comprensión de los requisitos, se van a organizar en dos categorías:

- **Requisitos de software funcionales:** especifican qué tiene que hacer el software. Definen el propósito del software.

Requisitos de software no funcionales: especifican como el software debe hacer las cosas y que debe cumplir.

2.2.2 Requisitos funcionales

Las siguientes tablas se corresponden con los requisitos funcionales que se han extraído del videojuego a desarrollar:

Identificación: RF-VG-001							
Título	Entrar nombre del personaje						
Descripción	El sistema permitirá que el usuario añada su nombre y que este sea utilizado en las preguntas y respuestas durante el juego.						
Prioridad	<u>Alta</u>	Media	Baja	Estabilidad	<u>Sí</u>	No	
Claridad	<u>Alta</u>	Media	Baja	Verificabilidad	<u>Alta</u>	Media	Baja
Necesidad	Esencial		Deseable		Opcional		
Fuente	Autor del proyecto						

Tabla 05: RF-VG-001 – Entrar nombre del personaje.

Identificación: RS-VG-002							
Título	Sistema de locomoción y cámaras						
Descripción	El sistema permitirá al personaje moverse libremente por el entorno restringiendo solo las áreas con colisiones deseadas. La cámara perseguirá este movimiento.						
Prioridad	<u>Alta</u>	Media	Baja	Estabilidad	<u>Sí</u>	No	
Claridad	<u>Alta</u>	Media	Baja	Verificabilidad	<u>Alta</u>	Media	Baja
Necesidad	Esencial		Deseable		Opcional		
Fuente	Autor del proyecto						

Tabla 06: RF-VG-002 – Sistema de locomoción y cámaras.

Identificación: RF-VG-003							
Título	Sistema de preguntas y respuestas						
Descripción	El sistema permitirá la realización de preguntas predeterminadas al usuario y la selección de respuestas. A su vez evaluará las respuestas correctas.						
Prioridad	<u>Alta</u>	Media	Baja	Estabilidad	<u>Sí</u>	No	
Claridad	<u>Alta</u>	Media	Baja	Verificabilidad	<u>Alta</u>	Media	Baja
Necesidad	Esencial		Deseable		Opcional		

Fuente	Autor del proyecto
--------	--------------------

Tabla 07: RF-VG-003 – Sistema de preguntas y respuestas.

Identificación: RF-VG-004							
Título	Sistema de recolección						
Descripción	El sistema permitirá al personaje recolectar diferentes objetos predeterminados en el entorno.						
Prioridad	<u>Alta</u>	Media	Baja	Estabilidad	<u>Sí</u>	No	
Claridad	<u>Alta</u>	Media	Baja	Verificabilidad	<u>Alta</u>	Media	Baja
Necesidad	<u>Esencial</u>		Deseable		Opcional		
Fuente	Autor del proyecto						

Tabla 08: RF-VG-004 – Sistema de recolección.

Identificación: RF-VG-005							
Título	Sistema de interacción						
Descripción	El sistema permitirá al personaje interactuar con otros personajes y recibirá una respuesta de la interacción.						
Prioridad	<u>Alta</u>	Media	Baja	Estabilidad	<u>Sí</u>	No	
Claridad	<u>Alta</u>	Media	Baja	Verificabilidad	<u>Alta</u>	Media	Baja
Necesidad	<u>Esencial</u>		Deseable		Opcional		
Fuente	Autor del proyecto						

Tabla 09: RF-VG-005 – Sistema de interacción.

2.2.3 Requisitos no funcionales del Videojuego

Un requisito no funcional o atributo de calidad es, en la ingeniería de sistemas y la ingeniería de software, un requisito que especifica criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos. Las siguientes tablas se corresponden con los requisitos no funcionales del videojuego:

Identificación: RNF-VG-001							
Título	Rendimiento						
Descripción	El ordenador tiene que estar acordes a los requerimientos mínimos para que la aplicación pueda ejecutarse correctamente.						
Prioridad	<u>Alta</u>	Media	Baja	Estabilidad	<u>Sí</u>	No	

Claridad	<u>Alta</u>	Media	Baja	Verificabilidad	<u>Alta</u>	Media	Baja
Necesidad	<u>Esencial</u>		Deseable		Opcional		
Fuente	Desarrollador						

Tabla 10: RNF-VG-001 – Rendimiento.

Identificación: RNF-VG-002							
Título	Compatibilidad						
Descripción	El sistema deberá ser compatible con las diferentes versiones del sistema operativo Windows y con las diferentes resoluciones de pantalla.						
Prioridad	<u>Alta</u>	Media	Baja	Estabilidad	<u>Sí</u>	No	
Claridad	<u>Alta</u>	Media	Baja	Verificabilidad	<u>Alta</u>	Media	Baja
Necesidad	<u>Esencial</u>		Deseable		Opcional		
Fuente	Desarrollador						

Tabla 11: RNF-VG-002 – Compatibilidad.

Identificación: RNF-VG-003							
Título	Usabilidad						
Descripción	El videojuego puede ser jugado por cualquier persona sin tener en cuenta su nivel académico, con tener un conocimiento previo de informática puede disfrutar del mismo.						
Prioridad	<u>Alta</u>	Media	Baja	Estabilidad	<u>Sí</u>	No	
Claridad	<u>Alta</u>	Media	Baja	Verificabilidad	<u>Alta</u>	Media	Baja
Necesidad	<u>Esencial</u>		Deseable		Opcional		
Fuente	Desarrollador						

Tabla 12: RNF-VG-003 – Usabilidad.

2.3 Elementos Dinámicos del Videojuego

En esta sección se van a detallar los diferentes elementos dinámicos del videojuego (también se les conoce en el rol de ingeniería de software como casos de uso) que se han definido durante la fase de análisis del proyecto. Los elementos dinámicos cubren las distintas funcionalidades que podrán realizar los usuarios que interactúen con el producto.

La funcionalidad principal de la aplicación desarrollada es el aprendizaje de inglés mediante un videojuego, consiguiendo así el entretenimiento y la formación del usuario. En el siguiente apartado se definirán las funcionalidades del actor que interactuará con el videojuego.

2.3.1 Especificaciones de los elementos dinámicos

En este apartado se va a detallar el formato de las tablas utilizadas para definir los diferentes casos de uso que se extraen de las dos aplicaciones desarrolladas. Para ello se hará uso de la siguiente tabla:

Identificador	
Nombre	
Descripción	
Actor	
Precondiciones	
Pos-condiciones	
Escenario	

Tabla 13: Tabla ejemplo.

Donde cada campo significa lo siguiente:

- **Identificador:** Código que identifica de forma unívoca cada mecánica.
- **Nombre:** Nombre descriptivo de la mecánica.
- **Descripción:** Explicación clara y concisa de la mecánica que se está especificando.
- **Actor:** tipo de usuario de la aplicación.
- **Precondiciones:** Condiciones que se deben cumplir previamente para poder realizar una

determinada operación.

- **Pos-condiciones:** Estado que presenta el sistema tras la ejecución de una determinada operación.
- **Escenario:** Ejecución de la mecánica en el escenario.

El jugador, una vez iniciada la aplicación, podrá realizar un grupo de mecánicas de juego que conforman el sistema.

Identificador: MJ0001	
Nombre	Locomoción
Requisitos funcionales asociados	<ul style="list-style-type: none">• Sistema de locomoción y cámaras.
Descripción	El usuario utilizará las teclas Axis de Control (Up, Down, Left y Right) del teclado para mover al personaje por la pantalla.
Actor	Jugador
Precondiciones	El jugador debe de encontrarse en la pantalla del juego.
Pos-condiciones	El personaje se desplazará por la pantalla.
Escenario	El usuario desde el juego deberá pulsar las teclas Up, Down, Left y Right del teclado.

Tabla 14: MJ0001- Locomoción.

Identificador: MJ0002	
Nombre	Coger objetos
Requisitos funcionales asociados	<ul style="list-style-type: none">• Sistema de recolección.• Sistema de interacción
Descripción	El jugador podrá recoger los objetos acercándose a ellos.
Actor	Jugador
Precondiciones	Debe de tocarlos o pasarles por encima.
Pos-condiciones	El sistema almacenará los objetos.
Escenario	El usuario desde la escena deberá mover al personaje hasta acercarse a los objetos.

Tabla 15: MJ0002- Coger objetos.

Identificador: MJ0003	
Nombre	Interacción con personajes
Requisitos funcionales asociados	<ul style="list-style-type: none"> • Sistema de preguntas y respuestas
Descripción	El jugador al acercarse a otros personajes recibirá una respuesta, ya sea una información o una pregunta.
Actor	Jugador
Precondiciones	Debe acercarse a algún personaje
Pos-condiciones	Si el personaje posee alguna información personalizada se le mostrara al usuario.
Escenario	El usuario desde la escena deberá mover al personaje hasta acercarse a los personajes no jugables.

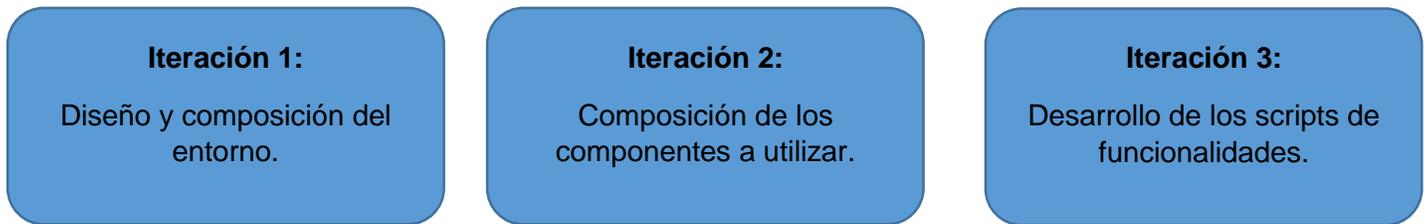
Tabla 16: MJ0003- Interacción con personajes

2.4 Implementación

En la subfase de implementación de la fase de Producción se llevarán a cabo dos elementos fundamentales: El diseño del entorno y la implementación de las mecánicas del juego.

Para la implementación del entorno se utilizarán assets liberados de la tienda online de Unity conocida como Assets Store y se modelarán algunos objetos específicos del juego. Para la implementación del videojuego se utiliza CSharp como lenguaje de programación. Estos scripts están clasificados en base a la función que cumplen en el videojuego conectándose entre sí y dando vida a los objetos que interactúan en los escenarios. Los GameObject están compuestos por un grupo de componentes que brinda Unity para darle determinadas propiedades a estos objetos y los scripts que dictarán el comportamiento de cada uno en

los diferentes casos. A continuación, se expondrán las diferentes iteraciones de esta subfase.



2.4.1 Iteración 1: Diseño y composición del entorno.

El diseño del escenario estuvo concebido como parte de un ambiente fantástico donde interactuaran animales y personas en el mismo mundo. El personaje principal saldría de una casa y su primer encuentro sería con un cocodrilo que le ayuda a cruzar el río. Como parte de la ambientación se hace uso de colores llamativos para llamar la atención del usuario y despierte su sentido de exploración por el terreno.

Para el diseño del entorno se utilizaron asset 3D de estilo PolyArt, un estilo muy de moda hoy en día por su simpleza y optimización en los videojuegos. Además, se crearon varios assets como herramientas para la jugabilidad. A continuación, se mostrarán algunas imágenes del diseño creado para la ambientación del videojuego.

El personaje antes de cada prueba tendrá que coleccionar algunos objetos antes de enfrentarse a una prueba nueva, esto estimula la curiosidad del jugador y la exploración, manteniendo los niveles de jugabilidad y diversión en un correcto balance evitando el aburrimiento del niño.

[Ver Figuras 4, 5, 6, 7, 8, 9, 10 y 11 de Diseño y Composición del Entorno.](#)

El jugador va recolectando puntos según la cantidad de respuestas correctas en el juego, al final a la hora de la calificación si está por encima de la media se le desbloquea un nivel de bono donde deberá salvar la villa de atacantes. Este nivel de bono es puramente enfocado al entretenimiento del niño, funcionando como premio después de haber realizado las tareas y respondido correctamente una cantidad considerable de preguntas.

2.4.2 Iteración 2: Componentes

Los componentes que provee Unity 3D para la interacción en las escenas se muestran en formas diversas. Pueden ser para crear comportamiento, definiendo apariencia, e influenciando otros aspectos de la función de un objeto en el juego. Los componentes comunes de producción de juego vienen contruidos dentro del Unity, desde el Rigidbody, hasta elementos más simples, como luces, las cámaras, los emisores de partículas y otros, que permiten estructurar de forma rápida un escenario. (UnityTechnologies, 2019)

Componentes de visión

- **Camera**

Las cámaras son los dispositivos que capturan y muestran el mundo al jugador. Mediante la personalización y la manipulación de cámaras, se puede hacer una presentación del juego verdaderamente única. El número de cámaras a utilizar en una escena es ilimitado y la configuración puede hacerse en cualquier orden, en cualquier lugar o sólo ciertas partes de la pantalla.

[Ver Figura 12. Componentes de Visión. Cámara.](#)

Componentes de audio

- **Audio Source:**

El componente Audio Source reproduce un clip de audio en la escena. Si el clip es un clip de audio 3D, la fuente se reproduce en una posición determinada y se atenúan con la distancia, esta puede ser controlada con las curvas de difuminación. Además, si el oyente está dentro de uno o múltiples zonas de reverberación estas se aplican a la fuente. Diversos filtros individuales se pueden aplicar a cada fuente de audio para una experiencia de audio aún más rica (Sólo PRO).

- **Audio Listener**

El *Audio Listener* actúa como un dispositivo de entrada de micrófono. Se recibe

información desde cualquier fuente de audio que figura en la escena y reproduce sonidos a través de los altavoces del ordenador. Para la mayoría de las aplicaciones tiene más sentido conectar este componente a la cámara principal. Si un detector de audio está dentro de los límites de una zona de reverberación se aplica a todos los sonidos audibles en la escena.

Componentes de colisiones.

•Box Collider

El *Box Collider* una colisión básica en forma de cubo primitivo. El *Box Collider* puede cambiar de tamaño en diferentes formas de prismas rectangulares. Funciona muy bien para las puertas, paredes plataformas, etc. También es eficaz como un torso humano en un muñeco de trapo o un casco de coche en un vehículo. Por supuesto, funciona perfectamente para las cajas. *Capsule Collider*

El *Capsule Collider* se compone de dos medias esferas unidas por un cilindro. Se trata de la misma forma que la cápsula primitiva. Puede ajustar radio y la altura del *Capsule Collider* independientemente unos de otros. Se utiliza en el controlador de caracteres y funciona bien para los polos, o se puede combinar con otros *Colliders* para formas inusuales.

•Sphere Collider

El *Sphere Collider* una colisión básica en forma de esfera primitiva. Puede ser redimensionada a escala uniforme, pero no a lo largo de cada eje. Funciona muy bien para la caída de piedras, pelotas de ping-pong y otros cuerpos en forma esférica.

•Character Controller

El *Character Controller* se utiliza principalmente para el control de personajes de tercera persona o en primera persona que no haga uso de física *Rigidbody*. Se trata simplemente de una cápsula que funciona para moverse en una dirección de un registro. El controlador entonces llevar a cabo el movimiento, si no es limitado por colisiones. Se deslizará a lo largo del suelo, sube las escaleras (si es que son más bajos que el Paso Offset) y pendientes dentro del límite de estas.

•Rigidbody

El *Rigidbody* permiten a los *GameObjects* actuar bajo el control de la física. El objeto que lo posea puede recibir fuerza y también ejercerla para que estos se muevan de una manera realista. Cualquier *GameObject* que contenga un *Rigidbody* va a estar influenciado por la gravedad, este actúa bajo fuerzas a través de secuencias de comandos, o interactúa con otros objetos a través del motor de física *PhysX* de *NVIDIA*.

[Ver Figura. 13: Componentes de colisión. Cuerpo Rígido.](#)

Componentes de animación

Cualquier *GameObject* que tiene un avatar también tendrá un componente animador, que es la relación entre el carácter y su comportamiento. Este componente referencia a componentes de un controlador animador que se utiliza para configurar el comportamiento del personaje. Esto incluye la instalación de máquinas de estado, se mezclan árboles, y los eventos que deben ser controlados desde el registro.

[Ver Figura 14: Componentes de animación.](#)

2.4.3 Iteración 3: Script de funcionalidades.

En este apartado se comenzará a trabajar en el desarrollo de script necesarios para dar solución a cada elemento dinámico (casos de uso según el software convencional) descrito en epígrafes anteriores. Para cada una de estas mecánicas se implementará uno o más script que cubran dicha funcionalidad en el juego. Pero antes de implementar existe la necesidad de buscar elementos para organizar el desarrollo y realizar un buen diseño de clases por lo cual se hace un estudio de los patrones de diseño de software y se deciden cuales se van a utilizar en el desarrollo. A continuación, se muestran estos resultados.

Patrones de Diseño

Los patrones de diseño son la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Son reusables ya que pueden ser aplicados en otros diseños o problemas.

Sus beneficios son muchos entre ellos podemos encontrar:

- Facilitan la reutilización de diseños y arquitecturas de software que han tenido éxito.
- Capturan la experiencia y la hacen accesible a los no expertos.
- Mejoran la adaptabilidad del sistema a futuros cambios, (orientado a cambios).
- Mejoran la comunicación entre los desarrolladores al ser identificados por nombres. (Contribuyen al lenguaje del desarrollador)
- Facilitan la rápida comprensión de un sistema al conocer los patrones empleados.
- Pueden ser usados como base para la documentación de la ingeniería de software.

(Blanco, Patrones de diseño y arquitectura de software , 2017)

Para la implementación de los scripts se tuvieron en cuenta los siguientes patrones *General Responsibility Assignment Software Patterns (GRASP)*: Experto, Bajo acoplamiento y Alta cohesión

Se hizo uso del patrón Experto, en la clase GameManager ya que esta clase es la encargada administrar todas las variables importantes del juego y de servir como flujo de información.

Beneficios:

El GRASP de experto en información es el principio básico de asignación de responsabilidades. Nos indica, por ejemplo, que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. De este modo obtendremos un diseño con mayor cohesión y así la información se mantiene encapsulada (disminución del acoplamiento). (Blanco, Patrones GRASP y principios SOLID. , 2017)

Se evidencia el uso del patrón Bajo Acoplamiento y Alta Cohesión, debido a que las clases fueron diseñadas de tal forma que existiera la menor relación entre ellas, a la vez logrando que la información almacenada en cada clase fuera coherente. Con este diseño se logra que de existir una modificación

tenga la mínima repercusión posible en el resto de las clases, se garantiza la reutilización de código y disminuye la dependencia entre las mismas.

Beneficios:

Los conceptos de cohesión y acoplamiento no están íntimamente relacionados, sin embargo se recomienda tener un mayor grado de cohesión con un menor grado de acoplamiento. De esta forma se tiene menor dependencia y se especifican los propósitos de cada objeto en el sistema. (Blanco, Patrones GRASP y principios SOLID. , 2017)

Teniendo los patrones a utilizar se puede combinar con la arquitectura de Unity, basada en componentes, para crear los scripts necesarios para implementar las mecánicas descritas. A continuación, se mostrará los scripts que se desarrollaron según la descripción de los elementos dinámicos.

Tarea	
Número de tarea: 1	ED: MJ0001
Nombre de la tarea: Implementación del sistema de locomoción del personaje	
Tipo de tarea: Implementación	Estimación: 2 días
Descripción: Se declaran los atributos configurables del componente de animación y establecerá una estrecha relación con el script PlayerMove.cs	

Tabla 17: Tarea 1 de la iteración 3.

Tarea	
Número de tarea: 2	ED: MJ0001
Nombre de la tarea: Implementación de la cámara del personaje	
Tipo de tarea: Implementación	Estimación: 1 días
Descripción: Se declaran los atributos configurables del componente de la cámara y se desarrolla el script SmoothFollow.cs para que siga los movimientos del personaje cierta distancia y altura.	

Tabla 18: Tarea 2 de la iteración 3.

Tarea	
Número de tarea: 3	ED: MJ0002
Nombre de la tarea: Implementación de la acción de recoger objetos.	
Tipo de tarea: Implementación	Estimación: 5 días
Descripción: Se declaran los atributos configurables de los componentes BoxCollider y Rigidbody. Una vez configurado los componentes se desarrolla el script PickObject.cs para interactuar con el objeto 'Player'.	

Tabla 19: Tarea 3 de la iteración 3.

Tarea	
Número de tarea: 4	ED: MJ0002
Nombre de la tarea: Implementación de la acción de interactuar con personajes no jugables.	
Tipo de tarea: Implementación	Estimación: 5 días
Descripción: Se declaran los atributos configurables de los componentes BoxCollider y Rigidbody. Una vez configurado los componentes se desarrolla el script AnimalInteract.cs para interactuar con el objeto 'Player' y funcionar como disparador para las preguntas.	

Tabla 20: Tarea 4 de la iteración 3.

Tarea	
Número de tarea: 5	ED: MJ0003
Nombre de la tarea: Implementación del sistema de preguntas y respuestas.	
Tipo de tarea: Implementación	Estimación: 5 días

Descripción: Se crea una base de conocimiento con las preguntas y posibles respuestas. Cuando el script AnimalInteract.cs decide que el objeto 'Player' posee los objetos suficientes lanza un panel con la pregunta y cuatro posibles respuestas seleccionadas aleatoriamente por el script Quest.cs. Si todas las preguntas son respondidas se habilita un objeto para continuar en la escena. Se recogen la cantidad de preguntas correctas que se seleccionaron.

Tabla 21: Tarea 5 de la iteración 3.

Conclusiones del capítulo

En este capítulo se mostraron diferentes momentos en la etapa de Producción según la metodología Huddle con sus artefactos, donde se definen los aspectos fundamentales del videojuego tanto técnicos como de diseño. Se logró organizar la implementación gracias a los patrones de diseño conjuntamente con la arquitectura basada en componentes de la plataforma Unity.

CAPÍTULO 3: Pruebas de la solución

No existe un método estándar para las pruebas de juego, y la mayoría de las metodologías son desarrolladas por desarrolladores de videojuegos individuales. Las metodologías se están continuamente refinando y las pruebas se llevan a cabo según el tipo de juegos. El sub-proceso de prueba o el testing es uno de los más importantes dentro del desarrollo de videojuegos, ya que a través de este se controla la calidad del videojuego. La función principal de las pruebas es el descubrimiento y documentación de los defectos de software. (JuegaLibre, 2019)

Después de concluido el proceso de desarrollo y la aplicación se encuentra en estado funcional se procede a ser probada por un equipo pequeño, que ha estado involucrado en el diseño y desarrollo del juego, en busca de errores para su refinamiento. Algunos de los aspectos a probar son la jugabilidad y el rendimiento.

3.1 Pruebas de Rendimiento.

Una vez concluido el videojuego queda propuesto para su correcto funcionamiento un ordenador con los siguientes requerimientos mínimos de hardware:

- Plataforma: Windows 7, 8 o 10.
- Memoria RAM: 1 GB.
- Tarjeta Gráfica: Integrada de 64 MB.
- Tarjeta de audio compatible con los formatos mp3 y OGG.

El equipo de desarrollo realizó pruebas en ordenadores con estas prestaciones y en otras de mayor procesamiento, con el objetivo de verificar el rendimiento del videojuego. Se tomó

en cuenta la cantidad de fotogramas por segundo a los que corría la aplicación sobre las diferentes computadoras para verificar la velocidad con que el sistema procesaba la información y las posibles causas de deficiencia. A continuación, se muestran los resultados obtenidos en esta prueba.

Prestaciones de la PC	Frames/seg obseados	Frames/seg total
<ul style="list-style-type: none"> • Intel Atom a 1.66 Gh • 1GB de RAM • Resolución: 1366x768 	19 FPS	30 FPS
Observaciones		
El sistema de iluminación usado en la escena disminuye considerablemente el rendimiento de la aplicación y el ordenador disminuye su rendimiento al utilizarse efectos de imagen avanzada.		
Recomendación		
Según las observaciones obtenidas de esta primera prueba se hacen necesario una mejora en el sistema de iluminación del primer escenario pues el exceso de uso de esta hace que disminuya el rendimiento, de igual modo el uso permanente de efectos de imágenes avanzado hace que el rendimiento también disminuya. En la segunda iteración se procedió al uso de lightmaps para mejorar el rendimiento de las luces en la escena.		

Prestaciones de la PC	Frames/seg obseados	Frames/seg total
<ul style="list-style-type: none"> • Dual Core a 2.1 Gh • 2GB de RAM • Resolución: 1024x768 	30 FPS	30 FPS
Observaciones		
Aunque no decaen los FPS las UI se muestra comprimida.		
Recomendación		
Según las observaciones obtenidas se crea una UI Canvas adaptable a cualquier resolución.		

Prestaciones de la PC	Frames/seg obseados	Frames/seg total
------------------------------	----------------------------	-------------------------

<ul style="list-style-type: none"> • Intel Atom a 1.66 Gh • 1GB de RAM • Resolución: 1366x768 	30 FPS	30 FPS
Observaciones		
El sistema acepta perfectamente la aplicación, siendo ejecutado a nivel medio de gráficos y la interfaz se adapta bien a la resolución de pantalla.		
Recomendación		
-		

Tabla 22: Pruebas de Rendimiento.

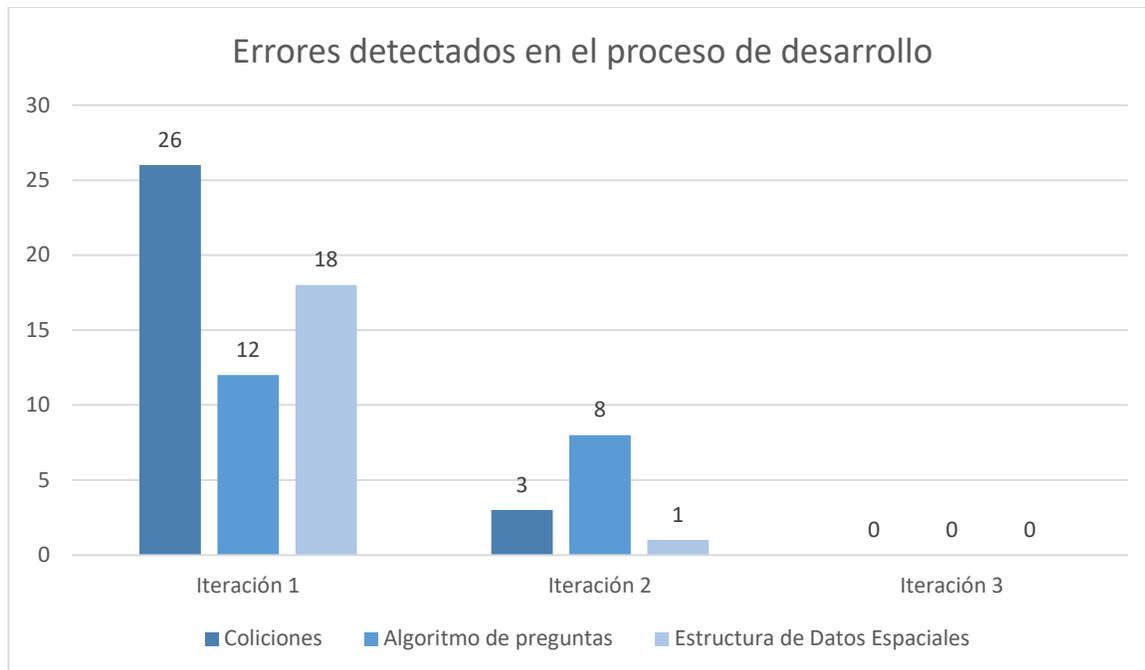
Una vez culminadas las pruebas de rendimiento y detectadas las deficiencias, se pasa a una segunda revisión del videojuego para corregir los errores tomados, y se vuelve a comenzar otra iteración de prueba para verificar la veracidad de las correcciones.

3.2 Pruebas unitarias

Son las pruebas diseñadas por los programadores y están enfocadas al código, consisten en verificar de manera manual o automatizada, si una parte específica del código, funciona de acuerdo con los requisitos del sistema. Deben ser definidas antes de realizar el código y repetirse hasta eliminar todos los errores para aumentar la calidad del desarrollo. (Kruchten, 2004)

Debido a que en la solución se emplea un gran número de procedimientos encargados de mostrar algún contenido visual (o contribuir a esto) y debido también a que son pocas las funciones que se usan, además de presentar una baja complejidad, se decidió realizar las pruebas unitarias de manera manual para adaptarlas a esta situación y verificar si este contenido visual se muestra correctamente. Se realizaron 3 iteraciones de prueba durante el desarrollo, para evitar el arrastre de errores lógicos a iteraciones posteriores. El sistema más complicado en los escenarios de prueba fue la detección de colisiones en el terreno.

En la siguiente grafica se muestra el resultado en cada iteración de prueba unitaria realizada. Los errores fueron solucionados en la última iteración.



Grafica 1: Errores detectados en el proceso de desarrollo

3.3 Valoraciones Económicas y Análisis de Costo

La estimación es el proceso de medición anticipada de la duración, esfuerzos y costes necesarios para realizar todas las actividades y obtener todos los productos asociados a un proyecto. Es necesario tener en cuenta numerosos aspectos que afectan a la estimación como la complejidad del proyecto, su estructuración, el tamaño, los recursos involucrados y los riesgos asociados. (Pressman, 2010)

El costo del proyecto considerando el salario mensual para el equipo de trabajo por el tiempo estimado de duración del proyecto, arrojando los siguientes resultados mediante la fórmula de Bohem:

Costo = CH * SM * TD (Pressman, 2010), donde:

CH: es la cantidad de hombres. La cantidad de desarrolladores en este caso equivale a 1.

SM: es el salario mensual por persona. Considerando un salario de 1000 pesos por el desarrollador.

TD: es el tiempo de desarrollo total estimado para el proyecto. El tiempo estimado para la realización del proyecto es de 20 semanas. Teniendo en cuenta que 4 semanas equivalen a 1 mes el proyecto se realiza en aproximadamente 5 meses.

$$\text{Costo} = 1 * 1000 * 5$$

$$\text{Costo} = 5000.$$

Se obtiene como costo final del proyecto 5000 pesos.

Conclusiones del capítulo

En este capítulo se evidencia la realización de varias pruebas a la aplicación para garantizar un funcionamiento óptimo. Aunque existen varios tipos de pruebas que se pueden realizar a un videojuego dependiendo del tipo y público se decidió realizar dos tipos de pruebas en diferentes momentos e iteración, las unitarias y las de rendimiento. Al final del ciclo de prueba se logró mitigar los errores correctamente y garantizar el funcionamiento correcto de la aplicación para utilizarla en futuras investigaciones.

Conclusiones

Después de haber desarrollado un videojuego en su ciclo completo, pasando desde las fases de planificación metodológica hasta las fases de implementación y prueba se llega a las siguientes conclusiones:

- 1- La utilización de la metodología Huddle fue una buena herramienta para guiar el proceso de desarrollo de un videojuego.
- 2- La plataforma Unity permitió crear un videojuego de forma fácil y rápida con un diseño llamativo para los niños.
- 3- Se desarrollaron mecánicas simples en el diseño lo que garantiza un rápido aprendizaje del infante.
- 4- Se realizaron dos tipos de pruebas, las unitarias y las de rendimiento, que resultaron fundamentales para la correcta culminación del videojuego.
- 5- Se desarrolló un videojuego que puede funcionar como herramienta para ejercitar el inglés en niños de 7 a 9 años de edad.

Recomendaciones

Después del desarrollo del videojuego y la investigación realizada se arrojan las siguientes recomendaciones:

- 1- Probar en aulas el juego para ver la evolución de los alumnos con el videojuego.
- 2- Mejorar el sistema de locomoción para futuras versiones.
- 3- Continuar la investigación pues es un desarrollo muy interesante para la creación de futuras herramientas.
- 4- Bono final también posea un remedial con el contenido no vencido.
- 5- Diálogos mejor pronunciados.
- 6- Sonidos al capturar los lápices.
- 7- La evaluación final que sea por contenido aprobado.

Bibliografía

- Adobe. (2019). *Adobe Photoshop*. Retrieved from <http://www.adobe.com/es/products/photoshop.html>
- Bethke, E. (2003). *Wordware Publishing*.
- Blanco, J. D. (2017). *Patrones de diseño y arquitectura de software* ., (p. 26).
- Blanco, J. D. (2017). *Patrones GRASP y principios SOLID*. .
- Blender*. (2019). Retrieved from blender.org: www.blender.org
- Brains Nursey Schools. (2019). *La importancia de aprender inglés desde la niñez*. Retrieved from Brains Nursey Schools: brainsnursery.com
- Cascales, F. (2007). *PROCESOS PSICOLÓGICOS BÁSICOS*. Universidad de Alicante.
- Clark, C. (1987). *Serious Games*. New York : Viking Press.
- Crawford, C. (2010). *The art of game design*. Retrieved from <http://www.vancouver.wsu.edu/fac/peabody/game-book/Coverpage.html>
- Flood, K. (2003). *Game Unified Process*. Retrieved from <http://www.gamedev.net/reference/articles/article1940.asp>
- JuegaLibre*. (2019). Retrieved from <http://juegalibre.virtual.uniandes.edu.co/index.php/2013/03/07/testing-prueba-de-juego/>
- Kruchten, P. (2004). *The Rational Unified Process: An Introduction*.
- Morales, J. (2013). *El diseño de serious games: una experiencia pedagógica en el ámbito de los estudios de Grado en Diseño*.

Morris, D., & Rollings, A. (n.d.). *Game Architecture and Design: A New Edition*. California: New Riders Publishing.

Pixlr. (2019). *Pixlr*. Retrieved from Pixlr: pixlr.com

Pressman, R. (2010). *Software Engineering*.

Scrum Alliance. (2019). Retrieved from http://www.scrumalliance.org/pages/what_is_scrum

The Photo Editor for Every. (2019). Retrieved from pixlr.com: pixlr.com

UCL. (2008). *Information behaviour of the researcher of the future*. London .

UnityTechnologies. (2019). *Unity 3D*. Retrieved from <http://unity3d.com>.

Zimmerman, S. (2003). *Rules of Play: Game Design Fundamentals*. MIT Press.

Zyda, M. (2005). From Visual Simulation to Virtual Reality to Game. *IEEE Computer Society*.

Anexos

Editor Visual de Unity 3D

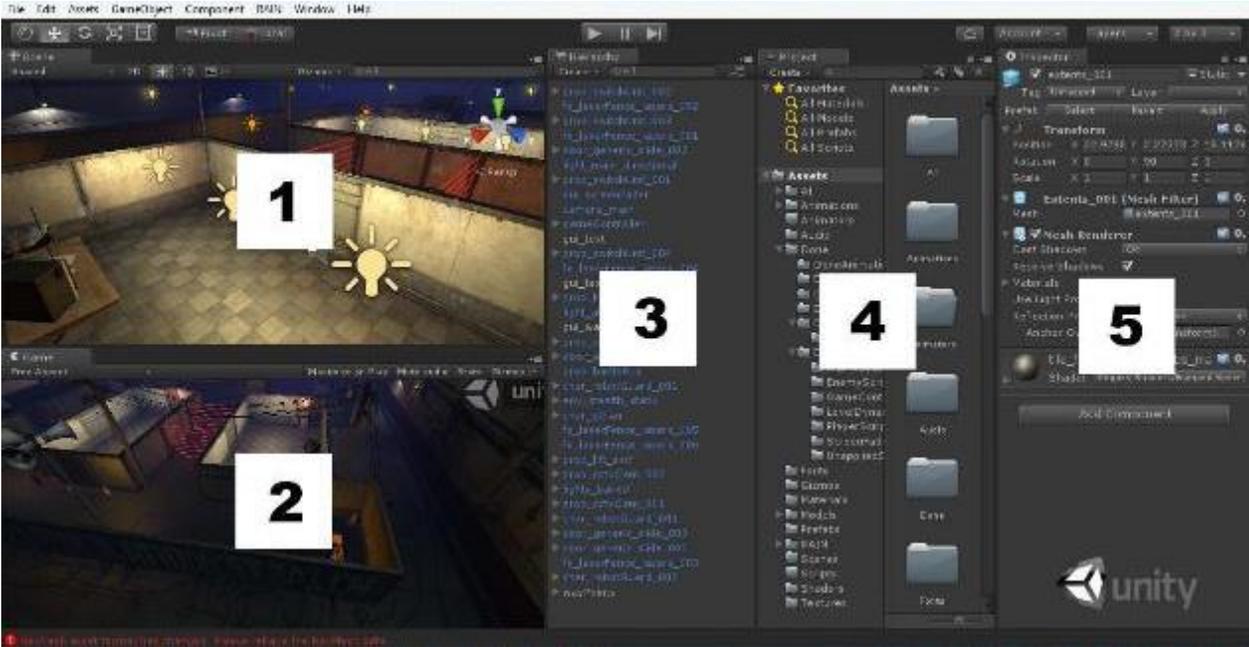


Fig. 3. Captura de pantalla de la interfaz del editor de Unity 3D.

Documento Diseño

Personajes secundarios

Cocodrilo

Nombre del Personaje

Cocodrilo

Descripción

Cocodrilo verde con expresión alegre y sombrero

Imagen



Encuentro

Es el personaje que hace las primeras preguntas al personaje principal.

Habilidades

Habilitar el puente para continuar el camino.

Niña

Nombre del Personaje	Niña
Descripción	Niña con motonetas que vive en la primera aldea del juego.
Imagen	
Encuentro	Este personaje se encuentra situado en la entrada de la aldea del juego. Te proporciona la pista de la persona que hará la próxima prueba.
Habilidades	Pistas.
Mamá de la niña	
Nombre del Personaje	Mamá de la niña

Descripción	Personaje femenino adulto con espejuelos.
Imagen	
Encuentro	Es el personaje que hace la segunda rueda de preguntas al personaje principal.
Habilidades	Abre el portón para continuar el camino.
Mago	
Nombre del Personaje	Mago
Descripción	Personaje anciano con toga y sombrero

<p>Imagen</p>	
<p>Encuentro</p>	<p>Es el personaje que hace la tercera rueda de preguntas al personaje principal.</p>
<p>Habilidades</p>	<p>Abre el portón para continuar el camino.</p>
<p>Sabio</p>	
<p>Nombre del Personaje</p>	<p>Sabio</p>
<p>Descripción</p>	<p>Personaje adulto masculino con ropas oscuras.</p>

<p>Imagen</p>	
<p>Encuentro</p>	<p>Es el personaje que hace la cuarta rueda de preguntas al personaje principal.</p>
<p>Habilidades</p>	<p>Abre el portón para continuar el camino.</p>
<p>Calabaza zombi</p>	
<p>Nombre del Personaje</p>	<p>Calabaza zombi</p>
<p>Descripción</p>	<p>Personaje con cabeza de calabaza fantasma</p>

<p>Imagen</p>	
<p>Encuentro</p>	<p>Este personaje aparece como parte del nivel de bono.</p>
<p>Habilidades</p>	<p>Quitar vida</p>

Diseño y composición del Entorno



Fig. 4: Composición de la casa.



Fig. 5: Composición del escenario de la primera prueba.



Fig. 6: Composición del escenario entre pruebas.



Fig. 7: Composición del escenario de Color Farm.



Fig. 8: Escenario de Color Farm.



Fig. 9: Composición del Valle de las Pelotas Saltarinas



Fig. 10: Composición del Bosque de los Sabios



Fig. 11: Composición del nivel de Bono

Componentes de Visión

Cámara

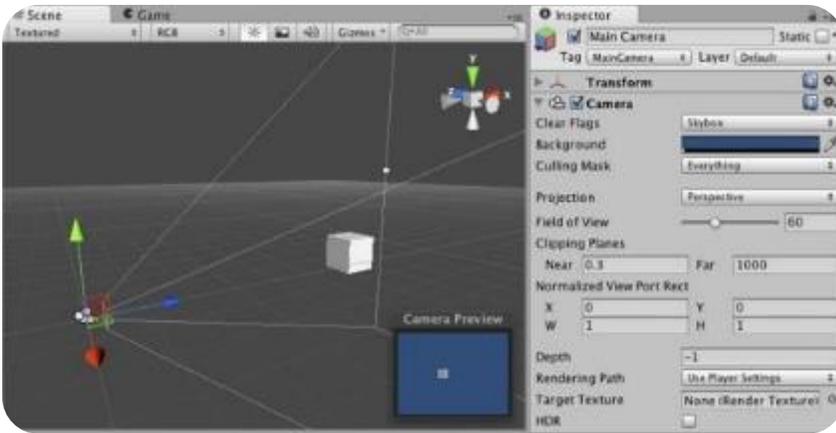


Fig. 12: Componentes de visión.

Componentes de Colisión

Cuerpo Rígido

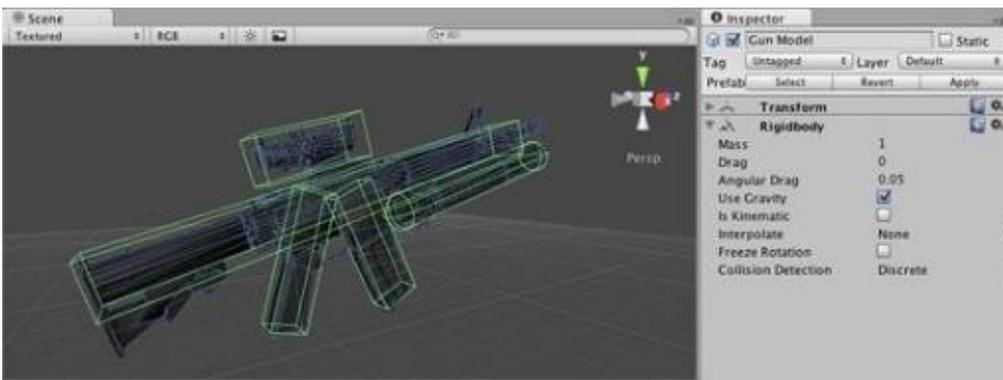


Fig. 13: Componentes de colisión.

Componentes de Animación

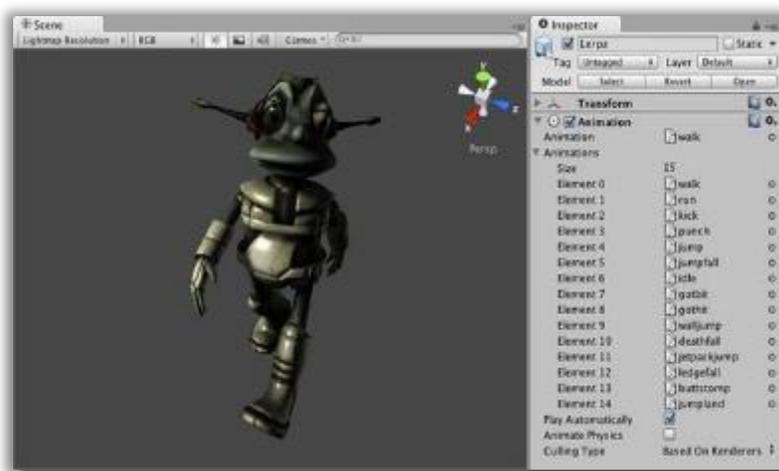


Fig. 14: Componentes de animación.