

**VIDEOJUEGO
CASTLE OF CHAOS**

CARLOS FELIPE MORENO CORTÉS

**CORPORACIÓN UNIVERSITARIA MINUTO DE DIOS
FACTULTAD DE INGENIERÍA
DEPARTAMENTO DE INFORMÁTICA Y ELECTRÓNICA
PROGRAMA DE TECNOLOGÍA EN INFORMÁTICA
BOGOTÁ
II-2015**

**VIDEOJUEGO
CASTLE OF CHAOS**

CARLOS FELIPE MORENO CORTÉS

**Trabajo de grado para optar al título de
Tecnólogo en Informática**

**Asesor:
Ing. Miguel A. Hernández Bejarano**

**CORPORACIÓN UNIVERSITARIA MINUTO DE DIOS
FACTULTAD DE INGENIERÍA
DEPARTAMENTO DE INFORMÁTICA Y ELECTRÓNICA
PROGRAMA DE TECNOLOGÍA EN INFORMÁTICA
BOGOTÁ
II-2015**

Nota de aceptación

Firma del presidente del jurado

Firma del jurado N°1

Firma del jurado N°2

Bogotá D.C. Noviembre 2015

Dedicatoria y Agradecimientos

Quiero agradecer en este espacio a todas las personas que hicieron posible que este proyecto se realizara, sobre todo a mi familia que me acompañó durante todo el proceso y mantuvo su apoyo en todo momento, que en las dificultades confió en mis habilidades y en la educación que me brindaron para seguir el camino correcto. Ellos sobretodo son la razón por la que estoy en donde estoy y como todo triunfo en mi vida se los dedico humildemente.

Por supuesto a la universidad por prestar una educación de calidad y que definitivamente ayudo a guiar mi camino hacia un futuro prometedor en el sector de los videojuegos, materia en la que puedo ver toda mi pasión y en lo que pondré todo mi esfuerzo.

Tabla de contenido

INTRODUCCIÓN	12
1.1 Título del proyecto	13
1.2 Planteamiento del problema	13
1.3 Alcance y justificación	14
1.4 Objetivos	15
1.4.1 Objetivo general	15
1.4.2 Objetivos específicos	15
2. Ingeniería del proyecto	16
2.1 Modelo de desarrollo	16
2.1.1 Preproducción	24
2.1.2 Producción	25
2.1.3 Postmortem	26
2.1.4 Roles	27
2.1.5 Documento de diseño	27
2.2 Preproducción	35
2.2.1 Título	35
2.2.2 Estudio/Diseñador	35
2.2.3 Genero	36
2.2.4 Plataforma	37
2.2.5 Versión	37
2.2.6 Sinopsis/Jugabilidad/Contenido	38
2.2.7 Categoría	39
2.2.8 Licencia	39
2.2.9 Mecánica	40
2.2.10 Tecnología	41
2.2.11 Publico	42
2.2.12 Mecánica/Controles	43
2.2.13 Cámara	44
2.2.14 Periféricos	44
2.2.15 Puntaje	45
2.2.16 Guardar/Cargar	45
2.2.17 Sprint plan	46
3. Análisis y diseño	58

3.1 Definición de requerimientos	58
3.1.1 Requerimientos funcionales	58
3.1.2 Requerimientos no funcionales	59
3.2 Descripción del sistema propuesto	60
3.2.1 Interfaces	61
3.3 Diseño del sistema propuesto	64
3.3.1 Diagramas estáticos	64
3.3.1.1 Diagramas de clases	64
3.3.2 Diagramas dinámicos	66
3.3.2.1 Estados del juego	66
3.3.2.2 Diagrama de estados	67
3.3.2.3 Diagramas de casos de uso	69
3.3.2.4 Diagrama de secuencia	77
4. Desarrollo	81
4.1 Especificaciones técnicas	81
4.1.1 Hardware	81
4.1.2 Software	81
5. Glosario	82
6. Conclusiones	84
7. Bibliografía	85
8. Manuales	86
8.1 Manual del sistema	86
8.1.1 Interfaz	86
8.2 Manual del usuario	88
8.2.1 Personajes	88
8.2.2 Niveles	90
8.2.3 Enemigos	93
8.2.4 Habilidades	93
8.2.5 Objetos	94
8.2.6 Trucos	94
8.2.7 Guion	94
8.2.8 Datos de producción	96

Lista de tablas

Tabla N°1: Procesos viables al desarrollo de videojuegos	19
Tabla N°2: Waterfall vs Scrum, ventajas	21
Tabla N°3: Waterfall vs Scrum, desventajas	22
Tabla N°4: Plantilla documento de diseño – Concepto	28
Tabla N°5: Plantilla documento de diseño – Historial de versiones	29
Tabla N°6: Plantilla documento de diseño – Visión general del juego	29
Tabla N°7: Plantilla documento de diseño – Mecánica del juego	29
Tabla N°8: Plantilla documento de diseño – Estados del juego	30
Tabla N°9: Plantilla documento de diseño – Interfaces	30
Tabla N°10: Plantilla documento de diseño – Niveles	31
Tabla N°11: Plantilla documento de diseño – Progreso del juego	31
Tabla N°12: Plantilla documento de diseño – Personajes	32
Tabla N°13: Plantilla documento de diseño – Enemigos	33
Tabla N°14: Plantilla documento de diseño – Habilidades	33
Tabla N°15: Plantilla documento de diseño – Objetos	33
Tabla N°16: Plantilla documento de diseño – Guion	33
Tabla N°17: Plantilla documento de diseño – Logros	33
Tabla N°18: Plantilla documento de diseño – Códigos Secretos	33
Tabla N°19: Plantilla documento de diseño – Música y sonidos	34
Tabla N°20: Plantilla documento de diseño – Imágenes de concepto	34
Tabla N°21: Plantilla documento de diseño – Miembros del equipo	34
Tabla N°22: Plantilla documento de diseño – Detalles de producción	34
Tabla N°23: Versión	37
Tabla N°24: Sprint 1	46
Tabla N°25: Sprint 2	47
Tabla N°26: Sprint 3	47
Tabla N°27: Sprint 4	48
Tabla N°28: Sprint 5	48
Tabla N°29: Sprint 6	49
Tabla N°30: Sprint 7	49
Tabla N°31: Sprint 8	50
Tabla N°32: Sprint 9	50
Tabla N°33: Sprint 10	51
Tabla N°34: Sprint 11	51

Tabla N°35: Sprint 12	52
Tabla N°36: Sprint 13	52
Tabla N°37: Sprint 14	53
Tabla N°38: Sprint 15	53
Tabla N°39: Sprint 16	54
Tabla N°40: Sprint 17	54
Tabla N°41: Sprint 18	55
Tabla N°42: Sprint 19	55
Tabla N°43: Sprint 20	56
Tabla N°44: Sprint 21	56
Tabla N°45: Sprint 22	57
Tabla N°46: Sprint 23	57
Tabla N°47: Diagrama caso 1	70
Tabla N°48: Diagrama caso 2	71
Tabla N°49: Diagrama caso 3	72
Tabla N°50: Diagrama caso 4	73
Tabla N°51: Diagrama caso 5	74
Tabla N°52: Diagrama caso 6	75

Lista de figuras

Figura N°1: Fases Huddle	23
Figura N°2: Preproducción	24
Figura N°3: Producción	25
Figura N°4: Postmortem	26
Figura N°5: Logo	36
Figura N°6: Metroid clásico	37
Figura N°7: Castlevania clásico	37
Figura N°8: Plataforma Pc	42
Figura N°9: Movimiento horizontal	43
Figura N°10: Movimiento salto	43
Figura N°11: Movimiento ataque	43
Figura N°12: Cámara 2D	44
Figura N°13: Teclado	44
Figura N°14: Control	44
Figura N°15: Victoria	45
Figura N°16: Guardar/Cargar	45
Figura N°17: Interfaz menú	61
Figura N°18: Interfaz escenario	62
Figura N°19: Interfaz escenario2	62
Figura N°20: Interfaz utilidades	63
Figura N°21: Diagrama clases – Videojuego	65
Figura N°21: Diagrama estado – Menú	67
Figura N°22: Diagrama estado – Movimiento	68
Figura N°23: Diagrama estado – Interacción	69
Figura N°24: Caso – Iniciar juego	70
Figura N°25: Caso – Iniciar nueva partida	71
Figura N°26: Caso – Cargar partida	72
Figura N°27: Caso – Jugar	73
Figura N°28: Caso – Cambiar opciones	74
Figura N°29: Caso – Salir	75
Figura N°30: Casos de uso videojuego	76
Figura N°31: Diagrama de secuencia 1 – Iniciar juego	77
Figura N°32: Diagrama de secuencia 2 – Nueva partida	78
Figura N°33: Diagrama de secuencia 3 – Cargar partida	79

Figura N°34: Diagrama de secuencia 4 – Cambiar opciones	80
Figura N°35: Diagrama de secuencia 5 – Salir	80
Figura N°36: Boceto interfaz menú	86
Figura N°37: Boceto interfaz escenario	87
Figura N°38: Escenario1	91
Figura N°39: Escenario2	91
Figura N°40: Escenario3	92
Figura N°41: Escenario4	92
Figura N°42: Código Konami	94
Figura N°43: Boceto guion	95

Resumen

Castle Of Chaos, un videojuego donde el personaje principal se adentra en una mansión que desde hace un tiempo se cree es el lugar al que llevan a toda la gente secuestrada de un pueblo alejado de la civilización, sin saber el cómo o el porqué de la situación, se tomara el papel de un viejo ex investigador que tendrá que volver a usar sus habilidades y aprender algunas para derrotar al mal que pretende volver a este mundo. Los jugadores se encontrarán con miles de enemigos diferentes, pruebas de agilidad y rompecabezas de todo tipo. Todo esto bajo un entorno 2D con escenarios llenos de bonitos detalles gráficos y una historia que mantendrá al jugador con ganas de saber más. Siendo el primer videojuego que la empresa desarrolla, se logró en gran proporción cada uno de los objetivos planeados, entregando un producto de calidad y accesible al público cómodamente. El proyecto usó en todo el desarrollo la metodología ágil Huddle, que como metodología experimental específica para videojuegos fue perfecta para su aplicación en este proyecto independiente, permitiendo un rápido desenvolvimiento de cada tarea y ofreciendo el respaldo teórico suficiente, tema importante para futuros proyectos.

Abstract

Castle of Chaos, a game where the main character goes into a mansion some time believed to be the place where all people carrying kidnapped from a village far from civilization, without knowing how or why the situation, the role of an old former researcher who will have to use their skills and relearn some to defeat the evil that seeks to return to this world was made. The players will encounter thousands of different enemies, agility and puzzles of all kinds. All this under a 2D environment with scenarios full of nice touches graphics and a story that will keep the player wanting to know more. It is the first game that the company develops, large proportion was achieved in each of the planned objectives, delivering a quality product and easily accessible to the public. The project used throughout the development agile methodology Huddle, which as game specific experimental methodology was perfect for application in in this independent project, allowing fast development of each task and providing sufficient theoretical backing, important topic for future projects.

Introducción

El presente proyecto busca el desarrollo de un videojuego que recuerde tanto a jugadores como a las grandes industrias que la diversión tiene que ser el punto inicial para tener en cuenta al trabajar la idea de cualquier juego, que además permita retar a nuestra propia habilidad para superar una y otra vez los obstáculos que se encontraran en cada escenario.

El videojuego tiene un enfoque ágil que proporcionara un reto al jugador, además de una historia interesante que llamara la atención de aquellos que sean conocedores del género y aun así, jugadores más contemporáneos se verán interesados en ella. Se busca demostrar como una buena combinación de elementos tanto artísticos como mecánicos y un guion a la altura, atrae la atención de muchos jugadores, pasando por encima aun de juegos triple A y como punto final divertir al usuario.

Así mismo se implementa una metodología específica para videojuegos, llamada Huddle, se explica de donde proviene, ventajas, desventajas, y por qué es tan importante como metodología única usada en esta industria, además de los roles que deben ser aplicados. En el siguiente proyecto se expone el planteamiento para el desarrollo del producto, los resultados obtenidos en el proceso y la importancia de estos para futuras referencias.

El documento actual abarca todo el proceso para el desarrollo del videojuego, desde el planteamiento de la idea y su posterior implementación, así mismo se presenta con imágenes y gráficos la forma en que este funciona. También se puede encontrar un apartado con información relevante al software en el que fue desarrollado (Construct2).

Finalmente se entrega un proyecto completo y un prototipo funcional, junto a un manual de instrucciones con cosas básicas, bocetos previos a la implementación del proyecto, el guion utilizado en la historia y un resumen tratando la música que fue usada para ambientar el juego.

1.1 Título del proyecto

Castle of Chaos, Videojuego.

1.2 Planteamiento del problema

Partiendo de la opinión de un conocedor de videojuegos y que trata de guiar su carrera hacia el desarrollo de los mismos, ¹ve continuamente como las nuevas tecnologías y la falta de ideas ha hecho que, aunque los juegos visualmente se vean mucho más sorprendentes y detallados (Llegando a superar al cine en estos aspectos) a la vez se han vuelto menos divertidos y mucho más fáciles que años atrás (Observar referencia numero). Bajo esta primicia se desarrolló un videojuego con toques muy clásicos, donde la habilidad del jugador, tanto mental como mecánica, sea esencial para avanzar por cada pantalla.

Este conjunto de cosas busca llamar la atención de los que disfrutaban de la dificultad de completar un juego, de quienes veían en algo simple gráficamente, todo un reto de habilidad. Actualmente algo indispensable y que los juegos han venido perdiendo es el factor “Diversión” que debería ser fácilmente lo más básico al desarrollar una idea de este tipo, esto gracias a que la creatividad se está viendo superada por el uso de los recursos tecnológicos. Castle of Chaos toma cada punto positivo de toda una generación de videojuegos para plasmarlo en un solo concepto que llame la atención de las personas que buscan un reto de verdad.

¹ Gambin Redondo Alfonso. BlastingNews (2014). Videojuegos encima del cine y la música. Disponible en la dirección. <http://es.blastingnews.com/economia/2014/12/el-videojuego-lidera-la-industria-del-ocio-audiovisual-superando-al-cine-o-la-musica-00217253.html>

El concepto original del juego nos muestra una problemática moderna de los videojuegos, ²las generaciones actuales son incapaces de usar sus habilidades para juegos con una dificultad que tiempo atrás era el estándar, este juego ofrece una experiencia total y promueve el aprendizaje de mecánicas de diferentes estilos, ya que como se mencionará más adelante, este estilo de juegos no se define por tener un estándar en cuanto a su jugabilidad, sino que incorpora una suma de varios estilos, así el jugador tendrá que hacer uso de más de una habilidad para seguir su camino, todo esto teniendo en cuenta también los avances en cuanto a gráficos y desarrollo de las diferentes tecnologías en los videojuegos.

1.3 Alcance y justificación

Desarrollado en HTML5, con esto se busca que cualquier dispositivo con acceso a un navegador web sea capaz de disfrutar del juego, diseñado en una perspectiva de dos dimensiones; el juego no cuenta con diferentes niveles sino con un nivel de gran tamaño estructurado en diferentes zonas únicas en un castillo en donde se desarrollaran todos los hechos de la historia, cuenta con una escala de dificultad (Los primeros escenarios serán mucho más fáciles que los últimos.) donde se tiene también un escenario básico a modo de tutorial. El juego tiende a tener pinceladas de RPG (Role Playing Game.) por lo que el protagonista a medida que transcurre la historia ira ganando nuevas habilidades, nuevas armas y sus características en general irán ganando experiencia.

En cuanto a enemigos, básicamente todos son monstruos, cada uno con diferentes habilidades, fortalezas y debilidades que el jugador tiene que descubrir, en el desarrollo de la historia también hay jefes de zona que serán un reto a vencer. Dentro de los aliados únicamente se contará con algunos NPCs (Non-Player Character.) que se encontraran dentro del castillo en algunos lugares y tendrán

² Sopitas (2013). Jugadores actuales vs jugadores clásicos. Disponible en la dirección. <http://www.sopitas.com/233725-videojugadores-actuales-no-pueden-pasar-el-nivel-1-1-de-super-mario-bros/?from=site>

como fin, ya sea guiar, contar algo de historia o vender objetos; estos objetos se compran con oro que los monstruos dejan al ser eliminados.

Como fue planteado antes, lo más importante es la diversión que el juego proporciona y bajo todo lo expuesto en párrafos anteriores, poder asegurar que la persona se tome un tiempo para entender las mecánicas, disfrutarlas y pasar una muy buena experiencia. El juego está orientado principalmente a jugadores experimentados, aunque no se descarta que atrape a un buen sector de jugadores casuales buscando algo nuevo.

En este proyecto se conceptualiza la idea hasta un prototipo funcional, que expone cada una de las ideas argumentadas en este documento y servirán como base sólida para posteriormente realizar un juego de mayor envergadura que será el primer proyecto con el que el estudio desarrollador incursionará en el mercado por medio de financiación externa o su aprobación en plataformas de venta de videojuegos.

1.4 Objetivos

1.4.1 Objetivo general:

Desarrollar un videojuego que, con su contenido y estilo, divierta.

1.4.2 Objetivos específicos:

- Escribir una historia que permita al jugador mayor inmersión en el juego.
- Diseñar un mapa extenso totalmente interconectado mediante diferentes salas.
- Desarrollar diferentes habilidades y objetos para el uso del personaje.
- Escribir un guion interesante, que permita al jugador interactuar con cada ser que encuentre en el escenario.

2. Ingeniería del Proyecto

Dentro de la rama de la ingeniería que se encarga de estudiar la realización de proyectos, comprende la concepción, planteamiento, organización, diseño, y verificación de un proceso. En este caso particular se analiza la metodología propuesta para desarrollar el videojuego, las fases en que se dividirá y el estudio del ciclo de vida de este.

2.1 Modelo de Desarrollo:

Durante muchos años las empresas desarrolladoras de videojuegos se han visto muy aferradas a la metodología cascada y aun en el presente algunas siguen este modelo para sus productos. Lo anterior debido a que en años anteriores no existía un proceso específico público (Existen algunos de forma privada.) para esta industria, lo cual claramente indico que había una necesidad de un modelo derivado de las metodologías ágiles que se acoplara perfectamente con el mundo de los videojuegos y ayudara en su proceso de creación.

Durante la planeación de este proyecto, este punto fue muy complicado de desarrollar, ya que, pese a que la universidad hace la tarea de enseñar adecuadamente cada uno de los modelos de desarrollo para aplicaciones, fue necesario una investigación más profunda por que los métodos conocidos en realidad ofrecen más de lo necesario o simplemente hacen complejo un proceso que en los videojuegos es intuitivo. En los siguientes párrafos se observa el cómo plasmar una idea que buscar divertir, sin recurrir a una base teórica innecesaria.

³Considerando que un videojuego es básicamente el desarrollo de una aplicación, la búsqueda de un método nuevo inicio en los procesos básicos existentes en el

³ Morales Urrutia Gerardo Abraham, Nava López Claudia Esther, Fernández Martínez Luis Felipe, Rey Corral Mirsha Aarón. Universidad Autónoma de Ciudad Juárez de México (2010). Metodología Huddle. Disponible en la dirección: www.fadp.edu.co/publicaciones/proyinvmul-03.docx

desarrollo de software y fue el punto de partida para suponer un nuevo método que fuera lo suficientemente ágil para esta clase de productos y no requiriera una cantidad innecesaria de recursos teóricos y brindara a los desarrolladores la posibilidad de elegir como querían trabajar. Los métodos viables considerados son los siguientes:

Proceso	Descripción	Valoración
<i>Waterfall Process</i>	<i>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).</i>	<i>El producto final se demora más de lo esperado ya que cualquiera problema que se presenta en una de las etapas se tiene que regresar a una anterior para corregirlo. Requiere hacer muchos cambios a los documentos y es propicio al desorden, sin embargo, es uno de los procesos más populares en la industria de los videojuegos.</i>
<i>Essential Unifed Process</i>	<i>Consiste en integrar los practicas acertadas que son recursos de los tres campos principales del proceso: el campo del proceso unificado, métodos ágiles y el campo de madurez del proceso. Cada uno de ellos apoya a capacidades diferentes. Se centra en las practicas esenciales que se cree deben tener todos los</i>	<i>Promueve un buen trabajo en equipo y está preparado para separar el trabajo creativo del mecánico, separa los artefactos de alfa y beta, siendo alfa lo más importante en el proyecto. Es posible separar las prácticas del proceso y adaptarlas al desarrollo de videojuegos.</i>

	<p>proyectos de desarrollo. (Jacobson 2009).</p>	
<p>OpenUp</p>	<p>Es un proceso mínimo y suficiente, lo que significa que solo el contenido fundamental y necesario es incluido. La mayoría de sus elementos están declarados para fomentar el intercambio de información entre los equipos de desarrollo y mantener un entendimiento compartido del proyecto, sus objetivos, alcance y avances.</p>	<p>La descripción del proceso insiste mucho en la colaboración en equipo y la conclusión de stakeholders como parte del proceso. Es de cierta manera similar a Scrum.</p>
<p>Team Software Process</p>	<p>Es un conjunto de procesos estructurados que indican que hacer en cada fase del desarrollo del proyecto y muestra como conectar cada fase para construir un producto completo (Humphrey 2000). El objetivo principal es completar con éxito a través de varios ciclos de desarrollo incremental, un pequeño proyecto de software con calidad, siguiendo fielmente el proceso y mantenimiento durante cada ciclo.</p>	<p>Proporciona un balance entre proceso, producto y equipo de trabajo. Sus fases y tareas están bien definidas. Nos enseña los procedimientos para iniciar un proyecto, los pasos para poder guiarlo y nos muestra como analizar y reportar los datos obtenidos. Se realiza un documento Postmortem por cada ciclo.</p>

<p><i>Microsoft Solution Framework</i></p>	<p><i>Es una serie de principios, modelos, disciplinas, conceptos y guías para diseñar aplicaciones de Microsoft (Kaeton 2006). Consiste en una serie de pequeños ciclos e iteraciones. Este modelo permite el desarrollo rápido con aprendizaje y refinación continúa debido al entendimiento progresivo de los requerimientos de los clientes. Metodología pesada y ágil.</i></p>	<p><i>Su filosofía es que no hay una sola estructura o proceso que se aplica óptimamente a los requerimientos y ambientes de todo tipo de proyecto, por lo tanto, se puede adaptar y soportar cualquier proyecto sin importar el tamaño o complejidad y teniendo una serie de principios y perspectivas que podrían ser adaptables al proceso de desarrollo de un videojuego. Insiste en la importancia de la contante retroalimentación.</i></p>
<p><i>Scrum Framework</i></p>	<p><i>Modelo iterativo-incremental utilizado en el desarrollo de software ágil. El trabajo está estructurado en ciclos conocidos como sprints. Durante cada uno de estos los equipos toman los requerimientos de una lista ordenada por prioridades conocidas como historias de usuario. Al terminar cada uno se tiene una versión potencialmente final del producto.</i></p>	<p><i>Permite a los equipos entregar características pulidas para probar la calidad del juego a lo largo de su desarrollo y así incorporar, la retroalimentación. Se considera ideal para los videojuegos debido a que cada vez involucra a más personas multidisciplinarias en el desarrollo.</i></p>

Tabla N°1: Procesos viables al desarrollo de videojuegos.

Después de evaluar cada uno de los métodos expuestos se llegó a la conclusión de que era necesario optar por un proceso nuevo, capaz de fusionar las características de algunos existentes, dando así un enfoque principal al Waterfall Process y a Scrum Framework. En la siguiente tabla se analiza las ventajas y desventajas de los dos métodos escogidos:

Ventajas	
<i>Waterfall</i>	<i>Scrum</i>
<i>La planificación es sencilla.</i>	<i>Incremento en la productividad.</i>
<i>Si se conoce el total de requerimientos desde el principio la calidad del videojuego es alta.</i>	<i>El producto total se convierte en una serie de pequeños pedazos manejables.</i>
<i>Se tiene todo bien organizado.</i>	<i>Existe un progreso, incluso si los requerimientos no están bien definidos.</i>
<i>No se mezclan las fases.</i>	<i>Todo el equipo tiene visibilidad de cada cosa del proyecto.</i>
<i>Es perfecto para aquellos videojuegos que son rígidos y se conoce muy bien la herramienta a utilizar.</i>	<i>Existe mucha comunicación en el equipo.</i>
	<i>El equipo comparte los éxitos desde el principio hasta el final.</i>
	<i>El cliente se mantiene informado en cada mejora del producto.</i>
	<i>Entrega de un producto funcional al finalizar cada ciclo.</i>
	<i>Posibilidad de ajustar la funcionalidad en base a las exigencias de los jugadores.</i>

	<i>Visualización del videojuego día a día.</i>
	<i>Alcance acotado y viable.</i>
	<i>Equipos integrados y comprometidos con el desarrollo del videojuego, toda vez que ellos definieron alcance y se auto-administran.</i>
	<i>Capacidad para aceptar modificaciones sobre la marcha.</i>
	<i>Prioridades a características del videojuego gracias al Product Backlog.</i>

Tabla N°2: Waterfall vs Scrum, ventajas.

Desventajas	
<i>Waterfall</i>	<i>Scrum</i>
<i>La duración de todo el ciclo es muy larga.</i>	<i>No genera toda la evidencia o documentación de otras metodologías.</i>
<i>Probabilidad alta de fracaso dado que existe poca comunicación con el usuario final.</i>	<i>Tal vez sea necesario complementarlo con otros procesos ágiles, como Xp.</i>
<i>El mantenimiento se realiza en el código fuente.</i>	<i>Un mal uso de la metodología puede dar lugar a un desarrollo sin final.</i>
<i>Las revisiones de videojuegos de gran complejidad son muy difíciles.</i>	<i>Si no se tiene experiencia en seguir procesos de desarrollo puede hacer caótico su uso.</i>
<i>Impone una estructura de gestión de proyectos.</i>	

<i>Para que el videojuego tenga éxito deben desarrollarse todas las iteraciones.</i>	
<i>Si se cambia el orden de las fases, el final será de menor calidad.</i>	
<i>Se retrasa la localización y corrección de errores.</i>	
<i>Puede producir videojuegos poco llamativos para los jugadores ya que no se le puede hacer muchas modificaciones según la marcha.</i>	
<i>Inflexibilidad del modelo: Dificultad para responder a cambios.</i>	

Tabla N°3: Waterfall vs Scrum, desventajas.

Teniendo en cuenta los dos métodos, nace Huddle, el cual será usado para el desarrollo del proyecto y es un método pensado por Gerardo Abraham Morales Urrutia, Claudia Esther Nava López, Luis Felipe Fernández Martínez, y Mirsha Aarón Rey Corral del instituto de Ingeniería y Tecnología de Universidad Autónoma de Ciudad Juárez/México en el 2010. Este sigue la analogía de Scrum, y viene del nombre que se les da a las reuniones antes de cada jugada en el fútbol americano. Su filosofía establece que, mediante breves reuniones a corto plazo, el equipo planea su “Jugada” a iniciar. De esta forma cada ciclo es pequeño y permite hacer correcciones tempranas a posibles desviaciones.

Pese a ser una metodología joven, ya ha sido utilizada por diferentes estudios del sector, dándole un nombre en constante crecimiento sobre todo entre las empresas independientes, además se observó que muchos de los proyectos de grados basados en videojuegos en Latinoamérica están tomando esta metodología como base. En los siguientes fragmentos se expone la forma en que Huddle divide el trabajo en diferentes fases, como plantea los roles y se exponen las características de las que el proyecto actual hace uso.

Huddle es el proceso específico para el desarrollo de videojuegos.

- Ágil.
- Optimo.
- Permite equipos pequeños.
- Iterativo.
- Incremental.
- Evolutivo.

Todo el proceso se divide a su vez en tres fases:

- Preproducción.
- Producción.
- Postmortem.



Figura N°1: Fases Huddle. Imagen autoría de Metodología Huddle.

2.1.1 Preproducción:

Tiene como objetivo principal, migrar la idea del diseñador al Feature Log y al Sprint Plan; estos documentos darán la pauta a la planeación y producción del videojuego.

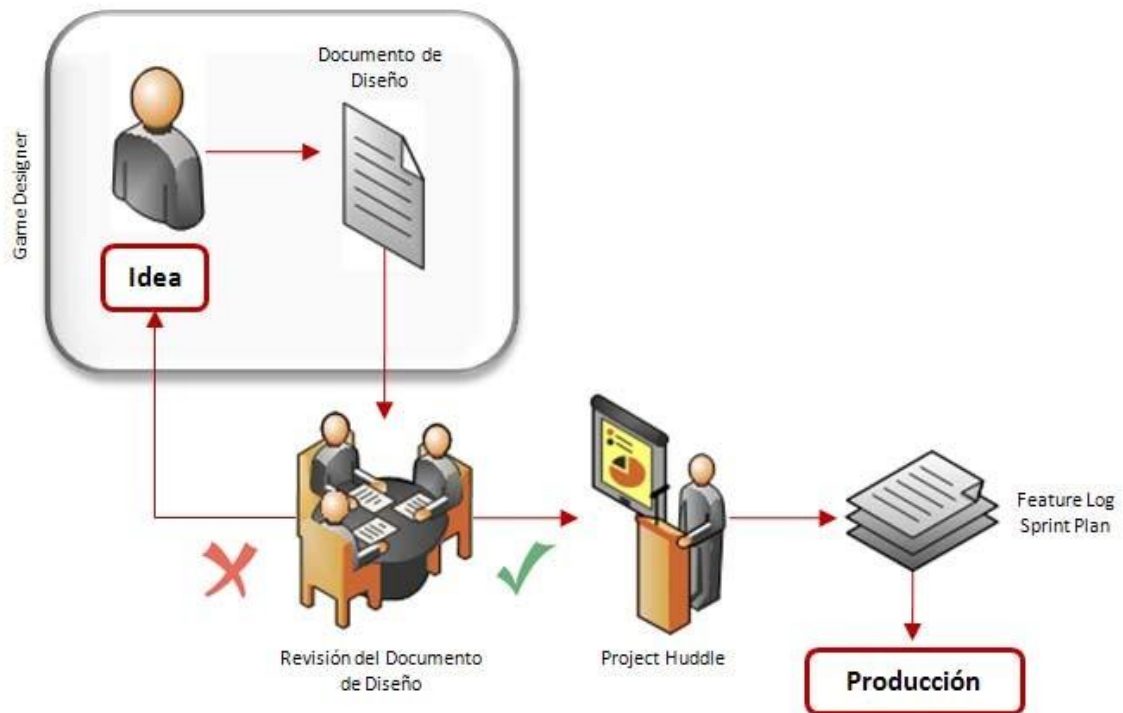


Figura N°2: Preproducción. Imagen autoría de Metodología Huddle.

Entre sus objetivos se encuentra el análisis del proyecto, fase en la que se revisará y se le dará aceptación, en este caso la idea inicial y la planeación completa de la fase de producción. Deben estar todos los miembros del proyecto involucrados.

Inicialmente se parte de un documento de diseño que expresa formalmente la idea principal y detalles de la propuesta del videojuego, este se debe revisar para analizar su factibilidad, sino es aceptado se puede modificar hasta encontrar su aprobación o en dado caso el rechazo definitivo. Luego de su aprobación el proyecto

pasa al Project Huddle el cual hará la planeación completa para iniciar la producción. Antes de terminar la fase ya se debe contar con el Feature Log y con un Sprint Plan.

2.1.2 Producción:

La producción es la etapa más importante y la más larga, ésta está ligada directamente con el *Scrum Framework* y sus herramientas como el *Daily Meeting*, *Sprint Reviews* y en sus artefactos como el *Sprint Backlog* y *Burn-down Charts*.

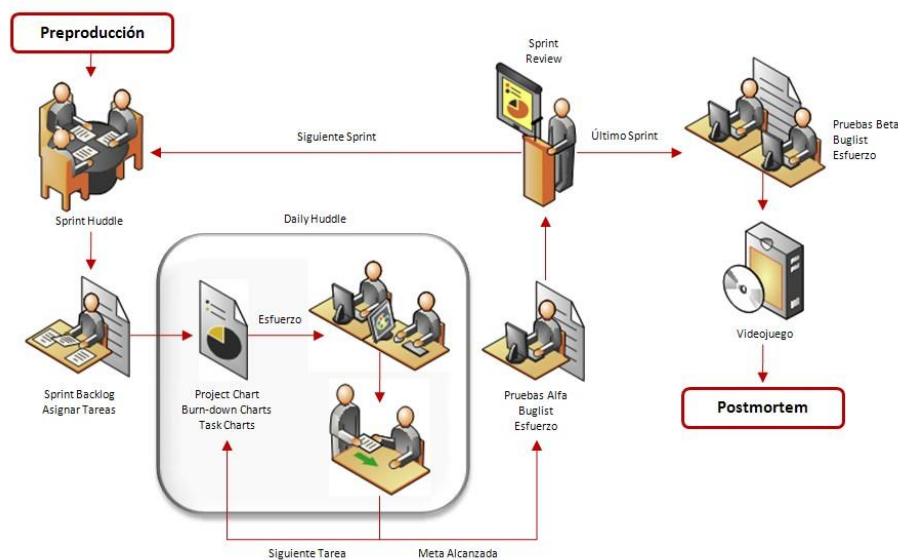


Figura N°3: Producción. Imagen autoría de Metodología Huddle.

Esta etapa comienza con un Sprint Iniciando el desarrollo del videojuego, se reúne todo el equipo y analizan los requerimientos que se definieron anteriormente, generando un nuevo documento con las tareas a realizar para poder lograr la meta del ciclo. Las tareas son seleccionadas por los miembros del equipo, ellos deciden el tiempo necesario para su desarrollo y si necesitar más personas involucradas. Diariamente se realizará una reunión para discutir el progreso y los obstáculos presentados. Al alcanzar la meta se procederá a la etapa de prueba durante esta actividad los miembros dedicados a prueba se dedican a probar cada característica implementada y se aseguran que la meta haya sido alcanzada, a la vez el ciclo

termina cuando se realiza la depuración de errores. Al terminar todos los ciclos se alcanza un hito de una versión beta del juego el cual será testeado por los clientes en un Closed Beta y un Open Beta.

2.1.3 Postmortem:

En esta fase el equipo realiza una última actividad llamada End-Game Huddle en la cual se analiza los aspectos positivos y negativos del proyecto, se realizan estas preguntas entre los miembros del equipo:

¿Que salió bien?

¿Que salió mal?

¿Qué obstáculos se presentaron?

Todas estas sugerencias deben ser analizadas y filtradas con la intención de generar un reporte para un próximo proyecto.

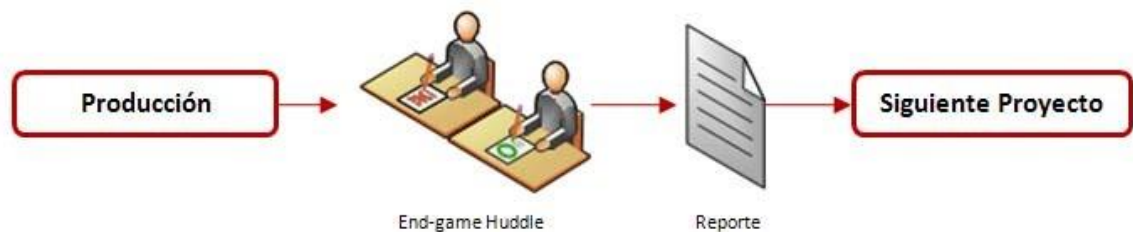


Figura N°4: Postmortem. Imagen autoría de Metodología Huddle.

Este reporte es un paso importante ya que resulta más sencillo comenzar un nuevo proyecto basándose en los resultados obtenidos en uno anterior. Dentro de este proyecto no se planteó un postmortem, ya que el juego queda en un estado de prototipo y no se vio correcto realizar esta fase para un proyecto que va a tener muchas más versiones antes de un resultado final. Sin embargo, sirvió para hacer una reflexión del trabajo realizado.

2.1.4 Roles:

Huddle no se apoya directamente en los roles de Scrum, existe un Project Manager que al igual que el Scrum Master que se encarga de que el proceso se siga al pie de la letra, también se encarga de que los artefactos estén al día y que el equipo trabaje de manera eficiente.

El Game Designer juega un papel muy importante en esta fase ya que estará pendiente de cómo se van generando los diferentes requerimientos específicos y de las características nuevas que se pueden agregar en cada Sprint.

En el proyecto actual una persona es suficiente para tomar los roles necesarios de la metodología, encargado de la idea inicial, analizar y desarrollarla hasta el producto final o prototipo funcional.

2.1.5 Documento de Diseño:

Es el artefacto más importante en el proceso se carga a lo largo del proyecto y debe estar dispuesto a sufrir varios cambios desde la etapa de revisión. Contiene todas las especificaciones necesarias para comenzar el proyecto, estas van desde del tema principal del videojuego hasta el número de niveles. La idea de este documento es plasmar la idea en un formato limpio con información detallada del videojuego, a su vez este documento es estándar se puede modificar a lo largo el proyecto, este documento es la guía básica de la metodología Huddle.

Esta parte es vital ya que decide el futuro del proyecto, es revisado por los desarrolladores independientes que conforman el equipo para seguir con la etapa de producción. Como todo método dentro del desarrollo de software, Huddle también otorga herramientas útiles para su uso, a continuación, se exponen las plantillas facilitadas, de las cuales se toman las necesarias según el concepto del juego original para este proyecto. Dentro de las mismas se presentan algunos términos utilizados en el desarrollo de videojuegos que son esenciales para plasmar la idea posteriormente.

Concepto	
<i>Campo</i>	<i>Descripción</i>
<i>Título</i>	<i>El título el juego, debe ser un nombre que capte la atención del jugador y del lector del documento tiene que hacer alusión al concepto del juego y ser un título memorable.</i>
<i>Estudio/Diseñadores</i>	<i>El nombre del estudio o del diseñador o diseñadores del documento.</i>
<i>Género</i>	<i>Qué tipo de juego será, cuál será su estilo.</i>
<i>Plataforma</i>	<i>Está diseñado para que hardware.</i>
<i>Versión</i>	<i>La versión del documento, debe ser número y no una fecha.</i>
<i>Sinopsis/Jugabilidad/Contenido</i>	<i>Escribir la esencia del juego, Incluir una descripción del contenido.</i>
<i>Categoría</i>	<i>Resaltar características principales del juego.</i>
<i>Licencia</i>	<i>Describir si el juego está basado en algo o si es original.</i>
<i>Mecánica</i>	<i>Describir la jugabilidad y el control del juego.</i>
<i>Tecnología</i>	<i>Listar el hardware y software que se requiere para la producción del juego.</i>
<i>Público</i>	<i>Describir a quién va dirigido el juego, que sector social va a verse más influenciado con él.</i>

Tabla N°4: Plantilla documento de diseño – Concepto.

Historial de Versiones
<i>El documento de diseño es un artefacto que siempre está sujeto a cambios, por lo anterior se requiere un control para las diferentes versiones que registre los cambios. El número de versión varía de acuerdo a la importancia en el cambio.</i>

Tabla N°5: Plantilla documento de diseño – Historial de versiones.

Visión General del Juego
<i>En este segmento se redacta el enfoque del juego que a su vez guiara el proyecto hasta el final. Básicamente debe mencionar lo interesante, características y conceptos de originalidad del juego. Este párrafo debe dejar al lector con ganas de saber más del proyecto.</i>

Tabla N°6: Plantilla documento de diseño – Visión general del juego.

Mecánica del Juego	
<i>Esta sección esencialmente describe lo que el jugador puede hacer y cómo puede hacerlo.</i>	
<i>Cámara</i>	<i>Describe que perspectiva tiene el jugador ante lo que está viendo en pantalla.</i>
<i>Periféricos</i>	<i>Que periféricos utilizar el jugador para lograr los objetivos mencionados.</i>
<i>Controles</i>	<i>Describir que botón hace cada acción.</i>
<i>Puntaje</i>	<i>Explica de qué manera luego se mantiene al tanto de los logros del jugador</i>
<i>Guardar/Cargar</i>	<i>Describir como el jugador guarda su progreso dentro del juego.</i>

Tabla N°7: Plantilla documento de diseño – Mecánica del juego.

Estados del Juego
<i>Esta sección se refiere al lugar en donde se encuentra el jugador durante el trayecto, si están el menú principal o en el menú de pausa, etc.</i>

Tabla N°8: Plantilla documento de diseño – Estados del juego.

Interfaces	
<i>Se debe describir la apariencia del juego, es decir colores y temática, todo esto relacionado con el concepto del juego.</i>	
<i>Nombre de la pantalla</i>	<i>Escribir el nombre de las diferentes pantallas a las que se puede acceder.</i>
<i>Descripción de la pantalla</i>	<i>Describir el funcionamiento de cada interfaz.</i>
<i>Estados del juego</i>	<i>Listar todos los estados de juego que invoquen esta pantalla.</i>
<i>Imagen</i>	<i>Una imagen que muestre el concepto de cómo se vería la pantalla.</i>

Tabla N°9: Plantilla documento de diseño – Interfaces.

Niveles	
<i>Normalmente un juego se divide en varios niveles, o mapas secuenciales dentro de los cuales se deben cumplir objetivos para avanzar.</i>	
<i>Título del nivel</i>	<i>El nombre del nivel.</i>
<i>Encuentro</i>	<i>Describir en qué momento llega el jugador a este nivel.</i>
<i>Descripción</i>	<i>Una descripción detallada del nivel.</i>

<i>Objetivos</i>	<i>Que debe hacer el jugador para terminar el nivel.</i>
<i>Progreso</i>	<i>Describir qué pasa cuando se termina nivel.</i>
<i>Enemigos</i>	<i>Listar enemigos encontrados en el nivel.</i>
<i>Objetos</i>	<i>Listar los objetos encontrados en el nivel.</i>
<i>Personajes</i>	<i>Listar personajes que aparecen en el nivel.</i>
<i>Sonido</i>	<i>Describir la música de este nivel al igual que los efectos de sonido.</i>
<i>BGM/SFX</i>	<i>Escribir todas las referencias que apliquen a la música o efectos de sonido.</i>

Tabla N°10: Plantilla documento de diseño – Niveles.

<i>Progreso del Jugador</i>
<i>Mostrar el flujo de eventos necesarios para avanzar en el juego.</i>

Tabla N°11: Plantilla documento de diseño – Progreso del juego.

<i>Personajes</i>	
<i>Describir los personajes principales y secundarios que se puede encontrar el jugador.</i>	
<i>Nombre</i>	<i>El nombre del personaje.</i>
<i>Descripción</i>	<i>Describir características físicas del personaje.</i>

<i>Imagen</i>	<i>Fotografía o dibujo conceptual.</i>
<i>Concepto</i>	<i>Describir la conducta y comportamiento, al igual que los motivos de eso. Mencionar su rol en la historia.</i>
<i>Encuentro</i>	<i>Cuando aparece este personaje en el juego.</i>
<i>Habilidades</i>	<i>Listar las habilidades del personaje.</i>
<i>Armas</i>	<i>Listar las armas del personaje.</i>
<i>Objetos</i>	<i>Listar objetos del personaje.</i>
<i>Npc</i>	<i>Describir el propósito de los personajes controlados por la máquina.</i>

Tabla N°12: Plantilla documento de diseño – Personajes.

Enemigos	
<i>En este apartado se describen los contrincantes que el jugador encontrara por el camino.</i>	
<i>Nombre</i>	<i>El nombre del enemigo.</i>
<i>Descripción</i>	<i>Describir el físico y comportamiento del enemigo.</i>
<i>Encuentro</i>	<i>Cuando aparece este enemigo en el juego.</i>
<i>Imagen</i>	<i>Fotografía o dibujo conceptual del enemigo.</i>
<i>Habilidades</i>	<i>Listar las habilidades del enemigo.</i>
<i>Armas</i>	<i>Listar armas del enemigo.</i>

Objetos	<i>Listar los objetos del enemigo.</i>
----------------	--

Tabla N°13: Plantilla documento de diseño – Enemigos.

Habilidades
<i>Los personajes y los enemigos suelen tener habilidades aparte de las comunes ya mencionadas, en esta sección se describe cada una de ellas.</i>

Tabla N°14: Plantilla documento de diseño – Habilidades.

Objetos
<i>Todos los objetos que el jugador puede encontrar y ayudaran en su progreso.</i>

Tabla N°15: Plantilla documento de diseño – Objetos.

Guion
<i>En esta sección se incluyen todos los diálogos del juego, debe incluir encabezados, nombres, dialogo, acción y transiciones.</i>

Tabla N°16: Plantilla documento de diseño – Guion.

Logros
<i>Describir todas las hazañas que el jugador puede conseguir mientras progresa, pueden ser secretos, medallas o cualquier extra poco usual.</i>

Tabla N°17: Plantilla documento de diseño – Logros.

Códigos Secretos
<i>Descubrir los códigos secretos que el jugador puede ingresar, lo que hacen y como son ingresados.</i>

Tabla N°18: Plantilla documento de diseño – Códigos Secretos.

Música y Sonidos
<i>La música y sonidos que se usaran dentro del juego, nombre, descripción, numero de referencia y como se utiliza en cada escenario. La referencia empieza con una "M" seguida de un número y con una "S" si es un efecto de sonido.</i>

Tabla N°19: Plantilla documento de diseño – Música y sonidos.

Imágenes de Concepto
<i>Todas las imágenes que muestren un posible nivel, personaje, objeto, etc. Deben estar enumeradas y con título.</i>

Tabla N°20: Plantilla documento de diseño – Imágenes de concepto.

Miembros del Equipo
<i>Información de las personas que trabajan en el proyecto, incluye nombre, el rol que desempeñan y medios para contactar.</i>

Tabla N°21: Plantilla documento de diseño – Miembros del equipo.

Detalles de Producción	
<i>Antes de entrar en la etapa de producción se definen en este documento algunos detalles básicos del proyecto.</i>	
<i>Fecha Inicio</i>	<i>Cuando empieza la producción del proyecto.</i>
<i>Fecha Fin</i>	<i>Cuando termina la etapa de producción del proyecto.</i>
<i>Presupuesto</i>	<i>Una estimación aproximada del presupuesto del juego.</i>

Tabla N°22: Plantilla documento de diseño – Detalles de producción.

2.2 Preproducción:

Siguiendo los lineamientos de la metodología Huddle se elaboró un plan inicial para el desarrollo del videojuego que puede ser actualizado en el documento según sea necesario. Se inicia concretamente con datos básicos del juego y se decidió abordar en este segmento la información del estudio desarrollador.

2.2.1 Título:

El nombre del juego se decidió basándose en la saga famosa de videojuegos a la que este proyecto hace referencia y homenaje “Castlevania”, este juego se desarrolla de principio a fin en un castillo donde se vive toda la historia, teniendo en mente esto, se inició el proceso para buscar un nombre elegante y que sea un referente a las acciones que el jugador tendrá que hacer. Se pensó además en un nombre que fuera fácil de recordar y que estuviera en un idioma que cualquier país podría reconocer, con esta primicia se llegó a la conclusión de nombrarlo “Castle of Chaos” o Castillo del Caos en español, nombre que además da una pauta de lo que el jugador o cliente se va a encontrar si adquiere el producto.

2.2.2 Estudio/Diseñador:

Aunque no se ha decidido de manera formal un nombre para empresa, teniendo en cuenta que sería un estudio independiente y de momento que cuenta con una sola persona en el equipo, que hace todas las funciones necesarias; se tiene inicialmente el nombre de “Gladheim” que hace alusión al lugar donde vivía el Dios nórdico Odín, el cual para esta cultura representaba la divinidad suprema.

El estudio no se plantea la creación de un solo estilo de videojuegos, sino que planea abarcar todo el mercado según las ideas de videojuegos que surjan a su diseñador, además tampoco se piensa estacionarse en un solo motor gráfico, sino aprender de cada uno de los existentes y enfocar el juego según la necesidad y facilidad de uso. De momento el motor gráfico que más llama la atención, además de Construct2, es Unity3D.

Como toda empresa o estudio se debe contar con un slogan, en este caso se optó por la frase “Life is play” referencia a Gilbert Keith Chesterton el cual era un periodista y escritor británico de inicio del siglo XX, que pensaba que la vida, aunque no era un juego, se podía jugar, vivir feliz en ella y encontrar el punto perfecto para crecer como persona.

El logo es un diseño estándar con el nombre del estudio y el slogan.



Figura N°5: Logo. Imagen autoría propia.

2.2.3 Genero:

El juego es del tipo Metroidvania o Castleroid, un estilo de juego que nació con las primeras versiones de dos juegos con una historia y un contexto totalmente diferente, pero que a medida que salían nuevas versiones fueron ligándose uno al otro por sus mecánicas parecidas y su mapeado. Literalmente este estilo de juegos se caracteriza por que el jugador debe seguir el mapa cuadrícula hasta encontrar un camino bloqueado y explorar hasta encontrar el objeto que permite seguir por ese camino. Estas mecánicas no permiten que se puede catalogar como simples juegos de acción, aventura o exploración, sino que tienen esas características que juntas lo hacen únicos e indescritibles. Metroid y Castlevania, estos dos títulos son los únicos juegos que tienen el honor atemporal de aportarle sus pertenencias a un estilo. Hoy en día, aunque no sea del todo sorpresivo, ningún juego que conserve este método ha podido subir el listón o lograr esos rasgos de calidad de aquella época, se han recibido innumerables títulos que trataron de ir por este camino, ninguno con el éxito de estas dos grandes sagas.



Figura N°6: Metroid clásico.
Imagen autoría de Nintendo.



Figura N°7: Castlevania clásico.
Imagen autoría de Konami.

2.2.4 Plataforma:

El juego inicialmente está pensado para ser jugado en computadora y en dispositivos móviles, aunque el software en el que es desarrollado tiene actualizaciones regulares según los sistemas de videojuegos accesibles en el mercado, por lo que posteriormente puede ser exportado a consolas fácilmente. La presentación del videojuego está pensada para hacerse desde cualquier navegador web, de esta manera se asegura su funcionalidad.

2.2.5 Versión:

La versión del documento será iniciada en el número 1 y será actualizada en la siguiente tabla según la importancia en los cambios añadidos en el transcurso del desarrollo del juego.

Versión: 1	Primera Versión del juego.
------------	----------------------------

Tabla N°23: Versión.

2.2.6 Sinopsis/Jugabilidad/Contenido:

Misteriosas desapariciones atacan un pequeño pueblo a las afueras de la ciudad, un ex investigador es designado para visitar el lugar donde ocurren los hechos, el único problema es que todas las pistas lo llevan a una vieja mansión donde tendrá que luchar por su vida para descubrir quién está detrás de todo. Su pasado será la clave para borrar el mal que azota esa tierra.

El juego transcurre en una mansión aparentemente abandonada donde se cree que hay centenares de personas secuestradas, pero no pueden dar con una prueba clara que lo confirme, el jugador tomara el control de este investigador retirado, llamado a descubrir que está sucediendo y liberar la región, todo esto bajo un ambiente de incertidumbre.

El jugador se enfrenta a todo tipo de enemigos, rompecabezas y pruebas de habilidad, todo esto para abrirse paso por esta mansión hasta encontrar el responsable de los hechos. Pero algo extraño sucede con el protagonista, ya que de alguna manera su pasado está conectado con lo que sucede en ese lugar, así que se topará con que no solo está resolviendo un caso, sino que también mientras transcurre la historia se conocerá a sí mismo, al final tendrá que usar todo lo que sabe y arriesgar su vida por salvar a la tierra del mal que lo asola.

Como se dijo en el preámbulo, existe un solo nivel, pero de unas grandes proporciones donde se vive toda la historia, con un entorno 2D bien cuidado y enemigos a cada paso, además de conceptos básicos de rol bien aplicados, se espera que el juego capte toda la atención de su público y sea un referente para el éxito del producto. Por otra parte, los controles planeados son muy intuitivos y permiten una jugabilidad ágil.

En cuanto a contenido, el jugador podrá encontrar una gran variedad de armas que podrá usar a su gusto, también interactuar con cada uno de los personajes que encuentre en el transcurso de su aventura y por ultimo habilidades necesarias para poder avanzar por cada uno de los escenarios.

2.2.7 Categoría:

Aunque se ha mencionado bastante las características de un juego del genero Metroidvania/Castleroid, vale la pena enumera todos los estilos que el jugador podrá encontrarse.

- **Lógica:** Se caracteriza por exigir agilidad mental al jugador.
- **Acción:** Exige que el jugador use su velocidad, destreza y tiempo de reacción.
- **Aventura:** Caracterizado por la exploración de niveles e interacción del personaje con NPCs.
- **Beat 'em up:** Se destaca el combate entre el personaje y muchos enemigos, con armas cuerpo a cuerpo.
- **Hack and Slash:** El mismo concepto que Beat 'em up pero con la diferencia que el personaje usa armas de filo.
- **Plataformas:** Juego donde el personaje tiene que correr, saltar o escalar sobre una serie de plataformas y acantilados con enemigos para completar el juego.
- **Rol:** Juego donde se interpreta un papel y el personaje evoluciona conforme transcurre el tiempo.

2.2.8 Licencia:

El juego mantendrá una licencia comercial o privada, no se podrá realizar ningún cambio en la estructura principal del juego, pero se brindarán facilidades para la creación de modificaciones por parte de la comunidad, buscando que se alargue el tiempo de vida del videojuego. No hay limitación de número de equipos en el que el comprador puede utilizar el producto y es libre de DRM o gestión de derechos digitales. Todas las imágenes y demás podrán ser usadas siempre y cuando no sea para fines económicos, en cuanto a la música su licencia queda ligada al concepto que sea definido por el autor de la misma.

2.2.9 Mecánica:

Para describir el movimiento del jugador primero se debe tener en cuenta la perspectiva en la que se ve el personaje y demás; se usa una perspectiva en dos dimensiones, así que de principio solo hay 3 espacios en los que el personaje puede moverse, de izquierda a derecha y saltando, los controles específicos se mostraran más adelante dentro de este mismo proyecto. También se habla que el jugador puede obtener más habilidades y ligeras modificaciones para ayudar su avance, por ejemplo, se plantea la posibilidad de doble salto, arrastrarse, salto en muro, etc.

Si se habla de combate el jugador tiene un ataque básico con su arma y una habilidad especial que produce mucho más daño a los enemigos, pero no puede ser usada constantemente y tiene un costo de mana. Así mismo los enemigos según su tipo y nivel de peligro tienen diferentes habilidades, por ejemplo, un enemigo básico en pantalla solo tendrá un ataque, pero los jefes tienden a tener una gama de habilidades diferentes.

En el mapa la mecánica es sencilla de explicar, el jugador debe explorar el mapa y al hacer esto se encontrará con secciones que simplemente no puede pasar a falta de alguna habilidad específica, por lo que debe investigar diferentes caminos para encontrar la adecuada y avanzar en el mapa, obviamente todo el recorrido estará lleno de monstruos, NPCs, jefes y demás elementos para enriquecer la jugabilidad.

Para profundizar un poco en la mecánica de los enemigos se plantea una inteligencia artificial sencilla, donde los enemigos estandar tienen un patron de movimiento en un area del juego y a su vez atacan en cuanto interactuan con el personaje, según el nivel del enemigo este puede hacer mas daño o tener alguna habilidad extra. En cuanto a los jefes de escenario, el cambio es mas grande, ya que tiene un tamaño mucho mayor, una gama de habilidades diferentes y un daño a tener en cuenta.

Para concluir, el jugador encuentra retos de todo tipo, no solo tendra que usar habilidades de movimiento y combate, tambien afrontar diferentes estrategias y sortear desafios logicos.

2.2.10 Tecnología:

En este apartado se enlista el hardware y software usado para la realización del videojuego, se comenzará por nombrar las especificaciones técnicas del equipo donde fue desarrollado.

- Windows 10 Edición profesional, 64 bits.
- Procesador AMD FX-8320, procesador de 8 núcleos.
- 8192 Megabytes de RAM.
- Tarjeta de video Nvidia Geforce GTX 660.
- 2 Discos duros de 1 Terabyte c/u.
- Pantalla de 19 Pulgadas, formato pantalla ancha.

A continuación, el hardware usado.

InkScape, software libre de diseño gráfico, perfecto para tareas avanzadas, que además maneja vectores como formato nativo en vez de mapas de bits. Fue usado para la creación de los fondos de cada escenario, diseño de personajes, iconos, pantalla de presentación y demás.

Construct 2, este el software que se utilizó para desarrollar el videojuego, es un motor basado en eventos y en html5 para la realización de juegos en 2D, con un estilo visual bastante intuitivo y herramientas potentes. Se usó su versión gratuita, que ofrece las herramientas necesarias para el proyecto en cuestión.

Celtx, es un software para la creación de guiones y se utilizó en su versión gratuita. Lo mejor de este programa es que con tan solo unos pasos permite dar rienda suelta a nuestras ideas sin pensar mucho en el formato que debe llevar el texto. Se pueden incorporar anotaciones, fotografías de localizaciones, descripciones de personajes, de escenas, etc. Además, no se limita solo a un estilo, se puede ajustar a novelas, a comics, a obras de teatro o a lo que sea necesario.

2.2.11 Público:

El género Metroidvania/Castleroid es por sí mismo una de las joyas en el mundo de los videojuegos, si un producto sale al mercado con las características esenciales de este estilo bien desarrolladas, va a tener éxito de una forma u otra, pese a esto claramente su estilo retro refleja el público que va a jugarlo; el juego estará entonces enfocado en un público mayor de 18 años, que haya tenido la oportunidad de jugar previamente alguna entrega de este género. Se puede intuir por lo anterior que tendrá un público adulto y si el juego llega a tomar renombre, sin duda alguna la comunidad joven lo tendrá en cuenta.

El lanzamiento del juego está decidido en computadora debido a que es en términos absolutos la maquina más potente para cualquier cosa en comparación a las consolas, ya sea potencia gráfica, accesibilidad, comodidad y costos. Así que el público del pc master race será el enfoque principal en esta primera incursión del juego al mercado.

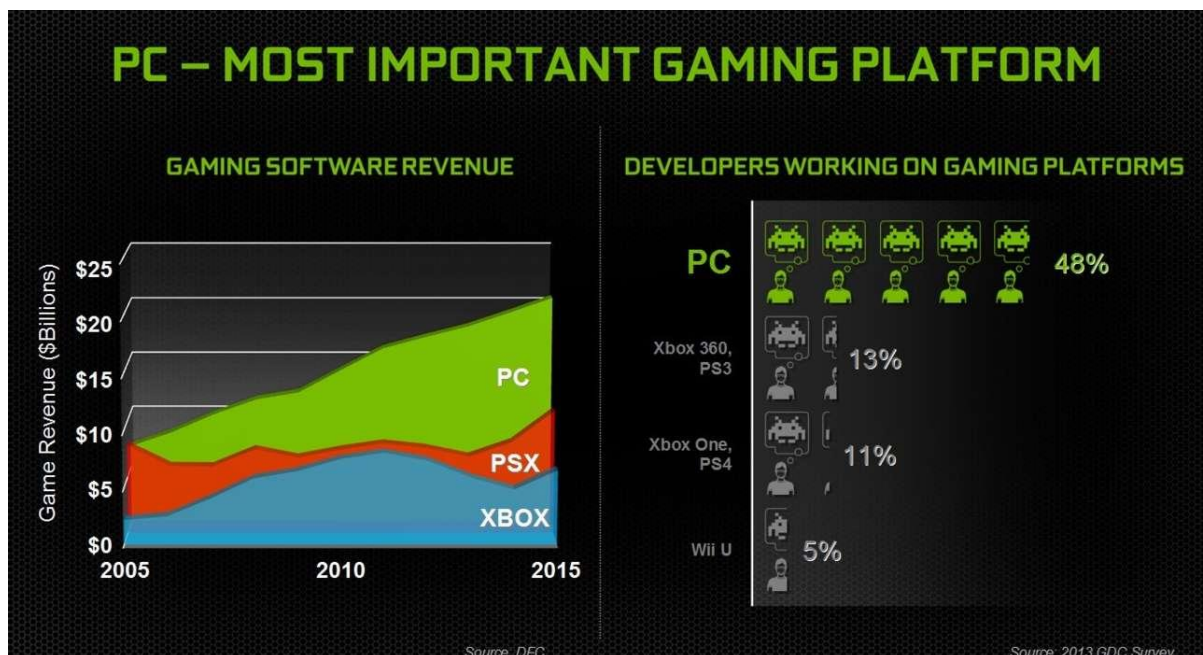


Figura N°8: Plataforma Pc. Imagen autoría de Nvidia.

2.2.12 Mecánica/Controles:

A continuación, se describirá los controles que se usaran específicamente en el juego, en este caso se hará para teclado y para control.

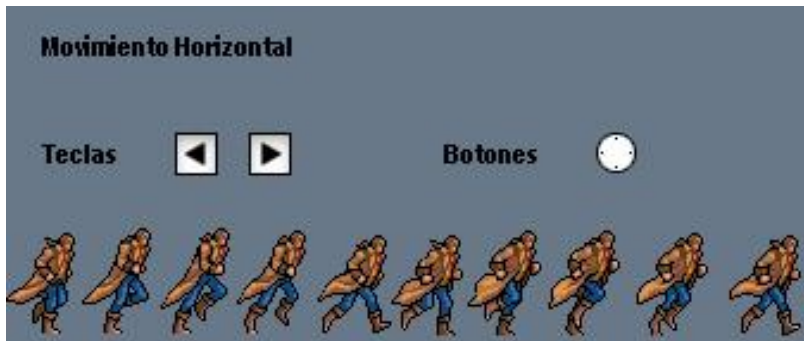


Figura N°9: Movimiento horizontal. Imagen autoría propia.

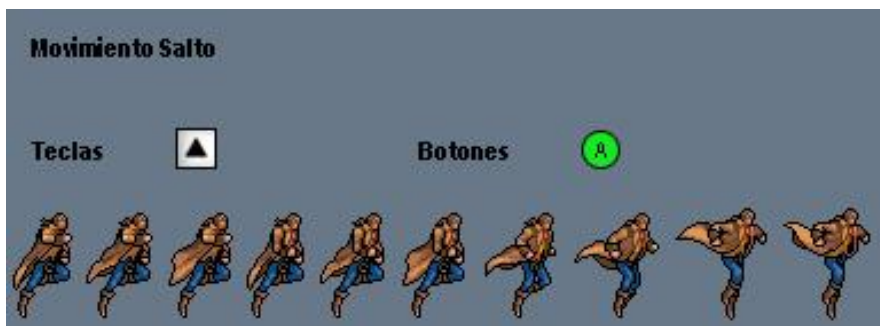


Figura N°10: Movimiento salto. Imagen autoría propia.

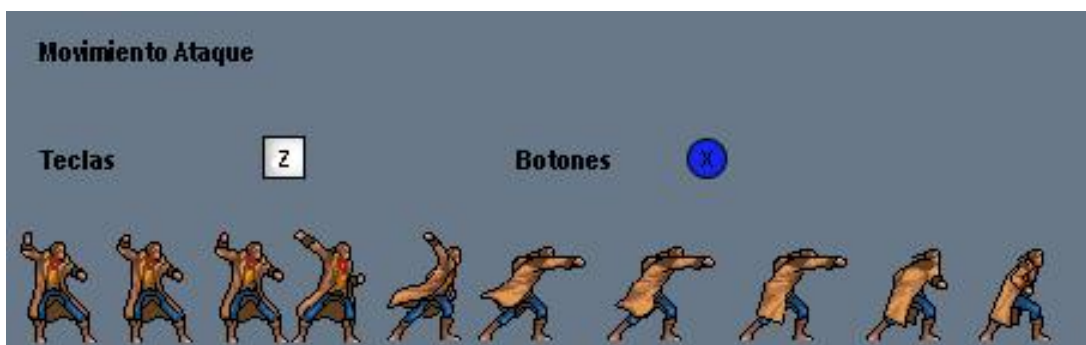


Figura N°11: Movimiento ataque. Imagen autoría propia.

2.2.13 Cámara:

La cámara del juego como se ha dicho reiteradamente, es en dos dimensiones, el jugador puede ver al personaje de forma lateral, así mismo se verá el nivel por supuesto. Usando un poco el concepto de profundidad y colores, se logra un efecto en el que la habitación es más grande de lo que parece.



Figura N°12: Cámara 2D. Imagen autoría propia.

2.2.14 Periféricos:

Los controles de principio son teclado o control estándar para videojuegos, esto si el jugador esta desde su computadora, en caso de ser usado en un teléfono inteligente, el control será totalmente con el tacto.



Figura N°13: Teclado.
Imagen autoría propia.



Figura N°14: Control.
Imagen autoría propia.

2.2.15 Puntaje:

No se utiliza puntaje en el juego, aunque posteriormente se pueden añadir modos de oleadas de enemigos, o contra reloj. La manera de ganar en este juego está ligada a llegar al jefe final y derrotarlo, esta es la forma en la que termina este prototipo, en versiones posteriores este tipo de juegos tiene varios jefes finales distribuidos en varias zonas del mapa, y un jefe maestro mucho más difícil a los anteriores.



Figura N°15: Victoria. Imagen autoría propia.

2.2.16 Guardar/Cargar:

Para guardar la partida el jugador debe llegar a una sala en especial que puede estar en cualquier parte del mapa y activar un objeto, de esta forma si el jugador falla, volverá a esta sala, igual si decide salir de la partida y retomar el juego después.



Figura N°16: Guardar/Cargar. Imagen autoría propia.

2.2.17 Sprint plan:

Analizando cada aspecto necesario en el juego se ideó un plan de desarrollo con tiempos establecidos, cada una de las acciones necesarias fue documentada en tablas con toda la información establecida. A diferencia de la metodología Scrum que utiliza cada Sprint para una serie de acciones, Huddle permite que cada uno de estos sea usado para realizar una acción en un determinado tiempo, así se abordó el videojuego para un desarrollo más ordenado y estructurado, recordar que aunque se termine con todo el plan propuesto posteriormente si es necesario se pueden añadir más cosas al videojuego.

Así mismo al culminar la etapa de planeación o preproducción según el modelo Huddle, se da paso a la etapa de producción donde se desarrolló todo el videojuego.

Sprint N° 1	
Nombre: <i>Idea</i>	Fecha:
Iteraciones: <i>1</i>	Id: <i>001</i>
Objetivo: <i>Profundizar la idea inicial y dar pautas para el inicio del proyecto.</i>	
Criterios de aceptación: <i>La idea inicial debe estar bien elaborada, se debe tener claro lo que se quiere hacer y los alcances que el proyecto tendrá.</i>	
Acciones: <ul style="list-style-type: none">• <i>Profundizar idea inicial.</i>• <i>Analizar documento de diseño propuesto.</i>• <i>Analizar viabilidad del proyecto.</i>• <i>Dar aprobación o modificar el documento según sea necesario.</i>	

Tabla N°24: Sprint 1.

Sprint N° 2	
Nombre: Motor	Fecha:
Iteraciones: 1	Id: 002
Objetivo: Identificar el motor en el que se desarrollara el videojuego.	
Criterios de aceptación: Basados en el documento de diseño, el motor deber permitir cada característica que se desea implementar.	
Acciones: <ul style="list-style-type: none"> • Investigar motores gratuitos y de fácil acceso. • Elegir el que más convenga de la lista. • Verificar que el motor permita hacer todo lo que se necesita. 	

Tabla N°25: Sprint 2.

Sprint N° 3	
Nombre: Logo	Fecha:
Iteraciones: 1	Id: 003
Objetivo: Diseñar la imagen con el logo del estudio que acompaña el inicio del juego.	
Criterios de aceptación: Debe tener un diseño claro y debe contener el logo y el slogan del estudio.	
Acciones: <ul style="list-style-type: none"> • Diseñar imagen de fondo y aplicarla. • Posicionar logo. • Posicionar slogan. 	

Tabla N°26: Sprint 3.

Sprint N° 4	
Nombre: Música	Fecha:
Iteraciones: 1	Id: 004
Objetivo: <i>Buscar música que se relacione con la temática del juego.</i>	
Criterios de aceptación: <i>La música debe ambientar correctamente según en donde se encuentre el jugador.</i>	
Acciones: <ul style="list-style-type: none"> • <i>Investigar música gratuita y de fácil acceso.</i> • <i>Elegir canciones que mejor se acoplen al proyecto.</i> • <i>Asignar cada canción en su capa correspondiente.</i> 	

Tabla N°27: Sprint 4.

Sprint N° 5	
Nombre: Menú inicio	Fecha:
Iteraciones: 1	Id: 005
Objetivo: <i>Diseñar un menú de inicio para el juego con las opciones básicas.</i>	
Criterios de aceptación: <i>El menú debe ser atractivo, sencillo de entender y funcional.</i>	
Acciones: <ul style="list-style-type: none"> • <i>Diseñar imágenes que expresen la temática del videojuego y aplicarlas.</i> • <i>Hacer botones con las características necesarias.</i> • <i>Enlazar cada capa según sea necesario.</i> • <i>Asignar música de entrada.</i> 	

Tabla N°28: Sprint 5.

Sprint N° 6	
Nombre: Escenario 1	Fecha:
Iteraciones: 1	Id: 006
Objetivo: Diseñar la primera habitación del mapa.	
Criterios de aceptación: Debe contar con un fondo claro, físicas y música de fondo.	
Acciones: <ul style="list-style-type: none"> • Diseñar imágenes de fondo y aplicarlas. • Desarrollar físicas para el contorno de la habitación. • Asignar música de fondo. • Desarrollar elementos interactivos. 	

Tabla N°29: Sprint 6.

Sprint N° 7	
Nombre: Escenario 2	Fecha:
Iteraciones: 1	Id: 007
Objetivo: Diseñar la segunda habitación del mapa.	
Criterios de aceptación: Debe contar con un fondo claro, físicas y música de fondo.	
Acciones: <ul style="list-style-type: none"> • Diseñar imágenes de fondo y aplicarlas. • Desarrollar físicas para el contorno de la habitación. • Asignar música de fondo. • Desarrollar elementos interactivos. 	

Tabla N°30: Sprint 7.

Sprint N° 8	
Nombre: Escenario 3	Fecha:
Iteraciones: 1	Id: 008
Objetivo: Diseñar la tercera habitación del mapa.	
Criterios de aceptación: Debe contar con un fondo claro, físicas y música de fondo.	
Acciones: <ul style="list-style-type: none"> • Diseñar imágenes de fondo y aplicarlas. • Desarrollar físicas para el contorno de la habitación. • Asignar música de fondo. • Desarrollar elementos interactivos. 	

Tabla N°31: Sprint 8.

Sprint N° 9	
Nombre: Escenario 4	Fecha:
Iteraciones: 1	Id: 009
Objetivo: Diseñar la cuarta habitación del mapa.	
Criterios de aceptación: Debe contar con un fondo claro, físicas y música de fondo.	
Acciones: <ul style="list-style-type: none"> • Diseñar imágenes de fondo y aplicarlas. • Desarrollar físicas para el contorno de la habitación. • Asignar música de fondo. • Desarrollar elementos interactivos. 	

Tabla N°32: Sprint 9.

Sprint N° 10	
Nombre: Mapa	Fecha:
Iteraciones: 2	Id: 010
Objetivo: Diseñar el mapa que guía al jugador.	
Criterios de aceptación: Debe informar la posición del jugador y lo que ha descubierto.	
Acciones: <ul style="list-style-type: none"> • Diseñar imágenes para el mapa. • Desarrollar e implementar su uso. 	

Tabla N°33: Sprint 10.

Sprint N° 11	
Nombre: Guion	Fecha:
Iteraciones: 1	Id: 0011
Objetivo: Diseñar la primera habitación del mapa.	
Criterios de aceptación: Debe llevar los lineamientos correctos de la creación de guiones y sobretodo ser atractivo.	
Acciones: <ul style="list-style-type: none"> • Elaborar personajes complejos, con historia y personalidad. • Describir claramente los sucesos que ocurren en el juego. • Lograr una trama interesante y que se acople al tiempo de juego. • Corresponder al mapeado. 	

Tabla N°34: Sprint 11.

Sprint N° 12	
Nombre: Personaje 1	Fecha:
Iteraciones: 3	Id: 012
Objetivo: Diseñar el primer personaje el cual será el llevado por el jugador.	
Criterios de aceptación: Debe contar con una estética definida acorde a la temática del juego.	
Acciones: <ul style="list-style-type: none"> • Diseñar imágenes para cada habilidad planeada y aplicarlas. • Desarrollar físicas para el personaje. 	

Tabla N°35: Sprint 12.

Sprint N° 13	
Nombre: Enemigo 1	Fecha:
Iteraciones: 2	Id: 013
Objetivo: Diseñar el primer enemigo del juego.	
Criterios de aceptación: Debe contar con una estética definida acorde a la temática del juego y ser diferente de otros enemigos.	
Acciones: <ul style="list-style-type: none"> • Diseñar imágenes para cada habilidad planeada y aplicarlas. • Desarrollar físicas para el enemigo. 	

Tabla N°36: Sprint 13.

Sprint N° 14	
Nombre: <i>Enemigo 2</i>	Fecha:
Iteraciones: <i>2</i>	Id: <i>014</i>
Objetivo: <i>Diseñar el segundo enemigo del juego.</i>	
Criterios de aceptación: <i>Debe contar con una estética definida acorde a la temática del juego y ser diferente de otros enemigos.</i>	
Acciones: <ul style="list-style-type: none"> • <i>Diseñar imágenes para cada habilidad planeada y aplicarlas.</i> • <i>Desarrollar físicas para el enemigo.</i> 	

Tabla N°37: Sprint 14.

Sprint N° 15	
Nombre: <i>Enemigo 3</i>	Fecha:
Iteraciones: <i>2</i>	Id: <i>015</i>
Objetivo: <i>Diseñar el segundo enemigo del juego.</i>	
Criterios de aceptación: <i>Debe contar con una estética definida acorde a la temática del juego y ser diferente de otros enemigos.</i>	
Acciones: <ul style="list-style-type: none"> • <i>Diseñar imágenes para cada habilidad planeada y aplicarlas.</i> • <i>Desarrollar físicas para el enemigo.</i> 	

Tabla N°38: Sprint 15.

Sprint N° 16	
Nombre: <i>Enemigo 4</i>	Fecha:
Iteraciones: <i>2</i>	Id: <i>016</i>
Objetivo: <i>Diseñar el segundo enemigo del juego.</i>	
Criterios de aceptación: <i>Debe contar con una estética definida acorde a la temática del juego y ser diferente de otros enemigos.</i>	
Acciones: <ul style="list-style-type: none"> • <i>Diseñar imágenes para cada habilidad planeada y aplicarlas</i> • <i>Desarrollar físicas para el enemigo.</i> 	

Tabla N°39: Sprint 16.

Sprint N° 17	
Nombre: <i>Arma</i>	Fecha:
Iteraciones: <i>1</i>	Id: <i>017</i>
Objetivo: <i>Diseñar el arma del protagonista.</i>	
Criterios de aceptación: <i>Debe contar con una estética definida acorde a la temática del juego y ser única.</i>	
Acciones: <ul style="list-style-type: none"> • <i>Diseñar imagen para el arma y aplicarla.</i> • <i>Desarrollar método de obtención.</i> • <i>Desarrollar físicas para su uso.</i> 	

Tabla N°40: Sprint 17.

Sprint N° 18	
Nombre: <i>Habilidad</i>	Fecha:
Iteraciones: 1	Id: 018
Objetivo: <i>Diseñar una habilidad para protagonista.</i>	
Criterios de aceptación: <i>Debe contar con una acción no convencional y ser única.</i>	
Acciones: <ul style="list-style-type: none"> • <i>Diseñar imágenes para cada acción planeada y aplicarlas.</i> • <i>Desarrollar método de obtención.</i> • <i>Desarrollar físicas para su uso.</i> 	

Tabla N°41: Sprint 18.

Sprint N° 19	
Nombre: <i>Pausa</i>	Fecha:
Iteraciones: 2	Id: 019
Objetivo: <i>Diseñar el menú de pausa.</i>	
Criterios de aceptación: <i>Debe mostrar estadísticas del jugador y el inventario.</i>	
Acciones: <ul style="list-style-type: none"> • <i>Desarrollar mecánica para pausar el juego.</i> • <i>Modificar volumen en pantalla de pausa.</i> • <i>Asignar vista de inventario en menú de pausa.</i> 	

Tabla N°42: Sprint 19.

Sprint N° 20	
Nombre: <i>Inventario</i>	Fecha:
Iteraciones: <i>2</i>	Id: <i>020</i>
Objetivo: <i>Diseñar el inventario básico del personaje.</i>	
Criterios de aceptación: <i>Debe mostrar los elementos que ha encontrado, y la cantidad.</i>	
Acciones: <ul style="list-style-type: none"> • <i>Desarrollar inventario básico.</i> • <i>Diseñar imagen de cajón para el inventario.</i> 	

Tabla N°43: Sprint 20.

Sprint N° 21	
Nombre: <i>NPC 1</i>	Fecha:
Iteraciones: <i>2</i>	Id: <i>021</i>
Objetivo: <i>Diseñar el primer personaje con el que el jugador puede interactuar.</i>	
Criterios de aceptación: <i>Debe contar con una estética definida acorde a la temática del juego y brindar información relevante al jugador.</i>	
Acciones: <ul style="list-style-type: none"> • <i>Diseñar imágenes para cada habilidad planeada y aplicarlas.</i> • <i>Desarrollar físicas para el personaje.</i> 	

Tabla N°44: Sprint 21.

Sprint N° 22	
Nombre: Jefe	Fecha:
Iteraciones: 2	Id: 022
Objetivo: Diseñar jefe final del escenario.	
Criterios de aceptación: Debe contar con una estética definida acorde a la temática del juego y tener características importantes.	
Acciones: <ul style="list-style-type: none"> • Diseñar imágenes para cada habilidad planeada y aplicarlas. • Desarrollar físicas para el enemigo. 	

Tabla N°45: Sprint 22.

Sprint N° 23	
Nombre: Pruebas	Fecha:
Iteraciones: 3	Id: 023
Objetivo: Realizar pruebas y corrección de errores.	
Criterios de aceptación: El juego debe ser probado por diferentes personas que a su vez entregar una lista de problemas encontrados para su solución.	
Acciones: <ul style="list-style-type: none"> • Distribuir el juego entre los probadores. • Recoger lista de errores encontrados. • Desarrollar correcciones y realizar pruebas nuevamente. 	

Tabla N°46: Sprint 23.

3. Análisis y Diseño

3.1 Definición de requerimientos:

Dada la complejidad del juego y sus diferentes matices, se determinaron los siguientes requerimientos funcionales basados en los diferentes géneros que se pueden encontrar y en su jugabilidad.

3.1.1 Requerimientos funcionales:

- El juego debe permitir total movilidad por cada escenario.
- El juego debe mostrar una buena variedad de escenarios.
- El juego debe brindar la opción de guardar/cargar una partida.
- El juego debe mostrar una buena variedad de enemigos.
- El juego debe brindar la posibilidad de usar diferentes armas secundarias.
- El juego debe permitir la obtención de diferentes tipos de habilidades.
- El juego debe presentar diferentes estilos de desafíos lógicos.
- El juego debe implementar un sistema de comercio óptimo.
- El juego debe implementar una escala de dificultad definida.
- El juego debe mostrar un mapa de cada zona.
- El juego debe permitir cambiar la dificultad.
- El juego debe permitir la interacción con algunos NPCs en cada zona.
- El juego debe presentar una historia clara y concisa al jugador.
- El juego debe permitir el uso de ítems para regenera salud o magia.

Basándose en la accesibilidad esperada del juego y del alcance al que se quiere llegar se definieron los siguientes requerimientos no funcionales, enfatizando sobretodo en bajas limitaciones y requisitos técnicos mínimos. Sin embargo, se pone un estándar más alto para evitar dificultades con algunos usuarios.

3.1.2 Requerimientos no funcionales:

- El juego debe mantener una tasa de 60 cuadros por segundo estables.
- El juego debe estar optimizado para trabajar en equipos de media/alta gama.
- El juego debe ser accesible a través de un navegador con soporte HTML5.
- El juego debe poderse controlar con un mando o un teclado.
- El juego debe estar disponible en formato digital y posteriormente en físico.
- El juego debe contar con una interfaz limpia y entendible.
- El juego debe implementar música y sonidos que ambienten cada escenario.
- El juego debe mostrar de manera clara el texto en pantalla.
- El juego debe mantener una armonía de colores buena.
- El juego debe brindar soporte a varias resoluciones de pantalla.
- El juego debe poder ser jugado desde dispositivos móviles.
- El juego debe tener una licencia comercial detallada por el desarrollador.
- El juego debe emplear un control muy intuitivo.
- El juego debe contar con un financiamiento por medio de Kickstarter o Greelight.
- El juego debe permitir la inclusión de contenido adicional.
- El juego debe seguir la metodología Huddle.
- El juego debe ser documentado con lenguaje de modelamiento UML.
- El juego no debe tardar más de 10 segundos entre cada capa o ventana.
- El juego debe ser accesible en Windows.
- El juego debe contar con un documento que lo respalde.
- El juego debe incluir un manual de instrucciones.
- El juego debe ser desarrollado en Construct2.
- El juego debe correr sin problemas en los equipos con las especificaciones mínimas requeridas.
- El juego debe contar con soporte para usuarios.

3.2 Descripción del sistema propuesto:

El sistema propiamente llamado videojuego en este proyecto, abarca el desarrollo de principio a fin de un videojuego con todo lo que esto significa, un juego como único fin divertir, enfocado desde su idea inicial a un público adulto conocedor del género y con un estilo retro definido, todo esto aprovechando las características de algunos de los más famosos videojuegos y la facilidad para plasmarlos en el presente.

Se usa un motor basado en HTML5 y eventos para diseñar un juego en dos dimensiones, con muchas características fácilmente reconocibles de otros juegos, además de esto, se elaboró un guion que, aunque básico, permite al jugador un entendimiento total de lo que hace y por qué.

El videojuego cuenta con interfaces claras, un nivel de buena proporción, objetos diferentes, varias habilidades, etc. Todo esto para superar el reto propuesto por el estudio.

En el apartado de enemigos, el jugador puede encontrar varios tipos, desde momias, hasta ojos demoniacos que tendrá que ya sea esquivar o eliminar con sus armas, por último, en cada sección final del mapa, el jugador debe enfrentarse a un jefe final si quiere seguir su camino y enriquecer el conocimiento que tenga de lo que está pasando. Pero no solo hay enemigos en este juego, también se puede encontrar con personajes controlados por la computadora que ayudarán o añadirán información importante para el transcurso del juego.

Castle of Chaos es un juego enfocado para navegadores web, así que esté será el que se encargue de la comunicación entre el juego y el usuario, hay que recordar que el motor en el que se trabajó, está basado en HTML5 y una implementación de JavaScript, pero de momento trabaja todos los elementos multimedia por medio de DirectX, así que, aunque en Windows trabaja sin ningún problema, en Linux se debe buscar una manera diferente para lograr su funcionamiento. En cuanto a móviles, el programa permite exportar el juego a estos, o simplemente jugar desde el propio navegador de cada teléfono inteligente.

3.2.1 Interfaces:

La primera interfaz con la que se encuentra el usuario es la pantalla de inicio, esta es la presentación del juego, es lo que en muchas ocasiones es juzgado por las personas para decidir si jugar o no. Contiene un fondo asociado a la temática del juego y un menú con opciones básicas.

1. Nuevo Juego: Botón que permite al jugador iniciar un nuevo juego desde cero, abrirá la capa de escenario en la posición por defecto.
2. Cargar: Botón que permite al jugador iniciar el juego desde donde lo dejó la última vez, abrirá la capa escenario en la sala donde guardó el juego.
3. Opciones: Botón que permite al jugador cambiar algunas características básicas del juego, abrirá una capa parecida al menú con opciones editables.
4. Salir: Botón que permite al jugador finalizar la aplicación, cerrará inmediatamente la pestaña o navegador en la que el juego se esté ejecutando.



Figura N°17: Interfaz menú. Imagen autoría propia.

La siguiente interfaz es el escenario básico en donde se desarrollará todo el juego, este a la vez está dividido en varias zonas o salas fácilmente identificables, al ser la entrada al videojuego como tal, también debe estar bien trabajada.

1. Personaje: El jugador identificara rápidamente el personaje que puede controlar, así mismo a enemigos que circulan por el mapa.
2. Puerta: Para diferenciar el acceso de una zona a otra, el jugador cuenta con unas puertas básicas dibujadas en el fondo.
3. Fondo: Es el arte que contiene el escenario para dar contexto al juego.
4. Pared: Estos muros definen claramente el lugar por el que jugador puede moverse, delimita la sala.



Figura N°18: Interfaz escenario. Imagen autoría propia.

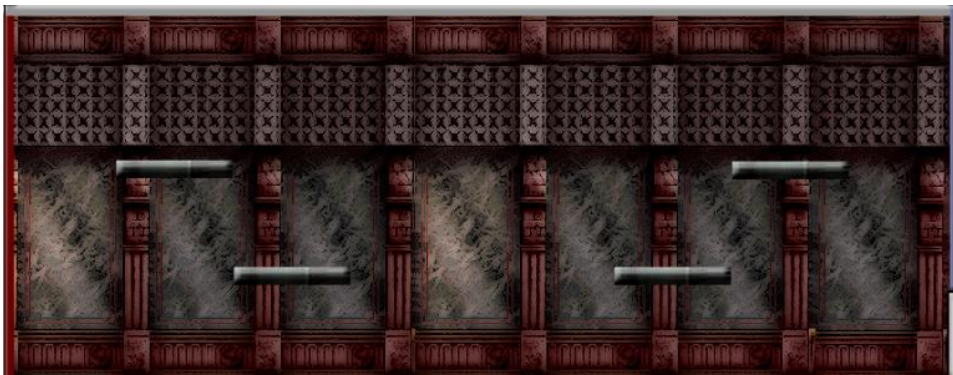


Figura N°19: Interfaz escenario2. Imagen autoría propia.

Por último, los diferentes elementos que brindan información al jugador están representados en una interfaz sencilla, una parte siempre esta visible y la otra es accesible a criterio del jugador por medio de los controles, ofrecen desde estadísticas del personaje hasta la ubicación de este en el mapa o el estado de la partida, se ideó de esta forma para evitar que la pantalla se llene con muchos elementos y molesten el recorrido del jugador.

1. Vida: La barra roja representa la vida del personaje, varía según las acciones del jugador.
2. Inventario: Cada casilla representa un espacio de inventario, el cual será usado para los objetos que se pueden recoger.
3. Mapa: Es el mini mapa que permite al jugador guiarse dentro del escenario, va aumentando según el recorrido.
4. Pausa: Este botón simboliza un estado en el que toda la capa se encuentra detenida.



Figura N°20: Interfaz utilidades. Imagen autoría propia.

3.3 Diseño del sistema propuesto:

Dada la magnitud del proyecto se diseñó una serie de diagramas, que permiten el entendimiento de cada sección del juego y de los elementos que lo contienen. Básicamente en esta sección se aplican técnicas y principios con el fin de definir el videojuego como aplicación, dando importancia a los detalles y permitiendo su realización. Para esto usaremos diagramas UML (Lenguaje unificado de modelamiento), un lenguaje estándar de software para describir un sistema.

3.3.1 Diagramas estáticos:

Este tipo de diagramas también llamados estructurados, se encargan de evaluar qué cosas (entidades, objetos, áreas, clases, etc.) deben estar definidas dentro de un sistema y como están estructuradas. Para este proyecto se tuvo en cuenta particularmente el diagrama de clases, el cual desde hace mucho tiempo representa muy bien la estructura de una aplicación.

3.3.1.1 Diagrama de clases:

Pese a que Construct2 (software que ya fue mencionado anteriormente como el motor de videojuegos usado en este proyecto.) utiliza hojas de eventos con una estructura basada en HTML5 y el usuario no observa o interfiere de forma directa en el código, se planteó un diagrama de clases básico que expone la estructura de las diferentes capas y sus funciones. Esto es posible debido a que el software sobre el que se hizo el proyecto, tiene una función de desarrollo dividida en módulos muy bien definida, así que trabajando de forma ordenada es bastante sencillo resaltar las características de los objetos usados y como se relacionan unos con otros.

El diagrama de clases permite además de mostrar lo que el sistema puede hacer (análisis) mostrar cómo puede ser construido (diseño), este tipo de diagramas es el pilar de UML, ya que representa de la mejor forma posible un sistema, la implementación de proyectos grandes no sería ni siquiera abordable sin uno de estos diagramas.

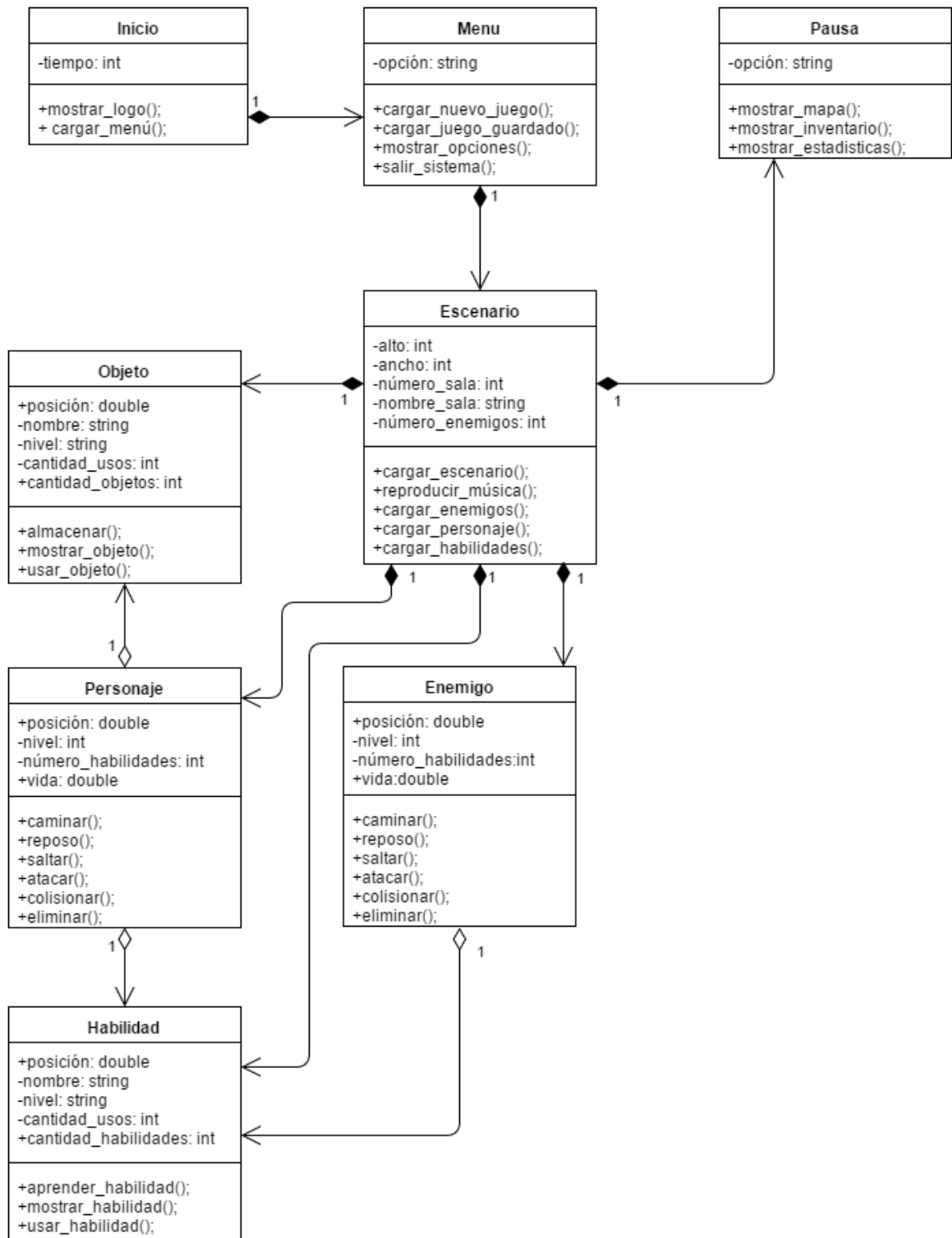


Figura N°21: Diagrama clases – Videojuego. Imagen autoría propia.

3.3.2 Diagramas dinámicos:

A diferencia de los diagramas estáticos, estos buscan modelar de forma entendible el comportamiento de cada uno de los elementos dentro de una aplicación, para este proyecto se planeó el uso de diagramas de estado y de casos de uso, así se observa cómo funciona cada módulo del videojuego y el papel que toma el usuario en esté.

3.3.2.1 Estados del juego:

En este proyecto se pueden observar varios estados del juego, esto quiere decir que el jugador puede acceder a diferentes momentos o lugares mientras se encuentre dentro de la aplicación realizando diferentes acciones, se listan a continuación y se presenta el diagrama tanto para las interfaces como para las acciones, propiamente se detalla cualquier acción desde el momento en que la aplicación comienza a funcionar, hasta que termina el juego o decide salir de esta.

Básicamente en los diagramas de estado que se diseñaron, se modela la vida del juego en sus diferentes fases y el flujo de control entre cada una de estas. Gracias a que Construct2 maneja eventos para el desarrollo de videojuegos, este tipo de diagramas expone muy bien la forma en que el jugador interactúa con cada elemento en las diferentes interfaces, así en caso de proveer solo los diagramas, sería muy sencilla su implementación posterior en el motor, estos son los estados identificados:

- Pantalla de Inicio.
- Nuevo Juego/Cargar Juego.
- Opciones.
- Menú de Pausa.
- Inventario.
- Escenario.
- Mapa.

3.3.2.2 Diagrama de estados:

Expresan cada objeto de un sistema con un comportamiento significativo, con esto se quiere decir que muestra los cambios que puede atravesar un elemento durante su existencia en respuesta a las acciones que otros tienen sobre él.

El primer diagrama de estado esta basado en la interacción del jugador con el menu principal, se expone cada opcion posible y su deselance hasta iniciar la partida o terminar la aplicación.

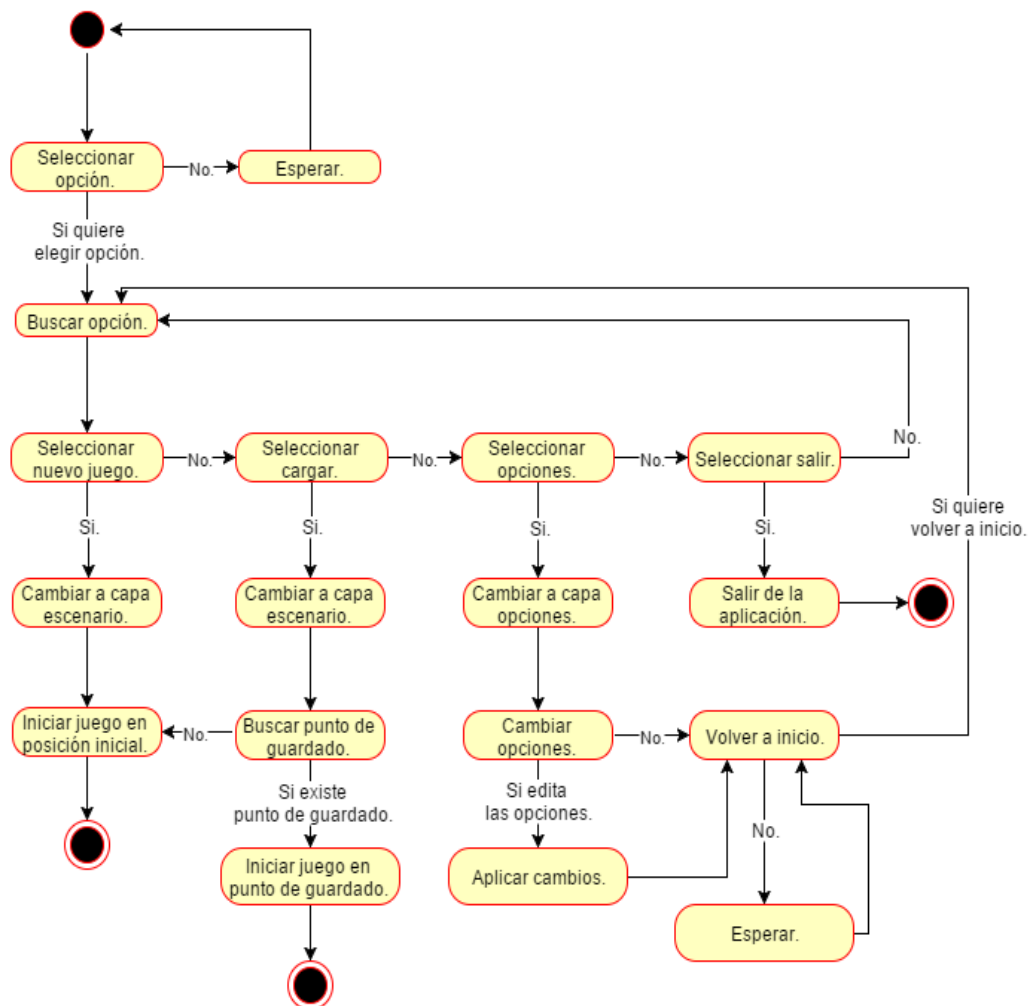


Figura N°21: Diagrama estado – Menú. Imagen autoría propia.

El segundo diagrama de estado muestra sobre todo las acciones posibles del jugador, sin contar las habilidades extras, por lo cual se observa cada movimiento básico del personaje, también los menús a los que puede acceder en cualquier momento de la partida, ya sea para ver información o tomar un descanso.

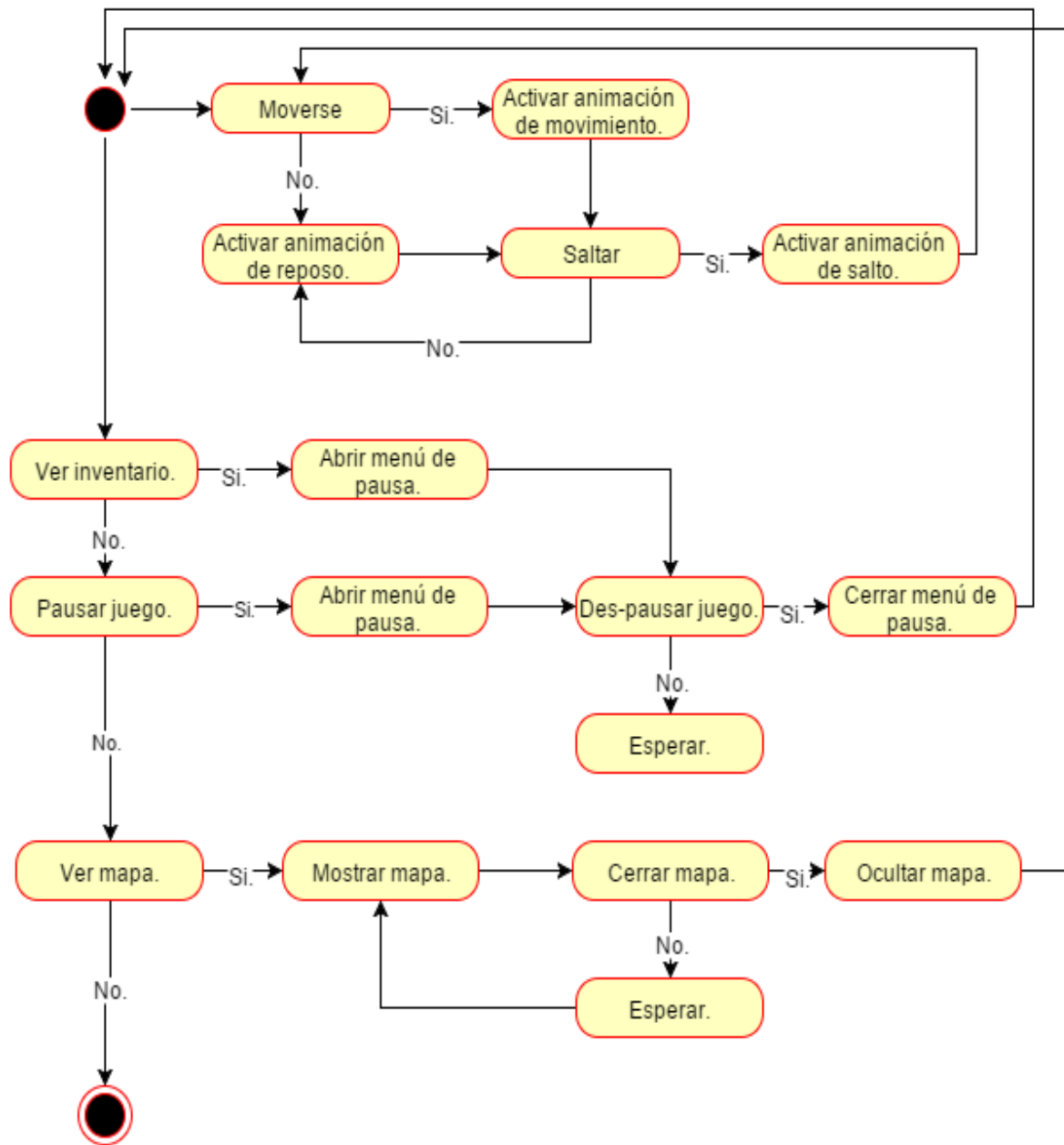


Figura N°22: Diagrama estado – Movimiento. Imagen autoría propia.

Por último, se ideó un diagrama que mostrara la interacción del personaje con los enemigos y NPCs, así mismo el uso de habilidades extras.

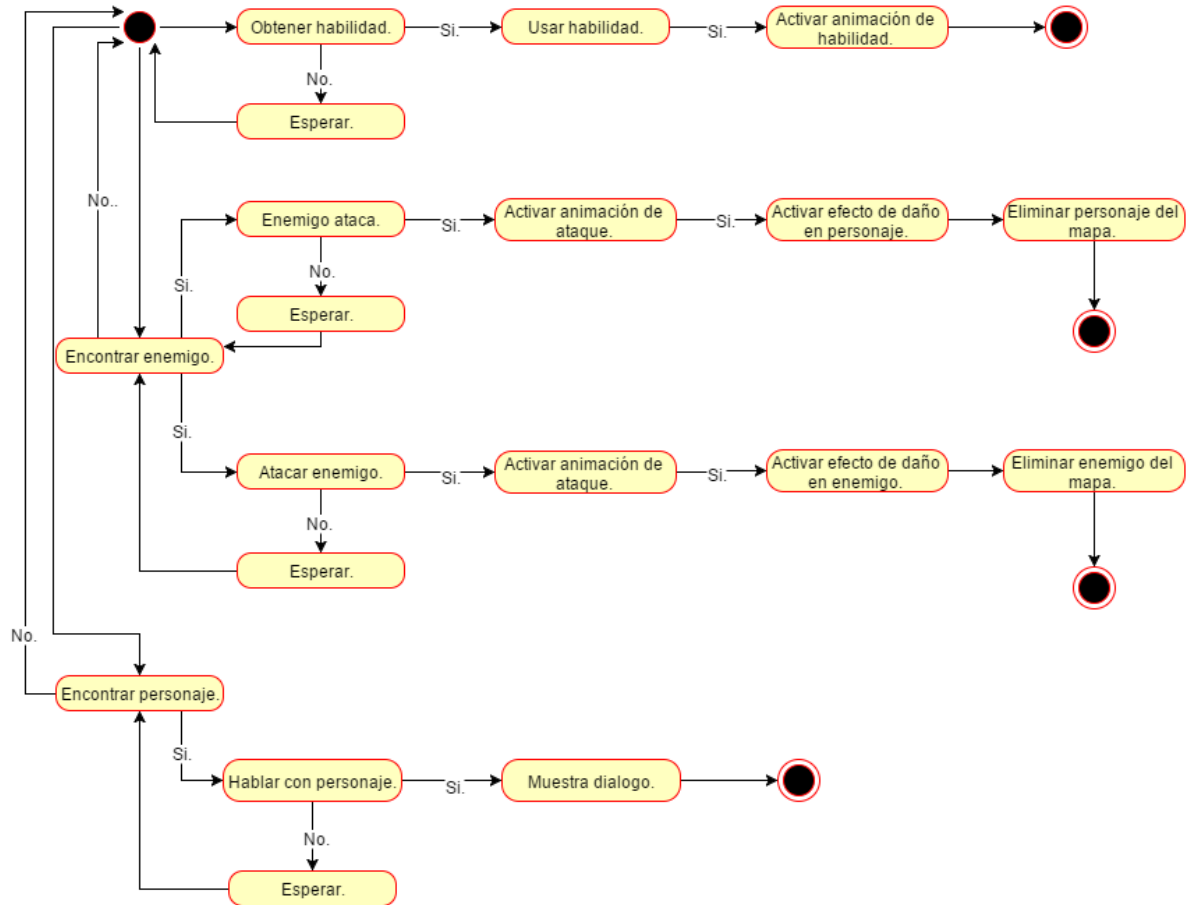


Figura N°23: Diagrama estado – Interacción. Imagen autoría propia.

3.3.2.3 Diagrama casos de uso:

Un caso de uso representa una unidad funcional coherente dentro de un sistema, subsistema o clase. Este tipo de diagramas exponen la interacción de uno o más usuarios con las acciones del sistema. Los personajes o entidades que participan en un caso de uso se denominan actores. Para este proyecto se tomó en cuenta la interacción del usuario con el menú y el escenario en sí.

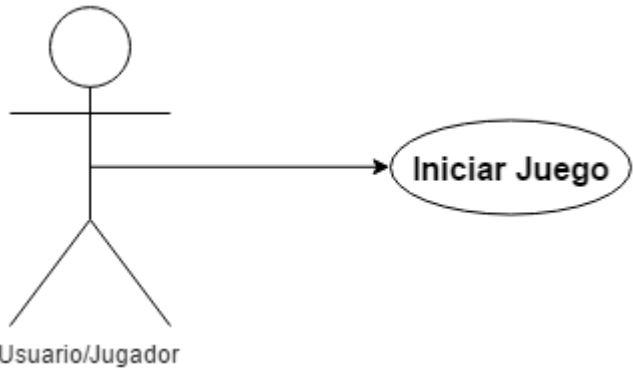
 <p>Usuario/Jugador</p> <p>Figura N°24: Caso – Iniciar juego. Imagen autoría propia.</p>	<p>Nombre: Iniciar juego.</p> <p>Descripción: El usuario entra al servidor donde se encuentra el juego o abre la aplicación.</p> <p>Flujo de eventos:</p> <ol style="list-style-type: none"> 1. El usuario demanda el ingreso al juego. 2. El sistema precarga cada elemento del juego. 3. Se presenta el logo del estudio desarrollador. 4. Se presenta el menú principal del juego. <p>Flujo alternativo:</p> <ol style="list-style-type: none"> 1. La precarga de los elementos falla. 2. El usuario debe reiniciar el juego. 3. La carga del menú falla. <p>Precondiciones: El navegador debe ser compatible con HTML5, equipo con controladores actualizados, dirección web del juego en correcto estado y pruebas del juego concluidas.</p> <p>Postcondiciones: El usuario se encuentra con el menú principal del juego.</p>
--	---

Tabla N°47: Diagrama caso 1.

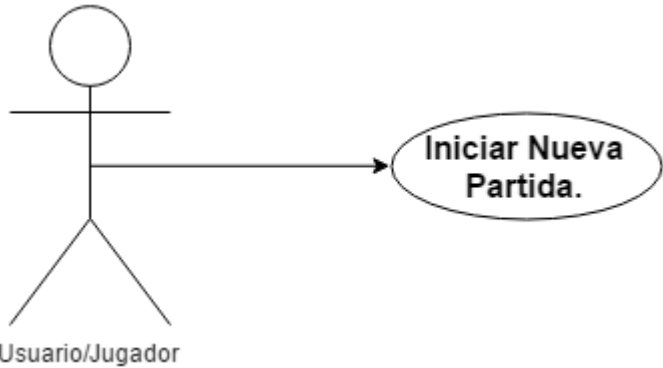
 <p>Usuario/Jugador</p> <p>Figura N°25: Caso – Iniciar nueva partida. Imagen autoría propia.</p>	<p>Nombre: Iniciar nueva partida.</p> <p>Descripción: El usuario elige comenzar una nueva partida, tomando esta opción en el menú.</p> <p>Flujo de eventos:</p> <ol style="list-style-type: none"> 1. El usuario elige “Nueva partida” en el menú principal. 2. Se carga la capa escenario e inicia el juego en la posición designada. <p>Flujo alternativo:</p> <ol style="list-style-type: none"> 1. La carga del escenario falla. 2. El usuario debe reiniciar el juego. 3. El usuario elige cualquiera de las otras opciones del menú. <p>Precondiciones: El escenario debe precargarse bien y cada opción del menú hacer la acción correspondiente.</p> <p>Postcondiciones: El usuario se encuentra en el escenario.</p>
--	--

Tabla N°48: Diagrama caso 2.

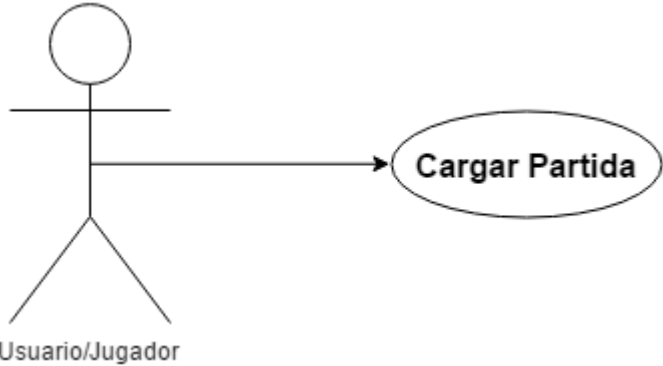
 <p>Usuario/Jugador</p> <p>Figura N°26: Caso – Cargar partida. Imagen autoría propia.</p>	<p>Nombre: Cargar partida.</p> <p>Descripción: El usuario elige cargar una partida, tomando esta opción en el menú.</p> <p>Flujo de eventos:</p> <ol style="list-style-type: none"> 1. El usuario elige “Cargar partida” en el menú principal. 2. Se carga la capa escenario e inicia el juego en la posición guardada anteriormente. <p>Flujo alternativo:</p> <ol style="list-style-type: none"> 1. La carga del escenario falla. 2. El usuario debe reiniciar el juego. 3. El usuario elige cualquiera de las otras opciones del menú. 4. No hay partida guardada. 5. El jugador inicia desde cero. <p>Precondiciones: El escenario debe precargarse bien, debe existir una partida guardada y cada opción del menú hacer la acción correspondiente.</p> <p>Postcondiciones: El usuario se encuentra en el escenario.</p>
---	---

Tabla N°49: Diagrama caso 3.

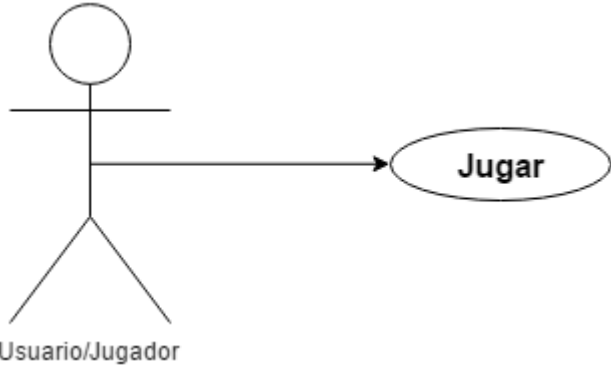
 <p>Usuario/Jugador</p> <p>Figura N°27: Caso – Jugar. Imagen autoría propia.</p>	<p>Nombre: Jugar.</p> <p>Descripción: Tras cargar el escenario el usuario podrá iniciar su recorrido por el juego.</p> <p>Flujo de eventos:</p> <ol style="list-style-type: none"> 1. El usuario está en el punto inicial o en el punto de guardado dentro del escenario. 2. El usuario toma el control del personaje principal. <p>Flujo alternativo:</p> <ol style="list-style-type: none"> 1. La carga del escenario falla. 2. El usuario debe reiniciar el juego. <p>Precondiciones: El escenario debe precargarse bien.</p> <p>Postcondiciones: El usuario controla cada acción del personaje.</p>
--	---

Tabla N°50: Diagrama caso 4.

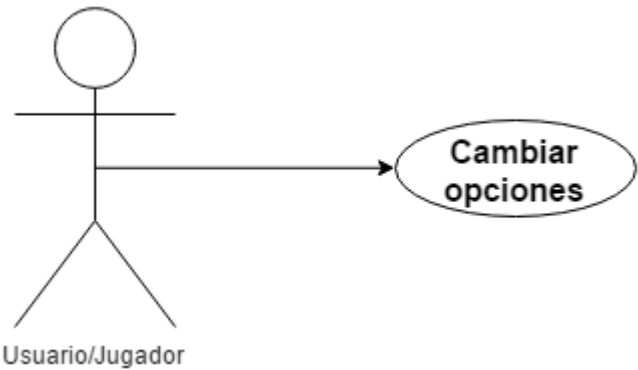
 <p>Usuario/Jugador</p> <p>Figura N°28: Caso – Cambiar opciones. Imagen autoría propia.</p>	<p>Nombre: Cambiar opciones.</p> <p>Descripción: El usuario entra a la sección de cambiar opciones dentro del menú principal.</p> <p>Flujo de eventos:</p> <ol style="list-style-type: none"> 1. El usuario elige “Opciones” en el menú principal. 2. Se carga la capa opciones. 3. Editar y aplicar cambios. <p>Flujo alternativo:</p> <ol style="list-style-type: none"> 1. La carga de las opciones falla. 2. El usuario debe reiniciar el juego. 3. El usuario elige cualquiera de las otras opciones del menú. 4. El usuario no cambia nada y regresa al menú. 5. Las opciones no se aplican. 6. El usuario debe repetir el proceso. <p>Precondiciones: La capa de opciones debe precargarse bien, debe existir mínimo una opción editable.</p> <p>Postcondiciones: El usuario hace los cambios necesarios y el sistema los aplica correctamente.</p>
---	---

Tabla N°51: Diagrama caso 5.

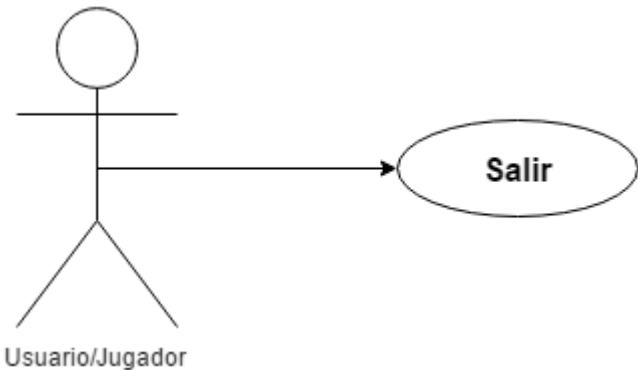
 <p>Usuario/Jugador</p> <p>Figura N°29: Caso – Salir. Imagen autoría propia.</p>	<p>Nombre: Salir.</p> <p>Descripción: El usuario elige salir desde el menú inicial o en medio de la partida.</p> <p>Flujo de eventos:</p> <ol style="list-style-type: none"> 1. El usuario elige “Salir” en el menú principal o en el menú de pausa. 2. El juego se cierra tras confirmar la decisión. <p>Flujo alternativo:</p> <ol style="list-style-type: none"> 1. La carga del escenario falla. 2. La carga del menú falla. 3. El usuario debe reiniciar el juego. 4. El usuario elige cualquiera de las otras opciones del menú. 5. Falla el cierre del juego. 6. El usuario debe repetir el proceso. <p>Precondiciones: El escenario y el menú deben precargarse bien, debe existir la posibilidad de salir del juego.</p> <p>Postcondiciones: El usuario cierra el juego y vuelve al escritorio o cierra la pestaña del navegador.</p>
--	--

Tabla N°52: Diagrama caso 6.

A continuación, se recopila cada caso de uso en el que el usuario interfiere.

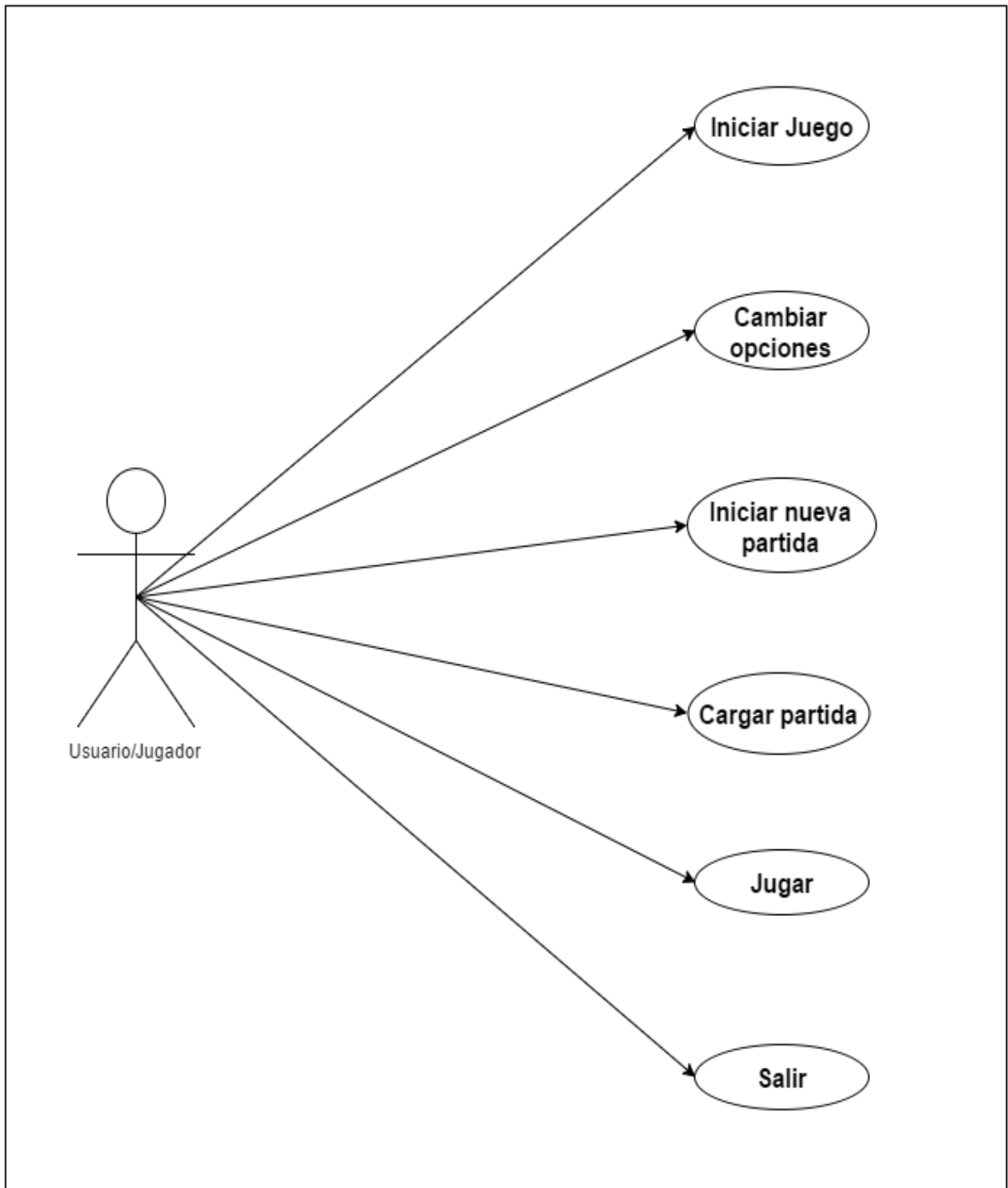


Figura N°30: Casos de uso videojuego. Imagen autoría propia.

3.3.2.4 Diagrama de secuencia:

Este tipo de diagramas lo que hace es mostrar la forma en la que los objetos se comunican entre sí al transcurrir el tiempo. Básicamente el diagrama muestra los objetos participando en la interacción y la secuencia de mensajes intercambiados. Normalmente se hace un diagrama de secuencia para cada uno de los casos de uso trabajado, pero para evitar ser repetitivo se redujeron a estos:

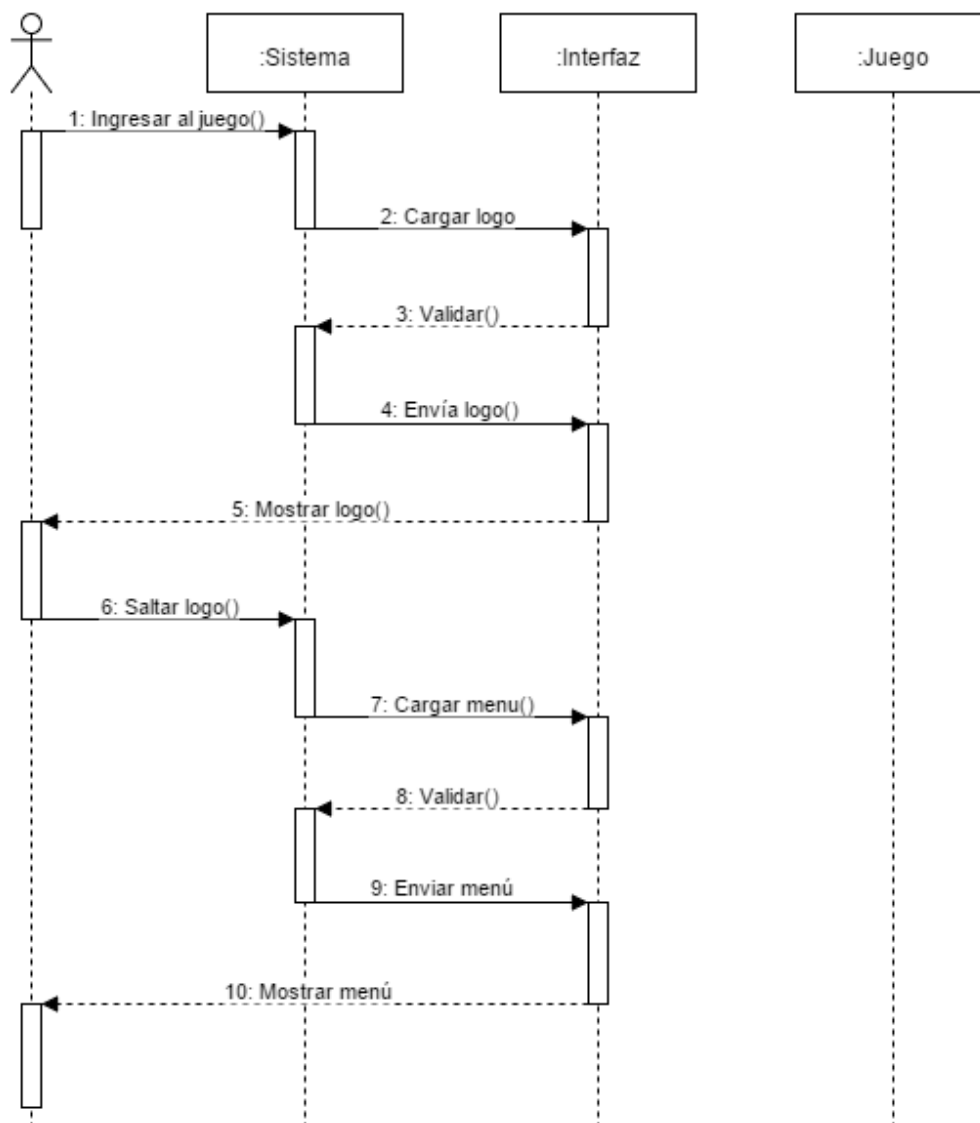


Figura N°31: Diagrama de secuencia 1 – Iniciar juego. Imagen autoría propia.

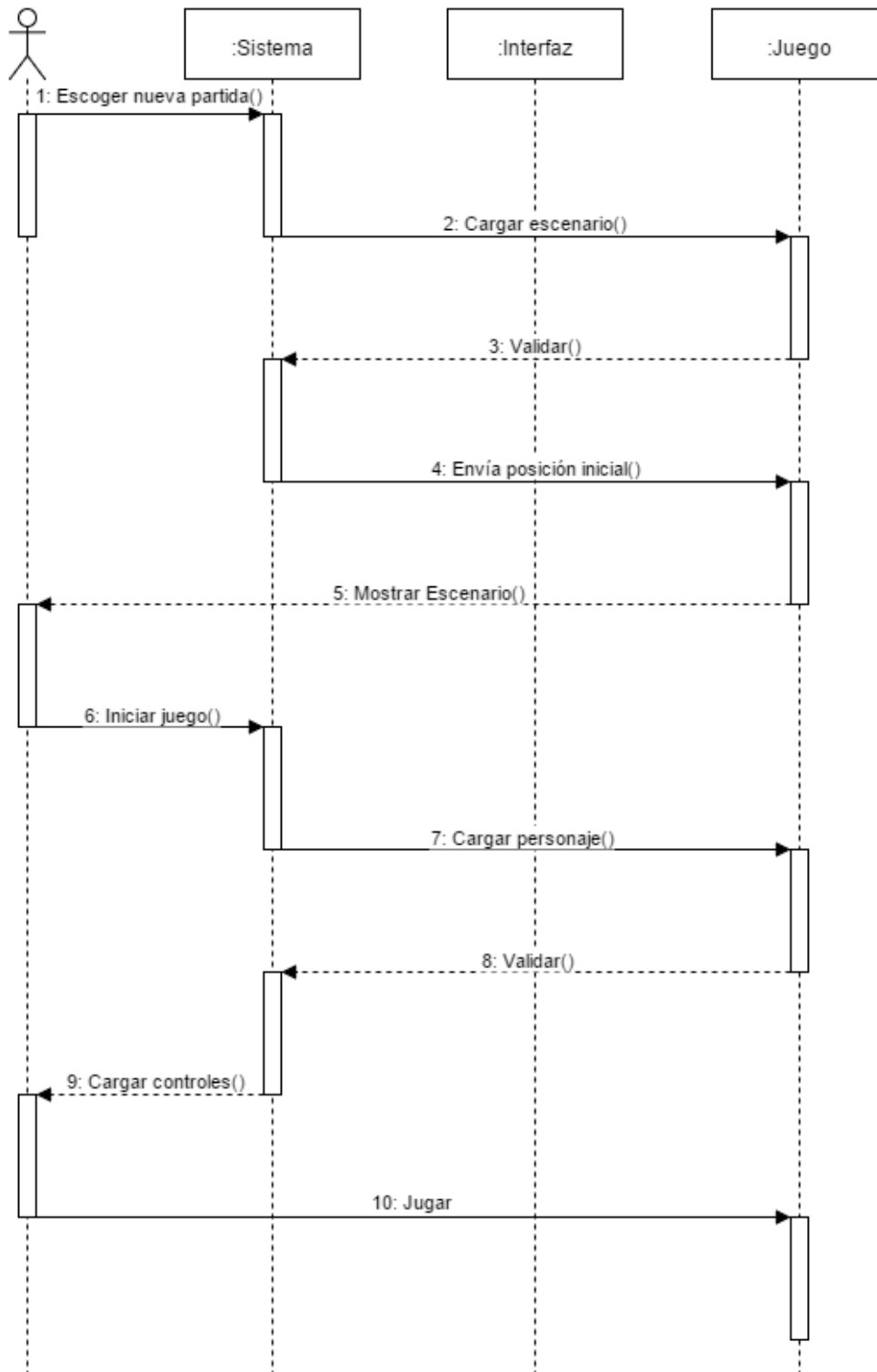


Figura N°32: Diagrama de secuencia 2 – Nueva partida. Imagen autoría propia.

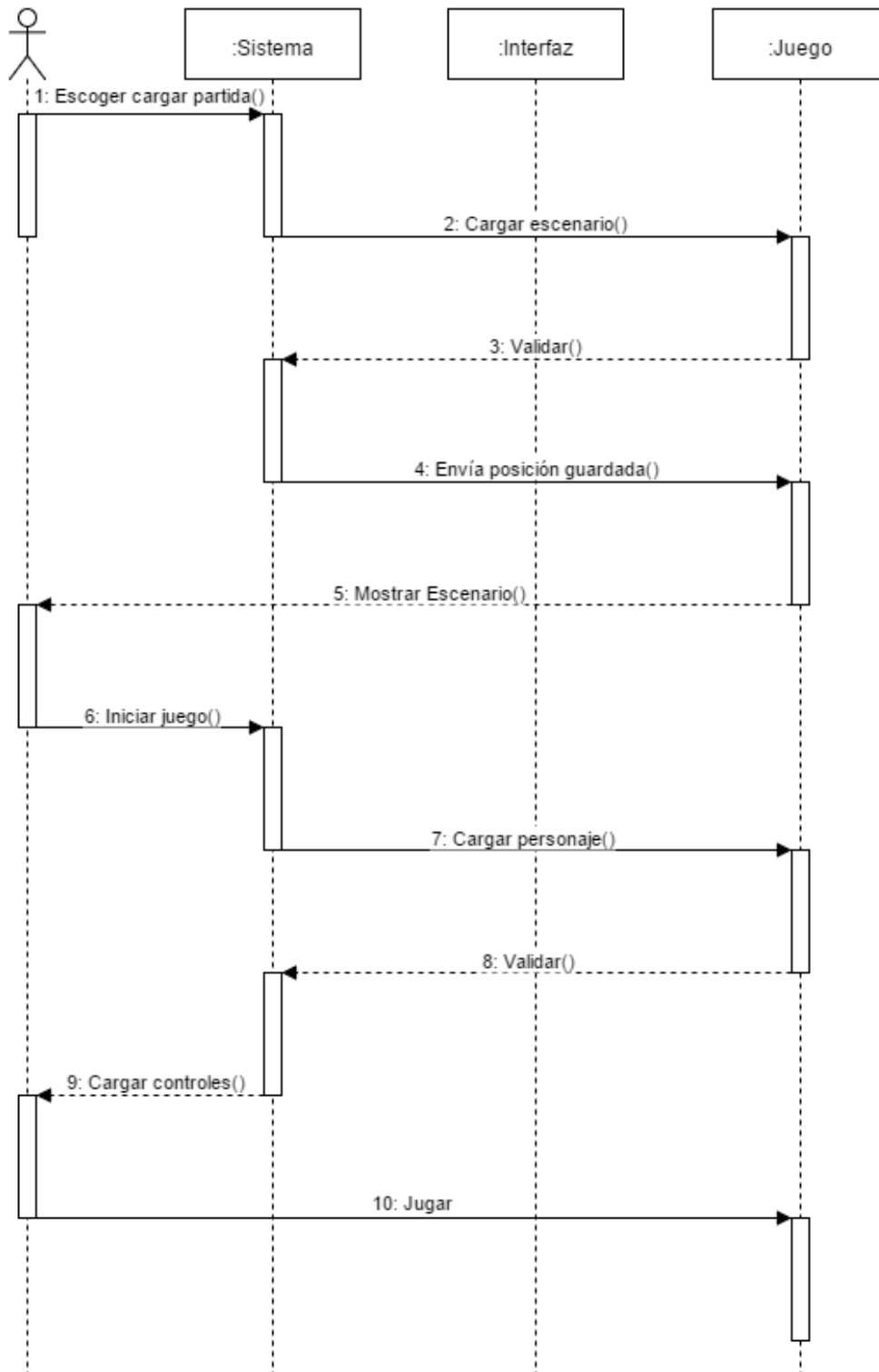


Figura N°33: Diagrama de secuencia 3 – Cargar partida. Imagen autoría propia.

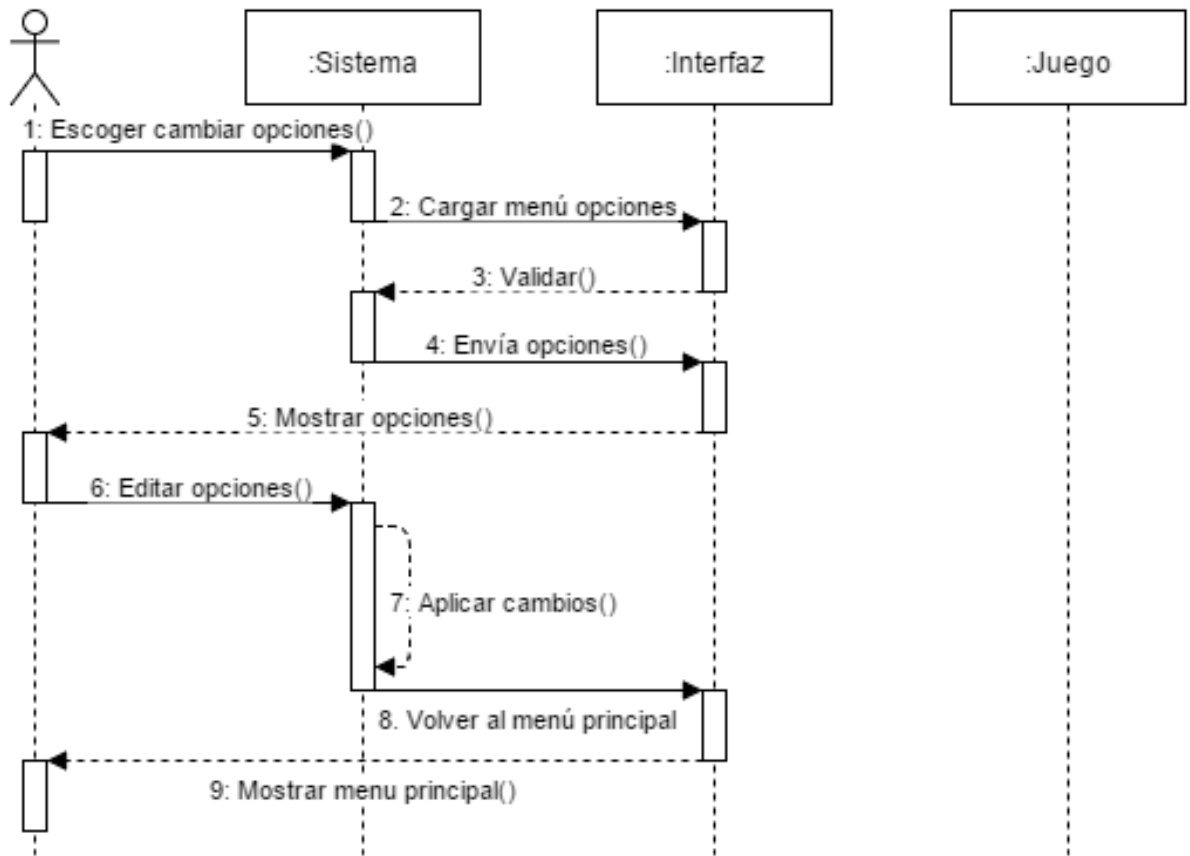


Figura N°34: Diagrama de secuencia 4 – Cambiar opciones. Imagen autoría propia.

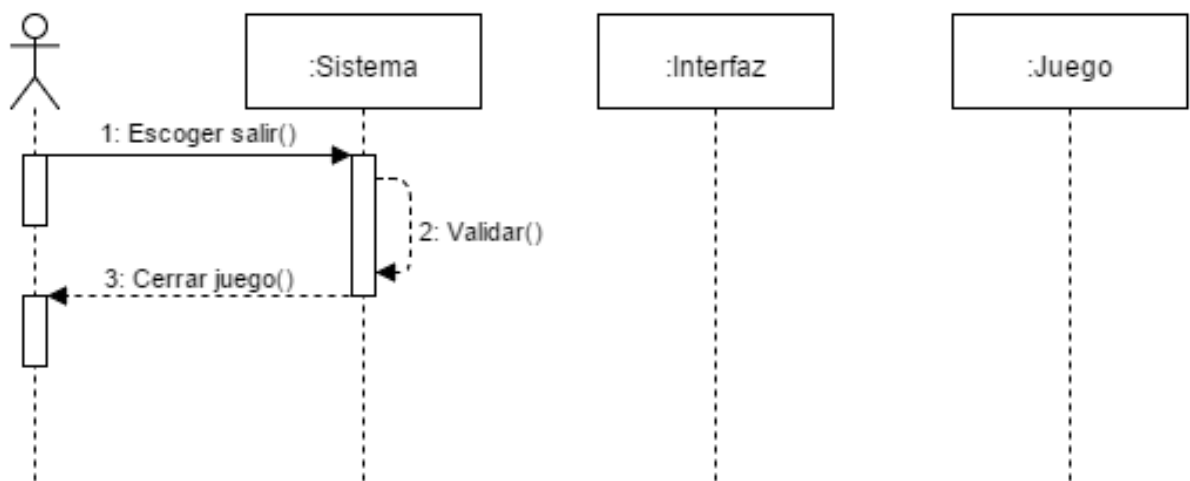


Figura N°35: Diagrama de secuencia 5 – Salir. Imagen autoría propia.

4. Desarrollo

En este apartado se tiene en cuenta dos aspectos importantes para el funcionamiento correcto de la aplicación, las especificaciones técnicas tanto del software como del hardware, necesarias para correr el juego y medir la capacidad de procesamiento, el impacto dentro de los diferentes dispositivos y el alcance que puede o no llegar a tener.

4.1 Especificaciones técnicas:

A continuación, se brindará una lista de componentes de hardware y software que son requerimientos mínimos para brindar la mejor experiencia entre el usuario y la aplicación, de ahí en adelante estos pueden cambiar según el gusto del usuario. Como se observa son requerimientos bastante bajos y va a permitir a más gente llegar a probar o consumir el videojuego, gracias al desarrollo en HTML5 y el uso apropiado de recursos en esté.

4.1.1 Software:

1. Navegador web compatible con HTML5.
2. Sistema operativo Windows.
3. DirectX actualizado.

4.1.2 Hardware:

1. Internet superior a 1 Mb.
2. Espacio en disco duro mínimo de 100 MB.
3. Memoria RAM mínimo de 256 MB.
4. Tarjeta gráfica mínimo de 128 MB con controladores actualizados.
5. Procesador mínimo Pentium 4 2500MHz.

5. Glosario

WWW: Sigla de la expresión inglesa World Wide Web, 'red informática mundial', sistema lógico de acceso y búsqueda de la información disponible en Internet, cuyas unidades informativas son las páginas web.

HTML5: (HyperText Markup Language, versión 5) es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML.

CSS: (Cascading Style Sheets, u Hojas de Estilo en Cascada) es la tecnología desarrollada por el World Wide Web Consortium (W3C) con el fin de separar la estructura de la presentación, aun así, se podía asegurar una visualización correcta de la misma hoja de estilos.

JavaScript: (abreviado comúnmente "JS") es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

UML: (Lenguaje Unificado de Modelado) es un estándar para la representación de procesos o esquemas de software (programas informáticos).

Navegador: Programa que permite navegar por internet u otra red informática de comunicaciones.

Videojuego: Juego electrónico que se visualiza por medio de una pantalla.

2D: Este término describe lo bidimensional, es decir; aquello que sólo tiene dos dimensiones. Aquello que se proyecta de manera plana en el espacio físico.

WEB: Conjunto de información que se encuentra en una dirección determinada de internet.

Sistema operativo: Conjunto de órdenes y programas que controlan los procesos básicos de una computadora y permiten el funcionamiento de otros programas.

Aplicación: Programa o conjunto de programas informáticos que realizan un trabajo específico, diseñado para el beneficio del usuario final.

Hipertexto: Sistema de organización y presentación de datos que se basa en la vinculación de fragmentos textuales o gráficos a otros fragmentos, lo cual permite al usuario acceder a la información no necesariamente de forma secuencial sino desde cualquiera de los distintos ítems relacionados.

Multimedia: Que reúne distintos medios integrados, como audio, video y televisión.

RPG: (Role Playing Game u juego de rol) Se trata de un juego que lleva a los participantes a asumir un rol o papel, interpretando a un personaje.

Indie: Es un término inglés que significa independiente o independencia.

Motor de videojuegos: Un motor de videojuego es un término que hace referencia a una serie de rutinas de programación que permiten el diseño, la creación y la representación de un videojuego. Del mismo modo existen motores de juegos que operan tanto en consolas de videojuegos como en sistemas operativos y para dispositivos móviles.

NPC: Acrónimo del término inglés Non Playable Character (Personaje No Jugable).

Hardware: Conjunto de elementos físicos o materiales que constituyen una computadora o un sistema informático.

Software: Conjunto de programas y rutinas que permiten a la computadora realizar determinadas tareas.

6. Conclusiones

El desarrollo de un videojuego requiere un conocimiento previo importante para saber afrontar las limitaciones técnicas que se pueden encontrar, además de un estudio inicial que permita ver más allá de la idea planteada, esto sin contar el equipo de trabajo además de desarrolladores que se requiere en un proyecto de este tipo.

La metodología Huddle es casi perfecta para el desarrollo de videojuegos, permite un proceso ágil, una facilidad increíble para adaptarse a los cambios que se quieran hacer, ofrece las herramientas necesarias y evita el contenido extra o basura que pueden ofrecer otras metodologías para este tipo de aplicaciones.

Se logró conocer varios motores de videojuegos gracias a la investigación previa al desarrollo del proyecto, Construct2 es un motor bastante potente para varios tipos de videojuegos y se planea el salto a Unity para el desarrollo de videojuegos en tres dimensiones, motor que posiblemente es el más grande usado por desarrolladores independientes en la actualidad, pero nunca se descarta el uso de herramientas más potentes.

El desarrollo independiente de videojuegos tiene gran fuerza en estas últimas generaciones gracias al abandono de conceptos básicos por parte de algunas empresas elite y la gran cantidad de herramientas tecnológicas al alcance de todos.

Castle of Chaos es un proyecto que en versiones posteriores puede llegar a triunfar en el mercado y calar en el gusto de muchos usuarios, se espera poder seguir trabajando en él y lograr metas más grandes.

La redacción de un documento de esta índole para terminar la carrera universitaria es una experiencia que no se puede obtener de otra manera, la investigación previa, el desarrollo de un proyecto profesional y el proceso de escribir de una forma más seria, son cosas que servirán inminentemente para el futuro.

7. Bibliografía

Morales Urrutia Gerardo Abraham, Nava López Claudia Esther, Fernández Martínez Luis Felipe, Rey Corral Mirsha Aarón. Universidad Autónoma de Ciudad Juárez de México (2010). Metodología Huddle. Disponible en la dirección: www.fadp.edu.co/publicaciones/proyinvmul-03.docx

Caulfield Brian. Nvidia (2013). Encuesta Pc. Disponible en la dirección: <http://blogs.nvidia.com/blog/2013/06/13/seven-reasons/>

Ortega José. IGN (2015). Historia Metroidvania, Castlevania y Metroid. Disponible en la dirección: <http://es.ign.com/reportaje/92521/feature/repaso-a-la-historia-de-los-metroidvania>

KDE Documentation. Lenguaje de modelado unificado. Disponible en la dirección: <https://docs.kde.org/trunk4/es/kdesdk/umbrello/uml-basics.html>

Gullen Thomas, Gullen Ashley. Scirra (2011) Información Construct2. Disponible en la dirección: <https://www.scirra.com/construct2>

Garcerant Iván. Synergix (2008). Requisitos funcionales y no funcionales. Disponible en la dirección: <https://synergix.wordpress.com/2008/07/07/requisito-funcional-y-no-funcional/>

Parra Mariana. Secundarios (2015). Géneros de videojuegos. Disponible en la dirección: <http://secundarios.com/2015/01/04/manual-para-newbies-que-generos-de-videojuegos-existen/>

Gambin Redondo Alfonso. BlastingNews (2014). Videojuegos encima del cine y la música. Disponible en la dirección. <http://es.blastingnews.com/economia/2014/12/el-videojuego-lidera-la-industria-del-ocio-audiovisual-superando-al-cine-o-la-musica-00217253.html>

Owens Martin. InkSkape (2013). Información InkSkape. Disponible en la dirección: <https://inkscape.org/es/>

Sopitas (2013). Jugadores actuales vs jugadores clásicos. Disponible en la dirección. <http://www.sopitas.com/233725-videojugadores-actuales-no-pueden-pasar-el-nivel-1-1-de-super-mario-bros/?from=site>

Shiffman Daniel. (2012). Nature of code. Disponible en la dirección. <http://natureofcode.com/book/>

Hawkins Trip. (2007). Videogame marketing and pr. Disponible en la dirección. <http://www.sellmorevideogames.com/VideogameMarketingAndPR.pdf>

Barinaga Lopez Borja. (2010). Juego, historia, teoría y práctica del diseño conceptual de videojuegos. Disponible en la dirección. <http://www.danielparente.net/info/uploads/sites/2/2013/09/Juego-Borja-Lopez-Barinaga.pdf>

Tejedor Navarro Heliodoro, Duch i Gavalda. (2012). Sonido, interacción y redes. Disponible en la dirección. [https://www.exabyteinformatica.com/uoc/Informatica/Sonido_interaccion_y_redes/Sonido_interaccion_y_redes_\(Intro\).pdf](https://www.exabyteinformatica.com/uoc/Informatica/Sonido_interaccion_y_redes/Sonido_interaccion_y_redes_(Intro).pdf)

Gonzales Daniel. (2014). Arte de videojuegos, da forma a tus sueños. Disponible en la dirección. <http://pdflibro.com/arte-de-videojuegos-da-forma-a-tus-suenos/>

Thompson Jim, Gili Gustavo, Berbank-Green Bernaby. (2008). Videojuegos, manual para diseñadores gráficos. (2008). Disponible en la dirección. <http://www.casadellibro.com/libro-videojuegos-manual-para-disenadores-graficos/9788425222665/1224541>

8. Manuales

A continuación, se explicará detalladamente la transición entre los diferentes módulos del juego, el funcionamiento de este y algunos aspectos extra que facilitarían el entendimiento de la aplicación.

8.1 Manual del sistema:

En esta sección se abarcan dos aspectos importantes del videojuego, por una parte, se encuentra la interfaz de la aplicación (Detallada previamente en este proyecto.) y los diferentes medios multimedia usados.

8.1.1 Interfaz:

Previo al desarrollo del videojuego se identificaron las diferentes pantallas que el usuario va a encontrarse, en este caso se presentan los bocetos de cada una y el funcionamiento de sus diferentes partes, más que el funcionamiento que ya fue explicado se mencionara lo que se puede encontrar en cada imagen.

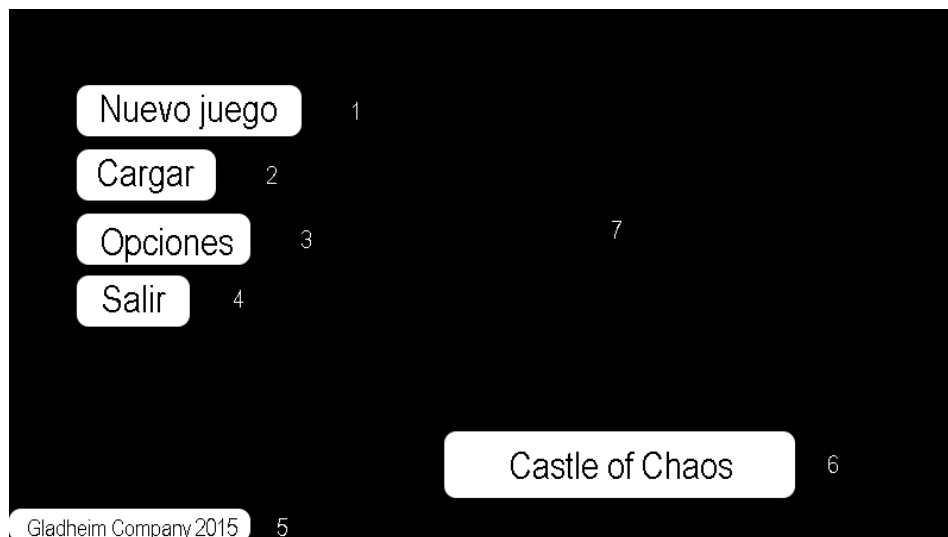


Figura N°36: Boceto interfaz menú. Imagen autoría propia.

1. Nuevo juego: Botón que permite al jugador iniciar una partida desde cero en el punto por defecto en el escenario.
2. Cargar: Botón que permite al jugador iniciar una partida desde un punto de guardado existente.
3. Opciones: Botón que lleva al jugador a una pantalla donde puede editar algunas características básicas del juego.
4. Salir: Botón que permite al jugar salir de toda la aplicación.
5. Gladheim Company 2015: Texto que indica el nombre del estudio desarrollador.
6. Castle of Chaos: Texto a modo de título que indica el nombre del videojuego.
7. Fondo: Imagen de fondo que da la bienvenida al usuario.

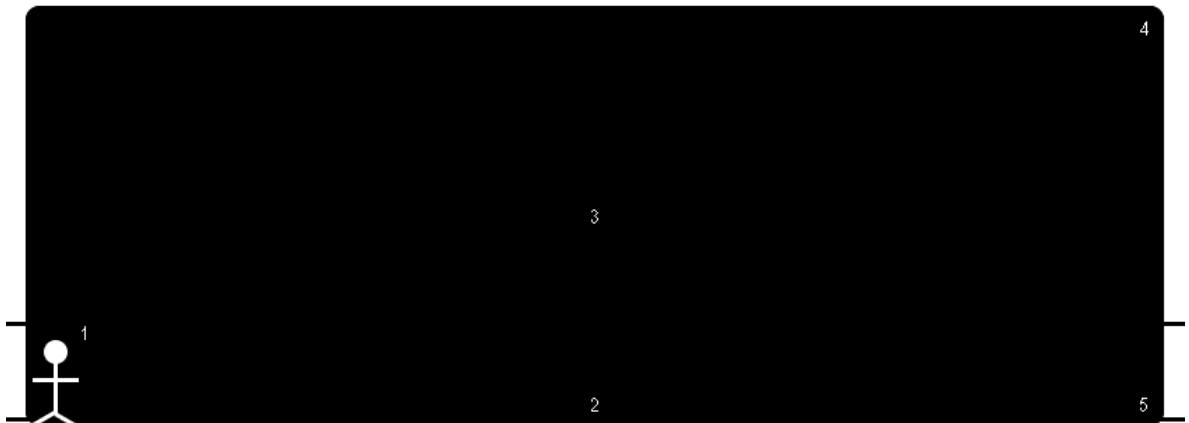


Figura N°37: Boceto interfaz escenario. Imagen autoría propia.

1. Personaje: Es la persona que es controlable por el jugador.
2. Suelo/Plataforma: Es el lugar donde se puede mantener en pie el personaje, puede ser a nivel del piso o plataformas de salto.
3. Fondo: Imagen de fondo que brinda decoración de entorno a cada escenario.
4. Muros: Es el contorno de cada escenario
5. Puerta: Imagen básica que sirve de puerta para separar cada escenario.

8.2 Manual del usuario:

El objetivo de este juego es avanzar por una serie de escenarios sorteando tanto enemigos como pruebas de habilidad lógica y mecánica, a la vez se necesita recoger armas y habilidades para poder avanzar por caminos donde de otra forma no se podría.

8.2.1 Personajes:

Personaje Principal

Nombre: *Lucían Belmont.*

Edad: 40 Años.

Peso: 80 Kg.

Altura: 1.80 cm.

Características Psicológicas: Confiado/Creído, Impulsivo, Arrogante, Caballero, Servicial, Estratega, Serio.

Características Físicas: Alto, Fornido, Atlético, Cabello castaño medio largo, Piel blanca.

Descripción: Lucían es el fruto de varias generaciones de cazadores de monstruos (Aunque al principio del juego él no tiene idea de esto.) que dedico la mayor parte de su vida a investigar diferentes hechos por todo el mundo, desde secuestros hasta asesinatos. Gracias a su experiencia de campo, Lucían logro un nivel deductivo bastante alto que lo ayudara a enfrentar cualquier situación; Esto junto a la buena forma física que mantiene son los puntos esenciales que se tomaron en cuenta para llamarlo a investigar los recientes hechos en el juego. Un personaje que, aunque confiado de sus habilidades, siempre va un paso delante de lo que sucede a su alrededor. Lo acompaña una pulsera con una cruz y una gabardina larga que cubre todo su cuerpo, sin duda su mayor preocupación en esta historia no es solo el sentirse conocido en aquella mansión misteriosa sino salvar a tantas personas de un mal tan grande que podría acabar con la vida en la tierra.

Personajes Secundarios

Nombre: Tomas Zoren.

Edad: 70 Años/Aparentemente.

Peso: 90 Kg.

Altura: 1.70 cm.

Características Psicológicas: Tranquilo, Misterioso, Cordial.

Características Físicas: Mediano, Robusto, Cabello café corto, Piel blanca.

Descripción: Tomas es la única persona que Lucían encuentra en el pueblo en donde inicia la investigación, un anciano misterioso que recibe cordialmente a todos los que visitan el lugar sin que ellos sepan sus verdaderas intenciones, sin muchas palabras invita a los turistas a conocer la mansión de la que nunca salen. Lleva consigo un bastón y una sonrisa falsa que adorna su rostro en todo momento, no habría manera de que la gente desconfiara de alguien así, por lo que su cuartada es perfecta, un servidor menor del mal que domina la zona.

Nombre: Luna Rigald.

Edad: 25 Años.

Peso: 55 Kg.

Altura: 1.65 cm.

Características Psicológicas: Tierna, Madura.

Características Físicas: Mediana, Delgada, Cabello negro largo, Piel blanca.

Descripción: Luna es una de las tantas víctimas dentro del juego, Alexander se encuentra con ella poco más tarde de entrar en la mansión, se encuentra prisionera de Muerte. Luna es una chica tierna que tuvo la mala suerte de caer en manos equivocadas, pese a encontrarse con Alexander al inicio de su recorrido, Muerte se la lleva a otra zona donde tendrá que esperar a que su héroe venga a salvarla. Luna se ofreció voluntariamente para ayudar con la situación misteriosa que ocurría en el pueblo, pero nunca imaginó lo que estaba pasando.

Nombre: Gilean Ferris.

Edad: 32 Años/Aparentemente.

Peso: 70 Kg.

Altura: 1.70 cm.

Características Psicológicas: Leal, Caballero, Servicial, Avaro.

Características Físicas: Mediano, Delgado, Cabello negro largo, Piel blanca.

Descripción: Gilean es un siervo de la mansión, que se sorprende al ver que un descendiente de sus antiguos amos (Alexander Belmont) es quien viene a terminar con el mal que lo tiene esclavizado, Gilean durante toda su estadía a coleccionado muchas de las cosas que encuentra abandonadas por la mansión, desde armas hasta accesorios misteriosos dejados por monstruos, así que aunque físicamente no puede ayudar, se convierte en un poderoso aliado que vende todo tipo de objetos a nuestro protagonista para facilitarse las cosas en su lucha. Carga consigo un libro donde anota todas sus tareas pendientes y un anillo de oro que al parecer mantiene su alma esclavizada en este mundo.

8.2.2 Niveles:

Escenario entrada.

El jugador se encuentra con este escenario justo al empezar la partida.

Es una pantalla pequeña sin obstáculos, simboliza la entrada al castillo y se muestra al jugador los controles básicos e información importante para la inmersión en la historia, no tiene que derrotar a ningún amigo ni lograr algo específico para avanzar, más que observar los consejos para aprender los controles, es una pantalla de transición y tutorial.

En este escenario tampoco se pueden encontrar objetos o armas de ningún tipo.

También es el escenario donde inicia la reproducción de la música de fondo, para este caso es una melodía con estilo retro, que tiene un inicio suave hasta tomar un ritmo constante.



Figura N°38: Escenario1. Imagen autoría propia.

Escenario calabozo.

El jugador se encuentra con este escenario al pasar la puerta del escenario entrada.

Esta pantalla resume algunas de las mecánicas esenciales para todo el juego, obtención de armas para derrotar a algunos enemigos y saltar plataformas para llegar de un lado a otro.

Conserva la melodía del escenario inicial.

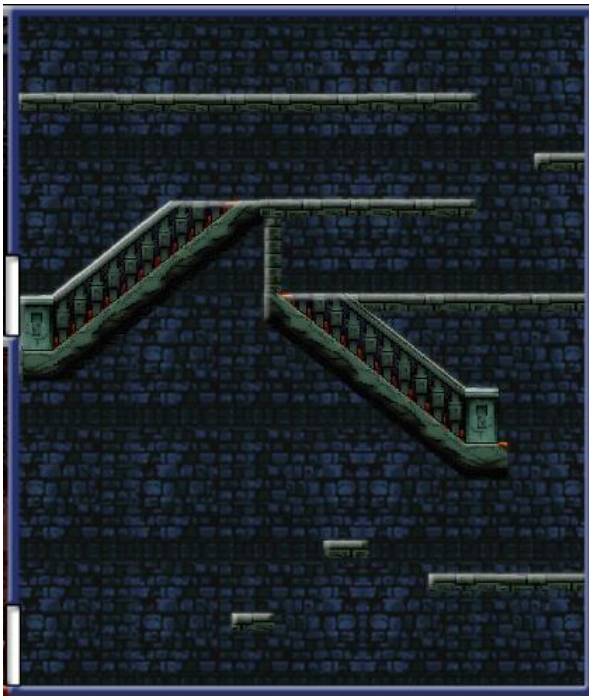


Figura N°39: Escenario2. Imagen autoría propia.

Escenario fuerte.

El jugador se encuentra con este escenario al pasar la puerta del escenario calabozo.

Esta pantalla es previa al jefe final, contiene otra de las cualidades básicas de este tipo de juegos, la obtención de habilidades, también se encuentra algunos monstruos y plataformas para poder atacar o sortear a estos enemigos.

Conserva la melodía del escenario inicial.



Figura N°40: Escenario3. Imagen autoría propia.

Escenario jefe final.

El jugador se encuentra con este escenario al pasar la puerta del escenario fuerte.

Esta pantalla es donde el jugador debe enfrentar al primer enemigo épico, tiene algunas plataformas para esquivar ataques y moverse por el escenario, al terminar con él finaliza el juego.

En este escenario la música cambia para un ambiente de batalla rápido.



Figura N°41: Escenario4. Imagen autoría propia.

8.2.3 Enemigos:

Momia.

Tiene una altura semejante a la del personaje principal, rodeado de vendas en todo su cuerpo. Ataca directamente con su cuerpo persiguiendo al jugador en las plataformas donde las dos entidades se encuentren, se mueve de forma horizontal.

Se encuentran en el escenario calabozo.

Ojo demoniaco.

Es un poco más grande que el personaje, es simplemente un ojo gigante que recorre el escenario para atacar al jugador, no posee ninguna habilidad especial recalable, ya que ataca cuerpo a cuerpo, se mueve de forma vertical.

Se encuentra en el escenario fuerte.

Muerte.

Es del doble del tamaño del personaje, es un monstruo gigante que vuela y ataca con diferentes habilidades al jugador, tanto con su arma como lanzando poderes, puede moverse en varias direcciones.

Se encuentra en el escenario jefe final.

8.2.4 Habilidades:

Para este proyecto se ideó una sola habilidad que demuestra el funcionamiento de este tema en el juego, para este caso se desarrolló el doble salto, que el jugador puede encontrar en el escenario fuerte y le permitirá avanzar más fácil por el escenario. Hay que recalcar que el jugador debe interactuar con la antorcha dentro del escenario para adquirir la habilidad y desde ese momento será capaz de duplicar la altura de su salto original. En versiones posteriores el juego debe incluir muchas más habilidades con diferentes efectos en el escenario, en el personaje o en los enemigos.

8.2.5 Objetos:

Para este proyecto se ideó solamente la obtención del arma principal del personaje, la cual es un látigo, capaz de acabar con monstruos de otros mundos (Arma muy reconocida en Castlevania.), el jugador puede encontrar este objeto en el escenario calabozo. Al igual que en el caso de las habilidades, el jugador debe interactuar con la antorcha del escenario para poder adquirir este beneficio, previo a esto el personaje no cuenta con ningún arma.

8.2.6 Trucos:

Como extra, se ideó un único truco para el juego, el modo Dios, donde simplemente el jugador no puede morir, obtiene el máximo nivel, desbloquea todas las armas y habilidades y justamente a modo de tributo se hace de la misma forma en la que Konami lo hizo en sus primeros juegos.

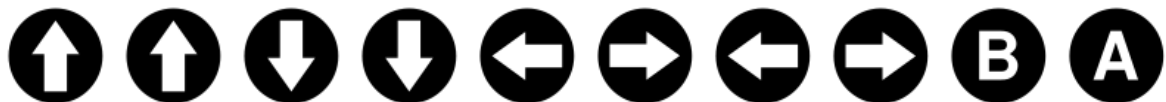


Figura N°42: Código Konami. Imagen autoría de Konami.

8.2.7 Guion:

Este guion fue pensado para una fase muy inicial del juego, pudo cambiar o no en el prototipo funcional del videojuego.

Gracias a la cantidad de géneros que se encuentran en este videojuego, es sencillo pensar una historia para el mismo, se ideó algo básico siguiendo los estándares de los juegos Castlevania, fácilmente reconocible por sus seguidores. El punto creativo se encuentra en la forma en la que el personaje conecta con su vida pasada, en un principio sin conocer nada, y termina utilizando esa información para lograr algo que ya hizo hace muchos años.



Figura N°42: Boceto guion. Imagen autoría propia.

8.2.8 Datos de producción:

Este prototipo fue desarrollado en seis meses aproximadamente, contando la etapa de preproducción donde solo se tiene la idea como tal. El presupuesto es un tema que se planea estudiar detenidamente más adelante con más cuidado.