



Universidad de las Ciencias Informáticas

Facultad 4

Trabajo de Diploma para optar por el título de  
Ingeniero en Ciencias Informáticas

Videojuego de supervivencia para el enriquecimiento mediante  
productos nacionales de la plataforma Cosmox.

Autores: Kevin Guimera Abad

Daniel Bagué Calvo

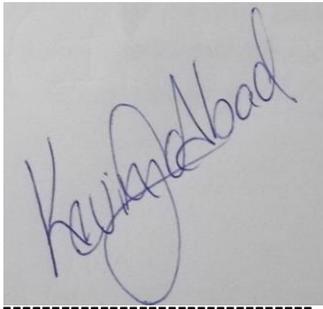
Tutor: Ing. Enelis Blanca Cuba Rondón

La Habana, diciembre de 2022

## DECLARACIÓN DE AUTORÍA

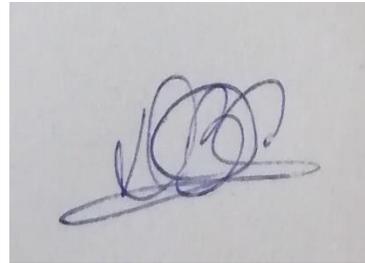
Declaramos ser autores únicos de este trabajo y autorizamos a nuestra tutora Ing. Enelis Blanca Cuba Rondón y al Centro de Tecnologías Interactivas (VERTEX), de la Universidad de las Ciencias Informáticas, para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los 7 días del mes de diciembre del año 2022



Kevin Guimera Abad

Autor



Daniel Bague Calvo

Autor



Ing. Enelis Blanca Cuba Rondón

Tutor

## **DEDICATORIA**

A mis padres, a mi mamá que donde quiera que esté, espero esté orgullosa de mí; a familiares y amigos, especialmente aquellos que hicieron esto posible.

Kevin

A mi madre, que durante toda mi vida me ha impulsado a mejorar en todos los aspectos de mi vida y sin importar cuanto he caído me ha dicho como volver a levantarme; a mi abuela, que aún sin tener la lucidez de años atrás siempre está pendiente de mis avances; a mi padre, que incluso en la lejanía pone todo su empeño en que sea mejor que él en la vida y a mi padrastro que más allá de todo ha sido y siempre será un padre para mí, apoyando sus manos he impulsándome hacia lo más alto con sus precisos consejos.

Daniel

## **AGRADECIMIENTOS**

A todos los que dieron su granito de arena, a familiares, amigos y a mí por ser fuerte y resistir. Y porqué no, a los que hoy no son nada, pero fueron todo y parte de esto es suyo también.

Kevin

Ha sido largo el viaje durante todos estos años de carrera, cayendo y levantándome, volver a caer, pensar que no podía lograr el sueño que me propuse. Por ello, quisiera agradecer a mi madre por toda la paciencia que ha puesto en mí, todo ese amor y cariño, a mi abuela por ser a luz que desde pequeño me ha mantenido a flote, a mi padre por ser junto a mi madre mi más grande orgullo, mi padrastro por siempre aparecer fuerte cuando lo he necesitado, a mi hermana, porque he querido ser el ejemplo de superación que se merece. A mi hermano de la vida, Leo, quien siempre me trata como a su hermano menor y a quién le tengo un enorme cariño e infinita confianza. También a todas aquellas personas que me he encontrado en el camino, a Evelyn, por hacerme saber lo que se siente ser amado y amar, a Kevin por ser con quien comparto este logro y ser un gran amigo y modelo de superación, a todos mis ángeles del 44102, entre ellas: Arisdalia, tú siempre tan enérgica y loca, haciendo que aquellos momentos de duda fueran momentos de gozo, a las dos Mari que siempre parecen estar en guerra pero se llevan tan bien, a Lia con la cuál he aprendido mucho, conversado más aún. A Yanita, por siempre escuchar y decirlo todo sin tapujos y hacerme reír, a Xavi, por ser tan espléndido amigo y consultor, a Emilio que más que un profesor ha sido un tutor más, a nuestra tutora Enelis que a pesar de sus ocupaciones siempre nos encontró un tiempo para corregir nuestra investigación. Al final, pero no menos importante, a mí por no rendirme incluso cuando todo mi cuerpo gritaba por ello, por seguir adelante en aquello que decidí por cuenta propia lograr incluso cuando muchos me aconsejaron no hacerlo.

Daniel

## RESUMEN

La industria de videojuegos en Cuba durante los últimos años ha mejorado, pero aún se encuentra en una posición en la que no ofrece una solución a la demanda nacional a pesar de las estrategias trazadas por distintas instituciones y grupos o personas ajenas a dichas instituciones. Cosmox es una plataforma para videojuegos en línea, creada para los usuarios cubanos, desarrollada en la Universidad de Ciencias Informáticas (UCI) con el fin de satisfacer la creciente demanda de productos a nivel nacional y a su vez mantener nuestra identidad, cultura e ideales. El desarrollo de esta investigación se centra en culminar un videojuego en primera persona del género supervivencia mediante el estudio de las principales características y mecánicas de dicho género a través de productos ya existentes en el mercado mundial, para enriquecer el catálogo de productos de la plataforma Cosmox. Para dicho propósito se utilizó como motor gráfico Unity, en el que se ensamblarán todas las partes y componentes necesarios para crear el videojuego. Para mantener una organización adecuada se utilizó como metodología el Marco de desarrollo ingenieril para el proceso de desarrollo de videojuegos y se seleccionó la arquitectura por Capas al ser la más adecuada según las necesidades y características de la solución. Se validó la solución a partir de la aplicación de pruebas Alfas y Betas.

**Palabras clave:** primera persona; supervivencia; videojuego; Unity

## **ABSTRACT**

The video game industry in Cuba in recent years has improved, but it is still in a position in which it does not offer a solution to the national demand despite the strategies drawn up by different institutions and groups or people outside these institutions. Cosmox is a platform for online video games, created for Cuban users, developed at the University of Informatics Sciences (UCI) in order to satisfy the growing demand for products at the national level and in turn maintain our identity, culture and ideals. The development of this research focuses on completing a first-person game of the survival genre by studying the main characteristics and mechanics of this genre through existing products on the world market. For this purpose, Unity was used as the graphic engine, in which all the parts and components necessary to create the video game will be assembled. To maintain an adequate organization, the engineering development framework for the video game development process was used as a methodology and the Layered architecture was selected as it was the most appropriate according to the needs and characteristics of the product to be developed. The software tests were carried out on the solution proposal, which demonstrated the quality of the developed video game and validated the fulfillment of the objective of the present investigation.

**Keywords:** mechanics; survival; videogame; Unity

**ÍNDICE**

INTRODUCCIÓN	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA DE LOS DESARROLLOS DE VIDEOJUEGOS DE SUPERVIVENCIA	4
1.1 Desarrollo de videojuegos	4
1.1.1 Géneros de videojuegos	4
1.1.2 Género acción	5
1.1.3 Videojuegos de acción de subgénero supervivencia	6
1.2 Mecánicas de videojuegos	6
1.3 Análisis de los videojuegos homólogos Rust y Minecraft	8
1.3.1 Mecánicas elementales de videojuegos del género supervivencia	11
1.4 Metodología, herramientas y tecnologías	12
1.5 Conclusiones parciales	16
CAPÍTULO 2. CONCEPTUALIZACIÓN Y DISEÑO DE UN VIDEOJUEGO DE SUPERVIVENCIA	17
INTRODUCCIÓN	17
2.1 Propuesta de solución	17
2.2 Resumen del videojuego	17
2.3 Diseño del videojuego	17
2.3.1 Elementos formales del videojuego	18
2.3.2 Elementos dramáticos	20
2.4 Diseño y descripción de las pantallas gráficas	21
2.5 Especificación de mecanismos	23
2.5.1 Especificación de mecanismos del videojuego	24
2.6 Requisitos no funcionales	26
2.7 Paquete de Mecanismos	27
2.7.1. Descripción de paquetes de mecanismos	28
2.8 Arquitectura de software	29
2.9 Diagrama de Clases	30
2.10 Patrones de Diseño	31
2.11 Representación del Comportamiento	31
	VII

2.12 Conclusiones Parciales	32
CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBAS DE UN VIDEOJUEGO DE SUPERVIVENCIA	33
3.1 Estándar de codificación	33
3.2 Diagrama de componentes	34
3.3 Fase de pruebas	35
3.4. Conclusiones parciales	37
CONCLUSIONES	38
RECOMENDACIONES	39
REFERENCIAS	40
ANEXOS	42
A.1 Diagrama de clases	42
A.1.2 Diagrama de clase del mecanismo interfaz	42
A.1.3 Diagrama de clase del mecanismo locomoción	42
A.1.4 Diagrama de clase del mecanismo barra de vida	43
A.1.5 Diagrama de clase del mecanismo barra de hambre	43
A.1.6 Diagrama de clase del mecanismo cámara	44
A.1.7 Diagrama de clase del mecanismo creación	44
A.1.8 Diagrama de clase del mecanismo inventario	45
A.1.9 Diagrama de clase del mecanismo abandonar	45
A.2 Errores y no conformidades registradas durante las pruebas alfas	45
A.3 Diagramas de estado	49
A.3.1 Diagrama de estado del mecanismo interfaz principal	49
A.3.2 Diagrama de estado del mecanismo abandonar	49
A.3.3 Diagrama de estado del mecanismo inventario	50
A.3.4 Diagrama de estado del mecanismo locomoción	50
A.3.5 Diagrama de estado del mecanismo crear objeto	50
A.3.6 Diagrama de estado del mecanismo combate	51
A.3.7 Diagrama de estado del mecanismo gestión de vida	51
A.3.8 Diagrama de estado del mecanismo gestión de hambre	51

A.3.9 Diagrama de estado del mecanismo misión	52
A.4 Diagrama de componentes	52
A.4.1 Diagrama de componente del mecanismo gestionar interfaz	52
A.4.2 Diagrama de componente del mecanismo locomoción	52
A.4.3 Diagrama de componente del mecanismo administrar inventario	53
A.4.4 Diagrama de componente del mecanismo crear objeto	53
A.4.5 Diagrama de componente del mecanismo administrar misión	53
A.4.6 Diagrama de componente del mecanismo gestionar vida	54
A.4.7 Diagrama de componente del mecanismo gestionar hambre	54
A.4.8 Diagrama de componente del mecanismo cámara	54
A.4.9 Diagrama de componente del mecanismo combate	55
A.5 Guía de jugabilidad del videojuego “La Isla”	55

## INTRODUCCIÓN

La industria de los videojuegos se ha visto aún más impulsada en 2021 por la pandemia de coronavirus. Y es que terminó el pasado año con un crecimiento del 20% y una facturación global de 200.000 millones de dólares. Se prevé que el entorno gaming seguirá expandiéndose y registrando altas cifras de consumo; las razones son amplias y variadas: disfrute, desconexión de la realidad, socialización, el atrapamiento de los sentidos, inmersión en una experiencia solo alcanzable a través del videojuego. Otro motivo clave es el de la autosuperación y el éxito. En la mayoría de videojuegos, se debe intentar vencer todos sus desafíos. Superar el juego en un ambiente de dificultad moderado y que a menudo se autoajusta a la habilidad del jugador. Pretenden ser complejos, para estimular la competitividad, pero no tanto como para que lo dé por imposible y renuncie a continuar intentándolo (Amberes, 2017) (Mordor Intelligence, 2022).

El aumento en el mercado de los videojuegos no sólo ha influido en el dinero ingresado, en el número de productos o en el aumento de las empresas de este sector, sino también ha provocado que éste sea un producto mediático y con un gran atractivo. Esto se debe, en gran medida, a su carácter interactivo y su capacidad de comunicación. Hace ya tiempo que generan más ingresos que el cine y el deporte juntos (Roberto Hernández, 2020).

Uno de los objetivos del proceso de informatización de la sociedad en Cuba es proveer contenidos nacionales para el entretenimiento, capaces de transmitir al público cubano un contenido acorde a los principios y valores de nuestra sociedad. En este sentido, los videojuegos resultan una alternativa atractiva e innegable. En Cuba se trabaja para impulsar la industria de videojuegos. Como explicara el Dr. C Hassán Lombera Rodríguez, director del Centro de Tecnologías Interactivas (VERTEX) de la Universidad de las Ciencias Informáticas (UCI), se desarrolla actualmente la plataforma de servicios para videojuegos Cosmox, con un conjunto de servicios orientados a satisfacer un grupo de demandas de los desarrolladores de videojuegos, en la cual se incluyen pasarela de pago, partidas en línea para videojuegos multijugador por turno y en tiempo real, torneos, así como un conjunto de herramientas para desarrolladores terceros y jugadores. «El desarrollo de esta plataforma sentará las bases para, de conjunto con el Parque Científico Tecnológico de La Habana, impulsar el crecimiento de la naciente industria de los videojuegos en Cuba, con productos que muestren la grandeza de nuestra historia e identidad nacional», destacó (Granma, 2021.).

En la actualidad, no son muchos los productos que tiene disponible esta plataforma, la misma cuenta con quince videojuegos y el Cosmox *Game Center* que incluye siete minijuegos. Sumado al llamado de

enriquecer la plataforma Cosmox; Se tiene como **problema de investigación**: ¿Cómo enriquecer el catálogo de productos de Cosmox?

Por lo tanto, como **objeto de estudio** se tiene: desarrollo de videojuegos.

El **objetivo general de la investigación** será: desarrollo de un videojuego de supervivencia para el enriquecimiento mediante productos nacionales de la plataforma Cosmox.

De esta manera el **campo de acción** queda enmarcado como: Proceso de desarrollo de videojuegos de tipo supervivencia.

Para dar solución al problema de investigación se plantean las siguientes tareas de investigación:

- Construcción del marco teórico de la investigación, relacionado al desarrollo de videojuegos tipo supervivencia.
- Selección de técnicas a utilizar para el desarrollo del videojuego y sus características.
- Definición de la propuesta de solución a través del levantamiento de sus requisitos.
- Desarrollo de los artefactos establecidos por la metodología seleccionada.
- Implementación de la jugabilidad del videojuego.
- Realización de pruebas para verificar el correcto funcionamiento.

Para la realización de la investigación y elaboración del presente trabajo se utilizarán varios métodos científicos de investigación, entre los cuales se pueden mencionar:

### **Métodos teóricos:**

**Histórico – Lógico:** se empleó para la fundamentación y sistematización de los aspectos teóricos contemplados en el desarrollo de la investigación acerca de la evolución y las tendencias actuales del desarrollo de mecánicas de videojuegos de tipo supervivencia.

**Analítico – Sintético:** se utilizó para analizar los videojuegos de tipo supervivencia existentes y arribar a conclusiones sobre sus características y mecánicas esenciales.

### **Métodos empíricos:**

**Observación:** se empleó como método referencial al observar el comportamiento de distintos videojuegos de supervivencia, para establecer una comparación y determinar las características y mecánicas comunes que poseen, con la integración a la solución propuesta.

A continuación, se muestra la estructura del presente trabajo, incluyendo una síntesis de los capítulos y secciones fundamentales:

### **Capítulo 1. Fundamentación teórica de los desarrollos de videojuegos de supervivencia**

En este capítulo se realiza un estudio acerca de elementos necesarios para la definición de videojuegos, sus géneros y mecánicas. Se analiza el subgénero de supervivencia, que pertenece al género de acción, para establecer los mecanismos que debe presentar la solución. También se

describen algunas soluciones homólogas existentes, así como metodología y herramientas a utilizar para el desarrollo de la solución.

### **Capítulo 2. Conceptualización y diseño de un videojuego de supervivencia**

En este capítulo se expone la propuesta de solución. También se describen fases y artefactos generados que correspondan a la metodología de desarrollo de software seleccionada. Se describe, además, la arquitectura, la estructura de clases y los patrones empleados para el desarrollo de la misma.

### **Capítulo 3. Implementación y pruebas de un videojuego de supervivencia**

En el capítulo se puede visualizar como se llevó a cabo la implementación del videojuego. Se realizaron pruebas alfas, betas y de integración por el programador para asegurar el correcto funcionamiento del código y de aceptación para verificar si la solución cumplía con las especificaciones.

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA DE LOS DESARROLLOS DE VIDEOJUEGOS DE SUPERVIVENCIA

## Introducción

En este capítulo se definen los principales conceptos asociados al dominio del problema, como videojuegos, sus géneros y dentro de estos se destaca el de supervivencia, sus mecánicas principales y su proceso de desarrollo. Se efectúa el análisis de dos videojuegos de supervivencia para agrupar sus comportamientos principales en mecánicas que el autor define y se realiza un estudio del estado del arte sobre los paquetes existentes para crear videojuegos de supervivencia con Unity 3D, además de describir las principales tecnologías, metodología y herramientas a emplear en el desarrollo de la solución.

## 1.1 Desarrollo de videojuegos

Un **videojuego**, o **juego de vídeo** es un juego electrónico que se visualiza en una pantalla, en el que una o más personas interactúan por medio de un controlador. Los videojuegos recrean entornos y situaciones virtuales en los que el jugador puede controlar a uno o varios personajes para conseguir algún objetivo por medio de reglas determinadas (Alejandro I.,2011) (Behrmann M,2012).

Según Dangeolo Rodríguez, son un tipo de juegos que hacen uso de la tecnología para permitir una interfaz de tipo virtual, la cual ofrece una experiencia interactiva tanto individual o multijugador, esto puede ser por medio de diversos dispositivos que van desde un ordenador, pasando por consolas especializadas, hasta dispositivos móviles inteligentes y tabletas. En la actualidad, jugar videojuegos ha alcanzado un nivel de popularidad de gran magnitud a nivel mundial, incluso convirtiéndose en un negocio global a través de su comercialización (Dangeolo Rodríguez, 2022).

### 1.1.1 Géneros de videojuegos

Los videojuegos se pueden clasificar en géneros atendiendo a factores como el sistema de juego, el tipo de interactividad con el jugador, sus objetivos, entre otros. La evolución de los videojuegos desde sus comienzos ha dado lugar a una variedad creciente y cambiante de géneros, muchas veces en relación con lo que los avances en la tecnología han ido haciendo posible. Entre los géneros de videojuegos más populares están:

- **Acción:** en el que el jugador debe usar su velocidad, destreza y tiempo de reacción (Call of Duty, Quake).

- Estrategia: requieren que el jugador ponga en práctica sus habilidades de planeamiento y pensamiento para maniobrar, gestionando recursos de diverso tipo (materiales, humanos, militares...) para conseguir la victoria. (Warcraft, Dota 2, Age of Empires, Starcraft)
- Rol: donde el jugador controla las acciones de un personaje (o de diversos miembros de un grupo) inmerso en algún detallado mundo. (Pokemon, World of Warcraft, Dark Soul)
- Aventura: caracterizados por la investigación, exploración, la solución de rompecabezas, la interacción con personajes del videojuego, y un enfoque en el relato en vez de desafíos basados en reflejos (Indiana Jones, Machinarium)
- Rompecabezas o de Lógica: se caracteriza por exigir agilidad mental al jugador para resolver enigmas por lo regular en una sola pantalla. Pueden o no tener una historia detrás. (Tetris, Buscaminas)
- Simulación: intentan recrear situaciones de la vida real. (The Sims, PC Building Simulator)
- Deportes: simula el campo de deportes tradicionales (FIFA Football, Madden NFL, Arc Rivals)
- Carreras: en el que se imitan competencias entre vehículos. Usualmente el objetivo es recorrer cierta distancia o ir de un sitio hacia otro en el menor tiempo posible (Need for Speed, Blur)

Por otro lado, hoy en día son habituales los videojuegos que toman elementos de más de un género, lo que ha dado lugar a géneros mixtos (por ejemplo, rol-acción, aventura-acción, entre otros). Los videojuegos de acción son los que tienen más beneficios cognitivos. Por ende se decidió escoger este género, ya que jugar a este tipo de videojuegos mejora diferentes tipos de atención (incluidas la sostenida, la selectiva y la dividida) y aumenta la eficiencia en el control de esta capacidad. También aumenta la velocidad con la que las personas procesan la información, permitiendo una mayor presteza de actuación en situaciones de presión.

### 1.1.2 Género acción

Entre los diversos géneros de videojuegos, el género de acción es el más amplio, englobando muchos subgéneros como videojuegos de disparos, de lucha, de sigilo o espionaje, *battle royale* (Batalla Real en inglés) y de supervivencia. Este tipo de juego consiste en luchar y pelear. Se basa en realizar ejercicios de repetición, por ejemplo, oprimir un botón para que el personaje ejecute una acción. Su objetivo principalmente es eliminar enemigos y su contenido suele ser violento. Es un género que se caracteriza por poner a los jugadores en situaciones donde deben atravesar obstáculos, poner a prueba su capacidad de reacción y de reflejo, mientras hay constantes ataques y movimientos. Uno de los subgéneros relacionados con el género de acción más atractivo es el de supervivencia, implican una inmersión grande, concentración y por lo general se relacionan con exposiciones cortas de tiempo, hace que sea de los más jugados (E. Pedersen 2014). Por lo tanto y sumado a que la plataforma Cosmox no presenta ningún producto con estas características se determinó seleccionar este subgénero.

### 1.1.3 Videojuegos de acción de subgénero supervivencia

Los videojuegos de supervivencia son un subgénero de los videojuegos de acción ambientado en un ambiente hostil, intenso y de mundo abierto, donde los jugadores generalmente comienzan con equipos mínimos y se les exige que recolecten recursos, herramientas de artesanía, armas y refugio, y sobrevivan el mayor tiempo posible.

Los videojuegos de supervivencia se consideran una extensión de un tema de videojuego común donde el jugador-personaje está varado o separado de los demás, y debe trabajar solo para sobrevivir y completar un objetivo. Los juegos de supervivencia se centran en las partes de supervivencia de estos juegos, al tiempo que fomentan la exploración de un mundo abierto. El manejo de recursos y la creación de objetos, se encuentran comúnmente en juegos de supervivencia. Al comienzo de un juego de supervivencia típico, el jugador generalmente se coloca solo en el mundo del juego con pocos recursos. No es raro que los jugadores gasten la mayoría o la totalidad del juego sin encontrar un personaje amigable que no sea jugador; dado que los NPC suelen ser hostiles para el jugador, se hace hincapié en la evasión, en lugar de la confrontación. Sin embargo, en algunos juegos, el combate es inevitable y proporciona al jugador recursos valiosos (por ejemplo, comida, armas y armaduras).

Por otra parte, también se distingue a unos juegos de otros, incluso dentro de un mismo género, por la perspectiva visual que adoptan (la posición de la cámara). Así, hay juegos con perspectiva 2D (ya sea con proyección paralela, vista lateral), 2.5D (mediante proyección isométrica, oblicua, entre otras) y 3D (en perspectiva fija, en primera persona, o en tercera persona).

Otro de los términos que es necesario dominar para el desarrollo del trabajo, es el de mecánicas de videojuegos, pues ellas garantizan internamente el funcionamiento del videojuego. Las mecánicas son acciones, comportamientos y control de mecanismos de los que dispone el jugador dentro del contexto del juego.

## 1.2 Mecánicas de videojuegos

Las mecánicas son cualquier acción realizada por el jugador que modifique el estado del videojuego, es decir, la posición y características concretas de todos los objetos y entornos de un juego en un momento preciso en el tiempo. El estado del videojuego, es modificado debido a una de las características definitorias de los juegos: la interactividad (Coterón, 2018). Las mecánicas también se encuentran clasificadas por categorías según la naturaleza de los mecanismos que las componen. A

continuación, pueden observarse cinco de estas categorías propuestas por Adams y Dormans, para agrupar mecánicas según su relación y función en el videojuego (Adams y Dormans 2018):

**Física:** La física en los videojuegos se suele definir como la ciencia del movimiento y fuerza, en el mundo del juego (que pueden desafiar las leyes de la física del mundo real). Las mecánicas de física están presentes al mover personajes de un lugar a otro, saltar, nadar, la dirección en la que este se mueve, así como la detección de colisiones con otros objetos.

**Economía interna:** Son las mecánicas de las transacciones que involucran los elementos del videojuego que se recogen, se consumen o se negocian. Abarca elementos fácilmente identificables como: recursos, dinero, energía y municiones. Sin embargo, la economía de un videojuego no se limita a elementos tangibles y concretos; también puede incluir abstracciones tales como la salud y el hambre, elementos presentes en la solución.

**Progresión:** En muchos videojuegos, tradicionalmente, el jugador tiene que llegar a un lugar en particular para rescatar a alguien o para derrotar al principal malhechor y completar el nivel. En este tipo de videojuego, el progreso del jugador está marcado por la capacidad en el manejo de los recursos, la creación de objetos y la exploración del mundo.

**Interacción Social:** estas mecánicas permiten la interacción social en el mundo del videojuego entre varios jugadores con fines e intereses comunes. Muchos videojuegos en línea incluyen la mecánica que recompensa dar regalos, invitar a nuevos amigos a unirse y participar en otras interacciones sociales.

**Maniobra táctica:** los videojuegos pueden tener mecánicas relacionadas con la colocación de unidades en el mapa con ventajas ofensivas o defensivas. La maniobra táctica es fundamental en la mayoría de los videojuegos de estrategia, pero también se encuentran en algunos de roles y simulación. Las mecánicas que rigen maniobras tácticas, típicamente especifican qué ventajas estratégicas cada tipo de unidad puede ganar al estar en cada posible ubicación. Estas aparecen en muchos juegos de mesa como el ajedrez, las damas chinas y en videojuegos de estrategia.

En el siguiente apartado poder determinar el comportamiento de estas mecánicas en algunos videojuegos se analizarán videojuegos reconocidos por su calidad en la comunidad de jugadores. El objetivo es identificar los comportamientos principales que presentan y agruparlos en mecánicas con nombres bien definidos, teniendo en cuenta los elementos conceptuales de las categorías mencionadas

en esta sección. Las mecánicas identificadas durante este análisis se asumirán a lo largo del trabajo como las mecánicas elementales de los videojuegos supervivencia que no deben faltar en la solución.

### **1.3 Análisis de los videojuegos homólogos Rust y Minecraft**

En el siguiente apartado se realiza el análisis de los videojuegos: Rust y Minecraft, dos de los que poseen mayor popularidad en el mercado de los videojuegos. El objetivo es identificar los comportamientos principales que presentan y agruparlos en mecánicas con nombres bien definidos, teniendo en cuenta los elementos conceptuales de las categorías mencionadas en esta sección. Las mecánicas identificadas durante este análisis se asumirán a lo largo del trabajo como las mecánicas elementales de los videojuegos de supervivencia que no deben faltar en la solución (Cultura Geek, sf). Rust es un videojuego de supervivencia en primera persona que fue lanzado en febrero de 2018 por Facepunch Studios. El juego comienza con la interfaz de creación del avatar o personaje, desprovisto de armas, ropa en condiciones y otros elementos básicos para la vida humana. El único objetivo es sobrevivir, y para ello se tendrá que recurrir a construir objetos, utilizar recursos naturales y alimentar al personaje, todo ello en un mundo abierto con una ambientación postapocalíptica que, a su vez, se desglosa en una serie de variados ecosistemas, teniendo que enfrentarse a animales salvajes, otros jugadores, forjar alianzas, construir diferentes estructuras, además de contar con un HUD en la esquina inferior derecha que indica la salud, la sed, y el hambre del personaje en cada momento, los cuales irán variando con el paso del tiempo.

El punto inicial de reaparición de Rust siempre será la playa, pero puede ser en un entorno nevado, boscoso o incluso desértico, y en cada uno de ellos encontramos elementos distintivos que dan mayor variedad al juego. Cada clima afecta directamente al personaje, que comienza la aventura desprovista de toda prenda lo que introduce el factor de la temperatura corporal. En los entornos nevados y mientras nada en el mar se notificará de que hace demasiado frío, lo que irá disminuyendo la salud poco a poco hasta que vuelva a entrar en calor.

Para recuperar salud se debe crear una hoguera y colocar cerca al personaje. Para conseguir alimentos hay dos opciones: recolectar alimentos vegetales como setas, champiñones o calabazas (las cuales se pueden plantar mediante semillas) o cazar animales salvajes, tales como gallinas, jabalíes, ciervos e incluso osos.

Los enemigos pueden asaltar la casa de los jugadores para robar objetos, y para evitar esto hay dos opciones: una, mejorar el material con el que se ha construido la base usando el martillo de madera, y dos, colocar una puerta con una cerradura.

No todo se puede *craftear*<sup>1</sup>, sino que también habrá diferentes objetos que se podrán recoger como parte de un botín al eliminar un enemigo. Entre ellos se encuentran diferentes tipos específicos de armas, vehículos o caballos para mejorar la movilidad.

La dinámica de Rust en cuanto al entorno, radica en que no solo nos tendremos que preocupar de las otras personas que puedan jugar con nosotros, sino también en los diferentes NPC<sup>2</sup> que habrá a lo largo del mapa. También tendremos que considerar aspectos como cambios atmosféricos, que pueden provocar desde hipotermia hasta radiación.

Una de las actividades más interesantes en Rust, es la de poder hacer grupos, las cuales consisten en atacar, solo o con tu clan, la base de otro equipo, destruirla y robarse los recursos que posean.



**Foto 1.** Videojuego "Rust".

Por otra parte, Minecraft fue creado por Mojang en 2009 y es una obra de Markus "Notch" Persson, es un videojuego de mundo abierto donde la exploración y las construcciones son fundamentales. Está fuertemente inspirado en juegos como Dwarf Fortress, Dungeon Keeper e Infiniminer. Minecraft cuenta con cuatro modos: supervivencia, creativo, aventura y espectador, que permiten a los jugadores satisfacer la experiencia de juego a su gusto (Minecraft, 2022).

El modo supervivencia es el principal, en este hay que conseguir recursos y sobrevivir al ataque de múltiples criaturas que surgen en la oscuridad, hasta que el jugador logre vencer al dragón. El personaje cuenta con diez corazones de vida para conseguir su objetivo. En él deberás sobrevivir a los 'enemigos' que aparecen por la noche (en Minecraft existe el ciclo del día natural, pero proporcionado para que las 24 horas sean realmente unos pocos minutos). Existen varios NPC agresivos, los *creepers* son los más

<sup>1</sup> Craftear: fabricar objetos a partir de otros ya existentes o de elementos básicos recolectables en un juego.

<sup>2</sup> NPC: son las siglas en inglés de 'Non Playable Character' (personaje no jugable), qué es el nombre que se les da a aquellos personajes que forman parte de la historia de un videojuego, pero que no pueden ser utilizados por el jugador; por lo general, están ahí solamente para dar indicaciones o como complementos.

conocidos, aunque la lista es bastante amplia y los hay de varios niveles y con diferentes ataques y habilidades.

Existe la opción de 'dormir' por la noche, evitando así el contacto con los NPC que actúan sólo en la oscuridad nocturna. Para empezar a jugar hay pequeñas guías que ayudan a los jugadores novatos a avanzar durante los primeros momentos del juego. Por ejemplo, esta guía habla sobre los primeros pasos que deberás dar en Minecraft, y que incluyen desde recolectar madera y alimentos, hacer las primeras armas, picar piedra o construir un refugio con una cama para descansar. Deberás hacerlo rápido, pues la noche llegará y tendrás que estar preparado.

A medida que vaya pasando el tiempo el juego irá avanzando y mejorando la supervivencia. Cada vez será más difícil encontrar ciertos materiales, se deberá ir creando tus propias minas para los menos habituales, como Redstone o diamante. Minecraft, como su nombre indica, es una combinación entre ir creando minas y mineando en ellas para conseguir los elementos necesarios para mejorar el equipo y las posibilidades de sobrevivir.

Con los *mods*<sup>3</sup> se añaden nuevos objetos, modificaciones en los monstruos, mejoras de rendimiento del juego en PCs con poca potencia, los que transforman el mundo, los que facilitan la creación de objetos o los que añaden mapas mejorados.



**Foto 2.** Videojuego "Minecraft".

En la siguiente tabla se describen las mecánicas recurrentes de estos videojuegos, tomando como guía las definidas en el epígrafe anterior.

---

<sup>3</sup> Mods: (un acortamiento del inglés modification, «modificación») En el mundo de los videojuegos, es una extensión del software que modifica un videojuego original proporcionando nuevas posibilidades, ambientaciones, personajes, diálogos, objetos y mapas.

Tabla 1. Descripción de las mecánicas de los videojuegos Rust y Minecraft

	Videojuegos	
Mecánicas	Rust	Minecraft
Cámara	El enfoque de la cámara es en primera persona.	El enfoque de la cámara es en primera persona, aunque se puede cambiar a tercera persona.
Mecanismo interfaz principal	Sí	Sí
Mecanismos de barra de estados	Sí . Muestra tres barras, una de salud o vida, una de sed y otra de hambre	Sí. Muestra solo la barra de vida y la de hambre.
Locomoción	Sí	Sí
Inventario	Sí	Sí
Craftear	Sí, se debe tener la receta para la fabricación	Sí
Combate	Sí	Sí

A partir de la descripción de las mecánicas de los videojuegos estudiados como homólogos en la tabla anterior, se puede arribar a la conclusión de que ambos presentan un enfoque de cámara en primera persona, aunque en Minecraft también se puede cambiar a una perspectiva de tercera persona mucho más cómoda. Tienen una interfaz sencilla y práctica, muestran a través de barras de estados elementos como la salud, la sed y el hambre, un inventario donde se guardan los diferentes objetos y presentan un paquete de mecánicas que permiten la locomoción del jugador, el combate y la fabricación de objetos, aspectos fundamentales de este género que sirven como guía para la propuesta de solución.

### 1.3.1 Mecánicas elementales de videojuegos del género supervivencia

Las mecánicas pueden ser definidas muy generales y englobar múltiples mecanismos o pueden ser subdivididas en otras de menor amplitud y más específicas, tan pequeñas como se decida en el proceso de desarrollo del videojuego. Actualmente son muchos los videojuegos de este subgénero que salen al mercado debido a su gran demanda, pero son pocos los que logran tener aceptación y generar un impacto en la comunidad de jugadores. Claros ejemplos de éxito son aquellos reconocidos entre la comunidad de jugadores, ya que logran aplicar correctamente y de manera atractiva mecánicas que

son propias de este tipo de videojuego, como son la exploración del mundo abierto, el manejo de recursos y la creación de objetos.

Las mecánicas de juego describen lo que un jugador puede hacer, cómo lo hace y las reglas que gobiernan esas acciones. Entre las principales mecánicas a tener en cuenta en este tipo de videojuegos:

- Mecanismo control de interfaz (Permite al jugador administrar opciones del juego como el sonido y los gráficos).
- Mecanismo control de vida (El jugador y algunas otras entidades tienen una cantidad específica de puntos de salud, cuando llegan a 0 mueren o son destruidos).
- Mecanismo control de hambre (El jugador posee una cantidad de puntos de “hambre” que disminuirán con el tiempo, si llegan a 0 esto afectará la barra de vida, se reponen sus puntos de hambre comiendo varios alimentos.).
- Mecanismo locomoción (Caminar, saltar, agacharse, correr, nadar).
- Mecanismo inventario (Componente que el jugador usa para gestionar los recursos que puede consumir o utilizar).
- Mecanismo recolección (El jugador puede recoger diferentes objetos).
- Mecanismo craftear (El jugador puede crear una variedad de objetos).
- Mecanismo combate (Las entidades pueden recibir daño y potencialmente morir o ser destruidas si reciben demasiado).

### 1.4 Metodología, herramientas y tecnologías

#### **Marco de Trabajo Ingenieril para el desarrollo de videojuegos.**

Basado en las herramientas que utiliza el Centro de Tecnologías Interactivas (VERTEX) para el desarrollo de videojuegos, se determinó como herramientas para Ingeniería de Software Asistida por Computadora (CASE, por sus siglas en inglés), Visual Paradigm, para guiar este proceso se utilizó el Marco de trabajo ingenieril para el proceso de desarrollo de videojuegos, como entorno de desarrollo se utilizó el motor de videojuegos Unity 3D y como lenguaje de programación C#.

El desarrollo de videojuegos se manifiesta de forma diferente a un proceso de desarrollo de software común. Un proceso de desarrollo de software se debe regir por guías, modelos, metodologías y/o marcos de trabajo que garanticen la implementación de un producto de alta calidad. Las actividades

ingenieriles que se deben ejecutar durante este proceso generan resultados diferentes a un proyecto de software de cualquier otro dominio de aplicación. Para el desarrollo de este videojuego se utilizó el Marco de trabajo propuesto por el Ms.C. Andy Hernández Paez, el mismo está compuesto por 5 etapas que complementan el proceso de desarrollo de los productos. Dichas etapas son: Conceptualización, Diseño, Implementación, Prueba y Mantenimiento. La estructura del marco se define a continuación (Hernández 2017):

### Etapa 1. Conceptualización:

- Definir el género sobre el cual se desarrollará el videojuego.
- Describir la mecánica del videojuego.
- Especificar las metas para la experiencia del jugador.

### Etapa 2. Diseño:

- Describir los elementos formales que definen la estructura del videojuego.
- Describir los elementos dramáticos que definen el entretenimiento del videojuego.
- Diseñar las pantallas gráficas elementales que forman la estructura del videojuego.
- Describir los elementos dinámicos que definen las mecánicas o mecanismos del videojuego.
- Validar los mecanismos teniendo en cuenta criterios técnicos para su implementación.
- Modelar el diagrama de paquetes de mecanismos teniendo en cuenta la distribución arquitectónica.
- Describir la concepción de los mecanismos sobre la distribución arquitectónica diseñada.
- Modelar el comportamiento de los mecanismos mediante diagramas de transición de estado.
- Mantener una trazabilidad bidireccional entre cada elemento del videojuego.
- Describir las características no funcionales del videojuego.

### Etapa 3. Implementación:

- Diseñar los componentes que encapsulan la implementación.
- Desarrollar las mecánicas especificadas y diseñadas.

### Etapa 4. Prueba:

- Desarrollar pruebas Alpha.
- Desarrollar pruebas Beta.

- Registrar defectos durante las pruebas realizadas.

Etapa 5. Mantenimiento:

- Realizar análisis Postmortem.
- Retomar la etapa de “Diseño”.

### **Visual Paradigm v. 15.1**

Es una herramienta que ha sido concebida para soportar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Fue diseñado para una amplia gama de usuarios interesados en la construcción de sistemas de software de forma fiable a través de la utilización de un enfoque orientado a objetos. Dicha herramienta mediante la utilización de Lenguaje de Modelado Unificado (UML, por sus siglas en inglés) colabora con el desarrollo de la aplicación, desde la planificación, el análisis, el diseño y la generación de artefactos necesarios en el desarrollo del videojuego. Entre las principales ventajas del uso de Visual Paradigm se encuentran que presenta una licencia gratuita y comercial, es fácil de instalar, actualizar y es compatible entre ediciones, soporta todos los diagramas de UML y su disponibilidad para múltiples plataformas (Visual Paradigm,2022).

### **Lenguaje de modelado**

Según Pressman el UML es “un lenguaje estándar para escribir diseños de software”, puede usarse para visualizar, especificar, construir y documentar los artefactos de un sistema de software intensivo. Este lenguaje está pensado para utilizarse en todos los métodos de desarrollo, etapas del ciclo de vida de un software, dominios de aplicación y medios. Es un sistema notacional destinado a los sistemas de modelado que utilizan conceptos orientados a objetos. Ofrece una variedad de diagramas para diferentes propósitos de modelado. Es el estándar mundial que utilizan los desarrolladores, autores y proveedores de Herramientas para Ingeniería de Software Asistida por Computación.

Los principales beneficios de UML son:

- Mejores tiempos totales de desarrollo.
- Modelar sistemas utilizando conceptos orientados a objetos.
- Establecer conceptos y artefactos ejecutables.
- Encaminar el desarrollo del escalamiento en sistemas complejos de misión crítica.

- Crear un lenguaje de modelado utilizado tanto por humanos como por máquinas.
- Mejor soporte a la planeación y al control de proyectos.
- Alta reutilización y minimización de costos.

## Unity

Un motor de videojuego es un sistema diseñado para la creación de videojuegos que reúne un conjunto de aplicaciones necesarias para su desarrollo (López 2016). Para el desarrollo de la presente investigación se utilizará como motor de videojuego Unity en su versión 2020.2.7f1. Esta herramienta ofrece una amplia gama de características y su interfaz es bastante sencilla, es multiplataforma y se utiliza en VERTEX para el proceso de desarrollo de videojuegos. En Unity, las mecánicas del juego son implementadas mediante scripts y compiladas usando el lenguaje de Java Scripts o C#, por lo que es de fácil asimilación. Este motor está enfocado en los bloques de construcción (*assets*<sup>4</sup>) que permiten desarrollar *plugins*<sup>5</sup> y módulos integrables a la interfaz de usuario de manera flexible, lo cual potencia la reutilización, obteniéndose como consecuencia la aceleración y optimización del tiempo de desarrollo y minimizar el esfuerzo necesario para su realización ya que las soluciones creadas son fácilmente adaptables a futuros cambios, los cuales pueden ser texturas, modelos 3D, archivos de audio, prefabricados, materiales, animaciones, *scripts*<sup>6</sup> y cualquier contenido utilizado en la creación del videojuego.(López 2016)

## Visual Studio Code

Visual Studio Code es un editor de código fuente desarrollado por Microsoft. Es un software gratuito, libre, de código abierto y multiplataforma, está disponible para Windows, GNU/Linux y macOS. VS Code tiene una buena integración con Git, cuenta con soporte para depuración de código, y dispone de un sinnúmero de extensiones, que básicamente te da la posibilidad de escribir y ejecutar código en cualquier lenguaje de programación (Frankier Flores, 2022).

## Lenguaje de programación C#

---

4 Assets: el término hace referencia a los recursos que utiliza un videojuego y que forman parte de él en el momento de su creación.

5 Plugins: son complementos que añaden funcionalidades extra o mejoras a los programas. Es decir, son miniprogramas que suman alguna característica que no venía por defecto en el programa original.

6 Scripts: secuencia de comandos(traduciendo desde inglés) es un término informal que se usa para designar a un programa relativamente simple.

C# es uno de los lenguajes de programación que emplea Unity para compilar los *scripts*, es orientado a objetos, lo cual facilita el trabajo ya que en esta herramienta todo componente o elemento del juego es un objeto o clase. Al empezar a programar, se pueden definir una o más clases dentro de un mismo espacio de nombres. La compatibilidad del lenguaje con las operaciones asíncronas proporciona la sintaxis para crear sistemas distribuidos. El control de excepciones proporciona un enfoque estructurado y extensible para la detección y recuperación de errores. C# resalta el control de versiones para garantizar que los programas y las bibliotecas puedan evolucionar con el tiempo de manera compatible. Presenta un rango más amplio y definido de tipos de datos que otros lenguajes. Soporta todas las características propias del paradigma de programación orientada a objetos: encapsulación, herencia y polimorfismo (Ivar 2018).

## JSON

JavaScript Object Notation (JSON) es un formato ligero empleado para el intercambio de datos, mejora la representación estructural de los datos; requiere una baja codificación y procesamiento. Soporta dos tipos de estructuras: los objetos que contienen una colección de pares llave-valor y los arreglos de valores. JSON no tiene espacios de nombres, cada objeto es un conjunto de claves independientes de cualquier otro objeto. Puede representar cualquier estructura de datos permitiendo añadir nuevos campos con total facilidad, característica que lo hace ser flexible (Ivan de Souza, 2021).

### 1.5 Conclusiones parciales

En el desarrollo del capítulo se obtuvo una mejor comprensión acerca del problema planteado dando cumplimiento a las primeras tareas trazadas en la investigación a partir del análisis del concepto de videojuegos. El estudio de los diferentes géneros, específicamente el de supervivencia y el análisis de los homólogos escogidos por su gran aceptación en la comunidad, permitió determinar los elementos fundamentales que no deben faltar en la solución. Además, permitió identificar el correcto comportamiento de las mecánicas definidas dentro de la solución, de tal forma que se logre un producto con mayor aceptación. Basado en las herramientas que utiliza el Centro de Tecnologías Interactivas (VERTEX) se hizo la selección de la metodología, marco de trabajo para el desarrollo de videojuegos, las herramientas y las tecnologías: Visual Paradigm por estar concebida para soportar el ciclo de vida completo del proceso de desarrollo del software, Unity por ser un motor gráfico que ofrece una amplia gama de características y su interfaz es bastante sencilla y C# como lenguaje de programación fundamental.

## **CAPÍTULO 2. CONCEPTUALIZACIÓN Y DISEÑO DE UN VIDEOJUEGO DE SUPERVIVENCIA**

### **INTRODUCCIÓN**

En el siguiente capítulo se describen las principales características del videojuego, que será la propuesta de solución. Se especifican los principales aspectos que define el Marco de trabajo ingenieril para el proceso de desarrollo de videojuegos, como son el diseño del videojuego y la especificación de los mecanismos del mismo, se describen los procesos ingenieriles a realizar para la construcción del videojuego, los cuales permiten identificar las necesidades del producto.

### **2.1 Propuesta de solución**

Como propuesta de solución al problema de investigación, se desarrollará un demo de un videojuego llamado “La Isla”, el cual se vislumbra como un videojuego del género supervivencia, con gráficos en tres dimensiones y la cámara en tercera persona y disponible en el sistema operativo Windows. que servirá para aumentar el catálogo de productos de la plataforma Cosmox. Será un videojuego con un mapa amplio de mundo abierto elaborado a partir de un terreno de Unity, se utilizarán *assets* gratuitos, las interfaces a utilizar serán sencillas y de fácil comprensión. Las acciones a llevar a cabo por el jugador se llevarán a cabo mediante el uso del teclado y el mouse, entre ellas estarán las de recolectar materiales y forjar herramientas, moverse por el terreno, mover la cámara, combatir los NPC hostiles y cumplir con las misiones que brindan los NPC del campamento español para poder ganar la partida.

### **2.2 Resumen del videojuego**

El videojuego está basado en la historia de un niño intranquilo y curioso, Daniel, que es transportado a la Cuba del siglo XVI. Se encontrará en un escenario amplio en el que podrá desplazarse libremente, rodeado de vegetación y animales salvajes. Tendrá que recolectar diferentes objetos con el fin de sobrevivir el mayor tiempo posible y crear herramientas que le “faciliten” su estadía en este duro entorno. Deberá cumplir las misiones para cumplir su objetivo que es volver a su tiempo no sin antes llevarse una panorámica de aquel tiempo, su fauna, su flora y algunos elementos de aquella época. El videojuego va dirigido al público cubano y tiene como objetivo, proveer de contenidos nacionales para el entretenimiento a la plataforma para servicios de videojuegos Cosmox.

### **2.3 Diseño del videojuego**

El diseño del videojuego establece la visión y el enfoque que guiará al proyecto hasta el final del proceso. Se tendrá en cuenta los elementos que lo conforman (Rondon,2012):

- Formales: definen la estructura del juego.
- Dramáticos: definen el entretenimiento y el nivel de inmersión de los jugadores en el juego.

### 2.3.1 Elementos formales del videojuego

Estos elementos definen el videojuego y la estructura con que contará, para ello se plantean los siguientes elementos:

- Jugador: se caracterizan los jugadores según los aspectos definidos (Invitación a jugar, roles, patrón de interacción entre jugadores, relación con las metas trazadas.)
- Objetivos: definen qué es lo que el jugador trata de lograr bajo un conjunto de reglas y tienen la característica de ser retadores, pero siempre se pueden realizar.
- Procedimientos: definen métodos y acciones dentro del juego que el jugador puede ejecutar para lograr los objetivos.
- Reglas: definen objetos, limitan el comportamiento dentro del juego y determinan efectos.
- Recursos: definen elementos usados por el jugador con fin de cumplir los objetivos propuestos.
- Conflictos: se generan a través de reglas, procedimientos y situaciones que no permiten cumplir los objetivos directamente.
- Frontera o Límite: definen todo lo que separa el juego de lo que no es el juego y está estrechamente relacionado con las metas del mismo.
- Resultado: define la condición de ganado o perdido (en caso de existir).

#### Jugador

**Tabla 2. Descripción del juego**

Aspecto	Descripción
Invitación a jugar	Botón: Iniciar Aventura
Cantidad	Un solo jugador
Roles	Interactuar con el medio
Patrón de interacción	Jugador vs Mundo

#### Objetivo

Tabla 3. Objetivos

Aspecto	Descripción
Encontrar el cráneo que lo transporta a su época	El jugador debe procurar completar todas las misiones para encontrar el objeto
Tipo	Explícito
Categoría	Solución

### Reglas

- El videojuego tiene un mapa abierto que puede ser explorado en cualquier momento
- El jugador para crear nuevos objetos debe reunir los recursos necesarios
- Cada objeto se puede crear en el yunque a partir de los materiales necesarios
- El hambre aumenta con el paso del tiempo y si llega al máximo empieza afectar la vida del jugador haciendo que disminuya.
- Consumir alimento disminuye el hambre en 25
- Se puede atacar con armas cuerpo a cuerpo y a distancia ambas con un daño variable aleatorio entre 20 y 100
- Objetos del mismo tipo se guardan en la misma casilla del inventario, pero no con las herramientas
- Solo se puede tener equipada una herramienta o un arma
- El alimento y las pociones se agotan una vez consumidas
- La vida del jugador disminuye con el daño provocado por los NPC
- La vida del jugador disminuye gradualmente cuando el hambre llega a 0
- El jugador deberá cumplir las misiones para avanzar
- Los árboles se pueden talar para obtener madera
- Los arbustos se pueden recoger para obtener plantas
- Las rocas se pueden minar para obtener metal
- El jugador cuenta con un inventario de 20 casillas
- El jugador pierde una vez que su barra de vida llega a 0.
- Los animales son derrotados cuando su vida llegue a 0

### Recursos

- Existen objetos que pueden ser recogidos por el jugador: recursos, armas y herramientas.

- Las armas son la espada y el arco
- Las herramientas son el hacha, el pico y el martillo
- Los recursos son metal, madera y plantas
- Barra de vida: muestra la vida actual del jugador
- Barra de hambre: muestra el hambre actual del jugador
- Los españoles del campamento asignan misiones
- Una imagen que muestra la apariencia del jugador

### **Procedimientos**

- En la escena Menú Principal el jugador puede comenzar una partida, abandonar el videojuego y desactivar o activar el sonido
- Con la partida ya iniciada, el jugador puede acceder a su inventario y gestionarlo, interactuar con los NPC y el entorno, crear nuevos objetos o atacar a los NPC hostiles
- Al cumplir la última misión se devuelve al jugador al Menú Principal

### **Conflicto**

- Sobrevivir hasta cumplir las misiones para ganar

### **Resultado**

- Cuando se cumplen todas las misiones se le notifica al jugador que ha ganado, Daniel es devuelto a su tiempo

### **2.3.2 Elementos dramáticos**

Los elementos dramáticos se encargan de definir el nivel de inmersión de los jugadores en el juego, para ello se plantean los siguientes elementos dramáticos (Omar Correa, 2017):

- Premisa: se define la introducción a la historia del juego.
- Historia: se define lo que se quiere contar del videojuego y dónde se va a desenvolver el juego, describiendo el contexto teórico de motivación del videojuego al usuario.
- Retos: es lo que produce una experiencia de éxito y felicidad para ser alcanzado por el usuario

### **Premisa**

Daniel es un niño intranquilo y muy hábil (destacado en las actividades de exploración y campismo), que se encuentra cursando el 9no grado en una escuela del capitalino municipio del Cerro, La Habana, Cuba. Un día junto a sus compañeros de aula es llevado por su profesora, como parte de una tarea, al Museo de Arqueología de la Habana, donde se asombra al ver todas aquellas piezas que son parte de

nuestra historia, al punto que ignora las advertencias de su profesora de no tocar absolutamente nada. Este se ve tentado de tal forma que coge una de ellas para verla más detenidamente, se trataba de un cráneo que formaba parte de unos restos aborígenes. Como si fuera por arte de magia, Daniel es transportado a un tiempo del que solo había escuchado en las clases de historia, se trata de la Comunidad Primitiva. Es allí donde Daniel, un chico muy habilidoso, tendrá que poner todo su empeño en lograr descubrir como volver a su tiempo, pero para ello primero deberá tratar de sobrevivir al hostil entorno en que se encuentra.

## Historia

Daniel aparece en algún lugar de la geografía de Cuba, en plena Comunidad Primitiva justo cuando la Isla empezaba a ser colonizada. El aparece solo, con muchas preguntas, pero para darle respuestas primero tiene que sobrevivir, para esto deberá explorar el terreno, interactuar con los NPC, recolectar recursos, cazar y crear herramientas que se lo permitan. Una vez que cumpla las misiones que le asignan los NPC deberá llegar a la aldea aborígen donde se encuentra el cráneo que lo transportará a su tiempo.

## Retos

- Tratar de sobrevivir el mayor tiempo posible.
- Cumplir las misiones.
- Sobrevivir a los enemigos.
- Llegar a la meta.

## 2.4 Diseño y descripción de las pantallas gráficas

A continuación, se definen las pantallas gráficas, así como los elementos por las que están compuestas y su comportamiento.

Imagen	Descripción
--------	-------------

	<p><b>Pantalla Principal:</b> Interfaz inicial del juego.</p> <p>En ella el usuario puede acceder a diferentes opciones del videojuego a través de los botones:</p> <p>Iniciar Aventura: El jugador inicia el videojuego</p> <p>Salir: El jugador desea salir del videojuego</p> <p>Música: el jugador puede deshabilitar la música del videojuego.</p>
	<p><b>Escenario de juego:</b> el jugador interactúa con el mundo del videojuego.</p>
	<p><b>Sección Menú:</b> pulsando la tecla Esc puede acceder a un menú en el cual puede abandonar la partida, habilitar/deshabilitar el sonido.</p>
	<p><b>Sección Jugador:</b> Muestra el avatar del jugador, el nombre, la barra de vida y la barra de hambre.</p>

	<p><b>Sección de opciones:</b> el jugador puede controlar los gráficos, la resolución, si desea jugar con la pantalla completa o <i>fullscreen</i> y activar o desactivar algunos efectos.</p>
	<p><b>Sección Creación:</b> el jugador puede seleccionar qué objeto quiere fabricar y al seleccionar muestra los recursos que necesita, así como la cantidad de objetos que desea fabricar.</p>
	<p><b>Sección Inventario:</b> se muestran los elementos presentes en el inventario del jugador y la cantidad de unidades de estos.</p>

## 2.5 Especificación de mecanismos

La especificación de mecanismos tiene como objetivo identificar los mecanismos que conforman al videojuego, así como sus propiedades y organización arquitectónica. Dichos mecanismos se encargan de agrupar por paquetes las funcionalidades del videojuego, permitiendo la interacción del jugador con el sistema e incrementando la jugabilidad. La siguiente sección esencialmente describe lo que el jugador puede hacer y cómo puede hacerlo (Vazquez 2018). Dichos mecanismos han sido en su

mayoría detectados y definidos mediante el análisis del género supervivencia y los videojuegos homólogos.

### 2.5.1 Especificación de mecanismos del videojuego

No.	Nombre	Descripción	Organización arquitectónica perteneciente
M1	Gestionar Interfaz	<p><b>Objetos:</b> botones (Iniciar Aventura, Música, Salir)</p> <p><b>Propiedades:</b></p> <p>Los botones se encuentran ubicados en el centro inferior de la pantalla</p> <p><b>Comportamiento:</b></p> <p>Permite la gestión de los componentes de la interfaz:</p> <p>presionar el botón Iniciar Aventura comienza el videojuego, clic en el botón Música permite deshabilitar/habilitar la música del videojuego y pulsar el botón Salir se cierra el videojuego</p> <p><b>Relaciones:</b> M2, M8</p>	Mecanismo alterno
M2	Cámara	<p><b>Objeto:</b> jugador, cámara</p> <p><b>Propiedades:</b> permite la visualización del escenario a través de la perspectiva del jugador</p> <p><b>Comportamiento:</b> se puede alejar o acercar la cámara a través de la rueda del ratón</p> <p><b>Relaciones:</b> M1, M5, M7, M10</p>	Mecanismo núcleo
M3	Gestión de vida	<p><b>Objeto:</b> barra de vida, jugador, objetos</p>	Mecanismo núcleo

		<p><b>Propiedades:</b> una barra roja indica el nivel de su vida máxima y actual, cambia a color blanco a medida que disminuye</p> <p><b>Comportamiento:</b> disminuye si el jugador recibe daño y/o cuando la barra de hambre está vacía y/o aumenta si consume algún objeto</p> <p><b>Relaciones:</b> M6, M10</p>	
M4	Gestión de hambre	<p><b>Objeto:</b> barra de hambre, jugador, objetos</p> <p><b>Propiedades:</b> una barra azul indica el nivel de su hambre máxima y actual, cambia a color blanco a medida que aumenta</p> <p><b>Comportamiento:</b> indica el nivel de su hambre máxima y actual. Puede disminuir el hambre consumiendo objetos y aumentará gradualmente con el tiempo</p> <p><b>Relaciones:</b> M6, M10</p>	Mecanismo núcleo
M5	Locomoción	<p><b>Objeto:</b> jugador</p> <p><b>Propiedades:</b> Permite el desplazamiento del jugador por el escenario</p> <p><b>Comportamiento:</b> a través de las teclas (W, A, S y D) el jugador puede caminar, nadar, correr (presionar la tecla Shift) y saltar (Barra espaciadora)</p> <p><b>Relaciones:</b> M2</p>	Mecanismo núcleo
M6	Administrar Inventario	<p><b>Objeto:</b> jugador, inventario, objetos</p> <p><b>Propiedades:</b> Permite al jugador recolectar y gestionar sus objetos y herramientas</p>	Mecanismo núcleo

		<b>Relaciones:</b> M3, M4, M9	
M7	Administrar Misión	<p><b>Objeto:</b> jugador</p> <p><b>Propiedades:</b> el jugador debe cumplir las misiones que se muestran en el panel para avanzar en el juego</p> <p><b>Relaciones:</b> M2, M5</p>	Mecanismo núcleo
M8	Abandonar Partida	<p><b>Objeto:</b> jugador</p> <p><b>Propiedades:</b> el jugador puede abandonar la partida actual en cualquier momento que lo desee</p> <p><b>Relaciones:</b> M1</p>	Mecanismo Alternativo
M9	Crear objetos	<p><b>Objeto:</b> Inventario</p> <p><b>Propiedades:</b> el jugador puede crear nuevos objetos o herramientas a partir de otros objetos en su inventario</p> <p><b>Comportamiento:</b> a través del sistema de <i>craft</i> el jugador puede crear nuevas herramientas u objetos</p> <p><b>Relaciones:</b> M6</p>	Mecanismo núcleo
M10	Combate	<p><b>Objeto:</b> jugador, NPC</p> <p><b>Propiedades:</b> el jugador puede combatir contra los NPC para obtener recompensas como alimentos u otros objetos</p> <p><b>Relaciones:</b>M2, M5, M6</p>	Mecanismo núcleo

## 2.6 Requisitos no funcionales

Los requisitos no funcionales (RnF) hacen referencia a las cualidades, restricciones y características de un software, pero que no determinan una funcionalidad del mismo. Dentro del videojuego se definieron los siguientes:

➤ **RnF 1: Usabilidad.**

- RnF 1.1: El videojuego está diseñado fundamentalmente para personas mayores de los 12 años.
- RnF 1.2: Para hacer uso del videojuego no se necesita conocimientos técnicos.
- **RnF 2: Apariencia o Interfaz externa.**
- RnF 2.1: El videojuego debe tener una interfaz agradable al usuario para que el mismo se sienta cómodo.
- **RnF 3: Software.**
- RnF 3.1: Debe estar disponible para los sistemas operativos Windows a partir de su versión 8.0
- RnF 3.2: El sistema debe tener compatibilidad con Unity en sus versiones de 2020.2.7f1 o superiores.
- **RnF 4: Hardware.**
- RnF 4.1: Es necesario para el funcionamiento del sistema mínimo, un microprocesador Intel(R) Celeron(R) CPU G1840 @ 2.80GHz, 2 GB de memoria RAM.
- **RnF 5: Restricciones del diseño y la implementación.**
- RnF 5.1: para el desarrollo del videojuego se utilizarán las herramientas definidas para el proceso de desarrollo de videojuego en VERTEX:
  - Motor de videojuego: Unity 2020.2.7f1.
  - Metodología de Software: Proceso Unificado Ágil (AUP), utilizando el Marco de trabajo ingenieril para el proceso de desarrollo de videojuegos utilizado por VERTEX.
  - Lenguaje de programación: C#.
  - Herramientas para CASE: Visual Paradigm.
  - Los datos del videojuego serán exportados en un fichero JSON.
  - Framework de desarrollo: Visual Studio Code.

## 2.7 Paquete de Mecanismos

Los diagramas de paquetes son diagramas estructurales que se emplean para mostrar la organización y disposición de diversos elementos de un modelo en forma de paquetes. Un paquete es una agrupación de elementos UML relacionados, como diagramas, documentos, clases o, incluso, otros paquetes. Cada elemento está anidado dentro de un paquete, que se representa como una carpeta de archivos dentro del diagrama, y que luego se organiza jerárquicamente dentro del diagrama. Los diagramas de paquetes se usan con frecuencia para proporcionar una organización visual de la

arquitectura en capas dentro de cualquier clasificador UML, por ejemplo, un sistema de software (Scott Ambler, 2017).

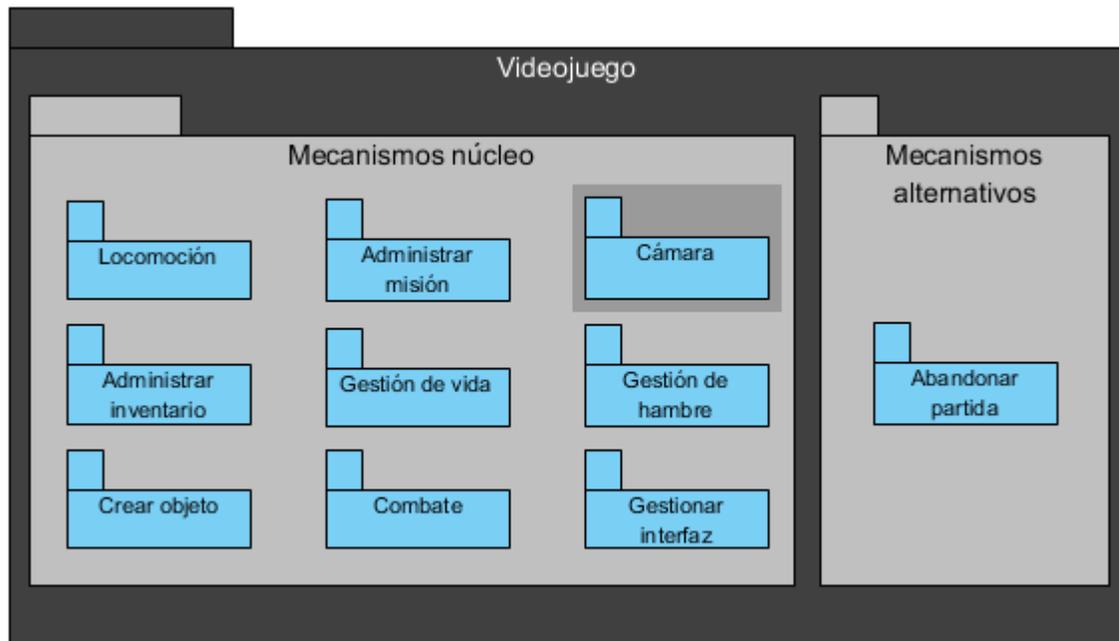


Imagen 1 Diagrama de paquete de los mecanismos.

### 2.7.1. Descripción de paquetes de mecanismos

**Mecanismos Núcleo:** son los mecanismos indispensables para el juego que constituyen el sentido de motivación del jugador y el propósito esencial del juego como es el caso de los mecanismos gestionar interfaz, cámara, locomoción, administrar inventario, administrar misión, gestión de vida, gestión de hambre, combate y crear objeto. Además, los mecanismos núcleos permiten la interacción del jugador con cada uno de los elementos que contienen las escenas, así como la navegación por ellas, con el propósito de cumplir con cada uno de los objetivos del juego.

**Mecanismos Alternativos:** amplían el espectro de funcionalidades del videojuego a través de los mecanismos núcleo. Estos mecanismos sin realizar ninguna modificación de los mecanismos núcleo son usados con un propósito completamente distinto al que fue concebido, en este caso el mecanismo abandonar partida.

## 2.8 Arquitectura de software

Para la realización del sistema se creó una arquitectura basada en capas, esta se enfoca en la distribución de roles y responsabilidades de forma jerárquica, una forma muy efectiva de separación de responsabilidades y estructurar los diferentes elementos necesarios del videojuego. Las capas de la arquitectura se definen a continuación (Cruz, Alejandro. 2017):

**Capa de Interfaz:** esta capa está formada por los principales elementos del videojuego, como son: controlador principal del juego (Game Manager), los datos (Data Manager) y las escenas (Interfaz Manager).

- Game Manager: es el controlador principal del videojuego, se implementa como un GameObject que se encuentra en todas las escenas del juego.
- Interface Manager: maneja las escenas del juego y los cambios que ocurren entre ellas.
- Data Manager: maneja todos los datos del juego.

**Capa de Interacción:** en esta capa se encuentran los diferentes escenarios y los objetos que pertenecen a los mismos.

Escenario del menú principal: contiene una escena de un barco en movimiento y un panel que muestra los botones: Iniciar Aventura, Música y Salir

Escenario de carga: muestra una barra que indica que la escena del videojuego está cargando

Escenario del videojuego: es la escena principal del videojuego

**Capa de Caracteres:** en esta capa se encuentran los caracteres encargados del manejo del personaje del videojuego.

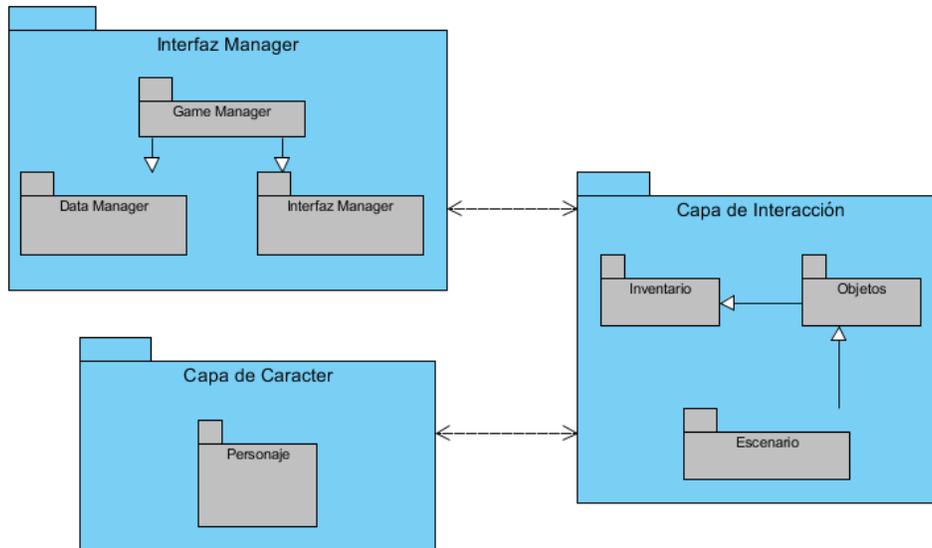


Imagen 2. Arquitectura del videojuego.

### 2.9 Diagrama de Clases

Para una mayor comprensión, de cómo estará estructurado el sistema de las clases que componen la solución, se muestra el siguiente diagrama de clases perteneciente al mecanismo inventario. El resto de los diagramas de clases se encuentran en el anexo A1.

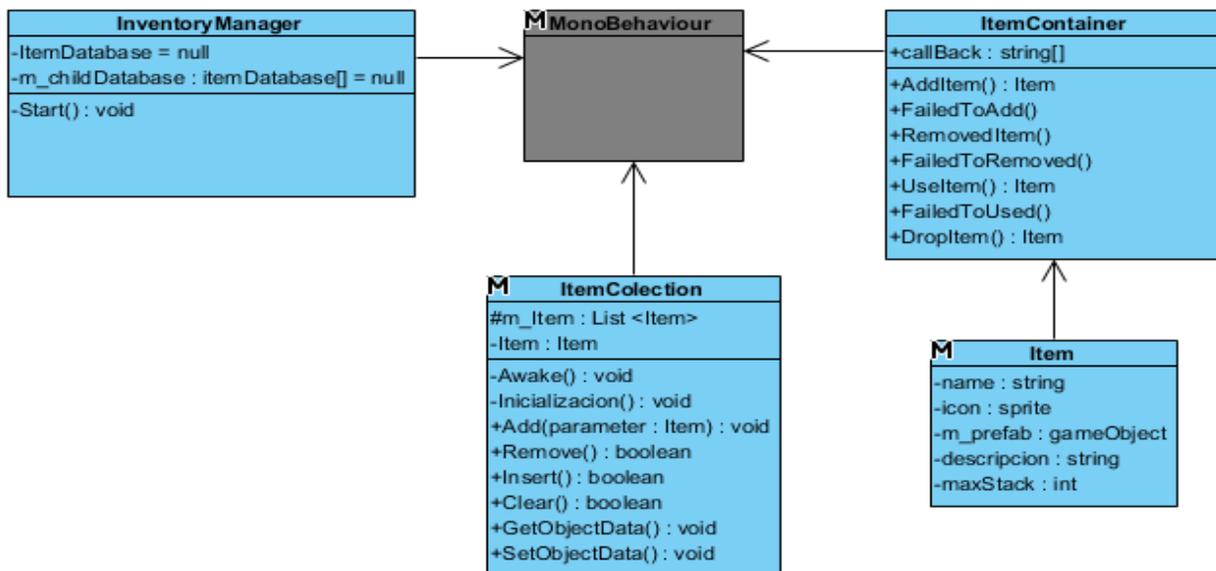


Imagen 3. Diagrama de clase del mecanismo inventario.

## 2.10 Patrones de Diseño

Un patrón es una descripción del problema y la esencia de su solución, codifica conocimiento específico acumulado por la experiencia en un dominio. Los desarrolladores lo usan como una forma de reutilizar la experiencia, clasificando las soluciones con términos de común denominación y van formando un amplio repertorio de principios generales y de expresiones que los guían al crear un software (Craig, 2018). En el diseño de la solución se utilizarán algunos de los Patrones Generales de Software de Asignación de Responsabilidad (GRASP) y algunos de los patrones Pandilla de cuatro (GOF). A continuación, se describen los patrones GRASP que se usarán en el desarrollo de la solución.

- **Alta cohesión:** es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Este patrón se observa en los *scripts* y es la medida en la que un componente o clase realiza únicamente la tarea para la cual fue diseñada.
- **Bajo acoplamiento:** el acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases. El uso de este patrón permitirá que el número de relaciones entre las clases de la solución sea el menor posible. De tal forma que, en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, lo cual potencia la reutilización y disminuye la dependencia entre estas.

También se utilizarán para el diseño de la solución el patrón GoF siguiente:

- **Fachada:** el patrón *Facade* eleva el nivel de abstracción de un determinado sistema para ocultar ciertos detalles de implementación y hacer más sencillo su uso. Provee una interfaz a una serie de clases para evitar el exceso de instancias de las mismas, dejando a las subclasses accesibles para ser usadas directamente. Se implementa en el *GameManager*, clase encargada de gestionar los aspectos principales del juego, permitiendo el acceso al resto de las clases asociadas a ella o convirtiendo al *GameManager* en la fachada para todos los elementos de la Capa Interfaz (SOMMERVILLE, 2011).
- **Creacionales:** los patrones creacionales tratan con las formas de crear instancias de objetos. Su objetivo es abstraer el proceso de instanciación y ocultar los detalles de cómo los objetos son creados o inicializados. Ejemplos de estos son los siguientes (Fábrica abstracta, Constructor, Método de fabricación, Prototipo, Instancia única).

## 2.11 Representación del Comportamiento

La representación del comportamiento es uno de los principios fundamentales de todos los métodos de análisis de requisitos. El modelo de comportamiento indica la forma en la que responderá el software a

eventos o estímulos externos. Los diagramas de estados se emplean para representar el comportamiento del videojuego según la acción del jugador. En la Imagen 4. se muestra el diagrama de estados del mecanismo gestionar interfaz, donde se indican las posibles acciones que puede realizar el jugador y los estados destino a los que puede llegar en este escenario. El resto de los diagramas de clases se encuentran en el anexo A4 (Hernández, 2017).

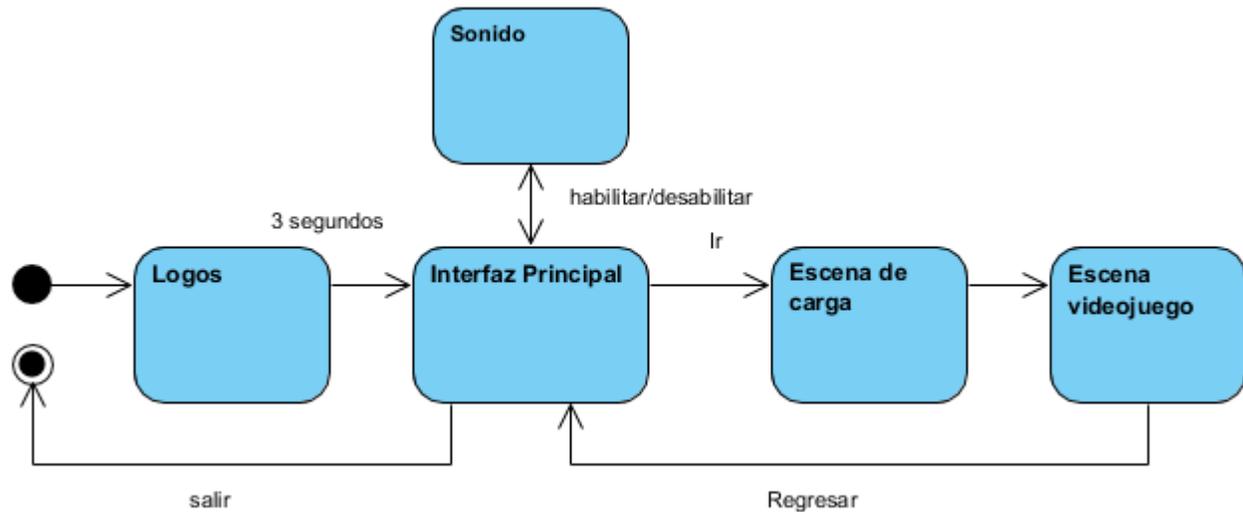


Imagen 4 Diagrama de estado de mecanismo gestionar interfaz

## 2.12 Conclusiones Parciales

Con la culminación del capítulo, se arribó a la conclusión de que las bases de la solución del videojuego, quedaron bien definidas gracias a los artefactos generados por la metodología empleada, la cual permitió comprender las funcionalidades y el contenido concreto para la implementación del comportamiento de cada elemento que intervienen en los diferentes escenarios, para ello se utilizó la arquitectura basada en capas. Con el diseño del videojuego se pautaron los elementos visuales y de jugabilidad que va a presentar la solución. La descripción de los mecanismos y el modelado del comportamiento permitirá un mayor entendimiento al desarrollador a la hora de implementar la jugabilidad en la fase de implementación.

## CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBAS DE UN VIDEOJUEGO DE SUPERVIVENCIA

En el presente capítulo se define el estándar de codificación y se representa la estructura del videojuego. Se valida la propuesta de solución a partir de las pruebas de aceptación.

### 3.1 Estándar de codificación

Los estándares de codificación son un conjunto de convenciones, establecidas para la escritura de código. Estos varían en dependencia del lenguaje de programación, lo que posibilitan una mejor lectura e interpretación del software. Su empleo permite que el código sea de fácil comprensión por parte de los programadores (Ampuero y Trujillo, 2017).

Para el desarrollo de la solución se especifican estándares para la escritura del código fuente. Por ejemplo: el nombre de las variables, métodos, clases y estilo

**Definición de clases:** las clases comienzan con mayúscula al inicio de la palabra y en caso de estar conformada por palabras compuestas, la definición debe ser continua y cada palabra debe iniciar con mayúscula siguiendo el estilo determinado. Ejemplo de ello se muestra a continuación:

```
public class FPC1 : MonoBehaviour
```

**Declaración de variables:** los nombres de las variables comienzan con minúscula y en caso de estar conformada por palabras compuestas, la segunda palabra empezaría con mayúsculas, a continuación, se muestra un ejemplo de cada caso.

```
private int allSlots;
private int slotsDisponibles;
```

**Métodos:** el nombre de los métodos debe comenzar con mayúscula y en caso de estar conformada por palabras compuestas la siguiente palabra también comenzará con mayúsculas.

```
private void OnTriggerEnter(Collider other)
{
    ...
}

public void AddItem(GameObject itemObject, int itemID, string itemType, string itemDescription, Sprite itemIcon)
```

**Estilo:** el estilo utilizado en la implementación de la propuesta de solución es propio para lenguajes de programación que usan llaves para indentar o delimitar bloques lógicos de código y es también un punto clave para hacer el código más legible. Está presente en los ciclos y estructuras de control.

```
private void OnTriggerEnter(Collider other)
{
    if(other.tag == "Item" )
    {
        GameObject itemPickedUp = other.gameObject;

        Item item = itemPickedUp.GetComponent<Item>();

        AddItem(itemPickedUp, item.ID, item.type, item.description , item.icon);
    }
}
```

### 3.2 Diagrama de componentes

El diagrama de componentes es un tipo de diagrama del UML. Representa la organización lógica de la implementación de un sistema, a través de los componentes y las relaciones de dependencia entre ellos (Pressman y Troya 2020). A continuación, se muestran dos de los diagramas de componentes de los mecanismos del videojuego gestionar interfaz y crear objeto.A.5

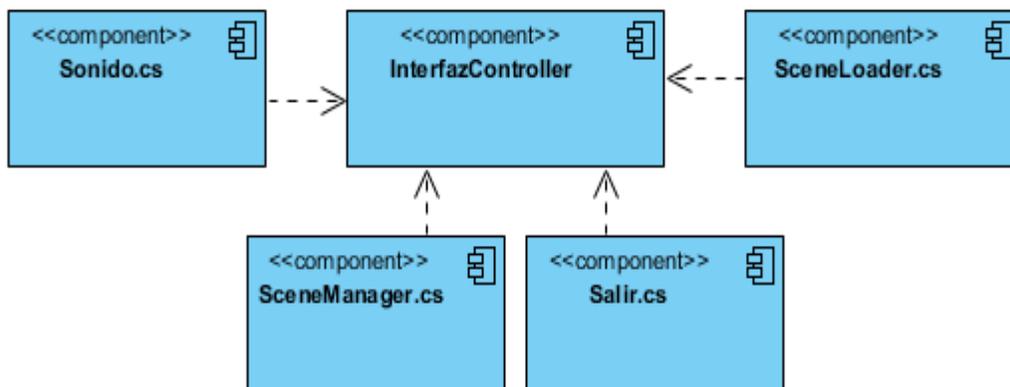


Imagen 8 Diagrama de componentes del mecanismo gestionar interfaz

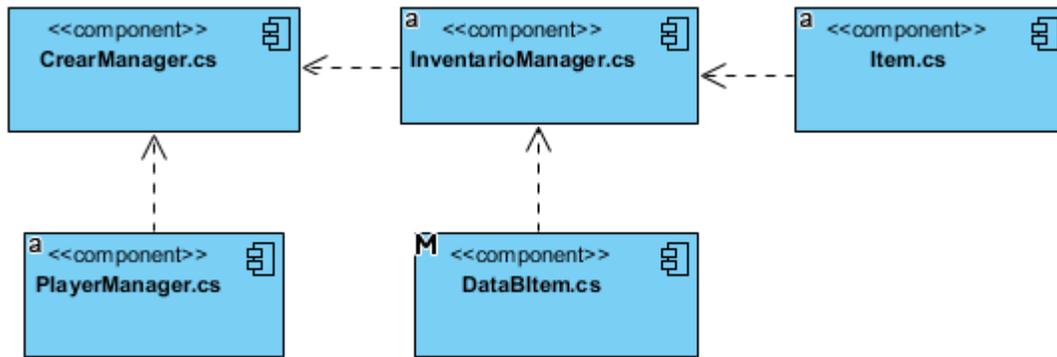


Imagen 9 Diagrama de componentes del mecanismo crear objeto

### 3.3 Fase de pruebas

Las pruebas de software son un elemento crítico para la garantía de la calidad del sistema. Representan una revisión final de las especificaciones del diseño y de la implementación. Su principal objetivo es diseñar pruebas que, sistemáticamente, saquen a la luz diferentes clases de errores, haciéndolo con la menor cantidad de tiempo y esfuerzo (Pressman y Troya 2020).

#### Pruebas Alfa y Beta

Pruebas alfa y beta son las metodologías de validación de clientes (tipos de pruebas de aceptación) que ayudan a generar confianza para lanzar el producto y, por lo tanto, resultan en el éxito del producto en el mercado. Aunque ambos se basan en usuarios reales y en diferentes comentarios del equipo, están impulsados por distintos procesos, estrategias y objetivos. Estos dos tipos de pruebas en conjunto aumentan el éxito y la vida útil de un producto en el mercado.

Las fases de prueba alfa y beta se centran principalmente en descubrir los errores de un producto ya probado y dan una imagen clara de cómo los usuarios en tiempo real utilizan realmente el producto. También ayudan a ganar experiencia con el producto antes de su lanzamiento y los valiosos comentarios se implementan de manera efectiva para aumentar la usabilidad del producto.

Cuando se construye un software se realizan una serie de pruebas a fin de permitir la validación de todos los requerimientos. Son realizadas por el usuario final en lugar de por los ingenieros de software, estas pruebas pueden variar desde una “prueba de conducción” informal hasta una serie de pruebas planificadas y ejecutadas sistemáticamente.

**Pruebas Alfa:** La prueba alfa se lleva a cabo en el sitio del desarrollador por un grupo representativo de usuarios finales. El software se usa en un escenario natural con el desarrollador “mirando sobre el hombro” de los usuarios y registrando los errores y problemas de uso. Las pruebas alfa se realizan en un ambiente controlado.

**Pruebas Beta:** La prueba beta se realiza en uno o más sitios del usuario final. A diferencia de la prueba alfa, por lo general el desarrollador no está presente. Por tanto, la prueba beta es una aplicación “en vivo” del software en un ambiente que no puede controlar el desarrollador. El cliente registra todos los problemas (reales o imaginarios) que se encuentran durante la prueba beta y los reporta al desarrollador periódicamente. Como resultado de los problemas reportados durante las pruebas beta, es posible hacer modificaciones y luego preparar la liberación del producto de software a toda la base de clientes.

Para la realización de las pruebas alfas se tomaron en cuenta personas, que han desempeñado algún rol en el desarrollo de videojuegos, a las cuales se puso a su disposición una guía de jugabilidad referenciada en el anexo A.5:

- Ing. Enelis Blanca Cuba Rondón
- Ing. Felix Armando Abreu Hernández
- Kevin Guimera Abad (Desarrollador)
- Daniel Bagué Calvo (Desarrollador)

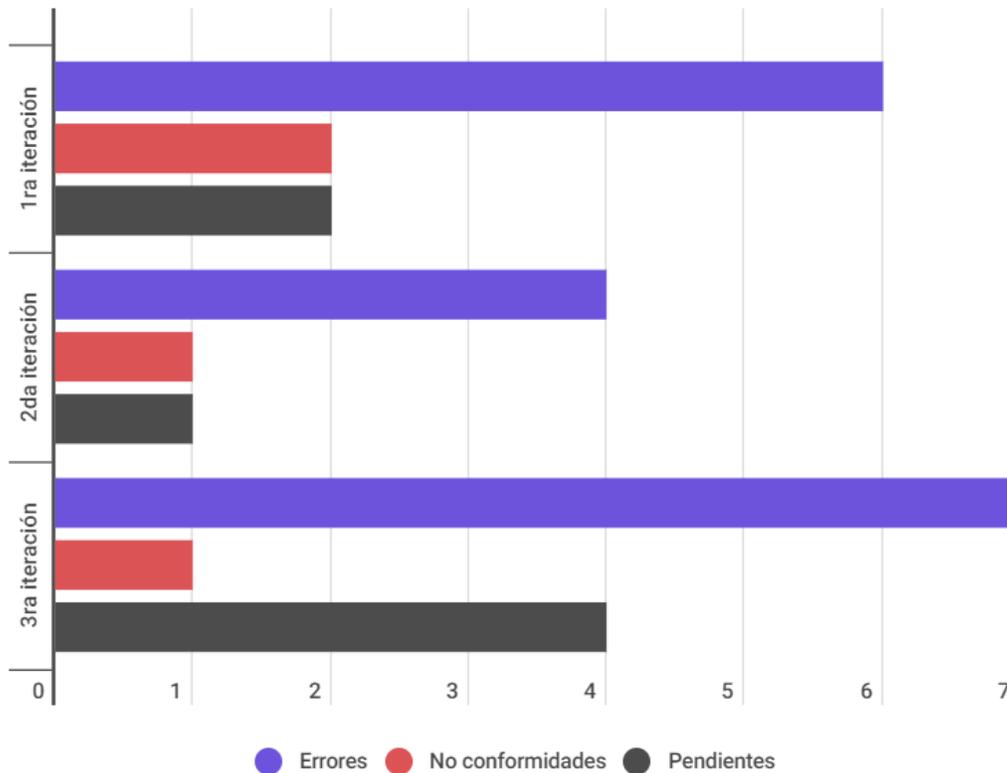
De la misma forma se seleccionaron un grupo de personas ajenas al desarrollo que ejecutarán las pruebas betas en busca de errores o inconformidades.

- Ing. Yanays Fernández Miranda
- Daniel Chaviano Pérez (Estudiante universitario de 3er año de Ingeniería en Ciencias Informáticas)
- Manuel Enrique Varela Rodríguez (Estudiante universitario de 5to año de Ingeniería en Ciencias Informáticas)
- Yuliet Guillot Calvo (Estudiante de 10mo grado)

Para la fase de pruebas alfas se ejecutaron 3 iteraciones de las mismas, las cuales arrojaron una cantidad determinada de errores y no conformidades, se pueden encontrar en los anexos A.2, las cuales se fueron corrigiendo antes de avanzar a la siguiente iteración.

La gráfica siguiente muestra el comportamiento de los resultados obtenidos con las pruebas realizadas.

## Gráfico de no conformidades y errores



### Pruebas Alfas

Gráfico 1. Resultado de las pruebas alfas

### 3.4. Conclusiones parciales

Al finalizar el presente capítulo quedó desarrollado satisfactoriamente la solución propuesta, logrando así la implementación de una versión beta de un videojuego que sirva para el enriquecimiento de la plataforma Cosmox. Se definieron los estándares de codificación y se representaron los diagramas de componente lo que permite un mejor entendimiento de la implementación. Se definieron y aplicaron las pruebas alfas, con lo que se elevó la calidad del software a partir de los resultados obtenidos y las acciones que se llevaron a cabo para darle solución a las no conformidades encontradas.

## CONCLUSIONES

El desarrollo de la presente investigación permitió arribar a las siguientes conclusiones generales:

- La elaboración del diseño teórico ratificó la necesidad de enriquecer la plataforma Cosmox con productos nacionales como el videojuego de supervivencia propuesto.
- La investigación acerca de las tendencias tecnológicas actuales, técnicas y características utilizadas para el desarrollo de videojuegos de supervivencia, mostró elementos a tener en cuenta en la solución, como fueron el motor de videojuego Unity y el lenguaje de programación C#.
- El empleo de patrones de diseño, el uso de una arquitectura por capas y la elaboración de mecanismos de jugabilidad, posibilitaron obtener un videojuego que da cumplimiento al objetivo general de la presente investigación.
- Las pruebas realizadas permitieron verificar que el videojuego y los mecanismos desarrollados cumplen con la solución propuesta, así como el funcionamiento que se espera de la jugabilidad y la experiencia del usuario, validando además la calidad de la solución.

## RECOMENDACIONES

1. Agregar más elementos visuales que enriquezcan la experiencia del jugador y del entorno.
2. Optimización del videojuego para mejorar el desempeño.
3. Implementación del consumo de servicios de Cosmox.

## REFERENCIAS

Amberes.2017. ¿Por qué están los adolescentes más enganchados a los videojuegos? ¿Y yo?. <https://amberesrevista.com/por-que-estan-los-adolescentes-mas-enganchados-a-los-videojuegos-y-yo>.

«Videojuegos hechos en Cuba: la partida apenas ha comenzado». 2021. Granma.cu. <https://www.gramma.cu/cuba/2021-11-17/videojuegos-hechos-en-cuba-la-partida- apenas-ha-comenzado-17-11-2021-23-11-20>.

«Juego Tamaño del mercado, crecimiento, industria». 2022. <https://www.mordorintelligence.com/es/industry-reports/global-gaming-market>.

«La industria del videojuego». 2020. MeriStation. 26 de diciembre de 2020. [https://as.com/meristation/2020/12/26/noticias/1608992024\\_963325.html](https://as.com/meristation/2020/12/26/noticias/1608992024_963325.html).

Iglesias, Alejandro A..2011. «Tesis Alejandro Adrián Iglesias.pdf».

Behrmann M, Noyons M, Johnstone B, MacQueen D, Robertson E, Palm T and Point J (2012). «State of the Art of the European Mobile Games Industry». Mobile Game Arch Project. Archivado desde el original el 2 de febrero de 2014. Consultado el 12 de agosto de 2013.

Cultura Geek. (sf). ¿De qué se trata Rust? El nuevo juego de supervivencia que gana popularidad en Twitch. <https://culturageek.com.ar/de-que-se-trata-rust-el-nuevo-juego-de-supervivencia-que-gana-popularidad-en-twitch>.

MinecraftWkiEs. (sf). Minecraft. [https://minecraft.fandom.com/es/wiki/Minecraft\\_Wiki](https://minecraft.fandom.com/es/wiki/Minecraft_Wiki).

Visual Paradigm [En línea.] (sf). <https://www.visual-paradigm.com>.

Rondón, E.B. Cuba 2012. Fusión de la información de los servicios de la comunidad universitaria de la UCI con el metaverso OpenSim. La Habana: Universidad de las Ciencias Informáticas.

Omar Correa, Andy Hernández Paez Karina Pérez T. «Marco de trabajo ingenieril para el proceso de desarrollo de videojuegos» (2017).

Vazquez, Leonel. 2018. «Videojuego para la rehabilitación cognitiva enfocado en la percepción».

Ambler, Scott. 2017. *The Elements of UML 2.0 Style*.

Adams, E, y J Dormans. 2018. *Game Mechanics, Advance Game Designs. United States of América*. ISBN 978-0-321-82027-3.

Ampuero, Margarita André, y Yucely López Trujillo. 2017. *Creando un profesional con disciplina en el proceso de desarrollo de software*.

Coterón, L.S. 2018. «Arte y videojuegos: mecánicas, estéticas y diseño de juegos en prácticas de creación contemporánea.» *Universidad Complutense de Madrid*.

Cruz, Alejandro. 2017. Trabajo de diploma. Arquitectura de software para videojuegos desarrollados sobre el motor de juego Unity 3D.

Craig, Larman. 2018. *UML y patrones*.

E. Pedersen, Roger. 2014. *Game Design Foundations*.

Hernández, Paez Andy. 2017. «Marco de trabajo ingenieril para el proceso de desarrollo de videojuegos». Maestría, Universidad de las Ciencias Informáticas.

Ivar, Booch Jacobson. 2018. *El proceso unificado de desarrollo de software/The unified software development process*. 004.41.

López, Ramón. 2016. «Unity Technologies». <https://docs.unity3d.com>.

Lopez, Frankie. 2022. OpenWebinars. 22 de Julio, 2022. Metodologías y Herramientas.

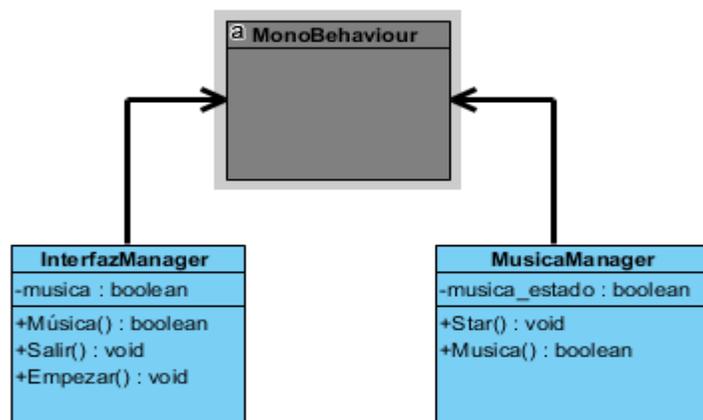
Pressman, Roger S, y Jose Maria Troya. 2020. *Ingeniería del software*.

Sommerville, IAN. 2011. *Ingeniería del software*.

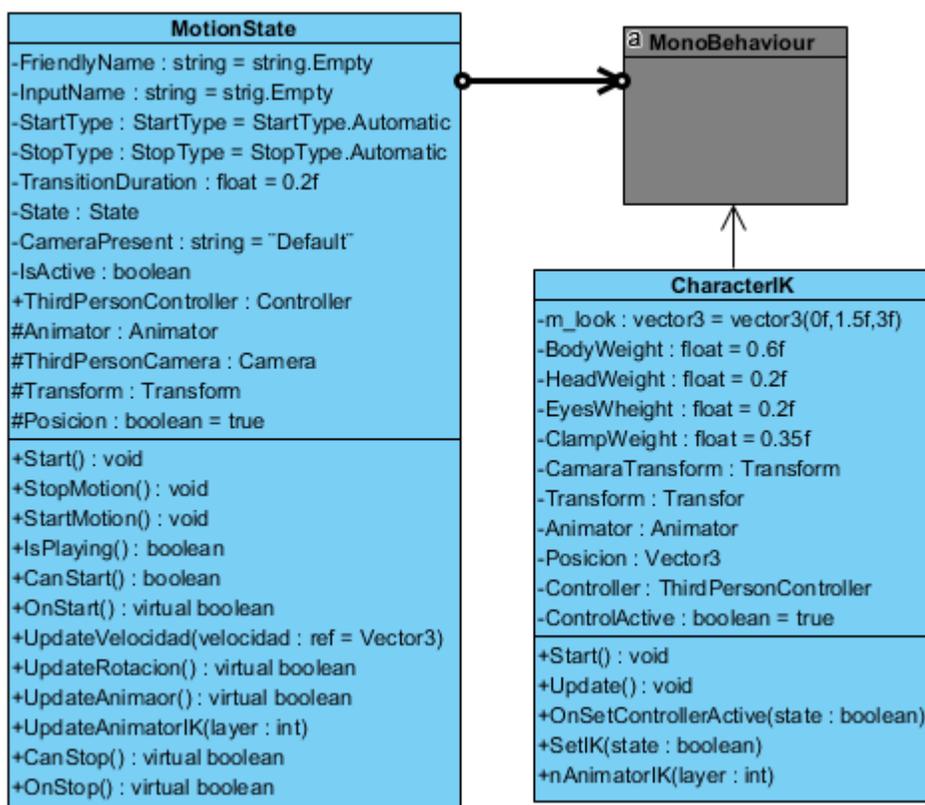
## ANEXOS

## A.1 Diagrama de clases

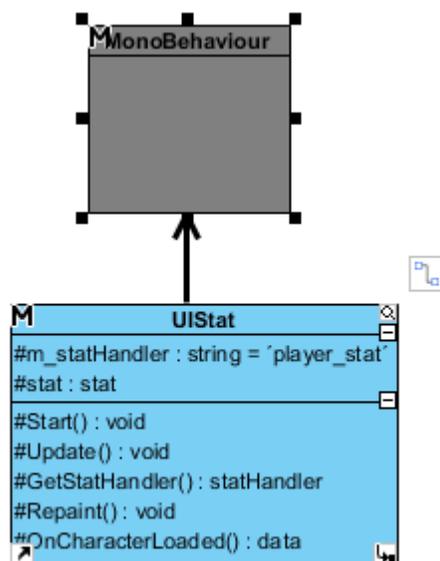
## A.1.2 Diagrama de clase del mecanismo interfaz



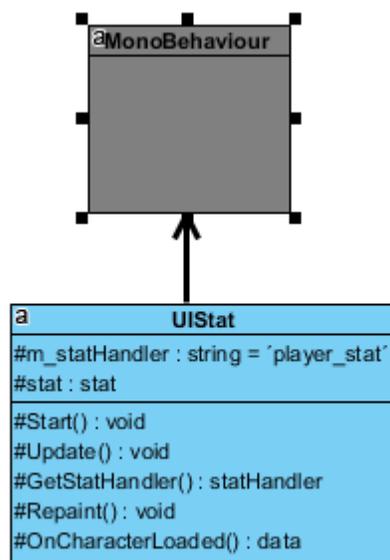
## A.1.3 Diagrama de clase del mecanismo locomoción



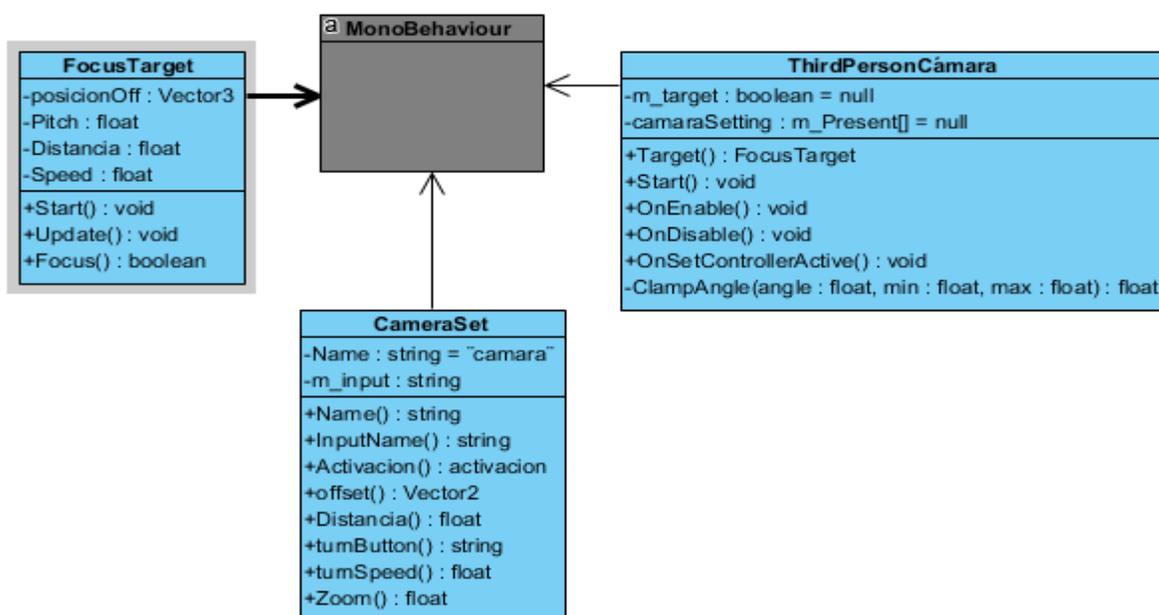
## A.1.4 Diagrama de clase del mecanismo barra de vida



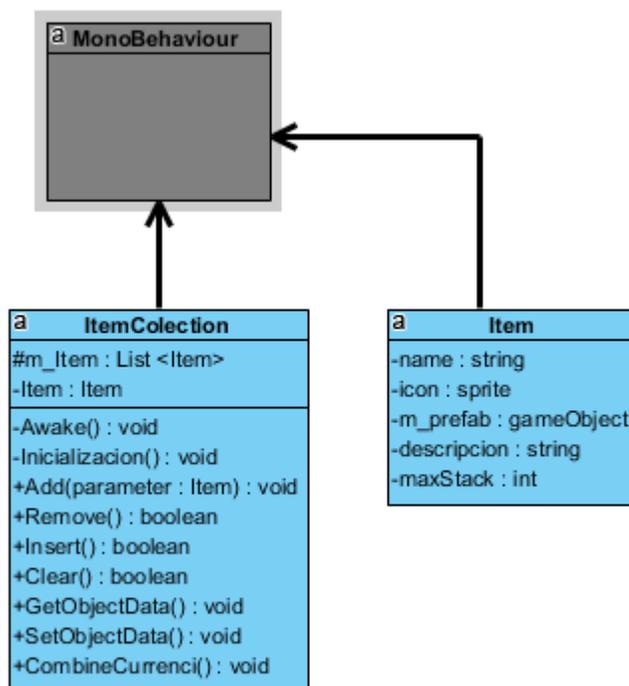
## A.1.5 Diagrama de clase del mecanismo barra de hambre



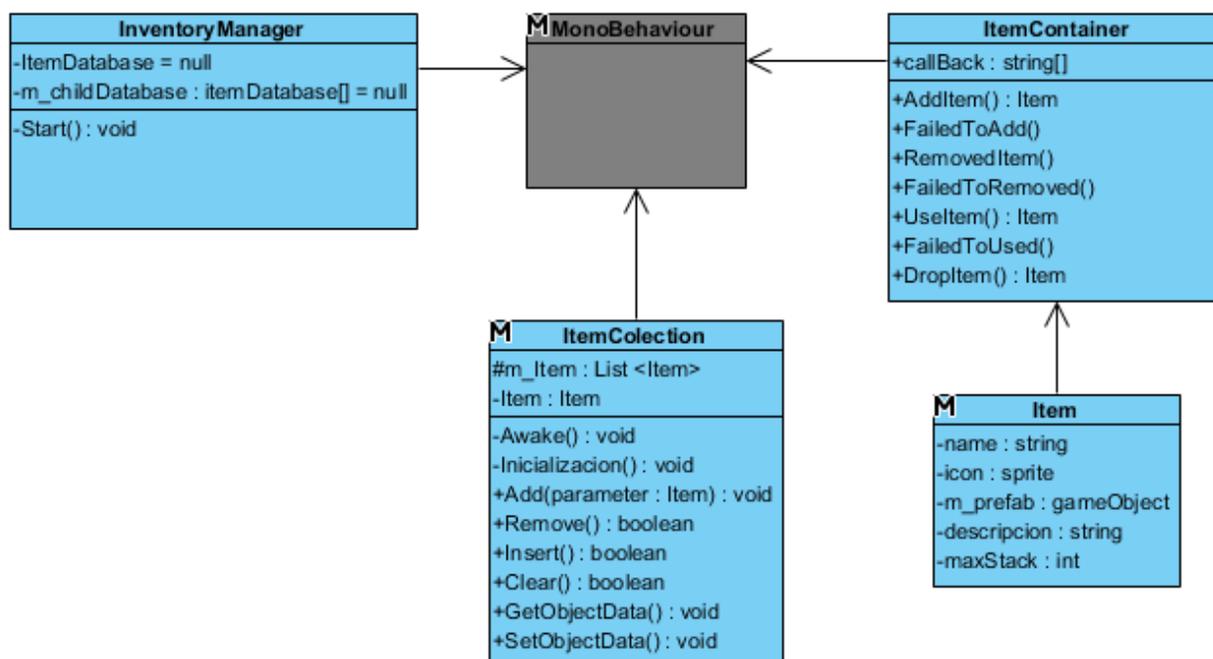
## A.1.6 Diagrama de clase del mecanismo cámara



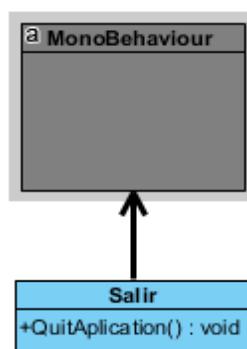
## A.1.7 Diagrama de clase del mecanismo creación



## A.1.8 Diagrama de clase del mecanismo inventario



## A.1.9 Diagrama de clase del mecanismo abandonar



## A.2 Errores y no conformidades registradas durante las pruebas alfas

Iteración	Número	Prioridad	Descripción	Ubicación	Tipo	Respuesta
1	1.1	Baja	No se mostraba la pantalla de carga al iniciar la partida.	Escena de carga	Error de Interfaz	Resuelto

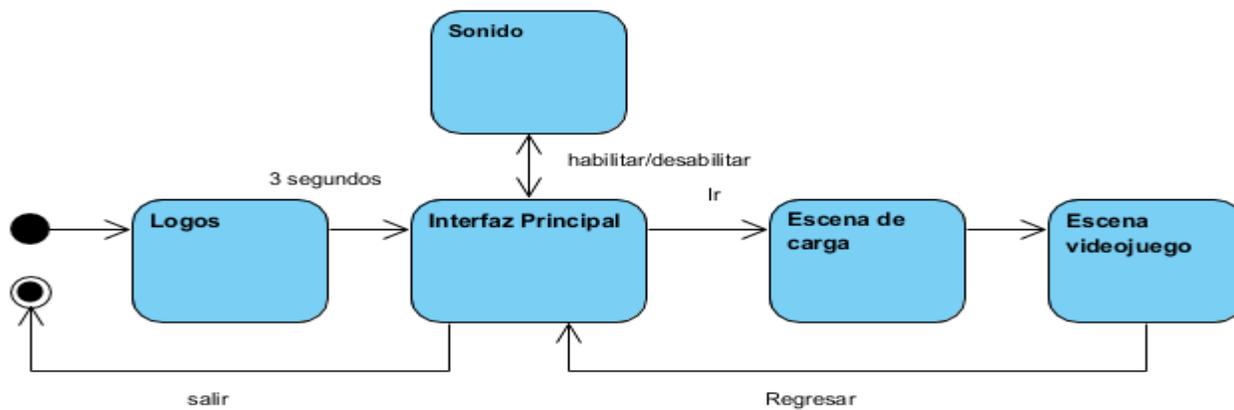
1	1.2	Media	El botón de música no activa ni desactiva el sonido	Escenario del menú principal	Funcionalidad	Pendiente
1	1.3	Baja	La animación del menú principal no se ejecuta correctamente	Escenario del menú principal	Funcionalidad	Resuelto
1	1.4	Baja	Los botones no resultan agradables a la vista del jugador	Escenario del menú principal	Apariencia o interfaz externa	Resuelto
1	1.5	Media	El jugador nada por encima de la superficie del agua	Escenario del videojuego	Funcionalidad	Pendiente
1	1.6	Alta	Las armas no se equipan	Escenario del videojuego	Funcionalidad	Resuelto
1	1.7	Alta	Los árboles se hunden en el suelo	Escenario del videojuego	Funcionalidad	Resuelto
1	1.8	Baja	El inventario y los distintos menús del videojuego tienen un aspecto muy	Escenario del videojuego	Apariencia o interfaz externa	Resuelto

			básico			
2	2.1	Alta	Los objetos creados no se guardan en el inventario	Escenario del videojuego	Funcionalidad	Resuelto
2	2.2	Media	Palabras en inglés presentes en los menús	Escenario del videojuego	Apariencia o interfaz	Resuelto
2	2.3	Alta	El jugador no puede recoger materiales	Escenario del videojuego	Funcionalidad	Resuelto
2	2.4	Media	Los NPC no reciben daño	Escenario del videojuego	Funcionalidad	Resuelto
2	2.5	Baja	La animación de la barra de carga no se ejecuta	Escena de carga	Funcionalidad	Pendiente
3	3.1	Alta	El jugador deja de responder y se desliza solo por el suelo	Escenario del videojuego	Funcionalidad	Resuelto
3	3.2	Alta	Al iniciar la partida aparece una fogata que sigue al jugador	Escenario del videojuego	Funcionalidad	Resuelto
3	3.3	Media	Los carteles cuando te	Escenario del videojuego	Funcionalidad	Pendiente

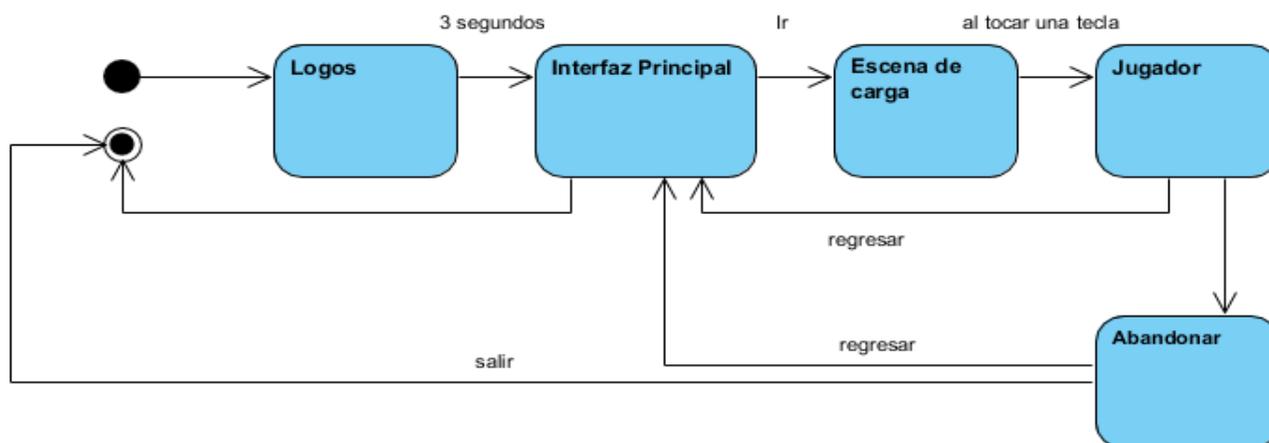
			aproximas a algún objeto para ejecutar una acción no salen			
3	3.4	Alta	La primera misión no se completa incluso al cumplir con los objetivos de la misma	Escenario del videojuego	Funcionalidad	Resuelto
3	3.5	Alta	Se crean los objetos sin recursos	Escenario del videojuego	Funcionalidad	Resuelto
3	3.6	Alta	Las animaciones de los animales no se ejecutan	Escenario del videojuego	Funcionalidad	Pendiente
3	3.7	Media	Las armas no se muestran al ser equipadas	Escenario del videojuego	Funcionalidad	Pendiente
3	3.8	Media	La cámara que muestra la cara del jugador junto a las barras de vida y hambre sobrecarga el sistema	Escenario del videojuego	Rendimiento	Pendiente

### A.3 Diagramas de estado

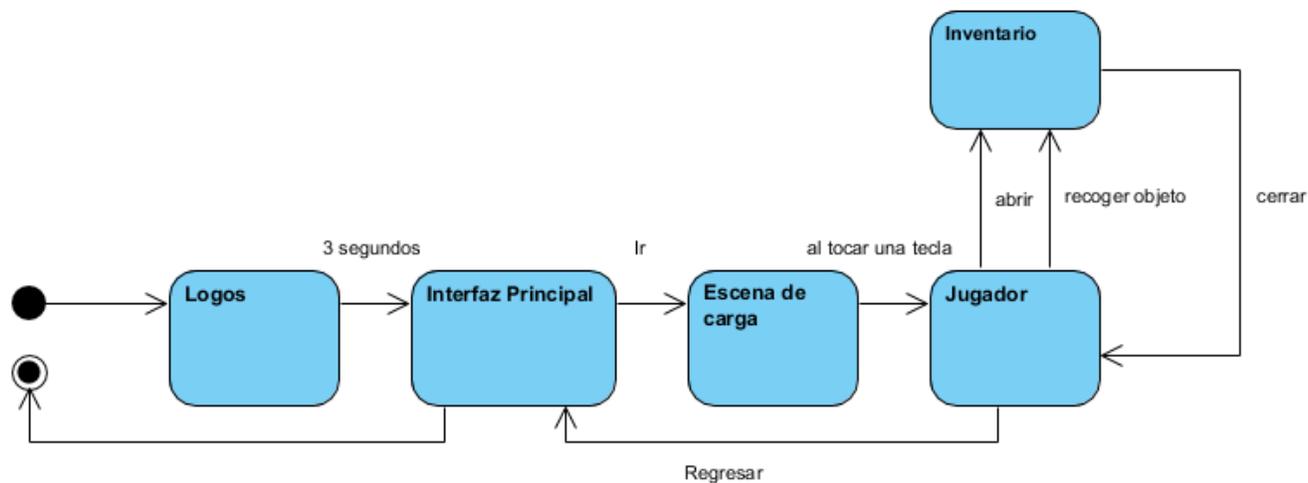
#### A.3.1 Diagrama de estado del mecanismo interfaz principal



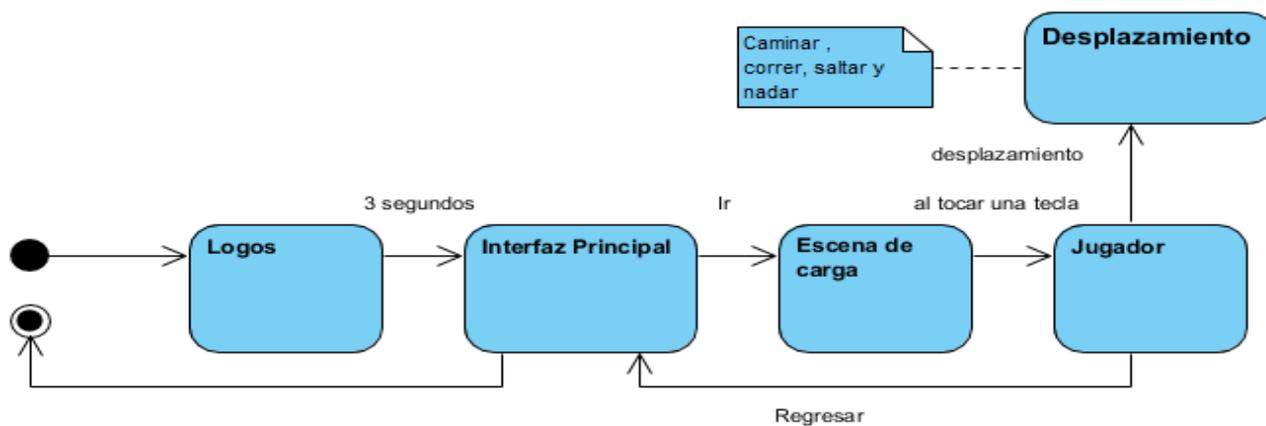
#### A.3.2 Diagrama de estado del mecanismo abandonar



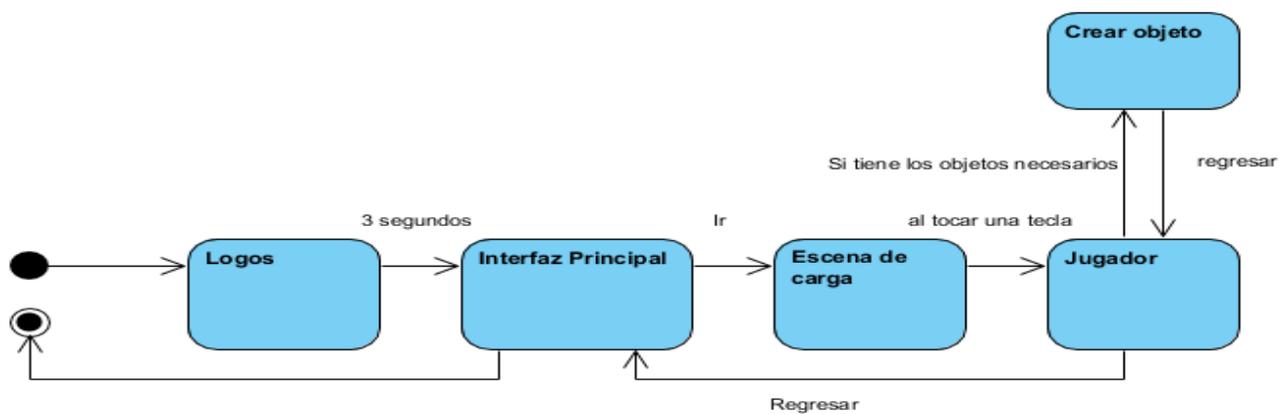
A.3.3 Diagrama de estado del mecanismo inventario



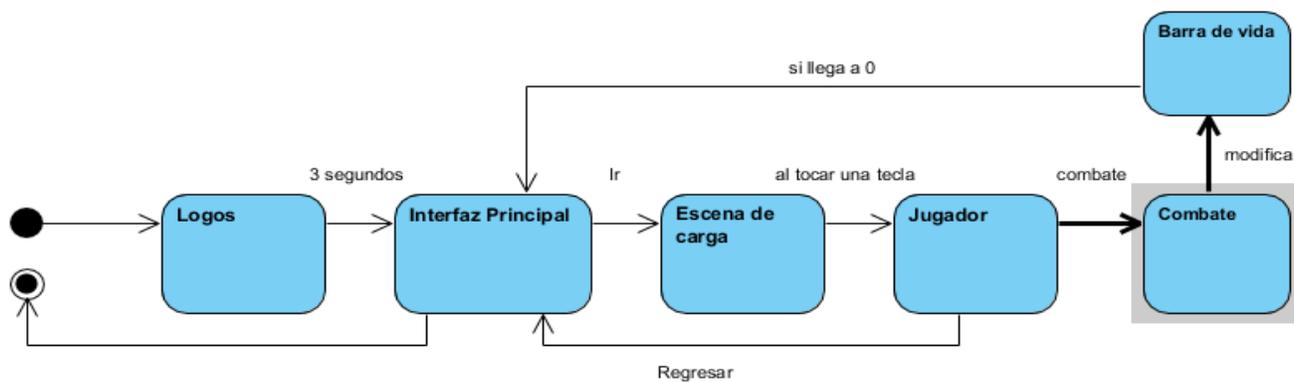
A.3.4 Diagrama de estado del mecanismo locomoción



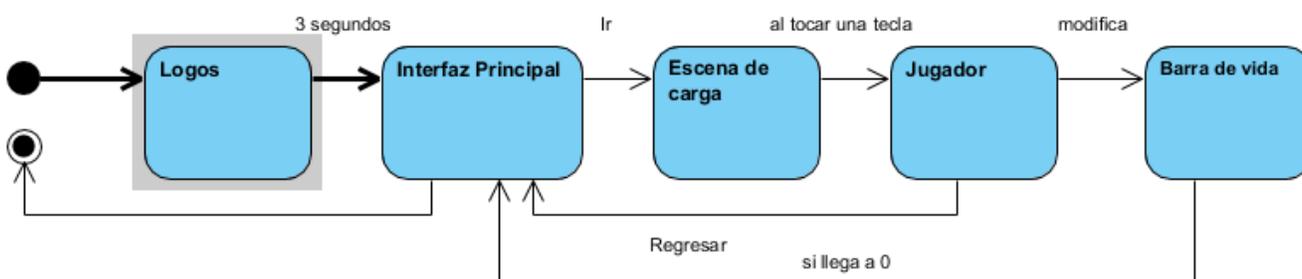
A.3.5 Diagrama de estado del mecanismo crear objeto



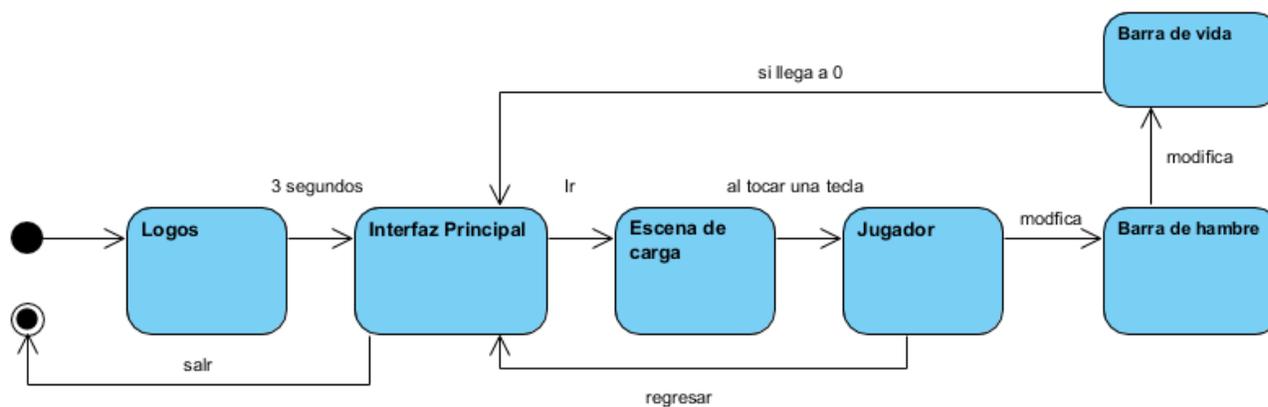
## A.3.6 Diagrama de estado del mecanismo combate



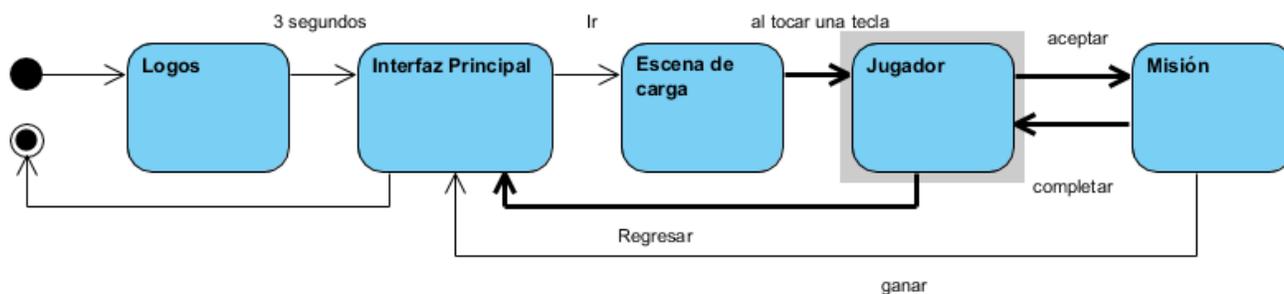
## A.3.7 Diagrama de estado del mecanismo gestión de vida



## A.3.8 Diagrama de estado del mecanismo gestión de hambre

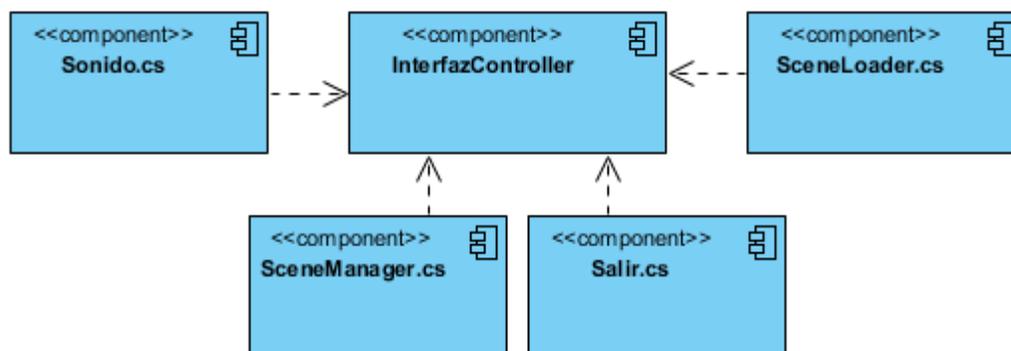


## A.3.9 Diagrama de estado del mecanismo misión

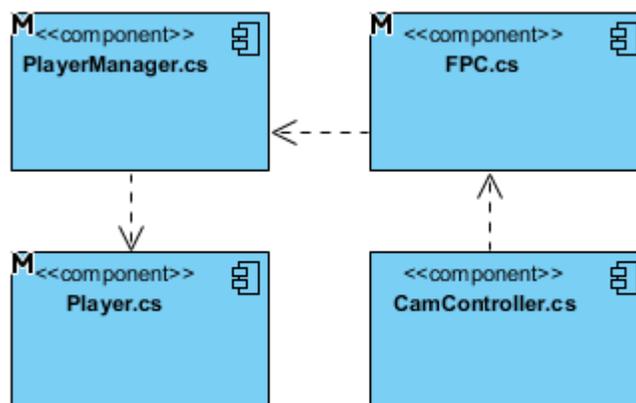


## A.4 Diagrama de componentes

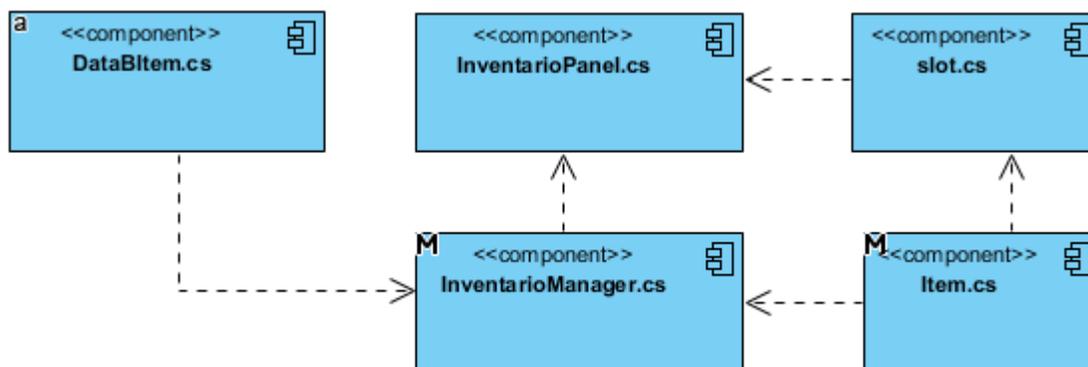
## A.4.1 Diagrama de componente del mecanismo gestionar interfaz



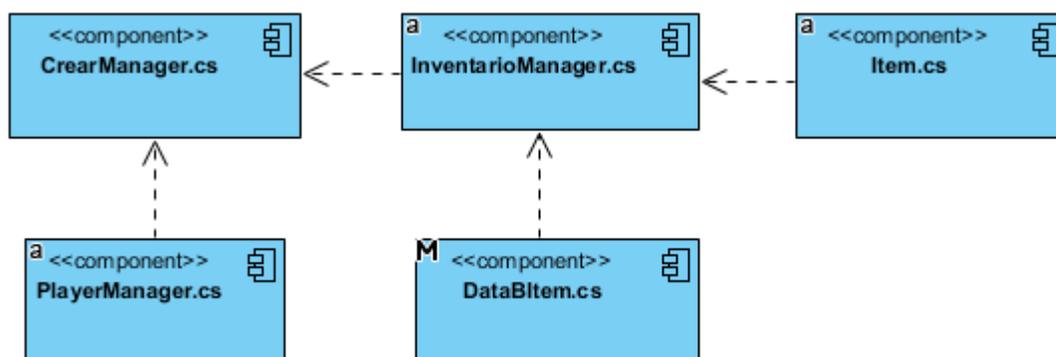
## A.4.2 Diagrama de componente del mecanismo locomoción



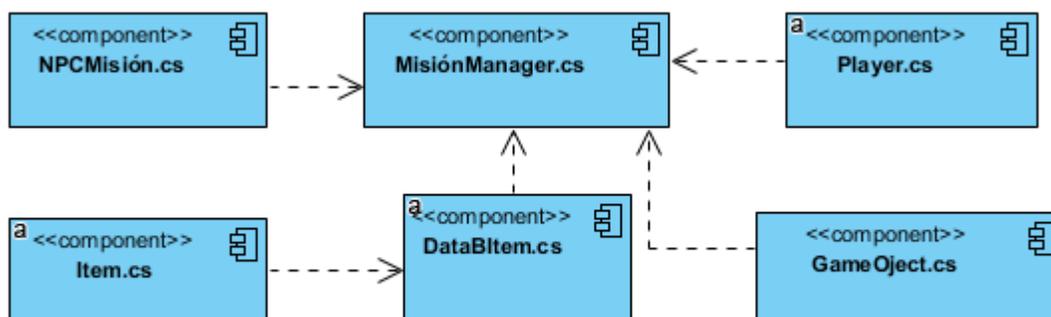
## A.4.3 Diagrama de componente del mecanismo administrar inventario



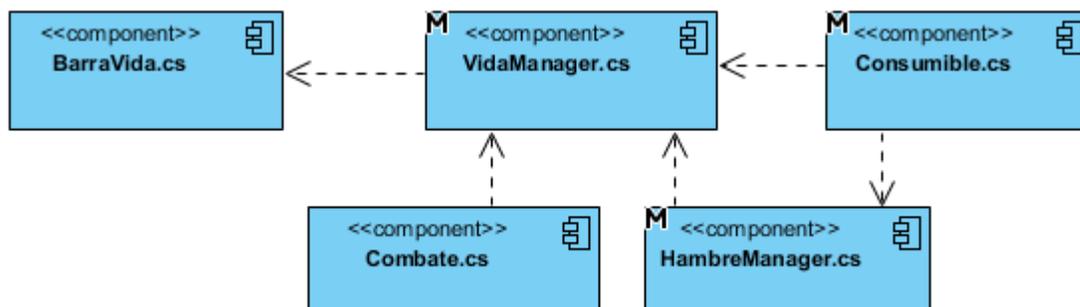
## A.4.4 Diagrama de componente del mecanismo crear objeto



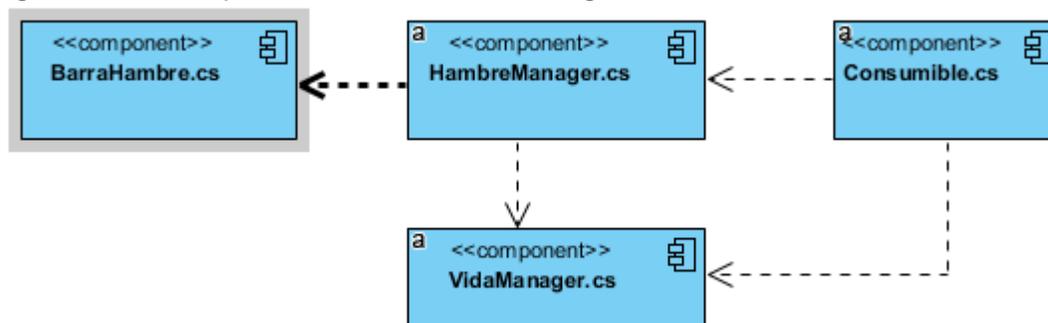
## A.4.5 Diagrama de componente del mecanismo administrar misión



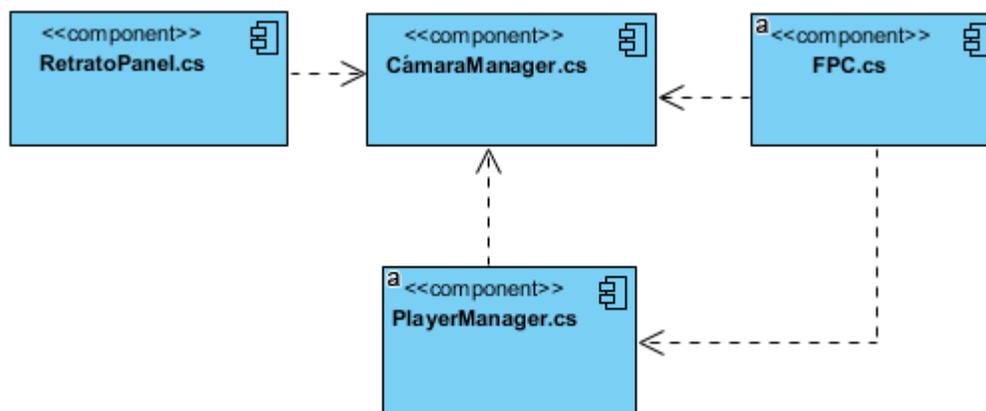
## A.4.6 Diagrama de componente del mecanismo gestionar vida



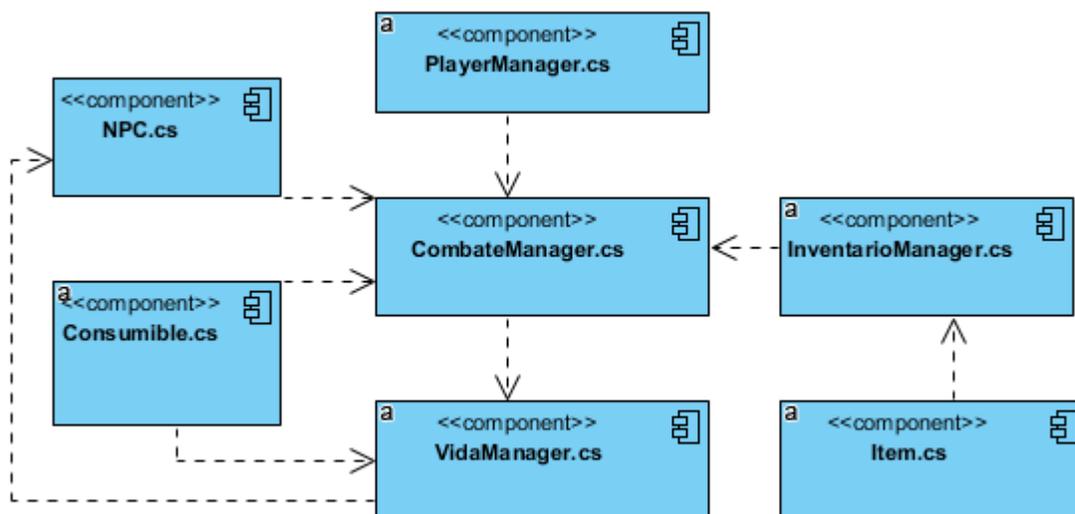
## A.4.7 Diagrama de componente del mecanismo gestionar hambre



## A.4.8 Diagrama de componente del mecanismo cámara



## A.4.9 Diagrama de componente del mecanismo combate



## A.5 Guía de jugabilidad del videojuego “La Isla”

- 1.El jugador abrirá el juego y accederá a la escena Menú Principal, donde podrá realizar las siguientes acciones:
  - 1.1. Desactivar o activar el sonido del videojuego seleccionando el botón Música
  - 1.2. Salir de la aplicación seleccionando el botón Salir
  - 1.3. Empezar una partida seleccionando el botón Iniciar Aventura
- 2.Al empezar la partida el jugador podrá acceder a las siguientes funciones:
  - 2.1. Acceder al menú de pausa presionando la tecla “ESC”
  - 2.2. Dentro del menú de pausa podrá guardar la partida, cargar la partida, acceder a las opciones o salir de la aplicación
  - 2.3. Al abrir las opciones, el jugador, puede acceder a la configuración de video, audio, controles, idioma o salir de este menú presionando el botón de salir en la esquina superior derecha de la pantalla
  - 2.4. Acceder al inventario presionando la letra “i”
  - 2.5. Acceder al menú de creación de objetos al interactuar con el yunque
  - 2.6. Recibir misiones al interactuar con los NPC que tengan una cápsula sobre sus cabezas
- 3.Al vencer la misión final se debe mostrar la pantalla de inicio.