

**VIDEOJUEGO  
NIGHTMARES OF ZOMBIES**

**JHOAN SEBASTIAN CHAVEZ QUIROGA  
CAROLINA ROZO ARIAS**

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

**VIDEOJUEGO  
NIGHTMARES OF ZOMBIES**

**JHOAN SEBASTIAN CHAVEZ QUIROGA  
CAROLINA ROZO ARIAS**

Trabajo de grado para optar el  
título de Tecnólogo en Informática.

Asesor:

ING. Sócrates Rojas Amador

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

**Nota de aceptación**

---

---

---

---

---

\_\_\_\_\_  
**Firma del presidente del jurado**

\_\_\_\_\_  
**Firma del jurado No. 1**

\_\_\_\_\_  
**Firma del jurado No. 2**

Bogotá D.C. / 16/ Enero del 2015

## **Dedicatoria**

Dedicado a nuestras familias por apoyarnos en esta etapa primordial para alcanzar nuestros sueños. Gracias por seguir siendo nuestra guía, estamos creciendo para que se sientan orgullosos de nosotros. Seguiremos siendo personas responsables, honestas y comprometidas con los proyectos que componen nuestro plan de vida.

## **Agradecimientos**

Agradecemos a la Corporación Universitaria Minuto de Dios por darnos la oportunidad de formarnos con educación de la calidad y seguir creciendo profesionalmente. A los profesores que desde el primer semestre nos brindaron su apoyo y tiempo para corregir nuestros errores.

Y por último a nuestros compañeros que nos aportaron conocimientos y una amistad a lo largo de la carrera.

## TABLA DE CONTENIDO

	Pág.
<b>INTRODUCCION</b>	
1.1 Nombre del proyecto .....	4
1.2 Planteamiento del problema.....	4
1.3 Alcance y justificación.....	4
1.4 Objetivos	
1.4.1 Objetivo General.....	6
1.4.2 Objetivos Específicos.....	6
<b>2. Ingeniería del proyecto</b>	
2.1 Modelo de desarrollo .....	7
<b>3. Análisis y diseño</b>	
3.1 Definición de requerimientos.....	12
3.2 Descripción del sistema propuesto.....	18
3.3 Diseño del sistema propuesto.....	20
<b>4. Desarrollo</b>	
4.1 Especificaciones técnicas.....	30
4.1.1 Software.....	31
4.1.2 Hardware.....	32
<b>5. Glosario</b> .....	33
<b>6. Conclusiones</b> .....	35
<b>7. Bibliografía</b> .....	36
<b>8. Manuales</b> .....	38
8.1 Manual del sistema.....	II
8.2 Manual del usuario.....	II

## LISTA DE GRÁFICOS

	<b>Pág.</b>
Gráfico N°1 Gráfico Metodología SCRUM.....	9
Gráfico N°2 Implementación Metodología SCRUM.....	10
Gráfico N°3 Implementación Metodología SCRUM.....	11

## LISTA DE DIAGRAMAS

	<b>Pág.</b>
Diagrama N° 1: Diagrama de clases .....	22
Diagrama N° 2: Diagrama caso de uso del usuario .....	26
Diagrama N° 3: Diagrama de caso de uso general .....	27
Diagrama N° 4: Diagrama de secuencia .....	29



## LISTA DE TABLAS

	<b>Pág.</b>
Tabla N° 1: Caso de uso Menú .....	23
Tabla N° 2: Caso de uso Ingresar .....	24
Tabla N° 3: Caso de uso Jugador .....	24
Tabla N° 4: Caso de uso Cargar escenario .....	25
Tabla N° 5: Caso de uso Salir .....	25

## TABLA DE INTERFACES

	Pág.
Interfaz N°1: Menú inicio .....	18
Interfaz N° 2: Nombre de usuario .....	19
Interfaz N°3: Cargar Partida .....	19
Interfaz N° 4: Salir .....	20

### Interfaces Manuales

#### *Manual del Sistema*

Interfaz N° 5: Funciones primera escenario .....	3
Interfaz N° 6: Nombre del jugador .....	4
Interfaz N° 7: Interfaz ciudad.....	5
Interfaz N° 8: Efectos.....	7

#### *Manual del Usuario*

Interfaz N° 9: Inventario.....	1
Interfaz N° 10: Inventario.....	2
Interfaz N° 11: Escenario.....	3
Interfaz N° 12: Interfaz Jugador.....	4
Interfaz N° 13: Menú de pausa 1.....	5
Interfaz N° 14: Menú de pausa 2.....	5
Interfaz N° 15: Mapa 1.....	6
Interfaz N° 16: Mapa 2.....	7
Interfaz N° 17: Guardar partida 1.....	8
Interfaz N° 18: Guardar partida 2 .....	8

## RESUMEN

Este documento representa el video juego Nigthmares of Zombies. Un video juego con género Survival Horror dirigido a los fanáticos de películas de horror y zombies. Desarrollado en el programa Unity con ambientación en 3D, colisión para personajes, texturas, sonidos reales y además cuenta con una tienda Asset Store para exportar y visualizar gráficos, descargándolos al espacio de trabajo. También se utiliza el lenguaje de programación Java Script, que es el código que genera acciones sobre el personaje y los enemigos.

Nigthmares of Zombies recrea una ciudad llena de zombies, animales extraños y ruidos tenebrosos. El personaje tiene a su disposición cinco armas para protegerse de los enemigos y lograr un tiempo determinado en el juego. Una de las ventajas que tiene el personaje es estar en primera persona, así puede visualizar y sentir que tiene cerca a los enemigos. Para resguardarse y mantenerse con vida en el juego, las armas son potentes, la puntería será la clave para eliminar a los zombies.

## **ABSTRACT**

This document represents the video game Nighthmares of Zombies. A video game survival horror genre aimed at fans of horror movies and zombies. Developed on the Unity 3D program setting, collision to characters, textures, realistic sounds and also has a shop Asset Store to export and display graphics, downloaded to the workspace. The programming language Java Script, which is the code that generates actions on the character and enemies are also used.

Nighthmares of Zombies recreates a city full of zombies, weird animals and dark noise. The character has available five weapons to protect themselves from enemies and achieve a given time in the game. One of the advantages of the character is to be in first person and can see and feel it has some enemies. To guard and stay alive in the game, weapons are powerful, the aim will be key to eliminate the zombies.

## 1. INTRODUCCIÓN

Este proyecto de grado se enfoca en el desarrollo y diseño de un software poco común en Colombia. Pocos desarrolladores se están enfocando en video juegos y por el momento son pequeños proyectos que tienden a hacer innovadores en el mercado. Existen varios programas con tutoriales y ayudas extras para construir un video juego y poder exportarlo en navegadores o plataformas.

El enfoque del software es afianzar las habilidades que tienen los jugadores frente a diversos juegos, en este caso de horror, ya que este es poco común en sitios web, juegos para PC y consolas. Buscamos que los llamados “gamers” fanáticos de videojuegos conozcan y compartan con los demás la experiencia de jugar Nigthmares of Zombies. Así mismo compartir este juego por medio de redes sociales y sitios de interés, contando con la participación de estudiantes de la Universidad Minuto de Dios.

Cada etapa de desarrollo del juego fue guiado por los conocimientos que se aprendiendo en cada semestre, adicional a esto por cursos autónomos y apoyo del asesor que tiene énfasis en desarrollo de videojuegos. Además se realizaron investigaciones para afianzar los conceptos y llevar paso a paso la construcción del software, con el apoyo de la Librería de Unity, motor gráfico para el desarrollo de videojuegos.

### **1.1. Nombre del proyecto**

Nightmares of zombies (pesadillas de zombies)

### **1.2. Planteamiento del Problema**

Este juego tiene el género Survival Horror con el nombre de Nightmares of zombies (pesadillas de zombies), el cual tiene como objetivo entretener, divertir y reunir a la gente, este juego se desarrolla en instalaciones secretas del gobierno las cuales están ubicadas en un bosque el cual contiene árboles, pasto y zombies, este juego será multijugador además cuentan con 2 armas cada uno la primera es una metralleta y la segunda una escopeta, este juego busca asustar a la gente pero también entretenerla ya que deberán pasar 2 capítulos de esta emocionante aventura, este juego es multiplataforma para que puedan jugarlo en diversos dispositivos.

### **1.3. Alcance y Justificación**

Nightmares of zombies es un juego del género Survival Horror, está diseñado con unainterfaz 3D además tiene un escenario en el bosque con árboles, pasto, de día o de noche el cual se encuentra cerca de unas instalaciones militares secretas en la cual surgió el virus, está constituido en 2 capítulos los cuales se centran en la invasión zombie además este juego es multiplataforma, el número de jugadores permitidos es de 2 aunque también se puede disfrutar de 1 jugador, el jugador contara con 2 armas las cuales serán una metralleta y una escopeta las cuales

serán utilizadas para matar a los zombies, el programa que se escogió para desarrollar este videojuego es Unity 3D ya que este programa nos da la facilidad de crear juegos en 3D sin mencionar que uno de los lenguajes de programación utilizados es JavaScript y tiene una gran variedad de modelos 3D por esta razón lo escogimos, este juego es para personas mayores a 18 años por el contenido de violencia.

Actualmente el país no cuenta con muchos desarrolladores de videojuegos ya que este mercado está empezando a florecer y busca alternativas para su implementación. Queremos ser los primeros en modernizar los géneros de videojuegos que se desarrollan en Colombia, nos centramos en el género Survival Horror ya que actualmente hay muchos amantes de este tipo incluyéndonos además queremos darle la oportunidad a varias personas amantes de los zombies en que puedan jugar, disfrutar y compartir con este tipo de videojuego.

## **1.4 Objetivos**

### **1.4.1 Objetivo General**

Desarrollar un videojuego que entretenga a personas seguidoras del género Survival Horror. Facilitando el modo multijugador y multiplataforma para que los jugadores tengan mayores posibilidades de jugarlo, añadir animaciones y sonidos al personaje para que el juego sea más realista, destacan los detalles de cada persona y sus movimientos dentro del juego.

### **1.4.2 Objetivos Específicos**

- Desarrollar un software que permita reflejar películas de zombies en un juego divertido y fácil para aficionados del género Survival Horror.
- Determinar todos los componentes necesarios para el desarrollo y diseño del videojuego, con el fin de obtener un sistema de alto rendimiento y ágil ejecución.
- Manejar diferentes programas para el desarrollo de gráficos reales, sonidos y ambientación facilitando al jugador una interfaz real y fácil de jugar.



## 2. INGENIERÍA DEL PROYECTO

### 2.1 Modelo de Desarrollo

El proyecto se irá implementando con la Metodología SCRUM, facilitando la gestión de tareas y recursos basados en un desarrollo ágil de software.

#### **Metodología SCRUM**

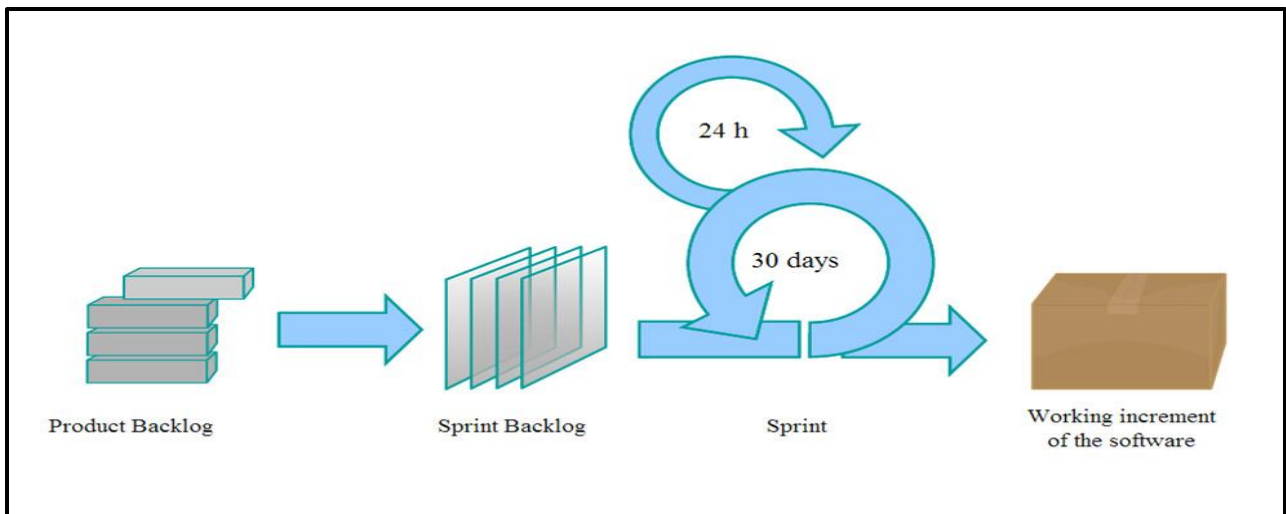
Es una metodología ágil y flexible para gestionar el desarrollo de software, cuyo principal objetivo es maximizar el retorno de la inversión para su empresa. Se basa en construir primero la funcionalidad de mayor valor para el cliente y en los principios de inspección continua, adaptación, auto-gestión e innovación. En SCRUM se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, SCRUM está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

## *Proceso SCRUM*

- ProductBacklog: funcionalidades del producto. Elaborado por el Product Owner.
- Sprint Backlog: subconjunto de ítems del ProductBacklog, que son seleccionados por el equipo para realizar durante el Sprint sobre el que se va a trabajar.
- Sprint Planning Meeting: Esta reunión se hace al comienzo de cada **Sprint** y se define cómo se va a enfocar el proyecto que viene del ProductBacklog las etapas y los plazos.
- DailyScrum: Es una reunión breve que se realiza a diario mientras dura el periodo de Sprint. Se responden individualmente tres preguntas: ¿Qué hice ayer?, ¿Qué voy a hacer hoy?, ¿Qué ayuda necesito?
- Sprint Review: Se revisa el sprint terminado, y ya debería haber un avance claro y tangible para presentárselo al cliente.

## *Roles en SCRUM*

- ProductOwner: voz del cliente. Se asegura que el proceso Scrum trabaje de forma desde la perspectiva del software.
- Scrum Master: Lidera las reuniones y ayuda al equipo si es que tienen problemas. Además, minimiza los obstáculos para cumplir el objetivo del Sprint, es un “facilitador” pero no es un gestor.
- ScrumTeam: Son los encargados de desarrollar y cumplir lo que les asigna el ProductOwner.



Fuente: Introducción a SCRUM

[https://upload.wikimedia.org/wikipedia/commons/5/58/Scrum\\_process.svg](https://upload.wikimedia.org/wikipedia/commons/5/58/Scrum_process.svg)

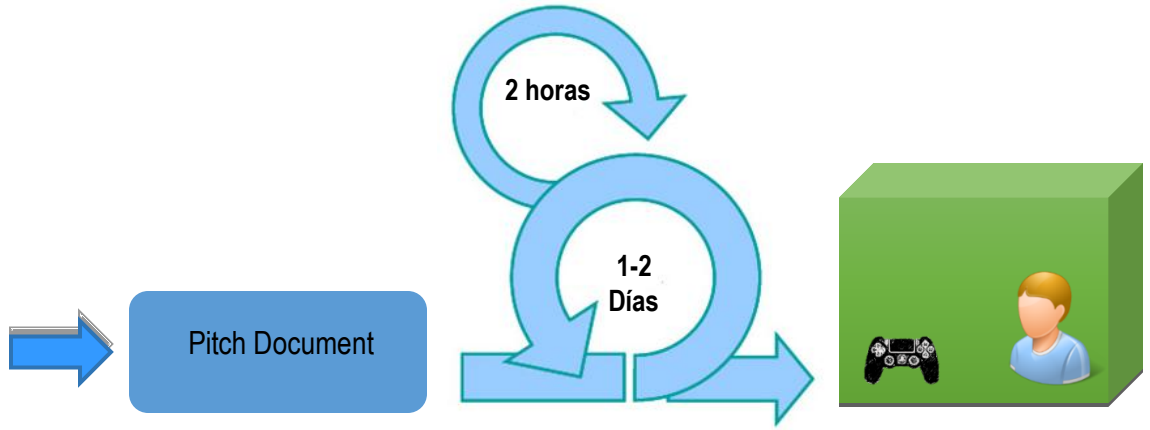
### GráficoNº 1. Gráfico Metodología SCRUM

#### Documentación

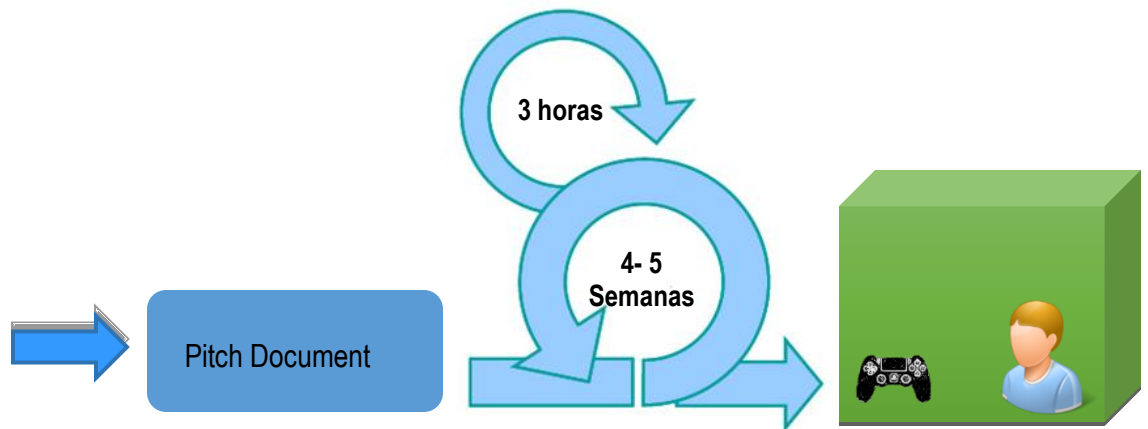
A continuación se describe el funcionamiento de cada paso que necesita el juego para aplicar la Metodología SCRUM.

- **ProductBacklog:** para la realización de tareas, es indispensable especificar detalladamente cada tema y subtema.
- **Sprint Backlog:** el avance de cada tarea se manifiesta en un documento llamado (PitchDocument), este especifica el proceso que va llevando el juego en su desarrollo.
- **Sprint:** el proyecto es elaborado por dos personas las cuales se les asignas horas y días para que en una reunión semanal muestren sus avances y dudas sobre cada Sprint.
- **Software:** para que nuestro software cumpla con los objetivos propuestos que especifica cada tarea, debemos seguir los pasos que nos otorga la metodología. Al final el software será probado e implementado de acuerdo a las necesidades del usuario final.

- + Descargar programa Unity.
- + Instalación del programa.
- + Tutoriales
- + Creación New Unity Project
- + Exportar paquetes adicionales



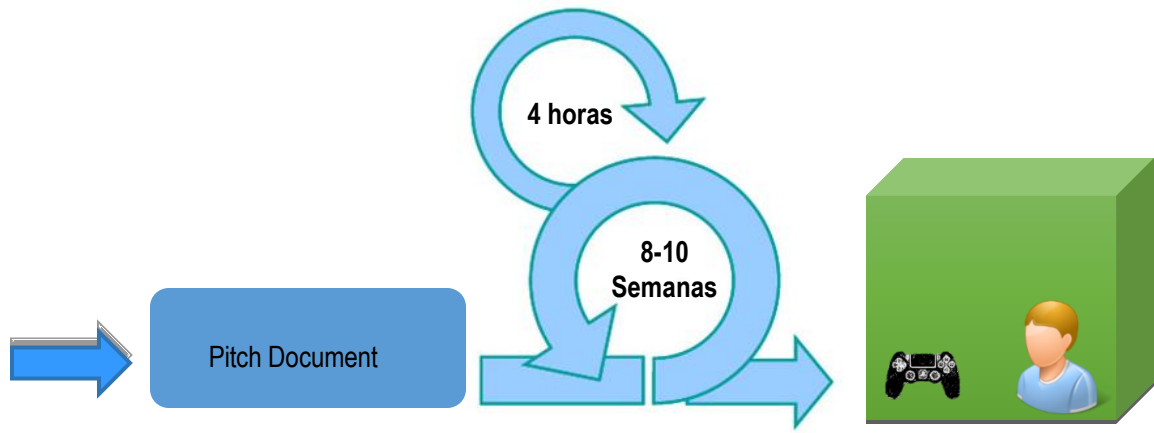
- + Exportar y diseñar la ciudad y ambientación (Asset Store)
- + Generación de colisiones
- + Diseñar iluminación para: el cielo, sol y ciudad
- + Ajuste del escenario agregando: texturas, tamaño y componentes adicionales



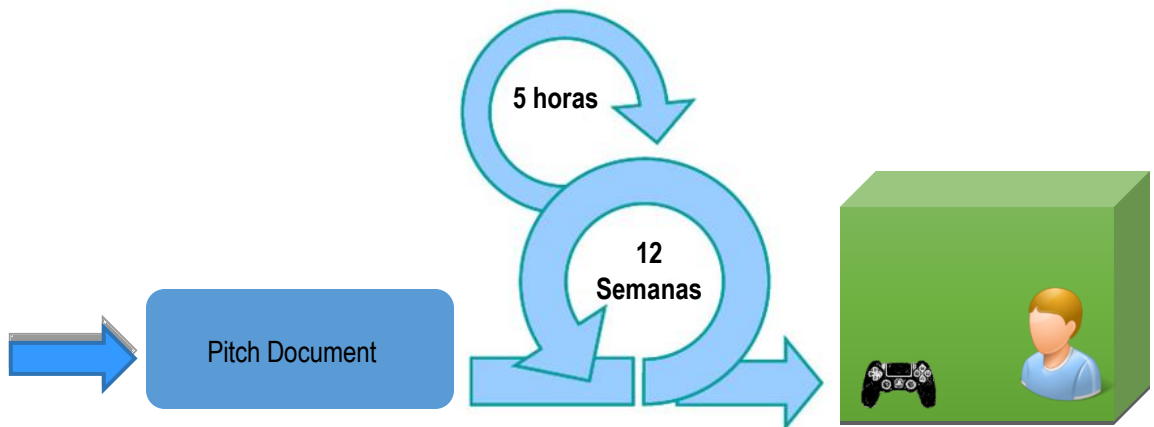
Fuente: Autores

Gráfico N° 2. Implementación Metodología SCRUM

- + Personaje: diseño, colisiones, texturas, posición en el escenario primera persona.
- + Desarrollo en JavaScript:
  - personaje
  - sonidos
  - animaciones
  - ambientación
  - enemigos
  - armas
  - mira
  - inventario (munición, botiquín, contador "balas y vida")



- + Menú:
  - desarrollo en JavaScript, diseño de fuentes, texturas y escenario animado.
  - botones: desarrollo en JavaScript e interacción con el juego.
- Jugar
- Instrucciones
- Cargar partida
- Salir
- escenario de muerte del personaje



Fuente: Autores

Gráfico N° 3. Implementación Metodología SCRUM

### 3. ANÁLISIS Y DISEÑO

Es un método donde se crean un conjunto de modelos utilizando una notación acordada, aplicando técnicas de modelado de objetos para analizar los requerimientos. Las metodologías de análisis y diseño más modernas son casos de uso guiados a través de requerimientos, diseño, implementación, pruebas, y despliegue. El proyecto se trabajará con los siguientes aspectos.

#### 3.1 Definición de Requerimientos

Es una necesidad documentada sobre el contenido, forma o funcionalidad de un producto o servicio. Los requerimientos son declaraciones que identifican atributos, capacidades, características y/o cualidades que necesita cumplir un sistema (o un sistema de software) para que tenga valor y utilidad para el usuario. Muestran qué elementos y funciones son necesarias para un proyecto.

##### *Etapas de la fase de Requerimientos*

- Obtención de requerimientos: búsqueda y obtención de los requerimientos desde los grupos de interés.
- Análisis: comprobación de la consistencia y completitud de los requerimientos.
- Verificación: constatación de que los requerimientos especificados son correctos.

### **3.1.1 Requerimientos Funcionales**

Describe lo que es sistema debe hacer. Estos requerimientos dependen del tipo de software que se desarrolle, de los posibles usuarios del software y del enfoque general tomado por el sistema al solicitar el requerimiento. Los requerimientos de comportamiento se muestran en los casos de uso y por requerimientos del sistema.

Los requerimientos funcionales son:

- El sistema debe captar la atención de su producción mediante una introducción precisa y clara.
- El sistema debe desarrollarse en un motor multiplataforma con herramientas completas para ir perfeccionando el videojuego.
- El sistema debe desarrollarse en ambiente 3D.
- El sistema debe proporcionar ambientación, personajes, audio y efectos para reconstruir el tema del videojuego.
- El sistema debe evidenciar el menú principal para guiar al jugar antes de iniciarlo para tener claridad sobre cada escena que se presentara allí.
- El sistema debe proporcionar un inventario de beneficios para que el jugar se mantenga en el tiempo de juego.
- El sistema debe ofrecer una tienda de mejoras y venta de municiones.

- El sistema debe mostrar el número de municiones y las vidas que tienen durante el tiempo de juego.
- El sistema debe adaptar el cambio de clima según la hora del juego.
- El sistema deberá adquirir dos formas de salir del juego, con una opción configurada en el sistema o muerte del jugador.

### **3.1.2 Requerimientos No Funcionales**

Son aquellos requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de este como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento.

Los requerimientos no funcionales son:

#### Desempeño

- El sistema deberá funcionar y mantenerse en estado activo y de alto funcionamiento acorde con la capacidad del equipo o la velocidad de navegación. Además los tiempos de respuesta serán inmediatos identificando errores o suministrando calidad al software.
- Asegurar el funcionamiento completo del sistema suministrando rendimiento cuando se ejecuta el software, teniendo cuenta las especificaciones técnicas y el estado del equipo.



- El sistema deberá estar disponible para respuestas de alerta, actualizaciones y correctos procesos que se irán generando a medida que se utilice en software.

### Disponibilidad

- Estar disponible al 100% o mantenerse en buen funcionamiento para suministrar calidad y seguridad a los jugadores, el uso es libre y personalizado.

### Confiabilidad

- El sistema debe reconocer los fallos por medio de un sistema transaccional automatizando procesos principales.

### Facilidad de Uso e ingreso de información

- El sistema deberá ser de fácil uso y entrenamiento para demostrar las habilidades que tiene el jugador y desarrollar otras para tener un perfil alto y poder explorar más juegos.
- El sistema debe guardar información automáticamente sin generar restricciones, asimismo si el usuario quiere buscar y visualizar lo que se guardó, automáticamente genera la solicitud pedida sin pedir respuesta.
- El sistema debe dar a conocer manual del sistema y manual del usuario los cuales guían el proceso de desarrollo del software y especifican los detalles que reúne el videojuego para su total funcionamiento.

### Flexibilidad

- El sistema debe ser desarrollado con gráficos en 3D y texturas casi reales para darle vida al personaje sus acciones y métricas del juego son programadas en Java Script y C++.

### Instalación

- El sistema debe facilitar al jugador un ejecutable único del juego para poder instalarlo en su equipo de preferencia y empezar a disfrutar del videojuego. Además el software proporciona manual del sistema y manual de usuario juntos complementados para llevar paso a paso su proceso de instalación y modo de juego.

### Mantenibilidad

- El sistema deberá estar organizado para resolver errores, por ello documentar las líneas de código orientándose con el Manual de Unity donde hay acceso libre, siendo una de las ayudas principales para automatizar errores de programación.
- El sistema debe visualizar escenario principal el cual encabeza acciones primordiales para el funcionamiento del software. Puesto que es el punto de partida para acceder a las escenas que traerán las misiones e historia del juego.
- El sistema debe contar con capacidad para realizar un mantenimiento preventivo y poder en un futuro actualizarlo a nuevas versiones y plataformas de desarrollo.

### Operatividad

- El escenario principal es el acceso directo al videojuego, no presenta restricción alguna y no necesita un administrador para ejecutarse en el equipo.
- El sistema deberá suministrar acceso a los autores del videojuego para verificar y mantener en alto rendimiento el software, mejorar detalles en los personajes y actualizar versión en futuros proyectos.

### Seguridad

- La información estará capturada al momento de jugar y cerrar la partida, dicha información es guardada en el sistema, la cual se almacena automáticamente y está disponible cuando el usuario quiera verla y el administrador restaurarla.
- El sistema poco requiere de procesos extras de seguridad, sin embargo el administrador proporciona automáticamente confirmación de guardado y protegido.

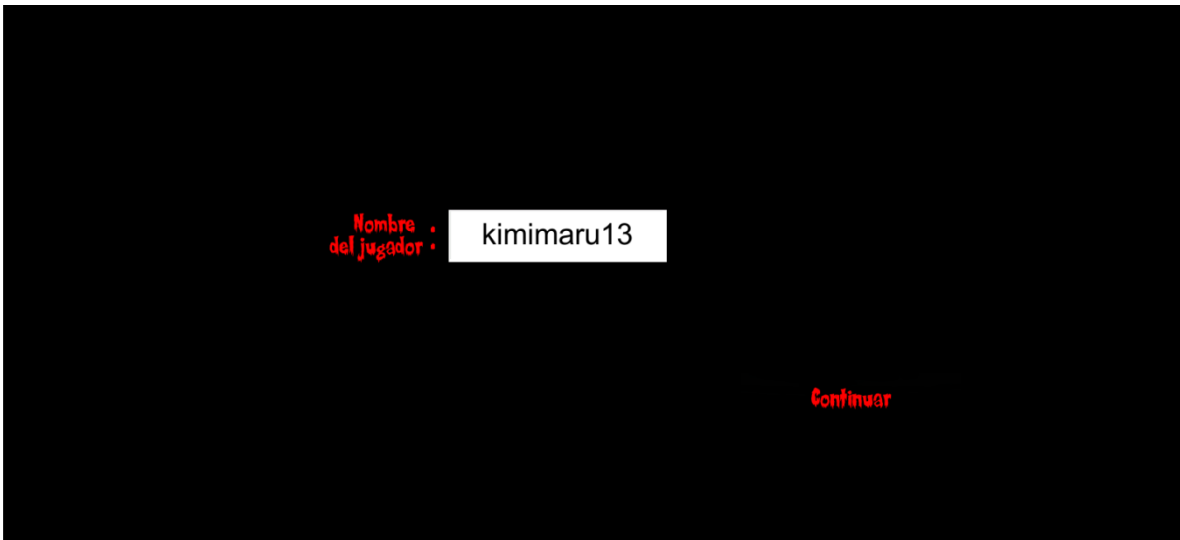
### 3.2 Descripción del sistema propuesto

Nightmares of zombies es un juego en 3D el cual una epidemia zombies es desatada y la lucha por sobrevivir en la ciudad será constante, para mantenerse en el juego hay que realizar diversas acciones, por ejemplo, resguardarse, tener en la mira al objetivo y matar a todos los zombies. Por ello al iniciar el juego se presentara una introducción y más adelante se recreara la misión, siendo el objetivo que tendrá el jugador en la partida que inicie. El videojuego se puede jugar con en ejecutable o compartiéndolo por un navegador.



**Interfaz No. 1. Menú inicio**

**Jugar:** Al presionar el botón “Jugar” nos lleva a la pantalla en la cual debemos ingresar el nombre del jugador, en la parte de abajo se encuentra un botón “Continuar” el cual nos lleva al juego. Nombre de usuario (Interfaz No. 2)



### Interfaz No. 2. Nombre de Usuario

**Cargar Partida:** En la interfaz de cargar partida podemos cargar el juego en donde lo dejamos en nuestra partida anterior. Cargar Partida (Interfaz No. 3)



### Interfaz No. 3. Cargar Partida

**Salir:** Con esta opción podemos salir del juego para así regresar al escritorio del equipo. Salir (Interfaz No. 4)



**Interfaz No. 4. Salir**

### **3.3 Diseño del sistema propuesto**

El diseño de sistemas se ocupa de desarrollar las directrices propuestas durante el análisis en términos de aquella configuración que tenga más posibilidades de satisfacer los objetivos planteados, implementando UML (Lenguaje Unificado de Modelamiento) que muestra la estructura del funcionamiento del juego. Es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes del desarrollo de software. Sirve para representar visualmente las reglas de creación, estructura y comportamiento de relación de objetos y procesos. Esta estructura se emplea en el diseño y diagramación del juego de la siguiente manera.

### ➤ **Diagramas Estáticos**

Los diagramas estáticos o estructurales se encargan de definir: entidades, objetos, áreas, clases, etc. Deben estar definidas dentro del sistema, dando vista estática del sistema. Para explicar la estructura del video juego, utilizaremos el diagrama de clases y diagrama de casos de uso a continuación.

#### ✓ **Diagrama de clases**

Descripción de objetos que comparten características comunes. En el siguiente diagrama se estructura los atributos y operaciones que componen cada clase y la relación que existe entre estos.

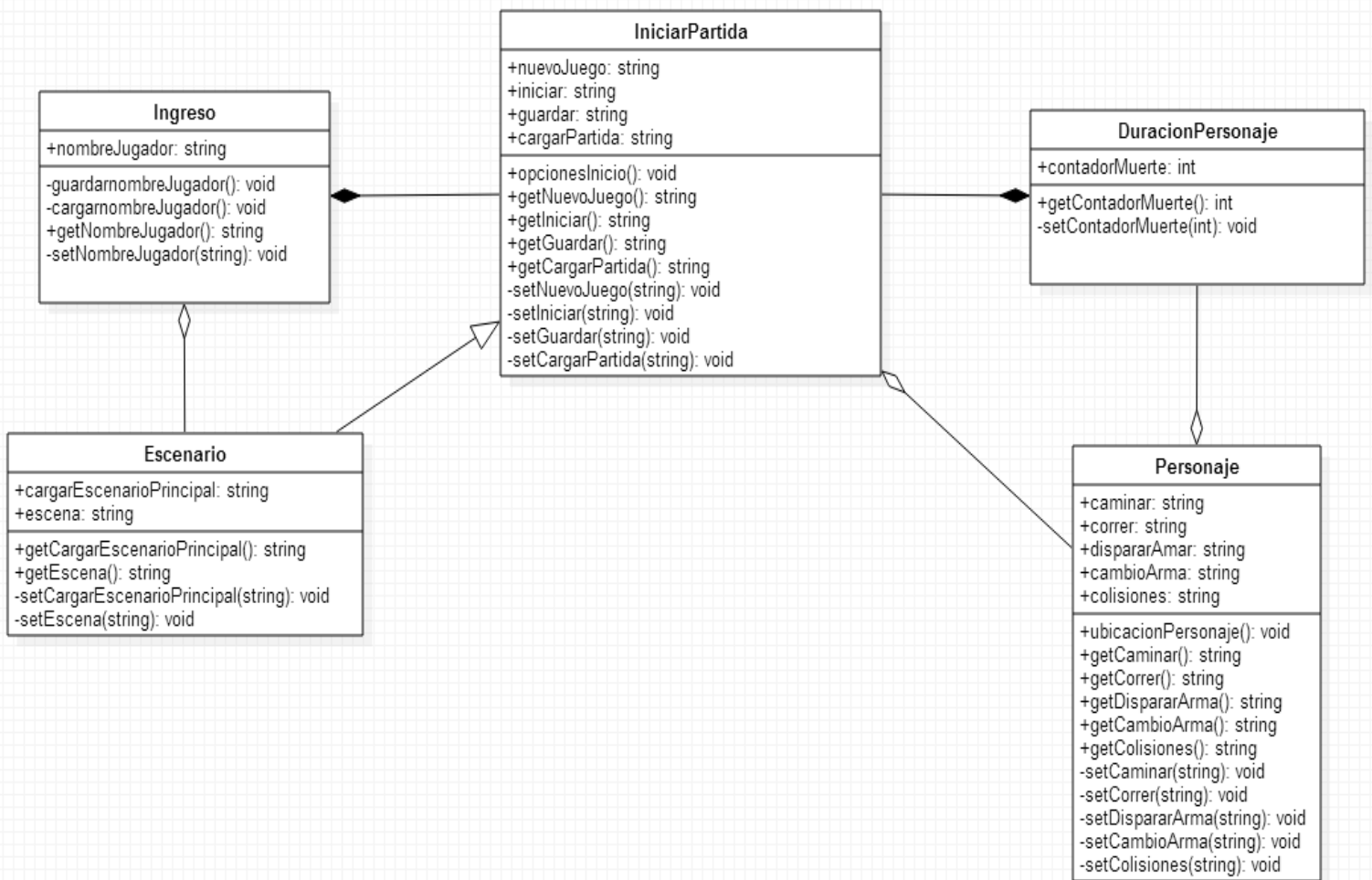


Diagrama Nº 1. Diagrama de clases



✓ **Diagrama de casos de uso**

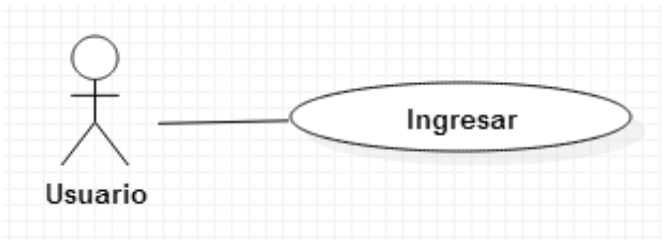
Documentan el comportamiento de un sistema desde el punto de vista del usuario. Mostrando los requerimientos funcionales del sistema, donde estos representan las funciones que un sistema puede ejecutar. Por consiguiente diagramas de casos de uso del juego.

Nombre	Menú
Descripción	El usuario visualiza las opciones para informarse del funcionamiento del videojuego.
Precondiciones	Tener claro la historia, instrucciones y marcación de puntaje para ingresar al juego.
Flujo Normal	Flujo básico dar clic en las opciones para explorar cada parte que compone el videojuego. Registrarse para ingresar.
Flujo Alternativo	Sino no se presenta registro con Nombre del Jugador, no se accede al escenario principal.
Poscondiciones	Clic en Ingresar



Caso de uso N° 1: Menú

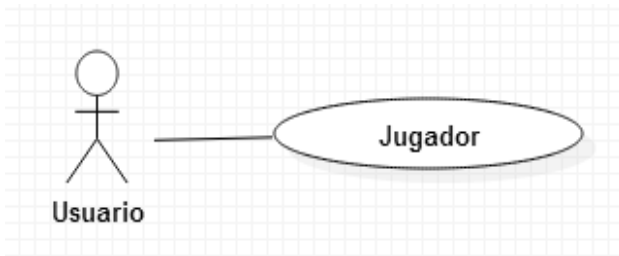
**Tabla N°1 Caso de uso Menú**



Caso de uso N° 2: Ingresar

Nombre	Ingresar
Descripción	El jugador entra al juego escribiendo su nombre o alias.
Precondiciones	El jugador debe tener un nombre único.
Flujo Normal	Flujo básico escribir nombre para identificarse dentro del juego.
Flujo Alternativo	No ingresar nombre el sistema automáticamente no continua en el proceso de registro para empezar una partida.
Poscondiciones	Validación nombre del jugador

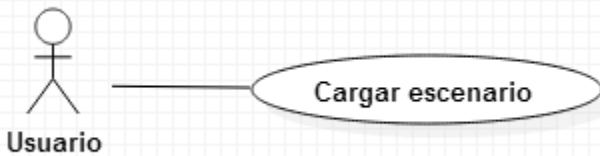
**Tabla N° 2 Caso de uso Ingresar**



Caso de uso N° 3: Jugador

Nombre	Jugador
Descripción	Para tener acceso al videojuego, el usuario necesita tener un nombre o alias para identificarse entre los demás jugadores.
Precondiciones	Teniendo una identificación dentro del juego, se activa todas las armas y vidas que están por toda la ciudad.
Flujo Normal	Flujo básico ingresar con nombre a la interfaz del juego.
Flujo Alternativo	No insertar nombre el registro es vacío.
Poscondiciones	Nivel habilitado para jugar

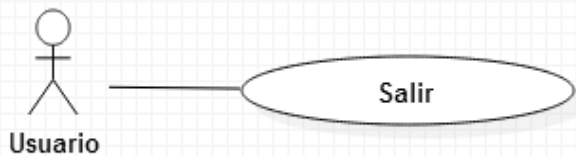
**Tabla N° 3 Caso de uso Jugador**



Caso de uso N° 4 Cargar escenario

Nombre	Cargar escenario
Descripción	Es el escenario principal para que el jugador demuestre sus habilidades para combatir contra los zombies y alcanzar el mayor puntaje dentro el escalafón.
Precondiciones	Validación correcta de registro.
Flujo Normal	Flujo básico manejo de controles que domina el jugador para mantenerse vivo en el nivel.
Flujo Alternativo	No leer instrucciones afecta el juego y la orientación en el escenario.
Poscondiciones	Se finaliza el juego para cargar partida.

**Tabla N° 4 Caso de uso Cargar escenario**



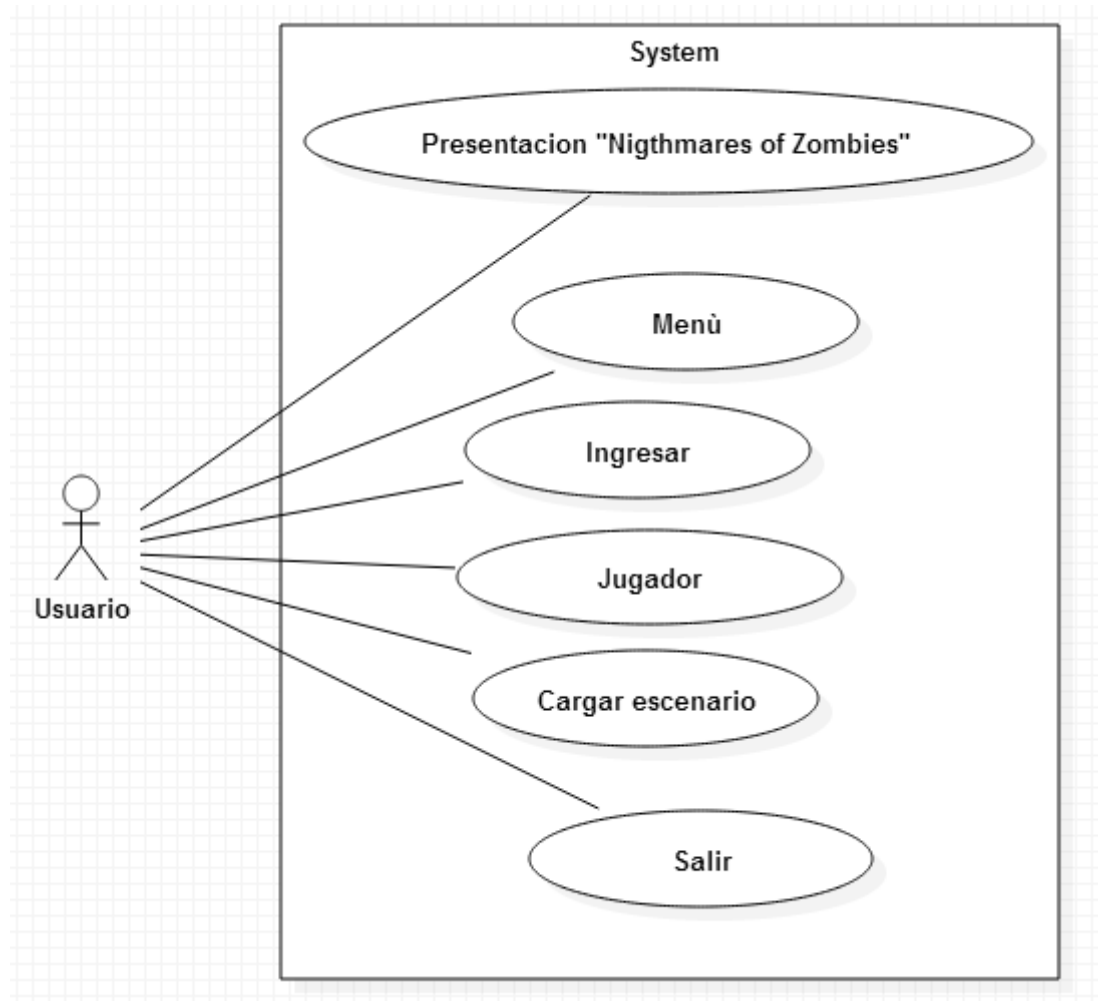
Caso de uso N° 5: Salir

Nombre	Salir
Descripción	Para salir del juego el jugador deberá ganar la partida o perder vidas.
Precondiciones	El usuario registrado se encuentra activo dentro del juego.
Flujo Normal	Flujo básico el jugador está activo dentro de la partida y tiene diferentes ayudas para durar y poder ganar.
Flujo Alternativo	El videojuego muestra por un mensaje el cierre del juego o el puntaje de la partida.
Poscondiciones	Finalización del juego correctamente.

**Tabla N° 5 Caso de uso Salir**

✓ **Diagrama de caso de uso del usuario**

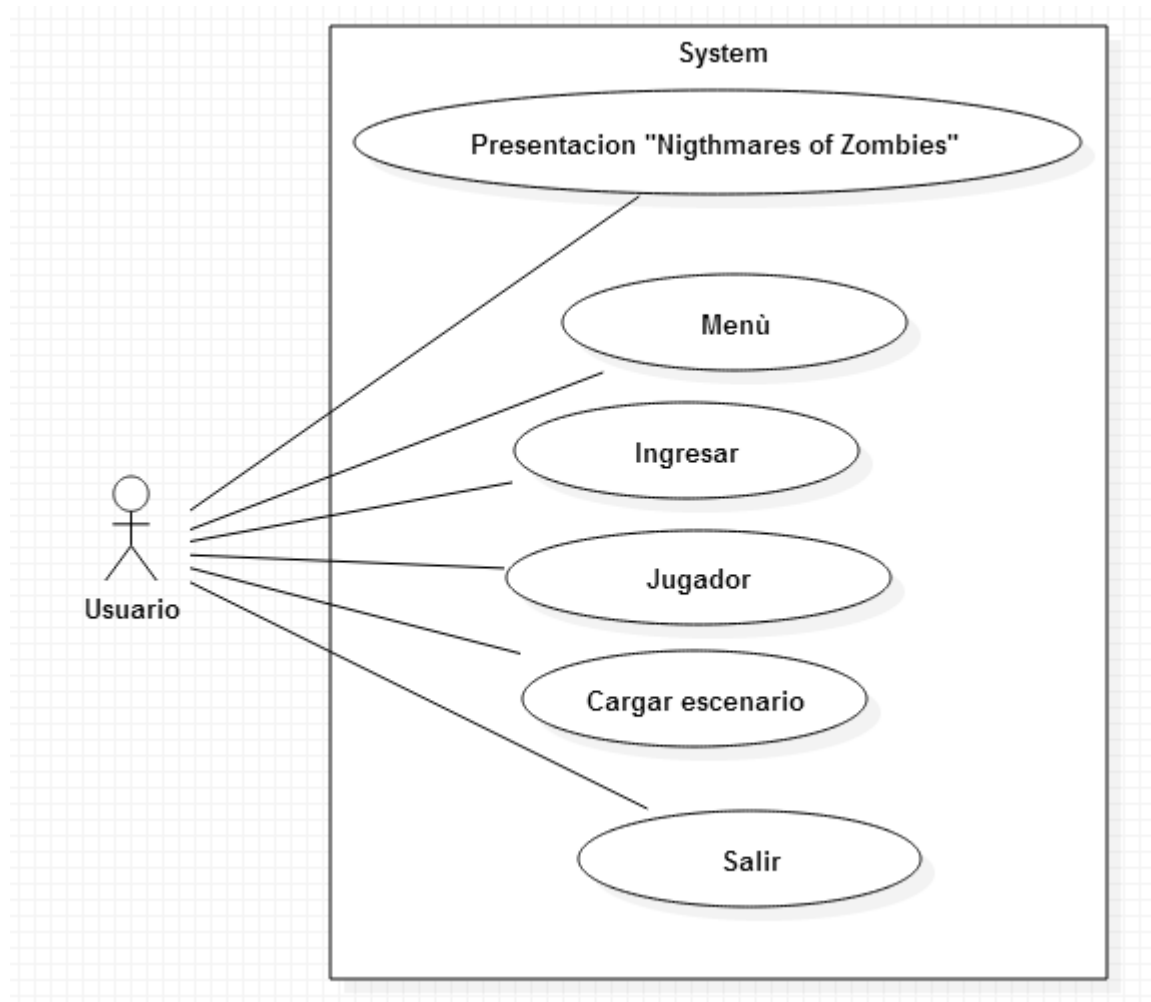
Se presenta el diagrama general para el usuario especificando los procesos que deberá realizar el sistema en el momento de la ejecución del mismo.



**Diagrama Nº 2. Diagrama caso de uso del usuario**

✓ **Diagrama de caso de uso general del sistema**

Se presenta el diagrama que conforma casos de uso primordiales que realiza el actor formando la estructura del sistema.



**Diagrama N° 3. Diagrama de caso de uso general**

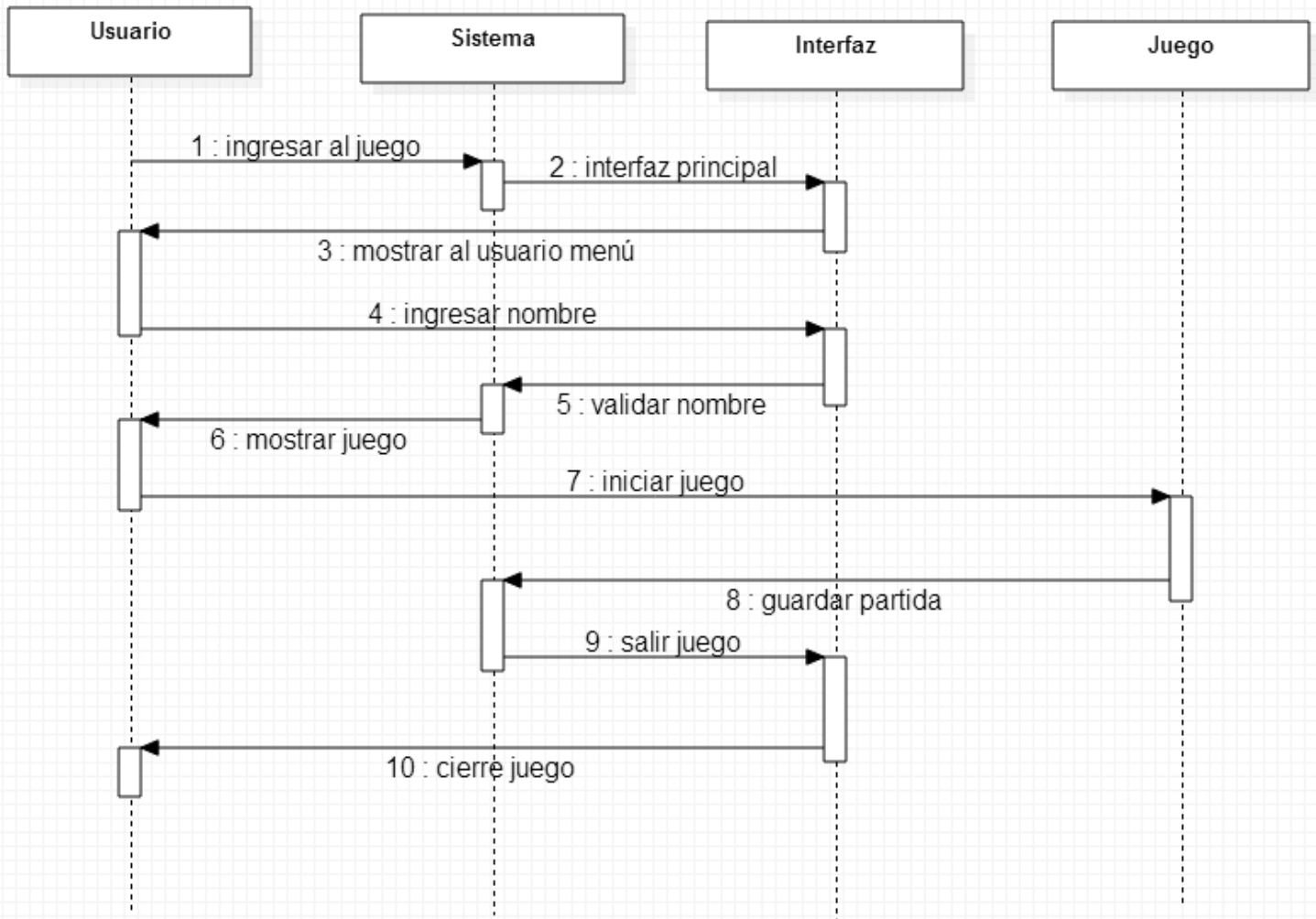
## ➤ **Diagramas Dinámicos**

Muestra el comportamiento del sistema, permitiendo identificar los objetos necesarios que trabajan entre sí a través de métodos y mensajes. Su estructura es un árbol que facilita el análisis y el resumen de datos. Dentro de los diagramas dinámicos encontramos: diagrama de estados, diagrama de actividad y diagrama de interacción. Para mostrar las interacciones entre objetos en un escenario se muestra por medio del diagrama de interacción o secuencia.

### ✓ **Diagrama de interacción/ secuencia**

Este diagrama muestran los diferentes objetos y las relaciones que tiene entre ellos, pueden pasar la información uno de otro sin perderla. Contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario y mensajes intercambiados del sistema. Representando un conjunto de roles en una interacción. Tiene dos dimensiones: tiempo (generalmente vertical) y diferentes instancias (generalmente horizontal) sin importar el orden de los sucesos.

Para el desarrollo del software mediante secuencias de tiempos y acciones, el siguiente diagrama especifica cada proceso que compone el sistema.



**Diagrama Nº 4. Diagrama de secuencia**

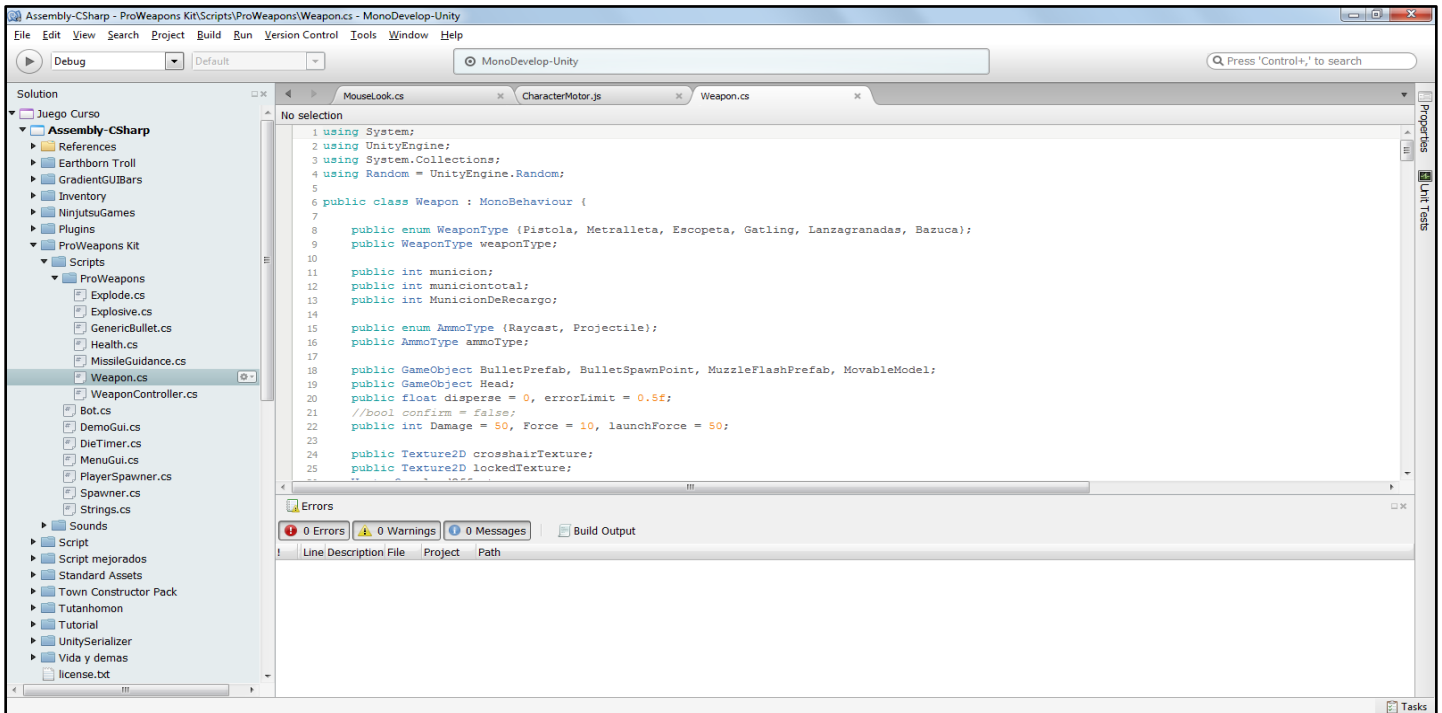
## **4. DESARROLLO**

Para el desarrollo del videojuego se tiene en cuenta dos aspectos fundamentales para el funcionamiento, hardware y software. Los cuales sirven para la ejecución del video juego y la optimización del entorno gráfico. Además ampliamos los parámetros para que en distintos sistemas operativos y navegadores puedan tener un rendimiento óptimo. Por ello es indispensable conocer las especificaciones técnicas para ejecutar el juego.

### **4.1 Especificaciones técnicas**

Describen detalladamente las características de un proyecto, organizándolas en una ficha técnica. En ella se recogen datos claves en forma clara concisa, para la ejecución de este juego no es necesario gastar muchos recursos del equipo y la ejecución es muy sencilla. Se presenta a continuación el proceso de instalación y ejecución del sistema.





Script de las armas: permite disparar, configurar la mira y nos muestra el contador de balas. Lenguaje C++ y JavaScript.

#### 4.1.1 Software

Para mayor efectividad del juego, los sistemas operativos proveen una interfaz cómoda y completa siendo intermediario para que el usuario personalice los programas de su interés y puedan ejecutarlos o abrirlos enlazados con el hardware del sistema. Existen varios tipos de sistemas operativos que han estado actualizándolos para mayor efectividad y comodidad. Estos son los sistemas operativos apropiados para que el jugador pueda demostrar sus habilidades y el juego se ejecute a la perfección:

- ✓ Windows ( XP, Vista, 7 y 8)
- ✓ Mac ( Todas las versiones)
- ✓ Linux

Para los anteriores sistemas operativos el tipo de sistema es 32 bits y 64 bits.

El juego también se puede ejecutar en diversos navegadores con mucha facilidad y con poco internet ya que el juego se encuentra comprimido, esto lo hace muy liviano para el navegador, este suministra al usuario un plugin de instalación el cual se ejecuta con tan solo un clic para iniciar la instalación, inmediatamente direcciona a la pantalla de juego. Navegadores:

- ✓ Google Chrome
- ✓ Internet Explorer
- ✓ FireFox
- ✓ Opera
- ✓ Safari

#### **4.1.2 Hardware**

Conjunto de componentes que integran la parte material de una computadora. El hardware se caracteriza por ser el material que se emplea para que un ordenador o cualquier aparato electrónico puedan funcionar y ejecutar las tareas para las que han sido diseñados. Se clasifica en dos categorías básico y complementario, para el videojuego nos enfocaremos en lo básico siendo componentes indispensables para la funcionalidad de un computador. Los requerimientos mínimos para ejecutar el juego son:

- ✓ Banda ancha 1 Mega en adelante
- ✓ Disco Duro 150 MB o más
- ✓ Memoria RAM 1 Giga
- ✓ Procesador Core2Duo

## 4. GLOSARIO

**Unity:** Es un motor gráfico multiplataforma que está disponible como plataforma de desarrollo para Microsoft Windows y OS X, y permite crear juegos para Windows, OS X, Linux, Xbox 360, PlayStation 3, PlayStation Vita, Wii, Wii U, iPad, iPhone, Android y Windows Phone.

**UML:** (Lenguaje Unificado de Modelo) Es un lenguaje para especificar, construir, visualizar y documentar los artefactos de un sistema de software orientado a objetos (OO). Un artefacto es una información que es utilizada o producida mediante un proceso de desarrollo de software.

**Motor Físico:** Software capaz de realizar simulaciones de ciertos sistemas físicos como la dinámica del cuerpo rígido, el movimiento de un fluido y la elasticidad.

**Motor Gráfico:** La funcionalidad básica de un motor es proveer al videojuego de un motor de renderizado para los gráficos 2D y 3D, con detector de colisiones, sonidos, scripting, animación, inteligencia artificial, redes, streaming, administración de memoria y un escenario gráfico.

**Java Script:** Está orientado a objetos, basado en prototipos, imperativo, débilmente dinámico. Se utiliza principalmente como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web.

**3D:** Objeto tridimensional con dimensiones de anchura, longitud y profundidad. Imágenes de arte gráfico que son creados con ayuda de software y programas especiales

**Lenguaje de programación:** Describe el conjunto de acciones consecutivas que un equipo debe ejecutar. Reglas sintéticas y semánticas que expresan instrucciones que serán interpretadas por un usuario.

**Navegadores:** Permite al acceso a Internet, interpretando la información de archivos y sitios web para que pueden ser leídos.

**Interacción:** Representa la forma en cómo un Cliente (Actor) u Objetos (Clases) se comunican entre sí en petición a un evento. Esto implica recorrer toda la secuencia en donde se obtienen las responsabilidades claramente.

**Modelos de Desarrollo:** Se conocen como ciclos de vida del desarrollo del software, incluyendo el proceso de construir, entregar y hacer evolucionar el software, están divididos es: modelo en cascada o clásico, modelo de prototipos, modelo en espiral, modelo de desarrollo por etapas, modelo estructurado, modelo orientado a objetos, etc.

**Interfaz de Usuario:** Medio por el cual el usuario puede comunicarse con el equipo, incluyendo elementos como menús, ventanas, ratón, etc. Además controla las funciones del equipo, intercambio de datos entre aplicaciones y control de acceso.

## 6. CONCLUSIONES

- Nigthmares of Zombies es un software que suministra rendimiento y facilidad para usarlo en un equipo o navegador predeterminado, buscando entretener a los usuarios y destacando sus habilidades al jugar.
- El motor gráfico Unity aporta lo necesario para desarrollar un videojuego básico y poderlo ejecutar en plataformas o en navegadores. Además existen otros programas para diseñar y animar personajes en 3D.
- Aprendemos de una forma rápida y eficaz a desarrollar videojuegos con diversos lenguajes de programación, nutriendo el mercado Colombiano en el área de videojuegos.

## 7. BIBLIOGRAFÍA

- Catalina Gutiérrez. (2014) ¿Para qué sirve el Scrum en la Metodología Ágil? Intelligences to Bussiness. Recuperado de: <http://www.i2btech.com/blog-i2b/tech-deployment/para-que-sirve-el-scrum-en-la-metogologia-agil/>
- Sommerville. (2005).E 7° Editorial Pearson Educación S.A Madrid. *Ingeniería del software* Recuperado de: [http://books.google.com.co/books?id=gQWd49zSut4C&pg=PA118&lpg=PA118&dq=que+es+requerimiento+de+un+sistema&source=bl&ots=s605vnxvd&sig=4TvMcwULM\\_BtaAy5B1QNEgeeSRo&hl=es&sa=X&ei=yk1RVO6pE8WogwTPHYPoAQ&ved=0CEoQ6AEwBzgK#v=onepage&q=que%20es%20requerimiento%20de%20un%20sistema&f=false](http://books.google.com.co/books?id=gQWd49zSut4C&pg=PA118&lpg=PA118&dq=que+es+requerimiento+de+un+sistema&source=bl&ots=s605vnxvd&sig=4TvMcwULM_BtaAy5B1QNEgeeSRo&hl=es&sa=X&ei=yk1RVO6pE8WogwTPHYPoAQ&ved=0CEoQ6AEwBzgK#v=onepage&q=que%20es%20requerimiento%20de%20un%20sistema&f=false)
- Jesús Cáceres Telio. Universidad de Alcalá. Diagramas de Casos de uso. Recuperado de: <http://www2.uah.es/jcaceres/capsulas/DiagramaCasosDeUso.pdf>
- UML tema 4. (2008). UML Lenguaje Unificado de Modelado. Recuperado de: [http://astreo.ii.uam.es/~jlara/TACCII/5\\_UML\\_rev1.pdf](http://astreo.ii.uam.es/~jlara/TACCII/5_UML_rev1.pdf)
- EISC. Casos de uso reales. Recuperado de: [http://eisc.univalle.edu.co/materias/Material\\_Desarrollo\\_Software/DISDIAG\\_COLAB-A8.pdf](http://eisc.univalle.edu.co/materias/Material_Desarrollo_Software/DISDIAG_COLAB-A8.pdf)

- Programación Avanzada. Requerimientos de Software. Recuperado de: [https://eva.fing.edu.uy/pluginfile.php/58481/mod\\_resource/content/1/pavan-teorico04-requerimientos.pdf](https://eva.fing.edu.uy/pluginfile.php/58481/mod_resource/content/1/pavan-teorico04-requerimientos.pdf)
- Damián Gutiérrez. (2011). UML Diagramas de Clases. Universidad de los Andes Venezuela. Recuperado de: [http://www.codecompiling.net/files/slides/UML\\_clase\\_04\\_UML\\_clases.pdf](http://www.codecompiling.net/files/slides/UML_clase_04_UML_clases.pdf)
- EISC. Casos de uso reales. Recuperado de: [http://eisc.univalle.edu.co/materias/Material\\_Desarrollo\\_Software/DISDIAG COLAB-A8.pdf](http://eisc.univalle.edu.co/materias/Material_Desarrollo_Software/DISDIAG_COLAB-A8.pdf)
- David Vallejo Fernández, Cleto Martin Angelina. Edición electrónica. *Desarrollo de Videojuegos Arquitectura del Motor de Videojuegos*. Recuperado de: <https://books.google.com.co/books?id=ylnABAAAQBAJ&pg=PA157&dq=desarrollo+de+videojuegos&hl=es&sa=X&ei=qku5VPXRLY3hggT8toLIDw&ved=0CCMQ6AEwAA#v=onepage&q=desarrollo%20de%20videojuegos&f=false>

#### *Programas*

- Mixamo. Personajes en 3D con animación. Recuperado de: <https://www.mixamo.com/>
- Unity. Motor gráfico. Recuperado de: <http://unity3d.com/es>
- Asset Store Unity. Recuperado de: <https://unity3d.com/es/learn/tutorials/modules/beginner/asset-store>

# MANUALES





**VIDEOJUEGO  
NIGHTMARES OF ZOMBIES  
MANUAL DEL SISTEMA**

**JHOAN SEBASTIAN CHAVEZ QUIROGA  
CAROLINA ROZO ARIAS**

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

## TABLA DE CONTENIDO

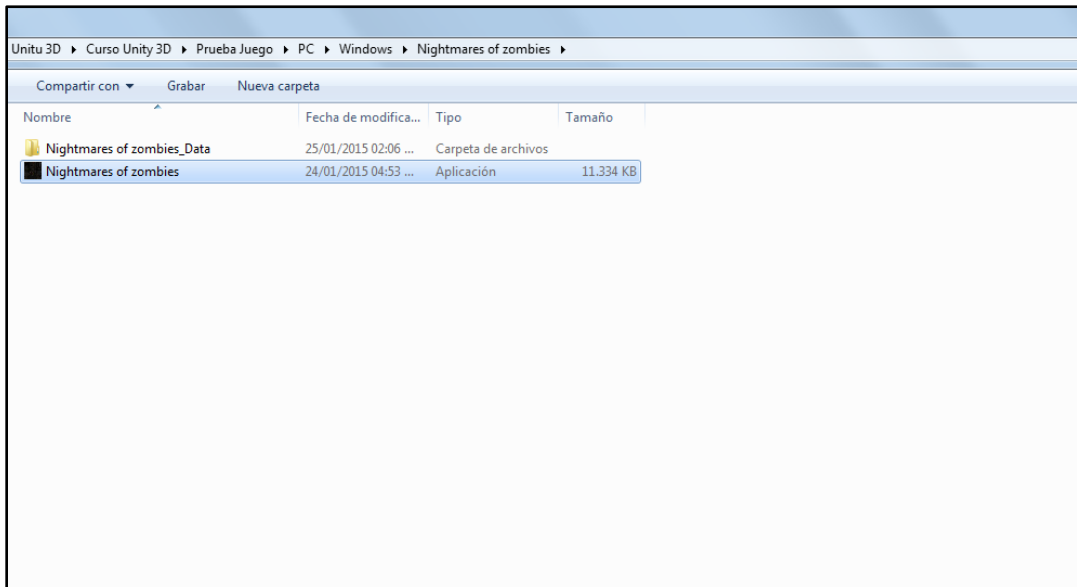
	<b>Pág.</b>
<b>8. Manuales</b>	
8.1 Manual del sistema	
8.1.1 Instalación.....	1
8.1.2 Interfaz.....	2
8.1.3 Opciones.....	3
8.1.4 Sonidos.....	6
8.1.5 Efectos.....	7

## 8.1 MANUEL DEL SISTEMA

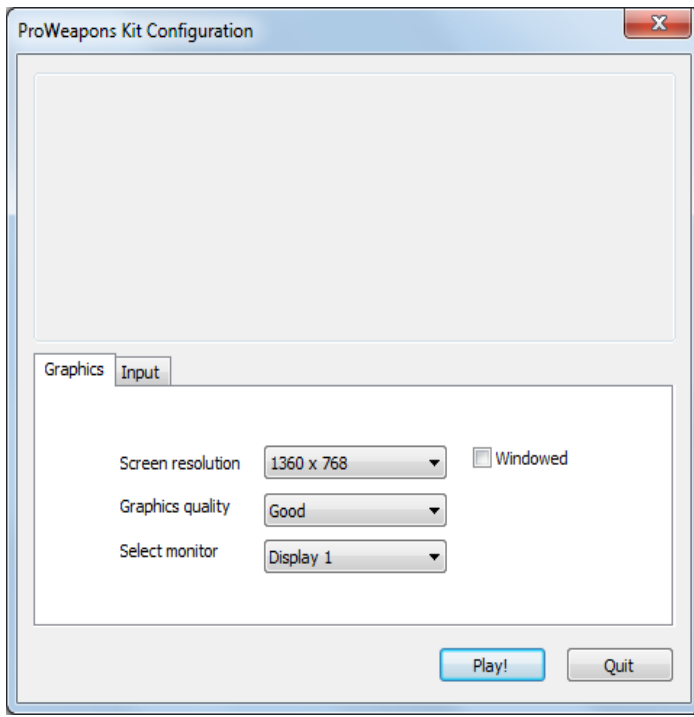
Este documento se elabora para ver lo que conlleva la creación del juegos, los botones, imágenes, sonidos y opciones que este nos proporciona, además sirve de guía o comparación con otros juegos en el mercado, en este documento podemos apreciar las opciones que nos brinda el juego, armas, niveles y demás.

### 8.1.1 Instalación

Nightmares of Zombies está disponible al jugador en un ejecutable preciso y fácil, solo debe seguir dos pasos que se muestran a continuación.



Encontramos 2 archivos le damos doble clic al segundo o ejecutable



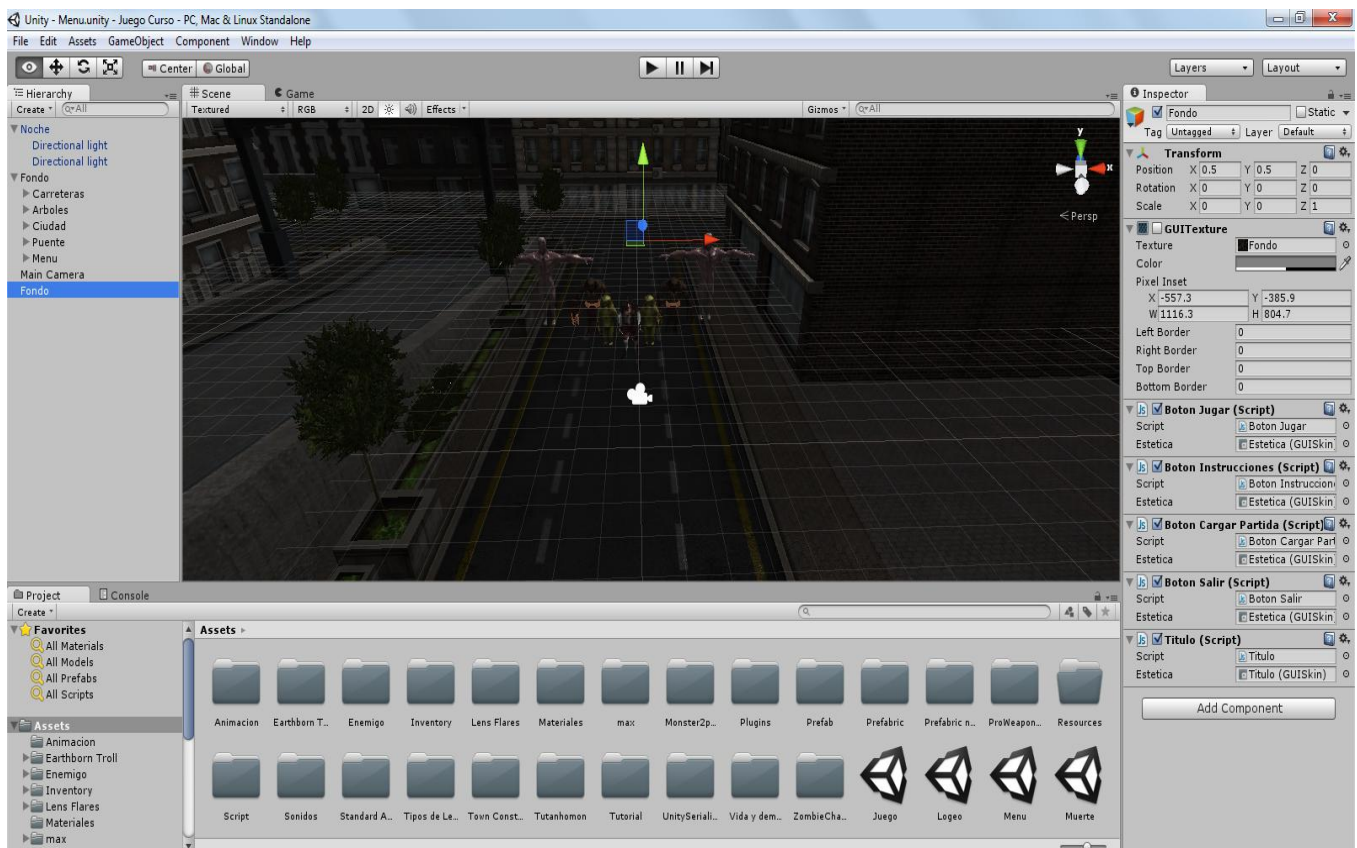
Nos aparece la siguiente pantalla, en la primera opción “Screen resolution” podemos escoger la resolución, la segunda nos permite escoger si los personajes se van a ver nítidos u opacos, la tercera en que pantalla se va a visualizar y el recuadro de “Windowed” es para saber si la pantalla se reproduce completa o no.

### 8.1.2 Interfaz

En este espacio podemos visualizar las opciones graficas del juego, su resolución, texturas, en qué tipo de ambiente es desarrollado, como se realizó el menú, los personajes, la ciudad y demás, este espacio también cuenta con la resolución en que se ejecutara el video juego además nos muestra toda la interfaz y sus opciones de una manera más específica.

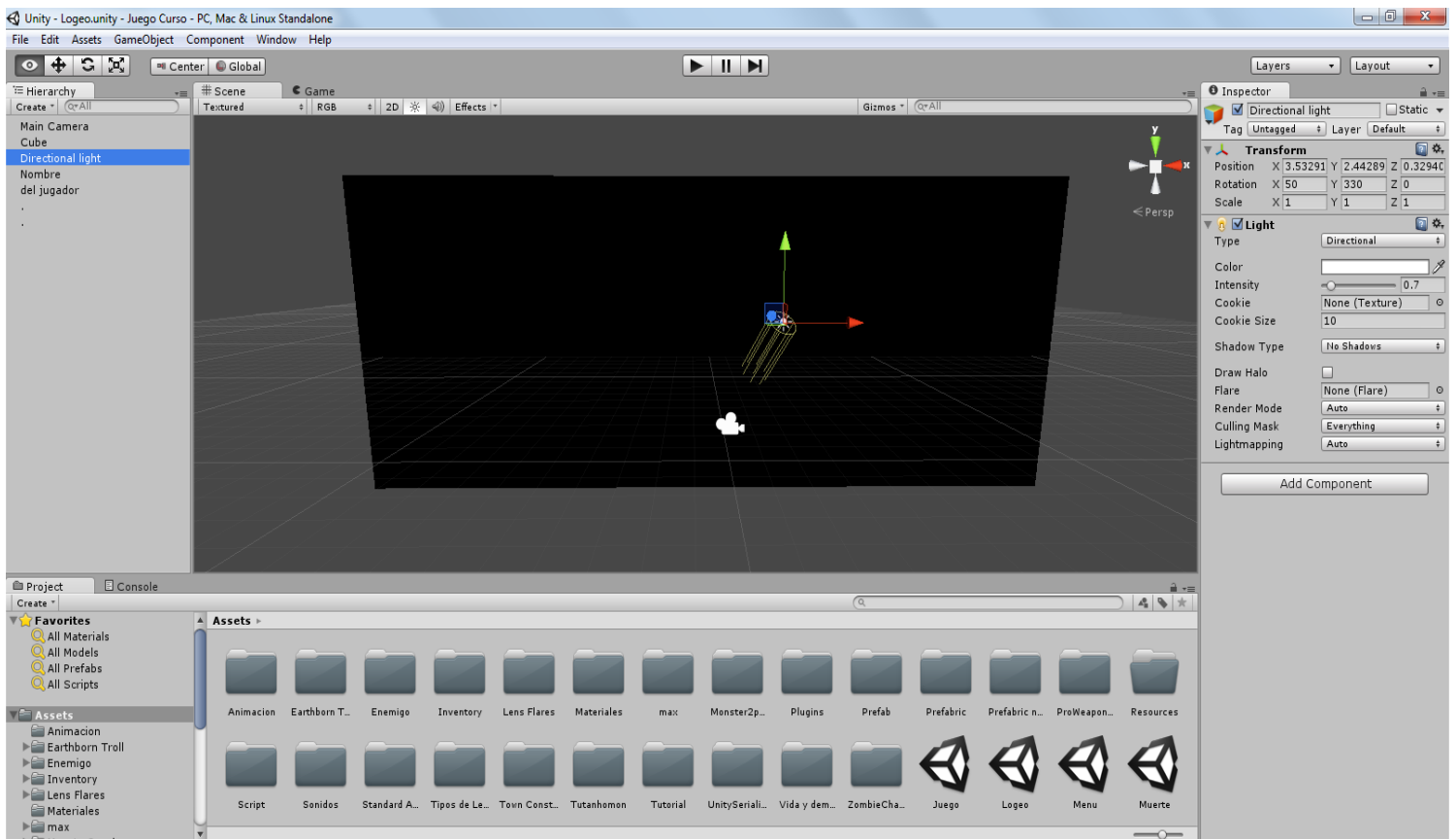
### 8.1.3 Opciones

En esta pantalla visualizamos como se ve en el motor de video juegos Unity 3D, en el centro vemos la ambientacion que se utiliza en el menu, en la parte derecha los Script de los botones los cuales se visualizan unicamente cuando el juego se ejecuta, en la parte izquierda los objetos que estamos utilizando en esta escena, el menu esta compuesto de 5 Scripts el primero es Jugar el cual no lleva a la pantalla de ingresar nombre del jugador, en la segunda que es instrucciones visualizamos los botones o comandos que necesitamos para manejar el jugados, en la tercera podemos cargar las partidas guardadas del juego, el cuarto boton espara salirnosdel juego y al quinta es el titulo deljuego que se visualiza en el menu.



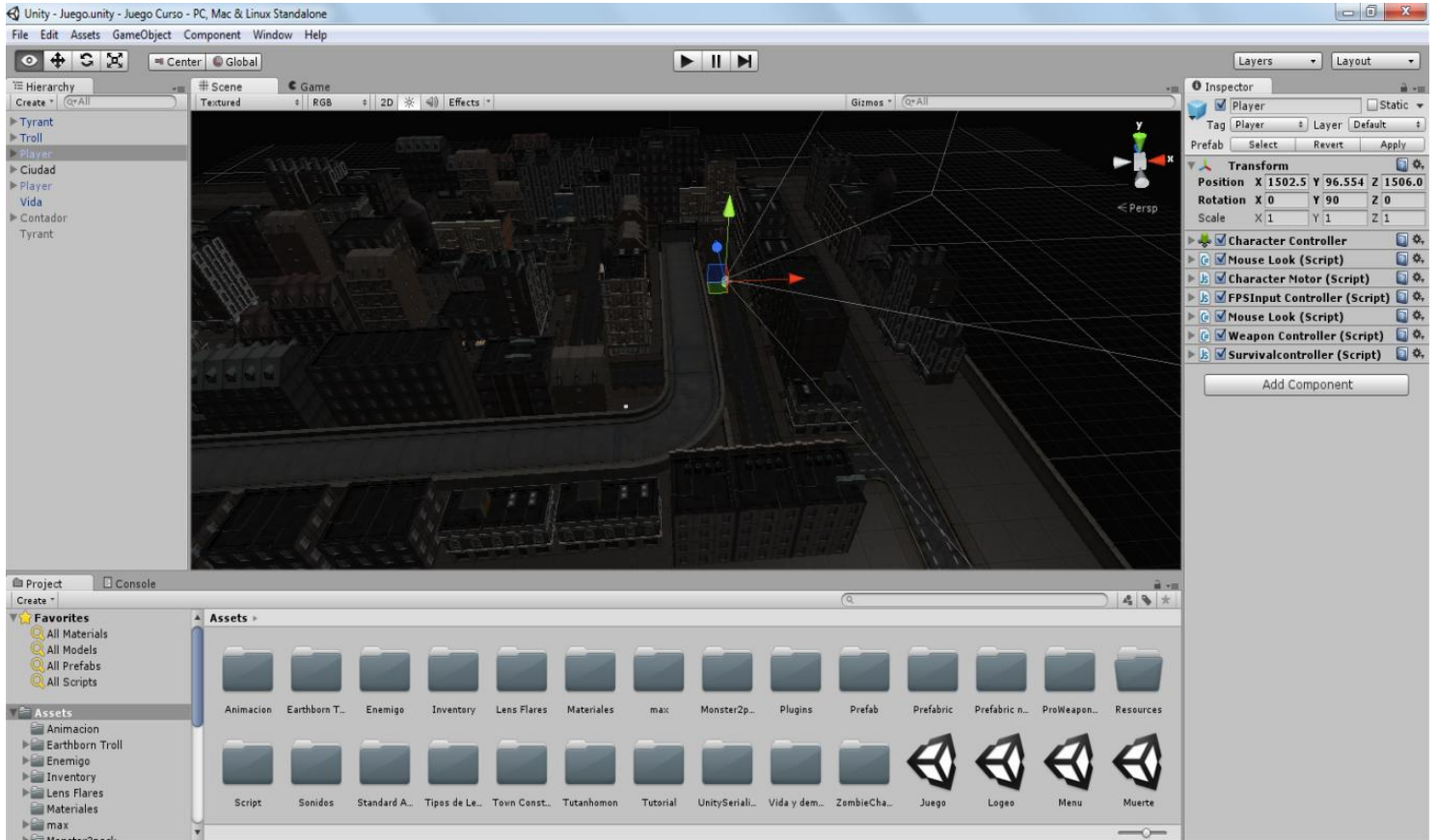
Interfaz No. 5: Funciones primer escenario

En esta ventana ingresamos el nombre del usuario, la cual esta compuesta por un plano, un material negro, iluminacion y Scripts los cuales muestran el mensaje y el boton “Continuar” el cual no lleva a el juego.



## Interfaz No. 6: Nombre del jugador

En esta opcion visualizamos la ciudad con sus componentes, materiales, luz y demas en la parte izquierda, en la parte derecha los Scripts del personaje los cuales generan un motor de fisica para el muñeco, en el cual se integra la gravedad, movimiento del muñeco, armas y demas, ademas visualizamos en que posicion esta apuntando el jugador.



**Interfaz No. 7: Interfaz ciudad**



#### 8.1.4 Sonido

El sonido es muy importante en el juego ya que con el definimos y apoyamos la ambientación, por eso las diversas escenas y ambientaciones tienen un sonido diferente, con esto buscamos hacer más realista la experiencia del jugador a la hora de jugar, principalmente el menú tiene un sonido de suspenso y conforma va pasando la historia además de las escenas el sonido se va adaptando al juego, en el cual buscamos implementar un sonido de terror en las escenas más oscuras para que su impacto fuera mayor. En este aspecto también buscamos destacar los efectos, de las armas, zombies de una forma que se adecuaran con el ambiente y el sonido principal, para así dar mayor contraste y realismo a la escena.

Los sonidos se reparten en:

- Menú
- Nombre del jugador
- Instrucciones
- Cargar Partida
- Escena de día
- Escena de noche
- Armas
- Zombies (Cada zombie tiene un sonido diferente)
- Inventario
- Colisión de la Munición.

### 8.1.5 Efectos

Cuando se va a desarrollar un video juego se tienen que tener varios aspectos importantes entre los cuales se encuentra este ya que los efectos visuales del juego ayudan que el juego sea más realista o no, en este caso nos enfocamos por crear efectos en las municiones cuando impacten con un objeto dejen rastro en caso de la balas dejan un agujero y según el arma esto varia, en las municiones como bazuca nos enfocamos que el efecto fuera de una explosión la cual deja humo, así fuimos implementando efectos en todo el juego para que la sensación del jugador fuera verdadera.



**Interfaz No. 8: Efectos**

**VIDEOJUEGO  
NIGHTMARES OF ZOMBIES  
MANUAL DEL USUARIO**

**JHOAN SEBASTIAN CHAVEZ QUIROGA  
CAROLINA ROZO ARIAS**

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

## TABLA DE CONTENIDO

Pág.

### 8. Manuales

#### 8.2 Manual del usuario

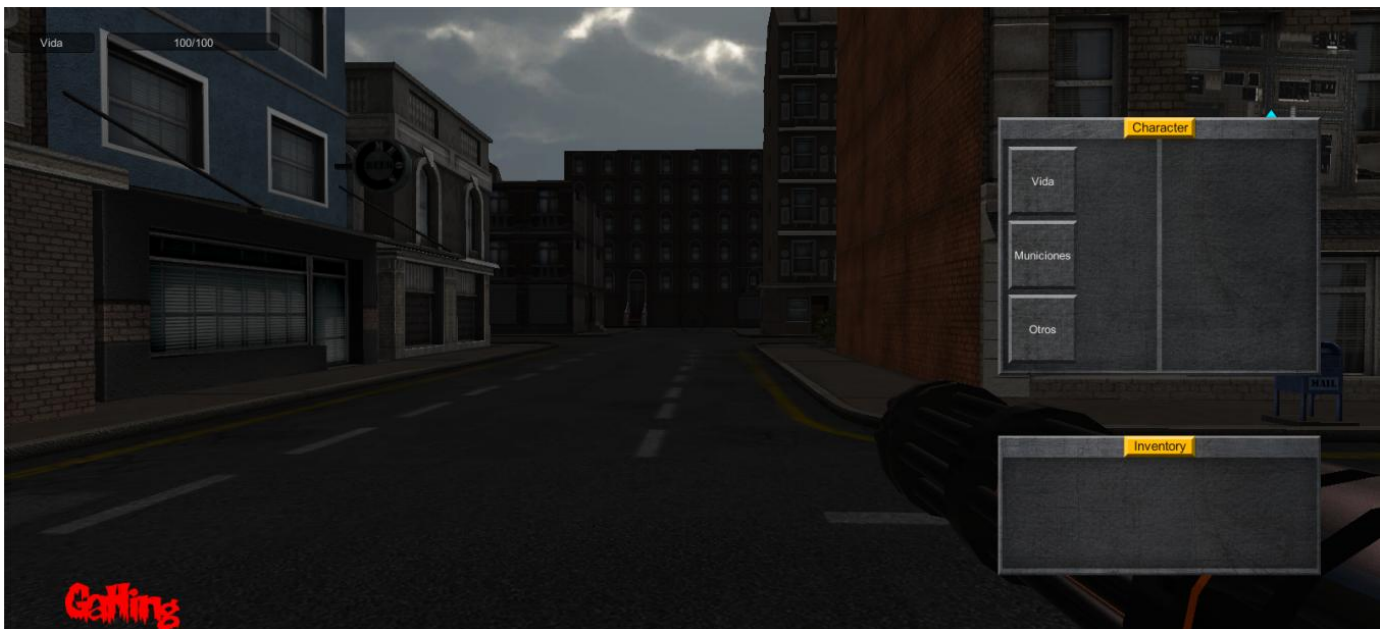
8.2.1 Inventario.....	1
8.2.2 Escenario.....	2
8.2.3 Armas.....	3
8.2.4 Interfaz del jugador.....	4
8.2.5 Menú pausa.....	4
8.2.6 Mapa.....	6
8.2.7 Guardar partida .....	7
8.2.8 Generó.....	9
8.2.9 Controles.....	9

## 8.2 MANUAL DEL USUARIO

En este manual vamos a definir los aspectos a nivel de juego, ya que en este espacio mostramos el inventario, teclas para moverse, escenarios, armas además hablar sobre la historia.

### 8.2.1 Inventario

En esta parte mostramos el inventario el cual se muestra con la tecla (I) en el juego, en el cual podemos almacenar botiquines los cuales nos proporcionan vida y municiones las cuales nos servirán para las diversas armas que ofrecemos.



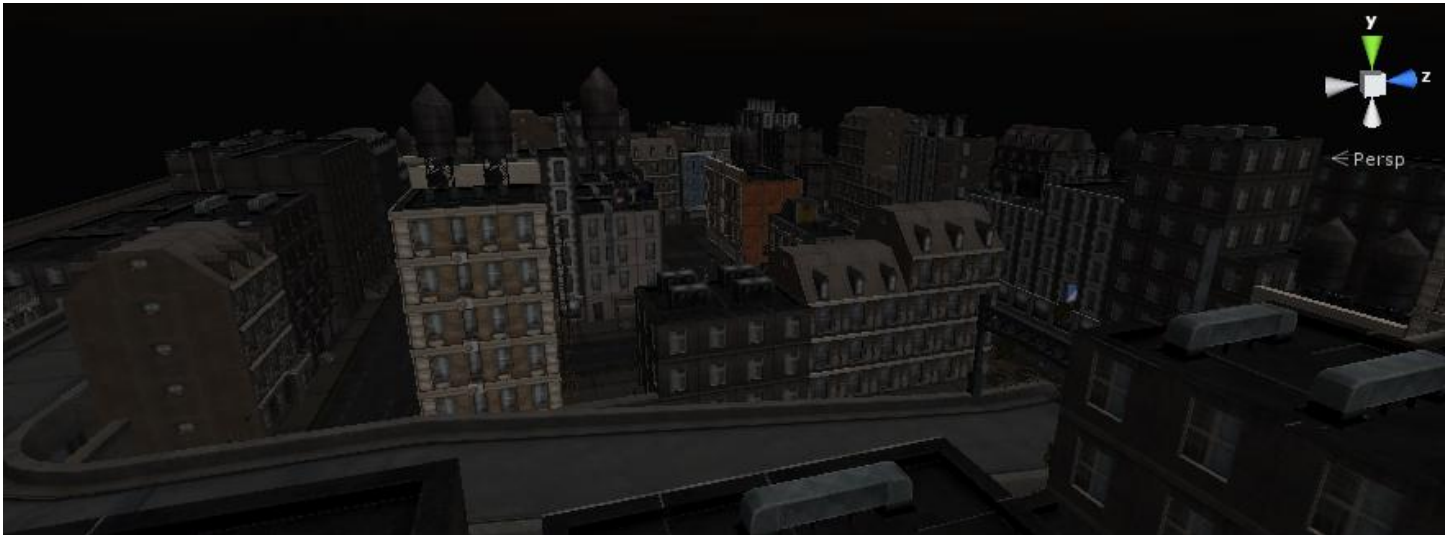
Interfaz No. 9: Inventario



### **Interfaz No. 10: Inventario**

#### **8.2.2 Escenarios**

En el escenario podemos apreciar que es una ciudad en el cual se desarrollara el video juego, está compuesto de día y de noche en el cual se van a desatar diversos peligros además se contara la historia de cómo comenzó el virus desde el primer día.



## Interfaz No. 11: Escenario

### 8.2.3 Armas

En este aspecto tratamos de ampliar las opciones para el jugador, las armas son:

- Pistola
- Metralleta
- Escopeta
- Bazuca
- Bazuca rastreadora

Las cuales nos van a servir en esta aventura ya que cada una cuenta con cierto nivel de daño y para conseguirlas el jugador deberá sobrepasar las adversidades en cada escenario.

### 8.2.4 Interfaz de Jugador

La pantalla cuando la persona comience a jugar tendrá el nombre de las armas en la parte inferior izquierda, en la parte superior izquierda se encuentra la vida del jugador la cual es de 100 y a medida que un zombie lo ataque la vida disminuye, en la parte superior derecha podemos visualizar un mini mapa de la ciudad en el cual se puede ver al jugador como una flecha, los zombies de color rojo y cuando mueren como una calavera.



### Interfaz No. 12: Interfaz Jugador

#### 8.2.5 Menú de pausa

En este menú tienen 3 opciones, regresar al menú principal, cambiar las gráficas del juego y salir del juego, mientras este menú este activado el juego se pausaran los zombies y el jugador no podrá moverse.





**Interfaz No. 13: Menú de pausa 1**

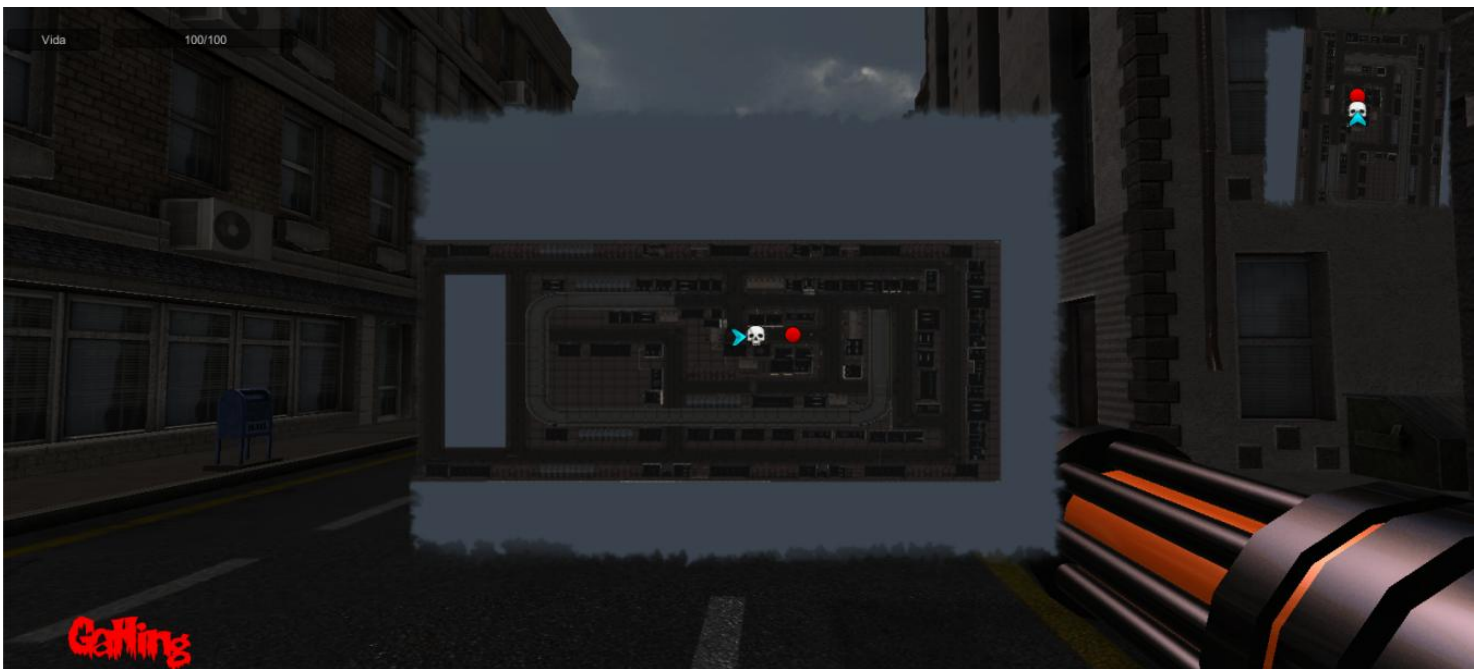


**Interfaz No. 14: Menu de pausa 2**

### 8.2.6 Mapa

El juego cuenta con un mini mapa pero si el jugador quiere visualizar toda la ciudad puede presionar la tecla “m” en la cual observara parte de la ciudad más detalladamente, cuando se visualiza el mapa el juego sigue ejecutándose, en el mapa se pueden ver diferentes iconos:

- A) Punto rojo: El punto rojo lo tienen los zombies.
- B) Calavera: La calavera es para los zombies que están muertos.
- C) Signo de interrogación: En este signo podemos comenzar a realizar misiones del juego.
- D) Flecha: Esta le corresponde a nuestro personaje



Interfaz No. 15: Mapa 1



## Interfaz No. 16: Mapa 2

### 8.2.7 Guardar Partida

Esta opción nos permite guardar el juego en cualquier punto de la historia, se activa con la tecla “G” después presionamos el botón “Guardar Juego” y nos muestra la partida guardada para que sigamos jugando en el punto que la dejamos.



**Interfaz No. 17: Guardar partida 1**



**Interfaz No. 18: Guardar partida 2**

### 8.2.8 Género

El género de este juego es Survival Horror, en este ambiente vamos a conocer la historia de los hechos que desataron la epidemia zombie, además requiere de estrategia para sobrevivir en esta ciudad tan hostil en la cual se desataran hechos jamás vistos en la cual nos llenaremos de suspenso, intriga y terror.

### 8.2.9 Controles

- Movimiento: Para que el personaje se mueva se pueden utilizar:
  - Flecha hacia adelante o la tecla “W”
  - Flecha hacia atrás o la tecla “S”
  - Flecha hacia la derecha o la tecla “D”
  - Flecha hacia la izquierda o la tecla “A”
  - Correr es con la tecla “Shift”
- Mover cámara: la cámara se puede mover con el mouse
- Disparar: Se puede disparar con el botón del mouse izquierdo
- Inventario: El inventario funciona con la tecla “I”
- Saltar: el jugador puede saltar con la tecla “Espaciadora”
- Mapa: El mapa se visualiza con la tecla “M”
- Mini Mapa: Este se activa automáticamente apenas comience el juego.
- Menú: El menú se activa con la tecla “P”
- Guardar partida: Para guardar una partida presionamos la tecla “G”