# Tabla de contenidos

[Tabla de contenidos 2](#_Toc103610967)

[Capítulo 1 - Introducción 6](#_Toc103610968)

[1.1 Motivación del proyecto 6](#_Toc103610969)

[1.2 Objetivos 6](#_Toc103610970)

[1.3 Concepto de videojuegos 7](#_Toc103610971)

[1.4 Géneros de videojuegos 7](#_Toc103610972)

[1.4.1 Juegos de plataforma 8](#_Toc103610973)

[1.4.2 Endless Runner 8](#_Toc103610974)

[1.5 Partes de un videojuego 9](#_Toc103610975)

[1.5.1 Motor 9](#_Toc103610976)

[1.5.2 Assets 10](#_Toc103610977)

[1.5.3 Scripts 10](#_Toc103610978)

[Capítulo 2 – Diseño de un videojuego 11](#_Toc103610979)

[2.1 Introducción 11](#_Toc103610980)

Capítulo 2: Diseño de un videojuego .............................................................. 10

2.1 Introducción …………………………………………………………………………… 10

2.2 Arquitectura software de videojuegos ……………………………………………… 11

2.2.1 Arquitectura software general ........................................................................... 12

2.2.2 Arquitectura software para las entidades de juego ……………………………… 13

2.2.3 Jerarquías de clases ......................................................................................... 14

2.2.4 Arquitectura basada en componentes …………………………………………… 15

2.3 Metodologías para el desarrollo de videojuegos …………………………………... 16

2.4 Metodología ágil para el desarrollo de este proyecto: SUM ……………………… 17

2.4.1 Introducción a la Metodología ........................................................................... 18

2.4.2 Roles ................................................................................................................ 19

2.4.3 Ciclo de Vida .................................................................................................... 20

2.4.4 Concepto .......................................................................................................... 21

2.4.5 Planificación ..................................................................................................... 22

2.4.6 Elaboración ...................................................................................................... 23

2.4.7 Beta .................................................................................................................. 24

2.4.8 Cierre ............................................................................................................... 25

2.4.9 Gestión de Riesgos .......................................................................................... 26

2.4.10 Guías .............................................................................................................. 27

Capítulo 3: Adaptación de la Metodología al Proyecto ................................... 28

3.1 Fase 1: Concepto ................................................................................................ 29

3.1.1 Herramientas .................................................................................................... 30

3.1.2 Elementos de Diseño ........................................................................................ 31

3.2 Fase 2: Planificación ............................................................................................ 47

3.2.1 Planificación Administrativa .............................................................................. 47

3.2.2 Especificación del Videojuego .......................................................................... 47

3.3 Fase 3: Elaboración ............................................................................................. 48

3.3.1 Instalación de Unity y Vuforia ............................................................................ 48

3.3.2 Interconexión de Escenas o Niveles ................................................................. 50

3.3.3 Personaje ......................................................................................................... 63

3.3.4 Enemigos ......................................................................................................... 70

3.3.5 Personaje vs Enemigos .................................................................................... 75

3.3.6 Personaje y la UI .............................................................................................. 79

3.3.7 Ítems ................................................................................................................ 82

3.3.8 Escenarios ....................................................................................................... 90

3.4 Fase 4: Beta ........................................................................................................ 91

3.5 Fase 5: Cierre ...................................................................................................... 92

Capítulo 4: Especificación de Requisitos de Software ................................... 99

4.1 Análisis de Requisitos del Sistema ..................................................................... 99

4.2 Identificación de los Usuarios Participantes ........................................................ 99

4.3 Catálogo de Requisitos del Sistema .................................................................... 99

4.4 Objetivos y Alcance del Sistema ......................................................................... 99

4.5 Definiciones, Acrónimos y Abreviaturas............................................................. 100

4.6 Descripción General .......................................................................................... 100

4.7 Requisitos Funcionales ..................................................................................... 101

4.8 Suposiciones y Dependencias ........................................................................... 101

4.9 Requisitos de Usuario y Tecnológicos ............................................................... 102

4.10 Requisitos de Interfaces Externas ................................................................... 102

4.11 Requisitos de Rendimiento .............................................................................. 102

4.12 Requisitos de Desarrollo .................................................................................. 103

4.13 Restricciones de Diseño .................................................................................. 103

Capítulo 5: Selección del Modelo de Ciclo de Vida y Gestión del Proyecto ..................................................................................................................... 104

5.1 Introducción ....................................................................................................... 104

5.2 Selección de Metodología de Desarrollo de Software ....................................... 104

5.3 Selección de un Modelo de Ciclo de Vida.......................................................... 106

Capítulo 6: Gestión del Proyecto .................................................................. 108

6.1 Introducción ....................................................................................................... 108

6.2 Procedimiento ................................................................................................... 108

6.3 Planificación del Proyecto .................................................................................. 113

6.3.1 Introducción .................................................................................................... 113

6.3.2 Plan del Desarrollo ......................................................................................... 113

Capítulo 7: Modelización y Diseño ............................................................... 115

7.1 Actores .............................................................................................................. 115

7.2 Diagrama de Contexto ....................................................................................... 115

7.3 Listado de Casos de Uso ................................................................................... 115

7.4 Diagrama de Casos de Uso ............................................................................... 115

7.5 Descripción Textual de los Casos de Uso ......................................................... 116

7.6 Diagrama de Actividad ....................................................................................... 117

7.7 Diagrama de Secuencia .................................................................................... 118

7.8 Identificación de Roles ....................................................................................... 119

7.9 Guiones y Escenarios: Diagrama de Transición de Escenarios ........................ 119

7.10 Tabla de Transición de Escenarios .................................................................. 120

7.11 Descripción de Escenarios y Objetos de Escenarios ...................................... 120

Capítulo 8: Implementación y Pruebas ......................................................... 123

8.1 Herramienta Principal para el Desarrollo de la Aplicación ................................ 123

8.1.1 Introducción .................................................................................................... 123

8.2 Desarrollo de la Aplicación ................................................................................ 134

Observaciones ............................................................................................. 144

Conclusiones ............................................................................................... 147

Bibliografía ................................................................................................... 148

# Capítulo 1 - Introducción

## 1.1 Motivación del proyecto

Hoy en día los videojuegos son considerados una de las principales industrias del arte y entretenimiento, estando al nivel o por encima de las industrias del cine y música juntas. El crecimiento del desarrollo *indie* o independiente en los últimos años ha crecido de manera exponencial, principalmente debido a las nuevas herramientas de desarrollo.

Ser desarrollador independiente de videojuegos significa no tener ataduras con lo establecido y luchar por ideas innovadoras que además pueden terminar formando parte de la gran industria. En la actualidad existen muchos desarrolladores independientes que presentaron formulas en la experiencia de la vivencia de un videojuego que no fueron mostradas aún, y terminaron consolidando una nueva forma de diseñarlos.

Por otra parte, dada la circunstancia en la que se encuentra el mundo actualmente, hubo un crecimiento en desarrollo de videojuegos que tienen como temática central a una pandemia o un virus maligno. Es por esa razón que Eggy Runner toma como idea principal combatir al Covid-19.

## 1.2 Objetivos

* Objetivo General: Realizar un videojuego para computadora de escritorio que fomente la competitividad y mejore el uso de reflejos de los jugadores, en una realidad cuyo objetivo sea cuidarse del Covid-19.
* Objetivos Específicos:
* Entender qué es un videojuego y los géneros que existen con sus características, las partes de las que se compone y cómo es usada en cada una de ellas.
* Conocer el proceso de diseño de un videojuego.
* Aplicar en el juego conceptos aprendidos en la carrera, tanto los vistos en asignaturas del ciclo básico como las del ciclo superior.
* Implementar una base de datos en línea que permita acceder a todos los usuarios a información acerca de partidas de otros jugadores.

## 1.3 Concepto de videojuegos

Cuando hablamos de videojuegos, podemos referirnos a los mismos como una experiencia interactiva, o como una herramienta artística o un medio audiovisual más para contar historias.

Sin embargo, las diferentes definiciones hacia estos tienen ciertos aspectos en común:



Fig. 1.1 – Diagrama representativo de las partes de un videojuego

* Un videojuego es un juego que se reproduce a través de un medio electrónico (plataforma)
* Los usuarios (jugadores) se comunican con ese medio a través de una interfaz.
* La cual puede estar compuesta por una pantalla (*output*) y un controlador (*input*) (teclado, mouse, joystick, etc.).
* Los comandos que ingresan por los mismos provocan cambios en el juego (acciones).
* Y el juego comunica estos cambios de estado mediante un *feedback* (mostrar algo por pantalla, sonido, vibrar el mando, etc.).

## 1.4 Géneros de videojuegos

El género de videojuego es una forma de clasificar un videojuego en función fundamentalmente de su mecánica de juego. La mecánica de juego es una regla o conjunto de reglas cuyo objetivo consiste en obtener una serie de resultados coherentes en el seno de un juego.

Existen muchos géneros: juegos de acción, aventura, *survival horror*, RPG *(Role Playing Games)*, estrategia, deportivos, entre otros.

Para este informe se hará foco en dos géneros: juegos de plataforma y *Endless Runner*.

### 1.4.1 Juegos de plataforma

Los videojuegos de plataformas o, simplemente, plataformas, son un género de videojuegos en los que el jugador tiene que saltar sobre plataformas suspendidas o a lo largo de obstáculos y enemigos hasta llegar a una meta. Se considera la presencia de un botón o función de saltar como parte central de la forma de juego de este género, aunque en algunos casos la capacidad de saltar del personaje es reemplazada por otras similares como escalar con ganchos o utilizar resortes. Es muy común que incluyan también elementos de otros géneros como disparos, *beat 'em up* o aventura.

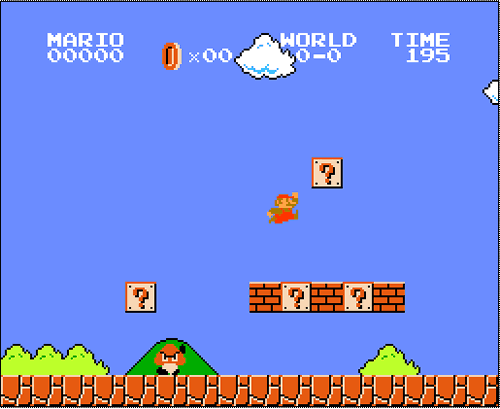


Fig. 1.2 – Gameplay del juego Super Mario Bros

### 1.4.2 Endless Runner

Los juegos Endless Runner son un género de videojuegos que se define fundamentalmente por dos características principales:

* El personaje protagonista avanza sin parar, su impulso es ir siempre hacia adelante.
* El escenario nunca termina.

Cuanto más se avanza en la partida, mayores son la velocidad y los premios que se consiguen. El avatar controlado por el jugador también va cada vez más rápido; pasado un tiempo es casi incontrolable. Es interesante señalar que el progreso en estos juegos se mide en función de la distancia recorrida, con el objeto de llegar cada vez más lejos evitando toda clase de obstáculos.



Fig. 1.3 – Gameplay del juego Temple Run

## 1.5 Partes de un videojuego

Se puede resumir las partes de todo videojuego en las siguientes tres:

* Motor
* *Assets* (texturas, mallas, sonidos, animaciones, etc.)
* *Scripts*

### 1.5.1 Motor

Los motores de videojuegos suelen proporcionar un conjunto de herramientas de desarrollo visual y componentes de software que puedan ser reutilizables. Estas herramientas generalmente se proporcionan en un entorno de desarrollo integrado que permiten crear videojuegos de forma rápida y simple a través de una base de datos.​ En otros casos, los motores se distribuyen con una interfaz de programación de aplicaciones (API) incorporada;​ y otros motores, sin embargo, se distribuyen como un conjunto de herramientas que agilizan y simplifican aún más el desarrollo de un videojuego, como por ejemplo: los entornos de desarrollo integrado, scripts pre-programados, y los middlewares (capaces de interconectar varios programas). Esto resulta útil a la hora de conseguir una plataforma de software flexible y reutilizable que evite la compra de otros recursos ajenos, lo cual ayuda a tener todo lo necesario para hacerlo funcional de manera inmediata, reducir los costos, complejidades y tiempos de comercialización.



Fig. 1.4 – Los motores gráficos más icónicos de los videojuegos

Un motor de videojuego se puede dividir en dos grandes categorías: motor gráfico y motor físico. Los motores gráficos tratan el aspecto visual del videojuego, que generan imágenes sintéticas integrando o cambiando información visual y espacial. Los motores físicos se ocupan de integrar las leyes de la física, siendo responsables de simular acciones reales, a través de variables como la gravedad, la masa, la fricción, la fuerza y la flexibilidad. ​

A pesar de la especificidad del nombre, los motores de videojuego también se utilizan para crear otros tipos de aplicaciones interactivas denominadas como "juegos serios",​ tales como visualizaciones arquitectónicas, educación, ​​ avances científicos, ​ simulaciones de entrenamiento, ​herramientas de modelado​ y simulaciones físicas para recrear animaciones.

### 1.5.2 Assets

Son los distintos elementos que realizan los artistas del equipo de trabajo:

* Mallas: son objetos 3D expresados como secuencia de puntos 3D.
* Texturas: son imágenes 2D que se pintan sobre los objetos 3D.
* Animaciones: son secuencias de puntos 3D que representan el movimiento de la malla.
* Sonidos: son los efectos especiales que responden a un evento o un lugar específico.

Cada objeto 3D de un videojuego está modelado con un programa especial para ello (Maya, 3D Studio Max, etc.). Un modelo 3D es un grupo de puntos en el espacio unidos por líneas, conformando una “malla”. Esta malla está compuesta por triángulos. Mientras más triángulos tenga un modelo, más costará pintarlo en pantalla, y será más carga para el CPU. Además de la "malla“, un objeto 3D está compuesto por un material que contiene la textura del modelo, y otra serie de componentes como el mapa de iluminación, normales, etc. Las animaciones son copias de la malla en distintas posiciones emulando un movimiento.

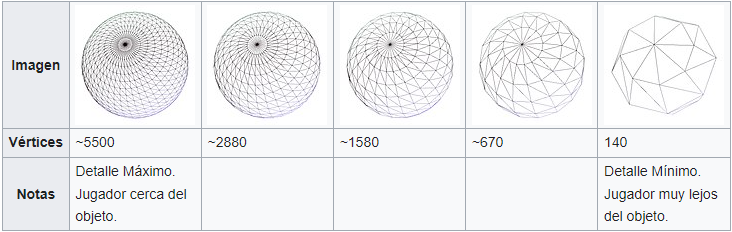


Fig. 1.5 – Nivel de detalle y distancia de dibujado de una malla tridimensional

### 1.5.3 Scripts

Lenguaje de programación de alto nivel, de codificación simple, con pocas funciones (todas ellas relativas al motor del juego). Utilizado para codificar los eventos del juego. Los scripts se pueden usar para hacer todo o casi todo en un videojuego. Se pueden crear objetos dinámicamente, destruirlos, añadirles componentes, quitárselos o crear comportamientos (tanto para el objeto, como para cada uno de sus componentes).

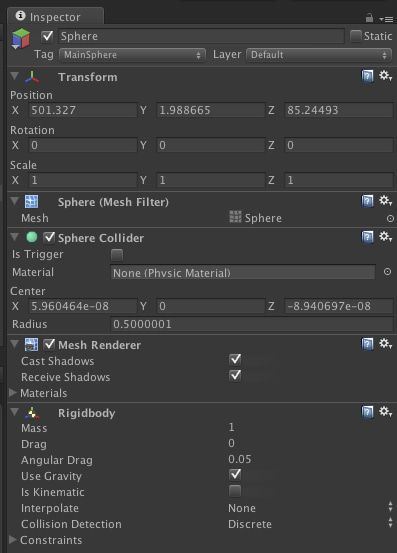


Fig. 1.6 – Componentes de un objeto en Unity

# Capítulo 2 – Diseño de un videojuego

## 2.1 Introducción

Desde la aparición de los primeros videojuegos hasta el día de hoy, los juegos y su proceso de desarrollo han recorrido un largo camino. Los primeros videojuegos no mostraban más que unos puntos, líneas o figuras geométricas en la pantalla y se controlaban mediante el uso de simples potenciómetros o pulsadores. Sin embargo, hoy en día, muestran gráficos 3D extremadamente realistas y, los modos de control, son tan abundantes como diversos, yendo desde teclado o mandos hasta reconocimiento de movimiento mediante el uso de cámaras.

El aumento de la popularidad de los videojuegos ha provocado un incremento en la exigencia y, por tanto, el número de personas involucradas en el desarrollo ha ido creciendo con los años. Antiguamente uno o dos programadores se encargaban de todo el proceso de desarrollo mientras que hoy en día, las grandes producciones, involucran a una gran cantidad de gente que cumple diferentes roles dentro del desarrollo. Debido a este fenómeno de especialización de equipo, ahora hay diferentes roles dentro de un desarrollo, donde ya no existen solamente programadores que se encarguen de todas las tareas (al menos en desarrollos medios y grandes), sino que ahora se tiene un equipo multidisciplinar de personas que provienen de áreas muy diferentes. Además de los programadores, que implementan la funcionalidad del juego, y los artistas, que generan los recursos gráficos, se puede destacar el rol de diseñador. Los diseñadores son los encargados del aspecto creativo del desarrollo y se dedican a idear y desarrollar la jugabilidad del juego, sus mecánicas, la historia de trasfondo con sus posibles acontecimientos y diálogos, los tipos de personajes, cómo éstos deben comportarse, etc. Al incremento de la exigencia y la dificultad de gestionar un grupo multidisciplinar se le suma el hecho de que el diseño de mecánicas de juego requiere de un desarrollo rápido de funcionalidad para probar si las ideas diseñadas “funcionan” o debe replantearse el diseño del juego.

Todas estas cuestiones provocan que las metodologías clásicas de desarrollo software no sirvan y se deba optar por metodologías actuales, más flexibles y que asimilen relativamente bien cambios en la especificación, primando la velocidad de desarrollo frente a la sobreingeniería. La sobreingeniería, a grandes rasgos, es el hecho de agregar diversas opciones o funcionalidades a un producto, con la finalidad de que se utilicen en un futuro. En el mundo de la programación, la sobreingeniería implica tener un montón de código, que solo hace más pesado el programa, y que además enreda a los desarrolladores, pero que está ahí porque quizá en algún momento se llegue a utilizar, cosa que rara vez pasa.

El desarrollo de mecánicas o características de un juego suele ser un proceso iterativo donde las ideas que van surgiendo se documentan, se implementan, se evalúan y, en función de esa evaluación, se descartan, se aceptan o se refinan, volviendo a pasar de nuevo por todas las fases anteriores. Con el paso de los años la tecnología usada para soportar este tipo de metodologías ha ido evolucionando y, hoy en día, prácticamente todos los juegos del mercado usan una arquitectura basada en componentes para implementar las diferentes entidades u objetos del juego. Este tipo de tecnología es muy flexible, permitiendo todo tipo de modificaciones en las especificaciones y premiando la reusabilidad de código. Estas arquitecturas están basadas en composición en lugar de la herencia, la funcionalidad de las entidades del juego se encuentra fraccionada y repartida por varios componentes software. Esto aporta velocidad a la hora de formar entidades nuevas o modificar ya creadas y ahorra grandes tiempos de compilación, pero dificulta también la comprensión a alto nivel de qué es cada entidad, ya que se limita a un conjunto de componentes.

El desarrollo de videojuegos es una tarea llevada a cabo por un equipo compuesto por perfiles de campos muy diferentes. Los roles que se necesitan varían de proyecto a proyecto. De todas maneras, se puede considerar que hay tres roles básicos que aparecen siempre en el proceso de producción: artista, programador y diseñador.

Los artistas son los responsables de la creación de todos los recursos gráficos del juego. Su trabajo es crear y texturizar los modelos de los personajes con sus correspondientes animaciones, elementos del mundo, niveles y todo tipo de contenido 3D así como realizar las imágenes usadas en el juego para los menús y la interfaz de usuario.

La responsabilidad del programador es crear todo el código necesario para hacer que todas las ideas descritas por los diseñadores se hagan realidad y funcionen usando los recursos que han generado los artistas. Los programadores están involucrados en todos los aspectos del juego: gráficos, animaciones, lógica de juego, gameplay (jugabilidad), interfaz de usuario, física, sonido, scripts, etc. Los programadores suelen ser ingenieros informáticos y, debido a que están implicados en todos los aspectos del juego y son los que saben acerca de las limitaciones técnicas existentes, deben evaluar los riesgos técnicos del diseño del juego y acordar con los diseñadores qué cosas se pueden hacer y qué cosas no son viables y deben ser rediseñadas. De la misma manera, son los encargados de poner restricciones técnicas a los recursos que generan los artistas y que deben ser incluidos en el juego.

Los diseñadores son la cabeza pensante del proyecto, son los que deciden el rumbo que éste debe llevar y cual es el resultado que se desea obtener. Están involucrados desde el principio al fin del proyecto, siendo los encargados de desarrollar la idea inicial del juego y pensar en hasta el último detalle del producto que se quiere obtener, controlando que realmente el rumbo sea el adecuado. Entre sus responsabilidades, aparte de decidir en qué va a consistir el juego, está establecer cuáles van a ser las mecánicas principales de éste, las reglas, cómo va a interaccionar el jugador, qué respuesta visual y sonora va a obtener, cómo van a ser los personajes, cuál va a ser la historia del juego, guión, diálogos o diseñar los diferentes niveles que se tendrán. Una de las tareas más importantes del diseñador es saber transmitir exactamente qué es lo que quiere de cada miembro del equipo y de plasmar perfectamente cuáles son los requisitos.

2.1 Introducción …………………………………………………………………………… 10

2.2 Arquitectura software de videojuegos ……………………………………………… 11

2.2.1 Arquitectura software general ........................................................................... 12

2.2.2 Arquitectura software para las entidades de juego ……………………………… 13

2.2.3 Jerarquías de clases ......................................................................................... 14

2.2.4 Arquitectura basada en componentes …………………………………………… 15

2.3 Metodologías para el desarrollo de videojuegos …………………………………... 16

2.4 Metodología ágil para el desarrollo de este proyecto: SUM ……………………… 17

2.4.1 Introducción a la Metodología ........................................................................... 18

2.4.2 Roles ................................................................................................................ 19

2.4.3 Ciclo de Vida .................................................................................................... 20

2.4.4 Concepto .......................................................................................................... 21

2.4.5 Planificación ..................................................................................................... 22

2.4.6 Elaboración ...................................................................................................... 23

2.4.7 Beta .................................................................................................................. 24

2.4.8 Cierre ............................................................................................................... 25

2.4.9 Gestión de Riesgos .......................................................................................... 26

2.4.10 Guías .............................................................................................................. 27