



Universidad Austral de Chile

Facultad de Ciencias de la Ingeniería
Escuela de Ingeniería Civil en Informática

PROTOTIPO DE ENTORNO DE REALIDAD VIRTUAL PARA EL USO DE TESTING, CON EL OBJETO DE CAPTURAR EXPERIENCIAS DE USUARIOS

Proyecto para optar al título de
Ingeniero Civil en Informática

PROFESOR PATROCINANTE:
SR LUIS VIDAL VIDAL
INGENIERO CIVIL EN INFORMÁTICA
M.B.A.
DIPL. BIG DATA PARA LA TOMA DE DECISIONES

PROFESOR CO-PATROCINANTE
SR CHRISTIAN LAZO RAMÍREZ
INGENIERO GESTIÓN INFORMÁTICA
DOCTOR EN INGENIERÍA TELEMÁTICA

PROFESOR INFORMANTE
SR. LUIS ÁLVAREZ GONZÁLEZ
INGENIERO CIVIL ELECTRICISTA
MAGÍSTER EN INGENIERÍA INFORMÁTICA

ISRAEL ANDRÉS DÍAZ ROSAS

VALDIVIA – CHILE
2020

ÍNDICE DE CONTENIDOS

ÍNDICE DE CONTENIDOS	I
ÍNDICE DE TABLAS	IV
ÍNDICE DE FIGURAS.....	V
RESUMEN.....	VII
ABSTRACT.....	VII
1. INTRODUCCIÓN	1
1.1. Definición e historia de la Realidad Virtual.....	1
1.2. Sistemas de Realidad Virtual (<i>VR Headsets</i>) en la actualidad.....	2
1.2.1. Sistemas inmersivos de realidad virtual.....	2
1.2.2. Sistemas VR semi inmersivos.....	5
1.2.3. Sistemas de escritorio VR	7
1.3. Problemática.....	8
1.4. Objetivos General y específicos.....	9
1.4.1. Objetivo General	9
1.4.2. Objetivos Específicos.....	9
1.5. Organización del documento.....	10
2. MARCO TEÓRICO.....	11
2.1. Simulaciones de entornos empleando de realidad virtual	11
2.1.1. Simulaciones RV para entrenamiento de personal de empresas mineras en Sudáfrica	11
2.1.2. Entrenamiento de Bus Ambulancia para personal de primeros auxilios.....	11
2.1.3. Gestión del riesgo de desastres y preparación para emergencias: una simulación de capacitación basada en casos que utiliza la realidad virtual inmersiva	12
2.2. Áreas de aplicación	12
2.2.1. Medicina.....	12
2.2.2. Terapia.....	14
2.2.3. Videojuegos	14
2.2.4. Ingeniería y Arquitectura	15
2.3. Tecnologías para el desarrollo en Realidad Virtual	16
2.3.1. Software	16
2.3.1.1. Google VR	16

2.3.1.2. Unity.....	16
2.3.1.3. Unreal Engine.....	16
2.3.1.4. Comparación entre Motores	17
2.3.2. Hardware	18
2.3.2.1. Tecnología <i>Google Cardboard</i>	18
2.3.2.2. Tecnología Oculus	18
2.3.2.3. Comparación de tecnologías	21
2.4. Consideraciones para el Diseño de entornos de Realidad Virtual	22
2.4.1. Problemas del Desarrollo en Realidad Virtual.....	22
2.4.1.1. Problemas desde el punto de vista del usuario.....	22
2.4.1.2. Problemas desde el punto de vista del desarrollador	22
2.4.2. Consideraciones	22
2.5. Herramientas seleccionadas	23
2.5.1 Hardware	23
2.5.1.1. Smartphone	23
2.5.1.2. Oculus	24
2.5.2 Software	24
3. DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE VR PARA SIMULAR SITUACIONES DE EMERGENCIA	25
3.1. Toma de requerimientos.....	25
3.1.1. Requerimientos del negocio.....	25
3.1.2. Requerimientos de <i>Stakeholder</i>	25
3.1.3. Requerimientos de Implementación.....	26
3.1.3.1. Requerimientos Funcionales	26
3.1.3.1. Requerimientos No funcionales	27
3.1.4. Requerimientos de transición.....	27
3.2. Diseño	28
3.2.1. Casos de uso	28
3.2.2. Diagramas de secuencia	36
3.2.3. <i>Storyboard</i>	38
3.3. Implementación.....	41
3.3.1 Configuración entorno de Desarrollo.....	41
3.3.2. Pruebas Unitarias	42
3.3.3. Aplicación	43

3.3.3.1. Construcción de una escena	45
3.3.3.2 Desarrollo orientado a Google VR.....	47
3.3.3.3 Desarrollo orientado a <i>Oculus Quest</i>	50
3.3.4. Compilar Aplicación	54
4. EVALUACIÓN	58
4.1. Instrumento de análisis.....	58
4.2. Percepción del usuario desarrollador	58
4.2.1. Resultados	58
4.3. Percepción de la experiencia.....	61
4.3.1. Resultados	61
5. CONCLUSIONES	63
6. REFERENCIAS.....	65
ANEXOS	68
Anexo A: <i>Degrees of Freedom</i>	68
Anexo B: Traslación y Rotación en espacio Virtual.....	69
Anexo C: Script ReticleTimer.cs	70
Anexo D: Script TeleportManager.cs	71
Anexo E: Script GVRTeleportPoint.cs	73
Anexo F: Script Extintor.cs.....	74
Anexo G: Encuesta ‘Desarrollo de Entornos de TESTING en Realidad Virtual’	75
Anexo H: Encuesta ‘Reacción ante situación de incendio en Realidad Virtual’	76

ÍNDICE DE TABLAS

Tabla 1: Comparación de Software para desarrollo en Realidad Virtual.....	17
Tabla 2: Comparación de cascos de Realidad Virtual	21
Tabla 3: Requerimientos del Negocio.....	25
Tabla 4: Requerimientos de los <i>Stakeholders</i>	26
Tabla 5: Requerimientos Funcionales.....	27
Tabla 6: Requerimientos No Funcionales	27
Tabla 7: Caso de Uso - Integrar Software en entorno de Desarrollo	29
Tabla 8: Caso de Uso - Crear Escenario	30
Tabla 9: Caso de Uso - Implementar Locomoción del 'Jugador'	30
Tabla 10: Caso de Uso - Creación de Interfaz de Usuario.....	30
Tabla 11: Caso de Uso - Interacción con Entorno Virtual	31
Tabla 12: Caso de Uso - Compilar Aplicación	31
Tabla 13: Caso de Uso - Instalar Aplicación	31
Tabla 14: Caso de Uso - Seleccionar Aplicación y Escenario	32
Tabla 15: Caso de Uso - Interactuar con Escenario	32
Tabla 16: Caso de Uso Expandido - Implementar Locomoción del Jugador	33
Tabla 17: Caso de Uso Expandido - Creación de Interfaz de Usuario.....	34
Tabla 18: Caso de Uso Expandido - Interacción con entorno VR	35

ÍNDICE DE FIGURAS

Figura 1: <i>Google Daydream View</i>	3
Figura 2: Disposición general Cascos Integrados	4
Figura 3:Prototipo de Cascos Independientes	5
Figura 4: Esquema de uso de CAVE	6
Figura 5: Funcionamiento de CAVE	6
Figura 6: Realidad Virtual de Escritorio Antigua	7
Figura 7: Realidad Virtual de Escritorio Moderna.....	8
Figura 8: Realidad Virtual y entrenamiento quirúrgico	13
Figura 9: Realidad Virtual en Telemedicina	13
Figura 10: Jugando Beat Saber	14
Figura 11: Superhot VR	15
Figura 12: Realidad Virtual integrada en la arquitectura	15
Figura 13: <i>Google Cardboard</i>	18
Figura 14: Componentes Oculus Rift S	19
Figura 15: <i>Oculus Go</i>	20
Figura 16: <i>Oculus Quest</i>	20
Figura 17: Smartphone Xiaomi Redmi 4 Pro.....	23
Figura 18: Diagrama de Casos de Uso para el Desarrollador	28
Figura 19: Diagrama de Casos de Uso para el Cliente	29
Figura 20: Diagrama de Casos de Uso para el Usuario Final	29
Figura 21: Diagrama de Secuencia - Interacción con entorno VR.....	36
Figura 22: Diagrama de Secuencia - Locomoción de Jugador	37
Figura 23: Diagrama de Secuencia - Creación e Interfaz de Usuario	38
Figura 24: <i>Storyboard</i> - Presentación de la Simulación	39
Figura 25: <i>Storyboard</i> - Resultados posibles de la simulación.....	40
Figura 26: Módulos necesarios en instalación de Unity	41
Figura 27: Como importar paquete externo en proyecto	42
Figura 28: Error de sintaxis detectado por Unity	43
Figura 29: Secciones de Editor Unity (parte 1).....	44
Figura 30: Secciones de Editor Unity (parte 2).....	44
Figura 31: Secciones de Editor Unity (parte 3).....	45
Figura 32: Objetos en escena de Unity	46
Figura 33: Objeto <i>Game Manager</i> (GM) dentro de escena	46
Figura 34: Jerarquía de Objetos para Jugador (GVRPlayer)	47
Figura 35: Uso de <i>ReticleTimer</i> dentro de GM.....	48
Figura 36: Uso de <i>TeleportManager</i> dentro de GM	48
Figura 37: Event Trigger de objeto <i>Teleport Point</i>	49
Figura 38: Componentes objeto alarma	49
Figura 39: Configuración de elementos en <i>ReticleTimer</i> , objeto GM	50
Figura 40: Composición del Objeto Jugador para Oculus VR (Player).....	51
Figura 41: Configuración de componente <i>OVRGrabber</i>	51
Figura 42: Objeto usando componente <i>OVRGrabbable</i>	52
Figura 43: Modificación de Código en Script <i>OVRGrabber</i>	53

Figura 44: Objeto usando Scripts <i>OVRGrabbable</i> y Extintor.....	53
Figura 45: Aplicación de Scripts <i>OVRGrabbable</i> y Extintor	54
Figura 46: Ventana <i>Build Settings</i> y configuración preferida para compilación	55
Figura 47: <i>Player Settings</i> configurado para VR en Android (parte 1)	56
Figura 48: <i>Player Settings</i> configurado para VR en Android (parte 2)	57
Figura 49: Gráfica para Experiencia con Realidad Virtual.....	59
Figura 50: Gráfica para Experiencia en Desarrollo en Realidad Virtual	59
Figura 51: Gráfica para Aceptación de Marco de Trabajo.....	59
Figura 52: Captura del video presentado en encuesta (1)	60
Figura 53: Captura del video presentado en encuesta (2)	60
Figura 54: Gráfica para Identificación de Situación de Emergencia	61
Figura 55: Gráfica para Elementos Suficientes en Escenario	62
Figura 56: Gráfica para Realización de Protocolo Esperado	62
Figura 57: Diferencia entre 3DoF y 6DoF	68
Figura 58: Descomposición de Movimiento y Rotación en un espacio virtual	69
Figura 59: Captura de video.....	75
Figura 60: Captura de Video 1	76
Figura 61: Captura de Video 2	77

RESUMEN

Las cifras de accidentes laborales y de afectados en situaciones catastróficas tienen una tendencia ascendente preocupante, que incluso es capaz de causar el mal funcionamiento de los servicios de emergencia tales como los hospitales de urgencias.

Para mitigar estas situaciones, el proyecto Entornos de *Testing* en Realidad Virtual, desarrollado para LEK SpA., empresa de desarrollo informático ubicada en la ciudad de Valdivia, se enfoca en la capacitación de las personas ante situaciones de emergencias y en el seguimiento de protocolos de seguridad, exponiendo a los usuarios a situaciones simuladas.

El objetivo de este proyecto de título es la construcción de un marco de trabajo estandarizado para poder construir dichas simulaciones, reduciendo la carga de trabajo de los diseñadores y desarrolladores.

ABSTRACT

The figures for accidents at work and those affected in catastrophic situations have a worrying upward trend, which is even capable of causing the malfunction of emergency services such as emergency hospitals.

To mitigate these situations, the Virtual Reality Testing Environments project, developed for LEK SpA., A computer development company located in the city of Valdivia, focuses on training people in emergency situations and on monitoring protocols for security, exposing users to simulated situations.

The objective of this title project is to build a standardized framework to be able to build such simulations, reducing the workload of designers and developers.

1. INTRODUCCIÓN

1.1. Definición e historia de la Realidad Virtual

No existe una definición formal del concepto Realidad Virtual (*Virtual Reality*, RV, o VR), pero una de las muchas que hay en la actualidad dice que es un “entorno generado por computadora que simula experiencias a través de la percepción, introduciendo al usuario en un mundo totalmente inmersivo”, por otra parte, la RAE la define como una “representación de escenas o imágenes de objetos producida por un sistema informático, que da la sensación de su existencia real”.

Al igual que la definición del concepto, sus inicios también son difusos, pero se puede remontar el origen a la primera mitad del siglo XIX, con la invención del estereoscopio, y por tanto de las imágenes estereoscópicas. Formalmente, el primer avance en esta tecnología se remonta a 1957 (patentado posteriormente en 1962 (EEUU Patente nº 3050870, 1962)), con la invención de *Sensorama*, un simulador que combina imágenes 3D con sonido, vientos y olores para crear una ilusión de la realidad.

Luego en 1961, la empresa PHILCO crea las primeras gafas llamadas *Headsight*. Para 1968, ingenieros del MIT¹ crean un casco de Realidad Virtual conectado a una PC, dicho casco debía colgarse a unos ganchos en el techo para poder funcionar sin aplastar al usuario (How did virtual reality begin?, s.f.; The Rise and Fall and Rise of Virtual Reality, s.f.).

En 1982, se estrena la película TRON, dando a conocer a las masas el concepto de la Realidad Virtual, para posteriormente volver a pasar desapercibido a excepción de algunas apariciones esporádicas en televisión.

Ya en la década de 1990, en la industria de los videojuegos, para el año 1993, SEGA desarrolla un prototipo, nombrado *SEGA VR*, que nunca llegó a comercializarse, y en 1995 Nintendo lanza el *Virtual Boy*, con muy poca penetración en el mercado. En 1999, se estrena THE MATRIX, devolviendo el concepto a la industria del espectáculo y expandiendo el concepto de mundo virtual y Realidad Virtual alrededor del globo.

En 2012, Oculus presenta el primer prototipo de Oculus Rift, iniciativa KickStarter que rindió enormes frutos, recaudando la cifra no menor de USD \$2.437.429 (Nafees, 2017), impulsando enormemente el campo del diseño y desarrollo en Realidad Virtual para finalmente en 2014, ser adquirido por Facebook, quien en años posteriores ha continuado absorbiendo y asociándose con empresas de desarrollo de Software y Hardware para Realidad Virtual.

¹ Instituto Tecnológico de Massachusetts

1.2. Sistemas de Realidad Virtual (*VR Headsets*) en la actualidad

La tecnología de realidad virtual está evolucionando rápidamente, y sería arriesgado definir la realidad virtual en términos de dispositivos específicos que podrían caer en desgracia a corto plazo. En general, la realidad virtual se refiere a una tecnología en la que un usuario interactúa con un entorno tridimensional simulado por computadora que se percibe como comparable a los objetos y eventos del mundo real. A medida que el hardware y el software de la computadora mejoran y la tecnología se actualiza con frecuencia, la facilidad con la que se pueden desarrollar e implementar simulaciones interactivas ha mejorado considerablemente y las herramientas de desarrollo de alta calidad y menor costo han estado disponibles. Según el nivel de inmersión, se puede distinguir entre tres categorías de sistema de realidad virtual.

1.2.1. Sistemas inmersivos de realidad virtual

Los sistemas inmersivos son los más avanzados técnicamente. El usuario está esencialmente aislado del mundo exterior y completamente envuelto en un entorno generado por computadora. Los sistemas de realidad virtual inmersiva requieren que los usuarios usen una unidad de pantalla montada en la cabeza (HMD) que presenta una imagen directamente en frente de cada ojo y la magnífica para llenar un amplio campo de visión. Esto crea la impresión de estar realmente dentro de un entorno, en lugar de observar una pantalla. La vista se basa en imágenes generadas por computadora que reaccionan a la posición y orientación de la cabeza del usuario. La tecnología HMD tiene ciertas desventajas, incluida la molestia debido al uso de cascos con cables, una sensación de aislamiento al ver el mundo virtual sin saber lo que sucede a su alrededor en el mundo real, el alto costo y la enfermedad y la desorientación ocasionales del simulador. Los cascos de RV modernos ahora se ajustan a una de tres categorías o enfoques: **móviles (smartphone)**, **integrados (tethered)** o **independientes (standalone)**.

- Los **cascos de RV móviles** son carcasa con lentes en las que el usuario coloca su smartphone. Las lentes separan la pantalla en dos imágenes para sus ojos, convirtiendo su teléfono inteligente en un dispositivo de realidad virtual. Los cascos móviles como el *Samsung Gear VR* y el *Google Daydream View* (Ver Figura 1) son relativamente baratos a alrededor de USD \$100 (Henschel, 2019), y debido a que todo el procesamiento se realiza en el teléfono, no necesita conectar ningún cable a los cascos.



Figura 1: *Google Daydream View*²

- Los **cascos integrados (Tethered headsets)** como el *Oculus Rift S*, el *HTC Vive Cosmos* y la *PlayStation VR* están físicamente conectados a las PC (o en el caso de la *PS VR*, una *PlayStation 4*). El cable los hace un poco difíciles de manejar, pero poner todo el procesamiento de vídeo en una caja que no necesita atarse directamente a la cara significa que la experiencia de realidad virtual puede ser mucho más compleja (Ver Figura 2). El uso de una pantalla dedicada en el casco en lugar de un teléfono inteligente mejora drásticamente la fidelidad de la imagen, y los sensores externos o las cámaras que miran hacia afuera en el casco brindan un seguimiento completo del movimiento de 6DoF³ (ver en Anexo A).

El punto en contra de estos dispositivos, además de los cables, es el precio. Las opciones integradas menos costosas son actualmente alrededor de USD \$400 (*Oculus Rift S*, 2018). Y eso es antes de abordar el problema del procesamiento; los cascos *Oculus Rift S* y *HTC Vive* necesitan PC bastante potentes para funcionar, mientras que el *PS VR* requiere una *PlayStation 4*.

² Extraido de <https://www.amazon.com/-/es/Google-Daydream-View-Auriculares-realidad/dp/B01N634P7O>

³ *Degrees Of Freedom* o grados de libertad



Figura 2: Disposición general Cascos Integrados⁴

- Los **cascos independientes (Standalone Headsets)** fueron al principio una novedad útil que ofrecía una idea de la realidad virtual sin una inversión en una PC para juegos o un teléfono insignia. El *Oculus Go* y el *Lenovo Mirage Solo* son cascos capaces que funcionan bien por sí mismos, pero tienen los mismos controles limitados y seguimiento de movimiento que los cascos móviles (Ver Figura 3). Sin embargo, el *Oculus Quest* lanzado hace poco, realmente destaca en esta categoría. *Quest* utiliza cámaras similares al exterior del nuevo *Rift S* para proporcionar un seguimiento de movimiento 6DoF, y utiliza los mismos controles de movimiento *Oculus Touch*. Combinado con un procesador más rápido en comparación con el de *Oculus Go*, el *Quest* ofrece una experiencia de realidad virtual mucho más convincente e inmersiva, todo sin el requisito de cable o PC de la *Rift S*.

⁴ Extraído de <https://www.aniwaa.com/guide/vr-ar/vr-headset-buying-guide/>



Figura 3:Prototipo de Cascos Independientes⁵

1.2.2. Sistemas VR semi inmersivos

En los sistemas de realidad virtual semi inmersivos, las imágenes generadas por computadora se muestran en pantallas grandes mediante proyección estéreo y se ven a través de lentes estéreo especiales. La interacción con los menús en pantalla ocurre a través de un teclado remoto, mientras que otras entradas son manejadas por dispositivos como controladores 3D o joysticks.

El uso de múltiples sistemas basados en proyección puede resultar en un costo sustancial, que puede funcionar en el orden de decenas de miles de dólares estadounidenses, pero se pueden producir imágenes de alta resolución. La configuración en la que el usuario está rodeado de pantallas de proyección a veces se denomina CAVE⁶, tal como se muestra en la Figura 4.

Una CAVE crea la ilusión de inmersión al proyectar imágenes estéreo de alta resolución en las paredes y el piso de un cubo del tamaño de una habitación (Ver Figura 5). Varias personas que usan lentes estéreo livianos pueden entrar y caminar libremente dentro de la CAVE.

Aunque su posición de visualización fija y su área de visualización limitada restringen el rango de interacciones del usuario, los sistemas de realidad virtual proyectados tienen la ventaja de que el usuario no está limitado dentro de un auricular y puede comunicarse libremente con otros participantes.

⁵ Extraido de <https://pixovr.com/2019/09/26/6dof-and-standalone-vr/>

⁶ Entorno Virtual Automático de Cueva

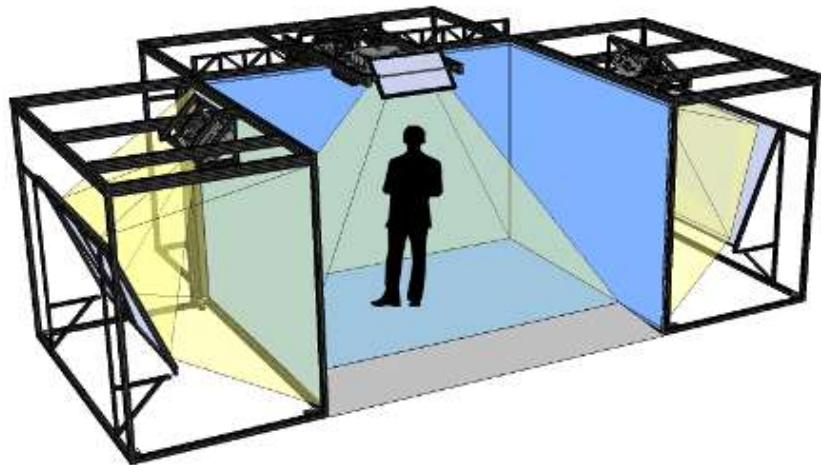


Figura 4: Esquema de uso de CAVE⁷

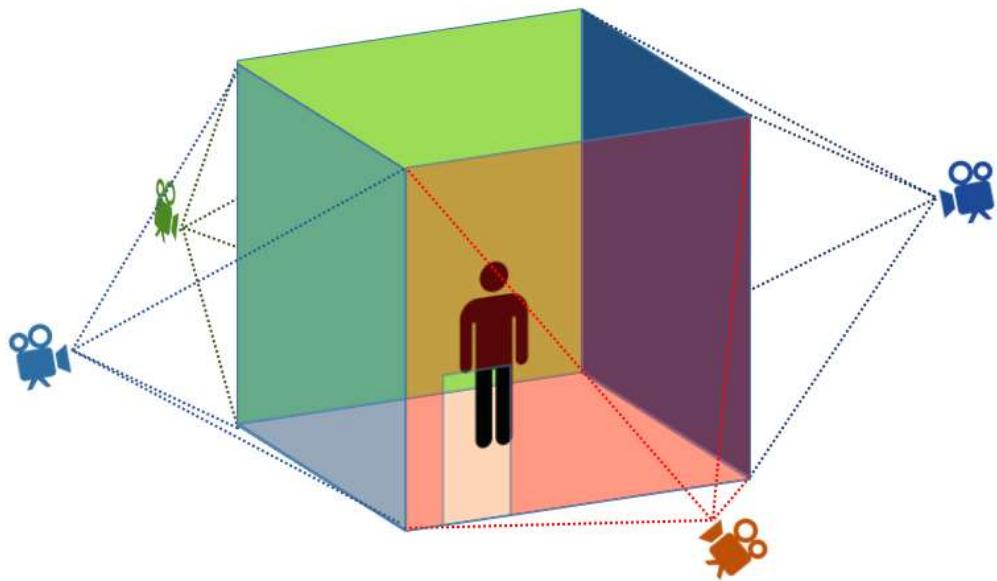


Figura 5: Funcionamiento de CAVE⁸

⁷ Extraído de <http://www.visbox.com/products/cave/>

⁸ Extraído de <https://www.themarketingtechnologist.co/virtual-reality-connecting-unity-to-the-cave/>

1.2.3. Sistemas de escritorio VR

Dado que no todas las aplicaciones requieren inmersión en la medida descrita en las categorías anteriores, los sistemas de realidad virtual más asequibles y no inmersivos ofrecen alternativas prácticas. Una característica importante de la realidad virtual es proporcionar una sensación de presencia real en el entorno simulado. La presencia se refiere a la experiencia subjetiva de "estar" en el entorno generado por computadora, en lugar de en el entorno real del mundo real.

En los sistemas de realidad virtual de escritorio, el usuario permanece visualmente consciente del mundo real, pero también puede observar el mundo virtual en un monitor de alta resolución. En comparación con los sistemas inmersivos, la realidad virtual de escritorio ofrece un menor nivel de presencia, pero el menor costo de capital del hardware, el software y los periféricos hacen que los sistemas de realidad virtual de escritorio sean un sustituto atractivo y realista (Ver Figura 6 y Figura 7).

Los sistemas de escritorio utilizan hardware informático estándar. Los dispositivos de entrada incluyen un teclado, ratón, controlador 3D, *joystick* y *trackpad* o voz para interactuar y manipular el entorno virtual. La sensación de inmersión subjetiva en los sistemas de realidad virtual de escritorio se puede mejorar a través de lentes estereoscópicos, que proporcionan profundidad tridimensional.



Figura 6: Realidad Virtual de Escritorio Antigua⁹

⁹ Extraído de <https://slideplayer.com/slide/7709663/>



Figura 7: Realidad Virtual de Escritorio Moderna¹⁰

Los límites entre estos tipos de sistemas de realidad virtual no son claros. El uso creativo de periféricos de pantalla y audio en sistemas de escritorio o semi-inmersivos puede promover una sensación de presencia como la experimentada en sistemas inmersivos, incluso sin la capacidad de controlar completamente el entorno virtual.

1.3. Problemática

En Chile, durante el año 2018 se registraron más de 515.052 (Reyes, 2019) denuncias por accidentes laborales, accidentes que podrían evitarse, casi en su totalidad según los gremios de construcción (El 98% de los accidentes laborales son evitables, según los sindicatos, 2007; El 90% de los accidentes laborales son evitables, según un informe de CCOO, 1999), si el personal siguiera las normas de seguridad y, en caso de una situación catastrófica, estuviera correctamente preparado.

Las empresas invierten mucho en la contratación de empresas OTEC (en esta ocasión, la empresa DECOTEC¹¹), para capacitar a sus trabajadores en las tecnologías de la compañía, y particularmente para prepararlos para situaciones extraordinarias, que abarcan desde el fallo del motor de una maquinaria, un accidente, a una situación catastrófica, tal como un derrumbe o un incendio.

¹⁰ Extraído de <https://slideplayer.com/slide/7709663/>

¹¹ <http://www.decotec.cl/>

Estas capacitaciones, por lo general, son bastante tediosas, y poco efectivas, puesto que las jornadas de entrenamiento son largas y monótonas.

Para hacer más amena, lúdica y rápida dicha actividad, nace el proyecto de Entornos de *TESTING* en Realidad Virtual para capturas de Experiencias de Usuario. Este proyecto permite, primeramente, interrumpir la monotonía de la clase impartida y, sumergir al usuario en una situación de emergencia simulada, en la cual puede demostrar lo aprendido en la capacitación.

En el contexto de este trabajo de título se implementará un ambiente de *testing* para desarrollo de aplicaciones de realidad virtual que permitan evaluar reacciones ante situaciones de emergencia.

El resultado esperado es un entorno de Realidad Virtual que sirva como alojamiento de distintos escenarios para en posteriores iteraciones apoyar a la toma de datos del comportamiento de las personas y como sistema de entrenamiento para dichos escenarios.

1.4. Objetivos General y específicos

1.4.1. Objetivo General

Este proyecto de título tiene como objetivo el Implementar un ambiente de *testing* para desarrollo de aplicaciones de realidad virtual que permitan evaluar reacciones ante situaciones de emergencia

1.4.2. Objetivos Específicos

Los objetivos específicos para este proyecto son:

- Identificar las herramientas disponibles para el desarrollo de Realidad Virtual, proyectos similares, y normativas para potenciar la correcta construcción de la simulación.
- Desarrollar prototipo de entorno de *TESTING* y primer escenario de prueba.
- Desarrollar métricas de evaluación y modelos que permitan una correcta interacción con el usuario final
- Realizar pruebas para validar que la simulación se realice correctamente.

1.5. Organización del documento

El contenido del documento se encuentra dividido en cinco capítulos organizados de la siguiente manera:

- El primer capítulo aborda la “Introducción” del proyecto de título, definiendo conceptos, los objetivos y la problemática que dan origen al proyecto.
- El segundo capítulo, denominado “Marco teórico”, profundiza en los contenidos más importantes que son tratados a lo largo del documento, referente a simulaciones en Realidad Virtual, aplicaciones de la Realidad Virtual en diversas áreas, las distintas tecnologías (Hardware y Software) existentes y consideraciones a tomar al realizar un desarrollo en Realidad Virtual, finalizando con la descripción de las herramientas a utilizar en el proyecto.
- El tercer capítulo, nombrado “Diseño e implementación de un prototipo de Realidad Virtual para simular situaciones de emergencia” plantea las etapas del Ciclo de Desarrollo de Software para poder analizar, diseñar e implementar la solución propuesta.
- El cuarto capítulo, denominado “Evaluación” describe los pasos realizados para la validación de la solución, así como los resultados de la retroalimentación proporcionada por los involucrados en esta etapa.
- El quinto capítulo aborda las “Conclusiones” del trabajo realizado y futuras mejoras al proyecto.

2. MARCO TEÓRICO

2.1. Simulaciones de entornos empleando de realidad virtual

Entre los múltiples simuladores diseñados para el entrenamiento y educación de personal se puede encontrar tanto propuestas para la industria minera (van Wyk & de Villiers, 2019), o de entrenamiento para personal de salud y primeros auxilios (Koutitas, et al., 2019), como aproximaciones para el público en general, orientadas a preparación contra incendios (Gu, Wang, & He, 2018) o ante otras situaciones de emergencia (Caballero & Niguidula, 2018).

2.1.1. Simulaciones RV para entrenamiento de personal de empresas mineras en Sudáfrica

La minería en Sudáfrica ha sido la principal fuerza impulsora de la historia y el desarrollo de la economía más rica y avanzada de África. Las empresas mineras se esfuerzan por aumentar y mantener la producción y, al mismo tiempo, seguir siendo competitivas en la economía mundial. Además, deben garantizar la seguridad de los trabajadores y mantener buenos registros de seguridad.

El uso de la realidad virtual facilita el desarrollo de herramientas y sistemas para diversos fines que pueden mejorar el conocimiento y la comprensión del entorno laboral. El diseño, desarrollo e implementación de sistemas interactivos de capacitación en realidad virtual se propone como un enfoque innovador para aumentar la capacitación en seguridad. Sin embargo, para evaluar el impacto de tales sistemas de entrenamiento de realidad virtual, es de particular importancia determinar la efectividad del diseño de tales sistemas. Dentro de los estudios realizados y los marcos propuestos se define el trabajo con sistemas de realidad virtual de escritorio para la capacitación de los operadores de las diversas minas en Sudáfrica, dado el gran volumen de personas que son y los costes implicados en los otros sistemas de realidad virtual.

2.1.2. Entrenamiento de Bus Ambulancia para personal de primeros auxilios.

AmBus¹² es una ambulancia del tamaño de un autobús que el personal de EMS¹³ utiliza durante emergencias a gran escala. Aunque el personal de EMS recibe capacitación anual, la evidencia muestra que los esfuerzos de capacitación actuales dejan a algunos miembros del personal sin estar familiarizados con el sistema AmBus y sin estar preparados para responder a una emergencia. Para contrarrestar esto se presenta una nueva aplicación de capacitación interactiva, que utiliza tecnologías emergentes en realidad virtual y aumentada, que se puede entregar de forma remota al personal de EMS distribuido antes de que se ensamblen o mientras se ensamblan. Los hallazgos iniciales muestran que una

¹² Ambulance Bus

¹³ Emergency Medical Services o Servicios Médicos de Emergencia, EE. UU.

aplicación de este tipo puede preparar mejor a los socorristas para que sean lo más efectivos posible en el uso de las funciones de AmBus que salvan vidas. La metodología descrita en este trabajo se puede ampliar para incluir a otros socorristas y, en última instancia, se pueden salvar vidas porque el personal está mejor preparado.

2.1.3. Gestión del riesgo de desastres y preparación para emergencias: una simulación de capacitación basada en casos que utiliza la realidad virtual inmersiva

El riesgo de desastre es la probabilidad de consecuencias dañinas o exposición a condiciones peligrosas que pueden resultar en daños a los medios de vida o la propiedad o la pérdida de vidas. El aumento del riesgo se puede gestionar mediante una planificación adecuada y mediante la realización de una formación adecuada para la gestión de desastres. Este estudio se centra en el desarrollo de capacitación simulada sobre gestión del riesgo de desastres y preparación para emergencias para facilitar programas de concientización sobre desastres, actividades de desarrollo de capacidades y capacitación ambiental real para responder adecuadamente a los desastres. Este estudio también tiene como objetivo implementar una simulación de capacitación basada en casos utilizando tecnología de realidad virtual. La simulación basada en casos sirve como una herramienta de evaluación para medir la competencia de los usuarios inmersos en el entorno virtual. La metodología aplicada en este estudio fue el enfoque cualitativo para diseñar la aplicación de entrenamiento simulado utilizando prototipos con técnicas de gamificación. Con el uso de la tecnología, se logró la simulación de capacitaciones sobre desastres como terremotos, tifones, tsunamis e incendios.

2.2. Áreas de aplicación

Anteriormente se analizó el uso de la Realidad Virtual para el entrenamiento para situaciones de emergencia, pero desde sus orígenes, se han manifestado diversas aplicaciones para la misma, y con los cambios tecnológicos abundantes, las áreas de aplicación incrementan y se perfeccionan constantemente. Algunas de las áreas analizadas son:

2.2.1. Medicina

Las oportunidades que brinda la Realidad Virtual en esta área se enfocan enormemente en el entrenamiento de los médicos cirujanos (ver Figura 8) y sus colaboradores (Gorman, Meier, & Krummel, 1999; Ota, Loftin, Saito, Lea, & Keller, 1995), y en la telemedicina (Chou, Hsu, & Yao, 1997; Satava, 1995), permitiendo a un doctor realizar diagnósticos y proporcionar o guiar el tratamiento de un paciente en una locación distante (ver Figura 9), aunque también hay aplicaciones en el tratamiento de pacientes (Schmitt, et al., 2011).



Figura 8: Realidad Virtual y entrenamiento quirúrgico¹⁴



Figura 9: Realidad Virtual en Telemedicina¹⁵

¹⁴ Obtenido de <https://www.healthysimulation.com/14505/imsh-2018-year-of-the-vr-a-breakdown-of-new-vr-training-opportunities-from-the-exhibit-hall/>

¹⁵ Obtenido de <https://www.elmundo.es/elmundo/2010/09/09/galicia/1284024534.html>

2.2.2. Terapia

Hay diversas aplicaciones para las simulaciones en VR, simples como el tratamiento de fobias (Botella, Osma, García-Palacios, Quero, & Baños, 2004; Kraft & Kraft, 2004), o desórdenes de algún tipo (Meyerbröker & Emmelkamp, 2010; Riva, et al., 2000), e incluso dolores crónicos (Gromala, Tong, Choo, Karamnejad, & Shaw, 2015). Otra aplicación sería la terapia de exposición, utilizada especialmente para soldados en servicio activo (Reger & Gahm, 2008). O para algo más simple, como el tratamiento de la ansiedad por hablar en público (Anderson, Zimand, Hodges, & Rothbaum, 2005).

2.2.3. Videojuegos

El área más conocida de aplicación de la Realidad Virtual. La gran mayoría del software de Realidad Virtual ofrecido mundialmente está orientado al ocio y recreación. A nivel consumidor es el área que ha promocionado la industria del desarrollo de aplicaciones y la manufactura de dispositivos de Realidad Virtual. Algunos de los videojuegos más reconocidos en esta área son *Beat Saber* (ver Figura 10), *Minecraft VR*, *Tetris Effect*, *Superhot VR* (ver Figura 11) y *Doom VFR*. (Gurwin, 2020; Hood, Knapp, & Griliopoulos, 2020)



Figura 10: Jugando Beat Saber¹⁶

¹⁶ Obtenido de <https://medium.com/virtual-reality-virtual-people/beat-saber-63a96e198f63>



Figura 11: Superhot VR¹⁷

2.2.4. Ingeniería y Arquitectura

En el ámbito de la Ingeniería se han diseñado diversos entornos para el estudio y entrenamiento, demostrando que el uso de tecnologías VR y prototipado 3D en contexto de aprendizaje basado en proyectos promueve la comunicación efectiva, incrementa habilidades para resolver problemas, y mejora los resultados del aprendizaje (Halabi, 2020). Se han diseñado entornos para el entrenamiento de ingenieros constructores (Wang, Wu, Wang, Chi, & Wang, 2018), arquitectos (Chou, Hsu, & Yao, 1997), tal como se propone en la Figura 12, e incluso ingenieros mecánicos y (Kamińska, et al., 2017)



Figura 12: Realidad Virtual integrada en la arquitectura¹⁸

¹⁷ Obtenido de <https://www.blogdot.tv/superhot-vr-gets-an-arcade-upgrade/>

¹⁸ Recuperado desde <https://redshift.autodesk.com/big-data-vr-in-architecture/>

2.3. Tecnologías para el desarrollo en Realidad Virtual

Las tecnologías de Realidad Virtual se pueden categorizar en dos ámbitos: Software y Hardware. Es necesario tener una comprensión de las distintas alternativas presentes para poder elegir con qué herramientas trabajar.

2.3.1. Software

2.3.1.1. Google VR

SDK¹⁹ de Google, el cual permite construir aplicaciones móviles en VR tanto para Android como iOS, utilizando AndroidStudio (para desarrollo en Android) o Xcode (para desarrollo en iOS), o motores de juego como Unity y Unreal Engine (Quickstart for Google VR SDK for Android, s.f.).

2.3.1.2. Unity

Unity es un motor de videojuego multiplataforma creado por Unity Technologies. Unity está disponible como plataforma de desarrollo para Microsoft Windows, OS X y GNU/Linux, el cual permite desarrollar sin la necesidad de escribir código, y en caso de desarrollar usando algún lenguaje de programación, permite programar en C# y JavaScript.

Unity cuenta con dos tipos de licencia: Personal y Empresa. Una Cuenta Personal es gratuita, mientras que la licencia de Empresa consta de pagos de mensualidades, ofreciendo descuentos al momento de comprar productos en la tienda de Unity y ofreciendo mayor soporte (Unity Technologies, s.f.).

La plataforma de desarrollo tiene soporte de compilación con diferentes tipos de plataformas, entre ellas Web, PC (Windows, OS X, GNU/Linux), Dispositivos Móviles. Consolas y Dispositivos de Realidad Extendida (XR).

Entre los dispositivos de XR soportados se encuentran Oculus, HTC, *Google Cardboard* y *Google Daydream View*, y mediante el uso de extensiones es posible generar aplicaciones que se ejecuten en los otros dispositivos existentes

2.3.1.3. Unreal Engine

Unreal Engine es un motor de juego popular preferido tanto por estudios AAA como por estudios Indie para desarrollar sus juegos. Grandes títulos como *Mortal Kombat 11*,

¹⁹ *Software Development Kit*

Darksiders III, la serie de *Batman Arkham*, *Ace Combat 7*, *Gears 5*, *Borderlands 3*, *Star Wars Jedi: Fallen Order*, *The Outer Worlds*, etc., se construyeron en Unreal Engine. Entonces, si alguien quiere aprender a usar un motor de juego moderno ampliamente utilizado por estudios de todo el mundo, entonces Unreal Engine es el camino para seguir.

Unreal Engine tiene una arquitectura extremadamente compleja que resulta abrumadora al principio; sin embargo, este es un motor gratificante para aprender porque puede conseguir fácilmente trabajos en el campo del desarrollo de juegos si uno está bien versado en Unreal Engine (Kumar, Integration with Unreal Engine 4, 2020).

2.3.1.4. Comparación entre Motores

La **siguiente tabla** muestra una comparativa entre los distintos motores para desarrollo en Realidad Virtual analizados para el proyecto.

Tabla 1: Comparación de Software para desarrollo en Realidad Virtual²⁰

	Google VR	Unity	Unreal Engine
Lenguaje de programación	Java (Android), Objective-C (iOS)	C#	C++
Interfaz de Usuario	Interfaz del Editor Anfitrión	Compleja, de aspecto plano	Compleja, elaborada
Curva de aprendizaje	Intermedia	Simple	Compleja
Compatibilidad (Sistema Anfitrión)	GNU/Linux, Windows y macOS	Windows, macOS	Windows, GNU/Linux
Licencia	Gratis	Gratis/ Planes de pago	Gratis/ diversas modalidades
Documentación y Cursos	Alto contenido disponible	Basto contenido disponible	Alto contenido disponible

²⁰ Información extraída de:

- <https://sundaysundae.co/unity-vs-unreal/>
- <https://www.creativebloq.com/advice/unity-vs-unreal-engine-which-game-engine-is-for-you#:~:text=One%20of%20the%20main%20differentiators,the%20same%20level%20as%20Unreal>

2.3.2. Hardware

2.3.2.1. Tecnología *Google Cardboard*

Propuesta de Google de bajo costo que convierte cualquier teléfono con Sistema Operativo Android 4.4 o superior en un visor de Realidad Virtual, consta de, como indica su nombre, una base de cartón plegable y unos lentes para armar uno mismo los visores, y luego utilizar el smartphone para desplegar el contenido de VR (ver Figura 13). La calidad de la experiencia, por lo tanto, dependerá explícitamente de las especificaciones técnicas del smartphone utilizado.



Figura 13: *Google Cardboard*²¹

2.3.2.2. Tecnología Oculus

Oculus VR, llamado simplemente Oculus, es una compañía estadounidense que desarrolla tecnología de realidad virtual, fundada en julio de 2012 en Irvine, California, actualmente su sede se encuentra en Menlo Park. Oculus VR se especializa en productos de software y hardware de realidad virtual.

En marzo de 2014, el CEO de Facebook, Mark Zuckerberg, acordó adquirir Oculus VR por US \$2,3 mil millones en efectivo y acciones. En 2015, Oculus VR adquirió Surreal Vision, una startup británica que se centró en la reconstrucción 3D y realidad mixta, afirmando que podría ser posible para Oculus VR desarrollar productos con el concepto de telepresencia.

²¹ Obtenido de <https://tech.hindustantimes.com/tech/news/google-open-sources-cardboard-virtual-reality-software-story-SPFBomLqNqY3ac7J0UlIXM.html>

En los últimos años, Facebook ha adquirido o formado alianzas con diversas empresas, lo que ha expandido tanto el desarrollo de Software como Hardware para Realidad Virtual.

En la actualidad Oculus ofrece diversas alternativas para ingresar al mundo de la Realidad Virtual:

- *Oculus Rift S*, la segunda generación de la *Oculus Rift* original, dispositivo insignia de la empresa, es un casco integrado con grandes especificaciones (ver Figura 14).



Figura 14: Componentes Oculus Rift S²²

- *Oculus Go*, un visor todo en uno diseñado para el entretenimiento ligero, casco independiente de entrada para nuevos usuarios de Realidad Virtual, el cual ofrece una experiencia grata un nivel más arriba que el enfoque móvil (ver Figura 15).
- *Oculus Quest*, un visor todo en uno diseñado para videojuegos, es la novedad de Oculus, un casco independiente que se posiciona entre la *Oculus Go* y *Oculus Rift*, que ofrece gran libertad de movimiento y seguridad con sus cámaras integradas (ver Figura 16).

²² Extraído de https://www.oculus.com/rift-s/?locale=es_ES



Figura 15: *Oculus Go*²³



Figura 16: *Oculus Quest*²⁴

²³ Extraído de <https://www.amazon.co.uk/Oculus-Standalone-Virtual-Reality-Headset/dp/B07D7HPSFC>

²⁴ Extraído de <https://www.grover.com/nl-en/products/oculus-quest-vr-64gb>

2.3.2.3. Comparación de tecnologías

En la Tabla 2 se pueden presenciar algunas de las gafas de Realidad Virtual más utilizadas y novedosas en la actualidad, incluidas las analizadas en las secciones previas.

Tabla 2: Comparación de cascos de Realidad Virtual²⁵

Dispositivo	Tipo	Resolución	Tasa de refresco	Detección de movimiento	Controles	Plataforma Hardware	Plataforma Software
Oculus Quest	Independiente	1440x1600	72 Hz	6DoF	<i>Oculus Touch</i>	Android	Oculus
Oculus Go	Independiente	1280x1440	72 Hz 60 Hz	3DoF	<i>Oculus Go Controller</i>	Android	Oculus
Oculus Rift S	Integrado	1280x1440	80 Hz	6DoF	<i>Oculus Touch</i>	PC	Oculus y SteamVR
HTC VIVE PRO	Integrado	1440x1600	90 Hz	6DoF	<i>Dual controllers</i>	PC	SteamVR
HTC VIVE Cosmos	Integrado	1440x1700	90 Hz	6DoF	<i>Motion Controllers</i>	PC	SteamVR
Google Daydream View	Móvil	depende del dispositivo	depende del dispositivo	3DoF	<i>Controller</i>	Android	Android

²⁵ Información extraída de:

- <https://www.pc当地.com/picks/the-best-vr-headsets>
- https://en.wikipedia.org/wiki/Comparison_of_virtual_reality_headsets
- <https://www.xataka.com/seleccion/guia-compra-gafas-realidad-virtual-16-modelos-para-todas-expectativas-necesidades-presupuestos>

2.4. Consideraciones para el Diseño de entornos de Realidad Virtual

2.4.1. Problemas del Desarrollo en Realidad Virtual

De igual manera que en todo tipo de desarrollo de Software o Hardware, se pueden encontrar distintos problemas, impedimentos o consideraciones para tener en cuenta, tanto por parte del consumidor como por parte del equipo desarrollador.

2.4.1.1. Problemas desde el punto de vista del usuario

Dentro de las complicaciones por parte de los usuarios se pueden encontrar preocupaciones por:

- Falta de contenido.
- Cascos pesados.
- Gafas VR costosas.
- Generalmente se requiere de un computador compatible con VR, de precio elevado por demás.
- Presencia de cables que entorpecen la experiencia.
- No es apto para usuarios con epilepsia.

2.4.1.2. Problemas desde el punto de vista del desarrollador

En cambio, las preocupaciones de los desarrolladores abarcan (Stanney, et al., 2020; Human Factors and User Studies, s.f.) son:

- Mareos.
- Sobre estimulación visual.
- Desorientación en el espacio.
- Fragmentación del usuario.
- Base de usuarios pequeña.

2.4.2. Consideraciones

Teniendo en cuenta las preocupaciones mencionadas anteriormente y otros factores, se puede determinar que al momento de desarrollar una experiencia en Realidad Virtual hay que tener presentes los siguientes puntos:

- Sobre estimulación visual: no sobrecargar los escenarios virtuales puesto que entorpecen la experiencia y causa fatiga en el usuario.
- Locomoción: el movimiento del usuario dentro del mundo virtual es esencial, no es necesario que sea realista en todos los casos, pero si debe ser fluido.
- Realismo: dependiendo de la experiencia a construir, va a depender del factor realista que se deba implementar en el desarrollo
- Interacción con el entorno: una simulación requiere que se interactúe con los objetos que rodean al usuario, definir el cómo interactuar con ellos y cómo

- reaccionarán dichos objetos es un determinante al momento de definir con que herramientas trabajar.
- Interfaz de Usuario / Como desplegar información en el mundo virtual: hay que definir la forma en la que se le entregará información al usuario, como se le darán instrucciones, o se le notificará de algo, etc.

2.5. Herramientas seleccionadas

Una vez realizado el análisis de las herramientas disponibles en la actualidad, las consideraciones a tomar y la complejidad para trabajar con ellas se optó por trabajar con las siguientes tecnologías:

2.5.1 Hardware

Se desarrolla en base a dos tecnologías de Realidad Virtual, Google VR, y Oculus VR.

La primera tecnología fue simple de elegir, es simple y barata, lo único necesario es un smartphone apto para ejecutar aplicaciones de Realidad Virtual. En cambio, la tecnología Oculus fue elegida por su prestigio y las constantes novedades que traen al mercado.

2.5.1.1. Smartphone

Para probar la aplicación resultante se utiliza el smartphone Xiaomi Redmi 4 Pro (Ver Figura 17), pues es un dispositivo relativamente económico y con buenas especificaciones, posee sistema operativo Android 6.0 *Marshmallow*, y 3 GB de memoria RAM.



Figura 17: Smartphone Xiaomi Redmi 4 Pro²⁶

²⁶ Extraído de <https://xiaomi.reviews/xiaomi-redmi-4-pro-prime/>

2.5.1.2. Oculus

Para las pruebas para la tecnología Oculus, se opta por el dispositivo *Oculus Quest*, la novedad de la empresa, la cual permite el movimiento en el escenario sin depender de artefactos externos, tales como un PC o los sensores que deben utilizarse en otros dispositivos. Además, este dispositivo se encuentra en constante evolución, con mejoras de *firmware* constantes e integración de nuevas capacidades futuras tales como la detección de manos revelada en *Oculus Connect 6* (Vanian, 2019).

2.5.2 Software

Para el desarrollo del marco de trabajo se seleccionó como base la plataforma de desarrollo de videojuegos Unity, considerando su curva de aprendizaje sencilla, la gran comunidad de desarrolladores a nivel local y mundial, y conocimiento previo de la plataforma. Y con ello las bibliotecas de desarrollo para los dispositivos hardware utilizados.

Específicamente se utiliza:

- Unity versión 2019.2.2f1,
- Oculus Integration para Unity versión 1.38.00 y
- SDK Google VR para Unity versión 1.200.1

pues son las últimas versiones de los softwares a utilizar disponibles al momento de comenzar el desarrollo.

La posición dentro de la simulación de los objetos y el punto de inicio (coordenadas X, Y, Z) y la orientación (ver Anexo B) del Jugador es el definido dentro de Unity, este luego puede moverse dentro del espacio dentro de los límites del Área de Juego definido por el sistema guardián de la *Oculus Quest*.

3. DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE VR PARA SIMULAR SITUACIONES DE EMERGENCIA

Este capítulo explica el Ciclo de Desarrollo de Software realizado para poder proporcionar el producto final de este trabajo de título. Iniciando con la toma de requerimientos pertinentes al proyecto, el diseño general de la solución para finalizar con la construcción del Software mismo.

3.1. Toma de requerimientos

Para definir claramente las necesidades o deficiencias a solventar y oportunidades a explotar se realizaron varias reuniones con clientes y expertos, en las que se discutieron diversos aspectos de la solución a implementar.

3.1.1. Requerimientos del negocio

Los principales objetivos que fundamentan al proyecto de Entornos de Realidad Virtual se pueden apreciar en la Tabla 3.

Tabla 3: Requerimientos del Negocio²⁷

Requerimiento	Descripción
RN1	Implementar escenarios de estudio para empresas capacitadoras
RN2	Desarrollar Analítica en base a los datos capturados de simulaciones

3.1.2. Requerimientos de *Stakeholder*

A partir de los requerimientos del negocio, y tras reuniones y charlas con los interesados en el proyecto, se pueden apreciar los requerimientos descritos en la Tabla 4.

²⁷ Elaboración propia

Tabla 4: Requerimientos de los *Stakeholders*²⁸

Stakeholder	Descripción
Desarrollador	Utilizar una metodología estándar.
Desarrollador	Todo código escrito debe ser estándar.
Desarrollador	el ‘Código’ debe ser fácil de transferir (importar/exportar) e implementar.
Desarrollador	Es preferible usar objetos comprobados previamente.
Desarrollador	Se prioriza el uso de objetos prefabricados (plantillas, código base, modelos 3d, etc.).
Usuario Final	Debe tener actividades llamativas.
Usuario Final	Los escenarios virtuales deben ser realistas.
Usuario Final	La aplicación debe ofrecer libertad de movimiento
Cliente	evaluar el comportamiento de los usuarios finales luego del periodo de capacitación.
Cliente	Ofrecer múltiples escenarios realistas en Realidad Virtual.
Cliente	alterar/mejorar el proceso normal de capacitación utilizado hasta el momento ('clases de jornada completa').
Cliente	Metodologías y Herramientas deben ser simples de instalar y utilizar.
Gerencia	el software final desarrollado debe de ser de fácil y rápida implementación.

3.1.3. Requerimientos de Implementación

Una vez entendidos los requerimientos de los principales interesados en el proyecto, se puede avanzar a la definición de los requerimientos de implementación (Ver Tabla 5 y Tabla 6).

3.1.3.1. Requerimientos Funcionales

Para los siguientes requerimientos, se entiende como Objeto como cualquier componente de la simulación. En Unity serían elementos por lo general del tipo *GameObject*.

²⁸ Elaboración propia

Tabla 5: Requerimientos Funcionales²⁹

# Req.	Descripción
RF 1	el software debe ser capaz de emplear objetos prefabricados para el objeto del Usuario/Jugador.
RF 2	el software necesita ofrecer integración con al menos un sistema de desarrollo en Realidad Virtual.
RF 3	el software debe permitir el movimiento del usuario.
RF 4	el software permitirá al desarrollador crear y modificar objetos.
RF 5	el software permite capturar la posición del usuario.
RF 6	los datos generados por el software no incluirán información personal del usuario.
RF 7	el software debe permitir modificaciones posteriores.
RF 8	el software deberá ser capaz de simular situaciones de emergencia en un entorno cerrado controlado.
RF 9	el software deberá proveer interacción con los objetos dispuestos en el espacio virtual.

3.1.3.1. Requerimientos No funcionales

Tabla 6: Requerimientos No Funcionales³⁰

# Req.	Descripción
RNF 1	La aplicación deberá ser fácil de instalar.
RNF 2	Las simulaciones generadas deberán ejecutarse fluidamente.

3.1.4. Requerimientos de transición

El Instituto Internacional de Análisis de Negocio (2009) dice: “Estos requerimientos describen las capacidades que la solución debe tener con el fin de facilitar la transición

²⁹ Elaboración propia

³⁰ Elaboración propia

del estado actual de la empresa hacia el estado futuro deseado, pero que no serán requeridos una vez que la transición se haya llevado a cabo”.

Siguiendo esa definición, se considera que el requerimiento de transición más importante para este caso es:

- Exportar/compilar Aplicación a formato pertinente

3.2. Diseño

Teniendo claros los requerimientos a cumplir, se tienen varias charlas para aclarar el uso y los posibles escenarios de uso del software a implementar. Como herramientas se utilizan: Casos de Uso, Diagramas de Secuencia y *Storyboards* para poder plasmar el uso de la solución a construir.

3.2.1. Casos de uso

Tras la definición de los requerimientos de software se construyen los casos de uso. Se pueden apreciar nueve casos de uso (Ver Figuras Figura 18, Figura 19 y Figura 20).

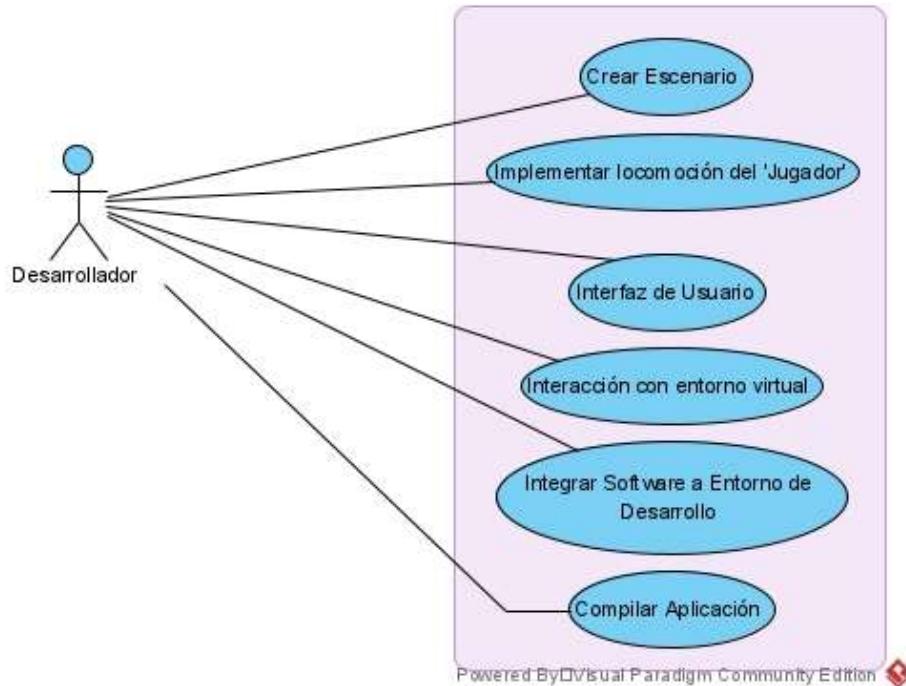


Figura 18: Diagrama de Casos de Uso para el Desarrollador³¹

³¹ Elaboración propia

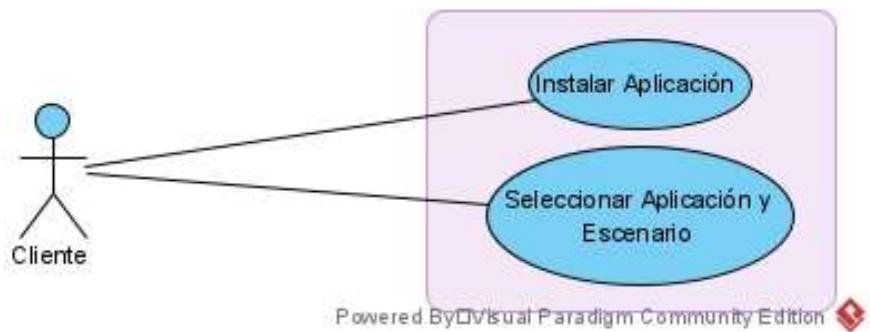


Figura 19: Diagrama de Casos de Uso para el Cliente³²

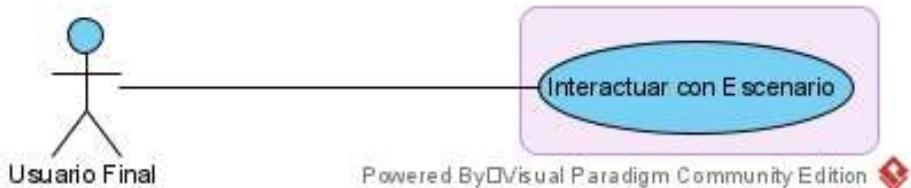


Figura 20: Diagrama de Casos de Uso para el Usuario Final³³

A continuación, se describen los casos de uso en formato de alto nivel desde la Tabla 7 a la Tabla 15. Después de esto, se definen los casos de uso en formato extendido considerando los más representativos del problema desde la Tabla 16 a la Tabla 18.

Tabla 7: Caso de Uso - Integrar Software en entorno de Desarrollo³⁴

CU-01	Integrar Software en Entorno de Desarrollo
Actores	Desarrollador
Tipo	secundario, concreto
Descripción	Este caso comienza cuando el desarrollador decide iniciar un proyecto, importando en su entorno de desarrollo el software que contiene lo necesario para el funcionamiento de la aplicación.

³² Elaboración propia

³³ Elaboración propia

³⁴ Elaboración propia

Tabla 8: Caso de Uso - Crear Escenario³⁵

CU-02	Crear Escenario
Actores	Desarrollador
Tipo	primario, concreto
Descripción	Este caso comienza cuando el desarrollador, dentro del entorno de desarrollo, crea una escena, luego integra y crea objetos dentro de ella para construir un escenario simulado

Tabla 9: Caso de Uso - Implementar Locomoción del 'Jugador'³⁶

CU-03	Implementar Locomoción del 'Jugador'
Actores	Desarrollador
Tipo	secundario, concreto
Descripción	Este caso comienza cuando el desarrollador integra el objeto Jugador dentro de un escenario, y luego implementa la modalidad de movimiento que tendrá el Jugador.

Tabla 10: Caso de Uso - Creación de Interfaz de Usuario³⁷

CU-04	Creación de Interfaz de Usuario
Actores	Desarrollador
Tipo	opcional, concreto
Descripción	Este caso comienza cuando el desarrollador decide que es necesario agregar mensajes para el usuario en el escenario, para lo cual crea un Lienzo dentro del escenario y publica el mensaje a ser leído por el Jugador.

³⁵ Elaboración propia

³⁶ Elaboración propia

³⁷ Elaboración propia

Tabla 11: Caso de Uso - Interacción con Entorno Virtual³⁸

CU-05	Interacción con Entorno Virtual
Actores	Desarrollador
Tipo	primario, concreto
Descripción	Este caso comienza cuando el desarrollador abre el entorno de desarrollo, importa el objeto Jugador y construye objetos con propiedades y funcionalidades específicas para el escenario construido, características que están ligadas al contacto con el objeto Jugador.

Tabla 12: Caso de Uso - Compilar Aplicación³⁹

CU-06	Compilar Aplicación
Actores	Desarrollador
Tipo	secundario, esencial
Descripción	Este caso comienza cuando el desarrollador ha concluido que ha avanzado lo suficiente para probar el escenario construido en un dispositivo de Realidad Virtual, para lo cual selecciona el tipo de dispositivo objetivo, selecciona el escenario a evaluar y lo compila en una Aplicación para ser probada posteriormente.

Tabla 13: Caso de Uso - Instalar Aplicación⁴⁰

CU-07	Instalar Aplicación
Actores	Cliente
Tipo	secundario, esencial
Descripción	Este caso comienza cuando el cliente decide implementar un nuevo escenario de simulación, selecciona la aplicación con el escenario correspondiente, y lo instala en el dispositivo de Realidad Virtual

³⁸ Elaboración propia

³⁹ Elaboración propia

⁴⁰ Elaboración propia

Tabla 14: Caso de Uso - Seleccionar Aplicación y Escenario⁴¹

CU-08	Seleccionar Aplicación y Escenario
Actores	Cliente
Tipo	secundario, esencial
Descripción	Este caso comienza cuando el Cliente decide probar un escenario diseñado, para lo cual inicia el dispositivo de Realidad Virtual, busca entre las Aplicaciones instaladas y selecciona la Aplicación con el escenario a probar

Tabla 15: Caso de Uso - Interactuar con Escenario⁴²

CU-09	Interactuar con Escenario
Actores	Usuario Final, Cliente
Tipo	primario, esencial
Descripción	Este caso comienza cuando el Cliente inicializa un Escenario, y le entrega el dispositivo de Realidad Virtual al Usuario Final, el cual comenzará su actuación siguiendo las instrucciones entregadas por el Cliente y/o el Escenario.

El caso de uso de la Locomoción del Jugador es uno de los más interesantes de estudiar, puesto que determinará las capacidades de exploración y percepción del entorno del usuario. En la Tabla 16 se detallan los procedimientos para implementar dicha función.

Otro caso de uso interesante es el del diseño de la Interfaz de Usuario, que determinará qué, y cómo se le entregará información al usuario, lo cual también afecta la forma de afrontar la simulación. En la Tabla 17 se aprecian los pasos para integrar una Interfaz de Usuario en la simulación.

El último caso de uso estudiado en profundidad es la Interacción con Entorno, punto esencial que determina las características del escenario simulado, las posibles acciones y reacciones del Jugador. En la Tabla 18 se puede ver el detalle de este caso de uso.

⁴¹ Elaboración propia

⁴² Elaboración propia

Tabla 16: Caso de Uso Expandido - Implementar Locomoción del Jugador⁴³

CU-03	
Caso de Uso	Implementar locomoción de Jugador
Actores	Desarrollador
Objetivo / Propósito	Permitir el movimiento del usuario dentro del escenario virtual
Tipo	secundario, concreto
Referencias cruzadas	RF1, RF3, RF5, RF9, CU-01, CU-02
Precondiciones	<ul style="list-style-type: none"> • Software integrado en Entorno de desarrollo • Escenario creado
Postcondiciones	<ul style="list-style-type: none"> • Escenario preparado para interacción con entorno
Resumen: El desarrollador agrega el Jugador y las coordenadas de movilización en la escena para habilitar el movimiento dentro del Escenario	
Curso normal de eventos	
Acción de los Actores	Respuesta del Sistema
1. Desarrollador ingresa a escenario dentro del entorno de desarrollo	2. Carga Escenario
3. Agrega objeto Jugador a la escena	4. Activa cámara de personaje
5. Agrega objetos/coordenadas de transporte (teletransporte)	6. Reconstruye escena de acuerdo al objeto creado y guarda Escenario.
7. Cierra el escenario	
Cursos alternativos	
5. Integra script para movimiento en objeto Jugador (Agitar los brazos)	
5. Diseña ruta de movimiento programado	

⁴³ Elaboración propia

Tabla 17: Caso de Uso Expandido - Creación de Interfaz de Usuario⁴⁴

CU-04	
Caso de Uso	Creación de Interfaz de Usuario
Actores	Desarrollador
Objetivo / Propósito	Entregar información o instrucciones al usuario dentro del Escenario
Tipo	opcional, concreto
Referencias cruzadas	RF1, RF3, RF5, RF9, CU-01, CU-02
Precondiciones	<ul style="list-style-type: none"> • Software integrado en Entorno de desarrollo • Escenario creado
Resumen: El desarrollador agrega el Jugador dentro de la escena y crea lienzos con información acerca del Escenario	
Curso normal de eventos	
Acción de los Actores	Respuesta del Sistema
1. Desarrollador ingresa a escenario dentro del entorno de desarrollo	2. Carga Escenario
3. Agrega objeto Jugador a la escena	4. Activa cámara de personaje
5. Crea Lienzo dentro de la escena	6. Reconstruye escena
7. Anota información a desplegar	8. Reconstruye escena y guarda Escenario
9. Cierra el escenario	

⁴⁴ Elaboración propia

Tabla 18: Caso de Uso Expandido - Interacción con entorno VR⁴⁵

CU-05	
Caso de Uso	Interacción con entorno VR
Actores	Desarrollador
Objetivo / Propósito	Implementar objetos que tengan comportamientos específicos al entrar en contacto con el usuario
Tipo	primario, concreto
Referencias cruzadas	RF1, RF8, RF9, CU-01, CU-02, CU-03
Precondiciones	<ul style="list-style-type: none"> • Software integrado en Entorno de desarrollo • Escenario creado
Postcondiciones	<ul style="list-style-type: none"> • Escenario preparado para compilar como Aplicación
Resumen: el desarrollador agrega el Jugador dentro de la escena y agrega objetos con componentes interactivos.	
Curso normal de eventos	
Acción de los Actores	Respuesta del Sistema
1. El desarrollador ingresa a escenario dentro del entorno de desarrollo	
2. Importa objeto Jugador dentro de la escena	3. Activa cámara de personaje
4. Crea o importa objetos dentro de la escena	5. reconstruye escena
6. Aplica elemento de acción	7. reconstruye objetos interactivos y guarda Escenario
8. Cierra Escenario	
Cursos alternativos	
2. Desarrollador crea objetos y aplica acciones antes de integrar el Jugador a la escena	

⁴⁵ Elaboración propia

3.2.2. Diagramas de secuencia

En las Figuras Figura 21, Figura 22 y Figura 23 se pueden apreciar las interacciones del desarrollador con el Sistema de Desarrollo.

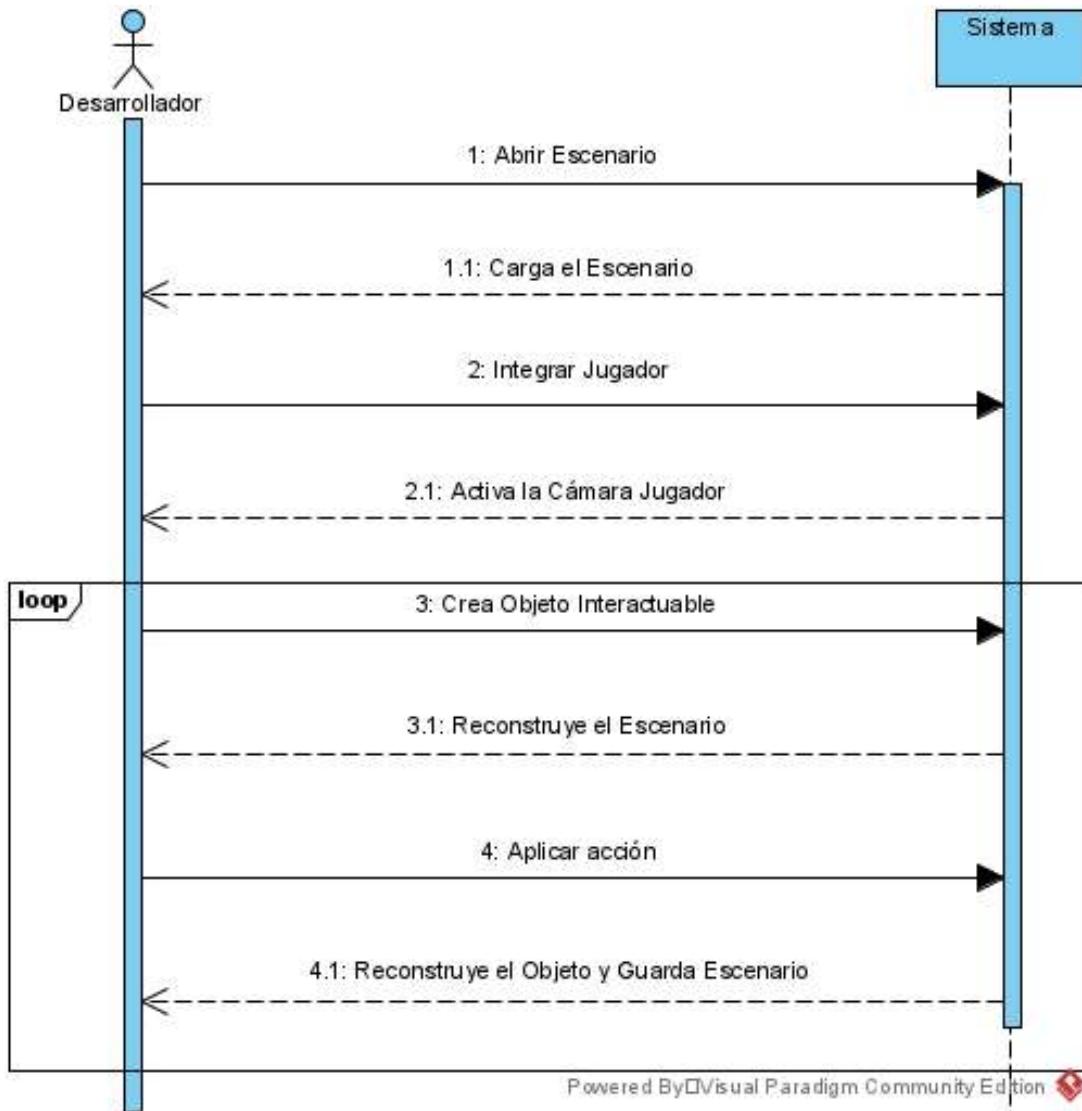


Figura 21: Diagrama de Secuencia - Interacción con entorno VR⁴⁶

⁴⁶ Elaboración propia

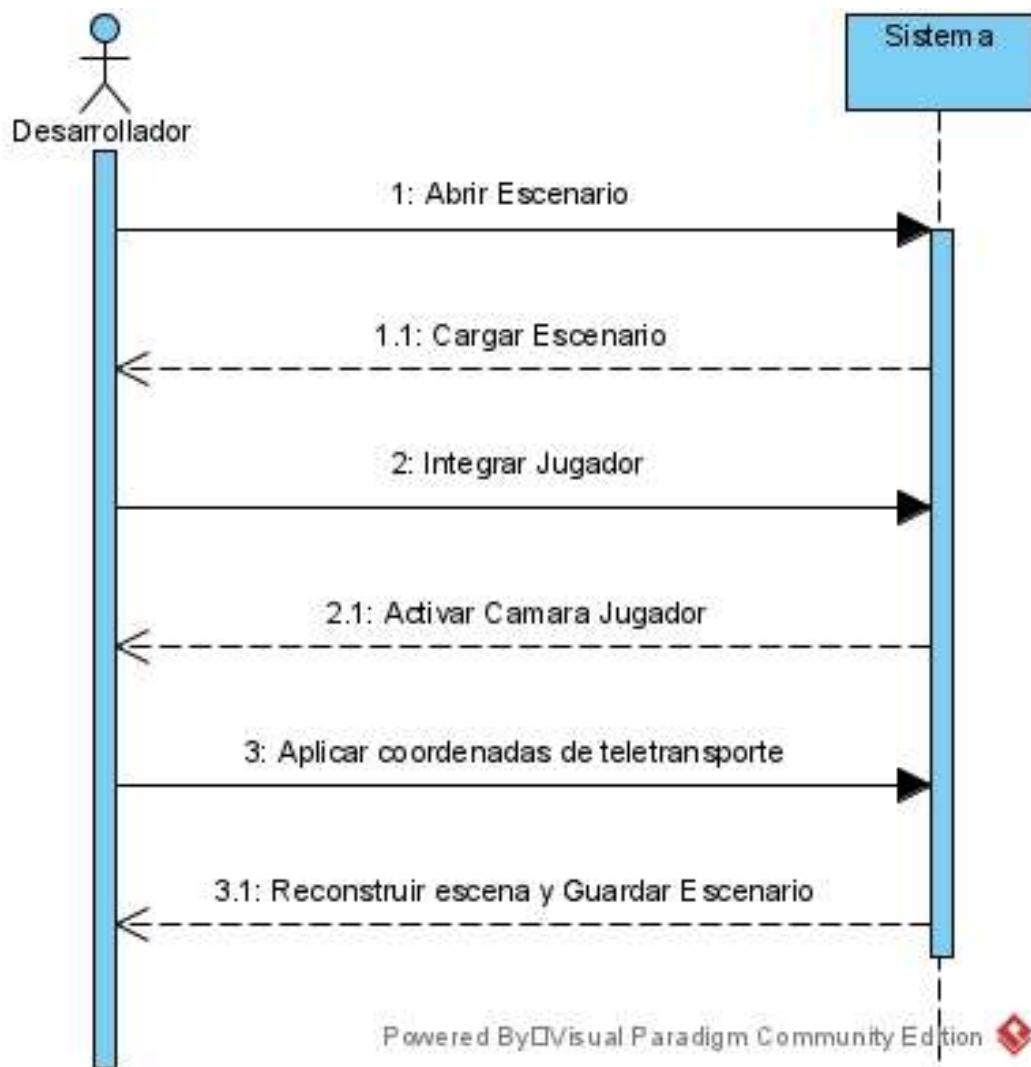


Figura 22: Diagrama de Secuencia - Locomoción de Jugador⁴⁷

⁴⁷ Elaboración propia

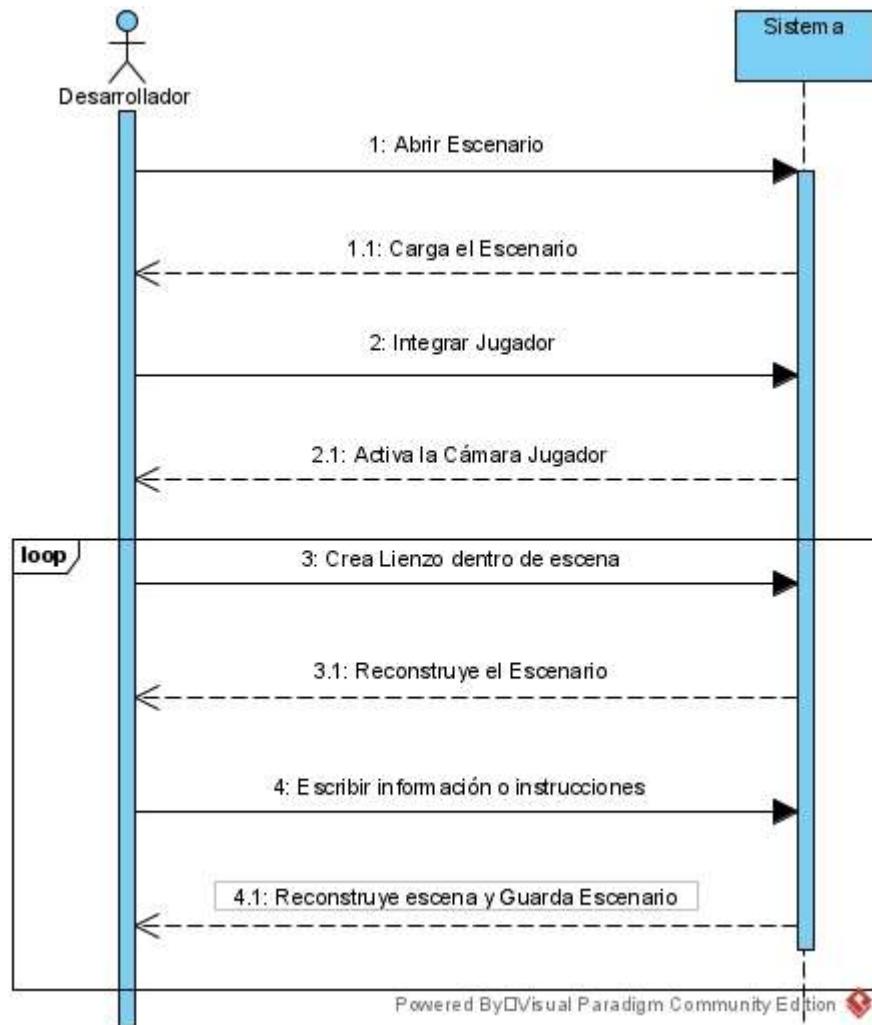


Figura 23: Diagrama de Secuencia - Creación e Interfaz de Usuario⁴⁸

3.2.3. Storyboard

Para tener una idea fresca y contextualizada del propósito del software a implementar, se elaboró un *Storyboard*, el cual demuestra de manera básica las situaciones en las cuales se utilizarán las aplicaciones desarrolladas, como se le presenta la tecnología al usuario (ver Figura 24) y las posibilidades que tiene el usuario dentro del mundo virtual construido (ver Figura 25).

⁴⁸ Elaboración propia

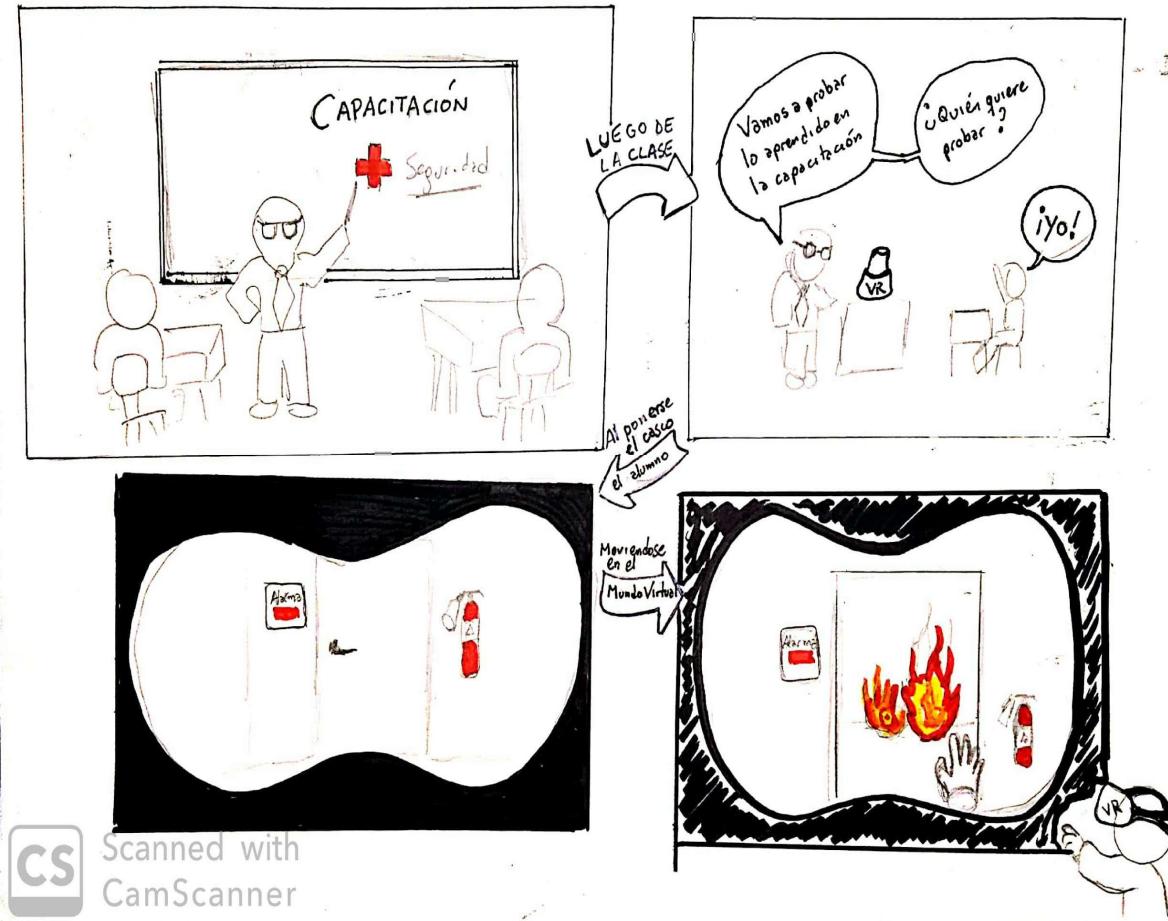


Figura 24: *Storyboard - Presentación de la Simulación*⁴⁹

⁴⁹ Elaboración propia

OPCIONES/SALIDAS

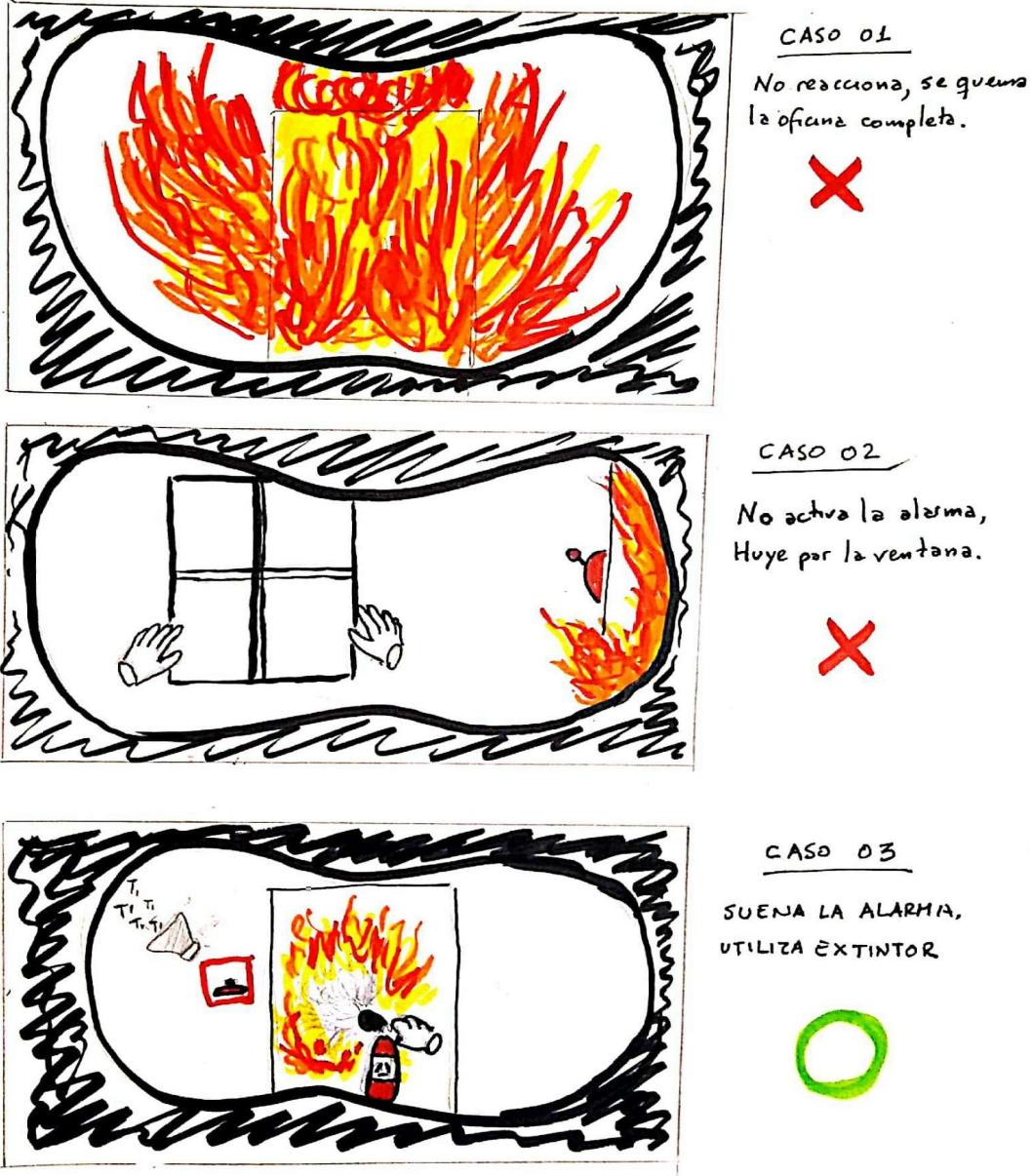


Figura 25: *Storyboard* - Resultados posibles de la simulación⁵⁰

⁵⁰ Elaboración propia

3.3. Implementación

Esta sección muestra el proceso de desarrollo del software, describiendo brevemente la configuración previa del entorno de trabajo, el uso de Unity, y la configuración necesaria para poder compilar la aplicación para el Hardware específico.

3.3.1 Configuración entorno de Desarrollo

En primer lugar, es necesario instalar la aplicación de Unity, para la fecha en que se desarrolló el software, era necesario instalar manualmente la versión de Unity, pero hoy en día, se puede utilizar el nuevo administrador de Unity, Unity Hub (Unity Technologies, s.f.), y lo único que hay que tener en cuenta al momento de instalar Unity, es preocuparse e instalar los componentes para desarrollar en Android (ver Figura 26), puesto que tanto el Smartphone como los cascos *Oculus Quest* utilizan el mismo formato de aplicaciones (plataforma Android).

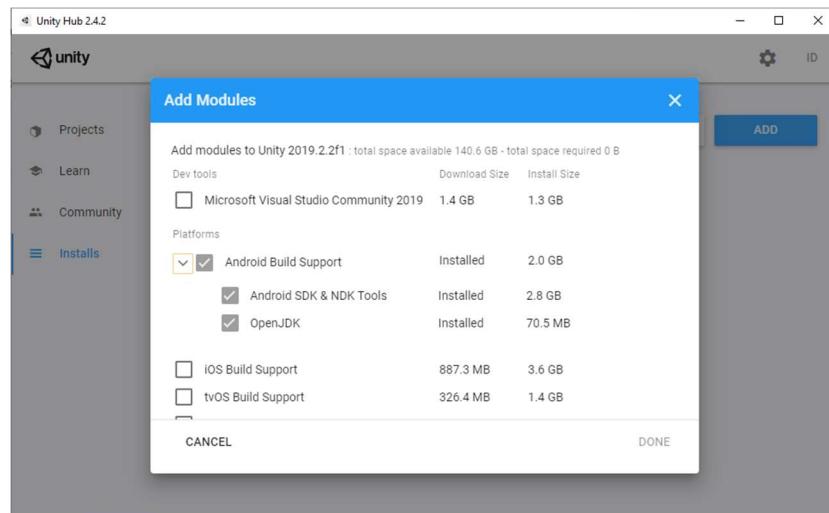


Figura 26: Módulos necesarios en instalación de Unity⁵¹

Luego de instalar Unity, hay que crear un proyecto 3D nuevo, y una vez dentro del proyecto hay que instalar los paquetes de integración con los sistemas de RV, ya sea Oculus o Google VR (en la Figura 27 se puede apreciar cómo se importa un paquete).

⁵¹ Elaboración propia

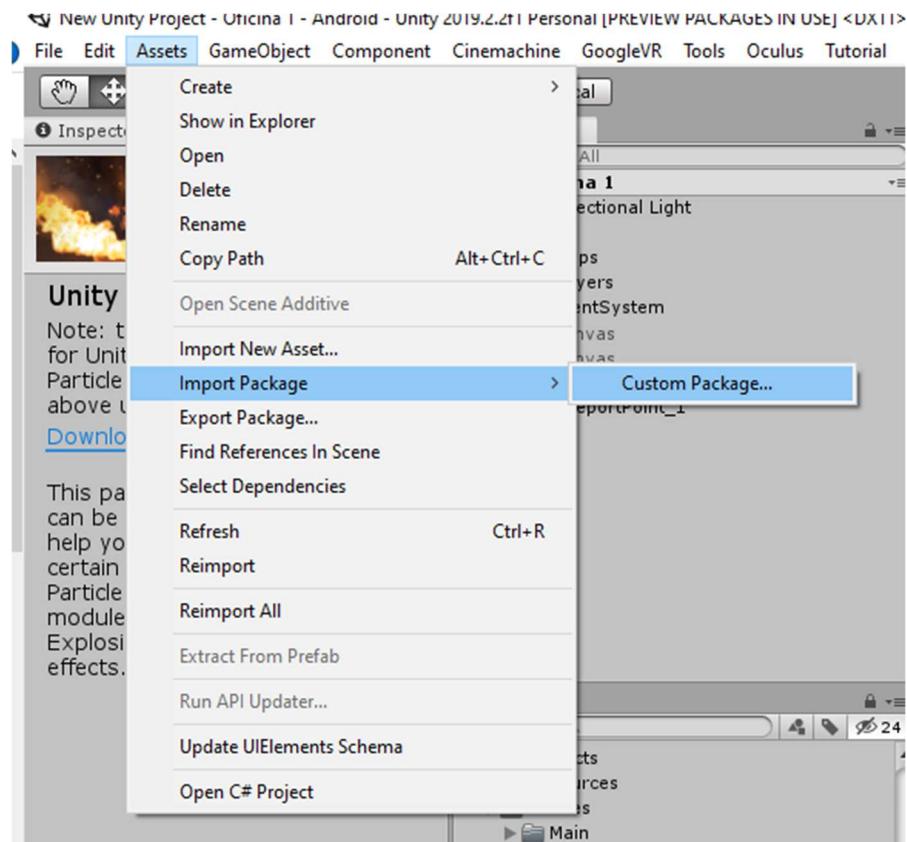


Figura 27: Como importar paquete externo en proyecto⁵²

Ahora se pueden diseñar escenas para construir una aplicación para Oculus o Google VR.

3.3.2. Pruebas Unitarias

Como el software se desarrolla en el motor de videojuegos Unity, no se presenta la necesidad de pruebas unitarias, puesto que, ante cualquier error, ya sea sintáctico o de otra índole, Unity le informa al desarrollador oportunamente, e incluso en caso de ser un Error (ver Figura 28), no simplemente una Advertencia, no deja compilar ni ejecutar la simulación de la escena.

⁵² Elaboración propia

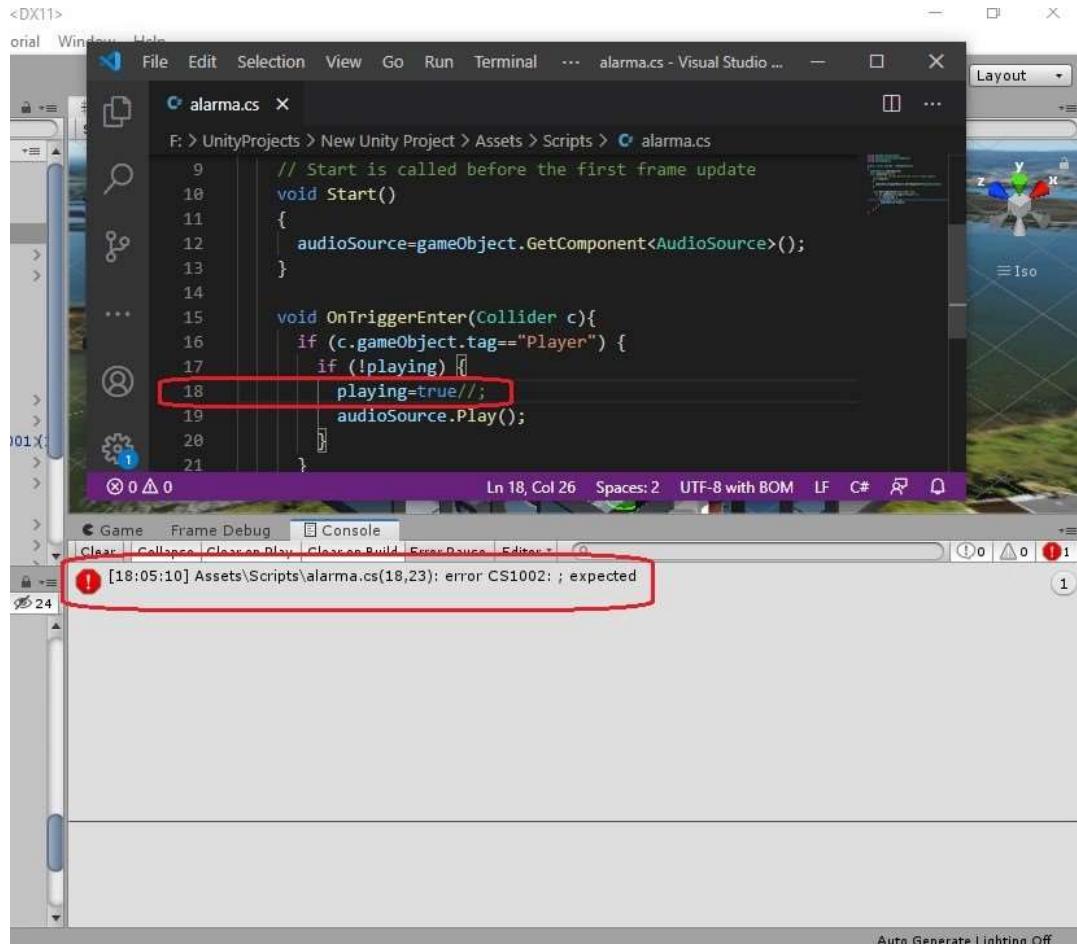


Figura 28: Error de sintaxis detectado por Unity⁵³

3.3.3. Aplicación

En las Figura 29, Figura 30 y Figura 31 se pueden identificar las secciones del Entorno de Desarrollo:

1. Jerarquía de la escena: despliega la lista de objetos presentes en la escena
2. Vista de la Escena: muestra los objetos de la escena y su configuración
3. Vista de Juego: muestra la escena emulando cómo se vería la escena al jugar.
4. Inspector de Componentes: despliega la información del objeto seleccionado, tal como el nombre, si está activo o no, posición y rotación, y otros componentes disponibles.
5. Explorador de Proyecto: muestra todos los objetos asociados al proyecto (denominados *Assets*)
6. Consola: despliega mensajes definidos por el usuario, advertencias y alertas de errores concernientes a la compilación y ejecución de una o más escenas.
7. *Asset Store*: Sitio donde se pueden comprar, descargar e importar *Assets* diseñados por la comunidad, y con funcionamiento verificado, directamente al proyecto.

⁵³ Elaboración propia

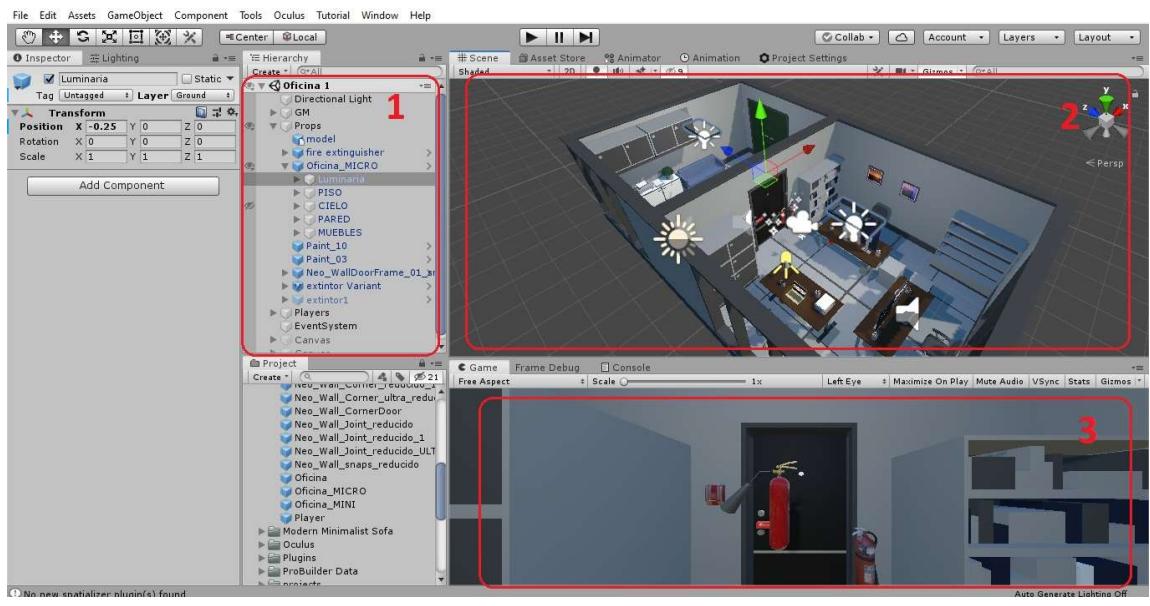


Figura 29: Secciones de Editor Unity (parte 1)⁵⁴

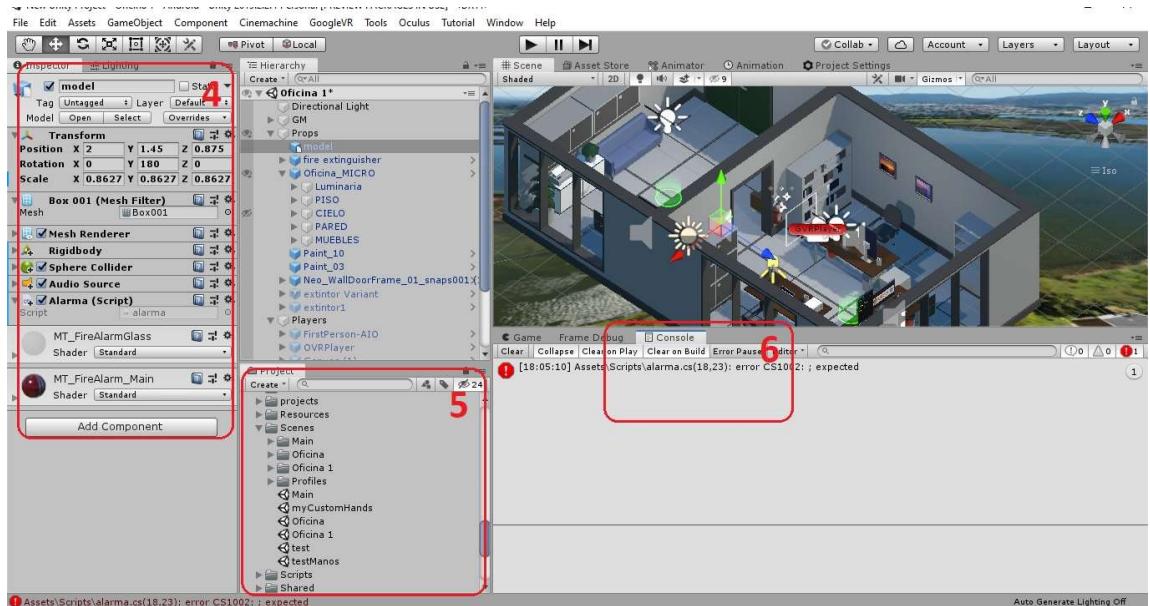


Figura 30: Secciones de Editor Unity (parte 2)⁵⁵

⁵⁴ Elaboración propia

⁵⁵ Elaboración propia

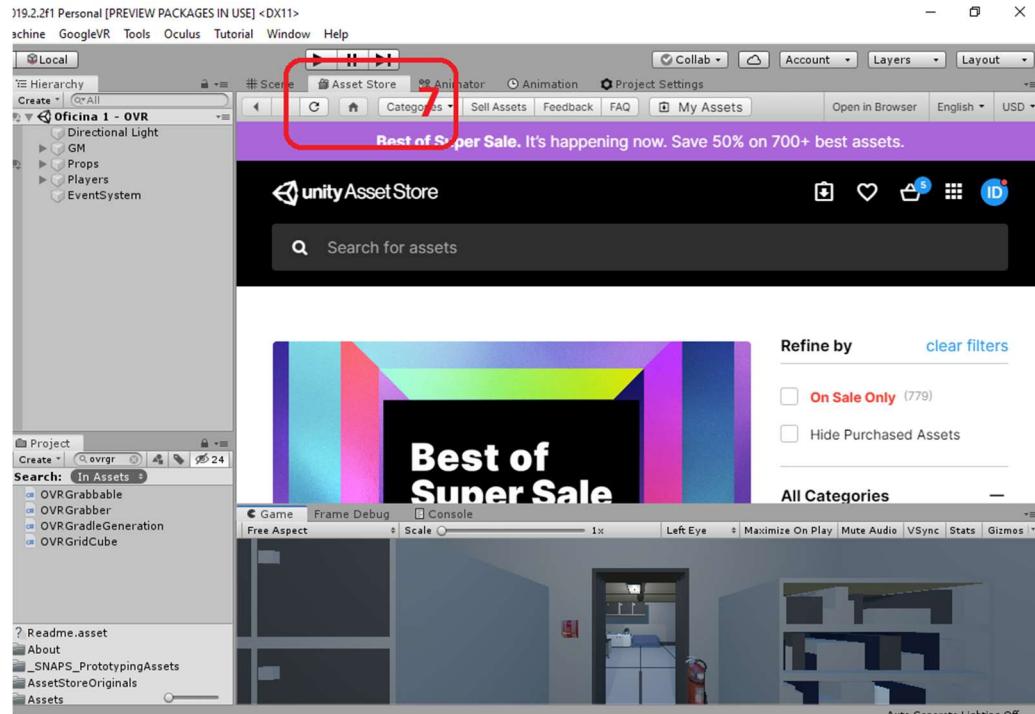


Figura 31: Secciones de Editor Unity (parte 3)⁵⁶

3.3.3.1. Construcción de una escena

La disposición de los objetos dentro de la escena será la disposición de los objetos en el Simulador final (ver Figura 32), esto incluye, pero no se limita a: objetos 3D (cubos, esferas, etc.), objetos 2D (planos, círculos, etc.), partículas (efectos visuales), entre otros.

Los objetos que forman la escena pueden construirse en base a objetos primitivos (cubos, esferas y otros por el estilo), o pueden descargarse o importarse tal como se hizo con los paquetes de Oculus Integration y SDK Google VR, o desde la *Asset Store*.

Un objeto extremadamente importante es el objeto conocido como ‘*Game Manager*’ (o GM, como se ve en la Figura 33), este objeto, por lo general sin forma visible, se encargará de controlar los comportamientos generales de la simulación, que hayan sido definidos por el desarrollador. Si bien se puede trabajar sin el GM, este es muy importante para el desarrollo usando el SDK de Google VR.

Una vez construido el espacio de la simulación, se prosigue al desarrollo de la aplicación orientado a las plataformas objetivo.

⁵⁶ Elaboración propia

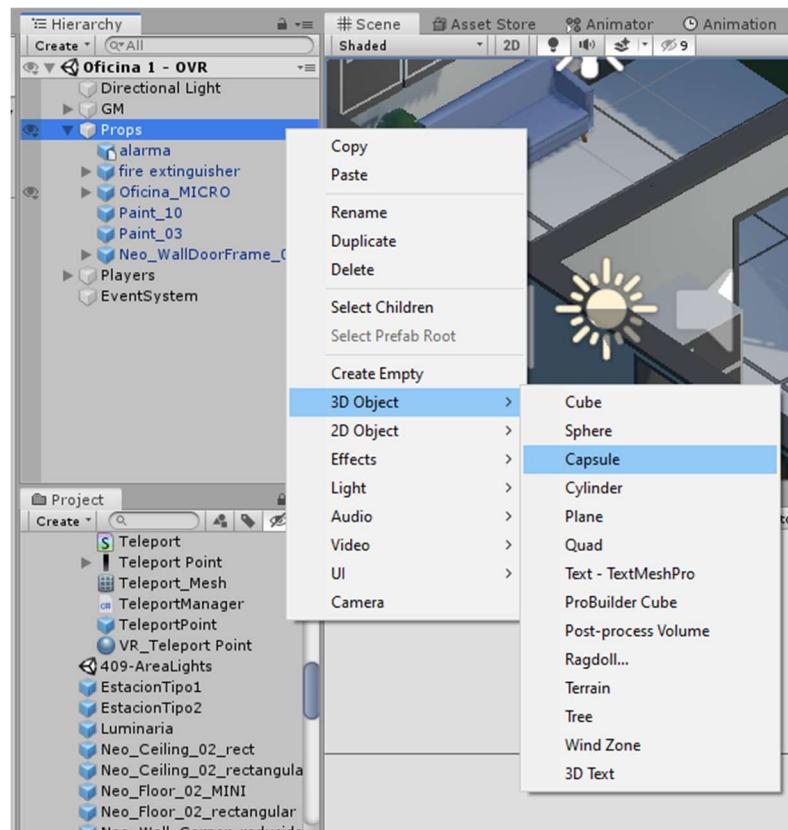


Figura 32: Objetos en escena de Unity⁵⁷

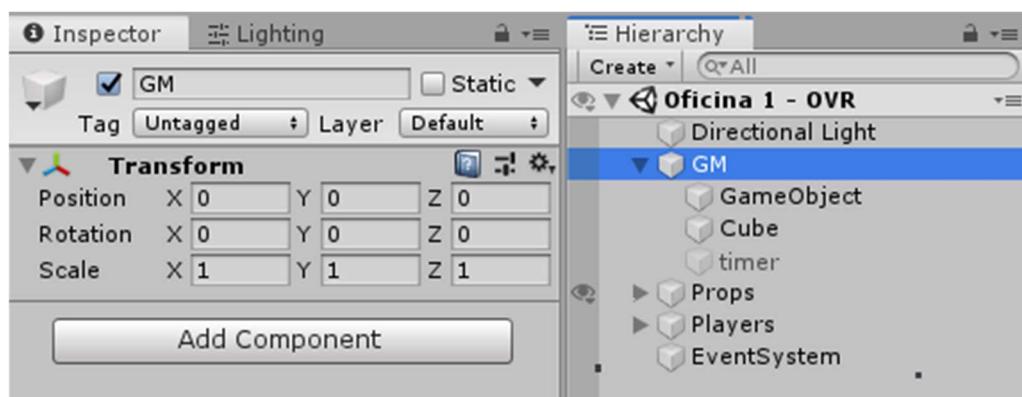


Figura 33: Objeto *Game Manager* (GM) dentro de escena⁵⁸

⁵⁷ Elaboración propia

⁵⁸ Elaboración propia

3.3.3.2 Desarrollo orientado a Google VR

El primer objeto especializado construido es el Jugador (GVR en Figura 34), objeto construido a partir de los objetos presentes en las escenas demostrativas que vienen en el paquete Google VR.

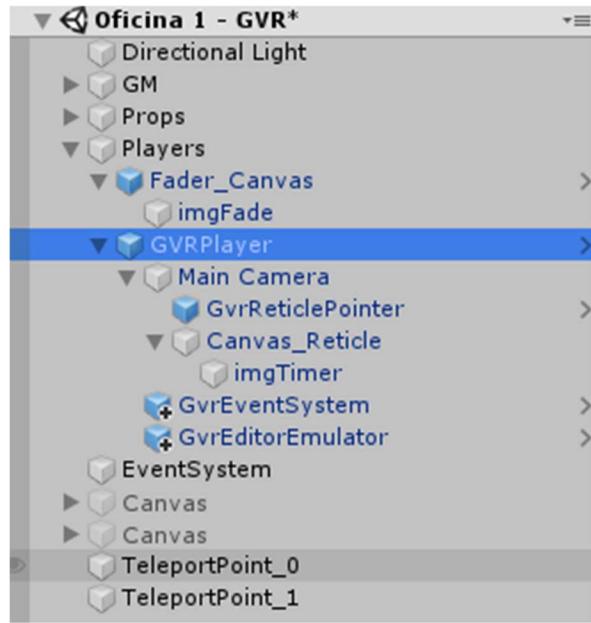


Figura 34: Jerarquía de Objetos para Jugador (GVRPlayer)⁵⁹

Una vez definido el Jugador, se agrega a la escena construida, y se puede definir el comportamiento de los objetos a utilizar en la escena.

Ahora que está presente el objeto Jugador, se crean dos Scripts que se agregarán a GM. El primer Script es *ReticleTimer* (ver Figura 35 y Anexo C), el cual se encarga de controlar los objetos que son apuntados por el Jugador, y que acciones se desencadenan después de haber sido ‘activados’. El segundo Script es *TeleportManager* (ver Figura 36 y Anexo D), que se preocupa de manejar el proceso de teletransporte a implementarse posteriormente, realizando una transición suave que no incomode al usuario.

⁵⁹ Elaboración propia

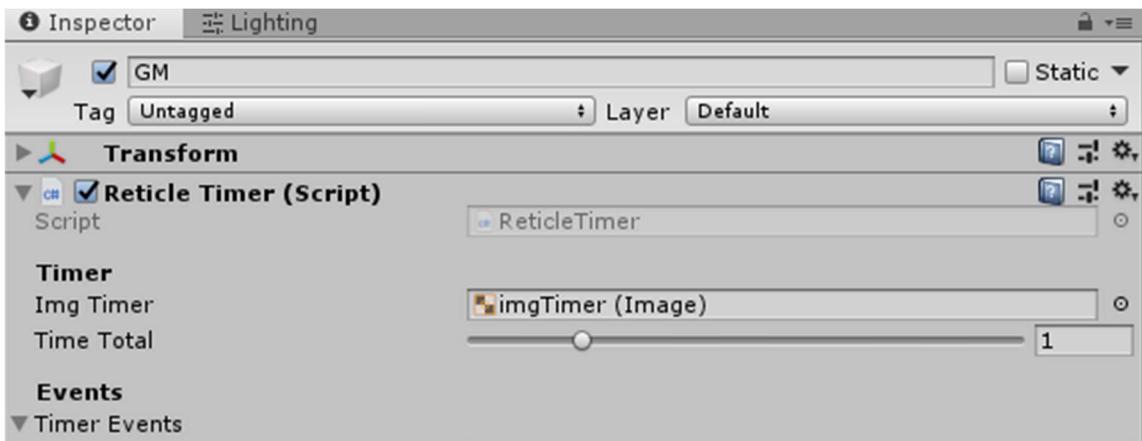


Figura 35: Uso de *ReticleTimer* dentro de GM⁶⁰

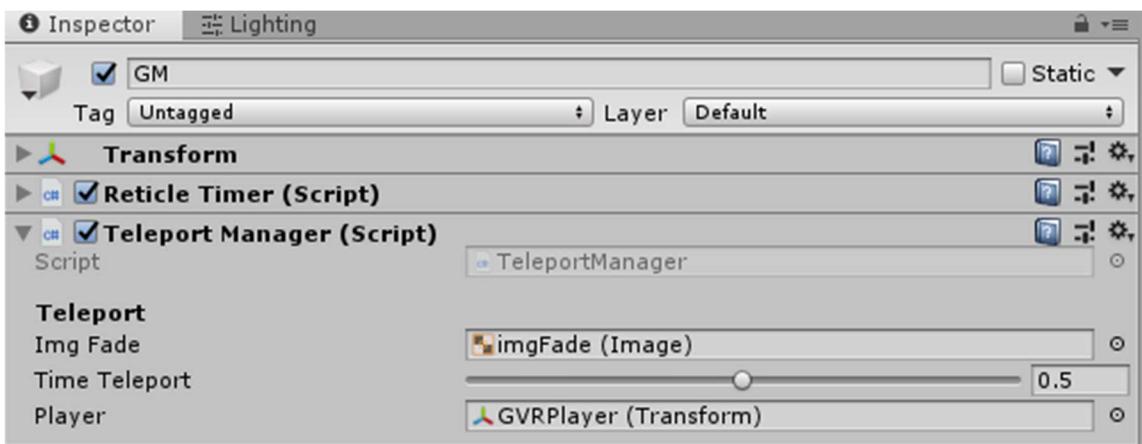


Figura 36: Uso de *TeleportManager* dentro de GM⁶¹

Ya estando construido e implementado *ReticleManager*, se pueden construir los objetos designados como punto de transporte, nombrados *TeleportPoint*. Estos objetos, guardan las coordenadas para posicionar el Jugador una vez son completamente activados (ver Anexo E).

Luego de eso, se agrega un *Event Trigger* a este objeto, con los tipos de evento *Pointer Enter* y *Pointer Exit*, a cada uno de ellos se le agrega un elemento, asociando el GM. En el caso de *Pointer Enter*, se asocia con el *ReticleTimer.Timer_Enter* presente en GM, indicando posteriormente el ID que se le otorgará al objeto. En cambio, para *Pointer Exit*, se asocia con *ReticleTimer.Timer_Exit*, tal como se muestra en la Figura 37.

⁶⁰ Elaboración propia

⁶¹ Elaboración propia

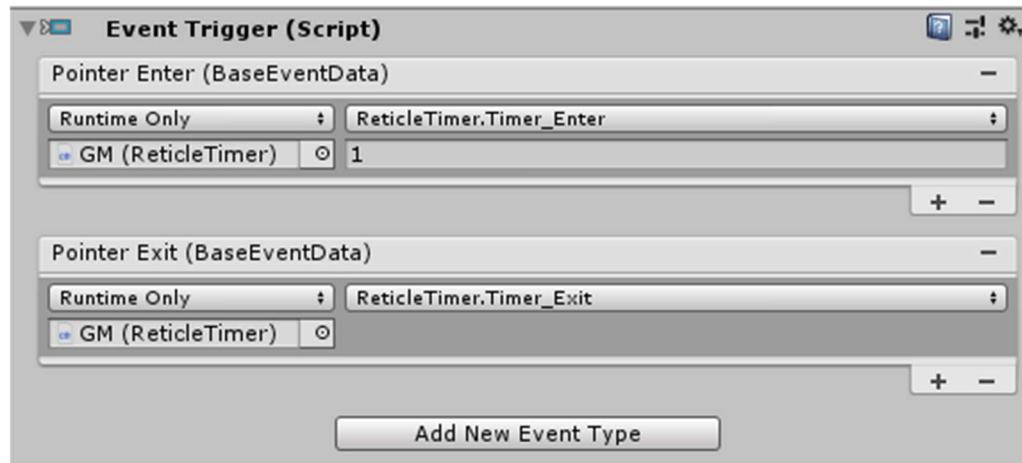


Figura 37: Event Trigger de objeto *Teleport Point*⁶²

Este mismo procedimiento se puede aplicar para otros objetos con distintas funcionalidades, siempre teniendo en cuenta el ID a inscribir en *Pointer Enter*. Tal como la alarma dentro de la escena (ver Figura 38).

Una vez configurados los objetos interactivos, se procede a declarar los objetos dentro de *ReticleTimer* dentro de GM, y la función a gatillar (ver Figura 39).



Figura 38: Componentes objeto alarma⁶³

⁶² Elaboración propia

⁶³ Elaboración propia

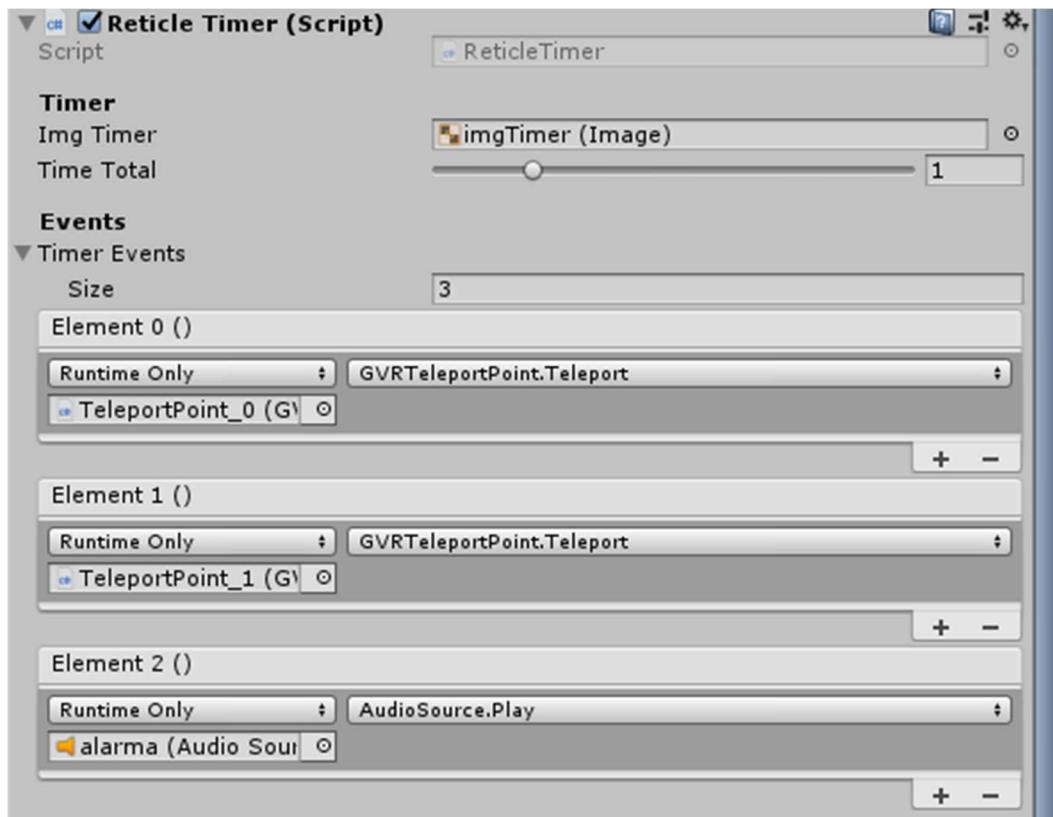


Figura 39: Configuración de elementos en *ReticleTimer*, objeto GM⁶⁴

3.3.3.3 Desarrollo orientado a *Oculus Quest*

Al igual que en el desarrollo para Google VR, el primer objeto a construir es el Jugador (*Player* en la Figura 40), igualmente extraído de una escena demostrativa. Ahora con ese objeto construido se pudo importar en la escena y agregar características a los objetos de la escena.

Los componentes (específicamente de tipo Script) más importantes a utilizar en este caso son: *OVRGrabber* y *OVRGrabbable* (ver Figura 41 y Figura 42), el primero para definir que el controlador (mano del jugador) puede agarrar objetos, y el segundo, para definir que un objeto es ‘agarrable’, y la posición en la cual se sostendrá.

⁶⁴ Elaboración propia

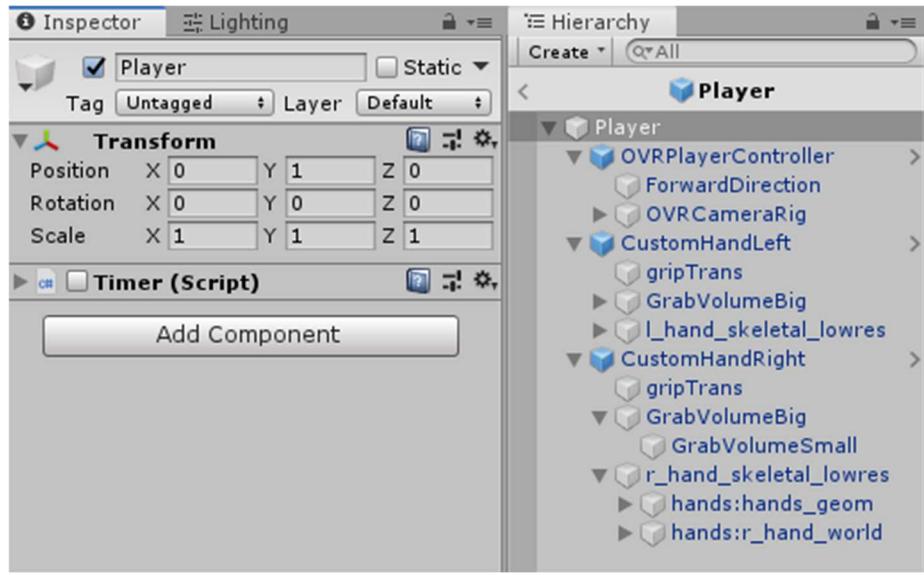


Figura 40: Composición del Objeto Jugador para Oculus VR (Player)⁶⁵

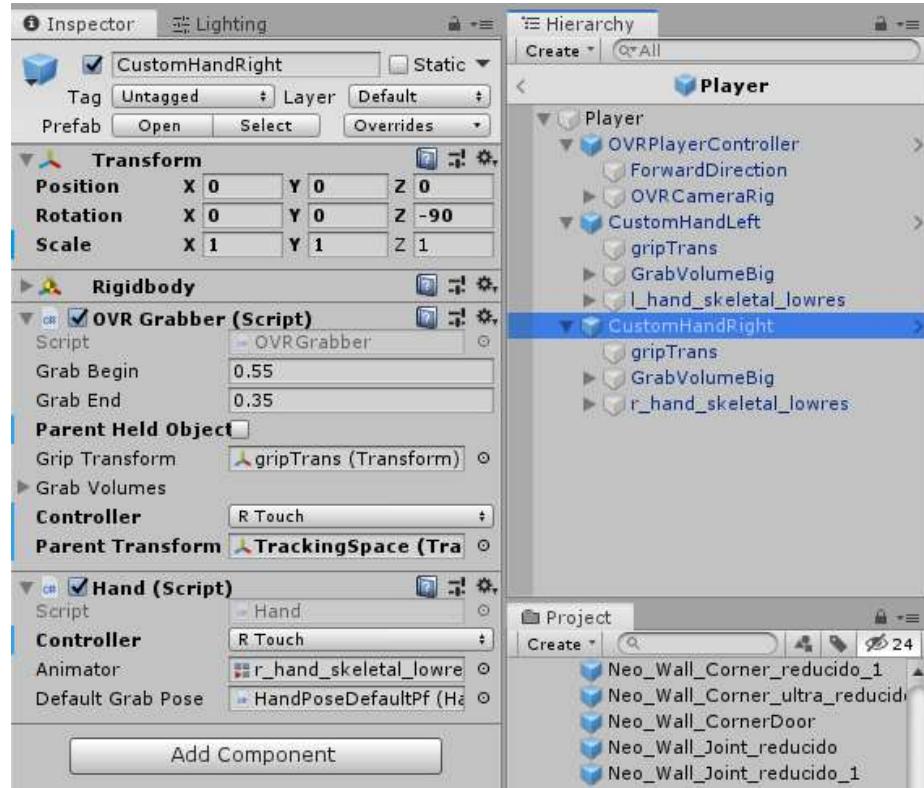


Figura 41: Configuración de componente *OVRGrabber*⁶⁶

⁶⁵ Elaboración propia

⁶⁶ Elaboración propia

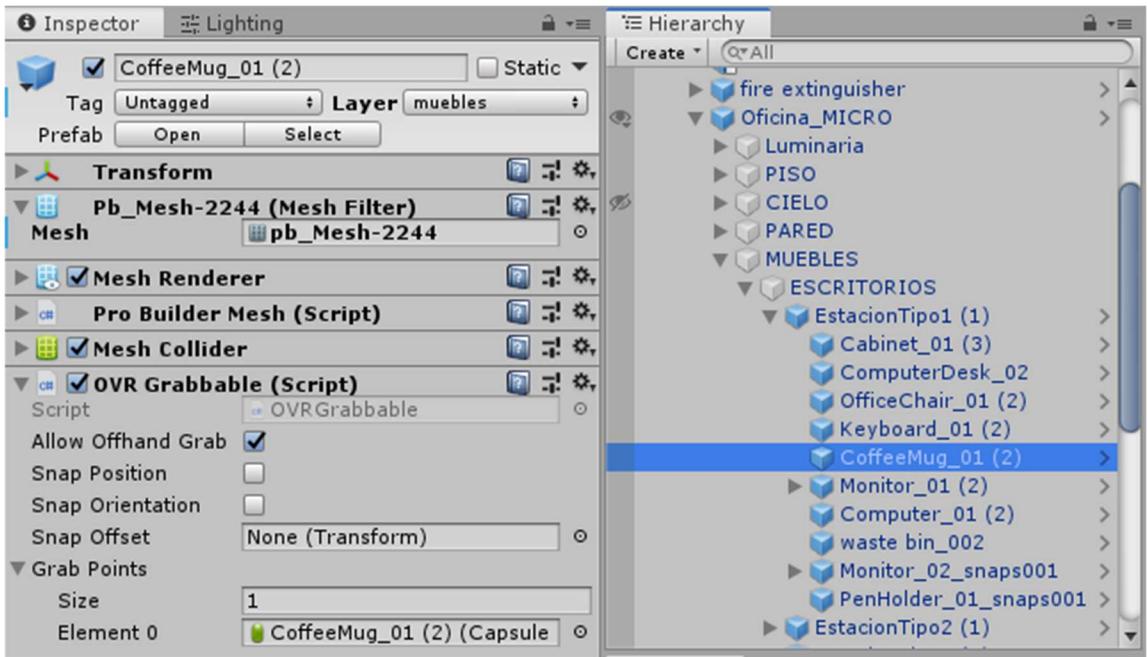


Figura 42: Objeto usando componente *OVRGrabbable*⁶⁷

Especialmente se debió modificar el Script *OVRGrabber* para poder implementar una característica que no viene definida en el código original, la habilidad de pulsar o activar el objeto sostenido. Se agregó la siguiente función al Script:

```
public OVRInput.Controller GetController() {
    return m_controller;
}
```

Tal como se puede apreciar en la Figura 43.

Una vez agregada esa porción de código se pudo construir el Script para accionar una característica del objeto agarrado. Para la simulación construida, se aplicó en el Script Extintor, que como su nombre indica es para la activación de un extintor (ver Anexo F, Figura 44 y Figura 45).

⁶⁷ Elaboración propia

```

52     protected bool m_grabVolumeEnabled = true;
53     protected Vector3 m_lastPos;
54     protected Quaternion m_lastRot;
55     protected Quaternion m_anchorOffsetRotation;
56     protected Vector3 m_anchorOffsetPosition;
57     protected float m_prevFlex;
58     protected OVRGrabbable m_grabbedObj = null;
59     protected Vector3 m_grabbedObjectPosOff;
60     protected Quaternion m_grabbedObjectRotOff;
61     protected Dictionary<OVRGrabbable, int> m_grabCandidates = new Dictionary<OVRGrabbable, int>();
62     protected bool operatingWithoutOVRCameraRig = true;
63
64     /// <summary>
65     /// The currently grabbed object.
66     /// </summary>
67     public OVRGrabbable grabbedObject
68     {
69         get { return m_grabbedObj; }
70     }
71
72     public OVRInput.Controller GetController()
73     {
74         return m_controller;
75     }
76
77     public void ForceRelease(OVRGrabbable grabbable)
78     {
79         ...
80     }

```

Figura 43: Modificación de Código en Script *OVRGrabber*⁶⁸

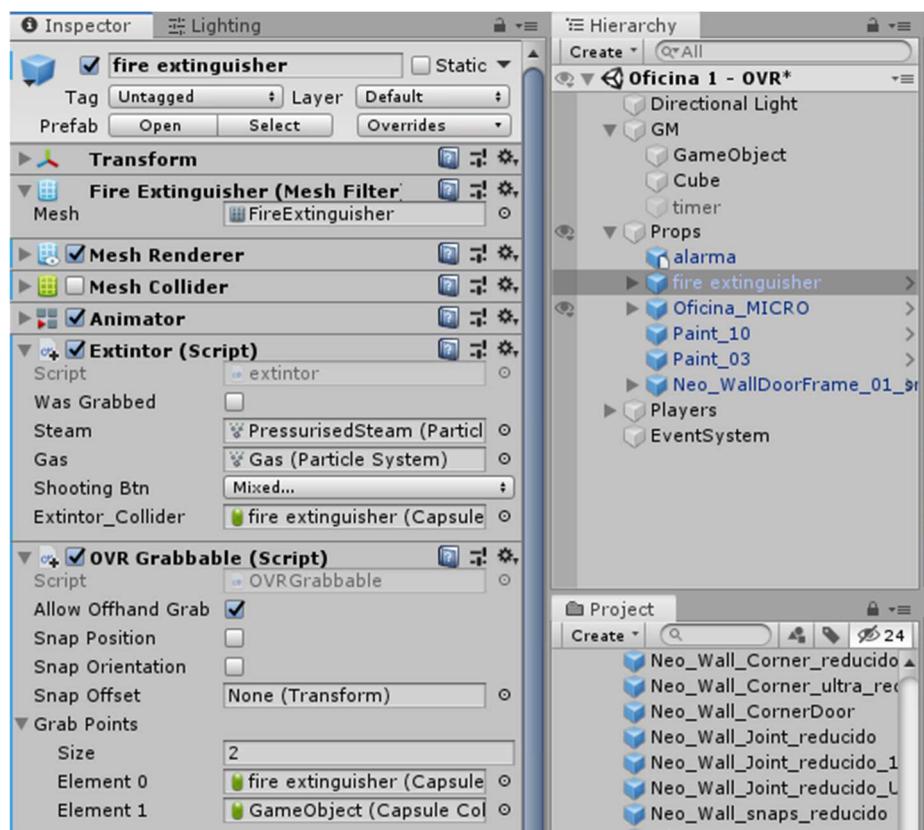


Figura 44: Objeto usando Scripts *OVRGrabbable* y *Extintor*⁶⁹

⁶⁸ Elaboración propia

⁶⁹ Elaboración propia



Figura 45: Aplicación de Scripts *OVRGrabbable* y *Extintor*⁷⁰

3.3.4. Compilar Aplicación

Para poder compilar la aplicación desarrollada hay que definir ciertas configuraciones de la aplicación:

1. En ventana ‘*Build Settings*’ (ver Figura 46), definir las escenas a integrar en la aplicación, la plataforma objetivo, y sus configuraciones.
2. En la misma ventana, pulsar el botón ‘*Player Settings*’ para acceder a dicha configuración y poder aplicar las configuraciones descritas en Figura 47 y Figura 48.
3. Luego de dichas modificaciones, se vuelve a la ventana ‘*Build Settings*’ para poder finalmente construir la aplicación.

⁷⁰ Elaboración propia

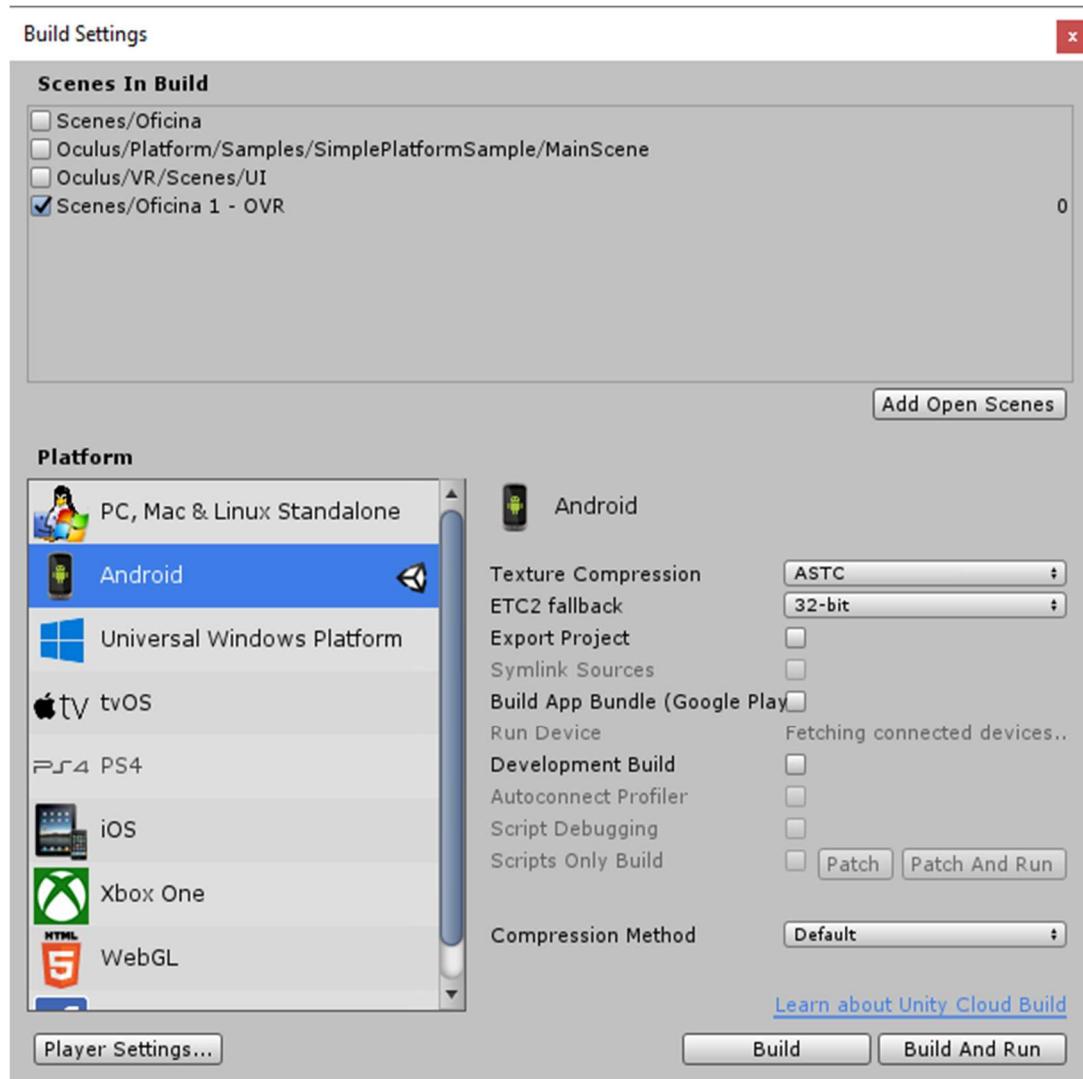


Figura 46: Ventana *Build Settings* y configuración preferida para compilación⁷¹

⁷¹ Elaboración propia

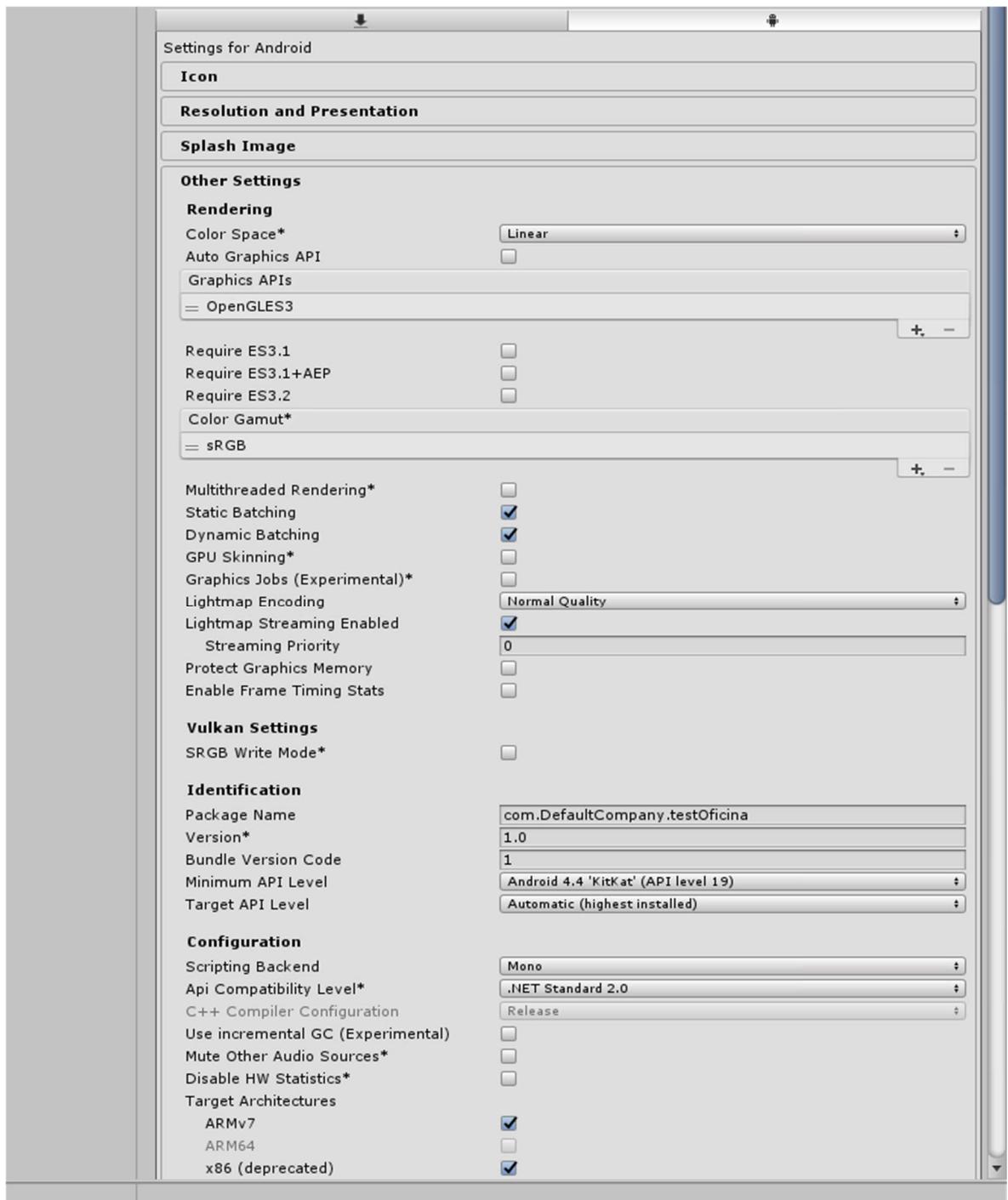


Figura 47: *Player Settings* configurado para VR en Android (parte 1)⁷²

⁷² Elaboración propia

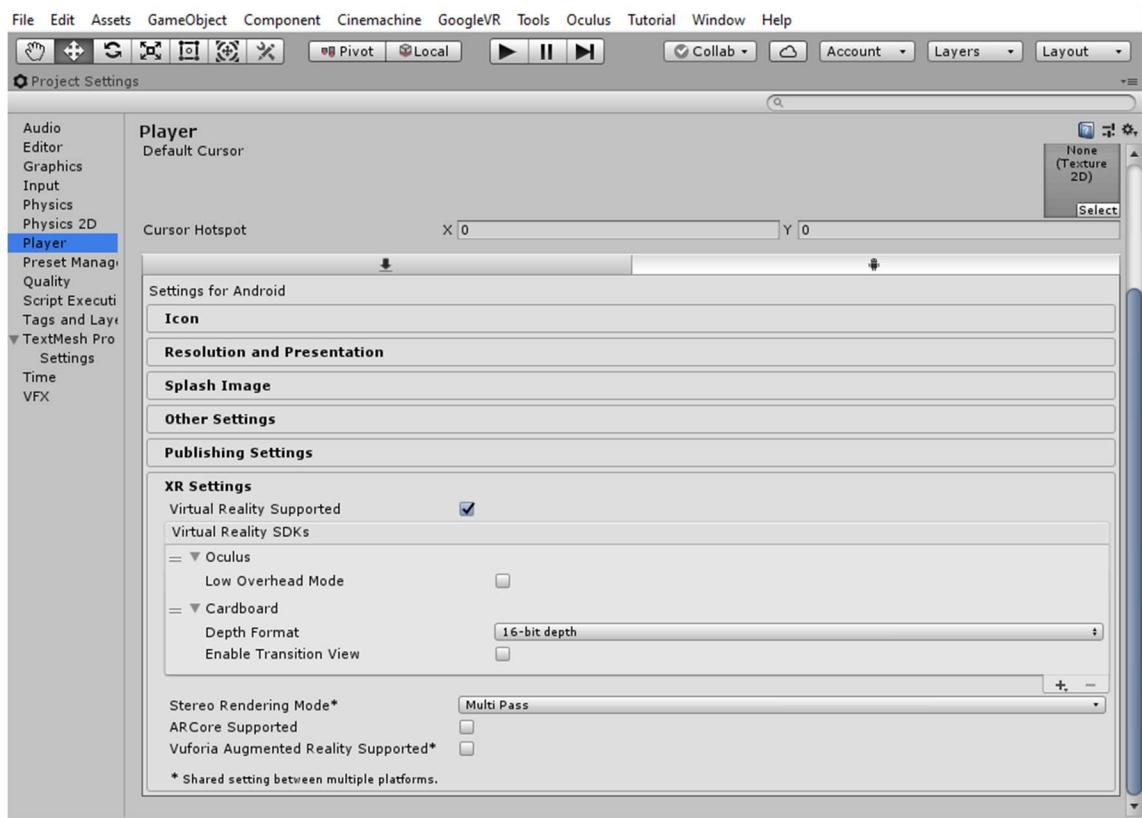


Figura 48: *Player Settings* configurado para VR en Android (parte 2)⁷³

Para que los desarrolladores no tengan que pasar por todos estos procedimientos, se han recopilado los materiales utilizados (paquetes Oculus Integration y GoogleVR SDK), además de paquetes creados aplicando las modificaciones realizadas a los códigos fuente a un repositorio alojado en la plataforma GitHub⁷⁴ junto con el proyecto completo de Unity realizado, dándoles a los desarrolladores la libertad de utilizar las herramientas que consideren convenientes.

⁷³ Elaboración propia

⁷⁴ https://github.com/iadr/simulador_VR.git

4. EVALUACIÓN

En este capítulo se describe el proceso de validación del proyecto como un simulador para la capacitación de personas ante situaciones de emergencias.

4.1. Instrumento de análisis

Para evaluar el prototipo, se opta por el uso de dos encuestas anónimas:

- la primera para conocer la percepción de la comunidad de desarrolladores de videojuegos, quienes están más experimentados en el campo y las tecnologías utilizadas (ver Anexo G), y
- una segunda encuesta, orientada a consultarle al público en general su punto de vista acerca de la simulación, su realismo y su efectividad (ver Anexo H).

Ambas encuestas se realizaron utilizando grabaciones de las experiencias dentro de la aplicación desarrollada para *Oculus Quest*.

4.2. Percepción del usuario desarrollador

Se encuestaron 20 profesores y estudiantes de la carrera Técnico en Diseño de Videojuegos⁷⁵ de la Universidad Santo Tomás, sede Valdivia, con el objeto de sondear el grupo de profesionales capacitado para desarrollar los escenarios a simular y, además obtener retroalimentación del proyecto desarrollado.

4.2.1. Resultados

Para la primera pregunta, enfocada a sondear el uso de dispositivos de Realidad Virtual, un 75% de los encuestados respondieron afirmativamente (ver Figura 49).

Los resultados de la segunda pregunta indican que un 85% de los desarrolladores han trabajado con Realidad Virtual (ver Figura 50).

La tercera pregunta, orientada a determinar las preferencias de los desarrolladores, rescató que los encuestados, a nivel de Software, prefieren el trabajo en Unity y Unreal Engine, mientras que, a nivel de Hardware, optan por los dispositivos Oculus.

La cuarta pregunta, enfocada en saber la disposición del encuestado a usar un marco de trabajo estandarizado, resultó en un 90% de respuestas positivas (ver Figura 51).

⁷⁵ <https://www.tupuedes.cl/carreras/instituto-profesional/diseno-de-videojuegos/>

¿Has probado alguna vez unas gafas de Realidad Virtual?
20 respuestas

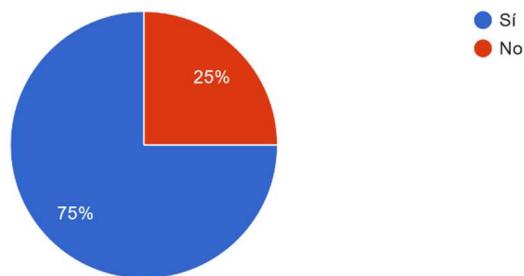


Figura 49: Gráfica para Experiencia con Realidad Virtual⁷⁶

¿Has desarrollado en Realidad Virtual?
20 respuestas

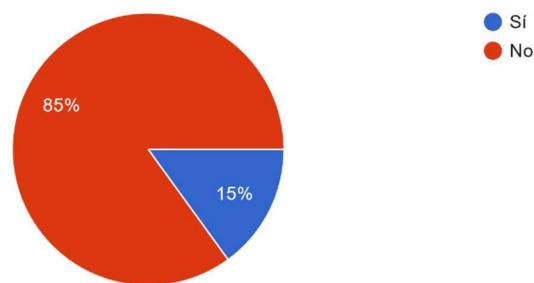


Figura 50: Gráfica para Experiencia en Desarrollo en Realidad Virtual⁷⁷

¿Utilizarías un marco de trabajo que estandarice el desarrollo en Realidad Virtual?
20 respuestas

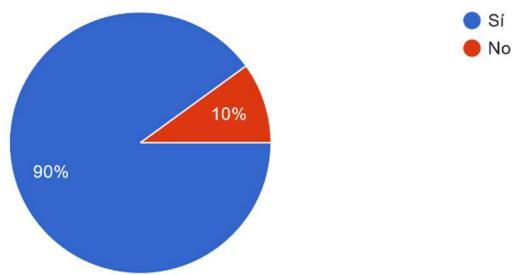


Figura 51: Gráfica para Aceptación de Marco de Trabajo⁷⁸

⁷⁶ Elaboración propia

⁷⁷ Elaboración propia

⁷⁸ Elaboración propia

Para la quinta pregunta se les pidió analizar un video (ver Figura 52 y Figura 53) capturando la experiencia de un usuario en el escenario construido en el capítulo anterior, y proponer ideas para mejorar el escenario diseñado. Las respuestas se enfocaron en el realismo del escenario, en sacar provecho del sonido ambiente, efectos visuales que refuerzen el comportamiento esperado, y la preocupación por la fluidez de la aplicación y con ello la salud del usuario.



Figura 52: Captura del video presentado en encuesta (1)



Figura 53: Captura del video presentado en encuesta (2)

Para la última pregunta, solicitaba ideas de escenarios para probar las reacciones de los usuarios, los encuestados se tomaron la libertad de plantear diversos escenarios, entre los cuales se encuentran: terremotos, atentados terroristas, avería de ascensores, pruebas de manejo, entre otros.

4.3. Percepción de la experiencia.

Se consultó a 19 personas anónimas, a los cuales se les mostró un par de grabaciones de lo experimentado por un usuario, para luego evaluar la reacción del usuario y si era posible realizar el procedimiento de seguridad correcto.

4.3.1. Resultados

Ante la primera pregunta ‘¿el escenario permite identificar la situación de emergencia?’, un 94.7% de los encuestados respondió positivamente (ver Figura 54).

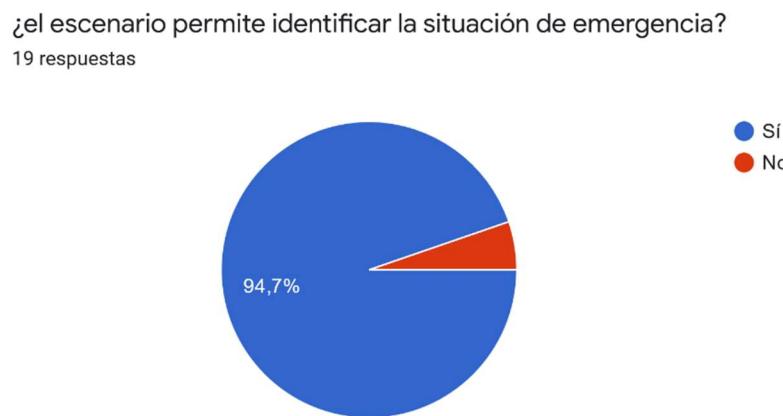


Figura 54: Gráfica para Identificación de Situación de Emergencia⁷⁹

Para la segunda pregunta, ‘¿el escenario tiene todos los elementos necesarios para tomar las decisiones?’, el 78.9% respondió afirmativamente (ver Figura 55), considerando que los elementos dispuestos dentro del escenario son los suficientes para poder decidir las acciones a tomar correctas.

⁷⁹ Elaboración propia

¿el escenario tiene todos los elementos necesarios para tomar las decisiones?
19 respuestas

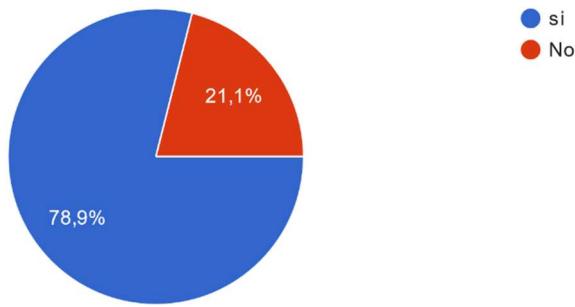


Figura 55: Gráfica para Elementos Suficientes en Escenario⁸⁰

La última pregunta, ¿el escenario permitió realizar acciones de acuerdo al protocolo esperado?, enfocada en la disposición de los objetos en la simulación, y la facilidad para poder reaccionar ante los protocolos correctos, o los más cercanos a los definidos por las empresas de seguridad, un 84.2% de los encuestados respondió afirmativamente (ver Figura 56).

¿El escenario permitió realizar acciones de acuerdo al protocolo esperado?
19 respuestas

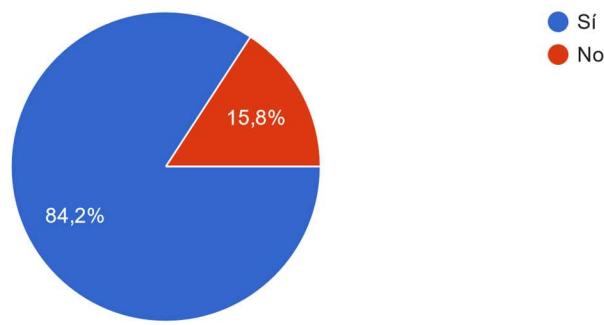


Figura 56: Gráfica para Realización de Protocolo Esperado⁸¹

⁸⁰ Elaboración propia

⁸¹ Elaboración propia

5. CONCLUSIONES

El desarrollo en Realidad Virtual ha tenido un camino largo y “pedregoso” con altos y bajos, múltiples intentos y fallos, pero, aun así, ha salido adelante, la inversión de grandes compañías en esta tecnología y otras similares ha ayudado a generar una comunidad que crece cada día más.

El desarrollo de este proyecto ha demostrado ser una aventura, puesto que toma en cuenta material que no se considera por lo general al momento de un desarrollo estándar. El uso de herramientas con un enfoque distinto como el desarrollo de videojuegos, en vez de plataformas empresariales o servicios web requiere de un cambio de mentalidad al momento de efectuar el trabajo.

El trabajo permitió evaluar 3 tecnologías, Google VR, Unity y Unreal Engine, utilizadas hoy en día en la implementación de Realidad Virtual basado en la literatura e identificando las mejores técnicas que se pueden aplicar al desarrollo de este tipo de tecnologías con el objeto de implementar soluciones de simulación.

La investigación permitió comprender que un desarrollo de *testing* en Ingeniería de Software es un módulo funcional en el cual se pueden realizar pruebas de evaluación, y en este caso se pudo desarrollar un prototipo, el cual consistió en la simulación del escenario de prueba sobre el que se pudieron validar no solo temas funcionales de la aplicación, sino algunos otros también de percepción y posibles aplicaciones que son de real interés para desarrollos futuros.

Gracias al diseño, basado en técnicas como UML⁸², en el cual la identificación de actores y posteriormente la elaboración de los diagramas de secuencia, permitieron identificar claramente los procesos que involucran el desarrollo de este tipo de tecnologías, gracias a lo cual se logran identificar indicadores que pueden homologarse a métricas: el correcto desplazamiento del usuario en el espacio virtual, procurar evitar la sobre estimulación visual, el realismo del entorno diseñado dependiendo de la finalidad de la simulación, entre otros.

Gracias a poder acceder no solo a una plataforma basada en Android (Smartphone), sino que al participar en el proyecto en conjunto con la empresa LEK, se tuvo acceso a cascos de última generación, tal como el *Oculus Quest*, el cual permitió implementar la simulación, y realizar pruebas con usuarios finales, y finalmente realizar cuestionarios para evaluar la experiencia, no solo de usabilidad sino de posibles facilidades para el desarrollo para este tipo de herramientas, considerando siempre este escenario de *testing* creado como un entorno parte del proyecto de título.

Aunque es cierto que se lograron hacer pruebas con usuarios finales durante la fase de implementación, nunca fue de manera formal, y para el momento en que era posible evaluar el prototipo final, la situación social que ha afectado al país en 2020 no permitió

⁸² *Unified Modeling Language*, Lenguaje de Modelado Unificado

realizar una evaluación formal, por lo que se debió recurrir a formas de validación indirectas, como las encuestas realizadas.

Este proyecto de título es solo un prototipo para un sistema mucho más grande y mejor, para poder generar escenarios simulados en Realidad Virtual. El prototipo desarrollado es el punto de partida, y se plantea que en futuras implementaciones se incluyan complementos estéticos, e integración con otros dispositivos de Realidad Virtual.

6. REFERENCIAS

- Anderson, P. L., Zimand, E., Hodges, L. F., & Rothbaum, B. O. (2005). Cognitive behavioral therapy for public-speaking anxiety using virtual reality for exposure. *Depression and anxiety*, 156-158.
- Botella, C., Osma, J., Garcia-Palacios, A., Quero, S., & Baños, R. M. (2004). Treatment of flying phobia using virtual reality: data from a 1-year follow-up using a multiple baseline design. *Clinical Psychology & Psychotherapy*, 311-323.
- Caballero, A. R., & Niguidula, J. D. (2018). Disaster Risk Management and Emergency Preparedness: A Case-Driven Training Simulation Using Immersive Virtual Reality. *Proceedings of the 4th International Conference on Human-Computer Interaction and User Experience in Indonesia, CHIuXiD '18 - CHIuXiD '18* (págs. 31-37). Yogyakarta, Indonesia: ACM Press.
- catherine. (2019, Diciembre 17). *4 things to know about VR before you buy a headset*. Retrieved from Toast: <https://toast.gg/4-things-to-know-about-vr-before-you-buy-a-headset/>
- Cherdo, L. (2019, Febrero 12). *VR headset buying guide: how to choose a virtual reality HMD?* Retrieved from Aniwaa: <https://www.aniwaa.com/guide/vr-ar/vr-headset-buying-guide/>
- Chou, C., Hsu, H.-L., & Yao, Y.-S. (1997). Construction of a Virtual Reality Learning Environment for Teaching Structural Analysis. *Computer Applications in Engineering Education*, 223-230.
- El 90% de los accidentes laborales son evitables, según un informe de CCOO. (1999, Febrero 9). *El Mundo*. Retrieved Junio 26, 2020, from <https://www.elmundo.es/elmundo/1999/febrero/09/economia/siniestralidad.html>
- El 98% de los accidentes laborales son evitables, según los sindicatos. (2007, Diciembre 16). *Diario Sur*. Retrieved Junio 26, 2020, from <https://www.diariosur.es/20071216/economia/accidentes-laborales-evitables-segun-20071216.html>
- Gorman, P. J., Meier, A. H., & Krummel, T. M. (1999, Noviembre 1). Simulation and Virtual Reality in Surgical Education: Real or Unreal? *Archives of surgery*, 134(11), 1203-1208.
- Gromala, D., Tong, X., Choo, A., Karamnejad, M., & Shaw, C. D. (2015). The Virtual Meditative Walk: Virtual Reality Therapy for Chronic Pain Management. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15* (pp. 521-524). Seúl: ACM Press.
- Gu, T., Wang, C., & He, G. (2018). A VR-based, hybrid modeling approach to fire evacuation simulation. *Proceedings of the 16th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry - VRCAI '18* (pp. 1-8). Tokyo: ACM Press.
- Gurwin, G. (2020, Noviembre 23). *The best VR games*. Retrieved Noviembre 11, 2020, from Digital Trends: <https://www.digitaltrends.com/gaming/best-virtual-reality-games/>
- Halabi, O. (2020). Immersive virtual reality to enforce teaching in engineering education. *Multimedia tools and applications*, 2987-3004.

- Hartley, R., & Zisserman, A. (2003). *Multiple View Geometry in Computer Vision* (Segunda ed.). Reino Unido: Cambridge University Press.
- Heilig, M. L. (1962, Agosto 28). *EEUU Patent No. 3050870*.
- Henschel, H. (2019, Junio 14). *Google's Daydream View VR headset is at its lowest price ever on Amazon — save \$40*. Retrieved Noviembre 11, 2020, from Mashable: <https://mashable.com/shopping/deal-june-14-google-daydream-view-vr-headset-amazon/>
- Hood, V., Knapp, M., & Griliopoulos, D. (2020, Octubre 8). *Best VR games 2020: the top virtual reality games to play right now*. Retrieved Noviembre 11, 2020, from TechRadar: <https://www.techradar.com/best/the-best-vr-games>
- How did virtual reality begin?* (n.d.). Retrieved Noviembre 11, 2020, from Virtual Reality Society: <https://www.vrs.org.uk/virtual-reality/beginning.html>
- Human Factors and User Studies*. (n.d.). Retrieved Noviembre 11, 2020, from Virtual Reality Society: <https://www.vrs.org.uk/virtual-reality/human-factors-and-user-studies.html>
- Instituto Internacional de Análisis de Negocio. (2009). *Guía sobre Los fundamentos del conocimiento del Análisis de Negocio (Guía BABOK®)* (2.0 ed.). Toronto, Ontario, Canadá.
- Kamińska, D., Sapiński, T., Aitken, N., Della Rocca, A., Barańska, M., & Wietsma, R. (2017). Virtual reality as a new trend in mechanical and electrical engineering education. *Open Physics*, 936-941.
- Koutitas, G., Smith, K. S., Lawrence, G., Metsis, V., Stamper, C., Trahan, M., & Lehr, T. (2019). A virtual and augmented reality platform for the training of first responders of the ambulance bus. *Proceedings of the 12th ACM International Conference on PErvasive Technologies Related to Assistive Environments* (pp. 299-302). Rodas: ACM.
- Kraft, T., & Kraft, D. (2004). Creating a virtual reality in hypnosis: a case of driving phobia. *Contemporary hypnosis: the journal of the British Society of Experimental and Clinical Hypnosis*, 79-85.
- Kumar, A. (2020). Integration with Unreal Engine 4. In *Beginning PBR Texturing: Learn Physically Based Rendering with Allegorithmic's Substance Painter* (p. 221). Apress. doi:<https://doi.org/10.1007/978-1-4842-5899-6>
- Meyerbröker, K., & Emmelkamp, P. M. (2010). Virtual reality exposure therapy in anxiety disorders: a systematic review of process-and-outcome studies. *Depression and anxiety*, 933-944.
- Nafees, A. (2017). Oculus Rift: A Rift in Reality.
- Oculus Rift S*. (2018, Noviembre 28). Retrieved Noviembre 11, 2020, from Oculus: <https://www.oculus.com/rift-s/>
- Ota, D., Loftin, B., Saito, T., Lea, R., & Keller, J. (1995). Virtual reality in surgical education. *Computers in biology and medicine*, 127-137.
- Quickstart for Google VR SDK for Android*. (n.d.). Retrieved Junio 29, 2020, from Google Developers: <https://developers.google.com/vr/develop/android/get-started>
- Reger, G. M., & Gahm, G. A. (2008). Virtual reality exposure therapy for active duty soldiers. *Journal of clinical psychology*, 940-946.
- Reyes, C. (2019). *Estadísticas de Accidentabilidad 2018*. Superintendencia de Seguridad Social. SISESAT SUSESOSO.

- Riva, G., Bacchetta, M., Baruffi, M., Rinaldi, S., Vincelli, F., & Molinari, E. (2000). Virtual reality-based experiential cognitive treatment of obesity and binge-eating disorders. *Clinical Psychology & Psychotherapy*, 209-219.
- Satava, R. M. (1995). Virtual reality and telepresence for military medicine. *Computers in biology and medicine*, 229-236.
- Schmitt, Y. S., Hoffman, H. G., Blough, D. K., Patterson, D. R., Jensen, M. P., Soltani, M., . . . Sharar, S. R. (2011). A randomized, controlled trial of immersive virtual reality analgesia, during physical therapy for pediatric burns. *Burns: journal of the International Society for Burn Injuries*, 61-68.
- Stanney, K., Lawson, B. D., Rokers, B., Dennison, M., Fidopiastis, C., Stoffregen, T., . . . Fulvio, J. M. (2020). Identifying Causes of and Solutions for Cybersickness in Immersive Technology: Reformulation of a Research and Development Agenda. *International Journal of Human-Computer Interaction*, 36(19), 1783-1803. doi:10.1080/10447318.2020.1828535
- The Rise and Fall and Rise of Virtual Reality*. (n.d.). Retrieved Noviembre 11, 2020, from The Verge: <http://www.theverge.com/a/virtual-reality>
- Unity Technologies. (n.d.). *Installing the Unity Hub*. Retrieved Noviembre 12, 2020, from Unity 3D: <https://docs.unity3d.com/Manual/GettingStartedInstallingHub.html>
- Unity Technologies. (n.d.). *Real-time tools for 3D, AR, & VR development*. Retrieved Octubre 29, 2020, from Unity: <https://unity.com/products>
- van Wyk, E. A., & de Villiers, M. R. (2019). An evaluation framework for virtual reality safety training systems in the South African mining industry. *Journal of the South African Institute of Mining and Metallurgy*, 427-436.
- Vanian, J. (2019, Septiembre 25). *Facebook's Mark Zuckerberg Reveals Hand Tracking, Other Updates for Oculus Quest VR Headsets*. Retrieved Noviembre 11, 2020, from Fortune: <https://fortune.com/2019/09/25/facebook-mark-zuckerberg-oculus-virtual-reality-hands/>
- Wang, P., Wu, P., Wang, J., Chi, H.-L., & Wang, X. (2018). A Critical Review of the Use of Virtual Reality in Construction Engineering Education and Training. *International journal of environmental research and public health*.

ANEXOS

Anexo A: *Degrees of Freedom*

DoF (*Degrees of Freedom*), significa grados de libertad. Moverse verticalmente, horizontalmente y rotar: todos estos son diferentes grados de movimiento.

6 grados de libertad (6DoF) le permiten al usuario moverse e interactuar con un entorno virtual. Es decir, se realiza un seguimiento de su cabeza, controladores y movimiento corporal en el espacio.

3 grados de libertad (3DoF) solo se registra la rotación. Eso significa que solo se rastrea la rotación de la cabeza del usuario y la dirección en la que apunta su controlador (ver Figura 57). 3DoF es más adecuado para videos y experiencias, mientras que 6DOF se usa para juegos y experiencias de realidad virtual más inmersivos que requieren movimiento (Cherdo, 2019; catherine, 2019).

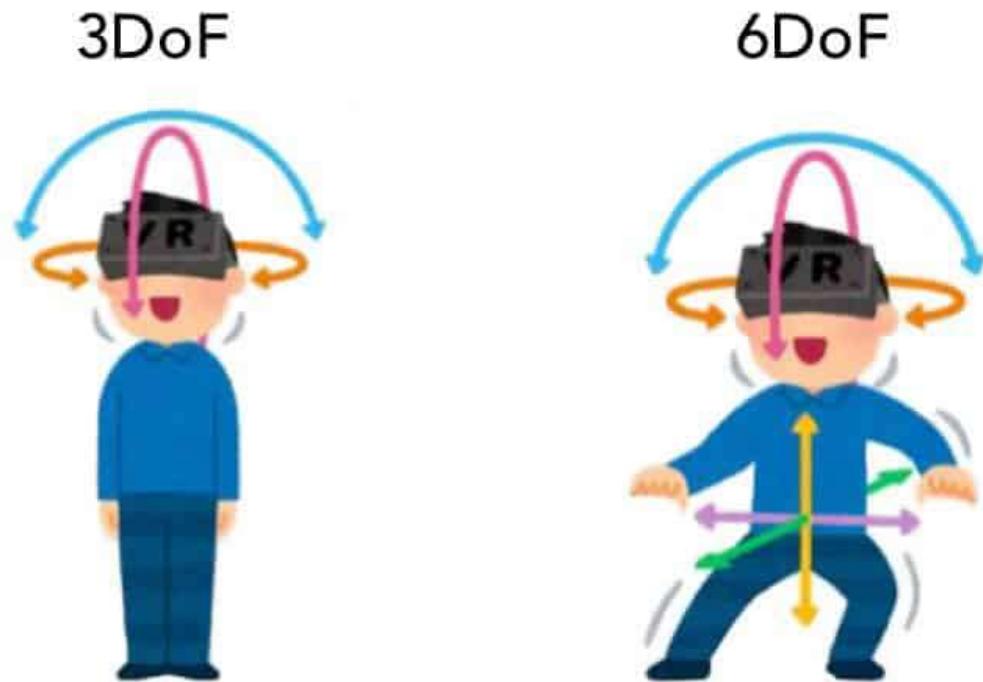


Figura 57: Diferencia entre 3DoF y 6DoF⁸³

⁸³ Extraido de <https://www.aniwaa.com/guide/vr-ar/vr-headset-buying-guide/>

Anexo B: Traslación y Rotación en espacio Virtual

Según Hartley & Zisserman (2003), las coordenadas homogéneas de un punto (x, y, z) del espacio 3D son $(x, y, z, 1)$ o cualquier vector no nulo proporcional que siga la forma:

$$(X, Y, Z, T) = \alpha(x, y, z, 1) \quad \text{Ecuación 1}$$

Y cualquier traslación y rotación en particular es equivalente a una rotación alrededor de un eje helicoidal junto con una traslación a lo largo del eje helicoidal. El eje helicoidal es paralelo al eje de rotación. En el caso de una traslación y un eje de rotación ortogonal (denominado movimiento plano), el movimiento es equivalente a una rotación sola alrededor del eje helicoidal.

Dibujando una prueba geométrica constructiva que se puede visualizar fácilmente. Para el caso 3D, se descompone la traslación t en dos componentes

$$t = t_{\parallel} + t_{\perp} \quad \text{Ecuación 2}$$

paralelo y ortogonal respectivamente a la dirección del eje de rotación (ver Ecuación 3 y Ecuación 4).

$$t_{\parallel} = (t \cdot a) a \quad \text{Ecuación 3}$$

$$t_{\perp} = t - (t \cdot a) a \quad \text{Ecuación 4}$$

Luego, el movimiento euclíadiano se divide en dos partes: primero una rotación alrededor del eje helicoidal, que cubre la rotación $y t_{\perp}$; segundo una traslación por $t_{\parallel}a$ a lo largo del eje helicoidal. El movimiento completo se ilustra en la Figura 58.

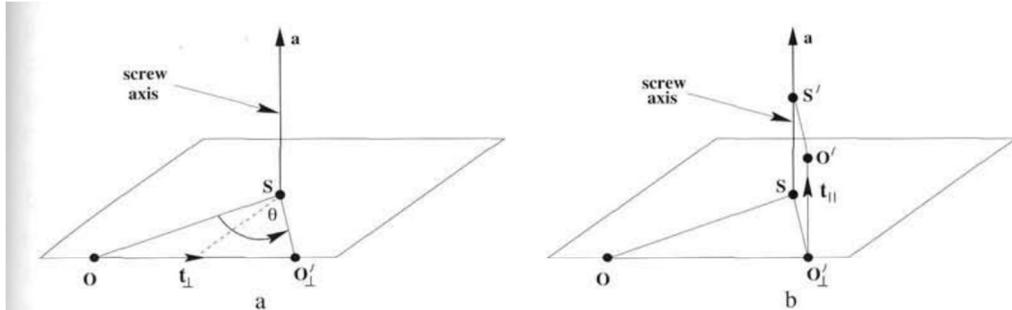


Fig. 3.7. 3D Euclidean motion and the screw decomposition. Any Euclidean rotation R and translation t may be achieved by (a) a rotation about the screw axis, followed by (b) a translation along the screw axis by t_{\parallel} . Here a is the (unit) direction of the rotation axis (so that $Ra = a$), and t is decomposed as $t = t_{\parallel} + t_{\perp}$, which are vector components parallel and orthogonal respectively to the rotation axis direction. The point S is closest to O on the screw axis (so that the line from S to O is perpendicular to the direction of a). Similarly S' is the point on the screw axis closest to O' .

Figura 58: Descomposición de Movimiento y Rotación en un espacio virtual

La descomposición helicoidal se puede determinar a partir de los puntos fijos de la matriz 4×4 que representa la transformación euclíadiana.

Anexo C: Script ReticleTimer.cs

```
using UnityEngine;
using UnityEngine.Events;
using UnityEngine.UI;

public class ReticleTimer : MonoBehaviour
{
    [Header ("Timer")]
    public Image imgTimer;
    [Range (0f, 5f)] public float timeTotal = 1;
    [Header ("Events")]
    public UnityEvent[] timerEvents;
    int idEvent;
    float timeCurrent;
    bool isEnabled;

    void Start ()
    {
        Timer_Exit ();
    }

    void Update ()
    {
        Timer ();
    }

    private void Timer ()
    {
        if (isEnabled)
        {
            timeCurrent += Time.deltaTime;
            imgTimer.fillAmount = timeCurrent / timeTotal;
            if (timeCurrent >= timeTotal)
            {
                Timer_Exit (); //! isEnabled = false;
                timerEvents[idEvent].Invoke ();
            }
        }
    }

    public void Timer_Enter (int _ID)
    {
        isEnabled = true;
        idEvent = _ID;
    }

    public void Timer_Exit ()
    {
        isEnabled = false;
        imgTimer.fillAmount = 0;
        timeCurrent = 0;
    }
}
```

Anexo D: Script TeleportManager.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class TeleportManager : MonoBehaviour
{
    #region Singleton
    private static TeleportManager instance;
    public static TeleportManager Instance
    {
        get
        {
            if (instance == null)
            {
                instance = new TeleportManager ();
            }
            return instance;
        }
    }
    #endregion

    [Header ("Teleport")]
    public Image imgFade;
    [Range (0f, 1f)] public float timeTeleport = 0.5f;
    public Transform player;
    float playerGroundPos;

    private void Awake ()
    {
        if (instance == null)
        {
            instance = this;
        }
        else
        {
            Destroy (gameObject);
        }
    }

    private void Start ()
    {
        playerGroundPos = player.position.y;
        Fade (true);
    }

    public void Fade (bool isFadeIn)
    {
        if (isFadeIn) imgFade.CrossFadeAlpha (0, timeTeleport,
true);
        else imgFade.CrossFadeAlpha (1, timeTeleport, true);
    }
}
```

```
public void Teleport (Vector3 _newPos)
{
    StartCoroutine ("MovePosition", _newPos);
}

IEnumerator MovePosition (Vector3 newPos)
{
    Fade (false);
    yield return new WaitForSeconds (timeTeleport);
    player.position = new Vector3 (newPos.x, newPos.y +
playerGroundPos, newPos.z);
    yield return new WaitForSeconds (timeTeleport);
    Fade (true);
}
}
```

Anexo E: Script GVRTeleportPoint.cs

```
using UnityEngine;

public class GVRTeleportPoint : MonoBehaviour
{
    public void Teleport ()
    {
        TeleportManager.Instance.Teleport(transform.position);
    }
}
```

Anexo F: Script Extintor.cs

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class extintor : MonoBehaviour
{
    bool activo=false;
    public bool wasGrabbed=false;
    public ParticleSystem steam,gas;
    private OVRGrabbable ovrGrabbable;
    public OVRInput.Button shootingBtn;
    Rigidbody rb;
    public Collider extintor_Collider;
    void Start()
    {
        ovrGrabbable=GetComponent<OVRGrabbable>();
        rb=GetComponent<Rigidbody>();
    }

    void Update()
    {
        if (wasGrabbed || ovrGrabbable.isGrabbed) {
            rb.constraints=RigidbodyConstraints.None;
            if (!wasGrabbed) {
                wasGrabbed=true;
            }
            if (ovrGrabbable.isGrabbed) {
                extintor_Collider.isTrigger=true;
            }
            else{
                extintor_Collider.isTrigger=false;
            }
        }
        if (ovrGrabbable.isGrabbed && OVRInput.Get(shootingBtn,
ovrGrabbable.grabbedBy.GetController())) {
            if (!activo) {
                activo=true;
                steam.Play(true);
                // gas.Play();
            }
        }
        else{
            activo=false;
            steam.Stop(true, ParticleSystemStopBehavior.StopEmitting);
            //gas.Stop(true, ParticleSystemStopBehavior.StopEmmiting);
            // steam.SetActive(false);
        }
    }
}
```

Anexo G: Encuesta ‘Desarrollo de Entornos de TESTING en Realidad Virtual’

Desarrollo de Entornos de TESTING en Realidad Virtual.

Estamos realizando esta encuesta para conocer las opiniones de la comunidad de desarrolladores de videojuegos y desarrolladores en general acerca del desarrollo en Realidad Virtual.

1. ¿Has probado alguna vez unas gafas de Realidad Virtual?
 Sí
 No
2. ¿Has desarrollado en Realidad Virtual?
 Sí
 No
3. ¿Qué tecnología crees que es mejor para el desarrollo en Realidad Virtual?
4. ¿Utilizarías un marco de trabajo que estandarice el desarrollo en Realidad Virtual?
 Sí
 No

El siguiente vídeo muestra una escena construida para simular una situación de emergencia. Ve el vídeo (ver Figura 59) y responde la siguiente pregunta por favor
<http://youtube.com/watch?v=g9r4C5ZeuRo>

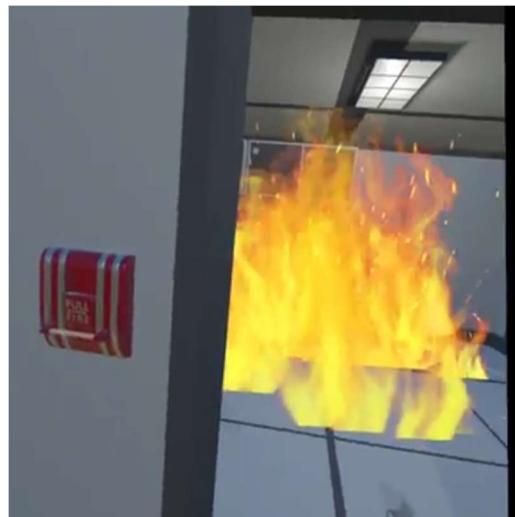


Figura 59: Captura de video

5. ¿Qué aspectos mejorarías o corregirías de esta simulación?
6. ¿Qué escenario construirías para probar o jugar con las reacciones del jugador?

Anexo H: Encuesta ‘Reacción ante situación de incendio en Realidad Virtual’

Reacción ante situación de incendio

Este formulario es para evaluar las reacciones de una persona ante una situación de emergencia, específicamente un incendio, en un entorno de Realidad Virtual.

Por favor vea los videos a continuación y luego responda las preguntas.

Cabe recalcar que el procedimiento esperado para la escena construida, indica que:

1. Luego de detectar una llama sin control (incendio inevitable con el extintor), evacuar la zona.
2. activar la alarma de incendios.
3. utilizar el extintor de incendios.
4. al no apagarse las llamas alejarse lo más posible del origen de las llamas.

Video 1 (ver Figura 60): <http://youtube.com/watch?v=z5SPUp-dONQ>



Figura 60: Captura de Video 1

Video 2 (ver Figura 61): <http://youtube.com/watch?v=g9r4C5ZeuRo>



Figura 61: Captura de Video 2

1. ¿El escenario permite identificar la situación de emergencia?
 - Sí
 - No
2. ¿El escenario tiene todos los elementos necesarios para tomar las decisiones?
 - Sí
 - No
3. ¿El escenario permitió realizar acciones de acuerdo al protocolo esperado?
 - Sí
 - No