

# **VRSurvival: Desarrollo de un Videojuego en Realidad Virtual (RV)**

Trabajo de fin de Grado

GRADO EN INGENIERÍA INFORMÁTICA



**VNiVERSiDAD  
D SALAMANCA**

Julio de 2023

**Autor**

Fiz Rey Armesto

**Tutor/a**

Álvaro Lozano Murciego







# Certificado del tutor

D. Álvaro Lozano Murciego, profesor del Departamento de Informática y Automática de la Universidad de Salamanca,

HACE CONSTAR:

Que el trabajo titulado “VRSurvival: Desarrollo de un Videjuego en Realidad Virtual (RV)”, que se presenta, ha sido realizado por Fiz Rey Armesto, con DNI 34292873B y constituye la memoria del trabajo realizado para la superación de la asignatura Trabajo de Fin de Grado en Ingeniería Informática en esta Universidad.

Salamanca, 5 de Julio de 2023

Fdo.:



# Resumen

La aparición de nuevas tecnologías como la Realidad Virtual ha permitido el desarrollo de videojuegos con un grado de inmersión muy avanzado, llegando hasta el punto de que al usuario le pueda costar distinguir lo real de lo virtual.

A pesar de llevar ya algunos años en el mercado, la novedad de estas tecnologías y el limitado número de jugadores hace que el desarrollo de este tipo de juegos todavía no sea *mainstream* en el mundo del desarrollo de videojuegos y no existan muchos juegos en la categoría *survival*.

Por este motivo, en este Trabajo Fin de Grado plantea el desarrollo de un videojuego de tipo *survival* en Realidad Virtual para la plataforma Windows utilizando el motor gráfico Unreal Engine 5, haciendo uso de su nueva tecnología de Nanite que permite cargar modelos 3D con una cantidad ingente de polígonos sin perder calidad de renderizado a un coste computacional muy bajo. Esto permite ofrecer calidad de gráficos del estado del arte (como los modelos de fotogrametría creados con herramientas como RealityCapture) en plataformas de Realidad Virtual, algo impensable hasta la fecha. Esta tecnología se convertirá en el nuevo paradigma para experiencias realistas con renderizado en tiempo real tanto para Realidad Virtual como para experiencias de escritorio. Esta tecnología combinada con DLSS (Deep Learning Super Sampling o supermuestreo de *deep learning*) de NVIDIA ya permite tener experiencias hiperrealistas y fluidas al mismo tiempo, lo que un juego estilo *survival* necesita.

El videojuego desarrollado consiste en una experiencia inmersiva en la que los jugadores deben enfrentarse a varios tipos de enemigos, solos o con otros jugadores, superando una serie de dificultades hasta cumplir un objetivo: llegar hasta una zona determinada o sobrevivir el máximo tiempo posible.

Para comenzar la experiencia, los jugadores deberán crear una cuenta en el sistema para posteriormente autenticarse y almacenar sus estadísticas de juego.

La experiencia está dividida en niveles en los cuales se almacenan las estadísticas de los jugadores, estas se pueden comprobar dentro del menú principal de la experiencia y compararse con las de otros jugadores mediante dos tipos de marcadores.

El desarrollo de este proyecto ha permitido la integración de los conocimientos adquiridos en el Grado en Ingeniería Informática como los relativos a la Programación Orientada a Objetos, conceptos de Ingeniería del Software, Bases de Datos, Informática Gráfica y otros nuevos directamente relacionados con las tecnologías empleadas en el desarrollo de videojuegos con el motor Unreal Engine.

**Palabras clave:** videojuego, realidad virtual, unreal engine, DLSS, realitycapture, fotogrametría, Meta Quest, survival.

# Abstract

The emergence of new technologies such as Virtual Reality has enabled the development of video games with a highly advanced degree of immersion, to the point where it can be difficult for the user to distinguish between what is real and what is virtual.

Despite having been on the market for some years now, the novelty of these technologies and the limited number of players means that the development of this type of game is still not mainstream in the world of video game development and there are not many games in the survival category.

For this reason, this thesis proposes the development of a survival type videogame in Virtual Reality for the Windows platform using the Unreal Engine 5 graphics engine, making use of its new Nanite technology that allows loading 3D models with a huge amount of polygons without losing rendering quality at a very low computational cost. This makes it possible to deliver state-of-the-art graphics quality (such as photogrammetry models created with tools like RealityCapture) on Virtual Reality platforms, something unthinkable until now. This technology will become the new paradigm for realistic experiences with real-time rendering for both Virtual Reality and desktop experiences. This technology combined with NVIDIA's Deep Learning Super Sampling (DLSS) is already enabling hyper-realistic and fluid experiences at the same time, which is what a survival style game needs.

The developed game is an immersive experience in which players must face various types of enemies, alone or with other players, overcoming a series of challenges to achieve a goal: reach a certain area or survive as long as possible.

To begin the experience, players must create an account in the system and then authenticate and store their game statistics.

The experience is divided into levels in which the players' statistics are stored and can be checked in the main menu of the experience and compared with those of other players by means of two types of scoreboards.

The development of this project has allowed the integration of the knowledge acquired in the Degree in Computer Engineering such as those related to Object Oriented Programming, Software Engineering concepts, Databases, Computer Graphics and other new ones directly related to the technologies used in the development of video games with the Unreal Engine.

**Keywords:** video game, virtual reality, unreal engine, DLSS, realitycapture, photogrammetry, Meta Quest, survival.

# Tabla de contenido

1.	Introducción y antecedentes .....	1
1.1.	Introducción .....	1
1.2.	Organización del documento .....	3
2.	Objetivos.....	4
2.1.	Objetivos técnicos.....	4
2.2.	Objetivos personales .....	5
3.	Conceptos teóricos.....	6
3.1.	Fotogrametría.....	6
3.2.	Realidad Virtual .....	7
3.2.1.	Gafas de Realidad Virtual (HMD) .....	8
3.3.	Trabajos relacionados .....	12
3.3.1.	Género Survival Horror .....	12
3.3.2.	Saga Resident Evil.....	12
3.3.3.	Half Life: Alyx .....	13
4.	Técnicas y herramientas .....	13
4.1.	SCRUM .....	13
4.2.	UML.....	14
4.3.	Herramientas CASE.....	15
4.3.1.	Jira .....	15
4.3.2.	Diagrams.net.....	15
4.3.3.	Excalidraw .....	16
4.3.1.	Visual Paradigm .....	17
4.4.	Herramientas de control de versiones .....	18
4.4.1.	Git.....	18
4.4.2.	Microsoft Azure Repos.....	18
4.4.3.	GitKraken .....	18
4.5.	Herramientas de desarrollo .....	19
4.5.1.	Unreal Engine 5 .....	19
4.5.2.	Blueprints .....	19
4.5.3.	VRExpansionPlugin .....	20
4.5.4.	Visual Studio 2022 .....	20
4.5.5.	Firebase.....	21
4.6.	Herramientas Modelado y Animación 3D .....	22
4.6.1.	Blender .....	22
4.6.2.	RealityCapture .....	22

4.6.3.	Mixamo.....	23
4.6.4.	Mixamo Converter.....	23
4.7.	Herramientas gráficas .....	23
4.7.1.	Adobe Lightroom .....	23
4.7.2.	Adobe Photoshop.....	23
5.	Aspectos Relevantes .....	24
5.1.	Planificación Temporal y marco de trabajo .....	24
5.2.	Especificación de Requisitos del Sistema.....	26
5.3.	Análisis y Diseño del Software.....	30
5.4.	Documentación Técnica de Programación .....	35
5.5.	Manual del Usuario.....	37
5.6.	Problemas durante el desarrollo y soluciones encontradas.....	41
6.	Conclusiones.....	42
7.	Líneas de trabajo futuras.....	43
8.	Referencias .....	44

## Tabla ilustraciones

Ilustración 1 - Índice de popularidad por géneros de los videojuegos de la tienda de Quest [4] .....	1
Ilustración 2 - Realización de fotografías de la Iglesia de San Pedro Fiz do Incio .....	6
Ilustración 3 - Modelo 3D hecho con fotogrametría de la Iglesia de San Pedro Fiz do Incio en Unreal Engine .....	7
Ilustración 4 - Esquema de la instalación del HMD Valve Index [20].....	8
Ilustración 5 - HP Reverb G2 con cámaras de seguimiento en primer plano [21] .....	9
Ilustración 6 - Usuario jugando con las gafas de RV Meta Quest 2 [27] .....	11
Ilustración 7- Captura de pantalla de RE7 con mod para RV [71].....	12
Ilustración 8 - Captura de pantalla de Half Life: Alyx [76] .....	13
Ilustración 9 - Esquema del framework Scrum [29].....	14
Ilustración 10 - Pestaña del backlog del proyecto VRSurvival .....	15
Ilustración 11 – Creación de un diagrama UML con Draw.io [35].....	16
Ilustración 12 - Captura de pantalla de la interfaz de Excalidraw .....	16
Ilustración 13 - Captura de pantalla de Visual Paradigm .....	17
Ilustración 14 - Captura de pantalla de GitKraken con el proyecto VRSurvival .....	18
Ilustración 15 - Fragmento de la clase Blueprint ZombieBase .....	20
Ilustración 16 - Recompilación del proyecto VRSurvival para la versión 5.2 de UE....	21
Ilustración 17 - Base de datos Firestore .....	21
Ilustración 18 - Edición de un modelo 3D en Blender .....	22
Ilustración 19 - Nube de puntos de la iglesia de San Pedro Fiz de Hospital do Incio en RealityCapture.....	22
Ilustración 20 - Edición de una foto utilizada en uno de los menús en Adobe Photoshop .....	23
Ilustración 21 - Calendario: Febrero.....	24
Ilustración 22 - Burndown Chart: Sprint 1 .....	25
Ilustración 23- Roadmap .....	26
Ilustración 24 - Diagrama Arquitectónico .....	27
Ilustración 25 - Diagrama de casos de uso: Gestión de usuarios .....	28
Ilustración 26 - Diagrama del modelo del dominio.....	30
Ilustración 27 - Diagrama de secuencia: UC11 - Seleccionar arma.....	30
Ilustración 28 - Diagrama de niveles.....	31
Ilustración 29 - Behaviour Tree de ZombieBase.....	32
Ilustración 30 - Firestore Database.....	33
Ilustración 31- Diagrama de secuencia: UC01 - Registro .....	33
Ilustración 32 - Diagrama de despliegue .....	34
Ilustración 33 - Simulación de una Blueprint.....	34
Ilustración 34 - Compilación del proyecto en Visual Studio.....	35
Ilustración 35 - Archivo index.js del proyecto Firebase.....	36
Ilustración 36 - Tráiler del videojuego en Youtube.....	37
Ilustración 37 - Etiqueta PEGI 18.....	37
Ilustración 38 - Advertencia de contenido para mayores de 18 .....	38
Ilustración 39 - Configuración del sistema guardián – 2 [8] .....	38
Ilustración 40 - Controles .....	39
Ilustración 41 - Menú de selección de arma .....	40
Ilustración 42 - Zombie recibe un disparo en la cabeza .....	40
Ilustración 43 – Recarga de pistola 1911.....	41



# Índice de tablas

Tabla 1 - Valve Index .....	10
Tabla 2 - HP Reverb 2 .....	10
Tabla 3 - Meta Quest 2 .....	10
Tabla 4 - Pico 4.....	10
Tabla 5 - PSVR2.....	11
Tabla 6 - Sprint 1 .....	25
Tabla 7 - IRQ-001 Usuarios .....	26
Tabla 8 – VRSUR-116 .....	27
Tabla 9 - UC-001: Registrarse.....	28
Tabla 10 - RNF-01.....	29



# Glosario

**FPS:** Disparos en primera persona, de las siglas en inglés (*First Person Shooter*). Referido a un género de videojuegos.

**RV:** Realidad Virtual.

**Triple A:** Se dice de un videojuego creado por un estudio de fama mundial y con alto presupuesto.

**Mainstream:** De actualidad y amplio uso.

**BBDD:** Base de Datos.

**Gameplay:** Parte jugable de un videojuego.

**Arcade:** Refiriéndose a un género de videojuegos, se dice de los videojuegos pensados para usuarios no expertos, juegos del estilo a los presentes en máquinas recreativas.

**Shooter:** Juego de disparos.

**RE:** Resident Evil.

**PCVR:** RV para la plataforma Windows.

**DLSS:** Supermuestreo de Aprendizaje Profundo, de las siglas en inglés (*Deep Learning Super Sampling*)

**Framework:** Marco de trabajo.

**Portfolio:** Recopilación de trabajos visuales más relevantes de una persona.

**LiDAR:** Dispositivo que permite determinar la distancia desde un emisor láser a un objeto o superficie utilizando un haz láser pulsado, de las siglas en inglés (*Laser Imaging Detection And Ranging*).

**UE:** Unreal Engine.

**GUI:** Interfaz gráfica de usuario, de las siglas en inglés (*Graphical User Interface*).

**CLI:** Interfaz de línea de comandos, de las siglas en inglés (*Command Line Interface*).

**Asset:** Cualquier elemento usado en la creación de videojuegos.

**LOD:** Complejidad de una representación de un modelo 3D, normalmente se cambia de un LOD más complejo a uno más simple con respecto a la distancia del modelo con el espectador.

**Shadow mapping:** Es un proceso utilizado en el renderizado de gráficos 3D para añadir sombras, se utiliza tanto en escenas pre-renderizadas o escenas renderizadas en tiempo real, como sería el caso de los videojuegos.

**Indie:** Se dice del videojuego que no tiene una distribuidora, son publicados por los propios estudios que los crean.

**Networking:** En el contexto informático se refiere al hecho de comunicar 2 o más PCs

**Locomotion:** Lógica que permite el movimiento del personaje controlable por el mundo virtual.

**Gripping:** Sistema que permite el “agarre” de elementos en simulaciones de RV.

# 1. Introducción y antecedentes

## 1.1. Introducción

La tecnología de RV (Realidad Virtual) está pasando por un auge en los últimos años teniendo un punto de inflexión durante la pandemia debida a la COVID [1], que propició que muchas empresas y usuarios vieran en esta tecnología una nueva forma de vivir experiencias sin tener que desplazarse de su domicilio. Todos estos nuevos usuarios en su mayoría ven en la RV no solo una tecnología para los videojuegos, sino que descubrieron su uso en muchas otras facetas como educación, turismo, etc. [2].

Los géneros de videojuegos más vendidos en la tienda de Oculus[3] en 2022 han sido los siguientes, como se puede comprobar en la Ilustración I:

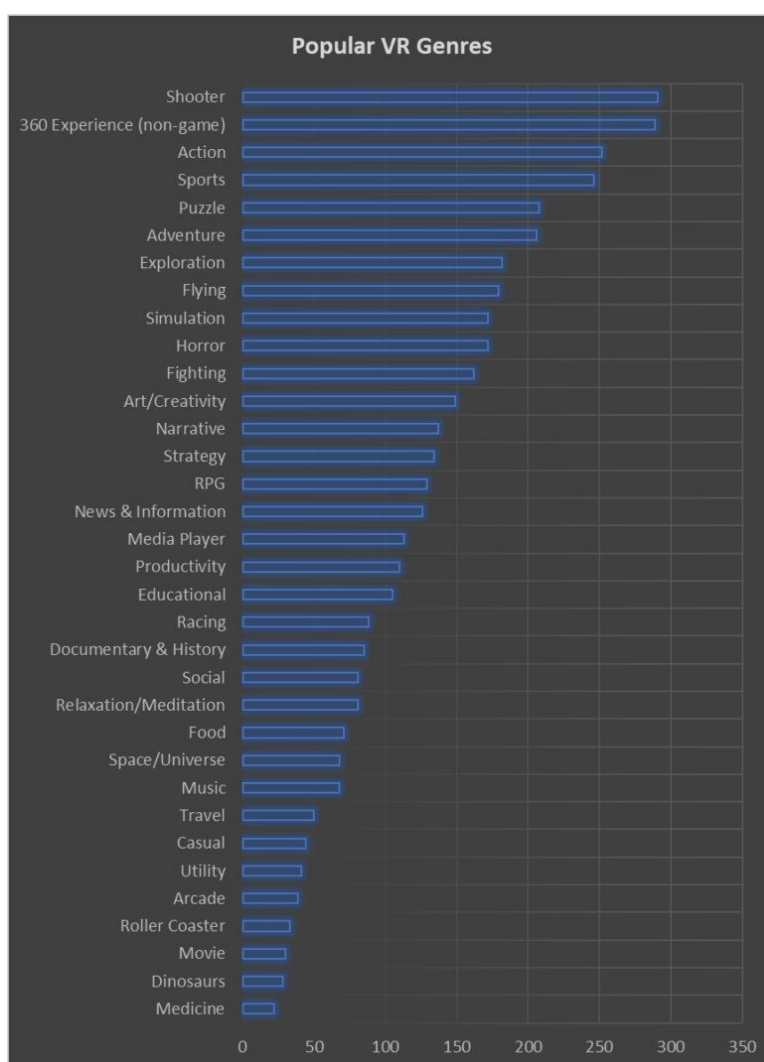


Ilustración 1 - Índice de popularidad por géneros de los videojuegos de la tienda de Quest [4]

Los géneros que se encuentran en la tienda son bastante amplios y la mayoría de los juegos podrían entrar en la definición de varios géneros, por ejemplo, la mayoría de los videojuegos del género *Shooter* podría entrar en la descripción del género *Action* o

## Introducción

*Adventure*. Dentro de los géneros de la gráfica se pueden destacar, a diferencia de los géneros que veríamos en una plataforma de escritorio, los siguientes:

- *Puzzle*: Es un género mucho más popular en plataformas como los smartphones y dedicado a un público general, pero que ve en la RV nuevas posibilidades de crear rompecabezas con mundos interactivos en tres dimensiones.
- Experiencias (sin ser videojuegos): como los géneros *Educational*, *News & Information*, *Social* y *Art/Creativity*. En estos géneros se pueden destacar aplicaciones como VRChat, que permite comunicarse con otros usuarios en RV para charlar, bailar o disfrutar de conciertos en escenarios creados por los propios usuarios.[5]
- Simulador: Los videojuegos de simuladores son muy populares en RV ya que estos buscan tener una experiencia lo más cercana a la realidad posible, destacan los simuladores de conducción y aviación.

Viendo estas estadísticas todo parece normal, el problema reside en que la mayoría de los títulos más jugados no son recientes. Las listas de títulos más jugados siguen dominadas por juegos como Beat Saber [6] (2018), Superhot VR (2016) [7] y Half-Life: Alyx (2020) [8], [9]. Además, estos juegos en su mayoría son títulos pensados para jugadores casuales y, aunque tengan una gran jugabilidad por ello, no mantienen el ecosistema vivo. Una plataforma como la RV necesita juegos AAA y no un gran catálogo de juegos *indies* de dudosa calidad. El ejemplo que deberían tomar todos los desarrolladores de videojuegos para RV es el del gran éxito comercial de Half Life:Alyx, que demostró que se pueden hacer videojuegos exclusivos para RV de gran presupuesto y mayores beneficios. Aun así, desde el lanzamiento de Half Life: Alyx en 2020, ningún gran estudio de videojuegos ha lanzado un título exclusivo en RV, al contrario, la mayoría de los estudios han cancelado títulos o retrasado la salida de estos como, por ejemplo, la cancelación de Splinter Cell VR [10] por parte de Ubisoft. Se podría decir que hay una gran sequía de videojuegos en RV de gran presupuesto, aun con los grandes éxitos de Half Life: Alyx y los modos de RV de juegos de escritorio como Resident Evil 7 Biohazard[11] que consiguió superar los 1.25 millones de jugadores en una sola plataforma (PlaystationVR) [12] lo que lo convierte en el segundo videojuego de RV más jugado de la historia solo por detrás de Beat Saber[9]. La tendencia del público es clara y este quiere juegos de gran presupuesto para la RV pero las compañías desarrolladoras no acompañan.

Por esto se propone el desarrollo de un videojuego de RV multijugador para PC del género *survival horror* siguiendo el éxito que tienen estos títulos en la plataforma de escritorio y lo bien que se adaptan a la experiencia en RV. El desarrollo en PC aprovecha toda la fuerza de computación que da el hardware de última generación permitiendo tener gráficos hiperrealistas. Para conseguir este objetivo se utilizará el motor Unreal Engine 5 [13]aprovechando sus últimas tecnologías como Nanite [14]. Nanite permite tener cantidades muy altas de polígonos en modelos 3D sin que afecten prácticamente al rendimiento, además con Nanite los diseñadores no tienen que realizar LODs (*Levels of Detail*) ya que el motor lo hace automáticamente. Como consecuencia, la utilización del motor UE5 reduce el tiempo de desarrollo incrementando el rendimiento y la calidad gráfica, se puede ver con más detalle el funcionamiento del motor UE5 en el apartado 4.5.1 Unreal Engine 5. En el desarrollo también se incorporará la tecnología de DLSS [15] (*Deep Learning Super Sample and Frame Generation*) que utiliza la IA (Inteligencia Artificial) para mejorar el rendimiento del videojuego. Además, se utiliza el *framework* VRExpansionPlugin, del que se puede encontrar más información en el apartado 4.5.1.2 VRExpansionPlugin, para facilitar el desarrollo sobre todo en la parte referida al

multijugador por su exponencial dificultad y poca información de cara a las diferencias en el desarrollo comparado con un videojuego de escritorio.

El producto final del proyecto se ha proyectado como una demostración técnica del estado del arte de un videojuego de estilo supervivencia utilizando las mecánicas del género traducidas a la Realidad Virtual. En esta demostración (*demo*), el jugador deberá afrontar una serie de dificultades y derrotar a enemigos hasta llegar al final del nivel y/o hasta que no pueda con estas adversidades. El usuario puede crear e iniciar sesión en una cuenta en la cual se almacenarán sus estadísticas en una BBDD (Base de Datos) y estas también pueden ser visualizadas y comparadas con las de otros usuarios dentro del juego. Todas las interfaces están integradas dentro del entorno del nivel jugable y su interacción en RV es intuitiva para no perder la sensación de inmersión que nos proporciona la tecnología de RV.

La motivación detrás del proyecto está fundamentada en la idea de poder hacer videojuegos al nivel del estado del arte utilizando todas las últimas tecnologías disponibles para cualquier desarrollador, con la visión de poder empezar una carrera profesional en este sector en un futuro.

## 1.2. Organización del documento

En esta memoria se presenta todo el proceso de realización del proyecto VRSurvival en todas sus fases. Este documento se divide en varias secciones:

1. **Introducción:** Se presenta el proyecto y la memoria de su realización.
2. **Objetivos:** Se especifican los objetivos que se quieren cumplir a la finalización del proyecto.
3. **Conceptos Teóricos:** Se exponen los conceptos teóricos más importantes y necesarios en su comprensión para el desarrollo del proyecto.
4. **Técnicas y herramientas:** Se muestran todas las técnicas y herramientas utilizadas durante el desarrollo del proyecto
5. **Aspectos Relevantes:** se detallan las partes más importantes que vienen explicadas detalladamente en los siguientes anexos:
  - Anexo I: Planificación Temporal
  - Anexo II: Especificación de Requisitos del Sistema
  - Anexo III: Especificación de Diseño
  - Anexo IV: Documentación Técnica de Programación
  - Anexo V: Manual del Usuario
6. **Conclusiones:** Se comentan las resoluciones a las que se llegan en la finalización del proyecto.
7. **Líneas de trabajo futuras:** Se exponen las posibles vías de desarrollo a seguir para el proyecto.
8. **Bibliografía:** Se detalla toda la bibliografía utilizada para la realización de esta memoria.

## 2. Objetivos

El objetivo del proyecto es crear un videojuego de RV con las nuevas tecnologías que proporciona Unreal Engine 5 y que probablemente formarán parte del estado del arte de la industria del videojuego en los próximos años. Para conseguir este objetivo general se tendrán que cumplir los siguientes objetivos técnicos.

### 2.1. Objetivos técnicos

Los objetivos técnicos del proyecto son los siguientes:

- **Desarrollo de un nivel tutorial:** Desarrollar un nivel de tutorial en el cual se irán introduciendo los elementos del *gameplay* interactivamente.
- **Diseño e implementación de un sistema de oleadas:** Diseñar e implementar un nivel con sistema de oleadas de enemigos infinito, un nivel en el que el usuario podrá jugar indefinidamente hasta que cometa un error que provoque el fin de la partida.
- **Diseño y desarrollo de mecánicas Survival:** Diseñar y desarrollar el *gameplay* de un jugador que se basará en mecánicas *survival*, siendo estas un poco distendidas en el modo cooperativo dada la naturaleza del mismo, acercándose más a mecánicas *arcade/shooter*.
- **Implementación de un *Gameplay* multijugador:** Implementar en el videojuego un modo multijugador cooperativo entre 2 personas. Los usuarios podrán crear salas a las que se podrán añadir otros jugadores para poder jugar todos los niveles que deseen.
- **Diseño e implementación de un módulo de salas multijugador:** Diseño e implementación de salas multijugador empleando los servicios ofrecidos por Epic, (empresa propietaria de Unreal Engine) para poder jugar partidas cooperativas con otros jugadores. Los usuarios podrán crear ellos mismos estas salas multijugador.
- **Implementación de un sistema de almacenamiento de estadísticas:** Implementación de distintas estadísticas del jugador durante las partidas, que se almacenarán en la BBDD de la nube de Firebase. Los usuarios podrán ver las estadísticas de su partida dentro del juego. Los usuarios también podrán ver las estadísticas de otros usuarios en varios marcadores (global y local).
- **Diseño e implementación de un sistema de autenticación de usuarios:** Implementación de un sistema de autenticación basado en la nube de Firebase.
- **Integración de modelos 3D obtenidos mediante fotogrametría:** Integración de modelos 3D de fotogrametría para su inclusión en los escenarios del juego, estos serán creados con herramientas de fotogrametría como RealityCapture gracias a la tecnología de Nanite de Unreal Engine.

## 2.2. Objetivos personales

Los objetivos personales del desarrollador para este proyecto son los siguientes:

- **Revisión de las tecnologías RV utilizadas en el desarrollo de videojuegos:** Al finalizar el proyecto se pretende contar con conocimientos actualizados acerca del ecosistema de desarrollo de videojuegos RV y de las tecnologías empleadas, así como cuáles son sus limitaciones. Se pondrá el foco en adquirir competencias en las tecnologías que se utilizan en el desarrollo del proyecto.
- **Aprendizaje del motor Unreal Engine 5:** Se busca adquirir un amplio conocimiento del funcionamiento del motor Unreal Engine 5, haciendo hincapié en estos apartados:
  - Estructura de las clases predefinidas del motor (Gameplay Framework).
  - Jerarquía de las clases (Programación Orientada a Objetos).
  - Empleo de eventos y funciones.
  - La tecnología Nanite.
- **Manejo del *framework* VRExpansionPlugin:** Se busca tener una comprensión completa del funcionamiento de este *framework* suficiente para poder empezar otro proyecto utilizando este *framework* desde cero sin ninguna traba.
- **Prototipado de videojuego comercial:** El desarrollo del proyecto tiene como objetivo final mostrar una demo técnica del videojuego, pero este no es un videojuego listo para salir al mercado. Para ello, además del tiempo y recursos necesarios más allá del alcance de una asignatura como el TFG, sería necesario un equipo con conocimientos artísticos para desarrollar escenarios y modelos 3D realistas y expertos en doblaje y sonido para poder tener una ambientación propia de un juego de horror. Sin embargo, este proyecto supone una antesala a la posible publicación de un videojuego con estas características.
- **Puerta de entrada en la industria del videojuego:** Se busca con este proyecto que sirva como parte de un porfolio para poder conseguir un trabajo en la industria del videojuego. Estos porfolios pueden estar en plataformas digitales de la industria como ArtStation.[16]

### 3. Conceptos teóricos

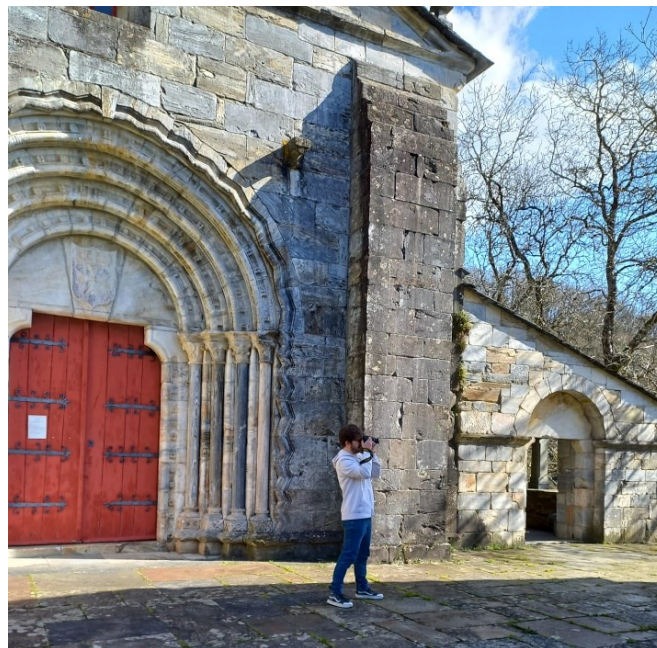
En este apartado se presentan los conceptos teóricos relacionados con el dominio del problema tratado en este Trabajo Fin de Grado. Se describen conceptos relacionados con el desarrollo de videojuegos, la Informática Gráfica y distintos conceptos necesarios para comprender proyecto tratado.

#### 3.1. Fotogrametría

Según el Ministerio de Transportes, Movilidad y Agenda Urbana [17]:

*“La fotogrametría es la técnica cuyo fin es estudiar y definir con precisión la forma, dimensiones y posición en el espacio de un objeto cualquiera, utilizando esencialmente medidas hechas sobre una o varias fotografías de ese objeto. (Bonneval).”*

Esta tecnología nos permite obtener modelos 3D muy precisos de objetos reales simplemente con la ayuda de una cámara y/o un sensor láser como un LiDAR. Para ello, primero se deben sacar fotos del objeto y/o zona que se desee capturar desde todas las perspectivas que sea posible, como se muestra en la Ilustración 2, y, con la ayuda de un programa como, por ejemplo, RealityCapture se podría crear una nube de puntos de la zona y posteriormente un modelo 3D muy aproximado al real.



*Ilustración 2 - Realización de fotografías de la Iglesia de San Pedro Fiz do Incio*

Estos modelos fotorrealistas son muy pesados y costosos computacionalmente debido a su gran número de polígonos como para hacer un renderizado en tiempo real. Gracias a la tecnología Nanite del motor Unreal Engine 5 esto ha cambiado, con esta tecnología se pueden renderizar con un coste computacional aceptable, lo que permitiría incorporarlos a proyectos dentro de los límites de rendimiento aceptables para una aplicación en RV. Como ejemplo, es posible ver en la Ilustración 3 un plano detalle del modelo 3D en Unreal Engine de la puerta de la Iglesia de San Pedro Fiz do Incio previamente mostrado la Ilustración 2.



*Ilustración 3 - Modelo 3D hecho con fotogrametría de la Iglesia de San Pedro Fiz do Incio en Unreal Engine*

## 3.2. Realidad Virtual

De acuerdo con la Real Academia Española [18], la Realidad Virtual (RV) se define como:

*“Representación de escenas o imágenes de objetos producida por un sistema informático, que da la sensación de su existencia real.”*

La RV moderna se ha convertido es una tecnología ampliamente conocida hoy en día y comenzó con el lanzamiento del *Kickstarter* [19] en 2012 del prototipo de gafas de RV que se acabaría convirtiendo en el primer modelo de Oculus Rift gracias a los 2.437.429\$ recaudados para el proyecto por más de 9.522 mecenas. Esta tecnología se usa en muchas industrias diferentes como: militar, médica, educación, etc. Sin embargo, a nivel de consumidor más generalista, su mercado más conocido es el de los videojuegos. Normalmente, para utilizar estos sistemas, los usuarios están equipados con unas gafas de RV acompañadas de un mando en cada mano. Las gafas tienen incorporada una pantalla en la cual se muestran escenarios virtuales que reaccionan a los movimientos de la cabeza y las manos del usuario dándole la sensación de que se encuentra en dicho escenario virtual.

### 3.2.1. Gafas de Realidad Virtual (HMD)

A continuación, se muestra una gráfica con las especificaciones más importantes de los HMD (Head Mounted Display) más populares, pero primero se explicarán varios apartados

- Tipo de Seguimiento: Interior vs Exterior. Se necesita un seguimiento de la posición de los mandos con respecto al HMD para calcular correctamente la posición de estos en el mundo real. Para ello existen 2 técnicas, el *tracking* Exterior y el Interior. En el exterior se posicionan previamente cámaras en el espacio en que se va a utilizar el HMD y estas se comunican con el HMD para enviar la posición relativa como podemos ver en la Ilustración 4, las 2 cámaras necesarias serían los objetos 214a y 214b en la Ilustración 4. La ventaja de este método es que siempre se tiene la visión de los mandos y el HMD y la desventaja es la poca movilidad que tiene este sistema, ya que habría que instalar las cámaras en cualquier zona en la cual se quiera utilizar el HMD (normalmente se hace una instalación fija en la pared).

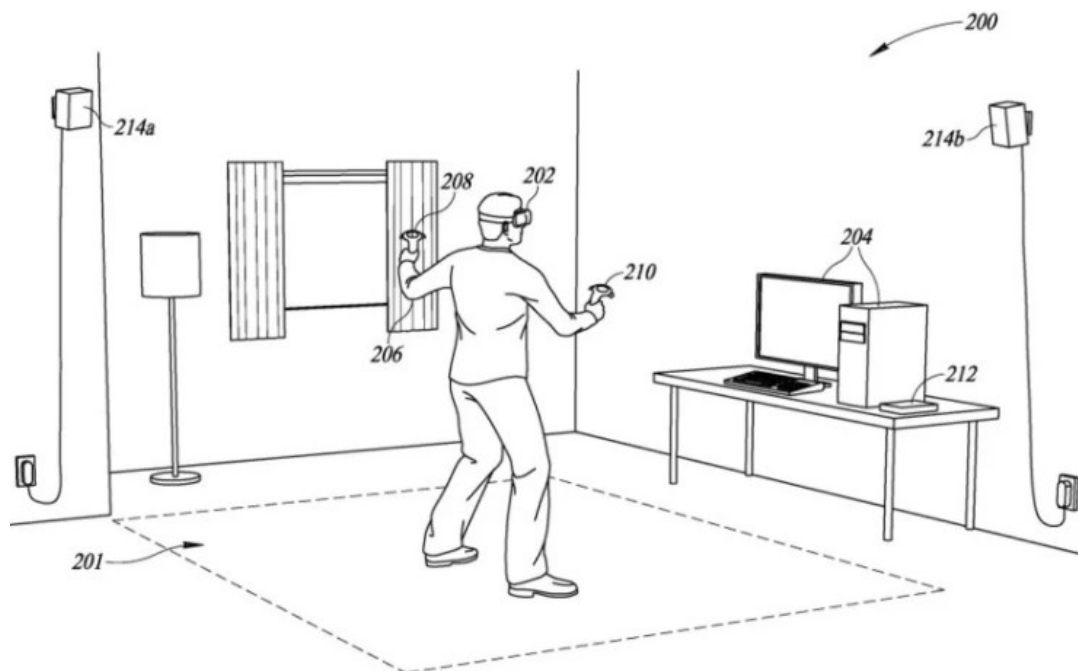


Ilustración 4 - Esquema de la instalación del HMD Valve Index [20]

El método de *tracking* interior funciona con varias cámaras colocadas por el HMD, este forma es la más ampliamente utilizada a día de hoy porque no requiere instalación, así las gafas se vuelven mucho más flexibles y pueden utilizarse en otros espacios fácilmente, el inconveniente que tiene este técnica es que en ciertas posiciones se pueden ocultar los mandos a la vista de las cámaras del HMD y por tanto perder el seguimiento durante cierto tiempo, por ejemplo, si se acercan los mandos demasiado hacia la cara o hacia la espalda.



Ilustración 5 - HP Reverb G2 con cámaras de seguimiento en primer plano [21]

- *Standalone*: Los HMD se dividen en dos tipos, los que tienen hardware para ejecutar las experiencias de RV dentro del propio HMD, es decir los HMD *standalone* o todo-en-uno, y los que no. Los HMD *standalone* están incorporados con un procesador ARM que permite ejecutar las experiencias directamente dentro del HMD, esto da beneficios como que la latencia es casi inexistente y que se puede ejecutar libremente sin necesidad de otro componente hardware externo como un ordenador o una videoconsola. Por otra parte, los HMD se calientan bastante debido al procesador ARM, lo cual es bastante incómodo teniendo en cuenta la ubicación de este y puede hacer que el usuario sude, por otro lado, las baterías y todos los componentes hardware necesarios para ello añaden un peso que no es agradable para largas sesiones de uso del dispositivo. Los HMD no *standalone* necesitan de otro dispositivo secundario para que ejecute las experiencias de RV y se las envíe al HMD, esto puede ser mediante un cable o inalámbricamente:
  - El uso de cable es la opción más estable, pero limita el rango de movimientos del usuario y suele ser bastante incómodo para este ya que se puede tropezar con él al realizar movimientos varios.
  - El uso inalámbrico mediante Wifi local requiere una conexión con un ancho de banda alto, mínimo una conexión de 5GHz para evitar la pérdida de calidad de la imagen. El mayor problema es que puede dar errores dependiendo de la estabilidad de la red y que se necesita estar en el rango de una red Wifi de 5GHz a la que esté conectado el ordenador.

<b>Nombre</b>	<b>Valve Index [22]</b>
<b>Año</b>	2019
<b>Tipo de panel</b>	LCD
<b>Resolución de pantalla (por ojo)</b>	1600 x 1440
<b>Tasa de refresco</b>	80, 90, 120 o 144 Hz
<b>Campo de visión</b>	130° Diagonal
<b>Tipo de seguimiento</b>	Externo
<b>Plataforma / <i>Standalone</i></b>	PCVR / No
<b>Batería</b>	No dispone (cableado)

<b>Almacenamiento</b>	No dispone
<b>Peso</b>	810g
<b>Precio en euros en España</b>	1079 (Kit básico listo para jugar)

Tabla 1 - Valve Index

<b>Nombre</b>	<b>HP Reverb 2 [23]</b>
<b>Año</b>	2020
<b>Tipo de panel</b>	LCD
<b>Resolución de pantalla (por ojo)</b>	2160 x 2160
<b>Tasa de refresco</b>	90 Hz
<b>Campo de visión</b>	114° Diagonal
<b>Tipo de seguimiento</b>	Interno
<b>Plataforma / Standalone</b>	PCVR / No
<b>Batería</b>	No dispone (cableado)
<b>Almacenamiento</b>	No dispone
<b>Peso</b>	550g
<b>Precio en euros en España</b>	700

Tabla 2 - HP Reverb 2

<b>Nombre</b>	<b>Meta Quest 2[24]</b>
<b>Año</b>	2020
<b>Tipo de panel</b>	LCD
<b>Resolución de pantalla (por ojo)</b>	1920 x 1832
<b>Tasa de refresco</b>	72, 90 o 120 Hz
<b>Campo de visión</b>	90°
<b>Tipo de seguimiento</b>	Interno
<b>Plataforma / Standalone</b>	Quest, compatible con PCVR / Si
<b>Batería</b>	3640 mAh
<b>Almacenamiento</b>	64, 128 o 256GB
<b>Peso</b>	503g
<b>Precio en euros en España</b>	349 (64GB), 449 (256GB)

Tabla 3 - Meta Quest 2

<b>Nombre</b>	<b>Pico 4[25]</b>
<b>Año</b>	2022
<b>Tipo de panel</b>	LCD
<b>Resolución de pantalla (por ojo)</b>	2160 x 2160
<b>Tasa de refresco</b>	72, 90Hz
<b>Campo de visión</b>	105°
<b>Tipo de seguimiento</b>	Interno
<b>Plataforma / Standalone</b>	Pico OS, compatible con PCVR / Si
<b>Batería</b>	5300 mAh
<b>Almacenamiento</b>	128 o 256GB
<b>Peso</b>	503g
<b>Precio en euros en España</b>	349 (64GB), 449 (256GB)

Tabla 4 - Pico 4

<b>Nombre</b>	<b>PSVR2[26]</b>
<b>Año</b>	2023
<b>Tipo de panel</b>	OLED HDR
<b>Resolución de pantalla (por ojo)</b>	2040 x 2000

<b>Tasa de refresco</b>	90 y 120Hz
<b>Campo de visión</b>	110° Diagonal
<b>Tipo de seguimiento</b>	Interno
<b>Plataforma / <i>Standalone</i></b>	PS5 / No
<b>Batería</b>	No dispone (cableado)
<b>Almacenamiento</b>	No dispone
<b>Peso</b>	560g
<b>Precio en euros en España</b>	600

*Tabla 5 - PSVR2*

Después de estudiar las especificaciones de todas las gafas (que se muestran en las tablas 1 a 5), leer distintas opiniones y habiendo probado personalmente los HMD: Valve Index, HP Reverb 2, Meta Quest 1 y 2, Meta Quest Pro y Microsoft Hololens, se llega a la conclusión de que se necesita un HMD ligero y que permita la conexión inalámbrica por comodidad, como se puede observar en la Ilustración 6, ya que va a tener muchas horas de uso seguidas. Además, las gafas de RV Meta Quest 2 (anteriormente Oculus Quest 2) son las gafas de RV más vendidas de la historia y las más utilizadas desde su salida en 2020[27]. Esto permite hacer las pruebas en el hardware que tendrá la mayoría del público objetivo.



*Ilustración 6 - Usuario jugando con las gafas de RV Meta Quest 2 [27]*

### 3.3. Trabajos relacionados

#### 3.3.1. Género Survival Horror

El género *Survival Horror* es un subgénero de los videojuegos de acción pensado para adultos que mezcla elementos de los videojuegos de terror en el que el objetivo es la supervivencia del jugador. Las mecánicas se basan en que el personaje del jugador es vulnerable, está aislado y suele estar poco armado, por lo que tendrá que racionar sus recursos como la munición o medicinas y/o pensar en evitar enemigos y utilizar el sigilo. También se añaden elementos de puzle y rompecabezas. El término *Survival Horror* se acuña con el lanzamiento del videojuego *Resident Evil* en 1996 [60]. Aunque la saga RE (*Resident Evil*) comenzó hace muchos años, en los últimos está teniendo un repunte con otros juegos del género como *The Last of Us Part II* [61], *Dead Island 2* [62] o *The Callisto Protocol* [63]. También se puede ver la tendencia a hacer *remakes* de juegos clásicos del género con los avances gráficos disponibles actualmente, ejemplos de esto son los *remakes* de: *Resident Evil 2* [64], 3 [65] y 4 [66], *Dead Space* [67], *Silent Hill 2* [68] *Metro* [69] o el de *Alone in the Dark* [70] (el cual se considera el primer juego del género [60]) y también sus respectivas versiones/mods para RV. Por ejemplo, los mods de RV para PCVR de la mayoría de los juegos del motor RE Engine (motor gráfico de Capcom creado para el desarrollo de RE7 [11] y que se usó posteriormente en los remakes de RE2, RE3, RE4...) son obra del *modder praydog* [71].

#### 3.3.2. Saga Resident Evil

La saga Resident Evil es un punto de partida para las mecánicas que definen el género *survival horror* y no puede pasar desapercibido, siendo esta la saga del género más vendida de toda la historia con 138 millones de unidades vendidas en todo el mundo [3]. Aunque en entregas pasadas la saga perdió un poco la esencia del survival horror, las últimas dos entregas: Resident Evil 7 Biohazard (ver ilustración 7) [11] y Resident Evil 8 Village [72] vuelven a las mecánicas rompedoras del género coincidiendo con su vuelta al mainstream. [73], [74].



Ilustración 7- Captura de pantalla de RE7 con mod para RV [71]

### 3.3.3. Half Life: Alyx

No se puede hablar de juegos en RV sin hablar de *Half Life: Alyx* (ver ilustración 8) [8]. Considerado el mejor juego jamás creado en RV tanto por el público como por la crítica [75]. Este es el primer gran título triple A creado expresamente para RV y dentro de una de las sagas más esperadas de la historia, 13 años después de su última entrega. Este videojuego trata una historia lineal desde el género FPS en la cual la protagonista se tiene que enfrentar a una serie de enemigos y puzzles para avanzar en la historia utilizando todo tipo de armas e interacciones con el entorno, lo que genera una gran sensación de inmersión al jugador ayudándose de la gran calidad gráfica del título. Muchos de los elementos interactivos de este proyecto, tienen su inspiración en las interacciones e interfaces utilizadas en este videojuego por su utilidad y resultado frente al público.



Ilustración 8 - Captura de pantalla de *Half Life: Alyx* [76]

## 4. Técnicas y herramientas

Las técnicas que se han llevado a cabo durante el desarrollo del videojuego son las descritas a continuación:

### 4.1. SCRUM

SCRUM ha sido el *framework* más popular y utilizado dentro de las metodologías ágiles durante años y ha sufrido un aumento en su uso a nivel de equipo del 54% al 87% en los dos últimos años según las empresas que participaron en el State of Agile 2022 [28].

Se ha optado por este *framework* dada la familiaridad del desarrollador con él (utilizado en prácticas curriculares en empresas del sector de 3D) y por la naturaleza del proyecto. En un proyecto de estas características, en el que se aprenden muchos conceptos a lo largo del proyecto, es muy fácil que surjan cambios a lo largo de este. Esta naturaleza cambiante hace que una metodología ágil como SCRUM sea muy útil. Al tratarse de un proyecto realizado por una única persona, la metodología no se puede aplicar de forma clásica por completo por varios motivos:

## UML

- No se tienen los participantes necesarios para los diferentes roles.
- Algunos de los eventos no son necesarios con un solo integrante, como el *daily standup*.
- Algunas historias de usuario son demasiado grandes para un solo desarrollador, pero dividir las en historias más pequeñas desvirtuaría la naturaleza de la historia de usuario. En estos casos se opta por dividir la historia de usuario en subtareas y hacerla en varios *sprints*.

Esto dejaría un proceso siguiendo este orden:

- Se añaden historias de usuario y tareas al *Product Backlog* como se muestra en la ilustración 7.
- Se hace una planificación del siguiente sprint usando las tareas e historias de usuario del *Product Backlog*.
- Cada día se van añadiendo incrementos, pero sin tener un *Daily Standup* como indica la ilustración 9.
- Luego se tiene una reunión con el tutor que sirve a la vez como *Sprint Review*, *Sprint Retrospective*, y en la cual se modifica la planificación del sprint de ser necesario.

Aun así, la filosofía de la metodología scrum sigue utilizándose, el objetivo del *sprint* se cumple y se itera al siguiente sin problema.

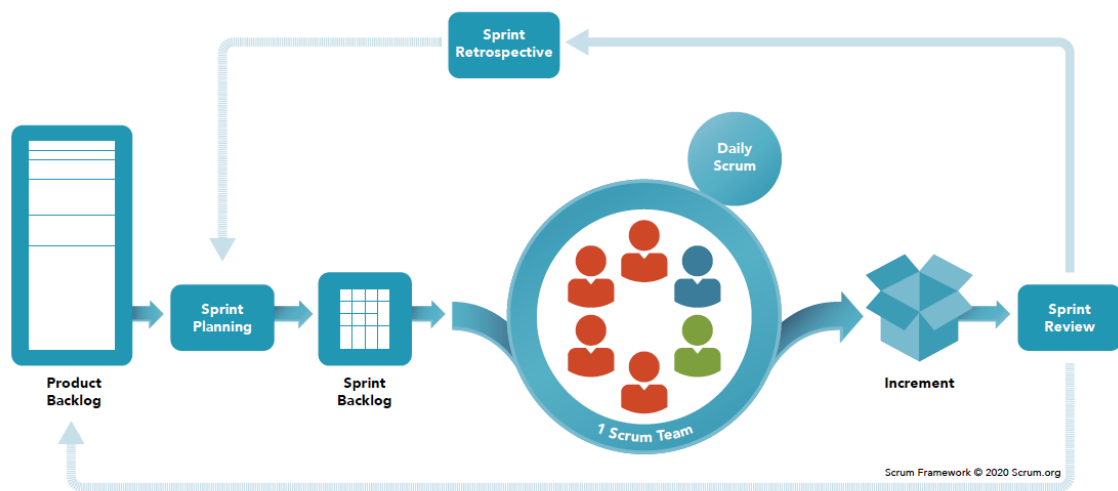


Ilustración 9 - Esquema del framework Scrum [29]

## 4.2. UML

UML o lenguaje unificado de modelado por sus siglas, es el estándar de lenguaje de modelado de sistemas software más popular y utilizado actualmente[30]. Consiste en un set de diagramas, desarrollados para ayudar a los desarrolladores de sistemas y software. Este estándar es ampliamente utilizado durante el Grado de Ingeniería Informática.

Se ha empleado para modelar las entidades de información software que posee el proyecto planteado.

## 4.3. Herramientas CASE

### 4.3.1. Jira

Jira Software [31] es una herramienta de planificación de proyectos y seguimiento de incidencias. Dentro de las empresas del *Fortune 500* el 83% utilizan productos de Atlassian como Jira Software [32]. Con Jira Software se puede realizar la planificación de proyectos para equipos con cualquier metodología ágil, en el caso de este proyecto es SCRUM. Facilita la creación de historias de usuario, tareas, en resumen, la creación del producto backlog, como se muestra en la ilustración 10, y después su integración en los distintos *sprints*. Finalmente facilita los eventos de scrum creando modelos para el *sprint review* y *planning* y también creando automáticamente gráficos como el *sprint burndown* chat. En definitiva, tiene todas las herramientas para la planificación de un proyecto software siguiendo el *framework* de SCRUM.

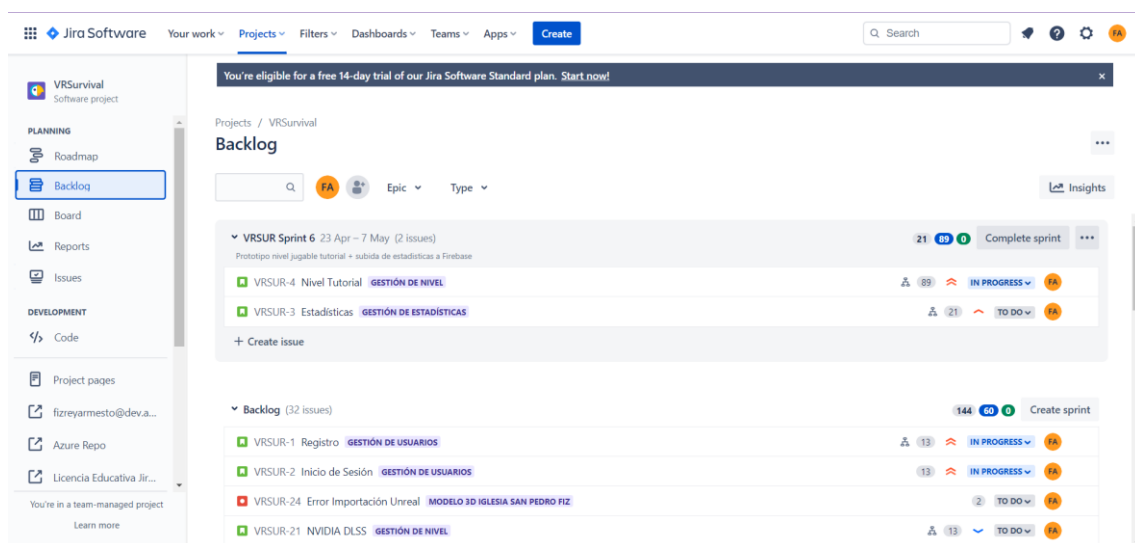


Ilustración 10 - Pestaña del backlog del proyecto VRSurvival

### 4.3.2. Diagrams.net

Diagrams.net [33] es una plataforma online para creación y edición de diagramas que nos permite utilizar todos los elementos de UML para la creación de estos, como el que se muestra en la ilustración 11. Además de su fácil uso, no necesita la instalación de ningún tipo de software ya que se maneja todo desde la web y nos permite guardar los diagramas en la nube. También posee integración con varias plataformas como Jira o Documentos de Google [34] a través de *plugins*.



### 4.3.1. Visual Paradigm

Visual Paradigm [37] es una herramienta para planificación y análisis de proyectos que permite crear diagramas en UML. Se utilizará por su facilidad para realizar diagramas de secuencia y por la experiencia del desarrollador con el programa, ya utilizado durante el Grado de Ingeniería Informática.

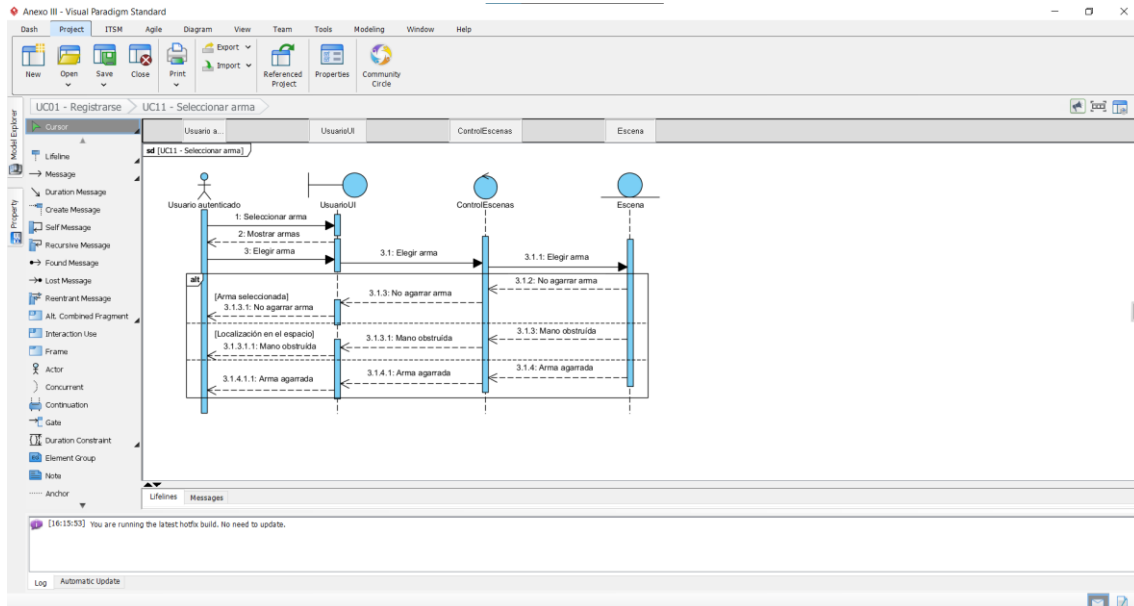


Ilustración 13 - Captura de pantalla de Visual Paradigm

## 4.4. Herramientas de control de versiones

### 4.4.1. Git

Git es el sistema de control de versiones más usado y conocido, aunque en proyectos con archivos binarios muy pesados con UE se suelen usar otras herramientas de control de versiones como Perforce, SVN o Plastic[38]. Se elige Git, en concreto la versión para Windows[39] (que difiere con el resto de las versiones de Git [40]) y no las anteriores, por los siguientes motivos:

- El desarrollador ya está acostumbrado al uso de Git y su utilización no supondría una curva de aprendizaje.
- Hay sistemas de almacenamientos gratuitos en la nube para proyectos de Git.

### 4.4.2. Microsoft Azure Repos

Microsoft Azure [41] es una plataforma online de herramientas en la nube de Microsoft, pero nos vamos a centrar en la herramienta Repos, la cual permite almacenar repositorios en la nube de Microsoft, de la misma forma que se haría con GitHub o cualquier otra plataforma. Durante el desarrollo del proyecto primero se utilizó la plataforma GitHub [42] para el almacenamiento del repositorio de Git, pero se abandonó debido al exceso del límite de peso del repositorio. A continuación, el remote del proyecto se cambió a Microsoft Azure, que no tiene límite de peso del repositorio para proyectos con las características de VRSurvival, que tiene un repositorio que ocupa más de 100Gb.

### 4.4.3. GitKraken

GitKraken [43] es una interfaz gráfica de usuario de usuario para Git. Como el desarrollo de la aplicación se realiza en Windows se prefiere el uso de una GUI (*Graphical User Interface*). Aunque se utilice la CLI (*Command Line Interface*) para la ejecución de los comandos realizados anteriormente, GitKraken ofrece una interfaz para ver el progreso de los *commits* y las ramas de una forma muy intuitiva, como se puede observar en la ilustración 14.

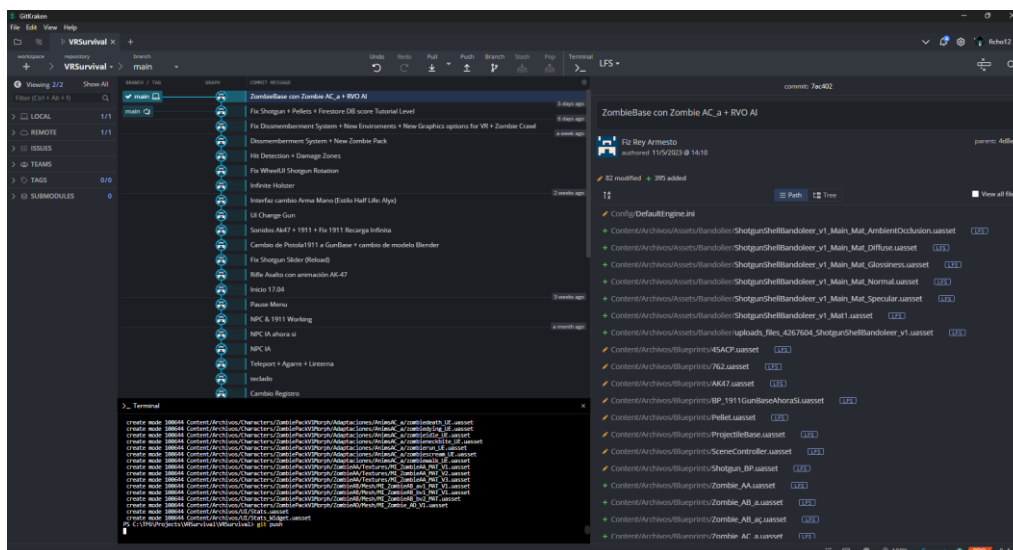


Ilustración 14 - Captura de pantalla de GitKraken con el proyecto VRSurvival

## 4.5. Herramientas de desarrollo

### 4.5.1. Unreal Engine 5

Unreal Engine 5 [13] es la última versión del motor UE, que salió de manera oficial en su versión 5.0 el 5 de abril de 2022. Uno de sus elementos principales es la tecnología de geometría virtualizada Nanite, que permite que modelos 3D fotorrealistas de gran detalle sean importados en videojuegos. El objetivo de UE5 es que sea lo más sencillo posible para los desarrolladores crear mundos jugables de gran detalle sin tener que pasar mucho tiempo creando *assets* con gran detalle[44]. Nanite permite importar casi cualquier modelo 3D preexistente, este después maneja automáticamente los niveles de detalle (LODs) de estos objetos 3D de manera apropiada para el rendimiento necesario para la máquina en la que se está ejecutando y de acuerdo con la distancia de dibujado. El manejo automático de los LODs es algo que tendría que hacer un artista manualmente y que consumiría mucho tiempo de no ser por esta tecnología. El uso de Mapas virtuales de sombras (*Virtual Shadow Maps*) es un nuevo método de *shadow mapping* que se utiliza para renderizar sombras detalladas de alta resolución y arregla el típico error de cascadas de sombras con el método tradicional de mapas de sombras. Esta nueva tecnología es necesaria para el uso de Nanite. UE5 también incorpora un nuevo método de cálculo de iluminación en tiempo real llamado Lumen, pero no se va a utilizar en este proyecto por su gran coste computacional combinado con el uso de Nanite en un proyecto de RV. Para que un proyecto en RV con Nanite y Lumen funcione de forma estable se necesitaría un equipo muy potente, como mínimo una gráfica NVIDIA Geforce RTX 4080 o superior [45]. Además, en el estado actual, la combinación de estas tecnologías en RV es inestable para un desarrollo destinado a producción.

### 4.5.2. Blueprints

Según la documentación de Unreal Engine 5.1 en inglés [46], y traducido:

*“El sistema de Scripting Visual de Blueprints en Unreal Engine es un sistema de scripting de gameplay basado en el concepto de usar interfaces gráficas basadas en nodos para crear elementos de gameplay dentro del editor de Unreal. Como en muchos lenguajes de scripting, se usa para definir clases u objetos empleando un paradigma orientado a objeto (OO) con el que está construido el motor. A medida que uses Unreal, te encontrarás habitualmente que los objetos definidos usando Blueprint se refieren coloquialmente simplemente como “Blueprints”.*

*Este sistema es extremadamente flexible y poderoso ya que provee a los diseñadores con la habilidad de usar virtualmente todo el rango de conceptos y herramientas generalmente solo disponibles para programadores. Adicionalmente, el mercado específico de Blueprint disponible en la implementación de C++ de Unreal Engine permite a los programadores crear sistemas base que pueden ser extendidos por diseñadores”.*

Como se muestra en la ilustración 15, se utiliza la programación visual para crear todo tipo de lógica, siendo posible plasmar prácticamente cualquier idea tanto en *blueprints* como en C++ (con algunas excepciones). En el caso de la ilustración 15 se muestra la detección de colisión de uno de los ataques del Zombie. Este *blueprint* es muy interesante ya que hacer lo mismo en un lenguaje de más bajo nivel como C++ llevaría mucho más

tiempo. Por razones como esta se utilizan *blueprints* por encima de C++, ya que es una opción muy buena para hacer prototipado. Una vez comienza el desarrollo con el prototipado en *blueprints* hecho, se traducen las partes más intensivas computacionalmente a C++ y se combinan con la lógica de *blueprints*. Así se puede tener el rendimiento de C++ y la facilidad y rapidez de utilización de las *blueprints*, que permite a diseñadores y demás personas que no son ingenieros de *gameplay* modificar la lógica del videojuego a su estilo. De esta misma forma se pueden aplicar muchos *plugins* y *frameworks* ya creados desde la tienda de Unreal o desde código fuente como es el caso de VRExpansionPlugin, utilizado en este proyecto.

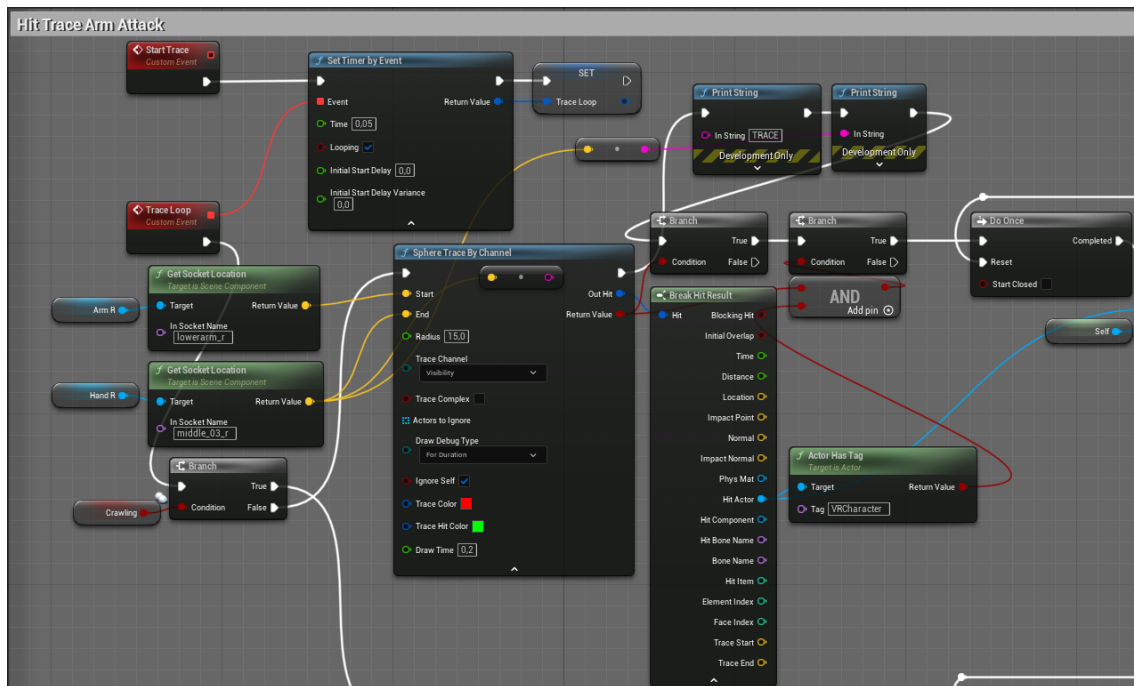


Ilustración 15 - Fragmento de la clase Blueprint ZombieBase

### 4.5.3. VRExpansionPlugin

VRExpansionPlugin [47] es un *framework* de UE para el desarrollo de experiencias en RV con licencia MIT. Este plugin facilita las interacciones y eventos del *gameplay* en RV, centrándose en el multijugador, *networking*, *locomotion* y *gripping* añadiendo funcionalidad a OpenVR, SteamVR y OpenXR. Se utilizará por ser un muy buen punto de partida para un proyecto en RV en UE, por su amplia documentación y por el excelente soporte del creador a través de Discord (llega a resolver dudas de la comunidad en pocas horas).

### 4.5.4. Visual Studio 2022

Se utilizará el popular IDE Visual Studio 2022 [48] por ser la última versión del IDE con integración con UE5. Aunque el desarrollo del videojuego se realiza con *Blueprints* de UE y no con código de C++, algunos de los *plugins* y *assets* que se utilizan en el proyecto necesitan ser compilados, por lo que se necesita que el IDE genere los archivos de proyecto y después compile la solución, como se puede observar en la ilustración 16.

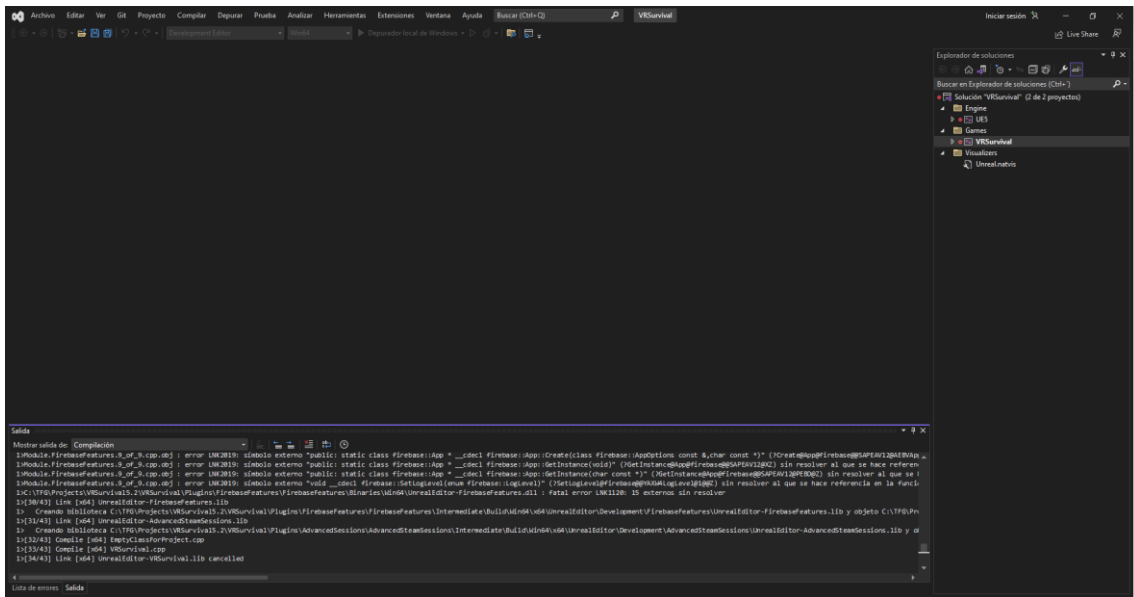


Ilustración 16 - Recompilación del proyecto VRSurvival para la versión 5.2 de UE

### 4.5.5. Firebase

Firebase [49] es una plataforma en la nube para el desarrollo de aplicaciones. Su principal función es la facilitación del desarrollo de aplicaciones, para ello ofrece distintos servicios como autenticación, anuncios o manejo de estadísticas. Para el desarrollo de este proyecto se utilizará el servicio de autenticación, la BBDD Firestore, como se muestra en la Ilustración 16; y Functions (para *scripting* en la nube).

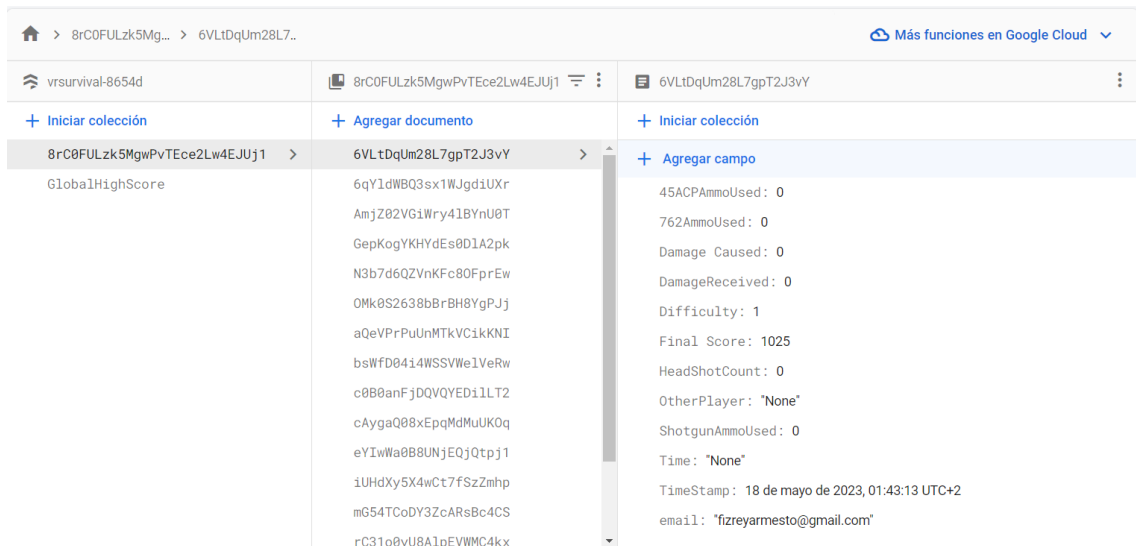
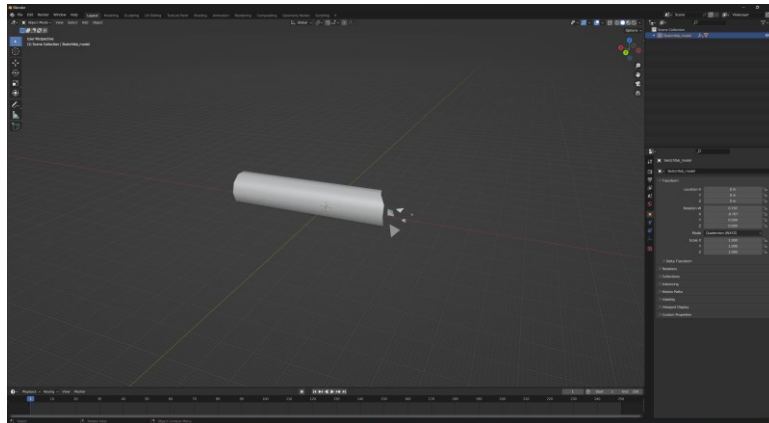


Ilustración 17 - Base de datos Firestore

## 4.6. Herramientas Modelado y Animación 3D

### 4.6.1. Blender

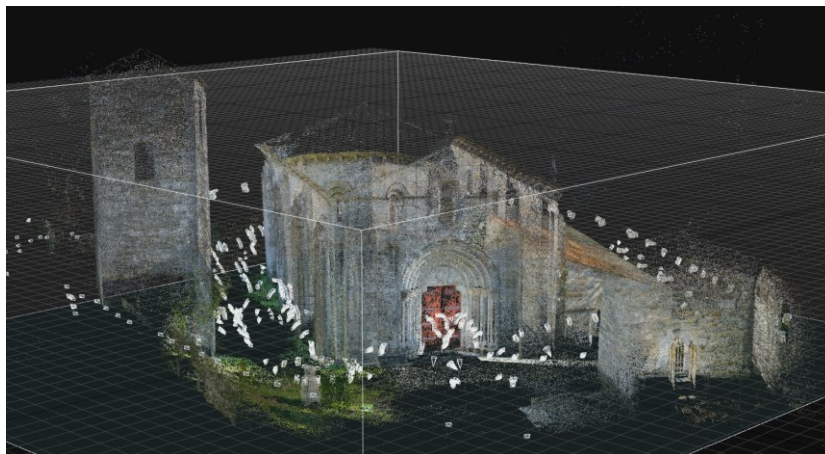
Blender [50] es una suite de creación 3D gratuita y *open source*, que permite el modelado, *rigging*, animación, etc. de modelos 3D. Para este proyecto se va a usar para la adaptación de modelos 3D obtenidos en internet, por ejemplo, el modelo que se puede observar en la ilustración 18 se editó para borrar los trozos de tiza esparcidos por el modelo y dejar una sola pieza central de tiza. Se elige esta herramienta y no otra por su amplia popularidad, gran facilidad de aprendizaje dado el gran número de tutoriales disponibles de la herramienta y su integración con UE.



*Ilustración 18 - Edición de un modelo 3D en Blender*

### 4.6.2. RealityCapture

RealityCapture [51] es una herramienta software para la creación de modelos 3D con fotogrametría, actualmente es la herramienta más rápida del mercado. Esta herramienta se usa para la creación de los modelos 3D de la iglesia de San Pedro Fiz de Hospital do Incio, como se puede observar en la ilustración 19, que se utilizarán en los niveles del videojuego. Se utiliza esta herramienta de fotogrametría por su gran integración con UE5 (Epic Games compró la empresa creadora de RealityCapture, Capturing Reality, en 2021 [52]) y por la gran cantidad de tutoriales de alta calidad que existen de la herramienta, como del desarrollado por William Faucher [53]



*Ilustración 19 - Nube de puntos de la iglesia de San Pedro Fiz de Hospital do Incio en RealityCapture*

### 4.6.3. Mixamo

Mixamo [54] es una herramienta software online de Adobe para obtención de personajes y animaciones, tiene una herramienta de *auto-rigging* que permite subir personajes con un esqueleto personalizado y utilizarlo con sus animaciones sin ningún problema. Esta herramienta se utilizará para animar los enemigos del videojuego.

### 4.6.4. Mixamo Converter

Mixamo Converter [55] es una herramienta que permite convertir las animaciones de Mixamo en animaciones válidas para esqueletos de UE, ya que la jerarquía de los esqueletos usados por ambas no coincide al exportar las animaciones. Esta herramienta ahorra el trabajo que tendría que hacer el desarrollador del *retargeting* de los esqueletos de las animaciones.

## 4.7. Herramientas gráficas

### 4.7.1. Adobe Lightroom

Adobe Lightroom [56] es una herramienta de edición de fotografías muy popular que se utilizará en el proyecto para editar las fotos realizadas para el modelo de fotogrametría de la iglesia de San Pedro Fiz de Hospital do Incio. Se utiliza este programa de edición y no otro por la experiencia del desarrollador con él.

### 4.7.2. Adobe Photoshop

Adobe Photoshop [57] es una herramienta de edición de fotografías muy popular que se utilizará en el proyecto para editar las fotos utilizadas en el videojuego para menús, como podemos ver en la ilustración 20, y para editar texturas. Se utiliza este programa de edición y no otro por la experiencia del desarrollador con él.

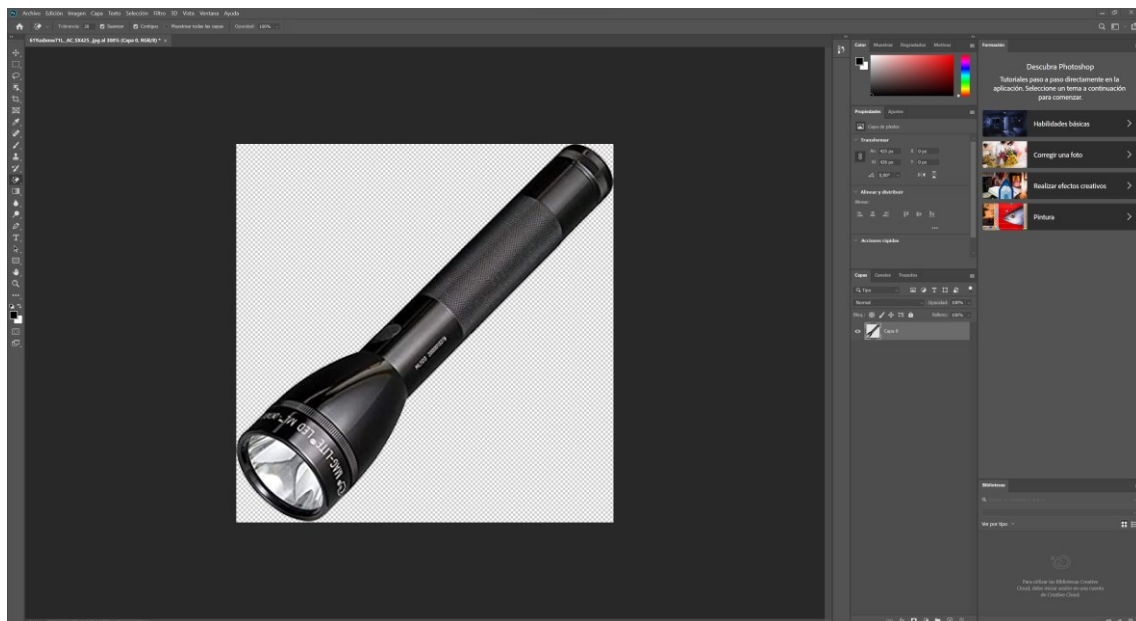


Ilustración 20 - Edición de una foto utilizada en uno de los menús en Adobe Photoshop

## 5. Aspectos Relevantes

En este apartado se explican las partes más importantes del proyecto. Se presentan las etapas de desarrollo del proyecto (detalladas en los anexos técnicos de esta documentación) y después se sigue con los apartados de Trabajos Relacionados y Problemas Durante el Desarrollo y Soluciones.

### 5.1. Planificación Temporal y marco de trabajo

Para la planificación temporal se ha creado un calendario de trabajo, como se muestra en la ilustración 21.

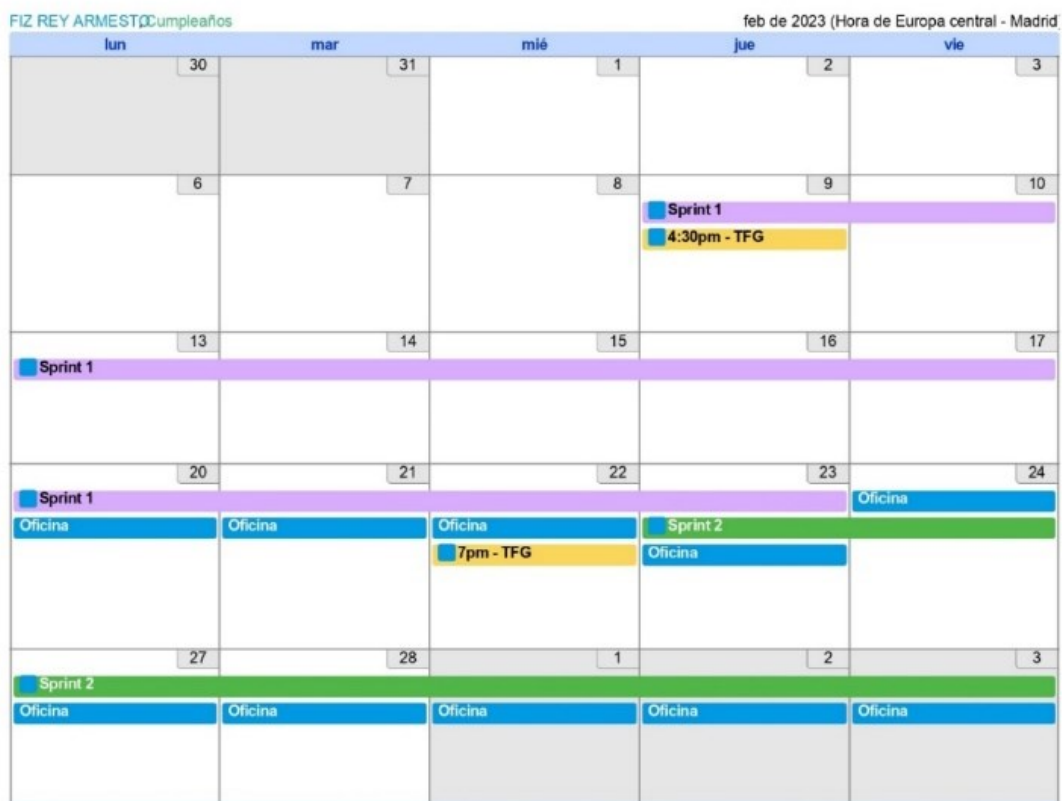


Ilustración 21 - Calendario: Febrero

Además, se han creado *burndown charts* para cada sprint, como se muestra en la ilustración 20. A continuación, se muestra el resumen del primer *sprint* del proyecto.

**Objetivo del sprint:** Sacar fotos modelo 3D. Instalar los *plugins* y conocer cómo funcionan. Ponerse al día con Unreal Engine 5 y viabilidad del proyecto.

Sprint 1	Sprint Story point estimate change	Sprint Story point estimate completed	Sprint Story point estimate not completed
VRSUR-18 Fotogrametría - Sacar suficientes fotos en todos los ángulos	4.00	4.00	0.00
VRSUR-19 Editar Fotos	2.00	2.00	0.00
VRSUR-56 Instalación Plugin VRExpansionPlugin	1.00	1.00	0.00
VRSUR-57 Instalación FirebaseFeatures	1.00	1.00	0.00
VRSUR-58 Puesta al día con Unreal Engine 5.1	4.00	4.00	0.00
VRSUR-59 Creación de un repositorio git con remoto en Github	1.00	1.00	0.00
VRSUR-61 Instalación de la herramienta de control de Git GithubDesktop	1.00	1.00	0.00
<b>Total</b>	<b>14.00</b>	<b>14.00</b>	<b>0.00</b>

Tabla 6 - Sprint 1

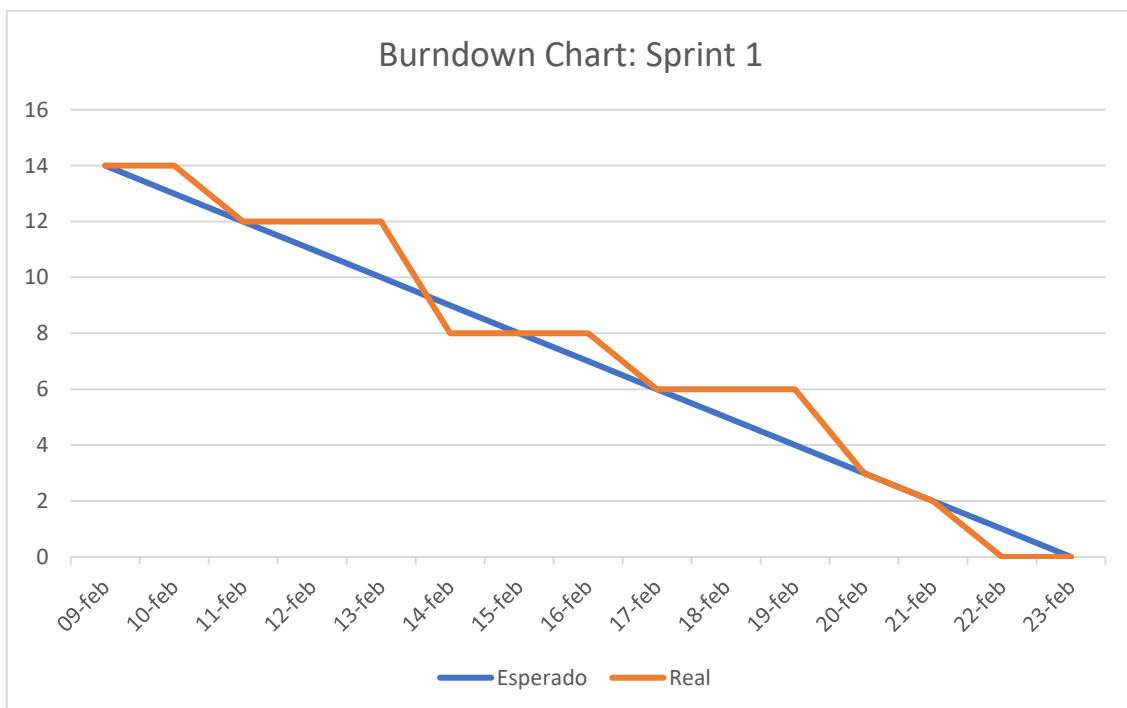


Ilustración 22 - Burndown Chart: Sprint 1

Además, se ha realizado un *product roadmap*, como se muestra en la ilustración 23, para visualizar cómo se espera que evolucionará la solución con el paso del tiempo.

## Especificación de Requisitos del Sistema

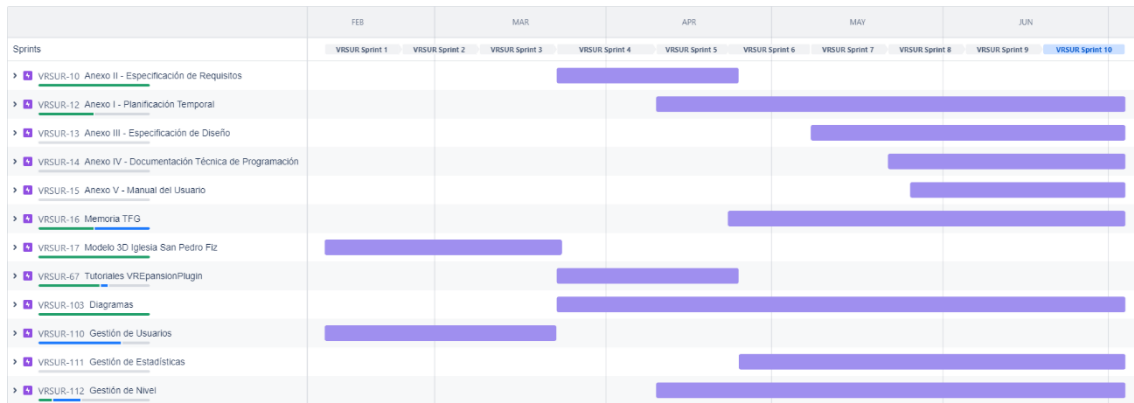


Ilustración 23- Roadmap

Para la realización del proyecto se ha utilizado el *framework* SCRUM y como herramienta para ponerlo en práctica se ha utilizado la plataforma Jira. La planificación temporal se detalla en profundidad en el Anexo I.

## 5.2. Especificación de Requisitos del Sistema

Para la especificación de requisitos del sistema y modelo de análisis a desarrollar se ha utilizado la metodología SCRUM. Para exponer la especificación de requisitos se ha utilizado la metodología para la elicitación de requisitos de sistemas software de Durán y Bernárdez[58].

Para la especificación de los requisitos de información se realizaron las siguientes tablas, como se muestra en la tabla 7.

IRQ-1	Usuarios
<b>Versión</b>	1.0
<b>Autores</b>	Fiz Rey Armesto
<b>Fuentes</b>	Ninguna
<b>Dependencias</b>	IRQ-002 Estadísticas de partida
<b>Descripción</b>	El sistema deberá almacenar la siguiente información sobre cada usuario.
<b>Datos específicos</b>	<ul style="list-style-type: none"> <li>• Identificador único</li> <li>• Nombre de usuario</li> <li>• Email</li> <li>• Contraseña</li> <li>• Estadísticas de partida</li> </ul>
<b>Importancia</b>	Vital
<b>Urgencia</b>	Alta
<b>Estado</b>	Validado
<b>Estabilidad</b>	Alta
<b>Comentarios</b>	Ninguno

Tabla 7 - IRQ-001 Usuarios

Se realiza un diagrama arquitectónico del sistema, como se muestra en la ilustración 24.

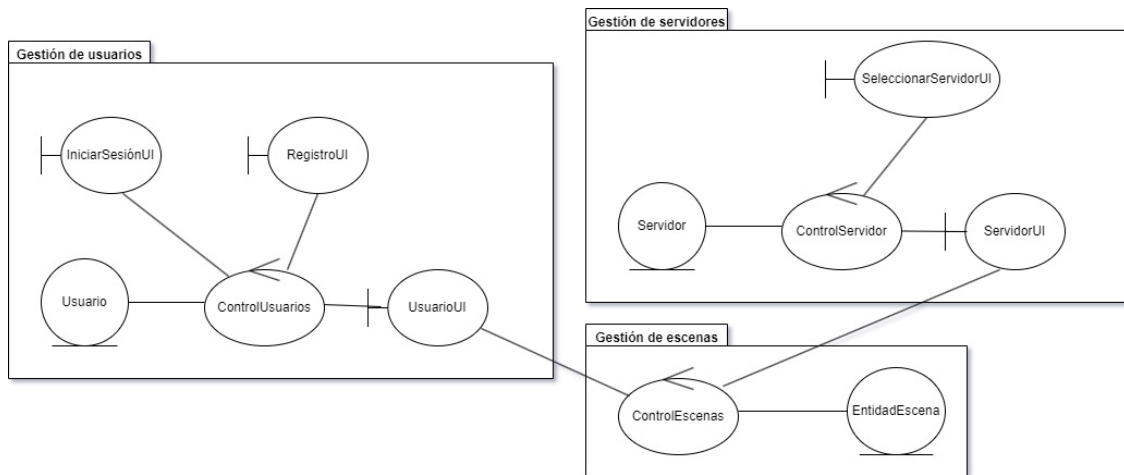


Ilustración 24 - Diagrama Arquitectónico

Para la descripción de los requisitos funcionales se utilizaron las historias de usuario de SCRUM, estas se obtienen de la página de Jira del proyecto y son del estilo de la Tabla 8 que se muestra a continuación:

<b>[VRSUR-116] Nivel de Oleadas Infinito</b>			
<b>Status:</b>	To Do		
<b>Project:</b>	<a href="#">VRSurvival</a>		
<b>Type:</b>	Story	<b>Priority:</b>	Medium
<b>Reporter:</b>	<a href="#">FIZ REY ARMESTO</a>	<b>Assignee:</b>	<a href="#">FIZ REY ARMESTO</a>
<b>Story estimate:</b>	point 55		
<b>Change risk:</b>	Critical		
<b>Validation:</b>	<ul style="list-style-type: none"> <li>• Dificultad incremental con el paso del tiempo, pero conservando la posibilidad de que se supere el nivel (no hacer un nivel insuperable por el usuario).</li> <li>• Realizado de forma que sea fácil diseñar otro nivel con las mismas herramientas.</li> <li>• Sistema de rondas y <i>spawners</i> de enemigos.</li> </ul>		
<b>Description</b>	Como Usuario quiero jugar un nivel de oleadas infinito		

Tabla 8 – VRSUR-116

Los casos de uso se muestran en su respectiva tabla de Durán y Bernárdez como se ve en la tabla 9 y de cada paquete se realiza un diagrama de casos de uso como se ve en la ilustración 25.

## Especificación de Requisitos del Sistema

<b>UC-01</b>	<b>Registro</b>	
<b>Versión</b>	1.0	
<b>Autores</b>	Fiz Rey Armesto	
<b>Fuentes</b>	-	
<b>Dependencias</b>	<ul style="list-style-type: none"> <li>• [IRQ-1] Usuarios</li> <li>• [VRSUR-1] Registro</li> </ul>	
<b>Descripción</b>	El usuario se registra en el sistema con un email y contraseña.	
<b>Precondición</b>	El usuario no está registrado con el mismo email.	
<b>Secuencia normal</b>	<b>Paso</b>	<b>Acción</b>
	1	El ACT-1 solicita al sistema realizar un registro.
	2	El sistema solicita al usuario los datos de email y contraseña.
	3	El ACT-1 proporciona los datos solicitados y solicita el registro
	4	El sistema comprueba que los datos son válidos y registra, autentica y muestra el menú de usuario autenticado al ACT-1.
<b>Postcondición</b>	El usuario se autentica.	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	4	Si los datos no son válidos, no se permite el registro y se comunica el error al ACT-1.
<b>Frecuencia</b>	Baja	
<b>Importancia</b>	Vital	
<b>Urgencia</b>	Alta	
<b>Estado</b>	Validado	
<b>Estabilidad</b>	Alta	
<b>Comentarios</b>	Ninguno	

Tabla 9 - UC-001: Registrarse



Ilustración 25 - Diagrama de casos de uso: Gestión de usuarios

Para finalizar la especificación de requisitos se muestran las tablas de los requisitos no funcionales, como se muestra en la tabla 10 y por último se muestra una Matriz de Rastreabilidad.

<b>RNF-1</b>	<b>Aplicación Nativa</b>
<b>Versión</b>	1.0
<b>Autores</b>	Fiz Rey Armesto
<b>Fuentes</b>	-
<b>Dependencias</b>	Ninguna
<b>Descripción</b>	La aplicación será nativa para PCVR (Windows) <sup>1</sup>
<b>Importancia</b>	Vital
<b>Urgencia</b>	Alta
<b>Estado</b>	Validado
<b>Estabilidad</b>	Alta
<b>Comentarios</b>	Ninguno

*Tabla 10 - RNF-01*

La especificación de requisitos del sistema se detalla en profundidad en el Anexo II.

---

1. Aunque la mayoría de los videojuegos de PC que se desarrollan solo tienen versión para Windows, con la capa de compatibilidad Proton para Linux se pueden ejecutar juegos nativos de Windows en Linux con una pérdida mínima de rendimiento e incluso mejora de este en algunos casos [77]

### 5.3. Análisis y Diseño del Software

Para exponer la especificación de diseño se utiliza el lenguaje unificado de modelado, UML [59]. Para comenzar el análisis se muestra un diagrama del modelo del dominio, como se puede ver en la ilustración 26.

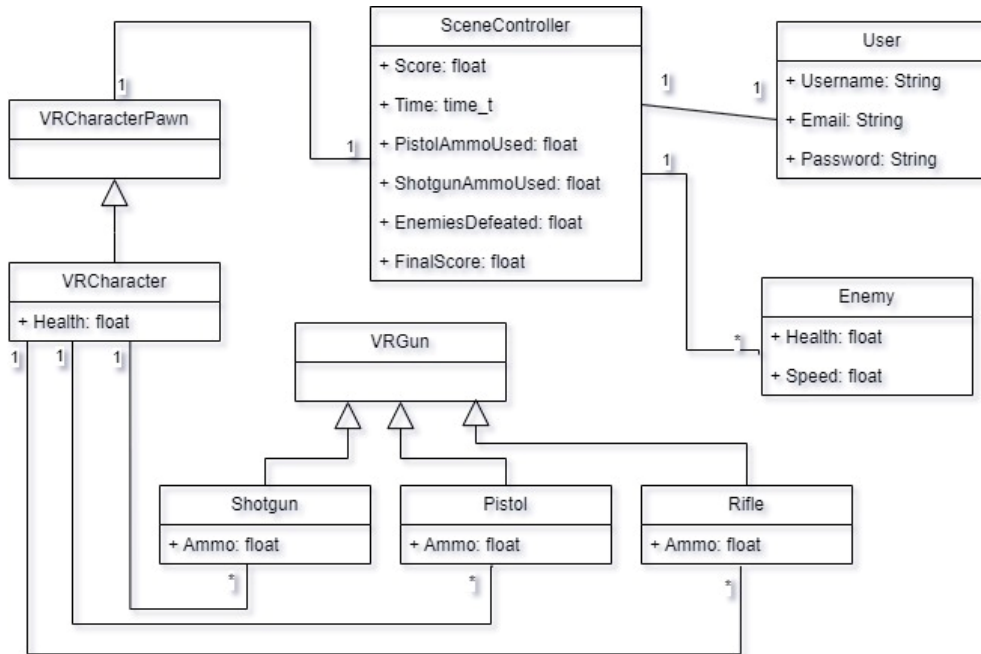


Ilustración 26 - Diagrama del modelo del dominio

Después se amplía la información sobre los casos de uso del Anexo II con diagramas de secuencia, como el que se muestra en la ilustración 27.

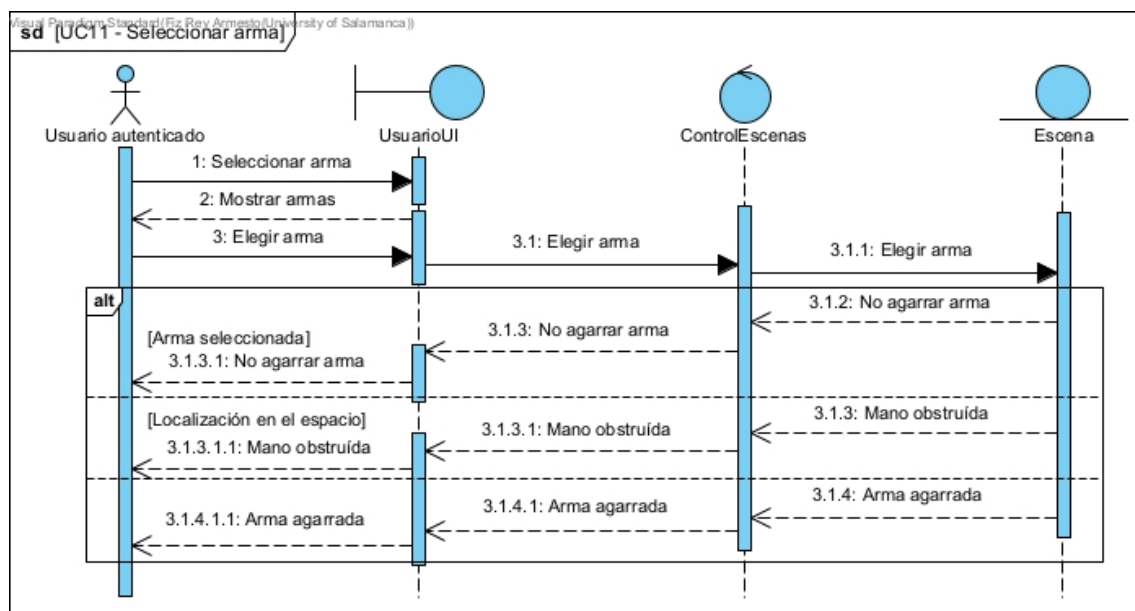


Ilustración 27 - Diagrama de secuencia: UC11 - Seleccionar arma

La parte de diseño comienza con una explicación de la arquitectura de UE y después se muestra el diagrama de niveles del juego realizado, que se muestra en la ilustración 28.

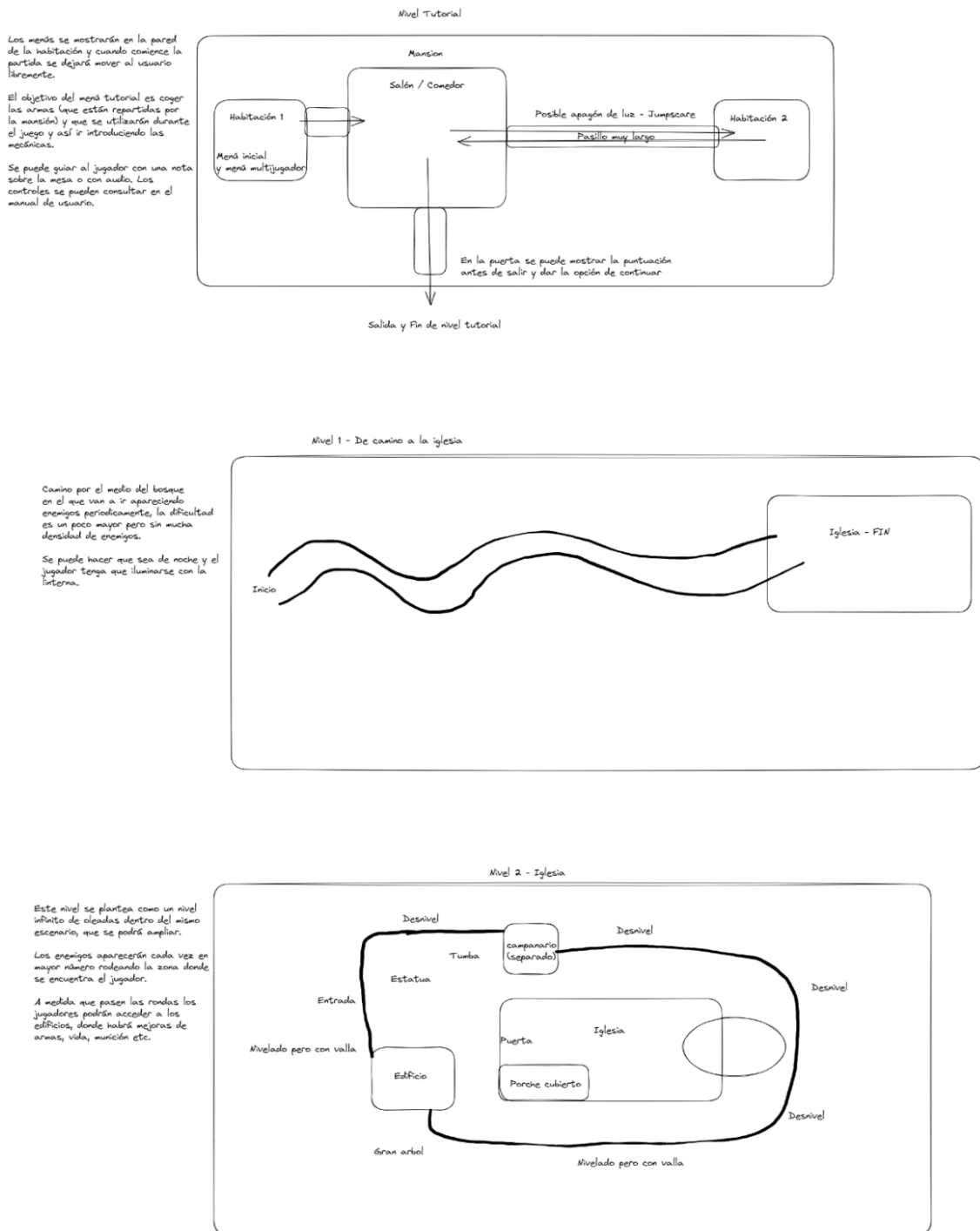


Ilustración 28 - Diagrama de niveles

A continuación, se explica el diseño de los modelos 3D realizados en RealityCapture mediante la técnica de fotogrametría y después comienza la explicación de las clases y la lógica utilizada en UE. Dentro de esta, cabe destacar la clase *ZombieBase*, por la que pasa la mayor parte de la lógica del videojuego. Se hace hincapié en la inteligencia artificial utilizada para la ejecución de los distintos comportamientos de los enemigos. La creación de esta inteligencia artificial supone la utilización de distintos patrones de diseño como los patrones *composite*, *observer* y *decorator*. La jerarquía de los distintos comportamientos se puede ver en el *Behaviour Tree* o árbol de comportamiento de la ilustración 29.

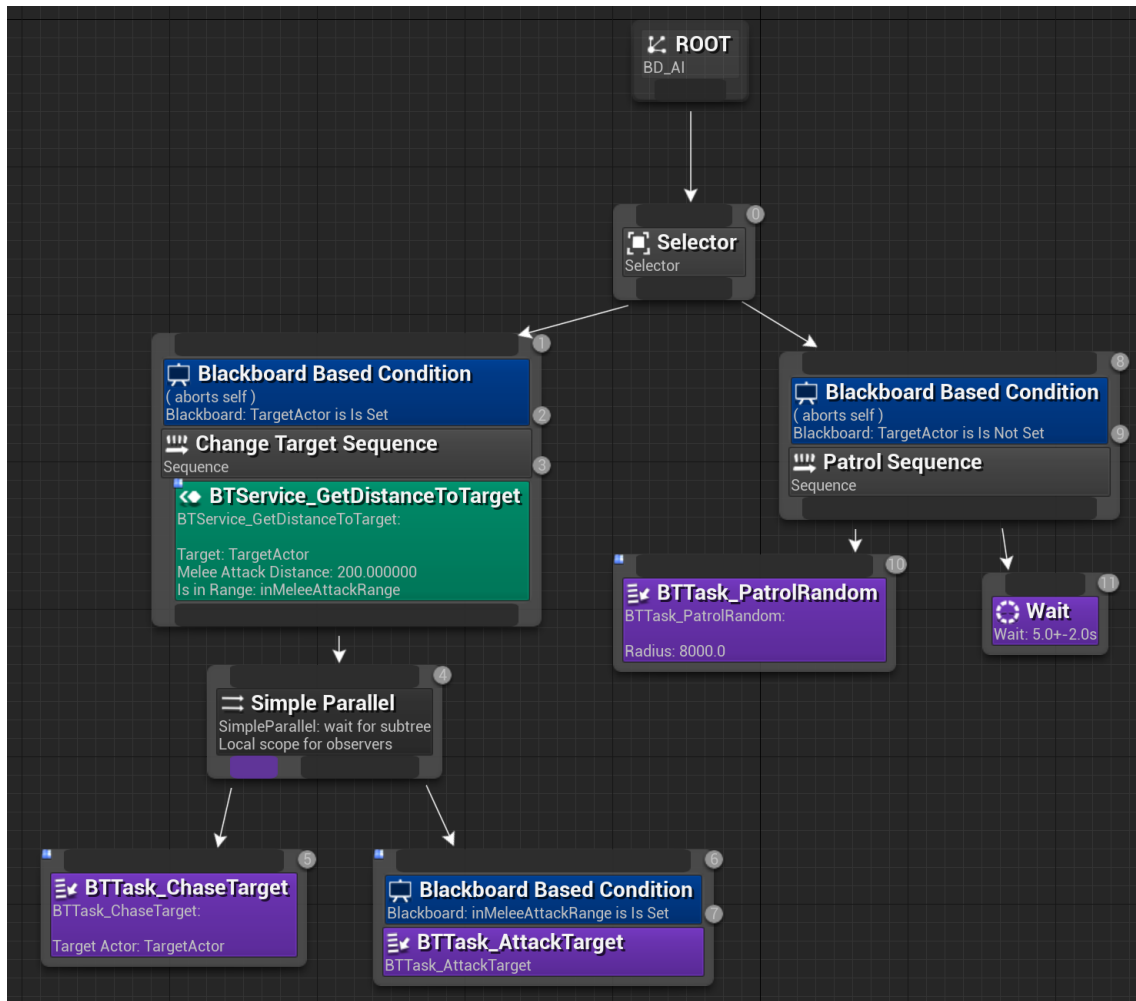


Ilustración 29 - Behaviour Tree de ZombieBase

A continuación, se muestra el diseño de la plataforma Firebase para el proyecto, de la cual se utilizan las herramientas *Auth*, *Firestore Database* y *Cloud Functions*. De este apartado se puede destacar la estructura que se elige para la BBDD, que se muestra en la ilustración 30.

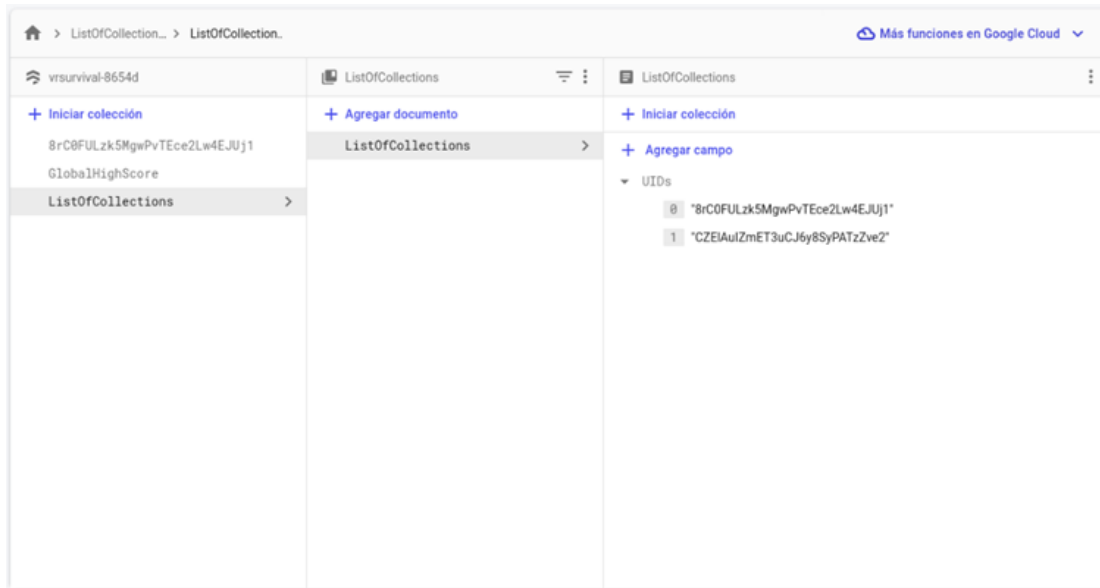


Ilustración 30 - Firestore Database

De la misma forma que se habían realizado los diagramas de secuencia para el apartado de análisis, también se realizan para el apartado de diseño, como se muestra en la ilustración 31.

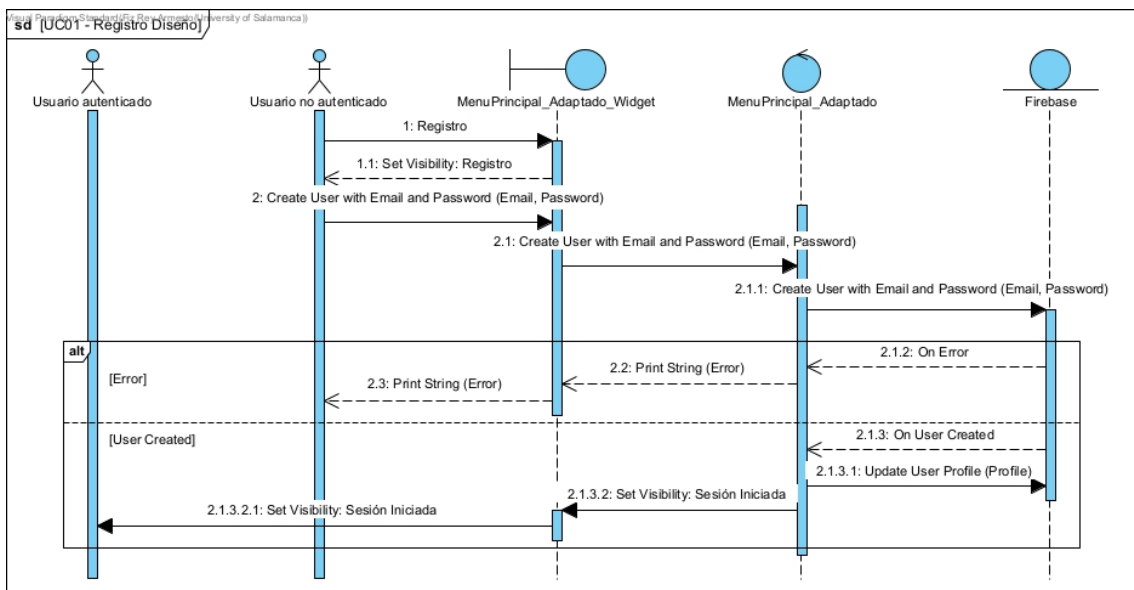


Ilustración 31- Diagrama de secuencia: UC01 - Registro

Para finalizar el anexo, se realiza un diagrama de despliegue del proyecto, que se muestra en la ilustración 32.

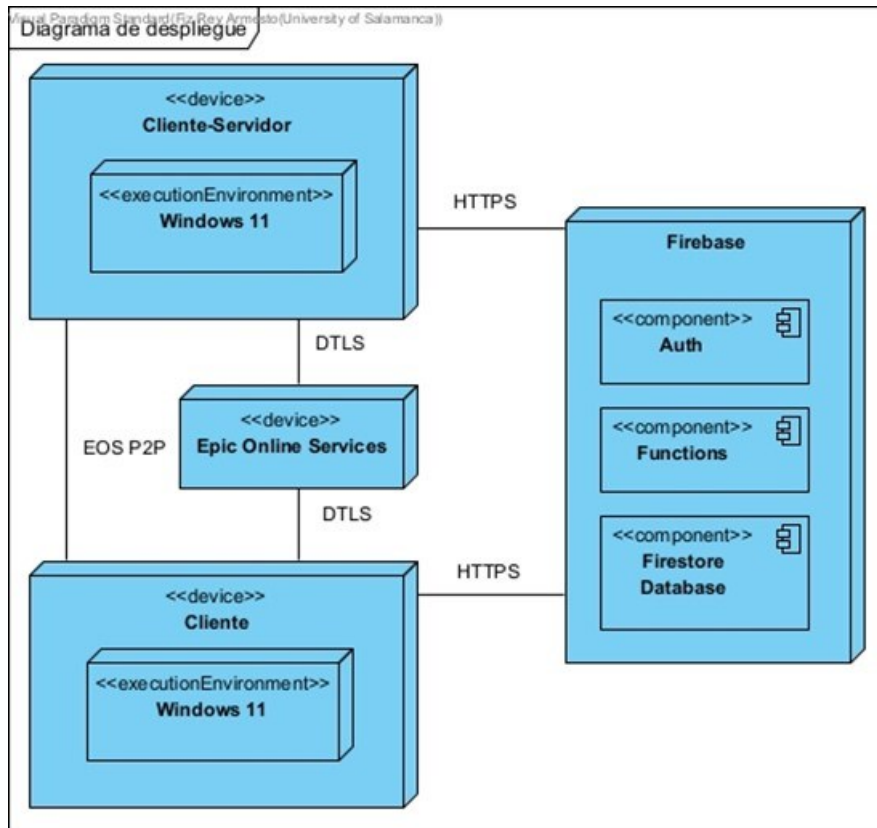


Ilustración 32 - Diagrama de despliegue

En la parte de pruebas se explican las pruebas realizadas de tipo *End to End* y pruebas de integración, haciendo uso de la herramienta de simulación de blueprints, como se muestra en la ilustración 33.

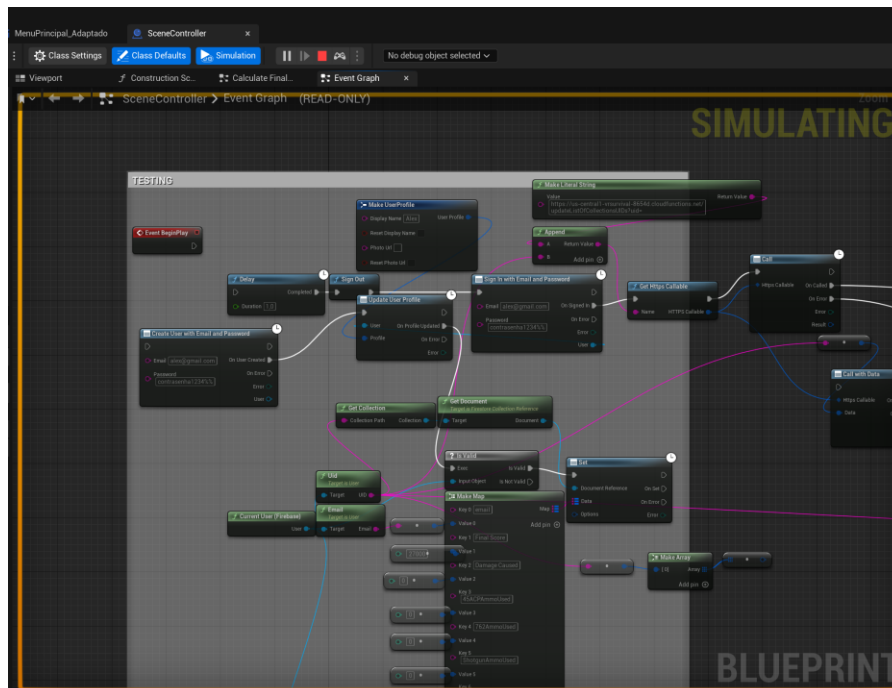


Ilustración 33 - Simulación de una Blueprint

El análisis y el diseño del software se detalla en profundidad en el Anexo III.

## 5.4. Documentación Técnica de Programación

La documentación técnica de programación comienza explicando la obtención del proyecto desde su repositorio de Azure Repos y cómo se utilizó Git durante el desarrollo. A continuación, se muestra la forma de compilar el proyecto manualmente, ya que al tener *plugins* de código fuente es necesario compilarlos manualmente desde Visual Studio, como se muestra en la ilustración 34.

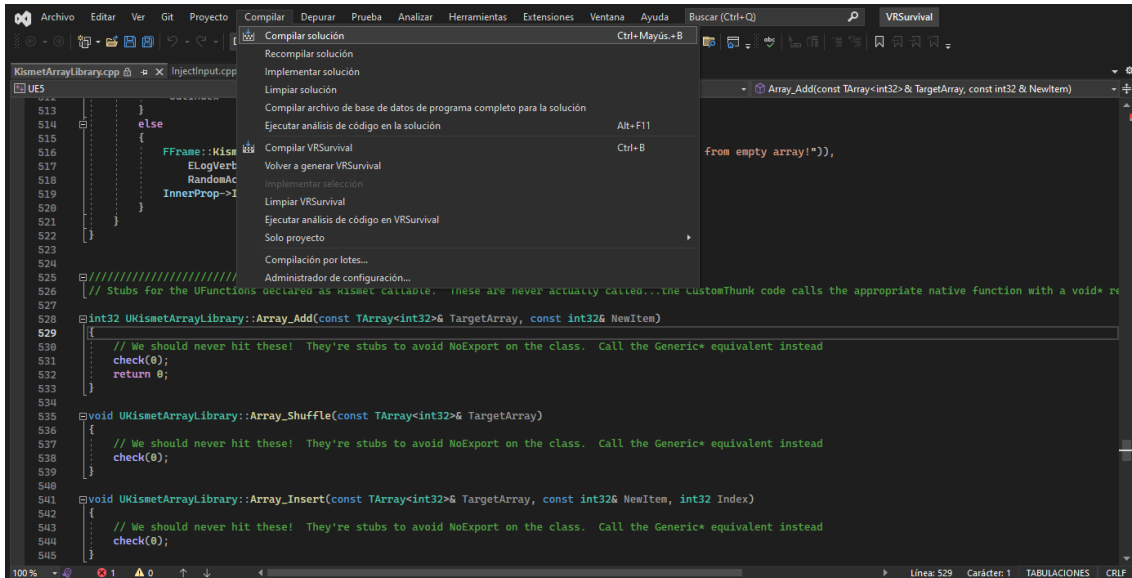


Ilustración 34 - Compilación del proyecto en Visual Studio

A continuación, se explica el despliegue del servidor *Firebase* junto con el plugin *Firebase Features* que se utiliza en el proyecto. Primero, se necesita descargar el archivo `Google-services.json` de la página de *Firebase* del proyecto y colocarlo en una carpeta llamada `Services` en la carpeta raíz del proyecto, si esta carpeta no existe ha de ser creada.

Después se explica el despliegue de un proyecto de *Firebase* en una plataforma Linux, esto será necesario para modificar cualquiera de las *Cloud Functions* del proyecto. A continuación se muestran todos los comandos utilizados en un sistema `POP!_OS 21.04 LTS`:

```
sudo apt install npm
npm init -y
npm i firebase
npm -g firebase-tools
firebase login
firebase init
```

Una vez realizados estos comandos estaría finalizado el proceso de instalación del proyecto de *firebase*. A continuación, una vez modificadas las *cloud functions* en el archivo `index.js`, como se muestra en la ilustración 35, habría que utilizar el siguiente comando para desplegar las *cloud functions* a la nube de *firebase*:

### firebase deploy --only functions

```
JS index.js x
updateGlobalHighScoresHTTPS > functions.https.onRequest() callback > promises > uidList.map() callback > userScoresSnapshot
58   const listOfCollectionsDoc = await db.collection("ListOfCollections")
59     .doc("ListOfCollections")
60     .get();
61   if (!listOfCollectionsDoc.exists) {
62     console.log("1. ListOfCollections document not found");
63     throw new Error("1. ListOfCollections document not found error");
64   } else {
65     console.log("1. ListOfCollections document found");
66   }
67
68   const uidList = listOfCollectionsDoc.data().UIDs;
69
70   // Fetch the highest scores for each user
71   const promises = uidList.map(async (uid) => {
72     try {
73       const userScoresRef = db.collection(uid);
74       const userScoresQuery = userScoresRef
75         .orderBy("Final Score", "desc")
76         .limit(5);
77       const userScoresSnapshot = await userScoresQuery.get();
78
79       const userScores = [];
80       userScoresSnapshot.forEach((doc) => {
81         const score = doc.data()["Final Score"];
82         const user = doc.data().email;
83         console.log("Score: ", score);
84         console.log("User: ", user);
85         if (score !== undefined) {
86           userScores.push({score, user});
87         }
88       });
89     } catch (error) {
90       console.error("Error fetching scores for user:", uid, error);
91     }
92   });
93   await Promise.all(promises);
94 }
```

Ilustración 35 - Archivo index.js del proyecto Firebase

A continuación, se enlazan los repositorios que se crean para el control de versiones del proyecto:

[https://dev.azure.com/fizreyarmesto/\\_git/VRSurvival](https://dev.azure.com/fizreyarmesto/_git/VRSurvival)

<https://github.com/ficho12/Firebase>

La documentación técnica de programación se detalla en profundidad en el Anexo IV.

## 5.5. Manual del Usuario

Una de las partes cruciales del manual de usuario es el tráiler del videojuego, en este, se puede apreciar toda la funcionalidad y las interacciones del usuario que ocurren en todo el proyecto. El video está disponible en el siguiente enlace de YouTube, como se muestra en la ilustración 36.

<https://www.youtube.com/watch?v=VH80UwQjDwA>



*Ilustración 36 - Tráiler del videojuego en Youtube*

Se advierte al usuario de los posibles riesgos de seguridad y salud, como pueden ser las posibles convulsiones por epilepsia fotosensible o lesiones por la falta de precaución sobre la percepción del espacio jugable.

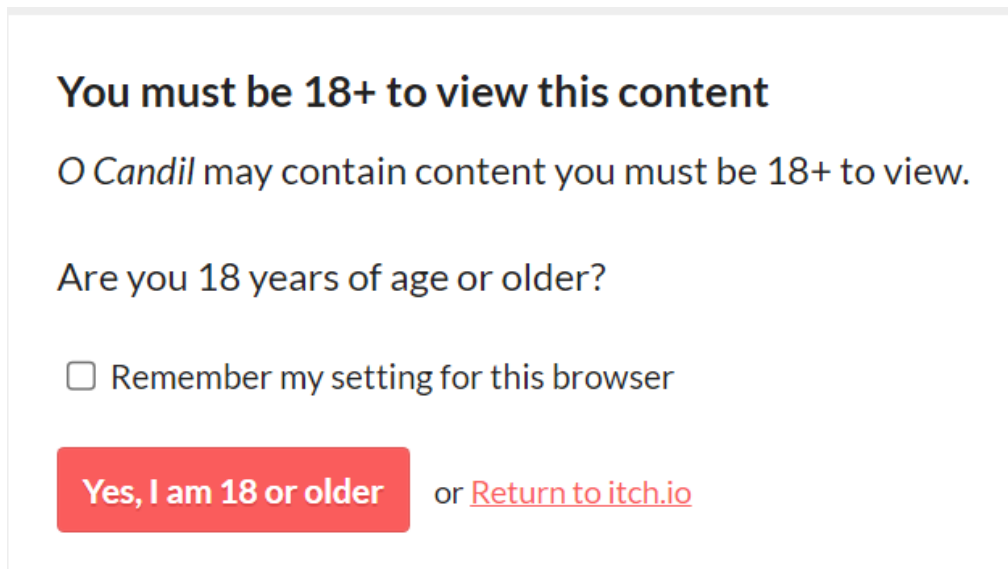
Para advertir de la edad recomendada para jugar al videojuego se utiliza la clasificación PEGI. El proyecto, por su contenido violento entra dentro del marco de clasificación PEGI 18 (ver ilustración 37).



*Ilustración 37 - Etiqueta PEGI 18*

Por ello, no se deja descargar el videojuego antes de verificar la edad en la plataforma de distribución itch.io (como se muestra en la ilustración 38), que se enlaza a continuación:

<https://ficho12.itch.io/o-candil>



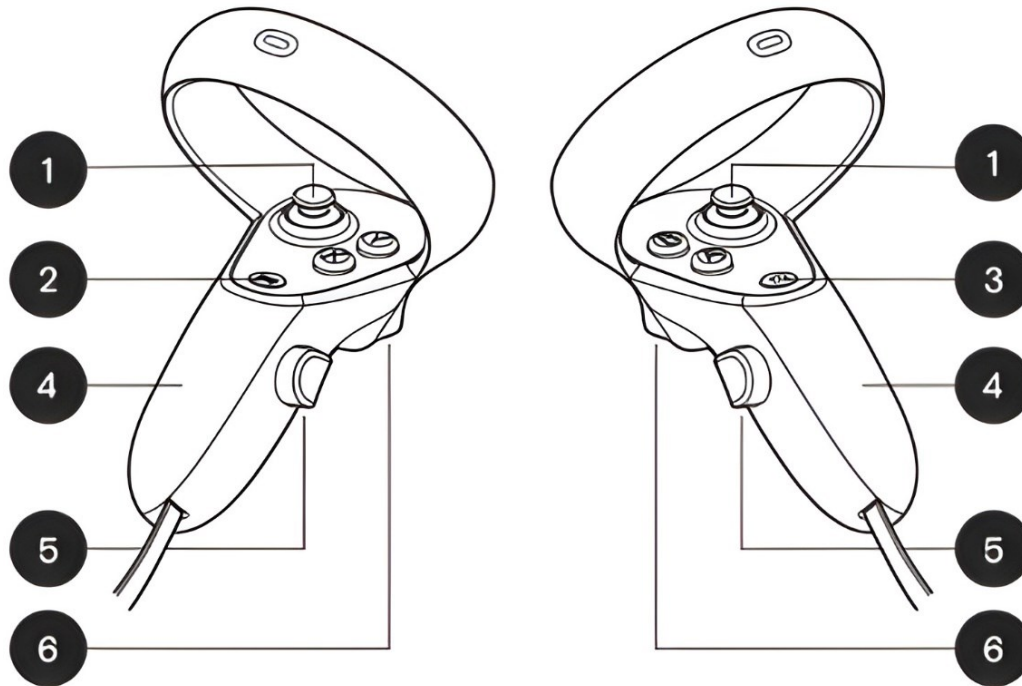
*Ilustración 38 - Advertencia de contenido para mayores de 18*

Después, se continúa con la configuración del espacio de juego, como se muestra en la ilustración 39.



*Ilustración 39 - Configuración del sistema guardián – 2 [8]*

A continuación, en la ilustración 40, se muestran los controles para los controladores Meta Touch:



- |                           |                       |
|---------------------------|-----------------------|
| 1. Moverse / Rotar cámara | 4.                    |
| 2. Pausa                  | 5.                    |
| 3. Botón Oculus           | 6. Disparar / Agarrar |
| X. Soltar arma            | A. Soltar arma        |
| Y. Seleccionar arma       | B. Seleccionar arma   |

*Ilustración 40 - Controles*

Con el punto de atención en las interacciones usuario-mundo se muestran a continuación las ilustraciones 41, 42 y 43 de la selección de arma, disparo y recarga de armas. Aunque para poner las interacciones en contexto, está recomendado ver el video de YouTube que se mencionó anteriormente.



*Ilustración 41 - Menú de selección de arma*



*Ilustración 42 - Zombie recibe un disparo en la cabeza*



*Ilustración 43 – Recarga de pistola 1911*

El manual de usuario se detalla en profundidad en el Anexo V.

## **5.6. Problemas durante el desarrollo y soluciones encontradas**

Durante el desarrollo del proyecto se encontró un problema en la realización de una de las tareas planteadas. La inclusión de modelos 3D con Nanite no funcionaba en plataformas de RV en la versión 5.1.1 de UE, la que se utilizó para la realización del proyecto. Esto suponía un retraso sobre el que era uno de los pilares fundamentales del proyecto. Al principio se pensó que el error era del desarrollador, pero una vez investigado lo suficiente se vio que este error era debido al motor gráfico, y por lo cual, no se podía solucionar. A continuación, se muestra uno de los hilos en el foro de UE sobre el error ocurrido:

<https://forums.unrealengine.com/t/5-1-lumen-nanite-vr-producing-dark-trails-when-hmd-moves/691545>

Lamentablemente la solución propuesta por los moderadores no funciona en las gafas de RV Meta Quest 2 en las que se realiza el desarrollo del proyecto.

Llegados a este punto se plantearon dos posibles alternativas: comprobar si este error estaba arreglado en la versión 5.2 de UE y portar el proyecto a esta versión o no incluir modelos con Nanite en el videojuego. Debido a la carga de trabajo que supondría portar

el proyecto a la versión 5.2 se optó por no incluir los modelos con Nanite en el proyecto, aunque el modelo de la iglesia de San Pedro Fiz do Inicio ya estaba listo para su incorporación en el nivel tal y como se muestra en la documentación de este proyecto.

Debido a esta problemática, también se ha descartado el plan inicial que buscaba la realización de 3 niveles, ya que el principal aliciente artístico en el juego era la inclusión de estos modelos 3D. En cambio, se decide centrar el proyecto en un nivel tutorial que tenga toda la funcionalidad posible, tratándose este de una demostración técnica del producto final. Como contrapartida a esta problemática, se ha optado también por mejorar el modo multijugador a cambio de la pérdida de estos niveles.

## 6. Conclusiones

Tras la finalización del proyecto VRSurvival se pueden extraer varias conclusiones con relación al desarrollo, tecnologías y también otras de ámbito personal del desarrollador. A pesar de los problemas y dificultades (técnicas y personales) que se han dado durante el desarrollo y los retrasos que han ocasionado, es posible afirmar que se han alcanzado la mayoría de los objetivos propuestos al inicio del proyecto:

- Se ha desarrollado un nivel de tutorial en el que se introducen los elementos del *gameplay* interactivamente.
- Se han diseñado mecánicas de tipo survival, adaptándolas al modo cooperativo multijugador.
- Se ha diseñado un *gameplay* multijugador entre 2 personas.
- Se ha implementado un sistema de salas multijugador empleando los servicios online de Epic, siendo los usuarios los propios anfitriones de la sala.
- Se ha implementado un sistema de almacenamiento de estadísticas utilizando la BBDD *Firestore*, en la cual se almacenan rankings de puntuaciones global y local que pueden ser visualizados por los usuarios dentro del juego.
- Se ha implementado un sistema de autenticación en el videojuego utilizando los servicios de *Firebase*. Sin embargo, otros objetivos no han podido alcanzarse como son:
- Fallo de Nanite en los modelos 3D, mencionado anteriormente en el apartado 5.7
- Existen varios errores o *bugs* en el *gameplay*, lo cual es normal en el estado que está el desarrollo para un proyecto de estas características.

Respecto a los objetivos personales del desarrollador se puede decir que se han cumplido todos satisfactoriamente a pesar del retraso durante el desarrollo.

- Se ha podido finalizar con esfuerzo el proyecto en el tiempo esperado, aun con los retrasos ocurridos.
- Se han obtenido amplios conocimientos en el campo de desarrollo de videojuegos y en concreto en la arquitectura de UE y el *framework* *VRExpansionPlugin*.
- Se ha completado un proyecto software de principio a fin con resultado en una demo técnica de un videojuego.
- Se han conseguido varias entrevistas de trabajo para puestos de ingeniero en Unreal Engine en empresas de sector.

## 7. Líneas de trabajo futuras

Debido a la naturaleza de demo técnica del proyecto, hay un gran número de posibles puntos para la continuación del desarrollo. Se proponen a continuación algunos de ellos:

- Portar el proyecto a una versión de UE que permita el uso de Nanite en VR, como se menciona en el apartado 5.7 Problemas durante el desarrollo y soluciones.
- Creación de los niveles mencionados en el Diagrama de niveles realizado.
- Añadir físicas a los diferentes elementos del videojuego y no solo a los principales.
- Incorporar una integración con plataformas de distribución de videojuegos como Steam y permitir el inicio de sesión con la cuenta de esta, evitando la creación de una cuenta específica para el videojuego. Esto permitiría seguir usando los servicios de Firebase (se podría crear una cuenta para el usuario sin la intervención de este), pero sería mucho más fácil para el usuario.

## 8. Referencias

- [1] V. Miranda, «El auge de las experiencias inmersivas | Noticias de actualidad | EL PAÍS», 29 de enero de 2021. [https://elpais.com/elpais/2021/01/25/actualidad/1611587796\\_676866.html](https://elpais.com/elpais/2021/01/25/actualidad/1611587796_676866.html) (accedido 26 de mayo de 2023).
- [2] C. Ball, K. T. Huang, y J. Francis, «Virtual reality adoption during the COVID-19 pandemic: A uses and gratifications perspective», *Telematics and Informatics*, vol. 65, p. 101728, dic. 2021, doi: 10.1016/J.TELE.2021.101728.
- [3] «Oculus Store: VR Games, Apps, & More | Oculus». [https://www.oculus.com/experiences/quest/view/777072216186618/?utm\\_source=www.google.com&utm\\_medium=oculusredirect](https://www.oculus.com/experiences/quest/view/777072216186618/?utm_source=www.google.com&utm_medium=oculusredirect) (accedido 27 de mayo de 2023).
- [4] W. Nguyen, «What are the Most Popular VR Genres? (Statistics) - VR Heaven», 16 de mayo de 2022. [https://vrheaven.io/vr-gaming-statistics/#Most\\_Popular\\_VR\\_Genres](https://vrheaven.io/vr-gaming-statistics/#Most_Popular_VR_Genres) (accedido 27 de mayo de 2023).
- [5] T. Faber, «Finding Community, and Freedom, on VRChat - The New York Times», 27 de diciembre de 2022. <https://www.nytimes.com/2022/12/27/arts/music/vrchat-virtual-reality-clubbing.html> (accedido 27 de mayo de 2023).
- [6] Beat Games, «Beat Saber». 2018.
- [7] Superhot Team, «Superhot VR». 2016.
- [8] Valve, «Half Life: Alyx». 2020.
- [9] C. Wankhede, «In 2022, I stopped using my VR headset», 27 de diciembre de 2022. <https://www.androidauthority.com/stopped-using-vr-headset-3257772/> (accedido 29 de mayo de 2023).
- [10] «Ubisoft ha cancelado el desarrollo de Splinter Cell VR», 22 de julio de 2022. <https://www.realovirtual.com/noticias/11650/ubisoft-ha-cancelado-desarrollo-splinter-cell-vr> (accedido 29 de mayo de 2023).
- [11] Capcom, «Resident Evil 7 Biohazard». 2017.
- [12] B. Lang, «Latest Figures Suggest Resident Evil 7 Has Reached 1.25M PSVR Players», 19 de mayo de 2020. <https://www.roadtovr.com/latest-figures-suggest-resident-evil-7-could-have-some-280000-psvr-players/> (accedido 29 de mayo de 2023).
- [13] «The most powerful real-time 3D creation tool - Unreal Engine». <https://www.unrealengine.com/en-US> (accedido 29 de mayo de 2023).
- [14] «Nanite Virtualized Geometry in Unreal Engine | Unreal Engine 5.1 Documentation». <https://docs.unrealengine.com/5.1/en-US/nanite-virtualized-geometry-in-unreal-engine/> (accedido 29 de mayo de 2023).

- [15] «Tecnología de supermuestreo de deep learning (DLSS) | NVIDIA». <https://www.nvidia.com/es-es/geforce/technologies/dlss/> (accedido 29 de mayo de 2023).
- [16] «ArtStation - Studios». [https://www.artstation.com/studios?software\\_ids=44](https://www.artstation.com/studios?software_ids=44) (accedido 28 de mayo de 2023).
- [17] «Fotogrametría | Ministerio de Transportes, Movilidad y Agenda Urbana». <https://www.mitma.gob.es/instituto-geografico-nacional/observacion-del-territorio/fotogrametria> (accedido 4 de julio de 2023).
- [18] «realidad | Definición | Diccionario de la lengua española | RAE - ASALE». <https://dle.rae.es/realidad> (accedido 7 de mayo de 2023).
- [19] «Oculus Rift: Step Into the Game by Oculus — Kickstarter». <https://www.kickstarter.com/projects/1523379957/oculus-rift-step-into-the-game> (accedido 30 de mayo de 2023).
- [20] «Is Valve Index Still a Good Choice Today». <https://circuitstream.com/blog/valve-index> (accedido 30 de mayo de 2023).
- [21] B. Lang, «Exclusive Preview, Part Two: Everything About HP Reverb G2», 10 de junio de 2020. <https://www.roadtovr.com/hp-reverb-g2-hands-on-preview-part-2/2/> (accedido 30 de mayo de 2023).
- [22] «Valve Index». <https://store.steampowered.com/valveindex> (accedido 30 de mayo de 2023).
- [23] «Meta Quest 2: gafas inmersivas de realidad virtual todo en uno | Meta Store | Meta Store». <https://www.meta.com/es/quest/products/quest-2/> (accedido 30 de mayo de 2023).
- [24] «Gafas de realidad virtual HP Reverb G2 - HP Store España». <https://www.hp.com/es-es/shop/offer.aspx?p=hp-reverb-g2-vr-headset> (accedido 30 de mayo de 2023).
- [25] «PICO 4 All-In-One VR Headset | PICO Spain | PICO Spain». <https://www.picoxr.com/es/products/pico4> (accedido 30 de mayo de 2023).
- [26] «PlayStation®VR2 | La nueva generación de realidad virtual para PS5 | PlayStation (España)». <https://www.playstation.com/es-es/ps-vr2/> (accedido 30 de mayo de 2023).
- [27] M. Delgado, «Meta Quest 2 se convierte en el headset VR más exitoso con 14,8 millones de unidades vendidas - Vandal», 2 de julio de 2022. <https://vandal.elespanol.com/noticia/1350754902/meta-quest-2-se-convierte-en-el-headset-vr-mas-exitoso-con-148-millones-de-unidades-vendidas/> (accedido 7 de mayo de 2023).
- [28] «State of Agile Report», 2022.

## Problemas durante el desarrollo y soluciones encontradas

- [29] «What is Scrum? | Scrum.org». <https://www.scrum.org/learning-series/what-is-scrum/what-is-scrum> (accedido 7 de mayo de 2023).
- [30] «What is Unified Modeling Language (UML)?» <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/> (accedido 7 de mayo de 2023).
- [31] «Jira | Software de seguimiento de proyectos e incidencias». <https://www.atlassian.com/es/software/jira> (accedido 6 de junio de 2023).
- [32] «Los mejores equipos del mundo trabajan mejor juntos con Atlassian». <https://www.atlassian.com/es/customers> (accedido 7 de mayo de 2023).
- [33] «diagrams.net». <https://app.diagrams.net/> (accedido 29 de mayo de 2023).
- [34] «Documentos de Google». <https://docs.google.com/document/u/0/?hl=es&tgif=d> (accedido 29 de mayo de 2023).
- [35] «draw.io – Diagrams for Confluence and Jira - draw.io». <https://drawio-app.com/> (accedido 7 de mayo de 2023).
- [36] «Excalidraw | Hand-drawn look & feel • Collaborative • Secure». <https://excalidraw.com/> (accedido 28 de mayo de 2023).
- [37] «Ideal Modeling & Diagramming Tool for Agile Team Collaboration». <https://www.visual-paradigm.com/> (accedido 14 de junio de 2023).
- [38] «Every project needs a version control system | Unreal Engine Community Wiki». <https://unrealcommunity.wiki/every-project-needs-a-version-control-system-qgpkrf2> (accedido 7 de mayo de 2023).
- [39] «Git». <https://git-scm.com/> (accedido 29 de mayo de 2023).
- [40] «git-for-windows/git-for-windows.github.io: Git for Windows' home page». <https://github.com/git-for-windows/git-for-windows.github.io> (accedido 29 de mayo de 2023).
- [41] «Servicios de informática en la nube | Microsoft Azure». <https://azure.microsoft.com/es-es> (accedido 29 de mayo de 2023).
- [42] «GitHub». <https://github.com/> (accedido 29 de mayo de 2023).
- [43] «GitKraken Legendary Git Tools | GitKraken». <https://www.gitkraken.com/> (accedido 29 de mayo de 2023).
- [44] D. Takahashi, «Epic Games: Unreal Engine 5 will bring a generational change to graphics | VentureBeat», 13 de mayo de 2020. <https://venturebeat.com/business/how-epic-games-is-tailoring-unreal-engine-5-to-make-next-gen-graphics-shine/> (accedido 9 de mayo de 2023).
- [45] Jay, «(368) Unreal Engine 5.1 VR Warehouse - YouTube», 30 de abril de 2023. <https://www.youtube.com/watch?v=G7xzlBv8r64> (accedido 9 de mayo de 2023).

- [46] «Introduction to Blueprints Visual Scripting in Unreal Engine | Unreal Engine 5.1 Documentation». <https://docs.unrealengine.com/5.1/en-US/introduction-to-blueprints-visual-scripting-in-unreal-engine/> (accedido 29 de mayo de 2023).
- [47] J. Statzer, «mordentral/VRExpansionPlugin: A UE4 VR framework». Accedido: 29 de mayo de 2023. [En línea]. Disponible en: <https://github.com/mordentral/VRExpansionPlugin>
- [48] «IDE de Visual Studio 2022: herramienta de programación para desarrolladores de software». <https://visualstudio.microsoft.com/es/vs/> (accedido 29 de mayo de 2023).
- [49] «Firebase». <https://firebase.google.com/?hl=es> (accedido 2 de junio de 2023).
- [50] «blender.org - Home of the Blender project - Free and Open 3D Creation Software». <https://www.blender.org/> (accedido 29 de mayo de 2023).
- [51] «RealityCapture - Photogrammetry Software». <https://www.capturingreality.com/realitycapture> (accedido 29 de mayo de 2023).
- [52] C. Kerr, «Epic Games acquires photogrammetry software developer Capturing Reality», 9 de marzo de 2021. <https://www.gamedeveloper.com/business/epic-games-acquires-photogrammetry-software-developer-capturing-reality> (accedido 9 de mayo de 2023).
- [53] W. Faucher, «(368) RealityCapture to UE5 - Workflow Tutorial - YouTube», 18 de diciembre de 2021. <https://www.youtube.com/watch?v=WrcOhes1Zgg> (accedido 9 de mayo de 2023).
- [54] «Mixamo». <https://www.mixamo.com/#/> (accedido 29 de mayo de 2023).
- [55] «Mixamo Converter». <https://terribilisstudio.fr/?section=MC> (accedido 29 de mayo de 2023).
- [56] «Adobe Lightroom | Software de edición y organización de fotografía». <https://www.adobe.com/es/products/photoshop-lightroom.html> (accedido 29 de mayo de 2023).
- [57] «Adobe Photoshop Oficial | Programa de edición de fotos y diseño». <https://www.adobe.com/es/products/photoshop.html> (accedido 29 de mayo de 2023).
- [58] A. Durán y T. Beatriz Bernárdez Jiménez, «Metodología para la Elicitación de Requisitos de Sistemas Software Versión 2.1», *Informe Técnico LSI-2000-10*, Universidad de Sevilla, oct. 2000, Accedido: 31 de mayo de 2023. [En línea]. Disponible en: <http://www.lsi.us.es/docs/informes/lsi-2000-10.pdf>
- [59] «About the Unified Modeling Language Specification Version 2.0». <https://www.omg.org/spec/UML/2.0/> (accedido 14 de junio de 2023).

- [60] B. Todd, «A Modern History of Horror Games», 2007. [https://web.archive.org/web/20070518141649/http://uk.gamespot.com/gamespot/features/pc/history\\_horror\\_pt1/p2\\_01.html](https://web.archive.org/web/20070518141649/http://uk.gamespot.com/gamespot/features/pc/history_horror_pt1/p2_01.html) (accedido 2 de mayo de 2023).
- [61] Naughty Dog, «The Last of Us Part II». 2020.
- [62] Dumbuster Studios, «Dead Island 2 ». 2023.
- [63] Striking Distance Studios, «The Callisto Protocol». 2022.
- [64] Capcom, «Resident Evil 2 Remake». 2019.
- [65] Capcom, K2-Inc, y Redworks, «Resident Evil 3 Remake». 2020.
- [66] Capcom, «Resident Evil 4 Remake». 2023.
- [67] Motive Studio y Ea Redwoos Shores, «Dead Space Remake». 2023.
- [68] Bloober Team, «Silent Hill 2 Remake». 2023.
- [69] 4A Games, «Metro Exodus». 2021.
- [70] Pieces Interactive, «Alone in the Dark Remake». 2023.
- [71] praydog, «RE7 PCVR test #2 (motion controls, roomscale, full body IK) - YouTube», 5 de marzo de 2022. <https://www.youtube.com/watch?v=9ddvqUcgIkQ> (accedido 6 de mayo de 2023).
- [72] Capcom, «Resident Evil Village». 2021.
- [73] J. Cano, «Análisis Resident Evil 7 - PS4, Xbox Series X/S, PS5, Xbox One, PC», 2020. <https://vandal.elespanol.com/analisis/ps4/resident-evil-7/39839#p-73> (accedido 3 de mayo de 2023).
- [74] C. Leiva, «Análisis Resident Evil 8 Village, la aldea de las sombras», 2021. <https://vandal.elespanol.com/analisis/ps5/resident-evil-8-village/86453#p-89> (accedido 3 de mayo de 2023).
- [75] B. Lang, «Half-Life: Alyx Has Become Steam's Best & Most Rated VR Game», *'Half-Life: Alyx' Has Become Steam's Best & Most Rated VR Game Less Than a Year After Launch*, 15 de diciembre de 2020. <https://www.roadtovr.com/half-life-alyx-steam-best-most-rated-game/> (accedido 28 de abril de 2023).
- [76] londoncitynights, «Half-Life: Alyx (PC, 2020) – Pixel Hunted», 25 de mayo de 2022. <https://pixelhunted.wordpress.com/2022/05/25/half-life-alyx-pc-2020/> (accedido 6 de mayo de 2023).
- [77] Dribbleondo, «(357) Hitman 2 Runs Really Well On Linux Via Proton - YouTube», 11 de febrero de 2019. [https://www.youtube.com/watch?v=GiSVGChZ\\_xw](https://www.youtube.com/watch?v=GiSVGChZ_xw) (accedido 6 de mayo de 2023).