



Universidad de Oviedo

TRABAJO DE FIN DE MÁSTER

MÁSTER UNIVERSITARIO EN INGENIERÍA
INFORMÁTICA POR LA UNIVERSIDAD DE OVIEDO

ENTORNO DE REALIDAD VIRTUAL PARA LA EDUCACIÓN Y DIVULGACIÓN MÚSICAL

AUTOR: D. Javier Álvarez de la Fuente
TUTOR: Dr. Víctor Manuel Álvarez García

FECHA: Julio de 2022

MEMORIA

Tabla de contenido

1	Introducción	9
1.1	Justificación	9
1.2	Qué es WebXR	12
1.3	Situación inicial	12
1.4	Estudio de alternativas	18
1.5	Alternativa escogida	26
1.6	Hardware	28
2	Planificación del proyecto	29
2.1	Planificación inicial	29
2.2	Planificación final	32
3	Fase de análisis	33
3.1	Requisitos	33
3.2	Especificación de casos de uso	34
4	Fase de diseño	50
4.1	Diagrama de clases	50
4.2	Arquitectura	53
5	Estudio inicial	53
5.1	Características y objetivo del escenario	57
5.2	Tecnologías y desarrollo	59
5.3	Acceso y repositorio	62
6	Diseño del escenario	62
6.1	Boceto del escenario	64
6.2	Creación del escenario	65
6.3	Salas secundarias	69
7	Implementación de la aplicación	76

7.1	Exportado del escenario a Unity.....	76
7.2	Problemas de rendimiento iniciales	82
7.3	Scripts.....	84
7.4	Contenido interactivo	98
7.5	Dificultades encontradas	114
7.6	Objetos modelados	117
8	Despliegue y arquitectura final	118
8.1	Arquitectura.....	119
8.2	Acceso al despliegue	122
8.3	Lanzamiento del despliegue.....	124
9	Pruebas.....	125
9.1	Pruebas de estado y menú	127
9.2	Pruebas de interacción	136
9.3	Pruebas de vídeo	157
9.4	Arreglo de errores	163
10	Resultados y trabajo futuro	164
11	Conclusiones	167
12	Referencias.....	171
13	Documentación adicional	173

Índice de ilustraciones

Ilustración 1.1. Electronauts.	13
Ilustración 1.2. Music Inside: A VR Rhythm Game	14
Ilustración 1.3. Jam Studio.	15
Ilustración 1.4. Billie Eilish actuando en Horizon Venues (beta).....	18
Ilustración 2.5. Diagrama de Gantt (Planificación inicial)	31
Ilustración 3.6. Casos de uso (General).....	35
Ilustración 3.7. Casos de uso (Sala principal).....	35
Ilustración 3.8. Casos de uso (Sala secundaria).....	36
Ilustración 4.9. Diagrama de clases (General).....	51
Ilustración 4.10. Diagrama de clases (Jugador).....	51
Ilustración 4.11. Diagrama de clases (Interactivos).....	52
Ilustración 5.12. Escenario del estudio inicial.....	54
Ilustración 5.13. Interacción inmersiva en el estudio inicial.	56
Ilustración 5.14. Interacción no inmersiva en el estudio inicial	57
Ilustración 5.15. Logotipo de WebXR.....	59
Ilustración 5.16. Acceso a la aplicación.	60
Ilustración 6.17. Plano del escenario.	64
Ilustración 6.18. Prototipado del escenario principal en Unity.	65
Ilustración 6.19. Malla básica del escenario principal en Blender.	66
Ilustración 6.20. Escenario principal en Blender.	69
Ilustración 6.21. Sala secundaria de instrumentos en Blender.	71
Ilustración 6.22. Sala secundaria asturiana en Blender.	73
Ilustración 6.23. Sala secundaria de teoría y juego en Blender.	76
Ilustración 7.24. Primera versión del escenario principal en Unity.	79
Ilustración 7.25. Salas secundarias en Unity.	82

Ilustración 7.26. Escena del menú modo no XR	98
Ilustración 7.27. Escena del menú en modo XR	99
Ilustración 7.28. Sala de entrada en XR lanzando teletransporte.....	100
Ilustración 7.29. Interacciones en XR en el vestíbulo.....	102
Ilustración 7.30. Interacción XR en el patio.....	103
Ilustración 7.31. Proceso de instanciación de VideoTapes	104
Ilustración 7.32. Instanciación de cintas en XR.....	106
Ilustración 7.33. Proceso para reproducir vídeo.....	106
Ilustración 7.34. Proceso para reproducir videos en XR	108
Ilustración 7.35. Interacción con instrumento en XR.....	109
Ilustración 7.36. Interacción XR en la sala asturiana	110
Ilustración 7.37. Visualizando teoría en modo XR.	111
Ilustración 7.38. Colocando una cinta y tocando en la sala de juego (XR).....	114
Ilustración 8.39. Arquitectura del proyecto.....	120
Ilustración 8.40. Acceso al proyecto.	123
Ilustración 8.41. Acceso a los servicios	124

Índice de tablas

Tabla 2.1. Planificación a priori.....	30
Tabla 2.2. Planificación final.	32
Tabla 2.3. Hitos en la planificación final.....	33
Tabla 3.4. Requisitos funcionales.	34
Tabla 3.5. Requisitos no funcionales.	34
Tabla 6.6. Descripción de salas del escenario.....	63
Tabla 7.7. Scripts del usuario	85
Tabla 7.8. Scripts para la interacción.....	87
Tabla 7.9. Otros scripts.	94
Tabla 7.10. Objetos modelados.	117
Tabla 10.11. Objetivos cumplidos.....	165

1 Introducción

1.1 Justificación

El presente proyecto se ha realizado como Trabajo de fin de Máster del Máster universitario en Ingeniería Informática por la Universidad de Oviedo, impartido en la Escuela Politécnica de Ingeniería de Gijón (Universidad de Oviedo). Ha sido llevado a cabo gracias a la colaboración con la Fundación CTIC, lugar donde se han realizado las prácticas en empresa y donde se ha facilitado las herramientas y el hardware específico para llevar a cabo el trabajo.

Este proyecto nace de intentar juntar los tres campos que resultan interesantes al alumno: El diseño y desarrollo **3D**, la **música** y la **educación**. Se busca **experimentar dentro del campo de las tecnologías inmersivas y de desarrollo 3D**, poniendo especial enfoque en el trabajo con motores gráficos e investigando sobre **estándares WebXR**, con el objetivo de **diseñar y crear un escenario 3D inmersivo accesible a través de web** que permita la divulgación, consulta y visualización de aspectos relacionados con la cultura musical.

Garantizando el acceso al escenario a través de web se busca **promover la cultura musical** (incluyendo la música tradicional asturiana) de una forma moderna tecnológicamente hablando, con el objetivo que los usuarios de todas las edades y niveles se vean interesados por ella.

Por último, se tiene el objetivo de aprender sobre la realización de proyectos software abarcados dentro del ámbito 3D, poniendo en marcha **metodologías ágiles** y técnicas de dirección de proyectos.

1.1.1 *Objetivos y alcance.*

El proyecto planteado pretende **desarrollar un prototipo de aplicación web que permita a los alumnos de centros educativos no universitarios el acceso a una aplicación 3D** que divulgue conocimientos básicos culturales y les ayude a orientarse en una futura carrera en el campo del arte. Aunque se espera diseñar la aplicación para divulgar conocimientos culturales genéricos, en el ámbito de este proyecto se centrará en la divulgación musical.

Es más que probable que los alumnos y los centros en los que se encuentre matriculados no dispongan del Hardware necesario (gafas de realidad virtual) para disfrutar de una experiencia inmersiva, es por ello por lo que también se permitirá el acceso sin él haciendo que vean la aplicación como un juego en versión web estándar en el que adquirirán los mismos conocimientos.

Por otro lado, también se espera que los usuarios visualicen contactos de escuelas de música o conservatorios, con el fin de que los alumnos puedan obtener más información si así lo ven conveniente.

Por tanto y, en resumen, se definen los siguientes objetivos:

- Llevar a cabo el **diseño y creación de un entorno 3D** para la divulgación de diversos aspectos culturales, que sirva como entorno inmersivo para la visualización de contenido interactivo orientado a la enseñanza en diversos ámbitos.
 - Para el ejemplo de este proyecto se presenta un entorno 3D para la divulgación de conocimientos musicales y para la visualización de contenido interactivo orientado a este campo.
 - La divulgación de contenido también conllevará la reproducción de vídeos que en todo momento debe ser dinámica y

personalizada, permitiendo el visionado de un listado de videos sin tener que realizar cambios en el código o en el proyecto.

- Permitir un **acceso público a la aplicación** y, si se detecta la conexión de hardware inmersivo permitir el acceso desde el mismo gracias al estándar **WebXR**.
- Permitir la interacción con diferentes ámbitos, por ejemplo:
 - Escalas y teoría musical básica.
 - Cultura asturiana.
 - Visualización de vídeos almacenados en nube.
 - Instrumentos, sus sonidos o papeles.
- Divulgar la música y cultura asturiana en todos sus ámbitos culturales: instrumentos, bandas de gaitas, danza...
- Permitir el acceso a contactos de diferentes escuelas de música o conservatorios de la región para facilitar la obtención de más información por parte de los usuarios.

Se debe tener en cuenta que **el diseño de un escenario o modelos 3D complejos y personalizados no es el objetivo del presente trabajo, por lo que el resultado final será un prototipado del proyecto**. Esto quiere decir que puede mostrarse con un aspecto “inacabado” pero con todas las funcionalidades y comportamientos necesarios. Es decir, a pesar de que podrán modelarse o incluirse algunos modelos del escenario o de los objetos interactivos, **el objetivo principal es cumplir los objetivos de *scripting* que permitan un acceso inmersivo a la aplicación y una interacción fluida**.

Es por esto por lo que **se priorizará el uso de formas geométricas básicas** (cubos, cilindros, etc.), **dejando el modelado definitivo del escenario y la finalización del proyecto** (añadido de todo el contenido) **para una fase post-entrega del trabajo**.

Esto se hace para poder cumplir los plazos de entrega que ofrece el presente curso, pero también para mantener el enfoque en la parte del proyecto que más se adapta a la titulación cursada a la que pertenece el trabajo.

1.1.2 Actores en el proyecto

Como bien se ha explicado anteriormente, el presente proyecto busca la divulgación cultural en el ámbito de los centros educativos. No obstante, podría ser accesible por cualquier usuario teniendo en cuenta que sería una aplicación de acceso público.

Por tanto, los actores principales del proyecto no serán otros que los usuarios, no se prevé ni se necesita varios tipos de roles o de permisos de usuario.

1.2 Qué es WebXR

WebXR [1] es un grupo de estándares de **W3C** que se utilizan en conjunto para representar escenas 3D en hardware específico para representar mundos virtuales (realidad virtual) o para añadir imágenes al mundo real (realidad aumentada).

La ventaja que ofrece el uso de estos estándares frente a otras tecnologías es la posibilidad de acceder a una aplicación web desde el navegador y que esta sea capaz de detectar el hardware necesario y representar el contenido en este sin necesidad de instalar ninguna aplicación de terceros tanto en el equipo como en las gafas.

1.3 Situación inicial

Actualmente existen muchos periódicos, blogs, podcast e incluso jornadas de divulgación de arte y de cultura asturiana. No obstante, no existe en el mercado una aplicación web que permita la adquisición de conocimientos básicos de forma tecnológica e inmersiva y totalmente gratuita y que, además, permita la promoción de

entidades musicales para orientar a los usuarios en la búsqueda de su futuro *hobby* o carrera profesional.

Lo que si existe en el mercado son diversas aplicaciones de Realidad Virtual orientadas a la práctica, a la creación de música o al disfrute de esta de una forma diferente a la tradicional. Estas aplicaciones permiten al usuario diseñar de una forma rápida, divertida e inmersiva sus creaciones musicales, practicar técnicas de DJ, visualizar escenarios interactivos de luces mientras se escuchan piezas musicales como si de un espectáculo de luces se tratase e incluso tocar instrumentos como puede ser un piano.

Pero, como se comentaba al comienzo del presente apartado, no se encuentra disponible ninguna aplicación gratuita que permita el aprendizaje y divulgación básica ni que incluya aspectos que representen a nuestra región.

Aun así, se detallarán unas pocas aplicaciones existentes para estudiar la situación actual del mercado:

1.3.1 *Electronauts*



Ilustración 1.1. *Electronauts*.

Electronauts [2] es un videojuego inmersivo en realidad virtual que permite la

composición y la creación de *remixes* de música electrónica. Se permite al usuario pinchar música electrónica como si en un escenario se encontrase, permitiéndole crear sus propios *beats* y *loops*, no importa el nivel del jugador pues se busca que la música sea creada de una forma sencilla, intuitiva y divertida.

Se pueden mezclar muchas pistas de diferentes famosos artistas de diversos estilos urbanos. El juego está desarrollado mediante un motor propio: *Music Reality Engine* que transforma los sonidos y la música en objetos con los que el jugador puede interactuar.

Tiene soporte para hardware HTC Vive y Oculus Rift, permite colaborar con otro amigo o usuario y se puede encontrar en *Steam* a un precio de aproximadamente 17€ (o más si incluimos los diferentes packs de expansión). Por último, no puede usarse sin realidad virtual.

1.3.2 *Music Inside: A VR Rhythm Game*



Ilustración 1.2. *Music Inside: A VR Rhythm Game*

Como su propio nombre indica, *Music Inside: A VR Rhythm Game* [3] es un videojuego Indie en realidad virtual orientado, una vez más, a la creación de ritmos y músicas de una forma divertida y sencilla. Se permite seleccionar audio de una gran

librería (pudiendo importar nuestra propia música), una vez seleccionado un algoritmo se encarga de transformar la música en secuencia de notas que el usuario debe pulsar al ritmo de los compases (hay 5 niveles de dificultad y 4 modos de interacción).

Se permite el acceso desde Hardware *HTC Vive*, *Oculus Rift* y *Valve Index* y una vez más puede jugarse en modalidad cooperativa con otra persona. Puede encontrarse en *Steam* a un precio de aproximadamente 10€ (incluyendo, a parte, diferentes packs de expansión). Por último, tampoco puede usarse sin realidad virtual.

1.3.3 *Jam Studio VR*



Ilustración 1.3. *Jam Studio*.

Jam Studio [4] es quizá la alternativa que más se acerca al propósito del presente proyecto. Se trata de una serie de aplicaciones inmersivas en realidad virtual que permiten el manejo de diferentes instrumentos virtuales de diversos géneros: Rock, Jazz, Pop, música Techno, urbana...

La propia descripción de la aplicación indica que no se trata de un juego, si no de una experiencia que te permite practicar música sin ninguna regla técnica ni teórica. Pueden usarse guitarras, teclados, baterías e incluso instrumentos de vientos como trompetas o saxofones. Se permite el acceso a decenas de canciones interactivas para

practicar sobre ellas.

A parte de la aplicación *Jam Studio VR* antes detallada, también podemos encontrar la variante *Jam Studio VR Education & Health Care Edition* que no solo permite practicar instrumentos, sino que también permite el acceso a recursos para profesores, terapeutas y estudiantes:

- Acceso a más canciones interactivas, cientos más que en la versión estándar.
- Acceso a planes y guías para practicar conocimientos básicos teóricos de música: escalas, sonidos, armonía, técnica, etc.
- Acceso a guías de terapia para demostrar como la música y el manejo de instrumentos puede ayudar a desarrollar y alcanzar objetivos cognitivos, sensoriales, de comunicación y sociales.

Tiene soporte para *HTC Vive*, *Oculus Rift* y *Windows MR* y puede encontrarse tanto en la tienda de *Meta* (Antiguo *Facebook*) como en *Steam* con un precio de aproximadamente 17€ para la versión estándar y de 170€ para la versión educativa (packs de expansión no incluidos).

Cómo se indicaba al comienzo de este apartado, se trata de la alternativa que más se acerca al objetivo del proyecto pues se pueden adquirir conocimientos básicos de música con ella y aprender como suenan los diferentes instrumentos, sin embargo, presenta varios inconvenientes:

- Tiene un precio elevado si se busca obtener la versión educativa: 170€ es un precio que puede salirse de los límites de muchos centros educativos.
- No permite el acceso ni el uso sin hardware de realidad virtual, muchos centros educativos y usuarios no disponen de hardware de este tipo.

- No indica datos de contactos de centros para continuar la formación tales como escuelas de música o conservatorios.
- No divulga cultura específica o de la región del presente proyecto: Cultura Asturiana.

1.3.4 Experiencias inmersivas

Hay diversas aplicaciones o videos de realidad virtual orientados al disfrute y escucha de música y conciertos. Se permite al usuario el acceso desde hardware de realidad virtual o sin él (si es el caso de un video alojado en plataformas como *Youtube*).

Hay cientos de videos de ejemplo de estas experiencias, uno por ejemplo puede ser *Fantasynth: Chez Nous* [5], sencillamente solo se debe visualizar, escuchar y disfrutar... no hay ningún tipo de interacción salvo el movimiento de cámara o el control del vídeo.

También hay casos especiales, aplicaciones que permiten la escucha de conciertos en directo real: En vez de asistir al concierto real puedes verlo y escucharlo desde casa como si estuvieses en él. En la propia página de *Oculus* [6] pueden encontrarse varios eventos: Por ejemplo, nos encontramos con el caso de *Horizon Venues* [7], la aplicación sucesora de *Oculus Venues* que aún se encuentra en *beta* que se centra en una experiencia enfocada a la interacción social inmersiva que permite la asistencia a eventos como conciertos, monólogos o espectáculos (aunque teniendo en cuenta que estos espectáculos se tratan de imágenes 2D proyectadas en un escenario 3D sin ningún tipo de interacción más allá que la social con otros usuarios).



Ilustración 1.4. Billie Eilish actuando en Horizon Venues (beta).

Realmente puede visualizarse cualquier contenido en formato de realidad virtual desde varias plataformas si se usan las cámaras adecuadas para grabarlo, por lo que podría ser una alternativa que cumpla todos los objetivos de este proyecto: Videos para la impartición de contenidos, para la divulgación de cultura musical o asturiana, para mostrar contactos y orientar a los usuarios, etc.

No obstante, estaríamos perdiendo toda la interacción que puede permitir usar motores gráficos y toda la libertad del usuario (le limitaríamos a seleccionar videos y a visualizarlos, no a moverse por el escenario y a visualizar lo que él vea oportuno). Es una alternativa interesante para realizar otro tipo de proyectos, pero en este caso perderíamos todo el enfoque de desarrollo tecnológico e investigación del presente proyecto.

1.4 Estudio de alternativas

Como bien se ha visto en el apartado anterior, existen varias aplicaciones de realidad virtual orientadas al campo de la música, pero ninguna cumple con la totalidad de los objetivos de este proyecto. Tan solo la alternativa de utilizar videos en realidad virtual podría cumplirlos, pero como ya se comentó la perdida de interacción y de

desarrollo hace que sea descartada por completo.

Por ello, y ahora conociendo que debemos desarrollar una aplicación inmersiva a medida, vamos a detallar las alternativas de desarrollo que pueden utilizarse para llevar a cabo este trabajo dividiéndolas en varios apartados: Motor gráfico, modelado 3D, base de datos e IDE (entorno de desarrollo).

1.4.1 Motor gráfico

El motor gráfico es el cerebro de la aplicación, es lo que permite abstraernos de muchos aspectos que harían que el desarrollo desde cero pudiese llevar varios años. El motor gráfico facilita todas las tareas de creación de Scripts (determina el lenguaje de programación a usar) y las herramientas necesarias para la creación de la escena e importación de paquetes. También nos abstrae del cálculo de físicas y de iluminación del entorno.

1.4.1.1 Unity

Unity [8] es uno de los motores gráficos para el desarrollo de videojuegos tanto 2D como 3D más populares en la comunidad de desarrollo. Es intuitivo de usar (para realizar aplicaciones sencillas, gracias a funcionar mediante objetos con componentes) y trabaja con el lenguaje C# para el desarrollo Scripts.

Realizar un prototipado del proyecto es relativamente rápido y es altamente configurable. Además, permite integrar la realidad virtual y aumentada de forma rápida y tiene a su disposición un alto número de librerías gracias a su amplia comunidad, a su *Package Manager* (donde puedes importar paquetes y librerías rápidamente) y al uso que hace esta de la *Asset Store* (lugar donde puedes acceder rápidamente y obtener recursos para tu proyecto).

Es por ello por lo que el uso de Unity hace mucho tiempo que ha dejado de estar

encausado en el desarrollo de los videojuegos y ha pasado a ser usado para el desarrollo de aplicaciones inmersivas, de experiencias 3D, aplicaciones de realidad aumentada, arquitectura e ingeniería, sector del transporte e incluso para la creación de películas de animación.

También permite exportar los proyectos para su uso en el navegador Web, utilizando WebGL.

Es gratuito siempre que no se llegue a facturar más de 75.000 euros anuales, por lo que para el ámbito de este proyecto podemos decir que es totalmente gratuito.

1.4.1.2 Unreal Engine

Otro de los grandes motores gráficos para videojuegos del mercado [9]. Funciona con C++ por debajo para el funcionamiento de Scripts, pero por lo general su punto fuerte es que no es necesario tener conocimientos de programación para desarrollar tu videojuego gracias a sus *Blueprints*: una forma de *scripting* totalmente visual integrada desde la versión 4 del motor que permite programar todo únicamente arrastrando y soltando nodos (efectos visuales, IA, interfaces gráficas, movimientos, etc).

Por ello, gracias a las *Blueprints*, este motor es indicado para su uso por parte de diseñadores en vez de programadores y desarrolladores (aunque también permite programar manualmente los scripts, con un lenguaje más “áspero” como es C++ respecto al C# de Unity).

Por otro lado, los gráficos son más espectaculares y esto hace que muchos videojuegos AAA hagan uso de este motor. Las opciones de post-procesado vienen por defecto, lo que hace que sean aún más realistas.

También tiene soporte para realidad extendida (realidad virtual y aumentada) de forma nativa, lo que permite usarlo de forma rápida e intuitiva. También permite realizar videojuegos para el navegador usando HTML5.

La salida del motor en su versión 5 es inminente, lo que trae consigo muchas funcionalidades y mejoras gráficas, pero aún se encuentra en fase *Early Access* y no se planea su lanzamiento hasta 2022.

Su precio es gratuito siempre que se facture menos de 3000 dólares por trimestre, por lo que una vez más podemos asumir que es gratuito.

1.4.1.3 Godot

Se trata de un motor para la creación de videojuegos tanto en 2D como en 3D, multiplataforma, libre y de código abierto (liberado en *GitHub* bajo la licencia *MIT*).

Hace uso de OpenGL en el proceso de renderización para las plataformas soportadas, aunque se trata de un motor que se encuentra en constante mejora teniendo previsto un uso futuro de Vulkan.

No tiene nada que envidiar a otras alternativas más famosas como las ya estudiadas Unity y Unreal Engine, disponiendo a su vez de sus propios editores de animación, sombreadores, etc.

Como punto destacable, sobre otras alternativas, el motor tiene soporte para scripting tanto usando C++, como C#, Python o GDScript (un lenguaje de programación, similar a Python, creado especialmente para este motor).

Su punto fuerte es el 2D, aunque también dispone de capacidad para la creación de aplicaciones XR (Realidad virtual y aumentada) de forma nativa.

1.4.1.4 Three JS

ThreeJS [10] es una librería de JavaScript que permite crear y mostrar gráficos animados en 3D desde el navegador, puede ser utilizada en conjunto con HTML5, SVG y WebGL (este último es el más popular).

Se trata de la alternativa con una curva de aprendizaje más pronunciada y que

requiere de mayor tiempo de desarrollo: el no utilizar un motor no nos abstraerá de todo el cálculo de físicas por lo que serán necesarios conocimientos avanzados de 3D de los que no se dispone actualmente.

Como ventaja podemos decir que es una alternativa totalmente *open source* y gratuita, con soporte nativo para navegador y que además es extremadamente liviana.

1.4.2 IDE

Aunque se utilice un motor gráfico, lo que facilitaría mucho el trabajo y el desarrollo, la edición de código en los scripts del proyecto no podrá realizarse desde él. Es por ello por lo que se estudiarán algunas alternativas para el entorno de desarrollo, lo que comúnmente se denomina IDE (*Integration Development Environment*).

1.4.2.1 Visual Studio Code

Propiedad de *Microsoft* [11], es un editor de código gratuito y de código abierto, es multiplataforma y es muy personalizable gracias a la posibilidad de integrar e instalar múltiples paquetes de la comunidad.

Permite el desarrollo y la depuración en todo tipo de lenguajes de programación (C, C++, C#, Python, JavaScript...) sin importar el tipo. También dispone de terminal integrada.

1.4.2.2 Visual Studio

También propiedad de *Microsoft* [12] es la versión principal de su IDE para el desarrollo en *.Net*. En sí no es gratuito, pues tiene un precio mínimo de 15 dólares mensuales (que dependerán del tamaño de la empresa), pero dispone de licencias gratuitas para estudiantes.

Es un IDE para Windows rico en contenido que permite el desarrollo y depuración principalmente en C, C# y C++.

1.4.2.3 Rider JetBrains

El IDE de JetBrains [13] para *.NET* es multiplataforma y está basado en la plataforma *IntelliJ* y *ReSharper*. Su precio oscila entre 14 y 35 dólares mensuales dependiendo del tipo de empresa.

Como se comentaba, es multiplataforma, por lo que es un punto a favor respecto a Visual Studio (*Code* también lo es).

1.4.3 *Modelado 3D*

Aunque no es el objetivo principal del proyecto, también será necesario realizar algunos simples modelos 3D para crear una escena más adecuada al objetivo del presente proyecto.

Las alternativas de modelado 3D incluyen software orientado a crear y diseñar modelos 3D personalizados.

1.4.3.1 Blender

Es un software multiplataforma [14] dedicado especialmente al modelado, iluminación, renderizado y creación de gráficos 3D (incluso a la creación de animaciones para estos).

Es totalmente gratuito y de código abierto, lo que hace que sea ampliamente utilizado por la comunidad y este altamente documentado. Una vez comprendido su funcionamiento y su manejo, se pueden conseguir resultados profesionales en muy poco tiempo gracias a todos los atajos de teclado de los que dispone.

1.4.3.2 SketchUp

SketchUp [15] (anteriormente Google Sketchup) es un programa de modelaje 3D propiedad de Google y tiene un precio de 109 euros anuales (para la versión más básica).

Permite la creación de objetos tridimensionales de forma realmente sencilla, pero más orientada a diseños arquitectónicos y dibujo técnico, lo que lo hace una alternativa idónea para arquitectos y constructores.

Tiene soporte para visor de realidad aumentada.

1.4.3.3 Autodesk Maya

Maya [16] es una alternativa directa para Blender, no es open source y tiene un precio muy alto para el proyecto: Una licencia cuesta 2000€ anuales (aunque dispone de licencia gratuita para estudiantes).

Era el líder del sector hasta la salida de la versión 2.8 de Blender (momento en el que se convirtió en el principal competidor). No obstante, esta altamente extendido y permite realizar resultados muy profesionales y avanzados en diseño 3D.

1.4.4 Almacenamiento de vídeos

El visionado de vídeos se convierte en un requisito indispensable del proyecto. En la medida de lo posible se permitirá el visionado de vídeos almacenados en internet, pero el objetivo principal será visualizar de forma dinámica una serie de vídeos que el cliente estime oportuno.

Permitir un sencillo almacenamiento de vídeos para el usuario y un acceso dinámico a estos precisa de realizar un breve estudio sobre tecnologías de almacenamiento existente:

1.4.4.1 Cloud Object-based Storage

El almacenamiento de objetos (data storage) se basa en almacenar cualquier tipo de dato u objeto no estructurado independiente de su tamaño o relación con otros datos, lo que lo hace especialmente útil para albergar copias de seguridad, multimedia, registros, etc. Existen múltiples alternativas ofrecidas por grandes proveedores en nube,

como puede ser Amazon con AWS Simple Storage Service [17], Google con su Cloud Storage [18] y Microsoft Azure con Azure Blob Storage [19].

En el caso de la alternativa de Amazon nos encontramos con el uso de S3: Su servicio de almacenamiento que ofrece escalabilidad, disponibilidad, seguridad y un rendimiento que sin duda es líder en el sector. La alternativa de *Google* es altamente compatible con dicho sistema (lo que facilita migraciones) y la alternativa de *Azure* no lo es (lo que dificulta en gran medida las migraciones).

El uso de soluciones ofrecidas por proveedores externos facilita la implementación, la redundancia y el mantenimiento del almacenamiento, no obstante, ello conlleva costes recurrentes que en muchos casos llegan a ser elevados pues no se resumen únicamente en la capacidad contratada, sino también en el número de operaciones de lectura y escritura (que en el caso que ocupa el presente proyecto podría no llegar a ser problemático, pero ello conllevaría costes extras no contemplados en el presupuesto).

1.4.4.2 MinIO

MinIO [20] es una tecnología de almacenamiento de objetos de datos no estructurados haciendo que estos sean accesibles a través de API HTTP, lo que la hace especialmente adecuada para el almacenamiento escalable en soluciones donde la arquitectura se encuentre en servidores dedicados, nube o borde (*edge*) y se busque prescindir de opciones de grandes proveedores que conllevan elevados costes recurrentes.

Se trata de un software de código abierto que ha sido desarrollado para pertenecer al conjunto de almacenamiento en nube y para ser especialmente compatible con la API de AWS S3: el popular servicio propietario de almacenamiento de objetos disponible en el catálogo de Amazon Web Services (AWS) . Esto quiere decir que todas

las aplicaciones que sean capaces de comunicarse y trabajar con AWS S3 serán capaces de trabajar con MinIO.

MinIO ofrece un cliente en línea y una interfaz gráfica, además de soporte para colas de mensajes y otras bases de datos.

1.4.4.3 MongoDB

MongoDB [21] es quizá la base de datos NoSQL más famosa del mercado, y utiliza un almacenamiento en forma de documentos en formato BSON: un formato binario de serialización de datos que presenta muchos paralelismos con JSON.

Se trata de un producto software de código abierto maduro, multipropósito y con un catálogo extenso de funcionalidades. Aunque no proporciona estrictamente las mismas garantías ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad) que sí ofrece una base de datos relacional clásica (e.g. PostgreSQL), el equipo de desarrollo de MongoDB está realizando avances significativos en cubrir estas necesidades en los últimos años.

MongoDB no es un reemplazo para una base de datos relacional, pero se ajusta mejor a casos de uso que requieren de la flexibilidad de un modelo de datos que no tienen especial relación entre ellos. En el caso del presente proyecto, nos encontramos con que los vídeos a almacenar son ficheros con nula relación entre ellos.

MongoDB no está optimizado para el almacenamiento de archivos multimedia para su posterior acceso (es necesario realizar operaciones de codificación y decodificación de la información), pero sus cortos tiempo de respuesta y su conversión en un estándar en el mercado hace que su estudio sea casi ineludible.

1.5 Alternativa escogida

La alternativa escogida es la siguiente:

- **Unity** como motor gráfico y entorno de trabajo.
- **Blender** para el modelado ocasional de modelos 3D.
- **Visual Studio** para editar código (gracias a disponer de licencia por ser estudiante), pues se utilizará Windows como sistema operativo y Visual Studio es una alternativa que se integra a la perfección con **Unity** (que recordemos, a su vez, utiliza el lenguaje C#).
- **MinIO** o un sencillo servidor de ficheros para el almacenamiento y visionado de videos que permita realizar las pruebas en el prototipado del proyecto.

La razón de haber escogido estas alternativas frente al resto de las descritas no es otra que el conocimiento previo que se tiene de ellas: Se ha utilizado **Unity** en un breve taller del máster, en cursos ajenos a la universidad y en las prácticas en empresa de la titulación (además, se tiene amplia experiencia en el desarrollo con C#). Por otro lado, es obvio escoger **Blender** si prestamos atención al precio de las alternativas y que se trata de un software de modelado 3D (no arquitectónico) gratuito.

Respecto a **MinIO** se opta por esta alternativa debido a su alta eficiencia en el almacenamiento de objetos y a ofrecer una alternativa local y gratuita al almacenamiento de objetos S3 que ofrece Amazon Web Services. Por otro lado, si en un futuro se precisa de hacer uso de una alternativa cloud (con los gastos que ello conlleva), MinIO es altamente compatible por lo que la migración podría realizarse sin ningún problema y el acceso a los datos se realizaría de forma análoga, lo que no elevaría la carga de trabajo en el desarrollo. En el caso de no ser compatible con el ámbito de la aplicación (inmersiva y web), puede optarse por almacenarse los vídeos como archivos estáticos en un servidor de ficheros local para el desarrollo de pruebas sencillas.

También cabe destacar que, como se comentaba al comienzo del presente documento, la colaboración con la Fundación CTIC hace que gran parte del desarrollo

de este trabajo se lleve a cabo en sus oficinas con sus herramientas. Es por ello por lo que las alternativas utilizadas han ganado peso debido a que su departamento de tecnologías inmersivas hace uso de:

- *Unity* en su versión 2019.4.27 (LTS), cuya licencia han adquirido.
 - *Con WebGL Build Support.*
 - *A su vez, para la exportación del proyecto en Web se utilizará el estándar WebXR investigando y documentando para la Fundación CTIC su uso y posterior exportación utilizando Unity.*
- *Blender* en su versión 2.93.
- *Visual Studio* en su versión de 2017.
- *MinIO* debido al requisito de estudiar la compatibilidad entre Unity y dicho sistema de almacenamiento de objetos.
 - En el caso de no obtener el funcionamiento deseado puede optarse por desarrollarse un servidor sencillo de ficheros que permita almacenar vídeos y acceder a estos mediante peticiones http.

1.6 Hardware

Cuando el objetivo de un proyecto lleva consigo el trabajo y realización de una aplicación inmersiva es obvio pensar que debe hacerse uso de Hardware específico para la visualización.

En el momento de escoger el software con el que llevar a cabo el proyecto existían una cierta flexibilidad que no existe con el hardware. Los visores de realidad virtual son costosos y el autor de este trabajo no posee uno de su propiedad, es por ello por lo que se hace uso de los adquiridos por el departamento de tecnologías inmersivas de la Fundación CTIC. La aplicación será soportada, como mínimo, por estos dos

modelos de gafas de realidad virtual:

- **Oculus Quest.**
- **Meta Quest 2** (Anteriormente conocidas como Oculus Quest 2).

Casualmente la Universidad de Oviedo tiene el mismo modelo, por lo que se justifica también su uso para la presentación final del proyecto.

2 Planificación del proyecto

2.1 Planificación inicial

Se entiende como planificación inicial el plan a seguir para el desarrollo del proyecto en un inicio, comenzando en a mediados del mes de octubre de 2021 (mes en el que fue aprobado el proyecto por parte de la universidad) y finalizándolo a mediados del mes de junio.

2.1.1 *Tabla*

En la siguiente tabla se encuentra la planificación inicial del proyecto usando los delimitadores de fecha de inicio y de fin de las tareas y contabilizando los días laborales entre estas (lunes a viernes sin días festivos), se estimaba trabajar unas 20 horas semanales.

Teniendo en cuenta lo indicado en la siguiente tabla, la planificación inicial del proyecto consta de 740h horas de trabajo a lo largo de 7 meses a razón de (aproximadamente) 20 horas semanales de dedicación estimadas. En total: 185 días laborales.

Tabla 2.1. Planificación a priori.

Tarea	Fecha Inicio	Fecha Fin	Duración(d)
Sprint 1	18/10/2021	29/10/2021	12
Análisis de requisitos iniciales	18/10/2021	21/10/2021	4
Documentación de requisitos iniciales	21/10/2021	22/10/2021	2
Estudio y documentación de alternativas	25/10/2021	27/10/2021	3
Planificación y presupuesto	27/10/2021	29/10/2021	3
Sprint 2	02/11/2021	23/12/2021	38
Estudio inicial (VR, WebXR, controladores, físicas, etc.).	02/11/2021	23/12/2021	38
Sprint 3	27/12/2021	02/02/2022	21
Diseño del escenario	27/12/2021	29/12/2021	3
Creación del escenario	30/12/2021	25/01/2022	18
Sprint 4	26/01/2022	20/04/2022	93
Implementación de la aplicación	26/01/2022	20/04/2022	58
Diseño de la arquitectura	21/03/2022	25/03/2022	5
Desarrollo de la arquitectura	28/03/2022	20/04/2022	16
Grabación de contenido	01/04/2022	20/04/2022	12
Sprint 5	20/04/2022	11/05/2022	14
Diseño de pruebas	20/04/2022	25/04/2022	4
Pruebas	26/04/2022	05/05/2022	7
Documentación de pruebas	06/05/2022	10/05/2022	3
Sprint 6	11/05/2022	17/06/2022	5
Arreglo de errores	11/05/2022	17/05/2022	5
Presentación	18/05/2022	19/05/2022	2
Preparación de la presentación (Sprint 6)	18/05/2022	19/05/2022	2

2.1.2 Diagrama de Gantt

A continuación, se muestra en forma de diagrama de Gantt por meses el desglose de tareas de la planificación inicial del proyecto.

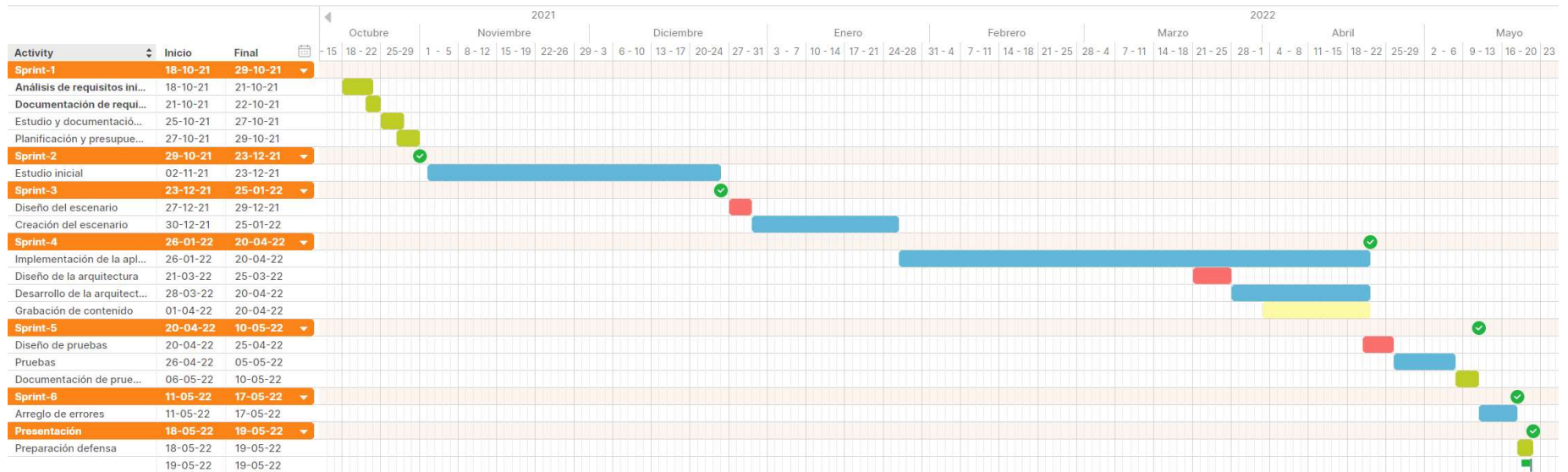


Ilustración 2.5. Diagrama de Gantt (Planificación inicial)

2.2 Planificación final

Debido a que el autor del presente proyecto se encontraba trabajando a tiempo completo, el seguimiento de la planificación no ha sido posible. La planificación inicial no se ha seguido rigurosamente y el desarrollo del proyecto se ha extendido más en el tiempo. Por tanto, se indica a continuación una **planificación final aproximada** de las horas de trabajo realizadas en el presente proyecto.

Esta planificación se indica en horas y no en días, debido a la imposibilidad de dedicarse a tiempo completo al proyecto (es decir, no se ha podido disponer de un número fijo de horas de desarrollo al día). Suma un total de **742 horas** aproximadas de desarrollo del proyecto.

Tabla 2.2. Planificación final.

Tarea	Duración(h)
Análisis de requisitos iniciales	10
Documentación de requisitos iniciales	4
Estudio y documentación de alternativas	16
Planificación y presupuesto	12
Estudio inicial (VR, controladores, físicas, etc.).	200
Diseño del escenario	8
Creación del escenario	60
Implementación de la aplicación	335
Diseño de la arquitectura	8
Desarrollo de la arquitectura	20
Grabación de contenido	1
Diseño de pruebas	10
Pruebas	20
Documentación de pruebas	8
Arreglo de errores	30
Preparación de la presentación	6

2.2.1 Hitos

Se mantienen los hitos fechados en la planificación inicial, siendo estos entregados en las siguientes fechas:

Tabla 2.3. Hitos en la planificación final.

Hito	Descripción	Fecha de entrega
Sprint 1	Documentación inicial, requisitos, alternativas, planificación y presupuesto.	02/11/2021
Sprint 2	Estudio inicial con mecánicas (VR, físicas, controladores, etc.).	28/01/2022
Sprint 3	Generación del escenario e inserción el motor.	29/03/2022
Sprint 4	Implementación de la aplicación (inserción de los resultados del estudio inicial en el nuevo escenario), arquitectura del proyecto y despliegue de esta.	11/07/2022
Sprint 5	Desarrollo de pruebas y documentación de resultados.	15/07/2022
Pruebas	Versión final del prototipado de la aplicación para su posterior defensa frente al tribunal.	17/07/2022

3 Fase de análisis

Se procede a documentar el análisis previo al desarrollo, donde se establecen los requisitos que debe cumplirse y los casos de uso que deberán pasar a desarrollarse en las posteriores fases del trabajo.

3.1 Requisitos

La aplicación deberá ser capaz de representar un mundo virtual de libre acceso, ofreciendo una interacción sencilla haciendo uso de la realidad virtual.

Deberá ser intuitiva y fácilmente usable para usuarios sin conocimientos específicos de informática, además tendrá que ser capaz de mostrar contenidos de

forma clara y concisa. A continuación, se listan los requisitos funcionales del sistema:

Tabla 3.4. Requisitos funcionales.

Código	Descripción
RF.1	La aplicación deberá ser accesible desde un navegador web sin necesidad de instalar ningún hardware específico para su acceso.
RF.2	La aplicación deberá ser de público acceso.
RF.3	La aplicación debe permitir el acceso, el movimiento, la interacción y la visualización de contenidos se disponga o no de hardware de realidad virtual
RF.4	La aplicación deberá contener todos los componentes básicos del motor a utilizar: Físicas, luces, scripting...
RF.5	La aplicación deberá ser capaz de mostrar contenidos dinámicos y de reproducir multimedia (audio, vídeo e imagen).
RF.6	La aplicación deberá mostrar información de contacto de diversas instituciones.
RF.7	La aplicación deberá permitir al usuario modificar aspectos de configuración que influyan en la interacción.
RF.8	Se proveerá de un sencillo método de añadido de vídeos para visualizar de forma dinámica en la escena.
RF.9	El despliegue de la aplicación debe realizarse de forma sencilla permitiendo una alta compatibilidad independientemente del sistema que la aloje.
...	...

Y, a continuación, otros requisitos no funcionales:

Tabla 3.5. Requisitos no funcionales.

Código	Descripción
RNF.1	La aplicación deberá seguir todos los estándares de usabilidad de W3C.
RNF.2	La aplicación deberá de experimentar tiempos de respuesta bajos y poca latencia.
RNF.3	La aplicación deberá de optimizar en todo momento la instanciación de objetos y la ejecución de ordenes en los bucles y funciones del motor.
RNF.4	La aplicación deberá optimizar el uso de luces, evitando las luces en tiempo real y priorizando las preprocesadas.
RNF.5	La aplicación deberá ser capaz de usarse por todo tipo de usuarios, ofreciendo una interfaz sencilla y amigable.
RNF.6	La aplicación deberá ser usable por usuarios con baja accesibilidad, permitiendo el movimiento por teletransporte (en lugar de translación) en el caso de la experiencia inmersiva con el fin de evitar posibles mareos o problemas de salud.
RNF.7	Se facilitará un despliegue que permita una rápida y sencilla actualización de la aplicación.
...	...

3.2 Especificación de casos de uso

A continuación, se detallarán los casos de uso del desarrollo. Se prevé

implementar la aplicación dividida en diversas salas o escenas con funcionalidades diferentes entre ellas, por tanto se muestran los casos de uso primero desde un punto de vista general y después de una manera más detallada para cada uno de estas salas.

3.2.1 Casos de uso generales

A continuación, se listan los diagramas generales de los casos de uso que podrán encontrarse en cada una de las escenas del presente proyecto.

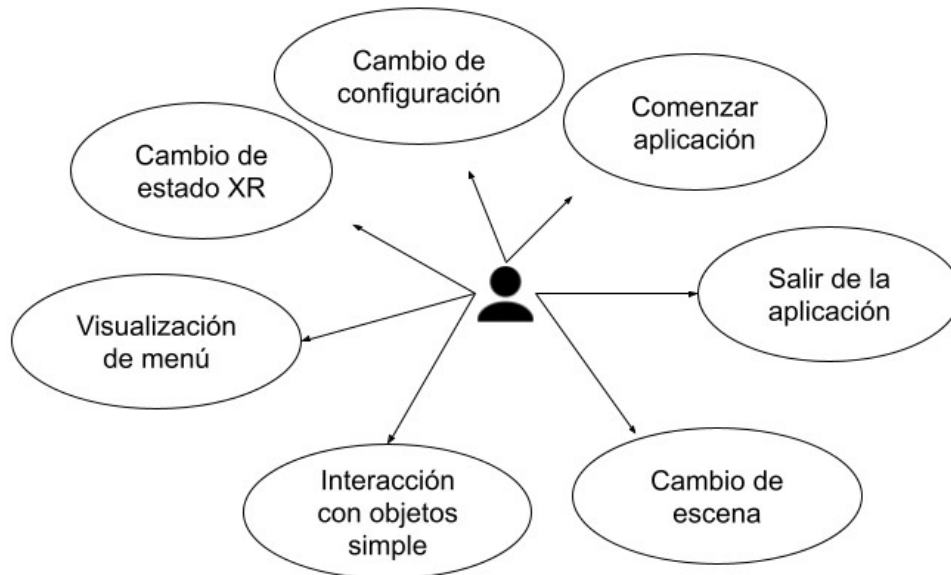


Ilustración 3.6. Casos de uso (General)

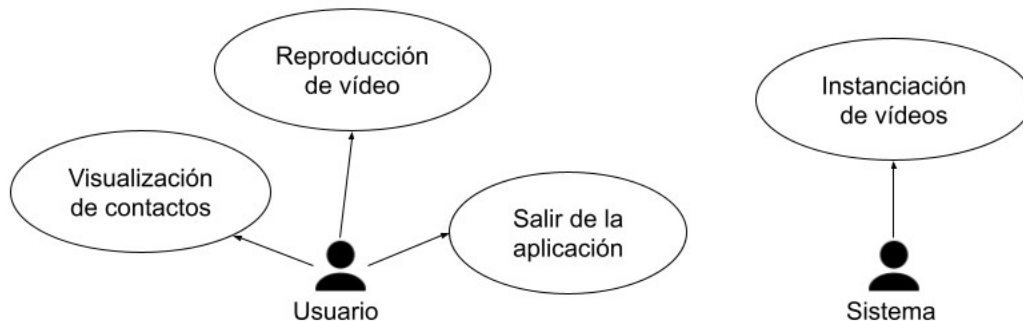


Ilustración 3.7. Casos de uso (Sala principal)

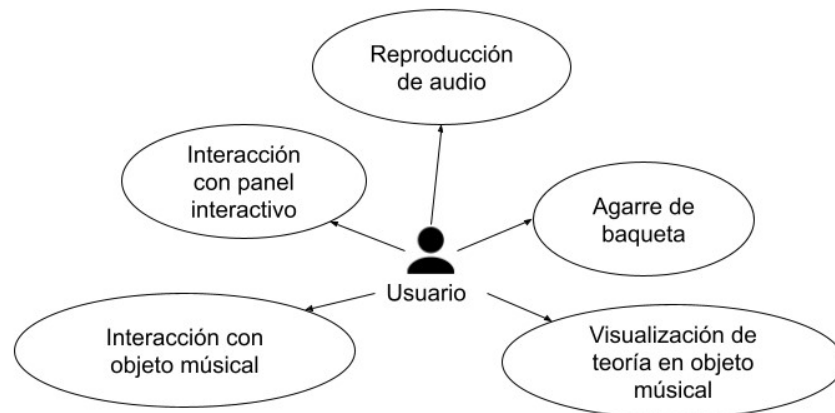


Ilustración 3.8. Casos de uso (Sala secundaria)

3.2.2 Casos de uso detallados

En el presente subapartado se detallarán en formato tabla los casos de uso antes listados. Fíjese que el usuario podrá llevar a cabo el mismo caso de uso en varias escenas posibles (Todos los casos del diagrama general, como por ejemplo *Cambio de estado XR* o *interacción con el menú*) por lo que para evitar duplicidades y teniendo en cuenta que se llevarán a cabo de la misma forma, este tipo de casos de uso se detallarán una sola vez.

También se debe de tener en cuenta que las tablas que detallarán los casos de uso no poseerán un título de tabla para evitar alargar de manera innecesaria el índice de tablas.

XR-001		Cambiar a modo inmersivo
Dependencias		
Comienza operación	la	El usuario
Finaliza operación	la	El usuario
Precondición		El usuario ha accedido a la aplicación y dispone de hardware de realidad virtual (ha accedido desde él o lo tiene conectado y configurado en el equipo con el que accede).
Descripción		El usuario ha accedido a la aplicación y la visualiza desde el navegador, puede cambiar el modo XR.

XR-001		Cambiar a modo inmersivo	
Secuencia normal	Paso	Acción	
	1	Se habilita el botón VR.	
	2	El usuario clica en el botón VR.	
	3	Se solicita permiso para acceder al <i>hardware</i> .	
	4	El contenido pasa a reproducirse en el hardware de realidad virtual y se habilita la interacción VR.	
Postcondición	El estado XR se modifica y pasa a estar activo.		
Excepciones			
Comentarios	<p>Aunque existen diversas formas de acceder a la aplicación en modo inmersivo (desde el equipo con el visor conectado a este o desde el propio visor en modo <i>standalone</i>), el proceso para activar la realidad virtual es el mismo: Presionar en el botón VR que se sitúa fuera de la pantalla de juego.</p> <p>Este proceso se puede llevar a cabo en cualquier escena.</p>		

XR-002		Cambiar a modo no inmersivo	
Dependencias			
Comienza operación	la	El usuario	
Finaliza operación	la	El sistema	
Precondición	El usuario ha accedido a la aplicación y dispone de hardware de realidad virtual conectado al equipo, desde donde está reproduciendo el contenido.		
Descripción	El usuario busca cambiar el modo a no inmersivo y usar el equipo en vez del visor de realidad virtual.		
Secuencia normal	Paso	Acción	
	1	El usuario se quita el visor de realidad virtual	
	2	El sistema detecta que el equipo se ha desconectado y pasa a mostrar el contenido en el navegador.	
	3	Se modifica el estado XR a no inmersivo.	
Postcondición	El estado XR se modifica y pasa a estar desactivado.		
Excepciones			
Comentarios	Este proceso se puede llevar a cabo en cualquier escena.		

XR-003		Cambiar a modo no inmersivo (<i>standalone</i>)	
Dependencias		<ul style="list-style-type: none"> • XR-002 	
Comienza operación	la	El usuario	
Finaliza operación	la	El sistema	
Precondición		El usuario ha accedido a la aplicación desde el navegador del <i>hardware</i> de realidad virtual y ha activado el modo inmersivo.	
Descripción		El usuario busca cambiar el modo a no inmersivo.	
Secuencia normal	Paso	Acción	
	1	El usuario cancela el modo inmersivo desde el menú del <i>hardware</i> de realidad virtual.	
	2	El sistema detecta que el equipo se ha desconectado y pasa a mostrar en modo no inmersivo.	
	3	Se modifica el estado XR a no inmersivo.	
Postcondición		El estado XR se modifica y pasa a estar desactivado.	
Excepciones			
Comentarios		Este proceso se puede llevar a cabo en cualquier escena.	

MENÚ-001		Modificar configuración	
Dependencias			
Comienza operación	la	El usuario	
Finaliza operación	la	El usuario	
Precondición		El usuario ha accedido a la aplicación y no ha activado el modo inmersivo	
Descripción		El usuario acaba de acceder a la aplicación y visualiza el menú principal, donde puede modificar varias configuraciones.	
Secuencia normal	Paso	Acción	
	1	El usuario accede a la url de la aplicación.	
	2	El usuario visualiza el menú principal	
	3	El usuario puede modificar ciertos aspectos de la aplicación tanto para modo VR como no VR.	
Postcondición		La configuración es modificada.	
Excepciones			
Comentarios			

MENÚ-003		Comenzar la aplicación	
Dependencias			
Comienza operación	la	El usuario	
Finaliza operación	la	El sistema	
Precondición		El usuario ha accedido a la aplicación y no ha activado el modo inmersivo.	
Descripción		El usuario acaba de acceder a la aplicación y visualiza el menú principal, donde puede modificar varias configuraciones.	
Secuencia normal	Paso	Acción	
	1	Se muestra el menú principal.	
	2	El usuario clica en el botón de comenzar.	
	3	El usuario es teletransportado a la escena principal.	
4	El contenido pasa a reproducirse en el hardware de realidad virtual y se habilita la interacción VR.		
Postcondición		El estado XR se modifica y pasa a estar activo.	
Excepciones			
Comentarios		Aunque existen diversas formas de acceder a la aplicación en modo inmersivo (desde el equipo con el visor conectado a este o desde el propio visor en modo <i>standalone</i>), el proceso para activar la realidad virtual es el mismo: Presionar en el botón VR que se sitúa fuera de la pantalla de juego.	

MENÚ-004		Modificar configuración (aplicación)	
Dependencias		<ul style="list-style-type: none"> • MENÚ-003 	
Comienza operación	la	El usuario	
Finaliza operación	la	El usuario	
Precondición		El usuario ha comenzado la aplicación y se encuentra en cualquier otra escena que no es la de menú.	
Descripción		El usuario busca abrir el menú para modificar parámetros en la configuración.	
Secuencia normal	Paso	Acción	
	1	El usuario abre el menú.	
	2	El usuario modifica la configuración (sensibilidad, activación de teletransporte, etc).	
3	Se actualiza la configuración		

MENÚ-004	Modificar configuración (aplicación)
Postcondición	El menú pasa a estar activo.
Excepciones	
Comentarios	Este proceso puede realizarse desde cualquier escena y en cualquier estado XR.

MENÚ-005	Cerrar menú	
Dependencias	<ul style="list-style-type: none"> • MENÚ-004 	
Comienza operación	la	El usuario
Finaliza operación	la	El usuario
Precondición	El usuario ha comenzado la aplicación y se encuentra en cualquier otra escena visualizando el menú.	
Descripción	El usuario ya ha modificado los parámetros necesarios en el menú y busca reanudar la actividad.	
Secuencia normal	Paso	Acción
	1	El usuario clicla en el botón reanudar.
	2	El menú pasa a ocultarse.
Postcondición	El menú pasa a estar desactivado.	
Excepciones		
Comentarios	Este proceso puede realizarse desde cualquier escena y en cualquier estado XR.	

MENÚ-005	Salir de la aplicación	
Dependencias	<ul style="list-style-type: none"> • MENÚ-004 	
Comienza operación	la	El usuario
Finaliza operación	la	El usuario
Precondición	El usuario ha accedido a la aplicación y decide salir de esta.	
Descripción	El usuario desea desconectarse de la aplicación	
Secuencia normal	Paso	Acción
	1	El usuario abre el menú.
	2	El usuario clicla el botón de salir.
	3	El usuario sale de la aplicación.

MENÚ-005	Salir de la aplicación
Postcondición	-
Excepciones	
Comentarios	Este proceso puede realizarse desde cualquier escena y en cualquier estado XR. Otro proceso para llevarlo a cabo es, directamente y debido a desarrollar una aplicación web, cerrar la pestaña del navegador.

MAIN-001	Cambio de escena	
Dependencias	<ul style="list-style-type: none"> • MENÚ-003 	
Comienza operación	la	El sistema
Finaliza operación	la	El sistema
Precondición	El usuario se encuentra en la aplicación y comienza el proceso de cambio de escena.	
Descripción	El usuario busca cambiar de escena, por lo que se acerca a la zona que activa este evento.	
Secuencia normal	Paso	Acción
	1	El usuario se acerca a la zona de cambio de escena.
	2	El sistema detecta la colisión y lanza un evento.
	3	El sistema teletransporta al usuario a una nueva escena.
Postcondición	El usuario se encuentra en una escena diferente.	
Excepciones		
Comentarios	Este proceso puede realizarse desde cualquier escena y en cualquier estado XR.	

MAIN-002	Interacción simple (no inmersiva)	
Dependencias	<ul style="list-style-type: none"> • MENÚ-003 	
Comienza operación	la	El usuario
Finaliza operación	la	El usuario
Precondición	El usuario se encuentra en estado XR desactivado y quiere agarrar un objeto y lanzarlo, se encuentra en cualquier sala (excepto escena de menú).	

MAIN-002		Interacción simple (no inmersiva)	
Descripción	El usuario busca interactuar de forma simple con un objeto (agarrarlo, soltarlo o lanzarlo).		
Secuencia normal	Paso	Acción	
	1	El usuario hace clic sobre un objeto interactivo.	
	2	El objeto pasa a estar en manos del usuario (en el cursor del ratón), manteniendo el clic pulsado.	
	3	El usuario lo lanza, soltando el clic,	
4	El objeto deja de permanecer frente al usuario, sigue la inercia y colisiona con el escenario.		
Postcondición	El usuario ya no tiene un objeto y el objeto no se encuentra en su posición inicial.		
Excepciones	Paso	Acción	
	2	El objeto en el que ha hecho clic el usuario no se trata de un objeto interactivo.	
		E1.1	El objeto se mantiene su lugar y no hay interacción posible.
Comentarios			

MAIN-003		Interacción simple (inmersiva)	
Dependencias	<ul style="list-style-type: none"> • MENÚ-003 		
Comienza operación	la	El usuario	
Finaliza operación	la	El usuario	
Precondición	El usuario se encuentra en estado XR activado y quiere agarrar un objeto con sus manos y lanzarlo, se encuentra en cualquier sala (excepto escena de menú).		
Descripción	El usuario busca interactuar de forma simple con un objeto (agarrarlo, soltarlo o lanzarlo).		
Secuencia normal	Paso	Acción	
	1	El usuario acerca una de sus manos al objeto	
	2	El usuario clicca en el botón del controlador correspondiente a la mano acercada.	
	3	El objeto pasa a encontrarse en la mano del usuario, manteniendo el botón del controlador pulsado.	
	4	El usuario lo lanza, soltando el botón del controlador	
5	El objeto deja de permanecer en la mano del usuario, sigue la inercia y colisiona con el escenario.		

MAIN-003		Interacción simple (inmersiva)	
Postcondición	El usuario ya no tiene un objeto en la mano y el objeto no se encuentra en su posición inicial.		
Excepciones	3	Paso Acción	
			El objeto que ha intentado agarrar el usuario no se trata de un objeto interactivo.
	E1.1	El objeto se mantiene su lugar y no hay interacción posible.	
Comentarios			

MAIN-004		Mostrar contacto de instituciones	
Dependencias	<ul style="list-style-type: none"> • MENÚ-003 		
Comienza operación	la	El usuario	
Finaliza operación	la	El sistema	
Precondición	El usuario, independientemente de su estado XR, se encuentra en la sala principal.		
Descripción	El usuario busca consultar el listado de instituciones musicales de la región y sus vías de contacto.		
Secuencia normal	Paso	Acción	
	1	El usuario interactúa con el objeto interactivo.	
	2	El listado de instituciones es desplegado y resulta visible para el usuario.	
Postcondición	El usuario ya no tiene un objeto en la mano y el objeto no se encuentra en su posición inicial.		
Excepciones			
Comentarios	El listado de contactos se prevé que se muestre al usuario en un objeto desplegable. Si el listado es muy amplio o se implementa de tal manera que se indique una descripción detallada de cada institución es probable implementar un sistema de paginación para que el usuario visualice uno a uno cada elemento del listado.		

MAIN-005		Instanciación de vídeos	
Dependencias	<ul style="list-style-type: none"> • MENÚ-003 		
Comienza operación	la	El sistema	

MAIN-005		Instanciación de vídeos	
Finaliza operación	la	El sistema	
Precondición		El usuario, independientemente de su estado XR, se encuentra en la sala principal.	
Descripción		El sistema comienza el proceso de instanciación de vídeos. Se prevé que cada vídeo se implemente en un objeto interactivo especial.	
Secuencia normal		Paso	Acción
		1	El usuario aparece en la escena principal.
		2	El sistema comienza el proceso de instanciación de vídeos. Lanza una petición al servicio de base de datos
		3	El sistema obtiene el listado de vídeos almacenados en la base de datos.
		4	El sistema instancia un objeto interactivo para cada uno de los vídeos almacenados.
Postcondición		Han aparecido varios objetos en escena, uno por vídeo.	
Excepciones		Paso	Acción
		3	No hay conexión con la base de datos.
			E1.1 Se muestra un error por consola a modo de depuración, la excepción debe estar controlada y el flujo del resto de procesos de la aplicación debe seguir su curso.
Comentarios			

MAIN-006		Reproducción de vídeo	
Dependencias		<ul style="list-style-type: none"> • MAIN-005 	
Comienza operación	la	El usuario	
Finaliza operación	la	El sistema	
Precondición		El usuario, independientemente de su estado XR, se encuentra en la sala principal.	
Descripción		El usuario selecciona un vídeo concreto para su posterior reproducción y visualización	
Secuencia normal		Paso	Acción
		1	El usuario selecciona uno de los objetos interactivos de vídeo instanciados por el sistema.

MAIN-006		Reproducción de vídeo	
	2	El usuario coloca el objeto interactivo de vídeo en el reproductor.	
	3	El objeto interactúa con el botón de <i>Play</i> .	
	4	El vídeo se reproduce.	
Postcondición	El vídeo se encuentra reproduciéndose.		
Excepciones	Paso	Acción	
	4	No hay conexión con la base de datos.	
		E1.1	Se muestra un error por consola a modo de depuración, la excepción debe estar controlada y el flujo del resto de procesos de la aplicación debe seguir su curso.
Comentarios			

MAIN-007		Pausa de vídeo	
Dependencias	<ul style="list-style-type: none"> MAIN-006 		
Comienza operación	la	El usuario	
Finaliza operación	la	El sistema	
Precondición	El usuario, independientemente de su estado XR, se encuentra en la sala principal con un vídeo reproduciéndose.		
Descripción	El usuario selecciona busca pausar la reproducción del vídeo actual.		
Secuencia normal	Paso	Acción	
	1	El usuario interactúa con el botón de <i>Stop</i> .	
	2	El vídeo deja de reproducirse.	
	3	El objeto interactúa con el botón de <i>Play</i> .	
	4	El vídeo se reproduce.	
Postcondición	El vídeo ya no se está reproduciendo.		
Excepciones			
Comentarios			

SECOND-001		Interacción con objeto interactivo (texto).	
Dependencias	<ul style="list-style-type: none"> MAIN-003 		

SECOND-001		Interacción con objeto interactivo (texto).	
Comienza operación	la	El usuario	
Finaliza operación	la	El sistema	
Precondición	El usuario se encuentra en una de las salas secundarias independientemente de su estado XR.		
Descripción	Interacción con el objeto interactivo más común en la aplicación; Panel de texto.		
Secuencia normal	Paso	Acción	
	1	El usuario interactúa con el objeto interactivo.	
	2	Se despliega el panel de texto adjunto al objeto.	
	3	El panel desplegado se gira a la cámara del usuario	
Postcondición	El panel de texto se muestra en el escenario		
Excepciones	Paso	Acción	
	2	Ya hay otro objeto con su panel desplegado.	
		E1.1	Se oculta el panel desplegado y se continua con el proceso de interacción.
Comentarios			

SECOND-002		Interacción con objeto interactivo (texto) – Cerrar.	
Dependencias	<ul style="list-style-type: none"> SECOND-001 		
Comienza operación	la	El usuario	
Finaliza operación	la	El sistema	
Precondición	El usuario se encuentra en una de las salas secundarias independientemente de su estado XR con un panel de texto desplegado.		
Descripción	Proceso de cierre del objeto interactivo más común, el panel de texto.		
Secuencia normal	Paso	Acción	
	1	El usuario interactúa con el objeto interactivo cuyo panel esta desplegado.	
	2	Se oculta el panel de texto adjunto al objeto.	
Postcondición	El panel de texto ya no se muestra en el escenario.		
Excepciones			
Comentarios			

SECOND-003		Interacción con objeto interactivo (texto y audio).	
Dependencias		• MAIN-003	
Comienza operación	la	El usuario	
Finaliza operación	la	El sistema	
Precondición		El usuario se encuentra en una de las salas secundarias independientemente de su estado XR.	
Descripción		Interacción con otro objeto interactivo común en la aplicación; Panel de texto con botones de audio.	
Secuencia normal		Paso	Acción
		1	El usuario interactúa con el objeto interactivo.
		2	Se despliega el panel de texto con botones adjunto al objeto.
		3	El panel desplegado se gira a la cámara del usuario
		4	El usuario interactúa con uno de los botones del panel
		5	Se reproduce el audio adherido al botón presionado.
Postcondición		El panel de texto con botones se muestra en el escenario.	
Excepciones		Paso	Acción
		2	Ya hay otro objeto con su panel desplegado.
			E1.1 Se oculta el panel desplegado y se continúa con el proceso de interacción.
		5	Ya hay otro audio reproduciéndose en el panel.
	E2.1 Se pausa el audio anterior y se continúa con el proceso de interacción.		
Comentarios			

SECOND-004		Agarre de baqueta	
Dependencias		• MAIN-003	
Comienza operación	la	El usuario	
Finaliza operación	la	El sistema	
Precondición		El usuario se encuentra en una de las salas secundarias donde se puede agarrar una baqueta para golpear objetos musicales.	

SECOND-004		Agarre de baqueta	
Descripción	Proceso de agarre de un objeto interactivo especial: baqueta.		
Secuencia normal	Paso	Acción	
	1	El usuario interactúa en modo inmersivo con el objeto interactivo para agarrarlo con su mano.	
	2	El objeto modifica su rotación y posición para colocarse de forma vertical en la mano del usuario.	
Postcondición	El panel de texto ya no se muestra en el escenario.		
Excepciones	Paso	Acción	
	2	El usuario no se encuentra en modo inmersivo.	
E1.1		El objeto se agarra como un objeto interactivo estándar, sin modificar su rotación.	
Comentarios			

SECOND-005		Interacción con objeto musical	
Dependencias	<ul style="list-style-type: none"> • MAIN-003 		
Comienza operación	la	El usuario	
Finaliza operación	la	El sistema	
Precondición	El usuario se encuentra en una de las salas secundarias donde visualiza objetos musicales.		
Descripción	Proceso para jugar y tocar música.		
Secuencia normal	Paso	Acción	
	1	El usuario golpea el objeto musical con una baqueta	
	2	El objeto emite su sonido correspondiente.	
Postcondición	-		
Excepciones	Paso	Acción	
	1	El usuario no tiene un objeto para golpear.	
E1.1		El usuario puede interactuar con el objeto, siguiendo la secuencia normal de este caso de uso.	
Comentarios			

SECOND-006		Visualización de teoría en objeto musical	
Dependencias	<ul style="list-style-type: none"> • MAIN-003 		

SECOND-006		Visualización de teoría en objeto musical	
Comienza operación	la	El usuario	
Finaliza operación	la	El sistema	
Precondición	El usuario se encuentra en una de las salas secundarias donde visualiza objetos musicales y cintas de teoría.		
Descripción	Proceso para jugar y tocar música sobre una escala.		
Secuencia normal	Paso	Acción	
	1	El usuario interacciona con la cinta de teoría.	
	2	La cinta de teoría pasa a manos del jugador.	
	3	El usuario coloca la cinta de teoría en el lector de cintas.	
	4	El sistema ilumina los objetos musicales siguiendo la escala asignada en la cinta en el tono escogido.	
	5	El usuario presiona el botón del lector para cambiar la tonalidad de la escala.	
6	Se modifican los objetos iluminados y ahora se resaltan los correspondientes a la nueva tonalidad, siguiendo la escala de la cinta introducida.		
Postcondición	Los objetos musicales correspondientes a la escala introducida y al tono escogido son iluminados.		
Excepciones			
Comentarios			

SECOND-007		Cancelar estudio de escala	
Dependencias		<ul style="list-style-type: none"> SECOND-007 	
Comienza operación	la	El usuario	
Finaliza operación	la	El sistema	
Precondición	El usuario se encuentra en una de las salas secundarias donde visualiza objetos musicales y cintas de teoría. Una cinta de teoría está colocada en el lector.		
Descripción	Proceso para jugar y tocar música sobre una escala.		
Secuencia normal	Paso	Acción	
	1	El usuario interacciona con la cinta de teoría.	
	2	La cinta de teoría pasa a manos del jugador.	

SECOND-007		Cancelar estudio de escala	
	3	Los objetos musicales pasan a no estar iluminados.	
Postcondición	No hay ningún objeto musical iluminado.		
Excepciones			
Comentarios			

4 Fase de diseño

Se procede a documentar el diseño previo al desarrollo, donde se establecen a muy alto nivel que clases han de implementarse y que atributos y métodos contendrán cada una de estas. Además, se introducirá la arquitectura del sistema.

4.1 Diagrama de clases

Como parte de la fase de diseño a continuación se muestran los diagramas de clase que se prevén implementar:

Para hacer el diagrama más legible se ha optado por dividirlo en tres diagramas independientes: Uno general, otro para el jugador y otro para el resto de los objetos interactivos.

Hay que destacar que, tal y como se muestra en el diagrama de clases general, todos los scripts de los diagramas específicos heredaran de *MonoBehaviour* (clase padre de Unity necesaria para el funcionamiento de los scripts en el motor).

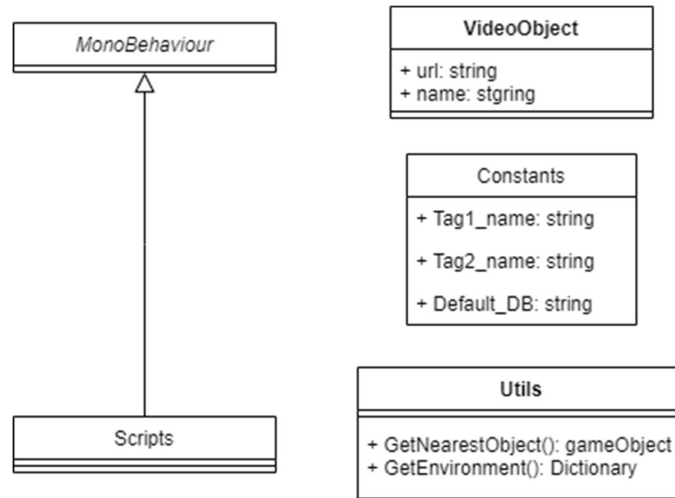


Ilustración 4.9. Diagrama de clases (General)

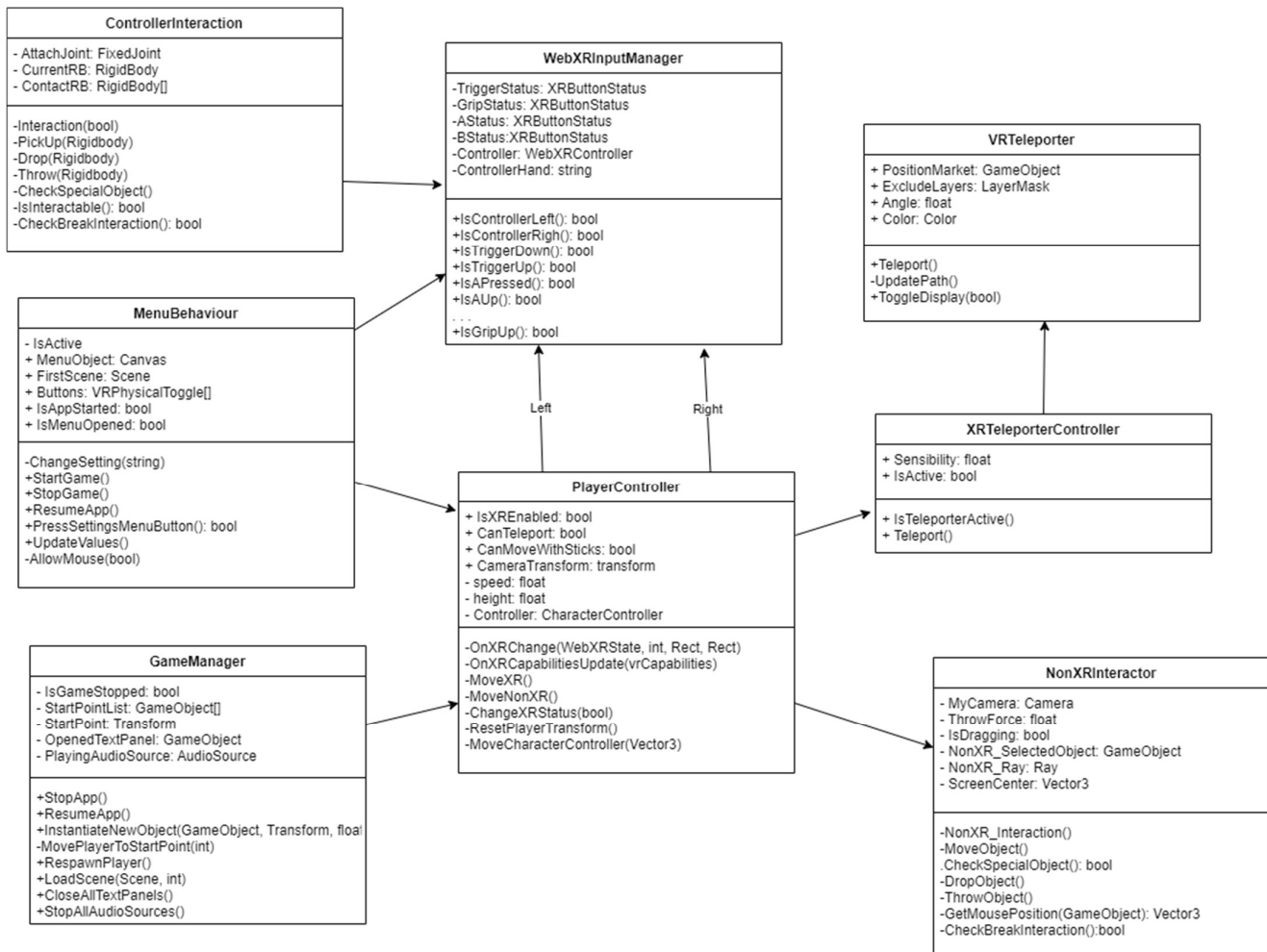


Ilustración 4.10. Diagrama de clases (Jugador)

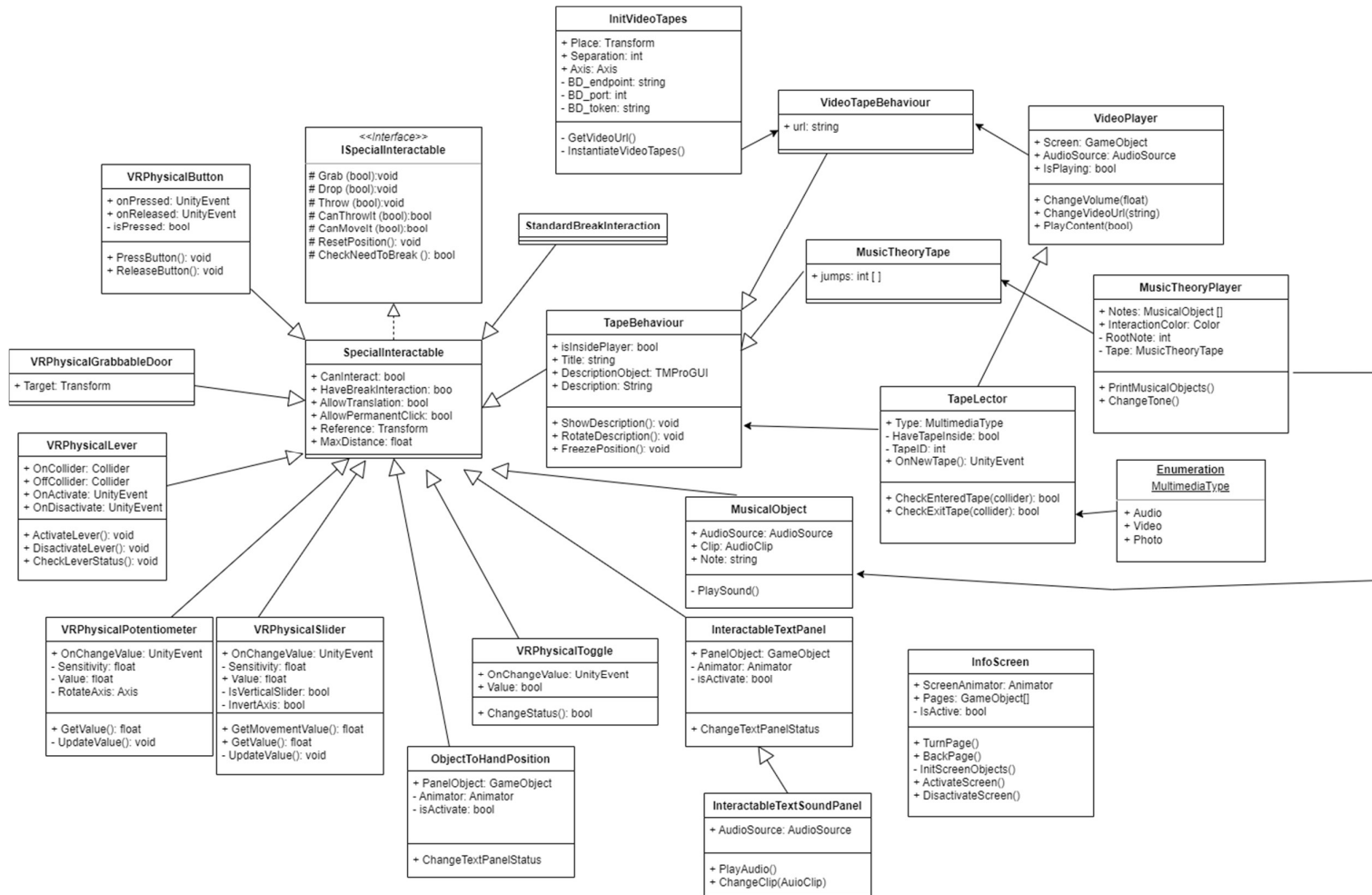


Ilustración 4.11. Diagrama de clases (Interactivos)

4.2 Arquitectura

La arquitectura del sistema es muy sencilla. Tan solo encontraremos **dos servicios independientes**: Un **servicio de servidor web** que alojará la aplicación Unity para hacerla accesible a través del navegador y un **servicio de Base de Datos** para albergar los videos que el usuario podrá visualizar.

Se prevé realizar un despliegue de la arquitectura en forma de microservicios utilizando tecnología de contenedores.

5 Estudio inicial

Antes de trabajar en la escena del proyecto, se ha realizado un pequeño estudio inicial para cumplir con algunos objetivos:

- Realizar un estudio de alternativas para escoger el método de implementación de una aplicación inmersiva, tal y como se detalló en el apartado *1.4 Estudio de alternativas*,
- Familiarizarse y recordar el **entorno de trabajo de Unity**.
- Practicar y **comprender** el funcionamiento de las **físicas, luces, mecánicas y configuraciones del motor**.
- Principalmente, practicar en investigar sobre una **programación para realidad virtual** que incluya la **interacción** del usuario.
- Estudiar el comportamiento de las cámaras, tanto en realidad virtual como sin ella.
- Realizar modelos sencillos en **Blender** para familiarizarse con la herramienta.
- Realizar el desarrollo y pruebas sobre los scripts creados para el presente proyecto.

Es por ello por lo que se ha desarrollado un escenario formado por tres escenas (una para el menú, la principal y otra muy básica para practicar cambios entre escenas), que contienen objetos básicos, algunos elementos interactivos y las mecánicas necesarias para el movimiento e interacción del usuario, se muestra a continuación:

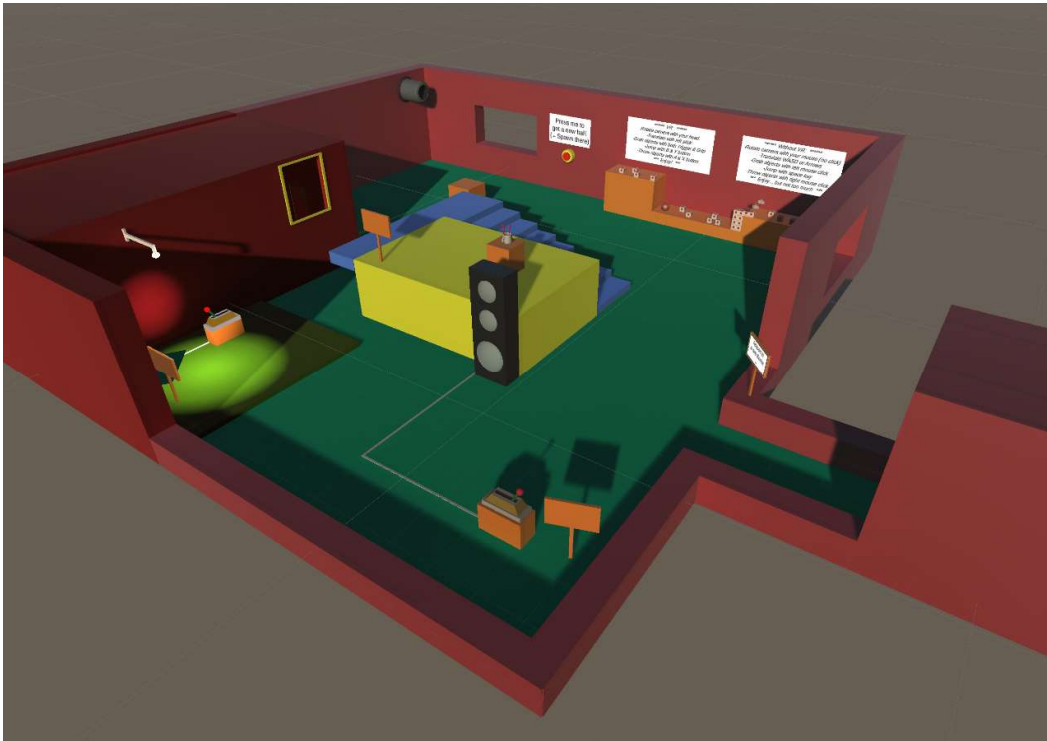


Ilustración 5.12. Escenario del estudio inicial.

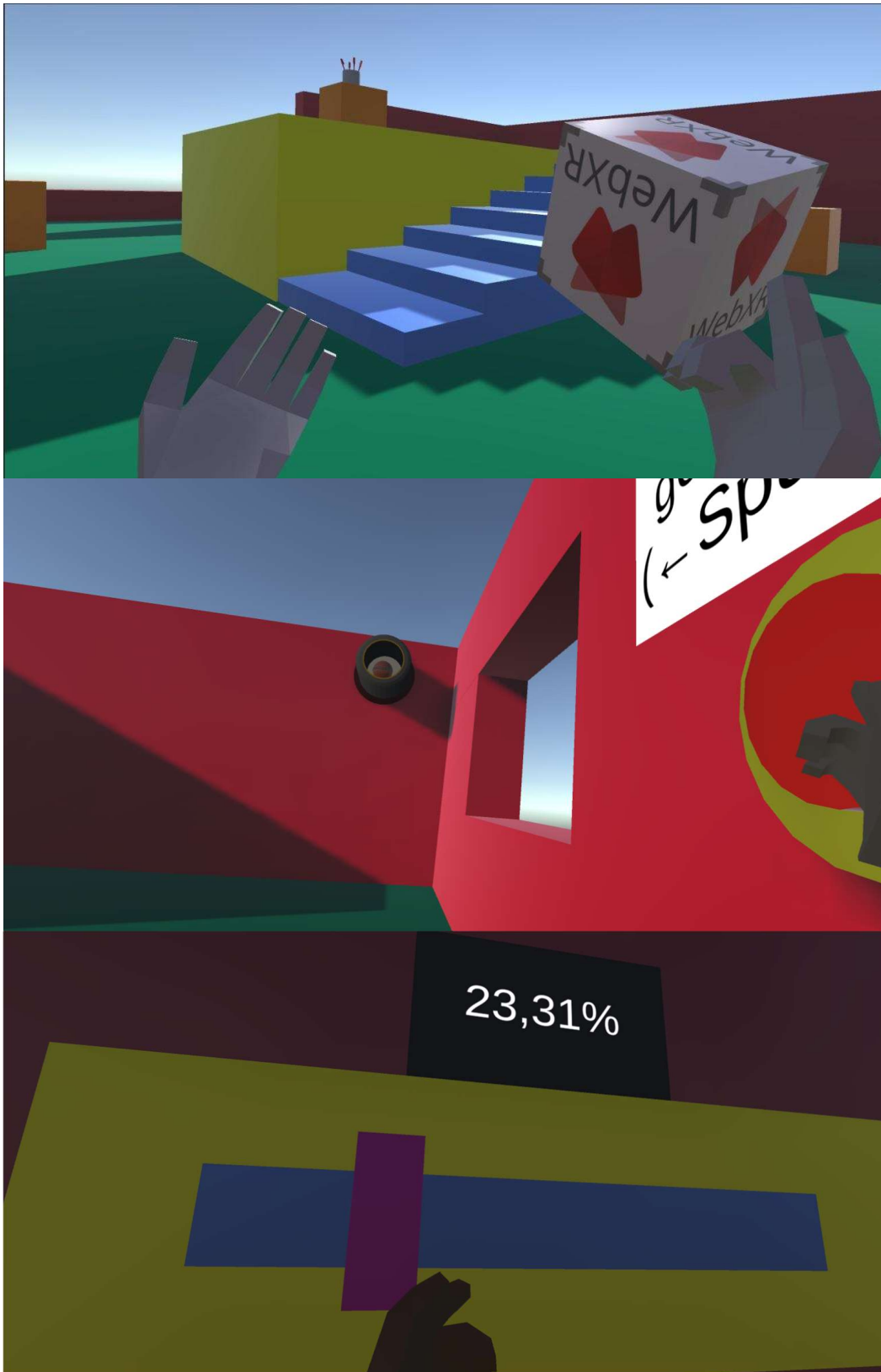




Ilustración 5.13. Interacción inmersiva en el estudio inicial.



Ilustración 5.14. Interacción no inmersiva en el estudio inicial

5.1 Características y objetivo del escenario

Este pequeño proyecto de investigación inicial cumple con las siguientes **características** que son, a su vez, requisitos del proyecto principal:

- Es accesible mediante web, por lo que permite la exportación del proyecto desde Unity utilizando WebXR.
- Permite la interacción tanto con Hardware de realidad virtual como sin él.
- Permite un movimiento libre del usuario.

- En el caso de la experiencia inmersiva con hardware de realidad virtual se permite tanto el movimiento por translación como por teletransporte usando los controladores.
- En el caso de una experiencia sin hardware de realidad virtual se permite el movimiento con teclado y ratón.

Las **interacciones** que el usuario puede llevar a cabo son las siguientes:

- Agarre y lanzamiento de objetos marcados para tal propósito, comprobando a su vez cómo se comportan las físicas del motor.
 - Con ambas manos si usa hardware de realidad virtual.
 - Con el ratón si no usa hardware específico.
- Accionamiento de mecanismos u objetos interactivos especiales tales como: Interruptores, palancas, potenciómetros, *sliders*, botones, puertas, etc.
- Cambio entre escenas.
- Reproducción de música.

Debido a que el proyecto principal deberá contener todas las interacciones, controladores y mecanismos que este estudio inicial ha desarrollado, gracias al desarrollo de esta pequeña escena inicial es posible la futura exportación de los objetos y componentes utilizados (denominamos *prefabs*) y de las *scripts* desarrolladas para su posterior uso en el proyecto principal, adelantando y facilitando el posterior trabajo.

5.2 Tecnologías y desarrollo

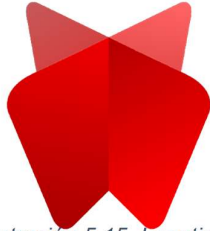


Ilustración 5.15. Logotipo de WebXR.

Se ha utilizado el paquete *Unity WebXR Export* [22] propiedad de Mozilla, aunque usando una versión de un usuario tercero: *De-Panther* [23] que subsana varios errores que se encuentran en la versión oficial (que se encuentra desactualizada e inservible).

Gracias al uso de los paquetes descritos anteriormente se han obtenido plantillas para la posterior exportación del proyecto utilizando *WebGL* y permitiendo al usuario el uso tanto inmersivo (permitiendo el acceso al hardware específico) como no en un navegador web.

Además, se han otorgado diversos *scripts* que permitían la detección y manejo de dicho hardware de realidad virtual para activar o desactivar diversas funciones y para acceder a los controladores de este.

Se han utilizado otros paquetes de terceros, pero todos los orientados al movimiento e interacción han sido sustancialmente modificados para adecuarse a las necesidades del proyecto.

5.2.1 Acceso a la aplicación

Hay tres maneras de acceder a una aplicación exportada (tanto el estudio inicial como en el proyecto, en el futuro) con el método utilizado:

1. Desde el navegador, en un ordenador, sin hardware de realidad virtual. Se accede sin él y se interacciona con ratón y teclado.
2. Desde el navegador, en un ordenador, con hardware de realidad virtual conectado. Este es detectado y se permite el cambio a modo inmersivo clicando en el botón pertinente.

3. Desde el propio equipo de realidad virtual (en modo *standalone*). Se accede desde el navegador del visor y se interactúa en modo inmersivo.
 - a. Nota: Accede desde el navegador del equipo de realidad virtual será como acceder desde un navegador estándar. Aunque lógicamente el equipo de realidad virtual será detectado, habrá que habilitar el modo VR en el botón pertinente de todas formas.

5.2.2 Funcionamiento

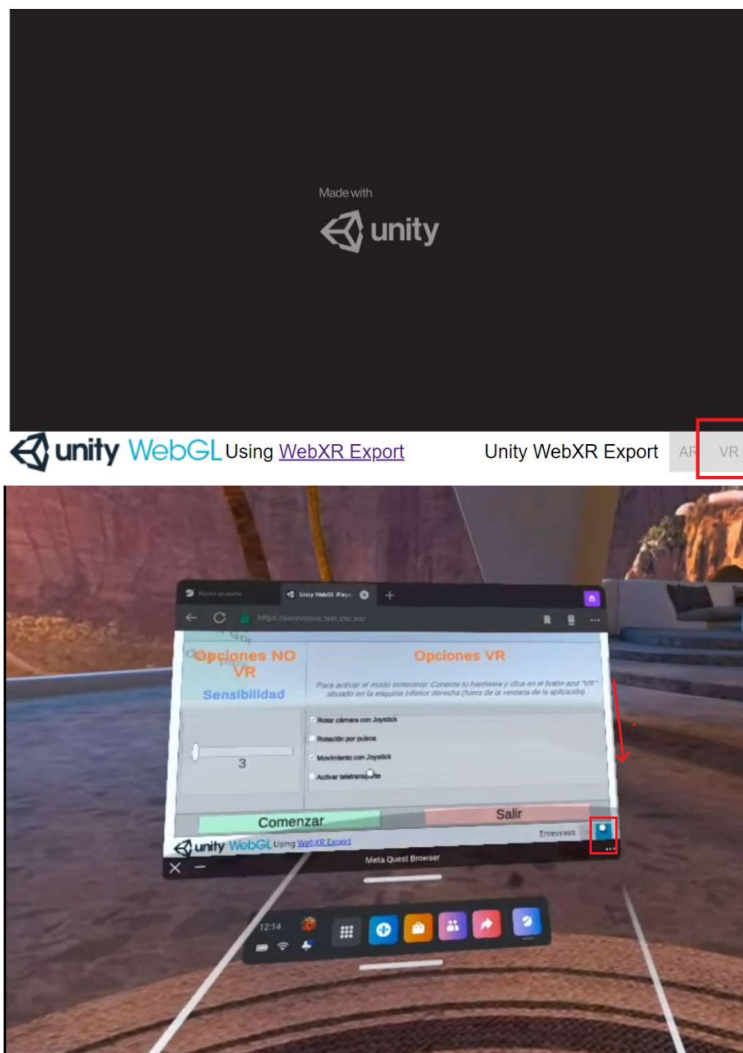


Ilustración 5.16. Acceso a la aplicación.

Una vez que el proyecto es exportado utilizando una de las plantillas de *WebXR*

y colocado los archivos resultantes en un servidor *Https* se permite el acceso desde Web.

Por defecto se accede a la aplicación sin experiencia inmersiva, pero cuando la aplicación obtiene permisos para acceder al hardware de realidad virtual (si se dispone de él conectado en el equipo o se accede desde el mismo), sucede lo siguiente:

1. Cuando se detecta el hardware de realidad virtual, se lanza un evento que actualiza las capacidades de la aplicación con relación a la realidad virtual: Si hay hardware, se habilita el botón **VR** (esquina inferior derecha) para llevar a cabo el cambio.
 - a. **Nota:** Para visualizar este botón desde el navegador del hardware de realidad virtual, es posible que se deba realizar un *scroll* de la ventana hacia abajo.
2. Cuando se pulsa el botón para el cambio, se lanza el evento que controla los cambios en XR, si el nuevo estado pasa a ser de realidad virtual se actualiza el estado de la aplicación:
 - a. Se habilita XR.
 - b. Se deshabilita el cursor y la interacción con el ratón.
 - c. Se habilita las “manos” del jugador y la interacción con ellas, además de las funciones de teletransporte.
 - d. La cámara pasará a moverse con el controlador o girando la cabeza, en vez de usando el ratón.
 - e. Se actualiza la altura del jugador y la posición de este.
3. Una vez que se está en modo inmersivo, las manos han aparecido y se pueden mover con los controladores. Se podrá interactuar con el entorno:
 - a. Cuando el *collider* de las manos entra en contacto con el collider de un objeto marcado como *Interactable* se podrá agarrar este

pulsando los botones del controlador: Lo que hará que el objeto se “anclé” a la mano del usuario y se mueva con él.

- b. La detección de objetos se realiza en tiempo real y se comprueba en el bucle *Update* de los controladores del usuario si se ha pulsado algún botón para llevar a cabo la interacción.

5.3 Acceso y repositorio

Este estudio inicial, como se ha comentado anteriormente, se ha realizado con el objetivo de realizar la investigación y el desarrollo de algunos componentes para el proyecto principal. Pero también se ha realizado para documentar el proceso de desarrollo y posterior exportación utilizando *Unity* y el paquete *Unity WebXR Export* para el departamento de Tecnologías inmersivas de la Fundación CTIC.

Así mismo, para facilitar el trabajo a futuros desarrolladores, investigadores o alumnos, todo el trabajo realizado en este estudio inicial se mantiene en un repositorio público al que se puede acceder en el siguiente enlace:

- https://github.com/Javiera195/Getting_Started_With_UnityWebXR [24]
- Y puede probarse aquí: <webxrunitytest.herokuapp.com>

6 Diseño del escenario

Se busca realizar un escenario compuesto de un total de cuatro escenas: Una principal y otras tres pequeñas con objetos interactivos. Se detallan a continuación:

Tabla 6.6. Descripción de salas del escenario.

Escena	Sala	Descripción
Principal	Entrada	Lugar donde aparece el usuario. En ella podrá visualizar los logos de las entidades del presente proyecto (EPI Gijón, perteneciente a la Universidad de Oviedo y la Fundación CTIC), así como un pequeño tutorial y el autor del trabajo.
	Hall	Siguiente sala a la entrada, será intermedia y contendrá algunos aspectos decorativos e interactivos, así como una descripción del proyecto y el contacto de diferentes entidades del panorama asturiano de la región
	Salón de eventos	La sala más grande de la escena se accede desde el <i>Hall</i> . Compuesta de un escenario al que se puede acceder y donde se podrá seleccionar que video de los disponibles reproducir y visualizar. Así mismo dispone de un cuadro de mando para modificar aspectos como el volumen.
	Salas laterales	Dos pequeñas salas que se encuentran a los laterales del <i>Hall</i> . En la sala de la izquierda se podrá acceder mediante escaleras a la sala de instrumentos y a la sala asturiana. En la sala de la derecha se podrá acceder a la sala de teoría y juego.
Sala de instrumentos	Se accede desde la escena principal, en esta sala se encontrarán diferentes instrumentos. Se puede visualizar el modelo 3D de estos, leer una breve descripción sobre ellos y escuchar pistas de audio grabadas utilizándolos.	
Sala Asturiana	Se accede desde la escena principal, en esta sala se busca promover la cultura asturiana. Podrán visualizarse modelos 3D sobre instrumentos asturianos, así como leer la historia de la cultura de la región y escuchar pistas de audios de, por ejemplo, bandas de gaitas.	
Sala de teoría & juego	Se accede desde la escena principal, en esta sala podrá estudiarse teoría musical básica: Escalas visualizando modelos de estas y los sonidos que transmiten. También se ofrece un pequeño juego al usuario (pulsar sobre modelos 3D que emiten, cada uno de ellos, una nota diferente).	

6.1 Boceto del escenario

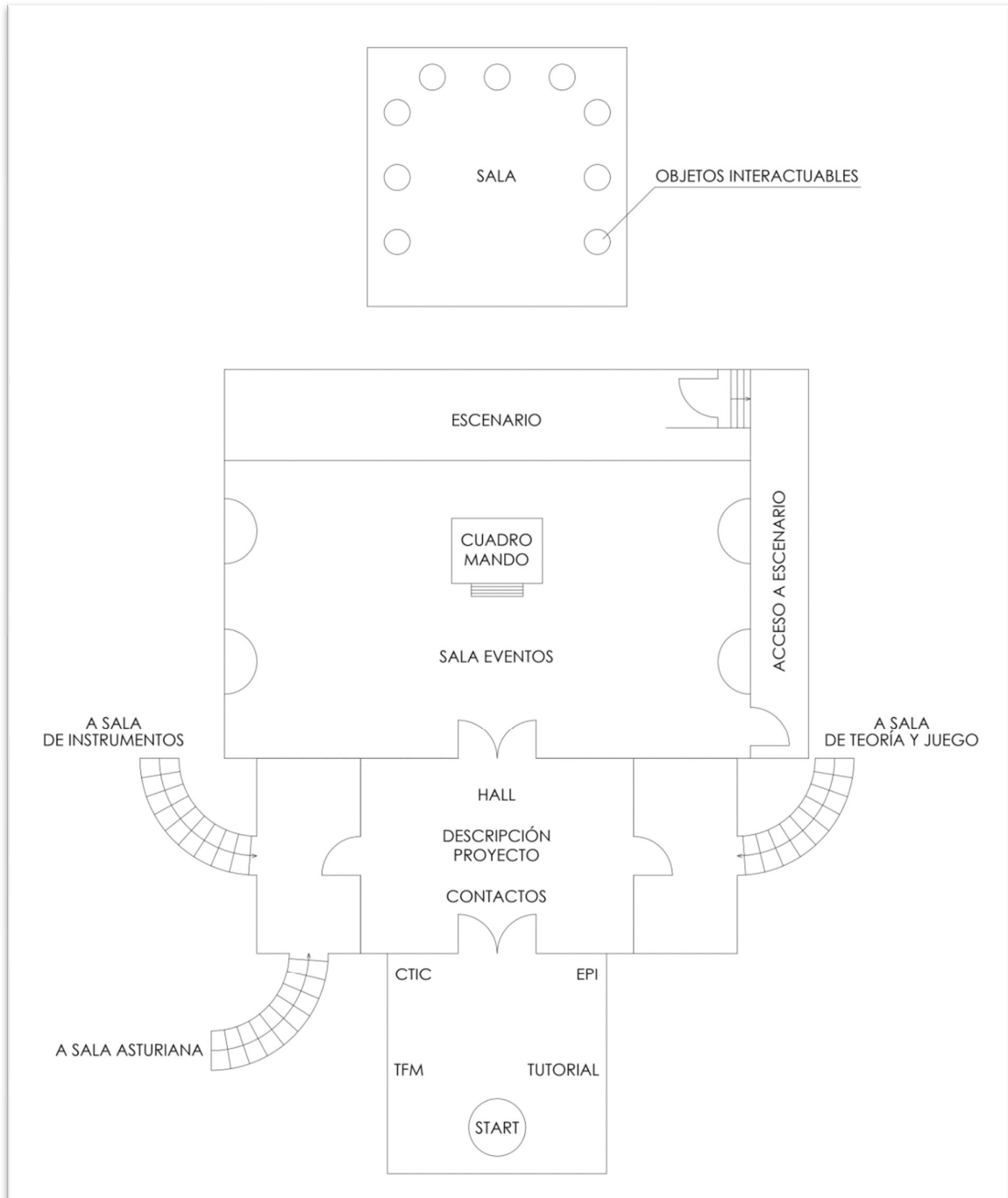


Ilustración 6.17. Plano del escenario.

6.2 Creación del escenario

6.2.1 Prototipado en Unity

En primer lugar, se diseña el escenario desde el motor de Unity utilizando figuras geométricas básicas: Cubos y planos. Se obtiene un resultado como el siguiente:

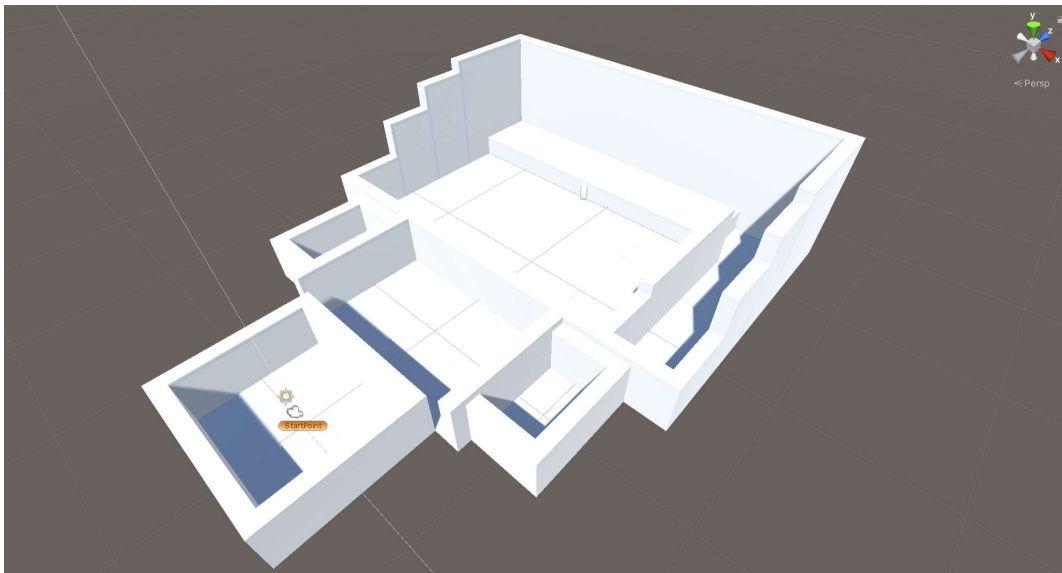


Ilustración 6.18. Prototipado del escenario principal en Unity.

Podemos visualizar en la captura anterior las diferentes salas que se mostraban en el plano de la escena principal (ocultando los tejados). Gracias a utilizar este flujo de trabajo se obtienen las medidas que tendrá cada uno de los muros y cada una de las salas.

6.2.2 Prototipado en Blender

Se procede a diseñar, de forma básica, el escenario en *Blender*. Se obtiene, ocultando los techos del escenario para visualizar mejor el resultado, algo parecido a lo siguiente:

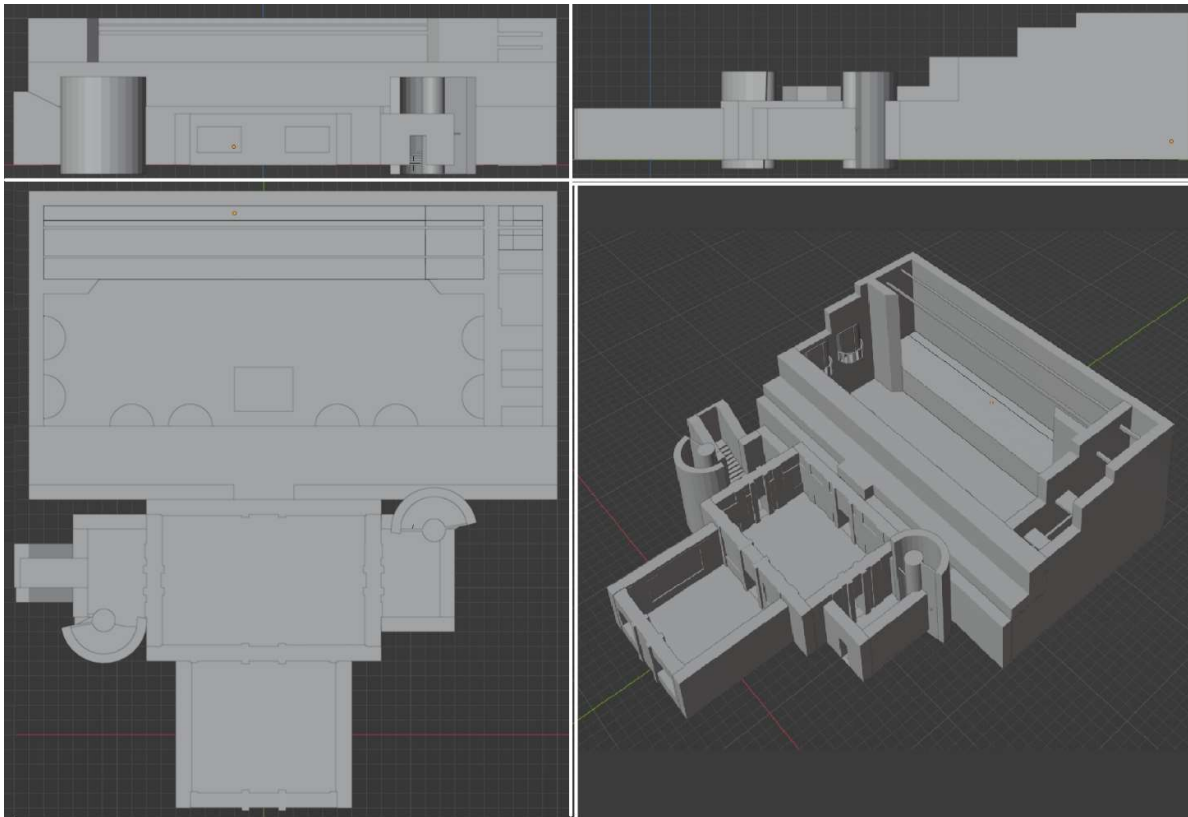


Ilustración 6.19. Malla básica del escenario principal en Blender.

6.2.3 Texturas en Blender

El siguiente paso será realizar una búsqueda de las texturas adecuadas para hacer el escenario algo más vistoso.

Las texturas son obtenidas de la página de recursos **PolyHaven** [25], más en concreto de su apartado de texturas. De esta forma se ha obtenido cada una de las imágenes que decoran el escenario: suelos, paredes y techos. Añadiendo a posteriori en el motor gráfico las imágenes que corresponden con las rugosidades, normales o reflexiones de las texturas utilizadas.

El resultado, una vez pintadas las texturas en *Blender* y colocadas algunas luces, es el siguiente:

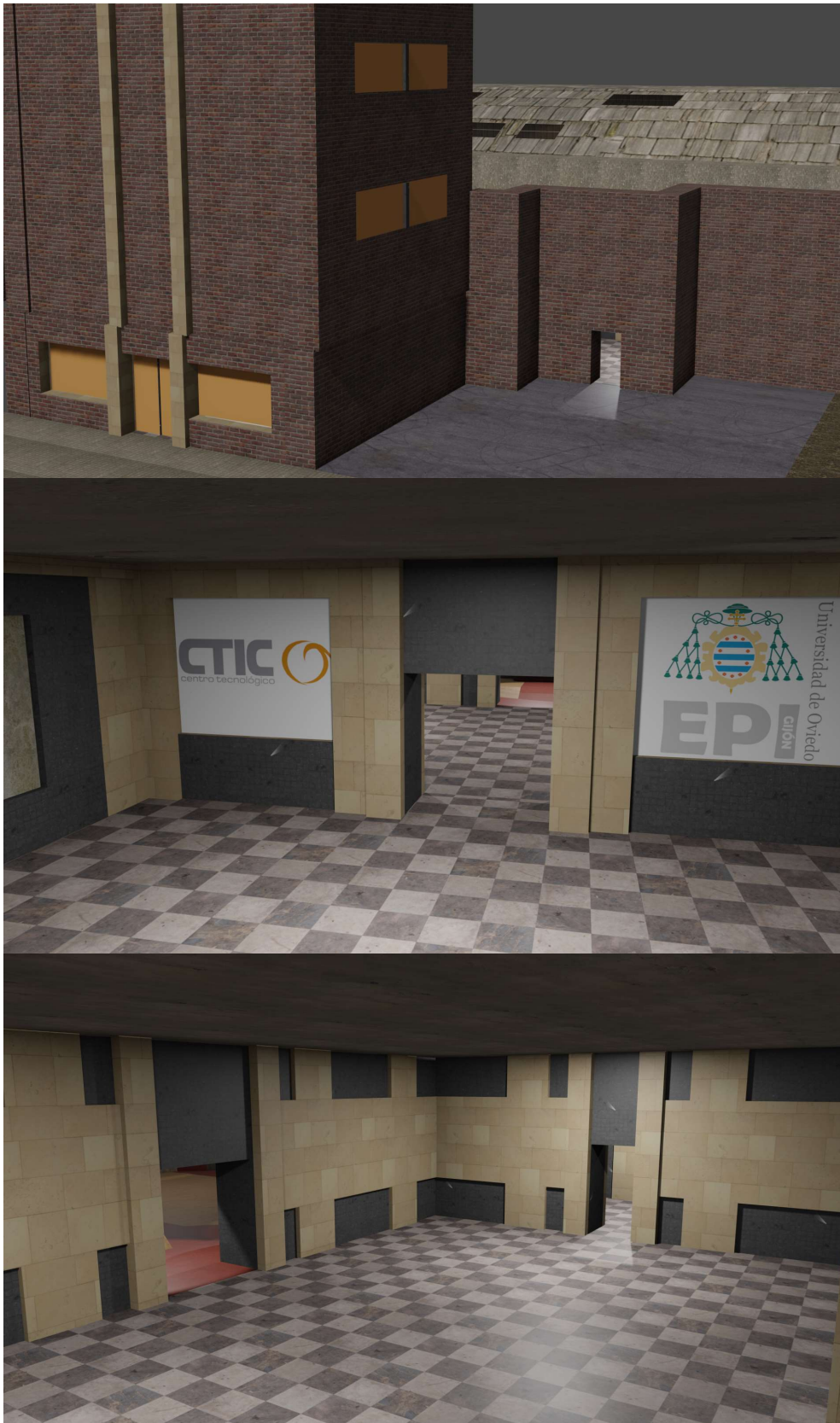






Ilustración 6.20. Escenario principal en Blender.

6.3 Salas secundarias

Las salas, quizá incorrectamente denominadas como *secundarias*, son los lugares donde el usuario podrá acceder y hacer uso del núcleo de la aplicación desarrollada: la interacción y la escucha de música.

Recordemos las salas, accesibles desde la sala principal, que se construirán:

- Sala de instrumentos: Para visualizar y escuchar diferentes instrumentos.

- Sala Asturiana: Para visualizar y escuchar diferentes aspectos de la cultura regional asturiana.
- Sala de teoría y juego: Para aprender aspectos sobre teórica musical y con un pequeño minijuego para divertir al usuario haciendo música.

Todas estas salas (a excepción del salón de eventos, integrado en la escena principal) iban a seguir un diseño similar: Una pequeña sala simétrica donde se encuentran los objetos con los que el usuario podrá interactuar (modelos, paneles con texto, audio, etc.). Pero finalmente se ha optado por modelar escenarios diferentes para cada una de ellas y así dotar de dinamismo a la aplicación.

6.3.1 Sala de instrumentos

La sala de instrumentos costa de un gran salón con diversas bifurcaciones donde el usuario podrá consultar instrumentos que se organizarán por su tipo (percusión, cuerda, viento, etc.), escuchar algunas pistas sobre ellos y conocer parte de su historia.

A continuación, el modelaje 3D en *Blender* de esta sala secundaria:





Ilustración 6.21. Sala secundaria de instrumentos en Blender.

6.3.2 Sala Asturiana

La sala secundaria asturiana busca representar la cultura de la región del Principado de Asturias.

Este escenario consta de un largo pasillo desde el que se accede a una gran sala con aspecto rustico con un pequeño puente en su centro. En esta sala podrá interactuarse con objetos que muestren descripciones de la cultura asturiana, así como

estudiar pistas de audios de instrumentos de la región.

A continuación, algunas instantáneas de la sala modelada en *Blender*:





Ilustración 6.22. Sala secundaria asturiana en Blender.

6.3.3 Sala de teoría y juego

Esta sala secundaria se subdivide en dos salas independientes: Una sala grande donde consultar, mediante objetos interactivos, algunos aspectos de teoría básica (escalas, organizadas por su tipo) y otra pequeña sala donde incluir un pequeño juego para el usuario (golpear algunos objetos que reproducen diferentes sonidos o notas, con los que se busca ofrecer al usuario un toque de diversión y creatividad).

A continuación, algunas capturas del modelado en *Blender*.

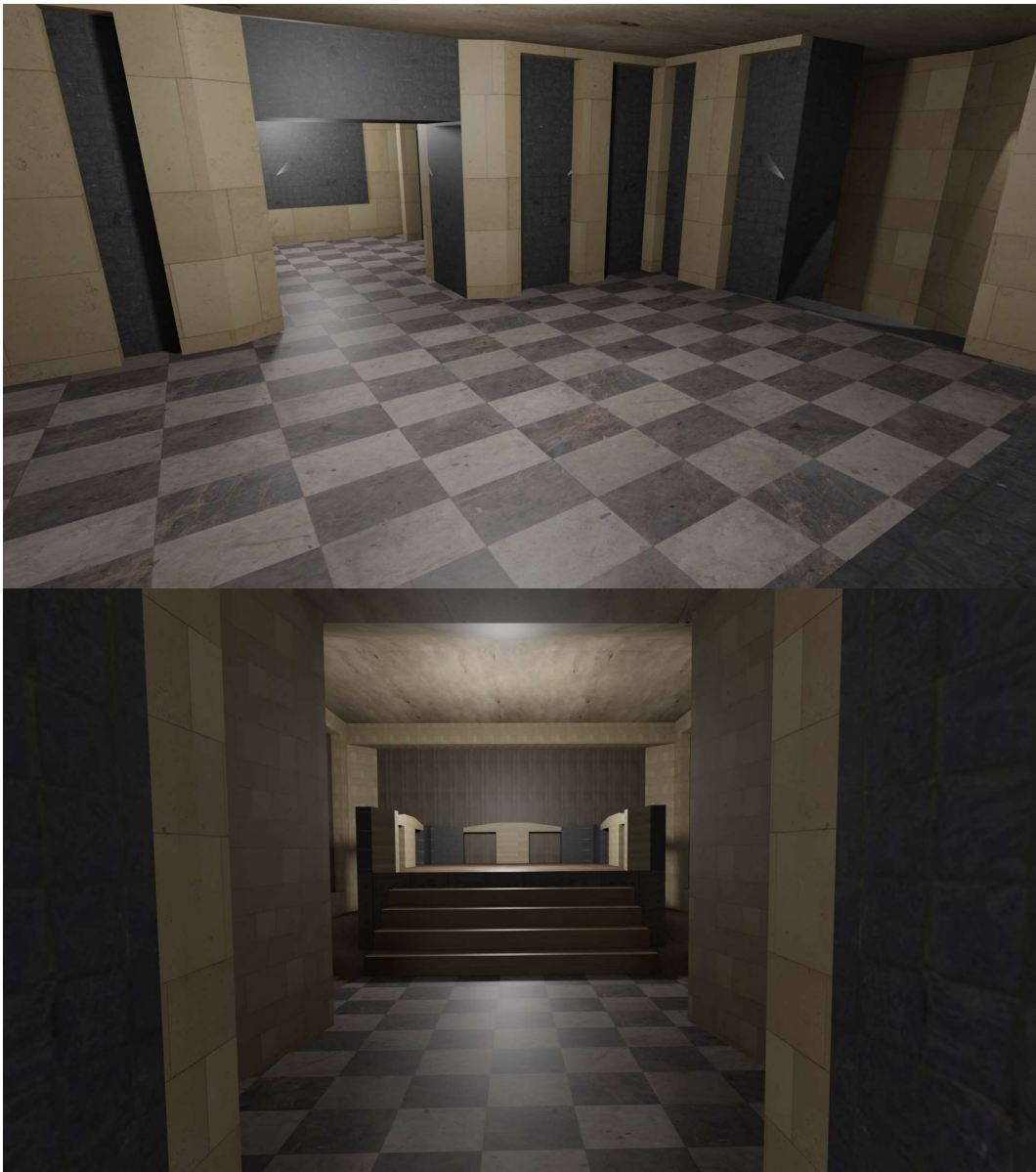






Ilustración 6.23. Sala secundaria de teoría y juego en Blender.

7 Implementación de la aplicación

Con el diseño del escenario completado y creado en el software de modelado, se procede a la creación de la aplicación web que se entregará como resultado del presente proyecto.

En este apartado se detallará el proceso de trabajo que se ha llevado a cabo en el motor gráfico *Unity*, prestando especial atención a los *scripts* escritos para dotar de contenido a la aplicación. También se hará hincapié en los problemas surgidos en el desarrollo y en el diseño de los diferentes objetos interactivos que el usuario se encontrará cuando acceda a la web.

7.1 Exportado del escenario a Unity

Una vez completados los escenarios en *Blender*, se procede al exportado de los modelos para su posterior uso en el proyecto. Los modelos son exportados en *.fbx* e importados en el proyecto de *Unity*.

El primer paso para comenzar a trabajar es el exportado del escenario al motor gráfico, añadiendo los modelos desarrollados anteriormente y añadiendo algunas luces propias del motor:

- *DirectionalLight*: Que funciona como el sol de la escena.
- *SpotLight* y *PointLight*: Para iluminar las salas interiores, todas serán *bakeadas* (pre-procesadas) para mejorar el rendimiento.

Así mismo se añaden los siguientes mapas a las texturas utilizadas, formando así los materiales de la escena:

- *Diffuse*: La ya utilizada en *Blender*, la foto estándar de la textura que da el color a esta.
- *Normal*: También conocida como *Bump*, indica como afecta la luz a la textura dando un aspecto 3D a una imagen en dos dimensiones.
- *Displacement*: Se añade en algunas texturas de superficies para añadir “relieve” y dar un aspecto más realista.
- *Reflection*: También conocida como *Specularity*, da un aspecto metálico a las texturas permitiendo que sea capaces de reflejar la luz que incide sobre ellas.

La malla (exportada desde *Blender*) junto con los materiales y un *Mesh Collider* (para evitar que el usuario traspase el escenario, generado por cada uno de los objetos que forman el escenario) son transformados en un *prefab* para su colocación en el motor.

También se añaden algunos otros *prefabs* (objetos del motor *Unity* previamente creados) de la *Asset store* para añadir puertas y las verjas de la zona exterior. La primera versión del escenario de la escena principal pasa a tener este aspecto.

7.1.1 Escena principal





Ilustración 7.24. Primera versión del escenario principal en Unity.

7.1.2 Salas secundarias

Además del escenario principal, también se exportan y se incluyen al proyecto las tres salas secundarias, a continuación, se muestran estas tres salas una vez importadas en Unity añadiendo las correspondientes luces del motor (en orden: sala de instrumentos, sala asturiana y sala de teoría y juego).

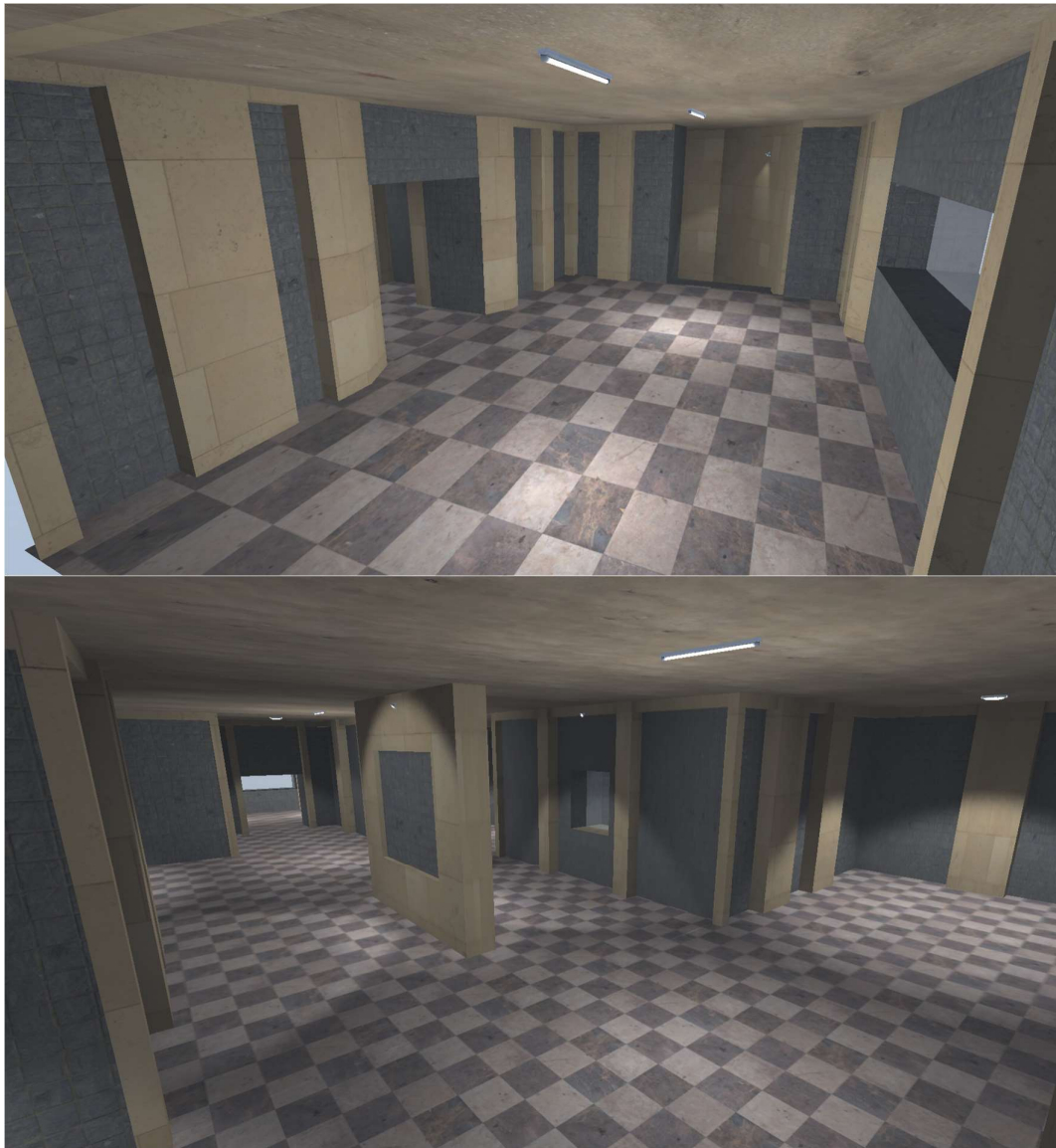






Ilustración 7.25. Salas secundarias en Unity.

7.2 Problemas de rendimiento iniciales

Las escenas cumplen con los objetivos visuales esperados y funcionan a la perfección sin hardware de realidad virtual. No obstante, una vez que se intenta reproducir el contenido en unas *Oculus Quest (V1 o V2)* nos encontramos con un muy bajo rendimiento y una tasa de imágenes por segundo demasiado baja, lo que imposibilita la movilidad causando, incluso, mareos y malestar.

Es necesario realizar algunas modificaciones en el escenario para intentar mejorar el rendimiento, se enumeran a continuación:

1. Se edita el escenario en *Blender* reduciendo el número de vértices y bordes, así como las caras que no son visibles (y por tanto no necesarias).
 - a. Había un gran número de bordes y vértices duplicados.
 - b. Se obtiene una mejora de rendimiento, pero aún no es posible la experiencia inmersiva.
2. Se simplifican algunas partes del escenario, reduciendo así aún más los bordes y vértices de la escena.
3. Se pre-procesa la luz ambiental (luz propia de la escena que proporciona el *skybox*) añadiéndola al conjunto de luces *bakeadas* y eliminando su procesamiento en tiempo real.
4. Se separa la sala *de eventos* de la sala principal obteniendo dos salas independientes.
5. Se modifican los parámetros de calidad en Unity para disminuir la calidad de las texturas.
6. Se reduce la distancia de procesamiento de la cámara y su ángulo de visión.
7. Se activa el *Occlusion Culling* (Eliminación selectiva) en Unity, esta característica permite desactivar el renderizado de objetos cuando estos no son visibles por la cámara debido a estar tapados por otros.

El rendimiento mejora notablemente y ya es posible una interacción más cómoda, aunque quedan pendientes tareas de optimización a la hora de realizar pruebas software sobre el proyecto.

7.3 Scripts

Para permitir que el usuario pueda realizar acciones en la aplicación es necesario el desarrollo de una serie de *scripts*. Estos *scripts* (escritos en C#) permiten al jugador moverse por el escenario, interactuar con el entorno o dotar de funcionalidad a los diferentes objetos del proyecto.

Es por esto último por lo que, realmente, cualquier funcionalidad o característica que se implemente en un objeto dentro de una aplicación 3D en Unity lleva consigo un *script* personalizado para hacerla funcionar.

Cabe destacar que se han implementado de tal manera que se permita la modificación de muchos de sus parámetros a través del editor de Unity (permitiendo modificar por completo muchos comportamientos haciéndolos altamente dinámicos). Además, los eventos que los objetos interactivos invocan se configuran también desde el editor del motor Unity: Esto hace que los *scripts* sean especialmente dinámicos, permitiendo utilizar el mismo para infinidad de objetos con funcionalidades diferentes entre sí.

En este apartado se indicarán los más importantes que, aunque han sido desarrollados para el proyecto, se han implementado y escrito de tal manera que se permita su reusabilidad (tanto por lo indicado anteriormente como por la forma de implementar el código) en otros proyectos diferentes desarrollados en el futuro tanto por el autor como por los trabajadores de La Fundación CTIC.

7.3.1 Controladores del usuario

Todos los *scripts* que permiten al usuario mover la cámara y moverse por el escenario se encuentran en el *prefab* *WebXRCameraSet*. Este objeto contiene en su interior las manos del usuario y todas las cámaras necesarias para una interacción tanto

en VR como no (un total de cinco cámaras).

Tabla 7.7. Scripts del usuario

Script	Descripción
WebXRCameraSet	
WebXRManager	Añadido directamente desde el paquete WebXR Exporter, gestiona los cambios a XR (VR).
PlayerController	<p>Añadido desde el paquete WebXR Exporter, pero altamente modificado para el presente proyecto.</p> <p>Permite el movimiento (translación y rotación) del usuario tanto en XR como en no XR por igual gracias al componente <i>Character Controller</i>. En este <i>script</i> se gestiona el cambio al modo inmersivo detectando el hardware de realidad virtual gracias a los eventos <i>onXRChange</i> y <i>onXRCapabilitiesUpdate</i>.</p> <p>Este script es el encargado de activar o no la interacción en modo NoXR, así como habilitar y deshabilitar las manos, el cursor y modificar aspectos como la posición y la altura del usuario.</p> <p>Tiene referencias al cursor (modo no inmersivo), a todos los aspectos que el usuario puede configurar (habilitar rotaciones, teletransporte o sensibilidad de mouse) además de a las cámaras y a las manos para el modo inmersivo.</p>
NonXRInteraction	<p>Permite la interacción con objetos cuando no se está en modo inmersivo.</p> <p>La interacción se realiza lanzando un <i>raycast</i> desde el centro de la pantalla (cámara) del usuario, si se detecta un objeto con la etiqueta <i>Interactable</i> (por ejemplo) y el usuario clicca sobre él se comienza la interacción: El objeto se mantiene en el centro de la pantalla mientras se mantenga el clic pulsado, al soltarlo el objeto se dejará caer.</p> <p>También se permite lanzar (impulsar) algunos objetos.</p>
XRTeleporterController	Si el usuario está en modo inmersivo y ha habilitado el teletransporte, este script detecta cuando el usuario clicca y suelta el botón de teletransporte para llamar a las funciones de este.
FootStepsSound	Reproduce sonido de pasos al desplazarse desde una fuente de audio situada en los pies.

Script	Descripción
HandL y HandR	
WebXRController	<p>Añadido directamente desde el paquete WebXR Exporter, identifica si se trata de la mano derecha o izquierda y permite detectar la interacción del usuario con los controladores (que botón pulsa o cuanto mueve los <i>joysticks</i>).</p> <p>El cambio de posición de la mano (translación) para moverlo junto a los controladores se consigue gracias al componente <i>Tracked Pose Driver</i>.</p>
CloseHands	<p>Añade una pequeña animación para hacer el movimiento de cerrar el mano más fluido (la animación se lleva a cabo a medida que el usuario pulsa el botón de interacción).</p>
WebXRInputManager	<p>Importado directamente desde el paquete WebXR Exporter. Controla los estados de los botones de los controladores en tiempo real (cuando se pulsan y se sueltan).</p> <p>Modificado para controlar el estado de más botones que los iniciales (Añadidos botones A-B y X-Y).</p>
ControllerInteraction	<p>Importado del paquete WebXR, pero altamente modificado.</p> <p>Permite la interacción con objetos en modo inmersivo. Cuando el <i>collider</i> de la mano entra en contacto con un objeto con la etiqueta <i>Interactable</i> (por ejemplo) este se añade a la lista de objetos con los que se ha colisionado (si se aleja de él, el objeto se retira de la lista).</p> <p>Cuando el usuario clicca el botón de interacción del controlador correspondiente a esa mano, se interactúa con el objeto de la lista más cercano: Se crea un <i>joint</i> entre ambos objetos (una unión) para moverlos juntos mientras el usuario mantenga pulsado el botón, cuando se suelte se soltará el objeto.</p> <p>También se permite lanzar (impulsar) algunos objetos. Si se mantiene la mano cerrada (botón pulsado) se permite golpear objetos.</p>
Cameras	
WebXRCamera	<p>Importado del paquete WebXR Exporter. Gestiona las cámaras del usuario, en total son 5:</p> <ul style="list-style-type: none"> • CameraMain: Para el modo no inmersivo. • CameraL y CameraR: Para el modo en realidad virtual. • CameraARL y CameraARR: Para el modo en realidad aumentada (fuera del enfoque de este proyecto y por tanto no utilizado).

7.3.2 Scripts para la interacción

Cómo se comentaba anteriormente, cualquier aspecto de funcionalidad que busque implementar el proyecto pasa por el desarrollo de un Script, es por ello por lo que se han escrito varios que han ido colocándose en diferentes objetos dentro de las escenas, estos son algunos de los ejemplos más importantes:

Tabla 7.8. Scripts para la interacción

Script	Descripción
MenuManager	
MenuBehaviour	<p>Hijo del objeto WebXRManager (que al igual que el jugador, tampoco se destruye entre escenas), el objeto <i>MenuManager</i> contiene el script <i>MenuBehaviour</i>.</p> <p>Este script es el encargado de gestionar el menú del usuario, tanto en modo inmersivo como no. Se comunica con el <i>PlayerController</i> para modificar todas las opciones que el jugador seleccione.</p> <p>Cuando el usuario se encuentra en modo inmersivo, el menú (objeto VRMenu, formado por varios botones) aparece en la mano izquierda del jugador (trasladándolo y rotándolo en cada <i>fotograma</i>). Cuando el jugador está en modo no inmersivo, el menú se forma con un <i>Canvas</i> que ocupa toda la pantalla y con el que se interacciona mediante el uso del ratón.</p>

Script	Descripción
Objetos interactivos	
SpecialInteractable	<p>Clase abstracta de la que heredarán Interfaz todos los objetos interactivos especiales (que tienen alguna funcionalidad más que no sea agarrarlo, soltarlo o lanzarlo). Esta clase implementa la interfaz <i>ISpecialInteractable</i> que detalla los métodos principales de la clase.</p> <p>Por tanto, implementa tres métodos abstractos que deben programarse para cada tipo de objeto interactivo: <i>Drop</i>, <i>Grab</i> y <i>Throw</i>. Cuando el usuario agarra un nuevo objeto se comprueba (dentro de los scripts de interacción del usuario) si se trata de un objeto especial, de ser así se llamarán a estas tres funciones cuando el usuario agarre, suelte o lance el objeto.</p> <p>También añade, de forma opcional, la propiedad de <i>Break Interaction</i>: Esto permite que la interacción entre el objeto y el usuario se rompa cuando el objeto se encuentre a una determinada distancia del usuario o de una referencia escogida (otro objeto).</p> <p>Por último, permite modificar aspectos como: Booleano que cancela la interacción con el objeto, si se permite su movimiento, si se permite un clic permanente (mantener pulsado), si se permite desplazarlo o si se permite su lanzamiento.</p>
Ejemplos de objetos interactivos (Todos implementan <i>SpecialInteractable</i>)	
VRPhysicalButton	<p>Permite activar el botón cuando este se pulsa físicamente en modo inmersivo o cuando se hace clic con el ratón en modo no inmersivo.</p> <p>Activa un evento al activar el botón y otro al desactivarlo.</p>
VRGrabbableDoor	<p>Permite agarrar el poyete de una puerta para abrir esta (tanto con la mano como con el ratón).</p>
VRPhysicalLever	<p>Permite añadir funcionalidad una palanca clásica. La base es estática moviéndose únicamente la palanca. Se activa un evento cuando la palanca entra en el collider <i>Off</i> (izquierda) y cuando la palanca entre en el collider <i>On</i> (derecha).</p>
VRPhysical Potentiometer	<p>Añade la funcionalidad de potenciómetro a un objeto (rosca giratoria que cambia un valor), puede cambiarse el valor tanto con la mano como con el ratón.</p> <p>El valor es calculado usando el movimiento del eje X cuando se está en modo no inmersivo y usando la diferencia de distancia entre el objeto y la mano en el eje seleccionado.</p> <p>Se llama a un evento cuando el valor cambia.</p>

Script	Descripción
VRPhysicalSlider	<p>Implementa un control deslizante para cambiar un valor entre un límite inferior y uno superior. Una vez más se utiliza el eje X en modo no inmersivo y la distancia entre el objeto y la mano en el eje seleccionado en modo inmersivo.</p> <p>Se llama a un evento cuando el valor cambia.</p>
VRPhysicalToggle	<p>Con cierta similitud al botón físico, pero activándose cuando el usuario “agarra” el objeto (clic o selección con la mano).</p> <p>Tiene el estado activado y desactivado, llamándose a un evento cada vez que el valor del estado cambia.</p>
InteractableTextPanel	<p>Cuando el usuario interactúa con este objeto deja mostrar un panel descriptivo sobre el modelo utilizando una animación.</p> <p>El panel puede consistir en un objeto personalizado o en uno o varios <i>canvas</i> con texto. Es posible modificar el texto de todos los <i>canvas</i> indicados como hijos del objeto que lleve este <i>script</i> modificando una lista de textos.</p> <p>El panel mostrado mira en todo momento (rota en el eje y) al jugador para facilitar la lectura del contenido.</p>
InteractableTextSoundPanel	<p>Hereda de <i>InteractableTextPanel</i>, añade el requerimiento de hacer uso de un <i>AudioSource</i> en el objeto que implemente el script y además implementa los métodos que permiten modificar el clip de audio de este y los métodos para reproducirlo.</p>
ObjectToHandPosition	<p>Normalmente cuando un objeto interactivo se agarra, este pasa a tener una relación que lo une a la mano del jugador para desplazarlo, pero sin hacer ningún cambio en rotación. Este Script permite que el objeto se recoloca a la posición y rotación exacta de la mano del jugador, además implementa el reiniciar la posición inicial de este si se pulsa el botón de lanzar.</p>
Objetos Multimedia (Audio/Video/Foto)	
TapeBehaviour	<p>Hereda de <i>SpecialInteractable</i>, es un <i>script</i> indicado para las <i>cintas</i> que se encontrarán por el escenario.</p> <p>El <i>script</i> permite la colocación automática (romper la interacción y congelar su posición y rotación hasta que el usuario las vuelva a agarrar) de estas cintas en lectores que permitan modificar el comportamiento de otros objetos. Además, se permite mostrar un título de la cinta en la malla de este y una descripción del contenido de la cinta cuando el usuario la mantiene en sus manos y clicla el botón de lanzar.</p>

Script	Descripción
TapeLectorBehaviour	<p>Implementa los métodos necesarios para los lectores de cintas. Estos se deben llamar en los eventos <i>onTriggerEnter</i> y <i>onTriggerExit</i> del objeto donde se vaya a colocar la cinta.</p> <p>El usuario indica en el editor que tipo de cinta se leerá y los métodos comprueban si la cinta introducida es correcta, si lo es se llama a su método de colocación automática y se retorna <i>true</i>. Lo mismo ocurre con el evento al desconectar la cinta, se comprueba que la cinta que sale del <i>collider</i> es la ya introducida, si es así se llama a su método de desconexión.</p>
VideoTapeBehaviour	<p>Hereda de <i>TapeBehaviour</i> y permite, en esta primera versión, indicar una Url de la que se obtendrá el vídeo</p>
VideoTapeGenerator	<p>Este script permite añadir una posición (si no se indica se utilizará la que tenga este script adjunto), un eje de coordenadas y una separación, además del <i>prefab</i> de VideoTape que se instanciará utilizando los parámetros anteriores.</p> <p>Contiene el método abstracto que deben implementar las clases que hereden de ella y que tiene como objetivo inicializar la lista de <i>VideoTapes</i> a inicializar, también contiene el método que instanciará dicha lista utilizando la posición, eje y separación indicados.</p> <p>Gracias a este script se añade de dinamismo a la sala de eventos, permitiendo modificar los videos que pueden visualizarse sin necesidad de realizar modificaciones en el proyecto de Unity, realizar un nuevo Build ni desplegar nuevos cambios (estas se generan cada vez que el usuario accede a la sala).</p>
Http_VideoTapeGenerator	<p>Hereda de VideoTapeGenerator.</p> <p>El objetivo es realizar una petición http/s a una url específica donde se espera encontrar un archivo JSON con el listado de videos y las url de estos. Una vez leído el JSON, este se parsea en un listado de <i>OnlineVideoTapes</i> (objeto donde se configura si el video es local u online, indicando el nombre del fichero o la url de este).</p> <p>La petición se realiza mediante el uso de <i>Corutinas</i>, por lo que el resto de las ejecuciones pueden continuar mientras esta se lleva a cabo. Una vez finalizada se llama a la función pasada como parámetro a modo de <i>callback</i>.</p> <p>Si el listado es obtenido con éxito, este es enviado a la función que se encuentra en la clase padre para instanciar las cintas.</p> <p>Este script se encuentra en desuso a favor del <i>MinIO_VideoTapeGenerator</i> que se detalla a continuación.</p>

Script	Descripción
<p>MinIO_ VideoTapeGenerator</p>	<p>Hereda de VideoTapeGenerator y tiene dos modos de funcionamiento diferentes que se escogen automáticamente dependiendo de la plataforma de ejecución:</p> <ol style="list-style-type: none"> 1. Si se está ejecutando desde el propio editor de Unity, se inicializa un cliente MinIO interno (obtenido a través del gestor de paquetes <i>Nuget</i> para <i>C#</i>). Este cliente comprueba la conexión y la existencia del <i>bucket</i> de videos, si no existen errores se obtiene el listado de objetos de dicho <i>bucket</i> en forma de listado de <i>Items</i> que se convierten en <i>VideoTapes</i>. 2. Si se está ejecutando en WebGL una vez compilado, no se inicializa ni se usa ningún cliente de <i>MinIO</i>, si no que se lanza una petición de tipo <i>GET</i> a la <i>API</i> de un cliente externo de <i>MinIO</i> desarrollado para el presente proyecto (más adelante se detallará en el apartado de arquitectura). La petición (utilizando <i>Corutinas</i>) obtiene un <i>Json</i> con los datos de los videos almacenados en el <i>Bucket</i> de videos, este <i>Json</i> es transformado en una lista de <i>VideoTapes</i>. <p>Independientemente del camino seguido, la lista de <i>VideoTapes</i> es enviada a la función de instanciación del padre donde se generarán las <i>VideoTapes</i> que el usuario podrá visualizar.</p>

Script	Descripción
<p>VideoTapeGenerator SourceSettings</p>	<p>Este script se coloca a un objeto junto con <i>Http_VideoTapeGenerator</i> y <i>MinIO_VideoTapeGenerator</i> (si estos dos últimos no están en el objeto se adjuntan automáticamente).</p> <p>Este script realiza una petición http de lectura sobre un fichero de configuración <i>json</i> del directorio <i>StreamingAssets</i> del proyecto (esto permite modificar el fichero incluso después de compilar), se debe realizar la lectura con la petición y no con IO estándar debido a que <i>WebGL</i> no soporta este tipo de operaciones.</p> <p>Una vez obtenida la información del fichero de configuración se procede a inicializar la configuración del generador de <i>VideoTapes</i> utilizando dos fuentes:</p> <ol style="list-style-type: none"> 1. Variables de entorno: Cada parámetro de la configuración se almacena en una variable de entorno diferente. Estas variables de entorno se definen a la hora de desplegar los servicios utilizando tecnología <i>Docker</i> (se detallará en el apartado de Arquitectura). 2. Fichero de configuración: El fichero leído anteriormente, si no se es capaz de leer alguna variable de entorno, estas son erróneas o no existen se procede a aplicar la configuración del fichero <i>json</i>. <p>Las variables que indican la configuración del generador de <i>VideoTapes</i> son:</p> <ul style="list-style-type: none"> • Uso de MinIO (deshabilita o habilita el generador MinIO y el de Http según el valor). • Si se usa MinIO: Host, puerto, <i>access</i> y <i>secret</i> key, además del uso de SSL y bucket de MinIO. • Si se usa el cliente externo de MinIO: host y puerto de su dirección y si se usa o no SSL. • Si se usa el generador http: Host, puerto y nombre del fichero <i>json</i> de videos.

Script	Descripción
<p>VideoPlayer Behaviour</p>	<p>Hereda de <i>TapeLectorBehaviour</i>, el objeto deberá contener un componente de tipo <i>Collider</i> que será donde se coloque la cinta.</p> <p>Permite introducir un objeto con el script <i>VideoTapeBehaviour</i> en su interior, comprobando previamente si la cinta introducida es correcta con los métodos implementados en <i>TapeLector</i>.</p> <p>Cuando se pulsa el botón de <i>play</i> (indicado desde el editor) se reproduce el contenido del vídeo configurado en la cinta de vídeo en un objeto en específico utilizando una <i>RenderTexture</i>. Si no se indica un botón para <i>play</i> el vídeo se reproduce automáticamente y se pausa al desconectar la cinta.</p> <p>Se permite indicar una fuente de audio externa.</p>
<p>ScreenVideo Behaviour</p>	<p>Se trata del script que se coloca al objeto que reproduce vídeo. Permite la configuración de este indicando una Url o un clip de vídeo, así como configurando si usa una fuente de audio externa (objeto de donde saldrá el sonido), si reproducirá un clip externo de audio, etc.</p> <p>Implementa los métodos que permiten configurar automáticamente el audio, el cambio de Clip de vídeo, el cambio de volumen o los métodos que implementan el <i>Play/Stop</i> del contenido.</p> <p>Al utilizar <i>WebGL</i> la reproducción de vídeo es un tanto problemática, por eso se implementan métodos específicos para la configuración del reproductor.</p>
<p>MusicalObject Behaviour</p>	<p>Se permite atribuir un texto que se muestra sobre el objeto y un archivo de sonido que se reproduce al interaccionar con el objeto o cuando otro objeto interactivo colisiona con este.</p> <p>También modifica el color del objeto durante un <i>fotograma</i> cuando reproduce el sonido.</p>
<p>MusicTheoryTape Behaviour</p>	<p>Hereda de <i>TapeBehaviour</i>, permite indicar un <i>array</i> de 6 enteros que indican los saltos que se producen en las escalas musicales. Siendo el valor 1 un semitono y el valor 2 un tono (se indican 6 saltos y no 7 debido a que el primer salto es 0 para indicar la raíz de la escala).</p> <p>Por ejemplo (siendo ST semitono y T tono), la escala mayor natural es <i>T-T-ST-T-T-ST</i>, por tanto, los saltos indicados en la cinta de esta escala será el array [2,2,1,2,2,1].</p>

Script	Descripción
MusicTheoryLector	<p>Hereda de <i>TapeLectorBehaviour</i>, el objeto deberá contener un componente de tipo <i>Collider</i> que será donde se coloque la cinta.</p> <p>Este <i>script</i> también recibe como parámetro un color y una lista de 12 <i>MusicalObjects</i> (12 notas ordenadas de <i>Do</i> a <i>Si</i>), cuando una cinta es introducida en el lector (comprobando previamente si la cinta introducida es correcta con los métodos implementados en <i>TapeLector</i>) se modifica el color de los <i>MusicalObject</i> que pertenezcan a la escala introducida.</p> <p>Se permite también modificar la tonalidad (modificando la nota raíz de la escala), de tal forma que se modifican los objetos a resaltar permitiendo, por ejemplo, iluminar tanto la escala mayor natural de <i>Do</i>, como la de <i>Mi</i> o <i>Re Bemol</i>.</p>
PitchCorrector Behaviour	Permite modificar el <i>Pitch</i> del audio de una lista de objetos con el <i>script</i> <i>MusicalObject</i> .
InfoScreen Behaviour	<p><i>Script</i> diseñado para implementar una pantalla donde se permite pasar de página mostrando diferentes objetos. Recibe como parámetro en el editor un objeto cuyos hijos serán el contenido al mostrar.</p> <p>El cambio de página se realiza mediante animaciones y permite tanto avanzar como retroceder de página. También se actualiza un contador que se muestra como <i>página actual / máximo de páginas</i>.</p>

7.3.3 Otros scripts

Existen algunos pocos scripts adicionales que añaden funcionalidades que nada tienen que ver con el usuario o con la interacción.

Tabla 7.9. Otros scripts.

Script	Descripción
GameManager	<p>El <i>script</i> más importante en el desarrollo de un videojuego en Unity (incluso tiene el icono del archivo varía su icono cuando tiene este nombre).</p> <p>Permite el cambio entre escenas, el almacenamiento de variables globales y compartidas entre varios scripts o la llamada de funciones comunes.</p> <p>Implementa el patrón <i>Singleton</i>.</p>

Script	Descripción
ColliderTrigger	Se llama a un a evento cuando un objeto en específico entra en contacto con el objeto de que contiene este script.
DontDestroy	Permite al objeto no destruirse cuando carga una nueva escena, de esta forma se mantiene los valores en todo momento y se permite una interacción más fluida.
ChangeInteractionColor	El objeto varía de color cuando el usuario pasa su mano sobre él.
CustomEvents	Una lista de eventos personalizados fuertemente tipados para utilizar en este proyecto.
Utils	Clase estática no atribuida a ningún objeto con funciones que pueden llamarse desde todos los scripts: Lectura de variables de entorno, detección de objetos cercanos, etc.
ConditionalHideAttribute	Permite ocultar algunos atributos en el editor de Unity para algunos scripts cuando se activan o se desactivan otros atributos del mismo script.
ConditionalHidePropertyDrawer	
SceneTeleporter	Cuando el usuario entra en contacto con este objeto se le teletransporta a otra escena, en la ubicación de un punto de encuentro en concreto.
CollisionSound	Permite reproducir un audio concreto cuando un objeto colisiona con otro (se pueden excluir capas).
AutomaticallyPlacing	Teniendo en cuenta una posición inicial y un eje de coordenadas, ordena una lista de objetos de tal forma que exista entre todos los objetos la separación indicada por el usuario en el editor. Puede instanciar una lista de objetos personalizada o recolocar todos los hijos (de primer nivel en la jerarquía) del <i>gameobject</i> que lleve el <i>script</i> .
ColliderTrigger	Se añade a un objeto con <i>Collider</i> y permite indicarle cuatro tipos de eventos: <i>OnEnter</i> , <i>OnExit</i> , <i>Stay</i> y <i>StayOut</i> . También permite configurarse para que solo se detecten las colisiones con una etiqueta específica. El objeto con el que se colisiona es pasado como parámetro en el evento llamado.

Script	Descripción
BasketCase	Se permite anotar puntos lanzado objetos interactivos a una canasta. Los objetos que se acerquen al objetivo serán contabilizados como lanzados y los que acierten puntuarán (puede configurarse para contabilizar únicamente los objetos que tengan velocidad negativa en el eje y, es decir, que “entren por arriba”).
KillzoneBehaviour	Nombre común en el desarrollo de videojuegos, cuando los objetos entran en contacto con esta zona son destruidos. Si es el usuario (o un objeto que implementa <i>SpecialInteractable</i>) se le teletransporta al punto donde se encontraba al inicio de la escena. Normalmente es una amplia zona bajo el escenario para evitar que los objetos caigan indefinidamente mientras siguen ocupando espacio en memoria.
AppearanceBehaviour	Permite modificar el estado del objeto (activo o no) haciendo uso de una animación. Esto hace uso de un <i>Animator</i> junto con dos animaciones (<i>appear</i> y <i>disappear</i>) que se llaman con la variable booleana <i>isActivated</i> . Sirve para cualquier objeto que aparezca y desaparezca de forma animada.
ObjectGenerator	Instancia un prefab concreto en una posición concreta. Puede aplicar una fuerza al instanciarlo para hacer el efecto de propulsión.

7.3.4 Cliente MinIO externo

No se trata de un *Script* a lo sumo, pero es parte del código del despliegue a entregar en el presente trabajo.

En el caso de hacer uso de *MinIO* (*MinIO_VideoTapeGenerator*) una vez el programa es compilado nos encontramos con grandes dificultades y problemas: Las operaciones asíncronas que hacen uso de tareas o hilos NO están soportadas por WebGL, las funciones del cliente de MinIO en C# hacen uso frecuente de estas características, lo que provoca enormes errores a la hora de utilizar la aplicación desde el navegador una vez compilada y desplegada.

Se debe tener en cuenta que este cliente externo **no** es utilizado cuando se lanza la aplicación desde el editor de Unity.

Tras horas de investigaciones y desarrollo se lleva a cabo que la solución pasa por trasladar la ejecución de las consultas a un servicio externo, es por ello por lo que se desarrolla un servicio JavaScript independiente que tiene el siguiente flujo de trabajo:

- Se mantiene escuchando en un host y puerto definido a recibir peticiones.
- Se inicializa un cliente de MinIO que mantiene una conexión con el host y puerto de dicho servicio.
- Cuando se recibe una petición, el cliente de *MinIO* realiza la consulta asíncrona pertinente. Las operaciones disponibles son:
 - Listado de *Buckets* (cada bucket contiene un grupo de objetos).
 - Comprobación de existencia de *Buckets*.
 - Listado de objetos en un *Bucket* (en este contexto: videos).
- El resultado de la consulta es enviado al usuario que la realizó en formato *Json*.

La configuración del puerto en el que escucha este servicio (además de la dirección del servicio MinIO y las claves de acceso) se define en las variables de entorno del despliegue o, si estas no se existen, se leen desde un fichero de configuración: Esta configuración debe coincidir con la del generador de video tapes (por ello el uso de variables de entorno es altamente recomendable).

Este servicio se construye de forma sencilla y rápida gracias al uso de *NodeJS* (entorno de desarrollo asíncrono *JavaScript* para la capa de servidor) y su gestor de paquetes *NPM* desde donde se importa:

- *Express*: Permite levantar una infraestructura web rápida, minimalista y flexible para NodeJS. En resumen, levantamos un servidor con sus rutas

en cuestión de segundos.

- *MinIO*: Cliente que permite conectarse a una base de datos *MinIO* y realizar diferentes operaciones.

Este servicio implementará *https* con certificados auto firmados para evitar la aparición de errores durante la ejecución.

7.4 Contenido interactivo

7.4.1 Menú

Aunque no forma parte del diseño del escenario es necesario comentar acerca de la escena del menú, pues el lugar donde el usuario comienza en la aplicación.

En esta escena se inicializa todo lo necesario para la interacción con el usuario: Cámaras, jugador, manos, menú, etc. Estos objetos que se inicializan disponen del script *DontDestroy* por lo que se mantienen en la jerarquía de objetos, aunque se cambie a otra escena diferente.

En esta escena hay dos caminos posibles:



Ilustración 7.26. Escena del menú modo no XR

1. El usuario se encuentra en modo no inmersivo, visualizará el menú de la aplicación donde podrá modificar los parámetros (tanto XR como no XR) y donde se le indicará como activar el modo inmersivo si dispone de hardware.

El usuario no podrá desplazarse (de hecho, el tiempo se mantiene congelado) y solo podrá interactuar con el menú a base de clics de ratón.

Una vez que clique en el botón de comenzar, será tele portado a la escena principal, a partir de este momento puede visualizar el menú en todo momento con la tecla *shift*.

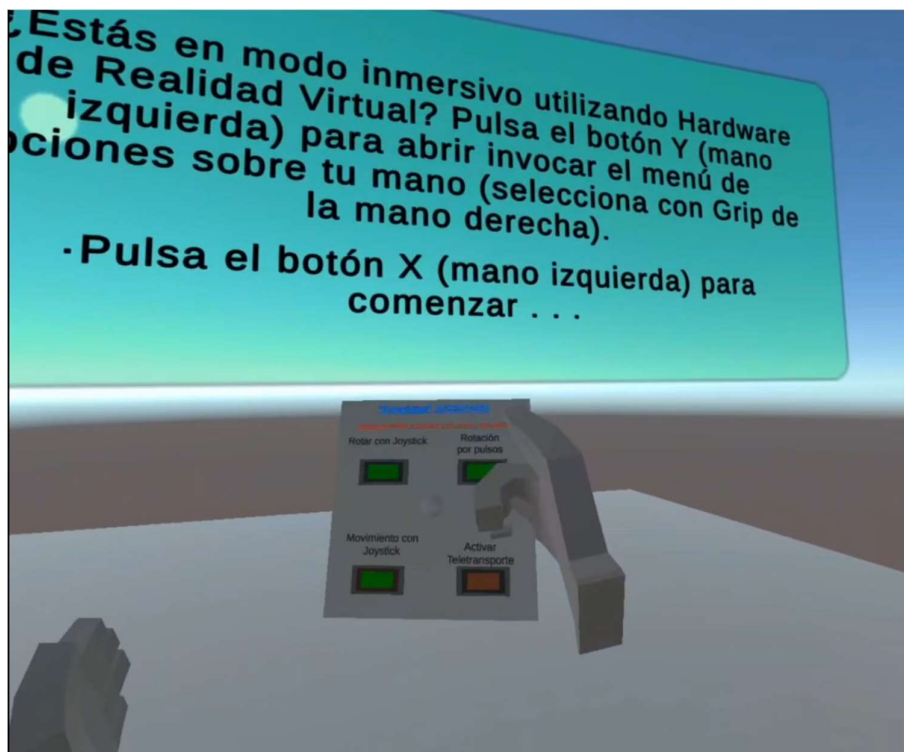


Ilustración 7.27. Escena del menú en modo XR

2. El usuario se encuentra en modo inmersivo. Podrá desplazarse por un plano reducido (si se precipita al vacío será tele portado al lugar de origen).

Un cartel a su izquierda le indicará como consultar el menú y comenzar la aplicación. Podrá consultar el menú en su mano izquierda manteniendo clicado el botón Y de dicho controlador y modificar las opciones con su mano derecha.

Lo único que diferencia la interacción en este aspecto junto con el resto de las escenas es que, al pulsar el botón X de su mano izquierda, el usuario será tele portado a la escena principal.

7.4.2 Sala principal

La sala principal es la escena donde el usuario aparece una vez inicial aplicación superando el menú principal.

Tal y como se preveía en el diseño del escenario, en esta escena nos encontramos diversas salas con diversas funcionalidades.



Ilustración 7.28. Sala de entrada en XR lanzando teletransporte

1. **Sala de entrada:** El usuario aparece aquí y puede consultar un breve

tutorial de los controles de la aplicación en formato texto, una descripción del proyecto y los logotipos de las entidades que han formado parte de él.

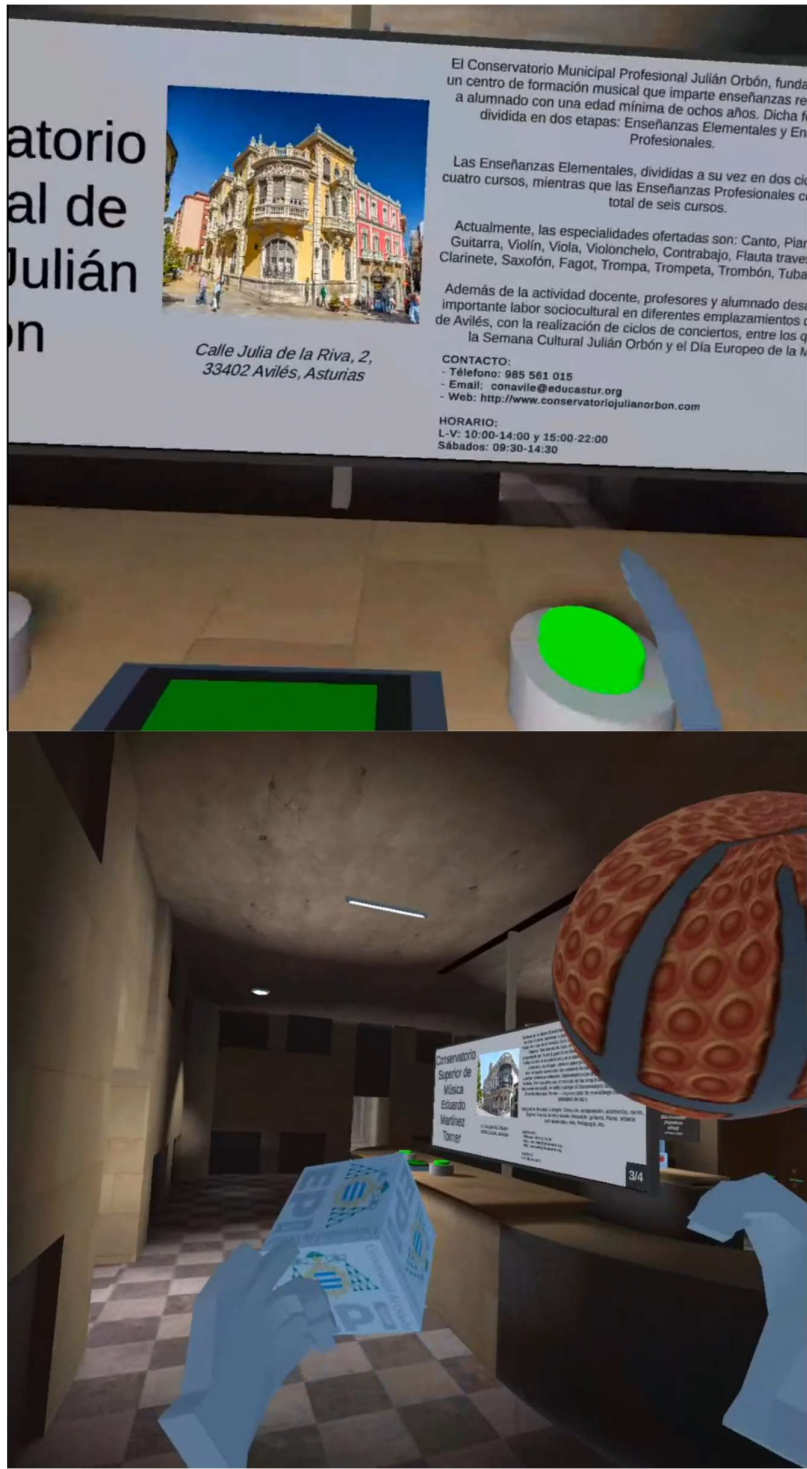




Ilustración 7.29. Interacciones en XR en el vestíbulo

2. Vestíbulo: La única sala a la que se puede acceder desde la sala de entrada. El vestíbulo contiene diversos objetos interactivos que pueden agarrarse y lanzarse y, además, contiene un panel de información sobre instituciones de música de la región.

Este panel puede desplegarse activándolo en un conmutador. Una vez que el panel este desplegado el usuario podrá moverse entre las páginas de este utilizando dos botones sobre la mesa.

Desde esta sala se puede acceder a las salas laterales y a la sala de eventos tras traspasar una puerta que se abre de forma manual.



Ilustración 7.30. Interacción XR en el patio.

- 3. Patio:** Como parte extra de la escena y llevada a cabo al margen del diseño inicial del proyecto nos encontramos con el único punto exterior de todo el proyecto al que el usuario puede acceder.

En este patio, a modo de *minijuego sorpresa* el usuario podrá lanzar pelotas a una canasta de baloncesto. Su puntuación se visualizará en la pared y será recompensado cada vez que enceste.

El motivo por añadir este aspecto es para probar las funcionalidades referidas a las físicas del motor y a ofrecer al usuario una forma de divertirse entre tanto visionado de información cultural.

- 4. Salas laterales:** Las salas laterales, a las que se acceden desde el vestíbulo, no ofrecen ninguna función interactiva, tan solo sirven de puente para acceder a otras escenas.

Desde la sala izquierda se puede acceder a la sala de instrumentos y a la sala asturiana. Desde la sala derecha se puede acceder al patio y a la sala de teoría y juego.

7.4.3 Sala de eventos

La interacción en la escena del salón de eventos se centra en el visionado de videos. Los vídeos son seleccionados por el usuario de entre varias posibilidades que se presentan en varias “cintas de vídeo”.

Por un lado, en esta primera versión de la aplicación se ofrece la reproducción de dos vídeos preparados para test: *Big Buck Bunny* y *The Elephants Dream*, dos vídeos de animación clásicos para realizar pruebas en aplicaciones y *benchmarks*.

Por otro lado, se instancia de forma dinámica varias cintas de vídeo en un lateral de la sala. Estos vídeos se almacenan en una base de datos MinIO que se despliega junto a la presente aplicación. Los detalles del despliegue, entre los que se encuentra el servicio de base de datos, se detallarán más adelante en el apartado *8.Arquitectura y despliegue final*.

Estas cintas de vídeo se instancian cuando el usuario accede a la sala siguiendo el siguiente proceso:

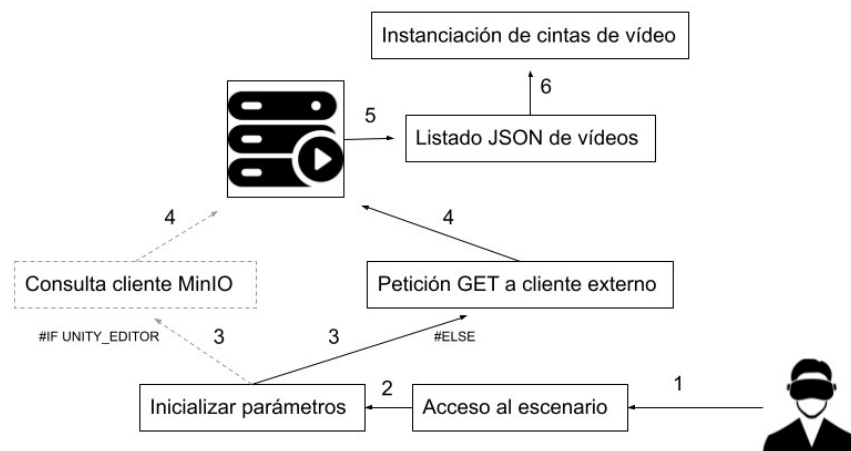


Ilustración 7.31. Proceso de instanciación de VideoTapes

1. El usuario accede a la sala de eventos a través de la escena principal. La escena es cargada y se instancian todos los elementos de esta.
2. El evento *Start* del script *TapeGeneratorSourceSettings* se ejecuta, lo que

inicializa los parámetros:

- a. Se leen las variables de entorno y el fichero de configuración (si una variable no es encontrada en el entorno se toma de este fichero).
 - b. Se activa o se desactiva el script *http_videoTapeGenerator* o *MinIO_videoTapeGenerator* en función de si la configuración especifica usar MinIO o no.
 - c. Se leen los parámetros de configuración: Tanto para MinIO (host, puerto, uso de SSL y claves para el caso de MinIO), como para Http (host, puerto y nombre del fichero del listado de videos) y para el cliente externo de MinIO (url, si se ejecuta desde fuera del editor de Unity).
3. Se realiza la consulta del listado de vídeos:
- a. Si la aplicación se está ejecutando el editor de Unity, el cliente de MinIO lanza la petición y obtiene el listado de objetos del *bucket* especificado.
 - b. Si la aplicación se ejecuta desde *WebGL* una vez realizado el *Build*, se lanza la petición *Get* al cliente externo y, si este está desplegado, se retorna el listado de objetos almacenados en el *bucket* especificado.
 - c. Sea cual sea el camino (a o b), el listado de objetos se obtiene en formato JSON.
4. El listado JSON se transforma en objetos *VideoTape* y estos son instanciados en la posición específica.

Si por algún motivo no se tiene conexión a la base de datos o el servicio o cliente externo se encuentra inactivo, las cintas de vídeo no se instancian, pero el usuario puede seguir la interacción sin problema.

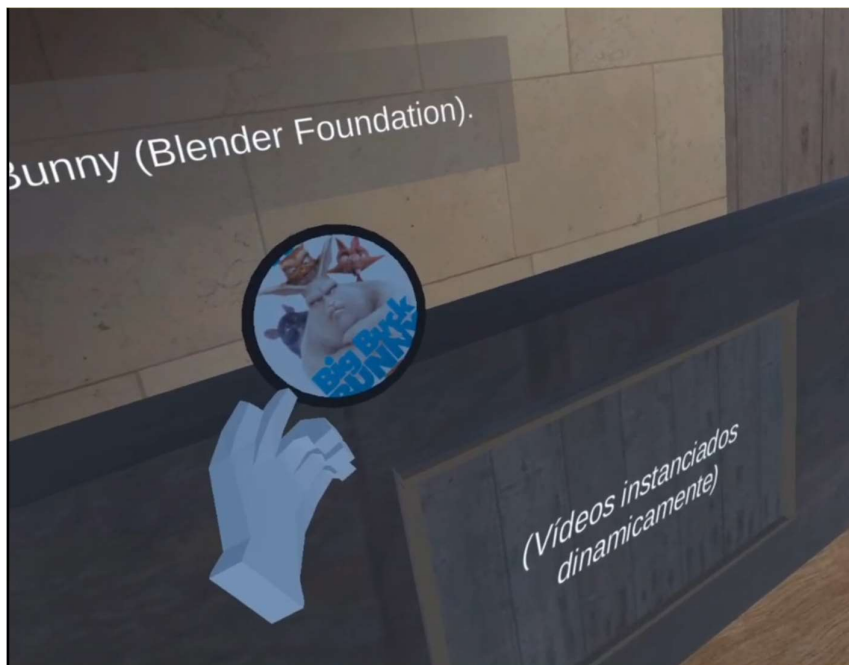


Ilustración 7.32. Instanciación de cintas en XR.

Por otro lado, cuando el usuario ya se encuentre en la sala y tiene libertad para escoger las cintas de vídeo se puede llevar el siguiente proceso para reproducir vídeos en la pantalla:

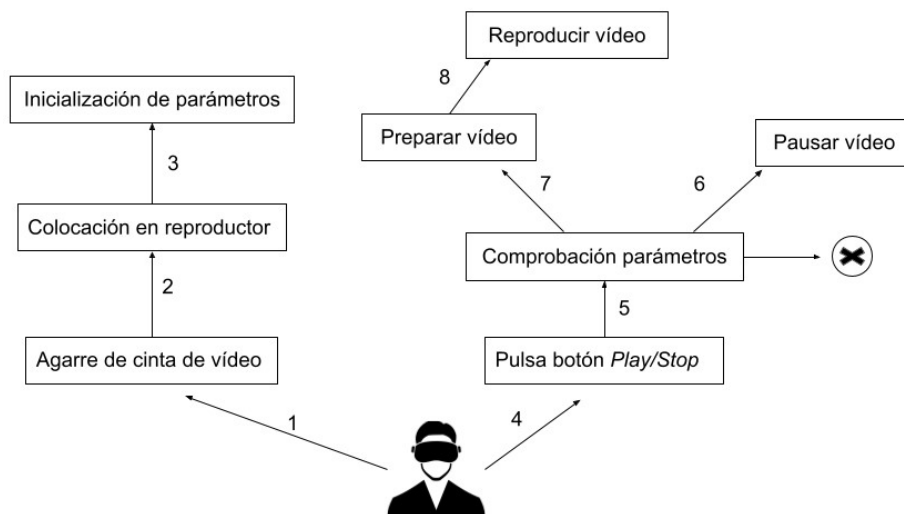


Ilustración 7.33. Proceso para reproducir vídeo.

1. El usuario agarra una cinta de vídeo.
2. El usuario coloca la cinta de vídeo en el reproductor (en las ruedas superiores del proyector).

3. Los parámetros del reproductor de vídeo se inicializan:
 - a. Url a reproducir.
 - b. Se inicializa la fuente de audio para reproducir el sonido del vídeo.
4. El usuario puede reproducir el vídeo pulsando el botón de Play/Stop.
 - a. Para que el botón de Play/Stop se encuentre habilitado primero se deberá colocar la cinta de vídeo.
 - b. Para que el botón de Play/Stop se encuentre habilitado también deberá desplegarse la pantalla activando la palanca del centro de control.
5. Los parámetros se comprueban, si hay algún error el vídeo no es reproducido y el botón no cambia de estado.
6. Si el vídeo se estaba reproduciendo, se pausa.
7. Si el vídeo no se estaba reproduciendo, el vídeo se prepara. Una vez el vídeo se encuentre preparado (se comprueba la petición para reproducirlo en *streaming*) se espera al siguiente *fotograma*.
8. La luz del proyector se activa y el vídeo se reproduce en la pantalla.





Ilustración 7.34. Proceso para reproducir videos en XR

7.4.4 Sala de instrumentos

La sala de instrumentos es simple respecto a lo que interacción respecta. En

esta sala encontraremos diferentes objetos con los que el usuario podrá interactuar.

Cuando el usuario interactuare con un objeto (cada objeto se corresponde a un instrumento diferente) se desplegará una ventana donde podrá visualizar una descripción del instrumento. Bajo el texto se le permitirá activar botones que reproducen audio.

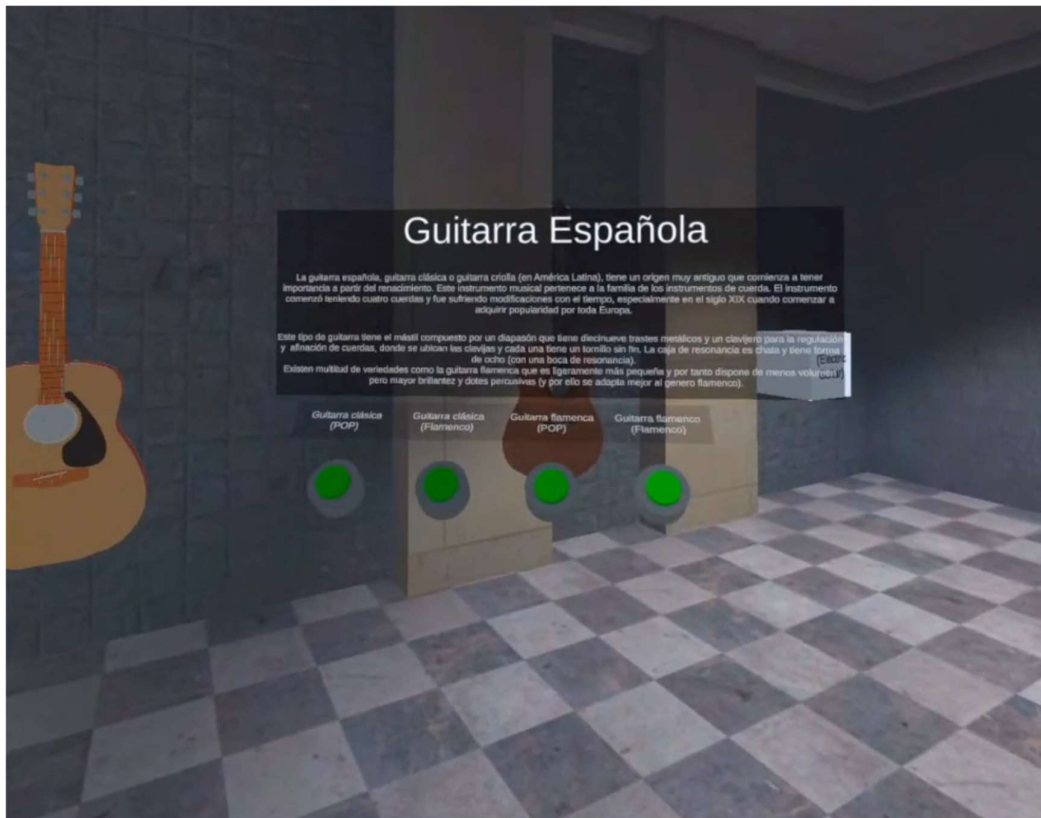


Ilustración 7.35. Interacción con instrumento en XR

7.4.5 Sala asturiana

La sala asturiana es la segunda escena de simple interacción. En esta sala encontraremos también encontraremos únicamente, diferentes objetos con los que el usuario podrá interactuar.

Estos objetos se corresponden con instrumentos o elementos de la cultura de la región. Cuando interactué con uno de los objetos, el usuario volverá a visualizar texto descriptivo y controladores en forma de botones que reproducen audio al activarlos.



Ilustración 7.36. Interacción XR en la sala asturiana

7.4.6 Sala de teoría y juego

La sala de teoría y juego sirve para que el usuario pueda estudiar nociones de teoría musical básicas. Se divide en dos secciones:

A mano **izquierda**, el usuario accederá a la sala de la teoría.

1. En esta sala encontrará diversos objetos interactivos que mostrarán texto o texto y audio cuando el usuario interacciona con ellos.
2. Estos objetos se corresponden con: Teoría musical básica, escalas de diversos tipos y construcción de acordes básicos.
3. Cada objeto que se corresponde con una escala musical tiene a su lado una cinta de teoría musical.
 - a. Las cintas de teoría musical recuerdan a las cintas de vídeo que pueden visualizarse en el salón de eventos, pero tienen otra función

que se detallará en la siguiente ala de la escena.

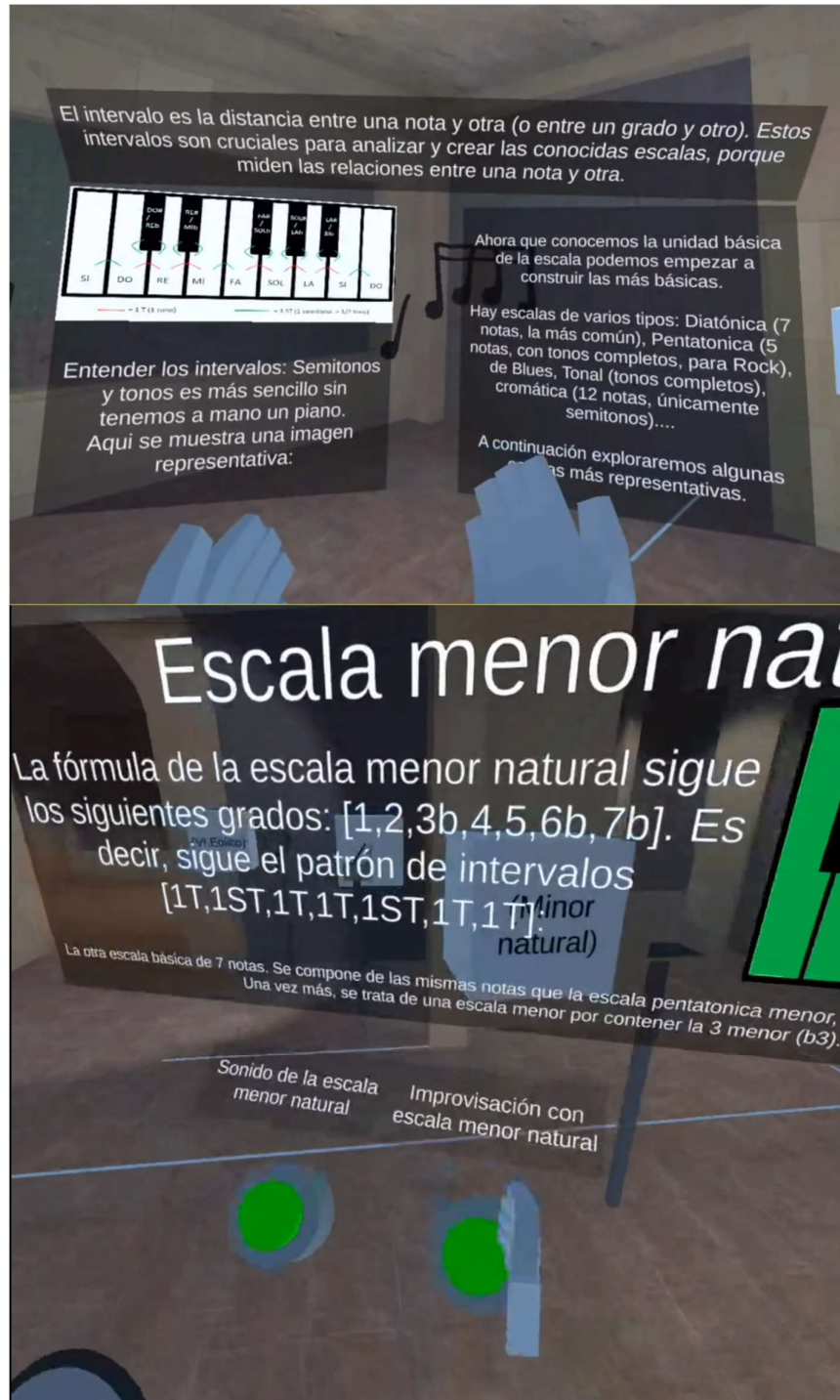


Ilustración 7.37. Visualizando teoría en modo XR.

A mano **derecha**, el usuario accederá a la sala de juego.

1. El juego en realidad es sencillo. El usuario podrá interactuar con un panel

música (con el clic del ratón si está en modo no inmersivo con las manos o las baquetas si está en modo inmersivo).

2. Se muestran, en esta primera versión, dos paneles musicales:
 - a. Uno sencillo (sin semitonos, 7 notas)
 - b. y un panel completo (12 notas).
 - c. Cada elemento de los paneles se corresponde con una nota.
3. El usuario puede modificar el *pitch* de ambos paneles (agudeza del sonido que emiten).
4. El usuario puede traer una de las cintas de teoría musical ubicadas en la sala de teoría y colocarlas en el reproductor que se encuentra junto al panel completo.
 - a. Al colocar la cinta en el reproductor, se iluminarán las notas correspondientes a la escala introducida.
 - b. El reproductor tiene consigo dos botones para variar la tonalidad de la escala: Por ejemplo, si coloca la escala mayor natural podrá ir variando si iluminar la escala de Do Mayor, la de Re Mayor, etc.



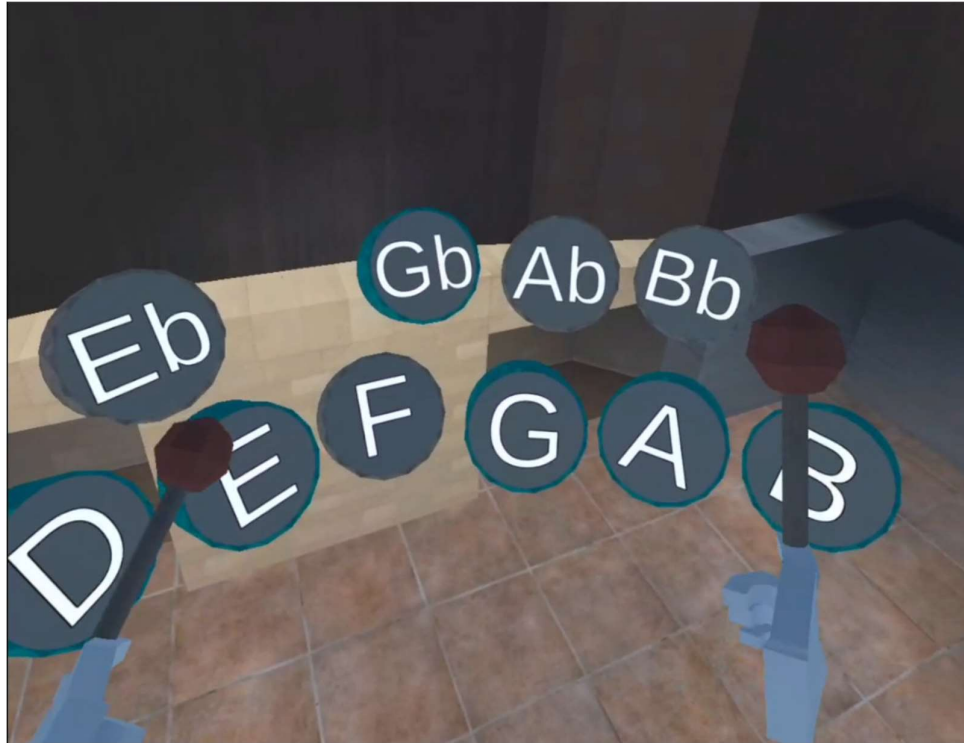


Ilustración 7.38. Colocando una cinta y tocando en la sala de juego (XR)

7.5 Dificultades encontradas

Este apartado se centrará en las dificultades encontradas durante el desarrollo de la aplicación resultado del presente proyecto y, si procede, la solución dada al problema:

- Es obvio pensar que la mayor dificultad se encuentra en el uso de Unity, el manejo del motor ha sido algo muy novedoso (apenas se ha utilizado en un taller de cuatro horas en el máster). Ello juntado a necesitar desarrollar el trabajo para realidad virtual y exportarlo a web ha hecho que este sea el aspecto más destacado de las dificultades de desarrollo.
- El uso de WebGL (para reproducir el contenido en el navegador) acarrea una serie importante de desventajas:
 - Algunos parámetros tienen diferentes posibles valores cuando se ejecuta una vez realizado el build (e.g. El pitch del sonido).

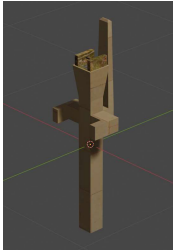
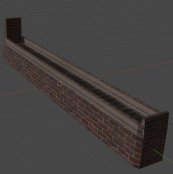
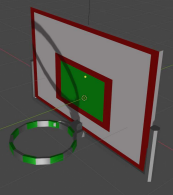

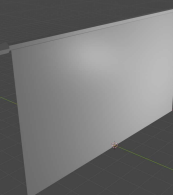
- No es posible modificar el volumen de los audios emitidos en ejecución.
- Al correr la aplicación en un navegador nos encontramos con restricciones de seguridad al intentar reproducir cierto contenido multimedia.
- El uso en el navegador precisa de una mayor optimización a la hora de desarrollar los objetos y el escenario, pues el rendimiento será menor que en una aplicación de escritorio. Respecto a este punto podemos referirnos al apartado 7.2. *Problemas de rendimiento iniciales* visto anteriormente.
- El manejo del ratón, debido a las restricciones de seguridad mencionadas anteriormente, también se hace difícil en WebGL. Es preciso su bloqueo en el centro de la pantalla mientras se está en modo no inmersivo y hacer uso de *Raycast* desde el centro de la pantalla para la interacción en vez del clic estándar del ratón.
- WebGL no soporta hilos ni llamadas a métodos asíncronos, lo que hace que muchas funciones de paquetes específicos de C# conlleven el lanzamiento de excepciones no controladas (esto se ve claramente con el cliente de MinIO, razón por la que se desarrolló el cliente externo).
- Todos los errores anteriores y sus posteriores soluciones solo podían probarse una vez realizado el build y el lanzamiento de la aplicación, por lo que la carga de tiempo fue mayor en el desarrollo.
- Muchos problemas anteriores se solucionan con directivas, por ejemplo: `#IF UNITY_WEBGL` o `#IF UNITY_EDITOR`.

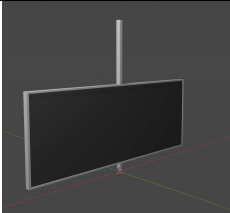
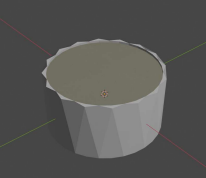
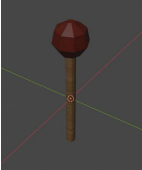


- Problemas de rendimiento: En las fases iniciales del proyecto el rendimiento a la hora de utilizar hardware inmersivo era muy bajo, el contenido se reproduce a una tasa de imágenes por segundo muy baja lo que imposibilitaba su uso (de nuevo, nos referenciamos al apartado 7.2. *Problemas de rendimiento iniciales*).
 - La reducción de polígonos o el *bakeo* de luces, así como el uso de *Occlusion* pudo solventar este problema en gran medida.
- El manejo de UI (interfaz gráfica) en modo inmersivo no se hacía posible en WebGL. Existe multitud de métodos y paquetes para poder realizar *raycast* desde la mano del usuario a elementos de interfaz gráfica (e.g. Botones), pero en este proyecto no ha sido posible por utilizar *WebGL*.
 - Se optó por una “interacción física”, como se puede observar en la interacción con todos los controladores y, en especial, con el menú del usuario en modo inmersivo.
- El exportado de objetos desde *Blender* conllevaba errores en escalas y rotaciones, así como en el mapa de luces de dichos objetos.
 - La corrección de los parámetros para encontrar la configuración necesaria elevó la carga de trabajo en este aspecto.
- El uso de *http* conlleva la imposibilidad de reproducir el contenido en modo inmersivo, por lo que la implementación del servicio utilizando SSL se tuvo que realizar en una fase muy temprana del proyecto.
- La interacción en modo inmersivo que permite el lanzamiento de objetos conllevó una gran carga de trabajo en cuanto a tiempo se refiere en pruebas en el uso de las físicas del motor.

7.6 Objetos modelados

Reiterando que no es el objetivo principal del presente trabajo y dejando a un lado el escenario, que ha sido modelado completamente por el autor del presente trabajo, se han modelado también algunos otros objetos simples para acompañar a ciertas funcionalidades de la aplicación. A continuación, se listan los modelados en *Blender*:

Tabla 7.10. Objetos modelados.

Nombre	Foto	Descripción
<i>AudioLector</i>		Base para el lector de cintas de teoría musical. La parte superior sirve como base para fijar una cinta y mostrar la tonalidad escogida, los brazos inferiores para modificar la tonalidad mediante dos botones.
<i>BallsBase</i>		Sirve para recibir las bolas que son instanciadas y propulsadas.
<i>BasketCase</i>		Canasta estándar de baloncesto, su objetivo es que el usuario lance bolas hacia ella.
<i>CinemaProjector</i>		Proyector de cine para colocar las cintas de video de la sala de eventos. Las ruedas superiores sirven para fijar al usuario donde deberá colocar la cinta.
<i>CinemaProjectorScreen</i>		Pantalla donde se visualizará el video escogida. Se la parte de la pantalla es independiente a la superior, pues esta se aparecerá utilizando una

Nombre	Foto	Descripción
		animación.
<i>CustomPanel</i>		Panel pensado para aparecer del techo y mostrar contenido de texto e imágenes en el vestíbulo de la sala principal.
<i>DrumPad</i>		Tambor que servirá para que el usuario reproduzca sonidos al interactuar con él.
<i>DrumStick</i>		Baqueta que el usuario agarre y golpee los Drumpads u objetos musicales con ello.
<i>MainTable</i>		Mesa grande cuyo objetivo será albergar los objetos del vestíbulo en la sala principal.
<i>Tape</i>		Malla de las cintas, tanto de vídeo como de teoría musical. Contiene una base que podrá mostrar texto o una imagen.

8 Despliegue y arquitectura final

Con el fin de facilitar el desarrollo de las pruebas, la actualización de la aplicación y el acceso a los recursos se propone el uso de una arquitectura basada en contenedores.

Al comienzo del desarrollo y para las pertinentes entregas de los hitos del proyecto al tutor, se utilizó un despliegue sencillo en *Heroku* [26]. El despliegue en esta plataforma de computación en la nube (o PaaS o Platform as a Service) tan solo contenía un sencillo servidor http lanzado en Python que alojaba los ficheros estáticos

correspondientes a la aplicación compilada. Este es el enfoque que sigue utilizando el despliegue del Estudio inicial (en concreto apartado 5.2.1 del presente documento).

El uso de contenedores (Tecnología *Docker* [27]) dota al despliegue de una compatibilidad total para ejecutarlo en cualquier tipo de máquina independientemente de su sistema operativo, ello facilita el despliegue enormemente por tener que programar la aplicación una sola vez. Además, se obtiene una mayor consistencia entre los entornos de prueba y los entornos de producción: Cuando se trabaja con *Docker* las pruebas se realizan dentro del contenedor, haciendo que el entorno de producción sea idéntico al de desarrollo.

Por otro lado, se obtiene un diseño más modular. Como veremos más adelante, a pesar de que este proyecto presente una arquitectura extremadamente sencilla, el uso de *Docker* permitirá lanzar varios servicios en la misma máquina que trabajen juntos entre sí para cumplir los objetivos del proyecto. Esto no conllevará necesariamente un aumento de la dificultad a la hora de desplegar el proyecto, con el uso de *Docker-compose* [28] el desplegar todos los servicios se hará lanzando con un solo comando.

8.1 Arquitectura

Cómo se ha comentado anteriormente, se hará uso de *Docker* junto con *Docker-Compose* para la arquitectura del proyecto.

La arquitectura constará de, en total, tres contenedores independientes que tendrán una función específica cada uno de ellos:

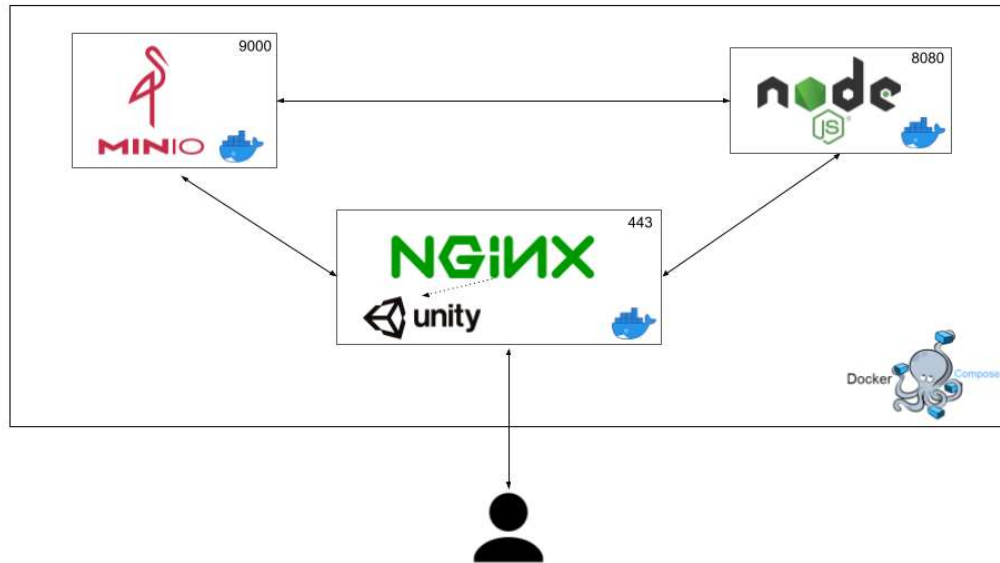


Ilustración 8.39. Arquitectura del proyecto.

- En primer lugar, nos encontramos con el contenedor principal que alojará un servicio de **NGINX**. Este servicio se trata de un estándar a la hora de levantar servidores web o proxys inversos ligeros de alto rendimiento, es una alternativa de código abierto. En el caso concreto de este proyecto:
 - Es un servidor de ficheros estáticos en el que se encuentra la aplicación Unity WebXR ya compilada.
 - La aplicación se visualiza en el fichero index.html resultante de compilar la aplicación.
 - Se accede desde el puerto SSL (443). El uso de HTTPS es imprescindible para la experiencia inmersiva, es por ello por lo que se hace uso de certificados auto firmados.
- En segundo lugar, nos encontramos con un contenedor que alberga un servicio de **MinIO**. MinIO es un servidor de almacenamiento escalable de objetos en la nube que es altamente compatible con otras alternativas de pago (Amazon Web Services, Google Cloud, etc.). En este proyecto:
 - Albergará los vídeos que el usuario podrá visualizar en el salón

de eventos. Estos son visualizados directamente de MinIO accediendo a las url de los archivos.

- Su alta compatibilidad hace que, en un futuro y si las circunstancias lo requieren, el migrar el almacenamiento a una alternativa cloud sea un proceso extremadamente sencillo.
- En un comienzo, el almacenamiento de videos se iba a llevar a cabo en otra instancia de NGINX (cliente http). Pero el uso de MinIO ofrece dinamismo y sencillez (para añadir un nuevo vídeo basta con acceder a su interfaz y arrastrar el archivo). Además, la integración de Unity WebXR y MinIO era un objetivo que cumplir en uno de los proyectos llevados a cabo por CTIC.
- Por último, nos encontramos con un servicio **NodeJS** que levantará una sencilla API. Las llamadas a esta API comunicarán un cliente MinIO con dicho servicio de almacenamiento con el objetivo de: Acceder al listado de *buckets* (grupos de objetos), listar los objetos de un *bucket*... etc.
 - El objetivo de este cliente es obtener cuantos vídeos se encuentran almacenados en MinIO para generar automáticamente y de forma dinámica las cintas de vídeo cada vez que el usuario accede al salón de eventos.
 - Es utilizado un cliente de MinIO para *JavaScript* en un servicio dentro de un contenedor externo en vez de un cliente para C# en Unity debido a que las funciones utilizadas por este último precisan de usar en todos los casos *async/await*: Los hilos y las tareas (entre otras muchas cosas) NO son compatibles con WebGL e imposibilitan su uso.
 - La aplicación hace uso del cliente de C# (obtenido a través

del gestor de paquetes propio de *Visual Studio: Nuget*) cuando se ejecuta desde el editor de Unity, una vez realizado el *Build* se pasa a utilizar el cliente externo, cuyo cliente ha sido obtenido a través del gestor de paquetes para node: *npm*.

8.2 Acceso al despliegue

El despliegue detallado en el apartado anterior se encuentra levantado en los servidores de CTIC (más concretamente en el clúster del departamento de Web Of Things, al que pertenece el autor del presente proyecto).

Se ha usado este enfoque para facilitar un acceso público para el desarrollo de las pruebas pertinentes y, posteriormente, para la evaluación y defensa del proyecto.

Resulta obvio pensar que la red de CTIC es privada, es por ello por lo que se hace uso de la herramienta *test.ctic*, un proxy inverso diseñado en el centro tecnológico para hacer públicos servicios levantados en máquinas de la red privada con el objetivo de realizar pruebas.

Por tanto, el flujo de acceso es el que se muestra en la siguiente ilustración:

1. El usuario accede desde su navegador a la url pública expuesta para el acceso: <https://enrevimus.test.ctic.es>
2. La herramienta de mapeos (proxy inverso) redirige la petición al puerto 443 (ssl) de wot-cluster-07. Esta dirección es donde se encuentra realmente alojada la aplicación.
3. La dirección anterior lleva al usuario al contenedor con el servidor NGINX en el que se encuentra el proyecto compilado, mostrando el contenido Unity en la página *Index.html*.

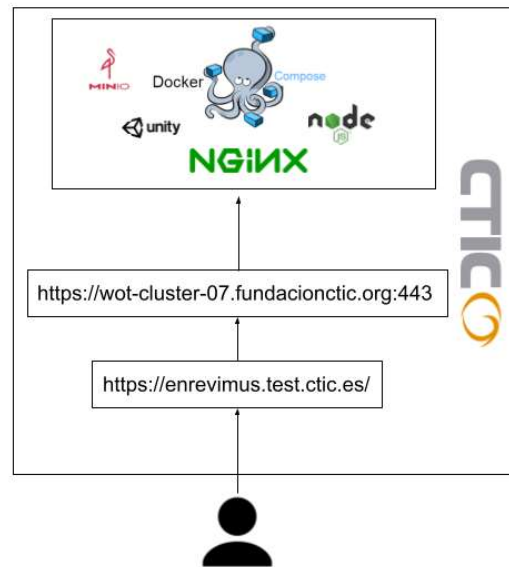


Ilustración 8.40. Acceso al proyecto.

8.2.1 Otros servicios

El resto de los servicios también están incluidos en la herramienta de mapeos para garantizar un acceso público (y para que en el entorno de pruebas los servicios puedan comunicarse entre sí).

El acceso es análogo a lo explicado en el apartado anterior, pero aplicado al resto de servicios:

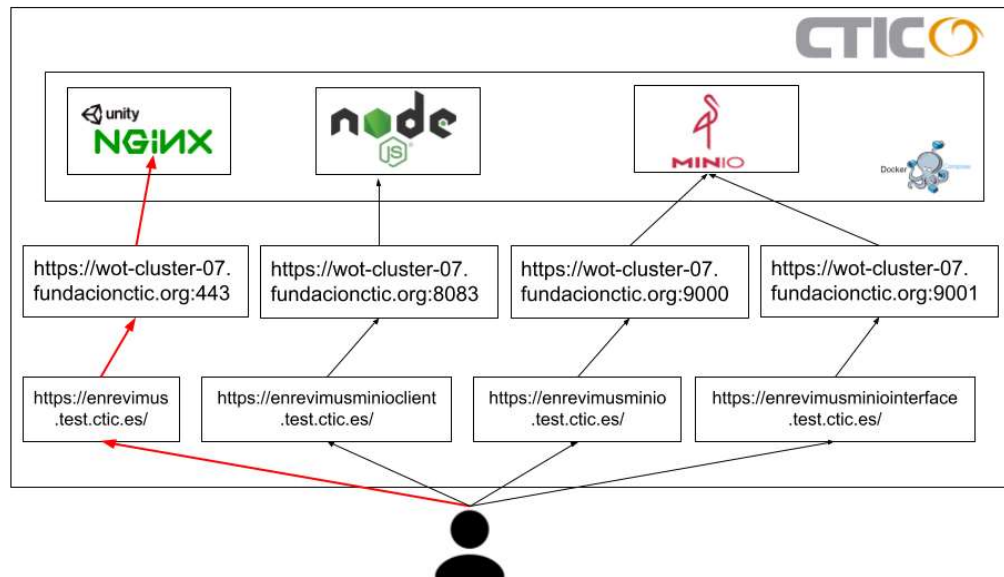


Ilustración 8.41. Acceso a los servicios

En este primer entorno de pruebas la aplicación Unity precisa de realizar peticiones al cliente de MinIO y a la base de datos MinIO, el cliente también realiza peticiones a este último servicio.

Las peticiones están configuradas para lanzarse a un host y a un puerto. En este caso se lanzan a la URL pública ([https://enrevimus\[X\].test.ctic.es](https://enrevimus[X].test.ctic.es)) y al puerto 443 (SSL). La herramienta de mapeos se encargará de redirigir la petición a donde sea conveniente.

8.3 Lanzamiento del despliegue

El despliegue se encuentra en un repositorio independiente (también adjuntado como código en la entrega del trabajo).

Bastará con encontrarse en el directorio de dicho repositorio, donde deberá crearse un nuevo fichero `.env` (puede hacerse una copia de `.env.default` teniendo en cuenta que las variables que se encuentran en este fichero son de prueba) y ejecutar:

```
Run . sh
```

El script anterior no hará más que forzar la construcción de las imágenes

(`docker-compose up --force-recreate --build -d`) y levantar el despliegue (o reiniciarlo si ya estaba levantado), de esta forma se mantiene actualizado con tan solo lanzar un comando.

Véase de nuevo las ventajas del uso de *Docker*, independientemente de donde se vaya a desplegar la aplicación el proceso es exactamente el mismo. También añadimos compatibilidad pues para desplegar los servicios en otros puertos o en otra ubicación basta con modificar las variables de entorno.

NOTA: Para evitar aumentar innecesariamente el tamaño del repositorio de despliegue (y para cumplir las restricciones de tamaño de *GitHub*) la aplicación no se encuentra compilada. Por lo que para que el despliegue se lleve a cabo de forma correcta es necesario compilarla dentro del directorio `/data/app`).

9 Pruebas

Aunque se han ido llevado a cabo pruebas iterativas a lo largo de todo el desarrollo (tanto del proyecto resultante como del estudio inicial), se llevarán a cabo una serie de pruebas diseñadas para asegurar que la aplicación es entregada en un estado funcional óptimo que evite errores en la interacción.

Estas pruebas forman parte de la parte final del desarrollo, pero no por ello son menos importantes. Normalmente, y por desgracia, se dedica poco tiempo al desarrollo de unas pruebas concretas que permitan concluir que la aplicación es robusta y fiable para su uso en un entorno real.

Las pruebas serán indicadas de forma tabular, detallando para cada caso de prueba los siguientes apartados.

- **ID:** Identificador numérico único de la prueba
- **Descripción:** En que consiste la prueba a realizar.

- **Precondición:** Indica en qué estado se encuentra el sistema antes de realizar la prueba.
- **Postcondición:** Indica en qué estado se encuentra el sistema después de realizar la prueba.
- **Resultado esperado:** El resultado que debe de suceder para que la prueba determine que la operación ha sido correcta.
- **Resultado obtenido:** El resultado obtenido realmente al realizar la prueba. Puede indicarse:
 - Resultado esperado: Se ha obtenido un resultado igual al esperado, lo que concluye que la prueba ha sido correcta y favorable.
 - Resultado diferente al esperado: Indicando en este caso que resultado se ha obtenido.

Los casos de prueba habitualmente conllevan la inserción de datos de prueba, pero en el caso que ocupa la presente aplicación el usuario no introduce ningún tipo de información en el sistema por lo que se omite este aspecto.

También debe destacarse que los casos se reducen en gran medida si tenemos en cuenta que muchas funcionalidades se activan interactuando con controladores físicos que ya son *prefabs*. Por ejemplo, si sabemos que un botón funciona correctamente en modo no inmersivo (se clica en el él y se lanza un evento) y que también lanza el evento en modo inmersivo (al interactuar con él con la mano), podemos asegurar que el resto de los botones del proyecto lanzarán el evento asignado correctamente tanto en modo inmersivo como no.

Todas las pruebas han sido desarrolladas accediendo a la aplicación desde el navegador web del equipo de desarrollo una vez la aplicación ha sido desplegada y también accediendo desde unas *Meta Quest 2* desde su navegador (*Oculus Browser*).

Por otro lado, y como una nota indicativa del presente documento: Las tablas donde se describen los casos de prueba no llevaran consigo un título debido a que la longitud de este apartado podría conllevar alargar el índice de tablas de forma innecesaria.

9.1 Pruebas de estado y menú

A continuación, se listan las pruebas referidas al usuario, al cambio de estado (a modo inmersivo y viceversa) y a la gestión de la configuración desde el menú.

ID: 1	
Descripción	Acceso a la aplicación web desde el navegador
Precondición	El usuario accede a la url desde un navegador sin tener conectado hardware de realidad virtual.
Postcondición	La web carga el contenido Unity y, una vez finalizado el proceso, se muestra el menú UI con interacción no inmersiva. El botón VR se encuentra deshabilitado.
Resultado esperado	El usuario visualiza el menú principal y no puede acceder al modo inmersivo.
Resultado obtenido	Resultado esperado

ID: 1.1	
Descripción	Acceso a la aplicación web desde el navegador (Oculus)
Precondición	El usuario accede a la url desde un navegador con el hardware de realidad virtual conectado.
Postcondición	La web carga el contenido Unity y, una vez finalizado el proceso, se muestra el menú UI con interacción no inmersiva. El botón VR se encuentra habilitado.
Resultado esperado	El usuario visualiza el menú principal y se le ofrece la posibilidad de acceder al modo inmersivo clicando en el botón pertinente.
Resultado obtenido	Resultado esperado

ID: 1.2	
Descripción	Acceso a la aplicación web desde el navegador (Oculus Browser)
Precondición	El usuario accede a la url desde el navegador interno del hardware de realidad virtual.
Postcondición	La web carga el contenido Unity y, una vez finalizado el proceso, se muestra el menú UI con interacción no inmersiva. El botón VR se encuentra habilitado.
Resultado esperado	El usuario visualiza el menú principal y se le ofrece la posibilidad de acceder al modo inmersivo clicando en el botón pertinente.
Resultado obtenido	Resultado esperado

ID: 2	
Descripción	Interacción no inmersiva con el menú principal
Precondición	El usuario ha accedido y visualiza el menú en modo no inmersivo.
Postcondición	Las opciones son modificadas por el usuario
Resultado esperado	El usuario es capaz de modificar la configuración de la aplicación (tanto la configuración inmersiva como la no inmersiva), la aplicación responde a los clics del usuario.
Resultado obtenido	Resultado esperado

ID: 3	
Descripción	Cambio a modo inmersivo en el menú
Precondición	El usuario ha accedido y visualiza el menú en modo no inmersivo, tiene hardware de realidad virtual conectado al equipo.
Postcondición	La aplicación ahora es cargada en las gafas de realidad virtual y no en el navegador. El usuario ahora interacciona con el entorno en realidad virtual.
Resultado esperado	El contenido se carga en las gafas de realidad virtual. Se muestran las manos del usuario y se permite la interacción inmersiva moviendo la cámara con la cabeza y las manos con los controladores. Se oculta la UI del menú.
Resultado obtenido	Resultado esperado

ID: 3.1	
Descripción	Cambio a modo inmersivo en el menú (Oculus Browser)

ID: 3.1	
Precondición	El usuario ha accedido y visualiza el menú en modo no inmersivo desde el navegador de su hardware de realidad virtual.
Postcondición	El usuario ahora interacciona con el entorno en realidad virtual.
Resultado esperado	Se muestran las manos del usuario y se permite la interacción inmersiva moviendo la cámara con la cabeza y las manos con los controladores. Se oculta la UI del menú.
Resultado obtenido	Resultado esperado

ID: 4	
Descripción	Cambio a modo no inmersivo en el menú
Precondición	El usuario ha accedido al menú y ha activado el modo inmersivo, visualizando el contenido en las gafas que se encuentran conectadas al equipo.
Postcondición	El usuario deja de visualizar el contenido en las gafas y comienza a visualizarlo en el navegador del equipo.
Resultado esperado	Se ocultan las manos del usuario y se muestra el menú principal, la interacción pasa a realizarse con el ratón.
Resultado obtenido	Resultado esperado

ID: 4.1	
Descripción	Cambio a modo no inmersivo en el menú (Oculus Browser)
Precondición	El usuario ha accedido al menú desde el navegador de sus gafas y ha activado el modo inmersivo, visualizando el contenido en el visor. Cancela el modo inmersivo.
Postcondición	El usuario pasa a controlar el menú de forma no inmersiva a modo de clics.
Resultado esperado	Se ocultan las manos del usuario y se muestra el menú principal, la interacción pasa a realizarse mediante clics.
Resultado obtenido	Resultado esperado

ID: 5	
Descripción	Interacción del menú en modo inmersivo
Precondición	El usuario ha activado el modo inmersivo y visualiza el contenido en sus gafas (conectadas o <i>standalone</i>).

ID: 5	
Postcondición	El menú es mostrado en la mano izquierda del jugador.
Resultado esperado	En interacción inmersiva, el usuario mantiene pulsado el botón Y de su mano izquierda y visualiza el menú en dicha mano. El menú mira constantemente de frente a la cámara del usuario.
Resultado obtenido	Resultado esperado

ID: 5.1.1	
Descripción	Interacción del menú en modo inmersivo (Activar rotar Joystick)
Precondición	El usuario ha activado el modo inmersivo y visualiza el contenido en sus gafas (conectadas o <i>standalone</i>). Esta visualizando el menú en su mano izquierda e interacciona con su mano derecha (pulsando el botón <i>Grip</i>) en el botón del menú referido a la opción <i>Rotar con joystick</i> (opción que se encuentra desactivada). La rotación por pulsos se encuentra desactivada.
Postcondición	La opción de rotar con Joystick se encuentra activada.
Resultado esperado	La opción <i>Rotar con joystick</i> es activada (su conmutador se ilumina en verde) y el usuario es capaz de rotar la cámara horizontalmente con el stick derecho.
Resultado obtenido	Resultado esperado

ID: 5.1.2	
Descripción	Interacción del menú en modo inmersivo (Desactivar rotar Joystick)
Precondición	El usuario ha activado el modo inmersivo y visualiza el contenido en sus gafas (conectadas o <i>standalone</i>). Esta visualizando el menú en su mano izquierda e interacciona con su mano derecha (pulsando el botón <i>Grip</i>) en el botón del menú referido a la opción <i>Rotar con joystick</i> (opción que se encuentra activada).
Postcondición	La opción de rotar con Joystick se encuentra desactivada.
Resultado esperado	La opción <i>Rotar con joystick</i> es desactivada (su conmutador se ilumina en rojo) y el usuario no es capaz de rotar la cámara con el stick derecho.
Resultado obtenido	Resultado esperado

ID: 5.2.1	
Descripción	Interacción del menú en modo inmersivo (Activar movimiento Joystick)
Precondición	El usuario ha activado el modo inmersivo y visualiza el contenido en sus gafas (conectadas o <i>standalone</i>). Esta visualizando el menú en su mano izquierda e interacciona con su mano derecha (pulsando el botón <i>Grip</i>) en el botón del menú referido a la opción movimiento <i>con joystick</i> (opción que se encuentra desactivada).
Postcondición	La opción de movimiento con Joystick se encuentra activada.
Resultado esperado	La opción Movimiento <i>con joystick</i> es activada (su conmutador se ilumina en verde) y el usuario es desplazarse con el joystick izquierdo.
Resultado obtenido	Resultado esperado

ID: 5.2.2	
Descripción	Interacción del menú en modo inmersivo (Desactivar movimiento Joystick)
Precondición	El usuario ha activado el modo inmersivo y visualiza el contenido en sus gafas (conectadas o <i>standalone</i>). Esta visualizando el menú en su mano izquierda e interacciona con su mano derecha (pulsando el botón <i>Grip</i>) en el botón del menú referido a la opción movimiento <i>con joystick</i> (opción que se encuentra activada).
Postcondición	La opción de movimiento con Joystick se encuentra desactivada.
Resultado esperado	La opción Movimiento <i>con joystick</i> es desactivada (su conmutador se ilumina en rojo) y el usuario no es capaz de desplazarse con el joystick izquierdo.
Resultado obtenido	Resultado esperado

ID: 5.3.1	
Descripción	Interacción del menú en modo inmersivo (Activar rotación por pulsos)
Precondición	El usuario ha activado el modo inmersivo y visualiza el contenido en sus gafas (conectadas o <i>standalone</i>). Esta visualizando el menú en su mano izquierda e interacciona con su mano derecha (pulsando el botón <i>Grip</i>) en el botón del menú referido a la opción rotación por pulsos (opción que se encuentra desactivada). La opción <i>rotar con joystick</i> esta activada.
Postcondición	La opción de rotar con por pulsos se encuentra activada.

ID: 5.3.1	
Resultado esperado	La opción <i>Rotar por pulsos</i> es activada (su conmutador se ilumina en verde) y el usuario es capaz de rotar la cámara horizontalmente con el stick derecho, pero el movimiento se hace a toques de Xº cada vez que el usuario mueve el joystick.
Resultado obtenido	Resultado esperado

ID: 5.3.2	
Descripción	Interacción del menú en modo inmersivo (Desactivar rotación por pulsos)
Precondición	El usuario ha activado el modo inmersivo y visualiza el contenido en sus gafas (conectadas o <i>standalone</i>). Esta visualizando el menú en su mano izquierda e interacciona con su mano derecha (pulsando el botón <i>Grip</i>) en el botón del menú referido a la opción <i>Rotar por pulsos</i> (opción que se encuentra activada). La opción <i>rotar con joystick</i> esta activada.
Postcondición	La opción de rotar por pulsos se encuentra desactivada.
Resultado esperado	La opción <i>rotar por pulsos</i> es desactivada (su conmutador se ilumina en rojo) y el usuario es capaz de rotar la cámara horizontalmente con el stick derecho, el movimiento se hace de forma fluida si el usuario mantiene el stick hacia un lado.
Resultado obtenido	Resultado esperado

ID: 5.4.1	
Descripción	Interacción del menú en modo inmersivo (Activar teletransporte)
Precondición	El usuario ha activado el modo inmersivo y visualiza el contenido en sus gafas (conectadas o <i>standalone</i>). Esta visualizando el menú en su mano izquierda e interacciona con su mano derecha (pulsando el botón <i>Grip</i>) en el botón del menú referido a la opción <i>Activar teletransporte</i> (opción que se encuentra desactivada).
Postcondición	La opción de <i>Activar teletransporte</i> se encuentra activada.
Resultado esperado	La opción <i>Activar teletransporte</i> es activada (su conmutador se ilumina en verde) y el usuario es capaz de invocar el teletransporte en su mano derecha (manteniendo pulsado el botón <i>Trigger</i> de dicha mano).
Resultado obtenido	Resultado esperado

ID: 5.4.2	
Descripción	Interacción del menú en modo inmersivo (Desactivar teletransporte)
Precondición	El usuario ha activado el modo inmersivo y visualiza el contenido en sus gafas (conectadas o <i>standalone</i>). Esta visualizando el menú en su mano izquierda e interacciona con su mano derecha (pulsando el botón <i>Grip</i>) en el botón del menú referido a la opción <i>Activar teletransporte</i> (opción que se encuentra activada).
Postcondición	La opción de <i>Activar teletransporte</i> se encuentra desactivada.
Resultado esperado	La opción <i>Activar teletransporte</i> es desactivada (su conmutador se ilumina en rojo) y el usuario no es capaz de invocar el teletransporte en su mano derecha (manteniendo pulsado el botón <i>Trigger</i>).
Resultado obtenido	Resultado esperado

ID: 5.5	
Descripción	Acceso a la aplicación (no inmersivo)
Precondición	El usuario ha accedido a la aplicación web y se encuentra en el menú principal. Clica en el botón <i>Comenzar</i> .
Postcondición	Se realiza un cambio de escena y el usuario pasa a encontrarse en la sala principal.
Resultado esperado	El usuario es teletransportado a la sala principal donde el estado se mantiene y visualiza el puntero en el centro de la pantalla. Puede desplazarse con WASD (o flechas) y mover la cámara con el ratón.
Resultado obtenido	Resultado esperado

ID: 5.5.1	
Descripción	Acceso a la aplicación (inmersivo)
Precondición	El usuario ha accedido a la aplicación web y se encuentra visualizando el contenido en su visor de realidad virtual (conectado o <i>standalone</i>). El usuario pulsa el botón X en su mano izquierda.
Postcondición	Se realiza un cambio de escena y el usuario pasa a encontrarse en la sala principal.
Resultado esperado	El usuario es teletransportado a la sala principal donde sigue en modo inmersivo, visualiza sus manos y puede desplazarse tal y como hacía en la escena del menú.

ID: 5.5.1	
Resultado obtenido	Resultado esperado

ID: 5.5.2	
Descripción	Precipitación al vacío (inmersivo)
Precondición	El usuario ha accedido a la aplicación web y se encuentra visualizando el contenido en su visor de realidad virtual (conectado o <i>standalone</i>). El usuario se desplaza hasta el borde del escenario y se precipita al vacío.
Postcondición	El usuario vuelve al punto inicial.
Resultado esperado	El usuario se precipita, pero vuelve al punto inicial.
Resultado obtenido	Resultado esperado

ID: 5.6	
Descripción	Abrir menú en la aplicación (no inmersivo)
Precondición	El usuario ha accedido a la aplicación web y ya ha superado el menú principal. Se encuentra en modo no inmersivo y el usuario pulsa el botón <i>Shift</i> para mostrar el menú.
Postcondición	El tiempo se detiene y el menú de configuración UI es mostrado en pantalla, se libera el ratón del usuario.
Resultado esperado	Se muestra el menú de configuración y se puede interaccionar con la interfaz utilizando el ratón a modo de clic.
Resultado obtenido	Resultado esperado

ID: 5.6.1	
Descripción	Cerrar menú en la aplicación (no inmersivo, botón).
Precondición	El usuario ha accedido a la aplicación web y ya ha superado el menú principal. Se encuentra visualizando el menú en modo no inmersivo. El usuario pulsa en el botón reanudar.
Postcondición	El tiempo se reanuda y el menú de configuración UI se oculta, el ratón del usuario vuelve a bloquearse en el centro de la pantalla.
Resultado esperado	Se bloquea el ratón en el centro de la pantalla y el menú de configuración desaparece.

ID: 5.6.1	
Resultado obtenido	Resultado esperado

ID: 5.6.2	
Descripción	Cerrar menú en la aplicación (no inmersivo, teclado).
Precondición	El usuario ha accedido a la aplicación web y ya ha superado el menú principal. Se encuentra visualizando el menú en modo no inmersivo. El usuario pulsa la tecla <i>Shift</i> del teclado.
Postcondición	El tiempo se reanuda y el menú de configuración UI se oculta, el ratón del usuario vuelve a bloquearse en el centro de la pantalla.
Resultado esperado	Se bloquea el ratón en el centro de la pantalla y el menú de configuración desaparece.
Resultado obtenido	Resultado esperado

ID: 5.6.2	
Descripción	Modificar sensibilidad del ratón (no inmersivo).
Precondición	El usuario ha accedido a la aplicación web y ya ha superado el menú principal. Ha abierto el menú donde visualiza la información y modificado el parámetro de sensibilidad de ratón, ha vuelto a cerrar el menú.
Postcondición	El parámetro de sensibilidad del ratón se encuentra modificado en las posteriores aberturas del menú.
Resultado esperado	El usuario modifica la sensibilidad del ratón y el cambio se ve reflejado en los movimientos de la cámara en modo no inmersivo.
Resultado obtenido	Resultado esperado

ID: 6	
Descripción	Cambio a modo inmersivo en ejecución
Precondición	El usuario ha accedido a la aplicación en modo no inmersivo y ha superado el menú principal. Se encuentra en cualquier escena y ya se ha desplazado. Tiene Hardware de realidad virtual conectado al equipo y realiza el cambio a modo inmersivo.
Postcondición	La aplicación ahora es cargada en las gafas de realidad virtual y no en el navegador. El usuario ahora interacciona con el entorno en realidad virtual.

ID: 6	
Resultado esperado	El contenido se carga en las gafas de realidad virtual. Se muestran las manos del usuario y se permite la interacción inmersiva moviendo la cámara con la cabeza y las manos con los controladores. El usuario es teletransportado al punto de inicio.
Resultado obtenido	Resultado esperado

ID: 6.1	
Descripción	Cambio a modo no inmersivo en ejecución
Precondición	El usuario ha accedido a la aplicación en modo inmersivo con el visor conectado al equipo y ha superado el menú principal. Se encuentra en cualquier escena y ya se ha desplazado.
Postcondición	La aplicación ahora es visualizada en el navegador en vez de en el visor. El usuario ahora interactúa con el entorno con ratón y teclado.
Resultado esperado	El contenido se carga en la ventana del navegador y se ocultan las manos del usuario. El usuario es teletransportado al punto de inicio.
Resultado obtenido	Resultado esperado

ID: 6.1	
Descripción	Cambio a modo no inmersivo en ejecución (Oculus Browser)
Precondición	El usuario ha accedido a la aplicación en modo inmersivo desde el navegador de las gafas (<i>standalone</i>) y ha superado el menú principal. Se encuentra en cualquier escena y ya se ha desplazado.
Postcondición	La aplicación ahora es visualizada en el navegador del visor, pero sin interacción inmersiva. El usuario ahora puede rotar la cámara a golpes de "ratón".
Resultado esperado	El contenido se carga en la ventana del navegador y se ocultan las manos del usuario. El usuario es teletransportado al punto de inicio.
Resultado obtenido	Resultado esperado

9.2 Pruebas de interacción

A continuación, se listan las tablas que contienen los casos de prueba para la

interacción con objetos del escenario.

En todas estas pruebas **se entiende que el usuario ha superado la escena del menú** y se encuentra en cualquiera del resto de escenas.

ID: 7	
Descripción	Salto (no inmersivo)
Precondición	El usuario ha accedido a la aplicación en modo no inmersivo. Pulsa el botón espacio mientras se desplaza sobre el suelo.
Postcondición	-
Resultado esperado	El usuario realiza un salto (desplazándose en el eje positivo de las Y).
Resultado obtenido	Resultado esperado

ID: 7.1	
Descripción	Salto (inmersivo)
Precondición	El usuario ha accedido a la aplicación en modo inmersivo. Mientras se mueve por el suelo pulsa el botón X de su mano izquierda.
Postcondición	-
Resultado esperado	El usuario realiza un salto (desplazándose en el eje positivo de las Y).
Resultado obtenido	Resultado esperado

ID: 7.2	
Descripción	Salto (en el aire)
Precondición	El usuario ha accedido a la aplicación e, independientemente del estado, intenta realizar un salto mientras su velocidad en el eje de las Y no es 0 (se encuentra saltando o cayendo).
Postcondición	-
Resultado esperado	El salto no se lleva a cabo.
Resultado obtenido	Resultado esperado

ID: 8	
Descripción	Interacción simple (no inmersiva) - Agarrar
Precondición	El usuario ha accedido a la aplicación en modo no inmersivo. Hace clic izquierdo sobre un objeto marcado como <i>interactivo</i> y mantiene el clic pulsado.
Postcondición	El objeto pasa a mantenerse frente a la cámara del usuario en la ubicación del puntero.
Resultado esperado	El usuario agarra el objeto y lo puede desplazar por el escenario.
Resultado obtenido	Resultado esperado

ID: 8.1	
Descripción	Interacción simple (no inmersiva) - Soltar
Precondición	El usuario ha accedido a la aplicación en modo no inmersivo. Tiene un objeto agarrado en sus manos y el clic izquierdo pulsado. El usuario suelta el clic izquierdo.
Postcondición	El objeto deja de mantenerse frente a la cámara del usuario y cae al suelo.
Resultado esperado	El objeto agarrado cae al suelo y colisiona con este. Se emite un sonido.
Resultado obtenido	Resultado esperado

ID: 8.2	
Descripción	Interacción simple (no inmersiva) – Soltar con movimiento
Precondición	El usuario ha accedido a la aplicación en modo no inmersivo. Tiene un objeto agarrado en sus manos y el clic izquierdo pulsado. El usuario suelta el clic izquierdo a la vez que mueve la cámara.
Postcondición	El objeto deja de mantenerse frente a la cámara del usuario y cae al suelo aplicando una fuerza.
Resultado esperado	El objeto agarrado es lanzado por el usuario, cae al suelo y colisiona con este emitiendo un sonido.
Resultado obtenido	Resultado esperado

ID: 8.3	
Descripción	Interacción simple (no inmersiva) – Lanzar

ID: 8.3	
Precondición	El usuario ha accedido a la aplicación en modo no inmersivo. Tiene un objeto agarrado en sus manos y el clic izquierdo pulsado. El usuario presiona el clic derecho.
Postcondición	El objeto deja de mantenerse frente a la cámara del usuario y es lanzado con fuerza.
Resultado esperado	El objeto agarrado es lanzado aplicando una fuerza en la dirección donde la cámara está mirando. El objeto emite un sonido al colisionar.
Resultado obtenido	Resultado esperado

ID: 9	
Descripción	Interacción simple (inmersiva) - Agarrar
Precondición	El usuario ha accedido a la aplicación en modo inmersivo. Agarra con una de sus manos un objeto pulsando el botón <i>Grip</i> , mantiene pulsado el botón.
Postcondición	El objeto pasa a mantenerse en la mano que lo ha agarrado.
Resultado esperado	El usuario agarra el objeto y lo puede desplazar por el escenario.
Resultado obtenido	Resultado esperado

ID: 9.1	
Descripción	Interacción simple (inmersiva) - Soltar
Precondición	El usuario ha accedido a la aplicación en modo inmersivo. Tiene agarrado en una de sus manos un objeto manteniendo pulsado el botón <i>Grip</i> de dicha mano, suelta el botón.
Postcondición	El objeto deja de mantenerse en la mano y cae al suelo
Resultado esperado	El objeto agarrado cae al suelo y colisiona con este. Se emite un sonido.
Resultado obtenido	Resultado esperado

ID: 9.2	
Descripción	Interacción simple (inmersiva) – Soltar con movimiento

ID: 9.2	
Precondición	El usuario ha accedido a la aplicación en modo inmersivo. Tiene agarrado en una de sus manos un objeto manteniendo pulsado el botón <i>Grip</i> de dicha mano, suelta el botón a la vez que mueve la mano.
Postcondición	El objeto deja de mantenerse en la mano y cae al suelo aplicando una fuerza en la dirección del movimiento de la mano.
Resultado esperado	El objeto agarrado es lanzado por el usuario, cae al suelo y colisiona con este emitiendo un sonido.
Resultado obtenido	Resultado esperado

ID: 9.3	
Descripción	Interacción simple (inmersiva) – Lanzar
Precondición	El usuario ha accedido a la aplicación en modo inmersivo. Tiene agarrado en una de sus manos un objeto manteniendo pulsado el botón <i>Grip</i> de dicha mano. El usuario presiona el botón X o A (según la mano que tenga agarrado el objeto)
Postcondición	El objeto deja de mantenerse en la mano y es lanzado con fuerza.
Resultado esperado	El objeto agarrado es lanzado aplicando una fuerza en la dirección de la mano que lo estaba agarrando. El objeto emite un sonido al colisionar.
Resultado obtenido	Resultado esperado

ID: 9.4	
Descripción	Interacción simple (inmersiva) – Cambio de mano
Precondición	El usuario ha accedido a la aplicación en modo inmersivo. Tiene agarrado en una de sus manos un objeto manteniendo pulsado el botón <i>Grip</i> de dicha mano, acerca la otra mano y pulsa el botón <i>Grip</i> junto al objeto.
Postcondición	El objeto deja de mantenerse en la mano anterior y pasa a mantenerse sobre la nueva mano.
Resultado esperado	El objeto cambia de mano.
Resultado obtenido	Resultado esperado

ID: 10	
Descripción	Traslación (no inmersiva)
Precondición	El usuario ha accedido a la aplicación en modo no inmersivo y presiona las teclas de dirección.
Postcondición	El usuario se encuentra en una posición diferente.
Resultado esperado	El usuario se desplaza por el entorno tomando como dirección "hacia adelante" el lugar hacia donde enfoca la cámara. Se reproducen sonidos de paso al desplazarse y estos cesan al detenerse.
Resultado obtenido	Resultado esperado

ID: 10.1	
Descripción	Traslación (inmersiva)
Precondición	El usuario ha accedido a la aplicación en modo inmersivo, dispone de las opciones de movimiento y ro <i>sticks</i> activadas. Mueve los sticks.
Postcondición	El usuario se encuentra en una posición diferente.
Resultado esperado	El usuario se desplaza por el entorno tomando como dirección "hacia adelante" el lugar hacia donde enfoca la cámara. Se reproducen sonidos de paso al desplazarse y estos cesan al detenerse.
Resultado obtenido	Resultado esperado

ID: 11	
Descripción	Teletransporte (inmersiva)
Precondición	El usuario ha accedido a la aplicación en modo inmersivo, dispone de la opción de Activar teletransporte activada. Pulsa el botón de teletransporte, enfoca hacia un lugar del escenario y lo suelta.
Postcondición	El usuario se encuentra en una posición diferente.
Resultado esperado	El usuario se teletransporta al lugar donde había apuntado con el teletransporte.
Resultado obtenido	Resultado esperado

ID: 11.1	
Descripción	Teletransporte (inmersiva) - Error

ID: 11.1	
Precondición	El usuario ha accedido a la aplicación en modo inmersivo, dispone de la opción de Activar teletransporte activada. Pulsa el botón de teletransporte, enfoca hacia un lugar del escenario al que no es posible acceder (paredes, techos, exterior del escenario, objetos interactivos, etc.).
Postcondición	El usuario se encuentra en la misma posición anterior.
Resultado esperado	El teletransporte desaparece, pero el usuario no se desplaza.
Resultado obtenido	Resultado esperado

9.2.1 Interacción en la sala principal

A continuación, se detallan las pruebas de interacción específica referida a la sala principal y a todas sus salas, por tanto, se entiende que en estos casos de prueba el usuario **se encuentra en la sala principal**.

ID: 12	
Descripción	Despliegue del panel de información (no inmersivo)
Precondición	El usuario clica, en modo no VR, sobre el interruptor que habilita el panel de información del vestíbulo, en la sala principal.
Postcondición	El panel se mantiene desplegado y puede pasarse de página.
Resultado esperado	El panel se despliega siguiendo una animación y reproduciendo el sonido pertinente.
Resultado obtenido	Resultado esperado

ID: 12.1	
Descripción	Despliegue del panel de información (inmersivo)
Precondición	El usuario interactúa en modo VR con el interruptor que habilita el panel de información del vestíbulo, en la sala principal.
Postcondición	El panel se mantiene desplegado y puede pasarse de página.
Resultado esperado	El panel se despliega siguiendo una animación y reproduciendo el sonido pertinente.
Resultado obtenido	Resultado esperado

ID: 12.2	
Descripción	Avanzar página del panel de información (no inmersivo)
Precondición	El panel del vestíbulo se encuentra desplegado y el usuario se encuentra en modo no inmersivo. Clica sobre el botón de pasar página
Postcondición	El panel se mantiene desplegado y se visualiza la página siguiente. El numerador de página se mantiene actualizado
Resultado esperado	El panel cambia su contenido siguiendo una animación y el botón reproduce el sonido pertinente.
Resultado obtenido	Resultado esperado

ID: 12.2.1	
Descripción	Retroceder página del panel de información (no inmersivo)
Precondición	El panel del vestíbulo se encuentra desplegado y el usuario se encuentra en modo no inmersivo. Clica sobre el botón de retroceder página
Postcondición	El panel se mantiene desplegado y se visualiza la página anterior. El numerador de página se mantiene actualizado
Resultado esperado	El panel cambia su contenido siguiendo una animación y el botón reproduce el sonido pertinente.
Resultado obtenido	Resultado esperado

ID: 12.2.2	
Descripción	Avanzar página del panel de información excede máximo (no VR)
Precondición	El panel del vestíbulo se encuentra desplegado y el usuario se encuentra en modo no inmersivo. Clica sobre el botón de avanzar página cuando ya se encuentra en la última página.
Postcondición	El panel se mantiene desplegado y se visualiza la misma página. El numerador de página se mantiene actualizado
Resultado esperado	El botón reproduce el sonido pertinente y el contenido no se modifica.
Resultado obtenido	Resultado esperado

ID: 12.2.3	
Descripción	Avanzar página del panel de información excede mínimo (no VR)
Precondición	El panel del vestíbulo se encuentra desplegado y el usuario se encuentra en modo no inmersivo. Clica sobre el botón de retroceder página cuando ya se encuentra en la primera página.
Postcondición	El panel se mantiene desplegado y se visualiza la misma página. El numerador de página se mantiene actualizado
Resultado esperado	El botón reproduce el sonido pertinente y el contenido no se modifica.
Resultado obtenido	Resultado esperado

ID: 12.3	
Descripción	Avanzar página del panel de información (inmersivo)
Precondición	El panel del vestíbulo se encuentra desplegado y el usuario se encuentra en modo inmersivo. Pulsa el botón de pasar página
Postcondición	El panel se mantiene desplegado y se visualiza la página siguiente. El numerador de página se mantiene actualizado
Resultado esperado	El panel cambia su contenido siguiendo una animación y el botón reproduce el sonido pertinente.
Resultado obtenido	Resultado esperado

ID: 12.3.1	
Descripción	Retroceder página del panel de información (inmersivo)
Precondición	El panel del vestíbulo se encuentra desplegado y el usuario se encuentra en modo inmersivo. Pulsa el botón de retroceder página
Postcondición	El panel se mantiene desplegado y se visualiza la página anterior. El numerador de página se mantiene actualizado
Resultado esperado	El panel cambia su contenido siguiendo una animación y el botón reproduce el sonido pertinente.
Resultado obtenido	Resultado esperado

ID: 12.3.2	
Descripción	Avanzar página del panel de información excede máximo (VR)

ID: 12.3.2	
Precondición	El panel del vestíbulo se encuentra desplegado y el usuario se encuentra en modo inmersivo. Pulsa el botón de avanzar página cuando ya se encuentra en la última página.
Postcondición	El panel se mantiene desplegado y se visualiza la misma página. El numerador de página se mantiene actualizado
Resultado esperado	El botón reproduce el sonido pertinente y el contenido no se modifica.
Resultado obtenido	Resultado esperado

ID: 12.3.3	
Descripción	Avanzar página del panel de información excede mínimo (VR)
Precondición	El panel del vestíbulo se encuentra desplegado y el usuario se encuentra en modo inmersivo. Pulsa el botón de retroceder página cuando ya se encuentra en la primera página.
Postcondición	El panel se mantiene desplegado y se visualiza la misma página. El numerador de página se mantiene actualizado
Resultado esperado	El botón reproduce el sonido pertinente y el contenido no se modifica.
Resultado obtenido	Resultado esperado

ID: 13	
Descripción	Instanciación de pelota
Precondición	El usuario se encuentra en el patio de la escena principal y activa el botón de invocar pelota (tanto en VR como en no VR).
Postcondición	La pelota se encuentra en la base para pelotas.
Resultado esperado	Se reproduce el sonido pertinente, la pelota es instanciada y propulsada terminando en la base para pelotas.
Resultado obtenido	Resultado esperado

ID: 13.1	
Descripción	Lanzamiento a canasta (fallo)
Precondición	El usuario se encuentra en el patio de la escena principal y lanza un objeto interactivo hacia la canasta, no encesta.

ID: 13.1	
Postcondición	La puntuación se actualiza y aumenta el número de disparos.
Resultado esperado	Se actualiza la puntuación y el disparo no cuenta como acertado.
Resultado obtenido	Resultado esperado

ID: 13.2	
Descripción	Lanzamiento a canasta (acierto)
Precondición	El usuario se encuentra en el patio de la escena principal y lanza un objeto interactivo hacia la canasta y encesta.
Postcondición	La puntuación se actualiza y aumenta el número de disparos y de aciertos.
Resultado esperado	Se actualiza la puntuación y el disparo cuenta como acertado. Se reproduce el audio de acierto.
Resultado obtenido	Resultado esperado

ID: 14	
Descripción	Apertura de puerta
Precondición	El usuario agarra el pomo de la puerta (en modo inmersivo o no) y lo desplaza.
Postcondición	La puerta se abre.
Resultado esperado	La puerta se ve atraída por los movimientos del pomo y se abre haciendo uso de las visagras.
Resultado obtenido	Resultado esperado

ID: 14.1	
Descripción	Apertura de puerta (Rotura de interacción)
Precondición	El usuario agarra el pomo de la puerta (en modo inmersivo o no) y lo desplaza, se aleja de la puerta trasladándose aun sujetando el pomo.
Postcondición	El usuario ya no interactúa con la puerta.
Resultado esperado	La interacción se rompe cuando el usuario se aleja de la puerta.

ID: 14.1	
Resultado obtenido	Resultado esperado

ID: 14.2	
Descripción	Apertura de puerta (Lanzamiento)
Precondición	El usuario agarra el pomo de la puerta (en modo inmersivo o no) y activa la opción de lanzar.
Postcondición	-
Resultado esperado	No sucede nada, no es posible lanzar la puerta.
Resultado obtenido	Resultado esperado

ID: 15	
Descripción	Teletransporte a sala de eventos
Precondición	El usuario se mueve hasta acceder a la sala de eventos desde la sala principal.
Postcondición	El usuario se encuentra en la sala de eventos.
Resultado esperado	Se realiza el cambio de escena y el usuario pasa a encontrarse en la sala de eventos.
Resultado obtenido	Resultado esperado

ID: 15.1	
Descripción	Teletransporte a sala de instrumentos
Precondición	El usuario se mueve hasta acceder a la sala de instrumentos desde la sala principal.
Postcondición	El usuario se encuentra en la sala de instrumentos.
Resultado esperado	Se realiza el cambio de escena y el usuario pasa a encontrarse en la sala de instrumentos.
Resultado obtenido	Resultado esperado

ID: 15.2	
Descripción	Teletransporte a sala asturiana
Precondición	El usuario se mueve hasta acceder a la sala asturiana desde la sala principal.
Postcondición	El usuario se encuentra en la sala asturiana.
Resultado esperado	Se realiza el cambio de escena y el usuario pasa a encontrarse en la sala asturiana.
Resultado obtenido	Resultado esperado

ID: 15.3	
Descripción	Teletransporte a sala de teoría y juego.
Precondición	El usuario se mueve hasta acceder a la sala de teoría y juego desde la sala principal.
Postcondición	El usuario se encuentra en la sala de teoría y juego.
Resultado esperado	Se realiza el cambio de escena y el usuario pasa a encontrarse en la sala de teoría y juego.
Resultado obtenido	Resultado esperado

9.2.2 Interacción en sala secundaria

A continuación, se detallan las pruebas de interacción específica referida a la sala principal y a todas sus salas, por tanto, se entiende que en estos casos de prueba el usuario **se encuentra en una de las salas secundarias**. Si se da el caso de probar la funcionalidad de un objeto específico que solo se encuentra en una sala concreta, se indicará en la Precondición del caso de prueba.

ID: 16	
Descripción	Interacción con panel de texto (no inmersivo)
Precondición	El usuario se encuentra en modo no inmersivo y hace clic sobre uno de los paneles de textos interactivos que muestran información sobre el escenario.
Postcondición	El contenido pasa a estar activado y se encuentra desplegado.

ID: 16	
Resultado esperado	El contenido se activa y aparece siguiendo una animación.
Resultado obtenido	Resultado esperado

ID: 16.1	
Descripción	Interacción con panel de texto (inmersivo)
Precondición	El usuario se encuentra en modo inmersivo y pulsa sobre uno de los paneles de textos interactivos que muestran información sobre el escenario.
Postcondición	El contenido pasa a estar activado y se encuentra desplegado.
Resultado esperado	El contenido se activa y aparece siguiendo una animación.
Resultado obtenido	Resultado esperado

ID: 16.2	
Descripción	Texto interactivo desplegado
Precondición	El usuario se encuentra ha activado un panel de texto interactivo y se encuentra visualizándolo.
Postcondición	El contenido sigue activado
Resultado esperado	El contenido se mira permanentemente al usuario, haciendo que siempre este frente a él.
Resultado obtenido	Resultado esperado

ID: 16.3	
Descripción	Interacción con panel de texto (no inmersivo) – Cerrar
Precondición	El usuario se encuentra en modo no inmersivo y hace clic sobre uno de los paneles de textos interactivos que muestran información sobre el escenario, el panel ya se encuentra ya activado y desplegado.
Postcondición	El contenido pasa a estar desactivado y se encuentra oculto.
Resultado esperado	El contenido se desactiva y desaparece siguiendo una animación.

ID: 16.3	
Resultado obtenido	Resultado esperado

ID: 16.4	
Descripción	Interacción con panel de texto (inmersivo) - Cerrar
Precondición	El usuario se encuentra en modo inmersivo y pulsa sobre uno de los paneles de textos interactivos que muestran información sobre el escenario, el panel ya se encuentra ya activado y desplegado.
Postcondición	El contenido pasa a estar desactivado y se encuentra oculto.
Resultado esperado	El contenido se desactiva y desaparece siguiendo una animación.
Resultado obtenido	Resultado esperado

ID: 16.5	
Descripción	Interacción con panel de texto con otro panel activado.
Precondición	El usuario interactúa con un panel interactivo de texto mientras ya hay desplegado un panel de texto interactivo.
Postcondición	El contenido del panel anterior permanece oculto y el del nuevo panel desplegado.
Resultado esperado	El panel antiguo se oculta siguiendo una animación y el nuevo panel se activa siguiendo otra.
Resultado obtenido	Resultado esperado

ID: 17	
Descripción	Interacción con panel de texto con audio
Precondición	El usuario ha activado uno de los paneles de texto interactivo que también contienen audio en forma de botones. Interacciona con uno de los botones (que tienen asignado un audio).
Postcondición	El audio asignado se reproduce.
Resultado esperado	El audio se reproduce utilizando como fuente de audio el panel.
Resultado obtenido	Resultado esperado

ID: 17.1	
Descripción	Interacción con panel de texto con audio (Pulsar el mismo botón)
Precondición	El usuario ha activado uno de los paneles de texto interactivo que también contienen audio en forma de botones. Interacciona por segunda vez consecutiva con un botón (con audio asignado que ya se encuentra reproduciéndose).
Postcondición	El audio asignado se detiene.
Resultado esperado	La fuente de audio se detiene.
Resultado obtenido	Resultado esperado

ID: 17.2	
Descripción	Interacción con panel de texto con audio (Pulsar otro botón)
Precondición	El usuario ha activado uno de los paneles de texto interactivo que también contienen audio en forma de botones. Interacciona con un botón diferente del panel mientras se reproduce el audio de otro botón.
Postcondición	El audio anterior se detiene y el nuevo está reproduciéndose.
Resultado esperado	La fuente de audio cambia el clip de audio y se reproduce.
Resultado obtenido	Resultado esperado

ID: 17.3	
Descripción	Interacción con panel de texto con audio (desactivar panel)
Precondición	El usuario ha activado uno de los paneles de texto interactivo que también contienen audio en forma de botones. Ha interactuado con un botón y su audio se está reproduciendo, el usuario pliega el panel en cuestión (bien interactuando con él o bien interactuando con otro).
Postcondición	El audio ya no se encuentra reproduciéndose.
Resultado esperado	La fuente de audio se detiene.
Resultado obtenido	Resultado esperado

ID: 18	
Descripción	Lanzamiento de cinta de teoría musical
Precondición	El usuario se encuentra en la sala de teoría y juego, donde agarra una cinta de teoría musical que se encuentra al lado de un objeto interactivo de panel de texto de teoría. Con la cinta en la mano pulsa el botón de lanzar (tanto inmersivo como no).
Postcondición	La cinta permanece agarrada por el usuario.
Resultado esperado	Se muestra una descripción sobre la cinta de vídeo que mira permanentemente al usuario. La cinta no es lanzada.
Resultado obtenido	Resultado esperado

ID: 19	
Descripción	Colocación de cinta de teoría en reproductor
Precondición	El usuario se encuentra en la sala de teoría y juego, donde agarra una cinta de teoría musical que se encuentra al lado de un objeto interactivo de panel de texto de teoría y la acerca al lector que se encuentra en la sala de juego.
Postcondición	El usuario ya no agarra la cinta y los objetos musicales se iluminan siguiendo el patrón definido en esta.
Resultado esperado	La interacción con la cinta se rompe y esta pasa a mantenerse fijada en el lector. Los objetos musicales de la escala escogida se iluminan.
Resultado obtenido	Resultado esperado

ID: 19.1	
Descripción	Agarre de cinta de teoría del reproductor
Precondición	El usuario se encuentra en la sala de teoría y juego, donde ha colocado una cinta de teoría en el reproductor. El usuario agarra la cinta en cuestión.
Postcondición	El usuario agarra la cinta. Los objetos musicales ya no están iluminados.
Resultado esperado	Comienza la interacción con la cinta de teoría, que pasa a estar en manos del usuario. Los objetos musicales ya no se iluminan.
Resultado obtenido	Resultado esperado

ID: 19.2	
Descripción	Cambiar tonalidad de la escala introducida (+1)
Precondición	El usuario se encuentra en la sala de teoría y juego, donde ha colocado una cinta de teoría en el reproductor con tono Re. El usuario pulsa el botón de aumentar la tonalidad.
Postcondición	Los objetos iluminados son diferentes.
Resultado esperado	La tonalidad de la escala introducida cambia, pasando de mostrar iluminada la escala en tonalidad Mib. También cambia el tono en la pantalla del reproductor.
Resultado obtenido	Resultado esperado

ID: 19.3	
Descripción	Cambiar tonalidad de la escala introducida (-1)
Precondición	El usuario se encuentra en la sala de teoría y juego, donde ha colocado una cinta de teoría en el reproductor con tono Re. El usuario pulsa el botón de disminuir la tonalidad.
Postcondición	Los objetos iluminados son diferentes.
Resultado esperado	La tonalidad de la escala introducida cambia, pasando de mostrar iluminada la escala en tonalidad Reb. También cambia el tono en la pantalla del reproductor.
Resultado obtenido	Resultado esperado

ID: 19.4	
Descripción	Cambiar tonalidad de la escala introducida (Max +1)
Precondición	El usuario se encuentra en la sala de teoría y juego, donde ha colocado una cinta de teoría en el reproductor con tono Si (nota máxima). El usuario pulsa el botón de aumentar la tonalidad.
Postcondición	Los objetos iluminados son diferentes.
Resultado esperado	La tonalidad de la escala introducida cambia, pasando de mostrar iluminada la escala en tonalidad Do. También cambia el tono en la pantalla del reproductor.
Resultado obtenido	Resultado esperado

ID: 19.5	
Descripción	Cambiar tonalidad de la escala introducida (Min -1)

ID: 19.5	
Precondición	El usuario se encuentra en la sala de teoría y juego, donde ha colocado una cinta de teoría en el reproductor con tono Do (nota máxima). El usuario pulsa el botón de disminuir la tonalidad.
Postcondición	Los objetos iluminados son diferentes.
Resultado esperado	La tonalidad de la escala introducida cambia, pasando de mostrar iluminada la escala en tonalidad Si. También cambia el tono en la pantalla del reproductor.
Resultado obtenido	Resultado esperado

ID: 19.6	
Descripción	Cambiar tonalidad sin escala
Precondición	El usuario se encuentra en la sala de teoría y juego, no hay ninguna cinta en el reproductor de cintas de teoría. El usuario pulsa un botón de cambio de tonalidad.
Postcondición	La pantalla muestra una tonalidad diferente, los objetos no están iluminados.
Resultado esperado	La tonalidad varía, pero no se ilumina ningún objeto.
Resultado obtenido	Resultado esperado

ID: 20	
Descripción	Agarre de baqueta (modo inmersivo)
Precondición	El usuario se encuentra en la sala de teoría y juego, en modo inmersivo, donde agarra una baqueta con una de sus manos.
Postcondición	La baqueta se ajusta en la mano del jugador y este pasa a llevarla en su mano.
Resultado esperado	El usuario agarra la baqueta, pero la posición de esta se ajusta en la mano del jugador para un manejo más cómodo.
Resultado obtenido	Resultado esperado

ID: 20.1	
Descripción	Lanzar baqueta

ID: 20.1	
Precondición	El usuario se encuentra en la sala de teoría y juego, tanto en modo inmersivo como no, con una baqueta agarrada. El usuario ejecuta la opción de lanzar el objeto.
Postcondición	La baqueta se encuentra en su posición inicial
Resultado esperado	El usuario rompe la interacción con la baqueta, pero esta no es lanzada si no que se teletransporta al punto inicial donde se encontraba al comienzo de la escena.
Resultado obtenido	Resultado esperado

ID: 21	
Descripción	Objeto musical (interacción)
Precondición	El usuario se encuentra en la sala de teoría y juego, tanto en modo inmersivo como no, interacciona con un objeto musical.
Postcondición	-
Resultado esperado	El audio del objeto musical se reproduce.
Resultado obtenido	Resultado esperado

ID: 21.1	
Descripción	Objeto musical (golpe)
Precondición	El usuario se encuentra en la sala de teoría y juego, tanto en modo inmersivo como no, interacciona con un objeto musical golpeándolo con un objeto interactivo (e.g una baqueta).
Postcondición	-
Resultado esperado	El audio del objeto musical se reproduce.
Resultado obtenido	Resultado esperado

ID: 21.2	
Descripción	Objeto musical (disminuir pitch)
Precondición	El usuario se encuentra en la sala de teoría y juego, tanto en modo inmersivo como no, interacciona con el botón de disminuir pitch.

ID: 21.2	
Postcondición	Los objetos musicales ahora emiten el sonido con un tono más grave.
Resultado esperado	El pitch mostrado en pantalla se actualiza (disminuye 0.2 puntos) y los sonidos de los objetos musicales se reproduce en un tono más grave.
Resultado obtenido	Resultado esperado

ID: 21.3	
Descripción	Objeto musical (aumentar pitch)
Precondición	El usuario se encuentra en la sala de teoría y juego, tanto en modo inmersivo como no, interacciona con el botón de aumentar pitch.
Postcondición	Los objetos musicales ahora emiten el sonido con un tono más agudo.
Resultado esperado	El pitch mostrado en pantalla se actualiza (aumenta 0.2 puntos) y los sonidos de los objetos musicales se reproduce en un tono más agudo.
Resultado obtenido	Resultado esperado

ID: 21.3	
Descripción	Objeto musical (disminuir pitch en el valor mínimo)
Precondición	El usuario se encuentra en la sala de teoría y juego, tanto en modo inmersivo como no, interacciona con el botón de disminuir pitch cuando el valor de este es 0.2.
Postcondición	-
Resultado esperado	El pitch no disminuye.
Resultado obtenido	Resultado esperado

ID: 21.4	
Descripción	Objeto musical (aumentar pitch en el valor máximo)
Precondición	El usuario se encuentra en la sala de teoría y juego, tanto en modo inmersivo como no, interacciona con el botón de aumentar pitch cuando el valor de este es 3.
Postcondición	-

ID: 21.4	
Resultado esperado	El pitch no aumenta.
Resultado obtenido	Resultado esperado

9.3 Pruebas de vídeo

La reproducción de vídeos es un proceso muy importante que se lleva a cabo en la aplicación del proyecto, es por ello por lo que será probado exhaustivamente y por eso se mantiene en un apartado independiente.

Se entiende que **todas estas pruebas se llevan a cabo** una vez superado el menú y **con el usuario encontrándose en la sala de eventos**.

ID: 22	
Descripción	Instanciación de vídeos.
Precondición	El usuario accede al salón de eventos. Tanto el servicio de MinIO como el cliente externo se encuentran desplegados. El bucket de la base de datos de MinIO almacena 3 archivos de vídeo.
Postcondición	Se carga la escena y hay tres cintas de vídeo en el mostrador.
Resultado esperado	Se lanza la petición GET al cliente externo de MinIO, se obtiene la respuesta con el listado de los tres vídeos y se instancia una cinta de vídeo para cada uno de ellos cumpliendo los parámetros de posición y separación entre estas.
Resultado obtenido	Resultado esperado

ID: 22.1	
Descripción	Instanciación de vídeos (Añadir vídeo)
Precondición	El usuario accede al salón de eventos por segunda vez (tras cumplir el caso de prueba ID22). Tanto el servicio de MinIO como el cliente externo se encuentran desplegados. Se añade un nuevo vídeo por lo que el bucket de la base de datos de MinIO almacena 4 archivos de vídeo.
Postcondición	Se carga la escena y hay cuatro cintas de vídeo en el mostrador.

ID: 22.1	
Resultado esperado	Se lanza la petición GET al cliente externo de MinIO, se obtiene la respuesta con el listado de los cuatro vídeos y se instancia una cinta de vídeo para cada uno de ellos cumpliendo los parámetros de posición y separación entre estas.
Resultado obtenido	Resultado esperado

ID: 22.2	
Descripción	Instanciación de vídeos (Eliminar vídeo)
Precondición	El usuario accede al salón de eventos por tercera vez (tras cumplir el caso de prueba ID22.1). Tanto el servicio de MinIO como el cliente externo se encuentran desplegados. Se eliminan dos vídeos por lo que el bucket de la base de datos de MinIO almacena 2 archivos de vídeo.
Postcondición	Se carga la escena y hay dos cintas de vídeo en el mostrador.
Resultado esperado	Se lanza la petición GET al cliente externo de MinIO, se obtiene la respuesta con el listado de los dos vídeos y se instancia una cinta de vídeo para cada uno de ellos cumpliendo los parámetros de posición y separación entre estas.
Resultado obtenido	Resultado esperado

ID: 22.3	
Descripción	Instanciación de vídeos (MinIO no operativo)
Precondición	El usuario accede al salón de eventos. Se desconecta el servicio de MinIO.
Postcondición	Se carga la escena, no hay cintas de vídeo en el mostrador.
Resultado esperado	Se intenta lanzar la petición GET al cliente, pero no hay respuesta. La aplicación sigue su curso sin instanciar cintas de vídeo y el cliente de MinIO se mantiene operativo.
Resultado obtenido	Resultado esperado

ID: 22.4	
Descripción	Instanciación de vídeos (Cliente MinIO no operativo)
Precondición	El usuario accede al salón de eventos. Se desconecta el servicio del cliente externo de MinIO.

ID: 22.4	
Postcondición	Se carga la escena, no hay cintas de vídeo en el mostrador.
Resultado esperado	Se intenta lanzar la petición GET al cliente, pero no hay respuesta. La aplicación sigue su curso sin instanciar cintas de vídeo.
Resultado obtenido	Resultado esperado

ID: 23	
Descripción	Lanzamiento de cinta de video
Precondición	El usuario se encuentra en la sala de teoría y juego, donde agarra un vídeo que se encuentra. Con la cinta en la mano pulsa el botón de lanzar (tanto inmersivo como no).
Postcondición	La cinta permanece agarrada por el usuario.
Resultado esperado	Se muestra una descripción sobre la cinta de vídeo que mira permanentemente al usuario. La cinta no es lanzada.
Resultado obtenido	Resultado esperado

ID: 24	
Descripción	Colocación de cinta de vídeo en el proyector
Precondición	El usuario se encuentra en la sala agarra una cinta de vídeo y la aproxima a las ruedas superiores del proyector.
Postcondición	El usuario ya no agarra la cinta. La cinta se encuentra fijada en las ruedas del proyector.
Resultado esperado	La interacción con la cinta se rompe y esta pasa a mantenerse fijada en el proyector. El vídeo está listo para reproducirse
Resultado obtenido	Resultado esperado

ID: 24.1	
Descripción	Agarre de cinta de vídeo del proyector
Precondición	El usuario ha colocado una cinta de vídeo en el proyector. El usuario agarra la cinta en cuestión.
Postcondición	El usuario agarra la cinta.
Resultado esperado	Comienza la interacción con la cinta de vídeo, que pasa a estar en manos del usuario.

ID: 24.1	
Resultado obtenido	Resultado esperado

ID: 25	
Descripción	Despliegue de pantalla
Precondición	La pantalla del reproductor se encuentra plegada y la palanca en posición <i>off</i> . El usuario agarra la palanca y la coloca en posición <i>on</i> .
Postcondición	La pantalla se encuentra desplegada y la palanca en posición <i>on</i> .
Resultado esperado	Se reproduce el sonido pertinente y la pantalla del proyector se despliega haciendo uso de una animación.
Resultado obtenido	Resultado esperado

ID: 25.1	
Descripción	Pliegue de pantalla
Precondición	La pantalla del reproductor se encuentra desplegada y la palanca en posición <i>on</i> . El usuario agarra la palanca y la coloca en posición <i>off</i> .
Postcondición	La pantalla se encuentra plegada y la palanca en posición <i>off</i> .
Resultado esperado	Se reproduce el sonido pertinente y la pantalla del proyector se pliega haciendo uso de una animación.
Resultado obtenido	Resultado esperado

ID: 25.2	
Descripción	Palanca - Romper interacción
Precondición	El usuario agarra la palanca y se aleja de ella desplazándose en el escenario.
Postcondición	El usuario ya no se encuentra interactuando con la palanca.
Resultado esperado	La interacción con la palanca se rompe y esta pasa a la posición más cercana, activando el evento pertinente si se da el caso (<i>on</i> u <i>off</i>).
Resultado obtenido	Resultado esperado

ID: 26	
Descripción	Reproducir vídeo
Precondición	El usuario activa el botón de <i>Play</i> con una cinta colocada en el proyector y con la pantalla desplegada.
Postcondición	El usuario visualiza un vídeo en la pantalla.
Resultado esperado	Comienza a reproducirse el vídeo correspondiente a la cinta de vídeo y el botón pasa a <i>Stop</i> .
Resultado obtenido	Resultado esperado

ID: 26.1	
Descripción	Pausar vídeo
Precondición	El usuario activa el botón de <i>Stop</i> con una cinta colocada en el proyector y con la pantalla desplegada. El vídeo ya se encuentra reproduciéndose.
Postcondición	El vídeo ahora se congela.
Resultado esperado	Se pausa la reproducción del vídeo y el botón pasa a <i>Play</i> .
Resultado obtenido	Resultado esperado

ID: 26.2	
Descripción	Pausar vídeo – Quitar cinta
Precondición	El usuario se encuentra visualizando un video con una cinta colocada en el proyector y con la pantalla desplegada. El usuario agarra la cinta del proyector.
Postcondición	El vídeo ahora se congela.
Resultado esperado	Se pausa la reproducción del vídeo y el botón pasa a <i>Play</i> .
Resultado obtenido	Resultado esperado

ID: 26.3	
Descripción	Pausar vídeo – Plegar pantalla
Precondición	El usuario se encuentra visualizando un video con una cinta colocada en el proyector y con la pantalla desplegada. El usuario pliega la pantalla activando la palanca.

ID: 26.3	
Postcondición	El vídeo ahora se congela.
Resultado esperado	Se pausa la reproducción del vídeo y el botón pasa a <i>Play</i> .
Resultado obtenido	Resultado esperado

ID: 26.4	
Descripción	Reproducir vídeo con la pantalla plegada.
Precondición	El usuario ha colocado una cinta en el proyector, pero no ha desplegado la pantalla. Pulsa el botón de <i>Play</i> .
Postcondición	-
Resultado esperado	No sucede nada
Resultado obtenido	Resultado esperado

ID: 26.5	
Descripción	Reproducir vídeo sin cinta
Precondición	El usuario ha desplegado la pantalla, pero no ha colocado una cinta de vídeo. Pulsa el botón de <i>Play</i> .
Postcondición	-
Resultado esperado	No sucede nada
Resultado obtenido	Resultado esperado

ID: 26.5	
Descripción	Reproducir vídeo sin cinta y pantalla plegada
Precondición	El usuario pulsa el botón de <i>Play</i> sin ninguna cinta en el proyector y sin desplegar la pantalla
Postcondición	-
Resultado esperado	No sucede nada
Resultado obtenido	Resultado esperado

ID: 26.6	
Descripción	Reproducir vídeo, servicio de MinIO inactivo
Precondición	El usuario pulsa el botón de <i>Play</i> con una cinta en el proyector y con la pantalla desplegada. El servicio de MinIO no se encuentra operativo.
Postcondición	-
Resultado esperado	Se lanza la petición al vídeo, pero no se obtiene respuesta. La aplicación sigue su curso y no se reproduce el vídeo manteniendo el estado anterior.
Resultado obtenido	Resultado esperado

9.4 Arreglo de errores

Como resultado de una primera iteración de las pruebas anteriores se encontraron algunos errores que se debieron solucionar antes de realizar la entrega del trabajo:

1. Algunos objetos traspasaban el suelo al soltarlos, incluso a veces permanecían bajo el suelo y no se podían recoger.
 - a. Se descartó el uso de *MeshCollider* para los modelos desarrollados y se pasó a delimitarlos con *BoxColliders* más simples hechos a medida.
2. Era posible reproducir contenido sin desplegar pantalla y el estado del botón *PlayStop* no variaba si se desconectaba la cinta de vídeo.
3. El volumen del vídeo no variaba a la hora de reproducirlo.
 - a. Se tuvo que ajustar el volumen del *AudioListener* en vez del de *AudioSource* que emite el sonido (por restricciones por el uso de *WebGL*).
 - b. En algunos navegadores el error persiste.
4. Algunos objetos no se colocaban bien en la mano del usuario.

- a. Se ajustó el contenido del script *MoveToHandPosition*
5. El usuario se teletransportaba con la cámara de espaldas a otras escenas e invertía los controles en el movimiento.
 - a. Se modificó el método de teletransporte entre escenas para modificar la posición inicial y rotación.
6. Los textos de algunos objetos se mostraban girados 180°, hubo que ajustar su posición.
7. Había errores de acceso al listado de vídeos de MinIO.
 - a. Se tuvo que hacer público el acceso al resto de servicios del despliegue.
8. Existían desajustes en la altura del usuario cuando cambiaba de modo inmersivo o viceversa.
 - a. Ahora existe una altura diferente para cada modo de interacción.
9. Si el usuario cambiaba de modo a inmersiva o viceversa podía llegar a traspasarse el suelo o a existir desajustes en las posiciones de la cámara.
 - a. Se optó por reiniciar la posición del usuario cada vez que llevaba a cabo un modo en el estado XR.

Una vez solventados se realizó otra iteración de las pruebas dando los resultados plasmados en los anteriores subapartados.

10 Resultados y trabajo futuro

A continuación, se listará un breve resumen de los objetivos indicados al comienzo del presente proyecto junto con el aspecto del trabajo que hace que estos se cumplan;

Tabla 10.11. Objetivos cumplidos.

Objetivo	Conseguido	Aspecto del proyecto
Llevar a cabo el diseño Creación de un entorno 3D para la divulgación de diversos aspectos culturales, que sirva como entorno inmersivo para la visualización contenido interactivo orientado a la enseñanza en diversos ámbitos.	Si	Se ha modelado un escenario separado en diferentes salas que permiten la adición de contenido interactivo. Se han desarrollado los scripts y los <i>prefabs</i> u objetos que permiten el visionado y la divulgación de contenido interactivo sea cual sea su ámbito.
Permitir un acceso público a la aplicación y, si se detecta la conexión de hardware inmersivo permitir el acceso desde el mismo gracias al estándar WebXR.	Si	Se ha realizado una investigación sobre el estándar WebXR y se ha innovado haciendo uso de ella en un motor gráfico como es Unity. El acceso a la aplicación es abierto al público a través de internet y se permite su acceso a cualquier usuario independientemente de su situación. Se ha implementado el proyecto haciendo uso del estándar <i>WebXR</i> lo que nos permite detectar conexión con hardware inmersivo. Además, la forma de desarrollar la aplicación hace que esta también pueda ser utilizada sin utilizar este hardware y las pruebas respaldan el correcto funcionamiento en este aspecto.
Permitir la interacción con diferentes ámbitos, por ejemplo: Escalas y teoría musical básica, cultura asturiana, visualización de vídeos almacenados en nube, instrumentos, sus sonidos o papeles.	Si	El diseño del escenario separado en diversas salas independientes ofrece interacción con objetos relacionados con los ámbitos listados en el objetivo. Cada una de las salas es temática y cubre cada ámbito de la lista.
Divulgar la música y cultura asturiana en todos sus ámbitos culturales: instrumentos, bandas de gaitas, danza...	Si	Cómo se detalló en el objetivo anterior, existe una sala con objetos orientados única y exclusivamente a la cultura asturiana.

Objetivo	Conseguido	Aspecto del proyecto
Permitir el acceso a contactos de diferentes escuelas de música o conservatorios de la región para facilitar la obtención de más información por parte de los usuarios.	Si	En la sala principal existe una pantalla con menú desplegable donde se puede consultar diferentes conservatorios de la región y sus contactos.
La divulgación de contenido también conllevará la reproducción de vídeos que en todo momento debe ser dinámica y personalizada, permitiendo el visionado de un listado de vídeos sin tener que realizar cambios en el código o en el proyecto.	Si	Se permite la reproducción de vídeos bajo demanda almacenados en una base de datos externa. La alternativa utilizada se trata de <i>MinIO</i> , la adición de vídeos en dicha base de datos es sencilla y la actualización de los vídeos a reproducir se realiza de forma dinámica por lo que es sencillo para el administrador modificar el listado de vídeos sin hacer cambios en el código o despliegue. Por otro lado, esto ha permitido estudiar la integración de <i>WebXR</i> junto con otras tecnologías.

También es importante destacar que la entrega de este trabajo no supone un punto final en el desarrollo de la aplicación. El objetivo será continuar desarrollándolo para cumplir otra serie de objetivos que se listan a continuación:

- Adición de efectos de post-procesado.
- Completar el modelado 3D incluyendo modelos personalizados para cada uno de los objetos interactivos de las escenas.
- Finalizar la adición de contenido, incluyendo la reproducción de contenido personalizado para cada uno de los objetos interactivos.
- Contactar con instituciones culturales y/o musicales de la región para estudiar la posible implementación personalizada en centros educativos o escuelas de música.

11 Conclusiones

Tras finalizar un proyecto como el realizado durante todo este curso, resulta imprescindible realizar una reflexión y extraer unas **conclusiones** de todo el trabajo realizado. Se ha de tener en cuenta que todas las conclusiones plasmadas en el presente apartado se llevan a cabo desde un punto **subjetivo** del autor.

Realizar un trabajo de fin de estudios (Grado, Máster, etc.), ya sea un desarrollo o una investigación, resulta ser una tarea fundamental para lograr poner fin a una etapa de formación. A pesar de que esta titulación (Máster universitario en ingeniería informática) no es tan densa como la titulación de Grado, pues es tan solo de “un curso” de duración (un curso de asignaturas y otro aplicado únicamente a prácticas en empresa y al desarrollo del trabajo de fin de máster), desarrollar este trabajo sirve para aplicar contenidos vistos en la titulación y para ampliar otros nuevos.

La oportunidad que te brinda realizar un proyecto de estas características permite **aumentar la experiencia y madurar como ingeniero** por toda la gestión de trabajo y tiempo que lleva detrás este trabajo. Sobre todo, si se tiene en cuenta el haber estado trabajando a tiempo completo por cuenta ajena en la totalidad del desarrollo.

Respecto a los contenidos visualizados en el máster y aplicados en este proyecto podemos destacar:

- El uso de la tecnología **Docker** y despliegue de microservicios. Se permite incluso, modificando los parámetros de configuración pertinentes, el desplegar el presente proyecto como una aplicación distribuida.
- El estudio de alternativas **Cloud**.
- Control de versiones **Git** (Todo el desarrollo se ha ido almacenando en

un repositorio privado de **GitHub**).

- El despliegue de la aplicación con el uso de servicios **PaaS** (en este caso, *Heroku*).
- Gestión de **proyectos** y presupuestos.
- Estudio y ejecución de casos de **prueba**.
- Programación en **C#**, en especial para Unity VR.

Por otro lado, lo más llamativo es quizá el uso de un nuevo enfoque que da un vuelco a todo lo aprendido durante el Grado y el Máster: **El uso de un motor gráfico como Unity y el uso de programas de diseño 3D como Blender**. Nunca se había realizado nada relacionado con el 3D en la vida académica el alumno (tan solo en media asignatura del Máster: Computación gráfica y multimedia, que se basó en el desarrollo mediante *ThreeJS*) y menos aún con tecnologías inmersivas (nunca se había desarrollado en este campo ni se había utilizado hardware de este tipo), por lo que ha sido un **enorme desafío** entregar un proyecto que en su comienzo parecía muy ambicioso en tan solo un curso académico. Al final, con la entrega de este proyecto se han utilizado una gran variedad de herramientas y recursos de Unity:

- Scripting en **C#**.
- Luces.
- Físicas.
- Colisiones.
- Interacciones.
- UI.
- Interacción en realidad virtual.
- Exportado a WebGL.
- Integración con otras tecnologías (*MinIO*).
- Reproducción de contenido bajo demanda.

- Optimización de la escena.
- Etc.

Cabe destacar también el impacto que ha tenido la titulación sobre el alumno gracias al ámbito 3D: Cursar la asignatura antes mencionada, la impartición de un breve taller de Unity impartido por el tutor del presente trabajo y el desarrollo de dicho proyecto han hecho que se quiera **dar un vuelco en la carrera del alumno** buscando especializarse en el ámbito 3D y tecnologías inmersivas.

Por otro lado, las tecnologías inmersivas se encuentran a la orden del día (gracias a lo que algunos denominan **Metaverso**) por lo que haber profundizado en este campo y, en especial, en un estándar del **W3C** como es **WebXR** hace que se vea este trabajo no solo como una entrega más de la titulación si no como el comienzo de una etapa formativa en estas tecnologías.

Poniendo especial enfoque en el estándar **WebXR**, haber realizado un estudio y haber desarrollado una aplicación haciendo uso de él ha supuesto un trabajo de investigación importante. No solo es una novedad en el mercado hacer uso de **WebXR** para mostrar contenido inmersivo en web, sino que también lo es su integración con motores gráficos (Unity) y con otras tecnologías totalmente diferentes (*MinIO*) por lo que se destaca especialmente el trabajo de investigación y desarrollo en nuevas tecnologías que ha supuesto este proyecto.

También cabe destacar que el trabajo junto a una empresa (**Fundación CTIC**, sede del W3C) ha permitido desarrollar este trabajo de una forma más eficiente por diversos motivos: El equipo hardware era propiedad de la fundación, se tuvo soporte del departamento de tecnologías inmersivas y se pudo obtener una visión práctica del proyecto. Se entiende que el proyecto ha tenido una visión práctica pues la investigación y desarrollo con el estándar *WebXR* estaba en la lista de tareas pendientes de la

empresa, por lo que **el resultado de este trabajo ha servido para el desarrollo de otros proyectos que están llevando a cabo** (los controladores y exportado del proyecto se utilizan actualmente en otras tareas).

Por supuesto, cabe añadir que la **figura del tutor** resulta **imprescindible**. El tutor es la figura que puede ayudarte a visualizar las cosas de una forma que un alumno sin experiencia en el campo no puede. Esto hace que la implementación sea lo más óptima posible y con menos errores. Por supuesto, esta reflexión hace agradecer al tutor del trabajo por toda la ayuda recibida a lo largo del año que se ha alargado el desarrollo del trabajo.

Se debe añadir también que el presente trabajo ha supuesto **un antes y un después en la forma de entender el trabajo**. Haber descubierto nuevas tecnologías y haber reforzado el uso de otras ya conocidas ha supuesto un cambio importante en la formación y ha cambiado mi vida de ver el trabajo algo que sin duda puede ayudar a buscar ofertas mucho más especializadas en un mercado laboral que hoy en día es muy exigente.

12 Referencias

- [1] W3C. [En línea]. Available: <https://www.w3.org/TR/webxr/>. [Último acceso: 25 10 2021].
- [2] «Electronauts,» [En línea]. Available: https://store.steampowered.com/app/691160/Electronauts__VR_Music/. [Último acceso: 13 12 2021].
- [3] «Music Inside: A VR Rhythm Game,» [En línea]. Available: https://store.steampowered.com/app/520470/Music_Inside_A_VR_Rhythm_Game/?l=spanish. [Último acceso: 13 12 2021].
- [4] «Jam Studio VR,» [En línea]. Available: <https://jamstudiovr.com/>. [Último acceso: 13 12 2021].
- [5] [En línea]. Available: <https://www.youtube.com/watch?v=Topz0NkGlus>. [Último acceso: 13 12 2021].
- [6] [En línea]. Available: <https://www.oculus.com/experiences/>. [Último acceso: 13 12 2021].
- [7] [En línea]. Available: https://www.oculus.com/experiences/quest/3002729676463989/?locale=es_ES. [Último acceso: 13 12 2021].
- [8] Unity. [En línea]. Available: <https://unity.com/es>. [Último acceso: 16 12 2021].
- [9] EpicGames. [En línea]. Available: <https://www.unrealengine.com/en-US/>. [Último acceso: 16 12 2021].
- [10] [En línea]. Available: <https://threejs.org/>. [Último acceso: 16 12 2021].
- [11] [En línea]. Available: <https://code.visualstudio.com/>. [Último acceso: 16 12 2021].
- [12] [En línea]. Available: <https://visualstudio.microsoft.com/es/>. [Último acceso: 16 12 2021].
- [13] [En línea]. Available: <https://www.jetbrains.com/es-es/rider/>. [Último acceso: 16 12 2021].
- [14] [En línea]. Available: <https://www.blender.org/>. [Último acceso: 16 12 2021].
- [15] [En línea]. Available: <https://www.sketchup.com/es>. [Último acceso: 16 12 2021].
- [16] [En línea]. Available: <https://www.autodesk.com/>. [Último acceso: 16 12 2021].
- [17] «AWS,» [En línea]. Available: <https://aws.amazon.com/es/s3/>. [Último acceso: 23

] 12 2021].

[18 [En línea]. Available: <https://cloud.google.com/storage?hl=es>. [Último acceso: 23 12
] 2021].

[19 [En línea]. Available: <https://azure.microsoft.com/es-es/services/storage/blobs/>.
] [Último acceso: 23 12 2021].

[20 «MiniO,» [En línea]. Available: <https://min.io/>. [Último acceso: 23 12 2021].
]

[21 [En línea]. Available: <https://www.mongodb.com/es>. [Último acceso: 23 12 2022].
]

[22 Mozilla. [En línea]. Available: <https://blog.mozvr.com/unity-webxr-exporter-update/>.
] [Último acceso: 17 12 2021].

[23 De-Panther. [En línea]. Available: [https://github.com/De-Panther/unity-webxr-](https://github.com/De-Panther/unity-webxr-export)
] [export](https://github.com/De-Panther/unity-webxr-export). [Último acceso: 17 12 2021].

[24 J. Á. d. I. Fuente, «Github,» [En línea]. Available:
] https://github.com/Javieral95/Getting_Started_With_UnityWebXR. [Último acceso:
17 12 2021].

[25 «PolyHaven,» [En línea]. Available: <https://polyhaven.com/textures>. [Último acceso:
] 28 01 2022].

[26 «Heroku,» [En línea]. Available: <https://www.heroku.com/>. [Último acceso: 1 7
] 2022].

[27 «Docker,» [En línea]. Available: <https://www.docker.com/>. [Último acceso: 1 7
] 2022].

[28 «Docker-Compose,» [En línea]. [Último acceso: 1 7 2022].
]

13 Documentación adicional

Debido a que el presente documento de memoria del proyecto se trata del principal archivo que se entregará en este trabajo, contiene tanto aspectos de estudio, como análisis y otros elementos técnicos.

No obstante, a pesar de ser el documento principal, se entregan otros documentos adjuntos que acompañan al desarrollo de este trabajo de fin de máster.

Documento 1: Presupuesto.

Este documento recoge las partidas presupuestarias divididas en capítulos de: Recursos humanos, hardware, software y fungibles.

Debido a existir una planificación inicial y otra final se tienen los apartados:

1. Resumen
2. Presupuesto a priori
3. Presupuesto final
4. Presupuesto total

Código del producto software.

No se trata de un documento a lo sumo, pero se adjunta un fichero de texto con la url para descargar el código software.

En la [url](#) se indica la ruta para acceder a un documento comprimido albergado en la nube de *OneDrive* de la Universidad de Oviedo. El fichero comprimido contiene dos subdirectorios:

- Uno para el despliegue, con todos los archivos necesarios para levantar el despliegue haciendo uso de contenedores.
- Otro para el desarrollo de la aplicación.

Para evitar aumentar innecesariamente el tamaño del comprimido, se entrega únicamente lo necesario: Es decir, los directorios *Library* y *obj* no se incluyen en el archivo (se generan automáticamente al abrir el proyecto en Unity). Tampoco se incluye la aplicación compilada en el directorio del despliegue.