



Universidad de Valladolid

Escuela de Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática
(Mención en Computación)

EDUCAVERSE

Plataforma para la educación diabética a través de una experiencia compatible con el metaverso

Autor:

Jhon Steeven Cabanilla Alvarado

Tutores:

**Jose Luis Vidal de la Rosa
César Pablo Gutiérrez**

Dedicado a mi familia y amigos.

Agradecimientos

En primer lugar, quiero agradecer a mi familia por ser mi mayor fuente de inspiración y apoyo incondicional. Gracias por estar siempre a mi lado, por brindarme su amor y aliento en cada paso que he dado. Vuestra confianza en mí ha sido fundamental para superar los desafíos y alcanzar este logro.

Darle las gracias también a mis tutores, Jose Luis Vidal de la Rosa y César Pablo Gutiérrez, por su invaluable guía y orientación a lo largo de todo el proceso. Sin vuestra ayuda, no habría sido posible llegar hasta aquí.

Por último, me gustaría agradecer a mis compañeros de carrera. Hemos compartido momentos inolvidables, hemos aprendido juntos y hemos crecido como profesionales. Me habéis ayudado a superar mis límites y a dar lo mejor de mí.

¡Gracias a todos!

Resumen

Educaverse es una aplicación de realidad virtual diseñada para brindar educación intuitiva y adaptada a debutantes diabéticos familiarizados con la tecnología. El objetivo principal de Educaverse consiste en ofrecer una experiencia inmersiva a través de la interacción con un avatar médico y la posibilidad de acceder a vídeos educativos sobre la diabetes. El resultado de este TFG tiene como propósito promover la comprensión de conceptos médicos relacionados con la enfermedad, estimular la curiosidad y facilitar el aprendizaje en un entorno virtual. Con Educaverse, se busca superar las barreras tradicionales de la educación diabetológica y ofrecer una alternativa atractiva y accesible para los debutantes en el mundo de la diabetes.

Palabras clave: Unity 3D, Realidad Virtual, Metaverso, Windows Mixed Reality, Diabetes, Educación diabetológica.

Abstract

Educaverse is a virtual reality application designed to provide intuitive and tailored education for tech-savvy diabetic beginners. The main objective of Educaverse is to offer an immersive experience through interaction with a medical avatar and the possibility of accessing educational videos about diabetes. The result of this project aims to promote the understanding of medical concepts related to the disease, stimulate curiosity and facilitate learning in a virtual environment. With Educaverse, the goal is to overcome the traditional barriers of diabetes education and offer an attractive and accessible alternative for newcomers to the world of diabetes.

Keywords: Unity 3D, Virtual Reality, Metaverse, Windows Mixed Reality, Diabetes, Diabetes education.

Índice general

1. Introducción	1
1.1. Contexto	1
1.2. Motivación	1
1.3. Objetivos	2
1.4. Metodología	2
1.5. Estructura de la memoria	3
2. Realidad Virtual	5
2.1. Definición y funcionamiento	5
2.1.1. Tipos de Realidad Virtual	5
2.1.2. Diferencias entre realidad virtual y realidad aumentada	6
2.1.3. ¿Cómo funciona la realidad virtual?	6
2.2. Tecnología	6
2.2.1. VR Headsets	6
2.2.2. Accesorios de Realidad virtual	7
2.3. Estado del arte	8
2.3.1. Aplicaciones	8
2.4. Beneficios de la realidad virtual	10
2.5. Retos de la realidad virtual	11
2.6. Proyección de futuro	11
3. Metaverso	13
3.1. Concepto	13
3.1.1. Tipos de metaversos	13
3.2. Características	14
3.3. Cómo acceder al Metaverso	15
3.4. Futuro	15

4. Entorno y herramientas tecnológicas	17
4.1. Unity	17
4.1.1. Interfaz de Unity	18
4.1.2. Términos Generales	19
4.1.3. Características y componentes de Unity	19
4.2. Unity Hub	21
4.3. C#	21
4.4. Meta	22
4.4.1. Proyectos	23
4.5. GitLab	24
4.6. Astah	25
4.7. Overleaf	26
5. Planificación	27
5.1. Planificación inicial	27
5.1.1. Análisis de riesgos	28
5.1.2. Plan de presupuesto	32
5.1.3. Plan de trabajo	33
5.2. Seguimiento y control	35
5.2.1. Desviaciones en la planificación	38
6. Análisis	39
6.1. Descripción del sistema	39
6.2. Actores de la aplicación	39
6.3. Requisitos	39
6.3.1. Requisitos Funcionales	39
6.3.2. Requisitos No Funcionales	40
6.4. Derechos de autor	41
6.5. Casos de Uso	41
6.5.1. Diagrama de casos de uso	41
6.5.2. Especificación de escenarios	42
6.6. Modelo de Dominio	45
6.7. Realización en análisis de los casos de uso	46

7. Diseño	51
7.1. Entorno tecnológico	51
7.2. Arquitectura del sistema	52
7.2.1. Estilo arquitectónico	52
7.3. Patrones de diseño	53
7.3.1. Patrón MVC (Modelo-Vista-Controlador)	53
7.3.2. Patrón Observer	54
7.3.3. Patrón State	54
7.3.4. Patrón Singleton	55
7.4. Diagrama de paquetes	55
7.5. Diagrama de componentes	59
7.6. Diagrama de despliegue	60
7.7. Diseño de la Interfaz de Usuario	61
7.7.1. Principios de diseño de interfaz de usuario	62
7.7.2. Sala médica	62
7.7.3. Avatar médico	62
7.7.4. Canvas	63
8. Implementación	65
8.1. Iteración 1	65
8.1.1. Issues	65
8.1.2. Tareas completadas	66
8.1.3. Funcionalidad realizada	67
8.2. Iteración 2	68
8.3. Iteración 3	68
8.3.1. Tareas completadas	68
8.3.2. Funcionalidad realizada	71
8.4. Iteración 4	73
8.4.1. Tareas completadas	73
8.4.2. Funcionalidad Realizada	75
8.5. Iteración 5	80
8.5.1. Tareas Implementadas	81
8.5.2. Funcionalidad Realizada	83

8.6. Iteración 6	87
8.6.1. Tareas bloqueadas	87
8.6.2. Implementación de la Realidad Virtual: XR Interaction Toolkit	88
8.6.3. Compatibilidad META	92
9. Pruebas y Resultados	97
9.1. Test Suite	97
9.2. Pruebas con usuarios	100
10. Conclusiones y trabajo futuro	103
10.1. Conclusiones	103
10.2. Trabajo futuro	104
Bibliografía	105
A. Manual de despliegue	111
A.1. Despliegue del proyecto en Unity	111
A.1.1. Requisitos del sistema	111
A.1.2. Proceso de despliegue	111
B. Manual de usuario	113
B.1. Interfaz de usuario	113
B.2. Funcionalidades principales	117
B.3. Comandos de voz	117
B.4. Ayuda	118

Índice de figuras

1.1. Representación de la metodología ágil. Fuente [6]	3
2.1. PlayStation VR 2. Fuente [8]	7
2.2. Meta Quest 2. Fuente [10]	7
2.3. Samsung Gear VR. Fuente [11]	8
2.4. Cadete utiliza un sistema de RV para practicar habilidades de vuelo. Fuente [12]	9
2.5. RendeverFit. Fuente [14]	9
2.6. Dubai VR Park. Fuente [15]	9
2.7. Pasajero de Qantas visualizando destinos virtuales en 3D. Fuente [16]	10
2.8. Videojuego de realidad virtual	10
4.1. Logo de Unity	17
4.2. Interfaz de Unity. Fuente [25]	18
4.3. Controlador de animaciones para gestionar estados. Fuente [27]	20
4.4. Logo de Unity Hub	21
4.5. Logo de C#	22
4.6. Logo de Meta	22
4.7. Logo de GitLab	24
4.8. Logo de Astah Professional	25
4.9. Interfaz de Astah Professional. Fuente [45]	25
4.10. Logo de Overleaf	26
5.1. Diagrama de Gantt de las iteraciones del proyecto	28
5.2. Matriz de Riesgos	29
6.1. Diagrama de casos de uso	41
6.2. Modelo del Dominio	45

6.3. Realización en análisis del caso de uso Interactuar con el Avatar	47
6.4. Realización en análisis del caso de uso Activar Reconocedor de Voz	47
6.5. Realización en análisis del caso de uso Introducir comando de voz	48
6.6. Realización en análisis del caso de uso Visualizar catálogo	48
6.7. Realización en análisis del caso de uso Reproducir vídeo	49
6.8. Realización en análisis del caso de uso Controlar vídeo	50
6.9. Realización en análisis del caso de uso Salir modo visualización	50
7.1. Capas de la aplicación	52
7.2. Estructura del patrón Observador. Fuente [61]	54
7.3. Estructura del patrón State. Fuente [62]	54
7.4. Estructura del patrón Singleton. Fuente [63]	55
7.5. Estructura de Unity por paquetes	56
7.6. Diseño detallado del paquete Scripts	56
7.7. Diseño detallado del paquete models	57
7.8. Diseño detallado del paquete view	57
7.9. Diseño detallado del paquete Doctor’sOfficeView	57
7.10. Diseño detallado del paquete DoctorView	58
7.11. Diseño detallado del paquete controllers	58
7.12. Diagrama de componentes	59
7.13. Diagrama de despliegue	61
7.14. Consulta médica. Fuente [67]	62
7.15. Avatar médico. Fuente [68]	63
7.16. Avatar médico activado	63
7.17. Catálogo de vídeos sobre el “Glucómetro”	64
7.18. Reproducción del vídeo “Glucemia capilar”	64
8.1. Fin de la primera iteración	65
8.2. Ventana Installs de Unity Hub	67
8.3. Ventana de New Project de Unity Hub	67
8.4. Fin de la tercera iteración	69
8.5. Package Manager	72
8.6. Scene View	72
8.7. Fin de la cuarta iteración	73

8.8. Barra de menú	76
8.9. Ventana Browse Sketchfab	76
8.10. Modelo médico. Fuente [68]	77
8.11. Introducción del avatar en la sala médica	77
8.12. Fin de la quinta iteración	80
8.13. Sección On Click() del Avatar Médico	85
8.14. AnimationClip - Take 001	86
8.15. Aspecto de la capa Base Layer	86
8.16. Avatar médico interactuable desde VR	90
8.17. Configuración sección Select	90
8.18. Avatar médico pulsado desde VR	91
8.19. Vista del canvas desde VR	91
8.20. Reproducción vídeo desde VR	92
8.21. Módulos de dependencia	92
8.22. Build Settings	93
8.23. Configuración inicial	93
8.24. Configuración de identificación del paquete	94
8.25. Configuración de la aplicación	94
8.26. Gestión de XR	94
8.27. Configuración de renderización	95
8.28. Configuración de calidad	95
B.1. Sala médica de Educaverse	114
B.2. Avatar médico de Educaverse	114
B.3. Avatar médico pulsado	115
B.4. Catálogo de vídeos de Educaverse	115
B.5. Vídeo educativo de Educaverse	116
B.6. Ventana emergente con el título del vídeo	117

Índice de tablas

5.1. Riesgo 01 - Retrasos en la adquisición de recursos técnicos	29
5.2. Riesgo 02 - Cambios en los requisitos	29
5.3. Riesgo 03 - Problemas de compatibilidad con las plataformas del Metaverso	29
5.4. Riesgo 04 - Desmotivación a la hora de trabajar	30
5.5. Riesgo 05 - Mala estimación de tiempo	30
5.6. Riesgo 06 - Avería del ordenador	30
5.7. Riesgo 07 - Complejidad del proyecto	31
5.8. Riesgo 08 - Uso de una tecnología con la que no se ha trabajado anteriormente	31
5.9. Riesgo 09 - Contracción de enfermedades	32
5.10. Riesgo 10 - Inexperiencia en la gestión de proyectos	32
5.11. Presupuesto del proyecto	33
5.12. Plan de trabajo de la primera iteración	33
5.13. Plan de trabajo de la segunda iteración	34
5.14. Plan de trabajo de la tercera iteración	34
5.15. Plan de trabajo de la cuarta iteración	34
5.16. Plan de trabajo de la quinta iteración	35
5.17. Plan de trabajo de la sexta iteración	35
5.18. Distribución temporal inicial	35
5.19. Plan de trabajo real de la primera iteración	36
5.20. Plan de trabajo real de la segunda iteración	36
5.21. Plan de trabajo real de la tercera iteración	36
5.22. Plan de trabajo real de la cuarta iteración	37
5.23. Plan de trabajo real de la quinta iteración	37
5.24. Plan de trabajo real de la sexta iteración	37
5.25. Distribución temporal final	38

6.1. Requisitos funcionales del sistema	40
6.2. Requisitos no funcionales del sistema	40
6.3. Caso de Uso CU-01	42
6.4. Caso de Uso CU-02	42
6.5. Caso de Uso CU-03	43
6.6. Caso de Uso CU-04	43
6.7. Caso de Uso CU-05	44
6.8. Caso de Uso CU-06	44
6.9. Caso de Uso CU-07	45
8.1. Issue 1	66
8.2. Issue 2	69
8.3. Issue 3	70
8.4. Issue 4	70
8.5. Issue 5	71
8.6. Issue 6	71
8.7. Issue 7	74
8.8. Issue 8	74
8.9. Issue 9	74
8.10. Issue 10	75
8.11. Issue 11	75
8.12. Issue 12	81
8.13. Issue 13	81
8.14. Issue 14	82
8.15. Issue 15	82
8.16. Issue 16	83
8.17. Issue 17	83
9.1. Test Case 1	98
9.2. Test Case 2	98
9.3. Test Case 3	99
9.4. Test Case 4	99
9.5. Test Case 5	100
9.6. Test Case 6	100

Capítulo 1

Introducción

1.1. Contexto

La empresa HP SCDS [1] ofrece a los estudiantes de diversas universidades la oportunidad de llevar a cabo sus Trabajos de Fin de Grado en un entorno empresarial a través del Observatorio Tecnológico HP [2]. En este contexto, los estudiantes reciben la asistencia de un tutor experimentado para orientarlos durante el desarrollo de sus proyectos.

El proceso de selección y asignación de proyectos a los estudiantes consta de varios pasos, siendo uno de ellos la selección de candidatos en función de su expediente en la carrera. Posteriormente, los tutores profesionales de HP SCDS se comunican con los estudiantes seleccionados para una reunión inicial, donde se establecen los objetivos, el enfoque de trabajo, la metodología a seguir y la duración del proyecto. En el caso específico de este Trabajo de Fin de Grado, el tutor designado por la empresa es Jose Luis Vidal de la Rosa, quien posee experiencia previa en la tutorización de este tipo de proyectos, además de ser un experto en las tecnologías requeridas y en la metodología Scrum.

Educaverse surge como una idea original del tutor de la empresa, con el propósito de llevar la educación diabética al metaverso. Una vez planteada esta idea, es responsabilidad del estudiante decidir el enfoque que seguirá el proyecto para alcanzar el objetivo establecido, recibiendo la guía y apoyo del tutor profesional de HP SCDS y del tutor académico, el profesor universitario César Pablo Gutiérrez.

1.2. Motivación

Durante las últimas décadas, hemos sido testigos de un alarmante aumento en el número de personas afectadas por la diabetes, especialmente entre la población adolescente y joven [3]. Este crecimiento representa un cambio significativo en el estilo de vida de los pacientes, quienes deben enfrentarse a nuevos desafíos y realizar un seguimiento constante de su condición. Es fundamental para ellos mantener un control riguroso de la enfermedad y comprender su propio cuerpo.

Esta transformación en la vida de los pacientes diabéticos puede dar lugar a la aparición de enfermedades relacionadas, que afectan órganos como el corazón, los riñones, los pies y pueden incluso causar complicaciones oculares, como la retinopatía [4]. Ante este panorama, es crucial brindar una educación diabetológica adecuada, especialmente entre los adolescentes, quienes a menudo enfrentan desafíos adicionales debido a su etapa de desarrollo y su susceptibilidad a hábitos poco saludables.

Es en este contexto que Educaverse adquiere una relevancia significativa. Al llevar la educación diabetológica al metaverso mediante la realidad virtual, esta aplicación ofrece una plataforma interactiva y envolvente que permite a los adolescentes aprender de manera efectiva sobre su condición. Al proporcionar una experiencia inmersiva, Educaverse busca aumentar la conciencia sobre la diabetes y su manejo adecuado, brindando información precisa sobre la enfermedad, hábitos de vida saludables, control de glucosa o cuidado personal.

Desde mi perspectiva, este proyecto me brinda la oportunidad de adquirir nuevos conocimientos y habilidades en el campo de las tecnologías emergentes como lo es la realidad virtual y, al mismo tiempo, poder proporcionar a los pacientes diabéticos una herramienta innovadora y accesible que les permita tomar el control de su salud y promover un mejor manejo de la diabetes en su vida diaria.

1.3. Objetivos

El objetivo final de Educaverse es crear una aplicación de realidad virtual que sea compatible con el Metaverso, donde usuarios de cualquier edad puedan encontrar una forma fácil e intuitiva de formarse en diabetes en primera instancia. Se trata por tanto, de llevar la consulta de educación diabetológica al metaverso, donde un avatar nos estará esperando para resolver todas nuestras dudas.

La fase inicial de esta aplicación implementará la interacción con un avatar del metaverso, que responderá a las preguntas del usuario a través de una batería de videos relacionados con el concepto de búsqueda que plantee el usuario.

1.4. Metodología

Para el desarrollo del proyecto se ha decidido utilizar una metodología Agile, es decir, un enfoque de desarrollo de software basado en la flexibilidad, la colaboración y la adaptabilidad para lograr proyectos exitosos. A diferencia de los enfoques tradicionales que siguen un enfoque de cascada, la metodología Agile se centra en la entrega incremental y en la mejora continua a lo largo del ciclo de vida del proyecto [5].

Dentro de este enfoque ágil se ha seleccionado la metodología Scrum, diseñado para mejorar la productividad, la calidad y la satisfacción del cliente a través de un marco de trabajo estructurado. Este marco de trabajo se basa en Sprints, que son iteraciones de tiempo fijo en las que se desarrolla una parte del producto. Durante un sprint, se llevan a cabo las siguientes actividades:

- **Planificación del Sprint:** El equipo de desarrollo y el Propietario del Producto definen las funcionalidades que se desarrollarán durante el sprint y las priorizan en el Backlog del Producto.
- **Reuniones Diarias (Daily Scrum):** El equipo de desarrollo se reúne diariamente para revisar el progreso del sprint, discutir los obstáculos y planificar el trabajo para el día siguiente.
- **Revisión del Sprint:** Al final del sprint, el equipo de desarrollo muestra las funcionalidades desarrolladas durante el sprint en una reunión de revisión del sprint con los interesados. Se recopila retroalimentación y se ajusta el Backlog del Producto en consecuencia.
- **Retrospectiva del Sprint:** El equipo de desarrollo se reúne para reflexionar sobre el sprint y realizar mejoras en el proceso. Se identifican los puntos fuertes y débiles y se establecen acciones de mejora.

En la metodología Scrum, se asignan roles a los diferentes miembros del equipo pero, al tratarse de un proyecto individual, una sola persona ejercerá dichos roles. Los principales roles en Scrum son:

- **Equipo de Desarrollo:** Es el grupo de personas encargado de desarrollar el software. El equipo de desarrollo es multifuncional y autoorganizado, y tiene la responsabilidad de planificar, diseñar, desarrollar, probar y entregar las funcionalidades del producto.
- **Scrum Master:** Es el encargado de asegurar que se sigan las reglas de Scrum y de facilitar el proceso. El Scrum Master actúa como líder del equipo, elimina obstáculos y promueve la mejora continua.
- **Propietario del Producto (Product Owner):** Es el representante de los interesados o del cliente, y tiene la responsabilidad de definir y priorizar los requisitos del producto. El Propietario del Producto trabaja de cerca con el equipo de desarrollo para asegurar que se estén desarrollando las funcionalidades correctas.

Además de estas actividades, Scrum utiliza artefactos como el Backlog del Producto (que contiene la lista de requisitos pendientes de priorizar), el Backlog del Sprint (que contiene las funcionalidades que se desarrollarán en el sprint) y el Incremento del Producto (la suma de todas las funcionalidades completadas hasta el momento).

Una de las características principales de Scrum es su capacidad para adaptarse a los cambios. Los requisitos y prioridades del producto pueden cambiar durante el desarrollo, y Scrum permite ajustar el trabajo del equipo de desarrollo en consecuencia. Scrum también promueve la transparencia, la inspección y la adaptación constante, lo que facilita la mejora continua del proceso de desarrollo.



Figura 1.1: Representación de la metodología ágil. Fuente [6]

1.5. Estructura de la memoria

La memoria se encuentra dividada en los siguientes capítulos principales:

- **Introducción.** Presentación del proyecto de una forma general, mediante el trato de su contexto, motivación, los objetivos visualizados y la metodología a seguir.
- **Realidad Virtual.** Descripción detallada de la realidad virtual, así como su funcionamiento, tecnología e incluso beneficios y retos.
- **Metaverso.** Definición del metaverso, explicando los distintos tipos que existen, además de las características generales y una especificación de cómo acceder al metaverso.
- **Entorno y herramientas utilizadas.** Descripción del entorno de desarrollo utilizado, además de las herramientas empleadas para llevar a cabo el proyecto
- **Planificación.** Análisis, descripción y estimación de las iteraciones con sus tiempos de entrega, estimación del coste del proyecto y descripción de riesgos y planes de acción.
- **Análisis.** Esta sección se centra en el análisis del problema a resolver, incluyendo el estudio de requisitos, la identificación de funcionalidades y la elaboración de modelos y diagramas para comprender la estructura del proyecto.
- **Diseño.** En esta etapa se desarrolla el diseño del proyecto, definiendo la arquitectura del software, la interfaz de usuario, la estructura de datos y otros aspectos relacionados con la implementación.
- **Implementación.** Descripción detallada del trabajo realizado en cada una de las iteraciones que conforman el proyecto.
- **Pruebas y Resultados.** Este capítulo describe las pruebas realizadas con el objetivo de detectar errores y garantizar la calidad del producto final, y además se muestran los resultados obtenidos.

- **Conclusiones y trabajo futuro.** En esta parte se presentan las conclusiones obtenidas a partir del desarrollo del proyecto, así como las posibles mejoras o trabajos futuros que podrían llevarse a cabo para continuar con el desarrollo o ampliar las funcionalidades.
- **Bibliografía.** Contiene las referencias bibliográficas utilizadas durante todo el transcurso del proyecto.
- **Apéndices.** Aquí se incluyen documentos adicionales que complementan la información presentada en los capítulos anteriores, tales como un manual de despliegue y un manual de usuario.

Capítulo 2

Realidad Virtual

La realidad virtual es un fascinante campo que ha experimentado un rápido crecimiento en los últimos años, transformando la forma en que interactuamos con el mundo digital. Con su capacidad para sumergir a los usuarios en entornos virtuales envolventes y sensoriales, la realidad virtual ha abierto nuevas posibilidades en diversos ámbitos, desde el entretenimiento y la educación hasta la medicina y la industria.

En este capítulo, exploraremos en detalle la realidad virtual, comenzando por su definición y funcionamiento. Examinaremos las tecnologías subyacentes que hacen posible esta experiencia inmersiva y los avances que han permitido su desarrollo y adopción a gran escala.

2.1. Definición y funcionamiento

La realidad virtual se refiere al uso de tecnología informática para crear entornos simulados que sitúan al usuario en una experiencia tridimensional [7]. En lugar de limitarse a ver una pantalla, los usuarios se sumergen en mundos tridimensionales e interactúan con ellos. Esto se consigue mediante la simulación de los sentidos humanos, lo que permite al ordenador actuar como vehículo hacia nuevos mundos. Los únicos límites reales a una experiencia de realidad virtual de primera categoría son la potencia informática disponible y los contenidos a disposición de los usuarios.

2.1.1. Tipos de Realidad Virtual

Existen tres tipos principales de experiencias de realidad virtual (RV), que se denominan colectivamente realidad extendida. Estos tres tipos ofrecen distintos niveles de simulación generada por ordenador: RV no inmersiva, RV semiinmersiva y RV totalmente inmersiva [7].

- **Realidad Virtual no inmersiva:** Este tipo de RV presenta un entorno virtual generado por ordenador en el que el usuario sigue siendo consciente de su entorno físico y controlado por él. Los videojuegos son un ejemplo de RV no inmersiva.
- **Realidad Virtual semiinmersiva:** Proporciona un entorno parcialmente virtual y suele utilizarse con fines educativos y de formación. Este tipo de RV puede conseguirse mediante la informática gráfica y los grandes sistemas de proyectores, como los simuladores de vuelo para aprendices de piloto.
- **Realidad Virtual totalmente inmersiva:** Es el tipo más realista de experiencia de simulación. Aunque en la actualidad no existen tecnologías de RV completamente inmersivas, los avances en este campo se suceden a un ritmo vertiginoso. Los juegos de carreras de coches son un ejemplo de realidad virtual inmersiva que proporciona a los usuarios la sensación de velocidad y habilidad al volante. Aunque se ha desarrollado principalmente con fines lúdicos y de entretenimiento, la tecnología de RV se utiliza cada vez más en otros sectores.

2.1.2. Diferencias entre realidad virtual y realidad aumentada

Cuando se trata de experiencias digitales, hay dos tipos principales: la realidad virtual y la realidad aumentada [7]. La realidad virtual es una experiencia totalmente inmersiva que saca a los usuarios del mundo real y los lleva a un entorno completamente artificial. Utiliza estímulos sensoriales para crear entornos sintéticos, y las acciones de los usuarios pueden influir en lo que ocurre dentro de este entorno. Estos entornos a menudo se modelan a partir de lugares reales y existen separados del mundo físico.

Por otro lado, la realidad aumentada mejora las vistas del mundo real de los usuarios con superposiciones digitales que incorporan objetos artificiales. La realidad aumentada puede experimentarse directamente a través del campo de visión del usuario o mediante una cámara u otro dispositivo. Esta tecnología añade entradas digitales, como gráficos, audio o vídeo, al mundo real en tiempo real. A diferencia de la RV, la RA no crea una experiencia completamente nueva a partir de cero, sino que se basa en la experiencia existente en el mundo real.

2.1.3. ¿Cómo funciona la realidad virtual?

La tecnología de realidad virtual está diseñada para crear experiencias inmersivas que engañan al cerebro para que perciba como real un entorno simulado. Esto se consigue mediante una combinación de hardware y software.

El hardware utilizado en la tecnología de realidad virtual está diseñado para estimular los sentidos y crear una simulación realista del entorno. Esto puede incluir elementos como sonidos, tacto, olfato o incluso cambios en la intensidad del calor. Por su parte, el software se encarga de crear el entorno virtual que experimentará el usuario.

El funcionamiento de la tecnología de realidad virtual se basa en la forma en que el ojo y el cerebro humanos perciben la profundidad y crean imágenes en 3D. Cuando miramos un objeto, cada uno de nuestros ojos ve una imagen ligeramente distinta debido a su posición, lo que permite al cerebro crear una sensación de profundidad y dimensionalidad.

Las aplicaciones de realidad virtual reproducen este fenómeno presentando al usuario dos imágenes idénticas, cada una desde una perspectiva ligeramente distinta, para crear una visualización estereoscópica. Estas imágenes se muestran en una o varias pantallas dentro del casco de RV, que se coloca cerca de los ojos del usuario. Las ligeras diferencias entre las imágenes crean la ilusión de profundidad, y se engaña al cerebro para que perciba la imagen como multidimensional. [7]

2.2. Tecnología

La tecnología de RV suele constar de cascos y accesorios como controladores y rastreadores de movimiento. Estos componentes se utilizan para crear un entorno virtual totalmente inmersivo con el que el usuario puede interactuar.

2.2.1. VR Headsets

El componente más destacado del hardware de RV es el casco de RV, que es un dispositivo montado en la cabeza que crea una experiencia visual totalmente inmersiva para el usuario. El casco es como un par de gafas que el usuario lleva en la cabeza y suele incluir sensores o cámaras de última generación para el seguimiento del movimiento de los ojos o la cabeza. El casco es lo que permite a los usuarios ver e interactuar con el entorno virtual. Existen los siguientes tipos de headsets:

- **Cascos de RV para PC:** Los cascos de RV para PC ofrecen las experiencias de RV más inmersivas y se consideran los dispositivos más caros debido a sus características avanzadas. Suelen estar conectados por cable al casco y se alimentan de hardware externo, como un PC o una videoconsola. La pantalla dedicada, los sensores de movimiento incorporados y los rastreadores de cámara externos ofrecen sonido e imagen de alta calidad y seguimiento de la cabeza para un mayor realismo. Proporcionan una experiencia altamente

inmersiva con un amplio campo de visión y gráficos de alta calidad, por lo que son ideales para jugadores y profesionales.



Figura 2.1: PlayStation VR 2. Fuente [8]

- **Cascos de RV autónomos:** Los cascos de RV autónomos son piezas de hardware inalámbricas e integradas que no necesitan ningún dispositivo externo para funcionar. Son un punto intermedio entre los cascos para PC y los móviles. Proporcionan una experiencia altamente inmersiva sin necesidad de ningún dispositivo externo, lo que los hace muy portátiles. Estos cascos suelen llevar sensores de movimiento incorporados, y algunos pueden incluir cámaras de seguimiento externas para un seguimiento más preciso de la cabeza. Proporcionan una buena experiencia de RV, pero no tan inmersiva como los cascos para PC. Dentro de esta categoría se encontrarían las Apple Vision Pro [9], las nuevas gafas de Apple que mezclan la realidad virtual con la realidad aumentada.



Figura 2.2: Meta Quest 2. Fuente [10]

- **Cascos móviles:** Los auriculares para móviles son relativamente baratos y utilizan lentes que cubren la pantalla de un smartphone. Las lentes separan la pantalla para crear una imagen estereoscópica que transforma el smartphone en un dispositivo de RV. Los auriculares móviles no necesitan cables, ya que el teléfono se encarga del procesamiento. Sin embargo, no ofrecen las mejores experiencias visuales y su potencia es inferior a la de la RV basada en videoconsolas o PC. No ofrecen seguimiento posicional, lo que significa que el entorno generado se muestra desde un único punto, y no es posible mirar alrededor de los objetos de una escena. Los cascos móviles son adecuados para experiencias casuales de RV, como ver vídeos de 360 grados o juegos básicos de RV. (Ver figura 2.3).

2.2.2. Accesorios de Realidad virtual

Los accesorios de realidad virtual son dispositivos físicos que mejoran la experiencia de RV. Algunos de los accesorios más utilizados son los ratones 3D, los rastreadores ópticos, los guantes con cable, los controladores de movimiento, los trajes corporales, las cintas de correr e incluso los dispositivos que producen olores.



Figura 2.3: Samsung Gear VR. Fuente [11]

- **Un ratón 3D:** dispositivo señalador diseñado para el movimiento en espacios virtuales 3D, y emplea varios métodos para controlar el movimiento 3D y el señalamiento 2D.
- **Los rastreadores ópticos** son dispositivos visuales que controlan la posición del usuario y pueden utilizar cámaras de vídeo fijas para seguir objetos o personas.
- **Los guantes con cable** se llevan en las manos y también se denominan guantes cibernéticos o de datos. Captan datos de movimiento físico y disponen de un rastreador de movimiento que capta los datos de rotación y posición global del guante.
- **Los controladores de movimiento** son accesorios que permiten a los usuarios interactuar con objetos digitales y tener una posición precisa en el espacio.
- **Los controladores de movimiento omnidireccionales (ODTs)** permiten a los usuarios moverse físicamente en cualquier dirección, proporcionando una experiencia de inmersión total en entornos de RV.
- **Dispositivos olfativos:** son uno de los accesorios más recientes en el mundo de la RV que emiten olores para transmitir una experiencia realista.

2.3. Estado del arte

El estado del arte relacionado con la realidad virtual está en constante evolución, debido a los avances tecnológicos y a la creciente demanda de experiencias inmersivas y de alta calidad en diversos campos. A continuación, se presentan algunas tendencias y desarrollos destacados en el ámbito de la RV:

2.3.1. Aplicaciones

La realidad virtual tiene una amplia gama de usos en diversos sectores. He aquí algunos ejemplos:

- **Educación y formación:** La realidad virtual puede proporcionar una simulación realista de diversos escenarios de formación que son caros, peligrosos o difíciles de reproducir en la vida real. Por ejemplo, puede utilizarse para simular vuelos de entrenamiento para pilotos o entrenamiento militar para soldados [12].
- **Atención sanitaria:** La realidad virtual en el cuidado de la salud ofrece numerosos beneficios ya que permite practicar y mejorar habilidades sin riesgo para los pacientes reales, puede utilizarse también en terapias de salud mental o en la terapia física para mejorar la recuperación y el entrenamiento de los pacientes. Incluso puede utilizarse para la rehabilitación de accidentes cerebrovasculares [13].
- **Entretenimiento:** La realidad virtual puede utilizarse en el cine para permitir a los espectadores no solo ver películas, sino ser parte de la experiencia, en museos para ayudar a las personas a explorar el pasado de manera interactiva, en la galerías de arte proporcionando una experiencia inmersiva y facilitando la interacción con



Figura 2.4: Cadete utiliza un sistema de RV para practicar habilidades de vuelo. Fuente [12]



Figura 2.5: RendeeverFit. Fuente [14]



Figura 2.6: Dubai VR Park. Fuente [15]

las obras, o incluso en el ámbito de la música y las discotecas, la realidad virtual puede transportar a los usuarios a conciertos y discotecas virtuales [15].

- **Marketing y publicidad:** La realidad virtual puede utilizarse también para crear experiencias de marketing interactivas y envolventes, como demostraciones virtuales de productos y experiencias de marca.
- **Gaming:** La realidad virtual en los juegos brinda a los jugadores la posibilidad de sumergirse en entornos virtuales tridimensionales y emocionantes a través de avanzadas consolas, controladores y cascos de RV. Los sistemas de juego de RV más populares, como HTC Vive u Oculus Quest, proporcionan una experiencia de juego fluida que no interrumpe la inmersión en la realidad virtual. Con los juegos de RV, los jugadores



Figura 2.7: Pasajero de Qantas visualizando destinos virtuales en 3D. Fuente [16]

pueden interactuar de manera natural e intuitiva con el entorno y los objetos, lo que agrega un nuevo nivel de emoción y sumergimiento al mundo del juego. Los movimientos y acciones del jugador se reflejan en el entorno virtual, lo que permite una mayor interactividad y una experiencia más realista [17].



Figura 2.8: Videojuego de realidad virtual

2.4. Beneficios de la realidad virtual

La realidad virtual ofrece una serie de beneficios en diversas áreas, tales como el aprendizaje, sanidad, seguridad o entretenimiento [18].

- **Mejora del aprendizaje:** La realidad virtual está demostrando ser una herramienta excelente para la educación y la formación. Con la realidad virtual, los alumnos pueden experimentar conceptos y situaciones complejas en un entorno realista e interactivo. Este enfoque inmersivo del aprendizaje puede mejorar la retención de conocimientos y el compromiso.
- **Mejora de la atención sanitaria:** La realidad virtual ya ha demostrado su potencial en el sector sanitario. Puede utilizarse para formar a profesionales médicos y simular procedimientos, lo que les permite practicar sin riesgo de dañar a pacientes reales. Además, la terapia de RV puede utilizarse para tratar diversas enfermedades mentales, como el TEPT y los trastornos de ansiedad.
- **Mayor seguridad:** La realidad virtual es una herramienta excelente para situaciones peligrosas o de riesgo, como la formación militar, la extinción de incendios o la minería. En estos escenarios, la realidad virtual puede simular situaciones de alto riesgo, lo que permite a las personas aprender a reaccionar en un entorno seguro y controlado.

- **Entretenimiento mejorado:** La realidad virtual proporciona una experiencia de entretenimiento única y envolvente. Puede transportar a los usuarios a nuevos mundos, proporcionándoles una experiencia inmersiva e interactiva que no puede ser igualada por las formas tradicionales de los medios de comunicación.
- **Mejora del diseño y la ingeniería:** La realidad virtual puede ayudar a los equipos de diseño e ingeniería a visualizar diseños y estructuras complejas, permitiéndoles identificar y resolver problemas antes de que comience la construcción. También puede permitir a los arquitectos presentar diseños a los clientes de una manera más realista e interactiva.
- **Mayor accesibilidad:** La realidad virtual puede utilizarse para ofrecer experiencias inmersivas a personas que no pueden desplazarse, como ancianos o discapacitados. También puede utilizarse para ofrecer visitas virtuales a lugares inaccesibles, como lugares históricos o museos.

2.5. Retos de la realidad virtual

En general, aunque la tecnología de la realidad virtual tiene mucho potencial, aún se enfrenta a algunos retos importantes. Superarlos exigirá una labor continua de investigación y desarrollo, así como esfuerzos para que la realidad virtual sea más accesible y asequible para todas las personas y organizaciones [18]. He aquí algunos de los retos más destacados que debe superar:

- **Coste:** La tecnología de RV puede ser cara, ya que se necesitan auriculares de gama alta y ordenadores potentes para ejecutar muchas aplicaciones de RV. Esto puede dificultar que particulares y organizaciones inviertan en esta tecnología.
- **Fatiga digital:** Una duración prolongada con cascos de RV provoca fatiga digital y agotamiento. Como la RV es una tecnología basada en el láser, las córneas humanas son reacias a la exposición a los rayos láser, lo que provoca mareos y lagrimeo.
- **Contenido limitado:** Aunque cada vez hay más contenidos de RV disponibles, siguen siendo limitados en comparación con los medios tradicionales. Esto puede limitar el potencial de la RV para convertirse en una opción de entretenimiento generalizada.
- **Aislamiento social:** La RV puede ser una experiencia de aislamiento, ya que los usuarios se ven inmersos en un mundo virtual y aislados del mundo real. Esto puede ser motivo de preocupación para las personas que ya luchan contra el aislamiento social o que utilizan la RV en exceso.
- **Limitaciones técnicas:** La tecnología de la RV sigue evolucionando y existen limitaciones técnicas que deben superarse. Por ejemplo, la tecnología de RV actual tiene dificultades para reproducir el sentido del tacto, lo que limita su capacidad para simular experiencias realistas.

2.6. Proyección de futuro

El futuro de la realidad virtual se presenta como una perspectiva prometedora y llena de avances tecnológicos. Según las proyecciones, se espera un crecimiento significativo en la inversión y adopción de RV y realidad aumentada en los próximos años. Las empresas están integrando cada vez más estas tecnologías en sus planes de transformación digital, lo que indica su importancia en el ámbito empresarial [19].

La demanda del mercado está evolucionando hacia aplicaciones más allá del entretenimiento, el turismo y el marketing, buscando soluciones más accesibles para los usuarios. Se requiere mejorar las interfaces virtuales para intentar minimizar los efectos secundarios que pueden experimentar los usuarios, como el mareo por movimiento. Las principales compañías tecnológicas están trabajando en el desarrollo de auriculares inalámbricos con imágenes en alta definición y procesadores más potentes. Incluso se está explorando la integración de la inteligencia artificial en los sistemas de RV.

La llegada del estándar 5G también tendrá un impacto significativo en el futuro de la realidad virtual. Esta nueva tecnología permitirá una mayor conectividad entre dispositivos y comunidades de usuarios, con una latencia

casi imperceptible. Los consumidores podrán recibir imágenes en tiempo real, brindando una experiencia visual prácticamente indistinguible de lo que se ve con los propios ojos.

Estos avances indican que la realidad virtual ya no es ciencia ficción, sino una realidad integrada en nuestro presente. En los próximos años, se espera que la realidad virtual continúe evolucionando y dando forma al futuro en diversos ámbitos, como la medicina, la educación, la industria, el arte y el entretenimiento. Se vislumbra un futuro en el que la realidad virtual sea aún más inmersiva, interactiva y accesible para las masas, transformando la forma en que vivimos, trabajamos y nos relacionamos con el mundo que nos rodea.

Capítulo 3

Metaverso

Muy posiblemente el Metaverso será el inicio de una nueva era de conectividad digital, experiencia de realidad virtual y comercio electrónico. Prueba de ello, lo encontramos en las fuertes apuestas en el Metaverso por parte de grandes tecnológicas como Microsoft o Meta, antiguo Facebook, como muestra de una gran ambición de su fundador, Mark Zuckerberg, de reimaginar la plataforma de medios sociales como *“una empresa metaverso, que construye el futuro de la conexión social”* [20].

3.1. Concepto

El Metaverso es un espacio 3D compartido en línea en el que los usuarios pueden interactuar entre sí y con objetos y avatares generados por ordenador. Es un mundo virtual que utiliza Internet como red subyacente. Los metaversos se utilizan para diversos fines, como las redes sociales, los juegos en línea, la educación y la formación. Pueden utilizarse para crear mundos virtuales que reflejen el mundo real o para crear mundos totalmente nuevos e imaginarios. Ofrecen una experiencia única y envolvente como ninguna otra en Internet. Están en constante evolución y expansión y ofrecen infinitas posibilidades de exploración e interacción [22].

El metaverso utiliza tecnologías como la realidad virtual y aumentada para crear estas experiencias, pero también puede ser accesible a través de dispositivos más tradicionales, como ordenadores y teléfonos móviles. El objetivo es crear un universo virtual colectivo donde los usuarios puedan interactuar y colaborar en un espacio que trasciende la realidad física.

3.1.1. Tipos de metaversos

Los metaversos son entornos virtuales que permiten a los usuarios interactuar entre sí en un espacio tridimensional. Estos entornos se crean utilizando diversas tecnologías y plataformas, y pueden diferir significativamente en cuanto a sus características y capacidades.

Una forma de clasificar los metaversos es en función de su nivel de centralización [22]. **Los metaversos centralizados** son aquellos en los que una única entidad u organización controla el mundo virtual. Estas organizaciones tienen pleno control sobre los datos de los usuarios y la economía virtual del metaverso. Ejemplos de metaversos centralizados son Second Life y Los Sims.

En cambio, **los metaversos descentralizados** son aquellos en los que el control del mundo virtual está distribuido entre los propios usuarios. Estos metaversos se construyen utilizando la tecnología blockchain, que permite a los usuarios poseer y comerciar con activos virtuales de forma segura y transparente. Los metaversos descentralizados suelen diseñarse como Organizaciones Autónomas Descentralizadas, lo que significa que los usuarios gobiernan colectivamente el metaverso y toman decisiones sobre su futuro desarrollo. Dentro de la categoría de metaversos descentralizados, existen dos tipos principales de metaversos: los totalmente descentralizados y los parcialmente descentralizados.

Los metaversos totalmente descentralizados son aquellos en los que todos los aspectos del mundo virtual están controlados por los usuarios, incluida la creación de contenidos, la gobernanza de la plataforma y la gestión de la economía virtual. Ejemplos de metaversos totalmente descentralizados son Decentraland, The Sandbox y Cryptovoxels.

Por otro lado, los metaversos parcialmente descentralizados son aquellos en los que algunos aspectos del mundo virtual están controlados por una autoridad u organización central, mientras que otros son controlados por los usuarios. Estos metaversos suelen utilizar un enfoque híbrido, en el que la autoridad central gestiona ciertos aspectos de la plataforma, como la infraestructura central, mientras que los usuarios crean y gestionan los contenidos y la economía virtual. Algunos ejemplos de metaversos parcialmente descentralizados son Somnium Space y NeosVR.

3.2. Características

Algunas de las características principales de este nuevo mundo digital son:

■ Son espacios interactivos

El metaverso es un espacio virtual interactivo en el que los usuarios pueden interactuar tanto con otros usuarios como con el universo virtual en sí mismo. Además, está sujeto a leyes físicas y existe escasez de recursos, lo que lo convierte en un entorno corpóreo similar al mundo real.

■ Es persistente y autónomo por sí mismo

El metaverso es persistente y autónomo, lo que significa que sigue funcionando independientemente de que los usuarios estén o no conectados a él. Es un organismo vivo en el que las dinámicas del mundo virtual siguen funcionando incluso cuando los usuarios no están presentes.

■ Está descentralizado

Es descentralizado, lo que significa que no es propiedad de una sola empresa o plataforma, sino de todos sus usuarios, quienes también pueden controlar sus datos privados. La tecnología blockchain es una parte fundamental de esto, ya que garantiza la seguridad y transparencia de las transacciones.

■ Sin límites

El metaverso es un espacio virtual sin límites, lo que significa que no hay barreras físicas o de otro tipo que limiten su uso. Es un espacio infinito en el que cualquier persona puede participar en actividades y economías virtuales descentralizadas impulsadas por criptomonedas, como la compra, venta e intercambio de activos digitales.

■ Economías virtuales

Los usuarios pueden participar en economías virtuales descentralizadas en las que pueden comprar, vender e intercambiar activos digitales como avatares, ropa virtual, NFT (tokens no fungibles) o entradas para eventos. Estos activos digitales son propiedad exclusiva de sus dueños y pueden ser comercializados en los mercados virtuales del metaverso.

■ Personalización y autoexpresión

Los usuarios pueden personalizar sus avatares, los objetos virtuales que poseen y los entornos virtuales que habitan. Esto permite la autoexpresión y un sentimiento de propiedad dentro del metaverso.

■ Compatibilidad entre plataformas

Los metaversos son accesibles a través de múltiples dispositivos y plataformas, incluyendo teléfonos inteligentes, tablets, gafas de realidad virtual y ordenadores. Esto permite una mayor base de usuarios y una mayor accesibilidad.

■ Experiencias inmersivas y atractivas

El metaverso ofrece experiencias inmersivas y atractivas mediante el uso de tecnologías como la RV, la RA y la retroalimentación háptica. Los usuarios pueden sentir que están realmente presentes en el entorno virtual, lo que da lugar a una experiencia más intensa y realista.

- **Potencial para aplicaciones en el mundo real**

Los metaversos tienen potencial para ser utilizados en aplicaciones del mundo real, como la educación, la sanidad y los negocios. Pueden proporcionar un entorno seguro y controlado para fines de formación y simulación.

- **Privacidad y seguridad**

Los metaversos tienen el potencial de proporcionar mayor privacidad y seguridad a los usuarios, ya que las transacciones e interacciones dentro del metaverso se registran en una cadena de bloques (blockchain) y, por tanto, son más seguras y transparentes. Los usuarios también pueden controlar sus datos personales y decidir quién tiene acceso a ellos.

3.3. Cómo acceder al Metaverso

Para acceder al metaverso, primero hay que entender que no existe un único dispositivo o tecnología universal que permita acceder a todos los mundos virtuales que existen. Cada plataforma tiene su propia pasarela a la que se puede acceder desde un ordenador, una tablet, un teléfono móvil o unas gafas de realidad virtual. Actualmente, existen varias opciones disponibles para acceder al metaverso [22], entre las que se encuentran:

- **Los cascos de realidad virtual** son dispositivos avanzados que los usuarios se ponen en los ojos para ver e interactuar con el metaverso. Existen diferentes opciones de cascos, como la línea Meta Quest de Meta, que antes se conocía como Oculus Quest.
- **Los equipos de realidad aumentada** son similares a los de realidad virtual, pero en lugar de crear un mundo completamente digital, superponen elementos digitales al mundo real. Los equipos de realidad aumentada son más flexibles, lo que permite a los usuarios una mayor libertad de movimiento.
- A veces se utilizan **aplicaciones móviles** para acceder al metaverso sin equipos de realidad virtual o aumentada. A algunas plataformas del metaverso se puede acceder desde un smartphone sin necesidad de hardware adicional. Por ejemplo, se puede acceder a Roblox en smartphones con o sin tecnología de RV.
- **Las consolas de videojuegos** como PlayStation y Xbox One pueden utilizarse para acceder al metaverso. Se puede acceder, por ejemplo a Roblox, sin gafas de realidad virtual, pero con ellas se pueden desbloquear otros mundos del metaverso.
- **La mayoría de los ordenadores** pueden ejecutar sistemas de RV y RA, pero para disfrutar de la mejor experiencia, un ordenador debe tener buenos gráficos y al menos 2 GB de RAM.

3.4. Futuro

El futuro del metaverso es emocionante y prometedor. A medida que la tecnología avanza, se espera que el metaverso se convierta en un espacio virtual cada vez más inmersivo, interactivo y social. También se espera que el metaverso evolucione para convertirse en una plataforma sólida para la educación, el comercio y el entretenimiento [21].

Además, se espera que las economías virtuales dentro del metaverso crezcan y se desarrollen, impulsadas por las criptomonedas y la tecnología blockchain, permitiendo a los usuarios poder ganar dinero real participando en el metaverso y creando y vendiendo activos digitales.

Sin embargo, todavía hay algunas barreras importantes que deben superarse antes de que el Metaverso pueda convertirse en una realidad generalizada. Una de las principales barreras es la infraestructura tecnológica. Para que el Metaverso funcione de manera efectiva, se necesitan dispositivos de realidad virtual más cómodos y ligeros, así como una red de comunicación rápida y fiable. En particular, la expansión del 5G es fundamental para garantizar que los usuarios puedan acceder al Metaverso sin problemas.

Otro tema crítico es el tema de la privacidad de los datos. La gestión de la información personal y la ética en el uso de los datos son temas muy delicados en la era de Internet. Los consumidores son cada vez más conscientes de

los riesgos que conlleva compartir información personal en línea, y es probable que esta preocupación se extienda al Metaverso.

En definitiva, aunque el Metaverso es una idea emocionante, todavía hay algunos obstáculos importantes que deben superarse antes de que se convierta en una realidad generalizada. Es probable que se necesiten algunos años más de desarrollo antes de que el Metaverso pueda comenzar a ser implantado gradualmente, y es importante abordar cuidadosamente cuestiones como la privacidad de los datos y la ética en su diseño y funcionamiento.

Capítulo 4

Entorno y herramientas tecnológicas

En este capítulo, se proporcionará un resumen detallado de las tecnologías utilizadas tanto en el desarrollo de la aplicación Educaverse como en la gestión del proyecto en su conjunto.

4.1. Unity

Unity es un motor de videojuegos multiplataforma desarrollado por Unity Technologies [23]. Es utilizado para crear videojuegos en 2D y 3D, así como aplicaciones interactivas y experiencias de realidad virtual y aumentada. Proporciona una amplia gama de herramientas y funcionalidades para el desarrollo de videojuegos, incluyendo un editor visual, un motor de física, un motor de renderizado, un motor de animación, un sistema de scripting, herramientas de audio, y soporte para diferentes plataformas, como PC, consolas y dispositivos móviles, de realidad virtual y de realidad aumentada.



Figura 4.1: Logo de Unity

Los motores de juego, son software especial que los desarrolladores utilizan para crear videojuegos tanto en 2D como en 3D. Como motor de juegos, Unity es una gran plataforma para diseñar mundos imaginarios, calcular la física para que los objetos de estos mundos se muevan, salten o choquen, reproducir sonidos y mucho más [24]. Como la mayoría de los motores de juego, Unity tiene cinco componentes.

- Programa de lógica del juego: Este componente contiene la lógica y el comportamiento del juego, incluyendo las reglas, las interacciones y las mecánicas del juego.
- Motor de renderizado: Es responsable de generar los gráficos animados en 3D que se visualizan en el juego. Utiliza técnicas avanzadas de renderizado para crear efectos visuales realistas y atractivos.
- Motor de audio: Proporciona algoritmos y herramientas para crear y manipular sonidos en el juego. Permite la reproducción de música, efectos de sonido y la integración de audio en el entorno del juego.
- Motor de física: Se encarga de simular las leyes físicas en el juego, como el movimiento de los objetos, las colisiones y las interacciones. Calcula la dinámica de los cuerpos y aplica fuerzas y restricciones para simular el comportamiento realista de los objetos en el mundo del juego.
- Componente de Inteligencia Artificial: Este componente permite crear y controlar el comportamiento de los personajes no jugadores (NPCs) en el juego. Proporciona herramientas y algoritmos para la toma de decisiones, el movimiento autónomo y la interacción de los NPCs con el entorno y los jugadores.

Estos componentes trabajan juntos para ofrecer una experiencia de juego completa y envolvente. Unity proporciona una interfaz de desarrollo integrada (IDE) que permite a los desarrolladores acceder a todas las herramientas necesarias en un solo lugar. También ofrece un Asset Store donde los desarrolladores pueden compartir y descargar activos, como gráficos, sonidos y scripts, para acelerar el proceso de desarrollo.

La arquitectura modular y orientada a objetos de Unity permite a los desarrolladores extender y personalizar el motor de juego según sus necesidades. Pueden contribuir al desarrollo del motor de juego y compartir su propio software middleware para mejorar y ampliar las capacidades del motor.

4.1.1. Interfaz de Unity

La interfaz de Unity es una interfaz fácil de usar e intuitiva que permite crear y editar objetos de juego, componentes y escenas.

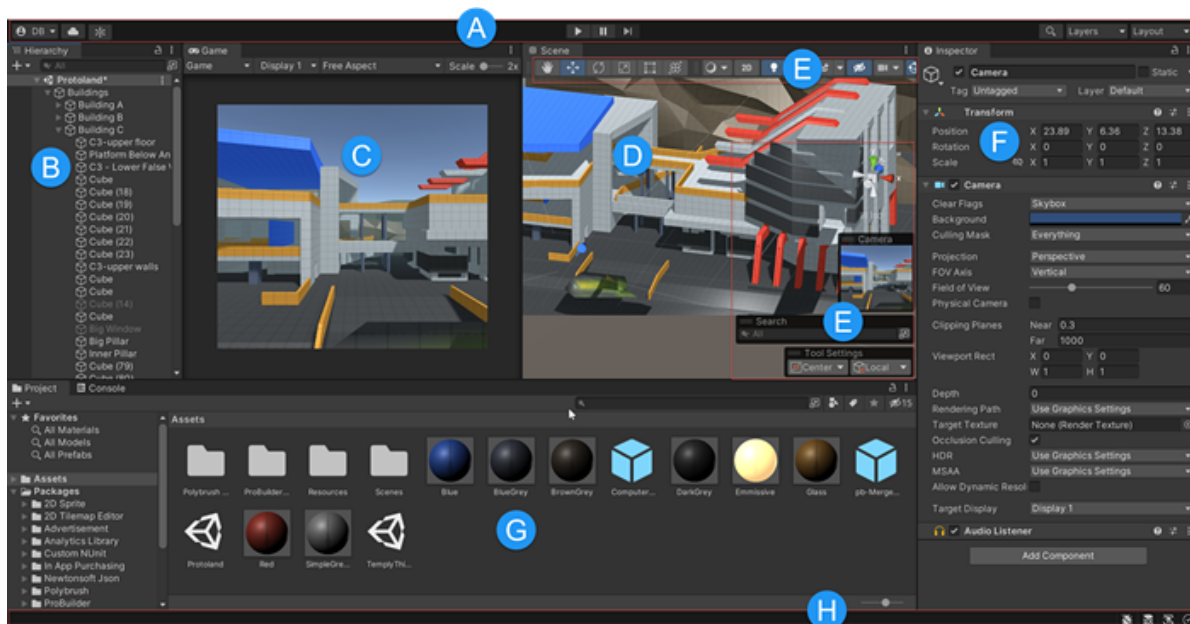


Figura 4.2: Interfaz de Unity. Fuente [25]

Como se puede ver en la Figura 4.2, se identifican las siguientes secciones:

- A) **Toolbar**: Proporciona un acceso rápido a las herramientas y comandos más utilizados, como la navegación por la escena, la creación de objetos y la selección de componentes.
- B) **Hierarchy window**: Muestra una lista jerárquica de todos los GameObjects en la escena actual revelando cómo los objetos se unen entre sí. Se utiliza para organizar los objetos, crear relaciones padre-hijo entre objetos y ajustar el orden de representación de los objetos.
- C) **Game view**: Simula el aspecto que tendrá el juego final renderizado a través de las cámaras de escena. Al hacer clic en el botón Reproducir, comienza la simulación.
- D) **Scene view**: Área principal donde los desarrolladores pueden crear, ver y editar sus objetos y escenas de juego. Proporciona un panel de vista 3D que permite a los desarrolladores moverse por la escena, ajustar la configuración de la cámara y manipular objetos.
- E) **Overlays**: Contienen las herramientas básicas para manipular la Scene View y los GameObjects dentro de ella. También se pueden añadir Overlays personalizados para mejorar el flujo de trabajo.
- F) **Inspector window**: Muestra información sobre el objeto o componente del juego seleccionado en ese momento. Los desarrolladores pueden utilizarla para ver y ajustar las propiedades y configuraciones de los objetos, como su posición, rotación, escala y materiales.

- G) **Project window:** Muestra todos los activos y recursos que están actualmente disponibles para el proyecto. Se utiliza para importar y gestionar activos, como modelos 3D, texturas y archivos de audio.
- H) **Status bar:** Proporciona notificaciones acerca de varios procesos de Unity, y acceso rápido a herramientas y configuraciones relacionadas.

4.1.2. Términos Generales

Tras describir las secciones de la interfaz, procedemos a explicar una serie de conceptos [26] que se han utilizado anteriormente:

- **GameObject:** El objeto fundamental en las escenas de Unity que puede representar personajes, accesorios, escenarios, cámaras, puntos de ruta, etc. La funcionalidad de un GameObject está definida por los componentes adjuntos a él.
- **Scene:** Una Escena contiene los entornos y menús del juego. En cada Escena, se colocan los entornos, obstáculos y decoraciones, esencialmente diseñando y construyendo el juego por piezas.
- **Prefabs:** Tipo especial de componente que permite guardar GameObjects completamente configurados en el Proyecto para su reutilización. Estos activos pueden ser compartidos entre escenas, o incluso otros proyectos sin tener que ser configurados de nuevo. Esto es muy útil para los objetos que se utilizarán muchas veces, como las plataformas. Una gran ventaja de los Prefabs es que son esencialmente copias enlazadas del activo que existe en la ventana del proyecto, lo que significa que los cambios realizados o aplicados al Prefab original se propagarán a todas las demás instancias. Esto hace que corregir errores de objeto, cambiar el arte o hacer otros cambios estilísticos sea muy eficiente.
- **Assets:** Un activo es cualquier elemento que se utiliza en un proyecto Unity para crear un juego o aplicación. Los activos pueden representar elementos visuales o de audio en el proyecto, como modelos 3D, texturas, sprites efectos de sonido o música. Los activos también pueden representar elementos más abstractos como gradientes de color, máscaras de animación, o texto arbitrario o datos numéricos para cualquier uso.
- **Cameras:** Componente que crea una imagen de un punto de vista concreto de la escena. El resultado se dibuja en la pantalla o se captura como textura.

4.1.3. Características y componentes de Unity

■ Gráficos 2D y 3D

En el caso de los juegos 3D, emplean geometría tridimensional para representar entornos sólidos con materiales y texturas. Se utiliza una cámara para capturar la perspectiva, con objetos que parecen más grandes si se colocan más cerca de la cámara. Por otro lado, los juegos 3D ortográficos emplean una cámara ortográfica, que proporciona una vista de pájaro de la acción. En cambio, los juegos 2D utilizan gráficos planos sin perspectiva de cámara. Sin embargo, los juegos 2D pueden incorporar gráficos 3D para ofrecer una experiencia de juego más elegante y profunda [23].

■ Física

El motor de física de Unity permite a los desarrolladores garantizar una aceleración precisa de los objetos y manejar diversas fuerzas de la naturaleza, incluidas las colisiones de objetos y la gravedad. Unity tiene dos tipos de motores de física, siendo el motor de física orientado a objetos el predeterminado. Este motor se ejecuta en un único hilo y un único núcleo. Sin embargo, la pila de tecnología orientada a datos de Unity es una arquitectura interna más reciente que es más rápida, ligera y optimizada para multihilos.

■ Scripting

Unity considera el scripting como un componente crucial en su framework. Se refiere al código C# que responde a las entradas del usuario, organiza la jerarquía de objetos y programa eventos para su ejecución oportuna. Además, los scripts de Unity se encargan de generar efectos visuales y regular el comportamiento físico de los objetos. También gobiernan el sistema de inteligencia artificial de los personajes dentro de la experiencia.

■ Audio y Vídeo

Unity ofrece una amplia compatibilidad con la reproducción de audio y vídeo para mejorar la experiencia de juego en general. En cuanto al audio, Unity permite a los desarrolladores importar y editar archivos de audio dentro del software. Los archivos de audio se pueden reproducir, pausar o detener durante el juego, y los efectos de sonido se pueden activar en respuesta a eventos específicos del juego. También es posible controlar el volumen, el tono y la espacialización de las fuentes de audio para crear paisajes sonoros envolventes. Para la reproducción de vídeo, Unity es compatible con una amplia gama de formatos de vídeo, incluyendo MP4, MOV y AVI. Los vídeos pueden reproducirse en el juego como parte de las escenas o como elementos de fondo para mejorar la atmósfera del juego. Los desarrolladores pueden manipular la reproducción de vídeo, incluyendo el control de la velocidad de reproducción y la reproducción de vídeos en bucle. Unity también admite texturas de vídeo, que permiten mostrar vídeos en objetos del juego.

■ Animación

La animación es un componente crucial en el marco de desarrollo de juegos de Unity, que permite a los desarrolladores crear experiencias de juego visualmente atractivas y dinámicas. En Unity, la animación funciona manipulando las propiedades de un objeto del juego a lo largo del tiempo, como su posición, rotación y escala, para crear la ilusión de movimiento. El proceso de animación en Unity implica la creación de animaciones a través de la animación de fotogramas clave o secuencias de comandos. La animación de fotogramas clave implica especificar posiciones de un objeto en diferentes puntos en el tiempo, mientras que el scripting permite a los desarrolladores crear animaciones a través de código [23].

Unity también ofrece un sofisticado sistema de animación llamado Mecanim [27], que simplifica el proceso de animación y permite a los desarrolladores crear complejas animaciones de personajes con el mínimo esfuerzo. Mecanim (Figura 4.3) permite definir clips de animación y mezclar entre ellos, creando transiciones suaves entre diferentes animaciones. Además, Unity es compatible con la animación esquelética, que se utiliza habitualmente en la animación de personajes. La animación esquelética implica la animación de personajes mediante la manipulación de los huesos y las articulaciones de un personaje, proporcionando un movimiento y una apariencia más realistas.

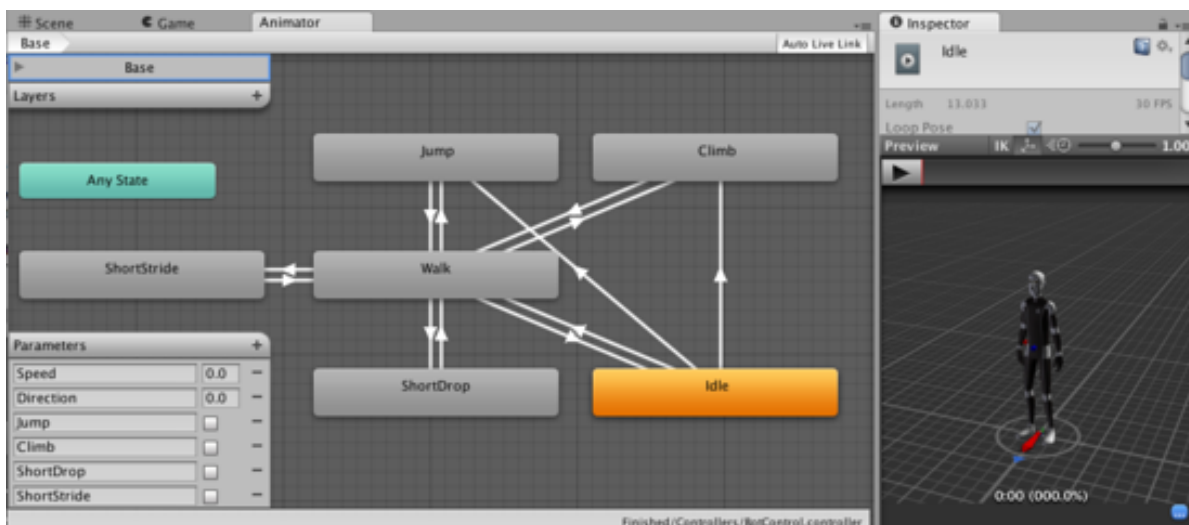


Figura 4.3: Controlador de animaciones para gestionar estados. Fuente [27]

■ Funciones de realidad virtual y aumentada

En el caso de la realidad virtual, Unity brinda un amplio soporte a través de una extensa gama de paquetes que se adaptan prácticamente a todos los auriculares de realidad virtual disponibles. Estos paquetes se actualizan regularmente para mantenerse alineados con el panorama tecnológico en constante evolución, asegurando flexibilidad y compatibilidad. Además, Unity permite a los desarrolladores probar cómodamente juegos de realidad virtual dentro de su motor.

En el ámbito de la realidad aumentada, Unity también se destaca al ofrecer varios paquetes diseñados específicamente para ARCore y ARKit. En particular, Unity ha desarrollado AR Foundation, un marco que capacita a los desarrolladores para crear aplicaciones de realidad aumentada para Android e iOS simultáneamente, eliminando la necesidad de proyectos de desarrollo separados [28].

Además, Unity ha presentado XR Interaction Toolkit, un recurso valioso que simplifica el desarrollo de juegos de RV y RA. Con el compromiso inquebrantable de Unity con las tecnologías XR, queda claro que están a la vanguardia del apoyo y el avance de estas experiencias inmersivas.

4.2. Unity Hub

Unity Hub [29] es una herramienta de gestión centralizada para software y proyectos Unity. Proporciona un flujo de trabajo optimizado para la gestión de múltiples versiones y proyectos de Unity, permitiendo a los desarrolladores cambiar rápidamente entre diferentes versiones y gestionar las dependencias del proyecto.

Una de las funcionalidades más importantes de Unity Hub es la capacidad de gestionar las instalaciones de Unity. Con Unity Hub, los desarrolladores pueden descargar e instalar varias versiones de Unity simultáneamente, lo que facilita el trabajo en proyectos que requieren diferentes versiones del software. Unity Hub también proporciona acceso a versiones beta y experimentales de Unity, lo que permite a los desarrolladores probar nuevas características y funcionalidades.



Figura 4.4: Logo de Unity Hub

Otra característica clave de Unity Hub es la posibilidad de gestionar proyectos de Unity. Los desarrolladores pueden crear y organizar proyectos dentro de Unity Hub, facilitando el seguimiento de múltiples proyectos y sus dependencias. Unity Hub también proporciona un acceso rápido a la configuración del proyecto, como la configuración de gráficos y renderizado, y permite a los desarrolladores abrir proyectos en versiones específicas de Unity.

Unity Hub también ofrece herramientas de colaboración basadas en la nube, lo que facilita que los equipos trabajen juntos en los proyectos. Los desarrolladores pueden compartir proyectos con los miembros del equipo y colaborar en los cambios en tiempo real, utilizando herramientas como Git y Unity Collaborate [30].

Por último, Unity Hub también proporciona acceso a Unity Asset Store [31], un mercado para descargar y compartir activos, plugins y herramientas para el desarrollo de Unity. Los desarrolladores pueden buscar y descargar activos directamente desde Unity Hub, lo que facilita la integración de activos en sus proyectos.

En general, Unity Hub es una potente herramienta de gestión para el desarrollo de Unity que proporciona a los desarrolladores un flujo de trabajo optimizado para la gestión de múltiples versiones, proyectos y activos. Ayuda a simplificar el proceso de desarrollo y permite a los desarrolladores centrarse en la creación de experiencias atractivas e inmersivas.

4.3. C#

C# [32] es un lenguaje de programación de alto nivel y propósito general desarrollado por Microsoft. C# se diseñó para crear aplicaciones robustas y escalables en el marco de la plataforma .NET y se utiliza principalmente para desarrollar aplicaciones de escritorio, aplicaciones web y servicios web.

Entre las principales características de C# [33] se incluyen:



Figura 4.5: Logo de C#

- **Tipado fuerte:** es un lenguaje de programación tipado estáticamente, lo que significa que se requiere que todas las variables se declaren con un tipo específico antes de su uso.
- **Orientación a objetos:** es un lenguaje de programación orientado a objetos, lo que significa que utiliza objetos para modelar el mundo real y ofrece todas las características de la programación orientada a objetos, como encapsulamiento, herencia y polimorfismo.
- **Eficiente:** es un lenguaje de programación eficiente y rápido que ofrece una ejecución de código optimizada en comparación con otros lenguajes de programación.
- **Plataforma independiente:** se puede utilizar en diferentes plataformas, incluyendo Windows, Linux y macOS.
- **Interoperable:** es interoperable con otros lenguajes de programación, lo que significa que se puede utilizar junto con otros lenguajes de programación como Java y C++.
- **Sintaxis legible:** tiene una sintaxis legible y fácil de entender que facilita la lectura y el mantenimiento del código.
- **Seguridad:** tiene características de seguridad integradas, como la administración de memoria automatizada y la prevención de vulnerabilidades comunes.

Las principales ventajas de C# incluyen:

- **Productividad:** es un lenguaje de programación de alto nivel que permite desarrollar aplicaciones complejas con menos líneas de código.
- **Comunidad:** tiene una gran comunidad de desarrolladores y una amplia gama de herramientas y recursos disponibles en línea.
- **Plataforma .NET:** es parte de la plataforma .NET de Microsoft, que ofrece una amplia gama de bibliotecas y herramientas para el desarrollo de aplicaciones.
- **Escalabilidad:** es un lenguaje de programación escalable que se adapta bien a proyectos grandes y complejos.
- **Rentabilidad:** es un lenguaje de programación gratuito y de código abierto que no requiere licencias costosas.

4.4. Meta

El metaverso de Facebook [34] es una plataforma digital que permite a los usuarios adentrarse en un mundo virtual inmersivo, donde pueden interactuar con amigos, familiares y otras personas en un entorno tridimensional sin preocuparse por las limitaciones de la ubicación.



Figura 4.6: Logo de Meta

A través del Metaverso, Facebook pretende transformar la forma en que los usuarios socializan, compran o juegan. Una de las características principales de Meta es que pretende que los usuarios utilicen avatares 3D como una representación virtual de su yo físico.

Las monedas digitales también desempeñan un papel fundamental en el metaverso de Facebook, ya que todas las transacciones se realizan digitalmente. Esto dice mucho de la importancia de las criptomonedas y la cadena de bloques en esta era de innovación tecnológica. El metaverso de Facebook no es sólo para jugar y socializar; también incluye economía, identidades digitales, gobierno descentralizado y otros usos para facilitar los juegos y las redes sociales.

Meta permite a los usuarios teletransportarse de un lugar a otro, como pasar de una tienda a otra sección para comprar artículos virtuales. Debido a su enorme escala y a la capacidad de mantener diversos tipos de datos, como objetos, identidades, derechos, interacciones, pagos e historial, el metaverso de Facebook es único.

4.4.1. Proyectos

- **Horizon Home & Horizon Worlds**

Horizon Home [36] es uno de los elementos fundacionales más importantes del metaverso de Facebook que proporciona un hogar virtual a los usuarios de los auriculares Oculus Quest 2. Al principio era una experiencia limitada, pero ha evolucionado hasta convertirse en un hogar virtual más completo. La integración con Horizon Worlds [37] permite a los usuarios invitar a otras personas a sus hogares virtuales, que luego pueden utilizarse como punto de partida para explorar otros mundos virtuales. Además, con las secuencias de comandos de Horizon Worlds, la experiencia se vuelve más interactiva, lo que permite a los usuarios personalizar sus casas virtuales y hacerlas más atractivas. En conjunto, Horizon Home es un elemento esencial del metaverso de Facebook que sirve de centro para que los usuarios accedan a otras experiencias virtuales y las exploren.

- **Horizon Works**

Horizon Works [38] no es un programa único, sino un conjunto de aplicaciones y una metodología. Del mismo modo que se utiliza un ordenador y un sistema operativo para acceder a paquetes ofimáticos, Horizon Works proporcionará acceso a sistemas relacionados con la productividad dentro del metaverso de Facebook. Horizon Works puede aportar un nuevo tipo de productividad al lugar de trabajo, haciendo posible esbozar rápidamente notas para tus compañeros de trabajo con poco más que un pensamiento, o incluso contar con plena integración IDP/SSO, gestión de cuentas e incluso aplicaciones de terceros disponibles en todo momento [35].

- **Horizon Venues**

Horizon Venues [39] es una plataforma de realidad virtual que pretende revolucionar la industria del entretenimiento creando experiencias inmersivas y sociales para los usuarios. La plataforma permite a los usuarios acceder a eventos en directo, conciertos y otros espectáculos de entretenimiento a través de sus cascos de realidad virtual. El objetivo de Horizon Venues es crear un sentimiento de comunidad y acercar a las personas, independientemente del lugar del mundo en el que se encuentren. La plataforma está diseñada para ofrecer una experiencia más personalizada e interactiva, que permita a los usuarios interactuar con los artistas y otros asistentes.

- **Fitness**

Según Meta, la industria del fitness está dispuesta a utilizar tecnología punta como la realidad virtual para hacer más agradable el entrenamiento. A medida que aumenta el número de personas que prefieren los entornos virtuales a los gimnasios reales (especialmente tras la pandemia), dispositivos como Meta Quest 2 pueden convertirse en su equipo de entrenamiento preferido. En el metaverso de Facebook se incluirán planes de actividades físicas, como demuestra el lanzamiento del FitXR [40] Quest 2.

- **Educación**

El metaverso de Facebook promete un futuro apasionante para la educación. Ofrece un aula virtual que permite a los estudiantes acceder a una amplia gama de recursos y herramientas. En este espacio virtual, los cirujanos pueden formarse y mejorar sus habilidades, mientras que los técnicos pueden diseñar máquinas utilizando sólo sus manos. El potencial educativo del metaverso no se limita a la instrucción técnica, ya que el espacio virtual es intrínsecamente social. Facebook pretende crear una experiencia de aprendizaje totalmente colaborativa que una a las personas. VictoryXR [41] es uno de los sistemas educativos más destacados actualmente en el metaverso.

- **Spark AR**

Spark AR es una tecnología que permite experiencias de realidad aumentada dentro del metaverso de Facebook. Permite colocar objetos virtuales en el mundo real y los usuarios pueden interactuar con ellos de diversas maneras. Con Spark AR, Meta pone mucho énfasis en el aspecto social de la realidad aumentada. Por ejemplo, los usuarios pueden crear búsquedas del tesoro u otras experiencias interactivas que pueden disfrutar con sus amigos. La app de Polar permite incluso a los no programadores crear sus propios contenidos, lo que facilita la participación de cualquiera en la creación de estas experiencias [35].

4.5. GitLab

GitLab es un gestor de repositorios Git basado en web que proporciona a los desarrolladores una plataforma para la gestión del código fuente, el control de versiones, la integración y el despliegue continuo (CI/CD) y la colaboración [42].



Figura 4.7: Logo de GitLab

Ahora bien, Git es un popular sistema de control de versiones distribuido cuyas características se basan en la confirmación de nuevos cambios, la ramificación, fusión y la comparación de versiones anteriores. Es eficiente y sencillo, pues tiene la capacidad de ser usado de forma básica, además de ofrecer potencial, cantidad de prestaciones y control sobre el proyecto.

GitLab proporciona una interfaz web intuitiva para trabajar con repositorios Git, facilitando la gestión del código y la colaboración entre equipos. Además del control de versiones basado en Git, GitLab ofrece una serie de funciones para gestionar el ciclo de vida del desarrollo de software. Ofrece herramientas para la gestión de proyectos, como tableros de tareas, seguimiento de hitos y seguimiento de incidencias, que permiten a los equipos planificar y organizar su trabajo de forma eficaz. También ofrece herramientas de revisión de código, como las solicitudes de fusión, que permiten a los equipos revisar y fusionar cambios en el código de forma rápida y eficaz.

Otra característica importante de GitLab es su funcionalidad CI/CD. GitLab ofrece un sistema CI/CD integrado que permite a los desarrolladores automatizar sus procesos de compilación, prueba y despliegue. Esta función puede utilizarse para crear procesos personalizados de creación y despliegue de cambios en el código, automatizar tareas repetitivas y mejorar la productividad del equipo.

GitLab también ofrece funciones avanzadas de seguridad, como el análisis estático de código integrado y la gestión de vulnerabilidades, para ayudar a los equipos a garantizar que su código sea seguro. También ofrece integración con diversas herramientas y servicios de terceros, lo que facilita a los equipos la integración de su flujo de trabajo de desarrollo con otras herramientas y servicios que utilicen.

Finalmente, estas son algunas de las ventajas [43] que ofrece esta herramienta:

- **Facilidad de configuración y administración:** GitLab es una herramienta fácil de configurar y administrar, y sus códigos están disponibles de forma gratuita.
- **Optimización del flujo de trabajo:** GitLab crea un flujo de trabajo optimizado que mejora la colaboración y eficiencia de los equipos.
- **Revisiones de código y solicitudes de extracción:** La herramienta ofrece revisiones de código y solicitudes de extracción que son fáciles de usar y compactas.
- **Seguridad:** GitLab es una aplicación nativa de la nube que cuenta con fuertes medidas de seguridad, como restricciones granulares y autenticación de usuario con *Kerberos*.

- **Integración mínima:** La herramienta permite una integración mínima que reduce el ciclo de vida del desarrollo y aumenta la productividad.
- **Gestión de proyectos:** GitLab facilita una gestión de proyectos amplia y adaptable para acelerar el flujo de trabajo.

4.6. Astah

Para las fases de análisis y diseño se ha utilizado Astah [44], herramienta de modelado visual y diseño de software que permite crear diagramas y modelos para representar y visualizar diferentes aspectos de un sistema o proyecto. Astah admite una variedad de notaciones y diagramas, incluyendo diagramas de casos de uso, diagramas de clases, diagramas de secuencia, diagramas de actividades, entre otros.



Figura 4.8: Logo de Astah Professional

Con Astah Professional, se pueden diseñar y representar de forma gráfica la estructura, el comportamiento y las relaciones entre los componentes de un sistema. Se pueden construir modelos conceptuales, especificaciones de requisitos, diagramas de flujo, diagramas de base de datos y muchos otros tipos de diagramas utilizados en el desarrollo de software y otros proyectos técnicos.

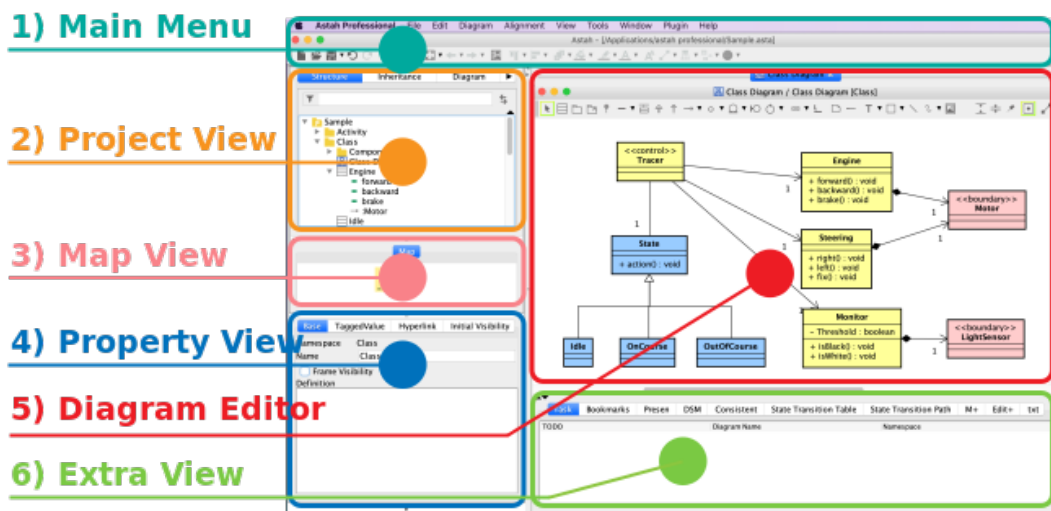


Figura 4.9: Interfaz de Astah Professional. Fuente [45]

En cuanto a la interfaz del programa 4.9, esta es fácil de utilizar y cuenta con un alto grado de facilidad de aprendizaje. Consta de las siguientes secciones [45]:

- **Menú principal:** Esta sección contiene una lista de opciones y comandos básicos del programa que permiten acceder a diferentes funciones y configuraciones.
- **Vista del proyecto:** Es una sección donde se muestra la información del modelo en forma de árbol o lista. Aquí se puede buscar y establecer alias para facilitar la navegación y organización del proyecto.
- **Vista Mapa:** Esta sección actúa como un controlador para visualizar y navegar por los diferentes diagramas del proyecto. Permite cambiar entre diagramas y enfocar áreas específicas del proyecto.

- **Vista de propiedades:** Aquí se puede editar y ver información detallada del elemento o modelo seleccionado. Proporciona una manera de ajustar configuraciones específicas y personalizar características de los elementos del proyecto.
- **Editor de diagramas:** Es un lienzo donde se dibujan y construyen los diferentes diagramas del proyecto. Proporciona herramientas y opciones para crear y modificar elementos visuales del diagrama.
- **Vista Extra:** Esta sección es opcional y se añade después de instalar plugins que requieren una vista adicional. Puede contener tablas de transición de estados, herramientas específicas u otros complementos adicionales que agregan funcionalidades extra al programa.

4.7. Overleaf

Para radactar la memoria del TFG se ha utilizado Overleaf [46]. Se trata de un editor en línea de textos en formato LaTeX que permite la colaboración en tiempo real y la escritura de documentos científicos, técnicos, tesis, informes y otros documentos académicos.



Figura 4.10: Logo de Overleaf

Algunas de las ventajas de Overleaf son:

- **Visualización en tiempo real:** Los cambios realizados se ven instantáneamente, lo que evita solapamientos y permite trabajar de manera conjunta y a distancia.
- **Fácil de usar:** Aunque LaTeX requiere cierto aprendizaje de código, Overleaf ofrece un editor de texto enriquecido similar a los editores tradicionales, lo que facilita su uso y reduce la curva de aprendizaje.
- **Amplia selección de plantillas:** Overleaf ofrece numerosas plantillas preinstaladas de los mejores journals, lo que permite seleccionar la adecuada desde el principio y ahorrar tiempo en la adaptación del texto.
- **Acceso desde cualquier lugar:** Al ser un editor en línea, se puede acceder y escribir en Overleaf desde cualquier lugar con conexión a internet, incluso sin tener instalado el programa en el dispositivo.

Capítulo 5

Planificación

Esta sección se centrará principalmente en la organización del trabajo para cumplir con los requisitos fundamentales del proyecto. Para ello se realizará una metodología ágil basada en iteraciones incrementales. Por tanto, se aproximarán unas horas de trabajo adecuadas para cumplir con todas las necesidades.

5.1. Planificación inicial

La realización de este TFG se llevará a cabo durante el segundo cuatrimestre del curso 2022-2023. La primera reunión con los tutores tiene lugar el día 23 de noviembre de 2022, mientras que la última iteración finalizará el 15 de junio. De esta manera se cuentan con 204 días para poder realizar el proyecto. Durante este período se trabajará de lunes a viernes, contando con los fines de semana libres de carga de trabajo, pero disponibles para poder realizar horas extras si así fuera necesario.

Como se explicó en la sección 1.4, se ha seguido una distribución ágil que promueve la flexibilidad y la adaptación a medida que se avanza en el proyecto. Se estableció un Kick-off inicial con los tutores que marcó el inicio del proyecto y se realizó una planificación de la primera iteración. Se decidió que al final de cada iteración, se decidirán los objetivos de la siguiente en función de los resultados de cada iteración. En cuanto al número de iteraciones, se establecieron seis iteraciones a lo largo del proyecto, lo que implica dividir el trabajo en ciclos cortos y manejables para obtener resultados iterativos y mejorar continuamente el sistema. A continuación se muestra la propuesta inicial:

- **Educaverse Kick-off (12 semanas, 60 horas)**

En esta primera reunión tendrá lugar la presentación del proyecto, donde se establecerán los objetivos, alcance y expectativas del TFG. Se discutirán los resultados y se establecerán los criterios de éxito del proyecto, como las herramientas de desarrollo o la metodología. Durante este tiempo, el principal trabajo consistirá en familiarizarse con la tecnología que se empleará, mediante documentación y realización de cursos.

- **1ª Iteración (4 semanas, 40 horas)**

Durante la primera iteración se llevará a cabo la configuración adecuada del entorno de trabajo y se realizará un análisis detallado de los requisitos del proyecto, los cuales podrán ir cambiando para incluir o excluir funcionalidad. El entregable para esta primera iteración consistirá en crear una versión inicial del proyecto y definir los requisitos del sistema.

- **2ª Iteración (3 semanas, 40 horas)**

Durante la segunda iteración, el trabajo se centrará en el diseño del proyecto, incluyendo la definición de la arquitectura, la creación de los diagramas correspondientes y la estructuración del código. El entregable para esta iteración consistirá en completar el diseño del proyecto y definir la interfaz de usuario.

■ **3ª Iteración (3 semanas, 45 horas)**

El trabajo principal durante la tercera iteración se basará en utilizar las herramientas de desarrollo de Unity para crear un prototipo funcional de la interfaz de usuario en 3D, además de investigar acerca de la implementación de los comandos de voz en el proyecto. El entregable para esta iteración consistirá en crear un prototipo de la escenografía inicial y añadir detección de voz.

■ **4ª Iteración (3 semanas, 45 horas)**

Durante la cuarta iteración, el trabajo consistirá en conseguir recursos para llevar a cabo la implementación del Avatar médico en la escena y se investigará acerca de la reproducción de los vídeos en Unity. El entregable para esta iteración consistirá en añadir el Avatar en la sala médica y reproducir un vídeo.

■ **5ª Iteración (3 semanas, 45 horas)**

Durante la quinta iteración, el trabajo principal consistirá en completar la interfaz con la que el usuario interactuará e investigar acerca del streaming de los vídeos. El entregable para esta iteración consistirá en llevar a cabo mejoras en la interfaz y experiencia de usuario.

■ **6ª Iteración (2 semanas, 25 horas)**

Durante la última iteración, se realizarán pruebas exhaustivas del sistema para identificar posibles errores o problemas y corregirlos. Se documentarán y elaborarán manuales de instrucciones de uso, configuración y cualquier otra información relevante. El entregable para esta iteración consistirá en realizar las pruebas finales, implementar la realidad virtual en la aplicación, completar la documentación y realizar la entrega final del TFG.

A continuación, se presenta (Figura 5.1) el diagrama de Gantt [47], herramienta de gestión de proyectos que permite visualizar de manera gráfica las tareas a realizar, su duración y las fechas de inicio y finalización. Se representa mediante una barra horizontal para cada tarea, donde se muestra la duración y la relación secuencial entre las tareas. El Diagrama de Gantt facilita la planificación, seguimiento y control de proyectos, permitiendo identificar la secuencia y duración de las actividades, así como detectar posibles solapamientos o retrasos en el cronograma. En dicha figura, se muestra el proceso del proyecto detallando las iteraciones.

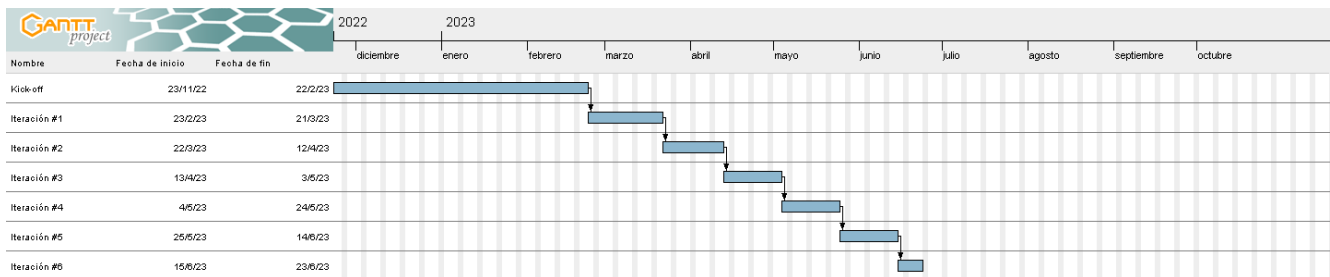


Figura 5.1: Diagrama de Gantt de las iteraciones del proyecto

5.1.1. Análisis de riesgos

En el desarrollo de cualquier proyecto, es fundamental reconocer que existen incertidumbres y posibles obstáculos que pueden afectar su éxito. La gestión de riesgos es un proceso clave que permite identificar, evaluar y abordar estos riesgos de manera proactiva [48].

En este apartado, se abordará en detalle la gestión de riesgos. Se identificarán los riesgos específicos asociados al proyecto y se describirán las estrategias para mitigarlos. Además, se establecerán planes de contingencia para estar preparado en caso de que ocurra algún riesgo significativo. Al abordar la gestión de riesgos de manera integral, estaremos fortaleciendo la probabilidad de éxito y reduciendo las posibles interrupciones en el desarrollo de la aplicación.

A continuación, se describen algunos de los posibles riesgos identificados más importantes para el proyecto, rellenando así una tabla por cada uno de ellos. Estas tablas nos sirven para recoger de forma completa y organizada

información para cada uno de estos riesgos. Para dotar a cada riesgo de una importancia, se ha optado por utilizar la matriz de probabilidad e impacto:

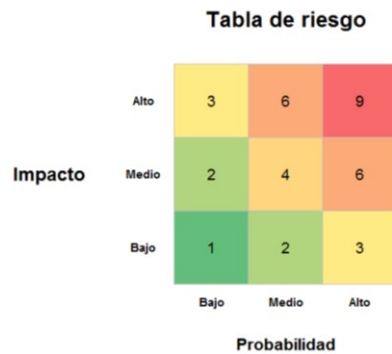


Figura 5.2: Matriz de Riesgos

Retrasos en la adquisición de recursos técnicos	
Descripción	Implica la posibilidad de demoras en la obtención de los recursos técnicos necesarios para el desarrollo de la aplicación, como la batería de vídeos.
Probabilidad Impacto Riesgo	Baja - 1 Medio - 2 Medio - 2
Acciones de Mitigación	Realizar pedidos anticipados del material.
Plan de Contingencia	Ajustar el cronograma.

Tabla 5.1: Riesgo 01 - Retrasos en la adquisición de recursos técnicos

Cambios en los requisitos	
Descripción	Posibilidad de que los requisitos de la aplicación cambien durante el desarrollo del proyecto, lo que podría requerir modificaciones significativas en el diseño y la implementación.
Probabilidad Impacto Riesgo	Medio - 2 Medio - 2 Medio - 4
Acciones de Mitigación	Aplicar la metodología ágil en el desarrollo software para reducir el impacto que supondrían los cambios.
Plan de Contingencia	En la reuniones, revisar el trabajo realizado y planificar una depuración del software, así como la implementación de los cambios.

Tabla 5.2: Riesgo 02 - Cambios en los requisitos

Problemas de compatibilidad con las plataformas del metaverso	
Descripción	Existe el riesgo de que la aplicación de realidad virtual encuentre dificultades para ser compatible con las plataformas del metaverso seleccionadas, lo que podría requerir ajustes y pruebas adicionales.
Probabilidad Impacto Riesgo	Medio - 2 Medio - 2 Medio - 4
Acciones de Mitigación	Realizar investigaciones exhaustivas sobre las plataformas del metaverso y asegurarse de que la aplicación cumpla con los requisitos técnicos y las directrices de compatibilidad.
Plan de Contingencia	Asignación de tiempo adicional para pruebas y ajustes o la exploración de alternativas de plataforma si es necesario.

Tabla 5.3: Riesgo 03 - Problemas de compatibilidad con las plataformas del Metaverso

Desmotivación a la hora de trabajar	
Descripción	El paso del tiempo puede originar cambios emocionales que pueden causar problemas en el desarrollo del proyecto.
Probabilidad Impacto Riesgo	Media - 2 Alto - 2 Alto - 4
Acciones de Mitigación	Fomentar un ambiente de comunicación abierto junto con los tutores donde el estudiante pueda expresar sus preocupaciones, ideas y necesidades.
Plan de Contingencia	Fomentar un equilibrio saludable entre el trabajo y la vida personal, evitando la sobrecarga de trabajo excesiva.

Tabla 5.4: Riesgo 04 - Desmotivación a la hora de trabajar

Mala estimación de tiempo	
Descripción	Realizar una mala estimación del tiempo necesario para completar las actividades del proyecto, lo que podría resultar en retrasos en la entrega final.
Probabilidad Impacto Riesgo	Media - 2 Medio - 2 Medio - 4
Acciones de Mitigación	<ul style="list-style-type: none"> ■ Realizar un análisis exhaustivo de todas las tareas y sub-tareas involucradas en el proyecto, identificando los posibles puntos de complejidad. ■ Priorizar la entrega del producto mínimo viable para validar la idea y obtener retroalimentación temprana, dejando aspectos menos importantes para etapas posteriores del proyecto
Plan de Contingencia	<ul style="list-style-type: none"> ■ Monitoreo y seguimiento constante. ■ Definir el alcance del proyecto acorde al tiempo disponible.

Tabla 5.5: Riesgo 05 - Mala estimación de tiempo

Avería del ordenador	
Descripción	El ordenador utilizado para el desarrollo del proyecto experimenta una avería, lo que supone la interrupción de las tareas y retrasos en el progreso del proyecto.
Probabilidad Impacto Riesgo	Bajo - 1 Medio - 2 Medio - 2
Acciones de Mitigación	<ul style="list-style-type: none"> ■ Realizar copias de seguridad periódicas de todos los archivos y documentos relevantes del proyecto, almacenándolos en dispositivos externos o en la nube. ■ Realizar un mantenimiento regular del ordenador, como actualizaciones de software, limpieza de archivos temporales y verificación del estado del hardware.
Plan de Contingencia	Mantener un equipo de respaldo disponible en caso de una avería del ordenador principal.

Tabla 5.6: Riesgo 06 - Avería del ordenador

Complejidad del proyecto	
Descripción	El proyecto presenta un nivel de complejidad mayor al esperado inicialmente lo que dificulta la ejecución del proyecto dentro de los plazos establecidos.
Probabilidad Impacto Riesgo	Bajo - 1 Medio - 2 Medio - 2
Acciones de Mitigación	Realizar un análisis exhaustivo y detallado del proyecto antes de su inicio, identificando y comprendiendo todas las posibles complejidades y desafíos asociados.
Plan de Contingencia	<ul style="list-style-type: none"> ■ En caso de que la complejidad del proyecto se vuelva demasiado desafiante, realizar una revisión del alcance para identificar elementos menos críticos o funcionalidades opcionales que puedan ser eliminados o postergados. ■ Mantener un contacto continuo con los tutores del proyecto, quienes pueden dar soluciones innovadoras para abordar los desafíos complejos.

Tabla 5.7: Riesgo 07 - Complejidad del proyecto

Uso de una tecnología con la que no se ha trabajado anteriormente	
Descripción	Las tecnologías empleadas para el desarrollo del proyecto son novedosas para el estudiante.
Probabilidad Impacto Riesgo	Medio - 2 Medio - 2 Medio - 4
Acciones de Mitigación	Realizar una investigación detallada de la tecnología que se va a utilizar antes de su implementación.
Plan de Contingencia	Obtener conocimientos a través de la documentación oficial o externa de la herramienta o tecnología al momento de su implementación en el proyecto.

Tabla 5.8: Riesgo 08 - Uso de una tecnología con la que no se ha trabajado anteriormente

Contracción de enfermedades	
Descripción	El estudiante contrae una enfermedad durante el desarrollo del proyecto lo que afecta su participación y desempeño en el mismo.
Probabilidad Impacto Riesgo	Media - 2 Medio - 2 Medio - 4
Acciones de Mitigación	<ul style="list-style-type: none"> ■ Mantener un ambiente limpio y libre de desechos, proporcionar suficiente ventilación y asegurarse de que los espacios de trabajo estén bien iluminados y sean cómodos. ■ Evitar actividades peligrosas o que puedan provocar algún tipo de incidente.
Plan de Contingencia	<ul style="list-style-type: none"> ■ Si la duración de la enfermedad es corta, redistribuir las horas perdidas en los próximos días. ■ Si la duración de la enfermedad es más duradera, traspasar la tarea al siguiente sprint.

Tabla 5.9: Riesgo 09 - Contracción de enfermedades

Inexperiencia en la gestión de proyectos	
Descripción	El estudiante tiene poca experiencia con la metodología establecida lo que dificulta la capacidad del estudiante para establecer metas claras, asignar recursos adecuadamente y cumplir con los plazos establecidos.
Probabilidad Impacto Riesgo	Media - 2 Alto - 3 Alto - 6
Acciones de Mitigación	Aprender sobre la metodología Agile con Scrum mediante talleres, cursos en línea o charlas con el tutor.
Plan de Contingencia	<ul style="list-style-type: none"> ■ Adaptación de los sprint. ■ Aprender sobre la marcha la metodología a base de prueba y error.

Tabla 5.10: Riesgo 10 - Inexperiencia en la gestión de proyectos

5.1.2. Plan de presupuesto

Como parte integral de la etapa de planificación, resulta imprescindible desarrollar un plan de presupuesto exhaustivo para el proyecto [49]. Dicho plan implica un análisis minucioso de los costos asociados al proyecto en diferentes niveles. Estos niveles comprenden el coste de personal, la amortización del equipo informático, los costes de equipos de realidad virtual y software, el espacio de trabajo y el consumo eléctrico. A continuación se presenta un desglose de los mismos:

- **Costes de personal.** En primer lugar, se atribuye el desarrollo de este proyecto a un Desarrollador. Con el fin de realizar una estimación lo más realista posible, se ha buscado información sobre el coste por hora trabajada [50]. La información obtenida es la siguiente: el salario de Desarrollador promedio en España es de 31.500€ al año o 16,15€ por hora. Por ello, obtenemos que el sueldo de un programador con estas características es de 2.625€ al mes. En nuestro caso, con un trabajo estimado de 356 horas, el coste total del salario bruto del desarrollador sería de 5.749,4€.

- **Amortización del equipo informático.** El equipo utilizado es un HP Pavilion Laptop de 2019. Con un precio de compra en 2019 de 1050€, y un valor actual de mercado que ronda los 650€, tras 4 años, el gasto anual de amortización es de $(1050 - 650)€ / 4 \text{ años} = 100€/\text{año}$. Tras la conversión correspondiente, equivale a 0,0115€/hora, que, al utilizarlo durante 356 horas, se obtiene un total de 4,094€.
- **Costes de equipos de RV y software.** Los costes de equipos de RV abarcan tanto el hardware como los dispositivos necesarios para la experiencia de realidad virtual, como cascos o mandos. Estos dispositivos fueron adquiridos a través de la colaboración con HP SCDS. Por otro lado, los costes de software incluyen las licencias necesarias para utilizar el software de realidad virtual y las herramientas de desarrollo asociadas al proyecto. En este caso, las licencias de software fueron proporcionadas de forma gratuita debido al estatus de estudiante, lo que permitió acceder a las funcionalidades y capacidades necesarias para el desarrollo del proyecto sin incurrir en costes adicionales.
- **Espacio de trabajo.** La mayoría del trabajo realizado para este proyecto se ha realizado en el hogar personal del estudiante, y para poder realizar el trabajo es necesario contar con conexión a Internet. El plan contratado es de unos 150,00€ al mes y en esta unidad familiar conviven 4 personas, por lo que el coste de Internet para el estudiante es de 37.5€/mes. Realizando la conversión correspondiente, esto equivale a 0,052€/hora, que al utilizarlo durante 356 horas, se obtiene un total de 18,51€.
- **Consumo eléctrico.** El análisis de costes del proyecto también incluye la factura de electricidad, que en este caso tiene un coste mensual de 70 euros, incluyendo impuestos, tasas y el consumo de energía en kilovatios-hora (KWh). Por tanto, se obtiene un coste de 0,098€/hora, por lo que tras 356 horas, nos da un total de 34,88€.

Después de realizar un análisis detallado de los costes, se muestran en la tabla siguiente con el propósito de simplificar la interpretación para el lector.

Gasto	Coste/hora	Coste total(356h)
Personal	16,15€	5.749,4€
Amortización	0,0115€	4,094€
Internet	0,052€	18,51€
Consumo eléctrico	0,098€	34,88€
Total	16,31€	5.823,194€

Tabla 5.11: Presupuesto del proyecto

5.1.3. Plan de trabajo

A continuación, se presenta el plan de trabajo que contempla las distintas tareas a realizar durante las iteraciones mencionadas. Este plan ha sido elaborado con el objetivo de establecer una estructura clara y organizada para el desarrollo del proyecto.

Iteración 1

Nombre de la tarea	Duración
Comprensión del trabajo a realizar	4 horas
Análisis de riesgos	3 horas
Análisis de costes	2 horas
Configuración del entorno de trabajo en Unity	3 horas
Establecimiento de requisitos	4 horas
Plan de trabajo	3 horas
Modelo de casos de uso	4 horas
Especificación de casos de uso	5 horas
Modelo de dominio	3 horas
Diagramas de secuencia	4 horas
Redacción de la memoria	5 horas
Total Iteración 1	40 horas

Tabla 5.12: Plan de trabajo de la primera iteración

Iteración 2

Nombre de la tarea	Duración
Arquitectura del sistema	7 horas
Diagramas de paquetes	6 horas
Diagrama de componentes	6 horas
Diagrama de despliegue	6 horas
Diseño de la interfaz de usuario	5 horas
Redacción de la memoria	10 horas
Total Iteración 2	40 horas

Tabla 5.13: Plan de trabajo de la segunda iteración

Iteración 3

Nombre de la tarea	Duración
Conseguir recursos 3D que simulen una consulta médica en Unity	6 horas
Implementación de un objeto 3D	3 horas
Implementar escena de la consulta médica 3D	8 horas
Crear ejecutable inicial que permita rotar la cámara	5 horas
Crear un script que permita mover un objeto 3D por voz	10 horas
Redacción de la memoria	8 horas
Total Iteración 3	40 horas

Tabla 5.14: Plan de trabajo de la tercera iteración

Iteración 4

Nombre de la tarea	Duración
Conseguir recursos 3D para un Avatar en Unity	6 horas
Incorporación del Avatar dentro de la sala médica	4 horas
Movimiento y posición del Avatar médico	2 horas
Implementar comandos de voz para que la interacción con el avatar sea más intuitiva y natural	15 horas
Reproducir un vídeo de forma aleatoria	10 horas
Redacción de la memoria	8 horas
Total Iteración 4	45 horas

Tabla 5.15: Plan de trabajo de la cuarta iteración

Iteración 5

Nombre de la tarea	Duración
Creación del catálogo de vídeos	8 horas
Animación y postura del avatar médico	2 horas
Activar/Desactivar Reconocedor de voz	4 horas
Proporcionar información acerca de los vídeos que aparecen en el catálogo	5 horas
Añadir audio al avatar médico	2 horas
Mostrar un catálogo de vídeos cuyo contenido sea el del comando introducido	4 horas
Mostrar material videográfico cuando se reconoce alguna frase de voz y corresponde con algún comando válido	4 horas
Investigar sobre reconocedor de dictados para que la interacción con el avatar sea más intuitiva y natural	4 horas
Investigar sobre la reproducción de vídeo en streaming para una visualización de alta calidad sin interrupciones ni retrasos en la descarga	4 horas
Controlar la reproducción de los vídeos	4 horas
Redacción de la memoria	9 horas
Total Iteración 5	50 horas

Tabla 5.16: Plan de trabajo de la quinta iteración

Iteración 6

Nombre de la tarea	Duración
Implementación de la realidad virtual en la aplicación	5 horas
Pruebas manuales de funcionalidad	4 horas
Elaboración del manual de instalación	5 horas
Elaboración del manual de usuario	4 horas
Revisión de la memoria	7 horas
Total Iteración 6	25 horas

Tabla 5.17: Plan de trabajo de la sexta iteración

Fases	Horas
Kick-off	60
Iteración 1	40
Iteración 2	40
Iteración 3	40
Iteración 4	45
Iteración 5	50
Iteración 6	25
Total	300 horas

Tabla 5.18: Distribución temporal inicial

5.2. Seguimiento y control

En este apartado, se aborda el seguimiento y control del proyecto, centrándonos en el análisis de los tiempos reales empleados para cada tarea en comparación con los tiempos previstos en la planificación inicial. El objetivo principal de esta sección es examinar las posibles desviaciones entre los tiempos planificados y los tiempos reales, y obtener conclusiones significativas sobre los motivos que originaron dichas desviaciones.

Iteración 1

Nombre de la tarea	Duración	Duración real
Comprensión del trabajo a realizar	4 horas	4 horas
Análisis de riesgos	3 horas	3 horas
Análisis de costes	2 horas	2 horas
Configuración del entorno de trabajo en Unity	3 horas	2 horas
Establecimiento de requisitos	4 horas	6 horas
Plan de trabajo	3 horas	3 horas
Modelo de casos de uso	4 horas	4 horas
Especificación de casos de uso	5 horas	7 horas
Modelo de dominio	3 horas	4 horas
Diagramas de secuencia	4 horas	6 horas
Redacción de la memoria	5 horas	9 horas
Total Iteración 1	40 horas	50 horas

Tabla 5.19: Plan de trabajo real de la primera iteración

Iteración 2

Nombre de la tarea	Duración	Duración real
Arquitectura del sistema	7 horas	7 horas
Diagramas de paquetes	6 horas	4 horas
Diagrama de componentes	6 horas	7 horas
Diagrama de despliegue	6 horas	6 horas
Diseño de la interfaz de usuario	5 horas	7 horas
Redacción de la memoria	10 horas	8 horas
Total Iteración 2	40 horas	39 horas

Tabla 5.20: Plan de trabajo real de la segunda iteración

Iteración 3

Nombre de la tarea	Duración	Duración real
Conseguir recursos 3D que simulen una consulta médica en Unity	6 horas	4 horas
Implementación de un objeto 3D	3 horas	2 horas
Implementar escena de la consulta médica 3D	8 horas	5 horas
Crear ejecutable inicial que permita rotar la cámara	5 horas	4 horas
Crear un script que permita mover un objeto 3D por voz	10 horas	12 horas
Redacción de la memoria	8 horas	8 horas
Total Iteración 3	40 horas	35 horas

Tabla 5.21: Plan de trabajo real de la tercera iteración

Iteración 4

Nombre de la tarea	Duración	Duración real
Conseguir recursos 3D para un Avatar en Unity	6 horas	4 horas
Incorporación del Avatar dentro de la sala médica	4 horas	4 horas
Movimiento y posición del Avatar médico	2 horas	1 hora
Implementar comandos de voz para que la interacción con el avatar sea más intuitiva y natural	15 horas	20 horas
Reproducir un vídeo de forma aleatoria	10 horas	10 horas
Redacción de la memoria	8 horas	10 horas
Total Iteración 4	45 horas	49 horas

Tabla 5.22: Plan de trabajo real de la cuarta iteración

Iteración 5

Nombre de la tarea	Duración	Duración real
Creación del catálogo de vídeos	8 horas	15 horas
Animación y postura del avatar médico	2 horas	2 horas
Activar/Desactivar Reconocedor de voz	4 horas	6 horas
Proporcionar información acerca de los vídeos que aparecen en el catálogo	5 horas	8 horas
Añadir audio al avatar médico	2 horas	2 horas
Mostrar un catálogo de vídeos cuyo contenido sea el del comando introducido	4 horas	6 horas
Mostrar material videográfico cuando se reconoce alguna frase de voz y corresponde con algún comando válido	4 horas	5 horas
Investigar sobre reconocedor de dictados para que la interacción con el avatar sea más intuitiva y natural	4 horas	4 horas
Investigar sobre la reproducción de vídeo en streaming para una visualización de alta calidad sin interrupciones ni retrasos en la descarga	4 horas	8 horas
Controlar la reproducción de los vídeos	4 horas	7 horas
Redacción de la memoria	9 horas	14 horas
Total Iteración 5	50 horas	77 horas

Tabla 5.23: Plan de trabajo real de la quinta iteración

Iteración 6

Nombre de la tarea	Duración	Duración real
Implementación de la realidad virtual en la aplicación	5 horas	10 horas
Pruebas manuales de funcionalidad	4 horas	12 horas
Elaboración del manual de instalación	5 horas	4 horas
Elaboración del manual de usuario	4 horas	5 horas
Revisión de la memoria	7 horas	15 horas
Total Iteración 6	25	46 horas

Tabla 5.24: Plan de trabajo real de la sexta iteración

Fases	Horas	Horas reales
Kick-off	60	60
Iteración 1	40	50
Iteración 2	40	39
Iteración 3	40	35
Iteración 4	45	49
Iteración 5	50	77
Iteración 6	25	46
Total	300 horas	356 horas

Tabla 5.25: Distribución temporal final

5.2.1. Desviaciones en la planificación

Iteración 1

Durante esta primera iteración, se observó una desviación temporal con respecto a la planificación inicial. El tiempo dedicado a esta etapa fue superior al estimado debido a la necesidad de realizar cambios y correcciones en el análisis de la aplicación. Estas modificaciones adicionales generaron una extensión en el tiempo asignado a esta fase.

Iteración 4

En esta iteración en particular, no hubo mucha desviación respecto a lo establecido inicialmente. Sin embargo, se identificó que la tarea relacionada con la mejora de la implementación de los comandos de voz demandó más tiempo de trabajo del previsto originalmente. Esto se debió a la necesidad de realizar varias implementaciones y ajustes para lograr obtener un resultado satisfactorio.

Iteración 5

Durante esta fase, se llevaron a cabo mejoras en la interfaz de usuario y se añadieron nuevas funcionalidades a la aplicación. Este proceso conllevó un desarrollo intenso en Unity, lo que implicó la necesidad constante de consultar la documentación oficial para realizar las tareas pertinentes. Esta búsqueda adicional de información contribuyó a un mayor consumo de tiempo en comparación con lo planificado.

Iteración 6

Finalmente, durante la última iteración se llevó a cabo la implementación de la realidad virtual en la aplicación. No obstante, durante las últimas semanas, surgieron dificultades para disponer de los dispositivos de realidad virtual necesarios para llevar a cabo las pruebas. Como resultado, se optó por realizar el desarrollo y las pruebas utilizando un simulador de dispositivos, detallado en el apartado 8.6.2. Este cambio en el proceso conllevó un retraso y un aumento en el número de horas requeridas en comparación con la planificación inicialmente establecida.

Debido a las desviaciones temporales mencionadas en las iteraciones anteriores y los correspondientes retrasos, el TFG se concluyó dos semanas más tarde de lo previsto. En consecuencia, el TFG se presentó durante la segunda convocatoria.

Capítulo 6

Análisis

En este capítulo, se abordará en detalle el análisis del proyecto, una etapa fundamental en el desarrollo de software, que sienta las bases para el diseño y la implementación de la aplicación. El análisis del proyecto se enfoca en diversas áreas clave, como los actores involucrados, los requisitos, los casos de uso o el modelo de dominio.

6.1. Descripción del sistema

El proyecto consiste en una aplicación 3D basada en Unity que se enfoca en simular una consulta médica interactiva y educativa. Su propósito principal es proporcionar a los usuarios una experiencia inmersiva donde puedan interactuar con un avatar médico y acceder a contenido multimedia relacionado con la diabetes. Además, la aplicación tiene como objetivo ofrecer una experiencia compatible con la realidad virtual y la posibilidad de exportarse al metaverso, lo que amplía su alcance y potencial de distribución.

6.2. Actores de la aplicación

El único actor principal es el usuario, dado que es quien interactúa con la interfaz y utiliza las funcionalidades proporcionadas por el sistema. Es importante comprender las necesidades y expectativas del usuario para diseñar una experiencia de usuario óptima y satisfactoria.

6.3. Requisitos

El siguiente apartado describe los requisitos de la aplicación. Estos requisitos establecen las funcionalidades y características que deben ser implementadas para cumplir con los objetivos de la aplicación. Se pueden clasificar en dos categorías: requisitos funcionales y requisitos no funcionales.

6.3.1. Requisitos Funcionales

Los requisitos funcionales describen las acciones y funcionalidades específicas que el sistema debe realizar para satisfacer las necesidades del usuario. En cuanto a los requisitos funcionales de la aplicación tenemos los siguientes:

ID	Nombre	Descripción
RF01	Interacción avatar médico	El usuario deberá poder pulsar sobre el avatar médico para comenzar a introducir comandos de voz
RF02	Reproducción de audio	El usuario deberá poder escuchar un audio que le proporcione información sobre cómo interactuar con el avatar
RF03	Activar reconocedor de voz	El usuario deberá poder activar el reconocedor de voz
RF04	Reconocimiento de voz	El usuario deberá poder ingresar comandos de voz y obtener una respuesta en función de su validación
RF05	Visualización catálogo de vídeos	El usuario deberá poder visualizar un catálogo con los vídeos disponibles
RF06	Visualizar imágenes asociadas a los vídeos	El usuario deberá poder visualizar las miniaturas de los vídeos que estén disponibles
RF07	Mostrar título	El usuario deberá poder visualizar los títulos de los vídeos que seleccione a través de una ventana emergente
RF08	Reproducción de vídeos	El usuario deberá poder reproducir un vídeo mediante la selección de la correspondiente imagen
RF09	Controlar vídeos	El usuario deberá poder controlar la reproducción de los vídeos según sus necesidades y preferencias
RF10	Salir modo visualización	El usuario deberá poder cerrar el catálogo con los vídeos y volver al inicio de la aplicación

Tabla 6.1: Requisitos funcionales del sistema

6.3.2. Requisitos No Funcionales

Los requisitos no funcionales establecen los criterios y restricciones relacionados con aspectos como el rendimiento, seguridad, usabilidad y otros atributos del sistema. En cuanto a los requisitos no funcionales de la aplicación tenemos los siguientes:

ID	Nombre	Descripción
RNF01	Usabilidad	El sistema deberá ser intuitivo y fácil de usar para que los usuarios puedan interactuar sin dificultad
RNF02	Rendimiento	El sistema deberá ser capaz de manejar y cargar rápidamente los recursos multimedia para una experiencia fluida
RNF03	Compatibilidad	El sistema deberá ser compatible con tecnologías de realidad virtual, asegurando una experiencia inmersiva y facilitando la interacción del usuario en este entorno digital
RNF04	Estabilidad y confiabilidad	El sistema deberá ser estable y no debe presentar bloqueos ni fallos inesperados durante su ejecución
RNF05	Seguridad	El sistema debe garantizar la seguridad de los datos del usuario, protegiendo la privacidad y evitando la manipulación o acceso no autorizado
RNF06	Autorización de vídeos	El sistema deberá contar con un proceso de autorización para evitar la subida de vídeos inapropiados o no adecuados para la aplicación
RNF07	Animación del Avatar médico	El sistema deberá tener animaciones que permitan mejorar la experiencia del usuario
RNF08	Lenguaje claro y comprensible	El sistema deberá utilizar un lenguaje claro y comprensible en su interfaz, mensajes de error e instrucciones, con el objetivo de garantizar una comunicación efectiva y facilitar la comprensión por parte de los usuarios

Tabla 6.2: Requisitos no funcionales del sistema

6.4. Derechos de autor

Durante la realización del proyecto, he tenido acceso a una colección de vídeos educativos proporcionados por HP SCDS. Estos vídeos son propiedad de SCLEDyN (Sociedad Castellano-Leonesa de Endocrinología, Diabetes y Nutrición) [51]. En el marco de esta colaboración, los vídeos educativos suministrados han sido utilizados como recursos fundamentales para el desarrollo de mi TFG. Dichos vídeos han sido seleccionados cuidadosamente debido a su relevancia y calidad en la temática de la diabetes, permitiendo ofrecer a los usuarios de la aplicación una experiencia educativa enriquecedora.

La Sociedad Castellano-Leonesa de Endocrinología, Diabetes y Nutrición es una entidad reconocida en el ámbito de la salud en la comunidad de Castilla y León, con una trayectoria de casi 30 años desde su fundación en junio de 1987. Sus especialistas se dedican al tratamiento de enfermedades como la obesidad, la diabetes mellitus, así como a patologías tiroideas, enfermedades endocrinas complejas y otros trastornos relacionados. Además de su enfoque asistencial, la sociedad colabora con la Administración Sanitaria en diversas iniciativas, como el desarrollo de protocolos médicos, el pilotaje de proyectos y el abordaje integral de enfermedades como la Esclerosis Lateral Amiotrófica. También dedica esfuerzos a la formación y la investigación mediante la organización de un Congreso Anual, talleres, foros de trabajo y cursos monográficos.

Es importante destacar que el acceso y uso de estos vídeos ha sido autorizado y respaldado por HP SCDS y SCLEDyN, quienes han otorgado el permiso correspondiente para su incorporación en este proyecto académico. La utilización de los vídeos se realizará estrictamente dentro del ámbito de este TFG, cumpliendo con los términos y condiciones establecidos por las entidades propietarias. Además de los vídeos proporcionados, se utilizarán otros recursos, como imágenes, que cumplirán con las normas y regulaciones de derechos de autor correspondientes.

6.5. Casos de Uso

6.5.1. Diagrama de casos de uso

En la Figura 6.1 se aporta una vista completa del modelo de casos de uso, el cual ofrece una representación gráfica clara y concisa de las interacciones entre el actor y el sistema. Este diagrama será de gran utilidad para comprender el alcance y la funcionalidad del sistema, así como para identificar las relaciones y las dependencias entre los diferentes casos de uso.

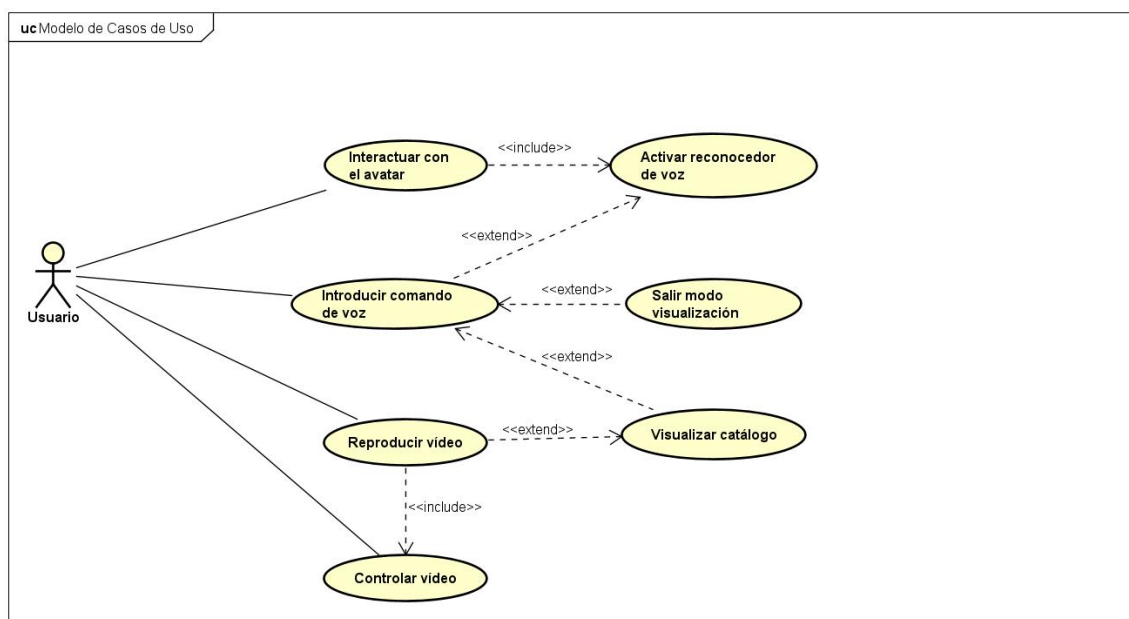


Figura 6.1: Diagrama de casos de uso

6.5.2. Especificación de escenarios

En este apartado, se presentan los casos de uso detallados, que describen el flujo normal de acciones entre el usuario y el sistema. Cada caso de uso especifica quién realiza la acción, en qué paso se encuentra y en qué consiste la acción en sí.

Caso de Uso	Interactuar con el Avatar
Identificador	CU-01
RF relacionado	RF01, RF02
Actor	Usuario
Precondición	El actor usuario deberá haber iniciado la aplicación.
Secuencia normal	<ol style="list-style-type: none"> 1. El actor usuario visualiza el avatar. 2. El sistema emite una fuente de audio que da instrucciones al usuario. 3. El sistema presenta la animación por parte del avatar médico. 4. El sistema crea una nueva instancia del reconocedor de voz. 5. El actor usuario interactúa con el avatar pulsando sobre él.
Alternativas y excepciones	<p>(1a) Si el avatar no está disponible o no es visible, el actor usuario no puede realizar la interacción y el caso de uso queda sin efecto.</p> <p>(5a) Si el actor no pulsa sobre el avatar médico el caso de uso queda sin efecto.</p>
Postcondición	Se ha registrado la interacción del actor usuario con el avatar médico.
Puntos de ampliación	Cuando el usuario pulse sobre el avatar se ejecuta el caso de uso “Activar reconocedor de voz”.

Tabla 6.3: Caso de Uso CU-01

Caso de Uso	Activar Reconocedor de Voz
Identificador	CU-02
RF relacionado	RF03
Actor	Usuario
Precondición	El actor usuario deberá haber interactuado con el avatar médico pulsando sobre él.
Secuencia normal	<ol style="list-style-type: none"> 1. El sistema verifica la disponibilidad del reconocedor de voz. 2. El sistema activa el reconocedor de voz. 3. El sistema registra una función callback para cuando se haya completado el reconocimiento del dictado. 4. El sistema inicia el reconocimiento del dictado.
Alternativas y excepciones	<p>(1a) Si el reconocedor de voz no está disponible o no se puede configurar correctamente, el caso de uso queda sin efecto.</p> <p>(2a) Si el reconocedor de voz ya estaba activo, el caso de uso queda sin efecto.</p>
Postcondición	El reconocedor de voz está activo y puede ser utilizado por el actor usuario.
Puntos de ampliación	Cuando el usuario active el reconocedor de voz se ejecuta el caso de uso “Introducir Comando de voz”.

Tabla 6.4: Caso de Uso CU-02

Caso de Uso	Introducir comando de voz
Identificador	CU-03
RF relacionado	RF04
Actor	Usuario
Precondición	El reconocedor de voz deberá estar activo.
Secuencia normal	<ol style="list-style-type: none"> 1. El actor usuario pronuncia un comando de voz frente al micrófono o dispositivo de entrada de voz. 2. El sistema captura el comando de voz del usuario. 3. El sistema verifica si el comando de voz introducido por el actor usuario coincide con los comandos válidos de la aplicación. 4. El sistema invoca la función que tiene asignada el comando válido detectado. 5. El sistema desactiva el reconocedor de voz.
Alternativas y excepciones	<p>(1a) Si el actor usuario no introduce ningún comando de voz, el caso de uso queda sin efecto.</p> <p>(2a) Si el reconocedor de voz no puede capturar correctamente el comando introducido por el actor usuario, el caso de uso vuelve al paso 1.</p> <p>(3a) Si el comando de voz no coincide con ningún comando válido, se muestra un mensaje indicativo al actor usuario.</p>
Postcondición	Se ha procesado y validado el comando de voz introducido por el actor usuario.
Puntos de ampliación	<ul style="list-style-type: none"> - Si el comando de voz validado coincide con el comando utilizado para cerrar la interfaz que contiene el catálogo de vídeos y además este es visible, entonces se procede a realizar el caso de uso denominado “Salir modo visualización”. - En caso de que el comando de voz validado coincida con un comando destinado a la visualización de vídeos, se procede a llevar a cabo el caso de uso denominado “Visualizar catálogo”.

Tabla 6.5: Caso de Uso CU-03

Caso de Uso	Visualizar catálogo
Identificador	CU-04
RF relacionado	RF05, RF06, RF07
Actor	Usuario
Precondición	El actor usuario deberá haber introducido un comando válido.
Secuencia normal	<ol style="list-style-type: none"> 1. El sistema crea y muestra el catálogo de vídeos asociado al comando detectado. 2. El actor usuario puede explorar y ver las imágenes relacionadas con cada vídeo. 3. El actor usuario puede seleccionar una imagen haciendo clic sobre ella.
Alternativas y excepciones	<p>(2a) Si el actor usuario pasa el cursor o dispositivo de RV sobre una imagen, se muestra una ventana emergente con el título del vídeo correspondiente.</p> <p>(2b) Si el actor usuario quita el cursor o dispositivo de RV sobre una imagen, se oculta la ventana emergente con el título del vídeo correspondiente.</p>
Postcondición	Se ha creado el catálogo de vídeos y el actor usuario lo ha visualizado correctamente en la interfaz gráfica.
Puntos de ampliación	Cuando el usuario selecciona una imagen de las que aparece en el catálogo, se ejecuta el caso de uso “Reproducir vídeo”.

Tabla 6.6: Caso de Uso CU-04

Caso de Uso	Reproducir vídeo
Identificador	CU-05
RF relacionado	RF08
Actor	Usuario
Precondición	El catálogo de vídeos debe haber sido creado.
Secuencia normal	<ol style="list-style-type: none"> 1. El actor usuario selecciona una imagen. 2. El sistema detecta la imagen que el actor usuario ha seleccionado. 3. El sistema oculta el catálogo de vídeos. 4. El sistema carga el archivo de vídeo y lo reproduce dentro de una interfaz gráfica. 5. El sistema muestra nuevamente el catálogo de vídeos cuando finaliza la reproducción.
Alternativas y excepciones	(4a) Si el archivo de vídeo no se puede cargar correctamente, el sistema muestra un mensaje de error y el caso de uso queda sin efecto.
Postcondición	El actor usuario ha visualizado el vídeo seleccionado.
Puntos de ampliación	Cuando el actor usuario esté visualizando el vídeo seleccionado, se activa el caso de uso "Controlar vídeo".

Tabla 6.7: Caso de Uso CU-05

Caso de Uso	Controlar vídeo
Identificador	CU-06
RF relacionado	RF09
Actor	Usuario
Precondición	El actor usuario deberá estar visualizando un vídeo.
Secuencia normal	<ol style="list-style-type: none"> 1. El actor usuario interactúa con los controles de reproducción disponibles. 2. El sistema interpreta la interacción del usuario y realiza la acción correspondiente en la reproducción del vídeo. 3. El sistema actualiza el estado de reproducción del vídeo según la interacción del usuario.
Alternativas y excepciones	<p>(2a) Si el actor usuario pausa la reproducción del vídeo, el sistema detiene temporalmente la reproducción.</p> <p>(2b) Si el actor usuario reanuda la reproducción del vídeo, el sistema continúa reproduciendo el vídeo desde el punto de pausa.</p> <p>(2c) Si el actor usuario adelanta el vídeo, el sistema salta hacia delante en la línea de tiempo del vídeo y continúa reproduciéndolo desde ese punto.</p> <p>(2d) Si el actor usuario retrocede el vídeo, el sistema retrocede en la línea de tiempo del vídeo y continúa reproduciéndolo desde ese punto.</p> <p>(2e) Si el actor usuario cancela el vídeo, el sistema detiene la reproducción del vídeo, oculta la interfaz y vuelve a mostrar el catálogo de vídeos.</p>
Postcondición	El actor usuario ha controlado el vídeo de forma exitosa.
Puntos de ampliación	No procede.

Tabla 6.8: Caso de Uso CU-06

Caso de Uso	Salir modo visualización
Identificador	CU-07
RF relacionado	RF10
Actor	Usuario
Precondición	El catálogo de vídeos debe haber sido creado.
Secuencia normal	<ol style="list-style-type: none"> 1. El actor usuario introduce un comando específico para salir del modo de visualización. 2. El sistema detecta el comando introducido por el actor usuario. 3. El sistema oculta las interfaces gráficas. 4. El sistema desactiva el reconocedor de voz. 5. El actor usuario vuelve al modo inicial de la aplicación.
Alternativas y excepciones	(3a) Si las interfaces gráficas ya estaban inactivas, el sistema no realiza ninguna acción adicional.
Postcondición	El catálogo de vídeos se ha cerrado, el reconocedor de voz está inactivo y el actor usuario vuelve al inicio de la aplicación.
Puntos de ampliación	No procede.

Tabla 6.9: Caso de Uso CU-07

6.6. Modelo de Dominio

La Figura 6.2 muestra el modelo del dominio que se ha obtenido del análisis, donde se aprecian las clases conceptuales significativas extraídas de los requisitos y de los casos de uso especificados. De esta manera, se describen las entidades que las conforman, así como sus atributos y las relaciones con otras entidades.

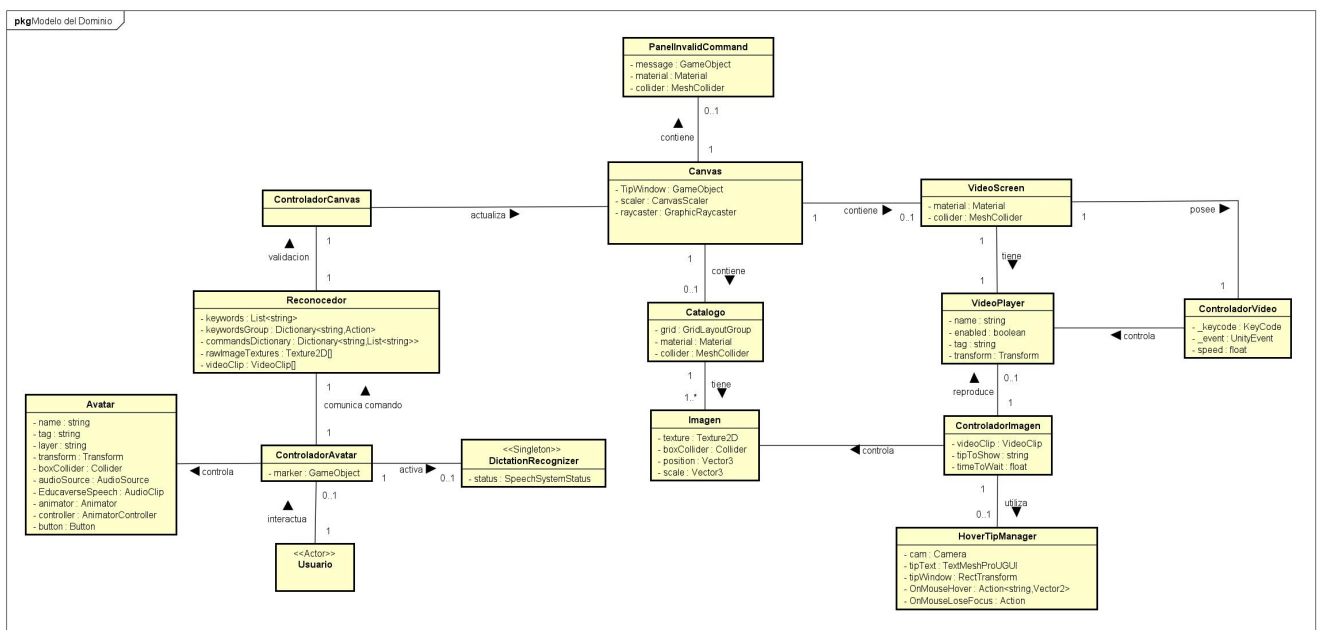


Figura 6.2: Modelo del Dominio

Clases identificadas:

- **Usuario:** Representa el usuario del sistema.
- **Avatar:** Representa al personaje médico con el que el usuario interactúa.

- **ControladorAvatar:** Clase responsable de gestionar y controlar las interacciones y acciones relacionadas con el avatar.
- **DictationRecognizer:** Se encarga de escuchar y analizar el habla del usuario en tiempo real, utilizando tecnologías de reconocimiento de voz. Permite convertir el discurso detectado en texto.
- **Reconocedor:** Se encarga de reconocer y procesar los comandos de voz del usuario, invocando las acciones correspondientes.
- **ControladorCanvas:** Clase encargada de gestionar la interfaz gráfica y la presentación visual de la aplicación.
- **Canvas:** Contenedor visual donde se presentan los elementos de la interfaz de usuario como las imágenes y los vídeos.
- **PanelInvalidCommand:** Clase responsable de mostrar un mensaje adecuado cuando no se ha detectado ningún comando válido.
- **Catalogo:** Representa un catálogo de vídeos, que contiene una colección de imágenes asociadas a los vídeos disponibles.
- **Imagen:** Representa una imagen dentro del catálogo de vídeos.
- **ControladorImagen:** Responsable de detectar las interacciones del usuario con las imágenes.
- **HoverTipManager:** Responsable de gestionar la aparición de ventanas emergentes con información adicional.
- **VideoScreen:** Elemento visual que se utiliza para presentar los vídeos a los usuarios.
- **VideoPlayer:** Clase responsable de cargar los archivos de vídeo y reproducirlos.
- **ControladorVídeo:** Responsable de controlar la reproducción de los vídeos.

6.7. Realización en análisis de los casos de uso

En la Figura 6.3 se muestra la realización en análisis de la secuencia principal del caso de uso Interactuar con el Avatar.

En la Figura 6.4 se muestra la realización en análisis de la secuencia principal del caso de uso Activar Reconocedor de Voz.

En la Figura 6.5 se muestra la realización en análisis de la secuencia principal del caso de uso Introducir comando de voz.

En la Figura 6.6 se muestra la realización en análisis de la secuencia principal del caso de uso Visualizar catálogo.

En la Figura 6.7 se muestra la realización en análisis de la secuencia principal del caso de uso Reproducir vídeo.

En la Figura 6.8 se muestra la realización en análisis de la secuencia principal del caso de uso Controlar vídeo.

En la Figura 6.9 se muestra la realización en análisis de la secuencia principal del caso de uso Salir modo visualización.

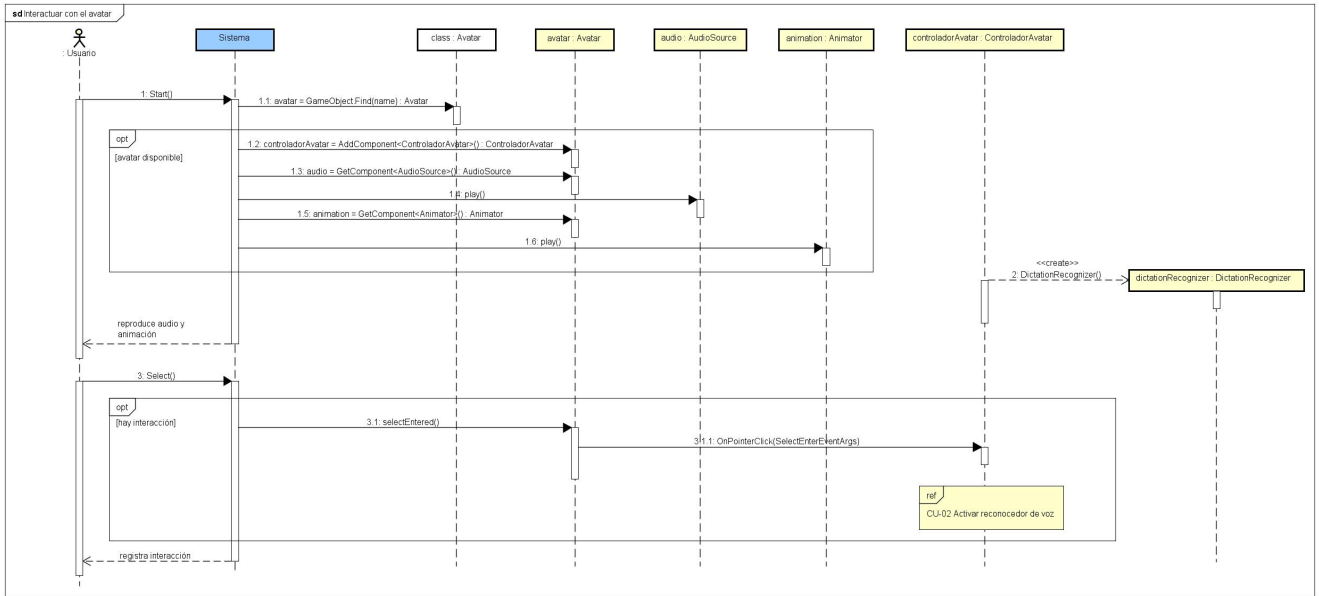


Figura 6.3: Realización en análisis del caso de uso Interactuar con el Avatar

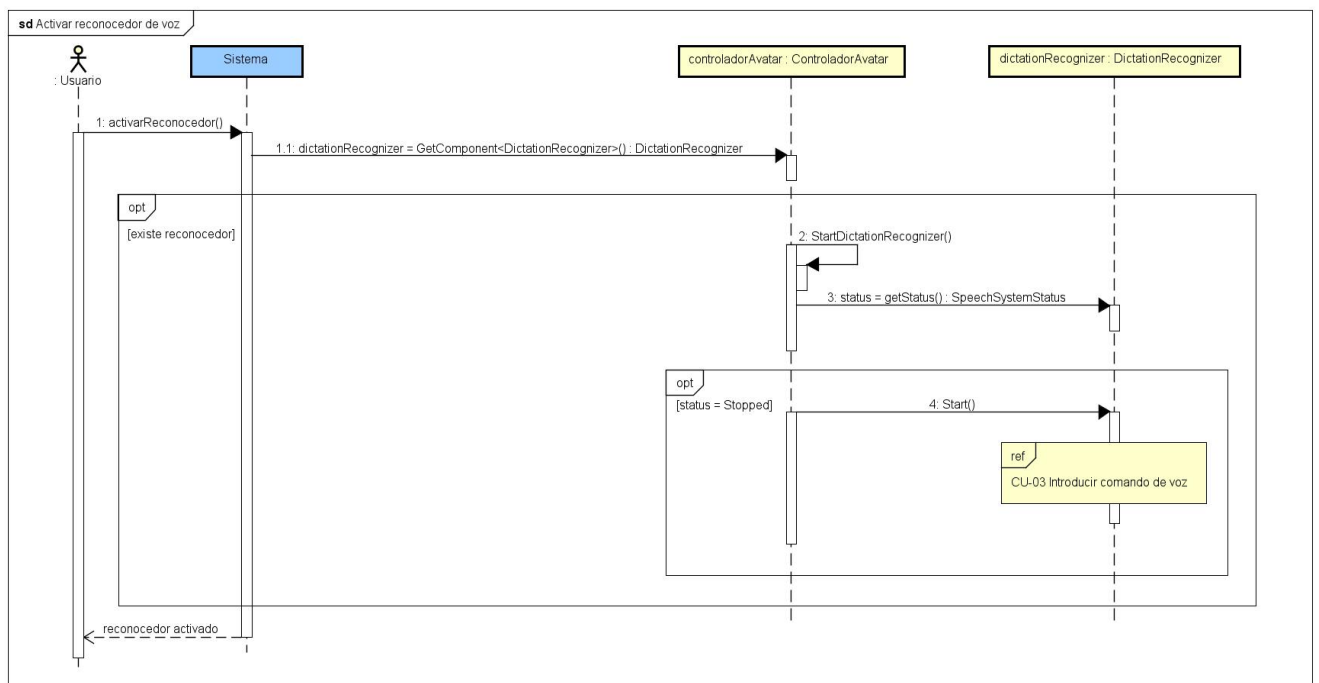


Figura 6.4: Realización en análisis del caso de uso Activar Reconocedor de Voz

6.7. REALIZACIÓN EN ANÁLISIS DE LOS CASOS DE USO

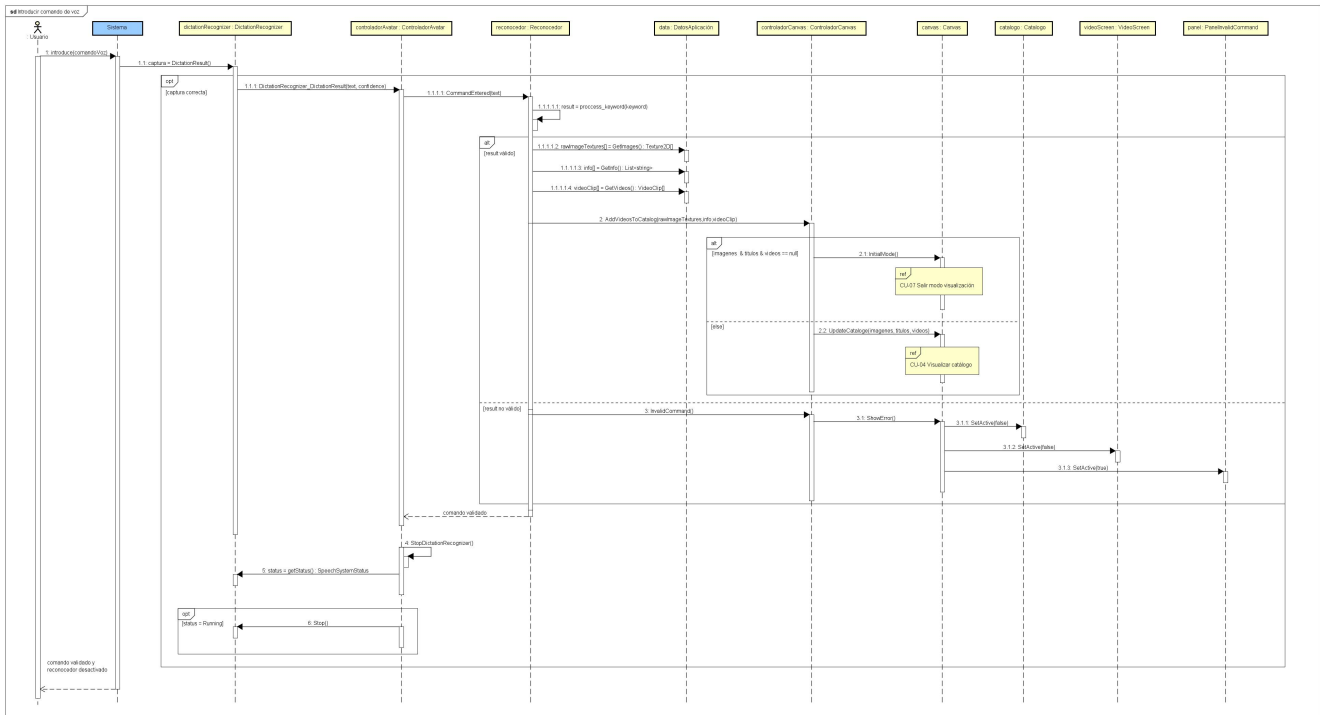


Figura 6.5: Realización en análisis del caso de uso Introducir comando de voz

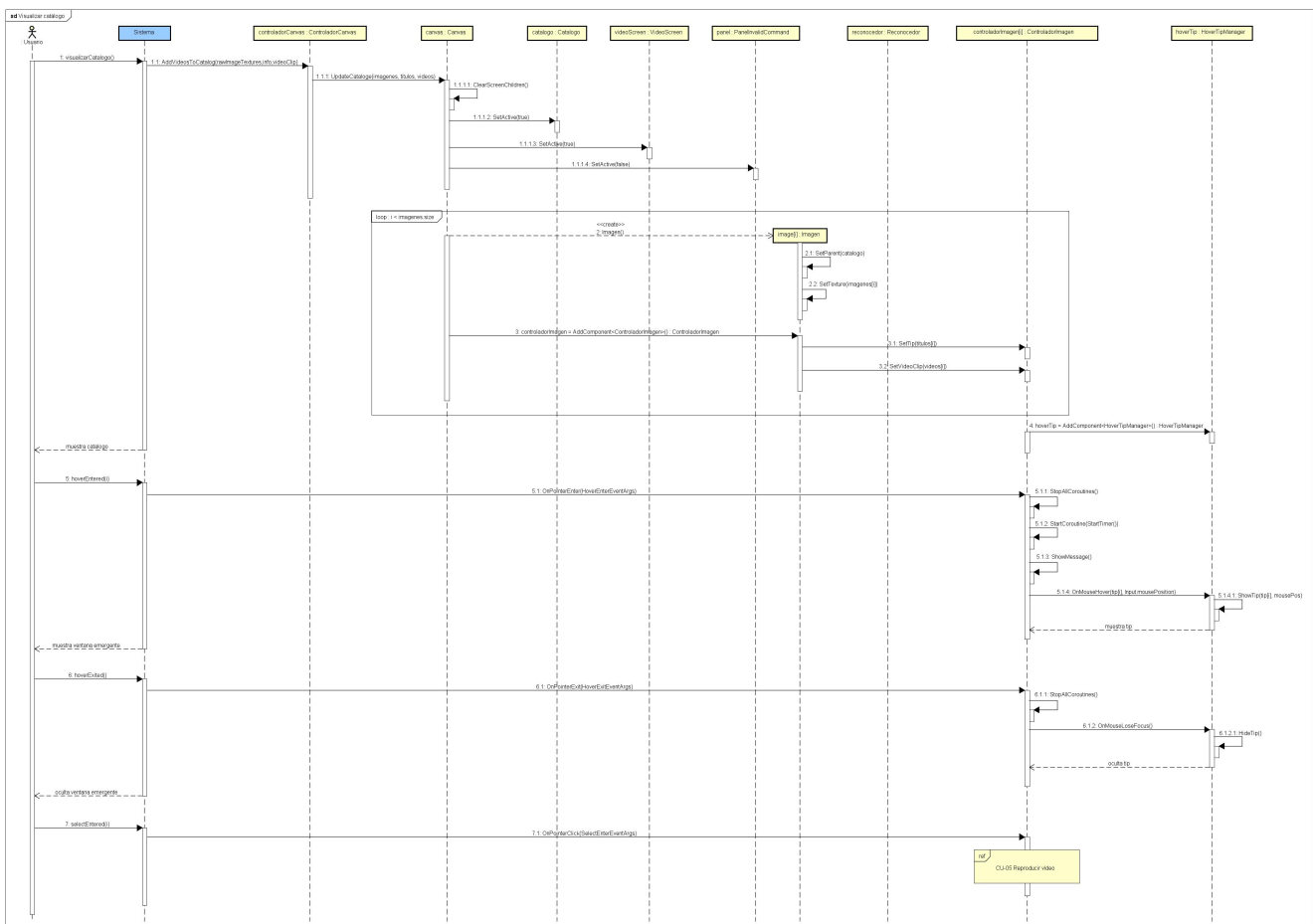


Figura 6.6: Realización en análisis del caso de uso Visualizar catálogo

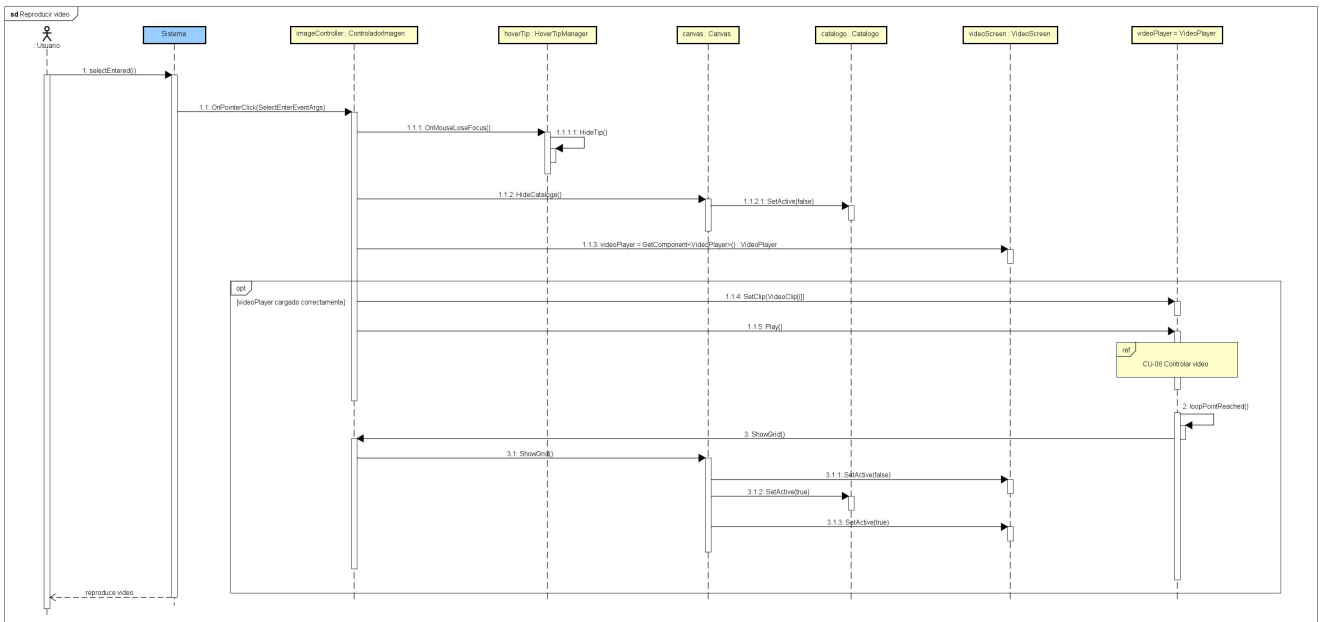


Figura 6.7: Realización en análisis del caso de uso Reproducir vídeo

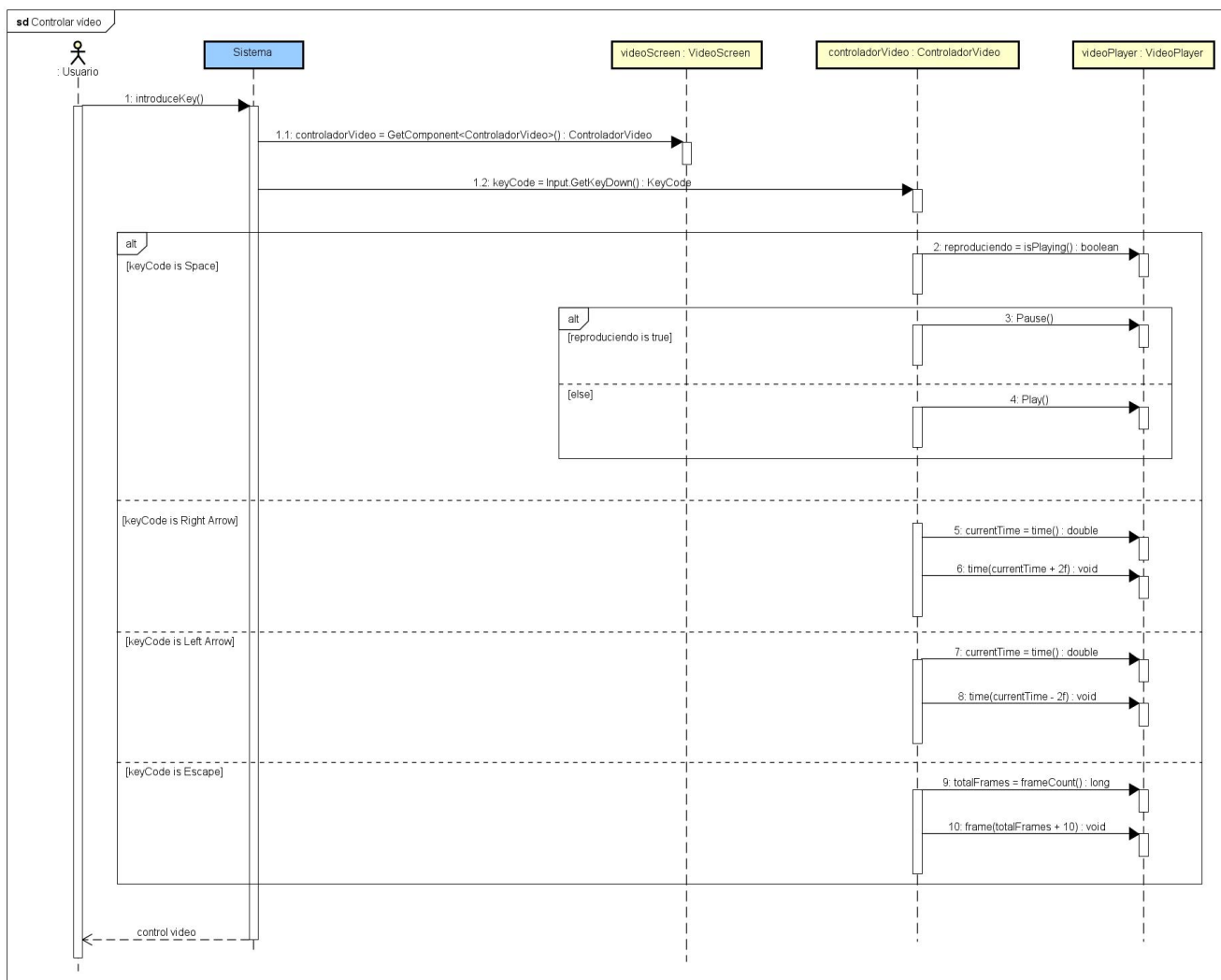


Figura 6.8: Realización en análisis del caso de uso Controlar vídeo

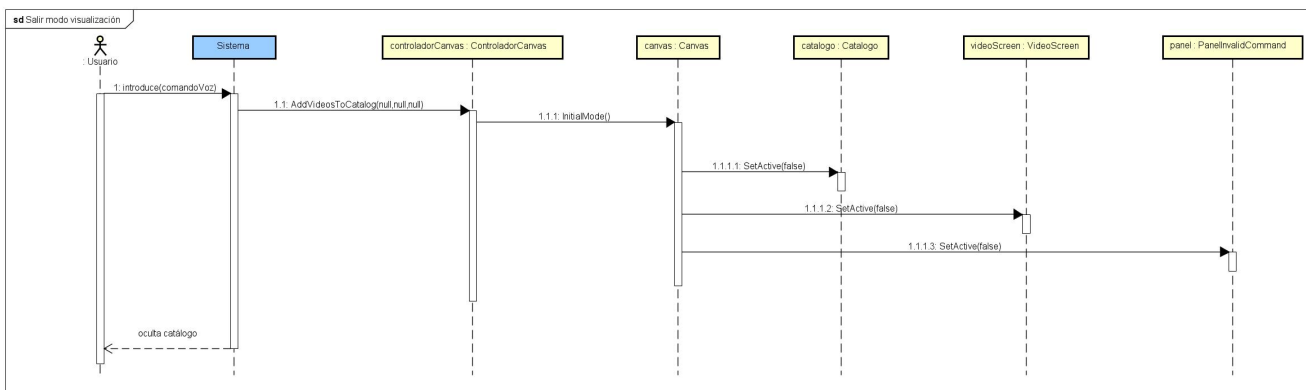


Figura 6.9: Realización en análisis del caso de uso Salir modo visualización

Capítulo 7

Diseño

En esta sección se presenta el desarrollo del diseño del proyecto Educaverse. Contiene las decisiones del entorno tecnológico, la arquitectura con los diagramas pertinentes, diseño de paquetes y diseño de la interfaz del proyecto.

7.1. Entorno tecnológico

El sistema se basará en una aplicación 3D compatible con la realidad virtual y que podrá exportarse el Metaverso, por lo cual el conjunto de tecnologías que se usarán en el desarrollo estarán orientadas a dicho propósito. A continuación, se procede a describir todas las herramientas utilizadas en el proyecto junto con las funcionalidades más destacadas e importantes de cada una de ellas.

- **Unity 2021.3.22f1**: Plataforma de desarrollo de juegos y aplicaciones 3D. Proporciona un entorno de desarrollo integrado (IDE) y herramientas visuales para la creación y edición de escenas, modelos, animaciones y efectos visuales [52].
- **C#**: La elección del lenguaje de programación C# se basa en su compatibilidad con Unity, su amplia comunidad y recursos disponibles, su orientación a objetos y facilidad de uso, así como su capacidad de integración con otras tecnologías y servicios.
- **Unity XR**: Framework de Unity que proporciona soporte para la realidad virtual y aumentada. Permite integrar y configurar dispositivos de RV, gestionar la entrada y la interacción en entornos virtuales [53].
- **OpenXR**: Implementación de software que permite crear aplicaciones de realidad aumentada y realidad virtual de forma más sencilla y compatible con una amplia gama de dispositivos. OpenXR es un estándar abierto y sin regalías desarrollado por Khronos Group, una organización que promueve estándares abiertos en la industria de los gráficos y medios interactivos [54].
- **Windows Mixed Reality**: Plataforma desarrollada por Microsoft que ofrece una experiencia de realidad mixta al combinar elementos de realidad virtual y realidad aumentada en un entorno interactivo. Unity ofrece herramientas y funcionalidades específicas diseñadas para aprovechar al máximo las capacidades de los cascos de Windows Mixed Reality. Estas capacidades incluyen características como el seguimiento de movimiento, que permite el rastreo preciso de los movimientos del usuario, el controlador de movimiento, que permite una interacción más precisa con los objetos virtuales, y la representación estereoscópica, que brinda una experiencia visual envolvente y tridimensional [55].
- **UnityEngine.Windows.Speech**: La API UnityEngine.Windows.Speech forma parte del Scripting API de Unity y está diseñada para proporcionar funcionalidad de reconocimiento de voz en aplicaciones de Unity en entornos de Windows. Esta API permite la integración de capacidades de reconocimiento de voz en tiempo real en la aplicación, lo que permite a los usuarios interactuar con el sistema utilizando comandos de voz [56].
- **Oculus Integration SDK**: El Oculus Integration SDK para Unity [57] es un conjunto de herramientas y recursos que proporciona soporte para el desarrollo de aplicaciones de Oculus en Unity. Esta integración permite crear experiencias inmersivas para los dispositivos de Oculus, como el Oculus Quest 2.

- Plataforma de desarrollo:** Se ha utilizado Visual Studio Code como IDE debido a que es altamente configurable y ofrece soporte para el desarrollo en C#. Es ampliamente utilizado en la comunidad de desarrolladores de Unity debido a su facilidad de uso, rendimiento y capacidad de extensión. En cuanto a la herramienta escogida como control de versiones se ha utilizado Gitlab debido a su facilidad de uso y robustez, lo que ayuda a mantener un control riguroso sobre el desarrollo del proyecto.

7.2. Arquitectura del sistema

7.2.1. Estilo arquitectónico

Para el sistema desarrollado, se ha elegido implementar una arquitectura de capas [58] debido a sus ventajas en términos de modularidad, escalabilidad y mantenibilidad. Este estilo arquitectónico permite organizar el sistema en capas lógicas que se comunican entre sí de forma jerárquica, brindando una estructura clara y separación de responsabilidades, lo que permite la incorporación de nuevas funcionalidades o adaptación a futuros requisitos. Por estos motivos, la arquitectura de capas se adapta adecuadamente al proyecto debido a la naturaleza modular y escalable de la aplicación. A continuación, se describen las capas que componen la arquitectura y sus responsabilidades:

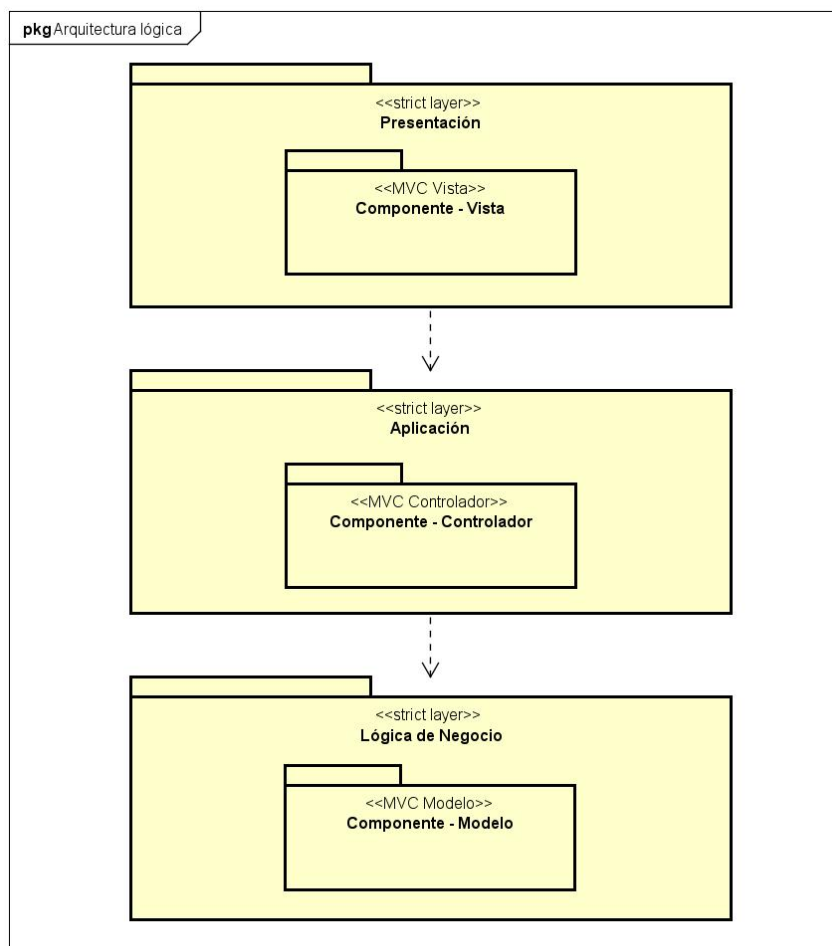


Figura 7.1: Capas de la aplicación

- Capa de Presentación «strict layer»:** Esta capa es responsable de la interacción directa del usuario y la presentación de la información. Dentro de esta capa se encuentran los componentes que conforman la vista de la aplicación y que se encargan de mostrar la interfaz gráfica.
- Capa de Aplicación «strict layer»:** Esta capa actúa como un intermediario entre la capa de Presentación y la capa de Lógica de Negocio, lo cual ayuda a separar la lógica de presentación de la lógica de negocio.

Aquí se encuentran los controladores, que se encargan de procesar las interacciones del usuario, coordinar las acciones y comunicarse con la capa de Lógica de Negocio según sea necesario.

- **Capa de Lógica de Negocio «strict layer»:** La capa de Lógica de Negocio, o capa de Dominio, contiene la lógica central de la aplicación y engloba tanto la lógica de negocio como la lógica de persistencia de los datos. Contiene los componentes encargados de implementar las reglas y operaciones específicas del dominio de la aplicación, así como de realizar cálculos, tomar decisiones de acuerdo a los datos de entrada y procesar los datos internos de la aplicación. Debido a que los datos de la aplicación se encuentran integrados dentro de la propia aplicación, no se ha utilizado una base de datos o ficheros externos para guardar la información. Por este motivo, la capa de Lógica de Negocio integra tanto la manipulación como la gestión de los datos internos de la aplicación, proporcionando una solución autónoma y autocontenida en el entorno de Unity.

7.3. Patrones de diseño

En el proceso de diseño de software, una de las tareas más importantes es la elección adecuada de los patrones de diseño. Los patrones de diseño son soluciones probadas para problemas recurrentes en el desarrollo de software. Proporcionan un enfoque estructurado y estandarizado para abordar desafíos específicos y promueven la reutilización de soluciones exitosas en diferentes contextos [59].

En el caso de nuestra aplicación, es fundamental seleccionar los patrones de diseño adecuados que nos ayudarán a lograr un diseño robusto, flexible y fácilmente mantenible. La elección correcta de los patrones de diseño nos permitirá resolver problemas de manera eficiente, mejorar la calidad del código y promover la escalabilidad y la extensibilidad del sistema.

A continuación, identificaremos y describiremos los patrones de diseño que consideramos necesarios y relevantes para nuestra aplicación:

7.3.1. Patrón MVC (Modelo-Vista-Controlador)

El patrón de arquitectura Modelo-Vista-Controlador es un enfoque para diseñar aplicaciones que separa la lógica de negocio de la interfaz de usuario y la gestión de datos. Esta separación permite una estructura más modular y facilita el mantenimiento y la escalabilidad del código [60].

El patrón MVC se compone de tres componentes principales:

- **Modelo:** Representa la lógica de negocio y los datos de la aplicación. Se encarga de la gestión y manipulación de los datos, así como de la implementación de las reglas de negocio. El modelo puede interactuar con una base de datos, servicios externos u otros recursos para obtener o persistir la información.
- **Vista:** Es la interfaz de usuario visible para el usuario final. Su función es presentar la información al usuario de una manera comprensible y visualmente atractiva. La vista se actualiza cuando hay cambios en el modelo y muestra los datos actualizados al usuario.
- **Controlador:** Actúa como intermediario entre el modelo y la vista. Recibe las interacciones del usuario desde la vista y procesa esas acciones. Luego, realiza las actualizaciones necesarias en el modelo y, si es necesario, notifica a la vista para que se actualice y refleje los cambios realizados. El controlador también puede manejar eventos externos que afecten al modelo.

La comunicación entre los componentes se realiza a través de mecanismos de observación o eventos. Cuando se produce un cambio en el modelo, el controlador actualiza la vista correspondiente para reflejar esos cambios. Por otro lado, cuando el usuario interactúa con la vista, el controlador procesa esas acciones y actualiza el modelo según sea necesario [60].

La ventaja del patrón MVC radica en la separación clara de responsabilidades entre los componentes. Esto facilita el desarrollo paralelo de la lógica de negocio y la interfaz de usuario, lo que resulta en un código más limpio y mantenible. Además, el patrón MVC permite reutilizar y extender los componentes de manera independiente, lo que mejora la escalabilidad del sistema [60].

7.3.2. Patrón Observer

El patrón Observador [61] es un patrón de diseño de comportamiento que permite establecer una relación de uno a muchos entre un objeto llamado sujeto y varios objetos llamados observadores, con el fin de notificar automáticamente a todos ellos cuando ocurre un cambio en el estado del objeto que está observando.

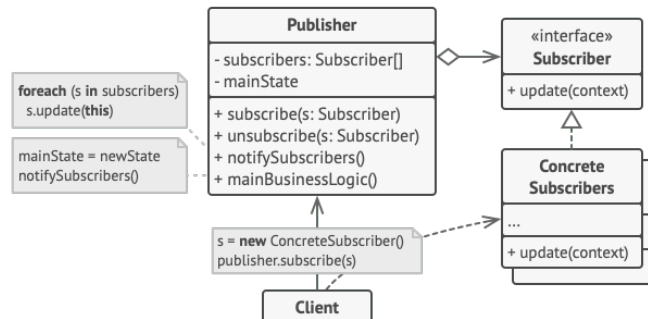


Figura 7.2: Estructura del patrón Observador. Fuente [61]

En relación con nuestra aplicación, este patrón se ha utilizado en la interacción entre la clase Reconocedor y los componentes que dependen de él. El reconocedor actúa como el sujeto observado, y los componentes de la interfaz de usuario que muestran las imágenes y videos son los observadores que reciben notificaciones cuando se detectan comandos válidos.

7.3.3. Patrón State

El patrón Estado [62], también conocido como State, se utiliza cuando un objeto debe cambiar su comportamiento en función de su estado interno. Permite que un objeto altere su comportamiento y sus acciones internas a medida que cambia su estado, sin que sea necesario cambiar la clase del objeto en sí. De esta manera, el patrón Estado promueve la encapsulación, la modularidad y la extensibilidad del código.

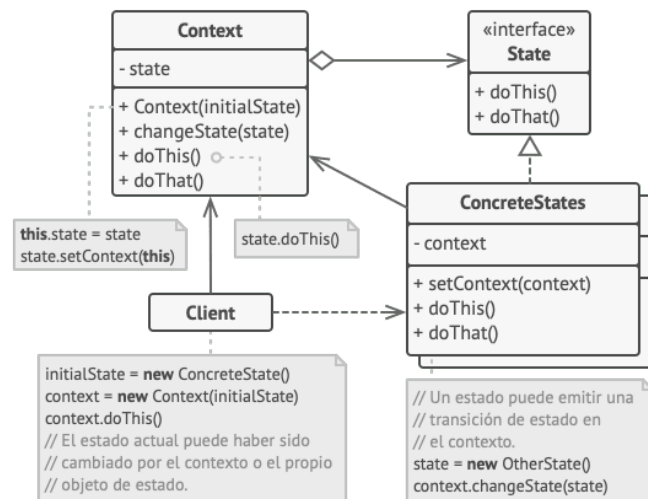


Figura 7.3: Estructura del patrón State. Fuente [62]

En nuestra aplicación, se ha utilizado el patrón Estado para gestionar la visibilidad de los componentes del Canvas, permitiendo la visualización de imágenes, vídeos y el cierre de la interfaz. Además, se ha aplicado este patrón al control de reproducción de los vídeos. El objeto VideoPlayer posee distintos estados que determinan la reproducción del vídeo, como reproducción, pausa o detención. Mediante la aplicación del patrón Estado, el VideoPlayer puede adaptar su comportamiento y acciones internas de acuerdo con su estado actual, sin necesidad de realizar modificaciones en la clase en sí. Esto proporciona una gestión eficiente y flexible de la reproducción de

vídeos, permitiendo una adaptación dinámica a los diferentes estados y ofreciendo al usuario una experiencia fluida y coherente.

7.3.4. Patrón Singleton

El patrón Singleton es un patrón de diseño creacional que se utiliza para garantizar que solo haya una única instancia de una clase en todo el sistema. La idea principal detrás del patrón Singleton es proporcionar un punto de acceso global a esta instancia única, lo que permite compartir su estado y comportamiento en diferentes partes de la aplicación [63].

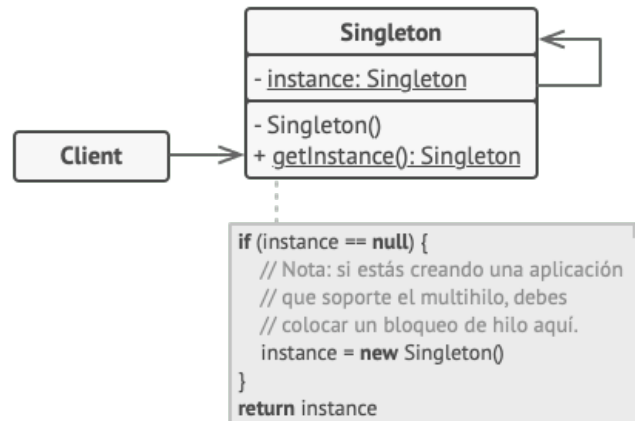


Figura 7.4: Estructura del patrón Singleton. Fuente [63]

En relación a nuestra aplicación, se ha utilizado el patrón Singleton en la clase DictationRecognizer. Al utilizar dicho patrón, garantizamos que solo exista una única instancia del reconocedor de voz en toda la aplicación. Esta instancia se crea al comienzo de la aplicación y luego puede ser activada o desactivada según los requerimientos del cliente. De esta manera, aseguramos que el reconocedor de voz mantenga su estado y configuración durante la ejecución de la aplicación.

7.4. Diagrama de paquetes

En el siguiente apartado, se muestra en detalle el diseño de los paquetes que conforman la aplicación final.

En la Figura 7.5, se muestra la estructura de Unity por paquetes resultantes del proyecto. Se resalta el paquete Assets, debido a que en él radica la aplicación.

En la Figura 7.6, se observa el diseño en profundidad del paquete Scripts, siguiendo el patrón MVC.

En la Figura 7.7, se muestra el paquete models, paquete que contiene los datos relacionados con la aplicación y el reconocedor.

En la Figura 7.8, se muestra el paquete view, paquete que contiene los elementos que componen la interfaz de la aplicación.

En la Figura 7.9, se muestra el paquete Doctor'sOfficeView, donde se encuentran todas las texturas, modelos y prefabs utilizados para contruir la sala médica.

En la Figura 7.10, se muestra el paquete DoctorView, donde se encuentran las animaciones, texturas y materiales relacionados con el avatar médico.

Por último, en la Figura 7.11, se muestra el paquete controllers, formado por los distintos controladores de la aplicación encargados de gestionar la interacción entre el modelo y la vista.

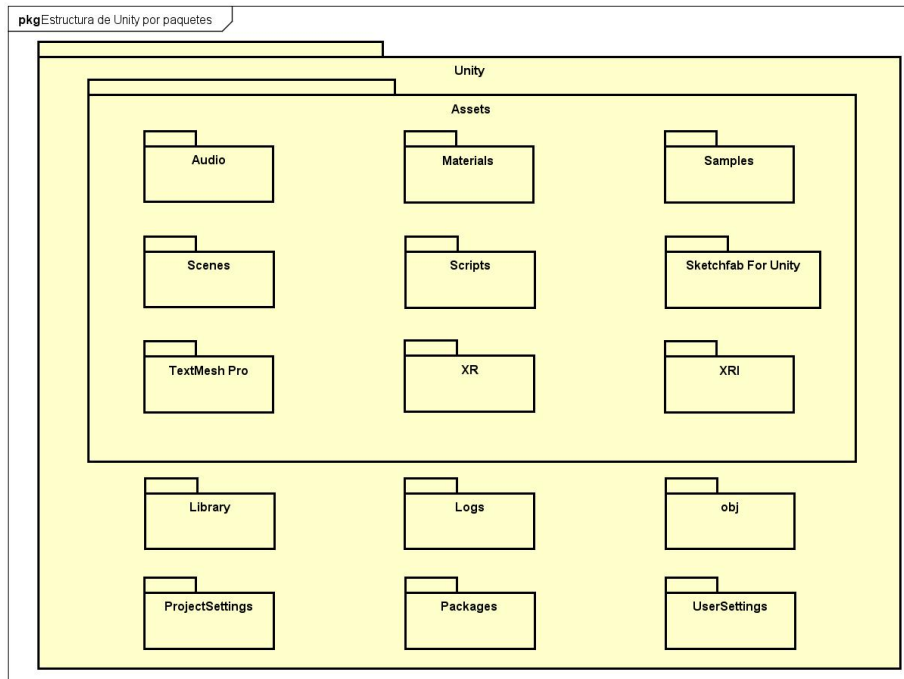


Figura 7.5: Estructura de Unity por paquetes

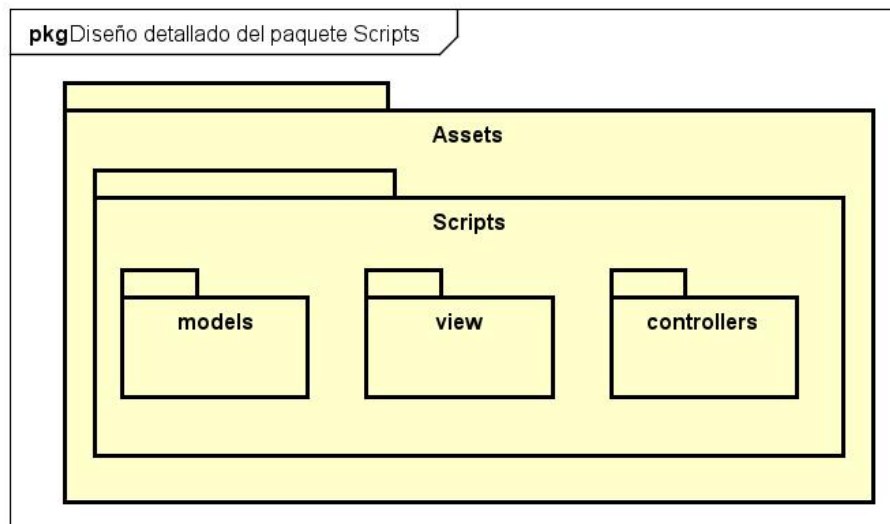


Figura 7.6: Diseño detallado del paquete Scripts

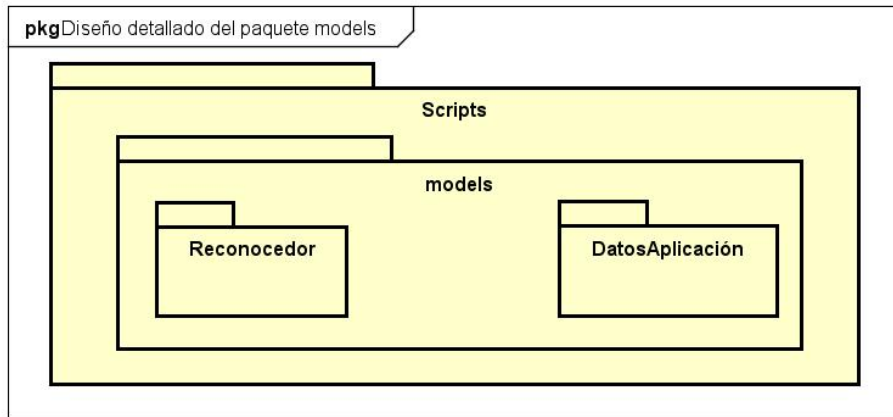


Figura 7.7: Diseño detallado del paquete models

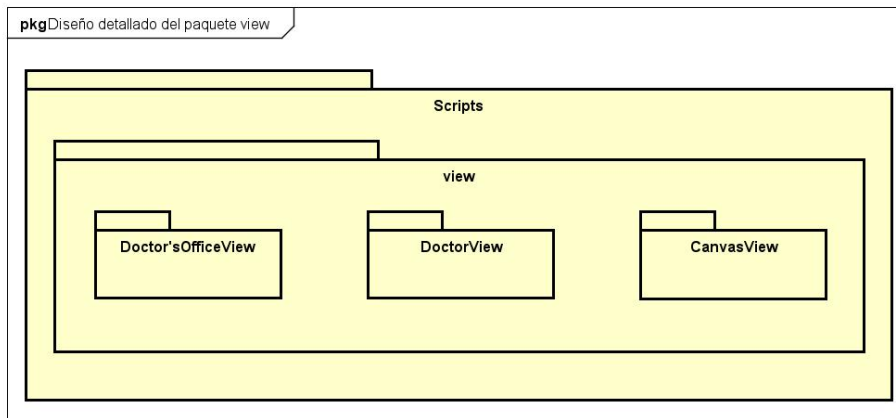


Figura 7.8: Diseño detallado del paquete view

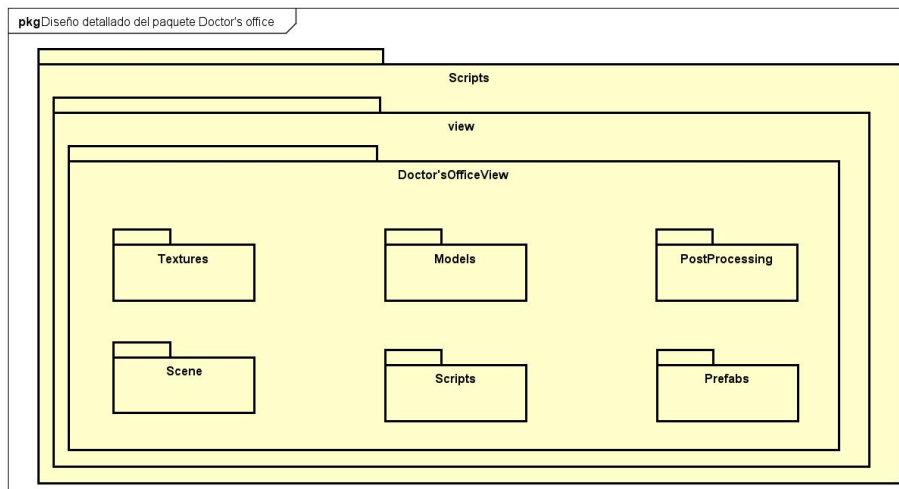


Figura 7.9: Diseño detallado del paquete Doctor'sOfficeView

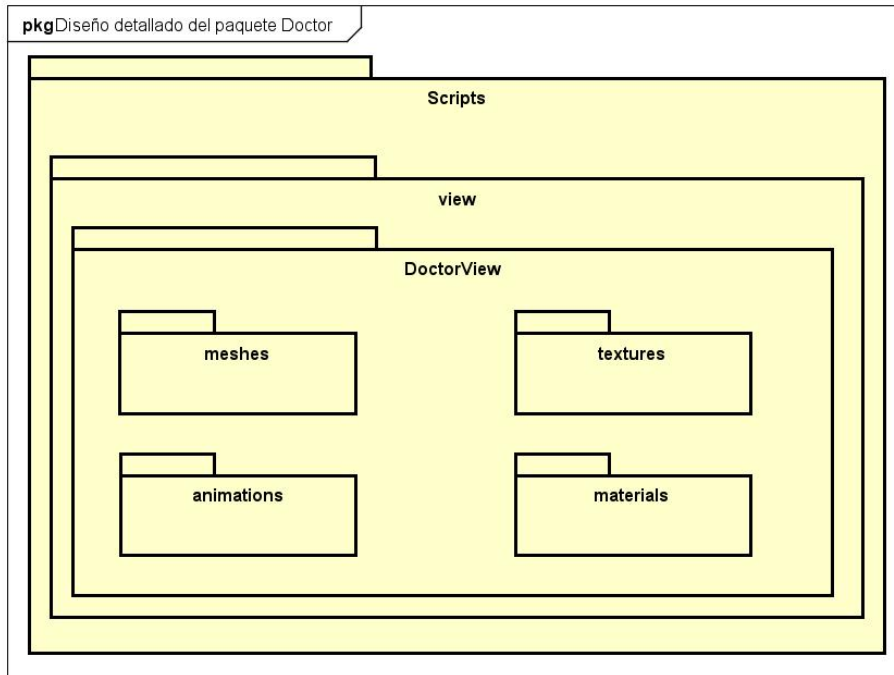


Figura 7.10: Diseño detallado del paquete DoctorView

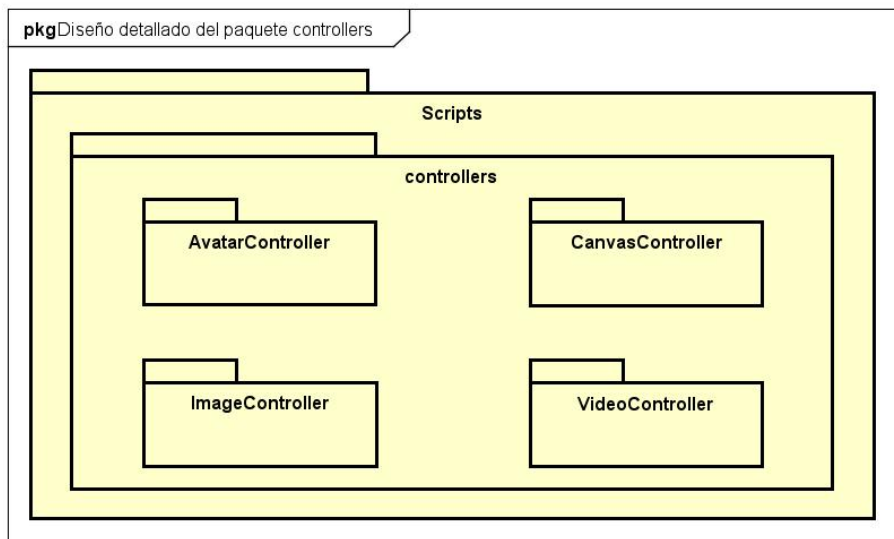


Figura 7.11: Diseño detallado del paquete controllers

7.5. Diagrama de componentes

El diagrama de componentes es una técnica de modelado utilizada en ingeniería de software para representar la estructura y organización de un sistema en términos de componentes y las relaciones entre ellos. Un componente puede ser considerado como una unidad modular y autocontenida de funcionalidad dentro del sistema. Puede ser un módulo, una biblioteca, una clase, un paquete o incluso un subsistema completo [64].

El objetivo principal del diagrama de componentes es proporcionar una visión general de alto nivel de la arquitectura del sistema, mostrando cómo los diferentes componentes se relacionan entre sí y cómo se comunican para lograr una funcionalidad específica. Además de representar los componentes y sus relaciones, el diagrama de componentes también puede mostrar las interfaces proporcionadas y utilizadas por cada componente, lo que ayuda a entender cómo se comunican y colaboran entre sí. A continuación se muestra el diagrama de componentes del sistema:

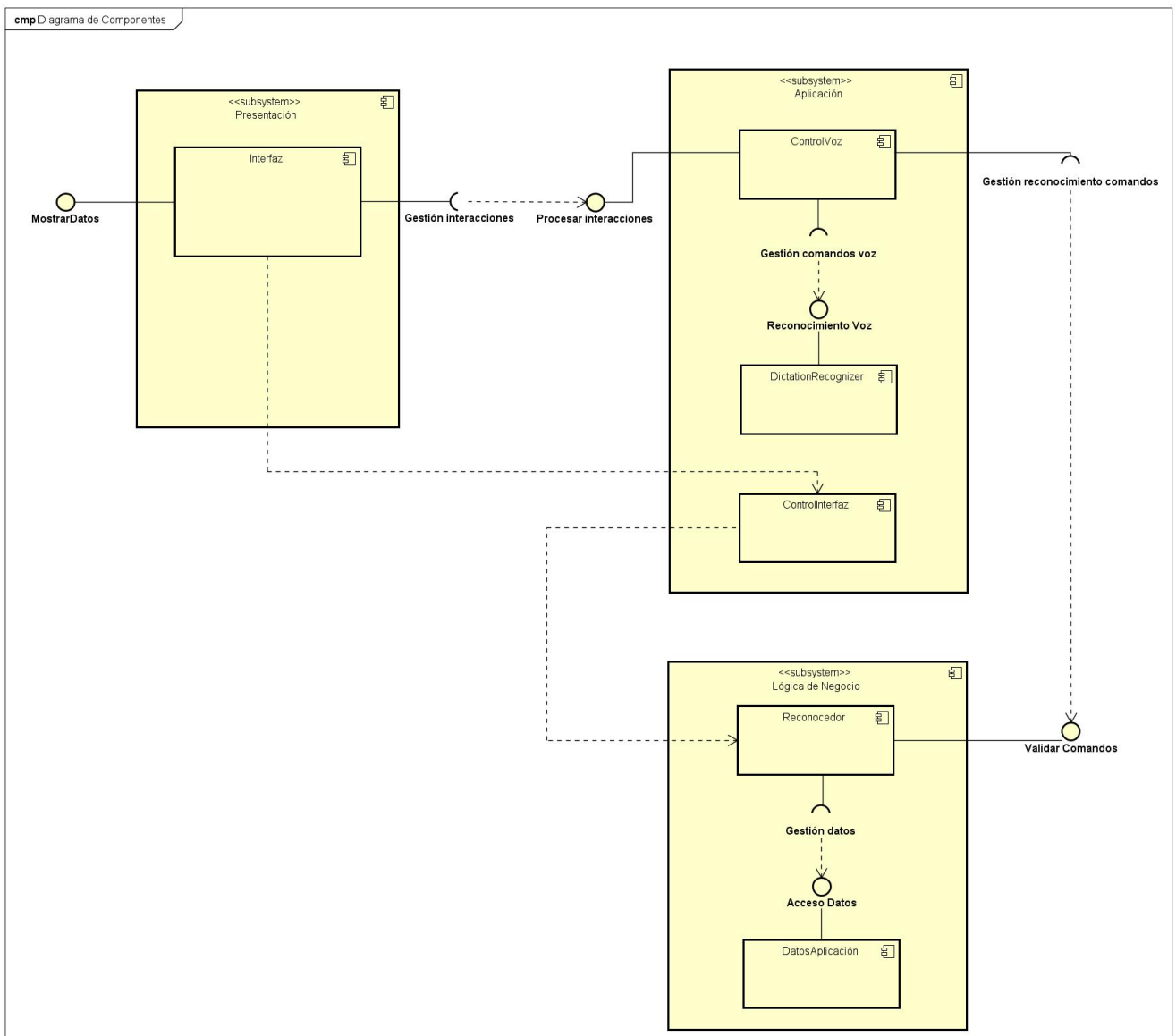


Figura 7.12: Diagrama de componentes

El enfoque de subsistemas se ha adoptado para brindar una visión de alto nivel del sistema, permitiendo una mejor comprensión de las responsabilidades y relaciones entre los componentes. Esta distribución facilita la modularidad, el mantenimiento y la escalabilidad del sistema, al establecer áreas de responsabilidad claramente

definidas y permitir el desarrollo independiente de cada subsistema.

En cuanto a las interfaces expuestas o necesarias por cada componente, se han definido las siguientes:

- **Componente: Interfaz**
 - Interfaz provista: *MostrarDatos*. Este componente implementa la interfaz que permite mostrar la información a los usuarios.
 - Interfaz necesaria: *Gestión interacciones*. Sin embargo, este componente requiere utilizar la interfaz *Procesar Interacciones* del componente *ControlVoz* para procesar las interacciones realizadas por el usuario. Depende de los métodos y funcionalidades proporcionados por la interfaz para realizar la gestión de las interacciones recibidas.
- **Componente: ControlVoz**
 - Interfaz provista: *Procesar Interacciones*. Este componente implementa una interfaz que se encarga de procesar las interacciones realizadas por el usuario en la aplicación. Proporciona métodos y propiedades para procesar las interacciones de la interfaz de usuario, activando el reconocimiento de voz y gestionando los comandos recibidos.
 - Interfaz necesaria: *Gestión Comandos Voz* y *Gestión Reconocimiento Comandos*. Por otra parte, este componente debe realizar una gestión de los comandos de voz a través de la interfaz provista por el componente *DictationRecognizer*. Posteriormente, debe realizar una gestión relacionada con el reconocimiento de comandos, utilizando en este caso la interfaz provista por el componente *Reconocedor*.
- **Componente: DictationRecognizer**
 - Interfaz provista: *Reconocimiento Voz*. Este componente implementa una interfaz que ofrece el servicio de reconocimiento de voz a la aplicación.
- **Componente: Reconocedor**
 - Interfaz provista: *Validar Comandos*. Este componente se encarga de validar los comandos introducidos y desencadena las acciones correspondientes para mostrar los vídeos a los usuarios a través de la interfaz implementada.
 - Interfaz necesaria: *Gestión Eventos*. Para poder llevar a cabo la validación de los comandos, este componente necesita gestionar los datos propios de la aplicación haciendo uso de la interfaz proporcionada por el componente *DatosAplicación*.
- **Componente: DatosAplicación**
 - Interfaz provista: *Acceso Datos*. Finalmente, este componente implementa una interfaz que proporciona acceso a los datos de la aplicación, proporcionando métodos y funcionalidades para obtener los vídeos e imágenes correspondientes en función de las validaciones realizadas.

7.6. Diagrama de despliegue

Un diagrama de despliegue UML es un tipo de diagrama de estructura utilizado en el modelado de sistemas orientados a objetos para representar los aspectos físicos de un sistema, concretamente la configuración de los nodos de procesamiento en tiempo de ejecución y los componentes que residen en ellos. Estos diagramas proporcionan una visión estática del despliegue o topología del hardware de un sistema [65]. En la Figura 7.13 se presenta el diagrama de despliegue del sistema:

El diagrama cuenta con los siguientes elementos:

- **VR device**: Representa el dispositivo de realidad virtual utilizado por el usuario para ejecutar la aplicación. Dicha aplicación se ejecuta desde el entorno de ejecución de Unity y depende de los siguientes componentes:
 - **Unity XR SDK (Artifact)**: biblioteca o conjunto de herramientas que permiten la integración de la realidad virtual en la aplicación. Este componente permite la compatibilidad con diferentes plataformas de realidad virtual como Windows Mixed Reality o Meta Quest.

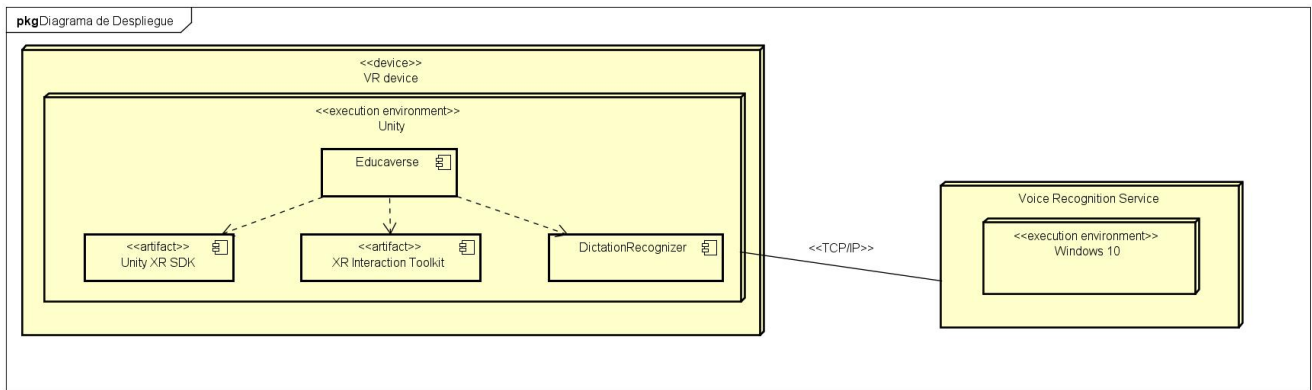


Figura 7.13: Diagrama de despliegue

- **XR Interaction Toolkit (Artifact):** conjunto de herramientas proporcionado por Unity que simplifica la interacción del usuario en la realidad virtual. Este componente ofrece funcionalidades predefinidas para la interacción con objetos virtuales, como la selección, el movimiento y la manipulación de objetos mediante los controladores del dispositivo de realidad virtual.
- **DictationRecognizer:** componente encargado de convertir los comandos de voz introducidos por el usuario en texto. Para ello, utiliza el servicio de reconocimiento de voz de Windows. Cuando el usuario emite un comando de voz en la aplicación, el DictationRecognizer captura el audio, lo envía al servicio de reconocimiento de voz y recibe como respuesta el texto correspondiente al comando pronunciado.
- **Voice Recognition Service:** Representa el servicio que proporciona la funcionalidad de reconocimiento de voz en la aplicación de realidad virtual. Dicho servicio se ejecuta dentro del sistema operativo Windows 10. Para establecer la comunicación entre los distintos nodos, se utiliza una conexión TCP/IP. Una vez que se ha realizado el reconocimiento de voz y se ha convertido el audio en texto, el servicio envía esta información a la aplicación de realidad virtual para su posterior procesamiento.

7.7. Diseño de la Interfaz de Usuario

Uno de los requerimientos principales de la aplicación consistía en lograr una interacción realista entre los usuarios y el avatar médico dentro de una simulación de una consulta médica. Para cumplir con este objetivo, se trabajó en estrecha colaboración con el tutor de HP SCDS y se exploraron diferentes recursos ya existentes. Después de un exhaustivo proceso, se logró obtener los recursos necesarios (Figura 7.14 y Figura 7.15) que permitieron la implementación de una interfaz en forma de sala médica 3D, brindando a los usuarios la oportunidad de interactuar con el avatar médico y explorar diversos elementos de la sala.

El diseño de la interfaz se enfocó en ofrecer una experiencia intuitiva y atractiva, creando una simulación que resultara lo más cercana posible a una consulta médica real. Esto no solo buscaba proporcionar un entorno visualmente agradable, sino también mejorar la interacción y la experiencia de los usuarios durante el uso de la aplicación.

En este contexto, la interfaz de usuario se refiere a la capa visual y tangible a través de la cual los usuarios interactúan con la aplicación [66]. Comprende elementos visuales, como botones, menús, campos de entrada y paneles, así como interacciones táctiles, movimientos del ratón o comandos de voz. El diseño de la interfaz de usuario tuvo como objetivo principal facilitar la navegación y el uso de las funcionalidades del sistema, permitiendo a los usuarios acceder de manera intuitiva a los diferentes recursos y opciones disponibles.

7.7.1. Principios de diseño de interfaz de usuario

Al diseñar la interfaz de usuario, es importante seguir algunos principios fundamentales [66] entre los cuales se destacan:

- **Simplicidad:** La interfaz debe ser simple y fácil de entender. Para ello se utilizó una disposición clara y organizada de los elementos visuales, así como una jerarquía visual efectiva para destacar los elementos más importantes, dando prioridad a la funcionalidad principal de la aplicación, que en este caso es la interacción con el avatar médico y la visualización de los elementos de la sala médica.
- **Consistencia:** Se mantuvo una estructura coherente y consistente en todo el diseño de la interfaz. Se utilizaron elementos visuales y de interacción de manera uniforme, como colores, tipografía y estilos, para que los usuarios puedan reconocer patrones y familiarizarse con la aplicación.
- **Retroalimentación:** Proporcionar una respuesta visual o auditiva inmediata cuando los usuarios realicen acciones o interactúen con elementos de la interfaz. En este proyecto las respuestas serán visuales, mostrando un panel con la información correspondiente. Además, cuando los usuarios interactúen con el avatar médico, se mostrará una marca sobre dicho avatar para ayudar a los usuarios a comprender que sus acciones se han registrado y están siendo procesadas por el sistema.
- **Usabilidad:** Diseño de la interfaz pensando en la facilidad de uso y en las necesidades del usuario. Se realizaron pruebas de usabilidad para identificar posibles obstáculos o dificultades que puedan surgir al interactuar con la aplicación.

7.7.2. Sala médica

La sala está compuesta por elementos típicos presentes en una consulta médica, como una camilla, una mesa de trabajo, varias sillas, estanterías y armarios con suministros médicos y equipos especializados. Estos elementos se representan con precisión y realismo, creando un ambiente familiar y auténtico. La sala médica 3D proporciona un contexto visualmente atractivo y coherente para que el usuario se sienta cómodo y comprometido durante su interacción con la aplicación.



Figura 7.14: Consulta médica. Fuente [67]

7.7.3. Avatar médico

El avatar médico es una representación virtual de un profesional médico que interactúa con el usuario dentro de la sala médica. El avatar está diseñado para ser realista y expresivo, utilizando tecnología de animación y modelado

3D avanzada. Su apariencia y comportamiento se asemejan a un médico humano, lo que genera confianza y empatía en el usuario. El avatar médico cumple varias funciones, como proporcionar información relevante, respondiendo a los comandos del usuario de manera natural, o guiar al usuario a través de un audio de bienvenida al inicio de la aplicación, proporcionando instrucciones sobre cómo interactuar con la interfaz y acceder al catálogo de vídeos. Su presencia ayuda a crear una experiencia más personalizada y atractiva para el usuario.



Figura 7.15: Avatar médico. Fuente [68]

Además, cuando el usuario interactúa con el avatar médico, se utiliza una marca visual para indicar que el avatar está activo y listo para recibir comandos de voz. Esta marca actúa como una retroalimentación visual para el usuario, permitiéndole saber cuándo es el momento adecuado para comenzar a interactuar con el avatar utilizando comandos de voz.



Figura 7.16: Avatar médico activado

7.7.4. Canvas

El Canvas es el componente de la interfaz de usuario que alberga y presenta los vídeos disponibles para su visualización. Este espacio se ha diseñado de manera organizada y estructurada para ofrecer una experiencia de navegación sencilla y accesible. La disposición de los vídeos se realiza en forma de una cuadrícula, donde el tamaño de las imágenes y el espaciado entre ellas se ajusta de forma dinámica en función de la cantidad de vídeos disponibles.

Cada vídeo se representa mediante una miniatura o imagen representativa, acompañada de su título correspondiente. Al situar el cursor sobre una de estas imágenes, se muestra el título del vídeo, proporcionando al usuario una información adicional antes de realizar la selección (Figura 7.17). Al elegir un vídeo específico, el usuario puede acceder a su reproducción y visualización a través del reproductor de vídeo incorporado (Figura 7.18).



Figura 7.17: Catálogo de vídeos sobre el “Glucómetro”



Figura 7.18: Reproducción del vídeo “Glucemia capilar”

El Canvas desempeña un papel crucial al permitir al usuario explorar y seleccionar los contenidos que desea ver. Su diseño intuitivo y personalizable brinda una experiencia interactiva, donde los usuarios pueden navegar cómodamente por los vídeos disponibles y elegir aquellos que les resulten más interesantes o relevantes. Esta sección de la interfaz de usuario facilita una experiencia fluida y atractiva, fomentando la interacción y el disfrute de los contenidos multimedia dentro de la aplicación.

Capítulo 8

Implementación

En este capítulo se describirá en detalle el desarrollo del proyecto, explicando cómo se han llevado a cabo las iteraciones, qué funcionalidades se han desarrollado en cada una y cómo se han abordado los cambios y desafíos.

8.1. Iteración 1

Una vez familiarizado con las herramientas de desarrollo, se creó una versión inicial del proyecto. La tarea principal de esta iteración fue, además de configurar el entorno de trabajo adecuadamente, identificar los requerimientos del proyecto, los cuales podrían ir cambiando para incluir o excluir funcionalidad.

Como se comentó en el apartado 4.5, la herramienta de gestión del proyecto fue GitLab, en donde se crearon todos los issues correspondientes y se fueron añadiendo al tablón de issues. Dicho tablón al final de la primera iteración quedó como se puede ver en la siguiente figura:

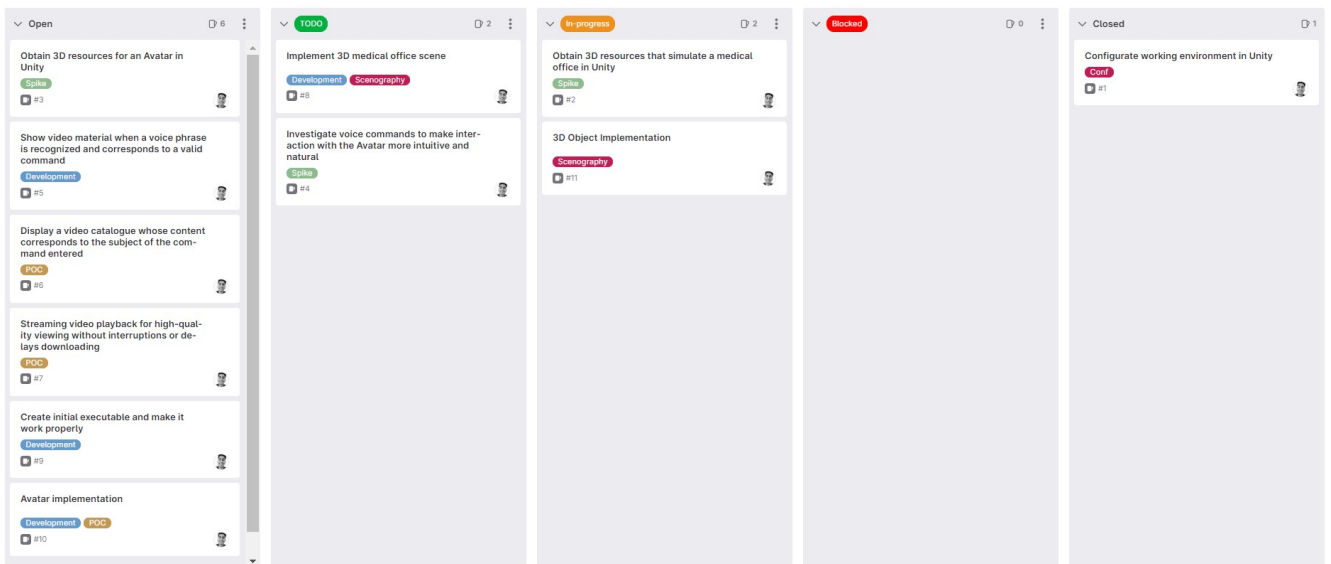


Figura 8.1: Fin de la primera iteración

8.1.1. Issues

Uno de los aspectos clave en la implementación del proyecto ha sido la gestión efectiva de las tareas y requerimientos a través de los issues en GitLab [69]. Los issues se han convertido en unidades de trabajo indivisibles que

representan funcionalidades, mejoras o correcciones de errores específicos que necesitaban ser abordados. Cada issue ha sido creado con un propósito claro y se ha estructurado siguiendo las mejores prácticas ágiles y el enfoque de User Stories [70], lo que ha permitido una comprensión más efectiva de los objetivos y beneficios que se esperaban lograr.

A continuación, se explica la estructura que se ha seguido para la creación de los issues:

- **Título:** Resumen conciso y descriptivo del issue que refleja el objetivo o la tarea que se va a abordar.
- **Descripción (User Story):** Se describe la funcionalidad o requisito desde la perspectiva del usuario o cliente. Sigue la estructura “Como [rol], quiero [funcionalidad/descripción] para [beneficio]”. Esto ayuda a proporcionar claridad sobre el propósito y el valor que se espera obtener al completar el issue.
- **Acceptance Criteria:** Son declaraciones claras y específicas que definen los requisitos y condiciones que deben cumplirse para que el issue se considere completado correctamente. Estos criterios deben ser verificables y medibles.
- **Label:** Son etiquetas o categorías asignadas al issue para clasificarlo y organizarlo mejor dentro del proyecto. Las labels que se utilizaron fueron las siguientes:
 - **Avatar:** Issues relacionadas con la configuración del avatar y la implementación de comandos de voz.
 - **Blocked:** Issues bloqueadas porque hay dependencias con terceros.
 - **Conf:** Issues relacionadas con la configuración del entorno de trabajo.
 - **Development:** Issues relacionadas con el proceso de desarrollo de software.
 - **Meta:** Issues relacionadas con la configuración del entorno en el Metaverso.
 - **POC:** Issues utilizadas para demostrar que una idea puede funcionar y para validar su viabilidad antes de invertir tiempo y recursos en su pleno desarrollo [71].
 - **Scenography:** Issues creadas para llevar a cabo el diseño de la sala médica e implementar su funcionalidad.
 - **Spike:** Issues creadas con el fin de investigar y explorar un problema técnico específico antes de abordarlo formalmente [72].
 - **Test:** Issues relativas a las pruebas de cualquier tipo.

8.1.2. Tareas completadas

A continuación se muestra la tarea que se llevó a cabo durante la primera iteración:

Título	Configurar entorno de desarrollo en Unity
User Story	Como desarrollador, Quiero configurar adecuadamente el entorno de desarrollo Para poder comenzar a implementar funcionalidad al proyecto.
Acceptance Criteria	<ul style="list-style-type: none"> ▪ Se debe descargar la aplicación Unity Hub, la cual agiliza la forma de encontrar, descargar y gestionar proyectos e instalaciones de Unity. ▪ Se debe realizar la instalación del Unity Editor. ▪ Se debe instalar el IDE : Visual Studio Code. ▪ Se deben instalar las siguientes extensiones: Unity Code Snippets y C#. ▪ Se crea el proyecto 3D donde se irán añadiendo todos los recursos.
Label	Conf

Tabla 8.1: Issue 1

8.1.3. Funcionalidad realizada

Configurar entorno de desarrollo

Para llevar a cabo la configuración del entorno de desarrollo en Unity, comenzamos descargando Unity Hub [73]. En mi caso, he descargado e instalado la versión Unity Hub 3.4.1. Continuamos descargando el Editor de Unity desde la ventana *Installs* de Unity Hub:

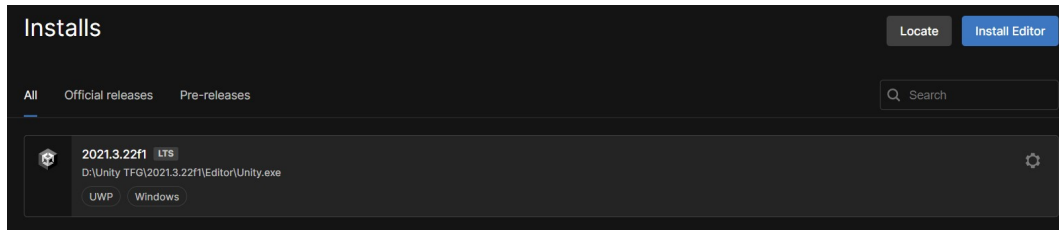


Figura 8.2: Ventana Installs de Unity Hub

Como se puede observar en la Figura 8.2, el Editor de Unity con el que trabajaré será el 2021.3.22f1. Una vez instalado el Editor de Unity, podemos crear nuestro proyecto 3D desde la ventana *Projects* de Unity Hub, clicando en el botón de *New Project*:

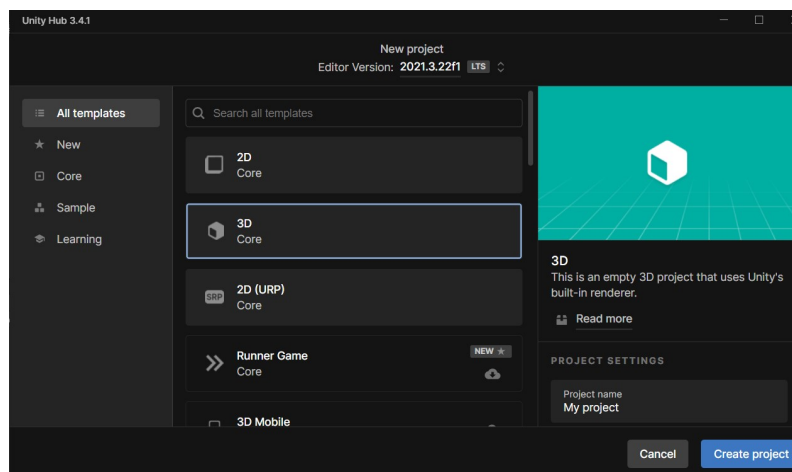


Figura 8.3: Ventana de New Project de Unity Hub

Una vez creado el proyecto, tenemos que llevar a cabo la integración de C# en Microsoft Visual Studio o en Visual Studio Code. Para ello, realizamos los siguientes pasos:

1. Ir al **Editor de Unity** > **Edit** > **Preferences**.
2. Dentro del panel de **Preferences** > **External Tools**.
3. En **External Script Editor** seleccionamos el IDE en cuestión. En mi caso, trabajaré con Visual Studio Code.

Finalmente, instalaremos las siguientes extensiones para Visual Studio Code:

- **Unity Code Snippets v1.3.0:** Nos ayuda a crear clases y métodos de Unity fácilmente. Crea clases de juego como `MonoBehaviours`, `NetworkBehaviours` y `StateMachineBehaviours` fácilmente y crea también métodos comunes como `Start()`, `Update()` u `OnTriggerEnter2D()` y llamadas de registro [74].
- **C#:** Extensión C# para Visual Studio Code. Esta extensión proporciona las siguientes características dentro de VS Code:

- Herramientas de desarrollo ligeras para .NET Core.
- Gran soporte de edición de C#, incluyendo Syntax Highlighting o IntelliSense.
- Soporte de depuración para .NET Core (CoreCLR).
- Compatibilidad con proyectos project.json y csproj en Windows, macOS y Linux.

8.2. Iteración 2

Para la segunda iteración se contaron con tres semanas durante las cuales el trabajo principal consistió en elaborar el Diseño del proyecto, llevando a cabo todas las tareas asociadas, e investigar sobre la implementación de los comandos de voz en Unity.

Entrada de voz en Unity

Se decidió utilizar la API de reconocimiento de voz en Windows: `Unity.Engine.Windows.Speech` [56]. Proporcionada por Microsoft, esta tecnología permite incorporar funciones de reconocimiento de voz a las aplicaciones ejecutadas en el sistema operativo Windows. Al integrar la API de reconocimiento de voz en Unity, es posible diseñar aplicaciones capaces de entender y responder a órdenes o entradas habladas por los usuarios. Esto nos abre la posibilidad para crear experiencias interactivas que es lo que buscamos con Educaverse.

Para implementar la funcionalidad de reconocimiento de voz en el proyecto de Unity, se ofrecen las siguientes clases:

- **KeywordRecognizer**. Se utiliza para definir un reconocedor de voz que reconozca palabras clave o frases específicas. Permite especificar una lista de palabras clave o frases a reconocer y proporciona eventos para manejar los resultados del reconocimiento [75].
- **GrammarRecognizer**. Proporciona un fichero SRGS que define una gramática específica para escuchar. El archivo de gramática especifica las frases, palabras clave y reglas que el reconocedor puede entender [76].
- **DictationRecognizer**. Permite a la aplicación escuchar cualquier palabra y proporcionar al usuario una nota u otro tipo de visualización de su discurso [77].
- **PhraseRecognitionSystem**. Representa el principal punto de entrada para gestionar la funcionalidad de reconocimiento de voz. Proporciona métodos para iniciar y detener el reconocimiento de voz, así como para registrar y anular el registro de los reconocedores de palabras clave y frases [78].
- **PhraseRecognizer**. Se utiliza para definir un reconocedor de voz que reconozca frases o palabras clave específicas. Permite especificar una lista de frases o palabras clave a reconocer y proporciona eventos para manejar los resultados del reconocimiento [79].

8.3. Iteración 3

Durante la tercera iteración, se realizaron grandes avances relacionados con la escenografía de la aplicación y la lógica relativa a los comandos de voz. Como se puede ver en la figura 8.4, dentro de la columna Closed, aparecen todas las issues que se realizaron durante esta iteración.

8.3.1. Tareas completadas

Las tareas que se completaron durante esta iteración fueron las siguientes:

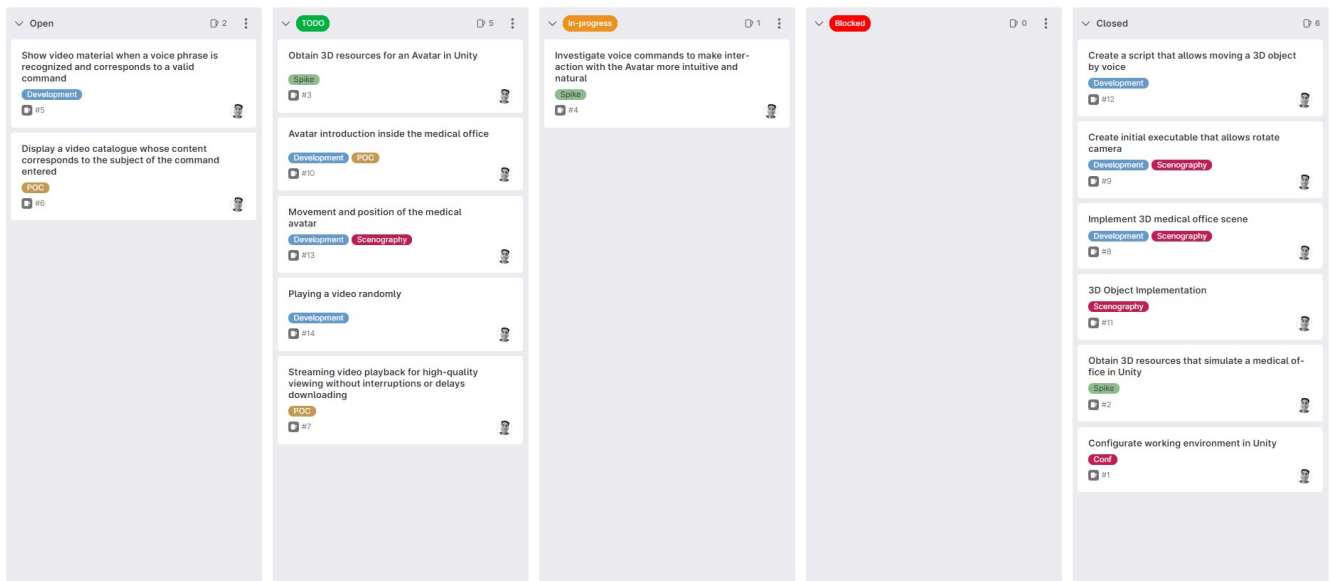


Figura 8.4: Fin de la tercera iteración

Título	Conseguir recursos 3D que simulen una consulta médica en Unity
User Story	<i>Como usuario,</i> <i>Quiero</i> visualizar una consulta médica realista en Unity <i>Para</i> poder disfrutar de una experiencia de simulación lo más cercana posible a la realidad.
Acceptance Criteria	<ul style="list-style-type: none"> ■ Los recursos 3D deben ser realistas y detallados para crear una simulación precisa de consulta médica. ■ Los recursos 3D deben ser compatibles con Unity para que puedan ser utilizados en la aplicación de la simulación. ■ Los recursos 3D deben ser proporcionados por una fuente confiable y de calidad para su precisión y realismo. ■ Los recursos 3D deben estar disponibles en diferentes formatos de archivo, para permitir su fácil integración en Unity. ■ Los recursos 3D deben ser legales y no infringir derechos de autor o propiedad intelectual. ■ La aplicación de simulación debe funcionar sin problemas y sin errores técnicos cuando se utilizan los recursos 3D proporcionados.
Label	Spike

Tabla 8.2: Issue 2

Título	Implementación de un objeto 3D
User Story	<i>Como usuario,</i> <i>Quiero</i> poder ver un objeto 3D básico proporcionado por Unity en la aplicación <i>Para</i> tener una idea de cómo lucirá el proyecto en su etapa inicial.
Acceptance Criteria	<ul style="list-style-type: none"> ■ Se incluirán varios objetos 3D básicos, entre los que se encuentran: Cubos, Esferas, Cápsulas, Cilindros, Planos o Quads. ■ Se establecerán sus propiedades como la posición, la rotación o la escala. ■ Las propiedades de los objetos 3D se modificarán para comenzar con una implementación inicial básica.
Label	Scenography

Tabla 8.3: Issue 3

Título	Implementar escena de la consulta médica 3D
User Story	<i>Como usuario,</i> <i>Quiero</i> experimentar una simulación realista de una consulta médica en la aplicación a partir de los recursos 3D, <i>Para</i> poder tener una sensación más auténtica al utilizar la aplicación y poder mejorar la interacción con ella.
Acceptance Criteria	<ul style="list-style-type: none"> ■ La escenografía debe estar basada en los recursos 3D que simulan una consulta médica, para dar la sensación de estar en un ambiente médico realista. ■ La escenografía debe ser detallada y precisa, incluyendo elementos como muebles, equipos médicos, suministros y otros objetos presentes en una consulta médica típica. ■ La escenografía debe ser coherente y uniforme, asegurando que la estética y el ambiente de la habitación se mantengan constantes a lo largo de la aplicación. ■ La escenografía debe ser compatible con otros elementos de la aplicación, como la función de comandos de voz o la reproducción de videos, permitiendo una integración perfecta de todas las funciones y elementos de la aplicación.
Label	Development, Scenography

Tabla 8.4: Issue 4

Título	Crear ejecutable inicial que permita rotar la cámara
User Story	<i>Como usuario,</i> <i>Quiero</i> poder rotar la cámara y mirar alrededor de la sala médica <i>Para</i> obtener una visión completa y detallada del entorno simulado.
Acceptance Criteria	<ul style="list-style-type: none"> ■ El ejecutable debe incluir la escenografía inicial de la aplicación, permitiendo al usuario comenzar a interactuar con ella de manera inmediata. ■ El usuario podrá mover la cámara mediante los movimientos del casco de realidad virtual y podrá mirar alrededor de la sala. ■ El ejecutable debe ser compatible con los dispositivos de realidad virtual. ■ El ejecutable debe estar libre de errores y bugs, garantizando una experiencia de usuario fluida y satisfactoria. ■ El ejecutable debe ser actualizable, permitiendo al usuario acceder a nuevas versiones de la aplicación y sus funciones de manera fácil y sencilla.
Label	Development

Tabla 8.5: Issue 5

Título	Crear un script que permita mover un objeto 3D por voz
User Story	<i>Como usuario,</i> <i>Quiero</i> interactuar con un objeto 3D básico dentro de la sala médica mediante comandos de voz <i>Para</i> poder controlar el objeto y permitir que realice movimientos básicos.
Acceptance Criteria	<ul style="list-style-type: none"> ■ El script debe importar UnityEngine.Windows.Speech para que la aplicación pueda escuchar las frases especificadas por el usuario y luego realizar las acciones en cuestión. ■ El script debe estar asociado a un objeto 3D básico para realizar los movimientos sobre él. Los movimientos a realizar son: Arriba, Abajo, Delante y Detrás. ■ La aplicación debe mostrar cómo el objeto 3D asociado al script se mueve correctamente según la palabra clave dicha por el usuario. ■ El script no debe contener bugs o errores que provoquen el fallo de la aplicación.
Label	Development

Tabla 8.6: Issue 6

8.3.2. Funcionalidad realizada

Implementación sala 3D

Durante esta segunda iteración, se obtuvieron los recursos necesarios para simular la consulta médica. Se acordó con el tutor utilizar recursos existentes, los cuales se consiguieron a través de **Unity Asset Store** [31].

Unity Asset Store es un mercado en línea donde los desarrolladores pueden comprar, vender y compartir activos, herramientas y recursos para su uso en Unity. La Asset Store ofrece una amplia variedad de recursos, como modelos 3D, texturas, animaciones, efectos de sonido, música, plugins y scripts, que pueden utilizarse para mejorar nuestro proyecto, juego o aplicación y reducir la carga de trabajo necesaria para crear herramientas o modelos desde cero. Los activos se adquieren en el sitio web de la Asset Store y pueden añadirse a los proyectos mediante el Package Manager dentro del Editor.

Una vez obtenidos los recursos 3D adecuados, procedemos a importarlos en el proyecto. Nos dirigimos a *My Assets* y ahí nos aparecerá el Asset que deseamos utilizar, en mi caso Hospital Doctor's Office (Figura 8.5). Una

vez importado el Asset en nuestro proyecto, podemos empezar a establecer cámaras, editar la sala, crear scripts o añadir objetos 3D básicos (Figura 8.6).

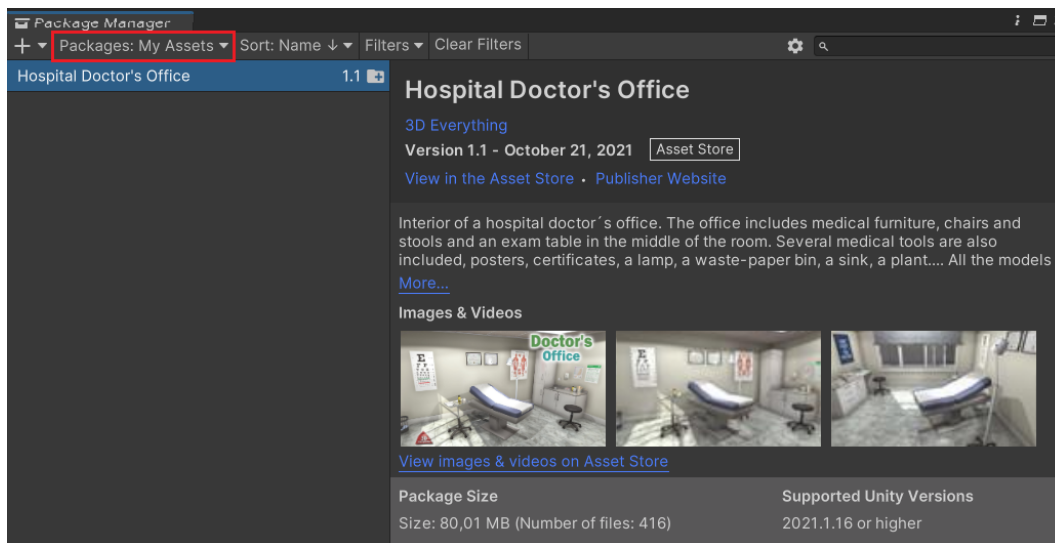


Figura 8.5: Package Manager

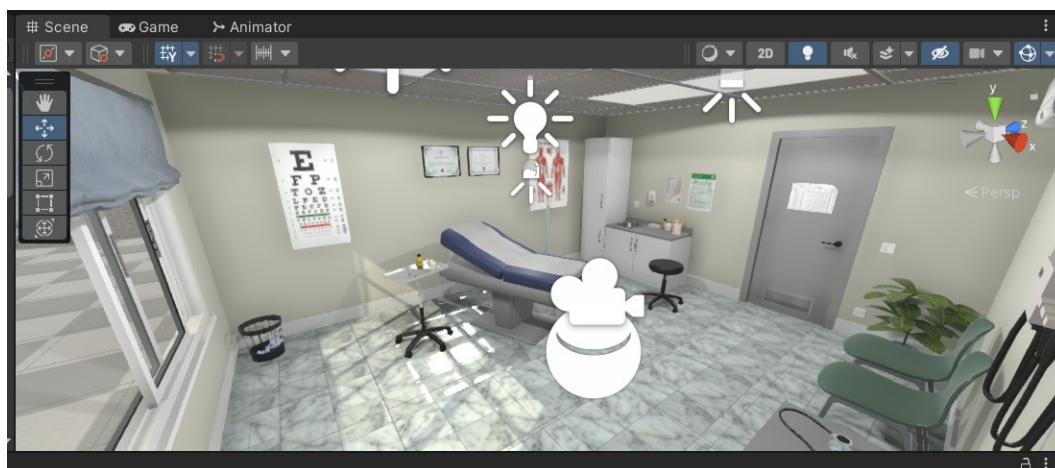


Figura 8.6: Scene View

Implementación de un objeto 3D

También se llevó a cabo la implementación de un objeto 3D básico en la sala médica. La idea de esto consistía en configurar las propiedades de dicho objeto y añadirle scripts para poder, por ejemplo, mover el objeto mediante comandos de voz, ya que durante la segunda iteración se estuvo investigando acerca de cómo llevar a cabo la entrada de voz en Unity.

Mover un objeto 3D por voz

Para llevar a cabo esta funcionalidad, se utilizó la clase `KeywordRecognizer`. Procedemos a explicar más en detalle en qué consiste dicho script:

- Comenzamos creando la clase `VoiceMovement` que hereda de `MonoBehavior` [80], clase base de la cual cada script de Unity deriva y que proporciona el framework que permite adjuntar un script a un `GameObject` en el Editor. Al heredar de `MonoBehavior`, se obtiene acceso a un conjunto de funciones y eventos incorporados que Unity llama automáticamente en momentos específicos durante la ejecución de la aplicación, como la función `Start()` o la función `Update()`.

- A continuación, creamos el KeywordRecognizer y el diccionario de palabras clave con sus respectivas acciones. Cuando comience el método Start(), se añadirán al diccionario las palabras indicadas junto con sus respectivos métodos, que van a permitir mover el objeto 3D en la dirección requerida. Luego creamos el reconocer, indicándole lo que queremos reconocer para después registrar el evento OnPhraseRecognized. Finalmente, comenzamos a reconocer.
- En cuanto al código del controlador, lo que se lleva a cabo consiste en verificar si la palabra reconocida se encuentra en el diccionario, y si es así, se procede a invocar el método correspondiente. En cuanto al código relacionado con los movimientos, este es bastante sencillo, ya que para mover el objeto 3D hacemos uso de la función *transform.Translate()* [81] cambiando los parámetros en función de la dirección.

Movimiento de la cámara

Finalmente se creó un script que permitía rotar la cámara para que el usuario pudiese visualizar toda la sala médica pero de forma estática. Para ello se creó el script RotateMoveCamera.

Dicho script, permitía al usuario rotar la cámara en respuesta a la entrada del mouse. Con esto se conseguía proporcionar una interacción intuitiva y la posibilidad de explorar diferentes ángulos y perspectivas en la aplicación. Más adelante se configurará al proyecto para realizar el movimiento de la cámara a través de los dispositivos de realidad virtual.

8.4. Iteración 4

Para la cuarta iteración se contó con tres semanas, durante las cuales las tareas principales consistieron en implementar el avatar médico dentro de la sala, crear un script que permitiese distinguir las categorías de vídeos y mostrar vídeos de forma aleatoria, además de continuar con la implementación de la memoria del proyecto. El aspecto del tablón de issues al final de la cuarta iteración se muestra en la siguiente figura:

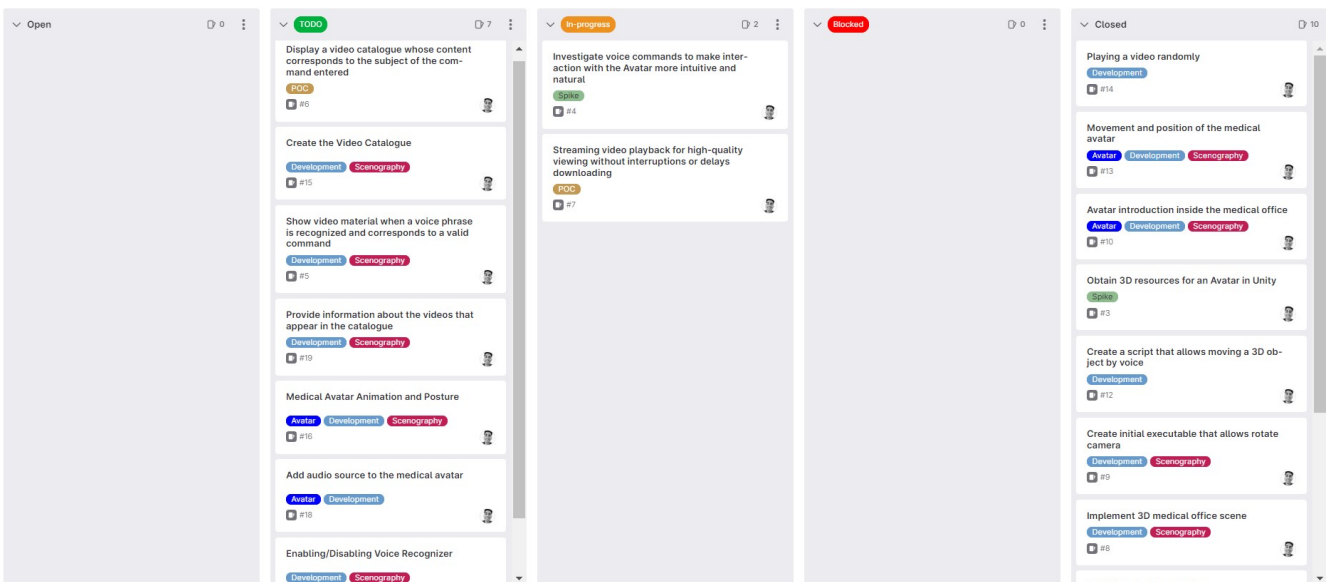


Figura 8.7: Fin de la cuarta iteración

8.4.1. Tareas completadas

Los issues que se implementaron durante esta iteración fueron los siguientes:

Título	Conseguir recursos 3D para un avatar en Unity
User Story	<i>Como usuario,</i> <i>Quiero</i> visualizar un avatar médico 3D dentro de la sala médica <i>Para</i> poder interactuar con él de forma sencilla y poder sentir una experiencia más inmersiva.
Acceptance Criteria	<ul style="list-style-type: none"> ▪ Los recursos 3D deben ser compatibles con Unity. ▪ Los recursos 3D deben ser fáciles de integrar y utilizar en la aplicación de realidad virtual. ▪ Los recursos 3D deben ser legales y no infringir derechos de autor o propiedad intelectual. ▪ La aplicación debe funcionar sin problemas y sin errores técnicos cuando se utilizan los recursos 3D proporcionados para configurar el avatar.
Label	Spike

Tabla 8.7: Issue 7

Título	Incorporación del avatar dentro de la sala médica
User Story	<i>Como usuario,</i> <i>Quiero</i> interactuar con el avatar médico <i>Para</i> poder obtener información sobre la enfermedad de la diabetes de una manera amigable y personalizada.
Acceptance Criteria	<ul style="list-style-type: none"> ▪ El avatar médico debe estar diseñado a partir de los recursos 3D conseguidos y ser capaz de interactuar con el usuario de manera natural. ▪ El avatar médico debe ser capaz de responder a las preguntas del usuario de manera clara y concisa, proporcionando información útil a través de los videos educativos.
Label	DEVELOPMENT, Scenography

Tabla 8.8: Issue 8

Título	Movimiento y posición del Avatar médico
User Story	<i>Como usuario,</i> <i>Quiero</i> poder visualizar el avatar médico de manera correcta <i>Para</i> interactuar con él sin dificultades.
Acceptance Criteria	<ul style="list-style-type: none"> ▪ El avatar medico estará situado en la misma posición en la que estaba el objeto 3D básico que se creó con anterioridad, es decir, justo delante de la visión que tendrá el usuario cuando comience la aplicación. ▪ El avatar médico deberá permanecer estático por el momento. Se le añadirán animaciones más adelante. ▪ El avatar debe poder visualizarse de manera correcta sin que haya ningún elemento de la sala médica que se sobreponga o lo oculte.
Label	DEVELOPMENT, Scenography

Tabla 8.9: Issue 9

Título	Implementar comandos de voz para que la interacción con el avatar sea más intuitiva y natural
User Story	<i>Como usuario,</i> <i>Quiero</i> interactuar con el avatar a través de comandos de voz relacionados con la diabetes <i>Para</i> poder visualizar el material videográfico relevante y poder obtener información en tiempo real mientras utilizo la aplicación.
Acceptance Criteria	<ul style="list-style-type: none"> ■ La aplicación debe permitir al usuario interactuar con el avatar mediante comandos de voz relacionados con la diabetes, como “<i>Necesito información sobre los tipos de insulina</i>” o “<i>¿Qué es la hipoglucemia?</i>”. ■ La aplicación debe tener un sistema de reconocimiento de voz preciso y eficaz que permita al usuario hablar con el avatar de manera fluida y sin interrupciones. ■ La aplicación debe tener un mecanismo para detectar errores en la interpretación de los comandos de voz, y debe proporcionar un mensaje adecuado en caso de que el comando no sea entendido correctamente. ■ La aplicación debe estar diseñada de manera que los comandos de voz se integren de manera fluida con otras funciones y características de la aplicación.
Label	Spike, POC

Tabla 8.10: Issue 10

Título	Reproducir un vídeo de forma aleatoria
User Story	<i>Como usuario,</i> <i>Quiero</i> reproducir un video aleatorio relacionado con el comando de voz que haya introducido, <i>Para</i> poder disfrutar de contenido audiovisual relevante y tener una experiencia funcional de la aplicación.
Acceptance Criteria	<ul style="list-style-type: none"> ■ Se debe mostrar un vídeo cuando se detecte un comando de voz del usuario. ■ Se filtrará el vídeo en función de la palabra clave, pero de momento no se mostrará el catálogo con los distintos vídeos disponibles, sino que se reproducirá un vídeo aleatorio. ■ El vídeo debe reproducirse en buena calidad, sin cortes ni interrupciones. ■ El tamaño de la ventana donde aparezca el vídeo debe tener un tamaño adecuado, para que el usuario pueda visualizarlo de manera correcta. ■ El vídeo no debe ser ocultado por ningún elemento de la sala médica o por el avatar médico.
Label	DEVELOPMENT

Tabla 8.11: Issue 11

8.4.2. Funcionalidad Realizada

Implementación del avatar médico 3D

Tal y como se muestra en las tareas completadas de esta iteración, se añadió el avatar a la sala médica. Los recursos de este avatar médico se consiguieron a través de Sketchfab [82].

Sketchfab es una plataforma en línea que permite a los usuarios cargar, compartir y ver modelos 3D interactivos. Proporciona una amplia biblioteca de modelos 3D creados por la comunidad, así como por artistas y diseñadores profesionales. Además, Sketchfab ofrece una variedad de herramientas y funciones para mejorar la visualización y la interacción con los modelos. Para añadir Sketchfab a Unity seguiremos los siguientes pasos:

1. Descargar e instalar el complemento de Sketchfab para Unity [83].
2. Una vez instalado el complemento, iniciamos el Editor de Unity. En la ventana del proyecto de Unity, vamos a la pestaña **Assets > Import Package > Custom Package**.
3. Buscamos el archivo del complemento de Sketchfab que descargamos previamente. Hacemos clic en Import para importar el paquete a nuestro proyecto de Unity.
4. Después de importar el complemento, se agregarán nuevos elementos al menú de Unity. Podemos acceder a ellos desde la barra de menú superior (Figura 8.8).

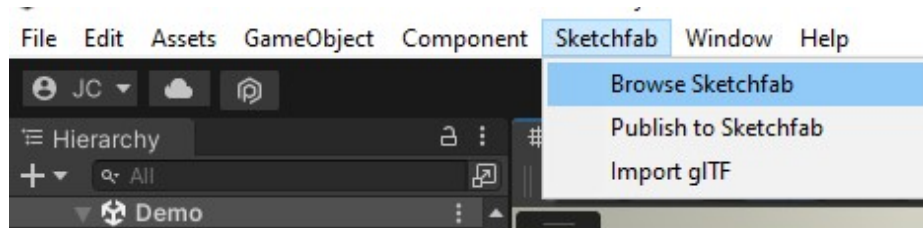


Figura 8.8: Barra de menú

5. Para añadir el avatar médico de Sketchfab a la escena de Unity, seleccionamos **Sketchfab > Browse Sketchfab**. Nos aparecerá una ventana con distintos modelos disponibles. Lo primero que deberemos hacer es iniciar sesión para después buscar el modelo que nos interese (Figura 8.9).

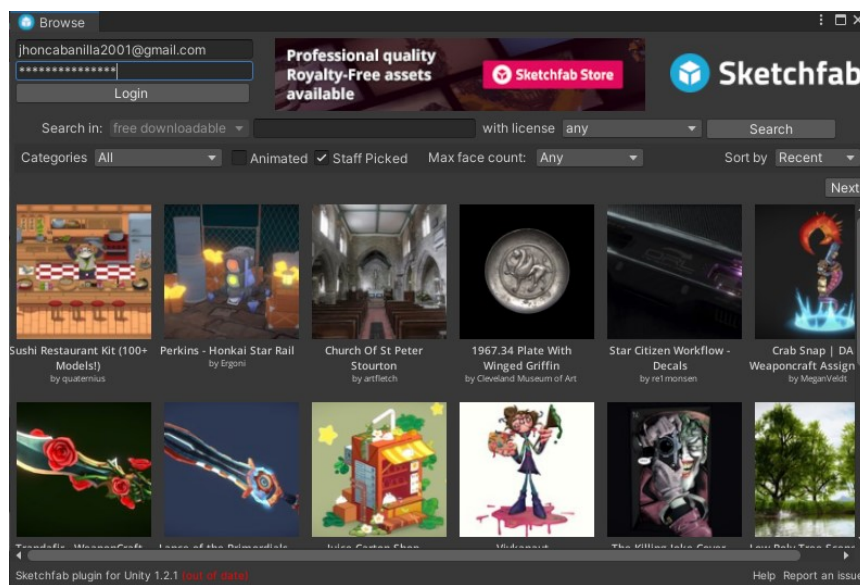


Figura 8.9: Ventana Browse Sketchfab

6. Procedemos a buscar el modelo del avatar médico [68]. Seleccionamos el modelo que deseamos añadir a la escena y nos aparecerá una ventana donde podremos descargar el modelo 3D (Figura 8.10).
7. Una vez descargado e importado el modelo, podremos arrastrarlo y soltarlo en nuestra escena de Unity. A partir de ahí, será posible manipularlo, ajustar su posición, escala, agregar animaciones u otras funcionalidades propias de Unity (Figura 8.11).

En cuanto a la posición del avatar, este se encuentra donde antes estaba situado el objeto 3D básico, es decir, justo en frente de la vista que tiene el usuario cuando arranca la aplicación. En relación con la animación del avatar, por el momento permanecerá estático, más adelante se le dará otra postura o incluso se implementarán animaciones.

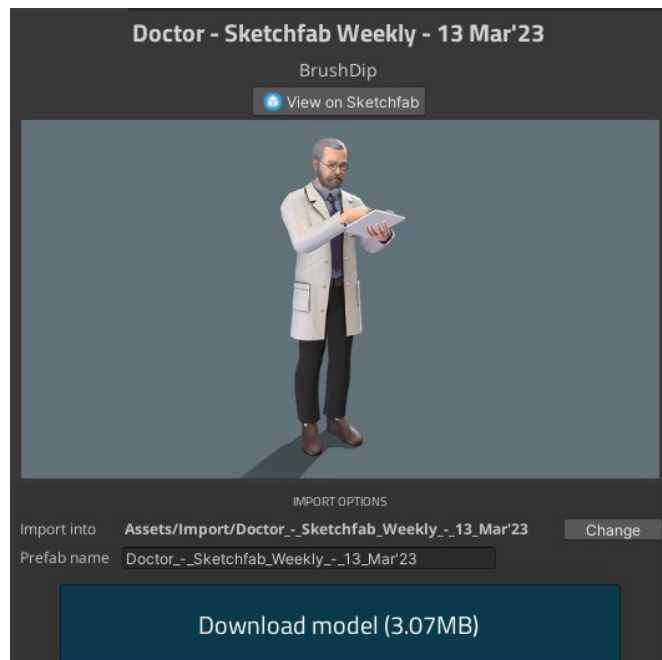


Figura 8.10: Modelo médico. Fuente [68]



Figura 8.11: Introducción del avatar en la sala médica

Implementación comandos de voz

Otro avance que se consiguió durante esta iteración fue el de mejorar el script para poder identificar palabras clave que el usuario decía, con el fin de poder mostrar un vídeo. La implementación inicial del script permitía mover un objeto 3D mediante el uso de la clase KeywordRecognizer, por lo que si el usuario dictaba la palabra “Arriba” o “Abajo”, dicho objeto se iba a desplazar en la dirección correspondiente. Sin embargo, esta implementación resultaba demasiado simple para reconocer comandos más específicos relacionados con la diabetes.

Por lo tanto, con el fin de permitir al usuario introducir comandos más elaborados se utilizó la clase DictationRecognizer, la cual permite convertir en texto todo lo que el usuario haya pronunciado. Gracias a esta mejora, los usuarios podían introducir comandos más detallados y así disfrutar de la visualización de vídeos. A continuación, se explica más en detalle en qué consistieron las mejoras realizadas:

Procesamiento de lenguaje natural

Como se ha explicado, mediante el uso de la clase DictationRecognizer, podemos transformar en texto la entrada de voz realizada por el usuario. A partir de ese momento, se deben aplicar ciertas técnicas del procesamiento del lenguaje natural para poder extraer información y poder clasificar en categorías predefinidas, las entidades que se encuentren en el texto.

Dichas categorías se corresponden con la clasificación realizada de los vídeos. En total, Educaverse cuenta con 28 vídeos clasificados de la siguiente manera:

- **Tipos de diabetes**

Vídeos que explican los tipos de diabetes que hay, como la diabetes gestacional o la diabetes mellitus tipo 1.

- **Tipos de insulina**

Vídeos relacionados con los diferentes tipos de insulina como la insulina basal o la insulina intermedia.

- **Técnica de inyección de insulina**

Vídeo descriptivo sobre cómo llevar a cabo la técnica de inyección de insulina.

- **Administración de la insulina**

Vídeo relacionado con la aplicación de la insulina.

- **Bomba de insulina**

Vídeo en el que se explica qué es y cómo funciona la bomba de insulina.

- **Medicamentos para la diabetes**

Vídeos relacionados con los antidiabéticos y sus correspondientes explicaciones.

- **Glucómetro**

Vídeos relacionados con el glucómetro, como por ejemplo, el control de la glucemia capilar o el sensor de glucosa.

- **Hipoglucemia**

Vídeos relacionados con la hipoglucemia, sus síntomas, la regla del 15 o la hipoglucemia grave.

- **Cuidado personal**

Vídeos relacionados con el control metabólico para evitar complicaciones en los pacientes, o sobre cómo programar viajes y vacaciones.

- **Nutrición**

Vídeos relacionados con la alimentación que debe seguir el paciente.

- **Actividad física**

Vídeos relacionados con el ejercicio en los pacientes, así como los controles glucémicos antes y después del ejercicio.

El método del procesamiento de lenguaje natural que se implementó para extraer información del texto es el método de Reconocimiento de entidades nombradas [84].

El reconocimiento de entidades nombradas (NER, por sus siglas en inglés) es un método que se utiliza para identificar y clasificar entidades específicas en el texto. Estas entidades pueden ser nombres de personas, organizaciones, lugares, fechas, cantidades, entre otros tipos de información.

El reconocimiento de entidades nombradas es utilizado en una variedad de aplicaciones del NLP, como la extracción de información, la traducción automática, la respuesta automática y la generación de resúmenes. Proporciona una forma eficiente de identificar y organizar información importante en el texto, lo que facilita el procesamiento y análisis posterior.

Evidentemente, las entidades que nos interesan para el proyecto no serán lugares o fechas, sino que se corresponderán con las categorías definidas de los vídeos. Para llevar a cabo la implementación correspondiente, se optó por utilizar un sistema basado en diccionarios, es decir, se utilizó un diccionario con un extenso vocabulario y una colección de sinónimos para identificar las entidades.

A continuación se muestran unos textos de ejemplo con la categorización correspondiente:

- ¿En qué consiste la diabetes mellitus tipo 1 (*tipos diabetes*) y la diabetes gestacional (*tipos diabetes*)?
- ¿Cómo se debe llevar a cabo la técnica de inyección de insulina (*técnica inyección*)?
- Control y monitorización de la glucemia capilar (*glucómetro*).
- Quiero saber qué son los antidiabéticos orales (*medicamentos*) y los antidiabéticos inyectables (*medicamentos*).
- Información sobre cómo realizar una lectura correcta del monitor de glucosa (*glucómetro*), cómo llevar una dieta equilibrada (*nutrición*) y cómo programar viajes (*cuidado personal*).

Como se puede observar en los ejemplos previos, se realiza una clasificación apropiada basada en el vocabulario definido para cada categoría. No obstante, si el usuario pronuncia un comando que pertenece a múltiples categorías de vídeos, como se ilustró en el último ejemplo anterior, la cantidad de vídeos que se mostrarían sería considerable y la presentación visual del catálogo complicaría la navegación del usuario. Por esta razón, se tomó la decisión, en colaboración con los tutores, de mostrar únicamente los vídeos relacionados con una sola categoría. Esto implica que el usuario debe introducir comandos de voz que pertenezcan a una única categoría, como se muestra en el resto de ejemplos anteriores, lo cual facilitará la visualización del catálogo para el usuario final.

Otro aspecto relevante es que los usuarios tendrán la capacidad de visualizar los vídeos al dictar frases simples y directamente relacionadas con la categoría de los mismos. Por ejemplo, frases como “tipos de diabetes” o “glucómetro” serán reconocidas por el sistema de reconocimiento de voz y les mostrará de manera directa los vídeos asociados a dichas categorías. Esta funcionalidad está diseñada especialmente para usuarios principiantes con un conocimiento limitado sobre la enfermedad de la diabetes. No obstante, a medida que los pacientes adquieran mayor conocimiento, podrán emplear comandos más elaborados para acceder a los vídeos proporcionados por Educaverse.

A continuación, se procede a explicar el resto de técnicas de NLP utilizadas para la validación de los comandos de voz:

- **Tokenización.** La tokenización es, por lo general, uno de los primeros pasos del proceso de NLP, un paso que divide las cadenas de texto más largas en trozos más pequeños, o tokens. Generalmente, el procesamiento posterior se realiza una vez que el texto ha sido debidamente tokenizado [86].
- **Stop words.** Esta técnica consiste en filtrar ciertas palabras antes de continuar procesando el texto. Estas palabras filtradas son conocidas como Stop words y suelen ser las palabras más comunes en un idioma, que aportan poco al significado global del texto [86].

Después de que se haya validado el comando de voz, se realiza la invocación de la acción correspondiente asociada a la entidad identificada. En caso de que no se encuentre ninguna coincidencia con el comando de voz, se muestra un panel al usuario que contiene un mensaje específico indicando que no se han encontrado resultados. Esta

estrategia tiene como objetivo proporcionar retroalimentación al usuario, informándole de la falta de resultados para el comando ingresado y motivándolo a introducir un nuevo comando. De esta manera, se busca mejorar la experiencia del usuario y facilitar su interacción con la aplicación.

Implementación reproducción vídeo

Para llevar a cabo la reproducción de un vídeo de forma aleatoria relacionado con la categoría reconocida, se siguieron los siguientes pasos:

- Una vez validado el comando, dentro de la función asociada a cada una de las acciones, se obtienen los videoClips [87] correspondientes de la aplicación. Una vez obtenidos, se procede a seleccionar uno de ellos al azar, utilizando un índice aleatorio de la lista de vídeos disponibles.
- Una vez que se ha determinado qué vídeo se reproducirá, se manda al controlador del canvas. Este controlador se encarga de actualizar el canvas para ocultar el catálogo, en el caso de que estuviese activo, y de activar el GameObject de la Interfaz de usuario (UI) [88] llamado VideoScreen. Este objeto representa la pantalla donde se reproducirá el vídeo seleccionado.
- Dentro del objeto VideoScreen, se encuentra el componente VideoPlayer [89]. Este componente es esencial para la reproducción del vídeo y proporciona las funcionalidades necesarias.
- El componente VideoPlayer tiene una propiedad llamada Target Texture, la cual se configura con una Render Texture [90]. Una Render Texture es un objeto que permite capturar y almacenar la salida visual de una cámara o de otro componente gráfico, donde se renderizará el contenido del vídeo.
- Por último, se establece la Render Texture creada como la textura del objeto VideoScreen. Esto permite que el vídeo renderizado en la Render Texture se muestre en la pantalla del VideoScreen, logrando así la reproducción visual del vídeo.

8.5. Iteración 5

Penúltima iteración en la que se llevaron a cabo grandes avances en la aplicación como la creación del catálogo de vídeos o la implementación del control de los vídeos.

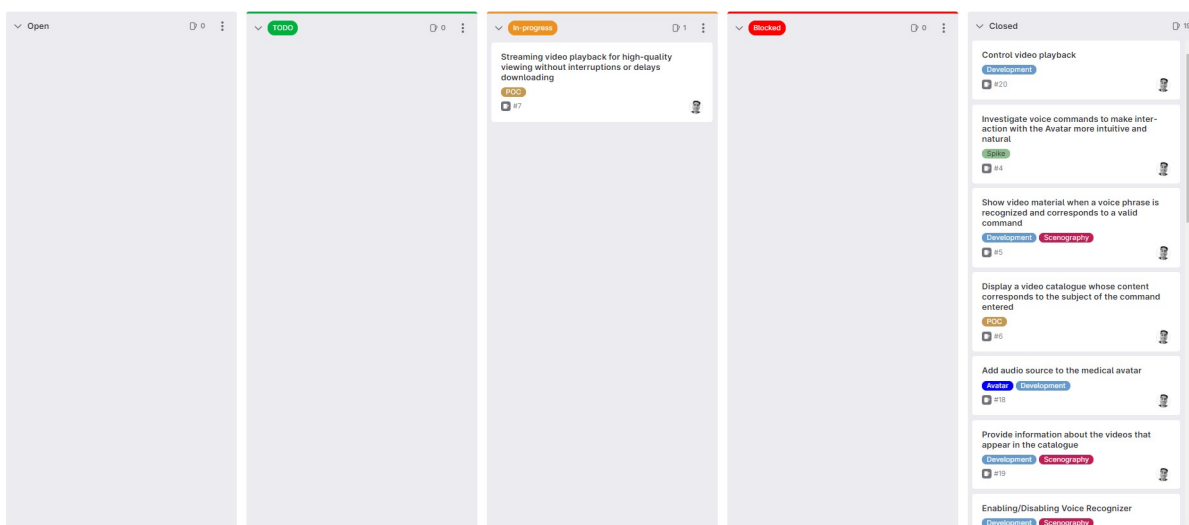


Figura 8.12: Fin de la quinta iteración

8.5.1. Tareas Implementadas

Los issues que se completaron durante la quinta iteración fueron los siguientes:

Título	Creación del Catálogo de Videos
User Story	<i>Como usuario,</i> Quiero visualizar un catálogo de vídeos <i>Para</i> poder encontrar y reproducir los vídeos disponibles en la aplicación.
Acceptance Criteria	<ul style="list-style-type: none"> ■ El catálogo de vídeos debe ser visible cuando se reconozca un comando válido, es decir, cuando se detecte una palabra clave relacionada con una categoría de vídeos. ■ El catálogo de vídeos debe incluir las miniaturas de los vídeos que estén disponibles en este momento. ■ El GameObject en el que esté situado el catálogo debe incluir un GridLayoutGroup al cual se le ajustará el valor de espaciado en función del número de vídeos que se vayan a mostrar. ■ El catálogo de vídeos debe tener un tamaño correcto para que el usuario pueda visualizar las miniaturas de los vídeos sin ningún problema. ■ El tamaño de las imágenes que aparecen en el catálogo debe ajustarse dinámicamente en función del número de vídeos que se muestren.
Label	Development, Scenography

Tabla 8.12: Issue 12

Título	Animación y Postura del Avatar médico
User Story	<i>Como usuario,</i> Quiero que el avatar médico realice una animación cuando comience a ejecutarse la aplicación y que adopte una postura interesante <i>Para</i> poder tener una experiencia más atractiva al interactuar con el avatar, evitando una postura estática y poco interesante.
Acceptance Criteria	<ul style="list-style-type: none"> ■ El avatar médico deberá realizar una animación cuando comience la aplicación, la cual tendrá una duración de 3.433 segundos. ■ El avatar médico será configurado con una postura que indicará visualmente que está tomando nota de la información proporcionada por el usuario. Esta postura se mantendrá consistente durante toda la aplicación.
Label	Avatar, Development, Scenography

Tabla 8.13: Issue 13

Título	Activar/Desactivar Reconocedor de Voz
User Story	<i>Como usuario,</i> Quiero poder activar el reconocedor de voz <i>Para</i> comenzar a introducir comandos de voz y poder visualizar el catálogo con los vídeos disponibles.
Acceptance Criteria	<ul style="list-style-type: none"> ■ El reconocedor de voz se deberá activar cuando el usuario pulse sobre el avatar médico. ■ Cuando se active el reconocedor de voz y se detecte un comando válido, entonces se mostrará el catálogo de vídeos y el reconocedor se volverá a desactivar. ■ Si el reconocedor de voz no detecta un comando válido, se mostrará un mensaje indicativo y el reconocedor se volverá a desactivar.
Label	Development, Scenography

Tabla 8.14: Issue 14

Título	Proporcionar información acerca de los vídeos que aparecen en el catálogo
User Story	<i>Como usuario,</i> Quiero visualizar el título de cada uno de los vídeos <i>Para</i> poder identificar rápidamente los videos disponibles y navegar por el catálogo para encontrar el video más adecuado a mis necesidades.
Acceptance Criteria	<ul style="list-style-type: none"> ■ Cuando el usuario pase con el ratón o un dispositivo de RV sobre una de las imágenes que se muestran en el catálogo, se debe mostrar una pequeña ventana emergente que contenga el título del vídeo correspondiente, permitiendo al usuario obtener información instantánea. ■ La posición en la que aparecerá la ventana emergente debe ser determinada convirtiendo la posición del ratón o del dispositivo de RV al espacio del mundo en Unity. ■ La ventana emergente con el título debe aparecer medio segundo después de que el usuario se coloque sobre la imagen del vídeo. Esta breve pausa proporciona al usuario un tiempo suficiente para que se muestre la información relevante sin ser demasiado intrusiva o interferir con la navegación fluida del catálogo. ■ El texto que aparece en la ventana debe ser legible y estar claramente asociado al vídeo al que pertenece la imagen.
Label	Development, Scenography

Tabla 8.15: Issue 15

Título	Añadir audio al Avatar médico
User Story	<i>Como usuario,</i> Quiero escuchar una fuente de audio por parte del avatar médico <i>Para</i> obtener una guía clara y concisa sobre cómo interactuar de manera adecuada con la aplicación. Al tener acceso a esta guía de audio, espero reducir la carga cognitiva al no tener que leer las instrucciones en la pantalla, facilitando el proceso de inicio.
Acceptance Criteria	<ul style="list-style-type: none"> ■ El audio debe reproducirse cuando comience a ejecutarse la aplicación. ■ La voz y el tono que se elijan para el audio deben ser adecuados, permitiendo al usuario entender correctamente las instrucciones de la aplicación. ■ El audio de bienvenida debe establecer la identidad de la aplicación y agregar un toque personalizado que permita al usuario una experiencia única que refleje el propósito de la aplicación.
Label	Development, Avatar

Tabla 8.16: Issue 16

Título	Controlar la reproducción de los vídeos
User Story	<i>Como usuario,</i> Quiero controlar la reproducción de los vídeos <i>Para</i> poder ajustar la reproducción según mis necesidades y preferencias.
Acceptance Criteria	<ul style="list-style-type: none"> ■ El usuario debe ser capaz de pausar y reanudar la reproducción de un vídeo mediante la tecla “Espacio” para tener la capacidad de detener temporalmente el contenido y continuar viéndolo más tarde. ■ El usuario debe ser capaz de adelantar o retroceder la reproducción de un vídeo mediante las teclas de “Flecha Izquierda” y “Flecha Derecha” para saltar partes del contenido o para revisar secciones anteriores. ■ El usuario debe ser capaz de detener la reproducción de un vídeo y quitarlo mediante la tecla “Escape” para volver a seleccionar un nuevo vídeo.
Label	Development

Tabla 8.17: Issue 17

8.5.2. Funcionalidad Realizada

Implementación del catálogo de vídeos

Para llevar a cabo la creación del catálogo de vídeos, se realizó el siguiente procedimiento:

- Se comenzó creando un Canvas [91], el cual proporciona un espacio en el que se pueden colocar elementos de interfaz, como imágenes, texto, botones y vídeos, para que se muestren en la pantalla de la aplicación.
- Dentro del Canvas, se crearon los siguientes GameObjects:
 - **Screen:** Es un GameObject que contiene el catálogo de vídeos, mostrando las imágenes correspondientes a cada vídeo disponible. El objetivo de este componente es proporcionar al usuario una interfaz visual para navegar y seleccionar los vídeos.
 - **VideoScreen:** Es otro GameObject que se utiliza para mostrar los vídeos. Este objeto tiene asociado el componente VideoPlayer, que se encargará de reproducir los vídeos seleccionados por el usuario.
 - **TipWindow:** Es un elemento de interfaz tipo Image que contiene el título del vídeo correspondiente. Este componente se utiliza para proporcionar información adicional al usuario sobre el contenido del vídeo que se está mostrando.

- **InvalidCommandPanel**: Es un panel que se muestra al usuario cuando el comando introducido no coincide con ninguno de los disponibles. Proporciona una retroalimentación clara y concisa al usuario en caso de que el comando no sea reconocido, evitando confusiones o malentendidos.
- Al `GameObject Screen` se le añadió el componente `GridLayoutGroup` [92] que permite controlar la disposición y el espaciado de los elementos secundarios, en nuestro caso `Raw Images`, en filas y columnas. Se estableció el número de filas a 2, y el número de columnas se establece en función del número de vídeos que haya disponible.
- Como se ha mencionado, para añadir las miniaturas de los vídeos, se utilizaron `Raw Images` [93], componente que se utiliza para mostrar imágenes o texturas de forma dinámica en un canvas. A diferencia del componente `Image`, el `Raw Image` permite trabajar con una variedad más amplia de texturas, incluyendo imágenes en formato JPEG, PNG y otros formatos comunes.
- En cuanto a las imágenes que se muestran en el catálogo en un momento dado, a cada una de ellas se le añade el script `ImageController`, el cual controla los eventos que desencadena el usuario sobre el catálogo. Para ello implementa tres interfaces: `IPointerClickHandler`, `IPointerEnterHandler` y `IPointerExitHandler`, lo que permite reproducir videos al hacer clic en las imágenes, mostrar tooltips al pasar el cursor sobre las imágenes, ocultar y mostrar los elementos adecuados en la interfaz del canvas, y restablecer el catálogo después de que un vídeo se haya reproducido por completo. Este script permitía la navegación por el catálogo utilizando el ratón como dispositivo de entrada. Sin embargo, posteriormente se realizó una actualización en el script 8.6.2 para controlar los eventos y poder interactuar con el catálogo utilizando dispositivos de realidad virtual.
- Otro componente importante que se añadió a cada `Raw Image` fue el `BoxCollider` [94], el cual permite detectar cuando el usuario interactúa con cada una de las imágenes del catálogo. Esto nos permitió detectar qué imagen pulsaba el usuario y reproducir el vídeo correspondiente.
- Por último, se implementó la funcionalidad relacionada con el ajuste del espaciado entre las imágenes que se encuentran en el grid y el tamaño de cada una de ellas en función del número de vídeos presentes en el catálogo.

Los vídeos que deben ser mostrados los recibe el controlador del canvas a través del Reconocedor, quien obtiene las imágenes, los títulos y los vídeos de la aplicación correspondientes al comando validado. El controlador del canvas comprueba los datos recibidos y actualiza la interfaz del canvas invocando el método correspondiente.

Implementación del control de los vídeos

Otra funcionalidad que se consiguió implementar fue permitir el control y ajuste de la reproducción de un vídeo de acuerdo a las necesidades del usuario. Para ello se utilizaron eventos de teclado [95] y se asignaron una serie de teclas para realizar las acciones específicas. Dicho controlador permite pausar o reproducir el vídeo actual mediante un método llamado `PauseVideo`. Además, se pueden realizar avances rápidos utilizando `FastForwardVideo` y rebobinados con `RewindVideo`. Asimismo, el método `ExitVideo` permite salir del vídeo, desencadenando el evento asociado. A continuación se explica en detalle las funciones anteriores:

- **PauseVideo (KeyCode = Space)**. Se comprueba si el vídeo está reproduciéndose actualmente, utilizando la propiedad `isPlaying` del `VideoPlayer`. En función del estado del vídeo, se llamará al método `Pause()` o al método `Play()` del `VideoPlayer`.
- **FastForwardVideo (KeyCode = RightArrow)**. Para adelantar el vídeo se utilizó la propiedad `time` del `VideoPlayer` que representa el tiempo de reproducción del vídeo. Por lo cual, incrementamos el valor de dicha propiedad en `2f`, lo que hará que el vídeo avance a una velocidad de 2 unidades por cada unidad de tiempo en la reproducción.
- **RewindVideo (KeyCode = LeftArrow)**. Para retroceder el vídeo seguimos el mismo procedimiento que para adelantar, lo único que ahora debemos restar `2f` al valor de la propiedad `time` del `VideoPlayer`.
- **ExitVideo (KeyCode = Escape)**. Para detener la reproducción del vídeo se utilizó la propiedad `frameCount` del `VideoPlayer`, la cual nos devuelve el número total de fotogramas del vídeo. Por lo tanto, se incrementa el valor de dicha propiedad y, mediante el controlador de imagen, se vuelve a mostrar el catálogo de los vídeos cuando se detecte el final de la reproducción.

Una propiedad que hubo que activar para todos los vídeo clips fue la siguiente: Transcode. Dicha propiedad se refiere al proceso de convertir o codificar los archivos de vídeo en un formato compatible y optimizado para su reproducción en tiempo real dentro del motor de Unity. Además, cuando se crea un ejecutable para Windows, es necesario transcodificar los VideoClips debido a las diferencias en la compatibilidad de formatos y códecs de vídeo entre plataformas.

Activar/Desactivar Reconocedor de Voz

Otro requisito importante que se implementó fue la posibilidad de activar y desactivar el reconocedor de voz, ya que en versiones anteriores dicho reconocedor siempre estaba activo, lo cual causaba ciertos problemas. Por ello, se crearon las siguientes funciones: `StartDictationRecognizer()` y `StopDictationRecognizer()`.

La función **`StartDictationRecognizer()`** se encarga de activar el marcador visual del avatar y el reconocimiento de dictado de voz, adjuntando un controlador de eventos para recibir y manejar los resultados del dictado, lanzando una excepción si el reconocimiento ya estaba en curso, o mostrando un mensaje de error en la consola de depuración si se produce algún error durante el inicio.

Por otra parte, la función **`StopDictationRecognizer()`** se encarga de detener el reconocimiento de dictado de voz en curso. Desactiva el marcador visual y, si el reconocimiento está en ejecución, se detiene. Si el reconocimiento no está en curso o se produce un error durante la detención, se muestra un mensaje de error en la consola de depuración.

Una vez creadas las funciones, se añade el componente Button al avatar médico para que cuando el usuario interactúe con él, se active la función indicada que se encuentra dentro del controlador del avatar. Para ello utilizamos la propiedad `On Click()` que nos quedará de la siguiente manera:

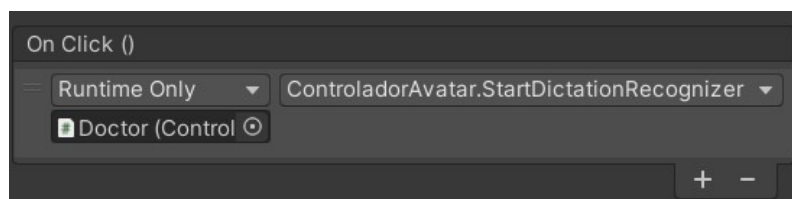


Figura 8.13: Sección `On Click()` del Avatar Médico

Por otra parte, cuando el reconocedor detecte un comando válido se creará y mostrará el catálogo correspondiente y se desactivará el reconocedor. Dicha desactivación se mantendrá hasta que el usuario no vuelva a interactuar con el avatar médico, momento en el cual se reactiva el reconocedor y vuelve a estar listo para capturar y procesar comandos de voz.

Implementación de la animación del avatar

Como se había comentado previamente, un objetivo pendiente era cambiarle la postura al avatar médico y añadirle una animación a este con el objetivo de aumentar el realismo, la comunicación no verbal, la orientación visual y la interacción con la aplicación. Estas características contribuyen a una experiencia más envolvente y atractiva para el usuario, fomentando una mayor participación e involucramiento.

Para llevar a cabo la animación del avatar se utilizó un `AnimationClip` [96] proporcionado en la carpeta `animations` del avatar. Un `AnimationClip` es un archivo que contiene información sobre la animación de un objeto en el juego. Consiste en una secuencia de fotogramas (`keyframes`) que definen cómo se mueve, transforma o cambia visualmente un objeto a lo largo del tiempo.

El `AnimationClip` se puede crear y editar en el Editor de Animación de Unity, donde se puede ajustar la duración, la velocidad y otros parámetros de la animación. También se pueden agregar curvas de animación para controlar la intensidad, la aceleración y otros aspectos de la animación en función del tiempo.

Una vez que se ha creado un `AnimationClip`, se puede asignar a un objeto en la escena y reproducirlo utilizando el componente `Animator` [97]. El `Animator` interpreta y aplica el `AnimationClip` al objeto, permitiendo que la animación se reproduzca de acuerdo con las configuraciones y transiciones definidas.

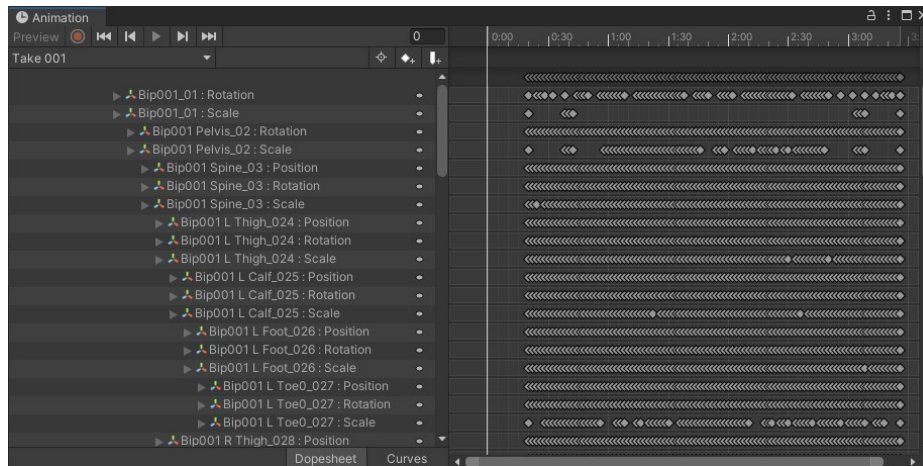


Figura 8.14: AnimationClip - Take 001

Dentro del Animator de Unity, se utilizó la capa **Base Layer** (Figura 8.15), la cual es una capa predeterminada que se utiliza para controlar la animación base de un objeto. Esta capa contiene estados que definen diferentes momentos o estados de animación a lo largo del tiempo. Dichos estados son los siguientes:

- El estado **Entry** es el punto de entrada de la animación. Se utiliza para iniciar la reproducción de la animación cuando se activa la capa. El estado de entrada puede tener una transición saliente hacia otro estado, en este caso Take 001, que es un AnimationClip. Esto significa que cuando se inicia la capa, la animación se reproducirá automáticamente utilizando dicho clip.
- El AnimationClip **Take 001** es el archivo que contiene la información y los fotogramas clave de la animación en sí. Define cómo se mueve, transforma o cambia visualmente el objeto durante la animación (Figura 8.14).
- El estado **Any State** es un estado especial que no está vinculado directamente a ningún estado específico. Se utiliza para definir transiciones y condiciones que se aplicarán independientemente del estado actual en el que se encuentre la animación. Es útil para establecer transiciones genéricas o de respaldo que se aplican en cualquier estado de animación en la capa.

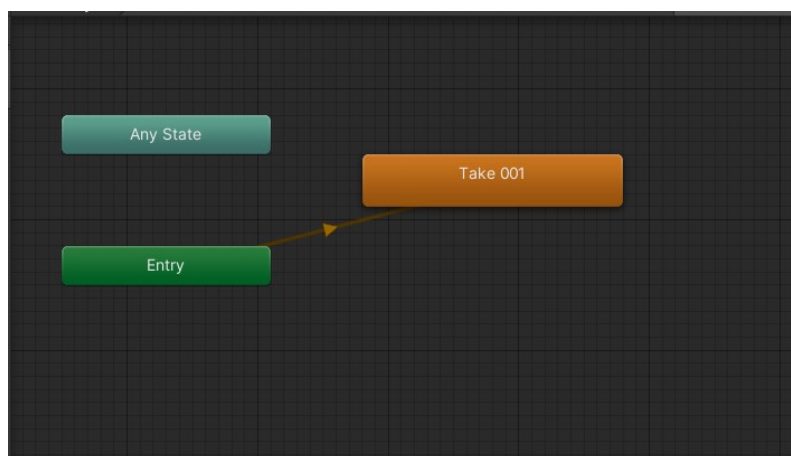


Figura 8.15: Aspecto de la capa Base Layer

Implementación fuente de audio

Uno de los temas a tratar era la forma en la que el usuario iba a tomar un primer contacto con la aplicación. Se trataron varias opciones como incluir texto al estilo de viñetas para darle instrucciones al usuario o añadirle voz al avatar. Finalmente, opté por implementar la segunda opción, ya que consideré que podría ser de gran ayuda para mejorar la experiencia del usuario y poder brindarle una introducción interactiva y amigable.

En cuanto a los propósitos y beneficios de esta funcionalidad se encuentran:

- **Instrucciones claras:** El audio de entrada proporciona instrucciones claras al usuario sobre cómo interactuar con el avatar médico y activar el reconocedor de voz para visualizar los videos educativos. Esto ayuda a orientar al usuario y facilita el proceso de inicio.
- **Experiencia personalizada:** El audio de entrada puede crear una experiencia más personalizada y atractiva para el usuario. Al dar la bienvenida y proporcionar instrucciones personalizadas, el avatar médico genera un ambiente más acogedor y amigable, lo que puede fomentar la confianza del usuario y promover una mayor participación con la aplicación.
- **Mejora de la usabilidad:** La inclusión de audio de entrada reduce la dependencia del usuario en instrucciones escritas o visuales. Esto es especialmente útil para aquellos con dificultades de lectura o visión, así como para mejorar la accesibilidad en general. Al proporcionar instrucciones auditivas, se facilita la comprensión y el uso del sistema para una variedad de usuarios.

Para crear el audio se utilizó la herramienta VEED [100], la cual ofrece una variedad de funciones para la edición de videos, incluyendo la capacidad de crear y personalizar audio a partir de texto. Utiliza tecnología de Inteligencia Artificial (IA) para convertir el texto en voz sintetizada de alta calidad. La IA detrás de esta herramienta utiliza técnicas de síntesis de voz basadas en redes neuronales, como la síntesis de voz basada en concatenación y la síntesis de voz basada en unidades. Estas técnicas permiten generar voces realistas y naturales a partir del texto proporcionado. El proceso de creación de audio en VEED generalmente implica escribir el texto que se desea convertir en voz y luego seleccionar la voz, el acento y otras opciones de personalización disponibles. La IA de VEED procesa el texto y genera el audio correspondiente en función de la configuración elegida.

Una vez que se generó el archivo de audio en formato .mp3, se procedió a agregar el componente Audio Source [98] al avatar médico en Unity. Este componente es una herramienta que permite reproducir y controlar el sonido en la aplicación. Para lograr la reproducción del archivo de audio, se procedió a realizar la configuración del Audio Clip [99], el cual se define como una representación digital del archivo .mp3. Dicho Audio Clip contiene todos los datos y metadatos necesarios para la correcta reproducción del sonido asociado al archivo de audio en cuestión.

Además de asignar el Audio Clip, se realizaron configuraciones adicionales en el componente Audio Source para garantizar una experiencia de audio adecuada. Estas configuraciones incluyeron la especificación de propiedades 3D, que permiten al sonido parecer que proviene de una ubicación espacial en el entorno virtual. Esto crea una sensación de inmersión y realismo, lo que es especialmente útil en aplicaciones de realidad virtual donde la experiencia auditiva contribuye a la sensación de presencia del usuario.

8.6. Iteración 6

Durante la sexta y última iteración, se llevó a cabo la implementación de la realidad virtual en la aplicación, además de ejecutar las pruebas de funcionalidad de la aplicación mediante test cases.

8.6.1. Tareas bloqueadas

Antes de adentrarnos en la explicación detallada de la implementación realizada, es pertinente abordar un issue que no pudo ser incorporado en la aplicación debido a su falta de viabilidad. Este issue, titulado *“Investigar sobre la reproducción de video en streaming para una visualización de alta calidad sin interrupciones ni retrasos en la descarga”*, fue clasificado con la etiqueta POC, lo que indica que se trataba de una exploración inicial para determinar la viabilidad técnica de la funcionalidad propuesta.

El objetivo de dicho issue era investigar y evaluar la posibilidad de implementar una solución que permitiera la reproducción de videos en streaming con una calidad óptima, evitando interrupciones o retrasos durante la descarga. Se realizó un análisis exhaustivo de esta posibilidad, teniendo en cuenta diversos factores y restricciones. Durante el análisis, se identificaron limitaciones relacionadas con la política para desarrolladores de la API de Youtube [101], lo que implicaba ciertas restricciones en cuanto al acceso y la reproducción de los videos en tiempo

real. Además, se evaluaron opciones disponibles para almacenar los vídeos en servidores externos, pero ninguna de las opciones gratuitas disponibles ofrecía una solución fiable y escalable.

A raíz de estas restricciones y limitaciones técnicas, se tomó la decisión, junto con los tutores, de no implementar esta funcionalidad en la aplicación final. En lugar de ello, se mantuvo la solución inicial que permitía a los usuarios acceder a los vídeos de manera local, evitando así los posibles problemas asociados con la reproducción en streaming.

Es importante destacar que la decisión de no incluir esta funcionalidad no tuvo un impacto negativo en la experiencia del usuario, ya que se priorizó la calidad y estabilidad de reproducción de los vídeos a través de la descarga y almacenamiento local.

8.6.2. Implementación de la Realidad Virtual: XR Interaction Toolkit

Una vez implementada y probada la aplicación en Unity 3D, se llevó a cabo la implementación de la realidad virtual en el proyecto. Para ello se siguieron los siguientes pasos:

- Para habilitar la realidad virtual en nuestro proyecto comenzamos instalando el XR Plugin Management [102]. Para ello realizamos lo siguiente: **Edit > Project Settings > XR Plugin Management**. El XR Plugin Management se utiliza para gestionar, configurar y optimizar la integración de plugins de realidad extendida en proyectos de Unity. Proporciona una manera conveniente de administrar los aspectos relacionados con XR y mejorar la experiencia general de la realidad extendida en las aplicaciones y juegos desarrollados con Unity.
- Una vez instalado, establecemos los Plugin Providers que se vayan a utilizar. En nuestro caso, seleccionaremos OpenXR [54], dentro del cual marcaremos Windows Mixed Reality [55], y también seleccionaremos Mock HMD Loader [103].
- Para continuar con el desarrollo del proceso, pasamos a importar un conjunto de herramientas que nos van a permitir interactuar y hacer nuestra aplicación más rápida. Para ello utilizamos el XR Interaction Toolkit. Este paquete proporciona un conjunto de scripts, componentes y prefabs que facilitan la interacción del usuario con la aplicación de realidad virtual [104].
- Una vez importado el XR Interaction Toolkit, procedemos a configurar los diferentes elementos de interacción en nuestra escena de Unity. Esto implica agregar componentes de interacción a los objetos con los que el usuario podrá interactuar, como el avatar médico o el canvas.
- A continuación, procedemos a probar la aplicación en un entorno de realidad virtual. Para esto, conectamos el dispositivo de realidad virtual al equipo de desarrollo y ejecutamos la aplicación en el modo de vista previa de realidad virtual dentro del entorno de Unity. Durante la prueba, verificamos que los elementos de interacción funcionen correctamente, que la visualización sea adecuada y que la experiencia general sea fluida y satisfactoria para el usuario.
- Una vez que la aplicación ha sido probada y se ha verificado su funcionamiento en realidad virtual, se realiza el proceso de compilación. Esto implica generar un archivo ejecutable específico para el dispositivo de realidad virtual objetivo.

XR Origin

El XR Origin [105] es un componente o elemento en Unity que se utiliza en proyectos de realidad extendida para establecer y controlar el punto de origen del seguimiento de movimiento, es decir, determina el punto en el espacio donde se coloca la cámara o el usuario virtual en la escena. Al agregar un XR Origin al proyecto, se puede establecer el modo de origen de seguimiento (Tracking Origin Mode) para definir la referencia espacial utilizada para el seguimiento de movimiento. Por ejemplo, el modo “Floor” establece que el suelo de la escena se convierte en el nivel de referencia para la posición y orientación del usuario virtual. Esto significa que la cámara o el punto de vista del usuario se ajustarán a la altura correspondiente al nivel del suelo en la escena.

Un componente que se debe añadir al XR Origin es el Input Action Manager [106], que se trata de una herramienta en Unity que permite gestionar y controlar las entradas de usuario en el proyecto. En el contexto de la realidad virtual, se utiliza el Input Action Manager con el paquete XRI Default Input Actions para configurar y

asignar acciones a los controladores de realidad virtual. El XRI Default Input Actions [107] es un paquete de Unity que proporciona una configuración predeterminada de acciones comunes en los controladores de realidad virtual. Estas acciones pueden incluir pulsaciones de botones, movimientos de joystick, movimientos de los controladores de mano, entre otros. Al utilizar el Input Action Manager junto con el XRI Default Input Actions, se simplifica la configuración de los controles y las interacciones del usuario en el entorno virtual.

Cuando creamos el XR Origin, nos encontramos con el “Left Controller” y el “Right Controller”. En el contexto de la realidad virtual, los controladores izquierdo y derecho se refieren a los dispositivos de entrada que los usuarios utilizan para interactuar con el entorno virtual. Estos controladores suelen tener una forma similar a los mandos de videojuegos y están diseñados para ser sostenidos en las manos. Los controladores permiten a los usuarios realizar acciones como agarrar y soltar objetos virtuales, moverse en el entorno virtual, seleccionar opciones y realizar gestos específicos. Estos controladores suelen tener botones, joysticks y sensores de movimiento que capturan los movimientos y acciones del usuario para traducirlos en interacciones en el mundo virtual.

XR Interaction Manager

El XR Interaction Manager [108] actúa como un controlador centralizado que administra los eventos y acciones relacionados con la interacción del usuario en un entorno virtual. Permite definir y gestionar diferentes tipos de interacciones, como tocar, agarrar, soltar, pulsar botones, apuntar y otras formas de interacción física o gestual.

Las principales funcionalidades y características del XR Interaction Manager incluyen:

- **Detección de eventos:** El XR Interaction Manager captura y procesa eventos relacionados con la interacción, como el inicio de toques o pulsaciones. Esto facilita la programación de respuestas y comportamientos en función de estos eventos.
- **Interfaz con controladores:** El XR Interaction Manager se integra con los controladores de movimiento y dispositivos de entrada, como mandos de realidad virtual, controladores de realidad aumentada o gestos realizados en dispositivos móviles. Esto permite que las interacciones se realicen de forma natural y se reflejen en el entorno virtual.
- **Flexibilidad y personalización:** Proporciona opciones de configuración y personalización para adaptar las interacciones según las necesidades del proyecto y los dispositivos utilizados.
- **Compatibilidad multiplataforma:** El XR Interaction Manager es compatible con diferentes plataformas y dispositivos de realidad virtual, lo que permite crear experiencias interactivas consistentes y portables en diferentes entornos.

XR Device Simulator

El XR Device Simulator es una herramienta integrada en Unity que permite simular y probar la experiencia de realidad virtual en el editor, sin necesidad de un dispositivo de realidad virtual físico [109].

Al no tener acceso constante a un dispositivo físico de realidad virtual, se utilizó el simulador para probar las interacciones y funcionalidades de la aplicación. Este simulador resultó especialmente útil durante la etapa de desarrollo, ya que me permitió realizar las pruebas y ajustes necesarios.

Implementación de la interacción con el avatar

Para habilitar la interacción con el avatar médico, se añadió el componente XR Simple Interactable [110]. Además, en el apartado de Interactable Events, se configuró la función correspondiente para activar el reconocedor de voz. A continuación, se explica en detalle el proceso para lograr esta interacción:

1. Componente XR Simple Interactable

- Se agregó el componente XR Simple Interactable al avatar médico en Unity. Este componente permite que el objeto sea interactivo y responda a eventos de interacción, como selección o activación.



Figura 8.16: Avatar médico interactuante desde VR

- Una vez añadido, cuando se sitúa el láser de los controladores sobre el avatar, se observa que el color del láser cambia, indicando que se puede interactuar con el objeto. Figura 8.16.

2. Configuración del evento de interacción

- Una vez añadido el componente XR Simple Interactable, nos dirigimos al apartado de Interactable Events en el inspector de Unity.
- Dentro de Interactable Events, se encuentra la opción “Select”. Esta opción se activa cuando el usuario selecciona el objeto interactivo.
- Para habilitar el reconocimiento de voz, dentro de la sección “Select Entered (SelectEnterEventArgs)”, configuramos la función correspondiente (Figura 8.17).

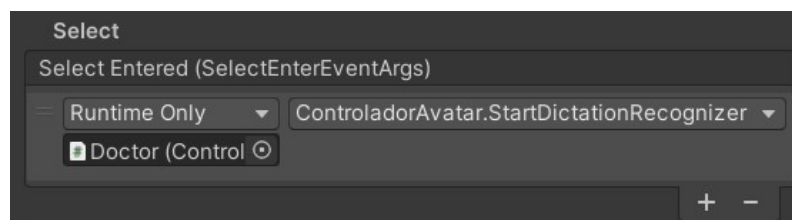


Figura 8.17: Configuración sección Select

3. Activación del reconocedor de voz

- Al configurar la función en el evento de selección, se activará el reconocedor de voz cuando el usuario seleccione al doctor en la escena de realidad virtual.
- Como se puede observar en la Figura 8.18, cuando el usuario haya seleccionado al doctor, se mostrará un marcador sobre el avatar que indica al usuario que el reconocedor está activo.

Implementación de la interacción con el Canvas

Para llevar a cabo la interacción con el canvas se realizó lo siguiente:

1. Componente XR Simple Interactable

- Se añadió a cada imagen disponible en el canvas el componente XR Simple Interactable de manera dinámica.



Figura 8.18: Avatar médico pulsado desde VR

2. Configuración de eventos de Hover

- Se establecieron los eventos de Hover para cada imagen. El evento de Hover se activa cuando el usuario acerca un controlador de realidad virtual a la imagen en cuestión.
- Cuando se produce el evento de Hover, se muestra el título correspondiente al vídeo asociado a la imagen. Esto permite al usuario tener una indicación visual de qué contenido se encuentra vinculado a cada imagen. Ver Figura 8.19.
- Además, el láser del controlador de realidad virtual se muestra de color blanco, lo cual indica que el usuario puede interactuar con las distintas imágenes presentes en el canvas.



Figura 8.19: Vista del canvas desde VR

3. Configuración del evento de selección

- También se configuró el evento de selección para cada imagen del canvas.
- Cuando el usuario selecciona una imagen mediante el controlador de realidad virtual, se activa el evento de selección. Este evento permite reproducir el vídeo asociado a la imagen seleccionada.

4. Reproducción del vídeo seleccionado

- Una vez que el usuario ha seleccionado una imagen específica del canvas, se procede a reproducir el vídeo correspondiente.

- La reproducción del vídeo se controla mediante el uso de teclas asignadas a las diferentes funciones del reproductor de vídeo. Ver Figura 8.20.



Figura 8.20: Reproducción vídeo desde VR

8.6.3. Compatibilidad META

Después de haber integrado exitosamente la funcionalidad de realidad virtual en el proyecto y completado las pruebas necesarias, se decidió configurar la aplicación para que fuera compatible con el metaverso proporcionado por Meta. Sin embargo, aquí surge un inconveniente, ya que no se dispone de equipos de realidad virtual propios de Meta, como Meta Quest o Meta Quest 2. Por lo tanto, se llevará a cabo la configuración correspondiente en Educaverse para garantizar su compatibilidad con Meta, pero no se realizará el despliegue final en los dispositivos de Meta.

Comenzamos configurando el proyecto para soportar Android, lo cual requiere las siguientes dependencias [112]:

- El módulo de soporte de compilación de Android.
- El kit de desarrollo de software de Android (SDK).
- El kit de desarrollo nativo (NDK).
- Un kit de desarrollo de Java. De forma predeterminada, Unity usa OpenJDK.

Para instalar las dependencias anteriores, utilizamos Unity Hub. En nuestro caso, como ya tenemos instalado un Unity Editor, nos dirigimos al apartado de instalaciones. Una vez dentro, pulsamos sobre el icono de ajustes del Editor y seleccionamos la opción *Add modules*. Los módulos a instalar son los que aparecen en la siguiente figura:

PLATFORMS	DOWNLOAD SIZE	SIZE ON DISK
<input checked="" type="checkbox"/> Android Build Support	367.51 MB	1.88 GB
<input checked="" type="checkbox"/> OpenJDK	145.91 MB	67.2 MB
<input checked="" type="checkbox"/> Android SDK & NDK Tools	1.35 GB	4.29 GB

Figura 8.21: Módulos de dependencia

Una vez instalados, debemos proceder a instalar el SDK de integración de Oculus [113], para posteriormente configurar una serie de parámetros para poder ejecutar la aplicación correctamente.

Antes de proceder con cualquier otra configuración del proyecto, hay que establecer la plataforma objetivo para la aplicación, ya que cada plataforma tiene configuraciones únicas. La plataforma objetivo para Oculus Quest y Quest 2 es Android. Para enviar la aplicación a la tienda Oculus, se debe crear un paquete listo para el lanzamiento que los usuarios puedan instalar y ejecutar en sus auriculares Oculus Quest. Dicho paquete contiene el archivo APK, que incluye el código fuente compilado, los recursos, activos, archivo de manifiesto y un archivo de expansión APK (OBB), si está habilitado. Para ello, procedemos a realizar el cambio de plataforma correspondiente. Nos dirigimos a **File > Build Settings** y ahí seleccionamos la plataforma Android. Nos aparecerá una serie de configuraciones como se muestra en la Figura 8.22. Como se puede observar en esa figura, en el apartado Run Device, debería configurarse el dispositivo Meta Quest, en lugar de aparecer *Default device*.

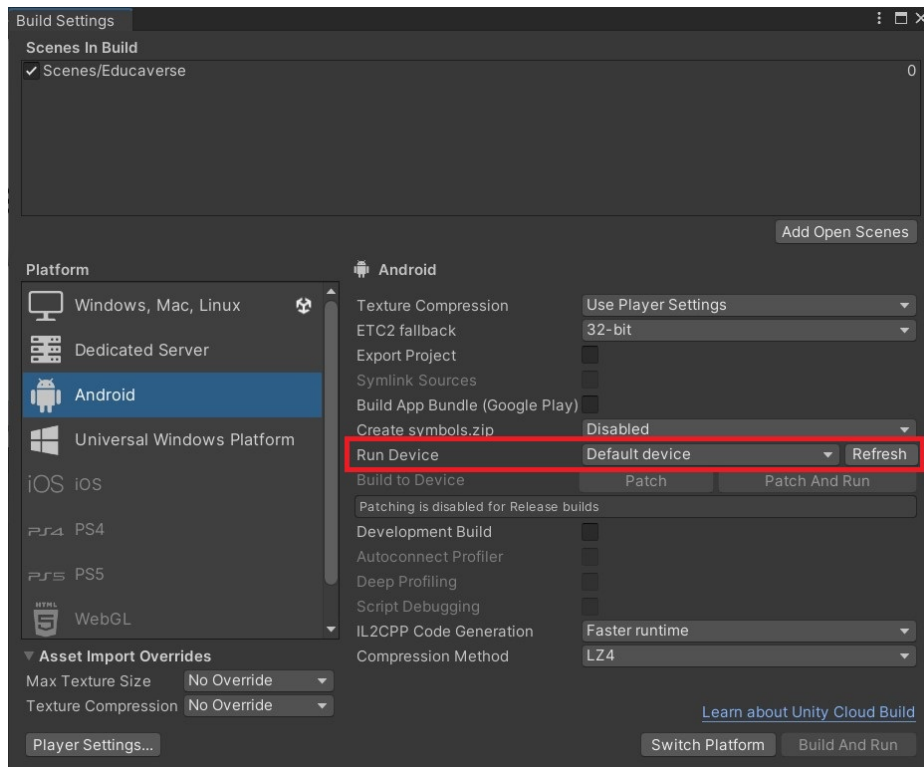


Figura 8.22: Build Settings

Configuración inicial

A continuación, empezamos a configurar los parámetros necesarios según [114]. Para ello, comenzamos definiendo el nombre de la empresa y del producto en el editor de Unity.

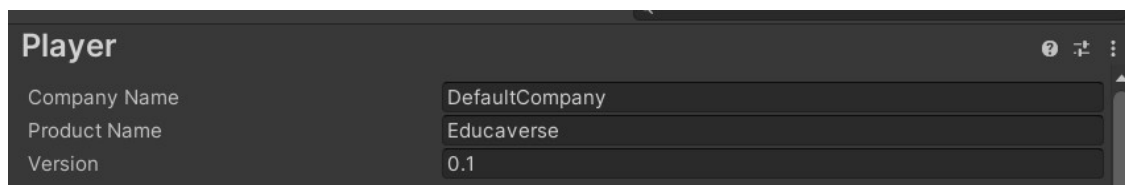


Figura 8.23: Configuración inicial

Identificación del paquete

Continuamos configurando la identificación del paquete. El sistema de compilación utiliza los atributos de identificación del paquete para identificar de manera única la aplicación en el ecosistema de Meta Quest. Ver Figura 8.24.

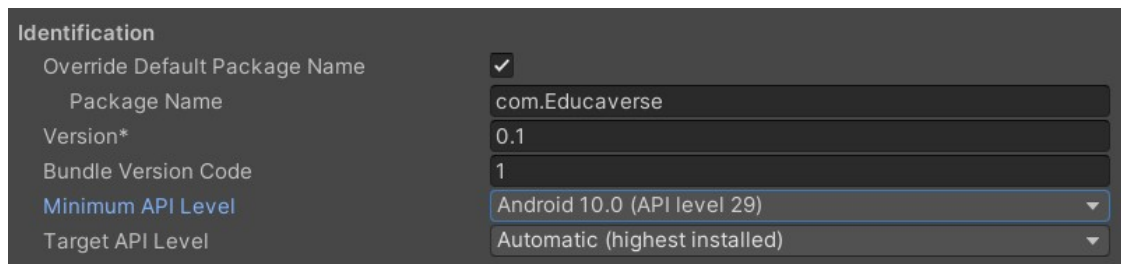


Figura 8.24: Configuración de identificación del paquete

Configuración de la aplicación

Lo siguiente será realizar la configuración necesaria para cumplir con los requisitos de la tienda Meta Quest.

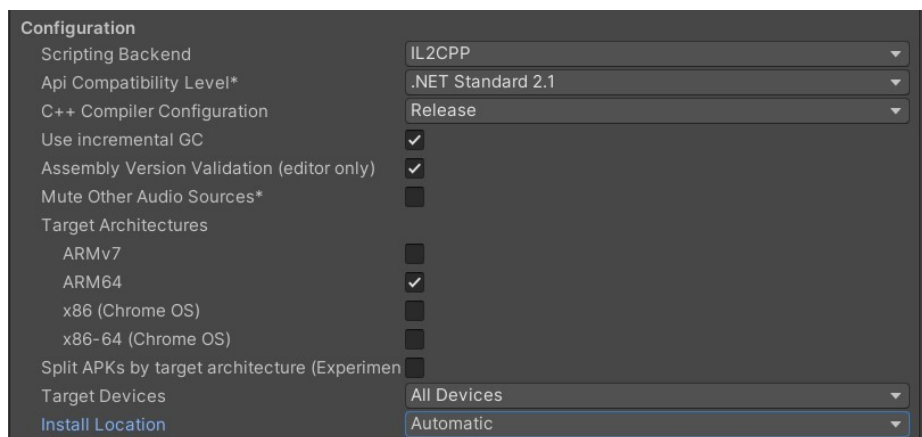


Figura 8.25: Configuración de la aplicación

Gestión de XR

Meta simplifica el soporte de XR mediante el paquete XR Plugin Management de Unity. Para construir aplicaciones inmersivas de Meta Quest, se debe utilizar Oculus XR Plugin.

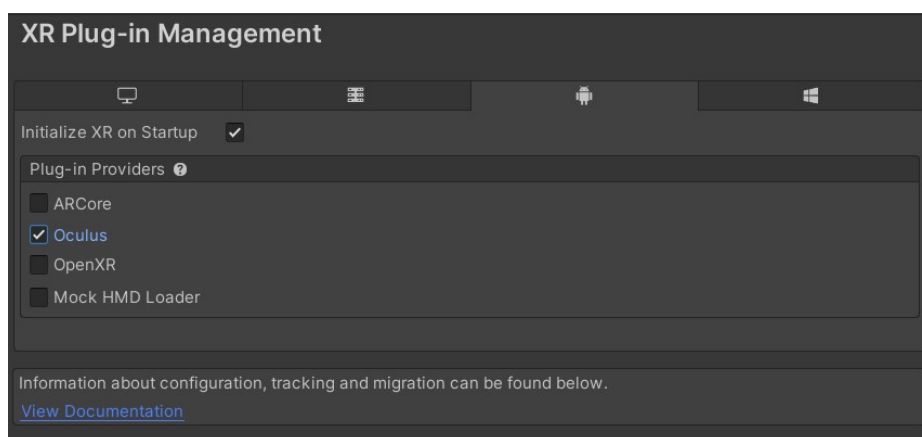


Figura 8.26: Gestión de XR

Configuración de renderización

Existen diversas opciones y configuraciones que permiten optimizar la renderización. Se puede definir un conjunto específico de API gráficas, elegir la propiedad del espacio de color o habilitar la renderización multihilo para optimizar el rendimiento. En la Figura 8.27, aparece el resultado tras realizar las configuraciones necesarias.

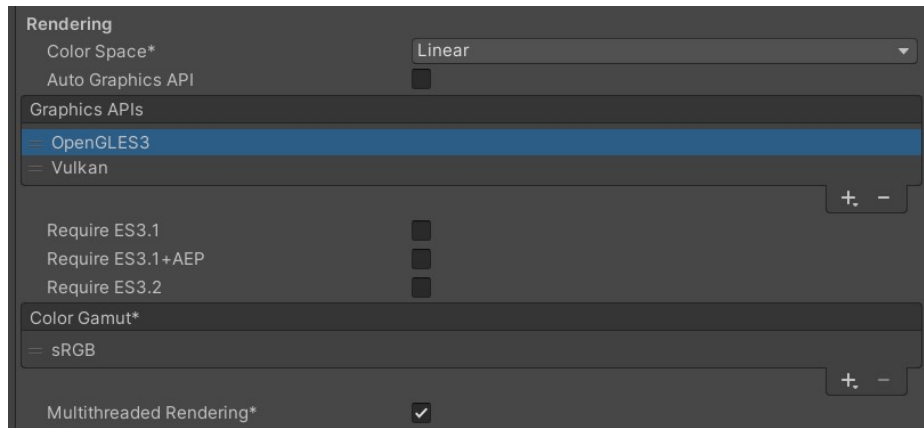


Figura 8.27: Configuración de renderización

Definir la configuración de calidad

Las siguientes configuraciones de calidad nos van a permitir definir la calidad gráfica.

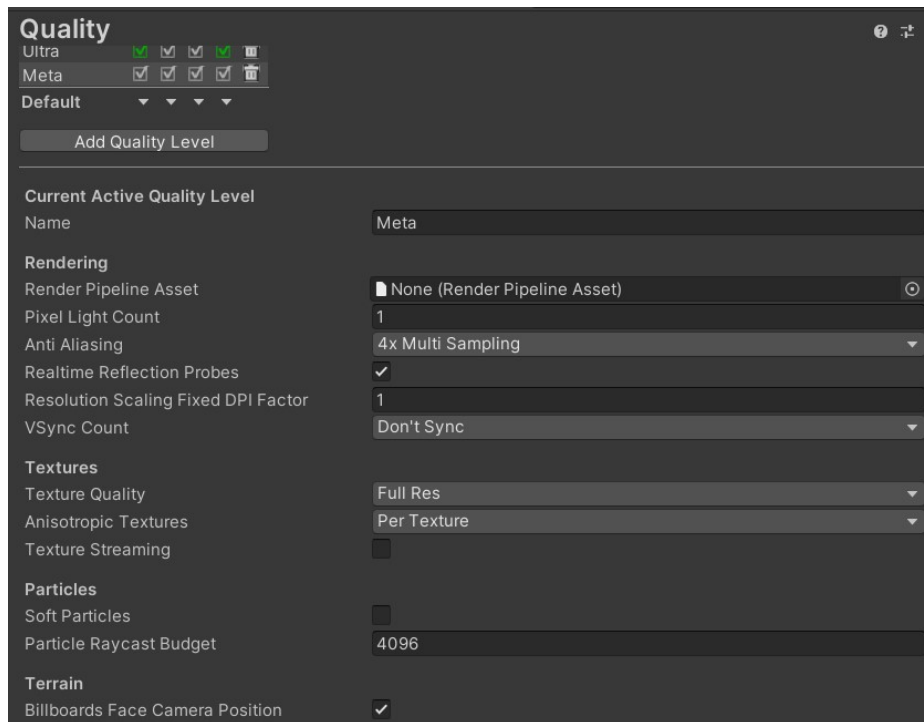


Figura 8.28: Configuración de calidad

Una vez completadas todas las configuraciones necesarias, la aplicación estaría lista para poder ejecutarse en el metaverso de Meta. Lo único que faltaría, sería crear el archivo .apk. Para ello, nos dirigimos a **File > Build And Run** [115].

Capítulo 9

Pruebas y Resultados

En este capítulo, se abordará el proceso de pruebas del software de nuestra aplicación. Se ha decidido utilizar un enfoque de pruebas manuales y seguir una estructura organizada mediante un Test Suite formado por diferentes Test Cases. De esta manera, se garantizará una evaluación exhaustiva de la aplicación y se facilitará la identificación de posibles fallos o mejoras necesarias para lograr un producto final de calidad y una experiencia de usuario satisfactoria. Además de las pruebas técnicas, también se llevaron a cabo pruebas con usuarios reales. Estas pruebas permitieron obtener comentarios y opiniones valiosas sobre la experiencia de uso de la aplicación por parte de los usuarios finales. Al final de este capítulo, se expondrán los resultados obtenidos.

9.1. Test Suite

El Test Suite es una colección de pruebas diseñadas para evaluar el correcto funcionamiento de la aplicación [116]. En este caso, se han desarrollado pruebas manuales para garantizar la calidad y la usabilidad de la interfaz y las funcionalidades implementadas. El Test Suite estará compuesto por varios Test Cases, cada uno de los cuales abordará un aspecto específico de la aplicación.

Estructura del Test Case

Cada Test Case incluirá la siguiente información:

1. **Título del Test Case**

Nombre descriptivo que identifica claramente el objetivo y el enfoque del caso de prueba.

2. **Descripción**

La descripción debe seguir el formato “Cuando [acción], dado [condición], entonces [salida]”. Aquí se debe detallar el escenario específico que se va a probar, incluyendo la acción que se realiza, las condiciones iniciales y la salida esperada.

3. **Precondición**

Condición o condiciones previas necesarias para llevar a cabo el caso de prueba de manera adecuada.

4. **Pasos de prueba**

Pasos específicos que se seguirán durante la ejecución del caso de prueba.

5. **Resultado esperado**

Descripción clara del resultado esperado después de completar el caso de prueba.

6. **Resultado obtenido**

Descripción clara del resultado obtenido después de completar el caso de prueba.

7. Resultado Test

Indicar si el caso de prueba fue aprobado o no.

El objetivo principal de cada Test Case es evaluar una funcionalidad o escenario específico de la aplicación, como la validación de comandos, la reproducción de vídeos, la interacción con la interfaz, etc. Al seguir los pasos descritos en la descripción del Test Case y verificar los resultados obtenidos con los resultados esperados, podremos determinar si la aplicación funciona correctamente y cumple con los requisitos establecidos. Se registrarán los resultados de cada prueba, indicando si ha sido aprobada o no, así como cualquier observación o problema encontrado durante la ejecución.

A continuación, se presentan los casos de prueba que se llevarán a cabo para evaluar la aplicación:

Test Case 01	
Título	Interacción con el avatar médico
Descripción	Cuando el usuario inicie la aplicación , dado que ha pulsado sobre el avatar médico con el controlador de VR, entonces se debe iniciar el reconocedor de voz.
Precondición	No procede.
Pasos de prueba	<ol style="list-style-type: none"> 1. El usuario inicia la aplicación. 2. El usuario escucha el audio de bienvenida y espera a que finalice. 3. El usuario debe situar cualquiera de los dispositivos de VR sobre el avatar médico. 4. El usuario pulsa el avatar médico.
Resultado esperado	Se muestra un marcador encima del avatar indicando que la interacción se ha registrado correctamente y que el reconocedor de voz está listo para comenzar a recibir comandos de voz.
Resultado obtenido	Se muestra un marcador encima del avatar indicando que la interacción se ha registrado correctamente y que el reconocedor de voz está listo para comenzar a recibir comandos de voz.
Resultado test	Aprobado

Tabla 9.1: Test Case 1

Test Case 02	
Título	Validación de comando correcto
Descripción	Cuando el usuario interactúa con el avatar, dado un comando válido, entonces se debe mostrar el catálogo con las imágenes de los vídeos asociados a dicho comando.
Precondición	El usuario debe haber interactuado con el avatar médico.
Pasos de prueba	<ol style="list-style-type: none"> 1. El usuario introduce un comando de voz válido a través del micrófono conectado.
Resultado esperado	Se muestra el catálogo con las imágenes de los vídeos asociados al comando.
Resultado obtenido	Se muestra el catálogo con las imágenes de los vídeos asociados al comando.
Resultado test	Aprobado

Tabla 9.2: Test Case 2

Test Case 03	
Título	Validación de comando incorrecto
Descripción	Cuando el usuario interactúa con el avatar, dado un comando inválido, entonces se debe mostrar al usuario un mensaje adecuado indicando que no hay vídeos disponibles para dicho comando.
Precondición	El usuario debe haber interactuado con el avatar médico.
Pasos de prueba	<ol style="list-style-type: none"> 1. El usuario introduce un comando inválido, es decir, un comando que no coincide con ninguno de los comandos aceptados por la aplicación, a través del micrófono conectado.
Resultado esperado	Se muestra un panel en el que aparece un mensaje indicando al usuario que no se encontraron resultados para el comando introducido.
Resultado obtenido	Se muestra un panel en el que aparece un mensaje indicando al usuario que no se encontraron resultados para el comando introducido.
Resultado test	Aprobado

Tabla 9.3: Test Case 3

Test Case 04	
Título	Visualización de vídeo asociado a una imagen
Descripción	Cuando el usuario selecciona una imagen del catálogo, dado una correcta asociación entre cada imagen y el vídeo correspondiente, entonces se debe reproducir el vídeo.
Precondición	El usuario debe haber introducido un comando válido y el catálogo se debe haber mostrado.
Pasos de prueba	<ol style="list-style-type: none"> 1. El usuario navega por el catálogo, visualizando las diferentes imágenes de los vídeos, mediante el uso de los controladores de VR. 2. El usuario sitúa un controlador sobre una imagen para poder observar el título del vídeo correspondiente. 3. El usuario selecciona una imagen del catálogo.
Resultado esperado	Comienza la reproducción del vídeo que ha sido seleccionado por el usuario.
Resultado obtenido	Comienza la reproducción del vídeo que ha sido seleccionado por el usuario.
Resultado test	Aprobado

Tabla 9.4: Test Case 4

Test Case 05	
Título	Control de reproducción de vídeos
Descripción	Cuando el usuario interactúa con los controles de reproducción de un vídeo, dado que el vídeo se encuentra en reproducción, entonces se espera que los controles respondan correctamente y ejecuten las acciones correspondientes.
Precondición	El usuario debe haber seleccionado una imagen y el vídeo se debe estar reproduciendo.
Pasos de prueba	<ol style="list-style-type: none"> 1. El usuario presiona la tecla de reproducción/pausa del vídeo. 2. El usuario presiona la tecla de adelantar la reproducción del vídeo. 3. El usuario presiona la tecla de retroceder la reproducción del vídeo. 4. El usuario presiona la tecla de detener la reproducción del vídeo.
Resultado esperado	Los controles de reproducción responden correctamente y ejecutan las acciones correspondientes.
Resultado obtenido	Los controles de reproducción responden correctamente y ejecutan las acciones correspondientes.
Resultado test	Aprobado

Tabla 9.5: Test Case 5

Test Case 06	
Título	Cierre del Canvas
Descripción	Cuando el usuario finalice de visualizar los vídeos necesarios e interactúe de nuevo con el avatar, dado un comando válido, entonces se debe ocultar el Canvas y se debe volver al estado inicial de la aplicación.
Precondición	El usuario debe haber introducido un comando válido y el catálogo se debe haber mostrado.
Pasos de prueba	<ol style="list-style-type: none"> 1. El usuario finaliza la visualización y obtiene toda la información necesaria. 2. El usuario introduce un comando válido específico para salir del modo visualización a través del micrófono conectado.
Resultado esperado	Se oculta el Canvas que contiene el catálogo con las imágenes y el usuario vuelve al modo inicial de la aplicación.
Resultado obtenido	Se oculta el Canvas que contiene el catálogo con las imágenes y el usuario vuelve al modo inicial de la aplicación.
Resultado test	Aprobado

Tabla 9.6: Test Case 6

9.2. Pruebas con usuarios

Se realizaron pruebas con usuarios, un método que consiste en evaluar la funcionalidad y la interfaz de la aplicación. El objetivo principal de este proceso fue evaluar la usabilidad de la aplicación, identificar posibles áreas de confusión, descubrir oportunidades de mejora, y en última instancia, determinar si el producto estaba listo para su lanzamiento a usuarios reales.

Estas pruebas son de gran importancia, ya que los desarrolladores a menudo pueden verse influenciados por su conocimiento profundo del producto, lo que puede llevar a pasar por alto problemas de usabilidad. Es especialmente crítico llevar a cabo pruebas con usuarios al desarrollar nuevos productos, como en el caso de Educaverse. Sin estas pruebas, existe el riesgo de quedarse atrapado en un proceso de diseño de experiencia de usuario que el desarrollador comprende, pero que no es adecuado para el público objetivo.

Para llevar a cabo las pruebas, se tuvo la oportunidad de contar con la participación de un paciente real con diabetes que fue diagnosticado durante su infancia. Se elaboró un plan de pruebas detallado que incluyó una evaluación exhaustiva de todas las funcionalidades de la aplicación, desde la interacción con el avatar hasta el control de los vídeos. El objetivo era permitir al usuario probar todas las características y proporcionar retroalimentación valiosa sobre su experiencia.

Se establecieron criterios de éxito y se registraron observaciones detalladas durante las pruebas. La retroalimentación del usuario fue analizada y considerada para garantizar que la aplicación cumpliera con los estándares de usabilidad y satisfacción del usuario antes de su lanzamiento.

Procedimiento

Se utilizó el método de prueba con usuarios moderados a distancia para llevar a cabo la evaluación. Este enfoque de pruebas remotas es una alternativa efectiva a las pruebas presenciales. Con las pruebas a distancia, tanto el moderador como el participante pueden encontrarse en diferentes ubicaciones, siempre y cuando estén viendo el mismo contenido en sus respectivas pantallas.

Dado que el usuario no contaba con un dispositivo de realidad virtual físico, se le proporcionó un ejecutable generado a partir de un simulador de realidad virtual. Se le brindaron instrucciones sobre cómo controlar correctamente el simulador y se logró llevar a cabo la prueba de manera exitosa. Al concluir la prueba, se recopilaron los siguientes resultados:

- Durante las pruebas, se observó que el usuario fue capaz de completar todas las funcionalidades de la aplicación sin encontrar contratiempos significativos. En algunas ocasiones muy puntuales, el usuario tuvo que repetir el comando de voz introducido debido a que la aplicación no lo detectó correctamente en la primera instancia. Sin embargo, al volver a pronunciar el mismo comando con un tono más claro, se logró acceder a los vídeos deseados sin problemas.
- Además, el usuario empleó tanto comandos de voz básicos como comandos más elaborados para acceder a los vídeos de su interés. Es importante destacar que el usuario en cuestión es un paciente real con varios años de experiencia en el manejo de la diabetes. Debido a su amplio conocimiento sobre la enfermedad, utilizar comandos más elaborados no representó un desafío significativo para él. En definitiva, el usuario valoró de manera muy positiva que la aplicación sea capaz de reconocer tanto comandos simples como comandos más complejos, ya que esto permite adaptarse a las necesidades y conocimientos del usuario de manera efectiva.
- Durante las pruebas, el usuario expresó que su debut en la diabetes fue un proceso complicado debido a los numerosos cambios que tuvo que realizar en su rutina diaria y a la necesidad de mantener un contacto constante con profesionales médicos para hacer frente a la enfermedad. Además, señaló que este proceso de adaptación no solo le afectó a él, sino también a sus padres, quienes se mostraron preocupados por los cambios que la enfermedad traería consigo. Ellos también tuvieron que adquirir un amplio conocimiento sobre la diabetes para poder brindarle apoyo adecuado. Sin embargo, acceder a información precisa y confiable sobre la diabetes a menudo resultaba complicado y requería la asistencia de un profesional experto en la materia. En este sentido, el usuario destacó que contar con una herramienta como Educaverse habría sido de gran ayuda tanto para él como para su familia, ya que les habría permitido adquirir conocimientos sólidos sobre la diabetes de manera más accesible y conveniente.
- Otro aspecto relevante que se identificó durante las pruebas fue el acceso constante a la información a través de la amplia selección de vídeos disponibles en Educaverse. Como se mencionó anteriormente, el usuario tuvo que acudir a numerosas citas médicas para obtener información sobre la enfermedad durante su debut. Con Educaverse, el usuario tiene la posibilidad de acceder a los vídeos cuantas veces sea necesario, incluso volver a ver aquellos que le resulten más relevantes, lo que le permite adquirir conocimientos más profundos sobre temas específicos relacionados con la diabetes. Además, destacó que la información proporcionada en la aplicación es impartida por expertos en diabetes, lo que brinda tranquilidad y confianza a los usuarios finales respecto a la veracidad y fiabilidad de la información proporcionada.
- En conclusión, el usuario expresó un alto grado de confianza en el potencial de Educaverse como una aplicación con un futuro prometedor. Basado en su experiencia personal, consideró que la aplicación podría ser de gran ayuda para los jóvenes que se enfrentan al diagnóstico de la diabetes y sus familias. Además, planteó una idea sugerente: en los hospitales que visita regularmente, existen salas de ocio frecuentadas por pacientes jóvenes. Sugirió que si se implementaran equipos de realidad virtual en esas salas específicamente para los

pacientes jóvenes con diabetes, se lograría brindar un apoyo más efectivo y personalizado a estos pacientes en su proceso de tratamiento y adaptación a la enfermedad.

Tras analizar y evaluar los resultados obtenidos, podemos afirmar con confianza que se ha logrado una satisfacción total por parte del usuario, superando incluso las expectativas iniciales. Es importante destacar que, si bien el número de usuarios participantes en las pruebas podría haber sido mayor, la naturaleza de los pacientes finales y las limitaciones de tiempo dificultan la obtención de una muestra más amplia. No obstante, la experiencia y situación vivida por nuestro usuario se alinean perfectamente con el perfil al que se dirige Educaverse.

Estos resultados positivos respaldan la efectividad de Educaverse como una herramienta invaluable para brindar apoyo y conocimiento a personas que se enfrentan al desafío de vivir con diabetes. Si bien es necesario realizar estudios adicionales con una muestra más representativa, los resultados obtenidos hasta ahora nos permiten afirmar que Educaverse tiene el potencial de marcar una diferencia significativa en la vida de los pacientes jóvenes y sus familias.

Capítulo 10

Conclusiones y trabajo futuro

10.1. Conclusiones

El objetivo principal del TFG consistía en crear una aplicación de realidad virtual compatible con el metaverso que permitiera acceder a vídeos educativos sobre la diabetes. Este objetivo se logró completar exitosamente mediante el uso de la plataforma Unity y el lenguaje de programación C#, herramientas que permitieron desarrollar una aplicación funcional y satisfactoria que cumple con los requisitos planteados.

Además, se logró cumplir otro objetivo importante relacionado con la interfaz de usuario en Educaverse. Se implementó una interfaz intuitiva y atractiva que proporciona a los usuarios la capacidad de acceder a los vídeos educativos mediante comandos de voz. Esto mejora la interacción y enriquece la experiencia del usuario al proporcionar un acceso rápido y cómodo a los contenidos educativos.

En lo que respecta a la lógica central de la aplicación, se desarrolló un sistema capaz de identificar y procesar en tiempo real los comandos de voz dictados por el usuario. Se utilizaron API y recursos adecuados para garantizar una respuesta eficiente a los comandos ingresados, ofreciendo un servicio óptimo a los usuarios.

Por otra parte, la planificación y gestión del proyecto presentaron desafíos inesperados. Aunque se establecieron iteraciones iniciales y entregables para cada una, ciertas tareas requirieron más tiempo del previsto, lo que resultó en un retraso en la entrega del TFG dentro del plazo de la primera convocatoria. No obstante, se superaron estos obstáculos y finalmente se entregó el proyecto en el plazo establecido para la segunda convocatoria.

En conclusión, el cumplimiento de los objetivos establecidos demuestran el valor y la efectividad de esta aplicación en el campo de la educación y apoyo para el manejo de la diabetes. A pesar de los desafíos enfrentados durante el desarrollo, el resultado final es satisfactorio y sienta las bases para futuras mejoras y ampliaciones en el ámbito de la realidad virtual y la educación diabética.

Apartado personal

Durante el desarrollo de este proyecto, he tenido la oportunidad de adquirir conocimientos y habilidades sólidas en el campo del desarrollo de aplicaciones 3D utilizando Unity. A lo largo del proceso, he explorado y aplicado diversas técnicas relacionadas con el procesamiento de lenguaje natural, lo que me ha permitido crear una lógica de reconocimiento de voz efectiva para la aplicación.

Además, este proyecto me ha proporcionado una inmersión más profunda en el emocionante mundo de la realidad virtual y el metaverso. He tenido la oportunidad de explorar las posibilidades y limitaciones de estas tecnologías emergentes, y he adquirido habilidades para trabajar con plataformas y dispositivos de realidad virtual. Esta experiencia me ha permitido comprender mejor el potencial de la realidad virtual como herramienta educativa y de entretenimiento, así como su impacto en la interacción y la experiencia del usuario.

Un aspecto que destaco y que ha sido relevante en este proyecto es la conexión con la enfermedad de la diabetes. La elección de este TFG se basó en mi interés personal y en la oportunidad de contribuir de alguna manera al bienestar de las personas que enfrentan esta enfermedad en su vida diaria. A través de la investigación y el desarrollo

de Educaverse, he adquirido conocimientos adicionales sobre la diabetes, sus características y la importancia de la educación y el apoyo en su manejo. Esta experiencia me ha brindado una perspectiva más amplia y una conciencia más profunda de los desafíos que enfrentan las personas con diabetes, y me ha motivado a desarrollar soluciones que puedan beneficiar su calidad de vida.

10.2. Trabajo futuro

Teniendo en cuenta los resultados obtenidos y las lecciones aprendidas durante el desarrollo de este TFG, se plantean diversas líneas de trabajo futuro para mejorar y ampliar la aplicación Educaverse. A continuación, se detallan algunas propuestas:

■ Ampliación del contenido

Una mejora propuesta para Educaverse consiste en enriquecer su biblioteca de vídeos relacionados con la enfermedad de la diabetes. Además, se podrían agregar nuevos comandos para ampliar las funcionalidades de la aplicación y abordar otros aspectos relevantes para el cuidado de la diabetes. Esta expansión de contenido permitiría a los usuarios acceder a información más completa y detallada, fomentando un aprendizaje continuo y enriquecedor.

■ Interacción con los objetos de la sala médica

Consistiría en implementar la funcionalidad de interacción con los objetos de la sala médica. Esto permitiría al usuario diabético explorar y aprender sobre los diferentes elementos y equipos presentes en la sala. El usuario podría moverse libremente por la sala utilizando los controles de realidad virtual y al acercarse a un objeto, se activaría una opción de interacción que proporcionaría información relevante sobre su uso y función. Además, se podrían mostrar descripciones detalladas en pantalla y se podrían incluir animaciones o secuencias interactivas para ilustrar el funcionamiento de los objetos. Además, se podría incluir una sección de evaluación para que el usuario pueda poner a prueba sus conocimientos adquiridos, mediante preguntas interactivas.

■ Añadir minijuego

Por último, se presenta la funcionalidad relacionada con añadir un minijuego en el que el usuario diabético sea el protagonista. El juego se desarrollaría en primera persona, situando al jugador en el escenario de una maratón. Antes de comenzar, el jugador deberá elegir la cantidad de raciones de hidratos de carbono y/o las unidades de insulina que va a ingerir o aplicar. Durante el juego, el jugador avanzará a través de diferentes niveles de la maratón, enfrentándose a obstáculos relacionados con la alimentación y la insulina que afectarán su nivel de glucosa. Deberá esquivar o interactuar con los obstáculos según su nivel de glucosa y las decisiones tomadas anteriormente. Cada categoría de alimento tendrá un efecto específico en el nivel de glucosa y el jugador también podrá aplicarse insulina para reducirlo. El objetivo del juego será mantener la glucosa dentro del rango objetivo y se otorgarán puntos según las acciones y el estado de la glucosa del jugador. La puntuación y los logros obtenidos servirán como indicadores del desempeño del jugador en el control de su glucosa, fomentando así el aprendizaje y la conciencia sobre la diabetes y su manejo adecuado.

Bibliografía

- [1] HP SCDS. [Online]. Available: <https://hpscads.com/> Last Access: 2023-02-23
- [2] Observatorio HP [Online]. Available: <https://hpscads.com/observatorio-tecnologico/observatorio-hp-22-23/> Last Access: 2023-02-23
- [3] A. Cordellat (Ene 10, 2023). La incidencia de la diabetes tipo 2 en el mundo se dispara casi un 60% en tres décadas entre la población adolescente y joven. [Online]. Available: <https://acortar.link/Tc1kGS> Last Access: 2023-05-03
- [4] Samuel (2021). 12 ENFERMEDADES ASOCIADAS A LA DIABETES. [Online]. Available: <https://vivolabs.es/12-enfermedades-asociadas-a-la-diabetes/> Last Access: 2023-02-28
- [5] S. Pursell (Dic 21, 2021). Metodología Agile: qué es y cómo aplicarla a tu proyecto. [Online]. Available: <https://blog.hubspot.es/marketing/metodologia-agile> Last Access: 2023-03-05
- [6] P. Lean (May 7, 2020). ¿Qué es la metodología Agile y por qué está de moda? [Online]. Available: <https://www.progressalean.com/metodologia-agile/> Last Access: 2023-03-05
- [7] J. Bardi (Mar 26, 2019). What Is Virtual Reality: Definitions, Devices, and Examples. [Online]. Available: <https://www.marxentlabs.com/what-is-virtual-reality/> Last Access: 2023-06-17
- [8] PlayStation (2023). PlayStation VR2. [Online]. Available: <https://www.playstation.com/es-es/ps-vr2/> Last Access: 2023-06-17
- [9] Metamandrill (2023). Apple Vision Pro; Everything To Know About Apple's New MR Device!. [Online]. Available: <https://metamandrill.com/apple-vision-pro/> Last Access: 2023-06-20
- [10] Meta (2020). Meta Quest 2. [Online]. Available: <https://www.meta.com/es/quest/products/quest-2/> Last Access: 2023-06-17
- [11] J. González (Feb 26, 2015). Samsung Gear VR, análisis: esperanzador comienzo de la realidad virtual. [Online]. Available: <https://www.xataka.com/analisis/samsung-gear-vr-analisis> Last Access: 2023-06-17
- [12] S. Losey (Oct 9, 2019). Academy brings VR pilot training to cadets. [Online]. Available: <https://www.airforcetimes.com/news/your-air-force/2019/10/08/academy-brings-vr-pilot-training-to-cadets/> Last Access: 2023-03-20
- [13] Metamandrill (2023). Virtual Reality in Healthcare; Clear Examples of VR in Healthcare. [Online]. Available: <https://metamandrill.com/virtual-reality-in-healthcare/> Last Access: 2023-03-20
- [14] A. Haupt (Nov 10, 2022). Virtually Active - RendeeverFit. [Online]. Available: <https://time.com/collection/best-inventions-2022/6227000/rendeeverfit/> Last Access: 2023-03-20
- [15] Metamandrill (2023). Virtual Reality in Entertainment; Examples of VR in Entertainment. [Online]. Available: <https://metamandrill.com/es/realidad-virtual-en-el-entretenimiento/> Last Access: 2023-03-20
- [16] Rick (March 1, 2018). QANTAS VR. [Online]. Available: <https://www.virtualrealitymarketing.com/case-studies/qantas-vr/> Last Access: 2023-03-20
- [17] Metamandrill (2023). Virtual Reality in the Gaming Industry; Biggest Players & Use Case Examples. [Online]. Available: <https://metamandrill.com/virtual-reality-in-the-gaming-industry/> Last Access: 2023-03-20

- [18] S. Mattoo (Sep 6, 2022). Virtual Reality: The Promising Future of Immersive Technology. [Online]. Available: <https://www.g2.com/articles/virtual-reality> Last Access: 2023-04-10
- [19] Iberdrola (2023). Virtual Reality, the technology of the future. [Online]. Available: <https://www.iberdrola.com/innovation/virtual-reality> Last Access: 2023-04-10
- [20] K. Paul (Feb 16, 2022). “Live in the future”: Zuckerberg unveils company overhaul amid shift to metaverse. [Online]. Available: <https://www.theguardian.com/technology/2022/feb/15/meta-mark-zuckerberg-facebook-metaverse> Last Access: 2023-05-08
- [21] E. Javier (Feb 28, 2022). Facebook Realidad Virtual: todo lo que debes saber sobre el Metaverso. [Online]. Available: <https://increnta.com/insights/facebook-realidad-virtual/> Last Access: 2023-05-08
- [22] J. Hurtado (Ago 22, 2022). Qué es el metaverso, ejemplos y cómo se accede. [Online]. Available: <https://www.iieschool.com/blog/el-metaverso-origen-definicion-y-la-apuesta-de-facebook-tecnologia/> Last Access: 2023-05-08
- [23] Udemy Team (Sep, 2021). What is the Unity Game Engine? [Online]. Available: <https://blog.udemy.com/unity-game-engine/> Last Access: 2023-05-09
- [24] F. Jerga (Mar 23, 2021). What Is The Unity Game Engine- All You Need To Know [Online]. Available: <https://medium.com/eincode/what-is-the-unity-game-engine-all-you-need-to-know-d4ce77a1b7d2> Last Access: 2023-05-09
- [25] Unity manual (2022). Unity’s interface. [Online]. Available: <https://docs.unity3d.com/Manual/UsingTheEditor.html> Last Access: 2023-04-12
- [26] Vishnu Sivan (May 30, 2022). Unity3D Fundamentals. [Online]. Available: <https://dev.to/codemaker2015/unity3d-fundamentals-3g9> Last Access: 2023-04-12
- [27] Unity Manual (2016). Mecanim Animation System. [Online]. Available: <https://docs.unity3d.com/462/Documentation/Manual/MecanimAnimationSystem.html> Last Access: 2023-03-20
- [28] Zenva (July 4, 2023). What is Unity? – A Guide for One of the Top Game Engines. [Online]. Available: <https://gamedevacademy.org/what-is-unity/> Last Access: 2023-04-12
- [29] Unity manual (2020). Unity Hub. [Online]. Available: <https://docs.unity3d.com/2020.1/Documentation/Manual/GettingStartedUnityHub.html> Last Access: 2023-04-12
- [30] Unity (2023). Unity Teams. [Online]. Available: <https://unity.com/products/unity-teams> Last Access: 2023-04-12
- [31] Unity Support (2022). What is the Unity Asset Store and how do I purchase Assets? [Online]. Available: <https://support.unity.com/hc/en-us/articles/210142503-What-is-the-Unity-Asset-Store-and-how-do-I-purchase-Assets> Last Access: 2023-05-06
- [32] Microsoft Documentation (April 05, 2023). A tour of the C# language. [Online]. Available: <https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/> Last Access: 2023-04-14
- [33] G. Ganesh (Ene 12, 2022). C# and its Features. [Online]. Available: <https://www.c-sharpcorner.com/article/C-Sharp-and-its-features/> Last Access: 2023-04-14
- [34] J. Boreham (Jul 28, 2022). What is The Facebook Metaverse & How Bright Is Its Future? [Online]. Available: <https://metaverseinsider.tech/2022/07/28/facebook-metaverse/> Last Access: 2023-05-25
- [35] Metamandrill (2023). Facebook Metaverse; Explained, Examples, Devices, Vision & Critics. [Online]. Available: <https://metamandrill.com/facebook-metaverse/> Last Access: 2023-05-25
- [36] Meta (2023). Meta Horizon Home. [Online]. Available: <https://www.meta.com/es-es/help/quest/articles/in-vr-experiences/social-features-and-sharing/meta-horizon-home/> Last Access: 2023-05-25
- [37] Meta (2023). Meta Horizon Worlds. [Online]. Available: <https://www.meta.com/es/horizon-worlds/> Last Access: 2023-05-25

- [38] Meta (Ago 19, 2021). Introducing Horizon Workrooms: Remote Collaboration Reimagined. [Online]. Available: <https://about.fb.com/news/2021/08/introducing-horizon-workrooms-remote-collaboration-reimagined/> Last Access: 2023-05-25
- [39] Meta Quest (Ago 13, 2020). Introducing Horizon Workrooms: Remote Collaboration Reimagined. [Online]. Available: <https://www.oculus.com/experiences/quest/3002729676463989/> Last Access: 2023-05-25
- [40] FITXR Quest (2023). Reality-Changing Fitness. [Online]. Available: <https://fitxr.com/> Last Access: 2023-05-25
- [41] VICTORYXR (2023). THE GLOBAL LEADER IN THE EDUCATION METAVERSE. [Online]. Available: <https://www.victoryxr.com/> Last Access: 2023-05-25
- [42] R. Kumar (Mar 10, 2022). What is Gitlab and How it works? An Overview and Its Use Cases? [Online]. Available: <https://www.devopsschool.com/blog/what-is-gitlab-and-how-it-works-an-overview-and-its-use-cases/> Last Access: 2023-05-07
- [43] I. Gomez (Mar 8, 2022). Conoce qué es GitLab y gestiona efectivamente los cambios que realices en tus proyectos. [Online]. Available: <https://www.crehana.com/blog/transformacion-digital/que-es-gitlab/> Last Access: 2023-05-07
- [44] Astah Professional (2023). Software Design Tool. [Online]. Available: <https://astah.net/products/astah-professional/> Last Access: 2023-05-07
- [45] Astah Professional (2023). Astah User Interface. [Online]. Available: <https://astah.net/support/astah-pro/astah-pro-uml-user-interface/> Last Access: 2023-05-07
- [46] A. Nerja (2023). Qué es overleaf. [Online]. Available: <https://www.adrinerja.com/que-es-overleaf/> Last Access: 2023-05-07
- [47] J. Martins (Sep 12, 2022). Diagrama de Gantt: qué es y cómo crear uno con ejemplos. [Online]. Available: <https://asana.com/es/resources/gantt-chart-basics> Last Access: 2023-03-18
- [48] T. Asana (Ago 15, 2022). Cómo realizar un análisis de riesgos y ejemplos. [Online]. Available: <https://asana.com/es/resources/project-risks> Last Access: 2023-03-23
- [49] “Planificación y Diseño de Sistemas Computacionales”, class notes for 46955, Departamento de Informática, Universidad de Valladolid, Curso 2022-2023.
- [50] talent (2023). Salario en España 2023. [Online]. Available: <https://es.talent.com/es/salary> Last Access: 2023-03-23
- [51] Sociedad Castellano-Leonesa de Endocrinología, Diabetes y Nutrición. [Online]. Available: <https://sclledyn.org/> Last Access: 2023-06-01
- [52] Unity (March 29, 2023). Unity 2021.3.22. [Online]. Available: <https://unity.com/releases/editor/whats-new/2021.3.22> Last Access: 2023-03-24
- [53] Unity manual (2022). XR architecture. [Online]. Available: <https://docs.unity3d.com/Manual/XRPluginArchitecture.html> Last Access: 2023-06-20
- [54] Microsoft Documentation (Mar 21, 2023). OpenXR. Available: <https://learn.microsoft.com/es-es/windows/mixed-reality/develop/native/openxr> Last Access: 2023-07-01
- [55] Microsoft Documentation (Abr 5, 2022). Unity development for VR and Windows Mixed Reality. Available: <https://learn.microsoft.com/en-us/windows/mixed-reality/develop/unity/unity-development-wmr-overview> Last Access: 2023-06-20
- [56] Microsoft Documentation (Jun 05, 2021). Voice input in Unity. [Online]. Available: <https://learn.microsoft.com/en-us/windows/mixed-reality/develop/unity/voice-input-in-unity> Last Access: 2023-06-10
- [57] Meta Quest (Ene 14, 2023). Oculus Integration SDK. [Online]. Available: <https://developer.oculus.com/downloads/package/unity-integration/> Last Access: 2023-07-03

- [58] M. Durán (Abr 11, 2023). Qué es la arquitectura en capas, ventajas y ejemplos. [Online]. Available: <https://blog.hubspot.es/website/que-es-arquitectura-en-capas> Last Access: 2023-04-10
- [59] REFACTORING GURU. PATRONES de DISEÑO. [Online]. Available: <https://refactoring.guru/es/design-patterns> Last Access: 2023-04-13
- [60] M. Alvarez (Jul 28, 2020). Qué es MVC. [Online]. Available: <https://desarrolloweb.com/articulos/que-es-mvc.html> Last Access: 2023-04-13
- [61] REFACTORING GURU. Observer. [Online]. Available: <https://refactoring.guru/es/design-patterns/observer> Last Access: 2023-04-13
- [62] REFACTORING GURU. State. [Online]. Available: <https://refactoring.guru/es/design-patterns/state> Last Access: 2023-04-13
- [63] REFACTORING GURU. Singleton. [Online]. Available: <https://refactoring.guru/es/design-patterns/singleton> Last Access: 2023-04-13
- [64] UML Component Diagrams. [Online]. Available: <https://www.uml-diagrams.org/component-diagrams.html> Last Access: 2023-06-08
- [65] Visual Paradigm (2023). What is Deployment Diagram?. [Online]. Available: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-deployment-diagram/> Last Access: 2023-06-10
- [66] P. Segui (2022). Interfaz de usuario: Qué es y cómo diseñar una UI. [Online]. Available: <https://ovacen.com/disenio-interfaz-usuario-ui/> Last Access: 2023-04-20
- [67] 3D Everything (Oct 21, 2021). Hospital Doctor's Office. [Online]. Available: <https://assetstore.unity.com/packages/3d/props/interior/hospital-doctor-s-office-65226> Last Access: 2023-03-14
- [68] BrushDip (Mar 13, 2023). Doctor - Sketchfab. [Online]. Available: <https://sketchfab.com/3d-models/doctor-sketchfab-weekly-13-mar23-9c89a438a5e940e59a0f9a07c22d6ade> Last Access: 2023-05-05
- [69] GitLab (2023). Issues. [Online]. Available: <https://docs.gitlab.com/ee/user/project/issues/> Last Access: 2023-03-21
- [70] M. Rehkopf (2023). User stories with examples and a template. [Online]. Available: <https://www.atlassian.com/agile/project-management/user-stories> Last Access: 2023-03-21
- [71] Equipo Redacción apser (Nov 15, 2018). PoC o Prueba de Concepto: qué es y cuándo usarla. [Online]. Available: <https://apser.es/poc-o-prueba-de-concepto-que-es-y-cuando-usarla/> Last Access: 2023-03-22
- [72] A. Onuta (Jul 14, 2019). How Agile Spikes help to Improve your Agile Product Delivery? [Online]. Available: <https://ancaonuta.medium.com/how-spikes-help-to-improve-your-agile-product-delivery-a0f104305911> Last Access: 2023-03-22
- [73] Descargar Unity Hub [Online]. Available: <https://unity.com/es/download> Last Access: 2022-11-25
- [74] K. Silva (Oct 11, 2017). Unity Code Snippets. [Online]. Available: <https://ancaonuta.medium.com/how-spikes-help-to-improve-your-agile-product-delivery-a0f104305911> Last Access: 2023-02-27
- [75] Unity Scripting API (2022). KeywordRecognizer [Online]. Available: <https://docs.unity3d.com/ScriptReference/Windows.Speech.KeywordRecognizer.html> Last Access: 2023-04-16
- [76] Unity Scripting API (2022). GrammarRecognizer. [Online]. Available: <https://docs.unity3d.com/ScriptReference/Windows.Speech.GrammarRecognizer.html> Last Access: 2023-03-27
- [77] Unity Scripting API (2022). DictationRecognizer. [Online]. Available: <https://docs.unity3d.com/ScriptReference/Windows.Speech.DictationRecognizer.html> Last Access: 2023-05-31
- [78] Unity Scripting API (2022). PhraseRecognition. [Online]. Available: <https://docs.unity3d.com/ScriptReference/Windows.Speech.PhraseRecognitionSystem.html> Last Access: 2023-03-27
- [79] Unity Scripting API (2022). PhraseRecognizer. [Online]. Available: <https://docs.unity3d.com/ScriptReference/Windows.Speech.PhraseRecognizer.html> Last Access: 2023-03-27

- [80] Unity Scripting API (2022). MonoBehaviour. [Online]. Available: <https://docs.unity3d.com/ScriptReference/MonoBehaviour.html> Last Access: 2023-04-13
- [81] Unity Scripting API (2022). Transform. [Online]. Available: <https://docs.unity3d.com/ScriptReference/Transform.Translate.html> Last Access: 2023-04-13
- [82] Sketchfab. The leading platform for 3D & AR on the web. [Online]. Available: <https://sketchfab.com/> Last Access: 2023-05-05
- [83] A. Chatelain (Sep 9, 2021). Sketchfab for Unity 1.2.1 [Online]. Available: <https://github.com/sketchfab/unity-plugin/releases/tag/1.2.1> Last Access: 2023-05-05
- [84] N. Barney (2023). Named entity recognition (NER). Available: <https://www.techtarget.com/whatis/definition/named-entity-recognition-NER> Last Access: 2023-06-14
- [85] pawangfg (Oct 18, 2022). Named Entity Recognition. Available: <https://www.geeksforgeeks.org/named-entity-recognition/> Last Access: 2023-06-14
- [86] M. Mayo (May 16, 2022). Natural Language Processing Key Terms, Explained. Available: <https://www.kdnuggets.com/2017/02/natural-language-processing-key-terms-explained.html> Last Access: 2023-06-14
- [87] Unity Scripting API (2022). VideoClip. Available: <https://docs.unity3d.com/es/2019.4/ScriptReference/Video.VideoClip.html> Last Access: 2023-05-20
- [88] Unity Scripting API (2018). UI (Interfaz de Usuario). Available: <https://docs.unity3d.com/es/2018.4/Manual/UISystem.html> Last Access: 2023-05-20
- [89] Unity manual (2022). Video Player component. [Online]. Available: <https://docs.unity3d.com/Manual/class-VideoPlayer.html> Last Access: 2023-05-20
- [90] Unity Scripting API (2022). Render Texture. Available: <https://docs.unity3d.com/Manual/class-RenderTexture.html> Last Access: 2023-05-20
- [91] Unity manual (2019). Canvas. Available: <https://docs.unity3d.com/es/2019.4/Manual/UICanvas.html> Last Access: 2023-05-27
- [92] Unity Scripting API (2019). GridLayoutGroup. Available: <https://docs.unity3d.com/2019.1/Documentation/ScriptReference/UI.GridLayoutGroup.html> Last Access: 2023-05-27
- [93] Unity manual (2018). Raw Image. Available: <https://docs.unity3d.com/es/2018.4/Manual/script-RawImage.html> Last Access: 2023-05-27
- [94] Unity manual (2018). Box Collider. Available: <https://docs.unity3d.com/es/2018.4/Manual/class-BoxCollider.html> Last Access: 2023-05-27
- [95] Unity manual (2022). Keyboard events. Available: <https://docs.unity3d.com/Manual/UIE-KeyBoard-Events.html> Last Access: 2023-05-30
- [96] Unity Scripting API (2022). AnimationClip. Available: <https://docs.unity3d.com/ScriptReference/AnimationClip.html> Last Access: 2023-05-30
- [97] Unity manual (2020). Animator Component. Available: <https://docs.unity3d.com/es/530/Manual/class-Animator.html> Last Access: 2023-06-01
- [98] Unity manual (2018). Audio Source (Fuente de Audio). Available: <https://docs.unity3d.com/es/2018.4/Manual/class-AudioSource.html> Last Access: 2023-06-05
- [99] Unity manual (2018). Audio Clip. Available: <https://docs.unity3d.com/es/2018.4/Manual/class-AudioClip.html> Last Access: 2023-06-05
- [100] VEED.IO. [Online]. Available: <https://www.veed.io/> Last Access: 2023-06-05
- [101] Youtube (2023). Servicios de la API de YouTube: Políticas para desarrolladores. [Online]. Available: <https://developers.google.com/youtube/terms/developer-policies?hl=es-419> Last Access: 2023-06-05

- [102] Unity manual (2022). XR Plugin Management. Available: <https://docs.unity3d.com/Manual/com.unity.xr.management.html> Last Access: 2023-06-20
- [103] Unity manual (2020). About the Mock HMD XR Plugin. Available: <https://docs.unity3d.com/Packages/com.unity.xr.mock-hmd@1.0/manual/index.html> Last Access: 2023-06-20
- [104] Unity manual (2023). XR Interaction Toolkit. Available: <https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@2.3/manual/index.html> Last Access: 2023-06-23
- [105] Unity manual (2022). XR Origin. Available: <https://docs.unity3d.com/Manual/xr-origin.html> Last Access: 2023-06-23
- [106] Unity manual (2022). Input Action Manager. Available: <https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@2.0/manual/input-action-manager.html> Last Access: 2023-06-23
- [107] Unity manual (2020). Samples. Available: <https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@1.0/manual/samples.html> Last Access: 2023-06-23
- [108] Unity manual (2022). XR Interaction Manager. Available: <https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@2.0/manual/xr-interaction-manager.html> Last Access: 2023-06-26
- [109] Unity manual (2022). XR Device Simulator. Available: <https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@2.0/manual/xr-device-simulator.html> Last Access: 2023-06-23
- [110] Unity manual (2022). XR Simple Interactable. Available: <https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@2.0/manual/xr-simple-interactable.html> Last Access: 2023-06-29
- [111] L. Santamaría (Nov 15, 2022). Tu primera aplicación en Meta Quest 2 usando Unity. Available: <https://www.furiouskoalas.com/primer-a-aplicacion-meta-quest-2-usando-unity/> Last Access: 2023-07-01
- [112] Unity manual (2022). Android environment setup. Available: <https://docs.unity3d.com/Manual/android-sdksetup.html> Last Access: 2023-07-01
- [113] Meta Quest (Jun 14, 2023). Oculus Integration SDK. Available: <https://developer.oculus.com/downloads/package/unity-integration/> Last Access: 2023-07-01
- [114] Meta Quest (2022). Configure Unity Settings. Available: <https://developer.oculus.com/documentation/unity/unity-conf-settings/> Last Access: 2023-07-3
- [115] Unity manual (2021). Build Settings. Available: <https://docs.unity.cn/2021.1/Documentation/Manual/BuildSettings.html> Last Access: 2023-07-03
- [116] Kitakabee (Dic 21, 2022). What is a Test Suite & Test Case? (with Examples). Available: <https://www.browserstack.com/guide/what-is-test-suite-and-test-case> Last Access: 2023-06-30

Apéndice A

Manual de despliegue

A.1. Despliegue del proyecto en Unity

A.1.1. Requisitos del sistema

- Versión: Windows 10.
- Unity Hub 3.4.1 o superior.
- Unity Editor 2021.3.22f1 o superior.
- Conexión a Internet estable.
- Disponer de un micrófono o dispositivo de entrada de voz.

A.1.2. Proceso de despliegue

1. Clonación del repositorio

Para comenzar, se realiza la clonación del repositorio que contiene el código fuente de la aplicación. Esto se realiza mediante el uso de un IDE, como Visual Studio Code. Desde el entorno del IDE, se abre una nueva terminal y se ejecuta el siguiente comando:

- `git clone https://gitlab.com/HP-SCDS/Observatorio/2022-2023/educaverse/uva-educaverse.git`

2. Apertura del proyecto en Unity Hub

Una vez clonado el proyecto, se accede a Unity Hub y se selecciona la opción **Projects**. Desde esta ventana, procedemos a buscar el proyecto a través del botón **Open**. Una vez encontrado el proyecto, se selecciona y se espera a que el Editor de Unity se abra.

3. Verificación del micrófono

Antes de ejecutar el proyecto, es necesario verificar que el micrófono esté activado. Para hacerlo, se siguen los siguientes pasos:

- Seleccionar **Edit > Project Settings > Player**.
- Seleccionar la pestaña **Universal Windows Platform Settings**.
- Dentro de **Publish Settings**, buscar la sección llamada **Capabilities** y verificar que la opción del micrófono esté activada.

4. Configuración del dispositivo de realidad virtual

Una vez se ha verificado que el micrófono está habilitado según los pasos anteriores, se procede a configurar el dispositivo de realidad virtual. En caso de no disponer de un dispositivo físico, se puede crear un ejecutable de la aplicación que se controlará mediante un simulador de realidad virtual. Para configurar las gafas de realidad virtual, se siguen los siguientes pasos:

- Seleccionar **Edit > Project Settings > XR Plug-in Management**
- Desplegar esta opción para mostrar los Plug-in Providers configurados en la aplicación.
- Seleccionar la opción **OpenXR** y añadir las gafas de realidad virtual a través de la sección **Interaction Profiles**. Para ellos, se selecciona el símbolo “+” que aparece en la esquina inferior derecha y se busca el dispositivo correspondiente.
- Una vez añadido a la lista, se debe asegurar que esté posicionado en la parte superior de la lista para garantizar que la aplicación lo detecte correctamente al ejecutarse.

5. Creación del ejecutable y prueba

Una vez realizadas las configuraciones necesarias, podemos crear un ejecutable y probarlo desde nuestras gafas. Para ello nos dirigimos a **File > Build Settings > Build And Run**. De esta manera se creará el ejecutable y se empezará a ejecutar en nuestras gafas de realidad virtual.

Apéndice B

Manual de usuario

Este manual proporciona instrucciones detalladas sobre cómo utilizar la aplicación, incluyendo la navegación por la interfaz de usuario, la activación del reconocimiento de voz y la visualización de los vídeos didácticos.

B.1. Interfaz de usuario

Educaverse es una aplicación que permite disfrutar de una simulación médica con el fin de aprender sobre la diabetes a través de una serie de vídeos didácticos. Para conseguir una experiencia lo más realista e inmersiva posible, la interfaz de usuario de Educaverse está compuesta por los siguientes elementos:

- **Sala médica**

Educaverse cuenta con una sala médica virtual que ha sido diseñada cuidadosamente para proporcionar una experiencia visualmente auténtica y detallada. Dentro de esta sala, el usuario podrá observar y explorar diversos elementos propios de una consulta médica. (Ver Figura B.1).

- **Avatar médico:** Dentro de la sala médica, se encuentra un avatar médico que cumple la función de guía en Educaverse. El avatar da la bienvenida al usuario y proporciona instrucciones sobre cómo interactuar con él para acceder a los vídeos didácticos de la aplicación. (Ver Figura B.2).

Cuando el usuario interactúa con el avatar, se muestra un marcador indicando que está listo para recibir los comandos de voz del usuario. (Ver Figura B.3).

- **Catálogo de vídeos:** Cuando el usuario pronuncia un comando válido, se muestra un catálogo de vídeos relacionados con el comando introducido. El usuario puede explorar los distintos vídeos disponibles y reproducir el que desee. (Ver Figura B.4).
- **Vídeos educativos:** Los vídeos educativos son el componente central de Educaverse. Impartidos por enfermeros y expertos en diabetes, estos vídeos brindan información precisa y completa sobre la enfermedad. Los vídeos cubren todos los aspectos relevantes de la diabetes y ofrecen al usuario los conocimientos necesarios para comprenderla y manejarla adecuadamente. (Ver Figura B.5).

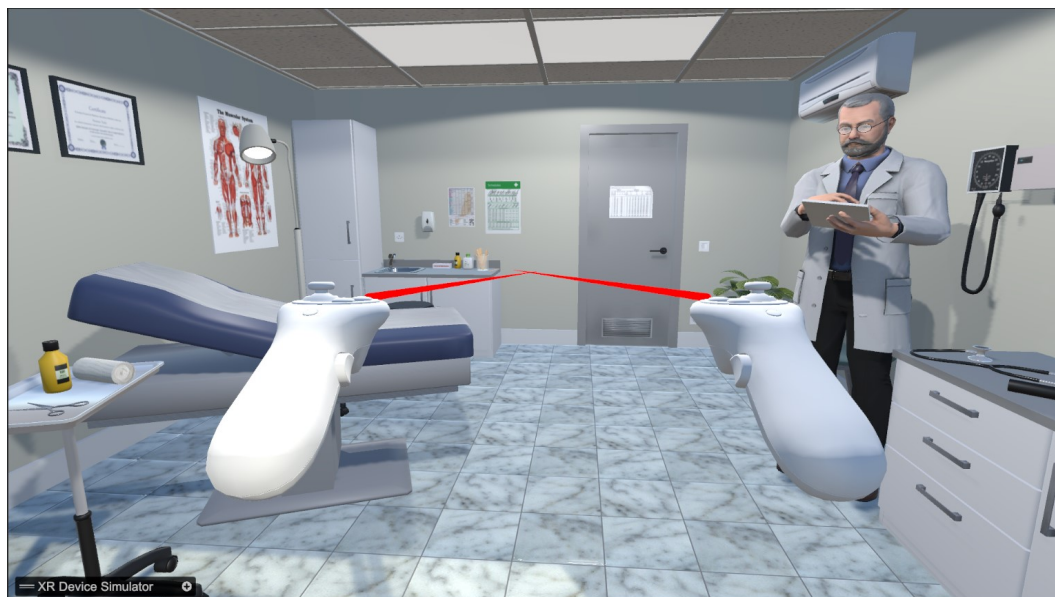


Figura B.1: Sala médica de Educaverse



Figura B.2: Avatar médico de Educaverse



Figura B.3: Avatar médico pulsado

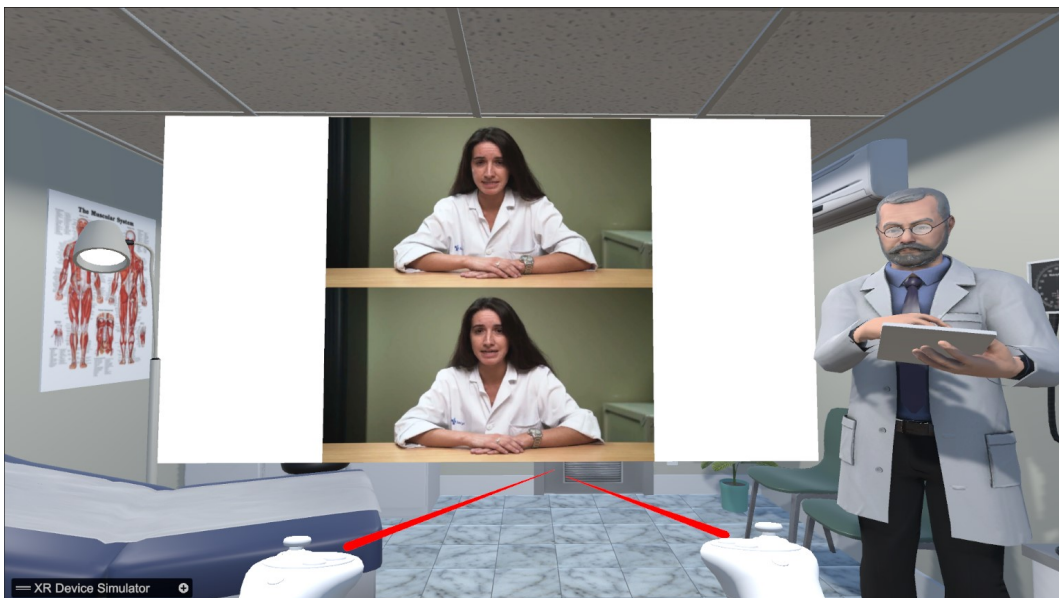


Figura B.4: Catálogo de vídeos de Educaverse

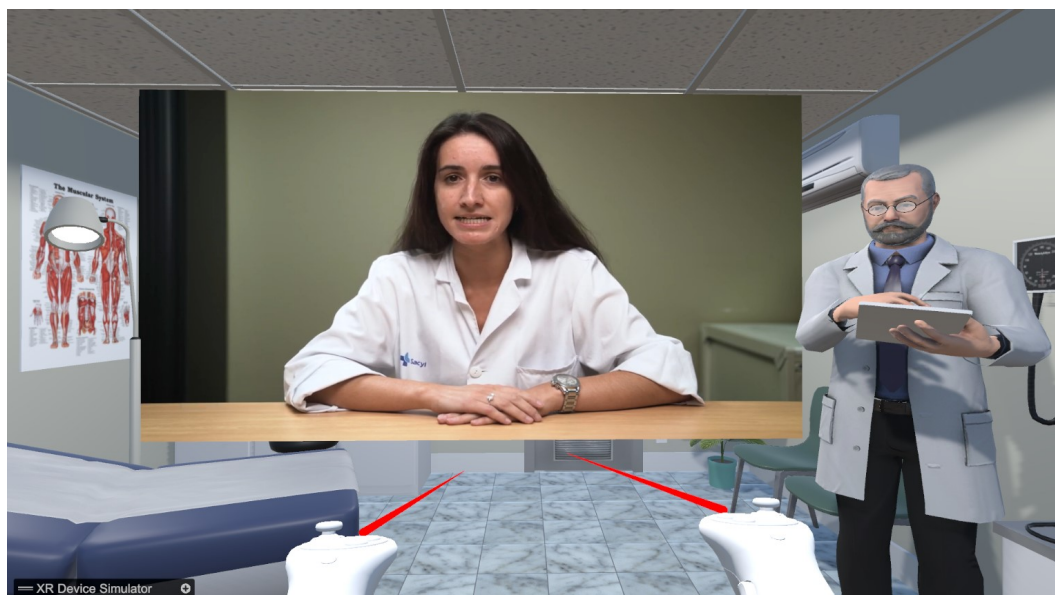


Figura B.5: Vídeo educativo de Educaverse

B.2. Funcionalidades principales

A continuación, se describen las principales características y funcionalidades de la aplicación, detallando cómo utilizar cada una de ellas paso a paso. Se incluyen ejemplos para una mejor comprensión.

■ Reconocimiento mediante comandos de voz

Educaverse cuenta con un preciso reconocedor de voz que permite visualizar los vídeos educativos con los que cuenta la aplicación. Dicho reconocedor será activado en el momento en el que el avatar médico sea pulsado. A partir de ese momento, el usuario deberá dictar comandos con un tono claro y alto para que puedan ser reconocidos de forma precisa. Si el comando que ha introducido es válido, se procede a mostrar el catálogo con los vídeos correspondiente. Sin embargo, si el comando no es válido, se mostrará un mensaje indicando que no hay resultados para dicho comando. En la sección B.3, se muestran unos ejemplos de comandos válidos.

■ Explorar catálogo

Como se ha explicado, si el usuario introdujo un comando válido, entonces se mostrará un catálogo de vídeos. En este momento, el usuario podrá navegar por el catálogo para ver los vídeos que hay disponibles. Para ello, cuando se sitúe sobre un vídeo, aparecerá una ventana emergente en la que se indicará el título del vídeo en cuestión.

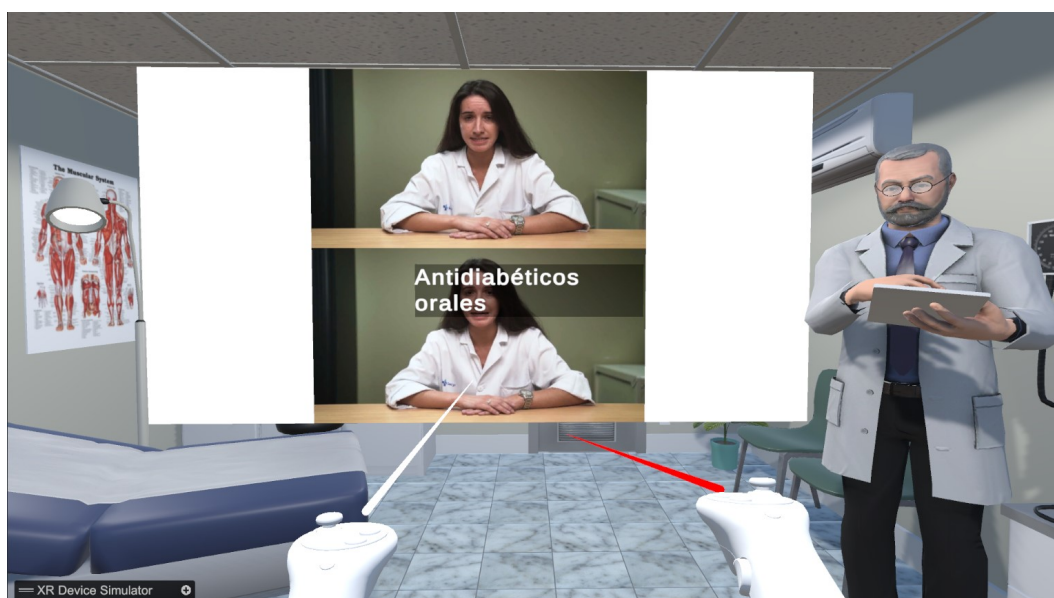


Figura B.6: Ventana emergente con el título del vídeo

■ Visualizar vídeos educativos

Cuando el usuario seleccione un vídeo, este comenzará a reproducirse. En ese momento, el usuario podrá utilizar los controles de reproducción del vídeo para pausar, reanudar, adelantar, retroceder o detener y quitar el vídeo. Todos estos controles se realizarán mediante las siguientes teclas:

- **Espacio** - Pausar/Reanudar vídeo.
- **Flecha Derecha** - Adelantar vídeo.
- **Flecha Izquierda** - Retroceder vídeo.
- **Escape** - Detener y quitar vídeo.

B.3. Comandos de voz

Educaverse ofrece una amplia variedad de comandos de voz que facilitan la navegación y el control de la aplicación. Mediante estos comandos, los usuarios serán capaces de poder visualizar los vídeos educativos según sus necesidades.

Los comandos de voz están relacionados con las categorías de los vídeos de la aplicación. Estas categorías se ofrecen en el apartado B.4.

Educaverse, cuenta con una implementación que permite detectar comandos de voz más básicos y comandos de voz más elaborados. Por ejemplo, el usuario puede dictar comandos básicos relacionados directamente con las categorías como pueden ser los siguientes ejemplos:

- “¿Qué tipos de diabetes hay?”
- “Información sobre el funcionamiento y uso del glucómetro”
- “Medicamentos disponibles para el tratamiento de la diabetes”

Pero también puede pronunciar comandos de voz más elaborados, relacionados con la información específica que se muestra en los distintos vídeos. Ejemplos de estos comandos podrían ser:

- “¿Cómo debo llevar a cabo la colocación del sensor de glucosa?”
- “Información sobre los inyectables orales”
- “¿En qué consiste la diabetes mellitus tipo 2?”

B.4. Ayuda

A continuación se muestran las categorías de los vídeos con los que cuenta Educaverse:

- **Tipos Diabetes**
- **Tipos Insulina**
- **Técnicas Insulina**
- **Administración Insulina**
- **Bomba Insulina**
- **Medicamentos Diabetes**
- **Glucómetro**
- **Hipoglucemia**
- **Cuidado Personal**
- **Nutrición**
- **Actividad Física**

¡Disfrute de su experiencia de aprendizaje con Educaverse!