

Grado Universitario en Ingeniería Informática
2021-2022

Trabajo Fin de Grado

“Aplicación inmersiva para definición de puntos de interés sobre mapas”

Miguel García Trigueros

Tutor/es

Alejandro Rey López

Leganés, Madrid

2022



[Incluir en el caso del interés de su publicación en el archivo abierto]

Esta obra se encuentra sujeta a la licencia Creative Commons **Reconocimiento – No Comercial – Sin Obra Derivada**

RESUMEN

La Realidad Virtual genera entornos con objetos y de apariencia real, haciendo que el usuario se sienta inmerso en un mundo virtual abstrayéndose de la realidad. Gracias a dispositivos como lo son las gafas de Realidad Virtual y sus mandos, el usuario puede moverse dentro del mundo virtual e incluso hacerlo en el real, pues la aplicación responderá a esa interacción en tiempo real.

Este trabajo en concreto se centra en el desarrollo de una aplicación de Realidad Virtual en la que ser capaz de explorar mapas en 3D. El sistema contará con distintos tipos de interacciones a través de las cuales realizar las acciones disponibles en el entorno, exploración del terreno, leer información de puntos de interés, generar marcadores a lo largo del mapa, disposición de rutas entre dichos marcadores o incluso viajes a otras ciudades pudiendo ver el terreno desde distintos puntos. Estas interacciones pueden variar entre los mandos disponibles en las Oculus Quest 2 (Oculus Touch), reconocimiento de gestos manuales o seguimiento de sensores adicionales distribuidos por el cuerpo y objetos. Cada uno de ellos dando al usuario cada vez más sensación de presencia e inmersión en el entorno virtual.

Para llevar a cabo el proyecto, se realiza un previo estudio de las posibilidades que ofrece una tecnología tan innovadora como lo es la Realidad Virtual y del empleo de los programas (Unity, Mapbox, etc.) y técnicas necesarias para que ese desarrollo sea lo más óptimo posible y la aplicación resulte en lo que se ha planeado.

ABSTRACT

Virtual Reality creates surroundings with elements that seem real, giving the user the impression that they are fully involved in a world that is abstracted from reality. The user can move within the virtual environment and even do so in the real world thanks to gadgets like Virtual Reality glasses and their controls because the program will react to that input in real time.

The development of a Virtual Reality application for 3D map exploration is the main goal of this particular project. The system will have various types of interactions that can be used to carry out the available actions in the environment, such as exploring the terrain, reading information about points of interest, creating markers along the map, providing routes between said markers, or even traveling to other cities while being able to see the land from various angles. The Oculus Quest 2 (Oculus Touch) controllers, hand gesture detection, or tracking of extra sensors dispersed throughout the body and objects can all be used for these interactions. Each one increases the user's sense of immersion and presence in the virtual world.

To carry out the project, a preliminary investigation is made into the potential benefits offered by a cutting-edge technology like virtual reality, as well as the usage of the tools (such as Unity, Mapbox, etc.) and methods required for this development to be as fruitful as feasible. as ideal as feasible, and the application produces the intended results.

AGRADECIMIENTOS

Quiero agradecer a mi familia y amigos por apoyarme en todo momento y por hacerme mucho más fácil todo este proceso, sin ellos hubiera sido todo más complicado y sobre todo a mi tutor Alejandro por guiarme, ayudarme y estar disponible en todo momento para cualquier duda.

Gracias.

ÍNDICE GENERAL

1. INTRODUCCIÓN	1
1.1 Contexto	1
1.2 Motivación	3
1.3 Objetivos	5
1.4 Herramientas utilizadas	5
1.5 Estructura del documento.....	6
2. ESTADO DEL ARTE	8
2.1 Realidad Virtual	8
2.2 Dispositivos	10
2.3 Motores de desarrollo.....	12
2.4 APIs o Librerías	14
2.4.1 APIs Mapas VR.....	15
2.5 Aplicaciones similares	16
2.6 Dispositivos de Entrada/Salida.....	19
2.7 Conclusiones.....	20
3. ANÁLISIS	22
3.1 Descripción de la aplicación	22
3.2 Requisitos de usuario	23
3.2.1 Características del usuario	24
3.2.2 Entorno operativo y de seguridad.....	24
3.2.3 Suposiciones.....	24
3.2.4 Requisitos específicos	25
3.3 Requisitos de software.....	33
3.3.1 Requisitos funcionales.....	34
3.3.2 Requisitos operacionales.....	43
3.3.3 Requisitos de usabilidad.....	44
3.4 Casos de uso.....	45
3.5 Trazabilidad	54
3.5.1 Matriz de trazabilidad Requisitos de usuario-Requisitos de software.....	54
3.5.2 Matriz de trazabilidad Casos de uso-Requisitos de software	56
4. DISEÑO E IMPLEMENTACIÓN	57
4.1 Diseño.....	57

4.1.1 Arquitectura	57
4.1.2 Componentes	59
4.1.3 Diseño gráfico.....	69
4.1.4 Controles de juego	73
4.2 Implementación	77
4.2.1 Escenas	77
4.2.2 Scene Controller (Controlador de escenas)	78
4.2.3 GameManager.....	78
4.2.4 Movimiento de cámara	78
4.2.5 Otros.....	79
5. ENTORNO SOCIOECONÓMICO	82
5.1 Impacto socioeconómico	82
5.2 Presupuesto	83
6. MARCO REGULADOR.....	85
7. PRUEBAS.....	86
8. PLANIFICACIÓN.....	90
9. CONCLUSIONES	93
9.1 Experiencia personal	93
9.2 Trabajos futuros	94
SUMMARY	95
1. Introduction.....	95
1.1 Motivation.....	95
1.2 Objectives.....	96
2. State of Art	96
2.1 Virtual Reality	96
2.2 Devices	97
2.3 APIs/Library	97
2.4 Similar Projects.....	98
2.5 Input/Output Devices.....	99
3. Analysis.....	99
4. Design and Implementation	100
4.1 Architecture.....	100
4.2 Components	101
4.3 Implementation.....	101
5. Socio-economic environment	102
5.1 Budget	102

6. Regulatory Framework.....	103
7. Planning.....	103
8. Conclusions.....	103
8.1 Personal experience	104
8.2 Future works	104
BIBLIOGRAFÍA.....	106

ÍNDICE DE FIGURAS

.....	i
Fig. 1.1 Gráfico del número de dispositivos de Realidad Virtual conectados a Steam.	2
Fig. 1.2 Estimación de ingresos mundiales en la Realidad Virtual.	3
Fig. 1.3 Evolución del nº de viajeros en el Mundo	4
Fig. 2.1 Diagrama del continuo de virtualidad.	8
Fig. 2.2 Prototipo del Simulador Sensorama (M. Heilig)	9
Fig. 2.3 Gafas Samsung VR	10
Fig. 2.5 Imagen Star Wars Jedi: Fallen Order	13
Fig. 2.7 Imagen de ruta creada por Mapbox entre dos puntos	16
Fig. 2.8 Imagen de los datos de sonido	17
Fig. 2.9 Imagen dentro del juego Google Earth VR	18
Fig. 2.10 Imagen de la aplicación Wander para Oculus	18
Fig. 2.11 Imagen de ejemplo de uso sensores Antilatency	20
Fig. 2.12 Imagen de la KAT Walk C2 Treadmill	20
Fig. 3.1 Diagrama de casos de uso	45
Fig. 4.1 Diagrama de Arquitectura del sistema	58
Fig. 4.2 Vistas disponibles del mapa.....	69
Fig. 4.3 Gráficos puntos de información	70
Fig. 4.4 Grafismo del marcador puesto por el usuario.....	71
Fig. 4.5 Imagen de ruta desde la vista de pájaro.....	71
Fig. 4.6 Imagen de los modelos de manos	72
Fig. 4.7 Imagen de los modelos para los sensores Antilatency.....	72
Fig. 4.8 Imagen de los menús disponibles en la aplicación	73
Fig. 4.9 Controles con los mandos de Oculus Quest 2	74
Fig. 4.10 Controles con los gestos manuales	76
Fig. 4.11 Escenas de la aplicación.....	77
Fig. 4.12 Configuración de los gestos manuales	80
Fig. 4.13 Entorno de sensores Antilatency.....	80
Fig. 8.1 Diagrama de Gantt de planificación del proyecto	69
Fig. A.3. System Architecture diagram.....	100
Fig. A.4. Images in game of the application	102

ÍNDICE DE TABLAS

TABLA 2.1 COMPARACIÓN ENTRE DISPOSITIVOS DE REALIDAD VIRTUAL	11
TABLA 2.2 COMPARACIÓN ENTRE MOTORES DE DESARROLLO	14
TABLA 3.1: PLANTILLA DE REQUISITOS DE USUARIO	25
TABLA 3.2 REQUISITO DE USUARIO RU-C1	26
TABLA 3.3 REQUISITO DE USUARIO RU-C2	26
TABLA 3.4 REQUISITO DE USUARIO RU-C3	26
TABLA 3.5 REQUISITO DE USUARIO RU-C4	27
TABLA 3.6 REQUISITO DE USUARIO RU-C5	27
TABLA 3.7 REQUISITO DE USUARIO RU-C6	27
TABLA 3.8 REQUISITO DE USUARIO RU-C7	28
TABLA 3.9 REQUISITO DE USUARIO RU-C8	28
TABLA 3.10 REQUISITO DE USUARIO RU-C9	28
TABLA 3.11 REQUISITO DE USUARIO RU-C10	29
TABLA 3.12 REQUISITO DE USUARIO RU-C11	29
TABLA 3.13 REQUISITO DE USUARIO RU-C12	29
TABLA 3.14 REQUISITO DE USUARIO RU-C13	30
TABLA 3.15 REQUISITO DE USUARIO RU-C14	30
TABLA 3.16 REQUISITO DE USUARIO RU-C15	30
TABLA 3.17 REQUISITO DE USUARIO RU-C16	31
TABLA 3.18 REQUISITO DE USUARIO RU-C17	31
TABLA 3.19 REQUISITO DE USUARIO RU-R1	31
TABLA 3.20 REQUISITO DE USUARIO RU-R2	32
TABLA 3.21 REQUISITO DE USUARIO RU-R3	32
TABLA 3.22 REQUISITO DE USUARIO RU-R4	32
TABLA 3.23 REQUISITO DE USUARIO RU-R5	33
TABLA 3.24: PLANTILLA DE REQUISITOS DE SOFTWARE	34
TABLA 3.25: REQUISITO DE SOFTWARE RS-F1	34
TABLA 3.26: REQUISITO DE SOFTWARE RS-F2	35
TABLA 3.27: REQUISITO DE SOFTWARE RS-F3	35
TABLA 3.28: REQUISITO DE SOFTWARE RS-F4	35
TABLA 3.29: REQUISITO DE SOFTWARE RS-F5	36
TABLA 3.30: REQUISITO DE SOFTWARE RS-F6	36
TABLA 3.31: REQUISITO DE SOFTWARE RS-F7	36

TABLA 3.32: REQUISITO DE SOFTWARE RS-F8	37
TABLA 3.33: REQUISITO DE SOFTWARE RS-F9	37
TABLA 3.34: REQUISITO DE SOFTWARE RS-F10	38
TABLA 3.35: REQUISITO DE SOFTWARE RS-F11	38
TABLA 3.36: REQUISITO DE SOFTWARE RS-F12	38
TABLA 3.37: REQUISITO DE SOFTWARE RS-F13	39
TABLA 3.38: REQUISITO DE SOFTWARE RS-F14	39
TABLA 3.39: REQUISITO DE SOFTWARE RS-F15	39
TABLA 3.40: REQUISITO DE SOFTWARE RS-F16	40
TABLA 3.41: REQUISITO DE SOFTWARE RS-F17	40
TABLA 3.42: REQUISITO DE SOFTWARE RS-F18	40
TABLA 3.43: REQUISITO DE SOFTWARE RS-F19	41
TABLA 3.44: REQUISITO DE SOFTWARE RS-F20	41
TABLA 3.45: REQUISITO DE SOFTWARE RS-F21	41
TABLA 3.46: REQUISITO DE SOFTWARE RS-F22	42
TABLA 3.47: REQUISITO DE SOFTWARE RS-F23	42
TABLA 3.48: REQUISITO DE SOFTWARE RS-F24	42
TABLA 3.49: REQUISITO DE SOFTWARE RS-O1	43
TABLA 3.50: REQUISITO DE SOFTWARE RS-O2	43
TABLA 3.51: REQUISITO DE SOFTWARE RS-O3	44
TABLA 3.52: REQUISITO DE SOFTWARE RS-O4	44
TABLA 3.53: REQUISITO DE SOFTWARE RS-U1	44
TABLA 3.54: PLANTILLA DE CASOS DE USO	46
TABLA 3.55: CASO DE USO CU-1	46
TABLA 3.56: CASO DE USO CU-2	47
TABLA 3.57: CASO DE USO CU-3	48
TABLA 3.58: CASO DE USO CU-4	48
TABLA 3.59: CASO DE USO CU-5	49
TABLA 3.60: CASO DE USO CU-6	50
TABLA 3.61: CASO DE USO CU-7	50
TABLA 3.62: CASO DE USO CU-8	51
TABLA 3.63: CASO DE USO CU-9	51
TABLA 3.64: CASO DE USO CU-10	52
TABLA 3.65: CASO DE USO CU-11	53
TABLA 3.66: CASO DE USO CU-12	53
TABLA 3.67: CASO DE USO CU-13	54

TABLA 3.68: MATRIZ DE TRAZABILIDAD RU-RS.....	1
TABLA 3.69: MATRIZ DE TRAZABILIDAD CU-RS.....	56
TABLA 4.1: PLANTILLA DE COMPONENTES	59
TABLA 4.2: COMPONENTE CO-1.....	60
TABLA 4.3: COMPONENTE CO-2.....	60
TABLA 4.4: COMPONENTE CO-3.....	60
TABLA 4.5: COMPONENTE CO-4.....	61
TABLA 4.6: COMPONENTE CO-5.....	61
TABLA 4.7: COMPONENTE CO-6.....	62
TABLA 4.8: COMPONENTE CO-7.....	62
TABLA 4.9: COMPONENTE CO-8.....	62
TABLA 4.10: COMPONENTE CO-9.....	63
TABLA 4.11: COMPONENTE CO-10.....	63
TABLA 4.12: COMPONENTE CO-11.....	64
TABLA 4.13: COMPONENTE CO-12.....	64
TABLA 4.14: COMPONENTE CO-13.....	64
TABLA 4.15: COMPONENTE CO-14.....	65
TABLA 4.16: COMPONENTE CO-15.....	65
TABLA 4.17: COMPONENTE CO-16.....	66
TABLA 4.18: COMPONENTE CO-17.....	66
TABLA 4.19: COMPONENTE CO-18.....	66
TABLA 4.20: MATRIZ DE TRAZABILIDAD RS-CO.....	56
TABLA 4.21: CONTROLES MANDOS OCULUS TOUCH.....	73
TABLA 4.22: CONTROLES/GESTOS CON RECONOCIMIENTO DE MANOS.....	75
TABLA 4.23: CONTROLES/GESTOS SENSORES ANTILATENCY.....	76
TABLA 6.1. COSTE DEL PERSONAL.....	83
TABLA 6.2: COSTE DE RECURSOS	84
TABLA 7.1: PLANTILLA DE PRUEBAS	86
TABLA 7.2: PRUEBA P-1	87
TABLA 7.3: PRUEBA P-2	87
TABLA 7.4: PRUEBA P-3	88
TABLA 7.5: PRUEBA P-4	89
TABLA 7.6: PRUEBA P-5	89

1. INTRODUCCIÓN

En esta parte del documento se va a hablar tanto del contexto en el que se encuentra el ámbito sobre el que se va a realizar el trabajo, así como explicar cuál es la motivación de este y qué problemas se van a intentar solventar con la aplicación propuesta.

1.1 Contexto

El desarrollo tecnológico llevado a cabo durante los últimos años ha permitido que las personas vean tecnologías que para padres o abuelos hubieran sido difíciles de imaginar años atrás, como por ejemplo poder realizar una videollamada y hablar y ver a una persona desde cualquier parte del mundo, realizar pagos con un smartphone y que el dinero en efectivo poco a poco esté desapareciendo o disminuyendo su uso, etc.

Dentro de estas tecnologías se encuentran los dispositivos de Realidad Virtual, esta Realidad Virtual, aunque posteriormente se vaya a realizar una mayor descripción de su historia y capacidades, consiste en un entorno informático y virtual que mediante el uso de unas gafas o cascos da la sensación al usuario de encontrarse dentro de ese mundo. Es la sensación de estar inmerso en un mundo o ambiente que parece real pero realmente no lo es.

En los últimos años estos dispositivos han pasado de ser algo relegados a tecnología de última generación accesible más fácilmente para empresas o universidades, a soluciones a las que puede llegar un mayor número de gente. A esto también se debe en parte a las mejoras que se han producido en los inconvenientes de uso que han implicado siempre el uso de Realidad Virtual, con avances en comodidad, interacción y demás han sido de gran ayuda. Se estima que para el año 2023 el mercado de Realidad Virtual en todo el mundo suponga unos 34.5 mil millones de dólares. [1]

Ya no es solo que el mercado de la Realidad Virtual aumente en su valor por empresas queriendo desarrollar sus propias soluciones o alternativas, sino que el número de usuarios reales también aumenta considerablemente.

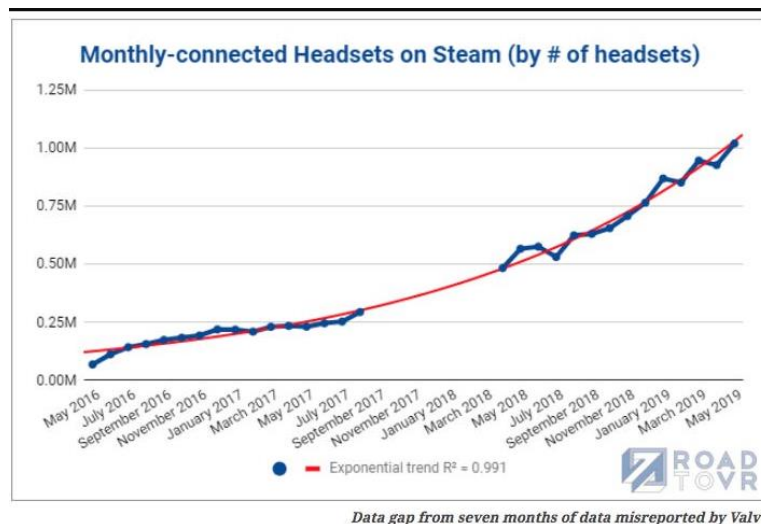


Fig. 1.1 Gráfico del número de dispositivos de Realidad Virtual conectados a Steam. [2]

Como se puede ver en la figura se muestra el número de distintos dispositivos VR (gafas, cascos, etc.) conectados mensualmente a Steam, la mayor plataforma de distribución de videojuegos de PC y en este caso de estos dispositivos. Año tras año este número de dispositivos conectados mensualmente aumenta en un 80%. Tanto es así que compañías muy reconocidas en el ámbito de videojuegos, pero más enfocadas a consolas de sobremesa o portátiles han sacado sus propias versiones de estos dispositivos para disfrutar de sus videojuegos de otra manera. Por ejemplo, está la compañía Sony creadora de PlayStation, sacó las PlayStation VR unas gafas de Realidad Virtual compatibles con la PS4 o la empresa Nintendo que ha desarrollado las Nintendo Labo: VR kit como complemento para la Nintendo Switch y poder disfrutar de esta Realidad Virtual. [2]

Pese a parecer solo tener éxito y usuarios en el mundo de los videojuegos, el abanico de posibilidades dentro de la Realidad Virtual va mucho más allá de juegos.

Es más, se predice que el número de ventas de dispositivos de Realidad Virtual compatibles con los smartphones sea mayor por ejemplo al de basados en PC. Básicamente debido al precio, que a pesar de ir disminuyendo al haber un mercado competitivo con distintas ofertas los primeros tienen un coste inferior a los segundos. [3]

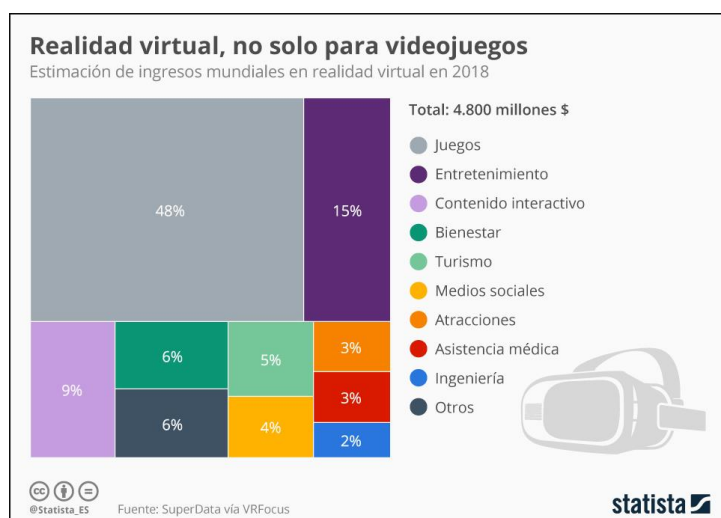


Fig. 1.2 Estimación de ingresos mundiales en la Realidad Virtual. [4]

La industria que más parte se lleva de todo este beneficio es la de videojuegos, con casi la mitad de los ingresos. Pero aparte están industrias como la del entretenimiento, por ejemplo, el consumo multimedia (películas, vídeos, llamadas, etc.), el turismo, la asistencia médica... [4] Por ello en este trabajo se va a tratar de explorar otro ámbito distinto del de videojuegos e intentar unirlo con otros ámbitos de importancia.

Este trabajo se va a centrar en el desarrollo de una aplicación más alejada del ámbito de los videojuegos y puede aportar otros elementos aparte del entretenimiento.

Concretamente con este proyecto se va a tratar de explorar un ámbito de Realidad Virtual trabajando con distintas formas novedosas de interacción y análisis de un entorno virtual.

1.2 Motivación

El sector del turismo que a lo largo de los últimos 2 años ha sufrido mucho debido a la pandemia de COVID-19, por sus debidas restricciones para contener la misma. Aun así, es un sector que mueve millones de personas a lo largo del año con sus debidos ingresos en el país correspondiente.

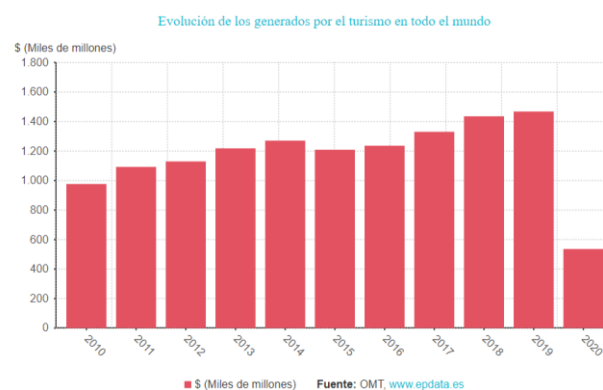
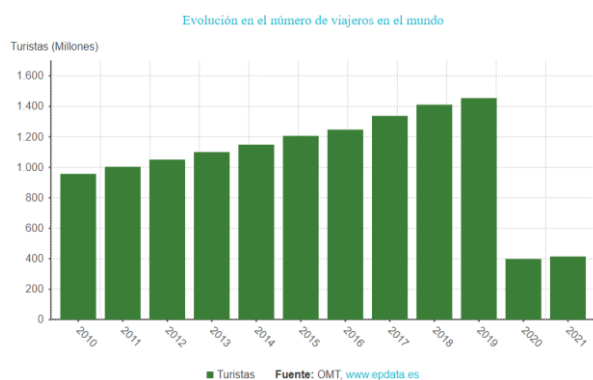


Fig. 1.3 Evolución del nº de viajeros en el Mundo [5]

Fig. 1.4 Evolución de ingresos por turismo [5]

En ambas gráficas se puede ver tanto la evolución del número de turistas totales alrededor del mundo como los ingresos generados por los mismos [5], es clara la bajada debida a la pandemia, pero como conclusión se puede sacar que el turismo mueve numerosas cantidades de personas y dinero, y es un sector en el que se puede innovar para encontrar otras posibilidades de viajar o incentivar el mismo.

La importancia global de la industria turística es muy marcada pues supone la creación de alrededor de 270 millones de empleos y además su aportación al PIB mundial en el último año consiguió superar los 4.5 billones de dólares. Todo esto no es lo que genera, sino que también atrae a otras empresas ya sean públicas o privadas a invertir en este sector [6], de ahí otro aspecto importante en la realización de este trabajo.

Estos datos más ligados con el turismo tradicional se pueden alejar un poco con el propósito principal de la aplicación, que es explorar un plano o mapa 3D, pero es una forma de exponer los usuarios potenciales que puede tener la aplicación. Pues como se ve es un sector con numerosos usuarios.

Una vez que una o un grupo de personas decide visitar una ciudad es complicado decidir qué ver en la misma y organizar ese viaje, pues son muchas las empresas que te dan la posibilidad de contratar tours turísticos que recorren los puntos más importantes de la ciudad y un guía explica los aspectos más curiosos de estos. Un ejemplo de ello es la empresa Civitatis que permite la reserva de distintos tours dependiendo de la ciudad, con distintos precios, imágenes ya bien proporcionadas por la empresa o perteneciente a opiniones subidas por otros usuarios. Teniendo en cuenta este tipo de empresas es complicado encontrar alternativas en las que sea el individuo el que organice ese viaje o este sea capaz de crear rutas, ver información de estas, etc.

Por todo ello este trabajo se centra en utilizar la tecnología de la Realidad Virtual que como se ha visto anteriormente está en auge para poder interactuar con datos de mapas, puntos de interés en un espacio inmersivo. Las ventajas que puede ofrecer la Realidad Virtual en la visualización e interacción con mapas es por ejemplo el amplio campo de visión que ofrece al usuario y hace que este se vea mucho más inmerso en la acción sin distracciones y poder disfrutar del proceso, así como una interacción más intuitiva con la información y los datos.

Se ha nombrado el aspecto de un turismo individual como caso de uso de la aplicación, pero también pueden estar dentro de la misma dándole otro uso por ejemplo una organización de una carrera de Fórmula 1, mediante la app se puede ver en Realidad Virtual las calles por las que puede pasar el circuito con puntos de interés que puedan ser llamativos a la hora de la carrera calculando la ruta por los puntos especificados y hallando distancias y demás. Todo ligado con la exploración de mapas en 3D y las distintas formas de interacción y posibilidades que ofrece el uso de estos tipos de entornos virtuales.

De igual forma y como otra motivación respecto a la realización de este trabajo era la posibilidad de desarrollar la aplicación con una tecnología que no había sido vista a lo largo de la carrera como lo es Unity para el desarrollo de videojuegos unido con la Realidad Virtual que tanto por separado como juntas tienen un número de posibilidades casi infinitas. Era un aspecto que a pesar de tener que aprender nuevas tecnologías suponía el desarrollo de nuevas aptitudes y conocimientos que pueden ser de gran ayuda en el futuro y en los que se puede seguir trabajando.

1.3 Objetivos

Tras haber visto las distintas motivaciones nombradas anteriormente, los objetivos que se quieren conseguir con la realización de este proyecto son los siguientes:

- Creación de una aplicación o juego de Realidad Virtual en el que poder visualizar mapas en un entorno 3D como si se tratara de una calle real, a su vez ser capaz de explorar el mapa con distinta información relevante y poder establecer posibles puntos de interés con sus rutas correspondientes.
- Creación de un entorno tanto virtual como físico en el que poder reproducir correctamente la aplicación.
- Crear distintas formas de interacción con la aplicación con las que realizar diversas acciones, ya sea con mandos de control, gestos manuales, sensores, etc.
- Estudio de nuevas formas de interacción con entornos y mapas virtuales. A pesar de no ser un objetivo principal como tal de la aplicación, con la realización de este trabajo se va a explorar las diferentes capacidades de la Realidad Virtual aun siendo una tecnología que se encuentra en desarrollo constante y las capacidades que ofrece para la interacción con este tipo de sistemas.

1.4 Herramientas utilizadas

Las principales herramientas que se han utilizado para llevar a cabo el desarrollo tanto del proyecto como de la aplicación son las siguientes:

- **Ordenador portátil.** Ordenador personal con el que se ha realizado la mayor parte del desarrollo de la aplicación, aunque las pruebas se realizaban de mejor forma en las propias gafas de Realidad Virtual. Se trata de un Xiaomi Mi Notebook Pro (2018) con procesador Intel Core i5-8250U, una memoria RAM de 8GB. Almacenamiento SSD de 512 GB y con una tarjeta gráfica dedicada Nvidia GeForce MX250.
- **Gafas de Realidad Virtual.** En este proyecto entre las distintas opciones del mercado se ha optado por el casco de Realidad Virtual de la compañía Oculus VR de Facebook, más en concreto de las Oculus Quest 2.

- **Plataforma de desarrollo.** El proyecto se ha llevado a cabo con Unity 3D, motor de videojuego que permite la creación de plataformas y entornos de desarrollo, en este caso para Microsoft Windows. Dentro de la propia plataforma se ha hecho el uso de distintos SDK de este como los son el de Android, Oculus Interaction o Mapbox con el que se ha implementado la mayor parte de interacción con los mapas y que se explicarán sus capacidades más adelante.
- **Sensores Antilatency.** Se trata de unos sensores para Realidad Virtual compatibles a su vez con Unity la plataforma de desarrollo, que sirve para el seguimiento de estos y poder implementar funcionalidades con movimientos reales que realice la persona, como movimiento de piernas, brazos, etc. [7]
- **Procesador de texto.** Para la redacción de este documento se ha utilizado tanto el Microsoft Word en el que realmente se ha escrito y redactado, como el complemento de Google Docs de Google Drive para ir haciendo copias de seguridad de distintas versiones y no perder el progreso.

1.5 Estructura del documento

Los contenidos de los que dispone este documento están estructurados en torno a los siguientes apartados o capítulos:

1. **Introducción.** Es el apartado en el que se está presente, contiene tanto el contexto como las motivaciones que han conducido a la realización de este proyecto en concreto, así como los objetivos que se intentan desarrollar para poder cumplir con las motivaciones nombradas o solucionar los problemas si ese fuera el caso. De la misma forma también se encuentra un breve listado de las principales herramientas que se han utilizado durante el periodo de realización del proyecto y la estructura que va a seguir el documento.
2. **Estado del arte.** En este capítulo se da un poco de contexto a dónde se encuentran las tecnologías utilizadas, analizando su historia si es posible y algunas de las alternativas que se han barajado en cada una de ellas. Del mismo modo se examinan otras aplicaciones similares a la del proyecto y hacia dónde puede ir orientado el futuro respecto a estos sistemas.
3. **Análisis.** Consiste en una explicación del sistema que se va a desarrollar, especificando sus funcionalidades y sus requisitos de usuario. Con los que después poder enumerar distintos posibles casos de uso.
4. **Diseño e Implementación.** Se define la arquitectura completa del sistema, dividiéndolo en componentes los cuales realizan una función específica. Se detalla los distintos aspectos de la implementación del sistema, es decir la forma en la que se ha llegado a la solución correspondiente para utilizar las gafas, moverse por el mundo de la aplicación, realizar acciones, etc.
5. **Entorno socioeconómico.** Se estudia y analiza el posible impacto tanto económico como social que puede tener la aplicación de este proyecto, incluyendo

a su vez un presupuesto que se tendría que cumplir a la hora de la realización de esta.

6. **Marco regulador.** Apartado en el que se hace un análisis de la legislación que se encuentra en torno a este trabajo, que problemas puede haber y como solucionarlos en todo caso. Así como otros estándares aplicables al mismo.
7. **Pruebas.** Consiste en la exposición de las pruebas a realizar para la comprobación del correcto funcionamiento del sistema.
8. **Planificación.** Especificación de la planificación del proyecto, desde las primeras decisiones a tomar hasta la finalización del mismo.
9. **Conclusiones.** Apartado con las conclusiones y comentarios personales finales acerca del proyecto, del periodo de desarrollo y posibles trabajos futuros con los que mejorar distintos aspectos.

En la parte final del documento se incluye tanto la bibliografía utilizada para la documentación del proyecto como el resumen en inglés.

2. ESTADO DEL ARTE

En este apartado se describen las distintas tecnologías utilizadas en el desarrollo del proyecto, teniendo en cuenta su historia y hacia dónde va dirigido el futuro de estas. Se detallarán aspectos importantes de la Realidad Virtual, variedad de dispositivos, alternativas similares a la aplicación propuesta, etc.

Con todo esto se especificará que decisiones se han tomado en cuanto a las alternativas disponibles y el porqué de estas.

2.1 Realidad Virtual

En primer lugar, hay que diferenciar qué significa Realidad Virtual y qué otras posibilidades hay con las que se puede confundir. Como se puede ver en la imagen existen distintos tipos de realidad o conceptos dependiendo de si se encuentra en un mundo completamente virtual o no. El espacio comprendido entre el mundo real y el virtual es el denominado de Realidad Mixta que dependiendo de la proporción de virtualidad se distingue entre Realidad Aumentada y Virtualidad Aumentada. Este trabajo se centra en la Realidad Virtual en el que el mundo que se simula es completamente virtual.

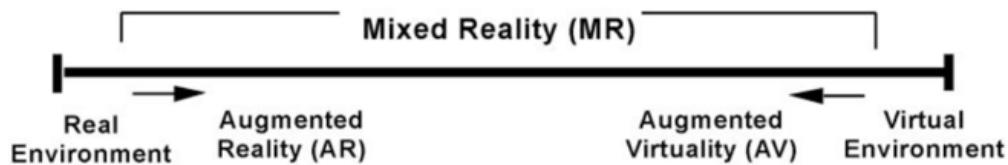


Fig. 2.1 Diagrama del continuo de virtualidad. [8]

Por ello la Realidad Virtual es un tipo de realidad creada por ordenador que responde a los "input" proporcionados por el usuario, es un aspecto clave el de interacción en tiempo real.

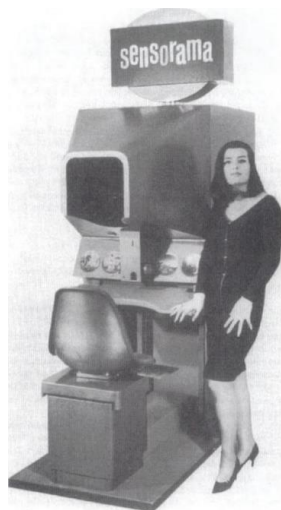


Fig. 2.2 Prototipo del Simulador Sensorama (M. Heilig) [9]

Es en 1962 cuando surge un gran invento de la Realidad Virtual por parte de Morton Heilig, un cineasta que creó un aparato llamado Sensorama con el que ya no solo estimulaba el sentido visual, sino que también el olfato, el tacto y el auditivo. Gracias a esto la Realidad Virtual entró en un espectro completamente distinto y considerablemente más avanzado, se alejaba de la simple impresión de imágenes y se hacía la búsqueda de acercar al usuario a un entorno que realmente no existe [9]. Como se ha mencionado anteriormente, se crea un mundo que no es real sino virtual.

Con este novedoso avance en el sector de Realidad Virtual, es claro ver que entran en juego dos aspectos que en un principio no eran perceptibles y que en la actualidad son indispensables para estos sistemas, son la inmersión y la presencia. Son conceptos que pueden llegar a confundir por lo que hay que saber la diferencia.

La inmersión se refiere al nivel sensorial que proporciona un sistema de Realidad Virtual desde el punto de vista objetivo. Este nivel de inmersión depende de los monitores y el software que renderiza el sistema, es objetivo y se puede medir ese nivel de inmersión. Sin embargo, la presencia es un aspecto individual que depende del contexto del usuario relacionado con la experiencia de estar en ese mundo virtual [10].

Por lo tanto, la presencia se entiende como algo más psicológico que depende individualmente del usuario es la percepción de la inmersión, por ejemplo, dos personas pueden sentirse presentes de distinta manera en un mismo entorno con las mismas características de inmersión o incluso una misma persona habiendo pasado un tiempo entre pruebas. Dependiendo del estado mental, acontecimientos u otros factores.

2.2 Dispositivos

En el campo de la Realidad Virtual existe un abanico muy amplio de dispositivos capaces de realizar las acciones necesarias para que el usuario potencial final de la aplicación se encuentre lo mayor inmerso posible dentro del mundo virtual. Para ello el tipo más apropiado son las denominadas gafas de Realidad Virtual, también denominados dispositivos HMD (Head Mounted Display) que como su propio nombre indica son aparatos que se colocan en la cabeza a la altura de los ojos para observar el entorno virtual en todo momento.

A su vez estos dispositivos tienen distintos sensores tanto internos como externos para poder controlar la posición y rotación de la cabeza y así responder a los movimientos del usuario en el mundo virtual de la forma más fiel a la realidad posible. Para conseguir como se ha mencionado anteriormente aspectos indispensables en la Realidad Virtual como la inmersión o la respuesta en tiempo real de la interacción.

Una opción que ha sido descartada debido a las pocas posibilidades que proporciona son las gafas sin pantalla que necesitan de un smartphone u otro dispositivo para poder funcionar. Al necesitar de un entorno de desarrollo profesional este tipo de aparato no cumple con las especificaciones necesarias en cuanto a potencia y capacidad para poder realizar las debidas pruebas.



Fig. 2.3 Gafas Samsung VR



Fig. 2.4 Nintendo Switch VR

Como se puede ver en ambas imágenes son dispositivos que sirven como meros accesorios a un smartphone o a una videoconsola y con ello disfrutar de contenidos multimedia con un alto componente de Realidad Virtual, son buenos ejemplos para las personas que quieran iniciarse en este mundo.

En el caso de este proyecto se han barajado distintas opciones dentro de la gama de dispositivos compatibles con ordenador y que dan posibilidades a desarrolladores para crear sus propias aplicaciones. Entre las más importantes y que realmente son competidoras entre sí están:

- **HTC Vive Pro.** Considerado uno de los visores más potentes del mercado, con una pantalla de 3.5 pulgadas con resolución de 2880 x 1600 píxeles. Con un visor con tasa de refresco de 90 Hz y un campo de visión de 110°. Además, incorporan unos auriculares capaces de proyectar sonido en 3D, perfecto para la interacción con el ambiente virtual. Compatibilidad con Steam y dispositivos Windows [11].
- **Oculus Rift S.** Dispositivo de la empresa Oculus perteneciente a la compañía Meta o Facebook que en colaboración con Lenovo han sacado estas gafas al mercado. Disponen de una pantalla con una resolución y tasa de refresco algo inferior a la anterior y del mismo modo necesitan estar conectadas por cable a un ordenador Windows para poder funcionar [12].
- **Valve Index.** Gafas creadas por la empresa Valve, de los creadores también de Steam, con resolución de pantalla de 2880x1600 píxeles y con tasas de refresco que pueden llegar a 144 Hz. Cuentan con uno de los campos de visión más amplios del mercado y con un sistema de sonido con auriculares muy similar a las de HTC Vive Pro [13].
- **Oculus Quest 2.** Gafas autónomas de Meta que, a pesar de tener características técnicas algo inferiores a sus competidores directos, cuenta por ejemplo con la posibilidad de tener un cable Link USB-C que aparte de conectar el dispositivo a un ordenador para poder jugar a juegos de escritorio, ofrece el servicio de Oculus Link con el cual el desarrollador podrá realizar pruebas y avances de forma mucho más sencilla. De igual modo entre los dispositivos investigados es el único que ofrece la posibilidad de su uso, ya sea juego previamente descargado o propio sin estar conectado a un ordenador [14].

Para realizar una comparación más detenida y visual de cada uno de estos dispositivos, se va a hacer el uso de una tabla con las características principales de las gafas de Realidad Virtual y así ver en qué aspectos destacan cada una de las alternativas.

TABLA 2.1 COMPARACIÓN ENTRE DISPOSITIVOS DE REALIDAD VIRTUAL

	HTC Vive Pro	Oculus Rift S	Valve Index	Oculus Quest 2
Resolución	2880x1600	2560x1440	2880x1600	1832x1920 por ojo
Tasa de refresco	90 Hz	80 Hz	144 Hz	90 Hz, capado a 72 Hz

Campo de visión	110°	110°	130°	110°
Peso	503 g	470 g	809 g	503 g
Capacidad	256 GB	256 GB	256 GB	256 GB
Auriculares	Sí	Sistema estéreo	Sí	Sistema estereo
Compatibilidad	Windows, Steam, HTC Viveport	Wndows, OculusStore, Meta	Windows, Steam VR	Windows, Oculus Store, Meta
Necesidad PC	Sí	Sí	Sí	No es nesario, pero si posible
Precio	789 €	449 €	1079 €	449 €
Imagen				

Todos estos dispositivos disponen de mandos que simulan las manos en el entorno virtual y sirven como método de input para poder controlar y realizar acciones en el mundo creado, aparte de la interacción con el propio sistema operativo de cada una de las gafas. Los mandos contienen botones, joysticks y sensores que detectan la proximidad de forma que se simule movimiento de dedos y manos y así conseguir una interacción intuitiva y natural.

De igual modo también poseen sensores para reconocer el movimiento externo como por ejemplo el de las manos y de esa forma no necesitar de otros dispositivos que puedan entorpecer de algún modo la interacción y ensuciar la experiencia del usuario.

2.3 Motores de desarrollo

En este apartado se va a entrar en detalle en las alternativas en cuanto a motores de juego se disponen y que a su vez sean compatibles para el desarrollo en Realidad Virtual. En este caso se han tenido en consideración dos opciones, una el motor Unreal Engine de Epic Games y por otro lado Unity3D. Ambos son dos de los motores más conocidos en el ámbito de desarrollos de videojuegos, los que tienen una mayor comunidad y como consecuencia un mayor número de recursos y soluciones externas, que a la hora de aprender una nueva herramienta es algo a tener en cuenta.

Unreal Engine con distintas versiones dependiendo del año y de las características necesarias para el desarrollo, es de los motores con los que se han creado videojuegos jugados alrededor del mundo. Ofrece herramientas muy llamativas y que son de gran ayuda a la hora de crear mundos con sus partículas, paisajes, terrenos, elementos de iluminación, animaciones, etc. Todo ello con la posibilidad de desarrollo basado en blueprints y no es necesario ningún tipo de programación, utiliza conceptos más cercanos a los desarrolladores de videojuegos con el uso de nodos [15].

Esto se puede considerar una desventaja pues lo visto a lo largo del grado ha sido más próximo a la generación de scripts y programación. Del mismo modo para poder simular entornos con Unreal es necesario un equipo más potente que por ejemplo con Unity3D, pues como se ha mencionado se pueden crear ambientes mucho más reales, esto también puede suponer una desventaja.



Fig. 2.5 Imagen Star Wars Jedi: Fallen Order [16]



Fig. 2.6 Imagen del videojuego Fortnite [17]

Como se puede ver en ambas imágenes con el uso de Unreal Engine se han hecho desde hace años videojuegos con renombre en el sector, como uno de los últimos de una saga con numerosos seguidores alrededor del mundo “Star Wars Jedi: Fallen Order”, o uno de los videojuegos más jugados en los últimos años como lo es Fortnite con aproximadamente 350 millones de usuarios alrededor del mundo. Unreal Engine es una plataforma con la que su uso es extendido en el sector mundial del videojuego.

Por otro lado, como ya se ha mencionado se encuentra Unity3D que es un motor 3D multiplataforma menos completo que Unreal en cuanto a gráficos, pero más sencillo para principiantes y aprender en este ámbito. A su vez este sí dispone de un desarrollo más cercano a la programación ya vista con lenguajes como C# o Javascript que hará que el proceso de aprendizaje sea menos costoso.

A continuación, se muestra una tabla con las principales diferencias entre ambos motores para que quede más claro.

TABLA 2.2 COMPARACIÓN ENTRE MOTORES DE DESARROLLO

	Unreal Engine	Unity3D
Soporta Realidad Virtual	Sí	Sí
Programación	Orientada a diseñadores	Orientada a programadores
Lenguaje de programación	C++	C#, Javascript
Gráficos	Última generación	Limitados
Especificaciones necesarias	Amplios recursos	Utiliza menos recursos
Contenido de terceros	Escaso	Amplio
Comunidad	Amplia y activa	Amplia y activa
Multiplataforma	Principalmente PC o consola	Todo tipo de dispositivos

Un aspecto de vital importancia es el contenido de terceros que pueden ofrecer de gran ayuda a la hora de desarrollo. En el caso de Unity3D posee una gran cantidad de librerías o assets ya sean propio de la empresa Unity o individuales que facilitan el desarrollo de la aplicación. Esto proporciona la posibilidad de poder centrarse en otros aspectos importantes de la solución final.

2.4 APIs o Librerías

En este apartado se especifican las alternativas en cuanto a APIs o librerías disponibles y cuales se han utilizado a la hora del desarrollo de la aplicación. Con el uso de estos de recursos proporcionados y creados por terceros, el proceso de creación de la aplicación se ve facilitado. Proporcionan soluciones útiles a problemas que se tienen durante el desarrollo.

Por una parte, se encuentra el SDK de Oculus o también llamado el Asset de Oculus Integration disponible en la tienda de Assets de Unity desarrollado por el propio Oculus y que proporcionan distintas facilidades a la hora de desarrollar un proyecto de Realidad Virtual. Contiene librerías y herramientas para facilitar por ejemplo el reconocimiento de

las entradas en el mando de control por parte del usuario o reconocimiento de gestos manuales, con los que poder crear distintas acciones y de esa forma interactuar en el mundo virtual. Del mismo modo también posee los llamado prefabs o modelos ya creados con sus físicas y características como lo es el de la cámara de Realidad Virtual, que posee visión de 360° con la que interaccionar [18].

Incluso en el proceso de programación dispone de objetos con los que poder acceder a todas las características y parámetros de los mandos y gafas para poder proporcionar las funcionalidades requeridas.

2.4.1 APIs Mapas VR

En el caso de este proyecto al tratarse a grandes rasgos de una aplicación de exploración de mapas en Realidad Virtual e interacción con el mismo, aunque posteriormente se explicará con más detalle cada una de las funcionalidades de esta, es necesario el uso de una librería que permita esa creación e interacción con el mapa.

Hay distintas alternativas que pueden ser válidas como solución y que en general trabajan de una forma parecida, se les proporciona una latitud y una longitud de una localización cualquiera y la librería con la ayuda del motor de desarrollo utilizado creará un modelo 3D simulando a la realidad con los edificios y calles debidas.

Algunos ejemplos son los siguientes:

- **Online Maps de Infinity Code.** Consiste en un asset de la tienda de Unity, es decir una extensión del propio motor de desarrollo en este caso de pago, alrededor de 80 dólares. Es una solución para todo tipo de plataformas que crea mapas ya sean en 2D o 3D que se pueden customizar al gusto y posee una de las APIs más potentes [19].
- **Maps SDK para Unity.** Se trata de un conjunto de herramientas y recursos de desarrollo que de igual forma que el caso anterior el entorno de desarrollo de Unity. Este SDK proporciona los datos geográficos de la base de datos de Google Maps, generando de ese modo los edificios, monumentos o carreteras en función de la ubicación del usuario o la que se haya dado antes de comenzar el juego [20].
- **Mapbox.** Es una solución de código abierto para la creación de mapas en 2D o 3D y con la que se pueden crear aplicaciones Android como si de un navegador se tratara. Del mismo modo existe una extensión para Unity lo cual podría servir de ayuda a la hora del desarrollo. La forma de trabajar es similar a los casos anteriores, crea modelo 3D a partir de una localización, pero lo que le diferencia aparte de dar la posibilidad de configurar totalmente la información que sale en el mapa (quitar nombres de calles, lugares, puntos de interés) también proporciona la opción de crear rutas a partir de puntos que el usuario pueda crear durante el proceso de juego y esta librería crea la ruta más corta que conecta los debidos puntos [21].

- **ArcGIS Maps SDK.** Al igual que las opciones anteriores se trata de una extensión para Unity que proporciona acceso a mapas reales y contenido en 3D creando componentes simulando la realidad y proporcionando experiencias más realistas. También se da la opción de creación de rutas dentro del mapa. El único inconveniente que puede dar esta alternativa es que al tratarse de mapas con mayor realismo esto cree problemas de rendimiento en la aplicación y con ello no se pueda interactuar del mismo modo con la aplicación [22].

Debe haber un equilibrio entre las prestaciones que proporciona la API que acabe sirviendo como solución y las que quite a su vez, es decir la aplicación no puede consistir únicamente en la visualización de mapas en Realidad Virtual y que no contenga aspectos diferenciales como las distintas formas de interacción con mandos, manos, movimientos o sensores y a su vez con la creación de aspectos dentro del propio entorno.

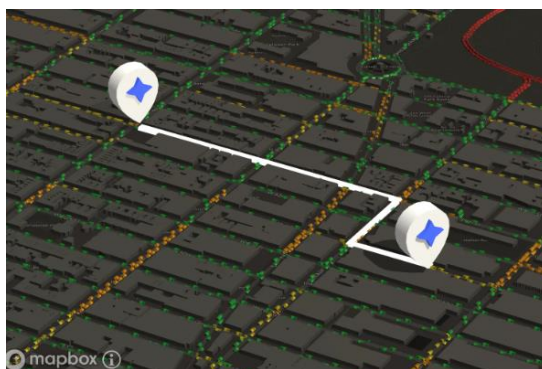


Fig. 2.7 Imagen de ruta creada por Mapbox entre dos puntos [23]

Con la funcionalidad que se quiere proporcionar a la aplicación, puede ser un aspecto diferencial que el usuario pueda crear sus puntos de interés y rutas y de esa forma poder interactuar de distintas maneras con esa información.

2.5 Aplicaciones similares

Para poder saber en qué lugar se encuentra la aplicación que se está desarrollando, se debe investigar qué otras soluciones o ejemplos similares hay. Con esto se podrá confeccionar este proyecto con distintas características que le hagan diferencial respecto a los demás. Los proyectos analizados deberán tener interacción y visualización de mapas en Realidad Virtual y poder realizar el análisis y cambio de información dentro del entorno visual que sea inmersivo y que responda totalmente al usuario.

- **Visualización VR y exploración de sonidos urbanos** [24]

Este proyecto como su nombre indica proporciona la posibilidad de visualización de mapas en 3D y Realidad Virtual y a su vez explorar los sonidos urbanos que se producen en la realidad. Más en concreto con el sonido del tráfico que puede ocurrir en la realidad

y que con esto se pueda notificar a las debidas autoridades de los problemas de salud, seguridad y calidad de vida que surgen de estos ruidos.

La forma de interacción con el entorno virtual consiste en el movimiento ya sea con los mandos o a través de un puntero que teletransporta al usuario donde esté apuntando y con ello poder moverse donde desee. La visualización de los datos de sonido ocurre ya sea con el propio sonido que se puede escuchar a través de los auriculares o con la visión de puntos o líneas de distinto color que indican el nivel de ruido.

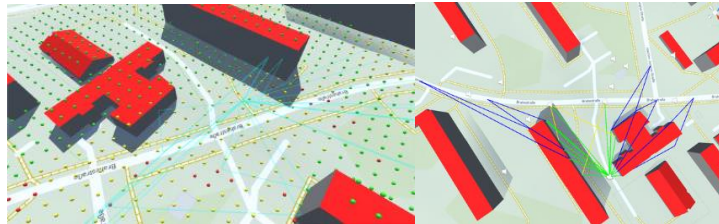


Fig. 2.8 Imagen de los datos de sonido

Como se puede ver a su vez tiene diferentes formas de encontrarse el usuario en el mapa, posición real en el suelo, vista de pájaro, etc.

Las formas de interactuar con el mapa y los datos no son muy novedosas, pero sí se centra en esa exploración de los datos del sonido con su debido análisis.

- **Google Earth VR [25]**

En este caso es una solución creada por el propio Google y similar a su producto ya conocido por todo el mundo Google Earth o también conocido como Google Maps, en el que se puede explorar cualquier punto del mundo completamente real. Pues las imágenes están tomadas por satélite y son fieles a la realidad, la única novedad es que está adaptado al movimiento y visualización con dispositivos de Realidad Virtual.

La aplicación solo permite la exploración del terreno a distintas alturas como si se estuviera volando y poder incluso escoger a dónde viajar pulsando en la bola del mundo disponible, como la mayoría de la gente ha fantaseado alguna vez en su vida.

Es un gran competidor pues pertenece a una de las compañías más grandes del sector tecnológico como es Google, con todos los avances e innovación que eso supone, además de ser una solución cuyos gráficos son a grandes rasgos imposibles de superar, fieles a la realidad.

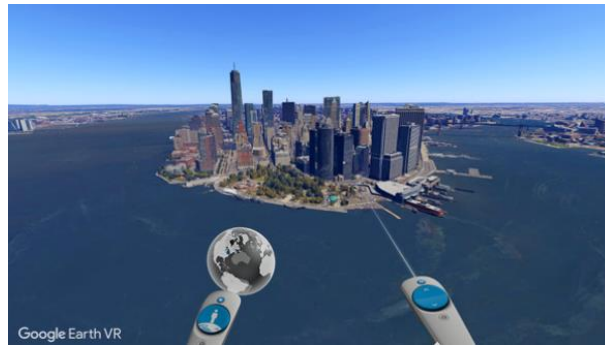


Fig. 2.9 Imagen dentro del juego Google Earth VR

Uno de los inconvenientes en este caso y en lo que se va a centrar en gran parte este proyecto, es en las distintas formas de interacción e información por ejemplo de monumentos o puntos de interés que se pueden ofrecer al usuario y así hacerlo diferencial de alguno de estos casos. En concreto la solución de Google solo permite la exploración de mapas con un par de formas de interacción.

- **Wander** [26]

Wander consiste en una aplicación disponible para los dispositivos de Oculus y simula el estar viajando alrededor del mundo. La principal función es la de explorar las distintas localizaciones deseadas dando la posibilidad incluso de leer información de un punto de interés como por ejemplo la torre Eiffel en París.

El inconveniente principal es que no se trata del consumo de un modelo 3D creado para la Realidad Virtual, sino que lo que se ve realmente es igual al Street View que proporciona por ejemplo Google Maps, moviéndose incluso como en este por la carretera y demás. También ofrece el poder teclear directamente la dirección a la que ir y viajar directamente al destino.

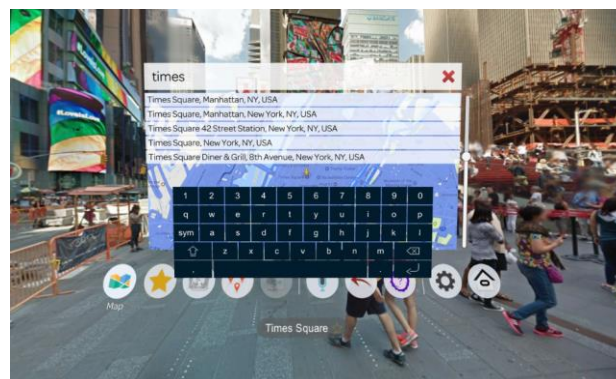


Fig. 2.10 Imagen de la aplicación Wander para Oculus

En conclusión, esta solución a pesar de ser llamativa en cuanto al poder viajar e inspeccionar lugares reales de una forma virtual, no proporciona la suficiente inmersión

y presencia que una aplicación de Realidad Virtual puede tener y hacia donde se está dirigiendo en la actualidad, de igual forma con la interacción como si se tratase de un teclado o juego de ordenador.

2.6 Dispositivos de Entrada/Salida

Para la interacción con los sistemas de Realidad Virtual y el entorno que se crea se necesitan diversos dispositivos para que esto sea posible. Dentro de estos se encuentran obviamente las gafas, la parte más importante que ofrece toda la inmersión en el mundo virtual y que a su vez como se ha explicado en su apartado correspondiente tienen sensores para detectar los movimientos de la cabeza e incluso gestos con las manos.

Algunas de las alternativas de dispositivos para la interacción con sistemas de Realidad Virtual son los siguientes:

- **Mandos.** Uno de los dispositivos que obviamente tienen que aparecer son los mandos de juego que vienen incluidos cuando se realiza la adquisición de unas gafas y son el principal sistema de input. Dependiendo del modelo tienen una forma u otra, pero en general disponen tanto de joysticks para poder moverse como si de un mando de consola tradicional se trata como botones. Lo diferencial es que disponen de sensores tanto para los dedos como en la parte posterior del mando, los primeros para detectar el movimiento de los dedos en los botones sin haber realizado la pulsación completa y los segundos para que el funcione como un puntero en el entorno virtual y poder pulsar cualquier elemento haciendo click en el mismo.
- **Sensores Antilatency.** Como ya se ha mencionado en un apartado anterior (apartado 1.4) son sensores compatibles con el desarrollo en Unity y la Realidad Virtual que pueden servir como dispositivo de entrada, incluso salida con vibraciones y demás. Son sensores que se pueden adherir a alguna parte del cuerpo del usuario para hacer el seguimiento de movimientos de brazos o piernas, y a su vez pegarlo a algún objeto y hacer que en el entorno virtual se vea por ejemplo una pelota, una llave, para hacerlo más realista.

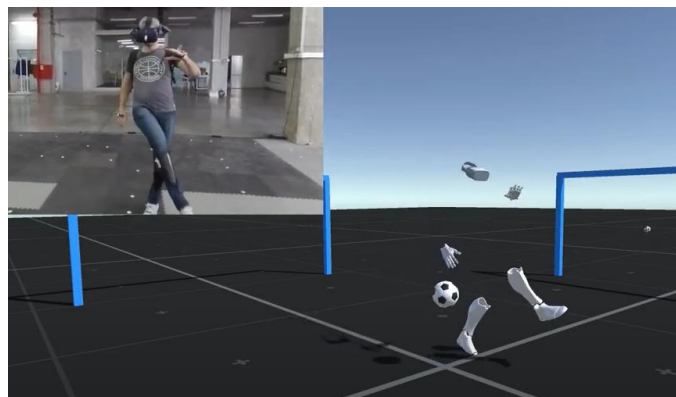


Fig. 2.11 Imagen de ejemplo de uso sensores Antilatency [27]

Como se puede ver en la imagen un caso de uso puede ser el tener sensores en ambas manos que con la ayuda de una especie de pulsera también detectan presión, otros en las piernas y con ello simular que alguien está jugando al fútbol. Este proceso es similar a cómo se hacen las capturas de movimiento para que posteriormente en el videojuego el movimiento de los personajes sea lo más real posible.

- **Omnidirectional Treadmill.** Aunque su nombre pueda no llamar demasiado la atención, estos tipos de dispositivos son el presente y el futuro de la Realidad Virtual. Se trata de una especie de cinta 360° es decir que permite moverse en cualquier dirección, de igual modo también poseen módulos de vibración integrados en las cintas con los que poder simular disparos, explosiones y cualquier tipo de vibración en el mundo real. Tienen un mecanismo en el que el usuario va enganchado y no supone ningún problema de seguridad, con esto se consigue una mayor sensación de inmersión comparado con otros dispositivos más tradicionales.



Fig. 2.12 Imagen de la KAT Walk C2 Treadmill [28]

A pesar del gran avance que supone, por el momento son dispositivos de difícil acceso para el público en general, ya que en primer lugar su precio es muy elevado y la oferta tampoco es muy amplia. Pero hacia posibles mejoras en el futuro en las aplicaciones de Realidad Virtual son un gran factor a tener en cuenta [28].

2.7 Conclusiones

Teniendo en cuenta la visión expuesta en esta sección, las conclusiones en cuanto a las alternativas que se van a utilizar en el desarrollo del proyecto son las siguientes.

En primer lugar, el dispositivo de Realidad Virtual que se va a utilizar va a ser las gafas Oculus Quest 2, en comparativa con las demás son las que presentan un mayor equilibrio entre sus características, ofrecen la inmersión necesaria con unos gráficos y calidad

buenos y sobre todo su precio no es muy elevado. También han sido escogidas debido a que ya se poseían por parte de la universidad y del tutor, lo cual será de gran ayuda a la hora del desarrollo.

En cuanto al motor de desarrollo escogido como se ha mencionado, Unity es el que ofrece un mayor contenido de terceros y sobre todo aprendizaje en internet, primordial para aprender por primera vez el uso de estos tipos de aplicaciones. A su vez Unity dispone de un abanico muy amplio de extensiones que supondrán una gran ayuda a la hora de la creación de una aplicación de Realidad Virtual.

La aplicación al tratarse de una solución para la exploración de mapas en 3D y Realidad Virtual, la API correspondiente que se vaya a utilizar es importante pues gran parte del desarrollo dependerá de esta. Tras la comparación de todas, Mapbox es la que puede dar las opciones necesarias para hacer que la aplicación pueda diferenciarse de otras. Aparte como se mencionó es una extensión de Unity, de la cual hay documentación online por parte incluso del propio Mapbox. Con su posibilidad de poder configurar lo que aparece en el mapa, retocar edificios, poner puntos de interés, crear rutas de distintas longitudes, etc.

Por último, tras analizar las aplicaciones similares que hay en el mercado, cabe destacar que este proyecto puede ser diferencial respecto a los otros. Es cierto que hay alguna opción de las analizadas que pueden tener mejores gráficos y más realistas como es el caso de Google Earth VR, pero con la solución que se intenta llegar en este proyecto es conseguir el equilibrio entre unos gráficos que a pesar de no ser tan realistas sí se consiga esa inmersión mencionada a lo largo del trabajo y que el usuario sepa reconocer donde se encuentra y poder explorar el entorno virtual. Todo ello con las distintas formas de interacción y movimiento que se pueden ofrecer que aparte de con los mandos, el usuario pueda moverse realizando gestos manuales, teletransportarse apuntando a una especie de mapa en miniatura o incluso con la ayuda de sensores adicionales hacer el movimiento de andar y que en el entorno virtual se avance donde se quiera.

Es claro ver en lo que este producto se diferencia de los otros con lo ya descrito, pero en apartados posteriores quedará más detallado con ejemplo visuales y explicaciones más extensas de las formas de interacción e información que aparece y está disponible dentro de la aplicación.

3. ANÁLISIS

El primer paso en el desarrollo de un proyecto software es en el análisis de la aplicación y la especificación de los requisitos que esa misma debe cumplir. Por lo tanto, en este apartado se va a explicar el funcionamiento del sistema que se va a desarrollar, con las aclaraciones precisas de las funcionalidades y restricciones.

Primeramente, se hará una descripción a grandes rasgos de la aplicación que se está desarrollando, qué puede ofrecer el sistema al usuario y cómo puede responder el mismo. Tras esta explicación ya empezará la especificación en primer lugar de los requisitos de usuario, con los que se define el alcance del sistema y un análisis del entorno en el que se puede utilizar.

Tras el apartado anterior se enumerarán los requisitos software, que se diferencian en que los de usuario especifican lo que el usuario puede hacer y los de software lo que hace el sistema respectivamente, se profundiza en mayor parte en aspectos técnicos. Son útiles también para definir posteriormente la arquitectura del sistema.

También se nombrarán los casos de uso a partir de los requisitos de usuario para que este sepa claramente lo que la aplicación puede realizar. Y por último se realizará un análisis con matrices de trazabilidad para ver qué requisitos tiene interacción entre sí.

3.1 Descripción de la aplicación

Se trata de una aplicación de visualización, exploración e interacción con mapas 3D en Realidad Virtual. En esta, el usuario se conecta a través de sus gafas de Oculus (Quest 2) y se ve inmerso en un entorno 3D que simula una ciudad con sus edificios y calles.

El usuario tiene distintas formas de interactuar con el entorno, por ejemplo, para moverse en cualquier dirección puede utilizar tanto los mandos que vienen con las gafas como si de un juego más común se tratara, a hacer el gesto de apuntar y moverse en esa dirección o incluso con la ayuda de sensores hacer el gesto con las piernas de andar, pero quedándose en el sitio y moverse en el entorno en esa dirección (lo que se conoce como "walk in-place").

El usuario tiene la capacidad de explorar puntos de interés marcados y leer una breve descripción de estos. A su vez, el usuario es capaz de poner puntos de interés donde crea conveniente también de distintas formas, con la ayuda de un menú que parece ir incorporado en el brazo dentro del entorno, dejando en el suelo un objeto con un sensor, o con los mandos de juego.

De igual modo hay distintas formas en las que el usuario puede ver el mapa, a la altura del suelo como si este se encontrara andando por las calles de la ciudad lo cual

proporciona una mayor sensación de inmersión, realizando un gesto pasar al modo “vuelo” o vista de pájaro y ver el mapa desde una parte más elevada lo cual puede dar una visión más general del entorno. Otra forma de ver el mapa puede ser como si de una maqueta se tratara, el usuario puede ver el mapa en miniatura y con un puntero disponible en los mandos de juego apuntar a donde desee y posteriormente aparecer a la altura del suelo en esa localización.

Por último, también cabe la posibilidad que quien se encuentre dentro del entorno quiera viajar a otra ciudad disponible en la aplicación, a través del menú incorporado en el brazo podrá encontrarse en ciudades como Madrid, París o Nueva York.

Con los siguientes ejemplos de tareas de usuario se va a poder entender mejor el fin de la aplicación:

- El usuario entra en la aplicación para explorar y visualizar los puntos de interés de una ciudad a la que por ejemplo tiene pensado viajar en un futuro, con el uso de esta puede ver qué lugares parecen más interesantes y crear una ruta propia a lo largo del mapa con distinta información. Las diferentes interacciones hacen que el usuario se vea atraído por esa innovación o características diferenciales y quiera realizar el uso de esta aplicación y no otra.
- Otro posible ejemplo derivado del anterior es que empresas de turismo y organización de viajes puedan ofrecer el servicio y que no sea solo de uso individual del usuario con su propio dispositivo. De nuevo es un aspecto que hará diferenciable la competencia respecto a otras empresas.
- Diferentes empresas de organización de eventos como carreras de Fórmula 1, ciclismo, maratones, etc, pueden utilizar la aplicación tanto para crear las rutas por donde van a pasar las distintas etapas de una manera más inmersiva y visual, como para presentárselo a patrocinadores que quieran unirse al evento. Es decir puede servir de herramienta diferencial para presentar estos eventos y llamar la atención tanto de los organizadores como los diferentes participantes.

3.2 Requisitos de usuario

En este apartado se definen tanto las capacidades como las restricciones del sistema y también un análisis de las características de los usuarios potenciales de la aplicación, aspectos del entorno operativo y de seguridad como suposiciones de las que partir antes de comenzar a utilizar el producto.

En primer lugar, se realizará el análisis de dichas características de usuario y suposiciones y posteriormente ya se irán nombrando cada uno de los requisitos.

3.2.1 Características del usuario

La aplicación principalmente está dirigida para usuarios que estén familiarizados con la tecnología y sean capaces de interactuar con entornos virtuales y quieran disponer de una aplicación en la que poder visualizar mapas en Realidad Virtual. Con la que explorar ciudades y organizar viajes futuros a dichos destinos, creando rutas, analizando puntos de interés, etc.

Al mismo tiempo pueden hacer uso de la aplicación empresas de turismo u organización de eventos para que ofrezcan a posibles clientes un elemento diferencial que llame la atención y los elijan a la hora de contratar dichos servicios. Para ello las empresas deberán proporcionar los mecanismos suficientes para que los usuarios puedan aprender a utilizar estos dispositivos y quieran repetir con estos sistemas.

3.2.2 Entorno operativo y de seguridad

Para el funcionamiento del sistema se necesitan las gafas Oculus Quest 2 con sus mandos y también del uso de los sensores Antilatency tanto en las piernas para simular el movimiento de andar, alguno en una pequeña alfombra y en el objeto requerido para poder poner puntos de interés y simular como si se estuviera poniendo una marca en el suelo.

La aplicación se puede utilizar directamente en las gafas sin la necesidad de que el ordenador esté conectado por cable, pero previamente sí que se tiene que haber hecho la configuración y haber pasado el archivo a las gafas.

En cuanto a la seguridad, como no se necesita el uso de credenciales, la aplicación no tiene responsabilidad respecto a esto. Solo en el caso de querer utilizar la aplicación y querer ver lo que aparece en las gafas en un ordenador habrá que aceptar el acceso al almacenamiento del dispositivo, pero es un aspecto propio del entorno del dispositivo y el ordenador, no de la aplicación en específico.

3.2.3 Suposiciones

Para realizar el análisis y el uso de la aplicación se asumen distintos aspectos:

- Se debe disponer de un dispositivo de Realidad Virtual compatible con Oculus, más en concreto las Oculus Quest 2.
- Será necesaria la disposición de distintos sensores tanto en las piernas del posible usuario, como en una pequeña alfombra en el suelo y en un objeto cualquiera.
- No se dispone de una guía de uso de dispositivos de Realidad Virtual.

- La aplicación se ejecutará en un entorno cerrado y sin posibles obstáculos que puedan ser un peligro de cara a los usuarios de la aplicación.

3.2.4 Requisitos específicos

A continuación, se van a detallar el conjunto de requisitos de usuario, pero previamente para que la identificación sea más simple se puede distinguir dos.

Los requisitos de capacidad definen las características que debe poseer el sistema para adecuarse a lo que el usuario necesita. Se consigue lograr un objetivo.

Los requisitos de restricción son las limitaciones que condicionan cómo se va a llegar a ese objetivo. Puede estar asociado a la implementación o rendimiento del sistema.

Todos los requisitos van a seguir un esquema similar con sus propiedades, como el siguiente:

- **Identificador.** Cada requisito tiene un identificador que lo diferencia del resto. Será del tipo RU-YX.
 - RU: indica requisito de usuario.
 - Y: indica el tipo de requisito. Puede tener el valor C si se trata de un requisito de capacidad o R si es uno de restricción.
 - X: es el número que identifica el requisito dentro de cada tipo. Son números arábigos y se van incrementando por unidades.
- **Nombre.** Nombre que además proporcione un resumen claro de qué trata el requisito.
- **Prioridad.** Indica la preferencia respecto a su realización. Puede ser alta, media o baja, esto se tendrá en cuenta los de mayor prioridad irán primeros.
- **Verificabilidad.** Indica la facilidad con la que se puede comprobar que ese requisito se encuentra desarrollado. Será alta si es fácil de comprobar o baja si es más complicado. La clave es crear requisitos que sean altamente verificables.
- **Descripción.** Explicación detallada de en qué consiste el requisito.

Para la definición de los requisitos se va a utilizar una tabla como la siguiente:

TABLA 3.1: PLANTILLA DE REQUISITOS DE USUARIO

Identificador	
Nombre	
Prioridad	
Verificabilidad	

Descripción	
--------------------	--

3.2.4.1 Requisitos de capacidad

En el siguiente apartado se enumeran los distintos requisitos de usuario pertenecientes al grupo de capacidad, es decir lo que el usuario puede hacer.

TABLA 3.2 REQUISITO DE USUARIO RU-C1

RU-C1	
Nombre	Visualización del mapa
Prioridad	Alta
Verificabilidad	Alta
Descripción	El usuario es capaz de visualizar, navegar e interactuar con un mapa en 3 dimensiones.

TABLA 3.3 REQUISITO DE USUARIO RU-C2

RU-C2	
Nombre	Vista de humano
Prioridad	Alta
Verificabilidad	Alta
Descripción	El usuario puede ver y navegar por el entorno y el mapa como si estuviera andando en la realidad, es decir a la altura del suelo.

TABLA 3.4 REQUISITO DE USUARIO RU-C3

RU-C3	
Nombre	Vista de pájaro
Prioridad	Alta

Verificabilidad	Alta
Descripción	El usuario puede ver y navegar por el mapa estando a una altura más elevada como si de un pájaro se tratara. Los elementos del mapa tienen menos detalle.

TABLA 3.5 REQUISITO DE USUARIO RU-C4

RU-C4	
Nombre	Maqueta del mapa
Prioridad	Alta
Verificabilidad	Alta
Descripción	El usuario es capaz de entrar en un modo en el que mapa parece una maqueta y ver todos sus elementos en miniatura.

TABLA 3.6 REQUISITO DE USUARIO RU-C5

RU-C5	
Nombre	Movimiento del usuario
Prioridad	Alta
Verificabilidad	Alta
Descripción	Tanto en la vista de humano como de pájaro el usuario es capaz de moverse con total libertad por el mapa.

TABLA 3.7 REQUISITO DE USUARIO RU-C6

RU-C6	
Nombre	Abrir punto de información
Prioridad	Media
Verificabilidad	Alta

Descripción	A lo largo del mapa se disponen de distintos puntos de información que el usuario puede pulsar y desplegar la información del punto de interés.
--------------------	---

TABLA 3.8 REQUISITO DE USUARIO RU-C7

RU-C7	
Nombre	Interacción con mandos
Prioridad	Alta
Verificabilidad	Alta
Descripción	El usuario puede interaccionar con el mapa y todos sus elementos con el uso de los mandos Oculus Touch.

TABLA 3.9 REQUISITO DE USUARIO RU-C8

RU-C8	
Nombre	Interacción con reconocimiento de manos
Prioridad	Alta
Verificabilidad	Alta
Descripción	El usuario de igual forma puede interaccionar con el entorno a través de gestos con las manos, gracias al reconocimiento proporcionado por Oculus

TABLA 3.10 REQUISITO DE USUARIO RU-C9

RU-C9	
Nombre	Interacción con sensores Antilatency
Prioridad	Alta
Verificabilidad	Alta

Descripción	El usuario puede moverse por el entorno y realizar acciones a través del reconocimiento de distancia y movimiento de los sensores Antilatency.
--------------------	--

TABLA 3.11 REQUISITO DE USUARIO RU-C10

RU-C10	
Nombre	Poner marcador
Prioridad	Alta
Verificabilidad	Alta
Descripción	El usuario tiene la posibilidad de crear un marcador en la localización que desee.

TABLA 3.12 REQUISITO DE USUARIO RU-C11

RU-C11	
Nombre	Eliminar marcador
Prioridad	Alta
Verificabilidad	Alta
Descripción	El usuario puede eliminar un marcador o punto de interés que haya creado.

TABLA 3.13 REQUISITO DE USUARIO RU-C12

RU-C12	
Nombre	Eliminar conjunto de marcadores
Prioridad	Media
Verificabilidad	Alta
Descripción	Si dentro del entorno virtual el usuario ha creado dos o más marcadores podrá eliminarlos todos a la vez.

TABLA 3.14 REQUISITO DE USUARIO RU-C13

RU-C13	
Nombre	Visualización de ruta
Prioridad	Alta
Verificabilidad	Alta
Descripción	Cuando el usuario crea dos o más marcadores y estos están espaciados, se crea una ruta amarilla que los une y busca el camino más corto entre ellos.

TABLA 3.15 REQUISITO DE USUARIO RU-C14

RU-C14	
Nombre	Menú en el brazo
Prioridad	Media
Verificabilidad	Media
Descripción	Si el usuario está con la forma de interacción de reconocimiento de manos, si se mira la palma izquierda abierta podrá ver un menú con distintas opciones.

TABLA 3.16 REQUISITO DE USUARIO RU-C15

RU-C15	
Nombre	Menú información de ruta
Prioridad	Media
Verificabilidad	Media
Descripción	Se dispone de un menú en el que aparece información de la ruta que se ha creado, distancia y número de marcadores.

TABLA 3.17 REQUISITO DE USUARIO RU-C16

RU-C16	
Nombre	Teletransportación de la maqueta
Prioridad	Media
Verificabilidad	Alta
Descripción	Si el usuario se encuentra en el modo del mapa de maqueta, a través del mando derecho y una especie de puntero puede apuntar a la parte de la maqueta y aparecer en dicha localización en el mapa real en la vista de humano.

TABLA 3.18 REQUISITO DE USUARIO RU-C17

RU-C17	
Nombre	Viaje a otras ciudades
Prioridad	Media
Verificabilidad	Media
Descripción	En el menú incorporado en el brazo hay una opción de ciudades en la que pulsando se despliega tres distintas ciudades a las que poder viajar.

3.2.4.2 Requisitos de restricción

En este apartado constan los requisitos de restricción que se cumplen en el desarrollo del sistema.

TABLA 3.19 REQUISITO DE USUARIO RU-R1

RU-R1	
Nombre	Aplicación de Realidad Virtual

Prioridad	Alta
Verificabilidad	Alta
Descripción	La aplicación debe ser compatible con la tecnología de visualización de Realidad Virtual, con las gafas correspondientes.

TABLA 3.20 REQUISITO DE USUARIO RU-R2

RU-R2	
Nombre	Motor de desarrollo
Prioridad	Alta
Verificabilidad	Alta
Descripción	El desarrollo de la aplicación se realizará utilizando el motor de desarrollo Unity.

TABLA 3.21 REQUISITO DE USUARIO RU-R3

RU-R3	
Nombre	Compatibilidad Oculus
Prioridad	Alta
Verificabilidad	Alta
Descripción	La aplicación debe ser compatible con las gafas Oculus Quest 2.

TABLA 3.22 REQUISITO DE USUARIO RU-R4

RU-R4	
Nombre	Controles dentro de la aplicación
Prioridad	Alta
Verificabilidad	Alta

Descripción	La interacción del usuario y la aplicación deberá realizarse mezclando controles con las Oculus Touch, seguimiento de manos y sensores Antilatency.
--------------------	---

TABLA 3.23 REQUISITO DE USUARIO RU-R5

RU-R5	
Nombre	Eliminar marcador
Prioridad	Media
Verificabilidad	Media
Descripción	El usuario solo puede borrar un marcador de forma individual si se encuentra a una distancia inferior a 10 metros de este.

3.3 Requisitos de software

Una vez explicados los requisitos de usuario se puede seguir profundizando en las capacidades del sistema, realizando los requisitos de software. Establecen qué es lo que puede ofrecer el sistema, son más detallados y contemplan todas las posibilidades que hay por ejemplo para una misma acción.

De igual modo que los de usuario se pueden dividir en distintos grupos para que quede mucho más claro:

- **Requisitos funcionales (F).** Se especifican las capacidades del software que se va a desarrollar. Tienen relación con los requisitos de capacidad enumerados anteriormente.
- **Requisitos operacionales (O).** Se indica cómo el sistema realiza las distintas tareas que es capaz de hacer.
- **Requisitos de interfaz (I).** Especifica los componentes hardware o software con los que el sistema debe interactuar.
- **Requisitos de usabilidad (U).** Son los requisitos relativos a la usabilidad del sistema, si es fácil de cara al usuario.

Los requisitos que se van a explicar a continuación tienen unas propiedades similares a los requisitos de usuario y son las siguientes:

- **Identificador.** Cada requisito dispone de un identificador único de la forma RS-YX. Siendo cada una de las letras:
 - RS: significa que se trata de un requisito de software.
 - Y: indica el tipo de requisito. Como cada uno de los que se ha explicado en el párrafo anterior, tomando Y el valor de cada una de las letras correspondientes.
 - X: indica el número dentro de cada tipo. Es único y va aumentado en unidades para cada uno.
- **Nombre.** Resumen claro de lo que el requisito se refiere.
- **Relación.** Propiedad para indicar con qué requisito de usuario tiene relación, pues en su mayoría deben tener relación. Será el identificador correspondiente.
- **Prioridad.** Es la preferencia de realización de ese requisito. Pudiendo ser alta, media o baja.
- **Verificabilidad.** Facilidad con la que se puede verificar si el requisito está implementado en el sistema o no. Con valores de alta, media o baja, siempre tratando de buscar los altamente verificables.
- **Descripción.** Explicación detallada de en qué consiste el requisito.

La forma que tendrán cada una de las tablas de requisitos será:

TABLA 3.24: PLANTILLA DE REQUISITOS DE SOFTWARE

Identificador			
Nombre			
Prioridad		Relación	
Verificabilidad			
Descripción			

3.3.1 Requisitos funcionales

TABLA 3.25: REQUISITO DE SOFTWARE RS-F1

RS-F1			
Nombre	Visualización del mapa		
Prioridad	Alta	Relación	RU-C1
Verificabilidad	Alta		

Descripción	El sistema permite al usuario la exploración de un mapa tridimensional.
--------------------	---

TABLA 3.26: REQUISITO DE SOFTWARE RS-F2

RS-F2			
Nombre	Vista de humano		
Prioridad	Alta	Relación	RU-C2
Verificabilidad	Alta		
Descripción	El sistema permite visualizar el desde el punto de vista a pie del suelo, similar al ir andando con normalidad por la calle.		

TABLA 3.27: REQUISITO DE SOFTWARE RS-F3

RS-F3			
Nombre	Gesto "Thumbs Up"		
Prioridad	Alta	Relación	RU-C2, RU-C3
Verificabilidad	Alta		
Descripción	El sistema ofrece la posibilidad de cambiar de la vista de humano a la de pájaro realizando un gesto con la mano izquierda de "Thumbs Up" o pulgar hacia arriba.		

TABLA 3.28: REQUISITO DE SOFTWARE RS-F4

RS-F4			
Nombre	Gesto brazo hacia arriba		
Prioridad	Media	Relación	RU-C2, RU-C3
Verificabilidad	Alta		
Descripción	Otra opción proporcionada por el sistema es la de cambiar de vista de humano a la de pájaro subiendo el brazo derecho hacia		

	arriba, ya sea sujetando el Oculus Touch correspondiente o con un sensor en la mano derecha.
--	--

TABLA 3.29: REQUISITO DE SOFTWARE RS-F5

RS-F5			
Nombre	Botón cambio vista		
Prioridad	Media	Relación	RU-C2, RU-C3
Verificabilidad	Alta		
Descripción	El sistema permite al usuario poder cambiar a la vista de pájaro pulsando el gatillo posterior del mando izquierdo de las Oculus Touch.		

TABLA 3.30: REQUISITO DE SOFTWARE RS-F6

RS-F6			
Nombre	Gesto "Thumbs down"		
Prioridad	Alta	Relación	RU-C3, RU-C2
Verificabilidad	Alta		
Descripción	Estando en la vista de pájaro se puede cambiar a la vista de suelo o humano realizando con la mano izquierda un gesto de pulgar hacia abajo.		

TABLA 3.31: REQUISITO DE SOFTWARE RS-F7

RS-F7			
Nombre	Botón mapa en maqueta		
Prioridad	Media	Relación	RU-C3, RU-C4
Verificabilidad	Alta		
Descripción	El sistema permite al usuario entrar en un modo de vista en la que el mapa se convierte en una maqueta, si el usuario está		

	previamente en la vista de pájaro y pulsa el gatillo lateral del mando derecho.
--	---

TABLA 3.32: REQUISITO DE SOFTWARE RS-F8

RS-F8			
Nombre	Movimiento con mandos		
Prioridad	Alta	Relación	RU-C5, RU-C2, RU-C3, RU-C7
Verificabilidad	Alta		
Descripción	El movimiento dentro del entorno virtual puede realizarse con los mandos Oculus Touch y sus joysticks, siendo el izquierdo para moverse atrás y delante y el derecho para rotar la cámara.		

TABLA 3.33: REQUISITO DE SOFTWARE RS-F9

RS-F9			
Nombre	Movimiento gesto apuntando hacia delante		
Prioridad	Alta	Relación	RU-C5, RU-C2, RU-C3, RU-C8
Verificabilidad	Alta		
Descripción	Otra forma de interactuar con el movimiento proporcionada por el sistema es mediante un gesto con la mano derecha de apuntar hacia delante con el dedo índice, la cámara se moverá en la dirección que se esté mirando con las gafas.		

TABLA 3.34: REQUISITO DE SOFTWARE RS-F10

RS-F10			
Nombre	Gesto apuntando con dos dedos		
Prioridad	Media	Relación	RU-C5, RU-C2, RU-C3, RU-C8
Verificabilidad	Alta		
Descripción	Al igual que en el requisito anterior, pero en este caso apuntando con el índice y el corazón el movimiento será el mismo, pero más rápido.		

TABLA 3.35: REQUISITO DE SOFTWARE RS-F11

RS-F11			
Nombre	Gesto de andar		
Prioridad	Alta	Relación	RU-C5, RU-C9
Verificabilidad	Alta		
Descripción	Con la ayuda de los sensores en las piernas del usuario y en la alfombra, el sistema puede registrar el movimiento de las piernas como si se estuviera andando en el sitio y en el entorno virtual la cámara se moverá en ese sentido.		

TABLA 3.36: REQUISITO DE SOFTWARE RS-F12

RS-F12			
Nombre	Abrir punto de información		
Prioridad	Media	Relación	RU-C6
Verificabilidad	Alta		
Descripción	El sistema permite abrir los puntos de información con gestos manuales y los mandos, apuntando con el puntero que sale de		

	las manos y haciendo el movimiento de pinza para pulsar el botón con la i. O en el caso de los mandos pulsando el botón A.
--	--

TABLA 3.37: REQUISITO DE SOFTWARE RS-F13

RS-F13			
Nombre	Cerrar punto de información		
Prioridad	Media	Relación	RU-C6
Verificabilidad	Alta		
Descripción	Se pueden cerrar los módulos de información de la misma forma en la que lo ha abierto, pero pulsando el botón de cerrar.		

TABLA 3.38: REQUISITO DE SOFTWARE RS-F14

RS-F14			
Nombre	Soltar objeto adicional en el suelo		
Prioridad	Alta	Relación	RU-C9, RU-C10
Verificabilidad	Alta		
Descripción	El sistema permite poner un marcador en el entorno virtual, cuando el usuario realiza el gesto de dejar en el suelo el objeto adicional con el sensor en el suelo.		

TABLA 3.39: REQUISITO DE SOFTWARE RS-F15

RS-F15			
Nombre	Botón poner marcador		
Prioridad	Media	Relación	RU-C10, RU-C7
Verificabilidad	Alta		
Descripción	En el mando izquierdo de Oculus Touch, el botón Y es otra opción para poner un marcador en el entorno virtual.		

TABLA 3.40: REQUISITO DE SOFTWARE RS-F16

RS-F16			
Nombre	Poner marcador con menú del brazo		
Prioridad	Alta	Relación	RU-C10, RU-C14, RU-C8
Verificabilidad	Alta		
Descripción	En el menú disponible en el entorno virtual al mirarse el brazo izquierdo con la palma abierta hay un botón para poner un marcador si el usuario así lo desea.		

TABLA 3.41: REQUISITO DE SOFTWARE RS-F17

RS-F17			
Nombre	Botón eliminar marcador individual		
Prioridad	Media	Relación	RU-C11, RU-C7
Verificabilidad	Alta		
Descripción	El sistema ofrece al usuario poder eliminar un marcador creado por él mismo si se encuentra a una distancia inferior a 10 metros de este, pulsando el botón X del Oculus Touch izquierdo.		

TABLA 3.42: REQUISITO DE SOFTWARE RS-F18

RS-F18			
Nombre	Eliminar marcador individual con menú del brazo		
Prioridad	Media	Relación	RU-C11, RU-C14, RU-C8
Verificabilidad	Alta		
Descripción	Es el mismo formato que el requisito anterior, pero en este caso se dispone de un botón en el menú incorporado del brazo para		

	eliminar un marcador individual. Cumpliendo la condición de los 10 metros.
--	--

TABLA 3.43: REQUISITO DE SOFTWARE RS-F19

RS-F19			
Nombre	Visualización de ruta entre marcadores		
Prioridad	Alta	Relación	RU-C13
Verificabilidad	Alta		
Descripción	El sistema crea una ruta amarilla que junta de la forma más óptima posible los marcadores que ha puesto el usuario a lo largo del mapa, siempre si son 2 o más.		

TABLA 3.44: REQUISITO DE SOFTWARE RS-F20

RS-F20			
Nombre	Eliminación de ruta y marcadores		
Prioridad	Media	Relación	RU-C12, RU-C13
Verificabilidad	Alta		
Descripción	Se dispone en el menú del brazo izquierdo una opción para eliminar el conjunto completo de marcadores y con ello además la ruta que los une.		

TABLA 3.45: REQUISITO DE SOFTWARE RS-F21

RS-F21			
Nombre	Menú en el brazo		
Prioridad	Media	Relación	RU-C14
Verificabilidad	Media		

Descripción	El sistema ofrece al usuario un menú en el brazo izquierdo que se abrirá cuando se mire la palma completamente abierta. Este menú como ya se ha comentado tiene distintas opciones.
--------------------	---

TABLA 3.46: REQUISITO DE SOFTWARE RS-F22

RS-F22			
Nombre	Menú información de ruta		
Prioridad	Media	Relación	RU-C15, RU-C14
Verificabilidad	Media		
Descripción	En el menú del brazo hay un apartado para mostrar información de la ruta, con esto se mostrará al frente un menú con información adicional de la ruta que se ha creado.		

TABLA 3.47: REQUISITO DE SOFTWARE RS-F23

RS-F23			
Nombre	Teletransporte en la maqueta		
Prioridad	Media	Relación	RU-C16
Verificabilidad	Alta		
Descripción	Estando en el modo de vista del mapa como una maqueta, con el mando derecho pulsando el gatillo posterior aparece un puntero rosa. Con este se puede seleccionar una parte de la maqueta y teletransportarse a esa dirección.		

TABLA 3.48: REQUISITO DE SOFTWARE RS-F24

RS-F24	
Nombre	Viaje a otras ciudades

Prioridad	Media	Relación	RU-C17
Verificabilidad	Media		
Descripción	El sistema ofrece la posibilidad de poder viajar y explorar otras ciudades abriendo la opción de ciudades en el menú del brazo y pulsando una de ellas.		

3.3.2 Requisitos operacionales

TABLA 3.49: REQUISITO DE SOFTWARE RS-O1

RS-O1			
Nombre	Aplicación de Realidad Virtual		
Prioridad	Alta	Relación	RU-R1
Verificabilidad	Alta		
Descripción	La aplicación es compatible con la tecnología de visualización e interacción de Realidad Virtual		

TABLA 3.50: REQUISITO DE SOFTWARE RS-O2

RS-O2			
Nombre	Plataforma de desarrollo		
Prioridad	Alta	Relación	RU-R2
Verificabilidad	Alta		
Descripción	El desarrollo completo de la aplicación será realizado con la ayuda del motor de desarrollo Unity.		

TABLA 3.51: REQUISITO DE SOFTWARE RS-O3

RS-O3			
Nombre	Compatibilidad Oculus		
Prioridad	Alta	Relación	RU-R3
Verificabilidad	Alta		
Descripción	La aplicación debe ser compatible con las gafas Oculus Quest 2.		

TABLA 3.52: REQUISITO DE SOFTWARE RS-O4

RS-O4			
Nombre	Controles dentro de la aplicación		
Prioridad	Alta	Relación	RU-R4
Verificabilidad	Alta		
Descripción	La interacción que se da de cara al usuario puede realizarse con los mando Oculus Touch, seguimiento gestual de manos o sensores Antilatency.		

3.3.3 Requisitos de usabilidad

TABLA 3.53: REQUISITO DE SOFTWARE RS-U1

RS-U1			
Nombre	Aplicación intuitiva		
Prioridad	Media	Relación	
Verificabilidad	Media		
Descripción	La interacción con el entorno virtual de la aplicación debe ser lo más intuitiva y simplificada posible.		

3.4 Casos de uso

En este apartado se especificarán los casos de uso dentro de la aplicación, que vienen dados tanto por los requisitos de usuario como por los de software. Con estos se simulan los distintos escenarios que se pueden dar utilizando la aplicación.

Primero se expondrá un diagrama para mostrar gráficamente los flujos de interacción y cuales son necesarios de cara a otros. Por ejemplo, no se puede eliminar un marcador que no haya sido creado previamente o pasar a un modo de vista estando anteriormente en otro distinto.

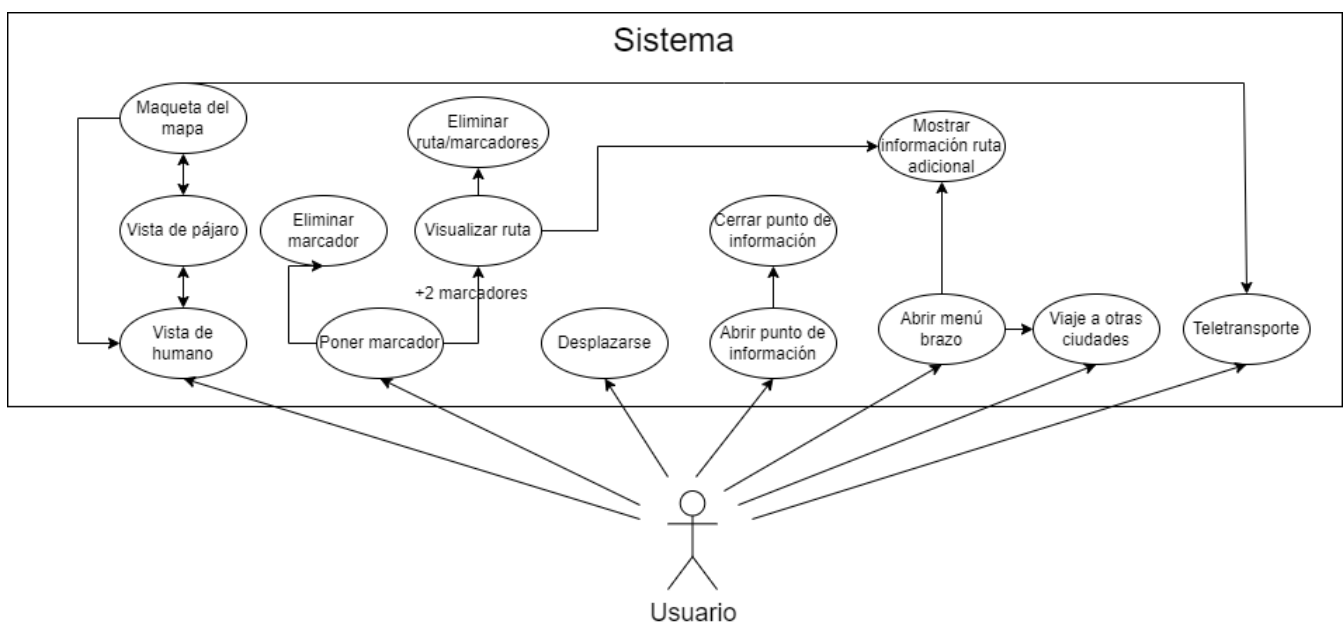


Fig. 3.1 Diagrama de casos de uso

Como se puede ver con el flujo de las flechas hay estados de los casos de uso que necesitan que otros hayan ocurrido antes como es comprensible.

Más en concreto a continuación se nombrarán cada uno de los casos de uso que pueden ocurrir en el sistema, esto constará de una tabla con distintas propiedades:

- **Identificador.** Cada uno posee un identificador distinto. El formato será CU-X, donde CU significa caso de uso y la X equivaldrá al número correspondiente para que se distingan. Irá aumentado en una unidad en cada uno.
- **Nombre.** Nombre que indique una breve descripción del caso.
- **Flujo.** Son los pasos necesarios para poder realizar ese caso de uso.

- **Flujo alternativo (si es necesario).** Hay casos de uso que en el caso de esta aplicación se puede llegar de distintas formas, con acciones diferentes se llega al mismo resultado. Se enumerarán los pasos de ese camino alternativo.
- **Precondiciones.** Es lo que se debe cumplir para que la operación del caso de uso se pueda realizar, como por ejemplo el sentido de las flechas del diagrama.
- **Postcondiciones.** Cómo responde el sistema ante las acciones realizadas.
- **Relación.** Como en el apartado de los requisitos de software, se indica los requisitos de usuario relacionados con el caso de uso.

La plantilla de la tabla tendrá esta forma.

TABLA 3.54: PLANTILLA DE CASOS DE USO

Identificador	
Nombre	
Flujo	Acción 1
	Acción 2
Flujo alternativo	Acción 1
	
Precondiciones	
Postcondiciones	
Relación	

Los casos de uso que se han mostrado en el diagrama se mostrarán a continuación.

TABLA 3.55: CASO DE USO CU-1

CU-1	
Nombre	Vista de humano
Flujo	El usuario se encuentra en el estado por defecto cuando comienza el uso de la aplicación.
Flujo alternativo	Estando en la vista de pájaro y reconocimiento de manos.

	Realizar gesto "Thumb Down".
Flujo alternativo 2	Vista de pájaro con los mandos Oculus Touch.
	Con el mando izquierdo tocar el gatillo lateral a la altura del dedo corazón.
Flujo alternativo 3	Vista de maqueta con el mapa.
	Con el uso de las Oculus Touch y el mando derecho activar el puntero con el gatillo posterior
	Apuntar con el puntero a cualquier lugar del mapa y presionar el botón A.
Precondiciones	Las precondiciones en el caso que no son la vista por defecto, es encontrarse en cualquiera de esas dos vistas.
Postcondiciones	El usuario acabará en la vista de humano, es decir a la altura del suelo.
Relación	RU-C1, RU-C2, RU-C3, RU-C4, RU-C16

TABLA 3.56: CASO DE USO CU-2

CU-2	
Nombre	Vista de pájaro
Flujo	En la vista de humano y con los sensores de las manos.
	Levantar la mano derecha lo más alto posible.
Flujo alternativo	Con los mandos de Oculus Touch.
	Pulsar el gatillo posterior del mando izquierdo.
Flujo alternativo 2	Con el seguimiento de manos y vista humano.
	Realizar el gesto de "Thumbs Up" con la mano izquierda.
Flujo alternativo 3	Estando en la vista de maqueta del mapa.
	Pulsar el gatillo lateral del mando derecho de las Oculus Touch.

Precondiciones	Para poder pasar a la vista de pájaro, se necesita estar previamente en la vista de humano o bien en la de la maqueta y querer volver a la de pájaro.
Postcondiciones	El usuario acaba en una vista en la que ve el mapa desde una localización más alta y poder inspeccionar un mayor rango de este, con menos detalle en cuanto a edificios y calles.
Relación	RU-C1, RU-C2, RU-C3, RU-C4

TABLA 3.57: CASO DE USO CU-3

CU-3	
Nombre	Vista maqueta del mapa
Flujo	El usuario se encuentra en la vista de pájaro y con el uso de las Oculus Touch.
	Con el mando derecho pulsando en el gatillo lateral entrará en modo de vista del mapa como una maqueta.
Precondiciones	El usuario debe estar en el modo de vista de pájaro para poder realizar este caso de uso.
Postcondiciones	Todos los elementos desaparecen y solo se queda el mapa en miniatura como si fuese una maqueta. Pudiendo realizar la acción de teletransporte.
Relación	RU-C1, RU-C3, RU-C4

TABLA 3.58: CASO DE USO CU-4

CU-4	
Nombre	Poner marcador
Flujo	Coger el objeto adicional con el sensor que sirve como objeto auxiliar.

	Dejar o acercar lo máximo el objeto en el suelo
Flujo alternativo	Abrir el menú disponible en el brazo izquierdo con el reconocimiento de manos.
	Pulsar el botón que pone "Put Point"
Flujo alternativo 2	Con el uso de los mandos Oculus Touch
	Pulsar el botón Y disponible en el mando izquierdo.
Precondiciones	La única precondición es estar en la vista de humano o pájaro y tener activo el reconocimiento que se considere
Postcondiciones	Aparecerá un marcador en el lugar donde se ha realizado la acción, una especie de objeto morado con un haz de luz similar para que se pueda ver a distintas distancias.
Relación	RU-C8, RU-C2, RU-C3, RU-C14

TABLA 3.59: CASO DE USO CU-5

CU-5	
Nombre	Eliminar marcador
Flujo	Abrir el menú disponible en el brazo izquierdo en el entorno virtual.
	Realizar el gesto de pulsar con el dedo de la mano derecha.
Flujo alternativo	Realizar el uso de los mandos Oculus Touch
	Pulsar el botón X del mando izquierdo.
Precondiciones	Debe haber al menos un marcador en el entorno y el usuario encontrarse a menos de 10 metros virtuales del mismo, además de estar en la visión de humano.
Postcondiciones	El marcador que previamente había sido creado es eliminado del entorno.
Relación	RU-C2, RU-C10, RU-C11, RU-C14

TABLA 3.60: CASO DE USO CU-6

CU-6	
Nombre	Visualizar ruta
Flujo	El usuario puede ver una ruta amarilla que uno los marcadores que el mismo ha creado.
Precondiciones	Debe haber en el entorno al menos 2 marcadores y el usuario encontrarse en la vista de humano o pájaro.
Postcondiciones	El usuario ve una ruta amarilla en las calles que une de una forma óptima los marcadores que él mismo ha puesto alrededor del mapa.
Relación	RU-C13, RU-C2, RU-C3

TABLA 3.61: CASO DE USO CU-7

CU-7	
Nombre	Eliminar ruta/marcadores del mapa
Flujo	Abrir menú del brazo izquierdo virtual.
	Realizar acción de pulsar con el dedo el botón "Delete All Points"
Precondiciones	Para poder realizar esta acción debe encontrarse en el mapa distintos marcadores y la ruta que los une.
Postcondiciones	Tanto la ruta como los marcadores son eliminados del mapa.
Relación	RU-C13, RU-C14

TABLA 3.62: CASO DE USO CU-8

CU-8	
Nombre	Desplazamiento del usuario
Flujo	Con el seguimiento de los sensores adicionales de las piernas y la alfombra.
	Realizar la acción de andar, pero en el sitio de la alfombra, es decir en estático.
Flujo alternativo	Poner en activo el seguimiento de manos de Oculus.
	Con la mano derecha hacer el gesto de apuntar hacia delante con el dedo índice.
Flujo alternativo 2	Poner en activo el seguimiento de manos de Oculus.
	Con la mano derecha hacer el gesto de apuntar hacia delante, pero esta vez con dos dedos a la vez, el índice y el corazón. El usuario se moverá más deprisa en comparación.
Flujo alternativo 3	Coger los mandos de Oculus Touch
	Con ambos Joysticks moverse en la dirección que se desee. Siendo el del mando izquierdo para moverse hacia delante y hacia atrás, y el derecho para rotar la cámara en la dirección que se quiera.
Precondiciones	La única precondición es tener activa la opción que se quiera realizar para poder hacer la acción.
Postcondiciones	En todos los casos menos en el del movimiento con los mandos, la cámara se moverá en la dirección que se esté mirando con las gafas.
Relación	RU-C5, RU-C1, RU-C2, RU-C2, RU-C4

TABLA 3.63: CASO DE USO CU-9

CU-9	
Nombre	Abrir punto de información

Flujo	Seguimiento de manos activo.
	Desplazarse hasta un marcador de información disponible, un símbolo de una i.
	Apuntar con los punteros que salen de las manos al módulo de información. Aparecerán dos círculos blancos en el módulo.
	Realizar el gesto de pinza con la mano deseada, juntando índice y pulgar.
Precondiciones	Encontrarse en un entorno con puntos de información y cerca de uno de ellos. Únicamente en la vista de humano.
Postcondiciones	Se abre un módulo con foto e información del punto de interés.
Relación	RU-C2, RU-C6

TABLA 3.64: CASO DE USO CU-10

CU-10	
Nombre	Cerrar punto de información
Flujo	Seguimiento de manos activo.
	Apuntar al botón de cerrar disponible en el módulo de información abierto, en la parte inferior derecha.
	Realizar el gesto de pinza con cualquiera de las dos manos.
Precondiciones	Las mismas condiciones que el caso anterior y además que esté abierto ese módulo de información.
Postcondiciones	El módulo de información se cierra.
Relación	RU-C2, RU-C6, RU-C7, RU-C8

TABLA 3.65: CASO DE USO CU-11

CU-11	
Nombre	Abrir menú del brazo
Flujo	Seguimiento de manos activo.
	Con la mano izquierda realizar el gesto de mirarse la palma completamente abierta.
Precondiciones	Tener el seguimiento de manos activo.
Postcondiciones	Aparece un menú con distintos botones en la parte que se corresponde con el antebrazo del usuario. Siendo posible pulsar cualquiera de ellos.
Relación	RU-C14

TABLA 3.66: CASO DE USO CU-12

CU-12	
Nombre	Mostrar información adicional de ruta
Flujo	Abrir el menú del brazo.
	Pulsar el botón de "Route Information"
Flujo alternativo	Con los mandos Oculus Touch.
	Pulsar el botón B del mando derecho.
Precondiciones	Tanto si hay ruta como no, la información saldrá. Por ejemplo, si no hay todos los valores serán 0.
Postcondiciones	Aparece un menú al frente tanto con el modo de visión como con información de la ruta, número de marcadores y distancia de esta.
Relación	RU-C15, RU-C14, RU-C1, RU-C13

TABLA 3.67: CASO DE USO CU-13

CU-13	
Nombre	Viaje a otras ciudades
Flujo	En el menú del brazo izquierdo pulsar el botón de "Cities"
	Con esto se abrirá otro pequeño menú con tres ciudades distintas siendo posible la elección de cualquiera.
Precondiciones	Realizar el gesto correspondiente para poder abrir el menú del brazo.
Postcondiciones	El entorno visual cambia, recargando el terreno y el mapa 3D adecuándose a la ciudad que se ha elegido.
Relación	RU-C17, RU-C14

3.5 Trazabilidad

El siguiente apartado sirve para comprobar la consistencia en los distintos apartados del análisis, con la ayuda de las llamadas matrices de trazabilidad para comprobar qué requisitos tienen relación con otros. Si existe relación entre ellos se marca con una X en la posición correspondiente.

3.5.1 Matriz de trazabilidad Requisitos de usuario-Requisitos de software

Con esta tabla en concreto se puede ver que los requisitos de software provienen de los requisitos de usuario, es decir se han tenido en cuenta. Solo se van a analizar los requisitos de usuario de capacidad y los requisitos de software funcionales, pues los demás son aspectos más de restricciones y cumplimiento de rendimiento que funcionalidad como tal.

TABLA 3.68: MATRIZ DE TRAZABILIDAD RU-RS

RU RS	RU - C1	RU- C2	RU- C3	RU- C4	RU- C5	RU- C6	RU- C7	RU- C8	RU- C9	RU- C10	RU- C11	RU- C12	RU- C13	RU- C14	RU- C15	RU- C16	RU- C17
RS-F1	X																
RS-F2	X	X															
RS-F3		X	X					X									
RS-F4		X	X				X		X								
RS-F5		X	X				X										
RS-F6		X	X					X									
RS-F7			X	X			X										
RS-F8		X	X		X		X										
RS-F9		X	X		X			X									
RS-F10		X	X		X			X									
RS-F11		X	X		X				X								
RS-F12		X				X	X	X									
RS-F13		X					X	X									
RS-F14		X							X	X							
RS-F15		X					X			X							
RS-F16								X		X				X			
RS-F17		X					X			X	X						
RS-F18								X			X			X			
RS-F19										X			X				
RS-F20								X				X	X				
RS-F21								X						X			
RS-F22														X	X		
RS-F23				X												X	
RS-F24														X			X

3.5.2 Matriz de trazabilidad Casos de uso-Requisitos de software

La siguiente tabla presentará la matriz de trazabilidad entre los casos de uso y los requisitos de software, como ambos casos son sacados de los requisitos de usuario se puede comprobar la consistencia de la especificación.

TABLA 3.69: MATRIZ DE TRAZABILIDAD CU-RS

CU RS	CU - 1	CU -2	CU -3	CU -4	CU -5	CU -6	CU -7	CU -8	CU -9	CU -10	CU -11	CU -12	CU -13
RS-F1	X	X	X	X	X	X		X	X	X			
RS-F2	X	X		X	X			X					
RS-F3		X											
RS-F4		X											
RS-F5		X											
RS-F6	X												
RS-F7			X										
RS-F8								X					
RS-F9								X					
RS-F10								X					
RS-F11								X					
RS-F12									X	X			
RS-F13										X			
RS-F14				X		X							
RS-F15				X		X							
RS-F16				X		X							
RS-F17					X								
RS-F18					X								
RS-F19						X	X					X	
RS-F20							X						
RS-F21							X				X	X	X
RS-F22												X	
RS-F23	X		X										
RS-F24													X

4. DISEÑO E IMPLEMENTACIÓN

4.1 Diseño

En esta sección se va a detallar el diseño de la arquitectura del sistema con pequeños diagramas y tablas para exponer de una forma más sencilla el caso.

Primero se va a definir la arquitectura pudiendo diferenciar distintas capas o elementos con su funcionalidad y posteriormente la enumeración de los componentes con una explicación de qué función realizan y su relación con los demás. Esto es una buena práctica también para comprobar la relación de estos componentes con los requisitos de software y la funcionalidad completa de la aplicación.

Por último, también se detallarán algunas especificaciones y características de los elementos gráficos de la aplicación, incluyendo capturas sacadas del juego con la visión del usuario en el entorno virtual.

4.1.1 Arquitectura

Como en la mayoría de los proyectos software se puede seguir un modelo de desarrollo llamado la programación por capas que consiste básicamente en dividir el sistema en distintas capas o partes bastante diferenciables. Lo cual es de gran ayuda a la hora de la implementación y el poder realizar cambios sin perjudicar a otras partes de la aplicación. En el caso de esta aplicación se puede distinguir:

- **Capa de Presentación.** Es lo que el usuario ve, es el puente entre el usuario y el sistema. Es la encargada de presentar la información al usuario y capturar las entradas de este, a su vez es la parte más gráfica de la aplicación.
- **Capa de Negocio.** Es la parte que contiene la lógica de la aplicación, es decir interpreta lo capturado por la capa de Presentación, realiza las acciones correspondientes y devuelve el resultado indicado.

El modelo de desarrollo de programación por capas tiene una tercera, la llamada capa de datos que se encarga de gestionar el almacenamiento y acceder a lo que pueda pedir la capa de negocio. En este caso, como la aplicación no guarda los datos de forma permanente no es necesaria esta capa.

A continuación, se va a mostrar un diagrama con ambas capas de la aplicación ya nombradas y cada una de las partes en las que se subdividen. Estas subdivisiones tienen los distintos componentes del sistema con su funcionalidad propia. En este subapartado

se describirá el diagrama y sus partes y en el siguiente se especificará cada uno de los componentes con sus tablas correspondientes.

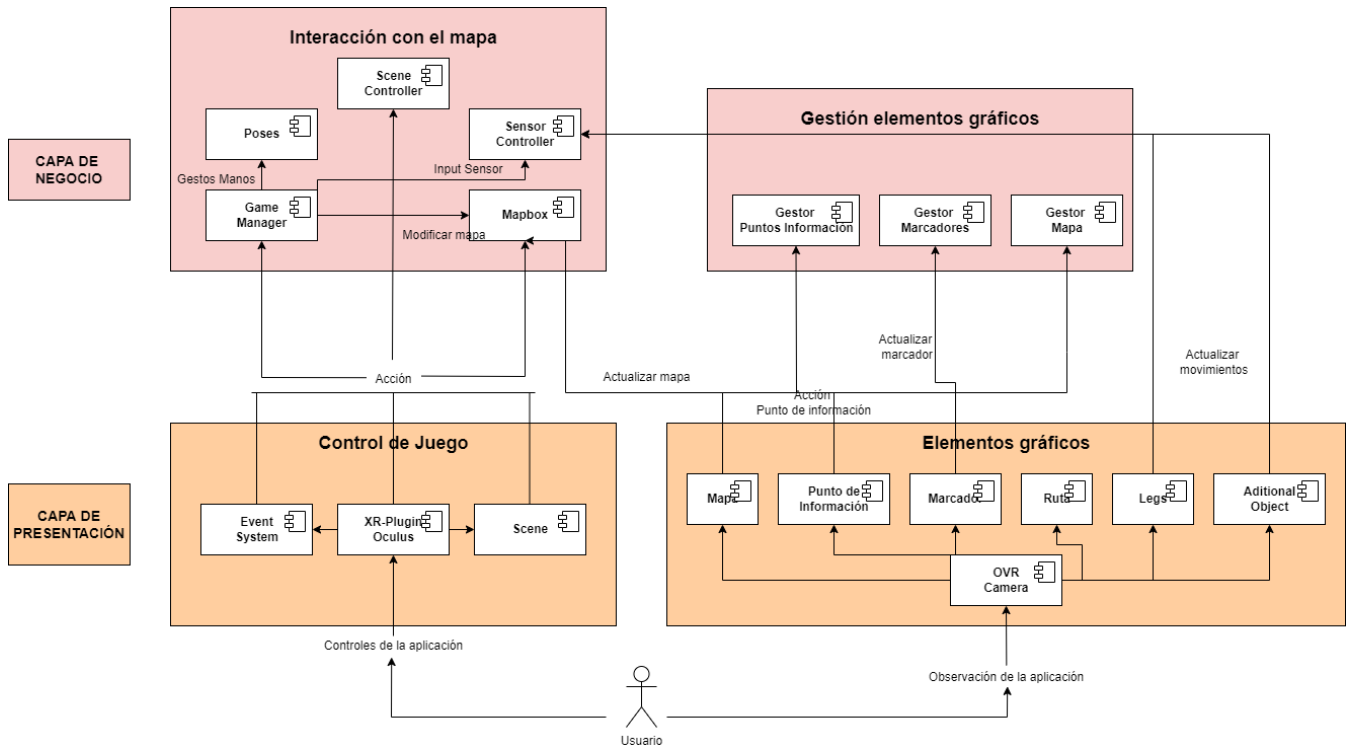


Fig. 4.1 Diagrama de Arquitectura del sistema

Como se puede ver cada una de las capas tiene otros dos cuadros del mismo color, esto son partes que se pueden sacar de esa capa o también llamados subsistemas. Cada subsistema tiene una función dentro de la aplicación, por ejemplo, en el caso de la capa de presentación que es con la que realmente el usuario se comunica, se puede distinguir el subsistema de control de juego y el de elementos gráficos. El primero tiene la función de recoger la información que el usuario proporciona a la hora del control de la aplicación y el segundo se refiere a los componentes u objetos que el usuario puede ver e interactuar en el entorno.

En la capa de negocio encargada de hacer que la aplicación responda ante los inputs del usuario se puede distinguir el subsistema de Interacción con el mapa y la gestión de elementos gráficos. En la interacción con el mapa se encuentran los componentes responsables de identificar la acción que el usuario realiza como ejemplo, el movimiento, poner un marcador o realizar un gesto, así como cambiar lo necesario en el entorno (desplazar cámara, identificación de gestos, etc.). En el subsistema de gestión de los

elementos gráficos se realiza todo lo relacionado con lo que realmente se ve, es decir, si el usuario abre un punto de información para leer la descripción, el componente encargado de su gestión lo tiene que identificar y desplegar. De la misma forma ocurre con los marcadores, la ruta o incluso el propio mapa.

4.1.2 Componentes

En este apartado se van a describir cada uno de los componentes que forman el sistema y que aparecen en el diagrama anterior de la arquitectura. A su vez estos tienen relación con los requisitos software por lo que al finalizar esta sección se mostrará la matriz de trazabilidad correspondiente entre ambos. Los distintos apartados que tendrá cada uno serán los siguientes:

- **Identificador.** Cada componente tendrá un identificador único para diferenciarlo de los demás con la forma CO-X, siendo CO la abreviatura de componente y la X un número entero que irá subiendo por unidades para cada componente.
- **Nombre.** Nombre del componente en el diagrama y que describa un poco su función.
- **Función.** Explica el cometido del componente en el conjunto del sistema, desde su tarea principal a las que pueden ser subordinadas.
- **Subsistema.** Especificación del subsistema al que pertenece dentro del diagrama anteriormente descrito.
- **Dependencias.** Lo que necesita el componente para poder funcionar.
- **Subordinados.** Indica que otros componentes pueden formar parte de ese flujo de trabajo.

Las tablas de los componentes seguirán la plantilla de la siguiente.

TABLA 4.1: PLANTILLA DE COMPONENTES

Identificador	
Nombre	
Función	
Subsistema	
Dependencias	
Subordinados	

A continuación, se realiza la especificación de cada uno de los componentes.

TABLA 4.2: COMPONENTE CO-1

CO-1	
Nombre	XR-Plugin Oculus
Función	Es el componente encargado de recoger los inputs del usuario, ya sea por mandos, manos, etc. Al ser una extensión de Oculus facilita el trabajo con el mismo.
Subsistema	Control de Juego
Dependencias	CO-2
Subordinados	CO-3, CO-14, CO-15, CO-16, CO-17, CO-18

TABLA 4.3: COMPONENTE CO-2

CO-2	
Nombre	Event System
Función	Al igual que el componente anterior se encarga de enviar eventos a objetos en la aplicación, de cambiar entre módulos de input, manejar estados, etc.
Subsistema	Control de Juego
Dependencias	CO-1
Subordinados	CO-3, CO-14, CO-15, CO-16, CO-17, CO-18

TABLA 4.4: COMPONENTE CO-3

CO-3	
Nombre	Scene
Función	Es el componente de la escena que se muestra en el entorno, por ejemplo, hay distintas escenas entre las que se va cambiando, se comienza en una habitación y posteriormente se cambia a la

	escena del mapa. Para cambiar de ciudad también se cambia de escena.
Subsistema	Control de Juego
Dependencias	CO-1, CO-2
Subordinados	CO-14, CO-15, CO-16, CO-17, CO-18

TABLA 4.5: COMPONENTE CO-4

CO-4	
Nombre	OVR Camera
Función	Componente de la cámara de la aplicación. Sin este no se podría disfrutar de ningún elemento gráfico del sistema. Contiene elementos como los controladores del juego, manos, etc.
Subsistema	Elementos gráficos
Dependencias	Ninguna
Subordinados	CO-5, CO-6, CO-7, CO-8, CO-9, CO-10

TABLA 4.6: COMPONENTE CO-5

CO-5	
Nombre	Mapa
Función	Componente gráfico del mapa, contiene todas las calles y edificios que se ven en el entorno virtual, por el que el usuario se puede mover e interactuar.
Subsistema	Elementos gráficos
Dependencias	CO-4, CO-15
Subordinados	CO-6, CO-7, CO-8

TABLA 4.7: COMPONENTE CO-6

CO-6	
Nombre	Punto de información
Función	Elemento gráfico que contiene el punto de información de un punto turístico de la ciudad. Si se pulsa tiene un desplegable con información y foto del lugar.
Subsistema	Elementos gráficos
Dependencias	CO-4, CO-5
Subordinados	CO-11

TABLA 4.8: COMPONENTE CO-7

CO-7	
Nombre	Marcador
Función	Componente gráfico del marcador que el usuario puede poner a lo largo del mapa y que posteriormente formará la ruta con los distintos dentro del entorno.
Subsistema	Elementos gráficos
Dependencias	CO-4, CO-5, CO-15
Subordinados	CO-12, CO-8

TABLA 4.9: COMPONENTE CO-8

CO-8	
Nombre	Ruta
Función	Elemento gráfico que compone la ruta entre los distintos marcadores del usuario en el mapa. Totalmente editable a la que se le puede añadir o quitar marcadores.

Subsistema	Elementos gráficos
Dependencias	CO-7, CO-5, CO-4, CO-15
Subordinados	Ninguno

TABLA 4.10: COMPONENTE CO-9

CO-9	
Nombre	Legs
Función	Componente gráfico que simula las piernas del usuario a través de los sensores Antilatency y que además recogen el movimiento. Pudiendo realizar la acción de andar dentro del sistema.
Subsistema	Elementos gráficos
Dependencias	CO-4
Subordinados	CO-18

TABLA 4.11: COMPONENTE CO-10

CO-10	
Nombre	Aditonal Object
Función	Componente del objeto adicional encargado de recoger si el usuario lo suelta o no en el suelo y con ello poner un marcador en mapa.
Subsistema	Elementos gráficos
Dependencias	CO-4
Subordinados	CO-18

TABLA 4.12: COMPONENTE CO-11

CO-11	
Nombre	Gestor de puntos de información
Función	Se trata del gestor de los puntos de información, es decir se encarga de recoger qué punto quiere abrir o cerrar el usuario y leer su información.
Subsistema	Gestión de elementos gráficos
Dependencias	CO-6
Subordinados	CO-6

TABLA 4.13: COMPONENTE CO-12

CO-12	
Nombre	Gestor de marcadores
Función	Se encarga de controlar cuando se pone o quita un marcador a través de lo que por ejemplo envíe el GameManager.
Subsistema	Gestión de elementos gráficos
Dependencias	CO-7, CO-14
Subordinados	CO-7

TABLA 4.14: COMPONENTE CO-13

CO-13	
Nombre	Gestor de mapa
Función	Componente que controla las características del mapa, al cambiar entre distintas vistas y el movimiento dentro del entorno.
Subsistema	Gestión de elementos gráficos

Dependencias	CO-5, CO-14, CO-15
Subordinados	CO-5

TABLA 4.15: COMPONENTE CO-14

CO-14	
Nombre	GameManager
Función	Componente que a través de la acción que se le envía desde el subsistema de Control de Juego, comprueba las condiciones correspondientes y toma la decisión, ya sea movimiento, cambiar vista, poner marcador, etc.
Subsistema	Interacción con el mapa
Dependencias	CO-1, CO-2
Subordinados	Todos los elementos gráficos

TABLA 4.16: COMPONENTE CO-15

CO-15	
Nombre	Mapbox
Función	Componente y extensión de Oculus, que controla todo lo relacionado con el mapa, se encarga por ejemplo de cargar los elementos a través de una localización, dispone a su vez de componente incorporados para crear la ruta entre marcadores.
Subsistema	Interacción con el mapa
Dependencias	CO-14
Subordinados	CO-13, CO-5

TABLA 4.17: COMPONENTE CO-16

CO-16	
Nombre	Poses
Función	Es el componente que contiene los distintos gestos de las manos que el usuario puede realizar, por lo tanto, cuando el GameManager detecte un gesto dentro de este componente realizará la acción debida.
Subsistema	Interacción con el mapa
Dependencias	CO-14
Subordinados	Ninguno

TABLA 4.18: COMPONENTE CO-17

CO-17	
Nombre	Scene Controller
Función	Su función es controlar en qué escena se encuentra el usuario y ser capaz de cambiar entre ellas. Dependiendo de lo quiera el usuario mostrará una u otra escena.
Subsistema	Interacción con el mapa
Dependencias	CO-14, CO-3
Subordinados	CO-3

TABLA 4.19: COMPONENTE CO-18

CO-18	
Nombre	Sensor Controller
Función	Como su nombre indica controla aquellos sensores adicionales que se utilizan en el entorno, los Antilatency. Con los que recoger

	posición y demás y con ello poder hacer que el usuario pueda moverse o poner un marcador.
Subsistema	Interacción con el mapa
Dependencias	CO-14, CO-9, CO-10
Subordinados	CO-9, CO-10

4.1.2.1 Matriz de trazabilidad Requisitos de software-Componentes

La siguiente tabla muestra la matriz de trazabilidad entre los requisitos de software previamente explicados en su apartado y los componentes del sistema. De esta forma se puede comprobar que la arquitectura del sistema se ha realizado teniendo en cuenta los requisitos y viceversa.

TABLA 4.20: MATRIZ DE TRAZABILIDAD RS-CO

CO RS	CO-1	CO-2	CO-3	CO-4	CO-5	CO-6	CO-7	CO-8	CO-9	CO-10	CO-11	CO-12	CO-13	CO-14	CO-15	CO-16	CO-17	CO-18
RS-F1	X		X	X	X										X			
RS-F2			X	X											X			
RS-F3	X	X		X									X	X		X		
RS-F4	X	X											X	X				
RS-F5	X	X											X	X				
RS-F6	X	X			X								X	X		X		
RS-F7	X	X		X	X								X	X				
RS-F8	X	X	X	X										X				
RS-F9	X	X		X										X		X		
RS-F10	X	X		X										X		X		
RS-F11		X		X					X					X				X
RS-F12	X	X				X					X			X				
RS-F13	X	X				X					X							
RS-F14		X			X		X			X		X		X	X			X
RS-F15	X	X			X		X					X		X	X			
RS-F16	X	X			X		X					X		X	X			
RS-F17	X				X		X					X		X	X			
RS-F18		X			X		X					X		X	X			
RS-F19				X	X			X					X	X	X			
RS-F20	X	X		X	X			X					X	X	X			
RS-F21		X		X														
RS-F22				X				X						X				
RS-F23	X	X	X	X										X				
RS-F24		X	X		X								X				X	

4.1.3 Diseño gráfico

En el siguiente apartado se van a explicar las decisiones de diseño que se han tomado a la hora del desarrollo de la aplicación, más en concreto en el aspecto gráfico. Es decir, mostrar y aclarar el porqué de los objetos gráficos del sistema, ya que siempre se tiene en cuenta la experiencia de usuario en estos casos y a su vez al tratarse de una aplicación de Realidad Virtual la presencia y la inmersión.

A continuación, se van a explicar las distintas partes gráficas del sistema, con su función y decisiones de diseño.

4.1.3.1 Elementos gráficos

- **Mapa.** Es la parte principal del sistema en torno al cual gira toda la lógica de la aplicación. Como ya se ha mencionado en puntos anteriores la forma en la que el mapa se genera es a través de Mapbox que dando una latitud y una longitud crea un modelo 3D de las calles y edificios en la realidad. A pesar de esto Mapbox ofrece la posibilidad de poder configurar dicho mapa, cambiando texturas, alturas, espacios, colores, etc.

La decisión principal ha sido la de intentar hacer el entorno lo más inmersivo posible sin perder rendimiento dentro de la aplicación, por ejemplo, si se hubieran utilizado texturas muy realistas el rendimiento hubiera sido muy inferior pues tanto las gafas como el equipo de trabajo no pueden soportar gráficos muy exigentes. No hay que perder de vista que el objetivo de la aplicación aparte de poder disfrutar de un entorno con mapas inmersivo es la interacción con el mismo y todas sus posibilidades.

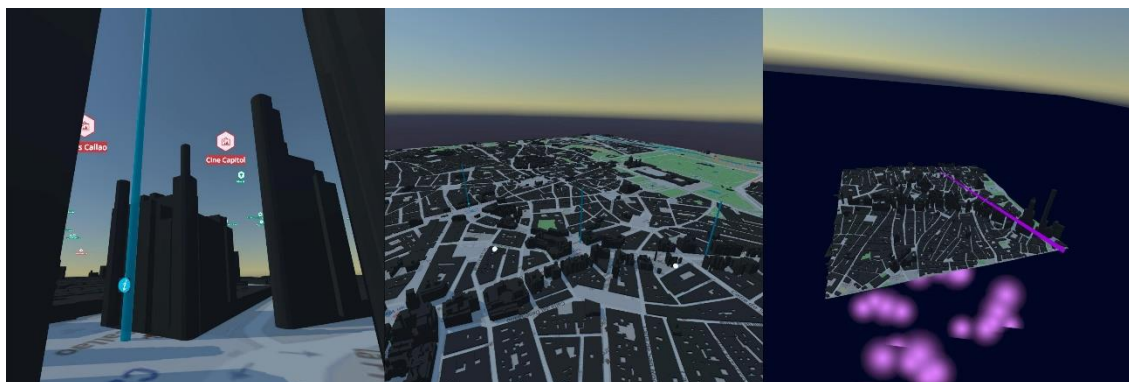


Fig. 4.2 Vistas disponibles del mapa

Las imágenes corresponden con las distintas vistas que el usuario puede hacer del mapa, con las que mirar distintas cosas. Esto se consigue configurando y cambiando los parámetros del mapa dados por Mapbox como extensión del motor de desarrollo Oculus.

- **Punto de información.** Los puntos de información están representados con un símbolo de i y un haz de luz azul que se puede ver a lo largo del mapa. Estos al pulsarlos con el seguimiento de manos despliegan información del sitio donde se encuentran, nombre, foto y breve descripción. Es un módulo que se puede abrir y cerrar dependiendo de lo que el usuario considere.

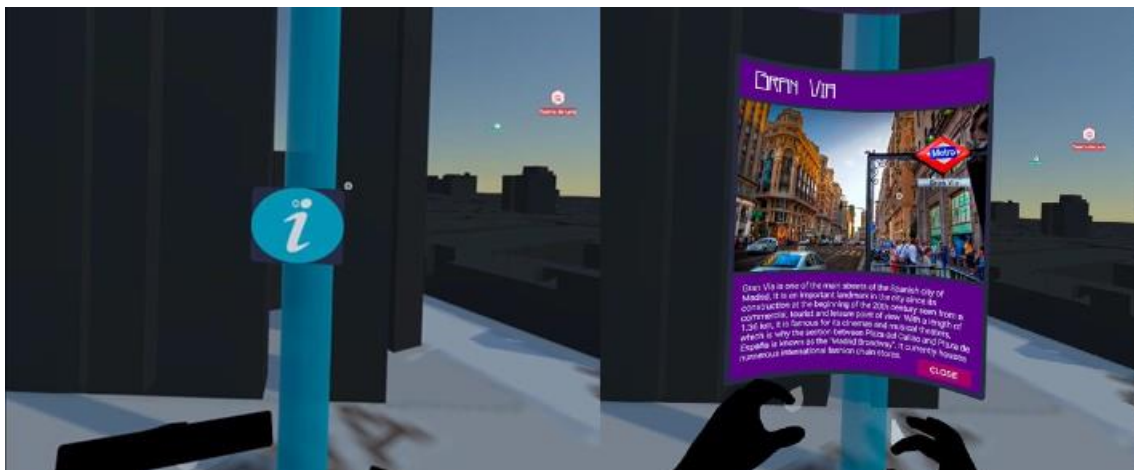


Fig. 4.3 Gráficos puntos de información

- **Marcador.** Los marcadores son los puntos que el usuario pone a lo largo del mapa cuando quiere marcar o destacar un lugar, con estos posteriormente se crea la ruta entre ellos. Su diseño es similar a los puntos de información, pero con la diferencia de que son morados, tanto la especie de cápsula que marca el punto y el haz de luz que sube a lo largo del entorno para que sea visible desde la mayoría de los puntos.

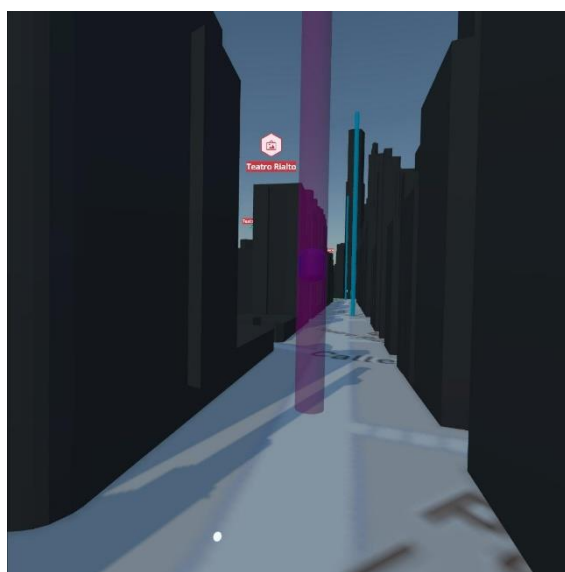


Fig. 4.4 Grafismo del marcador puesto por el usuario

- **Ruta.** La ruta es el elemento que junta de la manera más óptima posible los distintos marcadores que el usuario ha puesto a lo largo del mapa, siempre siguiendo las calles, es decir no traspasa edificios y demás. Esto se consigue gracias a Mapbox que posee una clase a la que se puede llamar para añadir puntos y generar el camino más corto junto a otras métricas e información. En el aspecto gráfico para no sobrecargar de objetos y detalles el entorno, la ruta se muestra como una línea 3D amarilla que va a lo largo de la calle como si de una aplicación de GPS se tratara. Pudiendo verla ya sea desde la vista de humano o la de pájaro.

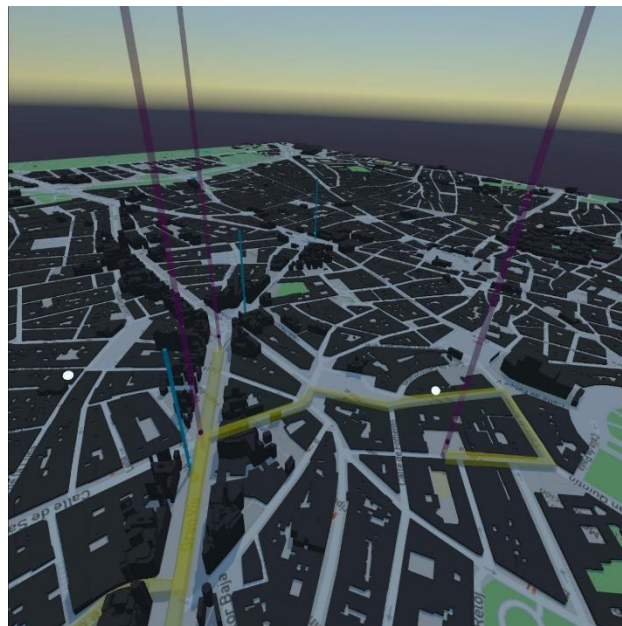


Fig. 4.5 Imagen de ruta desde la vista de pájaro

- **Manos.** Dentro de la aplicación se puede realizar una completa interacción con el uso de seguimiento de manos ofrecido por Oculus y las distintas comprobaciones y funciones que se han implementado en el sistema para que el usuario pueda moverse y realizar acciones. Es por eso que se han añadido gráficamente el modelo de las manos para sentir una mayor presencia en la aplicación, pues dentro del entorno las manos realizan lo que el usuario esté haciendo en la realidad.

Además, con los gestos que se han implementado queda mucho más claro esa importancia del modelo gráfico de las manos.



Fig. 4.6 Imagen de los modelos de manos

- **Modelos de los sensores.** Los sensores Antilatency que sirven como modo de input adicional para acciones como andar y que el movimiento dentro del entorno actúe y por ejemplo poner un marcador dejando en el suelo ese objeto adicional que se ha ido nombrando a lo largo del documento, necesitan que el usuario pueda ver dentro de la aplicación como responden a sus acciones. Es por ello que necesitan de unos modelos 3D en el entorno.

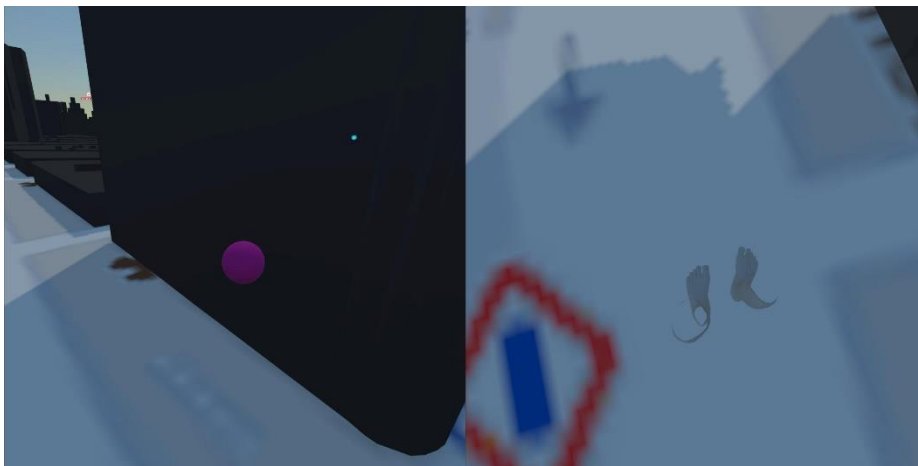


Fig. 4.7 Imagen de los modelos para los sensores Antilatency

Para los sensores de las piernas se han añadido dos modelos de piernas realizado a través de Blender y que darán la sensación de que se está andando en el entorno virtual y para el objeto adicional se ha optado por un modelo que pueda entenderse como un marcador.

- **Menús.** Dentro de la aplicación hay dos menús disponibles, en primer lugar, el menú incorporado en el antebrazo del usuario dentro del entorno que aparece cuando este se mira la palma de la mano izquierda completamente abierta y luego se cierra. Además, dispone de las principales funcionalidades del sistema como poner o quitar marcadores y una muy especial la de poder viajar a otras ciudades. El segundo menú es el disponible cuando está visible la ruta en el mapa y muestra información adicional de esta.

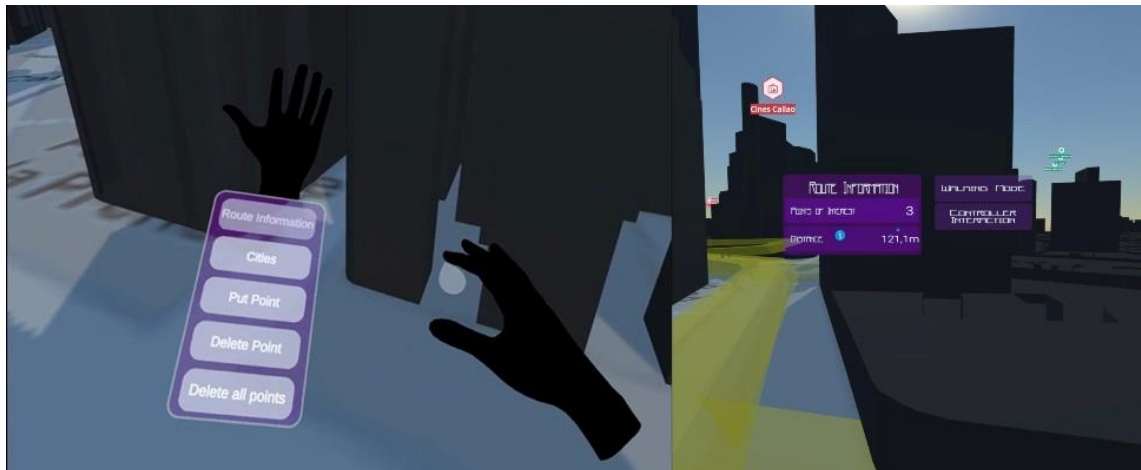


Fig. 4.8 Imagen de los menús disponibles en la aplicación

Las decisiones gráficas en cuanto a los menús han sido la de implementarlos de la manera simple posible, que los colores sean consistentes con los de la aplicación y que no entorpezcan la experiencia del usuario.

4.1.4 Controles de juego

Como se ha ido mencionando a lo largo de la redacción del documento existen distintas formas de interacción dentro de la aplicación, esto quiere decir que el usuario puede realizar las distintas acciones de formas diferentes. Ya sea con el uso de los mandos, reconocimiento de manos o algunas acciones con los sensores.

Primeramente, se mostrará una tabla con las acciones y sus controles correspondientes en la parte del uso de los mandos Oculus Touch, con imagen para que se vea claro a qué corresponde cada botón.

TABLA 4.21: CONTROLES MANDOS OCULUS TOUCH

Acción	Control
--------	---------

Moverse hacia delante	Mover joystick izquierdo hacia delante
Moverse hacia atrás	Mover joystick izquierdo hacia atrás
Girar la cámara	Mover joystick derecho a los lados
Vista de pájaro	Pulsar gatillo posterior mando izquierdo
Vista de humano	Pulsar gatillo lateral mando izquierdo
Maqueta del mapa	Pulsar gatillo lateral mando derecho
Mostrar/Ocultar puntero mando derecho	Pulsar gatillo posterior mando derecho
Elegir punto de la maqueta	Donde esté apuntando el puntero pulsar botón A
Poner marcador	Pulsar botón Y
Eliminar marcador	Pulsar botón X
Mostrar/Ocultar menú información de ruta	Pulsar botón B

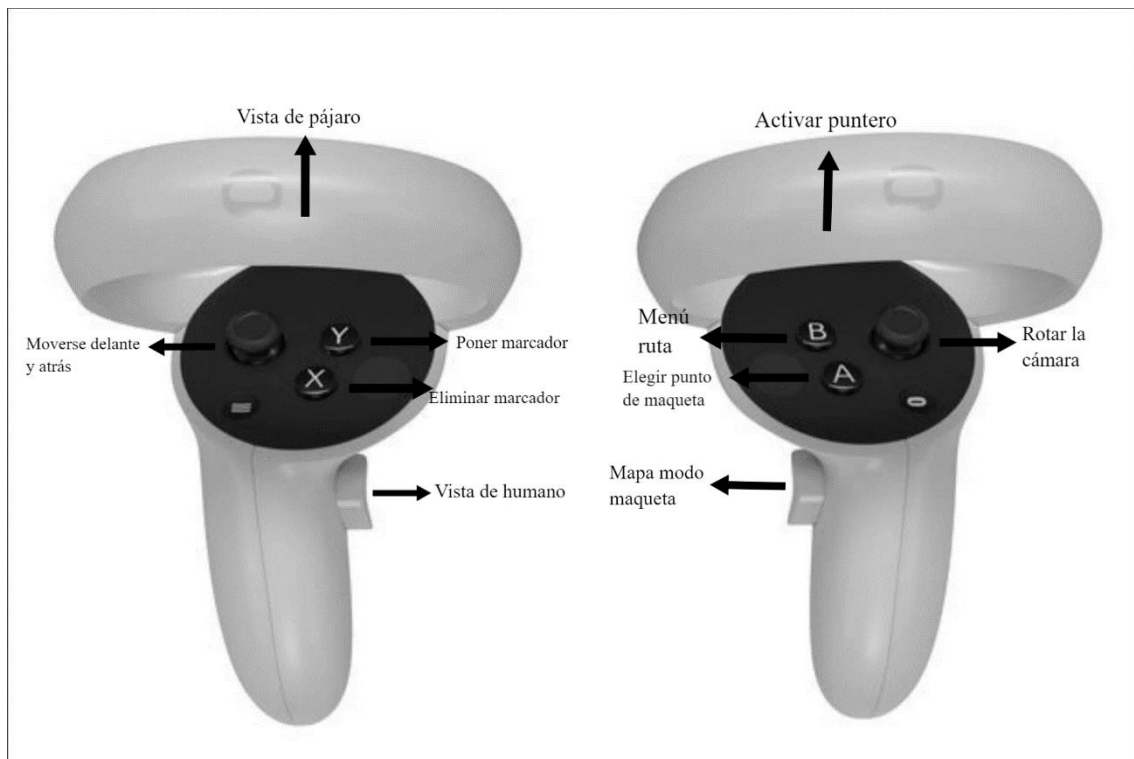


Fig. 4.9 Controles con los mandos de Oculus Quest 2

Tras esta parte de interacción con los mandos también se encuentran las diferentes acciones con los gestos manuales. En la tabla siguiente vendrá explicado la acción con su control o gesto correspondiente.

TABLA 4.22: CONTROLES/GESTOS CON RECONOCIMIENTO DE MANOS

Acción	Control/Gesto
Moverse hacia delante o atrás	Apuntar con el dedo índice de la mano derecha y los demás cerrados. Se mueve donde el usuario esté mirando.
Moverse hacia delante o atrás (más rápido)	Igual que el anterior, pero con dos dedos, el índice y el corazón.
Vista de pájaro	Gesto de "Thumbs Up" con la mano izquierda.
Vista de humano	Gesto de "Thumb Down" con la mano izquierda.
Abrir menú brazo	Mirar la palma completamente abierta de la mano izquierda.
Abrir/Cerrar punto de información	Apuntar al módulo con los punteros que salen de las manos y realizar gesto de pinza con el índice y pulgar para pulsar.

Igual que con la interacción anterior se va a mostrar una imagen con los principales gestos manuales y su función, para ejemplificar de mejor forma esta interacción manual.



Fig. 4.10 Controles con los gestos manuales

Para seguir y como última forma de interacción es a través de los sensores Antilatency, no disponen de una funcionalidad completa, pero aportan una mayor sensación de inmersión para cualquiera de los dos casos anteriores.

TABLA 4.23: CONTROLES/GESTOS SENSORES ANTILATENCY

Acción	Control/Gesto
Moverse hacia delante o atrás	Realizar gesto de andar en el sitio, levantando y separando las piernas de forma evidente. Teniendo un sensor en cada pierna.
Poner marcador	Dejar el objeto adicional con un sensor incorporado en el suelo o la alfombra con el sensor.

Son un complemento para hacer la experiencia de usuario lo más inmersiva posible y que este note que sus acciones se tienen en cuenta en el sistema.

4.2 Implementación

En el apartado de la implementación se trata de explicar cuáles han sido las decisiones que se han tomado para conseguir durante el desarrollo de la aplicación aquellos requisitos que previamente han sido descritos. Explicando a grandes rasgos clases o funciones que consiguen el fin de esa funcionalidad.

Para ello se van a dividir en distintos subapartados las funcionalidades y los conceptos principales de la aplicación, teniendo en cuenta sus características y relación.

4.2.1 Escenas

Las escenas en cualquier proyecto de Unity son muy importantes, pues son los elementos principales entorno a los que giran todos los demás aspectos. Se pueden entender como los distintos niveles dentro de un juego, por ejemplo, en el tan conocido Candy Crush cada uno de los niveles que se juegan y se pueden superar sería una escena distinta en la que realizar los cambios que se deseen que no afectarán a los demás niveles.

En esta aplicación en concreto al no haber niveles, se ha implementado cada escena como cada uno de los entornos disponibles, es decir el mapa tridimensional de Madrid que viene por defecto es una escena y si se cambia a otra ciudad como París se estaría cargando su escena correspondiente.

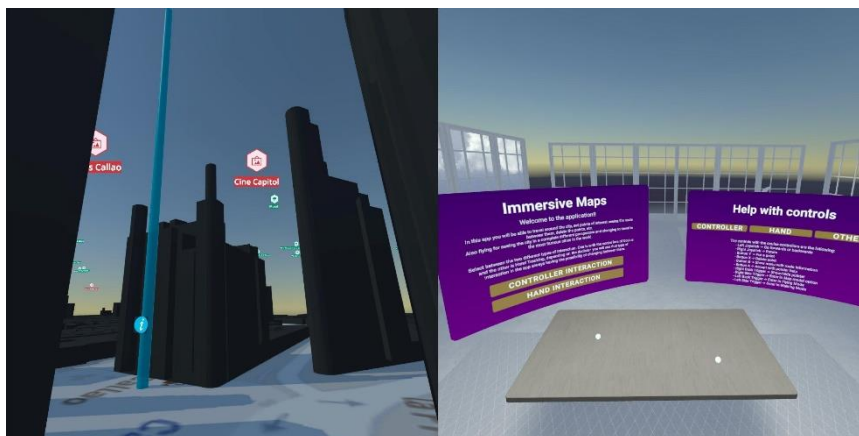


Fig. 4.11 Escenas de la aplicación

Como se puede ver en la imagen, la parte de la izquierda corresponde con la escena o entorno que se ha ido describiendo a lo largo del documento donde se realiza la interacción completa con el mapa. Sin embargo, la parte derecha pertenece a la escena de inicio cuando el usuario abre la aplicación. Se trata de un apartado que sirve como introducción a la aplicación, con una breve explicación de lo que se puede realizar y con paneles con los que se puede interactuar con los gestos manuales para saber por ejemplo los distintos controles dependiendo de la interacción (panel derecho) y elegir con qué interacción realizar posteriormente las acciones con los puntos de información.

4.2.2 Scene Controller (Controlador de escenas)

Como su nombre indica se encarga de controlar lo que ocurre en la escena, desde cambios entre ellas a controlar todo lo relacionado con los puntos de información. Por ejemplo, al comenzar la aplicación y al pulsar una de las dos interacciones en esta clase se guarda ese valor en un archivo de configuración llamado "GameManagerPreset" con el que poder almacenar valores necesarios en otras escenas y que estos no se pierdan.

Aparte se encarga de recoger los inputs por parte del usuario relacionados con los puntos de información, independientemente de la forma de interacción seleccionada, controla si el usuario se encuentra cerca o enfrente del punto, si ha pulsado el botón i e igualmente si lo ha cerrado.

Se trata de una clase o elemento que no dispone de mucha funcionalidad pero que es importante para el desarrollo de la experiencia del usuario.

4.2.3 GameManager

Esta clase controla la mayor parte de la funcionalidad del sistema, se encarga a grandes rasgos de filtrar y reconocer los inputs introducidos por parte del usuario y realizar o mandar realizar a otra clase la acción correspondiente.

Todo este reconocimiento de acciones se encuentra en la función Update que toda clase de C# (lenguaje de programación) en Unity tiene si se quiere realizar cualquier comando en tiempo de ejecución, la forma en la que esta función trabaja es comprobando en cada frame si se cumple alguna de las condiciones desarrolladas. Al cumplir alguna condición se llama a otra función fuera del Update que será la que realice la acción en el entorno virtual, como puede ser poner un marcador o cambiar de vista.

De igual forma todas las comprobaciones necesarias para que se cumpla el correcto funcionamiento de la aplicación también se llevan a cabo en este apartado, como pueden ser las comprobaciones de distancia a los puntos, en qué vista se encuentra el usuario para cambiar elementos del mapa y que la experiencia sea más fluida, etc.

4.2.4 Movimiento de cámara

La cámara es de los aspectos más importantes en una aplicación de Realidad Virtual y más si es de exploración e interacción con el entorno como es el caso, la posible inmersión del usuario dentro de la aplicación se ve influenciada en mayor parte por la cámara y los aspectos que la rodean.

En primer lugar, cabe destacar que se tiene en cuenta la posición de la cámara no solo para el movimiento del usuario sino para otras acciones dentro del sistema que también son muy importantes como a la hora de poner o eliminar un marcador o interactuar con los puntos de información. Estas comprobaciones como se ha mencionado en el apartado anterior se llevan a cabo en el "GameManager".

El movimiento dentro del entorno tridimensional puede venir dado por las tres distintas interacciones posibles en el sistema (uso de mandos Oculus Touch, gestos con las manos y sensores Antilatency), para reconocer cual se está utilizando está el GameManager que dependiendo del que sea llama a una fusión u otra en la clase del movimiento de cámara. Con el uso de los mandos está implementado el poder ir hacia delante y hacia atrás y poder rotar, sin embargo, en los otros dos casos solo se puede ir hacia delante en la dirección que el usuario esté mirando con las gafas lo que hace que se tenga en cuenta aspectos del usuario dando una mayor sensación de integración con el entorno.

4.2.5 Otros

Dentro de la implementación del sistema se encuentran otros dos aspectos importantes para el buen desarrollo de la experiencia de usuario y como funcionalidad principal de la aplicación.

4.2.5.1 Marcadores y ruta

El cómo se ha implementado la funcionalidad de los marcadores y la ruta correspondiente entre ellos viene dado por la ayuda del SDK de Mapbox.

Este contiene una clase llamada "DirectionsFactory" que a través de una lista de objetos con su posición correspondiente crea la ruta más óptima entre ellos, teniendo en cuenta las calles disponibles en el mapa. De esta forma al crear un marcador se añade un objeto a la lista de marcadores, comprobando que no haya dos en el mismo sitio o con una distancia inferior a 10 metros y que la lista contenga dos o más marcadores, tras esto se llama al método "Start()" de la clase de Mapbox creando la ruta correspondiente.

4.2.5.2 Reconocimiento de gestos

Por otro lado, se encuentran los gestos manuales que han de ser reconocidos por el sistema, para ello hay que crear modelos para que después en tiempo de ejecución se pueda comparar las posiciones y formas de las manos del usuario.

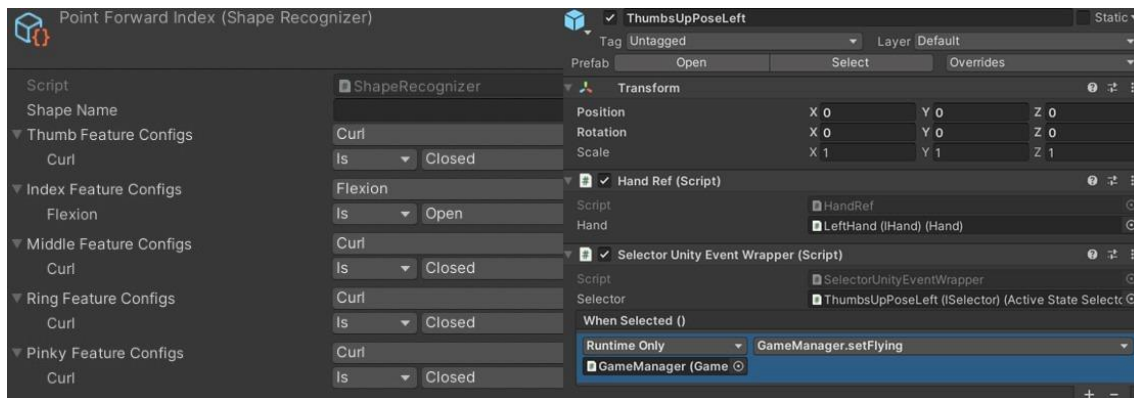


Fig. 4.12 Configuración de los gestos manuales

Como se ve en la imagen primero se crea la forma de la mano teniendo en cuenta cada uno de los dedos, pudiendo estar cerrados, abiertos o doblados. Por ejemplo, en el caso de apuntar hacia delante son todos los dedos cerrados menos el índice que está abierto. Una vez creada la forma hay que crear el modelo, especificando qué mano lo puede realizar, la posición de la muñeca y de la palma de la mano. Todo esto gracias a los componentes proporcionados por Unity y su SDK es un proceso más visual y simple, uno de estos componentes se lanza cuando detecta que todos los aspectos se cumplen y se puede llamar a la función deseada, como en el caso de la imagen de la derecha se ha detectado el gesto "ThumbsUp" que equivale a entrar en la vista de pájaro por eso se llama a la función "setFlying" del "GameManager" que realiza los cambios para entrar en ese modo.

4.2.5.3 Sensores Antilatency

Por último, mencionar la implementación y las decisiones que se han tomado a la hora de conseguir la interacción con los sensores Antilatency.



Fig. 4.13 Entorno de sensores Antilatency

Por una parte, se encuentra el objeto llamado "Environment" que contiene el área de seguimiento, es decir la alfombra con los sensores. Obtiene información de los movimientos que se realizan en esa localización. Otro objeto o elemento es "network", o red que almacena la información de todos los dispositivos Antilatency que se encuentran conectados, como en este caso el sensor del objeto, los sensores de las piernas, etc.

De igual forma todos los objetos o modelos 3D que se van a mover por el espacio y van a servir para las comprobaciones necesarias de distancia al suelo y entre ellos necesitan del script proporcionado por Unity para que también adquieran información de los dos elementos anteriores. Comprobación del que objeto adicional se encuentre lo suficientemente cerca del suelo y que los sensores de las piernas detecten el "Walk in place".

5. ENTORNO SOCIOECONÓMICO

5.1 Impacto socioeconómico

Antes de entrar en el impacto real que puede suponer la aplicación tanto en el entorno económico como en social, cabe destacar y analizar en que ámbitos se puede asociar el uso de este producto. Como se ha explicado un poco por encima y más en concreto en su debido apartado, la aplicación consiste en la posibilidad de interactuar con datos de mapas virtuales en un entorno completamente inmersivo en el que poder moverse con total libertad e inspeccionar un mundo que simula la realidad, con capacidades de establecer puntos, rutas, etc.

Es claro ver su utilidad en el sector turístico, ya sea una persona individual que quiera planificar sus propias vacaciones viendo antes de ir al destino que puntos de interés hay, la distancia entre los distintos puntos o las rutas con su longitud. Pero también puede ser de utilidad para empresas de viajes, con la aplicación pueden ofrecer una forma innovadora e intuitiva de ver lo que se van a encontrar los turistas a la llegada a la ciudad, puede servir como reclamo a clientes que elijan la opción de trabajar o contratar a esa empresa que le está dando opciones que ninguna otra tiene. Todo esto teniendo en cuenta que las distintas posibilidades de interacción que hacen diferenciarse de unas posibles competidoras, analizadas en el apartado del Estado del arte, hacen que sea un producto diferencial y más llamativo.

Más en concreto a nivel económico no supone un gran impacto, pues no está sustituyendo un producto similar ni directamente competidor por ello no se puede mejorar en precio. La inversión necesaria para la adquisición del producto viene dada por el precio del equipo necesario para poder simular el entorno (gafas, sensores, ordenador, etc.) además del de la aplicación en sí, proporcional a las horas invertidas y demás. Ese impacto puede venir dado como se ha mencionado anteriormente en compañías de viaje o similares que sí hagan el uso del producto y sean capaces de adquirir más clientes en comparación con otras que no lo hagan, en ese sí les supone un beneficio económico relevante.

En cuanto al impacto social es más fácil ver que supone un cambio en la forma de interactuar y viajar por mapas que son completamente inmersivos, que viene de la mano con la aceptación y auge de esas tecnologías de la Realidad Virtual que, como se ha comentado en la introducción, es un ámbito que está creciendo considerablemente y tiene un futuro exitoso.

Además, este proyecto puede servir para explorar en esos avances tecnológicos con las distintas formas de interacción que puede tener una persona con un mapa y sus datos y que no sea solo ver imágenes en una pantalla de ordenador dando click en la misma. Se puede visualizar los mapas en un mundo virtual que parezca real, en el que se pueda interactuar de distintas formas ya sea con los mandos correspondientes de las gafas utilizadas como si se tratara de un videojuego más convencional, gestos manuales que permiten moverse y acceder a datos o incluso moverse en el mundo real simulando pasos

y demás y que en el mundo virtual aparezcan unos pies como si se estuviera andando en el videojuego y moviéndose en ese propio espacio virtual. Todo esto teniendo en cuenta donde se encuentran en la actualidad las posibilidades de la Realidad Virtual que su evolución es muy marcada y puede ser en paralelo con esas distintas formas de interacción.

5.2 Presupuesto

El presupuesto de un proyecto de software como es el caso viene dado tanto por el personal que ha dedicado su tiempo a la realización de la aplicación, así como los recursos tecnológicos utilizados para el mismo. A continuación, se va a especificar cada uno de los casos.

En cuanto al personal cabe destacar que se diferencian distintos roles que tienen unas tareas en específico con un sueldo distinto. Esos roles principales son:

- **Jefe del proyecto.** Se encarga de controlar y gestionar los recursos y el buen funcionamiento del proyecto, así como organizar los distintos equipos y personas que trabajan en el proyecto.
- **Analista de software.** Es el encargado de mantener las conversaciones con el cliente y trabajar en paralelo con él, especifica las características técnicas que debe poseer el producto.
- **Arquitecto de software.** Se encarga de realizar el diseño de la arquitectura del sistema y que esto se ajuste a los requisitos especificados tanto por él como el de analista software.
- **Programador.** El encargado de traducir a código cada una de las características que debe cumplir el sistema especificadas por las dos personas anteriores.
- **Tester o responsable de pruebas.** Como su propio nombre indica su trabajo es realizar distintas pruebas funcionales de la aplicación y comprobar que estas funcionan correctamente y se rigen con las especificaciones debidas.

La siguiente tabla muestra una aproximación de las horas dedicadas por el personal involucrado en la aplicación con su coste.

TABLA 6.1. COSTE DEL PERSONAL

Rol	Tiempo dedicado (horas)	Salario/hora (€)	Coste total (€)
Jefe del proyecto	45	19,25	866,25
Analista software	42	16,60	697,20
Arquitecto software	30	17,90	537

Programador	125	14,25	1.781,25
Tester	35	12,40	434
Total	277		4.315,7

En el caso de los costes de recursos o dispositivos tecnológicos utilizados a lo largo del desarrollo tendremos en cuenta los siguientes:

TABLA 6.2: COSTE DE RECURSOS

Recurso	Precio (€)
Portátil Xiaomi Mi Notebook Pro (2018)	845
Meta Oculus Quest 2	449
Antilatency Tracker (x6)	5.600
Tracking Area	450
HMD Radio Socket	120
Tag (x3)	360
Bracer (x2)	290
Total	8.114

Por lo tanto, una estimación total del presupuesto del proyecto vendrá dado por la suma de los costes de personal con los costes de los recursos.

El total aproximado del presupuesto es $4.315,7 + 8.114$, lo que da alrededor de 12.429 €.

6. MARCO REGULADOR

Este apartado del marco regulador sirve como un análisis a la legislación que se puede encontrar alrededor del proyecto, así como algunos estándares aplicables en el trabajo.

En cuanto a condiciones legales a las que se debe regir el producto, al tratarse de una aplicación como tal tiene que cumplir unas normativas claras:

- Permisos y condiciones de uso. Consiste en explicar claramente al usuario de la aplicación los permisos que debe aceptar este a la hora de utilizar la app, como puede ser el acceso a sus datos, historial de búsqueda, etc. Todo esto para que la experiencia sea más personal y con ello más inmersiva [29].
- Privacidad y geolocalización. En este caso al tratarse de un producto en el que es importante saber la localización en el espacio del usuario y demás, hay que contar con la aceptación del usuario para poder hacer uso de esos datos [29].
- Derechos. Es muy importante disponer de las licencias de los recursos utilizados a la hora del desarrollo de la aplicación, como puede ser en este caso la licencia de Unity para desarrolladores, la de Mapbox, etc. De igual forma también es conveniente blindar de alguna manera el contenido y funcionalidad del producto para evitar así posibles copias futuras [29].

También es importante destacar algunos de los pequeños problemas que puede producir el uso de dispositivos de Realidad Virtual, como son los siguientes especificados en el propio manual de uso de la empresa Oculus para sus productos.

- El uso prolongado de dispositivos de este tipo puede producir problemas de salud relacionados con mareos, convulsiones, espasmos oculares o pequeños desmayos provocados por la luz. Son problemas que se puede producir en todo caso viendo la televisión o el uso de otros dispositivos, pero en este caso es más evidente [30].
- El peso que suponen los cascos o gafas de Realidad Virtual, así como el uso de los mandos y el constante movimiento que supone puede provocar si se hace un uso excesivo y prolongado en dolores musculares o en articulaciones [30].
- Si el usuario se encuentra inmerso en el entorno virtual durante un periodo de tiempo prolongado puede provocar una distorsión de la realidad y problemas psicológicos. De igual forma puede producir pequeños golpes con mobiliario o paredes si no se realiza en un espacio adecuado [30].

En estándares técnicos al tratarse de una tecnología en constante desarrollo, no hay muchos de los que se pueda destacar mucha información. Existe un estándar con el nombre de OpenXR, de la compañía Khronos que proporciona un estándar de motores con acceso nativo a un abanico muy amplio de dispositivos ya sean de Realidad Virtual, Mixta o Aumentada [31]. El problema viene precisamente del amplio número de dispositivos de este ámbito que hay en el mercado que necesitan de distintas plataformas

para llevar el desarrollo de una aplicación, lo que implica un aumento de los costes de una aplicación.

De ahí la importancia de OpenXR, surge como estándar o arquitectura común con la que toda plataforma de desarrollo pueda ser compatible con cualquier dispositivo. Diversas compañías del sector como Oculus, Nvidia o Intel están de acuerdo en llevar a cabo este estándar, pues permitirá que los desarrolladores y las empresas puedan centrarse en la mejora e innovación de sus productos y a su vez el sector se verá igual de favorecido [32].

7. PRUEBAS

En este apartado se van a exponer las distintas pruebas a realizar para poder comprobar el correcto funcionamiento del sistema, para ello se deben tener en cuenta cada uno de los requisitos especificados.

A pesar de que tras la codificación de cada uno de los apartados o funcionalidades de la aplicación se ha ido comprobando su funcionamiento estas pruebas en específico se realizan para los casos que mezclan más procedimientos y se asemejan a lo que un usuario final realizaría.

Cada una de las pruebas completas a realizar vendrán dadas por la especificación de los siguientes aspectos:

- **Identificador.** Identificador para poder diferenciar cada una de las pruebas, con el formato P-X siendo X un número entero.
- **Nombre.** Explicación breve del propósito de la prueba.
- **Procedimiento.** Consiste en enumerar y explicar cada uno de los pasos que el usuario realiza y la respuesta del sistema para ellos.
- **Observaciones.** Los posibles problemas u observaciones que se puede encontrar el usuario a la hora de realizar alguna acción.

El esquema de la tabla que va a seguir cada una de las pruebas es:

TABLA 7.1: PLANTILLA DE PRUEBAS

Identificador	
Nombre	
Procedimiento	Paso 1

	Paso ...
Observaciones	

A continuación, se realiza la especificación de cada una de las pruebas.

TABLA 7.2: PRUEBA P-1

P-1	
Nombre	Exploración del mapa con la interacción con los mandos, añadiendo marcadores.
Procedimiento	El usuario elige en la escena de inicio la interacción con los controladores de Oculus, pudiendo ver los controles en el panel correspondiente.
	A través de los mandos el usuario se mueve por las calles de la ciudad abriendo y cerrando los puntos de información.
	Se ponen 2 o más marcadores separados entre sí, observando la ruta generada y abriendo el menú de información adicional de la ruta.
	Por último, se entra en la vista de pájaro para poder ver de forma más amplia tanto marcadores, ruta y puntos de información.
Observaciones	El funcionamiento de todos los apartados concurre con total normalidad, teniendo en cuenta que la interacción elegida en el primer paso es todo lo relacionado con los puntos de información, es decir dentro del juego se pueden soltar los mandos y utilizar la interacción con las manos.

TABLA 7.3: PRUEBA P-2

P-2	
Nombre	Interacción y comprobación de todos los gestos manuales.
Procedimiento	Una vez elegida la interacción de seguimiento de manos, se comprueban los gestos de apuntar para desplazarse por el

	mapa, pudiendo abrir los puntos de información con el gesto de pinza y click.
	Tras esto con el gesto de Thumbs Up se entra en la vista de pájaro y con el Thumbs Down se vuelve a la de humano.
	El usuario o en este caso "tester" se mira la palma de la mano izquierda se abre el menú con opciones de poner marcador, borrar, eliminar todos, etc. Funcionando correctamente cada uno de ellos.
Observaciones	En el caso de la interacción con gestos manuales pueden surgir pequeños problemas o falta de identificación de los gestos si se realizan a una velocidad elevada o con falta de iluminación, pues ese reconocimiento no será tan claro.

TABLA 7.4: PRUEBA P-3

P-3	
Nombre	Acciones con los sensores
Procedimiento	La interacción o acciones que se pueden realizar con el seguimiento de los sensores se unen con cualquiera de las dos interacciones anteriores no entran en conflicto.
	Al realizar el movimiento de "Walk in place" o caminar en el sitio, la cámara del entorno se mueve tantos pasos como se den y en la dirección correspondiente.
	Al dejar el objeto adicional en el suelo se añade un marcador a la escena como si de otra implementación anterior se tratara.
	Al levantar el brazo o la mano derecha por encima de la cabeza se detecta que se quiere entrar en la vista de pájaro.
Observaciones	Al igual que en la prueba anterior los gestos deben realizarse de forma clara y tranquila para que el sistema los detecte correctamente.

TABLA 7.5: PRUEBA P-4

P-4	
Nombre	Acciones con la maqueta del mapa.
Procedimiento	Continuando con el primer caso de prueba y la interacción con las Oculus Touch, se entra en la vista de pájaro.
	Una vez en la vista de pájaro se entra en la vista de la maqueta del mapa, consiguiendo moverse alrededor de esta y señalando con el puntero el lugar deseado.
	Con la localización seleccionada y pulsando el botón A, el jugador se teletransporta al lugar indicado con la vista de humano y seguir realizando cualquiera de las acciones disponibles en el sistema.
Observaciones	Para poder entrar en el modo de la maqueta hay que pasar previamente por la vista de pájaro, siguiendo un flujo de ir viendo el mapa cada vez más pequeño. Esta interacción puede ser muy útil para mapas de grandes dimensiones y que al usuario se le pueda hacer pesado las demás opciones.

TABLA 7.6: PRUEBA P-5

P-5	
Nombre	Viaje a otras ciudades
Procedimiento	A través del menú disponible al mirar la palma de la mano izquierda, se pueden elegir otras dos ciudades a las que “viajar” e interaccionar.
	Pulsando cualquiera de las opciones y gracias al Scene Controller se carga la escena y mapa correspondientes, ya sea París, Nueva York o Madrid.
	Una vez cargado el mapa se puede interactuar de igual forma con cada una de las opciones disponibles.
Observaciones	Para la carga del nuevo mapa es posible que aparezca una espera breve de un par de segundos.

A pesar de haber dividido las pruebas dependiendo de interacciones y posibles casos de uso reales, toda la funcionalidad completa del sistema se puede realizar sin que el usuario tenga que salir de la aplicación.

Se ha seguido este esquema para proporcionar ejemplos en los que el flujo de la aplicación parece más intuitivo, pero sin problema se pueden cambiar entre interacciones, vistas, marcadores, ciudades, etc. sin necesidad de reiniciar el juego ni cambiando ningún aspecto.

Gracias a las pruebas se comprueba que se cumplen los requisitos especificados y por lo tanto la aplicación cumple la funcionalidad completa.

8. PLANIFICACIÓN

Para la especificación de la planificación del proyecto se van a tener en cuenta cada una de las etapas que se han seguido, y para ello en primer lugar se explicarán estas tareas y se mostrará una tabla con el denominado diagrama de Gantt.

Estos diagramas son de gran ayuda a la hora de gestión de proyectos software siguiendo el método ágil, a pesar de que su utilización es muy amplia en empresas de software donde trabajan sobre un mismo proyecto varias personas simultáneamente. En este caso el desarrollo no ha sido a través de un equipo, pero aun así es una herramienta muy útil para el seguimiento del proyecto y planificación.

El diagrama está dividido a grandes rasgos en dos partes, la izquierda que contiene todas las tareas a realizar y que a continuación se van a enumerar y la derecha con barras que representan las horas de trabajo. Pudiendo incluir también fechas, encargados de las tareas o incluso dependencias [33].

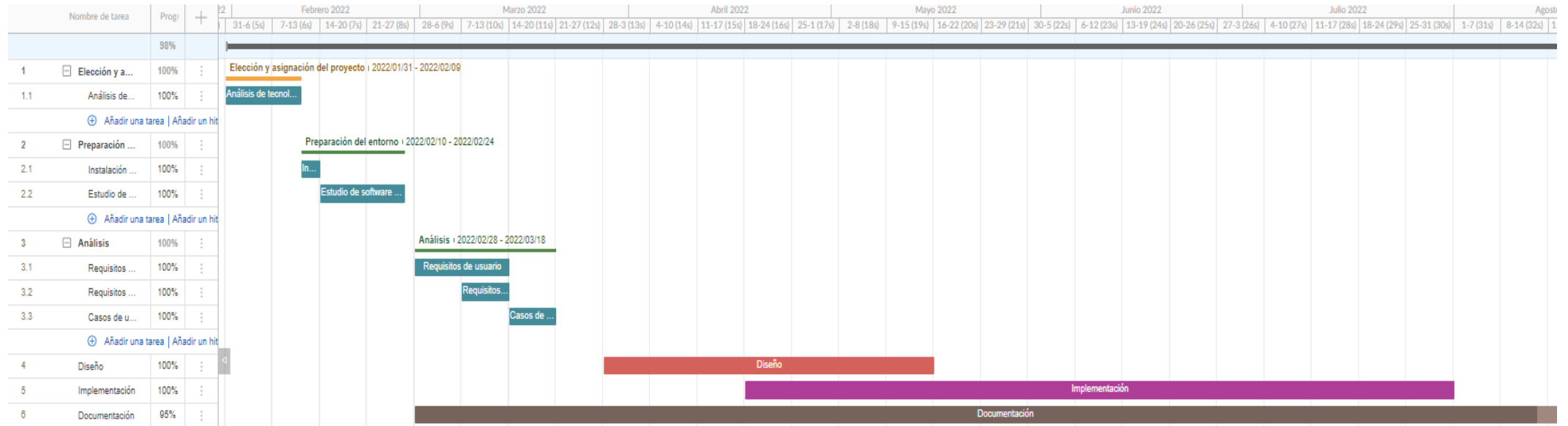
El desarrollo del proyecto contiene las siguientes etapas:

- **Elección y asignación del proyecto.** Se decide sobre cuál es el trabajo que se va a llevar a cabo, analizando trabajos similares e intentando diferenciarlo del resto.
 - **Análisis de tecnologías disponibles.** Se estudian las distintas tecnologías que pueden emplearse para el desarrollo del sistema, así como las limitaciones y ventajas que ofrecen los proyectos de Realidad Virtual.
- **Preparación del entorno.** Se divide tanto en la instalación del software necesario y su estudio:
 - **Instalación del software.** Consiste en la instalación del software utilizado tanto Unity, como Visual Studio y todas las librerías necesarias en Unity como Mapbox, Antilateny, etc.

- **Estudio del software y entorno.** Estudio y pruebas para familiarizarse con Unity, su flujo de trabajo y comprender el desarrollo de aplicaciones de Realidad Virtual.
- **Análisis.** Como su propio apartado en este documento se trata del análisis de la viabilidad del sistema, con apartados imprescindibles en un proyecto software como:
 - Requisitos de usuario. Especificación de los requisitos que establecen lo que ofrece el sistema dados por el usuario.
 - Requisitos de software. Definición de los requisitos que describen los servicios que debe ofrecer el sistema.
 - Casos de uso. Descripción de las acciones que puede realizar el usuario con diagramas de casos de uso.
- **Diseño.** Etapa en la que se realiza el diseño del sistema, desde la arquitectura global de la aplicación a la interfaz y elementos gráficos que van a componer la misma.
- **Implementación.** Consiste en la implementación propia de toda la funcionalidad del sistema para cumplir con los requisitos proporcionados en el apartado del análisis. Es la que supone el mayor porcentaje del trabajo.
- **Documentación.** Realización del documento actual con toda la información relacionada con el proyecto.

Tras establecer cada una las etapas y tareas a realizar, se concluye con la planificación temporal a través del ya nombrado diagrama de Gantt.

Fig. 8.1 Diagrama de Gantt de planificación del proyecto



9. CONCLUSIONES

En este apartado se van a definir tanto las conclusiones alcanzadas después del desarrollo del proyecto con su implementación y diseño como algunos aspectos a tener en cuenta en posibles trabajos futuros.

Como primer elemento a destacar se encuentran los dispositivos de Realidad Virtual, más en concreto las Oculus Quest 2. Son dispositivos que a pesar de que su popularidad está aumentando, su uso no está igual de extendido que el de los smartphones o PC, por ello los primeros clientes potenciales son un número inferior. A esto se les añade el difícil acceso a estas tecnologías, las HMD o gafas su precio puede rondar los 300 o 400 euros, pero por ejemplo el caso de este proyecto el entorno con los diferentes sensores y configuraciones es considerablemente de más difícil acceso.

Sin embargo, la Realidad Virtual presenta en el desarrollo de aplicaciones posibilidades nunca vistas y que pueden ser diferenciales de cara al futuro ofreciendo sensaciones como inmersión o presencia en entornos no reales con la capacidad de abstenerse de la realidad, aspectos muy difíciles de encontrar en otras tecnologías. Al tratarse de un mundo (el de la Realidad Virtual) todavía en desarrollo es complicada la interacción con las tecnologías al ser poco intuitivas en su aprendizaje.

9.1 Experiencia personal

La realización de este proyecto ha permitido la adquisición de múltiples conocimientos en un ámbito que era algo desconocido en el caso personal, pues como ya se ha mencionado la Realidad Virtual, Aumentada o Mixta no está de momento al alcance de todo el mundo, gracias a este trabajo y con la ayuda de los elementos proporcionados por la universidad y el tutor más en concreto ha sido posible el acceso a estas tecnologías.

Se ha llevado a cabo investigación y búsqueda de información en lo relacionado con el desarrollo de este tipo de aplicaciones, herramientas necesarias, etc. que a lo largo del curso académico no se habían visto. Elementos como Unity, herramienta indispensable para el desarrollo de videojuegos y aplicaciones de Realidad Virtual con todas sus librerías y APIs, los flujos de implementación o elementos del propio, a grandes rasgos cuál es el proceso de la creación de un videojuego o aplicación de estas características.

Ha sido una experiencia muy enriquecedora de aprendizaje descubriendo nuevas posibilidades tecnológicas y elementos en los que seguir trabajando e investigando en un futuro fuera de este proyecto.

9.2 Trabajos futuros

Como se ha mencionado con anterioridad al tratarse de este tipo de tecnologías que se encuentran en desarrollo y se siguen sacando constantemente nuevos dispositivos mejorados con nuevos aspectos es complicado ofrecer una solución para todos ellos, pero sí tienen en común el poder mejorar la interacción con los mismos, por ejemplo, dispositivos más ligeros con mejor seguimiento de movimientos y sobre todo más accesibles para el público general.

Más en concreto con este proyecto se podría buscar otra interacción diferente a la de los sensores, que son más complejos de obtener que solo las gafas Oculus, por interacción con voz y comandos con los que realizar las acciones dentro del entorno virtual. De igual forma extender la aplicación a dispositivos más convencionales como smartphones que a pesar de no poder tener esas interacciones más inmersivas y llamativas como es en el caso de las gafas, sí se pueda disfrutar de esa exploración e interacción de mapas tridimensionales.

Por último, en el aspecto más gráfico se puede llegar a una mejora en los edificios y calles haciéndolos más realistas y que esa sensación de ir andando o volando por una ciudad sea completa, teniendo en cuenta el alto coste computacional de una aplicación de estas características, siendo complejo su desarrollo e implementación dispositivos no tan profesionales como ha sido el caso.

SUMMARY

1. Introduction

The evolution in technologies has allowed people to see aspects that would have been difficult for parents or grandparents to imagine years ago. One of these technologies is what is called Virtual Reality, consists in a computer and virtual environment with the use of glasses or helmets gives the user the sensation of being in that world. It's the feeling of being immersed in a world or environment that seems real but really is not.

In the last few years these devices have gone from being relegated to state-of-the-art technology more easily accessible to companies or universities, to solutions that can be accessed by a greater number of people. Mainly thanks to improvements in inconveniences involved in the use of Virtual Reality such as comfort or interaction, thus considerably increasing the number of end users.

Another consideration to take into account is that despite seeming to only being successful in the world of videogames, the range of possibilities within Virtual Reality goes far beyond games. Industries such as entertainment, multimedia (movies, videos, calls), tourism or health care are going to be year after year more important.

1.1 Motivation

The tourism sector that over the last 2 years has suffered greatly due to the COVID-19 pandemic due to the restrictions to contain the same. Even so, it's a sector that moves millions of people in which you can innovate to find other possibilities of traveling or encourage it.

The importance of the tourism industry is very marked since it represents the creation of around 270 million jobs and its earnings in the last year managed to exceed 4.5 billion dollars. But is not only what it generates, also attracts other companies, whether public or private to invest in the sector.

The main purpose of the application is being able to explore a plane or a 3D map, use Virtual Reality technology to interact with map data or points of interest in an immersive space. The advantages that Virtual Reality can offer in the visualization and interaction with maps is for example, the wide field of vision that it offers the user and makes him see himself much more immersed in the action without any type of distractions and being able to enjoy the process, as well as a more intuitive interaction with information and data.

All of these are reasons for individual users to use the application and being able of travel around the world without moving of their houses, but for companies could be a very interesting tool to stand out from the rest and offer a differential experience. For example travel organization companies could offer the user the possibility of consuming the route or trip that is going to hire in a virtual form, so the user could see what he is going to see in the real place. In the same way for other vent organization companies can present their ideas to the other collaborators with Virtual Reality, aspect that will make those idea look more innovative and more attractive to anyone.

1.2 Objectives

The objectives that want to achieve with the developing of this project are:

- Creating a Virtual Reality application or game in which being able to visualize maps in a 3D environment as if it was a real street, explore the map with different relevant information and being able to establish possible points of interest with their corresponding routes.
- Creation of a both physical and virtual environment in which correctly reproduce the application.
- Create different forms of interaction with the application to carry out various actions whether with controls, manual gestures, sensors, etc.
- Study of new forms of interactions with virtual maps and environments. Although not being a main objective, with the making of this project its going to explore the different Virtual Reality capacities even though it is a technology that is in constant development.

2. State of Art

2.1 Virtual Reality

There are different types of reality or concepts depending on whether you are in a completely virtual world or not. The space between the real and the virtual world is the Mixed Reality that depending on the proportion of virtuality a distinction is made between Augmented Reality and Augmented Virtuality. This project is focused on the Virtual Reality in which the simulated world is completely virtual.

Virtual Reality is a type of reality created by computer that responds to the user's inputs, for which interaction and respond in real time is key.

Inside of the Virtual Reality two aspects that were not perceptible at the beginning and currently essential for these systems come into play, are immersion and presence. Although is tough to know the differences between them, immersion refers to the sensory

level a Virtual Reality system provides from the objective point of view. This level of immersion could depend on the monitors or software that renders the system, can be measured. The presence is understood as something more psychological that depends individually on the user related with the experience of being in that virtual world.

2.2 Devices

In Virtual Reality there is a very wide range of devices capable of performing the necessary actions, so the user is as immersed as possible in the virtual environment. The most appropriate type for this is the called HMD devices (Head Mounted Display), devices that are placed on the head at eye level to observe the virtual world in all moments. They also have internal and external sensors to be able to control the position and rotation of the head and respond as faithfully as possible to reality.

Between the devices that were analyzed the most balanced according to specs and prize, was the Oculus Quest 2. They offer the necessary immersion with good graphics and quality; their price is not very high around 400 €. They have also been chosen because were already owned by the university and tutor, which will be great help at development time.

2.3 APIs/Library

In the development process the use of these libraries or APIs make it way easier thanks to the help that they provide. They are resources developed by other companies or by third parties that help the process. In this project two of the most important are the Oculus SDK and one for VR Maps called Mapbox.

- **Oculus SDK.** It is an asset available in Unity itself, that is the development platform for many of the games that the people play every day, that provide different facilities when developing a Virtual Reality project. It contains tools for the recognition of inputs of the control panels or manual gestures, apart from this it also provides already made models or prefabs with their physics such as the camera with 360° vision.
- **Mapbox.** It is an open-source solution for creating 2D or 3D maps and exists an extension to Unity to help with the process. The way of working is creating a 3D model from a location point (latitude & longitude) and its surroundings, what makes it different is the possibility of configuring the information that appears on the map and the option of creating points and making the most optimal route between them.

2.4 Similar Projects

To know where the application that is being developed is located, it is necessary to investigate what other solutions or similar examples there are. With this it will be possible to make this project with different characteristics that make it different from others.

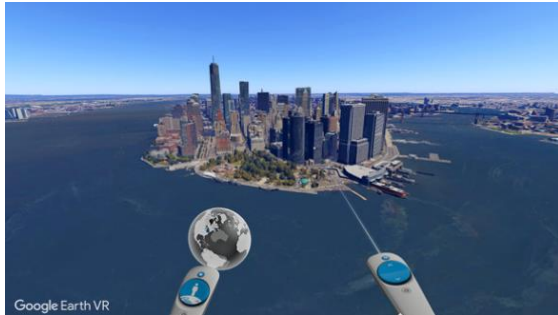


Fig A.1. Google Earth VR image

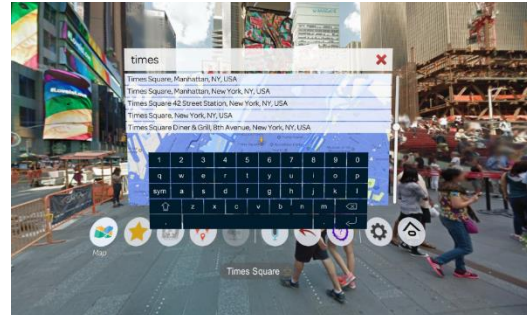


Fig A.2. Wander application for Oculus image

There are two similar projects according to the description of an application to explore 3D maps. This first one is called Google Earth VR, solution created by Google like Google Maps, in which you can explore any point in the real world. The images are taken by satellite and are faithful to reality, the only novelty is that is adapted to the movement and visualization with Virtual Reality devices. The app allows the exploration of the terrain at different heights as flying and being able to choose where to travel by clicking on the available globe.

The other one is Wander, consists of an application available for Oculus devices and simulates traveling around the world. The main function is to explore different locations giving the possibility of reading information about a point of interest. The main issue is that it is not the consumption of a 3D model created for Virtual Reality, what is really seen are normal images such as Street View provided by Google Maps.

With all of these is clearly to see that the actual project it differs from above adding the different interactions taking advantages of Virtual Reality, not like the examples explained. The other thing to take into account is that on these examples is more important the reality of the images or models, but the solution of the project find a balance between those models making them less realistic and offer the possibility of different interactions in which the user can feel immersed in the environment.

2.5 Input/Output Devices

For the interaction with the Virtual Reality systems and the environment that is created, various devices are needed to make it possible. Within these are the glasses that are the most important part which provides all the immersion but also have sensors to detect head movements and hand gestures. Of one the first possibilities of inputs are the hand gestures with which to do most of the actions in the world.

The Oculus Touch are the controls of the Oculus Quest 2, they have joysticks to be able to move as if a traditional console control is treated as buttons, the differential is that they have sensors both for the fingers and on the back so they could act as pointers.

The last type of interaction and input device are the Antilatency sensors, are sensors compatible with development in Unity and Virtual Reality. They can be attached to any part of the user's body to track arm or leg movements, or to an object and make a ball appear in the virtual environment to make it more realistic. Similar process to how motion captures are made in videogames.

3. Analysis

The study of the application and the specification of the requirements that it must meet are the initial steps in the analysis and development of a software project.

First, a general overview of the program will be given, along with an explanation of what the system can provide for users and how users can react. It's a program for Virtual Reality that allows users to view, explore and interact with three-dimensional maps. Through his Oculus glasses, the user is submerged in a 3D environment that resembles a city with its buildings and streets.

The user can explore marked points of interest and read a brief description of these, view the map from different heights or even being able to choose other cities, such as Paris or New York, to travel thanks to the incorporated arm menu.

The user and software requirements, as well as potential use cases, should all be specified in this part so that you can see how they relate to one another and ensure that the program was developed correctly.

The user requirements define both the capabilities and constraints of the system, what the user can do in the environment. For example, two requirements explained in section 3 of the document:

- From a greater height, the user can view and navigate the map as though it were a bird. Less detail is present in map elements.
- A yellow path is formed when the user creates two or more markers and spaces them apart; it connects the markers and determines the shortest path between them.

On the other hand, the software requirements establish what the system can offer, they are more detailed and contemplate all the possibilities that are, for example, for the same action.

With the use cases the different scenarios that can occur using the application are simulated and shows the flows of interaction and which states are necessary for others. For instance, you cannot move to a view mode while in another one or delete a bookmark that has not yet been established.

4. Design and Implementation

4.1 Architecture

Programming by layers is a development model that can be used, as it is in most software projects. It basically entails segmenting the system into various layers or distinct parts. On this project, the architecture is divided in:

- **Presentation Layer.** It represents what the user sees and serves as a link between them and the system. It is the most graphical component of the application because it is in charge of giving the user the information and recording their inputs.
- **Business Layer.** It is the section that houses the application logic, which analyzes what the Presentation layer captures, executes the necessary operations, and then returns the specified result.

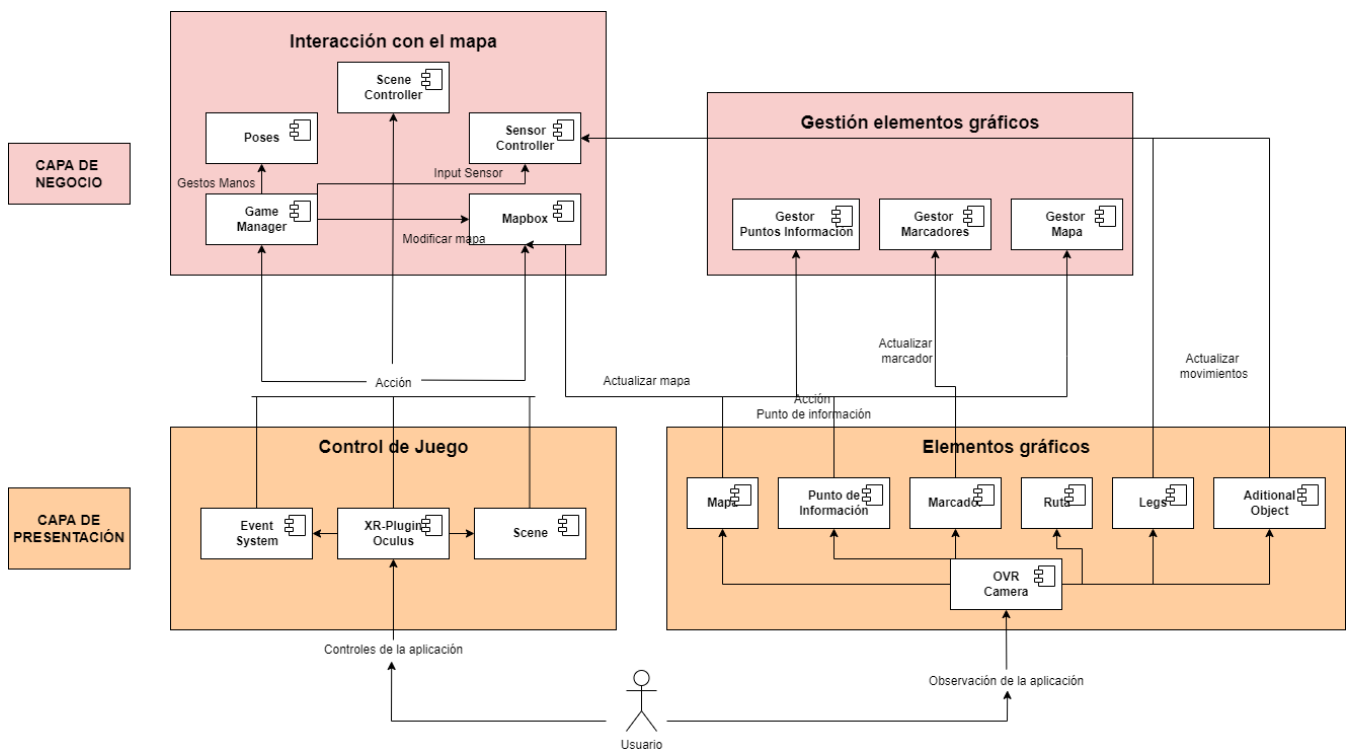


Fig. A.3. System Architecture diagram

Each layer has a subsystem, as shown in the picture, and each subsystem is composed of several components. For example, the game control subsystem and the graphic element subsystem, for instance, can be differentiated in the presentation layer. Or in the business layer, the Interaction subsystem with the map and the management of graphic elements are part of this.

4.2 Components

There are each of the parts that make up the system and also were shown in the last diagram. Some of these can be:

- **XR-Plugin Oculus.** It is the part responsible for gathering user inputs from controls, hands, etc.
- **OVR Camera.** A feature of the application's camera. You wouldn't be able to appreciate any of the system's graphic features without it, contains items like hands, game controllers, etc.
- **Information point manager.** Is in charge of tracking which information points the user wishes to open or close and read.

4.3 Implementation

This chapter entails outlining the choices that were made in order to fulfill the specifications that were previously outlined during the application's development. The application can be divided into some of the following sections:

- **Scenes.** They serve as the focal points around which all other factors are centered. They can be compared to the many levels in a game. Since there are no levels in this application, each scene has been implemented as each of the environments that are available. For example, the three-dimensional map of Madrid that is included by default is a scene, and if it is changed to another city, such as Paris, it would be loading its corresponding scene.
- **Game-Manager.** The majority of the system's functionality is controlled by this class, which oversees filtering and identifying user inputs before performing the necessary actions or delegating responsibility to another class. All this recognition of actions is found in the Update function, the way in which this function works is by checking in each frame if any of the conditions developed are met, if they do another function is called that will be the one that performs the action in the virtual environment, such as setting a marker or changing the view.
- **Route and Markers.** With the aid of the Mapbox SDK, it is provided. This includes a class called "DirectionsFactory" that, given a list of items and their related positions, finds the best path between them while considering the streets shown on the map. When creating a marker, an object is added to the list of

markers, checked to ensure that there aren't any duplicates there or separated by less than 10 meters, and that the list contains two or more markers before calling the method "Start()" of the Mapbox class to build the corresponding route.

- **Manual gestures.** First, the shape of the hand is created considering each of the fingers, which can be closed, open or bent. After the form has been made, a model that specifies which hand can perform it, the wrist's position, and the hand's palm must also be made. One of these components is launched when it detects that all requirements are completed and the required function may be called, making the process easier to see and understand owing to Unity's components and SDK.

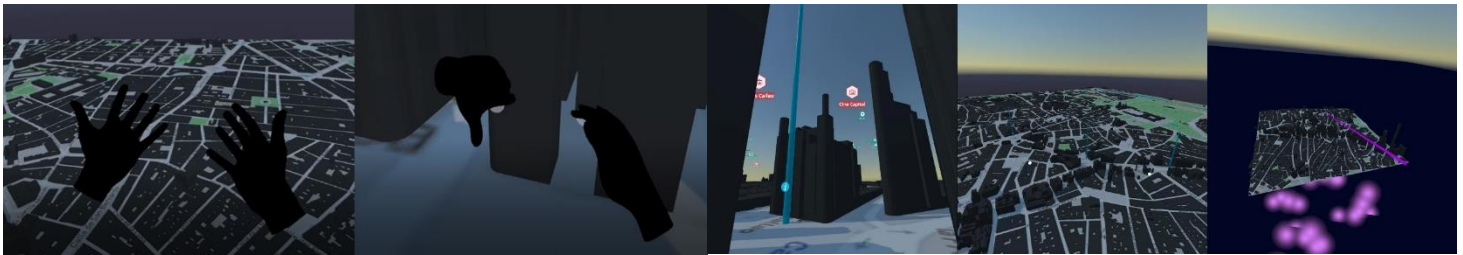


Fig. A.4. Images in game of the application

5. Socio-economic environment

Prior to discussing the application's potential effects on the economic and social landscape, it is important to highlight and examine the potential consequences of using this product. It is simple to see how useful it will be in the tourism industry, whether it is for a single person who wants to plan their own vacation or for travel agencies. With the application, they can provide a unique and user-friendly way for visitors to see what to expect when they arrive in the city, and it can act as a guarantee for customers who choose to work with or hire that company.

At an economic level, it does not have a great impact, since it is not replacing a similar or directly competitive product, therefore, it cannot be bettered in price. But regarding the social consequence, it is simpler to recognize that it signifies a shift in how people engage with and navigate fully immersive maps. Additionally, by examining the various ways in which a person can interact with a map and its data interactions that go beyond merely viewing images on a computer screen.

5.1 Budget

The budget necessary for the development of the project, dividing the costs for the personal staff required and the resources (development computer, Oculus Quest 2, Antilatency's trackers, etc.), the total cost amounts around 12.429 €.

6. Regulatory Framework

It acts as a study of the laws surrounding the project as well as certain relevant workplace regulations. As every other application, the product must meet the conditions:

- **Permissions and conditions of use.** It consists of clearly explaining to the application user the permissions that they must accept when using the app.
- **Geolocation and privacy.** In this instance, having the user's consent is required to utilize the data because the product in question requires knowledge of the user's location within their space in addition to other factors.
- **Rights.** It is crucial to have the licenses for the materials needed to create the application, such as the Mapbox license, the Unity license for developers, etc.

It's necessary to draw attention to a few of the minor issues that using VR equipment can bring about. For example, long-term usage of these gadgets may result in health issues like dizziness, seizures, ocular spasms, or light-induced blackouts. In the same way, the use of Virtual Reality glasses, being heavy and the controls, can cause muscular problems, or long-term immersion in a virtual environment can lead to psychological issues and a distortion of reality in the user.

7. Planning

Each stage that has been followed will be taken into consideration for the project planning specification. These stages are choice and assignment of the project, analysis of available technologies, preparation of the environment (software study and installation), analysis (requirements), design, implementation, and documentation.

Thanks to the help of a Gantt chart (available in the planning section of the document), the period for carrying out the work is established from the end of February to the beginning of September.

8. Conclusions

The virtual reality devices are the first thing to stand out. Despite their growing popularity, these gadgets are not as widely used as smartphones or personal computers, which results in a smaller pool of first potential clients. The cost of HMDs or glasses, which can be approximately 300 or 400 euros, is added to this along with the difficulty of gaining access to these technologies. For instance, in the case of this project, the environment with the many sensors and setups is much more challenging to access. On the other hand, it offers sensations like immersion or presence in unreal surroundings with

the potential to avoid reality, features that are very rare to discover in other technologies, and it introduces possibilities never seen before and that can be differentiating for the future.

8.1 Personal experience

Research and information gathering have been done regarding the creation of these kinds of programs, tools, etc. that they had not interacted with throughout the academic year. Elements such as Unity with all its libraries and APIs, the implementation flows or elements of its own, in broad strokes what is the process of creating a video game or application of these characteristics.

Summing it up finding new technology options and components on which to focus future work and research outside of this project has been an enlightening learning experience.

8.2 Future works

It is challenging to provide a solution for all these technologies because new, upgraded gadgets with new features are continually being developed. However, they do share the ability to enhance engagement with them. Lighter devices with better movement tracking and above all more accessible to the public.

Regarding this project the use of voice interaction and orders to carry out tasks within the virtual environment could be another interaction that differs from that of the sensors. Also expand the software to more traditional devices so that you can still explore and engage with three-dimensional maps on cellphones even though they lack the immersive and spectacular interactions that glasses can provide.

Last but not least, in the most graphic element, it is possible to improve the buildings and streets, making them more realistic and completing the experience of walking or flying through a city, taking into account if the high computational cost.

BIBLIOGRAFÍA

- [1] A. Jenkins, «The fall and rise of VR: The struggle to make virtual reality get real | Fortune», 20 de junio de 2019. <https://fortune.com/longform/virtual-reality-struggle-hope-vr/>.
- [2] S. Rogers, «2019: The Year Virtual Reality Gets Real», *Forbes*. <https://www.forbes.com/sites/solrogers/2019/06/21/2019-the-year-virtual-reality-gets-real/>.
- [3] P. Graham, «Superdata Releases New VR Market Forecasts», *GMW3*. <https://www.gmw3.com/2016/03/superdata-releases-new-vr-market-forecasts/>.
- [4] G. Moreno, «Infografía: No todo es un juego en la realidad virtual», *Statista Infografías*. <https://es.statista.com/grafico/7433/no-todo-es-un-juego-en-la-realidad-virtual/>.
- [5] «El turismo en España y en el mundo, en datos y gráficos». <https://www.epdata.es/datos/turismo-espana-mundo-datos-graficos/272>.
- [6] A. Orús, «Tema: El turismo en el mundo», *Statista*, 22 de octubre de 2021. <https://es.statista.com/temas/3612/el-turismo-en-el-mundo/>.
- [7] «Antilatency tracking system for VR/AR, Virtual Production, Drones and Robotics», *Antilatency*. <https://antilatency.com/>.
- [8] P. Milgram y F. Kishino, «A Taxonomy of Mixed Reality Visual Displays», *IEICE Trans Inf. Syst.*, vol. E77-D, n.º 12, pp. 1321-1329, dic. 1994.
- [9] G. C. Burdea y P. Coiffet, *Virtual Reality Technology*. John Wiley & Sons, 2003.
- [10] D. A. Bowman y R. P. McMahan, «Virtual Reality: How Much Immersion Is Enough?», *Computer*, vol. 40, n.º 7, pp. 36-43, jul. 2007, doi: 10.1109/MC.2007.257.
- [11] «HTC Vive PRO: Análisis y Opiniones de las Gafas de Realidad Virtual». <https://gafasrealidadvirtual.pro/htc-vive-pro/>.
- [12] E. Pérez, «Las nuevas Oculus Rift S mejoran la tecnología de rastreo y eliminan el lío de cables para ser unas gafas VR más fáciles de utilizar», *Xataka*, 21 de marzo de 2019. <https://www.xataka.com/realidad-virtual-aumentada/oculus-rift-s-caracteristicas-precio-ficha-tecnica>.
- [13] «Comprar Valve Index el visor de VR con mejor Tracking», *Alehandoro VR*. <https://alehandorovr.com/valve-index-y-accesorios/>.
- [14] J. García, «Oculus Quest 2, análisis: una de las mejores (y asequibles) opciones para iniciarse en la realidad virtual», *Xataka*, 17 de octubre de 2020. <https://www.xataka.com/analisis/oculus-quest-2-analisis-caracteristicas-precio-especificaciones>.
- [15] «Unreal Engine | Features», *Unreal Engine*. <https://www.unrealengine.com/en-US/features>.
- [16] E. Arts, «Star Wars Jedi: Fallen Order™ - Una nueva aventura de Star Wars™ - Sitio oficial de EA», *Electronic Arts Inc.*, 29 de abril de 2022. <https://www.ea.com/es-es/games/starwars/jedi/jedi-fallen-order>.
- [17] «Say cheese! Here's how to grab the best screenshots». <https://www.epicgames.com/fortnite/en-US/news/photography-blog>.
- [18] «Oculus Integration | Integration | Unity Asset Store». <https://assetstore.unity.com/packages/tools/integration/oculus-integration-82022>.

- [19] «Online Maps | Unity Map Asset | Infinity Code». <https://infinity-code.com/assets/online-maps>.
- [20] «Maps SDK for Unity Overview | Gaming Maps Platform Gaming Services», *Google Developers*.
https://developers.google.com/maps/documentation/gaming/overview_musk.
- [21] «Maps SDK for Unity: 3D worlds, AR, & POIs | Mapbox». <https://www.mapbox.com/unity>.
- [22] «ArcGIS Maps SDK for Unity | ArcGIS Developers», *ArcGIS Maps SDK for Unity*. <https://developers.arcgis.com/unity/>.
- [23] «Traffic and directions | Maps SDK | Unity», *Mapbox*.
<https://docs.mapbox.com/unity/maps/examples/traffic-and-directions/>.
- [24] M. Berger y R. Bill, «Combining VR Visualization and Sonification for Immersive Exploration of Urban Noise Standards», *Multimodal Technol. Interact.*, vol. 3, n.º 2, Art. n.º 2, jun. 2019, doi: 10.3390/mti3020034.
- [25] «Google Earth VR». <https://arvr.google.com/earth/>.
- [26] «Wander (Quest)». <https://www.realovirtual.com/rovdv/aplicaciones/15017/wander>.
- [27] *Antilatency wireless full body tracking*, (16 de julio de 2019). Accedido: 15 de junio de 2022. [En línea Video]. Disponible en:
<https://www.youtube.com/watch?v=RpAYMS5axkA>
- [28] S. Butler, «Omnidirectional Treadmills Could Boost VR Immersion (But They're Impractical)», *How-To Geek*.
<https://www.howtogeek.com/805135/omnidirectional-treadmills-vr-immersion/>.
- [29] «Los 8 requisitos legales para hacer una app», *Via empresa*.
http://www.viaempresa.cat/es/innovacion/los-8-requisitos-legales-para-hacer-una-app_15002_102.html.
- [30] «Advertencias de salud y seguridad | Meta Quest | Meta Store». https://store.facebook.com/es/es/legal/quest/health-and-safety-warnings/?utm_source=e-archivo.uc3m.es&utm_medium=oculusredirect.
- [31] thetuvix, «OpenXR - Mixed Reality». <https://docs.microsoft.com/es-es/windows/mixed-reality/develop/native/openxr>.
- [32] «Khronos Announces VR Standards Initiative», *The Khronos Group*, 6 de diciembre de 2016. <https://www.khronos.org/news/press/khronos-announces-vr-standards-initiative>.
- [33] Atlassian, «Tutorial completo sobre los diagramas de Gantt», *Atlassian*.
<https://www.atlassian.com/es/agile/project-management/gantt-chart>.