

DIVISIÓN DE ESTUDIOS POSGRADO E INVESTIGACIÓN

Opción 1.- TESIS
TRABAJO PROFESIONAL

“Aplicación de realidad aumentada usando un sistema de generación de mallas y detección de planos para mejorar la experiencia turística”

**QUE PARA OBTENER EL GRADO DE:
MAESTRO EN INGENIERÍA ELECTRÓNICA**

PRESENTA:
JOSE EDGARDO MARTÍNEZ FERNÁNDEZ

DIRECTOR DE TESIS:
DR. IGNACIO HERRERA ÁGUILAR

CODIRECTOR DE TESIS:
DR. GERARDO ÁGUILA RODRÍGUEZ





EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Instituto Tecnológico de Orizaba
División de Estudios de Posgrado e Investigación

Orizaba, Veracruz, 25/septiembre/2023

Dependencia: División de Estudios de
Posgrado e Investigación

Asunto: Autorización de Impresión
OPCION: I

C. JOSÉ EDGARDO MARTÍNEZ FERNÁNDEZ

Candidato(a) a Grado de Maestro(a) en:

INGENIERÍA ELECTRÓNICA

P R E S E N T E.-

De acuerdo con el Reglamento de Titulación vigente de los Centros e Institutos Tecnológicos Federales del Tecnológico Nacional de México, de la Secretaría de Educación Pública, y habiendo cumplido con todas las indicaciones que la Comisión Revisora le hizo respecto a su Trabajo Profesional titulado:

" APLICACIÓN DE REALIDAD AUMENTADA USANDO UN SISTEMA DE
GENERACIÓN DE MALLAS Y DETECCIÓN DE PLANOS PARA MEJORAR LA
EXPERIENCIA TURÍSTICA"

Comunico a Usted que este Departamento concede su autorización para que proceda a la impresión del mismo.

ATENTAMENTE

Excelencia en Educación Tecnológica®
CIENCIA - TÉCNICA - CULTURA®

Cuahtémoc Sánchez R.

Dr. CUAUHTÉMOC SÁNCHEZ RAMÍREZ
JEFE DE LA DIVISIÓN DE ESTUDIOS
DE POSGRADO E INVESTIGACIÓN



OG-13-F06



Av. Oriente 9 Núm.852, Colonia Emiliano Zapata. C.P. 94320 Orizaba, Veracruz.

Tel. 01 (272)1105360 e-mail: dir_orizaba@tecnm.mx tecnm.mx | orizaba.tecnm.mx



2023
AÑO DE
Francisco
VILLA



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Instituto Tecnológico de Orizaba
División de Estudios de Posgrado e Investigación

Orizaba, Veracruz, 4/septiembre/2023
Asunto: Revisión de trabajo escrito

C. CUAUHTÉMOC SÁNCHEZ RAMÍREZ
JEFE DE LA DIVISIÓN DE ESTUDIOS
DE POSGRADO E INVESTIGACIÓN
PRESENTE.-

Los que suscriben, miembros del jurado, han realizado la revisión de la Tesis del (la) C.

JOSÉ EDGARDO MARTÍNEZ FERNÁNDEZ

La cual lleva el título de:

**"APLICACIÓN DE REALIDAD AUMENTADA USANDO UN SISTEMA DE GENERACIÓN DE
MALLAS Y DETECCIÓN DE PLANOS PARA MEJORAR LA EXPERIENCIA TURÍSTICA"**

Y concluyen que se acepta.

ATENTAMENTE

Excelencia en Educación Tecnológica®
CIENCIA - TÉCNICA - CULTURA®

PRESIDENTE: DR. IGNACIO HERRERA AGUILAR

FIRMA

SECRETARIO: DR. GERARDO ÁGUILA RODRÍGUEZ

FIRMA

VOCAL: M.C. ENRIQUE ALEJANDRO CUELLAR
CORTÉS

FIRMA

VOCAL SUP.: DR. JOSÉ PASTOR RODRÍGUEZ JARQUIN

FIRMA

TA-09-18



Av. Oriente 9 Núm.852, Colonia Emiliano Zapata. C.P. 94320 Orizaba, Veracruz.
Tel. 01 (272)1105360 e-mail: dir_orizaba@tecnm.mx tecnm.mx | orizaba.tecnm.mx



2023
Francisco
VILLA

Dedicatorias.

"So tonight, you better stop and rebuild all your ruins, because peace and trust can win the day despite of all you losing". Led Zepellin

A mi madre M.I.A **Celia Fernandez Vásquez** que aun con problemas y su trabajo siempre mostro apoyo, por darme una buena vida y gran salud, así como sabiduría en cada paso que fui dando este camino. A mi madre abuela **Alicia Vásquez Calderón** que no importando lo que sea siempre está detrás de mí y a mi padre abuelo **Jose Armando Fernandez Cruz †** que siempre me guio para ser un gran profesionista y aun que ya no este presente en cuerpo se que el esta orgulloso de mi y siempre estará presente en mi corazón en cada día que pase.

A mi novia y futura esposa Ing. Yanil Badillo Malpica por cada animo que me dio en cada paso que fui dando, por toda la motivación que me daba cada vez que tenia pensado tirar la toalla, por aguantarme ni cuando yo mismo me soportaba y por siempre ayudarme a sonreír y por siempre ayudarme a caminar cuando ya ni yo mismo podía lograrlo gracias totales mi amor!

A mis amigos el Lic. Daniel Heredia Barrera y el Ing. Cesar Elías Carreño del Ángel que siempre han sido mas que mis amigos siendo mis hermanos en cada momento y acompañándome en cada momento y dándome motivación para seguir adelante, gracias como siempre por esas risas interminables y por acompañarme en cada locura.

Y finalmente me dedico este trabajo a mí, porque aún con todas las adversidades que se me presentaron, los momentos en que deje de creer en mí, aun bajo cada tropiezo que tuve y luchando contra mi mismo para lograrlo aun sigo sin creer que pudiera lograrlo y de verdad agradezco a dios el darme la fuerza para alcanzar este gran objetivo.

Agradecimientos.

“A veces te subes a la cama por las mañanas y piensas: no voy a lograrlo. Pero te ríes por dentro recordando, todas las veces que te has sentido de esa manera”.

Charles Bukowski

Quiero agradecer a mi director de tesis el **Dr. Ignacio Herrera Aguilar** por permitirme la oportunidad de formar parte de la maestría, al mismo tiempo por sus asesorías y consejos durante el desarrollo de esta tesis, su paciencia y el impulsarme a no rendirme pese a las circunstancias del trayecto orientándome a lograr mi objetivo.

Agradezco al **Dr. Gerardo Águila Rodríguez** por sus excelentes críticas constructivas y observaciones que, me ayudaron en gran medida a dar orden y poder plasmar de manera correcta las ideas y partes metodológicas de este trabajo de investigación.

Al **Dr. Jose Pastor Rodríguez Jarquín** por su enorme disponibilidad de tiempo para la revisión de esta tesis, por sus aportes y ayuda extra que me brindo, por compartir sus conocimientos en el aula y ser un excelente profesor.

A la **M.C Blanca Estela Gonzales Sánchez** por su enorme paciencia y tutorías para orientarme a lograr una tesis mas eficiente y encontrar la manera adecuada de lograr un trabajo profesional para el desarrollo de esta tesis.

Finalmente agradezco al **Consejo Nacional de Ciencia y Tecnología (CONACYT)** por la beca no. 1107083 otorgada durante el periodo Abril del 2021 a febrero del 2023.

RESUMEN.

“Aplicación de realidad aumentada usando un sistema de generación de mallas y detección de planos para mejorar la experiencia turística”

Elaborado por: ***Ing. Jose Edgardo Martinez Fernandez***

Dirigido por: ***Dr. Ignacio Herrera Aguilar***

Dr. Gerardo Águila Rodríguez

En el sector turístico existen diversos servicios, que incluye atractivos, accesos y otros elementos. Las atracciones turísticas no solo generan ventas, sino que también venden la experiencia que el visitante puede tener. Estas atracciones pueden provenir de sitios naturales, museos, monumentos históricos, manifestaciones culturales, eventos y logros técnicos de personas destacadas. La oferta turística de las comunidades es una opción interesante en el mercado, ya que sus productos valoran la identidad cultural y los intercambios vivenciales, permitiendo a los visitantes experimentar nuevas atracciones y conocer más sobre la historia detrás de los lugares. Con el objetivo de mejorar la aplicación Pepe creada por la compañía Abad Médico S.A. de C.V., se propone implementar un sistema de realidad aumentada que utilice puntos de interés para ofrecer una alternativa tecnológica que brinde entornos digitales realistas que el ojo humano pueda percibir como tal, utilizando solo la cámara de un celular. Esta investigación surge de la necesidad de destacar la importancia histórica de sitios olvidados o con poca importancia turística dentro de la ciudad, así como logros o aportaciones de personas destacadas para la ciudad. Con esta mejora, se busca darles un mayor realce y atractivo turístico.

PALABRAS CLAVE: Realidad aumentada, Realidad virtual, Turismo, Unity 3D, AR FOUNDATION, NIANTIC

ABSTRACT.

“Application of augmented reality using a mesh generation and plane detection system to improve the tourist experience”.

By: Ing. Jose Edgardo Martinez Fernandez

Advisor(s): Dr. Ignacio Herrera Aguilar

Dr. Gerardo Aguila Rodriguez

In the tourism sector there are several services, such as attractions, access and other elements. Tourist attractions not only generate sales, but also sell the experience that the visitor can live. These attractions can come from natural sites, museums, historical monuments, cultural manifestations, events and technical achievements of outstanding people. The communities' tourism offer is an interesting option in the market, as their products value cultural identity and experiential exchanges, allowing visitors to experience new attractions and learn more about the history behind the places. In order to improve the Pepe application created by the company Abad Médico S.A de C.V., it is proposed to implement an augmented reality system that uses points of interest to offer a technological alternative that provides realistic digital environments that the human eye can perceive as such, using only the camera of a cell phone. This research arises from the need to highlight the historical importance of forgotten sites or with little tourist importance within the city, as well as achievements or contributions of outstanding characters for the city. With this improvement, we seek to give them a greater enhancement and tourist attraction.

KEYWORDS: Augmented Reality, Virtual Reality, Tourism, Unity3D, AR Foundation, Niantic

INDICE

RESUMEN.	4
ABSTRACT.	5
INDICE.....	6
INDICE DE FIGURAS Y TABLAS.	7
CAPÍTULO 1.	12
1.Marco preliminar.	12
1.1 Introducción.....	12
1.2 Planteamiento del problema.	13
1.3 Justificación.....	13
1.4 Antecedentes.	14
1.5 Hipótesis.....	14
1.6 Metodología.....	15
1.7 Objetivos.....	16
1.7.1 Objetivo general.....	16
1.7.2 Objetivo específico.....	16
1.8 Aportes.....	16
1.9 Herramientas.....	17
1.10 Estado del arte.....	18
CAPÍTULO 2.....	23
2. Marco Teórico.....	23
2.1 Teoría de la realidad aumentada.....	23
2.2 Unity 3D y la tecnología de Realidad aumentada.	25
2.3 Unity 3D y como se puede utilizar en el turismo.	27
2.4 Experiencia turística y su mejoría con las tecnologías de unity.....	30
2.5 Tecnología unity y realidad aumentada para la promoción de destinos turísticos.	32
2.6 Desafíos y limitaciones para unity y la realidad aumentada en el turismo actual.....	33
2.7 Futuro de las tecnologías unity en el turismo usando realidad aumentada.....	35
CAPÍTULO 3.....	39
3. Desarrollo teórico.....	39
3.1 Descripción del sistema.....	40
3.2 Diseño de la interfaz.....	42
3.3 Diseño del administrador del juego.....	45
3.4 Diseño del control de sesión entrante al espacio de realidad aumentada.....	49
3.5 Diseño del sistema de mensaje de posicionamiento y comunicación de la sesión AR.	54

3.6 Diseño del sistema de interacción de realidad aumentada.....	57
3.7 Diseño del sistema controlador de objeto de realidad aumentada.	59
CAPÍTULO 4.....	61
4. RESULTADOS.	61
CAPITULO 5	74
5. CONCLUSIONES.	74
5.1 Trabajo a futuro	76
Apéndice A: Manual de diseño de la aplicación en unity.	77
Apéndice B: BIBLIOGRAFIA.....	102

INDICE DE FIGURAS Y TABLAS.

Figura 1: Diagrama de esquema de tesis.....	15
Figura 2: la continua evolución de la realidad física a la realidad aumentada y virtual.....	24
Figura 3: aplicación de realidad aumentada realizada por alumnos de la universidad Somaiya Institute of Engineering & Information [11, p. 8].....	29
Figura 4: ejemplo de aplicación de realidad aumentada hecha en unity (Riomba 2019 [3, p. 6])	30
Figura 5: resultados de la experiencia turística usando aplicaciones de realidad aumentada del caso de estudio realizado por B.I. Batuwanthudawa, K.P.N Jayasena (2020) [6, p. 7]	31
Figura 6: Aplicación WAM (World Around Me) creada por la compañía WT InfoTech [23]	34
Figura 7: Aplicación ARTour creada por la compañía Smartech Group dedicada a la creación de aplicaciones de realidad aumentada para promoción de lugares turísticos. [18].....	35
Figura 8: aplicación civilisations AR realizada por Nexus estudio creada para BBC'S dedicada para museos y galerías. [19]	37
Figura 9: último informe elaborado por la compañía PwC donde nos muestran el avance que podra tener la realidad virtual y la realidad aumentada y como mejorara la bolsa de valores en UK [20]	38
Figura 10: el modo shared AR nos permite crear experiencias compartidas en entornos de realidad aumentada, ARDK nos ofrece dos modos, high level API y low level API para este caso en particular y ya que se busca que dispositivos de gama media/baja lo soporten se prefirió el uso de una low level API	40
Figura 11: Diagrama de funcionamiento del sistema de inicio de sesión en el cual establecemos el cómo generaremos una nueva sesión cuando se cause algún error.....	41
Figura 12: diagrama de funcionamiento de mapeo de datos que nos permite entender cómo va a ir el flujo de información del host al server y del server al cliente que está ingresando a la aplicación.....	42
Figura 13: diseño de la interfaz inicial en la cual podemos observar se cuenta con un toogle log para observar	

quien está logeando, también se agregó un state control para poder observar si está conectado o tenemos falla de conexión con el servidor de Niantic.....	43
Figura 14: Algoritmo de UIMANAGER que controla todas las funciones de la app.....	44
Figura 15: Parte 1 del algoritmo de GameManager el cual es el backend principal de la aplicación en la cual se controlan y se llaman todos los scripts del mismo	46
Figura 16: parte 2 del script GameManager el cual su función es realizar el manejo de mensajes entrantes y control de acceso de los clientes que van ingresando para conocer posición por mensaje o requieren saber en qué posición se encuentra en ese momento.....	48
Figura 17: Etapa 1 NetARSessionManager en esta etapa establecemos las conexiones del ARSession y guardamos cada llamado.....	51
Figura 18: Parte 2 del script NetARSession en donde se habilitan métodos y se guardan variables en base de datos para host y cliente así como información de la sesión	53
Figura 19: Diagrama de NetMessenger el cual nos muestra todo el proceso de funcionamiento de cada etapa que se realiza al mandar un mensaje.....	56
Figura 20: Diagrama de funcionamiento de ARInteractor el cual nos muestra cómo se envían la señales de fotograma e impacto de la interacción de la sesión	58
Figura 21: diagrama del script ARObjecController en el cual se nos explica como realizara los movimientos nuestro modelo 3D al recibir un mensaje de animación.....	60
Figura 22: modelo de esquema virtual donde se comprueba el movimiento por hit de la aplicación.....	61
Figura 23: Posionamiento atrás de objetos para así generar un efecto más realista al momento de estar en el espacio de realidad aumentada.....	62
Figura 24: Dispositivo 1. Jugador/cliente en modo espera para generar un impacto o movimiento del host. Dispositivo 2. Host generando la malla en la cual podrá moverse dentro del espacio localizado y así actualizar el espacio del jugador.....	63
Figura 25: En esta prueba se realizó más allá de terreno donde no hubiera piso, para generar mallado en el pasto y cómo podemos observar ambos modelos están una posición diferente esto por la velocidad de procesamiento de la malla pero ubicados en puntos cercanos.....	64
Figura 26: Esta prueba se realizó en ambos tipos de suelo la luz que incide ya es menor pero genera el espacio en las dos zonas tanto en suelo como en el pasto esto para poder ayudar a ciertas partes de la ciudad a darles un enfoque diferente	65
Figura 27: captura desde un ángulo diferente pero justo en el pasto para notar que la malla sigue existente pero también los planos anteriores siguen activos solo generando más amplitud en el entorno.....	66

Figura 28: En esta prueba fuimos a una zona donde el terreno fuera menos uniforme para ver en qué puntos se enfocaban las mallas en este caso usamos el aparcadero de trenes para hacer esta prueba.....	67
Figura 29: Prueba en la misma zona pero usando profundidad del mismo esto para poder observar hasta qué punto podemos forzar los vectores y se logró una doble profundidad de detección además que también debemos considerar el procesamiento de los dispositivos	68
Figura 30: Dispositivo 1 fungiendo como HOST de la sesión el cual observa movimiento y posición original del líder de la sesión así como espera de los otros dispositivos	69
Figura 31:Dispositivo 2 este dispositivo usando la copia del modelo original esta se mueve en una posición diferentes pero en algunas se encuentra en la posición del original pero observado desde una posición más estable	69
Figura 32: Dispositivo 3 el cual estuvo moviendo su modelo en diversas posiciones y distancias para medir hasta que profundidad y perspectiva podíamos ver el modelo que representa al guía	70
Figura 33. Dispositivo 4 fungiendo como usuario este dispositivo al igual que el anterior se encargaron de medir profundidad además de hasta donde se generaba y el punto de alcance de la misma	70
Figura 34: Ventana principal de Unity HUB en la cual podremos crear nuestros proyectos	77
Figura 35:ventana para seleccionar módulos	78
Figura 36: sub- ventana para agregar módulos al proyecto en unity	78
Figura 37: Ventana donde podremos descargar y agregar los módulos necesarios para crear nuestro proyecto en unity.....	78
Figura 38: Ventana principal de la cuenta de Lighship ARDK para desarrollo de aplicaciones.	79
Figura 39: Ventana de Resources donde podemos descargar las herramientas necesarias para desarrollar en Unity en conjunto con Lighship ARDK.....	80
Figura 40: Apartado en donde instalaremos los packages para dev en unity	80
Figura 41: Apartado en el cual podremos crear nuestro autenticador del proyecto	81
Figura 42: Ventana donde podemos copiar nuestra clave API KEY para poder autenticar el proyecto	81
Figura 43: ventana de opciones para la ventana de ayuda de configuración.....	81
Figura 44: Ventana para colocar la API KEY de nuestro proyecto.....	82
Figura 45: Ventana de configuraciones de construcción de los proyectos	82
Figura 46: Secuencia en la cual configuremos los player settings de nuestro proyecto.	83
Figura 47: Código del gradle template generado en donde modificaremos una línea de código a la versión más reciente.....	84
Figura 48:Ventana de external preferences en donde colocaremos la ruta de nuestro archivo gradel.....	84

Figura 49: Ventana de virtual estudio que nos ayudara a simular un entorno virtual en realidad aumentada para probar nuestro proyecto	85
Figura 50: Esta referencia nos permitirá controlar todo lo relacionado al canvas desde un punto específico.	86
Figura 51: Esto es para agregarlo a las funciones completas de nuestro proyecto y controlarlo desde las escenas principales.....	87
Figura 52: Campos a referenciar en el objeto UIManager en conjunto con el script ya antes creado.....	88
Figura 53: Ventana de virtual estudio y como corroborará nuestros avances en nuestro proyecto	89
Figura 54: Se referencian los objetos para poder usarlos dentro de los scripts y así llamarlos desde otros scripts	92
Figura 55: Este archivo se referencia para poder reducir el consumo de procesamiento de los dispositivos móviles al hacer la detección de planos.....	92
Figura 56. Con esto siempre resetearé los datos de cualquier objeto que ingresemos a nuestro espacio ayudándonos a colocarlo en la posición que necesitemos.....	97
Figura 57: Necesitamos referenciar estos objetos para poder tener el control del modelo y su animación para que se vea real	101
Figura 58: Se referencia este objeto al script ya que debemos de tener el control total del objeto ya que de acuerdo a la interacción será la posición en donde deba de moverse.	101
Tabla 1: Herramientas a usar en el proyecto.....	17
Tabla 2: Especificaciones mínimas y recomendadas para desarrollar en Unity	79
Tabla 3: Script UIManager el cual se encarga de controlar el Frontend de la aplicación.....	88
Tabla 4: Inicio del Script GameManager, esta primera parte es para establecer conexión con el script NetARSessionManager para obtener los valores del mismo.....	89
Tabla 5: Script NetARSessionManager el cual se encargará de la conexión de los jugadores y control del Host	91
Tabla 6: Script ARInteractor el cual se encarga de la interacción de los jugadores y el host con el entorno ...	94
Tabla 7: Script NetMessenger el cual nos ayuda a mandar mensajes de estado de cada jugador al Host así como el Host a los jugadores para mantenerlos al tanto de cada uno de las acciones que se realizan en el entorno.....	96
Tabla 8: Parte final del Script GameManager el cual nos ayuda a tener control del Backend y Frontend de la aplicación sin acceder a uno por uno de los Scripts.....	98
Tabla 9: Scripts ARObjectController el cual se encarga del movimiento y control de las acciones que puede	

realizar el modelo, el cual tendrá un duplicado para cada uno de los jugadores.....100

CAPÍTULO 1.

1.Marco preliminar.

1.1 Introducción.

La importancia de los avances tecnológicos en los últimos años ha alcanzado niveles asombrosos. Las aplicaciones para dispositivos móviles en la actualidad son cada vez más relevantes tanto en el día a día como en el desempeño de diversas actividades, así como parte esencial del turismo, convirtiéndose en uno de los ejemplos más visibles de convergencia tecnológica. En la última década se han desarrollado aplicaciones que utilizan la visión artificial para ofrecer diversas formas de objetos o modelos 3D del mundo real a través de un dispositivo móvil. Esta tecnología, conocida como realidad aumentada o en sus siglas RA, se ha utilizado en diversas áreas de investigación como la medicina, entretenimiento, ingeniería entre otros.

Por ello, el objetivo de esta investigación es incorporar elementos tecnológicos, como la realidad aumentada mediante el uso de Unity y librerías especializadas, para establecer un sistema de mallas y vectores dentro del campo de visión de cualquier dispositivo móvil equipado con cámara y sensores integrados. Tomando como objetivo el proporcionar una experiencia realista e interactiva en un entorno virtual, permitiendo a los usuarios elegir dónde quieren ir y descubrir la magia de la realidad aumentada.

La propuesta se desarrolla dentro de la aplicación PEPE en su versión 2.0, la cual cuenta con mejoras realizadas en colaboración con el Instituto Tecnológico de Orizaba y los doctores del área de maestría en ingeniería electrónica. La versión anterior, PEPE 1.0, tenía como función principal la búsqueda de sepulcros en el panteón de la ciudad. Gracias a este proyecto, se pretende aplicar la tecnología propuesta en diversos atractivos turísticos para ayudar a preservar lugares turísticos que han sido olvidados y ofrecer un enfoque innovador al turismo en la región.

1.2 Planteamiento del problema.

La versión 1.0 de la aplicación Pepe abordó una necesidad en ciertas regiones: la búsqueda de los sepulcros de familiares difuntos y la trazabilidad de rutas para llegar a ellos. Sin embargo, tras algunas pruebas, se detectó que muchas personas buscaban información adicional, buscaban una experiencia diferente y detallada de la zona, como información de leyendas locales, personas de renombre, etc. Este hallazgo dio lugar a una problemática que surge de la necesidad de mejorar la versión 1 de la aplicación. Con esta mejora, la aplicación podría ser utilizada en el ámbito del turismo para preservar zonas turísticas olvidadas y brindar una experiencia diferente a la gente que visita a sus familias en el panteón así también brindarles una experiencia turística diferente, permitiendo abrir muchas más zonas turísticas a las cuales ya no se les toma importancia y dándole un enfoque moderno al turismo.

1.3 Justificación.

El turismo es un sector que ofrece productos y servicios que combinan diversos elementos atractivos, como paisajes naturales, patrimonio cultural e histórico, logros artísticos y tecnológicos contemporáneos, entre otros. En este sentido, no solamente se vende el atractivo en sí mismo, sino también la capacidad de este para satisfacer los deseos y expectativas de los turistas, ya que son estos atractivos los que los impulsan a elegir un destino frente a otro.

Las atracciones turísticas pueden provenir de diversos sitios, como museos, monumentos históricos, expresiones culturales y eventos programados. Sin embargo, muchas veces estas atracciones no reciben una adecuada preservación o actualización, lo que puede provocar que pierdan su atractivo y difusión dentro de los puntos turísticos del lugar. Es por eso que es importante que se busque la innovación y la incorporación de tecnología en la promoción y conservación de estas atracciones para mantener su atractivo y su relevancia para el turismo. La propuesta de mejora se enfoca en la integración de un sistema de mallado que reconocer cualquier punto de la ciudad y usando la realidad aumentada generaremos un recorrido usando un guía el cual nos llevara a conocer la zona y así conocer más sobre esta misma, especialmente en lugares que han perdido relevancia para el municipio. Además, se buscará enriquecer la experiencia del usuario. De esta manera, la aplicación PEPE no únicamente se enfocará en la búsqueda de familiares dentro de los panteones y la ubicación de sepulcros, sino que se convertirá en una herramienta útil y atractiva para el turismo en general.

La mejora en la aplicación permitirá brindar información completa y precisa a los visitantes, aumentando su satisfacción y contribuyendo a la preservación de la cultura y la historia del lugar.

Este trabajo posee una relevante utilidad metodológica, puesto que a lo largo de los años se han llevado a cabo diversas investigaciones y aplicaciones acerca del uso de la realidad aumentada en el ámbito turístico. Por tanto, es posible realizar un análisis conjunto, comparar los métodos empleados y evaluar los distintos trabajos realizados, lo que contribuirá a un mejor entendimiento y avance en este campo. Además, el desarrollo de esta investigación es viable, ya que se cuenta con los recursos y materiales necesarios para llevarlo a cabo.

1.4 Antecedentes.

La empresa Abad Médico S.A. de C.V. lanzó la aplicación PEPE en su versión 1.0, la cual cuenta con algunas funciones como la búsqueda de familiares dentro de los panteones, la ubicación y ruta para acceder al sepulcro del familiar y la identificación de puntos de interés cercanos al sepulcro.

En colaboración con el Instituto Tecnológico de Orizaba, se busca desarrollar un sistema que ofrezca un mayor atractivo turístico a lugares olvidados o que brinde información sobre personas de renombre que nacieron en esa ciudad, incluyendo cementerios como posibles lugares turísticos para preservar viejas atracciones. De esta manera, se busca no solo mejorar la aplicación PEPE en su versión 1, sino también promover el turismo en zonas turísticas olvidadas y brindar una experiencia única cuando visiten lugares que no se conocían en la ciudad. La presente investigación tiene como objetivo proporcionar una mejora significativa a la aplicación PEPE, permitiendo su aplicación en diversas zonas turísticas a través del uso de realidad aumentada. Con ello, se busca ofrecer una alternativa tecnológica para revitalizar atracciones turísticas olvidadas, brindando a los visitantes una experiencia enriquecedora e innovadora.

1.5 Hipótesis.

La implementación de un sistema de realidad aumentada basado en puntos de interés permitirá crear una alternativa tecnológica que proporcione una experiencia diferente para el usuario en comparación con una simple superposición de imágenes. Esto generará una mejora significativa en la aplicación en su versión 1 y, al mismo tiempo, ofrecerá una oportunidad única para aplicarla al sector turístico.

Para demostrar la funcionalidad del sistema, se realizarán pruebas rigurosas en un entorno controlado, utilizando diversos dispositivos móviles y situaciones de uso. Además, durante las pruebas beta en el sector turístico, se recogerán las opiniones de los usuarios para evaluar su satisfacción y la eficacia del sistema. Una vez concluidas estas pruebas, se introducirán mejoras adicionales y el sistema se lanzará en una versión final que se espera cumpla las normas de calidad y satisfacción de los usuarios. Brindará una oportunidad única para mejorar la experiencia de los usuarios y promocionar los destinos turísticos con mayor eficacia. Esto puede influir positivamente en la conservación del patrimonio cultural de la región.

1.6 Metodología.

El presente trabajo de investigación “Aplicación de realidad aumentada y puntos de interés para mejorar la experiencia turística”, por la modalidad corresponde a un proyecto de desarrollo por cuanto está encaminado a resolver problemas prácticos, a través de pruebas de campo y evaluación del proyecto en mención. Por la naturaleza es una investigación cualitativa en razón de que busca analizar el problema, mediante la generación de un prototipo que resuelva cada una de las necesidades que se han siempre buscado resolver dentro de diferentes puntos críticos y generar resultados que ayuden a favorecer estos mismos. Debido a los objetivos de la investigación, se llevará a cabo un estudio experimental que incluirá diferentes tipos de pruebas. El objetivo es establecer un punto específico en el cual las personas puedan interactuar con el entorno, convirtiéndolo en un modelo de entretenimiento y turismo con un enfoque más atractivo para los usuarios de la aplicación.

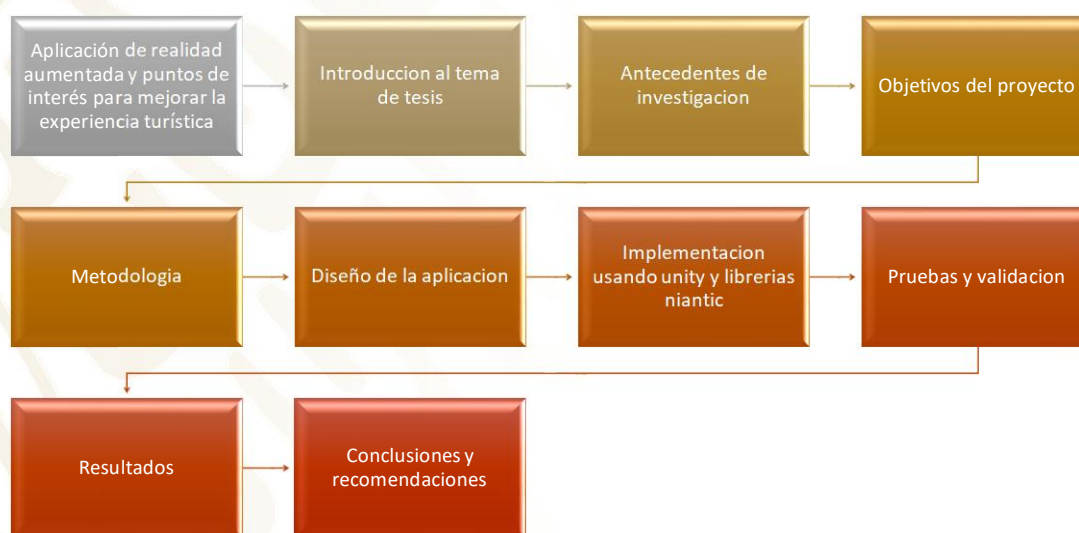


Figura 1: Diagrama de esquema de tesis

1.7 Objetivos.

1.7.1 Objetivo general

Se busca diseñar un sistema de realidad aumentada y usando un sistema de posicionamiento por vectores para mejorar la aplicación Pepe versión 1.0, con el objetivo de aplicarla en el ámbito del turismo. El sistema permitirá preservar zonas turísticas olvidadas Y de esta manera, los turistas podrán tener una experiencia más enriquecedora, didáctica y divertida.

1.7.2 Objetivo específico

1. Se propone introducir la realidad aumentada en diferentes lugares que actualmente no se les da mucha importancia, como, por ejemplo, en espacios públicos o museos.
2. Se plantea la creación de sectores adaptados para el uso de la realidad aumentada, teniendo como punto principal los panteones. Esto permitiría una experiencia diferente de solo hacer un recorrido del mismo.
3. Se sugiere el uso de tecnología smartphone en gamas bajas y medias para la adaptación con la realidad aumentada, con el fin de hacerla accesible para todos. De esta manera, se podrá ofrecer una experiencia diferente a un público más amplio.
4. Se propone llevar a cabo mejoras continuas para garantizar el soporte en las diversas plataformas móviles existentes hasta la actualidad. De esta forma, se podrá mantener la compatibilidad con las nuevas versiones de los sistemas operativos y garantizar una experiencia óptima para el usuario.

1.8 Aportes.

La contribución que se busca hacer con este trabajo de investigación es la siguiente:

- Un prototipo que mejora la aplicación Pepe para la preservación de atractivos turísticos, aplicando realidad aumentada y puntos de interés.
- El establecimiento de un sistema con diseño de interfaces que incorpora la realidad aumentada y la aplicación de un sistema de guía virtual interactivo a modo de videojuego.

Los aportes en términos de utilidad de los modelos de realidad aumentada y los puntos de interés, al usarlos para mejorar la aplicación Pepe en su versión 1, permitirán el diseño de una interfaz para el usuario. Esto nos permitirá lo siguiente:

- Mejorar la difusión de información sobre lugares turísticos olvidados.
- Reactivar la preservación de lugares que ya no son tomados en cuenta para el turismo de la zona.
- Dar apertura a la promoción de eventos culturales y ofertas diversas.

1.9 Herramientas.

Para realizar una aplicación con tecnología de realidad aumentada, se deben considerar diversas herramientas que permitan obtener resultados óptimos tanto en el uso de una computadora como en el uso de dispositivos móviles. Estas herramientas son fundamentales para crear una experiencia de usuario satisfactoria y efectiva.

Herramientas	Descripción	Uso
Unity 3D	Motor de diseño de videojuegos multiplataforma considerado uno de los mejores motores gráficos	Programa principal en la cual fue desarrollada tanto el frontend como el backend de la aplicación, así como donde mismo hare fue hecha el proceso de general los APK
Lighship ARDK	El kit de desarrollo de realidad aumentada de Lightship, conocido como ARDK, proporciona a los desarrolladores las herramientas que necesitan para crear experiencias inmersivas y multijugador, y darles vida en el mundo real.	Este kit desarrollo nos va a brindar algunas de las herramientas, librerías y algunos modelos de desarrollo que nos ayudar a realizar una APK con todo lo necesario.
Visual Estudio	entorno de desarrollo integrado (IDE, por sus siglas en inglés) creado por Microsoft que se utiliza para	Aquí fue desarrollado nuestro back end.

Tabla 1: Herramientas a usar en el proyecto.

1.10 Estado del arte.

En el artículo "THE APPLICATION OF VIRTUAL REALITY TECHNOLOGY IN THE TEACHING SYSTEM OF TOURISM TRAINING" [1] de los autores HUAN LIU, SHUFANG DING, QIAN ZENG. Describe el uso de la tecnología de realidad virtual en la formación turística y su impacto en la experiencia de aprendizaje. Los autores explican cómo la tecnología de realidad virtual puede mejorar la enseñanza al proporcionar una experiencia realista e inmersiva en un entorno turístico. La tecnología permite a los estudiantes explorar diferentes destinos de viaje de manera interactiva y práctica, lo que mejora la retención de información y la comprensión del material de estudio. El artículo presenta varios casos de tecnología de realidad virtual aplicada a la formación turística, incluyendo la creación de un entorno virtual para la enseñanza de idiomas, la simulación de escenarios de atención al cliente en hoteles y la visualización de turistas en los destinos. Además, el artículo analiza los desafíos técnicos y de diseño que surgen al usar entornos virtuales en la formación turística, como la necesidad de modelos 3D precisos y la selección correcta de hardware y software.

En el artículo "VIRTUAL TOURISM SIMULATION SYSTEM BASED ON VR TECHNOLOGY" [2] de los autores TINGFEI HOU, LIAONING JIANZHU. Nos describe un sistema de simulación de turismo virtual basado en la tecnología de realidad virtual (VR). El sistema permitirá experimentar los sitios turísticos de manera diferente en un entorno simulado, utilizando gafas de realidad virtual y controles de movimiento para interactuar con el entorno. El sistema utiliza gráficos de alta calidad para crear una experiencia inmersiva y realista para los usuarios, permitiéndoles explorar y conocer los destinos de viaje de una manera única. Además, el sistema incluye funciones interactivas, como la capacidad de interactuar con objetos y personas en el entorno. El sistema también es altamente personalizable, lo que permite a las agencias de viajes personalizar el contenido y las experiencias de acuerdo con las necesidades y preferencias de los clientes. Se espera que el sistema tenga un gran impacto en el turismo mundial, permitiendo a los usuarios experimentar destinos de viaje de formas innovadoras y emocionantes.

En el artículo el artículo "ARTOUR: REALIDAD AUMENTADA APLICADA AL TURISMO - CASO DE ESTUDIO EN RIOBAMBA, ECUADOR" [3] de los autores JUAN MALCA Y., BRAYAN CARRASCO C., VERÓNICA GUAMÁN P., JORGE DELGADO A., PAOLA G. VINUEZA N. Describe el desarrollo e implementación de ARTOUR, una aplicación de realidad aumentada para el turismo en la ciudad de Riobamba, Ecuador. La aplicación utiliza tecnología de realidad aumentada para trar información adicional sobre las

atracciones turísticas de la ciudad y brindar a los visitantes una experiencia interactiva. Los usuarios pueden descargar la aplicación en sus teléfonos inteligentes y utilizarla para explorar diferentes lugares de Riobamba, como monumentos históricos, iglesias y parques. El artículo describe en detalle el proceso de desarrollo de la aplicación, incluida la recopilación de información y datos, la creación de modelos 3D y la integración de la tecnología de realidad aumentada. También se analizan los desafíos técnicos y de diseño que surgieron durante el desarrollo y cómo se abordaron.

En el artículo “ATTRACTIONS ROAMING SYSTEM BASED ON UNITY3D” [4] de los autores JIAHUI AN, XINRONG CHENG, QING WANG AND HONG CHEN. Habló sobre la realización del sistema de roaming de lugares panorámicos basado en Unity 3D. El objetivo del sistema es permitir a los usuarios explorar virtualmente las atracciones a través de la tecnología de realidad virtual. Para implementar el sistema, se utilizaron varias herramientas y tecnologías, incluido el motor de juegos Unity 3D, así como el software de modelado 3D BLENDER. Además, se diseñaron modelos 3D detallados de atracciones para crear una experiencia inmersiva para los usuarios. El sistema de recorrido está diseñado para permitir a los usuarios explorar libremente las atracciones virtuales y acceder a información detallada sobre cada atracción. También se implementa un sistema de interacción para que los usuarios puedan interactuar con objetos y elementos dentro de la atracción virtual.

En el artículo “OBJECT VISUALIZATION USING MAPS MARKER BASED ON AUGMENTED REALITY” [5] de los autores DWI ELY KURNIAWAN, YENY ROKHAYATI, AFDHOL DZIKRI, ASEP NAJMURROKHMAN, D MUHAMMAD SURIYA. Se centra en el desarrollo de aplicaciones de realidad aumentada para visualizar objetos en entornos en tiempo real utilizando marcadores basados en mapas. La aplicación utiliza la cámara de un teléfono inteligente para escanear el entorno y detectar marcadores basados en mapas, lo que permite superponer objetos virtuales sobre el entorno real. Los objetos virtuales se crean mediante técnicas de modelado 3D y se colocan en el entorno en tiempo real mediante el sistema de seguimiento del dispositivo móvil. Este artículo describe el proceso de desarrollo de aplicaciones en detalle, incluida la creación de modelos 3D, la implementación de marcadores basados en mapas y la supervisión del entorno mediante la cámara del dispositivo móvil. Los autores también describen los desafíos técnicos que surgieron durante el desarrollo y cómo se abordaron, como la precisión del seguimiento y la estabilidad de la aplicación.

En el artículo “REAL-TIME LOCATION CASED AUGMENTED REALITY ADVERTISING PLATFORM” [6] de los autores B.I. BATUWANTHUDAWA, K.P.N JAYASENA. Este artículo se enfoca en una plataforma de publicidad en realidad aumentada basada en la ubicación en tiempo real. Es decir, permite a los anunciantes mostrar anuncios en tiempo real en función de la ubicación de los usuarios. La plataforma utiliza tecnología de realidad aumentada para superponer anuncios en el mundo real a través de la cámara de los dispositivos móviles de los usuarios. Esto permite una experiencia publicitaria más inmersiva y personalizada, ya que los anuncios pueden ser adaptados a la ubicación y contexto del usuario. Por ejemplo, un anuncio de una cafetería podría aparecer en la pantalla de un usuario mientras camina por una calle donde no hay muchas opciones de cafeterías cercanas. La plataforma también cuenta con herramientas de análisis para los anunciantes, lo que les permite medir la eficacia de sus campañas publicitarias en tiempo real y ajustarlas en consecuencia. Además, la plataforma se integra con otras aplicaciones y servicios, lo que la hace más fácil de usar y accesible para los anunciantes.

En el artículo “APPLICATION DEVELOPMENT WITH AUGMENTED REALITY TECHNIQUE USING UNITY 3D AND VUFORIA” [7] de los autores XINQI LIU, YOUNG-HO SOHN, DONG-WON PARK. Nos describe la creación de una aplicación de realidad aumentada utilizando las herramientas Unity 3D y Vuforia. El objetivo de la aplicación es mostrar información adicional sobre un objeto físico cuando se escanea con la cámara de un dispositivo móvil. El artículo discute el proceso de desarrollo de la aplicación, incluyendo la creación de modelos 3D, la implementación de la funcionalidad de realidad aumentada utilizando Vuforia, y la integración de la aplicación en un dispositivo móvil.

En el artículo “IMPLEMENTATION OF ATTRACTIONS ROAMING SYSTEM BASED ON UNITY 3D” [8] de los autores JIAHUI AN, XINRONG CHENG, QING WANG AND HONG CHEN. Nos habla sobre la implementación de un sistema de roaming de atracciones basado en Unity 3D. Este sistema permite a los visitantes de un parque de atracciones explorar y moverse libremente por el parque, mientras se les muestra información sobre las atracciones y los eventos que están sucediendo en ese momento. La implementación del sistema se basa en Unity 3D, una plataforma de desarrollo de videojuegos y realidad virtual. Se utilizaron técnicas de programación y diseño de videojuegos para crear un entorno inmersivo y atractivo para los visitantes del parque. El sistema también incluye características como la capacidad de personalizar el recorrido del usuario y la integración con dispositivos móviles para una experiencia más interactiva.

En el artículo “AUGMENTED REALITY GAME DEVELOPMENT USING UNITY3D” [9] de los autores DEEP KARIA, ABHISHEK SHAH, SHARAN VORA, PANJAB MANE, DR. BHAVESH PATEL. Nos habla del estudio que explora el uso de la plataforma Unity3D para el desarrollo de juegos de realidad aumentada. El documento comienza discutiendo las características y beneficios de la realidad aumentada y cómo se puede utilizar en los juegos para mejorar la experiencia del usuario.

El documento también incluye una sección de tutorial que guía al lector a través del proceso de crear un juego simple de realidad aumentada utilizando Unity3D. El tutorial incluye información detallada sobre cómo utilizar las herramientas de Unity3D para crear objetos 3D, definir marcadores de realidad aumentada y añadir interactividad al juego.

El artículo “USING UNITY 3D TO FACILITATE MOBILE AUGMENTED REALITY GAME DEVELOPMENT” [10] de los autores SUNG LAE KIM, HAE JUNG SUK, JEONG HWA KANG, JUN MO JUNG, TEEMU H. LAINE, JOONAS WESTLIN. El artículo se enfoca en cómo la plataforma Unity 3D puede ser utilizada para desarrollar juegos de realidad aumentada móvil de manera más fácil y efectiva. La realidad aumentada es una tecnología que permite superponer objetos digitales sobre el mundo real a través de la cámara de un dispositivo móvil. Los autores describen los beneficios de utilizar Unity 3D para desarrollar juegos de realidad aumentada móvil, como la facilidad de uso, la posibilidad de utilizar componentes preexistentes y la amplia gama de recursos disponibles en línea.

En el artículo “TOURISM APPLICATION USING AUGMENTED REALITY” [11] de los autores PRITHVI BHATT, KRUTIK PANCHAL, HARSHIL PATEL, UDAY ROTE. El artículo presenta el desarrollo de una aplicación turística utilizando tecnología de realidad aumentada. Esta aplicación permite a los turistas tener una experiencia más interactiva y enriquecedora mientras exploran destinos turísticos. La aplicación utiliza la cámara del dispositivo móvil del usuario para mostrar información en tiempo real sobre los lugares de interés cercanos y también proporciona información histórica y cultural. Además, la aplicación ofrece la posibilidad de realizar reservas de tours y actividades, así como de obtener recomendaciones personalizadas basadas en las preferencias del usuario. En general, la aplicación busca mejorar la experiencia turística al ofrecer información y servicios de una manera más accesible e interactiva.

En el artículo “TOURISM AND VIRTUAL REALITY (VR) IN DEVELOPING NATIONS” [12] de los autores ODUM CHIGOZIE JUDE Y UKEKWE CHIKE. Este artículo aborda la relación entre el turismo y la realidad virtual (VR) en países en desarrollo. Se argumenta que la VR puede ser una herramienta efectiva para promocionar destinos turísticos y mejorar la experiencia del usuario, especialmente en países donde el turismo es una fuente importante de ingresos. El artículo también examina los desafíos y oportunidades que presenta la adopción de la VR en el turismo en países en desarrollo, como la falta de infraestructura y tecnología, la necesidad de capacitación y el costo de implementación.

En el artículo “REALIDAD AUMENTADA Y TURISMO. POTENCIALES Y LÍMITES PARA LA MEJORA DE LA COMPETITIVIDAD EN LOS DESTINOS TURÍSTICOS” [13] del autor FRANCESC GONZÁLEZ REVERTÉ, investiga el creciente uso de la realidad en la industria turística y cómo puede mejorar la competitividad de los destinos. El aumento de la realidad proporciona a los turistas una experiencia más inmersiva y personalizada cuando exploran los destinos turísticos. Además, puede utilizarse para mejorar el acceso a la información y los servicios turísticos, lo que se traduce en una experiencia más satisfactoria para los visitantes.

En el artículo “Desafíos y oportunidades para el turismo en el marco de la pandemia COVID-19” de la autora VANESA CASTELLO [14]. El autor destaca los retos a los que se enfrentan los destinos turísticos, como la pérdida de puestos de trabajo, el cierre de empresas y la disminución del tráfico turístico. Sin embargo, el artículo también destaca las oportunidades para el turismo que surgen durante la pandemia. Por ejemplo, se menciona el potencial del turismo interno, que puede promoverse para impulsar las economías locales y ofrecer opciones a los turistas que no pueden viajar internacionalmente debido a las restricciones. Además, el artículo destaca la importancia de la tecnología y la innovación en el turismo en la era post-COVID. Existen soluciones digitales que pueden ayudar a los turistas a planificar y disfrutar de sus vacaciones de forma segura, como las aplicaciones de rasterización de contactos y los servicios de planificación de vacaciones.

El artículo “RESEARCH ON THE APPLICATION OF AR TECHNOLOGY BASED ON UNITY3D IN EDUCATION” [15] de los autores CHANGYUAN LI, BAIHUI TANG. Este artículo describe la aplicación de la tecnología de realidad aumentada (AR) basada en Unity3D en la educación. La tecnología AR se utiliza para crear objetos virtuales que se superponen en el mundo real, lo que proporciona una experiencia de aprendizaje interactiva e inmersiva.

CAPÍTULO 2.

2. Marco Teórico

2.1 Teoría de la realidad aumentada.

La teoría de realidad aumentada es un área de investigación interdisciplinaria que se enfoca en el estudio de cómo las tecnologías de realidad aumentada pueden ser utilizadas para mejorar la interacción entre el mundo físico y digital.

La realidad aumentada (RA) se refiere a una tecnología que permite superponer información digital en el mundo físico en tiempo real. Esta información puede ser visual, auditiva o táctil, y puede ser presentada a través de una variedad de dispositivos, incluyendo teléfonos inteligentes, tabletas, gafas inteligentes y otros dispositivos portátiles. La teoría de realidad aumentada se centra en cómo esta tecnología puede ser utilizada para mejorar la experiencia del usuario, así como en cómo las personas interactúan con la información digital en el mundo real. Algunos de los temas de investigación comunes en esta área incluyen:

- Diseño de interfaces de usuario de RA para mejorar la usabilidad y accesibilidad.
- Investigación de la percepción y cognición humana en entornos de RA.
- Desarrollo de algoritmos de seguimiento y calibración para mejorar la precisión de la superposición digital.
- Estudio de los efectos de la RA en la educación, la capacitación y otros campos de aplicación.

En general, la teoría de realidad aumentada se enfoca en comprender cómo la tecnología puede ser utilizada de manera efectiva para mejorar la interacción humano-computadora en el mundo físico, y en cómo la RA puede transformar la forma en que se interactuó con la información digital en nuestro entorno. Además, la teoría de realidad aumentada también se enfoca en la integración de la RA en diferentes industrias, como la medicina, la arquitectura, el turismo y el entretenimiento.

Por ejemplo, en la medicina, la RA puede ser utilizada para proporcionar información en tiempo real a los cirujanos durante los procedimientos, lo que puede mejorar la precisión y la eficacia de la intervención. En la arquitectura, la RA puede permitir a los diseñadores ver visualizaciones virtuales de sus diseños en el mundo real, lo que puede ayudar a mejorar la comprensión del espacio y la percepción de la escala.

Otro aspecto importante de la teoría de realidad aumentada es la comprensión de cómo la tecnología puede ser utilizada de manera ética y responsable. Como la RA puede afectar la forma en que interactúa con el mundo físico, es importante considerar las implicaciones éticas y sociales de su uso. Por ejemplo, ¿cómo se puede garantizar la privacidad y la seguridad de los usuarios en un entorno de RA? ¿Cómo se pueden evitar los sesgos y la discriminación en la presentación de información digital?

Al considerar la usabilidad, la percepción humana, la ética y la integración en diferentes industrias, la teoría de realidad aumentada puede ayudar a impulsar el desarrollo y la adopción de esta tecnología en el futuro.



Figura 2: la continua evolución de la realidad física a la realidad aumentada y virtual

Una de las características clave de la realidad aumentada es la capacidad de superponer objetos virtuales en el mundo real. Esto se logra mediante el uso de cámaras y sensores para detectar el entorno y luego colocar objetos virtuales en la ubicación correcta. Para que esto sea posible, se requiere una gran cantidad de procesamiento de datos y cálculos matemáticos en tiempo real.

Además de la superposición de objetos virtuales, la realidad aumentada también puede ser utilizada para proporcionar información contextual. Por ejemplo, un usuario podría apuntar su teléfono a un edificio histórico y recibir información sobre la historia y los eventos importantes relacionados con ese lugar.

2.2 Unity 3D y la tecnología de Realidad aumentada.

Unity y la realidad aumentada (RA) son tecnologías en constante evolución y que ofrecen un gran potencial para el sector turístico. En particular, la combinación de Unity y la RA puede proporcionar una experiencia inmersiva y enriquecedora para los turistas, lo que a su vez puede mejorar la promoción de los destinos turísticos y la satisfacción de los visitantes. En los últimos años, ha habido un creciente interés en el uso de Unity y la RA en el turismo. Los estudios y proyectos se han centrado en evaluar el impacto de estas tecnologías en la satisfacción del turista y la promoción de destinos turísticos.

Unity es un motor de juego popular que se utiliza para desarrollar experiencias interactivas en realidad virtual y aumentada. La realidad aumentada (RA) es una tecnología que combina elementos virtuales con el mundo real, mientras que la realidad virtual (RV) es una experiencia completamente virtual. Ambas tecnologías tienen el potencial de ofrecer experiencias inmersivas y personalizadas para los turistas.

El uso de Unity para desarrollar aplicaciones de RA ha sido objeto de numerosos estudios y proyectos en el ámbito del turismo. Uno de estos estudios es "ARTOUR: Realidad Aumentada aplicada al Turismo - Caso de Estudio en Riobamba, Ecuador" [3] . Este estudio presenta una plataforma de RA desarrollada con Unity para promocionar el turismo en la ciudad de Riobamba, Ecuador. La plataforma permite a los turistas explorar la ciudad de forma interactiva y obtener información sobre los lugares turísticos mediante la utilización de dispositivos móviles y gafas de RA.

Otro ejemplo de aplicación de Unity en el turismo es el "Virtual Tourism Simulation System Based on VR Technology" [2] . Este proyecto utiliza Unity para desarrollar un sistema de simulación de turismo en realidad virtual (RV). El sistema permite a los turistas explorar destinos turísticos de forma virtual y obtener información detallada sobre los lugares visitados. Además, el sistema utiliza tecnologías de inteligencia artificial para personalizar la experiencia del usuario y mejorar la satisfacción del visitante.

A pesar de estos desafíos, es importante seguir explorando el potencial de Unity y la RA en el turismo, especialmente en países en desarrollo donde el turismo es una fuente importante de ingresos. La implementación de estas tecnologías puede mejorar la promoción de destinos turísticos, aumentar la satisfacción del visitante y mejorar la sostenibilidad económica y social del sector turístico.

El uso de Unity también se ha extendido a la creación de sistemas de roaming de atracciones basados en RA. Un ejemplo de ello es el proyecto "Implementation of attractions roaming system based on Unity 3D [8]. Este proyecto utiliza Unity para desarrollar un sistema de roaming de atracciones que permite a los turistas explorar lugares turísticos de forma interactiva mediante la utilización de dispositivos móviles y tecnologías de RA. El sistema también proporciona información detallada sobre las atracciones turísticas y utiliza técnicas de gamificación para mejorar la experiencia del usuario.

En general, el uso de Unity en combinación con la RA ofrece una amplia gama de posibilidades para mejorar la promoción del turismo y la experiencia del visitante. Algunas de las ventajas de utilizar estas tecnologías incluyen:

- Mayor interacción: La RA permite a los turistas interactuar con su entorno de forma más inmersiva y enriquecedora. Esto puede aumentar la satisfacción del visitante y mejorar la percepción del destino turístico.
- Personalización de la experiencia: La utilización de tecnologías de inteligencia artificial y gamificación permite personalizar la experiencia del usuario, adaptándola a sus preferencias y necesidades.
- Mejora de la promoción: La RA puede utilizarse para crear contenido multimedia interactivo y atractivo que ayude a promocionar los destinos turísticos de forma más efectiva.
- Aumento de la accesibilidad: La RA puede hacer que la información turística sea más accesible para todos los turistas, incluyendo aquellos con discapacidades o dificultades para acceder a información impresa.

La industria turística está adoptando tecnologías avanzadas, como Unity y la RA, para mejorar la experiencia del turista y la promoción de destinos turísticos. Los estudios empíricos indican que el uso de estas tecnologías mejora significativamente la satisfacción del turista y su intención de visitar el destino turístico. La tecnología de RA ofrece una forma interactiva y personalizada de explorar destinos turísticos, mientras que la RV ofrece la posibilidad de explorar destinos de manera virtual. Los sistemas de roaming de atracciones turísticas basados en Unity ofrecen una forma interactiva y personalizada de explorar atracciones turísticas.

2.3 Unity 3D y como se puede utilizar en el turismo.

En la actualidad, el turismo es una de las industrias más importantes a nivel mundial. El desarrollo de la tecnología ha sido un gran impulso para la industria turística, permitiendo que los destinos turísticos y los prestadores de servicios turísticos tengan la capacidad de promocionarse y brindar experiencias a los turistas de una manera más efectiva. En este sentido, Unity se ha convertido en una herramienta importante en el desarrollo de experiencias turísticas interactivas y en la creación de aplicaciones móviles para turistas.

Unity es un motor de juego y una plataforma de desarrollo de aplicaciones móviles que se ha convertido en una herramienta popular en la creación de experiencias interactivas en diferentes industrias, incluyendo el turismo. Unity ofrece una interfaz gráfica fácil de usar y una amplia gama de herramientas de programación que permiten a los desarrolladores crear experiencias interactivas de alta calidad.

2.3.1 Aplicaciones de Unity en el turismo

1. Simulación de destinos turísticos

La simulación de destinos turísticos es una de las aplicaciones más populares de Unity en el turismo. Los destinos turísticos pueden ser simulados para que los turistas puedan experimentar la sensación de estar en el lugar antes de visitarlo físicamente. Los desarrolladores pueden crear simulaciones en 3D de los destinos turísticos, como las playas, montañas, monumentos históricos, y otros lugares de interés turístico, para que los turistas puedan explorarlos virtualmente antes de realizar el viaje.

2. Creación de juegos interactivos

Los juegos interactivos son una herramienta popular en la promoción de destinos turísticos y la mejora de la experiencia del turista. Los desarrolladores pueden utilizar Unity para crear juegos interactivos que involucren a los turistas en actividades específicas relacionadas con el destino turístico. Por ejemplo, se puede crear un juego interactivo que invite a los turistas a descubrir los secretos de una ciudad histórica.

3. Aplicaciones móviles para turistas

Las aplicaciones móviles para turistas son una herramienta importante para ayudar a los turistas a planificar sus viajes y obtener información sobre los destinos turísticos. Unity se utiliza para la creación de aplicaciones móviles para turistas, permitiendo que los desarrolladores creen aplicaciones que sean atractivas y fáciles de usar.

Estas aplicaciones pueden incluir información sobre lugares de interés turístico, hoteles, restaurantes, y otros servicios turísticos, así como también mapas interactivos y herramientas para la reserva de servicios turísticos.

4. Realidad Aumentada

La Realidad Aumentada (RA) es una tecnología que permite superponer elementos virtuales sobre la realidad física. En el turismo, la RA se utiliza para mejorar la experiencia del turista y crear nuevas formas de interactuar con los destinos turísticos. Unity se utiliza para crear experiencias de RA en el turismo, permitiendo que los turistas puedan experimentar una combinación de elementos virtuales y físicos. Por ejemplo, se pueden crear aplicaciones de RA que permitan a los turistas explorar una ciudad histórica con información adicional superpuesta en los edificios y monumentos históricos, o crear una experiencia de RA en un parque temático que permita a los visitantes interactuar con personajes virtuales.

2.3.2 Beneficios de utilizar Unity en el turismo

1. Mejora de la experiencia del turista

La utilización de Unity en el turismo permite mejorar la experiencia del turista. La simulación de destinos turísticos, la creación de juegos interactivos y las aplicaciones móviles para turistas, permiten que los turistas tengan una mejor comprensión de los lugares de interés turístico y que puedan planificar su viaje de manera más efectiva.

2. Mayor alcance y promoción

La utilización de Unity en el turismo también permite que los destinos turísticos tengan un mayor alcance y promoción. La creación de aplicaciones móviles para turistas y la simulación de destinos turísticos, permiten que los destinos turísticos sean promocionados de manera efectiva, llegando a un público más amplio.

3. Reducción de costos

La utilización de Unity en el turismo también puede ayudar a reducir los costos de promoción. La simulación de destinos turísticos y la creación de aplicaciones móviles para turistas, permiten que los destinos turísticos lleguen a un público más amplio, sin necesidad de gastar grandes cantidades de dinero en publicidad.

4. Innovación en la industria turística

La utilización de Unity en el turismo también permite la innovación en la industria turística. La creación de experiencias interactivas y la utilización de tecnologías como la RA, permiten que los destinos turísticos ofrezcan experiencias únicas y emocionantes para los turistas.

Unity ofrece una amplia gama de posibilidades para mejorar la experiencia del turismo. Desde la simulación de destinos turísticos, la creación de juegos interactivos y aplicaciones móviles para turistas, hasta la experiencia de RA y RV, Unity permite crear experiencias únicas y emocionantes para los turistas. Además, la tecnología de Unity permite la creación de experiencias turísticas interactivas de alta calidad de manera eficiente y efectiva, lo que puede reducir los costos de promoción y permitir que los destinos turísticos lleguen a un público más amplio.

La utilización de Unity en el turismo ofrece una amplia gama de beneficios, como la mejora de la experiencia del turista, mayor alcance y promoción, reducción de costos y la innovación en la industria turística. La creación de simulaciones de destinos turísticos, juegos interactivos, aplicaciones móviles para turistas y experiencias de RA, permiten que los destinos turísticos ofrezcan experiencias únicas y emocionantes para los turistas. Además, la tecnología de Unity permite la creación de experiencias turísticas interactivas de alta calidad de manera eficiente y efectiva.



Figura 3: aplicación de realidad aumentada realizada por alumnos de la universidad Somaiya Institute of Engineering & Information [11, p. 8]

2.4 Experiencia turística y su mejoría con las tecnologías de unity.

La experiencia turística es una parte importante de la industria turística, ya que influye en la satisfacción y el comportamiento de los turistas. La tecnología ha tenido un impacto significativo en la industria turística, permitiendo la creación de experiencias turísticas más atractivas y personalizadas. Unity, una plataforma de desarrollo de videojuegos que permite crear experiencias inmersivas en 3D, es una de las tecnologías que más se ha utilizado en la industria turística.

Unity se ha utilizado para crear diversas experiencias turísticas, como visitas virtuales a museos y lugares históricos, experiencias de realidad virtual en destinos turísticos y juegos de aventuras temáticos. Los turistas modernos esperan experiencias personalizadas y envolventes que les permitan explorar los destinos de formas nuevas y emocionantes.

Además de proporcionar una experiencia turística emocionante, la tecnología Unity tiene el potencial de mejorar la eficiencia y la rentabilidad de la industria turística. Por ejemplo, las visitas virtuales pueden reducir los costes de transporte y alojamiento, lo que puede disminuir el coste de la experiencia turística para el cliente. La tecnología Unity también puede ayudar a la industria turística a superar algunos retos actuales, como la pandemia de COVID-19. Las experiencias turísticas virtuales pueden ayudar a los turistas a explorar destinos sin poner en riesgo su salud ni la de los demás. Además, la tecnología Unity puede ayudar a la industria turística a adaptarse a las limitaciones de capacidad y distancia social. Hay que tener en cuenta algunos retos y consideraciones importantes a la hora de utilizar la tecnología Unity en el sector turístico. Por ejemplo, es fundamental garantizar que la tecnología sea accesible y esté disponible para todos los turistas, independientemente de sus capacidades o limitaciones físicas. Además, la calidad de las experiencias turísticas creadas con Unity puede variar enormemente, y es fundamental garantizar que las experiencias sean atractivas y de alta calidad para mantener la satisfacción del cliente.

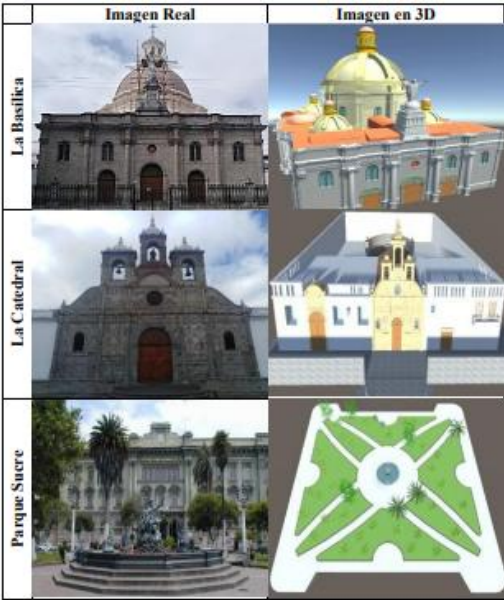


Figura 4: ejemplo de aplicación de realidad aumentada hecha en unity (Riomba 2019 [3, p. 6])

Aparte de los beneficios ya mencionados, la tecnología Unity puede ayudar al sector turístico a mejorar su sostenibilidad mediante la creación de experiencias virtuales, que pueden reducir la huella de carbono asociada a los viajes. Los turistas pueden visitar destinos sin dejar una gran huella de carbono, lo que es especialmente significativo en un mundo en el que hay que reducir las emisiones de gases de efecto invernadero.

Además, la tecnología Unity puede ayudar al sector turístico a llegar a nuevos mercados y atraer a un público más amplio. Los turistas que no pueden viajar por limitaciones físicas o económicas pueden explorar destinos a través de experiencias virtuales, lo que les permite conocer la cultura y la historia de un destino desde la comodidad de su propia casa. Para aprovechar al máximo las ventajas de la tecnología Unity en el sector turístico, es fundamental que los diseñadores de experiencias colaboren estrechamente con los proveedores de tecnología. Los desarrolladores deben conocer a fondo las herramientas de Unity para crear experiencias de alta calidad que satisfagan las expectativas de los visitantes. Además, la colaboración entre el sector turístico y los desarrolladores de tecnología puede permitir la creación de experiencias turísticas innovadoras y personalizadas que resulten únicas y atractivas. Es crucial señalar que la tecnología Unity no es la única herramienta disponible para mejorar la experiencia turística. Otras tecnologías emergentes, como la realidad mejorada y la inteligencia artificial, también pueden utilizarse para crear experiencias turísticas innovadoras y emocionales.



Figura 5: resultados de la experiencia turística usando aplicaciones de realidad aumentada del caso de estudio realizado por B.I. Batuwanthudawa, K.P.N Jayasena (2020) [6, p. 7]

2.5 Tecnología unity y realidad aumentada para la promoción de destinos turísticos.

La tecnología Unity y la realidad aumentada se han convertido en herramientas fundamentales para la promoción de destinos turísticos. La tecnología Unity es un motor de videojuegos que permite a los desarrolladores crear experiencias virtuales y de realidad aumentada, mientras que la realidad aumentada (RA) es una tecnología que utiliza un dispositivo móvil para superponer información virtual al mundo real.

En el sector turístico, la tecnología Unity y la realidad aumentada han demostrado su utilidad para promocionar destinos turísticos en todo el mundo. Estas tecnologías permiten a los turistas explorar virtualmente los destinos turísticos antes de visitarlos en persona. Además, estas herramientas permiten a los destinos turísticos ofrecer experiencias únicas a los visitantes, al tiempo que mejoran la comercialización. “La tecnología siempre ha sido un pilar de innovación para el turismo.

Desde la aparición de internet la alianza con la tecnología ha supuesto grandes transformaciones en los procesos de producción y consumo turístico que se han manifestado en varias evidencias” [13, p. 75]

El aumento de la realidad proporciona a los turistas una experiencia más inmersiva y personalizada cuando exploran los destinos turísticos. Además, puede utilizarse para mejorar el acceso a la información y los servicios turísticos, lo que se traduce en una experiencia más satisfactoria para los visitantes. Sin embargo, existen varios retos y limitaciones para un mayor uso de la realidad virtual en el turismo, como la necesidad de una infraestructura tecnológica adecuada y la dependencia de la calidad de las conexiones a Internet. También es importante destacar la importancia de una planificación adecuada y la incorporación de la tecnología a las estrategias de marketing y promoción. “¿Por qué la realidad aumentada está teniendo tanto éxito? Desde el punto de vista de los turistas, básicamente se esta ante una tecnología de bajo coste que, mediante un smartphone o una tableta, puede ser accesible a casi todo tipo de usuarios y porque aumenta de forma exponencial las posibilidades de interactuar y obtener información del entorno” [13, p. 76]

Un ejemplo de uso creciente de la realidad aumentada en la promoción turística es la aplicación "Visit Singapore AR", desarrollada por la Oficina de Turismo de Singapur. La aplicación utiliza la realidad aumentada para ofrecer a los usuarios una experiencia turística única. Los turistas pueden utilizar la aplicación para explorar diversas atracciones turísticas de Singapur, como el Jardín Botánico de la ciudad, y descubrir más información sobre ellas. [16]

Otro caso interesante es la campaña "Revealing Ancient Egypt" de la empresa de viajes Thomas Cook. La campaña utiliza la realidad mejorada para ofrecer una experiencia única a los turistas que visitan Egipto. Los turistas pueden utilizar la aplicación "Thomas Cook Explore" para escanear lugares y monumentos turísticos, como las pirámides de Guiza, y descubrir información adicional sobre ellos en tiempo real. La campaña fue todo un éxito, con un aumento del 40% en las reservas de viajes a Egipto tras el lanzamiento de la aplicación. [17]

La tecnología Unity y la realidad aumentada son herramientas fundamentales en la promoción de destinos turísticos de todo el mundo. Gracias a ellas, los turistas pueden explorar virtualmente las atracciones turísticas, y las posibilidades de crear experiencias únicas para los visitantes, lo que sin duda mejora la promoción de los atractivos turísticos.

2.6 Desafíos y limitaciones para unity y la realidad aumentada en el turismo actual.

La RA ha sido una de las tecnologías más prometedoras en el negocio turístico, y su uso en el turismo ha ido en aumento en los últimos años. La tecnología de RA permite a los turistas experimentar con la información en un entorno visual mejorado, lo que puede aumentar su interacción con el destino y mejorar su experiencia turística. Unity es un destacado motor de juegos utilizado para el desarrollo de aplicaciones de RA, y ha desempeñado un papel importante en la evolución de esta tecnología. Sin embargo, la aplicación de la RA en el turismo presenta una serie de retos y limitaciones que es preciso abordar. Uno de los retos más importantes en el uso de la RA en el turismo es la falta de contenidos relevantes y actualizados. Los turistas esperan información precisa y oportuna sobre su destino, pero producir contenidos actualizados y de alta calidad es caro y lleva mucho tiempo. Además, los contenidos generados por los usuarios pueden ser incoherentes o poco fiables, lo que repercute en la experiencia del usuario. "El segundo beneficio potencial deriva de la capacidad que la realidad aumentada tiene para mejorar la experiencia turística del visitante. La realidad aumentada ofrece información en tiempo real sobre los objetos de interés, a los que sitúa en un contexto reconocible." [13, p. 77]

Uno de los retos más importantes en el uso de la RA en el turismo es la falta de contenidos relevantes y actualizados. Los turistas esperan información precisa y oportuna sobre su destino, pero producir contenidos actualizados y de alta calidad es caro y lleva mucho tiempo.



Figura 6: Aplicación WAM (World Around Me) creada por la compañía WT InfoTech [23]

Además, los contenidos generados por los usuarios pueden ser incoherentes o poco fiables, lo que repercute en la experiencia del usuario.

Además, la implantación de la realidad aumentada en el turismo presenta retos en cuanto a la interacción con el usuario. Los turistas deben ser capaces de interactuar intuitivamente con la tecnología de RA para obtener información importante y mejorar su experiencia de viaje. La falta de conocimientos tecnológicos de algunos usuarios puede limitar su capacidad para interactuar eficazmente con la tecnología de RA, lo que puede limitar su adopción.

Por último, la falta de normas en el sector de la realidad aumentada es un reto importante para su implantación en el turismo. La falta de normas puede conducir a la fragmentación de la tecnología, lo que dificulta la adopción generalizada de la RA en el turismo y limita su capacidad para ofrecer una experiencia coherente y de alta calidad. Sin embargo, su aplicación en el turismo presenta una serie de retos y limitaciones que deben abordarse antes de que pueda adoptarse de forma generalizada. Los retos antes mencionados pueden abordarse mediante la colaboración entre la industria, el mundo académico y la Administración, así como el desarrollo de soluciones innovadoras y eficientes. “La preocupación por la saturación turística ha sido reemplazada por la preocupación alrededor de las restricciones en la circulación -las cuales no parecen vayan a modificarse en breve- y por los futuros cambios en el modo de viajar que amenazarían el devenir de las economías y sociedades de todo el mundo.” [14, p. 117]



Figura 7: Aplicación ARTour creada por la compañía Smartech Group dedicada a la creación de aplicaciones de realidad aumentada para promoción de lugares turísticos. [18]

2.7 Futuro de las tecnologías unity en el turismo usando realidad aumentada.

Unity es una plataforma de desarrollo de juegos en 2D y 3D que permite crear experiencias de realidad virtual y mejorada. Unity evoluciona junto con la tecnología, permitiendo la creación de experiencias cada vez más inmersivas y realistas. En este resumen, se observará la evolución de Unity y la realidad aumentada en el futuro, así como el efecto que estas tecnologías pueden tener en nuestras vidas. Los beneficios incluyen tanto a los turistas como a la industria. Los turistas, en su mayoría usuarios de dispositivos móviles, aumentan la interacción con el destino y obtienen información en tiempo real que les permite tener nuevas perspectivas de interpretación y comprensión de lo que visitan ([13, p. 76])

2.7.1 La evolución de unity con el turismo

Unity es una popular plataforma de desarrollo que se ha utilizado para crear una amplia gama de juegos y aplicaciones. Unity ha evolucionado junto con la tecnología, permitiendo la creación de experiencias cada vez más inmersivas y realistas. Algunas de las características más recientes de Unity son la compatibilidad con la realidad virtual y mejorada, el uso de inteligencia artificial, el aprendizaje automático y la creación de experiencias en 3D.

Se espera que Unity siga evolucionando en el futuro, permitiendo la creación de experiencias cada vez más realistas e inmersivas. Algunas de las características esperadas son:

1. Realismo mejorado: Unity seguirá mejorando su motor gráfico, lo que permitirá crear experiencias cada vez más realistas. Por ejemplo, se espera que los gráficos sean más detallados y precisos, lo que permitirá crear mundos virtuales cada vez más inmersivos.

2. Inteligencia artificial avanzada (IA): Unity también está implementando la inteligencia artificial (IA) y el aprendizaje automático (AA) para mejorar la experiencia del usuario. Se espera que estas tecnologías permitan una mayor personalización y adaptación de las experiencias virtuales al usuario, así como la creación de personajes virtuales más inteligentes y realistas.
3. Mayor accesibilidad: Con la creciente popularidad de la realidad virtual, se espera que Unity evolucione hacia una plataforma más accesible para desarrolladores y usuarios. Se espera que la creación de experiencias de RV y RA sea más fácil y accesible para quienes no tienen mucha experiencia en programación.

2.7.2 La evolución a futuro en la realidad aumentada con el turismo.

La realidad aumentada también evoluciona, con el auge de la realidad virtual y mejorada, así como los avances en la tecnología de visualización y conectividad. La forma en que se ha conectado con la realidad aumentada y su evolución para poder tener un mundo virtual al exterior. Algunas de ellas son:

1. Realidad virtual (RV): La realidad virtual (RV) se está convirtiendo en una tecnología cada vez más popular y accesible. Se espera que en el futuro la RV permita una experiencia más inmersiva y realista, con mayor resolución y mayor sensación de presencia.
2. Realidad aumentada: La Realidad Aumentada (RA) también está evolucionando, permitiendo experiencias más interactivas y personalizadas. Se espera que la RA se integre cada vez más en nuestra vida cotidiana, con aplicaciones en la industria, la educación y el entretenimiento.
3. Tecnología de visualización: La tecnología de visualización también está evolucionando, permitiendo la creación de pantallas más grandes y de mayor resolución, así como el desarrollo de dispositivos de realidad virtual más ligeros y compactos.
4. Conectividad: La tecnología actual nos permite tener una conectividad más allá de lo pensado gracias que los dispositivos móviles nos permiten alcanzar y lograr ver a nuestros familiares aun que las distancias sean muy grandes, además de ahora actualmente poder usar nuestras tarjetas de crédito por medio de una aplicación sin arriesgarnos al llevarla con nosotros, en la actualidad todas las conectividades se están volviendo más automáticas permitiéndonos acceder a ellas por comandos de voz como es el caso de Alexa.

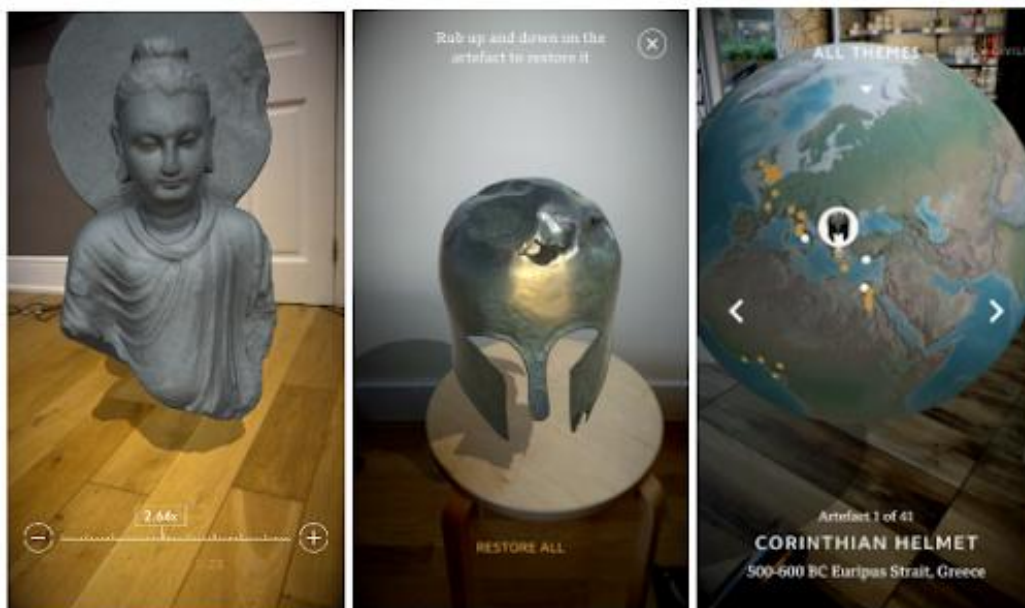


Figura 8: aplicación civilisations AR realizada por Nexus estudio creada para BBC'S dedicada para museos y galerías. [19]

2.7.3 impacto social de unity y la realidad aumentada.

La evolución de la Unidad y la Realidad tendrá una influencia significativa en la sociedad. Algunas de las formas en que estas tecnologías podrían afectar a nuestras vidas son las siguientes:

1. Entretenimiento: La realidad virtual y mejorada proporciona una experiencia de entretenimiento única, permitiendo a los usuarios sumergirse en mundos virtuales y disfrutar de experiencias envolventes. Se espera que la industria del entretenimiento maximice el uso de estas tecnologías, proporcionando experiencias de juego y entretenimiento más realistas y emocionantes.
2. Educación: La realidad virtual y mejorada también puede utilizarse en la educación, permitiendo a los estudiantes experimentar con conceptos complejos de una manera más interactiva y visible. Se espera que estas tecnologías se utilicen con más frecuencia en la educación, sobre todo en ciencia, tecnología, ingeniería y matemáticas.
3. Salud: La realidad virtual también puede utilizarse en el campo de la medicina, permitiendo a médicos y enfermeros realizar procedimientos complejos en un entorno más seguro y controlado. Se espera que estas tecnologías se utilicen con más frecuencia en la formación médica y la rehabilitación de pacientes.

4. Industria: La realidad virtual y mejorada puede tener una influencia significativa en la industria al permitir la creación de prototipos y modelos virtuales que pueden ayudar a diseñadores y fabricantes a ahorrar tiempo y dinero.

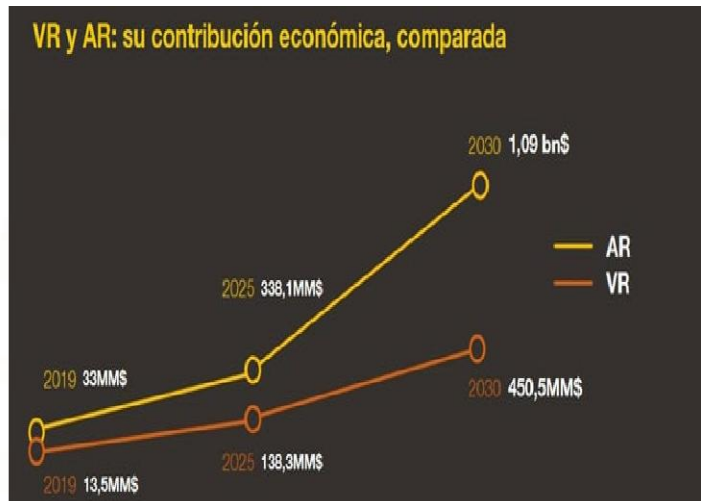
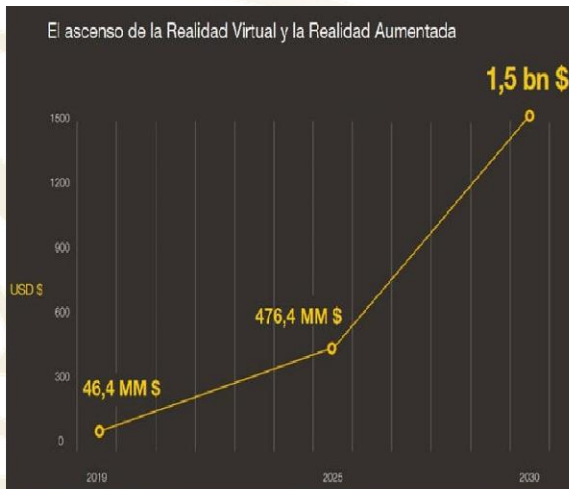


Figura 9: último informe elaborado por la compañía PwC donde nos muestran el avance que podrá tener la realidad virtual y la realidad aumentada y como mejorara la bolsa de valores en UK [20]

CAPÍTULO 3.

3. Desarrollo teórico.

Existen varias aplicaciones que se han realizado a lo largo de los años enfocadas en el turismo, cada aplicación, las cuales están diseñadas para cubrir diversas necesidades. Los modelos más comunes fueron desarrollándose tomando como base primero la realidad virtual como por ejemplo The application of virtual reality technology in the teaching system of tourism training [1] esta tecnología permite a los estudiantes explorar diferentes destinos turísticos de manera interactiva y práctica, lo que puede mejorar la retención de información y la comprensión del material de aprendizaje, basándose en entornos virtuales lo cual nos da una apertura para poder encontrar las claves para realizar un aprendizaje similar pero usando la realidad aumentada.

Otro tipo de aplicaciones que se han realizado son en modelos de realidad aumentada basados a marcadores o a relieve de objetos como es el caso de ARTOUR: Realidad Aumentada aplicada al Turismo - Caso de Estudio en Riobamba, Ecuador [3] La aplicación utiliza la tecnología de realidad aumentada para mostrar información adicional sobre los lugares turísticos de la ciudad, así como para proporcionar una experiencia interactiva a los visitantes. Esta aplicación usa un modelo 3D sobre puesto para crear un ambiente de realidad aumentada al momento que te ubiques en una zona de Riobamba. El modelo de algoritmos que usaron para esta aplicación fue unity 3D usando algunas de las librerías en conjunto con Python.

Además de estas aplicaciones, también existen modelos que usan algoritmos en Python para su desarrollo así como el uso en conjunto de la realidad virtual y la realidad aumentada, algunas de ella toman los mismos cimientos modificando el enfoque de estas mismas y usando diferentes modelos de realidad aumentada así como también realidad virtual.

3.1 Descripción del sistema.

El sistema se fundamenta en el uso de Unity en conjunto con Niantic's Lighthouse ARDK, ya que esta herramienta nos provee de herramientas necesarias para crear aplicaciones de realidad aumentada de diversas características.

La primera parte del sistema para la aplicación fue la generación fue pensar que nivel de API se quería realizar el primer análisis que se realizó fue genera una low level API para que muchas personas en simultaneo pudieran entrar a esta experiencia usando los servidores de Niantic, esto usando la función shared AR que nos permitirá que varias personas puedan estar dentro del entorno y puedan estar en tour turístico con un enfoque diferente

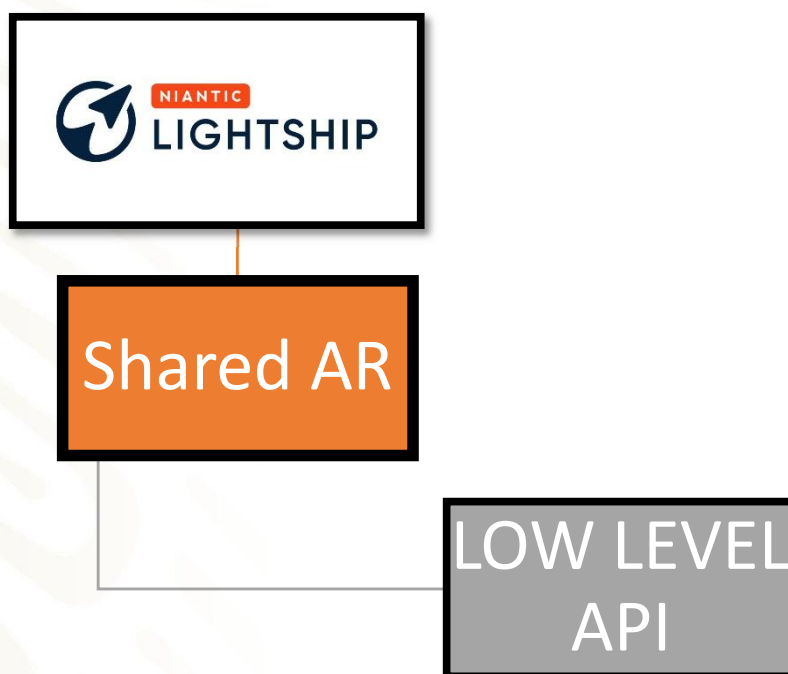


Figura 10: el modo shared AR nos permite crear experiencias compartidas en entornos de realidad aumentada, ARDK nos ofrece dos modos, high level API y low level API para este caso en particular y ya que se busca que dispositivos de gama media/baja lo soporten se prefirió el uso de una low level API

Usando la comunicación personalizada que nos permite el sistema de shared AR se va a generar un entorno en el que se podrá interactuar con un objeto basados en la posición de una malla para facilitar el posicionamiento de un punto en específico para después poder mostrar un modelo 3D en realidad aumentada.

El primer punto de partida es como fue generado que los usuarios se unan a una sala para poder estar dentro del entorno, primero debe generar un sesión ID para cada persona, este nos permitirá entrar a una sala donde el Host será el que mantenga la sesión para que las personas se unan a este espacio y disfruten de la experiencia, ahora como se explica en la figura siguiente en caso de no existir, para mantener la sesión activa el sistema habilitará una nueva sesión para generar un Host. Ahora en el caso contrario que todo funcione correctamente se continuará en la sesión creada inicialmente y la persona que entra se identificará como un Peer(cliente).

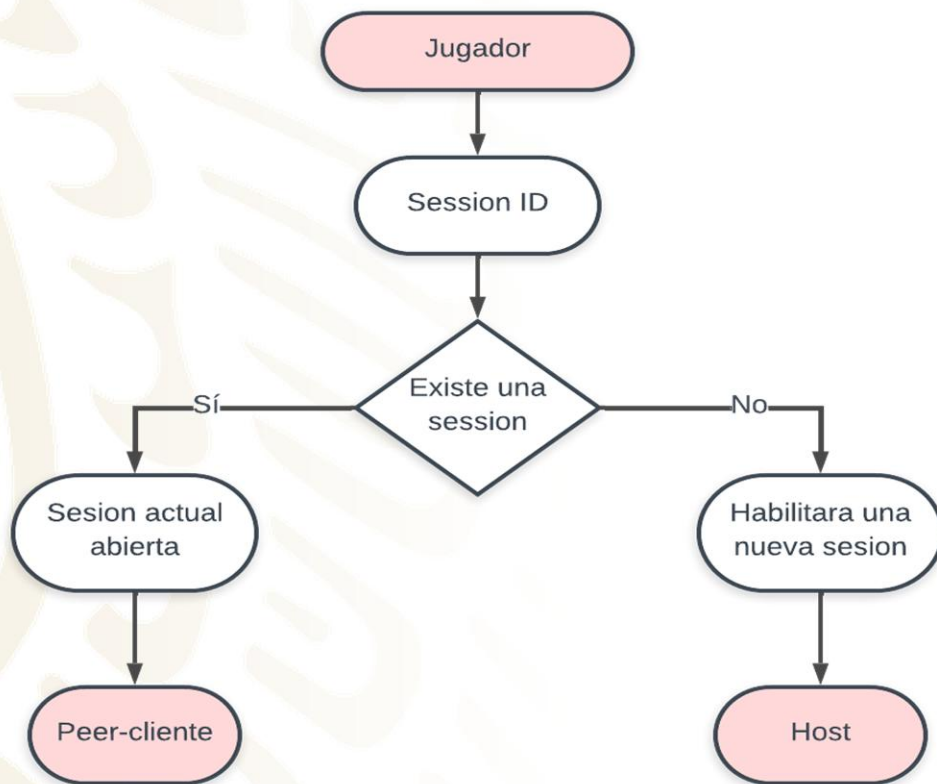


Figura 11: Diagrama de funcionamiento del sistema de inicio de sesión en el cual establecemos el cómo generaremos una nueva sesión cuando se cause algún error

Cuando el Host inicie la sesión iniciará con un estado de desconocido, esperando a que el host haga el escaneo del entorno y así envíe los datos de localización. Este escaneo se necesita para indicar que todo está estable y que se puede cargar el mapeo de datos, teniendo ya el mapeo el cliente pasará por las mismas fases de estado hasta llegar al estado de localización el cual buscará en el servidor los datos que realizó el host con anterioridad. Al ya tener estos datos pasa a estado estable, consiguiendo el servidor al detectar que ambos se encuentran estables habilita las funciones AR de la aplicación.

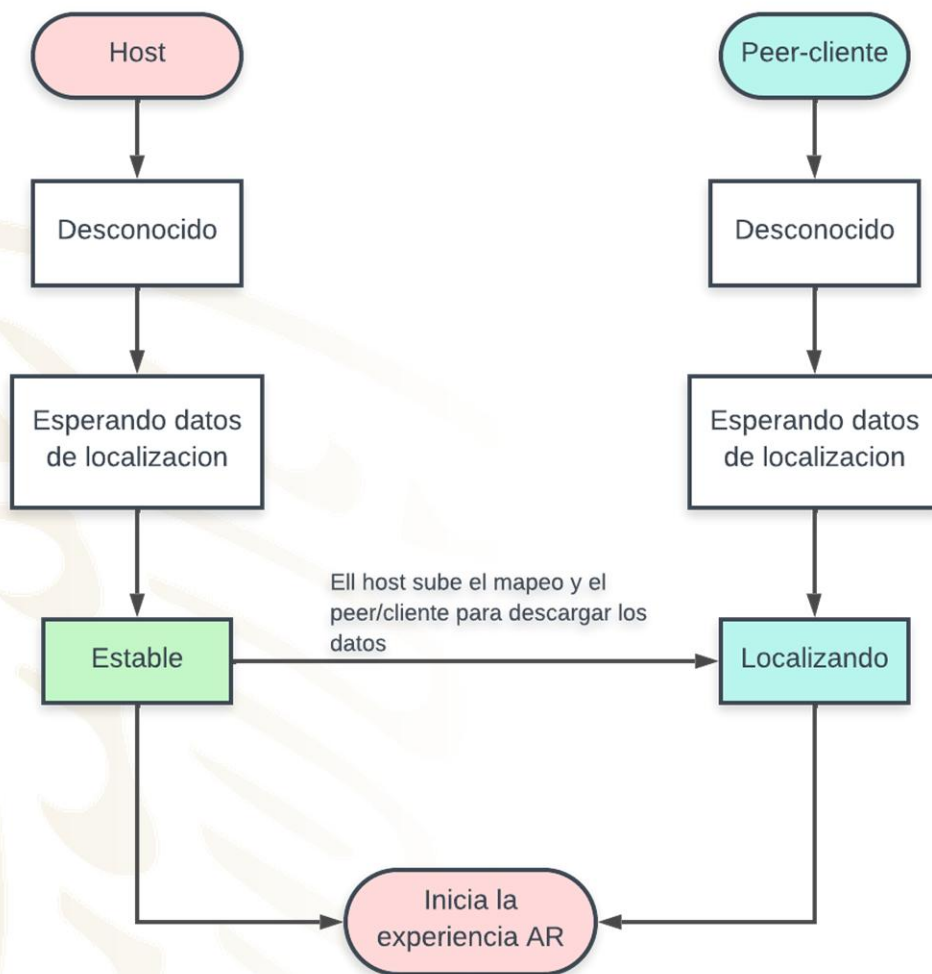


Figura 12: diagrama de funcionamiento de mapeo de datos que nos permite entender cómo va a ir el flujo de información del host al server y del server al cliente que está ingresando a la aplicación

3.2 Diseño de la interfaz.

La interfaz que se creó para esta aplicación está diseñada para que los jugadores que vayan a estar en el tour tengan un acceso único, que en este caso se les podría entregar al pagar su boleto y ahí mismo darles la clave con la que ellos podrán ingresar, esto permitiendo mantener un mayor control al estar el host iniciando el mapeo de la zona en la cual llevara a cabo el tour.

Gracias a las herramientas que Lighthouse ARDK nos brinda diversos kits que nos permiten ensamblar diversas interfaces que nos permitieron ensamblar nuestra parte visual al ser una aplicación dedicada para experiencia turística se decidió usar algo sencillo y fácil de usar, para que el cliente solo tenga que usar su código de acceso y disfrutar de la experiencia. En la figura se puede observar un poco del diseño que se usó para la interfaz.

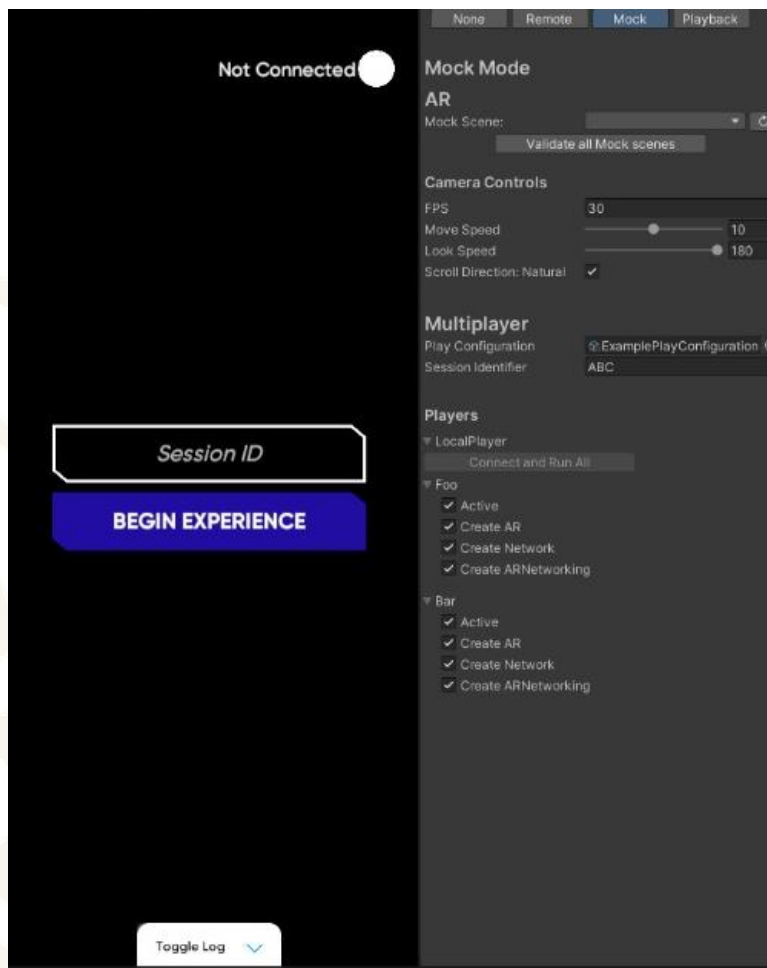


Figura 13: diseño de la interfaz inicial en la cual se puede observar se cuenta con un toggle log para observar quien está iniciando sesión, también se agregó un state control para poder observar si está conectado o tenemos falla de conexión con el servidor de Niantic

La interfaz recibirá el sesión ID otorgado que le permitirá al cliente acceder al espacio que el host estará creando, ya que este espacio este estable como se mencionó en el diagrama anterior el cliente podrá pasar a este estado y podrá observar todo lo que el host observe, esto para brindar un enfoque más interactivo.

Para la interfaz se creó un algoritmo que permita el control fluido ya que se genera una inicialización de variables base para que el cliente pueda inicializar sesión sin problema así como las interacciones en la pantalla de inicio, Dentro de la clase UIMANAGER, se declaran tres variables privadas serializadas: "JOINBUTTON", "SESSIONIDINPUT", y "SESSIONIDTEXT". Estas variables se asignan en el Editor de Unity a través de la etiqueta [SerializeField], lo que significa que se les pueden asignar valores directamente en el Inspector de Unity sin ser expuestas públicamente.

La función pública "PLAYERCONNECTED ()" es un método que puede ser llamado desde otra parte del código, muy probablemente en respuesta a un evento de conexión del jugador. Lo siguiente se hace dentro de la función "PLAYERCONNECTED ()":

1. El texto del objeto "SESSIONIDTEXT" se actualiza utilizando el texto introducido en el campo "SESSIONIDINPUT". Para ello se utiliza la función STRING.FORMAT(), que permite dar formato a una cadena añadiendo valores dinámicos.
2. El objeto "JOINBUTTON" se desactiva llamando al método "SETACTIVE (false)". Esto oculta el botón para evitar que el usuario haga clic de nuevo.
3. Del mismo modo, el objeto "SESSIONIDINPUT" se desactiva mediante el método "SETACTIVE (false)". Esto oculta el campo de entrada para evitar que el usuario vuelva a modificarlo.

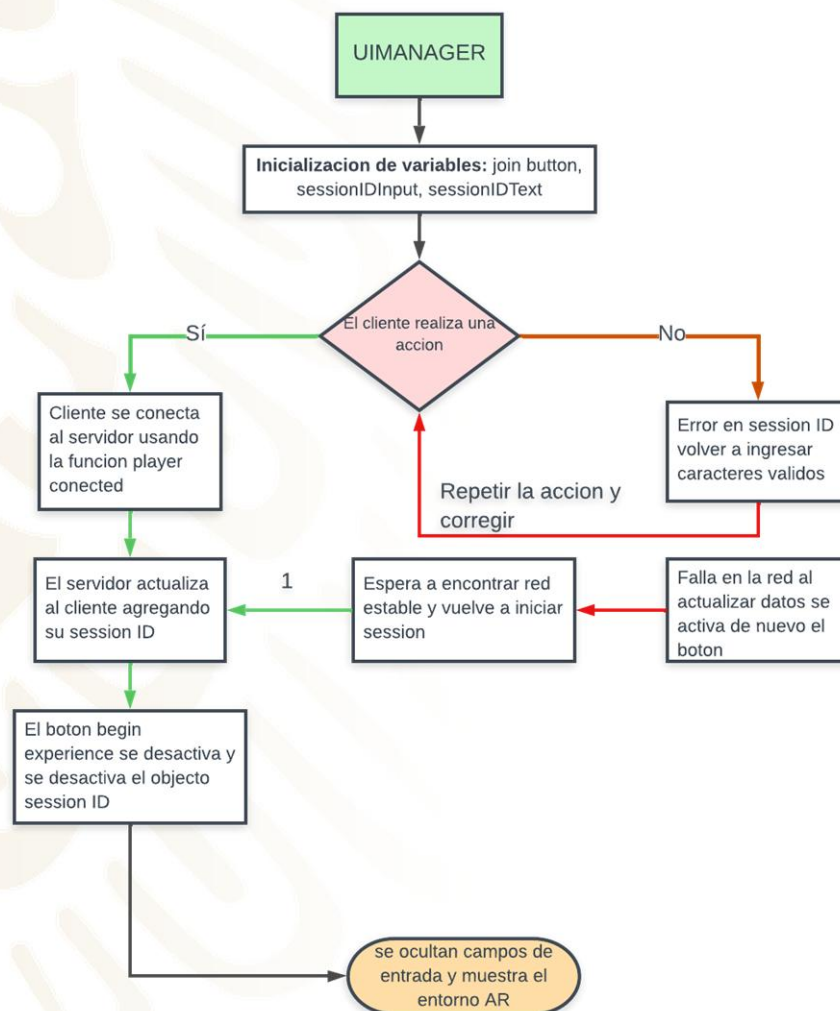


Figura 14: Algoritmo de UIMANAGER que controla todas las funciones de la app y mantener el orden entre funciones y scripts.

3.3 Diseño del administrador del juego.

Considerando algunos de los aspectos ya antes mencionados en los apéndices anteriores se diseñó un algoritmo de administración que se le dio por nombre 'GameManager'. El algoritmo nos permitirá mantener control de la aplicación de una manera más ordenada, generando un ambiente más controlado y evitando así los bugs y/o fallas que puedan llegar a pasar al generar la aplicación.

La primer parte del algoritmo tenemos que definir los componentes que se controló con GameManager para que los scripts que se definió para poder incluirlas dentro de este mismo. Definimos las variables 'aRInteractor' y una variable 'Instance', la primera se marcó como serializable ya que fue necesario asignarla desde el editor de unity, la segunda se marcó como estática ya que fue necesario llamarla desde otras partes del código para ciertos casos, sin tener la necesidad de llamar a GameManager. Después dentro nuestra clase fue generado un método 'Awake' este es un método especial de unity el cual nos ayuda a saber que solo se tuvo un GameManager activo, si este detecta otro GameManager instantáneamente destruirá este objeto. En caso contrario establecerá una instancia estática en este objeto.

En esta segunda parte de nuestro algoritmo se inició el método 'Start' el cual es un método especial de unity que siempre se llamara antes del primer Frame de actualización de la aplicación. En este método se obtuvo una referencia de un componente principal de uno de nuestros scripts del cual se habló más adelante. Dentro de esta referencia vamos a obtener un componente usando la función 'GetComponent' esta variable que se obtuvo se almacenara en la variable 'sessionManager', misma que se dio de alta en el evento 'ARNetworkingInitialized' este evento es una función especial de unity, esta nos habilitara la extensión de evento 'ARNetworkingFactory', esta misma nos proporcionara el método 'OnARNetworkingInitialized' el cual se quedara definido como el controlador de este mismo.

En el método mencionado anteriormente se ejecutó nuestra red AR. Este método recibirá un argumento de tipo 'AnyARNetworkingInitiaializedArgs', este método contendrá la información sobre nuestra red AR que estará inicializada. Consiguiente se asignó a nuestro objeto en la variable 'networking', creando una nueva instancia que lleva por nombre 'NetMessenger'. Se utilizo el objeto que se creó con anterioridad y se le asigno en una nueva variable que lleva por nombre 'messenger', además, esta se suscribirá en el evento 'PeerDataReceived'.

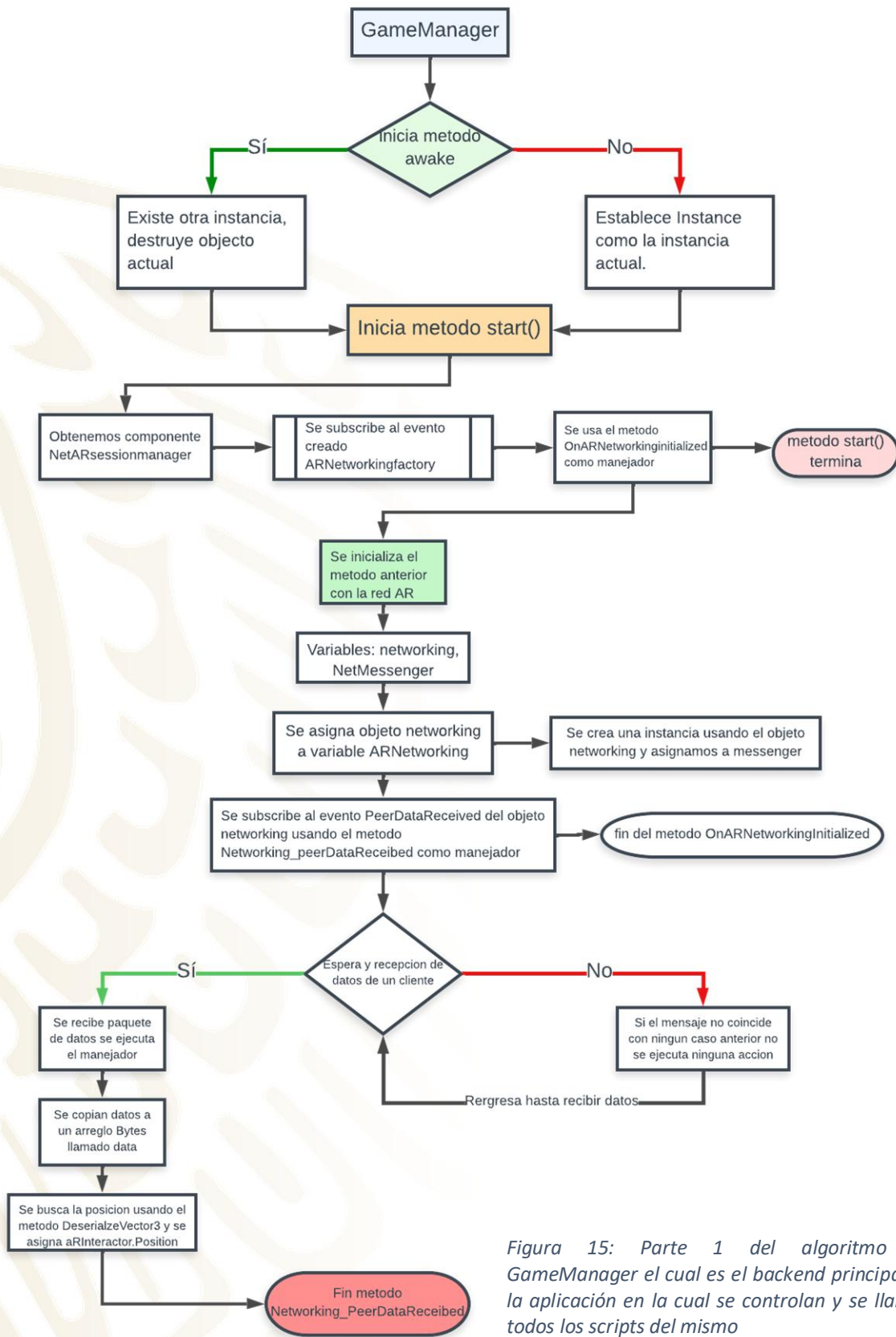


Figura 15: Parte 1 del algoritmo de GameManager el cual es el backend principal de la aplicación en la cual se controlan y se llaman todos los scripts del mismo

La segunda etapa del algoritmo de GameManager se inicializa cuando aRInteractor entra en función. Primero toma el argumento de tipo AnyARNetworkingInializedArgs que se le llamo para identificar de manera más rápida 'args'. Dentro del método se asignó el objeto networking que creo args a nuestra variable networking que se creó en la parte superior de nuestro algoritmo. Esto lo podemos realizar tomando como acceso la propiedad 'ARNetworking' de args y después tomamos la propiedad de Networking.

Al haber realizado lo anterior se creó una instancia NetMessenger usando el objeto de networking y se asignó a la variable Messenger, esta etapa se logra llamando el constructor NetMessenger y otra condición que se debe de cumplir es que se debe pasar el objeto de networking como un argumento. Cumpliendo las condiciones anteriores subscribimos al evento 'PeerDataReceibed' de nuestro objeto networking, esto lo se realizó usando el operador '+' y proporcionado el método 'Networking_PeerDataReceibed' como el manejador de este método. Este método lo que nos permitirá es inicializar la comunicación y establecer el objeto networking que es uno de los métodos principales, a su vez nos permitirá mantener un objeto de mensajería que se usará para futuras interacciones.

Inicializado lo anterior ejecutamos el método 'Networking_PeerDataReceibed' el cual recibirá el paquete de datos del cliente que este entrando en la red del Host dentro de la red ARNetworking. Primero se toma un argumento de tipo 'PeerDataReceibed' que llamaremos de igual manera 'args' como dato importante no afecta al programa el llamar dos variables de la misma manera ya que los procesos pasan de etapa a etapa permitiéndonos llamarlos de la misma manera. Dentro del método se copiaron todos los datos que recibió args pasándolos a un arreglo de bytes que se llamó 'data'. Esto se logró usando el método 'CopyData' de nuestro argumento args. Para mantener el control de nuestra estructura se usó un 'switch' esto pudo determinar qué tipo de mensaje se acabó de recibir, tomando en cuenta lo anterior nos basamos en el valor del enumerador 'TypeMessage' del objeto que se llamó 'args.Tag'. Si el tipo del mensaje entrante es 'PositionRequest' se generó una deserialización de nuestro arreglo de bytes 'data' utilizando el metodo'DeserializeVector3' del objeto Messenger, continuamente se realizó una transmisión de la posición del arreglo usando el método 'BroadcastPositio' del objeto messenger. Terminada la transmisión, la posición se asigna al atributo 'position' del objeto 'aRinteractor.ARObject.transform, esto actualiza la posición del objeto y la transmite a los clientes dentro de la red creada por el Host.

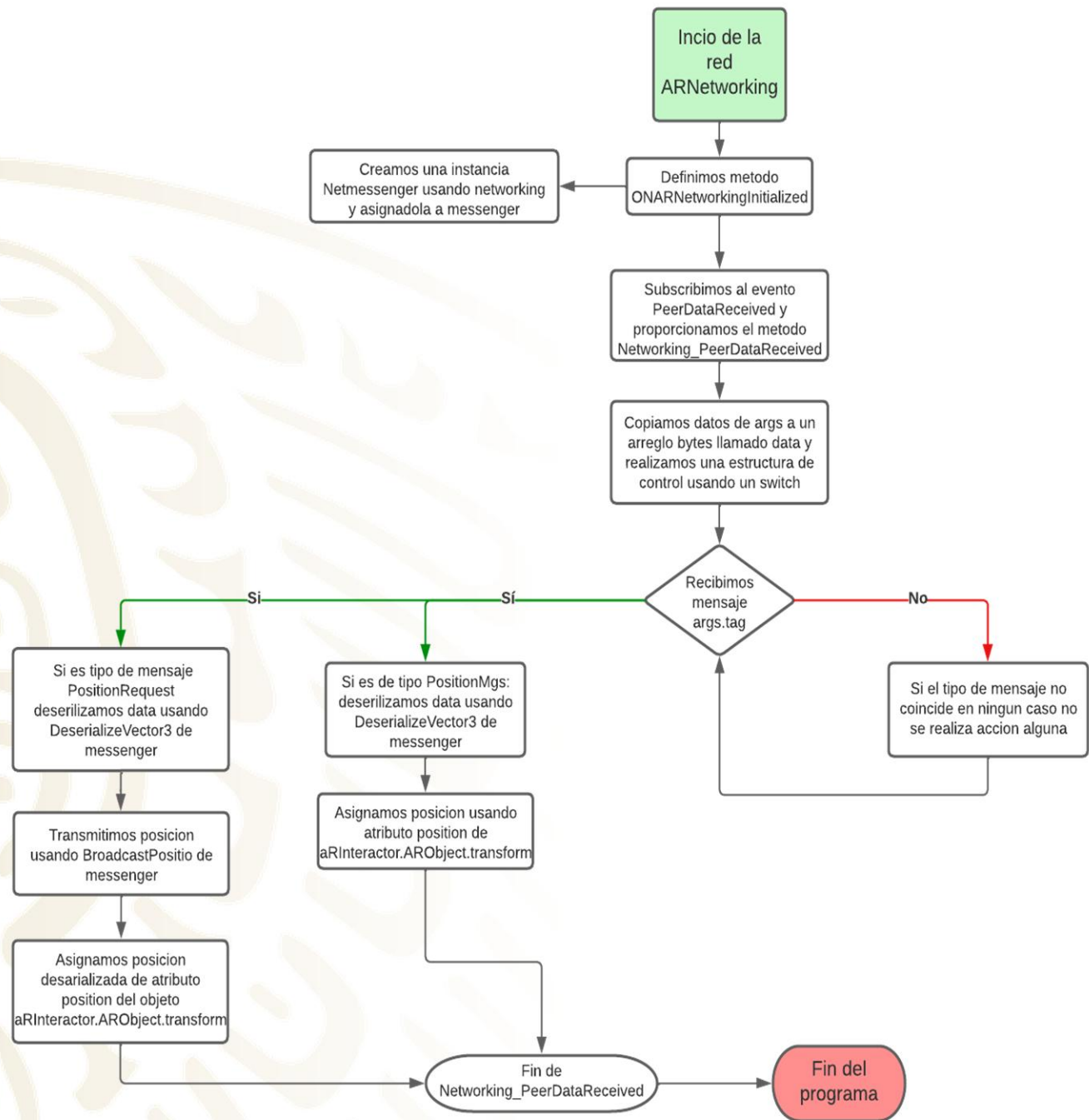


Figura 16: parte 2 del script GameManager el cual su función es realizar el manejo de mensajes entrantes y control de acceso de los clientes que van ingresando para conocer posición por mensaje o requieren saber en qué posición se encuentra en ese momento

3.4 Diseño del control de sesión entrante al espacio de realidad aumentada.

En el apéndice anterior se habló sobre el control principal de la aplicación pero se necesitó hablar de cada uno de los algoritmos usados para crear esta aplicación uno de ellos es del que hablamos a continuación. El algoritmo que lleva por nombre 'NetARSessionManager' tiene la función de gestionar la sesión de realidad aumentada y la comunicación entre pares, aunque en el algoritmo anterior gestione entradas, salidas y control completo de las funciones de la aplicación este algoritmo nos permitirá mantener el control de número de clientes que entran dentro de la aplicación.

Para la primer etapa del algoritmo se usaron variables privadas y serializadas ya que se necesita tener los accesos de cámara de los dispositivos donde se instalará la aplicación, a cada cliente se le asignó un 'playerPrefab' para poder estar dentro del ambiente AR. Se usaron estas variables privadas para almacenar información ya que se necesitó saber cuál dispositivo será el anfitrión se usó una variable con el nombre 'isHost' además de también agregar información de que la sesión actual este en un ambiente estable para identificarla se usó el nombre de variable 'isSessionStable'.

Para que el ambiente pueda iniciar se necesitó detectar que ambos tanto Host como cliente estén estables, como se menciona en la (Figura 15) si no se llega a detectar algún cliente entrante el espacio se cerrará y pedirá de nuevo el inicio de sesión.

Tomando como referencia esto ahora se accede a estas variables, para esto vamos a usar variables públicas usando un método que se usó con anterioridad 'ARNetworking'. Como primer paso debió inicializar el método y suscribirse al evento 'ARNetworkingInitialized' este método pertenece a ARNetworkingFactory este método nos proporcionará un manejador 'onARNetworkingInitialized'.

Para inicializar el método primero se debió inicializar la red, así como lo vimos en Figura 16 se debió inicializar dicho proceso. Inicializada la red se tomó el argumento 'AnyARNetworkingInitializedArgs' que llevara por nombre corto args. Dentro de dicho método que ya se usó con anterioridad se asignó el objeto de networking de args a nuestra variable 'aRNetworking'. Para esto se accedió a la propiedad de ARNetworking del método args. Posteriormente ya accedido al método, nos subscribiremos a los siguientes eventos del objeto networking:

1. **Connected:** En este primer evento se estableció la conexión con éxito entre los diversos pares. Al ya haber establecido la conexión, se proporcionó el método 'Networking_Connected' como el manejador de este evento.
2. **PeerStateReceived:** Este segundo evento entrara en función cuando reciba el estado de par, al entrar en función el evento nos proporcionara el evento 'ARNetworking_PeerStateReceived' el cual entrara en función como el manejador de este evento.
3. **PeerPoseReceived:** El tercer y último evento entrara en función cuando se recibió la pose o un par esto funcionara cuando entre un cliente en el servidor. Este último entrara y nos otorgara el método 'ARNetworking_PeerPoseReceived' como el manejador de este mismo.

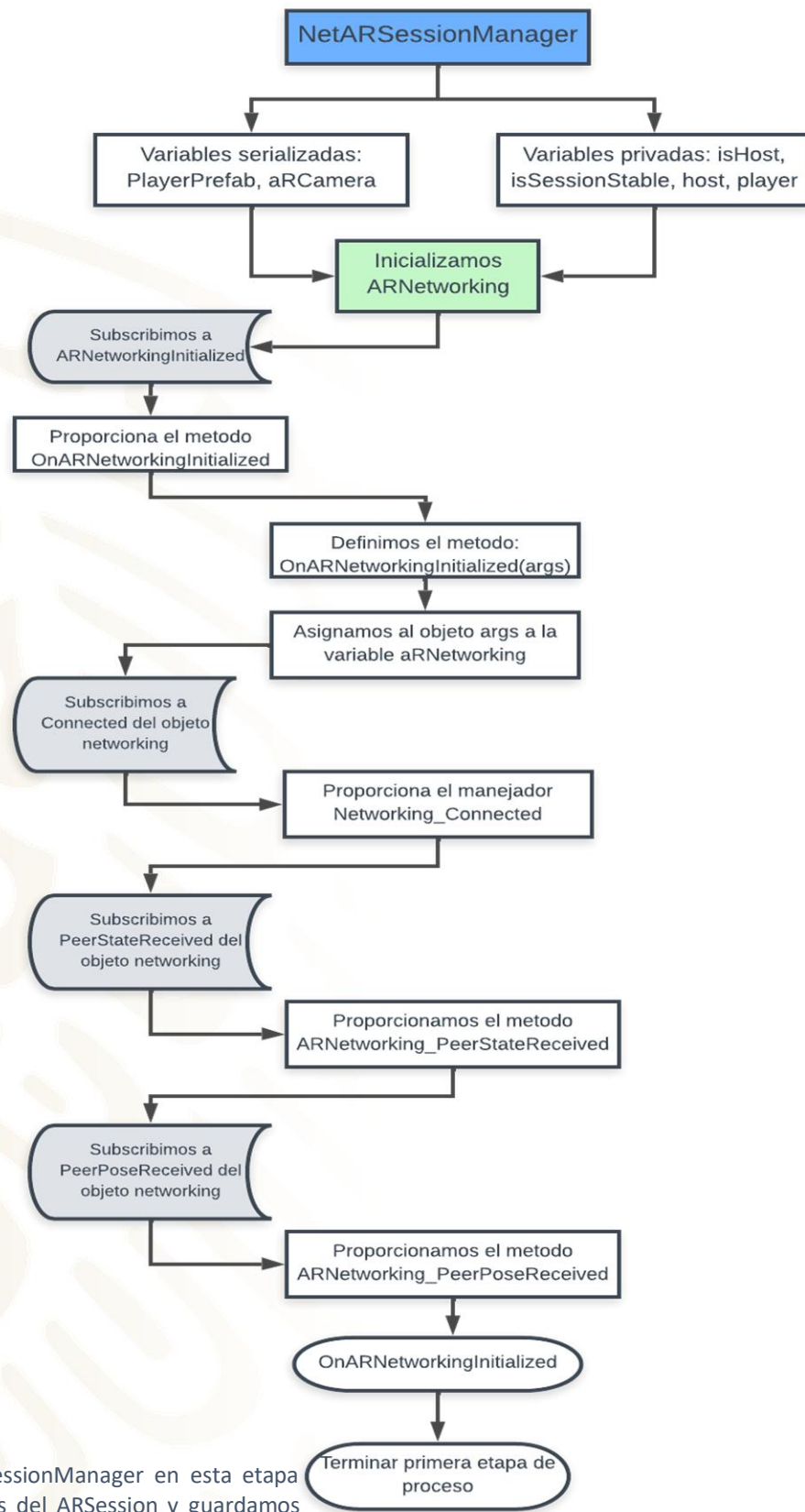


Figura 17: Etapa 1 NetARSessionManager en esta etapa establecemos las conexiones del ARSession y guardamos cada llamado

La primer etapa como se observó en la (Figura 17) es el establecimiento de permisos y accesos que nos piden las aplicaciones al abrirlas por primera vez, pero para mayor seguridad al usuario siempre se le solicitaran los permisos necesarios al ser usada dicha aplicación.

En esta segunda etapa se enfoca en la verificación de cada una de las instancias anteriores, además de verificar cada entrada por medio de diccionario, este diccionario funge como una base datos de búsqueda rápida. Primero se tomó el argumento 'PeerPoseReceibedArgs' que se llamó de forma corta args. Dentro del método se verifico usando el diccionario que lleva por nombre 'playerDictionary' no contiene la clave 'args.Peer' Y también verificar si contiene lo prefab de los jugadores que se instanciaron usando el método 'Instantiate'.

Obtenido lo anterior se usó el método 'TryGetValue' de nuestro diccionario para obtener el GameObject del cliente que corresponde a args.Peer y se debió almacenar en una variable 'player'. Se convirtio la pose recibida en un 'args.Pose' pasándolo a ser una posición 'playerPosition' utilizando el método 'ToPosition ()' Para terminar se actualiza la posición y rotación del objeto del jugador usando el método 'SetPositionAndRotation' del GameObject de nuestro cliente.

Obteniendo una conexión exitosa con la red AR entra en funcionamiento el método ARNetworking_PeerStateReceived. Este método toma un argumento de tipo 'PeerStateReceibedArgs' que se llamó de igual manera args. Dentro de este se verifico si se tiene un par con el estado 'args.State' Y está en PeerState.Stable. Si esto es correcto la variable 'isSessionStable' como verdadera(true).

Teniendo la sesión estable esto nos indica que nuestra sesión a cambiado 'Networking_Connected'. Este método al entrar en funcionamiento indicara una red estable y conectada al servidor. Al iniciar tomara como argumento de tipo 'ConnectedArgs' que al igual que en todos se llamó args. Dentro de este se asignó el valor de 'args.IsHost' a nuestra variable 'isHost' que se indicó en nuestro algoritmo anterior (Figura 17) para indicar que dispositivo es el anfitrión de la sesión. Se asigno el 'args.Host' a nuestra variable 'host' para en esta ir almacenado la información proporcionada por el anfitrión de la sesión. Después se asignó el valor de a args.Self a nuestra variable 'Player 'para almacenar la información proporcionada del dispositivo del cliente.

Para tener mayor control en la sesión se agregó el método 'OnDestroy', este método nos permitirá que si dentro del método aRNetworking el valor se vuelve nulo retornara todo y de suscribirá cada manejador de todos los eventos retornando ninguna acción adicional y generando un null.

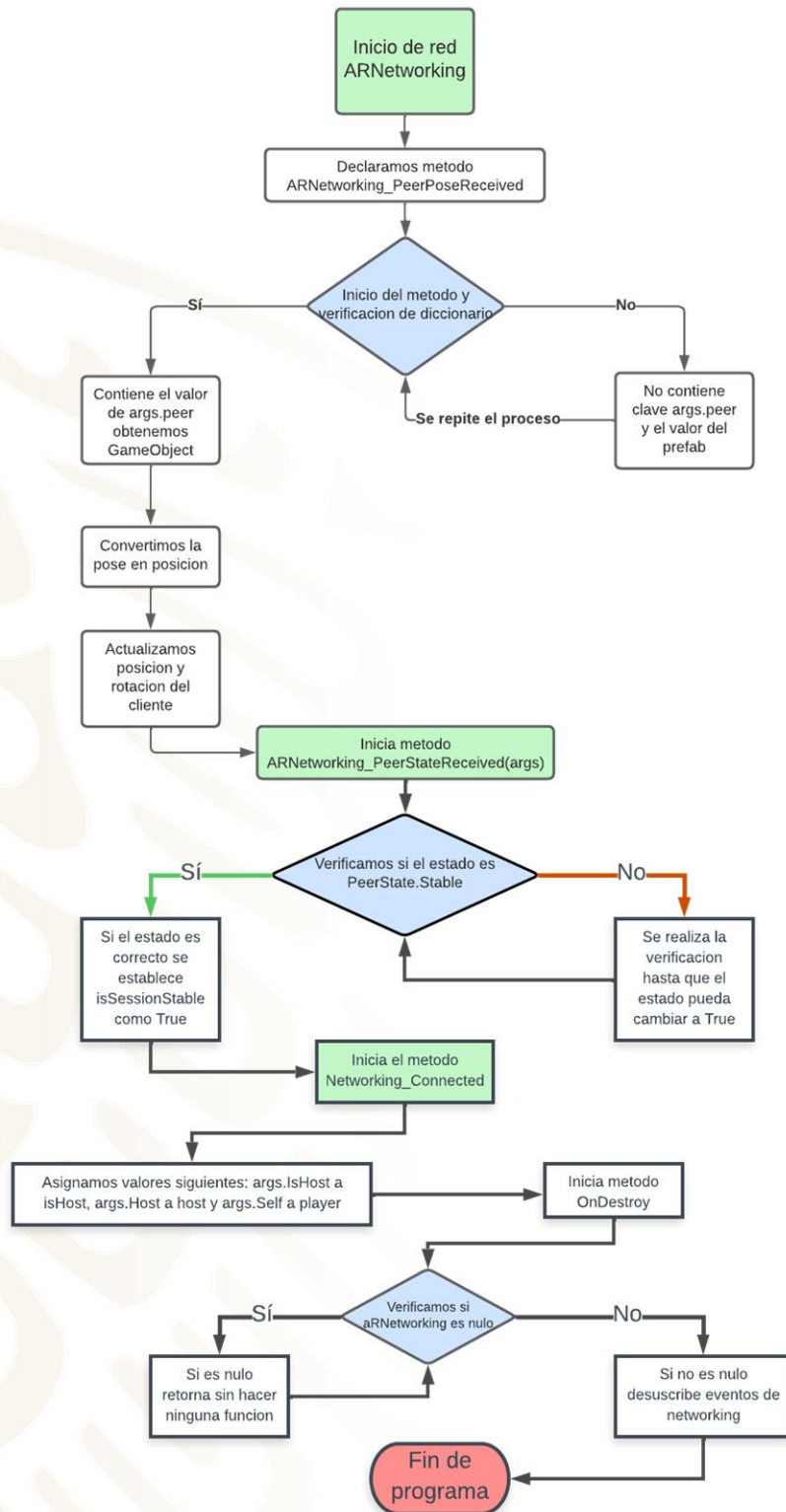


Figura 18: Parte 2 del script NetARSession en donde se habilitan métodos y se guardan variables en base de datos para host y cliente así como información de la sesión

3.5 Diseño del sistema de mensaje de posicionamiento y comunicación de la sesión AR.

En este apéndice se hablará del algoritmo encargado de la comunicación de mensajes a través de la red ARNetworking. Para iniciar este algoritmo se usó la clase 'NetMessenger', habiendo declarado esto se usó una variable 'networking' siendo esta de tipo 'IMultipeerNetworking' para que sea almacenada una referencia para nuestra red. Ahora se necesitó usar una variable que lleva por nombre 'memoryStream' esta variable tiene una capacidad de 24 Bytes que nos permitirán almacenar varios mensajes que se tengan dentro del espacio genere el host. Para poder usar estos mensajes se necesitó declarar un enumerador, para este caso se usó 'TypeMessage' ya que este contiene dos valores: 'PositionMgs' Y 'PositionRequest' cada uno de estos valores ya tienen un numero asociado como identificador.

Para poder usar la clase NetMessenger se necesitó usar un constructor. Para que los mensajes sean visibles se necesitó definirla como un constructor público que tomara como parámetro 'networking' el cual será declarado como de tipo 'IMultipeerNetworking' el constructor asignara el valor de networking a una variable de tipo privada que se declaró como 'this.networking'.

El funcionamiento de los mensajes además de mantenernos comunicados con el Host, nos permitirá conocer la posición del Host, además de poder hacer saber a los demás dentro de la sesión nuestra posición. Para esto se usó el método 'PositionRequestHost' este método se utilizó para enviarle una solicitud al anfitrión de la sesión AR abierta. Primero se tomó un parámetro 'vector3' que es del mismo tipo 'Vector3' que no representara la posición que se solicitó, utilizando el método 'SendDataToPeer' que pertenece a la variable networking para que se envíen los datos a la sesión y el anfitrión pueda aprobar la solicitud. El mensaje se identificará con un valor numérico 'TypeMessage.PosicionRequest' logrando esto serializaremos vector3 usando el método 'SerializeVector3' y el mensaje ya no se enviará como valor numérico si no como valor de dato.

Como se mencionó también se enviará la posición a todos los clientes dentro de la sesión. Para lograrlo se tuvo que realizar una secuencia similar a la anterior usando vector3 ya que este será la representación de la posición que se envió, el método que se utilizo es 'BroadcastData' que pertenece a networking esto para poder enviarlo a todos dentro de la sesión. Al igual que con al anterior se idéntica el valor como numérico y se usó la función 'TypeMessage.PositionMgs' de la misma manera se generó una serialización usando el mismo método 'SerializeVector3'.

Como último dato se debió especificar que el tipo de mensaje se transportara como 'TransportType.UnreliableUnordered'. Un punto importante a mencionar es como trabaja el método 'SerializeVector3'.

Como se ha mencionado con anterioridad este método es un arreglo de Bytes que nos permitirá mandar los mensajes dentro del entorno AR, para poder hacerlo de una manera continua se debió primero vaciar la memoria de stream esto se realizó con la función 'memoryStream', vaciada la memoria se debió crear una instancia 'BinarySerializer' que se usó con la misma función el siguiente paso es serializar vector3 usando 'Vector3Serializer' y por último se retorna el contenido a memoryStream como un arreglo de Bytes.

Como último punto se debió tener un método que nos permita deshacer algunos arreglos de bytes, para esto nos permita acceder al método 'Deserializevector3'. Este método como ya se mencionó nos permitirá deshacer los arreglos de bytes en la función vector3. Se uso MemoryStream para generar un arreglo de bytes de tipo 'data', con esto se generó la instancia 'BinaryDeserializer' utilizando una función de lectura que lleva por nombre 'readingStream'. Realizado este paso se utilizó el 'Vector3Serializer' para poder de serializar el vector3 usando la instancia que se creó con anterioridad. Terminado este proceso se procedió a devolver el valor vector3 que ya está deserializado.

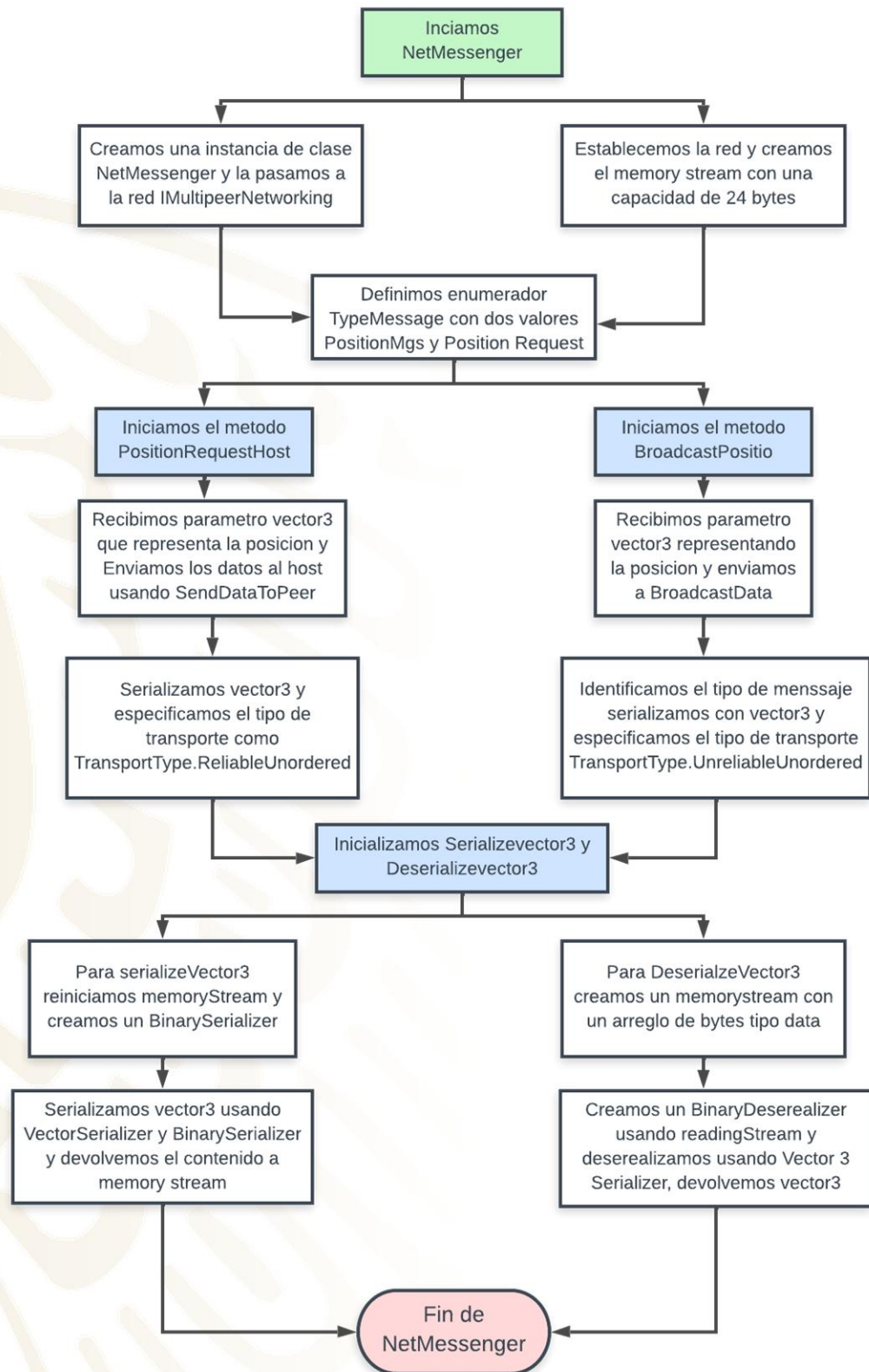


Figura 19: Diagrama de NetMessenger el cual nos muestra todo el proceso de funcionamiento de cada etapa que se realiza al mandar un mensaje

3.6 Diseño del sistema de interacción de realidad aumentada.

En los apéndices anteriores se han mencionado algunas de las funcionalidades que podemos usar dentro del entorno, pero para poder interactuar dentro del entorno se creó un script que nos permita mantener el control completo del espacio AR. Primero se usaron variables especiales que ya tenemos en otro script del cual se hablara más adelante, estas variables son dos, 'aObjectPrefab' y el controlador de objeto AR 'aObjectController' estas dos variables nos permitirán controlar el movimiento de nuestro modelo 3D.

Se inicializo el método de inicio, como primera acción nos subscribimos al evento ARNetworkingInitialized este evento se activará cuando la red AR este completa y estable, además que nos habilita el método OnARNetworkingInitialized. En el cual se asignó nuestra sesión de realidad aumentada que este activa. Al ya estar inicializada se suscribirá al evento connected, una vez conectado se debe llamar al método 'Networking_Connected', aquí debemos instanciar nuestro modelo 3D que está establecido como 'aObjectPrefab' y se asignará a nuestro controlador 'aObjectController'.

Asignado el controlador debemos habilitar el método 'Update()' el cual nos ayudara para comprobar si nuestra sesión de AR tiene una señal estable y si además en ese instante hay entradas táctiles por parte de los clientes que están en la sesión, si las condiciones se cumplen se procesara el toque mediante el método 'ProcessTouch'. Este método nos permitirá obtener los fotogramas actuales de la sesión AR y realizara una prueba de colisión establecida como 'hit test' la cual nos permitirá conocer en qué posición se realizó el toque, si el toque se encuentra en una superficie AR valida, se obtendrá la posición de impacto y se llamara con el método 'MoveARObjectCrossNetworking' para que nuestro modelo 3D se mueva.

El método mencionado anteriormente se encargará del movimiento del modelo, primero se debió detectar que dispositivo genero el impacto, si el dispositivo es el host, se estable una posición de destino directamente usando el controlador 'ARObjectController'. Si el dispositivo que envió el impacto no es el host, se debe enviar una solicitud de posición que nos proporcionara el host a través de la instancia 'GameManager.Instance.Messenger.PositionRequestHost', después enviara una señal de animación a través de una segunda instancia 'GameManager.Instance.Messenger.BroadcastAnimation'

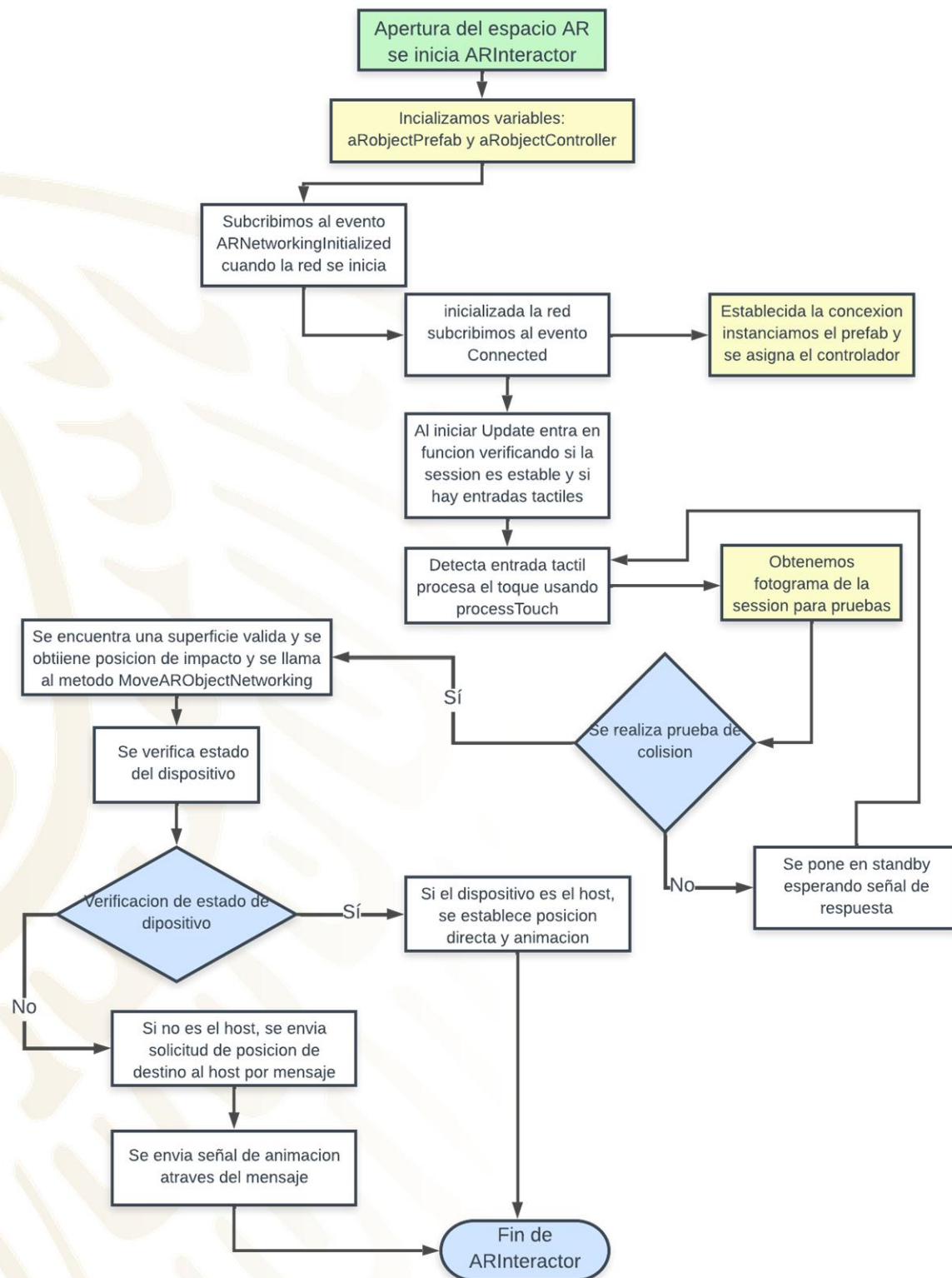


Figura 20: Diagrama de funcionamiento de ARInteractor el cual nos muestra cómo se envían la señales de fotograma e impacto de la interacción de la sesión

3.7 Diseño del sistema controlador de objeto de realidad aumentada.

En este apéndice hablará del script que nos permitirá tener el control del objeto de realidad aumentada, este script nos permitirá configurar el modelo 3D que se usó para poder configurar su movimiento y mantener control del mismo. Primero se necesitó declarar las variables privadas que son 'speed' y el 'objectanimator' estas variables las podemos ajustar desde nuestro editor de Unity, la primera variable definirá la velocidad del modelo y la segunda variable nos servirá como referencia de nuestro componente animator que es él nos permitirá observar algunos movimientos en el modelo. Además de las dos variables anteriores se usó una de las funciones de vector3 que es el 'targetposition' y una variable booleana que lleva por nombre 'isMoving'. Estas variables almacenaran una posición hacia la cual se moverá el objeto además se generó un indicador si el objeto está en movimiento o no, habilitado esto ejecutamos un update para cada frame que contenga la lógica principal del controlador del modelo dentro de este update vamos a realizar cada una de las verificaciones.

La primer verificación que se hizo, si no existe anfitrión o no detecta un host el código se detendrá y no se ejecutara en ninguna instancia, la segunda verificación que se realizo es si el modelo 3D no se encuentra en movimiento el código se detendrá y se cerrara el programa y la última verificación que se realizo es si el modelo 3D se mueve exitosamente llamaremos al método MoveARObject().

Para poder saber la posición y rotación de nuestro modelo se necesitó usar dos métodos que están vinculados a nuestro script GameManager. El primero método que se uso es 'BroadcastPositio' este método nos permitirá transmitir la posición del modelo en la zona actual, el segundo método que se uso es BroadcastEulerAngles, este método nos permitirá tener el control completo de la rotación de nuestro modelo.

Estos métodos se activarán cuando el host inicie los impactos para el movimiento o el cliente genere un impacto para solicitar movimiento en la zona AR. Para tener mayor control de la dirección del modelo se usó el método MoveARObject este calculará la dirección hacia la posición del objetivo del impacto y el modelo se desplazará hacia ese punto indicado usando las funciones Translate() y RotateTowards(), al haber cumplido exitosamente la posición objetivo se debió de llamar al método 'BroadcastAnimation' transmitiendo un mensaje de animación al GameManager mandando un estado isMoving hasta que llegue a cero para detener el movimiento del modelo 3D.

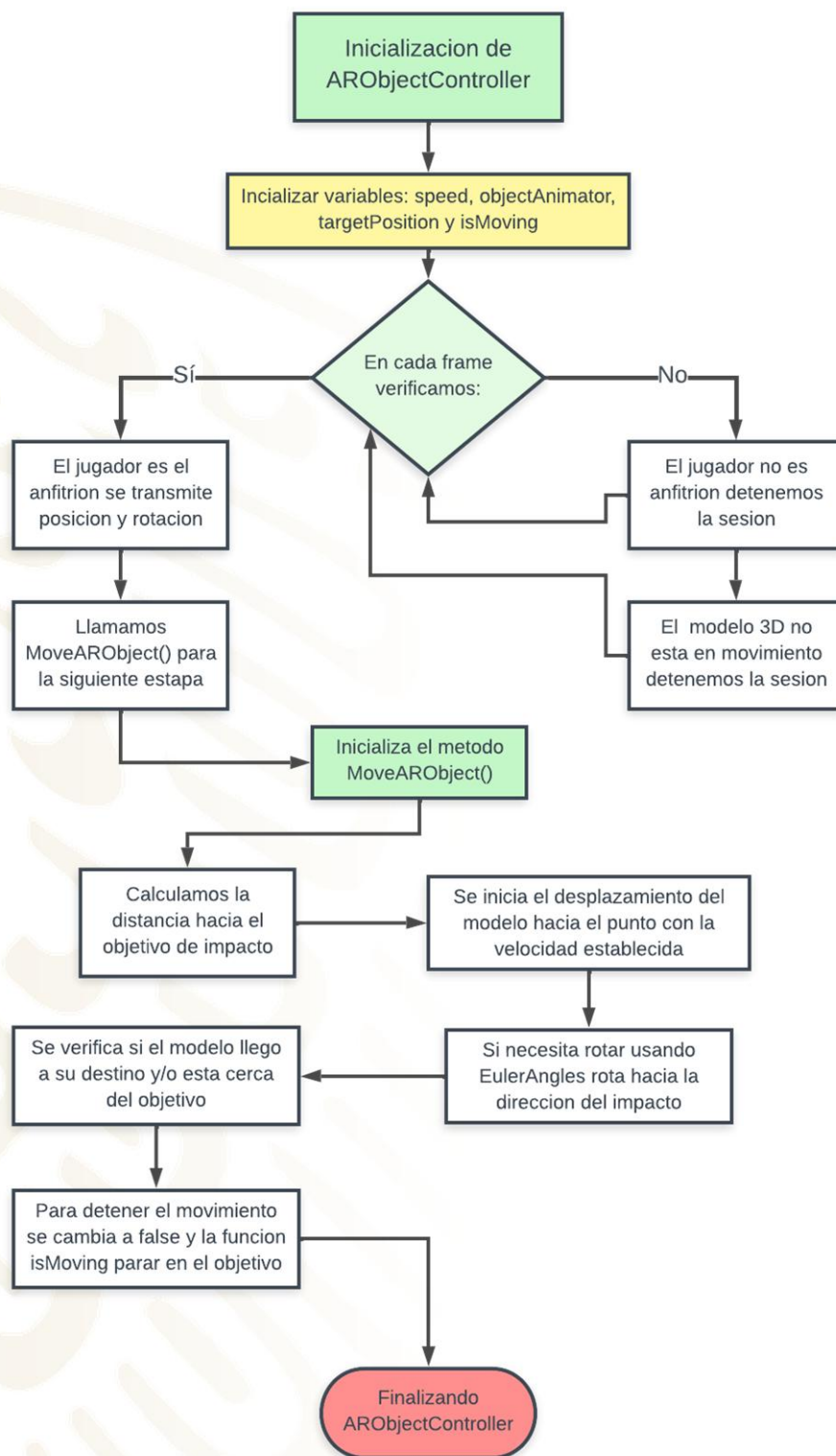


Figura 21: diagrama del script ARObjecController en el cual se nos explica como realizara los movimientos nuestro modelo 3D al recibir un mensaje de animación

CAPÍTULO 4.

4. RESULTADOS.

Los resultados presentados a continuación se hicieron en diversos espacios de la ciudad de Orizaba, Veracruz, esto con la finalidad de probar el sistema de mallas que se generó para hacer la detección y poder usarla sin depender de los diferentes tipos de luz y diferentes aspectos que puedan afectar en la detección.

Para probar primero que la aplicación y que el funcionamiento fuera el correcto, se usó el espacio virtual de unity para corroborar que cuando un host o un jugador seleccionaran una ubicación este mismo cambie de posición. Como se muestra en la figura 22, se observó que al hacer clic en un espacio el modelo 3D se coloca en la posición deseada, no importa que posición le indiques se moverá al punto indicado y como se muestra en la figura 23 moviendo la cámara del espacio virtual se pudo observar cómo está detrás del objeto.

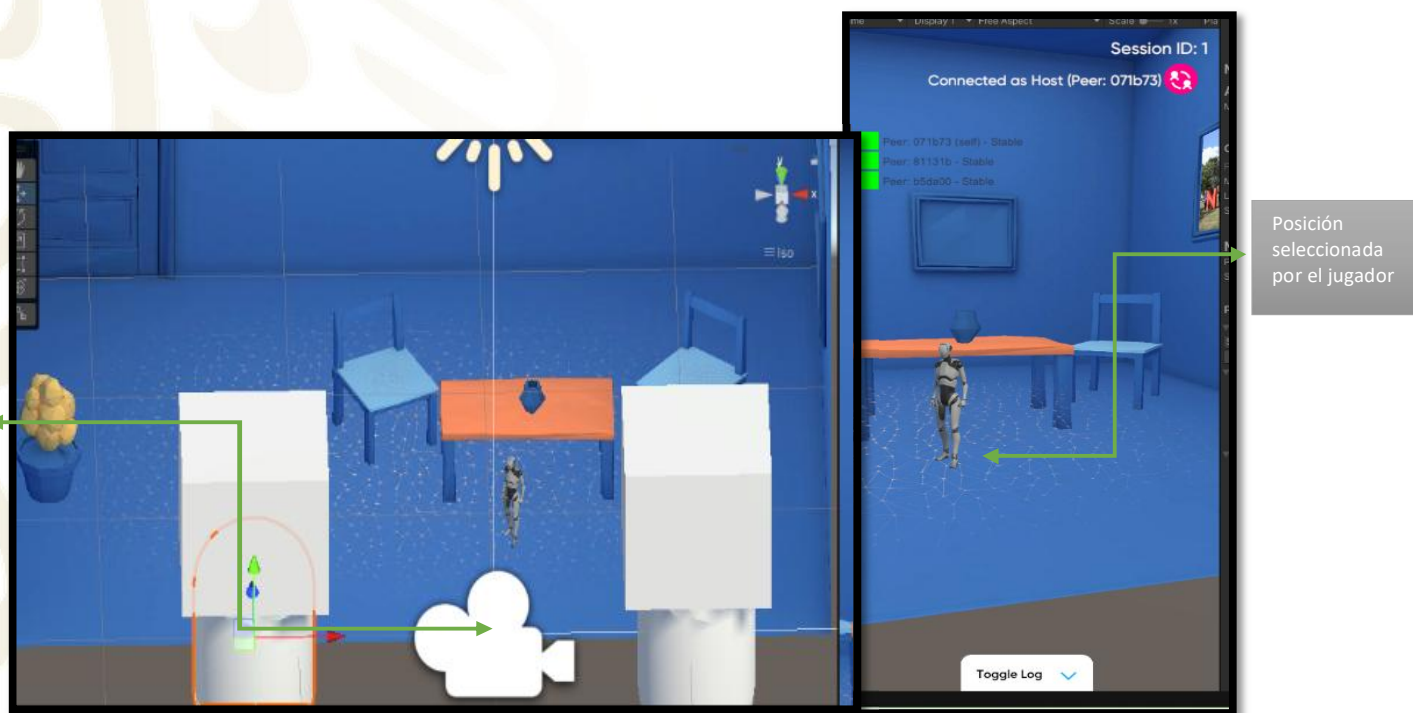


Figura 22: modelo de esquema virtual donde se comprueba el movimiento por hit de la aplicación

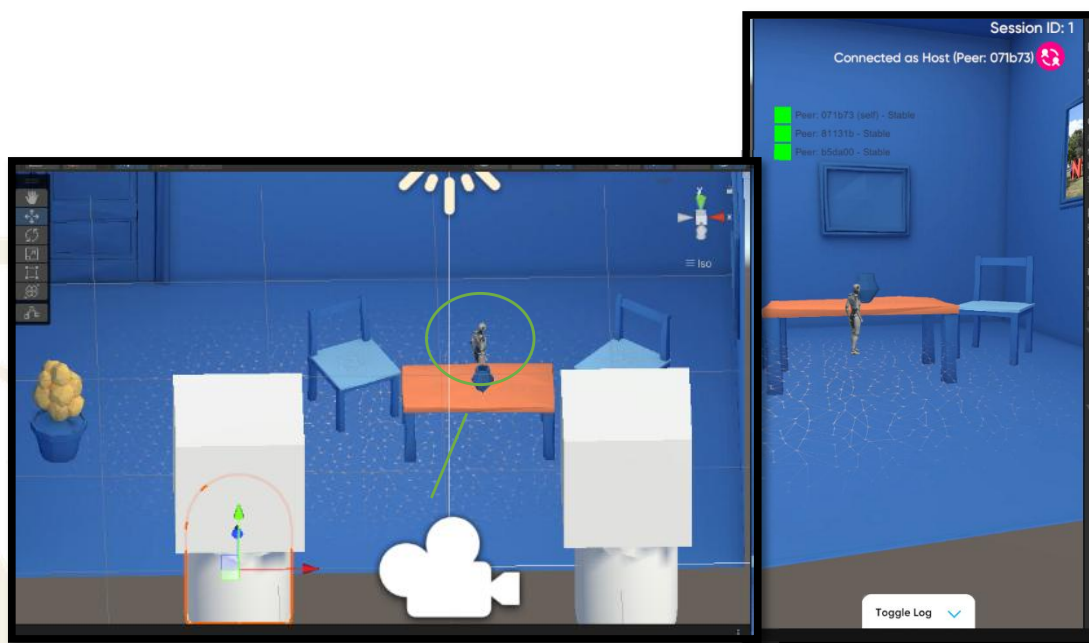
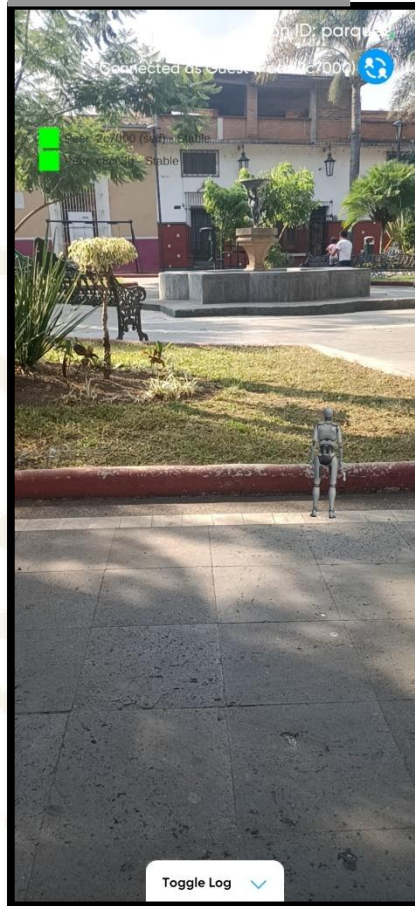


Figura 23: Posicionamiento atrás de objetos para así generar un efecto más realista al momento de estar en el espacio de realidad aumentada

Una vez corroborado el funcionamiento en el espacio de Unity se generó el APK para instalarlo en diversos dispositivos que iban desde la gama media-alta, media y media-baja. Esta aplicación fue realizada para Android 7 y superior, ya que actualmente es el Android de gama más baja.

Se realizó una primera prueba estática con una buena cantidad de luz sol para ver a qué velocidad generaba el Host el mapeo principal después que el Host a estableciendo la zona donde se realizara la actividad, empieza a generar el mapeo para jugador/cliente esto para que también pueda participar y decidir posicionamiento del modelo. Otro punto importante a observar en la Figura 24, las posiciones de ambas pueden variar por un poco dependiendo el procesamiento de los dispositivos móviles así como de los Frames por segundo que procese el mismo, como se pudo observar en el dispositivo 2 va generando la malla conforme la sesión va avanzando buscando espacios posibles el color del suelo no afecta en lo más mínimo y la luz incide en ese momento tampoco afecta a la generación de la malla ayudando que pueda generarla sin problema para poder seguir generando más espacios con el mallado.

Dispositivo 2



Dispositivo 1

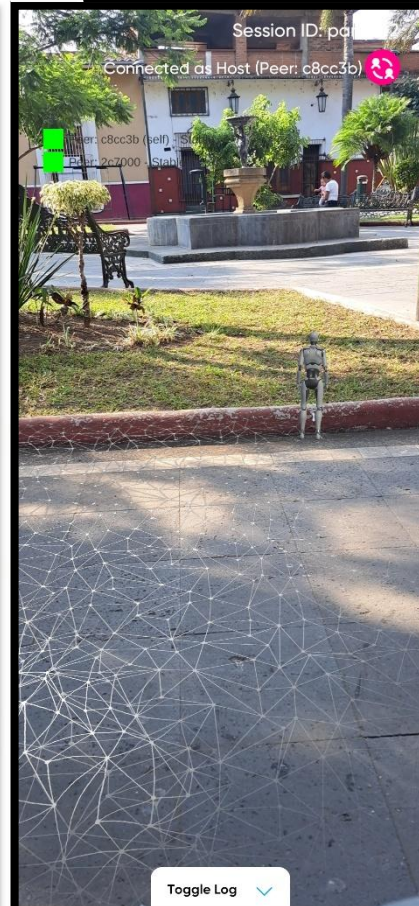
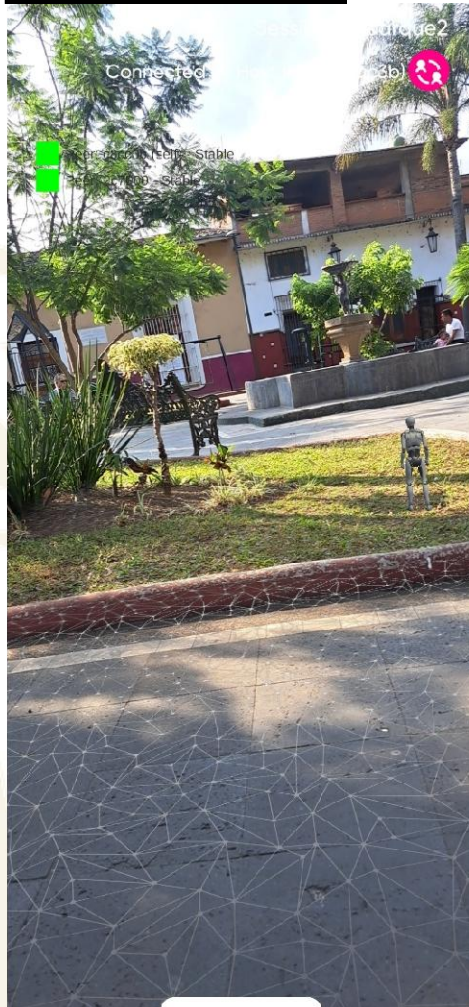


Figura 24: Dispositivo 1. Jugador/cliente en modo espera para generar un impacto o movimiento del host. Dispositivo 2. Host generando la malla en la cual podrá moverse dentro del espacio localizado y así actualizar el espacio del jugador

Para la segunda prueba misma que se realizó en el parque, pero generando el mallado de vectores en el pasto esto para corroborar que se genere esta malla en espacio irregular como lo es una área verde. Como se pudo observar en la Figura 25 la luz de sol no nos impide la generación de la malla, ya que la malla está diseñada para adaptarse a los entornos, además que también esto dependerá del dispositivo móvil que se use como HOST, este dispositivo al ser el líder de la sesión es el encargado de ir estableciendo los puntos donde se generara la malla los dos dispositivos que se muestran en las pruebas son dispositivos de gama media-alta, pero uno tiene un procesador con mayor velocidad que el otro en este caso es al que se designó como host de la sesión ayudando así a generar de una manera más ágil los espacios de realidad aumentada que se muestran y así permitiendo que el jugador/clientes pueden observar, jugar y disfrutar de la experiencia.

Dispositivo 1



Dispositivo 2



Figura 25: En esta prueba se realizó más allá de terreno donde no hubiera piso, para generar mallado en el pasto y cómo podemos observar ambos modelos están una posición diferente esto por la velocidad de procesamiento de la malla pero ubicados en puntos cercanos

Para el tercer prueba se realizó cerca de un pequeño monumento que se encuentra en el parque de San Juan de dios ubicado en la parte sur de la ciudad de Orizaba, Veracruz. En esta prueba la luz de sol ya no era tan fuerte, pero se logró que la malla se viera reflejada en diversos ángulos primero se hizo de frente como se pudo observar en la Figura 26, se colocaron los dispositivos en puntos iguales con alturas de enfoque un poco diferentes para que se observara que aun apuntando a puntos opuestos la malla nos permite siempre mantener nuestro punto fijo sin perder en que zona está establecida, pero siempre generando más malla al movernos nosotros, si observa la distancia en la que se ve el objeto varía en distancia y posición, pero ambas tienen generada la malla, ya que el HOST está actualizando constantemente el entorno.



Figura 26: Esta prueba se realizó en ambos tipos de suelo la luz que incide ya es menor pero genera el espacio en las dos zonas tanto en suelo como en el pasto esto para poder ayudar a ciertas partes de la ciudad a darles un enfoque diferente

Para el segundo ángulo se colocó el dispositivo 1 a una posición un poco más alta para poder observar hasta qué punto está generada la malla mientras que el otro dispositivo se alejó un poco para poder observar la segunda malla generada y cómo se observa en la Figura 27 el HOST tiene generada la malla completa y el segundo dispositivo está actualizando la posición para generar la malla por si quiere mover el modelo a otro punto esto para mantener el control entre el HOST y el jugador/cliente. El segundo modelo es un duplicado, pero aun que la malla se vea inexistente está presente dentro del plano.

Dispositivo 1



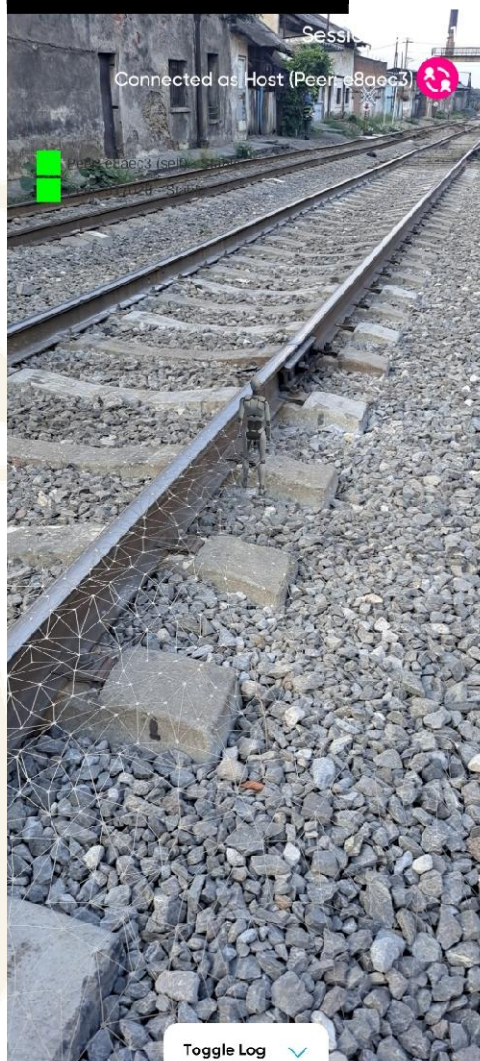
Dispositivo 2



Figura 27: captura desde un ángulo diferente pero justo en el pasto para notar que la malla sigue existente pero también los planos anteriores siguen activos solo generando más amplitud en el entorno

En la cuarta prueba se usó un mapeo más irregular como lo es las vías del tren esto para poder ver hasta qué punto nos generaba el mapeo como se observa en la Figura 28 el mapeo buscaba partes planas, pero gracias a la vectorización también lo genero en partes irregulares como lo es la zona donde están las piedras y también la zona donde se establecen las líneas para fijar las vías del tren esto ayudando que se pudo generar mapeos en zonas que no son puntos planos sino también con zonas sin un punto para fijarla pero logrando la generación del mallado de vectores.

Dispositivo 1



Dispositivo 2



Figura 28: En esta prueba fuimos a una zona donde el terreno fuera menos uniforme para ver en qué puntos se enfocaban las mallas en este caso usamos el aparcadero de trenes para hacer esta prueba

En la misma zona se realizó otra prueba, pero probando ahora la distancia y la profundidad en la cual se pudo trabajar con el mallado como observa en la Figura 29, el dispositivo 1 se encuentra cerca de nuestra posición, pero en el dispositivo 2 la distancia de posición del modelo se encuentra más centrada y ubicada en un punto de la malla esto se debe a que los procesamientos se están sincronizando, pero también el jugador/cliente puede decidir en qué posición ubicar su propio modelo sin depender del Host. Algo muy importante que tomar en cuenta es que en cuanto el HOST haga una actualización de la posición aun sin tener el otro dispositivo malla en esa posición actualizara ese punto generando un espacio vacío y pudiendo moverlo sin necesidad de un punto existente.

Dispositivo 1



Dispositivo 2

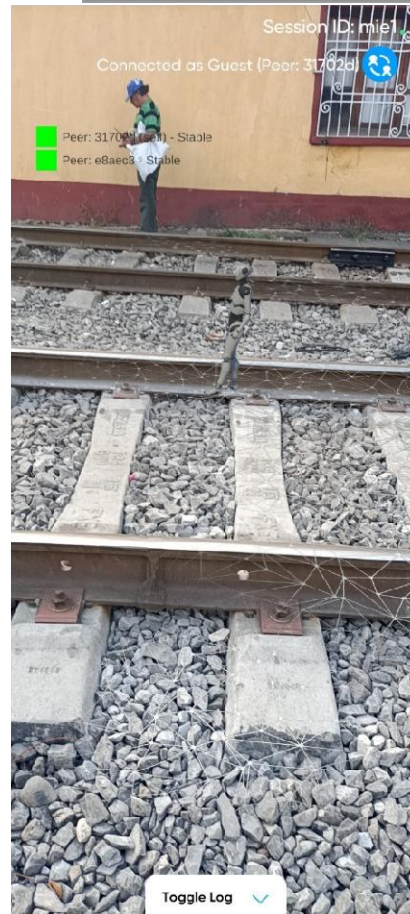


Figura 29: Prueba en la misma zona pero usando profundidad del mismo esto para poder observar hasta qué punto podemos forzar los vectores y se logró una doble profundidad de detección además que también debemos considerar el procesamiento de los dispositivos

Para la última prueba de la app se realizó un experimento en una zona con más interferencias para probar los alcances y fallas de la app. El río Orizaba es parte turística si no es que de las más importantes de la ciudad gracias a la cantidad de especies en peligro que están resguardadas por el municipio por ende el probar una aplicación que permita llevar un guía en este tipo de atracciones generaría un auge mucho mayor. Para esta prueba se usaron 4 dispositivos en simultáneo que nos permitieran ver que alcance se pudo lograr además de probar la velocidad de procesamiento de análisis del espacio y generación de la malla vectorizada para la ubicación del posicionamiento de cada dispositivo, como se observa en las imágenes la distancia de cada Modelo 3D variaba del dispositivo Host, para algunos casos tuvimos problemas con la ubicación GPS de otros dispositivos, el procesamiento era lento, pero se logró el objetivo de conectarse a la sesión y mover el modelo sin problemas y ubicarlo en posiciones diferentes para observar la profundidad y distancia a la que estaban del modelo líder que en estos casos es el HOST.

Dispositivo 1

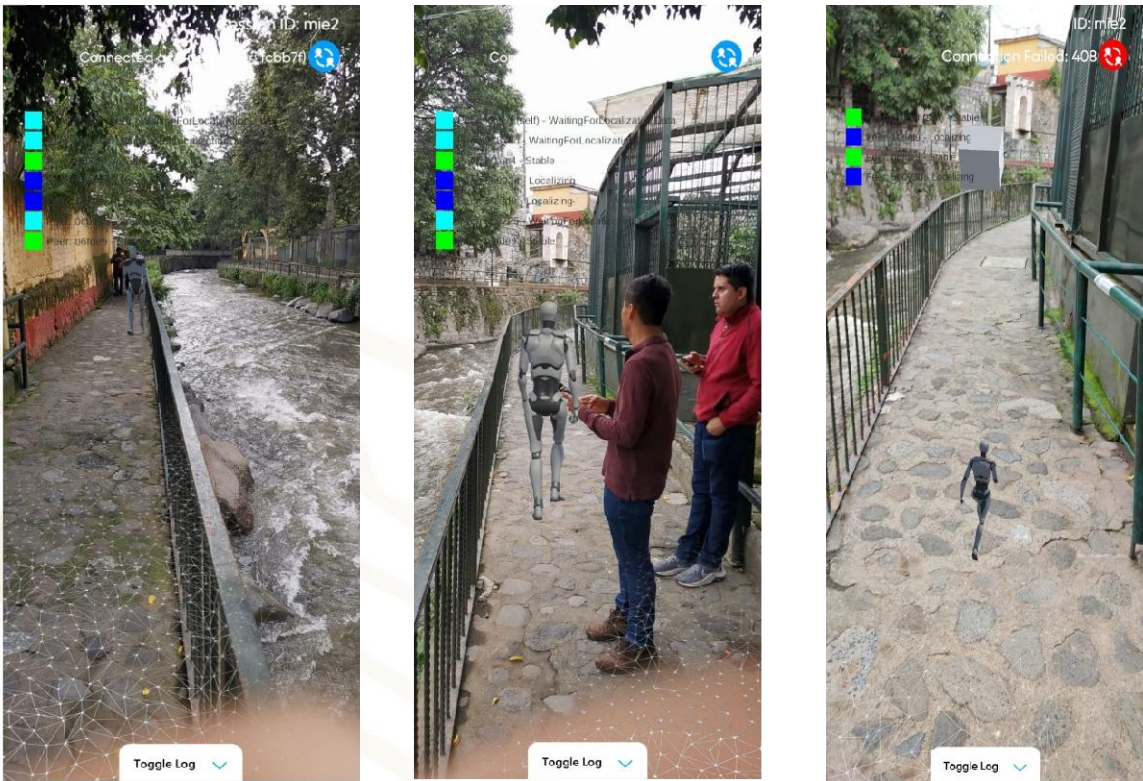


Figura 30: Dispositivo 1 funcionando como HOST de la sesión el cual observa movimiento y posición original del líder de la sesión así como espera de los otros dispositivos

Dispositivo 2

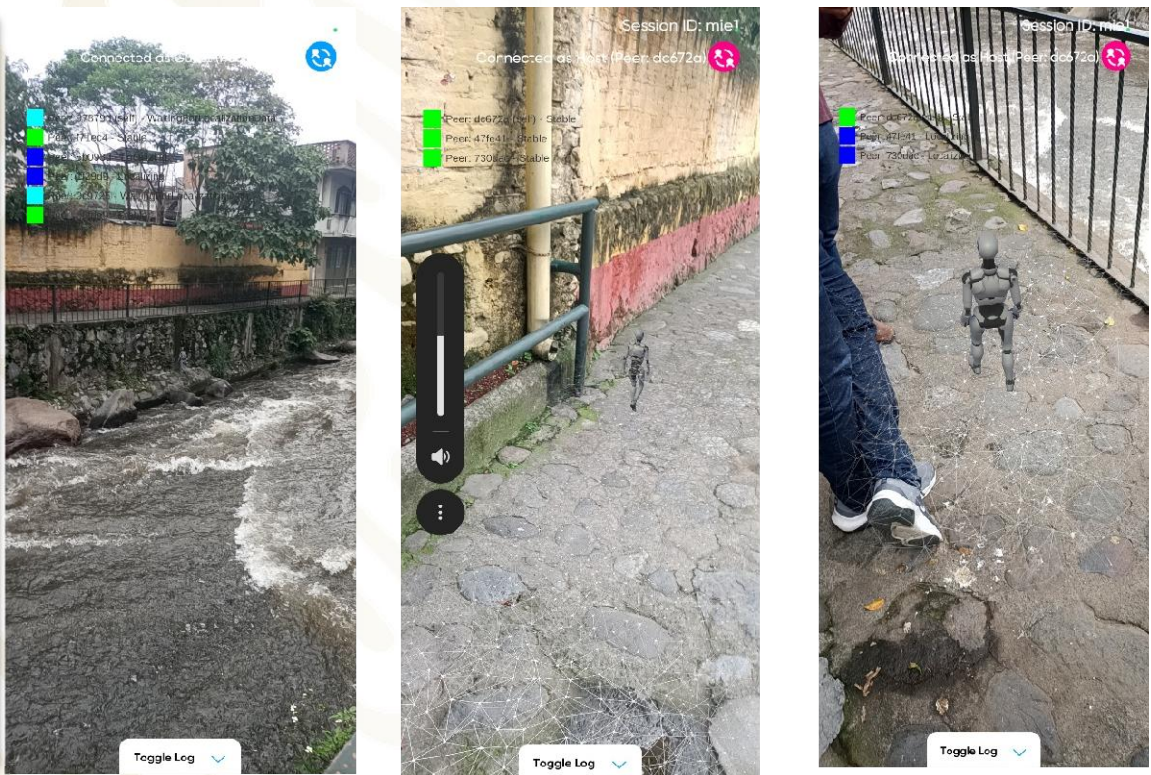


Figura 31:Dispositivo 2 este dispositivo usando la copia del modelo original esta se mueve en una posición diferentes pero en algunas se encuentra en la posición del original pero observado desde una posición más estable

Dispositivo 3

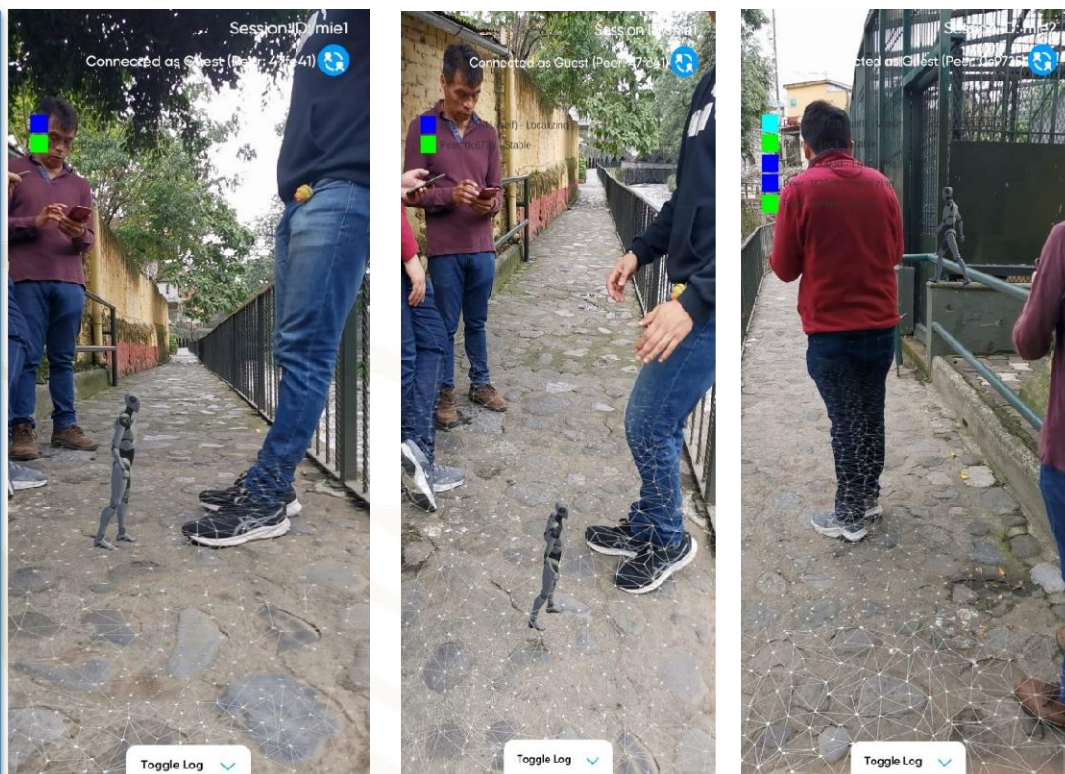


Figura 32: Dispositivo 3 el cual estuvo moviendo su modelo en diversas posiciones y distancias para medir hasta que profundidad y perspectiva podíamos ver el modelo que representa al guía

Dispositivo 4

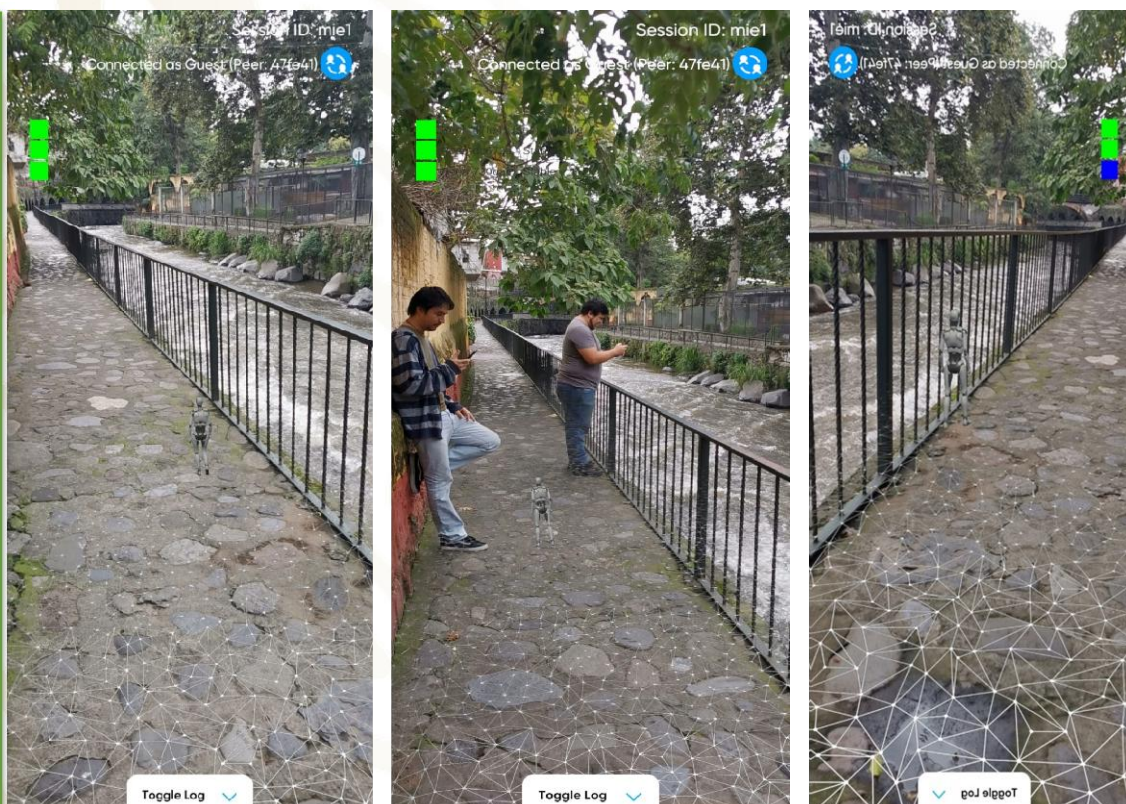


Figura 33. Dispositivo 4 fungiendo como usuario este dispositivo al igual que el anterior se encargaron de medir profundidad además de hasta donde se generaba y el punto de alcance de la misma

Como se observó en la última prueba se tuvieron varios dispositivos en simultáneo para probar hasta donde se logró generar el mallado, este mallado como se observa en la Figura 30, este dispositivo al ser el Host tenía control total de la sesión y cómo se observa en la serie de imágenes se ubicaron en diferentes posiciones para ir rotando los posicionamientos del guía de la sesión.

En el segundo dispositivo Figura 31 este dispositivo se encontraba en una ubicación diferente al Host aproximadamente unos 10 metros de distancia para corroborar hasta donde podía el lograr moverse y hasta en que puntos generaba la malla para poder ubicarse en otra posición, como se pudo observar en algunas de las pruebas se logró generar malla en espacios bastante fuera de un plano recto, esto ayuda a que nos podamos ubicar en la parte baja del río como se observa en las imágenes del dispositivo 2.

En el tercer dispositivo Figura 32, es el dispositivo que más estuvo cerca del HOST esto permitiendo un poco más exactitud en las posiciones, pero también observando las profundidades de movimiento del guía, se colocó en posiciones diversas y tomo algunas capturas más cercanas al guía para ver a que distancia y perspectiva poder observarlo sin perder de lado la ubicación del HOST.

El cuarto dispositivo Figura 33 estaba a una distancia de 3 metros del Host esto para corroborar un poco más las distancias de profundidad y perspectiva, así como la generación de la malla por parte del HOST, este dispositivo comprobó a que distancias podríamos ver más grande y pequeño al guía esto a su vez observar a que distancia puede el HOST llamar a todos a la posición de origen en donde el guía original se encuentre para que si se está teniendo un recorrido puedan continuar desde la donde el recorrido este actualmente sin afectar la sesión, pero también dando un poco de libertad a los usuarios de disfrutar de la experiencia.

El experimento realizado con anterioridad se realizó en la zona del río de Orizaba esta zona es una de las más concurridas en tiempos de vacaciones, esta zona como notamos en las imágenes anteriores es una zona boscosa lo cual en ciertas etapas de las pruebas nos causó algunas interferencias para algunos dispositivos, ya que dependiendo la versión de las antenas que vengan disponibles en cada uno de los dispositivos usados para estas pruebas dependía si tardaba un poco en localizar al jugador para poderlo ingresar a la sesión como observamos en la figura la zona que está marcada con un círculo es el radio de ubicación de la sesión el host está ubicado aproximadamente a 60 metros del grupo ellos seguían viendo todo lo que el host veía además de que jamás se perdió la localización de la sesión y aun que el host se moviera los jugadores se siguieron manteniendo dentro de la sesión.

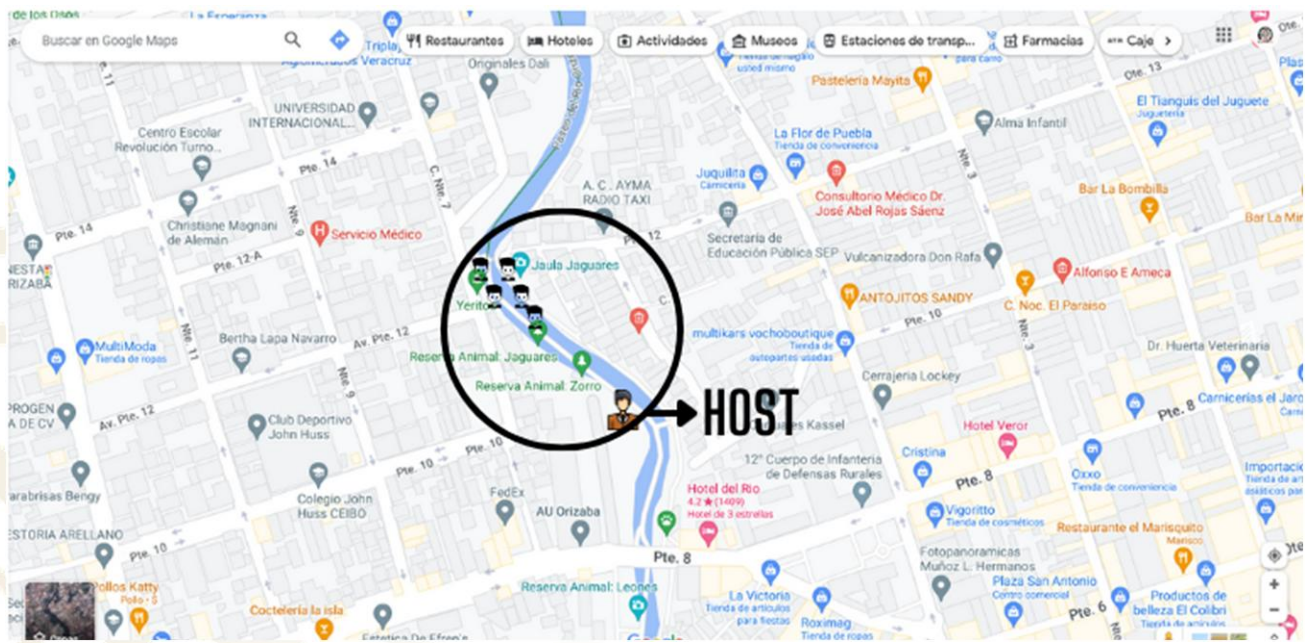


Figura 34: Mapa GPS del experimento en la zona del río Orizaba, el posicionamiento fue coordinado en este experimento para poder realizar diversas pruebas con cada uno de los jugadores para comprobar el alcance

La siguiente parte del experimento se realizó un poco más abajo en la misma área la situación que se generó aquí fue una distancia más separada entre cada uno de los jugadores así como una mayor distancia de apertura del Host esto para poder comprobar a que distancia se pudo seguir viendo el guía virtual así como a que profundidad se podían seguir del mismo, la creación de la malla fue simultánea, el host iba generando malla de camino nueva y los jugadores iban adaptándose a esta malla así como ellos iban generando una malla continua para que el guía virtual siempre tuviera una zona en donde poder desplazarse. En esta zona ya que no se tuvo mucha interferencia, mientras más se avanzaba la zona boscosa se reducía poco a poco, pero a pesar de eso tardaba un poco en unirse a la sesión esta parte del experimento se realizó una sesión nueva para ver en qué tiempos podíamos entrar y ver los tiempos de respuesta de cada dispositivo, cabe resaltar que cada dispositivo eran gamas media-baja y algunos eran media-alta, pero aun con este factor en vista el factor más importante es la velocidad de procesamiento de información de cada dispositivo.

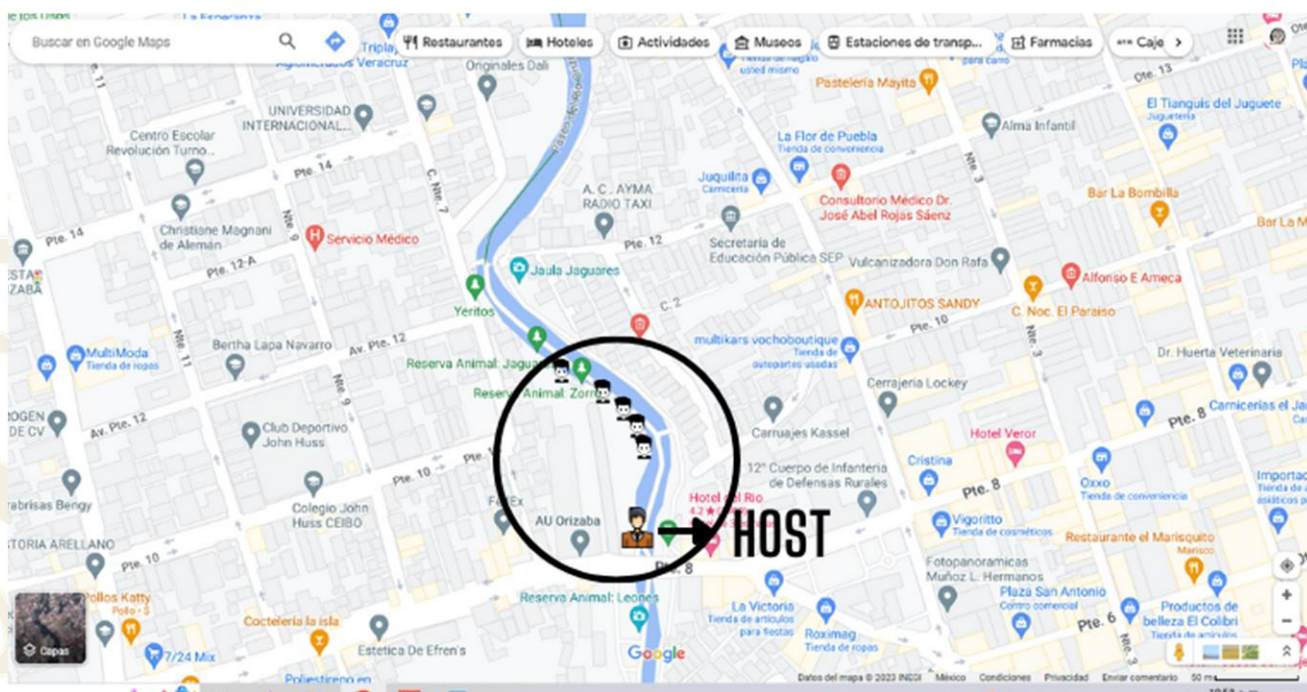


Figura 35: Experimento número 2 en el que realizamos una mayor distancia para este, el mayor obstáculo solo fue el procesamiento de los dispositivos ya que la detección de ubicación por medio de los GPS de cada dispositivo se realizó en un tiempo menor

Como se pudo observar en el mapa la ruta que se estableció no fue muy alejada de la primera pero la distancia a la que se colocó el host fue más alejada de los jugadores para poder crear una mayor amplitud de generación de la malla.

La malla que se generó en nuestro plano nos ayuda a ir generando nuevos caminos en los cuales los diferentes prefabs de cada jugador que está dentro de la sesión pueden ir siguiendo las rutas que establece el host o también ellos tener la libertad de poder moverse en diferentes posiciones, la malla se generaba en diversas posiciones no solo en los puntos en donde se enfocara si no hasta en el río generaban unos cuantos puntos también dentro de las jaulas.

Esta segunda parte del experimento su finalidad real fue ver hasta qué punto de distancia resistía la sesión sin perder conexión de que el host estaba muy alejado pero con ayuda del posicionamiento de cada celular iba configurando las distancias entre cada dispositivo permitiendo poder controlar la sesión a distancias largas y tal vez darnos una idea de que otras capacidades se podrían explorar a futuro con el proyecto.

CAPITULO 5

5. CONCLUSIONES.

A lo largo de esta tesis hemos analizado la funcionalidad de una aplicación y las diferentes alternativas que puede ofrecer en el futuro, para generar una aplicación optima y funcional para el entorno del turismo, su primer objetivo el ofrecer una alternativa viable y funcional que permita el recuperar sitios olvidados que no se toman en cuenta en la ciudad y volverlos a unir a parte de la historia como siempre se debió de haber hecho.

Comenzando desde el CAPÍTULO 1 el cómo se fue desarrollando la problemática a trabajar, así como el objetivo principal que era mejorar una aplicación que se dedica a encontrar sepulcros como los es la versión 1 de pepe. Tras estudiar exhaustivamente el conjunto de aplicaciones que se han intentado al paso de los años me di cuenta que aquí en la región, así como en el estado de Veracruz se tiene el potencial necesario para explotar este tipo de aplicaciones enfocadas a la realidad aumentada por el basto nivel de historia que existe.

La generación de sistemas de este tipo puede traer una extensibilidad en el turismo así como ayudar a la difusión de diferentes establecimientos que nacieron en la región ayudando a emprendedores a crear una nueva manera de promoción para ellos, algunas de las características fundamentales de esta aplicación ayudara a que sea amigable con teléfonos de gamas bajas, ya que se busca que todo mundo pueda tener el acceso a ellas sin la necesidad de tener teléfonos de gama plus como lo pueden los iPhone o los celulares Samsung de la gama Galaxy, ya que se buscó que teléfonos con capacidades media/bajas puedan soportar la aplicación y poder disfrutar de los atractivos que pueda ofrecer.

La principal carencia que puede a ver en este tipo de sistemas siempre será la compatibilidad de modelos y además la limitante en ciertos espacios para tener Datos o una Red WI-FI para poder conectarse, aunque la aplicación no requiera una cantidad exagerada de consumo de Megabytes de red, a veces la situación no nos puede brindar el poder pagar un paquete de datos para tener internet en nuestros dispositivos, pero con la posibilidad de que en algunas zonas de la ciudad se cuenta con internet gratuito y con la esperanza que en más zonas se habilite este servicio se buscaran métodos alternos para lograr un equilibrio para lograr que la aplicación tenga una mayor funcionalidad a futuro.

Para los fines de cumplir nuestros objetivos específicos de esta investigación se puede destacar que se logró cumplir cada uno de ellos, para poder entenderlo de una manera más detallada se hablara de ellos a continuación:

- El primer objetivo específico nos habla de introducir la realidad aumentada en diferentes punto, con ayuda de la generación de mallado en planos diversos nos ayuda a poder establecerla en cualquier zona, ya que el uso de puntos de interés no exige el uso de diversos tipos de luz para poder establecer que en ese punto queremos generar realidad aumentada creándonos varios conflictos por que depende mucho de que se quiera observar, usando esta malla en un plano nos permite estar en cualquier lugar sin depender de luz solo dependiendo de posicionamiento y procesamiento de información de los dispositivos.
- El segundo objetivo específico nos habla de sectores adaptados para el uso de realidad aumentada repitiendo un poco lo anterior, con ayuda de la generación de esta malla nos ayuda a poder establecer los puntos en cualquier zona la ventaja de todo esto que se va a poder usar hasta en zonas que son un poco complicadas de ingresar solamente usando esta malla nos permitirá tener las capacidades para poder usar esta tecnología en cualquier lugar.
- El tercer punto nos habla de la accesibilidad de esta tecnología en teléfonos de gama media-baja y media-alta lo cual podemos tener la certeza de que se logró ya que la aplicación fue probada en teléfonos tanto gamas bajas, medias y altas para tener la seguridad de que cuales dispositivos podrían funcionar con esta, teniendo algunos excluidos como algunos modelos de Huawei, OPPO y motorola, hablando en un porcentaje el 90% de dispositivos en los que se probó funciono la aplicación perfectamente, el otro 10% fue por problemas de compatibilidad con servidores.
- El cuarto y último punto nos habla de la mejora continua de la aplicación, Unity va mejorando continuamente sus versiones manteniendo constantemente actualizadas las versiones de Android y IOS para el desarrollo de aplicaciones así como los paquetes y librerías de desarrollo para aplicaciones de realidad aumentada permitiendo de esta manera tener continuamente actualizado cada versión que salga de los sistemas operativos de Android y IOS actualmente se realizó la aplicación para Android 7.0 que es el más bajo en la actualidad pero en un plano a futuro se espera que la aplicación se puede generar para una versión mayor adaptada para que versiones más abajo puedan soportarlo sin problema.

5.1 Trabajo a futuro

El trabajo en esta tesis nos da muchas aperturas para diversas mejoras que se pueden aplicar a esta aplicación ya que tener la ventaja de desarrollar en Unity, nos habilita muchas herramientas de desarrollo de animación, modelado, etc. Permitiendo muchas posibilidades, una de las mejoras que se plantean a futuro para la aplicación es el ingreso de modelos prefabricados más vistoso en conjunto con animaciones más exigentes, este tipo de trabajo requiere mucho trabajo pero aprendiendo las herramientas necesarias para esto podremos implementar animaciones aún más realistas que nos puedan ayudar a crear ambientes turísticos con mucho más vivacidad, atrayendo a gente que nunca le han atraído algunas atracciones y aumentando más el tráfico de gente en los atractivos turísticos y dando enfoques muchos más actuales.

Otra mejora que a futuro se tiene planteada es generar rutas móviles en conjunto con los diversos transportes que se tienen actualmente en la ciudad como lo es el tren chipi-chipi, el autobús el gallo y el Oribus, ayudando a las personas que realizan los recorridos dentro de los autobuses como los son el Oribus y el chipi-chipi ayudando que las personas tengan una experiencia mucho más didáctica y que disfruten más del viaje, ya que con ayuda de la realidad virtual se pueden crear actividades dentro del recorrido, ahora con el autobús del gallo se puede incluir algo que llame más la atención a solo usarlo de transporte como el crear una animación que de información de precios, datos importantes de los lugares que van pasando así como algunas promociones que se pueden tener en ese momento en los lugares donde se tengan zonas de comida, creando un modo nuevo para publicitar tu negocio.

Apéndice A: Manual de diseño de la aplicación en unity.

En este apéndice se habla de cómo elaborar una aplicación en unity usando Lighship y ARDK se explicará a detalle cómo se deben importar y exportar librería, hasta como debemos preparar todo lo necesario para generar la aplicación.

1. Como primer paso si no tenemos el UnityHub debemos descargarlo desde la página oficial de unity o usando el siguiente enlace: <https://unity.com/es/download>
2. Una vez descargado el UnityHub creando un nuevo proyecto, como se muestra en la (Figura 36) crearemos un 3D core asignando un nombre y daremos click en créate project.

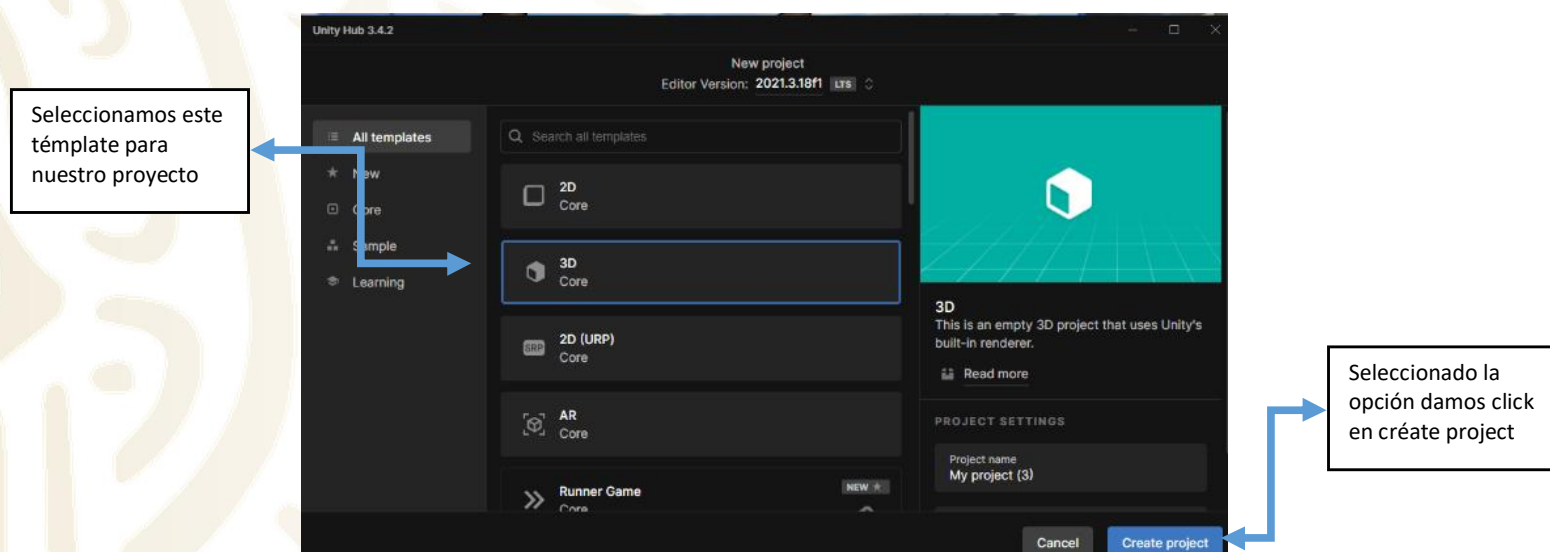


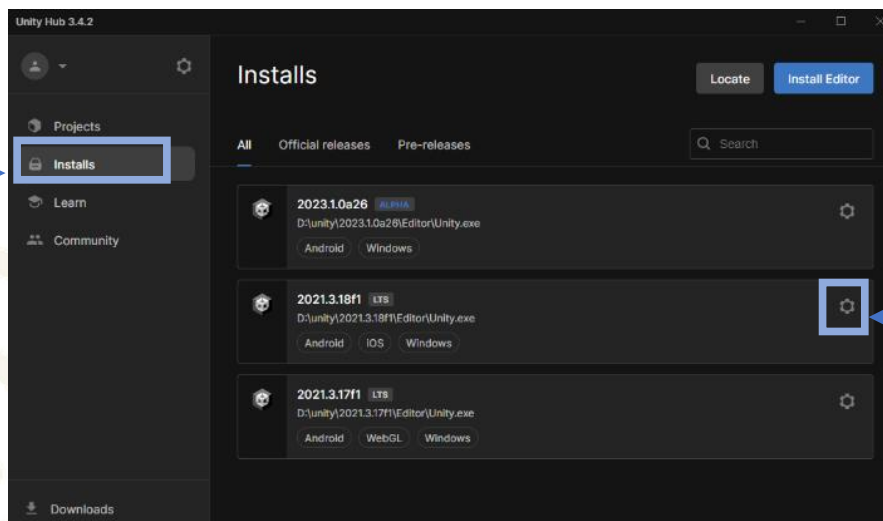
Figura 36: Ventana principal de Unity HUB en la cual podremos crear nuestros proyectos

3. Una vez creado el proyecto se deben añadir unos módulos a nuestro editor de unity, ya que si no se instalan el editor no nos permitirá generar aplicaciones Android o IOS y no se podrán cambiar nuestras preferencias a los mismos.

Primero se dará click en el apartado de installs como se muestra en la Figura 37 después ubicaremos que versión de editor estamos usando y daremos click en el engrane como se muestra en la figura. Una vez dado click nos abrirá una ventana desplegable en la cual daremos click en el apartado **add modules** como se muestra en la Figura 38.

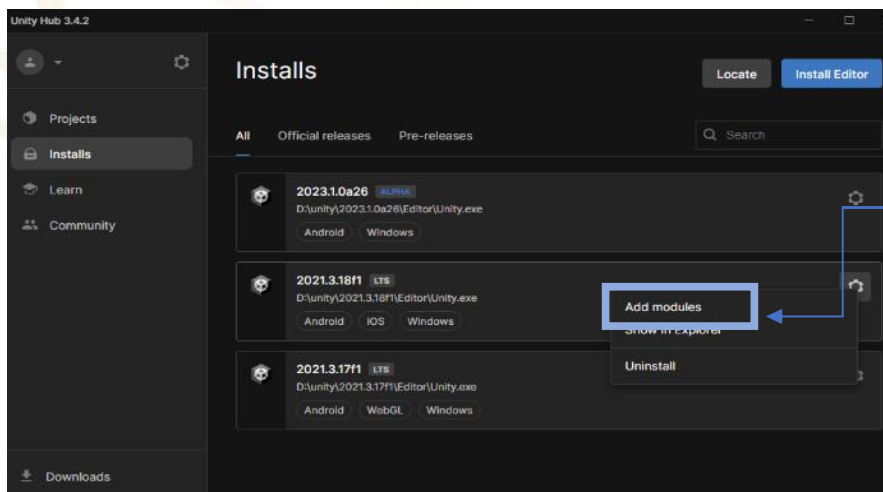
Una vez dentro añadiremos los módulos que necesitaremos, para el caso de este proyecto usamos Open JDK y Android SDK & NDK Tools, seleccionados los módulos daremos click en install todo esto lo podemos observar en el ejemplo de la Figura 39.

Damos click en installs



Damos click aquí para abrir la sub-ventana

Figura 37: ventana para seleccionar módulos



Daremos click en Add modules

Figura 38: sub-ventana para agregar módulos al proyecto en unity

En este apartado seleccionaremos que módulos necesitamos y para cuantas plataformas estamos desarrollando

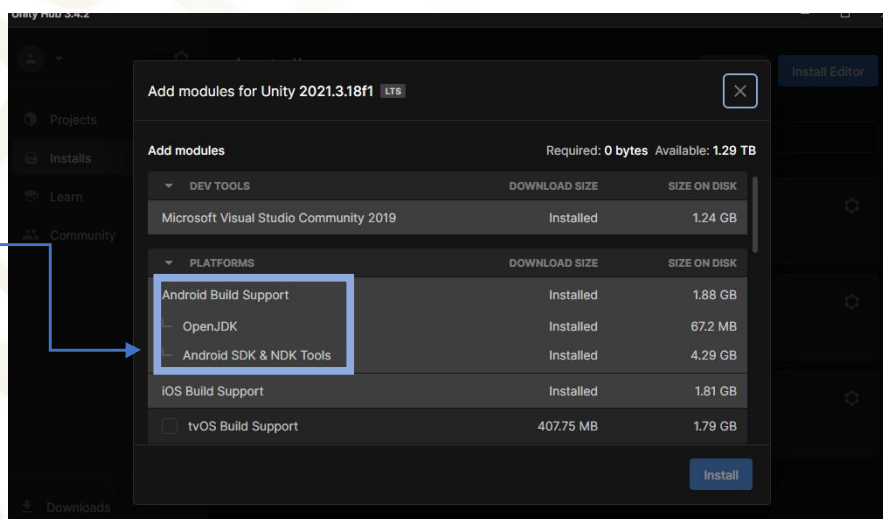


Figura 39: Ventana donde podremos descargar y agregar los módulos necesarios para crear nuestro proyecto en unity

4. Una vez cargados los módulos podemos crear el proyectos dependiendo la velocidad de procesamiento de su tarjeta gráfica es el tiempo en que unity creara su proyecto.

- **Nota: Si tu objetivo es desarrollar en unity debes tomar en cuenta las siguientes especificaciones que están a continuación:**

	REQUISITOS MINIMOS	REQUISITOS RECOMENDADOS
SISTEMA OPERATIVO	Windows 7(SP1+), windows 10. versiones de 64 bits	Windows 11 64 bit
CPU	Arquitectura x64 con instrucciones sse2 (Intel Core 2 Duo, Extreme o Quad, Intel Core i3, AMD Sempron, Phenom I y II y Athlon II)	Arquitectura x64 con instrucciones sse2 (Intel Core i5, i7, AMD FX Y Ryzen 7)
API DE GRAFICOS	DX 10 Y 11	DX 12
REQUISITOS ADICIONALES	Drivers de hardware compatibles	Drivers de hardware compatibles
GRAFICOS	Tarjeta de video de 512 Mb (GTX 650 en adelante)	RTX 3070(para proyectos 3D)
RAM	Desde 4Gb	16Gb
ESPACIO EN DISCO	10Gb	10Gb

Tabla 2: Especificaciones mínimas y recomendadas para desarrollar en Unity

5. Ahora debemos ir a la página <https://lightship.dev/> en la cual deberemos crear una cuenta para poder acceder a las herramientas que Niantic nos permitirán desarrollar aplicaciones de realidad aumentada.
6. Una vez creada nuestra cuenta nos iremos al apartado de Resources como se muestra en la Figura 40

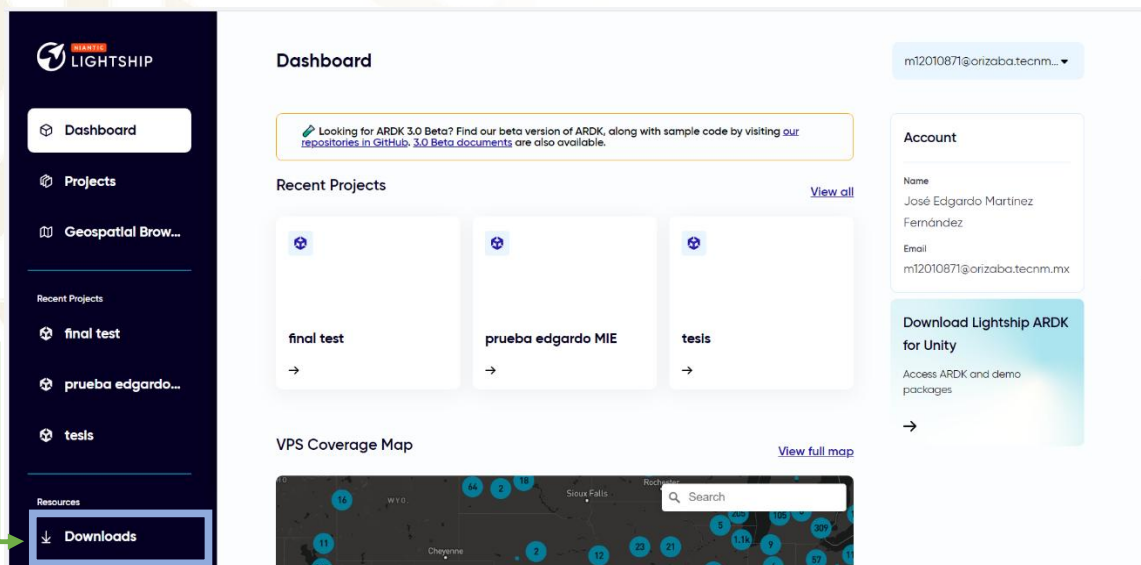


Figura 40: Ventana principal de la cuenta de Lighship ARDK para desarrollo de aplicaciones.

7. Una vez dentro de este apartado daremos click en el botón desplegable de ARDK y descargaremos todo lo que está dentro de este módulo como se indica en la Figura 41

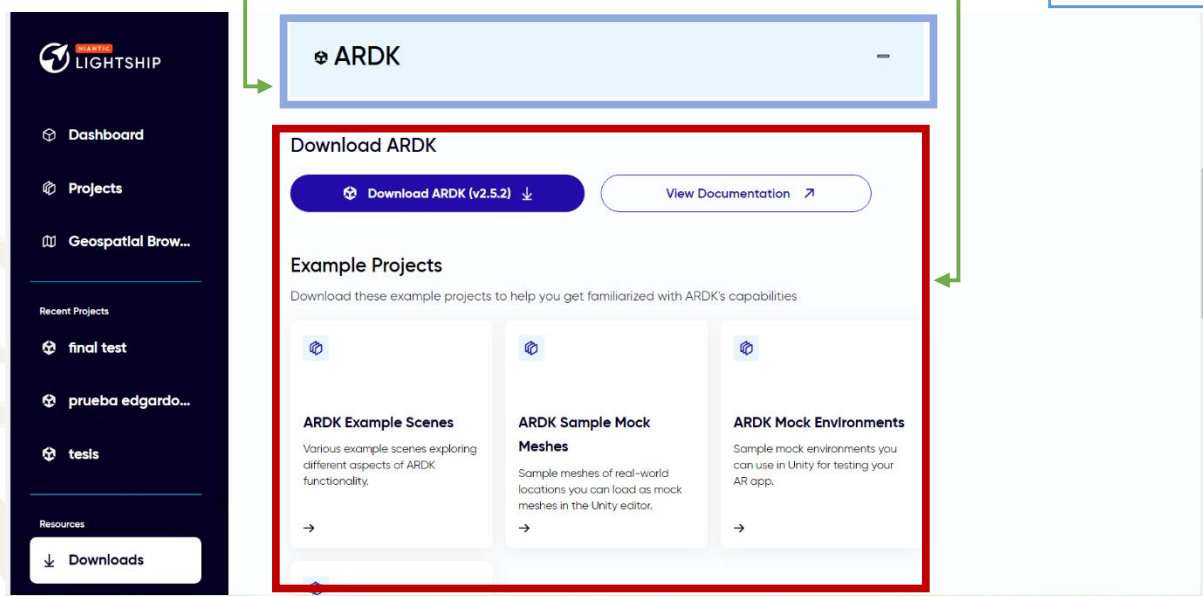


Figura 41: Ventana de Resources donde podemos descargar las herramientas necesarias para desarrollar en Unity en conjunto con Lighthouse ARDK

8. Una vez descargado todo lo necesario volveremos a unity para instalar cada package descargado, como se indica en la Figura 42 nos iremos a Assets, import Package se deben instalar en el siguiente orden:

- ARDK
- ARDK EXAMPLE SCENES
- ARDK SAMPLE MOCK MESHES
- ARDK MOCK ENVIROMENT
- LIGHSHIP HUB

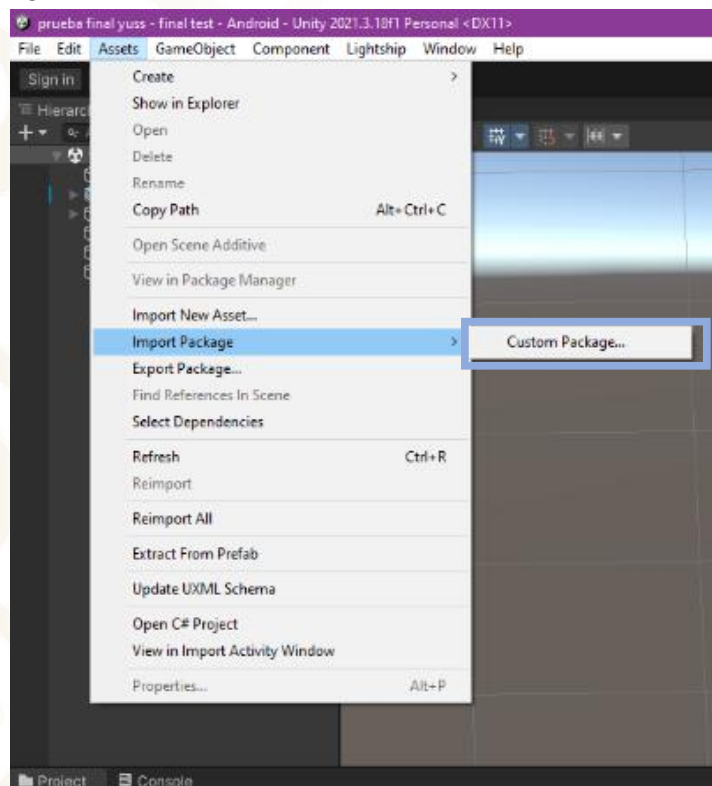


Figura 42: Apartado en donde instalaremos los packages para dev en unity

9. Una vez instalado todo deberemos tener las siguientes carpetas en nuestro apartado Assets de nuestro proyecto:
- ARDK
 - ARDKEXAMPLES
 - ARDKMOCKENVIROMENT
 - LIGHSHIPHUB
 - LIGHSHIPHUMOCKENVIROMENTS
 - MOCKMESHES
10. Si revisamos el apartado consola a veces nos aparecerá un error que no hemos autenticado el proyecto, para esto debemos regresar a la página de Lighship.Dev e ir al apartado projects y dar click en new Project.

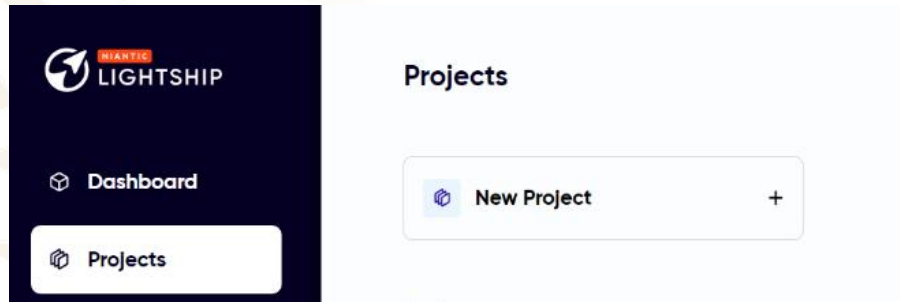


Figura 43: Apartado en el cual podremos crear nuestro autenticador del proyecto

11. Una vez creado el proyecto le asignaremos un nombre copiaremos nuestra API Key para autenticar el proyecto en unity.

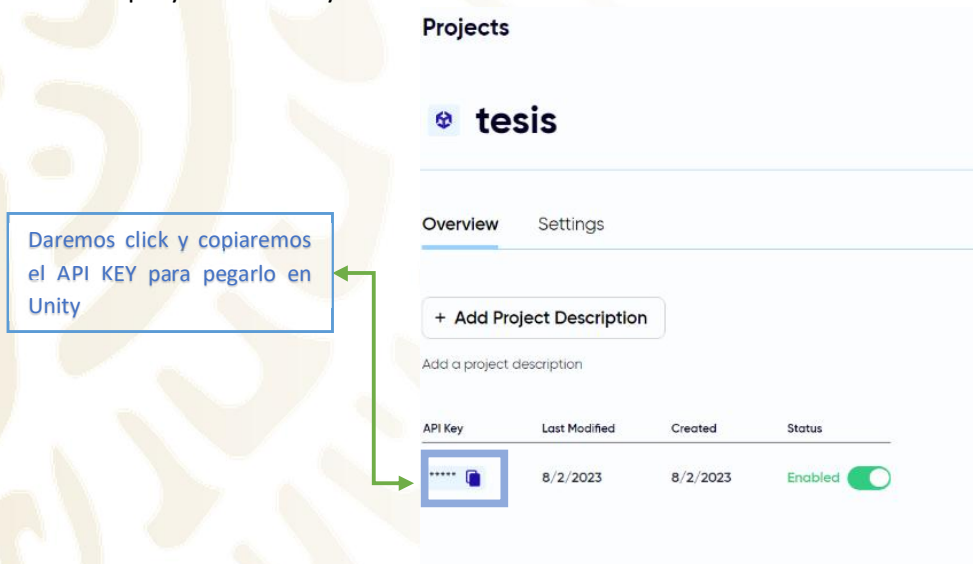
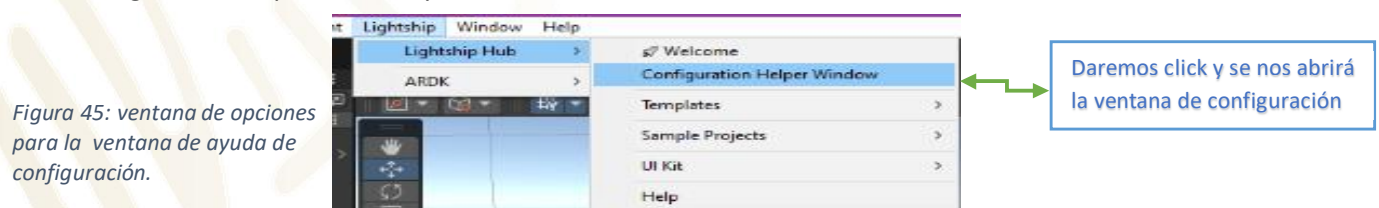


Figura 44: Ventana donde podemos copiar nuestra clave API KEY para poder autenticar el proyecto

12. Una vez que ya copiemos el API KEY regresaremos a unity y nos iremos al apartado de Lighship → configuration helper window y daremos click.



13. Al dar click nos aparecerá una nueva ventana en la cual solo debemos pegar la llave que copiamos con anterioridad y dar click en setup.

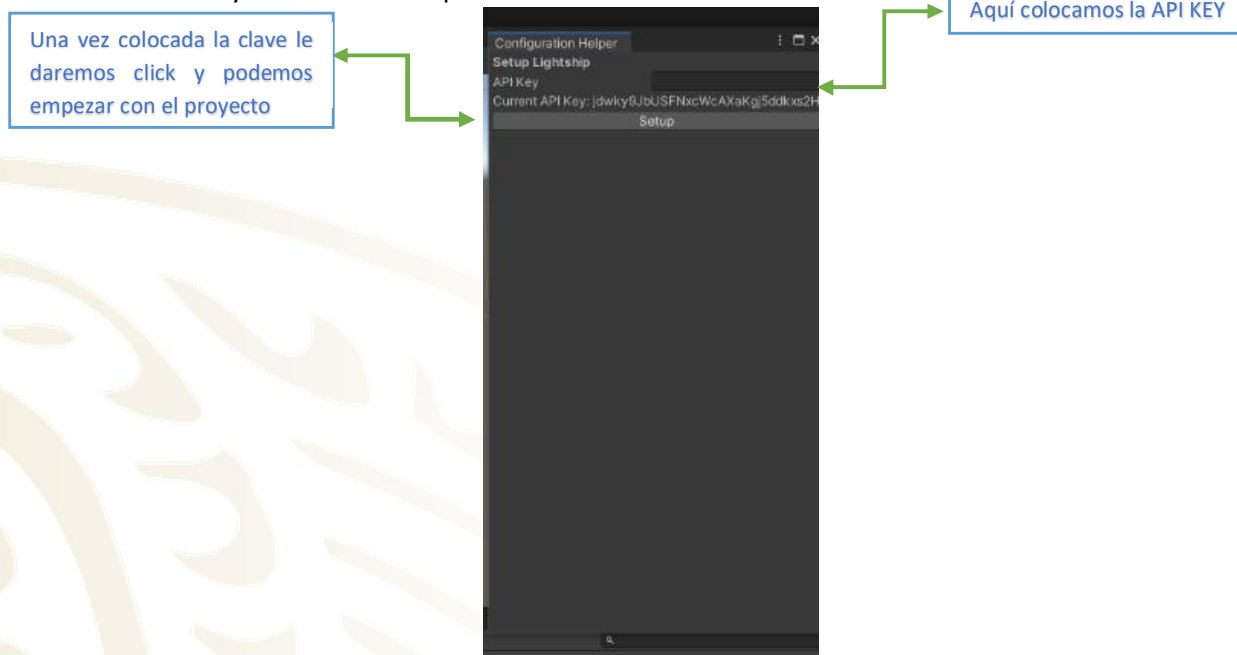


Figura 46: Ventana para colocar la API KEY de nuestro proyecto

14. Antes de iniciar con el proyecto dejaremos las configuraciones de jugador y las configuraciones de Android necesarias para el proyecto, para esto nos iremos a file → build settings → player settings.

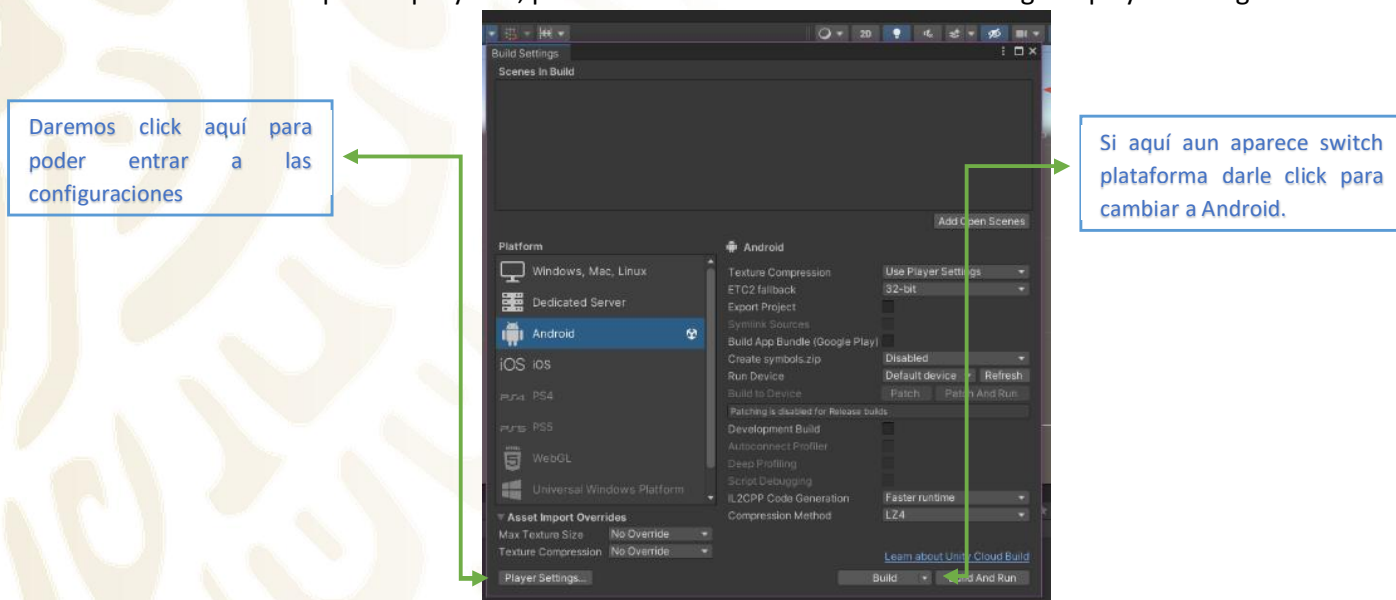
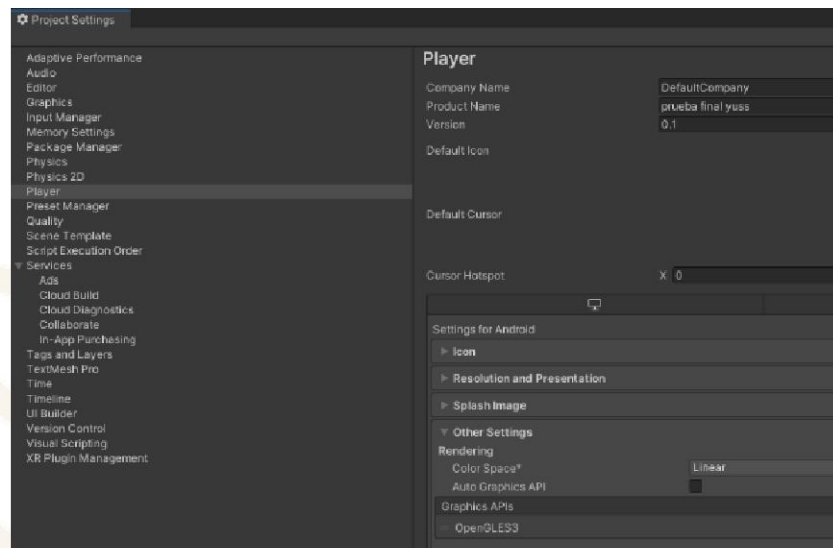


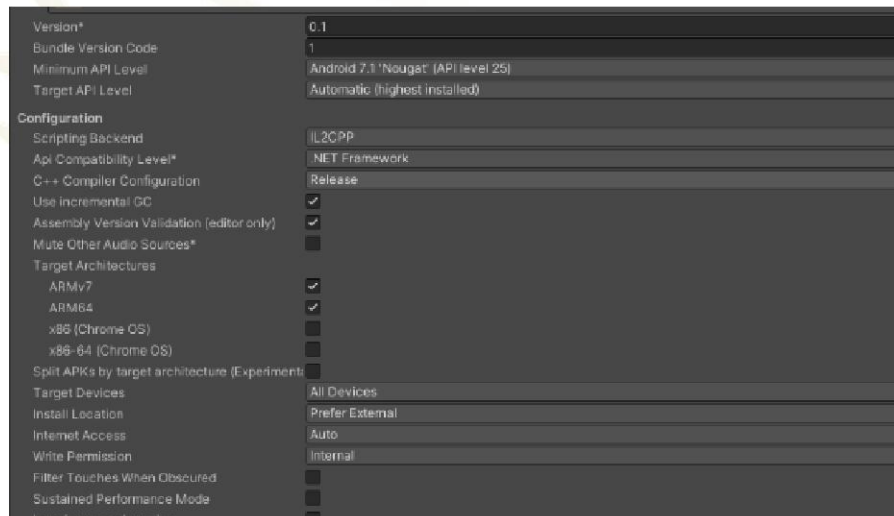
Figura 47: Ventana de configuraciones de construcción de los proyectos

15. Una vez adentro cambiaremos las configuraciones que están marcadas en la Figura 48 siguiendo el orden en el cual están acomodadas.



Eliminamos Vulkan

Si esta activada la desactivamos

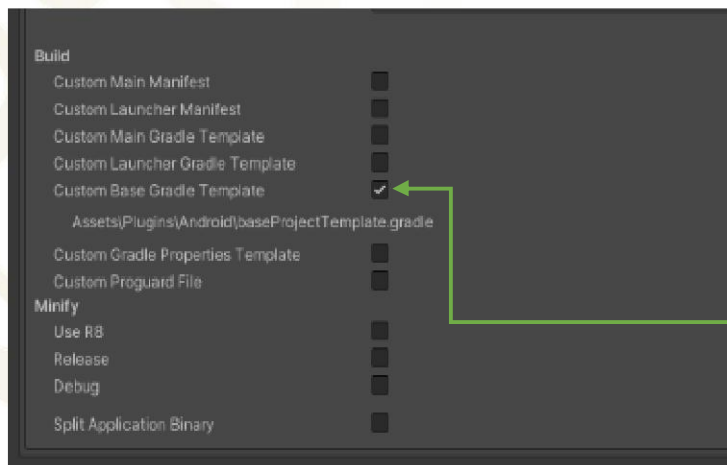


Seleccionamos el nivel de API que queremos desarrollar

Cambiamos de mono a IL2CPP

Ponemos el framework para el vamos a desarrollar

Activamos ARMX64



Iremos al apartado de publish settings para activar esta casilla esto nos generara un archivo

Figura 48: Secuencia en la cual configuremos los player settings de nuestro proyecto.

- El archivo generado anteriormente se encuentra en Assets → Plugin → Android, abrimos el archivo y como se indica en la Figura 49 y ponemos la siguiente línea de código: `classpath'com.android.tools.build:gradle:4.2.0'`

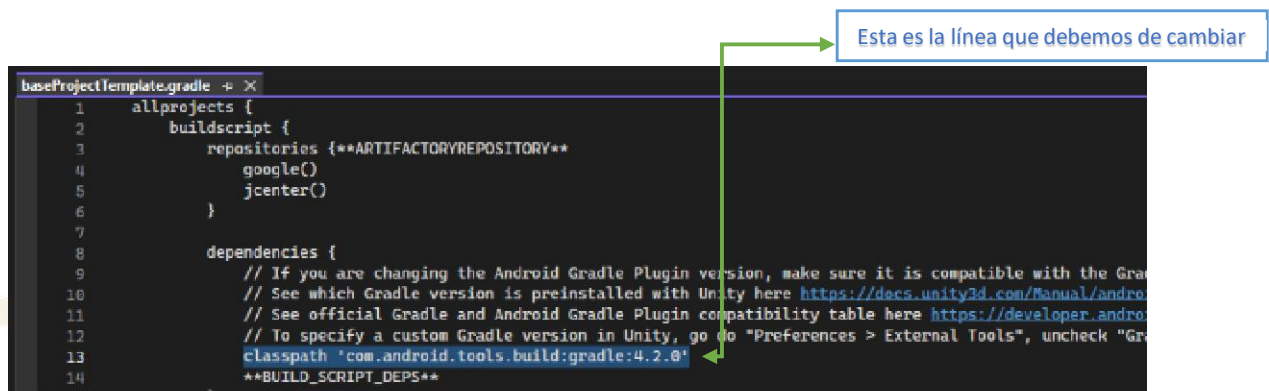


Figura 49: Código del gradle template generado en donde modificaremos una línea de código a la versión más reciente

17. Una vez hecho esto debemos instalar gradle en la versión que este activa. Para descargar el archivo de gradle usaremos el siguiente link → <https://gradle.org/releases/>
18. Cuando tengamos el archivo descargado debemos de regresar a Unity para instalarlo y seguiremos la siguiente ruta Edit → Preferences → External Tools y pegaremos la ruta de la carpeta donde tengamos el archivo como se indica en la imagen siguiente

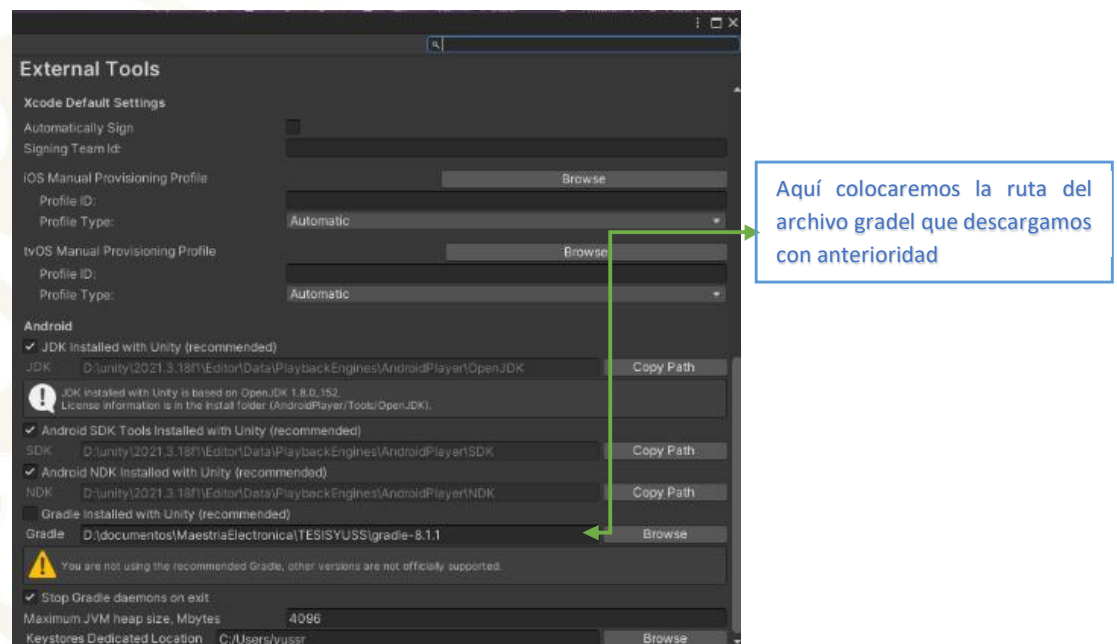


Figura 50: Ventana de external preferences en donde colocaremos la ruta de nuestro archivo gradle.

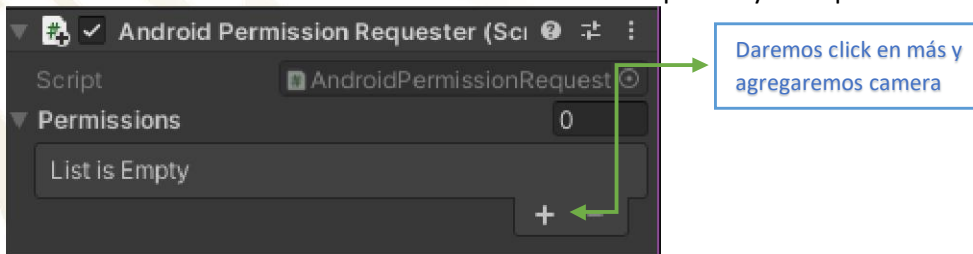
19. Una vez terminado podemos empezar a realizar el proyecto, el primer paso es buscar en la carpeta Assets → ARNetworkingSceneManager, este archivo lo moveremos a nuestra jerarquía y borraremos main camera.
20. Ahora ya teniendo el Game Object en nuestra jerarquía iremos a nuestro inspector para saber cómo funciona ya que podemos observar que tenemos varios scripts dentro este, que son los siguientes:
 - **AR Session Manager:** El cual es el encargado del ciclo de la experiencia de realidad aumentada, dentro de esta hay una casilla que debemos que tener marcada **Use With AR Network** ✓ si no está marcada debemos marcarla.

- Capability Checker: Este script es el encargado de verificar si el o los dispositivos cuentan con las capacidades necesarias para soportar ARDK.
- Network Session Manager: Este script se encarga de crear, ejecutar proporcionar una referencia MultiP erNetworking que nos ayudara a poder crear un espacio en donde pueden entrar muchos usuarios a la experiencia de realidad aumentada.
- ARNetworkingManager: Este script se encarga crear, ejecutar una referencia para Networking sesi n manager.

21. Antes de continuar debemos agregar un componente extra en el bot n add componente



22. Una vez dado click buscaremos Android Permission Requester y nos aparecer la siguiente ventana:



23. Para poder revisar nuestro avance podemos usar el virtual estudio, para esto debemos habilitarlo para esto iremos a Lighthouse → ARDK → Virtual Estudio.

24. Una vez dentro iremos al apartado de mock Scene y seleccionaremos InteriorScene como nos indica la figura siguiente:

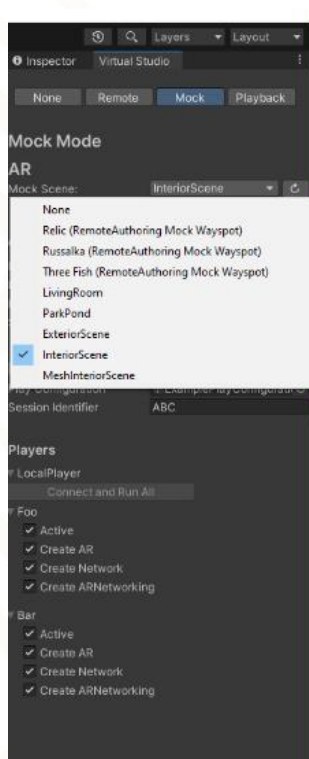


Figura 51: Ventana de virtual estudio que nos ayudara a simular un entorno virtual en realidad aumentada para probar nuestro proyecto

25. Para terminar de configurar en el apartado Multiplayer daremos click aqu  para poder habilitar el modo multijugador de pruebas.



26. El siguiente paso es generar nuestra interfaz, para esto usaremos las herramientas que nos brinda Lighship para crear una interfaz sencilla primero iremos a Lighship → LighshipHub → UI Kit → Core Components y seleccionamos el elemento que deseamos en este caso usaremos un Primary Button.
27. El siguiente elemento que generaremos es un Input Field así que siguiendo la misma ruta a Lighship → LighshipHub → UI Kit → Core Components la agregaremos y lo acomodaremos usando las siguientes medidas en nuestro inspector, este lo colocaremos en +65 en posición en Y el Button lo colaremos en -65 para que queden a una posición contraria.
28. El siguiente elemento que agregaremos a nuestra interfaz será una identificador de sesión para esto seguiremos la siguiente ruta Lighship → LighshipHub → UI Kit → Status Components y agregaremos Network Status.
29. Una vez agregado lo duplicaremos para generar un duplicado de este mismo, el cual dentro de nuestra jerarquía lo llamaremos SessionID, dentro de nuestro inspector borrarémos el texto que contiene este elemento duplicado. La función de este duplicado es mostrar el nombre que se le dio a la sesión que esta actualmente activa.
30. Como último elemento agregaremos un Scrolling Log Pannel esto para siempre saber posiciones, nombres de usuario entre otros detalles de la sesión lo agregamos siguiendo la misma ruta anterior Lighship → LighshipHub → UI Kit → Status Components
31. Como paso extra buscaremos dentro de nuestros assets el archivo SyncStateTrackingList ya que nos ayudara a siempre saber los estados de los usuarios que se encuentren dentro de la aplicación. Esta la arrastraremos a nuestro canvas para finalizar nuestra interfaz.
32. Teniendo todo esto listo referenciaremos nuestro InputField que se encuentra dentro de nuestra jerarquía como se muestra en la Figura 52:

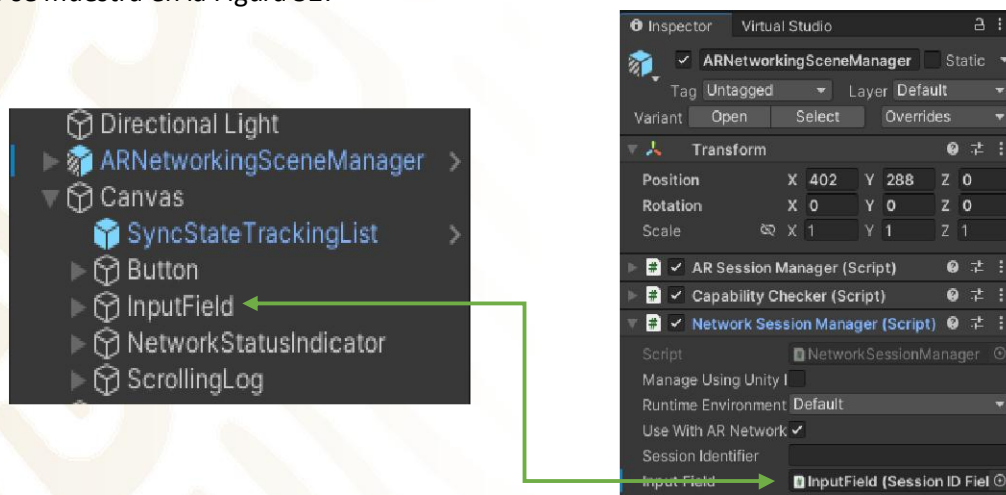


Figura 52: Esta referencia nos permitirá controlar todo lo relacionado al canvas desde un punto específico

33. En el inspector verificamos que en AR Networking Manager la casilla **Manage Using Unity I** no este activada, en caso contrario debemos desactivarla.

34. Dentro de nuestra jerarquía iremos a Button y como se muestra en la Figura 53 referenciaremos ARNetworkingSceneManager que está en nuestra jerarquía, primero daremos click en más, después arrastremos el objeto antes mencionado.

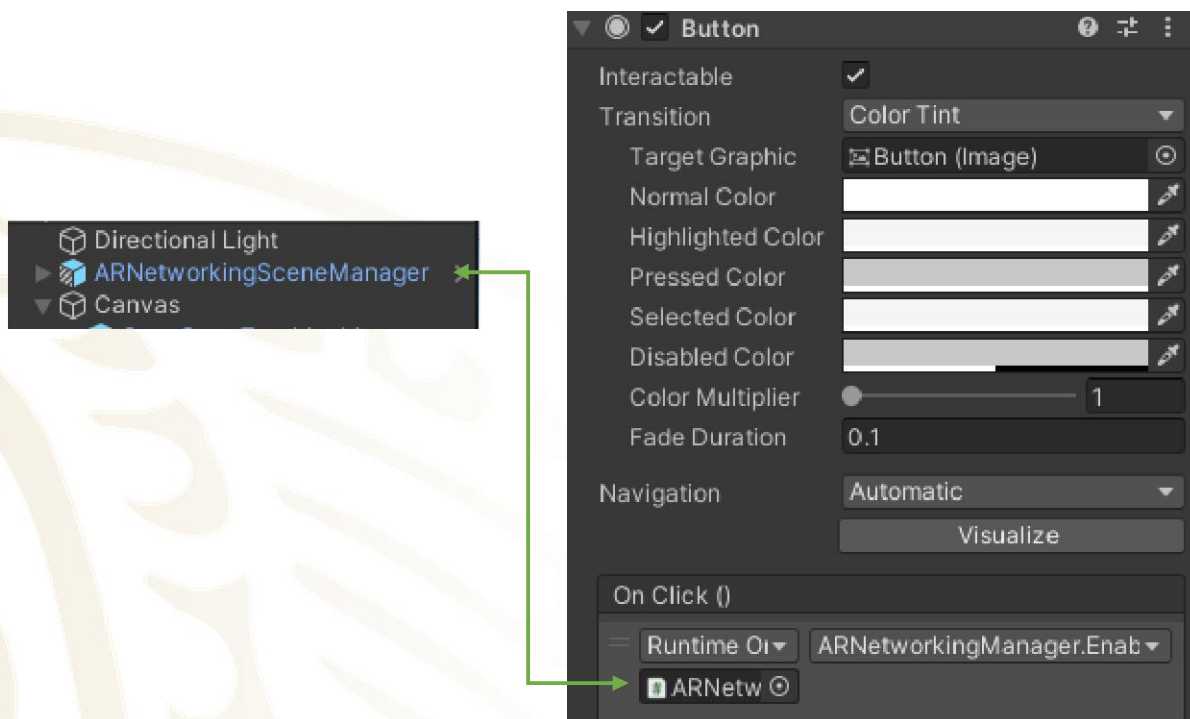


Figura 53: Esto es para agregarlo a las funciones completas de nuestro proyecto y controlarlo desde las escenas principales

35. Realizado lo anterior debemos activar la función siguiendo esta ruta ARNetworkingManager → EnableFeatures(). Terminado eso dentro Button en la jerarquía debemos cambiar el texto para darle un enfoque diferente a su aplicación.
36. Ahora crearemos un evento system siguiendo la siguiente ruta primero daremos click derecho → UI → Event System.
37. Dentro de assets crearemos una nueva carpeta que llamaremos Scripts aquí colocaremos cada uno de los códigos de nuestra aplicación para poder programar su funcionamiento.
38. El primer Script que generaremos para esta aplicación lo llamaremos UIManager el cual será el encargado de la interfaz de nuestra aplicación,
39. Para el script siguiente usaremos el código que se anexa a continuación en la Tabla 3:

```

1
2 using System.Collections;
3 using System.Collections.Generic;
4 using UnityEngine;
5 using UnityEngine.UI;
6 public class UIManager : MonoBehaviour
7 {
8     [SerializeField] private Button joinButton;
9     [SerializeField] private InputField sessionIDInput;
10    [SerializeField] private Text sessionIDText;
11
12    public void PlayerConnected()
13    {
14        sessionIDText.text = string.Format("Session ID: {0}",
15sessionIDInput.text);
16        joinButton.gameObject.SetActive(false);
17        sessionIDInput.gameObject.SetActive(false);
18    }
19

```

Tabla 3: Script UIManager el cual se encarga de controlar el Frontend de la aplicación

40. Teniendo nuestro script listo volvemos a unity y crearemos un objeto vacío usando la siguiente ruta Click derecho → Create Empty, este objeto lo llamaremos UIManager.
41. Una vez creado agregaremos el script a este objeto arrastrándolo hacia el inspector y nos aparecerán unos campos como se muestra en la figura siguiente, consiguiendo referenciar los campos tal y como se muestra a continuación:

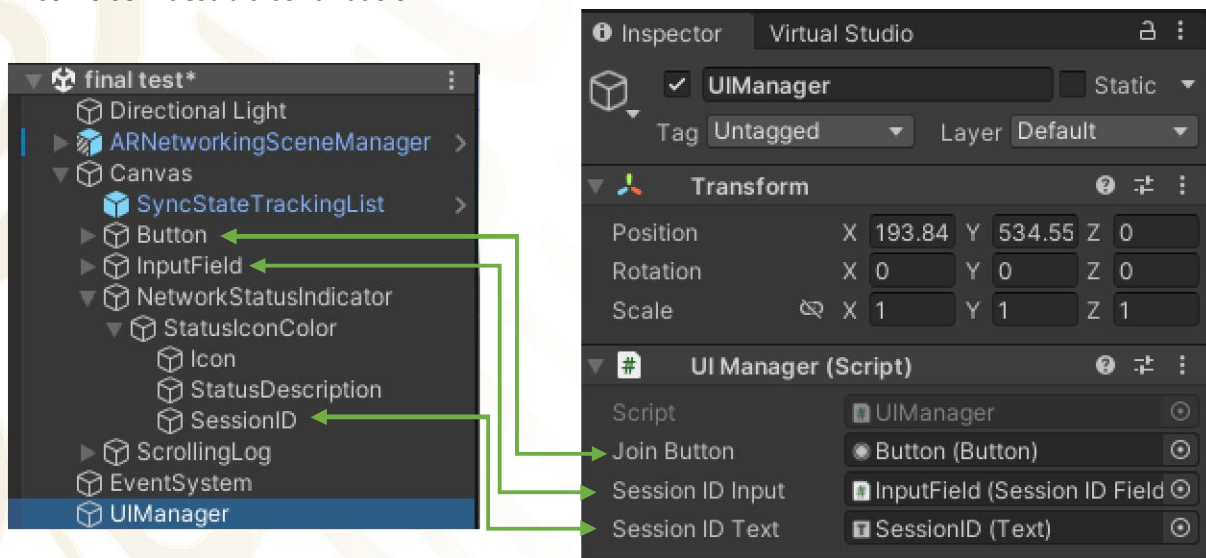


Figura 54: Campos a referenciar en el objeto UIManager en conjunto con el script ya antes creado

42. Una vez hecho esto iremos a nuestro botón y agregaremos nuestro UIManager a este, daremos click en más y referenciamos como en el paso 35 y dentro de la función buscaremos lo siguiente UIManager → PlayerConnected().
 - **NOTA:** Si queremos comprobar nuestro avance podemos dar click en play y corroborar que funcione todo correctamente colocando un ID y nos mostrara el espacio virtual de virtual estudio de ARDK y realizaremos los siguientes cambios como se muestra en la figura.

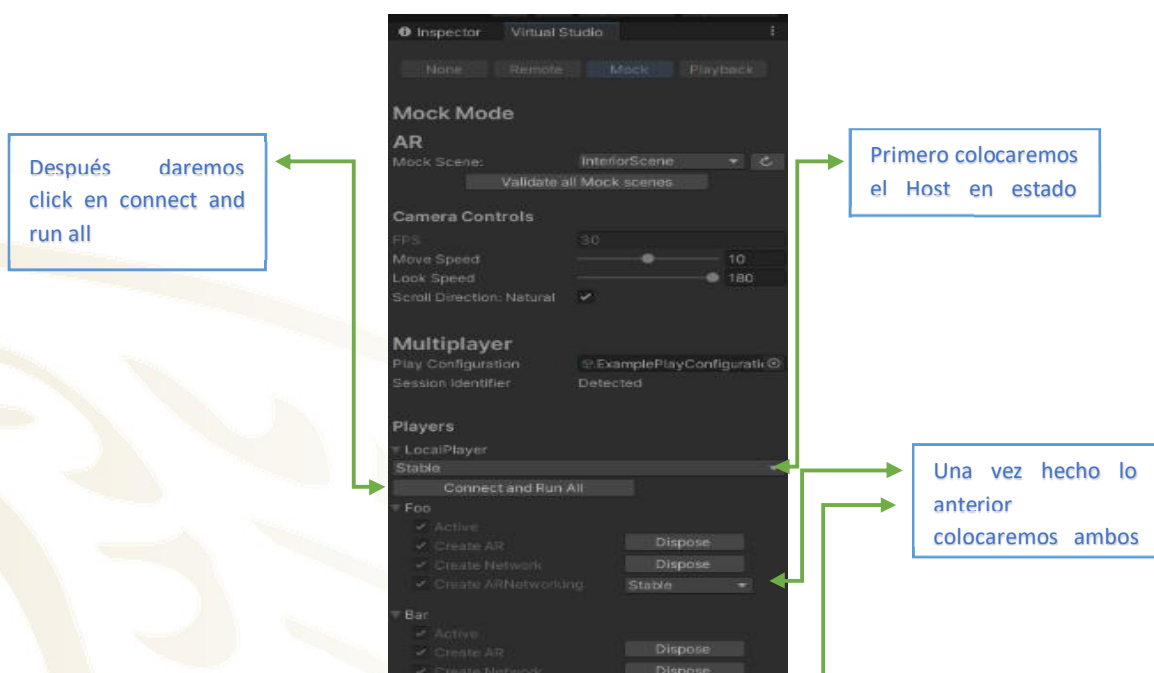


Figura 55: Ventana de virtual estudio y como corroborarán nuestros avances en nuestro proyecto

43. Ahora vamos a crear los siguientes scripts estos llevaran por nombre GameManager y NetARSessionManager.
44. El primer script que realizaremos será GameManager ya que es uno de los más importantes dentro del proyecto. Usaremos el código que está a continuación:

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

public class GameManager : MonoBehaviour
{
    public static GameManager Instance;
    private NetARSessionManager sessionManager;
    public NetARSessionManager SessionManager { get => sessionManager; }

    private void Awake()
    {
        if (Instance != this && Instance != null)
        {
            Destroy(this);
        }
        else
        {
            Instance = this;
        }
    }
}
```

Tabla 4: Inicio del Script GameManager, esta primer partes es para establecer conexión con el script NetARSessionManager.

45. Terminado el script anterior pasaremos al segundo script de esta sección que es NetARSessionManager para el cual usaremos el código que presentamos a continuación:

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

using Niantic.ARDKit.Networking;
using Niantic.ARDKit.AR.Networking;
using Niantic.ARDKit.Utilities;
using Niantic.ARDKit.AR.Networking.ARNetworkingEventArgs;
using Niantic.ARDKit.Networking.MultipeerNetworkingEventArgs;

public class NetARSessionManager : MonoBehaviour
{
    [SerializeField] private Camera aRCamera;
    [SerializeField] private GameObject playerPrefab;

    private bool isHost;
    private bool isSessionStable;
    private IPeer host;
    private IPeer player;

    public Camera ARCamera { get => aRCamera; }
    public bool IsHost { get => isHost; }
    public bool IsSessionStable { get => isSessionStable; }
    public IPeer Host { get => host; }
    public IPeer Player { get => player; }

    private IARNetworking aRNetworking;
    private Dictionary<IPeer, GameObject> playerDictionary = new Dictionary<IPeer,
GameObject>();
    // Start is called before the first frame update
    void Start()
    {
        ARNetworkingFactory.ARNetworkingInitialized += OnARNetworkingInitialized;
    }

    private void OnARNetworkingInitialized(AnyARNetworkingInitializedArgs args)
    {
        aRNetworking = args.ARNetworking;
        aRNetworking.Networking.Connected += Networking_Connected;
        aRNetworking.PeerStateReceived += ARNetworking_PeerStateReceived;
        aRNetworking.PeerPoseReceived += ARNetworking_PeerPoseReceived;
    }

    private void ARNetworking_PeerPoseReceived(PeerPoseReceivedArgs args)
    {
        if (!playerDictionary.ContainsKey(args.Peer))
        {
            playerDictionary.Add(args.Peer, Instantiate(playerPrefab));
        }
        if (playerDictionary.TryGetValue(args.Peer, out GameObject player))
```

```

        {
            Vector3 playerPosition = args.Pose.ToPosition();

            #if UNITY_EDITOR
                playerPosition = playerPosition + new Vector3(0, 0.75f, 0);
            #endif

            player.transform.SetPositionAndRotation(playerPosition,
args.Pose.rotation);
        }
    }

    private void ARNetworking_PeerStateReceived(PeerStateReceivedArgs args)
    {
        if (args.State == PeerState.Stable)
        {
            isSessionStable = true;

            #if UNITY_EDITOR
                isSessionStable = true;
            #endif
        }
    }

    private void Networking_Connected(ConnectedArgs args)
    {
        isHost = args.IsHost;
        host = args.Host;
        player = args.Self;
    }

    private void OnDestroy()
    {
        if (aRNetworking == null)
        {
            return;
        }
        aRNetworking.Networking.Connected -= Networking_Connected;
        aRNetworking.PeerStateReceived -= ARNetworking_PeerStateReceived;
        aRNetworking.PeerPoseReceived -= ARNetworking_PeerPoseReceived;
    }
}

```

Tabla 5: Script NetARSessionManager el cual se encarga de la conexión de los jugadores y control del Host

46. Una vez hecho referenciaremos un objeto de nuestra jerarquía a nuestro script como se muestra en la Figura 56 el otro objeto que nos muestra lo referenciaremos más adelante.

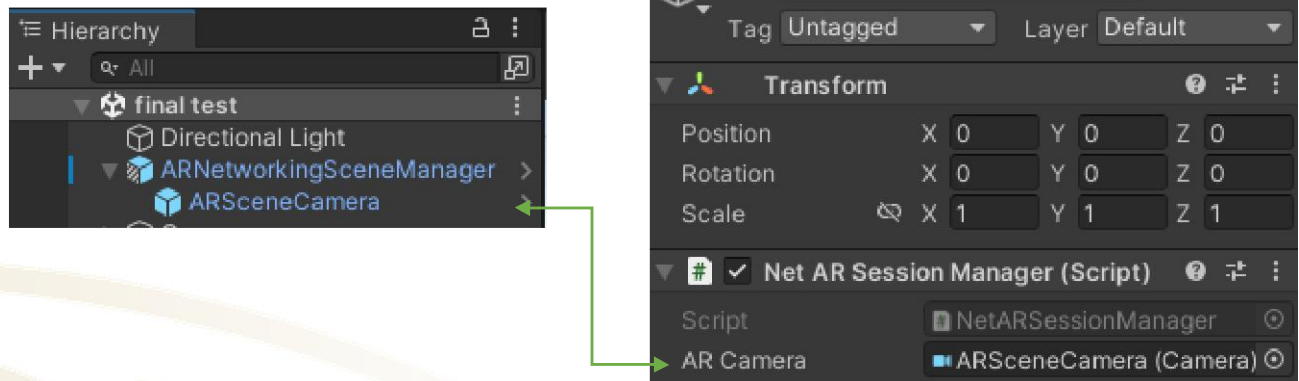


Figura 56: Se referencian los objetos para poder usarlos dentro de los scripts y así llamarlos desde otros scripts

47. Ahora necesitamos crear una carpeta que lleve con nombre Prefabs dentro de nuestra carpeta Assets ya que mas adelante la usaremos para nuestros prefabs.
48. Ahora agregaremos el detector de planos que nos ayudara para hacer parte de la detección ya que eso nos evitara generar un proceso pesado para los dispositivos móviles. En nuestra jerarquía iremos a ARNetworkingSceneManager y en el inspector añadiremos un componente que lleva por nombre AR Plane Manager.
49. Después dentro de Assets buscaremos un objeto que lleva por nombre PlanePrefab y como se muestra en la imagen referenciaremos este a nuestro AR Plane Manager. Después dentro de este mismo seleccionemos Horizontal en el Detected Plane Type

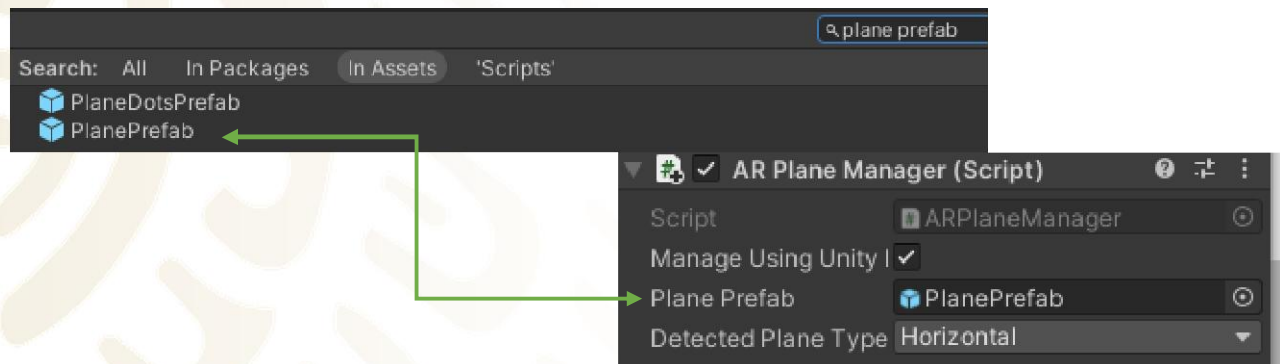


Figura 57: Este archivo se referencia para poder reducir el consumo de procesamiento de los dispositivos móviles al hacer la detección de planos

50. Ahora dentro de nuestra carpeta scripts crearemos dos nuevos uno llevara por nombre ARInteractor y el segundo llevara por nombre NetMessenger.
51. Primero empezaremos con ARInteractor. La función de este código es realizar la interacción con el entorno y el objeto todo esto usando una función llamada HitTest, esta función nos permite saber qué lugar y posición nuestro usuario realizo un toque y con esto sabremos en que posición se está moviendo el objeto. Para este script usaremos el código siguiente:

```

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

using Niantic.ARDKit.AR;
using Niantic.ARDKit.AR.HitTest;
using Niantic.ARDKit.AR.Networking;
using Niantic.ARDKit.Utilities;
using Niantic.ARDKit.Utilities.Input.Legacy;
using Niantic.ARDKit.AR.Networking.ARNetworkingEventArgs;
using Niantic.ARDKit.Networking.MultipeerNetworkingEventArgs;
using UnityEngine.UIElements;

public class ARInteractor : MonoBehaviour
{
    [SerializeField] private GameObject aObjectPrefab;
    private ARObjController aObjectController;
    public ARObjController ARObjController { get =>
aObjectController; }
    private IARSession aSession;
    // Start is called before the first frame update
    void Start()
    {
        ARNetworkingFactory.ARNeworkingInitialized +=
OnARNeworkingInitialized;
    }

    private void OnARNeworkingInitialized(AnyARNeworkingInitializedArgs
args)
    {
        aSession = args.ARNeworking.ARSession;
        args.ARNeworking.Networking.Connected += Networking_Connected;
    }

    private void Networking_Connected(ConnectedArgs args)
    {
        aObjectController =
Instantiate(aObjectPrefab).GetComponent<ARObjController>();
    }

    // Update is called once per frame
    void Update()
    {
        if (!GameManager.Instance.SessionManager.IsSessionStable)
        {
            return;
        }
        if (PlatformAgnosticInput.touchCount <= 0)
        {
            return;
        }

        Touch touch = PlatformAgnosticInput.GetTouch(0);
    }
}

```

```

        if (touch.phase == TouchPhase.Began)
        {
            ProcessTouch(touch);
        }
    }

    private void ProcessTouch(Touch touch)
    {
        var currentFrame = aRSession.CurrentFrame;
        if (currentFrame == null)
        {
            return;
        }

        var hitResult =
currentFrame.HitTest(GameManager.Instance.SessionManager.ARCamera.pixelWidth,
                    GameManager.Instance.SessionManager.ARCamera.pixelHeight,
                    touch.position,
                    ARHitTestResultType.ExistingPlaneUsingExtent);

        if (hitResult.Count <= 0)
        {
            return;
        }

        Vector3 hitPosition = hitResult[0].WorldTransform.ToPosition();

        MoveARObjectCrossNetworking(hitPosition);
    }

    private void MoveARObjectCrossNetworking(Vector3 Position)
    {
        if (GameManager.Instance.SessionManager.IsHost)
        {
            aRObjecController.SetDestination(Position);
        }
        else
        {
            GameManager.Instance.Messenger.PositionRequestHost(Position);
        }

        GameManager.Instance.Messenger.BroadcastAnimation(1);
    }

```

Tabla 6: Script ARInteractor el cual se encarga de la interacción de los jugadores y el host con el entorno

52. Ahora para el segundo script usaremos el código que está a continuación, este código nos ayudará a mantener coordinada la sesión ya que estará mandando mensajes a los diversos usuarios dentro del entorno, así como también al host ya que para este último el recibir datos de un usuario nos permitirá saber posición y movimiento de un usuario.

```

using System.Collections.Generic;
using UnityEngine;

using Niantic.ARDK.Networking;
using Niantic.ARDK.Utilities.BinarySerialization;
using Niantic.ARDK.Utilities.BinarySerialization.ItemSerializers;
using System.IO;
using System.Linq;

public class NetMessenger : MonoBehaviour
{
    public NetMessenger(IMultipeerNetworking networking)
    {
        this.networking = networking;
    }

    private IMultipeerNetworking networking;

    private readonly MemoryStream memoryStream = new MemoryStream(24);

    public enum TypeMessage : uint
    {
        PositionMgs,
        PositionRequest,
        RotationMgs,
        AnimationMgs
    }

    public void PositionRequestHost(Vector3 vector3)
    {
        networking.SendDataToPeer((uint)TypeMessage.PositionRequest,
            SerializeVector3(vector3),
            GameManager.Instance.SessionManager.Host,
            TransportType.ReliableUnordered);
    }

    public void BroadcastPositio(Vector3 vector3)
    {
        networking.BroadcastData((uint)TypeMessage.PositionMgs,
            SerializeVector3(vector3), TransportType.UnreliableUnordered);
    }

    public void BroadcastEulerAngles(Vector3 vector3)
    {
        networking.BroadcastData((uint)TypeMessage.RotationMgs,
            SerializeVector3(vector3), TransportType.UnreliableUnordered);
    }

    public void BroadcastAnimation(float animation)
    {
        List<IPeer> peers = networking.OtherPeers.ToList();
        networking.SendDataToPeers((uint)TypeMessage.AnimationMgs,
            SerializeFloat(animation), peers, TransportType.ReliableUnordered, true);
    }
}

```

```

public byte[] SerializeVector3(Vector3 vector3)
{
    memoryStream.Position = 0;
    memoryStream.SetLength(0);

    using var binarySerializer = new BinarySerializer(memoryStream);
    Vector3Serializer.Instance.Serialize(binarySerializer, vector3);
    return memoryStream.ToArray();
}

public Vector3 DeserializeVector3(byte[] data)
{
    using var readingStream = new MemoryStream(data);
    using var binaryDeserealizer = new
BinaryDeserializer(readingStream);
    return Vector3Serializer.Instance.Deserialize(binaryDeserealizer);
}

public byte[] SerializeFloat(float floatNum)
{
    memoryStream.Position = 0;
    memoryStream.SetLength(0);

    using var binarySerializer = new BinarySerializer(memoryStream);
    FloatSerializer.Instance.Serialize(binarySerializer, floatNum);
    return memoryStream.ToArray();
}

public float DeserializeFloat(byte[] data)
{
    using var readingStream = new MemoryStream(data);
    using var binaryDeserializer = new
BinaryDeserializer(readingStream);
    return FloatSerializer.Instance.Deserialize(binaryDeserializer);
}

```

Tabla 7: Script NetMessenger el cual nos ayuda a mandar mensajes de estado de cada jugador al Host así como el Host a los jugadores para mantenerlos al tanto de cada uno de las acciones que se realizan en el entorno

53. Una vez terminado vamos a regresar a unity para continuar de configurar algunos detalles de nuestro proyecto. Primero iremos a nuestra jerarquía y seleccionaremos GameManager, una vez que estemos en el objeto iremos a nuestro inspector y arrastraremos lo que se muestra a continuación:



54. Como se observa en la imagen de arriba, arrastramos el script y después el script GameManager ya que con esto podemos controlar la interacción y con esto podemos acceder a este mismo y modificar algunas propiedades dentro de este.
55. Habiendo hecho esto necesitamos un objeto para ubicar a nuestros jugadores, dentro de nuestra jerarquía crearemos un cubo a ruta para esto es la siguiente click derecho → 3D object → Cube.

56. Ahora como se muestra en la figura siguiente debemos resetear su transform y colocarle las escalas como se indica en la imagen.

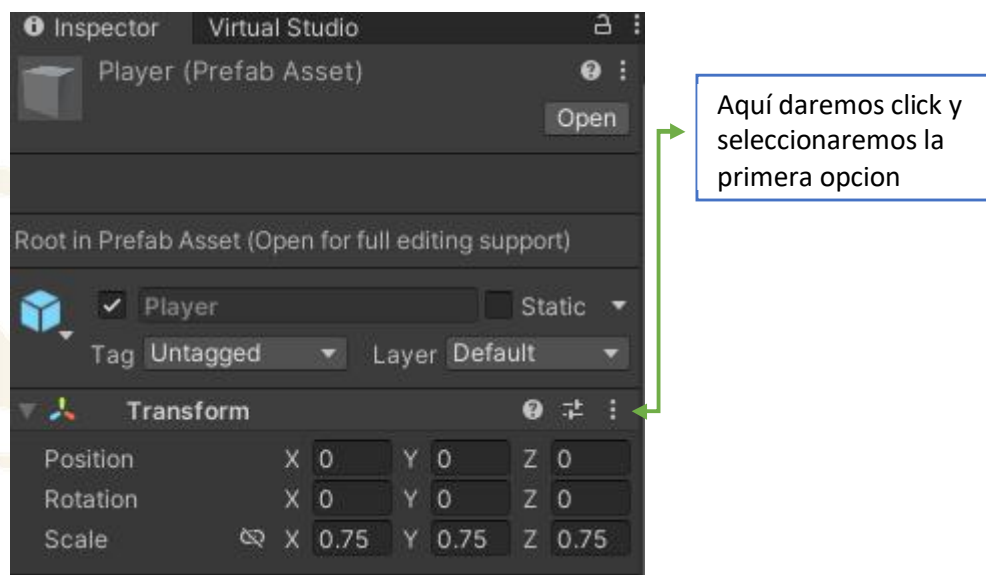


Figura 58. Con esto siempre resetearemos los datos de cualquier objeto que ingresemos a nuestro espacio ayudándonos a colocarlo en la posición que necesitamos

57. Una vez terminado esto volveremos al script GameManager para terminarlo y poder usar todas las nuevas funciones que hemos incluido hasta este momento, el código quedaría de la siguiente manera:

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

using Niantic.ARDK.AR.Networking;
using Niantic.ARDK.Networking;
using Niantic.ARDK.AR.Networking.ARNetworkingEventArgs;
using Niantic.ARDK.Networking.MultipeerNetworkingEventArgs;

public class GameManager : MonoBehaviour
{
    [SerializeField] private ARInteractor arInteractor;
    public static GameManager Instance;
    private NetARSessionManager sessionManager;
    public NetARSessionManager SessionManager { get => sessionManager; }

    private IMultipeerNetworking networking;
    private NetMessenger messenger;
    public NetMessenger Messenger { get => messenger; }
    private void Awake()
    {
        if (Instance != this && Instance != null)
        {
            Destroy(this);
        }
        else
        {

```

```

        Instance = this;
    }
}
// Start is called before the first frame update
void Start()
{
    sessionManager = GetComponent<NetARSessionManager>();
    ARNetworkingFactory.ARNetworkingInitialized +=
OnARNetworkingInitialized;
}

private void OnARNetworkingInitialized (AnyARNetworkingInitializedArgs
args)
{
    networking = args.ARNetworking.Networking;
    messenger = new NetMessenger(networking);
    networking.PeerDataReceived += Networking_PeerDataReceived;
}

private void Networking_PeerDataReceived(PeerDataReceivedArgs args)
{
    byte[] data = args.CopyData();
    switch ((NetMessenger.TypeMessage) args.Tag)
    {
        case NetMessenger.TypeMessage.PositionMgs:
aRInteractor.ARObjecController.SetPosition(messenger.DeserializeVector3(dat
a));
        break;
        case NetMessenger.TypeMessage.PositionRequest:
aRInteractor.ARObjecController.SetDestination(messenger.DeserializeVector3(
data));
        break;
        case NetMessenger.TypeMessage.RotationMgs:
aRInteractor.ARObjecController.SetRotation(messenger.DeserializeVector3(dat
a));
        break;
        case NetMessenger.TypeMessage.AnimationMgs:
aRInteractor.ARObjecController.PlayAnimation(messenger.DeserializeFloat(dat
a));
        break;
        default:
        break;
    }
}
}

```

Tabla 8: Parte final del Script GameManager el cual nos ayuda a tener control del Backend y Frontend de la aplicación sin acceder a uno por uno de los Scripts.

58. Terminado esto vamos a importar un package más para nuestro proyecto y es nuestro modelo 3D creado y diseñado para este proyecto así que iremos al siguiente link:
https://drive.google.com/file/d/1ipviUI-8ZUaDnVFUzTH-x9KBLkAh9R6M/view?usp=drive_link
59. Una vez descargado este package vamos a importarlo como importamos en la Figura 42 siguiendo estos mismos pasos.
60. Una vez hecho esto vamos a ir a nuestra carpeta Scripts y creamos el controlador de nuestro objeto así que seguiremos la ruta click derecho → C# script como lo hemos hecho con anterioridad y lo llamaremos ARObjController.
61. Una vez creado iremos al script y usando el código que se presenta a continuación haremos que nuestro objeto realice una acción, este nos indicara cuando el objeto este en movimiento así como la animación que se esté presentando en ese momento.

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

public class ARObjController : MonoBehaviour
{
    [SerializeField] private float speed = 0.75f;
    [SerializeField] private Animator objectAnimator;
    private Vector3 targetPosition;
    private bool isMoving;

    // Update is called once per frame
    void Update ()
    {
        if (!GameManager.Instance.SessionManager.IsHost)
        {
            return;
        }
        if (!isMoving)
        {
            return;
        }
        MoveARObject();
        //Broadcast Position and Rotation;
        GameManager.Instance.Messenger.BroadcastPosition(transform.position);
        GameManager.Instance.Messenger.BroadcastEulerAngles(transform.rotation.eulerAngles);
    }

    private void MoveARObject()
    {
        float step = speed * Time.deltaTime;
        Vector3 direction = (targetPosition -
transform.position).normalized;
```

```

        transform.Translate(Vector3.forward * step, Space.Self);
        transform.rotation = Quaternion.RotateTowards(transform.rotation,
Quaternion.LookRotation(direction, Vector3.up), 8.0f);

        if (Vector3.Distance(transform.position, targetPosition) < 0.1f)
        {
            GameManager.Instance.Messenger.BroadcastAnimation(0);
            isMoving = false;
        }
    }

    public void SetDestination(Vector3 position)
    {
        targetPosition = position;
        isMoving = true;
    }

    public void SetPosition(Vector3 position)
    {
        transform.position = position;
    }

    public void SetRotation(Vector3 eulerAngles)
    {
        transform.rotation = Quaternion.Euler(eulerAngles);
    }

    public void PlayAnimation(float animation)
    {
        objectAnimator.SetFloat("Walk", animation);
    }
}

```

Tabla 9: Scripts ARObjController el cual se encarga del movimiento y control de la acciones que puede realizar el modelo, el cual tendrá un duplicado para cada uno de los jugadores.

62. Vamos a terminar los últimos detalles de nuestro proyecto primero el cubo que creamos con anterioridad le cambiaremos el nombre a player y lo moveremos a nuestra carpeta prefabs.
63. Ahora iremos a nuestra carpeta prefabs e iremos al objeto que lleva por nombre  ARObjectHolder Y como se muestra en la imagen siguiente agregaremos nuestro script ARObjController usando el botón  botón.
64. Una vez hecho esto y agregado el script vamos a referenciar el objeto PlaterArmature como se muestra en la Figura 59 esto es para poder tener todo listo y referenciado el script de control de movimiento con el modelo que realizara las acciones.

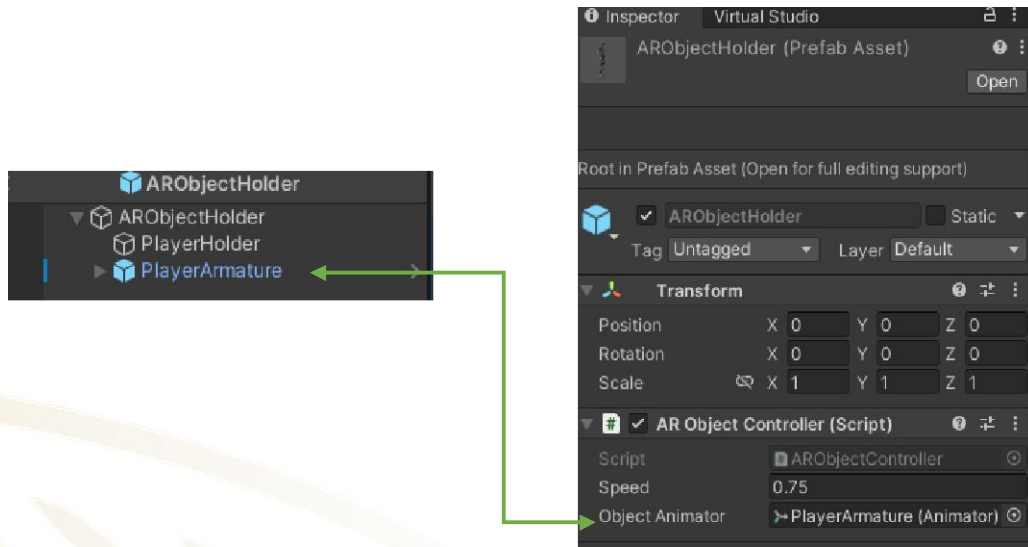


Figura 59: Necesitamos referenciar estos objetos para poder tener el control del modelo y su animación para que se vea real

65. Regresamos a nuestra jerarquía principal y vamos a nuestro GameManager y como se indica en la Figura 60 relacionaremos nuestro ARObjectController con nuestro ARInteractor.

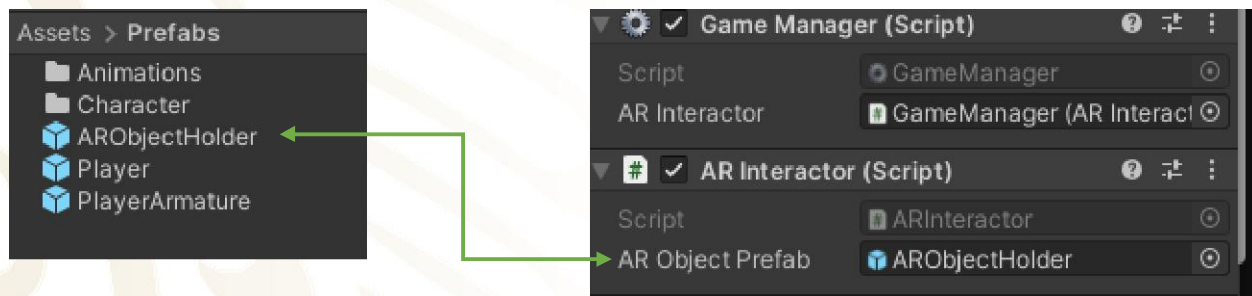
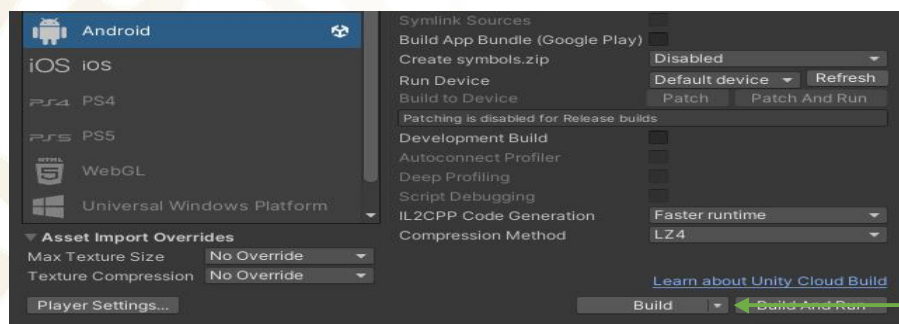


Figura 60: Se referencia este objeto al script ya que debemos de tener el control total del objeto ya que de acuerdo a la interacción será la posición en donde deba de moverse.

66. Una vez hecho todo estamos listos para generar nuestra aplicación para esto seguiremos la siguiente ruta File → BuildSettings y como se muestra en la imagen siguiente y daremos click en donde se indica.



Daremos click aquí seleccionamos la ruta de guardado y listo empezara a generar el APK

Apéndice B: BIBLIOGRAFIA

- [1] L. Huan, D. Shufang y Z. Qian, «"The Application of Virtual Reality Technology in the Teaching System of Tourism Training",» de *Electrical and Information Technology (IEIT)*, China, 2023.
- [2] T. Hou y j. Liaoning, «Virtual Tourism Simulation System Based on VR Technology,» *Journal of physics*, vol. 1881, nº 032084, p. 7, 2021.
- [3] J. Malca Y, B. Carrasco C., V. Guaman P. y J. Delgado A, «ARTOUR: Realidad Aumentada aplicada al Turismo - Caso de Estudio en Riobamba, Ecuador,» de *International Conference on Information Systems and Computer Science (INCISCOS)*, Riobamba-Ecuador, 2019.
- [4] J. An, X. Cheng, Q. Wang y H. Chen, «attractions roaming system based on Unity 3D,» *journal of Physics*, vol. 1237, nº 042049, p. 6, 2019.
- [5] D. E. kurianawan, A. Dzikri, M. Suriya, Y. Rokhayati y A. Najmurrokhman, «Object Visualization Using Maps Marker Based On,» de *IEEE*, Athens, Greece, 2018.
- [6] B. Batuwanwhudawa y K. Jayasena, de *International Conference on Advancements*, Sri Lanka, 2020.
- [7] X. Liu, Y. Ho-Sohn y D. Won Park, «Application Development with Augmented Reality Technique using Unity,» *International Journal of Applied Engineering Research*, vol. 13, nº 21, pp. 115068-15071, 2018.
- [8] J. An, Q. Wang, C. Xinrong y H. Chen, «Implementation of attractions roaming system based on Unity,» de *Conferecen Series: Journal of physics*, Beijing China, 2019.
- [9] D. Karia, A. Shak, S. Vora, P. Mane y D. B. Patel, «Augmented Reality Game Development using Unity3D,» *Journal of Emerging Technologies and Innovative Research*, vol. 6, nº 4, pp. 570-573, 2019.
- [10] S. Lae Kim, H. Suk Jung, T. Laine H. y J. Westlin, «Using Unity 3D to Facilitate Mobile Augmented,» de *IEEE World Forum on Internet of Things*, Seoul, 2014.
- [11] p. Bhat, P. Krutik, P. Harshil y R. Uday, «Tourism Application Using Augmented Reality,» de *Proceedings of the 3rd International Conference on Advances in Science & Technology (ICAST) 2020*, mumbai, 2020.
- [12] O. Chigozie Jude y C. Ukekwe, «Tourism and Virtual Reality (VR) in Developing Nations,» *African Journal of Hospitality, Tourism and Leisure*, vol. 9, nº 2, p. 1000, 2020.
- [13] R. González Reverté,, «oikonomics,» Universitat Oberta de Catalunya, 4 11 2016. [En línea]. Available: <http://oikonomics.uoc.edu> . [Último acceso: 20 04 2023].
- [14] V. Castello, «Desafíos y oportunidades para el turismo en el marco de la,» *Cuadernos de Política Exterior Argentina (Nueva Época)*, vol. 131, nº 2, pp. 115-118, 2020.
- [15] C. Yuan Li y B. Tang, «Research on The Application of AR Technology Based on,» de *Conference series: Journal of psisycs*, china, 2019.
- [16] C. d. T. d. Singapur., «Visit Singapore,» Consejo de turismo de singapur, 15 02 2018. [En línea]. Available: <https://www.visitsingapore.com/en/>. [Último acceso: 20 04 2023].
- [17] T. A. T. P. ABTA, «Thomascook,» Thomas Cook Tourism(UK) Company Ltd, 23 05 2020. [En línea]. Available: <https://www.thomascook.com/mobile-app>. [Último acceso: 21 04 2023].

- [18] S. group, «Smartech group,» smartech, 8 04 2017. [En línea]. Available: <https://smartechgroup.es/artour/barcelona/>. [Último acceso: 23 04 2023].
- [19] N. estuio, «bbc.co.uk,» nexus estudio, 1 12 2020. [En línea]. Available: <https://www.bbc.co.uk/taster/pilots/civilisations-ar>. [Último acceso: 3 05 2023].
- [20] u. immserse, The immersive economy in the uk 2019, UK: Catapult Digital, 2019.
- [21] J. Beck, M. Rainoldi y R. Egger, «Virtual reality in tourism: a state-of-the-art review,» *Tourism development & planning*, vol. 56, nº 2, p. 78, 2019.
- [22] C. YuanLi y B. Tang, «Research on The Application of AR Technology Based on,» *Conference series: journal of physics*, vol. 1168, nº 5, p. 1500, 2019.
- [23] W. InfoTech, «pitchbook,» WT InfoTech, 15 07 2012. [En línea]. Available: <https://pitchbook.com/profiles/company/164281-78#overview>. [Último acceso: 23 04 2023].