



UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería  
Informática**

**Asistente remoto en realidad  
aumentada y mixta con  
HoloLens 2  
Memoria**



Presentado por Miguel Martín Valdivielso  
en la Universidad de Burgos — 14 de junio de  
2021

Tutores académicos: Ángel Arroyo Puente y  
Nuño Basurto Hornillos

Tutor empresarial (ITCL): Alejandro  
Langarica Aparicio







UNIVERSIDAD DE BURGOS  
ESCUELA POLITÉCNICA SUPERIOR  
Grado en Ingeniería Informática



D. Ángel Arroyo Puente, profesor del departamento de Ingeniería Informática, área de Lenguajes y Sistemas Informáticos.

Expone:

Que el alumno D. Miguel Martín Valdivielso, con DNI 71302203H, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado Asistente remoto en realidad aumentada y mixta con HoloLens 2.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 14 de junio de 2021

Vº. Bº. del Tutor:

Vº. Bº. del Cotutor:

Vº. Bº. del Tutor:

D. Ángel Arroyo Puente

D. Nuño Basurto  
Hornillos

D. Alejandro Langarica  
Aparicio





## **Resumen**

En este documento se recoge la documentación relativa a la creación de un prototipo de videoconferencia con asistencia remota para el dispositivo de realidad mixta Microsoft HoloLens 2, orientado a la inspección, reparación y mantenimiento remoto en un entorno colectivo. Pudiendo conectar así a personas con diferentes roles en distintas ubicaciones.

Un lado de la aplicación contará con un técnico cualificado que, a través del vídeo recibido desde las gafas en tiempo real, podrá guiar, dar órdenes y señalar objetos desde un ordenador para que el operario con las HoloLens 2 vea los objetos señalados en realidad aumentada.

Este proyecto ha sido desarrollado con el software de Unity complementándolo con herramientas necesarias para el desarrollo como Mixed Reality Toolkit, WebRTC y Node.js. Todo lo relacionado con la investigación y la implementación de código ha sido realizado haciendo uso del material provisto por las instalaciones de ITCL.

Cabe destacar que este proyecto al ser un prototipo, es la base de uno de dimensiones mucho mayores del cual queda un amplio margen de mejora y evolución.

## **Descriptores**

HoloLens 2, Unity, realidad aumentada, realidad mixta, asistencia remota, videoconferencia.

## **Abstract**

This document collect the documentation related to the creation of a remote assist video conference prototype for the Microsoft mixed reality device called HoloLens 2, oriented at remote inspection, repair and maintenance in a collective environment connecting people with different roles in different locations.

One side of the application will have a qualified technician who, through the video recieved from the glasses in real time, will be able to guide, give orders and point to objects from the computer so the operator with the HoloLens 2 can see the pointed objects in augmented reality.

All this project has been developed in Unity complementing it with tools necessary for development such as Mixed Reality Toolkit, WebRTC and Node.js. Everything related to research and code implementation has been done using the material provided by ITCL facilities.

The project is a prototype that is the base of a larger one of which there is a wide range of improvement and evolution.

## **Keywords**

HoloLens 2, Unity, augmented reality, mixed reality, remote assist, video conference.



---

# Índice general

---

Índice general	III
Índice de figuras	V
Índice de tablas	VI
<b>Introducción</b>	<b>1</b>
1.1. Estructura de la memoria . . . . .	2
1.2. Anexos . . . . .	2
1.3. Materiales Adjuntos . . . . .	3
<b>Objetivos del proyecto</b>	<b>5</b>
2.1. Objetivos principales . . . . .	5
2.2. Objetivos secundarios . . . . .	6
2.3. Objetivos técnicos . . . . .	6
2.4. Objetivos personales . . . . .	6
<b>Conceptos teóricos</b>	<b>9</b>
3.1. HoloLens 2 . . . . .	9
3.2. Unity . . . . .	13
<b>Técnicas y herramientas</b>	<b>19</b>
4.1. Metodologías . . . . .	19
4.2. Control de versiones . . . . .	19
4.3. Repositorio . . . . .	20
4.4. Gestión de proyecto . . . . .	21
4.5. Unity . . . . .	21

4.6. <i>Plugins</i> Unity . . . . .	22
4.7. Entorno de desarrollo integrado (IDE) . . . . .	23
4.8. Otros . . . . .	23
<b>Aspectos relevantes del desarrollo del proyecto</b>	<b>25</b>
5.1. Inicio del proyecto . . . . .	25
5.2. Comunicación por vídeo y audio . . . . .	26
5.3. Asistencia Remota . . . . .	26
<b>Trabajos relacionados</b>	<b>29</b>
6.1. Dynamics 365   Remote Assist . . . . .	29
<b>Conclusiones y Líneas de trabajo futuras</b>	<b>31</b>
7.1. Conclusiones . . . . .	31
7.2. Líneas de trabajo futuras . . . . .	32
<b>Bibliografía</b>	<b>33</b>

---

# Índice de figuras

---

3.1.	Se muestra la comparación entre unas manos físicas y el resultado virtual del tracking. [13]	12
3.2.	Se muestran 3 ejemplos de asignación espacial variando la cantidad de triángulos que los conforman. [24]	13
5.3.	Se muestra el plano correspondiente a la resolución de las HoloLens 2 (verde) y el objeto (rosa) que se mueve sobre este para obtener la dirección del <i>raycast</i>	28

---

# Índice de tablas

---

6.1. Comparativa de las características de las aplicaciones. . . . .	30
--	----

---

# Introducción

---

A día de hoy uno de los mayores gastos a los que se puede enfrentar una empresa, y que suponen una gran pérdida de tiempo y dinero, se encuentra en el mantenimiento de las fábricas. Las que más mantenimiento requieren son aquellas con una cadena de producción, o que poseen diferentes máquinas y/o herramientas de la cuales se hacen un uso muy intensivo. Actualmente hay muy pocas opciones para llegar a la solución de este problema de alto mantenimiento económico, lo más común es tener a uno o varios operarios cualificados en cada una de las fábricas de la empresa para resolver los contratiempos en el menor tiempo posible. Pero esto tiene un inconveniente, el gran desembolso económico que supone contar con varios operarios por fábrica, lo cual no es asumible. El cual muchas empresas pueden no ser capaces de afrontar.

La solución propuesta por este TFG realizado en conjunto con el Instituto Tecnológico de Castilla y León (ITCL) [5] es crear una aplicación para las HoloLens 2 [11], dispositivo de realidad mixta desarrollado por Microsoft orientado a aumentar la productividad de las empresas, en la que un técnico pueda guiar y dar instrucciones a cualquier usuario que se encuentre disponible en la fábrica, aunque tenga poca o nula formación sobre mantenimiento, para así solucionar cualquier imprevisto que surja. Gracias a esto se puede conseguir reducir el número de técnicos por fábrica, lo cual supondría reducir considerablemente el gasto económico en términos de personal. Además, una gran ventaja que ofrece este sistema es que puede ser escalado a nivel global, significando esto que los técnicos puedan ofrecer sus conocimientos a cualquier parte del planeta donde se disponga de una conexión a internet.

Una vez completado el prototipo de la aplicación se esperan implementar funcionalidades como compartir documentos en realidad aumentada, reconocimiento de objetos, interacción con piezas o maquinaria en 3D.

## 1.1. Estructura de la memoria

La memoria sigue la siguiente estructura:

**Introducción:** breve descripción del problema a resolver y la solución planteada. Estructura de la memoria y los Materiales adjuntos.

**Objetivos del proyecto:** requisitos necesarios para cumplir los objetivos del proyecto.

**Conceptos teóricos:** explicación de los conceptos teóricos usados a lo largo de la memoria.

**Técnicas y herramientas:** metodología y herramientas utilizadas durante el desarrollo.

**Aspectos relevantes del desarrollo:** explicación de los aspectos más relevantes del desarrollo.

**Trabajos relacionados** pequeño resumen comentado de los trabajos y proyectos ya realizados relacionados con este campo.

**Conclusiones y líneas de trabajo futuras** conclusiones obtenidas tras la finalización del proyecto y posibilidades de expansión y mejora.

## 1.2. Anexos

**Plan de proyecto software:** planificación de la estructura del proyecto y la viabilidad económica.

**Especificación de requisitos:** requisitos marcados por los objetivos del proyecto.

**Especificación de diseño** diseño del software y sus diagramas correspondientes.

**Manual del programador:** recoge todos los aspectos necesarios para trabajar con los ficheros del proyecto.

**Manual del usuario:** guía del usuario para la aplicación.

## 1.3. Materiales Adjuntos

Los materiales adjuntos a este TFG son:

**Documentación:** Documentos en formato digital de la memoria y los anexos.

***Builds:*** Carpeta con los ejecutables de las aplicaciones.

**Fuentes:** Directorio con el proyecto de Unity.

**Fuentes de documentación:** Directorio del proyecto de L<sup>A</sup>T<sub>E</sub>X.

**Vídeos:** Vídeos del producto final.





---

# Objetivos del proyecto

---

El objetivo principal de este proyecto es el desarrollo de una aplicación de videoconferencia con asistencia remota, que establezca una comunicación entre un técnico en un ordenador y un usuario cualquiera que esté haciendo uso del dispositivo de realidad mixta HoloLens 2 de Microsoft.

Desde ITCL se marcaron unos objetivos mínimos del proyecto, los cuales eran completamente imprescindibles alcanzar para obtener como resultado una aplicación que cumpliera con las necesidades básicas de una vídeollamada asistida.

A continuación, establecimos unos objetivos secundarios de menor relevancia que serán efectuados si el tiempo restante del proyecto lo permite. El propósito de estos es añadir funcionalidades al resultado final de la aplicación una vez cumplidos los objetivos principales.

## 2.1. Objetivos principales

1. Desarrollar en Unity [28] una aplicación para HoloLens 2 y ordenador que permita realizar una videoconferencia con asistencia remota.
2. Especializarme en el uso de la plataforma de desarrollo Unity.
3. Conectar un ordenador y las HoloLens 2 mediante una aplicación en cada dispositivo, realizando las comunicaciones necesarias a través de un servidor en red local.
4. A través de la conexión ya establecida compartir audio entre los dos dispositivos para poder entablar una conversación.

5. Enviar en tiempo real un vídeo que muestre lo que se ve desde la perspectiva de la persona que lleva puesto el dispositivo de realidad mixta.
6. A partir de ese vídeo recibido poder hacer clic en la pantalla del ordenador y que aparezcan indicadores en realidad aumentada al usuario que esté usando las HoloLens 2.

## **2.2. Objetivos secundarios**

1. Implementar funcionalidades de una vídeollamada como: silenciar el micrófono, llamar, colgar, desactivar la webcam.
2. Diseñar una interfaz gráfica para ambas aplicaciones.
3. Realizar la conexión de las aplicaciones a través de un servidor accesible desde cualquier parte, no solo en red local.
4. Implementar un sistema de reconocimiento de objetos.
5. Enviar un documento en formato .pdf desde el ordenador para generarlo en realidad aumentada, haciendo así uso de él.
6. Poder cargar modelos 3D en las HoloLens 2 para poder visualizarlos en realidad aumentada.

## **2.3. Objetivos técnicos**

1. Conectar ambas aplicaciones usando un servidor en red local mediante la herramienta Node.js [21].
2. Uso de repositorio GitHub [4] junto con la herramienta ZenHub [39] para llevar a cabo la administración de tareas del proyecto.
3. Documentar correctamente este prototipo para que sirva como manual para futuros desarrolladores que se entren en el proyecto.

## **2.4. Objetivos personales**

1. Aprender a desarrollar aplicaciones en Unity [28] para las HoloLens 2.

2. Aprender a desarrollar aplicaciones para HoloLens 2.
3. Adentrarme en el campo de la realidad aumentada y mixta.
4. Utilizar todos los conocimientos adquiridos en la carrera.
5. Acostumbrarme al uso de metodologías ágiles de software.
6. Generación de una documentación mediante la herramienta  $\text{\LaTeX}$ .



---

# Conceptos teóricos

---

En este apartado se explicarán diferentes conceptos del proyecto relacionados con el desarrollo en Unity y con el dispositivo de realidad mixta HoloLens 2.

El desarrollo de aplicaciones en un motor gráfico como Unity es ligeramente diferente a la clásica programación orientada a objetos, por ello se cubrirán los aspectos básicos que son desconocidos para aquellas personas que nunca hayan trabajado con este tipo de entornos de desarrollo.

## 3.1. HoloLens 2

En este apartado de conceptos teóricos veremos toda la parte relacionada con las HoloLens 2.

### *Augmented Reality (AR)*

La realidad aumentada [34] permite la superposición de información y objetos virtuales sobre la información física del entorno que te rodea. Todo esto es posible a través de un dispositivo tecnológico que disponga de una cámara, para obtener los datos del espacio real, y de una pantalla para mostrar el resultado de la mezcla de los datos reales y virtuales.

Por lo general los teléfonos móviles son los más usados para la realidad aumentada, pero en este caso se va a hacer uso de un dispositivo especialmente diseñado para esto, las HoloLens 2, ya que estas ofrecen unas funcionalidades extra que no poseen los móviles.

## ***Virtual Reality (VR)***

La realidad virtual [38] consiste en la simulación virtual de un entorno de apariencia real donde puedas interactuar con este y los objetos que se encuentren dentro. El entorno generado tiene que dar al usuario la sensación de inmersión para una experiencia más realista.

Para experimentar esta simulación es necesario hacer uso de unas gafas o cascos de realidad virtual. Las gafas pueden ir complementadas de diferentes accesorios como mandos, guantes y trajes espaciales para *tracking* y estaciones base para detectar el espacio de la habitación donde realizas la inmersión.

## ***Mixed reality (XR)***

La realidad mixta [37, 18] es la combinación de realidad aumentada y virtual. En ella vamos a poder visualizar e interactuar tanto con el espacio y objetos reales, como con todos los elementos virtuales.

Este proyecto combina ambas realidades al señalar objetos del entorno con objetos virtuales en forma de flechas. Además, se puede interactuar con la interfaz virtual de la videoconferencia, posicionándola de la forma más cómoda en cualquier punto de la sala donde te encuentres.

## **Gafas inteligentes XR HoloLens 2**

Dispositivo de realidad mixta con forma de gafas inteligentes desarrolladas y fabricadas por Microsoft. Fueron lanzadas el 7 de noviembre de 2019 como sucesoras de las primeras HoloLens, siendo sus mejoras más relevantes el aumento del FOV 3.1 de las lentes de las gafas, mayor capacidad de procesamiento, funcionalidades de *hand tracking* y *eye tracking* eliminando así el mando que se usaba anteriormente para controlarlas.

Gracias a estas gafas podemos visualizar hologramas en realidad aumentada posicionados sobre el espacio que nos rodea, y así interactuar con ellos tocándolos, agarrándolos, girándolos y realizar todo tipo de transformaciones que podamos necesitar para hacer más cómoda la experiencia de la realidad mixta.

## **FOV**

El *field of view* [26] en español “campo de visión” es la extensión del entorno que podemos observar en un momento de tiempo dado. En este

caso vamos a tener dos FOVs. El virtual, que viene limitado por el tamaño y la resolución de las pantallas de las HoloLens 2. Y el real, se corresponde campo de visión que nos ofrecen nuestros ojos, y este variará dependiendo de la persona.

Contaremos con 52 grados de visión horizontal para observar el entorno virtual y unos 180 grados aproximadamente para observar el entorno real. Una diferencia bastante grande que generara dificultades para observar ambos entornos a la vez.

### ***Tracking***

En español “seguimiento” [17, 16] consiste en obtener la ubicación exacta de objetos en tiempo real a través del uso de diferentes sensores y/o cámaras complementándolo con algoritmos de inteligencia artificial. Las HoloLens 2 puede realizar el seguimiento tanto de manos como de ojos.

Toda la información que necesitan procesar para realizar el *tracking* proviene de:

- **Seguimiento de la cabeza:** 4 cámaras de luz visible.
- **Seguimiento de los ojos:** 2 cámaras de infrarrojos.
- **Profundidad:** 1 sensor de profundidad de tiempo de vuelo de 1 MP.
- **Unidad de medida inercial (IMU):** Acelerómetro, giroscopio, magnetómetro.
- **Cámara:** Vídeo 1080p30 de fotogramas 8-MP.

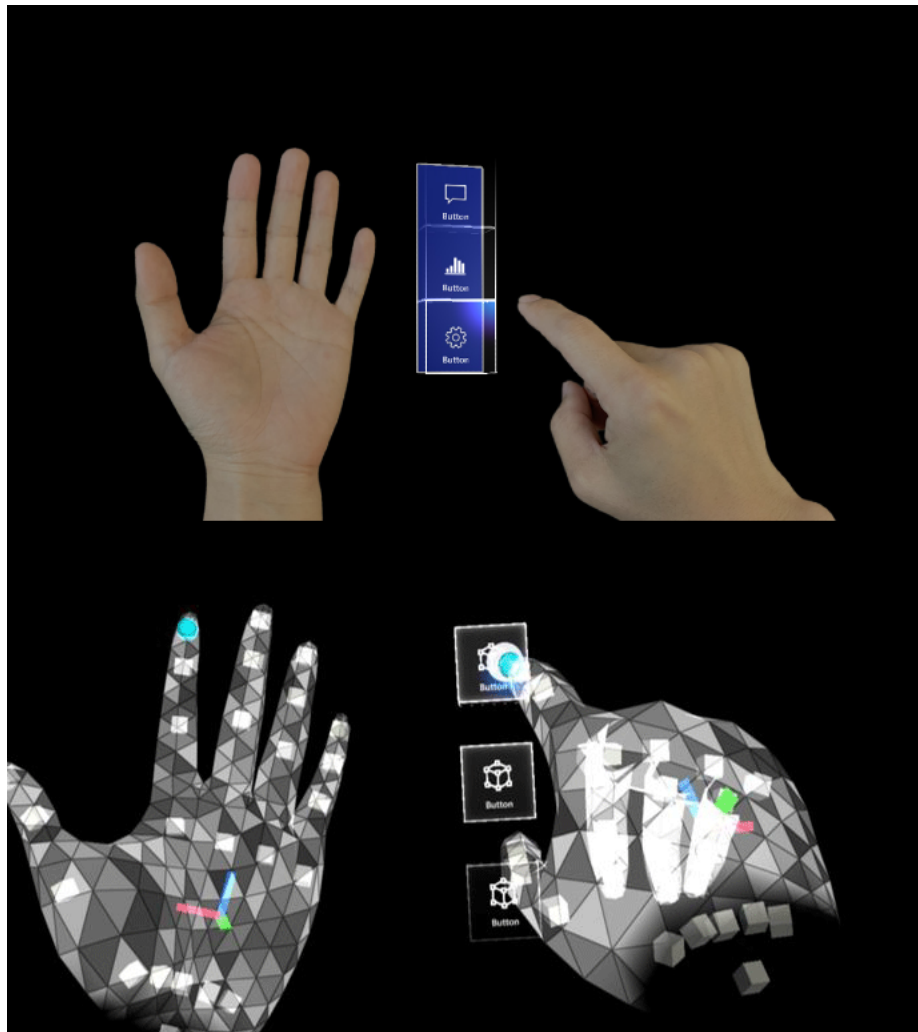


Figura 3.1: Se muestra la comparación entre unas manos físicas y el resultado virtual del tracking. [13]

## Asistencia remota

Esta nos permite conectar a través de internet dos dispositivos que estén en diferentes partes del mundo. Una vez realizada la conexión un técnico podrá ofrecer soluciones en tiempo real a través de la información (normalmente a través del envío de vídeo) que este reciba el usuario al otro lado de la conexión.



En esta aplicación se busca conectar un técnico en un ordenador y un usuario sin formación que lleve puestas las HoloLens 2, que ejecutará las órdenes que vaya recibiendo del experto.

### Asignación espacial

Es la capacidad que tiene un dispositivo de generar un mapa 3D completamente detallado del entorno que le rodea [7]. Con la asignación espacial somos capaces de anclar objetos virtuales al mundo real. Esto lo podemos conseguir gracias a diferentes sensores y cámaras que poseen las HoloLens 2 para detectar el espacio. Gracias a estas herramientas las gafas son capaces de comprender e interactuar con esa información del mundo real, para así dar mayor realismo y funcionalidad a los hologramas generados.

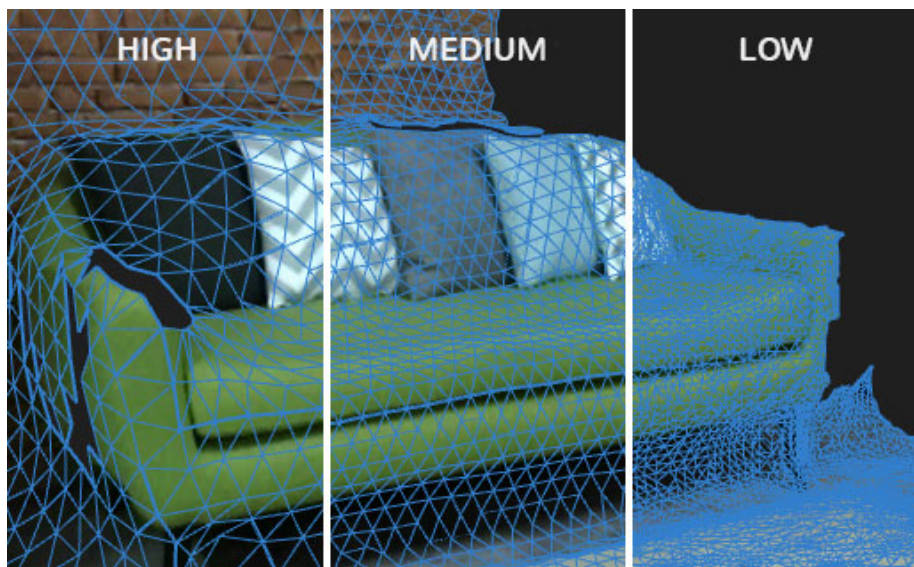


Figura 3.2: Se muestran 3 ejemplos de asignación espacial variando la cantidad de triángulos que los conforman. [24]

## 3.2. Unity

En este apartado se describen los conceptos relacionados con el motor gráfico Unity:

## Motor gráfico

Generalmente están orientados a la creación de videojuegos, pero también son utilizados para desarrollar simuladores u otro tipo de aplicaciones con gráficos 3D. Los motores gráficos [3] nos ofrecen un conjunto de herramientas que nos facilitan este desarrollo, donde las más importantes y utilizadas son:

- Renderización de gráficos 2D y 3D.
- Físicas de objetos y colisiones.
- Sistemas de partículas.
- Animación de objetos.
- Sonido y música.
- Programación de scripts.
- Inteligencia artificial.
- Conexión con la red.

## Unity

Este proyecto ha sido desarrollado en Unity [28]. Este es uno de los motores de videojuegos más importantes y utilizados en el desarrollo de videojuegos. Esto se debe a toda la cantidad de funcionalidades que ofrece, compatibilidad como todo tipo de programas del mismo sector y ser accesible para todo el mundo porque su licencia básica es completamente gratuita.

El lenguaje de programación que utiliza Unity es C#, un lenguaje orientado a objetos desarrollado por Microsoft. Dentro de este motor se le da un uso un tanto especial, ya que en vez de realizar una programación orientada a objetos, esta es orientada los componentes que forman los GameObjects.

## C#

Lenguaje de programación [9] desarrollado por Microsoft para ser parte de su *framework* .NET. Este deriva de C y C++ y es compatible con la mayoría de sistemas operativos más utilizados.

## *GameObject*

Es la clase más básica y principal del desarrollo en Unity. Esta representa a un objeto colocado en una escena con unas coordenadas correspondientes a espacio 2D o 3D. El cómo observemos este *GameObject* [27] dependerá de los componentes que estén asociados a este. Dependiendo de los componentes que le tenga, el objeto variará en forma, textura, color, comportamiento, funcionalidad, etc.

Todos los *GameObjects* tienen como mínimo un componente de transformación para poder existir. Este indica las coordenadas del objeto en el mundo creado.

Un *GameObject* podría ser definido como un contenedor de componentes.

## Componentes

Los componentes en Unity son aquellos que van a definir por completo todas las características de un objeto. Los más comunes que implementan las funcionalidades más importantes son:

- ***Transform.*** Posición, rotación y escala de un objeto.
- ***Camera.*** Componente que captura y muestra “el escenario” al usuario.
- ***Mesh.*** Da forma a la malla de los objetos y esta contiene toda la información de los vértices de los triángulos que componen la malla.
- ***Material.*** Tiene toda la información de las propiedades de un material, como textura, color, relieve... Sumándole el componente de la malla obtenemos la representación gráfica de los objetos.
- ***AudioSource.*** Genera sonidos en 3D con un objeto como origen.
- ***Physics.*** Conjunto de métodos que otorgan propiedades físicas a los objetos como masa, gravedad, velocidad, etc.
- ***Collider.*** Permite que un objeto colisione con otro.
- ***Script.*** Asocia un script con un objeto.

## ***Raycast***

Componente perteneciente al apartado de físicas. Desde un punto de origen lanza un rayo en una dirección hasta que alcanza su longitud máxima o choca contra un *collider* [30]. Una vez ha colisionado con un objeto devuelve las coordenadas exactas ese punto. En caso de no chocar con nada no devuelve ningún resultado.

## **Escenas**

Para evitar que el espacio sobre el que trabajemos este sobrecargado y tenga problemas de rendimiento, Unity divide “el mundo” en escenas [31]. Estas escenas pueden ser cargadas y quitadas en cualquier momento, y así poder organizar tu proyecto en diferentes espacios de trabajo.

En proyectos de gran tamaño son usadas para optimizar los recursos. Se cargan los niveles cuando se van a hacer uso y así evitar un derroche innecesario de recursos. Consiguiendo como fin obtener una mejora en el rendimiento que garantiza una experiencia para el usuario más agradable.

En este proyecto se hace uno de 2 escenas para separar la aplicación de las HoloLens 2 y la del ordenador.

## **Rendimiento**

Se entiende por rendimiento a maximizar la velocidad de renderización de los gráficos de la aplicación. Cuanto mejor sea el rendimiento mejor calidad tendrá el producto final y mejor experiencia ofrecerá al usuario. El rendimiento es medido en *Frames per Second* (FPS) 3.2, a mayor cantidad, mayor rendimiento.

Para obtener un mayor rendimiento existen 2 posibilidades, invertir en un hardware más potente u optimizar todos los objetos de la escena de Unity. En el caso de este proyecto la primera opción es imposible, puesto que, no existe ningún dispositivo con las funcionalidades que ofrece las HoloLens 2 que tenga mayor potencia.

Por lo que para obtener la mayor cantidad de FPS la única posibilidad es optimizar al máximo las escenas de Unity.

## **FPS**

Siglas de fotogramas por segundo [36], como su nombre indica es la cantidad de imágenes que son imprimidas en nuestra pantalla por segundo.

La frecuencia de actualización máxima de las HoloLens 2 tiene un límite establecido en 60 FPS, por lo que lo ideal sería alcanzar esa cifra. Lamentablemente cuando la vista en primera persona de las gafas es grabada, esta baja su frecuencia de actualización a la mitad. Esto se debe a un límite físico creado por la capacidad de procesamiento de la CPU.

Al ser completamente indispensable la grabación de la pantalla de las HoloLens 2 para la realización del proyecto, la mayor cantidad de rendimiento que se va a poder obtener es de 30 FPS.



---

# Técnicas y herramientas

---

En esta sección de la memoria se cubrirá las técnicas metodológicas y las herramientas han sido utilizadas en el desarrollo de este proyecto. También se mencionarán las alternativas que se han tenido en consideración de las diferentes herramientas utilizadas o que hayan sido desechadas según iba evolucionando el proyecto.

## 4.1. Metodologías

### Scrum

Dentro de ITCL la metodología usada es Scrum [22]. Esta metodología es un esquema de trabajo para el desarrollo de *software*, donde se aplican unas prácticas de trabajo colaborativas caracterizadas por un desarrollo incremental del producto. Se van realizando ciclos de un corto periodo de tiempo que constan con planificaciones, reuniones, desarrollo y revisiones del producto (*sprints*).

Tras tener en cuenta que se trata de un prototipo, con un tiempo y tamaño de desarrollo menor que los proyectos oficiales de la empresa, decidí optar por esta metodología debido a que era capaz de cubrir todas las necesidades del trabajo.

## 4.2. Control de versiones

El *software* de control de versiones [35] considerado para este proyecto fue Git, puesto que es el usado dentro de ITCL y necesario para trabajar con sus repositorios privados. Git es un sistema de control de versiones

distribuido donde cada desarrollador puede tener en local una copia propia del proyecto, e ir modificando sin ningún riesgo el proyecto de origen alojado en un repositorio. La GUI (Interfaz gráfica de usuario) a través de la que había trabajado con anterioridad fue GitHub Desktop.

Haciendo uso del GitHub Student Developer Pack al que tenemos acceso todos los estudiantes gracias a la Universidad de Burgos, utilizaré la aplicación GitKraken para el control de versiones.

## **GitKraken**

GitKraken es una interfaz gráfica de usuario multiplataforma para el control de versiones con Git. He tomado la elección de usar esta aplicación debido a su interfaz sencilla e intuitiva, funcionalidades y su aspecto visualmente agradable.

### **4.3. Repositorio**

Inicialmente alojé el proyecto en el servidor propio que tiene ITCL de Bitbucket [2] y posteriormente decidí crear una copia en la plataforma GitHub para poder acceder al proyecto de forma personal.

## **Bitbucket**

Es un servicio web de alojamiento de proyectos para el control de versiones colaborativo. Está desarrollado en python y permite trabajar tanto con Git como con Mercurial. Puede ser usado personalmente o comercialmente. ITCL tiene su propio servidor privado, lugar donde está subido este proyecto.

## **GitHub**

Plataforma web de desarrollo colaborativo para el control de versiones Git de proyectos. Es completamente gratuita y ofrece todas las funcionalidades de Git. Para muchos es considerado el mejor servicio de alojamiento de proyectos. Además, complementaremos esta plataforma con el uso de ZenHub para la gestión del proyecto.



## 4.4. Gestión de proyecto

Al igual que la elección del repositorio, comencé planificando y registrando mi trabajo en Jira [23], herramienta utilizada en ITCL para la administración de los proyectos. Más tarde, con la elección de GitHub como segundo repositorio, elegí ZenHub para realizar la copia del seguimiento del proyecto y así poder contar con un duplicado de toda la gestión del proyecto y tener acceso a ella en cualquier momento.

### Jira

Herramienta online para el desarrollo de software que hace uso de metodologías ágiles. Permite la administración de tareas, seguimiento de errores y gestión del proyecto. Todo esto es posible gracias a los tableros Scrum, hojas de ruta, informes y gráficos.

### Zenhub

ZenHub al igual que Jira, es una herramienta web para la gestión de proyectos con metodologías ágiles. Está diseñada para trabajar con el repositorio GitHub. Ofrece las funcionalidades típicas para la organización de proyectos. Cuenta con una aplicación web y una extensión para navegadores. Por comodidad he trabajado con la extensión de ZenHub para el navegador Google Chrome.

## 4.5. Unity

La elección de este motor gráfico para el desarrollo de la aplicación viene dada por ITCL, puesto que es la herramienta de trabajo utilizada por el departamento donde me encontraba.

### UnityHub

Herramienta desarrollada por Unity para gestionar los proyectos de forma local. Permite crear, copiar y editar proyectos, gestionar la instalación de múltiples versiones del editor de Unity y actualizar el estado de la licencia asociada a tu cuenta.

### Versión Unity

La versión de Unity Utilizada para este proyecto es 2019.4.14f

## 4.6. *Plugins* Unity

Los *plugins* de Unity son una parte importante del desarrollo, ya que ofrecen al usuario una serie de funcionalidades ya implementadas por terceros. Estos pueden ser gratuitos, de pago o un término medio entre las dos opciones, que dependerá de una serie de condiciones. Todos los *plugins* utilizados en este proyecto son completamente de código abierto, libres de ser usados y modificados a disposición del usuario.

### Mixed Reality WebRTC

Provee un soporte multiplataforma de comunicación en tiempo real de audio, vídeo y datos [33]. Incluyendo dispositivos como ambas generaciones de HoloLens.

### Mixed Reality WebRTC samples

Paquete que cuenta con ejemplos y escenarios típicos para Mixed Reality WebRTC.

### Rider Editor

Pequeño *plugin* que instala Rider para comunicarse con el editor de Unity.

### Text Mesh Pro

Mejora visual y de rendimiento a todas las interfaces clásicas de Unity.

### Windows Mixed Reality

Paquete que incluye todas las herramientas necesarias para desarrollar aplicaciones para las HoloLens 2 en Unity [32].

### Unity UI

Paquete instalado por defecto de Unity que provee herramientas necesarias para el diseño e implementación de interfaces.

## Visual Studio Code Editor

Paquete instalado por defecto de Unity que permite la edición de código con Visual Studio 2019.

## PUN 2

Photon Unity Networking [19] es un paquete desechado durante el desarrollo que permite conectar aplicaciones de forma global de manera sencilla.

## 4.7. Entorno de desarrollo integrado (IDE)

Por defecto Unity instala Microsoft Visual Studio [25]. No tenía intención de trabajar con este IDE, pero es completamente necesario para el desarrollo de aplicaciones para las HoloLens 2, debido a que una vez realizada la *build* del proyecto de Unity, se instala la aplicación en las gafas a través de Visual Studio. El uso que le di fue para probar los prototipos durante el desarrollo y el lanzamiento final de la aplicación en las gafas. Toda la parte de programación ha sido realizada desde Rider [6] debido a preferencias personales con este entorno de desarrollo.

## Visual Stuido

Entorno de desarrollo perteneciente a Microsoft con una gran compatibilidad con diferentes tipos de lenguajes de programación como C++, C#, Visual Basic .NET, F#, Java, Python, Ruby y PHP. Asignado por defecto para la creación y edición de scripts en Unity.

## Rider

Es un entorno de programación .NET desarrollado por JetBrains orientado a la programación con C# y completamente compatible con Unity. La licencia de este software está incluida en GitHub Student Developer Pack.

## 4.8. Otros

A esta sección pertenecen diferentes herramientas que no cuentan con una categoría en común donde ser agrupadas.

## Node.js

Entorno de ejecución de código abierto y multiplataforma utilizado para realizar la conexión de los 2 dispositivos a través del *plugin* de WebRTC. Haremos uso de un proyecto de código abierto llamado node.dss [20] para establecer la conexión.

## Windows Device Portal

Portal de dispositivos de realidad virtual y mixta de Microsoft [12] utilizado para la administración y configuración de las HoloLens 2 a través de la red. Esta aplicación fue usada durante las primeras fases de desarrollo del proyecto. Para instalarla es necesario en uso de la Microsoft Store de Windows.

## SDK Windows 10

Kit de desarrollo de software de Windows [15] completamente necesario para trabajar con las HoloLens 2, puesto que este *hardware* trabaja directamente sobre este sistema operativo.

## Microsoft Azure

Servicio en la nube perteneciente a Microsoft [14] que proporciona un conjunto de más de 200 herramientas diseñadas para, por ejemplo, lanzar aplicaciones, administrar servidores, añadir funcionalidades a nuestras aplicaciones, etc. Tres son las herramientas que se iban a hacer uso de este servicio: *streaming*, reconocimiento de objetos y renderizado de objetos 3D de manera remota.

En una fase del proyecto este servicio fue desechado debido a la complejidad de su implementación, la gran curva de aprendizaje y el alto coste de su licencia. No se descarta su uso en un futuro cuando el proyecto pase a las manos de ITCL y deje de ser un simple prototipo.

---

# Aspectos relevantes del desarrollo del proyecto

---

En este apartado se hará mención de los puntos más relevantes ocurridos durante el desarrollo de este proyecto. Comentaré los problemas de mayor importancia aparecidos durante la implementación de ciertas funcionalidades de la aplicación y las decisiones que tomé para eludirlos.

## 5.1. Inicio del proyecto

Las primeras semanas del proyecto fueron completamente dedicadas a la planificación de todas las tareas por realizar, investigación y a la toma de contacto con las herramientas con las que iba a trabajar.

Primero instalé Unity y creé un proyecto de prueba para aprender los todos conceptos básicos y sus herramientas. Más adelante cuando ya contaba con una base de conocimientos, programé unos pequeños scripts en C# para ver cómo afecta la manipulación de los componentes al funcionamiento de los *GameObjects*.

Al mismo tiempo configuré las HoloLens 2 y probé las aplicaciones ya instaladas para aprender cómo usar las gafas. Una vez adquirida una idea sobre el coste para desarrollar aplicaciones en las gafas, comencé con la gestión del proyecto. Dividí los objetivos principales en 3 e inicié la investigación y desarrollo de forma individual de cada una de las partes.

## 5.2. Comunicación por vídeo y audio

Esta es la parte del proyecto donde más tiempo se invirtió, dado que surgieron un número considerable de dudas sobre cuál de las herramientas investigadas era la más adecuada para continuar. El primer método que tuve en consideración para realizar la conexión y enviar vídeo en tiempo real fue el *plugin* propio de Unity (Multiplayer HLAPI) [29] pero descarté rápidamente su uso que estaba en fase beta y no me ofrecía todas las funcionalidades que necesitaba.

El siguiente *plugin* que del que hice uso fue PUN 2. En un principio era el *plugin* definitivo que iba a usar, ya que era gratuito y muy fácil de usar. Tras la realización de diferentes pruebas logré conectar las 2 aplicaciones, sin embargo, el paquete no estaba preparado para el envío de vídeo. Tras los malos resultados del prototipo y ver que la licencia era "gratis", pero estaba sujeta a unas condiciones que si se cumplían pasaba a ser de pago, descarté el uso de Photon Network 2.

Tras descartar 2 *plugins* de Unity, probé con el servicio en línea de Microsoft Azure. Este incluye muchas funcionalidades prometedoras para añadir a este proyecto, aunque cuentan con una alta dificultad para ser implementadas. Tras la complejidad que suponía usar Azure, más el alto coste de su licencia una vez terminado el periodo de prueba, descarté su uso para el prototipo.

Al final alcancé la solución final, el uso de WebRTC. Proyecto de código abierto para la comunicación en tiempo real para web, aplicaciones de ordenador y móviles. Permite la comunicación por voz y por vídeo, y cuenta con un *plugin* para Unity con compatibilidad para las HoloLens 2. Este *plugin* me ofrecía todas las características que necesitaba para el desarrollo del proyecto, por lo que lo instalé, creé un servidor local para conectar ambas aplicaciones y realicé un prototipo de videoconferencia que resultó en un éxito. Terminé añadiendo alguna mejora a este prototipo y pasé a la siguiente fase del proyecto.

## 5.3. Asistencia Remota

En esta sección hay dos partes destacables: los cálculos realizados para hacer aparecer de forma remota indicadores en realidad aumentada y los contratiempos que surgieron una vez implementada esta característica.

## Interpretación de la información y cálculos vectoriales

Los pasos seguidos para implementar la parte de asistencia remota son:

- Hacer clic en la pantalla del ordenador.
- Sacar las coordenadas del clic y enviarlas a las HoloLens 2.
- Recibir las coordenadas y traducirlas a la resolución de las HoloLens 2.
- Calcular la distancia y la posición para hacer aparecer la flecha.

En la aplicación del ordenador hay un plano invisible que se ajusta a la resolución de la pantalla del ordenador. La esquina inferior izquierda del plano corresponde a la coordenada X:0 Y:0 y la esquina superior derecha corresponderá a la resolución de la pantalla del ordenador, por ejemplo, si el monitor es de 1920x1080 las coordenadas irán de X:0 Y:0 a X:1920 Y:1080. Haciendo clic en la pantalla se recoge la información de la coordenada y es enviada al servidor. Las coordenadas enviadas son traducidas a un valor decimal que va del 0 al 1 (en este caso 1 correspondería a 1920 en la horizontal y 1080 en la vertical) para facilitar su posterior manipulación. Una vez enviada la información al servidor, la aplicación de las HoloLens 2 recoge este dato.

En las gafas existe una estructura similar a la del ordenador para interpretar la información recibida. Hay colocado un plano invisible, con el tamaño equivalente a la resolución a la que graban las gafas, que está ajustado con el campo de visión de las HoloLens 2. En la esquina inferior izquierda (posición por defecto) hay un objeto que únicamente cuenta con el componente de transformación, es decir, que solo tiene coordenadas. Este objeto es hijo del plano, lo que implica que obtiene unas coordenadas locales sobre este, y estas son: X:0 Y:0 para la esquina inferior izquierda y X:1 Y:1 para la esquina superior derecha. Una vez recibida la información del servidor, el *GameObject* en el plano adquiere las coordenadas del clic.

Para terminar, obtengo la dirección del *raycast* realizando la siguiente operación [1]: obtengo las coordenadas globales del objeto en el plano, le resto las de la cámara y normalizo el resultado. Lanzo el *raycast* con punto de origen las coordenadas de la cámara y la dirección ya calculada. El punto donde colisione con la asignación espacial será el lugar donde instancie la flecha.

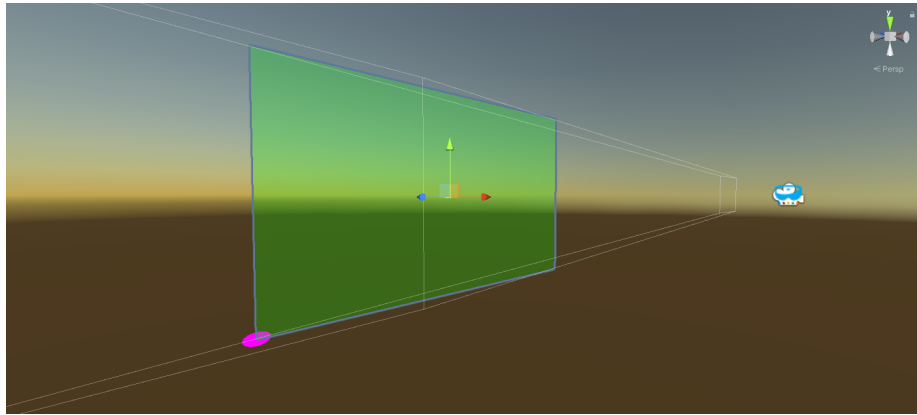


Figura 5.3: Se muestra el plano correspondiente a la resolución de las HoloLens 2 (verde) y el objeto (rosa) que se mueve sobre este para obtener la dirección del *raycast*

## Problemas con la resolución de la pantalla

Un problema en el que tuve que invertir bastante tiempo fue en calibrar todo lo mencionado anteriormente, ya que las HoloLens 2 según el uso que se le esté dando, graban su vista a una resolución u otra [8]. La lista de las resoluciones usadas es bastante amplia, por lo que me demoré para encontrar la correcta.

Otro contratiempo surgió debido a que tanto la vista de la realidad como los hologramas son grabados del ojo derecho de las gafas, por lo que a la hora de instanciar los objetos, no aparecían exactamente donde debían. Tuve que ajustar manualmente el plano e ir haciendo pruebas hasta lograr corregir este fallo.



---

## Trabajos relacionados

---

En esta sección analizaremos y compararemos los diferentes trabajos y proyectos que cuenten con características similares a este TFG.

Debido a exclusividad de las HoloLens 2 escasea bastante la cantidad de aplicaciones desarrolladas para estas. La única que se asemeje a mi proyecto en cuanto a características y funcionalidad es software Dynamics 365 | Remote Assist desarrollado por Microsoft.

### 6.1. Dynamics 365 | Remote Assist

Dynamics 365 [10] es un conjunto de aplicaciones desarrolladas por Microsoft que están enfocadas a un entorno empresarial. Uno de esos productos es Remote Assist, aplicación de comunicación de audio y vídeo diseñada para resolver problemas en tiempo real. Esta conecta a un experto desde un ordenador a un trabajador con un teléfono móvil u HoloLens que se encuentre presencialmente en el fallo a solucionar.

A continuación, veremos una comparativa entre lo que ofrecen ambas aplicaciones:

Características	Remote Assist	TFG
Comunicación por vídeo	Si	Si
Comunicación por audio	Si	Si
Interfaz llamada	Si	Si
Asistencia remota	Si	Si
Seguimiento y manipulación interfaz	No	Si
Esconder la interfaz	No	Si
Envío de PDFs	Si	No
Alcance global	Si	No
Interfaz de contactos	Si	No
Almacenamiento de capturas	Si	No

Tabla 6.1: Comparativa de las características de las aplicaciones.

Obviamente al comparar este prototipo con una aplicación desarrollada por una empresa de este calibre, vemos que mi proyecto es claramente inferior. Pero este resultado no es negativo, ya que se puede observar que ambas comparten una cantidad considerable de funcionalidades. Esto quiere decir que el prototipo desarrollado en este TFG establece una muy buena base para que un futuro, cuando ITCL retome este proyecto, supere fácilmente en funcionalidades la aplicación de Microsoft.

---

# Conclusiones y Líneas de trabajo futuras

---

## 7.1. Conclusiones

Tras la finalización de este proyecto se puede concluir que todos los objetivos principales han sido completados satisfactoriamente. Dando lugar a una aplicación prototipo que permite la comunicación mediante vídeo y audio del dispositivo de realidad mixta HoloLens 2 y un ordenador. Otra funcionalidad presente en el resultado final es la capacidad de asistir remotamente a través de los indicadores generados en realidad aumentada.

En el proyecto se han conseguido incluir parte de los objetivos secundarios propuestos. Se ha implementado funcionalidades básicas que caracterizan las aplicaciones de comunicación por voz y vídeo. Puedes silenciar tu micrófono, apagar la webcam, llamar, colgar, etc. También se ha desarrollado una interfaz gráfica basada en botones para poder acceder a estas funcionalidades.

Al comienzo del proyecto tenía cierta preocupación puesto que tenía un límite establecido de 300 horas y nunca había trabajado ninguna de las herramientas utilizadas. Esta cantidad de horas resultó ser muy justa, ya que terminé el prototipo en la última semana del contrato de prácticas. Tuve que invertir más horas de las establecidas en el TFG debido a que posteriormente a la finalización del contrato en ITCL le dediqué una cantidad bastante considerable de horas para trabajar en la documentación. Comencé completamente de 0, sin ningún conocimiento de Unity, C# ni de las HoloLens 2, por lo que al comienzo del proyecto no podía realizar una estimación clara de todo el trabajo a realizar.

Trabajar con Unity fue relativamente sencillo, las características principales de este motor gráfico son fáciles de aprender y la interfaz es bastante intuitiva. Afirmación que no puedo realizar con las HoloLens 2. A lo largo de la realización TFG, encontré una cantidad considerable de bugs relacionados con el software de las gafas. El rendimiento final de la aplicación también se vio disminuido debido a limitaciones de hardware.

Personalmente estoy satisfecho con el resultado final, por que se han cumplido todos objetivos principales y parte de los secundarios. Durante el proyecto he empleado muchas herramientas y conocimientos adquiridos en el grado, que me ha sido de gran utilidad durante ciertas fases del proyecto. También he adquirido nueva experiencia usando herramientas como Unity, L<sup>A</sup>T<sub>E</sub>X, Jira...

Para terminar, quiero dar las gracias a ITCL, puesto que con su colaboración, he tenido la oportunidad de trabajar con una tecnología a la que muy pocos tienen acceso debido a su exclusividad y elevado coste. También quiero agradecer a Ángel Arroyo, Nuño Basurto y Alejandro Langerica por ser mis tutores de este TFG.

## 7.2. Líneas de trabajo futuras

Las principales mejoras a incluir a este prototipo son:

- Evolucionar el método actual utilizado para realizar la conexión entre los 2 dispositivos. Obtener como resultado final una comunicación a través de internet de forma global.
- Implementar un sistema de reconocimiento de objetos combinando el proyecto actual con el servicio de Microsoft Azure.
- Crear una interfaz previa a la llamada donde puedas agregar contactos y comunicarte con ellos mediante texto.
- Cargar modelos 3D en las HoloLens 2 para visualizarlos en realidad aumentada.
- Enviar un documento en formato .pdf desde el ordenador para generarlo en realidad aumentada, haciendo así uso de él.

---

## Bibliografía

---

- [1] Berenger. How to calculate direction between 2 objects. <https://answers.unity.com/questions/697830/how-to-calculate-direction-between-2-objects.html>, 2021.
- [2] Bitbucket. Built for professional teams. <https://bitbucket.org/>, 2021.
- [3] Blogthinkbig. Qué son los motores gráficos y cuáles son los más populares. <https://blogthinkbig.com/motores-graficos#:~:text=Los%20desarrolladores%20de%20juegos%20no,lo%20largo%20de%20muchos%20a%C3%B1os.>, 2021.
- [4] GitHub. Where the world builds software. <https://github.com/>, 2021.
- [5] ITCL. Itcl - quiénes somos. <https://itcl.es/itcl-quienes-somos/>, 2021.
- [6] JetBrains. Rider: El ide .net multiplataforma de jetbrains. <https://www.jetbrains.com/es-es/rider/>, 2021.
- [7] Microsoft. Asignación espacial. <https://docs.microsoft.com/es-es/windows/mixed-reality/design/spatial-mapping>, 2021.
- [8] Microsoft. Cámara localizable. <https://docs.microsoft.com/es-es/windows/mixed-reality/develop/platform-capabilities-and-apis/locatable-camera>, 2021.
- [9] Microsoft. Documentación de c#. <https://docs.microsoft.com/es-es/dotnet/csharp/>, 2021.

- [10] Microsoft. Funcionalidades de dynamics 365 remote assist. <https://dynamics.microsoft.com/es-es/mixed-reality/remote-assist/capabilities/>, 2021.
- [11] Microsoft. Hololens 2. <https://www.microsoft.com/es-es/hololens>, 2021.
- [12] Microsoft. Introducción a windows device portal. <https://docs.microsoft.com/es-es/windows/uwp/debug-test-perf/device-portal>, 2021.
- [13] Microsoft. Menú mano. <https://docs.microsoft.com/es-es/windows/mixed-reality/design/hand-menu>, 2021.
- [14] Microsoft. Microsoft azure: Servicios de informática en la nube. <https://azure.microsoft.com/es-es/>, 2021.
- [15] Microsoft. Sdk de windows 10. <https://developer.microsoft.com/es-es/windows/downloads/windows-10-sdk/>, 2021.
- [16] Microsoft. Seguimiento de los ojos en hololens 2. <https://docs.microsoft.com/es-es/windows/mixed-reality/design/eye-tracking>, 2021.
- [17] Microsoft. Seguimiento de manos y ojos articulados en unity. <https://docs.microsoft.com/es-es/windows/mixed-reality/develop/unity/hand-eye-in-unity>, 2021.
- [18] Microsoft. ¿qué es mixed reality? <https://docs.microsoft.com/es-es/windows/mixed-reality/discover/mixed-reality>, 2021.
- [19] Photon Unity Networking. We make multiplayer simple. <https://www.photonengine.com/en-US/Photon>, 2021.
- [20] Node.js. node-dss. <https://github.com/bengreenier/node-dss>, 2021.
- [21] Node.js®. Entorno de ejecución node.js. <https://nodejs.org/es/>, 2021.
- [22] ProyectosAgiles. ¿qué es scrum? <https://proyectosagiles.org/que-es-scrum/>, 2021.
- [23] Jira Software. La herramienta de desarrollo de software líder de los equipos ágiles. <https://www.atlassian.com/es/software/jira>, 2021.

- [24] StereoLabs. Spatial mapping overview. <https://www.stereolabs.com/docs/spatial-mapping/>, 2021.
- [25] Visual Studio. Visual studio. <https://visualstudio.microsoft.com/es/>, 2021.
- [26] Unity. Field of view. <https://docs.unity3d.com/ScriptReference/Camera-fieldOfView.html>, 2021.
- [27] Unity. GameObject. <https://docs.unity3d.com/ScriptReference/GameObject.html>, 2021.
- [28] Unity. La plataforma líder para crear contenido interactivo en tiempo real. <https://unity.com/es>, 2021.
- [29] Unity. The multiplayer high level api. <https://docs.unity3d.com/Manual/UNetUsingHLAPI.html>, 2021.
- [30] Unity. Raycast. <https://docs.unity3d.com/ScriptReference/Physics.Raycast.html>, 2021.
- [31] Unity. Scenes. <https://docs.unity3d.com/Manual/CreatingScenes.html>, 2021.
- [32] Unity. Unity entra al terreno inmersivo con windows mixed reality. <https://unity3d.com/es/partners/microsoft/mixed-reality>, 2021.
- [33] WebRTC. Mixedreality-webrtc. <https://github.com/microsoft/MixedReality-WebRTC>, 2021.
- [34] Wikipedia. Augmented reality. [https://en.wikipedia.org/wiki/Augmented\\_reality](https://en.wikipedia.org/wiki/Augmented_reality), 2021.
- [35] Wikipedia. Control de versiones. [https://es.wikipedia.org/wiki/Control\\_de\\_versiones](https://es.wikipedia.org/wiki/Control_de_versiones), 2021.
- [36] Wikipedia. Fotogramas por segundo. [https://es.wikipedia.org/wiki/Fotogramas\\_por\\_segundo](https://es.wikipedia.org/wiki/Fotogramas_por_segundo), 2021.
- [37] Wikipedia. Mixed reality. [https://en.wikipedia.org/wiki/Mixed\\_reality](https://en.wikipedia.org/wiki/Mixed_reality), 2021.
- [38] Wikipedia. Virtual reality. [https://en.wikipedia.org/wiki/Virtual\\_reality](https://en.wikipedia.org/wiki/Virtual_reality), 2021.

- [39] ZenHub. Project management in github. <https://www.zenhub.com/>, 2021.