

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCOLA POLITECNICA SUPERIOR DE GANDIA

Grado en Ing. Sist. de Telecom., Sonido e Imagen



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCOLA POLITÈCNICA
SUPERIOR DE GANDIA

“Diseño, implementación y evaluación de un reproductor basado en web para vídeo omnidireccional, siguiendo las especificaciones de sincronización del estándar HbbTV v2.0.1”

TRABAJO FINAL DE GRADO

Autor:

Jair Daynor López Gutiérrez

Tutores:

**Fernando Boronat Seguí
Dani Marfil Reguero**

GANDIA, 2019

RESUMEN

El servicio de TV está en constante evolución. Como ejemplo, los fabricantes de *Smart TV's* están implementando el estándar *Hybrid Broadcast Broadband TV* en los dispositivos de venta en muchos países europeos. Una de las funcionalidades de dicho estándar (en la que se centra este TFG) consiste en la posibilidad de conectar y sincronizar dispositivos secundarios con el televisor. Los proveedores seguirán transmitiendo el contenido principal por *broadcast* (DVB-T), y, además, también proporcionarán contenido relacionado (incluso omnidireccional), vía *streaming* IP (*broadband*) para ser reproducidos bajo demanda en dispositivos externos, de forma sincronizada respecto al televisor, haciendo la experiencia de consumo más personalizada e inmersiva. Los fabricantes aún no han implementado dicha funcionalidad en sus receptores, por lo que aún queda un largo recorrido para proporcionar este tipo de experiencias.

En este proyecto se ha desarrollado una aplicación web (compatible con dicho estándar) para *smartphones*, *tablets* y PCs, incluyendo un escenario inmersivo, un reproductor de vídeos 2D y otro para vídeos 360° de forma sincronizada con el TV. Se ha evaluado, tanto objetiva como subjetivamente, consiguiendo unos resultados satisfactorios.

PALABRAS CLAVE

HbbTV; sincronización entre dispositivos; IDES; vídeo omnidireccional; vídeo 360

ABSTRACT

The way of TV content consumption is continuously evolving. In recent years, Smart TV manufacturers are implementing the Hybrid Broadcast Broadband TV standard in their devices. In Spain, some TV channels are providing hybrid services through hybrid TDT.

This is an interactive Spanish TV specification that integrates TDT and Internet in the same user experience. The HbbTV version that nowadays is used is v2.0.1, and this has the latest developments and more possibilities in relation to v1.5. One of them, that this TFG is focused on, is the possibility to connect and synchronize external devices to the TV to watch related content, even the same content, but with different perspectives or different formats. For example, through Internet it is possible to get related omnidirectional content (360 video) to that the TV is playing and so to get a more immersive user experience.

In this TFG is developed a web app that integrates an immersive scene, a video 2D player and omnidirectional video player. This app is compatible with HbbTV v2.0.1 because it implements inter-device synchronization specs. Moreover, the app was objectively and subjectively evaluated.

KEY WORDS

HbbTV; Inter-device synchronization; IDES; Omnidirectional video; video 360

ÍNDICE

1	INTRODUCCIÓN	5
1.1	INTRODUCCIÓN	5
1.2	OBJETIVOS INICIALES DEL TFG	7
1.3	ESTRUCTURA DE LA MEMORIA	8
2	ESTADO DEL ARTE	9
2.1	EL VIDEO OMNIDIRECCIONAL O 360°	9
2.2	SINCRONIZACIÓN ENTRE DISPOSITIVOS (IDES)	12
2.3	REPRODUCCIÓN DE VÍDEO OMNIDIRECCIONAL	14
3	TECNOLOGÍAS, ESTÁNDARES Y ESPECIFICACIONES UTILIZADAS	16
3.1.1	<i>HbbTV v2.0.1</i>	16
3.1.2	<i>MPEG-DASH</i>	16
3.1.3	<i>DIAL</i>	16
3.1.4	<i>WebSocket</i>	17
3.1.5	<i>TEMI</i>	18
3.1.6	<i>NTP</i>	18
3.1.7	<i>HTML5</i>	18
3.1.8	<i>CSS3</i>	18
3.1.9	<i>JavaScript</i>	18
3.1.10	<i>Three.js</i>	18
3.1.11	<i>JSON</i>	19
3.1.12	<i>Dash.js</i>	19
3.1.13	<i>Node.js</i>	19
3.1.14	<i>Apache HTTP Server</i>	19
3.1.15	<i>FFmpeg</i>	19
3.1.16	<i>GPAC</i>	19
3.1.17	<i>Bento4</i>	19
4	DESARROLLO DEL ESCENARIO INMERSIVO CAPAZ DE REPRODUCIR VIDEOS 360° Y COMPATIBLE CON EL ESTÁNDAR HBBTV 2.0.1	20
4.1	ARQUITECTURA DE LA PLATAFORMA HÍBRIDA	20
4.1.1	<i>Proveedor de contenidos</i>	20
4.1.2	<i>Aplicación para el televisor híbrido con HbbTV 2.0.1 (Main Screen)</i>	23
4.1.3	<i>Aplicación para los dispositivos secundarios (Companion Screens)</i>	25
5	EVALUACIÓN	36
5.1	EVALUACIÓN OBJETIVA (QOS)	36
5.2	EVALUACIÓN SUBJETIVA (QOE)	38
5.2.1	<i>Participantes</i>	38
5.2.2	<i>Evaluación</i>	39
6	CONCLUSIONES Y TRABAJO FUTURO	44
7	BIBLIOGRAFÍA	45
ANEXO I: OBTENCIÓN DE LAS VISTAS 2D A PARTIR DE UN VÍDEO 360		
ANEXO I: TUTORIAL DEL ESCENARIO INMERSIVO 360		
ANEXO II: RESULTADOS DE LA EVALUACIÓN SUBJETIVA DE LA APLICACIÓN EXTRA		

1 INTRODUCCIÓN

1.1 Introducción

Hoy en día el uso de las tecnologías híbridas (*broadcast* y *broadband*) en el televisor es una realidad. Esta combinación de tecnologías permite un nuevo escenario, nuevas posibilidades, donde la transmisión tradicional de televisión (*broadcast*) puede ser enriquecida con contenidos complementarios a través de redes *broadband* (p.ej. Internet) que pueden ser consumidos en varios dispositivos y de manera sincronizada, respecto al contenido que se está visualizando en el televisor.

Prueba del potencial de la combinación de tecnologías híbridas es la existencia de estándares, como HbbTV en Europa y alianzas entre fabricantes como la Smart TV Alliance. HbbTV (*Hybrid Broadcast Broadband TV*) [1] es un estándar paneuropeo de televisión híbrida cuyo objetivo es combinar las emisiones de televisión (*broadcast*) con servicios de banda ancha (*broadband*) para entregar al espectador un servicio de entretenimiento a través de una pantalla de televisión. Esta especificación es desarrollada por los líderes de la industria para mejorar la experiencia del usuario. La especificación HbbTV utiliza elementos de especificaciones existentes de otras normas, como OIPF (*Open IPTV Forum*), DVB (*Digital Video Broadcasting*), W3C (*World Wide Web Consortium*).

En particular, la versión 2.0.1 del estándar HbbTV [2] proporciona funcionalidades y mecanismos para consumir sincronizadamente contenido híbrido en el televisor principal también denominado MS (*Main Screen*) y opcionalmente en uno o más dispositivos secundarios o CS (*Companion Screen*), tales como *smartphones*, *tablets*, HMD's u ordenadores. Para ello se basa en los protocolos del estándar DVB-CSS (*DVB-Companion Screens & Streams*) [3].

Actualmente, la tendencia de consumo de contenidos a través de Internet usando diferentes dispositivos está aumentando. Según [4], por un lado, Netflix ha sido una de las 10 aplicaciones más descargadas en los smartphones en el año 2018 y, por otro lado, el 92% de los usuarios de Internet consume vídeos online.

Además, la generación y el consumo de vídeos omnidireccionales (360°) y contenido VR (*Virtual Reality*) está aumentando significativamente. Según [5], el 11% de la población adulta de EEUU tiene un dispositivo para consumir contenido VR. Dicho dato fue obtenido en noviembre de 2018, mientras que en agosto de 2017 el porcentaje era del 7%.

Respecto al consumo de contenidos híbridos simultáneos, existen todavía muchos retos por afrontar. Entre ellos, la sincronización es uno de los retos más importantes, ya que, los mecanismos de sincronización todavía necesitan ser mejorados. Sobre el estándar HbbTV, una encuesta en [6] muestra que, por un lado, el 35,8% de los participantes piensan que los servicios ofrecidos para este tipo de tecnología son mínimos y aún tienen muchas expectativas. Por otro lado, a pesar de esa visión pesimista sobre los servicios de HbbTV actuales, el 23,2% piensan que HbbTV es una tecnología muy atractiva y el 45,2% piensan que es moderadamente atractiva.

Según [7], a pesar del incremento de la tendencia de consumo multimedia a través de Internet, el consumo de televisión tradicional es todavía la principal fuente de entretenimiento para la población con edad mayor a 18 años.

Esos son los motivos por los cuales en este TFG (Trabajo Final de Grado) se propone mejorar el enfoque híbrido para el consumo televisivo, donde el vídeo 360° se entrega vía *broadband*, pero se sigue manteniendo la transmisión tradicional por TDT

(Televisión Digital Terrestre) utilizando el estándar DVB-T (*Digital Video Broadcasting - Terrestrial*) para difundir el contenido principal.

Además, para proporcionar un enfoque comercial realista, esta propuesta se ha diseñado siguiendo las pautas del estándar HbbTV v2.0.1 [2].

El TFG se ha realizado en el seno del grupo de investigación del Departamento de comunicaciones del Campus de Gandia *Immersive & Interactive Media* (IIM) *R&D Group*, de la UPV (Universidad Politécnica de Valencia).

Para que se entienda mejor el trabajo realizado en este TFG, a continuación, se muestra un ejemplo real del uso de tecnologías híbridas.

Imaginemos una producción televisiva que se transmitirá en diferido, es decir, que no se emitirá en directo. Se está grabando un evento deportivo, por ejemplo, una demostración de un coche de fórmula 1. Se dispone de tres cámaras convencionales y una cámara 360°. Las cámaras convencionales están repartidas en lugares estratégicos del circuito, y la cámara 360° se encuentra en el vehículo monoplace.

El realizador escogerá las mejores tomas de las cámaras convencionales y se creará una mezcla final del programa, la cual quedará grabada. Pero al mismo tiempo se tendrá la grabación completa de cada cámara. Por tanto, se tiene la mezcla final del programa a emitir, la grabación de las 3 cámaras convencionales y la grabación de la cámara 360.



FIGURA 1: MEZCLA FINAL DE UNA PRODUCCIÓN TELEVISIVA



FIGURA 2: TOMA 360



FIGURA 3: EJEMPLO DE TOMAS CAPTADAS POR LAS CÁMARAS CONVENCIONALES

Ahora la cadena de televisión debe preparar el vídeo de la mezcla final para transmitirlo por medio de ondas terrestres y que los usuarios puedan verlo en sus televisores. Además, también debe preparar todos los vídeos para dejarlos accesibles a través de Internet para que los usuarios los puedan ver en dispositivos secundarios (*smartphones*, *tablets*, PCs) a través de *streaming* IP.

Entonces, si el usuario dispone de un televisor híbrido con funcionalidades de sincronización entre dispositivos de HbbTV v2.0.1, observará que en la pantalla de su televisor aparece un ícono que señala la existencia de contenidos adicionales. Estos contenidos adicionales pueden ser descargados por *streaming* a través de Internet y se pueden visualizar en algún dispositivo secundario.

Si el usuario abre la aplicación programada en este TFG, por ejemplo, en su *tablet*, observará en la pantalla de dicha *tablet*, un entorno gráfico inmersivo, para que el usuario final se sienta integrado en el entorno que se le presenta.

Se ha decidido simular el salón de una vivienda, en donde haya un televisor virtual y a la izquierda de dicho televisor aparezcan los *posters* (miniaturas) seleccionables de las vistas adicionales al contenido principal, resaltando el texto "360" si la vista es de vídeo omnidireccional; y debajo del televisor virtual aparezcan algunos botones de control.

En la FIGURA 4 se puede observar el diseño inicial de la interfaz gráfica, y es lo primero que debe observar el usuario.

Inicialmente, en el televisor virtual se reproducirá el mismo vídeo que en el televisor real y de forma sincronizada, es decir, el televisor virtual idealmente debe mostrar el mismo *frame* del vídeo mostrado en el televisor real.

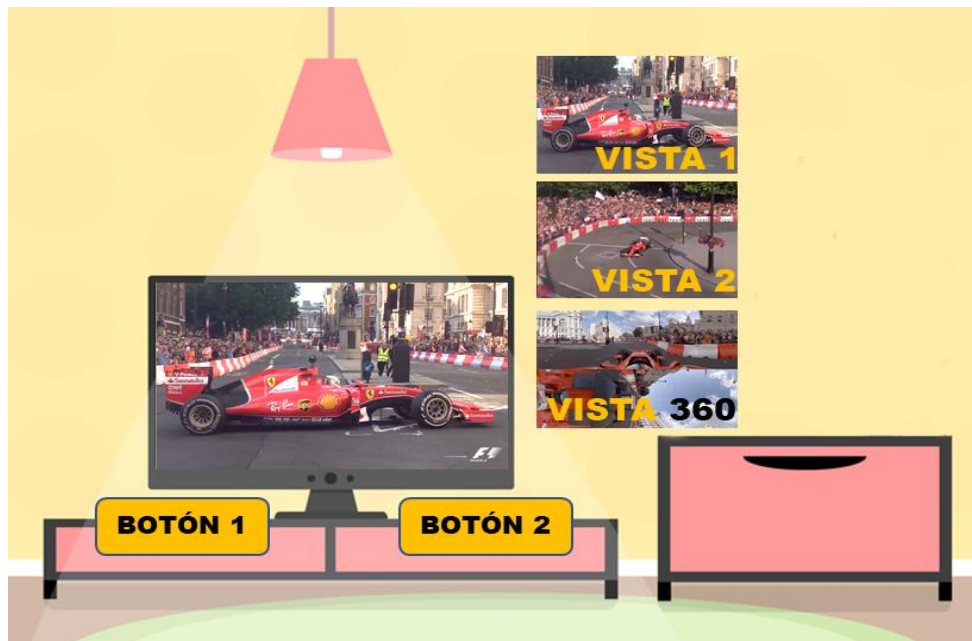


FIGURA 4: DISEÑO INICIAL DE LA INTERFAZ GRÁFICA DEL ENTORNO VIRTUAL

Después el usuario tiene la posibilidad de elegir la vista en 360° u otra vista en 2D deseada. Las vistas que tengan formato 2D se van a reproducir en el televisor virtual, pero si se selecciona la vista 360, el vídeo se debe reproducir en un reproductor especial, el cual ha sido diseñado y programado en este TFG.

Los botones servirán para que el espectador pueda tener algo de control sobre el contenido multimedia, por ejemplo, activar o desactivar el volumen, o también podrían servir para abrir o activar alguna función especial.

1.2 Objetivos iniciales del TFG

El objetivo principal que se ha marcado inicialmente para este TFG fue el diseño y la implementación de un escenario virtual basado en tecnologías web, incluyendo un reproductor de contenido de vídeo 360°, que permita el descubrimiento e intercambio de mensajes con el televisor híbrido para poder reproducir el contenido de forma sincronizada respecto al contenido principal visualizado en dicho televisor.

Dicho reproductor permite visualizar el contenido tanto en modo monoscópico, es decir, en una pantalla plana, con selección del punto de vista mediante movimientos del ratón en pantallas convencionales o su equivalente en pantallas táctiles; como en modo estereoscópico. Para esto último, se generan dos vistas del contenido, una para cada ojo, de tal forma que se puedan colocar los dispositivos de visualización (especialmente *smartphones*) en visores o gafas VR que facilitarán una experiencia mucho más inmersiva.

Además del reproductor, se ha programado e incluido en el mismo un mecanismo de descubrimiento del televisor híbrido y un algoritmo de sincronización, ambos cumpliendo con la especificación de la versión 2.0.1 del estándar HbbTV.

Adicionalmente, se han seleccionado contenidos apropiados, y se han preparado tanto para su emisión vía *broadcast* (por DVB-T) como vía *broadband* (por Internet). Para esto último, se ha utilizado *streaming* adaptativo de contenidos multimedia sobre HTTP (*HyperText Transfer Protocol*), concretamente el estándar MPEG-DASH (*Dynamic Adaptive Streaming over HTTP*) [8].

Finalmente se han realizado evaluaciones tanto objetivas (medidas de parámetros de QoS o *Quality of Service*), como subjetivas (medida de la calidad de la experiencia del usuario, QoE o *Quality of Experience*) obteniendo resultados satisfactorios.

1.3 Estructura de la memoria

Después de la introducción al tema y la aplicación, en el apartado 2 se describe el estado actual sobre vídeos 360°, sincronización entre dispositivos y sobre los reproductores de vídeos 360°. En el apartado 3 se describen todas las tecnologías, estándares y especificaciones utilizadas. En el apartado 4 se comienza a describir la plataforma híbrida, primeramente se muestra la arquitectura general de la plataforma híbrida en el apartado 4.1, y dentro de este apartado, concretamente en el punto 4.1.3 se explica con detalle la aplicación diseñada y programada en este TFG.

En el apartado 5 se explican los resultados de las evaluaciones tanto objetivas como subjetivas de la aplicación de este TFG.

Finalmente, en el punto 6 se comentan las conclusiones y trabajo futuro.

Además, se incluyen 3 anexos, en el Anexo I se describe la forma de obtener vistas 2D a partir de un vídeo 360 con proyección *CubeMap*. En el Anexo II se proporciona un breve tutorial de uso sobre la aplicación principal y extra, ambos desarrollados en este TFG. Y en el Anexo III se presentan los resultados obtenidos de la evaluación subjetiva de la aplicación extra.

2 ESTADO DEL ARTE

2.1 El video omnidireccional o 360°

Los vídeos omnidireccionales, o también conocidos como vídeos 360°, son vídeos grabados con una cámara omnidireccional o la unión de cámaras orientadas en direcciones complementarias. Estas cámaras captan los 360 grados de la escena, es decir, graban en todas las direcciones al mismo tiempo. En este tipo de contenido multimedia, el espectador tiene el control de la dirección de visualización en tiempo real [9]. Sin embargo, los usuarios sólo pueden visualizar en cada instante una pequeña sección respecto al total del contenido 360 de la escena visualizada (ver FIGURA 5). Dicha sección se conoce como campo de visión del inglés FoV (*Field of View*).

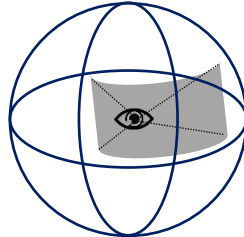


FIGURA 5: CAMPO DE VISIÓN O FoV DE UN VÍDEO 360°

Los videos omnidireccionales se empezaron a grabar masivamente a partir de la década de 2010. El motivo principal fue que hasta ese momento existía una carencia de hardware y de software para producción y edición de vídeos 360° grabados con cámaras físicas. No obstante, existía un gran interés en desarrollar tecnologías para poder grabar este tipo de contenido, pues se consideraba que la calidad de la imagen sería más realista e inmersiva que las imágenes generadas por ordenador. Por tanto, en cuanto estuvo disponible la tecnología que lo permitió, se empezó a generar contenido de este tipo [10].

Una cuestión que se presenta sobre los vídeos omnidireccionales es si se pueden considerar como RV (realidad virtual). Para responder a esta pregunta se debe definir el concepto de realidad virtual.

Según [10] en 1992, Zelter estudió los tres conceptos básicos que hacen de la RV una experiencia distinta a lo que cualquier otra tecnología digital puede ofrecer: la autonomía del usuario, la interacción que puede realizar con el espacio ficticio y la sensación de presencia del usuario en el entorno que se le presenta.

- Autonomía: Mide la capacidad del usuario para actuar y reaccionar ante el entorno ficticio en el que se encuentra.
- Interacción: Mide la capacidad del entorno de reaccionar ante las acciones del usuario. El entorno generado digitalmente tiene unos límites y cuantos menores sean esos límites, mayor será la experiencia de inmersión del usuario.
- Presencia: La presencia es la sensación del usuario de estar en el entorno digital que se le presenta.

Respecto a los vídeos 360°, al no proporcionar al usuario un entorno a través del cual pueda moverse con libertad, ya que el usuario permanece en un punto fijo y solamente es capaz de rotar sobre su propio eje, se pierde el concepto de autonomía. Los expertos del sector no se ponen de acuerdo en si debe ser considerado RV o no, por lo que no se puede afirmar o negar con firmeza. Lo que sí está claro es que el vídeo 360° mantiene aspectos en común con la RV y que el objetivo de ambos es conseguir que el usuario disfrute de una experiencia inmersiva y tenga sensación de presencia en el entorno que se le presenta.

Otra cuestión que se presenta respecto a este tipo de contenido es, cómo proyectar o convertir las imágenes esféricas en imágenes planas 2D para su posterior codificación y descodificación, ya que no existe un algoritmo de codificación de vídeo en el dominio esférico [11]. Se prefiere realizar esta conversión, debido a que ya existen varios estándares eficientes de codificación de vídeo convencional 2D. Desafortunadamente, este paso no es trivial y cuenta con muchos inconvenientes. Tal y como se expone en [11], existen numerosas técnicas para proyectar el contenido 360 en un plano bidimensional, algunas de ellas se mencionan en la TABLA 1.

TIPO DE PROYECCIÓN	ACRÓNIMO
EquiRectangular Projection	ERP
CubeMap Projection	CMP
EquiAngular Projection	EAP
Pyramidal Projection	

TABLA 1: ALGUNOS TIPOS DE PROYECCIONES 360 EN 2D

Estas técnicas pueden dividirse en dos grupos, según sean independientes del punto de vista del consumidor (p.ej. ERP o CMP), o dependientes del punto de vista del consumidor (p.ej. la proyección piramidal).

La proyección ERP se corresponde con la proyección plana de una esfera, tal y como se realizaba habitualmente al representar un globo terráqueo con un mapamundi, ver FIGURA 6.

Esta proyección es la más común, debido a que es intuitiva y fácil de generar, pero presenta inconvenientes en los polos, ya que en estas zonas la imagen se deforma, lo cual conlleva un incremento de píxeles y un mayor esfuerzo para la tecnología existente de codificación y compresión. La proporción estándar de la proyección ERP es de 2:1, es decir, el ancho de la imagen es el doble que su altura.

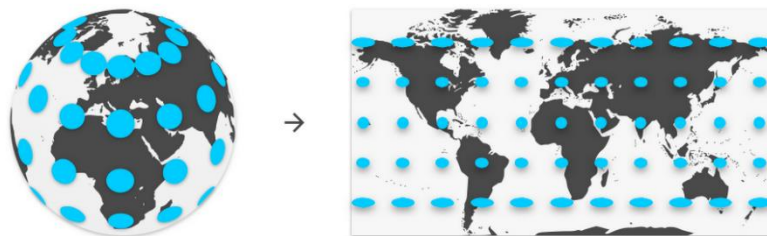


FIGURA 6: PROYECCIÓN ERP

FUENTE: <https://blog.google/products/google-ar-vr/bringing-pixels-front-and-center-vr-video/>

La proyección CMP se obtiene situando un cubo alrededor de la esfera y proyectando los píxeles de la esfera en las 6 caras internas del cubo [11]. Después se despliegan las 6 caras y se organizan de una forma compacta, algunos ejemplos de organización son la distribución T (ver FIGURA 7) o la distribución sugerida por Google y Youtube, EAC (Equi-Angular Cubemap) (ver FIGURA 8).

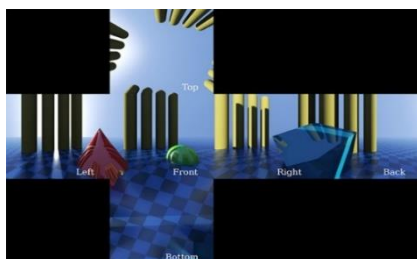


FIGURA 7: CMP

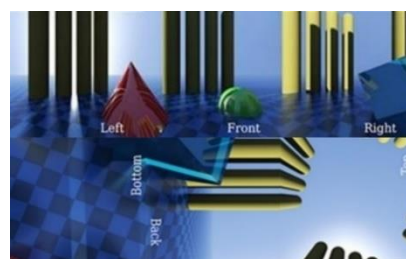


FIGURA 8: EAC

FUENTE: <https://docs.unity3d.com/Manual/VideoPanoramic.html>

Esta proyección no presenta distorsión y por tanto brinda una mejor eficiencia. En [12] se afirma que la proyección CMP reduce un 25% el tamaño del archivo respecto a la proyección ERP. Además, la proyección EAC, mejora la distribución de píxeles consiguiendo más uniformidad [13].

La plataforma de videos online de YOUTUBE utiliza la proyección CMP pero en la versión mejorada EAC en sus videos 360°.

La proyección piramidal fue propuesta por Facebook [12], y consiste en colocar una pirámide alrededor de la esfera y proyectar los píxeles sobre las caras de la pirámide, pero en este caso la base de la pirámide será el FoV y se codifica a resolución completa y los lados de la pirámide reducirán gradualmente su calidad hasta alcanzar el pico (ver FIGURA 9).

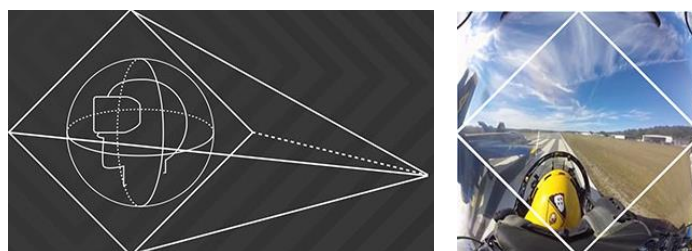


FIGURA 9: PROYECCIÓN PIRAMIDAL [12]

En [12] se afirma que la proyección piramidal reduce un 80% el tamaño del archivo frente a ERP, pero con un coste de almacenamiento excesivo que lo hace inviable. Esto se debe a que para un contenido 360, se generan 30 vistas diferentes con el fin de cubrir los diferentes puntos de vista que pueda escoger el usuario. Además, para cada una de esas 30 vistas, se generan 5 calidades diferentes para el *streaming* adaptativo, obteniendo finalmente un total de 150 versiones para el vídeo omnidireccional.

Respecto a los estándares de resolución de los vídeos 360°. Estos están basados en las resoluciones de vídeo tradicional, pero con relación de aspecto 2:1. Algunas de las resoluciones para vídeos 360° se muestran en la TABLA 2:

RESOLUCIÓN	2:1
8K	7680x3840
4K	4096x2048
4KHD (QuadHD)	3840x1920
2K	2048x1024
	1920x960
	1280x640

TABLA 2: ESTÁNDARES DE RESOLUCIÓN 2:1 DE VÍDEOS 360°

FUENTE: <http://blog.chameleondg.com/post/111891072017/resolution-aspect-ratio-cheat-sheet>

Es importante resaltar que la resolución del vídeo completo no se corresponde con la resolución que percibirá el usuario. Como se ha indicado anteriormente, el usuario solamente visualizará una pequeña sección correspondiente al FoV. Por tanto, no se está viendo la resolución completa del vídeo, sino solamente un trozo de esa resolución.

Con una resolución horizontal total de 4K (correspondiente a una visión total de 360°) y un campo de visión de 120° (tercera parte de 360°), el usuario verá el vídeo a una resolución horizontal de 1,3K (tercera parte de 4K) [10], (ver FIGURA 10).

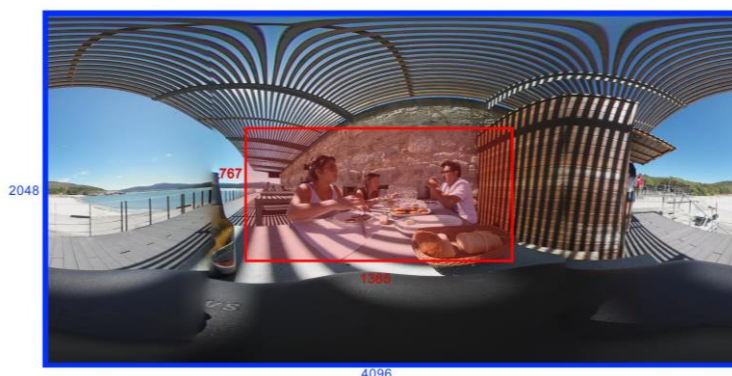


FIGURA 10: RESOLUCIÓN TOTAL FRENTE A RESOLUCIÓN DE VISIÓN [10]

Si se quiere disfrutar de una resolución real de 4K, entonces el vídeo debe estar a 12K (12288x6144 píxeles), y surge el problema de encontrar un reproductor que sea capaz de reproducir vídeos con esa resolución [10].

En este TFG, por una parte, se ha decidido utilizar la proyección cúbica ya que proporciona una reducción en el tamaño del archivo, otro motivo es que Youtube utiliza esta proyección y cuenta con vídeos con resoluciones de hasta 8K, por tanto, se han utilizado vídeos de esta plataforma. Por otra parte, se utilizan vídeos con una resolución total máxima de 4KHD (3840x1920) debido a que la mayoría de los dispositivos (ordenadores, *tablets* y *smartphones*) solamente son capaces de procesar y reproducir con fluidez vídeos con esta resolución.

2.2 Sincronización entre dispositivos (IDES)

IDES (*Inter-Device Synchronization*) consiste en la reproducción o representación de los mismos o diferentes elementos multimedia en diferentes dispositivos de manera sincronizada, es decir, cada elemento es representado en un instante de tiempo determinado o calculado.

Para una sincronización entre dispositivos, donde se sincronizan transmisiones de elementos multimedia en múltiples dispositivos se debe conseguir una precisión a nivel de *frame*, sobre todo cuando hay transmisiones de audio y se van a reproducir desde diferentes altavoces, es decir, una precisión de 40ms para vídeos de 25fps [14].

Pero en un escenario donde se está visualizando el contenido principal en el televisor real y contenido adicional en los dispositivos secundarios, lo ideal es solamente escuchar el audio del televisor real ya que es la fuente principal, por tanto, se puede optar por disminuir la precisión hasta valores de sincronización labial (*lip-sync*) del orden de ± 80 ms [15].

En este TFG se utilizarán vídeos de 25 fps, y se aceptarán valores de asincronía de hasta ± 80 ms, lo cual supone una diferencia máxima de 2 *frames*. Este umbral también evita que se produzcan ajustes continuos de la reproducción multimedia en la aplicación web, que producirían una disminución en la calidad de experiencia del usuario. Si el desfase supera los 80ms, se deberán realizar ajustes en el reproductor web, más adelante en el apartado 4.1.3.2.3 se explicará en qué consisten dichos ajustes.

Respecto a técnicas de sincronización y sincronización entre dispositivos, en [14] presentan un entorno donde un televisor recibe contenido vía broadcast y un *smartphone* o *tablet* (dispositivo secundario) recibe el mismo contenido, pero en mayor resolución vía *broadband* empleando *streaming* adaptativo utilizando HLS (HTTP *Live Streaming*) [16] y MPEG-DASH. Por una parte, los autores analizaron y afirmaron que

los valores PTS (*Presentation Time Stamp*) y DTS (*Decoding Time Stamp*), los cuales forman la base del mecanismo de sincronización entre flujos del contenedor MPEG-TS (*MPEG Transport Stream*), solamente son adecuados para la sincronización entre flujos presentes en el mismo múltiplex, debido a que estos valores podrían ser modificados por los sistemas intermedios y, por tanto, no se recibirían los valores establecidos por el proveedor de servicios. En dicho trabajo optan por utilizar los llamados paquetes SAD (*Synchronized Auxiliary Data*) de DVB, de tal forma que asignan un tiempo absoluto con la siguiente forma (00:00:00) a los valores PTS/DTS. El problema es que estos paquetes forman otro flujo elemental PES (*Packetized Elementary Stream*) y, por tanto, se ocupa más ancho de banda. Por otra parte, en dicho trabajo solamente se puede establecer una conexión *WebSocket*, y para el descubrimiento de dispositivos en la red local, se utiliza la combinación de mDNS (*multicast DNS*) y DNS-SD (*Domain Name System - Service Discovery*). La aplicación para el dispositivo secundario está diseñada para iOS (iPhone OS) y, por tanto, es necesaria la instalación de dicha aplicación en el terminal.

En [17] se presenta una serie de estándares para la sincronización entre flujos y entre dispositivos. Por una parte, también llegan a la conclusión de que las marcas de tiempo PTS/DTS no son adecuadas para la sincronización entre dispositivos y, por otra parte, deducen que utilizar un flujo PES dedicado para insertar marcas de reloj podría ser muy costoso en términos de ancho de banda. Debido a que se debe utilizar, al menos, un paquete TS (*Transport Stream*) que consta de 188 bytes y enviando el reloj con cada *frame*, se llega a 75 Kb/s para una señal de 50 Hz. Dicho valor es superior al ancho de banda de algunas transmisiones de audio y, por tanto, es inviable. Como solución explican que el grupo MPEG (*Moving Picture Experts Group*) ha definido una extensión a MPEG-2 TS, que permite el transporte de medios extrínsecos, es decir, medios externos a la naturaleza del flujo, junto con otras características, bajo el nombre de TEMI (*Timing and External Media Information*). TEMI proporciona diferentes formas de definir la línea de tiempo extrínseca y se puede insertar antes del inicio de cada *frame* en el mismo paquete TS, con lo que es posible reducir la señalización a 4 Kb/s.

En [18] se presenta el desarrollo de un prototipo de televisor híbrido y una aplicación web en un dispositivo secundario, también denominado CS (*Companion Screen*), ambos integrando la especificación DVB-CSS. Dicha especificación dicta las pautas para el descubrimiento del televisor híbrido en una red local, envío de datos y sincronización entre dispositivos. Utilizan la tecnología *WebSocket* para la intercomunicación entre el televisor híbrido y el dispositivo secundario. Después del descubrimiento, el CS estima y configura su reloj realizando consultas al televisor híbrido, asumiendo simetría en los retardos de ida y vuelta de las consultas.

En [19], se presenta una plataforma híbrida de extremo a extremo, es decir, desde la distribución de contenidos hasta el consumo por parte del usuario final. Dicha plataforma se desarrolló por miembros del grupo de investigación IIM en el que se desarrolla el presente TFG. Su diseño es compatible con la versión 2.0 de HbbTV (actualmente en desuso ya que dicha versión ha sido remplazada por la v2.0.1). En este trabajo, cuando codifican el contenido a MPEG-TS, insertan marcas de tiempo extrínsecas utilizando TEMI y también insertan una URL para señalar contenido adicional al transmitido vía broadcast. De esa manera en el televisor híbrido principal, también denominado MS (*Main Screen*), se pueden descodificar dichos datos y enviarlos a los CSs dentro del propio televisor o a dispositivos externos. Para la sincronización, además, se envía periódicamente a los CSs mediante *WebSocket*, dos marcas de tiempo, una es el tiempo de contenido que actualmente está reproduciendo el televisor principal y la otra es el tiempo de reloj actual, ambos representados en formato NTP. Para el descubrimiento del MS se utiliza el protocolo DIAL [20]. La aplicación que se desarrolló para los

dispositivos secundarios (CSs) está programada para Android, por lo que es necesario su instalación en el dispositivo y solamente es capaz de reproducir contenidos en 2D.

2.3 Reproducción de vídeo omnidireccional

Respecto a sistemas y aplicaciones que permiten el consumo de contenido inmersivo y vídeo omnidireccional, pero que no soportan sincronización entre dispositivos, en [21] se puede encontrar el reproductor Omnivirt, en donde los vídeos que se deseen consumir se deben almacenar en su propia nube. Su uso es gratuito, pero está restringido a ciertas limitaciones (p.ej. una capacidad de almacenamiento máxima de 2 GB o un acceso máximo mensual de 10K visitas). Dicho reproductor emplea la proyección ERP y hace uso de *streaming* adaptativo (HLS y MPEG-DASH). Para su utilización en dispositivos HMD (*Head Mounted Display*), este reproductor habilita la visión estereoscópica, aunque no realiza un desplazamiento de las imágenes mostradas en cada ojo. Este desplazamiento sirve para simular cierta sensación de profundidad. El reproductor JW Player [22], también es compatible con el *streaming* adaptativo (HLS y MPEG-DASH), el tipo de proyección soportado es la ERP y ofrece un modo estereoscópico con desplazamiento de las imágenes mostradas en cada ojo. Además, aloja los contenidos en sus propios servidores y el servicio tiene un coste asociado. El reproductor que ofrece Bitmovin [23] permite alojar el contenido en cualquier servidor, aunque se requiere de una licencia para utilizar el reproductor. También permite reproducir el contenido vía *streaming* adaptativo (HLS y MPEG-DASH) y ofrece visión estereoscópica con desplazamiento de la imagen en cada ojo. Solamente soporta la proyección ERP.

Todos los reproductores descritos en el párrafo anterior no requieren ningún tipo de instalación por parte del usuario, puesto que están basados en tecnologías web. Sin embargo, existen otros reproductores que sí requieren de una instalación previa, como, por ejemplo, el VLC Player [24], que es gratuito para el usuario. Sin embargo, no ofrece un modo estereoscópico. Dicho reproductor, también soporta solamente la proyección ERP y ofrece compatibilidad con *streaming* adaptativo (HLS y MPEG-DASH). De forma similar, fabricantes de cámaras de vídeo 360°, ofrecen sus propios reproductores para el consumo de dicho tipo de contenidos, como es el caso de GoPro VR [25]. Dicho reproductor es gratuito y solamente puede reproducir contenidos del explorador de archivos. Es compatible con la proyección ERP, pero no soporta *streaming* adaptativo, ni ofrece el modo de visión estereoscópica. La plataforma de vídeos online Youtube [26] también ofrece un reproductor de vídeos 360° gratuito que utiliza *streaming* adaptativo (HLS y MPEG-DASH). En este caso, es necesario subir los vídeos a la propia nube de Youtube y en los dispositivos móviles es necesario instalar su aplicación móvil para poder consumir los contenidos. Dicho reproductor proporciona visión estereoscópica con desplazamiento de la imagen en cada ojo y solo admite la subida de vídeo 360° con proyección ERP, aunque internamente se convierte a CMP.

Además, cabe destacar que también existe una gran variedad de fabricantes de dispositivos HMD que incorporan sus propios escenarios virtuales dentro de los cuales se puede consumir vídeo omnidireccional. Algunos ejemplos son los de Samsung Gear VR [27] o HTC Vive [28]. El reproductor Samsung VR es gratuito, pero se debe adquirir el HMD Oculus Gear VR y un *smartphone* de Samsung compatible, proporciona el modo de visión estereoscópica con desplazamiento de la imagen en cada ojo, pero no implementa *streaming* adaptativo. La interacción con los elementos del entorno virtual se realiza mediante un mando o mediante el sensor de clics integrado en el propio HMD. La proyección que soporta es la ERP. De la misma manera, el reproductor de HTC Vive ofrece un escenario VR para el consumo de vídeo omnidireccional, pero es necesario adquirir el HMD de HTC que debe estar conectado a un ordenador de elevadas

prestaciones, lo cual supone un coste asociado. Soporta *streaming* adaptativo y es compatible con la proyección ERP.

Cabe resaltar que todos los reproductores y dispositivos descritos anteriormente, son compatibles con contenidos de una resolución 4K y, cuando se utilizan a través de un HMD o *smartphone*, no permite la interacción sin manos (p.ej. utilizando punteros y temporizadores para ejecutar algún evento). A continuación, en la TABLA 3, se muestra una comparativa con el reproductor que se propone en este TFG.

Reproductor	Reproduce Vídeo 360°	Compatible con HbbTV	Basado en tecn. web	Requiere instalación	4K	Acepta contenido de otros servidores o explorador de archivos	Streaming Adaptativo
TFG	✓	✓	✓	X	✓	✓	✓
TNO [6]	X	X	X	✓	✓	✓	✓
BBC [8]	X	✓	✓	X	✓	✓	X
IIM [9]	X	✓	X	✓	✓	✓	✓
Omnivirt	✓	X	✓	X	✓	X	✓
JW player	✓	X	✓	X	✓	X	✓
Bitmovin	✓	X	✓	X	✓	✓	✓
VLC player	✓	X	X	✓	✓	✓	✓
GoProVR	✓	X	X	✓	✓	✓	X
Youtube	✓	X	✓	✓	✓	X	✓
Samsung Gear VR	✓	X	X	✓	✓	X	✓
HTC	✓	X	X	✓	✓	X	✓

Reproductor	Modo estereoscópico	Admite ERP	Admite CMP	Proporciona escenario VR 360	Interacción con ratón (PC) o táctil (Tablet)	Interacción sin manos (HMD o móvil)
TFG	✓	✓	✓	✓	✓	✓
TNO [6]	X	X	X	X	✓	X
BBC [8]	X	X	X	X	✓	X
IIM [9]	X	X	X	X	✓	X
Omnivirt	✓	✓	X	✓	✓	X
JW player	✓	✓	X	X	✓	X
Bitmovin	✓	✓	X	X	✓	X
VLC player	X	✓	X	X	✓	X
GoProVR	X	✓	X	X	✓	X
Youtube	✓	✓	✓	X	✓	X
Samsung Gear VR	✓	✓	X	✓	✓	X
HTC	✓	✓	X	✓	✓	X

TABLA 3: COMPARATIVA ENTRE LOS REPRODUCTORES 360 E HÍBRIDOS EXISTENTES Y EL PROPUESTO EN ESTE TFG

3 TECNOLOGÍAS, ESTÁNDARES Y ESPECIFICACIONES UTILIZADAS

En esta sección se van a describir las tecnologías, estándares y especificaciones que se utilizaron para programar el entorno inmersivo capaz de reproducir vídeos 360° y compatible con el estándar HbbTV 2.0.1.

3.1.1 HbbTV v2.0.1

Esta versión en concreto del estándar HbbTV marca el uso de *streaming* adaptativo para la transmisión de contenidos multimedia, particularmente MPEG-DASH [8]. Para el descubrimiento de dispositivos en la red local se marca el uso del protocolo DIAL (*Discovery And Launch*) [29]. Para el intercambio de mensajes entre el MS y los CS se fija el uso de la tecnología WebSocket. Para la sincronización entre dispositivos se marca el uso de la especificación TEMI (*Timing and External Media Information*) [30] utilizando marcas de tiempo NTP (*Network Time Protocol*) [31]. Y para el desarrollo de aplicaciones híbridas se marca el uso de tecnologías web.

3.1.2 MPEG-DASH

MPEG-DASH (*Dynamic Adaptive Streaming over HTTP*) [8] es una técnica de transmisión adaptativa de contenidos multimedia a través de una red IP (ej. Internet) desde servidores HTTP convencionales. Fue desarrollada por el grupo MPEG y estandarizada por ISO (*International Organization for Standardization*). El concepto adaptativo significa que la calidad del contenido multimedia varía en función de algunos parámetros, por ejemplo, el ancho de banda actual de la red (ver FIGURA 11).

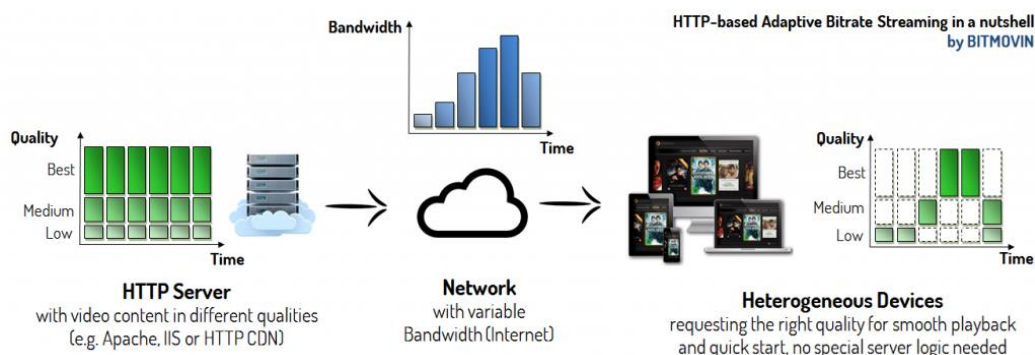


FIGURA 11: ESQUEMA DEL STREAMING ADAPTATIVO DE CONTENIDO MULTIMEDIA

FUENTE: <https://bitmovin.com/dynamic-adaptive-streaming-http-mpeg-dash/>

DASH consiste en generar diferentes representaciones (calidades) del mismo contenido, a su vez, dichas representaciones se trocean en pequeños segmentos (p.ej. de 1s) que son almacenados en un servidor HTTP. Las representaciones y el tamaño de los segmentos, así como la ubicación de los mismos, quedan definidos en un fichero conocido como MPD (*Media Presentation Description*). De esta manera, el reproductor DASH puede conocer las diferentes calidades disponibles y, según el estado de la red, puede optar por descargar un segmento de mayor o menor calidad.

3.1.3 DIAL

DIAL (*Discover and Launch*) es un protocolo simple que permite a las aplicaciones de los dispositivos de segundas pantallas descubrir y lanzar aplicaciones en los dispositivos de primeras pantallas [29]. Fue propuesto e implementado por Netflix y YouTube, en colaboración con Samsung y Sony.

El protocolo DIAL tiene dos componentes:

- DIAL *Service Discovery* que permite a los clientes DIAL descubrir a los servidores DIAL en la red local y obtener acceso al servicio DIAL REST en esos dispositivos.
- DIAL REST *Service* que permite a los clientes DIAL consultar, lanzar y opcionalmente detener aplicaciones en un dispositivo con servidor DIAL.

La parte de DIAL *Service Discovery* se consigue utilizando el protocolo SSDP (*Simple Service Discovery Protocol*) definido por UPnP (*Universal Plug and Play*).

Por una parte, al servicio REST de DIAL se accede mediante HTTP. Por otra parte, según [32] SSDP usa parte del formato de cabecera de HTTP 1.1, sin embargo, no está basado en HTTP 1.1 ya que usa UDP (*User Datagram Protocol*) en lugar de TCP (*Transmission Control Protocol*).

Cuando un cliente DIAL desea descubrir servidores DIAL, debe enviar una solicitud M-SEARCH por UDP a la dirección *multicast* IPv4 239.255.255.250 con puerto UDP 1900 incluyendo en el apartado ST (*Search Target*) de la cabecera el siguiente valor definido por esta especificación: **urn:dial-multiscreen-org:service:dial:1**

A continuación, se observa un ejemplo de petición M-SEARCH:

```
M-SEARCH * HTTP/1.1
HOST: 239.255.255.250:1900
MAN: "ssdp:discover"
MX: seconds to delay response
ST: urn:dial-multiscreen-org:service:dial:1
USER-AGENT: OS/version product/version
```

Un servidor SSDP/UPnP que recibe una petición M-SEARCH con ST debe responder como mínimo con la siguiente información:

- El encabezado LOCATION que debe contener una URL (*Uniform Resource Locator*) para la descripción de UPnP del dispositivo raíz. La parte del host de la URL debe resolverse como una dirección IPv4 o ser una dirección IPv4.
- El encabezado ST de la respuesta debe ser el mismo que el de la petición.

A continuación, se observa un ejemplo de una respuesta a la petición M-SEARCH:

```
HTTP/1.1 200 OK
LOCATION: http://192.168.1.1:52235/dd.xml
CACHE-CONTROL: max-age=1800
EXT:
BOOTID.UPNP.ORG: 1
SERVER: OS/version UPnP/1.1 product/version
ST: urn:dial-multiscreen-org:service:dial:1
```

En la aplicación programada en este TFG solamente se necesita la dirección y el puerto de la URL del apartado LOCATION de la respuesta del servidor DIAL que se ejecuta en el MS, ya que con esa dirección se realizará una conexión WebSocket con el MS para el intercambio de datos.

3.1.4 WebSocket

WebSocket es una tecnología que proporciona un canal de comunicación bidireccional y *full-duplex* sobre un único socket TCP [33]. Se basa en el protocolo ws (websocket) normalizado por la IETF (*Internet Engineering Task Force*) como el RFC 6455. Está diseñada para ser implementada en navegadores y servidores web, pero puede utilizarse en cualquier aplicación cliente/servidor. Se define un prefijo ws:// para indicar una conexión WebSocket.

En una conexión WebSocket intervienen dos partes, un servidor WebSocket y un cliente WebSocket. Una vez establecida la conexión, tanto el servidor como el cliente podrán enviarse mensajes simultáneamente sin necesidad de realizar peticiones.

3.1.5 TEMI

TEMI (*Timing and External Media Information*) [30] es una extensión a MPEG2-TS que permite transportar un reloj extrínseco, es decir, una línea de tiempo adicional [17]. Fue propuesta por DVB y MPEG, y especificada en ISO/IES 13818-1.

TEMI también proporciona herramientas para señalar la URL de uno o varios contenidos adicionales para reproducirlos de forma sincronizada con el flujo broadcast [17].

Las líneas temporales y la información sobre localización de contenidos adicionales se llaman descriptores TEMI, su frecuencia de inserción en el MPEG2-TS puede ser diferente y se pueden insertar de dos maneras [19]:

- En un flujo PES dedicado, el cual se tiene que señalar en la PMT (*Program Map Table*).
- En el campo AF (*Adaptation Field*) de la cabecera de un paquete de transporte (*Transport Packet*) del flujo MPEG2-TS. Esta opción es más eficiente en términos de consumo de ancho de banda.

3.1.6 NTP

NTP (*Network Time Protocol*) [31] es un protocolo diseñado para sincronizar los relojes de los ordenadores a través del encaminamiento de paquetes en una red [34]. NTP utiliza UDP como su capa de transporte, usando el puerto 123. Está diseñado para resistir los efectos de la latencia variable de la red. Las marcas de tiempo que envía un servidor NTP se referencian al 1 de enero del año 1900 y en segundos [31].

3.1.7 HTML5

HTML (*HyperText Markup Language*) versión 5 [35], es la última versión de HTML. HTML es un lenguaje basado en etiquetas que permite crear la estructura y el contenido web.

En esta última versión, se permite la inclusión de contenido multimedia (audio y video) de forma nativa, también proporciona una amplia gama de nuevas características que se ocupan de los gráficos en la web como canvas 2D, WebGL, etc.; y además proporciona una mayor optimización de la velocidad y un mejor uso del hardware.

3.1.8 CSS3

CSS (*Cascading Style Sheets*) versión 3 [36], es la última evolución del lenguaje de estilos que permite definir el diseño visual de los documentos web, e interfaces de usuario escritas en lenguajes de marcado como HTML. Esta última versión trae consigo muchas novedades, como las esquinas redondeadas, sombras, gradientes, transiciones o animaciones y nuevas formas de diseño (*layouts*).

3.1.9 JavaScript

JavaScript [37] es un lenguaje de programación que permite realizar actividades complejas en una página web, por ejemplo, mostrar actualizaciones de contenido, interactuar, animaciones gráficas 2D o 3D, controlar archivos multimedia y muchas otras cosas más.

3.1.10 Three.js

Three.js [38] es una librería 3D basada en JavaScript que trata de hacer lo más fácil posible la creación de elementos 3D en una página web.

A menudo se confunde con WebGL ya que la mayoría de las veces, pero no siempre, se usa WebGL para dibujar en 3D. WebGL es un sistema de muy bajo nivel que solo dibuja puntos, líneas y triángulos. Sin embargo, Three.js ya implementa código que permite crear directamente: escenas, luces, sombras, geometrías, materiales, texturas, matemáticas en 3D, etc.

3.1.11 JSON

JSON (*JavaScript Object Notation*) [39], es un formato de texto sencillo para el intercambio de datos. Se trata de un subconjunto de la notación literal de objetos de JavaScript. Es una alternativa a XML (*eXtensible Markup Language*). Una de las ventajas sobre XML como formato de intercambio de datos es que resulta mucho más sencillo escribir un analizador sintáctico (*parser*) para JSON.

3.1.12 Dash.js

Dash.js [40] es una iniciativa de DASH *Industry Forum*, que permite crear reproductores de video y audio capaces de reproducir contenido MPEG-DASH utilizando librerías JavaScript. Destaca por ser de código abierto, robusto, y es gratis para uso comercial.

3.1.13 Node.js

Node.js [41] es un entorno de ejecución de JavaScript orientado a eventos asíncronos. Fue construido con el motor de JavaScript V8 de Chrome. Está diseñado para construir aplicaciones escalables en red, por ejemplo, servidores web.

3.1.14 Apache HTTP Server

Apache HTTP Server [42] es un software de servidor web gratuito y de código abierto. Es desarrollado y mantenido por una comunidad de usuarios bajo la supervisión de la *Apache Software Foundation*. Entre sus ventajas destaca por ser modular, de código abierto, multiplataforma, extensible y muy popular, ya que es fácil conseguir soporte.

3.1.15 FFmpeg

FFmpeg [43] es una herramienta libre que es capaz de descodificar, codificar, transcodificar, multiplexar, demultiplexar, hacer streaming, filtrar y reproducir archivos multimedia de prácticamente la totalidad de los formatos existentes.

3.1.16 GPAC

GPAC [44] es una herramienta multimedia *open source* que se centra en tecnologías de gráficos y animación, así como en formatos de empaquetado multimedia como mp4.

GPAC proporciona tres herramientas basadas en la librería libgpac:

- Osmo4/MP4Client: Es un reproductor multimedia.
- MP4Box: Sirve para realizar manipulaciones en archivos multimedia como MPEG, TS, etc.
- Otras herramientas y aplicaciones MP42TS.

3.1.17 Bento4

Bento4 [45] es una herramienta desarrollada en C++ y Python que sirve para editar archivos ISO-MP4. Entre sus funcionalidades incluye una librería para generar contenido MPEG-DASH.

4 DESARROLLO DEL ESCENARIO INMERSIVO CAPAZ DE REPRODUCIR VIDEOS 360° Y COMPATIBLE CON EL ESTÁNDAR HbbTV 2.0.1

4.1 Arquitectura de la plataforma híbrida

En la FIGURA 12 se muestra la arquitectura general de la plataforma híbrida utilizada en este TFG, desarrollada con anterioridad por el grupo IIM. La plataforma se ha tenido que adaptar para poder incluir contenido de vídeo omnidireccional en una aplicación web, lo cual se abarca en este TFG (ver área roja de la FIGURA 12).

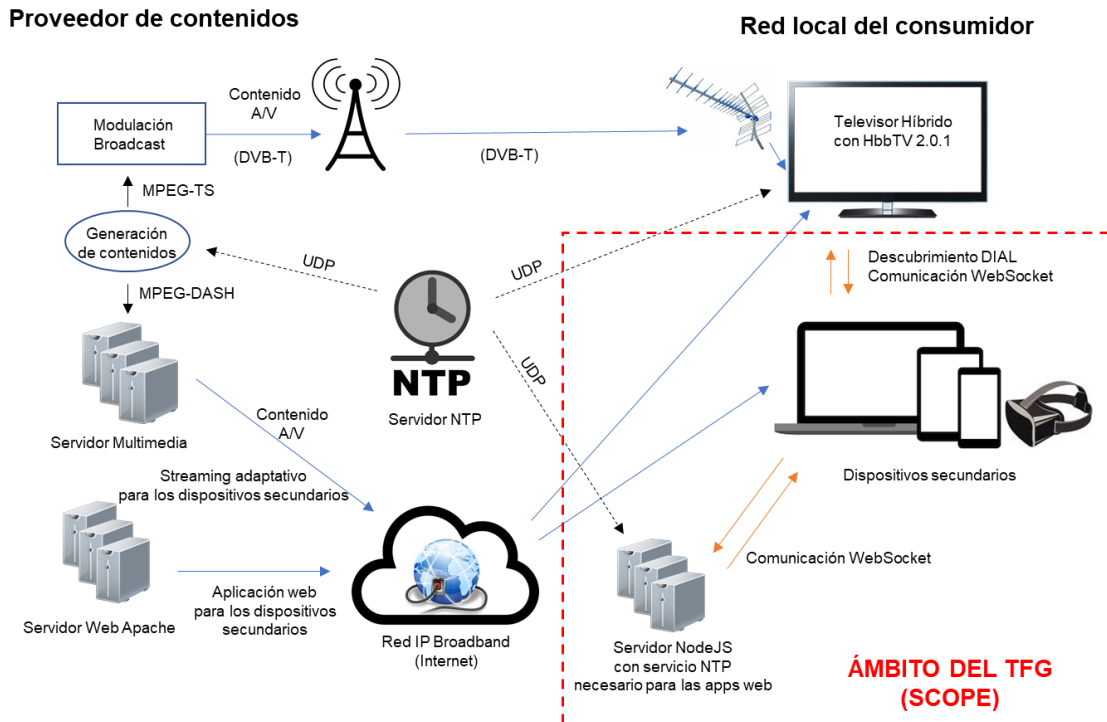


FIGURA 12: ARQUITECTURA GENERAL DE LA PLATAFORMA HÍBRIDA

A continuación, se explicarán cada uno de los componentes, mediante tres partes:

- Proveedor de contenidos
- Aplicación y sistemas para el televisor híbrido (Main Screen) con HbbTV 2.0.1.
- Aplicación web y sistemas para los dispositivos secundarios (*Companion Screens*).

Las partes (a) y (b) ya fueron programadas en [19] dentro del mismo grupo de investigación donde se desarrolló este TFG y no son objeto de este, pero se explicará el uso que se le da para probar la aplicación desarrollada en este TFG.

4.1.1 Proveedor de contenidos

En este subapartado se va a explicar la forma en que se generan los contenidos multimedia, para su difusión tradicional de televisión digital y su transmisión adaptativa por una red IP.

4.1.1.1 Generación del contenido broadcast (MPEG2-TS)

Para este procedimiento se han utilizado las mismas herramientas que en [19]. Básicamente, se utiliza la herramienta FFmpeg para preparar el contenido antes de convertirlo en formato MPEG2-TS.

Para ello se utilizó el siguiente comando FFmpeg, suponiendo que se tiene el archivo de la mezcla final destinada a la señal PGM (*Program*) con nombre f1_PGM.mp4:

```
ffmpeg -i f1_PGM.mp4 -strict experimental -c:a aac -ac 2 -ab 128k -preset slow -codec:v libx264 -profile:v baseline -map 0 -g 25 -r 25 f1_PGM_out.mp4
```

A continuación, en la TABLA 4 se explica cada parámetro del comando FFmpeg anterior.

PARÁMETRO	VALOR	FUNCIONALIDAD
-i	f1_PGM.mp4	Fichero de entrada
-strict	experimental	Permite el uso de codificadores experimentales
-c:a	aac	Codificación de audio AAC
-ac	2	Indica 2 canales de audio
-ab	128k	Bitrate del audio
-preset	slow	Tipo de codificación lenta
-codec:v	libx264	Codificación de vídeo h264
-profile:v	baseline	Perfil h264 recomendado
-map	0	Permite recodificar los flujos indicados y copiar sin modificaciones el resto de los flujos (p.ej., si hubiera subtítulos)
-g	25	Indica un tamaño de GOP de 25
-r	25	Indica una tasa de 25 fps
	f1_PGM_out.mp4	Fichero de salida

TABLA 4: PARÁMETROS FFMPEG PARA PREPARAR EL CONTENIDO ANTES DE CONVERTIRLO A MPEG2-TS

Ahora, procedemos a convertir el archivo mp4 a formato MPEG2-TS, pero también es necesario insertar en el flujo TS la información necesaria para proporcionar el servicio multimedia híbrido sincronizado mediante los mecanismos TEMI explicados en el apartado (3.1.5). Por una parte, hace falta insertar las marcas temporales estables y absolutas, y, por otra parte, también hace falta insertar una URL para señalar la existencia de contenidos *broadband* adicionales.

Para multiplexar contenidos multimedia en MPEG2-TS e insertar las marcas de tiempo y la URL definidas en TEMI, se hace uso de la herramienta MP42TS de GPAC con los siguientes parámetros:

```
MP42TS -real-time -rate 30000 -src=f1_PGM_out.mp4:ID=103 -pcr-init=750 -insert-ntp -temi http://192.168.0.37/index.xml -temi-delay 1000 -dst-file=f1_PGM_final.ts
```

En la TABLA 5 se detallan los parámetros del comando MP42TS anterior.

PARÁMETRO	VALOR	FUNCIONALIDAD
-real-time		El multiplexor funciona en tiempo real.
-rate	3000	Indica la tasa a la que se está generando el contenido TS
-src= :ID	f1_PGM_out.mp4 103	Fichero de entrada y número de programa dentro del multiplex
-pcr-init	750	Valor inicial del PCR
-insert-ntp		Inserta el tiempo NTP en el descriptor de la línea de tiempo TEMI
-temi	http://www.server.com/videos/1/index.xml	Inserta la URL en el descriptor de localización TEMI
-temi-delay	1000	Frecuencia de inserción TEMI cada 1000ms
-dst-file	f1_PGM_final.ts	Fichero de salida

TABLA 5: PARÁMETROS DE LA HERRAMIENTA MP42TS

El siguiente paso es modular el flujo MPEG2-TS creado y emitirlo vía DVB-T. Para ello, en el laboratorio del grupo de investigación IIM se dispone de una tarjeta hardware moduladora DVB-T y el software asociado, denominado DekTec StreamXpress.

Se ha configurado dicho modulador, siguiendo las especificaciones de DVB-T en España, ver TABLA 6.

PARÁMETRO	VALOR
Tipo de modulación	64-QAM
Mod. Jerárquica	NO
Modo	8K
Intervalo de guarda	1/4
R inner-code	2/3

TABLA 6: DVB-T - PARÁMETROS DE MODULACIÓN EN ESPAÑA

4.1.1.2 Generación del contenido broadband (MPEG-DASH)

Como se ha visto en el ejemplo real de uso de la plataforma en apartado 1.1, se tienen diferentes videos en una producción televisiva. Para proporcionar al espectador estos videos y que pueda visualizarlos en un dispositivo secundario conectado a Internet, se hará uso de *streaming* adaptativo, concretamente MPEG-DASH. Para ello, se deben generar los fragmentos de archivos MP4 del contenido en diferentes calidades y el correspondiente archivo MPD. Todo ello se consigue con el software Bento4 utilizando los siguientes scripts que son ejecutables desde la línea de comandos de un ordenador con sistema operativo Linux:

- mp4-dash-encode.py: Se utiliza para generar las diferentes calidades que contendrá el MPD.
- mp4-dash.py: Se utiliza para la fragmentación del archivo MP4 y generación del archivo MPD.

A continuación, se observa un ejemplo del script utilizado para generar el contenido DASH de un video 360° con las resoluciones 3840x1920, 2048x1024, 1920x960, 960x480 y 480x240.

```
#!/bin/bash

./mp4-dash-encode.py -r 3840x1920 -m 40000 -n 50000 -c libx264 f1_4k.mp4
./mp4-dash-encode.py -r 2048x1024 -m 16000 -n 39000 -c libx264 f1_2k.mp4
./mp4-dash-encode.py -r 1920x960 -m 8000 -n 15000 -c libx264 f1_high.mp4
./mp4-dash-encode.py -r 960x480 -m 5000 -n 7000 -c libx264 f1_mid.mp4
./mp4-dash-encode.py -r 480x240 -m 2500 -n 4000 -c libx264 f1_low.mp4

./mp4-dash.py f1_low.mp4 f1_mid.mp4 f1_high.mp4 f1_2k.mp4 f1_4k.mp4
```

La explicación de los parámetros de los comandos del script anterior se muestra en la TABLA 7.

PARÁMETROS mp4-dash-encode.py	FUNCIONALIDAD
-r	Para indicar la resolución que se desea.
-m	Bitrate mínimo aproximado en Kbps
-n	Bitrate máximo aproximado en Kbps
-c	Librería para codificación h264
filename.mp4	Fichero de salida
PARÁMETROS mp4-dash.py	FUNCIONALIDAD
file1.mp4 file2.mp4 file3...	Lista de videos para fragmentar y generar el archivo MPD

TABLA 7: EJEMPLO DE SCRIPT BENTO4 Y EXPLICACIÓN DE LOS PARÁMETROS

4.1.1.3 Estructura del archivo JSON para señalar los contenidos adicionales DASH

En [19] se utiliza un fichero XML para este fin, pero su aplicación Android no es capaz de reproducir videos 360 y si se utilizase este mismo fichero, habría incompatibilidades ya que su reproductor lo tomaría como si se tratara de un video convencional. Como se desea que esta aplicación sea compatible con la diseñada en [19] se ha decidido utilizar otro fichero para indicar la existencia de contenido adicional. Y como el reproductor programado en este TFG está basado en web y sobre todo en JS, se ha optado por utilizar la notación JSON, debido a que JS ya tiene implementados métodos para la interpretación de este tipo de notación. El MS enviará a los CSs la URL insertada en el flujo MPG2-TS que apuntará a un fichero con extensión .xml, por tanto, en esta aplicación simplemente se deberá cambiar la extensión a .json.

A continuación, se muestra la estructura de este archivo que consta de un pack. El pack es un conjunto de contenidos relacionados entre sí. Dentro del pack, en este caso, se tiene el apartado video, pero en futuras implementaciones se podrían tener otros contenidos. Cada vídeo tiene entre sus características detalladas, el nombre (nombreContenido), el *link* para obtener dicho vídeo (url), el *link* para obtener una imagen en miniatura que representará el vídeo (urlPoster) y el tipo de vídeo (tipoVideo) para saber si se trata de un vídeo 2D o un vídeo 360° y poder reproducirlo correctamente.

```
{ "pack": [
  { "video": [
    { "nombreContenido": "contenido0", "url": "www.server.com/f1/vistatv.mpd", "urlPoster": "img/vistatv.jpg", "tipoVideo": "2D"},
    { "nombreContenido": "contenido1", "url": " www.server.com/f1/vista1.mpd", "urlPoster": "img/vista1.jpg", "tipoVideo": "2D"},
    { "nombreContenido": "contenido2", "url": " www.server.com/f1/vista2.mpd", "urlPoster": "img/vista2.jpg", "tipoVideo": "2D"},
    { "nombreContenido": "contenido3", "url": " www.server.com /vista360mpd", "urlPoster": "img/vista360.jpg", "tipoVideo": "360"}
  ]
}
}
```

4.1.2 Aplicación para el televisor híbrido con HbbTV 2.0.1 (Main Screen)

Las funciones que debe realizar el MS son: sintonizar, descodificar y reproducir la señal recibida por antena (*broadcast*), además debe atender a las posibles peticiones que realicen los dispositivos secundarios para que puedan descubrir al MS y que puedan establecer un canal de comunicación. Dicho canal de comunicación servirá para que el MS pueda enviar al CS la información necesaria para que este pueda encontrar contenidos multimedia adicionales y, además, pueda reproducir el contenido multimedia de manera sincronizada respecto al MS.

En la plataforma utilizada en el TFG, presentada en [19], se integró al MS, en primer lugar, un servidor DIAL que permite a los dispositivos secundarios descubrir, de forma transparente para el usuario, la dirección IP del MS, utilizando la herramienta desarrollada por el instituto de investigación alemán FOKUS (*Fraunhofer Institute for Open Communication System*) [46] implementada en Node.js. Este servidor DIAL debe estar continuamente en ejecución en el MS. En segundo lugar, en paralelo al servidor DIAL, se debe ejecutar la aplicación capaz de sintonizar, descodificar, interpretar y reproducir la señal recibida por antena. Esta aplicación se programó con el *framework open source* GStreamer [47] sobre un ordenador con sistema operativo Linux Ubuntu 14.04. Además, según [19] se tuvo que modificar uno de los bloques de GStreamer para poder extraer los elementos TEMI del flujo MPEG2-TS, tanto la URL como las marcas temporales. Finalmente, se dispone de un servicio en ejecución en paralelo al resto de procesos, para atender las solicitudes de conexión WebSocket que realizarán los CSs. Cuando un CS realiza una petición de descubrimiento, el servidor DIAL del MS le responde y en un apartado de la respuesta indica la dirección IP del MS (la suya propia).

Una vez obtenida dicha dirección IP, el CS solicita establecer una conexión con el servidor WebSocket que tenga esa dirección IP, es decir, el MS.

Después de establecer la conexión WebSocket, el MS envía al CS una serie de mensajes. Algunos de ellos son datos del archivo XML descargado con la dirección URL que se extrae del flujo MPEG2-TS.

El primer mensaje es:

`pairingCompleted`

Sirve para indicar que la conexión WebSocket se ha establecido correctamente.

El segundo mensaje tiene el siguiente formato:

```
TEMI_INICIAL\n'  
TEMI_BROADBAND_1;TEMI_BROADBAND_2;TEMI_BROADBAND_3;TEMI_BROADBAND_n; '\n'  
URL_1;URL_2;URL_3;URL_n; '\n'
```

En la primera línea se especifica un valor que en [19] denominan TEMI_INICIAL. Se trata de un valor numérico de 19 cifras, donde los 10 primeros números comenzando por la izquierda se corresponden a segundos y los otros 9 dígitos son decimales. Es un tiempo representado desde el año 1900, es decir, este valor sirve para situar el inicio del contenido reproducido en el MS en una línea temporal desde 1900 hasta la actualidad.

En la segunda línea se especifican los tiempos de inicio de los contenidos adicionales (llamados TEMI_BROADBAND), en la misma escala temporal anterior. Si todos los contenidos adicionales son de la misma producción audiovisual y después de edición comienzan al mismo tiempo que el contenido reproducido en el MS, los tiempos TEMI_BROADBAND serán iguales al valor TEMI_INICIAL.

En la tercera y última línea, se indican las URL's para descargar los contenidos adicionales.

Posteriormente, el MS envía periódicamente, cada 0.5s, el siguiente mensaje:

`INSTANTE_DE_GENERACION INSTANTE_DE_REPRODUCCIÓN`

El valor temporal INSTANTE_DE_GENERACION se corresponde a un valor de 19 cifras, donde de la misma manera, los 10 primeros números comenzando por la izquierda se corresponden a segundos y los otros 9 dígitos son decimales desde el año 1900. Representa el instante en que se generó el contenido que se está reproduciendo en el televisor (MS), calculado a partir del TEMI_INICIAL y los valores PTS del flujo MPEG2-TS convertidos a segundos.

El valor temporal INSTANTE_DE_REPRODUCCION también se corresponde a un valor de 19 cifras, donde de la misma manera, los 10 primeros números comenzando por la izquierda se corresponden a segundos y los otros 9 dígitos son decimales desde el año 1900. Este dato es obtenido del servidor NTP al que consulta la aplicación del MS. Representa el instante en el cual se está reproduciendo en el MS el contenido que se generó en el instante indicado en el valor INSTANTE_DE_GENERACION.

Estos valores permitirán a la aplicación cliente conocer en todo momento el instante del contenido que el MS está reproduciendo y en qué instante NTP.

4.1.3 Aplicación para los dispositivos secundarios (*Companion Screens*)

4.1.3.1 Aplicación para descubrir al MS

En primer lugar, los dispositivos secundarios (PC, *tablet* o *smartphone*) deben ser capaces de descubrir al MS de forma transparente para el usuario, ya que el usuario no tiene por qué conocer datos técnicos del televisor híbrido tales como su dirección IP.

Por tanto, se hará uso del protocolo DIAL, para dicha funcionalidad. Dial utiliza el protocolo SSDP el cual está basado en UDP. Aquí surge un problema y es que, actualmente, los navegadores de páginas web no permiten realizar transmisiones UDP por temas de seguridad.

Como solución se ha optado por programar una aplicación cliente DIAL basada en Android para los *smartphones* y las *tablets*; y otra aplicación basada en Java para los ordenadores (ver FIGURA 13). De tal forma que cuando el usuario ejecute estas aplicaciones en sus dispositivos, la aplicación recibe la IP del MS, y abre el navegador web con la URL de la aplicación web adjuntando un *query string* [48] cuyo formato es (www.....com?name=value). El uso de *query string* permite insertar datos adicionales a la URL sin afectar el comportamiento del servidor, es decir, el servidor no interpreta estos datos adicionales. A continuación, se muestra un ejemplo de la URL con el dato adicional.

```
http://www.domain.com/player360TFG/index.html?address=192.168.0.100:8080
```

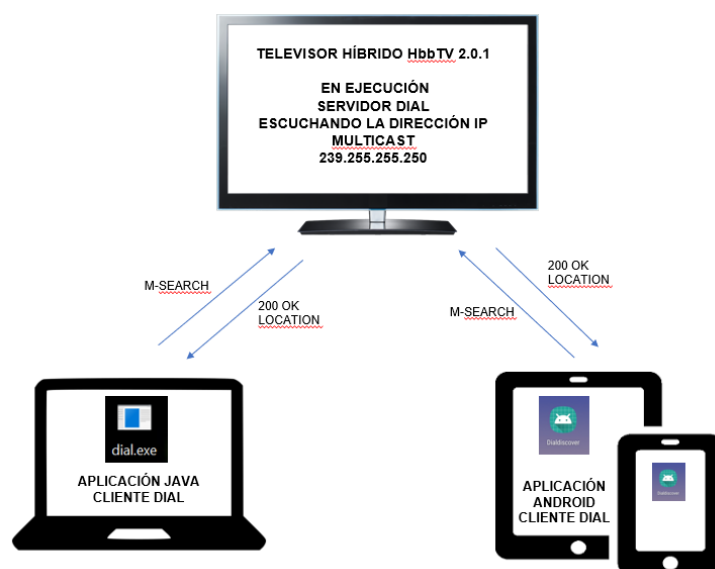


FIGURA 13: DIAGRAMA DEL DESCUBRIMIENTO DIAL

4.1.3.2 Aplicación web que integra el entorno inmersivo capaz de reproducir videos 360

4.1.3.2.1 Implementación de los gráficos iniciales

Como la aplicación debe poder abrirse en un navegador web convencional (p.ej. Chrome o FireFox), se deben utilizar estructuras y lenguajes web. Por tanto, se ha hecho uso de HTML5, CSS3 y JavaScript.

El flujo de trabajo de los procesos iniciales de la aplicación hasta el inicio de la reproducción se muestra en la FIGURA 14.

En primer lugar, en un script del archivo *index* se analiza si la URL de la página que se abre contiene el dato adicional de la IP del MS, si es así se crea el entorno inmersivo

donde posteriormente los videos reproducidos estarán sincronizados respecto al MS; y si la URL no tiene el dato adicional, se abre una aplicación independiente del MS, que se explicará posteriormente.

Al disponer de la IP del MS, entonces es posible crear un canal de comunicación entre el dispositivo secundario y el MS mediante un WebSocket, además de otro canal de comunicación entre el dispositivo secundario y el servidor con servicio NTP.

El servidor con servicio NTP se crea debido a que un servidor NTP utiliza el protocolo UDP para enviar mensajes y los navegadores no permiten la comunicación con dicho protocolo. Este servidor se utiliza como puente. En el mismo se ejecuta un cliente NTP, y a su vez, un servidor WebSocket. Este servidor puente servirá para que el dispositivo secundario pueda solicitar y obtener el tiempo NTP actual.

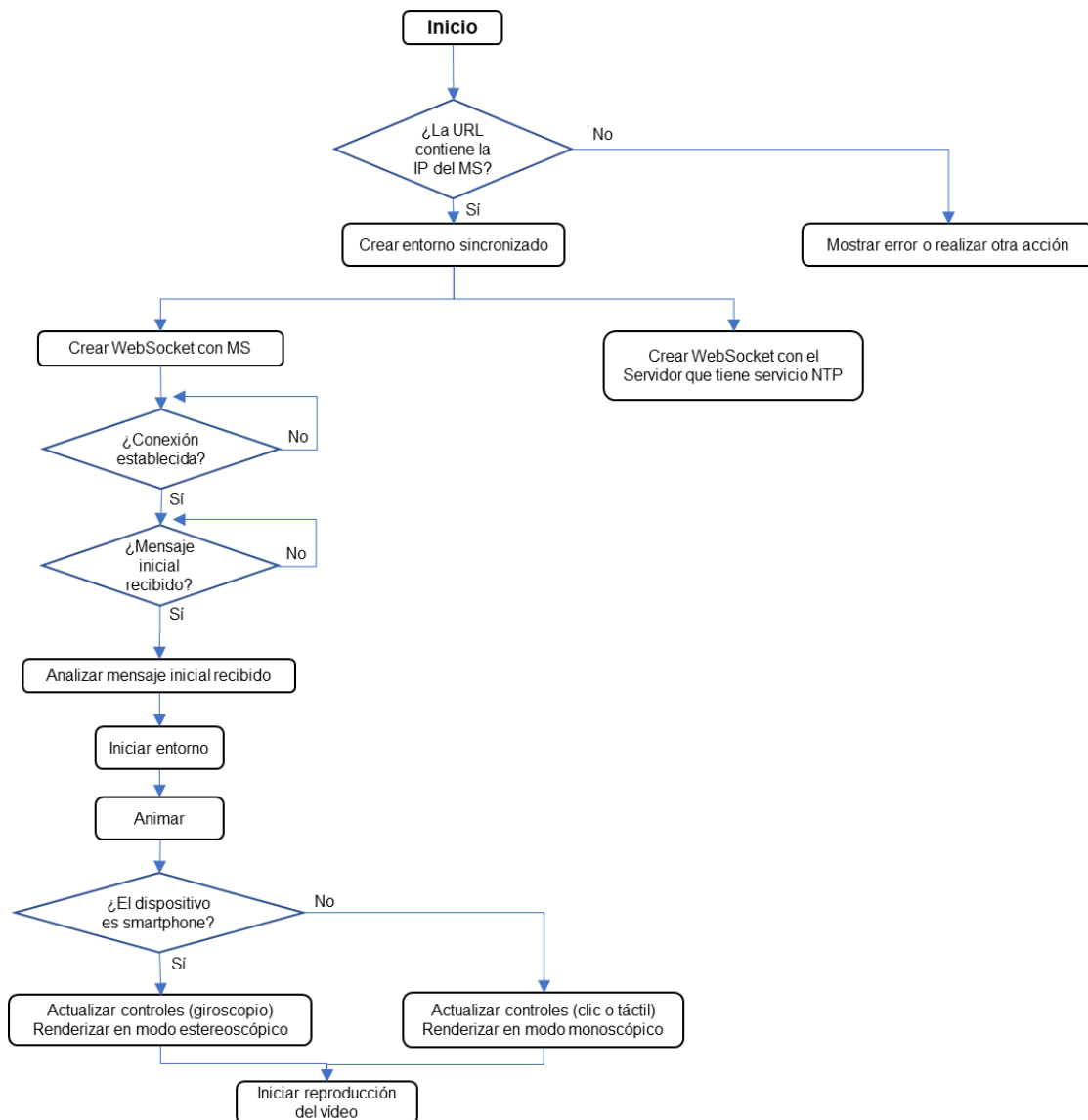


FIGURA 14: DIAGRAMA DE FLUJO INICIAL

La conexión WebSocket con el MS servirá para recibir la información sobre los contenidos adicionales y las marcas temporales comentados en el apartado 4.1.2.

Después que se haya establecido la conexión WebSocket con el MS, inmediatamente se recibe el mensaje inicial y una vez recibido se comienza a crear el entorno inmersivo.

En la **FIGURA 15** se muestra el diagrama de flujos correspondiente al bloque 'Iniciar entorno'.

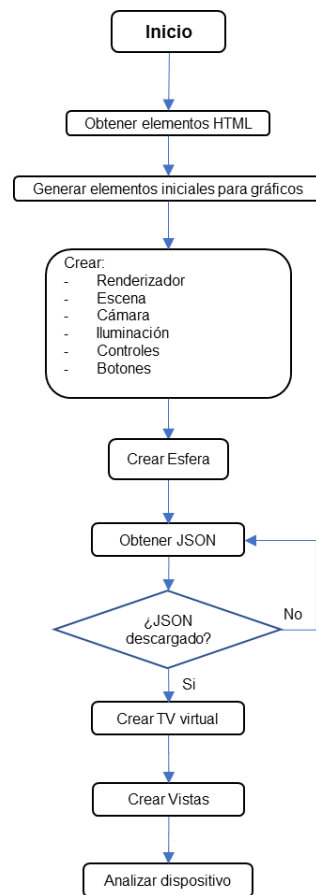


FIGURA 15: DESGLOSE DEL BLOQUE INICIAR ENTORNO

Con JavaScript se obtienen los elementos HTML necesarios, como, por ejemplo, botones o contenedores. A continuación, se crean los elementos iniciales de Three.js para poder renderizar y visualizar los gráficos como, por ejemplo, el elemento renderizador, la escena vacía, una cámara en el eje de coordenadas, etc.

```
//RENDERIZADOR
var rendererA = new THREE.WebGLRenderer();
rendererA.setPixelRatio( window.devicePixelRatio );
//ESCENA THREEJS
var scene = new THREE.Scene();
//CÁMARA
var camera = new THREE.PerspectiveCamera(90, window.innerWidth / window.innerHeight, 0.1, 1000);
camera.position.z = 0.01;
```

Luego se crea la esfera (ver FIGURA 16) donde se plasmará la imagen 360 con formato ERP de un salón de una vivienda.

```
//CREAR GEOMETRÍA ESFÉRICA
var geometry = new THREE.SphereBufferGeometry(500, 60, 40, 2.4, Math.PI * 2, 0, Math.PI);
geometry.scale( -1, 1, 1 );
//CREAR TEXTURA PARA CUBRIR LA ESFERA
var texture = new THREE.TextureLoader().load( 'img/escenas/sala_4k.jpg' );
//CREAR MATERIAL INCLUYENDO LA TEXTURA
var material = new THREE.MeshBasicMaterial({map: texture});
//CREAR LA MALLA DE LA GEOMETRÍA CON EL MATERIAL
var meshEscena = new THREE.Mesh( geometry, material );
//AÑADIR LA MALLA A LA ESCENA
scene.add( meshEscena );
```

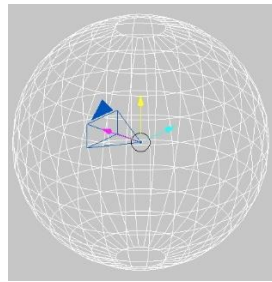


FIGURA 16: MALLA DE UNA GEOMETRÍA ESFÉRICA Y UNA CÁMARA EN SU INTERIOR
FUENTE: <https://stackoverflow.com/questions/41017734/rotate-camera-x-on-local-axis-using-three-js>

A su vez, se procede a descargar el archivo JSON que incluye la información de los contenidos adicionales.

Posteriormente, como ya se dispone de los *links* de descarga se procede a crear el televisor virtual, instanciando el reproductor DASH con la librería *dash.js*, y las miniaturas de las vistas disponibles.

```
var player = dashjs.MediaPlayer().create();  
player.initialize(document.querySelector("#video"), url, true);
```

Inicialmente, en el televisor virtual se carga una imagen que representa el estado *loading* (ver FIGURA 17), ya que al principio se reproduce el vídeo desde el instante cero y transcurre un cierto tiempo hasta que se produce la sincronización. Es por ello que se ha decidido no mostrar la fase inicial de sincronización para mejorar la calidad de experiencia del usuario final.



FIGURA 17: VISTA INICIAL DEL ESCENARIO INMERSIVO CON EL TELEVISOR VIRTUAL EN ESTADO LOADING

Después se procede a analizar algunas características del dispositivo secundario. Se realizaron diversas pruebas y como referencia se utilizó el dispositivo Samsung Galaxy Note 9 el cual es uno de los smartphones más grandes del mercado y además tiene el tamaño más grande que permiten las gafas VR. La lógica programada es la siguiente, si el dispositivo tiene sistema operativo Android o iOS y tiene un ancho de pantalla inferior a 800px y un alto de pantalla inferior a 750px, se considera que el dispositivo es un *smartphone*, pero si supera los tamaños se considera que es una *Tablet*. Si el dispositivo no tiene sistema operativo Android o iOS y supera los tamaños es un PC. Esto servirá para saber si renderizar los gráficos en modo estereoscópico o en modo monoscópico (modo normal).

A continuación, se ejecuta una función que sirve para animar o renderizar los gráficos, esta función se ejecuta en un bucle interno que se itera en cada actualización de la pantalla (en la mayoría de las pantallas 60 veces por segundo) [38]. En esta función se decide si se debe renderizar los gráficos en modo estereoscópico o en modo normal, además de configurar los controles adecuados para cada modo.

En modo normal, el entorno se renderiza en toda la pantalla y se utiliza para las *tablets* y los PC's, los controles se configuran en modo táctil para las *tablets* y con el ratón para los PC's.

El modo estereoscópico se utiliza para los smartphones y consiste en dividir la pantalla en dos partes iguales (ver FIGURA 18). Esto sirve para utilizarlo conjuntamente con unas gafas de realidad virtual en donde cada parte va destinada a cada ojo. El cerebro humano reconstruye las dos partes y da la sensación de presencia en el entorno. Los controles en este caso son mediante el movimiento de la cabeza y situando un punto de referencia sobre algún elemento del entorno con el que se quiera interactuar como, por ejemplo, para accionar los botones o seleccionar las vistas en miniatura se debe girar la cabeza situando el punto de referencia sobre la vista o el botón deseado y después de un intervalo de tiempo se acciona el evento configurado. Se ha decidido establecer un intervalo de tiempo antes de accionar el evento, debido a que es posible que el usuario pase el punto por encima de algún elemento por error, entonces no se accionará el evento instantáneamente.

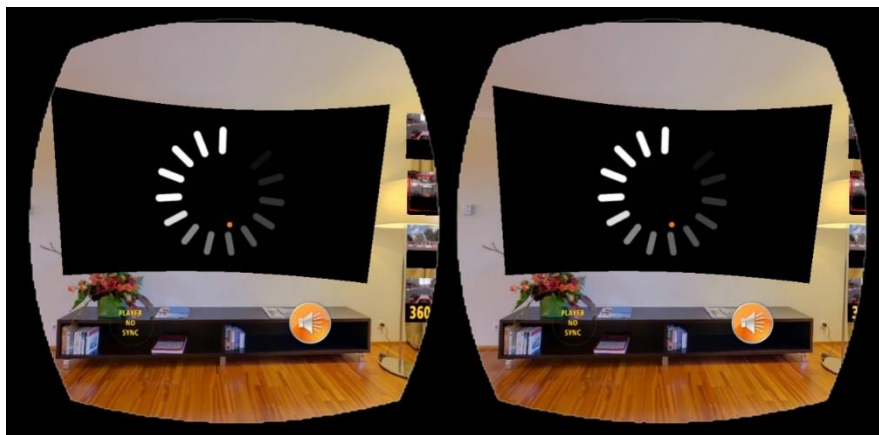


FIGURA 18: MODO ESTEREOSCÓPICO DEL ESCENARIO INMERSIVO

Finalmente se procede a iniciar la reproducción, a su vez se recibe del MS las marcas temporales cada medio segundo y se van actualizando las variables creadas para almacenar dichas marcas temporales.

4.1.3.2.2 Cálculo de la asincronía

En la FIGURA 19, se presenta el diagrama de flujos que permite calcular la asincronía de la reproducción en el dispositivo secundario respecto a la reproducción del contenido en el MS.

Después de iniciar la reproducción, se llama a la función de sincronización, en cada 20 actualizaciones de pantalla, es decir si la pantalla se actualiza 60 veces por segundo, la función de sincronización se ejecuta cada tercio de segundo.

Esto implica leer los instantes almacenados de GENERACIÓN Y REPRODUCCIÓN que fueron enviados por el MS. Después, se solicita el tiempo actual al servidor puente con servicio NTP y se toma el instante actual del vídeo reproducido en el entorno inmersivo que sumado al TEMI_INICIAL forma el instante de GENERACIÓN del CS, y con esos cuatro instantes temporales se calcula el valor de la asincronía respecto al MS utilizando un algoritmo privado diseñado por el grupo IIM de la UPV.

Después de calcular el valor de la asincronía, se procede a realizar la sincronización.

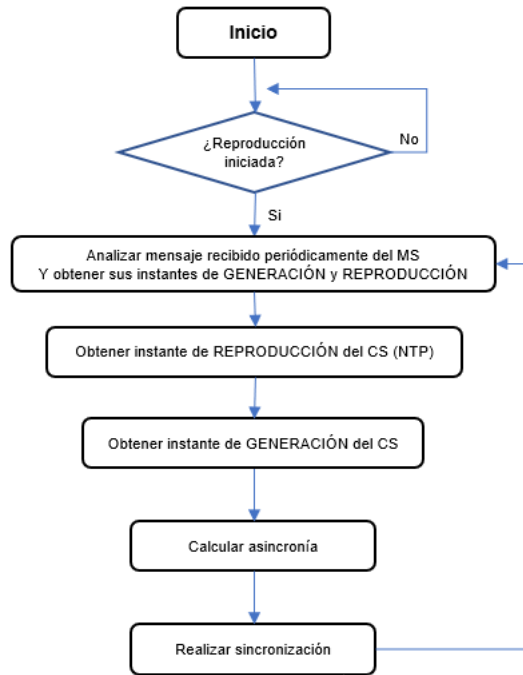


FIGURA 19: DIAGRAMA DE FLUJOS PARA LA CALCULAR LA ASINCRONÍA

4.1.3.2.3 Sincronización

En la FIGURA 20 se muestra el diagrama de flujos para conseguir la sincronización de la reproducción en el dispositivo secundario respecto a la reproducción del contenido en el MS.

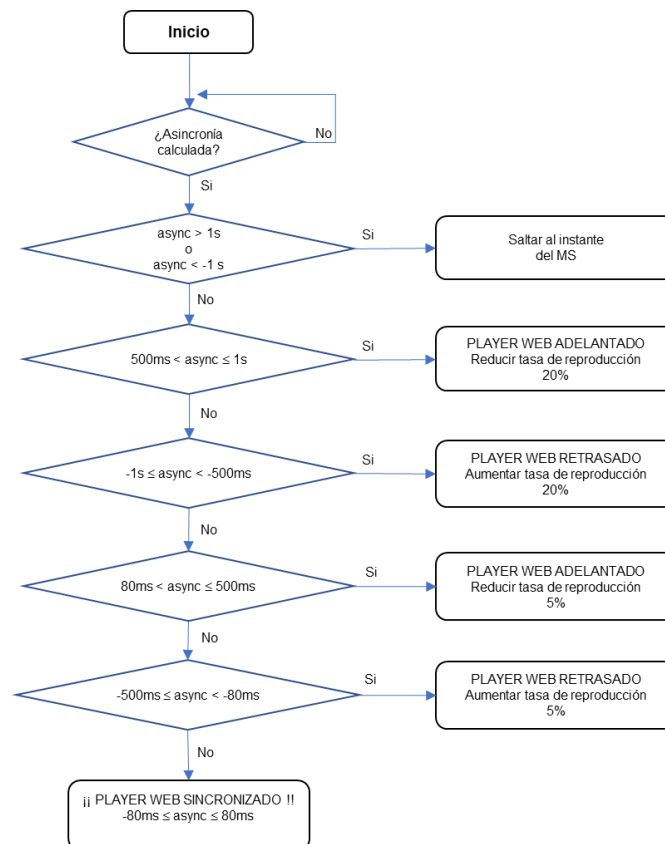


FIGURA 20: DIAGRAMA DE FLUJOS PARA SINCRONIZAR LA REPRODUCCIÓN DEL CONTENIDO

Si el valor de la asincronía es cero significa que el video del entorno inmersivo está sincronizado; si es positivo, el video del entorno inmersivo está adelantado; y si es negativo está retrasado respecto al MS.

Se ha decidido utilizar varios niveles de sincronización y 2 formas de ajustar el reproductor para conseguir la sincronización (ver FIGURA 20). Una forma es utilizando ajustes agresivos (saltos hacia adelante o hacia atrás) y otra forma es utilizando ajustes suavizados por AMP (*Adaptive Media Playout*).

El ajuste agresivo consiste en saltar a un punto del vídeo específico.

El ajuste suavizado por AMP consiste en ajustar, aumentando o disminuyendo, de manera suave y dinámica la velocidad de reproducción de los flujos multimedia para recuperarse de situaciones no deseadas, tales como, desbordamiento del buffer o situaciones de asincronía [49].

Como, al principio, se muestra en el televisor virtual una imagen que representa el estado *loading* del video, hasta que no se calculen varias veces consecutivas el valor de la asincronía y se consiga un valor entre $\pm 500\text{ms}$, el video no se muestra, es decir, el video se muestra solamente cuando mas o menos se haya establecido la sincronización.

4.1.3.2.4 Selección de la vista 360 y creación del reproductor adecuado

Cuando el usuario selecciona una de las vistas en miniatura, excepto las que ponen 360, se carga el respectivo vídeo en el televisor virtual y se reinicia el flujo de sincronización.

Sin embargo, cuando el usuario selecciona una vista en 360°, se crean otros gráficos. A continuación, en la FIGURA 21, se muestra el diagrama de flujos.

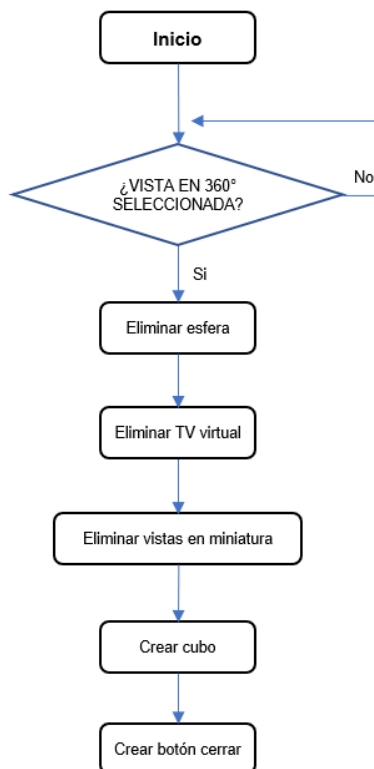


FIGURA 21: DIAGRAMA DE FLUJOS AL SELECCIONAR UNA VISTA EN 360°

Al seleccionar un vídeo en 360°, primeramente, se elimina la esfera donde se plasma la imagen del salón virtual, después se elimina el televisor virtual y, a su vez, se eliminan

las vistas en miniatura. Después, se crea un cubo donde se renderizará el video omnidireccional y, finalmente, se crea un botón para salir del modo 360 y volver al escenario inicial.

La creación del reproductor de vídeos 360° con proyección CMP (orden EAP), no ha sido nada trivial. Primeramente, se debe crear un cubo y mapear en sus caras interiores los *frames* del video omnidireccional con formato cúbico. En el interior del cubo se debe mantener la cámara para poder visualizar el vídeo renderizado en las caras interiores (ver FIGURA 22).

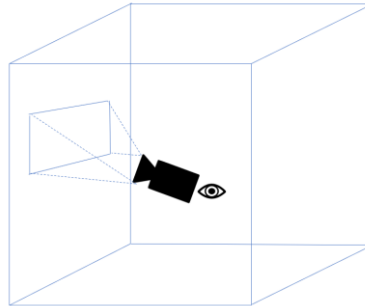


FIGURA 22: GEOMETRÍA CÚBICA CON UNA CÁMARA EN SU INTERIOR

En [50] se ha encontrado un ejemplo de cómo mapear un atlas de texturas en un objeto 3D utilizando la técnica de mapeo UV. Un atlas de texturas es una imagen grande que contiene una colección de imágenes más pequeñas, donde cada una de ellas será una textura para alguna parte del modelo 3D.

El mapeo UV es el proceso de proyectar una imagen 2D en la superficie de un modelo 3D. Las letras UV representan los ejes de la textura 2D y se utilizan esas letras para no confundirlas con las letras X e Y que se utilizan para representar los ejes del objeto 3D en el espacio [51].

En la siguiente imagen se observa la organización de las caras de un *frame* 360 en formato cúbico orden EAP. Además, se observa el mapeo con sus respectivas coordenadas. Los valores de las coordenadas varían de 0 a 1, donde (0,0) es la esquina inferior izquierda y (1,1) es la esquina superior derecha.

(0,1)	(1/3,1)	(2/3,1)	(1,1)
IZQUIERDA	CENTRO	DERECHA	
(0,0.5)	(1/3,0.5)	(2/3,0.5)	(1,0.5)
ABAJO	ATRAS	ARRIBA	
(0,0)	(1/3,0)	(2/3,0)	(1,0)

FIGURA 23: ORDEN DE LAS CARAS DE UN FRAME CMP

Para comenzar hay que crear la geometría cúbica.

```
var lado = 500;  
var geometryCubo = new THREE.BoxGeometry(lado, lado, lado);
```

Después, creamos la textura, y el material que integrará dicha textura.

```
var videoTexture = new THREE.TextureLoader().load( 'img/iconos/loading360.jpg' );
```



```
var materialCubo = new THREE.MeshLambertMaterial({ map: videoTexture });
```

Ahora debemos mapear cada cara según las coordenadas que se observan en la FIGURA 23, según como se quiera ver desde el interior del cubo, es decir, primero la esquina inferior derecha, luego la esquina inferior izquierda, después la esquina superior izquierda y finalmente la esquina superior derecha (ver FIGURA 24).

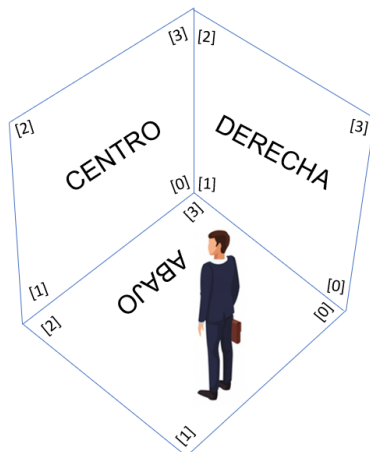


FIGURA 24: VISTA DE LOS VÉRTICES DE CADA CARA DESDE EL INTERIOR DEL CUBO

```
var abajo = [new THREE.Vector2(1/3, 0.5), new THREE.Vector2(1/3, 0), new THREE.Vector2(0, 0), new THREE.Vector2(0, 0.5)];
var atras = [new THREE.Vector2(1/3, 0), new THREE.Vector2(1/3, 0.5), new THREE.Vector2(2/3, 0.5), new THREE.Vector2(2/3, 0)];
var arriba = [new THREE.Vector2(1, 0.5), new THREE.Vector2(1, 0), new THREE.Vector2(2/3, 0), new THREE.Vector2(2/3, 0.5)];
var izquierda = [new THREE.Vector2(1/3, 0.5), new THREE.Vector2(0, 0.5), new THREE.Vector2(0, 1), new THREE.Vector2(1/3, 1)];
var centro = [new THREE.Vector2(2/3, 0.5), new THREE.Vector2(1/3, 0.5), new THREE.Vector2(1/3, 1), new THREE.Vector2(2/3, 1)];
var derecha = [new THREE.Vector2(1, 0.5), new THREE.Vector2(2/3, 0.5), new THREE.Vector2(2/3, 1), new THREE.Vector2(1, 1)];
```

Hay que tener en cuenta que Three.js no admite rectángulos, por lo que cada textura se asigna a dos triángulos para formar cada cara del cubo.

Como cada cara tiene asociado cuatro puntos en un array y se ha seguido el mismo orden de puntos para cada cara (ver FIGURA 24), la asociación de vértices será la misma para todos.

```
//DERECHA
geometryCubo.faceVertexUvs[0][0] = [derecha[2], derecha[1], derecha[3]];
geometryCubo.faceVertexUvs[0][1] = [derecha[1], derecha[0], derecha[3]];
//IZQUIERDA
geometryCubo.faceVertexUvs[0][2] = [izquierda[2], izquierda[1], izquierda[3]];
geometryCubo.faceVertexUvs[0][3] = [izquierda[1], izquierda[0], izquierda[3]];
//ARRIBA
geometryCubo.faceVertexUvs[0][4] = [arriba[2], arriba[1], arriba[3]];
geometryCubo.faceVertexUvs[0][5] = [arriba[1], arriba[0], arriba[3]];
//ABAJO
geometryCubo.faceVertexUvs[0][6] = [abajo[2], abajo[1], abajo[3]];
geometryCubo.faceVertexUvs[0][7] = [abajo[1], abajo[0], abajo[3]];
//CENTRO
geometryCubo.faceVertexUvs[0][8] = [centro[2], centro[1], centro[3]];
geometryCubo.faceVertexUvs[0][9] = [centro[1], centro[0], centro[3]];
//ATRAS
geometryCubo.faceVertexUvs[0][10] = [atras[2], atras[1], atras[3]];
geometryCubo.faceVertexUvs[0][11] = [atras[1], atras[0], atras[3]];
```

Ahora creamos la malla con la geometría y el material anteriormente creados.

```
var meshCubo = new THREE.Mesh(geometryCubo, materialCubo);
```

Finalmente, invertimos las caras del cubo hacia adentro.

```
meshCubo.geometry.scale( 1, 1, -1 );
```

Para crear el botón de cerrar el modo 360, se han utilizado las siguientes líneas de código:

```
//CREAR TEXTURA CON UN ICONO
var spriteMap = new THREE.TextureLoader().load( "img/iconos/exit.png" );
//CREAR MATERIAL Y ASOCIAR LA TEXTURA ANTERIOR
var spriteMaterial = new THREE.SpriteMaterial( { map: spriteMap } );
//CREAR ELEMENTO SPRITE (SE MUESTRA SIEMPRE HACIA LA CÁMARA)
var botonCerrar360 = new THREE.Sprite( spriteMaterial );
```

Ubicar el botón en la escena 3D con coordenadas cartesianas (x,y,z) no resulta intuitivo, por ese motivo se ha utilizado coordenadas esféricas y después se realizó su respectiva conversión a coordenadas cartesianas:

```
//COORDENADAS ESFÉRICAS
var r = 5
var phi =270;
var tetha = 110;
//CONVERSIÓN DE COORD. ESFÉRICAS A COORD. CARTESIANAS
var lx = r * Math.sin( tetha*Math.PI/180 )*Math.cos(phi*Math.PI/180);
var ly = r * Math.cos( tetha*Math.PI/180);
var lz = r * Math.sin( tetha*Math.PI/180 )*Math.sin(phi*Math.PI/180);
//CONFIGURAMOS LA POSICIÓN DEL ELEMENTO CON LAS COORD. CARTESIANAS
botonCerrar360.position.set(lx,ly,lz);
//ASIGNAMOS UN NOMBRE AL ELEMENTO
botonCerrar360.name = "botonCerrar360";
//AÑADIMOS EL BOTÓN A LA ESCENA
scene.add( botonCerrar360 );
```

El atributo *name* sirve para verificar si se ha pulsado o accionado dicho botón cuando se realiza una acción (p.ej. evento clic), y así no confundirlo con el resto de los elementos.

Cuando el usuario selecciona el botón cerrar se reinicia el proceso de creación inicial del escenario inmersivo (ver FIGURA 14), pero se reproduce la última vista 2D seleccionada por el usuario.

4.1.3.2.5 Capturas de la aplicación final y el entorno híbrido sincronizado

A continuación, se observan capturas del entorno híbrido y el escenario inmersivo en los diferentes dispositivos secundarios.

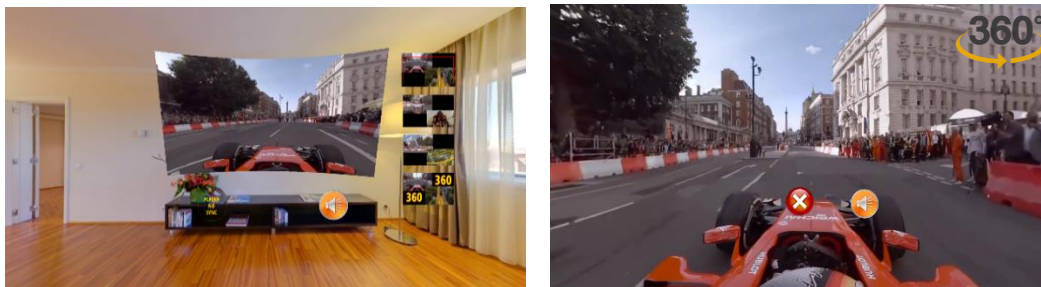


FIGURA 25: MODO MONOSCÓPICO, ESCENARIO INMERSIVO (IZQUIERDA), REPRODUCTOR DE VÍDEO 360 (DERECHA)

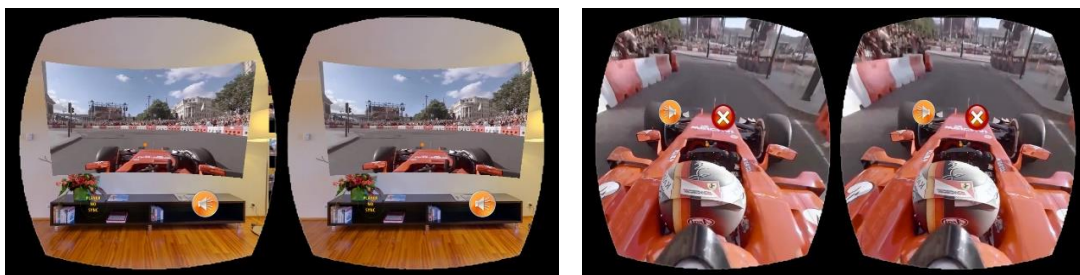


FIGURA 26: MODO ESTEREOSCÓPICO, ESCENARIO INMERSIVO (IZQUIERDA), REPRODUCTOR DE VÍDEO 360 (DERECHA)



FIGURA 27: TELEVISOR HÍBRIDO CON HbbTV 2.0.1 Y DISPOSITIVOS SECUNDARIOS REPRODUCIENDO VÍDEO 360 DE FORMA SINCRONIZADA

4.1.3.3 Aplicación extra

Si el usuario se cansa de visualizar el contenido que se está emitiendo por la televisión convencional, o en caso de no disponer de contenidos adicionales a esa emisión, se le ofrece una versión independiente de la aplicación para poder visualizar vídeos bajo demanda de varias temáticas y también multivista, en 2D y 360° (ver FIGURA 28). Esta aplicación sigue una estructura similar a la aplicación web explicada anteriormente, pero simplificada ya que no necesita los módulos de descubrimiento, comunicación con el MS y sincronización.



FIGURA 28: APLICACIÓN EXTRA

5 EVALUACIÓN

Para la evaluación tanto objetiva como subjetiva de la aplicación desarrollada en este TFG, se han utilizado los dispositivos detallados en la TABLA 8.

DISPOSITIVO	FUNCIÓN
Intel Core i7-6700 @ 3.40GHz CPU, 8GB RAM, W10 with DTA-2111 modulator	Transmisor DVB-T (broadcast)
Intel Xeon E5420, 8GB RAM, Ubuntu 14.04, Fast Ethernet 100	Servidor de archivos multimedia (broadband)
Celeron N3050 @ 1.60GHz CPU, 4GB RAM, Ubuntu 14.04 with DVB USB rtl2832, conectado vía HDMI a un TV LG 32LF592U LED IPS de 32"	MS conectado a TV de 32" marca LG
Tablet Samsung Galaxy Tab S	Dispositivo secundario 1
Móvil Samsung Galaxy S5	Dispositivo secundario 2
Móvil Samsung Galaxy S6	Dispositivo secundario 3
Móvil Samsung Galaxy Note 9	Dispositivo secundario 4
PC portátil w/ Intel Core i7-7700HQ @ 2.8GHz, 16GB RAM, W10	Dispositivo secundario 5
TP-Link AC1900	Wireless Router

TABLA 8: EQUIPOS Y DISPOSITIVOS UTILIZADOS PARA LA EVALUACIÓN

Los contenidos que se utilizaron para la evaluación, tanto objetiva como subjetiva, se detallan en la TABLA 9.

Nº	CONTENIDO	DURACIÓN (min:s)	Nº VISTAS 2D	Nº VISTAS 360	LINK
1	MIX F1 & MONTAÑA RUSA	06:00	3	1	https://www.youtube.com/watch?v=fQoVFraBOnc&t=9s https://www.youtube.com/watch?v=8IsB-P8nGSM&t=89s
2	STEPHEN CURRY NBA	02:23	1	1	https://www.youtube.com/watch?v=Y7TW6XZ_h_0
3	PARÍS	04:31	1	1	https://www.youtube.com/watch?v=E_kshFcLESPU
4	AVIÓN	11:24	1	1	https://www.youtube.com/watch?v=HEE1zZ7UjRg&t=159s
5	TIBURONES	03:18	1	1	https://www.youtube.com/watch?v=i4ZT_XlhZBQ&t=5s
6	CONCIERTO 1 SHAWN MENDES	05:03	2	1	https://www.youtube.com/watch?v=LueM9tEu2wI https://www.youtube.com/watch?v=We7ZKqwgFKU
7	CONCIERTO 2 SHAWN MENDES	04:00	2	1	https://www.youtube.com/watch?v=ttByPvz3XFQ https://www.youtube.com/watch?v=rWx_m1VAp_E

TABLA 9: LISTA DE CONTENIDOS AUDIOVISUALES UTILIZADOS EN LA EVALUACIÓN

Los contenidos 6 y 7 de la lista, disponen de una mezcla final 2D y de un vídeo 360. Pero el resto de contenidos solo están disponibles en vídeos 360, por tanto, en esos casos, se han recortado trozos de los vídeos 360 para la obtención de las vistas 2D, con la herramienta FFmpeg. En el Anexo I se explican los comandos utilizados para el recorte del trozo deseado.

5.1 Evaluación objetiva (QoS)

En la evaluación objetiva, se ha medido el grado de sincronización conseguida entre la reproducción del contenido principal en el MS (TV real) con la reproducción de vídeo 360 en el dispositivo secundario (CS), puesto que era una de las finalidades principales de este TFG.

Como se ha explicado en el punto 2.2 se tolerará un valor de asincronía máximo de 80ms. Por tanto, si la medida de asincronía no supera este valor no se realizará ningún ajuste en el proceso reproductor incluido en la aplicación web.

En la TABLA 10, se muestran los niveles de asincronía y el tipo de ajuste que se utiliza para conseguir la sincronización.

NIVEL	ASINCRONÍA	TIPO DE AJUSTE
3	$> 1s \parallel < -1s$	SALTO
2	$500ms < async < 1s \parallel -1s < async < -500ms$	AMP
1	$80ms < async < 500ms \parallel -500ms < async < -80ms$	AMP
0	$-80ms < async < 80ms$	NINGUNO

TABLA 10: NIVELES DE ASINCRONÍA Y TIPO DE AJUSTE

Para simular un entorno de uso real, se ha configurado una probabilidad de pérdida de paquetes de 0.001% en la transmisión broadcast, lo cual se puede configurar en el software StreamXpress (transmisor DVB-T); y un retardo en la red *broadband* de $60 \pm 20ms$, lo cual se ha configurado con la herramienta Netem [52] ejecutada en el servidor de archivos multimedia.

Para la evaluación, por una parte, se han realizado 10 sesiones de 3 minutos cada una, y se han recogido las mediciones de la asincronía en cada dispositivo secundario. La TABLA 11 resume los resultados obtenidos, incluyendo el valor medio de asincronía y el 95% del intervalo de confianza.

Dispositivo Secundario (CS)	Media de asincronía (ms)	95% Intervalo de confianza (ms)
Samsung Galaxy Tab S	67	± 3
Samsung Galaxy S5	66	± 3
Samsung Galaxy S6	59	± 7
Samsung Galaxy Note 9	62	± 5
PC portátil	60	± 2

TABLA 11: RESULTADOS DE LA MEDIA DE ASINCRONÍA Y EL 95% INTERVALO DE CONFIANZA EN LOS DISPOSITIVOS SECUNDARIOS EVALUADOS

Como se observa en la TABLA 11, la solución de sincronización IDES implementada en este TFG mantiene el valor medio de la asincronía de la sesión por debajo del umbral configurado ($\pm 80ms$). El mejor resultado ha sido obtenido al usar el PC portátil para consumir vídeo omnidireccional. Como era de esperar, los valores de asincronía medidos son mejores en los dispositivos más modernos y potentes, como el Galaxy Note 9, Galaxy S6 y el PC portátil, que, en los dispositivos más antiguos, como el Galaxy S5 y Galaxy Tab S. Sin embargo, las diferencias entre los valores obtenidos en los diferentes dispositivos son mínimas.

Por otra parte, se realizó una sesión adicional en donde se tomaron muestras de todos los dispositivos ejecutando la aplicación simultáneamente. En la FIGURA 29 se muestran las primeras 500 medidas de asincronía tomadas en el dispositivo Galaxy S5 así como los ajustes que se realizaron cuando dichos valores excedían el máximo valor tolerable de asincronía ($\pm 80ms$). Como la reproducción del contenido en el CS siempre comienza desde el instante cero y no es posible que el usuario salte a otro instante, es de suponer que, en un principio, el CS siempre esté retrasado respecto al MS y, por tanto, al principio los valores de asincronía son negativos y elevados. Se puede observar que hasta la muestra 20, aproximadamente, el valor de asincronía va acercándose al umbral tolerado y después, cada vez que hay una asincronía superior a 80ms la aplicación adapta la reproducción hasta obtener un valor de asincronía por debajo del valor máximo tolerable.

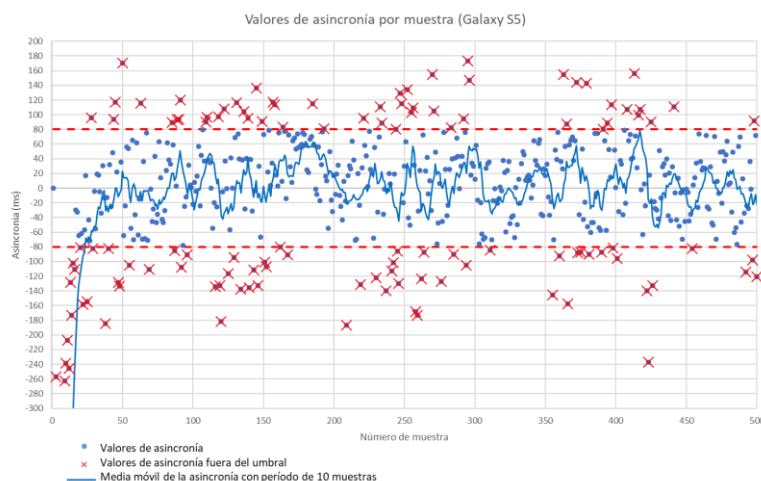


FIGURA 29: VALORES DE ASINCRONÍA POR MUESTRA (GALAXY S5)

En la FIGURA 30 se muestra las medias móviles, tomando grupos de 10 muestras, de las asincronías medidas en todos los dispositivos en la sesión simultánea. Cuando se detectan situaciones de asincronía son inmediatamente corregidas por la solución implementada, manteniendo las medias por debajo del máximo valor tolerable de asincronía (± 80 ms).

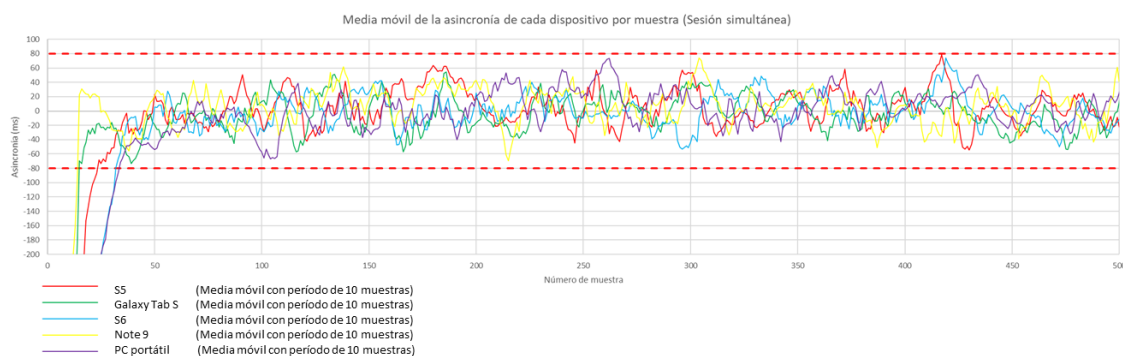


FIGURA 30: GRÁFICA RESULTADO DE UN EXTRACTO DE LA SESIÓN SIMULTÁNEA

Viendo los resultados, se puede afirmar que esta plataforma proporciona una sincronización satisfactoria entre las reproducciones del contenido broadcast (en el MS) y las del contenido de vídeo 360 *broadband* (en dispositivos secundarios o CSs).

5.2 Evaluación subjetiva (QoE)

Además de evaluar objetivamente la asincronía obtenida (parámetros de QoS), en la aplicación diseñada en este TFG, es necesario conocer cómo valoran los usuarios finales la experiencia de uso de dicha aplicación (QoE).

5.2.1 Participantes

Se han evaluado a 30 personas, de las cuales 11 tenían entre 21 y 25 años, 4 entre 26 y 30 años, 4 entre 36 y 40 años, 4 entre 41 y 45 años, y 7 más de 45 años. Respecto al género, el grupo de participantes estaba formado por 17 hombres y 13 mujeres.

En cuanto a la formación adquirida o la que están cursando actualmente, 9 son doctores, 4 tenían un master universitario, 3 habían estudiado formación profesional, 11 participantes están cursando un grado universitario y 3 tenían estudios obligatorios secundarios.

Ninguno de ellos presentaba problemas severos de audición o visión.

5.2.2 Evaluación

Para la evaluación subjetiva, se montaron nuevamente la plataforma y todos los sistemas para utilizar la aplicación y a cada usuario se le ofreció la posibilidad de utilizar una serie de dispositivos para el consumo de contenido multimedia híbrido: una *Smart TV*, un PC portátil, una *tablet* y un *smartphone* integrado en un HMD. El televisor es el dispositivo principal (o de referencia). La reproducción en el resto de dispositivos secundarios estará sincronizada con el televisor. Los dispositivos secundarios podrán reproducir bien el mismo contenido que aparece en el televisor real, o bien contenidos relacionados, pudiendo ser este tanto contenido tradicional (2D) como contenido omnidireccional (360).

Se han preparado una serie de fases, cada una incluyendo la contestación de un cuestionario con preguntas específicas de varios tipos (verdadero/falso, multiopción, de respuesta abierta...).

- Fase inicial de obtención de datos generales sobre los usuarios participantes en el estudio: En esta fase se trata de conocer los datos del usuario, experiencia y uso de tecnologías broadcast (TDT) o tecnologías *broadband* (Video bajo demanda u online), experiencia con gafas de realidad virtual, etc.
- Sesión 1: Prueba de la aplicación extra de manera individual en los tres dispositivos (*smartphone*, *tablet*, PC portátil). Los resultados se resumen en el Anexo III.
- Sesión 2: Elección libre de el/los dispositivo/s deseado/s para seguir probando la aplicación extra. Los resultados se resumen en el Anexo III.
- Sesión 3: Esta sesión dura 6 minutos y es por parejas (usuario A y B). Durante los primeros 3 minutos del vídeo N° 1 de la TABLA 9, el usuario A observa el televisor real y el usuario B utiliza el escenario inmersivo en el *smartphone* con las gafas de RV (HMD). Los siguientes 3 minutos se intercambian los puestos. A continuación, cada uno debe responder cuestiones sobre su experiencia tanto con el televisor real como con las gafas de RV (HMD).
- Sesión 4: Esta sesión también dura 6 minutos y también es por parejas (usuario A y B). Durante los primeros 3 minutos del vídeo N° 1 de la TABLA 9, el usuario A utiliza la *tablet* para evaluar su experiencia de uso y sincronización respecto al televisor real, y el usuario B utiliza el PC portátil para evaluar los mismos aspectos. Los siguientes 3 minutos se intercambian los puestos. A continuación, cada uno debe responder cuestiones sobre su experiencia tanto con la *tablet* como con el portátil.
- Sesión 5: Esta sesión es de fase libre y se le ofrece a la pareja de usuarios la posibilidad de elegir uno de los 3 dispositivos, incluso cambiarlo en el momento que deseen, además de elegir el vídeo que desean visualizar de manera sincronizada en el entorno híbrido. A continuación, cada uno debe responder cuestiones que servirán para conocer la preferencia de dispositivo por parte de los usuarios.
- Test de usabilidad: Cada usuario responde un test para evaluar la usabilidad de la aplicación desarrollada. Se siguen las pautas de [53]. Los resultados servirán para determinar la utilidad y facilidad de uso de la aplicación.
- Fase final: Cada participante debe contestar a un cuestionario final para permitir analizar, posteriormente, la experiencia completa de los mismos.

Antes de iniciar las fases de evaluación, se entregó a cada usuario participante un tutorial de uso de la aplicación (ver Anexo II).

5.2.2.1 Fase inicial de obtención de datos generales

En esta fase se obtuvieron los datos presentados anteriormente sobre la distribución de participantes por edades, género y formación. Además, se obtuvieron otros datos interesantes sobre los participantes que se resumen a continuación.

En cuanto a la cantidad de horas que los usuarios visualizan contenido online (p.ej. Netflix, HBO, Youtube, etc.) y TV tradicional o IPTV (p.ej. TVE, Antena 3, Movistar+, etc.), el 33,3% de los encuestados consume contenido online entre 1 y 2 horas, el 26,7% entre 2 y 3 horas diariamente y el 23,3% menos de 1 hora. Respecto al consumo de TV tradicional o IPTV, el 26,7% entre 1 y 2 horas, el 16,7% entre 2 y 3 horas y el 50% menos de 1 hora. Se puede deducir que hay más consumo de contenido online que de TV tradicional, por ello, hay que mejorar los servicios de TV tradicional.

La FIGURA 31 presenta los dispositivos que poseen y utilizan los usuarios para consumir contenidos multimedia. Se observa que el 93,3% tiene un *smartphone*, el 63,3% portátil y el 46,7% una *tablet*, lo cual significa que estos 3 dispositivos son los más utilizados y, por tanto, se ha acertado en diseñar la aplicación de este TFG para dichos dispositivos.

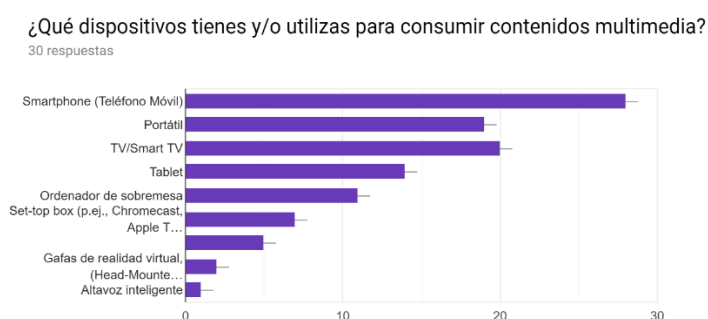


FIGURA 31: DISPOSITIVOS QUE POSEEN LOS ENCUESTADOS

En cuanto a si los participantes tenían experiencia de uso de gafas de realidad virtual, el 10% indica que sí y que, además, dispone de un HMD o más de uno; el 33,3% indica que sí tiene experiencia pero que no posee ningún HMD; y el resto aún no ha tenido experiencia.

Del 43,3% que indicó tener experiencia con HMD, el 76,9% están totalmente de acuerdo en que es un dispositivo que puede enriquecer la experiencia de consumo multimedia; el 15,4% están parcialmente de acuerdo; y, solamente, el 7,7% están en desacuerdo.

5.2.2.2 Sesión 3 (dispositivos: TV y HMD)

5.2.2.2.1 Usuarios con HMD

A los usuarios se les preguntó si al utilizar el HMD se sintieron aislados y no pudieron comunicarse con su compañero. El 53,3% estaban totalmente en desacuerdo; el 36,7% parcialmente en desacuerdo; y, solamente, el 3,3% respondió que estaba totalmente de acuerdo.

Después, se les preguntó si al utilizar el HMD tuvieron la sensación de falta de sincronización respecto a lo que estaba viendo el otro compañero. El 96,7% no tuvo sensación de falta de asincronía, solamente el 3,3% notó esa sensación. El 3,3% equivale a 1 persona y esa persona respondió que notó falta de sincronía entre 2 y 4 ocasiones y, por tanto, el resto del tiempo estaba sincronizado.

También se les preguntó si se han sentido incómodos al tener que moverse físicamente para visualizar las diferentes secciones del vídeo 360°. El 43,3% están totalmente en desacuerdo; el 23,3% parcialmente en desacuerdo; el 16,7% se mostraron neutrales; el

13,3% parcialmente de acuerdo; y, solamente, el 3,3% totalmente de acuerdo. Por tanto, la mayor parte de los usuarios no siente una gran incomodidad al tener que moverse o girar la cabeza para ver las diferentes secciones del vídeo omnidireccional. Incluso el 80% de los usuarios ni siquiera ha tenido que ponerse de pie para dicho fin. El 20% restante se ha puesto de pie en algún instante y, según sus comentarios, fue para poder ver mejor la parte trasera del vídeo, ya que si no tenían que girar mucho la cabeza.

El 36,7% de los encuestados están totalmente de acuerdo en que este tipo de plataformas proporcionan una mejor experiencia de consumo de TV si se comparten con otros usuarios en la misma sala, aunque no todos los usuarios utilicen HMD's. El 33,3% está parcialmente de acuerdo con dicha afirmación; el 26,7% se muestran neutrales; y, solamente, el 3,3% está parcialmente en desacuerdo.

El 83,3% de los participantes valoraría adquirir un dispositivo HMD si este tipo de plataformas estuviera disponible en el servicio de TV ofrecido por las cadenas y operadores. El 10% ya posee un HMD y, solamente, el 6,7% no valoraría adquirirlo. Además, si la plataforma ya estuviera disponible y los participantes dispusieran de un HMD, el 3,3% la utilizaría siempre; el 33,3% la utilizaría casi siempre, y el resto de los participantes la utilizaría a veces.

5.2.2.2.2 Usuarios viendo el televisor real

El 53,3% está totalmente en desacuerdo en que cuando ha estado visualizando sólo el TV real, se ha sentido aislada/o y no ha podido comunicarse con su compañera/o; el 23,3% están parcialmente en desacuerdo con dicha afirmación; el 10% se muestran neutrales; y el 13,3% están parcialmente de acuerdo.

Durante la sesión, el 83,3% de los participantes indica no haber detectado la existencia de asincronía; el 16,7% sí ha detectado, de los cuales el 60% solamente en 1 ocasión y el 40% entre 2 y 4 ocasiones.

Al 50% de los participantes no les ha molestado que su compañero con el HMD se haya movido en exceso; el 46,7% ni se han dado cuenta; y, solamente, el 3,3% se ha dado cuenta y le ha molestado un poco.

5.2.2.3 Sesión 4 (dispositivos: portátil y tablet – TV)

Durante la experiencia con el portátil, el 53,3% de los participantes están totalmente de acuerdo en que han visualizado más las vistas 360 que las vistas en 2D; el 3,3% está parcialmente de acuerdo; el 23,3% se muestran neutrales; el 3,3% parcialmente en desacuerdo; y el 3,3% restante totalmente en desacuerdo.

Durante la experiencia con la *tablet*, el 70% de los participantes están totalmente de acuerdo en que han visualizado más las vistas 360 que las vistas en 2D; el 10% está parcialmente de acuerdo; el 10% se muestran neutrales; el 3,3% parcialmente en desacuerdo; y el 3,3% restante totalmente en desacuerdo.

El 43,3% de los participantes está totalmente de acuerdo en que ha preferido usar el portátil respecto al televisor real; el 26,7% está parcialmente de acuerdo; el 10% se muestra neutral; el 13,3% parcialmente en desacuerdo; y el 3,3% totalmente en desacuerdo.

Respecto a la *tablet* y el televisor real, el 50% de los participantes está totalmente de acuerdo en que ha preferido visualizar la *tablet*; el 30% está parcialmente de acuerdo; el 3,3% se muestra neutral; el 3,3% parcialmente en desacuerdo; y el 3,3% restante totalmente en desacuerdo.

El 86,7% de los participantes no ha notado asincronías cuando estaba visualizando el portátil junto con el televisor real; mientras que el 13,3% si ha notado. De los participantes que han notado falta de sincronía (4), dos solamente la han notado en 1 ocasión, uno entre 2 y 4 ocasiones; y otro entre 5 y 10 ocasiones.

Respecto al uso de la *tablet* y el televisor real, el 90% de los participantes no ha notado falta de sincronía entre ambos dispositivos, sin embargo, el 10% equivalente a 3 personas sí lo han notado, de las cuales, una persona en 1 ocasión, otra persona entre 2 y 4 ocasiones y la otra entre 5 y 10 ocasiones.

5.2.2.4 Sesión 5 – fase libre

En esta fase el 36,7% no ha cambiado ninguna vez de dispositivo, es decir, se quedaron con el que eligieron, el 33,3% cambiaron una sola vez de dispositivo, el 26,7% entre 2 y 3 veces, y el 3,3% entre 4 y 5 veces.

De las personas que han cambiado de dispositivo se destaca el uso del HMD y la *tablet*, con mucha diferencia respecto al portátil. Según las respuestas el HMD porque parece más inmersivo, es más real, resulta más impresionante; y la *tablet* porque resulta más cómodo.

De las personas que solamente han utilizado un dispositivo, el 36,4% ha utilizado el HMD, el 36,4% el portátil, el 27,3% la *tablet*. Según las respuestas, el HMD porque parece que estuvieras dentro de la escena, porque es distinto a lo convencional, la *tablet* porque proporciona una interacción más cómoda, y el portátil por no utilizar las gafas de RV ya que marean.

El 23,3% de los encuestados ha querido visualizar en algún instante de la sesión el televisor real. Según las respuestas, para ver la referencia de la cámara principal, para ver si estaba sincronizado, en el vídeo del concierto cuando la información estaba centrada en el cantante y no en el ambiente. El 76,7% restante no ha querido ver específicamente el televisor real.

El 13,3% de los encuestados ha querido utilizar específicamente el portátil en algún momento de la sesión. Según las respuestas porque están más familiarizado con este dispositivo. El 86,7% restante no ha querido utilizar específicamente el portátil.

El 30% de los encuestados ha querido utilizar específicamente la *tablet* en algún momento de la sesión. Según las respuestas porque era más cómodo y rápido de utilizar. El 70% restante no ha querido utilizar específicamente la *tablet*.

El 50% de los encuestados ha querido utilizar específicamente el HMD en algún momento de la sesión. Según las respuestas porque resulta más interesante la inmersión. El 50% restante no ha querido utilizar específicamente el HMD.

5.2.2.5 Test de usabilidad

Para esta sesión se han preguntado las cuestiones dictadas en [53], para valorar la facilidad de uso de la aplicación. Según los resultados se ha obtenido una puntuación de 92,67 sobre 100 puntos con una desviación estándar de $\pm 13,17$. Según [54] con este resultado la solución implementada puede ser considerada como excelente. Esto significa que la aplicación programada en este TFG resulta intuitiva y fácil de usar.

5.2.2.6 Fase final

El 73,3% de los participantes está totalmente de acuerdo en que el consumo de TV híbrida con contenidos inmersivos ha sido una experiencia totalmente divertida; el 20% está parcialmente de acuerdo con dicha afirmación; el 3,3% están parcialmente en desacuerdo; y el 3,3% restante están totalmente en desacuerdo.

El 26,4% de los participantes están totalmente de acuerdo en que al utilizar el HMD se han sentido tan integrados en la escena que han perdido la noción del tiempo, el 40% están parcialmente de acuerdo, el 23,3% se muestran neutrales, y solamente el 10% están totalmente en desacuerdo.

De todos los participantes, el 16,7% se han sentido totalmente integrados en la experiencia de visualización que han perdido la noción del mundo real, el 36,7% están parcialmente de acuerdo, el 20% se muestran neutrales, el 6,7% están parcialmente en desacuerdo y el 20% totalmente en desacuerdo.

El 86,7% de los participantes están totalmente de acuerdo en que no les importaría repetir la experiencia; mientras que el 13,3% están parcialmente de acuerdo.

El 30% de los participantes están totalmente en desacuerdo en que al utilizar el HMD se han sentido mareadas/os o desorientadas/os; el 23,3% parcialmente en desacuerdo; el 3,3% se muestra neutral; el 26,7% parcialmente de acuerdo; y el 13,3% están totalmente de acuerdo.

El 80% de los participantes están totalmente de acuerdo en que la inclusión de vídeos 360° complementario al contenido de TV mejora la experiencia del usuario; el 16,7% están parcialmente de acuerdo; y el 3,3% se muestra neutral.

Además, el 86,6% de los participantes están parcial o totalmente de acuerdo en que la inclusión de vídeos 360° en los servicios de TV contribuirán a que los usuarios se enganchen a los canales que ofrezcan dicho servicio (ver FIGURA 32).

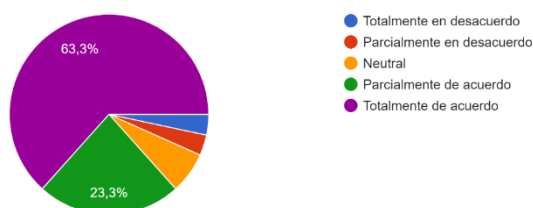


FIGURA 32: RESULTADO RESPECTO A QUE LOS VIDEOS 360 AYUDARÍAN A ENGANCHAR AL USUARIO AL CANAL DE TELEVISIÓN QUE LO OFREZCA

El 43,3% de los participantes indican que con la TV convencional se pierden puntos de vista interesantes.

El 96,6% de los participantes consideran de manera total o parcial que los usuarios aprenderían fácilmente a utilizar este tipo de plataformas en un plazo corto de tiempo.

Finalmente, entre los contenidos que más les interesa a los usuarios utilizando este tipo de plataformas, están los documentales y *reality shows*.

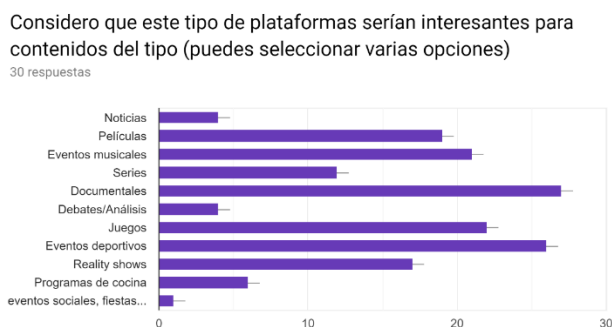


FIGURA 33: CONTENIDOS MÁS INTERESANTES PARA ESTE TIPO DE PLATAFORMAS SEGÚN LOS ENCUESTADOS

6 CONCLUSIONES Y TRABAJO FUTURO

Primeramente, se puede afirmar que los objetivos planteados para este TFG se han cumplido ampliamente, ya que se ha conseguido implementar el entorno inmersivo con capacidad de reproducir vídeos omnidireccionales o 360° y las especificaciones marcadas por el estándar HbbTV v2.0.1 respecto a las funcionalidades de segundas pantallas y sincronización entre dispositivos. Incluso se implementó una aplicación extra que funciona de forma independiente al televisor híbrido, para dar más opciones de consumo de contenido multimedia al usuario.

En segundo lugar, se han realizado evaluaciones objetivas para medir las asincronías que se producen en una sesión entre el televisor híbrido principal y un dispositivo secundario reproduciendo contenidos de vídeo 360. Después de analizar los resultados, se ha comprobado que los valores medios de asincronía no superan el umbral de asincronía permitida configurado (± 80 ms).

En tercer lugar, también se ha realizado la evaluación subjetiva de la aplicación con personas reales, las cuales probaron la aplicación en conjunto con toda la plataforma híbrida y después se les realizó unas preguntas. Después de analizar los resultados, se puede concluir que el entorno presentado resulta muy inmersivo, intuitivo y fácil de usar. Respecto a la sincronización entre el MS y los CSs, la mayoría de los usuarios apenas notó asincronías.

Finalmente, este TFG demuestra que la funcionalidad de segundas pantallas y sincronización entre el MS y diversos CSs resulta interesante para los consumidores de servicios de televisión, además el contenido más atractivo para los usuarios según las encuestas realizadas son los vídeos 360°.

Como trabajo futuro se propone mejorar la aplicación en base a las opiniones que mostraron los usuarios que la probaron, y como este TFG tiene un enfoque comercial a medio plazo, ya que la tecnología avanza rápidamente se plantea optimizar el funcionamiento de la aplicación. Por ejemplo, respecto a los vídeos 360, actualmente se están investigando nuevas formas de transmitir este tipo de contenidos. Una de ellas, conocida como *tiled streaming*, consiste en trocear cada *frame* del video 360 en pequeñas partes (*tiles*) y transmitir solamente las que se encuentren dentro del campo de visión del espectador y alrededores. De esta forma se puede ahorrar en el uso de ancho de banda. Por tanto, como trabajo futuro, se plantea investigar más sobre dicha tecnología y, si es posible, implementarlo en la aplicación para conseguir una mayor eficiencia en cuanto al consumo de ancho de banda.

Actualmente la aplicación no está disponible para los navegadores y sistemas operativos de Apple (Safari, iOS) debido a que tienen otra solución de *streaming* adaptativo denominada HLS, por tanto, como trabajo futuro se propone implementar HLS en la aplicación y así conseguir una mayor compatibilidad.

Respecto a la interfaz gráfica se propone incluir la posibilidad de cambiar el fondo del escenario inmersivo (p.ej. otros salones) a gusto del usuario. Además, también se propone incorporar en los smartphones la posibilidad de visualizar la aplicación en modo monoscópico y no solamente en modo estereoscópico como actualmente está programado.

7 BIBLIOGRAFÍA

- [1] «HbbTV,» Hybrid broadcast broadband TV, 2019. [En línea]. Available: <https://www.hbbtv.org>. [Último acceso: 6 septiembre 2019].
- [2] HbbTV, «HbbTV 2.0.1 Specification,» HbbTV Association, 2011-2018. [En línea]. Available: <https://www.hbbtv.org/resource-library/specifications/>. [Último acceso: 6 septiembre 2019].
- [3] Digital Video Broadcasting, «Companion Screens and Streams. Part 2: Content Identification and Media Synchronization,» ETSI TS 103 286-2 V1.2.1, 2017-08.
- [4] We Are Social Ltd., «We are social and Hootsuite, Global Digital Report 2019,» Enero 2019. [En línea]. Available: <https://wearesocial.com/global-digital-report-2019>. [Último acceso: 6 septiembre 2019].
- [5] YouGov, «VR adoption grows to 11% of US adults,» Marzo 2019. [En línea]. Available: <https://www.digitaltveurope.com/2019/03/22/yougov-vr-adoption-grows-to-11-of-us-adults/>. [Último acceso: 6 septiembre 2019].
- [6] Digital TV Europe, «Digital TV Europe Industry Survey 2019,» 2019. [En línea]. Available: <https://www.digitaltveurope.com/magazine/digital-tv-europe-industry-survey-2019/>. [Último acceso: 6 septiembre 2019].
- [7] Advanced Television Ltd., «Study: Traditional TV far from dead,» 2019. [En línea]. Available: https://advanced-television.com/2019/07/12/study-traditional-tv-far-from-dead/?utm_source=June2019&utm_campaign=9b26bc51a1-EMAIL_CAMPAIGN_2019_07_17_11_21&utm_medium=email&utm_term=0_5b9fcb3582-9b26bc51a1-67822723. [Último acceso: 6 septiembre 2019].
- [8] ISO International Organization for Standardization, «Dynamic Adaptive Streaming over HTTP (MPEG-DASH),» ISO/IEC 23009, 2011.
- [9] WIKIPEDIA, «Vídeo inmersivo,» [En línea]. Available: https://es.wikipedia.org/wiki/Video_inmersivo. [Último acceso: 6 septiembre 2019].
- [10] A. Díaz Amor, «El vídeo 360º monoscópico,» TFG Comunicación Audiovisual - Universidad De Barcelona, 2017.
- [11] Y. L. Y. Z. Zhenzhong Chen, «Recent advances in omnidirectional video coding for virtual reality: Projection and evaluation,» *ELSEVIER*, vol. 146, pp. 66-78, 2018.
- [12] D. P. Evgeny Kuzyakov, «Next-generation video encoding techniques for 360 video and VR,» facebook Engineering, 21 junio 2016. [En línea]. Available: <https://engineering.fb.com/virtual-reality/next-generation-video-encoding-techniques-for-360-video-and-vr/>. [Último acceso: 6 septiembre 2019].
- [13] A. Wheeler, «Engineering and Developers Blog,» Youtube, 14 marzo 2017. [En línea]. Available: <https://youtube-eng.googleblog.com/2017/03/improving-vr-videos.html>. [Último acceso: 6 septiembre 2019].
- [14] A. V. R. van Brandenburg, «Immersive second-screen experiences using hybrid media,» de *Media Synchronization Workshop (MediaSync) 2013, NEM Summit 2013 workshop, October 29, 2013, Nantes, France*, 2013.
- [15] R. Steinmetz, «Human perception of jitter and media synchronization,» *IEEE J. Sel. Areas Commun.*, vol. 14, nº 1, p. 61–72, 1996.
- [16] R. Pantos and E. W. May, «Http Live Streaming,» RFC 8216, 2017.
- [17] H. S. M. H. J. L. F. a. P. C. M. O. van Deventer, «Standards for multi-stream and multi-device media synchronization,» *IEEE Communications Magazine*, vol. 54, nº 3, pp. 16-21, 2016.

- [18] R. R. a. M. H. V. Vinayagamoorthy, «Enabling Frame-Accurate Synchronised Companion Screen Experiences,» de *Proceedings of the ACM International Conference on Interactive Experiences for TV and Online Video*, 2016.
- [19] D. M. Reguero, «Implementación de una plataforma de consumo sincronizado de contenidos multimedia, híbridos (broadcast/broadband) en entornos mono y multi-dispositivo, basada en GStreamer y compatible con el estándar HbbTV 2.0,» TFM Tecnologías, Sistemas y Redes de Comunicaciones - Universidad Politécnica de Valencia, 2015.
- [20] Netflix, «DIAL. Discovery and Launch protocol especificacion Version 1.7.2,» 2015.
- [21] «OmniVirt,» [En línea]. Available: <https://www.omnivirt.com>. [Último acceso: 6 septiembre 2019].
- [22] Longtail Ad Solutions, Inc., «JW Player,» [En línea]. Available: <https://www.jwplayer.com>. [Último acceso: 6 septiembre 2019].
- [23] Bitmovin Inc., «VR & 360°,» [En línea]. Available: <https://bitmovin.com/demos/vr-360>. [Último acceso: 6 septiembre 2019].
- [24] VideoLAN Organization, «VLC media player,» [En línea]. Available: <https://www.videolan.org/vlc/index.html>. [Último acceso: 6 septiembre 2019].
- [25] GoPro, Inc., «GoPro VR Player,» [En línea]. Available: <https://gopro.com/news/gopro-vr-player-2-now-available>. [Último acceso: 6 septiembre 2019].
- [26] Youtube, LLC Youtube, una empresa de Google, «YouTube,» 2019. [En línea]. Available: <https://www.youtube.com>. [Último acceso: 2019 septiembre 6].
- [27] SAMSUNG ELECTRONICS CO., LTD., «Samsung Gear VR,» [En línea]. Available: <https://www.samsung.com/global/galaxy/gear-vr>. [Último acceso: 6 septiembre 2019].
- [28] HTC Corporation, «HTC VIVE,» [En línea]. Available: <https://www.vive.com/eu>. [Último acceso: 6 septiembre 2019].
- [29] Netflix, Inc, «DIAL Discovery And Launch protocol specification Version 1.7,» 2014.
- [30] ISO/IEC, «Delivery of Timeline for External Data,» ISO/IEC 13818-1:2013/PDAM 6, 2013.
- [31] U. D. E. J. M. J. B. a. W. K. D. Mills, «Network Time Protocol Version 4: Protocol and Algorithms Specification,» IETF Internet Standard RFC 5905, 2010.
- [32] Open Connectivity Foundation, «UPnP Device Architecture version 2.0,» [En línea]. Available: <https://openconnectivity.org/developer/specifications/upnp-resources/upnp>. [Último acceso: 6 septiembre 2019].
- [33] WIKIPEDIA, «WebSocket,» [En línea]. Available: <https://es.wikipedia.org/wiki/WebSocket>. [Último acceso: 6 septiembre 2019].
- [34] N. T. Foundation, «NTP: The Network Time Protocol,» [En línea]. Available: <http://www.ntp.org>. [Último acceso: 6 septiembre 2019].
- [35] Mozilla y colaboradores, «HTML5,» [En línea]. Available: <https://developer.mozilla.org/es/docs/HTML/HTML5>. [Último acceso: 6 septiembre 2019].
- [36] Mozilla, «CSS3,» [En línea]. Available: <https://developer.mozilla.org/es/docs/Archive/CSS3>. [Último acceso: 6 septiembre 2019].
- [37] MDN Mozilla, «¿Qué es JavaScript?,» [En línea]. Available: https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/Qué_es_JavaScript. [Último acceso: 6 septiembre 2019].
- [38] threejs.org, «Three.js,» [En línea]. Available: <https://threejs.org>. [Último acceso: 6 septiembre 2019].

- [39] WIKIPEDIA, «JSON,» [En línea]. Available: <https://es.wikipedia.org/wiki/JSON>. [Último acceso: 6 septiembre 2019].
- [40] Dash Industry Forum, «Dash.js,» [En línea]. Available: <https://dashif.org/dash.js/>. [Último acceso: 6 septiembre 2019].
- [41] N. Foundation, «Node.js,» [En línea]. Available: <https://nodejs.org/es/>. [Último acceso: 6 septiembre 2019].
- [42] WIKIPEDIA, «Servidor HTTP Apache,» [En línea]. Available: https://es.wikipedia.org/wiki/Servidor_HTTP_Apache. [Último acceso: 6 septiembre 2019].
- [43] FFmpeg, «About FFmpeg,» [En línea]. Available: <https://ffmpeg.org/about.html>. [Último acceso: 6 septiembre 2019].
- [44] GPAC Multimedia Open Source Project, «What is GPAC?,» [En línea]. Available: <https://gpac.wp.imt.fr/home/>. [Último acceso: 6 septiembre 2019].
- [45] Axiomatic Systems LLC, «Bento4 MP4 & DASH Class Library, SDK and Tools,» [En línea]. Available: <https://www.bento4.com>. [Último acceso: 6 septiembre 2019].
- [46] Fraunhofer FOKUS, «node-hbbtv,» [En línea]. Available: <https://github.com/fraunhoferfokus/node-hbbtv>. [Último acceso: 6 septiembre 2019].
- [47] GStreamer LGPL, «gstreamer open source multimedia framework,» [En línea]. Available: <https://gstreamer.freedesktop.org>.
- [48] WIKIPEDIA, «Query string,» [En línea]. Available: https://en.wikipedia.org/wiki/Query_string.
- [49] F. B. B. R. A. S. Mario Montagud, «How to perform AMP? Cubic adjustments for improving the QoE,» *Comput. Commun.*, vol. 103, pp. 61–73, 2017.
- [50] Geo-F / X, «Texture Atlases,» [En línea]. Available: <https://www.geofx.com/graphics/nehe-three-js/lessons17-24/lesson17/lesson17.html>. [Último acceso: 6 septiembre 2019].
- [51] WIKIPEDIA, «UV mapping,» [En línea]. Available: https://en.wikipedia.org/wiki/UV_mapping. [Último acceso: 6 septiembre 2019].
- [52] T. L. FOUNDATION, «netem,» [En línea]. Available: <https://wiki.linuxfoundation.org/networking/netem>.
- [53] J. Brooke, «SUS - A quick and dirty usability scale,» Redhatch Consulting Ltd..
- [54] P. K. a. J. M. A. Bangor, «Determining what individual SUS scores mean: adding an adjective rating scale,» *Usability Stud.*, vol. 4, n° 3, pp. 114-123, 2009.
- [55] G. P. a. P. P. L. Zelnik-Manor, «Squaring the circle in panoramas,» *EEE International Conference on Computer Vision*, 2005.
- [56] Y. L. a. Y. Z. Z. Chen, «Recent advances in omnidirectional video coding for virtual reality: Projection and evaluation,» *Signal Processing*, 2018.
- [57] M. H. a. V. Swaminathan, «Adaptive 360 VR Video Streaming: Divide and Conquer,» *IEEE International Symposium on Multimedia (ISM)*, 2016.